Broadview
WWW.BROADVIEW.COM.CN

数据仓库与数据挖掘
技术应用丛书

更通俗的机器学习方法介绍
更广阔的数据挖掘应用视角
更丰富的机器学习与数据挖掘的解决方法

Machine Learning and Data Mining:Methods and Applications

机器学习与数据挖掘：方法和应用

〔美〕Ryszard S. Michalski　Ivan Bratko　Miroslav Kubat 等著
朱　明 等译

WILEY
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Broadview
博文视点资讯有限公司总策划

在当今商业和科学的世界里，人们面对着不断涌现的海量信息，正逐步陷入“数据丰富，知识贫乏”的尴尬境地。因而，从海量信息中提取有意义的模式和策略知识，已经越来越成为一种挑战。

相应地，为在来自于数据库、数据仓库和文档信息系统的信息中发现模式和一般规则，需要开发一些方法。这本书是第一本致力于机器学习和数据挖掘这两个领域交叉主题的书——这两个领域的交叉部分提供了这些方法的基础。

本书是由机器学习和数据挖掘这两个领域中的国际著名专家组成的写作小组精心成就的，融入了很多激动人心的成果。令人印象更深刻的是，本书描述了很多领域中的实际问题，如工程、计算机控制、生物、机械和音乐等方面。

ISBN 7-5053-9224-7

9 787505 392243 >

WILEY

责任编辑：孙学瑛
封面设计：张子建

ISBN 7-5053-9224-7 TP·5331 定价 58.00元

数据仓库与数据挖掘技术应用丛书

机器学习与数据挖掘：方法和应用

Machine Learning and Data Mining: Methods and Applications

[美]Ryszard S.Michalski Ivan Bratko Miroslav Kubat 等著

朱 明 等译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书分为5个部分，共18章，较为全面地介绍了机器学习的基本概念，并讨论了数据挖掘和知识发现中的有关问题及多策略学习方法，具体地阐述了机器学习与数据挖掘在工程设计，文本、图像和音乐，网页分析、计算机病毒和计算机控制，医疗诊断、生物医疗信号分析和水质分析中的生物信号处理等方面的应用情况。

本书收集众多不同领域中数据挖掘的实际案例，以此来说明数据挖掘的具体解决方法，以期为广大读者提供一个更为广阔的数据挖掘应用视角。

本书的读者，可以是任何对机器学习与数据挖掘感兴趣的工程技术人员、业务管理人员，或是从事具体技术工作的其他人员。本书也可作为大专院校相关课程的重要辅导教材。

版权贸易合同登记号：　图字：01-2002-6435

图书在版编目(CIP)数据

机器学习与数据挖掘：方法和应用/（美）米哈尔斯基（Michalski, R.S.）等著；朱明等译．—北京：电子工业出版社，2004.1

（数据仓库与数据挖掘技术应用丛书）

书名原文：Machine Learning and Data Mining:Methods and Applications

ISBN 7-5053-9224-7

Ⅰ.机…　Ⅱ.①米…　②朱…　Ⅲ.①机器学习　②数据采集　Ⅳ.①TP18　②TP274

中国版本图书馆CIP数据核字（2003）第090444号

责任编辑：孙学瑛

印　　刷：北京增富印刷有限公司

出版发行：电子工业出版社　http://www.phei.com.cn

　　北京市海淀区万寿路173信箱　邮编　100036

经　　销：各地新华书店

开　　本：787×980　1/16　印张：28　字数：625千字

版　　次：2004年1月第1版　2004年1月第1次印刷

印　　数：4 000册　　定价：58.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

出 版 说 明

如果没有对海量数据进行科学分析的能力，沃尔玛的老板再精明，也绝对想不到“啤酒与尿布”这两个风马牛不相及的东西之间还有着千丝万缕的联系，而将它们放在一起，竟然增加了啤酒销量，可见数据分析的巨大威力。

信息系统数年中收集了海量数据，且数据还正以指数级增长，企业迫切地需要高效、精确、科学地分析数据，以找出其背后的寓意，进而了解企业的经营状况和外部环境，做出科学的决断，在现代激烈的竞争中胜出。所以，如何将数据点石成金，更是摆在我们面前很现实也很诱人的一个问题。

现在，很多人已经意识到数据中潜在的大量商机，并踏踏实实地进行着从数据中沙里淘金的工作。特别是在信息化的大潮中，上至政府，下到企业，从银行到电信，再到网站、超市，人们都希望用数据分析这根魔杖赢得先机。与此同时，人们也在企盼着相关书籍，以便工作中学习参考。在广泛征询专家和用户的基础上，秉着选题全面、内容经典、译者严谨的原则，我们适时地推出了这套《数据仓库与数据挖掘技术应用丛书》，以飨读者。本丛书有如下几本：

- 数据仓库基础
- OLAP 解决方案：多维信息系统的构建技术
- 数据仓库工具箱：维度建模的完全指南（第二版）
- 数据仓库生命周期工具箱：设计、开发和部署数据仓库的专家方法
- 数据仓库及其在电信领域中的应用
- 疑难数据仓库专家解决方案
- IBM 数据仓库和商业智能工具
- 可视化数据挖掘：数据可视化和挖掘的技术和工具
- 点击流数据仓库
- Web 数据挖掘：将客户数据转化为客户价值
- 企业信息工厂
- 机器学习与数据挖掘：方法和应用

本丛书既包括商业智能（BI）的基础——数据仓库（DW），也包括数据仓库上的两类不同目的的数据增值操作——联机分析处理（OLAP）和数据挖掘（DM）；既覆盖基础理论，如数据仓库基础，又提供不同领域的解决方案，如数据仓库在电信、银行、保险等领域的应用。

本丛书来自国外数据库领域一些著名作者的畅销书，以及国内第一线实施者的精心总结。如一直位居 AMAZON 畅销书榜的数据仓库领域的畅销书作家 Ralph Kimball 的《数据仓库工具箱：维度建模的完全指南（第二版）》、《数据仓库生命周期工具箱：设计、开发和部署数据仓库的专家方法》，数据仓库之父 William H.Inmon 的《企业信息工厂》等。

从书的译者均来自工作在该领域一线的人员，既有该领域的理论和实践经验，又具备中英文翻译的功底。且多位译者先前均已读过原著，所以，自感翻译的过程不再是枯燥，而是情趣盎然，乐在其中。

出版高品位、高品质的图书是博文视点的努力目标。希望您对我们的工作多提宝贵意见。您的意见是我们创造精品的动力源泉。

如果您希望将您的工作经验感悟等总结成书，我们将为您提供一流的服务，共创精品图书。

我们的联系方式如下：

地址：北京复兴路 47 号天行建商务大厦 604

邮编：100036

电话：010-51922832，68216158

传真：010-51922823

E-mail：jsj@phei.com.cn；zsh@phei.com.cn

博文视点资讯有限公司

2003 年 10 月

博文视点资讯有限公司（BROADVIEW Information Co.,Ltd.）是信息产业部直属的中央一级科技与教育出版社——电子工业出版社（PHEI）与国内最大的IT技术网站CSDN.NET和最具专业水准的IT杂志社《程序员》合资成立的以IT图书出版为主业、开展相关信息和知识增值服务的资讯公司。

我们的理念是：创新专业出版体制；培养职业出版队伍；打造精品出版品牌；完善全面出版服务。

秉承博文视点的理念，博文视点的产品线为面向IT专业人员的出版物和相关服务。博文视点将重点做好以下工作：

（1）在技术领域开发专业作（译）者群体和高质量的原创图书

（2）在图书领域建立专业的选题策划和审读机制

（3）在市场领域开创有效的宣传手段和营销渠道

博文视点有效地综合了电子工业出版社、《程序员》杂志社和CSDN.NET的资源和人才，建立全新专业的立体出版机制，确立独特的出版特色和优势，将打造IT出版领域的著名品牌，并力争成为中国最具影响力的专业IT出版和服务提供商。

作为合资公司，博文视点的团队融合了各方面的精英力量：原电子工业出版社IT图书专业出版实力的代表部门——计算机图书事业部的团队；《程序员》杂志社和CSDN网站的主创人员；著名IT专业图书策划人周筠女士及其创作群。这是一个整合专业技术人员和专业出版人员的团队；这是一个充满创新意识和创作激情的团队；这是一个不断进取、追求卓越的团队。

电子工业出版社与《程序员》杂志和CSDN网站的合作以最有效率的方式形成了出版资源、媒体资源、网络资源的整合和互动，成为2003年IT出版界备受瞩目的事件。

“技术凝聚实力，专业创新出版”，BROADVIEW与您携手共迎信息时代的机遇与挑战！

博文视点

地址：北京市万寿路47号天行建商务大厦604室

邮编：100036

总机：010-51922832　传真：010-51922823

本版编辑部：010-51922839　外版编辑部：010-51922825

http://www.broadview.com.cn　投稿及读者反馈：editor@broadview.com.cn

译 者 序

随着数据库技术的应用越来越普及，人们正逐步陷入“数据丰富，知识贫乏”的尴尬境地。在此背景下，数据挖掘技术，又称数据库知识发现，于20世纪90年代开始迅速兴起。这一信息领域是基于机器学习、统计分析等多种学科的计算机技术的，它能够有效帮助人们将巨大的数据资源转换为有用的知识与信息资源，进而可以帮助人们科学地做出各种决策。

机器学习是一个有关对学习过程中的计算方法的研究，以及如何应用基于计算机的学习系统解决实际问题的学科领域。机器学习中一个重要的研究内容就是对于从样本中获取相应概念描述方法的研究。因此许多机器学习方法可直接用来解决数据挖掘问题。而数据挖掘问题就是从大规模数据库中搜索出有趣模式和重要规律的问题。

本书主要是为广大机器学习或数据挖掘的非专业人员，却对机器学习或数据挖掘应用及其入门知识感兴趣的读者而编写的。因此本书收集众多不同领域中数据挖掘的实际案例，以此来说明数据挖掘的具体解决方法，期望能为广大读者提供一个更为广阔的数据挖掘应用视角。而这一点正是本书与其他数据挖掘书籍的最大区别。

本书分为5个部分，共18章，较为全面地介绍数据挖掘在众多领域和实际问题求解中的应用情况。本书第1部分，共4章，分别介绍了机器学习的基本概念，并讨论了数据挖掘和知识发现中的有关问题及多策略学习方法。此外第1部分还介绍了机器学习与归纳逻辑编程的一些实际应用案例。

本书第2部分，共4章，主要介绍机器学习与数据挖掘在工程设计方面的应用情况。具体内容包括：有限元设计问题求解中的机器学习方法；工业机器故障检测中的基于事例推理方法；设备部件装配规划求解的多策略方法；防摩擦轴承系统中的归纳学习方法。

本书第3部分，共3章，主要介绍机器学习与数据挖掘在文本、图像和音乐方面的应用情况。具体内容包括：文本之间相互关系的挖掘方法；行李X光片中引爆雷管图像的模式识别方法；音乐表达规律的数据挖掘方法。

本书第 4 部分，共 4 章，主要介绍机器学习与数据挖掘在网页分析、计算机病毒和计算机控制方面的应用情况。具体内容包括：网页搜索中网页分类识别的机器学习方法；计算机病毒自动分析识别中的机器学习方法；控制技术中行为复制的机器学习方法；空中交通控制中的数据挖掘方法。

本书第 5 部分，共 3 章，主要介绍机器学习与数据挖掘在医疗诊断、生物医学信号分析和水质分析中生物信号处理方面的应用情况。具体内容包括：机器学习在医学诊断中的应用；生物信号分析中的机器学习方法；机器学习在河流水质生物分类中的应用。

本书内容从问题描述出发，着重介绍各个问题的实质，如何利用机器学习与数据挖掘来解决相关问题，以及所介绍的机器学习与数据挖掘方法的基本特点等，力求浅入深出，以便能够为广大非机器学习与数据挖掘专业的读者，提供一个了解机器学习与数据挖掘的应用视角。本书立足应用，介绍相关的机器学习与数据挖掘方法，因此更加通俗易懂。此外本书所涵盖的大量机器学习与数据挖掘实际案例，也会为读者联系自己的实际问题，提出有效的基于机器学习与数据挖掘的解决方法，奠定了较好的基础。

本书的读者，可以是任何对机器学习与数据挖掘感兴趣的工程技术人员、业务管理人员，或者是从事具体技术工作的其他人员。本书也可作为大专院校相关课程的重要辅导教材。

本书的翻译工作得到了吴炜、徐骞、王庆伟、顾智宇、钟捷飞、李靖、黄永刚、黄科、朱磊、殷俊、黄振等许多同志的积极协助，他们帮助完成了本书部分内容的翻译与校对工作，在此向他们表示感谢。此外，我妻子王晓岚帮助完成了本书部分内容的录入工作，特此致谢。

由于翻译时间仓卒，加上本书涉及诸多实际应用领域，专业用语和词汇较多，本书翻译内容可能会存在一些问题，敬请读者谅解，并批评指正。

译　者

2003.08.08 于合肥

前　言

本书的读者对象不一定是机器学习或数据挖掘方面的专家，而是对机器学习或数据挖掘的多种应用及其入门知识感兴趣的读者。机器学习是一个致力于有关学习过程中计算方法的开发和研究，以及应用计算机学习系统解决实际问题的领域。机器学习中一个重要的研究内容就是从样本中获取相应的概念描述的方法。样本表示可以采用多种形式，尤其是可以采用二维关系数据表形式来表示，因此许多机器学习方法可被直接应用于解决数据挖掘问题，并起着重要的作用。数据挖掘问题就是从大规模数据库中搜索出有趣模式和重要规律。而在数据挖掘问题中，具有挑战性且目前存在许多尚待解决问题的领域，则是有关从文本、图像或声音序列（音乐）之中抽取出模式或规则的问题。本书有一部分专门讨论这类问题的有关应用情况。

本书第 1 部分介绍了机器学习的基本概念，并讨论了数据挖掘和知识发现中的有关问题及多策略方法。这部分还介绍了机器学习与归纳逻辑编程的若干应用，其中后者是机器学习的当前一个分支领域。

随后各个章节是由若干应用主题组成的。这些主题包括：设计与工程，文本、图像和音乐中模式或规则的发现，计算机与控制系统及医药与生物。本书中所介绍的应用中有一些是已投入使用的，而另一些则是研究性应用。本书收集应用实例的指导性标准就是：机器学习方法必须应用于一个困难而有意义的现实世界问题中，并产生满意的或至少是真正有前景的结果。

本书是由一组国际科学家实际编写而成的。每个章节均有单独的作者，他们是机器学习或相关领域的主要专家，分别代表了来自 11 个国家的主要研究小组，这些国家分别是：澳大利亚、奥地利、比利时、加拿大、法国、以色列、韩国、波兰、斯洛文尼亚、英国和美国。然而编者有必要指出：由于现在有如此多的优秀研究人员在机器学习和数据挖掘领域工作，因此不可能挑选出特别“具有代表性”的作者。许多被邀请为本书撰稿的人士目前或以前就是本书编者的同事或合作者。然而被选中的作者都是杰出的科学家，他们的工作涵盖了本书主题的很大范围，因此编者为能够将他们所贡献的内容编入到本书中而感到非常荣幸。

编者借此机会希望感谢所有为本书工作的人们。特别要感谢 George Mason 大学机器学习和推理实验室的研究人员，他们在各种技术和文稿方面提供了无价的帮助，特别是 Zoran Duric，Ken Kaufman，Seokwon Lee 和 Qi Zhang。我们还要感谢 John Wiley & Sons 公司的 Roslyn Meredith 和 Gaynor Redvers-Mutton，他们在本项目中也提供了密切的合作。

最后，我们希望读者们将会从本书中得到有益的帮助和指导，并能够将它作为有关机器学习与数据挖掘及其无数实际应用的有价值的信息源。

Ryszard S. Michalski

Ivan Bratko

Miroslav Kubat

目　录

第1部分　基本概念

第 2 部分 设计与工程

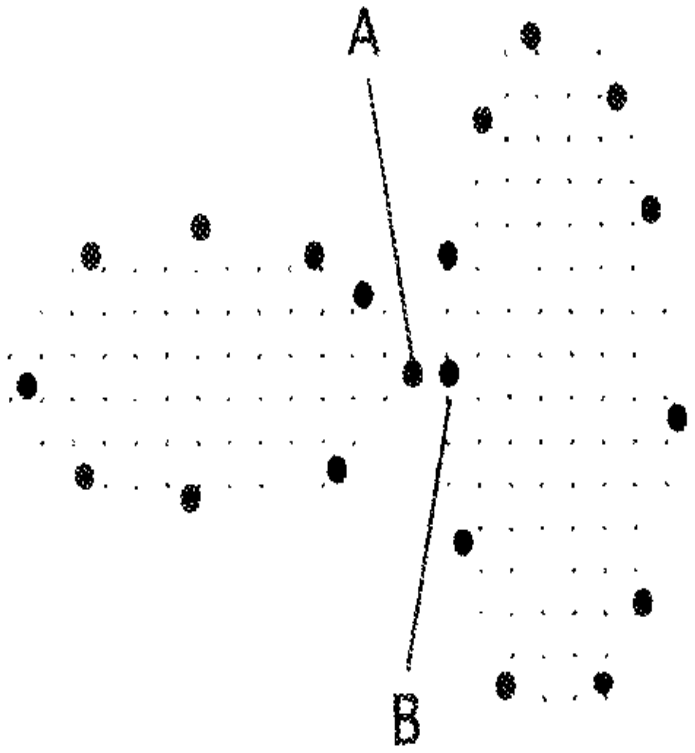
第3部分　文本、图像和音乐模式的测定

第 4 部分 计算机系统和控制系统

第 5 部分 医学和生物学

第 1 部分

基本概念



第1章　机器学习方法概述

Miroslav Kubat, Ivan Bratko 和 Ryszard S. Michalski

1.1 导论

早在40年前，人们就认为机器学习领域研究的主要目标就是开发能够实现各种学习形式的计算方法，尤其是能够从样本或数据中归纳出知识的机制。随着软件开发已越来越成为当今计算机技术的主要瓶颈之一，利用实例将知识引入计算机的思想似乎更加吸引人，更具有号召力。在缺乏求解算法，定义不明确或仅仅非正式表述的问题中，更加需要知识归纳这种形式。医疗或技术诊断、可视化概念识别、工程设计、材料行为、博弈或在大量数据中发现有趣规律，均是这类问题的实例。

人工智能研究中的重要发明之一就是，被确定为无法解决的问题可以通过扩展传统模式来加以解决：

$$\text{程序} = \text{算法} + \text{数据}$$

更为精确的模式为：

$$\text{程序} = \text{算法} + \text{数据} + \text{领域知识}$$

运用领域知识，以某种适当数据结构编码，构成了求解这类问题的基础。任何学过人工智能的人均知道产生式规则、框架、语义网，以及专家系统中的不确定推理的功能。机器学习系统也是从这一思想获益的。

然而，利用知识，只是把瓶颈从编程人员转移到了知识工程师那里，因为知识工程师必须从专家那里抽取出知识，将其编码到系统中。在任何现实世界应用中，知识获取和编码过程决不是一件容易的事。例如，计算机棋弈专家知道通过“暴力求解方法”可得到比人工智能方法更有效的程序，因为确定一个程序打败大师所需要的知识是非常困难的。大师们可以凭直觉利用他们的技能，但大多数情况下无法将其以产生式规则或其他表示系统的形式传授给一个人工智能系统。棋书上充满了抽象概念，诸如创造力、棋子协作、

弱卒结构、经典棋局，棋手需要数年经验去真正地理解这些概念。这样的概念通常缺乏准确的定义，因此很难将其编码到计算机中。

因此自然就会想到：利用一个学习系统，通过对与人类学习类似的样本进行的处理，来获取这类高层次概念和（或）问题求解策略。

机器学习中的大多数研究都致力于开发能够解决这类问题的有效方法。虽然进展比较缓慢，但还是取得了许多重要的成果。目前本领域反对者的主要论点就是：如果在一个专家系统中，编程已经被知识编码所取代，那么机器学习的知识编码就会被实例归纳所替代，但是可用的学习系统还没有强大到能在现实领域中取得成功。

本书的目的就是展示在许多实用领域中，机器学习应用已取得了的使用结果。我们给出具体案例研究来说明本领域已经达到了某种研究阶段，其中现有的技术和系统可被用于解决许多现实世界中的问题。

为使机器学习应用结出硕果，两组研究人员需要联合起来：一组熟悉现有的机器学习方法；另一组具有特定应用领域技能并能提供训练数据。本书旨在吸引计算机科学之外学科的潜在用户的注意。若能够唤起他们的兴趣并使他们能够考虑把机器学习应用于那些传统方法不奏效的问题中，也就不枉作者的辛苦。

为激发非专业人士机器学习的兴趣，本章将介绍理解本领域所必需的方法。在本书的案例介绍中就用到了这些方法。第 1.2 节讨论了概念的含义和其作为知识单元的作用，然后简要介绍了在计算机记忆中的概念表示的相关问题。第 1.3 节描述了基本学习任务是如何作为表示空间的一种搜索的。在这些了解的基础上，第 1.4 节讨论了概念学习的两种基本方法，由此为第 1.5 节打下了基础，第 1.5 节描述基于一阶逻辑的更为复杂的方法。

为加深读者对机器学习研究人员一般思想的理解，第 1.6 节描述了一些发现方法。第 1.7 节简要回顾了利用丰富表达语言建立概念描述的两种方法，这两种语言分别是类比和利用样本本身作为知识的表示。

最后两节，第 1.8 节和 1.9 节，简要介绍了通常不包括在传统符号机器学习中的技术，然而这些技术对于本领域任何专家却是必不可少的。这些技术包括：人工神经网络、遗传算法和各种混合系统。本导论的目标就是让读者具备阅读后面章节所必需的基本知识，而不是提供这一学科全面的介绍。有关更多细节，读者可以参考一些本领域的有关图书，如：Michalski，Carbonell

和 Mitchell（1983，1986），Kodratoff 和 Michalski（1990），Michalski 和 Tecuci（1994），Langley（1996）或 Mitchell（1996）。

1.2 机器学习的任务

机器学习的一般框架如图 1.1 所示。学习系统旨在根据教师所提供的一组概念样本和背景知识，确定特定概念的描述。

图 1.1 机器学习任务

概念样本可为正例（如：在学习哺乳动物时，一只狗）和反例（如：蝎子）。背景知识包括有关描述样本和概念的语言的情况。例如：它可以包括一个变量（属性）的可能取值及其层次、谓语、辅助句法规则、主观喜好等。然后根据样本类型，背景知识的广泛性与相关性，表示问题，所要获取概念的假设本质，以及设计者经验来构造学习算法。

一个重要需求就是学习系统应该能够处理不完善数据。样本常常包含一定的**噪声**，即描述或分类中的错误。例如：一个分类错误就是：若“蝎子”被一个粗心的教师错分为“哺乳动物”类。此外样本由于某些属性值丢失而在某种程度上也是不完全的。背景知识也不一定需要是完美的。

学习算法一般分为两大类别：**黑箱法**（诸如：神经网络或数学统计）和**基于知识方法**。黑箱法有其自己的概念表示方法来用于概念识别。然而其内部表示不易被用户所解读，且对其识别过程没有提供明确的说明或解释。黑箱法通常涉及相关系数、距离或权重的数值计算。

基于知识方法旨在创建满足可理解原则的符号知识结构（Michalski，1983）。Michie（1988）提出了三条标准来说明利用黑箱法与基于知识方法进行概念学习系统之间的差别。这些标准——弱、强和特强——说明它们在力求做到学习概念的可理解性这一方面的差异。

（1）弱标准（Weak Criterion）：系统利用样本数据产生更新，改进后续数据的性能。

（2）强标准（Strong Criterion）：满足弱标准，此外系统能够以明确的符号形式来交流内部更新情况。

（3）特强标准（Ultrastrong Criterion）：满足弱和强标准，此外系统能够以**可有效操作**的符号形式来交流内部更新情况。

任何学习方法，包括：人工神经网络和统计方法，均满足弱标准。由人工智能激发的机器学习方法与强标准相关。最后特强标准要求用户不仅能够理解归纳出的描述，而且无需计算机帮助就可使用这个描述。也就是说，用户记住所归纳出的描述的同时，就能够完成全部所需的相应计算。

从一般意义上讲，本章，乃至于本书，主要与基于知识方法相关，这一算法能够获得易于理解的描述。大多数这类方法是基于操作符号结构的。让我们首先介绍一下认知观点和表示问题，它们均与机器学习方法的重要标识——概念有关。

1.2.1　认知观点

概念（Concept）对机器学习而言，如同化合物对于化学，力场对于物理，数字对于数学，以及知识对于人工智能一样重要。贯穿本书，我们将概念理解为对一组具有某种共同性质而区别于其他对象的对象的抽象表示。

“鸟”、“老虎”、“脊椎动物”、“轿车”、“雨天”、“数学”、“素数”、“白血病”、“星系”、“专制统治者”或“肥沃的土地”都是概念。注意它们之间的边界并不总是很清楚。“鸟”和“素数”之间没有什么大问题，但要想准确定义“专制统治者”的内涵就会相当棘手，因为这个概念是主观的且与上下文有关。其他概念，诸如“数学”或“星系”都有模糊的边界。即使在概念可以准确定义时（如：白血病），但在根据现有数据对一个对象进行正确的分类（如：一个病人）的时候，也可能产生一个困难的问题。

在统计学中，常常使用**聚类**（Cluster）的概念。它的意思相近但却不同。对于一个聚类，统计学家通常是指根据所选定数值的距离（不一定就是欧氏距离）来确定的一组彼此接近的对象。教室中坐在一起的一组学生就是一个聚类。另一方面，“机器学习学生”这一概念就是根据诸如在所指领域具有共同兴趣等的特征来定义描述的。

在现实世界中，概念从来都不是孤立的。相关概念组常常能组织成树状

或图（图 1.2）所表示的泛化层次结构。在层次结构的一个特定层次中，概念通常不相交，但有时也层叠。它们之间的差异有小有大。在一个泛化层次结构中，一个概念不仅可通过低层对象来说明，而且也可以通过求解概念以下任何一级的子概念来说明。例如：概念“鸟”就是“脊椎动物”的一个实例，“老虎”也是一个。

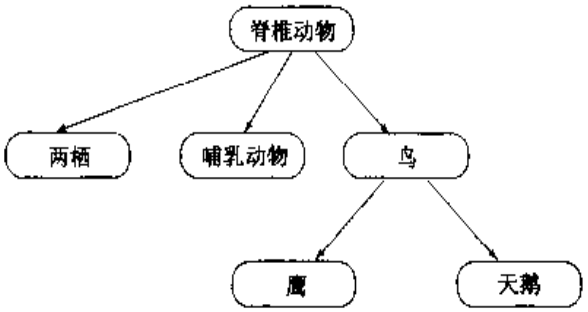


图 1.2　泛化层次结构

与概念之间的相互关系有密切联系的三个重要标识需要简要介绍一下：基础水平效果、典型和上下文相关。

心理学家的发现表明在一个有序的概念层次中（如：图 1.2 中的一个分支），一个层次可理解为**基础的**（Basic）。这就意味着这个层次的概念与其子概念共享许多特征，这些特征能够由可感知的识别项加以描述。

为说明清楚，考虑以下序列：

鹰 → 鸟 → 动物 → 生物

这里基础层次概念就是“鸟”，因为其子概念（“鹰”、“山鸟”、“鸵鸟”等）共享可被传感器检测出的特征（如：翅膀、羽毛、鸟嘴）。注意“动物”子概念——如：爬虫动物、鸟或哺乳动物——不共享这些特征，因此“动物”层次不是基础的，“生物”也一样。

在许多概念层次中都可以发现基础层次概念。经过一些思考，读者将会同意这种看法：

BMW → 轿车 → 交通工具

这里的基础概念就是“轿车”。

基础层次概念通过一些容易识别的特征加以描述，从而使得它们容易被人类学习。低水平概念可作为基础层次概念的细化来理解（如：能唱歌的鸟），而高水平的概念常常被定义为共享某些重要特征的基础层次概念组。

第二个有效方面就是说对于一个特定概念，一个实例的**典型**程度是如何

的。在学习时，实例的典型性起着重要的作用——通过企鹅、鸵鸟和鹅，来说明“鸟”，将很难使学习者很好地理解这个概念。在心理学文献中，度量典型性有两种方法：通过与其他子概念共享特征数目和从超概念继承特征的数目（所继承的、能在实例中发现的特征数越多，实例就越典型）。

第三个方面就是**上下文相关性（Context Dependency）**。在谈及学生时，谈论者心里会有多个概念：来自特定大学的学生、计算机科学学生或来自临近中学的学生。基于他们的知识、年龄和兴趣，他们中的每一个人都有不同的内涵。显然，现实世界概念仅仅在合适的上下文中才是可学习的。

对与心理学和概念获取，记忆和回想的认知方面的相关细节更感兴趣的读者，可以在 Klimesch（1988）发现更深入的内容。

1.2.2　表示问题

在由计算机解决某一任务的任何时候，要提出的第一个问题就是如何将问题翻译为计算项目。在机器学习中，这就意味着如何表示概念、样本和背景知识。为描述概念和样本，需要利用**表示语言**（Representation Languages）。接下来介绍一些在机器学习中遇到的语言。按照复杂度和表示能力，它们包括：零阶逻辑、属性-值逻辑、Horn 子句和二阶逻辑。为避免不必要的计算复杂性，我们仅给出这些语言的直观描述。

从现在起，若样本为真（或被满足），就说一个描述（Description）**覆盖**（Cover）一个样本。因此描述 has_four_legs 覆盖一个狮子，但不覆盖一个鹅。

1.2.2.1　零阶逻辑

零阶逻辑，又称为命题演算，利用代表单个特征的布尔**常量**（属性值）的合集来描述样本和概念。利用数学项，这类描述与以下类似：

$$c \Leftarrow x \wedge y \wedge z$$

它表示一个对象就是概念的一个实例 c，而同时满足条件 x，y 和 z。

为了说明清楚，考虑以下有关 Jane 潜在丈夫的基本描述：

$$can\_marry\_jane \Leftarrow male \wedge grown\_up \wedge single$$

其他连接符包括否定和析取。

虽然零阶逻辑有能力描述简单概念，但读者将会发现描述日常生活中的复

杂概念是较为困难的。换句话说，零阶逻辑具有**低级描述能力**。低级描述能力使得零阶逻辑在机器学习中不能得到广泛的应用，仅能用于说明简单算法。

1.2.2.2 属性逻辑

正规地来说，属性逻辑大致等价于零阶逻辑，但它使用了更丰富、更灵活的标记。基本思想就是通过某些事先定义好的一组**属性**取值来刻画样本和概念，诸如颜色或重量。对零阶逻辑（通过常量的合取来描述概念）的改进之处就是：属性即**变量**，它们可以取各种值。例如："颜色"属性值可以取"红"、"绿"、"蓝"，或以"*"表示任何颜色（连接两个或更多属性值的"或"被称为**内部析取**（Internal Disjunction））。

样本常常表示为一张表，其中每行代表一个样本，每列代表一个属性。因此表 1.1 即列出了吸引年轻企业家的轿车的正例（⊕）与反例（⊖）样本。

表 1.1 概念大型∨（中等∧贵重）的正例与反例

对 象	制 造 地	尺 寸	价 格	类 别
轿车 1	欧洲	大型	可负担得起	⊕
轿车 2	日本	大型	可负担得起	⊕
轿车 3	欧洲	中等	可负担得起	⊖
轿车 4	欧洲	小型	可负担得起	⊖
轿车 5	欧洲	中等	贵重	⊕
轿车 6	日本	中等	可负担得起	⊖
轿车 7	日本	中等	贵重	⊕
轿车 8	欧洲	大型	贵重	⊕

可以考虑布尔量、数值、符号量或混合值属性，并且这些量的取值范围常常受到背景知识的约束。合法取值常常有序或部分有序。直觉上讲，有序值是那些可以被整数所表示的，例如：长度和高度按照某种合适选择的单位所度量。部分有序值是那些可以构成一个层次结构的值。为说明清楚，可以看看图 1.2 中的"动物"变量可能的取值情况。注意"鹰"要比"鸟"更为具体，但与"两栖动物"没有关系。

作为一种描述语言，属性逻辑要比零阶逻辑更为实用，尽管对更严格的数学意义而言，它们具有相同的表达能力。由于这个原因，机器学习研究人员已经相当注意属性-值逻辑，并为有名的 TDIDT 算法（Quinlan，1986）或 AQ（Michalski，1983a）提供了依据。在**变量-值逻辑**中（Michalski，1973a）

为这类描述语言定义了正式的基础。

1.2.2.3　一阶谓词逻辑：Horn 子句

一阶逻辑提供了一个正规框架，可用于对象及其各部分，通过对象与/或其部分间关系进行描述和推理。一阶逻辑的一个重要子集就是 Horn 子句。一个 Horn 子句包括一个头部和一个主体，如以下祖父母定义描述所示。

$$\text{grandparent}(X,Y):-\text{parent}(X,Z),\text{parent}(Z,Y)$$

上述描述表示若可找到 Z，使得 X 是 Z 的父母，且 Z 是 Y 的父母，那么 X 就是 Y 的祖父母。符号“:-”左边部分称为子句的**头部**，符号“:-”右边部分称为子句的**主体**。逗号表示合取，且 X，Y，Z 均是普遍定量的变量。

单词“祖父母”和“人”被称为**谓词**，且括号中的变量称为**参数**（Arguments）。参数个数一般是任意的，但对于给定谓词则是确定的。若所有谓词均只有一个参数，语言就变为属性-值逻辑。若所有谓词均只有零个参数，则语言就变为零阶逻辑。

Horn 子句构成一个先进表示语言，可帮助实现非常复杂的描述。它们构成编程语言 Prolog 的基础，并用于学习系统，如 FOIL（Quinlan，1990）。

1.2.2.4　二阶谓词逻辑

二阶逻辑是建立在以下思想基础上，该思想就是谓词名称本身也可认为是变量。因此，例如，

$$\text{模式}\quad p(X,Y):-q(X,XW)\wedge q(Y,YW)\wedge r(XW,YW)$$

可以实例化为：

$$\text{brothers}(X,Y):-\text{son}(X,XW)\wedge\text{son}(Y,YW)\wedge\text{equal}(XW,YW)$$

通过以下替换

$$\Theta=\{p=\text{brothers},q=\text{son},r=\text{equal}\}$$

另一个可能的实例化就是

$$\text{lighter}(X,Y):-\text{weight}(X,XW)\wedge\text{weight}(Y,YW)\wedge\text{less}(XW,YW)$$

其相应的替换为：

$$\Theta=\{p=\text{lighter},q=\text{weight},r=\text{less}\}$$

因此子句框架保持不变，仅仅是谓词名称发生变化。这个思想的合理性在于：一组概念常常共享可能描述的相同结构。**二阶逻辑模式**用于保存最成功的结构以帮助概念的搜索。然而这种表示语言是相当复杂的，且很少使用。

一个例外就是 de Raedt（1992）所介绍的程序 CIA。

1.2.2.5 明确约束的语言

基于逻辑的表示语言有时非常丰富和灵活，以致将它们使用到机器学习时，其计算复杂度无法度量。因此，通常的做法就是引入各种约束，诸如子句中有限数目的谓词及其参数，或排除递归定义。

在一个子句中变量出现有限次数意味着子句主体中的变量数目不容许超过预设的阈值。例如：仅有出现在子句头部中的那些变量可以出现在子句主体中，反之，没有出现在头部的一个变量方可出现在主体中。还可给出类似的一些约束。

另一个约束就是从谓词参数中排除**函数**。这会成为很严重的限制，因为通常一个参数不一定就是一个简单变量，可能是一个计算式、复杂算术或逻辑表达式，或 n 元函数。函数的出现将会大大增加可能描述的空间。

最后，一个重要的约束可以**排除递归描述**（Recursive Descriptions）。采用一阶逻辑表示时，**递**归描述的能力常常通过**祖先**（Ancestor）定义来加以说明：

$$\text{ancestor}(X,Y):\text{-parent}(X,Y).$$

$$\text{ancestor}(X,Y):\text{-parent}(X,Z),\text{ancestor}(Z,Y).$$

上述描述就是：若 X 是 Y 的父母，或如果可以找到 Z，使得 X 是 Z 的父母且 Z 是 Y 的祖先，那么 X 就是 Y 的祖先。

虽然递归描述概念有时不可避免，但是它们倾向于将学习者的任务复杂化，且难以理解。因此有时语言定义就会明确禁止递归（Michalski，1980）。

1.2.2.6 其他表示

理论上，某些额外逻辑表示模式可能成为概念特征描述的候选。其中，Minsky 的框架（Minsky，1975）在人工智能领域中就很受欢迎。抽象数学结构，诸如**语法**（Grammars）或有**限自动机**（Finite Automata），也在推荐之中，因为它们拥有受到数学家广泛研究的结构性质，且对于某些应用程序很有用。

然而，这些表示获得了机器学习中相对有限的注意。读者仅需要记住，即使逻辑模式当前在相关文献中非常流行，将来研究中也需考虑其他的选择。

1.3　泛化空间的搜索

假设选择了一个表示语言且学习者想要从一组正例与反例样本中学习一个概念。即使描述是基于属性-值逻辑的，但全部可识别概念的空间也是非常大的。10 个属性，每个可取 5 个值，将会构成 5^{10}=9,765,625 个可能的向量。这些向量的任意子集都对应一个概念，这就意味根据这些属性可以定义 $2^{9,765,625}$ 个概念。在更复杂的语言中，这个数目甚至会增加得更快，即使背景知识能够限制表示空间的大小。

为处理问题的可计算性，学习者大多将两种有力的技术结合在一起，即归纳与启发搜索。相关技术的讨论以及对于学习者所使用的基础推理原理的分析，将是以下各节的任务。

1.3.1　学习的归纳本质

想像一个外星科学家在您城镇附近着陆，希望研究外星生物。科学家具有一些基本的语言知识，但需要磨合。这就是为什么作为接触的异类，他对您首先提出的问题就是："鸟"是什么。

开始时，您会把"山鸟"作为概念的正例。但是简单记住山鸟所有的特征，对于识别其他同类别的鸟是不够的。显然，这个例子的泛化是必需的。但泛化究竟有多大能力？为确定其局限性，外星科学家将需要一个反例样本，某个不是鸟的东西。相应地，您会推荐"狗"。显然，狗与山鸟之间的差别就是狗没有翅膀。（假设开始时，外星科学家对容易识别的特征感兴趣。）

为检查是否所有有翅膀的生物都是鸟，外星科学家将会问"苍蝇"是否也属于同一类。显然它们不是，这就说明了拥有翅膀对合适区分正例与反例太泛化了。对该描述进行细化是必要的。这可以通过取示例的一个特征并增加到当前描述中来加以实现。山鸟的一个显著特征——狗和苍蝇所没有的——就是黄色的鸟嘴。外星科学家可能又会认为乌鸦不是鸟，因为它没有黄色的鸟嘴。

鉴于这个特征仅在某些鸟中有，就说明这时的描述又过于特殊，因此还需要适当的泛化。因此科学家除去鸟嘴黄色的要求，从而简单得出鸟是有翅膀和鸟嘴的生物。这个描述将帮助您识别"麻雀"为"鸟"的一个正例。

这个简单的故事说明了一个基本的机器学习策略。让我们用更为科学的

词汇将这个过程重新说明一次。在第一个示例（山鸟）后，对于鸟的看法过于特殊。外星科学家仅知道此单一示例最特殊的描述而不知道其他事情。这个描述成为一组最特殊描述（用 S 表示）中的第一个元素。此时对最特殊描述进行任何泛化都是可能的。仅在第一个反例（狗）加入后，学习者可能对泛化进行某种限制，因此获得一组**最泛化**描述（用 G 表示）来正确地覆盖正例但不覆盖反例。从这个集合 G 中，科学家选择“有翅膀”作为最吸引人的特征。

下一个反例（苍蝇）表明描述**过于泛化**，翅膀不能区分山鸟（正例）和苍蝇（反例）。因此集合 G 必须进行细化。另一方面，下一个正例（乌鸦）将用一个更特殊的描述丰富集合 S，从而需要通过把“有翅膀”替换为“有翅膀”和“有黄色鸟嘴”来完成另一次泛化。

总而言之，当一个新正例样本加入时，就应用泛化来处理集合 S。相反，当一个反例加入集合 G 时，就进行细化。这个原则构成了一系列称为版本空间（Version Space）算法的技术的基础，这些算法是建立在概念描述当前版本空间不断缩减这一思路上的。该方法是由 Mitchell（1982）发明并发表的，有关的参考目录请参见 Hirsh（1990）。更早的一种方法将概念学习视为单个假设（而不是两组假设）的一系列泛化和细化，在 Winston（1970）一篇著名论文中也有详细介绍。

为满足我们的需要，算法的细节及其许多派生的东西并非很重要。上述故事意在展示这样的事实，即概念学习可以被认为是描述空间的搜索，基本的搜索操作有：**泛化（Generalization）**与**细化（Specialization）**。

本章的许多内容将直接或间接地提及这类操作。例如，可以通过将常量转换为变量，或除去一个条件而泛化获得 Horn 子句描述。因此子句：

$$p(X,Y):-q(X,2),r(Y,2)$$

可以泛化为

$$p(X,Y):-q(X,Z),r(Y,Z)$$

或泛化为

$$p(X,Y):-q(X,2)$$

细化可理解为补充操作。在这个意义上，一个 Horn 子句可以通过将变量变为一个常量，或增加一个文字量到子句中来进行细化。

选择合适的搜索操作符是（除了选择表示语言）一个学习程序设计者的

重要任务。Michalski（1983）就是系统地总结这些泛化操作的先驱者之一。

1.3.2　穷尽搜索

概念学习的普遍框架就是搜索由学习者表示语言所容许的描述空间。这种方法的优点非常明显：人工智能研究人员业已广泛研究并普遍理解这些搜索技术。以下术语在搜索技术定义中是必需的。

一般而言，一个搜索过程将探寻搜索空间中的每个状态：

- **初始状态**（Initial State）是搜索的起点。在机器学习中，初始状态常常对应最特殊的概念描述，也就是正例本身。
- **终止条件**（Termination Criterion）就是要达到的目标。满足终止条件的状态称为**最终状态**（Final States）。在机器学习中，终止条件要求描述覆盖所有正例且不包括任何反例。
- **搜索操作符**（Search Operator）实现从一个状态转到另一个状态的**搜索**。在机器学习中，这些操作符大多是概念描述的泛化和/或细化。
- **搜索策略**（Search Strategy）确定什么条件下，操作符可以用于什么样的状态。

系统搜索的两个基本策略就是：深度优先搜索和广度优先搜索。若将所有可能的状态空间视为一个有向图，就可以较容易地解释这些搜索。有向图的结点代表每个状态，边缘代表搜索操作符。

深度优先搜索（Depth-first Search）中，操作符应用于初始状态 S_1，进入一个新状态 S_2。若 S_2 不被认为是最终状态，那么再次应用一个操作符处理状态 S_2，达到状态 S_3。若这样无法到达一个新状态，那么就无法发现最终状态，这时系统**回溯**（Backtrack）到前一个状态，并应用一些其他的操作符。若不可能，系统将回溯更远，直到发现一个状态容许应用某些操作符为止。若无法发现这样的状态，搜索终止。图 1.3 描述了这种搜索的原则，矩形框中数字表示被访问状态的顺序。

广度优先搜索（Breadth-first Search）构成了一个补充方法。首先，应用所有操作符，一个接一个地应用于初始状态，然后测试所获得的状态。若其中的一些状态被认为是最终状态，算法停止。否则，应用操作符到随后的各状态，然后再到之后的状态，如此等等，直到满足终止条件。图 1.4 描述了这种搜索的原则。

需要注意的是，与深度优先搜索不同的是，广度优先搜索认为是无回溯的，从而使其变得比较简单。另一方面，搜索者需要保存许多中间状态，这将会加重系统的开销。

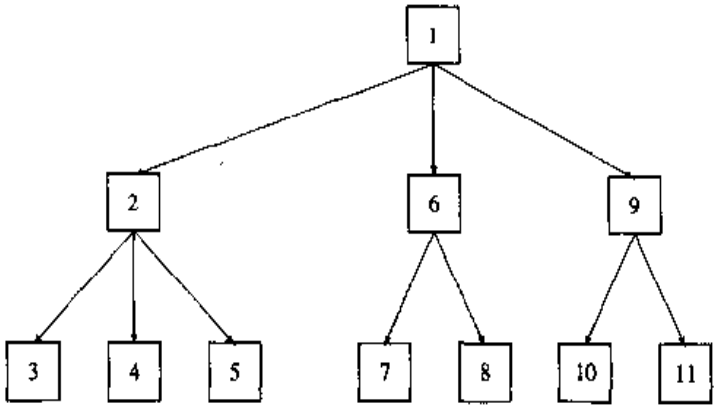


图 1.3　深度优先搜索

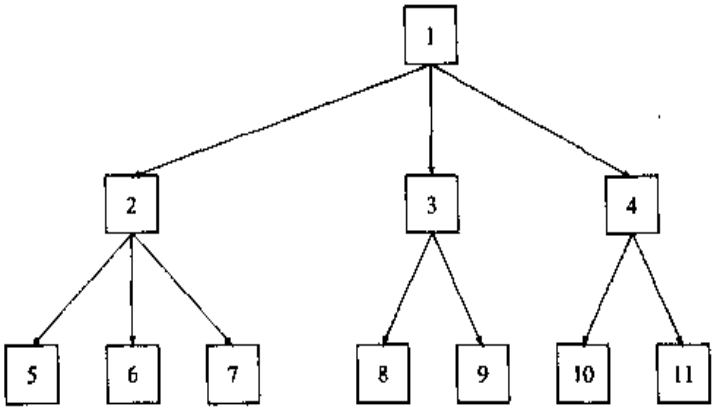


图 1.4　广度优先搜索

1.3.3　启发式搜索

基本的搜索技术对于大搜索空间而言效率较低，这时需要考虑**启发式**（Heuristics）来指导搜索。启发式任务就是确定哪些可用操作可使搜索最接近最终状态。这需要评价函数来评估每个所获状态的取值。假设这时已给出评价函数。

1.3.3.1　最好优先搜索算法

（1）设初始状态为**最优**状态，并设**当前状态**集包括这个状态；

（2）若最优状态满足给定终止条件，那么停止搜索（最优状态就是搜索的结果）；

（3）应用所有可用操作符处理最优状态，从而创建了一组新状态，并将其加入到当前状态集中；

（4）评估当前所有状态，确定哪个是最优的，然后转至（2）。

这个算法与广度优先搜索不同，因为它总是仅扩展最有希望的状态，希望以此来加速搜索。其代价就是可能陷入评价函数局部最优的窘境。

算法的实例说明如下。

示例

学习者试图从一组 8 个正例与反例中，获得概念描述。这些样本所涉及的属性值如表 1.2 所示。“⊕”表示“正例样本”，而“⊖”表示“反例样本”。假设有两个操作符：“通过增加一个合取将当前描述细化”和“通过增加一个析取将当前描述泛化”。

表 1.2　概念学习中的正例与反例样本

样本	at1	at2	at3	分类
e1	*a*	*x*	*n*	⊕
e2	*b*	*x*	*n*	⊕
e3	*a*	*y*	*n*	⊖
e4	*a*	*z*	*n*	⊖
e5	*a*	*y*	*m*	⊕
e6	*b*	*y*	*n*	⊖
e7	*b*	*y*	*m*	⊕
e8	*a*	*x*	*m*	⊕

设初始状态为“任何描述”。应用细化操作符（这时泛化无意义）将会产生以下描述：at1=*a*，at1=*b*，at2=*x*，at2=*y*，at2=*z*，at3=*m* 和 at3=*n*。其中，at2=*x* 和 at3=*m* 不覆盖任何⊖，有可能获得合理评价函数的最高值，据此，选择 at2=*x* 作为当前最优的描述。

由于表 1.2 中某些⊕目前尚没有被覆盖，学习者将试图应用搜索操作符处理当前最优状态。通过减少⊕被覆盖的数目所进行的细化操作符，只会使情况变糟。然而通过泛化描述成为 at2=*x* ∨ at2=*y*，覆盖的⊕数目将会增加。假设评价函数证实这个描述要优于 at2=*x*。

新描述覆盖所有⊕，但另一方面它也覆盖两个⊖。因此，下一步，描述就细化为 at2=*x* ∨ [(at2=*y*) ∧ *X*]，其中 *X* 表示任何以下合取：at1=*a*，at1=*b*，

at3=m 和 at3=n。

在新状态中，最好的就是 at2=x ∨ [(at2=y) ∧(at3=m)]。它覆盖了所有⊕但一个⊖都没有被覆盖，搜索结束。

最优搜索也许需要太多的存储空间，因为它要存储产生的所有状态。一种更经济的方法就是**集束搜索（Beam-search）**，它任何时候仅保留 N 个最好状态。

1.3.3.2 集束搜索算法

（1）设初始状态为**最优**状态；

（2）若最优状态满足某个终止条件，停止；

（3）若当前状态数目大于 N，仅保留 N 个最优状态并删除其他的；

（4）应用**搜**索操作符处理最优状态，将新增状态加入到当前状态**集**中；

（5）评估所有状态，转到（2）。

集束搜索算法的一个常见例子就是 N=1，有时被称为**爬山（Hill-Climbing）**算法。这个名字就是强调其与爬山类似，努力发现通向顶峰的最短路径且常常选择最陡的路径。

对搜索技术更详细情况感兴趣的读者，可参考人工智能文献，例如：Charniak 和 McDermott（1985），或 Bratko（1990）。

1.4 学习经典任务

在解释了泛化与细化操作原理，以及一些基本搜索技术之后，我们将继续介绍两个基本的学习原理，也就是分而治之学习法和 AQ 方法。这两个方法对于理解更先进的方法都是很重要的。

1.4.1 分而治之学习法

这种方法的实质非常简单：整个样本集分成许多更容易处理的子集。在属性逻辑中，根据属性取值进行划分以便一个子集中的所有样本都可共享给定属性的特定值。

在表 1.2 中，属性 at1 将 8 个样本的**集**合分为由 at1=a 划分的子集和由 at1=b 划分的子集。类似地，at3 也可将集合分为两个子集，一个由 at3=m 划分，另

一个由 at3=*n* 所划分。最后，at2 也可将整个集合划分为三份，它们分别由 at2=*x*，at2=*y* 和 at2=*z* 所确定。

这个原理构成受欢迎的归纳决策树（参见 Breiman 等人，1984，和 Quinlan，1986）的基础，用它们的首字母缩写分别命名为 TDIDT（Top-Down Induction of Decision Trees）或 ID3。借助合适定义的评价函数，以下描述 TDIDT 的算法将从如表 1.2 所示的数据中，获得如图 1.5 所示的决策树。读者可以检查决策树中的概念描述是否与数据表相符合。例如：实例 *e*1 具有 at2=*x*；沿着最左边分枝向下，最后到达标记为⊕的方框。实例 *e*3 具有 at2=*y*；沿着中间分枝向下，对属性 at3 进行测试；具有 at3=*n* 取值，沿右边分枝向下，最后到达标记为⊖的方框。

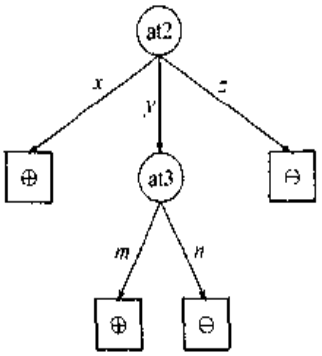


图 1.5　一棵决策树

注意相应树可以改写为以下逻辑表达式：

$$(\text{class}=\oplus) \leftarrow (\text{at2}=x) \vee [(\text{at2}=y) \wedge (\text{at3}=m)]$$

$$(\text{class}=\ominus) \leftarrow (\text{at2}=z) \vee [(\text{at2}=y) \wedge (\text{at3}=n)]$$

任何未来实例将会根据决策树或两个规则进行分类。不满足这些规则（如，若具有 at2=*w* 的实例，其中取值在训练过程中没有出现）的实例分类，则是根据实例描述与规则之间的距离进行分类。也可以给出“我不知道”作为答案。

1.4.1.1　TDIDT 算法

S 为样本实例集合。

（1）找出“最优”属性 at；

（2）将集合 *S* 分解为若干子集 S_1，S_2，…，而且子集 S_i 中的样本实例具有 at=v_i。每个子集构成决策树中的一个结点；

（3）对于每个 S_i，若 S_i 中所有样本实例属于同一类别（⊕或⊖），然后创建一个决策树并将其标记为相应类别。否则（转至（1））对 $S=S_i$ 完成类似过程。

在所有子集均被标记或未标记集合不能再往下分属性时（这种情况中，某些树叶将覆盖两个类别的实例），算法结束。

1.4.1.2 如何发现“最优”属性

现有由属性取值所描述的两个类别，任务是发现前面所介绍算法步骤（1）中的最优属性。一个可行的标准是基于由不同属性所产生的每个子集中⊕和⊖的数目。

经过思考，读者将会同意我们需要一个满足以下要求的函数：

（1）函数在所有子集均是同质的取得最大值，也就是 S_i 中所有实例都是⊕，或 S_i 中所有实例都是⊖。这种情况下，有关属性值信息将足以确定一个实例样本是正例或反例；

（2）函数在每个子集中一半样本实例为正例而另一半样本实例为反例时，取得最大值；

（3）函数应在接近极端点（100%正例且 0%反例，或反过来）时最陡；而在 50%—50%区域时最平坦。

数学家知道信息最大化时另一个重要数值——**熵**（Entropy）最小。熵描述了数据中随机性、“无结构性”和混乱程度。在这种情况下，子集 S_i 的熵可以通过以下公式计算获得：

$$H(S_i) = -p_i^+ \log p_i^+ - p_i^- \log p_i^-$$

其中 p_i^+ 是从子集 S_i 中随机选择样本为正例时的概率，其值可以通过相对频率 $p_i^+ = \frac{n_i^+}{n_i^+ + n_i^-}$ 计算获得；类似地，p_i^- 是从子集 S_i 中随机选择样本为反例时的概率，其值可以通过相对频率 $p_i^- = \frac{n_i^-}{n_i^+ + n_i^-}$ 计算获得。这里 n_i^+ 为子集 S_i 中正例的数目，而 n_i^- 为子集 S_i 中反例的数目。

设属性 at 取值将 S 划分为若干子集 S_i，i=1,2,…,K，然后子集 S_i 系统的熵为：

$$H(S,at) = \sum_{i=1}^{K} P(S_i) \cdot H(S_i)$$

其中 $H(S_i)$ 就是子集 S_i 的熵；$P(S_i)$ 是一个实例属于子集 S_i 的概率，它可以通过子集 S_i 在 S 的相对大小进行估计：$P(S_i) = \frac{|S_i|}{|S|}$。

通过按照 at 进行分解所获得的信息增益可以由随后熵的减少来度量：

$$I(S,\text{at}) = H(S) - H(S,\text{at})$$

这里 $H(S)$ 是 S 的**事先熵**（在分解之前），而 $H(S,\text{at})$ 则是由 at 取值所产生

子集系统的熵。

让我们通过从表 1.2 中数据建立决策树来说明这些公式的具体应用。

由于集合 S 的 8 个实例样本中包括 5 个正例和 3 个反例，因此系统 S 的事先熵为：

$$\begin{aligned} H(S) &= -p^{+}\log p^{+} - p^{-}\log p^{-} \\ &= -(5/8)\log(5/8) - (3/8)\log(3/8) \\ &= 0.954\text{bits} \end{aligned}$$

注意这个熵接近其最大值（0.954≈1），因为正例的数目与反例大致相同。若正例数目要比反例多得多（或反过来），那么我们通过猜测它总是正例的正确概率就非常大，这与小信息熵相对应。

通过不同属性而产生的不同划分所产生的熵为多少？例如，属性 $at2$ 能取三个不同值 x，y 和 z，对于每个值，可以获得相应的三个子集 S_x，S_y 和 S_z 的熵：

$$at2:\quad H(S_y) = -(2/4)\log(2/4) - (2/4)\log(2/4) = 1\text{bit}$$

$$H(S_x) = -1\cdot\log 1 - 0\cdot\log 0 = 0\text{bit}$$

$$H(S_z) = -0\cdot\log 0 - 1\cdot\log 1 = 0\text{bit}$$

由此按照加权累加获得整体熵：

$$H(S,at2) = (3/8)\cdot 0 + (1/8)\cdot 0 + (4/8)\cdot 1 = 0.5\text{bits}$$

类似地计算出其他属性的熵，从而获得以下信息熵：

$$I(S,at2) = H(S) - H(S,at2) = 0.954 - 0.500 = 0.454\text{ bits}$$

$$I(S,at1) = H(S) - H(S,at1) = 0.954 - 0.951 = 0.003\text{ bits}$$

$$I(S,at3) = H(S) - H(S,at3) = 0.954 - 0.607 = 0.347\text{ bits}$$

显然，$at2$ 产生最大的信息增益（0.454bits），因此应被选做树的根。这就是图 1.5 中树是如何创建的。

利用熵仅仅是多种可能之一。人们提出了若干属性选择标准。在 Breimann 等人（1984）和 Mingers（1989a）的论文中可以发现一些这样的标准。

1.4.1.3　概率估计

利用相对频率估计 p_i^{+} 和 p_i^{-} 的概率远不理想，因为仅当样本集合足够大时估计才是可靠的。然而，在决策树的产生过程中，随着不断分解，样本数目快速减少。假设仅留下两个样本，两个均是正例，然后一个基础学习者将会得出反例的概率为 100%，这确实是不对的。

由于这个原因，已提出了概率估计的改进方法。例如：m-估计（Cestnik，1990）根据以下公式计算概率：

$$p_{\oplus}=\frac{n_{\oplus}+mp_a}{N+m}$$

这里$n_{\oplus}$是正例的数目，N 是子集中所有数目（$N=n_{\oplus}+n_{\ominus}$），p_a是正例的事先概率，m 是估计的参数。在样本噪声较大的情况下，m 应设置较大，对于较小噪声，m 设置较小。在任何情况下，用户或专家，在了解领域情况时，应推荐相应的 m 和 p_a 的设置。

m-估计的一个特殊情况就是拉普拉斯连续定律（或简单拉普拉斯估计），对于两个类别：正例与反例，如：

$$p_{\oplus}=\frac{n_{\oplus}+1}{N+2}$$

这个公式的匹配性可通过表 1.3 的简单实验来加以说明。假设您扔硬币，每次获得两个可能结果之一：正面和反面。表 1.3 中给出根据相对频率所获正面的概率（百分比），以及根据拉普拉斯估计所获的概率。（十分清楚，拉普拉斯估计给出更多符合实际的值。）

表 1.3　扔硬币的正面概率（百分比）

扔的次数	1	2	3	4	5
结果	正面	正面	反面	正面	反面
相对频率	100	100	67	75	60
拉普拉斯估计	67	75	60	67	57

1.4.1.4　树的修剪

一些缺点会使决策树的应用变得有点问题。其中之一就是**过度逼近**（Overfitting）。在属性值或类别标记是错误的时候，一个树的分枝（以一个类别标记结束）可能会根据有噪声数据样本而产生。显然这个分枝，或一些决策树的测试将会产生误导。第二，若属性数目较大，树就会包含随机特征的测试，这些测试与正确分类实际上是无关的。因此，例如，若我们对一辆轿车的发动机原理感兴趣，那么颜色就无关紧要。尽管如此，只要是许多轿车具有的属性就可以出现在树中，也就是说，基于燃烧的发动机碰巧也是红色。最后，非常大的树较难解释，且用户可将其想像为一种黑箱表示。

对于所有的原因，通过如图 1.6 所示的方法对所产生树进行修剪或许是有

益的。

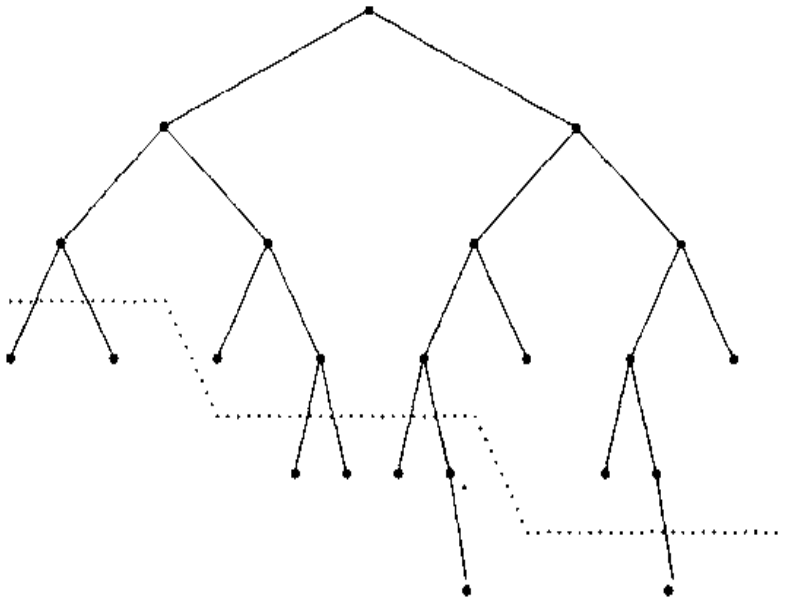


图 1.6　修剪决策树

原则上，可能有两种修剪方法：在线修剪和事后修剪。**在线修剪**本质就是在由样本集划分所引起的信息增益小于一定阈值时停止树的增长。**事后修剪**方法是在树产生后，删除其中某些分枝。

修剪的一种常用方法，称为**最小错误修剪**（Minimal-error），它是由 Niblett 和 Bratko（1986）所设计的。这项技术旨在使得修剪树到新样本的整个期望分类错误最小化。为了这个目的，对树中每个结点估计其分类错误。在叶结点，利用估计一个新对象落入相应叶结点被分错类的概率方法来估计错误。假设 N 是叶结点最后的样本数，e 为相应叶结点中被分错类的样本数目。Niblett 和 Bratko（1986）利用拉普拉斯估计（e+1）/(N+k)（其中 k 是所有类别的数目）来估计期望错误。Cestnik 和 Bratko（1991）介绍利用 m-估计来替换拉普拉斯估计获得了更好的结果。对于决策树中的非叶结点，其分类错误按照相应结点各子树分类错误加权累加来进行估计。其权重是根据从结点传到相应子树样本的相对频率来进行估计。非叶结点错误估计被称为回传错误。此时非叶结点的错误分类可以根据子树删除情况进行估计，这时子树就成为一个叶结点。若此时错误估计小于回传错误，子树将会被删除。这一子树删除过程从树的底层开始，然后上传直到回传错误大于“静态估计”。

Mingers（1989b）和 Esposito 等人介绍了一些其他的修剪方法。

1.4.1.5 简化树的其他动机

图 1.7 说明树的另一类简化，这次目标就是执行一种**构造归纳**（Michalski，1983a）。学习系统努力利用教师提供的属性构造的逻辑表达式来创造新属性。某棵子树在决策树中多处出现时，构造归纳可能更为有效——参见图 1.8 和图 1.9。

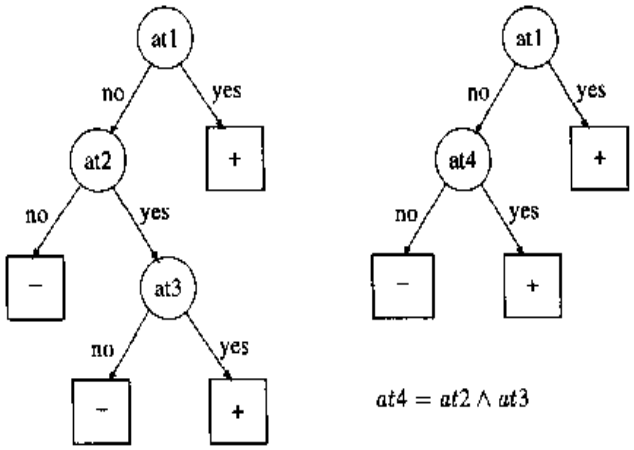


图 1.7 决策树中的构造归纳，构造新属性 $at4$

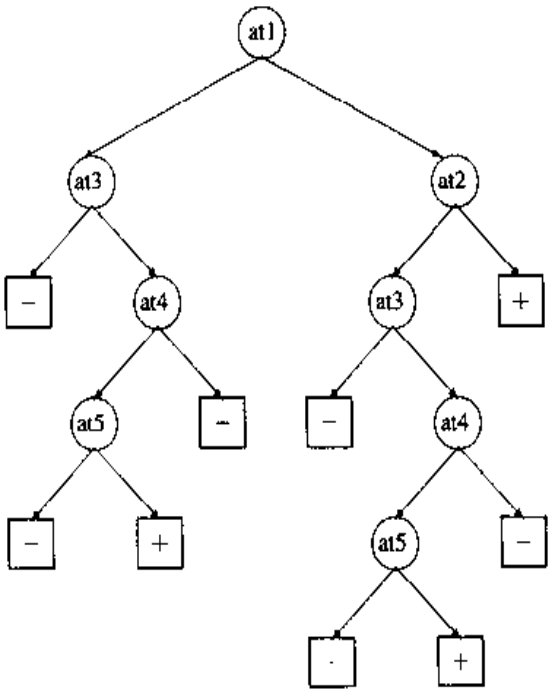


图 1.8 决策树中的复制问题

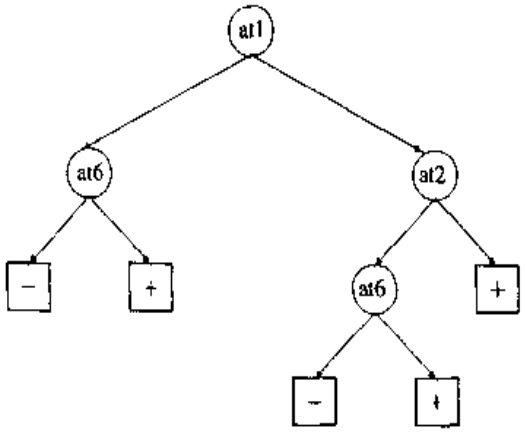


图 1.9　来自图 1.8 中的树的简化版本

1.4.1.6　处理数值数据

至今为止，分析一直被限制在符号属性中。然而决策树也可以从数值属性中归纳得出。一种可能就是提供额外的步骤：**数值属性二值化**（Binarization），这就意味把相应数值阈值化为若干区间对，以便于作为符号处理。图 1.10 描述了从数值数据中构造的决策树。每个结点，根据阈值 T_i 对相应的属性值进行测试。

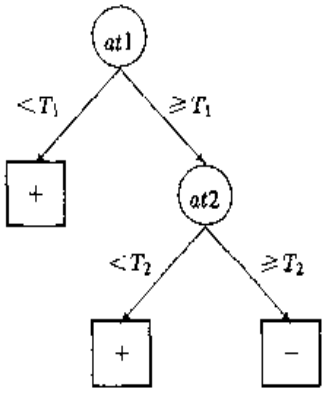


图 1.10　从数值属性归纳获得的决策树

在取值范围中阈值的位置，还是可以由熵来确定的。假设属性 *at*1 需要离散化。首先根据 *at*1 取值和分类值对所有样本进行排序。在图 1.11 所示的情况中，正例和反例的分类值将 80 个样本分为 4 个区域（现实情况下，区域的数目可能更大）。候选划分位于区域边界上。然后每个划分的信息增益按照先前介绍进行计算。选择信息增益最大的划分（更详细和更严格的有关机制讨论，请参见 Fayyad 和 Irani，1992）。

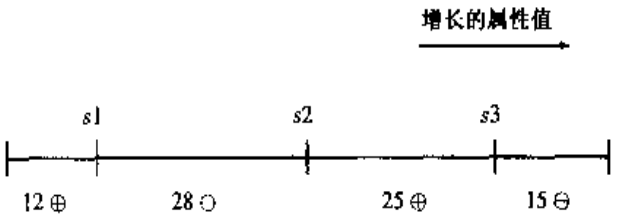


图 1.11　根据一个属性值进行排序；*s*1，*s*2，*s*3 是候选分解点

TDIDT 算法的数值版本说明如下。

数值 TDIDT 算法

（1）利用熵值找出每个数值属性的最佳分解点；

（2）确定可使熵最大化的最佳分解属性，将样本集合按照相应属性分为两部分；

（3）若终止条件没能满足，对每个子集不断递归重复上述过程。

注意对于每个新子树，必须重新计算划分。最佳划分位置在不同样本子集中可能也会不同。

1.4.2 主动覆盖：AQ 学习

AQ 学习是基于通过连续产生决策规则来主动覆盖训练数据的思想。该方法已从最初由 Michalski（1969）首先提出的一个算法，发展成为一个大的系列算法家族，并由 Michalski（1973b）根据机器学习目的进行了修改。

其本质就是搜索一组规则（属性-值对的合取，通常，任意谓词）以覆盖所有正例但不覆盖任一个反例。AQ 算法不再划分样本集合，而是一步一步对所选择的正例描述（称为种子）进行泛化。这样做容许规则在需要的时候逻辑上可有交叉。

1.4.2.1 基本原理

以下介绍了一个简化算法版本的原理。假设目标就是找出最小一组决策规则来描述特定的概念。决策规则将采用以下形式：

$$\textbf{if } A_1 \text{ and } A_2 \text{ and } \ldots \text{ and } A_n \textbf{ then } C$$

这里 C 为概念。条件 A_i 采用常见的属性-值形式 $at_i=V$，或更为通用的形式：$at_1=v_1 \vee v_2 \vee v_3 \ldots$，其中一个属性能取若干值之一（由“内部”析取连接）。

AQ 算法（简化版）

（1）将所有样本分为子集 PE（正例）和 NE（反例）；

（2）随机选择或通过指定，从 PE 中选择一个样本，称为**种子**（Seed）；

（3）找出描述种子实例的最大泛化的一组规则。泛化的限制则是通过 NE 中的对象来实现。由此获得的规则被称为**星**（Star）；

（4）根据某些**选择标准**（Preference Criterion），在星中选择最佳规则；

（5）若这条规则，与先前产生的规则一起，覆盖 *PE* 中的所有样本；那么就停止。否则在未覆盖的 *PE* 样本中发现另一个种子，转到（3）。

通过一个特别的**星泛化过程**（Star Generation Procedure）（Wnek 等人，1995）完成步骤（3）。步骤（4）所完成的**规则选择标准**（Rule Preference Criterion）反映了当前问题的需要。为了这个目的，可以是各种基本标准结合体，诸如需要最大化规则覆盖的正例数目，最小化所涉及的属性数目，最大一般化（覆盖正例数除以规则所覆盖的所有样本数），最小化属性-值度量成本等。也可以利用决策树学习中所使用的选择标准，诸如熵、增益率等。算法还可能构造一组决策规则，各规则间具有不同的关系。规则可以逻辑上相交、逻辑上不相交或线性有序（在连续顺序中需要评估它们）。

下一个样本描述了这个算法的简单版本。

例子

假设样本所涉及的属性为 *at*1，*at*2 和 *at*3，样本实例如表 1.4 所示，可视化如图 1.12 所示。在表 1.4 中，每行对应属性-值向量。行对应由⊕标记的正例和由⊖标记的反例。假设选择标准是以覆盖最大可能正例数目，以及规则能够相交为主，从以上样本中获得的有关规则的程序包含以下步骤。

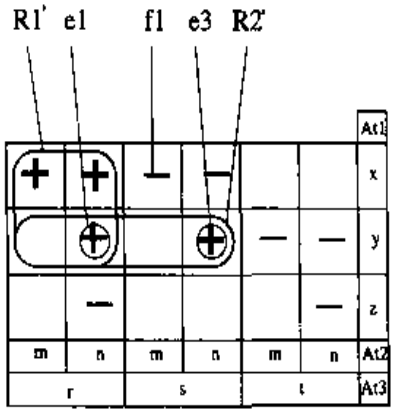


图 1.12　表 1.4 中样本的可视化描述

表 1.4　一个样本训练集合的说明

样本		*at*1	*at*2	*at*3	分类
PE	*e*1	y	n	r	⊕
	*e*2	x	m	r	⊕
	*e*3	y	n	s	⊕
	*e*4	x	n	r	⊕
NE	*f*1	x	m	s	⊖
	*f*2	y	m	t	⊖
	*f*3	y	n	t	⊖
	*f*4	z	n	t	○
	*f*5	z	n	r	⊖
	*f*5	x	n	s	⊖

选择第一种子：*e*1

选择第一个反例：*f*1。创建种子 *e*1 的星（即基于 *e*1 的最大泛化描述的集合）。通过创建不覆盖 *f*1 的 *e*1 所有描述集开始。它们如下所述。

$R1$：$(at3 = r \vee t)$

$R2$：$(at1 = y \vee z)$

$R3$：$(at2 = n)$

然而，每个描述覆盖一些反例。因此这些规则要进行细化以排除这些反例，这通过将这些反例的否定与当前规则进行结合来实现。应用吸收定律后所获得的结果如下：

$$R1'：(at1 = x \vee y) \& (at3 = r)$$

$$R2'：(at1 = y) \& (at3 = r \vee s)$$

这就是 $e1$ 的星。假设选择标准推荐从星中选择覆盖最多正例的规则。因此选择 $R1'$（$R1'$ 覆盖三个样本而 $R2'$ 仅覆盖两个）。下一步就是从没有覆盖的正例中选择一个新种子。仅存在一个这样的样本：$e3$

选择下一个种子：$e3$

再次，根据新种子确定其星，产生两个规则。一个覆盖与 $R2'$ 相同的样本。

由于没有未覆盖样本，所选择的 $R1'$ 和 $R2'$ 构成一个完整且一致的概念描述，以使得假设的选择标准最佳。

$$R1'：(at1 = x \vee y) \& (at3 = r)$$

$$R2'：(at1 = y) \& (at3 = r \vee s)$$

基于 AQ 算法的学习系统可以较容易地融合背景知识，因为这类知识常常由决策规则来表示。但是我们这里不会论及这种特性，因为利用背景知识对于基于谓词逻辑学习系统更为典型，稍后将会讨论。

1.4.2.2 *两层方法*

在基于 AQ 的方法中，输出具有决策规则的形式。为处理上下文不准确和噪声数据，提出了两层方法（Two-tiered）（详细情况请参见 Michalski，1990）。这个方法有助于处理上下文敏感和改进 AQ 系统中的可理解性。

主要思想就是将概念描述分为两部分：**基本概念表示**（Base Concept Representation，BCR）包含明确地存储于学习者记忆中的概念特征；**推理概念解释**（Inferential Concept Interpretation，ICI）包含一组推理规则以用于识别阶段。

为了说明清楚，BCR 可以包含以下产生式规则：

if A_1 and A_2 and ... and A_n **then** X

ICI 包含解释规则：

在 $A_1, A_2, \ldots, A_n$ 中至少有 3 个条件必须满足。

两层方法中一个非常简单的例子就是 TRUNC 方法，它包括以下一般步骤：

（1）利用 AQ 获取初始规则集；

（2）确定每个规则的“重要性”，简单的度量就是规则所覆盖的正例数目，仅保存 BCR 中最重要的规则；

（3）定义正确识别 BCR 中规则未覆盖样本的 ICI 过程。

在利用归纳出的表示进行识别时，根据 ICI，新样本被赋予 BCR 中提供“最佳匹配”规则的类别。为此目的提出了一个更灵活的匹配过程。

对两层方法更多细节以及部分实现感兴趣的读者，可参见 Michalski（1990），Zhang 和 Michalski（1991），Bergadano 等人（1992），或 Kubat（1996）文献。

1.4.3　学习算法评估

在过去 20 年，机器学习的研究人员已经提出了许多学习算法，因此对它们进行评估和分类的标准是必不可少的，如表 1.5 所列。

表 1.5　ML 算法评估

标　准	解　释
准确率	正例和反例正确分类的比例
有效性	所需要的样本数，计算的易操作性
鲁棒性	抗噪声，对付不完全性的能力
特别需求	增量，概念偏移
概念复杂性	问题表示（样本和背景知识）
透明性	人类用户的可理解性

或许最重要的标准就是**准确率**（Accuracy）。概念学习的一般动力就是正确识别未来实例，学习的成功可以直接通过测试样本正确分类的百分比来加以度量。假设学习者被要求对一组的 200 个样本分类，其中 100 个是正例，另 100 个是反例。若学习者正确分出 80 个正例和 60 个反例，那么其正确率就是（80+60）/200=0.7，即 70%。

有时知道系统可以正确识别多少个正例更为重要，同时反例可能就不太

关键（或反过来）。这种情况下，我们提出两类错误的差异：反向分类正例（忽视错误）和正向分类反例（强调错误）。这些错误能够告诉用户学习获得的概念描述是过于特别或过于一般。在过于特别的情况下，学习者倾向于更容易错误分类测试样本中的正例（与反例相比）。在过于一般的情况下，学习者更容易错误分类测试样本中的反例（将它们分类为正例）。

理想情况下，学习者应建立一个假设（内部概念描述），它是一致的（没覆盖任何反例）和完全的（覆盖所有正例）。图 1.13 描述不一致和不完全情况。在由两个数值属性描述的空间中（一个由水平轴表示，另一个由垂直轴表示），所发现的概念是在分割矩形为上下两部分的斜线以下的阴影部分。由椭圆表示的概念是不完全的，因为它没有覆盖概念的整个区域。矩形 ABCD 是不一致的，因为它覆盖了一部分反例区域。

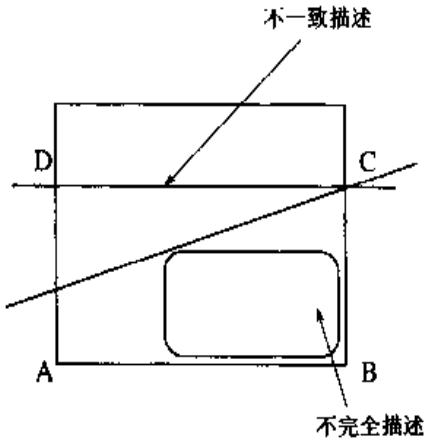


图 1.13　不一致和不完全情况的描述

分类准确率仅仅是评估机器学习算法的标准之一。学习应该要有效率——能够在最小学习样本数目下获得一定水平的准确率。教师并不能提供足够多的样本，而且，学习快的能力也是智能的标志。此外**计算能力**也是需要考虑的——计算机需要多少时间以获得一个好的假设。

另一个标准就是与归纳出概念描述的**可理解性**相关。产生的概念具有可理解性常常是重要的，因此用户可以从中学习到有关应用领域的一些新东西。这种描述也可直接为人所用，并加以理解作为他们知识的提高。这个标准在归纳出描述应用于基于知识系统时也是适用的。基于知识系统的行为应是透明的。可理解性的标准典型地将人工智能中机器学习与其他形式的学习区别开，包括神经网络和统计方法。如本章 1.2 节所提及的，Michie（1988）对于可理解性相关的标准做了进一步研究。Michalski（1983）对这方面早期工作也做了介绍。

学习中的一个很严重的复杂问题就是：样本和/或类别标记中存在噪声。提供属性值的测量设备可能不准确，或调节不准；由教师提供的属性值可能

过于主观，一个偶然事件可能会损坏数据，某些数据可能会丢失。在样本提交给机器之前，可能会有遇到一些意外事件。因此需要**噪声鲁棒性**和**遗失信息鲁棒性（遗失属性值）**。然而这并不是教条！某些学习任务的特点就是存在噪声和遗失信息，其他一些任务的特点就是样本完美。具体应用时必须确定合适的标准。

应用的具体条件应该认真对待。例如：用户要求学习者应该在线从一个接一个的样本流中获得知识（与一开始就提供所有样本的情况相反）。想像在一个初始概念描述形成后，提出一个新样本。在传统批量处理算法中，这就意味着对所有数据重新运行整个学习过程。这种行为几乎不能称为智能。学习者应有能力精炼先前的知识，像人一样，进行**增量学习**。

增量学习是在概念漂移或进化（Widmer 和 Kubat，1993，1996）中特别重要的内容。在某些领域中，概念的含义时时发生变化。如："时髦的服装"或"民主"一词就是其中之一。学习者应能够按照人的方式进行变化。

最后，设计者必须考虑表示问题，这就意味着尊重描述样本和背景知识的语言。读者已经看到了各种表达能力不同的表示语言。例如：TDIDT 算法属于属性-值逻辑，下一节将要讨论的更先进的系统应能够从谓词表示的样本中学习概念。

1.5　如何利用谓词逻辑

前面已经提过，属性-值语言是非常有用的但也有它们的局限性。即使可以给出许多描述，带谓词描述的对象或其部件间关系的逻辑描述一定更强大。考虑如图 1.14 所示的家庭关系的背景知识（或简单起见，人名用"1"，"2"，...来代替"John"，"Bill"，...）。这里利用单个谓词 parent（X，Y）来描述家庭关系，例如，parent（1，2）意义是指 1 是 2 的父母。

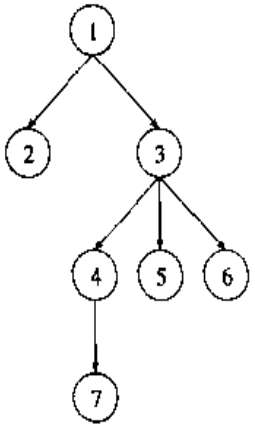


图 1.14　家庭关系

可以按以下关系对家庭树进行编码：

parent = {(1,2) (1,3) (3,4) (3,5) (3,6) (4,7)}

假设将这些关系作为背景知识，系统想要学

习 grandparent（X，Y）的含义。假设，更进一步，教师提供以下概念的正例：

grandparent（1，4）
grandparent（1，5）
grandparent（1，6）
grandparent（3，7）

再次，这些更容易利用以下关系进行编码：

grandparent = {（1，4）（1，5）（1，6）（3，7）}

在人 1，…，7 中其他家庭关系被认为是 grandparent 反例。

当然，这些关系也可以用属性值来进行描述。例如：每个可能的人对（X，Y）可以用一个布尔属性来表示，其真值就是由谓词关系 Parent（X，Y）的真值所确定。然而这种描述是很累赘且不灵活的。

高层次描述语言需要更复杂的算法，下一节任务就是要介绍有关谓词逻辑学习的一些思路。这些介绍以众所周知的原理开始，逐步过渡到更为复杂的技术。

最常用的谓词逻辑学习方法就是归纳逻辑编程（简称 ILP），它目前是机器学习中一个备受关注的研究分支。有关更多的情况，请参见 Muggleton（1992）或 Lavrač 和 Džeroski（1994）所著。

1.5.1 从关系中学习 Horn 子句

假设要求您写一个学习程序，它可以从家庭关系样本中学习 grandparent 概念。概念采用 Horn 子句来描述。您会采用什么学习策略呢？

试图用最可能简单的方式来描述概念，您会采用这种语言，该语言仅容许使用同时在主体和头部中出现过的参数。假设概念描述是一组以下形式的 Horn 子句：

$$C_1 :- L_{11}, L_{12}, \ldots, L_{1m}$$
$$C_2 :- L_{21}, L_{22}, \ldots, L_{2m}$$
$$\ldots$$

这里谓词 C_i 头部的一个子句，文字量 L_{ij} 构成了子句主体。逗号将主体中的文字量隔开，以指示它们是由合取连接在一起。每个文字量代表一个关系，它包含 $n \geqslant 0$ 的参数。

回想分而治之学习法或AQ方法，学习者从相对通用的描述开始，该描述仅包含一个属性，然后通过增加更多条件逐步进行细化。那这里为何不采用相同的原则呢？

由于添加一个文字量到一个子句主体中，与添加一个条件到AQ中的一条决策规则中，或在一个决策树底部增加一个结点，具有相似的细化效果。一个好策略就是从一个仅包含头部的子句开始，然后通过添加文字量到其主体中逐步进行细化。

由于除了parent外，背景知识中没有其他谓词，而且仅容许出现在头部的变量出现在主体中，采用这种语言定义grandparent的可能子句就是：

grandparent(*X*,*Y*):- parent(*X*,*Y*)
grandparent(*X*,*Y*):- parent(*Y*,*X*)
grandparent(*X*,*Y*):- parent(*X*,*X*)
grandparent(*X*,*Y*):- parent(*Y*,*Y*)

遗憾的是，没有一个子句覆盖任何正例样本。显然应该放松限制以容许没有出现在子句头部的一个参数。可以构成四个这类文字量：parent（*X*，*Z*），parent（*Y*，*Z*），parent（*Z*，*X*）和parent（*Z*，*Y*），假设系统选择以下选项。

grandparent(*X*,*Y*):- parent(*X*,*Z*)

让我们检查哪个三元组（*X*,*Z*,*Y*）满足这个子句，子句头部代表一个正例，而它代表一个反例。注意有7^3=343个可能的三元组（*X*，*Z*，*Y*）。

⊕：（1，2，4）（1，2，5）（1，2，6）（1，3，4）（1，3，5）（1，3，6）（3，4，7）（3，5，7）（3，6，7）

⊖：（1，2，1）（1，2，2）（1，2，3）（1，2，7）（1，3，1）（1，3，2）（1，3，3）（1，3，7）（3，4，1）（3，4，2）（3，4，3）（3，4，4）（3，4，5）（3，4，6）（3，5，1）（3，5，2）（3，5，3）（3，5，4）（3，5，5）（3，5，6）（3，6，1）（3，6，2）（3，6，3）（3，6，4）（3，6，5）（3，6，6）（4，7，1）（4，7，2）（4，7，3）（4，7，4）（4，7，5）（4，7，6）（4，7，7）

仔细观察可以发现：正例三元组包括所有4个正例，反例三元组包括17个反例：（1，1）（1，2）（1，3）（1，7）（3，1）（3，2）（3，3）（3，4）（3，5）（3，6）（4，1）（4，2）（4，3）（4，4）（4，5）（4，6）（4，7）。

我们说子句grandparent(*X*,*Y*):-parent(*X*,*Z*)是不一致的，因为它覆盖反

例。通过适当选择子句的细化可以减少不一致。利用对其增加一些容许的文字量可以实现这一点。让我们试试以下子句：

grandparent（*X*，*Y*）:- parent（*X*，*Z*），parent（*Z*，*Y*）

这个子句覆盖以下三元组（*X*，*Z*，*Y*）：

⊕：（1，3，4）（1，3，5）（1，3，6）（3，4，7）

⊖：没有

由于覆盖所有正例而没有覆盖一个反例，学习者对此子句满意，就此停止。

若可以，可以换其他文字量代替 parent（*Z*，*Y*）来添加上去？考虑下一个子句：

grandparent（*X*，*Y*）:- parent（*X*，*Z*），parent（*Y*，*Z*）

尽管子句没有覆盖任何反例，利用子句就有问题，因为它也没有覆盖任何正例。显然应有一个适当的标准来确定需要添加什么文字量到子句中。

让我们检查更复杂的概念 ancestor。基于 7 个人，他们的家庭关系如图 1.14 所示，提供以下正例：

ancestor = {（1，2）（1，3）（1，4）（1，5）（1，6）（1，7）（3，4）（3，5）（3，6）（3，7）（4，7）}

搜索最佳文字量（涉及与头部一样的参数）获得以下描述：

ancestor（*X*，*Y*）:- parent（*X*，*Y*）

这个子句是一致的，因此这里不需要细化。系统将保存它（因为描述是不完全的）并试图发现替换子句来覆盖在第一个子句之外的正例。结果可以解释为：至少子句之一应覆盖样本，若它是概念的一个正例的话。不断重复这一过程，可以确定以下三个子句：

ancestor（*X*，*Y*）:- parent（*X*，*Y*）

ancestor（*X*，*Y*）:- parent（*X*，*Z*），parent（*Z*，*Y*）

ancestor（*X*，*Y*）:- parent（*X*，*Z*），parent（*Z*，*W*），parent（*W*，*Y*）

即使这些子句覆盖所有学习样本，但我们知道它们也不是完全的，因为它们仅覆盖四代人。熟悉逻辑编程的读者，可能会推荐一个递归描述：

ancestor（*X*，*Y*）:- parent（*X*，*Y*）

ancestor（*X*，*Y*）:- parent（*X*，*Z*），ancestor（*Z*，*Y*）

一个学习系统能发现这个描述仅利用概念的初始理解来定义谓词 ancestor 。这就是递归原理。开始时，学习者利用谓词 parent。在确定第

一个子句后，学习者就可以利用谓词 ancestor。

上述描述的过程形成了 FOIL 系统的内核，FOIL 系统是由 Quinlan（1990b）开发的。以下算法描述了相应的方法。

FOIL 算法

（1）通过定义代表待学习概念名字的子句头部来初始化该子句，其子句主体为空；

（2）当子句覆盖反例时，发现一个“好的”文字量加入到子句主体中；

（3）除去子句所覆盖的所有样本；

（4）将子句加入到新出现的概念定义中，若还有任何没有被覆盖的正例，则转到（1）。

这一时刻的遗留问题就是如何发现一个需要加到子句中的“好的”文字量（算法的步骤（2））。为了这个目的，FOIL 利用一个与决策树归纳类似的信息标准。

T_i^+ 表示由 $L_1, L_2, ..., L_{i-1}$ 合取所覆盖的正例数目。T_i^- 表示由 $L_1, L_2, ..., L_{i-1}$ 合取所覆盖的反例数目。那么由这个子句所覆盖样本中包含一个正例所提供的信息就是：

$$I_i = -\log(\frac{T_i^+}{T_i^+ + T_i^-})$$

在添加一个新文字量 L_i，这个信息就变为：

$$I_{i+1} = -\log(\frac{T_{i+1}^+}{T_{i+1}^+ + T_{i+1}^-})$$

文字量 L_i 的成功程度可由两个因素来度量：

（1）剩余信息 I_{i+1}（越小越好）；

（2）T_i^{++} 为在添加 L_{i+1} 文字量后的子句所覆盖的正例数目（越多越好）。

因此，FOIL 度量成功的计算定义如下：

$$Gain(L_i) = T_i^{++} \times (I_i - I_{i+1})$$

观察 FOIL 是如何将前一节所介绍方法结合起来的，非常有益。原则上讲，程序执行 AQ-算法，试图利用一组子句覆盖正例空间。在内层循环中，由一个类似分而治之方法所使用函数控制的过程，来实现对子句的逐步细化。

1.5.2 反转归并

现在回到基于搜索的学习概念，我们可以较容易看出 FOIL 利用了两个搜索操作：泛化 add_a_clause 操作和细化 add_a_literal 操作。提供两个补充操作：delete_a_clause 和 delete_a_literal 分别用于细化和泛化操作，由此获得一阶逻辑的基本学习内容。以下四步将说明这些操作，利用这些操作将把子句 x:-a，b 逐步改变为子句 x:-d，e。

- 增加一个子句：$x:-a,b \Rightarrow \begin{Bmatrix} x:-a,b \\ x:-c,d \end{Bmatrix}$
- 删除一个子句：$\begin{Bmatrix} x:-a,b \\ x:-c,d \end{Bmatrix} \Rightarrow x:-c,d$
- 增加一个文字量：$x:-c,d \Rightarrow x:-c,d,e$
- 删除一个文字量：$x:-c,d,e \Rightarrow x:-d,e$

这一小节的主要任务就是介绍这个基本集是如何通过以下归纳搜索操作进行扩展的。

- 识别：$\begin{Bmatrix} a:-b,x \\ a:-b,c,d \end{Bmatrix} \Rightarrow \begin{Bmatrix} a:-b,x \\ x:-c,d \end{Bmatrix}$
- 吸收：$\begin{Bmatrix} x:-c,d \\ a:-b,c,d \end{Bmatrix} \Rightarrow \begin{Bmatrix} x:-c,d \\ a:-b,x \end{Bmatrix}$
- 内部构造：$\begin{Bmatrix} a:-v,b,c \\ a:-w,b,c \end{Bmatrix} \Rightarrow \begin{Bmatrix} a:-u,b,c \\ u:-v \\ u:-w \end{Bmatrix}$
- 外部构造：$\begin{Bmatrix} a:-v,b,c \\ a:-w,b,c \end{Bmatrix} \Rightarrow \begin{Bmatrix} a:-v,u \\ a:-w,u \\ u:-b,c \end{Bmatrix}$

上述所有操作均可以从**归并原理**（Resolution Principle）中推出。该原理在人工智能中非常流行。用两个 C_1 和 C_2 来表示谓词的析取；用 l 表示任意谓词。归并原理是推理性的且描述如下：

若 $(C_1 \vee l)$ 为真且 $(C_2 \vee \neg l)$ 为真，那么 $(C_1 \vee C_2)$ 也为真。

换种方式来说，假设两个析取表达式。其中之一包含文字量 l，另一个包含其否定 $\neg l$。两个表达式的析取，其中 l 和 $\neg l$ 已被删除，也是为真的且被称为可归并的。图 1.15 描述这个原理。

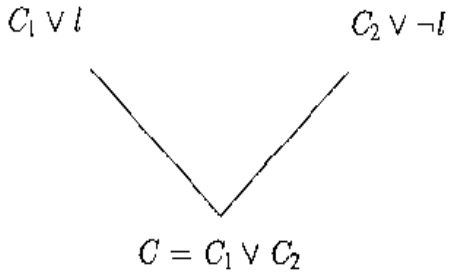


图 1.15　归并原理

非常有趣的是，归并的基本模式能够反转。知道了可归并和初始串中的任何一个，我们可以构造另一个初始串。根据可用子句是否包含谓词 l 的肯定或否定形式，我们分别介绍**识别**（Identification）或**吸收**（Absorption）。这两种方式的整个推导模式如图 1.16 所示。我们将只介绍识别，吸收的推导与之类似。内部与外部构造操作也可以通过稍微复杂的过程推导出来，具体内容这里就不介绍了。

假设给定两个子句：a:-b,x 和 a:-b,c,d。前者称为“初始”而后者称为“可归并”。任务是发现未知子句，它将和初始子句一起，产生可归并的子句，为简单起见，假设谓词均没有参数。

我们都知道公式 A:-B 可以改写为 $A \vee \neg B$，我们将两个子句转换为 $a \vee \neg b \vee \neg x$ 和 $a \vee \neg b \vee \neg c \vee \neg d$。两个子句均包含子串：$a \vee \neg b$。此外可归并子句还包含子串：$\neg c \vee \neg d$，它们可以从未知子句中继承下来。反过来，初始串包含谓词 $\neg x$，因此它的否定 x，就期望出现在未知子句中。将可归并的和初始的相应内容拼接在一起，就会获得字符串：$\neg c \vee \neg d \vee x$。转换 Horn 子句，字符串将变为：x:-c,d。这个新创建的子句将替换可归并子句。

遗憾的是，在现实世界的应用中任务会变得很复杂。因此在反转归并的情况下，学习者将不得不发现谓词 l 和 $\neg l$ 中合适的参数需要替换，以便它们是兼容的。例如：谓词 p_1=parent(john,bill)和 p_2=parent(X,eve)仅在第一个谓词替换为 θ_1 ={john/X,bill/Y}，第二个谓词替换为 θ_2 ={eve/Y}的时候，才是兼容的。有关更多相关问题的详细讨论，读者可参见 Muggleton（1991）文献。

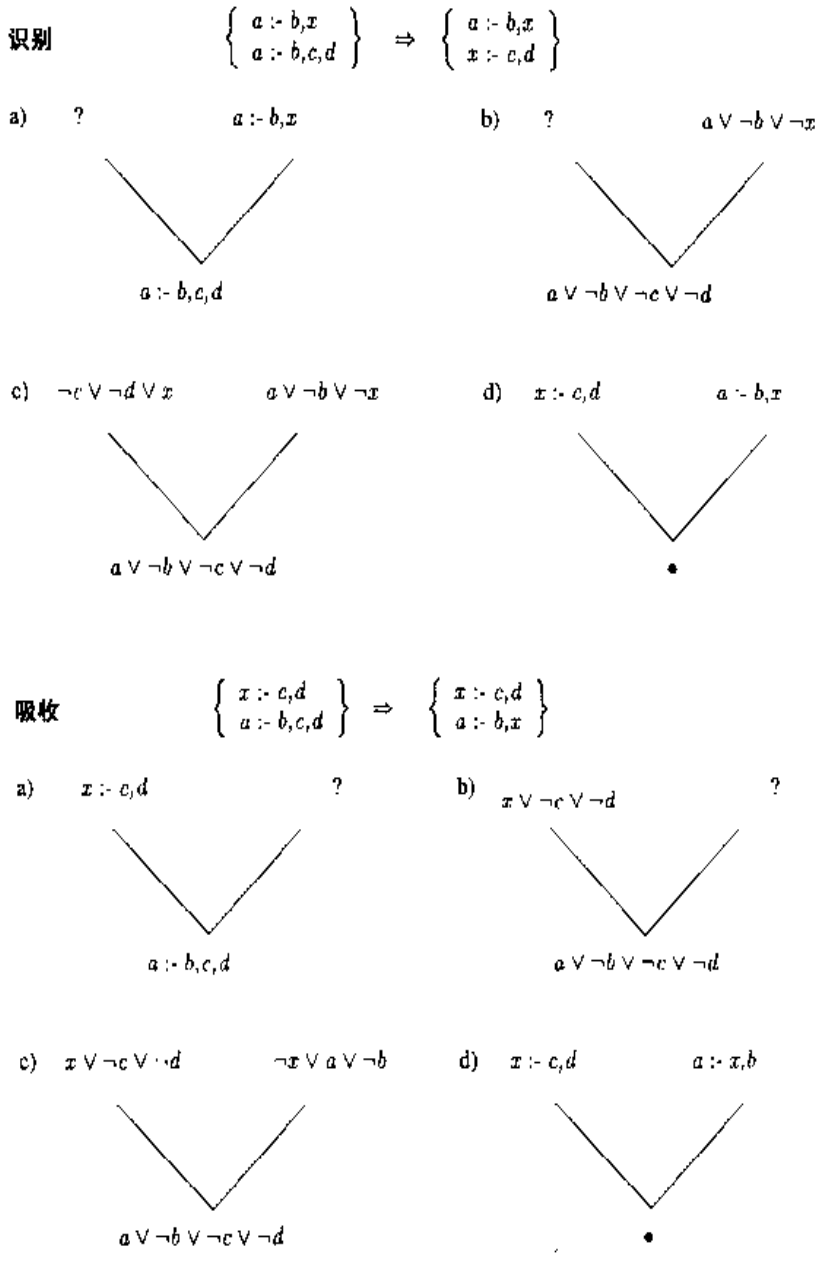


图 1.16　识别和吸收操作的推导

1.5.3　理论修正

有时可以通过获得**背景知识**的主体来指导学习过程。为说明清楚，假设背景知识包含部分有关家庭关系（与图 1.14 所示相似），有关个人性别的附加信息利用谓词 male 和 female 来描述。假设 Jack 是 Bill 的父亲，学习者将期望获得谓词 parent 的定义，这在之前的背景知识中并没有出现。这里所描述的方法大致建立在构成 CLINT 系统核心的算法的基础上，可参见 de Raedt

(1992)。

学习者从某个非常受限的语言开始，例如：开始约束要求子句主体中每个文字量容许包含在头部出现的常数和变量作为其参数，如：p(X,Y):-q(X,Y),r(X)。在 father(jack，bill)情况下，这就意味着系统搜索所有文字量，除了 jack 和 bill 它们不包含其他参数。发现这些文字量后，系统将它们用合取连接起来。

为说明清楚，假设系统背景知识包含以下谓词：

⋮

parent(jack,bill)

parent(tom,jack)

parent(tom,eve)

parent(eve,bill)

male(tom)

male(jack)

male(bill)

female(eve)

painter(bill)

singer(jack)

⋮

若没有其他谓词包含 jack 或 bill 参数，根据上述的简单语言构造的概念，就会获得以下描述：

father(jack,bill):-parent(jack,bill),male(jack),male(bill),painter(bill),singer(jack)

发现这个具体子句后，学习者通过将常数变为变量来对其进行泛化，因此获得被称为**初始子句**的内容：

father(X,Y):-parent(X,Y),male(X),male(Y),painter(Y),singer(X)

显然，这种子句构造方法相当落后，甚至仅看看“发明出”的子句就会发现它有一些问题：jack 是 bill 父亲的事实与 bill 是男性没有关系的，与他的专长也没有多大联系。为解决这个问题，CLINT 的作者为学习者提供了通过与用户对话来精炼对初始概念的描述能力。

在对话中，学习者逐个检查每个谓词，通过创建新样本和要求用户对其

分类来检查它们的必要性。例如，问题：father(tom,jack)为真吗？

若用户回答肯定，就表明文字量 painter（Y）是多余的（jack 是歌唱家，tom 仍是其父亲）。

下一个问题应检查 *Y* 是否为男性。知道 eve 是女性，学习者发现背景知识中文字量 parent（tom,eve）并要求用户回答以下问题：

father(tom,eve)为真吗？

若用户回答肯定，就表明文字量 male（Y）是多余的。另一方面，问题：

father（eve,bill）为真吗？

若用户回答否定，就表明文字量 male（X）不能从子句中除去。

显然，在证实过程中，初始子句完全可以改变，甚至所有文字量均可从主体中删除。另外没有发现任何初始子句。在两种情况下，系统不断减少某些约束，例如，一个加在谓词参数上的限制，那么主体谓词容许包含仅一个不在头部出现的参数，如以下情况：

grandparent(X,Y):- parent(X,Z),parent(Z,Y)

按照这种方式，系统泛化概念描述，期望覆盖所有尚没有被先前描述所覆盖的正例（由系统创建并提交给用户）。

当然，描述变得过于泛化时，它就会覆盖反例。这种情况下，需要采取相应措施以纠正这种不利情况。CLINT 的实现中包含建立反例的解释树，识别覆盖反例的问题子句 *c*，从知识库中删除 *c*，重新泛化产生的知识结构以确保之前被 *c* 覆盖的正例仍被覆盖。

有关 CLINT 的更多情况请参见 de Raedt（1992）。

1.5.4 构造归纳

让我们来注意有关确定学习的合适表示空间的问题，也就是与当前问题相关的属性或谓词。在标准方法中，学习者分析样本实例，用事先定义好的一组属性或谓词描述它们，利用描述语言的操作产生所期望的概念描述。Michalski（1991）将其称为简单的归纳**经验**。TDIDT 最简单版本和 AQ 算法均可实现这种学习，这里的概念是利用描述样本实例的一组属性子集来描述的。

一些方法，诸如反转归并，学习算法本身构造新谓词来帮助学习进程。这个方法需要交互式的归纳过程。由于概念是由机器发明的，用户（拥有更

多知识使谓词更有意义）被要求认可新谓词并给它命名。带爪的食肉动物可接受并命名为：predators。具有黄色皮肤的大动物可能不会成为一个有用的概念，用户不会使用它，因为他们相信两个特征一起出现仅仅出于巧合。

由此，应介绍**基于类比**学习中的构造归纳。虽然类比问题稍后会讨论，CIA 系统所实现的**二阶逻辑模式**思想（参见 de Raedt，1992）就属于构造归纳的范畴。

系统的本质包括存储典型谓词表达模式，诸如：

$$p(X,Y):- q(X,XW), q(Y,YW), r(XW,YW)$$

这里不仅参数 *X*,*XW*,*Y* 和 *YW*，而且谓词 *p*，*q* 和 *r* 均代表变量。因此先前的模式，通过适当的替换，可以实例化成以下子句（提供各自的替换）。

lighter(*X*,*Y*):- weight(*X*,*XW*).weight(*Y*,*YW*),less(*XW*,*YW*)

⊖={*p*/lighter,*q*/weight,*r*/less}

same-color(*X*,*Y*):- color(*X*,*XC*),color(*Y*,*YC*),eq(*XC*,*YC*)

⊖={*p*/same-color,*q*/color,*r*/eq}

brothers(*X*,*Y*):- son(*X*,*XP*),son(*Y*,*YP*),eq(*XP*,*YP*)

⊖={*p*/brothers,*q*/son,*r*/eq}

二阶逻辑模式在构造归纳中的应用十分直接。在 CIA 设置中，系统发现在适当替换后，模式主体成为某些子句主体的子集而且这些子句头部是未知的。为说明清楚，模式：

$$p(X,Y):-q(X,XW),q(YW,Y),r(XW,YW)$$

能够变成子句的一个子集：

:-male(F),male(c),parent(F,M1),parent(M2,C),eq(M1,M2)

通过以下替换：

⊖={*q*/parent,*r*/eq}和 *p*={*X*/F,*Y*/*C*,*XW*/M1,*YW*/M2}

实例化模式就是：

p(F,C):-parent(F,M1),parent(M2,C),eq(M1,M2)

若提示，用户将会确认新子句，并为谓词 p 命名为 grandparent。

作为结论，与对各种表示语言规律的深入研究相结合，构造归纳原则通常被认为是一个非常重要而且很有意义的研究课题。

1.6 人工发现

人工发现是机器学习一般思考方式的指导性演示说明。这里值得简要介绍一下，即使它与后续章节的内容不直接相关。

至今我们的兴趣集中在**监督**学习上，这里，学习者努力从由教师事前分类好的样本中形成一个概念描述。本节沿另外路径介绍**无监督**学习，这里的任务就是从未分类对象中产生一个概念层次结构。

实际上，这是科学家（生物学家）几个世纪以来一直做的工作，研究诸如脊椎动物的分类，哺乳动物或鸟的子分类等。这种层次结构和分类的用处是显然的：一个被识别为某个特定类别的对象将继承这个类别的一般性质。被告知马是一个哺乳动物，我们立刻就知道动物是否产蛋，是否会飞，或其皮肤是否带有皮毛或羽毛。

一些传统的统计技术，诸如**聚类分析**（Cluster Analysis）会完成一些相关任务。图 1.17 中的圆点代表由两个属性 x 和 y 所描述的对象，对象集可以分为两组，这两组很容易通过利用相似性且相对简单的算法发现，其中相似性可通过对象间的数值距离来度量。遗憾的是，不是每种相似性都可以通过数值来度量。确实，猫与长颈鹿之间的距离是否比狗与大象之间的要大呢？即使这些距离可以转换为数字，但任何这样的转换也是困难的和主观的。

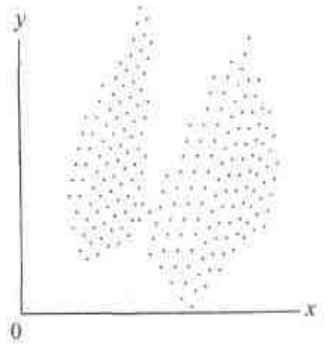


图 1.17　聚类分析的传统任务

为理解另外一个重要问题，首先考虑图 1.18 所描述的任务。这里，对象已经按照某种方式事先排列好了，这种方式可进行概念描述，依赖特定的上下文还可提供若干解释。显然，传统的基于距离的聚类分析对这些数据很难产生合理的结果，而对人类而言，这个任务是相当简单的，搜索隐藏在一组对象中的概念属于被称为**概念形成**（Concept Formation）这一学科所研究的范畴。

更进一步，不仅希望发现概念，而且希望发现定义概念间关系的定律，以便实现创造一个基于计算机系统在诸如化学或生物学等学科来帮助人类研究人员的野心。即使期望实现人工科学家或许过于乐观，但是已经开发出了一些惹人注目的系统，它们能够解决简单发现任务的问题。

以下我们用一节介绍概念形成，另一节介绍自动发现。

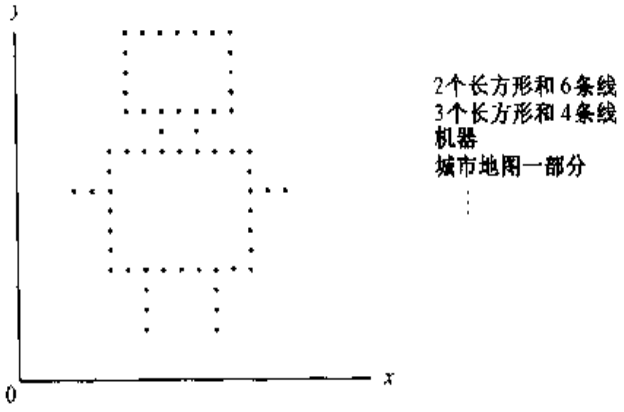


图 1.18　机器学习的概念发现任务

1.6.1　概念形成

Gennari 等人（1989）将无监督的概念学习分为两个不同子领域：概念发现（Concept Discovery），从头获取概念；增量概念形成，从一个样本流中逐步形成概念。

1.6.1.1　通过概念聚类实现概念发现

概念聚类被推荐为聚类的一种新形式，其中聚类不再仅仅是拥有数值相似性的实体集合。而且，聚类被理解为代表一个概念的对象组。概念聚类不仅产生聚类，而且产生相关概念的描述。CLUSTER 系统（参见 Michalski 和 Stepp，1983）就采用与 AQ 同样的种子-星方法，并事实上被认为就是未分类样本的领域的拓展。

图 1.19 描述了一个概念发现的简单任务。8 个未分类样本由 3 个属性描述。属性 *at*1 是符号量；属性 *at*2 是整数值；属性 *at*3 取可分解为三个符号值的整数值。背景知识为每个属性提供了类型和范围，并定义了 *at*3 的分解。

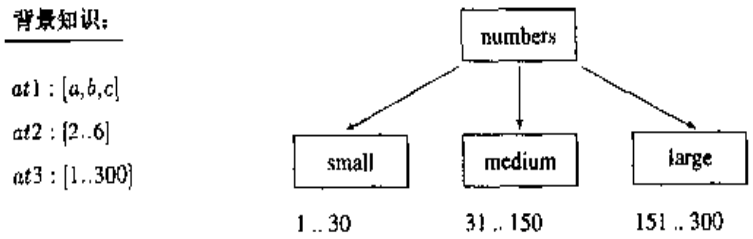


图 1.19　概念发现的简单任务

Michalski 和 Stepp 的思想就是学习者选择 k 个种子，将它们作为 k 个不同聚类的代表。在一个简化版本中，处理过程可用以下算法来描述：

Cluster 算法

（1）选择 k 个种子，其中 k 为用户定义的一个参数。

（2）建立 k 个星，每个星可理解为一个种子最一般描述的集合，种子泛化的限制就是其他种子。

（3）从每个星中选择一条规则，要求在产生的规则集中的每条规则与其他规则之间具有最小逻辑相交，并且这些规则的逻辑合取覆盖最大的样本实例。

（4）若还有没有被覆盖的样本实例，找出最“适合”它们的规则，精炼这些规则以便它们合在一起覆盖所有样本实例，且它们逻辑上不相交。属于规则相交的样本实例重新进行分布，使每个样本实例仅被惟一一个规则所覆盖。这时每个规则代表一组样本集，从每个集合中选择一个新种子。

（5）对于新种子重复上述过程，只要每次获得的求解能够对上次有所改进，就不断重复整个过程。对不同的 k，例如 k=2，3，…，7 不断重复，以确定最高“质量”的求解，根据不同标准，诸如一个聚类中规则的简单性和稀疏性（通过规则覆盖样本被每条泛化规则所覆盖的程度来度量，Michalski 和 Stepp，1983）来确定质量。

让我们将这个算法应用到如图 1.19 所示的数据中。为简单起见，假设数值取值已被用符号值“small”，“medium”和“large”所替换，依据图 1.19（通常 Cluster 本身会提出最合适的数值聚类）。算法将大致会完成以下步骤（k 假设为 2）。

随机选择 2 个种子，如为 $e1$ 和 $e5$。它们的描述为：

des($e1$) : ($at1$=a)&($at2$=2)&($at3$=large)

des($e5$) : ($at1$=c)&($at2$=5)&($at3$=small)

初始星为：

star($e1$) : ($at1\neq$c), ($at2\neq$5), ($at3\neq$small)

star($e5$) : ($at1\neq$a), ($at2\neq$4), ($at3\neq$large)

每个星具有三个单条件规则且来自不同星的规则相交。从每个星中选择一条规则，按照以下方式进行修改，以使所获规则集中的规则逻辑上不相交，

而且它们的合取覆盖所有样本实例（通过 Michalski 和 Stepp，1983 所描述的 NID 和 PRO 过程来完成）。结果就是以下求解。

簇 1：$(at1 = a \vee b) \& (at2 = 2 \vee 3)$

样本实例：e1,e3,e4

簇 2：$(at1 = b \vee c) \& (at2 = 4 \vee 5)$

样本实例：e2,e5,e6,e7,e8

由于从上述规则中所选择的新种子不能帮助改进所获聚类，所以上述规则构成了 k=2 时的求解。对于更大 k 不断重复执行算法也无法改善求解内容，因此以上就是最终结果。更多细节可参见 Michalski 和 Stepp 的文献（1983）。

1.6.1.2　脆弱的概念层次结构

从一组固定未分类样本中发现概念的算法，其计算代价可能是巨大的。另一方面，概念形成算法尽可能模拟人类层次结构的形成，在这个意义上，这个过程被认为是增量的。此外，重点通常（并不总是）都放在产生**层次结构序**（Hierarchically Ordered）的概念上。

这些系统的大多数都将分类与学习过程结合起来：在一个新实例到达时，系统将其集成（分类）到当前的知识结构中。图 1.20 描述了这一过程，其中 UNIMEM 系统（Lebowitz，1987）对 6 个细胞样本（由它们的大小，原子核数目和尾部数目所描述）已经形成了一个层次结构。当另一个样本（小的两个尾部，一个原子核）到达时，结果发现它与知识树中右边分枝的三元组最相似。遇到这个新情况时，系统将创建一个新子类，如图 1.21 所示。

概念形成算法被想像为典型的搜索系统——有初始状态、结束标准、搜索操作、搜索策略，当然还有表示问题。初始状态由描述第一个样本实例所确定，同时最后一个状态就是最后一个样本实例后的知识结构——系统假设只要有样本实例输入就不断学习。最普通的搜索策略就是由一个有适当选择标准的爬山搜索，以确定当前结构的质量。

如图 1.20 和图 1.21 所示，UNIMEM 所使用的表示方式就是自我解释。每个结点（代表系统所形成概念）由一组诸如 size(big)特征所定义（图中，文字量归为属性值）。每个特征伴随一个称为分数（Score）的整数，以告诉学习者至今遇到多少次特征。注意到分数也反映到其他聚类所替换的样本，例如，在右边分类中“2-tails”特征的分数。分数确定特征的力度。小分数表明特征是基本无关的或许可以忽略。相反，高分数表示特征应该“固定”在结构中，

不能删除。

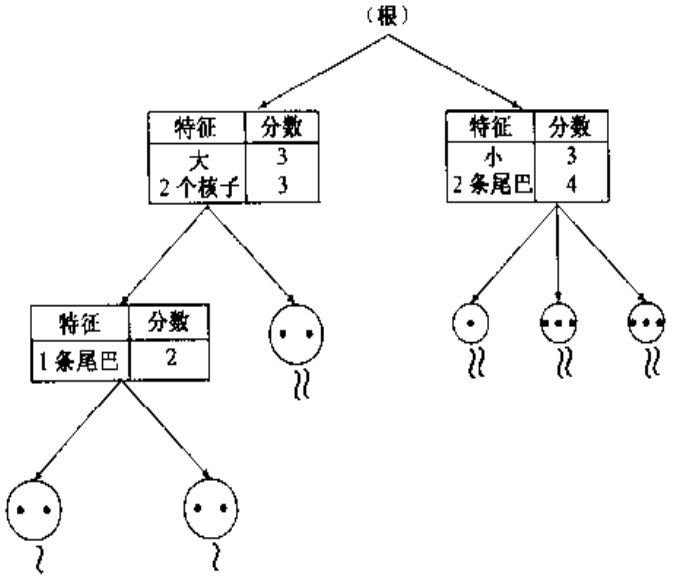


图 1.20　UNIMEM 表示方法

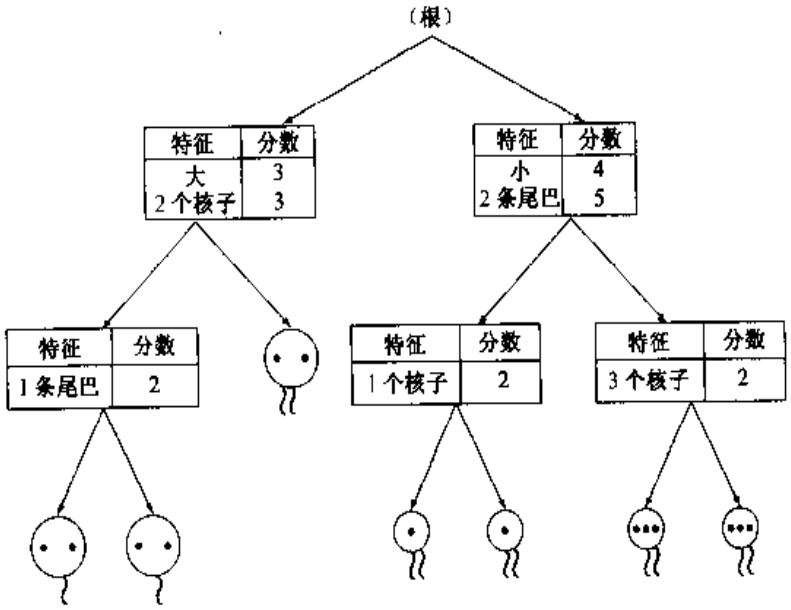


图 1.21　图 1.20 所示的结构吸收一个新样本

基本上，UNIMEM 中概念形成过程所涉及的搜索操作说明如下：

（1）在最近结点中存储一个新样本实例；

（2）若能够改善某些通用评估所创建概念的结构质量标准所给出的评价价值，则创建一个新结点；

（3）若特征分数超过预定义的阈值，则固定一个特征；

（4）若分数低于其他特征的分数，则删除这个特征；

（5）删除过于泛化的结点（包含且仅包含一些特征）。

有关 UNIMEM 系统执行过程的更多细节请参见 Lebowitz（1987）。

1.6.1.3　概率概念层次结构

其他概念形成系统，与 UNIMEM 内部的表示结构、描述语言（如：符号对数值属性）、搜索操作和指导搜索的评价函数有所不同。

因此 COBWEB 系统，层次结构中每个结点包含有关单个属性值的完全信息，如图 1.22 所示，其中概率简单地按照相对频率进行估计。

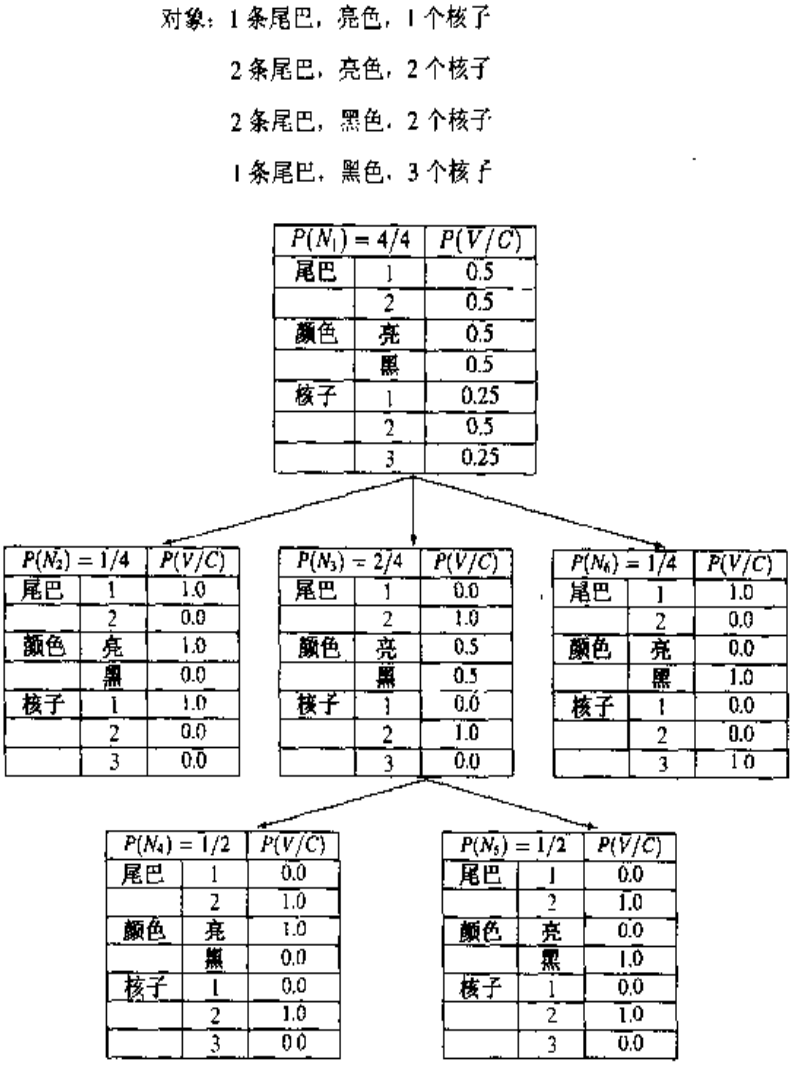


图 1.22　COBWEB 的表示结构

其表示的特殊之处在于系统不能保存脆弱的表示。而且，每个属性-值对伴随一个表示其概率的数字，以说明概念实例拥有这个特别的属性值。图 1.22

中的每个结点包含一个头部和二列的表。头部包含有关一个实例落入本分类的频率，$P(N_i)$。这个数据表包含相对任何属性值对出现的相对频率。

COBWEB 利用以下搜索操作：

（1）将新样本实例**结合**（Incorporate）到现存的某个结点中

（2）为样本实例**创建**（Create）一个新结点

（3）将两个结点**合并**（Merge）为一个结点

（4）将一个结点**分解**（Split）为两个结点

当遇到一个新样本实例时，学习者必须确定最好应用哪个操作。了解每个操作是如何改变概念层次结构的，系统利用公式来评估每个潜在新层次结构的功用：

$$\frac{IG-UG}{N}$$

其中 UG（Uninformed Guess）就是可从一组无序对象中正确猜出的属性值数目；IG（Informed Guess）就是给定概念层次树情况下，能够正确猜出属性值的期望数目；N 为当前层次结构中存在的分类数目。

更特别地，推荐以下公式（更多细节参见 Fisher，1987）：

$$\frac{\sum_{k=1}^{N} P(C_k)\sum_i \sum_j P^2(A_i=V_{ij}\mid C_k)-\sum_i\sum_j P^2(A_i=V_{ij})}{N}$$

这里 $P(C_k)$ 就是类别 C_k 的相对频率；$P(A_i=V_{ij})$ 就是属性 A_i 取 V_{ij} 的概率；$P(A_i=V_{ij}\mid C_k)$ 则是相对条件概率。

这个概率方法的基本点就是创建一个概念层次结构，使在给定落入有关实例类别信息后，预测出的未知实例中属性值数目最大。

1.6.2 寻找自然定律

许多研究人员声称拥有功能强大的概念形成算法，有人甚至想更进一步，试图建立一个系统，它不仅有能力构造新概念，而且有能力描述它们间关系的定律，就像物理和化学中的情况。

有若干理由支持这个领域的活动：

（1）今天，有来自许多科学领域的数据库，等待人们对它们进行分析；

（2）已经研究出功能强大的机器学习与人工智能技术，可以实现一种“智能”分析。

（3）即使没有建立自动智能分析，对人工发现的研究或许可以帮助说明人类发明活动中的一些秘密（如灵感、类比和抽象）。

1.6.2.1　定量经验定律

假设任务是再发现理想气体定律。读者可以从高中的学习中回想起这个定律的固有形式：$PV=8.32NT$，其中 P 是压力，V 是体积，N 是气体质量，T 为温度。Langley 等人（1987）提出了一个有能力完成这项任务的系统，并将其命名为 BACON。这里仅仅只能给出一个概要介绍，有关详细情况请参考他们的论文。

BACON 开始建议一系列实验，以提供测量数据。人类操作员完成这些实验并将结果提供给计算机。在获得足够的实验数据之后，系统搜索数学函数空间以期发现一个描述数据的方程。搜索这个方程的一个方法就是使一个变量依赖于其他相互独立的变量。设系统拥有一批典型的公式形式，如：$y=ax^2+bx+c$；$\sin(y)=ax+b$；$y^{-1}=ax+b$。

表 1.6　BACON 系统的典型数据

质量因子	温度	压力	体积
N=1	T=10	P=1000	V=2.36
.	.	P=2000	V=1.18
.	.	P=3000	V=0.78
.	T=20	P=1000	V=2.44
.	.	P=2000	V=1.22
.	.	P=3000	V=0.81
.	T=30	P=1000	V=...
.	.	P=2000	V=...
.	.	P=3000	V=...
.	⋮		
N=2			
.			
.	⋮		
N=3			
.			

基本原理包括：选择最好的公式形式，调整参数 a，b，…，以期发现一个最好描述观察数据的方程。

假设选定方程 $y^{-1}=ax+b$。首先，参数 a，b 取初始化值 1，0 和-1，因此考虑以下初始状态集合：[a=1，b=1]，[a=1，b=0]，[a=1，b=-1]，[a=0，b=1]，[a=0，b=0]等。

在搜索过程中，通过每次增加或减少一个参数值对参数进行调整，从0.5开始，然后 0.25，0.125，…。评估函数评估每个随后产生的方程，这些方程定义了被测数据和方程参数值之间的关系。

假设测量获得表 1.6 所示的值，BACON 将按照以下步骤检查这些数据。

（1）找出一个函数 $V = f(P)$ 来描述样本实例的三元组，如表 1.6 所示。对于三个温度 T=10，T=20 和 T=30。

假设 $V^{-1} = aP + b$，对于以下参数获得最佳匹配。

T=10：a=0.000 425，那么 $V^{-1} = 0.000\,425P$

T=20：a=0.000 410，那么 $V^{-1} = 0.000\,410P$

T=30：a=0.000 396，那么 $V^{-1} = 0.000\,396P$

（2）由于参数值明显依赖温度 T，下一个任务就是发现与 a 对 T 有关的函数。然后，利用公式形式 $a^{-1} = cT + d$ 获得最佳匹配，其中参数值 c 和 d 依赖 N。

N=1：c=8.32，d=2 271.4，那么 $a^{-1} = 8.32T + 2\,271.4$

N=2：c=16.64，d=4 542.7，那么 $a^{-1} = 16.64T + 4\,542.7$

N=3：c=24.96，d=6 814.1，那么 $a^{-1} = 24.96T + 6\,814.1$

（3）发现 c 对 N，d 对 N 有关的函数。通过 c=eN 和 d=fN 获得最佳匹配，其中 e=8.32 和 f=2 271.4，这些参数不依赖任何其他变量。

（4）利用先前发现的方程来替换原方程，由此系统获得以下方程：

$$V^{-1} = (8.32NT + 2\,271.4N)^{-1} P$$

上述方程可以较容易地转换为：

$$PV = 8.32NT + 2\,271.4N$$

这确实就是理想气体定律。注意 BACON 已经发现摄氏温标不合适。事实上，系统利用绝对温标，将所测量的摄氏温度值加上 273。

最后，BACON 本质就是应用普通搜索原理以寻找一个定量定律的理想形式，而不是仅仅发现最佳匹配参数，如同传统的回归技术一样。

与定量发现类似的定性发现就是 GLAUBER 系统，它试图形成定性化学定律和概念。GLAUBER 有能力再次发现酸和碱的概念，并推测出了这些概念的一些基本性质。有关更多细节，以及其他一些有自动发现能力的系统，请参见 Langley 等人（1987）文献。本小节结束前，我们简要介绍一下在上述介绍的基础上更为复杂的一些变化。

1.6.3　动态系统的发现

LAGRANGE 是一个可以发现数据中数值定律的程序，与 BACON 系统情况一样。然而 LAGRANGE 不同点在于，它从动态系统测量的数据中产生模型。LAGRANGE 模型具有不同方程形式。与控制工程中所采用的传统系统辨识技术不同，LAGRANGE 可发现方程结构，而不仅仅是参数的值。

为说明清楚，考虑生态建模领域的一个应用。两个变量 x 和 c，假设 x 是测试床的细菌浓度，c 为细菌营养液的浓度。LAGRANGE 的任务就是：给定 $x(t)$和 $c(t)$两条曲线的数据表示，发现一个差分方程，其数值解决方案与两个已知行为相对应。在生态领域这个特别情况下——Džeroski 和 Todorovski（1994）——LAGRANGE 发现以下差分方程：

$$\underline{c} = -\frac{1}{60}x - \frac{10}{6}\underline{x}$$

$$c\underline{x} = -x - 100\underline{x} + 0.09cx$$

有关系统参数的具体取值，诸如增长率，与生态建模中著名模型——莫诺模型对应。

一般，LAGRANGE 发现问题的描述如下。

已知：

一个动态系统的时间轨迹：$\vec{x}(t_0)$, $\vec{x}(t_0+h)$,…

参数：

o 为差分方程的阶

d 为新产生项的最大深度

r 为“独立回归参数”最大值

t_R 为重要性阈值

发现：

参数（o，d，r）的差分方程，在重要性阈值 t_R 下与数据相匹配。

据 Džeroski 和 Todorovski（1994）报告，LAGRANGE 成功发现（再次，应为重新发现）以下差分方程：三种化学物质的一个化学反应，食肉动物与猎物链的建模，所谓 Brusselator 化学反应器，pole-cart 系统等。

1.7 如何处理搜索空间过大

机器学习的一个基本问题就是所有可能描述的空间常常过大以致必须依赖启发式或成为难以处理的计算问题。评估函数的局部最优问题在大空间中将更为严重。

解决这个问题的两个技术需要单独一个章节来描述：利用类比和保存初始样本的思想代替泛化描述。

1.7.1 类比提供搜索启发

人工智能领域中已对类比原理进行广泛的研究，因为普遍相信发现合适类比的能力就是智能中的秘密之一。基于类比推理已有了许多研究工作。

从机器学习角度来看，这个机制的本质是什么呢？Kodratoff（1988）将图1.23所示的模式融合为一个通用类比框架。这里，S 代表源，SC 代表源概念，T 代表目标，TC 代表目标概念。任务就是从 T 按某种方式推出目标概念，这种方式与源概念从源中推出方式类似。因此，有了目标后，学习者必须发现一个合适的源。

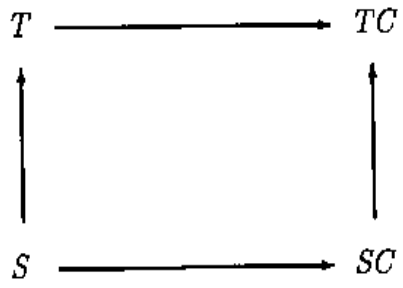


图 1.23 类比的一般模式

Greiner（1988）建议以下类比推理的通用过程。

类比推理的算法

（1）**识别**（Recognition）。给定目标概念，在背景理论中发现与 T“相似”的源 S。相似性可通过句法距离和存在一组统一替换的普通泛化来进行度量，或通过用户提供某种暗示进行度量。

（2）**推导**（Elaboration）。发现 SC，与推理链 $\vdash_S$ 一起，从源 S 推出。注

意对于每个 S，通常存在一组 SC。

（3）**评价**（Evaluation）。在 SC 中，发现最符合给定标准的一个。

（4）将一个与推理链 $\vdash_S$ “类似”的推理链 $\vdash_T$ 应用到 T 中，因此获得 TC，并对 TC 的功用进行评价。

（5）若必需的，循环重复步骤（1）～（4）以发现 S，SC，$\vdash_S$ 和 $\vdash_T$ 以产生最有前途（有用）的 TC。

（6）**合并**（Consolidation）。将 TC 与推理链 $\vdash_T$ 一起合并到背景理论中。

由于上述框架有些过于一般，通常需要某些合理的约束。因此源 S 可由用户明确提供以告诉系统任务是否通过一个管道结构来计算流速，那么就可以利用电子工程类似定律（Kirchhoff 定律）。另一种可能性就是用户接管评估过程并为系统所建议的源选择一个合适的 SC。Greiner（1988）描述一个有能力学习解决液体流动问题的系统，它利用了电路类比先验知识。

1.7.2　基于示例学习

一个明确的概念描述并不总是可明确获得的。若学习的惟一理由就是需要识别未来的样本实例，那么学习者就能采用其他策略：不用描述而是保存典型样本实例。它避免了许多由于搜索一个非常巨大泛化空间所引起的麻烦。注意早期的一些概念形成系统也采用类似的思想。

本节描述了 IBL 系统的原理（Aha 等人，1991），它有能力保存所选的样本实例（由属性值所描述），并根据所谓**最近邻**（Nearest-neighbor）原理利用它们：新的样本被赋予所保存样本实例中接近的样本实例的类别。

一个简单计算两个样本实例 x 和 y 的公式如下（x_i 和 y_i 为第 i 个属性值）：

$$\text{similarity}(x,y)=-\sqrt{\sum_{i=1}^{n} f(x_i,y_i)}$$

这里，函数 f 通过以下公式计算数值属性：

$$f(x_i,y_i)=(x_i-y_i)^2$$

对于符号量和布尔属性，计算公式如下：

$$f(x_i,y_i)=\begin{cases}1 & x_i \neq y_i \\ 0 & x_i = y_i\end{cases}$$

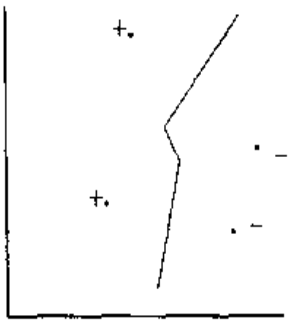


图 1.24　由+和－样本实例定义的正例与反例空间

图 1.24 所示的基本原理中，有四个样本实例由两个变量所表示，还有区分函数，它将正例空间与反例空间分割开来。

学习假设可获得反馈，此反馈会立刻通知学习者有关每个单独分类操作的成功或失败。IBL 算法的一个简化版本包含以下步骤。

基于示例学习算法

（1）开始时，定义一组**代表**以包含第一个样本实例；

（2）读入一个新样本实例 x；

（3）对于代表集合中的每个 y，确定 similarity(x, y)；

（4）将 x 标记为代表集合中最接近样本实例的类别；

（5）根据反馈，确定分类是否正确；

（6）将 x 并入代表集合，并转到（2）。

有两点不足破坏了这个基本版本的功能：由于需要保存所有样本实例而引起的过多存储要求，以及对噪声敏感。

纠正措施包括通过“等一下再看”策略来选择存储样本，其本质可概括为以下几条原则：

（1）在一个新样本实例分类后，先前每个样本实例的“重要性分数”被更新（以下介绍），本实例被保存起来；

（2）具有**好分数**的实例将用于分类；**分数较差**的实例被删除；

（3）中间的实例被保留作为候选。然而，它们将不被用于分类。

在分类阶段，若**好**样本实例存在的话，新样本实例被赋予最接近的**好**实例。否则新样本实例被赋予最接近的中间实例。

接着，系统增加那些与最接近的好实例相比，与新样本实例最接近的中间实例的分**数**。若没有好实例，系统会在一个新样本实例周围随机选择的超空间中更新其中间实例。

在由一个新实例所获的分类准确率高于样本的类别频率时，就被认为是好的分数。正例的分类**准确率**就是在整个样本集合中正确识别的正例比例。

基于示例学习被报道在属性值领域取得了重大的识别能力，特别是在样本实例非常大而妥善选择描述它们的属性时。还有，对付噪声的鲁棒性也令人满意。在另一方面，若样本描述包含不相关属性和/或若学习过程可用样本数较少时，系统将无法充分展示其功能。

1.8　机器学习的近邻

机器学习通常作为人工智能的相关技术，特别是对于那些目标为归纳出**有意义**或**可理解**，同时又有助于改善性能的符号描述。在一个更为广泛的意义上，机器学习任务可被定义为任何可导致知识增加或某些过程、技能，诸如对象识别的性能改善的计算过程。

特别是学习—识别任务就常常利用那些传统上并没有包含在机器学习范畴的方法来解决，这些方法具有相同或类似的目标。因此一种统计数据分析（参见 Everit，1981）和传统模式识别（参见 Duda 和 Hart，1973）产生了许多有用的技术。即使许多其他方法更为详细的讨论无法在这里展开，但必须简要介绍两个技术，由于其受欢迎程度，许多人试图将它们与机器学习算法相结合，这两个技术是：神经网络和遗传算法。

1.8.1　人工神经网络

在 20 世纪 50 年代后期，针对模式识别目的，Mark Rosenblatt 提议利用一种简单设计，由早期生物神经元数学模型启发。在他的著名论文（Rosenblatt，1958）和专著（Rosenblatt，1962）中，他将他的设计称为感知器，并说明了感知器可以被训练来通过自动调整其参数完成识别工作，在基于一组事先预分类样本的基础上。图 1.25 描述了其原理。若干输入信号 x_i，每个乘上一个权重 w_i，并与一个累加单元相联系。所产生的累加和 $\text{sum}=\sum_i w_i \cdot x_i$ 为一个步长函数，以确保若累加值超过一个阈值 θ，则感知器输出就是 1，否则就是 0。其他情况下，输出结果也可以为其他值，如 1 和-1。

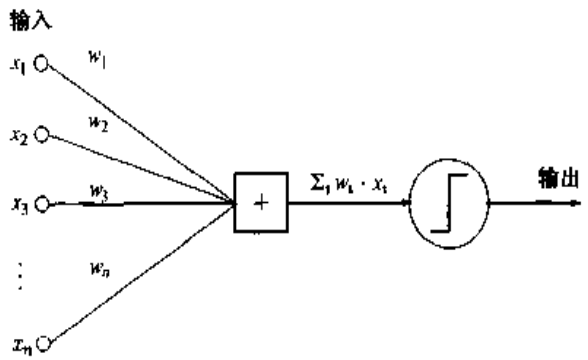


图 1.25　某感知器的一般模式

适当调整权重 w_i 和阈值 θ 以确保感知器将会对输入向量产生相应的输出值。因此信息被编码到权重并赋予每个单个的输入，每个输入代表一个属性。越相关的属性会被赋予越多权重，越少相关的属性会被赋予越小权重。感知器学习算法寻找适当权重以便完成所需的从输入向量到二元值的映射：$R^n \rightarrow \{0,1\}$。

遗憾的是，某些概念不能通过感知器来获取，例如，exclusiveOR，在 Minsky 和 Papert（1969）文献有介绍。这就是为什么感知器很少被独立使用，而是相互连接在一起，诸如图 1.26 所示的多层感知器（有关多层感知器，请参见 Rumelhart 等人的文献，1986）。

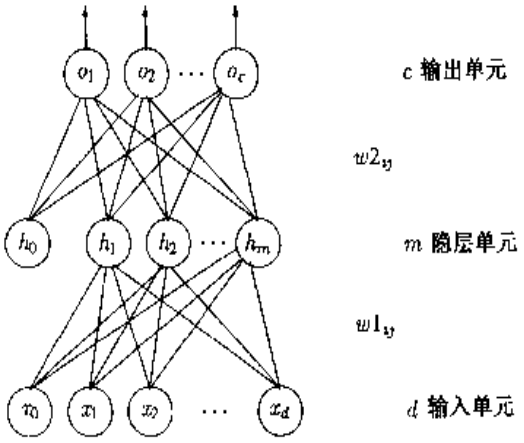


图 1.26　多层感知器

原则上，多层感知器包括一层输入结点、一层输出结点和它们之间一层或多层“隐藏”层结点。在识别阶段，输入向量的组件与输入层相联系。显然，某些感知器“触发”（输出为 1），同时它们输入的权重之和超过特定的阈值。然后 1 或 0 值传播到下一层，直到网络输出为止。

基本阈值函数太刚直（不容许噪声并不利于学习），因而通常用 sigmoid 函数根据输入计算单个单元的输出：

$$f(\text{sum}) = \frac{1}{1+e^{-\text{sum}}}$$

这里 sum 为单元输出信号量的加权累加值。根据这个公式，单元输出 0 和 1 之间的实值。对于 sum=0，输出为 0.5；对于 sum 大的负值输出趋向于 0；而对于 sum 大的正值输出趋向于 1。该公式比步长函数更能容忍噪声信号。

通常，仅利用一个隐层，如图 1.26 所示，然而，在许多复杂任务中，研究人员在利用两个到更多隐层时，取得了更好的成绩。

以下提供了自动调整权重的过程，但没有做更进一步的讨论。有兴趣的读者可以参见许多有关神经网络的专题论文。在许多现存神经网络教科书中，或许 Beale 和 Jackson 的著作（1990）被推荐为易读的导论。有关更多的全面描述，可参见 Haykin（1994）文献。

后传学习算法

（1）定义神经网络设置中的各层中的单元数；

（2）设置权重 $w1_{ij}$ 和 $w2_{ij}$ 为小的随机数，取自区间[-0.1，0.1]；

（3）选择一个样本实例并设其属性值为 $x_1, x_2, \cdots, x_k$，将样本实例与输入层相联系；

（4）从输入层传播输入值到隐藏层，第 j 个单元的输出按照以下公式进行计算：$h_j = \dfrac{1}{1+e^{-\sum_i w1_{ij} \cdot x_i}}$；然后传播所获得的值到输出层。该层上第 j 个单元的输出按照以下公式进行计算：

$$o_j = \frac{1}{1+e^{-\sum_i w2_{ij} \cdot h_i}}$$

（5）将输出 o_j 与教师分类 y_j 相比较；计算纠正偏差 $\delta 2_j = o_j(1-o_j)(y_j - o_j)$，并且按照以下公式调整 $w2_{ij}$ 的权重：

$$w2_{ij}(t+1) = w2_{ij}(t) + \delta 2_j \cdot h_i \cdot \eta$$

这里 $w2_{ij}(t)$ 是 t 时刻相应的权值，而 η 为一个常量且有 $\eta \in (0, 1)$；

（6）通过公式 $\delta 1_j = h_j(1-h_j)\sum_i \delta 2_i \cdot w2_{ij}$ 计算隐藏层的纠正偏差，并利用以下公式调整权重 $w1_{ij}$：

$$w1_{ij}(t+1) = w1_{ij}(t) + \delta 1_j \cdot x_i \cdot \eta;$$

（7）转到步骤（3）。

以上算法仅仅描述了多层感知器学习的基本原理，用户必须熟悉它在现实应用中的各种不足。但是这些情况已被详尽地研究过，今天神经网络代表了一个较完善的科学学科。

1.8.2　遗传算法

读者业已看到学习过程在许多情况下，被认为是一个对由给定语言所定

义的表示空间中的搜索。本小节将介绍一种补充传统启发搜索技术的强有力方法：遗传算法（Genetic Algorithm），它是受自然界类似原理的启发而产生的。

一般而言，自然界进化是受以下三个基本原则所控制的：

（1）**适者生存**（Survival of the Fittest）意味着最强壮的物种具有最大机会生存和繁衍，而弱者在其达到繁衍阶段前就可能死亡了；

（2）在**两性繁衍**（Sexual Reproduction）中，物种挑选最好的作为其伙伴，因此与适者生存原则相符合。然后它们重新组合遗传基因信息，创造具有不同特性的新物种；

（3）**突变**（Mutations）引起随机、相对减少的遗传信息的变化。

这种“搜索技术”在自然界毋庸置疑地成功激发了研究人员将其变为计算机算法程序的研究。Goldberg（1989）写了一篇介绍遗传算法原理的入门文章。

本节将仅介绍了解这一机制运作实现所必需的基本原理。利用遗传算法成功解决任何技术问题的开始，必须回答两个问题：如何用染色体对搜索空间进行编码？如何定义适应度函数（参见图 1.27），使它在启发式搜索中扮演着重要角色？在大多数实现中，染色体表示为位串。每位可代表一个二元属性，多值属性或一个谓词等。适应度函数，度量物种的生存机会，可被定义为由染色体推出的描述，由这一描述所划分而产生的熵（满足描述的样本实例对不满足的样本实例）等。

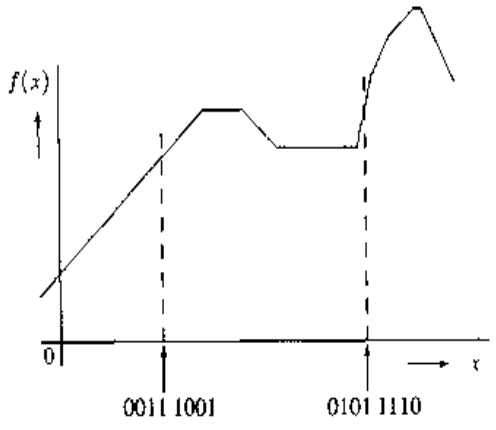


图 1.27　评价函数 $f(x)$ 与两个二元物种

遗传算法原理由如表 1.7 所示的样本实例来说明。这里适应度函数定义为 $f(x)=1/(x+1)$，其中 x 为二进制形式染色体（如“111”=7）所代表的数目。显然，$f(x)$ 的最大值将在串“000000”时达到。

表 1.7　遗传搜索中一步以获得函数 $1/(x+1)$ 的最大值（没有突变）

老一代		x	$1/(x+1)$	存活		新一代	x	$1/(x+1)$
(1)	100101	37	0.026	(4)	000|111	000011	3	0.250
(2)	001011	11	0.083	(2)	001|011	001111	15	0.063
(3)	010100	20	0.048	(4)	0001|11	000100	4	0.200
(4)	000111	7	0.125	(3)	0101|00	010111	23	0.042

表 1.7 说明了算法中的一步。老一代包含 4 个数：37，11，20 和 7。适应度函数最大值在 $f(7)=1/(7+1)=0.125$ 时达到，因此代表 x=7 的染色体具有最大生存机会。相反，x=37 时具有最小适应度函数值，$f(37)=1/(37+1)=0.027$，结果其生存机会就非常小。这个机会由随机数产生器给出，以确保最强的物种在生存空间中能够复制多次（这里，“技术上”遗传算法与“自然”的有差别），而最弱的物种将消失。这一步称为**繁衍**。

在下一步，每个生存者将选择一个配偶并交换它们部分遗传信息。这一步称为**重新结合**（Recombination），其模型为随机子串交换。为简单起见，表 1.7 中的染色体仅交换随机长度的尾部。这一步后，就会产生更强一代物种。确实，适应度函数的值表明其最大值与平均值都在增加。

突变操作（表 1.7 中没有采用）的模型相当直观：具有非常小的可能性，一位反转到它的相反值。通常可调整可能性常量，以便在一代中仅有几个（0 到 5）出现突变。

GA 算法

（1）定义初始物种群并将其作为一组二元串，它们由某个事先定义好的机制随机产生；

（2）通过一个机制复制物种群成为生存者集合，以确保具有较高适应度函数的物种具有更高的生存机会（可以复制多次）；

（3）对于每个生存者，找到一个配偶并与其交换部分编码为二进制串的信息。以非常小的频率，将单个一位反转成其相反值以模拟突变；

（4）若适应度函数在若干循环中均没有增长则停止，否则转到步骤（2）。

感兴趣的读者可以参考 Goldberg（1989）的专题论文，其中还可以找到其他许多相关文献。

1.9　混合系统与多策略学习

现实世界中的问题常常无法仅用以上介绍的基本技术就可以成功地解决。这些技术的每一个都有其优点和不足。例如：TDIDT 是针对属性值数据而设计的，它在面临更复杂描述语言且需要相当多背景知识时，将束手无策。同样，基于谓词逻辑的系统在处理 Horn 子句非常得心应手，但无法应付计算

量非常大的情况；神经网络擅长模式识别，但对初始的拓扑结构和权重，以及属性的选择都非常敏感。遗传算法，虽然能力很强，但需要精心编码产生染色体并且学习起来非常慢。

因此，非常顺理成章地，机器学习研究人员探索将单个方法结合起来以弥补彼此的不足。研究构造将不同策略或方法结合起来的系统在很早就开始了，并由此产生了机器学习一个新子领域，称为多策略学习（Michalski 和 Tecuci，1994；Wnek 等人，1995）。

1.9.1 熵网络

前面已经提到过，在输入向量包含无关特征时，神经网络的性能就会变差。相反，TDIDT 相关系统，虽然擅长删除噪声和无用属性，但由于倾向依赖严格属性顺序而建立过于严格的描述。这些不足使得人们努力将两个方法进行结合。**熵网络**（Entropy Nets），一个较为简单但却令人信服的结果，由 Sethi（1990）提出。设计的系统主要用来解决样本是由数值属性描述时的学习问题。

产生熵网络的过程包括三个步骤：树的成长，将树翻译为一个神经网络（因而被称为熵网络）和熵网络的训练。

有关决策树的生长，可以利用本章较早介绍的过程。而利用基于熵的度量来选择合适的属性也是显而易见的。

将决策树映射为一个神经网络，可以通过布尔属性的合取与析取（可由神经元简单模型实现）来完成。假设所有权重均为 1。然后将神经元阈值设为 n-0.5 以确保仅当所有输入为 1 时，神经元能被激活。这种情况下，输入的加权累加和就是 $\sum w_i a_i = n$，它将超过阈值。类似地，将神经元阈值设为 0.5 将使其实现输入的析取。

这种映射如图 1.28 所示。网络底层仅包含输入。第一隐层（称为划分层）中每个单元完成对树中每个内部结点的一项测试（诸如 $a_1 < t_1$）。决策树的每个叶结点映射到第二隐层中对应单元，称为 AND 层。这些单元完成沿树分枝测试的合取，最后输出层的每个单元（OR 层）代表一个分类值并模仿具有相同类别叶结点的析取。

随后利用后传算法（前面已经介绍过）完成网络的训练。其想法就是进

一步增加系统分类的准确率。与初始决策树相比，代价就是可解释的编码知识没有了。

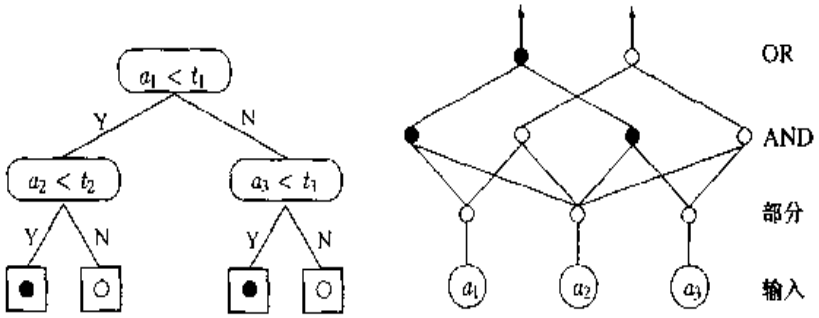


图 1.28　决策树功能与相应的熵网络

1.9.2　基于知识的神经网络

神经网络的另一个不足就是它们没能利用背景知识，并且由于搜索理想拓扑结构而变得复杂化了。这就是为什么 Towell 等人（1990）提出了 KBANN 系统，该系统能够按照逻辑进行学习并通过神经网络训练方法调整所获得的知识。

假设背景知识包含以下规则，它们合在一起，定义了某个概念 a。

```
a:- b, c
b:-g, not(f)
b:-not(h)
c:-i, j
```

a 定义为 b 和 c 的**中间概念**（Intermediate Concepts）。它们反过来依赖**支持事实**（Supporting Facts）g，f，h，i，j。支持事实就是那些可以在作为样本对象上特征的直接测量。中间概念由支持事实及其他中间概念来定义。

系统主要包括两步。第一步，知识翻译为网络，其中支持事实通过输入单元来建模，中间概念通过隐藏层以及最终概念通过输出单元来建模。不同层次中各个单元间的依赖性由权重来表示。在这个阶段，每个权重都有同样的绝对值。在第二步，网络被扩大以便为那些没有在背景知识中明确出现的谓词和事实提供一个机会。然后，对权重稍微地随机改变一下，并利用后传算法对网络进行训练。

这个原理如图 1.29 所示。规则被翻译为图 1.29 左边粗略的拓扑图，其中

虚线代表负权重（如规则 b:-not(h)）。然后通过引入补充的低权重连接对这个拓扑图进行细化，如图 1.29 右边所示。有关更多细节可参见 Towell 等人（1990）文献。Bala 等人（1994）提出了利用基于 AQ 算法初始化神经网络的方法。

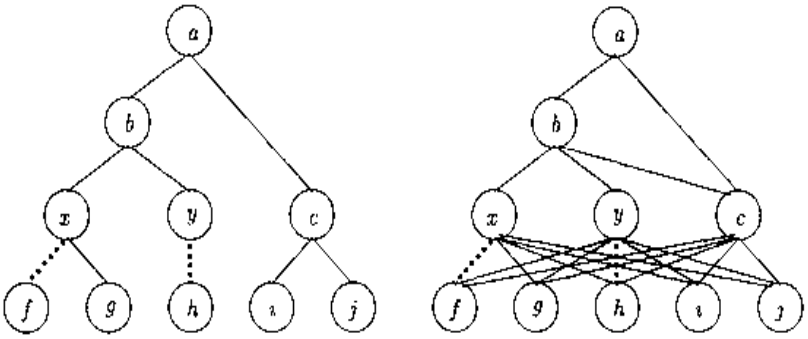


图 1.29　将知识翻译为一个神经网络

我们已经提到网络是一个黑箱系统，利用神经网络训练精练的知识的不利后果就是用户无法再解释结果了。为解决这个问题，Towell 等人（1991）提出了一个方法，该方法可以从训练后的神经网络中以产生式规则的形式提取相应知识。

1.9.3　AQ 泛化中的遗传搜索

AQ 算法中的弱点之一就是搜索最优泛化的种子，理由是所有可能泛化的数目如此之大，以致整个程序的可控制计算成为难点。

这个难点促使 Venturini（1993）研究了 SIA 系统，这里搜索泛化的理想种子是通过遗传算法的机制来完成的。每个染色体代表一条产生式规则。然而，繁衍模式以及突变与以上描述相同。重新结合仅在一个相对小比例物种范围中利用交配操作，以此来补充传统的泛化。物种规模是可变的。

为发现理想的泛化，SIA 开始时的物种群仅包含种子的最特别的描述（因此物种群初始规模为，N=1）。在随后每代中，随机选择以下操作之一并将按括号内的概率应用相应的操作：

（1）**创建**一条新规则（概率 10%）；

（2）选择任意一条规则并对其进行**泛化**（概率 80%）；

（3）通过交换它们之间的某些合取，完成两条规则的传统**交融**（概率 10%）。

有关更多的详细解释，请参见 Venturini（1993）的处女作。

1.9.4　GA 与神经网络的结合

最后，一些研究人员研究了利用遗传算法来发现神经网络结构和/或权重的可能性。Bornholdt 和 Graudenz（1992）的工作可作为这些努力的一个示例说明。这里，遗传算法搜索寻找网络的理想结构。染色体中的每个位置代表神经元，包含指向其他神经元的指针，因此这样的染色体比一个简单位串要复杂得多。用适应度函数来测定一个给定网络的质量。

然而，更多有关这些研究工作的详细讨论，与本章学习算法的主流有偏差，因此这里就不赘述了。

1.10　展望

正如以上所介绍的，机器学习领域已发展了许多方法与技术。在 Cohen，Feigenbaum（1982）和 Michalski 等人（1983）的文献中，可以找到这些方法与技术的演化进程的回顾与说明。

这里介绍的方法属于归纳概念学习的一般类别，它构成了机器学习中最先进的任务。大多数方法的内在假设就是学习者从已知样本实例中归纳出概念描述。这一过程本质是可归纳的，且无法保证所产生描述的正确性。因此，由这些技术创建出的概念通常必须在新数据上进行测试。

由于这些描述代表已知事实的泛化，且可能不正确，所以许多应用，在使用它们之前，需要由人类专家进行解释和理解。因此，我们强调概念学习中可理解性要求的重要性。

描述可以具有不同形式，诸如决策树、决策规则、神经网络、Horn 子句、语法等。每种表示需要某种不同的信息处理方法，每个都具有自己的优点和缺点。在应用它们解决一特定问题时，需要分析当前的问题，并确定何种表示方法和学习策略最为合适。

为完全起见，在结束时还需要介绍这一领域其他的一般方法，这些方法在本章中没有介绍。它们包括：

（1）**基于解释的学习**（Explanation-based Learning），一种从概念样本实

例中推理获得操作性知识，某个**先验**已知的抽象概念描述的方法论——参见，例如，DeJong 和 Mooney（1986）或 Mitchell 等人的文献（1986）；

（2）**基于事例学习**（Case-based Learning），一种学习方法，保存概念样本实例，新事例通过最接近的过去事例（事例集）类别来确定——参见，例如，Bareiss 等人（1987）或 Rissland 和 Ashley 的文献（1989）；

（3）**增强式学习**（Reinforcement Learning），在某个特定步骤有关性能的数值反馈被用于改进学习系统的参数——参见，例如，Sutton（1988）的文献。

机器学习是一个相对年轻的学科，未来很有可能会提出许多新的、更强大的方法。本书后面的章节将展示说明已成功应用解决许多实际问题的现有技术。

参考文献

Aha, D.W., Kibler, D., and Albert, M.K. (1991). Instance-Based Learning Algorithms. MachineLearning, 6:37-66

Bala J.W., Michalski, R.S., and Pachowicz, P.W. (1994). Progress on Vision through Learning at George Mason University. Proceedings of ARPA Image Understanding Workshop 191-207

Bareiss, E.R., Porter, B., and Wier, C.C. (1987). PROTOS: An Exemplar-Based Learning Apprentice.

Proceedings of the Fourth International Workshop on Machine Learning, Irvine, CA, Morgan Kaufmann,12-23

Beale, R. and Jackson, T. (1990). Neural Computing: An Introduction. Adam Hilger, Bristol Bergadano, F., Matwin, S., Michalski, R.S., and Zhang, J. (1992). Learning Two-Tiered Descriptions of Flexible Concepts: The POSEIDON System. Machine Learning, 8, 5-43

Bornholdt, S., and Graudenz, D. (1992). General Asymmetric Neural Networks and Structure Design by Genetic Algorithms. Neural Networks, 5:327-334

Bratko, I. (1990). PROLOG Programming for Artificial Intelligence, Addison-Wesley (Second Edi-tion)

Breiman, L., Friedman, J., Olshen, R., and Stone, C.J. (1984). Classification and Regression Trees.

Belmont, California, Wadsworth Int. Group Cestnik, B. (1990). Estimating probabilities: a crucial task in Machine Learning. Proc. ECAO 90,Stockholm, August Cestnik, B., and B Bratko, I. (1991). On estimating probability in decision tree pruning. Proc. EWSL-91,Porto, Portugal, March. Springer-Verlag

Cestnik, B., and Karalic, A. (1991). The Estimation of Probabilities in Attribute Selection Measures for Decision Tree Induction. Proceedings of the Information Technologies Interface, ITI-97, Cavtat,Croatia, June

Charniak, E., and McDermott, D. (1985). Introduction to Artificial Intelligence. Addison-Wesley

Cohen, P.R., and Feigenbaum, E. (eds.) (1992). The Handbook of Artificial Intelligence, vol. III, sec.XIV (written by T. Dietterich), 323-494

DeJong, G.F, and Mooney, R.J. (1986). Explanation-Based Learning: An Alternative View. MachineLearning, 1:145-176 de Raedt, L. (1992). Interactive Concept-Learning and Constructive Induction by Analogy. Machine Learning, 8:107-150

Duda, R.O., and Hart, P.E. (1973). Pattern Classification and Scene Analysis. New York, John Wiley &Sons

Dzeroski, S., and Todorovski, L. (1994). Discovering dynamics. J. Intelligent Information Systems

Esposito, F., Malerba. D., and Semeraro, D. (1993). Decision tree pruning as a search in the state space. Machine Learning: ECML-93 (ed. P. Brazdil), Proc. European Conf. Machine Learning, Vienna,April

Everitt, B. (1981). Cluster Analysis. London, Heinemann

Fayyad, U.M., and Irani, K.B. (1992). On the Handling of Continuous-Valued Attributes in Decision Tree Generation. Machine Learning, 8:87-102

Fisher. D.H. (1987). Knowledge Acquisition via Incremental Conceptual Clustering. Machine Learn-ing, 2:139-112

Fisher, D.H., Pazzani, M.J., and Langleym, P. (eds.) (1991). Concept Formation: Knowledge and Experience in Unsupervised Learning. San Mateo,

Morgan Kaufmann

Gennari, J., Langley, P., and Fisher, D. (1989). Models of Incremental Concept Formation. Artificial Intelligence, 40:11-62

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA, Addison-Wesley

第2章　数据挖掘与知识发现：对问题和多策略方法的回顾

Ryszard s.uichalski 和 Kenneth A.kaufman

摘要

几乎在每一个人类活动中的各类数据库一直都需要新的强有力的工具，来把数据转换成有用的、面向任务的知识。为了努力达到这一目标，研究者们在机器学习、图像识别、统计数据分析、数据可视化、神经性网络等领域中探索着相应的方法。这些努力使得数据挖掘与知识发现出现了。本章的第 1 部分主要是关于符号化机器学习方法能力的概况；第 2 部分讲述概念化数据分析的多策略方法，该方法能够利用数据与背景知识，通过对数据的符号推理，获得更高层次的概念与描述。该方法目前已在 INLEN 系统中与机器学习、数据库和基于知识技术相结合而得以实现。为展示系统的能力，我们应用其解决一个具体问题并给出了其处理结果，该问题就是从一个包含世界上所有国家的基本情况与统计数据的数据库中，发现经济与人口统计模式。结果显示：该方法在帮助进行实际数据挖掘与知识发现这一方面具有很高的潜力。

2.1　前言

现今信息时代的主要特点就是伴随人类各种工作活动而来的数据超速增加。这其中越来越多的数据都被被保存到了数据库中，以便利用计算机技术进行存取。这些超大海量的可利用的数据就产生了一个问题：如何从中抽取出有用的、面向任务的知识？

传统用于解决这类问题的数据分析技术包括：回归分析、聚类分析、数

值分类学、多维分析、多变量统计方法、随机模型、时序分析、非线性估计技术及其他技术（如[DW80]、[Tuk86]、[MT89]、[Did89]和[Sha96]）。这些技术已得以广泛应用并解决了许多实际问题。但这些方法主要定位在从数据中抽取出定量和统计数据特征，因而存在着先天不足。

例如：统计分析可以确定变量数据间的协方差和相关系数，但是它不能对在一个抽象、概念层面对变量间存在的联系进行描述，并解释存在这种联系的原因。它也不能采用更高抽象的逻辑形式的描述与定律来合理解释所存在的联系。统计数据分析可以确定给定因子的集中趋势和变化，回归分析可以使一条曲线逼近一组数据点。然而这些技术不能产生规律的定性描述，或确定数据中没有明确给出内在因素的相互依赖性，也不能把在已发现规律的领域与另一个领域中的规律进行类比。

数值分类学技术能够产生一组实体的分类，并描述归属到同一个或不同类别中各实体间存在的数值相似性。但是它不能给出所创建各类别的定性描述，以及将这些实体归为同一类的原因，而且必须在数据分析之前，定义好相似性的描述属性以及相似性。此外，这些技术本身不依赖背景领域知识来自动地产生相关属性分析，并随着数据分析问题不同而确定相应的变化。

为解决上述所列举的问题，一个数据分析系统就必须包含相当数量的背景知识，并有能力完成涉及知识与数据的符号推理。总之，传统的数据分析技术为进行数据解释提供了便利，并对隐藏在数据背后的过程有所洞察。这些解释与透视都是构造数据库的人们所寻求的最终知识，但是这些知识并不是由这些工具所产生的，而是由人类数据分析所获得的。

为满足对能够克服上述局限性的新的数据分析工具需要，研究人员转向机器学习以寻求相关的思想和方法。机器学习领域是这一目的的、自然而然的思想来源，因为这一领域的研究本质就是要研究相应的计算方法，以便能够从事实数据和背景知识中获得知识。这方面相关的努力业已促成了一个新研究领域，即常被称为数据挖掘与知识发现的领域，如[Lbo81]、[MBS82]、[ZG89]、[Mic91b]、[Zag91]、[MKKR92]、[VHMT93]、[FPSU96]、[EH96]、[BKKPS96]和[FHS96]。

本章第 1 部分是关于将符号机器学习方法应用于数据挖掘与知识发现有关思想的概要描述。本章主要介绍从数值和符号数据中抽取出知识的有关方法，以及用于文本、语音和图像数据的许多数据挖掘技术（如[BMM96]、

[Uma97]、[CGCME97]、[MRDMZ97])。

本章的第 2 部分则介绍**概念化数据分析**（Conceptual Data Exploration）方法，该方法能从数据中抽取更高层次的概念与描述。该方法主要起源于机器学习的各种研究成果，并应用相应的各种方法与工具以完成面向任务的数据特征描述和泛化。这些特征描述将采用一种逻辑形式来加以表示，便于理解和帮助做出决策。所谓**面向任务**（Task-oriented）这里意味着对同一组数据集进行分析可能会产生不同的知识，因此挖掘方法会将当前任务与数据分析方法联系在一起。这样面向任务的应用自然就需要一种多策略的方法，原因就是不同的任务需要探索不同的数据分析方法，以及产生知识的运算符。

挖掘方法的目的就是采用与人类专家产生知识描述类似的形式，描述自己所产生的知识。这种形式可能会是多种描述的组合，如：逻辑的、数学的和图形的。主要要求就是：这些描述对于一个领域专家而言，应是易于理解和解释的，也就是它们应满足“可理解原则”[Mic93]。我们在研究多策略数据分析方面所取得的成果已在 INLEN 系统（[MKKR92]）中实现了。INLEN 系统包含了一组机器学习方法与工具，以及更多传统的数据分析技术。这些工具为用户提供了进行不同种类的数据分析和从数据库中抽取多种不同类型知识的功能。

INLEN 系统所采用的智能数据挖掘方法直接反映了当前数据挖掘与知识发现研究的目标。在这种情况下，对数据挖掘与知识发现概念间的差别进行说明或许是有益的，正如[FPS96]所描述的。基于这种差异，数据挖掘应用机器学习或其他方法以“列举数据中的模式”；而知识发现则是指整个数据分析生命周期，从数据分析目标确认、初始数据获取与组织，到产生潜在有用的知识，知识的解释及其测试。根据这些定义，INLEN 中的挖掘方法是将数据挖掘与知识发现技术结合在一起了。

2.2　机器学习与多策略数据分析

本节将展示机器学习领域与数据挖掘和知识发现目标之间存在的紧密联系。特别是，将要介绍符号机器学习方法是如何应用于自动或半自动概念化数据分析任务的，以便从数据中挖掘出面向任务的知识。以下就简要介绍这些方法。

2.2.1 从具体实例中抽取通用规则

基于多策略数据的分析工具中主要的一类就是利用对数据进行符号归纳学习的方法。给定不同决策类别的样本（或一个关系实例），以及与问题相关的知识（“背景知识”），归纳学习方法先为每个类别假设一个通用描述。一些归纳方法利用一个固定标准来帮助从大量可能中选择一个描述，而另一些则容许用户自己定义一个能够反映当前问题的标准描述。描述的形式可以是一组决策规则、一个决策树、一个语义网等。决策规则可以采用多种不同的形式。这里我们假设采用以下的形式：

$$\text{CLASS} \Leftarrow \text{CONDITION}$$

这里 CLASS 是将要赋给一个实体（一个对象或情况）以类别、决策或概念名称的陈述，该实体（一个对象或情况）满足 CONDITION 指明的要求；CONDITION 包含一组基本条件的合取和蕴含符⇐，而基本条件则由描述对象的属性值组成。

我们还假设当 CLASS 需要一个析取描述时，就有若干与同一 CLASS 相应的（合取）规则。为说明这个问题，图 2.1 给出了一个在 EMERALD（一个大型展示机器学习与发现能力的系统[KM93]）中机器人角色类别析取描述的示例。

规则 ***A***：　类别 1 ⇐ 上衣是红色、绿色或蓝色，且头是圆形的或八角形的

规则 ***B***：　类别 1 ⇐ 头是方形的且上衣是黄色的

图 2.1　类别 1 的两个描述规则

如图 2.1 所示描述的意义就是：如果一个机器人的上衣是红色、绿色或蓝色，并且他的头是圆的或八角形的；或者，一个机器人的头是方的且其上衣是黄色，则他就属于类别 1（Class 1）。

上面提到的 EMERALD 系统，将五个程序结合到一起以展示不同的学习能力[KM93]。这些能力包括：从示例中学习规则（利用 AQ15 程序），学习不同结构间的差异（INDUCE），概念化聚类（CLUSTER/2），对象序列的预测（SPARC）和微分方程与描述物理过程数据的规则（ABACUS）。每个程序均可直接应用于概念化数据分析。例如：如图 2.1 所示的规则就是由 AQ15 规则

模块[MMHL86]产生的，而[HMM86]是从一组机器人角色类别 1“正例”和“反例”样本数据中产生出来的。

AQ15 学习实体的**属性**（Attributional）描述，即描述仅涉及有关的属性。而更加通用的描述，**结构**（Structural）或**关系**（Relational）描述，也仅涉及实体不同组成之间的关系，组成的属性和量词。EMERALD[Lar77]中的 INDUCE 模块[BMR87]可以产生这种描述。构造结构描述需要更为复杂的描述语言，该语言将包括多参数谓词，如 PROLOG 或注解谓词演算[Mic83]，[BMK97]。

从数据库分析角度来看，属性描述似乎最为重要，也最容易实现，因为典型的数据库是利用属性而非关系对实体进行描述的。一种简单而又常用的属性描述形式就是分类或决策树。这种树的结点对应属性，从结点引出的枝条对应属性的值，树的叶子对应类别（如：[Qui86]）。利用从根结点到各个叶结点的路径就可以将一棵决策树转换为一个规则集。通过检测规则中的多余条件而可以将规则简化（如：[Qui93]）。将一个规则集转换为一棵决策树的反过程并不是如此直观，因为规则的表达能力比一棵树的表示能力更强大。这里“更强大”一词意味着代表一组规则集的决策树或许需要多余的条件（如：[Mic90]）。

属性学习程序的输入包括一组具有各个类别的样本数据集和与给定学习问题相关的“背景知识”（简称 BK）。数据样本（决策实例）是用与各决策类别相关联的**属性**-值向量形式表示的。背景知识通常是描述各属性合法取值、属性类型（测量尺度）和选择可能结果假设的**倾向标准**（Preference Criterion）的相关信息。这种标准，或许代表描述的计算简洁**性**，并且/或者估计其预测的准确性。作为背景知识的补充，一种学习方法可能还有表示的倾向性，也就是说，它可将表达式的形式限定为一些特定的表示形式，如：单个合取、决策树、合取规则集或 DNF 表达式。

在一些学习方法中，BK 可以包含更多的信息，如对不同属性间的相互关系进行约束，产生更高层次概念的规则、新属性，以及一些初始假设[Mic83]。学习获得的规则通常完全地与输入数据是一致的，这也就意味着学习所获的规则能完全且正确地对所有“训练”数据进行分类。本章第 2.5 和 2.8 节将介绍归纳概念学习程序 AQ15c[WKBM95]中有关一致和完全数据样本的解决方法。在一些应用中，特别是涉及从有噪声数据中学习获得规则或学习获得灵

活性概念[Mic90]，因此能学习获得不完全和/或不一致的描述，或许很有价值[BMM92]。

通过将属性描述映射到（包含给定属性）离散多维空间中的平面表示而实现属性描述的可视化[Mic78]，[WSWM90]。图 2.2 就是图 2.1 中的规则可视化简图。图 2.2 的简图是利用概念可视化程序 DIAV 产生的[WSWM90]，[Wne95]。

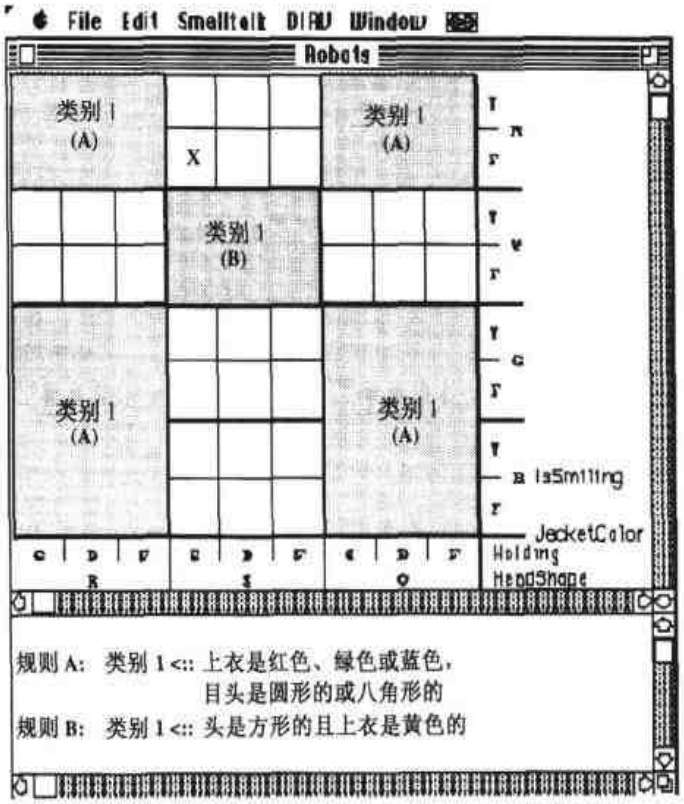


图 2.2　图 2.1 的规则可视化

图 2.2 中的每个单元代表属性取值的一个特定组合。如标记 X 的单元代表向量：（HeadShape=Square, Holding=Sword, JacketColor=Red, IsSimiling=False）。标记上类别 1（A）的四个阴影区域就表示规则 A，标记上类别 1（B）的阴影区域就表示规则 B。在这个简图中，合取规则对应配列规整且易于识别的单元区域[Mic78]。

可视化简图可用于显示（学习获得的）**目标概念**（Target Concept），训练样本（概念正例样本和反例样本），以及利用某种方法学习获得的实际概念。通过比较目标概念与学习获得概念，就可以确定**错误区域**（Error Area），也就是包含所有无法由学习获得的概念正确识别的样本。这样一种可视化简图的

方法能够表示出任何属性学习过程[WSWM90]。

下面两种类型的数据分析操作以从样本中学习获得概念的描述方法为基础：

- 确定一个数据集中的一组实体或指定实体组所对应的通用符号描述的操作运算。这种描述表示了每组中实体的常见性质。操作运算可借助构造归纳（稍后将要介绍）来利用数据中并不直接存在的抽象概念。这些操作运算均是基于学习获得的特征化概念描述的程序。
- 确定不同实体组之间差异的操作运算。这些差异可作为能有效区别不同实体组属性的规则。这些操作是基于学习获得的**差异性概念描述**的程序。

第 2.5 节将会阐述这两类操作。有关细节和定义请参见[Mic83]。概念学习的基本方法均假设样本没有错误，所有样本属性取值均存在。所有样本均存放在同一数据库，并且待学习的概念有一个准确（“脆弱”）地不随时间变化的描述。在许多情况下，一个或多个假设或许并不存在，这也就导致了许多更为复杂的机器学习和数据挖掘的问题。

- 从不正确的数据中学习（Learning From Incorrect Data），也就是：从包含一定错误或噪声的数据中进行样本学习（如：[Qui90]，[MKW91]）。这些问题对于从复杂现实世界观察中进行学习是非常重要的，因为这种情况下数据往往包含一定数量的噪声。
- 从不完全的数据中学习（Learning From Incomplete Data），也就是从某些数据属性没有取值的数据集进行样本学习（如：[Don88]，[LHGS96]）。
- 从分布式数据中学习（Learning From Distributed Data），也就是需要对若干分布的数据集合进行整合，以揭示其中存在的模式（如：[RKK95]）。
- 学习获得不定或渐变的概念（Learning Drifting or Evolving Concepts），也就是学习获得随机或在一个一般的特定方向上，不稳定的或随时间变化的概念。例如：一个用户的“兴趣领域”常常就是一个演变的概念（如：[WK96]）。
- 对随时间不断增加数据进行概念学习（Learning Concepts From Data Arriving Over Time），也就是增量学习，即学习获得的概念或许需要进行更新以适合新的数据（如[MM95]）。

- 对有偏差的数据进行学习（Learning From Biased Data），也就是从不能正确反映事件实际分布的数据集中进行学习（如：[Fee96]）。
- 学习获得灵活性的概念，所谓灵活性概念就是指本质上没有准确的定义且其含义是上下文相关的。与这一主题相关的思想包括：模糊集合（如[Zad65]，[DPY93]）、双层次概念表示（如：[Mic90]，[BMMZ92]）和粗糙集（如：[Paw91]，[Slo92]，[Zia94]）。
- 学习不同泛化层次的概念，也就是学习涉及代表不同抽象层次背景知识的概念（如：[KM96]）。
- 集成定性与定量的发现，也就是确定逼近一组数据点的方程集合，以及适应这些方程应用的定性条件（如：[FM90]）。
- 定性预测，即发现序列或过程中的模式且利用这些模式，定性预测给定序列或过程可能的连续性。

以上每个问题均涉及从收集而来的数据中（静态或动态）抽取出有用的知识。因此，解决这类问题的机器学习领域方法与数据挖掘和知识发现，特别是概念数据分析，直接相关。

2.2.2 概念聚类

与数据挖掘和知识发现相关的另一类机器学习方法，就是针对一个给定实体集，解决建立相应的概念分类问题。这个问题与传统聚类分析中的问题类似，但它是采用一种不同的方式来加以定义的。给定某些实体的一组属性描述，一种描述这些实体类别特性的语言和一个分类质量标准，即将实体集分为若干类别以便使得分类质量评估标准最大化，并且同时用给定描述语言确定这些类别的通用（扩展）描述。因此概念聚类方法不仅需要发现一个实体（系统树图）集的分类结构，而且还需要给出所获类别（聚类）的符号描述。概念聚类与聚类分析不同且特别之处就是：在确定每个类别（聚类）的过程中考虑类别描述特性。

为了解概念聚类与一般聚类的不同，值得一提的就是一般聚类方法通常根据相似性计算来确定聚类，而相似性计算函数仅仅参与比较实体属性取值而不涉及其他因素，如下式所示。

$$\text{Similarity}(A, B) = f(\text{properties}(A), \text{properties}(B))$$

其中 A 和 B 为进行比较的两个实体。

与一般聚类相反，概念聚类程序根据**概念性可粘合**（Conceptual Cohesiveness）对实体进行聚类，其中计算函数不仅包括实体的属性，而且还包括其他两个因素：描述语言 L（Description Language）（系统用其描述实体类别）以及环境 E（Environment E）。它是一个近邻样本集合：

Conceptual cohesiveness(A, B) = *f*(properties(A), properties(B), L, E)

因此两个对象或许相似，也就是根据某种距离（相似性）度量时比较接近，它们具有较低的概念粘合，反之也可能。图 2.3 所示就是第一种情况的一个例子。点（黑圈）A 和 B 彼此比较“接近”，因此基于仅仅依靠点之间的距离时，它们将被放入同一个聚类中。然而这些点由于属于代表不同概念的构造而具有较小的概念粘合。一个概念聚类方法，若采用合适的描述语言，会与人们常做的那样，将图 2.3 中点聚类为两个“椭圆”。

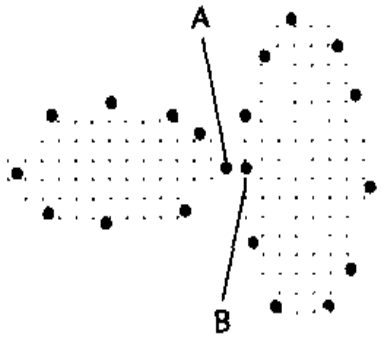


图 2.3　接近程度与概念粘合之间差异性的示意说明

在概念聚类中采用的分类质量评价标准可能涉及许多因素，诸如聚类描述相对数据而言的**适合程度**（又称细疏程度），描述的**简捷性**，以及实体的其他性质或描述它们的概念[MSD81]。第 2.5 节将介绍一个概念聚类的例子。

[CR95a]和[CR95b]文献中介绍了利用概念聚类对文本数据库进行处理的一些新思想，以及通过创建概念网格来发现数据中相互依赖关系的方法。通过聚类创造的概念在网格结构中相互连接，上下移动从而获得泛化和细化的关系。

2.2.3　构造性归纳

许多来自于示例样本的学习规则或决策树的方法，均假设描述样本数据的属性对于当前要解决的问题而言是充分的。这一假设实际上有时并不成立。描述样本的属性或许并不直接相关，一些属性或许是无关的或**不必要的**。符号方法优于统计方法的一个重要特点就是：符号方法可较为容易地确定无关的或不必要的属性。如果学习获得的类别或概念的完全和一致描述中不包含某个属性，那么这个属性就是不必要的。因此一个不必要的属性可以是无关的，也可以有关的，但是根据定义应是可有可无的。归纳学习程序，诸如规

则学习程序 AQ，或决策树学习 ID3，能够相对容易地处理输入数据中大量不必要的属性。

如果在样本初始描述中存在非常多的不必要属性，或许会大大增加其学习进程的复杂性。这种情况就需要能从初始给定的属性集中有效地确定出与给定问题最相关的属性。在描述学习进程中仅仅使用最相关的属性集。因此确定最相关属性集就是一种有益的数据分析运算。这种运算对于数据分析本身而言也是有益的，因为了解哪些属性最具有区分能力，对于一组给定类别而言是非常重要的。剔除相关程度较小的属性，就可以减小表示空间，从而问题就得以简化。因此这样一个过程可以视为一种表示空间的改善形式。[Zag72]和[Bai82]文献中介绍了一些发现最相关属性的方法。

在许多应用中，最初给定的属性可能是较弱的或与当前问题间接相关的。在这种情况下，就需要产生新的、更相关的属性，这些属性可能是初始属性的一个函数。这些函数或许简单，如一组初始属性的乘积或累加，或非常复杂，如由一个图像上是否存在直线或圆而确定的布尔值[Bon70]。最后，在一些情况下，可能希望抽象出某些属性，即将某些属性值组成一个单元，从而确定该属性可能的取值范围。将连续属性量化就是这种操作的一个例子。

所有以上操作：除去相关程度较小的属性，增加相关程度更大的属性以及属性抽象，均可改善用于学习的初始表示空间的不同形式。一个学习过程包括两个（相互交错）阶段：一个与构造“最好”的表示空间有关，另一个与在学习空间中产生“最好”的假设有关，后者被称为**构造性归纳**（Constructive Induction）[Mic78]，[Mic83]，[WM94]。AQ17[BWM93]就是其中的一种构造性归纳程序，它可以完成三种类型改进初始表示空间的操作。在这个程序中，通过对初始属性进行数学与/或逻辑运算可产生新的属性，并/或接受专家建议选择“最好”的运算组合[BWM93][BM96]。

2.2.4 选择最有代表性的样本

当一个数据库非常大时，确定表示不同概念的一般模式或规则可能很耗费时间。为使这一过程更加有效，从数据库中抽出最有代表性或最重要的给定类别或概念的实例（样本）是有用的。大多数这类情况或者最为典型，或者最为极端（假设数据不含太多的噪声）。确定后者的一种方法就是[ML78]文献介绍的所谓“显著表示方法”。

2.2.5　定性与定量结合的发现

在一个包含许多数值属性的数据库中，一个有价值的发现或许就是一个约束这些属性的方程。例如一个行星数据表，其中包含行星的质量、密度、离太阳的距离、自转周期、公转周期，从中可以自动获得开普勒定律，即行星与太阳的距离的立方与公转周期平方成正比。这就是定量发现的一个例子。将机器学习应用于定量发现的先驱就是 BACON 系统[LBS83]，之后出现许多这样系统，诸如 COPER[Kok86]，FAHRENHEIT[Zyt87]和 ABACUS[FM90]。Zagoruiko[Zag72]在经验预测中对解决类似问题的方法也进行了探索。

由于某常量可能有不合适的取值，或在不同定性条件下对应不同的方程，因此一些方程不能直接应用于数据。例如：应用 Stoke 定律确定球体下落速度，若球体在真空中下落时，其下落速度依赖于其下落的时间和作用于它的引力。而球体在某种流体中下落时，将达到某个平衡速度，该速度取决于球体半径和球体质量，以及流体的粘性。

ABACUS 程序[Gre88]，[FM90]，[Mic91a]能够确定在不同定性条件下的定量定律。它将数据分为样本集合，每个集合属于有一个定量发现模块所能确定的不同方程，然后，定性发现模块确定表述各样本集合的条件/规则（如 Stoke 定律，规则依赖于下落所处于的介质）。

2.2.6　定性预测

大多数从样本数据中获得规则的程序是根据不同类别的对象确定相应的规则。一个概念的实例可以说明相应的概念而不论它与其他样本的关系。与一个序列预测问题不同的是：后者一个概念的正例直接依赖序列中正例所在的位置。如图 2.4 显示 7 个图形序列。其问题就是第 8 个位置可能是何种图形？要回答这样的问题，就需要搜索序列的一种模式，然后利用该模式预测可能的序列后继。在**定性预测**（Qualitative Prediction）中，问题不是预测一个变量的值（如时序序列），而是定性描述最可能的未来对象，也就是描述未来对象的可能性质。

在图 2.4 所示的例子中，读者或许能够观察到序列是由带黑色顶的 T 形与带白色顶的 I 形组成。图形可能是白色或有阴影，并或许向不同方向旋转 45

度。但有一个一致的模式吗？

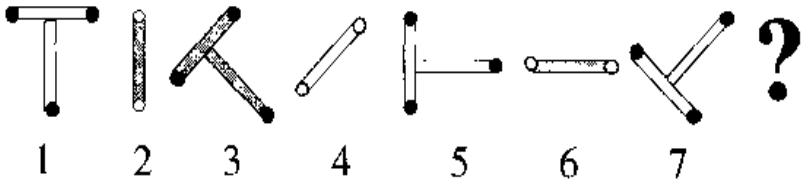


图 2.4 一个序列预测问题示意描述

为确定这样的模式，可以利用不同的**描述模型**（Descriptive Models），进行模型实例化以适合特定的序列。然后最适合数据的实例化模型就可用于预测。[DM86]文献中介绍了这种方法。该方法利用了三种描述模型：周期性、分解和 DNF。

周期性模型（Periodic Model）被用于检测序列中重复出现的模式。例如：图 2.4 描述了一个由 T 形对象和 I 形对象交替出现的重复模式。一般在一个周期性序列中还会有周期性的序列。在图 2.4 中，T 形对象构成一个序列，其中每个对象均逆时针旋转 45 度。

第二个模型，**分解模型**（Decomposition Model），被用于通过决策规则来刻画一个序列，其描述形式："如果一个序列中前面一个或更多元素具有一组给定的性质，那么下一个元素就会有下列的性质。"将一个这样的规则应用于图 2.4 就可得到：如果序列中的一个元素具有垂直分量，那么序列中下一个元素就会有阴影分量；否则它就不会有阴影分量。

第三个模型，DNF（析取范式）或"全能"模型，试图抓住刻画整个序列的通用性质。例如对于图 2.4 中的序列，它将会得出这样的结论："序列中所有元素是 T 形或 I 形，它们内部是白色或有阴影，并具有白色或黑色顶"等。

SPARC/G 程序[MKC86]利用这三个描述性模型来检测由任意对象组成的序列中的模式，然后利用这些模式来预测一个可能存在的序列后继。对于如图 2.4 中的序列，SPARC/G 根据周期性模型可发现以下强模式：

Period<[Shape=T-shape]&[orientation(i+1)=orientation(i)-45],[Shape=I-shape]&[orientation(i+1)=orientation(i)+45]&[Shaded(i+1)=unshaded(i)]>

模式可被解释为：一个重复周期中（用逗号隔开描述）存在两个部分，第一部分包括 T 形物体，而第二部分包含 I 形物体。T 形物体逆时针旋转，而 I 形物体相对它的前一个 I 形物体顺时针旋转 45 度。I 形物体阴影与无阴影交替出现。根据这个模式，序列中第 8 个位置可能会出现的物体为相对前一 I

形个物体顺时针旋转 45 度的无阴影 I 形物体。

以上介绍的定性预测能力可用于许多应用领域中时序数据库的概念分析，诸如农业、医药、机器人、经济预测等。

2.2.7　基于机器学习方法的总结

为帮助读者对以上所介绍的各方法之间的不同与新奇之处能够形成一个大致印象，这里将对传统多变量数据分析方法所能完成的典型操作做一总结。这些方法包括：计算均值或标准化变量，方差，标准方差，协方差和属性间相关性；主成分分析（确定正交属性线性组合以解释大部分的方差来源）；因素分析（确定相关程度较高的属性组）；聚类分析（根据某种度量方法确定相近数据点集合）；回归分析（描述一组数据点的拟合方程）；多变量方差分析；差异分析。所有这些方法可被视为对一个数据集的数值特性进行描述。

与之相反，前面介绍的机器学习方法主要对数据进行符号逻辑形式的描述，这种描述或许是定性描述一组或多组数据，不同类别之间的不同（由指定输出变量不同值所确定），产生数据的一个“概念”分类，选择最具有代表性的实例，定性预测序列数据等。这些技术特别适合于产生数据中涉及符号和等级变量的描述。

两种数据分析方法的另一个主要不同之处就是：统计方法常常用于全局性地描述一类对象（数据表），但不是用于描述对象未来所属哪个类别。例如：一个统计运算或许可以确定某种型号汽车平均寿命为 7.3 年。但有关特定类别汽车平均寿命的知识并不能帮助人们确定这样一辆汽车是否还可以继续使用，即使是可以获得相关汽车寿命信息。相反，一种符号机器学习方法或许可以产生一个描述，诸如：“如果一辆汽车的前部高度在 5 到 6 英尺，司机座位离地面 2 到 3 英尺，那么这辆汽车可能就是一辆小型货车。”这类描述特别适合用于根据实体属性将其归为相应类别的场合。

INLEN 方法结合了许多机器学习研究中的数据分析策略与运算，以及统计运算。采用这种多策略方法的原因就是一种数据分析或许对多个有关数据的不同信息感兴趣。不同类型的问题需要不同的分析策略和运算方法。

2.3 数据分析任务中的分类

以上介绍的问题可以用**通用数据表**（General Data Table，简称 GDT）来加以简单表示。这种数据表是数据分析所使用标准数据表的一个通用形式（如图 2.5 所示）。它包括一组关系表（数据表），按照与每个表中事例相关的时间分层进行摆放。利用 GDT 可以表示一系列随时间而变化的实体序列。一个 GDT 的例子就是一个病人医疗记录序列（每个记录均用一个数据表来表示其一系列的测试结果）；一种庄稼在田里生长时的一系列描述；记载一个公司在选定时间内状态的一系列数据等。

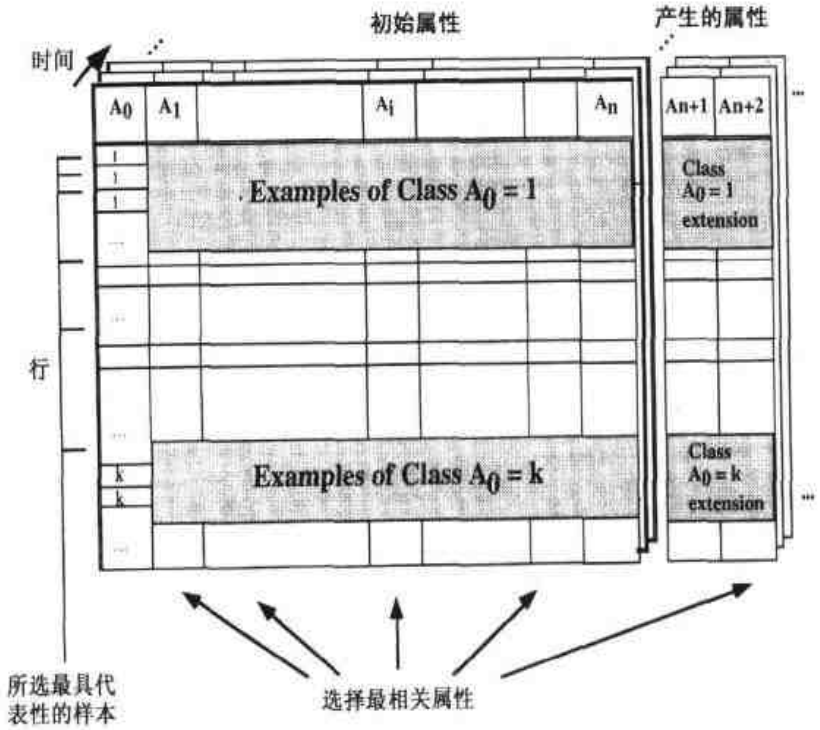


图 2.5 一张说明不同符号运算作用的 GDT

数据表中的列对应于表示实体的属性，而行与实体相关联。列对应的属性可能是初始属性，**事先**给定，或是通过构造性归纳（如[WM94]）过程而产生的附加属性。每个属性都会被指定值域和类型。值域是定义数据表中对应属性可以取的所有合法值。而类型则定义值域中各个取值的次序（如果有的话）。例如，AQ15 学习程序[MMHL86]容许属性有四种类型：名字属性（无序）、线性属性（全有序）、循环属性（循环有序）和结构属性（层次有序，请参见[KM96]文献）。属性类型决定了在学习过程中对相应属性的运算操作类型。

数据表中每一行输入的是与之相关联实体各属性的取值。通常，每行对

应一个实体。但是在大型数据库中，其每条记录代表普通的、可重复的交易，可增加一列以表示相应交易出现的次数。利用这些信息，发现工具就可以根据实例出现的频率进行有针对性的发现。

数据表中不同列的位置对应相应属性的取值，记号“?”表示特定实体相应属性的一个未知取值，若一个属性在一个特定实体中不存在，则用记号 N/A。例如“腿数”通常用于描述动物，而对植物则不合适。

概念数据分析中的一个重要问题就是确定数据表中的属性是否与其他属性相关。一个相关问题就是确定这种关系的通用形式，以便能够利用这种关系预测未来实体的某些属性值。例如：在知道符号属性依赖其他（独立）属性时，问题就是产生一个有关这种联系的通用描述，以便之后可以根据其他独立属性取值预测符号属性的取值。该问题等价于根据样本学习概念的问题，因此机器学习研究出来的方法可以直接进行应用。这种情况下，数据表中代表输出属性（Output Attribute）的列就代表非独立属性。该变量取值就是将要学习其描述的类别。如图 2.5 所示，其中假设第一列（属性 A_0）代表输出变量的取值。在实体没有先验类别时，就不会有这样的指定列了。这种情况下，就需要利用概念聚类方法来确定实体分类的情况。

下面，我们利用 GDT（图 2.5）将前面所介绍的机器学习方法应用到数据分析问题中。

从样本数据中学习规则

假设 GDT 中的一个离散属性被指定为输出属性，而其他所有属性被作为（独立的）输入属性。表中一组输出属性取相同值的行，可被视为一组被此值所表示的决策类别（概念）的训练样本数据。这样，任何经典的概念学习技术均可直接用于确定输入与输出之间关系的规则。对于数据集通用分析而言，每个离散属性（以及连续属性离散化后）均作为输出属性，可应用任何一个机器学习方法来确定该属性与其他属性之间的关系。有多个规则评价标准来帮助指导确定这样的关系（规则），如简单性、代价、预测准确性等。INLEN 系统利用 AQ 学习方法正是因为它简单，而且它产生的决策规则易于理解[WKBM95][BM96]。

确定时间相关的模式

这个问题是关于从数据表中的一个按照时间维进行排列的数据序列中检测出时序模式。在那些可以用于分析这样时序数据的新奇思想中，有的是用

于定性预测的多模型方法[DM86]、[MKC85]、[MKC86]。还有时序构造性归纳技术，它可以产生新属性以便能够描述与时间相关的模式[Dav81]、[BM96]。

样本选择（Example Selection）

这个问题就是从数据表中选择各类别最有代表的数据行。当一个数据库非常庞大时，非常有必要对其中具有代表性的数据样本进行分析。“显著代表方法”就是选择与其他样本最大不同的样本（行）[ML78]。

属性选择（Attribute Selection）

当 GDT 中有许多列（属性），就需要通过消去与当前指定学习任务最不相关的属性来减小数据表规模。可以利用许多属性选择方法之一来完成这一任务，如 Gain Ratio[Qui93]。

产生新属性（Generating New Attributes）

这个问题就是对应一个构造性归纳过程所产生的新属性而增加对应的列。利用问题的背景知识和/或构造性归纳文献，如[BWM93]所介绍的特别启发知识，可以帮助产生这些新属性。

聚类（Clustering）

这个问题就是将数据表的各行自动地分为若干组，分别对应“概念聚类”，也就是具有较高概念粘合的实体集[MSD81]。这样的聚类运算将在数据表中产生新的一列对应一个新属性“类名”。数据表中该属性的取值就表示实体所归属的类别。描述聚类的规则在知识库中单独存放，并通过**知识片断**（Knowledge Segment）与实体建立联系（见第 2.4 节）。第 2.5 节给出了一个聚类的示例。

确定属性的依赖性（Determining Attribute Dependencies）

这个问题就是要确定相互关系，如对给定的 GDT，利用统计和逻辑方法可以获得属性（列）间的相关、因果依赖、逻辑或函数依赖关系。

规则的增量更新（Incremental Rule Update）

这个问题就是要更新工作知识（尤其是对描述 GDT 中属性间关系的规则集）以适应新事例或数据表中的时间切片。为实现这点，必须应用增量学习程序以便对原先的知识和新情况进行整合。增量学习过程，根据初始训练数据在增量学习过程中的保存情况，可分为全记忆（Full-memory）、半记忆（Partial-memory）或无记忆（No-memory）的[HMM86]、[RM88]、[MM95]。

在不完善数据中搜索合适的模式（Searching for Approximate Patterns Inimperfect Data）

对于一些 GDT 来讲，要发现其中完全和一致的描述或许是不可能的。在这种情况下，就需要确定能够满足大多数实例的模式，而无需对所有实例均满足。这个问题的一个重要情况就是数据表中的一些入口没有确定取值或取值不正确，这时就需要确定最好情况（也就是可能的情况）的假设以满足当前的大多数数据。

填补丢失的数据（Filling in Missing Data）

若给定数据表中的一些入口没有数据，就需要分析当前已知的数据以确定这些数据入口可能的取值。一个有趣的方法就是根据人类似然推理中的核心理论，应用多行推理解决这个问题[CM81]、[Don88]、[CM89]。

从描述性知识中确定决策的结构（Determining Decision Structure From Declarative Knowledge （Decision Rules））

假设从给定数据集（GDT）中获得了一组通用决策规则（知识的一种描述形式）。若这个规则集将用于预测新的事例（通过一个计算机程序，或一个人类专家），或许就需要将其转换为决策树的形式（或更一般形式：决策结构）以适应特定的制订决策的环境（如考虑测量属性取值的成本）。[IM93]、[Ima95]和[MI97]文献中介绍了这项工作，以及赞成和反对这项工作（如同反对直接从样本学习决策树的传统方法）的有关方法。

完成上述数据表操作的方法已在各种机器学习程序中得以实现（如[MCM83]、[MCM86]、[FR86]、[Kod88]和[KM90]）。以下我们将要介绍 INLEN 系统，它旨在最终将所有这些程序整合成一个集成系统中的运算操作，以便从数据中产生知识。

2.4　INLEN 中各操作的集成

为使数据分析人员更加方便地使用以上介绍的数据分析运算操作，以及使得一个运算操作的输出可以作为下一个运算操作的输入，就需要将完成相应运算操作的程序集成到一个系统中。这个思想成为 INLEN 系统的基础[KMK91]、[MKKR92]、[MK97]。INLEN 一词取自推理（Inference）和学习（Learning）两个单词。系统将机器学习、统计数据分析工具、数据库、知识库、推理过程及各种支持程序集成于一个统一架构和图形界面之中。知识库用于保存、更新和应用规则和其他形式的知识，以帮助完成数据分析和根据

其分析结果发布分析报告任务。

INLEN 总体结构如图 2.6 所示。系统包括一个与知识库（简称 KB）相连接的数据库（简称 DB），以及一组运算操作。运算操作分为以下三类：

- DMOs：**数据管理运算操作**（Data Management Operators），它在数据库中运行。这些操作是传统数据管理运算操作，用于完成关系数据表的创建、修改和显示工作。
- KMOs：**知识管理运算操作**（Knowledge Management Operators），它完成对知识库的操作。这些操作功能与 DMOs 的操作类似，只是操作对象是规则和知识库中的其他结构。
- KGOs：**知识产生运算操作**（Knowledge Generation Operators），它完成对数据与知识库的操作。这些操作可完成符号与数值数据的分析任务。它们基于各种机器学习和推理程序，各种常规数据分析技术，以及图形化显示操作数据分析的结果。图解可视化方法 DIAV[Wne95]就是用于显示对数据的符号学习操作的结果的。

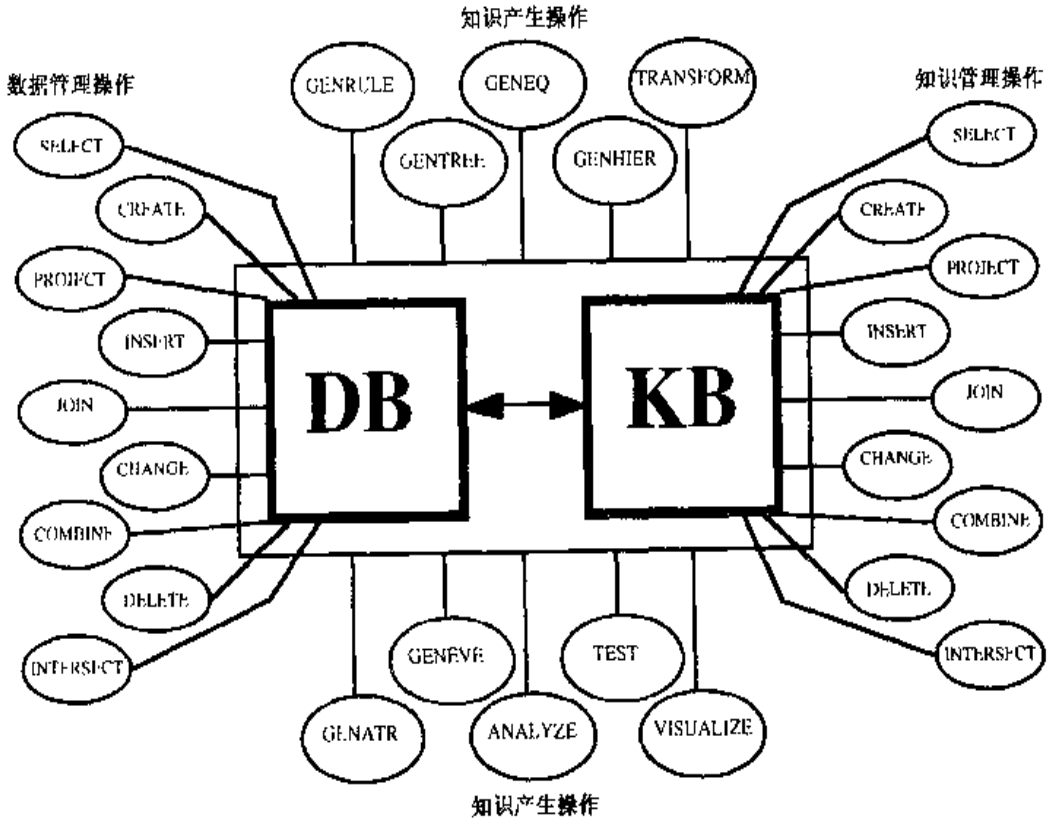


图 2.6　多策略数据分析 INLEN 系统的总体结构

KGO 是 INLEN 系统的核心。为方便其应用，引入了知识片断概念

[KMK91]、[MK97]。一个知识片断就是将一个或多个数据库关系表与知识库中一个或多个结构联系起来的结构。KGO 可以视为完成某些形式推理或知识片断转换的功能模块，因此它可以创建新的知识片断。知识片断既可作为 KGO 的输入，又可作为 KGO 的输出。因此它们可以帮助数据和知识从一个知识产生运算操作流转到另一个。

KGO 的执行通常需要某些背景知识，并由控制参数指导（若某些控制参数没有设置，则使用默认值）。背景知识或许包含某些一般知识，以及与给定应用领域相关的特别知识，诸如属性值域和类型定义，约束和属性间关系，专家给出的初始假定规则等。根据所完成的操作，可将 KGO 分为几组。每组包含若干带有一些参量的特定操作。这些基本运算操作组说明如下：

- GENRULE 运算操作从给定事实中产生不同类型的决策规则。一个特定运算操作或许可以产生某些规则以描述一组事实，描述不同事实组差异，一个事件序列，以及不同序列之间差异，利用诸如 AQ15c[WKBM95]和 SPARC/G[MKC86]程序完成这项工作。用于学习规则的 KGO 通常以批量或增量形式进行工作。在增量模式下，它将试图改进现有的知识，而在批量模式下，它将根据数据库的事实和知识库中的知识来试图创建全部新的知识。
- GENTREE 运算操作根据一组给定的决策规则（如[IM93]）或样本（如[Qui93]）建立一个决策结构。一个决策结构就是一个决策树概念的泛化，树中结点可被作为一个属性或一个属性的函数。每个分枝可被作为一组属性值。叶结点可被作为一组决策[IM93]、[Ima95]。
- GENEQ 运算操作产生描述数值数据集的方程，以及这些方程成立条件的定性描述（如[FM90]）。
- GENHIER 运算操作建立概念聚类或层次树。这些是根据 CLUSTER 程序的方法[MSD81]。INLEN 中的运算操作是根据 CLUSTER/2 的 C 程序重新改写的[Ste84]。
- TRANSFORM 运算操作，根据用户提供的标准完成各种知识片断的转换，如泛化或细化，抽象或具体，给定规则的优化等。例如：沿着一个属性的层次树向上进行泛化操作以构造一个更为通用的规则[KM96]。
- GENATR 运算操作通过创建新属性[BM96]，从初始属性集中选择最有

代表性的属性[Bai82]，以及通过属性抽象[Ker92]来产生新的属性集。

- GENEVE 运算操作产生满足给定规则的事件、事实或样本，从给定集合中选择最具有代表性的事件[ML78]，确定与给定样本相似的样本[CM89]，或利用一个专家系统命令或决策结构来预测一个给定变量的值。
- ANALYZE 运算操作分析数据中存在的各种关系，如确定两个样本间的相似程度，检查两个变量间是否存在内在联系等。统计与符号操作一样可以完成这些工作。
- TEST 运算操作测试一个给定规则集对于一组假定事实的性能。这些操作的输出结果是一个令人困惑的矩阵：一个数据表，其第（i, j）个元素表示类别 i 中有多少样本能够被类别 j 的规则所分类识别。这些操作运算可以用来将这些规则应用于任何给定的情况以确定一个决策。INLEN 中的 TEST 运算操作是基于 ATEST 程序的[Rei84]。
- VISUALIZE 运算操作利用一个方便、易懂的形式将数据和/或知识向用户展示[Wne95]。

总之，INLEN 将许多操作集成到一起以便能够完成对数据库、知识库或数据库与知识库组合的各种不同类型操作。

2.5 聚类和学习操作的说明

在 INLEN 所实现的知识产生运算中，产生数据分类（聚类）和利用其他属性学习与特定概念（属性）相关的通用规则是两个最重要的运算操作。前者利用 CLUSTER/2 概念聚类程序来实现[Ste84]。后者则是利用 AQ15c 规则学习程序[WKBM95]来实现的。本节通过一个硬盘驱动器数据库（如表 2.1 所示）的具体应用来说明这两个运算操作。这个数据库来自 1994 年 10 月所发行 MacUser 杂志上的数据。

在表 2.1 所示的数据表中，每行（除了第一行之外）利用第一行所列举的一组属性来描述一个硬盘驱动器。假设数据分析任务就是要获得一个分类知识以便将硬盘驱动器分为若干类别。现利用 CLUSTER 运算操作来完成这一任务。假设相应运算操作将寻找一种聚类，以便使得分类质量最好，其质量判断标准有两个：所产生类别描述的简单程度以及描述的粘合性（利用描述所

覆盖数据库中特定概念的实例数除以描述所覆盖数据库中的实例数而获得）。概念聚类运算操作的输入就是如表 2.1 所示的数据表（除去最右边的列，由于节省空间缘故，已经表示出了聚类结果）。

表 2.1　描述硬盘驱动器的数据表

Hard Drive	AC Outlet	SCSI 50-Pin	FCC Class B	Passwd Protect	Encrypt	5yr Warranty	Toll-free Support	Guarantee	Loaners	Capacity	Group
Apple 1050	no	yes	yes	yes	no	no	yes	by dealer	by dealer	low	1
Micropolis	no	yes	yes	yes	yes	yes	no	no	no	low	2
SEMO 1000	no	yes	Class A	yes	no	yes	no	no	yes	low	2
Focus 1G	yes	yes	yes	yes	no	no	yes	yes	yes	low	1
GHD 1200S	no	yes	yes	no	no	yes	no	no	no	low	2
Joule 1080	yes	no	yes	yes	no	yes	yes	yes	no	low	1
Liberty 1GB	no	25 pin	yes	yes	no	no	no	yes	yes	low	3
Spitfire 1GB	yes	yes	yes	yes	no	yes	no	yes	no	low	2
PowerUser 1020	no	yes	yes	no	no	no	yes	yes	no	low	1
P1000	no	yes	yes	yes	no	no	yes	yes	no	low	1
Seagate 3075	yes	yes	yes	yes	no	no	yes	yes	no	low	1
Minipak 1000	no	yes	yes	yes	no	no	no	yes	yes	low	3
PowerCity 1GB	yes	yes	yes	yes	no	on myself	yes	yes	no	low	1
Slim 1021	no	yes	yes	yes	no	yes	yes	yes	no	low	1
APS MS 1 7	no	yes	yes	yes	no	yes	yes	yes	no	high	1
Seagate 2GB	no	yes	yes	yes	yes	yes	no	no	no	high	2
SEMO 2000	no	yes	no	yes	no	yes	no	no	yes	high	2
Focus 2G	yes	yes	yes	yes	no	no	yes	yes	yes	high	1
FWB 1760MI	no	68 pin SCSI2	yes	yes	yes	no	no	no	If avail	high	3
Liberty 2GB	no	no	yes	yes	no	no	no	yes	yes	high	3
Laviel 1 2000	yes	yes	yes	yes	no	yes	no	yes	yes	high	2
Seagate 2 1	yes	yes	yes	yes	no	yes	no	yes	no	high	2
PowerUser 1801	no	yes	yes	no	no	no	yes	yes	no	high	1
MacPSc 28	no	yes	yes	yes	no	yes	no	yes	no	high	2

应用聚类运算操作的结果就是获得一个知识片断，它包含两个部分：一个新的拓展了的数据表和一组规则。与输入表相比，新数据表增加了一列，如表 2.1 所示标记为“Group”的最右边一列，它代表聚类运算操作所确定的相应硬盘驱动器的归属类别。第二部分就是描述所产生归属类别的规则集。以下是一个描述运算操作所产生归属类别的规则

[Class1]⇐[Toll_free_Support is yes]&[FCC_Class-B is yes]&[Encryption is no]&[SCSI_50-Pin is yes or no]&[Guarantee is yes or by dealer]

[Class2]⇐[Toll_free_Support is no]&[SCSI_50-Pin is yes]&[5yr_Warranty is yes]&[Guarantee is yes or no]&[Loaners is yes or no]

[Class3]⇐[Toll_free_Support is no]&[FCC_Class-B is yes]&[Ac outlet is yes]&[Passwd_Protect is yes]&[5yr_Warranty is no]&[Guarantee is not by dealer]&[Loaners is yes or if available]

因此运算操作产生硬盘驱动器的三个归属类别，并用规则形式描述每个归属类别。每个规则说明了一个特定归属类别的所有共同属性，也就是说，它代表了一个归属类别的**特征描述**（Characteristic Description）[Mic83]。（注意规则中的某些条件似乎是冗余的。例如：类别 2 规则中的最后一个条件表示：Loaners 是 yes 或 no。这可以通过第三个值“by dealer”的存在来加以解释，即 guarantee 和规则均不会是 loaner）。这些特征并没有刻画出给定类别与其他类别之间最明显的区别。

为了创建一个刻画不同类别之间最大区别的描述，就需要利用运算操作来创建**差异描述**（Discriminant Description）[Mic83]。将运算操作（GENRULE）应用于如表 2.1 所示已拓展的数据表，其中利用“Group”列作为其输出属性。以下就是所获得的一组新决策规则：

[Class1]⇐[Toll_free_Support is yes]

[Class2]⇐[Toll_free_Support is no]&[5yr_Warranty is yes]

[Class3]⇐[Toll_free_Support is no]&[5yr_Warranty is no]

这里所获得的规则要比 CLUSTER 运算操作所获得（有关三个归属类别）的规则更为简单和易于理解。原因就是差异描述仅仅列举了将特定类别与其他类别区别开的那些特征。差异描述只利用了区别不同实体类别的最少信息。相对与表 2.1 所示的所有数据样本，特征描述与差异描述都是完全的、一致的，也就是它们均可以分类识别所有样本数据。

2.6 数据与规则的可视化

对于数据分析而言，将不同运算操作结果可视化，以便使输入数据与从其中学习获得的规则之间的联系直观明了，这是非常必要的，这样可以清楚地看出哪些数据点符合或不符合这些规则，从而发现可能的错误等。为了这个目的，INLEN 利用 DIAV 程序所实现的**图解可视化方法**来支持数据和知识的可视化[Mic78]、[Wne95]。

这里我们以前一节的硬盘驱动器分类问题为例来说明可视化方法。表示空间分为 6 个属性，如图 2.7 所示。为简化可视化问题，这里利用一些属性来横跨图，Toll_free_Support(tf)，Loaners(lo)，SCSI_50-Pin(sc)，FCC_Class-B(fc)，Guarantee(gu)和 5yr_Warranty(wa)，均为在由概念聚类运算操作所获特征描述

中出现最为频繁的属性。图中的每个单元对应一个由相应行列所确定属性的值。因此左上方单元与一个数据点对应，该数据点中 6 个属性均取 yes(y)值。

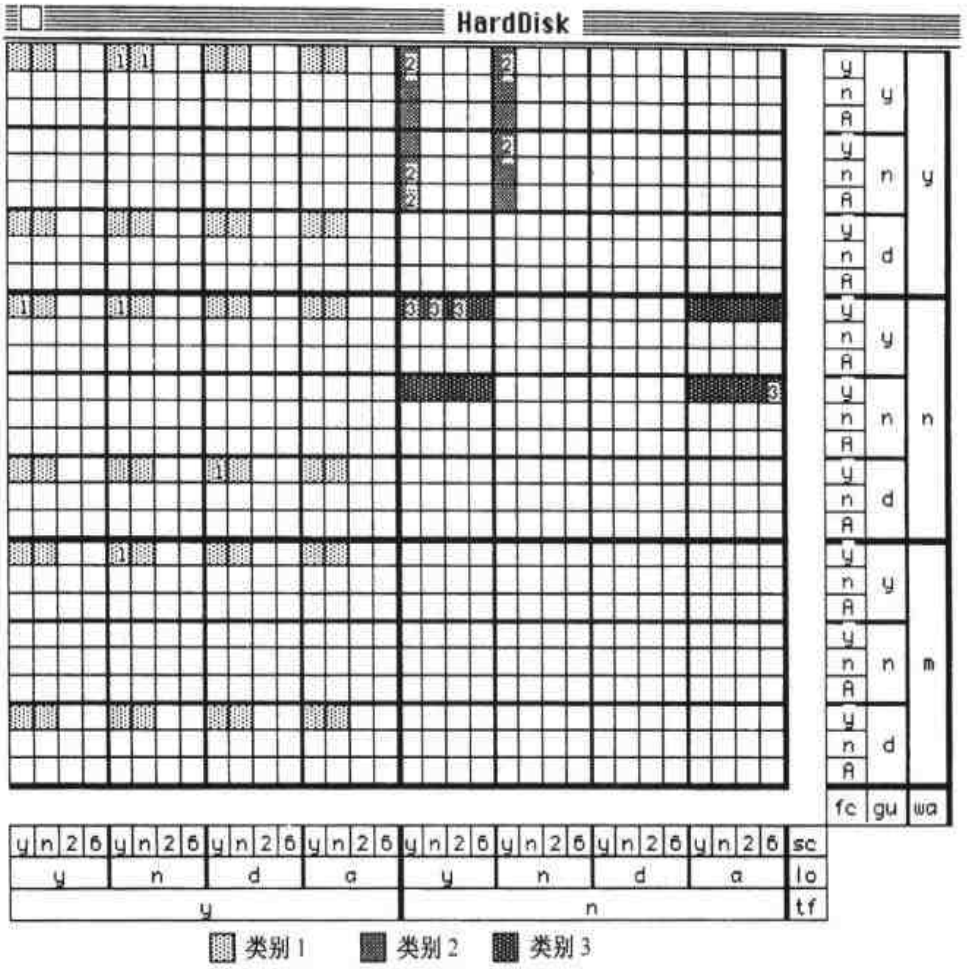


图 2.7　概念聚类运算操作创建的特征描述的可视化

将如表 2.1 所示的 24 个例子映射到这个空间，并将其类别号放到相应单元来加以表示。如图 2.8 所示，阴影区域表示由聚类运算操作所产生类别的特征描述：最淡的颜色代表类别 1，较深阴影颜色代表类别 2，颜色最深的阴影代表类别 3。正如表 2.1 所示，由聚类运算操作产生的描述就是输入实例的一般描述，因而它们覆盖了尚没有观察到的一些实例（没有编号的阴影区域）。

为了进行比较，图 2.8 给出了由规则学习运算操作处理输入样本所获得的差异描述的可视化说明，这些样本已根据先前产生的聚类进行了分类。图 2.8 所示图的结构与图 2.7 所示的结构，在样本、类别、行与列的标识方面均相同。因为差异描述主要集中在能够区分不同类别的特征上，它们会覆盖更广的表示空间区域。因此它们比特征描述更具有一般性。

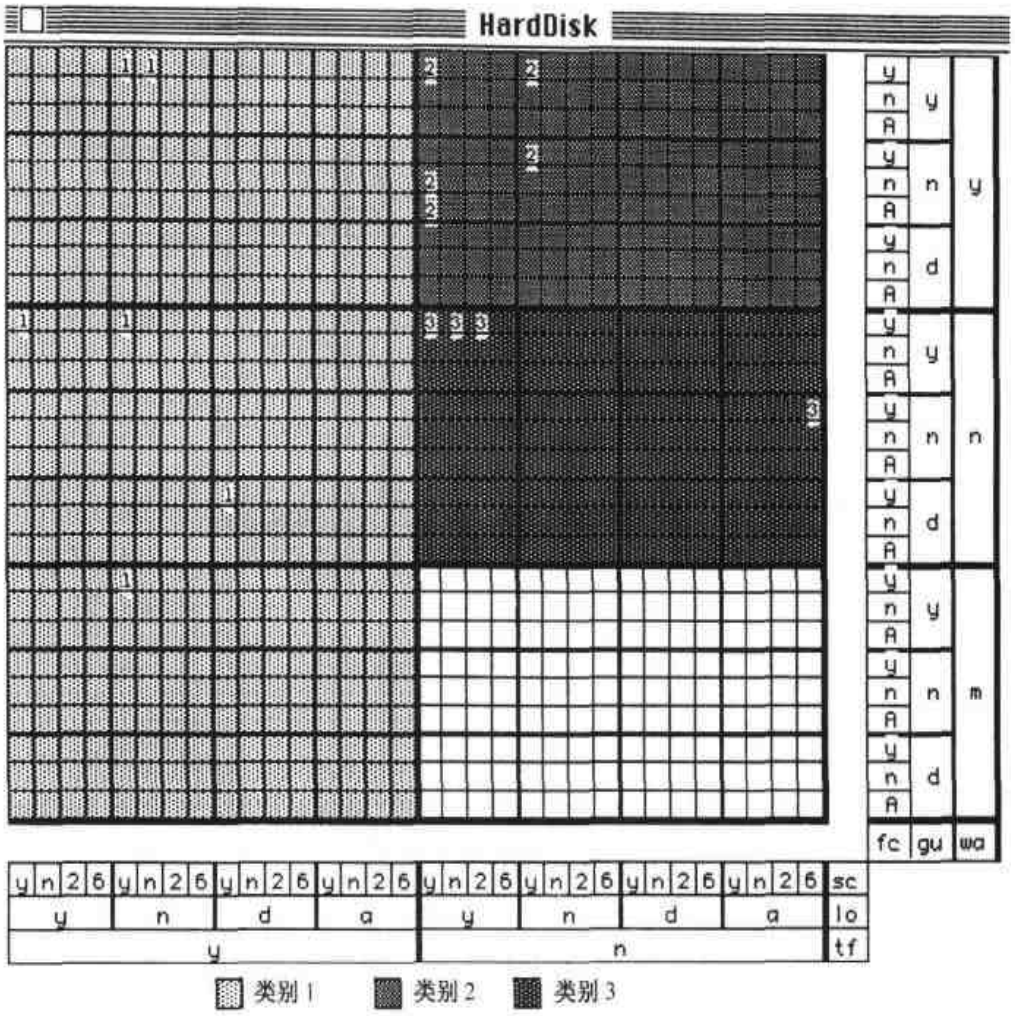


图 2.8　描述硬盘驱动器的数据库

所获得的差异描述将表示空间分为 4 部分，3 个部分与 3 个类别相对应，第 4 部分与事件空间中不确定的区域相对应，这一区域不包含 3 个类别的任何已知实例。后者部分利用属性组合来加以描述：Toll_free_Support = no 并且 5yr_Warranty = on_machanism。

同时也应注意到：由于差异描述具有更大的一般性，因此其不确定的区域比特征描述中的要小许多（图 2.7 中的空白区域）。

从图 2.7 和图 2.8 可以看出，所产生的差异描述对于当前所有样本而言是完全的、一致的，也就是说，它们保持了聚类运算操作所产生的分类情况。总之，以上所提出的可视化方法，使得人们更容易了解相应描述是如何产生的，从什么样的实例中产生的。

2.7　结构属性的规则学习

除了常规的符号与数值属性，INLEN 系统还支持一种新属性，称为**结构属性**（Structured）。这种属性取层次树中有序的值集[Mic80]。为在归纳学习执行过程中充分利用这种结构属性的性质，由此定义了一种新的泛化归纳规则。

一个泛化归纳规则（或变化）接受一个输入陈述及其相关的背景知识，并且假设一个更为通用的陈述[Mic80]、[Mic83]、[Mic94]。例如：从一个决策规则前提中除去一个条件就是泛化的变化（也称为**消去条件**（Dropping Condition）泛化规则），因为若规则前提包含更少条件，就会有更多实例满足这一规则。

AQ 学习程序所使用的一个功能强大的泛化归纳运算操作就是**相对扩展**（Extension-against）运算操作。若规则 R1：$C \Leftarrow [x_i = A]$ & CTX1 描述概念样本的一个正例子集，E+表示概念 C，规则 R2：$C \Leftarrow [x_i = B]$ & CTX2 描述概念样本的一个反例子集，这里 A 和 B 表示 x_i 不相交的值域子集，而 CTXs 则表示任何其他条件，那么规则 R1 相对规则 R2 沿 x_i 的扩展就表示为：

$$C \Leftarrow R1 \text{—|} R2/x_i$$

这样就产生了一个新规则 R3：$[x_i \neq B \cup \varepsilon]$，它是规则 R1 的一个**一致性泛化**，也就是说：这是一个逻辑上不与规则 R2 相交的泛化[MM71]、[Mic83]。参数 ε 控制着泛化的程度。若 ε 为空集 ϕ，那么规则 R3 就是规则 R1 的**最大一致性泛化**。若 ε 为 $D(x_i) \setminus (A \cup B)$（其中 $D(x_i)$ 为 x_i 的值域），那么规则 R3 就是规则 R1 仅涉及 x_i 的**最小一致性泛化**。在 AQ 程序中，相对扩展运算操作通常采用 $\varepsilon = \phi$。

反复运用相对扩展运算操作，直到所产生的规则不再覆盖任何反例样本，这时就可以获得一个一致性概念描述（不再覆盖任一个反例样本）。可以利用这样一个过程来产生针对所有样本的完全且一致的概念描述（覆盖）。

利用控制参数 ε 的不同取值进行相对扩展运算操作，就能获得不同泛化程度的描述。例如：在 AQ15c 程序中，为了学习获得一个特征规则，参数 ε 取 ϕ 的运算操作输出就是一种最小的泛化，以使得它能够通过不断扩展而覆盖所有的正例样本。若需要差异规则，就需要最大程度的泛化，只要它能够不覆盖概念的任何反例样本即可。

为了能够有效地应用相对扩展运算操作来处理结构属性，就需要定义一

种新的运算操作。这里我们以一个包含结构属性“Food”的例子来说明这一问题，如图 2.9 所示。每个非叶结点表示一个概念，它比其子结点更要广义。这些关系需要在对给定事实进行泛化时加以考虑。假设要学习的概念可以用以下语句来示意说明：“John 吃剥皮牛排”和“John 不吃香草冰淇淋”。那么对相应事实就有许多一致的泛化描述，例如：“John 吃剥皮牛排”、“John 吃牛排”、“John 吃牛肉”、“John 吃肉”、“John 吃肉和蔬菜”，或“John 除了香草冰淇淋不吃，其他什么都吃”。第一句代表了最人细化描述（没有进行泛化），最后一句代表最大程度的泛化，其他代表了中间某种程度的泛化。如何在特定情况下确定最大兴趣泛化就自然成为一个问题。我们主要是借鉴人类推理的内在机理来解决这一问题。

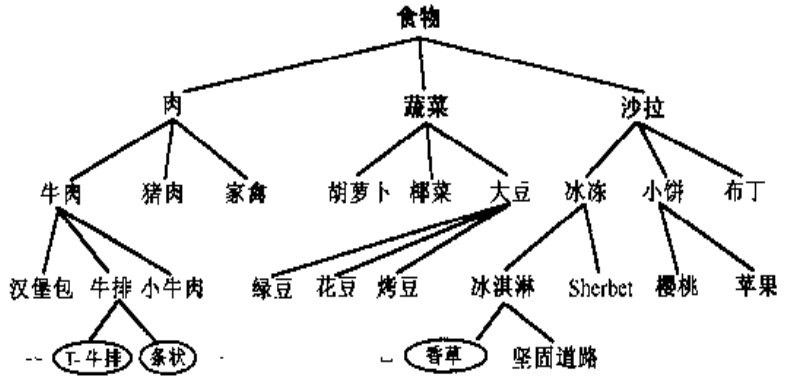


图 2.9　一个结构属性“食物”的应用领域

认知科学家已经注意到：在创建描述时，人类对泛化层次树（概念）中某些结点的偏好要胜过其他结点（如：[RMGJB76]）。影响一个概念（结点）选择的因素包括概念的典型性（概念特征在其兄弟概念中的常见情况）以及使用概念的上下文情况。例如，在看到一个知更鸟（一种鸟），我们或许会说“有一只鸟”，而不会说“有一只知更鸟”，假设给定情况下不需要说明鸟的种类。另一方面，当我们看到企鹅时，一种更少见的鸟，我们极有可能说“有一只企鹅”，而不是说“有一只鸟”。这样一个听众（未看过）就不会把鸟的典型特征，而仅仅是企鹅的特殊特征赋予未见过的鸟。这样就会使交流更加顺利。这里上下文也起着作用，比如在养鸟人聚会时，知更鸟就可能不再被称为是一只鸟而是用它的分类名称来称呼它了。

为了提供利用这种偏好的某种机制，INLEN 容许用户在层次树中定义**路标结点**（Anchor Node）。这类结点应反映特定应用的兴趣所在[KM96]。这里再以图 2.9 为例为说明这一点。在层次树中，香草和坚固道路是两种类型冰淇

淋，冰淇淋是冰冻点心，也是点心，也是一种食物。在每天使用中，取决于上下文，我们通常说香草或坚固道路是冰淇淋或点心，但很少说冰冻点心或食物。因此我们可以指定冰淇淋或点心作为 Food 层次树中的路标结点。利用有关路标结点的信息，就可以定义不同的规则偏好，诸如选择具有最一般路标结点的规则，或者能将正例样本泛化至其次的路标结点。

INLEN 支持将结构属性用做**独立**（Independent）（输入）和**关联**（Dependent）（输出）变量。独立的结构属性代表用于描述实体的层次值。关**联**的结构属性代表对实体所做的决策层次或分类。利用结构属性，INLEN 学习方法能够确定不同泛化程度的规则。

与**独**立属性处理类似，关联属性原则上可以充当不同的类型（符号、线性量、循环或结构），在实际应用中，它们常常是符号或线性量。概念学习中遇到最多的就是符号输出属性；它的值就代表要学习的概念或类别。线性输出属性（它通常就是一个比例标尺的测量值）用于表示需要根据过去数据进行预测的测量值。

在许多应用中，都希望利用一个结构属性作为关联变量，例如，在确定要买哪台个人电脑时，就需要首先决定电脑的类型，是 IBM PC 兼容机还是 Macintosh 兼容机。确定类型之后，就要将注意力集中到所选类型中的具体型号上。以上两个层次的决策过程要比单一层次决策要更容易执行，因为后者需要从更大集合中直接选择一台电脑。

在一个关联变量结构化之后，学习运算操作首先集中在（结点）顶层值，并创建相应的规则。随后它在祖先结点上下文环境下相继创建相应子结点的规则。这个过程产生的决策树规则比从一个平面（名字）组织的决策属性中学习获得的规则要更加简单和易于理解。

2.8　从决策规则中学习决策结构

数据分析的一个重要原因就是学习数据中的规则或模式以便使数据分析能够预测未来的情况。因此在学习获得这些规则之后，就需要有效利用这些规则进行预测。由于一个实现决策过程的常规结构就是决策树，因此需要解决决策树的转换问题。在传统机器学习方法中，决策树是直接从训练样本中学习获得的，因此就免去了首先要产生规则的步骤[HM86]、[Qui86]、[Qui93]。

从样本中直接学习可获得一个决策树，但是实际上，这存在较大的不足。决策树是知识表示的一种形式。一旦构造完成，就不容易被修改以适应决策制订环境的变化。例如，若一个属性（测试）被赋予给树中一个高层次结点，就不太可能进行测量或测量成本太高，则决策树仅仅提供了概率推理[Qui86]。

相反，人类做出决策时，可能会搜索不同的方法来执行。人们之所以可以这样做，是因为他们通常以定义形式存储了决策知识。根据知识的定义形式，诸如一组决策规则集，一个人通常可以构造许多不同但逻辑相同或近似的决策树。在某种决策制订环境中，一种这样的决策树可能比另一个要好。因此有必要先将知识存储起来，在需要时将其转换为最适合给定情况的过程形式。

决策树的另一个弱点就是由于它们有限的知识表示能力，而或许会变得不实用和不易理解。为克服上述限制，[Ima95]和[MI97]文献中研究出了一种新方法，该方法能够从决策规则中创建面向任务的**决策结构**（Decision Structures）。决策结构就是决策树的一种泛化，其中每个结点测试都不仅仅与单个属性有关，而是与多个属性函数相关联；分枝不仅与这些测试的单个值/结果相关联，而且也与一组这样的值有关；叶子不仅可以代表单个决策，而且也以某种概率代表一组可替换决策。

AQDT-2 程序已实现了这种方法，并采用一种 AQ 类型学习算法（AQ15c 和 AQ17-DCI）来从样本中确定决策规则。其优点就是有能力产生最适合特定任务的决策结构，并能避免或推迟测量代价高的属性。不同用户或许会希望从一组给定规则集中产生不同的决策结构，因此就要对结构进行修剪使其适应相应的不同情况。此外若一个属性很难测量，或根本不能测量，就需要构造程序以便避免使用这个属性，或仅在需要时才测量这个规则中的属性，并构造相应的决策结构。

这种方法的另一个优点就是：一旦确定了一个规则集，就可以从中产生一个决策结构，这样做比直接从样本中产生要快许多，因此处理时间就会少许多。另外，一组规则所占用的空间也比要学习的样本数据少。

AQDT-2 程序实验表明：从决策规则中学习而来的决策结构，要比从样本中学习获得的规则简单许多，而且常常还有较高的预测准确性。例如：AQDT-2 程序在一个抗风支撑设计问题中所学习获得的一个决策结构包含 5 个结点和 9 个叶子，其对新数据的预测准确率为 88.7%，而由流行的 C4.5 程序所产生的决策树包含 17 个结点和 47 个叶结点，其预测准确率为 84%[MI97]。在另一个

实验中，由 AQDT 分析国会选举的数据，从决策规则中学习获得的决策结构包含 7 个结点和 13 个叶结点，其预测准确率为 91.8%（AQDT 通过合并某些分枝，将叶结点数降至 8 个所构造的一个等价决策结构），而这时由 C4.5 程序从同样数据中学习获得的决策树包含 8 个结点和 15 个叶结点，其预测准确率为 85.7%[IM93]。

这种方法直接适合 INLEN 体系。一个规则库是由专家提供或通过规则学习运算操作获得的，因此也允许从规则中产生决策结构。

2.9　表示空间的自动改善

2.9.1　确定最相关的属性

在一个大型数据库中，或许有许多属性来描述特定实体。对于一个特定问题，需要确定指定输出属性与其他属性之间关系描述的规则，就希望将独立属性限制为最相关的几个。为实现这一点，就需要利用不同标准来评价一个属性对于一个特定问题的相关程度，诸如增益比率[Qui93]、gini 索引[BFOs84]、PROMISE[Bai82]和 chi 平方分析[Har84]、[Min89]。

这些标准是根据属性的全局性能对其进行评估的，这就意味着具有较高类别区分能力的属性将被评估为最相关的属性。

在确定一个描述性知识表示时，诸如决策规则，目标就有所不同了。这里每个类别均单独描述，且希望获得每个类别的最简单、最准确的规则。因此，若一个属性有一个值，则可以较好地描述某一特定类别，具有这个值的属性就可以有效地用于相应的决策规则中。相反，若一个属性具有全局较低区分能力的取值，那么在构造决策树时就会被忽略。这也就是说：确定决策树的属性与确定决策规则的属性须遵循不同的标准。

为说明这一点，这里以识别英文字母表中大写字母为例。为解决这个问题，需要考虑两个属性以表示相应字母是否有尾部，以及是否均由直线组成。在基于规则（定义）表示时，利用一个简洁性质就可将字母 Q 与其他字母区分开，该性质为：若字母有尾部，则它就是 Q。相反单独直线条件并不能完全区分任一个特定字母，但总体而言还是有效的。

因此，属性**“有尾部”**在学习特定类别时是非常有用的，虽然对于区分

其他类别不是很有效。因而它在规则学习中是较为合适的。然而在决策树学习中，它或许会被评价为总体作用较低的属性而被其他属性所替换。若 Q 字母相对少见时，这种情况极有可能发生。而测试一个字母是否有尾部会被认为是一个多余操作，因为它仅仅对排除为字母 Q 的可能有用，而对其他 25 个字母没有任何益处。同时测试“**均为直线**”条件可以立刻将搜索空间分为两部分，它可以将假设空间划分得更快，但当最后只剩 O 和 Q 两个字母时，再将尾部测试作为最后一步。这样 Q 字母识别就需要过多的测试，但同时也进行了其他字母的识别。

2.9.2 新属性的产生

在最初表示空间与当前问题相关性不大，或要学习的概念较难以属性决策规则形式，如 INLEN 所采用的形式来表示时，就需要产生新属性，它们会是初始属性的函数且更适合所要解决的问题。这可以通过基于 AQ17-DCI 程序的**构造性归纳**运算操作来完成。

由于数据库包含不断随时间而变化的对象信息，所以就需要构造性归纳机制，便于利用时间数据的顺序。例如：数据库或许包含每天一个特定地点的最高温度信息，其中每个记录中有一个字段记载温度记录的日期。通过构造性归纳可以产生许多与时间内在相关的属性，例如某个时期的最高温度，一段时期内的最小人口增长率，种植期间的状况等。

CONVART[Dav81]利用用户提供的和系统建议的默认值来搜索有用的、时间相关的属性，并将其增加到表示空间中。它利用建议列表中的各项来产生新属性，并测试它们与问题的相关程度。若它们超过一个有关的阈值，就将被增加到表示空间中，不断重复这一过程直到构造完所需要的新属性数目为止。作为属性构造能力的一部分，INLEN 将这种技术与产生时间相关属性结合到了一起。

2.10 应用展示：经济与人口统计数据中的发现

2.10.1 背景

经济分析是概念数据分析工具的很重要的一个应用领域。以下这个例子

将展示在从数据中抽取出知识的过程中，智能数据分析系统所扮演的角色。

美国政府保存了从世界各地进出口货物的记录。不同商品和原材料又被分为许多种类。早在 20 世纪 80 年代初期，有关数据显示从日本进口卡车的数量急剧减少，而同时从日本进口的汽车部件相应增加。经过好几年分析者才注意到这一情况并认为日本将底盘和车身分开运进美国，然后在那里再进行拼装，这样就可避免对卡车所征收的高额美国关税，主要针对欧洲，自二战以来这一情况已经登记在册了。在美国分析家推断出这种解释之后，美国和日本就开始了有关卡车进口的贸易谈判。

对于一个分析家而言，关于这种趋势究竟有多快就会被注意到的问题，就是一个可以利用概念数据分析方法以指示在两个有关类别上相反的变化趋势。在他们与日本最终达成新协议之前，还需要花费纳税人多少金钱？注意到上面所提到的经济趋势和模式是一个困难的任务，因为人很容易被淹没在海量数据中。

基于这种需求，经济与人口统计数据分析就成为 INLEN 开发与测试的一个焦点。这里将通过实验展示其发现能力。实验涉及两个类似的数据集：一个是由世界银行提供的，它包含从 1965 年至 1990 年世界上 171 个国家的情况信息；另一个则是从 1993 年的世界博览（由中情局出版）中抽取出的数据，它包含 190 个国家的几个数据库信息（涉及 17 个属性）。

2.10.2　实验 1：多操作的集成

世界银行数据使得我们能够进行大量实验来测试 INLEN 能力。有个实验就是区分东欧和东亚的发展模式，首先找出这种模式，然后产生差异规则[Kau94]。

一种概念聚类运算操作是根据每个国家在 1980 年至 1990 年之间劳动力人口比例变化来确定相应国家的聚类方式的。在这种分类情况下，东欧和东亚的典型国家落到两个独立的集合中。大多数欧洲国家劳动力变化均低于聚类程序为地区而设定的阈值，同时大多数亚洲国家劳动力变化超过所设置的相应阈值。

基于这种聚类，首先采用规则学习运算操作（利用 AQ15c 归纳学习程序）在特征描述模式下，对亚洲一类国家（超过区域阈值）和欧洲一类国家（低于区域阈值）进行特征描述，然后在差异规则优化模式下，对特征描述内容进行压缩使之成为简单的差异描述规则。所获得的差异描述规则如下：

如果满足下列条件，则是亚洲一类国家：

A.1. 劳动力组成的变化≥slight_gain，（9 个国家）

或者

B.1. 人口期望寿命在 60 岁，且，

2. 工作年龄人口≤64%，（2 个国家）

如果满足下列条件，则是欧洲一类国家：

A.1. 劳动力组成的变化接近于 0 或下降，且

2. 人口期望寿命不在 60 岁，（7 个国家）

或者

B.1. 工业劳动力比≥40（1 个国家）

上述规则表示：初始数据集合的 10 个属性中，只有 4 个属性在区分欧洲方式和亚洲方式发展模式时起作用，他们是：劳动力组成的变化、人口期望寿命、工作年龄人口和工业劳动力的比例。在亚洲和欧洲一类情况中，头一条（A）规则描述大部分这一类别的国家；而第二条（B）规则则描述了本类别的其余国家。

这个实验说明集成多个不同学习和发现策略方法的一个基础特性，即这种集成策略容许知识从一个运算操作无缝地输出到另一个运算操作，从而可以获得其他单个程序无法取得的结论。实验也表明该系统所创建的规则易于理解和解释。

2.10.3 实验 2：子群中的异常识别

利用 INLEN 所做的另一个实验就是检测它所创建的子群中有趣规律的问题。人口统计领域中的一个子群可以是具有共同点的成员国家和地区，在所构造子集中的一员与集合中其他成员明显不同时，就产生了一个值得注意的例外。这些例外反过来又会成为进一步发现的跳板。

INLEN 利用描述特征模式下的规则学习运算操作，从世界博览 PEOPLE 数据库中发现几条规则，这些规则描述了具有低人口增长率（每年低于 1%）的 55 个国家的特征。其中一个特征（如图 2.10 所示）有三个条件来共同描述了 19 个低人口增长率的国家，且只有一个国家具有较高的人口增长率。

如图 2.10 所示的描述中，列 Pos 和 Neg 分别表示满足条件的正例和反例样本数。**支持度**（Support Level，简写为 Supp）定义为：Pos/（Pos + Neg），以指

示相应条件在人口增长率低于 1%国家中的支持程度。**共同性**（Commonality Level，简写为 Comm）定义为：Pos/Total_pos，以指示相应条件在人口增长率低于 1%国家中发生的共同程度（在这个例子中，Total_pos = 55）。

具有增长率在 1/1000 人的国家特征描述：		Pos	Neg	Supp	Comm
1	出生率 =10 ~ 20 或者出生率≥50	46	20	69%	84%
2	主流宗教是东正教，或新教，或印度教，或神道教	40	68	37%	73%
3	移民率≤-20	32	104	23%	58%
	全部条件	19	1	95%	35%

图 2.10　具有人口低增长率国家的特征描述

第一个条件（也是支持度最高）表示低于 1%人口增长率国家拥有一个低（低于每千人 20 人）或高（高于每千人 50 人）的出生率。在人口低增长国家出现高出生率是与直觉相反的，对这种描述所涉及的 19 个国家进行仔细检查后发现：其中 18 个出生率低于 20%，仅有一个国家，马拉维，具有高出生率。进一步对马拉维国家进行研究发现，原来马拉维拥有一个巨大外移人口情况，超过 30 人/千人，也是世界上最高的对外迁移比率。INLEN 因此帮助发现了这个与普通模式不一样的异常情况。

2.10.4　实验 3：利用结构属性

上述例子中所介绍的规则包含一个属性“主流宗教（Predominant Religion）”，这个属性在初始数据集中是一个名字属性。为了方便研究属性的结构是如何影响知识发现过程的，就可利用 INLEN 对包含宗教属性和没有宗教属性的相同数据集进行相应属性的结构化[KM96]。图 2.11 给出了一部分属性值域的结构。

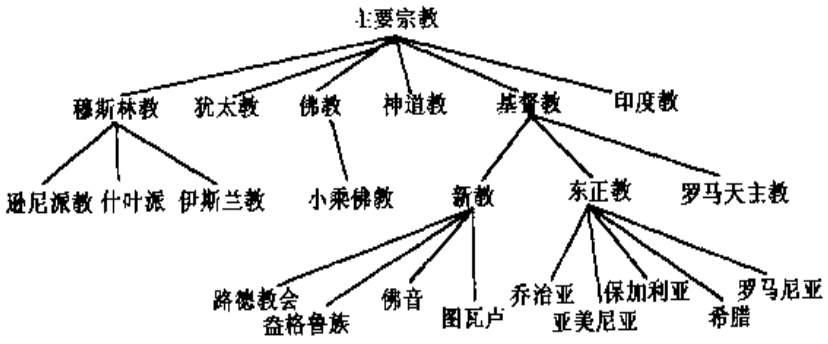


图 2.11　PEOPLE 数据库宗教属性的部分结构

进行结构化的一个重要理由就是：若主流宗教属性是一种非结构（名字）组织的话，那么“主流宗教是路德教（Lutheran）”的陈述会被认为与“主流宗教是基督教（Christian）”是对立的，就像与“主流宗教是佛教（Buddhist）”陈述一样，因为“路德教”、“基督教”和“佛教”之间的差异在一个“平面”值域被认为是相同的。这就导致产生以下陈述的可能，即“主流宗教是路德教但不是基督教”。

利用 INLEN-2 的实验支持有关无结构化和有结构化属性的各种假设。在它们用做独立变量时，可以发现利用结构化属性所产生的规则比没有利用它们的要简单。例如：INLEN 学习获得用低人口增长率来区分 55 个国家的规则，以及在 PEOPLE 数据库中，在属性“主流宗教（Predominant Religion）”没有结构化时，所发现一条规则如下。

如果满足下列条件则人口增长率＜1%：（20 个例子）

1. 文化率＝95%～99%

2. 人口期望寿命为 70 年～80 年

3. 主流宗教是罗马天主教，或东正教，或罗马尼亚，或路德教，或新教会，或日本神道教

4. 外迁率≤20 人/1000 人

55 个低人口增长率国家中有 20 个国家满足这条规则。在利用“宗教（Religion）”作为一个结构化属性进行类似学习实验时，就会发现如下的一个简单规则。

如果满足下列条件则人口增长率＜1%：（21 个例子，1 个例外）

1. 文化率＝95%～99%

2. 人口期望寿命为 70 年～80 年

3. 主流宗教是基督教

4. 外迁率≤10 人/1000 人

这条规则只有一个例外（美国，其 1993 年的人口增长率在 1%到 2%之间）。若需要完全一致，那么第三条规则仍然会比采用没有结构化（“宗教”）属性要简单，其方法就是对基督教（Christian）结点进行最小化的细化操作，这样所获得的规则就会覆盖同样的正例而且没有例外。

利用结构化关联属性也会获得类似的差异。通过将事件分为不同的泛化程度，规则就可以由此对他们进行分类，从而减少复杂性，增加不同泛化程

度规则的信息重要性。

这样的效果对于层次树的较低层次而言是显而易见的。在没有结构化的数据集中，需要五条规则，每个包含两到五个条件来定义描述逊尼派穆斯林（Sunni Muslim）国家。仅有的一个规则覆盖超过两个国家，其条件也是相当琐碎的。

如果满足下列条件，则逊尼派穆斯林是主流宗教：（4 个例子）

1. 文化率≠30%～99%
2. 婴儿死亡率为 25～40 或大于 55/1000 人
3. 生育率为 1～2 或 4～5 或 6～7/1000 人
4. 人口增长率在 1%～3%或大于 4%

这些条件的取值范围被分为多个片断，从而说明这不是一个较强的模式。相反利用一个结构化的（宗教）属性，学习运算操作就会产生两个简单而又易于理解的模式，每个仅包含一个条件。

如果满足下列条件，则逊尼派穆斯林是主流宗教：（10 个例子，1 个例外）

婴儿死亡率≥40 人/1000 人

如果满足下列条件，则逊尼派穆斯林是主流宗教：（4 个例子）

出生率为 30～40 人/1000 人

正如以上所描述的，这些规则仅仅在主流宗教为伊斯兰教的国家情况下才适用，并且建立在已做出决定的基础上。

2.10.5　实验 4：利用构造性归纳运算操作

一个由 Bloedorn 和 Michalski[BM96]所记载的实验展示了利用构造性归纳作为知识发现运算操作的能力。实验数据为 1986 年至 1990 年连续五年 11 个经济属性取值（每个记录共有 55 个属性），学习程序试图发现可以预测国家五年期间经济总产值变化的规则。通过利用三种数据驱动的构造性归纳运算操作，它们分别是：根据现有属性集产生新属性，除去与目标概念关联较小的属性和将数值属性抽象为若干区间间隔，由此对新数据的预测准确率提高了约一半（从 41.7%到 60.5%）。

新近构造相关程度较高的属性为 1986 年至 1988 年间能源消耗变化，1989 年人口出生率与 1990 年能源消耗比率和五年期间平均能源年消耗。

这些结果显示：构造性归纳是数据分析中的一个非常有用的工具，因为

它可以为知识发现创建更充分的表示空间。

2.11 总结

本章的主题就是符号机器学习所研究提出的现代方法，新运算操作在产生概念数据分析中起到了直接而重要的作用，提出了许多有关各种机器学习方法具体应用的思想。

两个非常重要的运算操作就是概念层次树（概念聚类）的构造和归纳，产生描述来指定输出与输入属性之间的关系规则。这些规则代表高层次知识，它们对数据分析而言具有很高价值，并可以直接用于帮助人类进行决策。其他重要运算操作包括：构造方程及其应用的前提逻辑条件，确定时间序列的符号描述，最相关属性的选择，产生新的更相关的属性以及选择代表性样本。

与许多数据挖掘方法相反，这里所提出的方法需要相当多的有关数据和应用领域的背景知识。这些背景知识或许包括，诸如，领域的特征描述和属性类型、属性间关系、因果依赖、产生数据的有关对象和过程的内在原理。所提出方法的一个重要特色就是它们利用这些背景知识的能力。

INLEN 系统中所实现的机器学习技术容许用户可以较容易地完成内容广泛的符号数据操作以及产生知识操作。演示的例子表明了所描述的多策略方法在解决数据挖掘与知识发现问题方面，具有很强的应用潜力。

致谢

这里感谢 Eric Bloedorn，Vinh Duong，Scott Fischthal，Seok Won Lee，Elizabeth Marchut-Michalski，Jim Mitchell 和 Qi Zhang，他们对本章草稿提出了许多有益的意见和建议。本项研究工作是在 George Mason 大学机器学习与推理实验室完成的。实验室的研究工作得到了国家科学基金合同号分别为 DMI-9496192 和 IRI-9020266 的部分支持，海军研究所合同号为 N00014-91-J-1351 的部分支持，以及海军研究所管理的国防先进研究项目局合同号为 N00014-91-J-1854 的部分支持，还得到了国防先进研究项目局合同号为 F49620-92-J-0549 的部分支持，以及空军科学研究所合同号为 F49620-95-1-0462 的支持。

参考文献

[Bai82]　Baim, P.W. The PROMISE Method for Selecting Most Relevant Attributes for Inductive Learning Systems. Report No. UIUCDCS-F-82-898, Department of Computer Science, University of Illinois, Urbana, 1982.

[BMR87]　Bentrup, J.A., Mehler, G.J. and Riedesel, J.D. INDUCE 4: A Program for Incrementally Learning Structural Descriptions From Examples. Reports of the Intelligent Systems Group, ISG 87-2. UIUCDCS-F-87-958, Department of Computer Science, University of Illinois, Urbana, 1987.

[BMMZ92]　Bergadano, F., Matwin, S., Michalski, R.S. and Zhang, J. Learning Two-Tiered Descriptions of Flexible Concepts: The POSEIDON System. Machine Learning, 8, pp. 5-43, 1992.

第3章　机器学习在多个领域的应用

Pat Langley 和 Herbert A. Simon

摘要

机器学习的一个重要应用领域就是帮助专家系统实现其知识库内容的自动获取。这里将要介绍决策树和规则归纳一些最近的应用情况，其中主要介绍如何应用机器学习，以及所研发的专家系统的现状。最后还将介绍在具体领域应用学习系统的若干阶段及一些成功应用的宝贵经验。

3.1　前言

获得专家级性能就需要专业领域的知识。尽管知识工程业已产生成千上万个工业级专家系统，但是专家系统的开发过程仍然是一个费时的、巨大的人力活动。机器学习的一个主要目标就是希望通过提供能够从训练数据中发现规律的技术，实现知识工程过程的自动化，改进知识获取的准确性和获取效率。因此机器学习效能最终测试的依据就是它所产生系统的能力，这些系统可以正常应用到工业、教育和其他领域。

事实上，过去10年里，已经有许多机器学习应用到多个领域知识系统的实例。Widrow 等人（1994年）对利用基于后向传播的人工神经网络和归纳方法的一些传统应用进行了回顾；而 Allen（1994年）则对利用基于示例的表示和学习方法，如最近邻方法在一些领域的应用进行了总结。

本章将重点介绍决策树和规则归纳方法的相关应用。这并不意味这些机器学习方法比其他方法更为重要或更具有鲁棒性，而仅仅是对前面所提到的基于人工神经网络和基于示例方法具体应用情况总结与回顾的补充[1]。我们假

[1] 其他方法，包括：分析学习（如：Samuelson 和 Rayner，1991）和概率学习（如：Manganaris 和 Fisher，1994），业已进行了应用研究，但就我们所知并没有实际投入应用的系统。

设读者已经具有了一定基础知识，能够理解逻辑描述，包括决策树算法（如C4.5，Quinlan，1993年）和规则归纳技术（如AQ，Michalski和Chilausky，1980年）。我们将重点介绍八个利用这类方法并投入实际领域应用的知识库，同时也将简要介绍其他应用工作的一些情况。最后我们将介绍从这些应用项目中所获得的一些经验。

值得一提的是，几乎所有的机器学习应用工作都是着眼于简单的分类或预测任务。这也不足为奇，因为最鲁棒的学习方法也是设计用来解决这类问题的。学习获得简单决策的局限性并无大碍，因为一个复杂过程，诸如设计、控制，或规划，均可以分解为一系列单个步骤，而每一单个步骤仅涉及简单的分类或预测工作。我们将会看到许多工作正是采用这种分解方法。

3.2 规则归纳在多个领域中的应用

为清楚地说明决策树和规则归纳在解决现实问题方面的潜力，本节将要介绍这类方法在多个领域的一些实际应用情况。介绍具体应用的主要内容包括：应用问题，以监督学习表示的实际需要，所生成的知识库的现状并列表突出描述这些主要内容。但是这里无法穷尽各领域的应用，因此最后我们还将简单介绍规则归纳方法最近的一些应用情况。

3.2.1 提高化工过程控制中的产量

原子能发电厂的燃料是通过将铀的氟化物气体转换为铀的氧化物粉末颗粒而获得的。这些小颗粒必须具有很高的质量，但专家无法预测一批小颗粒何时是好的或是不好的。Westinghouse的研究人员利用统计方法来预测小颗粒的质量并取得了有限的成功，但预测属性间的相互作用制约了这种方法的有效性。

Leech（1984年）采用了另一种基于决策树的不同做法。他的主要工作如表3.1所示。他收集在不同生产控制参数情况下（如颗粒参数、粉末特征）生产出来的具有高质量或低质量颗粒的批量数据样本。生产控制参数属性既有数值量也有符号量。然后将这些颗粒数据样本作为训练数据，利用决策树算法进行处理，并将得到的决策树转换为可以预测颗粒质量的规则。Leech不断

重复进行这一学习过程，以便发现可以预测定性粉末质量特征的规则，之后这些规则就会应用到最高层规则之中，并构成结构化知识库的基础。。

表 3.1 Leech（1984 年）在 Westinghouse 燃料处理方面工作的主要内容

问题描述	提高原子能发电厂燃料处理的产量
具体需求	学习预测颗粒质量的规则
数据表示	颗粒化质量和生产控制参数
数据收集	与专家交流，从颗粒生产中获取
分析评估	将规则交给有经验的过程工程师进行评估
应用状况	1984 年起应用，增加了生产量，提高了颗粒产量，降低了库存

经过慎重的分析，Leech 将这些规则提交给有经验的生产过程工程师，这些工程师认为这些规则是可以接受的。从而工厂技术员开始利用这些规则来控制颗粒的生产过程。而当一批新数据到来时，Leech 又对这些新数据重复进行归纳学习以获得更加准确的规则。这一领域的专家系统帮助提高生产量得到更高质量的颗粒产出，并降低了库存，为 Westinghouse 每年增加一千多万美元的销售额。

3.2.2 信用评估决策

贷款公司通常利用问卷形式来获取申请贷款人的有关信息，这些信息将帮助决定是否放贷。这一过程很久以前就已部分自动化了。如 American Express UK 就利用了基于判别式的统计决策过程。当一个人评估值低于特定的阈值时就被拒绝放贷，而当一个人评估值高于特定阈值时就同意放贷。这样大约 10%到 15%的申请人处于“边界”区域，他们将被移交到贷款主管做最后决定。但记录显示贷款主管以默认值确定给予“边界”申请人贷款时只有不超过 50%的准确率。

这些情况促使 American Express UK 尝试利用机器学习方法来改进决策过程。Michie（1989 年）和他的同事对 1 014 个训练样本和 18 个描述性属性（如一个雇员的年龄和工作时间），利用一个归纳方法进行分析处理并获得了一个决策树，这个决策树中包含大约 20 个结点，10 个原始属性，对“边界”申请人的预测准确率在 70%。除了获得较高的预测准确率之外，公司还发现这些规则还有强大的吸引力，因为它们可以帮助向申请人解释决策的理由。尽管这个项目仅仅只是作为一个探索性工作并只花费了开发团队一周的时间，但

American Express UK 还是很重视它，并将所获得的知识投入到实际应用中，但并没有做进一步研发。具体情况如表 3.2 所示。

表 3.2 Michie's（1989 年）在 American Express UK 贷款决策方面工作的主要内容

问题描述	减少采用默认值给边界申请人贷款而造成的损失
具体需求	学习决策树以预测申请人是否可以获得默认值
数据表示	申请人的标准描述属性
数据收集	1 014 个公司记录样本
分析评估	比贷款主管放贷准确率有明显提高
应用状况	经过一周的开发努力的结果被 American Express UK 所采用

3.2.3 机械设备故障诊断

电动泵在化学工业扮演着一个重要的角色。为降低运行中断次数，采取预防性维护措施已是常见的举措。在伊朗石油公司一个化工分部 Enichem，诊断人员定期检查每个泵，测量其不同位置的晃动情况以确定该泵是否需要进行修理。整个装置包含一个电机和一个泵，其轴是用弹性联轴节连接而成的，电机和泵均通过弹性支座被固定在地面上。典型的故障包括：泵失去平衡，轴承有毛病，底部变形。Enichem 的专家利用傅里叶方法分析晃动以帮助进行诊断决策。

Giordana 等人（当前）相信这项工作可以借助机器学习方法。早前他们与一名 Enichem 专家合作研制了一套诊断电动泵的专家系统，采用规则来表示知识，利用传统交互手段来获取知识并将其进行手工编写以构造规则库。在这一过程中，研究人员发现专家测量泵不同位置的晃动情况，然后利用数据分析来帮助进行故障诊断。他们的主要工作内容如表 3.3 所列。

表 3.3 Giordana 等人（当前）在 Enichem 预防性维护方面工作的主要内容

问题描述	防止一家大型化工厂的电动泵停机
具体需求	学习规则以预测将要发生的故障类型
数据表示	傅里叶分析晃动情况，以及专家的因果知识
数据收集	209 个泵测量数据样本，并由专家进行标示
分析评估	学习获得的规则比手工编写的规则更为准确
应用状况	应用于工厂，降低了因不适当停机而导致的空转时间

在收集了 209 个泵测量数据样本之后，他们请专家将这些样本标记上所

对应的不同故障，Giordana 等人利用一个归纳算法来处理这些数据，从而获得一组新的诊断规则。他们的方法利用了从专家那里收集的因果知识，并以此来约束规则归纳过程，从而提高了专家接受这些规则的可能性。实验表明学习获得的知识库比手工编写的规则更为准确，现在归纳出的规则业已替换掉了诊断系统中原来的规则。由于引入这些规则，因不适当停机而导致的空转时间大大减少，此外，在有经验的专家退休后，这些学习获得的规则可以有效帮助接替他们的经验不足的人。

3.2.4 天体对象的自动分类

第二Palomar 天体测量天文观测站已经产生了 3GB 的图像数据，其中包含了将近 20 亿个天体对象。过去，天文学家手工在照相图版上对这些对象进行识别和分类。但是这要在比现在目测检查或现有计算机方法所能支持的微弱得多的光线情况下，处理星体和星云。先前通过编写专家系统完成此项任务的努力并没有获得可靠的进展。

针对这一情况，Fayyad 等人（1995 年）采用机器学习来着手解决这一问题。主要工作内容如表 3.4 所示。首先他们利用图像处理技术所获得的标准数值属性来描述一组图像中的各个对象，这些数值属性包括对象大小、面积、椭圆率，以及统计动差和核心亮度；接着由天文学家对每个对象进行标示（星体或星系）；然后研究人员利用决策树算法对这些训练数据进行处理，以便获得一个可以对新对象进行分类识别的决策树。最初的实验结果并不令人满意，获得的新对象识别准确率较低。但是 Fayyad 等人同天文学家一起利用现有属性构造了其他的预测属性，从而将归纳知识的预测准确率提高到 94%，这一准确率超过了天文学家所设置的必须达到的科学数据分析的准确率。

表 3.4 Fayyad 等人（1995 年）在 Palomar 天体测量方面工作的主要内容

问题描述：对 Palomar 天文台天体测量所获的 20 亿个天体对象进行分类
具体需求：学习决策数规则以便从星系中识别出星体
数据表示：标准数值属性及高阶特征
数据收集：从天体测量所获的数据样本，并由专家进行了标示
分析评估：94%的准确率，超过了天文学家对数据分析所要求的标准
应用状况：植入了一个数据库管理系统，以帮助对天体中对象进行分类

研究人员将所获得的分类器植入了天文学家使用的数据库管理系统，这

种系统可以帮助天文学家完成多种工作，如恒星和星系分布的统计分析。该系统正用于对天体测量所获图像中的所有对象进行自动识别分类，而这项工作对人类而言是不可能的。系统可以识别比目前大规模测量所获类别小一个数量级的对象，从而产生了一个足没有采用机器学习所获种类数三倍的种类集。

3.2.5 监测旋转乳液的质量

在 Sendzimir 工厂里，通常利用由水和油混合乳液进行卷轧冷钢时的冷却和润滑。而钢的质量与这种乳液性质密切相关。由于这个原因，Jesenice 钢厂（位于斯洛文尼亚的 Jesenice）连续监测各种测量值，诸如油的浓度、细菌存在情况，根据这些测量值，工厂员工确定乳液质量及所需要的处理措施，如：增加磁过滤或替换乳液。在复杂情况下，还需要向化学专家讨教，并在其不在现场时，通过对话交流的方式获得其相关技术。

在这种方法不奏效时，开发人员与当地大学研究人员合作，利用归纳方法对专家决策所形成的训练数据进行分析（Karba 和 Drole，1989 年）。他们所获得的决策树就被应用到工厂实际工作中，但稍后由于乳液和供应商的变化，这些知识就不能满意当前的工作了。在手工调节不起作用之后，开发人员又收集了专家决策的一组新数据样本并用同样的归纳方法，又获得一个改进后的决策树。但是所有这些仅仅在与专家合作设计了一组新属性并用于归纳之后，方会取得成功。从 1989 年至今所获得的知识库就一直在工厂使用，如表 3.5 所列。

表 3.5 Karba 和 Drole（1989 年）在 Jesenice 钢厂乳液质量方面工作的主要内容

问题描述	维护一个钢厂的乳液质量
具体需求	学习决策树规则以使预测需要采取何种适当措施
数据表示	从专家那里获得的乳液属性
数据收集	工厂操作记录，包括专家的操作记录
分析评估	未知
应用状况	应用于钢厂，稍后的变化需要重新设计属性

3.2.6 减少照排印刷时的条纹现象

照排印刷就是利用一个合金板和浸满墨水的铜制转印滚筒，通过压印它

们之间的纸带而实现印刷。遗憾的是，在印刷过程中出现条纹，空道现象，进而发生在印刷出来的纸张上。这时印刷就必须停下来，有时还需要更换较为昂贵的圆筒。产生条带的原因基本上不清楚，专家也无法可靠地预测出何时出现条带。

Evans 和 Fisher（1994 年）决定利用一种决策树归纳方法来帮助降低条带出现的次数，而条纹问题已经成为一个大型美国印刷公司，R.R.Donnelley 工厂的头疼问题。他们与工厂技术人员合作，收集出现条带的正例和反例，以及出现这些事例时的环境因素（那些技术人员认为具有潜在影响的因素）。Evans 和 Fisher 利用归纳算法处理这些数据，并构造出一个决策树来帮助预测在各种情况下出现条带的可能性，工作主要内容如表 3.6 所示。

表 3.6 Evans 和 Fisher（1994 年）在 R.R.Donnelley 工厂条带减少方面工作的主要内容

问题描述：	减少照排印刷过程出现条带的情况
具体需求：	学习决策树以预测何时出现条带
数据表示：	专家所了解的环境与控制变量
数据收集：	由印刷技术人员手工收集
分析评估：	试验证明准确预测条带出现的能力
应用状况：	1991 年应用之后条带情况的大幅度减少

研究人员将归纳获得的决策树转换为一组规则集，并将它们张贴在 Donnelley 工厂以供印刷工人使用。现在技术人员利用这些规则对墨水饱和度和其他因素进行控制。采取这些新措施之后，出现条带情况的频率大幅下降。例如：出现条带事故数量从 1990 年的 384 次下降到 1991 年的 135 次，之后一年在印刷工作完全接受这些规则的情况下，下降到只有 66 次。

3.2.7 改善油气分离质量

原油从地下开采出来时含有天然气。在炼油厂处理原油之前，就必须首先将天然气从原油中分离出去。但由于这一过程涉及对多个参数，如分离容器大小、重量、尺寸和组成的配置，因此 British Petroleum 决定利用决策树，并根据油、气和水的相对数量、压力、饱和度和混合物温度等类似参数，来帮助确定其最佳配置。

配置任务的复杂性促使开发人员利用**结构化归纳**（Structured Induction）方法。该方法将某些决策树的输出结果作为最高层次决策树的输入，同时需

要将对每棵进行单独学习而将学习任务进行分解。Guilfoyle（1986 年）提到 British Petroleum 开发人员收集了 1 600 个训练样本，并将学习建立的含有 2 500 条规则的知识库分为 25 个集合，然后公司将这些规则编写为 14 000 行 FORTRAN 代码。到 1987 年，软件已在 4 个不同地方投入正常使用，将原来需要一天人类专家方可完成的任务减少到 10 分钟就可以完成。他们工作的主要内容如表 3.7 所列。

表 3.7　British Petroleum 在配置分离容器方面工作的主要内容（Guilfoyle，1986 年）

问题描述	配置从原油中分离出天然气的容器
具体需求	学习决策树以预测最佳的容器参数
数据表示	多个不同抽象层面的油气混合物特性
数据收集	从专家那里获得的 1 600 个实例
分析评估	准确率未知，但比人类专家更快
应用状况	规则转化为 FORTRAN 代码，1987 年在 4 个地方投入应用

3.2.8　预防电力变压器故障

电力公司通常利用大型满油的变压器来进行电力配送。但是逐渐变质的绝缘体、过热、接点失灵和其他问题常常会导致损失巨大的故障。由于变压器中油的气相色谱可以说明其化学变化，因此专家们可以利用这一点进行准确的预测。为减轻专家的工作负担，一家工业设备保险公司，Hartford Steam Boiler，资助了利用规则归纳解决这一问题的专家系统。Riese（1984 年）对所研制的系统进行了描述。该系统包含 27 组规则，这些规则负责检查数据的有效性，识别存在的特征状况，然后根据这些特征状况推断出故障，最后给出应采取的正确操作。对 859 个测试实例的实验结果表明：所归纳出的规则与人类专家的诊断仅仅在四个实例上不相符合。1990 年，系统投入正常使用，为保险公司的客户自动生成相关报告，主要工作内容如表 3.8 所示。

表 3.8　Riese（1984 年）在 Hartford Steam Boiler 预防诊断方面工作的主要内容

问题描述	确定是否对大型利用油的变压器进行保险
具体需求	学习规则以便根据油中的化学痕迹预测故障
数据表示	气相色谱的化验数据
数据收集	公司记录中的历史数据
分析评估	在对 859 个测试实例中与专家诊断非常接近
应用状况	1990 年投入使用，为保险公司客户生成报告

3.2.9 规则归纳在其他领域的应用

前面所介绍的应用仅仅是决策树和规则归纳在各领域应用中的一小部分，虽然只有很少的应用结果被发表在科学期刊上，如 Donald Michie（1987 年）就介绍了用于电路板故障诊断的四个归纳知识库，这些系统已被正式用于欧洲电子实验工厂，并且每年可以帮助节省数百万美元。Hayes-Michie（1990 年）介绍了一个专家系统，它也是利用决策树归纳而开发的，并被西门子正式采用，并用于配置建筑物中的防火设备。Gill Mowforth（个人通讯，1993 年）提到另一个系统，部分采用了决策树方法，现被南非银行用于信用卡发放评估。David Stirling（个人通讯，1994 年）也利用类似的一种方法来设计规则以预测轧钢厂的状况，目前应用于澳大利亚的 BHP Stainless。

实际上有一些专门从事决策树和规则归纳的软件公司。例如：Attar 软件的 David Isherwood（个人通讯，1994 年）介绍了一个系统，该系统可提供股票交易咨询服务，目前被六个欧洲国家超过 20 个证券经纪商所使用。Leeds Permanent 建筑协会使用了一个可以预测支付过期抵押的系统；一个用于公共付费电话的故障诊断系统，它可以降低工程师造访次数和提高修理速度；一个帮助保险公司预测留住好销售人员可能性的系统；一个由健康保险公司使用的系统，它通过描述不同医疗过程的平均索赔来监测客户和服务商提出的过分索赔请求。

类似的情况，Novacast 的 Rudolph Sillen（个人通讯，1995 年）介绍了一种为管理者在增值税方面提供建议的系统，该系统自 1992 年起已在瑞士的几个地方投入使用；自 1994 年瑞士一家铸造厂就使用了一个用于铁合金处理的热分析系统，通过减少废料每吨可以节省 50 美元并提高了产量，降低了能耗和其他添加材料；自 1993 年一个为金属和其他涂层过程选择颜料的咨询系统就在瑞士投入了商业应用。一个用于评估军队能力的系统每年可以帮助瑞士国防部物资管理部门节约一千万克朗；瑞士 Karlstad 中心医院的医生自 1993 年就使用一个系统，它可以对实施手术后的乳腺癌患者 5 年内是否会出现新肿瘤进行预测。在 Infolink Decision Services 的 Thamir Hassan（个人通讯，1994 年）介绍他的公司已经应用了采用类似方法开发的一些系统。

3.3 规则归纳的其他应用研究

作为上一节各领域应用的补充，本节将介绍一些在应用研究方面的工作

情况。虽然这些研究出的系统目前并没有投入正常应用，但这些系统所解决的问题可以作为决策树和规则归纳方法鲁棒性和灵活性的佐证。

3.3.1 填表工作的自动化

填写表格是一项单调乏味的工作，它占用了商业和政府部门大量的时间。即使将这一过程部分自动化也可以产生巨大的效益，但是为每个表格开发一个单独的专家系统的代价使得这种方法不切实际。Hermens 和 Schlimmer（1994年）开发一个填表辅助系统，它可以根据观察学习用户的偏爱。该系统利用决策树的增量版本来发现相关规则，以便能够根据其他填表项内容预测当前各填表项的默认值。用户通常可以修改预测值并修改学习出的规则。实验表明填表助手可以帮助节省87%的击键工作，其预测准确率高达90%。Hermens 和 Schlimmer 所在大学部门管理人员一直使用这一系统直到8个月后，由于硬件变化才结束了这个项目。

3.3.2 支持知识库维护

最早一批专家系统之一（西屋公司用于电机、发电机、变压器自动设计工作，1956年）经过若干年已停止使用，究其原因，就是对系统重新改写以包含新设计知识的成本太高（Simon，1993年）。随着专家系统技术的成熟，人们愈加明白专家系统生命周期中的一半成本是用于其知识库的维护。对专家系统知识库进行定期维护不仅是因为需要修正其编码错误，而且因为随着时间消逝，问题本身、设备和用户都在发生变化。

例如：Langley 等人（1994年）介绍了一个西门子公司采用的计算机断层扫描器的诊断系统，由于其知识库开始出现问题，Langley 等人考虑利用现有的归纳算法对知识进行修正以解决这种问题，但现有的知识修正方法基本是针对 Horn 子句和决策树表示的知识，而目前这个诊断系统采用了一个层次故障树。尽管如此，研究人员借鉴现有方法中的搜索框架，将学习处理替换成了适合层次故障树的处理。这种方法还没有在实际应用中得到验证，但用实际数据合成进行的初步测试令人满意。

3.3.3 航天飞机引擎的测试

航天飞机的主要引擎在其开始投入正常工作前，需要进行广泛的测试。每种点火测试会产生 100MB 数据，这些数据是由布置在引擎各处的压力、温度、速度、张力和加速度传感器收集的。各工程小组检查这些数据以确定是否已进行了足够的测试，引擎性能是否满足最严格的要求。他们必须决定是否需要进行下一个点火测试，是否需要替换引擎中的部件等。

由于这一评估过程耗资巨大。Rocketdyne 利用结构化归纳方法（与 British Petroleum 所使用的方法类似）来递归构造结构化决策树以解决有关问题。Modesitt（1990 年）介绍了其中一个这样的系统，该系统设计用来处理静态点火测试的数据，共包含 1 500 个规则，被分为 48 个规则集。另一个知识库被构造用来分析动态数据，如频率和摆动，也是采用类似的方式。两个系统均被植入了一个更大的软件系统中以支持测试过程。不同模块的实际测试结果令人鼓舞，并被建议作为整个系统的扩展。

3.3.4 严重暴风雨的预报

虽然数值方法可以帮助提前预测大范围的天气情况，但是当地的天气预报仍然依赖人类气象专家的技能。例如：他们利用低层潮湿情况和高低层不稳定的势场这些因素，来帮助确定严重暴风雨的可能性，同样他们还可利用这些因素来分析露点、水平对流变量和稳定性指标。Zubrick 和 Riese（1985 年）介绍了一个解决这类问题的专家系统，国家严重风暴预报中心的一个气象学家利用决策树归纳方法开发这个系统。该系统的层次结构有助于对预测做出解释。在一周的测试中，发生了五次严重的暴风雨，系统做出了比传统方法更为准确的预报。

3.3.5 直升机叶片的修理

修理直升机叶片通常需要 6 个月时间并需要若干工作小组参加。为减少修理活动的成本，Eurocopter 开发了一个专家系统，以便对其整队直升机修理提出建议。但是系统使用经验表明：新故障和新的修理不断出现导致其知识库需要更新。EI Attar 和 Hamery（1994 年）希望利用规则归纳方法来帮助解

决系统知识更新的问题，而且要使以前修理所形成的数据可以被很方便地用到。原来的专家系统包含 800 条规则，每条规则涉及两到八个属性；而归纳出来的规则只有 300 条，每条规则只涉及两到四个属性，从而更易被专家理解，而且在手工编制专家系统处理错误的实例中，学习获得的规则可以正确预测其中 88%的修理情况。

3.3.6 预测蛋白质结构

分子生物学中一个大的尚待解决的问题就是涉及根据氨基酸序列信息预测蛋白质第二结构（折叠）。虽然提出了一些理论，但它们的预测能力令人失望。Muggleton 等人（1992 年）考虑归纳逻辑编程较为适合这种关系领域而着手利用这种方法来解决这一问题。他们取出仅包含 α 螺旋的 16 种蛋白质并将这些蛋白质中每个位置作为一个训练样本，同时他们还引入了有关每个位置残余和这些残余的物理化学性质的背景知识。通过归纳算法所获得的初始规则是相当准确的，但是在增加了这些规则预测的背景知识之后，再进行一次归纳过程，所获得的规则集产生了更好的结果。另一种重复归纳策略所获得的规则集在单独四个测试数据集上取得了 81%的预测准率，大大高于有关文献所给出的结果。

3.3.7 钢厂调度自动化

钢厂中的物料调度是一项复杂的工作。专家将其分为三个主要部分：将来料放入仓库；从仓库将物料送到工厂进行粉碎、混合，或冲压；将铁矿通过筛选或粉碎后送到工厂。例如：根据一批物料中矿石块大小，可以进行粉碎，与其他物料混合，或将它们直接送到鼓风炉。Michie（1992 年）介绍了南朝鲜 Pohang 钢铁公司致力研发的一个专家系统，该系统利用结构化决策树归纳方法来解决这一问题。开发人员首先与专家交流以确定与每个过程潜在相关的属性。由此开发的调度系统包含 40 个规则集，在操作测试中其工作性能可以与领域专家相媲美。

3.3.8 更多应用及其相关方法

毋庸置疑，以上介绍的各种应用仅仅是决策树和规则归纳方法应用中的

一部分，正如本书其他章节所述。例如，有关人类医疗诊断的机器归纳应用的研究文献非常多（如 Kononenko 等人，1984 年；Quinlan 等人，1987 年），许多研究在线数据的实验也都可以归到应用范畴。读者应将本章所介绍的系统作为它们的代表而不是全部。

这里虽集中对机器学习范畴的技术做了介绍，但统计学的独立发展也产生了类似的方法。Breiman 等人（1984 年）介绍了一组归纳决策树的方法，他们对这组方法进行了广泛的应用测试，如预测最近发作过心脏病人的生存机会。有一些相关的统计方面工作，如被称为自动交互检测（Automated Interaction Detection）方法（Biggs 等人，1991 年）已被应用于调查数据分析。类似技术现在已被包括在了一个应用广泛的 SPSS 软件中，从而使得规则归纳能够被更多的人使用。

3.4 若干策略和经验

机器学习的应用研究遵循一个标准模式，但却很少有人将其在文献中明确地描述出来。本节我们将对这一过程（正如前面各表格所描述的）各主要阶段的特征做一描述，同时还将介绍早前介绍例子所涉及的一些经验。最后，我们对机器学习成功应用中影响效能的因素做一些初步的总结。

这里我们需要强调：虽然各阶段通常按照所给顺序进行，但整个过程实际上是迭代的。也就是说开发人员常常经历一些阶段，发现一些问题，然后回过来转到较早的阶段重新分析已得出的结论。开发一个领域应用或许需要反复进行若干这样的轮回，根据后面的反馈对前面的选择进行修改。

3.4.1 问题的明确描述

利用机器学习解决任何实际问题的第一步就是对问题进行重新定义描述，使之适合由某种归纳方法来解决。过程控制、诊断和调度均是复杂任务，但可以抽象出简单的分类问题，从而可以利用现有鲁棒的归纳学习算法来解决。我们已不断看到了开发人员将一个明显复杂的问题转换为一个简单的分类问题。在我们介绍过的应用中，只有 Langley 等人的工作采用了可以直接处理复杂问题的学习方法。但是这个项目，与其他简单方法相反，并没有产生一个应用领域知识库。

一些开发人员利用结构化归纳（Structured Induction）技术，该技术将一个复杂任务分解为若干子任务，然后为每个任务提供单独的训练数据样本。Zubrick 和 Reese（1985 年），Leech 和 Modesitt（1990 年）均采用这种方法，所获得的运行系统可以完成多步推理，但因将其分解从而使得归纳过程得以简化。Muggleton 等人（1992 年）采用的方法就是：将学到规则所产生的预测作为背景知识，以供下一轮归纳使用，以此作为学习任务分解的另一种解决方法。

对机器学习研究人员来说，实际问题最好的描述可能不是最直观的。在过程控制领域中，搜索规则或树以便直接预测过程变量值似乎是一件自然而然的事，如根据诸如湿度等环境参数预测印刷中墨水的饱和度。然而我们的介绍中有两个控制任务（Leech，1986 年；Evans 和 Fisher，1994 年），开发人员利用归纳方法来发现可以直接预测过程与环境变量的效果，显然这是因为用户更熟悉这种描述。而在另一方面，Michie（1992 年）所介绍的工作则采用了更“自然”的方法，因此无法给出更一般的结论。

3.4.2 确定表示方法

机器学习应用的第二步就是对于训练数据和要学习获得的知识，确定应采用的有效表示方法。这里我们涉及到表示的形式，诸如决策树或人工神经网络，也涉及描述样本和学习结果的属性或特性。

表示工程（发现某种现象的有效表示）是我们介绍的大多数项目的中心任务。在某些情况下，仅仅需要与领域专家进行交流以获得他们对具有预测价值的属性的意见和建议。而在另外一些情况下（如 Fayyad 等人，1995 年），就涉及对特征空间进行费力的搜索以寻找合适描述符，它具有大多数特征不具备的区分能力。

在一些情况下，“基础”特征可以通过现有方法计算获得。Fayyad 等人基本所依靠的现有图像处理技术（将数字图像转换为属性-值）的描述形式，能够由决策树来处理。Zubrick 和 Reese（1985 年）将传统的统计方法应用到他们的暴风雨预报工作中，Giordana 等人（目前）以傅里叶分析的输出来作为基础属性的取值。

3.4.3 训练数据的收集

在确定任务和表示方法之后，就需要为归纳过程收集所需的训练数据。

在一些应用领域，这一过程较为直接甚至可以自动化，但在另一些应用领域，数据收集就是一件富有挑战的工作。Evans 和 Fisher（1994 年）在照排印刷消除条带方面的工作中，研究人员要求印刷技术人员定期记录下印刷过程的检测参量和结果值，但是这些技术人员不太乐意将时间花费在收集工作正常机器的数据上，因此必须花费许多精力来劝说他们记录下机器工作正常或失灵时的有关数据。大多数领域应用处于这两个极端之间，需要借助专家帮助以标示训练数据或产生这些训练数据。

数据的获取，很大程度上依赖于被研究系统所使用的仪器。理想情况下，专家系统可以直接从运行系统使用仪器接入数据流。在专家系统变得越来越普及时，它们相应的测量仪器正逐步设计成可嵌入到它们所指导的系统中。然而在可预见的将来，在缺乏数据的地方，如何获取数据流和产生相应的数据将是机器学习应用工作中的一个重要内容。

3.4.4 评估学习获得的知识

从训练数据学习归纳获得的规则可能质量不高。必须对这种方式获得的知识性能进行有效评价以便这些知识能够投入正常应用。一种常用的评估方法就是将数据集分为两部分，第一部分用于训练，另一部分用于测试学习所获得的知识。也可以通过进行不同的分割而不断重复这一过程，然后计算测试结果的平均值以评价规则集在新问题上的性能。Kibler 和 Langley（1988 年）介绍了采用这种训练测试方法对众多学习算法的实验情况。

然而在许多应用领域都有领域专家，忽视他们的意见将是非常愚蠢的，即使他们无法充分将自己的知识表达清楚。因此评估过程的一个重要内容就是让专家检查学习获得的知识。若这一阶段出现重大问题，他们或许会建议对问题描述或表示方法进行修改。Evans 和 Fisher（1994 年）赞成在领域应用开发中采用这种循环方式，我们介绍的其他工作也采用了类似的方法。

3.4.5 知识库的具体应用

应用的最后一个阶段就是应用知识库的内容。这里我们倾向于从最广泛的语义上来描述这一情况。在某些情况下，所获得的知识无需植入一个计算机系统就可以进行应用。在 Evans 和 Fisher（1994 年）的工作中，写在纸上的

一组简单规则集就足以帮助人们做出正确决策以减少条带的出现。而其他情况，如 Fayyad 等人（1995 年）和 Modesitt（1990 年）的领域应用中，用户不仅要求学习获取知识的计算机实现，而且还希望得到许多与机器学习无关的软件支持。

应该考虑的是学习获得的知识是要投入实际应用的。图形界面在一些场合可以提高它们应用的可能，而在另外一些场合却会妨碍它们应用的可能。有些用户欢迎解释能力，而有些用户就不需要。一些情况下（Giordana 等人，当前），一个手工编制的领域专家系统会促进学习获得的知识的应用。已经相信基于知识的系统是有益的，一些用户不太可能反对完善现有的知识库，尽管机器学习所获得的知识可能对他们几乎没有意义。由于这个原因，与同时引入的专家系统和机器学习模块相比，将机器学习系统作为已投入运行的专家系统的一个扩展会比较容易。

我们对在设计学习系统并确保其实际应用安全方面的用户与专家角色问题上做了大量说明。这里强调鼓励用户和领域专家积极参与设计和应用过程的重要性，需要引入方便好用的计算机界面，以便他们为在工业及其他现实领域设计和应用好机器学习出一把力。

3.4.6 机器学习应用的效能来源

本章我们介绍了决策树和规则归纳的一些应用情况，有些已投入正常实际的使用而其他一些正向此目标努力。大多数应用均采用了充分了解且现成的归纳算法，对有监督的属性-值数据进行处理，而并没有利用研究文献所描述的更复杂的技术。开发人员无需为此感到羞愧，在实际应用时采用在其他应用中或实验测试中已被证实的功能以及可靠性、可拓展性的方法是极其合适的。

事实上，只要对这些项目进行分析就会发现：应用效能的因素并不是特定的归纳方法，而是对问题的清晰描述和方便归纳的表示方法。在这些情况中，机器学习并没有使知识工程完成自动化，但它的确将知识工程分解为两个简单任务：对问题特征进行描述以及设计一个好的表示方法。开发人员无需贬低这样的事实：尽管没有完全自动化，但减少了开发基于知识系统的时间和人力，并能够构造出具有实用价值的系统，正如前面我们所看到的那样。

虽然我们主要集中在规则归纳方法，读者或许会问：针对我们所给出的

效能因素，如果利用人工神经网络、遗传算法或基于示例学习技术来取代规则归纳算法，那么是否会获得相同的效能。最近文献中有关对比研究的结果表明：在涉及多个领域时，它们具有大致相同的效能。因此给定效能相同的工具，每个人肯定会选择自己熟悉且使用方便的工具。

这种情况决非偶然，即差别较大的过程在具体应用中会产生相似的结果。实际上，在将不同管理科学工具应用于调度问题时，就已经出现类似的现象了。出现这种情况可能是因为问题空间的内在特性。如果容易发现全局最优或局部最优与全局最优接近时，许多方法都可能产生相近的效能。已习惯处理复杂情况的工程师们很久以来就已经意识到这种事实了。人们可以在河面上架设吊桥、桁构桥、悬臂桥和其他设计相当不同的桥梁，而且常常并没有确定的理由来说明某个方法优于另一个。

机器学习或许无法完全替代知识工程而成为构造基于知识系统的框架，但是我们所介绍的例子表明：在通向自动化的道路上机器学习已取得了巨大的进展，而且我们期望规则归纳和其他学习方法将会随着其效能不断被人们所认识而日益普及。

致谢

这里感谢 Peter Clark，Donald Michie 和 Steve Muggleton，他们为我们提供了许多应用方面的研究情况。Donald Michie 在促成其中的许多项目并在为开发人员提供咨询建议方面做出了许多贡献。Ivan Bratko，Donald Michie，还有另外两位评阅人提出了许多建议和意见，帮助完善了本章的初稿（发表在 Communication ACM 杂志上）。这项工作得到了学习与技能研究院海军研究所 N00014-94-1-0505 合同和 N00014-94-1-0746 合同的支持，以及斯坦福大学空军科学研究所 F49620-94-1-0118 合同的部分支持。

参考文献

Allen, B. P. (1994). Case-based reasoning: Business applications. Communications of the ACM, 37,40-42.

Biggs, D., de Ville, B., & Suen, E. (1991). A method of choosing multiway

partitions for classification and decision trees. Journal of Applied Statistics, 18,49-62.

Bratko, I., & Muggleton, S. (1995). Applications of inductive logic programming. Communi-cations of the ACM, 38, November.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regressiontrees. Belmont: Wadsworth.

El Attar, M., & Hamery, X. (1994). Industrial expert system acquired by machine learning. Applied Artificial Intelligence, 8, 497-542.

Evans, B., & Fisher, D. (1994). Overcoming process delays with decision-tree induction. IEEE Expert, 9, 60-66.

Fayyad, U. M., Smyth, P., Weir, N., & Djorgovski, S. (1995). Automated analysis and explo-ration of image databases: Results, progress, and challenges. Journal of Intelligent Information Systems, 4, 1-19.

第4章　归纳逻辑编程的应用

Ivan Bratko,Stephen Muggleton 和 Aram Karalič

摘要

本章将介绍一种机器学习中基于逻辑的方法，称为归纳逻辑编程（ILP），以及相关的一些应用。这些应用借助了 ILP 可在学习过程中充分利用背景知识的灵活性。这里对利用 ILP 的不同应用实验结果以及对 ILP 优于其他机器学习的特点进行了介绍。实验表明 ILP 是一种强有力的“知识密集”数据挖掘工具。

4.1　前言

归纳逻辑编程（ILP）将归纳机器学习和逻辑编程结合在一起。在 ILP 中，基于逻辑的表示方法被用于从样本中归纳出规则。ILP 得力于由逻辑和逻辑编程所提供的坚固的理论框架。

最近已经开发了大量一阶逻辑水平的程序。它们包括 FOIL[Qui90]，Golem[MF90]和 Prolog[SMKS94]，Shapiro 的程序 MIS 是最早的先驱程序之一[Sha83]。这些研发工作促使机器学习新领域的产生，它被称为**归纳逻辑编程**（Inductive Logic Programming，[Mug91]）。最近的一些研发工作可参见[Mug92，LD94，MDR94]。

ILP 中的学习问题一般描述如下：给定背景知识 B，表示为一组谓词定义，正例样本为 E^+，反例样本为 E^-，一个 ILP 系统将会构造一个谓词公式 H，它可表示为：

（1）E^+中所有的样本可以从 $B \wedge H$ 中逻辑导出，且

（2）E^-中没有一个样本可以从 $B \wedge H$ 中逻辑导出。

这一定义与一般问题的归纳学习类似，但它坚持采用 $B \wedge H$ 的逻辑表示。

通常在 ILP 系统中，B，H，E^+和 E^-均为 Prolog 程序，$B \wedge H$ 仅仅将 B 的 Prolog 程序与 H 的 Prolog 程序合并在一起。由此 ILP 问题可以视为 Prolog 程序中的练习。现有一个 Prolog 程序 B，该程序不完全，需要进行扩展以便能够正确地处理给定样本集 E^+和 E^-。这些样本可以看成是 Prolog 查询。当提交一个 E^+中的查询时，程序 B 会回答一个“no”。增加一组 H 子句对程序 B 进行扩展，以便使新程序 $B \wedge H$ 对 E^+中查询可以回答“yes”，并对 E^-中查询回答“no”。这样一种普通的 ILP 练习就是对从样本中归纳出快速排序的程序，如将列表{f,e,g}排序为{e,f,g}。合适的背景知识包括列表合并操作的谓词的定义，它根据某些值，将列表分解为“小”列表和“大”元素。利用这些背景知识和一些正例、反例，一个典型的 ILP 系统可在短短几秒钟的 CPU 时间内归纳出著名的 Prolog 快速排序程序。图 4.1 描述了样本和（归纳学习所需不同列表的）背景知识，以及利用 Grobelnik 的 ILP 系统 Markus 的情况[Gro92]。

```
% 例子

example( qsort( [], [a], [a] ), true ).
example( qsort( [a], [a], [a] ), false ).
example( qsort( [d,f,b,e,c,g,a], [a,b,c,d,e,f,g], [] ), true ).
example( qsort( [f,e,g], [e,f,g], [] ), true ).
example( qsort( [b,c,a], [a,b,c,d,e,f,g], [d,e,f,g] ), true ).

% 背景知识

partition( X, [], [], []).

partition( X, [Y | Rest], [Y | Smalls], Bigs)  :-
  gt( X, Y), !,
  partition( X, Rest, Smalls, Bigs).

partition( X, [Y | Rest], Smalls, [Y | Bigs])  :-
  partition( X, Rest, Smalls, Bigs).

% 归纳定义

qsort( [], L, L).
qsort( [X | L], SL1A, SL2B)  :-
  partition( L, X, L1, L2),
  qsort( L2, SL2A, SL2B),
  qsort( L1, SL1A, [X | SL2A]).
```

图 4.1　利用 ILP 系统 Markus 和从不同列表中学习快速排序

全部采用 Prolog 具有的优势是将技术和理论与逻辑编程中方法结合在一起。此外采用逻辑描述容许高度通用地表示问题的各个部分。这种高度通用性反映在了 ILP 应用的广泛性方面。

4.2 ILP 方法与其他机器学习方法的比较

这里我们将介绍与其他更普遍的机器学习方法相比，ILP 所具有的优点和不足。

大多数机器学习应用依赖基于属性的学习，著名程序 CART[BFOS84]和 C4.5[Qui93]中决策树归纳方法就是其中的代表。从更广意义上讲，基于属性学习也包括人工神经网络和最近邻技术方法。基于属性学习的优点就是相对简单、有效，拥有处理噪声数据的有效方法，然而基于属性学习局限于对象间的非关系描述，也就是学习获得的描述不能详细说明对象各部分之间的关系。因此基于属性学习存在以下局限：

- 背景知识只能以相当有限的形式加以表示；
- 缺乏关系使得概念描述语言不适合某些应用领域。

这样的一些应用领域将在本章后面有所介绍。

ILP 优于基于属性学习的主要特点之一就是：ILP 表达背景知识方面的通用性。这种通用性使得用户可以用一种更为自然的方式来描述与学习任务相关的特定领域背景知识。而利用背景知识使得用户能够提出一个合适的问题描述，并能够将与问题相关的约束引入学习过程。相反，基于属性学习通常只能接受以有限表示形式所描述的背景知识。这就意味着在基于属性学习中，归纳几乎都是从头开始的。另一个方面，在 ILP 中可以充分利用在归纳前所了解的知识。因此在 ILP 中，如果问题是要学习化合物的性质，那么就可以以引入分子结构中的原子与它们间的连接作为背景知识。如果任务是要自动构造所观察行为的一个物理系统模型，就可以将与建模有关的数学方法作为背景知识引入。应用 ILP 还涉及开发一个好的、与相关背景知识配套的样本数据描述，然后应用一个通用的 ILP 系统。

当然 ILP 这些优点的代价就是 ILP 具有更复杂的逻辑与计算要求。

下面我们将介绍 ILP 的应用。这些精选的应用主要强调 ILP 谓词逻辑的描述以及 ILP 中背景知识的有效应用。

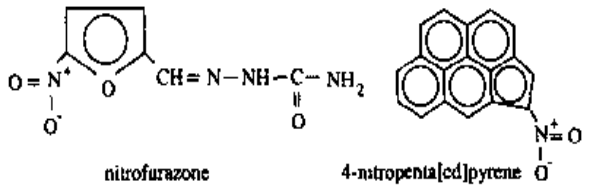
4.3　预测化合物的诱变性

从现实世界数据中构造新的科学知识仍然是机器学习中一个活跃的研究领域。其中一个问题就是化合物中结构/活动关系（简称 SAR）的获得。这就构成了理性药物设计的基础。SAR 中应用较广的一种方法起源于 Hansch[HMFM62]，它利用了回归/差分方法从分子性质，如：疏水性、δ 效果、摩尔反射和 LUMO（最低空闲分子轨道的能量）中来预测其活动性。这种方法和其他许多传统一样都有一个限制，即分子连接和结构描述问题。它们考虑了一个分子全局属性，但并没有充分考虑分子中的**结构关系**（Structural Relationship）。因此有一些可能是重要的信息，诸如分子结构中的模式，可能就无法利用。

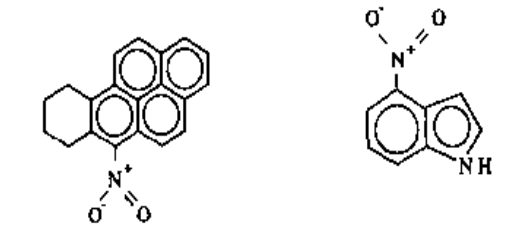
然而，ILP 方法容许充分考虑整个结构信息。一个 ILP 系统 Prolog 程序业已用于从一系列芳香杂环化合物中识别艾姆斯氏试验的诱变性[Mug94，SMKS94]。Hansch 和他的同事们利用传统的回归方法研究了 230 种化合物[DLdCD+91]。从其中的 188 种化合物中，他们利用疏水性、LUMO 和两个手工二值属性，成功地获得了一个描述某些结构特性的线性回归函数。这个回归公式所预测诱变性的准确性是可以接受的。然而剩下的 42 种化合物，就无法利用回归成功建模，也无法提出结构原理。因此这 42 种化合物的子集就被称为“回归不友好”。通过将化合物分为高诱变性和其他由 Hansch 与他同事所建议的部分，可利用 Prolog 处理这些诱变性数据。所有化合物均可采用原子、连接和所带电荷加以合理描述。利用建模程序 QUANTA™可自动产生这些信息，并将 230 个化合物表示为大约 18300 条 Prolog 事实(Horn 子句单位)。利用 LUMO 附加 Hansch 属性与疏水性，其中发现 188 个化合物符合回归规律。所有这些提供给 Prolog 作为学习的背景知识。对于这些化合物，Prolog 构造以下理论。若一个化合物满足：（1）其 LUMO 值小于等于-1.937；或（2）其 LUMO 值小于等于-1.570 且一个碳原子与六个芳香环在一起；或(3)其 LUMO 值小于等于-1.176 且苯环之间存在芳香基-芳香基结合；或（4）一个脂肪质碳所带电荷小于-0.022，则该化合物就是高诱导有机体突变的物质。该理论具有 89%预测准确率。这与 Hansch 与他同事的回归分析，以及最近利用人工神经[VCJ93]所获得的准确率相符合。然而值得注意的是，Prolog 理论更容易理解且可自动产生，而无需存取任何由该问题特定专家手工给定的结构标记变

量。

但是，ILP 优势在剩下的 42 个“回归不友好”化合物子集上就变得更为明显。对于这些化合物，Prolog 抽取出单一一条规则，它在 n 交叉验证测试中具有 88%的准确率（如图 4.2 所示）。这对于 $P<0.001$ 时是显著的。相反 Hansch 与他同事所使用属性的线性回归和线性差分产生的理论只有 69%，而 62%并不比默认的 69%有明显提高。或许比准确率更为重要的是，Prolog 规则提供了新的化合物见解，即存在一个与五个芳香环在一起的氮原子，由单链和双链连接着，以指示诱变性。Prolog 因此识别一个新的结构特征，作为诱变性的预警。

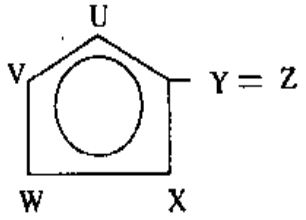


高诱导有机体突变



低诱导有机体突变

(A)



(B)

图 4.2　由 Prolog 发现的“回归不友好”化合物和结构特征。（A）某些化合物被发现不符合回归或差分统计分析方法得到的结果。在这些化合物中，从没有提出过有关这些化合物诱变性的结构规则/预警指示。（B）Prolog 能够识别出双链通过一个碳原子与一个五元原子环共轭的警告。原子 U-Z 不一定必须是碳原子。这是由 Prolog 使用的假设语言中，对 42 个化合物诱变性的最有说服力的解释。（A）中所示两个高诱变性化合物所存在的预警，在低诱变性化合物中不存在这样的预警

4.4　放电机器中的技能重建

在**电子放电机器**（Electrical Discharge Machine，简称 EDM）中，工件表面通过在两个**电**极（工具与工件）缝隙之间**放**电来进行加工。缝隙不断受到第三种元素（除电解质）冲洗[JFv93]。这一过程包括无数随机激活单独放电以产生有质感的表面。过程的稳定性和质量依赖于各种参数：除电解质类型、工件材料、表面粗糙度、电极间（也就是工具与工件间）缝隙大小、放电时间、放电周期、电流上限和电压。

一些参数可以在过程中（如缝隙、流量）加以控制，有一些则需要过程中断（放电持续时间……），还有一些与特别的加工任务相关并无法加以改变（如所需工件的粗糙度）。对于一组标准工件，存在预先设定（由一个 EDM 制造商所推荐的）的一组过程参数，以确保获得一定程度的表面加工质量。然而参数设置是非常保守的，以较好地完成任务所用时间角度看，并没有好的表现。因此通常需要人工来控制过程参数以使得加工时间最小。

很快就会看到自动再现人类操作员行为（一台由“自动操作员”辅助的 EDM 机器）的重要性。因此在这一领域，归纳的目标不是为一个动态系统建模，而是根据实例重建一个操作员的技能。

4.4.1　表示方法的设计

在与操作员交流过程中，发现他们监视以下情况：（1）放电过程所产生不同类型脉冲的不同份额（有三种脉冲：A（空）脉冲、B（有效）脉冲、C（弧）脉冲）；（2）平均电流；（3）缝隙；（4）流量。操作员仅控制缝隙和流量，因此它们被选为控制参数-输入变量。其他数值在整个实验中保持不变。最后 5 秒和 20 秒观测值的平均值和偏差作为属性。一个属性的名称包括三个字母。第一个字母表示参数（A，B，C 表示三种脉冲，I 表示电流）。第二个字母表示时间间隔：L 表示描述最后 20 秒过程状态的属性（长期）；S 表示（短期）描述最后 5 秒过程状态的属性。第三个字母就是 M 表示平均值，或 D 表示标准偏差。

有趣的是，在属性辨识过程中，操作员经常监视某些他们宣称很少监视，或根本不监视的数值。这种现象是在一轮测试中观察操作员行为时被发现的。

为能够检测属性变化速率，定义谓词“<”作为背景知识。属性具有若干合适类型，以使在描述同样数值时可以比较它们的取值（通过背景知识）。

因为我们试图利用两个属性（缝隙和流量）进行控制建模，学习任务就分解为两个学习任务——学习缝隙控制和流量控制。

对于每个控制变量，有三种可能的操作：增加控制变量（将一个数值增加 1.0 操作），没有操作（增加值为 0.0）和减少控制变量（减少 1.0 操作）。

对于每个操作（改变缝隙或流量），在每个**域值**中产生一个学习实例，例如：若增加缝隙，就要产生一个针对“缝隙”（子域）实例，其中操作增加 1.0；同时产生一个学习实例，描述“流量”子域，其中操作增加 0.0；此外还要提供相对稳定过程行为的学习实例；每过 60 秒（其中没有操作）就产生一个实例，其中两个子问题均为增加 0.0 操作。

4.4.2 学习结果和专家评估

利用 FORS 程序（一阶回归）[Kar95]产生若干操作员技能的模型。FORS 是一种具有数值处理能力的 ILP 系统。这些能力包括：利用数值回归、算术以及标准数值函数表示背景知识。在不同修剪程度的基础上归纳获得技能模型。以下就是一个归纳所获技能模型的一部分内容：

```
f(DGap,ASM,ASD,BSM,BSD，CSM,CSD,ISM,ISD,ALM,ALD,BLM,
        BLD,CLM,CLD,IL-M,ILD):-ILD>=0.55,DGap 是 0.3,!.
f(DGap,...):-CLM>=5.6300,DGap 是 0.7,!.
f(DGap,...):-BSM=<1.1000,DGap 是 -0.2,!.
f(DGap,...):-DGap 是-0.5,!.

f(DFlow,...):-ASD=<0.0100,DFlow 是 0.0,!.
f(DFlow,...):-ALD=<0.0800,DFlow 是 0.0,!.
f(DFlow,...):-ASM<ALM,ISD<ILD,BSD<BLD,DFlow 是 0.1,!.
f(DFlow,...):-BLM<BSM,DFlow 是 0.5,!.
f(DFlow,...):-Dflow 是 0.1,!.
```

DGap 意思为**缝隙变化**，DFlow 意思为**流量变化**。注意归纳出的模型或许在-1.0 到 1.0 之间取 DGap 和 DFlow 值，尽管学习实例中这些值仅为-1.0，0.0 和 1.0。在自动控制器中，规则可以按概率方式来确定缝隙和流量相应变

化趋势的比例。操作员和专家[Kom95]对模型的评论就是大多数归纳出的操作均是符合逻辑的，然而也有一些有关模型理解方面的评论：

- 根据操作员观点，问题分解（缝隙与流量的独立控制规则）降低了模型可理解性。将缝隙与流量操作结合起来将会更加直观。
- 在子句层次结构中出现较晚的子句倾向于更难理解，因为必须记住之前出现的所有子句的否定形式。
- 某些情况下，对子句内容（结合前一个评论）进行简单处理后将会改善它们的可理解性。例如：

f(...) :- ALM=<0.10,...,ALM=<0.05

显然第一个文字是冗余的，事实上，领域专家通常会在评价之前重写规则。

为改善模型可理解性，将两个归纳出的子模型结合到一起，以揭示同时控制缝隙与流量的集成策略，模型的结合按以下方式进行：

（1）对模型中每个子句进行扩展，以便每个子句能明确包含前面子句条件部分中的否定。这一步确保每个模型包含互不相交且与评估顺序无关的子句。

（2）缝隙模型中每个子句与流量模型中每个子句结合，以产生与每个子句条件组合相对应的新规则集。

上述过程产生包含 20 个不相交子句的模型，每个预测缝隙与流量操作。为了说明，我们给出从缝隙子模型的最后一个子句和流量子模型的最后一个子句所构造的子句：

```
f(DG/DF,...):-
    ILD<0.55,CLM<5.63,BSM>1.10,ASD>0.01,
    ALD>0.08,ASM>=ALM,ISD>=ILD,BSD>=BLD,BLM>BSM,
    DG is -0.5,DF is 0.1.!.
```

尽管事实上新模型包含的子句多于最初模型两倍以上，而且所有子句更长，操作员和专家声称该模型比前者更易理解。

领域专家定性定义过程和控制参数（缝隙与流量）之间的有效性关系。过程的定性状态按照图 4.3 左边两个维度进行描述。图定义了过程状态：稳定状态和火花状态。过程有效性随两个区域之间边界距离而减少。火花状态出于会伤害工件而需要尽量避免，然而也不推荐停留在稳定过程区域，因为其

特征是低效的。因此控制目标就是指导过程尽可能接近火花与稳定区域之间的边界。更准确地说，即使在边界上，性能也是不稳定的。边界上有一个（小）区域，代表最好的过程性能区域，该区域用星号标记。状态图进一步分为六个区域，定性代表不同的过程行为。

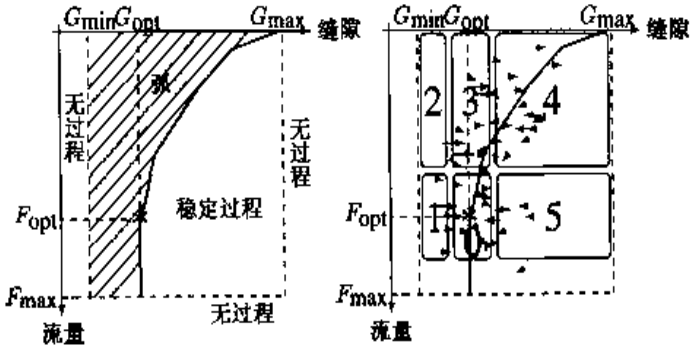


图 4.3　EDM 过程的陈述空间和由 FORS 归纳出的模型预测的控制行为

归纳出的模型通过在状态图上将推荐操作标出来进行评估。对于模型中每条规则，至少绘制一个向量代表推荐的缝隙与流量操作。对于某些规则，可能会绘制多于一个的向量，因为一般规则可能会覆盖状态图中多于一个的区域。图 4.3 的右边描述了由归纳出模型所推荐的操作。很快就会看到大多数向量指向区域（最优工作区域）之间的边界。此外我们看到其中许多指向最优性能的点。看起来似乎模型正确地再现了操作员的技能。模型已被安装在实际机器上并成功替代了人类操作员。

4.5　ILP 的一些其他应用

有限元网格设计

有限元（FE）方法被工程师和建模科学家广泛采用，以解决物理结构的受力分析。有限元方法需要被建模的结构分为有限数目的元素，从而产生有限元素所构成的网格。为了设计一个物理结构的数值模型，有必要确定网格的适当分辨率。需要相当的技能方可确定这些分辨率的值。网格太精致了会导致模型执行时产生过多不必要的计算量，而太粗糙了则会产生无法容忍的逼近误差。

通常对象的某些区域需要较密的网格，而其他区域则只要较粗的网格就可以满足实现较好的逼近。没有一个已知的通用方法可以自动获得最优，或

较为合理的网格。然而，对于有限元计算实际应用已经积累了许多特定对象的成功网格实例。这些网格可以用来作为学习获得构造好网格的实例来源。

由于关联依赖，对象的一个区域网格密度依赖于邻近的区域。ILP方法最自然地应用到网格设计问题中。Dolšak[Dol96]就应用各种 ILP 系统来学习获得好的有限元网格规则（参见[DJB94]和[DJB96]）。

河流质量的生物分类

通过观察各种生物种类生存情况，对河流质量进行监视与评价。尤其是，河床大型无脊椎动物被认为是河水质量合适的指示。不同物种对污染物有着不同的敏感性，因此河流中大型无脊椎动物群体结构就是与污染类型与程度密切相关。Dšeroski 等人[DDHRW94]利用 ILP 来分析大型无脊椎动物样本与水质之间的关系。为了学习，他们利用 292 个从英国 Midlands 河中提取的深海群体样本，由一位河流生态专家将它们分为五种水质的类别。他们构造了这些样本的一个关系表并利用 ILP 系统 Golem[MF90]和 CLAUDIEN[DB93]，以便从数据中归纳出逻辑子句。专家认为这些归纳出的子句直观上是较为吸引人的，基本与他们知识相一致的。尤其是，专家赞赏所产生描述的符号表示，具有明确性。他们认为这是该方法优于处理同样数据的神经网络最大之处。

利用 ILP 归纳程序不变性

在正式证明过程程序正确性时，需要发现在程序某些地方始终成立的适当条件。这种前提条件必须充分强壮到可以蕴含程序的条件结果。特别感兴趣的是发现在程序循环内部为真的合适条件问题，称为循环不变量。通常构造循环不变量被认为是困难的，常常通过猜测来完成。Bratko 和 Grobelnik[BG93]研究得出 ILP 技术可用于自动构造循环不变量。一个被证明是正确的程序能够运行，所产生的运行踪迹可以作为 ILP 系统的学习样本。在程序给定点程序变量的状态代表与程序该点关联条件的正例。而反例则通过利用一种“受控闭路假设”来产生。在[BG93]中，对用于典型正确性证明练习的简单程序可以直接推出合适的循环常量。对于并行程序，也可以自动归纳出一个常量。将此方法应用到更大程序上还没有被研究过。

程序设计中数据精炼

在从更高阶说明构造程序中，定义语言（更高级）的函数要在目标语言级（低级）实现。因此更高层次的抽象数据类型需要精炼为目标语言层

次中的具体数据类型。例如：集合可精炼为列表。在[BG93]中，这种精炼问题也可以在 ILP 框架中解答。作为演示说明，利用通用 ILP 程序 Markus[Gro92]通过从更高说明层面归纳集合中的联合运算来加以实现。

来自第一原理的创新设计

Bratko[Bra93]提出了一种将创新设计归结为 ILP 问题的方法。设计过程被视为构造可用基本部件的过程，这些部件拼在一起可以实现某些特定行为。方法从“第一原理”来解决设计问题，因为一个工件的功能行为可从设计者所使用基本部件的原理而获得。方法包括：通过目标行为样本说明目标工件，定性原理定义可用基本部件的行为，以及将 ILP 作为概念性构造设备的机制。作为演示说明，利用 Markus 程序[Gro92]从其目标行为和一些简单的基本部件定性原理样本中来构造简单的电路。

定性系统辨识

动态系统理论中的一个基本问题就是系统辨识。它可定义如下：给定一个动态系统行为实例，发现一个模型来解释这些实例。受到学习定性模型要比定量模型简单的假设的鼓舞，Bratko 等人[BMV92]提出了一种将定性辨识归结为 ILP 问题的方法。在他们的工作中，模型就是定性差分方程（QDES），它们约束系统变量的取值。通常用于定性系统的 QDE 约束的 Prolog 实现被作为学习的背景知识。被建模的系统的行为样本作为正例，而附近的产生作为反例。利用通用 ILP 系统可以归纳出简单的动态系统模型。

4.6 总结

将 ILP 应用于工业或相关科学上，尚没有具有满意解答的难题。在所介绍的一些应用（尤其是诱变性），利用 ILP 处理工业或环境数据所获得的结果要比应用其他（不管是否使用 ML）方法要好。网格设计和诱变性就是关系背景知识是最自然问题的好例子。而且若可能，将其转换为属性-值形式将至少是相当笨拙的。在这些应用中，用户——领域专家——正对归纳出的概念描述的可理解性和含义越来越感兴趣。这将帮助他们更加深入地了解问题本身。ILP 表示的灵活性有助于理解。经验表明 ILP 是一个功能强大的、“知识密集”的数据挖掘工具。

在所有应用中，均利用了通用功能的 ILP 系统。因此一个典型 ILP 应用

就是要设计一个好的问题的关系表示，包括定义相关背景知识。与其他机器学习方法相比，ILP 系统的一个主要能力就是它们可将背景知识用 Prolog 程序来加以表示。这常常就能改变表示工程的规律，而这恰恰就是由 Langley 和 Simon[LS95]所指出的：这是机器学习的基本组成。

另一方面，目前更有效利用 ILP 的一个主要障碍就是现有 ILP 系统相对低效率，以及它们目前有限的处理数值数据能力。因此，对于属性-值表示的问题是足够了，仅仅由于效率原因，基于属性学习更为实用。

致谢

这里作者特别感谢 Ashwin Srinivasan，Ross King 和 Michael Sternberg，他们参与了结构分子生物应用的实验，Mihael Junkar 和 Igor Komel 参与了电子放电设备应用的实验，还有 Bojan Dolšak 参与了网格设计应用的实验。这项工作得到了 Esprit 基础研究行动 ILP（归纳逻辑编程）和 ILP²，斯洛文尼亚科学与技术部的部分支持，还得到了 Stephen Muggleton 所获 SERC 先进研究奖学金及牛津 Wolfson 学院支持 Stephen Muggleton 的研究奖学金的支持。

参考文献

[BFOS84]　L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and Regression Trees. Wadsworth, Belmont, CA, 1984.

[BG93]　I. Bratko and M. Grobelnik. Inductive learning applied to program construction and verification. In J. Cuena, editor. Knowledge-based Techniques for Software Engineering.North-Holland, 1993. Also in: Proc. ILP'93 Workshop, Bled, Slovenia, April 1993.

[BMV92]　I. Bratko, S. Muggleton, and A. Varsek. Learning qualitative models of dynamic systems. In S. Muggleton, editor, Inductive Logic Programming, London, 1992. Academic Press.

[Bra93]　I. Bratko. Innovative design as learning from examples. In Proc. Int. Conf. Design to
Manufacture in Modern Industries, 1993. Bled, Slovenia.

[DB93] L. De Raedt and M. Bruynooghe. A theory of clausal discovery. In Proc. Thirteenth International Joint Conference on Artificial Intelligence, pages 1058-1063, San Mateo,CA, 1993. Morgan Kaufmann.

[DBJ94] B. Dolsak, I. Bratko, and A. Jezernik. Finite-element mesh design: an engineering domain for ILP application. In Proc. Fourth Int. Workshop on Inductive Logic Programming ILP-94, 1994. Bad Honnef/Bonn.

[DDHRW94] S. Dzeroski, L. De Haspe, B.M. Ruck, and W.J. Walley. Classification of river water quality data using machine learning. In Proc. Fifth International Conference on the Development and Application of Computer Techniques to Environmental Studies (ENVIROSOFT'94), 1994.

[DJB96] B. Dolsak, A. Jezernik, and I. Bratko: Application of machine learning in finite element computation. 1996. This volume.

[DLdCD+91] A.K. Debnath, R.L. Lopez de Compadre, G. Debnath, A.J. Schusterman, and C. Hansch.Jnl. Medicinal Chemistry, 34:786-797, 1991.

[Dol96] Dolsak. Contribution to Intelligent Mesh Design for Finite Element Analysis, 1996.

Univ. of Maribor: Ph.D. Thesis (in Slovenian).

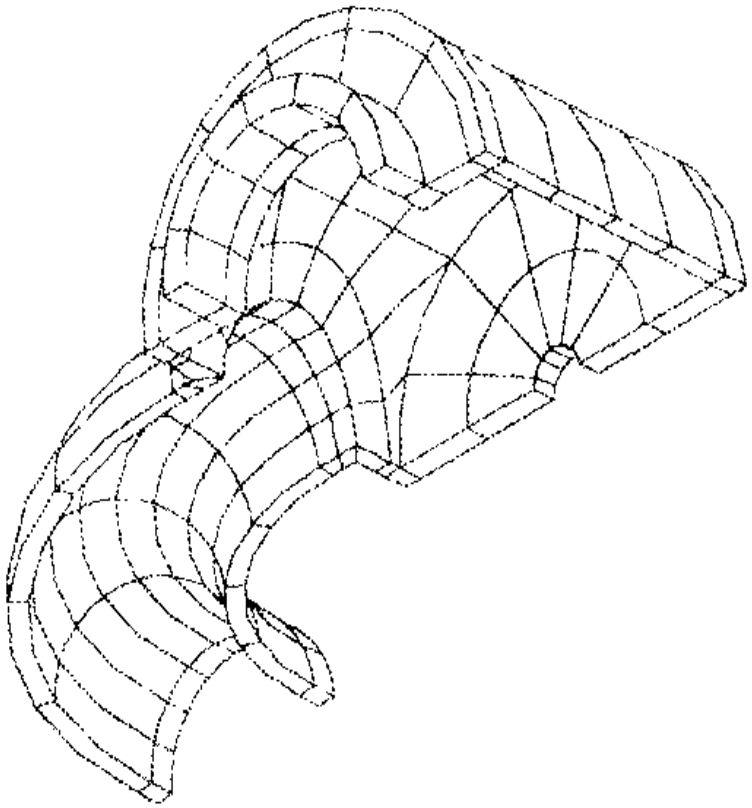
[Gro92] M. Grobelnik. Markus: an optimized model inference system. In Logic Approaches to Machine Learning Workshop, 1992. ECAI-92 Workshop, Vienna.

[HMFM62] C. Hansch, P.P. Malong, T. Fujita, and M. Muir. Nature. 194:178-180, 1962.

[JFv93] M. Junkar, B. Filipic, and M. Znidarsic. An AI approach to the selection of dielectricum in electrical discharge machining. In Proc. Third International Conference on Advanced

第 2 部分

设计与工程



第5章 机器学习在有限元计算中的应用

Bojan Dolšak,Ivan Bratko 和 Anton Jezernik

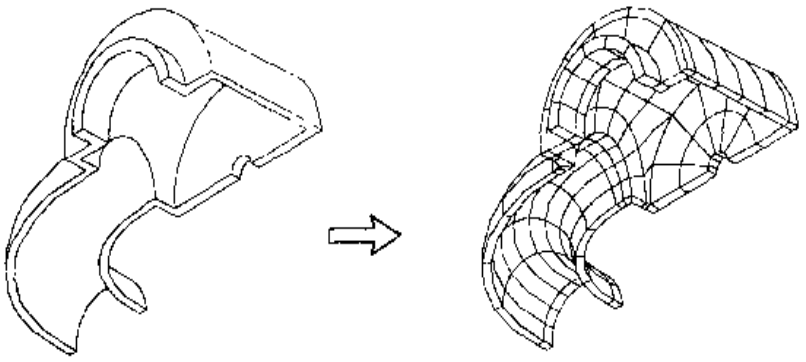
摘要

有限元计算方法（FEM）是工程中广泛用来分析物理结构受力与形变的最成功的数值方法。在 FEM 中，被分析的结构都要用有限元网格来描述。一个好的几何网格模型应该能够保证较小的近似误差并且避免不必要的计算开销，所以定义这样的模型是一项困难而耗时的工作，这是 FEM 分析方法的主要瓶颈。在实际应用中，FE 网格的设计是根据使用者的经验来完成的，还没有一个让人满意的通用方法来自动地构建 FE 网格。这里我们介绍几个 ML 系统在这方面的应用，它们根据已知的较好的网格实例来设计 FE 网格。在最近的大部分实验中，归纳逻辑规划（ILP）系统 CLAUDIEN 已经用来创建规则以决定合适的网格数值，十圆柱网格模型被用做训练例子的来源之一。对于得到的知识库的一项评估显示，归纳规则有效地获取了网格设计的模式。这种知识库产生的结果和常规的网格技术产生的结果比较也使人更加确认，对于网格设计问题来说，ILP 是一种有效的解决方法。

5.1 简介

有限元模型（Finite Element Method，FEM）是过去 30 年设计领域里最成功的数值方法，它已经被工程师和模型学家广泛地用于分析物理结构的受力的情况中。一个特定负荷的作用和机构的支撑情况可以用一组微分方程表示，但是，对于任意复杂的机构不可能在合理计算时间内解出这样的方程。因此，我们必须用一个网格模型（Mesh Model）来近似模拟实际的结构（如图 5.1（a）所示），这个模型有一组有限元（Finite Elements，FE）在结点处互相连接（如图 5.1（b）所示）。

结点处的置换被采用为关于问题的基本未知参数，按照结点处的置换选择一组函数来近似每个 FE 范围内的置换。作为离散化的结果，解出来的是一组线性代数方程而不是微分方程。Zienkiewicz 和 Taylor（1988）更详细地描述了 FEM。



（a）实际结构　　（b）FF 网格模型

图 5.1　FEM 离散模型

一个 FE 网格应该对应于结构的几何形状。在形状改变剧烈的区域需要更加精细的网格来保证低近的似误差。另一方面，由于每个增加的 FE 都会增加要解的方程的数量，所以在其他的区域使用比较粗糙的网格就足够了，这样可以避免不必要的计算开销。这样的“最优”网格是一个最粗糙的网格，同时又能够提供足够的准确率。图 5.1（b）展示了一个这种网格模型的例子（Dolšak，1996）。

不过要想知道网格什么时候应该精细和什么时候应该粗糙，需要相当多的关于 FEM 的经验和知识。很多参数，比如结构的形状、负荷和支撑等，都需要考虑到。现在大部分出售的 FEM 开发包都有自动建立 FE 网格的功能。

然而，这种情况只考虑了结构的几何形状，这只是众多影响参数中的一个，而且，在大多数情况下，并不是最重要的一个。通常，我们必须设计几个不同的网格模型，直到我们根据经验找到较好的一个（如图 5.2 所示）。

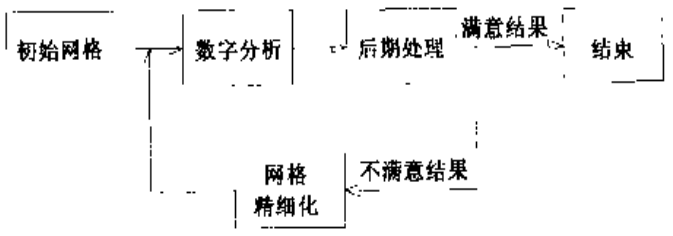


图 5.2　FE 网格模型设计过程

现在的问题是必须分析每个网格，因为下一个网格需要根据前一个网格得出的结果来产生。如果我们考虑到了每个 FEM 分析都要耗费几分钟到几小时（甚至几天）的计算时间，那么很显然我们有充足的动力去建立这样一个专家系统（ES），即它能在最初的循环里或至少在较少实验次数里设计出“最优”的 FE 网格。

在这一章，我们介绍一下关于这种 ES 的基本想法，以及用机器学习（ML）技术来产生 FE 网格设计知识库的一种尝试。这个研究很好地展示了应用 ML 工具的典型过程。这包括了用几种 ML 工具感知域和数据，根据初始实验所揭露出来的缺点改进数据集，探测 ML 工具合适的域定义的属性，相应地，如何选择工具使得这些属性更加适合将来的应用。这一章的结构是根据这个 ML 应用的例子来确定的。但是在接下来的一节，我们首先讨论如何将这个 ML 应用与现有的、常规的 FEM 软件结合起来。

5.2 向 FEM 产生器添加一个专家系统

除了自动的网格设计功能之外，所有的 FEM 开发包都包括手动创建 FE 网格的可选功能。这要求使用者指定分辨率值，例如元素的全局大小或是结构中边上元素的数目。由于网格自动设计在大多数情况下得不到满意的结果，手工进行网格设计是建立 FE 网格的常用方法。

给定了分辨率值，一个 FEM 预处理器就能够用一些内建的规则和算法来产生 FE 网格。

如果使用者指定了结构中位于边上元素的数目，可能会出现有的元素对应了一个距理想状况相差很远的几何结构的情况。在这种情况下，就需要使用自适应网格。在大多数的 FEM 开发包里都有这样的内建的选项。

手工设计网格的主要问题在于如何选取正确的分辨率值以保证得到“最优”的 FE 网格。我们设想这可以由结合了一个合适的 FEM 预处理器的 ES 来解决，如图 5.3 所示。

在任何情况下，使用者都要定义问题（几何形状、负荷、支撑）。关于问题的数据从 FEM 预处理器的格式转换为 ES 使用的符号化定性的描述。ES 应该可以确定网格分辨率值。根据 ES 产生的分辨率值，可以为网格生成器生成一个命令文件。

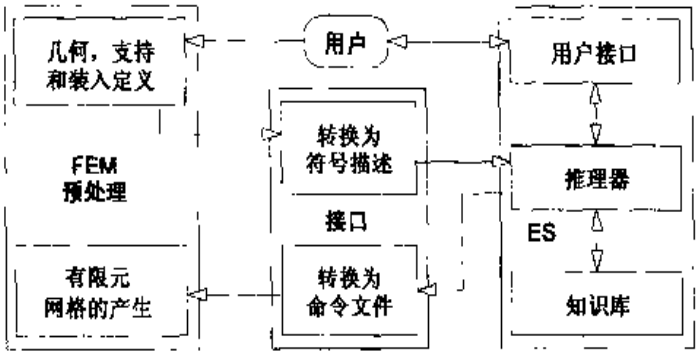


图 5.3　用于 FE 网格设计的一种集成 ES

主要的问题是如何建立一个网格设计知识库？FEM 已经广泛应用了 30 年，但是，仍然没有一个清晰的、令人满意的关于网格设计诀窍的形式化的方法，网格设计仍然是艺术与经验的结合体。不过，关于问题定义，一个令人满意的 FE 网格（几次实验后的选择）以及分析的结果，已经有了大量发表的报告。这些报告可以用做 ML 训练用例的来源之一。这为 FE 网格设计知识的自动获取提供了机会。给定的实例归纳后，报告中的问题定义和 FE 网格可以描述为 FE 网格设计构造规则的学习。在最后我们介绍 ML 的这样一个应用。

5.3　学习问题、实例和背景知识

5.3.1　问题的关系特性

一个 FE 模型包含由边组成的网络。边之间的关系对于 FE 网格的一个合适的分辨率来说很重要。为了把这种边之间的关系信息考虑进去，在关于网格设计的学习中应用关系学习技术就非常自然了。

因此我们应用几种归纳逻辑编程（ILP）系统，因为它们明确了关系学习的一种形式。一般来说，ILP 问题声明包括正面和反面实例以及背景知识（参看有关 ILP 应用的章节）。在这一节我们为基于实例和背景知识的网格设计学习设计了一套合适的表示方法。

5.3.2　实例来源

目前的学习集包括了 10 种不同的 FE 网格模型。把实际会出现的所有不同的结构全部考虑进去是不可能的。我们的学习集中的 FE 网格模型由于有以

下共同的特征而代表了一个体系。

- 所有的结构都是圆柱体的；
- 负荷都是由外力和压力造成的（没有热及其他的影响）；
- 不需要局部的高网格精细度。

关于 FE 网格模型的详细描述可以在 Dolšak（1996）里找到。图 5.1（b）显示了一个这样的模型。所有结构的 FE 网格是“手工制造”的，并且要修改好几次，直到保证在可以接受的计算机时间开销内得到的数值结果足够精确。

描述一条单一的边比整个结构要容易得多。单一的边代表一个较低层次的问题。相同类型的边可以在不同的结构中出现，这让我们可以用相对较小数目的边描述很大范围的结构。因此所有的 FE 网格模型都集中为一个边的集合。边由一个字母和数字的组合来标注。字母（a~i）指示整个网格模型，而连续的数字表示模型中单一的边。图 5.4 显示了图 5.1（b）中的 FE 网格模型里一些带标记的边。

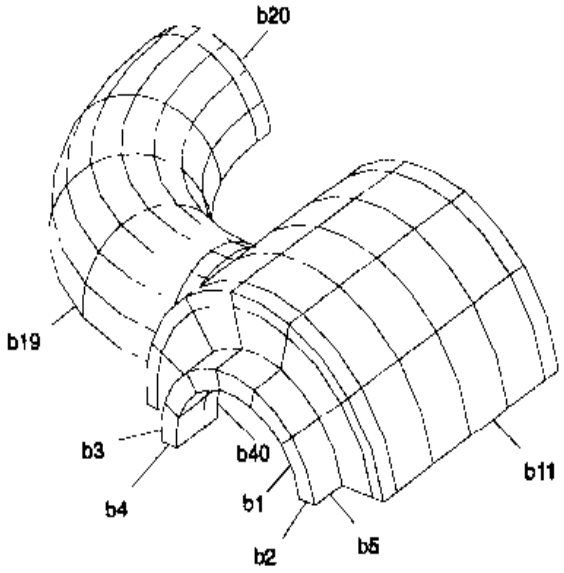


图 5.4　图 5.1（b）中的 FE 网格模型中的一些带标记的边

训练集中所有的边都定义了几何形式、负荷和支撑。另外，边之间的某些重要的拓扑关系也被定义了。正由于这些关系，一个关系学习算法非常自然地适合了这个问题。规定了边上面的有限元的数目就定义了 FE 网格。

5.3.3　正面实例

要被学习的关系表示为 mesh(E,N)，其中 E 是结构中一条边的名字，N

是这条边上面所推荐的有限元的数目。对于训练集中的每条边，有限元的数目都是由正面实例给出的。当然有些误差也是允许的。在 N≥8 的时候，对于每个实例 mesh(E,N)，都要加一个实例 mesh(E,N1)，N1=N±1。我们有时将一条边上面的有限元数目 N 称做边的**类**。

由于超过 12 个有限元的边是非常少的，我们决定只为 1 到 12 的类归纳规则。因此可能出现归纳出的规则无法对结构中所有的边归类的情况。不过这不是一个严重的问题，因为建立一个 FE 网格不需要为所有的边指定有限元数目。在这种情况下，FEM 预处理器会根据给定的分辨率值使用一些内建规则来确定缺少的值以构建一个 FE 网格。

5.3.4　反面实例

基本上，反面实例是根据有限维世界这个假设来构建的。所以反面实例是以边的名字的复合来构建，包括从 1 到 12 的所有数字，这与正面实例不同。另外，也考虑到了这产生的一些误差。对于所有含有 5 到 7 个有限元的边来说，1 个元素产生的误差对于反面实例来说无关紧要。对于包含 8 个或更多有限元的边，至多 2 个元素产生的误差是可以接受的。

针对单一类的学习，已经建立了一个反面实例的可选集。在这种情况下，除了当前**类**之外的所有类的正面实例都被当做反面实例。因此，举个例子，为了对类 1 学习规则，所有指定了边上有 1 个 FE 的实例都是正面实例，这种情况下的其他所有的实例都为反面实例。

5.3.5　背景知识

背景知识包括了谓词的定义，谓词可以用在关于目标关系 mesh/2 的假定中。它可以被分为两个部分：

- 边的属性值的描述；
- 边之间的拓扑关系。

5.3.5.1　边的属性值的描述

下列属于 FE 网格的边的属性描述为背景事实。

- 边的类型使用的谓词：

long,usual,short,circuit,half_circuit,quarter_circuit,short_for_hole,long_for_hole,circuit_hole,half_circuit_hole,quarter_circuit_hole,not_important.

- 支撑使用的谓词：

free,one_side_fixed,two_side_fixed,fixed.

- 负荷使用的谓词：

not_loaded,one_side_loaded,two_side_loaded,cont_loaded.

关于圆形的边有专门的谓词去描述边的类型。其他的边则根据它们的长度定性地分为“long”，“usual”和“short”。专门的谓词也用于结构中的一部件，因为这些部件通常对于分析很重要。不属于结构中重要部件的短边用谓词 not_important 来描述。

共有四个谓词描述支撑，只考虑边是固定的（完全固定或是端点固定）还是自由的，不根据自由度的数目来区分支撑。类似地，负荷的描述值仅表明负荷的位置。

每个定义了一条边的一个属性的谓词都有一个自变量——边的名字。训练集中所有的边的每个属性都由一个谓词描述。一条边可能会负载了压力（连续的）和力，这样属性要被两个谓词描述，但是我们的训练集中不包括这样的例子。

5.3.5.2 边之间的拓扑关系

由于适合 FE 网格的边之间的关系的存在，需要一种拓扑表示。我们认为相邻的和相对（平行）的边之间有一种连接，它影响着在边上的有限元的合适数目。另一种有趣的关系是边之间不仅是相对的，而且边都有相同的长度或形式。例如，同心圆有相同的形式。这样的边组成的偶对可用谓词 equal 来描述。所有三个谓词都是二元的。

一条边也可以与不只一条边相邻、相对或相等。正因为如此，所以最好使用允许非确定文字的 ILP 学习算法。一个形如 neighbour(a3,X)的文字称做非确定的，因为“输入”变量 a3 不必决定“输出”变量 X（如果 a3 有几个相邻，X 可以是其中的一个）。

尽管如此，“确定”性的背景知识描述也被 ILP 程序 GOLEM（Muggleton 和 Feng，1990）用在了我们早期的实验（Dolšak 和 Muggleton，1992；Dolšak 等人，1994）中，它只限于确定文字。

5.3.6　学习集概要

当前实验中使用的学习集包含了非确定性背景知识里的 4 029 条事实（如表 5.1 所列）和 644 个正面实例（如表 5.2 所列）。

表 5.1　背景事实分布状况

背景描述	训练样本										Σ
	a	b	c	d	e	f	g	h	i	j	
long	3				2		6	2	2	2	17
usual	2	6	6	4	59	6	18	21	16	14	152
short	14	10	2	15	20	10	8	18	4	8	109
circuit			10	28							38
half_circuit	11	8				10	20	2	2	2	55
quarter_circuit		4			6			10			20
short_for_hole	4	4	2		2			2			14
long_for_hole	1		1		2			2		2	8
circuit_hole			4			9			2		15
half_circuit_hole	4	6			4	2		3		2	21
quarter_circuit_hole											0
not_important	16	4	3	10	1	4	8	11			57
free	2		13	53	34	9		16	10	14	151
one_side_fixed	3		6		24			8	8	8	57
two_side_fixed	15	12			4	10	18	10	2	2	73
fixed	35	30	9	4	34	22	42	37	6	6	225
not_loaded	21	23	17	50	88	19	52	63	10	14	357
one_side_loaded	5			4	4	6	4	4	8	8	43
two_side_loaded					1				2	2	5
cont_loaded	29	19	11	3	3	16	4	4	6	6	101
neighbour	220	168	84	168	382	116	240	280	96	120	1 874
opposite	28	38	28	12	24	10	54	34	6	8	242
equal	34	32	32	20	19	38	80	68	34	38	395
总计	**447**	**364**	**228**	**371**	**713**	**287**	**554**	**595**	**214**	**256**	**4 029**

惟一一个没有出现在背景知识里的推荐的谓词是 quarter_circuit_hole。这样的边非常少，所以这个忽略无关紧要。

表 5.2 正面实例分布状况

类别	训练样本										Σ
(No. of FE)	a	b	c	d	e	f	g	h	i	j	
1	21	9	6	14	23	12	10	16	8	4	123
2	9	9	6	13	36	4	10	21		2	110
3	3	4	2		11	4	4	4		6	38
4	2	1		2	5		8	3	12	14	47
5	3				9	2		5			19
6		11			4	5	2	2		2	26
7		4					4	6	2		16
8	1	4	14		6		20	5	2	2	54
9	1	4	14		7	4	20	12	2	2	66
10	2				7	4	2	7	2	2	26
11	15			28	2	4	2	7			58
12	15			28	1		2				46
13								1			1
14								1			1
15						10					10
16									2		2
17	1										1
总计	**73**	**46**	**42**	**85**	**111**	**49**	**84**	**90**	**30**	**34**	**644**
原始	55	42	28	57	96	41	60	71	26	30	506
添加的	18	4	14	28	15	8	24	19	4	4	138

5.4 以前的实验

网格数据集有可能是第一个实际的关系数据集。这使得网格设计成为实验 ILP 系统的应用最广泛的领域。所以在最近的几年里已经有几个学习系统应用于 FE 网格设计问题。它们绝大部分都是关系学习算法。前面已经提到，第一个 GOLEM 的实验使用了一个比较小的训练集（Dolšak 和 Muggleton，1992）。在此之后训练集已经被扩展，并且改变成 Dolšak 等人（1994）里描述的形式。基本上，训练集的形式与前面章里讨论的一样，只有前五个训练模型（a～e）是它描述的。在这一节的最后，我们将讨论使用这样一个学习集的 ML 实验的要点。

5.4.1 GOLEM 的实验

GOLEM（Muggleton 和 Feng，1990）是一种基于最小关联归纳（Relative

Least General Generalisation,RLGG）（Plotkin,1969）的通用 ILP 算法。它数次应用于有限元网格设计问题中。迄今为止这一领域所取得的最好结果是由 Dolšak 等人发表（1994）。GOLEM 在以下的实验性架构中已经运行数次：

- 使用了确定的背景知识；
- 使用了根据有限维世界假定构造的反面实例；
- 尽管一些归纳规则涉及到反面实例，它们仍被接受。

为了排除那些从应用的实际观点来看，对于将新边进行归类没有任何用处的归纳规则，我们制订了一些条款。归纳规则的准确率用 10 次的交叉确认来衡量。根据这个估算方法，整个实例集被随机地划分成 10 个大小相等的子集。然后开始进行 10 次学习测试迭代。在每次迭代中，一个不同的子集从训练集中移除。根据如此得到的 90%的实例进行学习，而归纳规则的准确率根据移出的 10%的实例进行测试。测试集上次迭代的平均归类准确率是 78%，有偏大的标准差（为 5.7）。

一个类似的实验性结构被用在了由 Sašo Džeroski 完成的实验中，这在 Lavrač 和 Džeroski（1993，9.2 节）中有描述。在这个关于 GOLEM 的实验中，不允许递归子句存在。从实际观点来看，没有用处的规则并不能被排除。在分类时，规则根据由拉普拉斯概率估计得出的期望分类准确率来排序。在一项实验（Džeroski，1991）中，准确率低于 75%的子句被丢弃了，在他的第二个实验（Lavrač 和 Džeroski，1993）中，分类准确率极限提高到了 80%。

准确率是用“去除一个结构”方法估计的，这个方法在后来的实验里也被广泛接受了。这是一种 K-级交叉确认的变种。但是实例集不是随机地划分为 K 部分的，而是根据边属于哪个结构来划分的。每个实例子集中边都确切地属于结构中的一个（在这个最初的例子里是五个）。这种“去除一个结构”测试被认为是网格设计领域里最自然的方法，因为从训练结构中归纳出的规则用来为剩下的、用于测试的结构设计一个网格。在最后我们将这个测试简称为“去除一个”。

由去除一个测试度量出的准确率是很低的。Džeroski 得出在训练集准确率极限为 75%的情况下，测试集的准确率为 29%。

5.4.2　FOIL 的实验

FOIL（Van Laer 等人,1994）对 ILP 范例的属性值学习算法进行了一些扩

展。特别的是，它使用了类似于 AQ（Michalski,1983）的覆盖方法和类似于 ID3（Quinlan,1996）启发式的基于知识的搜索方法。FOIL 可以处理非确定性文字。除了 GOLEM 外，其他所有的算法都应用在了有限元网格设计问题中。FOIL 是由 Sašo Džeroski 以 GOLEM 实验中所用同样的参数来运行的。进行去除一个测试时，分类准确率只有 12%左右（Džeroski，1991）。谓词 neighbour 几乎从来没有用在归纳子句里，这可能与 FOIL 从候选文字里选取文字时使用的局部启发式算法有关。

FOIL 的另一个问题是编码长度的限制，这显然妨碍了一些非常有用的子句的归纳。

5.4.3 mFOIL 的实验

ILP 系统 mFOIL（Džeroski，1991）是基于 FOIL 的。它使用一个子句的准确率（由 m-概率估计方法估计）进行启发式搜索，而不是像 FOIL 那样用基于统一性的信息获取启发搜索。作为这项改进的一个结果，分类准确率提高到了 22%（Džeroski，1991）。与 FOIL 类似，mFOIL 也有相邻关系的问题。

5.4.4 CLAUDIEN 的实验

CLAUDIEN 在整个解空间里搜索有效的规则，但是并不保证归纳假设能够覆盖所有的正面实例。同时，一个假设可能会多次覆盖一个正面实例。CLAUDIEN 算法（de Racdt 和 Bruynooghe,1993）的主要特点就是有一个允许对语言偏好进行说明的优良机制。在我们的情况里这一点非常有用，因为这样一来实际没有用的归纳规则就可以很容易地被排除。但是，这一点并没有被 CLAUDIEN 的作者在他的实验（Van Laer 等人,1994）中采用。不过，去除一个测试的结果要好于 FOIL 和 mFOIL 得到的结果。报告的分类准确率是 28%。在这个实验里，学习过程的时间被限制为 Sparc 工作站的 1000 CPU 秒。

5.4.5 MILP 的实验

MILP 算法（Kovačič,1994）是随机搜索技术的一种实现，它取代了在现在的 ILP 算法里使用的贪婪搜索技术。有限元网格设计领域被用来测试这种

方法的功效。MILP 优于 FOIL，mFOIL 和 GOLEM。报告中说进行去除一个测试，归纳规则的分类准确率大约为 32%。

5.4.6　FOSSIL 的实验

FOSSIL（Fürnkranz，1994a）是一种类似于 FOIL 的 ILP 系统，它使用了基于统计相互关系的启发式搜索。FOSSIL 的作者 Johannes Fürnkranz 进行了一些实验，用 FOSSIL 从描述的 5 种结构数据集中归纳分类规则。然后，除去一个测试，从 5 个结构中的 4 个学习规则，并在剩下的第 5 个结构上进行测试。最好的分类准确率是 35%左右（Fürnkranz，1994b）。

5.4.7　属性值算法的实验

人们也进行了一些关于属性值学习算法的实验（Kononenko 等人,1994）。只使用属性描述并忽略边的关系描述，去除一个测试边分类的正确率在 27%到 34%之间！再加上源自关系背景知识的 12 项属性，准确率会更显著地提高。最好的结果(44%)是用 ASSISTANT-R 程序得到的，这个程序是对 ASSISTANT 学习系统从顶向下决策树（Cestnik 等人，1987）的 RELIEFF 属性估值方法的改进。这些属性值学习得到的准确率远比一个自然表现出需要关系学习领域所要求的要高。这一点在稍后还要结合更多最近的结果来讨论，人们仍在讨论关系学习是否比属性值学习更加适合这个领域。

5.5　选择一个合适的学习算法

初步的针对有限元网格设计问题的 ML 实验结果看起来相当不稳定。但是，在 Dolšak 等人（1994 年）关于 GOLEM 实验的描述中提出要区别对待，有两个原因：

（1）只有在这个实验里才没有通过去除一个策略来测试分类的准确率。

（2）为了消除从实际观点看没有用处的规则，我们使用了问题工程方面的额外知识。

在忽略掉这个实验后，对于我们前面提到的其他实验得到的结果，可以简要地评价如下：

- 用去除一个方法测试的分类准确率低于 50%（12%~44%）。因此这些规则的实际应用是很有限的。
- 属性值算法得到的结果要优于几乎所有 ILP 的结果。

我们可以说背景知识中的关系描述是没有实质作用的！另一方面，在 GOLEM（Dolšak 等人，1994）所归纳的知识库中，62 条规则中的 55 条包含拓扑关系。此外，在引入了由背景关系描述得出的 12 项额外属性后，属性值学习的结果得到明显提高也是事实。所有的这些让我们得出关系描述是有益的这一论断。更进一步，一些关系表示到等价属性的转换产生了一个不如原始关系描述更有意义的描述：产生的属性描述不再能够很自然地被解释。因此关系学习算法在这一领域更受欢迎。

但是，比起属性值学习算法来，ILP 算法的糟糕表现仍然让人心存疑虑。这种表现是由训练集的自然性质所造成的。对于五个训练模型中训练实例分布情况的进一步研究表明：在某种程度上说每种结构都是惟一的并且都在训练集中拥有相当数量的实例。这样去除一个测试就不合适，因为整个结构都很容易被去除而没有留下与测试实例特征相近的训练实例来。这五个结构之间的不同，表明去除一个测试不是很适合于这种情况。在基于属性的学习中，由于用于实验的原始贝叶斯公式没有被归纳策略树“覆盖”，所以较好地解决了这个问题。

为了确保进行去除一个测试所必需的条件，训练实例的分布被训练集的扩展而改善了。现在的训练集包含了 4 个额外的结构（f~i）。为了确保使用新训练集的 ML 有好的结果，选择学习算法时，注意下面的特点是很重要的：

- 关系学习；
- 能够处理非确定性文字；
- 能够容忍噪声，子句所覆盖的正面实例和反面实例的数量，都要考虑到；
- 在学习过程中实际上没有用处的归纳规则应该被除去。

ILP 系统 CLAUDIEN（De Raedt 和 Bruynooghe,1993）具有上面提到的所有性质。而且，它不需要反面实例。因此，CLAUDIEN 被用来描述新的训练集。CLAUDIEN 的一个非常有用的特性是，它允许使用者宽松地制订从归纳算法中隐含产生的通用形式和内容。

5.6　根据 CLAUDIEN 学习

决定有限元网格的分辨率值的规则的学习是分 6 步进行的。在每一步中都制订一种不同形式的规则让 CLAUDIEN 归纳。并且，我们也通过设置要求的准确率（被覆盖的实例中正面实例的比例）和覆盖度（被覆盖的实例的数量；这理解为训练集中使得归纳子句为真的代换的数量；所以，一个例子不需要对应一个实例）来影响学习进程。

在接下来的段落里我们将描述这 6 个步骤。我们会给出归纳规则和规则格式的例子。它们使用 Prolog 语法，也被 CLAUDIEN 接受。

第 1 步，CLAUDIEN 归纳 17 条只包含边的属性描述的分类规则。以下的规则格式是为这个目的制订的：

```
clausemodel( 'mesh(Edge1,[1,2,3,4,5,6,7,8,9,10,11,12]) <- \
        +1[(Type(Edge1),Support(Edge1),Load(Edge1)' )].
```

这就是说一条归纳规则的结论部分是对一个制订为边 1 上的 12 个有限元 mesh/2 形式的文字描述。规则的条件部分可以提到至少一项，最多三项边 1 的属性（类型、支撑和负荷）。例如，10 号规则有 100% 的准确率（perc_cov(1)），这是在 14 CPU 秒多一点的时间里归纳出的，它是：

```
rule(10,[perc_cov(1),body(4),cpu(14.1667)],
    (mesh(Edge1,11) :-long(Edge1),
    one_side_loaded(Edge1))).
```

它在一边负荷的长边上制订了 11 个有限元素。从规则的描述也可以看出，在训练集中有四个代换使规则体为真。

在学习过程的第 2 步允许使用单一的拓扑关系。为了避免归纳出实际无用的规则，目标假设的语言制订了以下的条件。

- 规则必须至少包含实际边描述的一项属性；
- 由拓扑关系引入的新边必须至少在一项属性里详尽描述过。

```
Clausemodel('mesh(Edge1,[1,2,3,4,5,6,7,8,9,10,11,12])<-\
        <+1[Type(Edge1),Support(Edge1),Load(Edge1)],\
        [<Relation(Edge1,Edge2),\
        +1[Type(Edge2),Support(Edge2),Load(Edge2)]>] ').
```

谓词变量“关系”被定义为相邻、相对或相等中的一个。由于指定的限

制，351 条规则在第 2 个学习步骤中被归纳。这里有一个例子，一条规则为“通常”的边指定了一个有限元素，这些边有连续的负荷并且与相邻边都是以四分之一圆周的几何形式相邻接的。

```
rule(340,[perc_cov(1),body(4),cpu(19375.2)],
    [mesh(Edge1,7):-
        usual(Edge1),
        neighbour(Edge1,Edge2),
        quarter_circuit(Edge2),
        two_side_fixed(Edge2),
        cont_loaded(Edge2)]).
```

有两个拓扑关系的分类规则是在两个学习步骤里归纳的。首先，建立都指向“目标”边的两个拓扑关系的规则。例如下面的规则，它在第 3 个学习步骤中归纳，包含一个当前边（Edge1）的两个拓扑关系（相邻和相对），它们作为每个子句的第一个参数：

```
rule(166,[perc_cov(0.909091),body(22),cpu(1026)],
    [mesh(Edge1,9):-
        half_circuit(Edge1),
        not_loaded(Edge1),
        neighbour(Edge1,Edge2),
        usual(Edge2),
        opposite(Edge1,Edge3),
        half_circuit(Edge3),
        not_loaded(Edge3)]).
```

CLAUDIEN 在一台 SUN 的 Sparc 工作站上用了 128 207 CPU 秒的时间归纳了 1 988 条规则！但是，当搜寻描述了一个边链包含了两个拓扑关系的规则时，第 4 个步骤里的搜索空间更加复杂。尽管要求准确率和覆盖度都提高了（如表 5.3 所示），在第 4 个学习步骤里用了差不多 300 000 CPU 秒去建立 1 700 条规则。一个例子如下：

```
rule(1535,[perc_cov(1),body(8),cpu(64226)],
    [mesh(Edge1,3):-
        cont_loaded(Edge1),
```

```
neighbour(Edge1,Edge2),
half_circuit(Edge2),
cont_loaded(Edge2),
neighbour(Edge2,Edge3),
not_important(Edge3),
one_side_loaded(Edge3))).
```

表 5.3　基本学习参数

步骤	规则形式	准确率	覆盖率	规则	CPU 秒
1	Class rules without relations	≥ 0.90	≥ 3	17	79
2	Class rules with one relation	≥ 0.90	≥ 3	351	26657
3	Class rules with two relations (Edge1)	≥ 0.90	≥ 3	1988	128207
4	Class rules with two relations in chain	≥ 0.95	≥ 10	1700	299833
5	Interval rules with one relation	≥ 0.98	≥ 20	395	894573
6	Limits depending on the edge type	≥ 1	≥ 8	11	8666
				4462	1356237

在这种情况下，两条新的边 Edge2 和 Edge3 作为目标边 Edge1 的相邻链被一起引进了。

在第 5 个学习步骤里，我们允许归纳的规则指定多于一个的类，也允许了一个拓扑关系的使用。在这个例子我们考虑训练集中出现的所有 17 个类。另一方面，规则的要求准确率和覆盖度进一步提高了。CLAUDIEN 用了 10 天，可能还要多的 CPU 时间来归纳 395 条符合这一步里给定的规定的规则。尽管也是一些丢失的类的实例，它们中的大部分规定了类间的间隔，例如：

```
rule(362,(perc_cov(1),body(26),cpu(775726)),
        (one(A);two(A);four(A):-
        mesh(Edge1,A),
                one_side_fixed(Edge1),
                neighbour(Edge1,Edge2),
                free(Edge2),
                cont_loaded(Edge2))).
```

这条规则为一端固定的，另一端为自由和连续的负荷的边指定了一个、两个或 4 个有限元素。

在最后，第 6 个步骤里，间隔指定又被允许了。这次在子句体里只考虑边的类型。

对于每个描述类型的谓词，都归纳了一条规则来根据边的类型规定可能的有限元数目。因此，例如，下面的规则说明训练集中的短边有 1~4 个有限元：

```
rule(2,(perc_cov(1),body(109),cpu(13.9333)),
    (one(A);two(A);three(A);four(A):-
    mesh(Edge1,A),
        short(Edge1))).
```

整个训练过程的基本参数列在表 5.3 中。CLAUDIEN 在大约 15 天的 CPU 时间里归纳了 4 462 条规则！学习过程可以在一个指定的 CPU 时间后停止，但是这个选项没有被采用，因为归纳的规则的次序与质量之间没有联系。

归纳的规则提到了所有的背景谓词以及所有出现在训练集中的类。

5.7 归纳的规则的后期处理

发现归纳得到的规则并不适合直接放入知识库。它们中的很多后来被去除了，剩下的规则的形式和次序为适合应用的要求而进行了调整。在后期处理过程中，确切规定了有限元数目的规则和有间隔的规则被分开处理。根据以下的条件来去除规则：

- 它们覆盖的正面实例是否少于 3 个；
- 它们是不是复制品；
- 它们是否被一个更通用的规则包含了；
- 它们是否只覆盖了增加的正面实例；
- 它们是否与其他的规则有相同的规则体而指定了不同的类（在有间隔规则的情况下）。

第一个条件看起来是多余的，因为训练集中最少有 3 个代换使得子句体为真已经在学习过程中考虑过了。但是，因为代换的数目并不总是与规则覆盖的正面实例的数目相等，所以这一点必须考虑。一个正面实例可能会造成不只一个的可行代换。例如，一条边可以有最多四条相同属性的相邻边。

在学习处理后，总共排除 2686 条规则。为了保证较高的效率，剩下的 1776 条规则的所有元素中，对于实际应用不是必要的，都被排除了。在另一方面我们启用了一个防止无限循环的机制来处理在前面的一个学习例子（Dolšak 等人，1994）里归纳出的并加入了知识库的递归规则。

在FEM预处理环境里，使用者必须为将要优化的结构的每一条边指定有限元数目。因此，由有间隔规则规定的，从推荐的类列表中选择最合适类的规制也被加到了知识库里。基本上，这些规则比较两种边的几何类型，从而为某一边给出训练集中平均使用类之间的差异。例如，下面的规则规定了通常的边拥有的有限元数目平均比长边少6个：

```
compare_type(Edge1,Edge2,-6):-
        usual(Edge1),
              long(Edge2).
```

比较规则是根据训练集的简单统计建立的。知识库中的比较规则使得推理机制可在从推荐类列表中选择时考虑到相对边的类型。如果一个单一类没有相对边，它就按照推荐类的数学平均值计算。

最后知识库里还加入了决定有限元的合适类型的规则。它们是手工建立的，考虑了空间维数、几何复杂度、负荷情况和三维结构的厚度，指定元素的主要和第二类型。元素的命名采用FEM包BERSAFE（Hellen，1970）里的规定。这里有一个例子，是一个有二阶近似函数的固体元素的规则：

```
finite_element(ez60,ez45):-
        space_dimension(3),
        thickness(thick),
        (geometry_complexity(high);
          loading_case(complex)).
```

知识库中所有的规则都是按照ES的有效性和准确率来排序的。因为自顶向下的搜索策略是用Prolog实现的，最可靠的规则放在知识库的顶层。

对于分类规则来说这一点尤其重要，这样可以保证使用最好的规则决定结构中每条边上面的有限元数目。这里有几个定义知识库中分类规则合适次序的标准。

- 对单一类分类的规则先于有间隔的规则放入知识库，这样可以使利用比较规则从推荐的类列表里选择最合适的类的需要最小化。由于相同的原因，递归规则紧跟着单一类的规则放入知识库。
- 没有覆盖反面实例的规则有优先权。
- 描述当前边类型的规则先于剩下的其他规则放入知识库，因为边的类型（相对长度）是最显著依赖于有限元数目的。

- 具有 100%准确率的规则根据一个和两个几何关系从简单属性规则到复杂规则排序。相反的排序就用在包含了反面实例的规则上。

5.8 结果

5.8.1 知识库与 ES Shell

最后，前面章节所描述的知识库接收了 1 873 条规则和 31 个事实。知识库中规则/事实的数量与排序如表 5.4 所列。目前的这个知识库是考虑了这一领域所有实验最全面的一个知识库。这样说一个原因无疑是训练集的扩展。另外，我们应该注意到第一次归纳了有间隔分类规则，而且在学习过程之后知识库里加入了一些额外的规则。知识库是在 Prolog 的规则和事实的形式下按照 Prolog 的语法写的。因此它有充足的透明度并且可以在需要的时候很容易地被扩展。

表 5.4 知识库中规则的数目、次序和事实

单个类别的分类规则	1 538 个规则
Acc=1	
包含对实际边类型描述的属性规则	9 个规则
具有一个几何关系和对实际边类型描述的规则	114 个规则
具有两个几何关系和对实际边类型描述的规则	768 个规则
具有两个几何关系，但没有对实际边类型描述的规则	417 个规则
具有一个几何关系，但没有对实际边类型描述的规则	44 个规则
0.9≤Acc＜1	
包含对实际边类型描述的属性规则	1 个规则
具有一个几何关系和对实际边类型描述的规则	19 个规则
具有两个几何关系和对实际边类型描述的规则	89 个规则
具有两个几何关系，但没有对实际边类型描述的规则	71 个规则
具有一个几何关系，但没有对实际边类型描述的规则	6 个规则
递归分类规则	1 个规则
间隔分类规则	227 个规则
Acc=1	
包含对实际边类型描述的属性规则	9 个规则
具有一个几何关系和对实际边类型描述的规则	114 个规则
具有一个几何关系，但没有对实际边类型描述的规则	44 个规则
0.9≤Acc＜1	
包含对实际边类型描述的属性规则	1 个规则
具有一个几何关系和对实际边类型描述的规则	19 个规则
具有一个几何关系，但没有对实际边类型描述的规则	6 个规则

续表

分类限制	11 条事实
比较规则	102 个规则
有关有限元的事实	11 条事实
本系统所使用的，有限元的引入	1 条事实
有关有限元兼容的事实	18 条事实
有限元选择规则	5 个规则
总计：1873 个规则+31 条事实	

ES Shell 也是用 Prolog 写的。它使得可以用知识库进行有限元网格设计，以及使用者与系统之间的通信比较简单。用户接口的另一个有趣的特性是解释推理过程的能力。

5.8.2　对专家系统的评价

已经进行了对于 ES 全面的评价。首先，准确率用几种方法测试。另外，引入了称做分类耗费的、包含信息更丰富的评判标准。它把错误分类时错误的多少也列入了考虑范围。错误分类耗费定义为：

$$Cost=|N1-N2|/max(N1,N2)$$

N1 是有限元的参考数目，N2 是 ES 规定的数目。错误分类耗费被规格化为 0 到 1 之间，较小的耗费意味着较好的分类。在我们的领域，可能的最坏错误是把 1 元的边分到“类”17。这会招致 0.94 的错误分类耗费。

ES 的结果已经在实际中应用于为 FEM 预处理器产生的网格确定分辨率值。这些测试的结果也在这一节出现了。

5.8.2.1　对于从训练集中取出的边的测试

目前的 ES 已经用来为训练集中所有的边确定有限元的数目。分类正确率为 78.26%，错误分类耗费为 0.092。

5.8.2.2　“去除一个”的测试

在这里对于每个训练模式都进行了分类准确率和耗费的测量。因为它的时间复杂性，学习过程并不是每次对于一个不同的训练集都重复，相反，对于当前结构不属于训练集而无法归纳出的规则，就要从知识库里去除。在 Dolšak（1996）里有关于排除过程的详细描述。尽管这个处理不能完全保证没

有被排除的结构而得到的结果是否会有不同，但结果不同是不大可能的。

分类准确率在 40.48%到 80.46%之间，平均为 59.09%。单一结构的最低分类错误耗费为 0.064，而最高为 0.244。

在流行的计算机辅助设计包 I-DEAS（I-DEAS，1993）中“去除”掉的结构的 ES 的结果也被用做 FEM 预处理器产生的网格的基本参数。让我们考虑一个拥有最差分类表现的结构情况作为例子。为了满足内在的网格构建方法的要求，结构被划分为 6 个子块（如图 5.5 所示）。

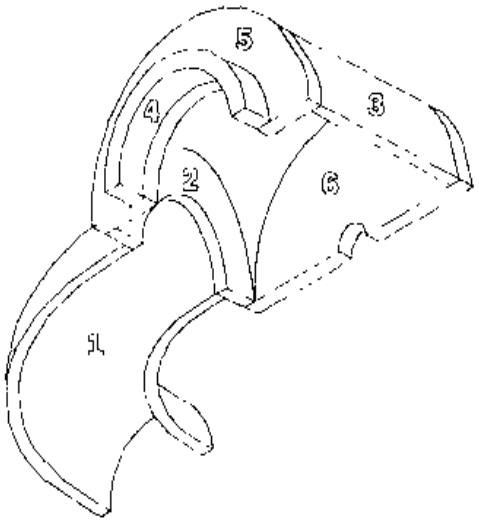
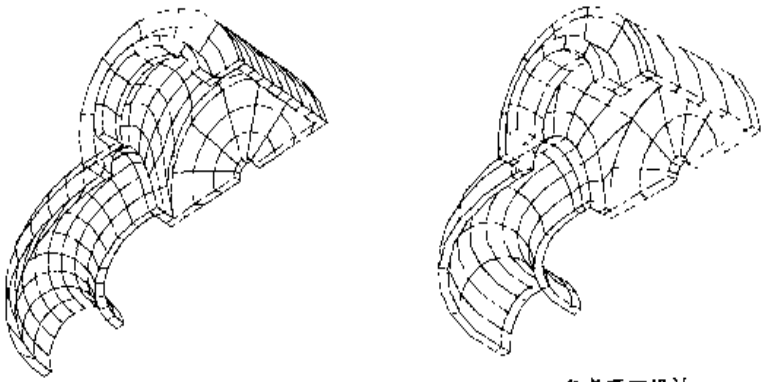


图 5.5　为匹配网格构建过程而进行的划分

尽管分类准确率很低，根据 ES 产生的结果（如图 5.6（a）所示）构建的网格作为一个初步的尝试来说还是不错的。与参考手工设计的用做数值分析的网格（如图 5.6（b）所示）的对比显示，ES 在一些边上面稍微多指定了一些元素，但是网格的整体图案几乎是相同的。



（a）专家系统的设计　　（b）参考手工设计

图 5.6　知识库与手工设计网格之间的一个对比

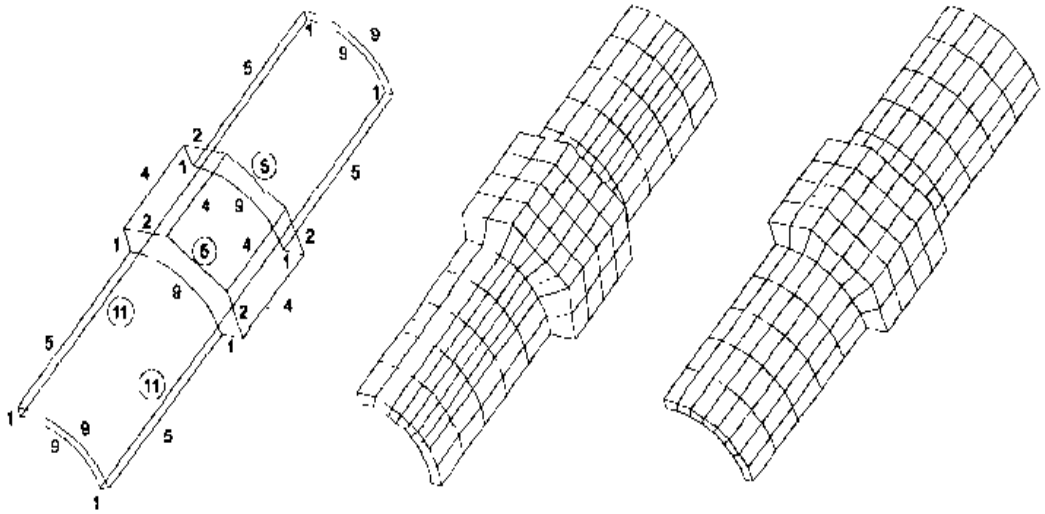
5.8.2.3　随机选取子集的 10 次交叉测试

训练集随机分为 10 个子集。从训练集中对于每个子集的删除与去除一个测试中的方法一样。平均有 70.16%的边正确分类。平均错误分类耗费为 0.127。

5.8.2.4　对于一种未曾出现的结构的测试

在最后的测试中，ES 为一种没有被包含在训练集中的全新的柱状结构指定网格分辨率值。分类准确率是 86.67%而错误分类耗费为 0.028，这个结果相当令人鼓舞。只有四条归类错误的边（图 5.7（a）中画圈者）。对于所有的四分之一圆周的边，ES 只比参考设计多指定了一个有限元，但是根据我们训练集的构建，这引起的误差是可以允许的。

根据 ES 的结果，FEM 预处理器构建了与手工的设计（如图 5.7（c）所示）几乎相同的有限元网格（如图 5.7（b）所示）。这里，为了满足映射网格程序的要求，结构再一次划分为最多 6 个部分。这个测试也在这个领域里显示分类准确率，作为成功的标志，有可能会使人误导。原因是自动网格生成器会自动更正一些错误。这种情况在图 5.7 中有发生。映射网格技术在表面上相对的一边上假定了相同的有限元数目。FEM 预处理器在 ES 指定了 11 个有限元的边上放置了 14 个有限元，因为在相对的另一边有限元的数目也是 14。



（a）ES 分类　　（b）知识库设计　　（c）参考手工设计

图 5.7　对于新结构的测试

5.9 总结

根据对 FEM 系统所做的实际实验，从人类专家的经验提取关于有限元网格设计的知识是很困难的。设计一个“优化”的有限元网格需要做大量的工作，也很难描述清楚。

另一方面，以前一直用于分析有限元网格的模型可用做机器学习实例的来源。归纳知识获取被证明为是解决有限元网格设计问题的有效方法。针对有限元网格设计的知识库并不是我们的实验的惟一结果，它同时也是 ILP 在这项工作上的能力以及通用方面前景的展示。

所呈现的知识库是非常广泛的，但是如果必要的话，为了提升性能，基于产生规则的构建也允许采用一个相对简单的方法。使用这个知识库的有限元网格设计 ES 是用 Prolog 建立的。这使得知识库的使用以及用户与系统的对话都很方便。它的一个特性是解释推理过程的能力。因此 ES 也可以用做给初学者的教学工具。ES 对时间和存储空间的要求都是适量的，没有表现出任何严重问题。

考虑对 ES 的评价的总体结果，可以做出以下结论：

- 与前面所有的实验相比，根据去除一个测试得出的分类准确率有明显的提高，这肯定了训练集的扩展是成功的。
- 去除一个测试与 10 次交叉确认测试得到的结果的对比也表明训练实例的分布是令人满意的。交叉确认的结果要稍微高一点，但是差别是很小的。
- ES 的结果表现为实际应用的坚实基础。尽管 ES 的结果看起来稍微有些低，但是根据 ES 的结果产生的网格与参考网格已经没有明显区别。对全新的结构的测试更加加强了 ES 的应用性。
- 在大多数情况下，ES 的结果适合网格的产生方法并因此易于受修改的影响。
- 当知识库里的规则指定了推荐的类的列表，就要决定“最适合”的类。尽管对决定“最适合”类的方法加以了特别的关注，知识库中的规则仍然允许对比目前使用的规则更好的规则加以应用。

人类专家检查归纳出的规则，参照他们自己的专家知识评定这些规则的意义。作为专家对归纳出规则评估的最重要的总体结论，规则的形式正是人

们所期待的，并且通常符合人类专家所使用的知识。

为了简化学习问题，学习集是按照对某种特定的结构有代表性的这个目标来设计的。但是现有的ES也可以用做通用的工具来为这些类型以外的结构决定网格的分辨率值。这些值随后必须根据特定分析的专门需要进行调整。

ES应用使得在类型范围内为一个结构设计合适的有限元网格成为可能。对于不同类型的结构，ES这样的效率是难以估计的。但是，ES所指定的分辨率值总是可以作为一个初始的有限元网格的基础，这个网格还可以根据数值分析的结果进行调整。为了得到一个合适的网格模型，选择一个好的初始网格并使得循环步数最少是非常重要的。现在的ES非常有用，特别是对于缺乏经验的使用者。

现有的ES有几个可能的进一步发展的方向。首先，应该找到能够更合适的，从推荐的类列表中选定单个类的规则。为了更广泛的应用，扩展有限元的范围也很重要。在另一方面，网格设计ES应该整合到整个FEM分析过程中的对关键决定的支持中去，这一点，在Jezernik和Dolšak（1993）中有所提及。

致谢

这项研究得到了斯洛文尼亚共和国科技部的经济支持，在与中东欧国家（PECO92）-ILPNET，合同号码 CIPA3510OCT920044 的科技合作中，欧共体给予了支持。来自ILPNET的研究人员参与了ML应用与有限元网格设计问题的研究，他们为这个目的修改了自己的学习程序。S.Muggleton 的 GOLEM 和 L.De Raedt 的 CLAUDIEN 对于我们的工作来说至关重要。我们要感谢伦敦的皇家科学技术医学学院的 T.K.Hellen 教授，TAM 研究与发展协会的 R.Kogler，来自机械科学系的 Maribor 和 S.Ulage，Maribor 提供了学习实例。特别感谢 Hellen 教授主动承担了对归纳出的规则进行专家评估的任务。

参考文献

Cestnik, B., Kononenko, I. and Bratko, I. (1987). ASSISTANT 86 - A Knowledge Elicitation Tool for Sophisticated Users. In: Bratko, I. and Lavrac, N.

(eds.). Progress in Machine Learning, Sigma Press.

De Raedt, L. and Bruynooghe, M. (1993). A Theory of Clausal Discovery. In: Proceedings of Thirteenth International Joint Conference on Artificial Intelligence, pages 1058-1063,San Mateo, CA, Morgan Kaufmann.

DolSak, B. and Muggleton, S. (1992). The Application of Inductive Logic Programming to Finite Element Mesh Design. In: Inductive Logic Programming, pages 453-472, Academic Press.

Dolsak, B., Jezernik, A. and Bratko, I. (1994). A Knowledge Base for Finitc Element Mesh Design. In: Artificial Intelligence in Engineering 9/94, pages 19-27, Elsevier.

Dolsak, B. (1996). A Contribution to Intelligent Mesh Design for FEM Analyses (in Slovene, with English abstract). Ph.D. Thesis, University of Maribor, Faculty of Mechanical Engineering, Slovenia.

Dzeroski, S. (1991). Handling Noise in Inductive Logic Programming. M.Sc. Thesis,University of Ljubljana, Faculty of Electrical Engineering and Computer Science, Slovenia.Furnkranz, J. (1994a). FOSSIL: A Robust relational Learner. In: Proceedings of the European Conference on Machine Learning, Catania, Italy, Springer-Verlag.

Furnkranz, J. (1994b). Top-down Pruning in Relational Learning. In: Proceedings of the 11th European Conference on Artificial Intelligence, pages 453-457, Amsterdam, The Netherlands.

Hellen, T. K. (1970). BERSAFE: A Computer System for Stress Analysis by Finite Elements In: Conference Stress Analysis Today, Stress Analysis Group of Inst. of Phys. And the Phys. Soc., Guildford, Surrey, UK.

第6章　归纳学习和基于事例的推理在工业机器故障检测方面的应用

Michel Manago 和 Eric Auriol

摘要

“数据是负担，知识是资产”。设备制造商经常会收集大量关于他们必须要支持的设备（内部的或者是通过合作者的）的数据：技术指导文件、故障文件、调试报告、预防维护报告，以及他们的客户或销售商提出的要求。遗憾的是，这些信息通常都没有很好地加以利用，相关的数据库主要在只写模式工作：

1. 没有人使用数据不是因为数据难以访问，而是因为数据没有被使用，所以无人致力于使它便于检索。

2. 数据很不可靠，因为其中有很多错误记录。正是因为这些数据不可信，所以又没有人关心提高数据的精确度。

这种情况很糟糕，因为这些材料往往可以为公司展现一些战略性的信息：错误诊断、维修方法、调试的时间与花费、故障预防等。这些数据的收集与开发利用可以让设备的售后服务和维护有显著的改善。术语“数据挖掘”，是“数据库中的知识发现”过程中的一部分。它是指从数据中提取决定性知识的一组技术。“数据挖掘”覆盖了多种技术，如神经网络、统计分析以及成熟的图形可视化工具。

6.1　简介

归纳学习和基于事例的推理（Case-Based Reasoning，CBR）是对已解决问题的信息的两种不同的利用方法。归纳学习[Qui83]首先为以前的例子建立

一个通用的描述，然后将这个描述应用于新的数据。基于事例的推理则存储以前的例子，然后将根据新的数据与以前例子的关系做出决定。一个事例是指根据其解决方法对已经成功解决问题的描述。当遇到一个新的问题时，CBR回想类似的事例，将以前使用的解决方法加以调整并用来解决当前的问题。归纳学习从所有的事例中提取出一棵决策树来，然后用它去解决问题。归纳学习和 CBR 可以互为补充，将它们结合到一起可以提升其性能[AABM95，AMAWD95]。在不同领域里的应用正飞速增长[AWAMT95]。最有说服力的系统应用于帮助平台领域，特别是针对于复杂设备的故障检测的方面。AcknoSoft公司分别为CFM International公司和Sepro Robotique公司发展的针对波音737的CFM56引擎的故障检测系统和针对机器人轴心位置问题的诊断系统是这一领域里的两个成功例子。

6.2 归纳学习与基于事例的推理

这里使用归纳方法从事例历史记录中建立一棵决策树，然后用这棵决策树来解决问题。归纳学习要求数据有结构化的组织，例如使用带有槽的对象类。一个标准的关系数据库可以很容易地映射到这种对象模型上。这样一来使用者可以自己定义描述事例的字典。比如说，关于控制面板、管道状态、继电器 I/O 状态等差错代码（如表 6.1 所示）。归纳学习同时也要求已经定义了一个目标决策轨迹（如错误诊断）。这个目标轨迹也可以是为了解决问题而改变过的多余部分的列表。给定数据模型、目标类和事例数据，系统会自动地产生一棵错误树（如图 6.1 所示）。归纳学习从而可以从事例历史记录中提取有关的决策知识。

表 6.1 一个事例库样本

诊断	IO 状态	错误代码	电缆状态	管道[5y2]	…
IO 卡	高	153	OK	密集	…
管道系统	低	153	OK	泄漏	…
工具夹子	高	无	?	?	…

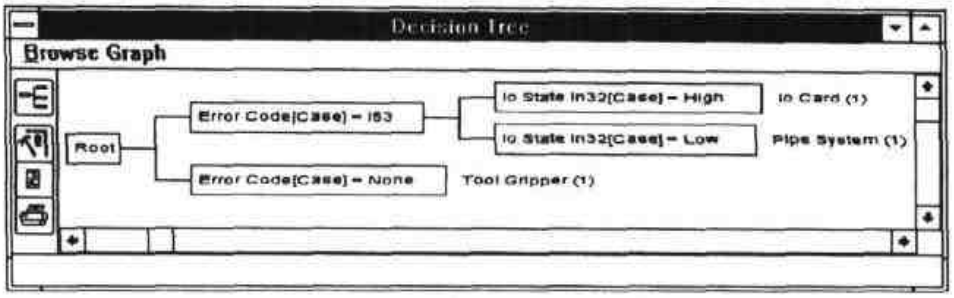


图 6.1　从表 6.1 所示的样本库中产生的决策树

与归纳学习不同，基于事例的推理不需要在解决问题之前产生决策树的结构。一个新的问题是靠寻找类似的已有事例并调整它们的解决方法来解决的。CBR 提供灵活的索引、检索和模糊匹配。对于强调安全性的应用，可以输入额外的可以改变相似度的参数，以使结论更加确定或不确定。CBR 用使那些专家们回想在类似情况下的做法的方式，来获取能够解决问题的专家的帮助。甚至在那些难以理解或是处理事务的规则有很多例外的领域归纳学习中都可以工作得很好。用来判断 CBR 技术是否适合特定的领域的一些特点如下。

- 经验与教科书上的知识一样有价值：CBR 直接使用过去的经验；
- 以往的事例被看成是应予以保留的资产，很明显记住以往的事例是很有用的；
- 专家以实例的方式描述他们的领域。

在开发一个归纳学习或 CBR 的应用的时候，系统的性能是很关键的主题。下面的比较评价是在一台标准的 486DX2/66 PC 上实现的，它表明在数据库中的事例少于 10 000 条时纯粹的 CBR 检索是很快的。检索使用一棵树，一旦这棵树建立起来，检索就是一瞬间的事了。当需要检索大型数据库的时候，CBR 可以和一棵由归纳学习产生的树结合使用，在这里归纳学习被用做一种索引机制。不管是哪种技术都可以利用先前的经验来改善决策制订过程，也可以实现“what-if”分析。

表 6.2　基于事例和归纳方法的性能对比

领域	事例数	参数个数		未知%	决策类别数（秒）	生成树的时间（秒）	纯 CBR 检索时间（秒）
		数值类	符号类				
汽车保险	205	14	11	极少	7	<5	<0.5
信用评估	735	12	12	80%	7	<8	<2
旅行代理	1 470	1	7	没有	93	<7	<2
引擎维护	3 610	7	8	20%	59	<18	<3

6.3 更好地利用经验

在今天，如果缺乏可靠的客户支持，想要卖出工业机器是越来越难了。耐用型设备的“所有权费用”经常远远超出它最初的价值，在买家的观念里，日常维护费用日益成为一个决定性的因素。如果能够用帮助平台软件以及随设备提供的诊断软件提高客户支持的质量，则销售商就能够在竞争里占得先机。CBR 和归纳学习可以帮助他们做到：

- 用帮助平台软件改善售后服务（热线电话）；
- 开发诊断和故障分析决策系统；
- 根据发现的故障有规则地更新故障检测手册；
- 捕捉并利用最具才干的维护专家的经验，而且将专家的意见在职员之间传送，建立企业记忆系统；
- 用经验的反馈来增强可靠性和可维护性。

CBR 帮助工程师将目前的问题与以往的经验联系起来。一个新的问题是由找到类似的事例并根据当前问题调整以前可用的方法来解决的。CBR 优化故障检测需要的测试数目，降低修复费用并且使关键设备的停机时间降到最低。这项技术会直接向产品支持工程师求助并且改变他们的思维模式：“我以前是否遇到过类似的问题，如果有，那么我以前是怎么做的？”。

CBR 技术的一个活跃的市场是面向帮助平台的决策支持软件。一个帮助平台位于一个电话中心，通常在设备制造商的网址上有。当用户所在地的设备出现了失灵之类的问题时，帮助平台可以通过电话帮助用户解决问题。帮助平台职员的一项任务就是为制造商的设备进行故障检测，并且判断用户可以自己解决（在这种情况下用户必须自己更换一组设备上标明的备件）还是必须要派一位属于制造商的现场技术员去解决问题（在这种情况下还要确定需要发送什么备件过去）。

6.4 应用

在 CBR 的市场上各种应用的数量在不断增加。人们已经开发了很多使用 AcknoSoft 公司的 KATE 系统的应用的经验反馈。这些应用有：法国电力公司 EDF 的核能装置安全；瑞士 New Sulzer Diesel 公司的大型船用柴油机故障检

测系统；法国领先的电气设备制造商的生产与产品成本快速评估经验反馈系统；GICEP 电气公司的电路板诊断系统；意大利 Ansaldo Trasporti 的火车维护系统；位于意大利 Schlumberger 和挪威 Norsk Hydro 的石油工业的关键设备质量管理系统；法国 Gas GDF 公司和德国 Gas Ruhrgas 公司的煤气表可靠性分析系统以及法国 Aerospatiale 的航空工业等。在这一章，我们来看两个例子。第一个例子，称为 CASSIOPEE，是帮助波音 737 飞机的 CFM56-3 引擎进行故障检测的。第二个例子，称做 LADI，是为 SEPRO Robotique 公司的售后服务系统，这家公司向全球出口塑料灌注压模机器人。LADI 为那些三轴心机器人检测轴心定位故障。

6.4.1　CFM 56-3 引擎的故障检测

CFM-international 公司为波音和空中客车飞机开发了 CFM56 系列引擎。CFM-international 公司是由通用电器飞行器引擎公司和法国 Snecma 公司联合投资的。CFM-international 的一个目标就是改进引擎的维护技术，以此为它的用户减少引擎的所有权费用。1993 年 8 月，他们实施了 Cassiopée 工程来实现引擎故障检测。工程是从一台 IBM 主机上的一个包含 23000 条事例的可靠性和可维护性数据库开始的。

针对 CFM56-3 飞机引擎技术维护的决策支持系统已经开发出来，它结合使用了归纳和基于事例的推理技术。所有的波音 737 都装备了 CFM56-3 引擎。这个系统帮助 CFM 的工程师，为执行在线故障检测（也就是说当飞机将要飞行时）的航空公司维护人员更快和更好地给出建议。它在包括英国的 British Airways 的好几家航空公司里测试过。整个系统是开发用于：

- 减少引擎的停机时间，避免航班的延迟；
- 将诊断费用降至最低；
- 减少诊断错误；
- 将技术熟练的维护专家的经验记录并整理成文档，以建立相应的记忆并帮助把专家的诀窍传递给初学者。

诊断所需要的时间占停机时间的大约 50%（其余的时间用来修理）。我们的目的是用一件事情的两个方面来划分它。

要从已有的事例中建立一个决策支持系统，必须首先收集事例数据。我们使用的应用数据最初是从前面提到的数据库继承来的。但是很多技术支持

功能所需要的信息不是用数据库的字段格式，而是用文本叙述的形式提供的。由于数据库的大小，而且为了优化系统在日常使用中的性能，我们决定不使用任何的文本检索技术。相反，我们用技术参数和已预处理了的数据来补充数据库事例的“模型”（也就是说，描述一个事例的参数的列表）。数据的预处理是由一位维护工程师完成的，他以每小时 15 个事例的速度审阅所有的事例。

接下来进行了会诊过程。归纳学习产生了一棵新的错误树。这与标准的错误树经常在设计阶段，根据理论上可能产生的错误来产生有所不同，这棵自动生成的错误树是根据实际观测到的错误生成的，并且可以根据新出现的错误进行更新。当浏览这样一棵错误树的时候，操作者要回答与原始问题的症状相关的一些问题（如图 6.2 所示）。在会诊过程的最后，系统按照相对频率给操作者一份可能的解决方案的列表。为了确认所选的解决方案是正确的，从列表中选择那个解决方案时会得到一个程序（在引擎上实施测试）。这样做是为了提高系统的精确度而不用更换正式的维修手册（它是经过授权的程序的附件）。然后支持结论的事例就被检索出来，使用者可以浏览它们，以确定或否定解决方案。

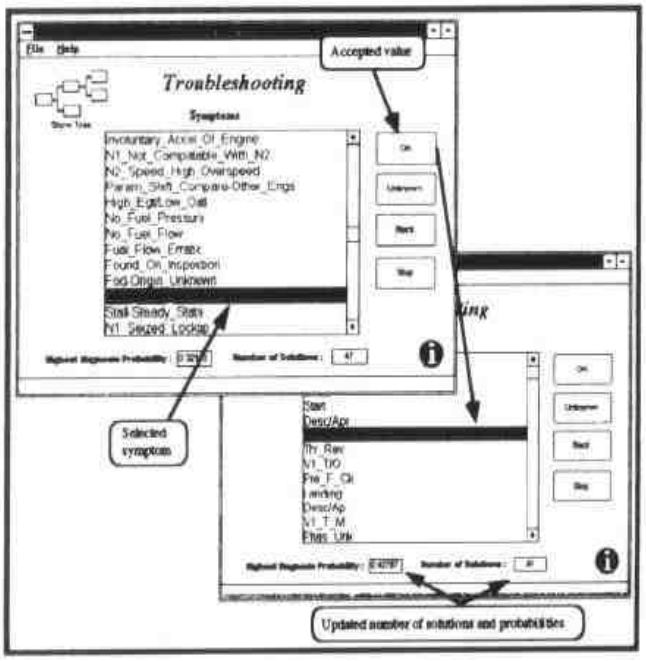


图 6.2　CFM56-3 引擎的故障检测

这个系统完全整合在最终用户的环境里。因此，一些重要的表现就不直接与技术本身相关：为了考虑引擎结构的分解（如图 6.3 所示）而将系统与一个图解部分目录(IPC)连接，显示可靠性与可维护性统计结果的 IPC 的 EXCEL

界面，支持通过 X400 网络在全球范围内用电子邮件收集事例等。

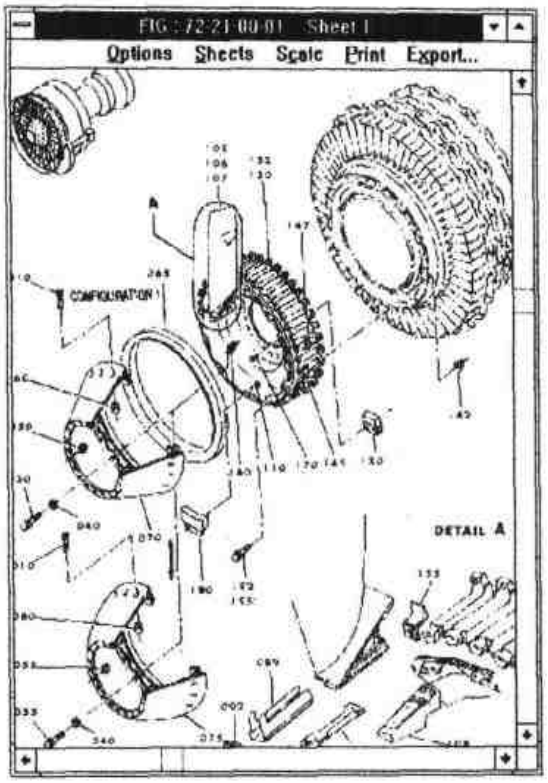


图 6.3　设计有缺陷部件图形化使用的部分专门术语

IPC 拓展了超媒体工具的用途，便于在部件图与它们的术语之间浏览。

6.4.2　机器人轴心的故障检测

SEPRO Robotiquc 公司是法国的一家 SME 会员，它拥有超过 100 名员工，生产压力注塑机器人已经有 10 年之久。SEPRO 已经在全球范围内销售了超过 2 600 台这样的机器人（现在它的产品超过 65%出口到欧洲、亚洲以及美国，在美国 Konair 公司销售这些机器人）。这种机器人是重型机械(如图 6.4 所示)，它自动完成从注塑模型压具中取出塑料部件，然后对它们进行不同的操作（放置一些插件，装载或卸载外围设备等）的过程。

每台机器人都根据客户的需要进行定制，它由三个主要模块组成：机械模块、电机和控制箱。SEPRO Robotique 公司的客户服务工程师和现场技术人员经常基于通用的模块来建立这些机器人故障之间的联系。一个困难是注塑模型机器人有很长的使用年限。十年前安装的机器人现在仍然在使用，而且一直要求 SEPRO 公司提供技术支持。对于 SEPRO 公司制造的现在仍然在客

户那里使用的老型号产品，新的技术支持人员无法获得“内部”的经验，比如说，SEPRO 公司的 Elec 88-SZ 系列产品。因此，SEPRO 公司的客户支持电话中心（同时有四个技术支持人员在那里）在那些对老型号机器人比较有经验的技术人员不值班的时候，经常会难以处理那些设备的故障检测问题。

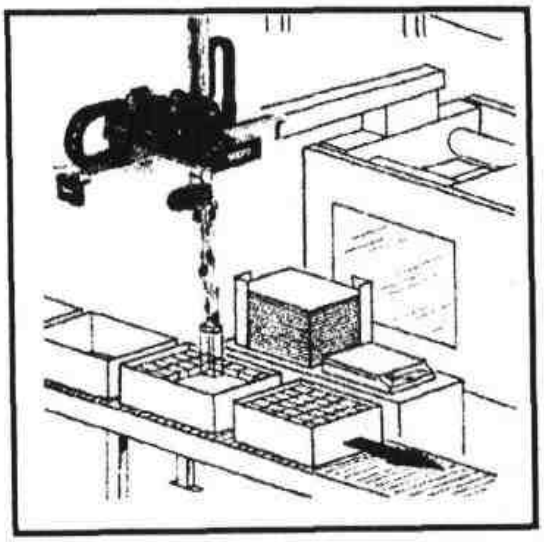


图 6.4　Sepro 机器人对注塑模型操作部件

由于销量的扩大引起了客户服务费用的增加，SEPRO 公司将提高它的售后服务的效率放在了优先发展的位置。在 1995 年 1 月，SEPRO 与 AcknoSoft 公司一起开始安装一套 CBR 帮助平台。这是一套称做 LADI 的，进行轴心定位故障检测的帮助平台系统（如图 6.5 所示）。1995 年 6 月首先交付了一个独立的原型，1996 年春，售后部门在一个五用户的网络上开始使用它。

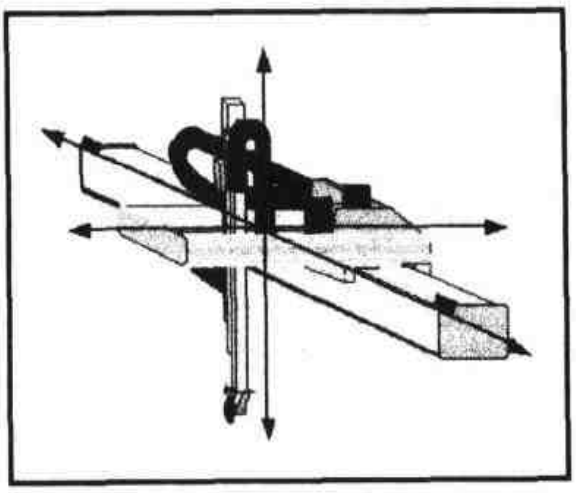


图 6.5　处理轴心定位故障的最复杂的故障检测问题

轴心定位故障占最难解决的问题的20%～30%。这个系统最初包含工程部门可以提供的重建的150个典型事例。记录呼入电话的电话追踪模块可以由四位技术支持人员通过网络共享。事例库也在PC组成的TCP/IP网络上共享。一个内部组织（事例指导委员会）在事例包含到参考事例库中之前进行审查，以控制库中事例的质量。事例数据库以每月10个事例的速度不断丰富起来。指导委员会也对R&D部门更快和更好地提出反馈以改善设计，有较好的作用。

为了适合技术人员的工作方法，这个系统整合使用了CBR和归纳学习技术。当有电话到达售后服务部门时，系统用一棵事先在数据库里建立的归纳树进行预处理，以解决最常见的问题。有75%的电话都可以在这一步得到解决。归纳树的大小有意地被置成比较小（三到六个问题），以使通话时间比较短。如果问题还没有被解决，比如说还有不只一个的诊断还悬而未决，系统就自动地启用“动态归纳”（也就是说，使用与归纳程序一样的评判标准，但是产生的问题列表却不像在树里那样是静态的）产生一个相关问题的列表。系统会自动产生一份包含目前的结论和问题列表的报告，并把它用传真发送到用户那里去。用户在接到电话回复之前要回答那些问题。由于系统是连接在位于一台SUN工作站的插页文档数据库上的，技术支持人员可以访问一个图解部件目录，剪切和粘贴部件的描述，并可以在发给用户的传真里包含一份故障检测报告。当打电话回复用户的时候，系统检索出他以前的那些问题，并根据用户对问题的回答将缺少的信息补充完整。然后系统会在相关的事例中进行最近邻搜索，并把最接近的事例及其相关诊断返回给用户（如图6.6所示）。

这个系统：

- 减少了解决那些对专门技术人员要求很高的问题所花的时间。
- 减少了帮助平台错误诊断的次数。这样的错误会导致向客户运送错误的备件并且由于现场的技术人员不得不等待正确的备件运到，给设备的维修带来了额外的延误。由于SEPRO Robotique公司现在要为更多遥远国家的客户服务，这个问题变得更加重要。
- 选择最具辨别力的测试来识别故障，将诊断过程格式化了。SEPRO公司的培训部门和售后服务部门一样积极地参与了这个项目。
- 由于将专家的经验传递给初学者，这样节省了培训费用。在他们的老

型号设备，如 Elec 88 系列，不再生产而新的技术人员无法获得关于这些设备的经验的情况下，这一点变得越来越重要。随着时间的流逝，新的技术人员将必须借助帮助平台，支持越来越多他们未曾熟悉的设备（技术人员的周转时间大约是 4 年，但是 15 年前安装的机器人到现在还在使用）。这是所有生产长寿命设备的生产商的一个普遍问题。进行电话跟踪并且为每位客户提供到故障记录的直接访问，以提高售后服务的质量并减少给用户电话的次数。

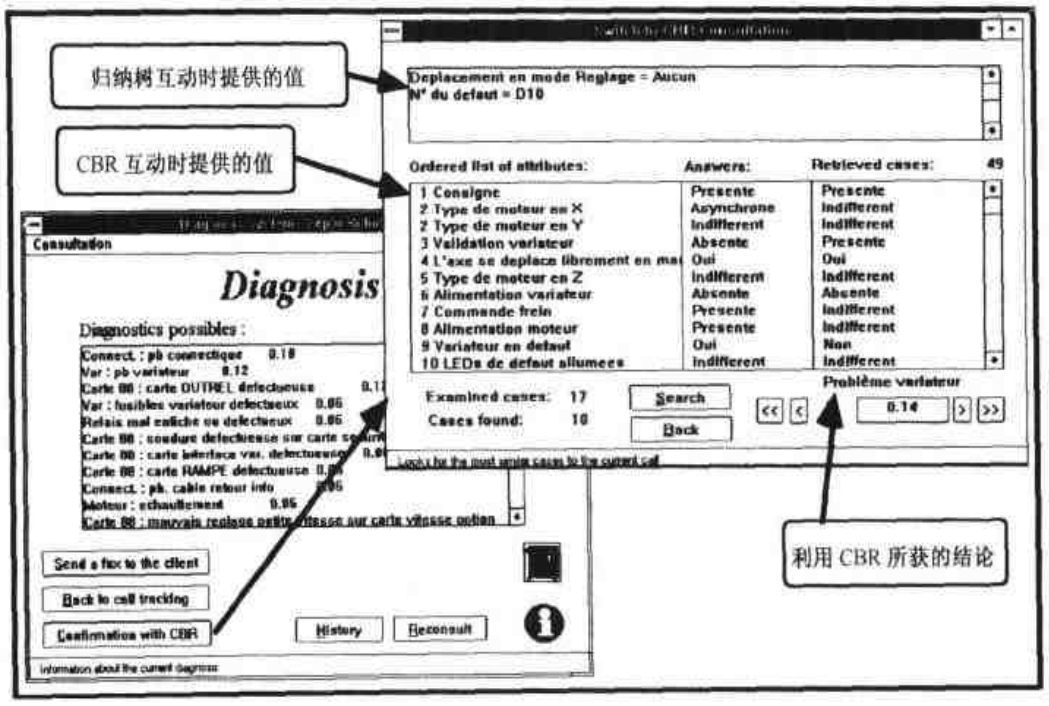


图 6.6　一个机器人轴心故障的诊断

参考文献

[AABM95]　ALTHOFF, K.-D., AURIOL, E., BARLETTA, R. & MANAGO, M. A Review of Industrial Case-Based Reasoning Tools. A. Goodall (ed.), AI Intelligence,Oxford, 1995.

[AWAMT95] AURIOL, E., WESS, S., ALTHOFF, K.-D., MANAGO, M. & TRAPHONER, R."INRECA: A seamlessly integrated system based on inductive inference and case-based reasoning". ICCBR 95, First International Conference on

Case-Based Reasoning, Veloso M. & Aamodt A. (eds.), Springer-Veriag, Heidelberg, 1995.

[AMAWD95] AURIOL, E., MANAGO, M., ALTHOFF, K.-D., WESS, S. & DITTRICH, S."Integrating Induction and Case-Based Reasoning: Methodological Approach and First Evaluations", in Advances in Case Based Reasoning Haton J. P., Keane M. & Manage M. (eds.), Springer-Verlag, Heidelberg,1995.

[Qui83] QUINLAN, J.R. "Learning Efficient Classification Procedures and their Application to Chess End Games", in Machine Learning I: an ArtificialIntelligence Approach, Michalski R. S., Carbonell J. G. & Mitchell T. M. (eds.), Morgan Kaufmann, Redwood City, CA, 1983.

第7章　经验装配序列规划：多策略构造学习方法

Heedong Ko

摘要

装配序列规划是一个涉及广泛的领域，它探索现实世界的制造工业中具有强大实用背景的学习与规划问题。其问题就是发现一系列装配步骤，以便能够通过执行这一序列构造出一个好的装配结构。遗憾的是，系统穷举所有可能序列的方法是不可行的，因为可能的序列数目将随部件数目而呈指数增长，并且几乎没有任何约束或启发知识能够使搜索过程变得可行。此外，最后所产生的与选择的序列将随制造行业以及工种的不同而互不相同。因此，一个熟悉生产环境且有经验的制造工程师能够解决装配序列规划问题。所以，构造一个装配序列规划器的关键就是从记忆中重组反映生产环境的装配情节。这里是利用一个多策略构造学习方法来实现规划的机制，从先前存储的经验装配情节中推断出最终的装配序列。

7.1　前言

在制造业中，利用计算机辅助设计系统（CAD）在电脑中创建并维护一个产品模型，已经变得非常普遍。在生产出最终产品之前产品模型可由许多活动所共享。例如：一个设计者利用一个CAD系统创建了一个机械部件以及数控（NC）机器的指令。计算机辅助制造系统（CAM）利用这些指令在数控机器上直接加工出相应部件。CAD 和 CAM 系统通过计算机辅助过程规划（CAPP）联系在一起，以便推断出在空白材料块上的切割区域，操作的序列，

切割位置和机器情况以及工具的准备情况。针对这种任务的商用 CAPP 系统近来业已推出。

与数控设备加工部件类似，一个用于装配的 CAPP 系统对于制造过程帮助更大，因为大多数产品都是装配而成的。利用 CAPP 系统进行装配，将根据设计者利用 CAD 系统所创建的装配模型，为工业机器人产生装配产品的指令。随着工业机器人在装配过程中的广泛应用，人们对能在装配任务中自动产生机器人操作序列的系统的兴趣越来越大。开发机器人编程语言业已发展到一个更高的抽象层次并更广泛地应用了知识。从连接、操作的角度，到任务的角度，都提出了相应的机器人编程语言。为实现一个装配线的 CAPP 系统，就需要高层次规划问题，即装配序列问题（de Fazio 和 Whitney，1988）。

装配序列问题就是如何产生一个配对操作序列，并使得前面的序列不会妨碍后面序列的操作。为创建这样的序列，规划者在产生一个装配序列时，必须确保前面序列不妨碍后面操作序列的进行。在由于一些部件（子装配集）妨碍了其他部件组装时，规划者必须遵守前面操作序列的约束，也就是那些碍事的部件必须在配对部件装配后方可进行装配。发现这样的先决约束，就需要演习检查部件装配过程是否存在相互矛盾之处。这种演习检查问题就称为 FIND-PATH 问题，它将基础子装配集作为障碍。在三维六自由度空间中解决 FIND-PATH 问题的算法并不是多项式时间的（Donald，1984），因为可移动部件可以作为与基础子装配集类似的装配结构，对于装配集中部件幂集中的每对，都需要解决 FIND-PATH 问题（Ko，1989），这是一个具有指数复杂度的 FIND-PATH 问题。

对于解决这样类似指数复杂度的装配序列问题，一个原始的状态空间搜索方法几乎没有任何用处。我们通过将基础集与可移动子装配集归属到一个装配层次结构中来解决这个复杂问题。装配序列问题的层次结构就是一个强有力的知识架构，它可以使得序列问题变得可控可解。首先，在父结点子装配集部件之前装配当前子装配集中的部件。然后兄弟子装配集可独立装配，也就是，跨相邻兄弟的部件在它们各自的子装配集的装配过程中并不相交。简而言之，一个装配层次提供附加的前提约束以及交叉的本地性，从而减少了 FIND-PATH 问题所要求解的数目。

在一些制造业中，已经形成了一些确定的传统。在汽车工业中，一个发动机的子装配集就是单独设计和在传输子装配中制造的。它们甚至是不同工厂所制造的。这种分离性或许是由发动机在汽车运行中传输所扮演的独特角

色所决定的。这种交互通过凸轮轴来进行。发动机、传输和凸轮轴形成了最终装配层次结构中的兄弟子装配集。因此可根据装配层次结构对一个产品进行功能分解。

遗憾的是，实际并不是这种情况：为了优化设计而有意容许进行交互。一些部件被设计用来承担多种角色来减少部件数目，但却可以维护同样的功能（Ulrich 和 Seering，1988）。这些共享的部件会引起它们相应子装配集的交互。此外，根据层次结构的应用会导致未预料子装配集的交互。例如：在制造一辆汽车时，不会遇到什么大问题，但维护就会有问题，如更改一个滤油器就需要拆卸发动机子装配集，以便可以够到滤油器。

在创建一个新装配层次结构时，没有确定的启发或实际知识来帮助避免所有事先的交互。相反借助长时间在汽车、航空和造船业中的生产经验，形成了被广泛接受的层次结构，它反映了相应业界做法的共同知识。类似，构造一棵有效装配层次结构的实际知识，将会使先前装配情节的规划能够在解决新装配序列问题中找到用武之地，因为它们已经通过了领域使用的测试，而且这些设计中的错误均已被清除了。我们假设设计者的经验和业界的实际做法业已构成了使装配序列规划变得可控的基础知识来源。

因此，构造有效规划器的关键实际就是一个学习问题：如何吸取先前所经历的装配情节，如何将它们应用于解决新装配问题。前者就是一个学习问题而后者则是一个规划问题，它们结合在一起就构成了装配序列规划问题的解决方法。我们将这种规划情况称为经验装配规划方法，强调其在构造规划时对学习的依赖。

这里的学习情况就是一个多概念学习，其中记忆中包含已经装配的结构和序列概念。此外，每个成功或失败的装配情节将不断地被记录下来，从而构成一个闭环学习。本章将要解释多策略构造学习（MCL）（Michalski，1993）系统是如何利用一个经验装配规划系统（它被称为 NOMAD，NOrmative Mechanical Assembling Device）来解决这个学习问题的。在介绍学习内容之前，将要介绍是如何应用 NOMAD 解决一个装配问题的，如何表示所记录的一个装配规划经历以及装配过程的。

7.2 NOMAD 中的表示与规划

一个 CAD 系统的装配结构包括部件与配对条件。CAD 系统中的一个部件实体模型包括：在部件内部与外部边界上的界面。而一个界面由边包围，

一条边由一对顶点间的连线构成。此外，这些受限的实体与一些几何实体相关，如一个平面界面与一个平面相关。部件模型中，通过两个部件之间的配对条件，在各自受的限实体中：一个“对着”配对条件定义，一个部件平面与另一个部件平面相接触（它们各自平面相对且共面）；一个“排列”配对条件定义，一个部件圆柱与另一个配对部件的圆柱共“圆柱”。一个装配结构可用一个标记图来表示，被称为配对图（Ko 和 Lee，1987），其中结点代表部件和配对条件的连接。

图 7.1 描述了一个包含三个部件的 Bell Head 装配结构，这三个部件是 Bell Head，Pin 和 Ring，其中配对图如图 7.2（a）。本章的示例均是二维的，其中部件的边界元素均是直线和顶点，虽然 NOMAD 拥有三维实体模型内核用于装配建模。全局框架是一个环境中部件的相对坐标系统：z 轴垂直向上。每个部件拥有自己本地的参考框架，被称为一个基础框架。图 7.1 描述了一个 Pin 的基础框架。

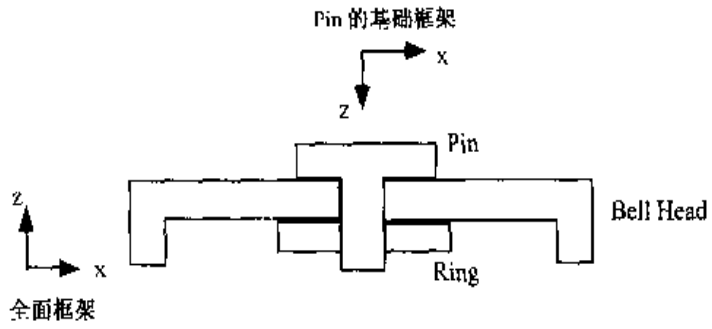


图 7.1　Bell Head 装配线

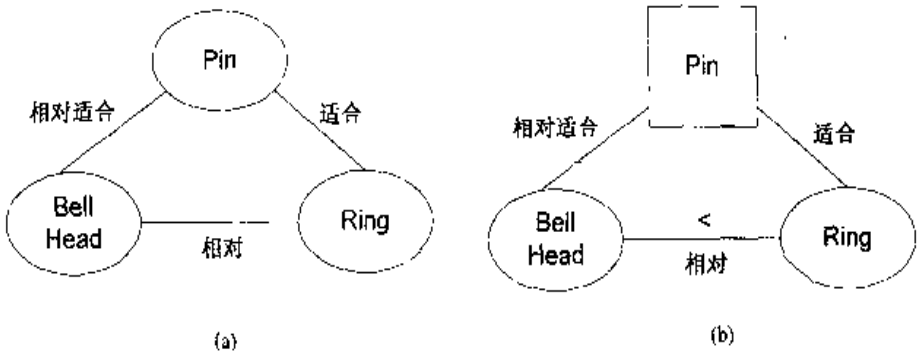


图 7.2　具有两个装配情节的配对

装配结构通过一个配对图来加以表示。如图 7.2（a）所示。为装配 Bell Head，利用配对图作为规划者的目标定义以便确定装配步骤的序列。通常，

每个装配步骤可能包含抓住、定位、配对和各种发给机器人的传感器命令。这就使得对规划结构进行比较变得较为困难。这里，每个装配步骤可能满足配对图中的一个或多个配对条件，假设每个配对操作包含定位、抓住和其他机器人操作，作为其在操作处理过程中的子操作（Mostow，1983）。

除了简化每个装配步骤外，规划结构必须规范化为存储中的基本块以便保存和进行多规划结构间的比较，即作为一个经验学习单元的基本步骤。这些基本块被称为**装配情节**（Assembly Episode）。装配情节就是包含一个序列和一个装配结构的规划段。与装配结构相应，我们定义一个组为一个规范的块，并据此创建一个内存结构。该组就是一组部件，其中有一个特别的部件，叫做基础部件，其他则是附属部件。在进行组中部件装配时，基础部件保持固定，而移动其他附属部件与基础部件进行配对。

在一个组的情况下，可以将一个装配序列表示为附属部件的一个序列。因此，装配情节就是一个组和附属部件一个序列的组合。图 7.2（b）描述了根据图 7.2（a）配对图所进行规划而得到的一个装配情节，其中 Pin 就是基础部件（方形结点）。Bell Head 和 Ring 就是附属部件，并构成装配到 Pin 的序列，这里“<”表示 Bell Head 和 Ring 之间的装配关系。

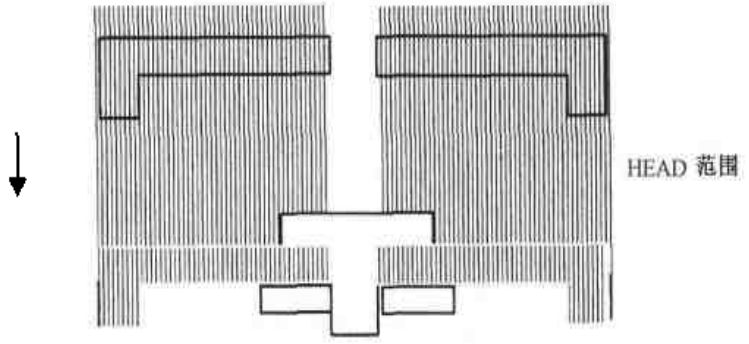
最终装配规划包含一个或多个装配情节。执行最终装配规划的结果分为存储中的装配情节正例或反例应用。这些样本将作为一个学习单元的输入，以便能够利用实例到模型的归纳方法产生一个新模式（Michalski，Ko 和 Chen，1987）。一个模式就是对一个领域对象的归纳描述，如：在装配领域，一个部件模式就是对于一组部件的归纳描述，一个组模式就是对于一系列组的归纳描述。组模式包含一个基础部件模式和附属部件模式。附属部件模式之间的顺序关系通过一个或多个前提约束来加以描述。它们利用部分对整体归纳方法存为前提模式（Michalski，Ko 和 Chen，1987）。因为有多种方式装配一组中的附属部件，因此组模式可能具有多个前提模式。总之，NOMAD 的存储结构包含两类相互关联的领域概念：前提和组模式。

在这些存储的领域概念中，规划过程经过三个阶段：回想、组合和**事后分析**。回想过程将配对图分解为装配情节。一个装配情节通过以下两步加以识别：首先通过一个组模式，其中，基础与附属部件模式是通过配对图中的部件来识别的；然后通过一个前提模式，其中，把前面步骤中组模式所识别的基础部件与附属部件间的装配步骤排列成一个序列。这些序列段将作为最

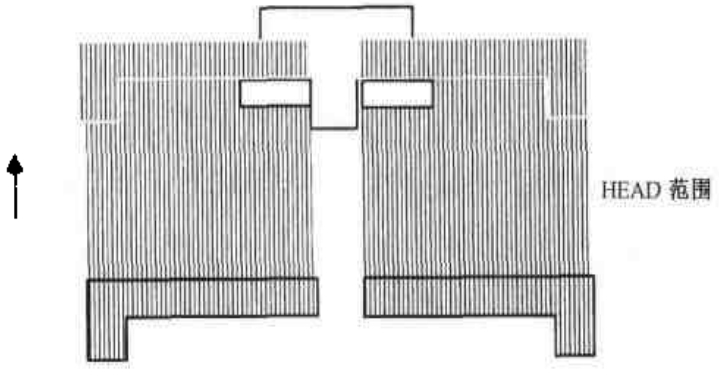
终装配序列要考虑的部分序列段的候选。

在回想过程步骤之后，规划者将候选装配情节结合到层次结构中，并保证其一致性和完全性。如果层次结构包含装配步骤涉及配对图中所有部件，层次结构是完全的；如果由候选序列段所提出的前提关系互不矛盾，层次结构是一致的。装配层次结构为装配规划者确定与装配序列描述相应的抽象结构。这就意味着，为一个层次结构根获得装配序列时，仅考虑其直接孩子的子装配集以满足这种假设：一个子装配集的装配步骤或许不能与其兄弟子装配集的装配步骤相交叉（子装配集独立定律，SIL）。更进一步，一个装配集中的装配步骤必须在其父的子装配集装配步骤之前完成装配（子装配前提定律，SPL）。由 SPL 和 SIL 提出的约束以及装配情节将从存储中调出，用于事后分析中的权值赋予过程。

在构造完一个装配层次结构后，就可从序列段的前提关联以及层次结构的 SPL 和 SIL 约束中，获得最终的装配序列。如果在到达步骤所期望的配置中无法避免冲突，序列中的一个装配序列可能会失败。例如：对于 Bell Head，在 Ring 已经安装后，就会由于空间交叉，而无法到达配对位置，如图 7.3（a）



(a) Bell Head 与 Pin 交叉



(b) Bell Head 与 Ring 交叉

图 7.3　空间交叉

和（b）所示，描述了不同的移动方向。因此，应用这个装配情节就是一个反例，但是规划者在图 7.2（b）所示的情节就可以获得成功。

从规划经验中学习涉及记忆中的多个概念，学习者对记忆中导致成功或失败的装配情节的概念进行事后分析。因此规划结构必须是一个关联结构（de Kleer，1986），其中在断言数据库中的每个前提关系均由一个装配情节，SPL 与 SIL 约束或由它们的组合来支持。

因为前提约束和组模式均被用于创建装配情节，所以事后分析必须识别起作用的两类概念。在前面一个例子中，在装配情节定位失败或成功时，一个前提模式被识别为起作用的概念。另一方面，在层次结构中的子装配集违反 SPL 或 SIL 约束时，或两个都违反时，利用组所构造的子装配集就被识别为导致的原因。赋权值过程识别那个记忆中的概念负责输出，并由此学习它。赋权值问题则是通过最终装配序列的前提，装配关系之间的依赖支持结构，以及通过实例化记忆中模式而引入的假设来解决的。下一节将介绍多策略构造学习，其作为先前概念学习的一个扩展，以对付 NOMAD 中多概念学习情况。

7.3 多策略构造学习

从规划经验中进行学习是一种多概念学习，因为规划经验可能涵盖多个要学习的概念。对于装配序列规划而言，需要学习两类概念：组和前提模式。然后学习者需要确定哪一个概念要学习，以及确定相应的训练样本。先前的学习系统大多都是单概念增量学习系统。例如，据报道，INDUCE（Larson，1977）给定多个向西或向东火车样本，作为单个概念的两个类别（火车行驶方向）。在这种情况下，对要学习概念的相应记忆进行索引，作用不大，就像这是由 INDUCE 中样本类别所确定的。

在多概念学习情况下，系统应该确定哪个概念或记忆中的概念要学习。事后分析则利用上一节装配序列中的依赖支持结构，识别哪个概念（组和前提模式）决定成败，从而需要学习。所以规划经验分解为装配情节，以作为记忆中组和前提模式的应用。在前面一个例子中，用于构造图 7.3 所示的装配情节的前提模式被识别为要学习的概念及由装配情节所提出的前提约束。该装配情节被作为训练示例，用于完善所选模式的概念描述。多概念学习情况中的这种增量学习就称为**闭环**（Closed-loop）学习，其中要学习的概念先前就

确定了，并不断完善更新。

开始时，记忆中概念描述几乎没有背景知识（模式），它通过对实验描述的归纳抽象而产生。在经验归纳学习中，学习者对观察到的样本进行抽象泛化，以形成它们相应一致与完全的描述。在机器学习中，构造经验抽象的程序仅利用描述性概念（从那些初始观察描述中选择出来的）。这种表面归纳被称为**选择性归纳**（Selective Induction）。在构造归纳中（Michalski，1983），学习者利用领域相关以及领域无关的背景知识对输入具有新描述符的观察进行分析，希望搜索出领域概念的倾向性描述空间中的归纳假设。因此，学习者在部件中，分析具有相对空间关系，如“高于”或“低于”的训练样本。

最终，系统会评估所构造出的知识，以便确定是否应在记忆中保留（这些知识）。利用关联结构，学习的情节在保存经验之前，被结合形成“更”有用的知识。这一步与构造归纳类似，但它被称为**演绎重构**（Deductive Restructuring）以突出本步骤是产生新“有趣”示例的一步，其产生方法就是将多个训练样本结合起来而不是对每个训练样本插入新的描述符。为从规划经验中进行学习，必须结合三个学习步骤。我们将这种集成学习机制称为多策略构造学习（MCL）：

$$\text{MCL} = \text{构造归纳} + \text{闭环} + \text{演绎重构}$$

下一节将通过一个学习情况来阐述本节归纳出的概念。

7.4　NOMAD 的学习场景

NOMAD 多策略构造学习方法由以下三步组成：

（1）归纳性地产生候选组和前提模式；

（2）通过应用维护每个候选模式的可信度；

（3）在新情况中应用有希望的模式。

假设一个用户指定如图 7.1 所示的 Bell Head 装配结构，而 NOMAD 对它并不了解。全局框架就是一个环境中所有部件的相对坐标系统：z 轴垂直向上。开始接收任务时，NOMAD 对 Bell Head 装配一无所知。因此用户可能会定义如图 7.2（a）所示的组：Pin 是一个基础部件，Ring 和 Bell Head 为附属部件。然后系统可能随机有选择性地对其他规划进行实验，因为它没有任何知识来指导规划进程。

在如图 7.2（b）所示的成功规划情节中，利用一个构造归纳规则，根据相对两个相关框架所确定的附属部件的位置，产生以下描述符。这两个框架为：全局框架（dloc-g）和基础部件的基础框架，Pin（dloc-b）。两个描述符为如下所述

- 相对全局框架：[dloc-g(Ring,Bell Head)<0]，Bell Head 的位置高于 Ring；
- 相对基础部件的基础框架，Pin：[dloc-b(Ring,Bell Head)>0]，Ring 位置高于 Bell Head。

此外，基础部件 Pin 是一个 T 型，而附属部件 Bell Head 和 Ring 是环状。

之后，NOMAD 利用“消除一个合取项”的归纳抽象原则（Michalski，1983），从规划情节中产生备用模式。在备用模式抽象中，这里针对如图 7.4 和 7.5 所示的组模式，分别给出了两个候选前提模式，P1 和 P2。如图 7.4 所示，在全局相对框架中（dloc-g），基础部件模式，$p1，与附属部件模式$p2 和$p3 排列在一起。与装配情节类似，方框代表基础部件模式。

图 7.5 描述了基础部件模式$p1 与附属模式$p2 和$p3 排列在一起，而$p2 在$p3 前装配，因为相对$p1，$p2 比$p3 要低。此外图 7.4 和 7.5 中的部件模式保存了部件的形状，$p1 是针状的，而$p2 和$p3 是环状。

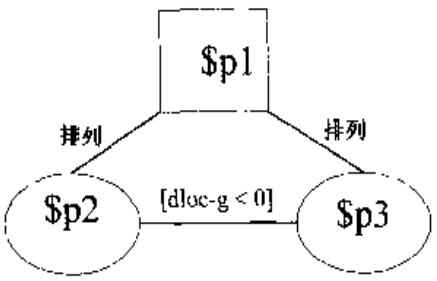


图 7.4　候选前提模式 P1

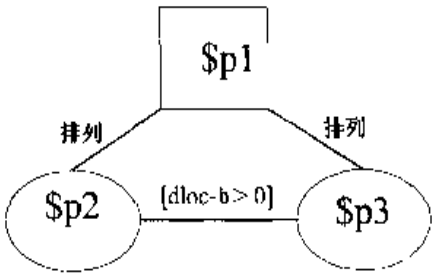


图 7.5　候选前提模式 P2

根据两个候选前提模式，P1 和 P2 及它们记忆中的组模式，系统开始装配一个支架，并将其作为新问题来加以解决，如图 7.6 所示。现在 NOMAD 应用候选模式来解决这个新问题。该新问题与图 7.3 所示的 Bell Head 装配问题非常类似：螺钉是 Pin 形状，螺母和支架是环状。根据 P1 和 P2 的组模式，NOMAD 将螺钉识别为基础部件，而将螺母和支架作为附属部件。

利用 P1，由于螺母比支架高，如图 7.7 所示，所以系统首先计划螺母要连接到螺钉上。然而，计划（或假设）由于与图 7.3 所示的类似原因而失败：如果螺母首先进行连接，则由于带有螺母和螺钉的两个支架空间相互干扰，

使得支架无法进行装配。这个失败的计划情节降低了 P1 的可信度。

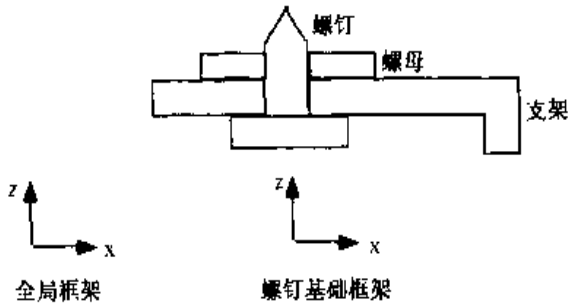


图 7.6　支架装配

另一个利用 P2 的可选择的规划则是：首先把支架连接到螺钉，如图 7.8 所示，因为相对螺钉，支架比螺母要低。这个规划情节成功了并由此加强了 P2 的可信度。作为最终结果，P2 要比 P1 在用于未来规划情况时，具有更高的可信度。

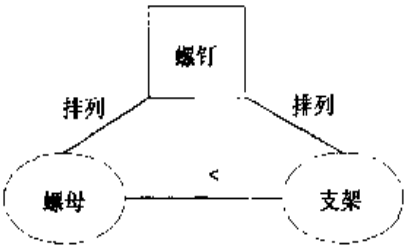


图 7.7　候选模式 P1 的失败应用

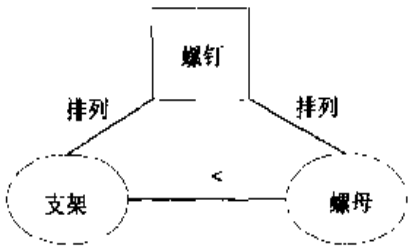


图 7.8　候选模式 P2 的成功应用

考虑一个螺杆装配，如图 7.9 所示。利用组模式，系统将螺杆识别为一个基础部件，因为它是针状的。垫圈 1 和垫圈 2 为组中的附属部件，因为它们是环状的。在组模式的前提模式中，根据过去经验，NOMAD 先于 P1 应用 P2。利用 P2，垫圈 1 在垫圈 2 之前装配，因为相对螺杆，垫圈 2 高于垫圈 1。这个规划情节如图 7.10 所示。

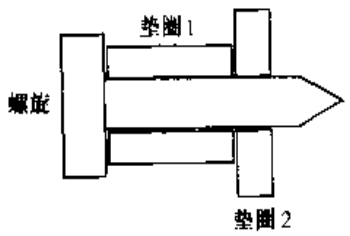


图 7.9　螺杆装配

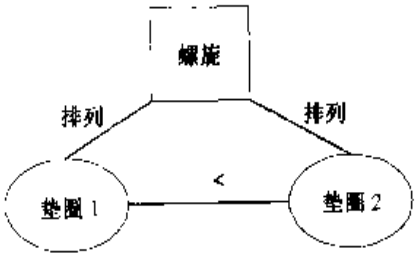


图 7.10　利用 P2 的螺杆装配情节

这个学习情况表明一个装配中如何识别基础部件和附属部件，以及如何通过在一个新环境中进行积极的实验来维护或删除记忆中前提关系的候选描述。学习情景描述了一个闭环，以及NOMAD 所使用的 MCL 构造归纳内容。根据经验证实的模式 P2，系统将面对一个新的更复杂的螺杆装配，如图 7.11 所示。

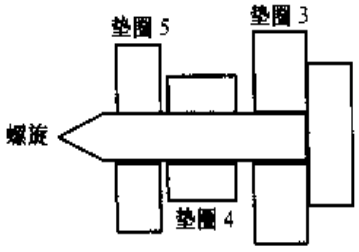


图 7.11　带有三个垫圈的螺杆装配

借助 P2 组模式，螺杆为基础部件，因为它是针状的。所有垫圈 3，4 和 5 为环状，那么，在垫圈和附属模式之间，共有六种可能的连接，$p2 和$p3。但是，在利用 P2 的前提下，仅有三种是可能的，其他的将被删除。它们如图 7.12 所示。情节 1 指示在垫圈 4 之前装配垫圈 3，因为相对螺杆，垫圈 4 比垫圈 3 高。情节 2 指示在垫圈 5 之前装配垫圈 3，因为相对螺杆，垫圈 5 比垫圈 3 高。情节 3 也如此，在垫圈 5 之前装配垫圈 4。这些前提约束都是正确的，相互之间并无冲突，但它们本身在如图 7.11 所示的装配结构中是都不完全的。

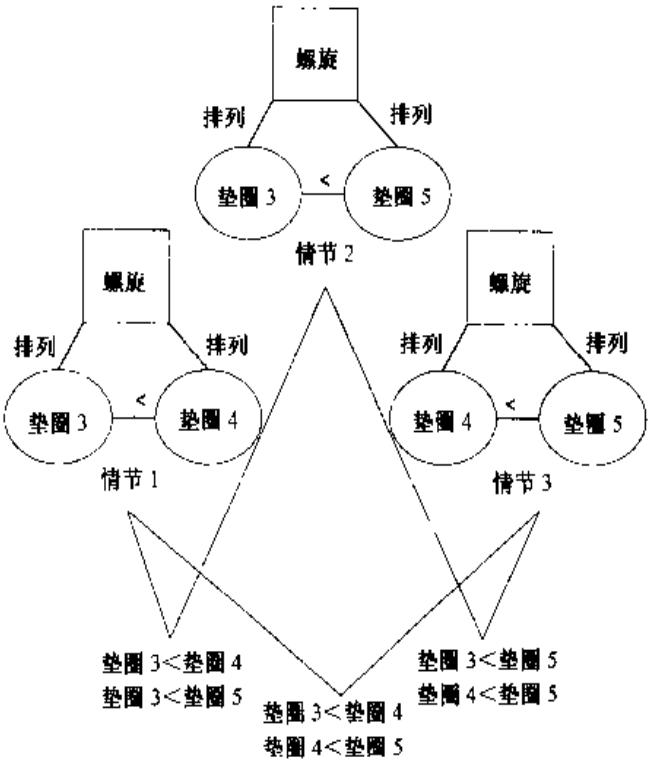


图 7.12　候选模式 P2 下的三个装配情节

在创建一个安全装配序列规划时，没有必要将图 7.12 中所示的三个情节都组合起来。情节 1 和情节 3 足够用以预测情节 2 的前提约束，因此情节 2 是多余的。其他候选装配情节则利用一个瓦解运算符对它们进行合并，这是 MCL 中的一个演绎重建归纳步骤的实例。通过情节 1 和情节 2 的瓦解，系统将能预测垫圈 3 需要最先装配，但垫圈 4 和垫圈 5 之间没有前提。通过情节 2 和情节 3 的瓦解，系统将能预测垫圈 5 需要最后装配，但垫圈 3 和垫圈 4 之间没有前提。通过情节 1 和情节 3 的瓦解，系统将能预测垫圈 3，4 和 5 序列装配。图 7.13 描述的瓦解结构将作为一个新的训练结构。NOMAD 利用瓦解的结构来应用部分-整体抽象方法到单个情节中。

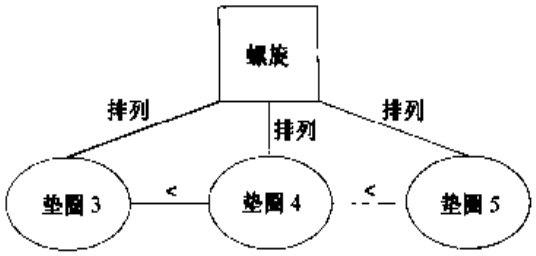


图 7.13　瓦解装配情节

7.5　与先前研究进行比较

先前大多数规划研究将启发式状态空间搜索作为其基本问题来求解机制：规划操作作为状态转换运算符，规划结构作为一些系列运算符，完成从初始状态到最终状态的转换。这个转换过程结果变为一个指数型搜索过程，从而需要引入一个学习模块进行事后分析，以减少从初始状态到达目标状态的搜索时间。在装配任务中，状态代表部分被装配的部件，目标状态代表被完整装配的部件，而运算符就是装配步骤，包括抓住、移动、堆砌等。这样学习部件就能够通过以下两种方式减少搜索过程：

（1）借助一个强有力的启发知识，帮助避免不属于最终装配过程的装配步骤。

（2）借助将规划情节记忆为一个宏操作，以便搜索过程能够减少单个运算符的组合爆炸数目。

在前一个情况下，通常对如何构造一个用于普通装配任务的强有力启发

知识知之甚少。在第二个方法中，利用对规划情节进行演绎抽象，可以产生一个宏装配模式。这是基于解释的学习（EBL）（Mitchell，Keller 和 Kedar-Cabelli，1986；De Jong 和 Mooney，1986）。对一个规划情节中的常数进行最小化抽象，以便操作模式与背景知识一起，在抽象后逻辑蕴含宏操作模式。抽象是演绎性的，因此学习过程被称为**分析性的宏操作模式学习**。遗憾的是，这些编译后的宏操作与基本操作一起被应用在搜索过程中，它们可能会产生额外的搜索负担。因此，EBL 宏操作学习过于保守以致无法产生一种知识的新形式，该知识可使装配序列问题变得可控。

经验装配规划可通过归纳抽象装配情节，从中获得基础的新知识。这里的主要学习任务就是发现一个覆盖某概念的所有正例，而无任何反例的可行抽象。这里，一个基本学习机制就是将对象作为一个概念实例或非概念实例，以便揭示这些实例中共同的相似性和差异性。因此，学习机制就需要用于对两个实例做比较的有效手段。

在利用属性描述实例时，一个实例对象的内部结构被抽象出来，不包含任何部件对象：它由一个单一对象表示，自己本身命名。然后就可以与下一个对象进行比较，由于总是有对象可以比较，因此这个问题比较简单。有许多基于属性训练实例进行学习的方法（Michalski，1973；Quinlan，1983）。遗憾的是，一个装配规划结构是一个高度结构化的对象，它涉及空间与时间的关系。因此，在这里没有哪个学习方法可以被直接应用。

结构化对象的比较过程也是比较复杂的，其中一个结构化对象包括部件对象与它们间的关系。在比较两个结构对象时，来自于结构对象中的一个部件对象对应于另一个来自其他结构对象的部件对象。这个问题被称为对**象对应**（Object-correspondence）问题。当两个实例中的一个有 M 个不同对象时，在它们之间就有 M 的阶乘个不同对象对应组合，假设是“一对一”对应（否则复杂度会更大）。因此，比较两个结构化对象本质是一个爆炸搜索过程。为克服这个困难，需要比较许多实例以发现内在规律。

Winston 的初始工作（Winston，1970）就提出了一个与领域无关的方法，用于从可视化场景中学习获得结构概念。它根据观察到的相似性和差异性，将场景中对象分为若干组，这是归属过程。将一个结构化对象的每个组与另一个作为一个结构概念正例或反例结构化对象进行比较，这是比较过程。归属过程后来由 CLUSTER（Stepp，1984）进行了改进，提出了一些有用的相似

性度量方法用于聚类，并提出了一个启发控制算法用于形成层次聚类。比较过程后来由 INDUCE（Larson，1977）进行了改进。它描述了一个明确的两阶段方法，在进行对象比较以发现对象对应之前，完成两个对象之间结构关系（连接）匹配的比较。每个结构对象由一个标记图来表示，并利用一个网络匹配算法来解决多形态子图问题。虽然装配结构和规划均是结构化对象，但是从中学习到的、结构概念，与领域无关的方法，仍无法直接应用于从先前装配规划经验中学习的过程。

上述领域无关的结构概念学习系统，按照与处理其他结构的序列的方式，处理时间相关结构对象。SPARC（Sequential PAttern ReCognition）（Michalski，Ko 和 Chen，1987）中研究了序列模式的学习问题。这个独特的学习问题需要部分到整体的抽象，与先前结构概念学习系统中的、从实例到类别的抽象相反，这里，序列中每个元素将某个潜在过程的说明作为一个实例，学习任务就是归纳出相应规则，这些规则应能够解释序列，并能作为一个符号预测过程来预测未来序列的连续情况。所以，学习机制集中在实例间的序列关系方面。依赖序列关系类型，SPARC 具有三种规则模型（分解、周期和析取），其中每个规则模型均是寻找不同类型的序列模式。规则模型与 NOMAD 中的归属与前提模式类似。

结构概念学习系统应用于空间结构识别，如装配问题、Winston 的 Arch 问题；序列模式学习系统应用于符号过程预测。NOMAD 从空间结构的序列中进行学习（装配结构），因为装配序列规划领域涉及空间与时间结构领域分析。

7.6　结束语

经验装配规划在装配规划上下文中利用了一种归纳学习机制。从规划经验中学习，必须确定要学习哪个概念，以及来自规划经验中相应的训练实例。NOMAD 将规划经验分解为组和前提关联，以此作为来自记忆中组和前提模式的规划情节。利用规划经验，可以不断地更新与修改这些模式的可信度。在多概念学习环境下，这种增量学习是一种闭环学习。此外，在产生最终装配序列时，学习模块环引入了情节的演绎重构。演绎重构产生更多的、没有被当前记忆中模式所覆盖的、“有趣”的训练实例，以便于学习者能够将工作

假设的重点转移到一个有新实例产生的新假设上。此外的构造归纳步骤中，NOMAD 这三个学习步骤通过多策略构造学习模块被组合起来。

为使当前的实现方法成为一个工业实际应用，NOMAD 必须保存大量的有效组和前提模式，这些模式概述总结了记忆中的经验规划实例。作为一种为学习系统提供大量规划实例的有效方法，开发出来了虚拟装配仿真系统。这里，教师提供了一种装配描述以及利用虚拟现实（VR）界面来表示装配过程的处理，一种可视化机器人编程环境。然后，教师或许提供大量装配情节，这些内容构成了应用多概念学习场景的基础，这个场景是本章介绍的更为实用的工业状况。当虚拟装配系统完成后，它将成为基于 CAD 智能装配建模和仿真系统的一部分，这些系统可应用于当前产品开发与生产的环境。

致谢

这里作者将深深地感谢我的论文导师 Ryszard S. Michalski 教授所给予的指导。此外，作者还将特别感谢 Urbana-Champaign 的 Illinois 大学机器学习小组的成员，作者在那里读博期间，他们提供了充满智慧与有益的反馈。

参考文献

De Fazio, T.L. and Whitney, D.E. (1988). Simplified Generation of All Mechanical Assembly Sequences. IEEE Journal of Robotics and Automation.

DeJong, G. and Mooney, R. (1986). Explanation-Based Learning: An Alternative View.Machine Learning Journal, vol. 2.

De Klecr, J. (1986). An Assumption-Based Truth Maintenance System. Artificial Intelligence, vol. 28, no. 1.

Donald, B.R. (1984). Motion Planning with Six Degrees of Freedom. TR-91, Massachusetts Institute of Science and Technology, Artificial Intelligence Laboratory. Ko, H. and Lee, K. (1987). Automatic Assembly Sequence Generation. Computer Aided Design vol. 19, no. 1, pp.3-10.

Ko, H. (1989). Empirical Assembly Planning: A Learning Approach. PhD Thesis,

Department of Computer Science, Univerisity of Illinois, Urbana. Larson, J.B.

(1977). Inductive Inference in Variable-valued Predicate Logic System VL21: A Methodology and Computer Implementation. Ph.D. Thesis, Report No. 869, Department of Computer Science, Univerisity of Illinois, Urbana.Michalski, R.S. (1973). Using Classifcation Rules using Variable-valued Logic System VL1. Proceedings of the Third International Joint Conference on Artificial Intelligence, Stanford.

Michalski, R. S. (1983). Theory and Methodology of Inductive Learning. Chapter in R.S. Michalski, J.G. Carbonell, T. M. Mitchell (Eds.), Machine Learning: An Artificial Intelligence Approach, Tioga Publishing.

Michalski, R. S., Ko, H. and Chen, K. (1987). Qualitative Prediction: The SPARC/G Methodology for Describing and Predicting Discrete Processes. Chapter in P. Dufour and A. Van Lamsweede (eds.). Expert Systems, Academic Press, pp. 125-158.

Michalski, R. S. (1993). Toward a unified theory of learning: Multistrategy task-adaptive learning. Readings in Knowledge Acquisition and Learning edited by B.G.Buchanan & D. Wilkins, Morgan Kaufmann.

Mitchell, T. M., Keller, T. and Kedar-Cabelli, S. (1986). Explanation-Based Generalization: A Unifying View. Machine Learning Journal, vol. 1.

Mostow, J. (1983). Machine Transformation of Advice into a Heuristic Search Procedure. Chapter in R. S. Michalski, J. G. Carbonell and T. M. Mitchell (eds.),Machine Learning: An Artificial Intelligence Approach, Vol. I, pp. 367-403, Morgan Kaufmann, Los Altos, CA.

Quinlan, J. R. (1979). Discovering Rules from Large Collections of Examples: A Case Study. Chapter in D. Michie (Ed.), Expert Systems in the Microelectronic Age,Edinburgh University Press, Edinburgh.

Stepp, R. E. (1984). Conjunctive Conceptual Clustering: A Methodology and Experimentation. Ph.D. Thesis, UIUCDCS-R-84-1189, Department of Computer Science, University of Illinois, Urbana.

Ulrich, K.T. and Seering W.P. (1988). Function Sharing in Mechanical Design. Draft,Massachusettes Institute of Technology, Artificial Intelligence Laboratory.

Winston, P. H. (1970). Learning Structural Descriptions from Examples. Ph.D. Thesis,MAC TR-76, Massachusetts Institute of Technology.

第8章 归纳学习设计入门：关于防摩擦轴承系统的设计方法和实例研究

Wojciech Moczulski

摘要

本章讨论关于把入门规则学习应用到工程设计上的问题。具体的研究实例涉及到给定设计标准时，对装置以及下属装置的设计完成过程。我们描述了基于 AQ-15 规则学习系统上的设计规则学习方法，并且通过一个设计防摩擦轴承系统的例子进一步阐明了这种方法。在这个应用领域，一个重要的方面是：在许多设计中，存在不只一种解决方案。因此，学习系统必须有能力建立起在逻辑上相交的描述。然而，绝大多数已存在的学习系统并没有达到这一点。另一个要求是学习的描述应该易于专家理解和解释。令人高兴的是，这里所采用的 AQ15c 学习程序对这两方面的要求都能满足。初步的结果是令人鼓舞的。这种方法似乎有助于自动化，对设计者的日常设计工作也很有帮助。

关键词：机器学习、机器学习设计、设计知识获取、入门规则学习、AQ 学习系统、防摩擦轴承系统。

8.1 导论

设计是一个创造性的过程，包括通过抽象的概念进行推理，得出满足所给定的规则和约束的解。设计的输出过程包括简捷的预设计和设计思想。设计思想包含对问题的解决方案的初步描述，这是通过应用设计操作符对输入数据（需求描述）进行处理而产生的。这些操作符代表不同的设计方法和技术。设计中所需要的知识和技能可以通过设计者在技术学习和实际工作中获得。

如果考虑整个设计过程，我们会发现这常常需要大量的创造性工作。这种工作是如此的复杂，以至于就现在的技术发展水平而言，基于知识的系统对它毫无助益。但是，一旦设计者建立了一个初步的设计构架，他就很容易知道哪些基于知识或者机器学习技术的子任务是有帮助的。许多这样的设计任务（或子任务）需要日常性而不是创造性的工作。既然如此，它们被部分或全部地自动化就相对容易了。这些帮助能显著降低设计者的整个工作量。要决定哪些任务可以接受自动化，一件很重要的事情是去认识能够被表示成规则形式的设计知识。只要这件事情做了，就能够应用机器学习方法了。

本章给出了一种用归纳规则学习系统 AQ15c 学习设计规则的方法（Wnek 等人，1995）。实例研究涉及到装置的设计以及相对于规则和输入数据的子装置的完成。对于具体的问题，我们运用了一种产生规则然后通过防摩擦轴承系统进行选择的学习方法。第一步，由一个有经验的设计者定义表征设计对象的属性，然后，他选择一些属性把设计对象描述出来。这些对象就是学习程序的输入。我们先讨论一种表示部件设计特征的方法，然后描述训练和测试样例（事件）的过程。这些例子是运用在 AQ15c 学习程序上的，并且假设规则都是需要的。这些学习得到的规则是用测试数据来评估的，经验上的错误估计是难免的。有关结论可供进一步研究。

8.2　一种学习设计规则的方法

8.2.1　概述

越来越多的人喜欢把归纳学习运用到设计知识的获得问题上面。这种运用领域的一个重要的问题是设计知识经常模棱两可，往往一个给定的设计任务有多种解决方法。因此，一个学习系统必须能足够表示这些模棱两可的情况。但是，大多数已有的学习程序（如决策树学习）对这种工作不合适，因为它们产生各自特有的类（概念）描述。

在学习程序的 AQ 族里运用了一种用于表示模糊知识的强有力的方法。这些程序是基于 A^q 算法的，对于给定的对象集产生覆盖（规则集）。对每个要学习的概念，程序产生一组逻辑上能和一条或更多别的概念的规则集交叉的规则（规则集），如果这对于已有的示例不矛盾的话。如果一个新的对象满足

不只一条规则，那么这些规则所表示的类作为可选项被提供。

要使得归纳学习成为可能，每个决策类必须设置一个测试数据集。如果我们考虑设计规则的学习，那么每个类别对应于某一种设计解决方法（如轴承类别）。每个训练例子描述一个属于该类的对象，由一些属性的值来表示。正例应该囊括学习该类所必需的分类规则，反之，反例应该不囊括这些规则以免产生规则多余的泛化。

某些例子可能同时属于多个类（特别有许多类是重合的），这些例子叫做**模糊例子**。处理这种例子有两种方法：对于当前类，可以作为正例处理也可以作为反例处理。如果模糊例子作为反例处理，那么它不应该囊括任何相应的分类规则。否则，它必包含了不只一条规则。后者对设计规则的学习是合适的，因为一个问题有多个解的情况经常发生。

要用A^q方法开始入门学习，学习系统必须有如下具体的相关背景知识：

- 属性域（属性、变量和值集的类型）；
- 输入假设（可选），它有两方面的含义，表示学习系统中的初始知识，或者作为在学习过程中必须优化的规则集。

作为学习的结果，我们得到了一个规则集（一种特殊的分类法），一条规则对应一个类。

8.2.2 学习规则集的经验性错误

我们用测试例子而不是用训练例子来估计规则的准确性，评估方法有如下几种：择一交叉确认或者k次交叉确认。

要评估规则集的性能，在所有可能的设计解中仅有一个很小的样本。我们仅仅能够通过计算经验性错误率来估计真正的错误率。我们考虑了如下经验性错误率。

- 整个的经验性错误率，往往被当成衡量规则质量的手段：

$$E_{ov}=\frac{\text{错误数}}{\text{测试次数}} \tag{8.1}$$

如果一个给定的测试事件被错误地分类，即表明错误发生。

- 经验性的遗漏错误率：

$$E_{om}=\frac{1}{n}\sum_{k=1}^{n}E_{om}^{k},\text{其中}E_{om}^{k}=\frac{k\text{类的遗漏错误数}}{k\text{类的正例数}} \tag{8.2}$$

如果对于一个给定的类，正例被归类到反例中，则认为遗漏错误发生。

- 经验性的非遗漏错误率：

$$E_{cm}=\frac{1}{n}\sum_{k=1}^{n}E_{cm}^{k},\text{其中}E_{cm}^{k}=\frac{k\text{类的非遗漏错误数}}{k\text{类的反例数}} \tag{8.3}$$

如果对于一个给定的类，反例被归类到正例中，则认为非遗漏错误发生。

8.2.3　应用已学习到的规则处理新例子

一旦我们从训练的例子中学会了准确的规则，那么可以把它应用到新的、没见过的例子的分类上。要确定一个例子属于哪一类，我们可以分别计算该例子对于每一类的规则的匹配程度。这个值叫做信任度（给定的例子属于该类的程度）。把我们基于 INLEN 方法对经验上的不同标准和我们实际上使用的标准做一比较。对于每一个类 k，我们评估其信任度为 DC_k。对于事前设定的阈值 T，$0<T<1$，我们可定义以下的推理结论：

（1）如果存在 k，使得 $DC_k>T$，而且 DC_k 大于其他的 DC 值，那么很可能该例子属于 k 类。这样的实例对应于分类上的一种“尖”的结果，也就是当仅仅存在一个设计方案能满足输入数据的时候。

（2）如果对于每个 k，我们有 $DC_k\leqslant T$，那么不存在任何解决方法（分类结果未知）。这样，很可能我们的知识基础还不完全，权宜之计是应用其他的数据引进一个更复杂的规则集。

（3）如果同时有几个 k 值满足条件 $DC_k>T$，而且没有哪一个明显比别的大，那么这是分类模糊的情况，我们不能决定该例子属于哪一类。

当我们面临有多个解决方案的设计问题的时候，已有的推理结论是有所帮助的。

8.3　一个示范问题的描述

对于在典型驱动中，设计防摩擦轴承配置的规则学习实现了上面所介绍的方法。图 8.1 给出了这种任务的一个可行解。但是，有相对很多的设计可行解，而缺乏对可能用于解决问题的一般设计规则的认识。让我们考虑汽车里轴承驱动的设计。我们发现有许多的解，而且每一种解都被成功地应用到成

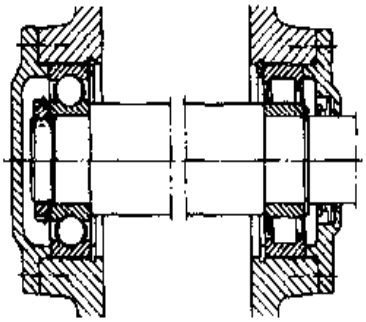


图8.1 某轴承配置的一个设计理念（SKF，一般目录）

千上万的汽车上。因此，通常设计的任务有多个解，我们需要给出多种解决方法，但仅仅可以建议用户哪一种设计是最“流行的”。

设计工作是一种日常性的设计的话，可能要分几个阶段来解决。所要讨论的这种工作并不是小事，没有对可以用图表示的描述解的设计特征进行选择的算法。设计的阶段在图 8.2 给出。

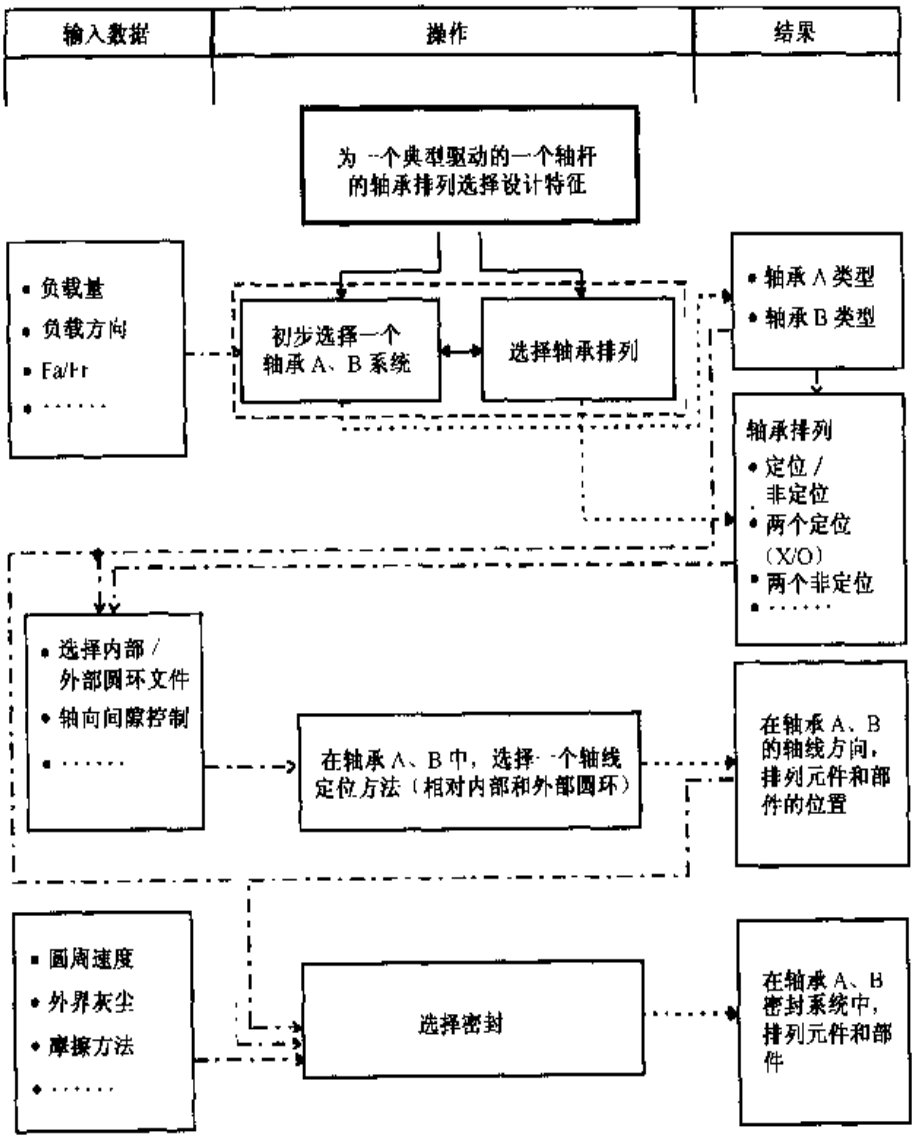


图 8.2 带两个滚动轴承 A，B 的轴的设计特征的选择阶段

关于防摩擦轴承设置的设计知识发展迅速。有很多可用的知识来源（如课本、防摩擦球轴承产品目录、资深设计者等）。该领域的知识是一门机械工程课程。但是，正如以前所解释的，设计知识在某种意义上是模糊的：对应于一个特定的输入数据集，可能存在不只一种可应用的轴承类型。

应该注意的是，设计过程的许多阶段是彼此密切相关的。虽然有一个确定的任务序列，但是我们发现几个任务间的更复杂的关系，而且有时还不能明确把他们区分开来。例如，若在应用锥形滚筒轴承时需要输入数据，则必须同时选择轴承系统的配置。几个阶段的结果就是给其他阶段的输入数据。例如，归纳阶段的结果是设计一个封闭系统的输入数据。

如果我们能获得设计规则形式的知识，那么也可能开发出可代替人类设计者的软件，解决轴承配置设计的日常任务，并通过画图来表达其设计特征。因此这个问题是非常重要的，可以归类为一种 CAD 方法。

下面，我们根据输入数据给出了对防摩擦轴承类型的研究所获得的一些初步结果。任务之间的相互关系如图 8.2 所示（阴暗部分表示的是任务）。要选择这些轴承，我们首先要考虑预计的产品的生命周期、负载、运转速度、环境条件以及必需的服务措施。轴承类型通常是以设计者优先考虑的知识，如相对的辐射、轴的载量、几何约束、预计校值、可服务性等为基础确定的。然后，设计者可以运用定义好的方程式，对轴承大小进行评估。

在设计者确定了轴承配置的情况下，我们希望学习到的规则集能被他们以同样的方式进行应用。因此，我们必须获得描述环境的数据集（机器的位置、它的环境等）和定量的数据（辐射或轴的载量、运转速度、轴承直径等），这两者构成设计过程的输入数据。当设计操作者将输入数据向答案进行转换的时候，机器学习过程中学到的规则集将产生作用，在我们的实例中这叫轴承类型。

8.4　归纳方法的应用

我们应用在可行资源中可获得的启发式研究和知识，是为了确定哪些域和域类适合表示关于用于机器学习阶段的示例数据。以下提出我们面对过的更重要的问题。

我们决定应用变量的定性值而不是定量值。这种解是由学习的规则集的

普遍特征产生的，这些特征可能广泛应用于关系设置的设计中，不依赖变量的特定值，如：负载数值、轴直径、运转速度等。因此，每个域将仅仅包含在域中已定义的一些属性的少数可能值。

选择的属性域和大量值及值名如表 8.1 中所列。“装载条件”的序列值指轴与辐射载量值的百分比（如 rd/25ax 指轴的载量是辐射载量的 25%）。如果可以对值排列顺序（引入关系≤），那么定性属性是线性的，除非属性只是名义上的（包括质量，属性值之间不存在关系）。

表 8.1 属性域

	类型	尺寸	属性名称	属性描述	属性值
1	线性	3	sh_dia	轴直径	小，中，大
2	符号	2	rd_spc	半径空间约束	是，否
3	符号	2	ax_spc	轴线空间约束	是，否
4	线性	3	mag_ld	负载大小	小，中，大
5	线性	13	cnd_ld	负载条件	半径，rd/25ax,…，轴线
6	线性	3	misalg	可能出现的错位	小，中，大
7	线性	4	rot_sp	转速	小，中，大，很大
8	线性	3	frictn	磨擦大小	小，中，大
9	线性	3	stiffn	刚性大小	小，中，大
10	符号	2	maintn	维护必要性	无用，容许

8.4.1 变量的定性值

连续变量以定性值表示是一个困难的问题，因为总是关系到一些信息的丢失。通常的做法是对属性的定量值进行聚类。开发的系统应该能够把连续变量作为输入数据，用来推理轴承类型之前将其转变成定性值。在已经发表的研究中，我们应用了一种基于由机器设计背景知识推理出来的启发性知识的方法。

考虑表 8.1 中包含的属性，我们能区别如下两类属性。

- 独立于轴承负载和其维数的属性（关系特征）：如属性“一轴承相对于密封垫圈的线速度”可以取以下定性值（启发式方法：与密封垫圈的具体类型相关，这里 v 值指轴承的线速度）；
- 小，即 $v<4$m/s（如果操作温度超过 100℃，可用橡皮密封垫圈——一种简单便宜的密封垫圈）；

- 中，即 $v<8$m/s（可用一种辐射唇型密封垫圈）；
- 大，即 $v\geq 8$m/s（应用一种耐摩擦的密封垫圈）；
- 依赖轴承负载大小、尺度和轴承类型的属性：如属性“载量大小”可取以下值（P=动态关系载量，C=基础动态载量率）：
- 轻，如果 $P\leqslant 0.06C$，
- 正常，如果 $0.06C<P\leqslant 0.12C$，
- 重，如果 $P>0.12C$。

应该强调的是，属于第二类的属性值的表示可能会很困难。例如，属性“载量大小”的极限值隐含了这样一种意思，即为了获得可能在未知关系集中应用的定性输入数据，设计者必须运用所选轴承的性质（基础动态载量率 C）。

8.5 训练与事件测试

基于归纳的机器学习方法需要大量的训练样本（即事件）。

每个样本都由 n 个属性值代表，例如，样本所属的类名。形成了设计意图后，我们将用到一个属性集。这个属性集描述了在机器背景下的一些关系情况，而且这个属性集也能够代表定性数据。这些属性由定性值所表示。我们在研究过程中所用到的事件数据库在 Maniak（1995）文献中有详细的描述。

8.5.1 设计知识源

关于轴承配置设计的相关知识可以在易得到的产品目录中找到。比如：滚动轴承的目录，由轴承制造商印刷的特殊刊物以及关于机器设计的教科书等。这些知识源包含了大量的样本和图表，构成了关于设计知识的信息载体。

带有图表及附带描述的样本，被编码成事件。按照准备事件人的意见，将定量值转换成定性值。然后将挑选出来的事件以机器学习软件能够接受的方式写进文件中。

应该强调的是，我们花费大量的时间来优化属性集，目的就是为了能够区分不同的轴承类。

在接下来的研究中，我们将从有技术、有经验的设计者那里获取知识。

8.5.2 样本数据库

我们参照下面的轴承类型收集样本：深槽球轴承、角接触球轴承、自校正球轴承、滚动柱轴承、滚球轴承、递减滚动轴承、针状滚动轴承、穿球轴承、穿柱体滚动轴承和穿滚球轴承。对于每种类型我们收集 10～26 个事件（总共 200 个事件），在表 8.2 中给出了其中的一些样本。整个代表空间包含 101088 个可能的事件，收集的样本仅代表事件的 0.2%。在我们看来所选出来的事件数目太小了，不能提供关于问题领域的可靠知识，因此该把研究领域解释成一种可能性，而不是最后阶段。因此，应该对更多数目的事件加以研究。

表 8.2 关于“深槽球”类训练事件举例

sh_dia	rd_spc	ax_spc	mag_ld	cnd_ld	misalg	rot_sp	frictn	stiffn	matntn
中	是	是	中	rd/25ax	小	很大	小	大	无用
小	是	否	中	rd/50ax	中	大	小	中	容许
中	否	是	中	rd/25ax	中	大	小	中	容许
中	是	是	小	轴线	中	中	小	中	无用
中	否	是	大	轴线	小	中	小	中	无用
大	否	是	小	rd/25ax	小	小	小	不	无用
中	否	是	小	rd/25ax	小	小	小	小	容许
大	否	否	中	轴线	中	很大	小	小	无用
中	否	否	中	rd/25ax	小	小	小	大	无用
小	否	是	中	轴线	中	很大	小	中	无用
⋮									

数据给出的样本对支持的条件（机器的位置及其周围环境等）和数值数据（圆周和/或垂直负载、旋转速度、轴直径等）加以描述，这也为真正的设计过程提供了输入数据。

8.6 结果分析

以下我们给出按 AQ15c 可选择归纳机器学习程序得到的结果。通过多次实验，我们找到了控制程序的最优参数。

- 我们创建最少数量的可能的扩展选择符，以及最小数目的对应扩展选择符的值。这使我们能够学到尽可能简单的规则。
- 对每一分配的类，我们认为是正例的模糊例子，因此，这些例子能够被一条以上的规则所覆盖。这种模式对应了应用领域的特殊性。

- 我们采用一套标准的默认集来定义一个词典函数。该默认集包含了最小损耗（属性的最小损耗）和最少选择（扩展选择的数目的最小化）的标准。

8.6.1　从训练样本中学习规则

所得到的引用结果看起来是非常鼓舞人心的。针对所得到的训练集的准确度，我们将代表空间（描述事件的域和可能的属性值）优化，这样就将分成了 13 类的准确度提高到了 92%。采用 VL_1 提出的对复杂性进行分离的方式，对于每一类我们都得到一条规则。这种方式的提出使得读者容易理解 AQ15c 机器学习程序产生的规则。表 8.3 给出了 VL_1 提出的研究方法得到的一条样本规则。每一行我们都可以看到一个单一结构的综合。此规则是几个综合的分离。在“strength”栏中，我们对每个复杂性加以定义：t 为支持规则的样本总数，u 表示单一的样本数量。此规则向自然语言的转换如表 8.4 所示。

表 8.3　有关深槽球轴承的样本规则

深槽球-代码		
#	样　本	强度
1	[cnd_ld=rd/25ax]	(t:10, u:10)
2	[mag_ld=中] [cnd_ld=轴线，rd/50ax] [misalg=中] [rot_sp=大……很大] [stiffn=小]	(t:4, u:4)
3	[rd_spc=是] [ax_spc=是] [cnd_ld=轴线] [stiffn=小……中]	(t:4, u:3)
4	[ax_spc=是] [cnd_ld=轴线] [rot_sp=中] [stiffn=中]	(t:4, u:3)
5	[rd_spc=否] [ax_spc=否] [mag_ld=小] [cnd_ld=轴线] [misalg=小……中]	(t:3, u:3)
6	[ax_spc=是] [cnd_ld=轴线] [rot_sp=很大]	(t:2, u:2)

表 8.4　样本规则向自然语言的转换

采用深槽球轴承需要满足以下条件之一：

（1）负载为 25%的轴向辐射负载。

（2）负载的辐度为中间值，负载类型为辐射型（或负载为 50%的轴向负载），不对齐的程度为中等，旋转速度尽可能大，轴承的韧性较小。

（3）辐射空间和轴空间受到限制，负载为辐射型，韧性小于中等水平。

（4）轴空间受到限制，负载条件为辐射型，旋转速度取中值，韧性为中等水平。

（5）辐射空间和轴空间不受限，负载最值较小，为辐射型，非线性小于中等水平。

（6）轴空间受限，负载为辐射型，旋转速度非常大。

表 8.5 给出了测试结果的汇总。此表的内容可以用于决定经验错误率的状

况。Eq.（8.1）中大体上的经验错误率 E_{ov}=0.080，因此规则集的准确率达到了92%。从表 8.5 我们分别可以得到默认的经验错误率 E_{om}=0.066 和期望的经验错误率 E_{em}=0.007。

表 8.5 用“除去一个”方法测试结果汇总

类别编号	轴承类型	数目			
		正例	正确分类	漏检	误检
1	深凹槽球轴承	26	20	6	4
2	单列向心推力球轴承	12	12	0	0
3	双列向心推力球轴承	10	10	0	0
4	自排列球轴承	13	11	2	0
5	类型 NU 圆筒形滚柱轴承	13	10	3	8
6	类型 NUP 圆筒形滚柱轴承	21	16	5	4
7	双列圆筒形滚柱轴承	15	15	0	0
8	滚针轴承	10	10	0	0
9	锥形滚柱轴承	17	17	0	0
10	球状滚柱轴承	25	5	0	0
11	推力球轴承	16	16	0	0
12	圆筒推力球轴承	10	10	0	0
13	球状推力球轴承	12	12	0	0
总计		200	184	16	16

进而我们就有足够的背景领域知识来探讨学习设计规则的质量。为了评估得到的结果，我们采用启发式方法来分析错误分类的例子。所有用来描述多类型的例子都可以采用（例如如果期望带有中等速度和中等部分系数的辐射性负载，我们就经常采用深槽球轴承或滚柱轴承）。从设计者的角度来看，这类事件也是数量巨大的。因此我们不希望得到模糊的分类结果。

8.6.2 以递增学习方法评估预备样本

为了评估样本数据库的正确性和完整性，我们进行了另一类预测测试——递增学习。正如原始规则那样，我们使用了防摩擦轴承目录中提出的规则集。

目录中提出的规则是完备的，以验证我们假设样本的正确性。作者提出的假说如下：如果预备样本集与原始规则是相符的，那么以学习样本方式，通过“提炼”原始规则得到的规则，将不会比仅仅从样本得到的规则更好。

在表 8.6 中我们给出了研究的选择结果，在 Maniak（1996）中有详细的描述。此阶段得到的规则集的大体准确性比先前讨论的要小。无论怎样，对

应于概念规则正确性的得失可以观测到：

- 类别 1，4 的准确性。训练样本对先前的规则集加以提炼，得到的规则集合会对非可视的测试样本有更好的分类。我们认为从与输入假设相符的训练样本中，可以学到另外的知识。
- 对应于数据和原始规则的一致性，类别 8，11，12，13 的准确性没有发生改变。然而，在此学习过程中，很有可能学习到新的知识。
- 为了保持住类别精确性的损失：这里，技术与规则是不相符的，于是我们得出结论，这些样本不能代表类别的概念，甚至是不正确的。

表 8.6　利用增量学习的训练实例评估

类别编号	轴承类型	准确率		准确收益（%）
		前期学习	增量学习	
1	深凹槽球轴承	76.9	84.6	+7.7
2	单列向心推力球轴承	100.0	58.3	-41.7
3	双列向心推力球轴承	100.0	70.0	-30.0
4	自排列球轴承	84.6	100.0	+15.4
5	类型 NU 圆筒形滚柱轴承	76.9	38.5	-38.5
6	类型 NUP 圆筒形滚柱轴承	76.2	57.1	-19.1
7	双列圆筒形滚柱轴承	100.0	26.7	-73.3
8	滚针轴承	100.0	100.0	0.0
9	锥形滚柱轴承	100.0	88.2	-11.8
10	球状滚柱轴承	100.0	84.0	-16.0
11	推力球轴承	100.0	100.0	0.0
12	圆筒推力球轴承	100.0	100.0	0.0
13	球状推力球轴承	100.0	100.0	0.0
总计		92.0	77.0	-15.0

从上面的例子我们可以得出，代表这些类的样本对于通常的领域知识是不充分的。那么为了更好地描述相应概念我们应该修正这些样本。

8.6.3　得到结果的可信度

从讨论的例子中，我们得到了非常令人心服的结果。使用样本深槽球轴承的分类结果，我们来演示怎样应用可信度数值评估学习到的规则集。

我们采用了常数 T 的三个值：0.25,0.50 和 0.75。采用这些阈值，整个规则的准确性会分别降到 86.5%，66.0%和 56.0%。对应于单类的一个样本可以更清晰地看出常数 T 的影响（如表 8.7 所示）。分析此表得到的结果我们会发现：对于给定值 T，正确分类样本的数目分别降到了 18，8 和 8。

表 8.7　深槽球轴承例的可信度值举例

时间号 #	所属类别	实例对类别的符合度（可信读）													可信度最大值	阈值 T		
		cl1	cl2	cl3	cl4	cl5	cl6	cl7	cl8	cl9	cl10	cl11	cl12	cl13		0.25	0.50	0.75
1		0 41	0.00	0 00	0 00	0.00	0 00	0.00	0.00	0.00	0.00	0.00	0 00	0.00	0.41	+	-	-
2		0.36	0.00	0.00	0.00	0.00	0 00	0.00	0.00	0 00	0 00	0 00	0.00	0 00	0 36	+	-	-
3		1 00	0.67	0.50	0.96	0.98	0.99	0.95	0.75	0 67	0 75	0 50	0.00	0 00	1 00	+	+	+
4	cl7	0.93	0 33	0 50	0 94	0.90	0 96	0 98	0 50	0 33	0 44	0 50	0.00	0 00	0 98	+	+	+
5		0.36	0.00	0 00	0.00	0.00	0 00	0 00	0.00	0.00	0.00	0 00	0.00	0.00	0.36	+	-	-
6	cl7	0 88	0.67	0 75	0.98	0.99	0 96	0.99	0.50	0 67	0.88	0 50	0.00	0.00	0.99	+	+	+
7		1 00	0 50	0.50	0 96	0.99	0.99	0.94	0.50	0.67	0 94	0 50	0 00	0 00	1 00	+	+	+
8	cl6	0.00	0 00	0 00	0.00	0 00	0.05	0 00	0.00	0 00	0 00	0.00	0 00	0.00	0.05	-	-	-
9		0.36	0 00	0 00	0 00	0 00	0.00	0 00	0 00	0 00	0.00	0.00	0.00	0.00	0.36	+	-	-
10		0.36	0 00	0.00	0 00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0 00	0.00	0.36	+	-	-
11	cl6	0 97	0 67	0.00	0.94	0.99	0.96	0.98	0.50	0.33	0.91	0.50	0.00	0 00	0.99	+	+	+
12		0.08	0.00	0 00	0.00	0 00	0 00	0 00	0 00	0 00	0 00	0 00	0.00	0 00	0 08	-	-	-
13		0 36	0.00	0.00	0 00	0 00	0 00	0.00	0.00	0.00	0 00	0.00	0.00	0.00	0 36	+	-	-
14	cl6	1.00	0.83	0.25	0 81	1.00	0 99	0 91	0.50	0 67	0 91	0 50	0.00	0 00	1.00	+	+	+
15		0.12	0.00	0.00	0.00	0.00	0.00	0 00	0.00	0 00	0 00	0.00	0 00	0 00	0 12	-	-	-
16		0.98	0.50	0.25	0.83	0.89	0 98	0 96	0.50	0 33	0.91	0.50	0.00	0 00	0 98	+	+	+
17		0 36	0.00	0 00	0.00	0.00	0 00	0.00	0 00	0.00	0 00	0.00	0 00	0.00	0 36	+	-	-
18		0 36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0 00	0.00	0.00	0.00	0.36	+	-	-
19		0.12	0 00	0.00	0.00	0.00	0.00	0 00	0.00	0.00	0 00	0.00	0.00	0 00	0.12	-	-	-
20		0.36	0.00	0.00	0.00	0 00	0 00	0.00	0.00	0.00	0 00	0 00	0 00	0.00	0.36	+	-	-
21		0 99	0.33	0.50	0 98	0 98	0.94	0.98	0.25	0.33	0 97	0.50	0.00	0.00	0.99	+	+	+
22	cl6	0.00	0.00	0.00	0 00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0 05	-	-	-
23		0.12	0.00	0.00	0 00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.12	-	-	-
24		0 12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0 00	0.00	0.00	0 12	-	-	-
25		0.36	0.00	0.00	0.00	0.00	0.00	0 00	0.00	0 00	0 00	0 00	0.00	0.00	0.36	+	-	-
26	cl6	0.08	0.00	0.00	0.08	0.23	0.00	0 00	0.00	0 00	0.00	0 00	0.00	0.00	0.23	-	-	-
对于不同阈值 T，正确分类的例子数																18	8	8

如果没有对于值的可信度的需求，我们会得到一个大概结论，因为有些训练样本呈现出非常低的可信度值。例如，样本 8 以 0.05 的低可信度被错误地分到了类 6 中，而样本 2 以 0.08 的可信度被正确分到了类 1 中。这些样本都不能加以接受。当属于某类的最大可信度少于给定阈值 T 时，这些样本就不能接受。

分类的巨大收获的另一个例子由大量的样本（样本 3，4，6，7，11，14，16，21）加以描述，这些样本属于类别 1，可信度总和达到了 1.0，但是它们也很有可能属于另一些类。在此例子中，两种方案都是可行的：我们不仅有正确分类的样本，而且有错误分类的样本。然而，此例中我们发现了一个有趣的现象：正如设计者解决设计任务时面对不同的问题，解决方案也不只有一个。

8.7　总结

通过研究证实，为了帮助设计者解决以往的设计任务，获得有用的设计知识，采用机器学习方法是可行的。可以应用归纳方法有效地从样本中获取学习设计规则。本章关于选择轴承类别的研究进一步证实了这种方法。所得到的规则集达到了 92%的高准确性，并且在按照准确度标准优化描述空间的过程中获得实现。此优化可解释为增加大量的背景知识到学习系统中。

我们引入一个采用可信度值的结论表。这种方法与设计过程保持很好的一致性，使得专家系统对于一项设计任务可以提出几个不同的方案。设计者本人必须参照自己的背景知识来解决任务。

我们简要证明了 AQ-15 系统在学习和描述大量知识上是十分有效的。这种例子在机器设计领域是十分普遍的。

在此项研究中，要想获得更进一步的证实结果，需要与有经验的设计者进行合作，这一点将在下一阶段加以探讨。我们会找到对于问题领域其他子任务的规则集，尤其是对于密封垫圈的运用，轴承的设置，以及考虑到轴杆和遮盖物而对轴心的设置。作者已经进行了相关的研究，P.Maniak 也已对此领域进行了研究。

利用启发式知识（问题领域的背景知识）可以将后继的属性转变为离散的属性值。

得到的结果表明，此方法对于创建基于帮助设计者解决传统机器设计任

务的知识系统而言，是十分有用的。

致谢

作者对美国 George Mason 大学的 Ryszard S．Michalski 教授表示感谢，RSM 教授友情提供了我们研究中使用的 AQ15c 归纳学习程序，还有他本人有益的建议也为本文的最后定稿奠定了基础。另外作者也要为 Janusz Wnek 和 Eric Bloedorn 提出的关于经验错误估计方法而对他们表示感谢。

Piotr Maniak 准备了大量的训练和测试样本，得到了经验错误的大致估计值，作者也对他的贡献致以谢意。

参考文献

Arciszewski, T. (1994): Machine Learning in Engineering Design. *3rd International Symposium 'Intelligent Information Systems'*. Institute of Computer Science, Polish Academy of Sciences, Warsaw-Wigry, pp. 40-54.

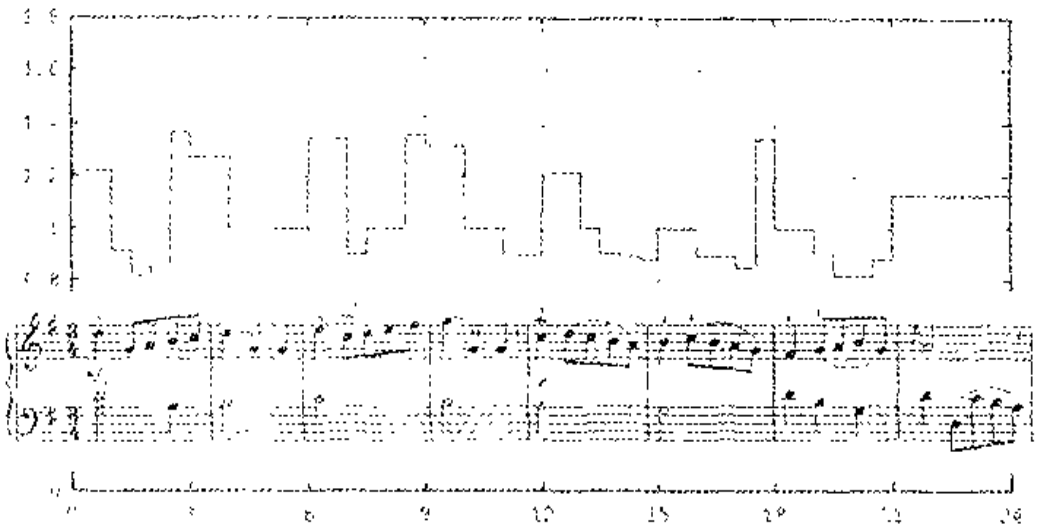
Arciszewski, T., Dybala T. and Wnek J. (1992): A Method for Evaluation of Learning Systems. *HEURISTICS, The Journal of Knowledge Engineering*, Special Issue on Knowledge Acquisition and Machine Learning, Vol. 5, No. 4, Winter 1992.

Dietrych, J. (1985): *System and Design* (in Polish). WNT, Warsaw.

Maniak, P. (1995): Application of Machine Learning Methods to Knowledge Acquisition on the Design of Machinery (in Polish). *M. Sc. Thesis*, Technical University of Silesia,Department of Fundamentals of Machine Design, Gliwice.

第 3 部分

文本、图像和音乐模式的测定



第9章　找出文本之间的关联

Ronen Feldman 和 Haym Hirsh

摘要

本章描述了 FACT 系统如何找出文献集中标注条目的关键字之间的关联——搭配模式。FACT 采用中心查询的知识挖掘观点，即发现关联的请求被认为是对由文献集所支持的可能潜在的关联的一种查询。而且，背景知识可以利用到文献标注关键字上时， FACT 能够将此信息利用到搜索进程当中，允许用户在查询结果上定义一些背景知识约束。更有甚者，关系查询的实现是有组织的，以使得这些背景知识约束能被用来搜索可能的结果。最后，与其要求用户用知识发现查询语言指定一个查询表达式，FACT 给用户提供了一个简单易用的查询语言绘制界面，提供了一套详尽的语义系统，用户通过界面可以执行查询操作。

9.1　介绍

在过去几年中，因特网资源信息利用方面取得了难以置信的发展，但它在带来好处的同时也带来了不少坏处。虽然现在我们有大量的免费信息可供利用，但是对大量的无用信息的探索与理解也变得非常困难和耗费时间。这能帮助用户访问和理解大量多重信息的工具的发展，对于我们从无数的网上信息中提取所需要的信息也是有必要的。

这也是许多数据库知识挖掘(KDD)工作的主要动力。一向的难点被描述为“对数据进行盲目的、未知的、潜在有用信息的特征提取”(Frawley 等人，1992)。遗憾的是，这方面的工作最初都集中在有组织的数据上，很少涉及到无组织的数据。本章讨论文章知识挖掘问题(KDT)(Dagan 和 Feldman，1995;

Feldman，1996)，目标在于从无组织的文献集中提取信息。

此工作的一个主要思想是：从信息恢复研究中获取教益，表明浅显的信息表示经常给一定范围的信息访问任务提供充足的支持(Salton 和 McGill，1983，Frakes 和 BaezaYates，1992)。我们的途径是使用领域水平的关键字来标注文献，以分析和操作它们。知识挖掘产生了，以用来找出每个文献标注关键字之间的相关性(Agrawal 等人，1993，1995; Mannila 等人，1994)。也就是说，例如，如果给出一个新闻文献集，你可以找出每篇文章中的令人感兴趣的同时出现的关键字，就像国家名通常用来标记一篇文献而不管何时其他国家名也标记此文献。

本章描述了 FACT(Finding Associations in Collections of Text)，一个集中用于找出有标注的文献集之间的关系的 KDD 工具。中心工作是发现进程的集中查询观点(Imielinski 等人，1996)。给出一组数据，有一个这组数据支持的协调的固有的、可能的结果集。FACT 为发现进程提供一种查询语言，以使得用户能够在数据支持的、可能的结果集上做详尽的查询。然而，与其要求用户指定一个明确的查询语言语法，不如 FACT 给用户提供一个易用的查询语言图形界面，在这儿用户可以详尽地指定多变的查询任务，它同时为用户通过界面执行查询操作提供了一套详尽的语义系统。

更进一步的是，FACT 可以开发对系统有用的背景知识形式。例如，在前述新闻文献中，系统要找出新闻文献标注关键字之间的关联，必须使用有关国家的知识——如国家的人口、大小、出口产品或者组织成员人数（NATO，G7，Arab League 等）——或者与国家相关的信息，如它们是否是邻国、贸易伙伴或有共同语言。这些背景知识可以从许多不同的资源中得到，如从领域事实的数据库，甚至其他文献信息资源中得到(工作中的一个实例)。FACT 允许用户在有关文献标注关键字的背景知识中包含一个对期望结果的强制查询。甚至，FACT 将使用这些强制来搜索可能的结果。背景知识反过来可以促进 FACT 发现关系，例如，国家 G7 作为一篇文献的标注，其他一些与 G7 不相关的国家也作为此文献的标注，背景知识有利于找出它们两者间的关联。

我们从 FACT 系统的总体结构来开始介绍。然后描述发现关系的一般性难题、关联挖掘查询语言、用此语言进行查询的算法和发现关系的工具。下一步我们将讨论 FACT 对路透社新闻集的作用，这些新闻都是利用背景知识从 CIA World Factbook 上自动提取的。最后，我们将结合上下文简要地介绍其

他 KDD 成就来结束本章。

9.2 FACT 系统结构

FACT 系统的总体结构如图 9.1 所示。顶端有三种信息资源提供给 FACT。最右端是数据集，挖掘处理在这里进行。既然有关信息恢复的文献中提到每篇文章由一组关键字标注，而我们的工作又从其中假设开始，那么输入的文集必须已经包含了这些关键字(如 9.7 节中讨论的路透社数据)或者经过一个文集分类系统以增加包含这些关键字的文献。

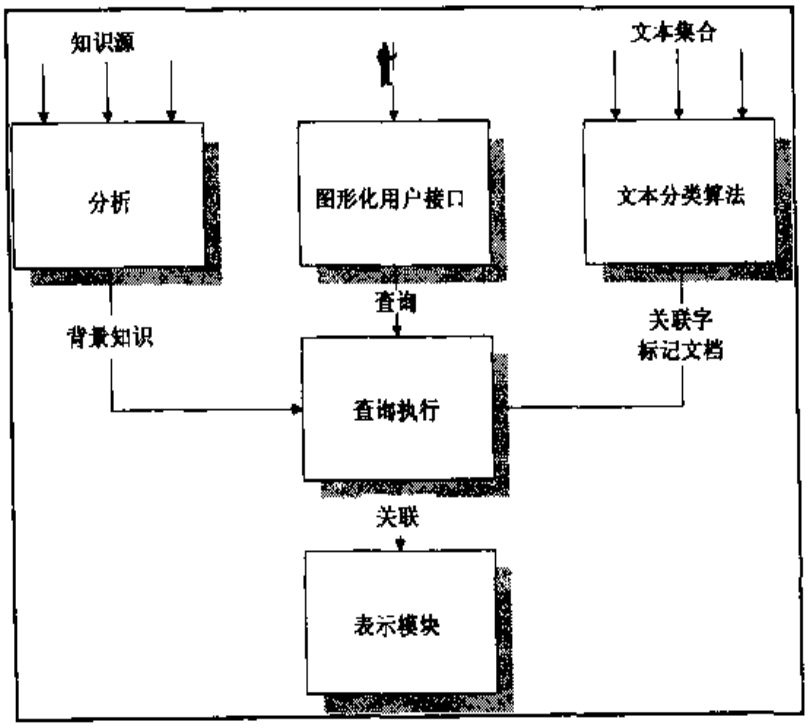


图 9.1 FACT 系统结构

最左端是知识库，为挖掘过程提供额外的背景知识。这对 FACT 是很有用的，这些知识必须详细定义文献关键字的一元和二元谓词，描述由关键字及其关系所体现的实体的特性。所以对于路透社新闻数据，例如，FACT 被告知：对于每个国家关键字，国家是组织的一员，于是对国家关键字定义一组二元谓词(每个组织一个)。FACT 也被告知有关邻国的信息，于是对国家关键字定义一组二元谓词。由于这些信息很少能被 FACT 系统的精确形式用到，有必要开发一种工具来理解信息库格式，并能够将之转换成 FACT 所需的格

式。例如，在我们的路透社新闻试验中，使用的背景知识是从 CIA World Factbook——一个关于世界各个国家信息的有组织的文献中提取而来。为使 FACT 能够运用这些知识，我们必须开发一种工具来分解 Factbook 的有组织的文章，并将其转换为 FACT 所能用的格式。

将 CIA World Factbook 转换为一元或二元谓词的一个难题是 Factbook 的词表与标注文献的关键词不相匹配。例如，在许多提及阿曼的文章中，我们有一个关键字“天然物”。然而，在 CIA World Factbook 中当描述阿曼的出口产品时，我们有鱼、石油、铜和纺织品等关键字。显然，如果一个用户进行一项阿曼和与其出口产品无关话题的查询时，阿曼与“天然物”关联将是返回结果，因为“天然物”在阿曼的出口产品中并没有被提到。因此，FACT 提供一个额外的背景知识库，定义以上讨论的在第一个背景知识库中定义的一元、二元谓词和标注文献关键字之间的同义词。图 9.2 展示了这样的功能，即允许用户选择了一个背景知识（此例中，谓词从 CIA World Factbook 中提取）定义的词后，可以选出一列与其同义的关键字。当后来的用户定义一个用到背景谓词的查询时，FACT 就能找出与标注文献关键字表相关联的谓词。(虽然通过辞典可以潜在地帮助定义同义词，但我们在这项工作中并没有研究它。)

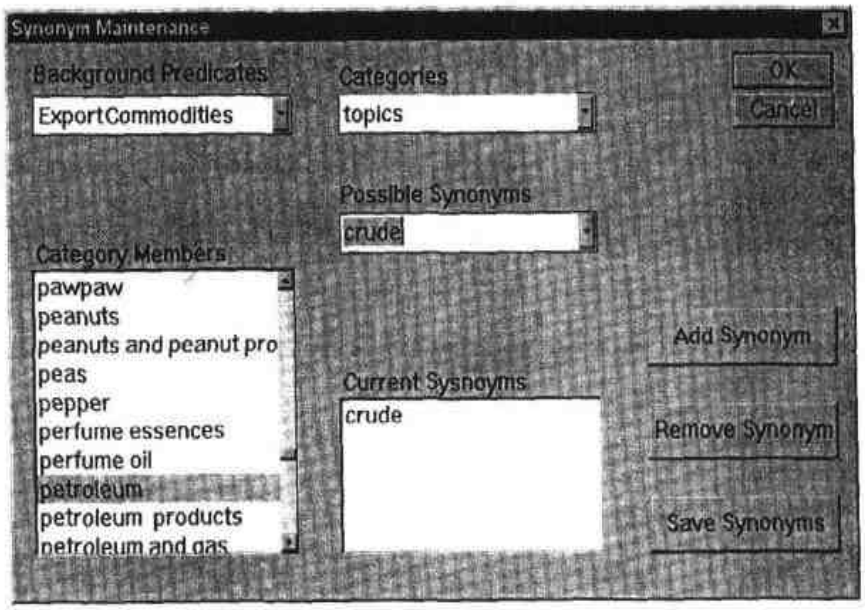


图 9.2 同义词定义功能

最后，中心输入是关于知识挖掘任务的用户规则，这可以通过一个简单的图形界面从用户处获得。此界面知道各种各样的标注文献的关键字以及由

背景知识定义的各种一元或二元谓词，这些背景知识是可以应用到这些关键词上的。界面允许用户通过一组菜单指定一个使用此关键字与谓词表的查询。用户通过建立由列表控件和选择按钮描述的区域来指定知识-发现查询。图 9.3 给出了一个通过图形界面，它关于在 9.7 节中讨论的路透社数据指定查询的例子。在此次查询中，我们寻求至少由五篇文献所支持(支持度)并置信度至少为 10%(置信度)的关联。

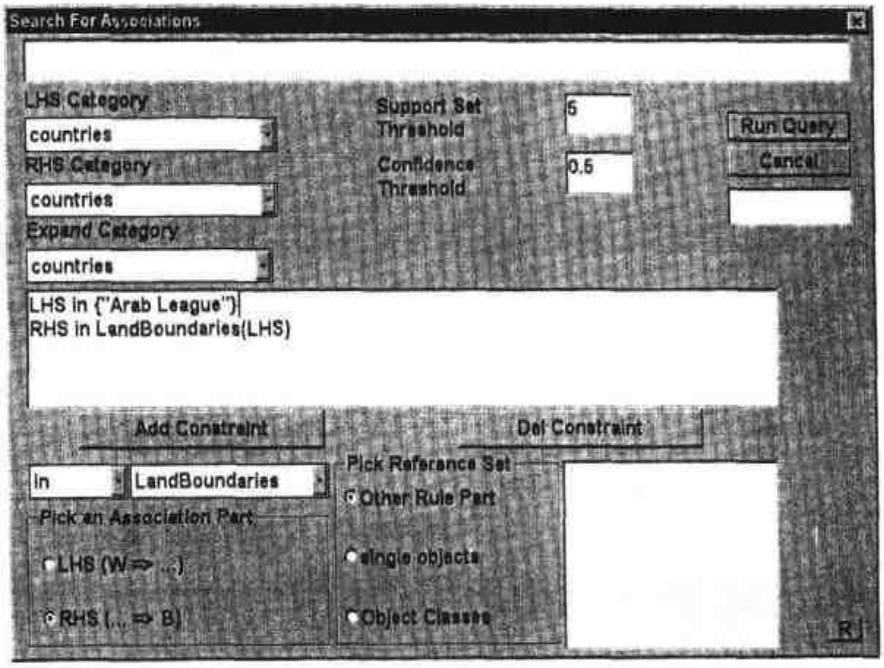


图 9.3　定义查询规范

这三个信息库——文集，背景知识和知识挖掘查询规则——是 FACT 的输入，如图 9.3 的中心所描述，它通过其查询操作编码来使用它们。挖掘进程的结果被传到工具中，如图 9.3 的下端所示，此工具可以高效率地呈现此结果并允许用户浏览它们。FACT 的这一组件过滤了多余的结果，分级别地组织它们，鉴别出了其中的一般性，通过减少无序性将其分类，允许用户访问和浏览那些支持个别供给用户结果的文献。图 9.4 所示的浏览器展示了图 9.3 中查询结果间的关联。最后，用户可以指出任何系统产生的关联（如图 9.4 所示），双击这些关联，系统将提供支持这些关联的所有文献的标题。例如，图 9.5 显示了支持关联伊拉克⇒伊朗的所有文献标题（也就是说，所有的文献都有关键词伊拉克和伊朗的标注）。

图 9.4　关系浏览器展示阿拉伯同盟国和与其搭界国家间的关系

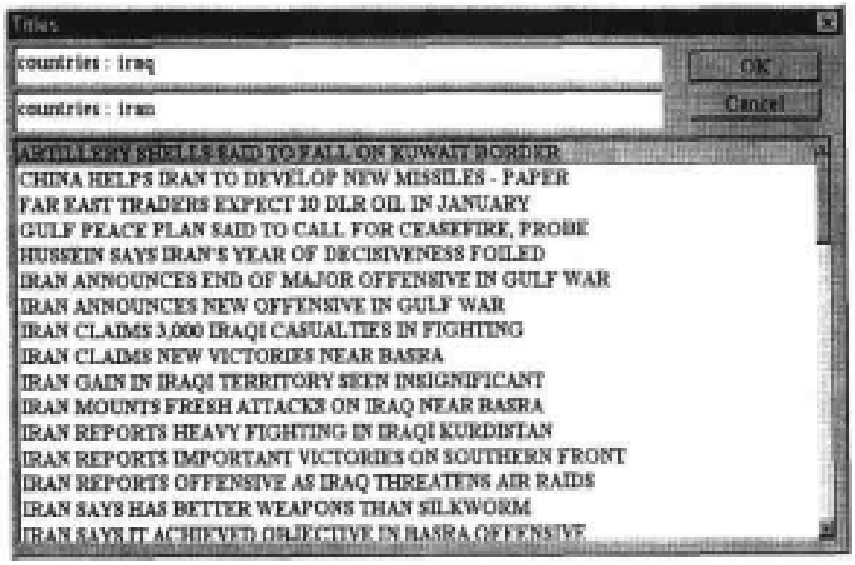


图 9.5　由关键字 Iran 和 Iraq 标注的所有文献标题

9.3　关联

FACT 的主要任务是找出文集间的关联。我们从介绍关联定义开始，在

Agrawal(等人 1993，1995)和 Mannila 等人(1994)的基础上，附加的定义和说明对本章的剩余部分是有用的。

设 $r=\{t_1, \cdots, t_n\}$是一组文集，每个都由关键字集 $R=\{I_1, I_2, \cdots, I_m\}$的子集的关键字所标注。给出关键字 A 和文献 t，我们将表达式 $t(A)=1$ 称为 A 是标注 t 的一个关键字，否则 $t(A)=0$。如果 W 是 R 的一个子集，而对于任意关键字 $A\in W$ 都满足 $t(A)=1$，则我们记做 $t(W)=1$。给出 R 中的关键字集合 X，定义$(X)=\{i \mid t_i(X)=1\}$-（X）表示对于文献 t_i，至少由 X 中的所有关键词标注。对于某一自然数σ，若满足$|(X)| \geqslant \sigma$，则称 X 是σ包含的。

若 $W\subseteq R$ 且 $B\subseteq R\backslash W$，我们称 $W\Rightarrow B$ 是 r 上的一条关系指令。我们称 W 是关系的左边(LHS)，B 是关系的右边(RHS)。若对于 $0<\gamma\leqslant 1$(置信度)和σ (支持度)，$W\cup B$ 是σ包含的且$|(W\cup B)|/|(W)|\geqslant\gamma$，我们说 r 是满足 $W\Rightarrow B$ 的。从直观上讲，在由属于 W 中关键字标注的所有文献中，至少有γ 的部分被 B 中的关键字标注，并且，至少有σ 个文档同时被 W 和 B 中所有的关键字标注。

9.4 查询语言

为执行关系-发现操作，用户必须用 FACT 的关系查询语言定义一个查询——关系-发现过程只返回满足此查询的结果(如 Klemettinen 在 1994 提出的关系与用户定义模板不匹配问题)。

每个关系-查询，查询有三个步骤。第一步先定义关系的左边和右边关键词的类型，并指定置信度和支持度。例如，用户要表达叙述用国家和人标注的两种文献的关系，只要此关系有足够的置信度和支持度。

查询操作的第二个步骤（可能默认）定义——通过背景知识定义的谓词——用户需要满足的关系约束。查询中有两种类型的背景知识可被应用。第一种是关键字的一元谓词。在定义查询中，每个一元谓词代表一类关键字，并对该组关键字它何时为真有定义。例如，当以欧盟国家名作为关键词为代表的类，存在这样一个关键字，即欧盟成员国名时，则一元谓词 EC 为真。用户可以通过定义某关键字是一元谓词所定义的类的一员，指定对关系中的某关键字该一元谓词为真。

可用于查询的背景知识的第二种类型是二元谓词，它定义了关键词之间的关系。例如，背景知识可以定义二元谓词 Nationality，第一个自变量是人名，

第二个自变量是国家名，当某人名的国籍恰好是某国家名时，此二元谓词为真。在查询语言中，每个二元谓词可以用一个函数表达：给出第一个自变量值，为使谓词为真，将返回第二个自变量的一组值。例如，谓词 Nationality 可表达为一个由人名可得到其所属国家名的函数；谓词 ExportCommodity 可表达为由国家名做关键词输入，以该国家产品出口关键词做输出的函数。而且，当函数被一组关键字应用时，对于其中任一个输入元素，函数将返回所有使谓词为真的二元变量。用户可以通过定义某关键字是某函数的返回值，其他一些关键字在关系中应用该函数，来要求某二元谓词对某些关键字为真。

最后，查询操作的第三步（也可能默认）对关系的各种组成部分定义约束。例如，用户可以要求关系的右边只有一个关键字，或要求最多五个国家关键字。

图 9.6 给出了关系查询语言的一则 BNF 语法，这里非终结字符用角符表示，字符“<integer>”描述一个整数，“（0，1）”描述 0（不包含）到 1（包含）之间的某确切数，“<CategoryType>”指对于给定的范围，将标注文献关键字分成某些子类(如“country”，“person”，等等，在路透社新闻数据中)。“*”算符的作用是描述其前的零和非终止字符拷贝；“+”是语言的结束符（表明当某变量被赋值时，其前有一个或更多关键字类型）。最后，对“<Var>”的扩展“<Arg>”是前面在“<Pattern>”中定义的变量（也就是说，这是 BNF 中不能描述的语言上下文的一部分）。

我们以一些查询语言实例及其英文意思来结束本节。

Find: (5/0.5) c1:country，c2:country ⇒ t:topic

Where: c1∈G7，c2∈{Arab League}，t∉ExportCommodities(c1)

此查询要求关系满足:当某文献被国家 G7 和属于 Arab League 的国家关键字标注时，至少有一半几率也被另一关键字标注，该关键字不是 G7 国家出口产品，并且查询在文集中进行至少 5 次。

Find: (10/0.2) c:country+ ⇒ p:person

Where: Nationality(p)∝ c，#(LHS)≤3

此查询要求关系满足:当某文献被一组至多三个国家关键字所标注时，至少有 20%的几率使得该文献被国籍不是 c 中国家的人名关键字所标注，并且查询在文集中至少进行 10 次。

```
<Query> ::= Find (<support>/<confidence>) <Pattern>
        Where: <BackgroundConstraint>*
               <KeywordConstraint>*

<support> ::= <integer>
<confidence> ::= (0,1)
<Pattern> ::= <VarList> ⇒ <VarList>
<VarList> ::= <VarExp> | <VarExp>, <VarList>
<VarExp> ::= <Var> : <TypeExp>
<TypeExp> ::= <CategoryType> | <CategoryType>+

<BackgroundConstraint> ::= <Arg> <Operator> <Arg>
<Operator> ::= ∈ | ∉ | ⊆ | ⊄ | = | ≠
<Arg> ::= <Var> | <Keyword> | <Class> | <BgExpression> | LHS | RHS | All
<BgExpression> ::= <BackgroundFunction>(<Arglist>)
<Arglist> ::= <Arg> | <Arg>,<Arglist>

<KeywordConstraint> ::= <#Exp> <CompOperator> <#Exp>
<#Exp> ::= <numeric constant> | #(<category>) | #(LHS) | #(RHS) | #(All)

<CompOperator> ::= > | ⩾ | < | ⩽ | = | ≠
```

图 9.6　FACT 查询语言的 BNF 语法

$$\text{Find: } (10/0.8)\ \text{c1:country+} \Rightarrow \text{c2:country}^{+}$$

$$\text{Where: c1} \subseteq \{\text{Arab League}\}, \text{c2} \subseteq \text{LandBoundaries(c1)}, \#(\text{RHS}) \leqslant 2, \#(\text{country}) \leqslant 5$$

此查询要求关系满足:当某文献被一组 Arab-League 国家关键字所标注时，至少有 80%几率使得该文献被一个或两个其他国家关键字标注——该国家关键字满足与 Arab-League 中——国家交界，并且有 10 次实例使之为真，关系中国家的总数不得超过 5。

9.5　查询操作

我们在背景知识中使用的关系查询操作算法是以 Agrawal 和 Srikant(1994)及 Mannila(1994)描述的算法为基础的。最基本的方法是首先找出所有σ 包含的 X，然后找出满足置信度为γ 的 X\B⇒B 的 X 的子集 B⊆X。经研究可知，σ 包含集合的每一个子集也是σ 包含集的，因此一个集合是σ 包含的当且仅当它的所有子集都是σ 包含的。

候补σ 包含集的建立从单个σ 包容元素开始增加元素，只要能保持集合的σ 包容性，一个新集合将被加到候补σ 包容集中，当且仅当它的所有子集

都包含于候补集合。图 9.7 给出了寻找σ 包含的一般算法。

```
Cand_1 = {(A)| (A) ≥ σ}
i = 1
While Cand_i ≠ Ø do
            Cand_i+1 = {S1 ∪ S2 | S1 ,S2 ∈ Cand_i, |S1 ∪ S2| = i + 1,
                        All subsets of S1 ∪ S2 are in Cand_i}
            i=i+1
      end do
Evaluate ⋃_i Cand_i
```

图 9.7　寻找σ 包含的基础算法

注意，与前面提到的算法相比，此算法仅仅用于检验当所有的候补集产生后，集合是否有必要的支持，因为数据库查询本身就是最耗费时间的工作，所以此步骤只操作一次。而且，为提高效率，我们实际上并不是如图中描述的那样从单个集合开始，而是从大小为 2 的 s 包含集开始，作为文献集关键字计算前的结果可以得到(Feldman 和 Dagan，1995)，过滤出不足维持的候补集，算法从这儿继续。它对开始算法的影响就好像经过一个循环产生由不足支持候补集组成的 Cand2。根据经验，这两种思想减少了必需的计算时间。

当所有的σ 包含集都被找到时，关系查询处理器尝试找出每个σ 包含集 X 的所有子集 B，满足 X\B⇒B 的置信度为γ。最直接扩展该算法运行查询中“Where”约束的方法是先用算法找出所有忽略约束的关系，然后再去掉不满足约束的关系。然而，我们用另一种思路使用σ 包含算法来搜索满足约束的关系，如此可以减少搜索空间，并提高关系产生处理器的效率。

为实现它，我们将可能的关系约束分成两类。第一类包括那些只涉及关系式一边的“简单”约束，如 LHS⊆Arab League，或 Iran∈RHS，或者那些要求整个关系性质的某些约束，如#(All)<5。第二类包括那些涉及关系式两边元素关联的“复杂”约束，如 RHS⊆LandBoundaries(LHS)。我们使用这两类约束来减少所考虑的可能的σ 包含空间。

以一个例子来说明第一类约束如何减少搜索空间，假设约束 LHS⊆Arab League。与其寻找包含所有可能的国家名的σ 包含空间，不如约束后只需考虑那些属于 Arab League 的国家。这当然可以减少σ 包含搜索空间。

第二类也能减少搜索空间，且更为敏感。例如，假设约束 RHS⊆Land

oundaries(LHS)。此约束仅仅当我们知道了关系的左边（LHS）值才能确定。然而，--些情况下，知道了 LHS 的可能值的约束将减少 RHS 的可能数量。例如，如果 LHS 也有约束，如 LHS⊆G7，约束 RHS⊆LandBoundaries(LHS)与约束 RHS⊆LandBoundaries(G7)等价。在σ 包含搜索开始前，这组约束将被发现，减少搜索空间。

当然，在一些情况下，关系的 LHS 将没有任何约束，减少搜索空间亦不可能。为处理这种情况，我们使用σ 包含算法的另一种略为不同的模式，为关系左边产生可能关键字组，并试图将其扩展到整个关系式。将左边全部定义，将可能使用第二类约束过滤出右边的关键字。

图 9.8 表述了找出存在这种约束的关系的算法。文献集作为输入，Ds；K（D）被用来指向标注文献 D 的关键字集合。算法搜索 LHS 中满足简单约束的可能候补集。对于每个结果算法认为能够出现在关系 RHS 的关键字相对存在。最后，它决定哪些关系满足对于给定的约束所创立的置信度和支持度。注意，虽然算法对于它所产生的每个集合 B 的所有子集计数，根据经验我们发现满足所有约束的 B 很小。

```
Use the σ-cover algorithm to create Ls, the set of all left-hand
sides that could satisfy the association-discovery query,
constrained to only consider those keywords satisfying the simple
constraints on the LHS.
For all D ∈ Ds
      For all X ∈ Ls do
            if X ⊆ K(D) then
                  B  = The keywords in K(D)\X  that satisfy the
                  constraints on RHS (either simple constraints or
                  composite constraints) and that appear with the
                  required support.
                  Update co-occurrence counters for X and all
                  subsets of B
            end if
       end do
end do

Form associations based on the accumulated co-occurrence counters.
Remove those associations that do not satisfy the required support
and confidence.
```

图 9.8　关系赋值算法

9.6 关系表达式

即使当数据是中等大小时，关系-寻找方法也常常产生实质数量的结果。关系查询工具的一个问题是帮助用户从所有产生的结果中找出感兴趣的结果。FACT 解决此问题的方法是提供一个浏览工具帮助用户很容易地集中相关结果。

关系浏览工具的主旨是将左边鉴别过的关系式集合在一起，然后打乱它们左边的顺序并显示出来。有更多左边的关系在更详细的关系之前将被列出来。等级树中的顶节点是按照支持所有关系的文献数目递减列出的。

例如，图 9.4 描述了图 9.3 中显示的查询结果的陈述。Iraq 在关系式中出现的次数比其他国家都多，所以它在树中最先被列出来。接着，所有的左边仅仅包括 Iraq 的关系将被列出来，然后我们将看到一个 Iraq 和 Kuwait 的节点，它列出了所有左边包含 Iraq 和 Kuwait 的关系，最后链在左边包含 Iraq 和 Kuwait 和 Saudi Arabia 的关系节点，并结束。在此链结束后，新的 Iraq 的分支将开始，然后开始 Iraq 和 Saudi Arabia 的分支。在此分类结构中每个关系只出现一次。树中的每个终结点也描述出了相关关系的置信度和支持度，就如出现在满足先前和结尾关系的文献关键字一样。

9.7 对新闻数据运用 FACT 系统

为研究寻找文章关系的 FACT 系统的作用，我们将其使用在信息修复技术中应用到的路透社新闻数据上。这有利于好好调查文献集，此工作重要的是，当集中的文献都被一组关键字标注过时，用不着再使用分类器。更特殊的是，我们使用 Reuters-22173 文章分类器检测出 1987 年的路透社新闻文献集。22173 篇文献被 Reuters Ltd.和 Carnegie Group 在 1987 年做过分类索引。David D. Lewis 和 Peter Shoemaker 在 1991 和 1992 年做了进一步格式化和产生数据文件。每篇文献都被路透社职员用涉及 5 个类别（国家，标题，人员，组织和进口交易）的一组 135 个关键字来标记。

我们的目的并不只是找出文章之间的关联，而必须在一定范围内的背景知识中这样做。为了调查 Reuters-22173 集关系挖掘中背景知识的任务，我们使用从 1995 年的 CIA World FactBook 中提取的背景知识，一个有组织的包括

世界上所有国家信息的文献。每个国家的信息被分为 6 类：地理、人物、政府、经济、交通和国防。根据 Reuters-22173 数据的经验，我们从某个国家 C 中提取以下背景信息。

属于：任何 C 所属的组织（例如，G7，Arab League，EC）

交界：和 C 交界的国家

自然资源：C 的自然资源（如天然矿物、煤矿、铜矿、金矿）

出口产品：C 的主要出口产品（如肉、木材、小麦）

出口国：C 的主要出口国

进口产品：C 的主要进口产品（如肉、木材、小麦）

进口国：C 的主要进口国

工业：C 的主要工业产品（如钢、铁、机器、纺织品、化学制品）

农业：C 的主要农业产品（如谷、水果、土豆、牛）

对标注 Reuters-22173 文献集的一组关键字，先定义一元谓词，再定义二元谓词。图 9.9 使用了 FACT 系统的浏览背景知识的功能，展示了部分背景知识。

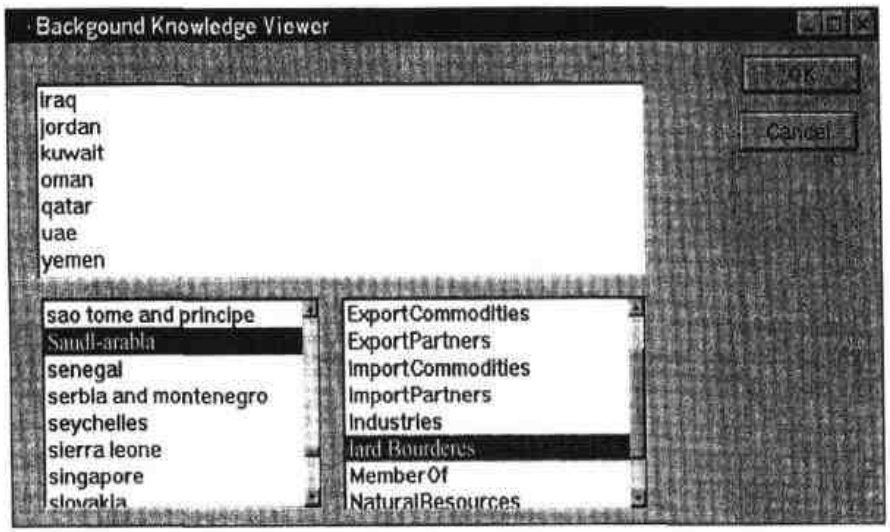


图 9.9　展示与 Saudi Arabia 交界的国家的背景知识浏览器

图 9.3 给出了一个通过 FACT 系统的用户界面，利用背景知识进行路透社数据查询的实例。在界面顶端的编辑框中用户可以看到查询集（转换成“准英语”），并可以通过编辑查询直接转换它。图 9.4 在 FACT 的关系浏览器中显示出了对该查询的关系。

进一步的例子是，给出一个寻找包含 Iran 和某个人的一组国家的所有关

系的查询，FACT 系统将返回结果如下：

{Iran，Nicaragua，USA} ⇒ Reagan 6/1.000

{Iran，USA} ⇒ Reagan 18/0.692

第一个置信度为 100%的关系当给出的路透社数据在“Irangate”时间发生的时段中时有意义。Ronald Reagan 在参议院给出了支持此关系的文献陈述。

寻找一组包括金矿和国家产量的主题之间的所有关系的查询的结果如下。

{gold，copper} ⇒ Canada 5/0.625

{gold，silver} ⇒ USA 12/0.571

{gold，gbond} ⇒ Switzerland 5/1.0

{gold，gbond} ⇒ Belgium 5/1.0

在此例中，我们看到没有一个国家仅仅与金矿紧密联系，虽然我们在关系左边加入了更多的条件使得 FACT 系统也才找出了 4 条关系。例如，在标注“gold”和“gbond”的文献中我们总能找到 Switzerland 和 Belgium。

作为我们算法效率的天然量度，我们用 FACT 系统运行一系列的查询，比较每次查询所找到的关系条数和耗用的 CPU 时间(486/50)。每个查询都由一个或两个查询模板示例创建。第一个模板(T1)包括背景知识约束，要求关系右边是关系左边的 LandBoundaries 函数。

T1 Find: (5/0.1) c1: country^{+} ⇒ c2: country +

Where: c1 ⊆ CountryGroup，　c2 ⊆ LandBoundaries(c1).

产生查询 CountryGroup 由在此模板中定义的其他背景知识国家组织来实现。第二个查询在模板(T2)中以同种方式产生，仅仅是没有 LandBoundaries 约束。

T2 Find: (5/0.1) c1: country+ ⇒ c2: country^{+}

Where: c1 ⊆ CountryGroup.

图 9.10 描出了一幅曲线图，给出了对于所有的国家或组织 FACT 系统计算每条查询所耗费的 CPU 时间。图 9.11 对于该查询各个国家或组织产生关系的数目（在 x 轴列出的组织顺序在两幅图中是一致的，是依照模板 T2 中产生的关系数量来设定的）。这些结果表明，不是减慢关系-挖掘进程，由于背景知识约束的规范实际上提供了我们的挖掘算法所需的信息，所以，反而加快了关系-挖掘进程。使用其他背景知识谓词的其他许多查询模板得到了相似的结果。

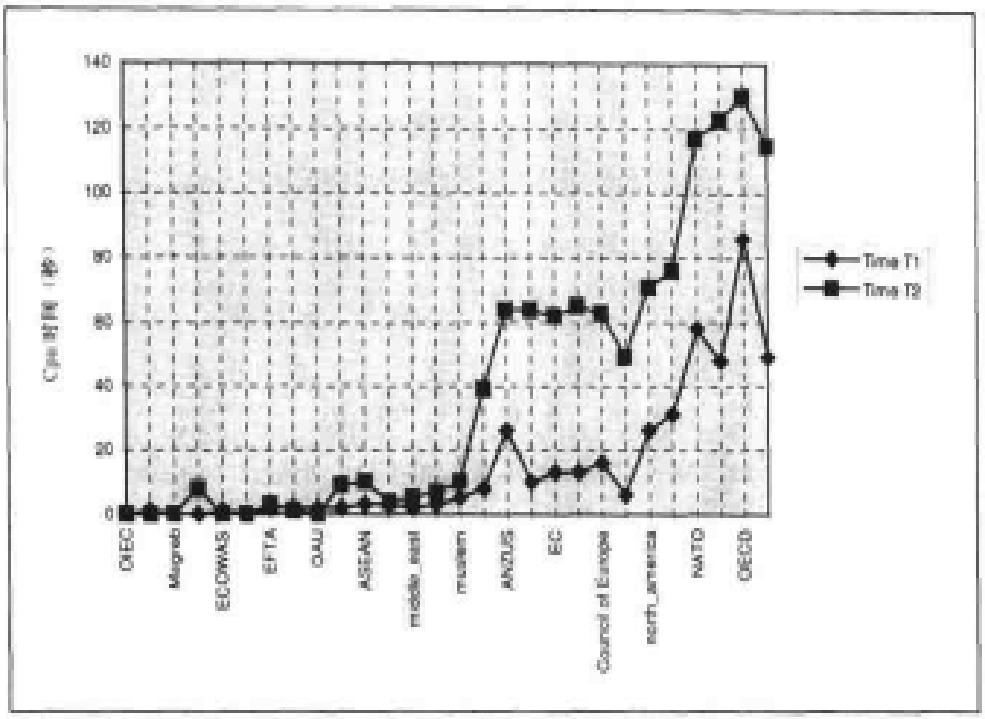


图 9.10　两组查询所耗费 CPU 时间

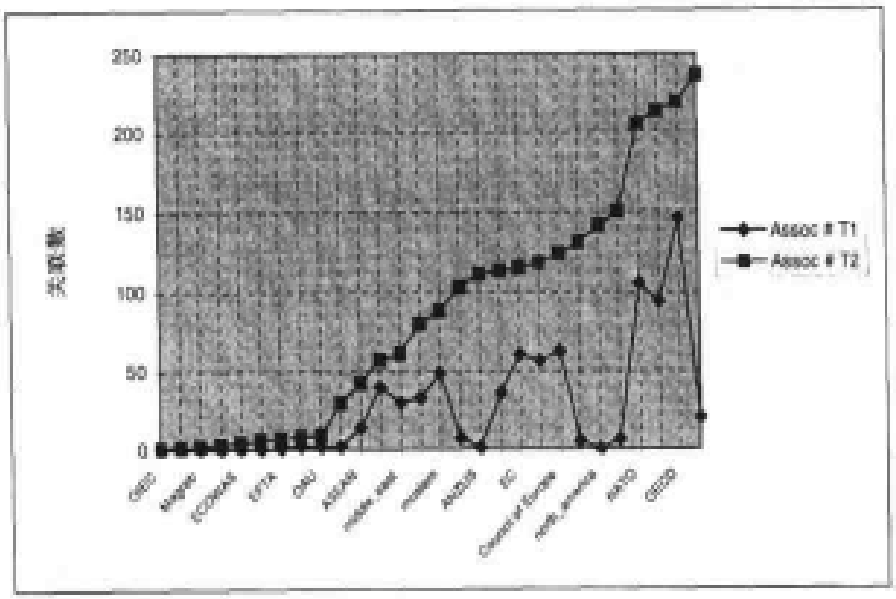


图 9.11　两组查询的所有关系

9.8　总结

此章描述了 FACT 系统在文集中进行的知识挖掘。它解决了从各种各样

的标注文献的关键字中找出关联的问题，并能在关联-挖掘进程中使用关键字信息。

系统在搜索进程中采用集中查询的观点，这样数据可以被其支持的指定的可能结果集所表示，并且用户能够从此结果集中通过查询来采集数据。它介于功能比数据库查询多一点的查询工具（“找出比老板赚得更多的职员”）和不确定目标的查询工具（“找出有趣的事”）之间，而且，FACT 系统不是强迫用户使用那些晦涩难懂的查询语言来定义查询，而是给用户提供了一个简单易用的图形操作界面，这里用户可以很容易地定义查询任务。

FACT 工具同样方便易用，它不像交互式分析工具那样，用户必须忍受使用数据建立可能的感兴趣的模式，也不像自动机学习工具，不能从用户处得到挖掘进程目标所相关的信息，它是介于两者之间的。实际上，用户只需给搜索进程定义一些一般参数，然后让 FACT 系统去搜索能满足用户搜索进程目标的任何感兴趣的模式。

虽然开发是 FACT 工具用来找出文集之间的关系的，但它在有组织的数据上使用甚少。标注文献关键字可用描述实体的二元特征表达，所以 FACT 系统可以找出每个实体的多种特征间的关系。在未来的工作中，我们计划将 FACT 系统的应用扩展到有组织的数据库上去，找出数据库中标注实体的特征之间的关系，从而开发一个寻找基于背景知识的数据间的关系的工具，并研究此基于查询的和 FACT 类似的数据库知识搜索图形工具的优点。

致谢

此项研究由 NSF 授权 IRI-9509819 和 Israeli Ministry of Sciences 授权 8615-1-96。

Ido Dagan，Tomasz Imielinski 和 Willi Kloesgen 对本章的起草和评注提供了宝贵的建议，Amir Zilberstien 为将 CIA World Fact Book 文献转换成 Prolog facts 语言付出了辛勤的劳动，在此作者一并表示感谢。

参考文献

Agrawal R., Mannila H., Srikant R., Toivonen H., and Verkamo I. (1995). Fast

Discovery of Association Rules. In Advances in Knowledge Discovery and Data Mining, Eds. U. Fayyad,G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, pages 307-328, AAAI Press.

Agrawal A., and Srikant R. (1994) Fast algorithms for mining association rules. In Proceedings of the VLDB Conference, Santiago, Chile.

Agrawal A., Imielinski T., and Swami A. (1993). Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD Conference on Management of Data,pages 207-216.

Apte C., Damerau F., and Weiss S. (1994). Towards language independent automated learning of text categorization models. In Proceedings of ACM-SIGIR Conference on Information Retrieval.

Dagan I., Feldman R., and Hirsh H. (1996). Keyword-Based Browsing and Analysis of Large Document Sets. In Proceedings of the 4 Symposium on Document Analysis and Information Retrieval, Las Vegas, Nevada.

Feldman R. (1996). The KDT System - Using Prolog for KDD, In Proceedings of the 4 Conference on Practical Applications of Prolog, London, April 1996.

Feldman R., Dagan I., and Kloesgen W. (1996) Efficient Algorithms for Mining and Manipulating Associations in Texts. In Proceedings of the 13 European Meeting on Cybernetics and Research, Vienna, Austria.

Feldman R. and Dagan I. (1995). KDT - knowledge discovery in texts. In Proceedings of the First International Conference on Knowledge Discovery (KDD-95).

Frakes W. B. and Baeza-Yates. R. (1992). Information Retrieval: Data Structures and Algorithms. Prentice Hall, Englewood Cliffs, NJ.

Imielinski T., Virmani A., and Abdulghani A. (1996). DataMine: Application Programming Interface and Query Language for Database Mining. In Proceedings of the Second International Conference on Knowledge Discovery (KDD-96).

Iwayama M. and Tokunaga T. (1994). A probabilistic model for text categorization based on a single random variable with multiple values. In Proceedings of the 4th Conference on Applied Natural Language Processing.

Klemettinen M., Mannila H., Ronkainen P., Toivonen H., and Verkamo A. (1994). Finding Interesting Rules from Large Sets of Discovered Association Rules. In Proceedings of the 3 International conference on Information and Knowledge Management.

Mannila H., Toivonen H., and Verkamo A. (1994). Efficient Algorithms for Discovering association rules. In KDD-94: AAAI workshop on Knowledge Discovery in Databases, pages 181-192.

Salton G. and McGill. M. J. (1983). Introduction to Modern Information Retrieval. McGraw-Hill, New York.

第 10 章　学习图像中的模式

Ryszard S. Michalski，Azriel Rosenfeld，Zoran Duric，Marcus Maloof 和 i Zhang

摘要

本章介绍图像或图像序列中的模式学习以及利用所获模式解释新图像的问题。本章集中介绍三个领域的应用问题：（1）室外场景的彩色图像语义解释；（2）行李 X 光图像中的引爆雷管检测；（3）视频图像序列中的动作识别。本章讨论了这些领域中的图像形成过程，以及解决这些问题的方法中的表示空间的选择。所给出的结果表明在视觉中应用机器学习的优点。

10.1　导论

本项研究的目标就是视觉系统需要具有学习能力，以便处理那些计算方法无法求解或很难求解的问题。学习能力也会使视觉系统更加容易适应于不同的视觉问题，并在处理各种的感知条件下更加灵活和健壮[MRA94]。

当前有关视觉系统学习的大多数研究着重点在有关神经网络的应用，例如，道路导航[Pom91]和各种图像中对象的检测与识别（可视，SAR 等）[FeB96，RBP96，RBK96]。这些方法的长处包括它们具有普遍性和学习连续转换的能力；而不足之处包括较难结合先验知识（特别是关系知识），较难学习复杂结构知识，学习速度较慢，而且学习得到的知识不易理解[MRA94]。

然而符号学习方法在解决这些问题时则会简单一些，它们大多应用于别的领域，不是计算机视觉领域。而在计算机视觉领域中，它们可能在产生新特征时特别有用，学习视觉表面描述，如纹理；学习复杂形状描述，获取对象的结构性或关系模型，构造或更新模型数据库，场景分割；学习上下文以便算法可以成功应用等[GrP96，MDMR96，MRADMZ96，StF95]。应用符号

方法解决视觉问题仍然是一个只需要少量探索就会产生丰硕成果的研究领域。

多策略学习系统包含多种不同表示方式和/或学习算法。一个实际多策略系统包含神经网络和符号学习。这种方法归纳出规则并用于构造一个神经网络结构。下一步就是完善网络权重。该方法提供一般性和非常快的识别速度[BMP94，MZMB96]。还可以将神经网络应用到更低层次的视觉过程，而将符号学习应用到较高层次的视觉过程。这些方法潜力非常大并具有较好的研究前景。

我们利用符号学习、神经网络和多策略学习方法来解决诸如室外场景理解，混乱环境中对象识别以及视频图像序列中动作识别。以下各节总结了“基于学习的计算机视觉”项目所获结果，该项目是由 George Mason 大学和 Maryland 大学共同进行[MRADMZ96]的。

在 10.2 节中，我们对应用于计算机视觉的机器学习工作进行了回顾。在 10.3 节中，我们介绍了室外场景彩色图像概念性分割问题的求解工作，为此，我们利用多层次图像采样和转换[MIST]方法，有关该方法的详细描述请参见文献[MZMB96]。在 10.4 节中，我们介绍了从行李 X 光图像中检测引爆雷管的方法，有关该方法的详细描述请参见文献[MaM96，MDMR96]。在 10.5 节中，我们介绍了从其运动中识别一个对象问题的求解，有关技术细节请参见文献[DFR96，DRR96]。

10.2　计算机视觉中机器学习的研究工作

Michalski[Mic72，Mic73]研究了符号 AQ 规则学习如何应用于纹理识别或简单结构的识别问题。这些学术论文提出了多层次逻辑模板（MLT）方法，该方法中，窗口操作扫描一幅图像并从中抽取出局部特征。这些特征将用于学习描述纹理的规则（或简单结构），然后这些规则将用于纹理（或简单结构）的识别。

Shepherd[She83]将实例编码成特征向量和学习获得决策树，主要解决工业检查问题，特别是巧克力形状的分类。对决策树 k 最近邻（k-nn）和最小距离的分类器之间分类的准确率进行了比较。这些分类器实验结果类似，其中基于最小距离的分类器具有最高分类准确率——82%。

Channic[Cha89]利用卷积操作（与原来的特征抽取中的窗口操作一起）对MLT方法进行了扩展。利用AQ学习系统，Channic研究了从薄板对象超声波图像进行增量和迭代学习的情况。

除了利用特征向量表示实例样本之外，Connell和Brady[CoB87]从锤子和商业飞机垂直视图的图像类中学习获得广义语义网络。通过一个视觉系统产生训练实例，该系统利用灰度图像作为输入，然后产生相应对象的语义网络。由Winston[Win84]的ANALOGY程序修改得到的一个学习系统，可以通过泛化训练样本进行学习。学习系统进行了扩展，以便能够析取概念以及仅从正例中学习。可以利用这些泛化表示对未知对象进行分类。

Cromwell和Kak[Crk91]沿着Shepherd的思路，利用特征向量来描述形状。利用一个符号归纳方法（根据Michalski的方法[Mic80]）学习获得电气部件形状。据报告，他们的方法对测试数据可达72%准确率，但却没有与其他学习方法进行比较。

Pachowicz和Bala[PaB91]，与Michalski[Mic72，Mic73]和Channic[Cha89]一样，也利用MLT方法，但增加了纹理特征抽取时Laws的Mask集合（进行了修改）。他们还利用了符号数据中处理噪声的技术。这些技术包括通过删截规则以及消去弱规则和再学习，来优化所学习获得的符号描述[MMHL86]。Bala，Michalski和Wnek介绍了能够学习大量类别数据的PRAX方法[BMW92，BMW93]。

Segen[Seg94]利用一个综合形状表示方法，由一个层次图组成，考虑了高曲率的局部特征以及这些局部特征之间的角度和距离。这种表示在平面旋转和转换中都会保持不变。形状为手势的轮廓。Segen系统在实时情况下运行，并应用于飞机模拟和图形编辑程序的控制。错误率在5%到10%之间，但是大多数错误是未知的而不是误分类。

Cho和Dunn[ChD94]描述了一个新学习算法以学习获得形状。该算法记住了性质列表并随着训练更新相关权重。遗忘机制消去无用性质列表。形状则通过一系列线段来描述。利用这些线段的定位，可以计算局部空间测量值并形成一个形状的性质列表。系统被用于分类工具与手势，并取得了92%和96%的预测准确率。

Dutta和Bhanu[DuB94]提出了一个基于CAD的三维识别系统，系统利用遗传算法来优化分割块参数，并给出了有关室内和室外运动序列的定量实验

结果，实验中的系统从地图图像的灰度和深度中来识别（图像的楔状地形（交通枢）和外壳）。

Sung 和 Poggio [SuP94] 应用于人脸的自动识别。利用复杂情景下的非封闭人脸对一种基于示例学习的方法进行了测试。人脸之间空间采用一些“脸”和“非脸”的模式原型来表示。对于图像中的每个位置，通过本图像和每个原型之间的比较计算出来两值距离，用训练得到的分类器来确定一个人脸是否存在。作者指出人脸距离的度量对系统的性能至关重要。

Zheng 和 Bhanu[ZhB96] 验证了 Hebbian 学习算法是如何提高图像阈值算法性能的，这种阈值算法是用于自动目标检测和识别的。而定性的结论就是对于 SAR 和 FLIR 图像，自适应阈值算法比分类阈值算法要优异。

Rowley 等人[RBK96]提出了一个基于神经网络的检测系统，通过使用一个视网膜神经网络来检测小窗口下的图像，并确定人脸是否存在。训练时采用 bootstrap 方法以便能够将错误的检测加到训练集中，因此可以手工选择“非脸”训练实例来减小训练难度。实验结果表明这种算法在检测率和误测率方面有很好的性能。

Romano 等人[RBP96] 提出了一个实时人脸校验系统。实验表明基于模板的简单相关策略对于大多数应用（涉及到图像中快速校验所识别的新图像）而言，已经是足够了。作者建议这种自动实时的人脸校验技术可以应用到诸如自动安全系统这样的人机界面应用中。该技术已被成功地集成到屏幕加锁应用中，它通过完成人脸的识别来作为系统的密码认证（作为口令的补充），以便确定是否容许进入工作站。

MLT 方法[MIC72, MIC73]最近已扩展到多层次图像采样与转换（MIST）。MIST 已被应用于解决各种各样的问题，包括自然场景的分割[MZMB96]和 x 光图像中的引爆雷管的识别[MDMR96]。为了对自然场景进行分类，比较了三种学习算法：AQ15c[WKBM95]、后传神经网络[Zur92]和 AQ-NN[BMP94]。

AQ-NN 是一个多策略的学习算法，它采用了两种不同的表达方式和学习策略，特别是，AQ 学习算法用于从训练样本学习获得基于属性的决策规则。这样，这些决策规则用于构造一个神经网络结构。然后利用后传算法对 AQ 归纳出来的规则进行更进一步的优化。在这样的系统中，与传统神经网络学习相比，学习时间和识别速度，在预测准确率这一方面得以改进的同时，大幅减少。为学习类别诸如地面、树木、天空、色彩和密度，需要利用卷积运算

从用户指定的训练区域抽取出特征，然后将这些样本提交给学习系统，并由此归纳出一个类别描述。单独使用 AQ15c，就取得 94%的预测准确率，而 AQ-NN 和一个标准的神经网络就可以获得近 100%的预测准确率。AQ-NN 的训练时间要比标准 NN 的训练时间少两个数量级。

10.3 室外场景彩色图像的语义解释

MIST 方法（多层次图像采样与转换）提供了将机器学习应用于计算机视觉的环境。可以通过应用到语义解释自然场景中来说明这个方法。这里介绍的实验使用了三个学习程序：AQ15c（从实例中学习决策规则）、NN（神经网络学习）、AQ-NN（结合符号和神经网络方法的多策略学习）。

下面给出的结果展示了这些程序的性能，在预测准确率、训练时间、识别时间、归纳出描述的复杂度等方面，在所选择自然场景解释问题中比较这些学习程序的性能。MIST 方法在这些试验中的表现很出色。从总体上说，AQ-NN 在试验中表现得最有前景。

本节将简要介绍 MIST 方法，并将通过它在自然场景解释中的应用对其进行阐述。正如[FiS88，StF91]所指出的，自然场景的语义解释和自然对象的识别是尚未解决的具有挑战性的视觉问题之一，而 MIST 方法提供了解决这些问题的一种新方法。

10.3.1 MIST 方法

MIST 方法在两种模式下工作：学习模式和解释模式。在学习模式下，系统建立或者更新图像知识库（IKB），其中包括类别描述以及与图像解释相关的背景知识等。可视化目录中的一种描述（或模型）是从训练者提供的实例中通过归纳推理而产生的。分类描述被设计为定义图像转换操作序列的过程。

在解释模式下，将学习过（或预定义）的图像转换过程应用于给定图像，生成一个带有注解符号的图像（简称 ASI）。在一个 ASI 图像中，利用指明类别的注解符号（例如，颜色）来标记对应初始图像中所识别出的类别的区域，并连接至注解（包括类别的附加信息，诸如识别的确定程度、类别的性质、与其他类别关系等）。（虽然均是独立开发，ASI 中的 MIST 概念与 ALISA 系统[HoB94]中类别概念相似），下面的章节将详细介绍这两种模式。

10.3.1.1　学习模式

如图 10.1 所示的这种模式按四阶段来运行：LP1——描述空间的产生和背景知识描述；LP2——事件产生；LP3——学习或者完善；LP4——图像的解释和评估。这四个阶段可以循环重复，建立不同层次的图像。

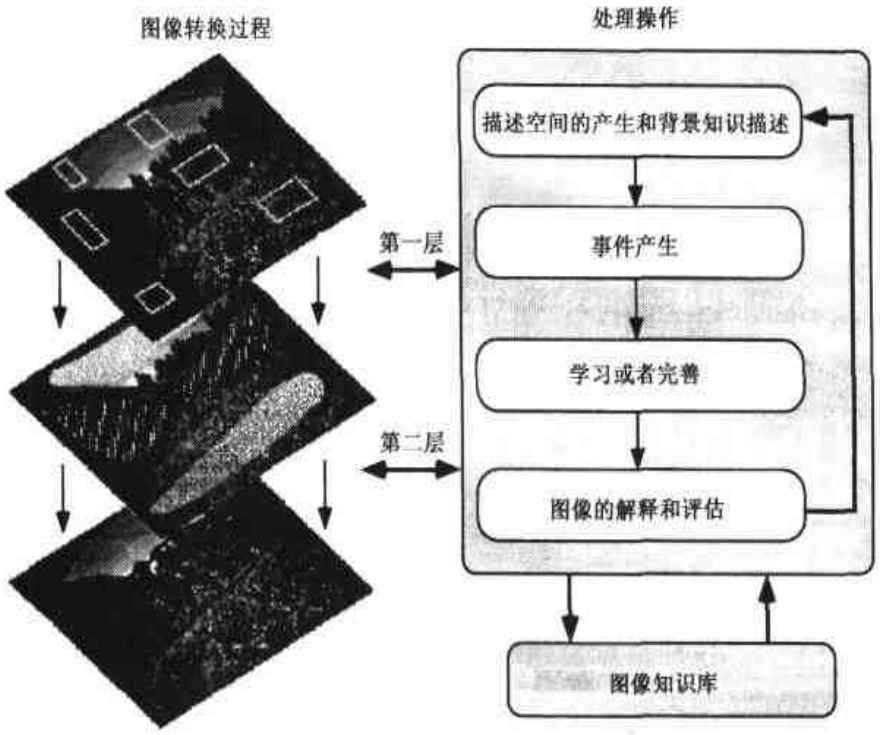


图 10.1　MIST 学习模式

LP1：描述空间的产生和背景知识描述

一个训练者分配类别名称赋给图像中的区域，这些区域包含所要学习的对象。这些区域被分割成训练和测试区域。要学习的对象以不同方式出现（通过改变感知条件），由此系统能够学习一种在转换中类别保持不变的描述。训练者同时还要定义初始描述空间，也就是初始属性和/或要在图像实例中测量的项，并指定它们的值（测量单位）和类型。这个阶段还包括了对图像体积的优化，也就是根据问题需要，降低图像分辨率和色彩等级（图像中色度和饱和度）。训练者或许需要定义描述空间的约束条件，初始识别规则，表示描述的可能形式（如合取规则、DNF、神经网结构等）。属性/项的测量过程将从预定义集合中选择。

LP2：事件产生

利用所选定的过程，系统将为各个区域生成初始训练实例（“训练事件”）。区域可以是全部的或者有选择的采样块。

LP3：学习或者完善

系统利用一个选定的机器学习程序来处理训练实例，产生一个类别描述。目前有下列程序可以使用：AQ15c 从实例中学习获得的决策规则；NN，后向传播的神经网络学习；AQ-NN，结合 AQ 规则学习和神经网络方法的多策略系统。

LP4：图像的解释和评估

所获得的描述被应用于测试区域以产生带注解标记的图像（Annotated Symbolic Image，ASI）。在一个 ASI 中，相应特定类别的区域用符号（数字、颜色等）来表示这些类别。这些区域还将连接至包含有关类别的附加信息，如识别的确定度、类别的性质与其他类别关系的文字上。所产生描述的质量可通过将 ASI 与初始图像中的测试区域相比较来确定。根据这些结果，系统或许停止，或许执行一个新的学习过程（循环），其中 ASI 就是输入（MIST 中“多层次”就是这个意思）。如果产生的描述无需改进，过程就终止。当获得的符号图像“足够接近”目标图像标记时，就会出现这种情况。完全对象的描述就是图像转换序列（由每次迭代所获得的描述来定义）以生成最终的 ASI。学习错误可通过比较目标标记（由训练者给出）和学习出标记来计算获得。

10.3.1.2 解释模式

在这个模式下，系统利用来自图像知识库的描述来语义地解释一个新图像。为实现这点，系统执行一系列操作（由描述所定义），将给定图像转换为一个 ASI。ASI 中的一个给定像素通过给一个单独事件或者一个事件的采样并以大多数投票方式将被赋予一个类别。在 ASI 中，不同的类别用不同颜色和/或材质来标注。ASI 中最简单的标注形式就是表示一个给定类别的 ASI 像素点的关联可信度。

10.3.2 实现和实验结果

目前的 MIST 方法可采用下列学习系统来实现：

- 符号规则学习程序 AQ15c[MMHL86，WKBM95]
- 结合决策规则学习和神经网络学习的多策略学习系统 AQ-NN[BMP94]
- 结合决策规则学习和遗传算法的多策略学习系统 AQ-GA[MBP93]
- 基于类别相似度的学习以建立大量类别的描述学习（PRAX）[BMW92，BMW93]

MIST 的一个早期版本被用于学习表面类别的描述[MBP93]。该描述的核心就是决策规则形式，它由归纳学习程序 AQ15[MMHL86]所确定，并用 VL_1（可变值逻辑系统）逻辑风格语言所表示[Mic73]。这些决策规则能以并行的或顺序的方式应用于一个图像。

MIST 方法的一个简单版本被用于语义解释室外场景，它利用多种学习方法。实验中，我们使用了一组图像集合来表达科罗拉多 Aspen 附近的山峦场景（如图 10.2 所示）。

图 10.2　实验中采用的典型的自然场景

学习过程的输入就是训练图像，其中选择要学习的视觉类别的实例已被训练者标记，例如树、天空、地面、道路和草地。我们利用不同的特性集进行实验，包括不同描述空间，不同感知情况下所获图像，不同训练区域的大小和不同的训练来源与测试图像实例（同一个图像区域上的不同部分，同一个图像的不同区域，以及不同的图像）。

在这里所介绍的实验中，描述空间可由属性诸如色度、饱和度、密度、水平和垂直渐变、高频点、水平和垂直 V 形状及拉普拉斯操作等定义。这些

属性可利用 5×5 窗口操作（采样大小）并扫描整个训练区域来计算。特性值向量构成了训练事件，这里采用三种学习方法：AQ15c，AQ-NN，NN。采用了三种不同大小的训练区域：10×10，20×20，40×40。校验方法采用了 Hold-out 方法，随机选择 60%的样本用于训练，40%用于测试。

图 10.3 介绍了一个训练图像实例，并通过将学习获得的单层图像应用到整个图像（利用多数投票评估方式）所获得的一个 ASI。从 10.3（b）中可以看出，大多数图像的区域被正确解释，尽管系统从相对较小的训练区域中进行学习（图 10.3（a））。这个实验中，AQ-NN 产生了一个稍微小一点的神经网络，解释时间也下降到比 NN 方法的少 50%的程度。

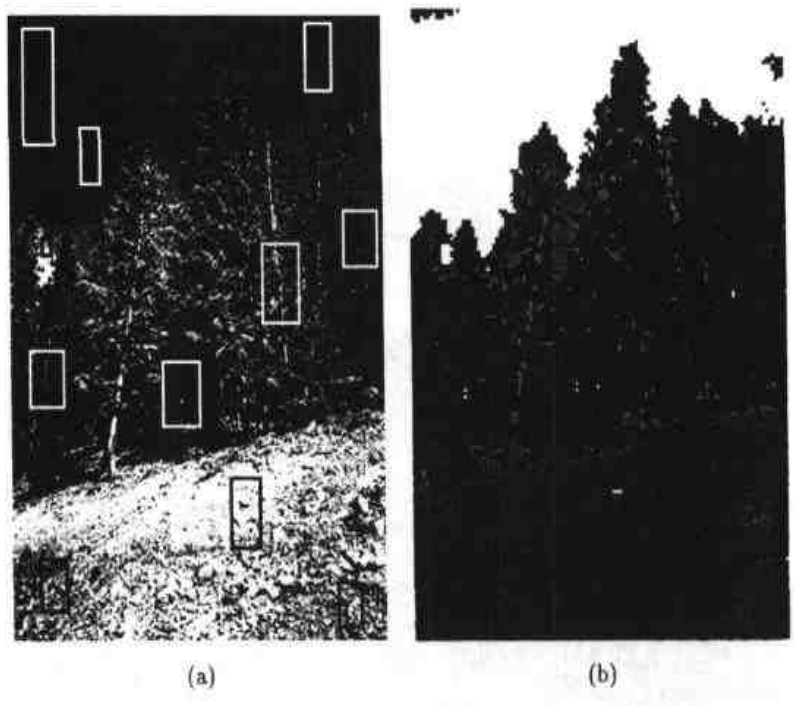


图 10.3　基于从所指示区域学习获得的规则进行图像解释的例子

（a）拥有天空、树木、地面的训练区域的图像（b）利用多数投票方式得到的 ASI

表 10.1 给出了利用不同学习程序对仅有一层图像进行转换所得到的结果。在这个实验中，每个类别的训练区域为 10×10。当训练区域扩大至 20×20 的时候，训练时间显著增加，但是图像区域解释的正确性基本上不变。

表 10.1　10×10 训练区域

学习方法	训练时间	识别时间	准确度(%)
AQ15c	0.43s	1.000s	94.00
AQ-NN	10.93s	0.016s	99.98
NN	4.38s	0.033s	99.97

表 10.1 学习解释图 10.3（a）图像的结果总结。161 个训练事件和 150 个测试事件选择来自 10×10 训练区域所获得的计算数据。

我们同时也利用基于数据的构造方法（AQ17-DCI）进行了这个实验。结果产生一些新属性，但是也给出了可比较的结果[BWMK93]。

10.4　检查行李 X 光图像中的引爆雷管

本节介绍识别 X 光图像中引爆雷管的方法的有关研究。该问题是这样一类问题的一个具体代表，视觉系统必须检查一系列图像以确定已知的对象。遗憾的是，对象的已知往往对于问题的解决毫无用处。如果没有已知对象的标准化，尝试给这些对象进行几何建模是不实际的。常常限制一类对象的就是功能性[FrN71，StB91a，RDR95]。在获得图像特征和对象功能之间关系方面，学习是很有用的[WCHBS95]。

我们的主要目标就是研究视觉和学习如何结合来发现引爆雷管以及包含在引爆雷管中的对象。在已有的研究中[MaM94，MaM96]，用学习来获得引爆雷管的描述。利用简单的分割技术来从背景中分离对象，并利用密度和几何特征来加以表示。

这里我们所介绍的工作是，对引爆雷管的功能性质进行分析以指导要学习的表示空间的设计，它结合了密度和形状特征。实验结果表明,归纳学习系统有能力获得图像特征和对象功能之间的关系。

这里的研究提供了研究视觉和学习过程互相作用的一个机会[MRA94],特别当与学习对象功能相关的时候。可以在机场行李的安全检查中应用一个检查引爆雷管的视觉系统。

10.4.1　预备知识

这里，我们回顾一下图像的形成过程和 X 射线图的成像模式。

典型 X 射线图像系统由一个 X 射线管（光子源）、反散射装置和接受器组成[Dan88]。射线管发射的光子进入对象并且可能被散射、吸收或者无影响地穿过。接受器接收主要的光子而形成了图像，但是散射的光子会形成背景信号（如噪音），它可能会影响图像的对比度。大多数情况下，放置在对象与图

像接受器之间的反散射装置可以消除大多数散射光子。

下面是图像处理的一个简单的数学模型。考虑一个单色 X 光源，它发射能量为 E 的光子，并距被检查对象（行李）足够远，所以可以被看成是平行发射（如图 10.4 所示）。入射光子束平行于 Z 轴，图像被记录在 XY 平面。假设所有到达的光子被吸收并且接受器的响应是线性的，这样，图像可以视为吸收能量的分布。如果单位面积 N 个光子入射到对象上，$I(x,y)\mathrm{d}x\mathrm{d}y$ 是 $\mathrm{d}x\mathrm{d}y$ 区域所吸收的能量，那么有

$$I(x,y)=\exp(-\int\mu(x,y,z)\mathrm{d}z)\cdot N\varepsilon(E,0)E(1+R)$$

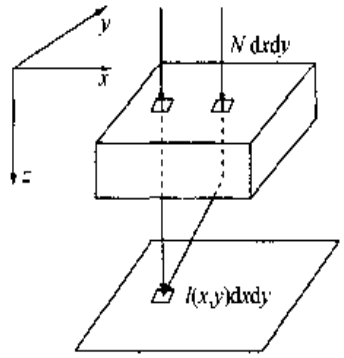


图 10.4　X 光成像的几何模型

这里积分是对主要光子到达点（x,y）的路径上的所有材料进行的。$\mu(x,y,z)$ 是线性衰减系数，$\varepsilon(E,0)$ 为接受器能量吸收系数，E 为光子在发射角度为零时的能量水平，而 R 为散列和主辐射之间的比例（它通常非常小）。

我们假设图像是正投影（如图 10.4 所示），则对象点(X,Y,Z)是这样的点(X,Y)，它满足：

$$x=sX,\ y=sY$$

这里 s 是一个常数。图像在像素点(x,y)的密度由在图像接受器上的像素区域积分 $I(x,y)$而得到。

X 射线图像的密度与 X 射线光子从源到接受器通过对象的数目成比例。由于不同的材质具有不同的穿透特性，因此 X 射线图像的密度依赖于源和接受器间材质的厚度和类型。此外，不能被一种对象吸收的光子可能会被路径上的另一种对象吸收。因此，一个半透明厚的材质可能和一个不透明薄的材质具有同样的效果。

10.4.2　问题描述

尽管引爆雷管是生产出来的对象，但它的制造过程存在巨大变数，以致无法建立一个基于 CAD 的识别系统。然而引爆雷管的共性在于它们的功能。最终，引爆雷管由它们的功能性质而非形状所定义。

一个典型引爆雷管（如图 10.5 所示）由一个填满爆炸物的柱状金属外壳组成。大约在中间，有一个小的重金属爆炸物。最后，导火索从电点火器延伸到另一端。密度最大（X 光最不透明）的部分是重金属爆炸物区域，它大约是中心对称的。导火索同样具有一定的密度，但是它很细。最后，铜或铝的管子里填满了基本对称的爆炸物，它们通常比行李周围的区域要密。

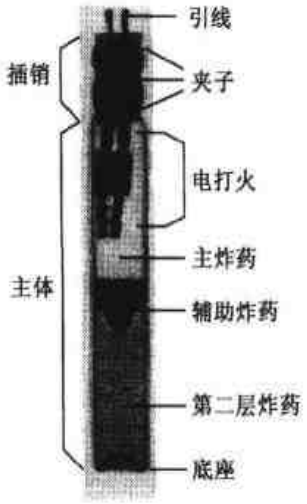


图 10.5　详细的引爆雷管 X 射线图

为了理解引爆雷管的图像，我们考虑一个没有被不透明材料遮蔽的基本引爆雷管。设一个大约圆柱形的引爆雷管长 l，半径为 r，σ 是引爆雷管的轴与图像接受器的夹角。考虑 X 射线通过对象的路径长度为 p，当 $\sigma=0$ 的时候，p 可以从 0 变化到 $2r$。通常，p 要乘上 $\sec\sigma$，但是不能长于 l。从 $I(x,y)$的公式中我们看出，p 增加时，通过的光子呈指数级地减小。从投影方程中可以看出，引爆雷管的图像是矩形，长 $ls\ \sec\sigma$，宽 $2rs$。其密度沿轴最小，外围最大，由此产生一个低对比度的边界。同时，重金属爆炸物图像（如图 10.5 所示）看上去比较小，大约为以引爆雷管轴上对称的斑点。斑点中心基本上是不透明的，故其密度接近零。斑点边界比较亮，但是其密度仍然较低。导火索是强特征，但是在图像上并不清晰可见。（在我们这个例子中，分辨率为 565×340，导火索基本上不可见。目前，我们尝试采用更高的分辨率以便使得导火索更容易被检测到。）

所以，引爆雷管的最强特征就是比较高密度的矩形带状区域，其中心是密度较低的斑点。斑点和带都是沿轴密度最低的，而外围最高。最后，如果引爆雷管被任何对象遮挡，相应图像会比不被遮挡时更黑。

10.4.3 方法和实验结果

我们提出了一种两阶段，从下到上和自上而下的学习方法来识别 X 光图像中的引爆雷管。第一阶段中，可以吸引注意力的低密度斑点，用来产生对象假设。这些斑点与相应次高密度爆炸物一起组成重金属混合物，通常位于引爆雷管的中间（如图 10.5 所示）。

第二阶段中，每个所产生的假设触发一个过程，试图以一个局部模型来模拟斑点周围的带状特征。这些特征对应于引爆雷管中的金属体（参见图 10.5）。局部模型可利用归纳学习系统 AQ15c 获得，它概括了低密度斑点和周围带状区域的密度和几何特征。一个灵活的匹配程序可用来匹配局部模型与图像特征，由此不仅可生成一个对象辨识，而且可以产生辨识中的可信度。

用于实验的 X 光图像包括了各种方向和不同的周围物体（包括衣服、鞋、计算器、笔、电池等）的引爆雷管的行李。这些行李在机场被模拟成像：和射线源平齐，但在图像平面上旋转。然后从 30 个图像中选出了 5 个，它们按照引爆雷管局围物体和位置变化情况具有低和中等复杂度。图 10.6 说明这些实验图像中的一个。

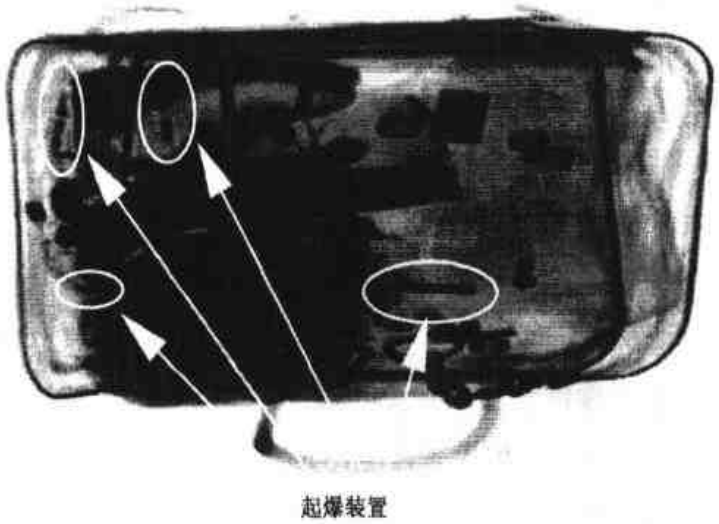


图 10.6 用于实验的图像示例

感兴趣区域被互相确定，其中包括了低密度斑点和带状区域，分别对应引爆雷管的正例和反例。对 64 个所选择的区域，分别计算 27 个几何（如紧密或相近）和基于密度（如最小、最大和平均）的特征，从而产生了 28 个引

爆雷管和 38 个非引爆雷管对象。AQ15c[WKBM95]归纳学习系统被用来学习引爆雷管和非引爆雷管的描述。

利用了 100 次迭代交叉法来对由 AQ15c 归纳获得的描述进行验证。这种验证方法包括 100 次学习和识别过程（运行）。每次运行过程中，抽取出的图像被随机分为训练集和测试集。在从训练集中进行学习后，利用测试集中的例子来对可归纳出的类别描述进行验证。我们根据测试集中的被正确或者错误分类的实例，来计算每次运行过程的预测准确率。实验总体预测准确率则是所有运行过程（计算值）的平均值。这些结果在表 10.2 中进行了总结，并按照 95%的可信度列出了总体和每个类别的预测准确率。

表 10.2　定量实验结果的总结

平均预测准确率		
总体	准确率	83.51±1.3
	差错率	16.49±1.3
起爆装置	准确率	85.82±2.1
	差错率	14.18±2.1
非起爆装置	准确率	81.19±2.4
	差错率	18.81±2.4

作为这种方法的定性说明，学习获得的类别定义也可用于没有看见的图像。AQ15c 从四个图像中得到的训练数据进行学习获得类别描述，并由从第 5 个未用过图像（如图 10.7 所示），中抽取出的数据对其进行测试，对象 1～6 为引爆雷管，7～10 不是。对象 5 是一个引爆雷管却被错误地分类，而其他所有对象均被正确地分类。

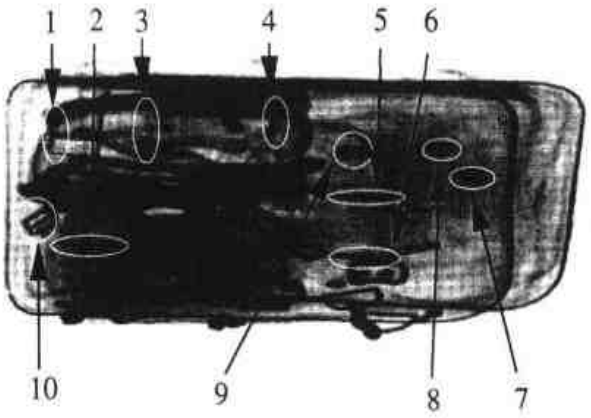


图 10.7　应用学习获得的类别定义测试图像

10.5 视频图像序列中的动作识别

认识对象的作用，常常是与它们进行交互的前提条件。一个对象的作用能够定义为一个对象对一个特别目的作用[BoB94]。

近来有关识别对象作用问题的研究工作已经进行了许多，简单介绍可参见[BoB94]。早期的功能识别工作可参看文献[FrN71，SoB83，WBKL83]。这项研究的目标是确定基于各种特征(诸如外部形状、物理特性和内部原因)对象的功能。最近，Stark 和 Bowyer[StB91a，StB91b，SHGB93]使用这种方法解决了传统的基于模型的对象识别中存在的大多数问题，但这项工作只适用于静态对象，不涉及运动。Green 等在最近的研究[GESB94]中讨论了关节对象的识别，其中利用动作来判定对象是否具有合适的功能性质。以往我们很少把注意力放在如何从对象动作中确定或学习获得对象的功能，事实上，动作中包含了很多如何确定对象功能的信息，特别是对象的速度、加速度和由运动而产生的冲击力都会有力地约束可能的功能。而在其他的功能识别方法中，我们不能把对象和对象的动作分开进行评估，而是要把它们联系起来。这个整体联系包括角色利用对象的特征和角色参照的框架。

在这个章节中，我们将解决以下问题：当某对象用于完成一个任务时，我们怎样依据动作来确定功能？我们解决这个问题的方法是从图像序列中抽取出一系列的动作描述。这些描述与那些已知的从动作和功能映射关系中引伸出的描述进行比较，从而识别出功能。

由于许多对象表现出相似的动作特征，因此我们需要建立一个对象模型，以便依据动作特征来确定对象的功能。所以我们的工作首先就是要把对象分割成基本的组成，然后分析这些部分的运动。

10.5.1 来自动作的功能

10.5.1.1 基本形状和基本运动

根据[Bie85，RRP93，RDR95]，我们把对象当做基本部件的组合体。在最粗糙的层次上，我们考虑四种类型的基本部件：棍、带、盘和斑点。它们在相对维上取值有所不同。如[RDR95]所说，用 a_1, a_2, a_3 来分别表示简单体的长、宽、高。四种类型定义如下。

棍：$a_1 \simeq a_2 << a_3 \vee a_1 \simeq a_3 << a_2 \vee a_2 \simeq a_3 << a_1$

带：$a_1 \neq a_2 \wedge a_2 \neq a_3 \wedge a_1 \neq a_3$

盘：$a_1 \simeq a_2 >> a_3 \vee a_1 \simeq a_3 >> a_2 \vee a_2 \simeq a_3 >> a_1$

斑点：$a_1 \simeq a_2 \simeq a_3$

我们把三维近似相同的对象作为斑点。如果仅有二维非常接近，就分 2 种情况讨论：如果相同二维远远大于另外一维，就称其为盘，否则就是称为棍。若没有任何两维相同，则称其为带。举个例子：一把刀的刀锋没有任何两维相似，这就是带。

基本体可以组合成复合对象。[RDR95]中描述了许多不同定性的基本体组合方法，例如：末端对末端，末端对侧面和末端对边缘等。此外注意到两个基本体表面连接时，我们依据它们连接的角度不同，将连接点分类为垂直、倾斜和附着等情况。还有一点就是描述两个基本体之间连接点在各自表面的位置，譬如是靠近中心、边缘、角落还是仅仅靠近表面的一端。使用定性特征，诸如主轴（直或曲）形状、交叉部分大小（不变或是渐变）等，我们能够更好地描述基本体。

功能的识别是基于与某些动作需求的兼容性的。一些基本的“动作”（支撑、包含等）是自然静态的，但是还有许多的动作涉及到移动的对象。为了说明，可使用一个基本形状，设定动作“切”具有一个锋利的带或盘，这里，锋利边缘与一个表面相交。可以从运动学角度来描述这种相交。相对于其主轴方向，基本体运动方向决定了基本体运动类型，譬如：刺、砍和切割。这些动作都包含了基本体的运动，这些运动具有基本体主轴相互平行或垂直的平移和旋转。这一节，我们将用基本体的运动来推断对象的功能。

10.5.1.2　根据基本体运动推断对象的功能

当观察一个移动对象的时候，我们就想推断出这个对象所完成的功能。这里，对象作为一个基本体集合。譬如，一把刀可以描述为两个基本体的组合：刀把（棍）和刀锋（带）。系统依据这个模型就可以推导出对象的姿势（正如[RDR95]）并把这个信息传递给运动评估模块。在模型和运动评估模块得出的结果的基础上，系统就可推断出对象正在完成的功能。

一个对象所完成的功能不仅与它的运动相关，而且与对象的坐标系统和对象动作承受者（Actee）相关。这个信息传递给我们有关运动方向、对象主轴和承受体表面之间的关系，这些关系可用来推断出想要完成的功能。举个

例子，当刀的运动方向与它的主轴之间是平行的，我们认为这个运动就是“刺”；同样，如果两者之间是相互垂直的，我们知道这是“砍”。上述两种情况中，刀运动方向都和它的承受体之间是垂直的。当我们用刀来切割时，刀前后反复运动的方向与刀的主轴线以及承受体的表面平行。

10.5.2 运动的计算

10.5.2.1 棍和带的运动

对于一个移动的对象 B，我们把它看成是一个惯性的椭圆体。椭圆的中心就是对象 B 的主要质量 C（占 B 的绝大部分）的中心，椭圆的轴线称为**主轴**。我们将坐标系统 $C_{x_1y_1z_1}$ 的轴线与椭圆体联系到一起，从 $C_{x_1y_1z_1}$ 中选择一个与主轴平行的轴。用 $\vec{i}_1$ 表示最长轴线 l_c 方向上的单位向量（这条轴线对应着最小的惯性矩）；用 $\vec{k}_1$ 表示最短轴线方向上的单位向量（这条轴线对应着最大的惯性矩）；用 $\vec{j}_1$ 表示剩下的一个主轴上的单位向量，且此向量是我们选择向量的方向，使得向量（$\vec{i}_1$，$\vec{j}_1$，$\vec{k}_1$）构成一个右手坐标系。

这里我们仅仅考虑那些近似于平面或是笔直的棍和带的对象。对一个平面的带而言，最大惯性矩的轴线与带的平面垂直；如果带很直，最小惯性矩的轴线接近平行于带的中轴线 l_c。同样，笔直的棍的中轴线 l_c 对应于惯性椭圆体的最长主轴，而另外的两条垂直于 l_c 的轴线则可以任意地选取。我们假设棍和带的运动是平面的，而且对观察者而言平面是可见的（“可见性”约束了摄像头的 z 轴与表面法线之间倾斜视角的夹角要≤30°）。当对象是带时，我们假定运动在带的平面上，而平动速度则平行于带的平面，而旋转速度则垂直于对象的平面。对于棍而言，它的连续位置定义了其运动平面；此时，平动速度与运动平面一致，而旋转速度则与运动平面垂直。这种情况下，我们选择最小惯性矩的轴线作为运动平面相垂直的方向。

10.5.2.2 基本体运动计算

现在我们简单地介绍一下计算棍和带基本运动的方法。有关的详细内容请参见文献[DRR96，DFR96]。

我们使用两个关联的垂直坐标系统来描述移动对象 B。一个（$Oxyz$）在空间的位置是固定的（摄像头框架）；另一个（$Cx_1y_1z_1$）固定在移动体 B 上并

随之移动（对象框架）。移动框架在任何时间的位置由以下因素决定：离 C（B 的主要部分）原先位置的距离 $\vec{d}_c=(X_C\ \ Y_C\ \ Z_C)^T$ 以及移动框架和固定框架轴线之间的九个方向余弦。向量组（$\vec{\omega}$，$\vec{T}$）决定了对象 B 的动作，其中 $\vec{\omega}=(A\ \ B\ \ C)^T$ 是移动框架的旋转矢量，移动体 C 的位置矢量由公式 $\dot{\vec{d}}_c=(\dot{X}_c\ \ \dot{Y}_c\ \ \dot{Z}_c)^T\equiv(U\ \ V\ \ W)^T\equiv\vec{T}$ 决定。移动体的旋转矢量 $\vec{\omega}_1=(A_1\ \ B_1\ \ C_1)^T$，并且 $\vec{\omega}=R\vec{\omega}_1$ 或者 $\vec{\omega}_1=R^T\vec{\omega}$，$R$ 是方向余弦的矩阵。从以上我们关于对象 B 运动的假设中，可以得到：$\vec{\omega}_1=C_1\vec{k}_1$ 和 $\vec{T}_1=U_1\vec{i}_1+V_1\vec{j}_1$。

用 f 表示摄像头的焦点长度，用 Z_C 表示 B 的 C 中心的深度。场景的点坐标 (X,Y,Z) 经过弱透视法投影在图像上的坐标 (x,y) 就是：

$$x=\frac{X}{Z_C}f,\ y=\frac{Y}{Z_C}f$$

从资料[DFR96]我们得到图像点 (x,y) 在弱透视作用下的瞬间速度：

$$\dot{x}=\frac{Uf-xW}{Z_C}-C_1(y-y_c)N_Z-C_1[(x-x_c)N_xN_y/N_z+(y-y_c)N_y^2/N_Z]$$

$$\dot{y}=\frac{Vf-yW}{Z_C}+C_1(x-x_c)N_Z+C_1[(x-x_c)N_x^2/N_z+(y-y_c)N_xN_y/N_Z]$$

其中：(x_c,y_c) 是 (X_c,Y_c) 的镜像，$\vec{N}=(N_x\ \ N_y\ \ N_z)^T=R\vec{k}_1$ 是运动平面的法线。这里用到了 $\vec{\omega}_1=R\vec{\omega}$ 的事实。

如果我们在图像点 (x,y) 的法线方向上选取一个单位方向矢量 $\vec{n}_r=n_x\vec{i}+n_y\vec{j}$（通常是图像密度的梯度方向），那么在 (x,y) 点的**法线运动场**就是：$\dot{\vec{r}}_n=(\dot{\vec{r}}\cdot\vec{n}_r)\vec{n}_r$，从而就有：$\dot{\vec{r}}_n=(\dot{x}n_x+\dot{y}n_y)\vec{n}_r$。

若 $I(x,y,t)$ 为图像密度函数。设图像梯度为 ∇I，I 对时间的偏微分为 I_t，由此得到：

$$\vec{u}_n=\frac{-I_t\nabla I}{\|\nabla I\|^2}$$

其中：将 $\vec{u}_n$ 称为**法线流**。

法线流 $\vec{u}_n$ 和法线运动场 $\dot{\vec{r}}_n$ 之间的差异大小与图像梯度的大小成反比。因此只有在 $\|\nabla I\|$ 很大的时候，$\dot{\vec{r}}_n\approx\vec{u}_n$。法线流表达式给出了 3-D 运动与图像偏微分之间的近似关系。[DFR96，DRR96]中使用法线流（一个可观测量）作为投影运动场的近似。这种最小平方估计方法常用来获得 C_1，U/Z_C，V/Z_C 和 W/Z_C 的估计值。在对象可视的情况下，这些估计值可用来获得 U_1/Z_C 和 V_1/Z_C 的值。

10.5.2.3 棍和带运动的描述参数

我们利用三个角度α，β和θ作为棍和带运动的描述参数。中轴方向α可通过以下算法得到：

1. 在计算法线流的时候，（根据它们方向的π模型）将所有边界元素排序（环形）。
2. 找到最短部分γ_1,γ_2，它满足至少包括列表中 3/4 方向。
3. 在上一步所选择的排序子表中，找出中轴方向α。
4. 如果α与原先估计的姿势之间的相差较大，则$\alpha \leftarrow \alpha+\pi$。
5. 将α作为中轴的方向。

估计 C 的图像位置(x_c, y_c)（对象的参考点和其质量中心），作为所有边缘元素坐标的平均值，并以此计算法线流。

定义β为向量$(U_1 \quad V_1 \quad 0)^T$与对应工具坐标系统的$Cx_1$轴之间的夹角。

$$\beta = \arctan\frac{V_1}{U_1}$$

定义θ为随时间变化的旋转角度之和：

$$\theta = \int_0^t C_1 \mathrm{d}t$$

10.5.3 实验

在观察一把刀完成一项任务时的运动的实验中，视觉系统每秒拍摄 25 张图像，拍摄 5 秒，每个实验共获得 125 张图像。图像序列记录完之后，从 125 张图像中选取具有代表性的样本做进一步处理。均匀间隔摆放样本共 11 组，每组包含三张连续的图像（例如：样本 1 和样本 2 在所有实验中分别使用图像 0～2 和 10～12）。从而获得每个实验所使用的 33 张图像。

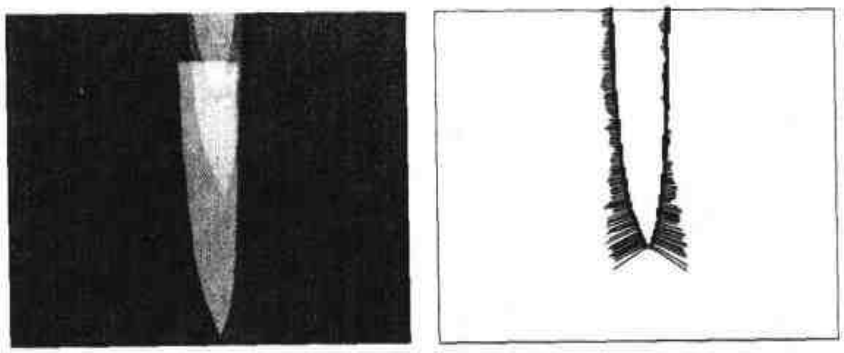
10.5.3.1 实验：刺

刺定义为一把刀的割运动，其中：α（轴l_c在平面$Z=Z_C$上的投影与 Ox 轴之间夹角）接近于$-\pi/2$或是$\pi/2$，β近似为 0，θ近似为一个小常数。

图 10.8 显示了第六个样本中的流向量以及用刀的刺实验中样本 1、样本 6 和样本 11 合成的图像。图 10.9 显示出三元组（α，β，θ）的点随时间（帧数目）变化的轨迹。

如我们所预期的，α 取值非常接近-π/2，β 和θ 近似为 0。一条描述刺的 VL_1 规则（Michalski [Mic72]）如下：

$\langle stabbing \rangle \langle :: [\alpha = -1.55..-1.35]\hat{} [\beta = -1..0.2]\hat{} [\theta = -0.2..0]$



(a)　　(b)

图 10.8　（a）刺的运动（b）刺的流向量

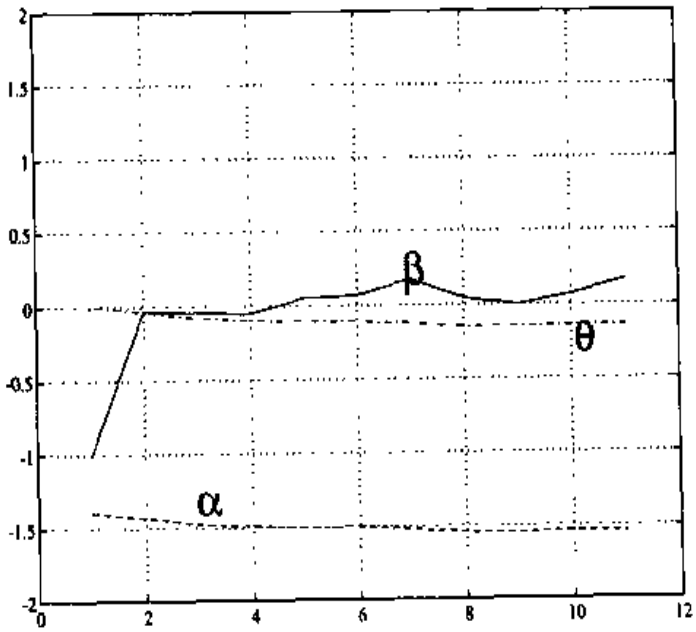


图 10.9　刺中的α，β 和θ 的角度

10.5.3.2　实验：砍

砍定义为一把刀的割运动，其中：α（轴 l_c 在平面 $Z = Z_C$ 上的投影与 Ox 轴之间的夹角）接近于 0 或是π，β 近似为π/2（此时α≈π）或者-π/2（此时α≈0），θ 近似为一个小常数。

图 10.10 显示了第六个样本中的流向量以及用刀砍的实验中样本 1、样本

6 和样本 11 合成的图像。图 10.11 显示了三元组（α，β，θ）随时间（帧数目）变化的点轨迹。如我们所预期的，α 非常接近于 0，β 接近-π/2，θ 近似为 0。一条描述砍的 VL_1 规则（Michalski[Mic72]）如下：

$\langle chopping \rangle \langle :: [\alpha = 0]\text{^}[\beta = -1.6..-1.5]\text{^}[\theta = 0]$

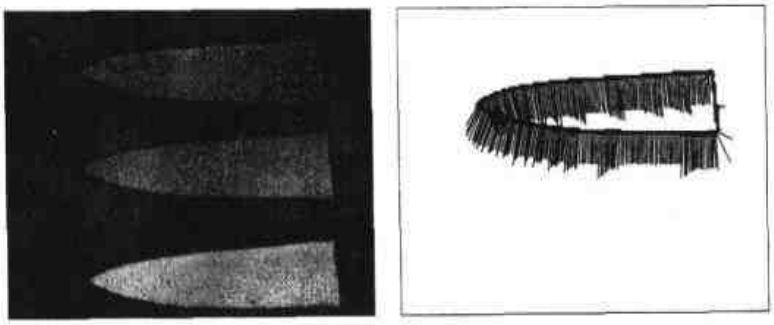


图 10.10 （a）砍的运动 （b）砍的流向量

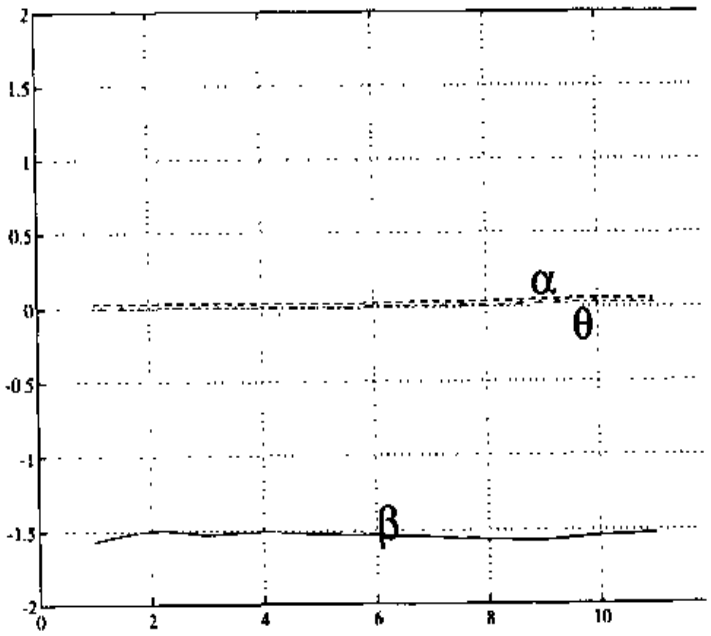


图 10.11 砍中α，β 和θ 的角度

10.5.3.3 实验：切片

切片定义为一把刀的割运动，其中：α 接近于 0（或者小于π/2），β 在 0 和π之间振荡，θ 近似为一个小常数。

图 10.12 显示了第 6 个样本中的流向量以及用切片实验中样本 1、样本 6

和样本 11 合成的图像（从图 10.12（a）左边出现了大量的矢量，是由于从图像中可以看出手的动作）。图 10.13 显示出三元组（α，β，θ）的点随时间（帧数目）变化的轨迹。如我们所预期的，α非常接近于 0，β 在π/2 和$-\frac{3}{2}\pi$之间振荡（注意到这两个近似值之间相差π的整数倍）。一条描述切片的VL_1规则（Michalski [Mic72]）如下：

$$\langle slicing\rangle ::[\alpha=-0.25..0]\wedge[\beta=-2.25..-1.75,0.75..1.25]\wedge[\theta=-0.2..0]\wedge[T_\beta=8..12]$$

这里T_β是β 的一段时间（β 在T_β两个范围内变化）。

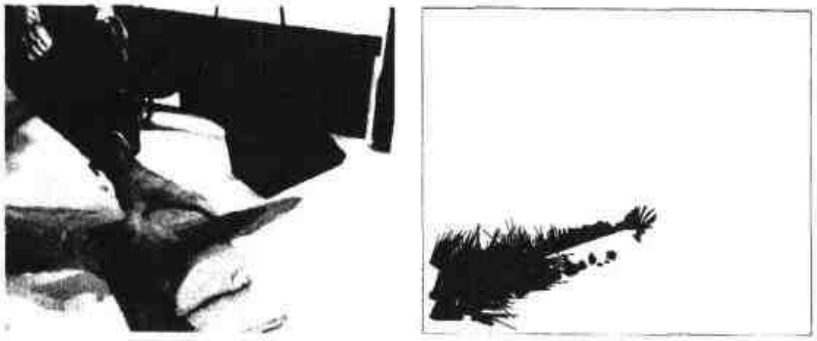


图 10.12　（a）切片的运动　（b）切片的流向量

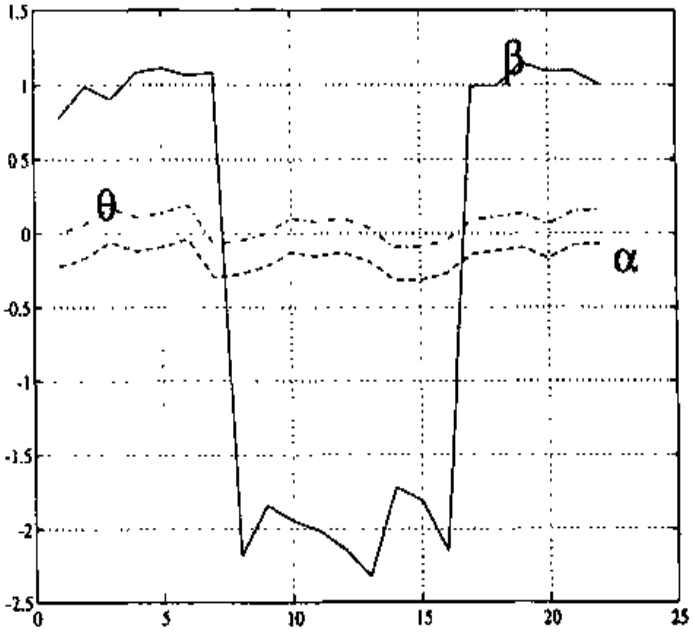


图 10.13　切片中α，β 和θ 的角度

10.6 结论与未来的研究

10.6.1 室外场景彩色图像的语义解释

在 10.3 节中，我们介绍了对于应用机器学习方法解决自然场景解释的问题，MIST 方法是非常有效的。至今所获得的结果都是有希望的，因为它们即使在只进行单层图像转换时，也表现出了较高的预测性能。特别是利用 AQ-NN 方法获得了好的结果。AQ-NN 方法将符号学习与神经网络结合到了一起。

这种方法具有几个重要的优势。它包括以一种统一方式应用和测试不同的学习方法的简易性，实现先进和复杂学习过程的潜力，在学习和解释图像时对背景知识的利用，进行图像学习和解释的合理性以及测试方法的性能的简便性。

当前研究涉及在明显不同的感知条件下，利用不同类型的初始属性和训练与测试（所获）图像区域的方法进行系统性的探索。

10.6.2 行李 X 管图像中的引爆雷管检测

在 10.4 节中，我们介绍了在 X 光图像中识别引爆雷管的研究工作进展。在一个两阶段学习方法的第一个阶段，低密度斑点作为注意力吸引点，这个自底而上的过程伴随一个自顶而下的识别过程，这个过程中，用一个学习得到的局部模型与（由低密度区域包围）带状图像区域进行匹配。利用对引爆雷管功能性质分析来设计学习的表示空间，同时结合密度与几何特征。实验结果表明：可以利用学习来获得对象功能的描述。这在利用几何建模进行对象分类不合适的情况下非常重要。

这一领域的未来研究将涉及对特征抽取和对象功能标示过程更一步的自动化。此外，其他在引爆雷管中提出的功能性质也需要研究探索。一个实例就是脚线存在（参见图 10.5）。遗憾的是，当前图像集并不具有能够检测这类功能性质的分辨率。我们希望获得更多适合进行这类分析的图像。

10.6.3 识别视频图像序列中的动作

从运动所感觉出的功能提供了一种了解由代理使用对象的方法。为此，

我们将对象的形状、运动，以及它与承受者（它作用的对象）间的关系等信息综合起来。假设将对象分解成若干基本体单元，分析一个单元相对于它主轴线的运动。基本体运动（相对主轴线的平移和旋转）在分析中是非常重要的因素。我们使用了一个相对承受者的引用框架，一旦建立起这样的框架，就可以描述一个动作的功能的主要内涵。

几个图像序列可以用来证明我们的方法。在 10.5 节所介绍的三个序列中，运动被用来区分三种割的动作：刺、砍、切片。在其他序列中，没有在这里介绍[DFR96]，我们利用运动信息来区分同一物体两个不同的功能，如用一个铲子挖掘和打击，用一个扳手锤打和拧紧。

本工作的自然扩展包括对更复杂对象的分析。可以利用部件形状或部件之间的连接方式来表示这种复杂性。一个有趣的领域就是分析带关节的对象。部件之间不同类型的连接限制了部件之间可能的相对运动。钳子和剪刀就是这样的简单例子，它们只有一个简单的带关节连接（在相对部件运动中只有一个自由度）。

10.6.4　在视觉系统中结合学习的优点

我们已经给出了三种示例来说明一个学习系统是如何用于帮助处理视觉问题的，在这些问题中不知道求解算法或者很难获得求解算法。特别是，我们已经研究了符号、神经网络和多策略学习方法来解决这些问题，这包括：室外场景解释，在复杂环境中物体的识别，视频图像序列中动作的识别。第一个问题涉及将一个图像分割为若干区域以对应草地、树木等物体，因为这些目录没有简单的定义，因此也就不能定义能够区分它们的最优算法。其他两个问题涉及对象类别或者动作分类，它们没有简单的几何定义，而是依靠功能来定义的：从行李的 X 光图像中检测引爆雷管，在视频图像序列中识别割的类型（刺、砍、切片）。这些例子中，我们能够设计一种合适的表示空间以使得学习（和识别）变得可行。

致谢

本研究一方面得到了国防高级研究项目处的支持，合同号为

F49620-92-J-0549 和 F49620-95-1-0462，并由空军科学研究局负责管理。本研究还得到了空军科学研究局的支持，合同号为 F49620-93-1-0039 以及国防先进研究项目处的支持，合同号为 N00014-91-J-1854，具体由海军研究局管理；还得到了海军研究局的支持，合同号为 N00014-91-J-1351 以及国家自然科学基金的支持，合同号为 DMI-9496192 和 IRI-9510644。

参考文献

[BMP94] Bala, J.W., Michalski, R.S., and Pachowicz, P.W., "Progress on vision through learning at George Mason University", in Proc. ARPA Image Understanding Workshop,191-207, 1994.

[BMW92] Bala, J., Michalski, R.S., and Wnek, J., "The principal axes method for construc-tive induction", in Proc. International Conference on Machine Learning, D. Sleeman and P. Edwards (Eds.), Aberdeen, Scotland, 1992.

[BMW93] Bala, J., Michalski, R.S., and Wnek, J., "The PRAX approach to learning a large number of texture concepts", in Proc. Machine Learning in Computer Vision: What,Why and How?, AAAI Fall Symposium on Machine Learning in Computer Vision, 1993.

[Bie85] Biederman, I., "Human image understanding: Recent research and a theory", Com-puter Vision, Graphics and Image Processing, 32:29-73, 1985.

[BWMK93] Bloedorn, E., Wnek, J., Michalski, R.S., and Kaufman, K., "AQ17-A multi-strategy learning system: The method and user's guide", Reports of the Machine Learn-ing and Inference Laboratory, MLI 93-12, George Mason University, Fairfax, VA, 1993.

[BoB94] Bogoni, L. and Bajcsy, R., "Active investigation of functionality", in Proc. CVPR Workshop on Visual Behaviors, June 1994.

[BABC84] Brady, M., Agre, P.E., Braunegg, D.J., and Connell, J., II, "The mechanic's mate"in Proc. European Conference on Artificial Intelligence, 79-94, 1984.

[Cha89] Channic, T., "TEXPERT: An application of machine learning to texture recogni-tion", Reports of the Machine Learning and Inference Laboratory,

MLI 89-27, George Mason University, Fairfax, VA, 1989.

[ChD94] Cho, K. and Dunn, S.M., "Learning shape classes", IEEE Transactions on Pattern Analysis and Machine Intelligence, 16:882-888, 1994.

[CrK91] Cromwell, R.L. and Kak, A.C., "Automatic generation of object class descriptions using symbolic learning techniques", in Proc. National Conference on Artificial Intelli-gence, 710-717,1991.

[CoB87] Connell, J.H. and Brady, M., "Generating and generalizing models of visual ob-jects", Artificial Intelligence, 34:159-183, 1987.

[Dan88] Dance, D.R, "Diagnostic radiology with x-rays", in The Physics of Medical Imaging,S. Webb (Ed.), 20-73, IOP Publishing, Philadelphia, PA, 1988.

[DFR96] Duric, Z., Fayman, E., and Rivlin, E., "Function from motion", IEEE Transactions on Pattern Analysis and Machine Intelligence, 579-591, 1996.

[DRR96] Duric, Z., Rivlin, E., and Rosenfeld, A., "Learning an object's function by observing the object in action", in Proc. ARPA Image Understanding Workshop, 1437-1445, 1996.

[DuB94] Dutta, R. and Bhanu, B., "A learning system for consolidated recognition and motion analysis", in Proc. ARPA Image Understanding Workshop, 773-776, 1994.

[Fah88] Fahlman, S.E., "An empirical study of learning speed in back-propagation networks",

第11章　机器学习在音乐研究领域的应用：深入音乐表达现象的经验调查

Gerhard Widmer

摘要

本章主要讨论的是机器学习在研究音调的基本音乐现象时的一个应用。本章还描述了由从音乐家的真实演奏的例子当中得到的富有表现力的乐曲演奏推导出的一般音乐规则的学习算法。鉴于一般的音乐知识在人类学习这项任务中扮演了一个举足轻重的角色，我们提供了两种基于知识的学习办法。在每种办法中，提供给学习器的领域知识都是基于确立的音调音乐理论的。实验结果展示了与纯粹的感应学习相比，每种算法都给学习结果带来了显著的提升。然而，本项目还不只是基本的机器学习研究。因为本项目的基础完全建立在音乐理论上，它还可以被看成是一个对音乐研究科学领域或者音乐学的贡献。本文的结果在该科学学科文献中已得到体现。我们也将在文章中涉及到这些。

11.1　介绍

本章讨论了音调音乐的学习算法，它是机器学习的一种应用，乍看起来颇为不寻常，甚至带有一丝神秘。在这项持续发展数年的科研项目中，我们使用了机器学习的方法去研究作为音乐这门艺术的核心基础的基本音乐技巧，我们称它为**富有表现力的音乐演奏**。我们已经开发了多个学习系统，用来从人类音乐家演奏的实例中得到富有表现力的演奏的一般规律。

项目开始于基于知识学习领域基本的机器学习的研究。起初的目标是调查把领域知识引入学习过程的多种方法和学习这种知识的一般特性及其作

用。选择音乐作为测试领域是因为它可以提供一套具有难度的学习任务，尤其特别的是，音乐理论为领域知识提供了丰富的资源，它通用性强，发展良好，而且远未达到周密完备。随着领域分析的进步和越来越把重点放在领域知识的原理化和拟真的模型上，此项目逐渐变成了一个真正跨学科的尝试。它也开始带来了音乐学上的有意义的成果，并且我们同时也发表出了这个科学学科的著作（见，如，Widmer,1993a,1995a,1995b,1996）。

本章所介绍的内容是机器学习的一个应用，而不是一个实际（例如工业化的）问题，是科学的另一个分支。机器学习作为对其他学科具有贡献的一项技术，其潜力特别是在生化和分子生物学的领域已被一大批科研工作者所证明。本章内容将要展示的是，哪怕是那些更为“非正式”的领域，如音乐，也能从机器学习实验中得到益处。

作为一个跨学科的科研项目，我们的工作追寻着问题的来源，并且产生两个学科都感兴趣的结果。从机器学习的远景看，我们的目标是研究多种不健全的（不严密和不完整的）领域知识，以及使用它来校正学习器的预测的方法。现在将要展示的结果是基于知识学习的两种不同的办法：第一种办法中，提供一个利用明确定性的领域知识来指导泛化搜寻（第 11.4 节）的感应学习算法。第 11.5 节描述了另一种策略，应用领域知识把全部的学习任务转化到更高的抽象层次，在这个层次上相关规则变得更容易清楚显现。特别地，这是一个知识-驱动建设性感应的形式（Wnek,Michalski,1994）。实验结果证明以上两种方法都改善了学习的结果。

从音乐学的角度上看，需要研究的核心问题是音乐知识的概念。相应的问题包括：音乐听众拥有什么样的一般音乐知识？它是怎样形式化的？在这种知识和富有表现力的演奏间有着什么样的关系？音乐章节的什么结构决定和影响了表演的被接受性？我们的观点是机器学习能够在这些问题上提供了新的解决思路，而最终实验的结果也确实验证了这样的观点。当然了，从音乐理论的角度对实验进行全面的介绍和分析超过了本文所要讨论的范围。在第 11.6 和 11.7 节，我们将稍微提一下最为有趣的结果。

本章只能展示我们的项目的大概情况、算法和结果，但我们殷切希望向读者们证明，机器学习可以用于如音乐学的科学领域，此外，我们还想展示基于人工智能的音乐研究的魅力。

11.2 学习对象：富有表现力的音乐演奏

如果完全照乐谱演奏，绝大多数音乐听起来是完全机械，毫无生命力的。富有表现力的演奏不是照本宣科而是对音乐段落“修形”的艺术，它需要在表演期间不断地改变某些音乐段落的参数，例如，快起来或者慢下去，演奏得更响亮或者轻柔，在段落间增加一些小的停顿等等。有许多种的参数维度可以受不同演奏者的影响，其中的一些是受特殊的手段限制的（例如颤音）。在这个项目中，我们集中研究最重要的两个富有表现力的维度：力度变化（响度的变化）和随意增减音符长度或者说是富有表现力的调速（局部节拍的变化）。我们的程序将演示谱写好的音乐音调和在一个钢琴家富有表现力的演奏时记录下的音调。通过这些程序，人们就会学到富有表现力的阐释的一般原理，这将影响到在演奏新的乐章时是更有表现力还是相反。

有时候作曲家在谱子上标上明确的表达标志（例如，在一个音乐段落下的渐强标记），但是更多的情况下富有表现力的形式是不会被明确标注的，需要演奏者根据他（她）个人对音乐的理解来决定如何表现。我们的系统将展示一段没有明确表达标注的音乐符。

巧的是，给学习器的输入是音乐段落（一段音乐符）的乐调，在段落里每个乐调中的音乐符号都与两个数值有关：演奏者（力度变化维度）演奏时音符的确切响度和精确的节拍（实际表演持续时间和乐谱记录持续时间的比）。学习器的任务是制定一个规则，用于决定在演奏一个新给定的乐章时应该奏得多响和多快。这样研究的问题就是一个**数字预测任务**。11.4.3 节将给出一个这类感应问题的基于知识的算法，不过首先让我们来探讨一下音乐理论对于这类问题能给我们提供一些什么。

11.3 背景知识的特性和价值

再次考虑抽象的学习任务：训练样本是乐调，也就是说，音乐符号的序列，其中每个音乐符号都与一定程度上描绘了响度或者局部节拍的准确率的数值相关，把这些都提供给一个音乐家的演奏参考。这些数值可以从一个定义在乐调上的曲线（一个执行曲线，音乐技术术语）看出。任务是去学习“绘制”、“修改”或者至少判断新的乐调上的“敏感”曲线，也就是音乐符号

的新序列。

图 11.1 想要给读者一个这样的直观感受，即对于一个对音乐知识一无所知的学习器而言问题看起来是什么样子的：对于一个基本的学习器，独立的各个乐符只是一些有多种内在特性和属性的普通符号。这些抽象的表示法表明了学习任务的困难。明显可以得出其中一个主要问题是上下文关系：一个孤立的符号并不惟一决定和它相关的数值（曲线的高度）。其实，无论是否仅仅拥有局部或者非局部的上下文影响，我们一点也不清楚相关的上下文到底是什么。

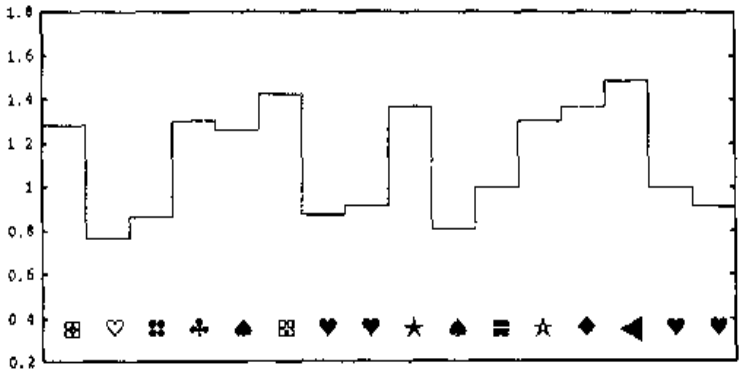


图 11.1　一个抽象的训练实例

实际上，从很少的几个例子就可以看出来，人类（例如音乐学生）可以十分有效率地学会富有表现力的演奏的一般规律。原因当然是，我们作为人类掌握了关于这些符号代表什么含义的附加知识。对我们而言，这是音乐，它给了我们符号的**说明框架**。听众们并不接受一个呈现给他们的不相关符号或者事件的一个简单序列组成的片断，但是他们很快地、主动地用结构术语来理解它。例如，他们把事件流分割成“块”（主题、组、短语等）；他们的直觉听到音乐的韵律结构，也就是说，知道一个规律的强弱节拍的间隔和了解在哪里停顿。线性上升或者下降的乐调曲线常作为一组来听，典型的节律图和其他音乐符号的组合也是这样。对一个人类学习者而言，以上的训练实例就如图 11.2 所示。

同样可以识别更多维的音乐结构，很明显适应相互文化的听众在更高相容的风格上吸取这些构成，而且绝大多数情况下他们并不知道这种风格（例如，见 Deutsch,1982; Sloboda,1985）。所有的这些结构和形式与在一场演奏中所观察到的响度的升降和节拍都相关。这就是听众和音乐家在听或者奏一个

片断时自动运用的音乐知识。

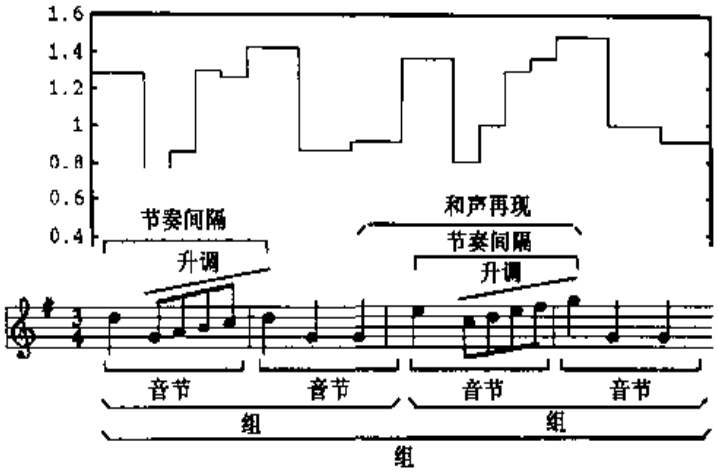


图 11.2　人类学习者所认识到的问题

来自音乐学的结论支持了这个假设。新近的众多研究告诉我们表达不是武断的，而是与听众和演奏者所感受到的音乐结构息息相关。实际上，表现力是演奏者强调或者弱化特定结构的手段，从而使得听众“听到”演奏者所理解的片断。

下面的章节讨论了关于提供学习算法的两种不同的办法，它们都拥有关于音乐结构的普通背景知识及其对于富有表现力的演奏的可能关系。知识本身基于众所周知的两个曲调音乐理论——Lerdahl 和 Jackendoff 的《曲调音乐的产生理论》（1983），Eugene Narmour 的《牵连-实现模型》（1977）。这两种理论都假定了主张可以由人类听众知觉到的多种结构类型。

这也是项目对于音乐学而言变得有趣的地方：通过这些学习系统的实验结果，可以提供实验化的证据，来支持或者反对多方面相关的潜在的音乐理论，并且一般可以帮助我们识别那些音乐的结构，其维数看起来在富有表现力的演奏关系时拥有最大解释性力量。

11.4　方法一：在音乐符号的层次上学习

我们将要研究的第一种办法严格遵循通常认可的**基于知识**或**知识密集**的学习：领域理论（虽然抽象、不完善、部分不一致）（Mitchell 等人,1986）清楚地阐明了关于音乐结构的理解的知识。我们研发了一个名叫 IBL-Smart 的可以利用知识的学习算法。

11.4.1　目标概念

在乐符的层次上进行学习。每个独立的乐符都是一个训练实例，感应规则也指向独立音乐符。目标是去学习一种决定一个片断中每个乐符的响度和节拍准确度的量值的规则。于是有了两个独立的（数字化）学习任务：力度变化和节拍。因此，系统将要学习两套规则。

为了使问题被一个符号的和基于知识的感应算法所理解，我们把它分解成符号分类和数字预报两个任务。在两个表达维度上，我们分辨了两类音乐符号，一类是与演奏曲线的上升（相对于前面的乐符）相联系的，另一类是与曲线的下降相关。在力度变化维度中，两类相应的音乐术语是渐强（响度的提高）和渐弱（响度的减弱），而在节拍维度中，则是渐快（节拍的加快，例如提速）和渐缓（减速）。这些都是普通的音乐概念。

11.4.2　定性的领域理论

音乐实例开始时仅仅通过独立乐符（例如，定调（调高）、持续时间、段落的相应位置）的本质特性和一些相邻乐符（两个乐符的间隔，间隔的趋向）之间的简单关系来描述。正如我们在 11.3 节中想要表示的那样，对于有效率的学习，独立乐符是很难满足的。我们还需要相关音乐结构的知识。我们已经设计了结构化的符号领域理论，它反映了我们所认为的普通听众拥有的一般音乐感觉。这种知识大部分是近似的和不确定的，但可使用理论公式来阐明它。图 11.3 绘出了该理论一般结构的草图。进一步的详细论述可以参见 Widmer(1993a,1995a)。

该模型由两个主要部分组成。下面的部分，称为**结构听觉模型**，如图 11.3 所示，它基于一组程序，这组程序执行给定曲调的结构化分析并且明确使用了多种我们认为普通听众所能感知的乐调的注解。这部分模型基于众所周知的由 Lerdahl 和 Jackendoff(1983)及 Narmour(1977)提出的音乐理论。本质上，其目标是构建具有重要价值的、更高层次上的且能捕捉音乐上下文的描述符号。于是这就可以导出感应过程。

该理论（定性的从属网络）的上面部分描述了我们关于音乐的结构化样式和适当的有表现力的执行决定（例如，符号目标概念）之间可能关系的直觉判断。在用于基于说明的泛化（EBG）(Mitchell 等人,1986)时，它与“古典

的”领域理论非常相似。在更多的可操作的、特定的情况下，它是一个与不可操作谓语（包括目标概念）关联的说明层次。然而，这些说明还可以描述不同力度和特征之间的关系。

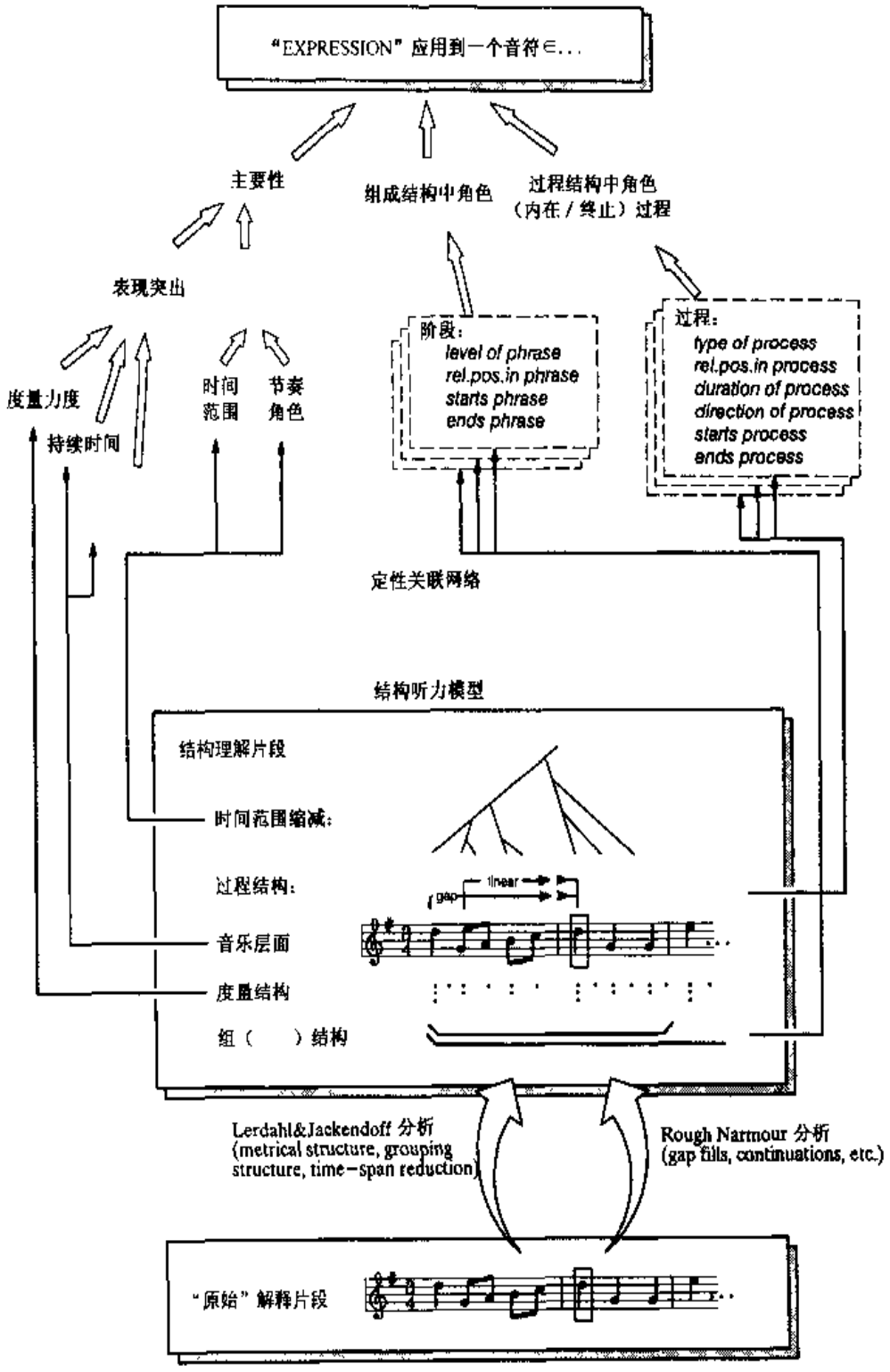


图 11.3 定性的背景模型的结构

严格（推论）的规则

在 EBG 中，领域理论包含一些严格推演的规则，其形式是 Q:-p1,p2,…，满足条件（p1,p2,…），从而使一些谓语（不可操作）Q 为真值。

定向的定性依赖关系

q+(A,B)的形式可以解释为："属性 A 和 B 值成一定的正比例关系"或者"在其他情况相同的情况下，A 的高（或低）值趋向于生成 B 的高（或低）值"。类似地可以定义负的依赖关系 q-(A,B)。

很明显，这一类的知识在精确度和逻辑上比严格的规则要弱一些。它不允许推导的推理。已经有文献提出许多相似的知识条款，其中有 Michalski(1983)的 M-和 R-描述符和 Collins 与 Michalski(1989)的正负依赖理论。

不定向的定性依赖关系

说明 depends_on(Q,[P_1,P_2,…])表示了一个在系列谓词 P_i 和（不可更改）谓词 Q 之间的不确定、不定向的关系。基本上，它表示 Q 的值（或真值）依赖于 P_i 的值（或真值），但是我们并不知道定义这种依赖关系的确切函数表达。Russell(1989)和 Bergadano 等人(1989)已经描述过相似类型的一般知识条款。他们习惯于在为准则求精的搜索中将学习器聚焦于相关的谓语或属性的集合。

在图 11.3 中，绝大多数箭头表示了定性依赖的关系。比如说，下面的这段理论的最高层次上的说明与一些抽象的音乐概念的响度变化现象相关：

```
depends_on( crescendo(Note,X),
(importance(Note,I),
 goal_directedness(Note,G),
 closure(Note,C))).
```

"声音渐强是否应用于一个乐符（如果是，值为 *X*）依赖于：在其他情况中，乐符的重要性 *I*，曲调目标定向度 *G* 和曲调的结束度 *C*。"

重要性、目标定向、结束等抽象概念又与低层次的音乐事件相联系，所有的方法都指向了训练事例的表面特征，例如：

```
q+( metrical_strength(Note,X), stability(Note,Y)).
q+( harmonic_stability(Note,X), stability(Note,Y)).
```

"乐符的稳定性感知度 *Y* 肯定与乐符的测量力度 *X* 有比例关系"等

测量力 metrical_strength 是一个数字，协和稳定性 harmonic_stability

是符号特性（离散的、排序的定性值）。这两者都定义为可操作和可由领域理论，即结构听觉模型的较低的部分计算出来。

11.4.3 IBL-SMART 学习算法

为了本项目（Widmer,1993b）的目的，我们研发了称为 IBL-Smart 的基于知识的学习算法。**IBL-S$_{MART}$ 算法**依照 11.4.1 节所定义的学习任务两部分结构，由两个主要组件（见图 11.4）组成：一个**符号学习组件**可以学习划分符号目标概念（例如声音渐强和渐弱），并且可以从定性模型的形式利用领域知识；一个**基于实例的组件**用精确的数字特征值存储实例而且可以通过在已有实例中添加数字的方式对一些新的乐符预测目标值。两者的连接关系如下：由符号组件学习到的每个规则（链接假设）描述了实例的一个子集；这些被假定为描述了目标概念（例如，声音渐强情况的特殊类型）的子类。把某一规则限定的所有实例提交给基于实例的学习器，并均存储在一个单独的实例空间中。这样，在新的段落中为一些新的乐符预测目标值时，应该对违背符号规则的乐符进行匹配，而且仅仅使用那些为能被乐符接受的相关规则预测而设的数值实例空间（添加表格）。

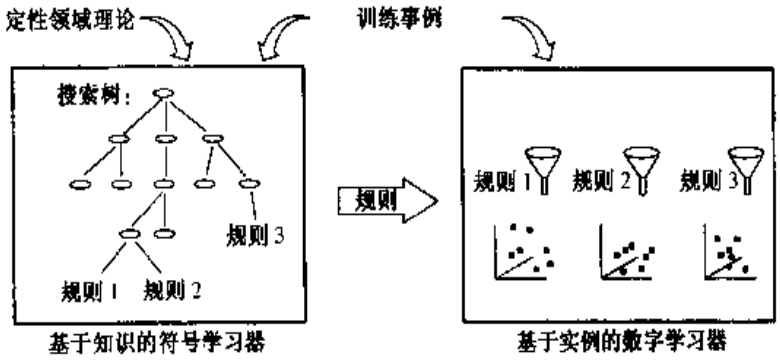


图 11.4 IBL-S$_{MART}$ 中的符号和数字学习的集成

IBL-S$_{MART}$ 符号组件是一个无增量的识别算法，它从分离的一般形式（DNF）学习分类规则。我们特别设计成它能够使用不精确的、定性的背景知识，就如包含在我们的领域理论中的一样。算法开始于对目标概念的不可操作定义（如声音渐强），并且生成一个启发式的首步最优的搜索树来执行单步方式从上到下的搜索。表达（树的节点）通过操作化不可操作的预测，或者通过诱导增加新条件来区别正反例的办法来完善。当仅仅包含正的训练

实例时，一个节点变成一个叶节点，随后，它在最后的 DNF 假设中表现为一个结合点（规则）。当覆盖了一定比例的正例后，一次搜索结束。

把一个不可操作的预测简化到更多基本单元的实施步骤是基于在领域理论中给出的规则或者从属说明的。在严格规则实例中，这与基于解释的泛化（Mitchell 等人,1986）有同样的办法。在定性依赖的实例中，我们说，q+(A,B) 这一可操作步骤在于用 A 替换不可操作的预测 B。算法对于在被当前节点覆盖的正实例中出现的所有值通过用 $A(X,a_i)$替换 B(X,.)来生成成功节点。这样，这些节点中的哪一个最有扩充希望，最有可能进一步扩张就由一个启发式的评估函数所确定，此函数指导搜索。函数考虑了经验主义的评估，如当前节点的“纯度”，也就是，被表达式所覆盖的正负实例的比例，也考虑到了语义标准，就如被包含在使用操作的观察中被一些领域理论中的定性说明所要求的比例关系的特征值的度。

在同时考虑似真的推论依赖的评估和实例覆盖数量的信息时，搜索启发由弱的、不精确的背景知识和训练数据中的经验主义的信息组成，它们生成了两个假定，在数据最大的允许下趋向相应的背景知识；一旦数据与知识矛盾，则数据高于一切背景知识。有关搜索策略更详细的描述可以参见 Widmer(1993b)。

11.4.4　实验

我们使用不同音乐时期和风格的片断（Bach 小步舞曲、Chopin 华尔兹、甚至标准爵士乐）来测试系统。这里我们提供两个典型的结果。

图 11.5 展示了举世闻名的摘自 J.S.Bach 的 Notenbüchlein für Anna Magdalena Bach 的三个小步舞曲的开始部分。每个段落都由两个部分组成。所有段落的第二部分都用于训练：由作者使用电子琴演奏，并且使用 MIDI 格式来记录。在学习后，我们用同一个段落的第一部分来测试系统。应用这种办法，我们在训练数据中把这些拥有一致风格（从同一时期出来并且有相似特征的三个段落；虽然不同，但测试数据和训练数据出自同一段落）的变奏结合起来。

训练输入由 212 个例子（乐符）组成，其中 79 个是声音渐强的例子，120 个是声音渐弱的例子（余下的都是中性的演奏）。系统学习了 14 种规则，相应地，14 个插入表格描绘了声音渐强的情况，同时还有 15 个声音渐弱的规则。相当数量的实例都被不只一个规则覆盖。

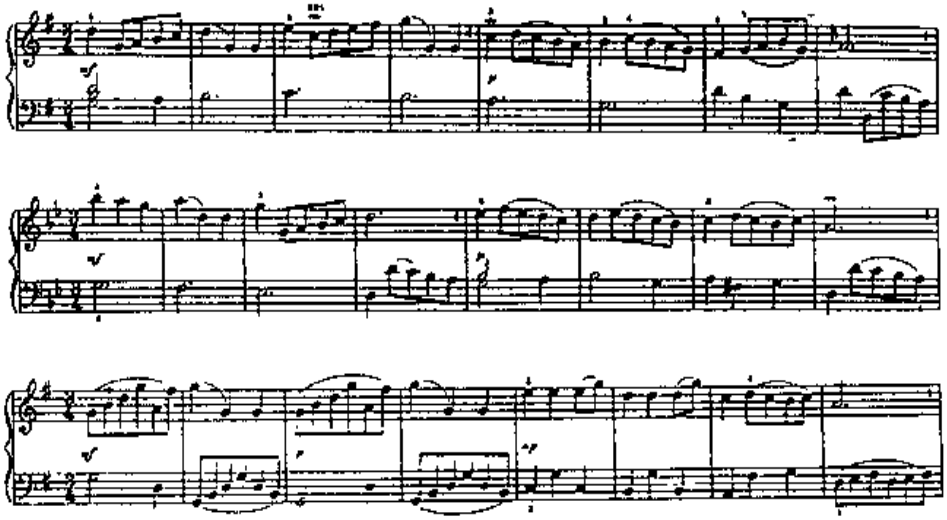


图 11.5　J.S.Bach 的三个小步舞曲的开始部分

把这些规则应用到新的段落来产生富有表现力的演奏。因为没有精确的标准来判定一些演奏是对还是错，所以这些品质是难于评估的。判断正确性是欣赏演奏的事情。遗憾的是，我们无法为本文配属一个唱片来使读者分享我们的成果。图 11.6 描述了一个训练段落的一部分（由作者以 G 大调演奏的第一个小步舞曲的第二部分），图 11.7 展示了在学习之后由系统为测试段落（同一小步舞曲的第一部分）生成的演奏。图片用演奏的独立的乐符来画出了相应的响度，1.0 水平反映了平均响度。

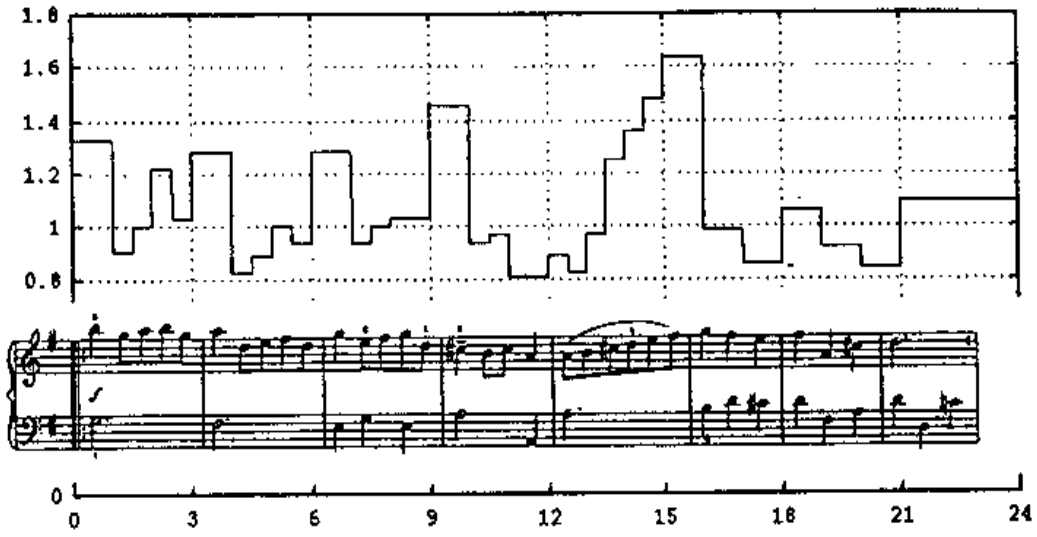


图 11.6　由老师演奏的训练段落的开始

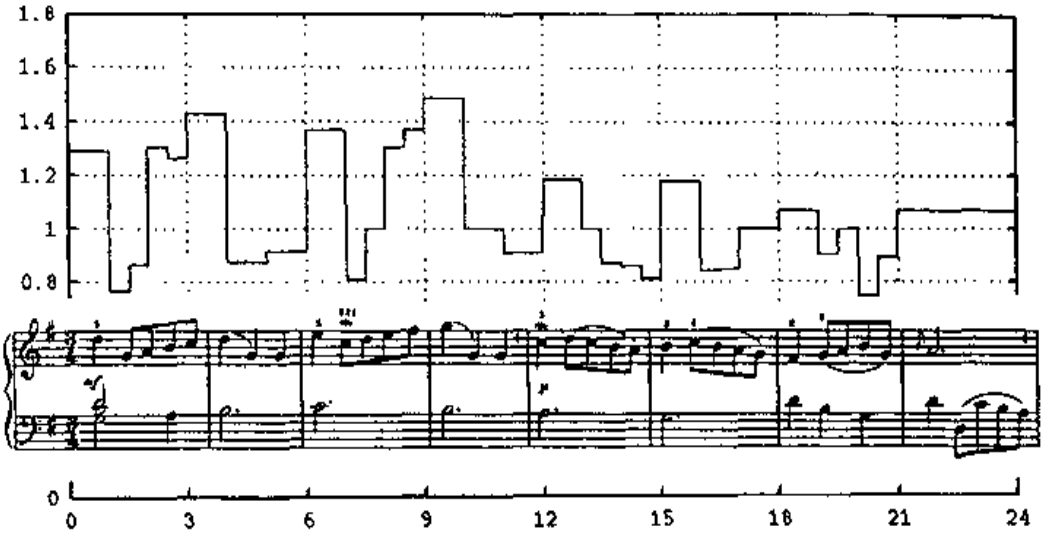


图 11.7　学习后的学习者演奏的测试段落的开始部分

熟悉标准音乐符号的读者也许会赞赏人类教师和学习器对于相近类型的乐句能演奏得如此相近。音符，比如说，声音渐强通过步进运动逐行提高，声音渐弱模式用四分之一音符量度。注意在开始测量时重音（响亮的音符）的一致模式。在所给训练数据有限时，泛化实现的程序是很显著的。此外，对本次实验中学习到的符号规则的检查揭示出系统重新发现了一些数年前被音乐理论家所制定的一些表达原则（见 11.7 节）。

当我们在没有领域理论的情况下进行同样的实验时，我们对音乐背景知识的重要性留下了深刻印象。在没有领域模型的情况下，IBL-S_{MART} 算法简化成了一个纯粹经验化的辨别算法。

图 11.8 展示了在使用这种办法学习 Bach 的小步舞曲时系统对同一个测试段落的执行。从有知识的情况学习（如图 11.7 所示）到无知识的情况学习（如图 11.8 所示）执行结果明显地变坏。由约束系统提供的变化则有着混合的品质。有时候（如在声音渐弱模式评估为 4 和 5 时），它们确实有意义，在别的情况下（例如，在评估为 1,3,6 时重音在最后乐符）系统的决策与音乐感受是相反的。很明显，领域理论对于成功学习的贡献意义深远，特别是在提供的训练例子数量较少就如当前情况一样时。

除了这样的定性评估以外，我们还执行了一些定性的测量来不容置疑地确定基于知识的办法的益处。关于这个问题，第 11.7.1 节将有更多的讨论。

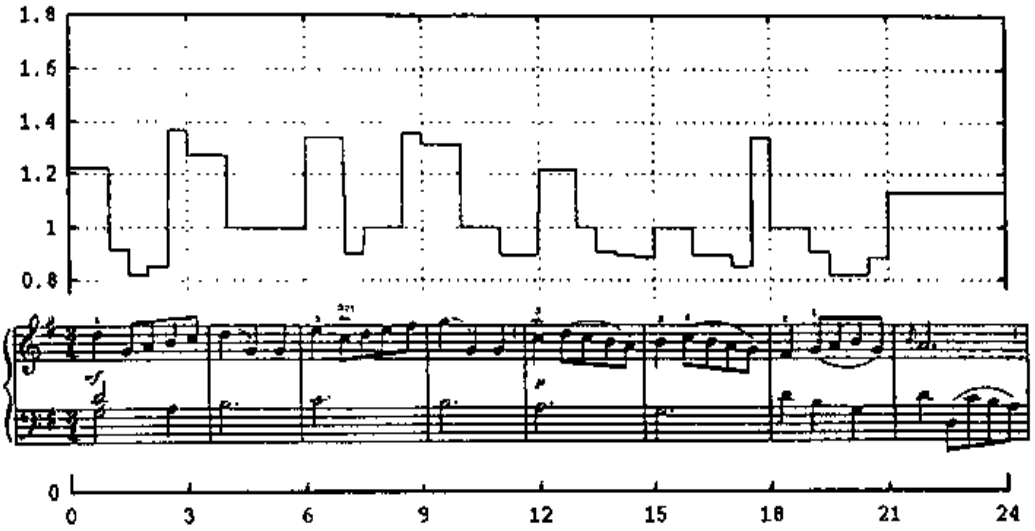


图 11.8　无领域理论的情况下学习后所要演奏的测试段落的开始部分

11.5　方法二：在结构层次上学习

尽管第一种方法有许多令人鼓舞的结果，可最终我们还是清楚地认识到，从音乐的观点来看音符层次并非是真正适当的。首先，尽管由系统生成的演奏在大部分音乐上有感觉，但它们还是缺少一定的流畅，无法同时拥有局部与全局形式的感觉。第二，依照单独的乐符考虑和决定一个纯粹局部的层次，对于演奏者在心理上是难以接受的。而且，他们宁可在更高层次的抽象形式上，比如乐句等等，来理解音乐。最终，如 Sloboda(1985) 所观察到的，表达是一个多层次的现象：富有表现力的形状，就像音乐结构，均以复杂层次形式出现。局部表达模式会被嵌入更大的模式中（例如，在全部的声音渐强中的装饰的修正）。音乐表达感觉的形式化将反映这一点。

因此，我们开发了另一种办法，放弃乐符层次而是尽量直接在音乐结构的层次上学习表达规则。这种办法的本质就是一个基于知识的抽象策略，即把训练样本和整体学习问题转化到音乐的拟真的抽象层次上。于是诱导出的表达规则就涉及到抽象层次。

问题的转化有两个阶段。系统首先对给出的乐调进行音乐上的分析。分析程序，基于由 Lerdahl 和 Jackendoff(1983)和 Narmour（1977）提出的理论摘选的部分，在可能被听众或者音乐家听成的个体或者“块”的乐调中识别不同的结构。结果是对识别好的结构中的乐调进行的详细注释。用 Bach 的小步

舞曲的一段摘录来进行这一步的结果如图 11.9 所示。在这里识别的知觉块中，有四个听起来是节奏元件的**评估**，三个在两个不同层次上听起来是乐调元件或者乐句的**组**，两个**线性上升的乐调线条**，两个称为**节奏缝隙填充**的节奏模块（引自 Narmour 理论的概念），还有其他一些元素。需要注意的是这些音乐结构拥有广泛的变化范围——有些仅由两个或三个音符组成，其他的横越好几个评估标准。由于训练实例是由这种结构定义的，系统将会学习在多层次上识别和应用表达。

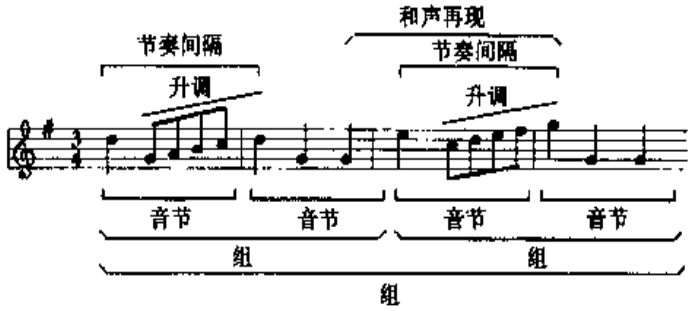


图 11.9　部分 Bach 的小步舞曲的结构整合

在第二步中，识别学习器所需的抽象目标概念。系统试图在给出的与这些结构相关的表达（动态的和节拍的）曲线中找出第一典型的**形状**。从曲线中可以识别出的第一典型形状只是个大致的趋势。系统可以被划分为五种形状：平滑水平（在由结构覆盖的时间间隔中，曲线没有可见的上升或者下降趋势）、上**升**（在时间间隔中从头至尾有一个上升的趋势）、下降、**升**降式（首先上**升**到某一个点，随后下降）和降升式（先下降，再**上升**）。系统选择这些形式是为了最小化实际曲线和由直线定义的理想形状的偏差。

与 Bach 的例子(得自于作者的一次演奏)相关的力度变化曲线，如图 11.10 所示，很好地说明了这一步。我们来看一看如图 11.9 所示的结构中的两个：在评估标准 1-2 中上升乐调线与上升相关，因此，在这一部分**的**记录中曲线展示了一个清晰的上升（下降）趋势。同时在评估标准 3-4 中‘节奏缝隙填充’模块也与降升式结合了起来。

变化乐句的结果传给了 IBL-S_{MART}。每一对〈音乐结构，表达形式〉都是一个训练实例。每个这样的例子都可进一步描述为一个定量形状（精确的响度/节拍值，相对于段落的平均响度和**节**拍，曲线的极点）和阐述，依照音乐理论特**性**、结构和包含在其中的乐符（例如，乐符持续时间、协和函数、测量力……）。因为不同类型的音乐结构被不同的特性分开描述，学习任务就

被分割成了相互关联的子任务。对于每一个音乐结构类型表达形式的预测也都被分别学习。

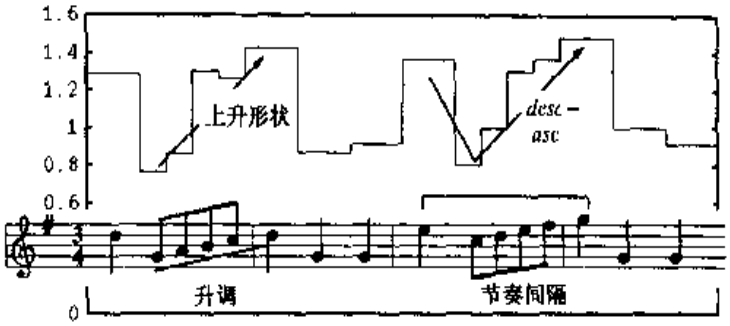


图 11.10　Bach 记录中发现的表示形状中的两个

于是 IBL-S_{MART} 的输出就成为了一组决策规则，给定音乐结构的类型和描述，则可确定将提供它什么类型的表达形式，怎样应用下降，加快等。

给新问题提供学习规则是很直截了当的：在对新的有表现力演奏的段落（乐调）给出了评价时，系统先重新通过音乐分析来把它转化为抽象结构层次。对于发现的每一个音乐结构，学习规则要考虑提出适当的表达曲线形状（为力度变化和增减音符长度）。我们使用与适配规则相关联的添加表格来计算曲线形状的精确数字细节。从一个平滑曲线开始，到整个片断（例如，对于所有乐符都相同的响度和节拍），把有表现力的曲线形状按从短到长的顺序应用于排好序的段落。有表现力的曲线形状通过分别计算力度变化和增减声音长度的值覆盖了已经提供的曲线形状。结果是对一个片断的富有表现力的诠释，它对于局部和全局表达模式一视同仁，因而把微观的和宏观的结构结合了起来。

需要注意的是在这个抽象的学习算法中，独立乐符的层次被完全弃之不用。只考虑那些与整个音乐结构相关的富有表现力的形式，对于单独乐符则从不进行这样的调查。因此，我们的第 2 种办法并不包容办法一（乐符层次的学习）。我们实验的结果（在下面简要地描述其中的一部分）显示出抽象思路（在细节层次上丢失信息）潜在的危险是会被增长的噪音忍耐度。同时，抽象办法一般地也会带来更多结构合理和平衡的结果。

11.5.1　实验

下面是使用 Frédéric Chopin 的华尔兹进行实验的一些结果。训练片断是

从三个华尔兹 Op.64 第二，Op.69 第二，Op.70 第三中摘选出的五个，由作者用电子琴演奏并以 MIDI 格式记录。因此通过让系统演奏其他从 Chopin 的华尔兹中摘录的片段来测试学习的结果。

根据响度和节拍的变化，在学习了五个训练片断后，系统演奏华尔兹 Op.18 的开始部分时的表现如图 11.11 所示。同样地，值 1.0 表示响度和节拍的平均值，值大意味着乐符被分别演奏得响亮或者快速。作者加入了一些箭头表示演奏过程中的不同结构规律性。注意到写好的音乐评估包括作曲家加入的一些明确的表达记号（例如，cresc(声音渐强)、sf(加重)或者 p(钢琴)和为大规模声音渐强和声音渐弱所要求的图画符号），系统并不知道这些，我们只给系统提供乐符。

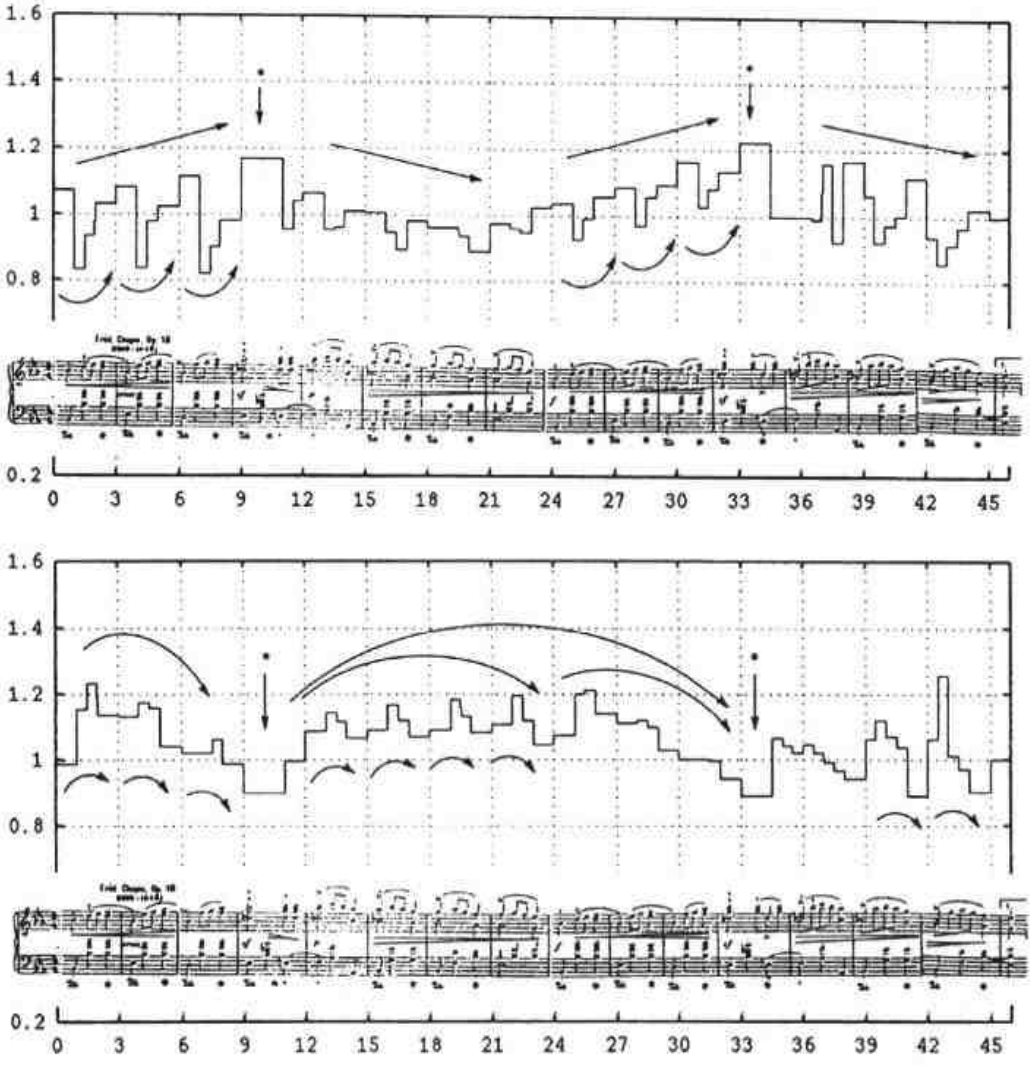


图 11.11　学习器演奏的华尔兹 Op.18 的 E 大调：响度（上）和节拍（下）

在略低的结构层次上，最明显的情况是独立评估标准的分段，它生成了

截然不同的华尔兹“感觉”：在力度变化方面，每个小节的第一个和节律性最强的那一个乐符几乎都通过比该小节的剩余乐符演奏得响亮的方式来强调。同时对旋律的附加考虑（比如升或降旋律线）决定了每个小节的细节结构。在节奏变化方面，小节是这样演奏的：第一个乐符比紧随其后的几个乐符演奏得要稍微长一些，然后在往小节结尾演奏时逐渐放慢。

最令人激动的是系统变奏和 Chopin 的明确的表示标记（对系统而言这些是不可见的）有着非常近似的关系。受过训练能读懂音乐符号的读者也许会十分高兴地看到：在第五小节中力度变化曲线与各种渐强、渐弱符号以及 P（钢琴）命令是符合得这么好。有两个音符是 Chopin 认为尤其值得加强的，同时也得到了 *sf* 的明确标注：在从头数的第 4 和第 12 小节的 Bb's。在这点上我们做得足够漂亮，系统得到了同样的结论并且极大地强调了它们：演奏它们时比段落中的其他任何乐符都要响和长。相应的两个音符在图 11.11 中都用带星号的箭头标出了。

为了比较，由作者演奏同一片段所得的独立纪录中的力度变化曲线如图 11.12 所示。宏观上说两者有很多相似点。然而，令人尴尬的是，作者本人的演奏还不够好：在很多绝妙的细节上不规则而且控制得不够好（由于电子琴键盘所限及作者本人并不高超的钢琴弹奏技巧所致），需要注意的是系统学习的训练片断也并不比这个质量好。系统从坏的实例中学习而生成流畅的演奏部分因为：这是从一个演奏样本的低层次细节上得到的富有表现力的表达形式的抽象。

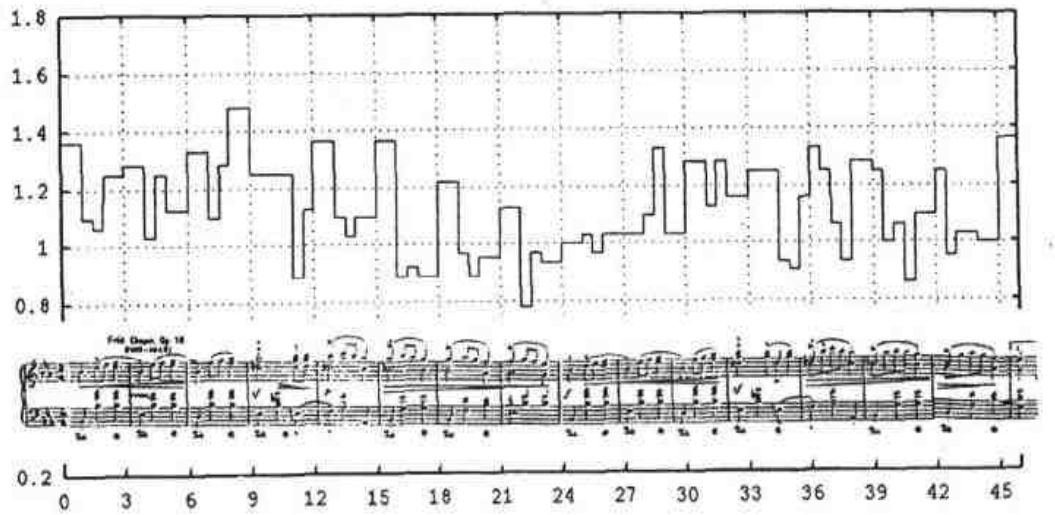


图 11.12　Chopin Waltz op.18，Eb 大调，由作者演奏（动力）

11.6　对真实艺术的演奏的一次机器学习分析

所有实验都尽量以作者本人的演奏为训练样本。一个可能的原因是人们对于这种数据的有意无意的青睐。

本节主要讨论用真实数据进行的实验，也就是说，由一群世界知名的国际钢琴家演奏的完整片断。结果清楚地表明在不同艺术家之间，个人的演奏风格有显著不同。这个实验也帮助我们准确找出了我们当前办法的许多弱点。现在我们正在进行策略和音乐理论单词表的适当提炼。这里我们不会提出一个详细的描述——下面仅仅想向读者简单介绍关于问题的复杂性，以及我们目前已得到的结果。进一步的细节可参见 Widmer（1995b）。

将要讨论的片断是 Robert Schumann（罗伯特·舒曼）的浪漫钢琴曲“Träumerei”（取自“Kinder szenen”,op.15）。图 11.13 展示了整个片断的谱子。Bruno Repp（1992）已经测量了由 24 个知名钢琴家对于该片断进行的 28 次演奏的偏差。我们应用这组数据集作为一系列实验的基础。Repp 的数据仅捕捉了富有表现力的时机（节奏变化的情况），而未考虑力度变化的情况。我们使用了应用基于知识的抽象的 IBL-S_{MART} 学习算法（方法二）。

在最高层次上，Träumerei 由长分别为 8 和 16 小节的两部分组成，其中第一部分要强制重复一遍。实验中，我们使用不同钢琴家演奏的第二部分去学习。第一部分用来进行测试。

Repp 名单中的最前面的三个钢琴家——Claudio Arrau，Vladimir Ashkenazy，Alfred Brendel 的演奏被选出来做第一个实验。把他们关于 Träumerei 的第二部分的演奏作为训练样本。他们各自的演奏曲线如图 11.14 所示（为了便于比较不同曲线，我们使用了略微不同的绘图风格）。和前面一样，X 坐标表示片断开始时关于四分音符的绝对长度（曲谱时间）。绘图显示了相应的节奏变化——曲线越高，局部节奏越快。

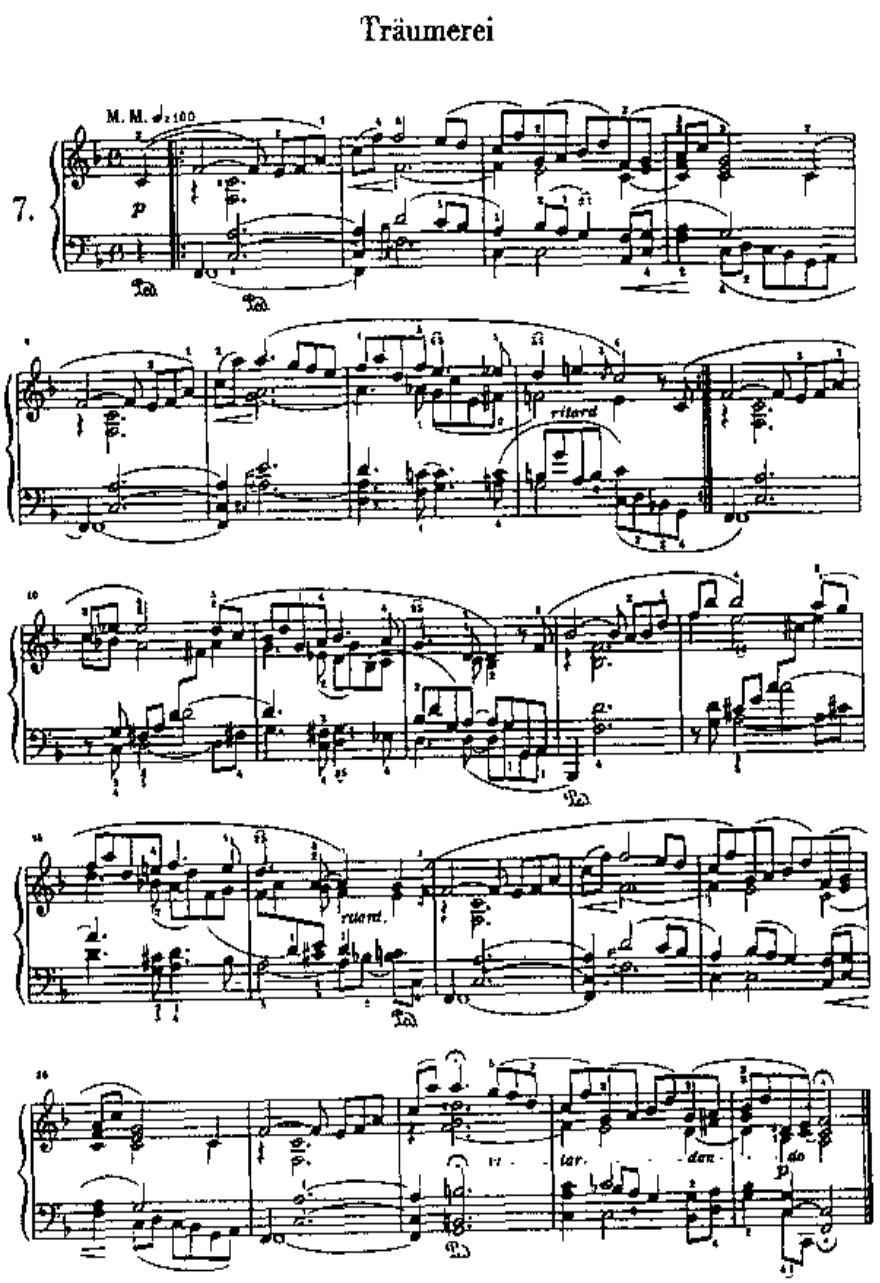


图 11.13　罗伯特·舒曼的“Träumerei”

很明显，在全局演奏层次上的结果也是一样的，不过在细节上仍有很多不同。三个钢琴家都观察到了通过重要结构边界（如主要乐句的结束）和（或）乐曲上的表示标记指示出的乐曲的逐渐徐缓。第三至最后小节的乐曲徐缓是因为乐谱上的延长记号。

系统在学习过这三个样本并且把它与同一部分由它的一个老师（Brendel）的演奏进行对比过以后，对于测试片断的表现结果（Träumerei 的第一部分），如图 11.15 所示。

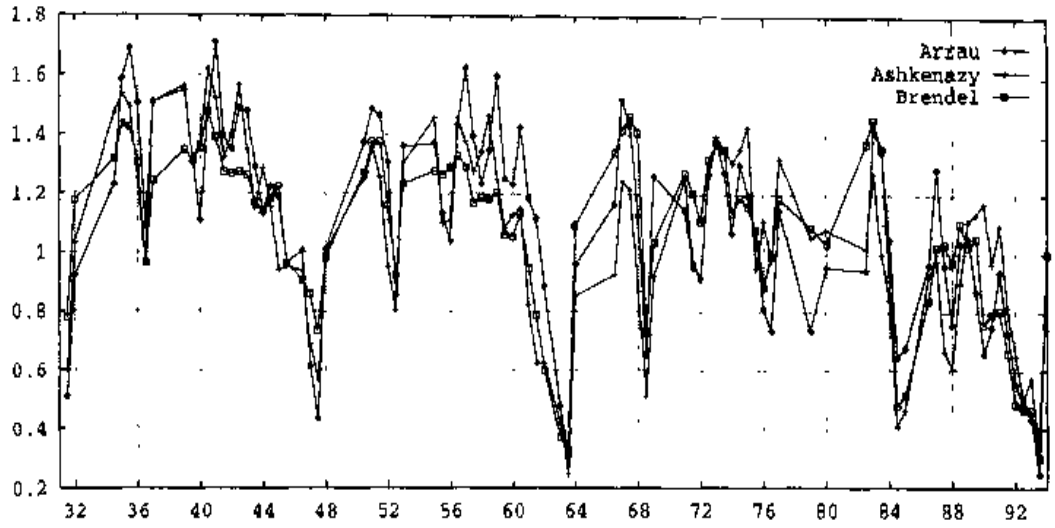


图 11.14　由三个钢琴家演奏的 Träumerei 的第二部分（节拍曲线）

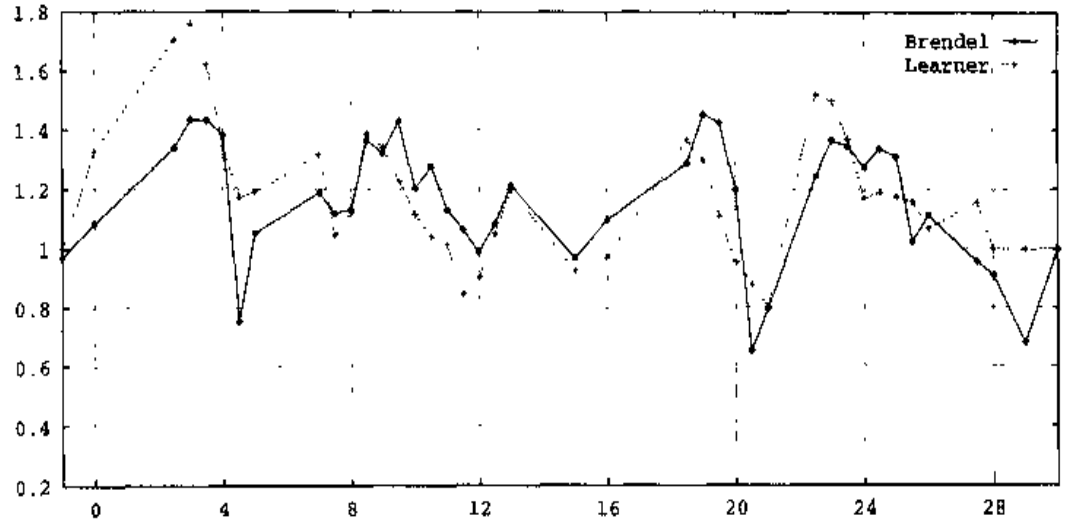


图 11.15　学习器与 Brendel 分别对测试样本演奏结果的比较

图 11.15 展示了结果在全面的、高层次的趋势上相当可观的一致性，不过同时在细节上也有一些差异（例如，在第 3 和第 7 小节中的乐句结构细节）。其中的一些差异也显示了我们当前系统的缺点。举个例子，系统无法复制 Brendel 的在第三和第七小节中的小旋律主题的乐句的演奏方式。进一步的分析显示了这是因为抽象的富有表现力的形式的有限集所致（参见第 11.5 节），学习器可以从给出的执行曲线上把它识别出来。我们计划在学习器的形式单词表中引入更多复杂的、抽象的模式。然而，一般地，我们认为结果还是非常令人满意的，尤其是在给定了前面提到的三个老师的演奏样本的情况下。尽管在高层次上是非常相近的而在低层次上还有很多不同点。

可以在机器学习帮助下研究的另一个有趣的方面是独立艺术家之间个人风格的不同。Repp 收集的数据还包括 Vladimir Horowitz（他以与众不同的音乐阐释而闻名）的三场演奏。在另一个实验中，我们把 Horowitz 的三场演奏（同样仅仅取片段的第二部分）用做训练样本。系统在学习过 Horowitz 的三个样本并把它与 Horowitz 的一场演奏相对比后，对于测试片断的演奏结果如图 11.16 所示。

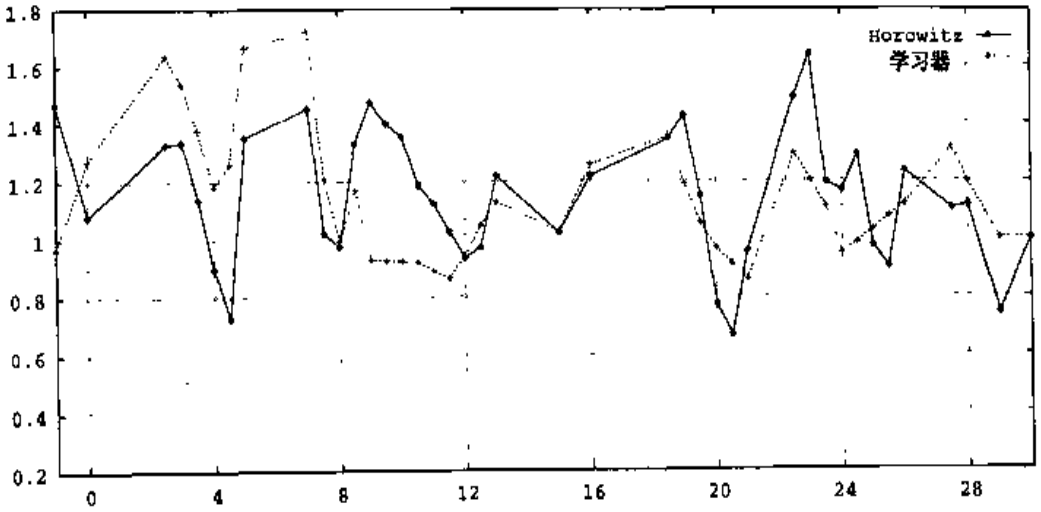


图 11.16　学习器和 Horowitz 分别对测试样本演奏的结果比较（Träumerei 的第一部分）

很明显，Horowitz 的演奏确实与别人的，比如 Alfred Brendel 的不同。看起来学习器确实想要复制一部分 Horowitz 的风格，但是与标准的音乐阐释风格如 Brendel 的相比还是不及的。对于这一点我们无法给出一个定性的解释，也许是 Horowitz 的风格太特殊了，他的演奏方式不能通过明显的音乐结构特征轻易地讨论或者解释。我们非常希望进一步对学习规则进行分析，更加细节化的实验能够提供演奏差异的明确的内部面貌，大体上这将对音乐学带来益处。无论如何，我们可以从实验上说明这两个基于知识的学习办法要好于没有音乐知识的学习。

11.7　实验结果的讨论

在本章的介绍中，我们主张，作为一个跨学科的项目，我们的工作应该为所涉及到的两个学科都能带来有益的结果。前面一节所提供的样本结果已经在一些方面暗示了这点。这里，我们将要从机器学习和音乐学观点上更近

地审视这些结果。

11.7.1　定量的分析

从机器学习的观点来看，本项目的主要贡献是在基于知识的学习上介绍和比较了两种不同的办法：第一个在于，在定性的领域理论形式上把不完整的、很不严密的领域知识解释清楚，并且设计使用该理论指导其启发式搜索的感应学习算法。另一个办法是使用领域知识把训练样本和全体学习问题转化为音乐的拟真抽象层次。我们关于 Bach 的小步舞曲的结论，主要在 11.4.4 节中讨论过，只是略微显示出附加知识的介绍（这方面是通过第一个办法）确实能够改善学习结果。无论如何，人们都愿意获得能清楚证明这个假设的定性结果。

我们的应用领域的一个基本问题，至少从机器学习的角度来看，是不可能对结果进行精确定性的评估。富有表现力的演奏的音乐质量是无法测定的。没有哪个正确的解释和美感的评估可以依赖众多的额外音乐因素，同时也很难形成一个全局的评估，例如一场演奏的一致性和平衡性。尽管如此，为了得到一些对我们的学习办法相对有价值的些许迹象，我们还是执行了一些简单的评估。

举例来说，我们对舒曼曲子的学习任务做实验，比较了三种学习算法：算法 0（基本算法）是没有任何领域知识的 IBL-S_{MART} 算法，因此限制了纯粹经验主义的学习。算法 1 是有了定性领域理论的同样的系统，如 11.4 节所述在乐符的层次上进行学习；算法 2 是如 11.5 节所述的有了基于知识的抽象的 IBL-S_{MART} 算法。每一个算法都使用三个钢琴家——Claudio Arrau，Vladimir Ashkenazy 和 Alfred Brendel 对关于 Träumerei 的第二段落的演奏来进行训练。随后把学习到的规则应用与该段落的第一部分，并且将生成的演奏结果与各自老师的演奏进行比较，主要是比较记录离散决策（也就是，对于一个乐符钢琴家和学习器使节奏舒缓和加速的频率）所相符合的数量。表 11.1 概述了这些“预测精确性”的测量值，三个钢琴家的平均值。读者应该知道，由于三个钢琴家的演奏在很多细节上是不同的，所以从严格意义上说 100% 的符合是不可能的。

表 11.1 学习器与老师相符的百分比

	原始方法（无知识）	方法 1（定性领域知识）	方法 2（抽象）
匹配率/上升旋律	58.46	61.54	55.38
匹配率/逐渐徐缓音	50.91	54.55	78.18
总匹配率	55.00	58.33	65.83

概要线（匹配的总百分比）揭示了基于知识的系统相对没有领域知识的学习器的一个明显的优势。在前者，办法 2（基于知识的抽象）很明显比办法 1（乐符水平上的学习）要好。这也巩固了我们先前的定性的评估（通过音乐分析和听取测试），并且也支持了理论上的猜想，即在音乐结构上的抽象比起在独立乐符层次上的知识的直接应用要更加符合实际。

但是这样的定量结果还是不够的。简单地数符合决策的个数实在是太粗陋了。在一个片断中并非所有的乐符都同等重要，有些错误比其他的要厉害得多。所有这些都依赖于一个有关音乐上下文样式的综合办法。一个音乐上的全面的比较应该考虑到所有相关的因素，这就预示着需要一个关于“正确”阐释的完备的理论，但是现在不存在这样的理论（也就是为什么我们首先开始经验主义的研究）。

更精细的比较如表 11.2 所列，表中列出了如果我们应用一个简单的计权方案来计数时的同一个实验的结果：系统和教师之间的每一个匹配/不匹配都用相对于基本乐符的**韵律度**来计权值。这意味着对于乐符的相对价值的一个非常粗糙的评估。在加权分析中，三个学习器之间的不同点表现得更加清晰，其中基于抽象的办法在很大范围内体现了优势。无论这些评估最终在音乐上的正确性如何，它们都确实在音乐背景知识的有效性和基于知识的学习器的效率性方面为我们提供了可靠的依据。

表 11.2 学习器与老师相符的百分比

	原始方法（无知识）	方法 1（定性领域知识）	方法 2（抽象）
匹配率/上升旋律	61.93	58.88	57.87
匹配率/逐渐徐缓音	40.83	55.03	76.92
总匹配率	52.19	57.10	66.67

11.7.2　对于音乐理论有用的定性结果

从音乐学的观点来看，我们的定性结果更具有信息价值。一般地，既然领域知识——在领域理论形式或者抽象算子形式中——是基于两个近代的音乐理论，我们学习器的有表现力的演奏的音乐质量（和相对于无知识学习的优越性）为这些音乐理论的相关性提供了额外的经验主义依据。

通过直接检查所学习到的有表现力的规则，我们可以得到细节上的更多见识。比如说，关于从不同类型音乐学得来的规则的分析可揭示相应音乐的结构维度（Widmer,1995a）的不同。同时，实验已经证明结构层次的抽象通常为不同经典音乐类型提供了更好的结果，对其他类型，如爵士乐，乐符层次会更加充分——乐符层次规则执行得更好并且拥有更好的解释潜力。

一个非常有趣的结果是系统还有效地发现了一些音乐理论家数年前假定的乐调规则（例如，Sundberg 等人,1983;Friberg,1991），主要基于音乐直觉和音乐经验。举例来说，我们的学习器发现的一个规则如下：

```
ritardando( Note, X) :-
        interval_prev( Note, I),
        at_least( I, maj6),
        dir_prev( Note, up).
```

它可以解释成“增加所有的结束一个向上旋律并且至少跳到第六大调的乐符所持续的时间（通过一定的 *X* 值）。”这是一个来自 Sundberg 等人（1983）的规则 4 的说明，即增加所有的结束一个旋律跳跃（不论朝上下哪个方向）的乐符的持续时间。通过学习我们还发现了其他几个不同的 Sundberg 规则。因此，我们的实验对于 Sundberg 规则的适当性提供了额外的经验支持。这些结果同样也是相应不同音乐-理论的单词表研究的新起点，现在我们正与 Johan Sundberg 及其同事合作进行这些研究。

11.8　总结

本章已经展示了机器学习是如何很好地应用于真实的问题，如音调音乐理论领域的研究。与其他的“硬”科学如物理和化学相比，音乐在很多方面

都“软”多了，即在许多方面都是无法计量的，这也使进行各种精确的实验和有关感应学习研究的分析变得十分困难。无论如何，机器学习可以带来有用的定性贡献，例如领域方面的现存理论的经验评估。这个项目成功的先决条件是对应用领域和现存理论的彻底分析和关于领域建模的有效办法。这就包括精心设计单词表和表示语言，因为它们包含（和隐藏）许多领域特征知识和不明确的假设。

我们的项目已经产生了许多有趣的音乐上的结果，我们把我们的“再综合分析”法（也就是，让机器学习程序复制观察到的现象并分析结果）看做是一个可行选择或者对更多音乐学的传统方法的补充。

从一个机器学习的观点看，这种跨学科的项目还有这样的好处：新的应用领域可以激励新的学习模型和算法的发展，它不需要具有领域特征。例如，我们的 IBL-S_{MART} 算法就是一个对其他领域很有用的通用感应学习器。

本项目的未来工作将主要集中于领域建模方面。不同音乐-理论单词表和不同类型音乐的实验将使我们进一步认识到关于演奏风格的规律性和可解释性。编辑一个真实演奏的大量数据将会很困难（主要是版权的原因），但这对计划的成功是必不可少的。

致谢

作者感谢 Bruno Repp(耶鲁大学、Haskins 实验室)允许我们使用他收集的舒曼（Schumann）的演奏数据。特别感谢 Miroslav Kubat 和 Robert Holte 的关于本文的有益的意见和建议。对于奥地利研究学院的人工智能项目的财政支持是由奥地利联邦政府的科学、运输与艺术部提供的。学院在知识发现和数据挖掘领域的研究也部分得到了奥地利 P10489-MAT 号 FWF 基金的支持。

参考文献

Aha, D., Kibler, D., and Albert, M.K. (1991). Instance-Based Learning Algorithms. Machine Learning 6(1), pp.37-66.

Bergadano, F and Giordana, A. (1988). A Knowledge Intensive Approach to Concept Induc-tion. In Proceedings of the Fifth International Conference on

Machine Learning, Ann Arbor,MI.

Bergadano, F., Giordana, A., and Ponsero, S. (1989). Deduction in Top-Down Inductive Learning. In Proceedings of the Sixth International Workshop on Machine Learning, Ithaca,N.Y.

Collins, A. and Michalski, R.S. (1989). The Logic of Plausible Reasoning: A Core Theory.Cognitive Science 13(1), pp.1-49.

Deutsch, D. (ed.) (1982). The Psychology of Music. New York: Academic Press.Friberg. A. (1991). Generative Rules for Music Performance: A Formal Description of a Rule System. Computer Music Journal 15(2), pp.56-71.

Hunter, L. (ed.) (1993). Artificial Intelligence and Molecular Biology. Menlo Park, CA: AAAI Press.

King, R.D., Muggleton, S., Lewis, R.A., and Sternberg, M.J.E. (1992). Drug Design by Machine Learning: The Use of Inductive Logic Programming to Model the Structure-activity Relationship of Trimethoprim Analogues Binding to Dihydrofolate Reductase. In Proceedings of the National Academy of Sciences, Vol. 89, pp.11322-11326.

Lerdahl, F. and Jackendoff, R. (1983). A Generative Theory of Tonal Music. Cambridge, MA:MIT Press.

Lindsay, R.K., Buchanan, B.G., Feigenbaum, E.A., and Lederberg, J. (1980). Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project. New York: McGraw-Hill.

Michalski, R.S. (1983). A Theory and Methodology of Inductive Learning. In R.S. Michal-ski, J.G. Carbonell and T.M. Mitchell (Eds.), Machine Learning: An Artificial Intelligence Approach, vol. I. Palo Alto, CA: Tioga.

Mitchell, T.M., Keller, R.M., and Kedar-Cabelli, S.T. (1986). Explanation-Based Generaliza-tion: A Unifying View. Machine Learning 1(1), pp.47-80.

Muggleton, S., King, R.D., and Sternberg, M.J.E. (1992). Protein Secondary Structure Predic-tion Using Logic-based Machine Learning. Protein Engineering 5(7), pp.647-657.

Narmour, E. (1977). Beyond Schenkerism: The Need for Alternatives in

Music Analysis.Chicago, Ill.: Chicago University Press.

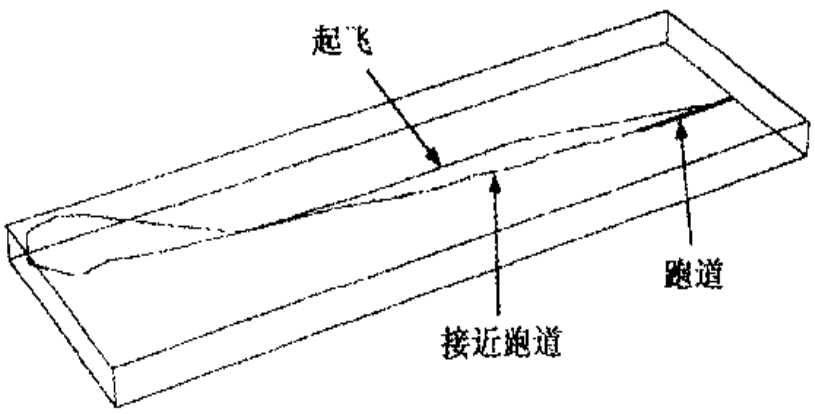
Quinlan, J.R. (1990). Learning Logical Definitious from Relations. Machine Learning 5(3),pp.239-266.

Repp, B. (1992). Diversity and Commonality in Music Performance: An Analysis of Timing Microstructure in Schumann's "Traumerei". Journal of the Acoustical Society of America 92(5), pp.2546-2568.

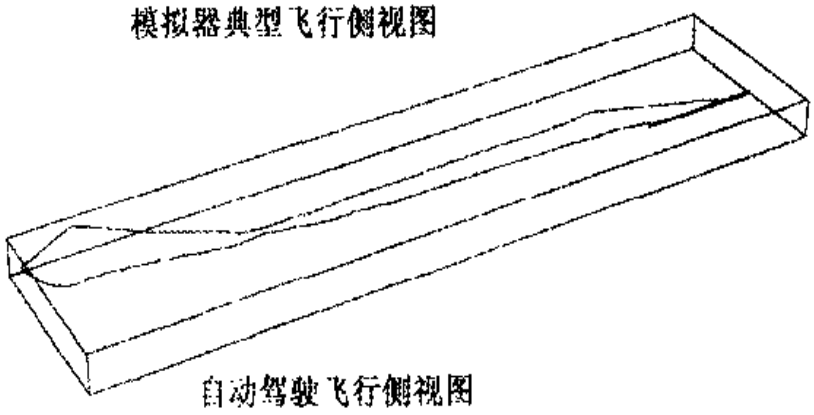
Russell, S.J. (1989). The Use of Knowledge in Analogy and Induction. London: Pitman.

第 4 部分

计算机系统和控制系统



模拟器典型飞行侧视图



自动驾驶飞行侧视图

第12章　网页哨兵：万维网页学习者

Robert Armstrong, Dayne Freitag, Thorsten Joachims 和 Tom Mitchell

摘要

这里介绍一种网页信息搜索助手，它称为网页哨兵，主要运用学习到的目标信息的知识互动地帮助用户定位所需信息的可能链接。我们最先关注的是：（1）用登录成功与否的数据训练、组织网页哨兵，为 Mosaic 用户提供互动信息；（2）在给定近期用户访问页面和用户信息搜索目标的情况下，运用机器学习方法自动获得选择恰当链接的知识。这里我们介绍网页哨兵的原始设计和初步的学习实验结果。

12.1　概述

很多时候会提及运用软件可以帮助人们定位互联网上的信息。本文介绍一种叫做网页哨兵的代理工具的原始设计和应用。这种网页哨兵旨在为正在浏览网页的人们提供互动搜索建议，并替他们自动去搜索信息。在互动模式下，网页哨兵扮演一个学徒的角色 [Mitchell 等人,1985; Mitchell 等人, 1994]，它给 Mosaic 用户提供关于下一个要访问的链接的互动建议，然后，通过观察用户对此建议的反应以及用户行为的成功与否来学习。网页哨兵最初只提供这种互动工作模式，并不拥有足够的知识来指导大规模的有用信息搜索。在本文中我们把网页哨兵作为基于 Web 的信息检索学习代理的一个案例。我们特别要关注有关激活网页哨兵观察和建议用户浏览网页地址的接口以及具有学习方法的原始实验结果。

12.2　网页哨兵

这一节通过网页哨兵的一个使用场景来介绍它的设计。网页哨兵是一个

信息搜索代理，它是先被连接到自身网页的超链接所调用，然后填写一个 Mosaic 表单来指定要寻找哪些信息，再从原始地点返回给用户这些网页（副本），并在用户浏览这些页面寻找目标信息的时候提供帮助。在用户浏览网页的时候，网页哨兵运用它学习到的知识，通过加粗一些目标链接的方式将其推荐给用户。在任何时候，用户可以通过单击一两个指示按钮指出搜索成功或者放弃本次搜索，以此来卸载网页哨兵。

用户在某一个特定场景下浏览的网页序列如图 12.1 到图 12.5 所示。第一个屏幕显示某个有关机器学习的典型网页。注意到在第三段，网页邀请用户试用网页哨兵，如图 12.1 所示。如果用户单击此链接，用户将到达网页哨兵

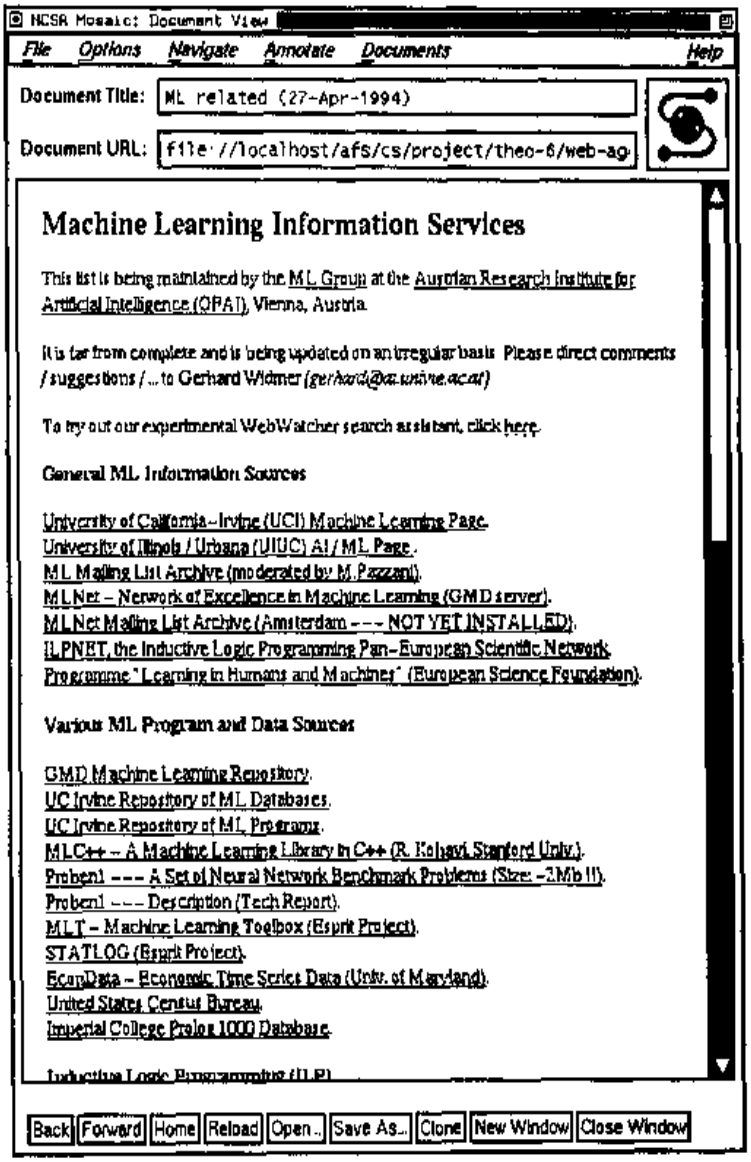


图 12.1　原始页

的起始页（如图 12.2 所示），它允许用户指定想要寻找的信息类型。在这个场景下，用户提出要寻找一篇文章，于是他看到一个新的带有细化需求信息表格的屏幕（如图 12.3 所示）。一旦完成，用户将返回携带有网页哨兵的第 1 页（如图 12.4 所示），注意到出现在屏幕顶端的网页哨兵图标和在屏幕下半端被加粗的链接。这个加粗的链接表示网页哨兵提示用户访问 UIUC 人工智能/机器学习页面。用户决定接受建议并来到如图 12.5 所示的包含网页哨兵新建议的页面。搜索的过程以这种方式继续进行着，直到用户单击“找到”或者“放弃”按钮来卸载网页哨兵。

图 12.2　网页哨兵的起始页

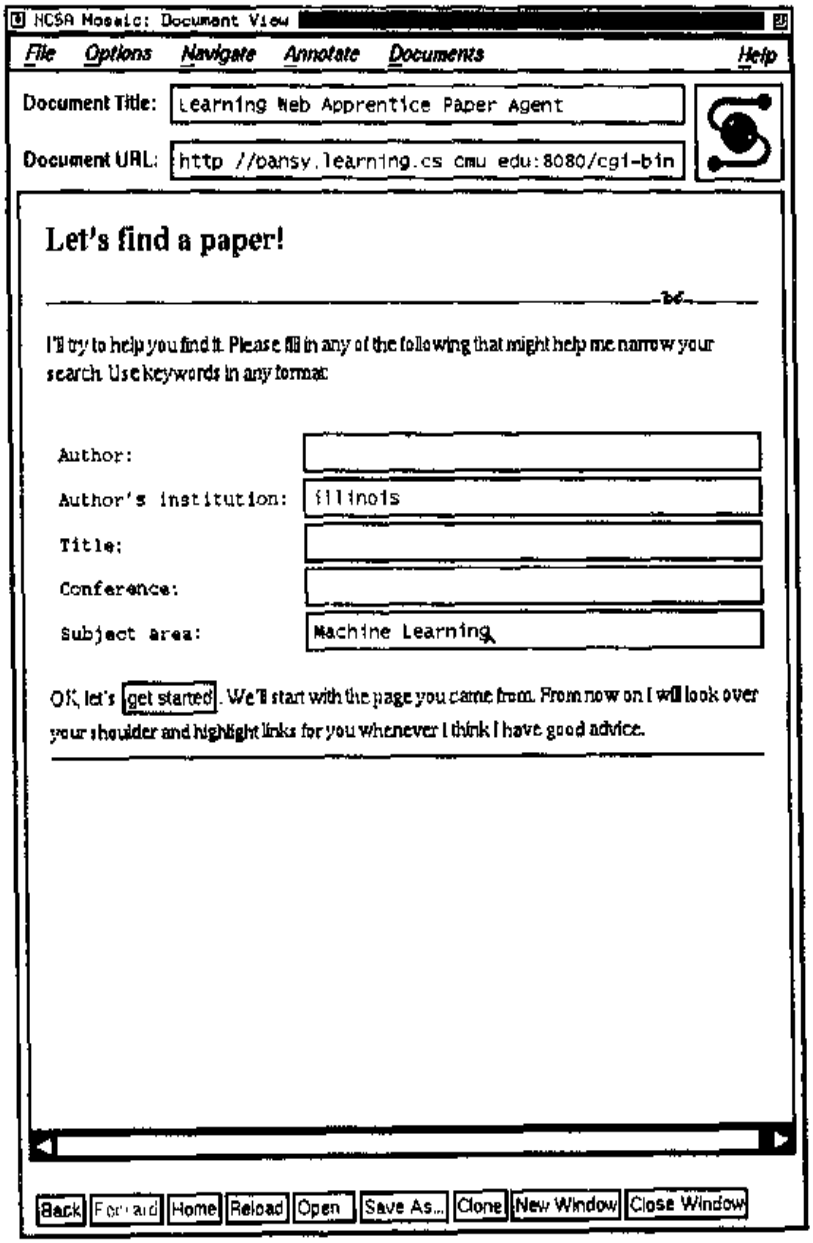


图 12.3　搜索表格

从用户的角度看，网页哨兵是一个具有关于如何从激活页面继续搜索目标的专业知识的代理。当网页哨兵建议用户应该访问哪个链接的时候，用户继续保持很强的控制权甚至可以在任一步操作时忽略系统给予的建议。我们感到让用户保持控制权是重要的，因为网页哨兵的知识可能提供不太完善的建议，而且网页哨兵也可能不能很好地理解用户的信息搜索目标。

从网页哨兵的角度，有点不同于上面的场景。当第一次被激活，它接受一个主题，嵌入到包含用户“返回地址”的访问 URL。返回地址就是用户来自的那个网页的 URL。一旦用户填写了信息搜索目标，在做了三处改动以后，

网页哨兵返回给用户一个原始页面的副本。首先，网页哨兵广告条被添加到网页顶部；然后，每个原始网页的超链接 URL 将被新的指向网页哨兵的 URL 取代；最后，如果网页哨兵发现网页存在一个被搜索知识特别推荐的链接，该最有价值的链接将被加亮，以便将其推荐给用户。它将把做如此修改后的返回页副本发送给用户，并且打开一个文件并始记录用户的信息搜索作为其训练数据。当处在等待用户的下一步指令状态时，它将预取推荐给用户的网页并开始做处理，以决定最佳的将要被加粗的超链接。当用户单击“下一个链接”，网页哨兵更新本次搜索的日志，检索到该页面（除非它已经被预取），执行相似的置换并返回给用户一个副本。

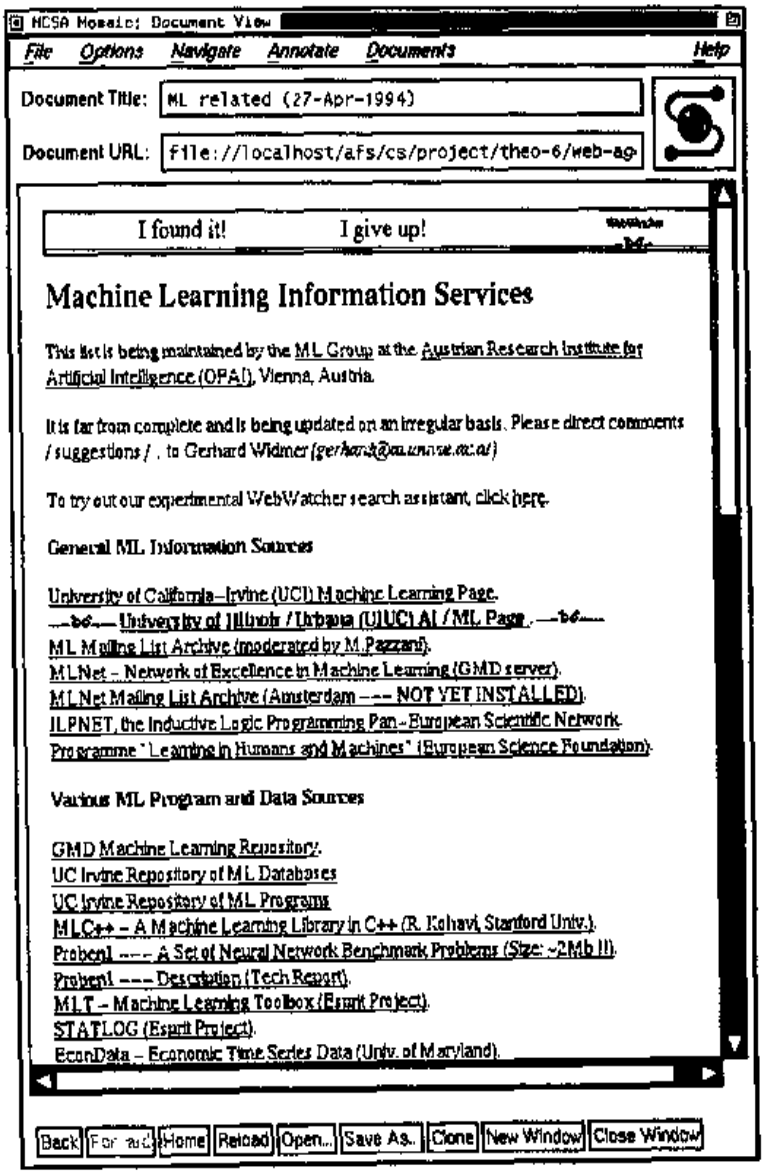


图 12.4　网页哨兵建议的第 1 页

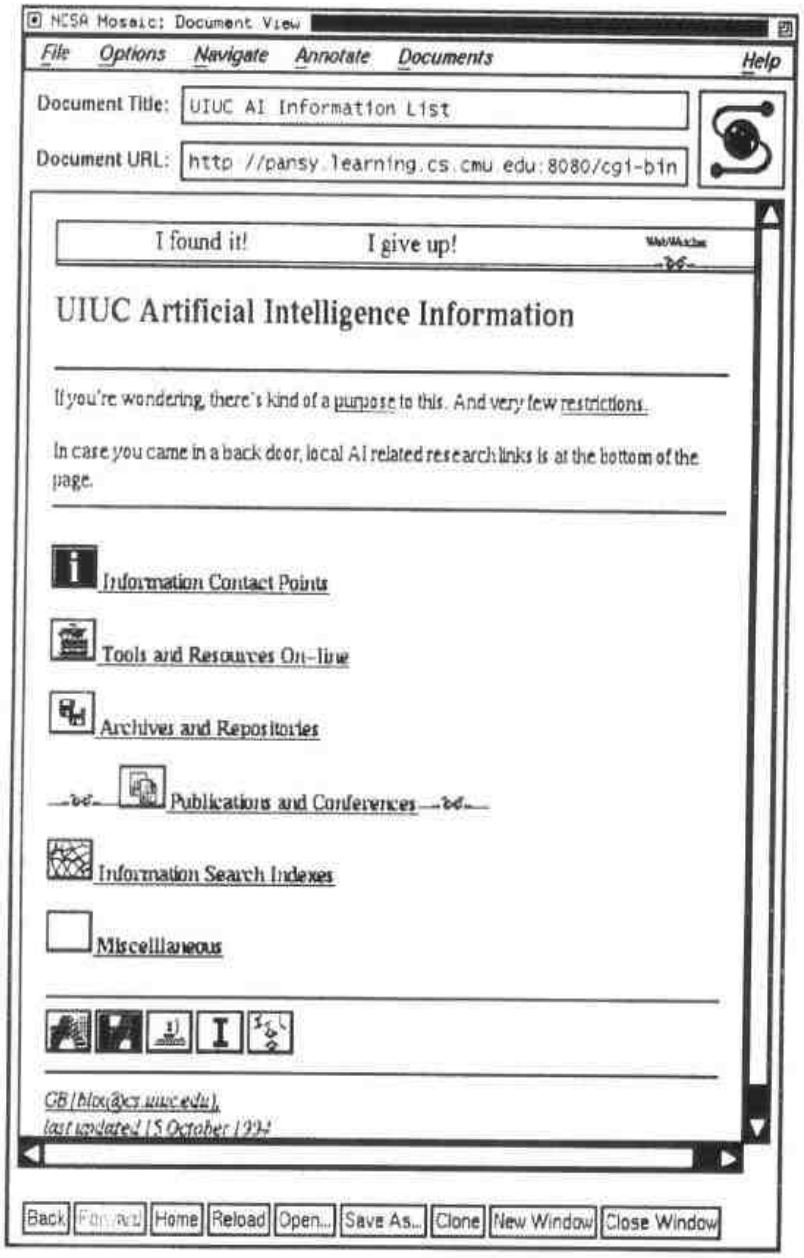


图 12.5　第 2 页，用户接受了网页哨兵的建议

这个过程将一直持续着，网页哨兵跟踪用户的 Web 搜索，对每一步提出建议，直到用户选择卸载这个代理。这时候，网页哨兵关闭此次会话的日志文件（指出搜索的成功与否，这取决于用户在卸载网页哨兵的时候所选择的按钮），并将用户返回到浏览原始未被置换的当前网页的副本的状态。

上面的场景描述了一个典型的与当前流行的网页哨兵的互动过程。我们打算从几个方面扩展这个原始系统。譬如：当等待用户下一步输入时，可以使网页哨兵根据自己的推荐提前搜索几个网页，以证明它提供的建议的品质。另外，如果在提前搜索网页的时候遇到一个特别有推荐价值的页面，可以建

议用户直接跳转到该页面，而避免将网页哨兵的访问路径都走一遍。

12.3 学习

网页哨兵成功与否的关键取决于它指导搜索的知识的质量。因为预知这些知识存在很大的难度，也因为我们希望很多不同的网页哨兵，变得能够拥有不同领域网页的渊博知识，所以，我们有必要研究从经验中**自动**学习控制搜索知识的方法。

12.3.1 该学些什么

网页哨兵要求知识满足什么样的格式呢？总的来说，它的任务是在给定用户、目标和网页的前提下推荐一个恰当的链接。因此，对应于函数知识，一个通用的知识形式为：

$$\text{LinkUtility} : \text{Page} \times \text{Goal} \times \text{User} \times \text{Link} \rightarrow [0,1]$$

这里，Page 表示当前网页，Goal 表示用户要寻找的信息，User 表示用户的标识，Link 是在 Page 中发现的一个超链接。LinkUtility 表示从 Page 上的 Link 到某一满足当前 User 当前 Goal 的页面的最短路径的概率。

在这里的学习实验中，我们考虑学习一个较为简单的函数。它的训练数据容易获得并且它仍然具有实际使用价值。这个函数为：

$$\text{UserChoice?} : \text{Page} \times \text{Goal} \times \text{Link} \rightarrow [0,1]$$

这里，UserChoice?取值为在给定当前 Page 和 Goal 的情况下，任意用户将选择 Link 的概率值。注意，这里的 User 并非一个显性输入，并且函数值预测的只是用户是否选择 Link——而不是它是否能直指最佳的目标。还要注意这里并未考虑用户到达当前网页的搜索路径信息。

在我们原始的实验中，关注 UserChoice?的一个原因是网页哨兵自动记录的数据提供了该函数的训练实例。特别地，每当用户选择一个新的超链接，对于当前网页的每个超链接的一个训练实例将被记录下来（对应于 Page, Goal, Link 和用户是否选择该 Link）。

12.3.2 怎样描述 Pages, Links 和 Goals

对于学习和使用目标函数 UserChoice?来说，首先必须为 Page × Goal ×

Link 选择一个恰当的描述。这个描述必须与有效的学习方法兼容，而且它必须允许代理有效地评估学习到的知识（譬如：能够区分可忽略延时与典型的网页访问延时）。注意到这样一个事实：网页中，与信息关联的超链接以及用户信息寻找的目标主要是基于文本格式的。然而，大多数机器学习方法采用类似于特征向量的较为结构化的数据描述。我们实验过用多种描述方式来重新将任意长度的关于网页、链接和目标的文本描述为定长的特征向量。这个思想常见于信息检索系统[Salton 和 McGill,1983]。它的优点是将任意量文本概括为一个适用于机器学习方法的定长特征向量。同时，它的缺点是丢失很多信息。

这里提到的实验都使用相同的描述形式。有关当前的信息、用户信息搜索 Goal 和特定的加粗 Link 被描述为一个包含大约 530 个布尔型特征值的向量，每一个特征表示在一个原始定义这三个属性的文本中出现的某个特定单词。这个由 530 个特征组成的向量又是由四个相连的子向量组成的：

1．超链接中带下划线的单词

200 个布尔型特征值被分配来对超文本链接中出现的单词进行有选择的编码（也就是用户看到的带下划线的单词）。这 200 个特征值只对应于 200 个在训练数据中的链接中最富信息量的单词。

2．在包含超链接的句子中的单词

200 个布尔型特征值被分配来表示在包含超链接的句子（如果有）中选定的单词的出现。

3．与超链接相关联的标题中的单词

100 个布尔型特征值被分配来表示出现在 Link 上的标题（如果有）中的选定单词。这包括出现在任何嵌套层次的标题中的单词，只要 Link 出现在标题范围。譬如，在图 12.4 中任何出现在标题“Machine Learning Information Services and General ML Information Sources”中的单词都可能被作为特征值来描述加亮的链接。

4．被用来定义用户目标的单词

这些特征值是指那些用户定义信息搜索目标的时候输入的单词。在我们的实验中，考虑的惟一目标是搜索技术文献。这里，用户可以任意输入标题、作者、机构等（参见图 12.3）。以该方式输入的单词均包括在整个训练集中（大约 30 个单词，但确切的数字根据特定实验中所用的训练集的不同而有所

不同）。在这里，布尔型特征值的编码可以这样进行：当且仅当这个单词出现在用户特定目标中，以及跟该实例相关的超链接、句子或标题时，相应的特征值被设为 1。

为了给前三个方面选择编码方式，必须选择一些单词。在每一种情况下，先搜集所有在训练集中出现的彼此不同的单词，然后根据关于正确分类训练数据的相互信息量对它们进行排序。最终选择了排序最靠前的 n 个单词。相互信息量是一种常用的统计尺度（参见[Quinlan,1993]），它用来评估一个单独的特征（这里指的是一个单词）能够在多大程度上正确区分观察数据。

表 12.1 概述了关于当前 Page, Link 和 Goal 的信息编码。

表 12.1　给定 Page, Link 和 Goal 时对选定信息的编码

200 个 超链接	200 个 句子	100 个 标题	≈30 个 用户目标

12.3.3　应该用什么样的学习方法

学习器的任务是在给定一个用户登录的训练数据样本的情况下，学习通用函数 UserChoice?。为了探求可行的学习方法以及评估学习代理获得的能力，我们对网页哨兵在 30 个信息搜索会话中搜集的训练数据使用了以下 4 种方法：

（1）Winnow[Littlestone,1988]学习一个布尔型概念，它被描述为一个关于实例特征值的单值线性阈值函数。采用一种倍增更新法则学习这个阈值函数的权值。在实验中，我们通过一个变换来充实原有的 530 个属性。样本向量的每个属性 a 变换为两个属性 a,ā。一个属性等同于原始属性，另一个是它的反向量。经过学习阶段，我们除去阈值，用学习过的线性函数的输出来作为对实例的估计。

（2）Wordstat 试图直接用独立单词的统计量来预报一个链接是否应该被跟从访问。对 Page×Goal×Link 向量中的每个特征值，它做两个计算：计算该特征在所有训练实例中被设置的次数（total）；还计算该特征被设置并且该实例被分类为正的次数（pos）。假定某特征发生，比例值 pos/total 就提供了对该链接被跟从的条件概率的一个估计。我们试验了这些比例的不同组合方式。在我们所试验的方法中，效果最好的一个（我们在这里所报道的结果）

是假定这些单个单词的估计是相互独立的。这个假设允许我们直接组合个体估计。如果 p_1，…，p_n 是各个个体概率值，I 为给定测试向量的标号集，那么，相应的链接的相关概率值就由 $1-\prod_{i\in I}(1-p_i)$ 决定。

（3）带有余弦相似性测度的 TFIDF[Salton 和 McGill,1983],[Lang,1995]是在信息检索领域开发出来的方法。在一般情况下，首先建立一个单词向量 V。在我们的实验中这已经由上面的描述给定了。每一个实例现在可以描述为一个跟 V 同等长度的向量，将每个单词用一个数字代替。这些数字可以由这个公式计算：$V_i = Freq(Word_i)*[\log_2(n)-\log_2(DocFreq(Word_i))]$。这里，$n$ 表示样本总数，$Freq(Word_i)$表示 $Word_i$ 在实际样本中出现的次数，$DocFreq(Word_i)$表示包含有 $Word_i$ 的文档数目。向量的长度实行归一化。每一类目标概念的原型向量由加入所有该类训练的向量来建立。在我们的实验中，目标概念有两类：正（被用户跟从的链接），负（未被用户跟从的链接）。实例的估计这样来计算：从实例向量和正原型向量之间的余弦减去实例向量与负原型向量之间的余弦。

（4）Random 为了提供一个比较学习方法的基准尺度，我们也评估从具有同一概率的网页中随机选择一个链接所获得的性能。所用数据中每页链接数的均值为 16，范围是 1～300。

12.4　实验结果

为了开发机器学习方法为网页哨兵自动获得搜索控制知识的潜力，我们收集了来自 30 个运用网页哨兵搜索技术文献的会话数据集。在每一个会话中，用户从如图 12.1 的页面开始，跟从上面的链接搜索特定类型的技术文献。搜索由三个不同用户进行。平均搜索深度为 6 个步骤，30 个中有 23 个搜索成功地找到文献。每一个搜索会话提供了一个训练样本集，它对应于发生在用户访问的每一个页面上的所有 Page×Link 对。

12.4.1　UserChoice?能学习到多精确的程度

给定上面的描述方式和学习方法，一个显然的问题是：“网页哨兵能学着指导用户到多好的程度？”为了评估这个问题的答案，可用数据被分割为

训练数据集和测试数据集。每个学习方法被运用于训练会话集，并且根据在各自的测试会话集中用户采纳推荐的链接的频率对该方法进行评估。

为了在统计上获得对学习准确性的更好的估计，我们采取了 30-Fold 的方法，即将训练数据分割为 29 个训练会话和 1 个测试会话，这有 30 个可能的分割。然后，每个学习方法被运用于每个训练会话集并在测试会话上进行评估。30 个实验的结果做平均处理。这个过程为 4 个不同的学习方法中的一个都会运行一次。

图 12.6 绘出了这个实验的结果。竖轴表示在测试情况下，用户选择链接包含在学习知识推荐的链接的比例值。横轴表示每个网页中被学习器推荐的链接的数目。因此，每条曲线从左边数第一个点表示用户选择学习器最推荐的链接的比例值，第二个点表示用户选择学习器最推荐的两个链接中的一个的比例值，依次类推。

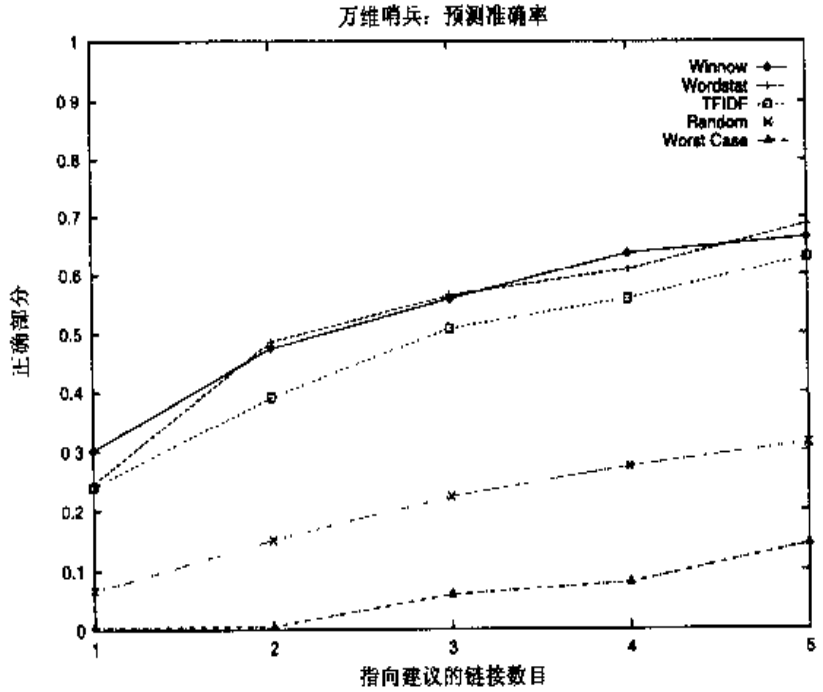


图 12.6　不同方法建议的准确性

竖轴表示网页中推荐超链接中包含用户选择链接的比例。横轴表示每页推荐的超链接数。最坏情况的曲线显示总共只包含 n 或更少链接的网页分数。

注意到三个学习方法都胜过随机产生建议的性能。譬如在测试时，用户选择的链接有 30%包含在 Winnow 推荐的链接中，并且 54%包含在前三位的推荐中。假定在该数据集中每页的平均链接数为 16 时，随机建议只选定了 6%的用户选择链接。

12.4.2　牺牲覆盖率能改进准确率吗

尽管这将要求推荐行为更保守，可还是有用户希望代理能提供更准确的建议。为了检测减小覆盖率来增加准确率的可行性，在实验中我们对建议增加了一个置信阈值。对这里考虑的每个学习方法，学习器的输出是一个实数，它可以用来评价推荐链接的可信度。因此，很容易在这些情况下引入置信阈值。

图 12.7 显示了当置信阈值变化时，建议的准确率是如何随覆盖率而变化的。在高置信阈值时，代理很少提供建议，但是常常能获得较高的准确率。在该情况下，准确率用所有实例中学习器最推荐的链接也是用户选择链接的比例值来衡量。因此，图 12.7 最右边的点对应于图 12.6 中最左边的点（即，100%覆盖情况）。

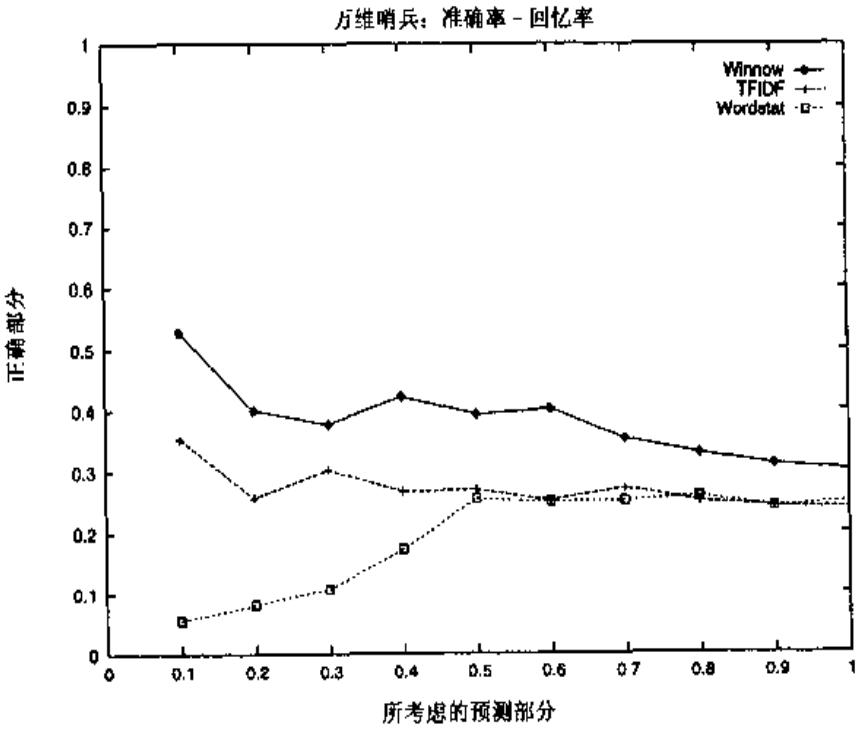


图 12.7　覆盖减小时准确率的上升

竖轴表示学习器最推荐的网页被用户采纳的比例值，横轴表示当置信阈值从高（左）到低（右）变化时，测试事例被建议覆盖的比例值

注意到当覆盖下降到更具选择性的 10%的实例时，Winnow 最佳推荐的准确率从 30%增加到 53%。有趣的是，Wordstat 的建议总体来说相对准确，而在高阈值时急剧退化。训练集中出现频率低的特征值导致极低的概率估计

值。训练集未经确证的特征彼此独立的假设造成了该现象。

12.5 总结

互联网上如洪水一般的信息量需要由搜索软件来帮助处理。网页哨兵的设计基于这样一个假设：搜索网页的知识可以通过互动辅助并观察用户的搜索行为来学习获得。如果成功，不同网页哨兵的副本可以很容易地附加在任意网页上，特殊的搜索助手对此是有用的。随着时间的推移，每个副本可以学习关于不同用户、信息需求和此类网页一般信息源的专门知识。

在这里提及的基本学习实验中，在当前网页、链接和目标的条件下，网页哨兵能够学习搜索控制知识，这些知识可以近似地预测用户选择的链接。实验还显示，可以通过使代理只有在高置信度时才给出建议来提高建议的准确性。尽管实验结果是正面的，但是他们是基于一个小数目的训练会话集的，并且是从特定领域的网页搜索特定类型的信息。我们还不知道这里的实验结果对于不同的搜索目标、用户和网页位置是否具有代表性。

基于我们最初的探索，我们乐观地认为建立一个互联网页学习者是可行的。尽管学习到的知识可能只是提供了不完善的建议，就算是稍微减少每个页面需要考虑的链接数目也会导致整个搜索呈指数级的改进。此外，我们相信，通过采集利用页面上很多用户的大量数据，考虑采用除此以外的方法可以使学习更有效地进行。

致谢

我们感谢 Ken Lang 提供了很多文本网页学习软件，并且提出了通过动态编辑网页来应用代理的思想。感谢 Michael Mauldin 基于 Web 的文本检索系统的软件和建议。我们非常感谢 Rich Caruana 和 Ken Lang 对本文具有帮助性的意见。该研究受到 Rotary 奖学金、NSF 研究生奖学金及 Arpa,F33615-93-1-1330 号拨款的资助。

参考文献

T. Joachims, D. Freitag, and T. Mitchell, "WebWatcher: A Tour Guide for the World Wide Web,"Proceedings of the 1997 International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers: San Francisco, August 1997.

K. Lang, "NewsWeeder: Learning to Filter Netnews", Proceedings of the International Conference on Machine Learning, 1995.

N. Littlestone, "Learning quickly when irrelevant attributes abound," Machine Learning, 2:4, pp. 285-318.

T. Mitchell, S. Mahadevan, and L. Steinberg, "LEAP: A Learning Apprentice for VLSI Design," Ninth International Joint Conference on Artificial Intelligence, August 1985.

T.M. Mitchell, R. Caruana, D. Freitag, J. McDermott, and D. Zabowski, " Experience with a Learning Personal Assistant," Communications of the ACM, Vol. 37, No. 7, pp. 81-91, July 1994.

G. Salton and M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Inc., 1983.

J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

第13章 计算机病毒的生物启发式防御

J.O.Kephart，G.B.Sorkin，W.C.Arnold，D.M.Chess，G.J.Tesauro 和 S.R.White

摘要

在人类专家对现存病毒分析的基础上产生的现代反病毒技术几乎跟不上每天产生三种新病毒的步伐。近几年，在网络上漫游的智能代理为新种类病毒的传播提供了极其丰富的温床。在 IBM 公司里，为了挫败今天和明天的病毒，正在开发十分新颖的生物启发式反病毒技术。这里，我们仅描述其中的两种：神经网络病毒检测器（区分染毒与非染毒程序）；计算机免疫系统（鉴别新病毒，自动分析病毒，依分析结果检测和清除现存系统的所有病毒复本）。神经网络技术已整合到 IBM 的商业反病毒产品中，而计算机免疫系统还处于雏形阶段。

13.1 介绍

每天，世界上大量病毒编者中的任一群体要造出三个以上计算机病毒。与此规模相当的反病毒软件开发者（代表每年工业产值近 1 亿美元）狂热地分析这些病毒，开发应对方法，并频繁地分发软件复本给用户。

目前，战斗已相当有规律。基于多年来对成百上千台机器样本总体的观察，我们的统计表明：在指定的年份里，商业媒体染毒情况大约占所有染毒机器的 1%。尽管具有有效的预防措施，但世界上的计算机总体上还是受到干扰。常用的大部分反病毒产品都能有效地检测到并清除病毒。在我们的样本总体中，在现实事件中仅有 10%的已知病毒可被观察到。曾经相对普遍的几种病毒现已成为危险种类表中的成员。今天，计算机病毒是可控且讨厌的东西。

有几种趋势已转向支持计算机病毒作者一方，令人烦恼。首先，新病毒

产生的速率已达到制服人类专家的边缘，并且大体上有增加的潜能。其次，为便利计算机用户，在网际互联及互操作上的连续发展也促使 DOS 和 Macintosh 病毒的兴旺。理论上的流行病学研究表明，计算机病毒在全球范围内传播的速度对于软件交流的混乱程度及速率是非常敏感的。诸如传播速度实质性提高的威胁因素及传统类型病毒的渗透性在预测中都有所重视。

此外，移动代理对全球网络的导航，也潜在地为新病毒的快速传播提供了丰富的媒介。无论主机在哪里，病毒都能自制复本，并利用主机间的交流来快速传播病毒。传统的检测及杀毒方法都是依靠专家分析，并随后给用户分配杀毒软件。但是，这些命令如此之多，如此之慢，以至于无法制服可在几日甚至几小时就遍及全球的病毒。

就上述问题，我们开发了生物启发式反病毒算法及技术。它取代了传统人类专家的部分杀毒任务并能快速自动地对新病毒做出反应。

在 19 世纪早期由 ADLEMAN 创造的“计算机病毒”这一术语，已表明了计算机病毒与它们生物同名者间的极大相似：附加自己到主人个体（组织或计算机）的小功能单元中（细胞或程序），并为新产生的病毒复本征用主人的资源。病毒通过用尽内存和 CPU 来使主机失灵。更糟的是病毒可能是致命的。在人类，DIPTHERIA 是由毒素感染细菌而产生的病毒。作为故意编制而来损坏主机的计算机病毒也同样是有害的。臭名昭著的 Michelangelo 病毒在 3 月 6 日发作，只要开机就损坏用户硬盘上的数据。

因此，从生物器官防御疾病这一防御生理反应中寻求启发也是自然的。用生物类比来防御计算机病毒的想法并不是我们原创，但我们却是首次把生物刺激性防御反应机理设计成反计算机病毒技术并收编成商业反病毒产品。

首先，我们简单介绍一下何为计算机病毒，它们如何自身复制及为何如此让人讨厌。其次，我们将描述专家分析计算机病毒的典型程序，并解释此方法即将过时的原因。最后，讨论受生物系统启发而出现的两种互补反病毒技术：神经网络病毒检测仪和计算机免疫系统。

13.2 背景

13.2.1 计算机病毒

计算机病毒是自复制软件实体，寄生性地依附于现存程序中：局限于 DOS, Macintosh 和其他微机系统。当用户执行染毒程序（可执行文件或开机程序段

磁区）时，病毒的代码段首先执行。病毒寻找更多具有写权限的程序，并附加自身复本（也可能是一个故意修饰的复本）到每个程序中。有些情况下，它可能执行有效负荷，如打印一奇怪信息，播放一段音乐，损坏数据等。最后，典型病毒反过来控制原程序。 除非病毒执行了明显的有效负荷，否则用户将对某些东西的丢失毫无察觉，并帮助病毒进行复制。病毒为提高其传播能力，把自己建成内存中的住户程序，最终导致主机停工。作为住户程序，它们可持续监视系统活动，并确认和感染开机程序。

一段时间后，此情况被重复，病毒可以传播到用户系统的多个程序中。最后，某染毒程序可能被拷贝，并可能通过软盘传到另一系统上。如果此程序在新系统被执行，则染毒循环将重新开始。以此方式，计算机病毒将由一个程序传染到另一程序，慢慢地由一台机器传到另一台机器。最厉害的 PC DOS 病毒可在数月内传遍世界。

蠕虫是另一种形式的自复制软件病毒，它有时不同于病毒。蠕虫是在多任务环境中的内存中能保持活跃并能自生的程序。它们通过“产卵”来复制复本，因此它们能自我决定复制，而不依靠人类执行染毒程序。它们比病毒传播得更快。1988 年的网络蠕虫可在一天内传播到全美数以千计的机器上。

13.2.2 病毒的检测、清除和分析

反病毒软件力求检测所有病毒并尽可能恢复染毒程序。

常用的互补反病毒技术有各种各样，其分类在[Spa91,KWC93]中给出。**活动监视器**警告用户带有病毒的系统活动，而不仅仅是与正常行为、合法程序相联系的。**完整管理系统**警告用户文件中的可疑变化。有两种非常普通的方法用来检测系统中存在迄今未知的病毒。然而，它们常常不能确定染毒程序的类型和位置，而且它们时常标记或阻止合法行为，干扰正常工作，或引导用户完全忽视警告。

病毒扫描仪查找染毒的文件、开机程序记录、内存及存放可执行代码的其他位置，以确定其中在一个或更多已知病毒中出现的字节模式（也称为“签字”）。病毒扫描仪能提供比活动监视器和实体管理系统的更特殊的检测。在建立实体和辨认病毒方面，病毒扫描仪是必不可少的。配备特殊知识并能恢复染毒程序的**驱除病毒仪**现已投入使用。扫描和维护设备的劣势在于只用于已知病毒或已知病毒的变异，此外，它们必须对每种病毒的血统进行分析，

以提取到允许消毒器清除病毒的信号和标志。一旦发现新病毒，病毒扫描仪和消毒器就会更新信息。况且分析需要花费人类病毒专家巨大的工作量。

一旦发现新病毒，它就迅速在国际反病毒专家间分发。获取到样本后，人类专家就拆开病毒并分析汇编代码，破译病毒行为及其自身依附主程序的方法。随后，专家选择一个标识（16 到 32 比特的序列），此标识表示一个指令序列：它在每个病毒例子中都确保存在，而且还要专家估计不可能在合法程序中找到。此标志可被编码到病毒扫描仪，依附方式和病毒机器码的描述被编码到核对器（核对器主要核对由扫描仪发现的病毒特征）。最后依附方式的破解码可被编码到消毒器中。

分析病毒是乏味并费时的，有时需花费几小时或几天。即使最好的专家也会选择弱标志：使病毒扫描仪误报合法程序染毒。这个负担的减轻要完全指望自动病毒分析的实现。

13.3　病毒种类的检测

鉴定计算机病毒有两种方法：过度性开通的源自事实的检测由活动监视仪和实体管理系统提供；过度性特性检测，由病毒扫描仪提供。有的地方也用处于这两种方法之间的方法，即理想“种类检测仪”：以程序代码作为输入，并由它决定系统染毒与否。完美的种类检测是一个算法性未定问题，正如[Coh87]的观察结果，对于停机问题，它是可还原的。然而，不完美的种类检测在实际中可行的情况也存在，在自动模式分类中也很自然地被作为一个问题。标准分类技术包括线性和非线性两种方法，如最近邻分类、决策树和多层神经网络。

在病毒检测问题中，开机程序的检测是重要的，也是相对易于操作的子问题。开机程序段是一个代码序列，告诉计算机如何通过引导自行启动。对于 IBM 的兼容 PC，开机程序段为 512 字节长，其主要功能是装载和执行存于其他处的附加代码。

尽管现有 4 000 多种不同的文件传染病毒，仅 250 种开机程序病毒，但 20 种常见的病毒中有 19 种是开机程序病毒，占所有病毒事件的 80%。开机程序段同样掌控着新观测病毒的名单。因此，在反病毒战中，检测开机程序新病毒的能力是非常重要的。

检测开机程序是相对有限的模式分类任务。所有合法的开机程序都有相近的功能。控制从其他处装载的合法开机程序之前，病毒开机程序与合法开机程序执行类似功能。

对于此应用程序，虚假肯定是有决定性的。虚假否认则意味着错过病毒。如果分类器使一病毒溜掉，则结果与无病毒保护同样糟。另一方面，虚假肯定可随时发生，其结果比无病毒保护更糟。而且，一个合法开机程序中的虚假肯定意味着在成千台电脑上的虚假肯定活动。虚假肯定是不能容忍的。

对合法开机程序与病毒开机程序的分类，最近邻分类法是简单且有吸引力的。两种开机程序属性的度量可用海明距离（看做 512 元素矢量）或编辑距离[CR94]（看做主题串）。对新开机程序分类，程序将从 250 种已知开机程序病毒和 100 种合法开机程序（只是一个代表）中寻找与新开机程序距离最近的一个。如果新开机程序的最近邻为病毒，则把它划为病毒。反之，划为合法的。

遗憾的是，最近邻分类对此问题不是很有效。原因在于一个病毒开机程序只是一个非常短的病毒代码，常写在合法程序边上，因此在总的比较中，一个病毒更像一个合法程序（合法成分占大部分）。也就是说，使病毒开机程序发作的不是全部病毒程序属性，而只是个别病毒功能的存在。

这些功能曾用来构建病毒分类器。如病毒的常见行为是尽可能减少可用内存的可见尺寸，以使它们所占的空间不被察觉。尽管这个行为在机器码中执行的方式各不相同，大部分机器码的执行匹配某一简单模式（一个虚假模式的象征形式是 C31B****AC348F**90D3D217——大约固定 10 比特和一些未知码）。以非传统方式低内存存储的病毒中大部分包括一个相同功能的两比特弱象征模式，但更倾向于虚假肯定。描述其他常见的病毒功能用强弱两种模式。

通过专家的病毒与非病毒开机程序知识和多天的广泛实验，我们手工制成一个 ad hoc 分类器（如图 13.1 所示）。分类器以四种病毒功能的强弱证据的模式来扫描一开机程序。赋予弱证据为 1 分，强证据为 2 分。如果一开机程序的总分为 3 分或更高，则把它划分为病毒。此分类器在划分 350 个样本时执行得很好，并且在 100 个反面样本划分中，虚假否认率为 18%，虚假肯定率小到无法度量，即病毒中 82%被检测到，无合法程序被划为病毒。

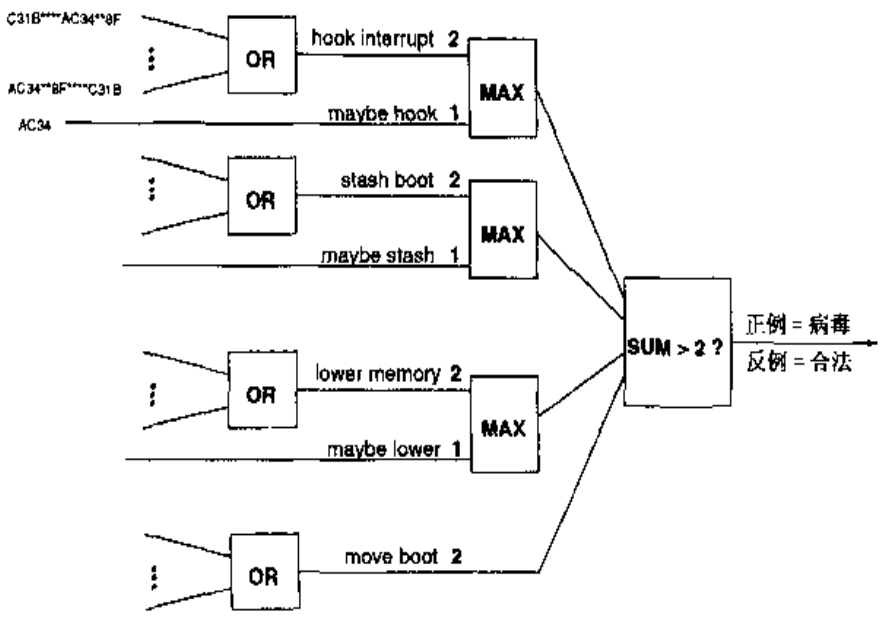


图 13.1　手工编写的多层次分类器。消除“MAX”方框以产生一个更传统的神经网络，但它是内部的，甚至在优化 7 个权重后也是如此

我们希望开发一个自动生成病毒分类的仪器（以相似属性作为神经网络的输入）的程序。由于 *ad hoc* 分类器整合了所有可得到的开机程序知识，因此有超负荷的可能性（对新数据归纳能力差）。评价自动生成分类器的归纳能力则相对更容易一些。我们希望属性的算法提取和网络权值的最优化能给出更好的分类结果，特别是在虚假肯定测量中。最后，我们相信一个自动化的程序能更加容易去适应引导扇区病毒的新趋势。如果新的类型的引导病毒变得普通，我们只要简单地重新训练分类器——这是个比处理一个特殊的分类器或从一开始就重写要容易得多的任务。

实际上，我们是这样做的：抽取一个 3 字节字符串或三字铭的集合，其中的元素经常出现在过滤性病毒的引导扇区，而在逻辑扇区并不常见。字符串里的“1”（出现）或“0”（没有）定义为单层神经网络的输入矢量（如图 13.2 所示）。其中超过一半以上的样本用来训练权值，剩下的一半用来测试网络性能的结果。在自动分类器的发展过程中，我们在**特征选取**和**不确定性学习**方面遇到了新的挑战，这些是我们认为在学习中经常引起人们兴趣的代表问题。这些将会在描述分类器构建的过程中有更为详细的介绍。

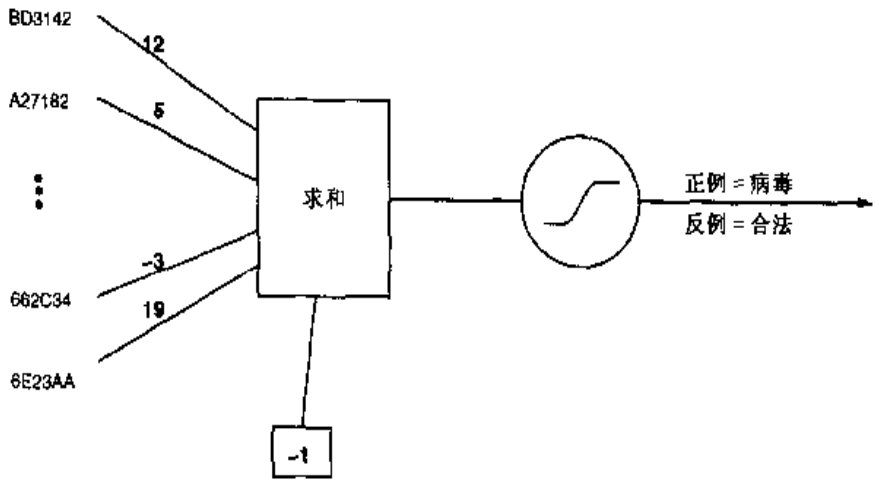


图 13.2　单层的神经分类器，大约有 50 个输入特征和权重，权重是虚构的

13.3.1　特征选取

构建的第一步是选取字符串来作为特征。对于一个能高度认识病毒、机器代码及用人工智能来选择包含通配符的复杂特征模式的专家而言，在算法特征产生中也只能胜任对 3 字节字符串特征的处理。一个有 150 多个 512 字节过滤病毒引导区的训练集合包含了 76500 个三字铭，其中 25 000 个是惟一的。

第一个挑战：出现了**特征筛选**。机器学习中一个著名的法则表明训练样本的数目必须远远大于调整参数的数目，为测试样本提取好的归纳。对所有有 150 个过滤病毒和 45 个非过滤病毒的样本的网络必须有远小于 195 个权——大约 50——也就是小于或等于输入的数量。某种程度上说，25000 个三字铭必须筛选到只有 50 个。

既然想要的是与逻辑行为对立的过滤病毒的指示三字铭，那么就可以很自然地把出现频率太高的三字铭从逻辑扇区中删掉。去掉所有在 45 个逻辑训练样本上的三字铭，即使它只出现过一次。这样减少的三字铭也仅仅占候选三字铭中的 8%，因为一般来说，在电脑程序中出现的一些三字铭类似于英语单词中的“the”，而它不是什么显著的特征，从而使筛选可以进一步进行。把出现频率超过 1/200 000 的三字铭删掉（平均出现率为每 200 000 个字节中超过 1 次），还可以再减少 8%，使得在 25 000 候选特征中只剩下 21 000 个。这样，还要求更为彻底的筛选。

假设要选择在过滤病毒的训练集合中显得重要的三字铭特征，一种方法

是选择至少在集合中出现过好多次的三字铭，但这会使一些没有任何三字铭表示的病毒的样本遗留下来。还有一个更好的办法是选择一个三字铭的覆盖：一个三字铭的集合中至少有一个三字铭来代表每个过滤病毒的样本。实际上，我们是可以提供接近 4-覆盖的东西，使得集合中每个过滤病毒的样本能被 4 个不同的三字铭表示（在一个有 21 000 个三字铭的全集中，有些样本的表示少于 4 个，在这样的情况下可能有的样本只需要 3-覆盖，2-覆盖或单覆盖）。4-覆盖产生一个有 50 个三字铭的特征集合，作为一个神经网络的输入这样已经足够少了（即使这样，一个完全的双层的有 h 个隐含节点的网络有 h 次和输入一样的权。在这里对于一个 2 或 3 的 h 是被禁止的。这就是为什么我们要用一个单层的网络的原因）。确切地说，大部分三字铭是类似于或更为复杂的 *adhoc* 分类器的模型的子串，然而，有些三字铭与任何模型都没有关系。在专家检查中，它们转而被定义为一个有意义的新的特征类。

13.3.2　分类器的训练和性能

通过构建，被选中的三字铭有很好的特征：在训练集合中，没有任何一个合法启动逻辑扇区包含它们，而大部分过滤的引导区则至少包含 4 个，似是而非地讲，特征的高质量引出了第二个挑战，我们称之为**不确定性学习**的问题。因为反例不包含任何特征，而所有特征正值的使用都给出了一个完美的分类器。

特别在如图 13.2 所示的神经网络中，单个 0 阈值和所有正权值给出一个训练样本的完美分类器，但由于即使单个特征也可以发出一个正值，这会使它对测试集合和真实世界中的虚假肯定很敏感。同样问题表现为不稳定性。当反向传播训练程序被用来优化权值时，权值越大越好，因为它们两格玛函数的输出，接近于一个 -1 到 1 的理想渐近线。

实际上，所有从有限中保持理想权值的是在一些反例中的特征的表现。由于没有特征在反例中表现，我们的解决方法是介绍一个新的样本。一个方法是加一个由同一矩阵定义的样本集合，就是对每一个在列的特征，产生一个人工反例。这个样本的特征输入是 1，其他所有输入为 0。这为每一个三字铭特征增加了一个人工样本，有助于更好地加强那些偶然出现的特征。

为此，我们使用的是从人工引导区中很多计算机程序的“原始进入点”部分获取的 512 字节编码，它们是用于引导机器的启动而不是应用的性能，

在5 000个人工引导区中，有100个包含一些过滤病毒的特征（和想像的一样，每一个被选的三字铭代码产生频率小于1/200 000，表示在512字节中找到任何50个三字铭的概率最多为13%，人工引导区的观察率为5%。）由于不是所有的50个三字铭都出现在每一个人工引导区中，我们把这个方法与“同一矩阵”法混用。

在这一点上，问题最终以最标准的（单层）前馈神经网络训练的形式出现。这可以由反向传播完成。在典型的训练和测试进行中，我们发现在人工引导区可以测得网络有10%～15%的伪负值率和0.02%的伪正值率（给出三字铭频率小于1/200000，如果它们出现是计算独立的，那么在512字节中找到2个三字铭的概率为最大0.8%）。保持0.02%的伪正值率，则在任何100个真实逻辑引导区中不用出现伪正值。

在网络学习中有一点与众不同。即使所有特征是过滤病毒行为的指示，但大部分训练竞赛产生一个或两个小的负值权。我们不能完全确定为什么会这样，但最简单的解释是如果两个特征完全有关联（一些是非完全有关联的），则只有它们整个权值是重要的。因此可能随机要求一个负权值和一些相应的更大的正权值。

对实际的引导病毒的检测，如果伪负值率为15%或更少且伪正值率为0.02%，则是一个很好的结果：85%的新的引导区病毒会被检测到，而伪正值很少有机会在合法引导区上出现。实际上，并入到IBM防毒的分类器已经查出一些新的病毒。至少有…个伪正值在一个类似病毒的安全引导区上，而不与典型代码的可能模式匹配。不是特别地允许这些引导区。重新训练小于1小时，确保了神经网络的负值分类，这可以帮助减少类似的伪正值。

在10%或15%的逃脱检查的病毒中，大部分不是因为没有包含三字铭特征，而是因为包括它们的代码区用不同的方式出现。如果出现的代码用独立的方式捕获，三字铭能从分类器上通过，这样病毒更容易被检查出来。

13.4 计算机免疫系统

尽管属性病毒检查可以比较好地作用于检查引导区病毒，也能永久地为文件提供有用的检测，但是在技术上至少有两个缺点存在：

（1）新的病毒只有在它们有足够大已知病毒中的代码数量时才能被检查

出来。

（2）这种方法只对过滤病毒有用，对想从引导区或文件中清除病毒是不适用的。彻底消除病毒感染的方法只能是删除或用别的文件代替被感染的引导区或文件。

属性分类器能被看成是一个类似于天生的或是非适应的非特定的表现在脊椎动物和更低级动物中的免疫系统。这个天然的免疫系统的一个重要组成部分可以认为是属性分类器系统。系统中的特征的认知基于以下方面：

（1）某些蛋白质的表现在内部细胞而不是外部细胞上。

（2）双海滨 RNA 的表现更大地集中在特别的病毒类而不是哺乳动物的细胞上。

（3）由一般的氨基酸开始的缩氨酸的表现由细菌大量生产，而哺乳动物只生产一小部分。

与属性分类器伴随的一个属性反应是一个使其无能或消灭它的病毒。然而，脊椎动物已经进化，并且有一个更高级更适应的免疫系统，与天然免疫系统相一致。基于对特殊病毒认知的基础上，它能检测和反应从前没有碰到的病毒，而不管它们与已知的病毒的相似程度有多少。这是一种搜索计算机病毒时精确的防御能力。

图 13.3 提供了计算机免疫系统的一个概观。这个免疫系统通过捕捉和分析过滤病毒样本对类似病毒的异常东西进行反应（和由不同活动和整体监视器定义的一样），从分析中，它推导出检测和删除病毒的方法。计算机免疫系统的许多组成部分是在实验室里工作的，并为 IBM 商业防毒产品提供有用的数据。

这节剩下的部分将讨论对免疫系统不同组分的设计及它们与类生物规则的关系。进一步对类生物的探索可以在[Kep94a]中找到。首先，我们要考虑作为现在 IBM 防毒软件中标记的组分的集合：未知检测、扫描已知病毒、去掉已知病毒。然后，我们要讨论一些现在病毒实验室所标记的组成的集合：用引诱物捕获样本，病毒算法分析，特征抽取。这些组成都是功能原型。最后，我们要讨论可以通知相邻机器关于病毒感染的情况的机制。

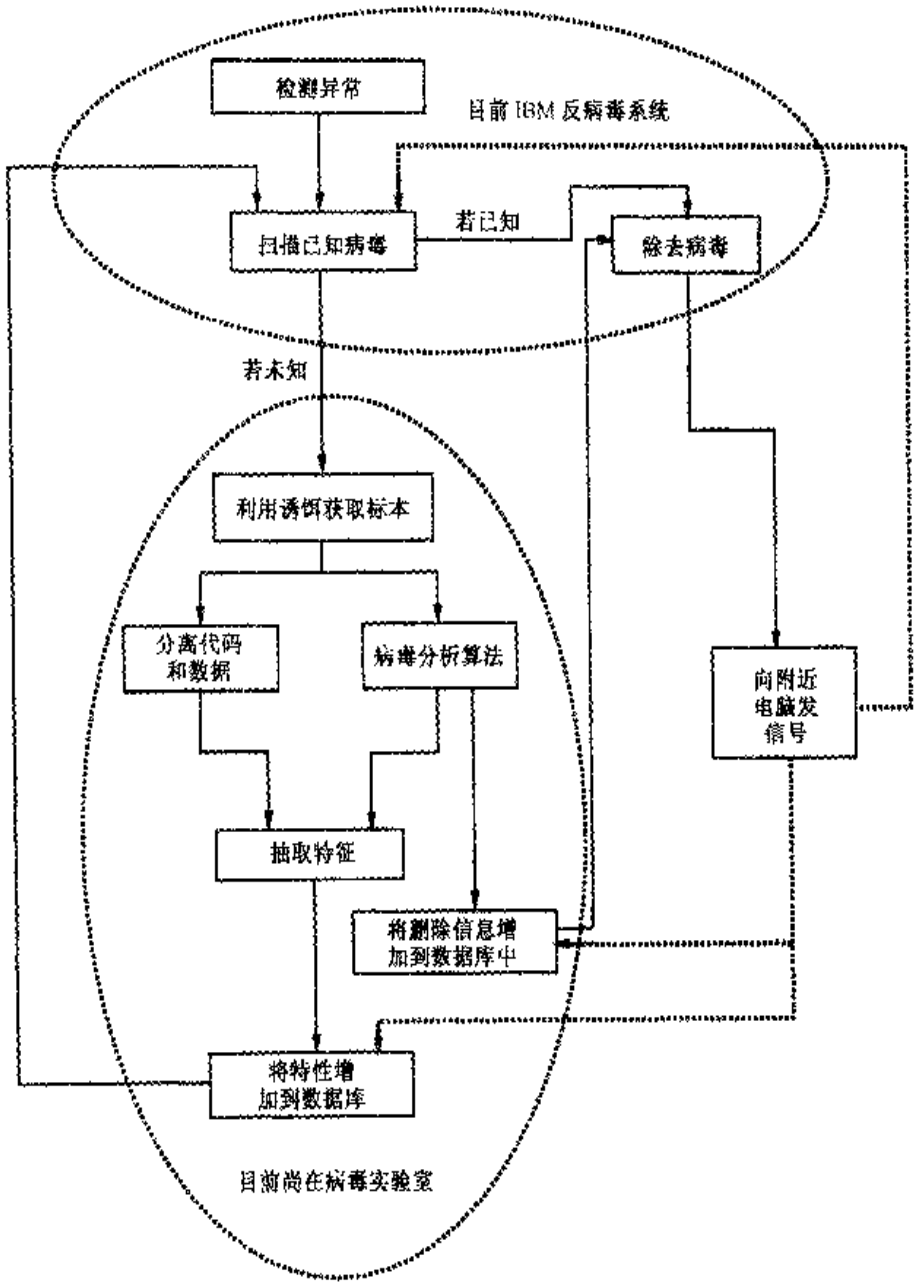


图 13.3　所提议免疫系统的主要组成，以描述计算机及其相互关系

13.4.1　未知检测

生物和计算机免疫系统同时面临的最基本问题是辨别所进入个体的恶意性和善意性。由于单个脊椎动物生命周期中身体化学成分的高稳定性，它们的免疫系统可以用简单得多的方法来辨别自身的和外来的东西。这是个很好的方法，因为自身的比恶意的好辨别得多。生物免疫系统只能简单地实现无

理的排外策略："知道自己（排斥一切别的）"，这个策略会对一些伪正值产生错误（比如善意进入的伪拒绝）。但是在血液输入和器官移植的情况下，则有不同，这些错误的影响是很小的。

在计算机中，同样的无理排外策略是未知检测的重要组成部分。整体监视器能用数量检查或其他方法来决定已存在的执行是否变化了。然而这仅仅是个部分的解决方法。"本身"的天性，如一个个体计算机上的软件集是一直随时间变化的——比生物器官中的更显著。人们不断在他们的系统中增加新的软件，通过购买新的版本或用新的源代码来更新软件。事实是一个执行新的或变化不足以使人怀疑。启用了一个补充的"知道你敌人的东西"的策略：未知物的属性很有可能是强烈地表示着一个病毒，一些未知检测装置的某些组成部分由可疑动态性能启动（如一个可执行代码或引导记录的写过程，或操作系统命令的异常序列包括对特殊中断信号的中途截取）；其他的会由和变化的确切本质相关的静态属性启动，这些变化是被完整性监视器确认的。

13.4.2　扫描已知病毒

如果异常探测器被触发，就会扫描系统是否存在已知病毒。由于目前已知的 PC DOS 病毒逾 4000 多种，这意味着大约 4000 种精确或者稍微不精确的匹配被并行搜索，每次匹配大约是 16 到 32 个字节。这本身是一个有趣的字符串匹配问题，有效的搜索方法是一个活跃的研究领域。但是，比起字符串匹配算法，我们更希望发明的是由脊椎动物免疫系统执行的并行搜索，大约 1 千万种不同种类的 T 细胞受体、1 亿种不同种类的抗体和 B 细胞受体不停地在体内巡逻以寻找抗原。如同一个计算机病毒扫描程序以病毒片段的匹配为基础来辨识病毒，T 细胞和 B 细胞受体及抗体利用捆绑抗原碎片来辨识抗原。

匹配碎片（而不是整个抗原）在生物免疫系统中是必要的；在计算机中，这种策略却不一定是必要的，但它有很多重要的优势，匹配片段更节省时间和内存，并允许系统辨识有轻微的变动，特别是一些不匹配被容许的时候。由变量辨识和效率所引发的问题也与生物有关。

对于生物和计算机的免疫系统来说，辨识变化的能力是很重要的，因为病毒经常有变形。如果说需要精确的匹配，那么，对一种病毒形式的免疫将对另一种稍微不同形式的病毒不起作用。类似地，疫苗将不起作用，因为它们依赖生物免疫系统的能力来合成已驯化的或已被杀死的病毒的抗体，那些

病毒和个体已经免疫的更为致命的病毒在形式上是相似的。

13.4.3 清除病毒

在生物免疫系统中，如果抗体碰到抗原，它们结合在一起，抗原将被有效地中和。这样，入侵者的辨认与中和同时发生。或者，杀手 T 细胞遇到有被某种感染载体感染的迹象，则杀死宿主细胞。这是一个相当灵敏的动作过程，因为一个被感染的宿主细胞总是要死，它被 T 细胞暗杀，可防止病毒成熟起来。

计算机免疫系统可以采取同样的基本方法来清除病毒：它可以除去或者禁止一个被感染的程序。但是，计算机病毒和生物病毒的一个重大差异是有可能会有更缓和的方法。

在生物器官中，大多数被感染的细胞都不值得去挽救，即便这是可能的，因为细胞是极易再生的资源。

比较而言，由一个计算机用户运行的所有的应用程序在功能上都是惟一的且是不可替换的（当然，除非有备份）。这样用户很容易注意到任何的功能故障，除非是那些以不破坏程序功能的方式攻击主程序的恶意设计的计算机病毒。病毒一般只是重新安排或者将它们的宿主进行可逆变形。所以一个被感染的程序通常可以表达为未受感染的一个可逆变形。

当扫描程序确认某个程序被某个病毒感染，清除过程的第一步就是确认病毒和一个已知的种类是一致的。确认是在对过滤代码区的检查的基础之上的，这些代码是根据病毒的不同实例而得知为常量的。事先必须已经获得了病毒的确切位置和结构，并且以能被确认算法理解的语言表达出来。如果确认不成功，用这种方法试图清除病毒就是冒险的，就要用到另外更一般的病毒清除方法（不在本书讨论范围之内）。如果确认成功，一个以简单的修补语言描述的修补算法将以恰当的顺序来执行清除病毒。执行步骤可以通过分析原程序的各部分的位置（也许还有变形）轻松获得。

尽管抽取确认和清除信息所需要的分析传统上是由专家来完成，在稍后的部分我们将讨论如何自动地获取这些信息。

13.4.4 诱饵

假设异常探测器发现了病毒的迹象，但是，扫描程序不能将它识别为一

种已知的病毒。大多数现有的杀毒软件都不能修复主机程序，除非它在被感染之前已经存储和分析过了。理想情况下，人们希望有有力的证据来证明系统已被感染，并得知更多关于病毒的本质，它所有的实例（不仅仅是被异常检测器发现的那个）能够被发现并清除出系统。

在计算机免疫系统中，比起异常探测器所提供的，对以前未知病毒出现的确认有更大的把握。思想是引诱病毒感染一个或更多的诱饵程序。诱饵程序被尽可能地设计为最容易传播的类型。对于病毒来说，一个应遵循的策略是感染那些与操作系统有关的程序。因为这些程序最易于被用户执行，因此也是进一步传播的好工具。所以，免疫系统通过执行、读写、拷贝或其他对引诱程序的操作，引诱一般公认的病毒并感染它们。这些活动会吸引很多活跃在内存中的病毒，甚至是在它们将控制权还给主程序之后。为了捕捉不是活跃在内存中的病毒，引诱程序被放在系统中通常所用的程序所在的位置，比如根目录、当前目录，或者路径中的其他目录，当被感染的文件再一次运行的时候，它可能会选择一个引诱程序作为牺牲品。引诱程序会不时地被检验是否被修改。一旦被修改，几乎可以肯定有未知病毒在系统中，每个被修改的引诱程序都含有该病毒的一个样本。这些病毒的样本以一种不能偶然地被执行的方式存储。现在它们就要被免疫系统的其他部件分析出来了。

引诱程序捕获一个病毒样本有点类似于巨噬细胞吞掉抗原。巨噬细胞还有别的一些种类的细胞把抗原粉碎成小的缩氨酸并且把它们放在表面上。这样它们就被 T 细胞用相应的受体束缚起来，由此引发各种进一步的事件，这些事件以各种方式在辨识和清除病原体中起重要作用。计算机引诱程序对入侵者的捕获或者生物的巨噬细胞允许它被处理成可以被免疫系统的其他部件中断的标准格式，提供了哪些部件可以获取关于入侵者信息的标准定位，并且掌握着免疫系统其他部分的动作。

13.4.5　病毒自动分析

一般来说，一个专家会把对机器指令序列的深入理解应用于病毒分析。有时，这和对计算机上病毒作用的效果的观察结合在一起。

我们的病毒分析算法在机器代码上并不复杂，但为了弥补这个不足，就要使用较多的数据，具体来说，就是数个病毒的实例。一旦捕捉到几个病毒的实例，算法将会对感染者之间，及感染者和未感染者进行比较来获得一个

关于病毒如何攻击主程序的精确描述。该描述是完全独立于主程序的长度和内容的。一个特定的简单的感染模式如图 13.4 所示。

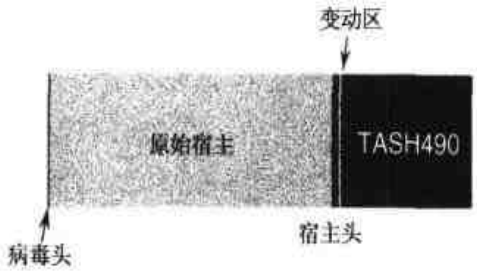


图 13.4　图示 TASH490 病毒的附件模式与结构，完全自动获得

自动病毒分析提供几种类型的有用信息：

（1）所有原主程序片在未感染的文件中的位置，不依赖于原主程序的长度和内容。这个信息被自动地转换为修复语言，这种语言是由 IBM 反病毒软件的病毒清除部分使用的。

（2）病毒所有成分的位置和结构。结构信息包括病毒各个部分的内容，这些部分在不同的实例中是不同的。这种信息有两个目的：

- 它被自动地转变为由 IBM 反病毒软件的确认部件所用的确认语言。
- 它被传递到特征抽取部件做进一步的处理。

13.4.6　自动特征抽取

自动特征抽取的基本目标是选出一个特征，此特征最有可能在所有的病毒实例中被发现，并且在未被感染的程序中被发现的可能性极小。也就是说，我们希望使虚假否定和虚假肯定最小化。虚假否定是危险的，因为它们使得用户极易被攻击。虚假肯定对顾客来说是很让人气愤的，可能会因为错误指控的软件而导致对卖主的诉讼。

为了减小虚假否定，我们先从已经被自动病毒分析程序识别的常量区的内容开始。但是，可以说样本中并不是所有的潜在变量已被捕获。一般来说，程序的非执行数据部分，包括数字常量、字符串、计算工作区等，比起代表机器指令的代码区，更倾向于各个实例之间有所差异。变量可能在病毒内部，或者病毒为了躲避病毒扫描器蓄意改变一些数据字节。保守地说，不把数据库区作为可能的特征做进一步的考虑。尽管从原则上把代码和数据区分开的任务不是很明确，但有许多方法，如用调试器或者病毒解释程序运行病毒程

序等，都表现得相当好。

在这一点上，有一个或者多个不变的机器代码序列，病毒特征可以从中被选择出来。我们把候选特征集合看做所有可能相邻的 S 字节的块，这里 S 是预先选定的或者由算法本身决定的一个特征的长度。（一般地，S 大概在 12 到 36 个字节之间变化）接下来的目标是从候选者中选择一个或几个倾向于导致虚假肯定的特征。

我们已经将最小化虚假肯定的可能性问题总结如下。对于每一个候选特征，估计它匹配一个由在相关平台的合法软件产生的概率分布产生的长度为 S 的随机序列的概率（当然，机器代码是由人或编译器而不是概率分布写的，所以概率分布是一个理论的并且有点非法定义的结构，但是我们估计从 DOS 和 OS/2 的 1 万多个程序中统计出的，包括大约 0.5GB）然后，我们选择估计概率为最小的候选特征。

稍微详细一些，算法的主要步骤如下：

（1）形成一个包含在输入数据中的所有 n 字节序列（$1 \leqslant n \leqslant N_{max}$）的列表。

（2）计算每一个这种 N 字节序列的概率。

（3）用一个简单的公式，把基于测量 N 字节的序列频率的条件概率和对每一个候选特征进行“错误肯定”的估计概率，即与从近似于“自身”的编码中选择出来的 S 随机序列的匹配概率联系起来。

（4）选择具有最小错误肯定概率的特征。

这种方法的特点表明概率估计在绝对范围之内是不好的，因为代码倾向于和比 5～8 字节长的范围有关。但是，候选特征的相关安排是相当好的，所以这种方法一般来说会选出一个最有可能的特征。事实上，从 IBM 反病毒软件相对低的错误率（相对于其他的制造商而言）来看，选择好的特征的算法能力比人类的专家还要好。

从某种意义上，特征抽取算法把过时的原理和现行的脊椎动物免疫系统对于新发现的病毒如何产生抗体和免疫细胞感受体的理论结合在一起。模板理论，从 20 世纪 30 年代中期到 20 世纪 60 年代初，认为抗体和感受体包围着抗原。**无性选择**理论认为产生大量的随机抗体和感受体，并且自知的那些随机抗体和感受体在成熟阶段被除去。在剩下的抗体和感受体中，至少有一些会匹配外来抗体。无性抗体理论在 20 世纪 60 年代得到支持，现在已经被接受。

我们的自动特征抽取理论一开始看起来像模板理论。我们在选择一个特征的初始阶段收集病毒的代码，而不是产生大量的以后可能有用的随机特征的集合。但是，我们的确用到无性选择理论中的一个重要原则：除去自我认识的特征。事实上，自动特征抽取方法在这方面甚至比起无性选择更积极，因为它只保留最好的特征。

13.4.7 免疫的记忆

脊椎动物的免疫系统对它所见到的病毒保持终生记忆的机制十分复杂，至今仍是研究和讨论的课题。

相对而言，免疫系统的记忆在计算机上实现起来却很简单。当它第一次碰到一种新病毒，计算机系统就会“得病”，比如，它会拿出相当一部分的CPU周期用于病毒的分析。分析完成之后，抽取所得的特征和确认/修复信息将被加入已知病毒库中。以后再碰到，病毒的检测和清除就会很快发生。这台计算机可以认为是对这种病毒免疫了。

13.4.8 用自我复制对付自我复制

在生物免疫系统中，带有刚好匹配一个病原体的感受体的免疫细胞会受到激励产生更多的副本。这给好的辨识器带来很大的选择压力。通过一些变异，免疫细胞一般可以产生对抗原十分匹配的免疫细胞。

这可以被视为以一种有效的方式用自我复制来对付自我复制的一个例子。还可以引用在自然和医药史上许多其他应用这种策略的例子，如20世纪50年代曾审慎地应用了多发性病毒，以截断澳大利亚兔子数量的爆炸式增长。

自我复制器本身不一定是病毒。在1966年由世界卫生组织发起的全世界范围的抵抗天花的运动中，那些和受到传染的人有亲密的接触的人都对天花有了免疫能力。这样一来免疫作为一种抗疾病在天花患者中传播。这种策略获得了惊人的成功：最近一例自然发生的天花病于1977年出现在索马里。

我们主张用一种类似的机制来镇压病毒在计算机网络中的传播，我们称之为“杀死信号”。当一台计算机发现它被感染，就会发出信号给相邻的机器。该信号告知接收者发送者已经被感染以及在检测和清除病毒中可能有用的任何特征和修复信息。如果接收者发现自己被感染了，将发送该信号给它的邻

居，依次下去。如果接收者没有被感染，它将忽略该信号，而它至少得知了数据库的更新，有效地对该病毒免疫。

理论建模表明杀死信号非常有效，特别是在高度局域化或者稀疏连接的拓扑结构中。

13.5　结论与展望

一般病毒探测器和计算机免疫系统的发展主要是由实际关心的问题推动的：人类病毒专家正处于即将被征服的边缘。我们需要尽可能将他们所做的工作自动化。

1994 年 5 月，一般病毒探测器被编入 IBM AntiVirus，从那时起，它已经成功地识别了几种新的引导病毒。这是一个未批准的专利。大多数计算机免疫系统的组成部件在我们的病毒实验室里是非常有用的原型。利用它们来处理世界各地的病毒专家从邮件里发来的大量新病毒。免疫系统本身，像它的几个部件，包括自动病毒分析和自动特征抽取，也是一个待批准的专利。

我们最终的目标是将免疫系统嵌入 IBM AntiVirus，并在几年之后，在网络中由宿于巡回的软件代理所继承。更多的实现，更多的发明部分是由生物指引的。

尽管我们实现一个计算机免疫系统的主要动机是十分实际的，但是采取一些哲学上的观点是有趣的。

考虑人类对付疾病的历史，几百万年以来，我们对付传染惟一的屏障就是我们的免疫系统，对于大多数的传染病，它工作得相当好。当我们患了通常的感冒，我们可能会有几天不舒服，那时免疫系统在努力识别并除去病毒，通常我们能抵挡侵袭。但是，有一小部分疾病，如天花、AIDS，免疫系统不能很好地对付。幸运的是，过去的几个世纪里，宏观和微观两个层次上，我们在对传染病的了解上已经有了很大的进步。在此基础上，医药增强了我们自然免疫系统的能力。

几百年以前，疾病开始在宏观层次上被认识。1760 年，数学物理的创始人，Daniel Bernoulli 对决定某种抵抗天花的接种对社会有益还是有害产生了兴趣。通过建立和求解一个数学模型，他发现接种可以平均提高三年的寿命。他的工作开创了数学流行病学领域。观察流行病学自 John Snow 开始得到了明

显的飞速进步，他在 1854 年通过在地图上定位感染区成功地减少了在伦敦爆发的严重的霍乱的感染源。

在 19 世纪后期细菌和病菌被确认为是传染疾病的原因之前，Snow 和 Bernoulli 的宏观方法被证明是有成效的。在 20 世纪，微观层次上的研究补充了流行病学。20 世纪 30 年代电子显微镜和 X 射线的出现使得病毒的结构和它们生命周期的复杂性可以被观察到。20 世纪 40 年代中期，生物化学得到广泛的研究。这些进步确立了“大陆”，在此之上数学流行病学家可以建立它们的模型。

今天，流行病学家，扮演着由 John Snow 开创的侦探性角色，发现着新病毒。生物化学家、分子生物学家和遗传学家为阐明病毒的奥秘，发明安全而有效的疫苗而工作着。流行病学家运用直觉和数学开展使人们用这些疫苗得到免疫的计划。1977 年，天花从地球上根除，这可以说是多学科合作的最大胜利。

有趣的是，人类抵抗计算机病毒的历史几乎是刚好相反的。计算机病毒最初是在微观层次上被理解的，多亏了 20 世纪 80 年代早期 Fred Cohen 的先驱工作。1987 年，第一个 DOS 病毒一出现，就被详细地分析，然后有了最初的反毒软件。1990 年才真正试图从一个宏观的角度去理解计算机病毒的传播。最终，20 世纪 90 年代中期，计划给计算机安装一个免疫系统，如同人类和其他脊椎动物依赖其为抵抗疾病的第一道防线。

当一种新的感冒横扫人群的时候，疾病控制中心并不会忙碌起来，而是集中有限的资源寻找治愈可怕的疾病如 ADIS 的良方。目前，世界反病毒研究团体花费大量时间分析计算机中像普通感冒一样的普通病毒。我们希望计算机免疫系统能够安静有效地处理大多数的标准的、普通的病毒，只留下一小部分特殊的病毒交给专家们去分析。

参考文献

[Bai75] Norman T.J. Bailey. The Mathematical Theory of Infectious Diseases and Its Ap-plications. Oxford University Press, second edition, 1975.

[CGH+95] David Chess, Benjamin Grosof, Colin Harrison, David Levine, and Colin Parris.Itinerant agents for mobile computing. IEEE Personal

Communications Magazine,1995. submitted.

[Coh87] Fred Cohen. Computer viruses, theory and experiments. In Computers and Secu-rity, volume 6, pages 22-35, 1987.

[CR94] Maxime Crochemore and Wojciech Rytter. Text Algorithms. Oxford University Press, 1994.

[ER89] M.W. Eichin and J.A. Rochlis. With microscope and tweezers: An analysis of the internet virus of november 1988. In Proceedings of the 1989 IEEE Symposium on Security and Privacy, pages 326-343, 1989.

[FPAC94] Stephanie Forrest, Alan S. Perelson, Lawrence Allen, and Rajesh Cherukuri. Self - nonself discrimination in a computer. In Proceedings of the 1994 IEEE Computer

第 14 章　控制技术的行为复制

Ivan Bratko，Tanja Urbančič 和 Claude Sammut

摘要

控制一个复杂的动态系统，比如驾驶一架飞机或一台起重机，通常需要一个技术熟练的操作员，特别对于一门需要通过经验获得且是代理认知的技术更是如此。因此，介入进去很困难，而通过自己观察，再合理重建起来更是不容易。在这一章，我们描述这样的一些实验，它们通过机器学习的方法，从操作员的控制行为中获得控制的技术，然后再重建系统。问题域包括杆平衡、起重机操作和飞机飞行。

14.1　引言

控制一个复杂的动态系统通常需要一个通过实际经验掌握很多技术的操作员。虽然这样的工作也许对人身安全是很重要或具有商业利益，但它们在执行时很大程度上是无需通过思考的，因为包含的很多技术是代理认知的。操作员通常仅仅能够不完整、大概地描述它。但这样的描述可以被用来作为建设自动管理机器的基本指导方针。正如我们所讨论的，例如在[UB94b]中，它们在被直接翻译成一个自动管理机方面不具有操作性。然而，在这一领域，自动或半自动控制的激励仍然很强烈。因为经济需求旺盛，需要给人类操作员提供辅助，使他们的工作更具有可靠性、有效性，并且在大量不同，有时甚至是在意想不到的环境下更有安全性。

Michie 等人建议使用标准机器学习技术从人们工作过程的踪迹中学习控制规则。他们宣告了在杆平衡方面的成功实验[MBHM90]。他们的先锋研究被其他从事更实际领域，如驾驶一架模拟飞机[SHKM92，MC94]，操作一台起重机[UB94b]和生产进度表[KK94]的人们所跟随。这种方法被命名为行为复制

[Mic93]。我们观察上面提到领域的一些实验，并且从这些实验中总结经验。

控制技术的行为复制如图 14.1 所示，其基本原理在一定程度上和其他技术相似，是归纳学习更“经典”的应用，比如医疗诊断。病人症状的医疗记录以及内科医生的相应诊断一起被输入进一个归纳程序，由这个程序建立规则。它以后就根据以前的数据自动地诊断新的病人。就像诊断的规则能够通过观察一个内科医生工作而被学习到一样，我们应该能够通过观察一个操作员的工作，学习如何控制一个系统。在这种意义下，提供给归纳程序的数据是通过操作员响应系统变化的行为的记录。一组归纳出来的规则组成一种分类器，它们把状态记录映射成行为名字，而不是把病人的记录映射成疾病的名称。

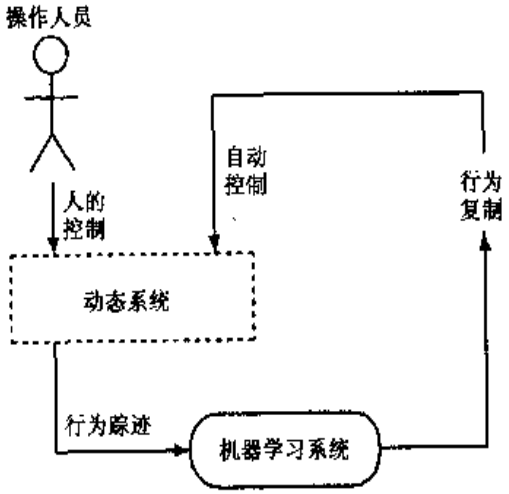


图 14.1　控制技术的行为复制

不像一些问题直接适用于标准 ML 技术，行为复制需要特别致力于把这个问题作为一项 ML 工作来格式化，包括为学习而设计一个合适的目标概念表达方式，从人类行为的记录中选择好的训练样本及解释 ML 的结果。由于缺少一般的方法论（因为手边某一特定主题必须去细心调查这些问题），虽然这种过程没有形成惯例，但在经验基础上，一些指导方针可以被用于以下领域：杆平衡，驾驶一架飞机，驾驶集装箱起重机和操作产品线进度表。所有这些领域都着眼于潜在地应用。甚至与在试验性领域很有用的杆平衡和太空船的控制也有很大关系。

在这章中描述的所有实验中，都使用动态系统的模拟装置。在[SHKM92]使用的飞行模拟器是由硅图公司提供的。在[MC94]里，作者使用的是一架 F-16 战斗机的 ACM 公共领域的模拟器。对于杆平衡和起重机控制，模拟装置在这

些研究方面很具体。在一个最近的应用[Kar95]里，控制电子发射机器过程(参考[BMK97]这一节)的复制已经被归纳出来了。如果是那样的话，复制就可以被用在实际的电子发射机器上，而不是在一模拟装置上。

14.2 行为复制

作为一个非典型的 ML 应用，一些具体特点有别于学习控制动态系统。我们有控制行为的整个结果，通常不包括明显的指示，关于

- 哪些行为对结果的成败起关键作用；
- 哪些不是最佳的但可由其他行为的效果来补偿；
- 什么是一个特别行为的实际激励物，等等。

对行为复制来说，基于 ML 的通常方法就如下所述。“行为踪迹”就是说，控制系统状态的发生顺序和操作人员的控制行为，被认为是正确控制决策的一组样本。每个样本是由一对（状态，行为）组成，状态是一个属性值向量，行为是一个学习程序的“类值”。在这样的说明下，基于属性学习技术能被应用于推导一个函数关系：

$$\text{Action}=f(\text{State})$$

状态和类值行为的属性都可以是离散或连续的。这样，学习函数 f 就可表达成一个人工控制或行为复制，这和原来的操作非常相似。

有时时间延迟被引入到状态和行为之间。原因是因为当前行为不被看做是对当前状态的操作员的反应，而是对先前的某一状态的反应。在状态和行为之间的时延认为是因为对状态的认知和操作员对控制做出的物理操作都需要一定的反应时间。在这种状态下，学习的函数关系是：

$$\text{Action(Time)}=f(\text{State(Time-Delay)})$$

在一些领域，有几个控制变量，如在飞行驾驶方面，有排气管、振动、副翼等，都可以同时操纵。在这样的领域里，学习问题被分成几个子问题，每个子问题处理一个控制变量。这些可以相互关联，所以，对一个控制变量的决定可以被用来作为其他控制变量决定的一个属性。

构造学习问题的一种方法是把行为踪迹分成几个阶段，例如，飞行可以被划分成起飞、直飞和水平的飞、转向一个特定的方向等阶段。可以从飞行每个阶段的踪迹归纳出各个独立的控制器。为了执行一个控制工作，根据飞

行计划的特定阶段，相应的控制器被激活，在这个阶段（计划和程序阶段）认知状况是人工操作，而不是自动学习的。

在以下章节，我们给出了关于在行为复制出现的现象的一种比较性分析，包括以上涉及的所有领域的结果。为了在上下文中放置结果，在表 14.1 中，我们描述了一些问题域的参数和相应的复制工作以及在每个特定情况下，人和机器学习的一些参数。这张表表明了这些工作的复杂性。然而，一些适当的评注还是需要的。一个域的一种有益特性是人类操作员需要掌握这项工作所花的时间，因为各个人之间差别可能会很大（参考[UB94b]），所以该项应该仅被当做一项粗糙的近似值。对于杆平衡，人类学习时间大概是 1 个小时，对于执行一个简单的起重机操作循环，训练大约需要 10 个小时左右，F-16 飞行问题比 Cessna variant 有更多的要求，事件的意义也应该被阐明。在飞行领域，一个事件相应于控制行为里的一个变化，但在起重机问题上，事件是真实的快照，其记录是有规则时间间隔的。

表 14.1　域特征和学习参数

	Pole and cart	Cessna	F-16	Crane
#状态变量	4	15	15	6
#控制变量	1	4	9	2
#控制变量的类型	boolean	real，integer	real，integer	integer
#主题	10	3	1	6
#踪迹	1	90	20	450
#数据组中的事件	3 500	90.000	25.000	450.000
#单个踪迹的长度	5 min	5 min	18 min	1-3min
#阶段	1	7	8	1
#学习程序	C4.5	C4.5	C4.5	Retis，M5
#时延[s]	0.4-0.5	1-3	1	0-0.1
#数据预处理需要	否	是	是	否
#预定义计划需要	否	是	是	否

14.3　杆平衡

14.3.1　问题

众所周知，关于一辆在有限长的轨道上运动的手推车上的杆平衡问题经

常被用来在控制合成里描述新手段，如在[WS64，MC68，BSA83，And87，VUF93]中所述。对于复制，Michie 及其同事在[MBHM90]中使用一种“线交叉”(Line-crossing)变量。这里，主要是在测试阶段关于路线的中线有尽可能多的交叉点，而保证没有碰撞。

14.3.2 杆的选择

有一个问题必须被解决，这就是当设计一个行为复制系统时，在所有可以利用的样本踪迹中选择哪条踪迹来学习。Michie 等人描述：

20 个心理学学生志愿者……通过试验和错误学习控制一个操作杆和手推车系统，并在一个个人计算机屏幕上动态模拟显示。这样做的目的是（1）从这项工作的三个学习标准中获得关于人类对最容易和最难的事的学习能力的想法。（2）至少找到一个主题能够用于训练到完全熟练的水平。类似于在飞机飞行员的模拟训练之前，应用预备选择过程。即通过模仿机器学习进行更集中的研究。

14.3.3 时间延迟

在一些领域里，在系统状态和控制行为之间时延好像是必需的。这是因为一个操作员不能对系统状态的刺激做出即时反应。因此关于用一组属性表达系统的状态和这种依赖变量组成的归纳程序的一个样本是被执行的行为，某些时候是在属性表达状态之后。

在杆平衡的研究里，Michie 等人做了关于时延多样性的实验，最终确定为 0.4s，这看起来是给出了最好的结果。

14.3.4 清除效果(Clean-up Effect)

Michie 和 Camacho[MC94]描述如下的清除效果：“当提取出来的归纳规则被作为一个‘自动的飞行员’安装到计算机时，执行这项工作和被训练过的人类(产生最初的行为踪迹)很相似，而且更可靠……”他们继续对这种效果做出解释：“一个训练过的人的技术被迫用一种易出错的感观系统执行，不一致性和疏忽时刻将会通过隐含在归纳概括的平均效果而被排除，这样重新恢

复实验者一个干净的最初生产规则的形式。”

Chamber 和 Michie[CM69]是最先宣告他们在杆平衡领域里发现清除效果的。Michie，Bain 和 Hayes-Michie[MBHM90]他们也给出关于清除效果的定量评价。当从一个操作员归纳出来的复制被应用于操作员行为的预言工具时，这种预言的错误率常常超过 20%。Michie 和 Camacho 简单地认为该错误率是人类感觉和执行错误的数目。这些错误都假定通过归纳程序被过滤掉(他们使用 Quinlan 的 C4.5[Qui87])。从结果上看，复制的行为就比人类操作员行为平滑，Michie 和同事[MBHM90]通过四个系统变量的访问的范围来测量清除的效果：位置、角度和它们的速率。一般来说，比较好的控制效果有比较小的范围。通过复制实现的范围比通过人类训练者实现的范围要紧得多。通过复制得到的范围降低到是起初人类的范围(依赖系统变量)的 17%到 45%之间。这个结果非常具有说明性，虽然对于这种简单的清除方法是有争议的，因为它把每个状态变量独立考虑。在[VUF93]里介绍的更恰当的函数可能是对工作情况的一种更好的测量方法，这在传统控制工程的实质里有更多的体现。

14.3.5　敏感性

对杆平衡最初的研究在测量复制(被生产出来)的健壮性上，没有做认真的尝试。这就是说，复制能够在环境中很大范围下执行它的工作。然而，后来的研究[BU]表明，可以为杆平衡建造健壮的控制机器，这是以踪迹能表示许多开始环境实例的假设为前提的。就如在[BU]中所描述的一样，从两个操作员的踪迹(在轨迹上以两个相反位置开始)Andrej Zalar 能够可靠地推导出健壮的复制，一个在 $x=2m$ 处，一个在 $x=-2m$ 处。作为一个例子，这儿规则是从树递归程序 Retis[Kar92]的踪迹上获得的，并对树做了相当程度的修剪。

If $x \leqslant -0.1259$ then $F1$ else

If $x \leqslant 0.8005$ then $F2$

else $F3$

这个函数 $F1$，$F2$ 和 $F3$ 是：

$F1=\mathrm{Sign}(0.0194x+0.6442\dot{x}+7.1169\varphi+0.8020\dot{\varphi}-0.3500)$

$F2=\mathrm{Sign}(0.5621x+0.4577\dot{x}+6.6172\varphi+0.7367\dot{\varphi}+0.0164)$

$F3=\mathrm{Sign}(1.0599x+1.5700\dot{x}+7.3821\varphi+1.4439\dot{\varphi}+2.4343)$

14.3.6 归纳规则的透明性

Michie 等人的一个目标就是“以结构化规则表达方式，传送给定可获得技术的明确数目”。他们在小数目可读规则上取得了成功。这些规则被产生并且能够成功完成这项工作。这些已经被其他研究者（上文通过控制工作列举的）分别地确认了。

归纳出来的规则的透明性仍是所有行为学习研究的一个目标。然而，我们将会看到，在更复杂的领域里，仍然有很多的工作要做。

14.4 学习飞行

14.4.1 问题

有两项研究已经被报道过了：飞行 Cessna[SHKM92]和 F-16[MC94]。在这两种情况里，任务就是根据预先定义的飞行计划飞行，包括起飞，爬升到一定的高度，直飞和水平的飞，转向和着陆。

在两项研究里，飞行模拟程序的原代码被修改成飞行员的在许多次模拟装置器飞行期间的行为日志。这些日志用来通过推导建造一组规则，这组规则能够通过与飞行员用的同样的飞行计划来驾驶飞机。收集的数据由特别控制的调节点，例如升降舵、副翼、震动等和模拟装置状态变量的值，如倾斜度、翻滚、盘旋、爬升速度、航行速度等组成。

每次飞行的数据被分割成许多不同的阶段，例如起飞和爬升、直飞和平飞、转向、降落和在跑道上滑翔、着落。对每个独立于决策树中的阶段，关于每个可能的控制行为都会被建造，如：升降舵、副翼、振动等。一个程序过滤飞行日志为每个控制行为的相应归纳程序产生输入文件。一个训练样本的属性就是模拟装置的飞行参数。这种依赖变量就是描述一个控制行为的属性。

有几个归纳程序已经被使用，包括 C4.5(Quinlan，1993)和 CART 回归树算法(Breiman 等人，1984)。为了检测归纳规则，原来在模拟装置器中的自动飞行码被规则取代，一个快速处理器把归纳程序 $\tilde{O}_S$ 的输出转化成用 C 语言描述的 if 语句，这样他们就能很容易地被嵌入到飞行模拟装置上，人编的(Hand-crafted)C 语言编码决定飞行已经达到哪个阶段，并决定什么时候改变阶段。在转换语句里，都会选择一种对每个阶段都适宜的规则。每个阶段有几

个独立的 if 语句，一个行为对应一个。那些已经被合成在一起的规则感觉上很成功，据飞行计划飞行就像人类训练者执行一样，并且能在跑道上安全着陆。

当应用 C4.5 到这个领域，有一个技术问题不得不克服，控制变量是实型数或整型数，然而，C4.5 仅能输出离散的类值。为了处理这种情况，连续的类变量被预先离散化以符合 C4.5。离散化被用在聚集类值的频率直方图的基础之上。实验也被用来处理输出连续值的回归树。这结果可和 C4.5 相比较。

在这个领域行为复制的动力看起来好像很值得怀疑，因为它是在和已存在的非常好的自动飞行员相竞争。然而，复制仍然是值得做的，至少有以下原因：

（1）用符号的形式重建该技术，这样它就是可检查的了。

（2）在传统方法失败的情况下，建立自动飞行员（Autopilot），如处理风剪。

14.4.2　样本选择

不同的飞行员之间，风格和控制策略有非常大的不同[SHKM92]。例如，一些很呆板（Heaby-handed）的飞行员为了保持飞机能正常工作而大量地练习控制。而其他人使用一点纠正方法，最终也能安全返航。商业航空公司喜欢第二种类型的飞行员，因为他们可使乘客们尽量减少不舒服的倾向。

来自不同飞行员的混合数据有“混淆”归纳算法的倾向，因为决策树不得不负责实现目标的不同路径。因此，自动飞行员是从单人飞行员的踪迹中构造出来的。这些自动飞行员与训练者的风格极其相似，因此术语“行为复制”是 Donald Michie 创造的。

14.4.3　时间延迟

Sammut 等人[SHKM92]在他们的实验中，使用了在 1 至 3 秒之间变化的延迟。他们时延的选择是实用的并且是由实验决定的。这种选择不是通过原则性的方法做出的，虽然在他们的讨论中，他们花了相当的注意力在这个问题上。他们相信，这种时延是很重要的，假设某种延迟是存在的。因为飞行数据带有一定延迟地被记录下来，归纳出来的规则仅当时延也存在于规则执行和根据规则定义的行为执行之间才有效。

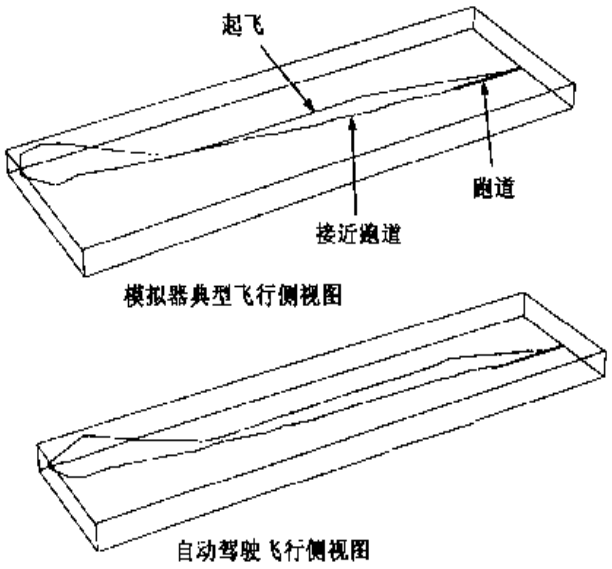


图 14.2　人执行的固定飞行计划(上面的轨迹)和模拟器执行相同的飞行计划 (下面的轨迹)

14.4.4　清除效果

Michie 和 Camacho [MC94]也做了有关 F-16 模拟飞机飞行清除的报告。他们以飞机偏离直线的偏差作为飞机在直线和水平方向的误差。模拟器的偏差只是人为偏差的 15% 左右。

在学习飞行 Cessna[SHKM92]时，也可观察到清除效果。图 14.2 表示两架飞机在三维图上的轨迹线，一架由人操作而另一架由模拟器按照相同的飞行轨迹来操作。两架飞机均成功地完成了相同的飞行计划。清除效果在飞机接近跑道时很明显。

14.4.5　敏感性

成功的模拟器同样可运用到有关人的操作项目，虽然模拟器的轨道当然并不是原始轨道的简单再现。然而，在复杂的领域中，例如飞行和起重操作，最初的实验产生的模拟器通常对有关变化十分敏感，而且只在一个确定的计划里面是成功的。它们对特定任务的微小变化及该问题领域内的参数是十分敏感的。

这些问题可以在噪声环境中进行训练矫正。如，最初的领航模拟器是利用不包括湍流及风力干扰的飞行模拟器建立的。因此，在直线和水平飞行时，

飞机将沿它原来的高度和方向飞行而尽量不理会模拟器的控制。这样的飞行轨迹不能提供诸如，当飞机开始沿期望的路线飞行时该做些什么或若它偏离路线时该做些什么等。这可以引进扰动和风的漂流来加以矫正。人类的飞行员现在必须采取矫正的实例来训练更完善的模拟器。Arentz [Are94] 正是进行了这样的实验而发现模拟器可以构造得相当完善而坚固。显然，如果模拟器遇到的干扰强度超过训练样本遇到的干扰强度，那么由于所创造的环境超出了模拟器经历的范围，我们可以认为模拟器将会失去控制。

14.4.6　归纳规则的透明度

目前，操纵模拟飞行器所产生的规则庞大且迟钝，为什么会这样，仍然有许多的猜测。

用于归纳的属性仅仅是模拟器的不同状态而已，它们并不一定反映飞行员做出控制决定时注意的是什么。特别地，关于模拟器所显示的可视场景，他们并不提供任何信息。建立在简单属性上的高级属性，可能导致更简明的决策树。

尽管从这些实验收集的信息总量是巨大的，但相对于领域范围来说，数据仍然相当稀少。在这样的环境中，采用归纳程序以找出无关量是可行的，这样，偏离标准的影响会减小。当前进行中的实验是为了要确定其影响有多大。

当一架飞机尽量在期望高度下直线和水平飞行时，其目的往往是为了得到正向爬升率。这可能是由包括渐增的降速、仰冲动作组合起来完成的。目前的模拟方法没有考虑这一由两个阶段组成的过程制定目标以及决定如何实现它。这样，另外一个关于这个庞大的决策树的推测是目标构成及实现控制行为存在混乱。

14.5　集装箱起重机

14.5.1　问题

世界市场需要尽可能高容量的起重机。这里，容量是指在一定时间内所传送的总负荷。发展更大更快的起重机是一种解决方案，但由于高速运行的

不良后果，使得这种发展在实践中受到限制[Gri90]。替代方案是更好地利用已有的起重机，尤其是需要在最短的时间内将集装箱从起始位置传送到目标位置的时候。这需要很好的操作协调性：放置吊车并调节吊绳长度。对目标负荷位置上面的吊车做恰当的定位可以避免后来的纠正，因为这些纠正则会大大地影响运行性能。无论是对安全或效率，控制负荷的摆动也是非常重要的。反摆动能力将产生更大的加速度，若加以适当地控制，在运转周期的某些阶段，负荷的摇摆是有利的，并被熟练的操作工在实际中加以利用。当然，接近目标位置的时候，摆动必须尽可能保持小一点。为了使操作员成为高效、可靠的起重机吊运工，需要进行大量的训练，而他们其中的一部分人从未达到最高的水平。长时间地用同样的注意力来吊运也是很困难的。所以，在此过程中，计算机的协助将会十分有益。

我们知道集装箱自动化操作方面已经有一些尝试，但据起重机建筑工业专家们的[Nov]看法，目前还没有像训练良好的人类操作员那样好的控制器。举例来说，有一些研究采用经典的自动控制方法，如[SS82]。然而，这种方法对不可预测因素的出现的处理效果并不理想，控制器很迟钝。这些都促使人们寻找另一种方法，如模糊预测控制[YH86]。这种方法包括如描述人工操作策略，定义语言执行索引的含义等步骤。这个过程可能非常耗时，因为综合控制规则主要是直接归纳训练良好的操作员执行记录。

在行为的复杂研究中[UB94a]，任务是从一个初始的位置将集装箱传送到一个目标位置。执行的要求包括基本的安全限制，临时停靠的准确性和尽可能高的容量。一群志愿者首先学习控制集装箱起重箱的模拟器。然后控制规则由两个衰退子程序 RETIS[Kar92]和 M5[Qui93]根据对象的行为记录而生成。成功运行的模拟器正是由 RETIS 和 M5 引导的。

14.5.2 选择例子

在这个领域中对象的类型和控制策略是变化的，同飞行中的差不多。举例来说，一些对象趋向于快速而不太稳定的操作，其他的则慢一些，但却更保守、更稳定。有一些对象则避免以时间为代价的大的角加速度，这样的策略产生稳定但缓慢的性能。这同趋向于得到快速时间的操作者策略形成对比，但是需要吊车更大的加速度——这将引起更大的角度，并需要负荷在轨迹最后处达到精确的平衡。每个操作员达到子目标的次序也有不同之处。

为了要避免混合个体风格，普遍认可的行为模拟练习就是综合了对同一对象训练的例子。然而，甚至同一对象的样本轨迹可能变化较大。例如，在起重机领域中，同一对象使用相同的控制方式将形成与完成时间相去甚远的轨迹。依照这一点，当尝试做最“模范”的及无冒险的模拟器时，到目前为止，最保守的榜样轨迹被认为是最无用的。

14.5.3　时间延迟

利用不同的延迟[UB94a]做各种实验。在这个领域中，零延迟与其他延迟相比可能不会产生更差的结果。一些讨论认为操作员事实上有不同的延迟是依赖于实际位置。那些快速反应的决定和关键的长期决定都与设置新的中间目标有关，如开始加速直到达到目的速度。同时，一个熟练的操作员，可能具有通过对系统状态的短期预测进行反应时间延迟矫正的能力。在极端的情况下，甚至可能会显示“负延迟”。从心理学知道，对不可预知的事件，直接采取反应时间的延迟并不恰当。结论是，无论是理论上的还是实验上的，似乎没有明显的迹象可以表明什么才是恰当的延迟。

14.5.4　清除效果

在起重机领域中同样也发现清除效果。在集装箱起重机中，有六个系统变量：吊车的位置和它的速度，绳子的长度和它的速度，绳子的倾斜角和它的速度。任务是要将负荷从一个起始位置传送到一个给定的目标位置。当到达目标位置的时候，在最小的时间间隔内，状态变量必须尽量保持接近目标值。

图 14.3 显示了两种表现，一种是由人类控制，这精确地包含了在复制中所要用到的那些事件，用此来生成其他轨迹；复制是由 Quinlan 的 M5[Qui93] 所引起的，该 M5 是以默认彻底的剪枝生成的回归树。在前面的例子中，从图 14.3 可以看出复制产生的任务在风格上与原体极其相似。就需要在规定时间内完成任务而言，在这里清楚地反映了复制明显要比原体好得多(复制需要 75 秒，而原体需要 90 秒)。

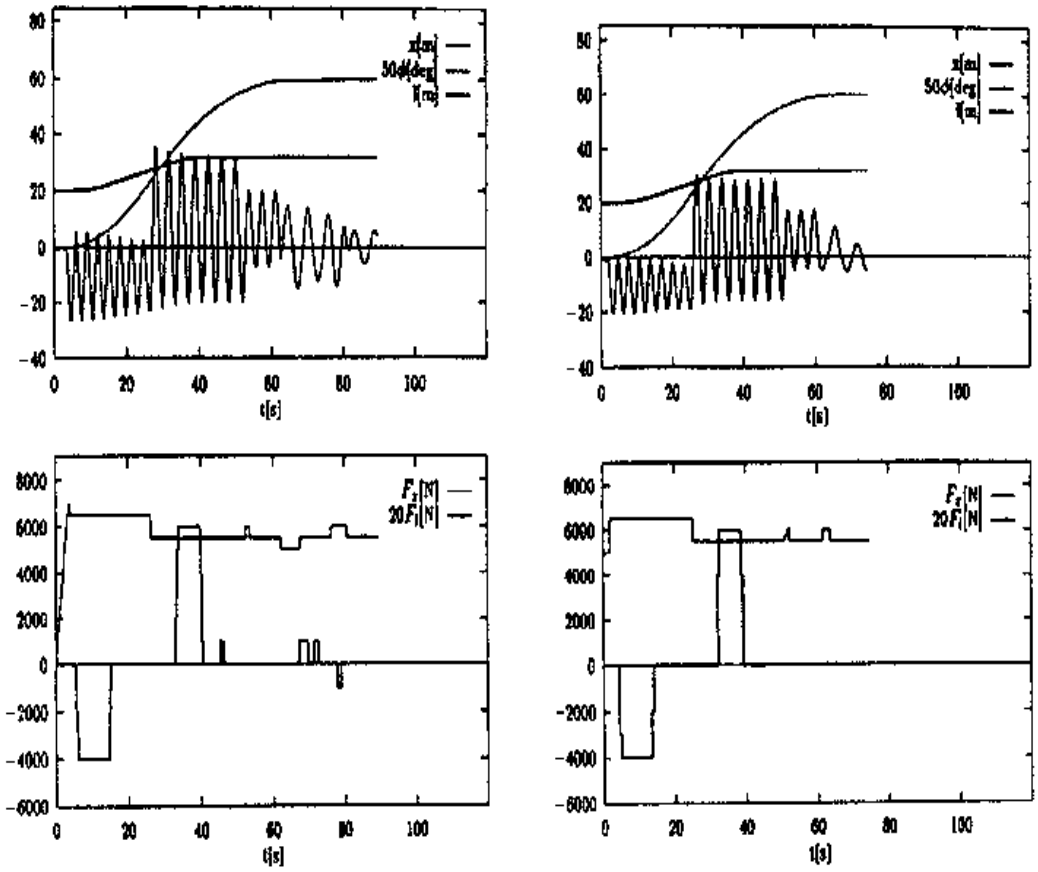


图 14.3　左边的图：由人控制的起重机一个最“典型”的轨迹；右边的图：人控制轨迹导出的起重机模拟轨迹。变量：x 是吊车的位置，ϕ 是绳子的角度，l 是绳子的长度，F_x 是施加于吊车的水平分力，F_l 是施加于绳子的垂直分力

14.5.5　敏感性

相对以上最优的结果，在这个域中进行实验查找，仍然有许多不理想的方面。敏感性分为两个层次：

- 直接按轨迹学习来复制所产生的结果并不可靠，它对于选择特定的训练轨迹和学习程序设置很敏感；
- 复制结果对于控制任务中的变化没有较好的鲁棒性（例如开始和结尾位置的不同），就算它们在规定时间内完成任务这一点上表现得比原体还要好。

虽然从多个轨迹复制的结果具有较强一些的鲁棒性，但是减少透明度的代价是树的维数的增加。

14.5.6　推导规则的透明度

从研究[UB94a]中,我们特别关注了透明规则的生成，这将有助于将人的技能用符号的形式表现出来。将一个出色的操作者的技能传授给一个天分平平的人具有重要的实际意义。作者认为行为复制的主要任务在于建立起由操作者轨迹而得到的复制结果和操作者指令之间的联系。从而他们可以用所产生的复制结果与操作者对于他们控制技能的口头描述进行比较。

关于控制作用在绳子上的力的直线学习，F_l，使用原始的六个属性来产生回归树使我们很难从概念上与指令做比较。树会倾向于变得非常的巨大，而在树中出现的指令与人类指令会具有差异。然而，通过指令的帮助，很有可能找到一组属性使得学习系统导出一个更加易懂的复制结果，如图 14.4 所示。

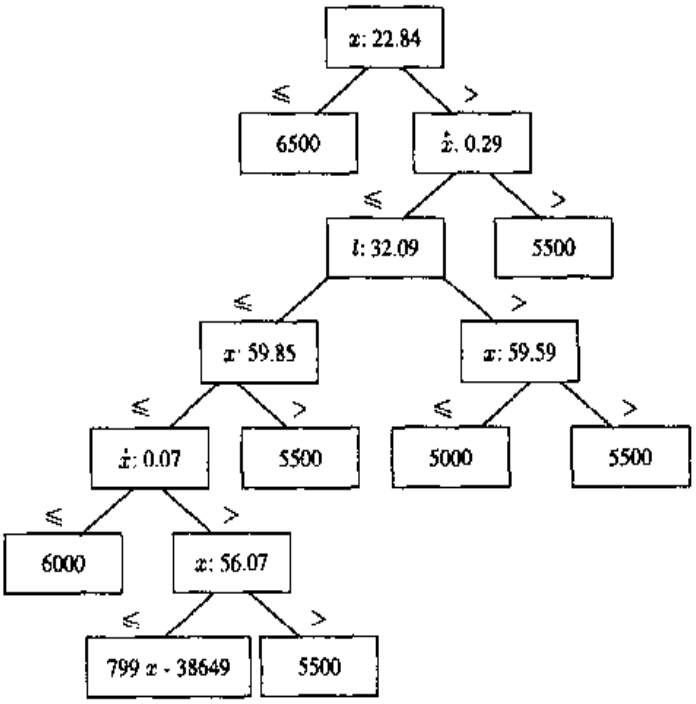


图 14.4　图 14.3 中样本轨迹推导出的 F_x 的控制器

在作用于小车的力 F_x 的作用下，要在复制结果和指令之间建立概念的相似性则更加困难。例如，涉及时间排序的指令在复制中都没有被提到。因此，树的不同部分在重演中执行的时间顺序被考虑了进去，以此来判断其与口头指令相对应的部分。这种比较有助于口头指令的操作化。这些指令直接翻译成控制程序通常显得不完整及不够精确。特殊情况下人所给出的一些域值非

常不精确，但是在复制的帮助下可以被精确地识别。此外，基于 ML 的分析，这个实验揭示出操作者事实上并非在做他所认为他在做的事情。

分析产生了一个新的行为复制模式，它包括了操作者不完整的口头指令用做背景知识的来源。这将在本质上改进机器学习的结果，反过来可以更好地理解以及改善指令本身。

14.6 生产线调度

14.6.1 问题

Kerr 和 Kibira[KK94] 指出“制造系统的复杂性和多变性使得利用分析方法发展自动调度变得困难”。因此，他们尝试着用行为复制的方法为一个电话制造系统调度资源。问题是为生产线的一段时间任何时候的轮班决定一个最优或次优的劳动力分配方案。由于要求在分割为几个小时间隔的时间点上进行控制决策，所以时间段是不规则的。此过程如图 14.5 所示。在制造过程的每个时间段之间，一个阶段成分输出被储存在缓存中，以备下个阶段作为输入使用。怎样分配劳动力的关键属性是缓存的层数。建立 4 个决策树，分别建立不同的劳动力要求。

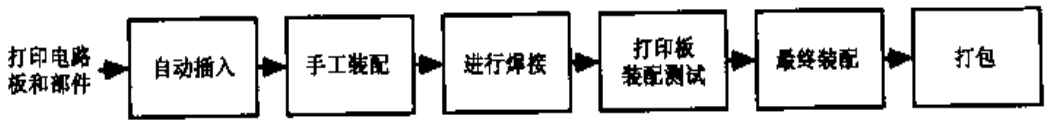


图 14.5 制造过程的基础阶段

14.6.2 样本的选择

用于推导的各个样本集是从经过在一个模拟工厂环境下进行过专业调度训练的独立调度处所得出的。

14.6.3 时延

因为调度决策是根据规则的若干小时为间隔的时间段而制订的，所以时延在这个问题中并不是一个重要的因素。决策是基于缓存的层数和编排时间

内的生产量的。

14.6.4　清除效果

在这一领域中，目标是在 8.5 小时间隔的终点，现存的子生产线上不同阶段的部件缓存层次数可以尽可能详细地说明目标层次数(在 Kibira 的实验中为 500)。时间间隔起点的缓存层次数大都源自于目标层次数。Kerr 和 Kibira[KK94] 给出了人类专家和复制体在流水线上不同点的长度。清除的效果反映了事实，从而复制最终层次数(每隔 8.5 小时) 总是像人一样接近 500。

14.6.5　敏感性

人类调度者在不同情况下会使用不同的方法。这就产生了一个具有鲁棒性的自动编表器。可以做实验来比较由人参与控制的自动编表器和一个分析推论性能分配器的效果。在面临实际制造系统通常都会出现的变量时，分析推论解决方案很脆弱。而这种情况下，复制的调度器的效果至少和人差不了多少。

14.6.6　归纳规则的透明度

这个任务中推导所生成的规则可以为人类专家所理解。与这一章节所讨论的其他领域相比，这里的属性相对简单、依赖性小。

14.7　讨论

在本文中描述的行为复制的实验指出混合方法学的一些要素，我们在接下来的几段中予以讨论。

训练实例轨道的选择

不同操作者的风格和控制策略明显地不同。为了避免混淆个人风格，在行为复制中被普遍接受的惯例仅用于训练同源实例。尽管如此，同源实例的轨道仍可能存在相当大的不同。而当试图引发最可信以及“最不冒险”的复制时，这一最谨慎的轨道实例却被发现与最有效的复制相差甚远。

状态与行为之间的时延

人类对于突然刺激的反应时间并不一定能在行为复制中寻找一个合适的延时。一个合理的方法是首先试用零延时，然后逐渐增加延时，力求获得最佳的性能。

设计陈述

也就是在选择属性时，考虑操作者对他或她的技能的口头描述是有益的。对“复制循环”（Cloning Cycle）这一描述的介绍在[UB94b]中讨论。

在此之前的所有工作中，复制遵循的形式是决策树、回归树或规则集。这些复制是纯粹反馈的，并作为人类技能（除非嵌入手工计划）的概念化是被不完全建立的。它们在人类控制策略中缺少典型的概念结构：目标和子目标，步骤和因果关系。复制的简单形式，如从系统陈述到行为的映射，并不足以表达这样一个概念结构。复制和人类对他们技能的自身描述在概念上的区别将在[UB94b]中予以分析。

这里，我们附注对类似人类的控制者(Human-like Controllers)的一些要求。首先，这样的控制者应当有一些内部记忆用于保持当前目标及任务步骤。此外，为了能够使学习程序发现行为轨迹的概念结构，此程序应该可以读取这一领域的背景知识。

Leech[Lee86]在他的改进控制处理领域的工作中发展了两步法，即首先归纳识别出输出的临界变量；再进一步归纳出构造控制规则来满足想要得到的临界变量值。Michie[Mic]建议有一种类似方案可被用于行为复制。这种方法目前正处于在领航领域的研究中。原始结果已显示出这种方法确有可能用于构造更多以目标为中心的复制。尽管如此，我们还有许多的工作要做。

不管陈述是怎样的，我们相信对于人类技能的研究应当考虑到人类可忍受的局限。其中一方面的局限在于人类一次只能观测到少量的状态变量。这里是我们的起重机操作者的意见：“在这一步，我只能看到 x 非常小，我从没有看到 $\dot{\theta}$。[之后]在这里我从未看到 $\dot{x}$，如果我看到 $\dot{x}$，我会感到非常迷惑”。这说明了在任何给定时间里，操作者的决定仅依赖于非常少量的属性：一个人类策略非常重要的方面是了解在任务的不同阶段需要看到什么仪表（Instruments）。

致谢

我们感谢 Donald Michie 为我们的实验给出许多有益的论述和建议。我们感谢 Marko Grobelnik 为实验提供起重机模拟器以及 Matjaž Siegl 在起重机实验中的帮助。感谢 Mike Bain 在杆平衡实验中的鲜明的论述。

参考文献

[And87] C.W. Anderson. Strategy learning with multilayer connectionist representations. In P. Lan-gley, editor. Proceedings of the 4th International Workshop on Machine Learning, pages 103-114. Morgan Kaufmann, 1987.

[Are94] D. Arentz. Experiments in learning to fly, 1994. Computer Engineering Thesis, School of Computer Science and Engineering, University of New South Wales.

[BMK97] I. Bratko, S. Muggleton, and A. Karalic. Applications of inductive logic programming.In Machine Learning and Data Mining: Methods and Applications. Wiley, 1997. (This volume.)

[BSA83] A.G. Barto, R.S. Sutton, and C.W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. IEEE Transactions on Systems, Man and Cybernetics,SMC-13(5):834-846, 1983.

[BU] I. Bratko and T. Urbancic. Skill reconstruction: machine learning vs. handcrafting. In D. Michie, S. Muggleton, and K. Furukawa, editors. Machine Intelligence 15. Clarendon Press, Oxford. To appear.

[CM69] R.A. Chambers and D. Michie. Man-machine co-operation on a learning task. In R.D.Parslow, R.W. Prowse, and R.E. Green, editors. Computer Graphics - Techniques and Applications, pages 179-185, 1969. Plenum Press.

[Gri90] D. Griffiths. Virtuoso performance: Pdi container crane survey 90. Cargo Systems, 17:XI-XII, 1990.

[Kar92] A. Karalic. Employing linear regression in regression tree leaves. In Proceedings of the 10th European Conference on Artificial Intelligence, pages 440-441. John Wiley & Sons,1992. Vienna, Austria.

[Kar95] A. Karalic. First Order Regression. 1995. Ljubljana University, Faculty of El. Eng. And Computer Sc.: Ph. D. Thesis.

[KK94] R.M. Kerr and D. Kibira. Intelligent reactive scheduling by human learning and machine induction. In IFAC Symposium On Intelligent Manufacturing, 1994. Vienna.

[Lee86] W.J. Leech. A rule-based process control method with feedback. In Proceedings of the ISA/86 International Conference and Exhibit, 1986. Houston, Texas.

[MBHM90] D. Michie, M. Bain, and J. Hayes-Michie. Cognitive models from subcognitive skills. In M. Grimble, J. McGhee, and P. Mowforth, editors, Knowledge-Based Systems in Industrial Control, pages 71-90, Stevenage, 1990. Peter Peregrinus.

[MC68] D. Michie and R.A. Chambers. Boxes: an experiment in adaptive control. In E. Dale and D. Michie, editors. Machine Intelligence 2, pages 137-152. Edinburgh University Press,1968.

[MC94] D. Michie and R. Camacho. Building symbolic representations of intuitive real-time skills from performance data. In K. Furukawa, S. Muggleton, and D. Michie, editors. Machine Intelligence 13, 1994. Oxford; Clarendon Press.

[Mic] D. Michie. Personal communication.

[Mic93] D. Michie. Knowledge, learning and machine intelligence. In L.S. Sterling, editor, Intel-ligent Systems, New York. 1993. Plenum Press.

[Nov] A. Novak. Personal communication.

[Qui87] R. Quinlan. Simplifying decision trees. International Journal of Man-Machine Studies,27(3):221-234, 1987.

[Qui93] R. Quinlan. Combining instance-based and model-based learning. In Proceedings of the 10th International Conference on Machine Learning, pages 236-243. Morgan Kaufmann,1993.

[SHKM92] C. Sammut, S. Hurst, D. Kedzier, and D. Michie. Learning to fly. In D. Sleeman and P. Ed-wards, editors. Proceedings of the Ninth International Workshop on Machine Learning,pages 385-393. Morgan Kaufmann, 1992.

[SS82] Y. Sakawa and Y. Shinido. Optimal control of container crane. Automatica, 18(3):257-266,1982.

第15章 空中交通控制一阶知识的获取

Yves Kodratoff 和 Christel Vrain

摘要

本章介绍了在空中交通控制领域中，基于知识的归纳是如何应用于知识获取的。在这里的应用领域中，我们解释为何基于知识性和一阶逻辑有时是必要的。一阶知识的一个很明显优点就是表达力强，但其缺点是需要大量的计算时间。本章还介绍了一些有关一阶逻辑其他不太明显的优点和缺点，特别是在必须通过 Horn 子句来表达知识时，仍然具有一定的计算效率。最后，本章强调亟待解决的大量的翻译问题，以便能够与专家进行有效地交互。要有两个翻译阶段：一是从专家语言到 Horn 子句；二是再从 Horn 子句翻译回专家语言。前者对于确保自动学习是必要的，而后者将使专家理解所学到的知识。两个阶段都非常重要。要求仔细选择以免丢失重要的信息。我们不期望的结果就是第二阶段的翻译扮演确认的角色，因而成为获取专家将问题形式化的相关知识的一种有效方法。利用一阶逻辑可以做些复杂的事情，结果是可得到一种抽取和确认已获取的知识的强有力的方法，特别是在领域专家不能用简单方式表达知识的情况下。

15.1 引言

在空中交通控制（ATC）领域中，显然，其中心问题就是避免飞行器的碰撞以及处理这一问题所涉及的大量工作。每个控制器负责空间的一部分，称为区域。实际上控制器忽略了超出它所控制的区域以外的一切情况。当飞机进入它的区域时，它负责确保此飞机不与已在区域内的其他飞机相撞。如果察觉可能发生碰撞（称为冲突），它必须修正飞机的轨道，以解决此冲突。一些人（如 Shively 和 Schwamb,1984; Planchon 和 Berrada,1988）已对控制器的

任务通过专家系统（ES）方法建模进行了尝试。然而必须指出的是，就 ATC 而言，这种技术的不同应用似乎无法做到自动化，这说明许多问题仍需要解决。这一领域的复杂性和专家知识表达形式的缺乏，阻碍了我们对专家行为的正确建模。PERSPICACE 工程（Cannat 和 Vrain，1988）的目的就是将机器学习（ML）技术应用于学习 ATC 规则，以获取通过与专家讨论不能获得的规则。对专家而言，为自己的行为给出例子似乎要比给出规则容易一些。因此我们计划使用有关概念学习的一些机器学习（ML）成果（Michalski，1983；Kodratoff 等人，1984；Kodratoff 和 Ganascia，1986；Kodratoff 和 Tecuci，1988）来学习专家行为的规则。

为专家系统（ES）学习规则，传统的知识获取（KA）的办法就是通过与某一领域专家进行交互。相反的是，我们的方法是通过观察好专家的行为顺序组成的，在我们实例中指的是一个空中控制员所采取的步骤以解决冲突的情况。我们的系统对每种行为都起作用的条件下对这些例子进行总结，由此获得的一般性规则还需要由领域专家来认可。如果认可了，则此学习完成，规则插入规则库。如果专家拒绝这一规则，则需要专家提供解释，以新的正例或反例，或通过改进背景知识（BK），或通过完善概念的形式给出。

因此与专家的对话分为四个主要步骤。在过程开始时，我们需要三个步骤来获取背景知识(BK)，以确定要学习的概念和收集描述概念的正例和反例。总结过程是归纳性质，不能保证其结果的正确性，因此我们需要第四个步骤来确认已学习的知识。在本书中我们将不讨论此问题。解决它的最简单方法就是将结果向专家求证，但效率比较低；另一种方法是在实例中检验。某些情况下，如由于在实例中，或背景知识中的错误，或由于所使用的表示形式，导致一些结果错误，目前我们还没有方法能够自动修正。我们必须通过与专家的相互讨论来发现这些错误。

通过总结进行学习的有关方法如图 15.1 所示。

事实上，机器学习技术的应用并不像图中所示的那么简单。收集实例对于一个专家来说是很容易的任务——在 ATC 应用中，一个月的工作可以交付大约 100 个例子，但建立背景知识则是个困难的任务。除了与专家对话，还要寻找一种表达语言，它适用于所研究的领域和归纳工具，并可适合表达背景知识和实例。在第 15.3 节中，我们介绍有关将把背景知识和实例翻译为一阶逻辑形式，再把结果翻译回领域专家能理解的语言的问题。

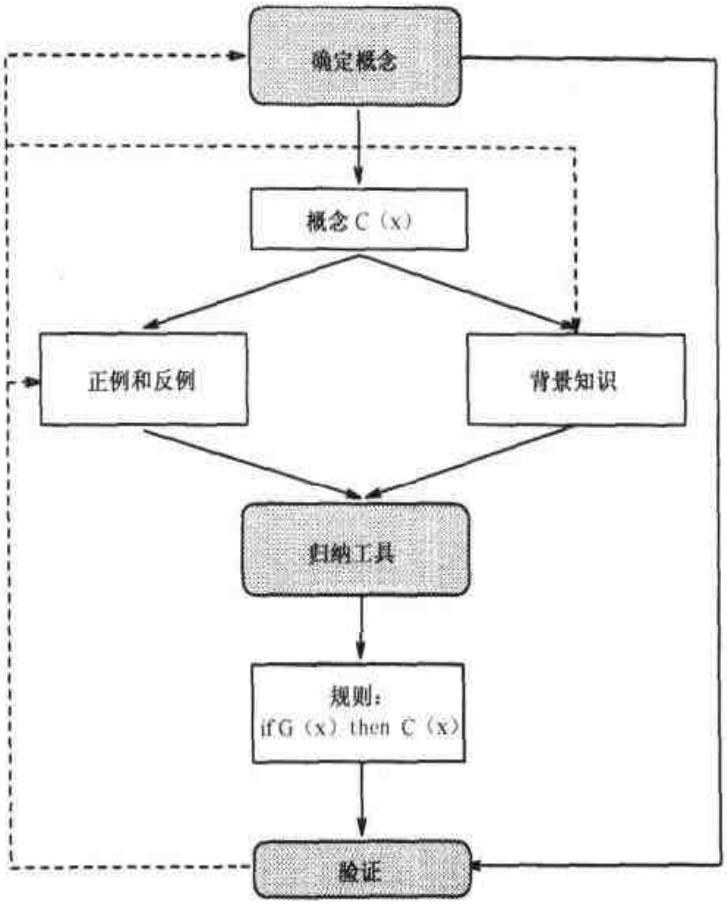


图 15.1　知识获取/验证的循环

在本节中，我们意在介绍已出现的问题，其中大部分是因使用一阶逻辑的需要而提出的。它们已被部分解决，我们将解释所采用的解决方法。所学习的知识可以有许多不同的形式。我们的系统学习由规则所表达的知识，如以下所示：

If<条件>then 执行<行为>

这里所采取的动作常常是向数据库中添加新的事实。在基于规则的专家系统中，有不同的方法来表示条件和规则结论（Shortliffe，1976；Forgy 和 McDermott，1977）。最常用的是基于逻辑形式，命题或一阶逻辑。遗憾的是，没有一种表示是完全令人满意的。零阶逻辑能快速推理，但有局限性；而一阶逻辑有很强的表达力，但在知识库和数据库进行匹配操作时可能代价很高，

主要表现在使用前向链式推理机制中。为了避免其中一些问题，人们常使用这样一种表示（属性，值），它可被视为零阶和一阶逻辑之间的中间体。在这种表示中，很难表示出不同对象和这些对象间的关系。在我们的应用中，不得不选择一阶逻辑形式，我们将解释做出这种选择的原因，并说明由此出现的问题。

本章使用了一种归纳技术，它被称为**基于知识的关系归纳**。我们现在介绍什么是关系归纳，强调其在准确表达所学知识方面的重要性，说明其在知识获取时所出现的新问题，也就是由专家妥当定义谓词和“Skolem 函数”（引号内是必需的，稍后将解释）。

15.2 基于知识的关系归纳

我们已提到了专家系统在关系逻辑表达能力与计算速度之间进行平衡的问题。这一问题在机器学习中也表现出来。例如，在基于相似性的学习中，一阶知识能代表包括不同对象和它们之间关系的复杂事件，但基于知识的归纳工具在实现上就有困难，且不得不面临组合爆炸的问题。

我们现在给出一些有关利用零阶与一阶逻辑进行表示问题的更详细的说明。这将解释为什么我们在 ATC 应用中需要一个关系表示方法。

15.2.1 零阶与一阶表示的对比

15.2.1.1 变量

在理论上，一阶逻辑允许存在全称量词变量和存在量词变量。例如，考虑一个化学分子的描述。一些分子可能具有所有碳原子都至少与一个氢原子相连的性质。这是一个只能用全称量词变量描述的典型性质。另一些分子可能被描述为至少包含一个氧分子。这是一个只能用存在量词变量描述的典型性质。

在实践中，我们想要从这里吸取选择这些表示的后果。更一般的情况下，一阶逻辑可以很好地表达这些关系，而零阶逻辑在这一点上的表现是很差的。这个例子说明选择一阶逻辑具有较高的应用价值：满足一定关系的实例的所有组合都做了尝试。因为穷举了所有要识别的模式，该特征都是符合需要的。

在有太多组合时，该特征也是无法满足的。这时就不得不选择启发式方法来减少其复杂性。

在学习上下文中，在构造一个识别函数时就需要进行这种组合搜索。例如，假定我们要学习父亲（概念），我们有三个父亲（概念）的例子：John,Peter 和 Tom。假定 John 和 Peter 有 A 类成绩的孩子，而 John 和 Tom 有对运动着迷的孩子。如果我们开始时只归纳 John 和 Peter，则我们将失去 John 的孩子也对运动着迷的信息。归纳到一阶技术必须包括顺序选择归纳例子的优化机制。

所以，一阶逻辑的主要应用特征（相对于零阶表示）就是：

（1）它允许特征之间关系的表达；

（2）它考虑了表达相同关系的可能的谓词组合以便产生识别函数。

在 ATC 的应用中，我们只有很少的对象，最少只有两个冲突的飞行器，但我们有它们之间的很多关系。可能甚至其他的飞机也会对专家的决定起作用。专家将会忽略一些关系：他将仅表示目前所关心的一些知识。这意味着我们不能表达在例子最初表示中的所有知识，我们需要定理，以修正专家对例子的描述。此外，大部分专家知识是一阶知识，例如速度、时间和距离的关系。

15.2.1.2　函数

一阶逻辑的表示能以函数的形式给出。例如，考虑速度、距离和时间的关系。我们不能用零阶逻辑形式来表示，但我们能用一阶逻辑表示，如下所示

$$\forall x \forall t[\text{distance}(x) \wedge time(t) \Rightarrow speed(x/t)]$$

也就是，如果距离为 x，时间为 t，那么速度为 x/t。

事实上，我们应该看到：在 ATC 应用中，我们不能利用上面那一点，因为我们目前的归纳工具的版本不能处理数字量。数字量将由符号量来代替，我们使用一种原始的定性系统来实现这种计算。例如，我们用一种符号来代替数字操作/，记作$/_s$，定义为：

$/_s$(large,small)=large,

$/_s$(small,large)=small

否则：$/_s(x,y)$=undefined

我们给出两个定理来表达$/_s$图：

$$\text{distance(large)} \wedge \text{times(small)} \Rightarrow \text{speed(large)}$$

$$\text{distance(small)} \wedge \text{times(large)} \Rightarrow \text{speed(small)}$$

15.2.1.3 子句和 Skolem 化操作

一阶逻辑能表达很多种类的知识。然而，众所周知，它有一个缺点，即证明过程的不确定性。既然，我们要使用一个定理来证明它能有效地反复应用于实例，则必须将其限制在 Horn 子句的更简单情况中。

将定理转化为一组子句，包括称为 Skolem 化操作的一个步骤，就会出现这种情况，因为在子句中所有变量都需要是全称量词。因此，必须删去所有的存在量词的变量。

如果存在量词不在全称量词的作用范围内，就像仅有存在量词变量时，那么就必须由一个新的常量来替换；如果在一个或几个全称量词的作用范围之内，那么这个存在量化变量必须由一个新的函数来替代，这个函数依赖于那些∀作用于∃之前的全称量词变量。换句话，假定我们有一形如$\forall x\forall y\exists u\forall z\exists v F(x,y,z,u,v)$的公式，那么它能转化为$\forall x\forall y\forall z F(x,y,z,h(x,y),g(x,y,z))$，这里 h 和 g 是新的函数符号。注意 u 是由 x 和 y 的函数替代的，因为它在 x 和 y 的作用范围之内；而 v 是在变量 x，y 和 z 的作用范围之内，所以由 x，y 和 z 的函数来替代。所引入的常量和函数是新的符号。Skolem 化操作不能将公式转化为相应的形式。现在考虑当这样的操作由专家来完成，像人工智能一样，将不引入对其毫无意义的新符号，而是，坚持寻找一个能保持其子句真值的“好”符号。换句话，由人来完成这一操作就是一种伪 Skolem 化操作（因此在谈到 Skolem 化操作时上面用了引号），在这里伪 Skolem 函数是关于存在的知识的。

让我们给 Skolem 化操作问题举个例子来加以说明。对于“每个人都犯错误”这句话的子句表示，在一阶逻辑中，它为：

$$\forall x\exists y[\text{human}(x)\Rightarrow(\text{does}(x,y)\wedge\text{mistake}(y))]$$

变量 y 是在全称量词变量 x 的∀作用范围内，因此 y 不得不 Skolem 化操作为一个 x 的函数。我们能产生任意的 Skolem 函数，并转化为定理：

$$\forall x[\text{human}(x)\Rightarrow(\text{does}(x,g(x))\wedge\text{mistake}(g(x)))]$$

这里 g 是一个新符号函数。实际上，它很难忽略任何有关于函数 g 的信息，它告诉我们，什么是一种不合适的行为。一个专家甚至坚持要明确这些不适合被考虑的专业技术领域的行为。假定我们在一个学校班级的背景下，老师想要坚持认为缺乏注意力是会犯错误的，那么他将这种错误归纳为缺乏注意力，用 unthoughtful_action(x)来替代 $g(x)$，表示为：

$$\forall x[\text{human}(x)\Rightarrow(\text{does}(x,\text{unthoughtful\_action}(x))\wedge\text{mistake}(\text{unthoughtful\_action}(x)))]$$

为了完成这个例子，我们必须指出，专家也不喜欢在其谓词中有函数存在，他宁愿给出以下定理形式：

$\forall x \forall y[(\text{human}(x) \text{Aunthoughtful\_action\_pred}(x, y)) \Rightarrow (\text{does}(x, y) \wedge \text{mistake}(y))]$

这里，unthoughtful_action_pred(x,y)表示函数 unthoughtful_action(x)的图解，它能由等价的逻辑来表达：

$$\forall x \forall y[\text{unthoughtful\_action\_pred}(x, y) \Leftrightarrow y = \text{unthoughtful\_action}(x)]$$

上面的形式与我们最初的通用定理不再完全等价。理论上，这很令人讨厌。而在应用中，我们想让专家满意于他已转达给系统的知识。Skolem 化操作将迫使他去明晰自己的思想，这是知识获取中非常积极的特点。

据我们所知，对于 Skolem 函数，至今无人能解决选择适当位置的自动求解方法。然而，下面的例子清楚地表明，在把表达某一特征的句子转化为定理，进而转化为子句（或规则）的时候，只有在领域专家帮助下，才能做好伪 Skolem 化操作。为获取知识以建立专家系统的技术必须能够让专家成功地指出正确的补充信息，以便通过这些信息使我们解决 Skolem 化操作问题。

在下节中，我们将解释 OGUST 是如何工作的，其中将解释所选择的表示方法。

15.2.2　OGUST 介绍

OGUST 系统是一种归纳工具，它能从一组例子中学习某一概念的识别函数。它是基于一种结构匹配（Kodratoff 和 Ganascia，1986）的简化版本，并是基于知识的。

15.2.2.1　表示语言

例子和归纳的表示语言局限于一阶逻辑的一个子集，其中仅有变量、谓词符号和常量，后两者可视为无参数的函数和不允许参数大于或等于 1 的其他函数。因此一个原子公式可表示为：$p(t_1,\cdots,t_n)$，这里 p 是 n 元谓词，$t_1,\cdots,t_n$ 是常量或变量。

例子可表示为原子公式的合取，也就是没有变量。例如，假定一家汽车保险公司想要学习一个“costly driver”概念，它对保险公司而言是倾向于昂贵的。让我们处理下面的两个例子。

E_1：John 和 Mary 结婚，他们有驾驶执照；John 有车，Mary 仅在假日使用。

E_2：Peter 有一个儿子叫 Bob。Bob 刚获得驾驶执照；有时他借他父亲的车。

有很多方法将两句转化为谓词逻辑，一方面依赖于我们想要表达的一些内在信息，另一方面依赖谓词符号和常量的选择。我们这里将不讨论这一问题，因为在第 15.3 节已说明了在 ATC 应用中是如何做这种选择的。根据这些选择，E_1 和 E_2 按照以下方式进行翻译，这里 RE_1 和 RE_2 分别表示 E_1 和 E_2。

$$RE_1 = \text{married}(JOHN, MARY) \wedge \text{driving\_license}(JOHN) \wedge \text{driving\_license}(MARY) \wedge \text{owns}(JOHN, C1) \wedge \text{acar}(C1) \wedge \text{uses\_on\_holiday}(MARY, C1)$$

$$RE_2 = \text{son}(BOB, PETER) \wedge \text{new\_driver}(BOB) \wedge \text{owns}(PETER, C2) \wedge \text{acar}(C2) \wedge \text{borrows}(BOB, C2, PETER)$$

在这些例子中，married，driving_license，owns，car，uses_on_holiday，son，new_driver，borrows 是谓词，而 JOHN，MARY，BOB，PETER，C1，C2 是常量。注意，我们所表示的仅是句子中所明确表达的知识。我们也能表示 JOHN，PETER 和 BOB 是男人，而 MARY 是女人，因为这些信息是隐含在句子中的。由于函数是不允许的，所以例子的一些属性不能直接表达。已经有人提出了将有函数子句转换为无函数子句的技术（Rouveirol，1991）。

背景知识表达为 Horn 子句。例如，

$$R_1: \forall x \quad \text{man}(x) \Rightarrow \text{person}(x)$$

$$R_2: \forall x \quad \text{woman}(x) \Rightarrow \text{person}(x)$$

$$R_3: \forall x \forall y \quad \text{married}(x, y) \Rightarrow \text{family\_relation}(x, y)$$

$$R_4: \forall x \forall y \quad \text{son}(x, y) \Rightarrow \text{family\_relation}(x, y)$$

$$R_5: \forall x \forall y \quad \text{family\_relation}(x, y) \Rightarrow \text{family\_relation}(y, x)$$

$$R_6: \forall x \quad \text{new\_driver}(x) \Rightarrow \text{driving\_license}(x)$$

$$R_7: \forall x \forall y \quad \text{uses\_on\_holidays}(x, y) \wedge \text{acar}(y) \Rightarrow \text{driver}(x, y)$$

$$R_8: \forall x \forall y \quad \text{owns}(x, y) \wedge \text{acar}(y) \Rightarrow \text{driving\_license}(x)$$

$$R_9: \forall x \forall y \forall z \quad \text{borrows}(x, y, z) \wedge \text{acar}(y) \Rightarrow \text{driver}(x, y)$$

$$R_{10}: \forall x \forall y \quad \text{uses\_on\_holidays}(x, y) \wedge \text{acar}(y) \Rightarrow \text{unskilled\_driver}(x)$$

$$R_{11}: \forall x \quad \text{new\_driver}(x) \Rightarrow \text{unskilled\_driver}(x)$$

$$R_{12}: etc.$$

15.2.2.2 算法

归纳算法能分解为两个主要步骤：先是演绎算法，之后是归纳算法（见算法 15.1）。

在第一步，对这些例子进行结构匹配（Kodratoff 和 Ganascia,1986），也就是，利用背景知识转换它们的表示，直到它们已经匹配或进一步转换无法改善匹配为止。第二步则归纳我们通过结构匹配所获得的两种表示。

为了理解第一步，我们设想例子是一组对象，每个对象由自身属性和与其他对象的关系来描述。例如，例子 E_1 由三个对象 JOHN、MARY 和 C1 组成。JOHN 的特征为：他有驾驶执照，与 MARY 和 C1 存在某些关系。为对这些例子进行结构匹配，首先必须在这些对象中寻找在某种意义下是非常相似的对象，也就是说，最有希望进行归纳的对象。然后我们必须明确这些对象间的普通属性。

换句话，结构匹配能够分解为两步（算法 15.2）。

—在每个例子中选择一个对象，用一个归纳变量替代其所有出现的地方；

—对于那些不匹配的属性，使用给定的背景知识尽可能改善它们的匹配。

这两步重复执行，直到在例子中不再有对象可选择。一旦循环结束，就可以进行演绎归纳步骤了。

为说明这些概念，再回过头来说明“costly driver”的概念和实例 E_1，E_2。

输入

- E_1，…，E_n：概念 C 的样本，表示为基本原子合取；
- BK：由 Horn 子句表示的背景知识。

输出

1. 对样本进行结构匹配（演绎步骤）：将样本 E_1，…，E_n 转换为 E_1'，…，E_n'，以便对于 i = 1…n，

- $E_i \leftrightarrow E_i'$，
- $E_i' = F(x_1,...,x_p) \wedge G_i(x_1,...,x_p) \wedge (x_1 = C_i^1) \wedge ... \wedge (x_p = C_i^p)$，其中：

  F：一个公式，它包括变量 x_1,…，x_p，对所有样本均成立，

  G_i：原子的合取，每个原子至少在一个样本 E_j' 中出现一次，且 $j \neq i$ 时不出现，

  $C_i^1,...,C_i^p$：在 Ei 中出现的常数。

2. 泛化（归纳步骤）：形成对于所有样本均成立的公式 F

算法 15.1　归纳算法的简化版。在这种版本中，我们假定：

—在所有例子中，常量的数量是 p；

—我们不使用“重命名规则”，$P(x) \leftrightarrow (P(y) \wedge (x = y))$

定义：在一个样本 E_i 中出现一次的变量 x 是可区别的，若它出现在 E_i 一个谓词 $P(...,x,...)$的 p_k 位置，同时存在一个样本 E_j，$j \neq i$，其中 x 在谓词组成的一个原子 p_k 位置不出现。

输入

- E_1，…，E_n：概念 C 的样本，表示为基本原子合取；
- BK：由 Horn 子句表示的背景知识。

输出

对样本进行结构匹配（演绎步骤）：将样本 E_1，…，E_n 转换为 E_1'，…，E_n'，以便对于 i = 1,…,n，

- $E_i \leftrightarrow E_i'$，
- $E_i' = F(x_1,...,x_p) \wedge G_i(x_1,...,x_p) \wedge (x_1 = C_i^1) \wedge ... \wedge (x_p = C_i^p)$，其中：

  F：一个公式，它包括变量 $x_1,...,x_p$，对所有样本均成立，

  G_i：原子的合取，每个原子至少在一个样本 E_j' 中出现一次，且 $j \neq i$ 时不出现，

  $C_i^1,...,C_i^p$：在 E_i 中出现的常数。

  初始时，VAR=ϕ

（1）选择每个样本 E_i 中一个常量 C_i^j，用一个变量 x 替换所有的 C_i^j，其中：$x \notin VAR$：$E_i \leftrightarrow E_i' \wedge (x = C_i^j)$；$VAR \leftarrow VAR \cup \{x\}$.

（2）比较样本中出现的 x：

- 发现出现所有区分的变量 x；
- 对于一个样本 E_i 中出现的每个可区分谓词 $P(...,x,...)$，
- 如果可能，利用 BK，在没有出现谓词 $P(...,x,...)$的样本中引入谓词 $P(...,x,...)$。
- 如果不可能，尝试在样本 E_i 中利用 BK 从谓词 $P(...,x,...)$演绎一个新出现的 x，其不是可区分的。

（3）重复步骤（1）和（2），直到样本中没有常量为止。

算法 15.2 结构匹配算法的简化版

15.2.2.3 例子

结构匹配（演绎步骤）

在 E_1 中，有三个常量 JOHN，MARY 和 C1。在 E_2 中，也有三个常量 BOB，PETER 和 C2。为了确定哪些对象是最相似的，OGUST 使用一个启发式算法，就是比较对象的确切属性并计算这些对象间的相似性。算法 15.3 描述一个可能的过程，即根据相似程度 *sim* 的给定值来选择最好的常量去匹配。此算法根据 *sim* 给出最好的选择，但如果在各例子中常量数量很大，则很难处理，为此不得不比较 JOHN 和 PETER，JOHN 和 BOB，JOHN 和 C2，MARY 和 PETER，MARY 和 BOB 等。为了减少需要比较的次数，例子的常量可以进行类型划分。例如，这里我们能说比较一个人和一个车是没有意义的，这就可以将比较的数目从 9 个减少到 4 个。划分常量的类型是在归纳过程中引入知识的另一种方法；它通过放弃归纳“没有意义”的过度归纳来约束可能归纳的空间。为减少复杂性，已经研究出来其他启发知识以及这种算法的增量版本（Vrain 和 Lu，1988）。

输入

- E_1，…，E_n；样本，
- *sim* 一个 n 元函数；$sim(C_1,...,C_n)$，它计算常量 $C_1,...,C_n$ 之间的相似度。

输出：常量 $C_1,....C_n$ 的组合，其中 C_i 是 E_i 一个常量，i = 1,...,n，使得 *sim* 最大。

（1）对于每个常量 $C_1,...,C_n$ 的组合，其中 C_i，i = 1,...,n，是 E_i 一个常量，计算 $sim(C_1,...,C_n)$。

（2）选择一个最大的组合。

算法 15.3　常量选择

如果假设常量已划分类型，我们仅比较 JOHN 和 PETER，JOHN 和 BOB，MARY 和 PETER，MARY 和 JOHN。我们将不详细说明相似性的计算，只简单地给出构造这种方法的直觉。直观上我们能说：JOHN 和 PETER 似乎是最相似的人，因为他们都有车。因此，我们用同一归纳变量，这里称 vg_1，来替代所有 JOHN 和 PETER 的出现地方，由此得到：

$$RE_1 = \mathbf{married}(vg_1, MARY) \wedge \mathbf{driving\_license}(vg_1) \wedge \text{driving\_license}(MARY) \wedge \text{owns}(vg_1, C1) \wedge \text{car}(C1) \wedge \text{uses\_on\_holiday}(MARY, C1) \wedge [(= vg_1\, JOHN)]$$

$$RE_2 = \mathbf{son}(BOB, vg_1) \wedge \text{new\_driver}(BOB) \wedge \text{owns}(vg_1, C2) \wedge \text{car}(C2) \wedge \mathbf{borrows}(BOB, C2, vg_1) \wedge [(= vg_1\, PETER)]$$

在每个例子中，vg_1 满足拥有一个对象所强调的属性。因为这些在两个例子中都出现了，它们被认为是没有差别的。相反，在 E_1 中，vg_1 有驾驶执照且结婚，而在 E_2 中，vg_1 有儿子且有人从他（她）那里借东西。如果那样的话，由于 vg_1 在不属于两个例子原子中出现的事件，因此它们被称为变量 vg_1 的差异发生事件。它们在 RE_1 和 RE_2 中用粗体标出。OGUST 的工作就是利用可用的背景知识来尝试消除这些差异。这里对实现此任务的算法就不做详细说明了，它在 Vrain（1990）中有详细描述。我们这里只给出这种方法工作的直观陈述。

- 在原子 driving_license(vg_1) 中出现 vg_1。我们利用定理 R_8 能证明在 E_2 中，vg_1 有驾驶执照。因此，它的出现在“初始未加工”例子中是有区别的，但通过利用 BK 证明它没有区别。在下一步中它将作为无差异事件来处理。
- 在原子 married(vg_1,MARY) 中出现 vg_1。我们不能证明，在 E_2 中，vg_1 结婚，但我们能证明，在 E_1 和 E_2 中，vg_1 满足一种家庭关系。因此，差异事件被保留，但我们基于背景知识引入新的原子，这是在未加工例子中的信息保留以及减少 E_1 和 E_2 之间差异的结果。
- 在原子 son(BOB,vg_1) 中出现 vg_1，我们不能证明，在 E_1 中，vg_1 有一个儿子，我们也仅能证明，在 E_1 和 E_2 中，它满足一种家庭关系。这不会减少在两个例子之间的任何差异。
- 在原子 borrows(BOB,C2,vg_1) 中出现 vg_1。我们不能证明在 E_1 中的这一属性，但能证明每个例子中 vg_1 不满足任何新属性。因此，这些事件仍有差异，且通过利用背景知识无法获得任何新东西。

我们现在用简单的斜体来标记剩下的差异事件，因而表示为：

RE_1=*married*(vg_1,MARY) ∧ driving_license(vg_1) ∧ driving_license(MARY) ∧ owns(vg_1,C1) ∧ car(C1) ∧ uses_on_holiday(MARY,C1) ∧ family_relation(vg_1,MARY) ∧ [(=vg_1 JOHN)]

RE_2=*son*(BOB,vg_1) ∧ new_driver(BOB) ∧ owns(vg_1,C2) ∧ car(C2) ∧ *borrows*(BOB,C2,vg_1) ∧ driving_license(vg_1) ∧ family_relation(vg_1,BOB) ∧ [(=vg_1 PETER)]

重复同样过程。在每个例子中都出现了一个人和一辆车。由于我们已经假定，不能将人和车做比较，所以我们只能选择 BOB 和 MARY 或 C1 和 C2 做比较。这里我们选择 BOB 和 MARY 来做比较，用一个新的归纳变量 vg_2 来代替他们

的出现之处。我们用粗体字来标明这个新的差异事件，而用下划线来标明没有差异的新事件。结果如下：

RE_1=**married**(vg_1,vg_2) Λ driving_license(vg_1) Λ **driving_license**(vg_2) Λ owns(vg_1,C1) Λ car(C1) Λ **uses_on_holiday**(vg_2,C1) Λ family_relation(vg_1,vg_2) Λ [(=vg_1 JOHN)(=vg_2 MARY)]

RE_2=**son**(vg_2,vg_1) Λ **new_driver**(vg_2) Λ owns(vg_1,C2) Λ car(C2) Λ **borrows**(vg_2,C2,vg_1) Λ driving_license(vg_1) Λ family_relation(vg_1,vg_2) Λ [(=vg_1 PETER)(=vg_2 BOB)]

再利用背景知识来尽可能消除差异事件，就能获得如下结果。

RE_1=married(vg_1,vg_2) Λ driving_license(vg_1) Λ driving_license(vg_2) Λ owns(vg_1,C1) Λ car(C1) Λ uses_on_holiday(vg_2,C1) Λ family_relation(vg_1,vg_2) Λ driver(vg_2,C1) Λ unskilled_driver(vg_2) Λ [(=vg_1 JOHN)(=vg_2 MARY)]

RE_2=son(vg_2,vg_1) Λ new_driver(vg_2) Λ owns(vg_1,C2) Λ car(C2) Λ borrows (vg_2,C2,vg_1) Λ driving_license(vg_1) Λ family_relation(vg_1,vg_2) Λ driver(vg_2,C2) Λ unskilled_driver(vg_2) Λ [(=vg_1 PETER)(=vg_2 BOB)]

现在，每个例子中都只剩下一个常量。我们用变量 vg_3 来替代它。

RE_1=married(vg_1,vg_2) Λ driving_license(vg_1) Λ driving_license(vg_2) Λ owns (vg_1,vg_3) Λ car(vg_3) Λ **uses_on_holiday**(vg_2,vg_3) Λ family_relation(vg_1,vg_2) Λ driver(vg_2,vg_3) Λ unskilled_driver(vg_2) Λ [(=vg_1 JOHN)(=vg_2 MARY)(=vg_3 C1)]

RE_2=son(vg_2,vg_1) Λ new_driver(vg_2) Λ owns(vg_1,vg_3) Λ car(vg_3) Λ **borrows** (vg_2, vg_3,vg_1) Λ driving_license (vg_1) Λ family_relation(vg_1,vg_2) Λ driver(vg_2, vg_3) Λ unskilled_driver(vg_2) Λ [(=vg_1 PETER)(=vg_2 BOB)(=vg_3 C2)]

我们再次试图利用领域知识处理这些差异事件。这样的话，我们就学习不到任何新的共同属性，因此：

RE_1=married(vg_1,vg_2) Λ driving_license(vg_1) Λ driving_license(vg_2) Λ owns (vg_1,vg_3) Λ car(vg3) Λ uses_on_holiday(vg_2,vg_3) Λ family_relation(vg_1,vg_2) Λ driver(vg_2,vg_3) Λ unskilled_driver(vg_2) Λ [(=vg_1 JOHN)(=vg_2 MARY)(=vg_3 C1)]

RE_2=son(vg_2,vg_1) Λ new_driver(vg_2) Λ owns(vg_1,vg_3) Λ car(vg_3) Λ borrows (vg_2, vg_3,vg_1) Λ driving_license(vg_1) Λ family_relation(vg_1,vg_2) Λ driver(vg_2, vg_3) Λ unskilled_driver(vg_2) Λ [(=vg_1 PETER)(=vg_2 BOB)(=vg_3 C2)]

例子中不再有常量存在，而且已尽可能地利用背景知识来消除差异属性。

因此，结构匹配的步骤就完成了。

归纳步骤

我们采用直接方式，通过保留其共同属性，归纳了两个例子。这产生了识别函数 G：

$G=owns(vg_1,vg_3)\wedge car(vg_3)\wedge driving\_license(vg_1)\wedge family\_relation\ (vg_1,vg_2)\wedge driver(vg_2,vg_3)\wedge unskilled\_driver(vg_2)$

15.2.2.4 结论

OGUST 的优点

- 它的表示语言的谓词逻辑比属性-值表示要丰富得多，因为它能有效地表示若干对象以及它们之间的关系；
- 系统利用背景知识来发现一个尽可能具体的概念的识别函数；
- 有能力对其结果生成解释。

OGUST 的缺点

- 指数复杂度。为发现例子的一个归纳，它研究要匹配对象所有可能的选择；
- 不能处理反例。仔细评估了利用启发式，根据反例来选择常量进行匹配，但它们不能使我们处理它们的全部内容，而且学习的识别函数可能不一致，也就是说，它可能将正例和反例混淆起来。

在 ATC 应用中，选择 OGUST 是因为如下原因。我们已阐述对于 ATC 而言，一阶知识的重要性。而且，在 ATC 领域中，存在背景知识。它不是很明确，作为工程任务的一部分，就是从专家那里获取这些知识。最后，我们需要对系统行为的解释，至少是因为以下两个原因：

- 必须验证学习获得的规则；
- 从专家获得的背景知识可能包含错误，且必须对它进行调试。

在这一领域的一个反例可能是，由于 ATC 的决定，两个飞行器发生碰撞，且这些反例对我们来说是无法获得的。由于 OGUST 不能处理反例，所以它的局限性在这个应用中并不是一个大的问题。

15.3　ATC 的应用

15.3.1　简介

在第 15.1 节中提出的知识获取方案应用于一个实际领域，并不像这个模式所说明的那样直接。在项目开始时,我们认为主要问题将是一阶知识的使用。实际上,我们的归纳工具能很好地适应 ATC 知识。主要问题来自背景知识的抽取,甚至令人吃惊的是，来自谓词的定义，也就是说，出人意料，专家倾向于提供基于自己词汇的错误定义。

我们将描述不得不面对的问题,且指出所采用的解决方法。对每个问题，我们将区分它是存在于学习专家规则的过程中，还是我们实际使用的特别归纳工具中。

完全获取的循环过程一定包含一些必需的步骤来转化知识，以便使专家和系统之间能够相互理解。由专家给定的知识（即例子+背景知识）必须翻译为适应归纳工具的表示。为了使验证步骤更容易，学习获得的知识必须翻译回能被专家理解的表示形式。在下面章节中，我们描述整个知识获取任务。

15.3.2　选择表示语言

这一问题能分解为两个相互联系的子问题：

1. 选择基本词汇；
2. 选择公式、逻辑以及命题的、谓词的或属性-值的表示等。

表示问题是人工智能中的一个基本问题。在我们应用中，由领域和归纳工具的选择来决定公式：它是一阶逻辑的子集。OGUST 与许多符号学习系统一样，不能处理数字量，因此需要预先转化为符号量。除此之外，由于 OGUST 的实际行为，我们在确定表示语言中的常量和谓词时需十分小心。上述这一点在第 15.3.4.2 节中有详细说明。

15.3.3　确定要学习的概念

此应用的目的就是要根据人类控制员在解决冲突时的动作例子来学习决策规则。我们定义了 6 个动作，这些动作的完成将有助于避免飞行器相互逼近。

- 不处理：对其他的飞行器执行这一动作

- 改变方向，这是对飞行器飞行路线的暂时修改
- 改变飞行器飞行路线
- 将飞行器稳定在一个给定高度
- 速度调整
- 改变高度调整，即改变垂直速度

两个飞行器之间冲突的解决方法就是对每个飞行器执行这些动作的组合。

对每次解决的方法，我们想要学习可再次明智地使用这种解决方法的情况描述。为了这一目的，我们不得不使用 OGUST 对需要相同解决方法的例子进行聚类和归纳。使用 OGUST 所获得的识别函数可能会太一般，甚至可能为空。这种情况可能会发生，因为 OGUST 学习一个概念的纯合取识别函数和最初定义概念，实际上是更多具体概念的析取。例如，在与专家做进一步讨论之后，**稳定性**的解决方法被分解为两类解决方案。它可能是飞行器对在所考虑区域飞行期间所要保持的**高度稳定性**，或为避免冲突的**暂时稳定性**，除此之外，任何稳定性都能直接地或通过一些中间层次以这两种方式来实现。

因此，一个重要任务就是定义要学习的中间概念。

15.3.4 获取例子并重写它们

语言表示和要学习的概念一旦确定，问题就是要获取这些概念的例子和重新按照期望形式改写它们。在大部分应用中，制定了一个调查表，专家在真实情况下填写这张表。此调查表可视为一组可能取值的属性列表，属性是由基本词汇确定的。对一个给定情况，专家核查每个属性的值是否正确，并说明它是概念的正例，还是反例，然后将例子翻译为正确的形式。

15.3.4.1 获取“未加工”例子

在我们的应用中，获取可用的例子前，还有问题要解决。建立一份调查表，但经验表示，在实际情况下，它是不可能被真实地填写，因为在冲突事件发生和解决之间的时间是很短的，专家希望用以下两个步骤来做这个工作。

第一步，他画一张表示冲突情况的蓝图，以简便的口头信息作为图中表示的补充。图 15.2 和图 15.3 描绘了专家在控制任务期间所绘的冲突情况。我们能看到增添到方案中的属性值就有了数值，例如在冲突点前，飞行器的收敛角度。

当前工作负载：高、稳定、低
预期工作负载：高、稳定、低
在冲突发生前飞行器路线的角度估计：10 度

	AF2470	BR1667
冲突发生后飞行器路线的角度估计：	0 度	0 度
飞行器扔对位置		
等距离		
稍微超前		×
超前		
不确定		
上升速率		
高		
标准	×	
低		
相对地面速率，相对空气速率	470/0.79	410/0.64
退出区域时垂直平面定性轨道：		
稳定		×
变化小	↑	
区域中垂直平面定性轨道：		
稳定		×
变化小	↑	
退出区域时垂直平面定性轨道：		
稳定	×	×
变化小		
目的地：		
接近		×
西欧	×	
很远		
环境限制：		
活动区域		
其他飞行		
区域附近		×
描述环境限制条件的参数：		
改变方向，向左，值，距离或灯塔		×
改变方向，向右，值，距离或灯塔	×>10 度	
改变方向，向左，值，距离		
改变方向，向右，值，距离		
不确定性：		
雷达联系差		
雷达图像可读性差		
不稳定信息		
仆解决距离冲突点的飞行距离方案实现时	40NM	40NM

解决方案：
永久的稳定态
暂时的稳定态
改变方向
路线并行
离开冲突点
离冲突点更近
仆后面交叉
在前面交叉
路线改变
速度规则
高度规则
比驾驶员估计偏左

AF2470，向右 10°
BR1667，方向不变

图 15.2　有关两个飞行器的冲突信息

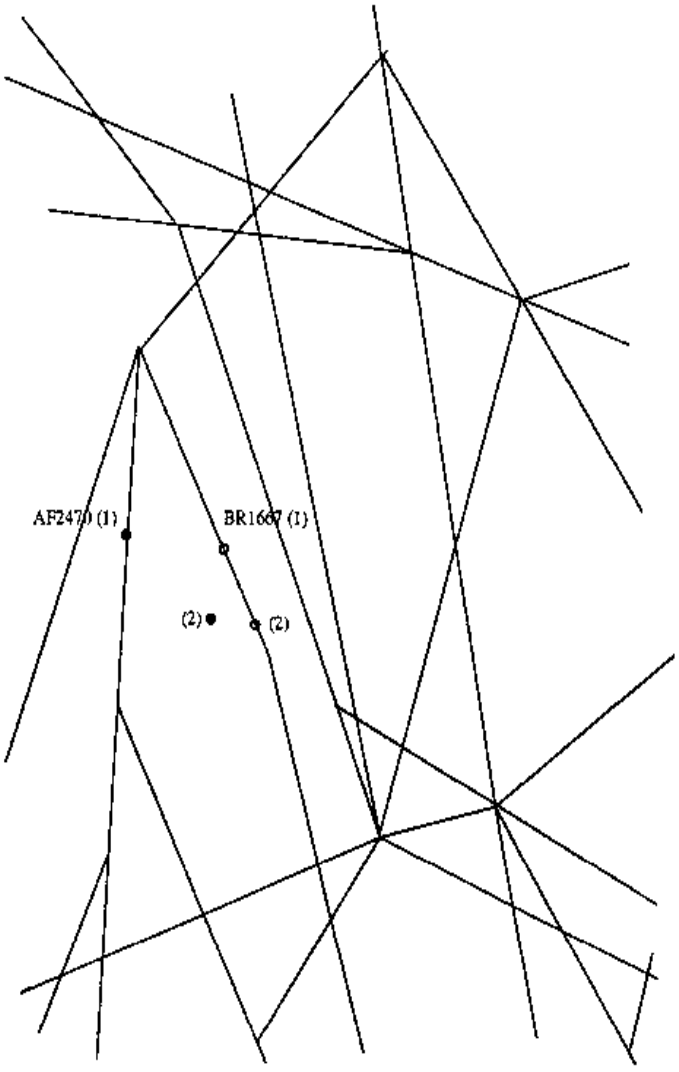


图 15.3　表明冲突情况的蓝图

第二步，他将蓝图和注释转换为对调查表的回答（如图 15.4 所示）。在调查表中，所有的数值已翻译为符号值。例如，在冲突点，飞行器的相遇角度，被限制的值：相同路线和分歧路线。

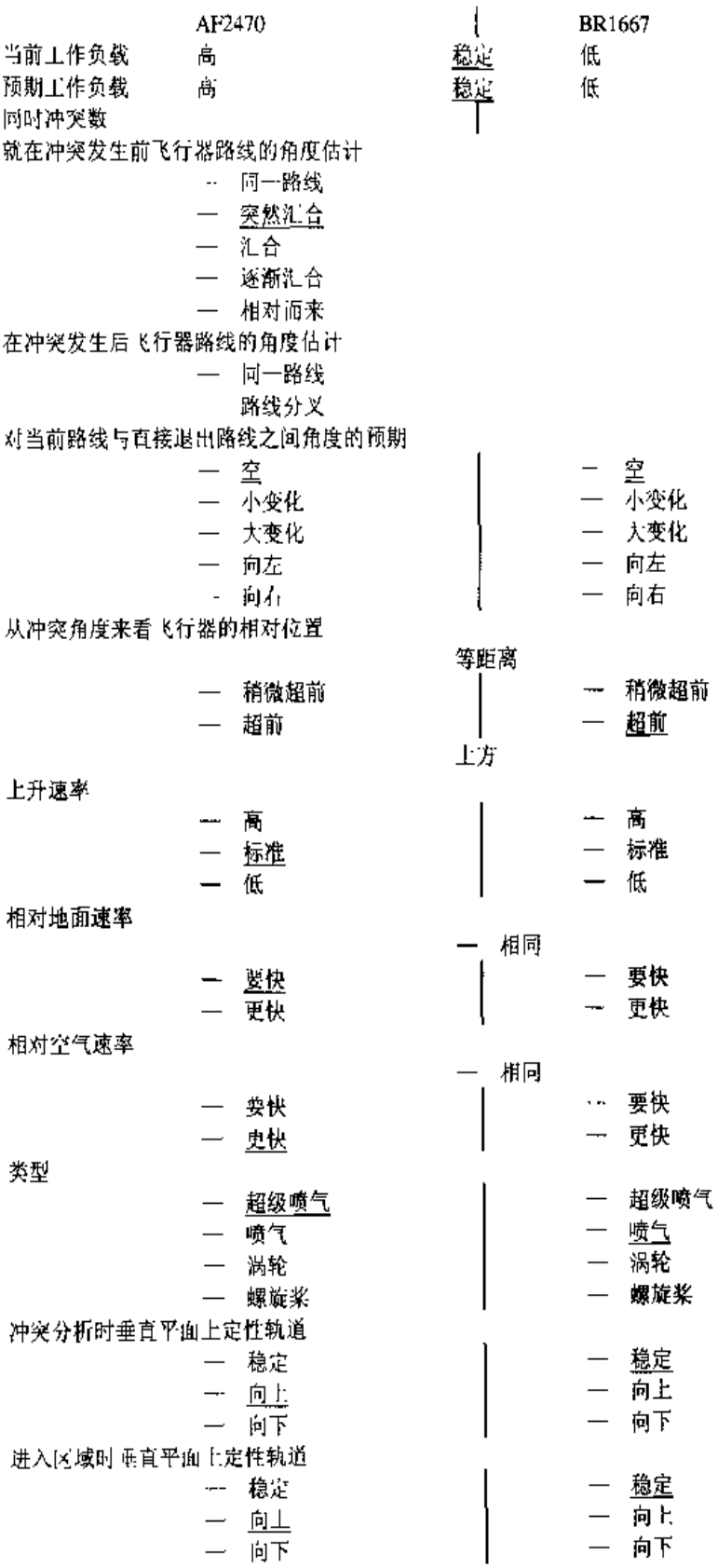


图 15.4　将图 15.3 所示的冲突情况转化成的调查表（上半部分）

区域中垂直平面上定性轨道

— 稳定	— 稳定
— 向上	— 向上
— 向下	— 向下

退出区域时垂直平面上定性轨道

— 稳定	稳定
— 向上	— 向上
向下	向下

目的

— 接近	— 接近
— 西欧	— 西欧
很远	— 很远

禁止操作

— 活动区域	活动区域
— 其他飞行	其他飞行
— 容许稍微变化	容许稍微变化
— 容许细微变化	容许细微变化
— 向左	向左
— 向右	向右

除非特别容许，否则禁止的操作

— 区域附近	— 区域附近
— 其他飞行	— 其他飞行
— 容许稍微变化	— 容许稍微变化
— 容许细微变化	— 容许细微变化
— 向左	— 向左
— 向右	— 向右

不确定

— 雷达联系差	— 雷达联系差
— 雷达图像可读性差	— 雷达图像可读性差
— 不稳定信息	— 不稳定信息

在解决方案实现时距离冲突点的飞行距离

— 很远
— 较近
- 附近

解决方案

— 永久的稳定态	— 永久的稳定态
— 暂时的稳定态	— 暂时的稳定态
— 改变方向	— 改变方向
— 改变路线	— 改变路线
— 稍微	— 稍微
— 较小	— 较小
— 向左	— 向左
— 向右	— 向右
— 路线并行	— 路线并行
— 离开冲突点	— 离开冲突点
— 离冲突点更近	— 离冲突点更近
— 在后面交叉	— 在后面交叉
— 在前面交叉	— 在前面交叉
- 调整速度	— 调整速度
— 调整高度	— 调整高度
— 比驾驶员估计偏左	— 比驾驶员估计偏左

图 15.4　将图 15.3 所示的冲突情况转化成的调查表（下半部分）

调查表由不同种类的属性组成。

- 描述控制器的整体工作环境属性，如当前工作量和预期工作量等。每一个这样的属性都有三种可能的值：低、稳态、高。它们将影响到控制人员的决策。例如，当当前工作量属性是高时，控制人员将更加谨慎，并常常在实际冲突情况发生前调整飞行器的路线。
- 描述每个飞行器特点的属性。例如类型属性的取值，像超音速喷气式飞机、喷气式飞机、涡轮飞机或直升飞机；在区域内部的垂直方向上的喷射属性有三个可能的值：稳态、上、下。控制人员给每个飞行器属性取值。这里一个特别的属性就是应用解决飞行器冲突的方法。这个属性是惟一的多值属性，它描述飞行器路线的改变方式。
- 描述飞行器之间关系的属性,它分为两种。

—描述两个冲突的飞行器方位关系。例如，两个飞行器之间相关位置的可能取值为 *ahead*，*slightly ahead*，*equidistant from conflict occurrence*，*just above each other*。

—描述飞行器间非方位属性。例如，飞行器冲突前预期的轨道角度，它们冲突发生后的角度等。

第一个模块将打印输出的注释转换为相似调查表的一种形式，第二步就是要将调查表转换为适合 OGUST 的形式。

15.3.4.2　用 OGUST 形式重写例子

如何将调查表转换为一阶表示

在大部分应用中，专家用属性和对象间关系来描述他不得不面对的情况。属性表示情况发生的上下文的全局描述符，例如天气，或在情况出现时对象的描述符，例如病人的体温，一个对象的颜色等。其他方面，由专家提供的描述符组成如下：

- 描述环境的一些二元组(属性，值)，例如(天气,好)；
- 一些三元组(属性，对象，值)，例如(温度，BOB，高)；
- 能定位的一些关系(关系，对象 $_1$，对象 $_2$，……)。

在这种情况下，对象在列表中出现的是重要的。经常情况下，这些关系是两元的。

我们至少能看到有四种方式可把一个（属性，值）对转换为一阶谓的词形式，但列表并不是穷举的。首先让我们考虑全局描述符，由（属性，值）来表示，例如（天气，好）意味着在给定的情况下，天气是好的。

♦ 第一种表示形式

属性（值），这里属性成为一个谓词，其中的值就是常量。这种表示最接近属性值。在这种情况下单词“天气”就成为一个潜在一阶语言中的谓词，单词“好”就成为常量。在谓词逻辑中，为了给这个公式做一个合理的解释，我们给出一个非空集合 D，并且在集合 D 内解释常量“好”和谓词“天气”。从直观上，我们把“好”当做一个实体有些不妥。

♦ 第二种表示形式

值（属性），值变成了谓词，属性变成了常量。在这种表示形式中，我们把二元谓词（天气，好）转换成了一元谓词“好（天气）”，这里“天气”是常量，“好”是谓词。从直观上，“天气”被解释为满足“好”特性的一个实体。

♦ 第三种表示形式

谓词（属性，值），“谓词”代表一个新的谓词，属性、值均为常量。在这种表示形式中，（属性，值）就变成谓词参数，而“谓词”就成为一个新的谓词符号。例如，我们可以说（天气，好）代表着天气预报“预报（天气，好）”。

♦ 第四种表示形式

属性（C）∧值（C），属性和值均是谓词。C 是常量符号。这种表示形式与面向对象的形式很接近。举个例子，在我们这个例子中。我们用 W 来表示一个对象，W 满足下面的两个特性：W 代表确定情况下的天气，并且天气是好的。我们将其表示为 weather (W)∧fine (W)。这里 fine、weather 都是谓词，W 是新引进的一个常量。引进一个虚对象的思想是很典型的，它能把用一阶谓词逻辑描述的事实转换为结构化对象表示（Nilsson，1980）。例如，为了把原子谓词 gives（PETER，MARY，BOOK1）转化成第四种表示形式。我们引进一个实体 G 来表示行为“给”的一个实例，这里给予者是 PETER，接收者是 MARY，对象是 BOOK1，于是我们有 give (G)∧actor(G，PETER)∧receiver(G，MARY)∧Object(G，BOOK1)。

现在我们将这四种描述实体的形式扩展为三元组(attribute,object,value)，

例如三元组（color-eyes ,JOHN ,BROWN）。

♦ 第一种表示

attribute (OBJECT,VALUE) 产生 *color_eyes* (JOHN,BROWN)。

♦ 第二种表示

value (OBJECT,ATTRIBUTE) 产生 *brown* (JOHN,COLOR_EYES)。

♦ 第三种表示

Pred (OBJECT,ATTRIBUTE,VALUE) 产生 *physical_descr* (JOHN, COLOR_EYES, BROWN)。

♦ 第四种表示

attribute (OBJECT,C)∧*value*(C) 产生 *color-eyes* (JOHN,C) ∧*brown*(C)，这里 C 是一个新的常量。

实体和谓词的选择依赖于归纳工具的行为，且学习好的归纳是基本的。现在我们将说明为何采用第四种表示方式。

采用第一和第三种表示方法，当不同的属性具有相同值时，就会使得 OGUST 产生错误的识别函数

例如，我们假设根据第一种表示，用 present_work(HIGH)∧foreseen_work (HIGH)来表示这种情况，即当前控制人员工作负荷和预期工作负荷都高。

OGUST 无法区分当前工作负荷中的 HIGH 值和预期工作负荷中 HIGH 值之间的区别，它们在 OGUST 中被当做同一个对象并用同一个变量来进行归纳，从而获得 present_work(vg_1)∧foreseen_work(vg_1)。变量 vg_1 是当前工作负荷和预期工作负荷相结合所对应的一个实体。事实上这样一种表示是无意义的，因为它将两个不同对象合并成了惟一一个对象。

同样也能说明第三种表示方式的不合理性。例如我们重写上面的例子 work(PRESENT_WORK, HIGH) ∧ work (FORESEEN_WORK, HIGH)，将其中的 HIGH 常量变为 vg_1 变量，于是就有 work(PRESENT_WORK, vg_1) ∧ work(FORESEEN_WORK,vg_1)。

我们可以看到，当采用第二种和第四种方式时，这种问题就不存在了。例如我们用第二种谓词表示法把上面公式重写为 high(PRESENT_WORK) ∧ high(FORESEEN_WORK)，然后两个不同的常量 PRESENT_WORK 和 FORSEEN_WORK 抽象成两个不同的变量 high (vg_1) ∧ high(vg_2)(= vg1PRESENT_WORK)(= vg_2FORESEEN_WORK)]。

用第四种表示方式时，例子就变为：present_work(C1) ∧ high (C1) ∧ foreseen_work(C2) ∧high (C2)。

应用第二种和第三种表示方式将使 OGUST 不能正确使用背景知识，这将导致过度归纳

举个例子，假设在 E_1 中预期的工作负荷（foreseen_work）是高(high)的，在 E_2 中当前的工作负荷(present_work)也是高（high）的。用第二种表示方式表示如下：

$$E_1 = \text{high (FORESEEN\_WORK)}$$

$$E_2 = \text{high(PRESENT\_WORK)}$$

这里也假设 present _work 和 foreseen_work 能被抽象成属性 amount_work。OGUST 在谓词中使用背景知识，而不是在变量中。所以它把两个常量 PRESENT_WORK 和 FORESEEN_WORK 抽象成抽象变量 vg_1 并学习 $G=high(vg_1)$。这里没有用到以前的任何知识，并且也失去 vg_1 作为工作量的任何信息。第三种表示也是同样的情况。

在第一种和第四种表示方式中，这种问题不存在。例如让我们考虑用第四种方式重新表示上面的问题，例子可以表示为：

$$E_1 = \text{foreseen\_work(C1)} \wedge \text{high(C1)}$$

$$E_2 = \text{present\_work(C2)} \wedge \text{high(C2)},$$

这里每个例子中常量 HIGH 可以被一个抽象变量 vg_1 所替代，然后可写做：

$$E_1 = \text{foreseen\_work}(vg_1) \wedge \text{high}(vg_1)[(=vg_1\text{C1})]$$

$$E_2 = \text{present\_work}(vg_1) \wedge \text{high}(vg_1)[(=vg_1\text{C2})]$$

OGUST 利用其自身的背景知识来抑制变量 vg_1 在例子中的差别，于是我们获得以下抽象：

$$G=\text{amount\_work}(vg_1) \wedge \text{high}(vg_1)$$

能同时解决这两个问题的惟一表示方式就是第四种表示方式，这就是我们选择它的原因。

实际表示方式的描述

每一个飞行器用一个常量表示，下面为简单起见，AIRCRAFT_i 代表第 i 个飞行器。

对于描述环境的每一个属性，我们用常量 AT_1 来表示一个虚构的实体。它

被创建并用以下合取式描述：$attr(AT_1) \wedge val(AT_1)$

这里 *attr* 是描述环境的相关属性名，*val* 是它的值。

对于每一个描述第 i 架飞机的属性，可创建一个虚构对象，称为 $ATTR_i$，它们被创建后由下面的合取式描述。

$$attr(AIRCRAFT_i, ATTR_i) \wedge val(ATTR_i),$$

这里 *val* 代表例中第 i 个飞行器的 *attr* 属性值，换句话说，我们创建一个虚构对象以表示第 i 个飞行器的 *attr* 属性，它的值为 *val*。

举个例子，我们用以下公式来表示在扇形区域内，第 1 架飞机的飞行轨迹是平稳的：

$$traj\_in(AIRCRAFT_1, TRAJ\_IN_1) \wedge steady(TRAJ\_IN_1)$$

如果要解决两架飞行器的碰撞问题需要让第 1 架飞机的飞行路径改向右。我们可以写为：

$$solution(AIRCRAFT_1, SOLUTION_1) \wedge change\_course(SOLUTION_1) \wedge$$

$$to\_the\_right(SOLUTION_1).$$

假如第 2 架飞机的类型是喷气式的，又可写成：

$$type(AIRCRAFT_2, TYPE_2) \wedge jet(TYPE_2).$$

对于代表着两架飞机定位关系每一个相关的属性，可以直接写为一个谓词，因此

$$attr(AIRCRAFT_i, AIRCRAFT_j)$$

表示在这个例子中，关系属性 *attr* 满足第 i 个和第 j 个飞行器之间的关系。因此我们有：

$$ahead(AIRCRAFT_i, AIRCRAFT_j).$$

对于代表两架飞行器非定位关系的每一个属性，一阶谓词逻辑就不适合，因为它只适合定位关系。为了避免这个问题，在我们的应用中，它们被当做一个全局描述符，并且与前面描述环境的符号有相同的表示。因此，这里我们引进一个虚构的对象 $ATTR_1$，并写为：

$$attr(ATTR_1) \wedge val(ATTR_1)$$

这里 *attr* 就是与非定位关系有关的名称。

举个例子，在一个特定的例子中，为了表述两架飞行器在碰撞前的飞行轨迹有最小的交角，可表示成：

$$angle\_before(ANGLE\_BEFORE_1) \wedge minimal(ANGLE\_BEFORE_1).$$

下面我们把图 15.2，15.3 和 15.4 中例子转换为如下表示：

（∧（present_work_load PRESENT _WORK _ $LOAD_1$）（steady PRESENT _WORK _ $LOAD_1$）

(foreseen_work_load FORESEEN_WORK_$LOAD_1$)(steady FORESEEN _ WORK _ $LOAD_1$)

(angle_routes_before ANGLE_ROUTES_$BEFORE_1$)(sharp ANGLE_ ROUTES _ $BEFORE_1$)

(angle_routes_after ANGLE_ROUTES_$AFTER_1$)(same ANGLE_ROUTES_ $AFTER_1$)(direct_route AF2470 DIRECT _ $ROUTE_1$)（null DIRECT _ $ROUTE_1$）

(direct_route BR1667 DIRECT _ $ROUTE_2$)（null DIRECT _ $ROUTE_2$）

(rate_raise AF2470 RATE_$RAISE_1$)(null RATE_$RAISE_1$)(rate_raise BR1667 RATE _ $RAISE_2$)

(type AF2470 $TYPE_1$)(superjet $TYPE_1$)

(type BR1667 $TYPE_2$)(jet $TYPE_2$)

(traj_initAF2470 TRAJ _ $INIT_1$)(up TRAJ _ $INIT_1$)

(traj_init BR1667 TRAJ _ $INIT_2$)(steady TRAJ _ $INIT_2$)…

(destination AF2470 $DESTINATION_1$)（west_european $DESTINATION_1$）

(destination BR1667 $DESTINATION_2$) (far_away $DESTINATION_2$)

(solution AF2470 $SOLUTION_1$)(change_direction $SOLUTION_1$)

(parallelization_routes $SOLUTION_1$)

(right $SOLUTION_1$)

(slight $SOLUTION_1$)

(solution BR 1667 $SOLUTION_2$)),

这里谓词符号是括号内的第一个符号。

作为这个问题的总结，我们说所选择的表示是一个结构化对象，用一阶谓词形式表述。我们能处理像控制人员当前工作负荷这样的一些基本实体，我们也能处理像飞行器（由一些基本对象组成）这样结构化的实体。例如，它们的类型或者它们在区域内轮廓等，但所有对象的类型都是固定的，以致我们无法匹配，例如，当前工作负荷和预期工作负荷。算法 15.4 解释了这种表示的变化是如何实现的，在我们的应用中，这个算法较为简单，因为我们仅有两个结构对象，即碰撞中的两架飞行器。两架飞行器的非定位关系被当

做全局描述符处理，并写入到列表 LIST_ENV 中。

输入

• 一个列表LIST_ENV，包含描述全局环境的属性/值 (Attribute,Value) 对。

• 一个列表LIST_OBJ=($L1$,$\cdots Lp$)，其中每项Li表示一个结构化对象，具有形式（ $Name_i$， $Type_i(Atrribute_i^1,Value_i^1),...,(Atrribute_i^n,Value_i^n)$ 。

• 一个列表LIST_REL ，描述对象间存在的联系。

输出　表示样本实例的基础原子合取。

1.对于LIST_ENV中的每个原子(Attribute,Value)，

— 通过将一个数字加到属性上来创建一个新名字（例如1）

— 将(Attribute,Value)转换为Attribute (ATTRIBUTE1) ∧ value(ATTRIBUTE1)

2.对于列表LIST_OBJ 中每个表项$L1$，描述对象名称 $Name_i$ 和类型 $Type_i$，并对于该列表中每个元素 $(Atrribute_i^j,Value_i^j)$

— 创建原子 $Type_i(\text{NAME}_i)$

— 通过将一个数字加到属性上来创建一个新名字（例如1）

— 将 $(Atrribute_i^j,Value_i^j)$ 转换为

$Atrribute_i^j(\text{NAME}_i,\text{ATTRIBUTE}_{i1}^j)\wedge(value_i^j(\text{ATTRIBUTE}_{i1}^j)$

3.列表 LIST_REL 中的每个元素保持不变

算法 15.4　表示的变化

15.3.5　用 Horn 子句重写背景知识

15.3.5.1　例子表示的影响

例子表示方法的选择对与 BK 有较大的影响。例如，假定需要表示下面的知识：

如果给定一架飞机，在进入、飞出、存在于区域时有稳定轨迹，则表明它在航行中。

飞行器在进入区域和飞出区域时的定性轨迹可作为调查表中的属性。相反，飞行器巡逻时的飞行特性却是一个新问题。为了和我们的结构化表示保持一致，我们必须定义一个新的飞行属性来描述飞行，该属性取三个值：巡

航、巡航中移动、巡航中部分移动。我们称这个新属性为 flight（飞行），定理表示如下：

TH_1: $\forall x \forall y \forall z \forall t$ [[traj_enter(x,y) $\wedge$ steady(y) $\wedge$ traj_init(x,z) $\wedge$ steady(z) $\wedge$ traj_exit(x,t) $\wedge$ steady(t)] => $\exists fl$[flight(x,fl) $\wedge$ cruising(fl)]]

定理的结论表示：一个飞行对象 x 存在三种满足于巡逻特性的表示方式。

15.3.5.2 Horn 子句

定理必须表示为 Horn 子句形式：一个定理的结论必须是单原子形式，并且定理的结论中不允许有存在量词。为了解决第一个问题，我们可以用下面两项来替代前面的定理：

TH_{21} : $\forall x \forall y \forall z \forall t$[[traj_enter($x$,$y$) $\wedge$ steady(y) $\wedge$ traj_in(x,z) $\wedge$steady(z) $\wedge$ traj_exit(x,t)$\wedge$ steady(t)]$\Rightarrow$ $\exists$ fl[flight(x,fl)]]

TH_{22} : $\forall x \forall y \forall z \forall t \forall fl$[traj_enter($x$,$y$) $\wedge$ steady(y) $\wedge$ traj_in(x,z) $\wedge$ steady (z) $\wedge$ traj_exit(x,t) $\wedge$ steady(t) $\wedge$ flight(x,fl)] $\Rightarrow$cruising(fl)]

对于第一个定理，在某些条件下，我们可以引入一个新的对象——飞机的航班（flight）。对于第二个定理，在相同的条件下，如果我们用变量 fl 来代表飞机的航班，那么我们就可以推出这个航班 fl 是巡航。

TH_1 是 TH_{21} ^TH_{22} 的逻辑结果，但它的逆命题是非真的，因此这两个公式是不等价的。实际上，TH_{21} 并不能准确地表达我们的知识，因为我们知道，存在一个惟一的对象，它代表 x 的航班。我们应当使用量词记号$\exists!$。但我们不允许实际中出现这种形式。

为了解决 TH_{21} 式中存在量词的问题，可以用归一化方法处理。引入一个新的函数符号 f 来重写这个式子：

TH_{31} :$\forall x \forall y \forall z \forall t$[[traj_enter($x$,$y$) $\wedge$ steady(y) $\wedge$ traj_in(x,z) $\wedge$ steady(z) $\wedge$ traj_exit(x,t)$\wedge$ steady(t)]$\Rightarrow$ flight(x,f(x,y,z,t))]

15.3.5.3 引入一个函数项

将一个定理应用到一个例子上，可以像 TH_{31} 这样引入一个函数项。在这种情况下，我们将函数符号用一个新的常量来代替，最好是一个对于专家来说有意义的常量。这是我们在第 15.2.1.3 节中所学习的解决归一化问题的关键一步。在下面的例子中，TH_{31} 所引入的函数 f(AIRCRAFT,TRAJ_ENT, TRAJ_IN, TRAJ_OUT)将被一个新的常量 NEL 代替，遗憾的是，由于我们的专家对此不

能做任何说明，所以对于这个常量我们不知道任何特殊的信息。

例如，假定我们的例子是：

E=traj_enter(AIRCRAFT,TRAJ_ENT) ∧ steady(TRAJ_ENT) ∧

traj_in(AIRCRAFT,TRAJ_IN) ∧ steady(TRAJ_IN) ∧

traj_exit(AIRCRAFT,TRAJ_OUT) ∧ steady(TRAJ_OUT) ∧…

应用定理TH_{31}时，我们引入一个新的常量NEL，这样可得到：

E=traj_enter(AIRCRAFT,TRAJ_ENT) ∧steady(TRAJ_ENT) ∧

traj_in(AIRCRAFT,TRAJ_IN) ∧ steady(TRAJ_IN) ∧

traj_exit(AIRCRAFT,TRAJ_OUT) ∧ steady(TRAJ_OUT) ∧

flight(AIRCRAFT,NEL) ∧…

而不会写成

NEL=f(AIRCRAFT,TRAJ_ENT,TRAJ_IN, TRAJ_OUT)

因为专家对这个 f 不能做任何说明。

15.3.5.4　符号表示

正如在第 15.2.1 节中所提到的，实际中所采取的解决方案是不能完全令人满意的，因为这个解决方案试图在一个有严格定义的领域中，使用一个粗糙的量化表示来代表一个数值函数。例如：当距离和时间都很小时，则不能确定速度值。

15.3.6　算法对结构化对象的应用

让我们首先回忆在 OGUST 中选择一个常量的匹配原则。在 OGUST 中，一个常量代表一个对象，为了发现它们的共同属性，我们寻找最相近的对象。在 ATC 的应用中，惟一真实的对象是飞机，但是，如同在第 15.3.4.2 节中所见到的，引入虚构的对象来代表这些实例。在这些虚构的对象中，我们可以挑选出两种：

- 描述全局环境并且独立于其他对象的对象，例如控制器的当前工作负载；
- 代表飞机属性的对象，比如飞机的类型、航道。

要计算两个飞机之间的相似度，我们必须将与后一种虚构对象相关的共有属性考虑在内。

让我们看下面两个简化了的例子：

E_1 =traj_init(AIRCRAFT$_1^1$, TRAJ _ IN$_1^1$) ∧ steady(TRAJ _ IN$_1^1$) ∧ type
(AIRCRAFT$_1^1$,TYPE$_1^1$)∧jet(TYPE$_1^1$)∧
solution(AIRCRAFT$_1^1$,SOLUTION$_1^1$) ∧
change_course(SOLUTION$_1^1$)∧
to_the_right(SOLUTION$_1^1$)∧
traj_in(AIRCRAFT$_1^2$, TRAJ _ IN$_1^2$) ∧
steady(TRAJ _ IN$_1^2$) ∧
type(AIRCRAFT$_1^2$,TYPE$_1^2$)∧
jet(TYPE$_1^2$)∧
solution(AIRCRAFT$_1^2$,SOLUTION$_1^2$) ∧
change_course(SOLUTION$_1^2$)∧
to_the_left(SOLUTION$_1^2$)
E_2 =traj_init(AIRCRAFT$_2^1$, TRAJ _ IN $_2^1$) ∧
steady(TRAJ _ IN$_2^1$) ∧
type(AIRCRAFT$_2^1$,TYPE$_2^1$)∧
jet(TYPE$_2^1$)∧
solution(AIRCRAFT$_2^1$,SOLUTION$_2^1$) ∧
change_course(SOLUTION$_2^1$)∧
to_the_right(SOLUTION$_2^1$)∧
traj_in(AIRCRAFT$_2^2$, TRAJ _ IN$_2^2$) ∧
steady(TRAJ _ IN$_2^2$) ∧
type(AIRCRAFT$_2^2$,TYPE$_2^2$)∧
super_jet(TYPE$_2^2$)∧
solution(AIRCRAFT$_2^2$, SOLUTION$_2^2$) ∧
speed_regulation(SOLUTION$_2^2$)

在每个例子中都有两架飞机，且选择一架飞机就有四种匹配它们的方式。为了知道哪架飞机匹配，OGUST 搜索飞机每种组合的共有属性。它必须找到一个飞机的属性的惟一途径就是在这些例子中搜索代表这架飞机出现的常量。例如，让我们来考虑第一个例子中的第 1 架飞机，AIRCRAFT$_1^1$，这个常量出现在原子公式 traj_init(AIRCRAFT$_1^1$， TRAJ _ IN$_1^1$),type(AIRCRAFT$_1^1$, TYPE$_1^1$) 和 solution(AIRCRAFT$_1^1$,SOLUTION$_1^1$)中。换句话说，这架飞机的属

性如下：轨道、类型及改动过的路线。这两个例子中的所有飞机都有这些属性。此外，我们所表达的最重要的信息不是飞机有类型这样的事实，而是类型是喷气式这个事实。因此，要计算两架飞机之间的相似度，我们必须考虑代表它们属性的虚构对象的相似度。

考虑到这种结构化对象，我们对两个对象间的相似度的计算做了修改，如下所示。

- 对于两个同类型的基本对象 O_1 和 O_2

  $\text{new\_sim}(O_1, O_2) = \text{sim}(O_1, O_2)$,

  对于两个同类型的结构化对象 O_1 和 O_2，组成它们的对象分别为（O^1_1，…，O^n_1）和（O^1_2，…，O^n_2）

  $\text{new\_sim}(O_1, O_2) = \sum_i \text{sim}(O^i_1, O^i_2)$.

注意，在这个应用中，我们只有一种结构化对象：飞机。每架飞机都由相同类型的基本对象组成，诸如在区域垂直平面中它的航迹，在退出区域时它的航迹，等等。因此，这两架飞机的基本对象是一一对应的。一般来说，计算 $\text{new_sim}(O_1, O_2)$会更加复杂。

一旦选中待匹配的常量，我们用一个变量 vg 代替这两个常量，接着，就必须开始 15.2 节所描述的结构匹配。仍然有两种可能的情况：

- 这两个常量 O_1，O_2 代表基本对象，结构匹配可以马上开始。
- 这两个常量 O_1，O_2 代表结构化对象。我们对所包含的基本对象进行结构匹配。我们用（O^1_1，…，O^n_1）和（O^1_2，…，O^n_2）代表组成 O_1 和 O_2 的基本对象。对同类型的每对对象（O^j_1，O^j_2），我们引入一个新的变量 vg_j，接着开始结构性匹配。

先前在 OGUST 中开发并且应用到这些例子中的所有常量的循环，现在只应用到代表飞机的结构化对象和代表全局环境的基本对象上。不过，这种新的循环可以很容易地被扩展到更加复杂的结构中。

15.3.7　重写归纳

表示学到的规则的形式对专家来说是不自然的。为了将其展现给专家，我们必须将 OGUST 学到的概念转化成令人熟悉的形式。我们使用的表达方式是问卷形式。我们用一个模块来使这个步骤自动化（算法 15.5）。它将第 15.3.4 节描述的过程反过来了。

输入

- 一个列表 ATT_ENV，用于描述全局环境的属性。
- 一个列表 TYPE_OBJ。由列表 $L_i(type_i, att_i^1, \ldots, att_i^n)$所组成。其中 $type_i$ 是一个类型名，

$att_i^1, \ldots, att_i^n$ 是此类型的一个对象的属性。

- 一个列表 SYM_REL，给出在对象间存在的定位关系的名字。
- 一个合取式 E，其元素描述一个例子。

输出

- 一个列表 LIST_ENV，由描述全局环境的(Attribute,Value)对组成，
- 一个列表 LIST_OBJ=（L_1，…，L_p），其中每个列表 L_i 代表一个结构化对象，形式为(Name,Type,(Attribute_i^1, Value_i^1),…，(Attribute_i^n, Value_i^n))，
- 一个列表 LIST_REL，给出对象间存在的关系。

初始，LIST_ENV=LIST_OBJ=LIST_REL=ϕ

1. 对于 ATT_ENV 中的每个属性 att,

- 搜索一个原子 att(O)，它的谓词是 att，
- 搜索一个原子 val（O）val≠att，val 是这个属性的值，
- 从 E 中取出 att(O) ∧val(O)，
- 把(att,val)对加入到 LIST_ENV 列表。

2. 对 TYPE_OBJ 的每个列表 Li:$(type_i, att_i^1, \ldots, att_i^n)$，

- 搜索 E 中类型为 $type_i$ 的对象 $O_i^1, \ldots, O_i^k$，等等，搜索原子 $type_i(O_i^j)$，从 E 中抽出 $\text{type}_i(O_i^j)$，加入 LIST_OBJ 中新的 $L_i^1, \ldots, L_i^k$ 列表，其中 $L_i^j = (O_i^j, \text{type}_i)$，
- 对每个对象 O_i^j，

—搜索原子 $p_l(O_i^j, v_l)$，然后再搜索原子 $val_l(V_l)$，

—把(p_l, val_l)对加入列表 $L_i^j = (O_i^j, \text{type}_i)$ 中，

-- 在 E 中抽出 $p_l(O_i^j, v_l) \wedge val_l(V_l)$。

3. LIST_REL 由剩余的原子所组成（我们可以检查是否每个谓词都属于 SYM_REL）。

算法 15.5 表示形式的改变

例如，由图 15.2、图 15.3 和图 15.4 描述的例子可以归纳为另一个例子，

我们对该例子变化的目的也是用来解决冲突：

G=(∧ (present_work_load(vg_3))(steady(vg_3))(foreseen_Word_load(vg_4)) (steady(vg_3)) (angle_routes_before(vg_5))(sharp(vg_5))(angle_routes_after(vg_6)) (same(vg_6)) (direct_route(vg_1 , vg_7)) (direct_route(vg_2 , vg_8)) (null(vg_8)) (rate_raise(vg_1 , vg_9)) (rate_raise(vg_2 , vg_{10}))(type(vg_1 , vg_{11})) (superjet(vg_{11})) (type(vg_2 , vg_{12})) (jet(vg_{12})) (traj_init(vg_1 , vg_{13})) (up(vg_{13})) (traj_init(vg_2 , vg_{14})) (steady(vg_{14}))(traj_enter(vg_1 , vg_{15})) (up(vg_{15})) (traj_enter(vg_2 , vg_{16})) (steady (vg_{16})) (traj_in(vg_1 , vg_{17})) (up(vg_{17})) (traj_in(vg_2 , vg_{18})) (steady(vg_{18})) (traj_exit(vg_1 , vg_{19})) (steady(vg_{19})) (traj_exit(vg_2 , vg_{20})) (steady(vg_{20})) (destination(vg_1 , vg_{21})) (destination(vg_2 , vg_{22}))...(solution(vg_1 , vg_{26})) (change_direction(vg_{26})) (parallelization_routes(vg_{26})) (solution(vg_2 , vg_{27}))

这个　阶的表达形式现在被转换回专家所熟悉的形式，如下所示。

预期工作负载

——稳定

冲撞发生前飞机的航线角度估计

——锐角

冲撞发生后飞机的航线角度估计

——相同航线

当前航线与直接出口航线之间的角度估计

——vg_1：—

——vg_2：0°

类型

——vg_1：超音喷气式

——vg_2：喷气式

在冲撞分析时垂直平面上的定性航迹

——vg_1：向上

——vg_2：平稳

在退出区域时垂直平面上的定性航迹

——vg_1：平稳

——vg_2：平稳

解决方案

——vg_1：改变方向，路线平行

这表示如果预知控制器载荷是 steady 并且有两架飞机 vg_1 和 vg_2：

- 它们在冲撞发生前的夹角是一个锐角，
- 它们在冲撞发生后航线是相同的，
- 对飞机 vg_2 来说，当前航线与退出方向航线之间的夹角为零度，
- vg_1 是一个超音喷气机，
- vg_2 是一个喷气机，
- 垂直平面上定性航迹在冲撞分析时，对于 vg_1 是上升的，并且对于 vg_2 是稳定的，
- 垂直平面上定性航迹在进入区域时，对于 vg_1 是上升的，并且对于 vg_2 是稳定的，
- 垂直平面上定性航迹在区域中，对于 vg_1 是上升的，并且对于 vg_2 是稳定的，
- 垂直平面上定性航迹退出区域时，对于 vg_1，vg_2 都是稳定的，

于是我们应该运用改变 vg_1 方向使其平行该路线。

15.4 总结

我们已经表明了基于知识的归纳技术中的知识，不仅仅只需要一种归纳工具来利用专家的领域知识（本章称为 BK），使用 BK 演绎出的一阶逻辑表示，对专家来说并不友好。因此，除了泛化工具之外，我们需要其他工具来转化成以一阶形式表示的例子和专家知识。一旦学习发生，我们需要将其翻译回可被专家接受的表达形式。这些翻译步骤并不简单，它们包括一些选择，必须与专家仔细讨论，因为它们会严重影响到学习。

学习过程的成功经常依赖于一种恰当的表示知识的形式，包括基本词汇的选择。显然，如果一个概念依赖于一个因素，而如果我们无法用自定义的语言的原始项来表示这一因素，我们将不能学到这个概念的良好定义。语言原始项的选择特别依赖于领域。因此我们不可能使这一步自动化，只能通过与专家对话的方式得到。后者可以通过源于知识获取的一些技术来获得改进。在 ATC 的应用中，这种选择导致了精心构造的调查表，它包含那些对于控制

人员的决策似乎非常重要的语言特征。我们还得选择一种逻辑形式，这里选择的是一阶形式，而不是更经典的属性-值表示。一阶逻辑的表达能力比属性-值表示要强。但是多数现存的系统并没有选择它，因为使用它缺乏效率。我们已经展示，在某些情况下，比如 ATC 中，我们会被迫使用它。一旦选定一阶形式，则必须确定语言中的哪些原始项将成为谓词，哪些必须是常量。在我们的应用中，我们的选择模仿了面向对象的表示。另一个任务是获取正例和反例，并把它们转换到我们选择的形式。对 ATC 来说，这项任务可以分为几个步骤。实际情况中，专家没有时间完成预定的问卷。因此，专家只写出为记住冲突所必要的信息及解决冲突的途径。之后，专家把它转录到一张问卷。一个友好的接口能使一个人存储答案，然后由一个模块将这些属性-值的表示形式翻译成一阶形式。

最难的问题是获得 BK 库。通过与专家讨论来获得，但我们意识到这远不能令人满意。当表达 if ...then...的规则时，专家经常忘记一些条件，这样会给出重要的但其实是错误的规则。而且，专家并没有给出所有的规则，所以我们得到的是一个不完全的 BK。不正确的知识会导致错误的泛化，而缺少知识会导致过度泛化。因此，领域专家验证所学规则这一步非常重要。当专家否决一项规则时，我们必须找出错误所在，无论是在概念的定义中，还是例子中或是 BK 里。

另一个问题是数值型值的问题，它只被解决了一部分。这个应用展示了将数值型的值引入到一阶泛化工具中的必要性。

最后一个被部分解决的问题是归一化问题。在这个特定的应用中，专家并不知道存在函数所隐含的关系。这就是为什么我们用最简单的解决办法，把归一化函数替换成虚构的常量的原因，其实际值将应用其他的定理计算得到。

在飞行器交通控制应用中，我们必须解决的所有任务中，需要实现的就是归纳工具，把例子存储为一个属性-值表示的接口，以及转换模块，这些都已经用 Lelisp15.2 版写成。

据我们所知，空中交通控制还没有实现完全的自动化。因此，我们不能宣称我们的工作已经直接应用到一个系统中。一旦我们的合同完成，它的成果将归 CENA 所有，而且它已经应用该系统来做研究了。

致谢

本项工作得到了CENA合同的部分支持，我们感谢J.J.Cannat和G.Bisson，他们在LRI致力于这个合同的工作。我们还要感谢GRECO-PRCIA的法国科学研究部不断的支持。

再版于 *Knowledge Acquisition*，Volume 5，Y.Kodratoff，C.Vrain，获取空中交通控制一阶知识，1—36，1993，得到了伦敦学术出版有限公司（Publisher Academic Press Limited）的批准。

参考文献

Avizienis A., Ball D.E.(1987). On the Achievement of a Highly Dependable and Fault-Tolerant Air Traffic Control System. Computer, February 1987, Volume 20, Number 2, pp. 84-92.

Bisson G. (1992). Learning in FOL with a similarity measure. Proceedings of AAAI, San Jose,California, 13-17 July 1992.

Bleistein S., Goettge R., Petroski F., Wiseman R. (1987). Capacity Management of Air Traffic Control Computer Systems. Computer, February 1987, Volume 20, Number 2, pp. 73-83.

Bratko I., Lavrac N. (1987). Progress in Machine Learning, Proceedings of the 2nd European Working Session on Learning, Bled, Yugoslavia, May 1987, Sigma Press.

Buchanan B.G., Shortliffe E.H. (1984). Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, 1984.

Cannat J.J., Vrain C. (1988). Machine Learning applied to air traffic control. Colloque Inter-national Homme-Machine et Intelligence Artificielle dans les domaines de 1'Aeronautique et de 1'Espace, Toulouse, France, 28-30 September 1988, pp. 265-274.

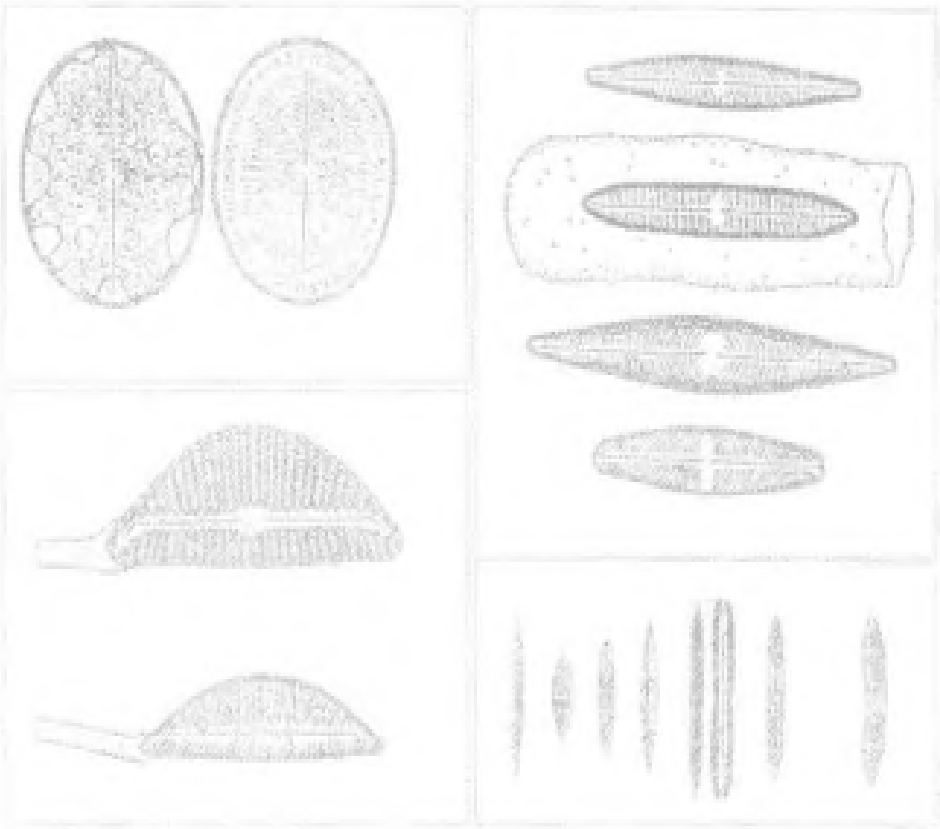
Chaib-draa B., Mandiau R., Millot P. (1988). CIS: Cooperating Intelligent System. Colloque International Homme-Machine et Intelligence Artificielle dans

les domaines de l'Aeronautique et de l'Espace, Toulouse, France, 28-30 September 1988, pp. 315-329.

Forgy C., McDermott J. (1977). OPS, a domain independant production system language. Pro-ceedings 5th International Joint Artificial Intelligence Conference, Cambridge, Massachusetts,1977,pp.933-939.

第 5 部分

医学和生物学



第16章 机器学习在医学诊断中的应用

Igor Kononenko， Ivan Bratko 和 Matjaž kukar

摘要

虽然机器学习能从有限的病人描述中归纳出可靠的诊断算法，这种诊断工具并不能真正地，也不打算代替医师，但却可以作为提高医师的工作效率的辅助工具。根据一些令人信服的试验及本章阐述的结果表明，在机器学习的帮助下，医师诊断病人的正确率将会提高。当在医学诊断中应用机器学习系统时，学习系统将遇到几个特殊需求。本章并不是全面回顾机器学习在医学方面的应用，而是讨论几个在医学诊断和预测问题中运用机器学习的相关论点，阐明过去发展的几个在应用中的有疑问的课题。我们将讨论在医科诊断中所使用的不同机器学习算法各自的优点和缺点。

16.1 介绍

机器学习算法主要分为三类（Michie 等人，1994）：统计或模式识别方法（如 k 邻近算法、判别分析和贝叶斯分类器）；符号法则的归纳学习（如决策树的自上而下归纳、判别法则、逻辑程序归纳）；人造神经网络（如反向传播学习的多层前馈神经网络、Kohonen 的自我组织网络和 Hopfield 的联想记忆）。

在一些小的专业诊断问题上，机器学习似乎很适用于医学诊断。在专业化的医院和部门中，正确诊断的数据将会以医学记录的形式得到应用。所有要做的只是将病人的记录和已知的正确的诊断数据输入电脑，运行学习算法。这当然过于简化，但原则上，医学诊断知识可以自动地从过去解决的病例描述中得到。应用所得到的分类器不仅可以帮助医师，在他们诊断新的病人时，提高诊断的速度、精度和可靠度，而且可以帮助训练实习生或非专业的医师诊断一些有特殊诊断问题的病人。

机器学习系统如今应用在许多医学领域，如肿瘤学(Bratko 和 Mulec，1980；Zwitter 等人，1983；Kononenko 等人，1984；Bratko 和 Kononenko，1987；Elomaa 和 Holsti，1989)，肝脏病理学(Lesmo 等人，1982)，肝炎的生存几率预测(Kononenko 等人，1984)，泌尿学(Kononenko 等人，1984；Bratko 和 Kononenko，1987；Roškar 等人，1986)，甲状腺病例诊断(Horn 等人，1985；Hojker 等人，1988；Quinlan 等人，1987)，风湿病学(Kononenko 等人，1988；Karalič 和 Pirnat，1990；Kern 等人，1990)，craniostenosis 综合病症诊断(Baim，1988)，皮肤病诊断(Chan 和 Wong，1989)，心脏病学(Bratko 等人，1989；Clark 和 Boswell，1991；Catlett，1991)，神经心理学(Muggleton，1990)，妇科医学(Nuñez，1990)和围产期学(Kern 等人，1990)。

然而，并不是所有的学习系统都是同样适合的。当在医学诊断中应用机器学习系统时，系统必须具备几个必要条件。本章分几个方面讨论机器学习在医学诊断和预测问题上的应用，如决议的可靠性和透明度。我们从过去发展的几个不同应用来举例说明问题：瘤起源定位，乳癌复发预测，甲状腺病例诊断，风湿病诊断，股颈侧骨折恢复中并发症预测。我们讨论在医学诊断中的几个不同的机器学习算法的优缺点：决策树的自上而下算法，基本和似然贝叶斯分类器，K 邻近算法，反向传播的使用权值削减学习算法的多层前馈神经网络，以及前向特征构造预测(LFC)。

本章由如下几部分组成。在下节中我们描述诊断病人的一个典型过程并记录得到的结果，这些结果是在以前的一些医学问题上应用机器学习所得到的。第 16.3 节比较医师和机器学习系统的诊断业绩。第 16.4 节描述机器学习（ML）系统在医学诊断应用时所必备的条件，并比较符合条件的 7 个 ML 算法。第 16.5 节讨论为什么在实践中 ML 在医学诊断上的应用被（不被）接受。

16.2　医学诊断

下面是一个典型的诊断过程。在医生给病人会诊时，将会得到病人的既往病史数据，然后在给病人初步检查后，医生将记录病状数据。依据病史数据和病状数据，病人还要接受试验性检查。然后医生根据整个与病人健康状况有关的有用记录做出诊断。根据诊断给病人进行治疗，治疗后，整个过程

将被重复进行。每次重复过程所做的诊断可能被确认、或被精简、或被否决。最后的诊断精确度依赖于医学方面的问题。有时第一次诊断的结果即是最后诊断结果，而有时最后的诊断结果要在治疗有效后才能确定：在某些问题上可能无法获得100%可信赖的最终诊断结果。例如，在肿瘤根源的定位问题上，最后的诊断结果要在确定肿瘤位置的手术中获得，尽管此检查有时是可避免的，或被其他的试验检查所代替，除非真的有必要获得核实的诊断结果。在做过胸部切除手术后的乳癌复发率预测的问题上，最后的预测结果不可能在手术后的 5 年内得出。在泌尿学中失禁类型的诊断问题上，实践中由于没有验证诊断的切实有效的方法所以无法得到最后的诊断结果。

众所周知，医学诊断是主观性的，它不仅依赖于一些有用的数据，而且还依赖于医生的经验、直觉、偏见，甚至其心理状况。一些研究表明，不同的医生给同一病人所做的诊断，甚至同一医生在不同时间（一周的不同天和同一天的不同时间）给同一病人所做的诊断可能会有很大的不同。

机器学习可用来自动地从过去治疗过的病人的验证过的最后诊断记录中提取诊断规则。自动抽取的诊断知识可以帮助医生给病人做出更加客观和可靠的诊断。

16.3 医生与机器学习诊断结果的比较

具有代表性的是，自动产生的诊断规则要比在同一条件下，即专门的医学专家在可以利用与机器相同的信息时所做的诊断的精确度稍好。表 16.1 比较了在四个不同的医学诊断问题上两种机器学习算法（基本贝叶斯判别法和辅助算法(Assistant) ）的效果与四个医学专家平均效果的比较。这些诊断问题包括：肿瘤的根源定位问题，乳癌复发率的预测问题，甲状腺病的诊断，以及风湿病学。试验中所使用的数据收集在卢布尔雅拉(斯洛文尼亚)的中心医科大学。以下是有关诊断问题的简单描述。

表 16.1 在 4 个医疗领域不同分类器性的比较

分类器	初步瘤		乳癌		甲状腺		风湿病	
原始贝叶斯	49%	1.60bit	78%	0.08bit	70%	0.79bit	67%	0.52bit
助手	44%	1.38bit	77%	0.07bit	73%	0.87bit	61%	0.46bit
医生	42%	1.22bit	64%	0.05bit	64%	0.59bit	56%	0.26bit

- **肿瘤定位问题**：如果病人体内的肿瘤位置是已知的话，对病人做医学

移置手术成功的可能性将会大大增加。诊断任务是根据病人的年龄、性别、癌组织类型、差异程度以及 13 处发现转移的可能位置，从 22 处可能的肿瘤源位置中确定一处。我们试验中所用的是卢布尔雅拉的肿瘤研究院的 339 个已知肿瘤位置的病人的数据。

- **乳癌复发率的预测**：做过乳癌切除手术的病人在 5 年内复发的几率大概为 20%。为了更好地治疗此类病症，有必要根据病人的年龄、肿瘤的大小及位置、淋巴腺数据来预测复发的可能性。内科医学专家的问题是，长时间的观察（5 年）往往只能获得很少的实际经验。卢布尔雅拉肿瘤研究机构关于 288 个在手术后 5 年内乳癌复发的病人的数据，被用在我们的试验中。
- **甲状腺疾病**：诊断问题是从 4 个可能的诊断结果中决定一个：年龄、性别、组织数据、试验测试结果。然而，每天内科医学专家使用了许多对电脑计算进程没有帮助的多余的诊断信息。我们试验将使用卢布尔雅拉医科大学中心医药中心门诊部提供的 884 个病人的最终诊断数据。
- **风湿病学**：诊断难题是从 6 组可能的记录和现状数据的诊断结果里选择一个，而风湿病诊断中内科医师使用超过 200 个诊断结果。然而，一般的医生必须得判断风湿病和整型疾病的病人是否需要接受专家进一步的诊断和治疗。这些判断是不可靠的，在风湿病专家的判断中，有超过 30%的判断是错误的。我们试验中将使用卢布尔雅拉美国医学中心风湿病门诊部提供的 355 个病人的最终诊断数据。所有的诊断都通过额外观测、试验测试和 X 射线的验证。

我们试验中使用的数据的特征总结在表 16-2 中 (分别为 PRIM，BREA，RHEU)。诊断的类别和平均信息量显示了诊断的难度。属性的个数可近似告诉我们病人病情好坏的程度。主要的类别百分数近似等于大多数可能诊断中的优先诊断的可能性。这实际上就是默认情况下，不管什么病人，总是选择最可能的诊断的分类器的精确度。

我们试验中每运行一次就随机地从实例中选择 70%用来学习，30%用来测试，表 16.1 中所列的结果是 10 次运行的平均结果。平均精确度随着平均每个答案的信息分给出来。信息分是消除类别优先可能性所带来的影响的性能度量，它能被用于各种各样的不完善和不确定的答案中。它的完整定义在附录中。这种测量非常重要，因为在每个领域，默认的分类器可以达到很高

的分类精确度。

表 16.2 医疗数据集的基本描述

领域	类别号	属性号	属性平均取值	实例号	主要类别比例	熵
THYR	4	15	9.1	884	56	1.59
PRIM	22	17	2.2	339	25	3.89
BREA	2	10	2.7	288	80	0.73
LYMP	4	18	3.3	148	55	1.28
RHEU	6	32	9.1	355	66	1.73
BONE	2	19	4.5	270	65	0.93
HEPA	2	19	3.8	155	79	0.74
DIAB	2	8	8 8	768	65	0.93
HEART	2	13	5.0	270	56	0.99

我们分别对每个领域的四名医学专家的诊断的精确度做了预计。从一组训练数据中随机抽取部分病人，将他们的状况打印在纸上，但并不标注诊断结果。要求医师为每位病人选择最可能的诊断结果。表 16.1 展示了每个领域四名医学专家的平均选择结果。医师们在卢布尔雅拉的美国医学中心接受了测试。从纸面上对乳癌和风湿病做出诊断有些违背常理，其他两个领域的诊断通常由实践来判断。

在分类的精确度和平均信息分这两方面，利用算法都要好于医师所做的诊断。然而，这些结论还需要一些限定，那就是必须强调这些试验中医师和电脑是使用相同的信息。这在医学实践中是不切实际的。在对病人的检测中，医生往往能通过直觉印象发现病人的某些状况，这些是无法描述的，也不能够输入到电脑中。这些信息在某些病例中对于能否得到更可靠的诊断可能起到决定性的作用。表 16.1 的精确度结论只能这样来理解，它是算法做得有多好的一种预测，并不能说医生做得有多坏。尽管机器学习能从有限的病人描述中产生更可靠的诊断算法，但这种诊断工具并不能，也不会代替医生。但它可以作为一个辅助工具来改善医生的诊断。本章所要阐述的结论和从其他试验获得的令人信服的描述就是，在机器学习的帮助下，医生的诊断精确度可以得到提升。

16.4 选择适当的机器学习系统

本节我们将给出机器学习用于医学诊断发展中所必须满足的具体条件，并简明地描述了几种学习算法。我们将比较所有的算法在医学诊断和预测问题上的效果，以及讨论它们在医学诊断上的适用性。

16.4.1　机器学习系统的具体要求

机器学习用于解决医学诊断任务必须满足以下几个特点：效果好；适当处理丢失数据和噪声（数据错误）的能力；诊断知识的透明度；对结果解释的能力；减少保证可靠诊断的测试的能力。

本节我们首先讨论这些要求，然后再比较一下几个典型的机器学习算法，更加具体地描述这几点。

- **效果好**：算法必须能够从有效数据中提取重要的信息。对新病例的诊断精确度必须尽可能高。一般来说，多数算法的效果至少要和医生的一样好。通常，在使用相同的病人的病状描述时，分类器的分类准确性要好于医生。所以，如果可以测量医生的准确度的话，其结果可以用来当做机器学习系统在给定问题上所需准确度的最低界限。

  尽管在一些病例上，有的算法做得比其他算法要好，但在大多数学习问题上，很多算法在分类精确度上达到类似的效果。所以，基于效果这个标准，几乎没有什么算法可以被排除。然而，一些学习途径要利用有效数据来测试，预测效果最好的一个或几个学习方法才可以被应用。

- **处理丢失数据**：在医学诊断中，经常有一些病人只有少部分描述数据。机器学习算法必须能够恰当地处理这些病人的不完善的记录。

- **处理噪声**：医学数据有时会不一致或出现错误。所以医学应用中的适当的机器学习算法必须能够有效地处理噪声。

- **诊断知识的透明度**：产生的知识和结论的解释必须清晰地展现给医生，以使得他或她能够分析和理解。理想的情况是，自动产生的知识能够针对于给定问题以一种新颖的观点提供给医生，并能够以一种清楚的形式揭示医生以前没有见过的新的联系和规律。

- **解释能力**：系统必须能够在诊断新的病人时解释结论。当面对一个新问题的古怪的解决方案时，医生要求系统能够做进一步的解释。否则他无法理解系统的建议。医生接受黑匣分类器的惟一可能是该分类器的分类精确度比其他分类器和医生本人好很多。然而，这种情况几乎不可能出现，本章的作者还不知道任何这样的例子。

- **减少测试次数**：在医学试验中，病人数据的收集往往是昂贵的、耗时

的和对病人有害的。因此，仅使用较少的病人数据就能得到可靠的诊断，这样的分类器是理想的。这可以通过给所有的候选算法提供有限的数据来进行考察。然而，确定一个合适的数据集的过程可能很耗时，因为这本来是一个组合问题。一些机器学习系统能自己选择出合适的属性子集，也就是说在学习进程中进行选择，这比没有学习功能的其他系统要可靠得多。

16.4.2 测试的算法描述

本节我们简要描述几个我们试验中使用到的算法：Assistant-R，Assistant-I，LFC，基本和似然贝叶斯分类器，反向传播权值消减分类以及 k 邻近算法。

- **Assistant-R**：它是辅助学习系统对决策树从上至下归纳的重复应用。其基本算法要追溯到 Hunt 发展的 CLS(概念学习系统)。它被几个作者重复应用和改善。其主要特点在于属性特征的二进制表示，决策树的α和β 剪枝，不完善数据的处理和计算空枝分类的基本贝叶斯分类器的使用。

  辅助算法（Assistant）及其扩展（Assistant-R）的主要区别在于 ReliefF 被用来做属性选择(Kononenko，1994)。ReliefF 是 Relief 的扩展版，由 Kira 和 Rendell(1992)开发，它是非近视的启发式的度量工具，能够评估属性的质量，甚至在属性间有很强的条件依赖性的情况下也能如此。例如，Relief 能在奇偶问题上有效地评估属性的质量。另外，在无论多合适的情况下，Assistant-R 都使用 m 似然评估代替关系频率。m 似然评估多次显示出改善机器学习算法效果的性能(Cestnik，1990)。
- **Assistant-I**：Assistant-R 的变体，不一样的是不采用 ReliefF，而是使用信息收获变量作为选择标准，这与最初的辅助算法（Assistant）一样。然而，和 Assistant 的另一个区别是保留了 m 似然评估。
- **前向特征构造（LFC）**：Ragavan(1993；Ragavan 和 Rendell，1993)在他们的前向特征构造(LFC)算法中使用有限扩展对决策树进行从上至下的归纳，找出属性间的有意义的条件依赖关系。他们用一些数据集展示了有趣的结果。Robnik(1993)对他们的算法进行了扩展，现在用在我们的试验中。LFC 产生二进制决策树。在每个节点，算法使用逻辑

操作(并、交、非)从原来的属性构建出新的属性。在扩展的二进制属性中选择出最好的属性，处理器在训练实例的两个子集中重复递归操作，以符合选定的属性的两个值。出于建设性的归纳，使用了有限的预计。可能有用的结构空间被强制，这取决于条件熵（对属性价值的评估标准）的几何描述。为进一步减少搜索空间，算法也限制了搜索的深度和宽度。

由于 LFC 使用预计，它比 Assistant 的贪吃算法更少近视。LFC 和 Assistant-R 的试验结果对比与使用贪吃搜索和 ReliefF 版的扩展策略相结合的效果形成了鲜明对比。为得到可与 Assistant-R 相比的结果，我们给 LFC 加上了剪枝和可能性评估工具。所有的测试都设定了一组默认参数(扩展深度 3，传播大小 20)，尽管在一些领域调节参数会得到更好的结果。然而，参数值过高可能增加 LFC 的搜索空间而使得算法不切实际。

- **基本贝叶斯分类器**：它使用基本贝叶斯公式来计算在给定了所有属性的值 V_i 的情况下，实例分到 C 类的几率。假设给定的类中的属性条件独立：

$$P(C \mid V_1 \cdots V_n) = P(C)\prod_i \frac{P(C \mid V_i)}{P(C)}$$

一个新的实例以最大的计算概率分到该类中。我们使用 m 似然估计(Cestnik，1990)。对于先验概率，我们使用连续性拉普拉斯定律：

$$P(C) = \frac{N(C)+1}{N+\#\_of\_\text{classes}}$$

这里，N 是所有试验的数目，$N(C)$是 C 类试验数目。这些先验概率在条件概率的 m 似然估计中用到：

$$P(C \mid V_i) = \frac{N(C \,\&\, V_i) + m \times P(C)}{N(V_i) + m}$$

这里，参数 m 在关系频率和先验概率的贡献间进行调节。在我们的试验中，参数 m 设为 2.0(此设定常被用做默认值，根据经验，能得到令人满意的结果(Cestnik，1990))。

基本贝叶斯分类器的相对性能可用于进行条件独立的属性评估。

- **似然贝叶斯分类器**：Kononenko(1991)开发出基本贝叶斯分类器的扩展版，能明确地找出不同属性值间的联系。如果两个不同属性值 V_i 和 V_j

的联系被找出，它们被认为不是相互条件独立的。因此，基本贝叶斯公式中的

$$\frac{P(C|V_i)}{P(C)}\times\frac{P(C|V_j)}{P(C)}$$

由下面的部分来代替：

$$\frac{P(C|V_i,V_j)}{P(C)}$$

对这种替换，条件概率 $P(C|V_i, V_j)$取可靠的近似值是必要的。所以，算法可在非基本和概率近似值的可靠度之间调节。

- **反向传播权值消减**：多层前馈人工神经网络是由两层或多层处理单位——神经元组成的相互关联的神经层所构成的分等级的网络。学习算法的任务是确定相互联络的神经元之间的合适的权值。多层前馈神经网络中（Rumelhart 和 McClelland，1986）的反向传播误差法是一个知名的学习算法，也是训练人造神经网络的最流行的算法。反向传播最著名的问题是如何选取合适的网络拓朴结构和适当的训练数据。基本算法的一个扩展是使用权值消减技术（Weigand，等人 1990）来解决此问题。方法是从许多隐藏的神经元开始，引进标准函数以惩罚神经元连接中大的权值。使用了这种标准函数，算法在训练中抵消了适当数目的权值和神经元，以便在训练数据上获得恰当的归纳。
- ***k*-NN**：*k* 最邻近算法。给定一个新的实例，算法寻找其 *k* 最邻近的训练实例，然后将此实例分类到出现那些 *k* 邻近实例最多的类中。下节中出现的结论是由曼哈顿距离得到的。使用欧几里得距离得到的结论实际也一样。可以给出关于参数 *k* 的最好的结论，尽管在公正的对比中参数调整仅仅允许出现在训练中，而不是测试数据集上。

16.4.3 医学问题上算法效果的比较

我们比较各种算法在一些医学数据集上的效果。

- 数据集从斯洛文尼亚卢布尔雅拉的医科大学中心获得：确定转移型病人主要肿瘤位置问题（PRIM），预测肿瘤移植手术后 5 年内乳癌复发率问题（BREA），确定淋巴腺疾病类型问题（LYMP），风湿病诊断（RHEU）以及大腿骨骨折恢复预测问题（BONE）。

- HEPA：肝炎病人的存活率的预测问题。数据由卡内基-梅隆大学的 Gail Gong 提供。
- 数据集从 StatLog 数据库获得（Michie，等人 1994）：糖尿病诊断（DIAB）和心脏病诊断（HEART）。

上述医学数据集的基本描述在表 16.2 中给出。这些数据集的试验结果展示在表 16.3 和 16.4 中。结果是对每个领域试验 10 次的平均值和标准偏差值。每次试验，随机抽取数据集 70%的数据用于学习，30%的数据用于测试。

表 16.3　医疗数据集上学习系统的分类准确率

领域	LFC	助手-I	助手-R	原始贝叶斯	半原始贝叶斯	后传	k-NN
PRIM	37.1±4.9	40.8±5.1	38.9±4.7	49.2±3.9	48.2±4.3	43.4±4.4	42.1±5.0
BREA	76 1±4.3	76.8±4.6	78.5±3.9	77.3±4.2	78.9±3.6	76.4±4.1	79.5±2 7
LYMP	82.4±5.2	77.0±5.5	77.0±5.9	84.2±2.7	84.5±2.5	82.5±4.5	82.6±5.7
RHEU	60 6±4 7	64.8±4.0	63.8±4.9	67.1±4.7	68.0±3.6	68.7±4.1	66.0±3.6
BONE	69.4±4.6	72.1±4 1	70.8±6.2	69 7±5.7	70.6±4.5	71.1±4.1	69.0±4.0
HEPA	79.0±5.3	77.2±5.3	82.3±5.4	85.3±3.9	86.4±2.9	80.2±3 6	82.6±4.9
DIAB	69.2±3.0	71.1±2.8	71.5±2.6	77.0±1.8	77.1±2.0	72.2±2.2	73 9±2.5
HEART	77.3±5.2	75 4±4.0	77.6±4.5	84.5±3.1	84.5±3.6	80.4±2.9	82.9±3.7

表 16.4　医疗数据集上学习系统的平均信息分

领域	LFC	助手-I	助手-R	原始贝叶斯	半原始贝叶斯	后传	k-NN
PRIM	1.02±.14	1.19±.11	1.07±.11	1.61±.14	1.52±.15	1.41±.15	1.15±.11
BREA	0.01±.09	0.02±.08	0.05± 06	0.06±.06	0.07±.04	0.07±.07	0.02±.02
LYMP	0.79±.10	0.62±.09	0 61± 09	0.78±.08	0.77±.08	0.77±.10	0.53±.08
RHEU	0.41±.10	0.43±.08	0 41± 08	0.53±.06	0.54±.08	0.58±.08	0.43±.05
BONE	0.41±.09	0.23±.08	0.25±.06	0.29±.07	0.36±.06	0.35±.06	0.40±.07
HEPA	0.19±.14	0.13±.09	0.22±.11	0.35±.10	0.40±.10	0.25±.07	0.21±.05
DIAB	0.26±.06	0.26±.04	0.27±.04	0.38± 03	0.38±.03	0.34±.05	0 24±.02
HEART	0.52±.10	0.45± 07	0.46± 07	0.64±.05	0.64±.06	0.58±.06	0.46±.04

16.4.4　医学诊断的实用性

本节我们讨论什么样的算法适合 16.1 节描述的要求。表 16.5 总结了关于医学诊断和预测问题的应用发展的实用算法的比较结果。

表 16.5　用于医疗诊断的不同算法适合程度

分类器	性能	透明度	解释	减少	处理遗失数据
助手—R	好	非常好	好	好	可接受
助手—I	好	非常好	好	好	可接受
LFC	好	好	好	好	可接受
原始贝叶斯	非常好	好	非常好	不行	非常好
半原始贝叶斯	非常好	好	非常好	不行	非常好
后传	非常好	差	差	不行	可接受
k-NN	非常好	差	可接受	不行	可接受

算法的比较中，只有决策树的创建者才可以选择合适的属性集。遵循了缩减测试数目的规范的算法比其他算法有明显的优势。

从结果的规范来看，这些算法变得更加类似。最好的结果是由基本和似然贝叶斯分类器达到的。在医学数据集中，在给出的类别中属性一般是相对条件独立。医生们尽可能定义条件独立的属性。人类的思考是线性的，独立的属性使得诊断处理更容易些。所以，贝叶斯分类器在医学数据上表现出明显的优势并不令人奇怪。有趣的是，*k*-NN 算法在这些领域的效果也很好。

BREA 数据集的信息分（如表 16.4 所示）显示出没有一个学习算法能解决这个问题。这表明了属性并不相关。此结论同时由内科专家确认。手术后五年内乳癌复发的预测问题是现在还没有解决的一个医学问题。

两个不同版本的辅助算法（Assistant）有类似的效果，除了在 HEPA 领域，Assistant-R 能得到明显更好的效果（99.95%的置信度，使用两个跟踪 t 检测）。一个详细的分析显示了，在此问题中，ReliefF 发现了给出类中两个属性的明显的条件独立性。当被认为独立时这两个属性的信息分很少。这就是为什么 Assistant-I 不能在数据中找出规律的原因。另一方面，此领域的其他（多余的）与这两个属性有相类似的信息关联的属性是可利用的。这就是为什么基本贝叶斯分类器能做得更好的原因。我们试图通过加入两个条件独立的属性给基本贝叶斯分类器以提供一个额外属性。然而，效果却是一样的。

对于 DIAB 数据集，Ragavan 和 Rendell（1993）宣称他们的 LFC 算法达到了 78.8%的分类精确度，同时指出了其他几个没有建设性归纳的算法的可怜的分类精确度（最多 58%）。然而，我们的结论（看下面）和 StatLog 工程（Michie，等人 1994）的结果显示了其他算法在此领域效果差并不是由于它们缺少建设性归纳。在使用 DIAB 数据集的试验中，除了贝叶斯分类器做得更好外，所有的其他分类器效果都一样。LFC 在 LYMP 领域达到了比其他两个指导式算法更好的结果，这就使是否使用建设性归纳看起来没有太大的作用。然而，LFC 在 RHEU 领域效果较差，而在其他领域这三个算法的效果几乎相等。

在透明度和解释能力标准上，算法间存在很大的差异。

- **k 最邻近算法：**由于 *k*-NN 不具备普遍性，所以其知识描述的透明度很差。然而，为了解释算法的结果，训练集中的最邻近点数目（*k*）被预先确定出来。这个方法类似于领域专家对基于已知的类似实例所使用的方法。医学家评价认为其解释能力是可以接受的。

- **基本和似然贝叶斯算法**：在此，知识描述由一张条件概率的表构成，医学家们似乎对这个感兴趣。所以，这种知识描述被认为是很好的。另一方面，贝叶斯分类器的结果可以被认为是信息收获的总和（Kononenko，1993）。对于找出实例属于 C 类的很重要的信息数量由下式给出

$$-\log_2 P(C|V_1,\cdots,V_n) = -\log_2 P(C) - \sum_i (-\log_2 P(C) + \log_2 P(C|V_i))$$

所以，贝叶斯分类器的结果可以被解释为从有利的属性中或从给定类的对立类中得到的信息的总和。对于似然贝叶斯分类器，除了产生加入的属性/评估值对，处理完全一样。此时，加入的评估值代替了简单的属性评估值。

这种信息收获可以列在表中来概括支持/反对结果的迹象。表 16.6 提供了一个结果的典型解释。每个属性都有一个相关的力度值，以描述该属性提供的信息比特数目。它可以有利于，也可反对分类结果。解释的主要好处之一是能用到所有的可利用的属性。医学家认为这种解释非常好。他们感到贝叶斯分类器解决问题的方式和他们诊断的方式很类似，即，他们也概括支持/反对给定诊断的迹象。

- **反向传播神经网络**：它采用不透明的知识描述，并且通常不能很容易地解释它们的结论。这是因为有大量的真实权值都在影响着结果。有时可以从训练过的神经网络中抽取有象征意义的规则。然而，这些规则很庞大并且关系复杂。Craven 和 Shavlik（1993）将 Quinlan（1993）的 C4.5 系统产生的规则与从一个神经网络中抽取的规则做了对比。从一个神经网络抽取的 *NetTalk* 数据集的规则平均每条规则超过 30 个先例，而相比之下，C4.5 系统只有 2 个先例。这些规则是如此复杂以至于对于一个非技术领域的首席专家几乎不能提供有用的解释。
- **决策树（Assistant-I 和 Assistant-R）**：在没有电脑的情况下，这种算法也能使用并且简单易懂。树中属性的位置，尤其是顶端的属性位置，通常直接取决于领域专家的知识。然而，为产生一般性规则，这些方法采用剪枝技术，彻底降低了树的维数。相对而言，从根部到叶子的路径较短，包含了虽然是最普通但却很少的属性。很多情况下，医学专家认为用这种树状方式来描述诊断的能力很差，并且信息量不是很充分(Pirnat 等人，1989)。在一些问题上医学专家更喜欢 Assistant-R 产

生的决策树。看起来 ReliefF 的评估更符合医学专家评估属性的重要性的方式。实际上，医学专家认为 Assistant-I 产生的决策树结构是奇怪的和不自然的。

- **前向特征构造（LFC）**：它也产生决策树。然而，在每个结点中，一种潜在的复杂的逻辑的表示法代替了简单的属性评估值。所以产生的树可以变得更小。解释可以描述领域中正确的观念。然而，在树的较低层解释经常太细而没有意义。基于节点的复杂逻辑描述法，用来将一个实例分类的属性的数目要多于一般的决策树。

16.5 实践中的认同

虽然在医学诊断中应用各种各样的机器学习算法的结果看起来很好，但是这项技术并没有广泛应用于医学实践中。医学专家自己给出了不同的原因：

- 知识表示太呆板。描述病人的属性集必须固定。当主观的、不正式的和模糊的概念（如知觉、影响等）不能以正式的符号的方式表达时，由规则使用得到最终诊断的信息被严格地限制在已定义好的参数上。
- 内科专家表示如果他们不能肯定最后的诊断结果时，通常需要进一步的考察（试验测试）以验证诊断结果。当进一步的考察很容易时，内科专家认为在诊断过程中辅助是没必要的。但在预测中进一步的考察也不能确定预测的准确性。所以预测问题对于机器学习比诊断问题更具吸引力（Zwritter，等人 1983）。
- 内科专家经常声称他们在做结论时太忙而没时间使用辅助工具。在每天的实践中将数据输入电脑以在诊断进程中使用电脑帮助显得太耗时间和精力。
- 个人主义也是内科专家拒绝使用新的诊断技术的主观原因。他们觉得诊断问题，特别有时可能是很紧急的和敏感的人物，全部交给机器做的话，他们自己会没有责任权和控制权。
- 我们有时也由于毫不理智的原因拒绝使用机器诊断。一些内科专家描述了如下的原因。一些内科专家认为诊断是他们职业中的额外的高智商任务。所以，此任务需要高深的知识，无法预料的思想，特别是直觉。诊断是一种艺术，它不可能被解释和形式化。它怎么能由电脑来

完成呢？如果电脑能做到这点的话，那将严重打击专家们的自信心。

在过去我们曾提出决策树技术在医学诊断问题上的几个应用（Kononenko，1984；Roškar，1986；Hojker，1988）。除了上面提到的问题外，决策树还存在以下的不足之处：

- 分类和学习对数据的丢失很敏感（Quinlan，等人 1989），这是医学数据中经常出现的情况。
- 产生的判别规则一般包含太少的属性（Pirnat，等人 1989）。因此，判别的解释太差，一般不能精确地支持产生的判别规则的结论。

为解决上面两个问题和提高自动产生式分类器的可靠度和透明度，多层策略逼近变得重要起来（Michalski 和 Tecuci，1994）。一种思想是开发多个用几种不同方法的分类器，然后在新的问题上使用所有的分类器并组合它们的判别结果。这种途径类似于医院里的决议，由一群内科医生共同解决一个较难的病例总比一个医生单独解决要好。我们在腿骨骨折恢复率的预计问题上采用这种思想（Kukar，1996）。在此项研究中不同分类器的分类结果被组合起来（使用似然贝叶斯公式）作为最终的判别结果，这可以解释为所有的单个判别的权重的总和。内科医生感到使用多层策略逼近比单独使用决策树时系统的可靠性更高并更易理解。

我们的目标是使机器学习系统更容易被未接触机器学习的专家们使用。这就要求工具必须很直观，其界面必须功能强大并能在视觉上吸引用户。许多算法为达到更好性能而设置一些数字变量，这就强烈需要一个自动设置变量的方案。为了使用户更容易地使用预备数据，机器学习系统必须具有标准数据库接口（如 DBase）。

16.6　总结

我们的试验表明在医学领域中，各种各样的分类器效果大体一样，所以，在应用中选择那种分类器主要取决于它的解释能力。试验表明医生们更喜欢由贝叶斯分类器和决策树分类器提供的解释：Assistant-R 和 LFC。然而，看起来在解决新问题上，与其选择一种最好的分类器，还不如使用所有的分类器，然后综合它们的判别结果。医师们发现组合分类器结果的方法很好地改进了诊断系统的可靠性和易理解性。

关于机器学习在医学诊断中未来的任务，我们有如下几个观点：

- 机器学习技术在医学诊断实践中还没有被接受，仅用做指出技术可能性的详细描述。然而，这种技术可能性和实际开发的不平衡的状态并不能维持很长时间。
- 机器学习在实践中应用的发展如此缓慢的原因之一（也许是最合理的原因）是，机器学习技术的应用将给医生增加一些应用工具和仪器。任何一个新工具的使用是不受欢迎的，它将使医师本已很复杂的工作变得更加复杂。所以机器学习技术必须融合到已有的工具中，这样使用起来就和往常一样简单和自然。
- 以机器学习为基础的诊断程序将和医师使用的其他仪器一样：作为另一个有用的信息源来帮助医师提高诊断的精确度。最后是否接受这些信息，像往常一样，都是医师的权利。

致谢

试验数据的收集和整理离不开卢布尔雅拉中心医科大学的内科专家 Matjaž Zwitter 教授、Sergej Hojker 教授、Tomaž Silvester 教授和 Vlado Pirnat 教授所提供的无价的帮助。我们感谢他们在收集数据，口译数据，测试中心医科大学的内科专家的诊断效果，对结果的口译以及对不同分类器的解释能力的估计等方面所做的一切。本项研究得到了斯洛文尼亚科技部门的大力支持。

附录：信息分

除了分类器的精确度，我们试验中还要测量平均信息分（Kononenko 和 Bratko，1991）。这种测量估计了分类器的适当的似然响应和先验概率产生的影响。平均信息分定义为：

$$Inf = \frac{\sum_{i=1}^{\#\text{testing instances}} Inf_i}{\#\text{testing instances}}$$

第 i 次测试实例的分类信息分定义为：

$$Inf_i = \begin{cases} -\log_2 P(C_i) + \log_2 P'(C_i), & P'(C_i) \geq P(C_i) \\ -(\log_2(1-P(C_i)) + \log_2(1-P'(C_i))), & P'(C_i) < P(C_i) \end{cases}$$

C_i 是第 i 次测试实例的类，$P(C)$是 C 类的先验概率，$P'(C)$是分类器返回的概率。如果正确类返回的概率比先验概率大，信息分是正的，即表明获得的信息是正确的。这可以解释为对正确分类必要的先验信息减去对正确分类必要的后验信息。如果正确类返回的概率小于先验概率，信息分是负的，即表明获得的信息是错误的。这可以解释为对不正确分类必要的先验信息减去对不正确分类必要的后验信息。

分类精确度和信息分之间最主要的区别在下例中说明。设原先的分类的概率分别为 $P(C_1)=0.2$ 和 $P(C_2)=0.8$，分类器返回的分类的概率变为 $P(C_1)=0.4$ 和 $P(C_2)=0.6$。如果正确的分类是分到 C_1，那么信息分是正的，而基于分类精确度来讲，分类器返回的分类是错误的。如果正确的分类是分到 C_2，则信息分是负的，而基于分类精确度来讲，分类器返回的分类是正确的。

分类的精确度在某些特殊的例子中表现得很不一致，而信息分就要稳定得多。在一个特殊的例子里，我们有一个数据集，它的属性间都是不相关的，而且刚好有 50%的实例属于一类，50%的实例属于另一类，则基于概率的分类器将达到的精确度近似为 50%，而对于事先默认的分类器，它将每个实例都分到主要的类中，精确度将为 0%。对训练实例分类的一个细小的改动将彻底地改变最终的分类精确度，使之近似于 50%。更大的改动，如假定 80%的实例属于一类，20%的实例属于另一类，默认分类器的精确度将为 80%，而基于概率的分类器达到的精确度近似为 0.8×0.8+0.2×0.2=68%。然而，对这两种分类器信息分将始终保持在 0 比特左右，它能够正确地反映出这两种分类器都不能从属性中抽取出任何有用的信息。

参考文献

Baim P.W., A Method for Attribute Selection in Inductive Learning Systems, IEEE Trans.on PAMI, Vol. 10, No. 6, 1988, pp. 888-896.

Bratko I., Kononenko I., Learning Rules from Incomplete and Noisy Data, in B. Phelps(ed.), Interactions in Artificial Intelligence and Statistical Methods, Hampshire: Technical Press, 1987.

Bratko I., Mozetic I., Lavrac N., KARDIO: A Study in Deep and Qualitative Knowledge for Expert Systems, Cambridge, MA: MIT Press, 1989.

Bratko I., Mulec P., An Experiment in Automatic Learning of Diagnostic Rules, Informat-
ica, Ljubljana, Vol. 4, No. 4, 1980, pp. 18-25.

Catlett J., On changing continuous attributes into ordered discrete attributes, Proc. Euro-pean Working Session on Learning-91, Porto, March 4-6 1991, pp. 164-178.

Cestnik B., Estimating Probabilities: A Crucial Task in Machine Learning, Proc. European Conf. on Artificial Intelligence, Stockholm, August, 1990, pp. 147-149.

Cestnik B., Kononenko I. and Bratko I., ASSISTANT 86: A knowledge elicitation tool for sophisticated users, in: I. Bratko, N. Lavrac (eds.): Progress in Machine learning, Wilmslow:Sigma Press, 1987.

Chan K.C.C., Wong A.K.C., Automatic Construction of Expert Systems from Data: A Sta-tistical Approach. Proc. IJCAI Workshop on Knowledge Discovery in Databases, Detroit,Michigan, August, 1989, pp. 37-48.

Clark P., Boswell R., Rule Induction with CN2: Some Recent Improvements, Proc. Euro-pean Working Session on Learning-91, Porto, Portugal, March, 1991, pp. 151-163.

Craven M.W., Shavlik J.W., Learning symbolic rules using artificial neural networks, Proc.10th Intern. Conf. on Machine Learning, Amherst, MA, Morgan Kaufmann, 1993, pp. 73-80.

Elomaa T., Holsti N., An Experimental Comparison of Inducing Decision Trees and Deci-sion Lists in Noisy Domains, Proc. 4th European Working Session on Learning, Montpeiller,Dec. 4-6 1989, pp. 59-69.

Hojker S., Kononenko I., Jauk A., Fidler V. Porenta M., Expert System's Development in the Management of Thyroid Diseases, Proc. European Congress for Nuclear Medicine, Mi-lano, Sept., 1988.

Horn K.A., Compton P., Lazarus L., Quinlan J.R., An Expert System for the Interpretation of Thyroid Assays in a Clinical Laboratory, The Australian Computer Journal, Vol. 17, No.

第17章　学习对生物医学信号进行分类

Miroslav Kubat，Irena Koprinska 和 Gert Pfurtscheller

摘要

医生们分析生物信号，如心电图（ECG）、脑电图（EEG）、心率或呼吸等，都是以要检测特定病理生理异常状态为目的。这一分析的花费，按照时间和高素质专业人员的努力来计量的话，就促使人们试图使这一过程在某种程度上实现自动化。本章着重在两个领域，应用一个学习系统来从样本中归纳出可用于模式识别的所需要的知识。这些领域的复杂本质使得应用符号机器学习方法，诸如规则学习或决策树，十分困难。同样，使用神经网络会引起匹配反向传播算法本质的严重问题。但是，事实已证明使用决策树来初始化神经网络可以旁路掉绝大多数警告，从而促使在对测试样本的分类准确率方面，性能上有较大提高。

17.1　介绍

从机器学习的观点，这一章主要涉及**概念学习**。希望系统能归纳出两个或者更多概念的内部表示（即模型）以便帮助进行分类：当一个新的尚未分类的样本被提交给电脑时，机器依据已经存储在内存中的模型检查它的属性值，并给这个样本标上与之最接近模型的概念名称。

在符号机器学习中，对概念的识别通常都伴随着一个解释，像那些医疗手术中一样，在决策伴随着不可逆转结果的应用中，这一解释是不可缺少的。在这些领域，模型最好被表达成容易解释的符号，诸如规则或决策树。相反，其他许多应用强调自身识别的正确性，而不需要解释。对一个确定邮政编码的系统而言，应该最大程度地保证被识别数字的正确性而不能指望操作员知道这成千上万个实例的分类。这里介绍的两个研究项目也是如此。在对睡眠

阶段分类和对大脑发出产生运动命令的预测上，要求系统完成无数次基于不能被直接解释的变量的识别。

这两个领域的主要特性将在下一节中介绍。接着，将研究特定的学习技巧。第 17.3 节将介绍基于树的神经网络（TBNN），第 17.4 节介绍基于树和半径函数的网络（TB-RBF）。第 17.5 节报告了记录在给定领域这些学习方法性能的试验结果。第 17.6 节从涉及这两个技术的角度，即初始化神经网络和对医学数据的分类，来简要地说明其贡献。

17.2 两个医学领域

为了使读者了解我们研究工作的背景，这里简要介绍机器学习技术已被证明很有用处的两个典型的医学领域。

17.2.1 睡眠分类

对于人类的睡眠，医学专家辨认出了 3 个或 4 个基本的阶段，以及做梦的阶段。其中最后一个阶段伴随着眼睑下快速的眼球运动，因此也被称为 REM 阶段。夜间分布在这些阶段里的失调现象可以显示出睡眠紊乱的症状，比如失眠症，成人的睡眠性呼吸停止，甚至婴儿急性死亡综合症（SIDS）。为了检测这些失调现象，专家们分析了人体睡眠活动图（Hypnogram），这张图的水平轴代表时间，垂直轴表示不同的睡眠阶段。图 17.1 就是一个经过简化的理想化的例子。

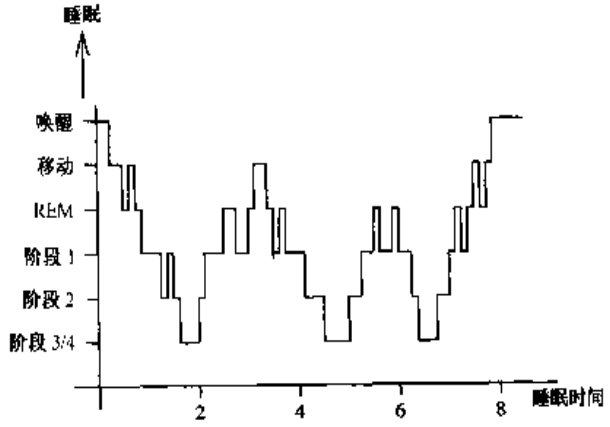


图 17.1 理想化例子。一个成人睡眠者，3/4 阶段分为两个不同阶段；在儿童中，另一个状态，即“运动”也考虑到了

为了画出睡眠活动图，专家使用了一个 8 小时的夜间脑电图（EEG）、眼球运动图（EOG）及肌动电流图（EMG）记录仪，分别显示脑的活动（EEG）、眼球的运动（EOG）和肌肉的收缩（EMG）。每一个独立的阶段对这些信号都具有自己特有的感觉模式。画出一个人一晚上的睡眠活动图就需要一个高水平专家数小时紧张的劳动（这是很昂贵的，有时甚至找不到这样的专家），这一事实使得用计算机程序自动记录睡眠活动的想法很具有吸引力。这就是本章第一个所需完成的研究任务：开发一个基于上述信号的分类器，用以识别不同的睡眠阶段，并可以准确到能与人类专家相比。

为了研制和测试这一系统，使用了 8 个不同的数据文件。每个文件都是某个主体 8 个小时的记录值。接受测试的都是 6 个月大的婴儿，为方便在科学期刊上发表，我们分别称这些受测的婴儿为：BR，KR，RA，KL，BU，PR，GO 和 FR。数据文件中包含了 780 到 960 个样本，每个样本都用 15 个数值属性来描述。这些样本又被专家分类成了 7 个睡眠阶段（包括非 REM 阶段、REM 阶段，也包括“运动阶段”、“清醒阶段”和“人造阶段”）。

重要的是，学习程序将看到与医学专家不同的属性。这样做的原因就是人类专家能依据信号特殊的波形来完成分类。这些波形，虽然训练有素的专家可以很容易找到它们，但是却很难被计算机的实现且形式化。为机器学习需要而用来描述样本的属性包括：从鼻腔和胸廓呼吸变化率的信号，1~4 赫兹的脑电波和眼运动信号的能量，心跳速率及其变化率，左右手的肌运动（微小变化），肌肉的收缩，以及从两个不同的电触头得到的脑活动信号的 Hjorth 参数（活跃度、运动度和复杂度）。因此用了 15 个属性来描述一次的测量。这一测量在整夜的睡眠中，每 30 秒进行一次采样。为了学习的需要，这些 15 维向量已被专家分类（他们了解更多在前面没有提到的更复杂的特性）。专家对样本的分类利用了 Guilleminault 和 Souquet 法则（1979）。

关于数据学习复杂性的几点说明。首先，专家的睡眠分类可以是不一致的，因为睡眠阶段并不能够准确定义。另外，属性值还包括了传感器的噪声，在（3%~5%）的几率上，所获得的只是任意数值。另一棘手的问题就是这些属性之间存在关联。最后，先前的研究结果表明，这些属性并不具有相等的重要性，某些值也许是冗余或完全不相关的（但在这一特定的研究中，我们假定相关性事先并不知道）。

较早的实验表明，对从一个主体获得的样本进行学习，然后运用归纳出

的知识对另一主体进行分类并不可行，因为所需的医学信号在很大程度上依赖于受测的睡眠者（Kubat,Pfurtscheller 和 Flotzinger，1994）。因此，定义一个更小的目标：专家对从单个睡眠者所获得样本集合的子集进行分类。系统在学习这一分类后，再对同一睡眠者剩余的数据自动进行分类，这一较小的目标甚至也能显著节省昂贵的专家人力费用。

17.2.2 从脑电波信号中识别肌肉运动指令

格拉茨技术大学目前开始了一项雄心勃勃的科研项目，其目的就是构建允许发展人脑-电脑直接交互的平台。这一想法将为同那些患有严重肌侧神经硬化症或那些患有被称为自闭症病人进行交流提供新的渠道。这种病人完全麻痹并且失去了对肌肉运动的控制能力，他们甚至不能使眼球运动。但是，他们的头脑工作正常及大脑皮质层活动伴随着特殊电信号的推动，从这些方面能够推测出某些具有明确定义的脑状态，它们可以通过脑电波信号检测出来，能够被用来做一些简单交流。

尽管人们不能期望机器可以读懂思想，但简单的命令，如“移动右手”或“移动左手”，都在感觉运动皮层区域上测量的脑电波信号中有着显著的相位波形，可以明显地分辨出它们。关于这一现象，本书的一位作者给出了详细的介绍（可参阅文献 Pfurtscheller，Flotzinger 和 Neuper，1994）。由于即使对于脑已瘫痪的的被测者（指令离开了他们的大脑但确没有到达肌肉），也存在信号的相位现象，所以对特定的脑电波模式与特定肌动命令在健康人体中关系的研究（这对实验来说更容易获得），可以为其后的以病人为中心的研究提供极为重要的线索。早期的研究强烈支持这一假说（Pfurtscheller，Flotzinger 和 Kalcher，1992）。

获取学习及测试样本的步骤如下所述。被测者坐在一张置于暗室中的舒适椅子上（为了使得不相关脑皮质层的活动减到最少），并被随机地命令做如下的肌肉运动：“移动右手”、“移动左手”、“移动一只脚”、“活动舌头”。其情形如图 17.2 所示。在得到警告信号（WS）的提示后，测试者会得到提示（CUE），告诉他/她应该做上述哪项运动。被测者在响应刺激（RS）后，会很快做出相应运动。

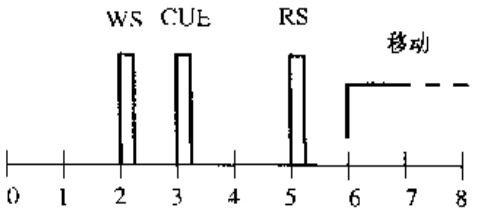


图 17.2　试验范例

在被测者头上紧紧地安上一个罩子，里面有几个电极以记录脑电波信号。这一信号经过滤波后，会再提交给某个附加的设备，用和机器学习没有联系的技术进行数据处理。关于数据获取和处理细节的讨论，超出了本章的要求和范围，但是，原则上讲，任一运动（被分类为以上四类中的一种），都用一矢量来描述运动前脑电波的能量。罩子中的电极按照国际标准排列（如图 17.3 所示）。在总共 56 个电极位置中，一个神经生理学家帮助我们从中挑选了 11 个与我们研究有关的电极。

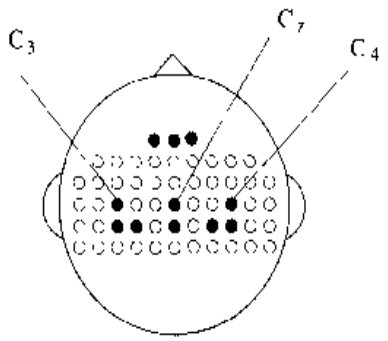


图 17.3　电极位置（(C_3，Cz，C_4)，根据国际 10-20 系统）
选择 11 个黑色电极位置用于研究

机器学习程序的任务就是归纳出四类运动模型，并使用这些模型对未来的测量值进行分类。若这一识别被证明是可行的，那就为将来在人脑-电脑交互这一领域的工作奠定了良好的基础。例如，“往右移”或“往左移”的概念，可以用来控制电脑显示器上鼠标箭头的移动，箭头所指向的屏幕上字符可显示出病人的愿望。

机器学习程序可获得的数据并不完美，这是因为信号不仅受给定概念的影响，而且受到主体不同程度的注意、他们不稳定的“内心设置”、警戒程度变化、记录时外界干扰、不使用最优频带，以及其他许多因素的影响。考虑到在测量向量中的大量属性参数，学习样本的数量是较少的，并且无法确保这些样本具有代表性。此外，一些属性也许和分类毫无关系，同时也许会忽

略一些实验者不知道的重要属性。这些性质之间具有很强的相关性，这使得学习任务更加复杂。

17.3 基于神经网络初始化的决策树方法

决定实际应用中使用概念学习程序的成败，最重要的因素或许就是表示语言的丰富程度以及对给定问题的适应程度。为说明清楚，考虑两个通常使用的具有代表性的概念学习的例子：神经网络和决策树。

神经网络的表示语言很丰富，因为它拥有大量的自由参数（通常是权值）。理论家已经证明了定理，即神经网络可以表示出任何一个从 m 维空间到单位超立方体的映射 f: $R^m \rightarrow [0,1]^n$（Cybenko，1989）。但是，这个映射能否被某一学习算法找到就是另一回事了。神经网络可以被训练成表示复杂决策，但是它也会遇到无数的陷阱，比如局部极小值、鞍点以及过度训练的危险。此外，假如学习程序可以收敛一个假设（结论），该结论对独立测试样本具有高质量的分类结果，那么，就必须提供许多的训练样本。

决策树也是一种通用的近似方案——在给定训练集合情况下，假设这个集合不包含逻辑矛盾，它可以完成任何映射。此外，它不需要像神经网络那么多的样本。另一方面，决策树利用一组与轴平行的超平面来勾画概念的事实，会在实际问题中限制其对独立测试集合的分类准确性。例如，二维概念仅能被大小不等的矩形来近似描述，这一模型在树规模较小时，会很粗糙；而规模大的树，又很容易过分逼近训练集合。这也意味着，学习中，无法较好地归纳描述未知的实例。

这些考虑促成了将这两种方法结合到一起来形成一个单一系统的思想：决策树产生近似概念描述，然后将这一近似描述交给神经网络做进一步的调整。神经网络训练开始时已经非常接近全局最小值了，超越大多数局部极小值点和鞍点。这在相当大的程度上减少了成功训练神经网络所需要的样本数，同时提供了一种比决策树更强的描述语言。

本节的目标就是提出一种在多层神经网络中可能的实现方法。下一节将介绍如何利用决策树来初始化另一个常用的连接形式——径向基函数网络。

17.3.1　TBNN 基本思想

自 Sethi（1990）提出一种将决策树映射到神经网络的简单算法之后，文献中又出现了几种可供选择的方案，可参阅 Brent（1991），Park（1994），Ivanova 和 Kubat（1995）以及 Sahami（1995）。这些算法中的绝大多数都是基于这样一个事实：决策树实际代表了一组 d 个逻辑合取的规则，这里 d 是不同类别的数目。

图 17.4 描述，图 17.5 说明了一个简单机制，它将一些析取范式映射到包含两个隐层的神经网络。OR 层中每个概念包含一个神经元，AND 层中的单元数等于决策树中的叶子数。位于输入层和 AND 层间的是第一个隐层，其作用是执行由决策树产生的测试要求。读者可以很容易看到用 at1=0.8∧at2=0.4∧at3=0.9 所描述的样本，将会激活与决策树最右边叶子标示相同的输出神经元，其他的输出神经元没有输出。值得注意的是，对这个特定的例子，at2 的值并不影响由树或神经网络所给出的类别标识。

（1）利用某个决策树归纳算法获得一个决策树；

（2）将树转换为一个多层感知单元，其中输出层完成析取，第一隐层完成合取，第二隐层确定哪个属性间隔满足条件；

（3）软化间隔边界，利用后传算法调整网络。

图 17.4　系统 TBNN 的算法

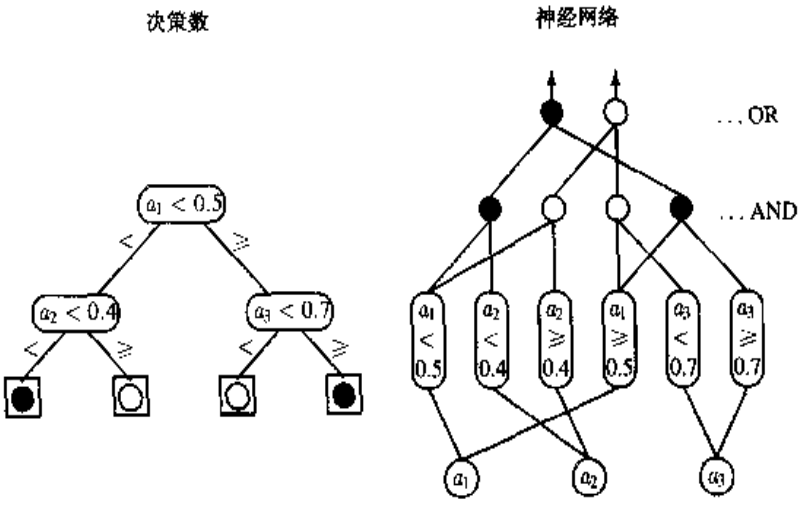


图 17.5　将决策树翻译为一个神经网络

从树到神经网络的映射仅仅是将一种表示机制变成另一种。重要的是，神经网络系统具有比原来决策树更大的自由度，这些新出现的参数可以用于“微调”概念模型。为了充分利用连接形式的丰富性，TBNN 系统（基于树的神经网络，Ivanova 和 Kubat，1995）用以下三个步骤拓展了表示的灵活性：①初始化权值并使相邻层之间完全连接，以确保网络用的是与原始决策树相同的概念标记样本。为此，需要有初始化权值的公式；②减弱严格的属性-值测试；③利用算法调整连接的权值。

下面各小节将详细介绍每个步骤。图 17.6 说明了下面将要用到的一些约定记号。连接到第 i 个隐含单元的权值记做 $\omega_{i\alpha}$；连接到第 i 个输出单元的权值记做 $\omega_{i\beta}$。

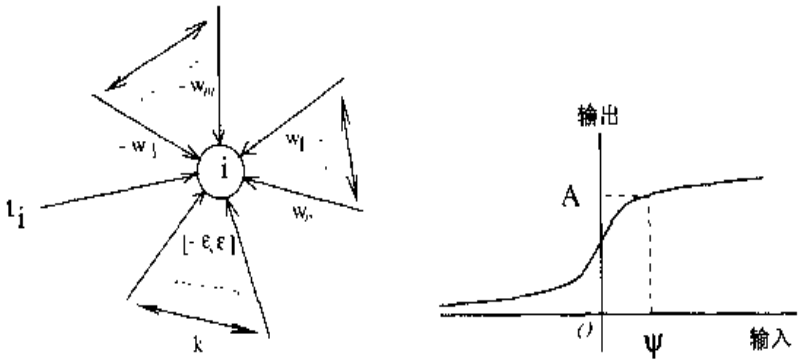


图 17.6　一个神经单元的输入和激活常量 A 的定义

小写 n 表示正权值的数目，小写 m 表示负权值的数目。注意，$\omega_{i\alpha}$ 和 $\omega_{i\beta}$ 表示的是绝对值。每个神经元的阈值分别用 $t_{i\alpha}$ 和 $t_{i\beta}$ 表示。激活常量 A 起着重要的作用，它定义了神经元将处于的活跃状态时的阈值。若输入总和超过某关键值 ψ，那么该神经元就被激活（具有大于 A 的输出值）。求解方程 $A=1/(1+\exp(-h\psi))$ 中的 ψ，确定在神经元激活时，输入的总和必须满足 $\psi > \frac{1}{h}\ln\frac{A}{1-A}$。

17.3.2　初始化权值和相邻层的完全连接

利用决策树中的逻辑编码，OR 层仅有正的输入值。假设在完成树到网的映射之后，增加额外 k 个连接，它们的权值在区间[-ε,ε]的小范围中随机变化，将它们加到一个神经元上，以使其与前一层完全连接。为使 OR 层能表示初始

叶子的析取，必须满足两个条件。

首先，第 i 个 OR 神经元在处于与初始树产生相应分类结果的相同条件下，必须被激活。这意味着其输入受阈值 $t_{i\beta}$ 的影响而减少，应当大于 ψ，这是在至少有一个正规连接下的 AND 神经元被激活的前提下才成立的（反馈给 OR 神经元的输入为 $1 \cdot A \cdot \omega_{i\beta}$，即使其他所有正规连接都提供 0 输入($(n-1) \cdot 0 \cdot \omega_{i\beta}$)，且若附加连接提供最大负输入（$-k \cdot \varepsilon$））。

第二，第 i 个 OR 神经元必须在初始树产生相应分类的相同情况下，处于非激活状态。这意味着其输入受阈值 $t_{i\beta}$ 的影响而减少，当所有正规连接的 AND 神经元均输出最大值但仍被视为未激活($n \cdot (1-A) \cdot \omega_{i\beta}$)时，其输入应小于 $-\psi$，即使在附加连接提供最大正值输入（$k \cdot \varepsilon$）时已是这样了。

可以证明，当权值和阈值设为以下值时，假设，已经选取的常量 A 满足 $A(n+1)-n>0$，这两个条件可被满足（Ivanova 和 Kubat，1995）：

$$\omega_{i\beta} \leqslant \frac{2(\psi + k\varepsilon)}{A(n+1)-n} \tag{17.1}$$

$$t_{i\beta} \leqslant (\psi + k\varepsilon)\frac{n-A(n-1)}{A(n+1)-n} \tag{17.2}$$

可对那些具有 $m \geqslant 0$ 负权值的 AND 神经元做类似处理。为了使 AND 层可以在附加连接存在的情况下，表示初始间隔的连接，就必须满足以下三个条件：

首先，第 i 个 AND 神经元必须在样本沿着树中各分支传播时处于激活状态。这意味着其输入受阈值 $t_{i\beta}$ 的影响而减少，当所有沿着具有正值的正规连接的父辈神经元都输出最小值且仍被视做已激活($n \cdot A \cdot \omega_{i\alpha}$)时，应当比 ψ 大，即使在所有具有负值的正规连接的父辈神经元均输出最大值，仍被视做未激活($m \cdot (1-A) \cdot (-\omega_{i\alpha})$)，并且即使所有附加连接均提供最大负值输入（$-k \cdot \varepsilon$）时也是这样的。

第二，第 i 个 AND 神经元必须在至少有一个具有正值的正规输入为非激活($1 \cdot (1-A) \cdot \omega_{i\alpha}$)时，也处于非激活状态，即使在其他所有具有正值的正规输入均给出最大值($(n-1) \cdot 1 \cdot \omega_{i\alpha}$)，以及即使所有具有负值的正规输入均为 0($m \cdot 0 \cdot (-\omega_{i\alpha})$)，且所有附加连接都提供最大正值输入($k \cdot \varepsilon$)时也是这样。

第三，第 i 个 AND 神经元必须在至少有一个具有负值的正规输入提供最小值时，仍被视为已激活($(1-A) \cdot (-\omega_{i\alpha})$)时，也处于非激活状态，即使在所

有具有正值的正规输入均处于最大值激活($n \cdot 1 \cdot \omega_{i\alpha}$)，以及即使所有其他具有负值的正规输入都是 0($(m-1) \cdot 0 \cdot (-\omega_{i\alpha})$)，且附加连接都提供最大正值输入($k \cdot \varepsilon$)时也是这样的。

当权值和阈值按以下公式设置时，这些条件可被满足。

$$\omega_{i\alpha} \leqslant \frac{2(\psi + k\varepsilon)}{A(n+m+1)-(n+m)} \tag{17.3}$$

$$t_{i\beta} \leqslant (\psi + k\varepsilon)\frac{A(n+m-1)+(n-m)}{A(n+m+1)-(n+m)} \tag{17.4}$$

在 $A(n+m+1)-(n+m)>0$ 时，分母为正值。现在实现的 TBNN 对所有神经元都使用同一个常量 A，这一常数被设为满足每个神经元的不等性的最小值。

17.3.3 弱化间隔和神经网络的微调

决策树的每一个分支都定义了一条规则，它将整个论域分为两部分：一部分对该规则为真，另一部分则为假。规则映射到 AND 神经元，其原理如图 17.7 所示。注意在单个分支上的测试是如何将属性的值域划分为两个或三个区间的。例如，在某一分支上属性 a_1 的测试为 $a_1 \geqslant 0.4$ 和 $a_1 < 0.7$，即 $a_1 \in [0.4,0.7]$，那么样本到达这一分支的叶节点。如果 $a_1 \in [0,1]$，那么在这一分支上的测试将定义以下 a_1 的区间为：[0,0.4)，[0.4,0.7)和[0.7,1]。

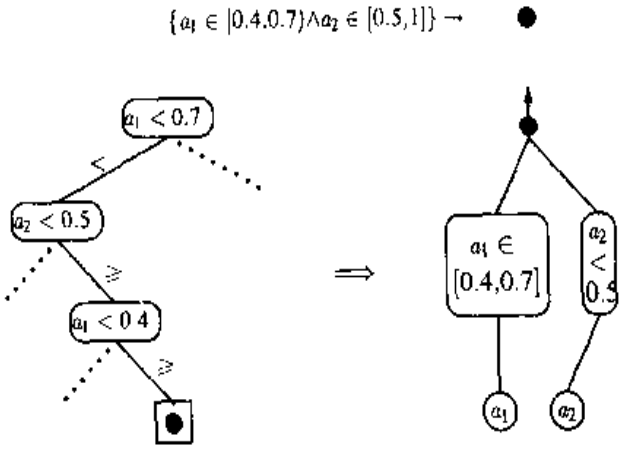


图 17.7 用逻辑形式描述和转换成 AND 神经元的决策树分支。a_1 和 a_2 范围为[0，1]

先前在决策树基础上初始化神经网络的工作表明，如果可以弱化对属性值的直接测试，那么可以使神经网络分类的性能得到改进。换句话说，那些接近区间边界的值在某种程度上应被视为属于两个相邻的区间。

如果样本用矢量 a={ a_1,...,a_n }描述，那么可以通过将每一属性值 a_i 用它的弱化值 $y_i = 1/(1+e^{v_i})$ 代替的方法来实现弱化，其中 v_i 是对每个间隔中心的接近程度，可用下式计算：

$$v_i = \frac{R_i - 2|\mu_i - a_i|}{2R_i} \tag{17.5}$$

其中，μ_i 是间隔的均值，R_i 是间隔的大小，a_i 是给定属性的实际值。

为了进一步微调概念模型，TBNN 应用了由 Rumelhart，Hintom 和 Williams（1986）提出带有动量函数的反传算法。在提交样本后，权值根据常见公式 $\omega_{t+1} = w_t + \eta \cdot \Delta \cdot o + \alpha(w_t - w_{t-1})$ 进行更新，其中 w_t 是第 t 次的权值，Δ是反传误差，o 是上次神经元的输出，η 是学习速率，α是动量因子。

为防止网络的过度训练，TBNN 的反传部分采取下述机制：将训练样本分为两个子集，2/3 的样本用做训练，1/3 的用做模型当前版本的在线测试。后一个子集称为“训练测试集”。在每轮处理训练样本后（也就是每个情节后），系统用训练测试集来检验性能。当对训练测试集的分类正确率下降时，便停止训练，即使此时对训练测试集的均方误差仍在改善。

在所有实验中，使用下列参数：学习速率η=0.2，动量因子α=0.5。

17.4　基于树的 RBF 网络初始化

另一个可以得益于决策树初始化的表示形式就是径向基函数网络（RBF）。下面这一节将首先介绍 RBF 的主要学习策略，然后介绍与前一个系统所采用思路类似的改进方法。

17.4.1　RBF 网络及其参数

在**基础函数**思想后面的合理性来自数学的发现：如果数值向量被非线性地映射到高维空间中，就更容易进行线性分割（Cover，1965）。即使一些理论和实际研究表明，选择进行这一映射的非线性函数对结果影响并不是很大（Powell，1988）。但通常的做法就是采用高斯钟形函数，并按我们的要求定义如下。

设有样本集 p，用向量 $\mathrm{X}_i = [x_{i1},...,x_{in}]^T$ 描述，可用于学习，且有映射函数 $\varphi_j : R^n \to R^m$ 定义如下：

$$\varphi_j(x_i)=\frac{1}{(2\pi)^{n/2}|\sum_j|^{(1/2)}}\exp\{-\tfrac{1}{2}(x_i-\mu_j)^T\sum_j^{-1}(x_i-\mu_j)\} \qquad (17.6)$$

其中，$\sum_j^{-1}$ 是输入向量协方差矩阵的逆，$|\sum_j|$ 是该矩阵的行列式。方程 17.6 描述了一个具有协方差矩阵 $\sum$ 和中心向量 $\mu_j=[\mu_{j1},...,\mu_{jm}]^T$ 的 n 维高斯面。X_i 和 μ_j 之间的距离越大，$\psi_j(x_i)$ 值就越小。每个函数 $\psi_j(x_i)$ 都输出一刻度值，这样第 i 个样本就被向量 $\varphi(X_i)=[\varphi_1(X_i),...,\varphi_m(X_i)]$ 重新描述了。

图 17.8 描述了如何在神经网络的隐层中实现这一映射（1988 年由 Broomhead 和 Lowe 首次将径向基网络引入神经网络一类）。这一层神经元的转换函数由公式 17.6 定义，而输出神经元则具有线性转换函数。仅仅对输出层权值进行训练。额外输入，永远设为 $\psi_0=1$，以便为输出层线性函数提供偏差。Park 和 Sandberg（1991）证明了 RBF 网络的广泛性，只要隐层神经元的数目足够的话，且使用适当定义的高斯中心和协方差矩阵，网络就能够在任意准确率下逼近任何的连续函数。我们将在概念识别任务应用 RBF，该任务可以被认为是函数逼近的一个特别情况：每个输出神经元代表一个概念，如果第 l 个输出神经元具有最高值，则样本将被标示为第 l 个概念。

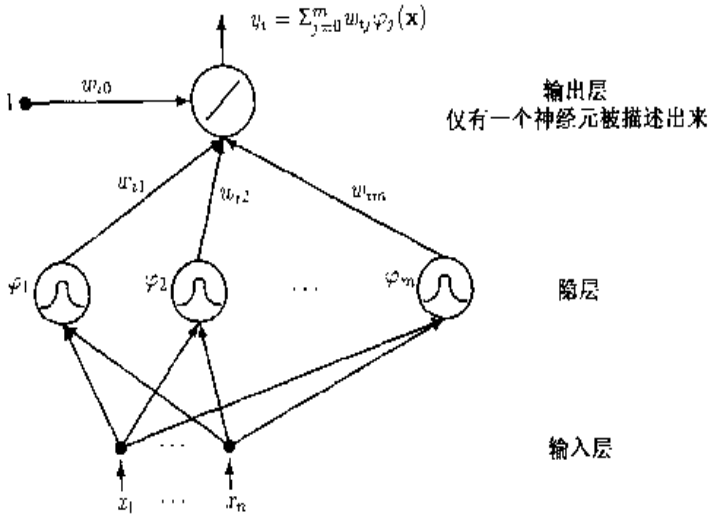


图 17.8 径向基函数网络。函数 $\psi_j(x_i)$ 由中心和标准偏差来定义。在输出层神经元的转换函数是线性的

一个 RBF 网络由其拓扑结构和一组可训练参数来定义。系统 TB-RBF 解决了这两个问题。一个 RBF 网络的可训练参数就是：高斯中心、协方差矩阵和权值 w_{ij}。权值可以利用德尔塔规则很容易训练而成（Widrow 和 Hoff，1960），而协方差矩阵可以直接从输入向量中计算得到，或为简单起见，用标准方差

代替。

通过拓扑结构，我们了解隐层神经元及其与其他层之间的相互连接。在许多实际领域，仅有一小部分属性与每个概念相关。Sanger（1991）利用在多个隐层中 RBF 单元结构中实现与树类似的方法来选择属性。这个方法在属性非常多时会产生许多树。Andrew，Kubat 和 Pfurtscheller（1995）描述了一个系统，它通过爬山搜索来减少输入-隐层的连接数目，这个技术计算量非常大。

至于高斯中心，Poggio 和 Girosi（1990）在输入向量 x_i 的坐标中定位它们，将任务视为一个函数插值问题。当样本很多时，网络就会变大，因此，Lowe（1989）利用了样本集的一个任意子集。将最相关样本包含到网络中来的思想激发了 Cheng 和 Lin（1994）应用一个带有操作符“增加一个神经元”和“删除一个神经元”的搜索技术。类似，Chen，Cowan 和 Grant（1991）应用了单个操作符，“增加一个神经元”，并利用了正交最小平方标准来选择下一个候选。一次增加一个神经元，以及随后微调中心坐标的方法，由 Wettschereck 和 Dietterich（1992），以及 Fritzke（1994）提出。其他还有利用统计聚类的技术来合理减少隐层神经元的数目（Musavi 等人，1992），以及将高斯中心设为聚类质量的中心等方法。

17.4.2　基于参数设置的决策树

系统 TB-RBF 利用一种决策树归纳技术来优化输入-隐层连接的数目及隐层神经元数目，这种方法的计算量相当合理。高斯函数的参数无需计算输入向量协方差矩阵的逆 Σ^{-1} 就可以被确定。

图 17.9 介绍了其基本原理。利用一个标准决策树产生器，从学习数据中归纳出决策树，该决策树将整个二维空间划分为四个不相交区域。其思想就是在每个区域中心定位一个隐层单元。事实上由决策树定义的区域是与轴平行的超平面区域。这就使利用标准方法替代公式 17.6 中的协方差成为可能，其中标准方差可以通过一个简单启发式（为记忆方便，我们在下面考虑，省去下标 j，例如用 ψ 代替 ψ_j）来确定。

假设输入空间仅有一维（n=1）。因此如果用 M_ψ 表示 ψ 的最大值（在高斯中心），在距离中心为 ψ $1.5\sigma^2$ 时，就有 $\psi = 0.3125M_\psi$。类似地，在距离中心 σ^2，

就有 $\psi = 0.6250 M_{\psi}$。我们现在将这个思想归纳到 n>1 维。用 L_k 来表示第 k 个超矩形的边，定义 $P_k = L_k / \sigma_k$，这样，公式 17.6 可以改写为：

$$\psi(x_i) = \prod_{k=1}^{n} \frac{P_k}{L_k \sqrt{2\pi}} \exp\{-\frac{P_k^2 (x_{ik} - c_k)^2}{2L_k^2}\} \qquad (17.7)$$

参数 P_k 决定 ψ 在沿超矩形边界一个维上的值。在我们的实验中，对于所有 k 维，利用了 P_k 的一个固定值。TB-RBF 算法如图 17.10 所示。

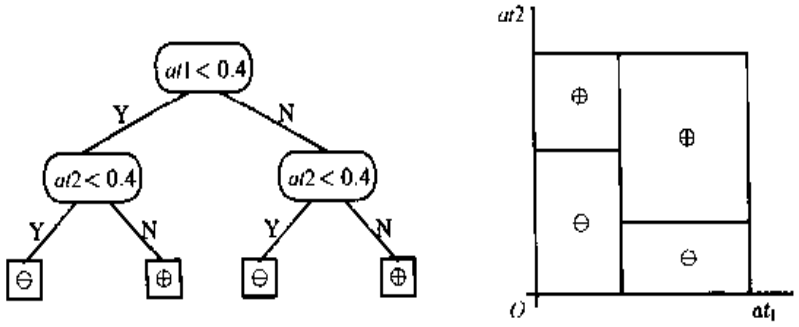


图 17.9　由一个决策树定义的超矩形区域。树的每个分支，被理解为一系列测试来定义这个区域

（1）利用某个决策树归纳算法获得一个决策树；

（2）将树转换为一个 RBF 网络。由树定义的每个超平面，用一个隐层原子来表示，其中转换函数采用等式 17.7；

（3）利用同样的训练样本和 Delta 规则来训练输出层的权重。

图 17.10　系统 TB-RBF 的算法

17.5 试验

试验的目的是研究用前面描述过的机器学习方法初始化神经网络是否更有利，以及新导出的系统在分类效果上是否比其他学习方法更有效。

前面已经提到过，关于这两个领域中的以前的研究揭示：很难归纳得到分类规则，用于归纳学习期间不可见的主题。出于这个原因，我们采纳下面的设定：某一主题的一个测量数据集已经由专家和所提供的学习系统分类。这样归纳分类器就被用来分类这一特定主题的沉睡阶段（Sleep Stages）或者电动命令（Motor Commands）。

对于两种学习系统的效果的评估性试验，其中使用的 8 个数据文件来自睡眠分类领域，3 个文件来自人机对话工程领域。这些数据文件的大小不同。睡眠数据文件包含 770~960 个样本，这些样本具有 15 种属性并被分为 7 类。人脑数据文件包含 150~250 个样本，这些样本具有 11 种属性（实例 A4）或者 44 种属性（实例 B6 和 B8），并且被分为 4 类。

例如，根据 Weiss 和 Kapouleas（1989）建议，对于不可预见数据的分类精确度的估计已经通过**随机样本子集策略**而进行。一个样本集合被随机分为两个无交集的子集，其中一个子集用于学习，另外一个用于测试。试验通过重复不同的划分（我们的实例中重复了 10 次）并将结果平均。对于每个数据文件和每个学习系统都进行这个过程，这样，在我们结果报表中的每个条目都代表了 10 次随机运行的平均值和标准偏差。每个训练集包含给定文件中的 60%的样本，对应的测试集包含剩余的 40%的样本。

出于比较的需要，表 17.1 和 17.2 分别总结了对睡眠数据和大脑数据采用不同的学习方法进行分类的结果。MNN 列包含的结果是根据提出后向传播算法的 Rumelhart、Hinton 和 Williams（1986）的建议，对多层神经网络采用小的随机权值进行初始化得到的。我们的试验包含大量的隐含的神经元，表中的结果是从那些能提供最好的分类精确度的结构中得到的。CN2 是由 Clark 和 Niblett（1989）提出的一个流行的机器学习程序。它的输出是生成规则的形式。两个决策树生成器 ID3（Quinlan，1986）和 C4.5（Quinlan，1993）也同时被运行。然而，只有 ID3 在 TBNN 中被用来初始化神经网络。Kohonen 开发的 LVQ 算法，优化地调节了每类的一系列原型，这些原型被用来基于近邻方法的分类。

表 17.1　各种学习器关于睡眠数据的分类精确度

%	MNN	CN2	ID3	C4.5	LVQ	TBNN
BR	60.0 ± 2.8	72.8 ± 1.7	77.3 ± 3.0	78.1 ± 1.8	81.0 ± 1.5	85.5 ± 3.6
KR	39.7 ± 5.3	56.7 ± 2.6	64.0 ± 1.6	61.7 ± 3.8	66.5 ± 2.5	68.3 ± 4.9
RA	68.3 ± 11.0	75.6 ± 2.4	77.5 ± 1.5	76.8 ± 1.9	78.5 ± 2.5	80.8 ± 0.6
KL	72.6 ± 1.4	70.4 ± 3.3	77.3 ± 1.7	76.6 ± 2.1	78.1 ± 1.4	80.2 ± 1.7
BU	66.4 ± 2.1	63.4 ± 2.9	65.5 ± 4.6	69.7 ± 2.3	72.9 ± 1.5	73.1 ± 1.8
PR	62.8 ± 3.0	61.8 ± 1.7	63.9 ± 2.6	64.2 ± 2.4	65.9 ± 2.1	66.8 ± 2.0
GO	56.8 ± 3.8	54.7 ± 1.8	63.0 ± 2.2	61.1 ± 1.9	65.8 ± 2.4	66.3 ± 2.7
FR	73.4 ± 4.6	74.9 ± 2.0	75.5 ± 1.8	78.8 ± 2.8	79.9 ± 1.5	81.3 ± 1.8

表 17.2 各种学习器关于大脑数据的分类精确度

%	MNN	CN2	ID3	C4.5	LVQ	TBNN
A4	43.2 ± 6.3	41.0 ± 3.1	48.3 ± 3.4	46.9 ± 2.5	50.5 ± 3.3	55.1 ± 1.8
B6	43.0 ± 6.7	35.6 ± 16.9	46.2 ± 5.1	42.0 ± 6.5	48.0 ± 5.3	54.1 ± 3.3
B8	35.6 ± 5.9	35.1 ± 7.5	38.1 ± 5.8	31.0 ± 7.0	42.4 ± 5.2	49.2 ± 3.4

由于带有少量隐藏神经元的 MNN 具有巨大的冒险性（特别是局部最小值），增加隐藏神经元的数量可以消减这些缺点，但是代价是增加的大量自由参数，这需要大量地增加训练样本，这些事实表明 MNN 初始化具有一些令人失望的结果。LVQ，通过分段线性边界来近似概念，在决策树中具有较好的效果并能形成类似的观察器。

总的说来，如果 MNN 被随机初始化，那么初始权值很有可能离解决方案太远而无法方便地集中。与之形成对比的是，通过映射决策树的方法初始化权值将能保证后向传播算法从一个良好的初始状态开始，甚至在限定样本数量后还能得到概念的相对精确的估计。

表 17.3 和 17.4 总结了通过 TB-RBF 获得的两个领域的结果。初步的试验似乎显示由 C4.5 生成的决策树可以很好地适合训练集。因此，我们决定利用 C4.5 能将决策树转变成具有较优的生成规则的事实。随着这些简单规则的生成，我们利用它们来初始化网络。然而，原则上，要保留如下一致性：对应每个隐藏的神经元的每个规则，定义一个超矩形区间。

表 17.3 RBF 网络的性能和特征 —睡眠数据

	C4.5r	TB-RBF	units	connections	trainable
BR	78.4 ± 1.7	81.4 ± 1.3	40.2 ± 2.9	106.8 ± 12.2	20.2 ± 2.1
KR	62.2 ± 2.3	66.0 ± 2.1	39.1 ± 3.4	88.4 ± 14.8	18.9 ± 2.8
RA	75.7 ± 3.0	80.1 ± 2.2	40.0 ± 2.8	94.7 ± 11.8	19.9 ± 2.0
KL	75.5 ± 2.6	78.8 ± 3.0	39.9 ± 3.3	89.7 ± 15.7	18.8 ± 2.6
BU	69.4 ± 1.8	73.7 ± 1.9	51.2 ± 3.1	157.7 ± 13.3	29.6 ± 3.0
PR	63.4 ± 2.7	70.7 ± 2.5	51.0 ± 3.1	154.4 ± 20.3	29.4 ± 3.3
GO	62.0 ± 1.7	68.8 ± 2.3	43.2 ± 2.5	116.9 ± 13.2	21.5 ± 2.4
FR	78.5 ± 2.0	82.8 ± 1.9	39.9 ± 1.7	96.9 ± 6.9	19.1 ± 1.3

表 17.4 RBF 网络的性能和特征——大脑数据

	C4.5r	TB-RBF	units	connections	trainable
A4	47.2 ± 4.0	57.6 ± 2.9	39.9 ± 1.7	54.5 ± 12.4	12.2 ± 3.3
B6	40.5 ± 5.3	49.6 ± 5.2	25.6 ± 2.9	96.9 ± 6.8	6.2 ± 2.1
B8	31.2 ± 6.6	47.5 ± 4.3	39.9 ± 1.7	96.6 ± 6.8	19.1 ± 1.3

仔细比较表 17.1 和 17.2 中的结果，可以发现尽管比较 TBNN 和 TB-RBF 效果上的差异不明显，但 TBNN 和 TB-RBF 均具有各自的优点。在两者之中，TB-RBF 用到的参数大大的节省。这不仅取决于基于发散功能的比较紧密的表达，而且取决于系统执行的方式。读者可以回想，不像 TBNN 要产生完全连接的网络，TB-RBF 系统在输入和隐藏层之间产生最小的连接数。

出于这个原因，表 17.3 和 17.4 同样给出单元平均数和连接 RBF 网络的神经元的平均数量。在分类阶段用 TB-RBF 产生网络是相当高效的。的确，一个全连接的 RBF 网络典型具有——通过相同数目的隐藏单元和 15 个属性的睡眠领域——40×15＝600 个连接，这些是远远大于 TB-RBF 的。在大脑领域这种现象更加显著：在实例 A4 中，完全的相互连接大约是 40×11＝440 个，而在 TB-RBF 中平均为 54 个连接；在实例 B8 中，完全的相互连接大约是 40×44＝1760 个，而在 TB-RBF 中平均约为 100 个连接。而且，由于本质上是基于发散功能的，所以输入层与隐藏层的连接的权值被设为 1。由于只有隐藏层与输出层之间的权值是通过训练得到的，这也使学习变得非常快（表中的最右列提供了训练参数的数目）。

17.6　讨论

研究者寻求实验证据，证明他们所偏爱的学习方法的优越性，但是通常发现每种学习器只适应独特的任务。神经网络好像在那些具有复杂的决策面并且训练数据充裕的领域比传统的规则生成技术要好。Weiss 和 Kapouleas(1989)观察到，只有当底层的概念能由少量的简单规则所描述时，符号方法会被随机初始化，然后使用反向传播算法得到的 MNN 提供更好的分类。类似的发现 Wnek 和 Michalski(1994)详细讨论过。不过，当 Fisher 和 McKusick（1989）以及 Fisher 等人(1989)实验更加复杂的概念时，他们意识到如果提供充足的训练例子而且训练时间充分长的话，MNN 可以和决策树相媲美（特别是在有噪声的领域）。

Atlas 等人(1990)报告在三个实际的领域中，MNN 明显比决策树要好。Ivanova，Kubat 和 Pfurtscheller(1994)报告的实验指出，对于异常复杂的概念，神经网络对初始的拓扑结构和权值特别敏感。尽管反向传播算法在随机初始化时易于陷入局部极小值，但是在当起始于恰当的初始状态时，TBNN 可以超

过决策树。

Ivanova 和 Kubat(1995)区分了对于一个给定任务决定一个神经网络拓扑结构的两个通用的方法。基于搜索的（Search-based）策略通过每次加入一个神经元逐步增长网络（Fahlman 和 Lebiere，1990；Frean，1990）或是通过删除神经元或连接来修剪大的网络（Mozer 和 Smolensky，1989；Le Cun，Denker 和 Solla，1990）。与此相反，启发式的（Informed）策略使用以产生式（Towell Shavlik 和 Noordewier，1990；Goodman 等人，1992；Bala，Michalski 和 Pachowicz，1994）或者决策树（Sethi，1990；Ivanova 和 Kubat，1995；Sahami，1995；Bioch，Carsouw 和 Potharst，1995）来表示的知识初始化神经网络。

本章介绍了一个使用决策树生成器初始化 MNN 和 RBF 网络的学习例子。在更快的训练后，神经网络可以和其他的学习器相媲美（在给定的领域中）。决策树归纳可以用来确定 MNN 中隐藏神经元的个数和初始化权值。在 RBF 的例子中，决策树确定径向基函数的个数、它们的相互连接和参数。基于树的初始化的贡献有两个方面：提高了未见实例的分类准确度；在 RBF 网络的例子中的一个紧凑的网络。在 TBNN 的例子中并没有研究第二个目标，因为前面 Ivanova 和 Kubat(1995)报告的结论表明了让 MNN 完全相互连接的优点。

本章介绍的两个学习系统可以在睡眠分类领域协助专家，它显著地减少了绘制脑电图的代价。TBNN 和 TB-RBF 在睡眠领域所达到的精度并不比人类专家差——Kemp 等人(1987)报告了在六个人类分类中一致性不超过 75%，这和我们自己的经验是相吻合的。学习系统达到了差不多相同的精度水平，而且通过后处理方法性能可以得到进一步的提高，尽管后处理方法超出了这篇论文的范围。人类专家证实 TBNN 和 RBF 产生的脑电图完全可用。

第二个领域，人脑-计算机接口会因为手头上有一个简单的工具来分类 EEG 信号而受益，其中，EEG 信号对于人脑所发出的肌肉命令是很重要的。在这里，用途是明显的，因为 EEG 的失调模式很难通过规则来描述，而学习看起来则是完成此任务的惟一途径。

致谢

本研究所使用的医疗数据属于奥地利 Graz 技术大学医疗信息系所有。睡眠分类数据的记录和分类是由“Fonds zur Förderung der wissenschaftlichen

Forschung,项目 S49/03 所支持。脑数据的记录和分类由奥地利的“Fonds zur Förderung der wissenschaftlichen Forschung”项目 P9043 和 P11208 所支持。

参考文献

L. Atlas, R. Cole, Y. Muthusamy, A. Lippman, J. Connor, D. Park, M. El-Sharkawi, and R.J. Marks(1990). A Performance Comparison of Trained Multilayer Perceptrons and Trained Classification Trees. Proceedings of the IEEE, 70, 1614-1619.

J.W. Bala, R.S. Michalski, and P.W. Pachowicz (1994). Progress on Vision through Learning at George Mason University. Proceedings of ARPA Image Understadning Workshop 191-207.

J.C. Bioch, R. Carsouw, and R. Potharst (1995). On the use of simple classifiers for the initialization of one-hidden-layer neural nets, Proceedings of the IEEE International Conference on Neural Networks(ICNN95), Vol.4, 1739-1743.

第 18 章　机器学习在河流水质的生物分类中的应用

Sašo Džeroski，Jasna Grbović 和 William J. Walley

摘要

我们将介绍机器学习的几种，特别是规则归纳，关于河流水质生物分类领域的应用。这些应用旨在减少目前应用的基于生物学指标的分类方法的主观性。基于生物指标从英国河流中的生物分类规则，可以从专家分类后的样本中归纳出来。通过分析从对斯洛文尼亚河流监控程序中获取的数据，以确定其物理和化学参数对该河中所选定生物体的影响。该方法能够被用来解决人们还不能完全了解的生物体的生态需求。这样就可以从斯洛文尼亚河流所获的物理和化学参数及生物指标的数据，归纳出可对斯洛文尼亚河流进行生物分类的规则。在所有三个例子中，有价值的知识都是从环境监视器来的数据和（或）专家对所取得样本的解释中抽取而来的。

18.1　简介

河流是人类最重要的淡水资源，它可用于许多方面，包括饮用水供应、农田灌溉、工业和城市用水的供应、工业和城市废弃物的处理、航海、渔业及人们休闲娱乐活动等（Friedrich 等人，1992）。河流水资源管理者们因此需要有关他们所掌握的水资源数量和质量的优质科学数据。地表水水质，包括河流在内，取决于它们的物理、化学和生物性质。后者受水中存在的生物体种类和密度（包括生物种群结构及其多样性）的影响。基于以上性质，地表水可分成几个等级（之一），这些级别表明该项水资源可被用做不同的用途。

众所周知，理化性质只能有限地描述出在特定点某一时间的水质情况，

而生物（生物活体）可作为水质在一段时期内水质连续的监视器（Cairns 等人，1968）。这就使得用生物的方法监测水质更重要（De Pauw 和 Hawkes，1993）。自 Kolkwitz 和 Marsson（1902）首次提出将生物体作为一种监测自然水质的方法之后，已经提出了很多种不同的方法，它们将生物学数据映射到离散的级别或连续的刻度（大致情况，可参见 Friedrich 等人，1992；De Pauw 和 Hawkes，1993；Grbović，1994）。大多数的这些方法使用的指标是生物体（生物指标），它们拥有已知的生态需求，且因对不同种类的污染有着相应的敏感/忍受而被选中。给定一个生物样本，该样本中出现的所有生物体指标的出现频度和密度，通常都可被结合起来，从而推断出能够反映样本采集地点水质的生物学指标。

一个著名的例子就是污水生物指标（Pantle 和 Buck，1955），它是建立在 Kolkwitz 和 Marsson（1902）污水生物系统上的，这一指标在许多中欧国家得到应用，包括德国和斯洛文尼亚。生物指标被分配到一个偏爱的污水生物区域 s，它定义了相应生物指标最频繁发现的水质级别，一个权重因子 g，它依赖于生物指标在不同水质中分布的情况（单个生物体可以出现在不同水质的水中）。生物指标发生频率 h（在样本中找到的生物体数量）也被考虑在内。给定采样点的污水生物指标 SI 的计算公式如下：

$$\mathrm{SI}=\frac{\sum_{i=1}^{n} s_i g_i h_i}{\sum_{i=1}^{n} g_i h_i}$$

其中，i 标定在样本中出现的所有生物指标的范围。

生物指标能在不同层次水平上被识别出来，如在种类层次或科属的层次。一科族、一种类或其他任何的分类层次，都被视为一种分类（多个分类）。在污水生物系统中，生物指标是在种类层次被识别的，这需要更多的样本处理工作，但却可以给出对水质更为精确的描述。科属层次的识别被用在生物监测工作会议标注中（ISO-BMWP，1979），简称为 BMWP，并由它衍生出每种同类平均指标（Average Score Per Taxon，ASPT）。两个指标目前都在英国使用。一个数字（指标），被指派给每一种生物指标。仅仅考虑这些生物指标存在或不存在。对所有在样本中出现的生物指标求和就可以得到 BMWP 值，该值除以出现的生物指标的数量就得到 APST 值。它不像污水生物指标那样包括动植物两类，BMWP 指标仅仅依赖于动物，更加明确地局限于水下大型无脊椎动物，也就是生活在河床的无脊椎动物。

以上介绍的这些生物指标所存在的主要问题就是它们的主观性（Walley，1993），也就是说，对单个生物指标而言，污水生物的区域偏好、权重因子，以及 BMWP 指标的赋值，都是由微生物学和生态学专家，或专门委员会委员来完成的，这些数字的确定依靠的是专家们对生物指标类别有关生态需求的知识，而这些知识并非总是完全的。这样，被指定的生物指标数值就并不准确，关于这一点，在最近一篇使用从微生物监视器得到的数据来重新评估 BMWP 指标的文章中被提到（Walley 和 Hawkes，1996）。通过基于求和平均和加权平均方法，将各个独立的生物指标结合起来，而不是采用完善的结合方法，这又增加额外一层的主观性。

前面陈述的生物指标的主观性并不能完全避免。水质有数以千计的度量特征，但这些特征并不与所有的用户都相关。因此，一个通用的水质指标本质上就是主观的。但其主观性应该最小化：仅当主观性不可避免时才将其引入方法中，也就是说，与分类目标有关。如上节所述，在中间层引入主观性能够而且必须使其最小化。本章将介绍应用机器学习方法向这个方向所做的努力。

我们首先应用机器学习方法，从英国河流中提取出且已被河流生态学专家分类的样本，归纳出其水质生物的分类规则。生态学专家对样本的分类绝对是一个主观的处理过程，但这是仅有的出现主观性的阶段：机器学习被用来识别出一套模仿专家行为的规则，而无需再对生物指标指定量值，以及通过加权平均合并这些量值的中间阶段。第 18.2 小节介绍的就是机器学习的这一应用。

利用一个对斯洛文尼亚河流进行监测的程序所得到的数据，我们归纳出了一组用以描述理化参数对选定生物指标活体的影响。这将有助于识别这些生物体的生态需求，从而有助于减少在指定污水生物区域偏好和权重因子上的主观性。从理化参数以及生物指标数据中，可以归纳出斯洛文尼亚河流的生物分类规则。第 18.3 节介绍了机器学习的这些应用。最后，第 18.4 节进行了总结，讨论并指出与此相关工作的链接作为本章的结束。

18.2 英国河流生物分类中的规则学习

可以利用以下方法获得一种适合于生物水质分类的方法。假如我们有一

组样本（它列出出现的生物指标及其数量）和它们正确的分类。随后我们就可以利用规则归纳方法来获得正确分类样本所需的知识。河流生态方面的专家或许可以提供正确的分类。本节将介绍该方法是如何得到英国河流的生态分类的方法的。

18.2.1　数据

这里考虑的数据是由 292 个水下大型无脊椎动物样本组成的，这些样本是英国国家河流委员会一个生物监测计划的一部分。它们源自英格兰特伦特河流域的上游地区，并以地点种类矩阵的形式给出，其中行对应样本，列表示 80 个不同的无脊椎动物科（或类，某种情况下）。对于每个样本，给出 80 个无脊椎动物科中每个出现的数量程度。

动物出现的数量程度被记做 0 至 6 之间的整数。0 表示在样本中未找到某科动物的个体，1 表示在样本中出现 1～2 个个体，2 表示出现 3～9 个个体，3 表示出现 10～49 个个体，4 表示出现 50～99 个，5 表示出现 100～999 个，6 表示出现的个体数超过 1000 个。矩阵中存在大量的 0（很稀疏的矩阵），因为在给定的任何样本中，绝大多数科的动物都是不存在的。在我们的实验中，所有科属的出现频度均被作为连续变量。

样本是由河流生态学家 H.A. Hawkes 来分类的，他现在是英国伯明翰的阿斯顿大学名誉审稿人。他曾经是生物监标工作协会（ISO-BMWP，1979）的成员，这个委员会是由环境署建立，旨在建立一套英国生物监测系统。他主管的下属专业委员会负责解释科属分数的分配，以试图反映每一科动物对整体水质的相对重要性和影响程度。换句话说，他是一个英国河流生物分类领域中公认的专家。

根据由无脊椎动物群所指示的有机物污染情况，样本被分成 5 个级别。这原本是一个基于贝叶斯判决规则的专家系统计划的一部分（Walley 等人，1992）。这 5 个等级被设计对应于目前在英国使用的 5 个化学级别（1a，1b，2，3 和 4），并被标为 B1a，B1b，B2，B3 和 B4，以表示其与化学等级的区别（Ruk 等人，1993）。B1a 级表示污染最少（最佳水质），B4 级代表最差的水质。在机器学习的术语中，每一科动物的出现频度都是一个“属性”，每个生物样本都是一个“实例”，被专家划分的等级被称为“类别”。

18.2.2 实验

我们提出两个要学习的问题：从原始的80个属性中预报水质等级；通过描述给定样本中的种群多样的附加属性来预测水质等级。对每一个问题，都使用了上面介绍的方法来产生一组规则。随后，这些规则被该领域的专家（H.A.Hawkes）检查和评估。它们的性能也可用分类正确性和信息内涵来度量。

由Džeroski等人（1993）修改过的CN2规则归纳系统（Clark和Niblett，1989；Clark和Boswell，1991）被用来从分类样本中归纳出规则。Džeroski等人（1993），修改了CN2以度量整套规则的相对信息得分情况（Kononenko和Bratko，1991），并在启发式搜索时使用m估计（Cestnik，1990）以代替拉普拉斯估计。相对信息分值考虑了类别中先验概率的差异，而m估计在从噪声实例中学习时，仍可出现更好的概率估计。

在我们的实验中，CN2被用来归纳整套的无序规则。这些规则必须有较高的重要性（在99%水平上）。如下所述，除了有效性和启发式搜索的设定外，CN2的其他参数都使用了默认值（可参阅Clark和Boswell，1991）。

为了选择合适的概率估计方法，也就是参数m的合适数值，我们使用了以下方法。参照早先实验情况（Cestnik，1990；Džeroski等人，1993）。我们首先尝试了参数m的15个不同取值（0，0.01，0.25，0.5，1，2，4，8，16，32，64，128，256，512和1 024）。同样也测试了拉氏估计的情况。对于一组给定样本，我们由此可以归纳出16套规则，并依次按照如下标准选择最优者：（1）信息分值；（2）准确性；（3）参数m值更小。根据训练样本估计准确性和信息分值。这一过程使我们可以选择出合适的正确层次：利用重要性阈值防止过度逼近。随着参数m的增加，在达到最大值之前，归纳规则的正确性和信息得分都在不断增加直到最优；之后便开始减少（Ličan-Milošević，1994）。值得一提的是，这种行为仅在使用了很高的重要性阈值时才出现。假如我们不使用重要性阈值，那么准确性和信息分值随着m的增加而减少：我们无法在训练集中选取合适的m值。

CN2系统（Džeroski等人，1993）从292个样本中归纳出12条规则，其中每个样本均含有80个属性，m参数的最佳取值是32。如上文所述，可将所使用重要性的阈值设为99%来确保归纳出规则的可信度。平均而言，每条规则覆盖了25个样本，并包含了5种情况。在训练集上，这些规则获得

了 86.3%的准确率以及 75%的信息内容。表 18.1 是这 12 条规则中的 3 条。

表 18.1　基于生物指标存在和出现频率所获得的英国河流生物分类规则

```
IF Hydrobiidae <= 3
AND Planorbidae <= 0
AND Gammaridae <= 5
AND Leuctridae > 0
THEN  Class = B1a
      [42 0 0 0 0]

IF Asellidae > 2
AND 0 < Gammaridae <= 4
AND Scirtidae <= 0
THEN Class = B2
     [0 0 41 0 0]

IF Planariidae <= 0
AND Tubificidae > 0
AND Lumbricidae <= 0
AND Glossiphoniidae <= 2
AND Asellidae > 0
AND Gammaridae <= 0
AND Veliidae <= 0
AND Hydropsychidae <= 0
AND Simulidae <= 0
AND Muscidae <= 0
THEN Class = B3
     [0 0 3 28 10]
```

表 18.1 的第一条规则用来预测等级为 B1a 的水质。它覆盖了等级为 B1a 的 42 个实例，且不包括其他 4 个等级的实例，即方括号中的数字。条件（IF）部分表明的是在样本中出现的 Hydrobiidae 生物为 50 个(≤3)，未出现 Planorbidae 生物(≤0)并出现少量 Leuctridae 生物。

表 18.1 的第二条规则预测的是水质 B2。它覆盖等级为 B2 的 41 个实例。它要求不能有 Scirtidae 生物，至少含有 50 个 Asellidae 生物及必须有 Gammaridae 生物且不可过多（不超过 100 个个体）。

最后，表 18.1 中的第三条规则预测的是水质等级 B3。应该注意到该规则覆盖了等级 B4 的 10 个实例和等级 B2 的 3 个实例，加上 B3 等级的 28 个实例。在归纳过程已考虑这个情况。将匹配所有分类样本的规则结合起来，该规则非常依赖这样几科生物的不存在：Planariidae，Lumbricidae，Gammaridae，Veliidae，Hydropsychidae，Simulidae 和 Muscidae。它需要有 Tubificidae 和 Asellidae 存在，且限制 Glossiphoniidae 的数量不得超过 10 个。

80 科属生物中有 35 个在这些归纳出的规则中被提到，并要求出现/不出现或给定其出现频度。规则归纳算法选择那些对水质等级最具有指示性的科

属，并参考其他科属动物出现的情况。与标准的学习过程相反，后者通常都是将不同生物指标通过权值结合在一起形成一个综合指标，而前者涉及特定种类之间的共同出现情况。

为利用现有专家知识来检验归纳出规则的一致性，12 个没有结论部分的规则被提交给了生态学专家。规则提交的顺序是随机的。随后，专家被要求为规则结论部分指明合适的等级。在大多数情况下，专家的结论都可以证实这些规则：有 5 条规则，他给出了与原结论相同的等级，有 3 条规则被指定在较差级别，有 3 条规则被指定在可能正确的等级和相邻错误级别的范围内，另有 1 条规则也许比实际归纳的还要好。对表 18.1 中的第一条规则，专家指定了 B1b 等级（实际归纳为 B1a 级），第二条指定了等级 B3（实际上为 B2），第三条规则指定为 B3 级（实际也为 B3 级）。

在这些规则大致与专家知识相当的同时，也存在一些缺陷，即规则所依赖样本中若干科属生物不存在（见表 18.1 中的最后一条规则）。某类生物不存在通常并不重要，在很多情况下（但并不是所有情况）几乎不提供更多信息。然而主要缺陷则是，这些规则仅仅使用了很少量的生物类别，而专家在分类时则考虑的是整个区域的所有生物。群体生物结构的多样性是水质的重要指示，专家不愿意说明某些规则，因为他觉得为了得到正确的结论，还需要更多的信息。

考虑到这些批评意见，我们设计了一个试验，在该试验中给学习系统附加 6 个属性，以获取样本的多样性。这些被称为附加 0（Morethan0），…附加 5（Morethan5）的属性，反映的是在一定出现频度上的生物科的数量。附加 0（Morethan0）是出现的生物科总数，附加 5（Morethan5）是在样本中至少出现了 1000 个个体的生物科目的数量。除了 m 取值为最优值 64，其他设定均如上所述。由此产生的 13 条规则，其正确率为 88.4%（在训练样本集上），信息内容为 80%。性能的改进说明了专家的批评是有道理的。

在表 18.2 中给出了利用附加属性的规则的例子。这条规则预报等级 B4（最差的水质），在样本中最多允许有 5 个不同的分类（Morethan0≤5），这些规则中至少有 2 条规则出现至少有 3 个生物个体（Morethan1>1），Dixidae 必须不出现，Asellidae 最多可以出现 50 个，Oligochaeta 最多可以出现 1000 个。这条规则覆盖了 B4 等级的 25 个样本和 B3 等级的 4 个样本。

表 18.2　利用生物多样性信息对英国河流生态分类的一条规则

```
IF Oligochaeta <= 5
AND Asellidae <=3
AND Dixidae <= 0
AND MoreThan0 <=5
AND MoreThan1 > 1
THEN Class = B4
     [0 0 0 4 25]
```

为了估计未知情况的性能，我们将把这组 292 个样本随机地分为含 195 个样本的训练集（三分之二）和含 97 个样本的测试集（三分之一）。首先，使用标准的方法从训练集产生一组规则，接着用测试集进行评估。对于初始学习问题（80 个属性值），m 的最佳值是 128（对测试集而言）。归纳出的 10 条规则对未知情况的估计有 61.9%的正确率，信息分值为 55%。相比而言，对训练集有 91.3%的正确率，信息分值为 80%。对于扩展的学习问题（86 个属性值），m 的最佳设定值是 32。这 12 条归纳出的规则对未知情况的估计有 68%的正确率和 56%的信息分值（对训练集可分别高达 93.8%和 83%）。从初始问题到扩展问题，其表现有着明显的提高，特别是对未知情况的分类正确性。

为了与更经典的分类方法做比较，我们对这两个分类问题使用了最近邻算法。最近邻法（NN）是最著名的分类算法之一。对它已进行了大量的研究工作，有关情况请参见 Dasarathy 的总结回顾（1990）。NN 算法将属性值视为欧氏空间的维，将样本视为这个空间中的点。在训练阶段，分类样本未经处理就被存储起来。当需要对一个新的样本进行分类时，就计算该样本与所有已知训练样本间的距离，并把与之相距最近的样本的类别赋予该新样本。

更通用的 *k*NN 算法使用 *k* 最近邻训练样本集并通过多数表决的方法来决定新样本的类别。在 *k*NN 的改进算法中，*k* 最近邻的每一次表决都是根据它到新样本的相邻程度来赋予权值的，利用“余一验证”方法可从训练集中自动获取 *k* 的最优值。最后，每一个属性对距离的贡献可以用属性和类别间的相互信息来赋予权值。这些所谓特征权值也能根据训练集自动获得。对我们的试验而言，采用的是 Wettschereck（1994）提出的 *k*NN 算法，该算法包含了上述的改进。

使用和 CN2 应用相同的 195 个训练样本和 97 个测试样本，不同的 kNN 算法的结果介绍如下：基本NN算法在没有多样性属性时可达55.7%的正确率，在有这一属性时可达 59.8%。没有特征权重的 *k*NN 方法在两个例子中都达到

了 55.7%的正确率，k 的最佳值是 3。最后，采用特征权重的 kNN 算法分别可获得 56.7%和 58.8%的正确率，k 的最佳值分别是 3 和 5。CN2 系统在两个例子中所归纳出的规则性能都有明显的提高，这意味着它们代表了对训练样本集的更有用的概括总结。

18.3 对斯洛文尼亚河流数据的分析

根据目前斯洛文尼亚的水质国家分类标准（OJSFRY，1978），若水适合饮用、洗浴和渔业，那么就属于第一级（最佳等级），第二等级的水适合渔业和休闲，包括洗浴，经过简单处理（凝固，过滤，消毒）之后，就能被用于工业，甚至是食品工业。第三级的水能用于灌溉工业（经条件变化），但不能用于食品工业。第四等级的水（最差等级）仅能在适当的处理后，用于比上述要求更低的用途。

斯洛文尼亚水资源管理局使用由 Pantle 和 Buck（1955）引入，Zelinka 和 Marvan（1961）改进的污水生物指标方法，将生物数据映射到连续的水质刻度上。从给定水样本中产生的污水生物指标是一个能够反映水质等级在 1 和 4 之间的单个数字。为表示方便，污水生物指标映射到 4 个基本等级和三个中间等级的离散质量刻度上，也就是这 7 个离散的等级：1，1-2，2，2-3，3，3-4 和 4。等级 1 对应于污水生物指标在 1.00 和 1.50 之间的清洁水质，而等级 4 代表污水生物指标在 3.51 和 4.00 之间的严重污染水质。等级 1-2 的水被称为柔和水质，而等级 2 为中等水质，等级 2-3 被称为较重污染水质。4 个基本等级与法定级别相对应，但略有不同，这是由于后者主要依赖于化学性质。

像第 18.1 节里说明的那样，污水生物指标是作为出现在给定生态样本中的生物指标污水生物区域偏好的加权平均值来计算的。这些权值是根据指示值和各个生物指标种类（或其他种群单元，通称为种属）的出现频率来计算的。这些生物指标的种属，诸如其生物特性，对水质的重要性及生态中角色都是已知的。从某种角度上说，它们反映了在一段时间内受理化指标影响的整个水质。然而，许多种属的生态角色和对水质的重要性都是未知的（因此不能用做生物指标）并还有可能随国家的不同而不相同（Grbović，1994）。水的理化性质对许多种属的影响，人们知道得也很少。从生态学和水质的观点来看，这些都是重要的研究主题。

有关斯洛文尼亚河流的数据来自该国水文气象监测机构（斯洛文尼亚共和国水文气象局，简称 HMZ），该机构对斯境内大多数河流的水质进行监测，并拥有一个水质样本的数据库。HMZ 提供的数据覆盖了 4 年时间，从 1990 到 1993 年。生物样本每年采样两次，一次在夏天，一次在冬天；而每个采样点理化指标的分析一年内要进行数次。理化样本包括对 50 个不同参数的测定，其中有如溶解氧气和硬度一类的指标，而生物样本则有一个所有在样本点出现的生物种类及其种群密度的列表。每一个种生物出现的频率（种群密度）都被生物学专家记录为定量的 3 个等级：1 级表示该种生物为偶然出现；3 代表时有出现；5 代表大量出现。生物样本还包括相应的污水生物指标及该指标所对应的水质等级。总计有 698 个既有理化分析，又有生物分析的可用样本，我们的试验就使用了这些样本。

给定了上述数据，我们就可应用 CN2 规则归纳系统（Clark 和 Boswell，1991；Džeroski 等人，1993）。我们对几个学习问题进行了形式化：分析所选定水的理化性质对所选定生物类别存在性的影响；在选定的一组生物指标中进行水质分类；根据所选定的一组理化属性来对水质进行分类。下面，我们简要说明学习和评价规则的方法，然后，给出每个学习问题的结果。对归纳出规则的评估包含了 Jasna 的注释（Jasna 是 HMZ 负责分析生物样本的专家，并且具有斯洛文尼亚河流中存在的动植物方面的生态学知识），分类的正确性及规则的信息分值（用以估计未知情况）。

CN2 系统与第 18.2.2 节中所述那样进行应用，以归纳出一组无序的规则。重要性指标下限设为 99%，其他参数均设为默认值。在给定的训练集上，试用了拉氏估计和 m 取 15 个不同数值的 m 估计，共产生 16 套归纳规则集，然后依次根据以下准则选择其中的最优者（1）信息分值，（2）正确率，（3）参数值 m 是否更小，这里，信息分值和正确率是根据训练集计算出来的。

对每一个学习问题，都要完成两套试验。第一套试验从所有的 698 个样本中归纳规则，旨在找出尽可能多的可靠模式（和期望的知识）。生物专家利用已知河流生态知识和水质标准来检查和评估这样获得的规则。第二套试验旨在评价归纳出的规则在未知情况中的正确率和信息分值。为此，我们用 10 种不同方式将整个数据库分为一个训练集（占 70%，489 个样本）和一个测试集（占 30%，209 个样本）。对于每一个划分，我们都用前面介绍的方法从测试集中归纳出一组规则，然后在测试集上检查这些规则的性能。在第 18.3.1

和 18.3.2 两节中介绍这 10 种不同划分的平均结果。

18.3.1 理化参数对选定生物体的影响

最能够影响植物的理化参数（水的属性）是：总固体悬浮物、氮化合物（NO_2，NO_3，NH_4）、磷化合物（PO_4）、硅族化合物（SiO_2）、铁（Fe）、表面活性剂（清洁剂）、化学氧需求（COD）及生化氧需求（BOD）。最后两项参数指示有机污染的程度：第一项表示可降解有机物总的数量；第二项显示可降解生物的数量。影响动物的是多个不同参数：水温、酸碱度（pH 值）、水中溶解的氧（O_2，饱和态 O_2）、总固体悬浮物、化学和生化氧需求（COD，BOD）。

本节提出的试验研究了上述理化参数对 10 种植物和 7 种动物的影响。在植物方面，研究了 8 种矽藻（Bacillariophyta）和 2 种绿色海藻（Chlorophyta）。选择研究的动物包括蠕虫类（Oligochaeta）、甲壳类（Amphipoda）和 5 种昆虫类。图 18.1 和 18.2 中就是这些生物种类的典型代表（4 种植物和 4 种动物）。

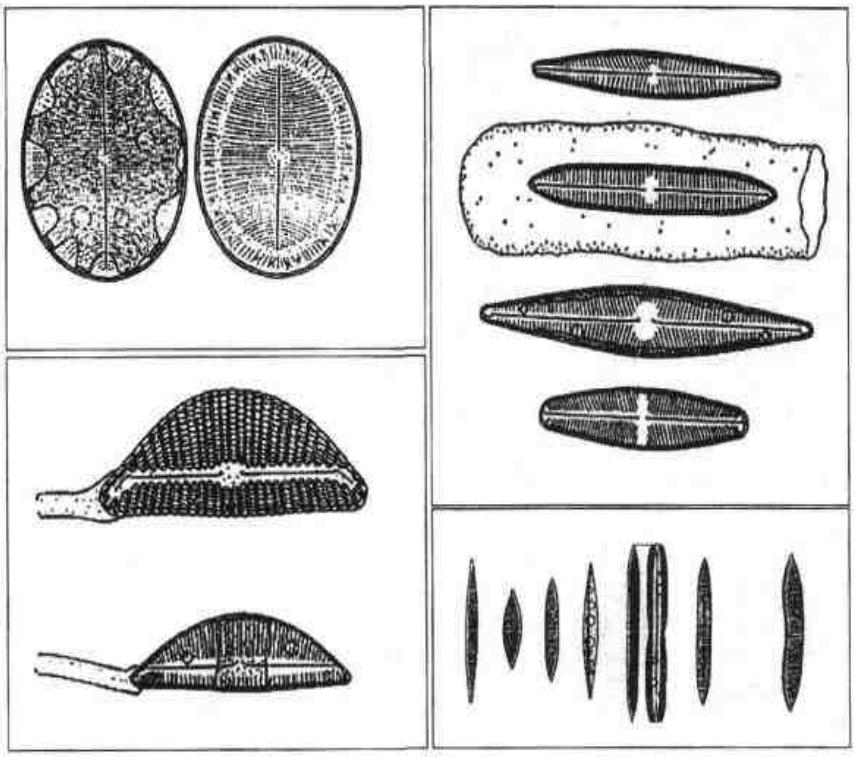


图 18.1 四种植物种类的代表个体，从左上开始顺时针为：
Cocconeis placentula，Navicula cryptocephala，Nitzschia palea 和 Cymbella sp.

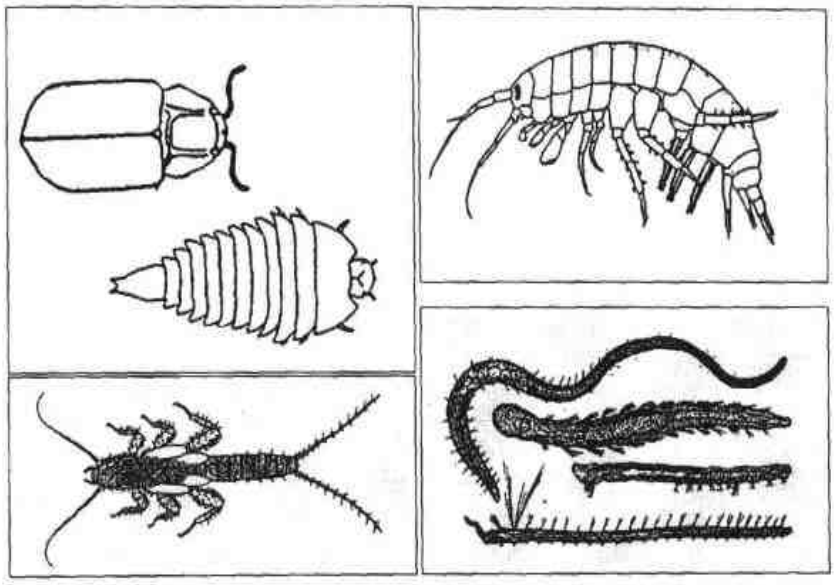


图 18.2　四种动物种类的代表个体，从左上开始顺时针为：Elmis sp.(成虫和幼虫)，Gammarus fossarum，Tubifex sp.和 Plecoptera leuctra sp.

对每一选定的生物类，我们都定义了基于属性的学习问题，属性就是那些选定的理化参数（植物的固体悬浮物，NO_2，NO_3，NH_4，PO_4，SiO_2，Fe，表面活性剂，COD，BOD，动物的水温，pH 值，O_2，溶解度，COD，BOD）。类别则是所选生物的存在与否（即值 Present 和 Absent）。因此就定义了 17 个不同的学习问题（域）。

现在，我们用上面的学习问题来简要说明整个试验，试验应用了前面介绍的特定方法。我们先总观这些归纳出规则的表现，既有从整个数据集（作为训练集）中归纳出的规则，也有从 10 个划分后数据集（仅计算其中测试集并取 10 个划分集合的平均值）中归纳出的规则。之后，再引述专家对选定动植物规则的评价。对选定生物种类、试验、归纳规则的性能及所选定规则集和专家对这些规则评价等的详细内容可以参阅 Ličan-Milošević 的学士学位论文（1994）。

对整个（训练）数据集的正确率在 66%和 85%之间（主要类别的频度在 50%和 70%之间），而信息分值在 23%到 50%之间。对于不同生物种类的规则集包含 10 到 20 条规则，每条规则的平均长度小于 5 个条件，每条规则平均覆盖 15 到 45 个样本。

由于规则的性能（正确率）不尽如人意，我们心里明白利用高的重要性阈值来防止过度逼近。更为重要的是，在一特定时刻的理化参数并不能完全

决定某一类生物的存在（或不存在），还取决于某一段时间上的理化参数，取决于生物的生存时间、水位和河床。使问题更加困难的是，一些种群混有非常不同的有机体，比如 Chironomidae（绿藻类），它在整个数据集上得到了最低的信息分值（1.4%）。

从 70%的数据集上归纳出的规则的信息分值（以剩下的 30%来度量）对所有生物种类来讲都要低得多，但仍为正值。这意味着规则包含了理化参数对生物种类的存在的影响的有用信息。尽管如此，17 种生物中有 5 种的正确率仍比默认值情况下要低。在本节剩下部分，我们将摘录专家对有关矽藻类植物（Nitzschia palea）和水生甲壳虫（Elmis sp.）规则的评价，这两种生物得到了最高的信息分值，对未知样本分别为 29.2%和 26.1%（10 个 70%训练—30%测试划分的平均值）及正确率 69.7%和 71.0%（默认值为 60.8%和 64.6%）。

矽藻类植物（Nitzschia palea）在 698 个样本中的 420 个里存在，是斯国河流中最常见的物种。它对污染有着很强的承受力，它生活在一个很大的水域范围内。它是水质等级 2-3（表面污水生物区域偏好）的特征：被用做较重污染水质的生物指标。

从整个数据集上建立起来的规则证实了较大程度的污染对这种生物有利。从它的 18 条规则中，在表 18.3 中列出了包含范围最大的前 6 条。这些规则显示了 Nitzschia palea 需要氮化合物、磷酸盐、矽土元素以及大量的可降解物（COD，BOD）。方括号中的数字是每条规则在每个等级中所覆盖的样本数。例如，[58 0]代表相应的规则包含了 58 个类别为“存在”的样本，[0 39]表示相应的规则包含了 39 个类别为“不存在”样本。

表 18.3　根据理化参数和所选规则预测 Nitzschia palea 种类

```
IF  PO4 > 0.065
AND Fe < 0.595
AND COD > 25.5
THEN Taxon = Present
     [58 0]

IF  4.25 < NO3 < 12.35
AND SiO2 > 1.65
AND Detergents > 0.055
THEN Taxon = Present
     [50 0]

IF  NO3 < 5.95
AND SiO2 > 4.75
AND COD > 7.95
AND 1.3 < BOD < 42.05
THEN Taxon = Present
     [59 0]

IF  NO3 > 1.3
AND NH4 < 0.97
AND 13.25 < COD < 16.35
THEN Taxon = Present
     [36 0]

IF  Hardness > 11.85
AND NO2 > 0.095
AND NH4 > 0.09
THEN Taxon = Present
     [82 0]

IF  NO2 < 0.005
AND NO3 < 7.1
AND PO4 < 0.125
AND Detergents < 0.055
AND BOD < 2
THEN Taxon = Absent
     [0 39]
```

属于 Elmis sp.纲的甲壳类动物 Coleoptera，在陆上很常见但在水中却很少。根据文献和专家的经验，我们知道这类生物居住在清洁的水中：它被视为水质等级 1-2 的标志。

在归纳出的 17 条规则里，表 18.4 列出了专家选择的 5 条规则，同上面介绍类似，方括号中的数字是每条规则在每个等级中所覆盖的样本数。例如，[36 0]代表相应的规则包含了 36 个类别为“存在”的样本，而[0 72]表示相应的规则包含了 72 个类别为“不存在”的样本。第一条规则要求数量相对较少的可降解物（污染），以利于 Elmis sp.的生存；这一数量在水温增加时甚至要求更低（见第二和第三条规则）。最后两条规则预测，在 BOD、COD 和 pH 值很高的过度污染的水中，该生物不能存在。这些规则确认了 Elmis sp.作为轻微污染水生物指标的理由。

表 18.4　根据理化参数和所选规则预测昆虫 Elmis sp.种类

```
IF  O2 < 11.45
AND Hardness > 10.35
AND COD > 2.35
AND BOD < 1.25
THEN Taxon = Present
     [36 0]

IF  Temperature > 11.75
AND 12.3 < Hardness <14.3
AND BOD < 1.75
THEN Taxon = Present
     [14 0]

IF  Temperature > 12.75
AND BOD < 0.65
THEN Taxon = Present
     [8 0]

IF  23 < COD < 46.45
THEN Taxon = Absent
     [0 72]

IF  PH > 7.05
AND BOD > 12.15
THEN Taxon = Absent
     [0 47]
```

并不是所有归纳出的规则都和已知的专家知识完全吻合。比如，让我们来考虑表 18.5 中所列的预测 Plecoptera leuctra sp.类生物存在的两条规则。第一条规则预言该种生物可以在清洁的水中（低 COD 值）找到，这和已知该生物的生态要求是一致的：Plecoptera leuctra sp.被用做清洁水质的生物指标。但第二条规则显示，这种生物可以在被污染得相当严重的水中存在（高 COD 值），提供足够氧（高饱和度）。该规则对现有专家知识进行重要的补充，因为绝大多数国境内的河流都是激流，因此均是高饱和的。

表 18.5　根据理化参数，两个预测 Plecoptera leuctra sp.存在的规则

```
IF  Temperature > 10
AND Saturation > 102.5
AND COD < 2.35
THEN Taxon = Present
     [14 0]

IF  Temperature < 23
AND 120 < Saturation < 150
AND COD > 10.9
AND BOD < 3.75
THEN Taxon = Present
     [8 0]
```

另一个例子是预测 Cymbella sp.类生物存在的几条规则，如表 18.6 所示。这些规则指出这种生物能够在中度污染到污染较重的水中存在（如规则中大量可降解物，即较高的 BOD 和 COD 值所示）。但在实际的水质监测中，Cymbella sp.被用做轻微污染的生物指标。

表 18.6 根据理化参数，两个预测 Cymbella sp.存在的规则

IF Hardness < 13.2 AND NO3 < 6.75 AND 0.13 < NH4 < 0.25 AND PO4 < 0.05 AND Detergents < 0.02 AND COD < 5.25 THEN Taxon = Present [29 2]	IF Hardness < 12.6 AND NO2 < 0.2 AND NO3 > 6.45 AND 0.02 < PO4 < 0.5 AND SiO2 < 1.95 AND BOD < 9.95 THEN Taxon = Present [17 0]

18.3.2. 生物分类

这一部分将描述预测斯洛文尼亚河流生物等级的试验，这种等级是根据污水生物指标（来自两套不同的属性集）所确定。第一套包含上一节中提到的所有理化参数，共 13 个。第二套包含前一节中的 17 个生物类种以及 10 种附加的生物，共 27 种。13 个参数是取实际值的属性，27 个种类给出的是 4 值的离散属性值（线性顺序）：0，1，3，5。如前所述，水质有 7 个等级。最多样本的等级是 2，它包含 698 个样本中的 339 个样本。因此，默认值准确率为 48.6%。

为说明清楚，我们观察表 18.7 中的两条规则。它们根据理化参数预测水质等级，覆盖所处理样本集的最大部分。第一条规则覆盖等级 1 中的 9 个样本，等级 1-2 中的 80 个样本及等级 2 的两个样本。第二条规则覆盖等级 1-2 中的 152 个样本及等级 2-3 的 5 个样本。

在我们需要有关水质的化学和生物方面技能来全面预测这些规则时，它们是直观的和可理解的。等级 1-2（第一条规则）需要相对冷的水和非常少的污染物（NO_2、NO_3、清洁剂、COD、BOD）。等级 2 的水（第二条规则）通常较暖，有时容许包含较多污染物，只要有足够的氧（饱和度>57.3）。

表 18.7　根据理化参数，两个预测水质的规则

```
IF  Temperature < 14.35
AND PH < 8.45
AND NO2 < 0.235
AND 1.75 < NO3 < 7.15
AND Detergents < 0.025
AND COD < 4.25
AND BOD < 2.35
THEN QualityClass = 1-2
     [9 80 2 0 0 0 0]

IF Temperature > 12.65
AND PH < 8.65
AND Saturation > 57.3
AND NO2 < 0.375
AND NH4 > 0.065
AND PO4 < 0.39
AND SiO2 < 10.75
AND COD > 2.65
AND 1.25 < BOD < 4.75
THEN QualityClass = 2
     [0 8 152 5 0 0 0]
```

从整个数据集中所归纳出的规则获得 81.5%的分类准确率，其中利用了理化参数。而利用生物指标达到 71.1%，信息分值分别为 62%和 44%。在学习 70%的数据集时，相应测试集的准确率分别为 60%和 58%，信息分值分别是 32%和 28%。有趣的是，依据理化参数进行预测可以得到更好的性能，尽管生物水质是可以预测的。然而要确定水质等级，需要用到比我们试验中用到的更多的生物指标。

最后我们再看看表 18.8 中的四条规则，它们根据 27 个生物指标预测水质等级。第一条规则覆盖了等级 1 中的 1 个样本，等级 1-2 中的 16 个样本，等级 2 中的 1 个样本。第二条规则覆盖了等级 1 中的 2 个样本，等级 1-2 中的 32 个样本，等级 2 中的 2 个样本。第三条规则覆盖了等级 1-2 中的 3 个样本，等级 2 中的 32 个样本，等级 2-3 中的 9 个样本，等级 3 中的 2 个样本，以及等级 4 中的 1 个样本。最后一条规则覆盖了等级 2 和等级 2-3 中各 1 个样本，等级 3 中的 16 个样本以及等级 3-4 中的 2 个样本。

表 18.8 根据所选生物指标出现频率，预测水质等级的四条规则

```
IF  BACILLARIOPHYTA_Navicula_cryptocephala = 0
AND CHLOROPHYTA_Scenedesmus_obliquus = 0
AND DIPTERA_Chironomidae_green = 3
AND COLEOPTERA_Elmis_sp. = 3
THEN QualityClass = 1-2
     [1 16 1 0 0 0 0]

IF  BACILLARIOPHYTA_Nitzschia_palea = 0
AND CHLOROPHYTA_Oedogonium_sp. = 0
AND OLIGOCHAETA_Tubifex_sp. = 0
AND AMPHIPODA_Gammarus_fossarum = 5
THEN QualityClass = 1-2
     [2 32 2 0 0 0 0]

IF  BACILLARIOPHYTA_Navicula_cryptocephala = 1
AND BACILLARIOPHYTA_Nitzschia_palea = 1
THEN QualityClass = 2
     [0 3 32 9 2 0 1]

IF  BACILLARIOPHYTA_Cocconeis_placentula = 0
AND BACILLARIOPHYTA_Cymbella_ventricosa = 0
AND CHLOROPHYTA_Cladophora_sp. = 0
AND TURBELLARIA_Dendrocoelum_lacteum = 0
AND OLIGOCHAETA_Tubifex_sp. = 5
AND DIPTERA_Simulium_sp. = 0
THEN QualityClass = 3
     [0 0 1 1 16 2 0]
```

第一条规则预测当 Elmis sp.和 Chironomidae（绿藻类）经常发生（3），并且物种 Scenedesmus obliquus 和 Navicula cryptocephala 不存在（0）时，等级为 1~2。这与专家知识是一致的，即 Elmis sp.和 Chironomidae（绿色）是清洁水的指标，而 Navicula cryptocephala 是污染水（等级 3）的指标。第二条规则预测等级 1~2。注意规则要求 Tubifex sp.不存在，这是一个严重污染水的指标。事实上，六条规则中的预测等级 1~2 的四条规则明确要求 Tubifex sp.不存在。第三条规则预测当 Navicula cryptocephala 和 Nitzschia palea 偶尔出现（1）。若它们大量发生，则两个物种均是严重污染的指标，而由于他们偶尔出现时，所以与专家知识保持一致。最后一条规预测等级 3，它要求 Tubifex sp.大量存在，几个指示清洁（Cocconeis placentula）到中等污染（Simulium sp.）水质的物种不存在。

18.4　讨论

我们已经利用规则归纳来解决若干有关水质生物分类问题。这包括从专家分类样本和水质数据分析综合出分类规则。这里有关英国和斯洛文尼亚河流的数据是有效的。

对于英国河流而言，可以获得专家对生物样本的分类情况。所归纳出的规则成功表示了专家的知识。专家也发现规则基本与其知识保持一致。种群多样性的信息更增加了信息内容和归纳出规则的质量。我们利用未知数据对归纳出的规则性能进行评估，并表明它们比 *k*NN 分类算法表现得更好，为论证规则集是否较好概括了训练样本集合的有效归纳提供了依据。所归纳出的规则可以用于完成按照水质自动解释生物样本的任务，也就是说，在给定相应地点无脊椎动物群体结构的情况下，预测一个特定地点的水质。

关于斯洛文尼亚河流，我们所完成的试验表明规则归纳可用于分析水质数据并发现不同类型的知识。我们归纳出的规则，可以描述斯洛文尼亚河流中水的物理化学性质对所选生物活体存活的影响，这些生物活体目前是河流水质的生物指标。专家对这些规则的评估表明它们的确表达了有用的知识，正如它们的正信息分值所指示的那样。在某些情况下，规则仅仅证实了生物专家有关生物指标的物种知识。而其他情况下，它们揭示了研究物种生物方面的新内容，这就扩展了现有的知识，而没有矛盾。甚至有时候，规则可以指出物种指标指示错了水质等级。

在以上有关 17 个物种（其中相对有名的生物）的分析被日常用于水质指标时，我们仍将有能力发现一些有关所研究的物种及其生物指标角色的新知识。这就意味着利用归纳能够完善有关河床生物和生态的知识。因此今后工作的一个有前景的方向就是将分析拓展到目前知道较少并没有作为生物指标的物种方面。这将会为生物学和水质监测贡献新的知识，因为某些新发现的物种可能是非常好的生物指标。另一种可能就是归纳出关于理化测量值的时序数据的生物指标出现，因为生物指标反映了一段时间的水质。类似归纳逻辑编程（ILP）技术（Lavrač 和 Džeroski，1994）也可以用于这个方面。

我们还归纳出可预测斯洛文尼亚河流中水质等级的规则（如同污水生物指标提供的那样）。为此目的，规则利用生物指标数据基本都与现有的专家知识一致：这是可以理解的，因为是用生物指标数据（虽然是更大规模的指标）

来推导污水生物指标的。根据水的理化性质预测生物水质等级的规则有着令人吃惊的准确率和信息量，值得更进一步的分析，由熟悉水质方面的生物和化学专家来进行。更一步工作的一个主题就是归纳出利用理化数据和生物指标的规则。在一定程度上，两者相互补充。

这里所介绍的应用旨在减少当前所采用的对水质分类生物方法的主观性。英国河流生物水质分类中，我们的方法仅在一点引入了主观性，专家对跨区域代表性样本进行分类以提供一组样本。然后我们利用规则归纳将样本直接映射到分类，不再引入（或非常少）主观性。对于斯洛文尼亚河流生物分类，在样本分类中通过使用污水生物指标，也引入了主观性。正如先前所提到的，目标分类的主观性是无法避免的。分析理化参数对斯洛文尼亚河流中所选生物体的影响将有助于了解这些生物体的生物特征，更加客观地给污水生物区域的偏好和指标赋值。在这两种情况中，规则归纳在水质分类中是一个更好、更客观的方法。

在相关的工作中，有一些应用不同统计或机器学习方法来解决英国河流分类问题。这其中的工作包括 Walley 等人（1992）和 Ruck 等人（1993）。Walley 等人利用贝叶斯推理，而 Ruck 等人则利用了神经网络来完成河流水质的生物分类。本工作的扩展就是比较几种方法，也就是说，贝叶斯，神经网络和机器学习方法（Walley 和 Džeroski，1995），这也包括与传统分类方法，诸如 ASPT（每个物种平均分数），进行比较。然而，这些研究将水质作为连续变量，因此无法与目前研究直接比较。ILP（Lavrač 和 Džeroski，1994）也被应用于同样的数据，但并没有与其他方法，在准确性和/或信息分值方面进行比较。但是专家对归纳出规则的评估表明 ILP 系统表示中的倾向性对于这类应用非常有利。

在更广一些背景下，Kompare 等人（1994），Kompare 和 Džeroski（1995），Križman 等人（1995）以及 Karalič 和 Bratko（1996）应用机器学习技术来解决海藻在湖和礁湖中生长的建模问题。值得一提的是，回归树（Breiman 等人，1984）和机器发现技术（Langley 和 Zytkow，1989）被用于威尼斯礁湖中海藻生长建模问题，ILP 也被用于对生长在 Bled 湖中的海藻建模（Karalič 和 Bratko，1996）。最有名就是从测量数据归纳出差分方程模型（Kompare 和 Džeroski，1995；Križman 等人，1995），它可以成功地预测在威尼斯礁湖中海藻 Ulva rigida 的顶峰和谷低。

作为结束语，我们已经介绍了规则归纳在生物水质分类领域中的若干应用。所产生的规则是透明的，并容易被专家所理解。在解决的所有问题中，所归纳出的规则包含有关所研究领域有价值的知识。一些情况下，这些知识拓展和补充了专家现有的知识。尽管需要对性能进行更为全面的评估，我们还是可以说机器学习技术，在河流水质领域和其他生态领域的分类和数据分析方面，仍然是一个有用的工具。

致谢

本章部分内容最初由 Džeroski 等人（1994），Džeroski 和 Grbovic（1995）和 Džeroski 等人（1997）发表过。作者感谢 H.A.Hawkes，他为生物样本分类提供了宝贵的帮助，并对所产生的规则给出专家的评价。NRA（Severn-Trent 地区）提供了英国河流的生物数据，斯洛文尼亚的水文气象学院提供了有关斯洛文尼亚河流的生物、物理和化学数据。还要感谢 Doris Ličan-Milošvić，作为其学士学位工作的一部分她完成了利用斯洛文尼亚河流数据的试验，她就读于斯洛文尼亚的 Ljubljana 大学，计算机科学和电气工程系。她的论文是由 Ivan Bratko，Sašo Džeroski 和 Jasna Grbović 教授们指导的。Sašo Džeroski 感谢斯洛文尼亚科学技术部（MZT）以及欧洲研究委员会信息与数学分会（ERCIM）所提供的经济资助。

参考文献

[1] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). Classification and Regression Trees. Wadsworth, Belmont.

[2] Cairns, J., Douglas, W.A., Busey, F., and Chaney, M.D. (1968). The sequential comparison index-a simplified method for non-biologists to estimate relative differences in biological diversities in stream pollution studies. J. Wat. Pollut. Control Fed., 40: 1607-1613.

[3] Cestnik, B. (1990). Estimating probabilities: A crucial task in machine learning. In Proc, Ninth European Conference on Artificial Intelligence, pages 147-149. Pitman, London.

[4] dark, P. and Boswell, R. (1991). Rule induction with CN2: Some recent improvements. In Proc. Fifth European Working Session on Learning, pages 151-163. Springer, Berlin.

[5] clark, P. and Niblett, T. (1989). The CN2 induction algorithm. Machine Learning, 3(4): 261-283.

[6] Dasarathy, B.V., editor. Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos, CA.

[7] De Pauw, N. and Hawkes, H.A. (1993). Biological monitoring of river water quality. In Proc.Freshwater Europe Symposium on River Water Quality Monitoring and Control, pages 87-111.Aston University, Birmingham.

[8] Dzeroski, S. (1996). Inductive logic programming and knowledge discovery in databases. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, pages 118-152. MIT Press, Cambridge, MA.

[9] Dzeroski, S., Cestnik, B., and Petrovski, I. (1993). Using the m-estimate in rule induction. Journal of Computing and Information Technology, 1: 37-46.

[10] Dzeroski, S., Dehaspe, L., Ruck, B., and Walley, W.J. (1994). Classification of river water quality data using machine learning. In P. Zannetti, editor. Computer Techniques in Environmental Studies V(Proc. Fifth International Conference on the Development and Application of Computer Techniques to Environmental Studies), Vol. I: Pollution modelling, pages 129-137, Computational Mechanics Publications, Southampton.

[11] Dzeroski, S., and Grbovic, J. (1995). Knowledge discovery in a water quality database. In Proc.First International Conference on Knowledge Discovery and Data Mining, pages 81-86. AAAI Press,Menlo Park, CA.

[12] Dzeroski, S., Grbovic, J., Walley, W.J., Kompare, B. (1997). Using machine learning techniques in the construction of models. II. Data analysis with rule induction. Ecological Modelling, 95: 95-111.

[13] Friedrich, G., Chapman, D., and Beim, A. (1992). The use of biological material. In Chapman, D.,editor. Water Quality Assessments, pages 171-238. Chapman and Hall, London.

[14] Grbovic, J. (1994). Applicability of Various Procedures for the Assessment of Quality of Torrential Streams. PhD Thesis, Biotechnical Faculty, University of Ljubljana, Slovenia. In Slovenian.

[14] Grbovic, J. (1994). Applicability of Various Procedures for the Assessment of Quality of Torrential Streams. PhD Thesis, Biotechnical Faculty, University of Ljubljana, Slovenia. In Slovenian.