第三章 数据预处理

笔记本：	数据挖掘：概念与技术
创建时间：	2017/12/20 15:42	更新时间：	2017/12/26 11:27
作者：	Passero

3.1数据预处理：概述

数据清理可以用来清除数据中的噪声，纠正不一致。数据集成将数据由多个数据源合并成一个一致的数据存储，如数据仓库。数据归约可以通过如聚集，删除冗余特征或聚类来降低数据的规模。数据变换（如规范化）可以用来把数据压缩到较小的区间，如0.0到1.1。

数据质量：为什么要对数据预处理

数据质量涉及很多因素，包括准确性，完整性，一致性，时效性，可信性和可解释性。

数据质量的三个要素：准确性，完整性和一致性

时效性也影响数据质量

可信性反应有多少数据是用户信赖的

可解释性反映数据是否容易理解

数据预处理的主要任务

数据预处理的主要步骤，即数据清理，数据集成，数据归约和数据变换

数据清理例程通过填写缺失的值，光滑噪声数据，识别或删除离群点，并解决不一致性来清理数据。

数据集成即集成多个数据库，数据立方体或文件
PS：通常，在为数据仓库准备数据时，数据清理和集成将作为预处理步骤进行。还可以再次进行数据清理，检测和删去可能由集成导致的冗余。

数据归约得到数据集的简化表示，它小得多，但能够产生同样或几乎同样的分析结果。数据归约策略包括维归约和数值归约。
[---在维归约中，使用数据编码方案以得到原始数据的简化或压缩表示。例子包括数据压缩技术，属性子集选择和属性构造。
 ---在数值归约中，使用参数模型或非参数模型，用较小的表示取代数据。]

规范化，数据离散化和概念分层产生都是某种形式的数据变换。

PS：以上的分类并不是互斥的。例如，冗余数据的删除既是一种数据清理形式，也是一种数据归约。

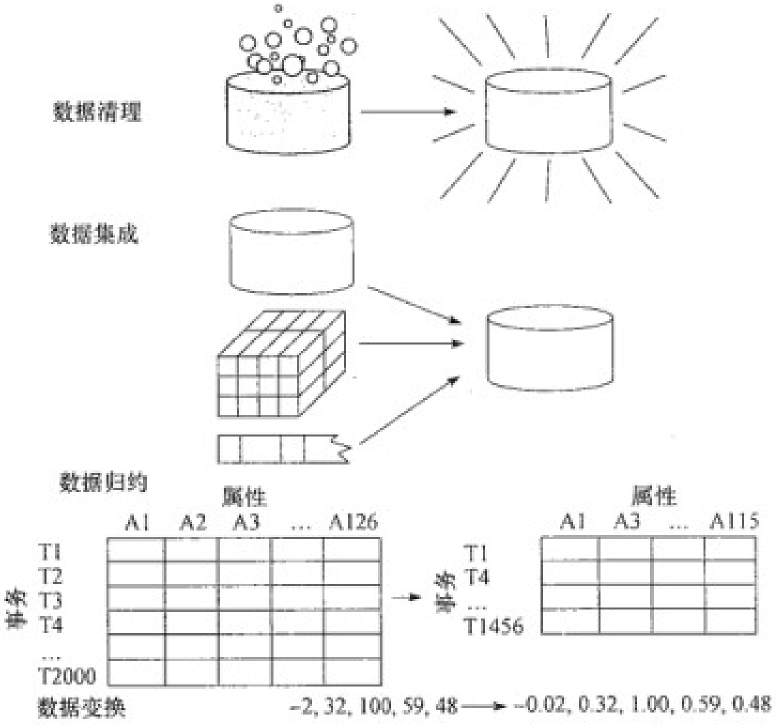

图 3.1　数据预处理的形式

3.2数据清理

缺失值

（1）**忽略元组**：当缺少类标号时通常这样做（假定挖掘任务涉及分类）。除非元组有多个属性缺少值，否则该方法不是很有效。当每个属性缺失值的百分比变化很大时，它的性能特别差。采用忽略元组，你不能使用该元组的剩余属性值。这些数据可能对手头的任务是有用的。

（2）**人工填写缺失值**：一般来说，该方法很费时，并且当数据集很大、缺失很多值时，该方法可能行不通。

（3）**使用一个全局常量填充缺失值**：将缺失的属性值用同一个常量（如“*Unknown*”或 $-\infty$）替换。如果缺失的值都用如“*Unknown*”替换，则挖掘程序可能误以为它们形成了一个有趣的概念，因为它们都具有相同的值——“*Unknown*”。因此，尽管该方法简单，但是并不十分可靠。

（4）**使用属性的中心度量（如均值或中位数）填充缺失值**：第 2 章讨论了中心趋势度量，它们指示数据分布的“中间”值。对于正常的（对称的）数据分布而言，可以使用均值，而倾斜数据分布应该使用中位数（2.2 节）。例如，假定 AllElectronics 的顾客收入的数据分布是对称的，并且平均收入为 56 000 美元，则使用该值替换 *income* 中的缺失值。

（5）**使用与给定元组属同一类的所有样本的属性均值或中位数**：例如，如果将顾客按 *credit_risk* 分类，则用具有相同信用风险的顾客的平均收入替换 *income* 中的缺失值。如果给定类的数据分布是倾斜的，则中位数是更好的选择。

（6）**使用最可能的值填充缺失值**：可以用回归、使用贝叶斯形式化方法的基于推理的工具或决策树归纳确定。例如，利用数据集中其他顾客的属性，可以构造一棵决策树，来预测 *income* 的缺失值。决策树和贝叶斯推理分别在第 8 章和第 9 章详细介绍，而回归在 3.4.5 节介绍。

方法（3）~方法（6）使数据有偏，填入的值可能不正确。然而，方法（6）是最流行的策略。与其他方法相比，它使用已有数据的大部分信息来预测缺失值。在估计 *income* 的缺失值时，通过考虑其他属性的值，有更大的机会保持 *income* 和其他属性之间的联系。

噪声数据

噪声是被测量的变量的随机误差或方差。

---分箱

分箱（binning）：分箱方法通过考察数据的“近邻”（即周围的值）来光滑有序数据值。这些有序的值被分布到一些“桶”或箱中。由于分箱方法考察近邻的值，因此它进行*局部*光滑。图 3.2 表示了一些分箱技术。在该例中，*price* 数据首先排序并被划分到大小为 3 的等频的箱中（即每个箱包含 3 个值）。对于**用箱均值光滑**，箱中每一个值都被替换为箱中的均值。例如，箱 1 中的值 4、8 和 15 的均值是 9。因此，该箱中的每一个值都被替换为 9。

类似地，可以使用**用箱中位数光滑**，此时，箱中的每一个值都被替换为该箱的中位数。对于**用箱边界光滑**，给定箱中的最大和最小值同样被视为*箱边界*，而箱中的每一个值都被替换为最近的边界值。一般而言，宽度越大，光滑效果越明显。箱也可以是*等宽的*，其中每个箱值的区间范围是常量。分箱也可以作为一种离散化技术使用，将在 3.5 节进一步讨论。

按 *price*（美元）排序后的数据：4, 8, 15, 21, 21, 24, 25, 28, 34

划分为（等频的）箱：

箱1：4, 8, 15
箱2：21, 21, 24
箱3：25, 28, 34

用箱均值光滑：

箱1：9, 9, 9
箱2：22, 22, 22
箱3：29, 29, 29

用箱边界光滑：

箱1：4, 4, 15
箱2：21, 21, 24
箱3：25, 25, 34

图 3.2　数据光滑的分箱方法

---回归：可以用一个函数拟合数据来光滑数据，这种技术成为回归。线性回归设计找出拟合两个属性或变量的最佳直线，使得一个属性可以用来预测另一个。多元线性回归是线性回归的扩充，其中涉及的属性多于两个，并且数据拟合到一个多维曲面。

---离群点分析：可以通过聚类来检测离群点。聚类将类似的值组织成群或簇。直观地说，落在簇集合之外的值被视为离群点。

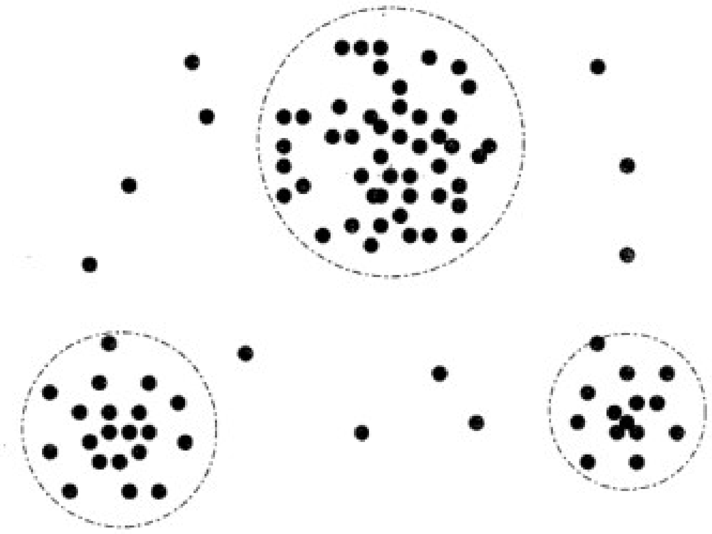

图 3.3　顾客在城市中的位置的 2-D 图，显示了 3 个数据簇。
可以将离群点看做落在簇集合之外的值来检测

数据清理作为一个过程
---偏差检测：元数据，字段过载，唯一性规则，连续性规则，空值规则，数据清洗工具，数据审计工具
---数据变换：数据迁移工具，ETL工具

3.3数据集成（未仔细看）

实体识别问题：来自多个信息源的现实世界的等价实体如何才能匹配 e.g.一个数据库的属性和另一个数据库的属性匹配时，如何确信这指的是相同的属性

冗余和相关分析
一个属性如果能由另一个或另一组属性导出，则这个属性可能是冗余的。
---标称数据可以使用卡方检验

1. 标称数据的χ^2相关检验

对于标称数据，两个属性A和B之间的相关联系可以通过χ^2（卡方）检验发现。假设A有c个不同值$a_1, a_2, \cdots, a_c$，B有r个不同值$b_1, b_2, \cdots, b_r$。用A和B描述的数据元组可以用一个**相依表**显示，其中A的c个值构成列，B的r个值构成行。令(A_i, B_j)表示属性A取值a_i、属性B取值b_j的联合事件，即$(A=a_i, B=b_j)$。每个可能的(A_i, B_j)联合事件都在表中有自己的单元。χ^2值（又称Pearson χ^2统计量）可以用下式计算：

$$\chi^2 = \sum_{i=1}^{c}\sum_{j=1}^{r}\frac{(o_{ij}-e_{ij})^2}{e_{ij}} \tag{3.1}$$

其中，o_{ij}是联合事件(A_i, B_j)的观测频度（即实际计数），而e_{ij}是(A_i, B_j)的期望频度，可以用下式计算：

$$e_{ij} = \frac{count(A=a_i)\times count(B=b_j)}{n} \tag{3.2}$$

其中，n是数据元组的个数，$count(A=a_i)$是A上具有值a_i的元组个数，而$count(B=b_j)$是B上具有值b_j的元组个数。(3.1)式中的和在所有$r\times c$个单元上计算。注意，对χ^2值贡献最大的单元是其实际计数与期望计数很不相同的单元。

χ^2统计检验假设A和B是独立的。检验基于显著水平，具有自由度$(r-1)\times(c-1)$。我们将用例3.1解释该统计量的使用。如果可以拒绝该假设，则我们说A和B是统计相关的。

---数值属性使用相关系数和协方差

2. 数值数据的相关系数

对于数值数据，我们可以通过计算属性A和B的**相关系数**（又称**Pearson积矩系数**，Pearson's product moment coefficient），用发明者Karl Pearson的名字命名），估计这两个属性的相关度$r_{A,B}$，

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_ib_i)-n\bar{A}\bar{B}}{n\sigma_A\sigma_B} \tag{3.3}$$

其中，n是元组的个数，a_i和b_i分别是元组i在A和B上的值，$\bar{A}$和$\bar{B}$分别是A和B的均值，σ_A和σ_B分别是A和B的标准差（在2.2.2节定义），而$\sum(a_ib_i)$是AB叉积和（即对于每个元组，A的值乘以该元组B的值）。注意，$-1\leqslant r_{A,B}\leqslant +1$。如果$r_{A,B}$大于0，则$A$和$B$是正相关的，这意味着$A$值随$B$值的增加而增加。该值越大，相关性越强（即每个属性蕴涵另一个的可能性越大）。因此，一个较高的$r_{A,B}$值表明A（或B）可以作为冗余而被删除。

如果该结果值等于0，则A和B是独立的，并且它们之间不存在相关性。如果该结果值小于0，则A和B是负相关的，一个值随另一个减少而增加。这意味着每一个属性都阻止另一个出现。散点图也可以用来观察属性之间的相关性（2.2.3节）。例如，图2.8的散点图分别显示了正相关和负相关数据，而图2.9显示了不相关数据。

注意，相关性并不蕴涵因果关系。也就是说，如果A和B是相关的，这并不意味着A导致B或B导致A。例如，在分析人口统计数据库时，我们可能发现一个地区的医院数与汽车盗窃数是相关的。这并不意味一个导致另一个。实际上，二者必然地关联到第三个属性——人口。

3. 数值数据的协方差

在概率论与统计学中，协方差和方差是两个类似的度量，评估两个属性如何一起变化。考虑两个数值属性 A、B 和 n 次观测的集合 $\{(a_1, b_1), \cdots, (a_n, b_n)\}$。$A$ 和 B 的均值又分别称为 A 和 B 的**期望值**，即

$$E(A) = \bar{A} = \frac{\sum_{i=1}^{n} a_i}{n}$$

且

$$E(B) = \bar{B} = \frac{\sum_{i=1}^{n} b_i}{n}$$

A 和 B 的**协方差**（covariance）定义为

$$Cov(A,B) = E((A - \bar{A})(B - \bar{B})) = \frac{\sum_{i=1}^{n} (a_i - \bar{A})(b_i - \bar{B})}{n} \tag{3.4}$$

如果我们把 $r_{A,B}$（协相关系数）的（3.3）式与（3.4）式相比较，则我们看到

$$r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B} \tag{3.5}$$

其中，σ_A 和 σ_B 分别是 A 和 B 的标准差。还可以证明

$$Cov(A,B) = E(A \cdot B) - \bar{A}\bar{B} \tag{3.6}$$

该式可以简化计算。

对于两个趋向于一起改变的属性 A 和 B，如果 A 大于 $\bar{A}$（A 的期望值），则 B 很可能大于 $\bar{B}$（B 的期望值）。因此，A 和 B 的协方差为正。另一方面，如果当一个属性小于它的期望值时，另一个属性趋向于大于它的期望值，则 A 和 B 的协方差为负。

如果 A 和 B 是*独立的*（即它们不具有相关性），则 $E(A \cdot B) = E(A) \cdot E(B)$。因此，协方差为 $Cov(A, B) = E(A \cdot B) - \bar{A}\bar{B} = E(A) \cdot E(B) - \bar{A}\bar{B} = 0$。然而，其逆不成立。某些随机变量（属性）对可能具有协方差 0，但是不是独立的。仅在某种附加的假设下（如数据遵守多元正态分布），协方差 0 蕴涵独立性。

元组重复

除了检测属性间的冗余外，还应当在元组级检测重复（例如，对于给定的唯一数据实体，存在两个或多个相同的元组）。去规范化表（denormalized table）的使用（这样做通常是通过避免连接来改善性能）是数据冗余的另一个来源。不一致通常出现在各种不同的副本之间，由于不正确的数据输入，或者由于更新了数据的某些出现，但未更新所有的出现。例如，如果订单数据库包含订货人的姓名和地址属性，而不是这些信息在订货人数据库中的码，则差异就可能出现，如同一订货人的名字可能以不同的地址出现在订单数据库中。

数据值冲突的检测与处理

数据集成还涉及*数据值冲突的检测与处理*。例如，对于现实世界的同一实体，来自不同数据源的属性值可能不同。这可能是因为表示、尺度或编码不同。例如，*重量*属性可能在一个系统中以公制单位存放，而在另一个系统中以英制单位存放。对于连锁旅馆，不同城市的房价不仅可能涉及不同的货币，而且可能涉及不同的服务（如免费早餐）和税收。例如，不同学校交换信息时，每个学校可能都有自己的课程计划和评分方案。一所大学可能采取学季制，开设 3 门数据库系统课程，用 A + ~ F 评分；而另一所大学可能采用学期制，开设两门数据库课程，用 1 ~ 10 评分。很难在这两所大学之间制定精确的课程成绩变换规则，这使得信息交换非常困难。

3.4数据归约（未仔细看）

数据归约策略概述

---维归约：减少所考虑的随机变量或属性的个数

---数量归约：用替代的，较小的数据表示形式替换原数据

---数据压缩：使用变换，以便得到原数据的归约或压缩表示[无损，有损]

小波变换（未仔细看）

离散小波变换（DWT）是一种线性信号处理技术，用于数据向量 $\mathbf{X}$ 时，将它变换成不同的数值**小波系数**向量 $\mathbf{X}'$。两个向量具有相同的长度。当这种技术用于数据归约时，每个元组看做一个 n 维数据向量，即 $\mathbf{X}=(x_1, x_2, \cdots, x_n)$，描述 n 个数据库属性在元组上的 n 个测量值㊀。

"如果小波变换后的数据与原数据的长度相等，这种技术如何能够用于数据压缩？"关键在于小波变换后的数据可以截短。仅存放一小部分最强的小波系数，就能保留近似的压缩数据。例如，保留大于用户设定的某个阈值的所有小波系数，其他系数置为 0。这样，结果数据表示非常稀疏，使得如果在小波空间进行计算的话，利用数据稀疏特点的操作计算得非常快。该技术也能用于消除噪声，而不会光滑掉数据的主要特征，使得它们也能有效地用于数据清理。给定一组系数，使用所用的 DWT 的逆，可以构造原数据的近似。

DWT 与离散傅里叶变换（DFT）有密切关系。DFT 是一种涉及正弦和余弦的信号处理技术。然而，一般地说，DWT 是一种更好的有损压缩。也就是说，对于给定的数据向量，如果 DWT 和 DFT 保留相同数目的系数，则 DWT 将提供原数据更准确的近似。因此，对于相同的近似，DWT 需要的空间比 DFT 小。与 DFT 不同，小波空间局部性相当好，有助于保留局部细节。

主成分分析

假设待归约的数据由用 n 个属性或维描述的元组或数据向量组成。**主成分分析**（principal components analysis）或 **PCA**（又称 Karhunen-Loeve 或 K-L 方法）搜索 k 个最能代表数据的 n 维正交向量，其中 $k \leqslant n$。这样，原数据投影到一个小得多的空间上，导致维归约。与属性子集选择（3.4.4 节）通过保留原属性集的一个子集来减少属性集的大小不同，PCA 通过创建一个替换的、较小的变量集"组合"属性的基本要素。原数据可以投影到该较小的

集合中。PCA 常常能够揭示先前未曾察觉的联系，并因此允许解释不寻常的结果。

基本过程如下：

(1) 对输入数据规范化，使得每个属性都落入相同的区间。此步有助于确保具有较大定义域的属性不会支配具有较小定义域的属性。

(2) PCA 计算 k 个标准正交向量，作为规范化输入数据的基。这些是单位向量，每一个都垂直于其他向量。这些向量称为*主成分*。输入数据是主成分的线性组合。

(3) 对主成分按"重要性"或强度降序排列。主成分本质上充当数据的新坐标系，提供关于方差的重要信息。也就是说，对坐标轴进行排序，使得第一个坐标轴显示数据的最大方差，第二个显示数据的次大方差，如此下去。例如，图 3.5 显示原来映射到轴 X_1 和 X_2 的给定数据集的前两个主成分 Y_1 和 Y_2。这一信息帮助识别数据中的组群或模式。

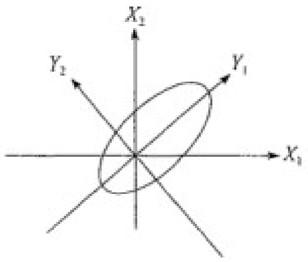

图 3.5　主成分分析。Y_1 和 Y_2 是给定数据的前两个主成分

(4) 既然主成分根据"重要性"降序排列，因此可以通过去掉较弱的成分（即方差较小的那些）来归约数据。使用最强的主成分，应当能够重构原数据的很好的近似。

PCA 可以用于有序和无序的属性，并且可以处理稀疏和倾斜数据。多于二维的多维数据可以通过将问题归约为二维问题来处理。主成分可以用做多元回归和聚类分析的输入。与小波变换相比，PCA 能够更好地处理稀疏数据，而小波变换更适合高维数据。

属性子集选择

用于分析的数据集可能包含数以百计的属性，其中大部分属性可能与挖掘任务不相关，或者是冗余的。例如，如果分析任务是按顾客听到广告后是否愿意在 AllElectronics 购买新的流行 CD 将顾客分类，与属性 *age*(年龄) 和 *music_taste*(音乐鉴赏力) 不同，诸如顾客的电话号码等属性多半是不相关的。尽管领域专家可以挑选出有用的属性，但这可能是一项困难而费时的任务，特别是当数据的行为不是十分清楚的时候更是如此（因此，需要分析）。遗漏相关属性或留下不相关属性都可能是有害的，会导致所用的挖掘算法无所适从。这可能导致发现质量很差的模式。此外，不相关或冗余的属性增加了数据量，可能会减慢挖掘进程。

属性子集选择[⊖]通过删除不相关或冗余的属性（或维）减少数据量。属性子集选择的目标是找出最小属性集，使得数据类的概率分布尽可能地接近使用所有属性得到的原分布。在缩小的属性集上挖掘还有其他的优点：它减少了出现在发现模式上的属性数目，使得模式更易于理解。

"*如何找出原属性的一个'好的'子集?*"对于 n 个属性，有 2^n 个可能的子集。穷举搜索找出属性的最佳子集可能是不现实的，特别是当 n 和数据类的数目增加时。因此，对于属性子集选择，通常使用压缩搜索空间的启发式算法。通常，这些方法是典型的**贪心**算法，在搜索属性空间时，总是做看上去是最佳的选择。它们的策略是做局部最优选择，期望由此导致全局最优解。在实践中，这种贪心方法是有效的，并可以逼近最优解。

"最好的"（和"最差的"）属性通常使用统计显著性检验来确定。这种检验假定属性是相互独立的。也可以使用一些其他属性评估度量，如建立分类决策树使用的*信息增益度量*[⊖]。

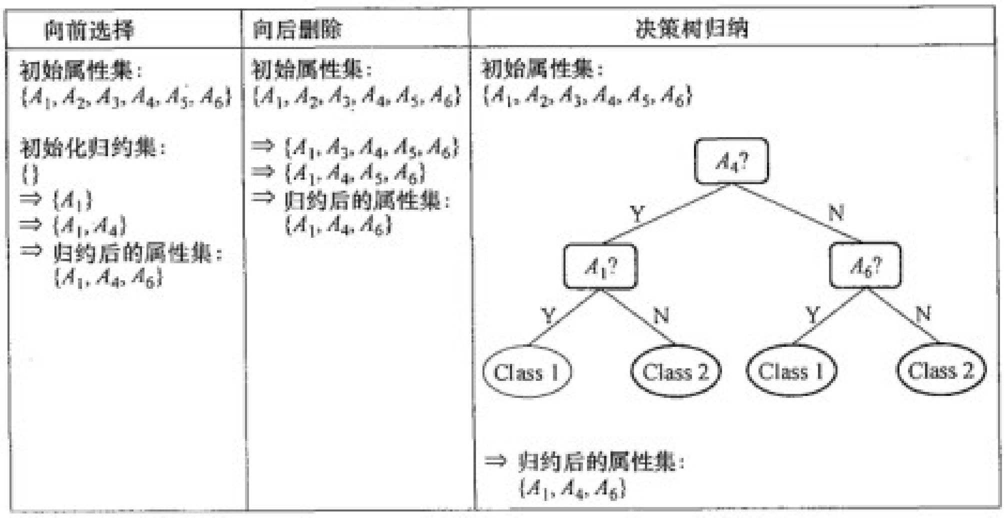

图 3.6　属性子集选择的贪心（启发式）方法

（1）**逐步向前选择**：该过程由空属性集作为归约集开始，确定原属性集中最好的属性，并将它添加到归约集中。在其后的每一次迭代，将剩下的原属性集中的最好的属性添加到该集合中。

（2）**逐步向后删除**：该过程由整个属性集开始。在每一步中，删除尚在属性集中最差的属性。

（3）**逐步向前选择和逐步向后删除的组合**：可以将逐步向前选择和逐步向后删除方法结合在一起，每一步选择一个最好的属性，并在剩余属性中删除一个最差的属性。

（4）**决策树归纳**：决策树算法（例如，ID3、C4.5 和 CART）最初是用于分类的。决策树归纳构造一个类似于流程图的结构，其中每个内部（非树叶）结点表示一个属性上的测试，每个分枝对应于测试的一个结果；每个外部（树叶）结点表示一个类预测。在每个结点上，算法选择“最好”的属性，将数据划分成类。

当决策树归纳用于属性子集选择时，由给定的数据构造决策树。不出现在树中的所有属性假定是不相关的。出现在树中的属性形成归约后的属性子集。

这些方法的结束条件可以不同。该过程可以使用一个度量阈值来决定何时停止属性选择过程。

在某些情况下，我们可能基于其他属性创建一些新属性。这种**属性构造**[⊖]可以帮助提高准确性和对高维数据结构的理解。例如，我们可能希望根据属性 *height*（高度）和 *width*（宽度）增加属性 *area*（面积）。通过组合属性，属性构造可以发现关于数据属性间联系的缺失信息，这对知识发现是有用的。

回归和对数线性模型：参数化数据归约

回归和对数线性模型可以用来近似给定的数据。在（简单）**线性回归**中，对数据建模，使之拟合到一条直线。例如，可以用以下公式，将随机变量 y（称做因变量）表示为另一随机变量 x（称为自变量）的线性函数，

$$y = wx + b \tag{3.7}$$

其中，假定 y 的方差是常量。在数据挖掘中，x 和 y 是数值数据库属性。系数 w 和 b（称做

回归系数）分别为直线的斜率和 y 轴截距。系数可以用最小二乘法求解，其最小化分离数据的实际直线与该直线的估计之间的误差。**多元回归**是（简单）线性回归的扩展，允许用两个或多个自变量的线性函数对因变量 y 建模。

对数线性模型（log-linear model）近似离散的多维概率分布。给定 n 维（例如，用 n 个属性描述）元组的集合，我们可以把每个元组看做 n 维空间的点。对于离散属性集，可以使用对数线性模型，基于维组合的一个较小子集，估计多维空间中每个点的概率。这使得高维数据空间可以由较低维空间构造。因此，对数线性模型也可以用于维归约（由于较低维空间的点通常比原来的数据点占据的空间要少）和数据光滑（因为与较高维空间的估计相比，较低维空间的聚集估计受抽样变化的影响较小）。

回归和对数线性模型都可以用于稀疏数据，尽管它们的应用可能是有限的。虽然两种方法都可以处理倾斜数据，但是回归可望更好。当用于高维数据时，回归可能是计算密集的，而对数线性模型表现出很好的可伸缩性，可以扩展到 10 维左右。

直方图

聚类

抽样

抽样可以作为一种数据归约技术使用，因为它允许用数据的小得多的随机样本（子集）表示大型数据集。假定大型数据集 D 包含 N 个元组。我们看看可以用于数据归约的、最常用的对 D 的抽样方法，如图 3.9 所示。

- s 个样本的**无放回简单随机抽样（SRSWOR）**：从 D 的 N 个元组中抽取 s 个样本（$s < N$），其中 D 中任意元组被抽取的概率均为 $1/N$，即所有元组的抽取是等可能的。
- s 个样本的**有放回简单随机抽样（SRSWR）**：该方法类似于 SRSWOR，不同之处在于当一个元组从 D 中抽取后，记录它，然后放回原处。也就是说，一个元组被抽取后，它又被放回 D，以便它可以被再次抽取。
- **簇抽样**：如果 D 中的元组被分组，放入 M 个互不相交的“簇”，则可以得到 s 个簇的简单随机抽样（SRS），其中 $s < M$。例如，数据库中元组通常一次取一页，这样每页就可以视为一个簇。例如，可以将 SRSWOR 用于页，得到元组的簇样本，由此得到数据的归约表示。也可以利用其他携带更丰富语义信息的聚类标准。例如，在空间数据库中，我们可以基于不同区域位置上的邻近程度定义簇。
- **分层抽样**：如果 D 被划分成互不相交的部分，称做“层”，则通过对每一层的 SRS 就可以得到 D 的分层抽样。特别是当数据倾斜时，这可以帮助确保样本的代表性。例如，可以得到关于顾客数据的一个分层抽样，其中分层对顾客的每个年龄组创建。这样，具有的顾客人数最少的年龄组肯定能够被代表。

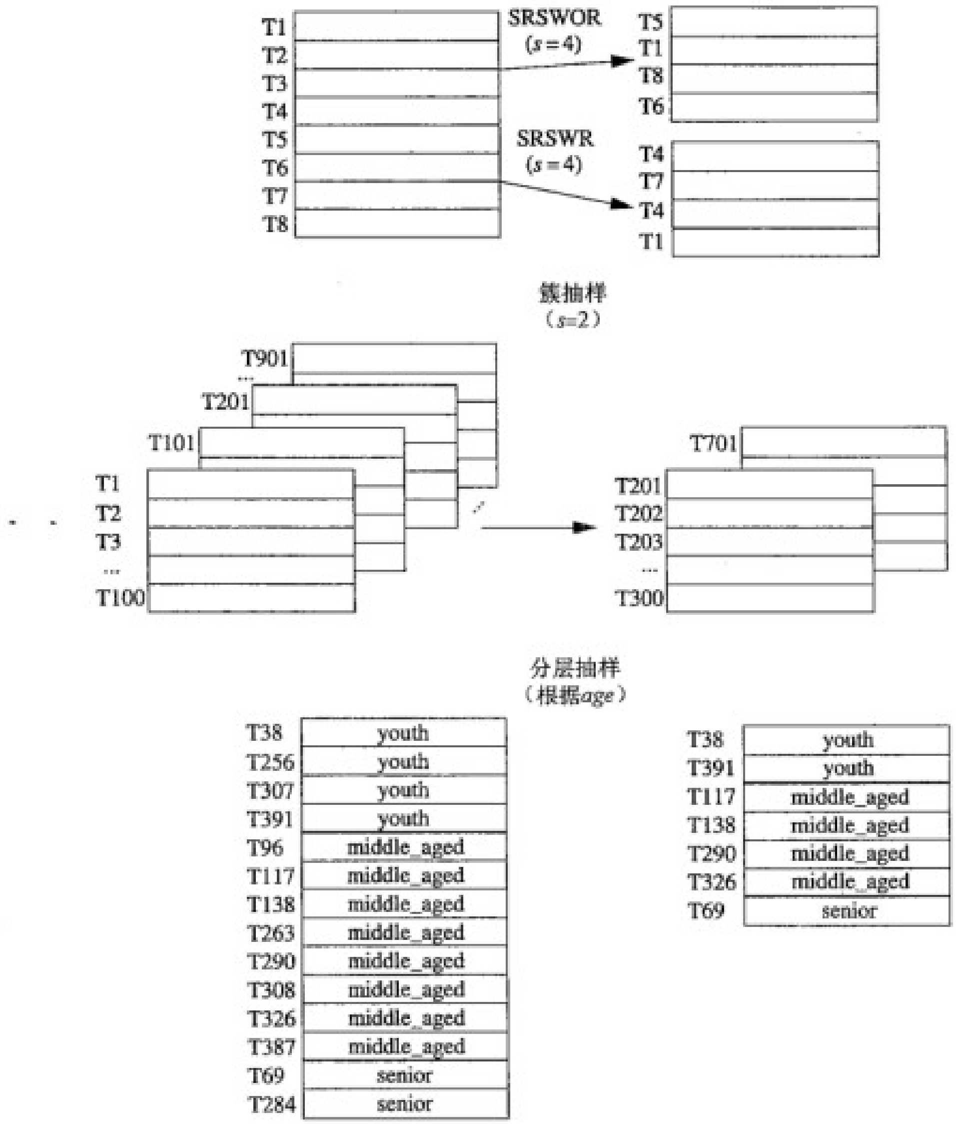

图 3.9 抽样可以用于数据归约

采用抽样进行数据归约的优点是，得到样本的花费正比例于样本集的大小 s，而不是数据集的大小 N。因此，抽样的复杂度可能亚线性（sublinear）于数据的大小。其他数据归约技术至少需要完全扫描 D。对于固定的样本大小，抽样的复杂度仅随数据的维数 n 线性地增加；而其他技术，如使用直方图，复杂度随 n 呈指数增长。

用于数据归约时，抽样最常用来估计聚集查询的回答。在指定的误差范围内，可以确定（使用中心极限定理）估计一个给定的函数所需的样本大小。样本的大小 s 相对于 N 可能非常小。对于归约数据的逐步求精，抽样是一种自然选择。通过简单地增加样本大小，这样的集合可以进一步求精。

数据立方体聚集

3.5数据变换与数据离散化（未仔细看）

数据变换策略概述

（1）**光滑**（smoothing）：去掉数据中的噪声。这类技术包括分箱、回归和聚类。

（2）**属性构造**（或特征构造）：可以由给定的属性构造新的属性并添加到属性集中，以帮助挖掘过程。

（3）**聚集**：对数据进行汇总或聚集。例如，可以聚集日销售数据，计算月和年销售量。通常，这一步用来为多个抽象层的数据分析构造数据立方体。

（4）**规范化**：把属性数据按比例缩放，使之落入一个特定的小区间，如 -1.0～1.0 或 0.0～1.0。

（5）**离散化**：数值属性（例如，年龄）的原始值用区间标签（例如，0～10，11～20 等）或概念标签（例如，*youth*、*adult*、*senior*）替换。这些标签可以递归地组织成更高层概念，导致数值属性的概念分层。图 3.12 显示了属性 *price* 的一个概念分层。对于同一个属性可以定义多个概念分层，以适合不同用户的需要。

（6）**由标称数据产生概念分层**：属性，如 *street*，可以泛化到较高的概念层，如 *city* 或 *country*。许多标称属性的概念分层都蕴含在数据库的模式中，可以在模式定义级自动定义。

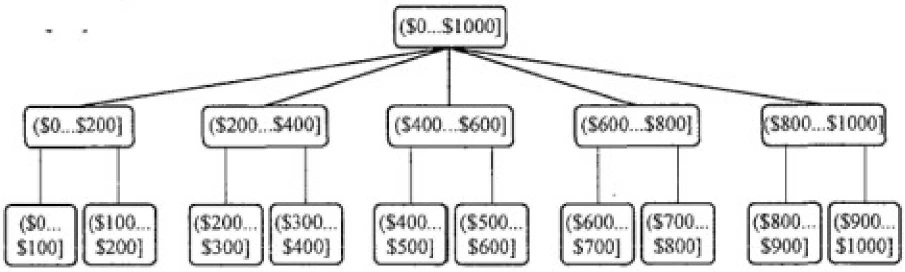

图 3.12　属性 *price* 的一个概念分层，其中区间（$X… $Y] 表示从 $X（不包括）到 $Y（包括）的区间

通过规范化变换数据

通过分箱离散化

通过直方图分析离散化

通过聚类、决策树和相关分析离散化

标称数据的概念分层产生