火龙果•整理
uml.org.cn

目录

第 1 章　引言 3
第 2 章　数据预处理 6
第 3 章　数据仓库与OLAP技术概述 14
第 4 章　数据立方体计算与数据泛化 23
第 5 章　挖掘频繁模式、关联和相关 28
第 6 章　分类和预测 32
第 7 章　聚类分析 35

火龙果•整理
uml.org.cn

第1章　引言

1.1 什么是数据挖掘？在你的回答中，针对以下问题：

1.2 1.6 定义下列数据挖掘功能：特征化、区分、关联和相关分析、预测聚类和演变分析。使用你熟悉的现实生活的数据库，给出每种数据挖掘功能的例子。

解答：

- **特征化**是一个目标类数据的一般特性或特性的汇总。例如，学生的特征可被提出，形成所有大学的计算机科学专业一年级学生的轮廓，这些特征包括作为一种高的年级平均成绩(GPA：Grade point aversge)的信息，还有所修的课程的最大数量。
- **区分**是将目标类数据对象的一般特性与一个或多个对比类对象的一般特性进行比较。例如，具有高 GPA 的学生的一般特性可被用来与具有低 GPA 的一般特性比较。最终的描述可能是学生的一个一般可比较的轮廓，就像具有高 GPA 的学生的 75%是四年级计算机科学专业的学生，而具有低 GPA 的学生的 65%不是。
- **关联**是指发现关联规则，这些规则表示一起频繁发生在给定数据集的特征值的条件。例如，一个数据挖掘系统可能发现的关联规则为：major(X, “computing science”)⇒owns(X, “personal computer”) [support=12%, confidence=98%]
  其中，X 是一个表示学生的变量。这个规则指出正在学习的学生，12%（**支持度**）主修计算机科学并且拥有一台个人计算机。这个组一个学生拥有一台个人电脑的概率是 98%（置信度，或确定度）。
- **分类**与**预测**不同，因为前者的作用是构造一系列能描述和区分数据类型或概念的模型（或功能），而后者是建立一个模型去预测缺失的或无效的、并且通常是数字的数据值。它们的相似性是他们都是预测的工具：分类被用作预测目标数据的类的标签，而预测典型的应用是预测缺失的数字型数据的值。

火龙果•整理
uml.org.cn

- **聚类**分析的数据对象不考虑已知的类标号。对象根据最大花蕾内部的相似性、最小化类之间的相似性的原则进行聚类或分组。形成的每一簇可以被看作一个对象类。聚类也便于分类法组织形式，将观测组织成类分层结构，把类似的事件组织在一起。
- **数据延边分析**描述和模型化随时间变化的对象的规律或趋势，尽管这可能包括时间相关数据的特征化、区分、关联和相关分析、分类、或预测，这种分析的明确特征包括时间序列数据分析、序列或周期模式匹配、和基于相似性的数据分析

1.3 1.9 列举并描述说明数据挖掘任务的五种原语。

解答：

用于指定数据挖掘任务的五种原语是：

- **任务相关数据：**这种原语指明给定挖掘所处理的数据。它包括指明数据库、数据库表、或数据仓库，其中包括包含关系数据、选择关系数据的条件、用于探索的关系数据的属性或维、关于修复的数据排序和分组。
- **挖掘的数据类型：**这种原语指明了所要执行的特定数据挖掘功能，如特征化、区分、关联、分类、聚类、或演化分析。同样，用户的要求可能更特殊，并可能提供所发现的模式必须匹配的模版。这些模版或超模式（也被称为超规则）能被用来指导发现过程。
- **背景知识：**这种原语允许用户指定已有的关于挖掘领域的知识。这样的知识能被用来指导知识发现过程，并且评估发现的模式。关于数据中关系的概念分层和用户信念是背景知识的形式。
- **模式兴趣度度量：**这种原语允许用户指定功能，用于从知识中分割不感兴趣的模式，并且被用来指导挖掘过程，也可评估发现的模式。这样就允许用户限制在挖掘过程返回的不感兴趣的模式的数量，因为一种数据挖掘系统可能产生大量的模式。兴趣度测量能被指定为简易性、确定性、适用性、和新颖性的特征。
- **发现模式的可视化：**这种原语述及发现的模式应该被显示出来。为了使数据挖掘能有效地将知识传给用户，数据挖掘系统应该能将发现的各种形式的模式展示出来，正如规则、表格、饼或条形图、决策树、立方体

火龙果•整理
uml.org.cn

或其它视觉的表示。

1.4 1.13 描述以下数据挖掘系统与数据库或数据仓库集成方法的差别：不耦合、松散耦合、半紧耦合和紧密耦合。你认为哪种方法最流行，为什么？

解答：

数据挖掘系统和数据库或数据仓库系统的集成的层次的差别如下。

- **不耦合：**数据挖掘系统用像平面文件这样的原始资料获得被挖掘的原始数据集，因为没有数据库系统或数据仓库系统的任何功能被作为处理过程的一部分执行。因此，这种构架是一种糟糕的设计。
- **松散耦合：**数据挖掘系统不与数据库或数据仓库集成，除了使用被挖掘的初始数据集的源数据和存储挖掘结果。这样，这种构架能得到数据库和数据仓库提供的灵活、高效、和特征的优点。但是，在大量的数据集中，由松散耦合得到高可测性和良好的性能是非常困难的，因为许多这种系统是基于内存的。
- **半紧密耦合：**一些数据挖掘原语，如聚合、分类、或统计功能的预计算，可在数据库或数据仓库系统有效的执行，以便数据挖掘系统在挖掘-查询过程的应用。另外，一些经常用到的中间挖掘结果能被预计算并存储到数据库或数据仓库系统中，从而增强了数据挖掘系统的性能。
- **紧密耦合：**数据库或数据仓库系统被完全整合成数据挖掘系统的一部份，并且因此提供了优化的数据查询处理。这样的话，数据挖掘子系统被视为一个信息系统的功能组件。这是一中高度期望的结构，因为它有利于数据挖掘功能、高系统性能和集成信息处理环境的有效实现。

从以上提供的体系结构的描述看，紧密耦合是最优的，没有值得顾虑的技术和执行问题。但紧密耦合系统所需的大量技术基础结构仍然在发展变化，其实现并非易事。因此，目前最流行的体系结构仍是半紧密耦合，因为它是松散耦合和紧密耦合的折中。

1.5 1.14 描述关于数据挖掘方法和用户交互问题的三个数据挖掘挑战。

火龙果•整理
uml.org.cn

第2章　数据预处理

2.1 2.2 假设给定的数据集的值已经分组为区间。区间和对应的频率如下。

年龄	频率
1~5	200
5~15	450
15~20	300
20~50	1500
50~80	700
80~110	44

计算数据的近似中位数值。

解答：

先判定中位数区间：N=200+450+300+1500+700+44=3194；N/2=1597

∵ 200+450+300=950<1597<2450=950+1500；

∴ 20~50 对应中位数区间。

我们有：L_1=20，N=3197，$(\sum freq)_l$=950，$freq_{median}$=1500，$width$=30，使用公式（2.3）：

$$median = L_1 + \left(\frac{N/2 - \left(\sum freq\right)_l}{freq_{median}} \right) width = 20 + \left(\frac{3197/2 - 950}{1500} \right) \times 30 = 32.97$$

∴ $median$=32.97 岁。

2.2 2.4 假定用于分析的数据包含属性 age。数据元组的 age 值（以递增序）是：13，15，16，16，19，20，20，21，22，22，25，25，25，25，30，33，33，35，35，35，35，36，40，45，46，52，70。

(a) 该数据的均值是什么？中位数是什么？

(b) 该数据的众数是什么？讨论数据的峰（即双峰、三峰等）。

(c) 数据的中列数是什么？

(d) 你能（粗略地）找出数据的第一个四分位数（Q_1）和第三个四分位数（Q_3）

火龙果•整理
uml.org.cn

吗？

(e) 给出数据的五数概括。

(f) 画出数据的盒图。

(g) 分位数—分位数图与分位数图的不同之处是什么？

解答：

(a) 该数据的均值是什么？中位数是什么？

均值是：$\bar{x}=\frac{1}{N}\sum_{i=1}^{N}x_i=809/27=29.96\cong 30$（公式 2.1）。中位数应是第 14 个，即$x_{14}$=25=$Q_2$。

(b) 该数据的众数是什么？讨论数据的峰（即双峰、三峰等）。

这个数集的众数有两个：25 和 35，发生在同样最高的频率处，因此是双峰众数。

(c) 数据的中列数是什么？

数据的中列数是最大术和最小是的均值。即：*midrange*=(70+13)/2=41.5。

(d) 你能（粗略地）找出数据的第一个四分位数（Q_1）和第三个四分位数（Q_3）吗？

数据集的第一个四分位数应发生在 25%处，即在(N+1)/4=7 处。所以：Q_1=20。而第三个四分位数应发生在 75%处，即在 3×(N+1)/4=21 处。所以：Q_3=35

(e) 给出数据的五数概括。

一个数据集的分布的 5 数概括由最小值、第一个四分位数、中位数、第三个四分位数、和最大值构成。它给出了分布形状良好的汇总，并且这些数据是：13、20、25、35、70。

(f) 画出数据的盒图。

略。

(g) 分位数—分位数图与分位数图的不同之处是什么？

分位数图是一种用来展示数据值低于或等于在一个单变量分布中独立的变量的粗略百分比。这样，他可以展示所有数的分位数信息，而为独立变量测得的值（纵轴）相对于它们的分位数（横轴）被描绘出来。

但分位数—分位数图用纵轴表示一种单变量分布的分位数，用横轴表示另一

火龙果•整理
uml.org.cn

单变量分布的分位数。两个坐标轴显示它们的测量值相应分布的值域，且点按照两种分布分位数值展示。一条线（y=x）可画到图中，以增加图像的信息。落在该线以上的点表示在 y 轴上显示的值的分布比 x 轴的相应的等同分位数对应的值的分布高。反之，对落在该线以下的点则低。

2.3 2.7 使用习题 2.4 给出的 age 数据回答下列问题：

(a) 使用分箱均值光滑对以上数据进行光滑，箱的深度为 3。解释你的步骤。评述对于给定的数据，该技术的效果。

(b) 如何确定数据中的离群点？

(c) 对于数据光滑，还有哪些其他方法？

解答：

(a) 使用分箱均值光滑对以上数据进行光滑，箱的深度为 3。解释你的步骤。评述对于给定的数据，该技术的效果。

用箱深度为 3 的分箱均值光滑对以上数据进行光滑需要以下步骤：

- 步骤 1：对数据排序。（因为数据已被排序，所以此时不需要该步骤。）
- 步骤 2：将数据划分到大小为 3 的等频箱中。

箱 1：13，15，16　　箱 2：16，19，20　　箱 3：20，21，22

箱 4：22，25，25　　箱 5：25，25，30　　箱 6：33，33，35

箱 7：35，35，35　　箱 8：36，40，45　　箱 9：46，52，70

- 步骤 3：计算每个等频箱的算数均值。
- 步骤 4：用各箱计算出的算数均值替换每箱中的每个值。

箱 1：44/3，44/3，44/3　箱 2：55/3，55/3，55/3　箱 3：21，21，21

箱 4：24，24，24　　箱 5：80/3，80/3，80/3　　箱 6：101/3，101/3，101/3

箱 7：35，35，35　　箱 8：121/3，121/3，121/3　箱 9：56，56，56

(b) 如何确定数据中的离群点？

聚类的方法可用来将相似的点分成组或“簇”，并检测离群点。落到簇的集外的值可以被视为离群点。作为选择，一种人机结合的检测可被采用，而计算机用一种事先决定的数据分布来区分可能的离群点。这些可能的离群点能被用人工轻松的检验，而不必检查整个数据集。

(c) 对于数据光滑，还有哪些其他方法？

火龙果•整理
uml.org.cn

其它可用来数据光滑的方法包括别的分箱光滑方法，如中位数光滑和箱边界光滑。作为选择，等宽箱可被用来执行任何分箱方式，其中每个箱中的数据范围均是常量。除了分箱方法外，可以使用回归技术拟合成函数来光滑数据，如通过线性或多线性回归。分类技术也能被用来对概念分层，这是通过将低级概念上卷到高级概念来光滑数据。

2.4 2.10 如下规范化方法的值域是什么？

(a) min-max 规范化。

(b) z-score 规范化。

(c) 小数定标规范化。

解答：

(a) min-max 规范化。

值域是[new_min, new_max]。

(b) z-score 规范化。

值域是[(old_min－mean)/σ，(old_max－mean)/σ]，总的来说，对于所有可能的数据集的值域是(－∞，+∞)。

(c) 小数定标规范化。

值域是(－1.0, 1.0)。

2.5 2.12 使用习题 2.4 给出的 age 数据，回答以下问题：

(a) 使用 min-max 规范化将 age 值 35 变换到[0.0，1.0]区间。

(b) 使用 z-score 规范化变换 age 值 35，其中 age 的标准差为 12.94 岁。

(c) 使用小数定标规范化变换 age 值 35。

(d) 对于给定的数据，你愿意使用哪种方法？陈述你的理由。

解答：

(a) 使用 min-max 规范化将 age 值 35 变换到[0.0，1.0]区间。

∵ min_A=13，max_A=70，new_min_A=0.0，new_max_A=1.0，而v=35，

$$v' = \frac{v - \min_A}{\max_A - \min_A}(new_\max_A - new_\min_A) + new_\min_A$$

$$= \frac{35-13}{70-13}(1.0-0.0) + 0.0 = 0.3860$$

火龙果•整理
uml.org.cn

(b) 使用 z-score 规范化变换 age 值 35，其中 age 的标准差为 12.94 岁。

$$\overline{A}=\frac{13+15+2\times16+19+2\times20+21+2\times22+4\times25}{27}+\frac{30+2\times33+4\times35+36+40+45+46+52+70}{27}=\frac{809}{27}=29.963$$

$$\sigma_A^2=\frac{\sum_{i=1}^{N}\left(A_i-\overline{A}\right)}{N}=161.2949，\quad \sigma_A=\sqrt{\sigma_A^2}=12.7002$$

或 $s_A^2=\frac{\sum_{i=1}^{N}\left(A_i-\overline{A}\right)}{N}=167.4986$， $s_A=\sqrt{s_A^2}=12.9421$

v=35

$$v_\sigma'=\frac{v-\overline{A}}{\sigma_A}=\frac{35-29.963}{12.7002}=\frac{5.037}{12.7002}=0.3966\approx0.400$$

或 $v_s'=\frac{v-\overline{A}}{s_A}=\frac{35-29.963}{12.9421}=\frac{5.037}{12.9421}=0.3892\approx0.39$

(c) 使用小数定标规范化变换 age 值 35。

由于最大的绝对值为 70，所以 j=2。 $v'=\frac{v}{10^j}=\frac{35}{10^2}=0.35$

(d) 对于给定的数据，你愿意使用哪种方法？陈述你的理由。

略。

2.6 2.14 假设 12 个销售价格记录组已经排序如下：5，10，11，13，15，35，50，55，72，92，204，215。使用如下每种方法将其划分成三个箱。

(a) 等频（等深）划分。

(b) 等宽划分。

(c) 聚类。

解答：

(a) 等频（等深）划分。

bin1	5,10,11,13
bin1	15,35,50,55

火龙果•整理
uml.org.cn

bin1	72,91,204,215

(b) 等宽划分。

每个区间的宽度是：(215-5)/3=70

bin1	5,10,11,13,15,35,50,55,72
bin1	91
bin1	204,215

(c) 聚类。

我们可以使用一种简单的聚类技术：用 2 个最大的间隙将数据分成 3 个箱。

bin1	5,10,11,13,15
bin1	35,50,55,72,91
bin1	204,215

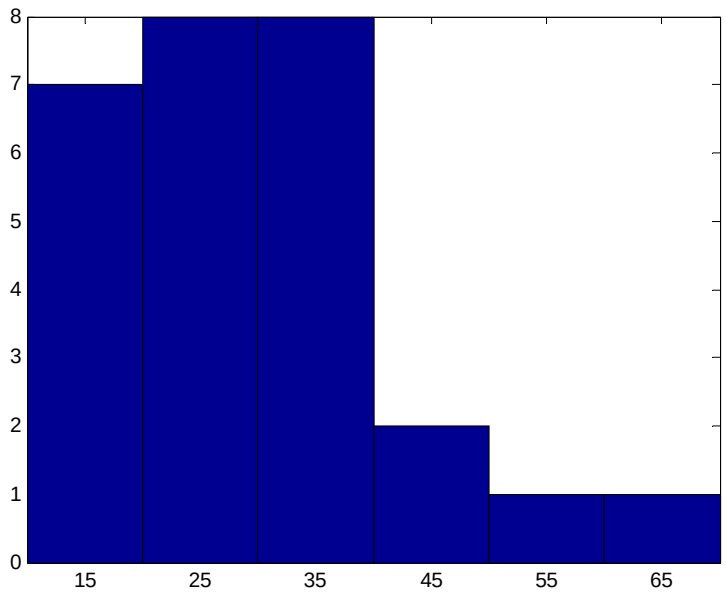
2.7 2.15 使用习题 2.4 给出的 age 数据，

(a) 画出一个等宽为 10 的等宽直方图；

(b) 为如下每种抽样技术勾画例子：SRSWOR，SRSWR，聚类抽样，分层抽样。使用大小为 5 的样本和层“青年”，“中年”和“老年”。

解答：

(a) 画出一个等宽为 10 的等宽直方图；



火龙果•整理
uml.org.cn

(b) 为如下每种抽样技术勾画例子：SRSWOR，SRSWR，聚类抽样，分层抽样。使用大小为 5 的样本和层“青年”，“中年”和“老年”。

元组：

T_1	13	T_{10}	22	T_{19}	35
T_2	15	T_{11}	25	T_{20}	35
T_3	16	T_{12}	25	T_{21}	35
T_4	16	T_{13}	25	T_{22}	36
T_5	19	T_{14}	25	T_{23}	40
T_6	20	T_{15}	30	T_{24}	45
T_7	20	T_{16}	33	T_{25}	46
T_8	21	T_{17}	33	T_{26}	52
T_9	22	T_{18}	35	T_{27}	70

SRSWOR 和 SRSWR：不是同次的随机抽样结果可以不同，但前者因无放回所以不能有相同的元组。

SRSWOR	(n=5)	SRSWR	(n=5)
T_4	16	T_7	20
T_6	20	T_7	20
T_{10}	22	T_{20}	35
T_{11}	25	T_{21}	35
T_{26}	52	T_{25}	46

聚类抽样：设起始聚类共有 6 类，可抽其中的 m 类。

Sample1		Sample2		Sample3		Sample4		Sample5		Sample6	
T_1	13	T_6	20	T_{11}	25	T_{16}	33	T_{21}	35	T_{26}	52
T_2	15	T_7	20	T_{12}	25	T_{17}	33	T_{22}	36	T_{27}	70
T_3	16	T_8	21	T_{13}	25	T_{18}	35	T_{23}	40		
T_4	16	T_9	22	T_{14}	25	T_{19}	35	T_{24}	45		
T_5	19	T_{10}	22	T_{15}	30	T_{20}	35	T_{25}	46		

Sample2	Sample5

火龙果•整理
uml.org.cn

T_6	20	T_{21}	35
T_7	20	T_{22}	36
T_8	21	T_{23}	40
T_9	22	T_{24}	45
T_{10}	22	T_{25}	46

分层抽样：按照年龄分层抽样时，不同的随机试验结果不同。

T_1	13	young	T_{10}	22	young	T_{19}	35	middle age
T_2	15	young	T_{11}	25	young	T_{20}	35	middle age
T_3	16	young	T_{12}	25	young	T_{21}	35	middle age
T_4	16	young	T_{13}	25	young	T_{22}	36	middle age
T_5	19	young	T_{14}	25	young	T_{23}	40	middle age
T_6	20	young	T_{15}	30	middle age	T_{24}	45	middle age
T_7	20	young	T_{16}	33	middle age	T_{25}	46	middle age
T_8	21	young	T_{17}	33	middle age	T_{26}	52	middle age
T_9	22	young	T_{18}	35	middle age	T_{27}	70	senior

T_4	16	young
T_{12}	25	young
T_{17}	33	middle age
T_{25}	46	middle age
T_{27}	70	Senior

2.8 5555555555555555555555555555

火龙果•整理
uml.org.cn

第3章　数据仓库与 OLAP 技术概述

3.1 3.3 假定数据仓库包含三维：time，doctor 和 patient；和两个度量：count 和 charge；其中，charge 是医生对病人一次诊治的收费。

(a) 列举三种流行的数据仓库建模模式。

(b) 使用(a)列举的模式之一，画出上面的数据仓库的模式图。

(c) 由基本方体[day, doctor, patient]开始，为了列出 2004 年每位医生的收费总数，应当执行哪些 OLAP 操作？

(d) 为了得到同样的结果，写出一个 SQL 查询。假定数据存放在关系数据库中，其模式为

fee(*day*, *month*, *year*, *doctor*, *hospital*, *patient*, *count*, *charge*)。

解答：

(a) 列举三种流行的数据仓库建模模式。

三种流行的数据仓库建模模式是星形模式、雪花形模式、实事星座模式。

(b) 使用(a)列举的模式之一，画出上面的数据仓库的模式图。

图 3.1 显示了一种星形模式。

火龙果•整理
uml.org.cn

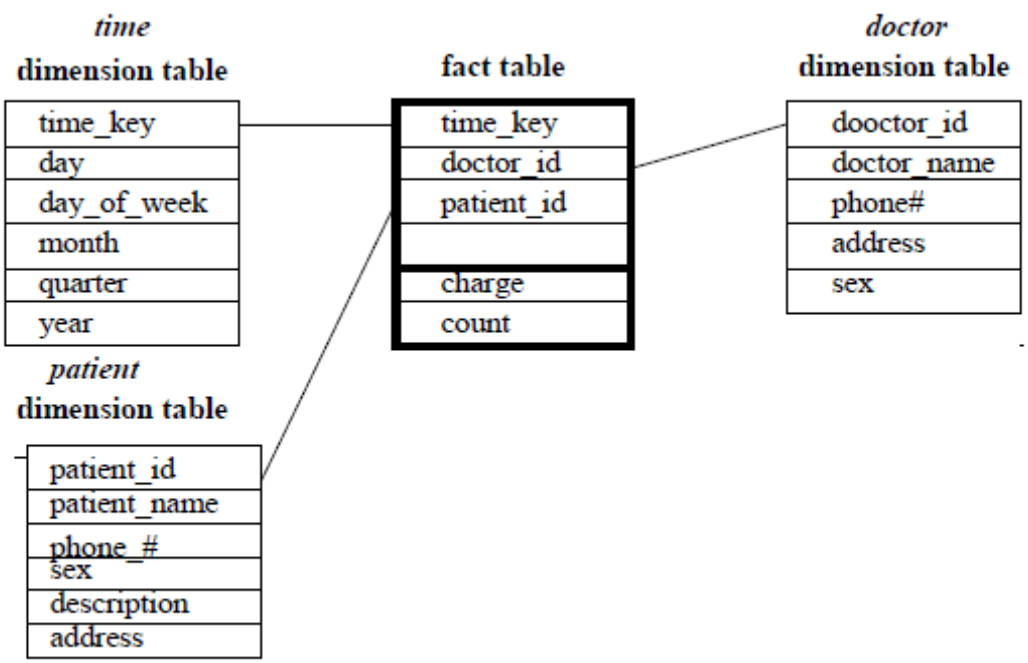


Figure 3.1: A star schema for data warehouse of Exercise 3.3.

(c) 由基本方体[*day*, *doctor*, *patient*]开始，为了列出 2004 年每位医生的收费总数，应当执行哪些 *OLAP* 操作？

应执行的操作包括：

- 在时间维从 *day* 上卷到 *year*。
- 在时间维切片，*time*=2004。
- 在病人维从单个病人 individual patient 上卷到所有病人 all。

(d) 为了得到同样的结果，写出一个 SQL 查询。假定数据存放在关系数据库中，其模式为 *fee*(*day*, *month*, *year*, *doctor*, *hospital*, *patient*, *count*, *charge*)。

Select doctor, SUM(charge)

from fee

where year=2004

group by doctor

■

3.2 3.4 假定 BigUniversity 的数据仓库包含如下 4 个维：student(student_name, area_id, major, status, university)，course(course_name, department)，semester(semester, year)和 instructor(dept, rank)；2 个度量：count 和 avg_grade。

火龙果•整理
uml.org.cn

在最低概念层，度量 avg_grade 存放学生的实际课程成绩。在较高概念层，avg_grade 存放给定组合的平均成绩。

(a) 为该数据仓库画出雪花形模式图。

(b) 由基本方体 [student, course, semester, instructor] 开始，为列出 BigUniversity 每个学生的 CS 课程的平均成绩，应当使用哪些特殊的 OLAP 操作。

(c) 如果每维有 5 层（包括 all），如“student<major<status<university<all”，该立方体包含多少方体？

解答：

a) 为该数据仓库画出雪花形模式图。雪花模式如图所示。

b) 由基本方体 [student, course, semester, instructor] 开始，为列出 BigUniversity 每个学生的 CS 课程的平均成绩，应当使用哪些特殊的 OLAP 操作。

这些特殊的联机分析处理（OLAP）操作有：

i. 沿课程（course）维从 course_id“上卷”到 department。

ii. 沿学生（student）维从 student_id“上卷”到 university。

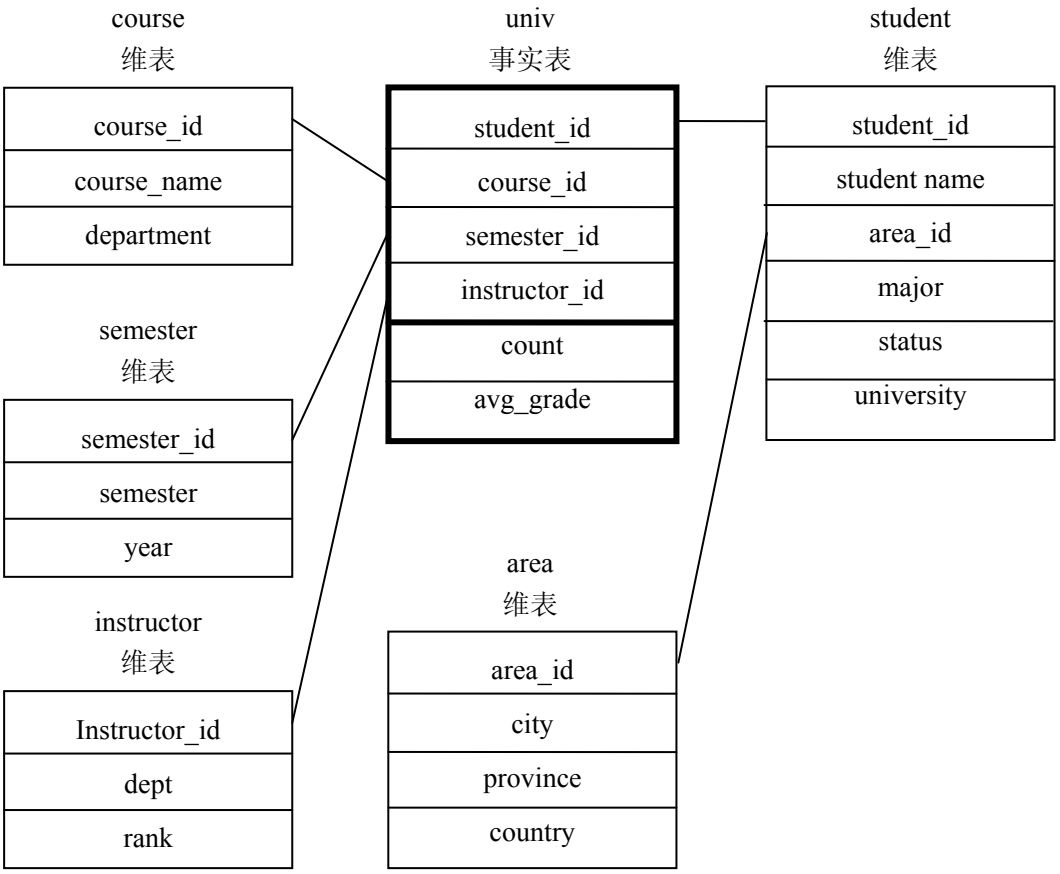
iii. 取 department=“CS”和 university=“Big University”，沿课程（course）维和学生（student）维切片。

iv. 沿学生（student）维从 university 下钻到 student_name。

c) 如果每维有 5 层（包括 all），如“student<major<status<university<all”，该立方体包含多少方体？

这个立方体将包含 5^4=625 个方体。

火龙果•整理
uml.org.cn



题 3.4 图 题 3.4 中数据仓库的雪花形模式

3.3 3.5 假定数据仓库包含 4 个维：date, spectator, location 和 game；和 2 个度量：count 和 charge，其中 charge 是观众在给定的日期观看节目的付费。观众可以是学生、成年人或老年人，每类观众有不同的收费标准。

(e) 画出该数据仓库的星形模式图。

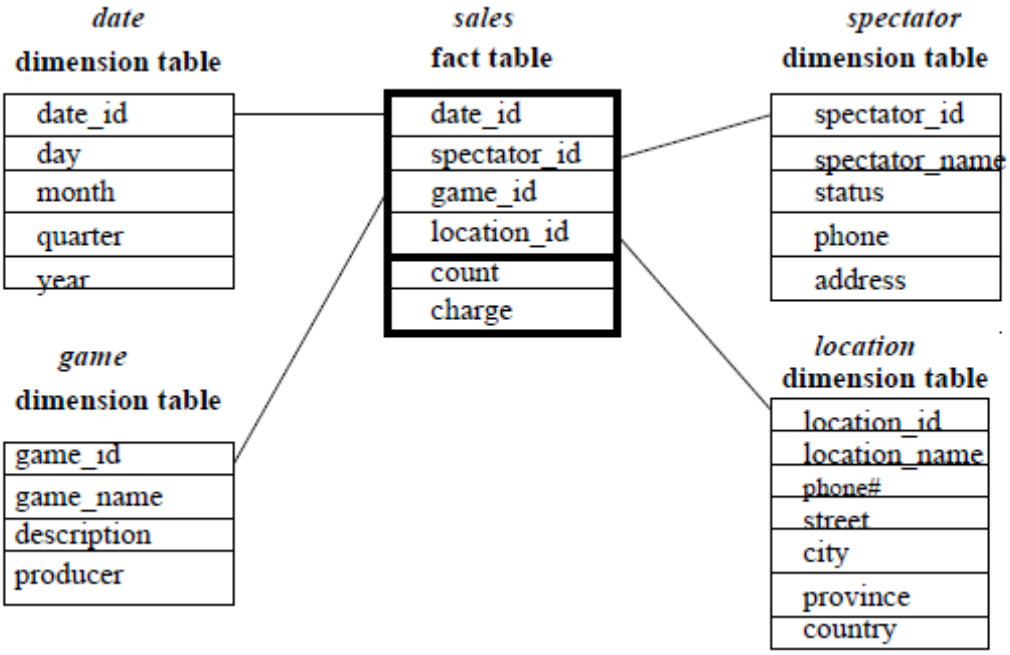
(f) 由基本方体[date, spectator, location, game]开始，为列出 2004 年学生观众在 GM_Place 的总付费，应当执行那些 OLAP 操作？

(g)对于数据仓库，位图索引是有用的。以该立方体为例，简略讨论使用位图索引结构的优点和问题。

解答：

(a) 画出该数据仓库的星形模式图。

火龙果•整理
uml.org.cn



题 3.5 图 练习 3.5 的用于数据仓库的星形模式

(b) 由基本方体[date, spectator, location, game]开始，为列出 2004 年学生观众在 GM_Place 的总付费，应当执行那些 OLAP 操作？

- 沿日期（date）轴从 date_id 上卷到 year。
- 沿节目（game）轴从 game_id 上卷到 all。
- 沿地点（location）轴从 location_id 上卷到 location_name。
- 沿观众（spectator）轴从 spectator_id 上卷到 status。
- 用 status_ = “status”，location= “GM_Place”，和 year=2004。

(c) 对于数据仓库，位图索引是有用的。以该立方体为例，简略讨论使用位图索引结构的优点和问题。

位图索引适用于集的势较低的领域。例如，对于这个立方体，如果地点维是位图索引，在 location 轴上的比较、连接、和聚集操作可被简化为位计算，从而减少了处理时间。而且，长的地点名称字符串能用一位来表示，这样就使存储空间和输入输出时间显著的减小。对于有较高的集的势的维，就像本例中的日期(date)维，用于表示位索引的向量会非常长。例如，一个历时 10 年的数据集合可导致 3650 个日记录，意味着实事表中的每个元组需要 3650 位（或大约 456 字节）去表示位图索引。

火龙果•整理
uml.org.cn

3.4 3.9 关于数据立方体度量计算：

(a) 根据计算数据立方体所用的聚集函数，列出度量的三种类型。

(b) 对于具有 3 个维time, location和product的数据立方体，函数variance（方差）属于哪一类？如果该立方体分割成一些块，描述如何计算它。提示：计算variance函数的公式是：$\frac{1}{N}\sum_{i=1}^{n}(x_i-\bar{x})^2$，其中，$\bar{x}$ 是这些x_i的平均值。

(c) 假定函数是“最高的 10 个销售额”。讨论如何在数据立方体里有效的计算该度量。

解答：

(a) 根据计算数据立方体所用的聚集函数，列出度量的三种类型。

三种度量分别是：分布式度量，代数式度量，整体式度量。

(b) 对于具有 3 个维time, location和product的数据立方体，函数variance（方差）属于哪一类？如果该立方体分割成一些块，描述如何计算它。提示：计算variance函数的公式是：$\frac{1}{N}\sum_{i=1}^{n}(x_i-\bar{x})^2$，其中，$\bar{x}$ 是这些x_i的平均值。

因为 variance 函数的计算公式是：

$$\frac{1}{N}\sum_{i=1}^{n}(x_i-\bar{x})^2=\frac{1}{N}\sum_{i=1}^{N}\left(x_i^{\ 2}-2x_i\bar{x}+\bar{x}^2\right)=\frac{1}{N}\left(\sum_{i=1}^{N}x_i^{\ 2}-2\left(\sum_{i=1}^{N}x_i\right)\bar{x}+N\bar{x}^2\right)$$

又由于：$\bar{x}=\frac{1}{N}\sum_{i=1}^{N}x_i$

因此方差函数可表示为：

$$\frac{1}{N}\sum_{i=1}^{n}(x_i-\bar{x})^2=\frac{1}{N}\left(\sum_{i=1}^{N}x_i^{\ 2}-\frac{2}{N}\left(\sum_{i=1}^{N}x_i\right)^2+\frac{1}{N}\left(\sum_{i=1}^{N}x_i\right)^2\right)=\frac{1}{N}\left(\sum_{i=1}^{N}x_i^{\ 2}-\frac{1}{N}\left(\sum_{i=1}^{N}x_i\right)^2\right)$$

所以，方差函数是代数度量。如果立方体可以被分成许多大块，方差可以用如下方法计算：逐个读取大块中的每一个数据，保留并累加如下聚集

(1) 元组的数量；

(2) (x_i^2)的和；

(3) x_i的和。

读完所有的块以后，将元组的数量N、(x_i^2)的和、x_i的和带入，就可应用上述

火龙果•整理
uml.org.cn

公式得到方差variance。

(c) 假定函数是“最高的 10 个销售额”。讨论如何在数据立方体里有效的计算该度量。

(1) 对每个立方体，初始时先用 10 个存储单元存储任意选定的 10 个销售额；

(2) 将这 10 个存储单元中的销售额由大到小排序；

(3) 在立方体中读一个元组，如果该元组中的销售额大于 10 个中最小销售额，就用当前销售额替换最小销售额。

(4) 重复(2)、(3)，直到读遍立方体中所有的元组后停止。

3.5 3.11 在数据仓库技术中，多维视图可以用关系数据库技术（ROLAP），或多维数据库技术（MOLAP），或混合数据库技术（HOLAP）实现。

(a) 简要描述每种实现技术。

(b) 对每种技术，解释如下函数如何实现：

i. 数据仓库的产生（包括聚集）。

ii. 上卷。

iii. 下钻。

iv. 增量更新。

(c) 你喜欢那种实现技术？为什么？

解答：

(a) 简要描述每种实现技术。

关系联机分析处理技术（ROLAP）是中间服务器，可用于执行多维视图任务，介于关系的后端服务器和客户前端工具之间。使用关系的或扩充关系的 DBMS 存储并管理数据仓库，而 OLAP 中间件支持其余部分。

多维联机分析处理服务器（MOLAP）由基于数组的多维存储引擎支持数据的多维式图，将多维视图直接映射到数据立方体数组结构。

混合联机分析处理服务器（HOLAP）结合了 ROLAP 和 MOLAP 技术，得益于 ROLAP 较大的可伸缩性和 MOLAP 的快速计算。HOLAP 允许将大量的详细数据存放在关系数据库中，而聚集保持在分离的 MOLAP 存储中。

(b) 对每种技术，解释如下函数如何实现：

i. 数据仓库的产生（包括聚集）。

火龙果•整理
uml.org.cn

ROLAP：采用一个 OLAP 服务器，数据仓库的产生可以由一个使用汇总实事表的关系型或扩展的关系型数据库管理系统（DBMS）实现，这个实事表可以存储给定立方体的聚集数据和由给定数据立方体的模式连接键指出的抽象级中的数据

MOLAP：为了产生一个数据仓库，MOLAP 技术使用多维数组结构来存储数据和多路数组聚集来计算数据立方体。

HOLAP：HOLAP 技术的特色是应用一个关系数据库来存储数据和一些低层次的聚集，并且应用一个 MOLAP 来存储高层次的聚集。

ii. 上卷。

ROLAP：沿一维用汇总的实事表上卷，我们在包含期望维的一个泛化的表中寻找记录。例如，从日到月上卷数据维，其中选择那些包含所有特定值的区间的日期的记录。记录中给定的测量域的值，例如以美元计的销售量，将会包含用于期望的上卷的部分和。

MOLAP：在数据立方体中执行上卷，为得到期望的维直接攀爬到相应的概念层。例如，沿着地点维从城市上卷的国家，得到更泛化的数据。

HOLAP：使用 HOLAP 技术进行上卷的方法将会与 ROLAP 或 MOLAP 相似，主要体现在依靠在相应维的执行中所使用的技术。

iii. 下钻。

ROLAP：我们使用汇总的事实表沿某一维下钻，是为了寻找表中的记录，这个表包含对期望维的一个泛化。例如，沿地点维从国家到省或者州下钻，选择下一概念层能最低区域的记录，即对应包含所有特定值的地点。在这种情况下，城市域可能包含所有的值。这个记录中给定的测量域的值将会包含下钻所期望得到求和。例如 dollars_sold。

MOLAP：为了在数据立方体中执行下钻操作，直接降低概念分层以得到期望的维。例如，你可以沿日期维从月份下钻到每日，用日聚集数据而不是用月。

HOLAP：使用 HOLAP 技术进行下钻的方法将会与 ROLAP 或 MOLAP 相似，主要体现在依靠在相应维的执行中所使用的技术。

iv. 增量更新。

火龙果•整理
uml.org.cn

OLAP：为了执行增量更新，先检查是否相应的元组在汇总的事实表中。如若不是，则插入元组到汇总的事实表中，并且向上传播结果。否则，更新元组值，并且也向上传播结果。

MOLAP：为了执行增量更新，先检查是否相应的元胞处在 MOLAP 立方体中。如若不是，则插入元胞到该立方体，并向上传播结果。否则，更新该元胞，并且向上传播结果。

HOLAP：与 ROLAP 或 MOLAP 相似，主要体现在依靠在相应维的执行中所使用的技术。

(c) 你喜欢那种实现技术？为什么？

HOLAP 经常被优先采用，因为它兼有 ROLAP 和 MOLAP 方法的优点，并且避免了它们的缺点。如果立方体是很稠密的，应该选择 MOLAP 方法。如果数据是稀疏的，且维比较高，那将会有许多元胞（服从指数增长），这种情况下，往往希望计算冰山立方体，而不是对立方体完全物化。

3.6 2009-12-8

3.7 2009-12-07

火龙果•整理
uml.org.cn

第4章　数据立方体计算与数据泛化

4.1 假定 10 维基本方体只包含 3 个基本单元：(1) $(a_1, d_2, d_3, d_4, \cdots, d_9, d_{10})$，(2) $(d_1, b_2, d_3, d_4, \cdots, d_9, d_{10})$和(3) $(d_1, b_2, c_3, d_4, \cdots, d_9, d_{10})$，其中$a_1 \neq d_1$，$b_2 \neq d_2$并且$c_3 \neq d_3$。该立方体的度量是*count*。

(a) 完全数据立方体中包含多少个非空方体？

(b) 完全立方体中包含多少个非空聚集（非基本）单元？

(c) 如果冰山立方体的条件是“$count \geq 2$”，冰山立方体包含多少个非空聚集单元？

(d) 单元 *c* 是闭单元，如果不存在单元 *d* 使得 *d* 是单元 *c* 的特殊化（即 *d* 通过用非“*”值替换 *c* 中的“*”得到），并且 *d* 与 *c* 具有相同的度量值。闭立方体是仅由闭单元组成的数据立方体。该完全立方体中有多少个闭单元？

解答：

(a) 完全数据立方体中包含多少个非空方体？

2^{10}个。

(b) 完全立方体中包含多少个非空聚集（非基本）单元？

(1) 每一个单元可以产生$2^{10}-1$个非空聚集的单元，这样包括需消除的重叠，我们总共有 $3\times2^{10}-3$ 个单元。

(2) 我们有3×2^7个只交叠一次（共记了 2 次数）的单元，1×2^7（即$(*, *, *, d_4, \cdots, d_9, d_{10})$）个交叠两次（共记了 3 次数）的单元，这样我们应去除总数达 5×2^7个交叠的单元。

(3) 这样就有：$3\times2^{10}-5\times2^7-3=19\times2^7-3$ 个非空聚集单元。

(c) 如果冰山立方体的条件是“$count \geq 2$”，冰山立方体包含多少个非空聚集单元？

(1) $(*, *, d_3, d_4, \cdots, d_9, d_{10})$有 2 个计数，因为它是由单元 1 和单元 2 产生的；

(2) $(*, d_2, *, d_4, \cdots, d_9, d_{10})$有 2 个计数，因为它是由单元 1 和单元 2 产生的；

(3) $(d_1, *, *, d_4, \cdots, d_9, d_{10})$ 有 2 个计数，因为它是由单元 1 和单元 2 产生的；

(4) $(*, *, *, d_4, \cdots, d_9, d_{10})$有 3 个计数，因为它是由单元 1、单元 2 和单元 3

火龙果•整理
uml.org.cn

产生的。

(d) 单元 c 是闭单元，如果不存在单元 d 使得 d 是单元 c 的特殊化（即 d 通过用非“*”值替换 c 中的“*”得到），并且 d 与 c 具有相同的度量值。闭立方体是仅由闭单元组成的的数据立方体。该完全立方体有多少闭单元？

共有 7 个闭单元，即

(1) $(a_1, d_2, d_3, d_4, \cdots, d_9, d_{10})$：1；

(2) $(d_1, b_2, d_3, d_4, \cdots, d_9, d_{10})$：1；

(3) $(d_1, d_2, c_3, d_4, \cdots, d_9, d_{10})$：1；

(4) $(*, *, d_3, d_4, \cdots, d_9, d_{10})$：2；

(5) $(*, d_2, *, d_4, \cdots, d_9, d_{10})$：2；

(6) $(d_1, *, *, d_4, \cdots, d_9, d_{10})$：2；

(7) $(*, *, *, d_4, \cdots, d_9, d_{10})$：3。

4.2 有几种典型的立方体计算方法，如多路数组计算（MultiWay）[ZDN97]、BUC（自底向上计算）[BR99]和 Star-Cubing[XHL W03]。简单地描述这三种方法（即用一两行列出要点），并在以下条件下比较它们的可行性和性能。

(a) 计算低维（例如，小于 8 维）、稠密的完全立方体。

(b) 计算具有高度倾斜数据分布的大约 10 位的冰山立方体。

(c) 计算高维（例如，超过 100 维）、稀疏冰山立方体。

解答：

注意到本教科书采用了一种将数据立方体视为方体的格的实用的世界观。这里，在方体的格中从顶点方体向下下钻。

Multiway：自底向上的、同时数组聚集、共享预计算结果、并且最小化存储需求。

BUC：自顶向下的、递归的划分和探查、共享排序。

Star-Cubing：使用星形树综合自顶向下和自底向上的计算模型，使 Apriori 剪枝和计算同时进行。

(a) 计算低维（例如，小于 8 维）、稠密的完全立方体。

MiltiWay 和 Star-Cubing 都比 BUC 好用。

(b) 计算具有高度倾斜数据分布的大约 10 位的冰山立方体。

火龙果•整理
uml.org.cn

MultiWay 能计算冰山立方体。对高度扭曲的数据集，Star-Cubing 比 BUC 好用。

(c)　计算高维（例如，超过 100 维）、稀疏冰山立方体。

MultiWay 能用于计算冰山立方体。BUC 和 Star-Cubing 计算高维数据均很有效。闭立方体和外壳片段方法也可被试用。

4.3 4.4 假定基本立方体有三维 A，B，C，其单元数如下：|A|=1 000 000，|B|=100，|C|=1 000。假定每维均等的分块成 10 部分。

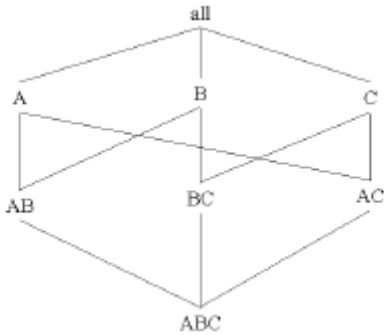
(a) 假定每维只有一层，画出完整立方体的格。

(b) 如果每个立方体单元存放一个 4 字节的度量，若立方体是稠密的，所计算的立方体有多大？

(c) 指出空间需求量最小的立方体中的块计算次序，并计算 2-D 平面计算所需的内存空间总量。

解答：

(a) 假定每维只有一层，画出完整立方体的格。



题 4.4 图 练习 4.4 中立方体的完全的格

(b) 如果每个立方体单元存放一个 4 字节的度量，若立方体是稠密的，所计算的立方体有多大？

所计算的立方体的总量如下：

0-D 顶点方体：all:1

1-D 一维方体：A:1,000,000; B:100; C1,000; 小计：1,001,100

1-D 二维方体：AB:100,000,000; BC:100,000; AC:1,000,000,000; 小计：1,100,100,000

3-D 三维方体：ABC:100,000,000,000

火龙果•整理
uml.org.cn

存储量的总数为：101,101,101,101 cells×4 bytes=404,404,404,404 byes

(c) 指出空间需求量最小的立方体中的块计算次序，并计算 2-D 平面计算所需的内存空间总量。

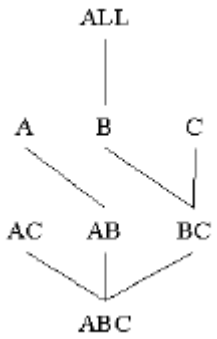
所需最少存储空间的计算顺序是 B-C-A。得到二维数据立方体的存储单元数为

AC：1 000 000×1 000

AB：1 000 000×100

BC：100×1 000

显然由二维立方体 AB 对 B 维聚集得到一维立方体 A 计算量最少，由二维立方体 BC 对 C 维聚集得到一维立方体 B 计算量最少，二维立方体 BC 对 B 维聚集得到一维立方体 C 计算量最少；再对一维立方体 B 聚集得到 all 计算量最少。如题 4.4 图所示。



题 4.4 图 练习 4.4 中所需最小存储空间对应的计算次序

计算所需的存储空间总量为：

Tolal space=(100×1000)+(100,000×100)+(100×100,000)

=20,100,000 cells=80,400,000 bytes 原版错误

“Tolal space=(100×1000)+(1,000,000×10)+(100×10,000)

=20,100,000 cells=80,400,000 bytes”

实际上是公式错误，结果正确。错误公式但计算正确的值是 11,100,000。

4.4 4.12 考虑下面的多特征立方体查询：按{item，region，month}的所有子集分组，对每组找出 2004 年的最小货架寿命，并对价格低于 100 美元、货架寿命在最小货架寿命的 1.25~1.5 倍之间的元组找出总销售额部分。

(a) 画出该查询的多特征立方体图。

火龙果•整理
uml.org.cn

(b) 用扩充的 SQL 表示该查询。

(c) 这是一个分布式多特征立方体吗？为什么？

解答：

(a) 画出该查询的多特征立方体图。

R0→R1(≥1.25*min(shelf)and≤1.5*min(shelf))

(b) 用扩充的 SQL 表示该查询。

select item, region, month, Min(shelf), SUM(R1)

from Purchase

where year=2004

cube by item, region, month: R1

such that R1.shelf≥1.25*MIN(Shelf) and (R1.Shelf≤1.5*MIN(Shelf) and R1.Price<100

(c) 这是一个分布式多特征立方体吗？为什么？

这不是一个分布多特征立方体，因为在“such that”语句中采用了“≤”条件。

4.5 2008-11-29

4.6 2008-11-29

火龙果•整理
uml.org.cn

第5章　挖掘频繁模式、关联和相关

5.1 Apriori 算法使用子集支持度性质的先验知识。

5.2 5.2.2 节介绍了由频繁项集产生关联规则的方法。提出了一个更有效的方法。解释它为什么比 5.2.2 节的方法更有效。（提示：考虑将习题 5.1(b)和习题 5.1(c)的性质结合到你的设计中。）

■

5.3 数据库有 5 个事物。设 min_sup=60%，min_conf=80。

TID	购买的商品
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y}
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I, E}

(a) 分别使用 Apriori 和 FP 增长算法找出所有的频繁项集。比较两种挖掘过程的效率。

(b) 列举所有与下面的元规则匹配的强关联规则（给出支持度 s 和置信度 c），其中，X 是代表顾客的变量，item 是表示项的变量（如“A”、“B”等）：

$\forall x \in transaction, buys(X, item_1) \wedge buys(X, item_2) \Rightarrow buys(X, item_3)$ [s, c]

解答：

(a) 分别使用 Apriori 和 FP 增长算法找出所有的频繁项集。比较两种挖掘过程的效率。

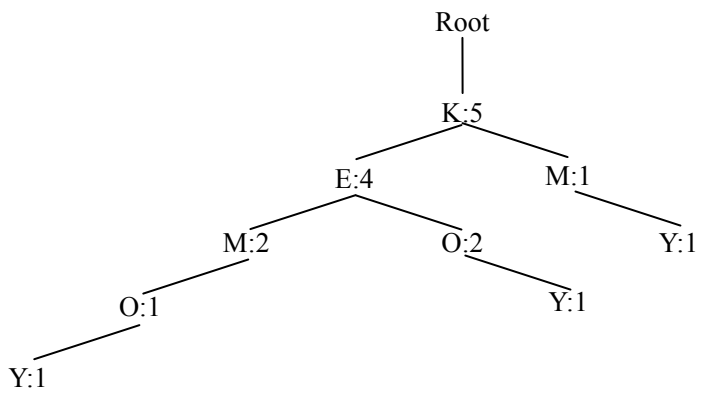
Apriori 算法：由于只有 5 次购买事件，所以绝对支持度是 5×min_sup=3。

火龙果•整理
uml.org.cn

$$C_1=\begin{bmatrix} M & 3 \\ O & 3 \\ N & 2 \\ K & 5 \\ E & 4 \\ Y & 3 \\ D & 1 \\ A & 1 \\ U & 1 \\ C & 2 \\ I & 1 \end{bmatrix} \quad L_1=\begin{bmatrix} M & 3 \\ O & 3 \\ K & 5 \\ E & 4 \\ Y & 3 \end{bmatrix} \quad C_2=\begin{bmatrix} MO & 1 \\ MK & 3 \\ ME & 2 \\ MY & 2 \\ OK & 3 \\ OE & 3 \\ OY & 2 \\ KE & 4 \\ KY & 3 \\ EY & 2 \end{bmatrix} \quad L_2=\begin{bmatrix} MK & 3 \\ OK & 3 \\ OE & 3 \\ KE & 4 \\ KY & 3 \end{bmatrix} \quad C_3=\begin{bmatrix} OKE & 3 \\ KEY & 2 \end{bmatrix}$$

$$L_3=\begin{bmatrix} OKE & 3 \end{bmatrix}$$

FP-growth：数据库的第一次扫描与Apriori算法相同，得到L_1。再按支持度计数的递减序排序，得到：L={(K:5), (E:4), (M:3), (O:3), (Y:3)}。扫描没个事务，按以上L的排序，从根节点开始，得到FP-树。



题 5.3 图 FP 增长算法

项	条件模式基	条件 FP 树	产生的频繁模式
Y	{{K,E,M,O:1}，{K,E,O:1}，{K,M:1}}	K:3	{K,Y:3}
O	{{K,E,M:1}，{K,E:2}}	K:3，E:3	{K,O:3}，{E,O:3}，{K,E,O:3}
M	{{K,E:2}，{K:1}}	K:3	{K,M:3}
E	{{K:4}}	K:4	{K,E:4}

火龙果•整理
uml.org.cn

效率比较：Apriori 算法的计算过程必须对数据库作多次扫描，而 FP-增长算法在构造过程中只需扫描一次数据库，再加上初始时为确定支持度递减排序的一次扫描，共计只需两次扫描。由于在 Apriori 算法中的自身连接过程产生候选项集，候选项集产生的计算代价非常高，而 FP-增长算法不需产生任何候选项。

(b)列举所有与下面的的元规则匹配的强关联规则（给出支持度 s 和置信度 c），其中，X 是代表顾客的变量，item 是表示项的变量（如“A”、“B”等）：

∀x∈transaction, buys(X, “K”)∧buys(X, “O”)⇒buys(X, “E”) [s=0.6, c=1]

∀x∈transaction, buys(X, “E”)∧buys(X, “E”)⇒buys(X, “K”) [s=0.6, c=1]

或也可表示为

K,O→E[s(support)=0.6 或 60%,c(confidence)=1 或 100%]

E,O→K[s(support)=0.6 或 60%,c(confidence)=1 或 100%]

■

5.4 （**实现项目**）使用你熟悉的程序设计语言（如 C++或 Java），实现本章介绍的三种频繁项集挖掘算法：

5.5 数据库有 4 个事物。设*min_sup*=60%，*min_conf*=80%。

cust_ID	*TID*	*items_bought* (in the form of *brand-item_category*)
01	T100	{King's-Crab, Sunset-Milk, Dairyland-Cheese, Best-Bread}
02	T200	{Best-Cheese, Dairyland-Milk, Goldenfarm-Apple, Tasty-Pie, Wonder-Bread}
01	T300	{Westcoast-Apple, Dairyland-Milk, Wonder-Bread, Tasty-Pie}
03	T400	{Wonder-Bread, Sunset-Milk, Dairyland-Cheese}

(c)在item_category粒度（例如，$item_i$可以是“Milk”），对于下面的规则模版

$$\forall X \in transaction, buys(X, item_1) \wedge buys(X, item_2) \Rightarrow buys(X, item_3) \quad [s, c]$$

对于最大的 k，列出频繁 k 项集和包含最大的 k 的频繁 k 项集的所有强关联规则（包括它们的支持度 s 和置信度 c）。

(d)在brand-item_category粒度（例如，$item_i$可以是“Sunset-Milk”），对于下面的规则模版

$$\forall X \in customer, buys(X, item_1) \wedge buys(X, item_2) \Rightarrow buys(X, item_3) \quad [s, c]$$

对于最大的 k 列出频繁 k 项集（但不输出任何规则）。

解答：

(a)在*item_category*粒度（例如，$item_i$可以是“*Milk*”），对于下面的规则模版

火龙果•整理
uml.org.cn

$\forall X \in transaction, buys(X, item_1) \wedge buys(X, item_2) \Rightarrow buys(X, item_3)$ [s,c]

对于最大的 k，列出频繁 k 项集和包含最大的 k 的频繁 k 项集的所有强关联规则（包括它们的支持度 s 和置信度 c）。

k=3 并且频繁 3 项集是{Bread,Milk,Cheese}。有如下强关联规则

$Bread \wedge Cheese \Rightarrow Milk$, [75%,75%] 原版错误“[75%，100%]”

$Cheese \wedge Milk \Rightarrow Bread$, [75%,100%]

$Cheese \Rightarrow Milk \wedge Bread$, [75%,100%]

(b)在*brand-item_category*粒度（例如，$item_i$可以是“*Sunset-Milk*”），对于下面的规则模版

$\forall X \in customer, buys(X, item_1) \wedge buys(X, item_2) \Rightarrow buys(X, item_3)$ [s,c]

对于最大的 k 列出频繁 k 项集（但不输出任何规则）。

k=3 并且这些频繁 3 项集是{(*Wonder-Bread*, *Dairyland-Milk*, *Tasty-Pie*: s=2/3>60%), (*Wonder-Bread*, *Sunset-Milk*, *Dairyland-Cheese*: s=2/3>60%)}。

5.6 2009-01-09

5.7 2009-12-10

火龙果•整理
uml.org.cn

第6章　分类和预测

6.1 简述决策树分类的主要步骤。

6.2 6.4 计算决策树算法在最坏情况下的计算复杂度是重要的。给定数据集 D，具有 n 个属性和$|D|$个元组，证明决策树生长的计算时间最多为 $n\times|D|\times log(|D|)$。

解答：

最不利的情况发生在对元组群的分类前不得不应用尽可能多的属性。树的最大深度是 $log(|D|)$。在每一层我们必须计算属性选择度量 $O(n)$次（每一个属性）。在每层上的元组总数是$|D|$（在所有的划分上增加）。这样，每层树的计算量是 $O(n\times|D|)$。合计所有层得到 $O(n\times|D|\times log(|D|))$。

6.3 6.11 下表由雇员数据库的训练数据组成。数据已泛化。例如，age“31…35”表示年龄在 31~35 之间。对于给定的行，count 表示 department，status，age 和 salary 在该行具有给定值的元组数。

department	status	age	salary	count
sales	senior	31…35	46K…50K	30
sales	junior	26…30	26K…30K	40
sales	junior	31…35	31K…35K	40
systems	junior	21…25	46K…50K	20
systems	senior	31…35	66K…70K	5
systems	junior	26…30	46K…50K	3
systems	senior	41…45	66K…70K	3
marketing	senior	36…40	46K…50K	10
marketing	junior	31…35	41K…45K	4
secretary	senior	46…50	36K…40K	4
secretary	junior	26…30	26K…30K	6

d)　　如何修改基本决策树算法，以便考虑每个广义数据元组（即每一行）

火龙果•整理
uml.org.cn

的 count？

e) 使用修改过的算法，构造给定数据的决策树。

f) 给定一个数据元组，它的属性 department，age 和 salary 的值分别为“systems”，“26…30”，和“46K…50K”。该元组 status 的朴素贝叶斯分类是什么？

g) 为给定的数据设计一个多层前馈神经网络。标记输入和输出层节点。

h) 使用上面得到的多层前馈神经网络，给定训练实例（sales，senior，31…35，46K…50K），给出后向传播算法一次迭代后的权重值。指出你使用的初始权重和偏倚以及学习率。

解答：

(a) 如何修改基本决策树算法，以便考虑每个广义数据元组（即每一行）的 count？

(b) 使用修改过的算法，构造给定数据的决策树。

(c) 给定一个数据元组，它的属性 department，age 和 salary 的值分别为“systems”，“26…30”，和“46K…50K”。该元组 status 的朴素贝叶斯分类是什么？

解一：设元组的各个属性之间相互独立，所以先求每个属性的类条件概率：

P(systems|junior)=(20+3)/(40+40+20+3+4+6)=23/113;

P(26-30|junior)=(40+3+6)/113=49/113；

P(46K-50K|junior)=(20+3)/113=23/113；

∵ X=(department=system,age=26…30,salary=46K…50K)；

∴ P(X|junior)=P(systems|junior)P(26-30|junior)P(46K-50K|junior)

$=23\times49\times23/113^3=25921/1442897=0.01796$；

P(systems|senior)=(5+3)/(30+5+3+10+4)=23/52;

P(26-30|senior)=(0)/53=0;

P(46K-50K|senior)=(30+10)/52=40/52；

∵ X=(department=system,age=26…30,salary=46K…50K)；

∴ P(X|senior)=P(systems|senior)P(26-30|senior)P(46K-50K|senior)=0;