第3章

数据挖掘的常用算法和工具

观看视频

【本章要点】

1. 数据挖掘与数据仓库和数据集市的区别与联系。
2. 数据挖掘理论知识。
3. 数据挖掘的常用算法。
4. 数据挖掘的工具。

3.1 数据仓库、数据集市与数据挖掘的关系

3.1.1 数据仓库

仓库是来自多个源的数据的存储库，它可通过 Internet 将不同的数据库连接起来，并将数据全部或部分复制到一个数据存储中心。数据仓库倾向于一个逻辑的概念，它建立在一定数量的数据库之上，这些数据库在物理上可以是分开的，甚至可以属于不同的国家。数据仓库通过 Internet 打破了地域界线，将数据库合成一个逻辑整体，把一个海量的数据库展现在用户面前。数据仓库作为服务于企业级的应用，概括来说为用户提供了以下 4 个方面的优越性：

(1) 减轻系统负担，简化日常维护和管理。

(2) 改进数据的完整性、兼容性和有效性。

(3) 提高数据存取的效率。

(4) 提供简单、统一的查询和报表机制。

数据仓库的应用开始于 1995 年。随着面向对象技术在数据库领域的应用,出现了面向对象数据仓库技术,例如 HP 公司 1996 年 5 月底发布的 HPDEPOT/J 面向对象的动态仓库技术。与典型的数据仓库系统不同,HPDEPOT/J 动态仓库并不要求把数据复制到一个数据存储中心,所有的数据还在它原来的存储地,从而避免了对额外磁盘存储设施和相关的行政管理费用的需求。HPDEPOT/J 动态仓库技术不仅增进了公司数据在转变成 Internet/Intranet 可用数据过程中的安全性,在此基础上还可把来自多个领域的数据存储资料和商业逻辑连接起来,生成商业对象,Java 程序设计员可以用这个对象产生交互的 Web 页和其他应用程序。数据类型兼容的问题在 HPDEPOT/J 中也得到了很好的解决,它能和几乎所有的数据库相连,包括 Sybase、Oracle、Informix 和 IBM 的数据库。

3.1.2　数据集市

数据仓库作为企业级应用,其涉及的范围和投入的成本常常是巨大的,它的建设很容易形成高投入、慢进度的大项目。这一切都是部门/工作组所不希望看到和不能接受的。部门/工作组要求在公司内部获得一种适合自身应用、容易使用,且自行定向、方便高效的开放式数据接口工具。与数据仓库相比,这种工具应更紧密集成、拥有完整的图形用户接口和更吸引人的价格。正是部门/工作组的这种需求使数据集市应运而生。然而,业界在数据集市应该是怎样的问题上意见不一,各公司对它的定义差别极大。目前,对于数据集市大家普遍能接受的描述简要概括为:数据集市是一种更小、更集中的数据仓库,它为公司提供了一条部门/工作组级的分析商业数据的廉价途径。数据集市应该具备的特性包括:规模小、面向特定的应用、面向部门/工作组、快速实现、投资规模小、易使用、全面支持异种机平台等。用户可根据自己的需求,以自己的方式来建立数据集市。不论是以自上而下还是自下而上的方式建立数据集市,最重要的都是保证数据集市间能相互对话,彼此不能沟通的数据集市是没用的。另外,允许人们经 WWW 访问数据集市,使之为更多的用户提供数据访问,也是必不可少的功能。当前,全世界对数据仓库总投资的一半以上在数据集市上。

IBM 和 Sybase 公司 1997 年第一季度携手推出的数据集市方案,其硬件以 IBMRS/6000 为基础,软件采用 SybaseQuickStartDataMart。该方案集成了功能强大的决策支持数据库服务器,以及用来从大型机系统和其他业务系统汲取数据的工具软件和多种供用户选择的查询工具软件。该数据集市方案提供了快速统计和分析数据的能力,帮助用户迅速将业务数据转换为企业竞争情报,利用已有的数据获得重要的竞争优势或找到进入新市场的解决途径。而不甘落后的 Oracle 公司于 1997 年 11 月正式发售其首先运行在

Windows NT 环境上的数据集市套件 DataMartSuite 市场销售版。Oracle 数据集市套件 DataMartSuite 市场销售版是一种全面的解决方案，即便是毫无此方面工作经验的用户，也可以极方便地安装和使用。现在的用户已不满足于功能单一的产品，他们需要的是更全面、集成所有必需技术的工具包产品，而 Oracle 数据集市套件正是针对这种需求而专门设计的。

3.1.3 数据挖掘

数据挖掘是从数据库或数据仓库中发现并提取隐藏在其中的信息的一种新技术。它建立在数据库，尤其是数据仓库的基础之上，面向非专业用户，定位于桌面，支持即兴的随机查询。数据挖掘技术能自动分析数据，对它们进行归纳性推理和联想，寻找数据间内在的某些关联，从中发掘出潜在的、对信息预测和决策行为起着重要作用的模式，从而建立新的业务模型，以达到帮助决策者制定市场策略，做出正确决策的目的。数据挖掘技术涉及数据库、人工智能（Artificial Intelligence，AI）、机器学习、神经计算和统计分析等多种技术，它使决策支持系统（Decision Supporting System，DSS）跨入了一个新的阶段。

1996 年 12 月，美国 Business Objects 公司宣布推出业界面向主流商业用户的数据挖掘解决方案——BusinessMiner。BusinessMiner 采用了基于直觉决定的树状技术，提供了简单易懂的数据组织形式，使用图形化方式描述数据关系，通过百分比和流程表等简单易用的用户界面告诉用户有关的数据信息。BusinessMiner 能对从数据仓库中传来的数据自动地进行挖掘分析工作，剖析任意层面数据的内在联系，最终确定商业发展趋势和规律。

3.2 数据挖掘理论简介

本节以美国 SAS 数据挖掘系统为例介绍数据挖掘理论。目前很多企业或组织已经有了成功的 MIS 系统、CMIS 系统，或者有了大量卓有成效的过程控制系统，甚至是办公自动化系统。其中的数据体系对应着一项项事务处理和一个个控制环节，它们定能完善地支持其原有的工作。当用户从企业级的角度来审视，并想进一步分析处理时，可能会感到这些数据过于分散，数量越来越大，并且难以整合。美国数据挖掘技术开拓者 Gregory Piatetsky-Shapiro 曾戏言说："原来曾希望计算机系统成为我们智慧的源泉，但从中涌出的却是洪水般的数据！"其实不必埋怨数据太多，也不必埋怨原来的数据结构不好，它们是适应原有工作任务的，只是不适合用户现在的要求而已。要支持用户的企业级决策，就需要"洪水般的数据"，但是要面向企业级的工作任务进行重组。SAS 有连续

两年获奖的数据仓库系统支持用户进行数据重组，并以全新的数据、信息的结构形式支持用户全新的工作方式。建立数据仓库就是进一步有成效地进行数据挖掘的基础。

要看清企业或组织运作的状况，第一步就是能查询到反映用户所关心的事情的相应数据、信息。以SAS的多维数据库产品MDDB构造的数据仓库从物理结构上保证了用户查询的速度以及方便性。E. F. Codd在提出联机分析处理（Online Analytical Processing，OLAP）概念时，多维数据结构是实现其任务的第一项要求。一些简单的决策支持所需要的就是有针对性的数据。在数据重组后的数据仓库中还建立了所谓的数据市场（Data Mart），它就可以针对决策支持的需要而设计，其中还可综合不同层次的汇总数据和跨数据仓库主题的数据。

SAS软件研究所对数据挖掘所下的定义是：数据挖掘是按照既定的业务目标，对大量的企业数据进行探索，揭示其中隐藏的规律性并进一步将之模型化的先进、有效的方法。

对数据的探索、挖掘首先要有一个明确的业务目标。一组生产数据可进行生产能力的分析，可进行生产成本核算的分析，也可进行影响产品质量诸因素的分析。目标决定了此后数据挖掘过程的各种运作，并导引了运作的方向。虽然说数据挖掘的业务目标在过程中不是不可修正的，也应当在工作进程中不断地进一步明确化，但其基本原则内容要保持稳定不变，否则数据挖掘工作是难以有效地进行的。

这里所指的大量企业数据最好是按照数据仓库的概念重组过的，在数据仓库中的数据、信息才能最有效地支持数据挖掘。假如所取用的数据并不足以反映企业的真实情况，当然也不可能挖掘出有用的规律。数据仓库的数据重组，首先是从企业正在运行的计算机系统中完整地将数据取出来。所谓完整，首先决策支持目标所涉及的各个环节不能有遗漏，其次各个环节的数据要按一定的规则有机、准确地衔接起来。从决策支持的主题来看，这些重新组织过的数据以极易取用的数据结构方式全面地描述了该主题。

有了反映业务主题全貌的数据后，在进行数据的分析、探索时，对于不同的人，可能会采用不同的方式和方法。Gartner Group在评价数据挖掘工具时，也特别提到了面对各种不同类型的人员的可伸缩性和完整性。SAS支持各层次的用户：

（1）业务水平和数学水平比较一般的人员，对这样的用户提供方便的数据查询是非常重要的。实际上早期的决策支持主要就是对数据查询的支持，可能也要做一些简单的数理统计分析。若统计分析的要求是较明确的，则可以事先做好，向用户提供统计分析的结果。这些可以做成SAS数据仓库中的信息市场（Information Mart）。对应用户随机的需求，应当提供方便进行菜单式选择的工具。

（2）业务水平较高，但数学水平一般，且没有时间和兴趣再钻研数学方法的人员。除了以上资源外，还应提供能简便地实现各种常用的数理统计的工具，让用户不必受累于繁杂的过程，通过简单的需求设定即可执行他们需要的操作。

（3）有计算机和数学知识，但对业务的熟悉程度一般的人员。对他们要提供较全面的数据处理工具，如数理统计、聚类分析、决策树、人工神经元网络方面的工具。

（4）对有很深计算机和数学造诣的数据分析专家，不仅要提供上述环境，还要提供实现各种算法的工具和开发平台。

SAS 系统提供了适合各类人员使用的既完整又有伸缩性的模块化的工具。

通过探索和模型化所得的结果可分成两种类型：一种是描述型的，另一种是预测型的。描述型的结果是指通过数据挖掘量化地搞清了业务目标的现状。例如在原来的工艺规程允许的范围内，生产出来的产品质量水平波动很大。通过数据挖掘找出了同一种产品在什么条件下产出的质量比较好，在什么条件下产出的质量比较差，同时通过数据挖掘描述清楚了产品质量高低的规律性，这就为修改原来的工艺规程提供了决策支持依据。

通过数据挖掘还可以建立企业或某个过程的各种不同类型的模型。这些模型不仅能描述当前的现状和规律，利用它还可以预测当条件变化后可能发生的状况。这就为企业开发新产品，甚至为企业业务重组提供了决策支持依据。

在世界走向信息化的今天，充分利用企业的信息资源，挖掘企业和所对应市场运作的规律性，以不断提高企业的经济效益，是先进企业的必由之路。世界有名的 Gartner Group 咨询顾问公司预计：不久的将来，先进的大企业将会设置“统一数据分析专家”的工作岗位。

在以 SAS 数据仓库和数据挖掘应用获奖的美国 LTV 钢铁公司阐述其获奖文章的题目是“DW＋DM＝ $aving”，即在企业中建立数据仓库进行数据挖掘就是挖取企业的经济效益。

正如你拿个镐在山上挖几下不能算是开采矿山一样，用数理统计方法或人工神经元网络进行数据分析也不能说是在进行数据挖掘。要开采矿山，首先要按照人类千百年来总结的经验所形成的理论规律来找矿；发现矿藏后，还要根据其实际地质情况，有针对性地采用相应的方法最有效地挖掘，才能获得有价值的宝藏。同样，要想有效地进行数据挖掘，必须有好的工具和一整套妥善的方法论。可以说在数据挖掘中，你采用的工具、使用工具的能力，以及在数据挖掘过程中的方法论，在很大程度上决定了你能开拓的成果。SAS 研究所不仅有丰富的工具供你选用，而且在多年的数据处理研究工作中积累了一套行之有效的数据挖掘方法论——SEMMA，通过使用 SAS 技术进行数据挖掘，我们愿意和你分享这些经验：

- Sample——数据抽样。
- Explore——数据特征探索、分析和预处理。
- Modify——问题明确化、数据调整和技术选择。
- Model——模型的研发、知识的发现。

• Assess——模型和知识的综合解释和评价。

1. Sample——数据抽样

当进行数据挖掘时，首先要从企业的大量数据中取出一个与你要探索的问题相关的样板数据子集，而不是动用全部企业数据。这就像要开采出来矿石，首先要进行选矿一样。通过数据样本的精选，不仅能减少数据处理量，节省系统资源，还能通过数据的筛选，使你想要反映的规律性更加凸显出来。

在进行数据抽样时，要把好数据的质量关。在任何时候都不要忽视数据的质量，即使你是从一个数据仓库中进行数据抽样，也不要忘记检查其质量如何。因为通过数据挖掘是要探索企业运作的规律性的，如果原始数据有误，还谈什么从中探索规律性。若你真的从中探索出来了“规律性”，再依此指导工作，则很可能是在进行误导。若你是从正在运行着的系统中进行数据抽样，则更要注意数据的完整性和有效性。再次提醒你在任何时候都不要忽视数据的质量，慎之又慎！

从巨大的企业数据母体中取出哪些数据作为样本数据呢？这要依你所要达到的目标来分别采用不同的办法：如果你要进行过程的观察、控制，这时可进行随机抽样，然后根据样本数据对企业或其中某个过程的状况做出估计。SAS 不仅支持这一抽样过程，还可对所取出的样本数据进行各种例行的检验。若你想通过数据挖掘得出企业或其某个过程的全面规律性，则必须获得在足够广泛的范围变化的数据，以使其有代表性。你还应当从实验设计的要求来考察所抽样数据的代表性。这样才能通过此后的分析研究得出反映本质规律性的结果。利用它支持你进行决策才是真正有效的，并能使企业进一步获得技术、经济效益。

2. Explore——数据特征探索、分析和预处理

前面所叙述的数据抽样，多少是带着人们对如何达到数据挖掘目的的先验认识进行操作的。当我们拿到一个样本数据集后，它是否达到我们原来设想的要求，其中有没有什么明显的规律和趋势，有没有出现你从未设想过的数据状态，各因素之间有什么相关性，它们可分为哪些类别等，这些都是首先要探索的内容。

进行数据特征的探索、分析，最好能够进行可视化的操作。SA 有 SAS/INSIGHT 和 SAS/SPECTRAVIEW 两个产品给用户提供了可视化数据操作的强有力的工具、方法和图形。它们不仅能进行各种不同类型的统计分析显示，还能进行多维、动态甚至旋转的显示。

这里的数据探索就是我们通常进行的深入调查的过程。你最终要达到的目的可能是搞清多因素相互影响的、十分复杂的关系。但是，这种复杂的关系不可能一下子建立起来。一开始，可以先观察众多因素之间的相关性，再按其相关的程度了解它们之间相互作用的情况。这些探索、分析并没有一成不变的操作规律性，相反，要有耐心地反复试

探，仔细观察。在此过程中，你原来的专业技术知识是非常有用的，它会帮助你进行有效的观察。但是，你也要注意，不要让你的专业知识束缚了你对数据特征观察的敏锐性，可能实际存在着你的先验知识认为不存在的关系。假如你的数据是真实可靠的话，那么你绝对不要轻易地否定数据呈现给你的新关系，很可能在这里就发现了新知识。有了它，也许会引导你在此后的分析中得出比原有的认识更加符合实际规律性的知识。假如在你的操作中出现了这种情况，应当说，你的数据挖掘已挖到了有效的矿脉。

在这里要提醒你的是要有耐心，做几种分析，就发现重大成果是不大可能的。所幸的是，SAS 向你提供了强有力的工具，它可跟随你的思维，可视化、快速地做出反应，免除了数学的复杂运算过程和编制结果展现程序的烦恼以及对你思维的干扰。这就使你的数据分析过程集聚于你的业务领域，并使你的思维保持一个集中的较高级的活动状态，从而加速你的思维过程，提高你的思维能力。

3. Modify——问题明确化、数据调整和技术选择

通过上述两个步骤的操作，你对数据的状态和趋势可能有了进一步的了解。对你原来要解决的问题可能会更加明确，这时要尽可能对解决问题的要求进一步量化。问题越明确，越能进一步量化，问题就向解决它的方向多前进一步。这是十分重要的。因为原来的问题很可能是诸如质量不好、生产率低等模糊的问题，没有问题的进一步明确，你简直无法进行有效的数据挖掘操作。

在问题进一步明确的基础上，你就可以按照问题的具体要求来审视你的数据集，看它是否适应你的问题需要。Gartner Group 在评论当前一些数据挖掘产品时特别强调指出：在数据挖掘的各个阶段，数据挖掘的产品都要使所使用的数据和所建立的模型处于十分易于调整、修改和变动的状态，这样才能保证数据挖掘有效地进行。

针对问题的需要可能要对数据进行增删，也可能要按照你对整个数据挖掘过程的新认识组合或者生成一些新的变量，以体现对状态的有效描述。SAS 对数据强有力的存取、管理和操作的能力保证了对数据的调整、修改和变动的可能性。若使用了 SAS 的数据仓库产品技术，则可以进一步保证有效、方便地进行这些操作。

在问题进一步明确，数据结构和内容进一步调整的基础上，下一步数据挖掘应采用的技术手段就更加清晰、明确了。

4. Model——模型的研发、知识的发现

这一步是数据挖掘工作的核心环节。虽然数据挖掘模型化工作涉及非常广阔的技术领域，但对 SAS 研究所来说并不是一件新鲜事。自从 SAS 问世以来，就一直是统计模型市场领域的领头羊，而且年年提供新产品，并以这些产品体现业界技术的最新发展。按照 SAS 提出的 SEMMA 方法论走到这一步时，你对应采用的技术已有了较明确的方向，你的数据结构和内容也有了充分的适应性。SAS 在这时向你提供了充分的可选择的

技术手段，如广泛的数理统计方法、人工神经元网络、决策树等。

正如 Gartner Group 评论中所指出的：数理统计方法还是数据挖掘工作中最常用的技术手段。SAS 的 SAS/STAT 软件包覆盖了所有的数理统计方法，并成为国际上统计分析领域的标准软件。SAS/STAT 提供了十多个过程可进行各种不同类型模型、不同特点数据的回归分析，如正交回归、响应面回归、Logistic 回归、非线性回归等，且有多种形式的模型化方法可供选择。可处理的数据有实型数据、有序数据和属性数据，并能产生各种有用的统计量和诊断信息。在方差分析方面，SAS/STAT 为多种试验设计模型提供了方差分析工具，它还有处理一般线性模型和广义线性模型的专用过程。在多变量统计分析方面，SAS/STAT 为主成分分析、典型相关分析、判别分析和因子分析提供了许多专用过程。SAS/STAT 提供多种聚类准则的聚类分析方法，可利用其进行生存分析等对客户保有程度分析特别有用的分析。同时，SAS/ETS 提供了丰富的计量经济学和时间序列分析方法，是研究复杂系统和进行预测的有力工具。它提供方便的模型设定手段以及多样的参数估计方法。实际上，SAS 的数理统计工具不仅能揭示企业已有数据间的新关系、隐藏着的规律性，还能反过来预测它的发展趋势，或是在一定条件下将会出现什么结果。

SAS 以 GUI 式的友好界面提供了人工神经元网络的应用环境。一般情况下，人工神经元网络对数据处理的要求比较多，在处理上资源的消耗也比较大。但在 SAS 的集成环境下，有规范的数据维护、管理机制，可在诸如 Client/Server 等综合调度环境中运行，这就保证了你的人工神经元网络应用更顺畅地实现。

人工神经元网络和决策树的方法结合起来可用于从相关性不强的多变量中选出重要的变量。SAS 还支持卡方自动交叉检验（Chi-Squared Automatic Interaction Detector，CHAID），分类和回归树（Classification And Regression Tree，CART）的软件包也即将交付使用。在你的数据挖掘中使用哪一种方法，用 SAS 软件包中的什么方法来实现，这主要取决于你的数据集的特征和你要实现的目标。实际上，这种选择也不一定是唯一的。好在 SAS 软件的运行效率十分高，你不妨多试几种方法，从实践中选出最适合你的方法和软件。

随着业界方法研究的进展，SAS 会不断向你提供实现它们的软件包，这将支持你的数据挖掘工作可持续地发展。

5. Assess——模型和知识的综合解释和评价

从上述过程中将会得出一系列的分析结果、模式或模型。若能得出一个直接的结论当然很好，但更多时候会得出对目标问题多侧面的描述。这时就要能很好地综合它们的影响规律性，提供合理的决策支持信息。所谓合理，实际上往往是要你在所付出的代价和达到预期目标的可靠性的平衡上做出选择。假如在你的数据挖掘过程中就预见到最后要进行这样的选择的话，那么你最好把这些平衡的指标尽可能地量化，以利于你的综

合抉择。

你提供的决策支持信息适用性如何，这显然是十分重要的问题。除了在数据处理过程中 SAS 软件提供给你的许多检验参数外，评价的一种办法是直接使用你原来建立模型的样板数据来进行检验。假如这一关就不能通过的话，那么你的决策支持信息的价值就不太大了。一般来说，在这一步应得到较好的评价。这说明你确实从这批数据样本中挖掘出了符合实际的规律性。

另一种办法是另外找一批数据，已知这些数据是反映客观实际的规律性的。这次的检验效果可能会比前一种差，差多少是要注意的。如果差到你不能容忍的程度，就要考虑第一次构建的样本数据是否具有充分的代表性，或是模型本身不够完善。这时候可能要对前面的工作进行反思了。如果这一步也得到了肯定的结果，那么你的数据挖掘应得到很好的评价。

还有一种办法是在实际运行的环境中取出新鲜数据进行检验。例如在一个应用实例中就进行了一个月的现场实际检验。

3.3 数据挖掘的常用算法

数据挖掘涉及的学科领域和方法很多，有多种分类法。根据挖掘任务，可分为分类或预测模型发现，数据抽取、聚类、关联规则发现，序列模式发现，依赖关系或依赖模型发现，异常和趋势发现等；根据挖掘对象，可分为关系数据库、面向对象数据库、空间数据库、时态数据库、文本数据源、多媒体数据库、异质数据库、遗产数据库以及环球网 Web；根据挖掘方法，可粗分为机器学习方法、统计方法、神经网络方法和数据库方法。在机器学习中，可细分为归纳学习方法（决策树、规则归纳等）、基于范例学习、遗传算法等。在统计方法中，可细分为回归分析（多元回归、自回归等）、判别分析（贝叶斯判别、费希尔判别、非参数判别等）、聚类分析（系统聚类、动态聚类等）、探索性分析（主元分析法、相关分析法等）等。在神经网络方法中，可细分为前向神经网络（BP 算法等）、自组织神经网络（自组织特征映射、竞争学习等）等。数据库方法主要是多维数据分析或 OLAP 方法，另外还有面向属性的归纳方法。

这里将主要从挖掘任务和挖掘方法的角度讨论数据抽取、分类发现、聚类和关联规则发现等常用算法。

3.3.1 数据抽取

数据抽取的目的是对数据进行浓缩，给出它的紧凑描述。传统的也是最简单的数据抽取方法是计算出数据库的各个字段上的求和值、平均值、方差值等统计值，或者用直方

图、饼状图等图形方式表示。数据挖掘主要从数据泛化的角度来讨论数据抽取。数据泛化是一种把数据库中的有关数据从低层次抽象到高层次的过程。由于数据库上的数据或对象所包含的信息总是最原始、基本的信息(这是为了不遗漏任何可能有用的数据信息),人们有时希望能从较高层次的视图上处理或浏览数据,因此需要对数据进行不同层次的泛化以适应各种查询要求。数据泛化目前主要有两种技术:多维数据分析方法和面向属性的归纳方法。

多维数据分析方法是一种数据仓库技术,也称作联机分析处理。数据仓库是面向决策支持的、集成的、稳定的、不同时间的历史数据集合。决策的前提是数据分析。在数据分析中经常要用到诸如求和、总计、平均、最大、最小等汇集操作,这类操作的计算量特别大。因此,一种很自然的想法是,把汇集操作结果预先计算并存储起来,以便于决策支持系统的使用。存储汇集操作结果的地方称作多维数据库。多维数据分析技术已经在决策支持系统中获得了成功的应用,如著名的 SAS 数据分析软件包、Business Object 公司的决策支持系统 Business Object 以及 IBM 公司的决策分析工具都使用了多维数据分析技术。

采用多维数据分析方法进行数据抽取,针对的是数据仓库,数据仓库存储的是脱机的历史数据。为了处理联机数据,研究人员提出了一种面向属性的归纳方法。它的思路是,直接对用户感兴趣的数据视图(用一般的 SQL 查询语言即可获得)进行泛化,而不是像多维数据分析方法那样预先存储好了泛化数据。方法的提出者将这种数据泛化技术称为面向属性的归纳方法。原始关系经过泛化操作后得到的是一个泛化关系,它从较高的层次上总结了在低层次上的原始关系。有了泛化关系后,就可以对它进行各种深入的操作而生成满足用户需要的知识,如在泛化关系基础上生成特性规则、判别规则、分类规则以及关联规则等。

在实际应用中,数据源较多采用的是关系数据库。从数据库中抽取数据一般有以下几种方式。

1. 全量抽取

全量抽取类似于数据迁移或数据复制,它将数据源中的表或视图的数据原封不动地从数据库中抽取出来,并转换成自己的 ETL 工具可以识别的格式。全量抽取比较简单。

2. 增量抽取

增量抽取指抽取自上次抽取以来数据库中要抽取的表中新增、修改、删除的数据。在 ETL 使用过程中,增量抽取比全量抽取应用更广。如何捕获变化的数据是增量抽取的关键。对捕获方法一般有两点要求:一是准确性,能够将业务系统中的变化数据准确地捕获到;二是性能,尽量减少对业务系统造成太大的压力,影响现有的业务。在增量数据的抽取中,常用的捕获变化数据的方法如下。

（1）触发器：在要抽取的表上建立需要的触发器，一般要建立插入、修改、删除 3 个触发器，每当源表中的数据发生变化时，就被相应的触发器将变化的数据写入一个临时表，抽取线程从临时表中抽取数据。触发器方式的优点是数据抽取的性能较高，缺点是要求在业务数据库中建立触发器，对业务系统有一定的性能影响。

（2）时间戳：它是一种基于递增数据比较的增量数据捕获方式，在源表上增加一个时间戳字段，在系统中更新、修改表数据的时候，同时修改时间戳字段的值。当进行数据抽取时，通过比较系统时间与时间戳字段的值来决定抽取哪些数据。有的数据库的时间戳支持自动更新，即表的其他字段的数据发生改变时，自动更新时间戳字段的值。有的数据库不支持时间戳的自动更新，这就要求业务系统在更新业务数据时，手工更新时间戳字段。同触发器方式一样，时间戳方式的性能也比较好，数据抽取相对清楚简单，但对业务系统有很大的倾入性（加入额外的时间戳字段），特别是对不支持时间戳的自动更新的数据库，还要求业务系统进行额外的更新时间戳操作。另外，无法捕获对时间戳以前数据的删除和更新操作，在数据准确性上受到了一定的限制。

（3）全表比对：典型的全表比对的方式是采用 MD5 校验码。ETL 工具事先为要抽取的表建立一个结构类似的 MD5 临时表，该临时表记录源表主键以及根据所有字段的数据计算出来的 MD5 校验码。每次进行数据抽取时，对源表和 MD5 临时表进行 MD5 校验码的比对，从而决定源表中的数据是新增、修改还是删除，同时更新 MD5 校验码。MD5 方式的优点是对源系统的倾入性较小（仅需要建立一个 MD5 临时表），但缺点也是显而易见的，与触发器和时间戳方式中的主动通知不同，MD5 方式是被动地进行全表数据的比对，性能较差。当表中没有主键或唯一列且含有重复记录时，MD5 方式的准确性较差。

（4）日志对比：通过分析数据库自身的日志来判断变化的数据。Oracle 的改变数据捕获（Changed Data Capture，CDC）技术是这方面的代表。CDC 技术的特性是在 Oracle 的 9i 数据库中引入的。CDC 技术能够帮助你识别从上次抽取之后发生变化的数据。利用 CDC 技术，在对源表进行插入、更新或删除等操作的同时就可以提取数据，并且变化的数据被保存在数据库的变化表中。这样就可以捕获发生变化的数据，然后利用数据库视图以一种可控的方式提供给目标系统。CDC 体系结构基于发布者/订阅者模型。发布者捕捉变化的数据并提供给订阅者。订阅者使用从发布者那里获得的变化的数据。通常，CDC 系统拥有一个发布者和多个订阅者。发布者首先需要识别捕获变化的数据所需的源表。然后，它捕捉变化的数据并将其保存在特别创建的变化表中。它还使订阅者能够控制对变化数据的访问。订阅者需要清楚自己感兴趣的是哪些变化的数据。一个订阅者可能不会对发布者发布的所有数据都感兴趣。订阅者需要创建一个订阅者视图来访问经发布者授权可以访问的变化的数据。CDC 分为同步模式和异步模式。同步模式实时地捕获变化的数据并存储到变化表中，发布者与订阅位于同一个数据库中。异步模

式则是基于 Oracle 的流复制技术。

ETL 处理的数据源除了关系数据库外，还可能是文件，例如 TXT 文件、Excel 文件、XML 文件等。对文件数据的抽取一般是进行全量抽取，一次抽取前可保存文件的时间戳或计算文件的 MD5 校验码，下次抽取时进行比对，如果相同，则可忽略本次抽取。

3.3.2　分类发现

分类在数据挖掘中是一项非常重要的任务，目前在商业上应用最多。分类的目的是学会一个分类函数或分类模型（也常称作分类器），该模型能把数据库中的数据项映射到给定类别中的某一个。分类和回归都可用于预测。预测的目的是从历史数据记录中自动推导出对给定数据的推广描述，从而能对未来的数据进行预测。和回归方法不同的是，分类的输出是离散的类别值，而回归的输出则是连续数值。这里我们不讨论回归方法。

要构造分类器，需要有一个训练样本数据集作为输入。训练集由一组数据库记录或元组构成，每个元组是一个由有关字段（又称属性或特征）值组成的特征向量，此外，训练样本还有一个类别标记。一个具体样本的形式可为（v_1，v_2，…，v_n；c），其中 v_i 表示字段值，c 表示类别。

分类器的构造方法有统计方法、机器学习方法、神经网络方法等。统计方法包括贝叶斯法和非参数法（近邻学习或基于事例的学习），对应的知识表示则为判别函数和原型事例。机器学习方法包括决策树法和规则归纳法，前者对应的表示为决策树或判别树，后者一般为产生式规则。神经网络方法主要是 BP 算法，它的模型表示是前向反馈神经网络模型（由代表神经元的节点和代表连接权值的边组成的一种体系结构），BP 算法本质上是一种非线性判别函数。另外，最近又兴起了一种新的方法：粗糙集（Rough Set），其知识表示是产生式规则。

不同的分类器有不同的特点。有 3 种分类器评价或比较尺度：一是预测准确度；二是计算复杂度；三是模型描述的简洁度。预测准确度是用得最多的一种比较尺度，特别是对于预测型分类任务，目前公认的方法是 10 折交叉验证法。计算复杂度依赖于具体的实现细节和硬件环境，在数据挖掘中，由于操作对象是巨量的数据库，因此空间和时间的复杂度问题将是非常重要的一个环节。对于描述型的分类任务，模型描述越简洁越受欢迎。例如，采用规则表示的分类器构造法就更有用，而神经网络方法产生的结果就难以理解。

另外要注意的是，分类的效果一般和数据的特点有关，有的数据噪声大，有的有缺值，有的分布稀疏，有的字段或属性间的相关性强，有的属性是离散的，而有的是连续值或混合式的。目前普遍认为不存在某种方法适用于各种特点的数据。

所谓分类，简单来说，就是根据文本的特征或属性划分到已有的类别中。常用的分类算法包括决策树分类法、朴素贝叶斯分类（Native Bayesian Classifier）算法、基于支持向量机（Support Vector Machine，SVM）的分类器、神经网络法、k-最近邻（k-Nearest Neighbor，kNN）法、模糊分类法等。

1. 决策树

决策树是一种用于对实例进行分类的树状结构，一种依托于策略抉择而建立起来的树。决策树由节点（Node）和有向边（Directed Edge）组成。节点的类型有两种：内部节点和叶节点。其中，内部节点表示一个特征或属性的测试条件（用于分开具有不同特性的记录），叶节点表示一个分类。

一旦我们构造了一个决策树模型，以它为基础来进行分类将是非常容易的。具体的做法是，从根节点开始，对实例的某一特征进行测试，根据测试结果将实例分配到其子节点（也就是选择适当的分支），沿着该分支可能到达叶节点或者到达另一个内部节点时，就使用新的测试条件递归地执行下去，直到抵达一个叶节点。当到达叶节点时，我们便得到了最终的分类结果。

从数据产生决策树的机器学习技术叫作决策树学习，通俗点说就是决策树，这是一种依托于分类、训练的预测树，根据已知预测、归类未来。决策树学习示例如图 3.1 所示。

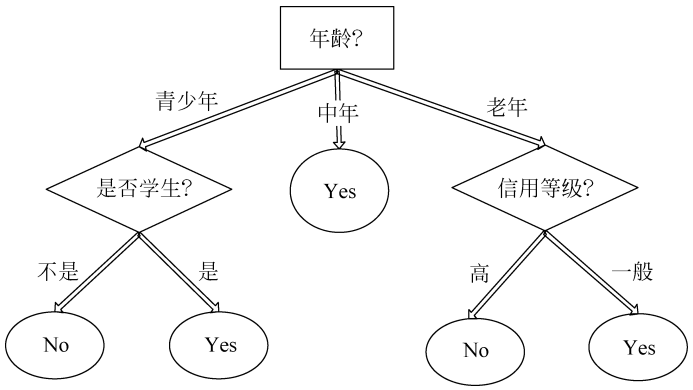


图 3.1　决策树学习示例

分类理论太过抽象，下面举两个浅显易懂的例子。

决策树分类的思想类似于找对象。现想象一个女孩的母亲要给这个女孩介绍男朋友，于是有了下面的对话。

女儿：多大年纪了？

母亲：26 岁。

女儿：长得帅不帅？

母亲：挺帅的。

女儿：收入高不高？

母亲：不算很高，中等情况。

女儿：是公务员不？

母亲：是，在税务局上班呢。

女儿：那好，我去见见。

这个女孩的决策过程就是典型的分类树决策。相当于通过年龄、长相、收入和是否公务员等将男人分为两个类别：见和不见。假设这个女孩对男人的要求是：30岁以下、长相中等以上并且是高收入者或中等以上收入的公务员，那么最终满足这些条件的才会选择去见。

2. 贝叶斯

贝叶斯(Bayes)分类算法是一类利用概率统计知识进行分类的算法，如朴素贝叶斯(Naive Bayes)算法。这些算法主要利用贝叶斯定理来预测一个未知类别的样本属于各个类别的可能性，选择其中可能性最大的一个类别作为该样本的最终类别。由于贝叶斯定理的成立本身需要一个很强的条件独立性假设前提，而此假设在实际情况中经常是不成立的，因而其分类准确性就会下降。为此就出现了许多降低独立性假设的贝叶斯分类算法，如树增强朴素贝叶斯(Tree Augmented Naive Bayes，TAN)算法，它是在贝叶斯网络结构的基础上增加属性对之间的关联来实现的。

通常，事件A在事件B的条件下的概率，与事件B在事件A的条件下的概率是不一样的；然而，这两者有确定的关系，贝叶斯定理就是这种关系的陈述。

贝叶斯定理是指概率统计中的应用所观察到的现象对有关概率分布的主观判断(先验概率)进行修正的标准方法。当分析样本大到接近总体数时，样本中事件发生的概率将接近总体中事件发生的概率。

作为一个规范的原理，贝叶斯法则对于所有概率的解释是有效的；然而，频率主义者和贝叶斯主义者对于在应用中概率如何被赋值有着不同的看法：频率主义者根据随机事件发生的频率，或者总体样本里面的个数来赋值概率；贝叶斯主义者根据未知的命题来赋值概率。

贝叶斯统计中的两个基本概念是先验分布和后验分布。

先验分布是总体分布参数θ的一个概率分布。贝叶斯学派的根本观点认为，在关于总体分布参数θ的任何统计推断问题中，除了使用样本所提供的信息外，还必须规定一个先验分布，它是在进行统计推断时不可缺少的一个要素。他们认为先验分布不必有客观的依据，可以部分地或完全地基于主观信念。后验分布是根据样本分布和未知参数的先验分布，用概率论中求条件概率分布的方法，求出的在样本已知的情况下，未知参数的条件分布。因为这个分布是在抽样以后才得到的，故称为后验分布。贝叶斯推断方法的

关键是任何推断都必须且只需根据后验分布，而不能再涉及样本分布。

贝叶斯公式为：

$$P(A \cap B) = P(A) \cdot P(B \mid A) = P(B) \cdot P(A \mid B)$$

$$P(A \mid B) = P(B \mid A) \cdot P(A)/P(B)$$

其中：

(1) $P(A)$是 A 的先验概率或边缘概率，称作“先验”是因为它不考虑 B 因素。

(2) $P(A|B)$是已知 B 发生后 A 的条件概率，也称作 A 的后验概率。

(3) $P(B|A)$是已知 A 发生后 B 的条件概率，也称作 B 的后验概率，这里称作似然度。

(4) $P(B)$是 B 的先验概率或边缘概率，这里称作标准化常量。

(5) $P(B|A)/P(B)$称作标准似然度。

贝叶斯法则又可表述为：

后验概率 =(似然度 × 先验概率)/ 标准化常量 = 标准似然度 × 先验概率

$P(A|B)$随着 $P(A)$和 $P(B|A)$的增长而增长，随着 $P(B)$的增长而减少，即如果 B 独立于 A 时被观察到的可能性越大，那么 B 对 A 的支持度越小。

贝叶斯公式为利用搜集到的信息对原有判断进行修正提供了有效手段。在采样之前，经济主体对各种假设有一个判断(先验概率)，关于先验概率的分布，通常可根据经济主体的经验判断确定(当无任何信息时，一般假设各先验概率相同)，较复杂精确的可利用最大熵技术、边际分布密度以及相互信息原理等方法来确定先验概率分布。

3. 人工神经网络

人工神经网络(Artificial Neural Network，ANN)是一种应用类似于大脑神经突触连接的结构进行信息处理的数学模型。在这种模型中，大量的节点(或称神经元，或单元)之间相互连接构成网络，即神经网络，以达到处理信息的目的。神经网络通常需要进行训练，训练的过程就是网络进行学习的过程。训练改变了网络节点的连接权的值使其具有分类的功能，经过训练的网络就可用于对象的识别。

目前，神经网络已有上百种不同的模型，常见的有 BP 网络、径向基 RBF 网络、Hopfield 网络、随机神经网络(Boltzmann 机)、竞争神经网络(Hamming 网络、自组织映射网络)等。然而，当前的神经网络仍然普遍存在收敛速度慢、计算量大、训练时间长和不可解释等缺点。人工神经网络的基本结构如图 3.2 所示。

4. *K* 最近邻

K 最近邻算法是一种基于实例的分类方法。该方法就是找出与未知样本 x 距离最近的 K 个训练样本，看这 K 个样本中多数属于哪一类，就把 x 归为哪一类。K 最近邻算法是一种懒惰学习方法，它存放样本，直到需要分类时才进行分类，如果样本集比较复

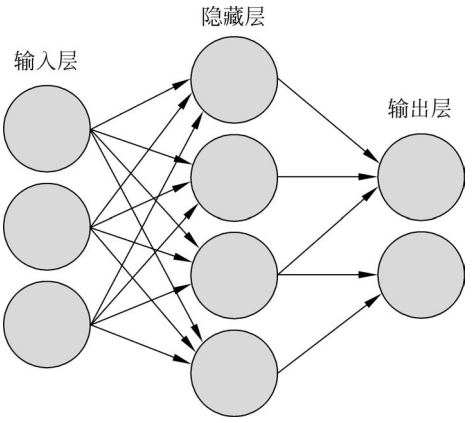


图 3.2　人工神经网络的基本结构

杂，可能会导致很大的计算开销，因此无法应用到实时性很强的场合。

5. 支持向量机

支持向量机(Support Vector Machine，SVM)的主要思想是建立一个最优决策超平面，使得该平面两侧距离该平面最近的两类样本之间的距离最大化，从而为分类问题提供良好的泛化能力。对于一个多维的样本集，系统随机产生一个超平面并不断移动，对样本进行分类，直到训练样本中属于不同类别的样本点正好位于该超平面的两侧，满足该条件的超平面可能有很多个，SVM 正是在保证分类精度的同时，寻找到这样一个超平面，使得超平面两侧的空白区域最大化，从而实现对线性可分样本的最优分类。

支持向量机中的支持向量(Support Vector)是指训练样本集中的某些训练点，这些点最靠近分类决策面，是最难分类的数据点。SVM 中的最优分类标准就是这些点距离分类超平面的距离达到最大值；“机”(Machine)是机器学习领域对一些算法的统称，常把算法看作一个机器或者学习函数。SVM 是一种有监督的学习方法，主要针对小样本数据进行学习、分类和预测，类似的根据样本进行学习的方法还有决策树归纳算法等。

SVM 有以下优点：

(1) 不需要很多样本。这并不意味着训练样本的绝对量很少，而是说相对于其他训练分类算法，在同样的问题复杂度下，SVM 需求的样本相对是较少的。并且由于 SVM 引入了核函数，因此对于高维的样本，SVM 也能轻松应对。

(2) 结构风险最小。这种风险是指分类器对问题真实模型的逼近与问题真实解之间的累积误差。

(3) 非线性。是指 SVM 擅长应付样本数据线性不可分的情况，主要通过松弛变量(也叫惩罚变量)和核函数技术来实现，这一部分也正是 SVM 的精髓所在。

支持向量机概述图如图 3.3 所示。

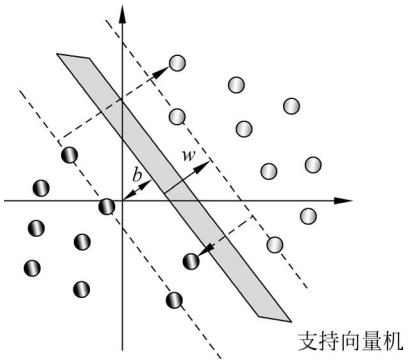


图 3.3　支持向量机概述图

6. 基于关联规则的分类

关联规则挖掘是数据挖掘中一个重要的研究领域。近年来，对于如何将关联规则挖掘用于分类问题，学者们进行了广泛的研究。关联分类方法挖掘形如 condset→C 的规则，其中 condset 是项（或属性-值对）的集合，而 C 是类标号，这种形式的规则称为类关联规则（Class Association Rules，CAR）。关联分类方法一般由两步组成：第一步用关联规则挖掘算法从训练数据集中挖掘出所有满足指定支持度和置信度的类关联规则；第二步使用启发式方法从挖掘出的类关联规则中挑选出一组高质量的规则用于分类。

3.3.3　聚类

聚类是把一组个体按照相似性归成若干类别，即“物以类聚”。它的目的是使得属于同一类别的个体之间的距离尽可能小，而不同类别的个体间的距离尽可能大。聚类分析的目标就是在相似的基础上收集数据来分类。聚类源于很多领域，包括数学、计算机科学、统计学、生物学和经济学。在不同的应用领域，很多聚类技术都得到了发展，这些技术方法被用作描述数据，衡量不同数据源间的相似性，以及把数据源分类到不同的簇中。聚类方法包括统计方法、机器学习方法、神经网络方法和面向数据库的方法。

在统计方法中，聚类称为聚类分析，它是多元数据分析的三大方法之一（其他两种是回归分析和判别分析）。它主要研究基于几何距离的聚类，如欧氏距离、明考斯基距离等。传统的统计聚类分析方法包括系统聚类法、分解法、加入法、动态聚类法、有序样品聚类法、有重叠聚类法和模糊聚类法等。这种聚类方法是一种基于全局比较的聚类，它需要考察所有的个体才能决定类的划分，因此它要求所有的数据必须预先给定，而不能动态增加新的数据对象。聚类分析方法不具有线性的计算复杂度，难以适用于数据库非常大的情况。

在机器学习中，聚类称作无监督或无教师归纳，因为和分类学习相比，分类学习的例

子或数据对象有类别标记，而要聚类的例子则没有标记，需要由聚类学习算法来自动确定。在很多人工智能文献中，聚类也称作概念聚类，因为这里的距离不再是统计方法中的几何距离，而是根据概念的描述来确定的。当聚类对象可以动态增加时，概念聚类则称为概念形成。

在神经网络中，有一类无监督学习方法：自组织神经网络方法，如 Kohonen 自组织特征映射网络、竞争学习网络等。在数据挖掘领域，见报道的神经网络聚类方法主要是自组织特征映射方法，IBM 在其发布的数据挖掘白皮书中就特别提到了使用此方法进行数据库聚类分割。

聚类与分类的不同在于，聚类要求划分的类是未知的。聚类是将数据分类到不同的类或者簇的过程，所以同一个簇中的对象有很大的相似性，而不同簇间的对象有很大的相异性。

从统计学的观点来看，聚类分析是通过数据建模简化数据的一种方法。采用 k-均值、k-中心点等算法的聚类分析工具已被加入许多著名的统计分析软件包中，如 SPSS、SAS 等。

从机器学习的角度来讲，簇相当于隐藏模式。聚类是搜索簇的无监督学习过程。与分类不同，无监督学习不依赖预先定义的类或带类标记的训练实例，需要由聚类学习算法自动确定标记，而分类学习的实例或数据对象有类别标记。聚类是观察式学习，而不是示例式学习。

聚类分析是一种探索性的分析，在分类的过程中，人们不必事先给出一个分类的标准，聚类分析能够从样本数据出发，自动进行分类。聚类分析所使用方法的不同，常常会得到不同的结论。不同研究者对于同一组数据进行聚类分析，所得到的聚类数未必一致。

从实际应用的角度来看，聚类分析是数据挖掘的主要任务之一。此外聚类可以作为一个独立的工具获得数据的分布状况，观察每一簇数据的特征，并集中对特定的聚簇集进行进一步的分析。聚类分析还可以作为其他算法（如分类和定性归纳算法）的预处理步骤。下面介绍几种常用的聚类算法。

1. BIRCH 算法

BIRCH 是一个综合的层次聚类方法。它用聚类特征和聚类特征（Clustering Feature，CF）树来概括聚类描述。描述如下：

对于一个具有 N 个 d 维数据点的簇 $\{\boldsymbol{x}_i\}$（$i=1,2,3,\cdots,N$），它的聚类特征向量定义为：

$$\mathrm{CF}=(N,\boldsymbol{LS},SS) \tag{3.1}$$

其中 N 为簇中点的个数；$\boldsymbol{LS}$ 表示 N 个点的线性和 $\left(\sum_{i=1}^{N}\boldsymbol{x}_i\right)$，反映了簇的重心，$SS$ 是数据

点的平方和$\left(\sum_{i=1}^{N}\boldsymbol{x}_i^2\right)$，反映了类直径的大小。

此外，对于聚类特征有如下定理：

定理 假设 $\mathrm{CF}_1=(N_1,\boldsymbol{LS}_1,SS_1)$与 $\mathrm{CF}_2=(N_2,\boldsymbol{LS}_2,SS_2)$分别为两个类的聚类特征，合并后的新类特征为：

$$\mathrm{CF}_1+\mathrm{CF}_2=(N_1+N_2,\boldsymbol{LS}_1+\boldsymbol{LS}_2,SS_1+SS_2) \tag{3.2}$$

该算法通过聚类特征可以方便地进行中心、半径、直径及类内距离、类间距离的运算。CF 树是一个具有两个参数分支因子 B 和阈值 T 的高度平衡树，它存储了层次聚类的聚类特征。分支因子定义了每个非叶节点孩子的最大数目，而阈值给出了存储在树的叶节点中的子聚类的最大直径。CF 树可以动态地构造，因此不要求所有的数据读入内存，还可在外存上逐个读入数据项。一个数据项总是被插入最近的叶子条目（子聚类）。如果插入后使得该叶节点中的子聚类的直径大于阈值，则该叶节点极可能有其他节点被分裂。新数据插入后，关于该数据的信息向树根传递。可以通过改变阈值来修改 CF 树的大小来控制其占用的内存容量。BIRCH 算法通过一次扫描就可以进行较好的聚类，故该算法的计算复杂度是 $O(n)$，n 是对象的数目。

2. COBWEB 算法

概念聚类是机器学习中的一种聚类方法，大多数概念聚类方法采用统计学的途径，在决定概念或聚类时使用概率度量。COBWEB 以一个分类树的形式创建层次聚类，它的输入对象用分类属性-值对来描述。

分类树和判定树不同。在分类树中，每个节点都代表一个概念，并包含该概念的一个概率描述。分类树用于概述被分在该节点下的对象。概率描述包括概念的概率和形如 $P(A_i=V_{ij}\mid C_k)$的条件概率，这里 $A_i=V_{ij}$ 是属性-值对，C_k 是概念类。在分类树某层次上的兄弟节点形成了一个划分。COBWEB 采用一个启发式估算度量——分类效用来指导树的构建。分类效用的定义如下：

$$\frac{\sum_{k=1}^{n}P(C_k)\left[\sum_i\sum_j P(A_i=V_{ij}\mid C_k)^2-\sum_i\sum_j P(A_i=V_{ij})^2\right]}{n} \tag{3.3}$$

n 是在树的某个层次上形成一个划分$\{C_1,C_2,\cdots,C_n\}$的节点、概念或“种类”的数目。分类效用回报类内相似性和类间相异性。

概率 $P(A_i=V_{ij}\mid C_k)$表示类内相似性。该值越大，共享该属性-值对的类成员比例就越大，更能预见该属性-值对是类成员。概率 $P(C_k\mid A_i=V_{ij})$表示类间相异性。该值越大，在对照类中的对象共享该属性-值对的就越少，更能预见该属性-值对是类成员。给定一个新的对象，COBWEB 沿一条适当的路径向下，修改计数，寻找可以分类该对象的最好的节点。该判定基于将对象临时置于每个节点，并计算结果划分的分类效用。产生

最高分类效用的位置应当是对象节点的一个好的选择。

3. 模糊聚类算法 FCM

聚类可以引入模糊逻辑概念。对于模糊集来说，一个数据点以一定程度属于某个类，也可以同时以不同的程度属于几个类。常用的模糊聚类算法是模糊 C 平均值(Fuzzy C-Means，FCM)算法。该算法是在传统 C 均值算法中应用模糊技术。在 FCM 算法中，用隶属度函数定义的聚类损失函数可以写为：

$$J_f = \sum_{j=1}^{c}\sum_{i=1}^{n}[\mu_j(\boldsymbol{x}_i)]^b \parallel \boldsymbol{x}_i - \boldsymbol{m}_j \parallel^2 \tag{3.4}$$

其中，$b>1$，是一个可以控制聚类结果的模糊程度的常数。要求一个样本对于各个聚类的隶属度之和为 1，即

$$\sum_{j=1}^{c}\mu_j(\boldsymbol{x}_i) = 1 \tag{3.5}$$

在条件式(3.5)下求式(3.4)的极小值，令 J_f 对 $\boldsymbol{m}_i$ 和 $\mu_j(\boldsymbol{x}_i)$ 的偏导数为 0，可得必要条件：

$$\boldsymbol{m}_j = \frac{\sum_{i=1}^{n}[\mu_j(\boldsymbol{x}_i)]^b \boldsymbol{x}_i}{\sum_{i=1}^{n}[\mu_j(\boldsymbol{x}_i)]^b}, \quad j = 1,2,\cdots,c \tag{3.6}$$

$$\mu_j(\boldsymbol{x}_i) = \frac{(1/\parallel \boldsymbol{x}_i - \boldsymbol{m}_j \parallel^2)^{1/(b-1)}}{\sum_{k=1}^{c}(1/\parallel \boldsymbol{x}_i - \boldsymbol{m}_k \parallel^2)^{1/(b-1)}}, \quad i = 1,2,\cdots,n \quad j = 1,2,\cdots,c \tag{3.7}$$

用迭代法求解式(3.6)和式(3.7)，就是 FCM 算法。

当算法收敛时，就得到了各类的聚类中心以及表示各个样本对各类隶属程度的隶属度矩阵，从而完成了模糊聚类划分。

4. 聚类方法的特征

聚类方法的特征如下：

(1) 聚类分析简单、直观。

(2) 聚类分析主要应用于探索性的研究，其分析的结果可以提供多个可能的解，选择最终的解需要研究者的主观判断和后续的分析。

(3) 无论实际数据中是否真正存在不同的类别，利用聚类分析都能得到分成若干类别的解。

(4) 聚类分析的解完全依赖于研究者所选择的聚类变量，增加或删除一些变量对最终的解都可能产生实质性的影响。

(5) 研究者在使用聚类分析时应特别注意可能影响结果的各个因素。异常值和特殊

的变量对聚类有较大影响，当分类变量的测量尺度不一致时，需要事先进行标准化处理。

5. 聚类分析的实施步骤

聚类分析的实施步骤如下：

（1）数据预处理。

（2）为衡量数据点间的相似度定义一个距离函数。

（3）聚类或分组。

（4）评估输出。

数据预处理包括选择数量、类型和特征的标度，它依靠特征选择和特征抽取，特征选择选择重要的特征，特征抽取把输入的特征转换为一个新的显著特征，它们经常被用来获取一个合适的特征集来为避免“维数灾”进行聚类。数据预处理还包括将孤立点移出数据，孤立点是不依附于一般数据行为或模型的数据，因此孤立点经常会导致有偏差的聚类结果，因此，为了得到正确的聚类，我们必须将它们剔除。

既然相似性是定义一个类的基础，那么不同数据之间在同一个特征空间相似度的衡量对于聚类步骤是很重要的，由于特征类型和特征标度的多样性，距离度量必须谨慎，它经常依赖于应用，例如，通常通过定义在特征空间的距离度量来评估不同对象的相异性，很多距离度量都应用在不同的领域，一个简单的距离度量，如 Euclidean 距离，经常被用作反映不同数据间的相异性，一些有关相似性的度量，例如 PMC 和 SMC，能够被用来特征化不同数据的概念相似性，在图像聚类上，子图图像的误差更正能够被用来衡量两个图形的相似性。

将数据对象分到不同的类中是一个很重要的步骤，数据基于不同的方法被分到不同的类中，划分方法和层次方法（层次方法的基本思想是：通过某种相似性测度计算节点之间的相似性，并按相似度由高到低排序，逐步重新连接各节点，该方法的优点是可随时停止划分）是聚类分析的两个主要方法，划分方法一般从初始划分和最优化一个聚类标准开始。Crisp Clustering 的每一个数据都属于单独的类，Fuzzy Clustering 的每个数据可能在任何一个类中，Crisp Clustering 和 Fuzzy Clustering 是划分方法的两个主要技术。划分方法聚类是基于某个标准产生一个嵌套的划分系列，它可以度量不同类之间的相似性或一个类的可分离性用来合并和分裂类，其他的聚类方法还包括基于密度的聚类、基于模型的聚类、基于网格的聚类。

评估聚类结果的质量是另一个重要的阶段，聚类是一个无管理的程序，也没有客观的标准来评价聚类结果，它是通过一个类的有效索引来评价的。一般来说，几何性质，包括类间的分离和类内部的耦合，用来评价聚类结果的质量，类的有效索引在决定类的数目时经常扮演一个重要角色，类的有效索引的最佳值被期望从真实的类数目中获取，通常决定类数目的方法是选择一个特定的类的有效索引的最佳值，这个索引能否真实地得出类的数目是判断该索引是否有效的标准，很多已经存在的标准对于相互分离的类数据

集合都能得出很好的结果，但是对于复杂的数据集，却通常行不通，例如对于交叠类的集合。

6. 聚类分析的应用

聚类分析可以作为其他算法的预处理步骤，这些算法再在生成的簇上进行处理。聚类分析可作为特征和分类算法的预处理步骤，也可将聚类结果用于进一步的关联分析。

聚类分析可作为一个独立的工具来获得数据分布的情况，观察每个簇的特点，集中对特定的某些簇进行进一步的分析。聚类分析的主要应用领域如下。

1）商业

聚类分析被用来发现不同的客户群，并且通过购买模式刻画不同的客户群的特征。

聚类分析是细分市场的有效工具，同时也可用于研究消费者行为，寻找新的潜在市场，选择实验的市场，并作为多元分析的预处理。

2）生物

聚类分析被用来进行动植物分类和对基因进行分类，获取对种群固有结构的认识。

3）地理

聚类分析能够帮助数据库厂商将获得的相似地理数据库归为同一类。

4）保险行业

聚类分析通过一个高的平均消费来鉴定汽车保险单持有者的分组，同时根据住宅类型、价值、地理位置来鉴定一个城市的房产分组。

5）因特网

聚类分析被用来在网上进行文档归类来修复信息。

6）电子商务

聚类分析在电子商务网站建设、数据挖掘中也是很重要的，通过分组聚类出具有相似浏览行为的客户，并分析客户的共同特征，可以更好地帮助电子商务网站的用户了解自己的客户，向客户提供更合适的服务。

3.3.4　关联规则发现

关联规则是形式如下的一种规则："在购买面包和黄油的顾客中，有 90％的人同时也买了牛奶(面包＋黄油＋牛奶)"。用于关联规则发现的主要对象是事务数据库，其中针对的应用则是售货数据，也称货篮数据。一个事务一般由如下几个部分组成：事务处理时间，一组顾客购买的物品，有时也有顾客标识号(如信用卡号)。

关联规则最初是针对购物篮分析(Market Basket Analysis)问题提出的。假设分店经理想更多地了解顾客的购物习惯，特别是想知道哪些商品顾客可能会在一次购物时同时购买。为回答该问题，可以对商店的顾客购买数量进行购物篮分析。该过程通过发现

顾客放入购物篮中的不同商品之间的关联，分析顾客的购物习惯。这种关联的发现可以帮助零售商了解哪些商品频繁地被顾客同时购买，从而帮助他们开发更好的营销策略。

1993 年，Agrawal 等在提出关联规则概念的同时，给出了其相应的挖掘算法 AIS，但是性能较差。1994 年，他们建立了项目集格空间理论，并依据上述两个定理提出了著名的 Apriori 算法，至今 Apriori 算法仍然作为关联规则挖掘的经典算法被广泛讨论，以后诸多的研究人员对关联规则的挖掘问题进行了大量的研究。

由于条形码技术的发展，零售部门可以利用前端收款机收集存储大量的售货数据。因此，如果对这些历史事务数据进行分析，则可为顾客的购买行为提供极有价值的信息。例如，可以帮助摆放货架上的商品（如把顾客经常同时买的商品放在一起），帮助规划市场（怎样相互搭配进货）。由此可见，从事务数据中发现关联规则，对于改进零售业等商业活动的决策非常重要。

1. 关联规则的定义

设 $I=\{i_1,i_2,\cdots,i_m\}$是一组物品集（一个商场的物品可能有上万种），D 是一组事务集（称之为事务数据库）。D 中的每个事务 T 是一组物品，显然满足 $T\subseteq I$。称事务 T 支持物品集 X，如果 $X\subseteq T$。关联规则是如下形式的一种蕴含：$X\Rightarrow Y$，其中 $X\subseteq I$，$Y\subseteq I$，且 $X\cap Y=I$。

（1）称物品集 X 具有大小为 s 的支持度，如果 D 中有 $s\%$的事务支持物品集 X。

（2）称关联规则 $X\Rightarrow Y$ 在事务数据库 D 中具有大小为 s 的支持度，如果物品集 $X\cup Y$ 的支持度为 s。

（3）称规则 $X\Rightarrow Y$ 在事务数据库 D 中具有大小为 c 的可信度，如果 D 中支持物品集 X 的事务中有 $c\%$的事务同时也支持物品集 Y。

如果不考虑关联规则的支持度和可信度，那么在事务数据库中存在无穷多的关联规则。事实上，人们一般只对满足一定的支持度和可信度的关联规则感兴趣。在文献中，一般称满足一定要求的（如较大的支持度和可信度）的规则为强规则。因此，为了发现出有意义的关联规则，需要给定两个阈值：最小支持度和最小可信度。前者即用户规定的关联规则必须满足的最小支持度，它表示一组物品集在统计意义上需满足的最低程度；后者即用户规定的关联规则必须满足的最小可信度，它反映了关联规则的最低可靠度。

在实际应用中，一种更有用的关联规则是泛化关联规则。因为物品概念间存在一种层次关系，如夹克衫、滑雪衫属于外套类，外套、衬衣又属于衣服类。有了层次关系后，可以帮助发现一些更多的有意义的规则。例如，“买外套⇒买鞋子”（此处，外套和鞋子是较高层次上的物品或概念，因而该规则是一种泛化的关联规则）。由于商店或超市中有成千上万种物品，平均来讲，每种物品（如滑雪衫）的支持度很低，因此有时难以发现有用规则，但如果考虑到较高层次的物品（如外套），则其支持度就较高，从而可能发现有用的规则。

另外，关联规则发现的思路还可以用于序列模式发现。用户在购买物品时，除了具有上述关联规律外，还有时间或序列上的规律，因为很多时候顾客会这次买这些东西，下次买同上次有关的一些东西，接着又买其他有关的东西。

2. 关联规则的挖掘过程

关联规则的挖掘过程主要包含两个阶段：第一阶段必须先从资料集合中找出所有的高频项目组（Frequent Itemsets），第二阶段再由这些高频项目组中产生关联规则（Association Rules）。

关联规则挖掘的第一阶段必须从原始资料集合中找出所有的高频项目组。高频是指某一项目组出现的频率相对于所有记录而言，必须达到某一水平。一个项目组出现的频率称为支持度（Support），以一个包含 A 与 B 两个项目的 2-itemset 为例，我们可以由频繁出现的次数与总次数之间的比例求得包含$\{A,B\}$项目组的支持度，若支持度大于或等于所设定的最小支持度（Minimum Support）门槛值，则$\{A,B\}$称为高频项目组。一个满足最小支持度的 k-itemset 称为高频 k-项目组（Frequent k-itemset），一般表示为 Large k 或 Frequent k。算法从 Large k 项目组中再产生 Large $k+1$，直到无法找到更长的高频项目组为止。

关联规则挖掘的第二阶段是产生关联规则。从高频项目组产生关联规则，是利用前一步的高频 k-项目组来产生规则，在最小信赖度（Minimum Confidence）的条件门槛下，若一规则所求得的信赖度满足最小信赖度，则称此规则为关联规则。例如，经由高频 k-项目组$\{A,B\}$所产生的规则 AB，计算信赖度，若信赖度大于或等于最小信赖度，则称 AB 为关联规则。

3. 关联规则的分类

1）基于规则中处理的变量的类别

关联规则处理的变量可以分为布尔型和数值型。布尔型关联规则处理的值都是离散的、种类化的，它显示了这些变量之间的关系；而数值型关联规则可以和多维关联或多层关联规则结合起来，对数值型字段进行处理，将其进行动态地分割，或者直接对原始的数据进行处理，当然数值型关联规则中也可以包含种类变量。例如，性别＝“女”＝＞职业＝“秘书”，是布尔型关联规则；性别＝“女”＝＞avg(收入)＝2300，涉及的收入是数值类型，所以是一个数值型关联规则。

2）基于规则中数据的抽象层次

基于规则中数据的抽象层次可以分为单层关联规则和多层关联规则。在单层关联规则中，所有的变量都没有考虑到现实的数据是具有多个不同的层次的；而在多层关联规则中，对数据的多层性已经进行了充分的考虑。例如，IBM 台式机＝＞Sony 打印机，是一个细节数据上的单层关联规则；台式机＝＞Sony 打印机，是一个较高层次和细节层

次之间的多层关联规则。

3）基于规则中涉及的数据的维数

关联规则中的数据可以分为单维的和多维的。在单维的关联规则中，只涉及数据的一个维度，如用户购买的物品；而在多维的关联规则中，要处理的数据将会涉及多个维度。换句话说，单维的关联规则用于处理单个属性中的一些关系，多维的关联规则用于处理各个属性之间的某些关系。例如，啤酒＝＞尿布，这条规则只涉及用户购买的物品；性别＝“女”＝＞职业＝“秘书”，这条规则就涉及两个字段的信息，是两个维度上的一条关联规则。

4. 关联规则的相关算法

1）Apriori 算法

Apriori 算法使用候选项集找频繁项集。

Apriori 算法是一种最有影响的挖掘布尔关联规则频繁项集的算法。其核心是基于两阶段频集思想的递推算法。该关联规则在分类上属于单维、单层、布尔关联规则。在这里，所有支持度大于最小支持度的项集都称为频繁项集，简称频集。

该算法的基本思想是：首先找出所有的频集，这些项集出现的频繁性至少和预定义的最小支持度一样。然后由频集产生强关联规则，这些规则必须满足最小支持度和最小可信度。接着使用前面找到的频集产生期望的规则，产生只包含集合的项的所有规则，其中每一条规则的右部只有一项，这里采用的是中规则的定义。一旦这些规则被生成，那么只有那些大于用户给定的最小可信度的规则才被留下来。为了生成所有频集，使用了递推的方法。

Apriori 算法采用逐层搜索的迭代方法，算法简单明了，没有复杂的理论推导，也易于实现。但该算法有一些难以克服的缺点：

（1）对数据库的扫描次数过多。

（2）会产生大量的中间项集。

（3）采用唯一支持度。

（4）算法的适应面窄。

2）基于划分的算法

Savasere 等设计了一个基于划分的算法。这个算法先把数据库从逻辑上分成几个互不相交的块，每次单独考虑一个分块并对它生成所有的频集，然后把产生的频集合并，用来生成所有可能的频集，最后计算这些项集的支持度。这里分块的大小选择要使得每个分块可以被放入主存，每个阶段只需被扫描一次。而算法的正确性是由每一个可能的频集至少在某一个分块中是频集保证的。该算法是可以高度并行的，可以把每一个分块分别分配给某一个处理器生成频集。产生频集的每一个循环结束后，处理器之间进行通信来产生全局的候选 k-项集。通常这里的通信过程是算法执行时间的主要瓶颈；同时，

每个独立的处理器生成频集的时间也是一个瓶颈。

3）FP-树频集算法

针对 Apriori 算法的固有缺陷，J. Han 等提出了不产生候选挖掘频繁项集的方法——FP-树频集算法。采用分而治之的策略，在经过第一遍扫描之后，把数据库中的频集压缩进一棵频繁模式树（FP-Tree），同时依然保留其中的关联信息，随后将 FP-Tree 分化成一些条件库，每个库和一个长度为 1 的频集相关，然后对这些条件库分别进行挖掘。当原始数据量很大的时候，也可以结合划分的方法使得一个 FP-Tree 可以放入主存中。实验表明，FP-Tree 频集算法对不同长度的规则都有很好的适应性，同时在效率上较 Apriori 算法有巨大的提高。

5. 关联规则的应用

关联规则挖掘技术已经被广泛应用在西方金融行业企业中，它可以成功预测银行客户的需求。一旦获得了这些信息，银行就可以改善自身的营销。银行天天都在开发新的沟通客户的方法。各银行在自己的 ATM 机上就捆绑了顾客可能感兴趣的本行产品信息，供使用本行 ATM 机的用户了解。如果数据库中显示，某个高信用限额的客户更换了地址，这个客户很有可能新近购买了一栋更大的住宅，因此可能需要更高信用限额、更高端的新信用卡，或者需要住房改善贷款，这些产品都可以通过信用卡账单邮寄给客户。当客户打电话咨询的时候，数据库可以有力地帮助电话销售代表。电话销售代表的计算机屏幕上可以显示出客户的特点，同时也可以显示出客户会对什么产品感兴趣。

再例如，市场的数据不仅十分庞大、复杂，而且包含着许多有用的信息。随着数据挖掘技术的发展以及各种数据挖掘方法的应用，从大型超市数据库中可以发现一些潜在的、有用的、有价值的信息，从而应用于超级市场的经营。通过对所积累的销售数据的分析，可以得出各种商品的销售信息，从而更合理地制定各种商品的订货情况，对各种商品的库存进行合理的控制。另外，根据各种商品销售的相关情况，可分析商品的销售关联性，从而可以进行商品的货篮分析和组合管理，以更加有利于商品销售。

同时，一些知名的电子商务站点也从强大的关联规则挖掘中受益。这些电子购物网站使用关联规则进行挖掘，然后设置用户有意要一起购买的捆绑包。也有一些购物网站使用关联规则设置相应的交叉销售，也就是购买某种商品的顾客会看到相关的另一种商品的广告。

但是在我国，“数据海量，信息缺乏”是商业银行在数据大集中之后普遍面对的尴尬。金融业实施的大多数数据库只能实现数据的录入、查询、统计等较低层次的功能，却无法发现数据中存在的各种有用的信息，譬如对这些数据进行分析，发现其数据模式及特征，然后可能发现某个客户、消费群体或组织的金融和商业兴趣，并观察金融市场的变化趋势。可以说，关联规则挖掘技术在我国的研究与应用并不是很广泛深入。

3.3.5 K 最近邻算法

K 最近邻(K-Nearest Neighbor,KNN)算法是一个理论上比较成熟的算法,也是最简单的机器学习算法之一。K 最近邻算法在 1968 年由 Cover 和 Hart 提出,该算法使用的模型实际上对应对特征空间的划分。K 最近邻算法不仅可以用于分类,还可以用于回归。

该算法的思路是：如果一个样本在特征空间中的 K 个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别,则该样本也属于这个类别。在 K 最近邻算法中,所选择的邻居都是已经正确分类的对象。该算法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分类样本所属的类别。K 最近邻算法虽然从原理上依赖于极限定理,但在进行类别决策时,只与极少量的相邻样本有关。由于 K 最近邻算法主要靠周围有限的邻近的样本,而不是靠判别类域的方法来确定所属的类别,因此对于类域交叉或重叠较多的待分类样本集来说,K 最近邻算法比其他算法更为适合。

简单来说,K 最近邻算法可以看成：有一堆你已经知道分类的数据,当一个新数据进入的时候,就开始跟训练数据中的每个点求距离,然后挑离这个训练数据最近的 K 个点看这几个点属于什么类型,最后用少数服从多数的原则为新数据归类。

K 最近邻算法不仅可以用于分类,还可以用于回归。通过找出一个样本的 K 个最近邻居,将这些邻居的属性的平均值赋给该样本,就可以得到该样本的属性。更有用的方法是将不同距离的邻居对该样本产生的影响给予不同的权值(Weight),如权值与距离成正比(组合函数)。

该算法步骤如下。

步骤 1：初始化距离为最大值。

步骤 2：计算未知样本和每个训练样本的距离 dist。

步骤 3：得到目前 K 个最近邻样本中的最大距离 maxdist。

步骤 4：如果 dist 小于 maxdist,则将该训练样本作为 K 最近邻样本。

步骤 5：重复步骤 2、3、4,直到未知样本和所有训练样本的距离都算完。

步骤 6：统计 K 最近邻样本中每个类标号出现的次数。

步骤 7：选择出现频率最高的类标号作为未知样本的类标号。

基于 scikit-learn 包实现机器学习之 K 最近邻算法的函数及其参数含义：KNeighborsClassifier 是一个类,它集成了其他的 NeighborsBase、KNeighborsMixin、SupervisedIntegerMixin 和 ClassifierMixin。这里我们暂时不管它,主要看它的几个方法。当然有的方法是它从父类那里继承过来的。

(1) __init__()：初始化函数(构造函数)，它主要有以下几个参数。

- n_neighbors=5：int 型参数，K 最近邻算法中指定以最近的几个最近邻样本具有投票权，默认参数为 5。
- weights='uniform'：str 型参数，即每个拥有投票权的样本是按什么比重投票的，'uniform'表示等比重投票，'distance'表示按距离反比投票，[callable]表示自己定义一个函数，这个函数接收一个距离数组，返回一个权值数组。默认参数为'uniform'。
- algrithm='auto'：str 型参数，即内部采用什么算法实现。有以下 4 种选择参数：'ball_tree'(球树)、'kd_tree'(KD 树)、'brute'(暴力搜索)、'auto'(自动根据数据的类型和结构选择合适的算法)。默认情况下是'auto'。暴力搜索就不用讲了，读者应该都知道。前两种树状数据结构哪种好要视情况而定。KD 树是依次对 K 维坐标轴以中值切分构造的树，每一个节点是一个超矩形，在维数小于 20 时效率最高，可以参看参考文献[10]。球树是为了克服 KD 树高维失效而发明的，其构造过程是以质心 C 和半径 r 分割样本空间，每一个节点是一个超球体。一般低维数据用 KD 树速度快，用球树相对较慢。超过 20 维之后的高维数据用 KD 树效果反而不佳，而球树的效果更好，具体构造过程及优劣势的理论读者有兴趣可以自行学习。
- leaf_size=30：int 型参数，基于以上介绍的算法，此参数给出了 KD 树或者球树叶节点的规模，叶节点的不同规模会影响树的构造和搜索速度，同样会影响树的内存大小。具体最优规模是多少视情况而定。
- matric='minkowski'：str 或者距离度量对象，即怎样度量距离。默认是闵氏距离，闵氏距离不是一种具体的距离度量方法，它包括其他距离度量方式，是其他距离度量的推广，具体各种距离度量只是参数 p 的取值不同或者是否去极限的不同情况，读者可以关注参考文献[10]，讲得非常详细。

$$\mathrm{dist}(x,y)=\left(\sum_{i=1}^{n}|x_i-y_i|^p\right)^{\frac{1}{p}} \tag{3.8}$$

其中，$p=2$：int 型参数，就是以上闵氏距离各种不同的距离参数，默认为 2，即欧氏距离。$p=1$ 代表曼哈顿距离。

- metric_params=None：距离度量函数的额外关键字参数，无须设置，默认为 None。
- n_jobs=1：int 型参数，指并行计算的线程数量，默认为 1，表示一个线程，为-1 时表示 CPU 的内核数，也可以指定为其他数量的线程，若不追求速度的话，则无须处理，需要用到的话可以去看多线程。

(2) fit()：训练函数，它是最主要的函数。接收的参数只有 1 个，就是训练数据集，每一行是一个样本，每一列是一个属性。它返回对象本身，即只修改对象内部属性，因此

直接调用就可以了，后面用该对象的预测函数取预测自然就用到了这个训练的结果。其实该函数并不是 KNeighborsClassifier 这个类的方法，而是它的父类 SupervisedIntegerMixin 继承下来的方法。

(3) predict()：预测函数，接收输入的数组类型测试样本，一般是二维数组，每一行是一个样本，每一列是一个属性返回数组类型的预测结果，如果每个样本只有一个输出，则输出为一个一维数组。如果每个样本的输出是多维的，则输出二维数组，每一行是一个样本，每一列是一维输出。

(4) predict_prob()：基于概率的软判决，也是预测函数，只是并不是给出某一个样本的输出是哪一个值，而是给出该输出是各种可能值的概率各是多少。接收的参数和前面一样，返回的参数和前面类似，只是前面该是值的地方全部替换成概率，例如输出结果有两种选择 0 或 1，前面的预测函数给出的是长为 n 的一维数组，代表各样本一次的输出是 0 还是 1，如果用概率预测函数的话，返回的是 $n\times2$ 的二维数组，每一行代表一个样本，每一行有两个数，分别是该样本输出为 0 的概率为多少，输出为 1 的概率为多少。各种可能的顺序都是按字典顺序排列，例如先 0 后 1。

(5) score()：计算准确率的函数，接收的参数有 3 个。X：接收输入的数组类型测试样本，一般是二维数组，每一行是一个样本，每一列是一个属性。y：X 这些预测样本的真实标签，是一维数组或者二维数组。sample_weight＝None：是一个和 X 第一位一样长的样本对准确率影响的权重，默认为 None。输出为一个 float 型数，表示准确率。内部计算是按照 predict()函数计算的结果进行计算的。其实该函数并不是 KNeighborsClassifier 这个类的方法，而是它的父类 KNeighborsMixin 继承下来的方法。

(6) kneighbors()：计算某些测试样本最近的几个近邻的训练样本。接收 3 个参数。X＝None：需要寻找最近邻的目标样本。n_neighbors＝None：表示需要寻找目标样本最近的几个最近邻样本，默认为 None，需要调用时标识语句 return_distance＝True 是否需要同时返回具体的距离值。返回最近邻样本在训练样本中的序号。其实该函数并不是 KNeighborsClassifier 这个类的方法，而是它的父类 KNeighborsMixin 继承下来的方法。

1. *K* 最近邻算法的核心思想

当无法判定当前待分类点是从属于已知分类中的哪一类时，依据统计学的理论看它所处的位置特征，衡量它周围邻居的权重，把它归类到权重更大的那一类中。

K 最近邻算法的输入是测试数据和训练样本数据集，输出是测试样本的类别。

K 最近邻算法中，所选择的邻居都是已经正确分类的对象。*K* 最近邻算法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分类样本所属的类别。

2. *K* 最近邻算法的要素

K 最近邻算法有 3 个基本要素。

(1) *K* 值的选择：*K* 值的选择会对算法的结果产生重大影响。*K* 值较小意味着只有与输入实例较近的训练实例才会对预测结果起作用，但容易发生过拟合；如果 *K* 值较大，优点是可以减少学习的估计误差，但缺点是学习的近似误差增大，这时与输入实例较远的训练实例也会对预测起作用，使预测发生错误。在实际应用中，*K* 值一般选择一个较小的数值，通常采用交叉验证的方法来选择最优的 *K* 值。随着训练实例数目趋向于无穷和 $K=1$，误差率不会超过贝叶斯误差率的 2 倍，如果 *K* 值也趋向于无穷，则误差率趋向于贝叶斯误差率。

(2) 距离度量：距离度量一般采用 L_p 距离，当 $p=2$ 时，即为欧氏距离，在度量之前，应该将每个属性的值规范化，这样有助于防止具有较大初始值域的属性比具有较小初始值域的属性的权重大。

对于文本分类来说，使用余弦(Cosine)来计算相似度比欧氏距离更合适。

(3) 分类决策规则：该算法中的分类决策规则往往是多数表决，即由输入实例的 *K* 个最邻近的训练实例中的多数类决定输入实例的类别。

K 最近邻算法的基本要素如图 3.4 所示。

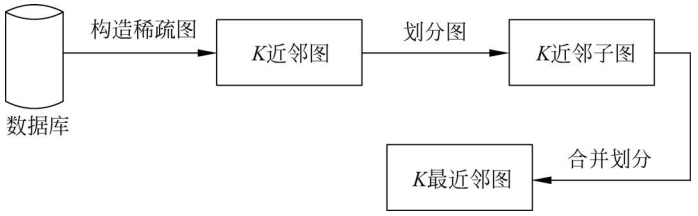


图 3.4　*K* 最近邻算法的基本要素

3. *K* 最近邻算法的流程

(1) 准备数据，对数据进行预处理。

(2) 选用合适的数据结构存储训练数据和测试元组。

(3) 设定参数，如 *K*。

(4) 维护一个距离由大到小的优先级队列(长度为 *K*)，用于存储最近邻训练元组。随机从训练元组中选取 *K* 个元组作为初始的最近邻元组，分别计算测试元组到这 *K* 个元组的距离，将训练元组标号和距离存入优先级队列。

(5) 遍历训练元组集，计算当前训练元组与测试元组的距离，将所得距离 *L* 与优先级队列中的最大距离 $L_{\max}$ 进行比较，若 $L \geqslant L_{\max}$，则舍弃该元组，遍历下一个元组。若 $L < L_{\max}$，则删除优先级队列中最大距离的元组，将当前训练元组存入优先级队列。

（6）遍历完毕，计算优先级队列中 K 个元组的多数类，并将其作为测试元组的类别。

（7）测试元组集测试完毕后计算误差率，继续设定不同的 K 值重新进行训练，最后取误差率最小的 K 值。

4. *K* 最近邻算法的优点

（1）K 最近邻算法从原理上依赖于极限定理，但在类别决策时只与极少量的相邻样本有关。

（2）由于 K 最近邻算法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别，因此对于类域的交叉或重叠较多的待分类样本集来说，K 最近邻算法比其他算法更为适合。

（3）算法本身简单有效，精度高，对异常值不敏感，易于实现，无须估计参数，分类器不需要使用训练集进行训练，训练时间复杂度为 0。

（4）K 最近邻分类的计算复杂度和训练集中的文档数目成正比，即如果训练集中的文档总数为 n，那么 K 最近邻算法的分类时间复杂度为 $O(n)$。

（5）适合对稀有事件进行分类。

（6）适用于多分类（Multi-Classification）问题，对象具有多个类别标签，K 最近邻算法比 SVM 算法的表现要好。

5. *K* 最近邻算法的缺点

（1）当样本不平衡时，样本数量并不能影响运行结果。

（2）算法计算量较大。

（3）可理解性差，无法给出像决策树那样的规则。

6. *K* 最近邻算法的改进策略

K 最近邻算法因其提出时间较早，随着其他技术的不断更新和完善，K 最近邻算法逐渐显示出诸多不足之处，因此许多 K 最近邻算法的改进算法应运而生。算法改进目标主要朝着分类效率和分类效果两个方向。

改进 1：通过找出一个样本的 K 个最近邻居，将这些邻居的属性的平均值赋给该样本，就可以得到该样本的属性。

改进 2：将不同距离的邻居对该样本产生的影响给予不同的权值（weight），如权值与距离成反比（$1/d$），即和该样本距离小的邻居权值大，称为可调整权重的 K 最近邻居法（Weighted Adjusted K Nearest Neighbor，WAKNN）。但 WAKNN 会造成计算量增大，因为对每一个待分类的文本都要计算它到全体已知样本的距离，才能求得它的 K 个最近邻点。

改进 3：事先对已知样本点进行剪辑（Editing 技术），事先去除（Condensing 技术）对分类作用不大的样本。该算法适用于样本容量比较大的类域的自动分类，而那些样本容量较小的类域采用这种算法比较容易产生误分。

3.3.6　支持向量机算法

支持向量机(Support Vector Machine)是一种分类算法,通过寻求结构化风险最小来提高学习机的泛化能力,实现经验风险和置信范围的最小化,从而达到在统计样本量较少的情况下,也能获得良好统计规律的目的。通俗来讲,它是一种二类分类模型,其基本模型定义为特征空间上间隔最大的线性分类器,即支持向量机的学习策略是间隔最大化,最终可转换为一个凸二次规划问题的求解。

支持向量机是由模式识别中的广义肖像算法(Generalized Portrait Algorithm)发展而来的分类器,其早期工作来自苏联学者 Vladimir N. Vapnik 和 Alexander Y. Lerner 在 1963 年发表的研究。1964 年,Vladimir N. Vapnik 和 Alexey Y. Chervonenkis 对广义肖像算法进行了进一步讨论并建立了硬边距的线性支持向量机。此后在 20 世纪 70—80 年代,随着模式识别中最大边距决策边界的理论研究、基于松弛变量(Slack Variable)的规划问题求解技术的出现,和 VC 维(Vapnik-Chervonenkis Dimension)的提出,支持向量机被逐步理论化并成为统计学习理论的一部分。1992 年,Bernhard E. Boser、Isabelle M. Guyon 和 Vapnik 通过该方法得到了非线性支持向量机。1995 年,Corinna Cortes 和 Vapnik 提出了软边距的非线性支持向量机并将其应用于手写字符识别问题,这份研究在发表后得到了关注和引用,为支持向量机在各领域的应用提供了参考。

1. 具体原理

在 n 维空间中找到一个分类超平面,将空间上的点分类。图 3.5 是线性分类的例子。

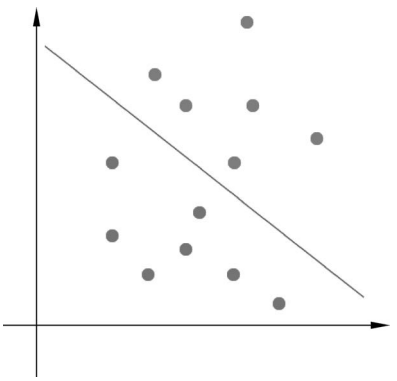


图 3.5　线性分类示例图

一般而言,一个点距离超平面的远近可以表示为分类预测的确信或准确程度。支持向量机就是要最大化这个间隔值。而在虚线上的点叫作支持向量(Support Vector),图 3.6 表示正常人与病人之间的间隔分类,图 3.7 表示支持向量对数据的分类。

在实际应用中,我们经常会遇到线性不可分的样例,此时常用的做法是把样例特征映射到高维空间中去,如图 3.8 所示。

线性不可分映射到高维空间可能会导致维度高到可怕(19 维乃至无穷维),导致计算复杂。核函数的价值在于它虽然也是对特征进行从低维到高维的转换,但核函数绝就绝在它事先在低维进行计算,而将实质上的分类效果表现在高维上,也就如前文所讲的避免了直接在高维空间的复杂计算。

使用松弛变量处理数据噪声如图 3.9 所示。

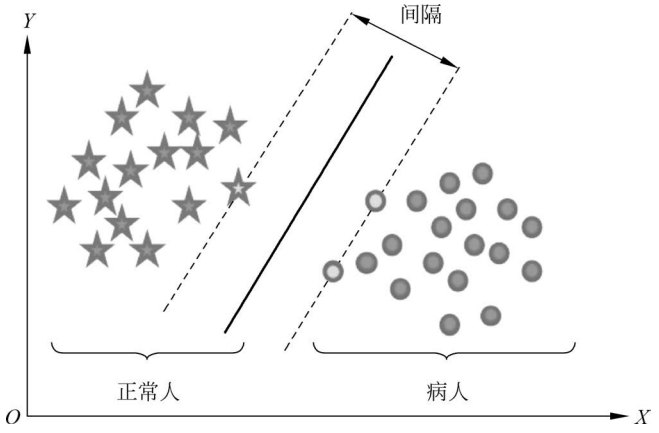


图 3.6　正常人和病人之间的间隔分类

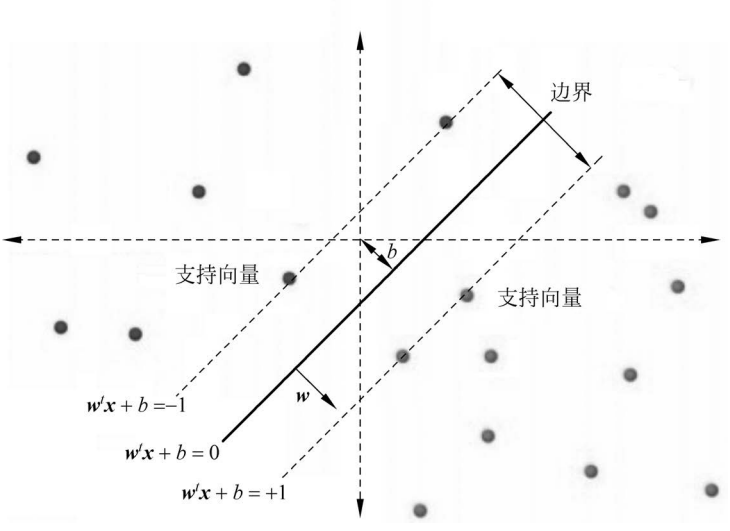


图 3.7　利用支持向量对数据进行分类

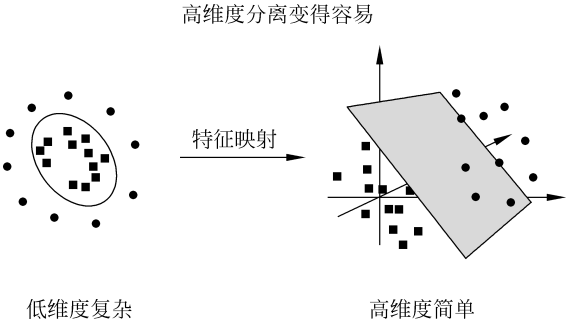


图 3.8　高维度空间特征分类

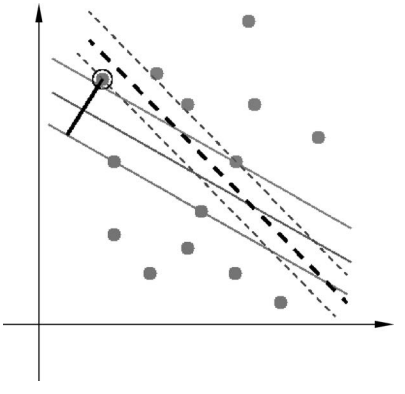


图 3.9　使用松弛变量处理数据噪声

2. 支持向量机的性质

稳健性与稀疏性：支持向量机的优化问题同时考虑了经验风险最小化和结构风险最小化，因此具有稳定性。从几何观点来看，支持向量机的稳定性体现在其构建超平面决策边界时要求边距最大，因此间隔边界之间有充裕的空间包容测试样本。支持向量机使用铰链损失函数作为代理损失，铰链损失函数的取值特点使支持向量机具有稀疏性，即其决策边界仅由支持向量决定，其余的样本点不参与经验风险最小化。在使用核方法的非线性学习中，支持向量机的稳健性和稀疏性在确保了可靠求解结果的同时降低了核矩阵的计算量和内存开销。

与其他线性分类器的关系：支持向量机是一个广义线性分类器，通过在支持向量机算法框架下修改损失函数和优化问题可以得到其他类型的线性分类器，例如将支持向量机的损失函数替换为 Logistic 损失函数就得到了接近 Logistic 回归的优化问题。支持向量机和 Logistic 回归是功能相近的分类器，二者的区别在于 Logistic 回归的输出具有概率意义，也容易扩展至多分类问题，而支持向量机的稀疏性和稳定性使它具有良好的泛化能力并在使用核方法时计算量更小。

作为核方法的性质：支持向量机不是唯一可以使用核技巧的机器学习算法，Logistic 回归、岭回归和线性判别分析（Linear Discriminant Analysis，LDA）也可通过核方法得到核 Logistic 回归（Kernel Logistic Regression）、核岭回归（Kernel Ridge Regression）和核线性判别分析（Kernelized Linear Discriminant Analysis，KLDA）方法。因此，支持向量机是广义上核学习的实现之一。

3. 支持向量机的应用

支持向量机在各领域的模式识别问题中都有应用，包括人像识别、文本分类、手写字符识别、生物信息学等。

4. 包含支持向量机的编程模块

按引用次数，由台湾大学资讯工程研究所开发的 LIBSVM 是使用最广的支持向量机工具。LIBSVM 包含标准支持向量机算法、概率输出、支持向量回归、多分类支持向量机等功能，其源代码由 C 编写，并有 Java、Python、R、MATLAB 等语言的调用接口，基于 CUDA 的 GPU 加速和其他功能性组件，例如多核并行计算、模型交叉验证等。

基于 Python 开发的机器学习模块 scikit-learn 提供预封装的支持向量机工具，其设计参考了 LIBSVM。其他包含支持向量机的 Python 模块有 MDP、MLPy、PyMVPA 等。TensorFlow 的高阶 API 组件 Estimators 提供了支持向量机的封装模型。

3.3.7　频繁项集挖掘算法

频繁项集挖掘算法用于挖掘经常一起出现的 item 集合，这些集合称为频繁项集，通

过挖掘出这些频繁项集，当在一个事务中出现频繁项集的其中一个 item 时，就可以把该频繁项集的其他 item 作为推荐。例如经典的购物篮分析中的啤酒和尿布的故事，啤酒和尿布经常在用户的购物篮中一起出现，通过挖掘出啤酒、尿布这个啤酒项集，当一个用户买了啤酒的时候可以为他推荐尿布，这样用户购买的可能性会比较大，从而达到组合营销的目的。

常见的频繁项集挖掘算法有两类：一类是 Apriori 算法，另一类是 FPGrowth 算法。Apriori 算法通过不断地构造候选集、筛选候选集挖掘出频繁项集，需要多次扫描原始数据，当原始数据较大时，磁盘 I/O 次数太多，效率比较低下。FPGrowth 算法则只需扫描原始数据两遍，通过 FP-Tree 数据结构对原始数据进行压缩，效率较高。

频繁项集挖掘算法最著名的一个实现是 Agrawal 和 R. Srikant 于 1994 年提出的 Apriori 算法。该算法与其他频繁项集挖掘算法相比，一个重大的优化是提出了先验性质。先验性质通俗地说，即若一个集合是非频繁的，则它的任何超集都是非频繁的，反过来，若一个集合是频繁的，则它的所有真子集都是频繁的。之所以说先验性质是 Apriori 算法相对于其他频繁项集挖掘算法的重大优化，是因为先验性质能够有效地减少算法的搜索空间，这一点我们将会在具体实现中体现出来。

1. Apriori 算法

1）Apriori 概述

该算法的实现原理其实非常简单，即使用一种逐层搜索的迭代算法，使用 k 项集来迭代产生 $k+1$ 项候选集(Candidate Set)，然后通过先验性质对 $k+1$ 项候选集进行剪枝操作，缩小算法搜索空间，进而由此产生 $k+1$ 项集。由 k 项集产生 $k+1$ 项候选集的过程可分为两步，第一步为连接步，笔者不想用过多的数学表达，仅在这里简单地描述连接步过程。假设我们已经获得了 k 项频繁项集合，则从该项集的第一项(记为第 i 项)开始向下寻找前 $k-1$ 项与其相同的项(记为第 j 项)，则第 i 项的全部元素和第 j 项的第 k 个元素组成了 $k+1$ 项“候选集”的一项，以此类推，直到遍历 k 项频繁项集。之所以“候选集”要打引号，是因为这里的集合还没有经过剪枝操作，还不能真正成为候选集。

集合的剪枝操作即在 $k+1$ 项集中，针对每个 $k+1$ 项集检验它的每个 k 项真子集是否都为事务集合的频繁项集，若是，则保留这个 $k+1$ 项集作为候选集，否则从集合中删除这个 $k+1$ 项集。这样我们就获得了 $k+1$ 项候选集，在这之后针对数据库进行扫描，计算出每个 $k+1$ 项集的频繁计数(出现了多少次)，大于最小频繁计数(min_sup)的保留，小于最小频繁计数的从集合中去除，这样就得到了 $k+1$ 项频繁项集合。然后重复以上工作，迭代搜索，直到找出一个 n 项频繁集合，无法再找出 $n+1$ 项频繁项集，则算法结束，释放资源，停机。Apriori 算法如图 3.10 所示。

2）Apriori 算法的实现

为了降低程序的 I/O 次数并提高运行速度，在程序启动时就读取数据进入内存，使

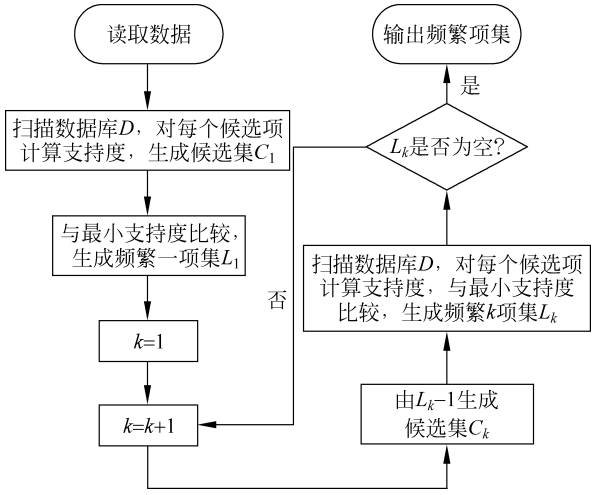


图 3.10　Apriori 算法示例图

用树状结构来存储数据。该数据结构中使用到两类节点，一类节点用于存储事务元素，另一类节点用于存储事务 ID，并为元素节点提供索引，整个数据结构使用简单的单链表实现。数据结构直观上很像一个散列表，如图 3.11 所示。

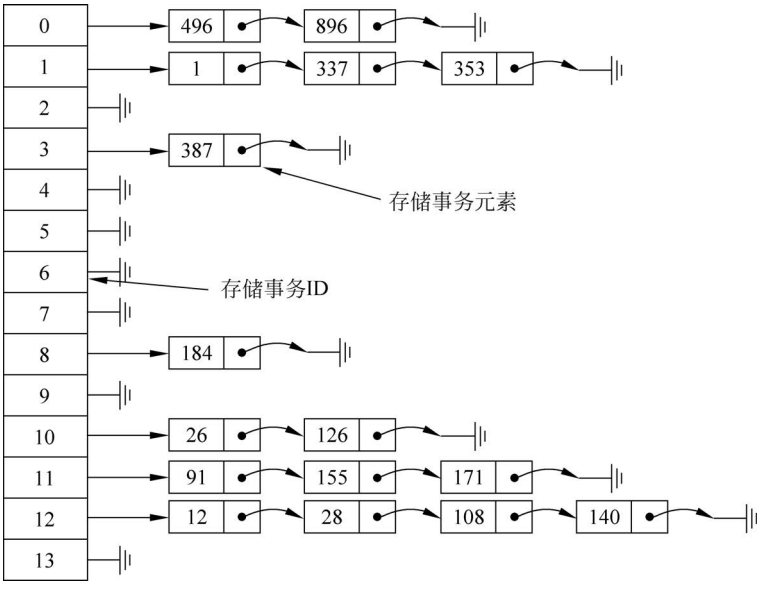


图 3.11　Apriori 算法的数据结构

如图 3.11 所示的 Apriori 算法返回一个映射，该映射中存储着所有的频繁一项集及其频繁计数，该映射作为下一级函数的输入，用于产生频繁二项集，频繁三项集……

从频繁二项集开始，算法就开始使用递归函数生成，具体算法过程如下：首先接受 k 项频繁项集，然后使用两个指针一前一后产生所有符合“连接步条件”的 $k+1$ 项集合。

这个 $k+1$ 项集合通过先验条件剪枝后得到 $k+1$ 项候选集，然后计算集合中每一个 $k+1$ 项集在事务集合中的计数，不符合 min_sup 的去掉，然后递归调用该函数，直到找不出更多的集合，则递归返回。

Apriori 算法需多次扫描数据库，每次利用候选频繁集产生频繁集；而 FP-Growth 则利用树状结构，无须产生候选频繁集而是直接得到频繁集，大大减少了扫描数据库的次数，从而提高了算法的效率。

FP-Growth 算法采取如下分治策略：首先，将代表频繁项集的数据库压缩到一棵频繁模式树（FP 树），该树仍保留项集的关联信息。然后，把这种压缩后的数据库划分成一组条件数据库，每个数据库关联一个频繁项或模式段，并分别挖掘每个条件数据库。对于每个“模式段”，只需要考察它相关联的数据集。因此，随着被考察的模式的“增长”，这种方法可以显著地压缩被搜索的数据集的大小。

FP-Growth 算法的基本思路如下：

（1）扫描一次事务数据库，找出频繁 1-项集合，记为 L，并把它们按支持度计数的降序进行排列。

（2）基于 L，再扫描一次事务数据库，构造表示事务数据库中项集关联的 FP 树。

（3）在 FP 树上递归地找出所有频繁项集。

（4）最后在所有频繁项集中产生强关联规则。

FP-Growth 算法主要分为两个步骤：FP 树构建和递归挖掘 FP 树。FP 树构建通过两次数据扫描，将原始数据中的事务压缩到一个 FP 树中，该 FP 树类似于前缀树，相同前缀的路径可以共用，从而达到压缩数据的目的。接着通过 FP 树找出每个 item 的条件模式基、条件 FP 树，递归地挖掘条件 FP 树得到所有的频繁项集。算法的主要计算瓶颈在 FP 树的递归挖掘上。

FP-Growth 算法是当前频繁项集挖掘算法中速度最快、应用最广，并且不需要候选项集的一种频繁项集挖掘算法，但是 FP-Growth 算法也存在着结构复杂和空间利用率低等缺点。Relim 算法是在 FP-Growth 算法的基础上提出的一种新的不需要候选项集的频繁项集挖掘算法。它具有算法结构简单，空间利用率高，易于实现等显著优点。

2. Relim 算法

1）Relim 算法的主要思想

Relim 算法的主要思想和 FP-Growth 算法相似，也是基于递归搜索（Recursive Exploration），但是和 FP-Growth 算法不同的是：Relim 算法在运行时不必创建频繁模式树，而是通过建立一个事务链表组（Transaction Lists）来找出所有频繁项集。

2）Relim 算法的方法描述

为了更好地描述该算法，我们通过一个实例来说明 Relim 算法的挖掘过程。该例基于如图 3.12(a)所示的事务数据集，数据集中有 10 个事务，设最小支持度为 3（min sup=

3/10＝30％)。

Relim 算法的挖掘过程如下：

(1) 与 Apriori 算法相同，首先对图 3.12(a)的数据集进行第一次扫描，找出候选 1 项集的集合，并得到它们的支持度计数(频繁性)。然后按照支持度计数递增排列各项集。

(2) 将支持度小于最小支持度 3 的元素(图 3.12(b)中的元素 g 和 f)从各事务中消除，然后根据元素的支持度计数递增地将图 3.12(a)中的事务重新进行排列，如图 3.12(c)所示。注意：事务元素的排列顺序不会影响挖掘结果，但对运行速度有影响。递增排列的运行速度最快，反之则最慢。

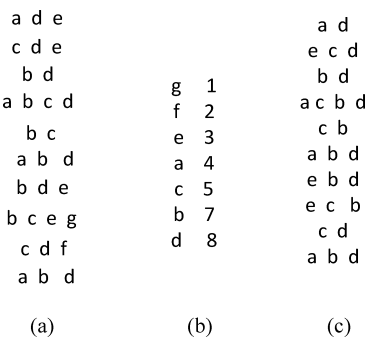


图 3.12　事务数据集

(3) 为事务数据集中的所有元素都创建一项单向数据链表，并且使每个元素的数据链表都包含一个计数器和一个指针。计数器的值表示在图 3.12(c)中以此元素为头元素的事务总数，称之为头元素数值。指针则用于保存图 3.12(c)中相关事务的关联信息，因此元素的数据链表也被称为事务链表(Transaction List)。把所有事务链表按照图 3.12(b)中元素支持度的计数递增排列，由此就创建了一组事务链表，在本书中称为事务链表组。

(4) 按照元素支持度计数由小到大的顺序，首先扫描以 e 元素为头元素的事务链表(简称为 e 事务链表)，如发现该链表中有项集的支持度大于或等于最小支持度 3，就将此项集输出。在 e 事务链表中，只有项集{e}的支持度等于 3，所以将{e}输出。扫描完成后，将 e 事务链表的头元素数值设为零，并将 e 事务链表从链表组中删除。

(5) 创建一个和第(3)步描述的结构相似的单向数据链表组，链表组保存着将 e 事务链表的头元素除掉后，其后继元素作为头元素的事务关联信息。这个数据链表组称为前缀 e 事务链表组。

(6) 将前缀 e 事务链表组和 e 事务链表为空的事务链表组进行合并，得到一个新的事务链表组。

(7) 根据第(4)～(6)步所述，递归地对新事务链表组中的第二个事务链表(a 事务链表)进行挖掘。其挖掘结果是：{a}，{a，b}，{a，b，d}，{a，d}的支持度都大于最小支持度 3，将其结果输出。

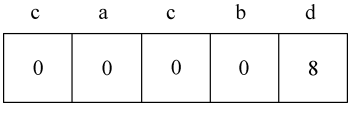


图 3.13　Relim 算法的数据挖掘结果

(8) 当挖到最后一个事务链表的时候，事务链表组如图 3.13 所示。该事务链表的计数器的值为 8，而链表指针指向空。输出频繁项集{d}后，结束频繁项集的挖掘。

3.4 数据挖掘的其他方法

3.4.1 多层次数据汇总归纳

数据库中的数据和对象经常包含原始概念层上的详细信息，将一个数据集合归纳成高概念层次信息的数据挖掘技术被称为数据汇总(Data Generalization)。概念汇总将数据库中的相关数据由低概念层抽象到高概念层，主要有数据立方体和面向属性两种方法。

(1) 数据立方体(多维数据库)方法的主要思想是将那些经常查询、代价高昂的运算，如 Count、Sum、Average、Max、Min 等汇总函数具体化，并存储在一个多维数据库中，为决策支持、知识发现及其他应用服务。

(2) 面向属性的抽取方法用一种类 SQL 数据挖掘查询语言表达查询要求，收集相关数据，并利用属性删除、概念层次树、门槛控制、数量传播及集合函数等技术进行数据汇总。汇总数据用汇总关系表示，可以将数据转换为不同类型的知识，或将其映射成不同的表，并从中抽取特征、判别式、分类等相关规则。

面向属性抽取的概念层次树是指某属性所具有的从具体概念值到某概念类的层次关系树。概念层次可由相关领域专家根据属性的领域知识提供，按特定属性的概念层次从一般到具体排序。树的根节点是用 ANY 表示最一般的概念，叶节点是最具体的概念，即属性的具体值，概念层次为归纳分析提供有用的信息，将概念组织为不同的层次，从而在高概念层次上用简单、确切的公式表示规则。

CaiCencone 利用属性值的概念层次关系提出了面向属性的树提升算法，并得到了一阶谓词逻辑表示的规则。面向属性的树提升方法主要是对目标类所有元组的属性值由低到高提升，使原来若干属性值不同的元组成为相同的元组，并进行合并，直到全部元组不超过最大规则数，再将其转换为一阶谓词逻辑表示的规则。

与面向元组的归纳方法相比，面向属性的归纳方法搜索空间减少，运行效率显著提高；对冗余元组的测试在概括属性的所有值后进行，提高了测试效率；最坏时间复杂度为 $O(N\log P)$，N 为元组个数，P 为最终概括关系表中的元组个数。处理过程可利用关系数据库的传统操作。此方法已在数据挖掘系统 DBMINE 中采用，除关系数据库外，也可扩展到面向对象数据库。

多维分析是对具有多个维度和指标所组成的数据模型进行的可视化分析手段的统称。常用的分析方式包括下钻、上卷、切片(切块)、旋转等各种分析操作，以便剖析数据，使分析者、决策者能从多个角度、多个侧面观察数据，从而深入了解包含在数据

中的信息和内涵。上卷是在数据立方体中执行聚集操作，通过在维级别中上升或通过消除某个或某些维来观察更概括的数据。上卷的另一种情况是通过消除一个或者多个维来观察更加概括的数据。下钻是在维级别中下降或者通过引入某个或者某些维来更细致地观察数据。切片是在给定的数据立方体的一个维上进行的选择操作。切片的结果是得到一个二维的平面数据(切块是在给定的数据立方体的两个或者多个维上进行选择操作。切块的结果是得到一个子立方块)。旋转相对比较简单，就是改变维的方向。

3.4.2　决策树方法

在博弈论中，常常使用决策树寻找最优决策，这些决策树往往是人工生成的。在数据挖掘过程中，决策树的生成通常是通过对数据的拟合、学习，从数据集中获取一棵决策树。

决策树的形式，从根节点到叶节点的路径就是决策的过程。其本质思路就是使用超平面对数据递归化划分。决策树的生成过程就是对数据集进行反复切割的过程，直到能够把决策类别区分开来为止，切割的过程就形成了一棵决策树。

而实例隶属于决策树每个终端节点的个数，就可以看作该节点的支持度，支持度越大，该规则越有力。

1. 决策树的构造过程

决策树解决的问题：

(1) 选择哪一个属性进行第一次划分？

(2) 什么时候停止继续划分？

对于第一个问题，需要定义由属性进行划分的度量，根据该度量可以计算出对当前数据子集来说最佳的划分属性。对于第二个问题，通常有两种方法：一种是定义一棵停止树进一步生长的条件，另一种是对生成完成的树进行再剪枝。

2. 选择属性进行划分

信息增益方法基于信息熵原理(一种对信息混乱程度的度量)。一般来说，如果信息是均匀混合分布的，信息熵就高。如果信息呈现一致性分布，信息熵就低。在决策树分类中，如果数据子集中的类别混合均匀分布，信息熵就高。如果类别单一分布，信息熵就低。

显然，选择信息熵向最小的方向变化的属性，就能使得决策树迅速达到叶节点，从而能够构造一棵决策树。对于每个数据集/数据子集，信息熵可如下定义：

$$\mathrm{ENT}(D_j)=-\sum_{i}=0c-1p_i\log 2P_i\,\mathrm{ENT}(D_j)=-\sum_{i}=0c-1p_i\log 2P_i \quad (3.9)$$

c 是数据集/子集 D_j 中决策类的个数，p_i 是第 i 个决策类在 D 中的比例。

对于任意一个属性，若将数据集划分为多个数据子集，则该属性的信息增益为未进行划分时的数据集的信息熵与划分后的数据子集的信息熵加权和的差：

$$\mathrm{GAIN}(A)=\mathrm{ENT}(D)-\sum_{j}=0k\mid D_j\mid\mathrm{DENT}(D_j)\mathrm{GAIN}(A)$$

$$=\mathrm{ENT}(D)-\sum_{j}=0k\mid D_j\mid\mathrm{DENT}(D_j)$$

A 是候选属性，k 是该属性的分支数，D 是未使用 A 进行划分时的数据集，D_j 是由 A 划分而成的子数据集，$|x|$ 代表数据集的实例个数。

Gini 系数则是从另一个方面刻画信息的纯度。该方法用于计算从相同的总体中随机选择的两个样本来自不同类别的概率：

$$\mathrm{Gini}(D_j)=1-\sum_{i}=0c-1p_i^2\mathrm{Gini}(D_j)=1-\sum_{i}=0c-1p_i^2 \tag{3.10}$$

c 是数据集/子集 D_j 中决策类的个数，p_i 是第 i 个决策类在 D 中的比例。

于是，对于任意一个属性，若将数据集划分为多个子集，则未进行划分时的数据集的 Gini 系数与划分后的数据子集的 Gini 系数加权和的差为：

$$\mathrm{Gini}(D)-\sum_{j}=1kD_jD\mathrm{Gini}(D_j)\mathrm{Gini}(D)-\sum_{j}=1kD_jD\mathrm{Gini}(D_j) \tag{3.11}$$

在所有属性中，具有最大 $G(A)$ 的属性被选为当前进行划分的节点。

由此可以看出，两者的计算结果是类似的，但是，信息增益方法和 Gini 系数法都有一个问题，即它们都偏向于具有很多不同值的属性，因为多个分支能降低信息熵或 Gini 系数。但决策树划分为很多分支会降低决策树的适用性。

CART 算法只生成二叉树。为生成二叉树，CART 算法使用 Gini 系数测试属性值两两组合，找出最好的二分方法。

C4.5 算法采用信息增益的改进形式增益率来解决问题。增益率在信息增益中引入了该节点的分支信息，对分支过多的情况进行惩罚：

$$\mathrm{GAIN}(A)-\sum k_i=1\mid D_j\mid\mid D\mid\log2\mid D_j\mid\mathrm{DGAIN}(A)-\sum_{i}$$

$$=1k\mid D_j\mid\mid D\mid\log2\mid D_j\mid D \tag{3.12}$$

k 为属性 A 的分支数。由此可知，k 越大，增益率越小。

CHAID 方法利用卡方检验来寻找最优的划分属性。对于连续的属性，使用卡方检验方法将其离散化。对于离散属性，使用卡方检验方法查找可以合并的值。

3. 获得大小适合的树

决策树的目的事实上是通过训练集获得一棵简洁的、预测能力强的树。但往往树完全生长会导致预测能力降低，因此最好是恰如其分。定义树停止生长的条件如下。

(1) 最小划分实例数：当当前节点对应的数据子集的大小小于指定的最小划分实例数时，即使它们不属于同一类，也不再进行进一步划分。

（2）划分阈值：当使用的划分方法所得的值与其父节点的值的差小于指定的阈值时，不再进行划分。

（3）最大树深度：当进一步划分超过最大树的深度的时候，停止划分。另一种方法是在生成完全决策树之后进行剪枝任务。

在CART算法中，对每棵子树构造一个成本复杂性指数（参考误分类错误率和树的结构复杂度），从一系列结构中选择该指数最小的树作为最好的树。其中，误分率需要用一个单独的剪枝集来评估。树的复杂度也通过树的终端节点个数与每个终端节点的成本的积来刻画。

下面以银行贷款决策树为例，利用树结构实现上述的判断流程，如图3.14所示。

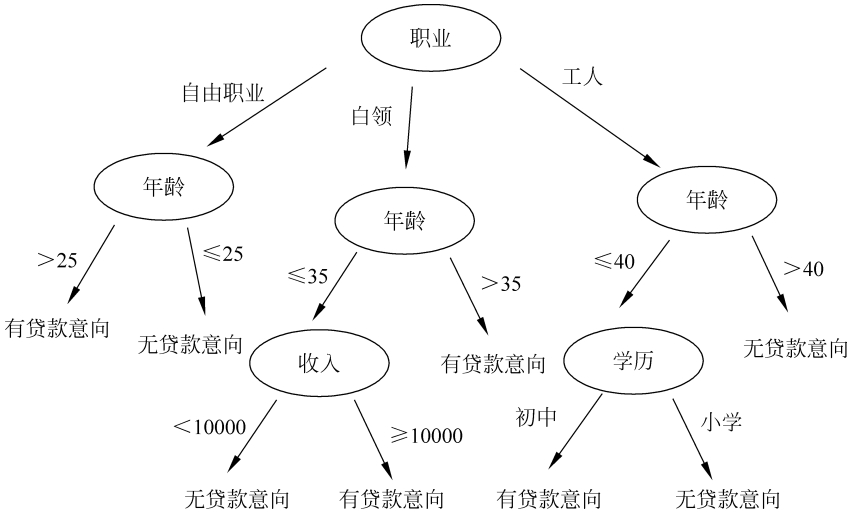
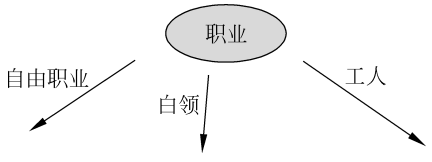


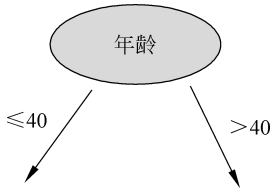
图3.14　银行贷款意向分析决策树示意图

通过训练数据，我们可以建议如图3.14所示的决策树，其输入是用户的信息，输出是用户的贷款意向。如果要判断某一客户是否有贷款的意向，直接根据用户的职业、收入、年龄以及学历就可以分析得出用户的类型。例如某客户的信息为：{职业，年龄，收入，学历}={工人，39，1800，小学}，将信息输入上述决策树，可以得到下列的分析步骤和结论。

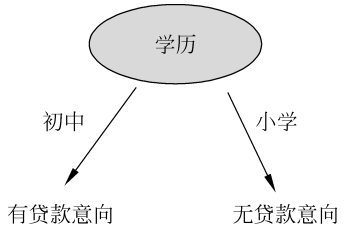
第一步：根据该客户的职业进行判断，选择“工人”分支。



第二步：根据客户的年龄进行选择，选择年龄“≤40”这一分支。



第三步：根据客户的学历进行选择，选择“小学”这一分支，得出该客户无贷款意向的结论。



3.4.3 神经网络方法

思维学普遍认为，人类大脑的思维分为抽象（逻辑）思维、形象（直观）思维和灵感（顿悟）思维 3 种基本方式。

人工神经网络就是模拟人思维的第二种方式。这是一个非线性动力学系统，其特色在于信息的分布式存储和并行协同处理。虽然单个神经元的结构极其简单，功能有限，但大量神经元构成的网络系统所能实现的行为却是极其丰富多彩的。

神经网络的研究内容相当广泛，反映了多学科交叉技术领域的特点。主要的研究工作集中在以下几个方面：

（1）生物原型研究。从生理学、心理学、解剖学、脑科学、病理学等生物科学方面研究神经细胞、神经网络、神经系统的生物原型结构及其功能机理。

（2）建立理论模型。根据生物原型的研究，建立神经元、神经网络的理论模型，其中包括概念模型、知识模型、物理化学模型、数学模型等。

（3）网络模型与算法研究。在理论模型研究的基础上构造具体的神经网络模型，以实现计算机模拟或准备制作硬件，包括网络学习算法的研究。这方面的工作也称为技术模型研究。

（4）人工神经网络应用系统。在网络模型与算法研究的基础上，利用人工神经网络组成实际的应用系统，例如完成某种信号处理或模式识别的功能、构造专家系统、制成机器人等。

纵观当代新兴科学技术的发展历史，人类在征服宇宙空间、基本粒子、生命起源等科学技术领域的进程中历经了崎岖不平的道路。我们也会看到，探索人脑功能和神经网络

的研究将伴随着重重困难的克服而日新月异。

1. 神经网络的工作原理

人工神经元的研究起源于脑神经元学说。19 世纪末，在生物、生理学领域，Waldeger 等创建了神经元学说。人们认识到复杂的神经系统是由数目繁多的神经元组合而成的。大脑皮层有 100 亿个以上的神经元，每立方毫米约有数万个，它们互相连接形成神经网络，通过感觉器官和神经接受来自身体内外的各种信息，传递至中枢神经系统内，经过对信息的分析和综合，再通过运动神经发出控制信息，以此来实现机体与内外环境的联系，协调全身的各种机能活动。

神经元也和其他类型的细胞一样，包括细胞膜、细胞质和细胞核。但是神经细胞的形态比较特殊，具有许多突起，因此又分为细胞体、轴突和树突 3 部分。细胞体内有细胞核，突起的作用是传递信息。树突是作为引入输入信号的突起，而轴突是作为输出端的突起，它只有一个。

树突是细胞体的延伸部分，它由细胞体发出后逐渐变细，全长各部位都可与其他神经元的轴突末梢相互联系，形成所谓的“突触”。在突触处两神经元并未连通，它只是发生信息传递功能的结合部，联系界面之间的间隙约为(15～50)×10 米。突触可分为兴奋性与抑制性两种类型，它相当于神经元之间耦合的极性。每个神经元的突触数目最高可达 10 个。各神经元之间的连接强度和极性有所不同，并且都可调整。基于这一特性，人脑具有存储信息的功能。利用大量神经元相互连接组成人工神经网络可显示出人的大脑的某些特征。

人工神经网络是由大量的简单基本元件——神经元相互连接而成的自适应非线性动态系统。每个神经元的结构和功能比较简单，但大量神经元组合产生的系统行为却非常复杂。

人工神经网络反映了人脑功能的若干基本特性，但并非生物系统的逼真描述，只是某种模仿、简化和抽象。

与数字计算机比较，人工神经网络在构成原理和功能特点等方面更加接近人脑，它不是按给定的程序一步一步地执行运算，而是能够自身适应环境、总结规律，完成某种运算、识别或过程控制。

人工神经网络首先要以一定的学习准则进行学习，然后才能工作。现以人工神经网络对于 A、B 两个字母的识别为例进行说明，规定当 A 输入网络时，应该输出 1，而当输入为 B 时，输出为 0。

所以网络学习的准则应该是：如果网络做出错误的判决，则通过网络的学习，应使得网络减少下次犯同样错误的可能性。首先，给网络的各连接权值赋予(0，1)区间内的随机值，将 A 所对应的图像模式输入网络，网络将输入模式加权求和，与门限比较，再进行非线性运算，得到网络的输出。在此情况下，网络输出为 1 和 0 的概率各为 50%，也就是

说是完全随机的。

这时如果输出为1(结果正确)，则使连接权值增大，以便使网络再次遇到A模式输入时，仍然能做出正确的判断。如果输出为0(结果错误)，则把网络连接权值朝着减小综合输入加权值的方向调整，其目的在于使网络下次再遇到A模式输入时，减少犯同样错误的可能性。如此操作调整，当给网络轮番输入若干手写字母A、B后，经过网络按以上学习方法进行若干次学习后，网络判断的正确率将大大提高。这说明网络对这两个模式的学习已经获得了成功，它已将这两个模式分布地记忆在网络的各个连接权值上。当网络再次遇到其中任何一个模式时，能够做出迅速、准确的判断和识别。一般来说，网络中所含的神经元个数越多，它能记忆、识别的模式也就越多。

2. 神经网络的特点

(1) 人类大脑有很强的自适应与自组织特性，后天的学习与训练可以开发许多各具特色的活动功能。例如盲人的听觉和触觉非常灵敏，聋哑人善于运用手势，训练有素的运动员可以表现出非凡的运动技巧等。

普通计算机的功能取决于程序中给出的知识和能力。显然，对于智能活动，要通过总结编制程序将十分困难。

人工神经网络具有初步的自适应与自组织能力，在学习或训练过程中改变突触权重值，以适应周围环境的要求。同一网络因学习方式及内容不同可具有不同的功能。人工神经网络是一个具有学习能力的系统，可以发展知识，以致超过设计者原有的知识水平。通常，它的学习训练方式可分为两种：一种是有监督学习，或称有导师的学习，这时利用给定的样本标准进行分类或模仿；另一种是无监督学习，或称无导师的学习，这时只规定学习方式或某些规则，具体的学习内容随系统所处的环境(输入信号情况)而异，系统可以自动发现环境特征和规律性，具有更近似人脑的功能。

(2) 泛化能力。泛化能力指对没有训练过的样本，有很好的预测能力和控制能力。特别是，当存在一些有噪声的样本时，网络具备很好的预测能力。

(3) 非线性映射能力。当系统对于设计人员来说很透彻或者很清楚时，一般利用数值分析、偏微分方程等数学工具建立精确的数学模型，但当系统很复杂，或者系统未知、系统信息量很少时，建立精确的数学模型很困难，神经网络的非线性映射能力则表现出优势，因为它不需要对系统进行透彻地了解，但是同时能达到输入与输出的映射关系，这就大大简化了设计的难度。

(4) 高度并行性。并行性具有一定的争议性。承认具有并行性的理由：神经网络是根据人的大脑而抽象出来的数学模型，由于人可以同时做一些事，因此从功能的模拟角度来看，神经网络也应具备很强的并行性。

多少年以来，人们从医学、生物学、生理学、哲学、信息学、计算机科学、认知学、组织协同学等各个角度企图认识并解答上述问题。在寻找上述问题答案的研究过程中，这些

年来逐渐形成了一个新兴的多学科交叉技术领域，称为“神经网络”。神经网络的研究涉及众多学科领域，这些领域互相结合、相互渗透并相互推动。不同领域的科学家又从各自学科的兴趣与特色出发，提出了不同的问题，从不同的角度进行研究。

神经网络是所谓深度学习的一个基础，也是必备的知识点，它是以人脑中的神经网络作为启发，最著名的算法就是反向传播(Back Propagation，BP)算法。下面简单地整理一下神经网络的相关参数和计算方法。

1) 多层前向神经网络

多层前向神经网络(Multilayer Feed-Forward Neural Network)由输入层(Input Layer)、隐藏层(Hidden Layer)和输出层(Output Layer)组成。

多层前向神经网络的拓扑结构如图 3.15 所示。

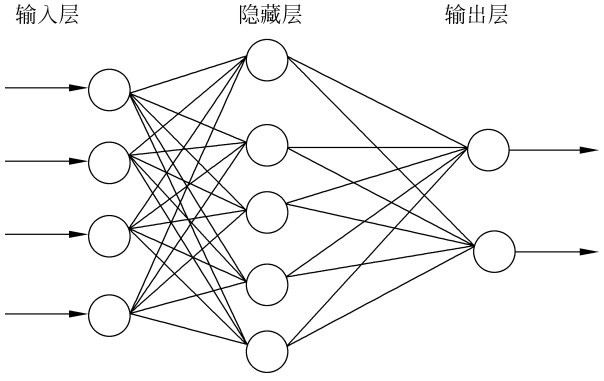


图 3.15　多层前向神经网络的拓扑结构

其特点如下：

(1) 每层由单元(Units)组成。

(2) 输入层是由训练集的实例特征向量传入的。

(3) 经过连接点的权重(Weight)传入下一层，一层的输出是下一层的输入。

(4) 隐藏层的个数可以是任意的，输入层有一层，输出层有一层。

(5) 每个单元也可以称为神经节点，根据生物学来源定义。

(6) 以上包含隐藏层和输出层，称为两层的神经网络，输入层是不算在里面的。

(7) 隐藏层中加权求和，然后根据非线性方程转换输出。

(8) 作为多层前向神经网络，理论上，如果有足够的隐藏层和足够的训练集，可以模拟出任何方程。

2) 设计神经网络结构

(1) 使用神经网络训练数据之前，必须确定神经网络的层数，以及每层单元的个数。

(2) 特征向量在被传入输入层时，通常要先标准化到 0～1(为了加速学习过程)。

(3) 离散型变量可以被编码成每一个输入单元对应一个特征值可能赋的值。

例如，特征值 A 可能取 3 个值（a_0、a_1 和 a_2），可以使用 3 个输入单元来代表 A。

如果 $A=a_0$，那么代表 a_0 的单元值就取 1，其他取 0；如果 $A=a_1$，那么代表 a_1 的单元值就取 1，其他取 0；以此类推。

（4）神经网络既可以用来解决分类（Classification）问题，也可以用来解决回归（Regression）问题。

对于分类问题，如果是 2 类，可以用一个输出单元表示（0 和 1 分别代表 1 类），如果多于 2 类，则每一个类别用一个输出单元表示。

没有明确的规则来设计最好有多少个隐藏层，可以根据实验测试和误差以及精准度来实验并改进。

3）交叉验证方法（cross-Validation）

这里有一堆数据，我们把它切分成三部分（当然还可以分得更多）：

- 第一部分做测试集，第二、三部分做训练集，计算出准确度；
- 第二部分做测试集，第一、三部分做训练集，计算出准确度；
- 第三部分做测试集，第一、二部分做训练集，计算出准确度。

之后计算出三个准确度的平均值，作为最后的准确度，如图 3.16 所示。

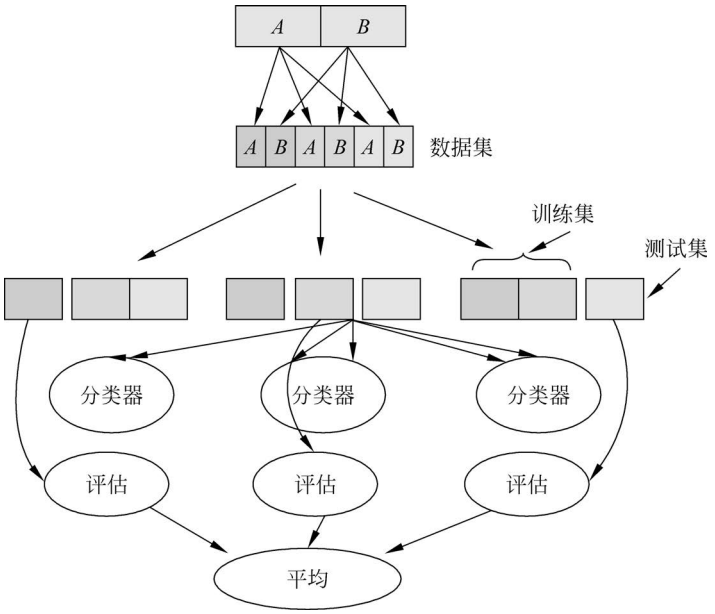


图 3.16　交叉验证方法流程图

4）反向传播算法

通过迭代性来处理训练集中的实例，对比经过神经网络后，输入层预测值与真实值之间的误差，再通过反向传播算法（输出层＝＞隐藏层＝＞输入层）以最小化误差来更新每个连接的权重。

算法的详细介绍如下。

- 输入：D(数据集)、学习率(Learning Rate)、一个多层前向神经网络。
- 输出：一个训练好的神经网络。

(1) 初始化权重和偏向：随机初始化在－1～1 或者－0.5～0.5，每个单元有一个偏向。

(2) 开始对数据进行训练，步骤如下：

① 由输入层向前传送。

② 根据误差(Error)反向传送。

终止条件如下。

- 方法一：权重的更新低于某个阈值。
- 方法二：预测的错误率低于某个阈值。
- 方法三：达到预设一定的循环次数。

3.4.4　覆盖正例排斥反例方法

覆盖正例排斥反例方法是利用覆盖所有正例、排斥所有反例的思想来寻找规则的。首先在正例集合中任选一个种子，到反例集合中逐个比较。与字段取值构成的选择子相容则舍去，相反则保留。按此思想循环所有正例种子，将得到正例的规则(选择子的合取式)。比较典型的算法有 Michalski 的 AQ11 方法、洪家荣改进的 AQ15 方法以及他的 AE5 方法。

AQ 算法共有 4 个输入参数：例子集合 dataSet、solution 集合最大容量 nSOL、consistent 集合最大容量 nCONS、候选表达式的数量 m。此外，还有自定义的优化标准，用以在算法执行过程中发现一些覆盖正例数相同的公式时，如何在这些公式中做出取舍，本书选择的是覆盖反例数最少的。nSOL 等参数会在后面算法的详细执行过程中解释。

AQ 算法返回值时规则 rule。主流程如图 3.17 所示，首先对 dataSet 进行划分，得到全正例集合 PE 和全反例集合 NE。整体思路是：从 PE 中选择一个例子 example 从这个 example 出发执行 induce 函数得到一个最优公式 formula，这个公式一定是一致的，rule＝rule∨formula 将 formula 覆盖的所有正例从 PE 中剔除，再从 PE 中重新选择一个例子执行这个过程直到 PE 为空，最后输出 rule。其实很多规则学习算法大体流程都是如此，例如 FOIL、GS 算法等，对正例集 PE 进行遍历，并不断删除新得到公式覆盖的例子。

AQ 算法的核心是 induce 函数，其主要目的是依据输入的例子 example 生成 solution 和 consistent 集合，solution 集合中存放的是一致且完备的公式，consistent 集合中存放的是一致但不完备的公式，如果执行一次后两个集合都没有满，即 solution 大小

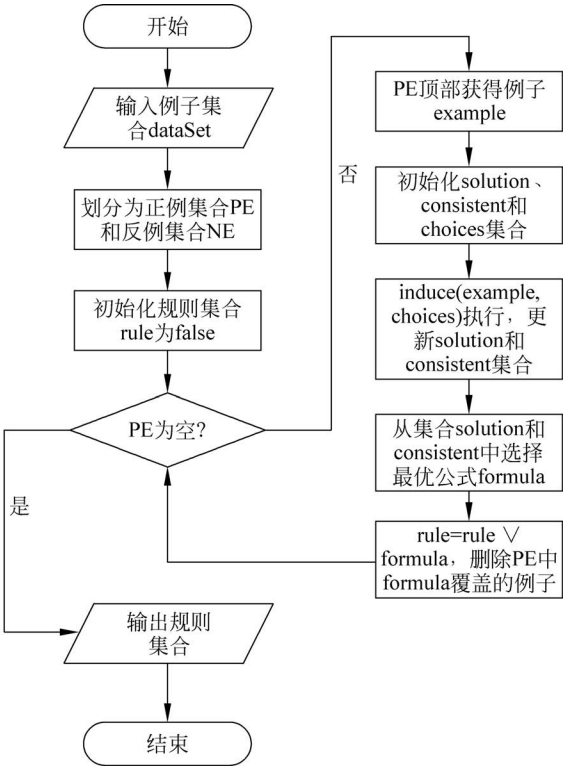


图 3.17　AQ 算法整体流程图

小于 nSOL、consistent 大小小于 nCONS，则会递归调用 induce 函数。AQ 算法的实现流程如图 3.18 所示。

设定 nCONS、nSOL 和 m 均为 1。对于例子 $e1$=（蓝色，淡黄，高），包含 3 个选择子 $x1$=蓝色，$x2$=淡黄，$x3$=高，考察这 3 个选择子的正反例覆盖情况：$x1$=蓝色，覆盖了 1 个正例、1 个反例；$x2$=淡黄，覆盖了 1 个正例、2 个反例；$x3$=高，覆盖了 2 个正例、1 个反例。很明显，这 3 个选择子没有一个是一致的，因此都不能直接加入 solution 或 consistent 集合中。假定 m=1，需要在这 3 个选择子中挑选 1 个进入下一个 induce 循环中，这里我们的选择标准是正例数和反例数的比值，比值越大，优先级就越高。

于是我们选择了 $x3$ =高，进入下一次 induce，现在尝试将其与 $x1$=蓝色、$x2$=淡黄进行合取，$x1$=蓝色 $\wedge$ $x3$=高覆盖了 1 个正例、0 个反例，是一致的，但不完备，$x2$=淡黄 $\wedge$ $x3$=高覆盖了 1 个正例、1 个反例，不一致。因此，将 $x1$=蓝色 $\wedge$ $x3$=高作为公式放入 consistent 中，因为事先设定 nCONS=1，因此对于例子 $e1$ 的 induce 执行结束。

比较 consistent 和 solution 集合中所有公式的表现，依然是使用（正例数/反例数）作为评价依据，选出最优的一个公式，这里只有一个 $x1$=蓝色 $\wedge$ $x3$=高，因此 rule=rule $\vee$（$x1$=蓝色 $\wedge$ $x3$=高）。发现新的规则覆盖了 $e1$，于是 PE 剔除 $e1$，继续下一轮循环，选

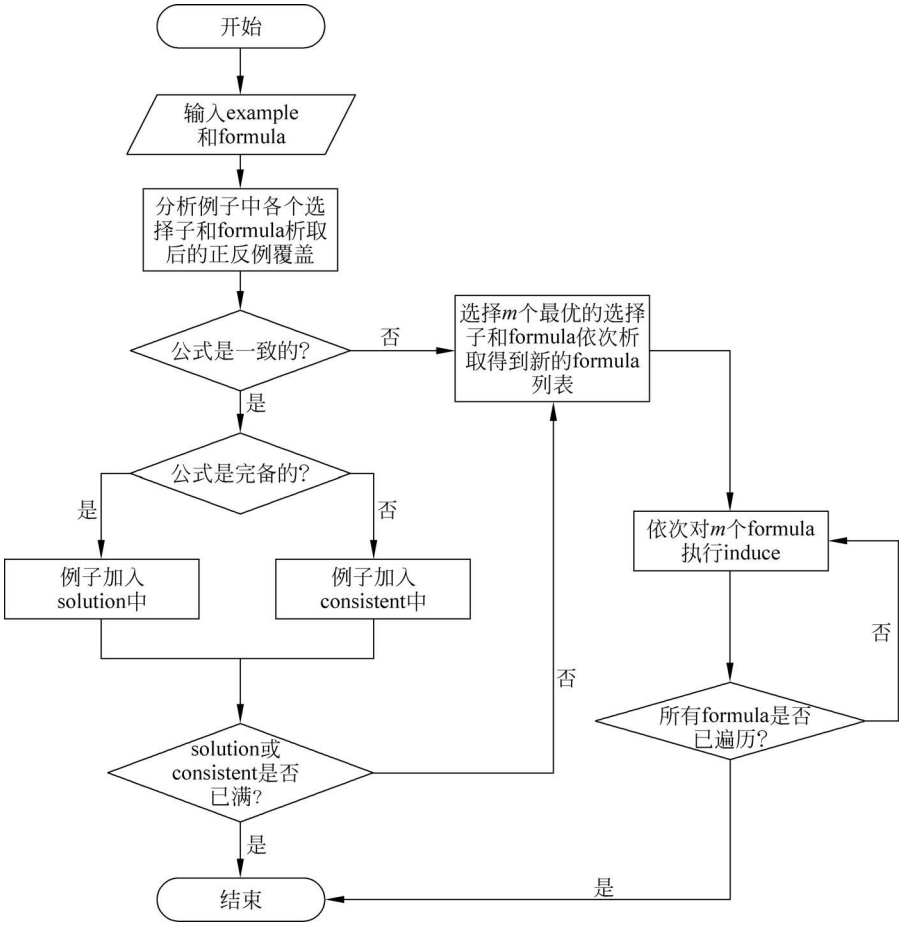


图 3.18　AQ 算法流程图

出 $e3$ 进行 induce。

最终得到一致且完备的规则为($x1$=蓝色 $\wedge$ $x3$=高) $\vee$ ($x2$=黑)。

3.4.5　粗糙集方法

粗糙集(Rough Set)理论是继概率论、模糊集、证据理论之后的又一个处理不确定性的数学工具。作为一种较新的软计算方法,粗糙集近年来越来越受重视,其有效性已在许多科学与工程领域的成功应用中得到证实,是当前国际上人工智能理论及其应用领域的研究热点之一。

在自然科学、社会科学和工程技术的很多领域中,都不同程度地涉及对不确定因素和不完备信息的处理。从实际系统中采集到的数据常常包含噪声,不够精确,甚至不完整。采用纯数学上的假设来消除或回避这种不确定性,效果往往不理想。反之,如果正

视它，对这些信息进行合适的处理，常常有助于相关实际系统问题的解决。

多年来，研究人员一直在努力寻找科学地处理不完整性和不确定性的有效途径。模糊集和基于概率方法的证据理论是处理不确定信息的两种方法，已应用于一些实际领域。但这些方法有时需要一些数据的附加信息或先验知识，如模糊隶属函数、基本概率指派函数和有关统计概率分布等，而这些信息有时并不容易得到。

1982 年，波兰学者 Z. Paw Lak 提出了粗糙集理论，它是一种刻画不完整性和不确定性的数学工具，能有效地分析不精确、不一致（Inconsistent）、不完整（Incomplete）等各种不完备的信息，还可以对数据进行分析和推理，从中发现隐含的知识，揭示潜在的规律。

粗糙集理论是建立在分类机制的基础上的，它将分类理解为在特定空间上的等价关系，而等价关系构成了对该空间的划分。粗糙集理论将知识理解为对数据的划分，每一被划分的集合称为概念。粗糙集理论的主要思想是利用已知的知识库，将不精确或不确定的知识用已知的知识库中的知识来（近似）刻画。

该理论与其他处理不确定和不精确问题的理论的显著区别是：它无须提供问题所需处理的数据集合之外的任何先验信息，所以对问题的不确定性的描述或处理可以说是比较客观的，由于这个理论不包含处理不精确或不确定原始数据的机制，因此这个理论与概率论、模糊数学和证据理论等其他处理不确定或不精确问题的理论有很强的互补性。

粗糙集是一种较有前途的处理不确定性的方法，相信今后将会在更多的领域中得到应用。但是，粗糙集理论还处在继续发展之中，正如粗糙集理论的提出人 Z. Paw Lak 所指出的那样，尚有一些理论上的问题需要解决，诸如用于不精确推理的粗糙逻辑（Rough Logic）方法，粗糙集理论与非标准分析（Nonstandard Analysis）和非参数化统计（Nonparametric statistics）等之间的关系等。将粗糙集与其他软计算方法（如模糊集、人工神经网络、遗传算法等）相综合，发挥出各自的优点，可望设计出具有较高的机器智商（Machine Intelligence Quotient，MIQ）的混合智能系统（Hybrid Intelligent System），这是一个值得努力的方向。

1）知识

“知识”这个概念在不同的范畴内有多种不同的含义。在粗糙集理论中，“知识”被认为是一种分类能力。人们的行为是基于分辨现实的或抽象的对象的能力，如在远古时代，人们为了生存，必须能分辨出什么可以食用，什么不可以食用；医生给病人诊断，必须辨别出患者得的是哪一种病。这些根据事物的特征差别将其分门别类的能力均可以看作是某种“知识”。

2）不可分辨关系

在分类过程中，相差不大的个体被归于同一类，它们的关系就是不可分辨关系（Indiscernibility Relation）。假定只用两种黑白颜色把空间中的物体分割为两类，即{黑

色物体}，{白色物体}，那么同为黑色的两个物体就是不可分辨的，因为描述它们特征属性的信息相同，都是黑色。

如果再引入方、圆的属性，又可以将物体进一步分割为4类，即{黑色方物体}，{黑色圆物体}，{白色方物体}，{白色圆物体}。这时，如果两个物体同为黑色方物体，则它们还是不可分辨的。不可分辨关系是一种等效关系(Equivalence Relationship)，两个白色圆物体间的不可分辨关系可以理解为它们在白、圆两种属性下存在等效关系。

3) 基本集

基本集(Elementary Set)定义为由论域中相互间不可分辨的对象组成的集合，它是组成论域知识的颗粒。不可分辨关系这一概念在粗糙集理论中十分重要，它深刻地揭示出知识的颗粒状结构是定义其他概念的基础。知识可认为是一族等效关系，它将论域分割成一系列的等效类。

4) 集合

粗糙集理论延拓了经典的集合论，把用于分类的知识嵌入集合内，作为集合组成的一部分。一个对象 a 是否属于集合 X，需根据现有的知识来判断，可分为3种情况：

(1) 对象 a 肯定属于集合 X。

(2) 对象 a 肯定不属于集合 X。

(3) 对象 a 可能属于集合 X，也可能不属于集合 X。

集合的划分密切依赖于我们所掌握的关于论域的知识，是相对的，而不是绝对的。给定一个有限的非空集合 U，称为论域，I 为 U 中的一组等效关系，即关于 U 的知识，则二元对 $K=(U,I)$ 称为一个近似空间(Approximation Space)。设 x 为 U 中的一个对象，X 为 U 的一个子集，$I(x)$ 表示所有与 x 不可分辨的对象所组成的集合，换句话说，是由 x 决定的等效类，即 $I(x)$ 中的每个对象都与 x 有相同的特征属性(Attribute)。

在数据库中，将行元素看成对象，将列元素看成属性(分为条件属性和决策属性)。等价关系 R 定义为不同对象在某个或几个属性上取值相同，满足等价关系的对象组成的集合被称为等价关系 R 的等价类。条件属性上的等价类 E 与决策属性上的等价类 Y 之间的关系分为以下3种情况。

(1) 下近似：Y 包含 E。对下近似建立确定性规则。

(2) 上近似：Y 和 E 的交非空。对上近似建立不确定性规则(含可信度)。

(3) 无关：Y 和 E 的交为空。无关情况不存在规则。

5) 粗糙集的特点

粗糙集方法的简单实用性是令人惊奇的，它能在创立后的不长时间内得到迅速应用是因为具有以下特点：

(1) 它能处理各种数据，包括不完整(Incomplete)的数据以及拥有众多变量的数据。

（2）它能处理数据的不精确性和模棱两可（Ambiguity），包括确定性和非确定性的情况。

（3）它能求得知识的最小表达（Reduct）和知识的各种不同颗粒（Granularity）层次。

（4）它能从数据中揭示出概念简单、易于操作的模式（Pattern）。

（5）它能产生精确而又易于检查和证实的规则，适用于智能控制中规则的自动生成。

6）粗糙集理论的应用

粗糙集理论是一门实用性很强的学科，从诞生到现在虽然只有十几年的时间，但已经在不少领域取得了丰硕的成果，如近似推理、数字逻辑分析和化简、建立预测模型、决策支持、控制算法获取、机器学习算法和模式识别等。粗糙集能有效地处理下列问题：

（1）不确定或不精确知识的表达。

（2）经验学习并从经验中获取知识。

（3）不一致信息的分析。

（4）根据不确定、不完整的知识进行推理。

（5）在保留信息的前提下进行数据化简。

（6）识别并评估数据之间的依赖关系。

7）神经网络样本化简

人工神经网络具有并行处理、高度容错和泛化能力强的特点，适合应用于预测、复杂对象建模和控制等场合。但是当神经网络规模较大、样本较多时，训练时间过于漫长，这个固有缺点是制约神经网络进一步实用化的一个主要因素。虽然各种提高训练速度的算法不断出现，问题远未彻底解决。化简训练样本集，消除冗余数据是另一条提高训练速度的途径。

8）控制算法获取

实际系统中有很多复杂对象难以建立严格的数学模型，这样传统的基于数学模型的控制方法就难以奏效。模糊控制模拟人的模糊推理和决策过程，将操作人员的控制经验总结为一系列语言控制规则，具有稳健性和简单性的特点，在工业控制等领域发展较快。但是有些复杂对象的控制规则难以人工提取，这样就在一定程度上限制了模糊控制的应用。

粗糙集能够自动抽取控制规则的特点为解决这一难题提供了新的手段。一种新的控制策略——模糊-粗糙控制（Fuzzy-Rough Control）正悄然兴起，成为一个有吸引力的发展方向。有学者应用这种控制方法研究了“小车-倒立摆系统”这一经典控制问题和水泥窑炉的过程控制问题，均取得了较好的控制效果。应用粗糙集进行控制的基本思路是：把控制过程的一些有代表性的状态以及操作人员在这些状态下所采取的控制策略都记录下来，然后利用粗糙集理论处理这些数据，分析操作人员在哪种条件下采取哪种控制策略，总结出了一系列控制规则，分别说明如下。

- 规则 1：IF Condition 1 满足 THEN 采取 decision 1。
- 规则 2：IF Condition 2 满足 THEN 采取 decision 2。
- 规则 3：IF Condition 3 满足 THEN 采取 decision 3。

这种根据观测数据获得控制策略的方法通常被称为从范例中学习(Learning From Examples)。粗糙控制(Rough Control)与模糊控制都是基于知识、基于规则的控制，但粗糙控制更加简单迅速，实现容易(因为粗糙控制有时可省去模糊化及去模糊化的步骤)；另一个优点在于控制算法可以完全来自数据本身，所以从软件工程的角度来看，其决策和推理过程与模糊(或神经网络)控制相比很容易被检验和证实(Validate)。有研究指出，在特别要求控制器结构与算法简单的场合，更适合采取粗糙控制。

9) 决策支持系统

面对大量的信息以及各种不确定因素，要做出科学、合理的决策是非常困难的。决策支持系统是一组协助制定决策的工具，其重要特征就是能够执行 IF THEN 规则进行判断分析。粗糙集理论可以在分析以往大量经验数据的基础上找到这些规则，基于粗糙集的决策支持系统在这方面弥补了常规决策方法的不足，允许决策对象中存在一些不太明确、不太完整的属性，并经过推理得出基本上肯定的结论。

3.4.6　遗传算法

遗传算法(Genetic Algorithm，GA)最早是由美国的 Johnholland 于 20 世纪 70 年代提出的，该算法是根据大自然中生物体的进化规律而设计提出的。该算法通过数学的方式，利用计算机仿真运算，将问题的求解过程转换成类似生物进化中的染色体基因的交叉、变异等过程。在求解较为复杂的组合优化问题时，相对一些常规的优化算法，通常能够较快地获得较好的优化结果。遗传算法已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域。

自然演变是一种基于群体的优化过程，在计算机上对这个过程进行仿真，产生了随机优化技术，在应用于解决现实世界中的难题时，这种技术常胜过经典的优化方法，遗传算法就是根据自然演变法则开发出来的。

遗传算法的起源可追溯到 20 世纪 60 年代初期。1967 年，美国密歇根大学 J. Holland 教授的学生 Bagley 在他的博士论文中首次提出了遗传算法这一术语，并讨论了遗传算法在博弈中的应用，但早期研究缺乏带有指导性的理论和计算工具的开拓。1975 年，J. Holland 等出版了专著《自然系统和人工系统的适配》，在书中系统地阐述了遗传算法的基本理论和方法，提出了对遗传算法理论研究极为重要的模式理论，推动了遗传算法的发展。20 世纪 80 年代后，遗传算法进入兴盛发展时期，被广泛应用于自动控制、生产计划、图像处理、机器人等研究领域。

1. 遗传算法的基本原理

遗传算法是不需要求导的随机优化方法，它以自然选择和演变过程为基础，但是联系又是不牢靠的。遗传算法是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。遗传算法是从代表问题可能潜在的解集的一个种群（Population）开始的，而一个种群则由经过基因（Gene）编码的一定数目的个体（individual）组成。每个个体实际上是染色体（Chromosome）带有特征的实体。染色体作为遗传物质的主要载体，即多个基因的集合，其内部表现（基因型）是某种基因组合，它决定了个体的形状的外部表现，如黑头发的特征是由染色体中控制这一特征的某种基因组合决定的。因此，在一开始需要实现从表现型到基因型的映射，即编码工作。由于仿照基因编码的工作很复杂，我们往往需要进行简化，如二进制编码，初代种群产生之后，按照适者生存和优胜劣汰的原理，逐代（Generation）演化产生越来越好的近似解，在每一代，根据问题域中个体的适应度（Fitness）大小选择（Selection）个体，并借助自然遗传学的遗传算子（Genetic Operator）进行组合交叉（Crossover）和变异（Mutation），产生代表新的解集的种群。这个过程将导致种群像自然进化一样的后生代种群比前代更加适应环境，末代种群中的最优个体经过解码（Decoding），可以作为问题近似最优解。

2. 遗传算法的特性

遗传算法是解决搜索问题的一种通用算法，对于各种通用问题都可以使用。搜索算法的共同特征如下：

（1）首先组成一组候选解。

（2）依据某些适应性条件测算这些候选解的适应度。

（3）根据适应度保留某些候选解，放弃其他候选解。

（4）对保留的候选解进行某些操作，生成新的候选解。

在遗传算法中，上述几个特征以一种特殊的方式组合在一起：基于染色体群的并行搜索，带有猜测性质的选择操作、交换操作和突变操作。这种特殊的组合方式将遗传算法与其他搜索算法区分开来。

遗传算法还具有以下几方面的特点：

（1）算法从问题解的串集开始搜索，而不是从单个解开始。这是遗传算法与传统优化算法的极大区别。传统优化算法是从单个初始值迭代求最优解的，容易误入局部最优解。遗传算法从串集开始搜索，覆盖面大，有利于全局择优。

（2）遗传算法同时处理群体中的多个个体，即对搜索空间中的多个解进行评估，减少了陷入局部最优解的风险，同时算法本身易于实现并行化。

（3）遗传算法基本上不用搜索空间的知识或其他辅助信息，而仅用适应度函数值来

评估个体，在此基础上进行遗传操作。适应度函数不仅不受连续可微的约束，而且其定义域可以任意设定。这一特点使得遗传算法的应用范围大大扩展。

(4) 遗传算法不是采用确定性规则，而是采用概率的变迁规则来指导它的搜索方向。

(5) 具有自组织、自适应和自学习性。遗传算法利用进化过程获得的信息自行组织搜索时，适应度大的个体具有较高的生存概率，并获得更适应环境的基因结构。

(6) 此外，算法本身也可以采用动态自适应技术，在进化过程中自动调整算法控制参数和编码精度，例如使用模糊自适应法。

3. 遗传算法的基本运算

(1) 初始化：设置进化代数计数器 $t=0$，设置最大进化代数 T，随机生成 M 个个体作为初始群体 $P(0)$。

(2) 个体评价：计算群体 $P(t)$中各个个体的适应度。

(3) 选择运算：将选择算子作用于群体。选择的目的是把优化的个体直接遗传到下一代或通过配对交叉产生新的个体再遗传到下一代。选择操作是建立在群体中个体的适应度评估基础上的。

(4) 交叉运算：将交叉算子作用于群体。遗传算法中起核心作用的就是交叉算子。

(5) 变异运算：将变异算子作用于群体，即对群体中的个体串的某些基因座上的基因值进行变动。群体 $P(t)$经过选择、交叉、变异运算之后得到下一代群体 $P(t+1)$。

(6) 终止条件判断：若 $t=T$，则以进化过程中所得到的具有最大适应度的个体作为最优解输出，终止计算。

4. 用遗传算法进行优化

(1) 编码方案和初始化。

(2) 适合度设计。

(3) 选择。

(4) 交叉。

(5) 突变。

这是采用遗传算法进行优化的 5 个步骤。

5. 遗传算法的简单例证

要对某个问题应用遗传算法，必须定义或选择以下 5 部分：

(1) 为问题的潜在解选择遗传表述或编码方案。

(2) 一种创建潜在解的初始群体的方法。

(3) 一个评价函数，它扮演着环境的决策，根据“适合度”对解进行评级。

(4) 改变后代成分的遗传算子。

（5）遗传算法使用的不同参数值(群体大小、应用算子的比率等)。

6. 遗传算法的不足之处

（1）编码不规范及编码存在表示的不准确性。

（2）单一的遗传算法编码不能全面地将优化问题的约束表示出来。考虑约束的一个方法就是对不可行解采用阈值，这样计算的时间必然增加。

（3）通常遗传算法的效率比其他传统的优化方法低。

（4）遗传算法容易过早收敛。

（5）遗传算法在算法的精度、可行度、计算复杂性等方面还没有有效的定量分析方法。

7. 遗传算法的应用

由于遗传算法的整体搜索策略和优化搜索方法在计算时不依赖于梯度信息或其他辅助知识，而只需要影响搜索方向的目标函数和相应的适应度函数，因此遗传算法提供了一种求解复杂系统问题的通用框架，它不依赖于问题的具体领域，对问题的种类有很强的稳健性，广泛应用于许多科学。下面介绍遗传算法的一些主要应用领域。

1）函数优化

函数优化是遗传算法的经典应用领域，也是遗传算法进行性能评价的常用算例，许多人构造出了各种各样复杂形式的测试函数：连续函数和离散函数、凸函数和凹函数、低维函数和高维函数、单峰函数和多峰函数等。对于一些非线性、多模型、多目标的函数优化问题，用其他优化方法较难求解，而遗传算法可以方便地得到较好的结果。

2）组合优化

随着问题规模的增大，组合优化问题的搜索空间也急剧增大，有时在计算上用枚举法很难求出最优解。对于这类复杂的问题，人们已经意识到应把主要精力放在寻求满意解上，而遗传算法是寻求这种满意解的最佳工具之一。实践证明，遗传算法对于组合优化中的 NP 问题非常有效。例如遗传算法已经在求解旅行商问题、背包问题、装箱问题、图形划分问题等方面得到成功的应用。

此外，GA 也在生产调度问题、自动控制、机器人学、图像处理、人工生命、遗传编码和机器学习等方面获得了广泛的运用。

3）车间调度

车间调度问题是一个典型的 NP-Hard 问题，遗传算法作为一种经典的智能算法，广泛用于车间调度中，很多学者都致力于用遗传算法解决车间调度问题，现今也取得了十分丰硕的成果。从最初的传统车间调度问题(JSP)到柔性作业车间调度问题(FJSP)，遗传算法都有优异的表现，在很多算例中都得到了最优或近优解。

3.4.7　公式发现

在工程和科学数据库中，对若干数据项进行一定的数学运算，可求得相应的数学公式。BACON 发现系统完成了对物理学的大量定律的重新发现。下面重点介绍 FDD 公式发现系统。

FDD 系统的基本思想是利用人工智能启发式搜索函数原型寻找具有最佳线性逼近关系的函数原型，并结合曲线拟合技术及可视化技术来寻找数据间的规律。

启发式方法是求解人工智能问题的一个重要方法。一般启发式方法是建立启发式函数，用以引导搜索方向，以便用尽量少的搜索次数，从开始状态达到最终状态。FDD 系统在执行搜索的过程中，对原型函数进行搜索以及对它们的组合函数进行搜索，也是一种组合爆炸现象。为解决这一问题，在设计系统时采用了启发式方法来实现。对某一变量取初等函数，和另一变量的初等函数或原始数据进行线性组合，即从原型库中选取逼近效果最好的少数几个初等函数作为基函数，并进一步形成组合函数，直至找到最后的目标函数。

FDD 算法的基本思想是不断对两组变量进行学习，找出学习后两组变量的数学关系式。具体做法是：首先固定变量 x_2，对 x_1 进行学习，即在现有原型库的基础上，对变量 x_1 进行匹配，得到一组新的数据 x_1'。将两组数据代入知识库中的启发式函数关系式，用最小二乘法求出 a、b 系数，将求出的 a、b 系数代入误差分析模块中的线性逼近误差公式，若求出的误差小于一个给定值，则学习成功，否则继续重复以上操作。如此循环下去，直到最后得出的误差值小于一个给定值。

FDD 算法主要包括以下 3 个步骤。

(1) 匹配步骤：对变量的学习过程。原型库中变量的选取尤为重要，函数选取恰当会起到事半功倍的效果；选取不当，FDD 算法的搜索方向会偏离预期的搜索方向，误差会随着搜索层数的增加而增加。原型库中函数原型的数量越多，变量的学习范围就越大，这样最佳的拟合函数就越容易挖掘到。

(2) 系数求解步骤：采用最小二乘法。将学习后的变量代入系统的启发式，$f(x_2)=a+bf_1(x_1)$中，组合成一个二元一次方程组，方程组的解就是所求的 a、b 系数。

(3) 误差求解步骤：分别求出每组变量的相对误差并求和。

从对 FDD 算法的分析中可以看出，要想找出最佳的拟合公式，就要反复进行以上的 3 步操作。对于第一个步骤，无休止地对变量进行匹配，最后得出的函数关系式并不是一个反映直观概念的关系式，因此有必要对匹配的层数硬性规定一个范围，如果层数达到最大限制，就要考虑选用搜索的其他方向。

3.4.8 统计分析方法

在数据库字段项之间存在两种关系：函数关系(能用函数公式表示的确定性关系)和相关关系(不能用函数公式表示,但仍是相关确定性关系),对它们的分析可采用回归分析、相关分析、主成分分析等方法。

统计分析方法包括逻辑思维方法和数量关系分析方法。在统计分析中,二者密不可分,应结合运用。

1. 逻辑思维方法

逻辑思维方法是指辩证唯物主义认识论的方法。统计分析必须以马克思主义哲学作为世界观和方法论的指导。唯物辩证法对于事物的认识要从简单到复杂,从特殊到一般,从偶然到必然,从现象到本质。坚持辩证的观点、发展的观点,从事物的发展变化中观察问题,从事物的相互依存、相互制约中分析问题,对统计分析具有重要的指导意义。

2. 数量关系分析方法

数量关系分析方法是运用统计学中论述的方法对社会经济现象的数量表现,包括社会经济现象的规模、水平、速度、结构比例、事物之间的联系进行分析的方法,如对比分析法、平均和变异分析法、综合评价分析法、结构分析法、平衡分析法、动态分析法、因素分析法、相关分析法等。

1) 回归分析法

在大数据分析中,回归分析是一种预测性的建模技术,它研究的是因变量(目标)和自变量(预测器)之间的关系。这种技术通常用于预测分析、时间序列模型以及发现变量之间的因果关系。例如,司机的鲁莽驾驶与道路交通事故数量之间的关系,最好的研究方法就是回归方法。

有各种各样的回归技术用于预测。这些技术主要有 3 个度量,分别是自变量的个数、因变量的类型以及回归线的形状。

(1) 线性回归。

线性回归(Linear Regression)是最为人熟知的建模技术之一。线性回归通常是人们在学习预测模型时首选的技术之一。在这种技术中,因变量是连续的,自变量可以是连续的,也可以是离散的,回归线的性质是线性的。

线性回归使用最佳的拟合直线(也就是回归线)在因变量(Y)和一个或多个自变量(X)之间建立一种关系。

多元线性回归可表示为 $Y=a+b_1\times X_1+b_2\times X_2+e$,其中 a 表示截距,b 表示直线的斜率,e 是误差项。多元线性回归可以根据给定的预测变量(s)来预测目标变量的值。

(2) 逻辑回归。

逻辑回归(Logistic Regression)用来计算“事件＝Success”和“事件＝Failure”的概率。当因变量的类型属于二元(1/0,真/假,是/否)变量时,应该使用逻辑回归。这里,Y的值为 0 或 1,它可以用以下方程表示。

$$\text{odds}=p/(1-p)=\text{事件发生的概率}/\text{事件未发生的概率}$$

$$\ln(\text{odds})=\ln(p/(1-p))$$

$$\text{Logit}(p)=\ln(p/(1-p))=b_0+b_1\times 1+b_2\times 2+b_3\times 3+\cdots+b_k\times k$$

上述式子中,p 表述具有某个特征的概率。你应该会问这样一个问题:“为什么要在公式中使用对数函数 ln 呢?”

因为在这里使用的是二项分布(因变量),需要选择一个对于这个分布最佳的连接函数,它就是 Logit 函数。在上述方程中,通过观测样本的极大似然估计值来选择参数,而不是最小化平方和误差(如在普通回归中使用)。

(3) 多项式回归。

对于一个回归方程,如果自变量的指数大于 1,它就是多项式回归(Polynomial Regression)方程。例如以下方程:

$$y=a+b\times x^2$$

在这种回归技术中,最佳拟合线不是直线,而是一个用于拟合数据点的曲线。

(4) 逐步回归。

在处理多个自变量时,可以使用逐步回归(Stepwise Regression)。在这种技术中,自变量的选择是在一个自动的过程中完成的,其中包括非人为操作。

这一壮举通过观察统计的值(如 R-square、t-stats 和 AIC 指标)来识别重要的变量。逐步回归通过同时添加/删除基于指定标准的协变量来拟合模型。下面列出一些常用的逐步回归方法:

- 标准逐步回归法做两件事情,即增加和删除每个步骤所需的预测。
- 向前选择法从模型中最显著的预测开始,为每一步添加变量。
- 向后剔除法与模型的所有预测同时开始,然后在每一步消除最小显著性的变量。

这种建模技术的目的是使用最少的预测变量数来最大化预测能力。这也是处理高维数据集的方法之一。

(5) 岭回归。

当数据之间存在多重共线性(自变量高度相关)时,就需要使用岭回归(Ridge Regression)分析。在存在多重共线性时,尽管普通最小二乘法(Ordinary Least Squares,OLS)测得的估计值不存在偏差,它们的方差也会很大,从而使得观测值与真实值相差甚远。岭回归通过给回归估计值添加一个偏差值来降低标准误差。

在线性等式中,预测误差可以划分为 2 个分量,一个是偏差造成的,另一个是方差造

成的。预测误差可能会由这两者或两者中的任何一个造成。在这里将讨论由方差造成的误差。

岭回归通过收缩参数λ(Lambda)解决多重共线性问题。请看下面的等式：

$$L_2 = \text{argmin} \| \boldsymbol{y} = x\boldsymbol{\beta} \| + \lambda \| \boldsymbol{\beta} \|$$

在这个公式中，有两个组成部分。一个是最小二乘项，另一个是$\boldsymbol{\beta}$的λ倍，其中$\boldsymbol{\beta}$是相关系数向量，与收缩参数一起添加到最小二乘项中以得到一个非常低的方差。

(6) 套索回归。

套索回归(Lasso Regression)类似于岭回归，LASSO(Least Absolute Shrinkage and Selection Operator)也会就回归系数向量给出惩罚值项。此外，它能够减少变化程度并提高线性回归模型的精度。请看下面的公式：

$$L_1 = \text{agrmin} \| \boldsymbol{y} - x\boldsymbol{\beta} \| + \lambda \| \boldsymbol{\beta} \|$$

LASSO 回归与 Ridge 回归有一点不同，它使用的惩罚函数是 L_1 范数，而不是 L_2 范数。这导致惩罚(或等于约束估计的绝对值之和)值使一些参数估计结果等于零。使用的惩罚值越大，进一步估计会使得缩小值越趋近于零。这将导致要从给定的 n 个变量中选择变量。

如果预测的一组变量是高度相关的，LASSO 会选出其中一个变量并且将其他的变量收缩为零。

(7) ElasticNet 回归。

ElasticNet 是 LASSO 和 Ridge 回归技术的混合体。它使用 L_1 来训练并且 L_2 优先作为正则化矩阵。当有多个相关的特征时，ElasticNet 是很有用的。LASSO 会随机挑选它们中的一个，而 ElasticNet 则会选择两个。

LASSO 和 Ridge 之间的实际优点是，它允许 ElasticNet 继承循环状态下 Ridge 的一些稳定性。

2) 相关分析法

相关分析就是对总体中确实具有联系的标志进行分析，其主体是对总体中具有因果关系标志的分析。相关分析是描述客观事物相互间关系的密切程度并用适当的统计指标表示出来的过程。在一段时期内，出生率随经济水平的上升而上升，这说明两个指标间是正相关关系；而在另一时期，随着经济水平的进一步发展，出现出生率下降的现象，两个指标间就是负相关关系。

为了确定相关变量之间的关系，首先应该收集一些数据，这些数据应该是成对的。例如，每个人的身高和体重。然后在直角坐标系上描述这些点，这一组点集称为“散点图”。

根据散点图，当自变量取某一值时，因变量对应为一个概率分布，如果对于所有的自变量取值的概率分布都相同，则说明因变量和自变量是没有相关关系的。反之，如果自

变量的取值不同，因变量的分布也不同，则说明两者是存在相关关系的。

两个变量之间的相关程度通过相关系数 r 来表示。相关系数 r 的值在 -1 和 1 之间，但可以是此范围内的任何值。正相关时，r 值在 0 和 1 之间，散点图是斜向上的，这时一个变量增加，另一个变量也增加；负相关时，r 值在 -1 和 0 之间，散点图是斜向下的，此时一个变量增加，另一个变量将减少。r 的绝对值越接近 1，两个变量的关联程度就越强，r 的绝对值越接近 0，两个变量的关联程度就越弱。

相关分析与回归分析在实际应用中有密切的关系。然而在回归分析中，所关心的是一个随机变量 Y 对另一个（或一组）随机变量 X 的依赖关系的函数形式。在相关分析中，所讨论的变量的地位一样，分析侧重于随机变量之间的种种相关特征。例如，以 X、Y 分别记小学生的数学与语文成绩，感兴趣的是二者的关系如何，而不在于由 X 去预测 Y。

要确定相关关系的存在，相关关系呈现的形态和方向，相关关系的密切程度，主要方法是绘制相关图表和计算相关系数。

（1）相关表。

编制相关表前，首先要通过实际调查取得一系列成对的标志值资料作为相关分析的原始数据。

相关表的分类：简单相关表和分组相关表。单变量分组相关表：自变量分组并计算次数，而对应的因变量不分组，只计算其平均值，该表的特点是使冗长的资料简化，能够更清晰地反映出两个变量之间的相关关系。双变量分组相关表：自变量和因变量都进行分组而制成的相关表，这种表形似棋盘，故又称棋盘式相关表。

（2）相关图。

利用直角坐标系第一象限把自变量置于横轴上，因变量置于纵轴上，而将两个变量相对应的变量值用坐标点形式描绘出来，用以表明相关点分布状况的图形。相关图被形象地称为相关散点图。因素标志分了组，结果标志表现为组平均数，所绘制的相关图就是一条折线，这种折线又叫相关曲线。

（3）相关系数。

① 相关系数是按积差方法计算，同样以两个变量与各自平均值的离差为基础，通过两个离差相乘来反映两个变量之间的相关程度，着重研究线性的单相关系数。

② 确定相关关系的数学表达式。

③ 确定因变量估计值误差的程度。

3）主成分分析法

主成分分析（Principal Component Analysis，PCA）是一种统计方法。通过正交变换将一组可能存在相关性的变量转换为一组线性不相关的变量，转换后的这组变量叫主成分。

在用统计分析方法研究多变量的课题时，变量个数太多就会增加课题的复杂性。人

们自然希望变量个数较少而得到的信息较多。在很多情形下，变量之间是有一定的相关关系的，当两个变量之间有一定的相关关系时，可以解释为这两个变量反映此课题的信息有一定的重叠。主成分分析是对于原先提出的所有变量，将重复的变量（关系紧密的变量）删去多余部分，建立尽可能少的新变量，使得这些新变量是两两不相关的，而且这些新变量在反映课题的信息方面尽可能保持原有的信息。

设法将原来的变量重新组合成一组新的互不相关的几个综合变量，同时根据实际需要从中取出几个较少的综合变量，尽可能多地反映原来的变量的信息的统计方法叫作主成分分析，或称为主分量分析，这是数学上用来降维的一种方法。

（1）基本思想。

主成分分析是设法将原来众多具有一定相关性的指标（例如 P 个指标），重新组合成一组新的互不相关的综合指标来代替原来的指标。

主成分分析是考查多个变量间的相关性的一种多元统计方法，研究如何通过少数几个主成分来揭示多个变量间的内部结构，即从原始变量中导出少数几个主成分，使它们尽可能多地保留原始变量的信息，且彼此间互不相关。通常数学上的处理就是将原来的 P 个指标进行线性组合，作为新的综合指标。

最经典的做法就是用 F_1（选取的第一个线性组合，即第一个综合指标）的方差来表达，即 $\mathrm{Var}(F_1)$越大，表示 F_1 包含的信息越多。因此，在所有的线性组合中，选取的 F_1 应该是方差最大的，故称 F_1 为第一主成分。如果第一主成分不足以代表原来的 P 个指标的信息，再考虑选取 F_2，即选取第二个线性组合。为了有效地反映原来的信息，F_1 已有的信息就不需要再出现在 F_2 中，用数学语言表达就是要求 $\mathrm{Cov}(F_1,F_2)=0$，则称 F_2 为第二主成分，以此类推，可以构造出第三主成分，第四主成分，……，第 P 主成分。

（2）步骤。

$$F_p = a_{1i} \cdot ZX_1 + a_{2i} \cdot ZX_2 + \cdots + a_{pi} \cdot ZX_p$$

其中 $a_{1i},a_{2i},\cdots,a_{pi}(i=1,\cdots,m)$ 为 X 的协方差阵 $\boldsymbol{\Sigma}$ 的特征值所对应的特征向量，$ZX_1,ZX_2,\cdots,ZX_p$ 是原始变量经过标准化处理的值，因为在实际应用中，往往存在指标的量纲不同，所以在计算之前需先消除量纲的影响，而将原始数据标准化，本书所采用的数据就存在量纲影响（注：本书指的数据标准化是指 Z 标准化）。

$\boldsymbol{A}=(a_{ij})p \cdot m=(\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_m)$，$\boldsymbol{R}\boldsymbol{a}_i=\lambda_i\boldsymbol{a}_i$，$\boldsymbol{R}$ 为相关系数矩阵，λ_i、$\boldsymbol{a}_i$ 是相应的特征值和单位特征向量，$\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_p\geqslant 0$。

进行主成分分析的主要步骤如下：

① 指标数据标准化（SPSS 软件自动执行）。

② 指标之间的相关性判定。

③ 确定主成分个数 m。

④ 主成分 F_i 表达式。

⑤ 主成分 F_i 命名。

(3) 基本原理。

主成分分析法是一种降维的统计方法，它借助一个正交变换，将其分量相关的原随机向量转换成其分量不相关的新随机向量，这在代数上表现为将原随机向量的协方差阵变换成对角形阵，在几何上表现为将原坐标系变换成新的正交坐标系，使之指向样本点散布最开的 p 个正交方向，然后对多维变量系统进行降维处理，使之能以一个较高的精度转换成低维变量系统，再通过构造适当的价值函数，进一步把低维系统转换成一维系统。

(4) 主成分分析的主要作用。

概括来说，主成分分析主要有以下几个方面的作用。

① 主成分分析能降低所研究的数据空间的维数，即用研究 m 维的 Y 空间代替 p 维的 X 空间($m<p$)，而低维的 Y 空间代替高维的 X 空间所损失的信息很少。即使只有一个主成分 Y_1($m=1$)，这个 Y_1 仍是使用全部 X 变量(p 个)得到的。例如要计算 Y_1 的均值，也得使用全部 X 的均值。在所选的前 m 个主成分中，如果某个 X_i 的系数全部近似于零，就可以把这个 X_i 删除，这也是一种删除多余变量的方法。

② 有时可通过因子负荷 a_{ij} 的结论弄清 X 变量间的某些关系。

③ 多维数据的一种图形表示方法。我们知道当维数大于 3 时，便不能画出几何图形，多元统计研究的问题大都多于 3 个变量。要把研究的问题用图形表示出来是不可能的。然而，经过主成分分析后，我们可以选取前两个主成分或其中某两个主成分，根据主成分的得分，画出 n 个样品在二维平面上的分布情况，由图形可以直观地看出各样品在主分量中的地位，进而对样本进行分类处理，可以由图形发现远离大多数样本点的离群点。

④ 由主成分分析法构造回归模型，即把各主成分作为新自变量代替原来的自变量 x 进行回归分析。

⑤ 用主成分分析筛选回归变量。回归变量的选择有着重要的实际意义，为了使模型本身易于进行结构分析、控制和预报，以便从原始变量所构成的子集合中选择最佳变量，构成最佳变量集合。用主成分分析筛选变量，可以用较少的计算量来选择量，以获得选择最佳变量子集合的效果。

(5) 主成分分析法的优点。

主成分分析法的优点是方法简单，工作量小。

(6) 主成分分析法的缺点。

主成分分析法的缺点是定额的准确性差，可靠性差。

① 对历史统计数据的完整性和准确性要求高，否则制定的标准没有任何意义。

② 统计数据分析方法选择不当会严重影响标准的科学性。

③ 统计资料只反映历史的情况而不反映现实条件的变化对标准的影响。

④ 利用本企业的历史性统计资料为某项工作确定标准，可能低于同行业的先进水平，甚至是平均水平。

3.4.9 模糊理论方法

模糊理论(Fuzzy Theory)是指用到了模糊集合的基本概念或连续隶属度函数的理论。它可分类为模糊数学、模糊系统、不确定性与信息、模糊决策、模糊逻辑与人工智能这 5 个分支，它们并不是完全独立的，它们之间有紧密的联系。例如，模糊控制就会用到模糊数学和模糊逻辑中的概念。从实际应用的观点来看，模糊理论的应用大部分集中在模糊系统上，尤其集中在模糊控制上，也有一些模糊专家系统应用于医疗诊断和决策支持。由于模糊理论从理论和实践的角度来看仍然是新生事物，因此我们期望随着模糊领域的成熟，将会出现更多可靠的实际应用。

1. 模糊理论简介

概念是思维的基本形式之一，它反映了客观事物的本质特征。人类在认识过程中，把感觉到的事物的共同特点抽象出来加以概括，这就形成了概念。例如从白雪、白马、白纸等事物中抽象出“白”的。一个概念有它的内涵和外延，内涵是指该概念所反映的事物本质属性的总和，也就是概念的内容。外延是指一个概念所确指的对象的范围。例如“人”这个概念的内涵是指能制造工具，并使用工具进行劳动的动物，外延是指古今中外一切的人。

所谓模糊概念，是指这个概念的外延具有不确定性，或者说它的外延是不清晰的，是模糊的。例如“青年”这个概念，它的内涵我们是清楚的，但是它的外延，即什么样的年龄阶段内的人是青年，恐怕就很难说清楚，因为在“年轻”和“不年轻”之间没有一个确定的边界，这就是一个模糊概念。

需要注意的几点：首先，人们在认识模糊性时，是允许有主观性的，也就是说每个人对模糊事物的界限不完全一样，承认一定的主观性是认识模糊性的一个特点。例如，我们让 100 个人说出“年轻人”的年龄范围，那么我们将得到 100 个不同的答案。尽管如此，当我们用模糊统计的方法进行分析时，年轻人的年龄界限分布又具有一定的规律性。

其次，模糊性是精确性的对立面，但不能消极地理解模糊性代表的是落后的生产力，恰恰相反，我们在处理客观事物时，经常借助于模糊性。例如，在一个有许多人的房间里，找一位“年老的高个子男人”，这是不难办到的。这里所说的“年老”“高个子”都是模糊概念，然而我们只要将这些模糊概念经过头脑的分析判断，很快就可以在人群中找到此人。如果我们要求用计算机查询，那么就要把所有人的年龄、身高的具体数据输入计算机，然后才可以从人群中找这样的人。

最后，人们对模糊性的认识往往会与随机性混淆起来，其实它们之间有着根本的区别。随机性是其本身具有明确的含义，只是由于发生的条件不充分，而使得在条件与事件之间不能出现确定的因果关系，从而事件的出现与否表现出一种不确定性。而事物的模糊性是指我们要处理的事物的概念本身就是模糊的，即一个对象是否符合这个概念难以确定，也就是由于概念外延模糊而带来的不确定性。

2. 模糊理论的发展

模糊理论是在美国加州大学伯克利分校电气工程系的 L. A. Zadeh（扎德）教授于1965 年创立的模糊集合理论的数学基础上发展起来的，主要包括模糊集合理论、模糊逻辑、模糊推理和模糊控制等方面的内容。早在 20 世纪 20 年代，著名的哲学家和数学家 B. Russell 就写出了有关“含糊性”的论文。他认为所有的自然语言均是模糊的，例如“红的”和“老的”等概念没有明确的内涵和外延，因而是不明确的和模糊的。可是，在特定的环境中，人们用这些概念来描述某个具体对象时却又能心领神会，很少引起误解和歧义。美国加州大学的 L. A. Zadeh 教授在 1965 年发表了著名的论文，文中首次提出表达事物模糊性的重要概念——隶属函数，从而突破了 19 世纪末康托尔的经典集合理论，奠定了模糊理论的基础。

1966 年，P. N. Marinos 发表模糊逻辑的研究报告，1974 年，L. A. Zadeh 发表模糊推理的研究报告，从此，模糊理论成了一个热门的课题。

1974 年，英国的 E. H. Mamdani 首次用模糊逻辑和模糊推理实现了世界上第一个实验性的蒸汽机控制，并取得了比传统的直接数字控制算法更好的效果，从而宣告模糊控制的诞生。1980 年，丹麦的 L. P. Holmblad 和 Ostergard 在水泥窑炉采用模糊控制并取得了成功，这是第一个商业化的有实际意义的模糊控制器。

3. 模糊理论的应用领域

事实上，模糊理论应用最有效、最广泛的领域就是模糊控制，模糊控制在各种领域出人意料地解决了传统控制理论无法解决的或难以解决的问题，并取得了一些令人信服的成效。

4. 模糊控制的基本思想

把人类专家对特定的被控对象或过程的控制策略总结成一系列以“IF（条件）THEN（作用）”形式表示的控制规则，通过模糊推理得到控制作用集，作用于被控对象或过程。控制作用集为一组条件语句，状态语句和控制作用均为一组被量化了的模糊语言集，如“正大”“负大”“正小”“负小”“零”等。

模糊控制的几个研究方向：

（1）模糊控制的稳定性研究。

（2）模糊模型及辨识。

（3）模糊最优控制。

（4）模糊自组织控制。

（5）模糊自适应控制。

（6）多模态模糊控制。

模糊控制的主要缺陷：信息简单的模糊处理将导致系统的控制精度降低和动态品质变差。若要提高精度，则必然增加量化级数，从而导致规则搜索范围扩大，降低决策速度，甚至不能实时控制。模糊控制的设计尚缺乏系统性，无法定义控制目标。控制规则的选择、论域的选择、模糊集的定义、量化因子的选取多采用试凑法，这对复杂系统的控制是难以奏效的。

5. 模糊理论的研究领域

模糊理论是指用到了模糊集合的基本概念或连续隶属度函数的理论。根据图 3.19，可对模糊理论进行大致的分类，主要有 5 个分支。

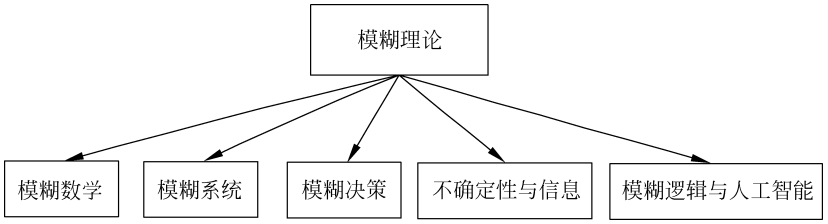


图 3.19　模糊理论的主要研究领域

（1）模糊数学：用模糊集合取代经典集合，从而扩展了经典数学中的概念。

（2）模糊系统：包含信号处理和通信中的模糊控制和模糊方法。

（3）模糊决策：用软约束来考虑优化问题。

（4）不确定性与信息：用于分析各种不确定性。

（5）模糊逻辑与人工智能：引入了经典逻辑学中的近似推理，且在模糊信息和近似推理的基础上开发了专家系统。

利用模糊集合理论，对实际问题进行模糊判断、模糊决策、模糊模式识别、模糊簇聚分析。系统的复杂性越高，精确能力就越低，模糊性就越强。这是 Zadeh 总结出的互克性原理。

3.4.10　可视化技术

可视化技术拓宽了传统的图表功能，使用户对数据的剖析更清楚。另外，还有归纳逻辑程序设计（Inductive Logic Programming，ILP）、Bayesian 网络等方法。

可视化技术的目标是帮助人们增强认知能力。基于计算机的可视化技术不仅把计

算机作为信息集成处理的工具，用计算机图形和其他技术来考虑更多的样本、变量和联系，更多的是作为跟用户之间的一种交流媒介，在认知激励和用户认知之间建立起一个反馈环。

1. 传统的可视化方法

传统的可视化方法多用于维数较小的数据，包括条形统计图、柱状图、折线图、饼图、锯齿图、分位数图、散点图、局部回归曲线图、时序图、核曲线、盒图、颜色编码、数据立方体等。其中，数据立方体(Data Cube)是将数据按多个维度组织形成的一种多维结构。用数据立方体描述多维数据时，用户可以多维形式组织数据，通过切片、切块、旋转、钻取等各种分析动作分析数据，使用户能从多个角度、多侧面地观察数据库中的数据，从而深入了解包含在数据中的信息和内涵。

2. 新兴的可视化技术

1) 基于几何投影技术的可视化方法

基于几何投影技术的可视化方法的目标是发现多维数据集的令人感兴趣的投影，从而将对多维数据的分析转换为仅对感兴趣的少量维度数据的分析，具体方法包括散点矩阵技术、格架(Trellis)图、测量图、平行坐标可视化技术和放射性可视化技术等。

散点矩阵把标准 2D 散点扩展到高维的标准方式。通过散点矩阵可以观察到维度间所有可能的双向交互作用和相关性。

格架图以多个二元图为基础。它针对一对要显示的特定变量，以其他一个或多个变量为条件画出一系列子图，子图中可以用其他任何类型的图形。测量图是在线图中扩展 n 维数据点的一种简单技术。样本的每维都在独立的轴上显示，轴上每个维度值都是关于轴中心对称的线段。平行坐标可视化技术是用一根平行于某显示轴的 k 根等距轴把 k 维空间映射成两个显示维度。轴对应于维度，并且与相应维度的最大值和最小值成线性比例。每个数据项都用一条折线来表示，折线和每根轴交点对应于尺度。放射性可视化表示的是对数据的非线性转换。这种转换保持某些对称性。该方法强调的是维度值之间的关系，而不是分开的、绝对值之间的关系。主分量分析是将数据向新的变量转换，将多元数据投影到数据可以最大限度分布的平面上。这使得可以在牺牲最少信息的条件下使分析数据可视化。

2) 基于图像技术的可视化方法

该方法把每个多维数据项映射为一个图像，如线条图、图标等。线条图把两个维度映射到显示维度中，剩下的维度映射为线条图像的角度或分量长度。这种技术限制了可视化的维度的数目。图标(Icon)是一些很小的图，其不同特征的大小是由特定变量的值决定的。例如在 Chernoff 面容图中，卡通画面部特征的尺寸(鼻子的长度、笑的程度、眼睛的形状等)代表了各个变量的值。这种方法所依据的原则是，人类的眼睛特

别善于识别和区分面容。这种图标方法有趣但很少用于严肃的数据分析。通常，图标显示只适用于少数实例的情况。

3）面向像素的可视化方法

面向像素的可视化方法把每个数据值映射到有色像素中，并在分开的窗口中表示属于每个属性的数据值。其优点是一次性可以描述大量信息并且不会产生重叠，不仅能够有效地保留用户感兴趣的小部分区域，还能纵览全局数据。这种技术适用于大容量数据（达到百万级数据值）的可视化。如果一个像素点代表一个数据值，主要的问题就是怎样在屏幕上排列这些像素，不同例图使用不同的排列策略。

3. 基于分层技术的可视化方法

分层技术对 k 维空间进行再分，并以分层的方式来表示子空间。维度层积就是一种分层技术可视化方法。每个维都离散化为少量的箱，陈列区域分裂成一个子图像栅格。子图像的数目要依据用户指定的两个“外部”维度相关联的箱的数目来确定。子图像根据两个更多维度的箱数被进一步分解。分解过程递归持续，直到所有的维都被指定完毕。此外，基于分层技术的可视化方法还有 Robertson、Mackinlay 和 Card 等提出的利用三维图形技术对层次结构进行可视化的方法，Shneiderman 等提出的利用屏幕空间的层次信息表示模型 Tree-Map、Lamping 和 Rao 等提出的基于双曲线几何的可视化等。

4. 可视化技术的新进展

1）多种可视化技术的组合应用

近年来，在可视化数据挖掘应用中涌现出一批新的可视化技术，它们综合了多种可视化方法，如 Parabox、数据星座、数据表单、时刻表、多景观（Multi-Landscape）等。Parabox 组合了盒图、平行坐标和起泡图。它既能处理连续数据，也能处理分类数据。合成盒图、平行坐标和起泡图的原因在于它们各自的能力不同。盒图适用于显示分布概括。平行坐标主要用于显示高维度异常点和带有异常值的样本。起泡图用于分类数据，泡中圆的大小表示样本数目和各自值的情况。按照一系列的平行轴来组建维度，就如带有平行坐标图一样。在泡和盒子之间画线，将每个样本的维度连接起来。这些技术的组合产生了一个可视化部件，它优于用单一方法建立起的可视化的表述。数据星座是可视化有几千个节点和链接的大型曲线图中的一个组成部分。用两个表确定数据星座的参数，一个对应节点，另一个对应链接。不同的布局算法动态决定节点的位置，使得模式显现出来（异常点、类等的可视化解释）。数据表单用于动态的可滚动文本的可视化，在文本和图像之间建立起桥梁。用户可以调整放大比例，逐步显示越来越小的字体，最后转到单像素表述。时刻表是一种显示数千个时间标记事件的技术。多景观是用 3D 来对 2D 景观信息进行编码的景观可视化方法。

2）失真技术

失真(Distortion)技术以高细节级别显示一部分数据，而另一些数据则以低且多的细节级别显示。失真技术在保持对数据有一个纵览的同时提供一种集中的方式，在交互式探测过程中是有帮助的。典型的失真技术有 Fish-Eye 视图、压缩失真技术 Hyperbolic Browsers 等。

3）交互技术

交互(Interaction)技术允许可视化依照探测对象发生动态的变化，而且也使得联系并组合多样的、独立的可视化成为可能，如交互式映射、投影、过滤、缩放、交互式链接和刷洗。通过链接技术把多种可视化连接起来，用户可以比较多个模型，充分利用不同可视化方法、不同模型描述方式的优点。当强调一个模型中的某一部分时，相关联的不同模型描述方式会同时、自动地显示在多个独立的窗口中。综合使用这类技术比独立考虑这些可视化组件要获得更多的信息。

4）钻过技术

钻过(Drill-Through)技术是指当选取模型中的某一部分时，可以知道这一部分模型是根据哪些原始数据提出来的，并且可以访问它们。例如，决策树可视化方法允许对决策树的分支进行选择和钻过，从而使用户可以访问与构造该分支有关的数据，而忽略其他数据的描述。

5）虚拟技术

虚拟技术可以将模型结果输出到虚拟设备或虚拟的可视化环境中，使用户置身其中。用户通过导航技术寻找自己感兴趣的信息，从而获得更为直观的数据理解和分析。从这种技术层面解决数据挖掘任务能够结合人的认知能力，使人充分融入数据挖掘的过程中。目前已经提出的虚拟技术有头盔(Head-Mounted)显示、虚拟数据立方体等。

5. 现有数据挖掘工具中的可视化技术应用概况

数据挖掘工具在很大程度上决定了是否能够实现数据可视化、挖掘模型可视化、挖掘过程可视化以及可视化程度、质量和交互灵活性，也严重影响数据挖掘系统的使用和解释能力。当前主流的数据挖掘工具，如 SAS Enterprise Miner、IBM Intelligent Miner、Teradata Warehouse Miner、SPSS Clementine 等，都能够提供常用的挖掘过程和挖掘模式，但是可视化技术的应用仍然很有限，方法也比较单一，而且主要集中在初始视图可视化、结果(模型)可视化，分析过程对大部分用户来说仍属于黑箱操作。可视化和分析式数据挖掘之间松散的联系代表了目前可视化数据挖掘技术的绝大多数情况。因此，尽管在数据挖掘的可视化方法研究中不断有新的技术提出，但在数据挖掘工具中的应用仍然不够深入和广泛。

3.5 空间数据库的数据挖掘

近年来，数据挖掘研究多针对关系数据库，但是空间数据库系统的发展为我们提供了丰富的空间数据，为数据分析和知识发现展示了广阔的前景。空间数据挖掘技术帮助人们从庞大的空间数据中抽取有用的信息。由于空间数据的数量庞大及空间问题的特殊性，因此发现隐含在空间数据中的特征和模式，已成为空间数据库的一个重要问题。现已在全球定位系统、图像数据库等领域得到了广泛应用。

本节介绍空间数据挖掘的方法。

3.5.1 归纳方法

基于归纳方法的空间数据挖掘算法必须由用户预先给定或系统自动生成概念层次树，发现的知识依赖于层次树结构，计算复杂性为 $O(\log N)$，N 为空间数据个数。

1. 简介

数学归纳法是一种数学证明方法，通常用于证明某个给定命题在整个（或者局部）自然数范围内成立。除了自然数以外，广义上的数学归纳法也可以用于证明一般良基结构，例如集合论中的树。这种广义的数学归纳法应用于数学逻辑和计算机科学领域，称作结构归纳法。

在数论中，数学归纳法是以一种不同的方式来证明任意一个给定的情形都是正确的（第一个，第二个，第三个，一直下去概不例外）的数学定理。

虽然数学归纳法名字中有“归纳”，但是数学归纳法并非不严谨的归纳推理法，它属于完全严谨的演绎推理法。事实上，所有数学证明都是演绎法。

2. 原理

最简单和常见的数学归纳法是证明当 n 等于任意一个自然数时某命题成立。证明分下面两步：

（1）证明当 $n=1$ 时命题成立。

（2）假设 $n=m$ 时命题成立，那么可以推导出在 $n=m+1$ 时命题也成立（m 代表任意自然数）。

这种方法的原理在于：首先证明在某个起点值时命题成立，然后证明从一个值到下一个值的过程有效。如果这两点都已经证明，那么任意值都可以通过反复使用这个方法推导出来。把这个方法想成多米诺效应也许更容易理解一些。例如，你有一列很长的直立着的多米诺骨牌，如果你可以：

- 证明第一张骨牌会倒。
- 证明只要任意一张骨牌倒了，那么与其相邻的下一张骨牌也会倒。

那么便可以下结论：所有的骨牌都会倒下。

3. 合理性

数学归纳法的原理，通常被规定作为自然数公理(参见皮亚诺公理)。但是在另一些公理的基础上，它可以用一些逻辑方法证明。数学归纳法原理可以由下面的良序性质(最小自然数原理)推出：

自然数集是良序的(每个非空的正整数集合都有一个最小的元素)。

例如{1,2,3,4,5}这个正整数集合中有最小的数——1。

下面将通过这个性质来证明数学归纳法。

对于一个已经完成上述两步证明的数学命题，我们假设它并不是对于所有的正整数都成立。

对于那些不成立的数所构成的集合 S，其中必定有一个最小的元素 k(1 是不属于集合 S 的，所以 $k>1$)。

k 已经是集合 S 中的最小元素了，所以 $k-1$ 不属于 S，这意味着 $k-1$ 对于命题而言是成立的——既然对于 $k-1$ 成立，那么对 k 也应该成立，这与我们完成的第二步证明矛盾。所以这个完成两个步骤的命题能够对所有 n 都成立。

注意到有些公理确实是数学归纳法原理可选的公理化形式。更确切地说，两者是等价的。

4. 发展历程

已知最早使用数学归纳法的证明出现于 Francesco Maurolico 的 *Arithmeticorum libri duo*(1575 年)。Maurolico 利用递推关系巧妙地证明出前 n 个奇数的总和是 n^2，由此总结出了数学归纳法。

最简单和常见的数学归纳法证明方法是证明当 n 属于所有正整数时一个表达式成立，这种方法由下面两步组成。

(1) 递推的基础：证明当 $n=1$ 时表达式成立。

(2) 递推的依据：证明如果当 $n=m$ 时表达式成立，那么当 $n=m+1$ 时同样成立。

这种方法的原理在于第一步证明起始值在表达式中是成立的，然后证明一个值到下一个值的证明过程是有效的。如果这两步都被证明了，那么任何一个值的证明都可以被包含在重复不断进行的过程中。

3.5.2　聚集方法

基于聚集(Clustering)方法的空间数据挖掘算法包括 CLARANS、BIRCH、DBSCAN 等

算法。

聚集是把相似的记录放在一起。其作用是让用户在较高的层次上观察数据库，常被用来做商业上的顾客分片(Segmentation)，找到不能与其他记录集合在一起的记录，做例外分析。

和分类一样，聚类的目的也是把所有的对象分成不同的群组，但和分类算法的最大不同在于采用聚类算法划分之前并不知道要把数据分成几组，也不知道依赖哪些变量来划分。

聚类有时也称分段，是指将具有相同特征的人归结为一组，将特征平均，以形成一个“特征矢量”或“矢心”。聚类系统通常能够把相似的对象通过静态分类的方法分成不同的组别或者更多的子集(Subset)，这样在同一个子集中的成员对象都有相似的一些属性。聚类被一些提供商用来直接提供不同访客群组或者客户群组特征的报告。聚类算法是数据挖掘的核心技术之一，除了本身的算法应用之外，聚类分析也可以作为数据挖掘算法中其他分析算法的一个预处理步骤。

1. CLARANS 算法

CLARANS(a Clustering Algorithm based on Randomized Search，基于随机选择的聚类算法)将采样技术(CLARA)和 PAM 算法结合起来。CLARA 的主要思想是：不考虑整个数据集合，而是选择实际数据的一小部分作为数据的代表。然后用 PAM 算法从样本中选择中心点。如果样本是以非常随机的方式选取的，那么它应当接近代表原来的数据集。从中选出的代表对象(中心点)很可能和从整个数据集合中选出的代表对象相似。CLARA 抽取数据集合的多个样本，对每个样本应用 PAM 算法，并返回最好的聚类结果作为输出。

CLARA 的有效性主要取决于样本的大小。如果任何一个最佳抽样中心点不在最佳的 K 个中心之中，则 CLARA 将永远不能找到数据集合的最佳聚类。同时这也是为了聚类效率所付出的代价。

CLARANS 则是将 CLARA 和 PAM 有效地结合起来，CLARANS 在任何时候都不把自身局限于任何样本，CLARANS 在搜索的每一步都以某种随机性选取样本。算法步骤如下(算法步骤摘自百度文库)：

(1) 输入参数 numlocal 和 maxneighbor。numlocal 表示抽样的次数，maxneighbor 表示一个节点可以与任意特定邻居进行比较的数目。令 $i=1$，i 用来表示已经选样的次数。mincost 为最小代价，初始时设为大数。

(2) 设置当前节点 current 为 Gn 中的任意一个节点。

(3) 令 $j=1$(j 用来表示已经与 current 进行比较的邻居的个数)。

(4) 考虑当前点的一个随机的邻居 S，并计算两个节点的代价差。

(5) 如果 S 的代价较低，则 current：$=S$，转到步骤(3)。

(6) 否则，令 $j=j+1$。如果 $j<=$maxneighbor，则转到步骤(4)。

(7) 否则，当 $j>$maxneighbor 时，当前节点为本次选样的最小代价节点。如果其代价小于 mincost，令 mincost 为当前节点的代价，则 bestnode 为当前的节点。

(8) 令 $i=i+1$，如果 $i>$numlocal，则输出 bestnode，运算中止；否则，转到步骤(2)。

对上面出现的一些概念进行说明：

(1) 代价值，主要描述一个对象被分到一个类别中的代价值，该代价值由每个对象与其簇中心点间的相异度(距离或者相似度)的总和来定义。代价差则是两次随机领域的代价差值。

(2) 更新邻节点，CLARANS 不会把搜索限制在局部区域，如果发现一个更好的近邻，CLARANS 就移到该近邻节点，处理过程重新开始；否则，当前的聚类产生一个局部最小解。如果找到一个局部最小解，CLARANS 从随机选择的新节点开始，搜索新的局部最小解。当搜索的局部最小解达到用户指定的数目时，最好的局部最小解作为算法的输出。从上面的算法步骤也可以看出这一思想。在第(5)步中更新节点 current。

2. BIRCH 算法

BIRCH(Balanced Iterative Reducing and Clustering using Hierarchies，利用层次方法的平衡迭代规约和聚类)算法是 1996 年由 Tian Zhang 提出来的。BIRCH 算法就是通过聚类特征(Clustering Feature，CF)形成一个聚类特征树，root 层的 CF 个数就是聚类个数。

整个算法实现共分为 4 个阶段：

(1) 扫描所有数据，建立初始化的 CF 树，把稠密数据分成簇，稀疏数据作为孤立点对待。

(2) 这个阶段是可选的，阶段 3 的全局或半全局聚类算法有着输入范围的要求，以达到速度与质量的要求，所以此阶段在阶段 1 的基础上，建立一个更小的 CF 树。

(3) 补救由于输入顺序和页面大小带来的分裂，使用全局/半全局算法对全部叶节点进行聚类。

(4) 这个阶段也是可选的，把阶段 3 的中心点作为种子，将数据点重新分配到最近的种子上，保证重复数据分到同一个簇中，同时添加簇标签。

该算法的缺点：由于使用半径和直径概念，适用于球形数据的聚类(可以在聚类前进行样本绘图观察后选择该算法)。

聚类特征(CF)：每一个 CF 都是一个三元组，可以用(N，LS，SS)表示。其中 N 代表这个 CF 中拥有的样本点的数量，LS 代表这个 CF 中拥有的样本点各特征维度的和向量，SS 代表这个 CF 中拥有的样本点各特征维度的平方和。

3. DBSCAN 算法

DBSCAN(Density-Based Spatial Clustering of Application with Noise)算法是一种

典型的基于密度的聚类方法。它将簇定义为密度相连的点的最大集合，能够把具有足够密度的区域划分为簇，并可以在有噪声的空间数据集中发现任意形状的簇。

DBSCAN 算法中有两个重要参数：Eps 和 MmPts。Eps 是定义密度时的邻域半径，MmPts 为定义核心点时的阈值。

在 DBSCAN 算法中将数据点分为以下 3 类。

（1）核心点：如果一个对象在其半径 Eps 内含有超过 MmPts 数目的点，则该对象为核心点。

（2）边界点：如果一个对象在其半径 Eps 内含有点的数量小于 MmPts，但是该对象落在核心点的邻域内，则该对象为边界点。

（3）噪声点：如果一个对象既不是核心点又不是边界点，则该对象为噪声点。

通俗地讲，核心点对应稠密区域内部的点，边界点对应稠密区域边缘的点，而噪声点对应稀疏区域中的点。

DBSCAN 算法对簇的定义很简单，由密度可达关系导出的最大密度相连的样本集合，即为最终聚类的一个簇。

DBSCAN 算法的簇中可以有一个或者多个核心点。如果只有一个核心点，则簇中其他的非核心点样本都在这个核心点的 Eps 邻域中。如果有多个核心点，则簇中的任意一个核心点的 Eps 邻域中一定有一个其他的核心点，否则这两个核心点无法密度可达。这些核心点的 Eps 邻域中所有的样本的集合组成一个 DBSCAN 聚类簇。

DBSCAN 算法的描述如下。

输入：数据集，邻域半径为 Eps，邻域中数据对象的数目阈值为 MmPts。

输出：密度联通簇。

处理流程如下。

（1）从数据集中任意选取一个数据对象点 p。

（2）如果对于参数 Eps 和 MmPts，所选取的数据对象点 p 为核心点，则找出所有从 p 密度可达的数据对象点，形成一个簇。

（3）如果选取的数据对象点 p 是边缘点，则选取另一个数据对象点。

（4）重复(2)、(3)步，直到所有点被处理。

DBSCAN 算法的计算复杂度为 $O(n^2)$，n 为数据对象的数目。这种算法对于输入参数 Eps 和 MmPts 是敏感的。

和传统的 K-Means 算法相比，DBSCAN 算法不需要输入簇数 k，而且可以发现任意形状的聚类簇，同时在聚类时可以找出异常点。

DBSCAN 算法的主要优点如下。

（1）可以对任意形状的稠密数据集进行聚类，而 K-Means 之类的聚类算法一般只适用于凸数据集。

(2) 可以在聚类的同时发现异常点,对数据集中的异常点不敏感。

(3) 聚类结果没有偏倚,而 K-Means 之类的聚类算法的初始值对聚类结果有很大影响。

DBSCAN 算法的主要缺点如下。

(1) 样本集的密度不均匀、聚类间距相差很大时,聚类质量较差,这时用 DBSCAN 算法一般不适合。

(2) 样本集较大时,聚类收敛时间较长,此时可以对搜索最近邻时建立的 KD 树或者球树进行规模限制来进行改进。

(3) 调试参数比较复杂时,主要需要对距离阈值 Eps 和邻域样本数阈值 MmPts 进行联合调参,不同的参数组合对最后的聚类效果有较大影响。

(4) 对于整个数据集只采用了一组参数。如果数据集中存在不同密度的簇或者嵌套簇,则 DBSCAN 算法不能处理。为了解决这个问题,有人提出了 OPTICS 算法。

(5) DBSCAN 算法可过滤噪声点,这同时也是其缺点,这造成了它不适用于某些领域,如对网络安全领域中恶意攻击的判断。

3.5.3　统计信息网格算法

统计信息网格(Statistical Information Grid,STING)算法是一个基于网格的多分辨率聚类技术,它将空间区域划分为矩形单元。针对不同级别的分辨率,通常存在多个级别的矩形单元,这些单元形成了一个层次结构:高层的每个单元被划分为多个低一层的单元。关于每个网格单元属性的统计信息(例如平均值、最大值和最小值)被预先计算和存储。这些统计变量可以方便下面描述的查询处理使用。高层单元的统计变量可以很容易地从低层单元的变量计算得到。这些统计变量包括:属性无关的变量 count,属性相关的变量 m(平均值)、s(标准偏差)、min(最小值)、max(最大值),以及该单元中属性值遵循的分布类型 distribution,例如正态的、均衡的、指数的或无(如果分布未知)。当数据被装载进数据库时,最底层单元的变量 count、m、s、min 和 max 直接进行计算。如果分布的类型事先知道,distribution 的值可以由用户指定,也可以通过假设检验来获得。一个高层单元的分布类型可以基于它对应的低层单元多数的分布类型,用一个阈值过滤过程来计算。如果低层单元的分布彼此不同,阈值检验失败,则高层单元的分布类型被置为 none。

统计变量的使用可以以自顶向下的基于网格的方法。首先,在层次结构中选定一层作为查询处理的开始点。通常,该层包含少量的单元。对当前层次的每个单元计算置信度区间(或者估算其概率),用以反映该单元与给定查询的关联程度。不相关的单元就不再考虑。低一层的处理就只检查剩余的相关单元。这个处理过程反复进行,直到达到最

底层。此时，如果查询要求被满足，那么返回相关单元的区域；否则，检索和进一步处理落在相关单元中的数据，直到它们满足查询要求。

1. 算法流程

(1) 针对不同的分辨率，通常有多个级别的矩形单元。

(2) 这些单元形成了一个层次结构，高层的每个单元被划分成多个低一层的单元。

(3) 关于每个网格单元属性的统计信息(例如平均值、最大值、最小值)被预先计算和存储，这些统计信息用于回答查询(统计信息是进行查询使用的)。

2. 查询流程

(1) 从一个层次开始。

(2) 对于这一个层次的每个单元格，计算查询相关的属性值。

(3) 在计算的属性值以及约束条件下，将每一个单元格标记成相关或者不相关(不相关的单元格不再考虑，下一个较低层的处理就只检查剩余的相关单元)。

(4) 如果这一层是底层，那么转到步骤(6)，否则转到步骤(5)。

(5) 由层次结构转到下一层，依照步骤(2)进行。

(6) 查询结果得到满足，转到步骤(8)，否则转到步骤(7)。

(7) 恢复数据到相关的单元格，进一步处理以得到满意的结果，转到步骤(8)。

(8) 停止。

3. 核心思想

根据属性的相关统计信息划分网格，而且网格是分层次的，下一层是上一层的继续划分。在一个网格内的数据点即为一个簇。

4. 性质

如果粒度趋向于0(朝向非常底层的数据)，则聚类结果趋向于 DBSCAN 聚类结果，即使用计数 count 和大小信息，使用 STING 算法可以近似地识别稠密的簇。

STING 算法有以下几个优点：

(1) 基于网格的计算是独立于查询的，因为存储在每个单元的统计信息提供了单元中的数据汇总信息，不依赖于查询。

(2) 网格结构有利于增量更新和并行处理。

(3) 效率高。STING 算法扫描数据库一次开始计算单元的统计信息，因此产生聚类的时间复杂度为 $O(n)$，在层次结构建立之后，查询处理时间为 $O(g)$，其中 g 为最底层网格单元的数目，通常远远小于 n。

STING 算法的缺点如下：

(1) 由于 STING 算法采用了一种多分辨率的方法来进行聚类分析，因此 STING 算法的聚类质量取决于网格结构的最底层的粒度。如果最底层的粒度很细，则处理的代价

会显著增加。然而如果粒度太粗，聚类质量难以得到保证。

(2) STING 算法在构建一个父亲单元时没有考虑到子女单元和其他相邻单元之间的联系。所有的簇边界不是水平的，就是竖直的，没有斜的分界线，降低了聚类质量。

该算法是一个查询无关算法，每个节点存储数据的统计信息，可以处理大量的查询。该算法采用增量修改，避免数据更新造成的所有单元重新计算，而且易于并行化。

3.5.4　空间聚集和特征邻近关系挖掘

1. 发现集合邻近关系

给定一个点的聚集，找到聚集的 K 个最邻近特征。CRH 算法寻找集合邻近关系，它是 Circle、Isothetic Rectangle 和 Convex Hull 的首字母的缩写形式。CRH 算法用筛选器逐步减少特征个数，直至找到 K 个最接近的特征。在 SPARC-10 工作站上的实验结果表明，CRH 作为一种近似算法，得出的结果相当精确，它能在约 1 秒 CPU 时间内从 5000 个特征中找到最近的 25 个。

2. 发现集合邻近的共性

给定 N 个聚集，找到与全部或大多数聚集最接近的公共特征类，即出现在同一分类中的相似特征，例如发现所有居民区都与中学相近，而不一定是同一所中学。Gencom 算法从 N 个聚集的 N 个最近的 K 个特征的集合中抽取集合邻近公共特征。

3. 空间数据采掘系统原型 GeoMiner

加拿大 Simon Fraser 大学开发出了一个空间数据采掘系统原型 GeoMiner。该系统在空间数据库建模中使用 SAND 体系结构，它包含三大模块：空间数据立方体构建模块、空间联机分析处理(OLAP)模块和空间数据采掘模块，采用的空间数据采掘语言是 GMQL。目前已能采掘 3 种类型的规则：特征规则、判别规则和关联规则。GeoMiner 的体系结构如图 3.20 所示，包含 4 部分。

(1) 图形用户界面，用于进行交互式的采掘并显示采掘结果。

(2) 发现模块集合，含有上述 3 个已实现的知识发现模块以及 4 个计划实现的模块(分别以实线框和虚线框表示)。

(3) 空间数据库服务器，包括 MapInfo、ESRI/Oracle SDE、Informix-Illustra 以及其他空间数据库引擎。

(4) 存储非空间数据、空间数据和概念层次的数据库和知识库。

4. 空间数据采掘的未来方向

空间数据采掘是一个非常年轻而富有前景的领域，有很多研究问题需要深入探讨，这也是该领域的未来方向。

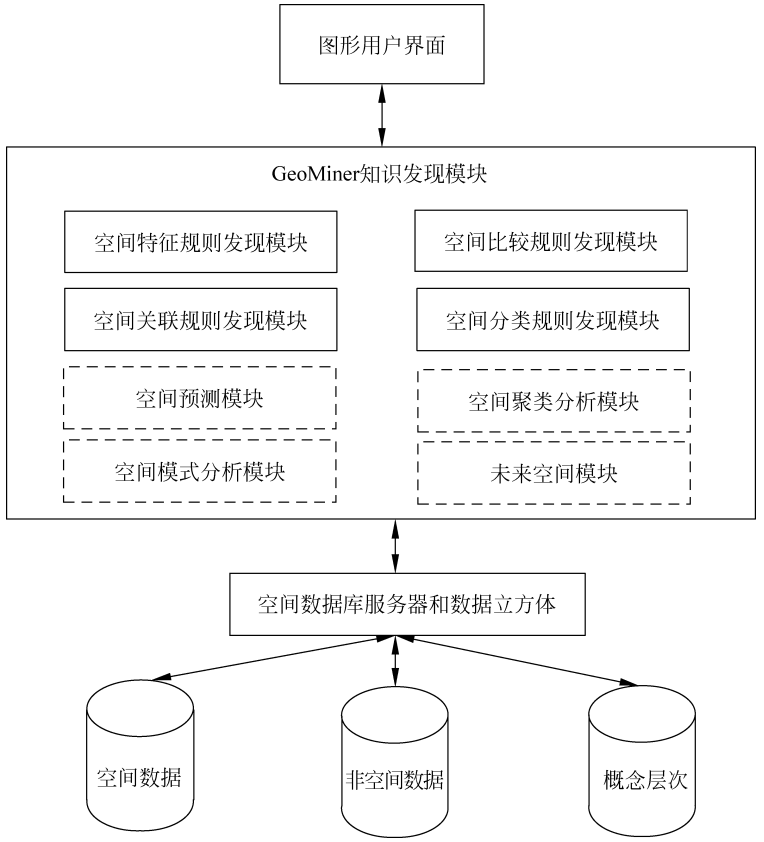


图 3.20　空间数据采掘系统原型 GeoMiner

（1）在面向对象（Object-Oriented，OO）的空间数据库中进行数据采掘。

目前在实际中应用的空间数据采掘方法都假定空间数据库中采用的是扩展的关系模型，而关系数据库并不能很好地处理空间数据。许多研究者指出，OO 模型比传统的关系模型或扩展关系模型更适合处理空间数据，如矩形、多边形和复杂的空间对象可在 OO 数据库中很自然地建模。因此，可以考虑建立面向对象的空间数据库以进行数据采掘。需要解决的问题是如何使用 OO 方法设计空间数据库，以及怎样从数据库中采掘知识。目前 OO 数据库技术正在走向成熟，在空间数据采掘中开发 OO 技术是一个具有极大潜力的领域。

（2）进行不确定性采掘。

证据推理方法可用于图像数据库的采掘，以及其他经过不确定性建模的数据库的分析。Bell 等证明，证据理论比传统的概率模型，如贝叶斯等方法进行不确定性建模的效果要好。另外，还可考虑通过利用统计学、模糊逻辑和粗糙集方法以处理不确定性和不完整的信息，该领域尚有待拓展。

(3) 多边形聚类技术。

目前空间聚类问题的解决方案尚局限在对点对象的聚类,该问题的未来方向是处理可能重叠的对象的聚类,如多边形聚类。

(4) 多维规则可视化。

理解所发现规则的最有效的方式是进行图形可视化。多维数据可视化已有相应的文献研究,而多维规则可视化仍是一个不成熟的领域,可考虑采用计算机图形学中的一些可视化技术。

(5) 基于泛化的空间数据采掘机制需要进一步地开拓,以处理多专题地图和多层次的交互式采掘,并与空间索引、空间存取方法和数据仓库技术有效结合。

(6) 空间数据分类领域尚需找到真正高效的空间分类方法,以处理带有不完整信息的问题。

(7) 基于模式或基于相似性的采掘以及元规则指导的空间数据采掘尚需探讨。

(8) 空间数据采掘查询语言 SDMQL 需要进行详细设计和标准化。

(9) 大量的遥感图像要求更多的数据采掘方法,用以检测异常、查找相似的图片以及发现不同现象间的关系。

3.6　数据挖掘工具

目前,国外有许多研究机构、公司和学术组织从事数据挖掘工具的研制和开发。这些工具主要采用基于人工智能的技术,包括决策树、规则归纳、神经元网络、可视化、模糊建模、簇聚等,另外也采用了传统的统计方法。这些数据挖掘工具差别很大,不仅体现在关键技术上,还体现在运行平台、数据存取、价格等方面。

数据挖掘工具可根据应用领域分为以下 3 类。

(1) 通用单任务类:仅支持 KDD 的数据挖掘步骤,并且需要大量的预处理和善后处理工作。主要采用决策树、神经网络、基于例子和规则的方法,发现任务大多属于分类范畴。

(2) 通用多任务类:可执行多个领域的知识发现任务,集成了分类、可视化、聚集、概括等多种策略,如 Clementine、IBMIntelligentMiner、SGIMineset。

(3) 专用领域类:现有的许多数据挖掘系统是专为特定目的开发的,用于专用领域的知识发现,对挖掘的数据库有语义要求,发现的知识也较单一。例如 Explora 用于超市销售分析,仅能处理特定形式的数据,知识发现也以关联规则和趋势分析为主。另外发现方法单一,有些系统虽然能发现多种形式的知识,但基本上以机器学习、统计分析为主,计算量大。

根据所采用的技术,挖掘工具大致分为以下 6 类。

（1）基于规则和决策树的工具：大部分数据挖掘工具采用规则发现和决策树分类技术来发现数据模式和规则，其核心是某种归纳算法，如 ID3 和 C4.5 算法。它通常先对数据库中的数据进行挖掘，生成规则和决策树，然后对新数据进行分析和预测，典型的产品有 AngossSoftware 开发的 KnowlegeSeeker 和 ATTARSoftware 开发的 XpertRuleProfiler。

（2）基于神经元网络的工具：基于神经元网络的工具由于具有对非线性数据的快速建模能力，因此越来越流行。挖掘过程基本上是将数据簇聚，然后分类计算权值。它在市场数据库的分析和建模方面应用广泛，典型产品有 AdvancedSoftware 开发的 PBProfile。

（3）数据可视化方法：这类工具大大扩展了传统商业图形的能力，支持多维数据的可视化，同时提供了多方向同时进行数据分析的图形方法。

（4）模糊发现方法：采用模糊语言表达的模糊集计算出各条件的数值及关系，并使用规范化的方法和技术来完成一系列的计算，以获得实际的决策结果。

（5）统计方法：这些工具没有使用人工智能技术，因此更适合分析现有信息，而不是从原始数据中发现数据模式和规则。

（6）综合多方法：许多工具采用了多种挖掘方法，一般规模较大。

工具系统的总体发展趋势是，使数据挖掘技术进一步为用户所接受和使用，另一方面也可以理解成以使用者的语言表达知识概念。

下面简单介绍几种数据挖掘工具。

1. QUEST

QUEST 是 IBM 公司 Almaden 研究中心开发的一个多任务数据挖掘系统，目的是为新一代决策支持系统的应用开发提供高效的数据开采基本构件。QUEST 系统具有如下特点：

- 提供了专门在大型数据库上进行各种开采的功能，如关联规则发现、序列模式发现、时间序列聚类、决策树分类、递增式主动开采等。
- 各种开采算法具有近似线性计算的复杂度（$O(n)$），可适用于任意大小的数据库。
- 算法具有找全性，即能将所有满足指定类型的模式全部寻找出来。
- 为各种发现功能设计了相应的并行算法。

2. MineSet

MineSet 是由 SGI 公司和美国 Standford 大学联合开发的多任务数据挖掘系统。MineSet 集成了多种数据挖掘算法和可视化工具，可以帮助用户直观地、实时地发掘、理解大量数据背后的知识。

MineSet 2.6 具有如下特点：

- MineSet 以先进的可视化方法闻名于世。MineSet 2.6 中使用了 7 种可视化工具来

表现数据和知识。对同一个挖掘结果可以用不同的可视化工具以各种形式表示，用户也可以按照个人的喜好调整最终效果，以便更好地理解。MineSet 2.6 中的可视化工具有 SplatVisualize、ScatterVisualize、MapVisualize、TreeVisualize、RecordViewer、StatisticsVisualize、ClusterVisualizer，其中 RecordViewer 是二维表，StatisticsVisualize 是二维统计图，其余都是三维图形，用户可以任意放大、旋转、移动图形，从不同的角度观看。

- 提供多种数据挖掘模式，包括分类器、回归模式、关联规则、聚类分析、判断列重要度。
- 支持多种关系数据库。可以直接从 Oracle、Informix、Sybase 的表读取数据，也可以通过 SQL 命令执行查询。
- 多种数据转换功能。在进行挖掘前，MineSet 可以去除不必要的数据项，统计、集合、分组数据，转换数据类型，构造表达式由已有数据项生成新的数据项，对数据进行采样等。
- 操作简单。支持国际字符，可以直接发布到 Web。

3. DBMiner

DBMiner 是加拿大 Simon Fraser 大学开发的一个多任务数据挖掘系统，它的前身是 DBLearn。该系统设计的目的是把关系数据库和数据开采集成在一起，以面向属性的多级概念为基础发现各种知识。DBMiner 系统具有如下特色：

- 能完成多种知识的发现，如泛化规则、特性规则、关联规则、分类规则、演化知识、偏离知识等。
- 综合了多种数据开采技术，如面向属性的归纳、统计分析、逐级深化发现多级规则、元规则引导发现等方法。
- 提出了一种交互式的类 SQL 语言——数据开采查询语言 DMQL，能与关系数据库平滑集成，实现了基于客户/服务器体系结构的 UNIX 和 PC(Windows/NT)版本的系统。

4. SAS 数据挖掘工具

作为企业的中高层管理人员，需要首先了解商业事务中发生了什么，接下来要分析它为什么发生，最后决定做什么，即可以采取什么行动。信息查询、报表和多维分析技术主要集中处理发生了什么，但很少能够揭示发生的原因。然而正是这种隐含在表象下的内在原因、机制和规律才是对企业发展最有价值的知识。知识挖掘正是为实现这方面的需求而产生的。SAS 公司在数据挖掘领域的成就是有目共睹的，我们将数据挖掘定义为："知识挖掘是按照既定的业务目标，对大量的企业数据进行探索，揭示隐藏在其中的规律性并进一步将之模型化的先进、有效的方法"。

知识挖掘的主流技术手段是统计学的方法，包括数理统计方法、多元统计方法、计量经济学和时间序列分析方法等。此外，运筹学、人工神经元网络和专家系统技术的发展也为知识挖掘提供了新的思路。

知识挖掘的内涵不仅在于使用这些方法进行数据分析，它有自己的一套完整的方法论，SAS 公司提出了 SEMMA 方法论。我们在此之前已经介绍过了。

下面对我们提供的数据挖掘模块，如 SAS/STAT、SAS/ETS、SAS/INSIGHT、SAS/EM 进行进一步的讨论。

1) SAS/STAT

SAS/STAT 提供的统计分析功能包括：

(1) 方差分析。

(2) 一般线性模型(包括因素分析、方差分量模型、混合模型等)、回归分析、多变量分析(主成分分析、因子分析和典型相关分析等)。

(3) 判别分析。

(4) 聚类分析。

(5) 属性数据分析。

(6) 生存分析等多个功能。

SAS/STAT 可以适应各种不同模型和不同特点数据的需要。例如回归分析方面除了通用的回归方法外，还有正交回归、响应面回归、Logistic 回归、非线性回归等。

可处理的数据有实型数据、有序数据和属性数据，并能产生各种有用的统计量和诊断信息，且具有多种模型(变量)选择方法。

在方差分析方面，SAS/STAT 为多种试验设计模型提供了方差分析工具。它还有处理一般线性模型和广义线性模型的专用过程。在多变量统计分析方面，SAS/STAT 为主成分分析、典型相关分析、判别分析和因子分析提供了许多专用过程。SAS/STAT 还包含多种聚类准则的聚类分析方法。

SAS/STAT 包含的菜单系统 Market Research 提供了用于市场分析的各种工具，如关联分析、对应分析、多维标度分析、多维偏好分析等。分析结果用直观的图形和表格显示给用户，并根据用户对产品的评价提供新设计产品市场占有率的预测。

在 SAS 的桌面系统中有一个专用的统计分析系统——ANALYST。它集成了 SAS/STAT、SAS/QC 和 SAS/GRAPH 的许多功能。按常用统计分析的任务组织菜单，在统计分析方面包括描述统计量计算、列联表分析、假设检验、方差分析、回归分析等。在图形显示方面除了配合各种分析提供图形显示外，还可用菜单设定制作三维散点图、三维曲面图和等高线图等。在 ANALYST 子系统中，对各种检验法的功效函数计算和

样本容量确定提供很方便的用法。

2) SAS/ETS

SAS/ETS为SAS提供具有丰富的计量经济学和时间序列分析方法的产品，包含方便的各种模型设定手段，多样的参数估计方法，是研究复杂系统和进行预测的有力工具。

ETS表示Econometric & Time Series，提供了用于进行预测、规划及商业模型(建模)的分析工具。与SAS其他产品类似，由若干过程组成，提供了目前所有实用的用于预测的数学模型。ETS主要应用在时间序列分析、线性和非线性系统模拟、贷款偿还(Amortization)、折旧(Depreciation)等领域，即ETS分析、模拟、预测、季节性调整、商业分析、报表以及时间序列数据的访问和操纵。ETS分析的目的是帮助人们进行正确决策。

SAS的时间序列预测系统利用序列历史值的趋势和格局或序列的其他变量进行外插来预测将来值。该系统提供了用SAS/ETS软件实现的方便灵活的窗口菜单驱动环境进行时间序列分析和预测的工具。

用户可按完全自动的方式使用系统，也可交互式地使用系统诊断功能及时间序列建模工具生成能最佳预测用户时间序列的预测模型。对于每个序列系统提供图形和统计参数，帮助用户选择最佳预测方法。

下面是SAS系统时间序列预测分析的主要功能和特点：

(1) 使用广泛的预测方法，包括指数平滑模型、Winter方法以及ARIMA(Box-Jenkins)模型。用户也可以通过组合模型产生新的预测模型或方法。

(2) 在预测模型中使用预测因子变量。预测模型包括时间趋势曲线、回归因子、干扰影响(哑元变量)、用户指定的调整以及动态回归(变换函数)模型。

(3) 浏览时间序列图，预测真实值，预测误差，预测置信区间。用户可以浏览序列的修改或变换图，放大图中任意一部分，浏览自相关图等。

(4) 使用HOLD-OUT样本选择最佳预测方法。

(5) 逐个比较任意两个预测模型的拟合优度值或按一特定拟合统计排序的所有模型。

(6) 用表格形式浏览每个模型的预测和误差值或任意选择两个模型比较预测结果。

(7) 检验每个预测模型的拟合参数及其统计显著性。

(8) 控制自动模型选择过程：备选预测模型集，用于选择最佳模型的拟合优度测量值，用于拟合和评估模型的时间段。

(9) 通过添加预测模型到自动模型选择处理列表及手工选择操作来定制系统配置。

(10) 将用户的工作保存在一个项目目录中。

(11) 打印预测过程的所有内容。

(12) 保存和打印包括表格、图形的系统输出。

3）SAS/INSIGHT

SAS/INSIGHT 是一个功能强大的可视化的数据探索与分析的工具。它将统计方法与交互式图形显示融合在一起，为用户提供一种全新的使用统计分析方法的环境。

使用 SAS/INSIGHT，用户可以考察单变量（或指标）的分布，显示多变量（或指标）数据，用回归分析、方差分析和广义线性模型等方法建立模型。生成的图形和分析都是动态的，可以通过三维旋转图形来探索数据，通过单击图形上的点来识别它们，方便、快捷地增加或删除一些变量。所有的图形和分析都是动态地连接在一起的。这些动态连接在一起的图形和分析使得用户可以方便、迅速地发现数据中的规律性。一旦发现了数据中的规律性，就可以快捷地建立模型并分析各指标间的关系。

SAS/INSIGHT 具有以下特点：

（1）方便、快捷地探索数据。

SAS/INSIGHT 以表格的形式显示数据。使用数据视窗，用户可以输入新的数据值、进行数据排序、查询数据，并可以同时打开任意多的数据集。通过单击方式，用户可以方便地生成多种探索性的图形，例如盒须图、直方图、散点图、Mosaic 图、二维可重叠线状图和三维旋转图。用户可以方便地计算描述性统计量和各种相关关系。对于高维数据，可以先进行主成分分析，之后画二维或三维主成分散点图，从而了解高维数据的结构。

（2）快捷地拟合最优模型。

一旦发现了数据中的潜在关系，就可以用模型拟合工具来拟合模型。例如，可以拟合带有置信线的多项式曲线，进行核估计、Splines 平滑、Loess 平滑。所有这些曲线都可以用滑动条进行动态调节。还可以拟合回归分析、方差分析、协方差分析和广义线性模型（如 Logistic 回归、Poisson 回归）等方法建立模型。

（3）有效地检查模型的条件。

用以概要统计量、方差分析、参数估计等评估建立的模型，并可进行共线诊断。用残差-预测值图检查误差项的方差是不是常数并识别异常点，还可以检查强影响点等。

4）SAS/EM

屡获大奖的数据挖掘产品 SAS/EM 是一个图形化界面，是菜单驱动的、拖拉式操作、对用户非常友好且功能强大的数据挖掘集成环境。其中集成了：

- 数据获取工具。
- 数据抽样工具。
- 数据筛选工具。
- 数据变量转换工具。
- 数据挖掘数据库。
- 数据挖掘过程。
- 多种形式的回归工具。

- 为建立决策树的数据剖分工具。
- 决策树浏览工具。
- 人工神经元网络。
- 数据挖掘的评价工具。

可利用 SAS/EM 中具有明确代表意义的图形化模块将这些数据挖掘的工具单元组成一个处理流程图，并以此来组织用户的数据挖掘过程。这一过程在任何时候均可根据具体情况的需要进行修改、更新并将适合用户需要的模式存储起来，以便此后重新调出来使用。SAS/EM 图形化的界面，可视化的操作，即使是数理统计经验不太多的使用者，也能按照 SEMMA 的原则成功地进行数据挖掘。对于有经验的专家，SAS/EM 也可让用户一展身手，精细地调整分析处理过程。

这一强大的数据挖掘工具组合阵容保证了可以支持企业级的数据挖掘各个方面的工作。

(1) 数据获取工具。

在 SAS/EM 的这个数据获取工具中，用户可以通过对话框指定要使用的数据集的名称，并指定要在数据挖掘中使用的数据变量。变量分为两类：区间变量(Interval Variable)和分类变量(Class Variable)。区间变量是指那些要进行统计处理的变量。对于这样的变量，在数据输入阶段用户就可以指定他们是否要进行最大值、最小值、平均值、标准差等的处理，还可给出该变量是否有值的缺漏、缺漏的百分比是多少等。利用这些指定可对输入数据在获取伊始就进行一次检查，并把结果告诉用户，用户可初步审视其质量如何。

区间变量以外的变量称为分类变量。在数据输入阶段将会提供给用户每个分类变量共有多少种值可供分类之用。

(2) 数据抽样工具。

对于获取的数据，可再从中进行抽样操作。抽样的方式是多种多样的，有随机抽样、等距抽样、分层抽样、从起始顺序抽样和分类抽样等方式。

① 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。

② 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 组、第 40 组、第 60 组、第 80 组和第 100 组 5 组观测值。

③ 分层抽样。

在进行这种抽样操作时，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，但对不同的层次用户可设定不同的

概率。这样的抽样结果可能具有更好的代表性，进而使模型具有更好的拟合精度。

④ 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。

⑤ 分类抽样。

在前面几种抽样方式中，抽样的单位都是一组观测值。分类抽样的单位是一类观测值。这里的分类是按观测值的某种属性进行区分的，如按客户名称分类、按地址区域分类等。显然在同一类中可能会有多组观测值。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。

设置多种形式的抽样方式不仅给用户抽样带来了灵活性，更重要的是从抽样阶段用户就能主动地考虑数据挖掘的目的性，强化了最后结论的效果。

(3) 数据筛选工具。

通过数据筛选工具用户可从观测值样本中筛选掉不希望包括进来的观测值。对于分类变量可给定某一类的类值说明此类观测值是要排除在抽样范围之外的。对于区间变量可指定其值大于或小于某值时的这些组观测值是要排除在抽样范围之外的。

通过数据筛选使样本数据更适合用户要进行数据挖掘的目标。

(4) 数据变量转换工具。

利用此工具可将某一个数据进行某种转换操作，然后将转换后的值作为新的变量存放在样本数据中。转换的目的是使用户的数据和将来要建立的模型拟合得更好，例如原来的非线性模型线性化、加强变量的稳定性等，可进行取幂、对数、开方等转换。当然，用户也可给定一个公式进行转换。

在进行数据挖掘分析模型的操作之前，要建立一个数据挖掘用的数据库(DMDB)，其中就放置此次要进行操作的数据。因为此后可能要进行许多复杂的数学运算，在这里建立一个专门的数据集将使用户的工作更加有效率。在处理之前，可对用户选进数据挖掘数据库的各个变量预先进行诸如最大值、最小值、平均值、标准差等处理。对一些要按其分类的变量的等级也要先放入 Meta Data 中，以利于接下来的操作。总之要在这个数据库中为数据挖掘建立一个良好的工作环境。

在数据挖掘的过程中，可以使用 SAS 广泛的数学方法，以及实现最新数学方法的环境。这给用户提供了几乎无所不能的数据挖掘天地。限于篇幅，这里主要介绍几种常用的工具。

(1) 多种形式的回归工具。

在图形化工具提供的回归操作中，主要有线性回归和 Logistic 回归。在线性回归中有若干不同的方法供用户选择，诸如向前、向后的逐步回归等，还给用户指定了多种回归运算结束的准则。

在 Logistic 回归过程中可拟合逻辑型的模型，其中响应变量可以是双值的或者是多值的，也可使用逐步法选择模型，还可以进行回归诊断及计算预测值和残差值。

回归处理结束后，将会给用户提供一份供讨论的详细结果，内容包括：对回归参数的评价；对于模型拟合的统计结果；回归结果的标准输出，如 F-检验、均方差、自由度等；回归运行的 LOG；全部回归处理程序的代码；对此次回归记录的文档资料。

(2) 为建立决策树的数据剖分工具。

对数据集进行聚类、剖分建立决策树，是近来数据处理、进行决策支持常用的方法。在 SAS/EM 中也支持这一功能。在建立决策树的过程中，有多种数据聚类、剖分的方法供用户选择。

图形化界面的交互式操作可分成 6 个层次：

- 对用户在数据挖掘数据库中选定的数据集的操作。
- 对数据集中的变量的处理。
- 聚类、剖分时的基本选择项。
- 聚类、剖分时的进一步操作选择项。
- 模型的初步确定。
- 结果的评价。

聚类、剖分可以多种不同的方法进行，不能说哪种方法更“准确”，这要看是否满足了用户决策问题的需要，也许用户应当试试不同方法所产生的结果。恰好 SAS/EM 不仅具有多种多样的处理方式可供选择，而且具有相当高的“自动化”程度，使用户能以极快的速度尝试多种方法，尽快得出最佳选择。

(3) 决策树浏览工具。

用户最后做出的满意的决策树可能是一个“枝繁叶茂”的架构。SAS/EM 给用户提供了可视化的浏览工具。这一点很重要，一个复杂的决策树若难以观察，则会影响用户实施决策时的效率，甚至是有效性。决策树浏览工具包括：

- 决策树基本内容和统计值的汇总表。
- 决策树的导航浏览器。
- 决策树的图形显示。
- 决策树的评价图表。
- 人工神经元网络。

人工神经元网络是近来使用越来越广的模型化方法，特别是对回归中难以处理的非线性关系问题，它往往能以更真实反映世界的能力使之得到更灵活的处理。在 SAS/EM 中有强有力地实现人工神经元网络模型的各种工具，使用户免除了繁杂的数据处理，把精力集中于模型本身的考虑。

在 SAS/EM 中的人工神经元网络应用功能可以处理线性模型、多层感知机模型

（Multilayer Perceptron，MLP，这是采用较多的默认方式）和放射型功能（Radial Basis Function，RBF）。在交互式图形化界面上，在一个在线的关于 SAS 人工神经元网络问答的支持下，使用户能高效地通过以下 4 个步骤建立人工神经元网络的模型：

- 数据准备。
- 神经网络的定义。
- 人工神经元网络的训练。
- 生成预报模型。

（4）用于统计分析的集成类数据挖掘工具。

① IBM SPSS。

SPSS（Statistical Package for the Social Sciences）是目前最流行的统计软件平台之一。自 2015 年开始提供统计产品和服务方案以来，该软件的各种高级功能被广泛地运用于学习算法、统计分析（包括描述性回归、聚类等）、文本分析以及与大数据集成等场景中。同时，SPSS 允许用户通过各种专业性的扩展，运用 Python 和 R 来改进其 SPSS 语法。SPSS 的图形界面如图 3.21 所示。

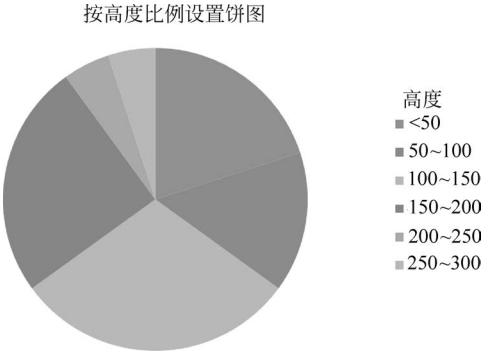


图 3.21　IBM 的 SPSS

② R。

如前所述，R 是一种编程语言，可用于统计计算与图形环境。它能够与 UNIX、FreeBSD、Linux、macOS 和 Windows 操作系统相兼容。R 可以被运用在诸如时间序列分析、聚类以及线性与非线性建模等各种统计分析场景中。同时，作为一种免费的统计计算环境，它还能够提供连贯的系统和各种出色的数据挖掘包，可用于数据分析的图形化工具以及大量的中间件工具。此外，它也是 SAS 和 IBM SPSS 等统计软件的开源解决方案。

③ SAS。

SAS（Statistical Analysis System）是数据与文本挖掘（Text Mining）及优化的合适选择。它能够根据组织的需求和目标，提供多种分析技术和方法功能。目前，它能够提

供描述性建模(有助于对客户进行分类和描述)、预测性建模(便于预测未知结果)和解析性建模(用于解析、过滤和转换诸如电子邮件、注释字段、书籍等非结构化数据)。此外,其分布式内存处理架构还具有高度的可扩展性。

④ Oracle Data Mining。

Oracle Data Mining(ODB)是 Oracle Advanced Analytics 的一部分。该数据挖掘工具提供了出色的数据预测算法,可用于分类、回归、聚类、关联、属性重要性判断,以及其他专业分析。此外,ODB 也可以使用 SQL、PL/SQL、R 和 Java 等接口来检索有价值的数据见解,并予以准确的预测。

(5) 开源的数据挖掘工具。

① KNIME。

于 2006 年首发的开源软件 KNIME(Konstanz Information Miner)如今已被广泛地应用在银行、生命科学、出版和咨询等行业的数据科学和机器学习领域。同时,它提供本地和云端连接器,以实现不同环境之间数据的迁移。虽然它是用 Java 实现的,但是 KNIME 提供了各种节点,以方便用户在 Ruby、Python 和 R 中运行它。KNIME 的工作流程如图 3.22 所示。

图 3.22　KNIME 的工作流程

② RapidMiner。

作为一种开源的数据挖掘工具,RapidMiner 可与 R 和 Python 无缝地集成。它通过提供丰富的产品来创建新的数据挖掘过程,并提供各种高级分析。同时,RapidMiner 是由 Java 编写的,可以与 WEKA 和 R-tool 相集成,是目前好用的预测分析系统之一。它能够提供诸如远程分析处理、创建和验证预测模型、多种数据管理方法、内置模板、可重复的工作流程、数据过滤以及合并与连接等多项实用功能。

③ Orange。

Orange 是基于 Python 的开源式数据挖掘软件。当然,除了提供基本的数据挖掘功能外,Orange 也支持用于数据建模、回归、聚类、预处理等领域的机器学习算法。同时,Orange 还提供了可视化的编程环境,以及方便用户拖放组件与链接的能力。

(6) 大数据类数据挖掘工具。

从概念上说,大数据既可以是结构化的,也可以是非结构化或半结构化的。它通常

涵盖5个V的特性，即体量(Volume，可能达到TB或PB级)、多样性(Variety)、速度(Velocity)、准确性(Veracity)和价值(Value)。鉴于其复杂性，我们对于海量数据的存储、模式的发现以及趋势的预测等，都很难在一台计算机上处理与实现，因此需要用到分布式的数据挖掘工具。

① Apache Spark。

Apache Spark凭借其处理大数据的易用性与高性能而备受欢迎。它具有针对Java、Python(PySpark)、R(SparkR)、SQL、Scala等多种接口，能够提供80多个高级运算符，以方便用户更快地编写出代码。另外，Apache Spark也提供了针对SQL and DataFrames、Spark Streaming、GraphX和MLlib的代码库，以实现快速的数据处理和数据流平台。

② Hadoop MapReduce。

Hadoop是处理大量数据和各种计算问题的开源工具集合。虽然是用Java编写而成的，但是任何编程语言都可以与Hadoop Streaming协同使用。其中MapReduce是Hadoop的实现和编程模型。它允许用户“映射(Map)”和“简化(Reduce)”各种常用的功能，并且可以横跨庞大的数据集，执行大型连接(Join)操作。此外，Hadoop也提供了诸如用户活动分析、非结构化数据处理、日志分析以及文本挖掘等应用。目前，它已成为一种针对大数据执行复杂数据挖掘的广泛适用方案。

③ Qlik。

Qlik是一个能够运用可扩展且灵活的方法来处理数据分析和挖掘的平台。它具有易用的拖放界面，并能够即时响应用户的修改和交互。为了支持多个数据源，Qlik通过各种连接器、扩展、内置应用以及API集实现与各种外部应用格式的无缝集成。同时，它也是集中式共享分析的绝佳工具。

(7) 小型数据挖掘方案。

① Scikit-Learn。

作为一款可用于Python机器学习的免费软件工具，Scikit-Learn能够提供出色的数据分析和挖掘功能。它具有诸如分类、回归、聚类、预处理、模型选择以及降维等多种功能。Scikit-Learn的分层聚类如图3.23所示。

② Rattle(R)。

由R语言开发的Rattle能够与macOS、Windows和Linux等操作系统相兼容。它主要被美国和澳大利亚的用户用于企业商业与学术目的。R语言的计算能力能够为用户提供诸如聚类、数据可视化、建模以及其他统计分析类功能。

③ Pandas(Python)。

Pandas是利用Python进行数据挖掘的“一把好手”。由它提供的代码库既可以被用

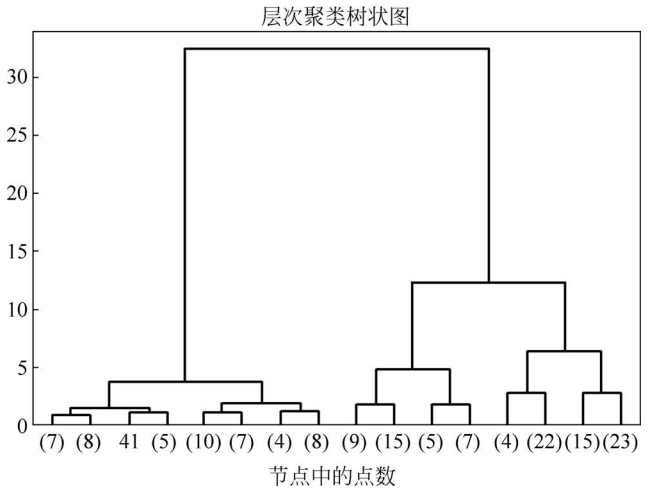


图 3.23　Scikit-Learn 的分层聚类

来进行数据分析，又可以管理目标系统的数据结构。

④ H3O。

作为一种开源的数据挖掘软件，H3O 可以被用来分析存储在云端架构中的数据。虽然是由 R 语言编写的，但是该工具不但能与 Python 兼容，而且可以用于构建各种模型。此外，得益于 Java 的语言支持，H3O 能够被快速、轻松地部署到生产环境中。

(8) 用于云端数据挖掘的方案。

通过实施云端数据挖掘技术，用户可以从虚拟的集成数据仓库中检索到重要的信息，进而降低存储和基础架构的成本。

① Amazon EMR。

作为处理大数据的云端解决方案，Amazon EMR 不仅可以被用于数据挖掘，还可以执行诸如 Web 索引、日志文件分析、财务分析、机器学习等数据科学工作。该平台提供了包括 Apache Spark 和 Apache Flink 在内的各种开源方案，并且能够通过自动调整集群之类的任务来提高大数据环境的可扩展性。

② Azure ML。

作为一种基于云服务的环境，Azure ML 可用于构建、训练和部署各种机器学习模型。针对各种数据分析、挖掘与预测任务，Azure ML 可以让用户在云平台中对不同体量的数据进行计算和操控。

③ Google AI Platform。

与 Amazon EMR 和 Azure ML 类似，基于云端的 Google AI Platform 也能够提供各种机器学习栈。Google AI Platform 包括各种数据库、机器学习库以及其他工具。用户可以在云端使用它们，以执行数据挖掘和其他数据科学类任务。

（9）使用神经网络的数据挖掘工具。

神经网络主要以人脑处理信息的方式来处理数据。换句话说，由于我们的大脑有着数百万个处理外部信息并随之产生输出的神经元，因此神经网络可以遵循此类原理，通过将原始数据转换为彼此相关的信息来实现数据挖掘的目的。

① PyTorch。

PyTorch 既是一个 Python 包，也是一个基于 Torch 库的深度学习框架。它最初是由 Facebook 的 AI 研究实验室（FAIR）开发的，属于深层的神经网络类数据科学工具。用户可以通过加载数据、预处理数据、定义模型执行训练和评估，这样的数据挖掘步骤通过 PyTorch 对整个神经网络进行编程。此外，借助强大的 GPU 加速能力，Torch 可以实现快速的阵列计算。截至 2020 年 9 月，Torch 的 R 生态系统中已包含 torch、torchvision、torchaudio 以及其他扩展。PyTorch 的神经网络结构如图 3.24 所示。

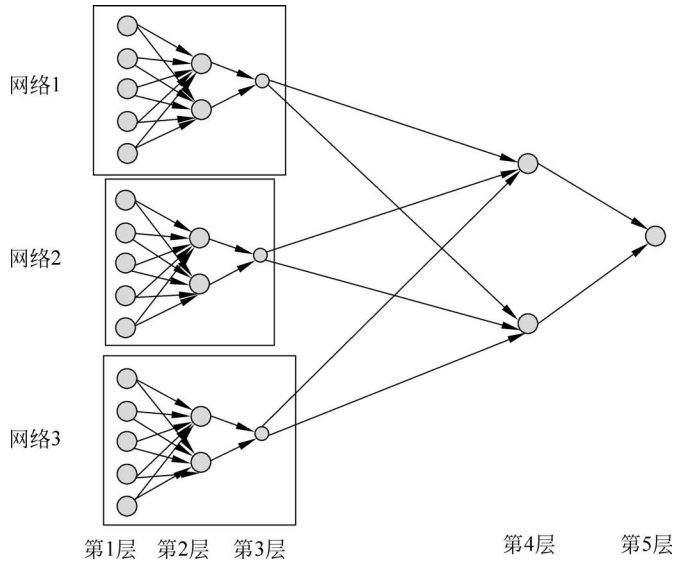


图 3.24　PyTorch 的神经网络结构

② TensorFlow。

与 PyTorch 相似，由 Google Brain Team 开发的 TensorFlow 也是基于 Python 的开源机器学习框架。它既可以被用于构建深度学习模型，又能够高度关注深度神经网络。TensorFlow 生态系统不但能够灵活地提供各种库和工具，而且拥有一个广泛的流行社区，开发人员可以进行各种问答和知识共享。尽管属于 Python 库，但是 TensorFlow 于 2017 年开始对 TensorFlow API 引入 R 接口。

（10）用于数据可视化的数据挖掘工具。

数据可视化是对从数据挖掘过程中提取的信息予以图形化表示。此类工具能够让用户通过图形、图表、映射图以及其他可视化元素直观地了解数据的趋势、模型和异

常值。

① Matplotlib。

Matplotlib 是使用 Python 进行数据可视化的出色工具库。它允许用户利用交互式的图形来创建诸如直方图、散点图、3D 图等质量图表，而且这些图表都可以从样式、轴属性、字体等方面被自定义。Matplotlib 的图表示例如图 3.25 所示。

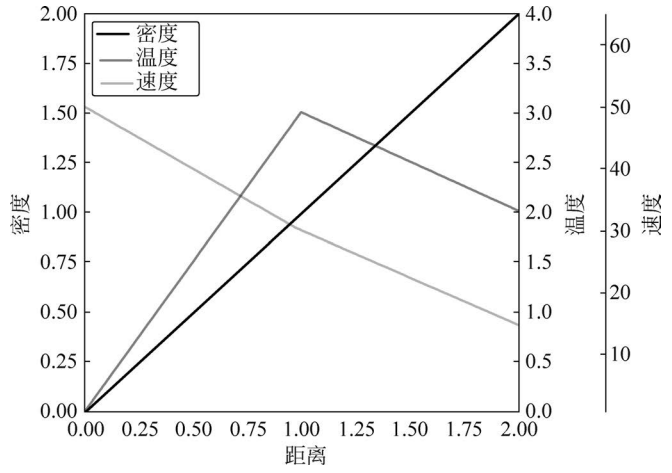


图 3.25　Matplotlib 的图表示例

② ggplot2。

ggplot2 是一款广受欢迎的数据可视化 R 工具包，它允许用户构建各类高质量且美观的图形。同时，用户也可以通过该工具高度抽象地修改图中的各种组件。

3.7　数据挖掘的评价工具

在 SAS/EM 的评价工具中，向用户提供了一个通用的数据挖掘评价的架构，可以比较不同的模型效果，预报各种不同类型分析工具的结果。

在进行了各种比较和预报的评价之后，将给出一系列标准的图表，供用户进行定量评价。可能用户会有自己独特的评价准则，在 SAS/EM 的评价工具中，还可以进行客户化的工作，对那些标准的评价图表按照用户的具体要求进行更改。这样一来，评价工作可能就会更有意义。

SAS/EM 让用户以可操作的规范性实现了 SEMMA 数据挖掘方法。它所涵盖的技术深度和广度用户是可以预见的。这对于各种不同类型的计算机用户来说都是非常适合的。如果让用户自己规划这样一个系统，可能很难想象得这样完整，更不要说用户是否有这么多的时间和精力像 SAS 的数据挖掘专家这样来开发这样的工具。

如何选择满足自己需要的数据挖掘工具呢？本节介绍评价一个数据挖掘工具需要从哪些方面来考虑。

3.7.1 可产生的模式种类的多少

分类模式是一种分类器，能够把数据集中的数据映射到某个给定的类上，从而应用于数据预测。它常表现为一棵分类树，根据数据的值从树根开始搜索，沿着数据满足的分支往上走，走到树叶就能确定类别。

回归模式与分类模式相似，其差别在于分类模式的预测值是离散的，回归模式的预测值是连续的。

1. 时间序列模式

根据数据随时间变化的趋势来预测将来的值。其中要考虑时间的特殊性质，只有充分考虑时间因素，利用现有的数据随时间变化的一系列值，才能更好地预测将来的值。

2. 聚类模式

把数据划分到不同的组中，组之间的差别尽可能大，组内的差别尽可能小。与分类模式不同，进行聚类前并不知道要划分成几个组和什么样的组，也不知道根据哪些数据项来定义组。

3. 关联模式

关联模式是数据项之间的关联规则。而关联规则是描述事物之间同时出现的规律的知识模式。在关联规则的挖掘中要注意以下几点：

(1) 充分理解数据。

(2) 目标明确。

(3) 数据准备工作要做好。

(4) 选取恰当的最小支持度和最小可信度。

(5) 很好地理解关联规则。

4. 序列模式

与关联模式相似，它把数据之间的关联性与时间联系起来。为了发现序列模式，不仅需要知道事件是否发生，而且需要确定事件发生的时间。

在解决实际问题时，经常要同时使用多种模式。分类模式和回归模式使用得最为普遍。

3.7.2 解决复杂问题的能力

数据量的增大，对模式精细度、准确度要求的增高都会导致问题复杂性的增大。数据挖掘系统可以提供下列方法解决复杂问题：

- 多种模式。多种类别模式的结合使用有助于发现有用的模式，降低问题的复杂性。例如，首先用聚类的方法把数据分组，然后在各个组上挖掘预测性的模式，将会比单纯在整个数据集上进行操作更有效、准确度更高。
- 多种算法。很多模式，特别是与分类有关的模式，可以由不同的算法来实现，各有各的优缺点，适用于不同的需求和环境。数据挖掘系统提供多种途径产生同种模式，将更有能力解决复杂问题。
- 验证方法。在评估模式时，有多种可能的验证方法。比较成熟的方法像 N 层交叉验证或 Bootstrapping 等可以控制，以达到最大的准确度。

N 层交叉验证采用数据选择和转换模式，该方法通常被大量的数据项所隐藏。有些数据是冗余的，有些数据是完全无关的。而这些数据项的存在会影响有价值的模式的发现。数据挖掘系统的一个很重要的功能就是能够处理数据复杂性，提供工具，选择正确的数据项和转换数据值。

可视化工具提供直观、简洁的机制表示大量的信息。这有助于定位重要的数据，评价模式的质量，从而减少建模的复杂性。

为了更有效地提高处理大量数据的效率，数据挖掘系统的扩展性十分重要。需要了解的是：数据挖掘系统能否充分利用硬件资源；是否支持并行计算；算法本身设计为并行的或利用了 DBMS 的并行性能；支持哪种并行计算机，SMP 服务器还是 MPP 服务器；当处理器的数量增加时，计算规模是否相应增长；是否支持数据并行存储。

为单处理器的计算机编写的数据挖掘算法不会在并行计算机上自动以更快的速度运行。为了充分发挥并行计算的优点，需要编写支持并行计算的算法。

3.7.3　易操作性

易操作性是一个重要的因素。有的工具有图形化界面，引导用户半自动化地执行任务，有的使用脚本语言。有些工具还提供数据挖掘的 API，可以嵌入像 C、Visual Basic、PowerBuilder 这样的编程语言中。

模式可以运用到已存在或新增加的数据上。有的工具有图形化的界面，有的允许通过使用 C 这样的程序语言或 SQL 中的规则集，把模式导出到程序或数据库中。

3.7.4　数据存取能力

好的数据挖掘工具可以使用 SQL 语句直接从 DBMS 中读取数据。这样可以简化数据准备工作，并且可以充分利用数据库的优点（比如平行读取）。没有一种工具可以支持大量的 DBMS，但可以通过通用的接口连接大多数流行的 DBMS。Microsoft 的 ODBC

就是一个这样的接口。

3.7.5 与其他产品的接口

有很多别的工具可以帮助用户理解数据，理解结果。这些工具可以是传统的查询工具、可视化工具、OLAP 工具。数据挖掘工具是否能提供与这些工具集成的简易途径？

因为数据挖掘工具需要考虑的因素很多，很难按照原则给工具排一个优劣次序。最重要的还是用户的需要，根据特定的需求加以选择。数据挖掘工具可以给很多产业带来收益。国外的许多行业如通信、信用卡公司、银行和股票交易所、保险公司、广告公司、商店等已经大量利用数据挖掘工具来协助其业务活动，国内在这方面的应用还处于起步阶段，对数据挖掘技术和工具的研究人员以及开发商来说，我国是一个有巨大潜力的市场。

习题

1. 数据仓库、数据集市与数据挖掘有哪些区别与联系？
2. 简述 SAS 的数据挖掘理论。
3. 数据挖掘的常用算法有哪些？试举例说明。
4. 空间数据挖掘的常用方法有哪些？
5. 简单介绍几种常用的数据挖掘工具。
6. 如何选择合适的数据挖掘工具？