电子科技大学
University of Electronic Science and Technology of China

# Lecture 4 数据预处理技术

- 数据预处理
- 数据规约

# 教学目标

- 认识数据挖掘前数据进行预处理的必要性
- 掌握常用数据预处理的方法。

# 为什么要预处理数据？

- 现实世界的数据是“脏的 (dirty data)”

| ID | 学号 | 姓名 | 年龄 | 地址 | 籍贯 |
|---|---|---|---|---|---|
| 1 | 201422030123 | 李勇 | 21 | 四川省成都市成华区建设北路二段四号 | 四川省 |
| 2 | 201422020101 | 张小东 | | 四川省成都市高新区（西区）西源大道2006号 | |
| 3 | 201422030203 | 陈海 | 22 | 四川省成都市成华区建设北路二段四号 | 海南省 |
| 4 | 201422030203 | 陈海 | 22 | 四川省成都市成华区建设北路二段四号 | 海南省 |
| 5 | 201422030112 | 赵凯 | 24 | 四川省成都市成华区建设北路二段四号 | 江西省 |
| 6 | 201422030112 | 赵凯 | 20 | 成都市高新区（西区）西源大道2006号 | 江西省 |
| 7 | 201422010111 | 罗毅 | 101 | 四川省成都市成华区建设北路二段四号 | 四川省 |
| 8 | 201421030132 | 姜波 | 22 | 四川省成都市成华区建设北路二段四号 | 北京市 |
| 9 | 201421030132 | 姜波 | 22 | 四川省成都市成华区建设北路二段四号 | 背景市 |
| 10 | 201422030111 | 王雯 | 22 | 成都高新西源大道2006号 | 云南省 |

# 为什么要预处理数据？(续)

- 数据缺失：记录为空&属性为空 (e.g, ID=2)
- 数据重复：完全重复&不完全重复(e.g., ID=3&4, ID=5&6)
- 数据错误：异常值&不一致(e.g., ID=7, ID=8&9)
- 数据不可用：数据正确，但不可用(e.g., ID=10)

| ID | 学号 | 姓名 | 年龄 | 地址 | 籍贯 |
|---|---|---|---|---|---|
| 1 | 201422030123 | 李勇 | 21 | 四川省成都市成华区建设北路二段四号 | 四川省 |
| 2 | 201422020101 | 张小东 | | 四川省成都市高新区（西区）西源大道2006号 | |
| 3 | 201422030203 | 陈海 | 22 | 四川省成都市成华区建设北路二段四号 | 海南省 |
| 4 | 201422030203 | 陈海 | 22 | 四川省成都市成华区建设北路二段四号 | 海南省 |
| 5 | 201422030112 | 赵凯 | 24 | 四川省成都市成华区建设北路二段四号 | 江西省 |
| 6 | 201422030112 | 赵凯 | 20 | 成都市高新区（西区）西源大道2006号 | 江西省 |
| 7 | 201422010111 | 罗毅 | 101 | 四川省成都市成华区建设北路二段四号 | 四川省 |
| 8 | 201421030132 | 姜波 | 22 | 四川省成都市成华区建设北路二段四号 | 北京市 |
| 9 | 201421030132 | 姜波 | 22 | 四川省成都市成华区建设北路二段四号 | 背景市 |
| 10 | 201422030111 | 王雯 | 22 | 成都高新西源大道2006号 | 云南省 |

# 如何预防脏数据出现

- 制定数据标准
  - 统一多数据源的属性值编码
  - 尽可能赋予属性名和属性值明确的含义
- 优化系统设计
  - 关键属性尽可能采用选项方式，而不是手动填写
  - 重要属性出现在醒目的位置，采用必填选项
  - 异常值要给出修改提示

***墨菲定律：凡事只要有可能出错，那就一定会出错***

# 处理数据缺失

- 引起缺失值的原因
  - 设备异常
  - 在输入时，有些数据得不到重视而没有被输入
- 缺失值要经过推断而补上
  - 忽略该记录
  - 使用默认值
  - 使用属性平均值
  - 使用同类样本平均值
  - 预测最可能的值

# 处理数据重复

- 引起重复值的原因
  - 整合多个数据源的数据
  - 在输入时，有些数据重复输入

- 重复值经过推断进行合并
  - 删除完全重复的记录
  - 合并不同的表时，增加部分冗余属性（例如时间）

# 处理数据错误：不一致

- 引起不一致的原因
  - 数据录入者习惯不好
  - 数据没有统一的标准

- 数据不一致通过匹配进行修改
  - 制定清洗规则表，进行匹配
  - 通过统计描述，找到异常值

# 处理数据错误：数据噪声

- 引起数据噪音的原因
  - 数据记录的过程中存在偏差
  - 设备测量数据的过程中存在偏差

- 数据噪音可以通过
  - 分箱算法
  - 聚类算法
  - 回归算法

# 噪声数据的处理——分箱

- 分箱：把待处理的数据按照一定的规则放进一些箱子中，考察每一个箱子中的数据，采用某种方法分别对各个箱子中的数据进行处理。
- 箱子：按照属性值划分的子区间，如果一个属性值处于某个子区间范围内，就称把该属性值放进这个子区间代表的“箱子”里。
- 分箱技术需要确定的主要问题：
  - 分箱方法，即如何分箱
  - 数据平滑方法，即如何对每个箱子中的数据进行平滑处理

# 噪声数据的处理——分箱（续1）

- 分箱的方法：分箱前对记录集按目标属性值的大小进行排序。
  - 等深分箱法
  - 等宽分箱法
  - 用户自定义区间

- 例：学生奖学金排序后的值（人民币元）：

  800 1000 1200 1500 1500 1800 2000 2300 2500 2800 3000 3500 4000 4500 4800 5000

# 噪声数据的处理——分箱（续2）

- 等深分箱法（统一权重）：按记录行数分箱，每箱具有**相同的记录数**，每箱记录数称为箱的权重，也称箱子的深度。
- 设定权重（箱子深度）为4，上述例子分箱后的结果如下。

  箱1：800 1000 1200 1500

  箱2：1500 1800 2000 2300

  箱3：2500 2800 3000 3500

  箱4：4000 4500 4800 5000

# 噪声数据的处理——分箱（续3）

- 等宽分箱法（统一区间）：在整个属性值的区间上平均分布，即**每个箱的区间范围**是一个常量，称为箱子宽度。

- 设定区间范围（箱子宽度）为1000元人民币，分箱后

  箱1：800 1000 1200 1500 1500 1800

  箱2：2000 2300 2500 2800 3000

  箱3：3500 4000 4500

  箱4：4800 5000

# 噪声数据的处理——分箱（续4）

- 用户自定义区间：用户根据需要自定义区间。

- 用户自定义：如将学生奖学金划分为1000元以下、1000~2000、2000~3000、3000~4000和4000元以上几组，分箱后：

箱1：800
箱2：1000 1200 1500 1500 1800 2000
箱3：2300 2500 2800 3000
箱4：3500 4000
箱5：4500 4800 5000

# 噪声数据的处理——平滑处理

- 分箱后对数据进行平滑处理：

①按平均值平滑

对同一箱值中的数据求平均值，用平均值替代该箱子中的所有数据。

②按边界值平滑

用距离较小的边界值替代箱中每一数据。

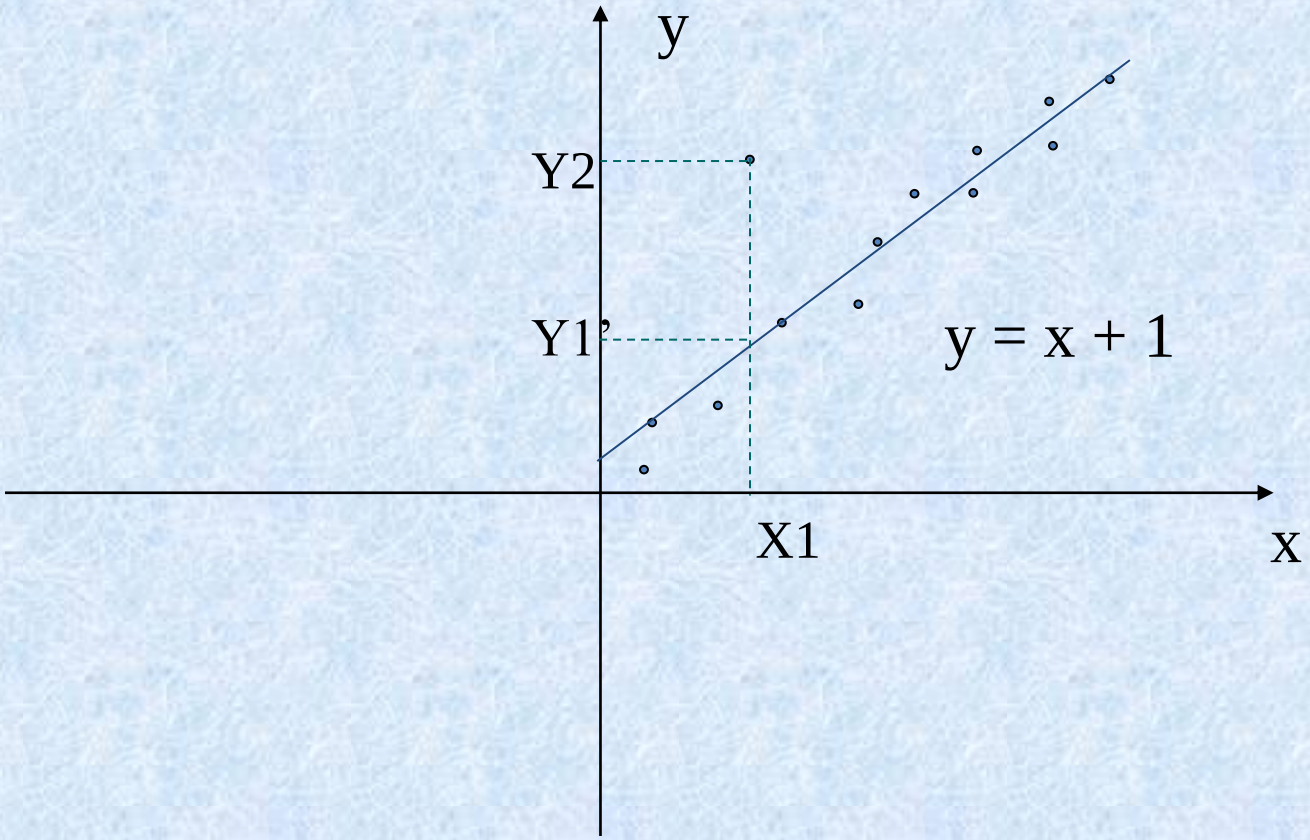
③按中值平滑

取箱子的中值，用来替代箱子中的所有数据。

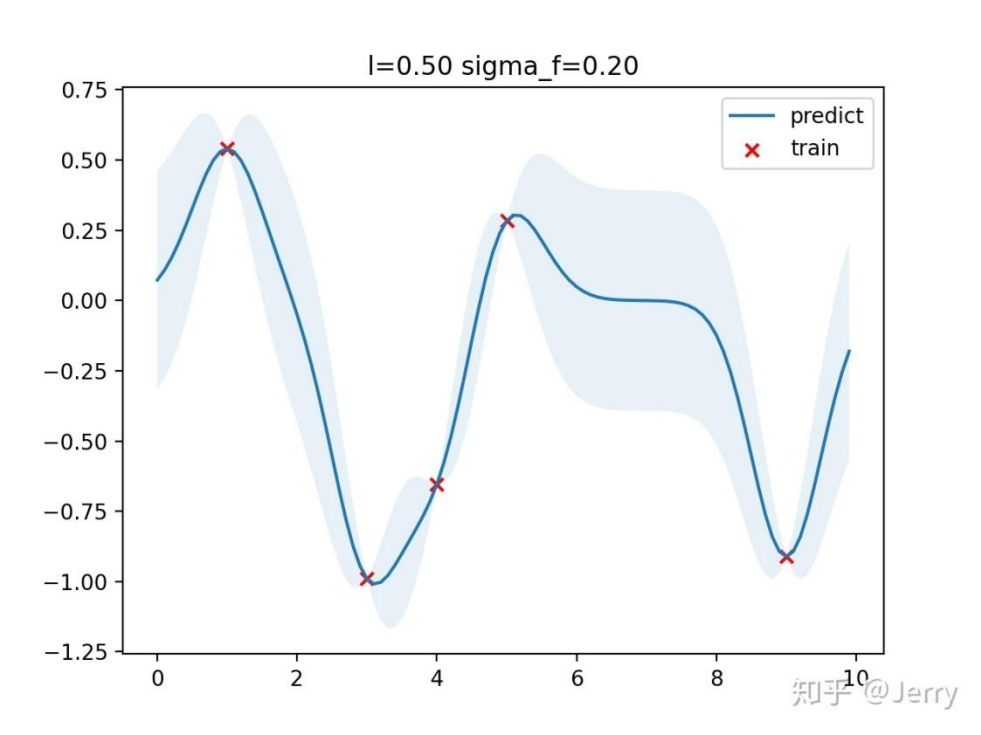
# 噪声数据的处理——回归

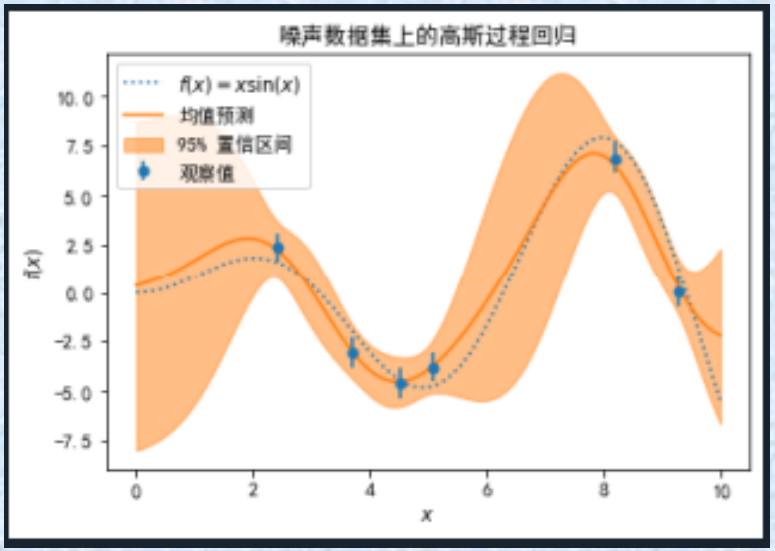
- 回归：发现两个相关的变量之间的变化模式，通过使数据适合一个函数来平滑数据，即利用**拟合函数**对数据进行平滑。
- 方法：
  - 线性回归（简单回归）：利用直线建模，将一个变量看作另一个变量的线性函数。
    如：Y=aX+b，其中a、b称为回归系数，可用最小二乘法求得a、b系数。
  - 非线性回归

# 噪声数据的处理——回归



# 噪声数据的处理——高斯回归

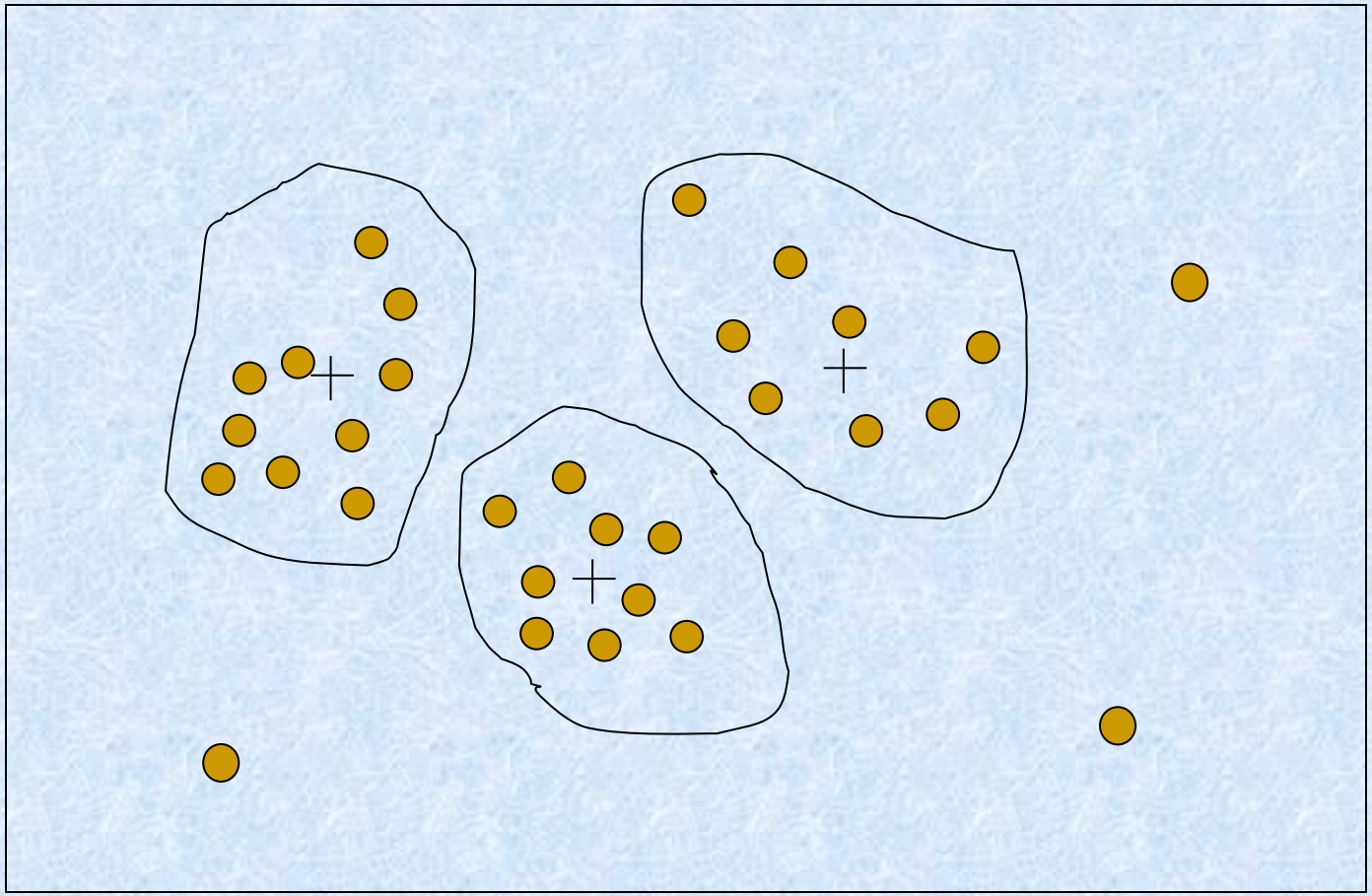


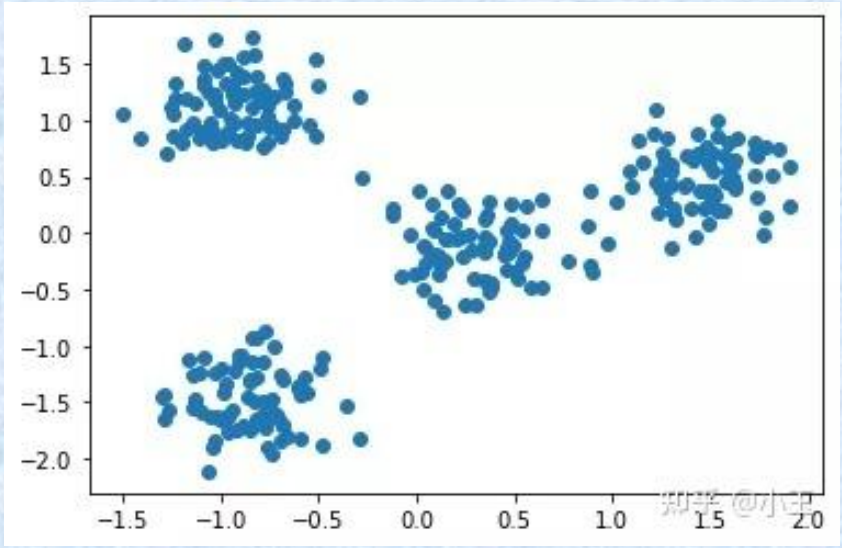
# 噪声数据的处理——聚类

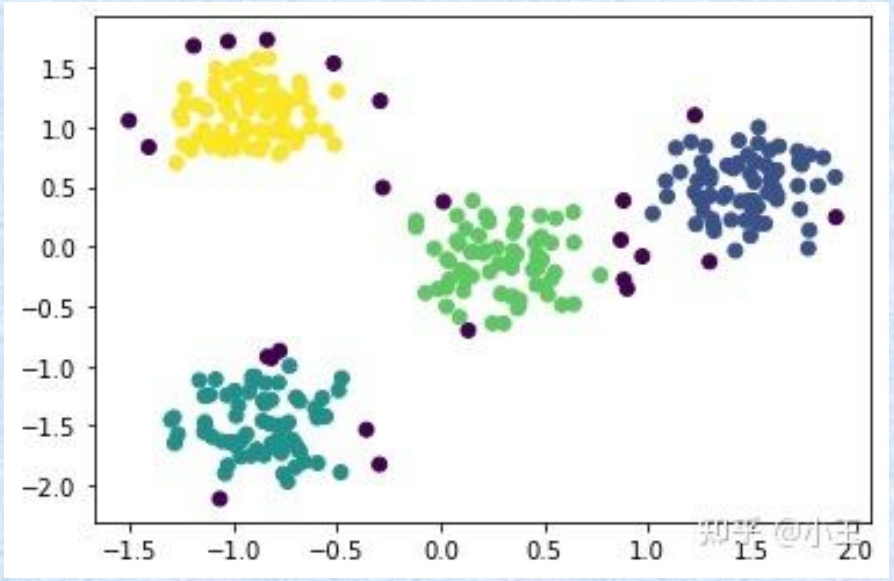
- 簇：一组数据对象集合。同一簇内的所有对象具有**相似性**，不同簇间对象具有较大差异性。
- 聚类：将物理的或抽象对象的集合分组为由不同簇，找出并清除那些落在簇之外的值（孤立点），这些**孤立点被视为噪声**。
- 通过聚类分析发现异常数据：相似或相邻近的数据聚合在一起形成了各个聚类集合，而那些位于这些聚类集合之外的数据对象，自然而然就被认为是异常数据。
- 特点：直接形成簇并对簇进行描述，不需要任何先验知识。

# 噪声数据的处理——聚类



# 基于密度的聚类方法——DBSCAN算法

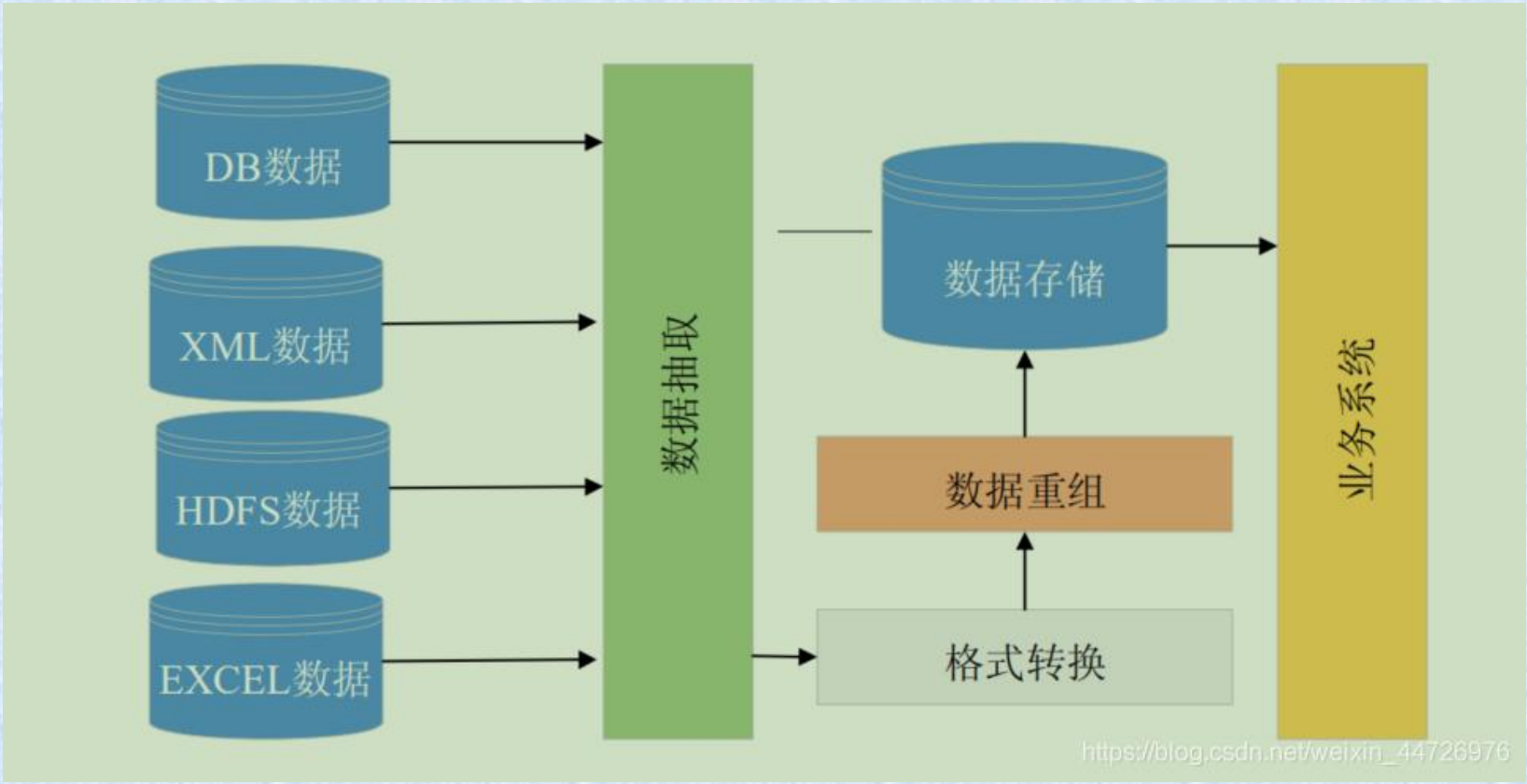


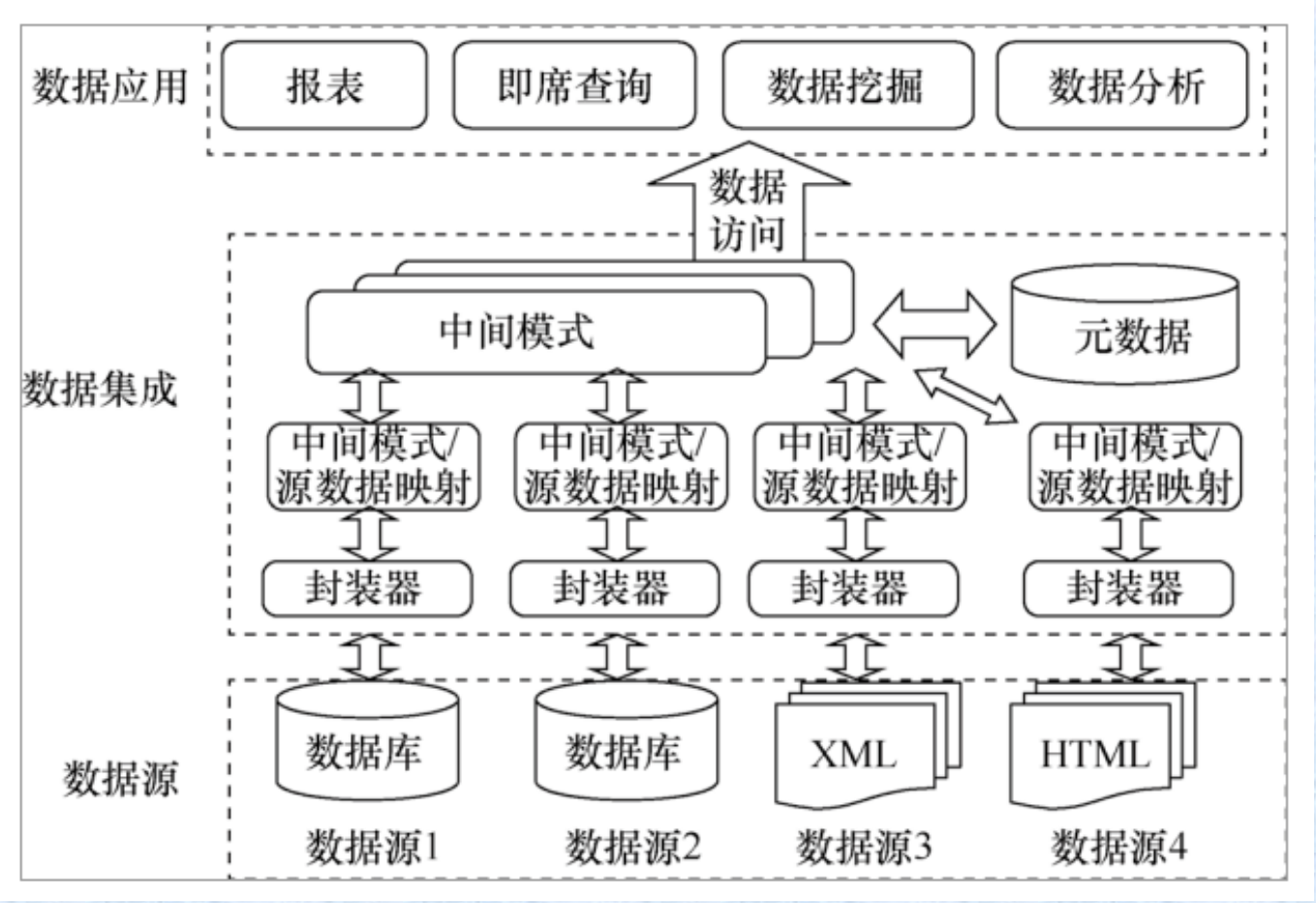
# 数据集成

- 数据集成：将多个数据源中的数据整合到一个一致的存储中
  - 模式匹配
  - 数据冗余
  - 数据值冲突

# 数据集成工作模式



# 基于中间模式的数据集成系统架构



# 数据集成——模式匹配

模式匹配主要用于发现并映射两个或多个异构数据源之间的属性对应关系。基于模式匹配实体对齐的目标是根据匹配属性的记录特征，将数据源中指代同一实体的记录连接起来。

- 整合不同数据源中的元数据。
- 实体识别问题：匹配来自不同数据源的现实世界的实体，比如：

  A.cust-id = B.customer_no 。

# 数据集成——数据冗余

- 同一属性在不同的数据库中会有不同的字段名。
- 一个属性可以由另外一个表导出。如：一个顾客数据表中的平均月收入属性，它可以根据月收入属性计算出来。
- 有些冗余可以被相关分析检测到

# 数据集成——数据值冲突

- 对于一个现实世界实体，其来自不同数据源的属性值或许不同。
- 产生的原因：表示的差异、比例尺度不同、或编码的差异等。例如：重量属性在一个系统中采用公制，而在另一个系统中却采用英制。同样价格属性不同地点采用不同货币单位。

# 数据变换——平滑（smoothing）

- 去除噪声，将连续的数据离散化，增加粒度
  - 分箱
  - 聚类
  - 回归

# 数据变换——聚集（clustering）

- 对数据进行汇总
  - avg(), count(), sum(), min(), max()…
  - 例如：每天销售额（数据）可以进行合计操作以获得每月或每年的总额。
  - 可以用来构造数据立方体

# 数据变换——数据概化（generalization）

- 用更抽象（更高层次）的概念来取代低层次或数据层的数据对象

- 例如：街道属性，就可以泛化到更高层次的概念，诸如：城市、国家。同样对于数值型的属性，如年龄属性，就可以映射到更高层次概念，如：年轻、中年和老年。

# 数据变换——规范化（normalization）

- 将数据按比例进行缩放，使之落入一个特定的区域，以消除数值型属性因大小不一而造成挖掘结果的偏差。如将工资收入属性值映射到[-1.0,1.0]范围内。
- 方法
  （1）最小-最大规范化
  （2）零-均值规范化（z-score规范化）
  （3）小数定标规范化

# 最小-最大规范化

- 已知属性的取值范围，将原取值区间[old_min,old_max]映射到new_min,new_max]

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

- 保留了原来数据中存在的关系。但若将来遇到超过目前属性[old_min,old_max]取值范围的数值，将会引起系统出错

# 最小-最大规范化

**示例 2.1**：假设属性 *income* 的最大最小值分别是 12,000 元和 98,000 元，若要利用最大最小规格化方法将属性 *income* 的值映射到 0 至 1 的范围内，那么对属性 *income* 的 73,600 元将被转化为 $\frac{73,600-12,000}{98,000-12,000}(1.0-0.0)=0.716$。 ■

# 零-均值规范化（z-score规范化）

- 根据属性A的均值和偏差来对A进行规格化，常用于属性最大值与最小值未知；或使用最大最小规格化方法时会出现异常数据的情况。

$$v' = \frac{v - \overline{A}}{\sigma_A}$$

其中的$\overline{A}$和$\sigma_A$分别为属性$A$的均值和方差

# 零-均值规范化（z-score规范化）

**示例 2.2**：假设属性 *income* 的均值与方差分别为 54,000 元和 16,000 元，使用零均值规格化方法将 73,600 元的属性 *income* 值映射为

$$\frac{73,600-54,000}{16,000}=1.225。$$

■

# 数据变换——属性构造

- 利用已有属性集构造出新的属性，并加入到现有属性集合中以帮助挖掘更深层次的模式知识，提高挖掘结果准确性。

- 例如：根据宽、高属性，可以构造一个新属性：面积。

# 数据归约（数据消减）

- 数据规约方法类似数据集压缩，它通过维度的减少或者数据量的减少，来达到降低数据规模的目的。
- 对大规模数据库内容进行复杂的数据分析通常需要耗费大量的时间。
- 数据归约（消减）技术用于帮助从原有庞大数据集中获得一个精简的数据集合，并使这一精简数据集保持原有数据集的完整性，这样在精简数据集上进行数据挖掘显然效率更高，并且挖掘出来的结果与使用原有数据集所获得结果基本相同。

# 数据归约标准

- 用于数据归约的时间不应当超过或“抵消”在归约后的数据上挖掘节省的时间
- 归约得到的数据比原数据小得多，但可以产生相同或几乎相同的分析结果

# 数据归约的方法

数据立方体聚集：

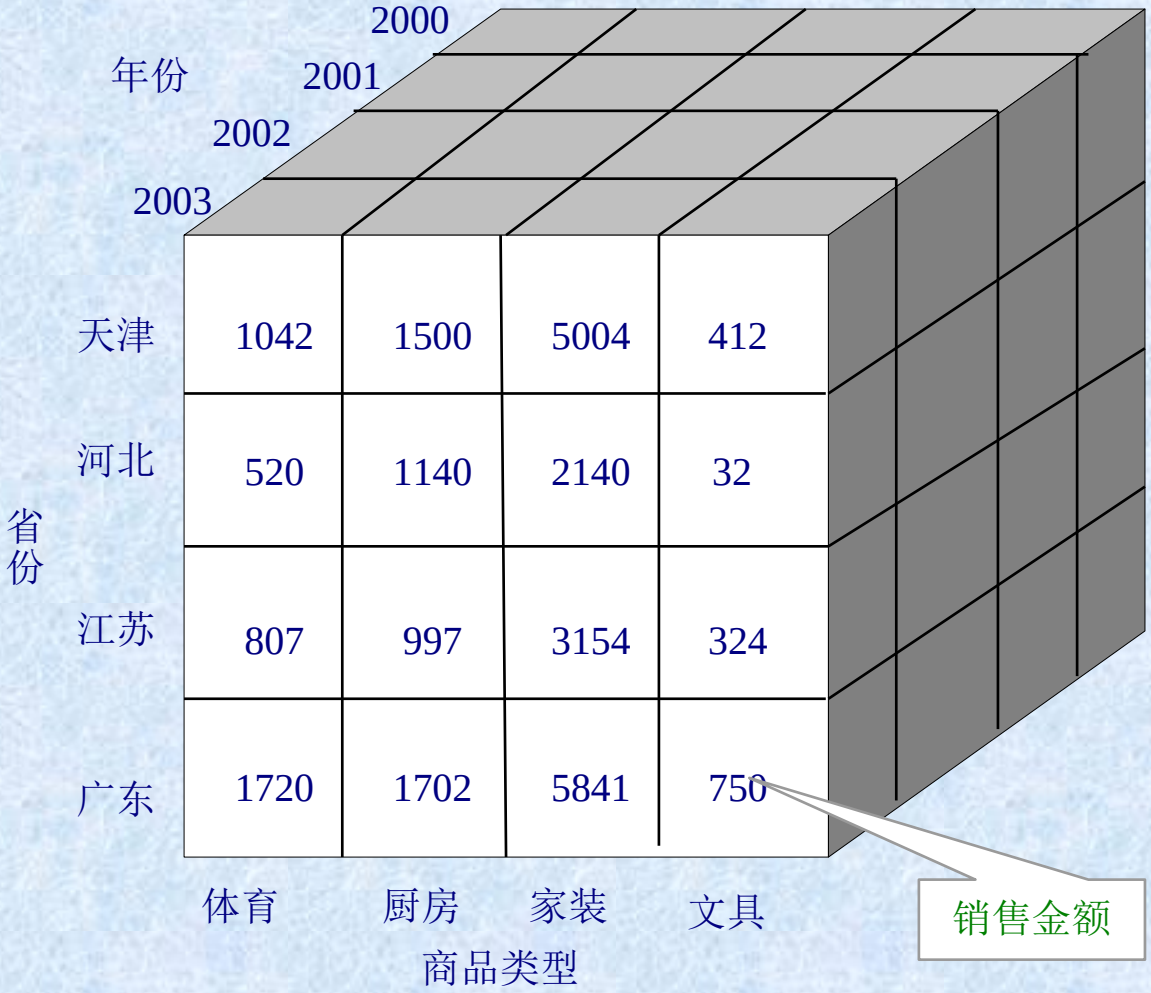
- 维归约
- 数据压缩
- 数值归约
- 离散化和概念分层生成

# 数据归约——数据立方体聚集

- 数据立方体基本概念：
  - 数据立方体是数据的多维建模和表示，由维和事实组成。
  - 维，即属性
  - 事实，即数据

- 数据立方体聚集定义——将n维数据立方体聚集为n-1维的数据立方体。

# 数据归约——数据立方体聚集



# 数据归约——数据立方体聚集

| 年份 | 体育 | 厨房 | 家装 | 文具 |
|---|---|---|---|---|
| 2004 | 3600 | 3514 | 6520 | 1546 |
| 2005 | 3124 | 4020 | 8160 | 1472 |
| 2006 | 3870 | 4966 | 11200 | 1460 |
| 2007 | 4089 | 5339 | 16139 | 1518 |

商品类型

销售金额

**聚集后的销售数据立方体**

# 数据归约——数据立方体聚集

- 下图数据是某商场2000～2002年每季度的销售数据，对这种数据进行聚集，使结果数据汇总每年的总销售额，而不是每季度的总销售额。

2002 年
2001 年
2000 年

| 季度 | 销售额 |
|---|---|
| 一季度 | 224 000 元 |
| 二季度 | 408 000 元 |
| 三季度 | 350 000 元 |
| 四季度 | 586 000 元 |

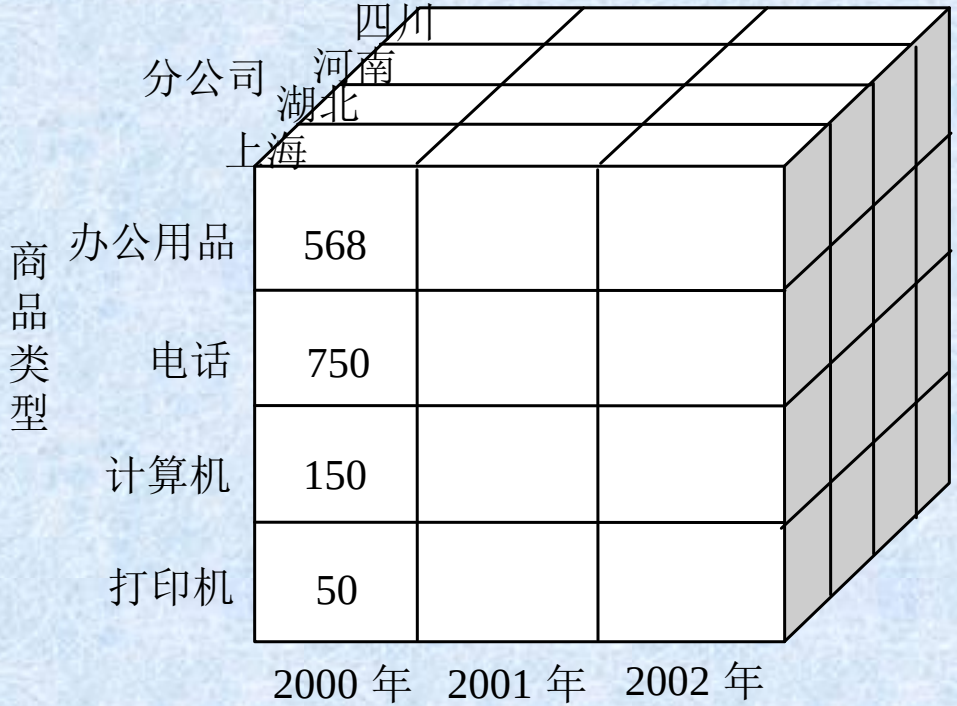
对年度内的各季度数据进行sum（求和）聚集

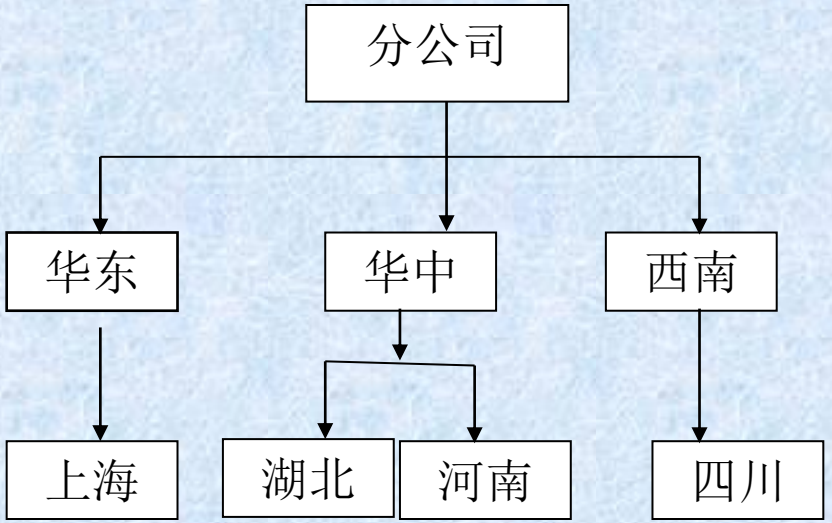
| 年 | 销售额 |
|---|---|
| 2000 | 1 568 000 元 |
| 2001 | 2 356 000 元 |
| 2002 | 3 594 000 元 |

**聚集后数据量明显减少，**
**但没有丢失分析任务所需的信息。**

# 数据归约——数据立方体聚集

- 下图所示数据立方体用于某销售企业每类商品在各分公司年销售多维数据分析。每个单元存放一个聚集值，对应于多维空间的一个数据点。每个属性可能存在概念分层，允许在多个抽象层进行数据分析。



# 数据归约——维归约

- 维归约——去掉无关的属性，减少数据挖掘处理的数据量。

- 例如：挖掘顾客是否会在商场购买 播放机的分类规则时，顾客的电话号码很可能与挖掘任务无关，应该可以去掉。

- 目标：寻找出最小的属性子集并确保新数据子集的概率分布尽可能接近原来数据集的概率分布。

# 维归约——选择相关属性子集

- 逐步向前选择
  - 从一个空属性集（作为属性子集初始值）开始，每次从原来属性集合中选择一个当前最优的属性添加到当前属性子集中。直到无法选择出最优属性或满足一定阈值约束为止。
- 逐步向后删除
  - 从一个全属性集（作为属性子集初始值）开始，每次从当前属性子集中选择一个当前最差的属性并将其从当前属性子集中消去。直到无法选择出最差属性为止或满足一定阈值约束为止。
- 向前选择和向后删除结合
- 判定树（决策树）归纳
  - 利用决策树的归纳方法对初始数据进行分类归纳学习，获得一个初始决策树，所有没有出现这个决策树上的属性均认为是无关属性，因此将这些属性从初始属性集合删除掉，就可以获得一个较优的属性子集。
- 基于统计分析的归约

# 数据归约——数据压缩

- 数据压缩——用数据编码或者变换，得到原始数据的压缩表示
- 在数据挖掘领域通常使用的两种数据压缩方法均是有损的：
  - 主成分分析法（PCA）
    假定待压缩的数据由$N$个取自$k$个维的元组或数据向量组成。主要成分分析并搜索得到$c$个最能代表数据的$k$维正交向量，这里 $c \leqslant k$。这样就可以把原数据投影到一个较小的空间，实现数据压缩
  - 小波转换

# 数据归约——数据压缩

- 压缩算法分类：
  - 无损(loseless)压缩：可以不丢失任何信息地还原压缩数据。
    - 例如：字符串压缩
    - 有广泛的理论基础和精妙的算法
  - 有损(lossy)压缩：只能重新构造原数据的近似表示。
    - 例如：音频/视频压缩
    - 有时可以在不解压整体数据的情况下，重构某个片断

# 数据归约——数值归约

- 数值归约——用较小的数据表示数据，或采用较短的数据单位，或者用数据模型代表数据，减少数据量。
- 常用的方法
  - 直方图
  - 用聚类数据表示实际数据
  - 抽样（采样）
  - 参数回归法

# 数值归约:直方图（频率一值对应关系）

- 利用分箱方法对数据分布情况进行近似

**示例 2.4**：以下是一个商场所销售商品的价格清单（按递增顺序排列，括号中的数表示前面数字出现次数）

1（2）、5（5）、8（2）、10（4）、12、14（3）、15（5）、18（8）、20（7）、21（4）、25（5）、28、30（3）

上述数据所形成属性值/频率对的直方图如图-2.6 所示。 ■

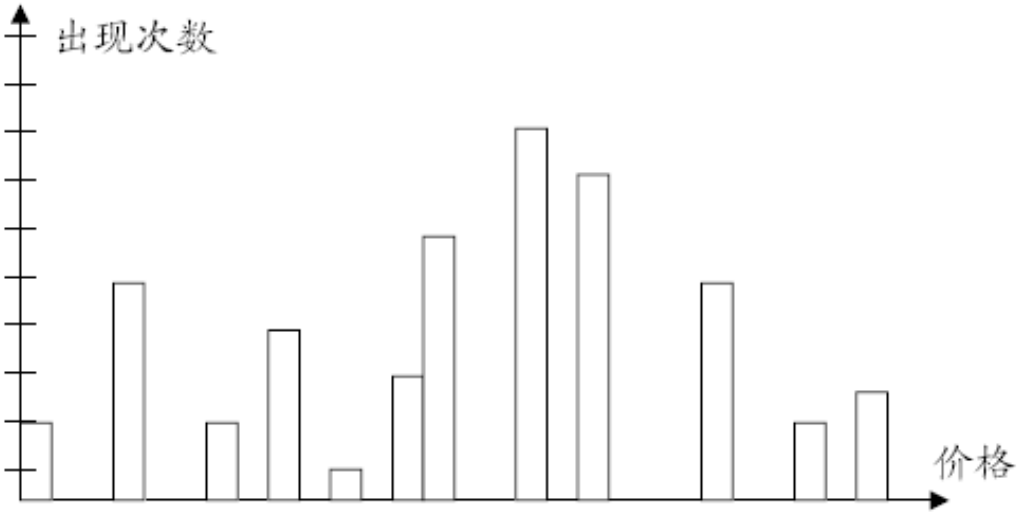
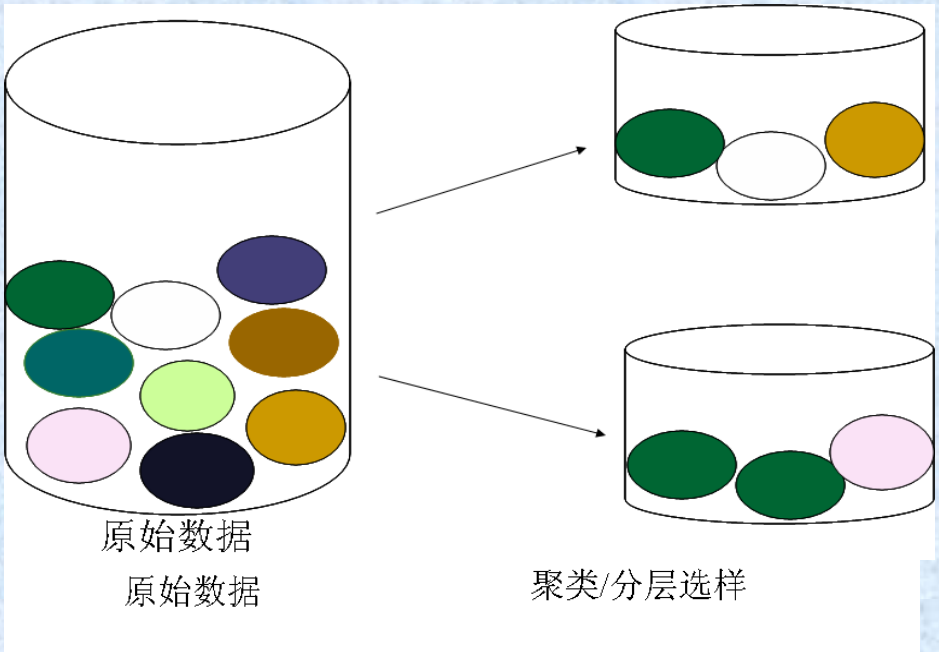
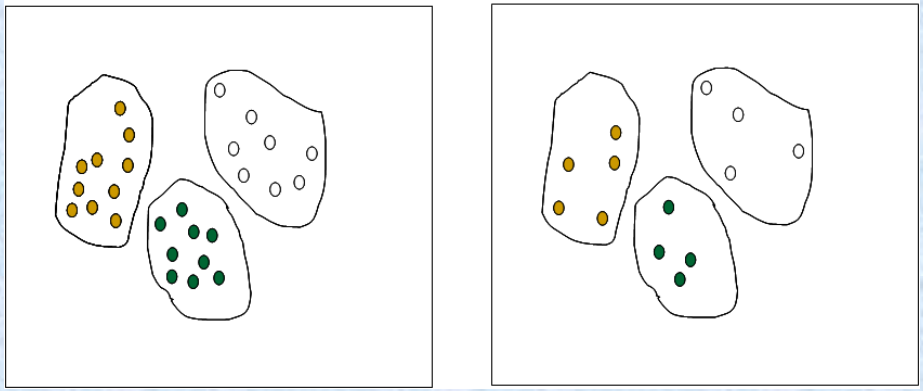



图-2.6 数据直方图描述示意（以 1 元为单位）

# 数值归约:用聚类数据表示实际数据



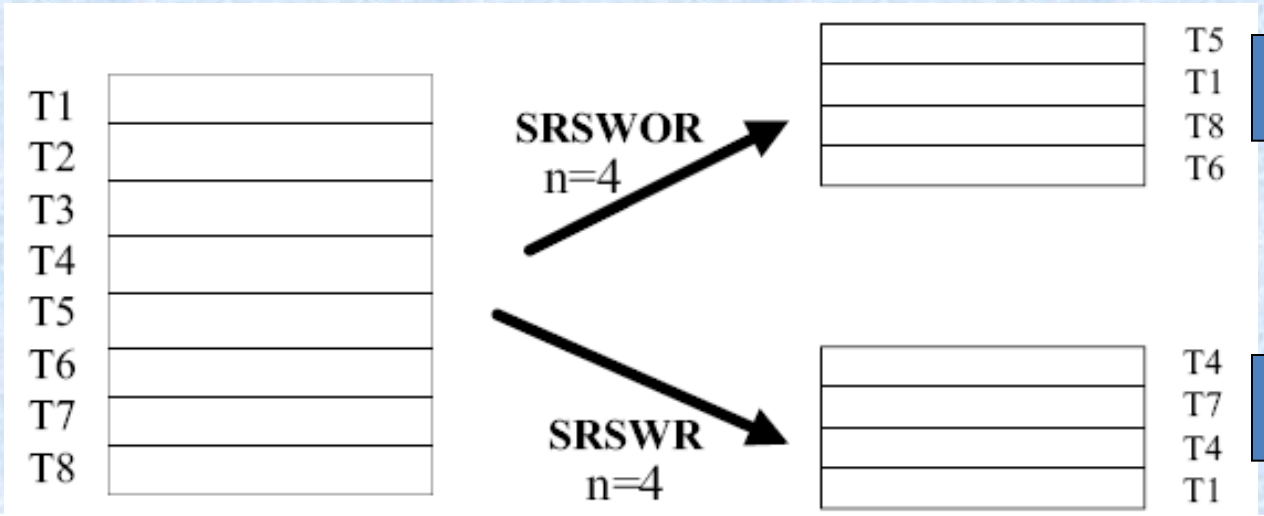



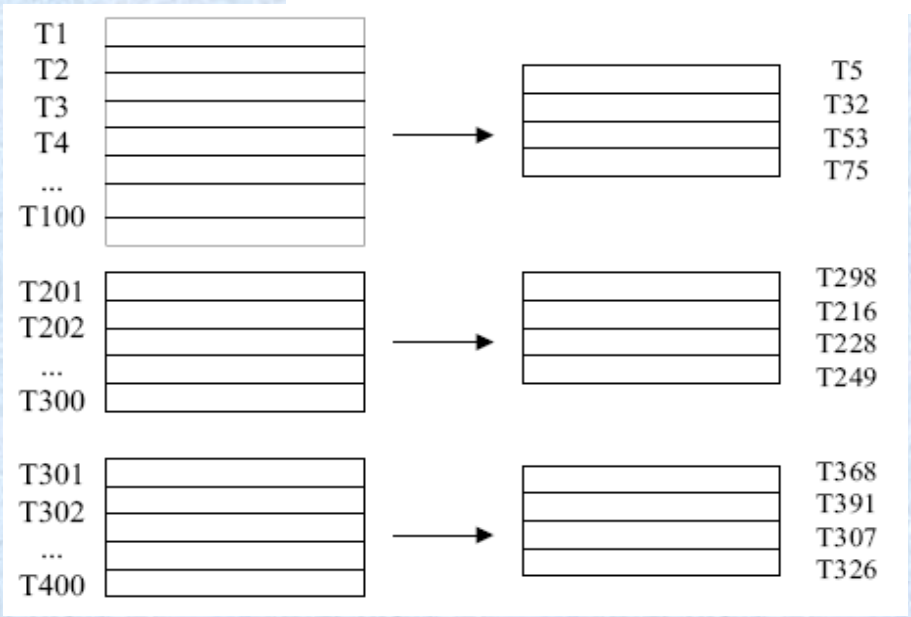
# 数值归约——抽样（采样）

- 优点：获取样本的时间仅与样本规模成正比
- 方法：
  - 不放回简单随机抽样
  - 放回简单随机抽样
  - 聚类抽样：先聚类，再抽样
  - 分层抽样：先分层，再抽样

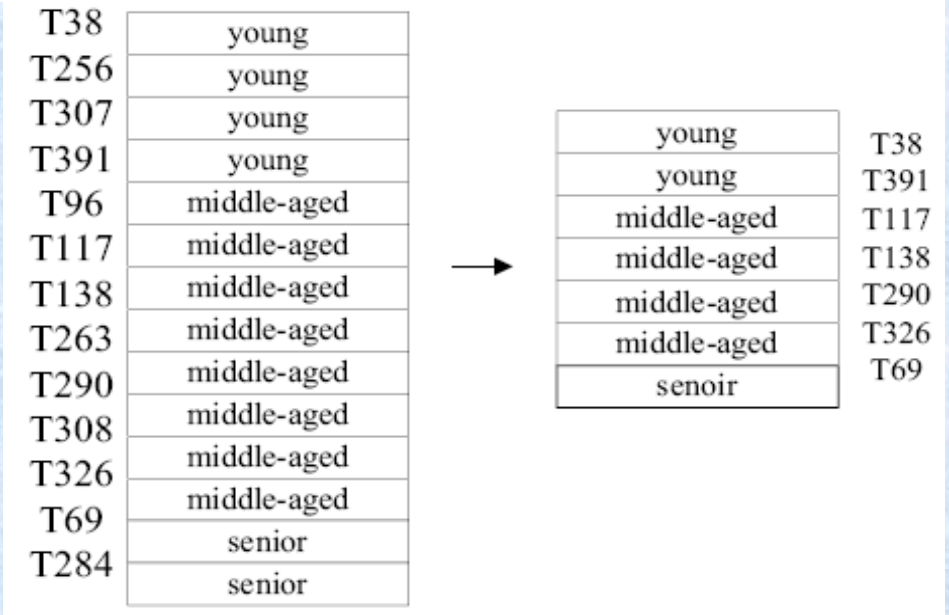
# 数值归约——参数回归法

- 通常采用一个模型来评估数据，该方法只需要存放参数，而不是实际数据。能大大简少数据量，但只对数值型数据有效。
- 方法：
  - 线性回归
  - 非线性回归


T1
T2
T3
T4
T5
T6
T7
T8
SRSWOR
n=4
T5
T1
T8
T6
不放回简单随机抽样
SRSWR
n=4
T4
T7
T4
T1
放回简单随机抽样


T1
T2
T3
T4
...
T100
T5
T32
T53
T75
T201
T202
...
T300
T298
T216
T228
T249
T301
T302
...
T400
T368
T391
T307
T326

聚类抽样

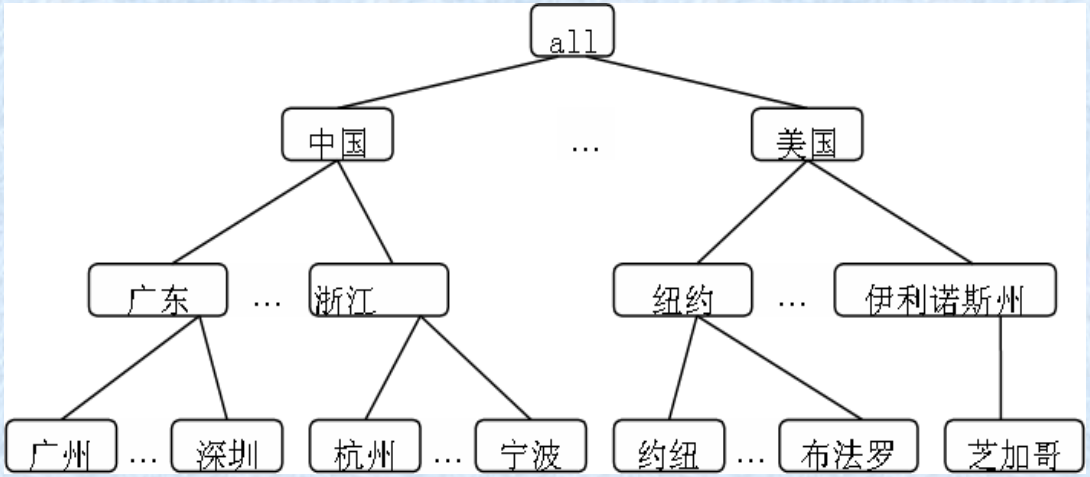

T38 young
T256 young
T307 young
T391 young
T96 middle-aged
T117 middle-aged
T138 middle-aged
T263 middle-aged
T290 middle-aged
T308 middle-aged
T326 middle-aged
T69 senior
T284 senior
young T38
young T391
middle-aged T117
middle-aged T138
middle-aged T290
middle-aged T326
senoir T69

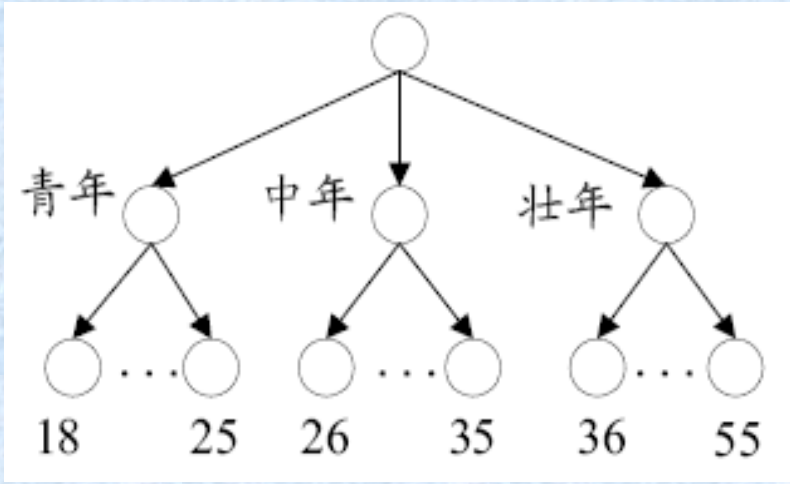
分层抽样

# 数据归约——离散化与概念分层生成

- 三种类型的属性值
  - 名称型——e.g. 无序集合中的值
  - 序数——e.g. 有序集合中的值
  - 连续值——e.g. 实数
- 离散化技术
  - 以通过将属性（连续取值）域值范围分为若干区间，来帮助消减一个连续（取值）属性的取值个数。
- 概念分层
  - 概念分层定义了一组由低层概念集到高层概念集的映射。它允许在各种抽象级别上处理数据，从而在多个抽象层上发现知识。用较高层次的概念替换低层次（如年龄的数值）的概念，以此来减少取值个数。虽然一些细节在数据泛化过程中消失了，但这样所获得的泛化数据或许会更易于理解、更有意义。在消减后的数据集上进行数据挖掘显然效率更高。
  - 概念分层结构可以用树来表示，树的每个节点代表一个概念。

# 数据归约——概念分层生成


all
中国
…
美国
广东
…
浙江
纽约
…
伊利诺斯州
广州
…
深圳
杭州
…
宁波
约纽
…
布法罗
芝加哥


青年
中年
壮年
18
25
26
35
36
55

# 数值数据的概念分层生成方法

- 分箱
  - 属性的值可以通过将其分配到各分箱中而将其离散化。利用每个分箱的均值和中数替换每个分箱中的值（利用均值或中数进行平滑）。循环应用这些操作处理每次操作结果，就可以获得一个概念层次树。
- 直方图
  - 循环应用直方图分析方法处理每次划分结果，从而最终自动获得多层次概念树，而当达到用户指定层次水平后划分结束。最小间隔大小也可以帮助控制循环过程，其中包括指定一个划分的最小宽度或每一个层次每一划分中数值个数等。
- 聚类
  - 聚类算法可以将数据集划分为若干类或组。每个类构成了概念层次树的一个节点；每个类还可以进一步分解为若干子类，从而构成更低水平的层次。当然类也可以合并起来构成更高层次的概念水平。
- 基于熵的离散化

# 数值数据的概念分层生成方法

- 自然划分分段
  - 将数值区域划分为相对一致的、易于阅读的、看上去更直观或自然的区间。
  - 聚类分析产生概念分层可能会将一个工资区间划分为：[51263.98, 60872.34]
  - 通常数据分析人员希望看到划分的形式为[50000，60000]
  - 划分方法：3-4-5规则
    - 如果一个区间最高有效位上包含3，6，7或9个不同的值，就将该区间划分为**3**个等宽子区间；(7→2,3,2)
    - 如果一个区间最高有效位上包含2，4，或8个不同的值，就将该区间划分为**4**个等宽子区间；
    - 如果一个区间最高有效位上包含1，5，或10个不同的值，就将该区间划分为**5**个等宽子区间；
    - 将该规则递归的应用于每个子区间，产生给定数值属性的概念分层；
    - 对于数据集中出现的最大值和最小值的极端分布，为了避免上述方法出现的结果扭曲，可以在顶层分段时，选用一个大部分的概率空间(如5%-95%)，越出顶层分段的特别高和特别低的采用类似的方法形成单独的区间。

# 数值数据的概念分层生成方法

**示例 2.5**：假设某个时期内一个商场不同分支的利润数从-351,976 元到4,700,896 元，要求利用 3-4-5 规则自动构造利润属性的一个概念层次树。

设在上述范围取值为 5%至 95%的区间为：-159,876 元至 1,838,761 元。而应用 3-4-5 规则具体步骤如下：

（1）属性的最小最大值分别为：MIN = -351,976 元、MAX = 4,700,896 元。而根据以上计算结果，取值 5%至 95%的区间范围（边界）应为：LOW = -159,876 元、HIGH = 1,838,761 元。

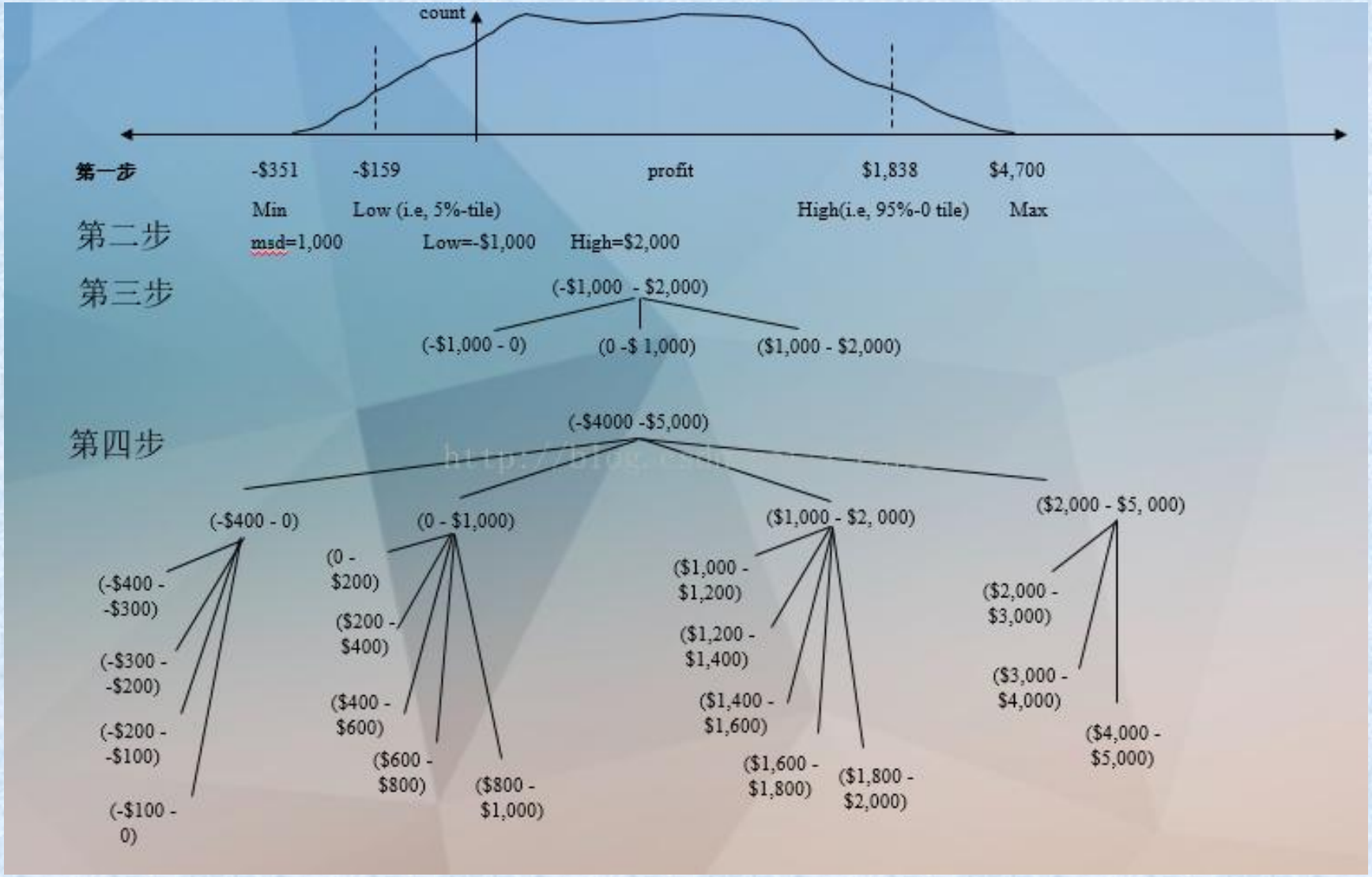
（2）依据 LOW 和 HIGH 及其取值范围，确定该取值范围应按 1,000,000 元单位进行区间分解，从而得到：LOW' = -1,000,000 元、HIGH'=2,000,000 元。

（3）由于 LOW'与 HIGH'之间有 3 个不同值,即（2,000,000 - （-1,000,000））/1,000,000 = 3。将 LOW'与 HIGH'之间区间分解为三个等宽小区间，它们分别是(–1,000,000 元 – 0 元]、(0 元 – 1,000,000 元]、(1,000,000 元 – 2,000,000 元]作为概念树的最高层组成。

（4）现在检查原来属性的 MIN 和 MAX 值与最高层区间的联系。MIN 值落入(–1,000,000 元 – 0 元]，因此调整左边界，对 MIN 取整后得-400,000 元，所以第一个区间（最左边区间）调整为(–400,000 – 0 元]。而由于 MAX 值不在最后一个区间(1,000,000 元 – 2,000,000 元]，因此需要新建一个区间（最右边区间），对 MAX 值取整后得 5,000,000 元，因此新区间就为(2,000,000 元 – 5,000,000 元]，这样概念树最高层就最终包含四个区间，它们分别是：(-400,000 元 – 0 元]、(0 元 – 1,000,000 元]、(1,000,000 元– 2,000,000 元]、(2,000,000 元– 5,000,000 元]。

（5）对上述分解所获得的区间继续应用 3–4–5 规则进行分解，以构成概念树的第二层区间组成内容。即：

- 第一个区间(-400,000 元 – 0 元]分解四个子区间，它们分别是(-400,000 元 – -300,000 元]、(-300,000 元 – -200,000 元]、(-200,000 元– -100,000 元]和(-100,000 元– 0 元]。
- 第二个区间(0 元 – 1,000,000 元]分解五个子区间，它们分别是(0 元 – 200,000 元]、(200,000 元 – 400,000 元]、(400,000 元– 600,000 元]、(600,000 元– 800,000 元]和(800,000 元– 1,000,000 元]。
- 第三个区间(1,000,000 元 – 2,000,000 元]分解五个子区间，它们分别是(1,000,000 元 – -1,200,000 元]、(1,200,000 元 – 1,400,000 元]、(1,400,000 元– 1,600,000 元]、(1,600,000 元– 1,800,000 元]和(1,800,000 元– 2,000,000 元]。
- 第四个区间(2,000,000 元 – 5,000,000 元]分解三个子区间，它们分别是(2,000,000 元 – -3,000,000 元]、(3,000,000 元 – 4,000,000 元]和(4,000,000 元– 5,000,000 元]。 ■


count
第一步
-$351
-$159
profit
$1,838
$4,700
Min
Low (i.e, 5%-tile)
High(i.e, 95%-0 tile)
Max
第二步
msd=1,000
Low=-$1,000
High=$2,000
第三步
(-$1,000 - $2,000)
(-$1,000 - 0)
(0 -$ 1,000)
($1,000 - $2,000)
第四步
(-$4000 -$5,000)
(-$400 - 0)
(0 - $1,000)
($1,000 - $2, 000)
($2,000 - $5, 000)
(-$400 - -$300)
(-$300 - -$200)
(-$200 - -$100)
(-$100 - 0)
(0 - $200)
($200 - $400)
($400 - $600)
($600 - $800)
($800 - $1,000)
($1,000 - $1,200)
($1,200 - $1,400)
($1,400 - $1,600)
($1,600 - $1,800)
($1,800 - $2,000)
($2,000 - $3,000)
($3,000 - $4,000)
($4,000 - $5,000)

# 分类(类别)数据的概念分层生成方法

- 类别属性可取有限个不同的值且这些值之间无大小和顺序。这样的属性有：国家、工作、商品类别等。
- 构造类别属性的概念层次树的主要方法：
  - 通过指定属性之间的包含关系产生分层
    - 例如：一个关系数据库中的地点属性将会涉及以下属性：街道、城市、省和国家。根据数据库模式定义时的描述，可以很容易地构造出（含有顺序语义）层次树，即：街道/城市/省/国家
  - 对数据进行分组（聚合）产生分层
    - 例如：在模式定义基础构造了省和国家的层次树，这时可以手工加入：安徽、江苏、山东⊂华东地区；广东、福建⊂华南地区等“地区”中间层次。

# 分类(类别)数据的概念分层生成方法

- 类别属性可取有限个不同的值且这些值之间无大小和顺序。这样的属性有：国家、工作、商品类别等。

- 构造类别属性的概念层次树的主要方法：
  - 通过指定属性之间的包含关系产生分层
  - 对数据进行分组（聚合）产生分层
  - 由属性值的个数产生分层
  - 根据数据语义产生分层