Rough集理论与知识获取

王国胤 编著

西安交通大学出版社

Rough 集理论与知识获取

王国胤　编著

西安交通大学出版社

图书在版编目(CIP)数据

Rough 集理论与知识获取 / 王国胤编著. --西安:西安交通大学出版社,2001.5
ISBN 7-5605-1409-X

Ⅰ. R… Ⅱ. 王… Ⅲ. 计算机应用-信息处理-方法
Ⅳ. TP391

中国版本图书馆 CIP 数据核字(2001)第 022116 号

*

西安交通大学出版社出版发行
(西安市兴庆南路 25 号 邮政编码:710049 电话:(029)2668316)
长安县第二印刷厂印装
各地新华书店经销

*

开本:850mm×1 168mm 1/32 印张:7.5 字数:184 千字
2001 年 5 月第 1 版 2001 年 5 月第 1 次印刷
印数:0 001~1 000 定价:15.00 元(平) 20.00 元(精)

作者简介

王国胤，男，1970 年 3 月生，重庆市人，汉族，工学博士，重庆邮电学院特聘学者、计算机科学与技术学院副院长、计算机科学与技术研究所所长、教授。1992 年毕业于西安交通大学计算机软件专业，获工学学士学位（陕西省优秀大学毕业生、西安交通大学优秀大学毕业生）；1994 年毕业于西安交通大学计算机软件专业，获工学硕士学位；1996 年毕业于西安交通大学计算机组织与系统结构专业，获工学博士学位（西安交通大学优秀博士毕业生）；1998 年至 1999 年以访问学者身份工作于美国 University of North Texas 计算机科学系（1 年）；1999 年以访问学者身份工作于加拿大 University of Regina 计算机科学系（2 个月）。现为 IEEE 会员，国际科学技术开发协会（IASTED International Association of Science and Technology for Development）信息学技术委员会委员，重庆市科技进步奖专业评审委员会委员。1992 年以来，一直从事智能信息系统的理论及应用研究，在 Rough 集理论、并行神经网络体系结构、逻辑神经网络、神经网络自动知识获取、集成智能系统、数据压缩、网络信息系统等领域开展了研究。作为主研人员参与完成了 2 项国家自然科学基金项目和 1 项高技术产品研制开发项目，作为项目负责人主持完成了 5 项省、部级科研项目，现正主持国家自然科学基金、国家 863 计划、攀登计划、高等学校骨干教师资助计划、教育部回国留学人员科研启动基金和重庆市应用基础研究基金等共 6 项科研项目的研究工作，先后多次赴香港、美国、加拿大、日本等地出席国际学术会议和进行学术访问，在国内、外主要学术刊物和学术会议上发表了 40 余篇学术论文，其中多篇被国际权威检索刊物《工程索引》（SCI），《科学引文索引》（EI），《国际科技会议索引》（ISTP），《英国科学文摘》（INSPEC）等收录。

内容简介

Rough 集理论是一种研究不完整、不确定知识和数据的表达、学习、归纳的理论方法，近年来在理论模型、算法研究、工程应用中取得了好的成果和应用。本书重点在于阐述 Rough 集理论的模型、算法以及基于 Rough 集理论的知识获取技术。全书共分 11 章。第 1 章介绍了集合论基础；第 2 章介绍了信息表知识表达系统；第 3 章介绍了 Rough 集基础理论；第 4 章介绍了知识获取的基本问题；第 5 章介绍了知识系统不确定性的表示与处理问题；第 6 章介绍了数据预处理技术；第 7 章介绍了信息表属性约简的理论与算法；第 8 章介绍了信息表值约简的理论与算法；第 9 章介绍了逻辑推理方法；第 10 章介绍了几个典型的 Rough 集工程实例；第 11 章介绍了几个 Rough 集演示软件系统。

本书的目的就是要向计算机学科、人工智能学科、智能信息处理学科、机器学习学科、自动化学科等研究领域的研究人员系统介绍 Rough 集理论这一新的理论工具及其应用技术。本书可以作为计算机、自动化等专业高年级本科生、硕士生和博士生的学习参考用书，同时对相关学科领域的科技工作者和工程技术人员也有重要的使用和参考价值。

前　言

智能信息处理是当前信息科学理论和应用研究中的一个热点领域，随着过去几十年中人们在专家系统、知识工程、人工神经网络、模糊集合等众多领域的不断实践和探索，取得了很多很好的成绩。随着信息时代的到来，信息量不断增长，对信息分析工具的要求也越来越高，人们希望自动地从数据中获取其潜在的依赖模型。这样，大量的数据就无须人的处理，甚至无须人的观察。因此，研究能够从大量信息中形成实际概括（归纳）的系统就显得越来越重要。虽然已经有很多对数据进行分析的简单统计技术，但高级的智能数据分析技术还远没有成熟。因此，数据信息的产生和对它的理解之间的差距越来越大。

Rough 集（Rough Sets，有的也称粗集、粗糙集）理论是由波兰华沙理工大学 Pawlak 教授于 20 世纪 80 年代初提出的一种研究不完整、不确定知识和数据的表达、学习、归纳的理论方法，近年来得到国际上众多学者的重视。我国也在国家自然科学基金、国家 863 计划和一些省、市科学研究基金的支持下开展了一定的研究工作，逐渐取得了一些研究成果。

Rough 集的研究对象是由一个多值属性（特征、症状、特性等）集合描述的一个对象（观察、病历等）集合，对于每个对象及其属性都有一个值作为其描述符号，对象、属性和描述符是表达决策问题的 3 个基本要素。这种表达形式也可以看成为一个二维表格，表格的行与对象相对应，列对应于对象的属性；各行包含了表示相应对象信息的描述符，还有关于各个对象的类别成员的信息。通常，关于对象的可得到的信息不一定足以划分其成员类别。换句话说，这种不精确性导致了对象的不可分辨性。给定对象间的一个等价关系，即导致由等价类构成的近似空间的不分明关系，Rough

集就用不分明对象类形成的上近似和下近似来描述。这些近似分别对应了确定属于给定类的最大的对象集合和可能属于给定类的最小的对象集合。下近似和上近似的差是一个边界集合，它包含了所有不能确切判定是否属于给定类的对象。这种处理可以定义近似的精度和质量。Rough 集方法可以解决重要的分类问题，所有冗余对象和属性的约简包含属性的最小子集，能够很好地近似分类，得到可以接受质量的分类。而且，它还可以用决策规则集合的形式表示最重要属性和特定分类之间的所有重要关系。

本书是在课题组几年来进行多项相关科研项目研究所取得成果的基础上总结而成的，对国内、外有关的研究成果也进行了归纳总结并融入各章节的内容中。本书从基本理论概念到实际应用分析，从理论模型到算法实现和应用系统，都进行了详尽的讨论，全书分为 11 章。

第 1 章对集合论的基础知识进行了介绍。讨论了集合论的基本概念、集合代数运算以及集合关系。

第 2 章讨论了信息表知识表达系统这种 Rough 集理论的特殊处理对象，讲述了知识的分类概念、决策表等基本概念。

第 3 章介绍了 Rough 集的基本理论基础，如近似集合的概念、粗糙度与分类质量、Rough 集的代数性质和 Rough 集关系、不完备信息系统中 Rough 集理论的扩充等概念，这是以后各章节内容的基础。

第 4 章对知识获取的基本问题进行讨论，对知识获取的模型、可辨识矩阵、属性重要性以及决策规则等内容进行了介绍和分析。

第 5 章讨论知识系统不确定性表示与处理问题，对知识表示的基本方法、概率模型、可信度模型、证据理论、模糊推理等不确定性推理模型和决策表的不确定性度量问题进行了研究分析。

第 6 章讨论数据预处理问题，介绍了决策表的几种补齐算法和离散化算法。

第 7 章对信息表属性约简的理论和算法进行了介绍，讨论了

属性约简的集合观念和信息熵观念，给出了几种可行的属性约简算法。

第 8 章讨论信息表值约简问题，介绍了几种值约简算法和缺省规则获取算法。

第 9 章介绍逻辑推理系统，对逻辑推理的几种推理方法和知识系统中的不一致性问题以及不一致情况下的推理策略进行了分析讨论。

第 10 章分析介绍了几个典型的应用 Rough 集理论来解决实际问题的实例系统，如水资源调度、临床医疗诊断、客户行为预测和文本分类等。

第 11 章介绍了世界各国研究人员所开发的几个演示系统。

本书的完成，是与课题组长期的辛勤努力工作分不开的，在此要特别感谢的是课题组吴渝博士，她在项目研究工作中做了大量的工作；还有多位研究生，如常犁云、刘锋、侯利娟等。本书是大家辛勤劳动的结果。

本书作者 1998 年至 1999 年在美国 University of North Texas 作访问学者期间，该校计算机科学系教授 Paul S Fisher 博士为作者的科研工作提供了大力的支持和帮助；1999 年在加拿大 University of Regina 作访问学者期间，该校计算机科学系教授 Y. Y. Yao 博士与作者进行了很多有益的学术交流和讨论，并给作者提供了大量的参考文献资料，这无疑对本书的写作起到了很好的促进作用。南昌大学刘清教授也与作者进行过很多交流讨论并提供了大量的参考文献资料。西南交通大学靳蕃教授、重庆大学曹长修教授、重庆大学程代杰教授、中国科学院软件研究所王驹教授等也给予了作者很多鼓励和帮助。在此，对他们的支持和帮助，表示衷心的感谢。本书中还引用了许多国内外同行专家的一些研究成果，在此也对他们表示深深的谢意。

另外，还要特别感谢我的导师施鸿宝教授（现为同济大学教授）。我在西安交通大学攻读硕士学位和博士学位期间，得到了他

悉心的指导，是在他多年精心培养下，我才具有了现在研究工作的能力，走上了科学研究的道路，他对我的教导和影响也必将在我以后的科研工作中起到很大的作用。

在此，我还要感谢我的妻子何晓行女士，是她的大力支持才使我能够完成研究工作和本书的写作。我的父母亲也给予了我无微不至的关心，这些都是我完成本书的基础。

还要感谢西安交通大学出版社为本书出版给予的帮助。是大家的共同努力才使得本书能够最终出版，与读者见面。

课题组的研究工作、本书的写作完成和出版，得到了国家自然科学基金（编号：69803014）、国家 863 计划（编号：863-317-04-18-99）、攀登计划、教育部高等学校骨干教师资助计划、教育部留学回国人员科研启动基金以及重庆市应用基础研究基金的部分资助，在此一并表示诚挚的谢意。

由于作者水平有限，时间仓促，而且部分内容还是课题组所取得的阶段性研究成果，不妥、错误之处在所难免，希望能够得到读者的批评指正。

王国胤

2000 年 9 月于重庆

本专著得到下列基金资助：

- 攀登——特别支持费
- 国家自然科学基金(编号：69803014)
- 国家863计划(编号：863-317-04-18-99)
- 教育部高等学校骨干教师资助计划
- 教育部留学回国人员科研启动基金
- 重庆市应用基础研究基金

目　　录

第1章　集合论基础

集合是现代数学和逻辑学的基本概念之一。本世纪以来，关于集合的理论——集合论，对现代数学和逻辑学的发展产生了巨大影响，今天它已成为数学和逻辑学的一种基础理论。集合论的创始人是康托尔(G. Cantor，1845～1918)，他所做的工作一般称为朴素集合论，由于在定义集合的方法上缺乏限制，会导致悖论。为了消除这些悖论，经过许多数学家的努力，20世纪初又创建了更精致的理论——公理化集合论，集合论至今仍在发展中。出于一些处理问题的需要，扎德(L. A. Zadeh)教授1965年提出了模糊集合的概念，模糊集理论在很多控制领域取得了很大的成功。近年来，波兰华沙理工大学坡那克(Z. Pawlak)教授等一批科学家提出了Rough集理论，研究不完整数据及不精确知识的表达、学习、归纳等方法。为了很好的理解Rough集理论，我们在本章首先对集合论进行简单的介绍。

我们首先介绍集合的基本概念，如集合、空集、子集，然后介绍定义在集合上的运算——集合代数，包括集合的基本运算(并、交、差、补)和集合运算的一些定律，最后对定义在集合上的关系进行介绍。

1.1　集合论的基本概念

用集合论的创始人康托尔曾经解释过的话来说：所谓集合，可以理解为由我们的知觉或思维确定的、能明确区分开的对象 m，聚集成的一个整体 M，这些对象 m 叫做 M 的“元素”。一般地说，集合就是把直观上或思想上的一些确定的、彼此不同的对象作为一

个整体，组成该整体的对象叫做该集合的元素。此时我们便说，这些元素组成该集合，这些元素属于该集合。

如下列集合：

1. 中华人民共和国的直辖市（北京市、上海市、天津市、重庆市）构成一个4元素的集合。

2. 所有三角形构成一三角形集合。

3. 坐标满足方程 $x^2+y^2\leq R^2$ 的全部点构成（如图1.1所示）的点集。

通常用大写字母 $A,B,C,\cdots$ 代表集合；用小写字母 $a,b,c,\cdots$ 代表元素。

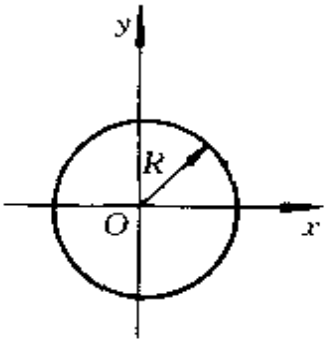


图 1.1

如果 a 是集合 A 的一个元素，则记为

$$a\in A。$$

如果 a 不是集合 A 的一个元素，则记为

$$a\notin A。$$

任一元素，对某一集合而言，或属于该集合，或不属于该集合，二者必居其一，也只居其一。

通常，集合有两种表示方法。

第一种为列举法，就是把集合中的元素一一列举出来，写在花括号内。

例 1.1 所有小于8的正整数组成的集合 A 可写成：

$$A=\{1,2,3,4,5,6,7\}。$$

例 1.2 全体自然数所组成的集合 $\mathbf{N}_+$ 可写成：

$$\mathbf{N}_+=\{1,2,3,\cdots,n,\cdots\}。$$

虽然集合 $\mathbf{N}_+$ 的元素是列举不尽的，但是例1.2已经列出了其中有代表性的元素，省略号表示可以继续顺次地写出它的元素。

第二种为描述法，就是用描述集合元素的共同性质的方法来表示这个集合。这种方法又叫做特征法。

例 1.3 所有小说组成的集合 A 可写成：

$$A=\{x \mid x \text{ 为小说}\}。$$

例 1.4　0 和 5之间的实数可写成：

$$A=\{x \mid 0<x<5\}。$$

比较这两种表示方法，可以看出：列举法的好处是可以具体看清集合的元素；描述法的好处是刻画出了集合元素的共同特征。应用时可根据需要任意选用，不受限制。

集合的元素可以是一个集合，如

$$A=\{a,b,C\},\quad \text{而 } C=\{0,1\}。$$

仅含有一个元素的集合称为单元素集合。

根据集合所包含的元素个数的多少，可以把集合分为无限集、有限集和空集。

如果集合是由无限多个元素组成的，则这样的集合称为无限集。如{自然数集}等。

如果集合是由有限个元素组成的，则这样的集合称为有限集。如{哲学系的学生}等。

如果一个集合不包含任何元素，则这样的集合称为空集。如{小于 1 大于 2 的实数}等。空集用符号“$\varnothing$”表示。

我们在讨论具有某种共同性质的对象所组成的集合时，把具有这种共同性质的一切元素组成的集合，叫作全集，用“Ω”表示。全集也称论域。

设 A 是一个集合，A 的补集 $\overline{A}$ 是由全集中不属于集合 A 的所有元素组成的集合。

$$\overline{A}=\{x \mid x\in\Omega \wedge x\notin A\}。$$

设 A 是一集合，A 的幂集 $\rho(A)$是 A 的所有子集组成的集合。

集合所包含元素的个数称为该集合的基数或势。集合 A 的基数记为 $|A|$或 $\mathrm{card}(A)$。例如：

若 $A=\{a,b\}$，则 $|A|=2$。

设 A 和 B 是集合，如果 A 的每一个元素都是 B 的一个元素，那么称 A 是 B 的子集，记为 $A\subseteq B$，用逻辑符号表示为：

$$A\subseteq B \Leftrightarrow \forall x(x\in A\rightarrow x\in B)。$$

如果 $A\subseteq B$ 且 $A\neq B$，那么称 A 是 B 的真子集，记作 $A\subset B$，用逻辑符号表示为：

$$A\subset B \Leftrightarrow (A\subseteq B)\wedge(A\neq B)$$
$$\Leftrightarrow \forall x(x\in A\rightarrow x\in B)\wedge\exists x(x\in B\wedge x\notin A)$$

定理 1.1 对任意集合 A，有 $A\subseteq\Omega$。

定理 1.2 设 A 和 B 是集合，$A=B$ 当且仅当 $A\subseteq B$ 并且 $B\subseteq A$。

定理 1.3 对任意集合 A，有 $A\subseteq A$。

定理 1.4 设 A，B 和 C 是集合，若 $A\subseteq B$ 且 $B\subseteq C$，则 $A\subseteq C$。

定理 1.5 对任意集合 A，有 $\varnothing\subseteq A$。

定理 1.6 空集是唯一的。

定理 1.7 如果集合 A 包含 n 个元素，则其幂集 $\rho(A)$ 的元素个数为 2^n；如果 A 是无限集，则 $\rho(A)$ 也是无限集。

1.2 集合代数运算

正如数学中两数之间可以进行加、减、乘、除等运算一样，两集合之间也有类似的运算。本节介绍集合代数运算的基本概念以及一些重要的性质。

定义 1.1 设 A 和 B 是集合，则

(a) A 和 B 的并(逻辑和)是集合，记为 $A\cup B$，且

$$A\cup B=\{x\mid x\in A\vee x\in B\}。$$

集合的并可用 Venn 图的阴影部分表示为图 1.2。

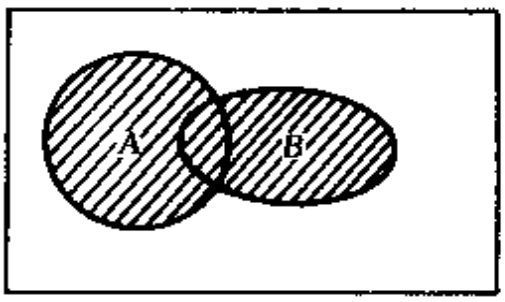


图 1.2 $A\cup B$

(b) A 和 B 的交(逻辑积)是集合，记为 $A\cap B$，且

$$A\cap B=\{x\mid x\in A\wedge x\in B\}。$$

集合的交可用 Venn 图的阴影部分表示为图 1.3。

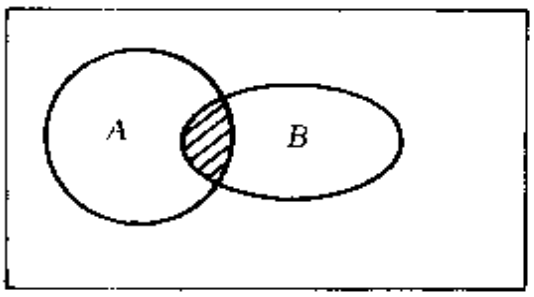


图 1.3　$A \cap B$

(c) A 和 B 的差(逻辑差)是集合，记为 $A \setminus B$，即

$A \setminus B = \{x \mid x \in A \wedge x \notin B\}$。

集合的差可用 Venn 图的阴影部分表示为图 1.4。

例 1.5　设 $A=\{a,b,c,d\}$，$B=\{b,c,e\}$，则

$A \cup B = \{a,b,c,d,e\}$；

$A \cap B = \{b,c\}$；

$A \setminus B = \{a,d\}$；

$B \setminus A = \{e\}$。

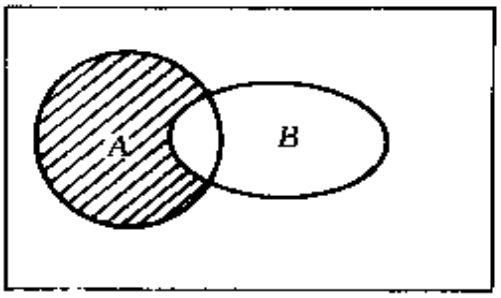


图 1.4　$A \setminus B$

定义 1.2　设 A 和 B 是集合，如果 $A \cap B = \varnothing$，那么称 A 和 B 是不相交的。

定理 1.8　(幂等律)

$A \cup A = A$；

$A \cap A = A$。

定理 1.9　(交换律)

$A \cup B = B \cup A$；

$A \cap B = B \cap A$。

定理 1.10　(结合律)

$(A \cup B) \cup C = A \cup (B \cup C)$；

$(A \cap B) \cap C = A \cap (B \cap C)$。

定理 1.11　(分配律)

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$；

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$。

定理 1.12　(恒等律)

$A \cup \varnothing = A$；

$A \cap \Omega = A$；

$A \cap \varnothing = \varnothing$；

$A \cup \Omega = \Omega$。

定理 1.13 （互补律）

$A \cup \overline{A} = \Omega$；

$A \cap \overline{A} = \varnothing$。

定理 1.14 （德摩根律）

$\overline{A \cup B} = \overline{A} \cap \overline{B}$；

$\overline{A \cap B} = \overline{A} \cup \overline{B}$。

定理 1.15 （吸收律）

$A \cup (A \cap B) = A$；

$A \cap (A \cup B) = A$。

定理 1.16 （双否律）

$\overline{\overline{A}} = A$。

集合的代数运算还有如下性质：

(1) $A = \overline{B} \rightarrow B = \overline{A}$ （逆反原则）；

(2) $A \cup B \neq \varnothing \rightarrow A \neq \varnothing \vee B \neq \varnothing$；

(3) $A \cap B \neq \varnothing \rightarrow A \neq \varnothing \wedge B \neq \varnothing$；

(4) $A \subseteq A \cup B$， $B \subseteq A \cup B$；

(5) $A \cap B \subseteq A$， $A \cap B \subseteq B$；

(6) $\overline{\varnothing} = \Omega$， $\overline{\Omega} = \varnothing$。

对于有限集合的基数，集合代数运算具有如下性质：

定理 1.17 设 A, B 都是有限集合，则以下公式成立：

(1) $|A \cup B| = |A| + |B| - |A \cap B|$；

(2) $|A \cap B| \leqslant \min(|A|, |B|)$；

(3) $|A \setminus B| \geqslant |A| - |B|$；

(4) $|\rho(A)| = 2^{|A|}$。

1.3 集合关系

为了理解集合关系的概念，我们首先给出集合的笛卡儿乘积（叉积）的概念。

定义 1.3 两个元素 a_1, a_2 组成的序列记作 (a_1, a_2)，称作二元组或序偶，a_1, a_2 分别称为二元组 (a_1, a_2) 的第一和第二分量。

定义 1.4 两个二元组 (a, b) 和 (c, d) 相等当且仅当 $a=c$ 并且 $b=d$。

定义 1.5 设 $a_1, a_2, \cdots, a_n$ 是 n 个元素，定义 $(a_1, a_2, \cdots, a_n) = ((a_1, a_2, \cdots, a_{n-1}), a_n)$ 为 n 元组，这里 $n>2$。

由元组的定义知道，元组中元素的次序是重要的，例如 $(1,2) \neq (2,1)$，这是和集合不一样的。

我们通常需要由集合簇 $A_1, A_2, \cdots, A_n$ 的元素生成的所有 n 元组，因而有如下定义：

定义 1.6 集合 A, B 的笛卡儿乘积记为 $A \times B$，它是二元组集合 $\{(a,b) \mid a \in A \wedge b \in B\}$。

定义 1.7 集合 $A_1, A_2, \cdots, A_n$ 的笛卡儿乘积记为 $A_1 \times A_2 \times \cdots \times A_n$ 或 $\overset{n}{\underset{i=1}{\times}} A_i$，定义为

$$\overset{n}{\underset{i=1}{\times}} A_i = (A_1 \times A_2 \times \cdots \times A_{n-1}) \times A_n, \qquad n>2。$$

由此可以看出，笛卡儿积 $\overset{n}{\underset{i=1}{\times}} A_i$ 是 n 元组集合

$$\{(a_1, a_2, \cdots, a_n) \mid a_i \in A_i \wedge n \geqslant i \geqslant 1\}。$$

另外，对一切 i，如果 $A_i = A$，$\overset{n}{\underset{i=1}{\times}} A_i$ 可简记为 A^n。

集合的笛卡儿积是不可交换的，结合律也不成立，因为 $(A \times B) \times C$ 和 $A \times (B \times C)$ 的元素的形式分别是 $((a,b),c)$ 和 $(a,(b,c))$，按照定义 1.4 可知二者不可能相等，但二元笛卡儿积在并与交上可分配。

定理 1.18 如果 A,B,C 都是集合，则

(1) $A\times(B\cup C)=(A\times B)\cup(A\times C)$；

(2) $A\times(B\cap C)=(A\times B)\cap(A\times C)$；

(3) $(A\cup B)\times C=(A\times C)\cup(B\times C)$；

(4) $(A\cap B)\times C=(A\times C)\cap(B\times C)$。

定理 1.19 如果所有 $A_i(i=1,2,\cdots,n)$ 都是有限集合，则

$$|A_1\times A_2\times\cdots\times A_n|=|A_1|\cdot|A_2|\cdot\cdots\cdot|A_n|。$$

在介绍了笛卡儿积之后，我们就可以讨论集合关系了。

定义 1.8 $A\times B$ 的子集叫做 A 到 B 的一个二元关系；$A_1\times A_2\times A_3$ 的子集叫做 $A_1\times A_2\times A_3$ 上的一个三元关系；$A_1\times A_2\times\cdots\times A_n$ 的子集叫做 $A_1\times A_2\times\cdots\times A_n$ 上的一个 n 元关系；A^n 的子集叫做 A 上的 n 元关系。

由此看出，关系是一个集合。

定义 1.9 设 R 是 $\overset{n}{\underset{i=1}{\times}}A_i$ 的子集，如果 $R=\varnothing$，则称 R 是空关系；如果 $R=\overset{n}{\underset{i=1}{\times}}A_i$，则称 R 为全关系。

最为重要的关系是二元关系，在不作特别说明的时候，本书中所说的关系均指二元关系。

二元关系有自己专用的记法和术语。

设 $A=\{a,b,c,d,e,f,g\}$，

$B=\{h,i,j,k,l\}$，

$R=\{(a,h),(c,j),(f,k)\}$，

关系 R 可如图 1.5 那样形象地表示。$(a,h)\in R$ 也可写成 aRh，称为中缀记法，读作 a 和 h 有关系 R。

$D(R)=\{x\mid\exists\ y(xRy)\}$ 叫做关系 R 的域。

$R(R)=\{y\mid\exists\ x(xRy)\}$ 叫做关系 R 的值域。A 叫做关系 R 的前域，B 叫做关系 R 的陪域。

关系是元组的集合，对它可进行集合运算，运算结果定义一个新的关系。设 R 和 S 是两个给定集合上的二元关系，则

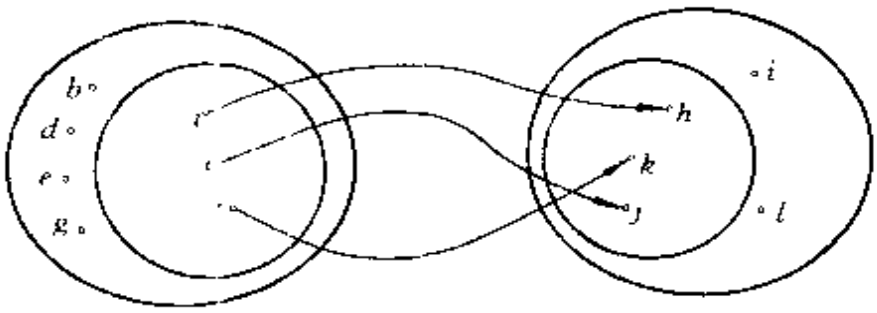


图 1.5

$x(R\cup S)y$, $x(R\cap S)y$, $x(R\setminus S)y$, $x(\overline{R})y$ 可以分别定义如下：

$$x(R\cup S)y \Leftrightarrow xRy \vee xSy;$$

$$x(R\cap S)y \Leftrightarrow xRy \wedge xSy;$$

$$x(R\setminus S)y \Leftrightarrow xRy \wedge \neg(xSy);$$

$$x(\overline{R})y \Leftrightarrow \neg(xRy)。$$

定义 1.10　对于从集合 A 到集合 B 的关系 $R\subseteq A\times B=\{(a,b)\mid a\in A\wedge b\in B\}$，其逆关系定义为

$$R^{-1}\subseteq B\times A=\{(b,a)\mid a\in A\wedge b\in B\wedge(a,b)\in R\}。$$

例如：非空集合 $A=\{1,2\}$ 到 $B=\{a,b,c\}$ 的关系 $R=\{(1,a),(2,b)\}$ 的逆关系 $R^{-1}=\{(a,1),(b,2)\}$。

定义 1.11　如果关系 R 是集合 A 和 B 的关系，S 是集合 B 和 C 的关系，则关系 R 和 S 的合成 $R\circ S$ 为

$$R\circ S=\{(a,c)\mid \exists\, b(b\in B\wedge(a,b)\in R\wedge(b,c)\in S)\}。$$

表达有限集合到有限集合的二元关系时，矩阵是直观有力的工具。

定义 1.12　给定集合 $A=\{a_1,a_2,\cdots,a_m\}$ 和 $B=\{b_1,b_2,\cdots,b_n\}$ 及一个 A 到 B 的二元关系 R，使得

$$r_{ij}=\begin{cases}1, & \text{如果 } a_iRb_j;\\ 0, & \text{如果 } a_i\overline{R}b_j。\end{cases}$$

则称矩阵 $M_R=[r_{ij}]$ 是关系 R 的关系矩阵。

例 1.6　设 $A=\{1,2,3,4\}$，A 上的二元关系 $R=\{(x,y)\mid x>$

$y\}$可以用关系矩阵表示如下：

$$\boldsymbol{M}_R=\begin{bmatrix}0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0\\ 1 & 1 & 1 & 0\end{bmatrix}$$

集合 A 上的二元关系，也可以用有向图表示，具体方法是：一般用小圆圈标上 a_i 表示元素 a_i，小圆圈叫图的结点，如果$(a_i,a_j)\in R$，则从结点 a_i 到 a_j 画一有向弧；如果$(a_i,a_i)\in R$，则通过结点 a_i 画一自回路（封闭）的有向弧。这样得到的图叫关系 R 的关系图。关系图中没有与弧相连接的结点称为孤立点。

例 1.7 设 $A=\{a,b,c,d,e\}$，关系 $R=\{(a,b),(b,b),(c,b),(c,d),(d,c)\}$，则关系 R 可用关系图表示为图 1.6。

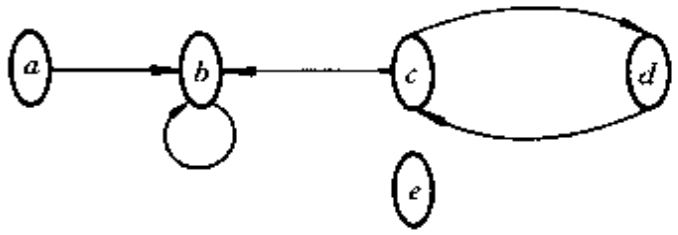


图 1.6

在对关系的研究中，关系的某些特性起着重要作用，这里简单概述如下：

定义 1.13 设 R 是 A 上的二元关系，定义：

1. 如果对 A 中每一个元素 x 都有 xRx，则称 R 是自反的。

2. 如果对 A 中每一个元素 x 都有 $\neg(xRx)$，则称 R 是非自反的。

3. 如果对 A 中任意元素 x,y 都有 xRy 蕴涵着 yRx，则称 R 是对称的。

4. 如果对 A 中任意元素 x,y，xRy 且 yRx 蕴涵着 $x=y$，则称 R 是反对称的。

5. 如果对 A 中任意元素 x,y,z，xRy 且 yRz 蕴涵着 xRz，则称 R 是传递的。

二元关系的一个重要类型是等价关系，这也是本书中经常要

用到的一种关系，其定义如下：

定义 1.14　如果集合 A 上的二元关系 R 是自反的、对称的和传递的，则称 R 是等价关系。

等价关系的有向图的每一个分图是完全图（指每一结点有自回路、每两结点间有两条不同方向的有向弧）。

定义 1.15　设 R 是集合 A 上的等价关系，对每一 $a\in A$，a 关于 R 的等价类是集合 $\{x\mid xRa\}$，记为 $[a]_R$，简记为 $[a]$；称 a 为等价类 $[a]$ 的表示元素。如果等价类个数有限，则 R 的不同等价类的个数叫做 R 的秩；否则称秩是无限的。

例 1.8　设集合 $A=\{a,b,c,d,e,f\}$，A 上的关系 $R=\{(a,a),(b,b),(c,c),(d,d),(e,e),(f,f),(a,b),(b,a),(a,c),(c,a),(b,c),(c,b),(d,e),(e,d)\}$，则等价关系 R 的等价类如下：

$$[a]=[b]=[c]=\{a,b,c\},$$
$$[d]=[e]=\{d,e\},$$
$$[f]=\{f\}.$$

等价关系 R 的秩是 3.

定理 1.20　设 R 是非空集合 A 上的等价关系，则 aRb 当且仅当 $[a]=[b]$。

定理 1.21　设 R 是非空集合 A 上的等价关系，则对所有 $a,b\in A$，或者 $[a]=[b]$，或者 $[a]\cap[b]=\varnothing$。

利用等价关系，我们可以得到集合的覆盖和划分的概念。

定义 1.16　给定非空集合 A 和非空集合簇 $\pi=\{A_1,A_2,\cdots,A_m\}$，如果 $A=\bigcup_{i=1}^{m}A_i$，那么称集合簇 π 是 A 的覆盖。

定义 1.17　给定非空集合 A 和非空集合簇 $\pi=\{A_1,A_2,\cdots,A_m\}$，如果

（1）π 是 A 的覆盖，即 $A=\bigcup_{i=1}^{m}A_i$；

（2）$A_i\cap A_j=\varnothing$ 或 $A_i=A_j$　$(i,j=1,2,\cdots,m)$。

那么称集合簇 π 是 A 的一个划分。划分的元素 A_i 称为划分 π 的

块。

定理 1.22 设 A 是非空集合，R 是 A 上的等价关系。R 的等价类集合 $\{[a]_R \mid a\in A\}$ 是 A 的划分。

定义 1.18 设 A 是非空集合，R 是 A 上的等价关系，称划分 $\{[a]_R \mid a\in A\}$ 为商集 A/R，也叫做 A 模 R。

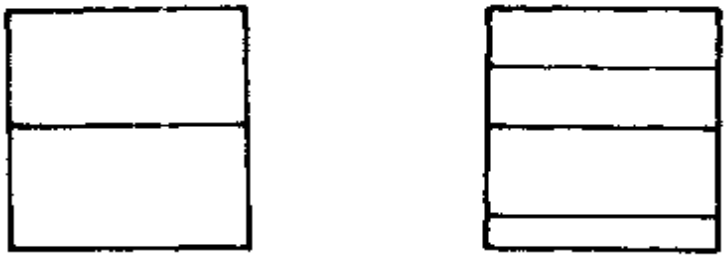
定理 1.23 设 A 是非空集合，R_1，R_2 是 A 上的等价关系，那么 $R_1=R_2$ 当且仅当 $A/R_1=A/R_2$。

这说明非空集合上的等价关系决定了该集合的一个划分，反之，由非空集合的划分也可以得到相应的一个等价关系。

定理 1.24 设 π 是非空集合 A 的一个划分，则 A 上的二元关系 $R=\bigcup_{B\in\pi} B\times B$（或写成 $aRb\leftrightarrow \exists B(B\in\pi\wedge a\in B\wedge b\in B)$）是 A 上的等价关系。

定义 1.19 设 π 和 π' 是非空集合 A 的划分。如果 π' 的每一块都包含于 π 的一块中，则称 π' 细分 π，或说 π' 是 π 的细分。如果 π' 细分 π 且 $\pi\neq\pi'$，则称 π' 是 π 的真细分。

例 1.9 设非空集合 A 上的关系 R 决定的划分 π 如图 1.7(a)所示，关系 R' 决定的划分 π' 如图 1.7(b)所示，称 π' 细分 π。



(a) π (b) π'

图 1.7

定理 1.25 设 π 和 π' 是非空集合 A 的划分，R 和 R' 分别是 π 和 π' 导出的 A 上的等价关系。则 π' 细分 π 当且仅当 $R'\subseteq R$。

定义 1.20 设 π_1 和 π_2 分别是非空集合 A 上的等价关系 R_1

和 R_2 导出的划分，则 $R=R_1 \cap R_2$ 所导出的划分 π 称为划分 π_1 和 π_2 的积，记为 $\pi_1 * \pi_2$。

例 1.10　图 1.8是划分 π_1 和 π_2 的积的一个图示。

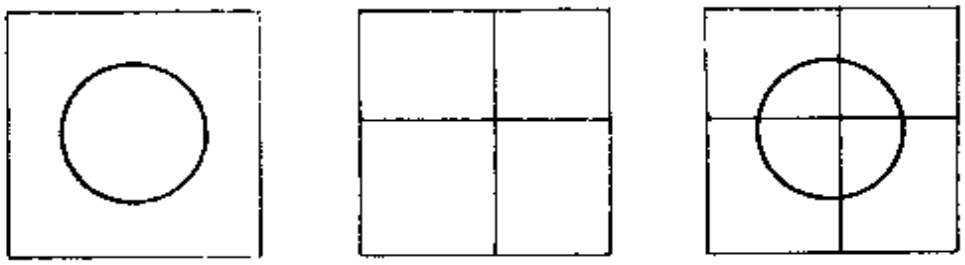


图 1.8　划分 π_1 和 π 的积

定义 1.21　设 π_1 和 π_2 分别是非空集合 A 上的等价关系 R_1 和 R_2 导出的划分，则 $R_1 \cup R_2$ 的传递闭包 R(包含 R_1 和 R_2 的最小的等价关系)所导出的划分 π 称为划分 π_1 和 π_2 的和，记为 $\pi_1 + \pi_2$。

例 1.11　图 1.9是划分 π_1 和 π_2 的和的一个图示。

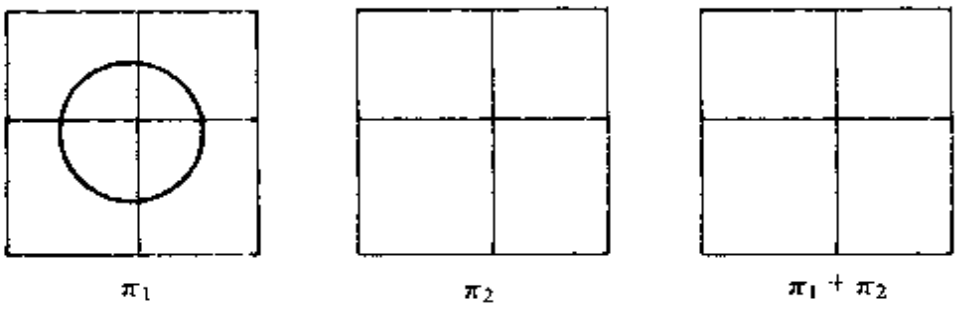


图 1.9　划分 π_1 和 π_2 的和

定理 1.26　设 π_1 和 π_2 分别是非空集合 A 上的划分，π_1 和 π_2 的和、π_1 和 π_2 的积都是唯一的。

第 2 章　信息表知识表达系统

知识表达是智能信息系统的关键。所谓知识获取，就是要从大量的原始数据信息中分析发现有用的规律信息，即是将知识从一种原来的表达形式（原始数据表达形式）转换为一种新的目标表达形式（人类或者计算机便于处理的形式，如逻辑规则等）。基于 Rough 集理论的知识发现，主要是借助于信息表这样一种有效的数据表知识表达方式。本章将介绍这种信息表知识表达系统。首先对知识利用分类的概念进行描述，然后对信息表描述的知识表达系统进行介绍，并加以形式化描述，再对决策表这种特殊的信息表进行详细讨论，最后讨论与决策表相应的决策规则。

2.1　知识的分类概念

知识是人类通过实践认识到的客观世界的规律性的东西，是人类实践经验的总结和提炼，具有抽象和普遍的特性。知识是信息经过加工处理、解释、挑选和改造而形成的。知识是命题、规则等的集合。知识一般可分为说明性知识、过程性知识和控制性知识。说明性知识提供概念和事实。例如，一个智能检索系统中，说明性知识包括说明具体事实的数据库内容。用规则表示问题的知识称作过程性知识。智能信息检索系统中利用过程性知识处理说明性知识。用控制策略表示问题的知识称为控制性知识。控制性知识包含有关各种处理过程、策略和结构的知识，常用来协调整个问题求解的过程。

从认知科学的一些观点来看，可以认为知识来源于人类以及其他物种的分类能力。本书中，我们认为知识即是将对象进行分类

的能力。上述的说明性知识可以认为是对现实世界客观个体的描述，即是区分客观个体的知识；过程性知识实质上是通过利用说明性知识对客观个体进行分类的知识；而控制性知识也是关于如何用过程性知识实现对客观个体进行分类的知识，也可以认为是关于对过程性知识的分类。

假定我们起初讨论域里的个体（对象）具有必要的信息或知识，通过这些知识能够将其划分到不同的类别。若我们对两个元素具有相同的信息，则他们是不可区分的，即根据已有的信息不能够将其划分开，显然这是一种等价关系。通常，我们在对现实问题进行处理的时候，会将我们讨论的现实个体（或称元素、对象、样本）局限在某一个特定的区域范围之内，这个区域内的所有个体就组成问题的论域 U。以分类为基础，可以将分类理解为等价关系，而这些等价关系对论域 U 进行划分。对于论域中由等价关系划分出的任意子集 X，都可称之为 U 中的一个概念。这里，我们认为空集 $\varnothing$ 也是一个特殊的概念。论域 U 中的任意概念簇称为关于 U 的抽象知识，简称为知识，它代表了对 U 中个体的分类。这样，知识就可以定义为：给定一组数据（集合）U 和等价关系集合 R，在等价关系集合 R 下对数据集合 U 的划分，称为知识，记为 U/R。U 上的一簇划分（对 U 的分类）称为关于 U 的知识库。关于 U 的一个知识库也可以理解为一个关系系统，其中 U 为论域，R 是 U 上的一簇等价关系，根据这些等价关系就可以对 U 进行不同的划分（知识），每种划分将把 U 分为不同的子集（概念）。

设 U 是一个论域，R 是 U 上的一个等价关系。U/R 表示 U 上由 R 导出的所有等价类。$[x]_R$ 表示包含元素 x 的 R 的等价类，$x \in U$。一个知识库就是一个关系系统 $K=(U,P)$，其中 U 是论域，P 是 U 上的一个等价关系簇。如果 $Q \subseteq P$ 且 $Q \neq \varnothing$，则 $\cap Q$（Q 的所有等价关系的交）也是一个等价关系，记作 $\mathrm{IND}(Q)$。

定义 2.1　设 $K=(U,P)$ 和 $K_1=(U,Q)$ 是两个知识库。如果 $\mathrm{IND}(P)=\mathrm{IND}(Q)$，则称 K 和 K_1（或 Q 和 P）是等价的，记作 $K \cong$

K_1(或 $P \subseteq Q$)。

知识库 K 和 K 等价,意味着 K 和 K_1 具有相同的基础类,因而它们具有相同的表达能力。

例 2.1 表 2.1所示的个体集合组成论域 U,其中包含 6 个个体,每个个体是一个四元组,元组的每一维表示个体的一个属性信息。这些个体都可以通过用其属性知识来描述。例如一个个体可以代表其是否头疼、是否肌肉疼、体温是否正常、是否是流感。如果我们按照某一个属性或多个属性来描述这些个体,就可以得到不同的分类知识。

表 2.1

个体编号	头疼	肌肉疼	体温	流感
e_1	是	是	正常	否
e_2	是	是	高	是
e_3	是	是	很高	是
e_4	否	是	正常	否
e_5	否	否	高	否
e_6	否	是	很高	是

按照头疼来分类:

$U/$头疼$=\{\{e_1,e_2,e_3\},\{e_4,e_5,e_6\}\}$。

这里,e_1,e_2 和 e_3 这三个个体在头疼这个属性上是不可区分的,即他们一起构成一个类,e_4,e_5 和 e_6 这三个个体构成另一个类。

按照肌肉疼、体温和流感这三个属性分别来分类:

$U/$肌肉疼$=\{\{e_1,e_2,e_3,e_4,e_6\},\{e_5\}\}$;

$U/$体温$=\{\{e_1,e_4\},\{e_2,e_5\},\{e_3,e_6\}\}$;

$U/$流感$=\{\{e_1,e_4,e_5\},\{e_2,e_3,e_6\}\}$。

按照头疼和肌肉疼这两个属性来共同分类:

$U/$头疼和肌肉疼$=\{\{e_1,e_2,e_3\},\{e_4,e_6\},\{e_5\}\}$。

按照头疼和体温这两个属性来共同分类:

$U/$头疼和体温$=\{\{e_1\},\{e_2\},\{e_3\},\{e_4\},\{e_5\},\{e_6\}\}$。

按照头疼和流感这两个属性来共同分类：

$U/$头疼和流感$=\{\{e_1\},\{e_2,e_4\},\{e_3,e_5\},\{e_6\}\}$。

由此，可以看出，我们可以用不同的标准来对论域进行分类，得到不同的概念和抽象，有的概念是我们需要的，有的概念是没有价值的，知识获取就是要探寻有用的概念，并得到概念之间的关系。

2.2　信息表知识表达系统

人之所以有智能行为是因为他们有知识。要让机器具有智能行为的能力，就必须让机器具有相应的知识，它需要以人的知识作为其工作基础。知识表示就是要研究用机器表示知识的可行的、有效的、通用的原则和方法。近年来知识表示的研究引起了广泛的注意。目前，常用的知识表示方法有逻辑模式、框架、语意网络、产生式规则、状态空间、剧本等，这些是知识工程需要研究的内容。本节中，我们将介绍一种基于信息表的知识表达形式，它是 Rough 集理论中对知识进行表达和处理的基本工具。

在人工智能研究中，一个实例（现实世界中的一个对象、个体）经常使用属性-值对的集合来表示，实例集就是这样的实例集合，记为 U。U 可被划分为有限个类 $X_1,X_2,\cdots,X_n$，使得

$$X_i\subseteq U,\ X_i\neq\varnothing,\qquad X_i\cap X_j=\varnothing(i\neq j),$$

$$(i,j=1,2,\cdots,n\ \text{且}\ \bigcup X_i=U)。$$

信息表知识表达系统的基本成分是研究对象的集合，关于这些对象的知识是通过指定对象的属性（特征）和它们的属性值（特征值）来描述的。一般地，一个信息表知识表达系统 S 可以表示为

$$S=(U,R,V,f)。$$

这里，U 是对象的集合，也称为论域，$R=C\cup D$ 是属性集合，子集 C 和 D 分别称为条件属性集和结果属性集，$V=\bigcup_{r\in R}V_r$ 是属性

值的集合，V_r 表示属性 $r\in R$ 的属性值范围，即属性 r 的值域，$f: U\times R\to V$ 是一个信息函数，它指定 U 中每一个对象 x 的属性值。

为了直观方便，U 也可以写成一个表，纵轴表示实例标记，横轴表示实例属性，实例标记与属性的交会点就是这个实例在这个属性的值。这个表称为信息表，是表达描述知识的数据表格。

对于每个属性子集 $B\subseteq R$，我们定义一个不可分辨二元关系（不分明关系）IND(B)，即

$$\mathrm{IND}(B)=\{(x,y)\mid (x,y)\in U^2,\forall\, b\in B(b(x)=b(y))\}。$$

显然，IND(B)是一个等价关系，且

$$\mathrm{IND}(B)=\bigcap_{b\in B}\mathrm{IND}(\{b\})。$$

每个子集 $B\subseteq R$ 也可称为一个属性，当 B 是单元素集时，称 B 为原始的，否则称 B 为复合的。属性 B 可以认为是用等价关系（在该属性上的取值相等）表示的知识的一个名称，称为标识属性。一个表可以看作是定义的一个等价关系簇，即知识库。

实际上，信息表这种数据表格知识表达系统是对客观对象的描述和罗列，表达的是属于说明性的知识。当信息表包含的数据足以反映论域的时候，通过属性所对应的等价关系就可以体现论域中的过程知识，即概念之间的逻辑关系或规则知识。事实上，从信息表所表达的说明性知识中发现过程性知识（规则知识）就是知识发现的研究内容。在对信息表进行进一步论述之前，我们先来看几个信息表知识表达系统的例子。

例 2.2 表 2.2给出了一个关于玩具积木的信息表。

根据这个信息表，我们可以得到有关的概念描述，如将玩具积木按照颜色、形状和大小可以分别进行如下分类，得到有关玩具积木的概念知识：

$$U/R_1=\{\{x_1,x_3,x_7\},\{x_2,x_4\},\{x_5,x_6,x_8\}\};$$

$$U/R_2=\{\{x_1,x_5\},\{x_2,x_6\},\{x_3,x_4,x_7,x_8\}\};$$

$$U/R_3=\{\{x_1,x_3,x_4,x_5,x_6\},\{x_2,x_7,x_8\}\}。$$

这里，信息表中所包含的属性集只有对对象（积木）进行描述的属性。

表 2.2

样本集	颜色(R_1)	形状(R_2)	大小(R_3)
x_1	Red	Round	Small
x_2	Blue	Square	Large
x_3	Red	Triangular	Small
x_4	Blue	Triangular	Small
x_5	Yellow	Round	Small
x_6	Yellow	Square	Small
x_7	Red	Triangular	Large
x_8	Yellow	Triangular	Large

例 2.3　如表 2.1所示的信息表，表示了流感病例的数据信息。有的病例属于流感，有的病例不是流感。论域$U=\{e_1,e_2,e_3,e_4,e_5,e_6\}$，它的属性集可以分为条件属性集$C=\{$头疼，肌肉疼，体温$\}$和结果属性集$D=\{$流感$\}$，属性头疼的值域是$\{$是，否$\}$，肌肉疼的值域是$\{$是，否$\}$，体温的值域是$\{$正常、高、很高$\}$，信息函数将每个对象的属性取值映射到具体的属性值上，例如

$f(e_1,$头疼$)=$“是”，

$f(e_2,$体温$)=$“高”。

从表 2.1 还可以看出，条件属性和结果属性之间还存在一定的关系，如当肌肉疼的属性值为“否”的时候，流感的属性值肯定是“否”，这可以形成诸如规则等形式的过程性知识，这是以后讨论知识获取的时候将要研究的问题。

用信息表来表示知识，我们对系统的实际语意、表中的取值的具体含义内容并不感兴趣。实际上，表中的属性值都是从现实问题中采集得到的，是对客观对象属性的抽象描述。我们以后将把信息表中的属性值仅当作数据来研究。下面，我们再对决策表这种数据

表格知识表达系统作形式化的讨论。

2.3 决策表

决策表是一类特殊而重要的知识表达系统，也是一种特殊的信息表，它表示当满足某些条件时，决策（行为、操作、控制）应当如何进行。决策表可以定义如下：

定义 2.2 一个决策表是一个信息表知识表达系统 $S=\langle U, R, V, f\rangle$，$R=C\cup D$ 是属性集合，子集 C 和 D 分别称为条件属性集和结果属性集，$D\neq\varnothing$。

条件属性 C 和结果属性 D 的等价关系 IND(C) 和 IND(D) 的等价类分别称为条件类和决策类。

一个决策表中的结果属性有时是唯一的，称为单一决策；有时是不唯一的，称为多决策。对于具有多个结果属性的决策表，我们可以通过如下两种方法变换成为单一决策的决策表。

方法一 如果决策表 $S=\langle U, R, V, f\rangle$，$R=C\cup D$ 是属性集合，子集 C 和 D 分别称为条件属性集和结果属性集且 $D=\{d_1, d_2, \cdots, d_n\}$，则可以将该决策表分解成为 n 个不同决策的单一决策表 $\{S_1, S_2, \cdots, S_n\}$，其中 $S_i=\langle U, R_i, V_i, f_i\rangle$，$U$ 是论域，$R_i=C\cup\{d_i\}$ 是属性集合，子集 C 和 $\{d_i\}$ 分别称为条件属性集和结果属性集，$V_i=\bigcup_{r\in R_i}V_r$ 是属性值的集合，V_r 表示属性 $r\in R_i$ 的属性值范围，即属性 r 的值域，$f_i: U\times R_i\to V_i$ 是信息函数。

显然，这种方法得到的每个单一决策表是通过将原决策表中其余决策（结果属性）所对应的列去掉而得到的新的决策表。通常情况下，这样得到的单一决策表中会包含条件属性和决策属性取值完全相同的重复记录，还需要将这些重复记录进一步合并为一个记录。

方法二 如果决策表 $S=\langle U, R, V, f\rangle$，$R=C\cup D$ 是属性集合，子集 C 和 D 分别称为条件属性集和结果属性集且 $D=\{d_1,$

$d_2,\cdots,d_n\}$，则可以构造一个新的决策表 $S'=(U,R',V',f')$，U 是论域，$R'=C\cup\{d\}$，子集 C 和 $\{d\}$ 分别称为条件属性集和新的结果属性集，$V'=\bigcup_{r\in R'}V_r$ 是属性值的集合，V_r 表示属性 $r\in R'$ 的属性值范围，即属性 r 的值域，f'：$U\times R'\to V'$ 是信息函数。这里，结果属性 d 的取值要满足如下要求：

$$\begin{aligned}\forall x\forall y(d(x)=d(y)\Leftrightarrow(&d_1(x)=d_1(y)\\&\wedge d_2(x)=d_2(y)\\&\vdots\\&\wedge d_n(x)=d_n(y)))。\end{aligned}$$

可以看出，这种方法是通过综合多决策表中所有结果属性的取值来形成一个综合决策(结果属性值)，从而将多决策问题转化为单一决策问题的。

在本书以后章节的讨论中，我们对于决策表，将只考虑单一决策表问题，因多决策表通常都是转化为单一决策问题来解决的，这有利于问题的简化和求解。

下面用一个实例来说明多决策表转化为单一决策表的方法。

例 2.4　如表 2.3所示的多决策表，此决策表可以通过方法一转化为两个单一决策表(表 2.4 和表 2.5)，通过方法二可以转化为表 2.6 所示的单一决策表。这里，综合决策 d 与决策 d_1 和 d_2 有如下对应关系：

$$d=1\Leftrightarrow(d_1=+)\wedge(d_2=y);$$
$$d=2\Leftrightarrow(d_1=+)\wedge(d_2=n);$$
$$d=3\Leftrightarrow(d_1=-)\wedge(d_2=y);$$
$$d=4\Leftrightarrow(d_1=-)\wedge(d_2=n)。$$

许多决策问题都可以用决策表来表达，这个工具在决策应用中起着相当重要的作用。例如，用决策表来描述一家医院，决策表的每个实例可能就是病人，条件属性是症状和检测，而决策是病症。每个病人都由检测的结果和症状来表征，而且由医生(专家)根据病症的严重程度来分类。如果用决策表来描述一个工业过程，则

这些实例可以代表在某些特定时刻采集的过程中的样品，条件属性是过程中的参数，而决策是由操作员（专家）采取的行动。

表 2.3 多决策表

样例	c_1	c_2	d	d_2
1	a	1		y
2	b	3		y
3	a	1		n
4	c	2		y
5	b	2		n
6	a	1		y
7	b	3		n

表 2.4 单一决策表

样例	c_1	c_2	d_1
1	a	1	+
2	b	3	+
3	a	1	
4	c	2	
5	b	2	+

表 2.5 单一决策表

样例	c_1	c_2	d_2
1	a	1	y
2	b	3	y
3	a	1	n
4	c	2	y
5	b	2	n
6	b	3	n

表 2.6 单一决策表

样例	c_1	c_2	d
1	a	1	1
2	b	3	1
3	a	1	4
4	c	2	3
5	b	2	2
6	a	1	3
7	b	3	2

第3章　Rough集理论基础

Rough集理论的研究已经历了10多年的时间，无论是在系统理论、计算模型的建立和应用系统的研制开发上，都已取得了很多成果，也建立了一套较为完善的Rough集理论体系。在本章中，我们将综合近年来国际上Rough集理论研究的成果，系统地对Rough集理论进行阐述，建立本书的理论基础。

首先，我们将对应于集合论提出相应的Rough集概念，对其基本概念如上近似、下近似、不分明关系、正域、负域、边界域、Rough等价进行分析，然后对定义在Rough集上的代数运算的性质进行介绍，最后对Rough集关系进行讨论，主要说明Rough集之间的包含和等价关系。

3.1　Rough集的基本概念

现实世界中的信息，通常可以用一个信息表来表示。信息表是一个二维表格，其每一行是一个元组，对应于现实世界中的一个个体。信息表的每一列代表信息空间的一维。信息表中的数据可以是从任意领域，诸如医药、财务或军事等领域中收集的。例如表3.1所示的决策表。

信息表中的每一行称为一个实例（实体、对象），我们标记为$e_1, e_2, e_3, e_4, e_5, e_6$。这些实例的性质是通过对一些变量的赋值体现出来的。如前一章所述，样例的属性集可以分为条件属性和结果属性（决策，也称决策属性）。

表 3.1

个体编号	条件属性			决策
	头疼	肌肉疼	体温	
e_1	是	是	正常	否
e_2	是	是	高	是
e_3	是	是	很高	是
e_4	否	是	正常	否
e_5	否	否	高	否
e_6	否	是	很高	是

前一章介绍的不分明关系是 Rough 集理论的一个关键概念，它通常是和一个属性集合联系在一起的。例如，在表 3.1 中，考虑条件属性头疼和肌肉疼。对于 e_1，e_2，e_3 这三个实例，其条件属性头疼的值都是“是”，条件属性肌肉疼的值也都是“是”，因此，从条件属性头疼和肌肉疼的角度来看，这三个实例是不可分辨的。同样，e_4，e_6 在这两个属性上也是不可分辨的。由此构成的不分明集 $\{e_1, e_2, e_3\}$，$\{e_4, e_6\}$ 和 $\{e_5\}$ 被称为基本集。任意有限多个基本集的并被称之为可定义集。

定义 3.1 令 $X \subseteq U$，当 X 能用属性子集 B 确切地描述（即是属性子集 B 所确定的 U 上的不分明集的并）时，称 X 是 B 可定义的，否则称 X 是 B 不可定义的。B 可定义集也称作 B 精确集，B 不可定义集也称为 B 非精确集或 B Rough 集（在不发生混淆的情况下也简称 Rough 集）。

例 3.1 在表 3.1 所示的决策表中，集合 $\{e_2, e_3, e_4, e_5\}$ 就是条件属性子集 $B=\{$头疼，肌肉疼$\}$ 不可定义的，是 B Rough 集，因为根据条件属性子集 B，样例 e_1 和 e_2，e_3 是不可分辨的，e_6 和 e_4 是不可分辨的。我们不能根据条件属性子集 B 来对所有实例是否属于集合 $\{e_2, e_3, e_4, e_5\}$ 作精确判定。但是，如果样例的属性取值是头疼＝“否”，肌肉疼＝“否”，则我们可以确定地说该样例属于集合 $\{e_2,$

$e_1, e_3, e_6\}$。

从例 3.1 可以看出，对于一个样例子集，也称为一个概念，根据一个条件属性子集所确定的不分明关系，我们有可能能够准确地判定一些样例是否属于该概念，也有可能不能够判定某些样例是否属于该概念。为了描述这个问题，Rough 集理论采用了上近似集、下近似集的概念。

定义 3.2　对每个概念 X（样例子集）和不分明关系 B，包含于 X 中的最大可定义集和包含 X 的最小可定义集，都是根据 B 能够确定的，前者称为 X 的下近似集（记为 $B_-(X)$），后者称为 X 的上近似集（记为 $B^-(X)$）。

下面再给出上近似集和下近似集的形式化定义。

定义 3.3　给定知识表达系统 $S=(U, R, V, f)$，对于每个子集 $X \subseteq U$ 和不分明关系 B，X 的上近似集和下近似集分别可以由 B 的基本集定义如下：

$$B_-(X)=\bigcup\{Y_i \mid (Y_i \in U/\mathrm{IND}(B) \wedge Y_i \subseteq X)\},$$

$$B^-(X)=\bigcup\{Y_i \mid (Y_i \in U/\mathrm{IND}(B) \wedge Y_i \cap X \neq \varnothing)\},$$

其中，$U/\mathrm{IND}(B)=\{X \mid (X \subseteq U \wedge \forall x \forall y \forall b(b(x)=b(y)))\}$ 是不分明关系 B 对 U 的划分，也是论域 U 的 B 基本集的集合。

上近似集和下近似集的概念也可以通过集合来定义：

$$B_-(X)=\{x \mid (x \in U \wedge [x]_B \subseteq X)\};$$

$$B^-(X)=\{x \mid (x \in U \wedge [x]_B \cap X \neq \varnothing)\}。$$

即当且仅当 $[x]_B \subseteq X$，$x \in B_-(X)$；当且仅当 $[x]_B \cap X \neq \varnothing$，$x \in B^-(X)$。

定义 3.4　集合 $\mathrm{BN}_B(X)=B^-(X) \setminus B_-(X)$ 称为 X 的 B 边界；$\mathrm{POS}_B(X)=B_-(X)$ 称为 X 的 B 正域；$\mathrm{NEG}_B(X)=U \setminus B_-(X)$ 称为 X 的 B 负域。

$B_-(X)$ 是根据知识 B（属性子集 B），U 中所有一定能归入集合 X 的元素构成的集合，即所有包含于 X 的基本集 Y_i 的并。$B^-(X)$ 是根据知识 B，U 中所有一定能和可能能归入集合 X 的元素构成的

集合，即所有与 X 的交不为空集的基本集 Y_i 的并。$\mathrm{BN}_B(X)$ 是根据知识 B，U 中既不能肯定归入集合 X，又不能肯定归入集合 X 的元素构成的集合。正域 $\mathrm{POS}_B(X)$ 是根据知识 B，U 中所有一定能归入集合 X 的元素构成的集合。负域 $\mathrm{NEG}_B(X)$ 是根据知识 B，U 中所有不能确定一定归入集合 X 的元素的集合。边界域 $\mathrm{BN}_B(X)$ 是某种意义上论域的不确定域，边界域中的元素既不能肯定地属于集合 X，也不能肯定地属于 $\overline{X}$。

有了边界域的定义，我们可以得到上近似集、下近似集、正域、边界域之间的如下关系：

$$\begin{aligned} B^-(X) &= \mathrm{POS}_B(X) \cup \mathrm{BN}_B(X) \\ &= B^-(X) \cup \mathrm{BN}_B(X) \\ &= U \setminus B^-(\overline{X})。 \end{aligned}$$

这几个集合的基数之间存在如下关系：

$$|U \setminus \mathrm{BN}_B(X)| = |U| - |B^-(X) \setminus B^-(X)|。$$

例 3.2 在表 3.1 所示的决策表中，对于属性子集 $B=\{$头疼，肌肉疼$\}$，集合 $X=\{e_2,e_4,e_5\}$ 是一个 B Rough 集，下面分别计算集合 X 的上近似集、下近似集、正域、边界域。

首先计算论域 U 的所有 B 基本集，

$$U|\mathrm{IND}(B)=\{\{e_1,e_2,e_3\},\{e_4,e_6\},\{e_5\}\},$$

令 $B_1=\{e_1,e_2,e_3\}$，$B_2=\{e_4,e_6\}$，$B_3=\{e_5\}$，集合 X 与基本集有如下关系：

$$X \cap B_1=\{e_2,e_3\} \neq \varnothing,$$

$$X \cap B_2=\varnothing,$$

$$X \cap B_3=B_3=\{e_5\} \neq \varnothing。$$

由此可得集合 X 的上近似集、下近似集、正域、边界域：

$$B^-(X)=B_1 \cup B_3=\{e_1,e_2,e_3,e_5\},$$

$$B^-(X)=B_3=\{e_5\},$$

$$\mathrm{POS}_B(X)=B_-(X)=\{e_5\},$$

$$\mathrm{BN}_B(X)=B_1=\{e_1,e_2,e_3\}。$$

集合 X 的上近似集、下近似集还可以形象化表示为图 3.1 的

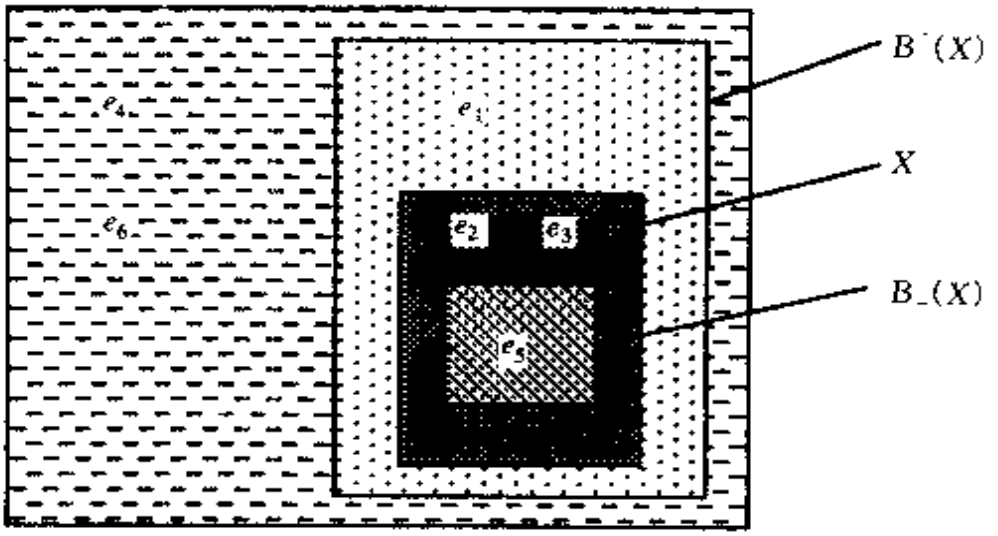


图 3.1　X 的上近似集、下近似集

形式。

根据上近似集、下近似集的定义，不难得到如下命题：

命题 3.1　(1) 当且仅当 $B^-(X)=B(X)$，称集合 X 是 B 可定义集；

(2) 当且仅当 $B^-(X)\neq B(X)$，称集合 X 是 B Rough 集。

对于边界域中的元素，它们和集合 X 之间的关系也不是完全一样的。也就是说，对于不能确定是否肯定属于或肯定不属于集合 X 的元素，我们还可以对它们属于或不属于集合 X 的确定度进行研究；而且，对于一个 B Rough 集，我们也可以定义其 B Rough 度。

3.2　Rough 度与分类质量

在 3.1 节中，我们讨论了 Rough 集的一些基本概念。Rough 集的不可定义性(不确定性)是由于 Rough 集 X 的边界不确定引起的。集合 X 的边界区域越大，其确定性程度就越小。我们可以用集合 X 的精度和 Rough 度这两个概念来描述 Rough 集 X 的不确定性程度。

定义 3.5 假定集合 X 是论域 U 上的一个关于知识 B 的 Rough 集，定义其 B 精度（在不发生混淆的情况下，也简称精度）为

$$d_B(X)=|B_-(X)|/|B^-(X)|。$$

其中，$X\neq\varnothing$；如果 $X=\varnothing$，可定义 $d_B(X)=1$。

由此可见，Rough 集 X 的精度是一个区间[0,1]上的实数，它定义了 Rough 集 X 的可定义程度，即集合 X 的确定度。

定义 3.6 假定集合 X 是论域 U 上的一个关于知识 B 的 Rough 集，定义其 B Rough 度（在不发生混淆的情况下，也简称 Rough 度）为

$$P_B(X)=1-d_B(X)。$$

X 的 Rough 度与精度恰恰相反，表示的是集合 X 的知识的不完全程度。

根据 Rough 集 X 的上近似集、下近似集的特征，我们对 Rough 集 X 的不确定程度也可以作如下定义：

定义 3.7 假定集合 X 是论域 U 上的一个关于知识 B 的 Rough 集，

（1）如果 $B_-(X)\neq\varnothing$ 且 $B^-(X)\neq U$，则称 X 为 B Rough 可定义的；

（2）如果 $B_-(X)=\varnothing$ 且 $B^-(X)\neq U$，则称 X 为 B 内不可定义的；

（3）如果 $B_-(X)\neq\varnothing$ 且 $B^-(X)=U$，则称 X 为 B 外不可定义的；

（4）如果 $B_-(X)=\varnothing$ 且 $B^-(X)=U$，则称 X 为 B 全不可定义的。

对定义 3.7，我们可以作如下的直观理解。

当 $B_-(X)=B^-(X)$ 时，集合 X 的边界域为空，即根据属性集 B 就可以完全肯定地判定任何元素是否属于集合 X，即 X 所对应的概念是一个确定的概念。对于 Rough 集，由于边界域的存在，导

致部分元素不能够被确定地判定。如果 X 为 B Rough 可定义的，意味着我们可以确定 U 中的部分元素是否属于 X 或 $\overline{X}$；如果 X 为 B 内不可定义的，意味着我们可以确定 U 中的部分元素是否属于 $\overline{X}$，但不能确定 U 中的任一元素是否属于 X；如果 X 为 B 外不可定义的，意味着我们可以确定 U 中的部分元素是否属于 X，但不能确定 U 中的任一元素是否属于 X；如果 X 为 B 全不可定义的，意味着我们不能确定 U 中的任一元素是否属于 X 或 $\overline{X}$。

定理 3.1　$\overline{X}$ 是 Rough 可定义(全不可定义)的，当且仅当 X 是 Rough 可定义(全不可定义)的；$\overline{X}$ 是内不可定义(外不可定义)的，当且仅当 X 是外不可定义(内不可定义)的。

对于例 3.2 中的 Rough 集 $X=\{e_2,e_3,e_5\}$，$B=\{$头疼，肌肉疼$\}$，X 就是 Rough 可定义的，因为

$$B^-(X)=B_1\cup B_3=\{e_1,e_2,e_3,e_4\}\neq U,$$

$$B_-(X)=B_3=\{e_5\}\neq\varnothing。$$

X 的精度和 Rough 度分别为

$$\mathrm{d}_B(X)=|B_-(X)|/|B^-(X)|=1/4=0.25,$$

$$\mathrm{P}_B(X)=1-\mathrm{d}_B(X)=0.75。$$

如果 $X=\{e_2,e_3\}$，则 X 就是内不可定义的，因为

$$B^-(X)=B_1=\{e_1,e_2,e_3\}\neq U,$$

$$B_-(X)=\varnothing。$$

X 的精度和 Rough 度则分别为

$$\mathrm{d}_B(X)=|B^-(X)|/|B^-(X)|=0/4=0,$$

$$\mathrm{P}_B(X)=1-\mathrm{d}_B(X)=1。$$

在论域 U 中，如果知道由集合簇 $F=\{X_1,X_2,\cdots,X_n\}(U=\bigcup_{i=1}^{n}X_i)$ 所定义的知识，我们也可以定义属性子集 B 描述这些知识的能力，即 B 对 F 完成分类的准确度。定义下面两个度量来对属性子集 B 的近似分类能力进行描述。

定义 3.8　设集合簇 $F=\{X_1,X_2,\cdots,X_n\}(U=\bigcup_{i=1}^{n}X_i)$ 是论域

U 上定义的知识，B 是一个属性子集，定义 B 对 F 近似分类的精度 $\mathrm{d}_B(F)$ 为

$$\mathrm{d}_B(F)=\sum_{i=1}^{n}|B(X_i)|/\sum_{i=1}^{n}|B(X_i)|。$$

定义 3.9 设集合簇 $F=\{X_1,X_2,\cdots,X_n\}(U=\bigcup_{i=1}^{n}X_i)$ 是论域 U 上定义的知识，B 是一个属性子集，定义 B 对 F 近似分类的质量 $\mathrm{r}_B(F)$ 为

$$\mathrm{r}_B(F)=\sum_{i=1}^{n}|B(X_i)|/|U|。$$

B 对 F 近似分类的精度描述的是当使用知识 B(属性子集 B)对对象进行分类时，在所有可能的决策中确定决策所占的比例；B 对 F 近似分类的质量是应用知识 B 对对象进行分类时，能够确定决策的对象在论域中所占的比例。

定理 3.2 设集合簇 $F=\{X_1,X_2,\cdots,X_n\}(U=\bigcup_{i=1}^{n}X_i,n>1)$ 是论域 U 上定义的知识，B 是一个属性子集。若存在 $i\in\{1,2,\cdots,n\}$ 使得 $B(X_i)\neq\varnothing$，则对于任意 $j(j\neq i,j\in\{1,2,\cdots,n\})$ 都有 $B(X_j)\neq U$。

证明 因 $B(X_i)\neq\varnothing$，则存在 $x(x\in X_i)$ 使得 $[x]_B\subseteq B(X_i)$，而 $B(X_i)\subseteq X_i$，这意味着对于任意 $j(j\neq i,j\in\{1,2,\cdots,n\})$，都有 $[x]_B\cap X_j=\varnothing$。这说明存在 x 使得 $x\notin B(X_j)$，即 $B(X_j)\neq U$。

定理 3.3 设集合簇 $F=\{X_1,X_2,\cdots,X_n\}(U=\bigcup_{i=1}^{n}X_i,n>1)$ 是论域 U 上定义的知识，B 是一个属性子集。若存在 $i\in\{1,2,\cdots,n\}$ 使得 $B(X_i)=U$，则对于任意 $j(j\neq i,j\in\{1,2,\cdots,n\})$ 都有 $B(X_j)=\varnothing$。

证明 $B(X_i)=U$，说明对于任意 x 都有 $[x]_B\cap X_i\neq\varnothing$，而对于任意 $j(j\neq i,j\in\{1,2,\cdots,n\})$，都有 $X_i\cap X_j=\varnothing$，因此有

$\neg([x]_B \subseteq X_1)$，这说明 x 在 $B_-(X_1)$，即 $B_-(X_1)$ 中没有任何元素，故 $B_-(X_1)=\varnothing$。

例 3.3　在表 3.1 所示的决策表中，若 $F=\{X_1, X_2\}$，$X_1=\{e_2, e_3, e_4, e_5\}$，$X_2=\{e_1, e_6\}$，对于属性子集 $B=\{$头疼，肌肉疼$\}$，论域 U 的所有 B 基本集为 $\{\{e_1, e_2, e_3\}, \{e_4, e_6\}, \{e_5\}\}$，令 $B_1=\{e_1, e_2, e_3\}$，$B_2=\{e_4, e_6\}$，$B_3=\{e_5\}$，有

$B_-(X_1)=\{e_5\}\neq\varnothing$，

$B_-(X_2)=\varnothing$，

$B^-(X_1)=\{e_1, e_2, e_3, e_4, e_5, e_6\}=U$，

$B^-(X_2)=\{e_1, e_2, e_4, e_4, e_6\}$，

$\mathrm{d}_B(F)=(1+0)/(6+5)=0.09$，

$\mathrm{r}_B(F)=(1+0)/6=0.17$。

3.3　Rough 集代数性质

与初等集合论相似，上近似集和下近似集也有一些类似的代数性质，这里作一个简单介绍。

命题 3.2

(1) $B_-(X)\subseteq X\subseteq B^-(X)$，

(2) $B_-(\varnothing)=B^-(\varnothing)=\varnothing$，　$B_-(U)=B^-(U)=U$，

(3) $B^-(X\cup Y)=B^-(X)\cup B^-(Y)$，

(4) $B_-(X\cap Y)=B_-(X)\cap B_-(Y)$，

(5) $X\subseteq Y \Rightarrow B_-(X)\subseteq B_-(Y)$，

(6) $X\subseteq Y \Rightarrow B^-(X)\subseteq B^-(Y)$，

(7) $B_-(X\cup Y)\supseteq B_-(X)\cup B_-(Y)$，

(8) $B^-(X\cap Y)\subseteq B^-(X)\cap B^-(Y)$，

(9) $B_-(\sim X)=\sim B^-(X)$，

(10) $B^-(\sim X)=\sim B_-(X)$，

(11) $B(B_-(X))=B(B(X))=B_-(X)$,

(12) $B^-(B(X))=B_-(B(X))=B(X)$。

下面对近似集的这些性质进行证明.

(1) **证明** 对于任意元素 x,如果 $x\in B(X)$,根据下近似集的定义知$[x]_B\subseteq X$。又根据划分的定义知 $x\in[x]_B$,所以有 $x\in X$。因此,根据集合子集的定义可得 $B_-(X)\subseteq X$。

对于任意元素 x,如果 $x\in X$,则$[x]_R\cap X\neq\varnothing$。根据上近似集的定义可知 $x\in B^-(X)$,因此,根据集合子集的定义可得 $X\subseteq B(X)$。

(2) **证明** 由(1)知:$B_-(\varnothing)\subseteq\varnothing$且$\varnothing\subseteq B_-(\varnothing)$(空集是任意集合的子集),所以 $B(\varnothing)=\varnothing$。

假设 $B(\varnothing)\neq\varnothing$,即存在 x 且 $x\in B(\varnothing)$。根据上近似集的定义可得$[x]_B\cap\varnothing\neq\varnothing$,这与$[x]_R\cap\varnothing=\varnothing$矛盾。所以,$B(\varnothing)=\varnothing$。

由(1)知 $B(U)\subseteq U$,如果元素 $x\in U$,因$[x]_B\subseteq U$,根据下近似集的定义可得 $x\in B(U)$。根据集合子集的定义可得 $U\subseteq B_-(U)$。由此可知,$B(U)=U$。

由(1)知$U\subseteq B^-(U)$,而由上近似集的定义知 $B^-(U)\subseteq U$,因此,$B^-(U)=U$。

(3) **证明**

$$\begin{aligned}x\in B(X\cup Y)&\Leftrightarrow[x]_B\cap(X\cup Y)\neq\varnothing\\&\Leftrightarrow([x]_B\cap X)\cup([x]_B\cap Y)\neq\varnothing\\&\Leftrightarrow([x]_B\cap X\neq\varnothing)\vee([x]_B\cap Y\neq\varnothing)\\&\Leftrightarrow x\in B^-(X)\vee x\in B^-(Y)\\&\Leftrightarrow x\in B^-(X)\cup B^-(Y),\end{aligned}$$

因此,$B^-(X\cup Y)=B^-(X)\cup B^-(Y)$。

(4) **证明**

$$\begin{aligned}x\in B(X\cap Y)&\Leftrightarrow[x]_B\subseteq(X\cap Y)\\&\Leftrightarrow([x]_B\subseteq X)\wedge([x]_B\subseteq Y)\end{aligned}$$

$$\Leftrightarrow (x\in B_-(X))\wedge(x\in B_-(Y))$$
$$\Leftrightarrow x\in B_-(X)\cap B_-(Y),$$

因此，$B_-(X\cap Y)=B_-(X)\cap B_-(Y)$。

(5) **证明**　对于元素 $x\in B_-(X)$，有 $x\in[x]_B\subseteq X$，又因 $X\subseteq Y$，故 $x\in[x]_B\subseteq Y$，由下近似集的定义知 $x\in B_-(Y)$。因此，根据集合子集的定义可得 $B_-(X)\subseteq B_-(Y)$。

(6) **证明**　对于元素 $x\in B^-(X)$，有 $[x]_B\cap X\neq\varnothing$，又因 $X\subseteq Y$，$[x]_B\cap Y\neq\varnothing$，由上近似集的定义知 $x\in B^-(X)$。因此，根据集合子集的定义可得 $B^-(X)\subseteq B^-(Y)$。

(7) **证明**　因 $X\subseteq X\cup Y$ 且 $Y\subseteq X\cup Y$，由(5)得

$$(B_-(X)\subseteq B_-(X\cup Y))\wedge(B_-(Y)\subseteq B_-(X\cup Y)).$$

所以，$B_-(X\cup Y)\supseteq B_-(X)\cup B_-(Y)$。

(8) **证明**　因 $(X\cap Y\subseteq X)\wedge(X\cap Y\subseteq Y)$，根据(6)得

$$(B^-(X\cap Y)\subseteq B^-(X))\wedge(B^-(X\cap Y)\subseteq B^-(Y)),$$

所以，$B^-(X\cap Y)\subseteq B^-(X)\cap B^-(Y)$。

(9) **证明**　$x\in B_-(\overline{X})\Leftrightarrow[x]_B\subseteq\overline{X}$
$$\Leftrightarrow[x]_B\cap X=\varnothing$$
$$\Leftrightarrow x\notin B^-(X)$$
$$\Leftrightarrow x\in\overline{B^-(X)},$$

所以，$B_-(\overline{X})=\overline{B^-(X)}$。

(10) **证明**　$x\in B^-(\overline{X})\Leftrightarrow[x]_B\cap\overline{X}\neq\varnothing$
$$\Leftrightarrow\neg([x]_B\subseteq X)$$
$$\Leftrightarrow\neg(x\in B_-(X))$$
$$\Leftrightarrow x\notin B_-(X)$$
$$\Leftrightarrow x\in\overline{B_-(X)},$$

所以，$B^-(\overline{X})=\overline{B_-(X)}$。

(11) **证明**　由(1)可得 $B_-(B_-(X))\subseteq B_-(X)$。

$$x\in B_-(X)\Rightarrow[x]_B\subseteq B_-(X)$$
$$\Rightarrow[x]_B\subseteq B(B(X)),$$

因此，$B_-(X)\subseteq B_-(B_-(X))$，故 $B_-(B_-(X))=B_-(X)$。

又，由(1)可得 $B_-(X)\subseteq B^-(B_-(X))$。

$$x\in B^-(B_-(X)) \Rightarrow [x]_B\cap B_-(X)\neq\varnothing$$
$$\Rightarrow [x]_B\subseteq B_-(X)$$
$$\Rightarrow [x]_B\subseteq X$$
$$\Rightarrow x\in B_-(X),$$

因此，$B^-(B_-(X))\subseteq B_-(X)$，故 $B_-(B^-(X))=B_-(X)$。

(12) **证明** 由(1)可得 $B_-(X)\subseteq B^-(B_-(X))$。

$$x\in B^-(B_-(X)) \Rightarrow [x]_B\cap B^-(X)\neq\varnothing$$
$$\Rightarrow [x]_B\subseteq B_-(X)$$
$$\Rightarrow x\in B^-(X),$$

因此，$B^-(B_-(X))\subseteq B^-(X)$，故 $B^-(B^-(X))=B^-(X)$。

又，由(1)可得 $B^-(B_-(X))\subseteq B^-(X)$。

$$x\in B_-(X) \Rightarrow [x]_B\subseteq B_-(X)$$
$$\Rightarrow x\in B^-(B_-(X)),$$

因此，$B_-(X)\subseteq B^-(B_-(X))$，故 $B_-(B^-(X))=B_-(X)$。

3.4 Rough 集关系

显然，初等集合论中的概念和 Rough 集之间的概念是有很大差别的，Rough 集是用上近似集和下近似集来描述集合的不确定性，而上近似集和下近似集却是边界域为空的精确集。为了以后讨论知识发现问题的需要，我们这里再对 Rough 集之间的关系进行说明。主要讨论包含和等价关系。

定义 3.10 X 和 Y 是论域 U 中的两个集合，B 为知识(属性子集)，定义

(1) 若 $B_-(X)\subseteq B_-(Y)$，则称集合 X 为 B 下包含于 Y(在不发生混淆的情况下，也可简称为下包含)，或者 Y B 下包含 X，记作 $X\subseteq_- Y$；

(2) 若 $B(X)\subseteq B(Y)$，则称集合 X 为 B 上包含于 Y(在不发生混淆的情况下，也可简称为上包含)，或者 Y B 上包含 X，记作 $X\subseteq Y$；

(3) 若 $X\underline{\subseteq} Y$ 且 $X\overline{\subseteq} Y$，则称集合 X 为 B Rough 包含于 Y(在不发生混淆的情况下，也可简称为 Rough 包含)，或者 Y B Rough 包含 X，记作 $X\subseteq Y$。

集合 X 为 B 下包含于 Y 意味着 X 的正例(X 的下近似集中的元素)同样是 Y 的正例；集合 X 为 B 上包含于 Y 意味着 Y 的负例($\overline{Y}$ 的下近似集中的元素)同样是 X 的负例。显然，Rough 包含不同于包含。

定理 3.4　Rough 集的包含关系具有如下性质：

(1) $X\subseteq Y\Rightarrow(X\underline{\subseteq} Y)\wedge(X\overline{\subseteq} Y)\wedge(X\subseteq Y)$；

(2) $(X'\underline{\subseteq} X)\wedge(Y'\underline{\subseteq} Y)\Rightarrow(X'\cup Y')\underline{\subseteq}(X\cup Y)$；

(3) $(X'\overline{\subseteq} X)\wedge(Y'\overline{\subseteq} Y)\Rightarrow(X'\cap Y')\overline{\subseteq}(X\cap Y)$。

定义 3.11　X 和 Y 是论域 U 中的两个集合，B 为知识(属性子集)，定义

(1) 若 $B(X)=B(Y)$，则称集合 X 为 B 下等价于 Y(在不发生混淆的情况下，也可简称为下等价)，记作 $X=Y$；

(2) 若 $\overline{B}(X)=\overline{B}(Y)$，则称集合 X 为 B 上等价于 Y(在不发生混淆的情况下，也可简称为上等价)，记作 $X=\overline{Y}$；

(3) 若 $X=\underline{Y}$ 且 $X=\overline{Y}$，则称集合 X 为 B Rough 等价于 Y(在不发生混淆的情况下，也可简称为 Rough 等价)，记作 $X=\overline{\underline{Y}}$。

$X=\underline{Y}$ 说明集合 X 和 Y 具有相同的正例集；$X=\overline{Y}$ 说明集合 X 和 Y 具有相同的负例集。同样，集合的 Rough 等价也不同于集合的等价。

定理 3.5　Rough 集的等价关系具有如下性质：

(1) $(X\cap Y=\underline{X})\wedge(X\cap Y=\underline{Y})\Rightarrow X=Y$；

(2) $(X\cup Y=\overline{X})\wedge(X\cup Y=\overline{Y})\Rightarrow X=Y$；

(3) $(X=\overline{X'})\wedge(Y=Y')\Rightarrow(X\cup Y)=(X'\cup Y')$；

(4) $(X=_{R} X')\wedge(Y=_{R} Y')\Rightarrow(X\cap Y)=_{R}(X'\cap Y')$;

(5) $(X=_{R} Y)\Rightarrow X\cup Y=_{R} U$;

(6) $(X\neq_{R} Y)\Rightarrow X\cap Y=_{R}\varnothing$;

(7) $(X\subseteq_{R} Y)\wedge(Y=_{R}\varnothing)\Rightarrow X=_{R}\varnothing$;

(8) $(X\subseteq_{R} Y)\wedge(X=_{R} U)\Rightarrow Y=_{R} U$;

(9) $X\neq_{R} Y\Leftrightarrow X=_{R} Y$;

(10) $(X=_{R}\varnothing)\vee(Y=_{R}\varnothing)\Rightarrow(X\cap Y)=_{R}\varnothing$;

(11) $(X=_{R} U)\vee(Y=_{R} U)\Rightarrow(X\cup Y)=_{R} U$。

定理 3.6 Rough 集的包含关系和等价关系之间具有如下性质：

(1) $(X\subseteq_{R} Y)\wedge(Y\subseteq_{R} X)\Rightarrow X=_{R} Y$;

(2) $(X\subseteq_{R} Y)\wedge(Y\subseteq_{R} X)\Rightarrow X=_{R} Y$;

(3) $(X\subseteq_{R} Y)\wedge(Y\subseteq_{R} X)\Rightarrow X=_{R} Y$;

(4) $(X\cup Y)=_{R} Y\Leftrightarrow X\subseteq_{R} Y$;

(5) $(X\cap Y)=_{R} Y\Leftrightarrow X\subseteq_{R} Y$;

(6) $(X\subseteq_{R} Y)\wedge(X=_{R} X')\wedge(Y=_{R} Y')\Rightarrow X'\subseteq_{R} Y'$;

(7) $(X\subseteq_{R} Y)\wedge(X=_{R} X')\wedge(Y=_{R} Y')\Rightarrow X'\subseteq_{R} Y'$;

(8) $(X\subseteq_{R} Y)\wedge(X=_{R} X')\wedge(Y=_{R} Y')\Rightarrow X'\subseteq_{R} Y'$;

(9) $(X\subseteq_{R} Y)\wedge(X=_{R} Z)\Rightarrow Z\subseteq_{R} Y$;

(10) $(X\subseteq_{R} Y)\wedge(X=_{R} Z)\Rightarrow Z\subseteq_{R} Y$;

(11) $(X\subseteq_{R} Y)\wedge(X=_{R} Z)\Rightarrow Z\subseteq_{R} Y$。

在讨论 Rough 集时，元素的成员关系(即一个元素是否属于一个集合)或者集合之间的包含和等价关系，都不同于初等集合中的概念，它们都是基于不分明关系的。一个元素是否属于某一集合，要根据我们对该元素的了解程度而定，和该元素所对应的不分明关系有关，不能仅仅依据该元素的属性值来简单判定。集合之间的等价和包含关系，也是根据知识 B 来判定的，在某一知识的情况下两集合可能具有包含(或等价)关系，但当知识变化为另一知识的情况下(我们对论域的了解发生了变化)，两集合之间的包含

(或等价)关系也会发生变化,原来不具有的关系可能具有了,原来具有的关系也可能失去了。

正是由于 Rough 集之间的这种丰富的关系,才使得它成为知识发现、数据挖掘研究的重要理论工具。我们将在下面的章节中对这个问题进行深入讨论。

3.5　可变精度 Rough 集模型

自 20 世纪 80 年代初期 Pawlak 教授提出 Rough 集概念以来,在很多领域取得了成功。但是,它还不能用于不确定信息建模,这严重影响了 Rough 集理论的应用。在前面介绍的 Rough 集理论中,我们是用精确集合包含来定义上近似集和下近似集的,在实际应用中,缺乏对噪音数据的适应能力。然而,现实处理的数据大多是不精确的。基于这个考虑,Ziarko 提出了一种可变精度的 Rough 集模型(VP- RS)。

在可变精度 Rough 集模型中,定义了下面的条件概率:

$$P(D_j|[u_i]_{\mathrm{IND}}) = \frac{P(D_j \cap [u_i]_{\mathrm{IND}})}{P([u_i]_{\mathrm{IND}})} = \frac{\mathrm{card}(D_j \cap [u_i]_{\mathrm{IND}})}{\mathrm{card}([u_i]_{\mathrm{IND}})}。$$

定义 3.12　给定一个决策表,假定由条件属性集合 C 导出的等价类为 $\mathrm{IND}(C)=\tilde{C}=\{\tilde{C}_1,\tilde{C}_2,\cdots,\tilde{C}_k\}$,$\beta$ 是依赖于数据中噪音程度的一个取值在$[0,0.5)$上的数,则

(1) β 正域定义为 $\mathrm{POS}_C(D_j)=\bigcup_{P(D_j|\tilde{C}_i)\geqslant 1-\beta}\{\tilde{C}_i\in\tilde{C}\}$;

(2) β 边界域定义为 $\mathrm{BOS}_C(D_j)=\bigcup_{\beta<P(D_j|\tilde{C}_i)<1-\beta}\{\tilde{C}_i\in\tilde{C}\}$;

(3) β 负域定义为 $\mathrm{NEG}_C(D_j)=\bigcup_{P(D_j|\tilde{C}_i)\leqslant\beta}\{\tilde{C}_i\in\tilde{C}\}$。

定义 3.13　条件属性集 C 和决策属性 D 之间的相关程度定义为

$$K_\beta(C,D)=\frac{\mathrm{card}(\mathrm{POS}_C(D)\bigcup\mathrm{NEG}_C(D))}{\mathrm{card}(U)}。$$

其中：$\mathrm{POS}_C(D)=\sum_i\mathrm{POS}_C(D_i)$，$\mathrm{NEG}_C(D)=\sum_i\mathrm{NEG}_C(D_i)$。

$K_\beta(C,D)$是决策表中能够 Rough 地或确定地划分到 β 正域和 β 负域的样本的百分比。

3.6 不完备信息系统中 Rough 集理论的扩充

基于传统不分明关系的 Rough 集理论是不能处理不完备信息系统的。本节中，我们将对传统的 Rough 集理论进行三种扩充：一是基于容差关系的扩充；二是基于非对称相似关系的扩充；三是基于量化容差关系的扩充。

3.6.1 不完备信息系统的特点

传统的 Rough 集理论是一种能够对仅仅概略地描述的对象进行分类的工具。即使个体对象是不相同的，也可以根据得到的关于对象的部分信息进行分类。也就是说，至少在所考虑的属性集范围内，不同个体对象可能具有相同或者相似的描述。这样的属性集合可以视作在给定知识条件下能够描述周围世界的可能元素。在传统 Rough 集理论中，存在一个明显的假设，即所有可以获得的个体对象由这个属性集合给出完全的描述。换句话说，用$U=\{a_1,a_2,\cdots,a_n\}$表示个体对象集合，$C=\{c_1,c_2,\cdots,c_m\}$表示属性集合，则对于任意$a_j\in U$，$c_i\in C$，属性值$c_i(a_j)$总是存在的，即$c_i(a_j)\neq\varnothing$。

这个假设虽然是合理的，但与很多现实情况有差异。在这些情况下，由于不可能得到一部分属性值（例如，如果集合U是关于病员的集合，属性是一些临床检验，并非所有的检验结果在给定时间内都是可以得到的），或者有些对象的某个属性值是肯定不可能得到的，这导致关于对象集合U的描述是不完全的。这样，就导致了不完备信息系统的出现（当然，导致这种情况的原因还可以有很

多，诸如由于存储介质的故障、传输媒体的故障、一些人为因素等）。本节中，我们就不完备信息系统中 Rough 集理论的扩充问题进行讨论。

对于不完备信息的理解，存在两种语意解释：遗漏（missing）语意和缺席（absent）语意。遗漏语意下，认为遗漏值（未知值）将来总是可以得到的，并可以与任意值相比较（匹配，相等）；而缺席语意下，认为缺席值（未知值）是无法再得到的，不能与任一值相比较（匹配，相等）。我们接下来讨论处理不完备信息表的三种关系：容差关系（tolerance relation）、非对称相似关系（non symmetric similarity relation）和量化容差关系（valued tolerance relation）。

3.6.2　容差关系

在 M. Kryszkiewicz 提出的容差关系中，最主要的一个概念是赋予信息表中没有值的元素一个“Null”值，“Null”值是一种任何值都有可能的值。这个解释与这样的值仅仅是被遗漏但又确实存在的解释相对应。换句话说，就是由于不精确的知识迫使我们去处理只有部分信息的不完备信息表。各个体对象具有潜在的完备信息，而我们当前只是遗漏了这些值。

给定信息表 $S=(U,C,V,f)$，对于具有遗漏属性值的属性子集 $B\subseteq C$，记遗漏值为“*”，我们引入如下容差关系 T 的定义。

定义 3.14　容差关系 T 的定义为：

$$T=\{(x,y)\mid x\in U\wedge y\in U\wedge\forall c_j(c_j\in B\Rightarrow (c_j(x)=c_j(y)\vee c_j(x)=*\vee c_j(y)=*))\}。$$

显然，T 是自反和对称的，但不一定是传递的。进一步，用符号 $I_B(x)$ 表示在属性集合 B 上满足关系 $T(x,y)$ 的个体对象 y 的集合，即对象 x 的容差类。基于容差类的定义，我们可以定义不完备信息表中对象集合 X 关于属性集 $B\subseteq C$ 的上近似和下近似。

定义 3.15　不完备信息表 $S=(U,C,V,f)$ 中对象集合 X 关于属性集 $B\subseteq C$ 的上近似（X^B）和下近似（X_B）分别为

$$X_R=\{x\in U\mid I_R(x)\subseteq X\},$$
$$X^R=\{x\in U\mid I_R(x)\cap X\neq\varnothing\}。$$

显然，$X^R=\bigcup_{x\in X}I_R(x)$。

下面用一个不完备信息表实例来说明。

例 3.4 对于如表 3.2所示的信息表，其中 $a_1,a_2,\cdots,a_{12}$是对象集合，c_1,c_2,c_3,c_4 是条件属性集合，值域均为$[0,1,2,3]$，d 是决策属性，将对象分为 Ψ 和 Φ 两个集合(决策类)。

表 3.2 不完备信息表

A	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}
c_1	3	2	2	*	*	2	3	*	3	1	*	3
c_2	2	3	3	2	2	3	*	0	2	*	2	2
c_3	1	2	2	*	*	2	*	0	1	*	*	1
c_4	0	0	0	1	1	1	3	*	3	*	*	*
d	Φ	Φ	Ψ	Φ	Ψ	Ψ	Φ	Ψ	Ψ	Φ	Ψ	Φ

用容差关系来分析这个信息表，可得结果：

$I_C(a_1)=\{a_1,a_{11},a_{12}\}$，

$I_C(a_2)=\{a_2,a_3\}$，

$I_C(a_3)=\{a_2,a_3\}$，

$I_C(a_4)=\{a_4,a_5,a_{10},a_{11},a_{12}\}$，

$I_C(a_5)=\{a_4,a_5,a_{10},a_{11},a_{12}\}$，

$I_C(a_6)=\{a_6\}$，

$I_C(a_7)=\{a_7,a_8,a_9,a_{11},a_{12}\}$，

$I_C(a_8)=\{a_7,a_8,a_{10}\}$，

$I_C(a_9)=\{a_7,a_9,a_{11},a_{12}\}$，

$I_C(a_{10})=\{a_4,a_5,a_8,a_{10},a_{11}\}$，

$I_C(a_{11})=\{a_1,a_4,a_5,a_7,a_9,a_{10},a_{11},a_{12}\}$，

$I_C(a_{12})=\{a_1,a_4,a_5,a_7,a_9,a_{11},a_{12}\}$，

$\Phi_C = \varnothing$，

$\Phi^C = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}\}$，

$\Psi_C = \{a_5\}$，

$\Psi^C = U$。

这样的结果，并不让人满意，甚至有些凭直觉就可以分类的个体对象在这里还不能进行分类。例如 a_1，我们完全知道关于它的信息，而且没有感觉到有其他对象与之相似，然而，a_1 却不在集合 Φ 的下近似中。导致这个结果的原因是对象 a_{11} 和 a_{12} 的遗漏属性值使得它们被认为与对象 a_1 相似。当然，这是"安全"的，因为这两个对象存在潜在的与对象 a_1 相同的属性值。

3.6.3　非对称相似关系

下面，我们再介绍一种基于不一定对称的相似关系概念来处理不完备信息表的方法。这里，我们认为对象可能被不完全描述的原因不仅可能是由于我们的知识不精确，还可能是由于干脆就不可能用所有的属性来描述它们。因此，我们不认为未知值是不确定的，而是当前不存在的，我们不允许比较未知值。

基于这种观点，各对象可能有或多或少的完全描述，这取决于可能采用多少属性。从这种观点看，只要两个对象的已知属性值相同，就可以认为一个个体对象 x 与另一个对象 y 相似。同样记未知值为"*"，对于给定信息表 $S=(U,C,V,f)$，属性子集 $B\subseteq C$，我们引入如下非对称相似关系 S 的定义：

定义 3.16　非对称相似关系 S 的定义为：

$$S=\{(x,y)\mid x\in U\wedge y\in U\wedge\forall c_i(c_i\in B\Rightarrow (c_i(x)=*\vee c_i(x)=c_i(y)))\}。$$

显然，非对称相似关系不对称，但是传递。关系 S 是对象集合 U 上的偏序。实际上，非对称相似关系可以认为是包含关系的一个代表，因为只要 x 的描述包含于 y 的描述就认为 x 与 y 相似。对于任意个体 $x\in U$，可以定义两个非对称相似集合。

定义 3.17 非对称相似于 x 的对象集合 $R(x)$，x 与之非对称相似的对象集合 $R^{-1}(x)$的定义为：

$$R(x)=\{y\in U\mid S(y,x)\},$$

$$R^{-1}(x)=\{y\in U\mid S(x,y)\}。$$

显然，$R(x)$与 $R^{-1}(x)$是两个不相同的集合。对象集合 X 的上近似、下近似可以进一步定义如下：

定义 3.18 不完备信息表 $S=\langle U,C,V,f\rangle$的对象集合 X 关于属性集 $B\subseteq C$ 的上近似(X^B)和下近似(X_B)定义为：

$$X_B=\{x\in U\mid R^{-1}(x)\subseteq X\},$$

$$X^B=\bigcup_{x\in X}R(x),$$

也就是说，如果对象 x 与之非对称相似的对象都包含于 X，则对象 x 肯定属于 X 类；相反，如果对象 x 非对称相似于 X 中的某个对象，则对象 x 可能属于 X 类。比较前一节的容差关系和这里的非对称相似关系，可以得到如下定理：

定理 3.7 给定信息表 $S=\langle U,C,V,f\rangle$和个体对象集合 X，在非对称相似关系下 X 的上近似和下近似是对在容差关系下 X 的上近似和下近似的改进。

证明 用 X_B^T 和 X_T^B 分别表示 X 在容差关系下的下近似和上近似，X_B^S 和 X_S^B 分别表示 X 在非对称相似关系下的下近似和上近似。为了证明这个定理，我们必须说明：$X_B^T\subseteq X_B^S$ 和 $X_S^B\subseteq X_T^B$。由于非对称相似关系 S 所满足的条件是容差关系 T 所满足的条件的子集，所以有$\forall x\forall y\ S(x,y)\rightarrow T(x,y)$。这样，就不难得到 $R(x)\subseteq I(x)$和 $R^{-1}(x)\subseteq I(x)$。

(1) 先证 $X_B^T\subseteq X_B^S$。由定义有 $X_B^T=\{x\in U\mid I(x)\subseteq X\}$和 $X_B^S=\{x\in U\mid R^{-1}(x)\subseteq X\}$，因此，如果对象 x 属于 X_B^T，就有 $I(x)\subseteq X$，又由于 $R^{-1}(x)\subseteq I(x)$，于是有 $R^{-1}(x)\subseteq X$，所以这个 x 一定属于 X_B^S。反过来不一定成立。因此，在非对称相似关系下 X 的下近似集合所包含的对象元素至少与在容差关系下 X 的下近似集合所

包含的元素同样丰富。

(2) 再证 $X_S^R \subseteq X_T^R$。由定义有 $X_S^R = \bigcup_{x \in X} R(x)$ 和 $X_T^R = \bigcup_{x \in X} I(x)$，由于 $R(x) \subseteq I(x)$，所以所有 $R(x)$ 的并集也一定是所有 $I(x)$ 的并集的子集。反过来不一定成立。因此，在非对称相似关系下 X 的上近似集合所包含的对象元素最多与在容差关系下 X 的上近似集合所包含的元素相同。

对于表 3.2，我们有如下的计算结果：

$R^{-1}(a_1) = \{a_1\}$，

$R(a_1) = \{a_1, a_{12}\}$，

$R^{-1}(a_2) = \{a_2, a_3\}$，

$R(a_2) = \{a_2, a_3\}$，

$R^{-1}(a_3) = \{a_2, a_3\}$，

$R(a_3) = \{a_2, a_3\}$，

$R^{-1}(a_4) = \{a_4, a_5\}$，

$R(a_4) = \{a_4, a_5, a_{11}\}$，

$R^{-1}(a_5) = \{a_4, a_5\}$，

$R(a_5) = \{a_4, a_5, a_{11}\}$，

$R^{-1}(a_6) = \{a_6\}$，

$R(a_6) = \{a_6\}$，

$R^{-1}(a_7) = \{a_7, a_9\}$，

$R(a_7) = \{a_7\}$，

$R^{-1}(a_8) = \{a_8\}$，

$R(a_8) = \{a_8\}$，

$R^{-1}(a_9) = \{a_9\}$，

$R(a_9) = \{a_7, a_9, a_{11}, a_{12}\}$，

$R^{-1}(a_{10}) = \{a_{10}\}$，

$R(a_{10}) = \{a_{10}\}$，

$R^{-1}(a_{11}) = \{a_1, a_4, a_5, a_9, a_{11}, a_{12}\}$，

$R(a_{11})=\{a_{11}\}$,

$R^{-1}(a_{11})=\{a_1,a_{11},a_{12}\}$,

$R(a_1)=\{a_{11},a_{12}\}$,

$\Phi_C=\{a_1,a_{10}\}$,

$\Phi^C=\{a_1,a_2,a_3,a_4,a_5,a_7,a_{10},a_{11},a_{12}\}$,

$\Psi_C=\{a_6,a_8,a_9\}$,

$\Psi^C=\{a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9,a_{11},a_{12}\}$。

根据非对称相似关系得到的近似集合比在容差关系下得到的近似集合含有更多的信息。而且，一些直觉上希望分类到 Ψ 和 Φ 的元素也分别包含在下近似集合 Ψ_C 和 Φ_C 中。显然，这种方法没有采用容差关系安全，因为一些对象在知之甚少的情况下被肯定地划分到某类中(如 a_{10})。但是，在缺席值语意下，我们不认为部分描述的对象是知之甚少的，而是在少部分属性上是知道的。

3.6.4 量化容差关系

首先再来看一下表 3.2 中的个体对象 a_1，a_{11} 和 a_{12}。由容差关系和非对称相似关系可以得到 $T(a_{11},a_1)$，$T(a_{12},a_1)$，$S(a_{11},a_1)$ 和 $S(a_{12},a_1)$。然而，凭直觉，我们希望有 a_{12} 比 a_{11} 更像 a_1，或者 a_{11} 没有 a_{12} 那么像 a_1 的结论。这是因为对象 a_{12} 只有一个未知值，其余的值均与 a_1 相等，而对象 a_{11} 有三个未知值，只有一个值与 a_1 相等。我们可以用量化容差关系来描述这种差别。

可以用不同的比较规则来定义不同的量化容差(近似)关系。而且，在完备信息表中同样可以定义量化容差(近似)关系。

给定一个量化容差关系，对于个体对象全集 U 中的每个元素，可以定义容差类的概念，它是一个用关于参考元素的容差度作为成员函数的模糊集合。如果容差度的值取为 1，就得到 3.6.2 节中的容差类概念。下面来定义对象集合 X 的上近似和下近似。给定待描述的集合 X 和对象集合 $Z\subseteq U$，我们来定义集合 Z 为集合 X 的上近似、下近似的程度。基于这种观点，对象全集 U 的每个子

集都可能是集合 X 的不同程度的上近似和下近似。为此，我们需要对通常的逻辑连接词给予函数表示。

定义 3.19　逻辑非（否定）函数 $N:[0,1]\to[0,1]$，要求 $N(0)=1, N(1)=0$。

通常将逻辑非函数表示为 $N(x)=1-x$。

定义 3.20　T norm 是一个连续非降函数 $T:[0,1]^2\to[0,1]$，要求 $T(x,1)=x$。

很明显，T - norm 代表合取。T - norm 通常有三种表示：

- 最小值：$T(x,y)=\min(x,y)$；
- 乘积：$T(x,y)=x\times y$；
- Lukasiewicz T norm：$T(x,y)=\max(x+y-1,0)$。

定义 3.21　T conorm 是一个连续非降函数 $S:[0,1]^2\to[0,1]$，要求 $S(0,y)=y$。

很明显，T - conorm 代表析取。T - conorm 通常有三种表示：

- 最大值：$S(x,y)=\max(x,y)$；
- 乘积：$S(x,y)=x+y-x\times y$；
- Lukasiewicz T - conorm：$S(x,y)=\min(x+y,1)$。

如果 $S(x,y)=N(T(N(x),N(y)))$，德摩根律就成立，我们称这样的 (N,T,S) 三元组为德摩根三元组。

定义 3.22　$I(x,y)$ 是 x 蕴含 y 的程度函数，$I:[0,1]^2\to[0,1]$。

然而，这种函数所满足的这些特性的定义并不是无异议的。下面两个基本特征是我们所想得到的：第一个是 $I(x,y)=S(N(x),y)$ 它解释通常的逻辑等价关系 $x\to y\Leftrightarrow \neg x\vee y$；第二个是只要 x 的真值不比 y 的真值大，就有 x 蕴涵 y 为真，即 $x\leqslant y\Leftrightarrow I(x,y)=1$。要同时满足这两个条件几乎是不可能的。在非常少的条件下，这种情况会发生，但其他一些特征又不能满足。

再回到上近似和下近似的定义上来。

定义 3.23　给定对象集合 $Z\subseteq U$，$X\subseteq U$ 和属性集合 $B\subseteq C$，

通常作如下定义：

(1) $Z=X_R \Leftrightarrow \forall z(z\in Z \rightarrow \Theta(z)\subseteq X)$，

(2) $Z=X^R \Leftrightarrow \forall z(z\in Z \rightarrow \Theta(z)\cap X\neq\varnothing)$。

其中，$\Theta(z)$是对象 z 的近似不分明类，对 $\Theta(z)$的定义可作如下函数解释：

$$\forall xX(x)\overset{\text{def}}{=\!=}T_x X(x),$$

$$\exists xX(x)\overset{\text{def}}{=\!=}S_x X(x),$$

$$X\subseteq Y \overset{\text{def}}{=\!=}T_x(I(\mu_X(x),\mu_Y(x))),$$

$$X\cap Y\neq\varnothing \overset{\text{def}}{=\!=}\exists x(X(x)\wedge Y(x))$$

$$\overset{\text{def}}{=\!=}S_x(T(\mu_X(x),\mu_Y(x))),$$

$$\mu_{X_R}(Z)=T_{z\in Z}(T_{x,\Theta(z)}(I(R(z,x),\dot{x}))),$$

$$\mu_{X^R}(Z)=T_{z\in Z}(S_{x\in\Theta(z)}(T(R(z,x),\dot{x}))),$$

其中 $\mu_{X_R}(Z)$是集合 Z 为 X 的下近似的程度，$\mu_{X^R}(Z)$是集合 Z 为 X 的上近似的程度，$\Theta(z)$是对象 z 的容差类，$R(z,x)$是元素 x 属于元素 z 的容差类的成员隶属度，$\dot{x}$ 是元素 x 属于集合 X 的成员隶属度($\dot{x}\in[0,1]$)。

下面以表 3.2 为例来进行说明。由于各属性可能取值的集合是离散的，我们假设在这些值中存在均匀分布，即，对于信息表 $S=(U,C,V,f)$的属性 $c_j\in C$，其值域为 $E_j=\{e_j^1,\cdots,e_j^m\}$，个体对象 $x\in U$，有 $c_j(x)=e_j^i$ 的概率为 $1/|E_j|$。因此，给定两个对象 $x,y\in A$ 和属性 c_j，如果 $c_j(y)=e_j^i$，则 x 在属性 c_j 上近似于 y 的概率 $R_j(x,y)=1/|E_j|$。以此为基础，我们就可以计算两个对象在属性全集上近似的概率，即两个对象在所有属性上取值相同的联合概率 $R(x,y)=\prod\limits_{c_j\in C}R_j(x,y)$。这样就可以得到表 3.3 所示的量化容差关系。

考察对象 a_1，量化容差关系 $R(a_1,x)$，x 的结果为向量(1,0,

0,0,0,0,0,0,0,0,1/64,1/4),这实际上代表了对象 a_1 的容差类 $\Theta(a_1)$，对象 a_1 的清晰容差类是集合 $\{a_1,a_{11},a_{12}\}$，它对应于向量(1,0,0,0,0,0,0,0,0,0,1,1)。这里，我们对 T,S,I 逻辑函数的选择为：$T(x,y)=x\times y$，$S(x,y)=x+y-x\times y$，$I(x,y)=1-x+x\times y$。这种蕴涵不满足蕴涵的第二个特征($x\leqslant y\Leftrightarrow I(x,y)=1$)。但是，在这种特殊情况下，从模糊集 $\Theta(z)$ 到普通集合 X 存在特殊蕴涵，使得 $\hat{x}\in[0,1]$。采用满足第二个特征的任意蕴涵都将缩小取值到集合{0,1}，$\mu_{X_L}(Z)$ 的取值缩小到集合{0,1}，从而退化成为通常的下近似。基于上述考虑，我们可以得到：

$$\mu_{X_L}(Z)=\prod_{z\in Z}\left(\prod_{x\in\Theta(z)}(1-R(z,x)+R(z,x)\times\hat{x})\right),$$

$$\mu_{X^U}(Z)=\prod_{z\in Z}\left(1-\prod_{x\in\Theta(z)}(1-R(z,x)\times\hat{x})\right)。$$

表 3.3　表 3.2 的量化容差关系

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}
a_1	1	0	0	0	0	0	0	0	0	0	1/64	1/4
a_2	0	1	1	0	0	0	0	0	0	0	0	0
a_3	0	1	1	0	0	0	0	0	0	0	0	0
a_4	0	0	0	1	1/256	0	0	0	0	1/1 024	1/1 024	1/64
a_5	0	0	0	1/256	1	0	0	0	0	1/1 024	1/1 024	1/64
a_6	0	0	0	0	0	1	0	0	0	0	0	0
a_7	0	0	0	0	0	0	1	1/256	1/16	0	1/1 024	1/64
a_8	0	0	0	0	0	0	1/256	1	0	1/1 024	0	0
a_9	0	0	0	0	0	0	1/16	0	1	0	1/64	1/4
a_{10}	0	0	0	1/1 024	1/1 024	0	0	1/1 024	0	1	1/4 096	0
a_{11}	1/64	0	0	1/1 024	1/1 024	0	1/1 024	0	1/64	1/4 096	1	1/256
a_{12}	1/4	0	0	1/64	1/64	0	1/64	0	1/4	0	1/256	1

考察集合 X,Z 和个体对象 a_1，量化容差关系 $R(a_1,x)$ 如上所述，$\hat{x}$ 取值为[1,1,0,1,0,0,1,0,0,1,0,1]，可以得到 $\mu_{X_L}(a_1)=0.98$，而 $\mu_{X^U}(a_1)=1$。按照如下方法，我们可以得到集合 X 的上近

似(下近似)集合 Z:

(1) 选出所有满足如下条件的对象 z:

$\mu(\Theta(z)\rightarrow X)=1$(或者 $\mu(\Theta(z)\cap X)=1$)。

(2) 按照 k 值递减的顺序(如 k 取 0.99,0.98 等等),添加使得 $\mu(\Theta(z)\rightarrow X)\geqslant k$(或者 $\mu(\Theta(z)\cap X)\geqslant k$)的对象 z,得到具有递减成员函数 $\mu_{X_B}(Z)$(或者 $\mu_{X^B}(Z)$)的一簇下近似(或者上近似)。

(3) 固定一个最小成员程度值 λ,得到满足条件 $\mu_{X_B}(Z)\geqslant\lambda$(或者 $\mu_{X^B}(Z)\geqslant\lambda$)的下近似(或者上近似)集合。

第4章　知识获取

本章将就知识获取问题进行一个比较全面的讨论，使读者对知识获取的基本概念、原理和方法有一个基本的认识，然后结合Rough集对知识获取进行讨论。

4.1　知识获取概述

目前，知识获取方法大多是基于机器学习、模式识别以及统计学等的。例如，决策树方法（如ID3和C4.5），主要是对数据库中的某类数据寻找出关于该类数据的描述和模型。非线性回归分析与分类方法是利用回归分析的方法产生一个将数据项映射到一个实值预测变量的函数，发现变量或属性间的依赖关系。聚类分析方法根据识别出的一组聚类规则，将数据分成若干类，其主要采用可能性稠密度估计法。依赖关系学习模型法是要构造一个描述变量之间函数依赖关系或相关关系的模型，例如信念网络。变化和偏差分析方法，偏差包括很大一类潜在有用的知识，如分类中的异常实例、模式例外、观察结果对期望的偏离以及量值随时间的变化等等，该方法的基本思想是寻找观察结果与参照量之间有意义的差别。观察可以是一组变量值的某个模式，参照可以是给定模型的预测、外界提供的标准量或另一个观察。

Fayyad等对数据库知识获取作了如下定义：知识获取是识别出存在于数据库中有效的、新颖的、具有潜在效用的乃至最终可理解的模式的非平凡过程。他们将知识获取大致归纳为如下步骤：

（1）理解领域知识和相关的先验知识，明确系统目标；

（2）创建相关的目标数据集（原始样例库），即选择需要进行

知识获取的变量或数据样本的一个子集；

(3) 数据整理和预处理，例如去除明显错误的冗余的噪音数据，收集噪音信息以决定在后续步骤采取何种解决噪音问题的方法；

(4) 数据约简和投影，寻找依赖于获取目标的表达数据的有用特征，以约简数据模式；

(5) 选择一种与第(1)步所选目标相应的知识获取方法，如分类、综合、回归、聚类等；

(6) 选择知识获取算法，即选择用于搜索数据中模式的方法；

(7) 实施知识获取算法，得到分类规则或聚类等形式来表达的感兴趣的模式；

(8) 解释得到的模式，也可采用可视化表示等方法；可重复第(1)步到第(7)步的迭代过程；

(9) 巩固得到的知识，如检查与其他知识是否冲突，将知识合并到另一系统，以进一步加工利用等。

上述各步都可以与用户交互，以得到用户感兴趣的好的知识结果。

目前，众多知识获取方法的研究主要专注于知识获取过程的第(6)步到第(7)步，这些算法大多应用和发展了机器学习理论。就知识获取过程整体而言，尚缺乏坚实的理论基础。

4.2 基于 Rough 集的知识获取

Rough 集理论可支持知识获取的多个步骤，如数据预处理、数据约简、规则生成、数据依赖关系获取等。近年来，Rough 集理论对模糊和不完全知识的处理比较出色，成为数据库知识获取研究中的有力工具。

下面对基于 Rough 集的知识获取的一些基本概念进行介绍，为以后章节讨论具体的知识获取问题奠定基础。

4.2.1　可辨识矩阵

可辨识矩阵(也称分明矩阵)是由斯科龙(Skowron)教授提出的。

定义 4.1　令决策表系统为 $S=(U,R,V,f)$，$R=P\cup D$ 是属性集合，子集 $P=\{a_i\mid i=1,\cdots,m\}$ 和 $D=\{d\}$ 分别称为条件属性集和决策属性集，$U=\{x_1,x_2,\cdots,x_n\}$ 是论域，$a_i(x_j)$ 是样本 x_j 在属性 a_i 上的取值。$C_D(i,j)$ 表示可辨识矩阵中第 i 行 j 列的元素，则可辨识矩阵 C_D 定义为

$$C_D(i,j)=\begin{cases}\{a_k\mid a_k\in P\wedge a_k(x_i)\neq a_k(x_j)\}, & d(x_i)\neq d(x_j);\\ 0, & d(x_i)=d(x_j)。\end{cases}$$

其中 $i,j=1,\cdots,n$。

显然，可辨识矩阵是一个依主对角线对称的矩阵，在考虑可辨识矩阵的时候，只需要考虑其上三角(或下三角)部分就可以了。

根据可辨识矩阵的定义可知，当两个样本(实例)的决策属性取值相同时，它们所对应的可辨识矩阵元素的取值为 0；当两个样本的决策属性不同且可以通过某些条件属性的取值不同加以区分时，它们所对应的可辨识矩阵元素的取值为这两个样本属性值不同的条件属性集合，即可以区分这两个样本的条件属性集合；当两个样本发生冲突时，即所有的条件属性取值相同而决策属性的取值不同时，则它们所对应的可辨识矩阵中的元素取值为空集。显然，可辨识矩阵元素中是否包含空集元素可以作为判定决策表系统中是否包含不一致(冲突)信息的依据。

对于一般的信息表，我们也可以定义相应的可辨识矩阵。

定义 4.2　令信息表系统为 $S=(U,R,V,f)$，R 是属性集合，$U=\{x_1,x_2,\cdots,x_n\}$ 是论域，$a_i(x_j)$ 是样本 x_j 在属性 a_i 上的取值。$C_D(i,j)$ 表示可辨识矩阵中第 i 行 j 列的元素，则可辨识矩阵 C_D 定义为

$$C_D(i,j)=\{a_k\mid a_k\in P\wedge a_k(x_i)\neq a_k(x_j)\},$$

其中 $i,j=1,\cdots,n$。

4.2.2 属性重要性

根据 3.2 节中介绍的知识 B 对集合簇 F 近似分类的质量 $\mathrm{r}_B(F)$ 这一概念，我们可以对论域样本属性的重要程度进行度量。在我们了解一个论域中的样例的时候，我们可以通过知道其属性值来对样例进行处理。在现实情况中，有时我们并不一定知道一个样例的所有属性值，需要根据部分属性值来进行判定，有时我们需要确定一个论域中是否每个属性值的重要程度是一样的，因为度量不同属性值的代价可能不一样。在专家系统中，也会遇到类似的问题，即权重的问题，重要性高的属性在作决策时赋予大的权重。但是，我们只能根据经验来选择权重，这就依赖于人的先验知识。利用 Rough 集，我们就可以对属性的重要性进行度量，这个度量是根据论域中的样例来得到的，不依赖于人的先验知识。

定义 4.3 对于 F 是属性集 D 导出的分类，属性子集 B' 在属性集 B 中的重要性（$B' \subseteq B$，如果属性集 B 是默认的，如 B 为条件属性全集，则可简称为属性子集 B' 的重要性）定义为

$$\mathrm{r}_B(F) - \mathrm{r}_{B\setminus B'}(F)。$$

这表示当我们从属性集 B 中去掉属性子集 B' 对 F 近似分类的质量的影响。

属性的重要性还可以有其他度量方法，如属性子集 B' 的重要性也可定义为

$$\mathrm{POS}_{B\setminus B'}(F)/\mathrm{POS}_B(F),$$

其中 $\mathrm{POS}_B(F) = \bigcup_{X\in F} \mathrm{POS}_B(X)$。

属性的重要性也是以后章节中讨论离散化、约简等问题的一个关键基础概念。

4.3 决策规则

在基于 Rough 集理论的知识获取研究中，主要是通过归纳学习和观察发现式学习来得到知识的。归纳学习是通过对大量的实

例进行推理归纳和对共性的分析，抽象出一般的概念和规则，使这些新概念和新规则能蕴涵所有实例。这种学习所接受的实例中，不仅有正例，还可能有反例，但这些反例是已被告知的，不属于噪音或矛盾，它们对学习的作用，甚至可能比正例还重要。观察发现式学习是归纳学习的更高层次，实例中包含噪音和矛盾，需要对它们进行鉴别和提纯，分析实例间的相互联系，实现概念聚类或发现新的概念和定律，因而有创新的成分。

通常，决策表包含了某一领域中的大量数据记录，是领域的实例数据库。它记录了大量实例的属性值和决策情况，是领域知识的载体，知识获取的目的就是要通过分析这个实例库来得到该领域中有用的、规律性知识。样例的记录，可能不完整，或者有差错，带有一定程度的噪音，甚至由于认识的不足和缺陷还可能有矛盾。从决策表分析得到的规律性知识，我们通常采用决策规则的形式记录下来，并可以在将来的决策过程中利用这些知识来对未知的观察实例进行决策判定。下面我们就来对决策规则进行形式化描述。

定义 4.4　定义公式如下：

(1) (a,v)(或写为 a_v，$a\in R$，$v\in V_a$，表示属性 a 的取值为 v)是原子公式；原子公式是公式。

(2) 如果 A 和 B 是公式，那么 $\neg A$，$A\wedge B$，$A\vee B$，(A)，$A\rightarrow B$ 都是公式。

(3) 只有按定义(1)和(2)所组成的式子是公式。

定义 4.5　公式 $A\rightarrow B$ 的逻辑含义称为决策规则，A 称为规则前件，B 称为规则后件，它们表达一种因果关系。其中，公式 A 中所包含的原子公式中只有决策表中的条件属性，B 中所包含的原子公式中只有决策表中的决策属性。

在第 2 章中，我们讲过，任何决策表都可以等价地转化成为具有单一决策属性的决策表。我们这里也针对单一决策属性这种决策情况来对决策规则作进一步讨论。

对于决策表 $S=(U,R,V,f)$，$R=C\cup\{d\}$ 是属性集合，子集 C

和$\{d\}$分别称为条件属性集和决策属性集，我们通常采用下面的决策规则形式。

定义 4.6 公式$(a_1,v_1)\wedge(a_2,v_2)\wedge\cdots\wedge(a_n,v_n)$称为$P$基本公式，这里$v_i\in V_{a_i}$，$\{a_1,a_2,\cdots,a_n\}\in P$，$P\subseteq C$。

定义 4.7 $A\rightarrow B$为决策规则，如果A是P基本公式且$B=(d,d_i)$，则$A\rightarrow B$为基本决策规则。

实际上，决策规则可以是基本决策规则的逻辑组合形式。

定理 4.1 任何决策规则都可以分解成为一个或多个等价的基本决策规则。

定义 4.8 在决策表S中，如果决策规则$A\rightarrow B$为真，即决策表中的所有实例都满足决策规则$A\rightarrow B$，则称决策规则$A\rightarrow B$在决策表S中是协调的；否则，称决策规则$A\rightarrow B$在决策表S中是不协调的。

下面举例对决策规则的协调性进行介绍。

例 4.1 考虑表 4.1所示的决策表。

表 4.1 一个决策表

U	a	b	c	d
1	1	2	0	1
2	0	0	1	0
3	1	1	0	2
4	2	1	2	1
5	0	1	1	1

该系统中条件属性集合为$\{a,b,c\}$，决策属性为d，考虑如下的决策规则：

(1) $(a,1)\wedge(b,2)\rightarrow(d,1)$；

(2) $(a,0)\wedge(b,0)\rightarrow(d,0)$；

(3) $(a,1)\wedge(b,1)\rightarrow(d,2)$；

(4) $(a,1)\wedge(c,0)\rightarrow(d,1)$。

其中，决策规则(1)，(2)，(3)是协调的，因为决策表中的每个实例都满足它们，而决策规则(4)是不协调的，因为第 3 个实例不满足这条规则。我们不能够根据前件$(a,1)\wedge(c,0)$得到肯定的决策$(d,1)$，因为第 3 个实例的条件属性值满足规则前件$(a,1)\wedge(c,0)$，而其决策属性值为 2。

因此，如何从决策表中最大限度地获取得到协调规则就是基于 Rough 集的知识获取所需要研究解决的问题。在确定性情况下，我们可以得到如上所述的确定规则；在不确定性情况下，我们需要得到包含不确定性信息的决策规则。为了获得最大限度的适应能力，有时在确定性情况下，我们也需要引入不确定性规则来提高规则的适应能力。我们将在下一章对不确定性进行讨论。

第5章　知识系统不确定性表示与处理

不确定性是智能问题的本质特征，无论是人类智能还是人工智能，都离不开不确定性的处理。可以说，智能主要反映在求解不确定性问题的能力上。不确定性是智能系统研究的核心课题。

本章主要对不确定信息的产生、影响和处理方法进行介绍。首先对不确定信息进行分析，然后介绍几种不确定知识的推理方法，最后针对基于Rough集理论的知识发现，对决策表的不确定性进行量化，并提出决策规则的不确定性表示方法，为下面章节的知识发现理论奠定基础。

5.1　知识表示

知识表示就是要研究用机器表示知识的可行的、有效的、通用的原则和方法。常用的知识表示方法有逻辑模式、框架、语义网络、产生式系统、状态空间、剧本等。我们这里只对本书中将要大量用到的产生式系统进行介绍，其他知识表示方法，有兴趣的读者可以参考相关的资料。

产生式系统是历史悠久且使用最多的知识表示系统，它与图灵机具有同样的计算能力，早已在自动机理论、形式文法和程序语言中得到广泛的应用。它用"IF　THEN"的规则形式捕获人类问题求解的行为特征，并通过认识——行动循环过程求解问题，其表现形式单一、直观，有利于知识的获取和形式化，其问题求解过程符合人的认知过程且易于计算机实现，有利于问题的求解和系统的开发。

产生式系统的基本结构是：

（1）一个规则库，每条规则是一个“条件—行动”产生式，且各规则之间的相互作用（调用关系）不大。规则可具有如下形式：

IF　〈触发事实 1 是真〉
　　〈触发事实 2 是真〉
　　⋮
　　〈触发事实 n 是真〉
THEN　〈结论事实 1〉
　　〈结论事实 2〉
　　⋮
　　〈结论事实 m〉

（2）工作存储器，也称数据或短期记忆缓冲区，是产生式规则注意的重点。规则库中每条产生式左侧所提的条件必须出现在工作存储器中，产生式才能发生动作（被激活）。工作存储器可以是简单的表、非常大的数组，或者更典型的具有本身某种内部结构的中等大小的缓冲器。

（3）解释程序是一个决定下一步做什么的程序。根据工作存储器选择规则，核实条件，激活并控制行动。

产生式系统有两种最基本的推理形式：前向（或正向）推理和后向（或反向）推理。前向推理又称数据驱动，反向推理又称目标驱动。

正向推理从已知事实出发，逐步推导出最后结论，其推理过程大致为：

（1）用工作存储器中的事实与产生式规则的前提条件进行匹配；

（2）按照冲突消解策略从匹配的规则实例中选择一条规则；

（3）执行选中规则的动作，依次修改工作存储器；

（4）用更新后的工作存储器，重复上述步骤，直到得出结论或工作存储器不再发生变化为止。

反向推理则是首先提出假设，然后验证这些假设的真假性，找

到假设成立的所有证据或事实。其推理过程大致是：

（1）看假设是否在工作存储器中，若在，则假设成立，推理结束；

（2）找出结论与此假设匹配的规则；

（3）按冲突消解策略，从匹配的规则实例中选择一条规则；

（4）将选中规则的前提条件作为新的假设，重复上述几步，直到假设的真假性被验证或不存在激活的规则。

产生式系统具有如下特征：

模块性：在产生式系统中，每条规则可自由增删、修改，像一个知识的知识块。规则间的关系通过工作存储器间接地表现出来，并非直接调用。

一致性：规则都具有一致、统一的结构。

自然性：产生式规则易于用来表示领域知识，解释知识的运用、推理过程。

效率低：在推理的各周期中要不断对全部规则的相应部分进行模式匹配，从原理上讲，这必然会使处理速度降低，规则越多，系统效率降低得越明显。

欠灵活：在问题求解过程中，有时希望根据情况对推理的方式进行调整，而产生式系统中推理控制方式的单一性阻碍了系统做到这一点。

5.2 不确定知识系统的几种推理方法

在许多实际领域中，如医学诊断、故障诊断、探矿、天气预报、军事指挥、市场分析、投资决策和调度控制等，问题的求解可利用的证据和知识常常是不确定的。造成这种不确定性的原因，既有客观原因，又有主观原因。

对于证据，随机性、模糊性可造成其不确定性，而且证据本身可能是错误的、不相关的、不完备的、不可靠的、近似的、部分的、度

量不准确的、记录有误的。这些都是导致证据不确定性的客观因素。造成证据不确定性也有人为的因素，如人的感情因素、认识心理上的偏差等。

对于知识，造成其不确定性的客观因素，除了随机性和模糊性外，还有知识之间的冲突（不一致）、经验知识形成的局限、知识获取的不完全等。人的经验知识往往是通过以下几种推理方式形成的：归纳推理、拓广推理、类比推理、统计推理、逆向推理、似然推理等。人们通过这些方法获得的经验知识是具有不确定性的。而且，人类对于客观世界的认识也不是完全的，而是不断发展和改进的，这就决定了我们在处理很多问题的时候所具有的知识不是完全的。

在建立系统、构造知识库的时候，也可能产生不确定性。首先，两个事件之间的因果关系不是简单的、容易指定的。任何事件都和很多复杂因素有关，并不是孤立的，我们在建立系统模型的时候，需要严格地限定模型，抓住主要因素，而忽略一些次要因素，这样得到的系统和知识库自然具有不确定性。其次，一些经验知识，特别是直觉知识，我们难于表达精确（实际上，在有些问题上，只能是模糊表达）。再次，知识表达语言也会引入不确定性，特别是用自然语言表达知识的时候，更是难于避免由于自然语言本身的含糊性所带来的不确定性。另外，机器中存储、使用的知识总是有限的，这也会导致不确定性。对于通过机器学习所获得的知识，也会由于学习样例的不足、学习算法的不完备、学习过程处理所带来的不确定（有时是不可预见的，有时是为了达到一定的目的而故意引入的）等等原因，产生不确定性。

对于推理过程，导致不确定性产生的主要原因在于知识不确定性的动态积累和传递过程。在推理的每一步都需要综合证据和规则的不确定因素，为此，通常要通过某种不确定的测度，寻找尽可能符合客观实际的计算模式，随着推理步骤的展开和不确定测度的传递计算，最终得到结果的不确定性测度。

对于不确定性知识，我们需要有有效的工具来表达和处理。经典逻辑缺乏这方面的能力。本节中，我们将对几种不确定知识的表达处理方法进行讨论。实际上，每种知识表示方法并不仅仅是一种表达知识的形式，还对应于相应的知识应用方法，即推理方法，也就是如何使用以某种知识表达形式所表示的知识进行智能计算（推理），实现不确定信息的传递，得到所需要的判定和决策。

5.2.1 概率模型

不确定性与概率有许多内在的联系。用概率来描述智能系统中的不确定性，必须将概率的定义加以拓展。这里，我们把概率解释为人对证据和规则的主观信任度。Bayes 定理是概率推理模型的基础。主观 Bayes 方法就是以概率论中的 Bayes 公式为基础的一种不确定推理模型，1976 年首先在 Duda 等人开发的矿物勘探专家系统 PROSPECTOR 中得到成功应用。

在概率模型中，证据 E 的不确定性为 E 发生的概率 $P(E)$，或者用 E 的几率 $O(E)$ 来表示：

$$O(E)=\frac{P(E)}{1-P(E)}。$$

显然，若 E 为真，$P(E)=1$，$O(E)=\infty$；

若 E 为假，$P(E)=0$，$O(E)=0$；

若对 E 一无所知，$P(E)$ 和 $O(E)$ 可取 E 的先验概率和先验几率。

规则可表示为 IF E THEN $H(LS,LN)$ 的形式，其中 LS,LN 用于表示规则的强度。

Bayes 公式可表示为

$$P(H|E)=\frac{P(E|H)\cdot P(H)}{P(E)}$$

和

$$P(\overline{H}|E)=\frac{P(E|\overline{H})\cdot P(\overline{H})}{P(E)},$$

两式相除得

$$O(H|E)=\frac{P(H|E)}{P(\overline{H}|E)}=\frac{P(E|H)\cdot P(H)}{P(E|\overline{H})\cdot P(\overline{H})}$$
$$=\frac{P(E|H)}{P(E|\overline{H})}\cdot O(H),$$

同理得

$$O(H|\overline{E})=\frac{P(\overline{E}|H)}{P(\overline{E}|\overline{H})}\cdot O(H),$$

定义

$$LS=\frac{P(E|H)}{P(E|\overline{H})},\qquad LN=\frac{P(\overline{E}|H)}{P(\overline{E}|\overline{H})},$$

则

$$O(H|E)=LS\cdot O(H),$$
$$O(H|\overline{E})=LN\cdot O(H)。$$

LS 称为规则的充分性度量，LN 称为规则的必要性度量，它们都是几率修改因子。

- 当 $P(E)=1$（即证据为真）时，可以利用 LS 将假设（结论）的先验几率 $O(H)$ 更新为后验几率 $O(H|E)$；
- 当 $P(E)=0$（即证据为假）时，可以利用 LN 将假设（结论）的先验几率 $O(H)$ 更新为后验几率。

LS 和 LN 具有如下性质：

(1) LS

当 $LS=1$ 时，$O(H|E)=O(H)$，说明证据 E 对结论 H 没有影响。

当 $LS>1$ 时，$O(H|E)>O(H)$，说明证据 E 支持结论 H，LS 越大，E 对 H 的支持越充分，$O(H|E)$ 比 $O(H)$ 大得越多。

当 $LS<1$ 时，$O(H|E)<O(H)$，说明证据 E 排斥结论 H。

(2) LN

当 $LN=1$ 时，$O(H|\overline{E})=O(H)$，说明证据 $\overline{E}$ 对结论 H 没有影响。

当 $LN>1$ 时，$O(H|\overline{E})>O(H)$，说明证据 $\overline{E}$ 支持结论 H。

当 $LN<1$ 时，$O(H|\overline{E})<O(H)$，说明证据 $\overline{E}$ 排斥结论 H，LN 越小，E 的不出现就越排斥结论 H，或者说结论 H 为真就越是必须要证据 E 为真。

(3) LS 与 LN 的关系

E 和 $\overline{E}$ 不会同时支持或排斥结论 H，只有下述三种情况发生：

$LS>1$ 且 $LN<1$：证据 E 支持结论 H；

$LS<1$ 且 $LN>1$：证据 E 排斥结论 H；

$LS=1$ 且 $LN=1$：证据 E 与结论 H 无关。

下面我们再来介绍基于概率的不确定性推理方法。

(1) 概率传播

当 $P(E)=1$ 或 $P(E)=0$ 时，我们有

$$O(H|E)=LS\cdot O(H),$$

$$O(H|\overline{E})=LN\cdot O(H)。$$

用概率表示为：

$$P(H|E)=\frac{LS\cdot P(H)}{(LS-1)\cdot P(H)+1},$$

$$P(H|\overline{E})=\frac{LN\cdot P(H)}{(LN-1)\cdot P(H)+1}。$$

证明

$$\frac{LS\cdot P(H)}{(LS-1)\cdot P(H)+1}=\frac{\frac{P(E|H)}{P(E|\overline{H})}\cdot P(H)}{(\frac{P(E|H)}{P(E|\overline{H})}-1)\cdot P(H)+1}$$

$$=\frac{P(H|E)}{(P(E|H)-P(E|\overline{H}))\cdot P(H)+P(E|\overline{H})}$$

$$=\frac{P(H|E)}{P(H|E)-\frac{1-P(H|E)}{1-P(H)}\cdot P(H)+\frac{1-P(H|E)}{1-P(H)}}$$

$$=\frac{P(H|E)(1-P(H))}{P(H|E)-P(H|E)\cdot P(H)-P(H)+P(H|E)\cdot P(H)+1-P(H|E)}$$

$$=\frac{P(H|E)(1-P(H))}{1-P(H)}$$

$$=P(H|E),$$

$$\frac{LN \cdot P(H)}{(LN-1) \cdot P(H)+1}=\frac{\frac{P(E|H)}{P(\overline{E}|H)} \cdot P(H)}{(\frac{P(E|H)}{P(\overline{E}|H)}-1) \cdot P(H)+1}$$

$$=\frac{P(H|E)}{(P(E|H)-P(E|H)) \cdot P(H)+P(E|H)}$$

$$=\frac{P(H|E)}{P(H|E)-\frac{1}{1}\frac{P(H|\overline{E})}{P(H)} \cdot P(H)+\frac{1-P(H|\overline{E})}{1-P(H)}}$$

$$=\frac{P(H|\overline{E})(1-P(H))}{P(H|E)-P(H|\overline{E}) \cdot P(H)-P(H)+P(H|\overline{E}) \cdot P(H)+1-P(H|E)}$$

$$=\frac{P(H|E)(1-P(H))}{1-P(H)}$$

$$=P(H|E)。$$

但是，当证据不确定时，即在 $0<P(E)<1$ 时，如何根据 $P(E)$ 更新 $P(H)$？这个问题可以表达为：在现实观察 S 下，证据 E 的概率为 $P(E|S)$，求解 $P(H|S)$。对于这个问题，可以采用下面的分段线性插值法解决。

我们有

$$\begin{aligned}P(H|S)&=P(H,E|S)+P(H,\overline{E}|S)\\&=P(H|E,S) \cdot P(E|S)+P(H|\overline{E},S) \cdot P(\overline{E}|S)\\&=P(H|E) \cdot P(E|S)+P(H|\overline{E},S) \cdot P(\overline{E}|S),\end{aligned}$$

在 $P(E|S)$ 的三个特殊点上，我们可以求得 $P(H|S)$ 的值：

- 当 $P(E|S)=1$ 时，$P(H|S)=P(H|E)$；
- 当 $P(E|S)=0$ 时，$P(H|S)=P(H|\overline{E})$；
- 当 $P(E|S)=P(E)$ 时，

$$\begin{aligned}P(H|S)&=P(H|E) \cdot P(E|S)+P(H|\overline{E}) \cdot P(\overline{E}|S)\\&=P(H|E) \cdot P(E)+P(H|\overline{E}) \cdot P(\overline{E})\\&=P(H)。\end{aligned}$$

有了上述三个特殊点的值后，可将 $P(H|S)$ 的函数近似取为这三个点的分段线性插值函数，如图 5.1 所示。函数的解析式是：

$$P(H,S)=\begin{cases}P(H|\bar{E})+\dfrac{P(H)-P(H|\bar{E})}{P(\bar{E})}\cdot P(E|S), \\ \qquad\qquad \text{当 } 0\leqslant P(E|S)<P(E) \text{ 时;} \\ P(H)+\dfrac{P(H|E)-P(H)}{1-P(E)}\cdot(P(E|S)-P(E)), \\ \qquad\qquad \text{当 } P(E)\leqslant P(E|S)\leqslant 1 \text{ 时。}\end{cases}$$

上式称为 EH 公式。

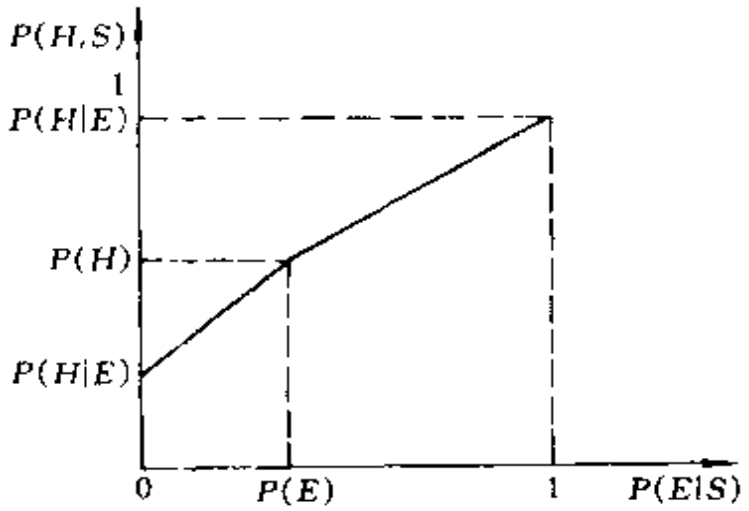


图 3.1 分段线性插值函数

(2) 独立证据导出同一结论(并行法则)

设独立证据 $E_1,E_2,\cdots,E_n$ 的观察分别为 $S_1,S_2,\cdots,S_n$,且有规则 $E_1\to H,E_2\to H,\cdots,E_n\to H$。假定由这些规则分别得到的结论 H 的后验几率为 $O(H|S_1),O(H|S_2),\cdots,O(H|S_n)$,则根据这些独立证据的组合所应该得到的结论 H 的后验几率为

$$O(H|S_1\wedge S_2\wedge\cdots\wedge S_n)=\frac{O(H|S_1)}{O(H)}\cdot\frac{O(H|S_2)}{O(H)}\cdot\cdots\cdot\frac{O(H|S_n)}{O(H)}\cdot O(H)。$$

(3) 证据的合取

在观察 S 下,若证据 $E_1,E_2,\cdots,E_n$ 的概率为

$$P(E_1|S),P(E_2|S),\cdots,P(E_n|S),$$

则
$$P(E_1 \wedge E_2 \wedge \cdots \wedge E_n \mid S)$$
$$=\min\{P(E_1|S), P(E_2|S), \cdots, P(E_n|S)\}。$$

(4) 证据的析取

在观察 S 下，若证据 $E_1, E_2, \cdots, E_n$ 的概率为

$$P(E_1|S), P(E_2|S), \cdots, P(E_n|S),$$

则
$$P(E_1 \vee E_2 \vee \cdots \vee E_n|S)$$
$$=\max\{P(E_1|S), P(E_2|S), \cdots, P(E_n|S)\}。$$

为了便于用户输入初始证据的确信度，可以用可信度 $C(E|S)$ 代替 $P(E|S)$。两者之间有如图 5.2 所示的简单的保持大小次序的对应

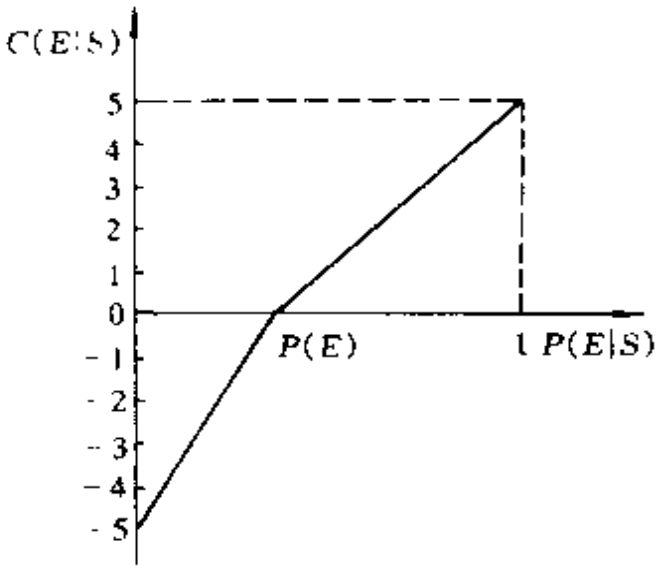


图 5.2　$C(E|S)$和$P(E|S)$的关系

关系，其解析式为

$$C(E|S)=\begin{cases}5\cdot\dfrac{P(E|S)-P(E)}{1-P(E)}, & \text{当 } P(E)<P(E|S)\leqslant 1 \text{ 时；}\\ 5\cdot\dfrac{P(E|S)-P(E)}{P(E)}, & \text{当 } 0\leqslant P(E|S)\leqslant P(E) \text{ 时。}\end{cases}$$

根据 $C(E|S)$和 $P(E|S)$的关系，可将 EH 公式修改为

$$P(H|S)=\begin{cases}P(H|E)-[P(H)-P(H|\bar{E})]\cdot[1-\frac{1}{5}C(E|S)], \\ \qquad\qquad \text{当}-5\leqslant C(E|S)\leqslant 0\text{ 时;} \\ P(H)+[P(H|E)-P(H)]\cdot\frac{1}{5}C(E|S), \\ \qquad\qquad \text{当}\ 0\leqslant C(E|S)\leqslant 5\text{ 时。}\end{cases}$$

这个公式称为 C-P 公式。

概率推理模型的缺点是，必须预先知道每个命题（包括初始证据、中间结果和最终结论）的先验概率，在很多情况下这是比较困难的。

5.2.2 可信度模型

可信度模型是 Shortliffe 与 Buchanan 等人开发的医疗专家系统 MYCIN 系统中使用的一种不确定性推理模型。该模型采用可信度 $CF(H|E)$ 作为不确定性测度，表示假设 H 在证据 E 下主观信任度的一种修改量，反映了人对不确定知识增加或减少信任的程度。$CF(H|E)$ 的取值范围为 $[-1,1]$，其定义如下：

$$CF(H|E)=MB(H|E)-MD(H|E)。$$

其中，$MB(H|E)$ 为信任增长度，表示因为证据 E 的出现对假设 H 为真的信任的增加程度，即当 $MB(H|E)>0$ 时，有 $P(H|E)>P(E)$。$MD(H|E)$ 为不信任增长度，表示因为证据 E 的出现对假设 H 为真的信任的减少程度，即当 $MD(H|E)>0$ 时，有 $P(H|E)<P(E)$。$MB(H|E)$ 和 $MD(H|E)$ 都是区间 $[0,1]$ 上的实数，其定义为：

$$MB(H|E)=\begin{cases}1, & \text{若 } P(H)=1; \\ \dfrac{\max\{P(H|E),P(H)\}-P(H)}{1-P(H)}, & \text{否则。}\end{cases}$$

$$MD(H|E)=\begin{cases}1, & \text{若 } P(H)=0; \\ \dfrac{\min\{P(H|E),P(H)\}-P(H)}{-P(H)}, & \text{否则。}\end{cases}$$

MB 和 MD 具有如下性质：

(1) 互斥性

当 $MB(H|E)>0$ 时，$MD(H|E)=0$；

当 $MD(H|E)>0$ 时，$MB(H|E)=0$。

(2) 典型值

若 $P(H|E)=1$，则

$$MB(H|E)=1, MD(H|E)=0, CF(H|E)=1;$$

若 $P(H|E)=0$，则

$$MB(H|E)=0, MD(H|E)=1, CF(H|E)=-1;$$

若 $P(H|E)=P(H)$，则

$$MB(H|E)=0, MD(H|E)=0, CF(H|E)=0。$$

(3) $CF(H|E)+CF(H|E)=0$。

(4) 互斥假设

若对于同一证据有 n 个互不相容的假设 $H_i(i=1,2,\cdots,n)$，则 $\sum_{i=1}^{n} CF(H_i|E)\leqslant 1$。

上式中，只有证据 E 在逻辑上蕴涵某个假设 H_i 时，等式才成立。

由 MB 和 MD 的形式定义，CF 可以直接用概率表示为：

$$CF(H|E)=\begin{cases}\dfrac{P(H|E)-P(H)}{1-P(H)}, & \text{若 } P(H|E)>P(H);\\ 0, & \text{若 } P(H|E)=P(H);\\ \dfrac{P(H|E)-P(H)}{P(H)}, & \text{若 } P(H|E)<P(H)。\end{cases}$$

下面再来讨论利用可信度的推理方法。

(1) 可信度传播

根据当前观察 S 下证据的可信度 $CF(E|S)$ 以及规则的可信度计算假设在当前观察下的可信度，

$$CF(H|S)=CF(H|E)\times \mathrm{MAX}\{0, CF(E|S)\},$$

或

$$MB(H|S)=MB(H|E)\times \mathrm{MAX}\{0,CF(E|S)\},$$

$$MD(H|S)=MD(H|E)\times \mathrm{MAX}\{0,CF(E|S)\}。$$

当 $CF(E|S)>0$ 时，规则前提以某种程度为真，结论的可信度 $CF(H|S)=CF(H|E)\times CF(E|S)$。

当 $CF(E|S)<0$ 时，规则前提为假，说明该规则不能用，$CF(H|S)=0$。

(2) 并行法则

假设由规则 IF E_1 THEN H 和 IF E_2 THEN H 分别得到 $CF(H|E_1)$ 和 $CF(H|E_2)$，则综合这两条规则得到的结论为：

$$CF(H|E_1\wedge E_2)=\begin{cases} CF(H|E_1)+CF(H|E_2)-CF(H|E_1)\cdot CF(H|E_2), \\ \qquad\qquad 若\ CF(H|E_1)\geqslant 0, CF(H|E_2)\geqslant 0; \\ CF(H|E_1)+CF(H|E_2)+CF(H|E_1)\cdot CF(H|E_2), \\ \qquad\qquad 若\ CF(H|E_1)<0, CF(H|E_2)<0; \\ \dfrac{CF(H|E_1)+CF(H|E_2)}{1-\min\{|CF(H|E_1)|,|CF(H|E_2)|\}}, \quad 否则。 \end{cases}$$

在组合两个以上的独立证据时，可以先组合两个，然后再依次与其他证据进行组合。

(3) 证据的合取

$$CF(E_1\wedge E_2\wedge\cdots\wedge E_n|S)$$
$$=\min\{CF(E_1|S),CF(E_2|S),\cdots,CF(E_n|S)\}。$$

(4) 证据的析取

$$CF(E_1\vee E_2\vee\cdots\vee E_n|S)$$
$$=\max\{CF(E_1|S),CF(E_2|S),\cdots,CF(E_n|S)\}。$$

这个模型的主要吸引力在于其对证据和假设不确定性的综合方法十分简单，且可信度赋值通常比 Bayes 方法中的概率赋值容易得多。但是，它仅是一种直觉化的特殊模型，并没有强有力的理论来保证可信度计算模型的正确性和协调性，在证据相关的情况下，并行法则和概率传播都会导致矛盾的结论。

例 5.1 已知规则

$$R_1: A \rightarrow X, CF(X|A)=0.8,$$

$$R_2: B \rightarrow X, CF(X|B)=0.5,$$

$$R_3: X \wedge E \rightarrow Y, CF(Y|X \wedge E)=0.8,$$

初始 A, B, E 的可信度 $CF(A|S), CF(B|S), CF(E|S)$ 均为 1，X 和 Y 未知，求 $CF(X|S)$ 和 $CF(Y|S)$。

解　根据可信度传播规则可得

$$CF(X|A)=0.8\times\max\{0, CF(A|S)\}=0.8,$$

$$CF(X|B)=0.5\times\max\{0, CF(B|S)\}=0.5。$$

再根据并行法则得

$$CF(X|S)=0.8+0.5-0.8\times0.5=0.9,$$

由证据合取得

$$CF(X \wedge E)=\min\{0.9, 1\}=0.9,$$

再由可信度的传播得

$$CF(Y|S)=0.8\times\max\{0, 0.9\}=0.72。$$

5.2.3　证据理论

证据理论也称 Dempster/Shafer 证据理论，是由 Dempster 首先提出，由 Shafer 进一步发展的一种不确定推理理论。概率推理模型中，必须给出先验概率，而证据理论则能够处理这种由不知道引起的不确定性。证据理论满足比概率论更弱的公理系统，当概率值已知时，证据理论就变成了概率论。

1. 基本理论

设 Ω 是变量 x 的所有可能值的穷举集合(值域)，Ω 中的各元素是相互排斥的，我们称 Ω 为辨别框。设 Ω 的元素个数为 N，则 Ω 的幂集 2^{Ω} 的元素个数为 2^N，幂集中的每个元素(值域的子集)对应于关于 x 取值情况的一个命题。

定义 5.1　对 Ω 的任一子集(命题)A，命它对应一个数 $m\in[0,1]$，而且满足：

$$m(\varnothing)=0 \quad 和 \quad \sum_{A\subseteq\Omega} m(A)=1,$$

则称 m 为 2^{Ω} 上的基本概率分配函数，称 $m(A)$ 为 A 的基本概率数。$m(A)$ 的意义为

- 若 $A\subseteq\Omega$ 且 $A\neq 1$，则 $m(A)$ 表示对 A 的信任程度；
- 若 $|A|=1$，则 $m(A)$ 表示对 A 的精确信任程度；
- 若 $A=\Omega$，则 $m(A)$ 表示这个数不知如何分配。

定义 5.2 若 $A\subseteq\Omega$ 且 $m(A)\neq 0$，称 A 是 m 的一个焦元。

例 5.2 设 $\Omega=\{$红，黄，白$\}$，2^{Ω} 上的基本概率分配函数 m 为 $m(\{$红$\},\{$黄$\},\{$白$\},\{$红，黄$\},\{$红，白$\},\{$黄，白$\},\{$红，黄，白$\},\{\})=(0.3,0.0,0.1,0.2,0.2,0.0,0.2,0.0)$，

其中：$m(\{$红$\})=0.3$ 表示对命题$\{$红$\}$的精确信任程度；

$m(\{$红，黄，白$\})=0.2$ 表示不知道这 0.2 如何分配；

$m(\{$红，黄$\})=0.2$ 表示这 0.2 是对命题$\{$红，黄$\}$的信任程度，但不知道这 0.2 如何分配给$\{$红$\}$，$\{$黄$\}$。

值得注意的是

$$m(\{\text{红}\})+m(\{\text{黄}\})+m(\{\text{白}\})=0.3+0.0+0.1=0.4<1,$$

因此，m 不同于概率 P，因为 $P(\{$红$\})+P(\{$黄$\})+P(\{$白$\})=1$。

定义 5.3 命题的信任函数 $Bel:2^{\Omega}\to[0,1]$ 为

$$Bel(A)=\sum_{B\subseteq A}m(B)。$$

$Bel(A)$ 表示对 A 的总的信任。如上例中

$$Bel\{\text{红，黄}\}=m(\{\text{红}\})+m(\{\text{黄}\})+m(\{\text{红，黄}\})=0.3+0.0+0.2=0.5,$$

根据定义可知：

$$Bel(\varnothing)=0,\quad Bel(\Omega)=1。$$

定义 5.4 命题 A 的似然函数 $Pl:2^{\Omega}\to[0,1]$ 为

$$Pl(A)=1-Bel(\overline{A})=\sum_{B\cap A\neq\varnothing}m(B),$$

$Pl(A)$ 表示不否定 A 的信任程度。

显然，$Pl(A) \geqslant Bel(A)$。

$Bel(A)$和 $Pl(A)$分别为命题 A 的信任度的下限和上限，记作 $A[Bel(A), Pl(A)]$。如在上例中

$$Bel\{红\} = m(\{红\}) + m(\{\}) = 0.3 + 0.0 = 0.3,$$

$$\begin{aligned} Pl\{红\} &= 1 - Bel(\overline{\{红\}}) \\ &= 1 - Bel\{黄, 白\} \\ &= 1 - (m\{黄\} + m\{白\} + m\{黄, 白\}) \\ &= 1 - (0 + 0.1 + 0) \\ &= 0.9。 \end{aligned}$$

命题的信任度的上限和下限反映了命题的许多重要信息。下面对一些典型值进行讨论：

$A[0,1]$：说明对 A 一无所知。

$A[1,1]$：说明 A 为真。

$A[0,0]$：说明 A 为假。

$A[0.6,1]$：说明 A 比较真(对 A 部分信任)。

$A[0,0.4]$：说明 A 比较假(对 $\overline{A}$ 部分信任)。

$A[0.3,0.9]$：说明对 A 和 $\overline{A}$ 都部分信任。

2. 证据的组合

对于同样的证据，由于来源不同，会得到不同的概率分配函数。Dempster 提出用正交和来组合这些函数。

定义 5.5　设 $m_1, m_2, \cdots, m_n$ 为 2^{Ω} 上的 n 个基本概率分配函数，他们的正交和 $m = m_1 \oplus m_2 \oplus \cdots \oplus m_n$ 为

$$\begin{cases} m(\varnothing) = 0, \\ m(A) = k \cdot \sum\limits_{\cap A_i = A} \prod\limits_{1 \leqslant i \leqslant n} m_i(A_i), \quad A \neq \varnothing。 \end{cases}$$

式中的 k 由 $k^{-1} = 1 - \sum\limits_{\cap A_i = \varnothing} \prod\limits_{1 \leqslant i \leqslant n} m_i(A_i) = \sum\limits_{\cap A_i \neq \varnothing} \prod\limits_{1 \leqslant i \leqslant n} m_i(A_i)$ 决定，若 $k^{-1} = 0$，则 m_i 之间是矛盾的。

例 5.3　设 $\Omega = \{a, b\}$，且从不同的知识来源得到的基本概率分配函数为

$$m_1(\{\},\{a\},\{b\},\{a,b\})=(0,0.4,0.5,0.1),$$
$$m_2(\{\},\{a\},\{b\},\{a,b\})=(0,0.6,0.2,0.2),$$

求正交和 $m=m_1\oplus m_2$

解 先求 k^{-1}：

$$k^{-1}=\sum_{\bigcap A_i\neq\varnothing}\prod_{1\leqslant i\leqslant 2}m_i(A_i)$$
$$=\sum_{X\cap Y\neq\varnothing}m_1(X)\cdot m_2(Y)$$
$$=m_1(\{a\})\cdot m_2(\{a\})+m_1(\{a\})\cdot m_2(\{a,b\})$$
$$+m_1(\{b\})\cdot m_2(\{b\})+m_1(\{b\})\cdot m_2(\{a,b\})$$
$$+m_1(\{a,b\})\cdot m_2(\{a\})+m_1(\{a,b\})\cdot m_2(\{b\})$$
$$+m_1(\{a,b\})\cdot m_2(\{a,b\})$$
$$=0.4\times0.6+0.4\times0.2+0.5\times0.2+0.5\times0.2$$
$$+0.1\times0.6+0.1\times0.2+0.1\times0.2$$
$$=0.62。$$

再求 $m(A)$：

$$m(\{a\})=k\cdot\sum_{X\cap Y=\{a\}}m_1(X)\cdot m_2(Y)$$
$$=\frac{1}{0.62}[m_1(\{a\})\cdot m_2(\{a\})+m_1(\{a\})\cdot m_2(\{a,b\})$$
$$+m_1(\{a,b\})\cdot m_2(\{a\})]$$
$$=\frac{1}{0.62}(0.4\times0.6+0.4\times0.2+0.1\times0.6)$$
$$=0.61。$$

同理，$m(\{b\})=0.36$，$m(\{a,b\})=0.03$，

所以 $m(\{\},\{a\},\{b\},\{a,b\})=(0,0.61,0.36,0.03)$。

3. 基本算法

上面介绍了证据理论的公理系统。由于基本概率分配函数的不同，会产生不同的算法。下面介绍其中的一种。

(1) 知识表示

设某个领域的辨别框 $\Omega=\{s_1,s_2,\cdots,s_n\}$，命题 $A,B,\cdots$ 是 Ω 的

子集，推理规则为

$$\text{IF}\quad E\quad \text{THEN}\quad H,CF,$$

其中 E 和 H 为命题的逻辑组合，CF 为可信度因子。

命题和可信度因子可表示为

$$A=\{a_1,a_2,\cdots,a_k\},$$

$$CF=\{c_1,c_2,\cdots,c_k\}。$$

其中 c_i 为 a_i 的可信度。

对任何命题 A，A 的可信度 CF 应满足：

a) $c_i\geqslant 0,\quad 1\leqslant i\leqslant k$；

b) $\sum_{i=1}^{k} c_i\leqslant 1$，

(2) 证据描述

设 m 为 2^{Ω} 上定义的基本概率分配函数，它满足如下条件：

a) $m(\{s_i\})\geqslant 0$，对 $s_i\in\Omega$；

b) $\sum_{i=1}^{n} m(\{s_i\})\leqslant 1$；

c) $m(\Omega)=1-\sum_{i=1}^{n} m(\{s_i\})$；

d) $m(A)=0$，对 $A\subset\Omega$，且 $|A|>1$ 或 $|A|=0$。

例 5.4　设 $\Omega=\{$红，黄，白$\}$，2^{Ω} 上的基本概率分配函数 m 为 $m(\{$红$\},\{$黄$\},\{$白$\},\{$红，黄$\},\{$红，白$\},\{$黄，白$\},\{$红，黄，白$\},\{\})=(0.6,0.2,0.1,0.0,0.0,0.0,0.1,0.0)$。

下面再来看满足上述条件的基本概率分配函数的正交和以及信任函数和似然函数的定义。

定义 5.6　设 m_1,m_2 为 2^{Ω} 上的两个基本概率分配函数，他们的正交和 $m=m_1\oplus m_2$ 为：

$$m(\{s_i\})=k\times[m_1(\{s_i\})\cdot m_2(\{s_i\})+m_1(\{s_i\})\cdot m_2(\Omega)+m_1(\Omega)\cdot m_1(\{s_i\})],$$

其中 $k^{-1}=m_1(\Omega)\cdot m_2(\Omega)+\sum_{i=1}^{n}[m_1(\{s_i\})\cdot m_2(\{s_i\})$

$$+m_1(\Omega)\cdot m_2(\{s_t\})+m_1(\{s_t\})\cdot m_2(\Omega)]。$$

若 $k\neq 0$，则 m_1 和 m_2 之间是矛盾的。

定义 5.7 对任意命题 $A\subseteq\Omega$，其信任函数为

$$Bel(A)=\sum_{B\subseteq A} m(B)=\sum_{a\in A} m(\{a\})。$$

定义 5.8 对任意命题 $A\subseteq\Omega$，其似然函数为

$$\begin{aligned}Pl(A)&=1-Bel(\overline{A})\\&=1-\sum_{a\notin A} m(\{a\})\\&=1-\Big[\sum_{a\in\Omega} m(\{a\})-\sum_{a\in A} m(\{a\})\Big]\\&=1-[1-m(\Omega)-Bel(A)]\\&=m(\Omega)+Bel(A)。\end{aligned}$$

由此可以看出，命题的信任函数和似然函数之间满足如下关系：

$$Pl(A)-Bel(A)=m(\Omega)\geqslant 0。$$

可以根据命题的信任函数和似然函数以及命题中的元素个数，定义命题的类概率函数，并作为命题的确定性度量。

定义 5.9 设 Ω 为有限域，对任意命题 $A\subseteq\Omega$，A 的类概率函数为

$$f(A)=Bel(A)+\frac{|A|}{|\Omega|}\times[Pl(A)-Bel(A)]。$$

容易证明，类概率函数具有下列性质：

a) $\sum_{a\in\Omega} f(\{a\})=1$；

b) $Bel(A)\leqslant f(A)\leqslant Pl(A)$；

c) $f(\overline{A})=1-f(A)$；

d) $f(\varnothing)=0$；

e) $f(\Omega)=1$；

f) $0\leqslant f(A)\leqslant 1$。

可以看出，类概率函数与概率函数具有相似的性质。

(3) 不确定性推理模型

我们将所有输入的已知数据的条件部分和结论部分的命题都称作证据。下面分别确定规则的条件部分和结论部分命题的确定性。

定义 5.10　令 A 是规则条件部分的命题，在证据 E' 的条件下，命题 A 和证据 E' 的匹配程度为：

$$MD(A,E')=\begin{cases}1, & \text{如果 } A \text{ 的所有元素都出现在 } E' \text{ 中；}\\ 0, & \text{否则。}\end{cases}$$

定义 5.11　规则条件部分命题 A 的确定性为

$$CER(A)=MD(A,E')f(A)。$$

下面讨论当规则的条件部分为命题的逻辑组合时，整个条件部分的确定性：

若 $A=A_1\wedge A_2\wedge\cdots\wedge A_n$，则

$$\begin{aligned}CER(A)&=CER(A_1\wedge A_2\wedge\cdots\wedge A_n)\\&=\min\{CER(A_1),CER(A_2),\cdots,CER(A_n)\};\end{aligned}$$

若 $A=A_1\vee A_2\vee\cdots\vee A_n$，则

$$\begin{aligned}CER(A)&=CER(A_1\vee A_2\vee\cdots\vee A_n)\\&=\max\{CER(A_1),CER(A_2),\cdots,CER(A_n)\}。\end{aligned}$$

下面考虑规则结论部分的命题的确定性。如果有规则

IF E THEN $H=\{h_1,h_2,\cdots,h_k\}$，$CF=\{c_1,c_2,\cdots,c_k\}$，

且 $\Omega=\{h_1,h_2,\cdots,h_k\}$，则 Ω 上的基本概率分配函数为

$$\begin{aligned}&m(\{h_1\},\{h_2\},\cdots,\{h_k\})\\&\quad=(CER(E)\cdot c_1,CER(E)\cdot c_2,\cdots,CER(E)\cdot c_k),\\&m(\Omega)=1-\sum_{i=1}^{k}[CER(E)\cdot c_i]。\end{aligned}$$

根据这个基本概率分配函数 m 就可以进一步求出结论部分命题的信任函数、似然函数，进而求出类概率函数和确定性。

如果有 n 条规则支持同一命题，总的基本概率分配函数 M 为各规则结论得到的基本概率分配函数的正交和为

$$m := m_1 \oplus m_2 \oplus \cdots \oplus m_n.$$

证据理论的缺点是要求辨别框中的元素满足相互排斥的条件，在实际系统中不易满足。而且，基本概率分配函数要求给的值太多，计算比较复杂。

5.2.4 模糊推理

不确定性产生的原因有多种：随机性、模糊性、多义性等等。处理随机性的理论基础是概率论，处理模糊性的基础是模糊集合理论。

Zadeh 于 1965 年提出了模糊集合理论，后来于 1978 年又发展出模糊逻辑和可能性理论。这里，我们简要介绍几种模糊知识推理模型。

在经典集合论中，一个元素是否属于某个集合可以用一个特征函数来表示。如果元素 x 是集合 A 的一个元素，则其特征函数 $\mu_A(x)=1$，否则，$\mu_A(x)=0$。但是在现实中，事物的界限往往不是清晰的，无法将一些元素确定地划分到某个集合中。如张三身高 1.75 m，我们是将他列入集合"高个"还是"矮个"呢？这就需要扩展经典集合来处理类似的情况。

定义 5.12 模糊集合是带有隶属程度的元素的集合。设 U 是论域，U 上的一个模糊集合 A 由隶属函数 μ_A 表征：

$$\mu_A: U \to [0,1]。$$

设 $x \in U$，则 $\mu_A(x)$ 表示 x 属于 A 的程度，称 $\mu_A(x)$ 为 x 关于模糊集合 A 的隶属度。

U 上的一个模糊集合 A 也可以用序偶集来表示：

$$A = \{(x, \mu_A(x)) \mid x \in U\}。$$

Zadeh 还提出了一种更方便的表示方法：

(1) 当 U 为有限集 $\{x_1, x_2, \cdots, x_n\}$ 时，模糊集合 A 可用和式记为

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \cdots + \mu_A(x_n)/x_n$$

$$= \sum_{i=1}^{n} \mu_A(x_i)/x_i$$

当 $\mu_A(x_i)=0$ 时，上式中相应的项 $\mu_A(x_i)/x_i$ 可以省略。

(2) 当 U 为无限集时，A 可用积分符号记为

$$A = \int_U \mu_A(x)/x。$$

在模糊集合的抽象表示中，A 通常用积分符号表示。

例 5.5　(1) 设 $U=\mathbf{N}_+$(正整数)，A 表示接近 5 的整数集合，A 可以表示为

$$A=0.2/2+0.4/3+0.7/4+1/5+0.7/6+0.4/7+0.2/8。$$

(2) 模糊集"年轻"可以表示为

$$\text{年轻} = \sum_{x=0}^{120} \left[1+\left|\frac{x}{30}\right|^{2}\right]^{-1}/x,$$

(3) 设 $U=\mathbf{R}$(实数)，A 表示小实数集合，则 A 可以表示为

$$A = \int_{\mathbf{R}} (1+x^2)^{-1}/x。$$

定义 5.13　两个模糊集合 A，B 相等，记为 $A=B$，当且仅当对于 U 中的任意元素 x，有 $\mu_A(x)=\mu_B(x)$。

定义 5.14　模糊集合 A 是 B 的子集，记为 $A\subseteq B$，若 $\forall x\in U$，$\mu_A(x)\leqslant\mu_B(x)$；若 $A\subseteq B$ 且 $\exists x\in U$，使 $\mu_A(x)<\mu_B(x)$，则称 A 为 B 的真子集，记为 $A\subset B$。

定义 5.15　设 A，B 是 U 上的模糊集，定义交集 $A\cap B$、并集 $A\cup B$ 和补集 $\overline{A}$ 为：

$$A\cap B = \int_U \min\{\mu_A(x),\mu_B(x)\}/x,$$

$$A\cup B = \int_U \max\{\mu_A(x),\mu_B(x)\}/x,$$

$$\overline{A} = \int_U (1-\mu_A(x))/x。$$

一般地，论域 U 上的所有模糊集合对于 $\cap$、$\cup$、求补的操作具

有交换律、结合律、幂等律、分配律、吸收律和德摩根律等性质。经典集合中只有排中律对模糊集不成立。

定义 5.16 设 U 为论域 $U_1,U_2,\cdots,U_n$ 的笛卡尔乘积，即 $U=U_1\times U_2\times\cdots\times U_n$，且 $A_1,A_2,\cdots,A_n$ 分别是 $U_1,U_2,\cdots,U_n$ 上的模糊集，则 $A_1,A_2,\cdots,A_n$ 的笛卡尔乘积是 U 上的模糊集，或称模糊子集

$$A_1\times A_2\times\cdots\times A_n$$
$$=\int_U \min\{\mu_{A_1}(x_1),\mu_{A_2}(x_2),\cdots,\mu_{A_n}(x_n)\}/(x_1,x_2,\cdots,x_n)。$$

设 $U_1,U_2,\cdots,U_n$ 是 n 个论域，$U_1,U_2,\cdots,U_n$ 中的一个 n 元模糊关系 R 是 $U_1\times U_2\times\cdots\times U_n$ 上的一个模糊集，设 R 的特征函数为 μ_R，则 R 可以记为

$$\int_{U_1\times U_2\times\cdots\times U_n}\mu_R(x_1,x_2,\cdots x_n)/(x_1,x_2,\cdots x_n)。$$

模糊关系是经典集合论中关系的推广，一个有限论域上的二元模糊关系 $U_1\times U_2$ 可以表示为一个隶属度矩阵的形式

$$\boldsymbol{R}=\begin{pmatrix}\mu_{11} & \cdots & \mu_{1n}\\ \vdots & & \vdots\\ \mu_{m1} & \cdots & \mu_{mn}\end{pmatrix},$$

其中：$U_1=\{x_1,x_2,\cdots,x_m\}$，$U_2=\{y_1,y_2,\cdots,y_n\}$，$\mu_{ij}=\mu_R(x_i,x_j),(i=1,\cdots,m;j=1,\cdots,n)$。

模糊关系的包含、相等、交、并、补等关系和操作与一般模糊集的概念一样。下面我们讨论模糊关系的合成操作。

定义 5.17 设 A 是 $U_1\times U_2$ 上的一个模糊关系，B 是 $U_2\times U_3$ 上的一个模糊关系，则 A,B 的合成关系 $A\circ B$ 为

$$A\circ B=\int_{U_1\times U_3}\max_{y\in U_2}\min\{\mu_A(x,y),\mu_B(y,z)\}/(x,z)。$$

在有限域上，$A\circ B$ 可以表示为一个隶属度矩阵的形式。设 $|U_1|=m,|U_2|=l,|U_3|=n$，记 $R=A\circ B$，则

$$\mu_R(x_i,z_j)=\max_{1\le k\le l}\min\{\mu_A(x_i,y_k),\mu_B(y_k,z_j)\}/(x_i,z_j),$$

$$(i=1,\cdots,m;\ j=1,\cdots,n)。$$

例 5.6　设 $\boldsymbol{A}=\begin{bmatrix}0.3 & 0.7 & 0.2\\ 1.0 & 0.0 & 0.4\\ 0.0 & 0.5 & 1.0\\ 0.6 & 0.7 & 0.8\end{bmatrix}$，$\boldsymbol{B}=\begin{bmatrix}0.1 & 0.9\\ 0.9 & 0.1\\ 0.6 & 0.4\end{bmatrix}$，

则

$$\boldsymbol{R}=\boldsymbol{A}\circ\boldsymbol{B}=\begin{bmatrix}0.7 & 0.3\\ 0.4 & 0.9\\ 0.6 & 0.4\\ 0.7 & 0.6\end{bmatrix},$$

其中：

$$\begin{aligned}R_{12}&=\max_{1\le k\le 3}\min\{A_{1k},B_{k2}\}\\&=\max\{\min\{A_{11},B_{12}\},\min\{A_{12},B_{22}\},\min\{A_{13},B_{32}\}\}\\&=\max\{\min\{0.3,0.9\},\min\{0.7,0.1\},\min\{0.2,0.4\}\}\\&=\max\{0.3,0.1,0.2\}\\&=0.3。\end{aligned}$$

在模糊逻辑系统中，为了方便使用，常常使用语言变量这一概念。一个语言变量的取值范围是一个项目集合 $T(\Psi)$，$T(\Psi)$中的元素一般可以分为基本语言项和含修饰词的语言项，形式上表示为

$$T(\Psi)=\{t_1,\cdots,t_k\}\cup m_1\{t_1,\cdots,t_k\}\cup\cdots\cup m_n\{t_1,\cdots,t_k\},$$

其中∪是一般的集合并操作，t_i 是基本语言项，m_i 为语言修饰词。

例 5.7　论域 U 为[0,120]的年龄值，语言变量为“年龄”，它的项目集可以表示为

$$\begin{aligned}T(\text{年龄})=&\{\text{年轻},\text{老}\}\cup\text{非常}\{\text{年轻},\text{老}\}\cup\\&\text{差不多}\{\text{年轻},\text{老}\}\cup\text{不}\{\text{年轻},\text{老}\}\\=&\{\text{年轻},\text{老},\text{非常年轻},\text{非常老},\text{差不多年轻},\\&\text{差不多老},\text{不年轻},\text{不老}\}\end{aligned}$$

这里，无论是基本语言项还是非基本语言项，它们每一个都代表一个模糊子集。设基本语言项 t 的隶属函数为 w，则非基本语言项的隶属函数通常可以从 w 计算出来。扎德(Zadeh)给出了一组经验公式如下：

非常 t 的隶属函数：w^2；

差不多 t 的隶属函数：$\sqrt{w}$；

不 t 的隶属函数 $1-w$。

模糊推理实质上是在模糊集合上进行操作。在同一论域中，证据的“与”、“或”、“非”运算通常可以对应于模糊集的交、并、补操作；不同论域中的逻辑运算一般需要拓广至笛卡尔意义下的相应操作。

逻辑推理通过逻辑蕴涵来实现。设 U 和 V 为两个论域，A 是 U 上的模糊子集，B 是 V 上的模糊子集，那么，规则

IF A THEN B

可以定义为 $U\times V$ 上的一个模糊关系：$(A\times B)\cup(\overline{A}\times V)$，或等价地表示为

$$A\rightarrow B=\int_{U\times V}\max\min\{\mu_A(x),\mu_B(y)\},1-\mu_A(x)\}/(x,y)。$$

在模糊推理中，推理规则中前件与结论以及前件所对应的事实都可能是模糊集。设 A_1 和 A_2 是 U 上的两个模糊子集，B 是 V 上的模糊子集，那么

$$\begin{array}{ll} A_1 & \text{模糊前件} \\ \underline{A_2\rightarrow B} & \underline{\text{模糊蕴涵}} \\ A_1\circ(A_2\rightarrow B) & \text{模糊结论} \end{array},$$

模糊结论等价于 $A_1\circ((A_2\times B)\cup(\overline{A_2}\times V))$，或写成下面的形式：

$$\int_V\max_{x\in U}\min(\mu_{A_1}(x),\mu_{A_2\rightarrow B}(x,y))/y。$$

例 5.8 设论域 $U=\{1,2,3\}$，U 上的模糊集合“小”为：

小 $=1/1+0.4/2$。

U 上的模糊集合"有点小"为：

有点小$=1/1+0.4/2+0.2/3$。

U 上的模糊集合"大"为：

大$=0.4/2+1/3$。

规则 R：IF　x 是小的　THEN　y 是大的，

规则 R 的模糊关系可以表示为

$$\boldsymbol{R}=\begin{bmatrix}0.0 & 0.4 & 1.0\\ 0.6 & 0.6 & 0.6\\ 1.0 & 1.0 & 1.0\end{bmatrix},$$

其中：$R_{11}=\max(\min(1,0),1-1)$

$=\max(0,0)$

$=0$，

$R_{12}=\max(\min(1,0.4),1-1)$

$=\max(0.4,0)$

$=0.4$，

$R_{13}=\max(\min(1,1),1-1)$

$=\max(1,0)$

$=1$.

$R_{21}=\max(\min(0.4,0),1-0.4)$

$=\max(0,0.6)$

$=0.6$，

$R_{22}=\max(\min(0.4,0.4),1-0.4)$

$=\max(0.4,0.6)$

$=0.6$，

$R_{23}=\max(\min(0.4,1),1-0.4)$

$=\max(0.4,0.6)$

$=0.6$，

$R_{31}=\max(\min(0,0),1-0)$

$=\max(0,1)$

$=1$,

$R_{32}=\max(\min(0,0.4),1-0)$

$=\max(0,1)$

$=1$,

$R_{33}=\max(\min(0,1),1-0)$

$=\max(0,1)$

$=1$,

有点小 $\circ \boldsymbol{R}=(1,0.4,0.2)\circ\begin{Bmatrix}0.0 & 0.4 & 1.0\\ 0.6 & 0.6 & 0.6\\ 1.0 & 1.0 & 1.0\end{Bmatrix}$

$=(\max\{\min(1,0),\min(0.4,0.6),\min(0.2,1)\},$

$\max\{\min(1,0.4),\min(0.4,0.6),\min(0.2,1)\},$

$\max\{\min(1,1),\min(0.4,0.6),\min(0.2,1)\})$

$=(\max\{0,0.4,0.2\},\max\{0.4,0.4,0.2\},$

$\max\{1,0.4,0.2\})$

$=(0.4,0.4,1)$,

它可近似为“有点大”.于是,我们有

模糊前件:x 有点小,

模糊蕴涵:IF x 是小的 THEN y 是大的,

模糊结论:y 有点大。

模糊集合理论特别适合于对事物各方面的不确定性进行描述。但是,模糊集合理论对不确定性描述的细致性、充分性也导致了其计算的复杂性,现在也已经有了一些实用的简化模糊推理系统。

5.3 决策表的不确定性度量

我们在第 2 章中已经对决策表进行了讨论。决策表是一个知

识表达系统 $S=(U,R,V,f)$，$R=C\cup D$ 是属性集合，子集 C 和 D 分别称为条件属性集和结果属性集，$D\neq\varnothing$。在决策表中也可能存在一定的不确定性信息。

决策表中包含的不确定性主要是指决策表中包含冲突(矛盾)样本的情况，即两个样本的条件属性值相同而决策(分类)属性值不同。这种不一致的产生，主要有三种可能性：

(1) 条件属性不充分，根据所采用的条件属性不能对样本进行正确的分类，必须增加额外的条件属性才能够正确区分样本；

(2) 样本属性值的测量或记录不准确；

(3) 在决策表的预处理过程中产生了冲突(如在离散化过程中可能会产生不精确的样本从而产生冲突)。

前两种情况是由于现实问题本身所导致的。对于一些未知领域的情况，我们可能还没有全面认识领域中的各种情况，对于某些结果，我们还无从知道其原由是什么，或者由于度量某些属性值所需要花费的代价太高，我们没有去度量这些属性，这是导致决策表包含不确定性信息的一个原因，它可以通过我们增加对领域知识的掌握和增加度量范围来解决。对于测量和记录的不准确，这是难免的，特别是大量数据测量和记录的时候，有时是由于测量仪器精度所带来的，有时是我们记录精度和记录失误所引起的。而且，我们在使用一些以前的记录数据的时候，数据的记录还可能是不完整的，即决策表中还有一些数据是未知的。

预处理过程中所产生的不确定性，严格说来，是可以杜绝的，但是，这样做，有时代价太高，预处理达不到我们所希望的结果，不得不采取折中的办法，允许在预处理过程中适量的引入不确定性信息。

对于一个知识系统而言，随着处理过程的进行，如果其不确定性程度减少了，我们实际上就是过滤掉了其中的一部分不确定性知识，这种情况下，我们丢失了部分信息；反之，如果其不确定性程度增加了(保持其确定信息)，实际上就是我们在系统中增加了不

确定性信息。在知识获取研究中，我们为了达到高的适应性能力，往往需要在系统中增加不确定性信息，也就是通过降低系统的确定性来提高系统的适应性。总之，无论是希望提高系统的不确定性，还是希望降低系统的不确定性，我们都需要对系统的不确定性程度进行度量，这是研究问题的基础。

在第 2 章中，我们介绍了知识的分类概念。在第 3 章里，我们又介绍了近似分类的精度和近似分类质量的概念，它们都是描述属性集合对概念簇进行分类的能力的概念。这里，我们将以近似分类为基础讨论决策表知识的不确定性度量问题。

当一个决策表中所有的条件属性的分类都不存在不一致的情况时，那么我们说决策表是确定的，可以从决策表中得到完全确定的规则。当其中有一些分类发生了冲突，那么与此分类有关的信息就不可能是完全确定的，从中得到的相关知识也应该是不确定的。基于上述的考虑，可以得到度量决策表的不确定性的公式。

对于 $S=\langle U,R,V,f\rangle$，$R=C\cup D$，C 为条件属性集，D 为决策属性集，分类 $E_i\in U|\mathrm{IND}(C)$ $(i=1,\cdots,m)$，m 为条件属性 C 所决定的分类的数量，$\{X_1,X_2,\cdots,X_n\}$ 是 U 上由决策属性集决定的概念簇(对 U 的一个划分)，则对于任意分类 $E\in U|\mathrm{IND}(C)$，其对于决策属性分类的确定性程度为：

$$\mu_{\max}(E)=\max(\{|E\cap X_i|/|E|:E_i\in U|\mathrm{IND}(D)\}),$$

那么，决策表 A 的不确定性可定义为：

$$\mu_{\mathrm{uncer}}(S)=1-\sum_{i=1}^{m}\frac{|E_i|}{|U|}\cdot\mu_{\max}(E_i)$$

$$=\frac{1}{|U|}\sum_{i=1}^{m}(|E_i|\cdot\mu_{\max}(E_i)),$$

显然，$0\leqslant\mu_{\mathrm{uncer}}(S)<1$。

当决策表不完整(即决策表中包含未知属性值)时，对于决策表中同一属性中未知的值可以假设为不同于已知属性值的各不相同的值，这样得到的 $\mu_{\mathrm{uncer}}(S)$ 为最小值。当对决策表进行完整化处

理(补齐)后，则其确定性应该下降，$\mu_{\text{uncer}}(S)$值变大。这个结论可以通过下面的定理得证。

定理 5.1　决策表 $S=(U,R,V,f)$，其中的属性值都是已知的，其不确定性程度为 $\mu_{\text{uncer}}(S)$，如果决策表中某个样例 a 在某一属性 $c\in C$ 上的取值与其他所有样例不同，即

$$\forall b\in U(b\neq a\Rightarrow c(a)\neq c(b)),$$

则改变 $c(a)$的值不会减少决策表的不确定性。

证明　不妨设将 $c(a)$的值改变为 $c(a^*)$，得到的新的决策表记为 S^*。

由于样例 a 在某一属性 $c\in C$ 上的取值与其他所有样例不同，所以在条件属性 C 所决定的对 S 的分类上一定存在一个不分明关系 $E_j=\{a\}$，$j\leqslant m$。为了下面证明过程中论证的方便，不妨设 $j=m$，即 $E_m=\{a\}$。

1. 如果 $\neg\exists\, b\in U(c(a^*)=c(b))$，则改变 $c(a)$的值不影响决策表中的分类情况，有 $\mu_{\text{uncer}}(S)=\mu_{\text{uncer}}(S^*)$。

2. 如果 $\exists\, b\in U(c(a^*)=c(b))$，假设 $b\in E_k$，$k<m$。可以分两种情况讨论：

(1) 如果样例 a^* 在新的决策表 S^* 中被合并分到 b 的类中，则

$$\mu_{\text{uncer}}(S^*)=1-\sum_{i=1}^{m-1}|E_i^*|\cdot\mu_{\max}(E_i^*)\Big/|U|$$

$$=1-\Big(\sum_{i=1}^{m}(|E_i|\cdot\mu_{\max}(E_i))-|E_m|\cdot\mu_{\max}(E_m)$$

$$-|E_k|\cdot\mu_{\max}(E_k)+|E_k^*|\cdot\mu_{\max}(E_k^*)\Big)\Big/|U|$$

$$\mu_{\text{uncer}}(S^*)-\mu_{\text{uncer}}(S)=1-\Big(\sum_{i=1}^{m}(|E_i|\cdot\mu_{\max}(E_i))-|E_m|\cdot\mu_{\max}(E_m)$$

$$-|E_k|\cdot\mu_{\max}(E_k)+|E_k^*|\cdot\mu_{\max}(E_k^*)\Big)\Big/|U|$$

$$-\Big(1-\sum_{i=1}^{m}(|E_i|\cdot\mu_{\max}(E_i))\Big/|U|\Big)$$

$$= \frac{1}{|U|} \cdot [|E_m| \cdot \mu_{\max}(E_m) + |E_k| \cdot \mu_{\max}(E_k) - |E_k^*| \cdot \mu_{\max}(E_c^*)]$$

$$= \frac{1}{|U|} \cdot [1 + |E_k| \cdot \mu_{\max}(E_k) - |E_k + 1| \cdot \mu_{\max}(E_k^*)]$$

$$= \frac{1}{|U|} \cdot [1 + \max\{|E_k \cap X_i| \mid X_i \in U/\mathrm{IND}(D)\} - \max\{|E_k^* \cap X_i| \mid X_i \in U/\mathrm{IND}(D)\}] \geqslant 0$$

等号只有在

$$\max\{|E_k^* \cap X_i| \mid X_i \in U/\mathrm{IND}(D)\} = \max\{|E_k \cap X_i| \mid X_i \in U/\mathrm{IND}(D)\} + 1$$

时成立。

(2) 如果样例 a^* 在新的决策表 S^* 中没有被合并分到 b 的类中，则改变 $c(a)$ 的值不影响决策表中的分类情况，有 $\mu_{\mathrm{uncer}}(S) = \mu_{\mathrm{uncer}}(S^*)$。

5.4 决策规则的不确定性表示与度量

在 Rough 集理论中，决策规则的不确定性也可以用规则的可信度来表示和度量。

定义 5.18 对于决策表 $S = \langle U, R, V, f \rangle$，$R = C \cup \{d\}$ 是属性集合，子集 C 和 $\{d\}$ 分别为条件属性集和决策属性集，决策规则 $A \rightarrow B$ 的可信度 $CF(A \rightarrow B)$ 定义为

$$CF(A \rightarrow B) = \frac{|X \cap Y|}{|X|},$$

其中：$X = \{x \mid x \in U \wedge A_x\}$，$Y = \{x \mid x \in U \wedge B_x\}$。$A_x$ 表示实例 x 的条件属性值满足公式 A，B_x 表示实例 x 的决策属性值满足公式 B，即集合 X 是条件属性值满足公式 A 的实例集合，集合 Y 是决策属性值满足公式 B 的实例集合。

例 5.9 如表 5.1所示的决策表，我们可以得到如下的决策规

则(规则可信度大于 0.55)：

表 5.1　决策表

U	a	b	c	d
1	1	2	3	1 (50x)
2	1	2	1	2 (5x)
3	2	2	3	2 (30x)
4	2	3	3	2 (10x)
5	3	5	1	3 (2x)
6	3	5	1	4 (1x)

$a_1c_3 \rightarrow d_1$　　1,

$a_1c_1 \rightarrow d_2$　　1,

$b_2c_1 \rightarrow d_2$　　1,

$a_2 \rightarrow d_2$　　1,

$b_4 \rightarrow d_2$　　1,

$a_1 \rightarrow d_1$　　50/55 = 0.91,

$a_3 \rightarrow d_3$　　4/5 = 0.8,

$b_5 \rightarrow d_3$　　4/5 = 0.8,

$b_2c_3 \rightarrow d_1$　　50/80 = 0.62,

$b_2 \rightarrow d_1$　　50/85 = 0.59,

$c_3 \rightarrow d_1$　　50/90 = 0.56。

基于这一可信度的概念，T. Mollestad 和 A. Skowron 提出了缺省规则的概念，即预先设定一个决策规则可信度的阈值 $\mu(0<\mu \leqslant 1)$，将可信度大于(和等于)阈值 μ 的规则作为缺省决策规则。

对于决策规则的可信度，我们只能知道利用该规则得到正确结论的概率估计，而忽略了规则在决策表中的覆盖程度，即该决策规则是基于多少实例而得到的，这一信息在不确定性推理中有时是很重要的。下面我们介绍决策规则不确定性的另一种表示方法。

定义 5.19　对于决策表 $S=\langle U,R,V,f\rangle$，$R=C\cup\{d\}$ 是属性

集合，子集 C 和 $\{d\}$ 分别为条件属性集和决策属性集，决策规则 $A \to B$ 的不确定性可以用参数对 (α,β) 来表示，其中 $\alpha=|X\cap Y|$，$\beta=|X|$，规则可以表示为如下形式：

$$A\to B\ |(\alpha,\beta),$$

其中：$X=\{x|x\in U\wedge A_x\}$，$Y=\{x|x\in U\wedge B_x\}$。这里，参数 β 表示了该规则在决策表中的(绝对)覆盖度，α/β 就是该规则的可信度。

这样，例 5.9 中的规则也可以表示为：

$a_1c_3\to d_1$ |(1,1),

$a_1c_1\to d_2$ |(1,1),

$b_2c_1\to d_2$ (1,1),

$a_2\to d_2$ |(2,2),

$b_3\to d_2$ |(1,1),

$a_1\to d_1$ (50,55),

$a_3\to d_3$ (4,5),

$b_5\to d_3$ |(4,5),

$b_2c_3\to d_1$ |(50,80),

$b_2\to d_1$ |(50,85),

$c_3\to d_1$ |(50,90)。

对于由一系列决策规则组成的规则系统，我们同样可以定义系统的不确定性。

对于由一系列决策规则组成的规则系统，这里所说的规则集的不确定性是指对于获得这些规则的某一特定的决策表而言的，脱离了决策表来度量规则集的不确定性没有意义。在决策表的度量中我们用到了分类的精度，因此，在对规则集的不确定性进行度量时，我们也必须知道，对于某一个分类，整个规则集对它的反映情况，从而才能知道整个规则集对原有决策表的反映情况。

对于某一规则 $R(A\to B)$，如果条件属性值满足规则 R 中公式 A 的样本的集合 X 与条件属性全集的某个分类 E 满足 $X\cap E=$

E，则称规则 R 与分类 E 有关。如果 $X \cap E = \varnothing$，则称规则 R 与分类 E 无关。规则 R 与分类 E 有关就表明规则 R 的产生依赖于分类 E。反之，规则 R 与分类 E 无关，表明规则 R 的产生不依赖分类 E。对于一个规则集来说，如果规则集中至少有一条规则与分类 E 有关，则我们说规则集与分类 E 有关。反之，如果规则集中没有一条规则与分类 E 有关，则规则集与分类 E 无关。

为了便于描述，我们定义规则 R 对某一分类 E 的反映程度为

$$\mu_R(E) = \frac{|E \cap X \cap Y|}{|E|},$$

显然，$0 \leqslant \mu_R \leqslant 1$。其中，集合 X 为条件属性值满足规则 R 中公式 A 的样本的集合，集合 Y 为决策属性值满足规则 R 中公式 B 的样本的集合。当规则 R 与分类 E 有关时，如果规则 R 只与分类 E 有关，则 $\mu_R(E)$ 就是此规则的可信度；如果规则 R 与多个分类有关，则 $\mu_R(E)$ 反映了规则 R 所代表的属于分类 E 的样本在分类 E 中所占的比例，即对分类 E 的反映情况。那么，整个规则集对与它有关的分类 E 的反映程度就可以定义为

$$\mu_{rule}(E) = \frac{\sum_{i=1}^{m} \mu_{R_i}(E)}{m},$$

显然，$0 \leqslant \mu_{rule}(E) \leqslant 1$。其中，$m$ 为规则集中与分类 E 有关的规则数，R_i 为与分类 E 有关的规则。$\mu_{rule}(E)$ 实际上就是整个规则集中与分类 E 有关的规则对分类 E 的反映程度的平均值。如果规则集与分类 E 无关，很明显规则集对分类的反映程度 $\mu_{rule}(E) = 0$。

有了上面的定义，就可以得到规则系统的不确定性度量公式

$$\mu_{uncer}(rule) = 1 - \frac{\sum_{i=1}^{n} (|E_i| \cdot \mu_{rule}(E_i))}{\sum_{i=1}^{n} |E_i|},$$

其中 E_i 为与规则集有关的分类，n 为与规则集有关的分类的数目。显然，$0 \leqslant \mu_{uncer}(rule) < 1$。

例 5.10 对于例 5.9的规则集，其不确定性为：

$$\mu_{rule}(E_1)=\frac{\mu_{R_1}(E_1)+\mu_{R_6}(E_1)+\mu_{R_9}(E_1)+\mu_{R_{10}}(E_1)+\mu_{R_{11}}(E_1)}{5}$$

$$=\frac{50/50+50/50+50/50+50/50+50/50}{5}$$

$$=1,$$

$$\mu_{rule}(E_2)=\frac{\mu_{R_2}(E_2)+\mu_{R_7}(E_2)+\mu_{R_8}(E_2)+\mu_{R_{10}}(E_2)}{4}$$

$$=\frac{5/5+5/5+0/5+0/5}{4}$$

$$=0.5,$$

$$\mu_{rule}(E_3)=\frac{\mu_{R_4}(E_3)+\mu_{R_5}(E_3)+\mu_{R_{10}}(E_3)+\mu_{R_{11}}(E_3)}{4}$$

$$=\frac{30/30+0/30+0/30+0/30}{4}$$

$$=0.25,$$

$$\mu_{rule}(E_4)=\frac{\mu_{R_5}(E_4)+\mu_{R_{11}}(E_4)}{2}=\frac{10/10+0/10}{2}$$

$$=0.5,$$

$$\mu_{rule}(E_5)=\frac{\mu_{R_9}(E_5)+\mu_{R_{11}}(E_5)}{2}=\frac{4/5+4/5}{2}$$

$$=0.8,$$

$$\mu_{unc}(rule)=1-\frac{50\times1+5\times0.5+30\times0.25+10\times0.5+5\times0.8}{50+5+30+10+5}$$

$$=0.31。$$

比较决策表的不确定性公式和规则集的不确定性公式，可以看出：

如果规则 R 与分类 E 有关，则 $X\cap E=E$，那么

$$|E\cap X\cap Y|=|E\cap Y|,$$

又因 $\max(|\{E\cap X_i:X_i\in U|\mathrm{IND}(D)\}|)\geqslant|E\cap Y|$，故

$$\max(|\{E\cap X_i:X_i\in U|\mathrm{IND}(D)\}|)\geqslant|E\cap X\cap Y|,$$

则 $$\mu_{\max}(E)\geqslant\mu_R(E),\quad \mu_{\max}(E)\geqslant\mu_{rule}(E)。$$

如果规则集与决策表的所有条件属性分类都有关，那么

$$\sum_{i=1}^{n}|E_i|=|U|,$$

故 $$\frac{\sum_{i=1}^{n}(|E_i| \cdot \mu_{max}(E_i))}{U} \geqslant \frac{\sum_{i=1}^{n}(|E_i| \cdot \mu_{rule}(E_i))}{\sum_{i=1}^{n}|E_i|},$$

因此，$\mu_{uncer}(S) \leqslant \mu_{uncer}(rule)$。

对于一个规则集来说，如果它是一个完备的规则集(能充分反映决策表信息的规则集)，则规则集与每一个条件属性的分类都有关。那么，从我们上面的讨论可以得到下面的结论：

命题 5.1　对于一个完备的规则集来说，它的不确定性必然大于产生它的决策表的不确定性。

这个结论可以作为判断一个规则集是否为完备规则集的一个基本条件，也就是说如果一个规则集的不确定性 $\mu_{uncer}(rule)$ 小于决策表的不确定性 $\mu_{uncer}(S)$，则这个规则集一定不是完备规则集。

从上面的讨论可以看出，决策表度量公式和规则集度量公式不但能够反映决策表在处理过程中不确定性逐渐增加的规律，而且可以比较同一处理阶段不同算法对决策表的不确定性的影响程度，从而为评价知识获取算法的优劣提供了一个依据。并且上面得到的结论也可以作为判断一个规则集是否为完备规则集的一个条件。

第6章　数据预处理

自动知识获取，就是要从领域历史记录数据中获取有用的知识。建立任何实际的知识获取系统，都需要认真研究数据的预处理问题，包括原始数据的采样、收集和整理。不同领域的原始数据，可以通过不同的方法取得，但是，取得的原始数据并不一定就适合直接用于知识获取，通常还需要进行预处理加工，对于原始数据资料中遗漏的信息，需要补充（在基于 Rough 集理论的知识获取中称为决策表补齐），对于原始资料中值域为实数值的数据，还需要进行离散化，因为 Rough 集理论研究的元素对象只能是离散值对象。本章将避过原始数据的采样和收集问题，对决策表的补齐和离散化问题进行探讨。

6.1　决策表补齐

在很多情况下，我们得到的待处理的信息表并不是一个完备的信息表，表中的某些属性值是被遗漏的，我们无从知道其原始值，这也是信息系统不确定性的一种主要原因。对于这种情况，目前主要通过以下途径来对信息表中的遗漏数据进行补齐。一种途径是简单地将存在空缺（遗漏）属性值的实例记录删除，从而得到一个完备的信息表。虽然这种方法不是严格意义上的数据补齐，然而在信息表数据量巨大并且有遗漏属性值的实例记录的数量远远小于信息表所包含的记录数的情况下，这种方法在删除不完整记录之后并不太影响信息表中信息的完整性，是一种可取的处理方法。但是，当信息表中的信息较少、存在遗漏信息的实例相对较多时，这种方法就会严重影响信息表中的信息量，不能采用这种方法

来对信息表的数据进行补齐。第二种途径是将空缺(遗漏)属性值作为一种特殊的属性值来处理,它不同于其他任何属性值.这样,一个不完备的信息表就成了完备信息表。第三种途径是采用统计学原理,根据信息表中其余实例在该属性上的取值的分布情况来对一个遗漏属性值进行估计补充,这样不会影响信息表中包含的信息量。第四种途径是根据 Rough 集理论中数据不可分辨关系来对不完备的数据进行补齐处理。下面,对后两种情况的数据补齐方法进行讨论。

6.1.1　Mean Completer 算法

Mean Completer 算法将信息表中的属性分为数值属性和非数值属性来分别处理。如果遗漏的属性值是数值型的,就根据该属性在其他所有实例的取值的平均值来补充该遗漏的属性值;如果遗漏的属性值是非数值型的,就用该属性在其他所有实例上的取值次数最多的值(即出现频率最高的取值)来补充该遗漏的属性值。

例 6.1　对表 6.1所示的信息表进行数据补齐。

表 6.1　包含不完整信息的信息表

U	Radius	Color	Year	Grade	Sold
	Float	String	Integer	Float	String
1	3.14	Red	1 970	1.0	No
2	2.71	Green	1 492	1.5	Yes
3	10.66	Red	1 814	2.0	Yes
4	0.99	Red			No
5	0.2	Blue	1 776	3.5	No
6		Yellow	1 865	2.5	No
7	4 925.6		1 968	6.0	Yes

在表 6.1 中,实例 6 的 Radius 属性的值未知,可以补齐为(3.14+2.71+10.66+0.99+0.2+4925.6)/6=823.88,实例 4

的 Year 属性的值未知，可以补齐为(1 970+1 492+1 814+1 776+1 865+1 968)/6=1 814，实例 4 的 Grade 属性的值未知，可以补齐为(1.0+1.5+2.0+3.5+2.5+6.0)/6=2.8，实例 7 的 Color 属性的值未知，是非数值属性，而"Red"这个属性值的出现频率最高，故可以补齐为"Red"。

Mean Completer 算法简单直接，在此基础上，还可以演绎出 Conditioned Mean Completer 算法。在 Conditioned Mean Completer 算法中，遗漏属性值的补齐同样是靠该属性在其他实例中的取值求平均得到，但不同的是用于求平均的值并不是从信息表所有实例中取，而是从与该实例具有相同决策属性值的实例中取得。在例 6.1 中(属性 Sold 为决策属性)，如果采用 Conditioned Mean Completer 算法，对于实例 4，与其决策相同的实例有 1，5 和 6，故实例 4 的 Year 属性的值可以补齐为(1 970+1 776+1 865)/3=1 870，Grade 属性的值可以补齐为(1.0+3.5+2.5)/3=2.3。

这两种数据补齐方法，其基本出发点都是一致的，以最大概率可能的取值来填补遗漏的属性值，只是在具体方法上有一点不同。至于补齐的效果如何，不能保证。

6.1.2 Combinatorial Completer 算法

另外一种数据补齐的方法是 Combinatorial Completer 算法，这种算法用空缺属性值的所有可能属性取值来试，并从最终属性约简结果中选择最好的一个作为填补的属性值。这是以约简为目的的数据补齐方法，能够得到好的约简结果；但是，当数据量很大或者遗漏属性值较多时，其计算代价很大。另一种称为 Conditioned Combinatorial Completer 算法，填补遗漏属性值的原则是一样的，不同的只是从决策相同的实例中尝试所有的属性值的可能情况，而不是根据信息表中所有实例进行尝试。Conditioned Combinatorial Completer 算法能够在一定程度上减小 Combinatorial Completer 算法的计算代价。以例 6.1 为例，属性 Year 有{1 970，1 492，1 814，1 776，1 865，1 968} 6 种可能的取

值，而 Grade 有{1.0，1.5，2.0，3.5，2.5，6.0} 6 种可能的取值，这样，采用 Combinatorial Completer 算法就有 $6\times6=36$ 种可能的补齐方案需要尝试，即使采用 Conditioned Combinatorial Completer 算法也有 9 种可能的补齐方案需要尝试。可以看出，在信息表包含不完整数据较多的情况下，可能的尝试方案将巨增。

这几种数据补齐方法，以 Conditioned Mean Completer 算法和 Mean Completer 算法具有较好的可行性，在多数情况下能够得到好的结果。但是，无论什么数据补齐方法，也仅仅是在无法获得真实属性值时所采取的权宜之计。而且，在进行数据补齐的时候，还要考虑信息表中信息的变化，特别是冲突情况的变化，我们要避免在进行数据补齐的过程中人为地引入冲突信息。比如，在采用 Conditioned Mean Completer 算法或者 Mean Completer 算法的时候，如果采用出现频率最高的属性值进行填充会引起该实例和表中其他实例发生冲突（矛盾），就采用次高频率的属性值来填充。

6.1.3　基于 Rough 集理论的不完备数据分析方法（ROUSTIDA）

一个信息系统中的数据基本反映了它所涉及的问题（或领域）的基本特征，尽管系统中可能存在遗失的数据。不完备信息系统中的遗失数据值的填补，应该尽可能反映此信息系统所反映的基本特征以及隐含的内在规律。ROUSTIDA 算法的基本思想是，遗失数据值的填补应使完整化后的信息系统产生的分类规则具有尽可能高的支持度，产生的规则应尽量集中。因为如果规则支持度较小，则产生的规则分布较广，这些规则中就可能隐含着由噪音导致的规则，这正是我们所不希望看到的。换句话说，我们的目标是使具有遗失值的对象与信息系统的其他相似对象的属性值尽可能保持一致，亦即，使属性值之间的差异尽可能保持最小。

可辨识矩阵反映了对象间的属性差异，因此，我们利用可辨识矩阵来作为算法的基础，是一种很自然的想法。为了介绍 ROUSTIDA 算法，我们先对第 4 章中介绍的可辨识矩阵（定义

4.2)进行如下扩充：

定义 6.1 令信息表系统为 $S=(U,A,V,f)$，$A=\{a_i|i=1,\cdots,m\}$ 是属性集，$U=\{x_1,x_2,\cdots,x_n\}$ 是论域，$a_i(x_j)$ 是样本 x_j 在属性 a_i 上的取值。$M(i,j)$ 表示经过扩充的可辨识矩阵中第 i 行 j 列的元素，则经过扩充的可辨识矩阵 M 定义为

$$M(i,j)=\{a_k|a_i\in A\wedge a_k(x_i)\neq a_k(x_j)\wedge a_k(x_i)\neq *\wedge a_k(x_j)\neq *\},$$

其中：$i,j=1,\cdots,n$；"$*$"表示遗失值。

这样，属性值之间的差异概念得到了扩展，使之能适合不完备信息系统。下面，引入对象 x_i 的遗失属性集 MS_i 以及无差别对象集 NS_i 的概念：

定义 6.2 信息表系统 $S=(U,A,V,f)$，$A=\{a_i|i=1,\cdots,m\}$ 是属性集，设 $x_i\in U$，则对象 x_i 的遗失属性集 MAS_i、对象 x_i 的无差别对象集 NS_i 和信息系统 S 的遗失对象集 MOS 分别定义为：

$$MAS_i=\{a_k|a_k(x_i)=*,k=1,\cdots,m\},$$

$$NS_i=\{j|M(i,j)=\varnothing,i\neq j,j=1,\cdots,n\},$$

$$MOS=\{i|MAS_i\neq\varnothing,i=1,\cdots,n\}。$$

由于不完备信息系统中存在多个遗失值和其不同的分布，因此对信息系统遗失数据值的填补往往不是通过对初始扩充可辨识矩阵的一次运算并加以完整化分析就能对所有的遗失值进行补齐的；实际上要经过许多次对扩充差异矩阵的计算和完整化分析，直至终止条件成立。这样，在完整化分析过程中，随着遗失数据值的逐步填补，将产生许多过渡性的临时信息系统，同时为下一步的遗失值的计算，还需计算其相应的扩充可辨识矩阵。

为此，设初始信息系统为 S^0，对象集为 $\{x_i^0\}$，相应的扩充可辨识矩阵为 M^0，x_i 的遗失属性集为 MAS_i^0，无差别对象集为 NS_i^0；第 r 次完整化分析后的信息系统为 S^r，对象集为 $\{x_i^r\}$，相应的扩充可辨识矩阵为 M^r，x_i 的遗失属性集为 MAS_i^r，无差别对象集为 NS_i^r。完整化分析所依赖可辨识矩阵的计算，可由下面定理实施：

定理 6.1　设 $\boldsymbol{M}^{r+1}=(\boldsymbol{M}^{r+1}(i,j))_{n\times n}$，$r=0,1,2,\cdots$，则 $\boldsymbol{M}^{r+1}(i,j)$计算如下：

(1)如果 $MAS_i^r \cup MAS_j^r=\varnothing$，则 $\boldsymbol{M}^{r+1}(i,j)=\boldsymbol{M}^r(i,j)$；

(2)否则，设 $k\in MAS_i^r \cup MAS_j^r$，有

$$\boldsymbol{M}^{r+1}(i,j)=\begin{cases}\boldsymbol{M}^r(i,j)\cup\{k\}, & \text{若}((a_k(x_i^{r+1})\neq *)\wedge(a_k(x_j^{r+1})\neq *)\wedge \\ & (a_k(x_i^{r+1})=a_k(x_j^{r+1}))); \\ \boldsymbol{M}^r(i,j), & \text{否则。}\end{cases}$$

证明　(1)如果 $MAS_i^r \cup MAS_j^r=\varnothing$，显然对象 x_i 和对象 x_j 均没有遗失值，因此，信息系统 S^r 中对象遗失值的填补，不影响 $\boldsymbol{M}^r(i,j)$，故有 $\boldsymbol{M}^{r+1}(i,j)=\boldsymbol{M}^r(i,j)$；

(2)如果 $MAS_i^r \cup MAS_j^r\neq\varnothing$，则有可能由于 S^r 中对象 x_i 或对象 x_j 遗失值的填补，原来为无区别的属性变为有区别的属性，从而改变 $\boldsymbol{M}^r(i,j)$。根据定义 6.1，在 S^{r+1}中只需考虑有遗失值的属性 $k\in MAS_i^r \cup MAS_j^r$，根据定义 6.1，得证。

定理 6.1 告诉我们，计算好初始的扩充可辨识矩阵后，在计算新的信息系统所对应的扩充可辨识矩阵时，不必重新计算，而只需计算上次可辨识矩阵中由于遗失值的填补而引起的局部元素值的修改，从而大大简化了计算复杂性。

下面介绍不完备数据分析算法 ROUSTIDA。

算法 ROUSTIDA：

输入：不完备信息系统 $S^0=\langle U^0,A,V,f^0\rangle$；

输出：完备的信息系统 $S^r=\langle U^r,A,V,f^r\rangle$；

步骤 1：计算初始可辨识矩阵 $\boldsymbol{M}^0$、MAS_i^0 和 MOS^0；令 $r=0$；

步骤 2：

1. 对于所有 $i\in MOS^r$，计算 NS_i^r；

2. 产生 S^{r+1}。

(1) 对于 $i\notin MOS^r$ 有

$$a_k(x_i^{r+1})=a_k(x_i^{r-1}),\quad k=1,2,\cdots,m;$$

(2) 对于所有 $i\in MOS^r$，对所有 $k\in MAS_i^r$ 作循环：

① 如果 $NS_i=1$，设 $j\in NS_i$，若 $a_k(x_j)=*$，则 $a_k(x_i^{r+1})=*$；否则 $a_k(x_i^{r+1})=a_k(x_j)$；

② 否则，

(i) 如存在 j_0 和 $j_1\in NS_i$，满足 $(a_k(x_{j_0})\neq *)\wedge(a_k(x_{j_1})\neq *)\wedge(a_k(x_{j_1})\neq a_k(x_{j_0}))$，则 $a_k(x_i^{r+1})=*$；

(ii) 否则，如果存在 $j_0\in NS_i$，满足 $(a_k(x_{j_0})\neq *)$，则 $a_k(x_i^{r+1})=a_k(x_{j_0})$；

(iii) 否则，$a_k(x_i^{r+1})=*$；

3. 如果 $S^{r+1}=S^r$，结束循环转步骤 3。

否则，计算 M^{r+1}，MAS_i^{r+1} 和 MOS^{r+1}；$r=r+1$；转步骤 2；

步骤 3：如果信息系统还有遗失值，可用取属性值中平均值(数字型)或最多出现值(符号型)的方法处理(当然，也可用其他方法)；

步骤 4：结束。

上述算法中步骤 2 的 3. 中运用了定理 6.1 来计算 M^{r+1}，从而有效地降低了算法的计算复杂性。

下面用一个简单的例子说明上述 ROUSTIDA 算法的实施过程。表 6.2 为一不完备的信息系统 S^0，经过上述完整化算法计算，临时中间信息系统 S^1 和最终完整化后的信息系统 S^2 分别如表 6.3 和表 6.4 所示。

表 6.2 信息系统 S^0

U	a_1	a_2	a_3	a_4
x_1	4	*	1	2
x_2	3	1	*	*
x_3	*	1	1	*
x_4	2	1	4	3
x_5	*	1	3	4

表 6.3 信息系统 S^1

U	a_1	a_2	a_3	a_4
x_1	4	1	1	2
x_2	3	1	*	4
x_3	*	1	1	2
x_4	2	1	4	3
x_5	3	1	3	4

表 6.4　信息系统 S^2

U	a_1	a_2	a_3	a_4
x_1	1	1	1	2
x_2	3	1	3	4
x_3	4	1	1	2
x_4	2	1	4	3
x_5	3	1	3	4

6.2　决策表离散化

运用 Rough 集理论处理决策表时，要求决策表中的值用离散(如整型、字符串型、枚举型)数据表达。如果某些条件属性或决策属性的值域为连续值(如浮点型数表达)，则在处理前必须进行离散化处理，而且，即使对于离散数据，有时也需要通过将离散值进行合并(抽象)得到更高抽象层次的离散值，这是 Rough 集理论中的一类重要研究课题。由于决策表的离散化问题是在 Rough 集理论分析的其他环节(如属性约简、值约简)之前进行，故它属于 Rough 集理论中的预处理问题之一。

6.2.1　离散化问题的描述

决策表 $S=\langle U,R,V,f\rangle$，$R=C\cup\{d\}$ 是属性集合，子集 C 和 $\{d\}$ 分别称为条件属性集和决策属性集，$U=\{x_1,\cdots,x_n\}$ 是有限的对象集合即论域。设决策种类的个数为 $r(d)$。属性 a 的值域 V_a 上的一个断点可以记为 (a,c)，其中 $a\in R$，c 为实数集。在值域 $V_a=[l_a,r_a]$ 上的任意一个断点集合 $\{(a,c_1^a),(a,c_2^a),\cdots,(a,c_{k_a}^a)\}$ 定义了 V_a 上的一个分类 P_a，

$$P_a=\{[c_0^a,c_1^a),[c_1^a,c_2^a),\cdots,[c_{k_a}^a,c_{k_a+1}^a]\},$$

$$l_a=c_0^a<c_1^a<c_2^a<\cdots<c_{k_a}^a<c_{k_a+1}^a=r_a,$$

$$V_a=[c_0^a,c_1^a)\cup[c_1^a,c_2^a)\cup\cdots\cup[c_{k_a}^a,c_{k_a+1}^a]。$$

因此，任意的 $P'=\bigcup_{a\in B}P_a$ 定义了一个新的决策表 $S^p=(U,R,V^p,f^p)$，$f^p(x_a)=i\Leftrightarrow f(x_a)\in[c_i^a,c_{i+1}^a)$，对于 $x\in U$，$i\in\{0,\cdots,K_a\}$，即经过离散化之后，原来的信息系统被一个新的信息系统所代替。

离散化本质上可归结为利用选取的断点来对条件属性构成的空间进行划分的问题，把这个 n（n 为条件属性的个数）维空间划分成有限个区域，使得每个区域中的对象的决策值相同。假设某个属性有 m 个属性值，则在此属性上就有 $m-1$ 个断点可取，随着属性个数的增加，可取的断点数将随着属性值的个数呈几何增长。选取断点的过程也是合并属性值的过程，通过合并属性值，减少属性值的个数，减小问题的复杂度，这也有利于提高知识获取过程中所得到的规则知识的适应度。

6.2.2 离散化问题的分类分析

关于连续数据的离散化并不是一个新课题，早在 Rough 集理论出现前，由于计算机对数值计算的要求，人们就对离散化（或称量化）问题进行了广泛研究，取得了大量研究成果。但是，离散化技术并不是各学科可以完全通用的，实际上它在不同领域中有自己独特的要求和处理方式：比如图像压缩中的量化问题，要求量化后的信息熵最小，这样在熵编码时可以得到最小的压缩图像长度；而运用 Rough 集理论对决策表进行分析和知识获取，是在决策表表达的不可分辨关系上进行的，对离散化的信息熵没有明确要求。

目前国际上针对 Rough 集理论中的离散化问题也提出了一些有价值的研究成果，大致可以分为两类，其中一类基本上是很少或不考虑 Rough 集理论的特殊性，而是把其他学科中的离散化方法借用到 Rough 集理论上来，因此离散化效果并不突出；另一类就是注意到了 Rough 集理论对决策表的特殊要求，采取了结合方法来解决离散化问题。这里，我们主要考虑后一种方法。

Rough 集理论和决策表相结合的离散化算法中，根据进行离

散化处理时是否考虑到信息系统的具体的属性值，可把离散化算法分为两类："非参照性的离散化算法"和"参照性的离散化算法"。"非参照性的离散化算法"在离散化过程中很少考虑或不考虑信息系统中具体的属性值。而"参照性的离散化算法"是参照信息系统中具体的属性值来进行的。根据离散化过程是否改变信息系统原有的不可分辨关系（主要考虑离散化后的信息系统在原信息系统的基础上是否引入新的冲突），可以把离散化算法分为"改变不可分辨关系的离散化算法"和"不改变不可分辨关系的离散化算法"。根据选取断点的过程是从包含所有可能断点的断点集中逐步删除不必要的断点得到离散化结果，还是一开始设断点集为空集，逐步增加候选断点得到离散化结果，又可以把离散化过程分为"逐步删除断点"和"逐步增加断点"的离散化算法。

针对离散化问题，人工智能的研究者提出了很多种方法，有等距离划分、等频率划分、适应离散法等等。但是这些方法需要人为地规定划分的维数，或者需要预先给定一个参数。实际上，Rough 集理论的优势在于它不需要先验知识便可完全从数据或经验中获取知识，生成决策规则。波兰华沙大学与挪威科技大学联合开发的 Rosetta 软件，采用的 Naive Scaler 算法和 Semi Naive Scaler 算法，虽然不需要额外的参数，直接根据信息表或数据库本身进行离散，但是他们一次仅考虑单个属性，因此可以看作是局部的。这两种算法经过离散化之后，得到的新的信息表有可能引入冲突。Nguyen Hung Son，Nguyen Sinh Hoa，Skowron 等人提出了通过对超平面的获取，从而得到 n 维（对应于 n 个属性）空间中的区域划分。但是这种方法比较抽象，不容易被理解。最直观且容易被人理解而且是从全局考虑的是 Nguyen H. S. 和 Skowron 提出的布尔逻辑和 Rough 集理论相结合的离散化算法。这种算法的优点是可以根据给出的信息表求出所有可能的断点集，而且采用任意的一种断点集，得到的新的信息表不会引入冲突。实际上，在求取断点集时，Nguyen H. S. 和 Skowron 在布尔逻辑和 Rough 集理论相

结合的离散化算法中采取的是贪心算法，根据断点值的重要性依次把断点加入到断点集中。这种算法的缺陷在于空间复杂度和时间复杂度都比较高，当信息表的数据量大的情况下，此种算法是不可取的。Nguyen S. H. 和 Nguyen H. S. 还提出了一种有效的能够减小空间复杂度和时间复杂度的算法，通过引入一个数学式子来衡量决策表中各个断点的重要性，根据断点值的重要性依次把断点加入到断点集中。

我们接下来对这些离散化算法进行讨论。

6.2.3 离散化算法介绍

6.2.3.1 等距离划分算法

算法 1 等距离划分(Equal Interval Width)：

此种离散化算法是在每个属性上，根据用户给定的参数来把属性值简单地划分为距离相等的断点段，不考虑每个断点段中属性值个数的多少。假设某个属性的最大属性值为 x_{max}，最小属性值为 x_{min}，用户给定的参数为 k，则断点间隔为 $\delta=(x_{max}-x_{min})/k$。为此得到此属性上的断点为 $x_{min}+i\delta, i=0,\cdots,k$。这些断点之间的距离相等。

6.2.3.2 等频率划分算法

算法 2 等频率划分(Equal Frequency Intervals)：

此种离散化算法是根据用户给定的参数 k 把 m 个对象分成段，每段中有 m/k 个对象。假设某个属性的最大属性值为 x_{max}，最小属性值为 x_{min}，用户给定的参数为 k，则需要将这个属性在所有实例上的取值从小到大进行排列，然后进行平均划分为 k 段即得到断点集。每两个相邻断点之间所包含的属性值的个数是相等的。

算法 1 和算法 2 需要人为地规定划分的维数，或者需要用户预先给定一个参数。根据给定的参数将各属性的值域按等距离或者等频率(属性值出现的频率)划分为几个离散的区间。离散化过程中几乎不考虑信息系统的具体的属性值，一次得到所有的断点值，不考虑信息系统的不可分辨关系，利用得到的断点对原有的信

息系统进行离散化处理往往会改变信息系统原有的不可分辨关系，离散化处理结果的质量没有保障。

6.2.3.3　Naive Scaler 算法

算法 3　Naive Scaler 算法：

Naive Scaler 算法如下：

对每一个属性 $a \in C$，进行下面的过程：

- 根据 $a(x)$ 的值，由小到大排列实例 $x \in U$；
- 从上到下扫描，设 x_i 和 x_j 代表两个相邻的实例：

  如果 $a(x_i) = a(x_j)$，则继续扫描；

  如果 $d(x_i) = d(x_j)$，即决策相同，则继续扫描；

  否则，得到一个断点 c，$c = (a(x_i) + a(x_j))/2$。

Naive Scaler 算法不需要额外的参数，能够根据信息系统或数据库本身进行离散化处理，并在波兰华沙大学与挪威科技大学联合开发的 Rosetta 软件中实现。Naive Scaler 算法对每一个属性，根据属性值由小到大的顺序对决策表中的实例进行排序，然后进行判断，对于两个相邻实例，在属性值和决策值都不同的情况下，选取两个属性值的平均值作为断点值。该算法根据具体的条件属性值和决策属性值选取断点值而不是盲目地选取断点值。不考虑信息系统的不可分辨关系，随着数据库本身的数据的排列情况不同，得到的断点也不相同。更坏的情况是，有些根据不可分辨关系应该选取的断点极可能被忽略掉。Naive Scaler 算法选取断点的过程是一开始设断点集为空集，逐步增加断点得到离散化结果，得到的断点数目很多。

6.2.3.4　Semi Naive Scaler 算法

算法 4　Semi Naive Scaler 算法：

Semi Naive Scaler 算法不同于 Naive Scaler 算法的是，通过对每个属性的每个候选断点进行进一步处理后再决定此断点是否可取，具体的实现如下：

- c 代表属性 a 的一个候选断点，x_i 和 x_j 是断点 c 的两个相

邻的属性值，$x_i<a$，$x_j>a$，

• D_i 代表 x_i 所属的等价类所对应的决策中出现频率最多的决策值的集合，如果有两个以上的决策值出现的频率相同，则$|D_i|>1$；

• 如果 $D_i\subseteq D_j$ 或者 $D_j\subseteq D_i$，则不选取此断点；否则，选取此断点。

由此可见，Semi Naive Scaler 算法所得到的断点数小于 Naive Scaler 算法所得到的断点数，去掉了 Naive Scaler 算法中的一些不必要的断点，是一种求取较少断点的方法。

6.2.3.5 布尔逻辑和 Rough 集理论相结合的离散化算法

算法 5 布尔逻辑和 Rough 集理论相结合的离散化算法：

此算法可以描述如下：

第 1 步 用集合表示所有的属性值；

第 2 步 以符号形式表示所有连续属性值之间的间隔；

第 3 步 对每两条决策不同的记录用上述符号的析取式进行表示；

第 4 步 将以上所有析取式用合取范式形式进行表示；

第 5 步 将合取范式化为析取范式形式；

第 6 步 在析取范式中任意选取一组合取式作为离散化结果。

例 6.2 离散化信息系统表 6.5，其中 d 为决策，V_a 和 V_b 分

表 6.5 一个信息系统表

U	a	b	d
x_1	0.8	2	1
x_2	1	0.5	0
x_3	1.3	3	0
x_4	1.4	1	1
x_5	1.4	2	0
x_6	1.6	3	1
x	1.3	1	1

别为属性 a、b 的值域，U 为论域。

在信息系统表 6.5 中，设 $V_a=[0,2)$、$V_b=[0,4)$，属性 a、b 的值分别为 $a(U)=\{0.8,1.1.3,1.4,1.6\}$ 以及 $b(U)=\{0.5,1,2,3\}$。这样，我们就可得到一个二值变量的集合，集合中每个元素均为两个属性值之间的间隔，即 $VB(S)=\{p_1^a,p_2^a,p_3^a,p_4^a,p_1^b,p_2^b,p_3^b\}$，其中，$VB(S)$ 为算法第 2 步中指出的用符号表示的属性值之间的间隔，p_i^j 表示属性 j 的第 i 个间隔。在本例中 $p_1^a=[0.8,1]$、$p_2^a=[1,1.3]$、$p_3^a=[1.3,1.4]$、$p_4^a=[1.4,1.6]$、$p_1^b=[0.5,1]$、$p_2^b=[1,2]$、$p_3^b=[2,3]$。我们用 $\varphi(i,j)$ 表示算法第 3 步表示的析取式，其中 i、j 分别为两个决策不同的实例，由此我们可以得到：

$$\varphi(2,1)=p_1^a\vee p_1^b\vee p_2^b,$$

$$\varphi(2,4)=p_2^a\vee p_3^a\vee p_1^b,$$

$$\varphi(2,6)=p_2^a\vee p_3^a\vee p_4^a\vee p_1^b\vee p_2^b\vee p_3^b,$$

$$\varphi(2,7)=p_2^a\vee p_1^b,$$

$$\varphi(3,1)=p_1^a\vee p_2^a\vee p_3^b,$$

$$\varphi(3,4)=p_3^a\vee p_2^b\vee p_3^b,$$

$$\varphi(3,6)=p_3^a\vee p_4^a,$$

$$\varphi(3,7)=p_2^b\vee p_3^b,$$

$$\varphi(5,1)=p_1^a\vee p_2^a\vee p_3^a,$$

$$\varphi(5,4)=p_2^b,$$

$$\varphi(5,6)=p_4^a\vee p_3^b,$$

$$\varphi(5,7)=p_3^a\vee p_2^b。$$

将所有析取式表示成合取范式的形式，我们可得到下列表达式：

$$\varphi^S=(p_1^a\vee p_1^b\vee p_2^b)\wedge(p_2^a\vee p_3^a\vee p_1^b)\wedge$$
$$(p_2^a\vee p_3^a\vee p_4^a\vee p_1^b\vee p_2^b\vee p_3^b)\wedge(p_2^a\vee p_1^b)\wedge$$
$$(p_1^a\vee p_2^a\vee p_3^b)\wedge(p_3^a\vee p_2^b\vee p_3^b)\wedge(p_3^a\vee p_4^a)\wedge$$
$$(p_2^b\vee p_3^b)\wedge(p_1^a\vee p_2^a\vee p_3^a)\wedge p_2^b\wedge(p_4^a\vee p_3^b)\wedge$$

$(p_1^a \vee p_2^b)$。

将其化为析取范式的形式，我们得到：

$$\varphi^3 = (p_1^a \wedge p_4^a \wedge p_2^b) \vee (p_2^a \wedge p_3^a \wedge p_1^b \wedge p_4^b) \vee$$
$$(p_3^a \wedge p_1^b \wedge p_2^b \wedge p_3^b) \vee (p_1^a \wedge p_2^a \wedge p_1^b \wedge p_2^b)。$$

对此信息系统来说，此算法得到四种可行的断点集。但是在信息系统的条件属性数目多且每个属性上的属性值也多的情况下，用此算法求取所有可行的断点集是很麻烦也非必须的。很多情况下，我们只需求出最小数目的断点集使得属性空间划分的数目最少。例如：信息系统表 6.5 只有两个属性，所以求断点的过程实际上是求二维空间上的划分，为了能够得到较大的匹配度，我们一般取断点数目最少的断点集，因为在这种情况下，属性空间被划分成的区域数目最少，规则的匹配度增加。用图 6.1 和图 6.2 可以清楚地说明这一点：

在图 6.1（第四种断点集）中，有四个区域中的对象是不能识别的，而在图 6.2（第一种断点集）中，只有一个区域中的对象是不能识

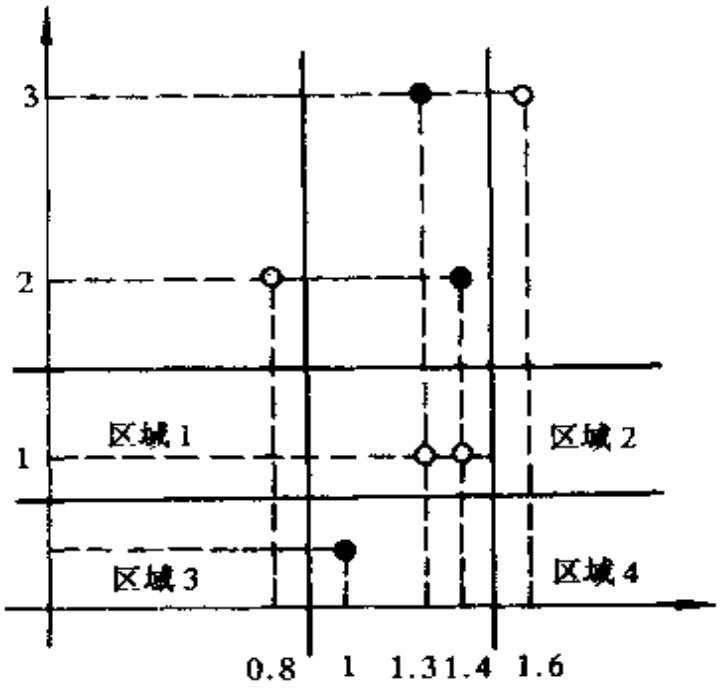


图 6.1 第四种断点集

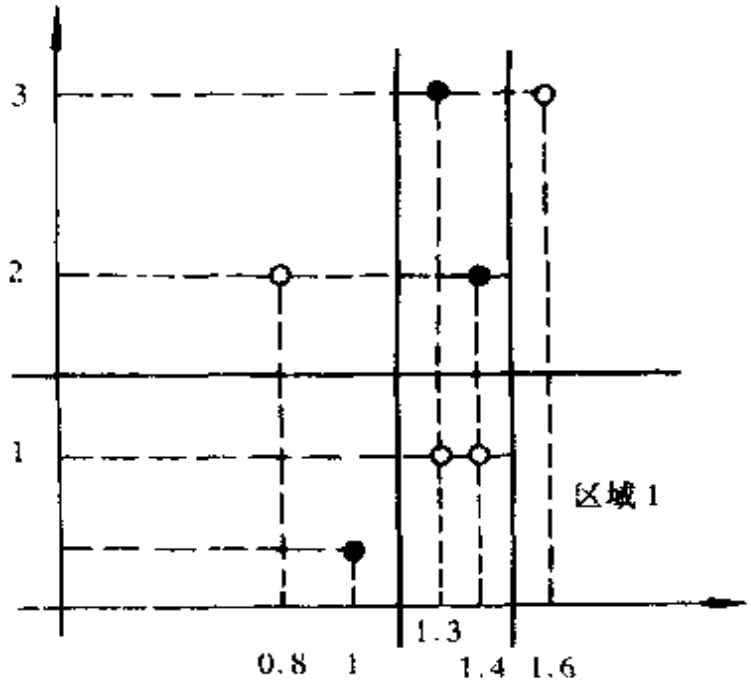


图 6.2　第一种断点集

别的。一般地说，断点数目越少，则能够识别的对象的数目越多。

所以，求断点时，倾向于求最小数目的断点集。由于求最小数目的断点集是 NP 完全问题，因此我们只能寻找近似最优的算法来求得最小数目的断点集。Nguyen H. S. 和 Skowron 采用启发式算法（贪心算法）来求得最小数目的断点集。下面以表 6.5 所示的信息系统为例来说明这个启发式算法。

首先构造一个信息表 $S^*=(U^*,R^*,V^*,f^*)$ 如下：

- $U^*=\{(x_i,x_j)\in U\times U \mid d(x_i)\neq d(x_j)\}$；
- $R^*=\{P_r^a \mid a\in C\}$，$P_r^a$ 是属性 a 的第 r 个断点 $[c_r^a,c_{r+1}^a)$。

对于任意 P_r^a，如果

$$[c_r^a,c_{r+1}^a)\subseteq[\min(a(x_i),a(x_j)),\max(a(x_i),a(x_j))),$$

那么 $P_r^a((x_i,x_j))=1$；否则，$P_r^a((x_i,x_j))=0$。

对于表 6.5 来说，$V_a=[0,2)$，$V_b=[0,4)$，

$p_1^a=[0.8,1)$，　$p_2^a=[1,1.3)$，　$p_3^a=[1.3,1.4)$，

$p_4^a = [1.4, 1.6]$,

$p_1^b = [0.5, 1]$, $p_2^b = [1, 2]$, $p_3^b = [2, 3]$。

构造的新的信息表如表 6.6 所示。

表 6.6 新的信息表

U^*	P_1^a	P_2^a	P_3^a	P_4^a	P_1^b	P_2^b	P_3^b
(x_1, x_2)	1	0	0	0	1	1	0
(x_1, x_3)	1	1	0	0	0	0	1
(x_1, x_5)	1	1	1	0	0	0	0
(x_2, x_4)	0	1	1	0	1	0	0
(x_2, x_6)	0	1	1	1	1	1	1
(x_2, x_7)	0	1	0	0	1	0	0
(x_3, x_4)	0	0	1	0	0	1	1
(x_3, x_6)	0	0	1	1	0	0	0
(x_3, x_7)	0	0	0	0	0	1	1
(x_4, x_5)	0	0	0	0	0	1	0
(x_5, x_6)	0	0	0	1	0	0	1
(x_5, x_7)	0	0	1	0	0	1	0

启发式算法为：

第 1 步　根据原来的信息表 S 构造新的信息表 S^*；

第 2 步　初始化最佳断点集 $CUT = \varnothing$；

第 3 步　选取信息表 S^* 所有列中 1 的个数最多的断点加入到 CUT 中，去掉此断点所在的列和在此断点上值为 1 的所有行；

第 4 步　如果信息表 S^* 中的元素不为空，则转第 3 步；否则停止。此时 CUT 即是所求的断点集。

布尔逻辑和 Rough 集理论相结合的离散化算法是 Rough 集理论的离散化算法在思想上的突破，是基于两个实例的不同的不分明关系而把区分这种分辨关系的任务让其中一个断点去执行。此种算法的思想是首先在保持信息系统的不分明关系不变的前提下，尽量以最少数目的断点集来把所有实例间的不分明关系区分

开。此算法的时间复杂度为 $n^3 \times k$，空间复杂度也为 $n^3 \times k$。所以在条件属性个数多及实例个数多的情况下，此种算法的计算代价仍然很高。

为了求得最小的断点集，贪心算法每次取重要性最高的断点，而断点的重要性是以各列中 1 的数目来衡量的，1 的个数多，则断点的重要性高。当两列 1 的个数相同时，会产生问题。如表 6.6 所示，P_3^a 和 P_2^b 两列中 1 的个数相等，究竟哪一个断点更重要呢？而且先选 P_3^a 和 P_2^b 还会导致最终离散化结果的不同。显然，这个贪心算法对于断点重要性的考虑是不完全的。

在新得到的信息表 S^* 中，如果把每行中 1 的数目累计增加到一个新的列中，就可以得到另一个信息表。如由表 6.6 可以得到表 6.7。各行 1 的数目实际上表示的是可以区分该实例对的断点的数目，也体现了断点的重要性。我们可以把断点所在的列值为 1 的行

表 6.7　新的信息表

U^*	P_1^a	P_2^a	P_3^a	P_4^a	P_1^b	P_2^b	P_3^b	各行 1 的数目
(x_1, x_2)	1	0	0	0	1	1	0	3
(x_1, x_3)	1	1	0	0	0	0	1	3
(x_1, x_5)	1	1	1	0	0	0	0	3
(x_2, x_4)	0	1	1	0	1	0	0	3
(x_2, x_6)	0	1	1	1	1	1	1	6
(x_2, x_7)	0	1	0	0	1	0	0	2
(x_3, x_4)	0	0	1	0	0	1	1	3
(x_3, x_6)	0	0	1	1	0	0	0	2
(x_3, x_7)	0	0	0	0	0	1	1	2
(x_4, x_5)	0	0	0	0	0	1	0	1
(x_5, x_6)	0	0	0	1	0	0	1	2
(x_5, x_7)	0	0	1	0	0	1	0	2
各列 1 的数目	3	5	6	3	4	6	5	

的 1 的数目相加，取和最小的断点为最重要的断点。表 6.7 中，P_3^a 所在的列值为 1 的行的 1 的数目加起来为(3+3+6+3+2+2)=19，P_2^b 所在的列值为 1 的行的 1 的数目加起来为(3+6+3+2+1+2)=17。因此我们可以优先取 P_2^b。由此，我们得到改进的贪心算法。

改进的贪心算法 1：

第 1 步　根据原来的信息表 S 构造新的信息表 S^*；

第 2 步　初始化断点集 $CUT=\varnothing$；

第 3 步　选取所有列中 1 的个数最多的断点加入到 CUT 中；去掉此断点所在的列和在此断点上值为 1 的行；当有一个以上的断点的列 1 的个数相同时，把列对应的断点所在的列值为 1 的行的 1 的数目相加，取和最小的断点。

第 4 步　如果信息表 S^* 中的元素不为空，则转第 3 步；否则停止。此时 CUT 即是得到的断点集。

上面衡量断点的重要性是以列的 1 的个数多少作为标准的。但是从上面的决策表中我们可以从各行的 1 的个数的多少来衡量断点的重要性。仔细观察最后一列的数据，这些值越小，表示能区分该行对应的两个实例的断点越少，因而表明这些断点越重要。最特殊的情况是：当该值取值为 1 时，表明此列对应的断点是唯一能区分该行对应的两个实例的断点，因此在保持不分明关系不变的大前提下进行离散化处理时，这个断点是必不可少的。这样的断点所组成的断点集合构成信息表的断点核，记为 CUT_{core}。

下面的改进算法是从取行为 1 的个数最小的断点开始的。

改进贪心算法 2：

第 1 步　根据原来的信息表 S 构造新的信息表 S^*；

第 2 步　初始化断点集 $CUT=\varnothing$；

第 3 步　首先取断点核加入到 CUT 中，去掉断点核所在的列和在此断点上值为 1 的行；

第 4 步　重新计算各行中 1 的个数，然后把 1 的数目最小的

行的列值为 1 的断点所对应的行的 1 的数目相加，取总和最小的断点加到断点集 CUT 中，去掉此断点所在的列和在此断点上值为 1 的行。

第 5 步　如果信息表 S' 中的元素不为空，则转第 4 步；否则停止。此时 CUT 即是得到的断点集。

两个改进贪心算法，实际上是对断点的重要性程度的度量作了不同的考虑，在一些情况下会取得比较好的结果。

6.2.3.6　基于断点重要性的离散化算法

算法 6　基于断点重要性的离散化算法：

为了有效地减少离散化算法的时间复杂度和空间复杂度，此算法提出了一个衡量断点重要性的式子来对断点进行选择。

将能够被给定的断点 c_m^a 区分开的实例对的个数定义为 $W^X(c_m^a)$。其中：c_m^a 为属性 a 上的第 m 个断点，$1\leqslant m\leqslant n_a$，$n_a$ 为属性 a 的断点总数，$X\subseteq U$ 是由断点 c_m^a 可以分开的实例的集合，U 为实例全集。

决策属性值为 $j(j=1,\cdots,r$，而 r 为决策的种类数）的实例中，属于集合 X 且属性 a 的值又小于断点 c_m^a 值的实例的个数记为

$$l_j^X(c_m^a)=|\{x|x\in X\wedge[a(x)<c_m^a]\wedge[d(x)=j]\}|。$$

决策属性值为 $j(j=1,\cdots,r$，而 r 为决策的种类数）的实例中，属于集合 X 且属性 a 的值又大于断点 c_m^a 值的实例的个数记为：

$$r_j^X(c_m^a)=|\{x|x\in X\wedge[a(x)>c_m^a]\wedge[d(x)=j]\}|。$$

所以有

$$l^X(c_m^a)=\sum_{j=1}^{r}l_j^X(c_m^a)=|\{x|x\in X\wedge a(x)<c_m^a\}|;$$

$$r^X(c_m^a)=\sum_{j=1}^{r}r_j^X(c_m^a)=|\{x|x\in X\wedge a(x)>c_m^a\}|。$$

从而可以得到

$$W^X(c_m^a)=l^X(c_m^a)\cdot r^X(c_m^a)-\sum_{i=1}^{r}l_i^X(c_m^a)\cdot r_i^X(c_m^a)。$$

$W^{Y}(c_m^a)$值越大，则说明断点 c_m^a 的重要性越高，在选取断点时被先选取的可能性也就应该越大。

假设 $X_1, X_2, \cdots, X_m$ 是信息表已经被选取的断点的集合 P 划分得到的等价类，那么能够被断点 $c \notin P$ 区分而不能够被 P 区分的实例对的个数为

$$W_P(c) = W^{Y_1}(c) + W^{Y_2}(c) + \cdots + W^{Y_m}(c),$$

其中 $W^{Y_i}(c)(i = 1, \cdots, m)$表示等价类 Y_i 中能够被断点 c 区分开的实例对的个数。

由此，可以得到下面的离散化算法：

P 为已选取的断点的集合，L 为实例被断点集合 P 所划分成的等价类集合，C 为候选断点的集合。

第 1 步　$P = \varnothing$；$L = \{U\}$；

第 2 步　对每一个 $c \in C$，计算 $W_P(c)$；

第 3 步　选择 $W_P(c)$最大的断点 c_{max}加到 P 中；

$$P = P \cup \{c_{max}\}; C = C \setminus \{c_{max}\};$$

第 4 步　对所有的 $X \in L$，如果 c_{max}把等价类 X 划分为 X_1 和 X_2，那么，从 L 中去掉 X，把等价类 X_1 和 X_2 加到 L 中。

第 5 步　如果 L 中各个等价类中的实例都具有相同的决策，则停止，否则转到第 2 步。

此算法的时间复杂度为 $k \times n$，空间复杂度为 $k \times n$，与算法 5（布尔逻辑和 Rough 集理论相结合的离散化算法）相比有明显减小，它是一种启发式方法。所以此算法在保持不可分辨关系不变的条件下是有效的离散化算法。

算法 3、算法 4、算法 5、算法 6 在对信息表进行离散化的时候，都没有改变信息表的不分明关系。

6.2.3.7　基于属性重要性的离散化算法

算法 7　基于属性重要性的离散化算法（自底向上的离散化算法）：

如第 4 章中所介绍的，属性的重要性是建立在属性的分类能

力上的，为了衡量条件属性的重要性程度，我们可从表中删除这一属性，再来考察信息系统的分类会产生怎样的变化：如果去掉某属性会相应地改变分类，则说明该属性重要（改变的程度越大，重要性越高）；反之说明该属性的重要性低。这里，我们采用定义 4.3 所定义的属性重要性。

例 6.3　计算信息系统表 6.8各条件属性的重要性。其中：$C=\{a,b\}$，$D=\{d\}$。

表 6.8　信息系统表

U	a	b	d
1	0.8	2	1
2	1	0.5	0
3	1.3	3	0
4	1.4	1	1
5	1.4	2	0
6	1.3	1	1
7	1.6	3	1
8	4	3	1

显然，

$U|\mathrm{IND}(a,b)=\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\}\}$；

$U|\mathrm{IND}(a)=\{\{1\},\{2\},\{3,6\},\{4,5\},\{7\},\{8\}\}$；

$U|\mathrm{IND}(b)=\{\{1,5\},\{2\},\{3,7,8\},\{4,6\}\}$。

因此，

$$\begin{aligned}r_C(D)&=\mathrm{card}(\mathrm{POS}_C(D))/\mathrm{card}(U)\\&=\mathrm{card}(\{1,2,3,4,5,6,7,8\})/\mathrm{card}(U)\\&=8/8\\&=1;\end{aligned}$$

$$\begin{aligned}r_{C\setminus a}(D)&=\mathrm{card}(\mathrm{POS}_{C\setminus a}(D))/\mathrm{card}(U)\\&=\mathrm{card}(\{2,4,6\})/\mathrm{card}(U)\end{aligned}$$

$$= 3/8$$
$$= 0.375;$$
$$r_{(C,b)}(D) = \mathrm{card}(\mathrm{POS}_{(C,b)}(D))/\mathrm{card}(U)$$
$$= \mathrm{card}(\{1,2,4,7,8\})/\mathrm{card}(U)$$
$$= 5/8$$
$$= 0.625。$$

属性 a 的重要性为

$$r_C(D) - r_{(C,a)}(D) = 1 - 0.375 = 0.625,$$

属性 b 的重要性为

$$r_C(D) - r_{(C,b)}(D) = 1 - 0.625 = 0.375。$$

由此可见，属性 a 是重要的。

基于属性重要性的离散化算法：

第 1 步　首先根据条件属性的重要性由小到大对条件属性 $V_i(i=1,\cdots,n)$ 进行排序；在属性重要性相同的情况下，按条件属性断点个数由多到少对条件属性进行排序。

第 2 步　对每个属性 $V_i \in A$ 进行下面的过程；

第 3 步　对属性 V_i 中的每一个断点 $C_j(j=1,\cdots,i_j)$，考虑它的存在性：

把信息系统中与 C_j 相邻的两个属性值的较小值改为较大值，如果信息系统不引入冲突，则 $C_{V_i} = C_{V_i} \setminus \{c_j\}$；否则，把修改过的属性值还原。

这个算法通过对每一个断点进行判定，去掉冗余的断点。从而得到简化的信息系统。

下面举例说明这个算法。

例 6.4　对信息系统表 6.8，信息系统的候选断点集为：

$C_a = \{0.9, 1.15, 1.35, 1.5, 2.8\}$；　$C_b = \{0.75, 1.5, 2.5\}$。

经过计算，属性 a 比属性 b 重要，我们从属性 b 的断点开始判定：

首先考虑断点值 0.75，它的相邻属性值为 0.5 和 1，把属性值 0.5 改为 1 后，不引起信息系统的冲突，则断点值 0.5 是多余的，

可以去掉，得到信息系统表 6.9。

接着考虑断点 1.5，1.5 的相邻属性值为 1 和 2，当把 1 改为 2 后实例 x_4 和 x_5 将会冲突，所以断点 1.5 是为保持信息系统的不分明关系所必须具有的断点值，信息系统不变。

再考虑断点 2.5，相邻的属性值为 2 和 3，把属性值 2 改为 3，信息系统不发生冲突，断点值 2.5 是多余的，可以去掉，得到信息系统表 6.10。

表 6.9　信息系统表

U	a	b	d
x_1	0.8	2	1
x_2	1	1	0
x_3	1.3	3	0
x_4	1.4	1	1
x_5	1.4	2	0
x_6	1.3	1	1
x_7	1.6	3	1
x_8	4	3	1

表 6.10　信息系统表

U	a	b	d
x_1	0.8	3	1
x_2	1	1	0
x_3	1.3	3	0
x_4	1.4	1	1
x_5	1.4	3	0
x_6	1.3	1	1
x_7	1.6	3	1
x_8	4	3	1

同样对属性 a 的断点 0.9，1.15，1.35，1.5，2.8 进行判定，求得最后的断点集为 $C_a=\{1.15,1.5\}$，$C_b=\{1.5\}$。最终得到信息系统表 6.11。

由最终的断点集得到的离散化后的信息系统为表 6.12。

这种算法的特点在于由得到的断点集求得的信息系统不会引入冲突。由于断点的判定是从它所属的属性的重要性由小到大的顺序进行的，所以，属性重要性较小的属性的断点被淘汰的可能性更大些。离散化的过程同时又是属性约简的过程（如果一个属性的所有断点都被去掉，则该属性也可以去掉）。之所以从属性重要性小的那些属性开始，是为了避免把属性重要性较大的那些属性的属性值去掉。

表 6.11 信息系统表

U	a	b	d
x_1	1	3	1
x_2	1	2	0
x_3	1.4	3	0
x_4	1.4	2	1
x_5	1.1	3	0
x_6	1.4	1	1
x_7	4	3	1
x_8	4	3	1

表 6.12 信息系统表

U	a	b	d
x_1	0	1	1
x_2	0	0	0
x_3	1	1	0
x_4	1	0	1
x_5	1	1	0
x_6	1	0	1
x_7	2	1	1
x_8	2	1	1

假设某条件属性有 $m+1$ 个属性值，分别为 $v_1, v_2, \cdots, v_{m+1}$，则此属性有 m 个候选断点，所对应的属性值的个数为 $n_1, n_2, \cdots, n_{m+1}$，则 $n_1+n_2+\cdots+n_{m+1}=n$，n 为实例的个数。假设被判定的断点的相邻两个属性值为 v_i, v_{i+1}，算法第 3 步要判定此断点所需的时间复杂度为 $(n_i \times n_{i+1}) \times k < n^2 \times k$，$k$ 为属性个数。存储空间只需要一个辅助的信息系统，其空间复杂度为 $n \times k$。

第 7 章　决策表属性约简

基于 Rough 集理论的知识获取，主要是通过对原始决策表的约简，在保持决策表决策属性和条件属性之间的依赖关系不发生变化的前提下对决策表进行约简（简化），包括属性约简和值约简。本章将对决策表的属性约简从代数集合观点和信息论的信息熵观点进行系统分析，并介绍几种有效的属性约简算法。

7.1　决策表属性约简概述

一个决策表就是一个决策信息系统，表中包含了大量领域样本（实例）的信息。在第 4 章中，我们曾经对决策规则进行了讨论，决策表中的一个样本就代表一条基本决策规则，如果我们把所有这样的决策规则罗列出来，就可以得到一个决策规则集合。但是，这样的决策规则集合是没有什么用处的，因为其中的基本决策规则没有适应性，只是机械地记录了一个样本的情况，不能适应新的、其他的情况。为了从决策表中抽取得到适应度大的规则，我们需要对决策表进行约简，使得经过约简处理的决策表中的一个记录就代表一类具有相同规律特性的样本，这样得到的决策规则就具有较高的适应性。

根据定义 2.1，我们可以进一步讨论决策表中属性的必要性和相应的约简算法。

定义 7.1　设 U 是一个论域，P 是定义在 U 上的一个等价关系簇，$R\in P$。如果 $\mathrm{IND}(P\setminus\{R\})=\mathrm{IND}(P)$，则称关系 R 在 P 中是绝对不必要的（多余的）；否则，称 R 在 P 中是绝对必要的。

绝对不必要的关系在知识库中是多余的，如果将它们从知识

库中去掉，不会改变该知识库的分类能力。相反，若知识库中去掉一个绝对必要的关系，则一定改变知识库的分类能力。

定义 7.2 设 U 为一个论域，P 为定义在 U 上的一个等价关系簇，$R\in P$。如果每个关系 $R\in P$ 在 P 中都是绝对必要的，则称关系簇 P 是独立的；否则，称 P 是相互依赖的。

对于相互依赖的关系簇来说，其中包含有冗余关系，可以对其约简；而对于独立的关系簇，去掉其中任何一个关系都将破坏知识库的分类能力。

定义 7.3 设 U 为一个论域，P 为定义在 U 上的一个等价关系簇，P 中所有绝对必要关系组成的集合称为关系簇 P 的绝对核，记作 $\mathrm{CORE}(P)$

定义 7.4 设 U 为一个论域，P 和 Q 为定义在 U 上的两个等价关系簇且 $Q\subseteq P$，如果

(1) $\mathrm{IND}(Q)=\mathrm{IND}(P)$，

(2) Q 是独立的，

则称 Q 是 P 的一个绝对约简。

如果知识 Q 是知识 P 的绝对约简，那么，U 中通过知识 P 可区分的对象，同样可以用知识 Q 来区分。

在讨论决策表信息系统约简的时候，一个条件属性 A 就对应着一个等价关系(也称不分明关系或不可分辨关系)，即在条件属性 A 上取值的相等关系，它对论域 U 形成一个划分 U/A。决策表的所有条件属性形成条件属性集合 P 对论域 U 的划分 U/P，同时，决策属性集 $D=\{d\}$ 也对论域形成一个划分 U/D。这两个划分形成了条件属性和决策属性在对论域样本分类上的知识。属性约简的目标就是要从条件属性集合中发现部分必要的条件属性，使得根据这部分条件属性形成的相对于决策属性的分类和所有条件属性所形成的相对于决策属性的分类一致，即和所有条件属性相对于决策属性 D 有相同的分类能力。这就是相对约简的概念。

定义 7.5 设 U 为一个论域，P 和 Q 为定义在 U 上的两个等

价关系簇，Q 的 P 正域记为 $\mathrm{POS}_P(Q)$，定义为

$$\mathrm{POS}_P(Q)=\bigcup_{X\in U/Q} \underline{P}(X)。$$

定义 7.6　设 U 为一个论域，P 和 Q 为定义在 U 上的两个等价关系簇，若 $\mathrm{POS}_P(Q)=\mathrm{POS}_{(P-\{r\})}(Q)$，则称 r 为 P 中相对于 Q 可省略的（不必要的），简称 P 中 Q 可省略的；否则，称 r 为 P 中相对于 Q 不可省略的（必要的）。

定义 7.7　设 U 为一个论域，P 和 Q 为定义在 U 上的两个等价关系簇，若 P 中的每一 r 都是 P 中 Q 不可省略的，则称 P 为（相对于）Q 独立的。

定义 7.8　设 U 为一个论域，P 和 Q 为定义在 U 上的两个等价关系簇，若 P 的 Q 独立子集 $S\subseteq P$ 有 $\mathrm{POS}_S(Q)=\mathrm{POS}_P(Q)$，则称 S 为 P 的 Q 约简。

可以记 P 的所有 Q 约简关系簇为 $\mathrm{RED}_Q(P)$。

定义 7.9　设 U 为一个论域，P 和 Q 为定义在 U 上的两个等价关系簇，P 的所有 Q 不可省略原始关系簇称为 P 的 Q 核，记为 $\mathrm{CORE}_Q(P)$。

定义 7.10　设 U 为一个论域，P 和 Q 为定义在 U 上的两个等价关系簇，如果 $\mathrm{POS}_P(Q)=U$，则称论域 U 是 P 上相对于 Q 一致的。

定理 7.1　设 U 为一个论域，P 和 Q 为定义在 U 上的两个等价关系簇，$\mathrm{RED}_Q(P)$ 为 P 的所有 Q 约简关系簇，$\mathrm{CORE}_Q(P)$ 为 P 的 Q 核，则 $\mathrm{CORE}_Q(P)=\bigcap \mathrm{RED}_Q(P)$。

下面再给出与可变精度 Rough 集模型相应的属性集之间依赖、独立以及约简的定义（定义 7.11 至定义 7.13）。

定义 7.11　如果 $K_\beta(C\backslash\{a\},D)=K_\beta(C,D)$，则称属性 a 是属性集 C 中相对于决策属性 D 是依赖的；否则称属性 a 是属性集 C 中相对于决策属性 D 是独立的。

定义 7.12　如果存在条件属性集 $B(B\subseteq C)$ 的真子集 E，使得 $K_\beta(E,D)=K_\beta(B,D)$，则称 B 相对于决策属性 D 是依赖的；否则，称 B 相对于决策属性 D 是独立的。

定义 7.13 决策表条件属性集合 C 的相对约简 C' 是条件属性集合 C 相对于决策属性 D 的最大的独立子集。

下面通过实例对决策表的约简问题加以说明。

如表 7.1 所示的一个关于气象信息的决策表系统。

表 7.1 关于气象信息的决策表系统

U	条件属性				决策属性(d)
	Outlook(a_1)	Temperature(a_2)	Humidity(a_3)	Windy(a_4)	
1	Sunny	Hot	High	False	N
2	Sunny	Hot	High	True	N
3	Overcast	Hot	High	False	P
4	Rain	Mild	High	False	P
5	Rain	Cool	Normal	False	P
6	Rain	Cool	Normal	True	N
7	Overcast	Cool	Normal	True	P
8	Sunny	Mild	High	False	N
9	Sunny	Cool	Normal	False	P
10	Rain	Mild	Normal	False	P
11	Sunny	Mild	Normal	True	P
12	Overcast	Mild	High	True	P
13	Overcast	Hot	Normal	False	P
14	Rain	Mild	High	True	N

令 Q=决策属性集=$\{d\}$，P=条件属性全集=$\{a_1, a_2, a_3, a_4\}$，则

$$IND(P)=\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\},\{9\},\{10\},\{11\},\{12\},\{13\},\{14\}\},$$

$$IND(Q)=\{\{1,2,6,8,14\},\{3,4,5,7,9,10,11,12,13\}\},$$

$$POS_P(Q)=U。$$

因此，论域 U 是 P 上相对于 Q 一致的，这说明该决策表是完全确定的决策表，决策表中不包含不一致信息(样本)。

$$IND(P\setminus\{a_1\})=\{\{1,3\},\{2\},\{4,8\},\{5,9\},\{6,7\},\{10\},\{11\},\{12,14\},\{13\}\},$$

$$\mathrm{IND}(P\backslash\{a_2\})=\{\{1,8\},\{2\},\{3\},\{4\},\{5,10\},\{6\},\{7\},\{9\},\{11\},\{12\},\{13\},\{14\}\},$$

$$\mathrm{IND}(P\backslash\{a_3\})=\{\{1\},\{2\},\{3,13\},\{4,10\},\{5\},\{6\},\{7\},\{8\},\{9\},\{11\},\{12\},\{14\}\},$$

$$\mathrm{IND}(P\backslash\{a_4\})=\{\{1,2\},\{3\},\{4,14\},\{5,6\},\{7\},\{8\},\{9\},\{10\},\{11\},\{12\},\{13\}\}。$$

从而，

$$\mathrm{POS}_{(P-\{a_1\})}(Q)=\{2,5,9,10,11\},$$

$$\mathrm{POS}_{(P-\{a_2\})}(Q)=U=\mathrm{POS}_P(Q),$$

$$\mathrm{POS}_{(P-\{a_3\})}(Q)=U=\mathrm{POS}_P(Q),$$

$$\mathrm{POS}_{(P-\{a_4\})}(Q)=\{1,2,3,7,8,9,10,11,12,13\}。$$

由此可知，属性 a_2，a_3 是相对于决策属性 d 可省略的，但不一定可以同时省略。而属性 a_1 和 a_4 是相对于决策属性 d 不可省略的，因此

$$\mathrm{CORE}_Q(P)=\{a_1,a_4\},$$

进一步，

$$\mathrm{IND}(P\backslash\{a_2,a_3\})=\{\{1,8,9\},\{2,11\},\{3,13\},\{4,5,10\},\{6,14\},\{7,12\}\},$$

$$\mathrm{POS}_{(P\backslash\{a_2,a_3\})}(Q)=\{3,4,5,6,7,10,12,13,14\},$$

故属性 a_2 是条件属性集 $P\backslash\{a_3\}$ 相对于决策属性 d 不可省略的，属性 a_3 也是条件属性集 $P\backslash\{a_2\}$ 相对于决策属性 d 不可省略的。条件属性集 $\{a_1,a_3,a_4\}$ 和 $\{a_1,a_2,a_4\}$ 为相对于决策属性集 $Q=\{d\}$ 独立的，

$$\mathrm{RED}_Q(P)=\{\{a_1,a_3,a_4\},\{a_1,a_2,a_4\}\},$$

$$\mathrm{CORE}_Q(P)=\bigcap\mathrm{RED}_Q(P)=\{a_1,a_3,a_4\}\bigcap\{a_1,a_2,a_4\}$$

$$=\{a_1,a_4\}。$$

去掉表 7.1 中的决策属性列，可以得到一个如表 7.2 所示的信息系统。

令 P = 属性全集 = $\{a_1, a_2, a_3, a_4\}$，根据前面的计算可知

$$\mathrm{IND}(P) \neq \mathrm{IND}(P \setminus \{a_i\}), \quad i = 1, 2, 3, 4.$$

即，在表 7.2 所示的信息系统中，所有的属性都是绝对必要的，去掉任何属性都会改变系统中的知识。

表 7.2 关于气象信息的信息表系统

U	Outlook(a_1)	Temperature(a_2)	Humidity(a_3)	Windy(a_4)
1	Sunny	Hot	High	False
2	Sunny	Hot	High	True
3	Overcast	Hot	High	False
4	Rain	Mild	High	False
5	Rain	Cool	Normal	False
6	Rain	Cool	Normal	True
7	Overcast	Cool	Normal	True
8	Sunny	Mild	High	False
9	Sunny	Cool	Normal	False
10	Rain	Mild	Normal	False
11	Sunny	Mild	Normal	True
12	Overcast	Mild	High	True
13	Overcast	Hot	Normal	False
14	Rain	Mild	High	true

由此，我们可以看出，要根据决策表中的数据信息分析得到条件属性对决策属性的分类(判定)规则，需要研究条件属性集合相对于决策属性的相对约简。

在智能数据分析研究中，原始的决策表信息系统中的知识(条件属性)并不是同等重要的，甚至其中某些条件属性是冗余的。冗余属性的存在，一方面是对资源的浪费(需要存储空间和处理时间)；另一方面，也干扰人们作出正确而简洁的决策。所谓决策表的属性约简，就是要在保持条件属性相对于决策属性的分类能力不变的条件下，删除其中不必要的或不重要的属性。一般来讲，一个决策表的条件属性对于决策属性的相对约简不是唯一的，即对同一个决策表可能存在多个相对约简。因为属性约简的目的是导出

关于决策表的决策规则，约简中属性的多少直接影响着决策规则的繁简和性能．因此，人们往往期望找到具有最少条件属性的约简，即最小约简。然而，S K M Wong 和 W Ziarko 已经证明了找出一个决策表的最小约简是 NP－hard 问题，导致 NP－hard 问题的主要原因是属性的组合爆炸问题。

7.2　决策表属性约简的信息熵表示

我们这里将对 Rough 集理论中的知识(属性集合，即属性集合对论域的划分)作新的理解，建立知识与信息熵的关系。

设 U 为一个论域，P 和 Q 为 U 上的两个等价关系簇(属性集)，可以认为 U 上任一等价关系簇是定义在 U 上的子集组成的 σ 代数上的一个随机变量，其概率分布可通过如下方法来确定。

定义 7.14　设 P，Q 在 U 上导出的划分分别为 X 和 Y，

$$X=\{X_1,X_2,\cdots,X_n\},\quad Y=\{Y_1,Y_2,\cdots,Y_m\},$$

则 P，Q 在 U 的子集组成的 σ 代数上的概率分布为

$$(X:p)=\begin{Bmatrix} X_1 & X_2 & \cdots & X_n \\ p(X_1) & p(X_2) & \cdots & p(X_n) \end{Bmatrix},$$

$$(Y:p)=\begin{Bmatrix} Y_1 & Y_2 & \cdots & Y_m \\ p(Y_1) & p(Y_2) & \cdots & p(Y_m) \end{Bmatrix}。$$

其中：$p(X_i)=\dfrac{|X_i|}{|U|}$，　$i=1,2,\cdots,n$；

$p(Y_j)=\dfrac{|Y_j|}{|U|}$，　$j=1,2,\cdots,m$。

有了知识的概率分布定义后，根据信息论就可以定义知识的熵与条件熵的概念。

定义 7.15　知识(属性集合)P 的熵 $H(P)$定义为

$$H(P)=-\sum_{i=1}^{n}p(X_i)\log(p(X_i))。$$

定义 7.16　知识(属性集合)Q $(U/\mathrm{IND}(Q)=\{Y_1,Y_2,\cdots,$

$Y_m\}$)相对于知识(属性集合)$P(U/\mathrm{IND}(P)=\{X_1,X_2,\cdots,X_n\})$的条件熵 $H(Q|P)$定义为

$$H(Q|P)=-\sum_{i=1}^{n}p(X_i)\sum_{j=1}^{m}p(Y_j|X_i)\log(p(Y_j|X_i))。$$

其中 $p(Y_j|X_i)=|Y_j\cap X_i|/|X_i|$，$i=1,2,\cdots,n$，$j=1,2,\cdots,m$。

定理 7.2 设 U 是一个论域，P 和 Q 是 U 上的两个等价关系簇(属性集合)，若 $\mathrm{IND}(Q)=\mathrm{IND}(P)$，则 $H(Q)=H(P)$。

证明 因为 $\mathrm{IND}(Q)=\mathrm{IND}(P)$，所以 P，Q 在 U 的子集上组成的 σ 代数上的概率分布相同，显然 $H(Q)=H(P)$。

注意：定理 7.2 的逆未必成立。

定理 7.3 设 U 是一个论域，P 和 Q 是 U 上的两个等价关系簇(属性集合)，且 $P\subseteq Q$。若 $H(Q)=H(P)$，则 $\mathrm{IND}(Q)=\mathrm{IND}(P)$。

证明 因为 $P\subseteq Q$，所以 $\mathrm{IND}(Q)\subseteq\mathrm{IND}(P)$。

下面证明 $\mathrm{IND}(P)\subseteq\mathrm{IND}(Q)$。

令 $$U/\mathrm{IND}(P)=\{A_1,A_2,\cdots,A_n\},$$

$$U/\mathrm{IND}(Q)=\{B_1,B_2,\cdots,B_m\}。$$

用反证法，假设 $\mathrm{IND}(P)\subseteq\mathrm{IND}(Q)$ 不成立。

因为 $\mathrm{IND}(Q)\subseteq\mathrm{IND}(P)$，所以对于任意 $B_j(j=1,2,\cdots,m)$ 都存在一个 $A_i(i=1,2,\cdots,n)$，使得 $B_j\subseteq A_i$，即 $U/\mathrm{IND}(P)$ 中的任意等价类都是由 $U/\mathrm{IND}(Q)$ 中的一个或者多个等价类合并而成的。

由于 $\mathrm{IND}(P)\subseteq\mathrm{IND}(Q)$ 不成立，则至少存在一个 $A_{i_0}\in U/\mathrm{IND}(P)$，是由 $U/\mathrm{IND}(Q)$ 中的多个等价类合并而成的。显然，存在一种构造方法，可从 $U/\mathrm{IND}(Q)$ 得到 $U/\mathrm{IND}(P)$：每次将 $U/\mathrm{IND}(Q)$ 中的某两个等价类合并为一个等价类，在得到的新的划分上反复进行这样的过程，则在有限步内一定能够得到 $U/\mathrm{IND}(P)$。

假定将 $U/\mathrm{IND}(Q)$ 中的任意等价类 B_i 和 $B_j(i\neq j$ 且 $i,j=1,2,\cdots,m)$ 合并后得到划分 $U/\mathrm{IND}(Q')$，则

$$H(Q)-H(Q')=-\frac{|B_i+B_j|}{|U|}\log\frac{|B_i+B_j|}{|U|}-\frac{|B_i|}{|U|}\log\frac{|B_i|}{|U|}$$

$$
\begin{aligned}
&-\frac{|B_j|}{|U|}\log\frac{|B_j|}{|U|}\\
>&-\frac{|B_i+B_j|}{|U|}\log\frac{|B_i+B_j|}{|U|}\\
&-\frac{|B_i|}{|U|}\log\frac{|B_i|}{|U|}-\frac{|B_j|}{|U|}\log\frac{|B_j|}{|U|}\\
=&\log\frac{|B_i+B_j|}{|U|}\left(\frac{|B_i+B_j|}{|U|}-\frac{|B_i|}{|U|}-\frac{|B_j|}{|U|}\right)\\
=&0
\end{aligned}
$$

所以，在通过逐步合并 $U|\mathrm{IND}(Q)$ 中的等价类得到 $U|\mathrm{IND}(P)$ 的过程是一个信息熵单调递减的过程，从而有 $H(Q)>H(P)$。这与已知 $H(Q)=H(P)$ 相矛盾。所以假设不成立，应该有 $\mathrm{IND}(P)\subseteq\mathrm{IND}(Q)$ 成立。

综上所述，定理 7.3 成立。

定理 7.4　设 U 是一个论域，P 是 U 上的一个等价关系簇（属性集合），P 中的一个关系 R（属性）是绝对不必要的（多余的），其充分必要条件为 $H(\{R\}|P\backslash\{R\})=0$。

证明　（必要性）设 R 是 P 中不必要的，则

$$\mathrm{IND}(P)=\mathrm{IND}(P\backslash\{R\}),$$

令　$U|\mathrm{IND}(P)=U|\mathrm{IND}(P\backslash\{R\})=\{A_1,A_2,\cdots,A_n\}$，

$U|\mathrm{IND}(\{R\})=\{B_1,B_2,\cdots,B_m\}$。

则任意 $A_i(i=1,2,\cdots,n)$ 中的所有记录在属性 R 上的取值相等，即对于任意 $A_i(i=1,2,\cdots,n)$，都存在一个 $B_j(j=1,2,\cdots,m)$ 使得 $A_i\subseteq B_j$。所以，

$$
\begin{aligned}
&H(\{R\}|P\backslash\{R\})\\
=&-\sum_{i=1}^{n}p(A_i)\sum_{j=1}^{m}p(B_j|A_i)\log(p(B_j|A_i))\\
=&0。
\end{aligned}
$$

（充分性）设 $H(\{R\}|P\backslash\{R\})=0$，

令

$$U|\mathrm{IND}(P\backslash\{R\})=\{A_1,A_2,\cdots,A_n\},$$

$$U|\mathrm{IND}(\{R\})=\{B_1,B_2,\cdots,B_m\},$$

则 $$H(\{R\}\mid P\backslash\{R\})$$

$$=\sum_{i=1}^{n}p(A_i)\sum_{j=1}^{m}p(B_j\mid A_i)\log(p(B_j\mid A_i)),$$

对于任意 $i(i=1,\cdots,n)$，有

$$p(A_i)\sum_{j=1}^{m}p(B_j\mid A_i)\log(p(B_j\mid A_i))\leqslant 0 \text{ 且 } p(A_i)>0,$$

所以 $$\sum_{j=1}^{m}p(B_j\mid A_i)\log(p(B_j\mid A_i))\leqslant 0。$$

而如果存在 $i(i=1,\cdots,n)$，$j(j=1,\cdots,m)$ 使得 $0<p(B_j\mid A_i)<1$，则必将使得 $\sum_{j=1}^{m}p(B_j\mid A_i)\log(p(B_j\mid A_i))<0$，这就必然导致

$$H(\{R\}\mid P\backslash\{R\})=-\sum_{i=1}^{n}p(A_i)\sum_{j=1}^{m}p(B_j\mid A_i)\log(p(B_j\mid A_i))>0,$$

这与已知 $H(\{R\}\mid P\backslash\{R\})=0$ 相矛盾，所以对于任意 $i(i=1,\cdots,n)$，$j(j=1,\cdots,m)$，都有 $p(B_j\mid A_i)=0$ 或 $p(B_j\mid A_i)=1$。也就是说 $U\mid \mathrm{IND}(P\backslash\{R\})$ 是对 $U\mid \mathrm{IND}(\{R\})$ 的细分，故有 $\mathrm{IND}(P\backslash\{R\})=\mathrm{IND}(P)$，即属性 R 在 P 中是不必要的。

这个定理说明不必要的知识(属性)不能够对信息系统的分类提供新的信息，反之亦然。

推论 7.1 P 中的一个关系 R(属性)是绝对必要的充分必要条件为 $H(\{R\}\mid P\backslash\{R\})>0$。

定理 7.5 设 U 是一个论域，P 是 U 上的一个等价关系簇(属性集合)，$Q\subseteq P$ 是 P 的一个约简的充分必要条件为

(1) $H(Q)=H(P)$；

(2) 对任意的 $q\in Q$，有 $H(\{q\}\mid Q\backslash\{q\})>0$。

证明 $Q\subseteq P$ 是 P 的一个约简的充分必要条件为 $\mathrm{IND}(Q)=\mathrm{IND}(P)$，且 Q 是独立的。

由定理 7.3 可知，$\mathrm{IND}(Q)=\mathrm{IND}(P)$ 成立的充分必要条件为 $H(Q)=H(P)$(因为 $Q\subseteq P$)。

由定理 7.4 可知：Q 是独立的成立的充分必要条件为对于任

意的 $q\in Q$，有 $H(\{q\}|Q\backslash\{q\})>0$。

故定理 7.5 成立。

由上述定理可知，对于属性约简而言，信息熵表示形式与前一节的代数表示是等价的，我们也可以从信息熵的角度来研究属性约简问题。

上述定理还仅仅是针对一般信息表而言的。对于决策表这样的特殊信息表，可以有如下定理成立。

定理 7.6　设 U 是一个论域，P 是 U 的一个条件属性集合，d 为决策属性，且论域 U 是在 P 上相对于 $\{d\}$ 一致的，则 P 中的一个属性 R 是 P 相对于决策属性 d 不必要的(多余的)，其充分必要条件为 $II(\{d\}|P)=H(\{d\}|P\backslash\{R\})$。

证明　首先，令

$$U|\mathrm{IND}(P)=\{X_1,X_2,\cdots,X_n\},$$

$$U|\mathrm{IND}(\{d\})=\{Y_1,Y_2,\cdots,Y_m\}。$$

因为论域 U 是在 P 上相对于 $\{d\}$ 一致的，即 $\mathrm{POS}_P(\{d\})=U$，所以 $U|\mathrm{IND}(P)$ 是 $U|\mathrm{IND}(\{d\})$ 的细分，

$$U|\mathrm{IND}(P+\{d\})=U|\mathrm{IND}(P)=\{X_1,X_2,\cdots,X_n\},$$

$$H(\{d\}|P)=-\sum_{i=1}^{n}p(X_i)\sum_{j=1}^{m}p(Y_j|X_i)\log(p(Y_j|X_i))=0。$$

(必要性)　假设属性 R 是 P 相对于决策属性 d 不必要的，则

$$\mathrm{POS}_{P\backslash\{R\}}(\{d\})=\mathrm{POS}_P(\{d\})=U,$$

所以，$U|\mathrm{IND}(P\backslash\{R\})$ 是 $U|\mathrm{IND}(\{d\})$ 的细分。

令　$U|\mathrm{IND}(P\backslash\{R\}+\{d\})=U|\mathrm{IND}(P\backslash\{R\})$

$$=\{Z_1,Z_2,\cdots,Z_k\},$$

$$II(\{d\}|P\backslash\{R\})$$

$$=-\sum_{i=1}^{k}p(Z_i)\sum_{j=1}^{m}p(Y_j|Z_i)\log(p(Y_j|Z_i))=0,$$

故　$$H(\{d\}|P)=H(\{d\}|P\backslash\{R\})。$$

(充分性)　假设 $\mathrm{POS}_{P\backslash\{R\}}(\{d\})\neq U=\mathrm{POS}_P(\{d\})$，

令 $U/\mathrm{IND}(P\setminus\{r\})=\{Z_1,Z_2,\cdots,Z_k\}$，则至少存在 $Z_i(Z_i\in U/\mathrm{IND}(P\setminus\{r\}))$，$Y_{j_1}(Y_{j_1}\in U/\mathrm{IND}(\{d\}))$ 和 $Y_{j_2}(Y_{j_2}\in U/\mathrm{IND}(\{d\}))$，$Y_{j_1}\neq Y_{j_2}$，使得

$$Z_i\cap Y_{j_1}\neq\varnothing \text{ 且 } Z_i\cap Y_{j_2}\neq\varnothing,$$

则有

$$H(\{d\},P\setminus\{r\})=-\sum_{i=1}^{k}p(Z_i)\sum_{j=1}^{m}p(Y_j|Z_i)\log(p(Y_j|Z_i))>0,$$

这与 $H(\{d\}|P\setminus\{r\})=H(\{d\}|P)=0$ 相矛盾。故假设 $\mathrm{POS}_{P\setminus\{r\}}(\{d\})\neq U$ 不成立。因此 $\mathrm{POS}_{P\setminus\{r\}}(\{d\})=U=\mathrm{POS}_P(\{d\})$，根据定义 7.6 知属性 r 是 P 相对于决策属性 d 不必要的。

因此，定理 7.6 成立。

定理 7.7 设 U 是一个论域，P 是 U 的一个条件属性集合，d 为决策属性，且论域 U 是在 P 上相对于 $\{d\}$ 一致的，则 P 是相对于决策属性 d 独立的，其充分必要条件为对于 P 中任意属性 R 都有 $H(\{d\}|P)\neq H(\{d\}|P\setminus\{R\})$ 成立。

定理 7.8 设 U 是一个论域，P 是 U 的一个条件属性集合，d 为决策属性，且论域 U 是在 P 上相对于 $\{d\}$ 一致的，则 $Q\subseteq P$ 是 P 相对于决策属性 d 的一个约简的充分必要条件为

(1) $H(\{d\}|Q)=H(\{d\}|P)$；

(2) Q 是相对于决策属性 d 独立的。

定义 7.17 设 $T=\langle U,R,V,f\rangle$ 是一个决策表系统，其中 $R=C\cup D$，C 是条件属性集合，$D=\{d\}$ 是决策属性集合且 $A\subseteq C$，则对于任意属性 $a\in C\setminus A$ 的重要性 $\mathrm{SGF}(a,A,D)$ 定义为

$$\mathrm{SGF}(a,A,D)=H(D|A)-H(D|A\cup\{a\});$$

若 $A=\varnothing$，则 $\mathrm{SGF}(a,A,D)=H(D)-H(D|\{a\})$，称为属性 a 和决策 D 的互信息，记为 $I(a,D)$。

$\mathrm{SGF}(a,A,D)$ 的值越大，说明在已知 A 的条件下，属性 a 对于决策 D 就越重要。

属性重要性是 Rough 集理论中很多运算都要涉及的基本概

念，定义 1.3 和定义 7.17 分别给出了 Rough 集理论中属性重要性概念的代数定义和信息熵定义。实际上，这两种定义具有互补的特性：属性重要性的代数定义考虑的是该属性对论域中确定分类子集的影响，而属性重要性的信息熵定义考虑的是该属性对于论域中不确定分类子集的影响。如果一个属性的增加，不改变论域中本身已确定分类的实例，且所有本身不能确定分类的实例仍然不能确定分类，只是不确定性有所变化，这样，该属性的重要性在代数定义下为 0，而其在信息熵定义下不为 0。例如，表 7.3 所示的决策信息系统，在代数定义下，属性 c 的重要性 $\mathrm{SGF}(c,\{a,b\},\{d\})=2/9-2/9=0$；而在信息熵定义下，

表 7.3　一个决策表系统

U	a	b	c	e	d
1	1	0	1	1	0
2	0	1	0	1	1
3	0	0	0	0	0
4	0	0	0	1	1
5	0	0	0	1	1
6	0	0	0	1	1
7	0	0	1	1	1
8	0	0	1	0	0
9	0	0	1	1	1

$$
\begin{aligned}
&\mathrm{SGF}(c,\{a,b\},\{d\})\\
&\quad=H(\{d\}\mid\{a,b\})-H(\{d\}\mid\{a,b,c\})\\
&\quad=\frac{3}{9}\left(\frac{1}{3}\log\left(\frac{1}{3}\right)+\frac{2}{3}\log\left(\frac{2}{3}\right)\right)+\frac{4}{9}\left(\frac{1}{4}\log\left(\frac{1}{4}\right)+\right.\\
&\quad\left.\frac{3}{4}\log\left(\frac{3}{4}\right)\right)-\frac{7}{9}\left(\frac{3}{7}\log\left(\frac{3}{7}\right)+\frac{4}{7}\log\left(\frac{4}{7}\right)\right)\\
&\quad=\frac{1}{9}(-\log(3)+\log(4)-2\log(3)-\log(4)+3\log(3)-\\
&\quad 3\log(4)-3\log(3)+3\log(7)-4\log(4)+4\log(7))
\end{aligned}
$$

$$=\frac{1}{9}(-3\log(3)-7\log(4)+7\log(7))$$

$$=\frac{1}{9}\log\left(\frac{7^7}{3^3\times 4^7}\right)=\frac{1}{9}\log\left(\frac{823\,543}{442\,368}\right)>0。$$

反过来，如果在信息熵定义下，属性的重要性 $\mathrm{SGF}(a,A,D)$ 为 0，则该属性的重要性在代数定义下也为 0。

定理 7.9 如果 $H(D|A\cup\{a\})=H(D|A)$，则 $\mathrm{POS}_{A\cup\{a\}}(F)=\mathrm{POS}_A(F)$。

为了证明这个定理，我们首先来证明如下的引理。

引理 7.1 设论域为 U，某个等价关系在 U 上形成的划分为 $A_1=\{X_1,X_2,\cdots,X_n\}$，而 $A_2=\{X_1,\cdots,X_{i-1},X_{i+1},\cdots,X_{j-1},X_{j+1},\cdots,X_n,X_i\cup X_j\}$ 是将划分 A_1 中的某两个等价块 X_i 与 X_j 合并为 $X_i\cup X_j$ 得到的新划分。$B=\{Y_1,Y_2,\cdots,Y_m\}$ 也是 U 上的一个划分，且记

$$H(B|A_1)=-\sum_{i=1}^{n}p(X_i)\sum_{j=1}^{m}p(Y_j|X_i)\log(p(Y_j|X_i)),$$

$$\begin{aligned}H(B|A_2)=&H(B|A_1)\\&-p(X_i\cup X_j)\sum_{k=1}^{m}p(Y_k|X_i\cup X_j)\log(p(Y_k|X_i\cup X_j))\\&+p(X_i)\sum_{k=1}^{m}p(Y_k|X_i)\log(p(Y_k|X_i))\\&+p(X_j)\sum_{k=1}^{m}p(Y_k|X_j)\log(p(Y_k|X_j)),\end{aligned}$$

则 $$H(B|A_2)\geqslant H(B|A_1)。$$

证明

$$\begin{aligned}H_\Delta=&H(B|A_2)-H(B|A_1)\\=&-p(X_i\cup X_j)\sum_{k=1}^{m}p(Y_k|X_i\cup X_j)\log(p(Y_k|X_i\cup X_j))\\&+p(X_i)\sum_{k=1}^{m}p(Y_k|X_i)\log(p(Y_k|X_i))\\&+p(X_j)\sum_{k=1}^{m}p(Y_k|X_j)\log(p(Y_k|X_j))\end{aligned}$$

$$
\begin{aligned}
&= \sum_{k=1}^{m} [p(X_i)p(Y_k|X_i)\log(p(Y_k|X_i)) \\
&\quad + p(X_j)p(Y_k|X_j)\log(p(Y_k|X_j)) \\
&\quad - p(X_i \cup X_j)p(Y_k|X_i \cup X_j)\log(p(Y_k|X_i \cup X_j))] \\
&= \sum_{k=1}^{m} \left[-\frac{|Y_k \cap X_i|}{|U|}\log\left(\frac{|Y_k \cap X_i|}{|X_i|}\right) - \frac{|Y_k \cap X_j|}{|U|}\log\left(\frac{|Y_k \cap X_j|}{|X_j|}\right) \right. \\
&\quad \left. - \frac{|Y_k \cap X_i| + |Y_k \cap X_j|}{|U|}\log\left(\frac{|Y_k \cap X_i| + |Y_k \cap X_j|}{|X_i \cup X_j|}\right) \right] \\
&= \frac{1}{|U|}\sum_{k=1}^{m} \left[|Y_k \cap X_i|\log|Y_k \cap X_i| - |Y_k \cap X_i|\log|X_i| \right. \\
&\quad + |Y_k \cap X_j|\log|Y_k \cap X_j| - |Y_k \cap X_j|\log|X_j| \\
&\quad |Y_k \cap X_i|\log(|Y_k \cap X_i| + |Y_k \cap X_j|) \\
&\quad - |Y_k \cap X_j|\log(|Y_k \cap X_i| + |Y_k \cap X_j|) \\
&\quad \left. + |Y_k \cap X_i|\log|X_i \cup X_j| + |Y_k \cap X_j|\log|X_i \cup X_j| \right] \\
&= \frac{1}{|U|}\sum_{k=1}^{m} \{|Y_k \cap X_i|[\log|Y_k \cap X_i| + \log|X_i \cup X_j| - \log|X_i| \\
&\quad - \log(|Y_k \cap X_i| + |Y_k \cap X_j|)] + |Y_k \cap X_j|[\log|Y_k \cap X_j| \\
&\quad + \log|X_i \cup X_j| - \log|X_j| - \log(|Y_k \cap X_i| + |Y_k \cap X_j|)]\} \\
&= \frac{1}{|U|}\sum_{k=1}^{m} \left[|Y_k \cap X_i|\log\left(\frac{|Y_k \cap X_i||X_i \cup X_j|}{|X_i||Y_k \cap (X_i \cup X_j)|}\right) \right. \\
&\quad \left. + |Y_k \cap X_j|\log\left(\frac{|Y_k \cap X_j||X_i \cup X_j|}{|X_j||Y_k \cap (X_i \cup X_j)|}\right) \right]。
\end{aligned}
$$

令 $|X_i| = x$，$|X_j| = y$，$|X_i \cap Y_k| = ax$，$|X_j \cap Y_k| = by$，显然有 $x>0, y>0, 0 \leqslant a \leqslant 1, 0 \leqslant b \leqslant 1$。则

$$H_\Delta = \frac{1}{|U|}\sum_{k=1}^{m}\left(ax\log\frac{ax+ay}{ax+by} + by\log\frac{bx+by}{ax+by}\right) = \frac{1}{|U|}\sum_{k=1}^{m} f_k,$$

对于任意 $k(k=1,\cdots,m)$，有

$$f_k = ax\log\frac{ax+ay}{ax+by} + by\log\frac{bx+by}{ax+by}。$$

显然，如果有 $a=0$ 或 $b=0$，均有 $f_k>0$；当 $a=b=0$ 时，有 $f_k=0$。我们在下面的证明过程中仅考虑 $0<a\leqslant 1, 0<b\leqslant 1$ 的情况。

令 $ax=\lambda, by=\beta$，显然有 $\lambda>0, \beta>0$，则

$$f_k=\lambda\log\frac{\lambda+\frac{a}{b}\beta}{\lambda+\beta}+\beta\log\frac{\beta-\frac{a}{b}\lambda}{\lambda+\beta}。$$

令 $\theta=a/b$，显然有 $\theta>0$，则

$$f_k=\lambda\log\frac{\lambda+\theta\beta}{\lambda+\beta}+\beta\log\frac{\beta+\frac{1}{\theta}\lambda}{\lambda+\beta}。$$

所以 $$\frac{\mathrm{d}(f_k)}{\mathrm{d}(\theta)}=\lambda\frac{\beta}{\lambda+\theta\beta}-\beta\frac{1}{\beta+\frac{1}{\theta}\lambda}\frac{\lambda}{\theta^2}=\frac{\lambda\beta(\theta-1)}{\theta(\lambda+\theta\beta)},$$

故
$$\frac{\mathrm{d}(f_k)}{\mathrm{d}(\theta)}<0,\quad 0<\theta<1;$$
$$\frac{\mathrm{d}(f_k)}{\mathrm{d}(\theta)}=0,\quad \theta=1;$$
$$\frac{\mathrm{d}(f_k)}{\mathrm{d}(\theta)}>0,\quad \theta>1。$$

因此，当 $\theta=a/b=1$ 时，函数 f_k 取最小值 $f_k|_{\theta=1}=0$。

综上可得，只有在对于任意 $k(k=1,\cdots,m)$ 都有 $|X_i\cap Y_k|/|X_i|=|X_j\cap Y_k|/|X_j|$ 的情况下，$H_\Delta=0$，在其他任何情况下均有 $H_\Delta>0$。故引理 7.1 得证。

有了引理 7.1，便可对定理 7.9 作如下证明：

首先，由引理 7.1 可知：如果将决策表条件属性的分类进行合并，将导致条件熵的单调上升，只有在发生合并的两个分类对于决策类的隶属度（概率）均相等的情况下，才可能不导致条件熵的变化。

其次，划分 $U|\mathrm{IND}(A)$ 是可以通过将划分 $U|\mathrm{IND}(A\cup\{a\})$ 中的部分等价块合并得到的，根据引理 7.1 可知：如果 $H(D|A\cup\{a\})=H(D|A)$，则所有被合并在一起的等价块对于决策类的隶属度（概率）均相等。因此，在合并后，每个条件属性分类中的等价块对于各个决策属性分类的隶属度不会发生变化。因此，

$$\mathrm{POS}_{A\cup\{a\}}(F)=\mathrm{POS}_A(F)。$$

由定理 7.9 可以看出，如果一个属性不能为另一个属性集合的分类增加任何信息，我们就可以将它进行约简，采用这种方法指导对决策表的约简过程，也就是说，约简的信息熵描述包含了代数描述。基于这个定理，我们就可以用条件熵为启发知识设计相应的启发式约简算法。

7.3　决策表属性约简算法

决策表属性约简的过程，就是从决策表系统的条件属性中去掉不必要（对得到决策不重要）的条件属性，从而分析所得约简中的条件属性对于决策属性的决策规则。在不同的系统中，或者在不同的条件环境下，人们对属性约简的要求和期望是不一致的。如果在某个系统中，存在一些属性，它们的属性值难于得到（测量这些属性值所需要花费的代价很高），我们希望将这些属性从决策表中去掉。通常，人们希望得到的约简结果所包含的条件数目尽可能少，或者得到的决策规则的规则数最少，等等。

下面，我们将具体介绍几种决策表属性约简的算法。

这里，设原始决策表的条件属性集合为 $P=\{a_i \mid i=1,\cdots,n\}$，决策属性集合为 $D=\{d\}$。

7.3.1　一般约简算法

对于决策表中的每一个条件属性 a_i，进行如下过程，直至条件属性集合不再发生变化为止。

{

如果删除该属性 a_i 使得 $\mathrm{POS}_{(P-\{a_i\})}(Q)=\mathrm{POS}_P(Q)$，则说明属性 a_i 是相对于决策属性 d 不必要的，从决策表中删除属性 a_i 所在列并将重复的行进行合并；

否则，说明属性 a_i 是相对于决策属性 d 必要的，不能删除。

}

上述算法，能够得到决策表的一个属性约简结果，但不一定能够得到一个满意的属性约简结果。利用这个算法，我们也可以采用搜索策略来得到所有可能的属性约简结果。如果采用宽度优先策略，可以首先从原始决策表中删除一个属性，得到所有可能的包含 $n-1$ 个条件属性的子决策表，然后再反复地在这些子决策表上进行上述操作，最终就可以得到所有可能的属性约简结果。同样，也可以采用深度优先策略。但不幸的是，由于这是一个组合爆炸问题，穷尽的搜索所需要的时间和空间代价都很高，实际计算属性约简的时候，往往采用某种启发式的算法。

7.3.2 基于可辨识矩阵和逻辑运算的属性约简算法

我们在第 4 章中介绍了决策表的可辨识矩阵概念。下面介绍基于可辨识矩阵和逻辑运算的属性约简算法。

第 1 步 计算决策表的可辨识矩阵 C_D；

第 2 步 对于可辨识矩阵中的所有取值为非空集合的元素 $C_{ij}(C_{ij}\neq 0, C_{ij}\neq\varnothing)$，建立相应的析取逻辑表达式 L_{ij}，

$$L_{ij}=\bigvee_{a_i\in C_{ij}} a_i;$$

第 3 步 将所有的析取逻辑表达式 L_{ij} 进行合取运算，得一个合取范式 L，即

$$L=\bigwedge_{C_{ij}\neq 0,\ C_{ij}\neq\varnothing} L_{ij};$$

第 4 步 将合取范式 L 转换为析取范式的形式，得

$$L'=\bigvee_{i} L_i;$$

第 5 步 输出属性约简结果。析取范式中的每个合取项就对应一个属性约简的结果，每个合取项中所包含的属性组成约简后的条件属性集合。

下面再通过实际例子来说明这一属性约简算法。

对于如表 7.1 所示的关于气象信息的决策表系统，我们首先得到其可辨识矩阵如下：

$$
\begin{Bmatrix}
0 & 0 & a_1 & a_1a_2 & a_1a_2a_3 & 0 & a_1a_2a_3a_4 & 0 & a_2a_3 & a_1a_2d_1 & a_2a_3d_4 & a_1a_2d_1 & a_1a_2 & 0 & a_1a_2a_3a_4 \\
0 & 0 & a_1a_3 & a_1a_2a_4 & a_1a_2a_4a_4 & 0 & a_1a_2a_3 & 0 & a_2a_3d_4 & a_1a_2a_4a_4 & a_2a_4 & a_1a_2 & a_1a_4 & 0 & a_1a_2a_3 \\
 & & 0 & 0 & 0 & a_1a_2a_4a_4 & 0 & a_1a_2 & 0 & 0 & 0 & 0 & 0 & a_1a_2 & 0 \\
 & & & 0 & 0 & a_2a_3a_4a_1 & 0 & a_1a_2 & 0 & 0 & 0 & 0 & 0 & a_1a_2a_3a_4 & a_1a_2a_3a_4 \\
 & & & & 0 & a_4 & 0 & a_1a_3 & 0 & 0 & 0 & 0 & 0 & a_3a_4 & a_3a_4 \\
 & & & & & 0 & a_1 & 0 & a_1a_4 & a_2a_4 & a_1a_2 & a_1a_2a_4 & 0 & a_1a_3 & a_1a_3 \\
 & & & & & & 0 & a_1a_2a_3a_4 & 0 & a_1a_3 & a_3a_4 & a_1a_2a_4 & 0 & a_1a_4 & a_1a_4 \\
 & & & & & & & 0 & a_2a_3 & 0 & 0 & 0 & 0 & a_1a_2a_3a_4 & 0
\end{Bmatrix}
$$

进而可以得到 15 个析取逻辑表达式，如

$$L_{1,3}=a_1,$$

$$L_{2,3}=a_3 \vee a_4,$$

$$L_{1,4}=a_1 \vee a_2,$$

$$L_{2,4}=a_1 \vee a_2 \vee a_4,$$

$$\vdots$$

$$L_{1,3,4}=a_1 \vee a_2 \vee a_3 \vee a_4.$$

将这些表达式进行合取得到合取表达式 L，

$$\begin{aligned} L &= L_{1,3} \wedge L_{2,3} \wedge L_{1,4} \wedge L_{2,4} \wedge \cdots \wedge L_{1,3,4} \\ &= a_1 \wedge (a_1 \vee a_4) \wedge (a_1 \vee a_2) \wedge (a_1 \vee a_2 \vee a_4) \wedge \cdots \wedge \\ &\quad (a_1 \vee a_2 \vee a_3 \vee a_4), \end{aligned}$$

对 L 进行转换，最终得到析取范式 L'，

$$L'=(a_1 \wedge a_2 \wedge a_4) \vee (a_1 \wedge a_3 \wedge a_4),$$

这样就得到了两个属性约简结果，分别如表 7.4 和表 7.5 所示。

表 7.4 约简结果 $\{a_1, a_2, a_4\}$

U	条件属性			决策属性(d)
	Outlook(a_1)	Temperature(a_2)	Windy(a_4)	
1	Sunny	Hot	False	N
2	Sunny	Hot	True	N
3	Overcast	Hot	False	P
4	Rain	Mild	False	P
5	Rain	Cool	False	P
6	Rain	Cool	True	N
7	Overcast	Cool	True	P
8	Sunny	Mild	False	N
9	Sunny	Cool	False	P
10	Sunny	Mild	True	P
11	Overcast	Mild	True	P
12	Rain	Mild	True	N

表 7.5　约简结果：a_1,a_3,a_4

U	条件属性			决策属性(d)
	Outlook(a_1)	Humidity(a_3)	Windy(a_4)	
1	Sunny	High	False	N
2	Sunny	High	True	N
3	Overcast	High	False	P
4	Rain	High	False	P
5	Rain	Normal	False	P
6	Rain	Normal	True	N
7	Overcast	Normal	True	P
8	Sunny	Normal	False	P
9	Sunny	Normal	True	P
10	Overcast	High	True	P
11	Overcast	Normal	False	P
12	Rain	High	True	N

基于可辨识矩阵和逻辑运算的属性约简算法可以得到决策表的所有可能的属性约简结果，它实际上是将对属性组合情况的搜索演变成为逻辑公式的化简，从而简化问题。但是，我们可以看到，算法第 2 步建立的析取逻辑表达式 L_{ij} 是很多的，甚至是重复的，这将导致逻辑公式化简时的计算量增大。我们可以采取一定的措施对该算法进行改进，进一步简化属性约简过程。

考察可辨识矩阵，不难发现，如果矩阵中存在一个元素，其取值为包含单属性元素的集合，则表明该属性是区分这个矩阵元素所对应的两个样本所必须的属性，也是唯一能够区分这两个样本的属性。可辨识矩阵中的这些元素所包含的属性组成的属性集合其实就是该决策表系统的相对属性核。我们首先就可以将这些属性取出，同时将可辨识矩阵中包含核属性的元素的值修改为 0，从而得到一个新的矩阵，再在这个新矩阵的基础上来实施算法的第 2,3,4 步，得到一个析取范式逻辑表达式，最后将所有的核属性加

入析取范式中的每个合取项，就得到属性约简的结果。

在上面的例子中，$C_{1,2}|=1$，$|C_{1,6}|=1$，$C_{6,7}|=1$，$|C_{4,11}|=1$，我们可以从可辨识矩阵得到属性核 $\mathrm{CORE}_Q(P)=\{a_1,a_4\}$，修改决策矩阵中的元素值后得到的新矩阵为

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_2a_3 & 0 & 0 & 0 & 0 & 0\\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_2a_3 & 0 & 0 & 0\\
 & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
 & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
 & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
 & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
 & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
 & & & & & & & 0 & a_2a_3 & 0 & 0 & 0 & 0 & 0\\
 & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0\\
 & & & & & & & & & 0 & 0 & 0 & 0 & 0\\
 & & & & & & & & & & 0 & 0 & 0 & 0\\
 & & & & & & & & & & & 0 & 0 & 0\\
 & & & & & & & & & & & & 0 & 0\\
 & & & & & & & & & & & & & 0
\end{pmatrix}.$$

这个矩阵只包含三个取值为非空集合的元素，这样，得到的逻辑公式就很简单，最终化简得到

$$L'=a_2\vee a_3,$$

将核属性加入到各合取项中得到结果 $(a_1\wedge a_2\wedge a_4)\vee(a_1\wedge a_2\wedge a_4)$。

根据得到的所有可能的属性约简结果，我们就可以根据实际问题的要求选取满意的结果。

7.3.3 归纳属性约简算法

归纳属性约简算法不仅考虑针对所有决策类提取条件属性的重要子集，而且利用核和约简的概念对各决策类的决策规则进行冗余属性约简，从而得到整个决策表系统的最小决策算法(规则)。

为了介绍归纳属性约简算法，我们需要先引入几个概念。

定义 7.18　记 $A=\{a_1,a_2,\cdots,a_m\}$ 为有限元素集合，准则函数 $C(a)$ 定义为 A 中元素的排序函数，$\forall a\in A$，有 $C(a)>0$。按 $C(a)$ 大小对 A 中元素进行排序得序集 $OA=\{a_1',a_2',\cdots,a_m'\}$，其中 $C(a_1')\geqslant C(a_2')\geqslant\cdots\geqslant C(a_m')$，定义 OA 中元素权值为 $C(a_i')=2^{m-i}$。

定义 7.19　令 $T(OA)$ 为集合 OA 的幂子集，$T_1(OA)$ 为集合 OA 的一阶幂集，给 $T(OA)$ 中元素赋以权值，有 $\forall A'\in T_1(OA)$，$w(A')=w(a_i')$，$a_i'\in A$。按 $w(A')$ 大小对 $T_1(OA)$ 中的元素进行排序，得到一阶有序幂子集 $OT_1(OA)$。

同理，$T_i(OA)$ 为集合 OA 的 $i(1\leqslant i\leqslant m)$ 阶幂子集，给 $T_i(OA)$ 中元素赋以权值，有 $\forall A'\in T_i(OA)$，$w(A')=\sum_{j=1,\cdots,i} w(a_j)$，$a_j\in A'$。按 $w(A')$ 大小对 $T_i(OA)$ 中的元素进行排序，得到 i 阶有序幂子集 $OT_i(OA)$。

下面举例说明序集、n 阶幂集和 n 阶有序幂子集的概念。

对于有限元素集合 $A=\{a_1,a_2,\cdots,a_m\}$，如果按照某种属性重要性度量方法得到重要性由大到小的一个排列 $a_1',a_2',\cdots,a_m'$，我们就可以得到序集 $OA=\{a_1',a_2',\cdots,a_m'\}$，相应地，$C(a_i')=2^{m-i}$。其各阶幂集为：

集合 A 的一阶幂集为

$$T_1(A)=T_1(OA)=\{\{a_i'\}\mid a_i'\in A\},$$

集合 A 的一阶有序幂集为

$$OT_1(A)=OT_1(OA)=\{\{a_1'\},\{a_2'\},\cdots,\{a_m'\}\},$$

集合 A 的二阶幂集为

$$T_2(A)=T_2(OA)=\{\{a_i',a_j'\}\mid a_i'\in A\wedge a_j'\in A\wedge i\neq j\wedge a_i'\neq a_j'\},$$

集合 A 的二阶有序幂集为

$$OT_2(A)=OT_2(OA)=\{\{a_1',a_2'\},\{a_1',a_3'\},\cdots,\{a_{m-1}',a_m'\}\}。$$

归纳属性约简算法为：

第 1 步　求取 P 的 D 核 $\mathrm{CORE}_D(P)$；

第 2 步　求取 P 的 D 最小属性约简 $mred_D(P)$：

(1) 令 $X=\mathrm{CORE}_D(P)$，$L=P\backslash X=\{a_1,a_2,\cdots,a_m\}$，$T(L)$ 表示 L 的幂集，$T_i(L)$ 为 L 的 $i(1\leqslant i\leqslant m)$ 阶幂子集；

(2) 如果 $\mathrm{POS}_X(D)=\mathrm{POS}_P(D)$，则 $mred_D(P)=X$，转(10)；

(3) $i=1$，$flag=0$，Z，A，X；

(4) $Y=T_i(L)$；

(5) 任取 $y\in Y$，$A=X\cup\{y\}$，

如果 $\mathrm{POS}_A(D)=\mathrm{POS}_P(D)$，则

如果 $flag=0$，则 $Z=A$，$flag=1$；

否则，如果 $\mathrm{card}(U|Z)>\mathrm{card}(U|A)$，则 $Z=A$；

(6) $Y=Y-\{y\}$；

(7) 如果 $Y\neq\varnothing$，转(5)；

(8) 如果 $flag=1$，则 $mred_D(P)=Z$，转(10)；

(9) $i=i+1$，如果 $i\leqslant m$，则转(4)；

(10) 结束。

归纳属性约简算法首先求得决策表的属性核，试图以属性核为基础来求决策表的最小属性约简。求属性核，可以采用不同的方法。实质上，这个算法还是一种对属性组合的搜索，采用了启发式知识来减小搜索空间。因为属性核是肯定在约简结果中的，所以首先得到属性核，这样就避免了对核属性之间的组合情况的搜索。同时，由于目标是求取最小属性约简，所以从属性核出发，逐步增加一个、两个……属性，直至得到约简结果，这样就避免了对超出最小属性约简属性个数的属性组合情况的搜索。因此，归纳属性约简算法是一种带启发式知识的搜索方法，并能够保证得到最小属性约简。与之相应的归纳值约简算法将在下一章中介绍。

7.3.4　基于互信息的属性约简算法——MIBARK 算法

在求取决策表属性约简的时候，可以利用决策表条件属性和决策属性之间的互信息。在决策表中增加某个属性所引起的互信息的变化的大小可以作为该属性重要性的度量。

MIBARK 算法(Mutual information – based algorithm for reduction of knowledge)：

输入：一个决策表 $T=(U,R,V,f)$，其中，U 是论域，$R=C\cup D$，C 是条件属性集合，$D=\{d\}$ 是决策属性集合。

输出：T 的一个相对约简。

步骤 1　计算决策表 T 中条件属性 C 和决策属性 D 的互信息 $I(C,D)=H(D)-H(D|C)$；

步骤 2　计算 C 相对于 D 的核 $\mathrm{CORE}_D(C)$；

步骤 3　令 $B=\mathrm{CORE}_D(C)$，对条件属性集 $C\backslash B$ 重复：

(1) 对 $C\backslash B$ 中的每个属性 p，计算条件互信息 $I(p,D|B)$；

(2) 选择使条件互信息 $I(p,D|B)$ 最大的属性，记为 p(若同时存在多个属性达到最大值，则从中选取一个与 B 的属性值组合数最少的属性作为 p)；并且 $B=B\cup\{p\}$；

(3) 若 $I(B,D)=I(C,D)$，则终止；否则，转(1)；

步骤 4　最后得到的 B 就是 C 相对于 D 的一个相对约简。

MIBARK 算法也是一种启发式算法，在多数情况下能够得到决策表的最小属性约简。但是，并不一定能够保证算法得到决策表的最小属性约简。

7.3.5　基于特征选择的属性约简算法

X Hu 在研究决策表约简问题的时候，对决策表中的属性进行了度量，并提出了特征(属性)选择算法，实际上也是属性约简算法。为了介绍这种方法，我们先对相应的特征度量问题加以讨论。

我们在第 4 章中对属性重要性进行了讨论，但是这些度量方法仅说明了单个特征对分类能力的影响。通常情况下，一个特征本身不足以分类，还需要综合考虑其他特征。X Hu 提出了一种用于对特征进行排列的特征价值度量方法。这种特征价值度量方法包含两部分：加权特征差(WFD，Weighted Feature Difference)和值差(VD，Value Difference)。

计算两个样本之间特征差的通常办法是计算它们之间的 Euclidean 距离。假定 D_1 类中的一个样本和 D_2 类中与其相应的样本的差异有三个特征，即 C_1,C_2 和 C_3。两个样本之间的距离为 3。这三个特征是区分这两个样本类的唯一依据。由此可以定义两个样本 t_i,t_j 之间的加权特征差：

$$WFD_{ij}=1/D_{ij}^2,$$

其中：$D_{ij}=\sum_{k=1}^{m} d_{ij}^{(k)}$，$d_{ij}^{(k)}=\begin{cases}1, & C_{ik}\neq C_{jk};\\ 0, & \text{否则}。\end{cases}$ m 是特征(属性)数，C_{ik}和 C_{jk}分别是 t_i,t_j 的第 k 个属性的取值。

Kullback 对属性值的信息容量给出了如下度量：

$$K(D|C_k=c)=\sum_{d\in D} P(d|c)\log\left(\frac{P(d|c)}{P(d)}\right),$$

其中 c,d 分别表示特征 C_k 和决策类 D 的取值。

值差(VD)可以由此而定义：

$$VD(C_{ki},C_{kj})^{(k)}=\frac{|K(D|C_k=C_{ki})-K(D|C_k=C_{kj})|}{K_{\max}-K_{\min}},$$

$K_{\max}$,$K_{\min}$分别是属性 C_k 最大和最小的 K 值。

X Hu 基于加权特征差和值差给出了上、下文敏感的价值度量(CM)。特征 C_k 的上、下文价值 CM_k 定义为：

$$CM_k=\sum_{t_i\in D_i}\sum_{t_j\in\neg D_i} WFD_{ij}\times VD(c_{ki},c_{kj})^{(k)},$$

其中，如果 $t_j\in\neg D_i$ 是 $t_i\in D_i$ 的 $K(K=\log_2 N$，N 为 $\neg D_i$ 中的样本个数)近邻中的样本，则 $WFD_{ij}=1/D_{ij}^2$；否则，$WFD_{ij}=0$。

X Hu 利用可变精度 Rough 集模型，计算所有特征的特征价值，并对特征进行排序，删除特征价值最小的特征，反复上述过程，直至特征集合与决策类的依赖关系达到期望的程度。期望的特征数可能大于约简所需要的特征数，这可以根据具体问题来设定，特别是在噪音量大的情况下，期望的特征数应该多一些，这有利于提高所得规则集的正确度。

Rough 集特征选择算法：

第 1 步　计算条件属性集 C 和决策属性 D 之间的相关程度 $K_\beta(C,D)$；

第 2 步　$RED=C$；

第 3 步　While $K_\beta(C,D)=K_\beta(RED,D)$ Do

(1) 计算 RED 中所有属性的上、下文价值 CM；

(2) 根据上、下文价值对 RED 中的属性进行排序；

(3) 选择属性 a_i,a_j 具有最小的上、下文价值，且

$K_\beta(RED,D)=K_\beta(RED\setminus\{a_i\},D)$；

(4) $RED\leftarrow RED\setminus\{a_i\}$；

(5) Endwhile；

第 4 步　输出 RED。

这个特征选择算法能够得到一个和决策属性具有很强依赖关系的条件属性子集。当 $\beta=0$ 时，得到条件属性集合的一个相对约简。算法的复杂度为 $O(N^2)$，N 为决策表中的样本数。

7.4　不完备信息系统的属性约简

对于不完备信息系统，我们可以首先采用第 6 章中所介绍的补齐算法先进行完备化处理，然后再对所得到的完备信息系统采用 7.3 节所介绍的约简算法进行约简处理。但是，补齐处理只是将未知值补以我们的主观估计值，不一定完全符合客观事实，在对不完备信息系统进行补齐处理的同时，我们或多或少地改变了原始的信息系统。因此，在有的情况下，我们还是希望在保持信息系统的原始信息不发生变化的前提下对信息系统进行处理，这就需要借助于我们在 3.6 节中介绍的关于不完备信息系统中的扩充 Rough 集理论。

7.4.1　容差关系

根据容差关系，不完备信息表的属性约简的定义与传统 Rough

集的属性约简定义相似，是保持对象分类对于所有属性的下近似不发生变化的最小的属性子集。对于表 3.2 所示的不完备信息表，存在唯一属性约简 $\{c_1, c_2, c_4\}$。若决策规则采用 $(\wedge_i (c_i, v) \rightarrow V(d, w)$ 的形式，且决策部分只有一个属性，则规则是确定规则。若 B 是出现在规则 $s \rightarrow t$ 的条件部分的条件属性集合，只有对满足条件部分 s 的每个对象 x 都有 $I_B(x) \subseteq [t]$ 时，决策规则为真。决策规则的条件部分也要求是没有冗余的。对于表 3.2 所示的不完备信息表，可以得到唯一的确定决策规则：$(c_1 = 2) \wedge (c_2 = 3) \wedge (c_4 = 1) \rightarrow (d = \Psi)$。

7.4.2 非对称相似关系

根据非对称相似关系，称属性集 $C' \subseteq C$ 是属性集 C 对于某个分类的约简的条件是 C' 是保持这个分类的下近似不发生变化的属性集 C 的最小子集。根据非对称相似关系的定义，一个所有属性值均未知的完全未知对象不相似于其它任何对象。如果我们删除一个(或一些)属性导致一个对象在所有剩余的属性上的值均为未知值，就丢失了分类的相对信息。所以，那样的属性是必须要保留在约简中的。对于表 3.2 所示的不完备信息表，存在一个属性约简 $\{c_1, c_2, c_4\}$。对于任何对象 x，由这个约简得到的 $R^{-1}(x)$ 和 $R(x)$ 与由全部属性得到的相同。

决策规则的形式可以采用 $s \rightarrow t$，其中 $s = \wedge_i (c_i, v), t = (d, \omega)$。对于每个满足 s 的对象，都有 $R(x) \subseteq [t]$，则规则为真。规则的条件部分不能有冗余属性。

对于表 3.2 所示的不完备信息表，可以得到下列确定决策规则：

$$(c_1 = 1) \rightarrow (d = \Phi),$$
$$(c_3 = 1) \wedge (c_4 = 0) \rightarrow (d = \Phi),$$
$$(c_1 = 3) \wedge (c_4 = 0) \rightarrow (d = \Phi),$$
$$(c_2 = 3) \wedge (c_4 = 1) \rightarrow (d = \Psi),$$
$$(c_2 = 0) \rightarrow (d = \Psi),$$
$$(c_3 = 0) \rightarrow (d = \Psi)。$$

$=0.905$。然而，规则 ρ_1 的条件部分是冗余的，可以转换为 ρ_1'：$(c_1=3)\wedge(c_4=0)\rightarrow(d=\Phi)$，$\mu(\rho_1')=0.905$。支持这条规则的对象集合是 $\{a_1, a_{11}, a_{12}\}$。对于类 Ψ，我们也有一条规则 ρ_2：$(c_1=2)\wedge(c_2=3)\wedge(c_4=1)\rightarrow(d=\Phi)$，$\mu(\rho_2)=1.0$。支持这条规则的对象集合是 $\{a_8\}$。

我们还可以先为决策规则选定一个可信度阈值，然后对规则集进行分析，得到满足要求的决策规则。

第8章　决策表值约简

值约简是在属性约简的基础上对决策表的进一步简化。本章将就决策表的值约简问题进行系统分析，并介绍几种主要的值约简算法。

8.1　决策表值约简概述

在第7章中，我们介绍了决策信息表的属性约简，通过属性约简，可以将决策表中对决策分类不必要的属性省略，从而实现决策表的简化，这有利于从决策表中分析发现对决策分类起作用的属性。但是，属性约简只是在一定程度上去掉了决策表中的冗余属性，但是还没有充分去掉决策表中的冗余信息。例如，在表7.4所示的关于气象信息的决策表的属性约简结果中，如果在条件 Outlook = Sunny ∧ Temperature = Hot 下，决策属性的取值肯定是N，而无需考虑条件属性 Windy 的取值是 True 还是 False。显然，这个属性约简结果，对于决策分类来说，仍然包含冗余信息。根据第4章中介绍的决策规则，我们不能够直接从该表中得到满意的决策规则。这就是说我们还需要进一步对决策表进行处理，得到更加简化的决策表，这就是我们本章将要讨论的决策表值约简问题。

与属性约简中的属性核一样，值约简中也可以定义相应的值核。

决策表 $S=\langle U,C,D,V,f\rangle$，对于任意的 $x\in U$，用 d_x 表示决策规则，即

$$\mathrm{d}_x:\mathrm{des}([x]_C)\Rightarrow\mathrm{des}([x]_D),\ \mathrm{d}_x(a)=a(x),\ a\in C\cup D,$$

$(\mathrm{d}_1, C, \mathrm{d}_2, D)$ 分别称为 d_2 的条件和决策。

定义 8.1 考虑一个相容知识表达系统 S，对决策规则 d_2 有 $[x]_C \subseteq [x]_D$，若 $\forall r \in C$，有 $[x]_{C-\{r\}} \not\subseteq [x]_D$，则 r 为 d_2 的核值属性，r 为 d_2 中不可省略的；若 $[x]_{C-\{r\}} \subseteq [x]_D$，则 r 不是 d_2 的核值属性，r 为 d_2 中可省略的。

8.2 决策表值约简算法

8.2.1 一般值约简算法

对于一个经过属性约简而得到的决策表，我们可以对应其中的每一个样本形成一条决策规则。因此，我们可以将决策表中的样本用规则来表示，这样，约简后的决策表实际上就是一个规则集合。对于这个规则集合，我们可以利用如下算法来进行简化：

对于规则集合中的每条规则

对于该规则中的任意条件属性

如果去掉该条件属性，该规则不和规则集中的其他规则冲突，则可以从该规则中去掉该条件属性。

经过这样处理得到的规则集合中的所有规则都不含有冗余条件属性，也就是说，规则的条件属性数目已经被尽可能减少了。但是，这个算法的实现有很多任意性，比如，由于处理规则的顺序不同，或者处理规则中条件属性的顺序不同，我们都可以得到不同的值约简结果，得到的规则集合就会有所不同。因此，我们往往需要一些启发式知识来指导这一过程的进行。

8.2.2 归纳值约简算法

我们在 7.3.3 节中对归纳属性约简进行了介绍，这里对归纳值约简加以讨论。

由核值的定义，求得每个规则 d_i 的核值属性，就可形成决策表的条件属性核值表。但是，这样做的工作量太大。为了介绍归纳

值约简算法，先看如下命题：

命题 8.1　对相容知识表达系统 $S=(U,C,D,V,f)$，则以属性 a 为核值属性的决策规则集合为

$$\mathrm{core}(a)=\{\mathrm{d}_i: x\in(U\setminus \mathrm{POS}_{C\setminus\{a\}}(D))\}。$$

证明　$\forall a\in C$，令 $B=\mathrm{POS}_{C\setminus\{a\}}(D)$，$\forall x\in U\setminus B$，如果规则 $\mathrm{d}_i:\mathrm{des}([x]_{C\setminus\{a\}})\Rightarrow\mathrm{des}([x]_D)$ 为不相容决策规则，则必存在一决策规则 d_j，使得 $\mathrm{d}_j(C\setminus\{a\})=\mathrm{d}_i(C\setminus\{a\})$，而 $\mathrm{d}_j|D\neq\mathrm{d}_i|D$，即 $x'\in[x]_{C\setminus\{a\}}$，但 $x'\notin[x]_D$，因此 $[x]_{C\setminus\{a\}}\subset[x]_D$。所以 a 为决策规则 d_i 的核值属性，即 $\mathrm{core}(a)=\{\mathrm{d}_i\mid x\in(U\setminus\mathrm{POS}_{C\setminus\{a\}}(D))\}$。

根据上述命题，可以方便地求取任意条件属性 a 的 $\mathrm{core}(a)$，从而得到决策表的条件属性核值表。在此基础上，我们来计算决策规则属性值的简化。

令 $U/D=\{y_1,y_2,\cdots,y_n\}$ 表示论域 U 上由决策属性划分的决策类集，对每一个决策等价类，定义决策规则类 DRC 为

$$\mathrm{DRC}(y)=\{\mathrm{d}_x:\mathrm{des}([x]_C)\Rightarrow\mathrm{des}([x]_D)\mid x\in U\ \text{且}\ [x]_C\subseteq y\},$$
$$\forall y\in U/D,$$

求解知识表达系统决策表的最小决策算法，可通过分别求解各个决策类的最小决策算法来实现。各决策类的最小决策算法则通过删除决策规则类中决策规则的冗余属性值及冗余规则来实现。

用 $\mathrm{core}(y),\forall y\in U/D$ 表示决策类 y 的核值属性集，$\mathrm{core}(\mathrm{d}_i)$ 表示决策规则 d_i 的核值属性集，则有

$$\mathrm{core}(y)\subseteq C,\ \mathrm{core}(\mathrm{d}_x)\subseteq C,\ \text{且}$$
$$\mathrm{core}(y)=\bigcup_{\mathrm{d}_i\in\mathrm{DRC}(y)}\mathrm{core}(\mathrm{d}_i)。$$

下面给出求取决策类 y 的最小决策算法步骤：

(1) 任取 $\mathrm{d}_i\in\mathrm{DRC}(y)$；

(2) 如果 $[x]_{\mathrm{core}(\mathrm{d}_i)}\subseteq y$，则输出决策规则 $\mathrm{d}_i:\mathrm{des}[x]_{\mathrm{core}(\mathrm{d}_i)}\Rightarrow\mathrm{des}([x]_D)$，$\mathrm{DRC}(y)=\mathrm{DRC}(y)/[x]_{\mathrm{core}(\mathrm{d}_i)}$，转(9)；

其中：$DRC(y)=DRC(y)/[x]_{core(d_r)}$ 表示从 $DRC(y)$ 中删除规则 $d_r:des([x']_C)\Rightarrow des([x']_D)$，这里，$x'\in[x]_{core(d_r)}$。

(3) 令 $A_1=core(y)\backslash core(d_r)$，$A_2=C\backslash core(y)$，在测度函数 $w(a)=|POS_{C-\{a\}}(D)|/|U|$ 下对 A_1，A_2 中元素排序，得有序集 OA_1，OA_2，则有序集 $OA=OA_1\bigcup OA_2$ 且 $|OA|=m$，OA 的 m 个有序幂子集分别为 $T_1(OA)$，$T_2(OA)$，…，$T_m(OA)$，相应的元素个数为 n_1，n_2，…，n_m。

(4) $j=1$

(5) $i=1$。

(6) 令 $B=core(d_r)\bigcup T_j'(OA)$，如果 $[x]_B\subseteq y$，输出 $d_r:des([x]_B)\Rightarrow des([x]_D)$，$DRC(y)=DRC(y)/[x]_B$，转(9)。

(7) $i=i+1$，如果 $i\leqslant n_j$，转(6)。

(8) $j=j+1$，如果 $j\leqslant m$，转(5)。

(9) 如果 $DRC(y)\neq\varnothing$，转(1)。

(10) 结束。

根据上述步骤，依次求得各决策类 $y\in U/D$ 的最小决策算法，就可以得到整个决策表的最小决策算法。

8.2.3 启发式值约简算法

分析最小值约简，也可以从值核入手。

算法输入：信息系统 T(假定系统有 n 条记录，$m-1$ 个条件属性，1 个决策属性)。

算法输出：T 的值约简 T'。

第 1 步 对信息表中条件属性进行逐列考察。若删除该列后产生冲突记录，则保留冲突记录的原该属性值；否则，如果有重复记录，则将重复记录的该属性值标记为“*”；对于其他记录，将该属性值标记为“?”。

For($j=1$ To $m-1$)

For($i=1$ To n)

If $\exists k(k\neq i\wedge\forall l((l\neq j\wedge l\neq m\wedge T'_{il}\neq *\wedge T''_{il}\neq ?)\rightarrow T'_{il}=T_{kl})\wedge T_{im}\neq T_{km}$

$T''_{ij}=T_{ij}$;

Else if $\exists k(k\neq i\wedge\forall l(l\neq j\wedge T_{il}\neq *\wedge T''_{il}\neq ?)\rightarrow T_{il}=T_{kl})$

$T''_{ij}=*$;

Else $T''_{ij}=?$;

}

For($i=1$ To n) $T''_{im}=T_{im}$;

第 2 步　删除可能产生的重复记录，并考察每条含有标记“?”的记录。若仅由未被标记的属性值即可以判断出决策，则将标记“?”改为“*”；否则，将标记“?”修改为原属性值；若某条记录的所有条件属性均被标记，则标记“?”修改为原属性值。

For($j=1$ To $m-1$)

For($i=1$ To n) {

If $T''_{ij}=?$ {

If $\forall l(l\neq m\rightarrow(T''_{il}=?\ \vee T''_{il}=*))$

$T'_{ij}=T_{ij}$;

Else If $\forall k(\forall l(l\neq m\wedge T''_{il}\neq ?\ \wedge T''_{il}\neq *\rightarrow T_{il}=T_{kl})\rightarrow T_{im}=T_{km})$

$T'_{ij}=*$;

Else $T''_{ij}=T_{ij}$;

}

}

第 3 步　删除所有条件属性均被标记为“*”的记录及可能产生的重复记录(假定 card(T')=n')。

第 4 步　如果两条记录仅有一个条件属性值不同，且其中一条记录该属性被标记为“*”，那么，对该记录如果可由未被标记的属性值判断出决策，则删除另外一条记录；否则，删除本记录。

For each tuple (i) in T'

If $\exists k\exists l(l\neq m\wedge T'_{il}\neq T''_{kl}\wedge T'_{il}=*\wedge\forall j(j\neq l\rightarrow T'_{ij}=T''_{kj}))$ {

If $\forall h(\forall j((j\neq m\wedge T''_{ij}\neq *)\rightarrow T_{hj}=T''_{ij})\rightarrow T_{hm}=T''_{im})$

删除记录 k;

Else 删除记录 i;

}

Else If $\exists k \exists l (l \neq m \wedge T'_{il} \neq T'_{ml} \wedge T'_{kl} \neq * \wedge \forall j (j \neq l \rightarrow T'_{ij} = T'_{kj}))$ {

If $\forall h (\forall j ((j \neq m \wedge T'_{hj} \neq *) \rightarrow T_{hj} = T_{ij}) \rightarrow T_{hm} \neq T_{km})$

删除记录 i;

Else 删除记录 i;

}

经过上述值约简之后得到的新信息表，所有属性值均为该表的值核，所有记录均对应为一条决策规则。

8.2.4 基于决策矩阵的值约简算法

这里对日阿可(Ziarko)等人用于获取具有最大适应度(一般化)规则的值约简算法进行介绍，采用的是可变精度 Rough 集模型。

对于一个属性约简结果信息表 RED，令 X_i $(i=1,2,\cdots,\gamma)$，X_j $(j=1,\cdots,\rho)$ 表示关系 $R^*(RED)$ 的等价类，$X_j^* \subseteq \mathrm{POS}^{\beta}_{RED}(Y)$，$X_j^* \subseteq \mathrm{NEG}^{\beta}_{RED}(Y)$，决策矩阵 $\boldsymbol{M}=(M_{ij})_{\gamma\rho}$ 定义为

$$M_{ij}=\{(a, f(X_i^*, a)): a \in RED,\ f(X_i^*, a) \neq f(X_j^*, a)\},$$

也就是说，M_{ij} 包含了在等价类 X_i^* 和 X_j 上具有不同值的所有属性值对。给定等价类 X_i，将 M_{ij} 的各个元素作为一个布尔表达式，决策规则集合可以表达为如下形式的布尔函数：

$$B_i = \bigwedge_j (\bigvee M_{ij})。$$

可以看出，布尔函数 B_i 的基本蕴涵实际上是属于正域 $\mathrm{POS}^{\beta}_{RED}(Y)$ 的等价类 X_i 的最大一般化规则。因此，通过发现所有决策函数 B_i $(i=1,2,\cdots,\gamma)$ 的基本蕴涵，就可以计算出正域 $\mathrm{POS}^{\beta}_{RED}(Y)$ 的所有最大一般化规则。

Ziarko 等人将此算法成功地应用于一个水资源调度系统的设计中，有关内容可以参考本书 10.2 节。

8.3 缺省规则获取算法

前面对属性约简和值约简的算法进行了介绍，经过约简，得到

的结果就直接和决策规则对应，因此也就是得到了决策规则。对于决策表，我们也不一定需要通过约简来学习得到决策规则。下面介绍斯科龙(Skowron)提出的一种通过投影得到缺省决策规则的算法。

针对包含不一致决策情况的决策表，Skowron 提出了相应的缺省规则获取方法

Skowron 的缺省规则获取方法：

输入：决策表 $\mathbf{A}^*=(U^*,A^*)$，其中 $A^*=\{C^*,D\}$，U^* 是决策表中个体(或称为元素、样本)的全集，A^* 是每个个体的属性集，包括条件属性集 C^* 和决策属性 D；

输出：缺省规则集。

第 1 步　根据条件属性计算 $\mathbf{A}^*$ 的不分明关系，即条件属性对决策表 $\mathbf{A}^*$ 的划分：$E_{(K,C^*)}$，($E_{(K,C^*)}$ 属于 $U^*/\mathrm{IND}(C^*)$，$K=1,\cdots,|U^*/\mathrm{IND}(C^*)|$)。如果某个划分 $E_{(K,C^*)}$ 对特定决策(如 X_j)的成员度超过一定阈值，则根据决策表 $\mathbf{A}^*$ 的可辨识矩阵产生相应的缺省规则，即如果

$$\mu_{C^*}(E_{(K,C^*)},X_j)=|E_{(K,C^*)}\cap X_j|/|E_{(K,C^*)}|\geqslant\mu_{pr},$$

则得到规则

$$R:Des(E_{(K,C^*)},C^*)\rightarrow Des(X_j,D)\ \Big|\ |E_{(K,C^*)}\cap X_j|/|E_{(K,C^*)}|,$$

其中：$|E_{(K,C^*)}\cap X_j|/|E_{(K,C^*)}|$ 是规则 $Des(E_{(K,C^*)},C^*)\rightarrow Des(X_j,D)$ 的可信度因子。

第 2 步　将决策表 $\mathbf{A}^*$ 加入决策表集合 Ψ，即 $\Psi=\{\mathbf{A}^*\}$。

第 3 步　如果 $\Psi=\varnothing$，则结束；否则，从 Ψ 中取出一个决策表 $A=(U',A)$，计算其属性核 $\mathrm{CORE}_D(C)$。通过删除某一核属性(如 C_{Cut})可以得到条件属性上的投影 $C_{Pr}=C\backslash C_{Cut}$，其中 $r=1,\cdots,\mathrm{card}(\mathrm{CORE}_D(C))$，$C$ 为该决策表的条件属性集合，C_{Cut} 是删掉的核条件属性。对每个投影 C_{Pr} 作如下处理：

(1) 如果 $C_{Pr}=\varnothing$，则不对该投影做任何操作；否则，做下面 4

步操作。

(2) 将投影得到的新决策表 $\mathbf{A}'=(U,A')$ 加入 $\Psi(\Psi=\Psi\cup\{\mathbf{A}'\})$，其中 $A'=(C_{Pr},D)$。

(3) 根据条件属性计算投影 C_{Pr} 的不分明关系，即条件属性对该投影决策表 $\mathbf{A}'$ 的划分 $E_{(K,C_{Pr})}(E_{(K,C_{Pr})}$ 属于 $U|\mathrm{IND}(C_{Pr}),K=1,\cdots,|U|\mathrm{IND}(C_{Pr})|)$。

(4) 如果某个划分 $E_{(K,C_{Pr})}$ 对特定决策(如 X_j)的成员度超过一定阈值，则根据决策表 $\mathbf{A}'$ 的可辨识矩阵产生相应的缺省规则，即如果

$$\mu_{C_{Pr}}(E_{(K,C_{Pr})},X_j)=|E_{(K,C_{Pr})}\cap X_j|\Big/|E_{(K,C_{Pr})}|\geqslant\mu_{Pr},$$

则得到规则

$$R':Des(E_{(K,C_{Pr})},C_{Pr})\to Des(X_j,D)\Big||E_{(K,C_{Pr})}\cap X_j|/|E_{(K,C_{Pr})}|。$$

(5) 为每条缺省规则 R' 构造封锁该规则的事实：

若存在 E_i,E_i 属于 $U|\mathrm{IND}(C)$，并且 E_i 是 $E(k,c_{pr})$ 的子集、$E_i\cap X_j=\varnothing$，则形成如下事实：

$$F':Des(E_i,C_{Out})\to NOT(R')。$$

第 4 步　转第 3 步。

下面举例说明该算法。

表 8.1 所示的决策表，其条件属性为 $C=\{a,b,c\}$，决策属性为 d，共有 100 个元素，分为 4 类，所有元素被条件属性划分为 5 个不分明关系。表 8.1 的可辨识矩阵如表 8.2 所示。其核属性为 a 和 c，如果所有的阈值取为 0.55，可以得到如下规则：

R_1: $a_1c_3\to d_1|\ 1.0$，　　R_2: $a_1c_1\to d_2|\ 1.0$，

R_3: $b_2c_1\to d_2|\ 1.0$，　　R_4: $a_2\to d_2|\ 1.0$，

R_5: $b_3\to d_2|\ 1.0$，　　R_6: $a_3\to d_3|\ 0.8$，

R_7: $b_5\to d_3|\ 0.8$。

这里，规则 $a_ib_jc_k\to d_p|\mu$ 的含义是：如果条件属性 a,b,c 的值分别为 i,j,k，则其结论 d 为 p，规则的可信度为 μ。

表 8.1　决策表例

U	a	b	c	d
E_1	1	2	3	1(50x)
E_2	1	2	1	2(5x)
E_3	2	2	3	2(30x)
E_4	2	3	3	2(10x)
$E_{5,1}$	3	5	1	3(4x)
$E_{5,2}$	3	5	1	4(1x)

表 8.2　表 8.1 所示决策表的可辨识矩阵

	E_1	E_2	E_3	E_4	E_5	
E_1		c	a	ab	abc	ac
E_2	c				ab	$c(a$ OR $b)$
E_3	a				abc	a
E_4	ab				abc	a OR b
E_5	abc	ab	abc	abc		a OR b

根据该算法，从其投影 $C\backslash\{a\}$，$C\backslash\{c\}$，$C\backslash\{a,b\}$，$C\backslash\{a,c\}$ 可分别得到如下规则和事实：

$$R_8(C\backslash\{a\}):\ b_2c_3\rightarrow d_1\,|0.62,$$

$$R_9(C\backslash\{c\}):\ a_1\rightarrow d_1\,|0.91,$$

$$R_{10}(C\backslash\{a,b\}):\ c_3\rightarrow d_1\,|0.56,$$

$$R_{11}(C\backslash\{a,c\}):\ b_2\rightarrow d_1\,|0.59,$$

$$F_1(C\backslash\{a\}):\ a_2\rightarrow \mathrm{NOT}(R_8),$$

$$F_2(C\backslash\{c\}):\ c_1\rightarrow \mathrm{NOT}(R_9),$$

$$F_3(C\backslash\{a,b\}):\ b_3\rightarrow \mathrm{NOT}(R_{10}),$$

$$F_4(C\backslash\{a,c\}):\ a_2\rightarrow \mathrm{NOT}(R_{11}),$$

$$c_1\rightarrow \mathrm{NOT}(R_{11})。$$

Skowron 解决了在决策表中有冲突和不一致情况下的规则获取问题，而且，即使对于一致的情况，为了能够得到适应度更大

的缺省规则，他也通过删掉决策表中的核属性来引入人为的不一致性，得到适应度更大的、具有不确定性的缺省规则，使得所得到的规则对待识样本具有更好的适应性。如果我们只是从决策表中获取确定规则（即可信度为 1.0 的规则），那么对一些待识样本就无法处理。假设有一个待识样本，我们只知道其属性 a 的取值为 1，我们就无法根据确定规则推断其结论是什么；但如果我们采用缺省规则，就能够在一定程度上判定这个样本可能是第 1 类（d_1）样本，可信度为 0.91。但是，Skowron 的这一方法并不完备。如果待识样本为 $a_1b_3c_3$，根据规则 R_1 可以得到结论 d_1，根据规则 R_5 可以得到结论 d_2，这两条规则的可信度均为 1.0，我们仍然无法判定该样本的类别。同样，如果是样本 $a_1b_5c_2$，根据规则 R_9 可以判定结论为 d_1，其可信度为 0.91；而根据规则 R_7 可以判定结论为 d_3，其可信度为 0.8。我们又如何判定该样本的类别呢？显然，出现这些问题的原因在于规则之间有冲突（矛盾）。

对于不一致性问题，我们将在下一章中进行讨论。

第9章　逻辑推理系统

逻辑推理系统是知识获取最终应用的体现，即将在知识获取过程中得到的知识应用于实际处理之中，根据对现实的观察，利用知识进行推理，得到所需要的结论，如判定结果、控制策略等。本章将讨论基于规则知识集的逻辑推理方法，以及相应的推理控制策略，并针对知识系统不一致性问题对不一致推理问题进行系统研究。

9.1　逻辑推理方法

虽然知识是人类（或系统）求解问题的基础，知识的多寡决定了一个人（或系统）处理问题水平的高低，但使用知识进行推理的能力对于一个人（或系统）来说同样是不可缺少的。人类的推理有演绎推理、类比推理、归纳推理等多种形式。在这里，我们主要讨论演绎推理。演绎推理是用判断性知识由已知信息得出新信息的推理。调度、使用知识的方法称为控制策略。控制策略的好坏决定着系统求解问题的速度和质量。一个控制策略由两部分组成：推理方法——按什么方式推理及如何评价结论的可靠性；搜索策略——如何构造一条花费较少的推理路线。

按推理所得结论的可靠性不同，可以将推理分为精确推理和非精确推理。在大多数实际应用系统中，由于现实世界的不确定性，需要在逻辑推理的过程中同时处理不确定性问题。

按照推理过程进行的方式，可以将推理方法分为正向推理、反向推理和混合推理。下面分别进行介绍。

9.1.1　正向推理

正向推理是一种数据驱动的推理方式，其基本思想是：从基本

事实出发，引用规则库中的规则，若某些规则的前提被满足，则执行这些规则的结论部分（或者得到规则结论部分的结论）；若这些规则的结论部分形成新的事实，则再用同样的方法，以这些新事实和原有的事实为基础进行正向推理。

从上面的论述可以看到，正向推理是一个递归过程，在程序实现时要用递归程序设计思想。另外，如果规则库中的规则出现了循环推理链，则推理过程难于结束。设有下列三条规则：$A\rightarrow B$，$B\rightarrow C$，$C\rightarrow A$ 及一个基本事实 A。根据正向推理的思想，由第一条规则可形成新的事实 B，再由 B 及第二条规则可形成事实 C，再由 C 及第三条规则可形成事实 A……如果每产生一个事实（不管它是否已经存在），都要搜索规则库中的规则，并激活条件成立的规则，那么很有可能会使推理永无止境。如上面三条规则在生成逻辑结果 A 后又要进行相同路线的推理，这在大多数应用中是不合理的。为此，既要解决推理中出现的循环推理链问题，又要使推理过程正确，就必须提供一种推理控制机制，以控制推理进程的扩展和结束。但由于应用领域中的要求不同，提供一种通用的控制机制似乎不大可能。下面是几种有用的控制策略：

1. 规定每一条规则只能成功地引用一次。

当一条规则的前提全为真时，说明该规则的结论部分可被执行（即成功地引用），此时立即把该规则从规则库中删去。这样，无论产生的事实性质怎样，都不会形成循环推理链。

2. 始终利用正向推理的方法，直到没有新事实产生为止。

一条规则的前提全为真时，就执行了该规则的结论部分。若结论中产生了事实，先判断它（们）是否已经存在于事实库中，若不是，则可在这些新事实的基础上继续推理；否则不能在这些事实驱动下进行推理。在这种方法中，用事实驱动推理时，规则库的每一条规则都被检查过，且被成功引用的规则都不从规则库中删除。其实，这种方法并不能保证不形成循环推理链，只是在较大程度上可使推理过程结束。

3. 限制推理扩展深度。

形成循环推理链的根本原因是允许推理过程的无限扩展，所以若能在推理过程中限制扩展深度，一旦推理深度超过了限制，立即断开推理链，并返回到上一层推理，就避免了循环推理链。如上面由三条规则形成的推理中，若规定推理深度为 2，则当引用了第二条规则并得到了事实 C 后，就不会再进行更深入的推理。

4. 循环标志法。

这种方法的核心是对每一次被驱动的事实，都记录其推理入口点。在同一个推理链中，若出现相同的推理入口点，则说明将要形成循环推理链，返回到上层推理中。这种方法与第三种方法有些类似，但它对推理深度没有限制，只是动态地检查是否形成循环推理链。如有如下规则库：

(1) $A \rightarrow B$，　(2) $B \rightarrow C$，　(3) $C \rightarrow A$，　(4) $C \rightarrow D$

和已知基本事实 A，E。方法 4 进行的一个推理链形成过程如表 9.1所示。

表 9.1　推理链

推理链	驱动事实	被引用规则	推理入口点	事实库
开始	A	$A \rightarrow B$	(1)	A,B,E
	B	$B \rightarrow C$	(2)	A,B,C,E
	C	$C \rightarrow A$	(3)	A,B,C,E
	A	×	×	A,B,C,E
	C	$C \rightarrow D$	(4)	A,B,C,D,E
终止	D	无	返回	A,B,C,D,E

当第二次用 A 作为驱动的事实时，其推理入口点在本次推理链中已经出现过，所以不再进入推理入口点(1)，而是返回到上层推理(即以 C 为驱动事实的推理)。

以上四种方法有各自的特点，读者可根据不同的需要作相应的选择或定义新的控制策略。这四种方法都假定：对驱动事实的选

取采用“深度优先”的策略，即新产生的事实先进行数据驱动（实际上是堆栈方法）。对驱动事实的选取当然还可采用“宽度优先”的策略，即：先产生的事实先被驱动（实际上是一种队列方法）。

9.1.2 逆向推理

逆向推理是一种目标驱动的推理方法，其基本思想是：从要求解的目标出发，寻找可得出有关该目标结论的规则，判断规则中的前提是否满足，若满足则该目标得到了证明。显然，没有一条这样的规则，该目标就得不到证明。在引用规则时，若规则前提中出现了没有得到证明的新的目标，那么先要用相同的方法验证该子目标是否成立。所以，逆向推理总是按如下的方式展开推理进程：

目标→建立子目标 ……→解决了目标→解决目标。

在推理过程中，最终的子目标实际上是一些基本事实，对这些基本事实的证明并不需要用规则，它们可以是事先设定的，也可以是在推理过程中由用户实时地输入的，所以推理总是能够终止的。

但是，与正向推理一样，逆向推理也会产生循环推理链。设有下列三条规则：

$$(1)\ A\rightarrow B,\qquad (2)\ B\rightarrow C,\qquad (3)\ C\rightarrow A。$$

现在把 C 作为顶层目标，并假设当前的事实库是空的。为了得到有关 C 的证明，就要用规则(2)；而在引用规则(2)时，由于没有有关前提 B 的事实，所以要引用能得出关于 B 结论的规则(1)；同样，在引用规则(1)时，没有有关 A 的事实，故要引用规则(3)，在规则(3)的前提中又出现了目标 C。所以形成了如下的循环推理链：

$$C \xrightarrow{(2)} B \xrightarrow{(1)} A \xrightarrow{(3)} C \xrightarrow{(2)} B\cdots。$$

如果没有相应的推理控制机制，该推理链无法终止。下面是几种有用的推理控制策略，可以保证推理链在一定的条件下终止：

1. 事先设置事实库。

对于一些已经知道的基本事实，先放入到事实库中，这样可以

在一定程度上避免出现循环推理链。如在上例中，若把 A 作为基本事实先放入到事实库中，则在引用规则(1)时，由于前提中需要的事实已经存在而不必再引用规则(3)。

2. 循环标志法

即在每一个推理链的开始，记录每个推理结点的入口处，在同一推理链中一旦出现两个相同的推理入口地址，就中断本次推理过程。如上例中，推理链中第二次出现 C 时就可中断推理而不引用规则(3)，此时可以由用户输入关于 A 的事实，从而可确定规则(2)的前提是否被满足，进一步确定规则(2)的结论 B 是否成立，这样可确定顶层目标 C 是否成立。

在逆向推理中，一般不使用对推理深度的限制，因为从建立目标到解决目标的过程中，到底要形成多少子目标事先无法预料，所以若限制了推理深度，可能会得不到正确的推理结果。对逆向推理控制策略的设计，读者还可根据实际的领域特性加以修改。

9.1.3　混合推理

正向推理和反向推理是两种极端的推理方法。正向推理可以充分利用用户已知信息，但它有漫无目的地进行推理的趋势，与之相反，反向推理的目的性较强，但却不能充分利用用户已知信息。混合推理可以扬长避短，它既能充分利用现有信息，又能有目的地进行推理。这种推理方法是人们处理问题时常用的推理方法。医生诊断疾病的过程就是一个非常典型的混合推理的例子。在诊断疾病时，医生首先根据患者的各种症状形成患者有某种疾病的假设(正向推理)，为了证实他的假设，医生确定出下步采集哪些信息对于验证假设是有用的(反向推理)；为获得这些信息，他可能安排患者去做某些化验(相当于反向推理中要求用户提供信息)，当化验结果出来之后，医生可以根据这些信息判断他的假设是否成立；如果化验结果不能证实他的假设，医生又根据已知信息形成新的假设(正向推理)并为此寻找有用的信息(反向推理)，如此下去直至推断出患者的疾病。因此可以把混合推理看成是“正向推理——

反向推理"的循环。

除了能较好地避免盲目推理之外，混合推理可以避免盲目采集数据，这对采集数据需要较高代价的应用领域来说是很重要的。在医疗诊断领域中，有些化验项目是需要很高代价的，它们不仅需要较高的经济花费，而且对患者的身体有较大损害。因此除非有比较充分的证据表明确有这种需要，否则应尽量避免让患者做这些化验。混合推理收集数据（要求用户输入数据）是根据正向推理结果而进行的。这意味着它是充分考虑现有信息，然后选择对验证假设最有意义的数据去问用户，所以它能较有效地避免采集无用数据。

9.2 知识表示系统的不一致性

由于现实世界中的信息往往是不确定的，而且人们认识事物的过程也是不断发展变化的，各人还有不同的主观观点，对事物的观察也会有一定的偏差，甚至在有的情况下，还不得不采用估计和猜测的方法……因此，我们的知识表达系统中往往存在一定程度上的不确定性。在 Rough 集理论采用的决策表知识表达系统中，这种不确定性主要表现在几个方面，例如我们在第 6 章中讨论的，信息表中可能存在遗失数据，我们需要对之进行补齐（或完整化），这显然是对现实事物（数据）的猜测；在离散化处理过程中，显然降低了数据的表示精度；而且，由于一些观察失误或者观察不足，还可能导致信息之间的冲突。本节中，我们将对冲突信息进行分析讨论，为不一致性的处理提供思路。

通过对决策表的仔细研究，我们发现，在决策表中可能会存在如下三种不一致信息：

（1）决策表中包含冲突（矛盾）样本，即两个样本的条件属性取值完全相同，而决策（分类）属性的取值不同。这种不一致性的产生，主要有三种可能性：

• 一是条件属性不充分，根据所采用的条件属性不能对样本进行正确分类，必须增加额外的条件属性才能够正确区分样本；

• 二是样本属性值的测量或记录不准确；

• 三是在得到决策表的预处理过程中产生了冲突（如在离散化过程中可能将一些本来可以区分的样本变得不能区分了）。

（2）决策表中没有冲突的情况，在决策表化简过程中产生的不一致。对于本身一致或不一致的决策表，有的化简算法将导致一些新的不一致性信息，比如 Skowron 的缺省规则获取方法。另外，为了使学习得到的规则系统具有更大的适应能力，我们在处理过程中往往还需要有意地引入一些不确定性。

（3）决策表只包含了所有可能样本（或者样本全集，问题空间）中的一部分，没有包括所有可能出现的样本情况，即待识样本和决策表中的样本冲突。这其实是一种很平常的冲突情况，因为我们用于获取规则的决策表所包含的样本是有限的，仅是样本全集中的一个子集（甚至是一个很小的子集），从决策表中获得的规则，即使在决策表中没有冲突，也很可能对待识新样本作出矛盾的判定。

前两种不一致情况，是从待处理的决策表中就可以直接发现的，而第三种不一致性是在规则知识的获取过程中所不能够预料的，在发现不一致情况之前，我们不能肯定系统是否包含不一致性。这样，不一致性问题就成了归纳机器学习系统中固有的问题，如何处理这个问题就是一个关键的问题。不一致性处理的效果如何，就直接影响到系统的性能。我们接下来就讨论在不一致条件下的推理策略问题。

9.3　不一致推理策略

不一致情况下的推理，就是要研究在不一致规则的作用下，如何得到一个合适的结论，这也可以视为冲突消解问题。假设在一定前提条件下，多个规则得到满足，而它们的结论又不相同，这样，我

们怎样综合它们的结论，得到一个统一的、合适的结论呢？人们曾经对此问题进行了很多研究。下面对这个问题作一个介绍。

9.3.1 加权综合法

所谓加权综合法，就是为每条规则赋以一定的权值系数，将每条规则所得到的不同的结论进行加权求和，得到最终的结论。这种处理方法，适用于数值型规则，即最终所得结论为数值。通常，在模糊推理系统中采用这种策略。

9.3.2 试探法

试探法的基本思想是，如果发生多条不一致规则同时得到满足的情况，就分别对这些规则所得到的不同的结论进行进一步的推理，如果发现某些规则的后续规则得不到结果，或者得到的结果不理想，就可以考虑删除这条规则的结论，最终保留一个合适的结论。这种方法的代价就是搜索范围太大，而且，在一些控制系统中，推理过程是一步结束的，无法根据后续规则来选择合适的规则，这种情况下，这种处理策略就不一定合适。

9.3.3 高信任度优先法

在不确定推理中，每条规则往往都对应于一定的可信度，推理得到的结论也有一个与之相应的可信度。我们在第 5 章中介绍了几种不确定知识的表达处理方法。根据每条规则所得结论的可信度，我们可以选择结论可信度最高的规则。这种策略的思想是，如果规则结论的可信度高，则系统最终得到的结论将更可信，这有利于得到用户满意的结论。但是，这种方法也有局限性，比如，在几条不一致规则的结论同时具有最大可信度的时候，无法运用这一策略来选择合适的规则；另外，当一个规则的结论可信度最高，而其他很多规则的结论具有稍微小一点的可信度，这种情况下也很难说规则结论可信度最高的规则就是最合适的规则。

9.3.4 多数优先原则

每条规则都是归纳问题空间中的一定样本而得到的，每条规

则都有一定的代表性，它代表着一定范围的样本。多数优先的规则选择策略，就是认为覆盖多数样本的规则（即根据多个样本得到的规则）具有更大的适应性，具有得到合适结论的更高的概率。

我们采用定义 5.19 定义的不确定决策规则为例来说明多数优先的规则选择策略。

多数优先推理方法的基本思想是：假设有两条不一致的规则 R_1 和 R_2 同时与一个待识样本匹配，则：

若 $\alpha_1/\beta_1=\alpha_2/\beta$，则 $\gamma=\gamma_i$，$\beta_i=\max\{\beta_i|i=1,2\}$；

若 $\alpha_1/\beta_1\neq\alpha_2/\beta$，则

（1）若 $\beta_1=\beta_2$，则 $\gamma=\gamma_i$，$\alpha_i=\max\{\alpha_i|i=1,2\}$（即在频度一样的情况下选择可信度较大的那条规则的结论）；

（2）若 $\beta_1\neq\beta_2$，则

① 若 $\beta_1>>\beta_2$，$\alpha_1/\beta_1>>\alpha_2/\beta_2$（出现频度大的规则的可信度高），则 $\gamma=\gamma_1$；

② 若 $\beta_1>>\beta_2$，$\alpha_2/\beta_2>>\alpha_1/\beta_1$（出现频度大的规则的可信度低），则选取 $\gamma=\gamma_i$，$\alpha_i^2/\beta_i=\max\{\alpha_i^2/\beta_i|i=1,2\}$。

多数优先的规则选择策略，在一些问题中是比较常见的。比如在表决问题上，如果我们可以采取“少数服从多数”的原则。但是，在一些问题中，可能就不能取得好的结果。可以这样考虑，如果我们采取多数优先的策略，则难于得到特例的合适结果。

9.3.5　少数优先原则

基于对特例的考虑，我们认为，在一些问题中，特例是很重要的，我们必须予以考虑。这种情况下，我们就需要考虑采用少数优先的规则选择策略。

不一致情况下的少数优先推理算法：

输入：待识样本所匹配的规则集 $\Omega=\{R_i|i=1,\cdots,n\}$；

其中，规则 R_i 的参数记为 (α_i,β_i)，结论记为 γ_i，并且 $\gamma_i\neq\gamma_j(i\neq j)$。对于有多条结论相同的规则与待识样本匹配的情况，取其中

可信度最高的一条规则。

输出：待识样本的结论及其可信度。

第 1 步　确定样本的结论 γ：

$$\gamma=\gamma_i,\ 若\ \alpha_i/\beta_i^2=\max\{\alpha_i/\beta_i^2\mid i=1,\cdots,n\}。$$

（如果有多个最大的 α/β_i^2，则选择其中可信度最大的一个）

第 2 步　确定结论的可信度 CF：

$$CF=\min\{\alpha_i/\beta_i\mid i=1,\cdots,n\}。$$

（即结论的可信度取所有匹配规则中最小的一个，相当于规则的合取运算）

下面对这种方法的特性作一简单分析。

假设有两条不一致的规则同时与一个待识样本匹配，则：

若 $\alpha_1/\beta_1=\alpha_2/\beta_2$，则 $\gamma=\gamma_i$，$\beta_i=\min\{\beta_i\mid i=1,2\}$；

（即在可信度相同的情况下选择频度小的那条规则的结论）

若 $\alpha_1/\beta_1\neq\alpha_2/\beta_2$，则

（1）若 $\beta_1=\beta_2$，则 $\gamma=\gamma_i$，$\alpha_i=\max\{\alpha_i\mid i=1,2\}$；

（即在频度一样的情况下选择可信度较大的那条规则的结论）

（2）若 $\beta_1\neq\beta_2$，则

① 若 $\beta_1>\beta_2$，$\alpha_2/\beta_2>\alpha_1/\beta_1$（规则 R_2 的出现频度小而可信度高），则 $\gamma=\gamma_2$；

② 若 $\beta_1>\beta_2$，$\alpha_1/\beta_1>\alpha_2/\beta_2$（出现频度大的规则的可信度高），则选取 $\gamma=\gamma_i$，$\alpha_i^2/\beta_i=\max\{\alpha_i^2/\beta_i\mid i=1,2\}$。特例：$\alpha_1=\beta_1$（出现频度大的规则的可信度为 1），不妨假设 $\beta_2=\beta_1-a$，$\alpha_2=\beta_2-b$，则如果 $a\leqslant b$，则 $\gamma=\gamma_1$；否则，如果 $a^2/(a-b)>\beta_1$，则 $\gamma=\gamma_1$；如果 $a^2/(a-b)<\beta_1$，则 $\gamma=\gamma_2$；如果 $a^2/(a-b)=\beta_1$，则 $\gamma=\gamma_i$，$\alpha_i/\beta_i=\max\{\alpha_i/\beta_i\mid i=1,2\}$。

表面看来，多数优先策略和少数优先策略似乎是一对矛盾的规则选择策略。其实，我们并不是说这两种策略在各种情况下都适用，而是在不同的情况下使用不同的策略会得到不同的结果，有的情况适用多数优先策略，有的情况适用少数优先策略。如果在生成

规则的时候，出现频率高的样本和出现频率低的样本都有相应的规则来反映它们，那么多数优先策略应该更合适；如果在生成规则的时候，我们优先生成对应于出现频率高的样本，此时如果仍然采用多数优先策略，就将会导致出现频率低的样本所对应的情况会被完全忽略掉，这显然是不合适的，因此，在这种情况下，应该选择少数优先策略。

第 10 章　实例系统分析

为了系统说明基于 Rough 集理论的知识获取方法，让读者理解 Rough 集理论知识获取的系统建模过程，本章将详细介绍几个成功的应用 Rough 集理论解决实际问题的知识获取实例系统，如水资源调度、临床医疗诊断、客户行为分析、文本分类等，其他的很多应用领域，如光学字符识别、通信信道分配、自动控制等，由于篇幅关系，就不一一举例分析了。

10.1　水资源调度系统

对用户需求的预测是对供水系统进行优化控制的先决条件。如果能够精确地估计对水的需求，就可以计算出最小代价的抽水调度方案。加拿大里贾纳大学(University of Regina)的尼克·舒孔(Nick Cercone)，日阿可(Wojciech Ziarko)等人利用 Rough 集方法成功地对用水需求进行了预测。我们这里对此作一个简单的介绍。

10.1.1　系统概述

Nick Cercone，Wojciech Ziarko 等人考虑了北美一个中等规模城市的供水系统。水资源包括一个湖和一些地下井。水被首先抽到分布于城中的一些蓄水池中，然后从这些蓄水池抽到配给系统或者在需要调节水位的时候从一个蓄水池抽到另外的蓄水池。系统中的水压和流速可以通过抽水站的水泵和水阀来控制。目前，人工操作员在中心抽水站控制配给系统的操作。操作员利用一些启发知识或规则来最小化水泵动力的耗费，进行用水需求的预估，将各个蓄水池中的水保持在合理范围。这些启发知识是基于经济、

环境和社会因素综合得到的。由于配给系统有很多操作员，难于将配给系统的操作进行标准化和优化。将经验最丰富的操作员的启发知识置入专家系统是降低操作代价的一个有效途径。为此，通过与操作员的交流学习得到了一些启发知识。对这些启发知识分析发现，为了制定最小代价的抽水调度方案，很重要的就是对用水需求进行准确的每日预测。然而，专家(操作员)是根据他们的经验来进行大致估计的，难于理解。而且，不准确的估计还导致低效的控制。因此，人工知识获取是不足以处理复杂工程应用中所产生的问题的。于是，Wojciech Ziarko 等人就基于 Rough 集研究了从大量日用水需求数据中获取控制规则的问题。由于数据样本是不完备和含糊的，必须从不完备数据集中获取分类知识。

10.1.2　数据采集和表示

市内对于水的瞬间消耗是由分布在该区域的大量工厂、商店、公共用户和居民用户的用水情况决定的。这种消耗受诸多因素影响，如天气条件、季节变换、星期几、节假日等。因此，对城市供水系统的总需求是随时间变化的、周期性的不稳定序列，难于仅仅通过计算方法对其建模。

从众多考虑因素中，我们可以分析得到对影响用水需求比较重要的 18 个因素，如表 10.1 所示。第一个需要考虑的因素是星期几，因为周末的日总用水量通常小于周日。城市也可以通过法律禁止在星期三对草坪浇水。而且，星期一是用水的高峰，因为很多人星期一洗衣。夏天，人们周末度假回来通常在星期一对自家的草坪浇水。其他 17 个因素是连续 3 天的气温、湿度、刮风、日照时间等。这些因素的取值是从加拿大环境部门提供的月气象报告中得到的。

对于决策信息，城市记录了关于用水消耗的历史数据，根据从各个抽水站的日配给流量表数据计算得到日用水需求。日用水需求总量冬季最少的时候为 50 ML，夏季最多为 180 ML。

表 10.1 用水需求预测的条件因素

编号	条件属性
a_1	Day of week
a_2	Today's maximum temperature
a_3	Today's minimum temperature
a_4	Today's average humidity
a_5	Today's rainfall
a_6	Today's snowfall
a_7	Today's average speed of wind
a_8	Yesterday's maximum temperature
a_9	Yesterday's minimum temperature
a_{10}	Yesterday's average humidity
a_{11}	Yesterday's rainfall
a_{12}	Yesterday's average speed of wind
a_{13}	Yesterday's bright sunshine hours
a_{14}	The day before yesterday's maximum temperature
a_{15}	The day before yesterday's average humidity
a_{16}	The day before yesterday's rainfall
a_{17}	The day before yesterday's average speed of wind
a_{18}	The day before yesterday's bright sunshine hours

分类系统(信息表)基于属性值来表示环境状态的分类。这里用于用水需求预测的观察样例集包含 300 多个样例,反映了从 1994 年 3 月到 11 月间每天的条件因素和决策属性的信息。表 10.2是其中 8 个样例在 14 个条件属性上的投影。这里,为了适应 Rough 集方法处理,一些原始属性值已经被离散值取代了,例如属性 a_2(minimum temperature)被离散化为 0,1,2,3,4,5,6,7,8,9 这 10 个等级,而在表 10.2 所示的 8 个样例中只出现了 7,8 两个等级,它们分别代表区间(6.46,10.34]和(10.34,14.22]。

表 10.2　用水需求预测分类系统

Objects	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	D
Obj_1	6	6	7	7	9	7	1	1	0	1	3	2	6	0	0
Obj_2	1	5	7	6	9	9	0	0	0	0	1	5	1	0	0
Obj_3	6	6	7	7	9	7	1	1	0	1	3	2	6	0	0
Obj_4	3	7	7	7	5	2	0	0	0	0	1	3	1	2	1
Obj_5	3	7	7	7	8	2	0	0	0	0	1	7	1	9	1
Obj_6	3	7	7	7	5	2	0	0	0	0	1	3	1	2	1
Obj_7	6	6	8	7	3	2	0	0	0	0	6	0	6	2	2
Obj_8	3	7	7	7	5	2	0	0	0	0	1	3	1	2	2

10.1.3　数据分析

分类系统仅仅包含了全域中部分样例子集的信息，而且所选择的属性集合对于表征这个样例子集不一定是充分的。任意两个样例，只要其属性值相同就认为是不可区分的。也就是说，利用所选择的属性和值可能不能区分所有的样例。表 10.2 中，Obj_1 和 Obj_3 就是根据这 14 个条件属性和决策属性所不能够区分的，Obj_4 及 Obj_6，Obj_8 也是根据这 14 个条件属性所不能够区分的。

由于这个问题中有很多不确定因素，采用了本书 3.5 节中介绍的可变精度 Rough 集模型。

假定一个概念 $Y_1=\{Obj_4,Obj_5,Obj_6\}$，则

1. 令 $\beta=1$，概念 Y_1 的 β 正域为 $\mathrm{POS}_C^{\beta}(Y_1)=\{Obj_5\}$；

概念 Y_1 的 β 负域为

$$\mathrm{NEG}_C^{\beta}(Y_1)=\{Obj_1,Obj_2,Obj_3,Obj_7\};$$

概念 Y_1 的 β 边界域为

$$\mathrm{BND}_C^{\beta}(Y_1)=\{Obj_4,Obj_6,Obj_8\}。$$

2. 令 $\beta=0.6$，概念 Y_1 的 β 正域为

$$\mathrm{POS}_C^{\beta}(Y_1)=\{Obj_4,Obj_5,Obj_6,Obj_8\};$$

概念 Y_1 的 β 负域为

$$\mathrm{NEG}_{\beta}^{\prime}(Y_1)=\{Obj_1,Obj_2,Obj_4,Obj_7\};$$

概念 Y_1 的 β 边界域为

$$\mathrm{BND}_{\beta}^{\prime}(Y_1)=\varnothing$$

根据 3.5 节中介绍的约简概念，可以得到信息系统的很多约简。对于表 10.2 所示的信息系统，就可以有 20 种不同的约简结果。表 10.3 是一个针对概念 Y_1 的部分约简结果。Y_1 列为 1 的样例属于概念 Y_1 的 β 正域，Y_1 列为 0 的样例属于概念 Y_1 的 β 负域。这里，$\beta=0.6$。

表 10.3 约简结果

Objects	C			Y_1	Y 中的样例数	$\neg Y_1$ 中的样例数
	a_2	a_4	a_7			
Obj_1,Obj_5,Obj_6,Obj_8	7	7	0	1	3	0
Obj_1,Obj_4	7	7	1	0	0	2
Obj_2	7	6	0	0	0	1
Obj_7	8	7	0	0	0	1

10.1.4 规则生成

根据约简后得到的信息系统，对于约简结果中的每行（约简后条件属性集的不可区分关系）X_i，我们可以直接得到如下形式的概率决策规则：

(1) $\mathrm{Des}(X_i)\xrightarrow{C_i}\mathrm{Des}(Y)$，if $P(Y|X_i)\geqslant\beta$，

(2) $\mathrm{Des}(X_i)\xrightarrow{C_i}\mathrm{Des}(\neg Y)$，if $P(Y|X_i)\leqslant 1-\beta$，

其中，C_i 是规则的可信度因子，在(1)式中等于 $P(Y|X_i)$，在(2)式中等于 $1-P(Y|X_i)$。

从表 10.3 中可以得到如下规则：

$$(a_{10}=7)\wedge(a_{11}=7)\wedge(a_{12}=0)\xrightarrow{0.75}(d=1),$$

$$(a_{10}=7)\wedge(a_{11}=7)\wedge(a_{12}=1)\xrightarrow{1}(d\neq 1),$$

$$(a_{10}=7)\wedge(a_{11}=6)\wedge(a_{12}=0)\xrightarrow{1}(d\neq 1),$$

$$(a_{10}=8)\wedge(a_{11}=7)\wedge(a_{12}=0)\xrightarrow{1}(d\neq 1).$$

正如我们在值约简部分讨论的那样，这些规则中的一些条件属性是冗余的，还需要通过值约简进行进一步的简化。这里，采用了决策矩阵的方法来进行值约简，最终可得到如下简化决策规则：

$$(a_{10}=7)\wedge(a_{11}=7)\wedge(a_{12}=0)\xrightarrow{0.75}(d=1),$$

$$(a_{12}=1)\xrightarrow{1}(d=0),$$

$$(a_{11}=6)\xrightarrow{1}(d=0),$$

$$(a_{12}=8)\xrightarrow{1}(d=2)。$$

10.1.5　实验结果

如前所述，在供水调度系统中有从 10 个月时间中收集的 300 多个训练数据。实验中决策属性值被离散化为 10 个区间，即信息系统具有 10 个概念。下面对获取到的一些规则进行分析。

关于概念 $D=(53\sim 63]$的最一般规则是

$$(-6.70<a_{10}\leqslant -1.74)\wedge(-6.70<a_{12}\leqslant -1.74)\wedge$$

$$(11.86<a_{11}\leqslant 17.42)\xrightarrow{1}(53<D\leqslant 63)。$$

这条规则覆盖了该概念 $D=(53\sim 63]$所有的样例中的 25%，其含义为：如果今天和昨天的最低气温均在 -6.70℃～-1.74℃之间，并且前天的最高气温在 11.86℃～17.42℃之间，则用水需求在 53 ML～63 ML 之间，其可信度等于 1。

关于概念 $D=(63\sim 73]$的最一般规则是

$$(81<a_{10}\leqslant 87)\xrightarrow{1}(63<D\leqslant 73)。$$

这条规则覆盖了该概念 $D=(63\sim 73]$所有的样例中的 12%，

其含义为：如果昨天的平均湿度在81%～87%之间，则用水需求在63 ML～73 ML之间，其可信度等于1。

关于概念 $D=(73\sim84]$ 的最一般规则是

$$(8.7<a_7\leqslant12.9)\wedge(-1.74<a_9\leqslant3.22)\xrightarrow{1}(73<D\leqslant84).$$

这条规则覆盖了该概念 $D=(73\sim84]$ 所有的样例中的6.25%，其含义为：如果今天的平均风速在8.7 km/h～12.9 km/h之间，并且昨天的最低气温在－1.74℃～3.22℃之间，则用水需求在73 ML～84 ML之间，其可信度等于1。

关于概念 $D=(94\sim104]$ 有一条规则是

$$(47<a_4\leqslant53)\wedge(17.42<a_8\leqslant22.98)\wedge(3.22<a_9\leqslant8.18)\xrightarrow{1}(94<D\leqslant104)。$$

这条规则覆盖了该概念 $D=(94\sim104]$ 所有的样例中的17.6%，其含义为：如果今天的平均湿度在47%～53%之间，昨天的最高气温在17.42℃～22.98℃之间，并且昨天的最低气温在3.22℃～8.18℃之间，则用水需求在94 ML～104 ML之间，其可信度等于1。

关于概念 $D=(104\sim114]$ 的最一般规则是

$$(22.98<a_2\leqslant28.54)\wedge(53<a_{15}\leqslant58)\wedge(4.5<a_{17}\leqslant8.7)\xrightarrow{1}(104<D\wedge114)。$$

这条规则覆盖了该概念 $D=(104\sim114]$ 所有的样例中的33.3%，其含义为：如果今天的最高气温在22.98℃～28.54℃之间，前天的平均湿度在53%～58%之间，并且前天的平均风速在4.5 km/h～8.7 km/h之间，则用水需求在104 ML～114 ML之间，其可信度等于1。

关于概念 $D=(114\sim124]$ 的最一般规则是

$$(36<a_{10}\leqslant41)\wedge(36<a_{15}\leqslant41)\xrightarrow{1}(114<D\leqslant124)。$$

这条规则覆盖了该概念 $D=(114\sim124]$所有的样例中的33.3%，其含义为：如果昨天和前天的平均湿度均在36%～41%之间，则用水需求在114 ML～124 ML之间，其可信度等于1。

关于概念 $D=(124\sim134]$的最一般规则是

$$(53<a_1\leqslant 58)\wedge(22.98<a_{14}\leqslant 28.54)\wedge$$

$$(13.30<a_{15}\leqslant 15.20)\xrightarrow{1}(124<D\leqslant 134)。$$

这条规则覆盖了该概念 $D=(124\sim134]$所有的样例中的66.7%，其含义为：如果今天的平均湿度在53%～58%之间，前天的最高气温在22.98℃～28.54℃之间，并且前天的日照时间在13.30 h～15.20 h之间，则用水需求在124 ML～134 ML之间，其可信度等于1。

这个系统经过测试，其预测的错误率最好达到6.67%，平均错误率为10.27%。

10.1.6　讨论

这里介绍的是一个根据训练数据获取得到预测规则知识的系统，该系统考虑和利用了知识系统中固有的统计信息，能够得到具有一定决策概率的不精确决策规则。这种方法也可以在其他领域中应用，例如一般客户需求预测、故障诊断和过程控制等。随着数据的增加，得到的规则性能还会得到改善，而且，还可以将得到的规则拿给领域专家进行验证测试。

10.2　临床医疗诊断系统

医疗诊断是基于Rough集理论进行知识获取研究的一个重要实践领域。日本岛根医科大学津本周作(Shusaka Tsumoto)博士领导的研究组在利用Rough集进行临床医疗诊断方面进行了很多工作。本节将对Tsumoto博士基于Rough集模型研究从临床数据中获取近似概念(规则)的工作作一介绍，分析如何将

Rough 集理论模型应用于临床诊断。

10.2.1 临床诊断概述

在一般的规则知识获取模型中，人们主要是从数据库中获取正知识(positive knowledge，即根据症状对可能的病症作出判定)，很少考虑获取负知识(negative knowledge，即根据症状排除某些病症的可能性)，这些负规则，往往在诊断过程中是很重要的，医生需要根据这些知识来判定病人是否在服用某些药物之后引起副作用。

规则获取包括获取确定性规则和概率规则两种。确定性规则可以视为 if-then 规则形式的命题，从集合观点看，满足确定性规则条件部分(C)的样例集合是该规则结论部分(D)的样例的一个子集，即($C\subseteq D$)。概率规则可以视为带概率信息的 if-then 规则，从集合观点看，C 不是 D 的子集，但 C 与 D 很大程度重叠，即 $C\cap D\neq\varnothing$ 且 $\frac{|C\cap D|}{|C|}\geq\delta$。概率规则是包含很多正例和少数反例的。确定性规则和概率规则的一个共同特点就是在一个样例满足其条件时推理得到相应的结论(规则结论部分 D 正确)。这种推理称为正推理。

然而，医生不但要用正推理，还要用负推理对病人进行诊断。负推理规则的结论中包含负项。例如，如果一个头疼病人没有跳动着作痛，偏头疼假设就不应该有高的概率。因此，负推理在减小诊断过程搜索空间上起着很重要的作用。这也是医生难于解释归纳规则和诊断过程的原因之一。因此，获取得到反映领域专家决策过程规则和专家易于理解的规则，是加强专家和计算机合作获取知识过程中的重要因素，为此，我们需要从数据库中获取得到负规则。

医疗推理的一个重要特征是注意机制，即从众多候选中选择最终诊断结论。例如，在不同的头疼诊断中，将需要检查 60 多种疾病，包括身体检查和化验。在诊断中，如果一种诊断结论所需要的某个症状没有观察到，就可以排除这种候选。这种诊断推理包括如

下两种推理过程：排除推理(exclusive reasoning)和包含推理(inclusive reasoning)。诊断过程如下进行：首先，排除推理在病人没有诊断某种疾病所需要的某个症状时从候选中排除该疾病；其次，包含推理在病人出现某种疾病特有的症状时在排除推理的输出结果中得到该疾病的假定。这两个过程分别使用负规则(排除规则)和正规则(包含规则)。

为了进一步讨论正规则和负规则的获取以及推理，我们首先来对这里所要用到的概率规则形式进行简单的介绍。

10.2.2　概率规则

设论域 U 是一个有限的非空集合，A 是非空的有限属性集，对于属性 $a\in A$，V_a 是 a 的域。决策表 $A=(U,A\cup\{d\})$ 是一个信息系统。属性集 $B=A\cup\{d\}$ 和值域 V 上的原子公式是形如 $[a=v]$ 的表达式，称为 B 上的描述符。B 上原子公式的集合 $F(B,V)$ 是包含 B 上所有原子公式的最小集，且对于析取、合取、非运算是封闭的。对于任一公式 $f\in F(B,V)$，f_A 是公式 f 在信息系统 A 中的解释(含义)，即论域 U 中具有特性 f 的所有对象(样例)的集合，可以归纳定义为：

(1) 如果 f 形如 $[a=v]$，则 $f_A=\{x\in U\mid a_x=v\}$；

(2) $(f\wedge g)_A=f_A\cap g_A$，$(f\vee g)_A=f_A\cup g_A$，$(\neg f)A=U\setminus fA$。

由此，可以进一步定义分类正确度(精度)、分类覆盖度或真正率(true positive rate)如下：

定义 10.1　令 R 和 D 表示 $F(B,V)$ 中的一个公式和属于决策 d 的样例集。规则 $R\to D$ 的分类正确度和覆盖度(真正率)定义为：

$$\alpha_R(D)=\frac{|R_A\cap D|}{|R_A|}(=P(D|R)),$$

$$\kappa_R(D)=\frac{|R_A\cap D|}{|D|}(=P(R|D))。$$

其中，$\alpha_R(D)$ 为 R 对分类 D 的分类正确度，$\kappa_R(D)$ 为 R 对 D 的覆

盖度(真正率)。注意,正确度对应于 D 对 R 的条件概率,覆盖度对应于 R 对 D 的条件概率。

利用分类正确度和覆盖度可以定义如下的概率规则:

$$R \xrightarrow{\alpha,\kappa} d, \text{ 其中 } R=\bigwedge_j \bigvee_k [a_j=v_k],$$
$$\alpha_R(D)\geqslant\delta_\alpha,\ \kappa_R(D)\geqslant\delta_\kappa。$$

正规则是只由正样例支持的规则,其分类正确度为 1。用 Rough 集的观点看,就是支持正规则的样例集合是目标样例集的下近似的子集。因此,正规则可表示为:

$$R \longrightarrow d, \text{ 其中 } R=\bigwedge_j [a_j=v_k],\quad \alpha_R(D)=1.0。$$

正规则经常也称为确定性规则。这里,我们为了和下面的负规则区别,称之为项和正确定规则。

在定义负规则之前,我们还需要定义一个排斥规则(Exclusive rule)的概念。排斥规则是由所有正例支持的规则,其覆盖度为 1。用 Rough 集的观点看,就是支持排斥规则的样例集合是目标样例集的上近似。因此,排斥规则可表示为:

$$R \longrightarrow d, \text{ 其中 } R=\bigvee_j [a_j=v_k],\quad \kappa_R(D)=1.0。$$

以概率逻辑的观点看,排斥规则的条件是其结论 d 的必要条件,排斥规则应该表示为:

$$d \longrightarrow \bigvee_j [a_j=v_k]。$$

由此,我们可以相应于排斥规则定义负规则为:

$$\bigwedge_j \neg[a_j=v_k] \rightarrow \neg d,$$

负规则意味着,如果一个样例不满足负规则中任意属性值对的要求,我们就可以从候选中排除结论 d。负规则也可以作为一类确定性规则,其覆盖度为 1。还应该注意,支持负规则的样例集对应于 Rough 集理论中所讲的负区域的一个子集。

10.2.3 规则获取算法

算法 10.1 是一个规则获取的基本算法。

算法 10.1　分类规则获取算法

```
Procedure  概率规则获取:
    var
      i: integer; M,L_i:List.
    begin
      L_i:= L_in;        /* L_in: 原子关系列表 */
      i:=1;
      M:={};
      for i:=1 to n do        * n: 属性总数 */
        begin
          while (L_i≠{ }) do
            begin
              Select one pair R=∧[a_i=v_j] from L_i;
              L_i:= L_i\{R};
              if (α_R(D)≥δ_α) and (κ_R(D)≥δ_κ)
                then do S_i:= S_i+{R};          /* R 是概率规则 */
                 else M:=M+{R};
             end
            L_i+1:=M 中合取公式的完全组合列表;
        end
    end
```

该算法首先将 L_1 赋值为一个由单描述符组成的公式的列表 L_{in}。然后，进行如下循环，直至 L_1 变为空：(a)从 L_1 中去掉公式[$a_i=v_j$]；(b)如果 $\alpha_R(D)$，$\kappa_R(D)$均大于其阈值，则该规则可以放入正规则条件部分的列表之中，否则，将该公式放入 M 用于继续产生合取公式。最后，当 L_1 变为空后，就产生了其后续 L_2，对 L_2 进行上述循环处理。

正规则获取算法是这个规则获取算法的特例，而负规则获取算法既包含该算法，同时也包含另一种属性选择。下面对这两种算

法进行介绍。

算法 10.2 负规则获取算法

```
Procedure 排斥和负规则获取算法
  var
    L:List;        /* 原子属性值对列表 */
  begin
    L:=P0;         /* P0:数据库中给定的原子属性值对列表 */
    while (L≠∅) do
      begin
        Select one pair [ai=vj] from L;
        if ([ai=vj]A∩D≠∅) then do        /* D:目标类 d 的正例 */
          begin
            Lir:=Lir+[ai=vj];                    /* 候选正规则 */
            if (κ[ai=vj](D)≥δκ)          /* δκ:给定的覆盖度阈值 */
              then Rer:=Rer∧[ai=vj];
                                  /* 将[ai=vj]加入排斥规则公式 */
          end
        L:=L-[ai=vj];
      end
      构造负规则:取为相应的 Rer。
  end
```

本算法的执行过程为:首先,从属性值对列表 L 中选择一个描述符 $[a_i=v_j]$。然后,检查该描述符是否和一个正例集合 D 相交,如果相交,就将该描述符放入候选正规则列表。再检查其覆盖度是否大于规定的阈值(这里阈值为 1),如果其覆盖度等于 1,则将该描述符加入 D 的排斥规则的前提条件部分公式 R_{er} 之中。接着,从列表 L 中删去 $[a_i=v_j]$,反复上述处理直至 $L=\varnothing$。最后,将所得到的排斥规则作为负规则输出。

算法 10.3　正规则获取算法

在正规则的获取过程中，正确度和覆盖度的阈值分别被设置为 1.0 和 0.0。采用算法 10.1 的一个特例。其执行过程可以描述为：首先，将 L_1 赋值为前面算法 10.1 得到的由单描述符组成的公式列表 L_{n-1}。然后，循环执行如下过程，直至 L_1 为空：(a) 从 L_1 中删去公式 $[a_i=v_j]$；(b) 检查 $\alpha_R(D)$ 是否大于阈值（这里即是检查 $\alpha_R(D)$ 是否等于 1），若大于，则将该公式放入正规则前提条件列表，否则，将其放入列表 M 中用于继续产生合取公式。最后，当 L_1 变为空后，就从列表 M 中产生其后续 L_n。

10.2.4　实验结果

利用前一节介绍的算法，Tsumoto 博士领导的研究组开发了一个专家系统概率规则获取系统 PRIMEROSE－REX2（Probabilistic Rule Induction Method for Rules of Expert System ver 2.0）。下面介绍这个系统在如下两个医疗数据库中的应用结果：

1. 抗生素过敏（allergy for antibiotic），训练集包括 31 119 个样例，包含 137 个属性，分为 4 个类。

2. 类固醇副作用（side－effects of steroid），训练集包括 3 620 个样例，包含 285 个属性，分为 11 个类。

在抗生素过敏反应研究中，得到了医生没有预料到的下述正规则：

$$[\text{Sex}=\text{F}]\wedge[\text{Food}=\text{Fish}]\rightarrow[\text{Effect}=\text{Urticaria}],$$

$$\text{Age}\leqslant 40]\wedge[\text{Food}=\text{Fish}]\rightarrow[\text{Effect}=\text{Urticaria}]。$$

有趣的是，医生通常认为年龄和性别不是判断过敏反应的重要属性，而规则获取实验却得到与年龄和性别有关的规则。这个发现表明，妇女比男性更容易得风疹（urticaria），年龄小于 40 岁也是产生过敏反应的重要因素。对于这一新发现的知识，Tsumoto 博士在医疗实践中进行了进一步的测试，实践证明其正确率达到 82.6%。

在类固醇副作用研究中，也得到了一些有趣的规则，其中关于丘脑出血的下列规则最为有价值：

$\neg[Sex=F] \wedge \neg[OGTT=DM] \rightarrow \neg[Effect=DM]$，

$[Hypertension=yes] \wedge \neg[Sex=F] \rightarrow \neg[Effect=HT]$。

由此可以得到有趣的知识，对于女性而言，OGTT=DM 是诊断由类固醇导致的糖尿病(diabetus mellitus)的重要因素。

上面所介绍的规则获取还仅仅是关于常规规则的获取，获取规则的过程是在数据库中一次性完成的。然而，正确度和覆盖度阈值的任意设置不一定能够发现有趣的知识，例如，如果覆盖度阈值设得太高，将只能得到没有价值的规则。为了克服这个问题，必须进行循环调整，但过多的循环又是很费时的，需要一个好的循环规则获取策略。

一种可行的策略是将覆盖度和正确度作为平衡一般化(generalization)和特殊化(specialization)的指标。覆盖度越高的规则的长度越短，正确度越高的规则的长度越长。因此，覆盖度可以用于获取一般(常规)规则，正确度可以用于获取特殊规则。这两个指标可以进行如下组合：首先，得到高覆盖度的一般规则。如果我们得到了没有预料到的、有趣的高覆盖度规则，就可以接着用这些规则作为进一步获取高正确度特殊规则的基础。这些高正确度的规则的预测能力弱，但它们反映了规则的特征，对应于隐藏在数据背后的知识。

Tsumoto 博士利用细菌测试数据库(bacterial test dataset)进行了循环规则获取实验研究。该数据库包含 101 343 个样例，有 254 个属性。这个数据库是从 1994 年到 1998 年 5 年中收集得到的，每年大约有 20 000 个样例。循环规则获取实验包含如下两步：

第 1 步　获取一般规则。

覆盖度和正确度的阈值分别设为 0.5 和 0.1，首先发现具有高覆盖度和适当正确度的规则。从 5 年的数据中总共得到 24 335 条规则，其中有 114 条规则是医生没有事先预料到的有趣规则。下

面 9 条规则最引起医生的注意，他们从每年的数据中都可以得到。

1. [β -lactamase = (-)] →Bacteria Detection(+)
（正确度：0.667，覆盖度：0.12）；

2. [β -lactamase = (3+)] →Bacteria_Detection(+)
（正确度：0.702，覆盖度：0.553）；

从感染历史的角度看，这些规则很有趣：抗青霉素细菌成了细菌感染的重要原因。

3. [Disease = Pneumonia] →Bacteria_Detection(—)
（正确度：0.826，覆盖度：0.12）；

4. [Fever(BT>=39)]→Bacteria Detection(-)
（正确度：0.790，覆盖度：0.11）；

5. [Disease = MalignantTumor]→Bacteria_Detection(-)
（正确度：0.77，覆盖度：0.13）；

这三条规则说明这些检查对诊断细菌没有重要价值。

6. Fusobacterium→PCG(Sensitive)
（正确度：0.92，覆盖度：0.26）；

7. MRSA→VCM(Sensitive)
（正确度：0.89，覆盖度：0.12）；

8. Tonsilitis→AUG(S)
（正确度：0.84，覆盖度：0.10）；

这三条规则显示了没有预料到的细菌和抗生素之间的关系。

9. [Dept. = Neurology]→MRSA
（正确度：0.6，覆盖度：0.10）；

这条规则发现了神经保护(neurological ward)和 MRSA 感染之间的关系。

表 10.4 中的 5 条规则说明了 1994 年到 1997 年和 1998 年之间的很大差别。这些规则显示出在 1998 年的数据中有很显著变化的模式存在。虽然他们很简单，但这些简单的规则可能体现了支持这些规则的数据中隐含的知识。特别是第 3 条规则，医生是没有预

见到这条规则的，因为亚氧性细菌(anaerobic bacteria)感染在心脏病科(cardiology)中是很少发生的。

表 10.4 1994 年到 1997 年和 1998 年数据中得到规则的差别

1994～1997	1	[Dept. =Surgery]→Anaerobic	正确度：0.65±0.03 覆盖度：0.64±0.05
	2	[Dept. =Emergency]→Anaerobic	正确度：0.73±0.04 覆盖度：0.20±0.03
1998	3	[Dept. =Cardiology]∧[Material =Blood]→Anaerobic	正确度：0.62 覆盖度：0.31
	4	[Dept. =Surgery]→Anaerobic	正确度：0.76 覆盖度：0.28
	5	[Dept. =Emergency]→Anaerobic	正确度：0.67 覆盖度：0.21

通过上面的分析，我们可以进一步在 1998 年的数据中去获取高正确度的特殊规则。为此，我们采用正规则获取算法。

第 2 步 获取正规则。

从 1998 年的数据中，得到 493 条规则，其中 25 条规则包含了属性值对[Dept. =Cardiology]。下面这条规则最让医生感兴趣：

[Dept. =Cardiology]∧[Ward=ICU]∧[Material=Blood]∧[Catheter=CVP]→ Anaerobic

(正确度：1.00，覆盖度：0.10)。

使用这条规则的结果是在医院内的感染应该有很大的可能性，因为规则中包含了两个在医院内感染的重要关键词：ICU 和 CVP。

10.2.5 讨论

正规则完全等价于 Rough 集理论中的确定性规则。因此，正规则的析取对应于目标概念(决策属性)的正域。另一方面，负规则对应于目标概念(决策属性)的负域。从这个角度看，概率规则就对

应了边界域和正域的和(主要是边界域)。

循环规则获取对于发现数据中特殊的结果是重要的。通常,在机器学习中这被误认为是过度针对数据。然而,规则获取也不能严格地去获取高预测能力的规则,因为高预测能力的规则对于领域专家是微不足道的。相反,特殊的模式恰恰是对于领域专家具有吸引力的,因为它们包含了有关数据收集环境的信息。虽然特殊模式对于形成规则不是很有用,但从不同的观点来看,它们具有重要信息。例如,前一节中包含[Dept. =Cardiology]和[Ward=ICU]的规则就是这个数据环境特有的,它在亚氧性细菌感染的一般知识中不常出现,但由它可以导致发现医生所没有预计到的知识。

10.3　市场潜在客户预测

电子商务在国内外都已经逐渐开展起来,分析客户的特点是最大限度提高商业活动的效率和成功机会、实现最大商业价值、降低商业成本的重要因素。本节将介绍 Poel 和 Piasta 开发的一个分析潜在客户的应用实例(Prob Rough)。

10.3.1　系统概述

在个体客户层次预测客户未来的购买行为是电子商务中的一个关键问题,这需要从现有的客户爱好数据中分析得到能够用于有目的选择客户的模式信息。由于通过直接邮寄广告(电子商函)、目录、Internet 网络等方式可以直接对单个客户进行处理,这才使得对个体客户进行预测和分别处理成为可能。这与传统的电视广告和印刷宣传广告不同,它们不具有这种性质。本系统(Prob Rough)用概率 Rough 分类来预测可能的购买行为,发现客户行为模式的知识,并将这些知识表达为用户易于理解的形式。Prob Rough 系统可以处理具有遗失属性值的不一致、噪音数据。背景知识包括决策的先验概率和错误分类代价。

10.3.2 知识获取过程

邮购公司经常遇到的一个问题是如何判定是否应该向特定的客户邮寄商品目录。随着邮资的提高和竞争的加剧，这显得越来越重要。这里将决策问题视为分类问题，根据所有可以得到的数据对客户是否会在一定时期内购买（或重买）进行预测。因此，得到的结果是二值的（0/1）。销售经理利用这个预测分类技术对所有客户进行分析，将商品目录寄给最有可能购买的那些客户。邮购公司根据成本/效益平衡分析和预算情况决定将商品目录邮寄给多少客户。这里有两个重要的问题需要考虑：一是先验概率；二是错误分类的代价。先验概率问题主要是，客户数据库中购物客户和不购物客户的比例通常并不代表所有客户中的实际比例。第二个问题主要是，将一个购物客户误认为不购物客户的代价（错过一个销售机会）比将一个不购物客户误认为购物客户的代价（一次邮资）高得多。

Prob Rough 是一个产生概率 Rough 分类的数据挖掘系统，它能够处理这里的两个关键问题，即先验概率和不同的错误分类代价。Prob Rough 系统产生的决策规则具有如下形式：

$$\Delta = \Delta_1 \times \Delta_2 \times \cdots \times \Delta_m,$$

其中，如果第 q 个属性的值是定的，则 Δ_q 是一个区间，否则是一个属性子集。这些 Δ_q 被称为划分（片段）。Δ 被称为属性空间中的可行子集。

得到分类规则知识的算法包括如下两个基本阶段：

1. 属性空间的全局划分；

2. 决策规则的约简。

在第一个阶段，Prob Rough 尽量减小制定决策的平均全局代价。这个阶段要对整个属性空间进行大量的划分。划分的数量是一个参数，也是算法迭代的次数。为了提高算法的预测质量，可以对迭代次数进行优化。每一步迭代将一个属性（或其子集）划分为两个片段。对于连续属性，必须事先离散化处理。没有包括在划分过程中的属性将被删掉（决策的时候将不考虑这些属性）。划分的

结果是将属性空间唯一地划分为可行子集。每个划分单元与一组具有相等重要性的决策相关联。在第二个阶段，我们只考虑那些与最小代价标准值关联的划分。这个阶段，只要具有相等重要性的决策集合的交集非空，划分单元就被放入更大的可行子集。

从欧洲一家匿名邮购公司的数据库中随机得到一个包含6 800个样本的数据集，其中50%的客户在6个月内对收到的商品目录有了回应，而50%的客户没有回应。将这个数据集随机地分为学习数据集和测试数据集。根据以往的交易数据可以构造很多属性，其中三个对于预测而言是特别重要的，即最近购买时间(recency)、购买频率(frequency)和利润(monetary)。表10.5给出了一些RFM变量的例子，其中所有变量均是经过离散化处理的，并由邮购公司为个体客户提供具体的取值。

表 10.5　数据集中属性的描述

变量名	类型	注　　释
Buy_{-1}	0/1	客户是否在前6个月中购买过？
Buy_{-2}	0/1	客户是否在6个月以前到1年以前时间中购买过？
Buy_{-3}	0/1	客户是否在1年以前到1年半以前时间中购买过？
Buy_{-4}	0/1	客户是否在1年半以前到2年以前时间中购买过？
Customer	6 cat.	该人作为客户以来的时间？
LastFreq	9 cat.	该客户在前6个月中的购买频率？
LastSales	5 cat.	该客户在前6个月中的购买量？
LastProfit	10 cat.	前6个月中从该客户得到的利润？
DaysSince	7 cat.	前一次购买至今的天数？
Unimulti	3 cat.	客户是居住在单独房子还是公寓？
Socclass	6 cat.	客户的社会分类？
VAT	0/1	客户是否是老板？
Household	4 cat.	客户拥有房子的类型？
Family	4 cat.	客户家的人口数？
Natclass	6 cat.	客户居住街道的种族分布？
State	9 cat.	客户所在的省？

10.3.3 实验结果

利用相等先验概率和错误分类代价，以及两种不相等先验概率和错误分类代价先验知识，实验得到了如下的结果：

1. 相等先验概率和错误分类代价。

表 10.6 中的等价的决策规则集合是 ProbRough 系统在相等的购买和不购买先验概率、相等的错误分类代价前提下得到的。对每个客户，基于这三组等价规则集的任何一个，都可以得到决策。客户首先被划分到寄(Do mail)或者不寄(Do not mail)，然后辅以各条规则的强度。可以用规则的强度来对划分中的客户进行排序。

表 10.6 在相等的购买和不购买先验概率、相等的错误分类代价前提下得到的决策规则集

1 **if** LastFreq>0	**then** d=Do mail
2 **if** LastFreq=0 **and** Buy_{t-2}=yes	**then** d=Do mail
3 **if** LastFreq=0 **and** Buy_{t-2}=no **and** DaysSince<180	**then** d=Do mail
4 **if** LastFreq=0 **and** Buy_{t-2}=no **and** DaysSince≥180	**then** d=Do not mail
1 **if** Buy_{t-1}=yes	**then** d=Do mail
2 **if** Buy_{t-1}=no **and** Buy_{t-2}=yes	**then** d=Do mail
3 **if** Buy_{t-1}=no **and** Buy_{t-2}=no **and** DaysSince<180	**then** d=Do mail
4 **if** Buy_{t-1}=no **and** Buy_{t-2}=no **and** DaysSince≥180	**then** d=Do not mail
1 **if** LastProfit>0	**then** d=Do mail
2 **if** LastProfit=0 **and** Buy_{t-2}=yes	**then** d=Do mail
3 **if** LastProfit=0 **and** Buy_{t-2}=no **and** DaysSince<180	**then** d=Do mail
4 **if** LastProfit=0 **and** Buy_{t-2}=no **and** DaysSince≥180	**then** d=Do not mail

不难发现，这三个规则集的结构是非常类似的。这些决策规则很接近，这是因为 Last Freq 和 Buy_{t-1} 的相关系数为0.80，Last

Freq 和 Last Profit 的相关系数为 0.91，都很高。这些决策规则也反映了市场营销理论中关于最近购买时间(属性 Buy_{t-1}、Buy_{t-2})、购买频率(属性 Last Freq)和利润(属性 Last Profit)是预测未来消费行为的最好条件的理论(即使这三个属性高度相关)。同时还可以发现，只有最近的销售数据才对预测是有用的。

用测试数据集来测试这些规则集，得到表 10.7 所示的结果。可见，对于两种决策结果(Do mail 和 Do not mail)，其正确率大致都达到 75%，三个规则集的分类能力几乎相同。

表 10.7　分类结果

	Ruleset1	Ruleset2	Ruleset3
d=Do mail ∧ 客户实际不买	0.25	0.22	0.25
d=Do not mail ∧ 客户实际不买	0.75	0.78	0.75
d=Do mail ∧ 客户实际要买	0.74	0.72	0.74
d=Do not mail ∧ 客户实际要买	0.26	0.28	0.26

2. 不相等先验概率和错误分类代价比例(2∶1)。

实际上，错误分类的代价是有区别的。通常将一个要买的客户划分为不买的客户的代价大大高于将一个不买的客户划分为要买的客户的代价，因为后一种错误的代价仅仅限于邮资，而前一种错误的代价却是更为重要的销售利润。因此，我们将错误分类的代价按 2∶1 的比例进行分配，并将客户要买的先验概率调整为 0.4，将客户不买的先验概率调整为 0.6，重新用 Prob Rough 来处理。Prob Rough 的搜索过程由平均错误分类代价来控制。错误分类代价 2∶1 的比例部分地得到假定先验概率的补偿。

这种情况下得到的规则集合基本与表 10.6 相同，仅仅需要将 Days Since 的值改为 360。因此，可以认为错误分类代价的改变导致了需要考虑的时间周期的延长。

3. 不相等先验概率和错误分类代价比例(3∶1)。

现在将错误分类的代价比例进一步提高到 3∶1，得到的规则

集如表 10.8 所示。

表 10.8 在不相等的购买和不购买先验概率、不相等的错误分类代价(3∶1)前提下得到的决策规则集

1 **if** LastFreq>0 & State∈{4,5}	**then** d=Do mail
2 **if** LastFreq>0 & State∈{4,5} & Family>2	**then** d=Do not mail
3 **if** LastFreq>0 & State∈{4,5} & Family≤2	**then** d=Do mail
4 **if** LastFreq>0 & State∈{4,5} & Buy_{t-2}=Yes	**then** d=Do mail
5 **if** LastFreq=0 & State∈{4,5} & Buy_{t-2}=No & DaysSince≤365	**then** d=Do mail
6 **if** LastFreq=0 & State∈{4,5} & Buy_{t-2}=No & DaysSince>365	**then** d=Do not mail
7 **if** LastFreq=0 & State∈{4,5} & Buy_{t-2}=Yes & Family≤2	**then** d=Do mail
8 **if** LastFreq=0 & State∈{4,5} & Buy_{t-2}=Yes & Family>2	**then** d=Do not mail
9 **if** LastFreq=0 & State∈{4,5} & Buy_{t-2}=No & Family>2 & DaysSince≤365	**then** d=Do not mail
10 **if** LastFreq=0 & State∈{4,5} & Buy_{t-2}=No & Family≤2 & DaysSince≤365	**then** d=Do mail

用表 10.8 的规则来对测试集数据进行处理，得到的结果与表 10.7 相比，错误地将要买客户认为是不买客户的比例有所下降，但错误地将不买客户认为是要买客户的比例有所上升。这是直接由错误分类代价比例而导致的结果。

从市场的观点看，在对特定客户行为进行预测的时候，地理因素成了一个重要的考虑因素。邮购公司的实践也证实了这一点。而且家庭成员数也成了决策的一个重要因素。但是，这些带有非 RFM 变量的规则的强度并不太大。同样，需要考虑的时间周期仍

然是 1 年。

10.3.4　讨论

Prob Rough 得到的结果与商业理论一致，均认为 RFM 变量是影响决策的最重要的因素。数据挖掘的结果发现：只有最近时间内销售记录数据对于预测客户未来的购买行为有用。错误分类代价的变化导致了需要考虑的时间周期的延长。随着错误分类代价比例的提高，非 RFM 变量对于预测的重要性也随着增加。错误分类代价的不同考虑，可能在其他很多应用领域也需要注意。

10.4　信息过滤与信息检索

10.4.1　系统简介

随着电子信息量的急剧增长，信息过滤(information filtering，IF)和信息检索(information retrieval，IR)越发重要。文本分类是信息过滤的关键，它将文件在文本全集中进行归类。这样，用户就可以用不同的方式来处理不同类型的文件，集中精力处理有价值的东西。例如，一个电子邮件系统可以将收到的邮件分类为商业讯息、个人讯息和无用讯息，并自动删除无关邮件。

然而，对高维数据进行文本分类却是一个很困难的事情。在多数 IF/IR 技术中，每个文件都用维数特别高的向量来描述，典型的是每个单词或一组单词用一个值来描述。向量的坐标作为对文件进行分类的规则的前提条件。文件向量通常高达上万维，虽然它表达了所有的问题，但甚至处理能力最强的计算机也难于对之进行处理。而且，向量之间通过计算余弦来进行比较还将进一步增大文件分类的处理量。

英国爱丁堡(Edinburgh)大学的 Alexios Chouchoulas 教授等人基于 Rough 集理论对这个问题进行了研究，提出了文本分类的方法。给定文件全集和一个已经分类的文件集合，能够很快区别文

件类型的最小的并列关键词集合，这就大大降低了关键词空间的维数。这样小的关键词集合(前提条件)是能够为人所理解的，并且简化了基于知识的 IF/IR 系统的建立过程，规则库也可以编辑。

为了说明这个系统，我们首先来看一下文本分类问题。

10.4.2 文本分类

文本分类的目的是将文件全集空间进行划分。与其他分类任务类似，可以采用将新文件和以前已分类的文件进行比较(计算距离)的方法，也可以采用基于规则的方法。

绝大多数的文本分类技术没有在语意空间考虑。语意空间是一个超平面，其坐标轴代表不同关键词的存在。采用特定的技术，这个关键词空间的坐标轴可能是离散值(如布尔值)或者连续值(如关键词的频率、关键词的重要性等)。关键词空间的维数取决于关键词全集的基数，而关键词全集定义为被考察的所有样本中的所有可能关键词的并集。全集中的任意文件 D_j 就可以表示为如下的多维关键词向量 x_{D_j}：

$$U=\bigcup_i D_i=\{k_1,k_2,\cdots,k_n\},$$

$$x_{D_j}=(f(D_j,k_1),f(D_j,k_2),\cdots,f(D_j,k_n)),$$

其中，坐标 $f(D_j,k_i)$ $(1\leqslant i\leqslant n)$ 表示空间 U 上关键词 k_i 的权值，它是关键词 k_i 在文件中是否出现、出现的频率或者其他重要性度量等的综合。

下面先对现有的文本分类技术作一个简要的介绍。

首先是基于距离的文本分类技术。这种技术基于高维关键词向量的比较。一个向量描述了一组文件，可把它作为一个文件簇的焦点(中心)。通过比较向量来对文件进行分类。向量比较通常是计算两个向量之间夹角的余弦值，也可以采用其他的方法。

表示一个文件(或一簇文件)的关键词集合可以通过扫描很多不同的关键词及文件全集，并根据重要性进行排列来得到。

这类系统的数据集通常是自动建立和维护的，采用的方法有

观察学习、样例学习、模仿学习等，以减少权值和向量格式的实际计算。这使得用户花很少的力气就可以实现复杂的智能文本分类。然而不幸的是，文件向量的维数太高（通常上万维），分类极慢，而且文件向量的存储也是很昂贵的。

其次是基于规则的文本分类技术。这是一个长期以来得到了很多应用的文本分类技术。如 Usenet 客户软件所用的文章过滤器 kill - file 和 van den berg 的自动电子邮件过滤器 Procmail 均采用了这种技术。

文件中，关键词向量被作为规则前提条件，文件所属的类被用作规则决策：

$$r_i(D) = p(D,k_1) \wedge p(D,k_2) \wedge \cdots \wedge p(D,k_n) \Rightarrow D \in [D],$$
$$k_1, k_2, \cdots, k_n \in U,$$

其中：k_i 是关键词，U 是关键词全集，D 是一个文件，$[D]$ 是一个文件类，$r_i(D)$ 是应用于 D 的第 i 条规则，$p(D,k_i)$ 是一个函数。如果 D 包含关键词 k_i 且满足一定的度量（如最小频率或权值），则 $p(D,k_i)$ 取值为 True。规则不检查全集 U 中所有的关键词，这就使得基于规则的文本分类器不需要像基于向量（距离）的分类器那样做到每个向量的维数必须等于关键词空间的维数。

在多数系统中，规则是人为设置的。很多典型的规则库只是简单地测试特定关键词是否在文件中出现，即 $p(D,k_i)$ 等价于 $k_i \in D$。例如，Usenet 客户可以用 kill - file 通过检查邮件“From”域中的人名实现对邮件的过滤。这种基于规则的系统易于理解。然而，复杂的应用需要复杂的规则库，这时用户就不可能手工维护了。

10.4.3　基于 Rough 集的文本分类系统

很多电子邮件用户将相关的邮件按相同的方式保存在文件夹中。这就给系统提供了训练数据。与很多文件相似，邮件是结构化的文件，一个邮件由一个包含很多域的邮件头和邮件体组成。根据关键词在邮件中出现的位置，需要对之进行不同的处理。例如，邮

件标识号(Message ID)是邮件的唯一标识,而这类关键词却往往导致过度学习。

这个文本分类系统的结构如图 10.1 所示,包括两个阶段。关键词获取阶段读取所有的文件(相近似邮件的文件夹),确定候选关键词,估算其重要性并建立一个高维数据集。Rough 集属性约简(RSAR)阶段测试数据集并去掉冗余。这样,就得到由包含少量前提条件的规则组成的规则库。

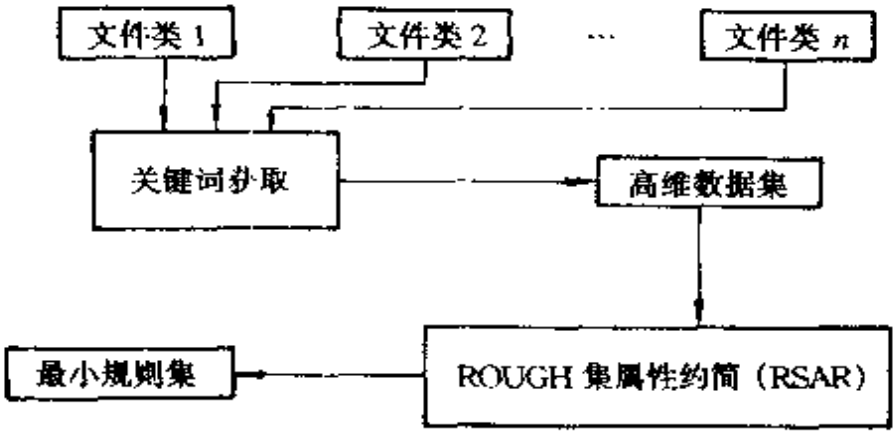


图 10.1 系统数据流图

关键词获取子系统的输入是邮件文件夹集合,并假定每个文件夹包含的邮件是相似的。文件夹中的邮件是 RFC-822 标准格式的邮件。邮件被顺序地读入,每个邮件中的每个域(关键词/值对)单独处理。邮件体被作为一个非常长的域。

在每个域中,单词被分离出来并预过滤,以避免非常短和非常长的关键词,以及不是单词的关键词(如长数字、随机字符串等)。文本中的每个单词或短语被作为候选关键词,当前域名也作为关键词,这样就可以体现关键词在邮件中出现的位置。

这个阶段,有两种方法获取关键词:一是单簇绰号(dubbed single cluster)方法,生成一个关键词集合来对各个文件夹进行描述;二是为各个文件夹中的每个邮件生成单独的关键词集合,最终为每个邮件生成一条分类规则,因此,这也被称为单邮件绰号

(dubbed one-per-message)方法。前一种情况，关键词的权值按如下公式计算，多数邮件都共有的关键词的重要性肯定更高：

$$w_1(k) = -\log\left(\frac{N_k}{N}\right) f_k w_f;$$

后一种情况，关键词的权值度量更着重于那些区分不同邮件的关键词，这使得关键词集合多样化，而不是对各个邮件生成相同关键词集合的多个拷贝，其计算公式为

$$w_2(k) = -\log\left(\frac{N}{N_k}\right) f_k w_f;$$

其中 $w_1(k)$ 和 $w_2(k)$ 分别是关键词 k 在这两种情况下的权值度量。N_k 为文件夹中包含关键词 k 的邮件数量；N 是文件夹中的邮件总数；f_k 是关键词 k 在当前邮件中出现的频率；w_f 是当前域对分类的重要性，其取值依赖于具体的应用和用户的偏爱选择。为避免过度训练和其他缺陷，一些特定域的重要性设置得相对很小，这也同样影响该域中关键词的权值。例如，我们可以合理地假设邮件的主题(subject)和邮件体(body)包含了非常重要的信息，而邮件的转发路径(received)不会提供有用的关键词。

最后，一个关键词在被加入到关键词集合之前，还需要经过两个过滤器处理：一个是低通过滤器，将那些很不常用的、肯定不是好关键词的滤掉；另一个是高通过滤器，将诸如辅助动词的那些出现频率很高的常用词滤掉。这使得本关键词算法能够独立于语言：多数类似的方法依赖于英语辞典和公共英语单词列表来实现这个功能。最后，在关键词输出之前，还需要对所有权值进行规范化处理。这就使得下一阶段对权值的处理更均匀，以避免产生与直观感觉相违背的结果。

RSAR 接受前面关键词获取算法得到的关键词集合。这是通过评价所有加权关键词集合的并集而得到的数据集合，关键词按权值递减的顺序排列。如果一个关键词具有两个以上不同的权值，取其最大的一个。每个关键词映射到数据集中的一个条件属性。决

策属性是产生关键词的文件夹名。与数据集中的遗失值相应的关键词在特定关键词集合中没有出现。

举个例子来说，下面两个关键词集分别描述了两个邮件，可以用他们来建立表 10.9 所示的数据集：

$D_1 = \{(k_1, 0.19), (k_2, 0.98), (k_3, 0.72), (k_4, 0.87)\}$，

$D_2 = \{(k_3, 0.31), (k_5, 0.42), (k_6, 0.56)\}$。

表 10.9 数据集

	k_2	k_4	k_3	k_6	k_5	k_1	→	Class
D_1	0.98	0.87	0.72			0.19	→	α
D_2		0.31		0.56	0.12		→	β

注：其中，α 和 β 分别是两个邮件所在的文件夹。

由于 RSAR 更适用于处理名词性数据集，因此，数据集需要量化。有两种可用的不同量化方法：一是布尔量化(Boolean, binary)，1 表示关键词的出现，0 表示关键词不出现；二是将规范化的权值空间取整量化为 11 个值(weighted)，表示为 ent(10w)(其中：w 是关键词权值，ent(10w)表示小于等于 10w 的最大整数)。这两种量化方法可以为不同的分类器提供更好的接口，也可以为应用领域中的权值量化提供最佳技术。

在量化完这个中间的高维数据集之后，RSAR 就可以执行快速约简算法(QUICKREDUC)删除所有的冗余属性，生成一个维数得到根本减少的数据集。数据集中的每个对象(个体，元素、样例)包含一组条件属性和一个决策属性，可以视之为产生式规则，条件属性值作为规则的前提条件，而文件的分类作为规则的结论。多余的重复规则被进一步删除，最后输出规则库。

10.4.4 实验结果

在实验中，采用了 7 个不同的电子邮件集合。邮件集合的选取尽量使得文件集合中的特征更广；有的是相似的，是由同一个人写

的邮件；有的包含了不同人用不同书写方式写的文本。大大小小的邮件集合混在一起，以保证文件数量的多少不影响所得的规则库的质量。一个邮件集合代表一类文件。

随机地选用 2 到 5 个邮件文件夹作为输入，测试关键词生成方法（单聚簇和单邮件）与量化方法（布尔和取整）的所有组合。表 10.10 给出了每种组合情况下得到的平均维数约简结果。维数的约简量是以 10 的幂表示的。需要注意，虽然单邮件数据集生成技术的约简效果更好，但其维数却比单聚簇方法高得多。单邮件方法在约简前的平均维数是 26 827，而单聚簇方法是 338。结果规则库中的规则有 1～6 个前提条件。由于布尔值的信息量少，所以布尔规则库的维数要稍微高一些。

表 10.10　四种组合操作模式的平均维数约简量

	单聚簇	单文件
布尔	2.02	3.62
取整	2.29	3.70

图 10.2 给出了所得规则库的平均分类能力（discernibility）。分类能力为 1 表示在执行 RSAR 算法后没有信息丢失。使用规则库的分类器假定训练集是一个好的统计样例，得到的分类精度很接近于规则库的分类能力。如图所示，为一个电子邮件文件夹只生成一条规则是得不到好的分类效果的。由于文本内容的变化，分类能力会降至不可接受的程度，而且有很大的变化。相反，生成多条规则就可以更好地覆盖特征空间，分类能力可以高到满意的程度，并基本稳定。布尔量化的结果没有取整量化的结果那么满意。

从规则库语意自然的观点看，人类更易于阅读和理解覆盖整个空间的规则，难于理解那些描述单个文件的规则，也可以考虑简单地通过阅读的直觉方式来评判规则库的质量。

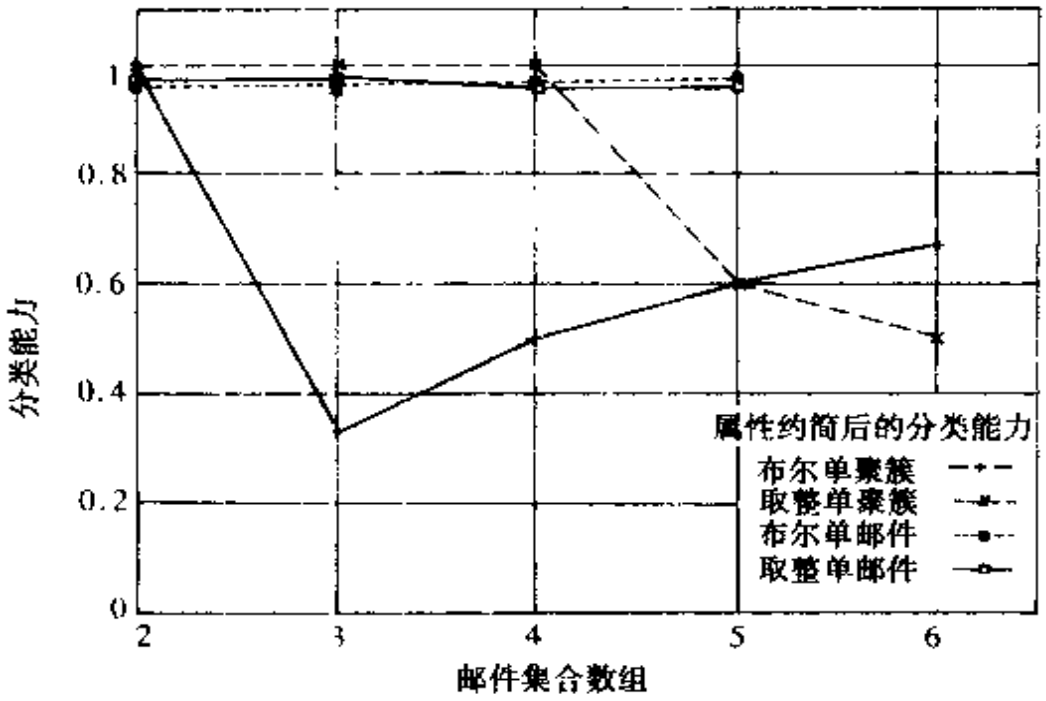


图 10.2 维数约简之后的分类能力

10.4.5 讨论

文本分类很大程度上依赖于获取描述文件的关键词集合。可以利用 Rough 集方法来降低关键词集合的维数，而又保持关键词集合中的信息。这个技术是很有效的，而且是与具体语言独立的，其模块结构也使得系统很灵活。这个系统可以与很多文本分类器接口，优化分类规则库，允许用户直接阅读和理解规则库。

这里的单聚簇规则生成方法得到的结果还是用户不能接受的。还需要寻求更合适的关键词重要性度量标准，进一步优化分类规则，为系统达到实用程度做更多的实际测试工作。

10.5 电信信道噪音抑制

Rough 集理论在电信通讯行业中也有广泛的应用，例如噪音抑制、信道调度、设备故障诊断、顾客行为分析等。波兰 Rafal Krolikowski 和 Andrzej Czyzewski 对影响电信信道语音信号传输

的非固定噪音抑制问题进行了研究，并综合考虑了人类听觉系统特征以及基于 Rough 集的推理和神经处理等因素。

10.5.1　概述

通常的噪音抑制方法不考虑人类听觉系统的一些主观特征。而通过实验发现，还可以利用人类听觉特征来抑制噪音。所有的噪音抑制方法都至少需要知道噪音的近似统计特征。在非固定噪音情况下，这个问题就变得很复杂，导致难于得到有效的决策系统。因此，可以考虑采用诸如 Rough 集、神经网络等智能算法来构造这些应用领域的决策系统。

10.5.2　生理学原理

临界波段与人类听觉系统中声学信号的传输和处理有关。内耳的功能像许多带通滤波器，独立分析一定带宽区域的子波。这些子波就是临界波段，并由此引入一个频率的知觉单位——Bark，它与单个子波的带宽有关。通常使用 Zwicker 提出的下述变换公式来作为听觉的这个主观量度：

$$b=13\times\arctan(0.76\times10^{-3}\times f)+3.5\times\arctan\left[\left(\frac{f}{7500}\right)^2\right],$$

其中，b 和 f 分别为用 Bark 和 Hz 为单位表示的频率。

另一个生理学现象与掩盖有关。一些音调在其他音调出现的时候听不见，特别是当其中某个音调音量比较大而且它们之间的频率差别不是很大的时候。掩盖其他音调的音调称为掩盖音。这个现象是现代语音编码标准的基础。

10.5.3　知觉噪音抑制系统的描述

知觉噪音抑制系统如图 10.3 所示，有两个输入：噪音信号 $y(m)$ 和噪音模型 $\tilde{n}(m)$。信号 $y(m)$ 由受噪音 $n(m)$ 破坏的原始语音信号 $x(m)$ 组成，经 DFT 变换为频谱表示的 $y(j\omega)$。同样，假设噪音模型 $\tilde{n}(m)$ 与噪音 $n(m)$ 相关，是从电信信道中传输的空信号包得到的。噪音估计模块从输入的信号 $\tilde{n}(m)$ 中提取噪音 $n(m)$ 的

基本信息，其输出为时频噪音估计 $\rho(t,j\omega)$。时频噪音估计 $\rho(t,j\omega)$和被破坏的语音 $y(j\omega)$都被输入到决策系统。决策系统的首要任务是选择给定时刻与破坏噪音相关性最好的 $\rho(j\omega)\subset\rho(t,j\omega)$，其次是为两个分离集合(有用元素集合 U 和无用元素集合 D)限定信号 $y(j\omega)$的元素，有必要知道哪些波谱成分是掩盖音(有用元素)，哪些是被掩盖音(无用元素)。

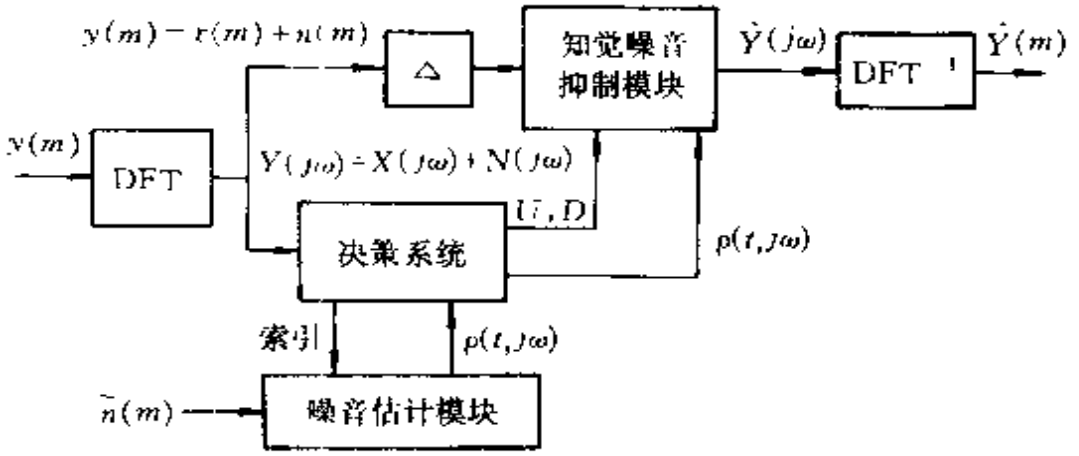


图 10.3 噪音抑制系统的一般框图

被破坏信号的波谱 $y(j\omega)$以及集合 U，D 和选定的噪音估计 $\rho(t,j\omega)$一起被输入到知觉噪音抑制模块，在这里经过噪音抑制知觉算法的处理，然后，输出 $\hat{Y}(j\omega)$经过逆 DFT 变换，最后得到比原始信号噪音少的还原信号 $\hat{Y}(m)$。

10.5.4 噪音抑制系统的实现

1. 噪音估计模块

噪音估计模块工作于噪音分析模式和噪音抑制模式这两种模式。在噪音分析模式(如图10.4的(a)所示)下，噪音模型 $\tilde{n}(m)$经噪音参数抽取模块分析，转换到光谱域并根据后续 L 帧取平均和分析，得到两种输出信号：平均功率光谱 $\hat{N}_k$ 和与光谱 $\hat{N}_k$ 相关的关联系数向量 V_k^n。索引 k 表示计算这些向量元素的时间间隔。这两种向量随后就进入向量表。向量表中的内容将在决策系统决策算

法训练过程和噪音抑制模式(如图10.4的(b)所示)中用到。在噪音抑制模式下,根据对向量表的查询输出可能与当前破坏语音信号的噪音最相关的合适的光谱 $\hat{N}_i$。查询索引值是由决策系统得到的,代表向量表中最想得到的光谱的索引。

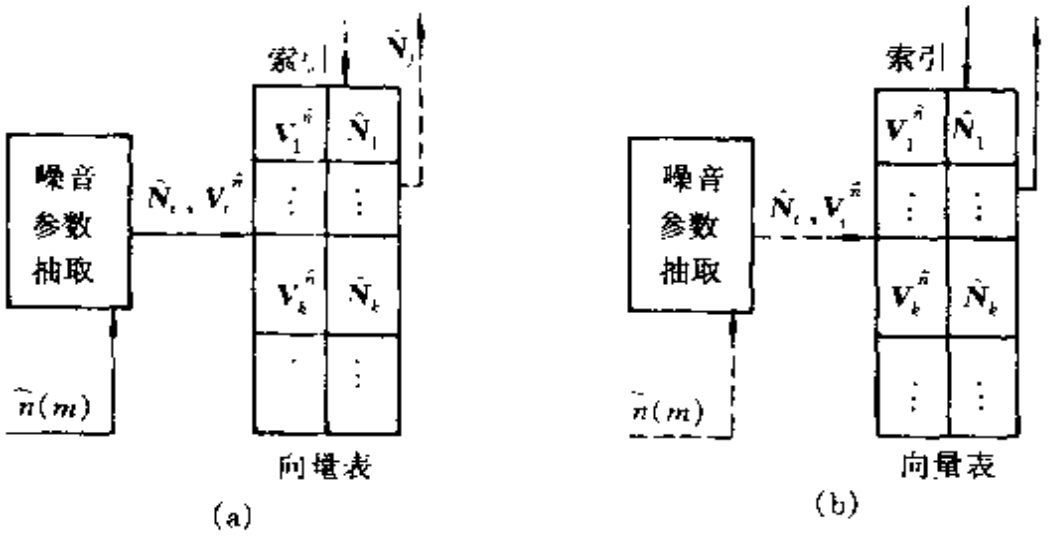


图 10.4　噪音估计模块图示

(a) 噪音分析模式；　(b) 噪音抑制模式

如果采用 N 点 DFT 变换,则向量 $\hat{N}_k$ 的定义为

$$\hat{N}_k=(\hat{N}_{1,k}\cdots\ \hat{N}_{n,k}\cdots\ \hat{N}_{N/2,k})^{\mathrm{T}},\text{且 }\hat{N}_{n,k}=\frac{1}{L}\times\sum_{l=(k-1)\times L+1}^{k\times L}\tilde{N}_n^{(l)},$$

其中 $\hat{N}_{n,k}$基于光谱功率 $\tilde{N}_n$ 的最后 L 个值取平均。

关联向量 V_k^n 作为关键向量在噪音抑制模式下当检索到光谱 $\hat{N}_i$ 时被使用。这个向量应该是唯一的,但实际上难于保证这个条件。期望其元素值定量反映平均光谱 $\hat{N}_k$ 的噪音参数。因此,要考虑光谱平面度量(SFM)和不可预知性度量(UM)这两个现代知觉编码方案中非常鲁棒的参数。这些参数在各个临界波段中计算,并进一步定义它们在第 l 帧的定义。下面来考虑如何应用光谱平面度量和不可预知性度量这两个参数。

(1) 光谱平面度量的应用。

SFM 参数定义为功率光谱几何平均数 G_m 在算术平均数 A_m 中的比例，单位是 dB。在第 b 子波段，这个参数又可以重新定义如下：

$$SFM_b^{(l)}=10\times\log_{10}G_m^{(l)}/A_m^{(l)}。$$

因此，向量 V_k^n 可以描述为

$$V_k^n=(SFM_{1,k}\ \cdots\ SFM_{b,k}\ \cdots\ SFM_{B,k})^{\mathrm{T}},$$

且

$$SFM_{b,k}=\frac{1}{L}\times\sum_{l=(k-1)\times L+1}^{k\times L}SFM_b^{(l)}。$$

(2) 不可预知性度量的应用。

基于其最后两个真实值，对第 i 个光谱成分的幅度预测 $\hat{r}_i^{(l)}$ 和相位预测 $\hat{\varphi}_i^{(l)}$ 为

$$\hat{r}_i^{(l)}=r_i^{(l-1)}+(r_i^{(l-1)}-r_i^{(l-2)})$$

和

$$\hat{\varphi}_i^{(l)}=\varphi_i^{(l-1)}+(\varphi_i^{(l-1)}-\varphi_i^{(l-2)})。$$

不可预知性度量 $c_i^{(l)}$ 定义为真实值 $(r_i^{(l)},\varphi_i^{(l)})$ 和预测值 $(\hat{r}_i^{(l)},\hat{\varphi}_i^{(l)})$ 的 Euclidean 距离：

$$c_i^{(l)}=\frac{\mathrm{dist}((\hat{r}_i^{(l)},\hat{\varphi}_i^{(l)}),(r_i^{(l)},\varphi_i^{(l)}))}{r_i^{(l)}+\mathrm{abs}(\hat{r}_i^{(l)})}。$$

在这种情况下，向量 $V_k^n=(C_{1,k}\ \cdots\ C_{b,k}\ \cdots\ C_{B,k})^{\mathrm{T}}$ 的第 k 个元素基于第 b 临界波段最后 L 帧取平均：

$$C_{b,k}=\frac{1}{L}\times\sum_{l=(k-1)\times L+1}^{k\times L}C_b^{(l)},$$

其中 $C_b^{(l)}=\frac{1}{\mathrm{count}(b)}\times\sum_{i=\mathrm{lower}(b)}^{\mathrm{upper}(b)}c_i^{(l)}$，$\mathrm{upper}(b)$ 和 $\mathrm{lower}(b)$ 是包含 $\mathrm{count}(b)$ 的第 b 子波段中的第一光谱和最末光谱的索引。

2. 决策系统

决策系统模块(如图 10.5)的输入是用光谱表示的噪音信号 $Y(j\omega)$。首先，输入信号由参数抽取模块处理得到参数向量 V_i^y，期望这些参数与输入 $Y(j\omega)$ 的噪音特征最相关。因此，V_i^y 的元素通过模仿关键向量 V_k^n 的元素来定义，可以用前面的公式来计算。向

量 V_t^v 随后输入决策系统 I，得到与噪音信号 $Y(j\omega)$ 中出现的噪音最相关的噪音光谱 $\hat{N}_i$ 的索引值。从向量表中取得想要的向量，并在决策系统 II 中与光谱表示的 $Y(j\omega)$ 进行比较，得到两个输出集合：有用元素集合 U 和无用元素集合 D。

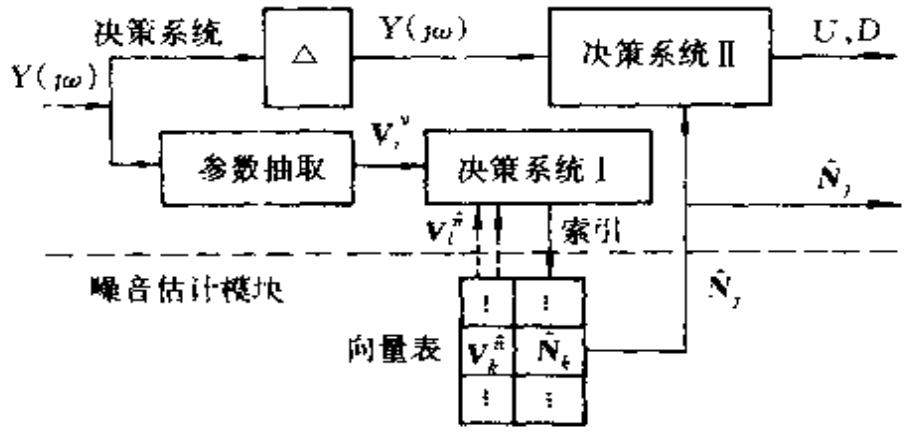


图 10.5　决策模块图示

下面分别讨论决策系统 I 和决策系统 II 的实现。

1. 决策系统 I 的实现

决策系统 I 的决策制定可以基于 Rough 集和神经推理，系统的运行可以分为 2 种模式：训练模式和执行模式。在训练模式，采用向量表中的内容，如图 10.5 中的虚线。下面讨论智能决策算法。

• 应用 Rough 集

在训练模式，向量表中的一部分被视为决策表，关键向量 V_k^n 的元素作为条件属性，向量表中向量的索引作为决策属性。这样，表中第 k 个对象可以用如下的关系来描述：

$$SFM_{1,k},\cdots,SFM_{b,k},\cdots,SFM_{B,k}\Rightarrow k$$

或者

$$C_{1,k},\cdots,C_{b,k},\cdots,C_{B,k}\Rightarrow k。$$

根据 Rough 集理论和相应的约简算法可以得到规则。这里只有条件属性需要离散化处理。在执行模式，输入噪音语音参数 V_t^v 向量被离散化并由训练得到的规则集处理，得到向量表中噪音光谱 $\hat{N}_i$ 的索引值。

● 应用神经网络

这里采用的神经网络是具有隐神经元的经典的前馈结构神经网络。在初步实验中，只考虑了一个隐层的情况，输入节点的数目等于向量 V_k^n 的元素数目。只有一个输出层神经元，其输出为向量表中的索引值。所有输入神经元节点和隐层神经元节点都采用 Sigmoid 活动函数。然而，由于索引值是整数，输出层神经元节点的活动函数可以是 Sigmoid 函数或者线性函数。

在训练模式，向量表中的一部分作为训练集，关键向量 V_k^n 是输入向量，其在表中的索引作为目标输出。因此，训练集如下：

$$\{(V_1^n,1),\cdots,(V_k^n,k),\cdots,(V_K^n,K)\}。$$

学习算法采用标准的误差反向传播算法，误差度量采用均方差。如果输出层神经元的活动函数为 Sigmoid 函数，目标索引值就需要按比例变换到区间[0,1]。

在执行模式，网络的输出(索引值)必须取整为最相近的整数。而且，如果输出层神经元的活动函数是 Sigmoid 函数，在取整之前还需要按同样比例进行逆变换。

2. 决策系统Ⅱ的实现

在决策系统Ⅱ中，将根据下面的简单过程对有用元素和无用元素进行划分。光谱功率 Y 超过典型噪音估计 $\hat{N}_j$ 平均值两倍的所有元素被视为有用元素，其余的元素被视为无用元素。因此，在采用 N 点 DFT 变换的情况下，集合 U,D 可以定义如下：

$$U=\{n,Y_n:Y_n\geqslant 2\times\hat{N}_{n,j}\text{ and }n=1,\cdots,N/2\},$$

$$D=\{n,Y_n:Y_n<2\times\hat{N}_{n,j}\text{ and }n=1,\cdots,N/2\}。$$

通常，也可以采用其他的划分方法，但好的选择对于还原的语音的主观音质具有很重要的影响。

3. 知觉噪音抑制模块

本模块的功能是对以光谱表示的噪音信号进行下面的处理：所有的有用光谱成分按照光谱减少原理减少，剩下的无用成分采用心理学方法进行掩盖。

10.5.5　仿真实验

实验的目的有两个：验证非固定噪音抑制方法和对不同的决策算法进行比较。为了检查智能技术是否能够提高被还原语音信号的音质，首先进行验证测试。验证测试的结果对于后面的对比实验很理想。

为了评价决策算法的质量，就需要将算法得到的结果和理想结果进行比较。在研究中，有必要知道由决策系统指定的噪音光谱是否是最好的选择，如果不是，就需要对误差进行度量。

为了进行对比实验，做了两个录音：一个男子的语音(5.81 s)和从收音机信道中采样的非固定噪音(2.79 s)。然后，在原始语音中加入噪音，同时计算和采集噪音的关键向量元素。由于已知原始语音和噪音，就知道于哪部分噪音语音由哪个关键向量和噪音光谱向量描述。这些录音都是单声道，16 位编码，采样频率 8 192 Hz，有 $B=18$ 个临界波段。在语音处理过程中，信号按帧划分并有重叠。由于采用了 Hamming 窗口函数，重叠区域为帧长 N 的一半。实验中用到 3 种帧长 N：128，256 和 512，它们对时间分辨率和频率分辨率的影响如表 10.11 所示。表 10.11 中的噪音关键向量是在假定信号由最后 $L=4$ 帧取平均的条件下得到的。注意，这些向量的数目也是决策表中个体的数目或者是训练向量的数目。

表 10.11　帧长 N 对时间分辨率和频率分辨率的影响以及训练集

N	时间分辨率	频率分辨率	向量数目
128	7.83 ms	64 Hz	88
256	15.63 ms	32 Hz	44
512	31.25 ms	16 Hz	22

依据下面的变量来分别进行对比测试：

- 不同的帧长：$N=128$，$N=256$，$N=512$；
- 不同的关键向量类型：基于 SFM 参数，基于不可预知性度

量参数：

- 不同的离散化(量化)步长：0.1、0.5、1；
- 不同的隐层神经元数目：10、15、20；
- 不同的输出层神经元活动函数：Sigmoid 函数，线性函数。

因此，单一测试可以用对于给定决策算法有效的一个参数集合来描述。从而，可以采用下述表示：(N，向量，RS，离散化步长)用于 Rough 集方法，(N，向量，NN，隐层神经元数目，输出节点类型)用于神经网络方法。总共进行了 32 个测试，如(512，SFM，NN，10，Sigmoid)，(512，C，NN，10，Sigmoid)，(512，SFM，RS，0.5)。

为了评价决策系统的性能，引入误差度量 E。它是噪音信号的所有 I 帧产生的误差 $E^{(i)}$ 的平均值，根据如下表达式来定义：

$$E=\frac{1}{I}\times\sum_{i=1}^{I}E^{(i)},\qquad E^{(i)}=\sum_{b=1}^{B}(V_{b,i}^{y}-V_{b,index}^{n})^2,$$

其中，$V_{b,i}^{y}$ 与噪音语音参数向量相关，$V_{b,index}^{n}$ 是位于向量表 $index$ 位置的关键向量的第 b 个元素。

通过模拟，可以得到最佳误差度量 E_{opt} 和最大误差 E_{max}。假定语音的第 i 帧被由向量表中第 j 个向量描述的噪音破坏，则这些度量可以定义如下：

$$E_{opt}=\frac{1}{I}\times\sum_{i=1}^{I}E_{opt}^{(i)},$$

$$E_{opt}^{(i)}=\sum_{b=1}^{B}(V_{b,i}^{y}-V_{b,j}^{n})^2,$$

$$E_{max}=\frac{1}{I}\times\sum_{i=1}^{I}E_{opt}^{(i)},$$

$$E_{max}^{(i)}=\max_{k=1,\cdots,K}\left[\sum_{b=1}^{B}(V_{b,i}^{y}-V_{b,k}^{n})^2\right],$$

其中 K 是向量表中的向量数目。

最后，质量系数 q 定义为：

$$q=1-\frac{E-E_{opt}}{E_{max}-E_{opt}}。$$

下面是得到的几个实验结果：

- (512,*SFM*,*NN*,10,Sigmoid)：　$q=81.38\%$；
- (512,*C*,*NN*,10,Sigmoid)：　$q=85.21\%$；
- (512,*SFM*,*RS*,0.5)：　$q=78.43\%$。

可以这样推测，测试(512,*C*,*NN*,10,Sigmoid)得到的最好结果是因为采用了更为精确的关键向量类型(基于不可预知性度量参数)。Rough 集方法得到的最差结果可能主要是由于采用了不好的离散化步长。

10.5.6　讨论

本节介绍了一个用于非固定噪音抑制的工程系统，采用了基于神经处理和 Rough 集的智能推理。为评价特定决策系统的性能而进行了一系列实验，实验结果说明计算智能和软计算方法可以用于对噪音抑制的知觉编码算法进行控制。

第 11 章　Rough 集理论的实验系统

为了便于 Rough 集理论学习者和研究人员的参考，本章将对国际上已经研制出来的一些 Rough 集工具软件进行介绍，如 Rough Enough，Rose，Rosetta，KDD-R，LERS 等。其他的一些系统，如 Rough Set Library，Grobian，Datalogics，K－Days，Rough Analysis 等，限于篇幅就不再具体介绍了，有兴趣的读者可以到 Electronic Bulletin of the Rough Set Community 的站点(http://www.cs.uregina.ca/～roughset)和相关的网络站点去查询。

11.1　Rough Enough

Rough Enough 是由挪威 Troll Data Inc. 在 4GL DBMS Paradox for Windows 下开发的一个基于 Rough 集理论的数据挖掘工具系统，目前已经发展到 4.0 版，可以在 http://www.trolldata.no/renough 下载该软件。

数据挖掘的流程图如图 11-1 所示。

下面按这个流程图对 Rough Enough 系统进行介绍。

Rough Enough 系统不包括最前面的数据获取和初始预处理两个步骤。

输入到 Rough Enough 中，支持多数 PC 数据库和电子数据表格式，RSES(Rough Set Expert System)和 Rough Set Library 2.0 版的格式也支持。只需转换数据库驱动程序就可以方便地访问 SQL 服务器。

预处理阶段可以选择如下的方法：

- 绝对改变：按照样本的编号计算当前样本和前一个样本之

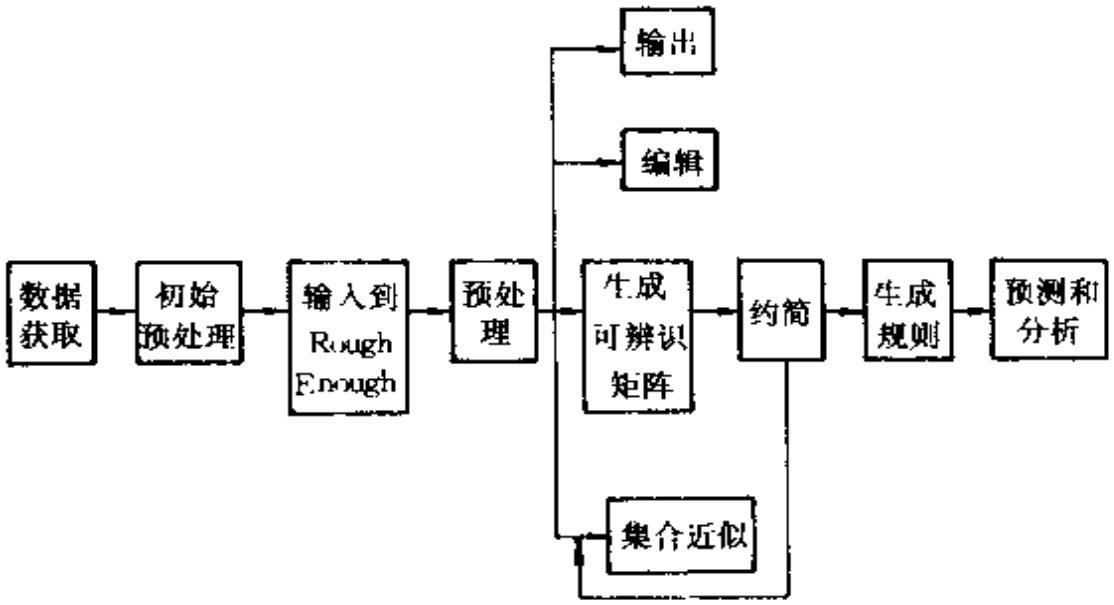


图 11-1　数据挖掘的流程图

间的差值。

• 百分比改变：以百分比变化的方式计算当前样本和前一个样本之间的差值。

• 离散化。

• 划分表格：根据用户对于保留样本量的选择，将表格划分为两个，划分掉的样本另外存储，用作测试规则的样本。

• TIS 和 IS：将一个时序信息系统转换成为一个信息系统。

除此之外，用户还可以直接访问 SQL 和 QBE，这能够让用户实现一些特殊的功能。

编辑阶段中，用户可以在电子表格中编辑数据。

生成可辨识矩阵阶段，根据信息系统计算得到可辨识矩阵。

集合近似阶段，系统有很多工具：等价类、决策类、下近似、上近似、边界域、Rough 成员值和一般化决策规则。这些计算是依赖于用户所选择的属性的。

约简阶段，用遗传算法生成约简结果。

生成规则、预测和分析阶段：得到决策规则并对测试样本集进

行测试。这个阶段，根据分析测试的结果，也许还需要返回前面的阶段进行循环，以得到满意的结果。

11.2 ROSE

ROSE(Rough set data explorer)系统是由波兰 Poznan 工业大学计算科学研究所智能决策支持系统实验室开发的一个模块化软件系统，它实现了 Rough 集理论的基本理论和规则获取技术。这个系统实现了 Pawlak 的基本 Rough 集模型和 Ziarko 的可变精度 Rough 集模型，ROSE 系统是 Rough DAS & Rough Class 系统的新版。

ROSE 是由几个独立模块集成而成的。首先，在高级的计算机(如 UNIX 工作站)上建立计算引擎，这有利于对大数据集进行快速分析；然后，在 Windows 平台上开发友好的用户界面。这些模块可以独立进行重新设计和开发，然后重新编译。其中严重依赖于平台的部件是图形用户界面(GUI)。所有这些都保证了系统易于适应将来的操作系统和平台。

ROSE 是运行在 PC 兼容机 32 位图形用户界面操作系统 Windows95/NT 4.0 上的交互式软件系统。其核心模块用标准 ANSI C++程序设计语言编制，接口界面模块用 Borland C++(带对象窗口库)和 Borland Delphi 编制。

系统包括一个图形用户界面和一系列单独的计算模块。这些模块是与平台独立的，可以被包括 UNIX 机在内的不同目标机重新编译。图形用户界面覆盖了所有的计算模块。因此，系统具有很好的可扩充能力。

ROSE 的操作简单，鼠标操作，菜单驱动，用户界面友好，专家和普通用户均可以用它来进行数据分析。系统通过对话窗和用户交互，所有的结果按环境表示，数据可以用电子数据表编辑。

ROSE 接受信息表数据输入。属性分为条件属性和决策属性。

数据按照一定的文件格式(information system file,ISF)存储在一个纯文本文件中。ROSE 还可以接受 Rough DAS 的数据,并按照几种其他格式输出,如 LERS 和 C4.5 的格式。

ISF 文件格式允许最多 30 个字符的长属性名和字符串、实数、整数值。由于它是纯文本文件,因而可以不作任何变化在不同操作系统之间传递,也易于编辑数据和校验文件中数据的正确性。

ISF 文件有开放的文件格式,被分为节,可以按照将来应用的需要增加尚未定义的新节。用户仅通过改变属性的限制就可以决定忽略掉一些属性。

除了可视化图形用户界面,所有的结果也写入纯文本文件,即使在 ROSE 系统外也可以读取这些文件,而且易于转换为其他文件格式。

现在,ROSE 的计算模块具有如下特征:

- 数据校验和预处理;
- 采用 Fayyad & Irani 离散化算法和用户自己离散化(user-driven discretization)对连续值属性进行自动离散化处理;
- 用标准 Rough 集模型或可变精度 Rough 集模型对条件属性近似目标分类的能力进行定性估计;
- 用多种算法(S. Romanski 和 A. Skowron 等人的算法)发现属性核以及信息表的约简(所有约简,或者一组预定规模大小的约简);
- 通过观察分类质量的变化,考察一个给定属性对于目标分类的相对重要性;
- 为目标分类选择最重要的属性,删去冗余属性(实现了几种能够保证目标分类质量的属性子集选择技术);
- 用 LEM2 算法或 Explore 算法获取决策规则;
- 获取规则的后处理(剪除规则、根据用户的要求选择感兴趣的规则);
- 基于不同的规则匹配技术,用决策规则对新目标进行分

类；

- 用 K 叠交叉验证方法对决策规则集进行评价。

还可以很容易地增加新的模块到这个系统中。

ROSE 系统实现的功能还是有限的，有待于进一步扩充，其开放的系统结构也为此奠定了基础。ROSE 系统已经成功地应用于处理很多实际数据集，如医学、药剂学、技术诊断、金融和管理科学、图像与信号处理、地质、软件工程评估等。

11.3 Rosetta

Rosetta 是由挪威科技大学计算机与信息科学系和波兰华沙大学数学研究所合作开发的一个基于 Rough 集理论框架的表格逻辑数据分析工具包，包括了计算核和图形用户界面，能够在微机的 Windows NT/98/95 操作系统上运行。Rosetta 的设计实现了对数据挖掘和知识获取的支持从数据的初始浏览和预处理，计算最小属性约简和产生 if - then 决策规则或描述模式，到对所得到的规则或模式的验证和分析。Rosetta 的目的是要作为基于不可分辨关系模型的通用工具，而不是为某个特定的应用领域设计的专用系统。

Rosetta 提供了一个很直观的图形用户接口，采用了数据导航的技术，图 11-2 给出了 Rosetta 的界面结构示意图。图形用户界面是高度面向对象的，所有的操作对象被表示为独立的图形用户界面的元素项，每个元素项有自己的与上下文相关的菜单集合。

Rosetta 的计算核心也可以采用命令行程序。计算核心提供了如下的功能：

- 输入/输出
  - ◆ 通过 ODBC 和 DBMSs 部分集成。
  - ◆ 输出格式包括规则、约简、表格、图像以及 C++ 和 Prolog 等格式。

- 预处理
  - ◆ 不完备数据表的完备化处理(数据补齐)。
  - ◆ 连续属性值的离散化。
- 计算
  - ◆ 支持有教师学习和无教师学习。
  - ◆ 支持用户自己定义的不可分辨关系概念。
  - ◆ 对不同类型的不可分辨关系有效地计算精确约简和近似约简。
  - ◆ 产生 if - then 规则或以约简形式表达的描述模式。
  - ◆ 执行文件
  - ◆ 支持交叉验证测试。
- 后处理
  - ◆ 过滤约简结果和所得到的规则。
- 验证与分析
  - ◆ 用得到的规则处理未知样本。
  - ◆ 产生混淆矩阵、ROC 曲线和标度曲线。
  - ◆ 用一定的质量标准对规则进行评价。
  - ◆ 统计假设测试。
- 其他
  - ◆ 公差关系聚类。
  - ◆ 计算划分和可变精度 Rough 集近似。
  - ◆ 支持对观察的随机抽样。

Rosetta 在上述的功能中提供了很多可选的算法，是一个很好的研究实验平台。一个非商用的 Rosetta 系统版本可以在 HTTP://WWW.IDT.UNIT.NO/~ALEKS/ROSETTA/ROSETTA.HTML 下载得到。

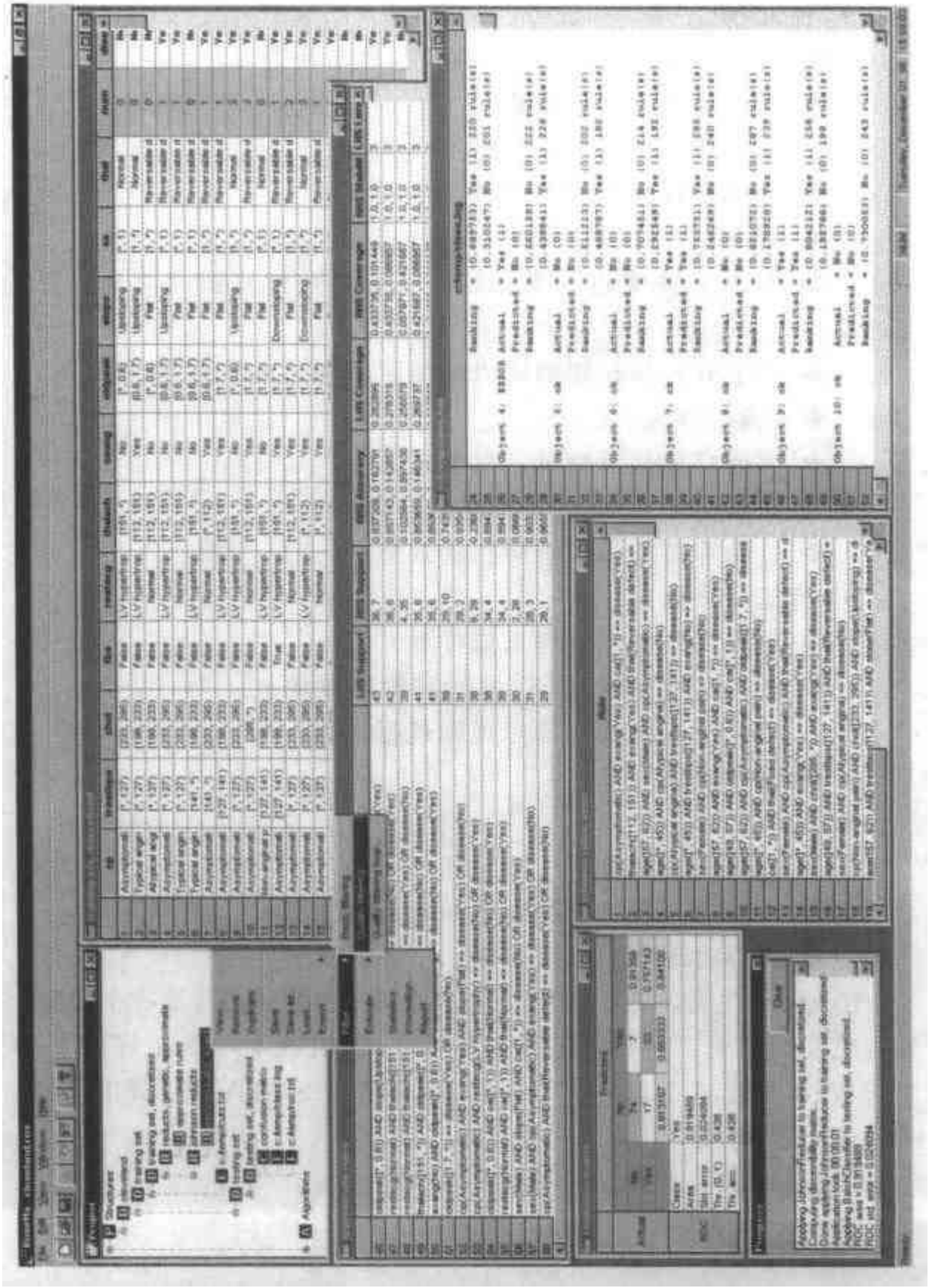


图 11-2 Rosetta 系统的界面结构

11.4　KDD－R

KDD－R 是由加拿大 Regina 大学研制开发的基于可变精度 Rough 集模型(VPRS,variable precision rough set)的数据库知识获取 KDD 系统。该系统是在 UNIX 系统下用 C 语言实现的,它具有 X－Windows 的菜单驱动界面。KDD－R 系统曾成功应用于医学数据分析和电信市场的决策分析等。

该系统由四大部分组成:

1. 数据预处理单元;
2. 属性依赖分析和消除冗余属性单元;
3. 规则提取单元;
4. 决策单元。

数据预处理单元把原始信息表中的数据进行离散化处理。首先,KDD－R 对于每个感兴趣的决策属性值 v 构造一个辅助表 T_v,在表中条件属性不变,而把决策属性划分为属于 v 和不属于 v 两部分。这样,就把 m 个决策属性值的原问题分解成 m 个子问题,每个都只有一个决策属性值。其次,对每个子表的条件属性值进行离散化。KDD－R 允许用户自己定义合适的区间范围来离散化数据值(手工离散化),也可以由系统自动进行离散化处理。系统通过查找质量准则 $Q(r_i)$ 最大的值范围,把每个实值属性替换为相应值范围上的一个或多个三值离散属性。

属性依赖分析和消除冗余属性单元是基于 VPRS 模型的。同原始 Rough 集模型相比,VPRS 在计算集合 Y 的下界、边界区和负区域时有一定的灵活性。具体而言,给定上限参数 β 和下限参数 $\mu(0\leqslant\beta\leqslant\mu\leqslant1)$,集合 Y 的 β 下近似定义为 $R_\beta(Y)=\bigcup\{E|(E\in R^*)\wedge c(E,Y)\leqslant\beta\}$,其中,$R^*$ 是等价关系簇,而 $c(E,Y)=1-\mathrm{card}(E\cap Y)/\mathrm{card}(E)$ 称为分类因子。相应地可定义边界区域和负区域。用户需要提供相应的参数 β 和 μ,并表明分析是集中在 β 下

界还是 μ 上界。所谓 μ 上界是 β 下界和边界区域的并集。接着，KDD－R 使用 Rough 集相应的公式来计算条件属性和决策属性之间的依赖性、相对约简和核。

规则提取单元计算所有或部分带有决策概率(可信度)的近似规则，其中概率由上限参数 β 和下限参数 μ 来决定。可以使用决策矩阵方法来进行约简得到规则，也可以计算得到最大近似规则，即根据支持每条规则的数据集合的包含关系所确定的偏序关系中的最大元素的计算。这类规则在可用数据的支持意义上是最有力的，并且是相互独立的。用户也可以选择生成所有规则或指定规则的最小覆盖。

决策单元是对决策规则的控制单元。许多系统所采用的简单的决策方法是找到并使用前件满足条件的规则。这些方法忽略了从数据中获取的规则具有不确定和相关的决策概率。与之相反，KDD－R 使用最大条数的规则，把尽量多的规则组合在一起，并对输入计算每个决策类的决策分。为避免受每个决策类中规则数不同的影响，KDD－R 将决策分进行了规范，最后得到[－1,1]之间的决策分，分别表示对某类决策的支持程度。

11.5 LERS

LERS 是美国 Kansas 大学开发的基于 Rough 集的实例学习系统，它是用 Common Lisp 在 VAX9000 上实现的。LERS 系统曾用于医学研究、气候预测和环境保护等。

LERS 的输入采用特定的文件格式。该格式类似于信息表，但采用附加信息的方式来表明条件属性、决策属性、属性的优先级等信息。LERS 从输入文件中形成决策表并试图从主要的、优先级高的属性中生成规则。系统首先检查输入数据的一致性，若输入数据不一致，就计算每个概念的上、下近似。用户可以选择系统提供的机器学习或知识获取算法。对于机器学习算法，系统对每个概念产

生最小判别描述，这意味着系统导出一组足够完全描述每个概念的规则，但仅有部分属性-值对包含在规则中，还有许多规则未被发现。对于知识获取算法，系统产生所有能从输入数据中归纳出的规则，每个都是最小形式，可以用于专家系统等需要知道知识尽量多的场合。这两种情形中都可选择全局或局部算法。在局部算法中，系统计算属性-值对的最小覆盖。在全局算法中，每个概念由两块组成的置换划分来表示：一块是一个概念的上或下近似，另一块是它的补集。置换划分所依赖的所有属性集的最小子集，称为全局覆盖。

尽管机器学习选项不能产生全部规则，但是它的时间复杂度是多项式的，而知识获取算法的复杂度是指数的，这样对于大的输入文件可能不太实际。

参考文献

1 Parson S, Kubat M, Dohnal M. A Rough Set Approach to Reasoning under Uncertainty. J Expt Theor Artif Intell, 1995(7): 175～193

2 Duntsch I, Gediga G. Uncertainty Measures of Rough Set Prediction. J of Artificial Intelligence, 1998, 106(1): 77～107

3 Yao Y Y. Constructive and Algebraic Methods of the Theory of Rough Sets. Journal of Information Sciences, 1998(109): 21～47

4 Skowron A. Boolean Reasoning for Decision Rules Generation. 7th International Symposium on Methodologies for Intelligent Systems: 295～305

5 Pawlak Z. Rough Classification. Int J Man Machine Studies, 1984, 20: 469～483

6 Chen P, Toyota T. Multi-valued Neural Network and the Knowledge Acquisition Method by the Rough Sets for Ambiguous Recognition Problem. IEEE Int Conf on Syst, Man, and Cybern, 1996: 736～740

7 Dougherty J, Kohavi R, Sahami M. Supervised and Unsupervised Discretization of Continuous Features. 12th International Conference on Machine Learning, 1995

8 Hu X, Shan N, Cercone N, et al. DBROUGH: A Rough Set Based Knowledge Discovery System. 8th International Symposium on Methodologies for Intelligent Systems, 1994: 386～395

9 Shapiro G P. Data Mining and Knowledge Discovery in Business Databases. 9th International Symposium on Foundations of Intelligent Systems, 1996: 56～67

10 Tsumoto S, Ziarko W. The Application of Rough Sets - Based Data Mining Technique to Differential Diagnosis of Meningoenchepahlitis. 9th International Symposium on Foundations of Intelligent Systems, 1996: 438～447

11 Tsumoto S, Tanaka H. Extraction of Medical Diagnostic Knowledge

based on Rough Set Based Model Selection and Rule Induction. The Fourth International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery, 1996: 126--435

12 Komorowski O J, Skowron A, Synak P. The Design and Implementation of a Knowledge Discovery Toolkit Based on Rough Sets: The ROSETTA System. In: Polkowski L, Skowron A, ed. Rough Sets in Knowledge Discovery 1: Methodology and Applications. Studies in Fuzziness and Soft Computing, 18(19). Physica - Verlag

13 Komorowski O J, Skowron A, Synak P. The ROSETTA Software System, In: Polkowski L, Skowron A, ed. Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems. Studies in Fuzziness and Soft Computing, Physica - Verlag, 19: 572～576

14 Komorowski O J. ROSETTA - A Rough Set Toolkit for Analysis of Data. Fifth International Workshop on Rough Sets and Soft Computing, 1997, 403～407

15 Machado O L O. Using Boolean Reasoning to Anonymize Databases. Artificial Intelligence in Medicine, 1999, 15(3): 235～254

16 Krolikowski R, Czyzewski A. Noise Reduction in Telecommunication Channels Using Rough Sets and Neural Networks. 7th International Workshop on New Directions in Rough Sets, Data Mining, and Granular - Soft Computing, 1999, 100～108

17 Chouchoulas, Shen Q. A Rough Set - Based Approach to Text Classification. 7th International Workshop on New Directions in Rough Sets, Data Mining, and Granular - Soft Computing, 1999, 118～127

18 Shan A N, Chan C, Cercone N, et al. Discovering Rules for Water Demand Prediction: An Enhanced Rough - set Approach. Engng Applie Artif Intell, 1996, 9(6): 645～653

19 Nguyen H S, Skowron A. Boolean Reasoning for Feature Extraction Problems. 10th International Symposium on Foundations of Intelligent Systems, 1997, 116～127

20 Mollestad T, Skowron A. A Rough Set Framework for Data Mining of Propositional Default Rules. 9th International Symposium on

Foundations of Intelligent Systems, 1996, 148～457

21 Mazlack L. J. Autonomous Database Mining and Disorder Measures. 10th International Symposium on Foundations of Intelligent Systems, 1997, 308～317

22 Williklosgen. Knowledge Discovery in Database and Data Mining. 9th International Symposium on Foundations of Intelligent Systems, 1996, 623～632

23 Nguyen S H, Skowron A. Quantization of Real Value Attributes-Rough Set and Boolean Reasoning Approach. Proc of the Second Joint Conference on Information Sciences, 1995, 31～37

24 Pawlak Z. Vagueness — A Rough Set View. In: Mycielski J, Rozenberg G, Salomaa A, ed. Structures in Logic and Computer Science. Springer-Verlag, 1997, 106～117

25 Tsumoto S, Tanaka H. Induction of Expert System Rules from Databases based on Rough Set Theory and Resampling Methods. 9th International Symposium on Foundation of Intelligent Systems, 1996, 128～138

26 Shapiro G P. Data Mining and Knowledge Discovery in Business Database. 9th International Symposium on Foundation of Intelligent Systems, 1996, 56～67

27 Lin T Y, Liu Q, Yao Y Y. Logic System for Approximate Reasoning: Via Rough Sets and Topology. 8th International Symposium on Methodologies for Intelligent Systems, 1994, 65～74

28 Hatonen K, Klemettinen M, Mannila H, et al. Knowledge Discovery from Telecommunication Network Alarm Databases. Twelfth International Conference on Data Engineering, 1996, 115～122

29 Gediga D G. The Rough Set Engine GROBIAN. In: Sydow A, ed. Proc 15th IMACS World Congress. Wissenschaft und Technik Verlag, 1997, 613～618

30 Kohavi R. The Power of Decision Tables. European Conference on Machine Learning (ECML), 1995

31 Tsumoto S. Extraction of Experts Decision Process from Clinical

Databases Using Rough Set Model. 1th European Symposium on Principles of Data Mining and Knowledge Discovery, 1997, 58～67

32 Ziarko W, Cercone N, Hu X. Rule Discovery from Databases with Decision Matrices. 9th Int. Symposium on Foundation of Intelligent Systems, 1996, 653～662

33 Hu X, Cercone N. Mining Knowledge Rules from Databases: A Rough Set Approach. Twelfth International Conference On Data Engineering, 1996, 96～105

34 Shan N, Hamilton H J, Cercone N. Induction of Classification Rules from Imperfect Data. 9th Int Symposium on Foundation of Intelligent Systems, 1996, 118～127

35 Kryszkiewicz M. Generation of Rules from Incomplete Information Systems. 1st European Symposium on Principles of Data Mining and Knowledge Discovery, 1997, 156～166

36 Kryszkiewicz M. Rough Set Approach to Incomplete Information System. Information Sciences, 1998, 112: 39～49

37 Kryszkiewicz M. Properties of Incomplete Information Systems in the Framework of Rough Sets. In: Polkowski L, Skowron A, ed. Rough Sets in Data Mining and Knowledge Discovery. Physica Verlag, 1998, 422～450

38 Stefanowski J, Slowinski K. Rough Set Theory and Induction Techniques for Discovery of Attribute Dependencies in Medical Information Systems. First European Symposium on Principles of Data Mining and Knowledge Discovery, 1997, 36～46

39 Stefanowski J, Tsoukias A. On the Extension of Rough Sets under Incomplete Information. 7th International Workshop on New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, 1999, 73～81

40 Bazan J G, Skowron A, Synak P. Dynamic Reducts as a Tool for Extracting Laws from Decisions Tables. 8th International Symposium on Methodologies for Intelligent Systems, 1994, 346～355

41 Skowron A. Boolean Reasoning for Decision Rules Generation. 7th

International Symposium on Methodologies for Intelligent Systems, 1993, 295～305

42 Torres G L, Quintana V H, da Silva A P A, et al. Classification of Power System Operation Point using Rough Set Techniques. IEEE International Conference on Systems, Man and Cybernetics, Information Intelligence and Systems, 1996, 1898～1903

43 Slowinski P R, Stefanowski J, Susmaga R, et al. ROSE-Software Implementation of the Rough Set Theory. First International Conference on Rough Sets and Current Trends in Computing, 1998, 605～608

44 Zhou Y, Wang J. Rule - Exception Modeling Based on Rough Set Theory. First International Conference on Rough Sets and Current Trends in Computing, 1998, 529～536

45 Zhong N. Methodologies for Knowledge Discovery and Data Mining. Third Pacific-Asia Conference, Beijing, 1999

46 Ohrn. ROSETTA Technical Reference Manual, Knowledge Systems Group. Dept Of Computer and Information Science, Norwegian University of Science and Technology, 2000

47 Ohrn. The ROSETTA C++ Library: Overview of Files and Classes, Knowledge Systems Group. Dept. Of Computer and Information Science, Norwegian University of Science and Technology, 2000

48 Pawlak Z. Rough Sets. International Journal of computer and information Sciences, 1982(11): 341～356.

49 Slowinski R. Rough Set Learning of Preferential Attitude in Multi - criteria Decision Making. 7th International Symposium on Methodologies for Intelligent Systems, 1993, 642～651

50 Slowinski R, Stefanowski J. Rough Classification in Incomplete Information Systems. Math Computing Modelling, 1989, 12(10/11): 1347～1357

51 Busse J W G. On the Unknown Attribute Values in Learning From Examples. Proc of Int Symp On Methodologies for Intelligent Systems, 1991, 368～377

52 Tsumoto S. Induction of Positive and Negative Deterministic Rules

based on Rough Set Model. 10th International Symposium on Foundations of Intelligent Systems, 1997, 298～307

53 Mazlack L J. Autonomous Database Mining and Disorder Measures. 10th International Symposium on Foundations of Intelligent Systems, 1997, 308～317

54 Tsumoto S, Tanaka H. Induction of Probabilistic Rules Based on Rough Set Theory. 4th International Workshop on Algorithmic Learning Theory, 1993, 410～423

55 Tsumoto S. Modeling Medical Diagnostic Rules Based on Rough Sets. First International Conference on Rough Sets and Current Trends in Computing, 475～482

56 Pawlak Z, Busse J G, Slowinski R, et al. Rough Sets. Communication of the ACM, 1995, 38(11): 89～95

57 Nguyen H S, Nguyen S H, Skowron A. Searching for Features Defined by Hyperplanes. 9th International Symposium on Foundations of Intelligent Systems, 1996, 366～375

58 Vinterbo O S, Szymanski P. Modeling Cardiac Patient Set Residuals Using Rough Sets. Proc AMIA Annual Fall Symposium, 1997, 203～207

59 Kohavi R, Frasca B. Useful Feature Subsets and Rough Set Reducts. 3th International Workshop on Rough Sets and Soft Computing, 1994

60 Kryszkiewicz M, Ryhinski H. Reducing Information Systems with Uncertain Attributes. 9th International Symposium on Foundations of Intelligent Systems, 1996, 285～294

61 Bonikowski Z, Bryniarski E. Extensions and Intentions in the Rough Set Theory. Journal of Information Sciences, 1998, 107: 149～167

62 Pawlak Z, Wong S K M, Ziarko W. Rough Sets: Probabilistic versus Deterministic Approach. Int J Man-Machine Studies, 1998, 29: 81～95

63 Duntsch B I, Gediga G. IRIS Revisited: A Comparison of Discriminant and Enhanced Rough Set Data Analysis. In: Polkowski L, Skowron A, ed. Rough Sets in Knowledge Discovery. Physica-Verlag, 1998, 345～368

64 Ziarko W. Variable Precision Rough Set Model. Journal of Computer and System Sciences, 1993, 46: 39～59

65 Pawlak Z, Slowinski K, Slowinski R. Rough Classification of Patients after highly Selective Vagotomy duodenal ulcer. International Journal of Man-Machine Studies, 1986, 24: 113～433

66 Greco S, Matarazzo B, Slowinski R. Handling Missing Values in Rough Set Analysis of Multi - attribute and Multi - criteria Decision Problems. 7th International Workshop on New Directions in Rough Sets, Data Mining, and Granular - Soft Computing, 1999, 146～157

67 Gediga D G. Rough Set Data Analysis. Encyclopedia of Computer Science and Technology, Marcel Dekker, 2000

68 Polkowski L, Skowron A. First International Conference on Rough Sets and Current Trends in Computing. Springer, 1998

69 Zhong N, Skowron A, Ohsuga S. 7th International Workshop on New Directions in Rough Sets, Data Mining, and Granular - Soft Computing. Springer, 1999

70 Krolikowski R, Czyzewski A. Noise Reduction in Telecommunication Channels Using Rough Sets and Neural Networks. 7th International Workshop on New Directions in Rough Sets, Data Mining, and Granular - Soft Computing, Yamaguchi, Japan, 1999, 100～108

71 Wang G Y, Fisher P S. Rule Generation Based on Rough Set Theory. In: Dasarathy B V, ed. Data Mining and Knowledge Discovery: Theory, Tools, and Technology II, Proceedings of SPIE Vol. 4057, 2000, 181～189

72 Wang G Y, Wu Y, Liu F. Generating Rules and Reasoning under Inconsistencies. IEEE International Conference on Industrial Electronics, Control and Instrumentation, Nagoya, Japan, 2000, 2536～2541

73 Wang G Y, Liu F. The Inconsistency in Rough Set Based Rule Generation. The Second International Conference on Rough Sets and Current Trends in Computing, 2000, 332～339

74 Wang G Y. The Algebra View and Information View of Rough Sets Theory. SPIE Conference on Dare Mining and Knowledge Discovery Ⅲ, Proceedings of SPIE Vol. 4384, Orlando, Florida USA, 2001

75 王珏，王任，苗夺谦等. 基于 Rough Set 理论的"数据浓缩". 计算机学报，1998，21(5)：393～400

76 苗夺谦，王珏. 基于粗糙集的多变量决策树构造方法. 软件学报，1997，8(6)：425～431

77 周育健，王珏. RSL-基于 Rough Set 的表示语言. 软件学报，1997，8(8)：569～576

78 王珏，苗夺谦，周育健. 关于 Rough Set 理论与应用的综述. 模式识别与人工智能，1994，9(1)：337～344

79 王志海，胡可云，胡学钢等. 基于粗糙集合理论的知识发现综述. 模式识别与人工智能，1998，11(2)：176～183

80 刘真. Rough 集. 计算机科学，1997，24(1)：15～19

81 刘清，黄兆华，刘少辉等. 带 Rough 算子的决策规则及数据挖掘中的软计算. 计算机研究与发展，1999，36(7)：800～804

82 吴福保，李奇，宋文忠. 基于粗集理论知识表达系统的一种归纳学习方法. 控制与决策，1999，11(3)：206～211

83 常犁云，王国胤，吴渝. 一种 Rough Set 理论的属性约简及规则提取方法. 软件学报，1999，10(11)：1206～1211

84 侯利娟，王国胤，吴渝等. 粗糙集理论中的离散化问题. 计算机科学，2000，27(12)：89～94

85 王国胤，刘锋，吴渝等. Rough 集规则知识获取研究中的不一致问题. 重庆邮电学院学报，2000，12(3)，16～21

86 苗夺谦，胡桂荣. 知识约简的一种启发式算法. 计算机研究与发展，1999，36(6)：681～684

87 苗夺谦，王珏. 粗糙集理论中概念与运算的信息表示. 软件学报，1999，10(2)：113～116

88 陈遵德. Rough Set 神经网络智能系统及其应用. 模式识别与人工智能，1999，12(1)：1～5

89 马志锋，邢汉承，郑晓妹等. 基于不分明与相似关系的 Rough 集的超图描述. 计算机科学，1999，26(9)：35～39

90 曾黄麟. 粗集理论及其应用——关于数据推理的新方法. 重庆：重庆大学出版社，1998

91 尹旭日，周志华，陈世福. 一种基于 Rough 集理论的不完备数据分析方

法，待发表
92 蒋远承，胜跃飞，王驹. Formalization of the Conditional Entropy in Rough Set Theory，待发表
93 罗旭东，邱玉辉. 专家系统中的不确定推理——模型、方法和理论. 北京：科学技术文献出版社，1995
94 方世昌. 离散数学. 西安：西安电子科技大学出版社，1989
95 黄可鸣. 专家系统. 南京：东南大学出版社，1991
96 靳蕃. 神经计算智能基础原理·方法. 成都：西南交通大学出版社，2000