

# 作业反馈

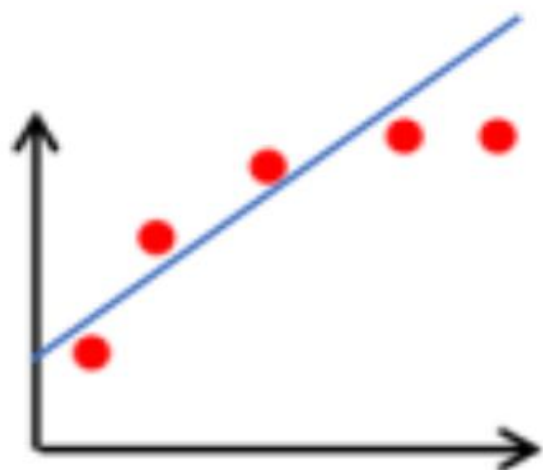


过拟合: overfitting, 在训练集上误差低, 测试集上误差高;

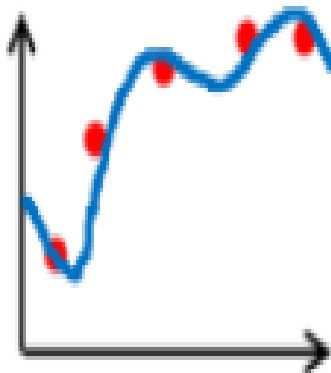
拟合: fitting

欠拟合: underfitting, 模型在训练集上误差很高;

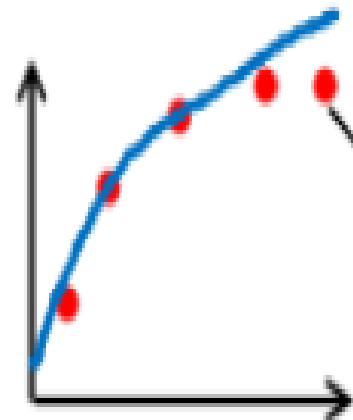
$$y = ax + b \quad y = ax^4 + bx^3 + cx^2 + dx + e \quad y = ax^2 + bx + c$$



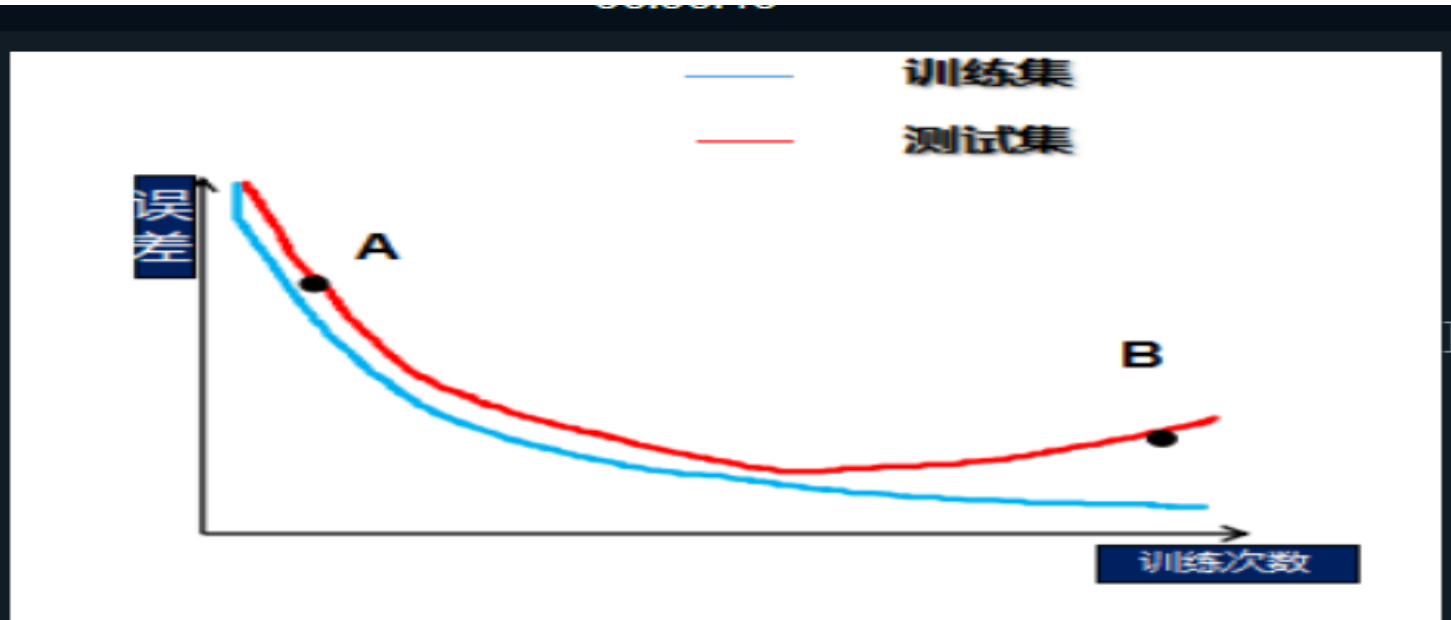
欠拟合



过拟合



刚刚好



1、 请问，图中A与B分别处于什么状态？

- ☐ A、欠拟合，欠拟合
- ☒ B、欠拟合，过拟合
- ☐ C、过拟合，欠拟合
- ☐ D、过拟合，过拟合



表 2.1 分类结果混淆矩阵

真实情况	预测结果	
	正例	反例
正例	$TP$ (真正例)	$FN$ (假反例)
反例	$FP$ (假正例)	$TN$ (真反例)

□ 查准率: 
$$P = \frac{TP}{TP + FP}$$

□ 查全率: 
$$R = \frac{TP}{TP + FN}$$

## 作业:

混淆矩阵中每个格子所代表的意义也很明显，意义如下：

真实预测	0	1
0	预测 0 正确的数量	预测 1 错误的数量
1	预测 0 错误的数量	预测 1 正确的数量

继续以癌症检测系统为例，为了方便，就用 1 表示患有癌测试，其中有 9978 条数据数据的真实类别是 1 却预测 1，有 8 条数据的真实类别

如果将正确看成是 True，错误看成是 False，0 看成是 Negative，1 看成是 Positive。然后将上表中的文字替换掉，混淆矩阵如下：

真实预测	0	1
0	TN	FP
1	FN	TP

如果我们把这些结果组成如

真实预测	0	1
0	9978	12
1	2	8



- 知识发现的基本过程
- 数据库中的知识发现处理过程模型
- 知识发现软件或工具的发展
- 知识发现项目的过程化管理
- 数据挖掘语言介绍





- KDD是一个多步骤的处理过程：
  - 1、问题定义
  - 2、数据采集
  - 3、数据预处理
  - 4、数据挖掘
  - 5、模式评估



### 数据挖掘实用案例分析——香水销售分析

从某电商网站上抓取到的香水产品销量数据，分析香水销售的影响因素，为香水销售商判定采购计划以及用户选择香水提供依据。

- 1、获取香水销售的相关数据
- 2、香水销售数据预处理
- 3、香水销售数据统计分析
- 4、影响香水销量的因素分析
- 5、香水适用场所关联分析
- 6、香水聚类分析
- 7、香水营销建议



- KDD是一个多步骤的处理过程：
  - 1、问题定义
  - 2、数据采集
  - 3、数据预处理
  - 4、数据挖掘
  - 5、模式评估





- KDD是为了在大量数据中发现有用的令人感兴趣的信息，因此发现何种知识就成为整个过程中第一个也是最重要的一个阶段。
- 在问题定义过程中，数据挖掘人员必须和领域专家以及最终用户紧密协作
  - 一方面了解相关领域的有关情况，熟悉背景知识，弄清用户要求，确定挖掘的目标等要求；
  - 另一方面通过对各种学习算法的对比进而确定可用的学习算法。后续的学习算法选择和数据集准备都是在此基础上进行的。



- KDD是一个多步骤的处理过程：
  - 1、问题定义
  - 2、**数据采集**
  - 3、数据预处理
  - 4、数据挖掘
  - 5、模式评估



- 数据采集是指从传感器和智能设备、企业在线系统、企业离线系统、社交网络和互联网平台等获取数据的过程。
- 数据包括 RFID 数据、传感器数据、用户行为数据、社交网络交互数据及移动互联网数据等各种类型的结构化、半结构化及非结构化的海量数据。
- 数据源的种类多，数据的类型繁杂，数据量大，并且产生的速度快，传统的数据采集方法完全无法胜任。所以，**数据采集技术**面临着许多技术挑战，一方面需要保证数据采集的可靠性和高效性，同时还要避免重复数据。

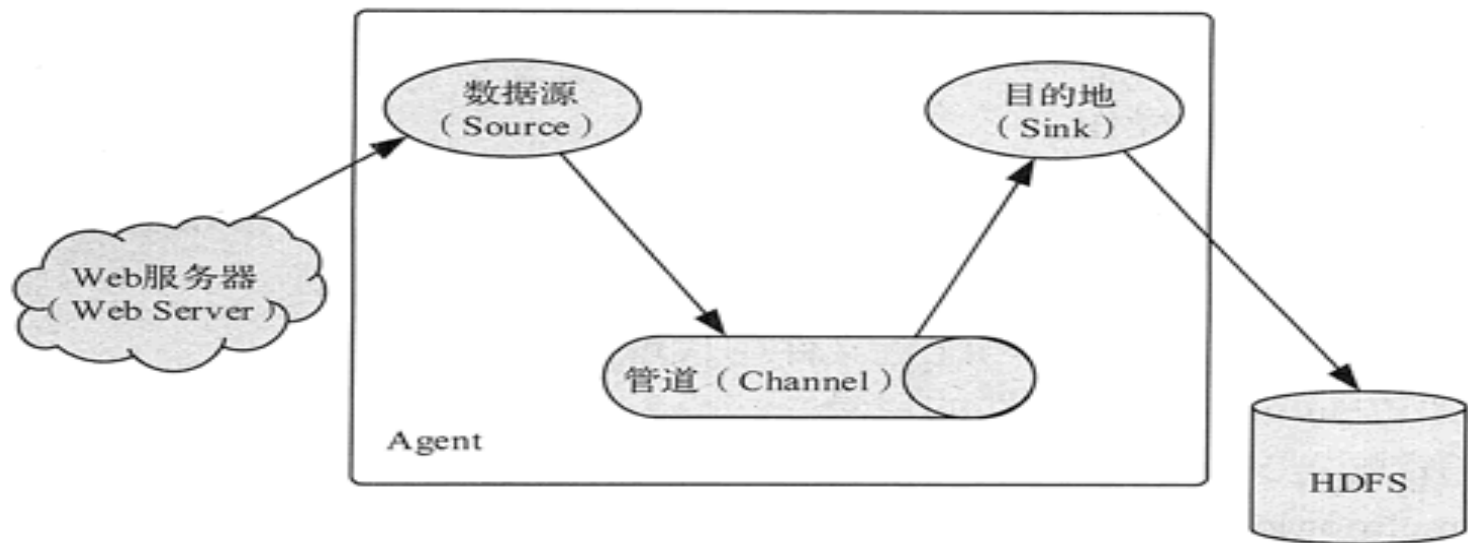


- 1. **数据库采集**：传统企业会使用传统的关系型数据库 MySQL 和 Oracle 等来存储数据。
- 2. **系统日志采集**：主要是收集公司业务平台日常产生的大量日志数据，供离线和在线的大数据分析系统使用。
- 3. **网络数据采集**：指通过网络爬虫或网站公开 API 等方式从网站上获取数据信息的过程。
- 4. **感知设备数据采集**：指通过传感器、摄像头和其他智能终端自动采集信号、图片或录像来获取数据。

# 通过系统日志采集大数据



- 目前使用最广泛的、用于系统日志采集的海量数据采集工具有 Hadoop 的 Chukwa、Apache Flume、Facebook 的 Scribe 和 LinkedIn 的 Kafka 等。以上工具均采用分布式架构，能满足每秒数百 MB 的日志数据采集和传输需求
- 以 Flume 系统为例对系统日志采集方法





- Flume 的用法：主要是编写一个用户配置文件。在配置文件当中描述 Source、Channel 与 Sink 的具体实现，而后运行一个 Agent 实例。

1) 从整体上描述 Agent 中 Sources、Sinks、Channels 所涉及的组件。

```
#Name the components on this agent
```

```
a1.sources = r1
```

```
a1.sinks = k1
```

```
a1.channels = c1
```



2) 详细描述 Agent 中每一个 Source、Sink 与 Channel 的具体实现。

对于 Source，需要指定是接收文件、接收 HTTP，还是接收 Thrift。

对于 Sink，需要指定结果是输出到 HDFS 中，还是 HBase 中等。

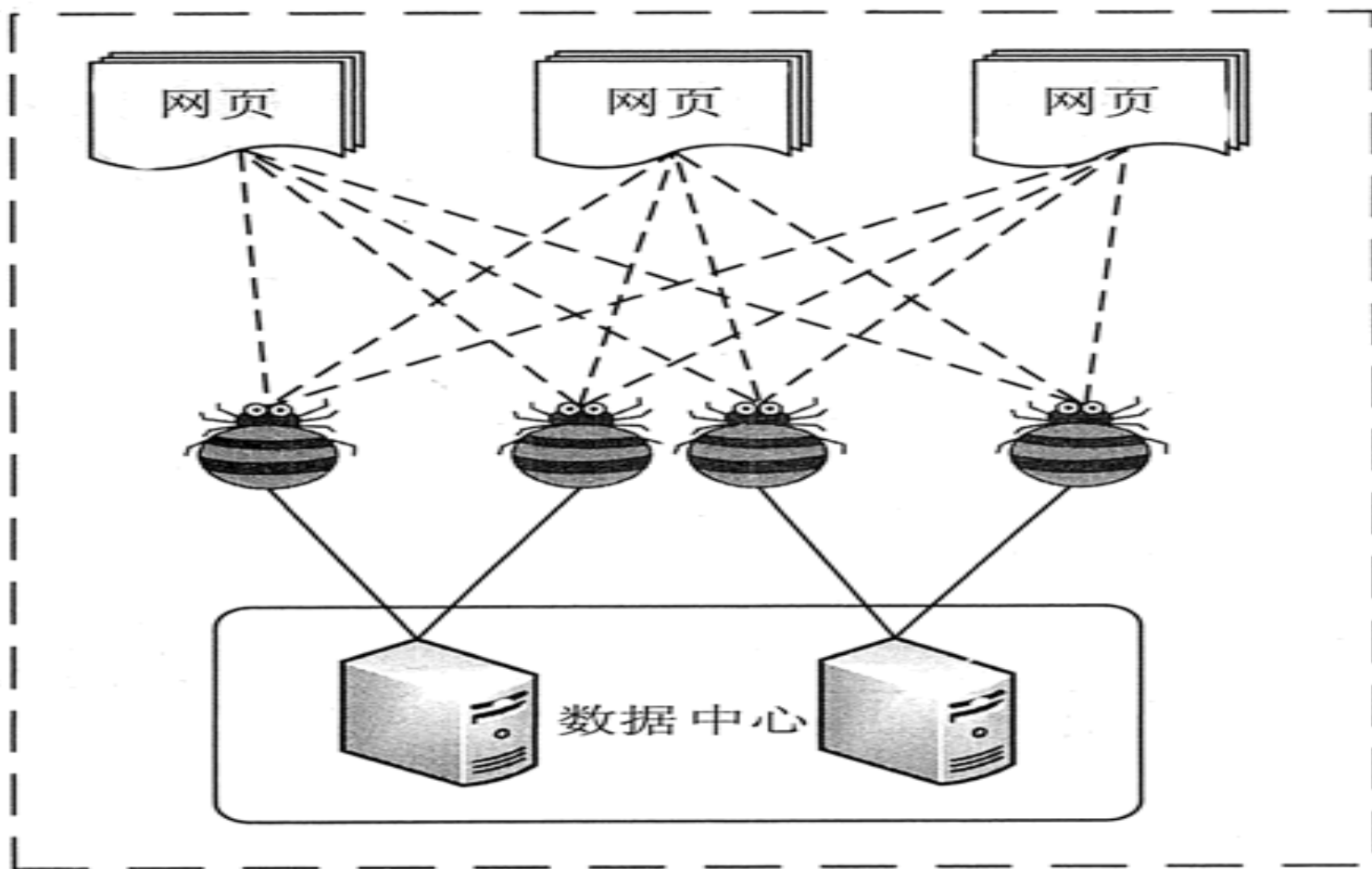
对于 Channel，需要指定格式是内存、数据库，还是文件等。

3) 通过 Channel 将 Source 与 Sink 连接起来。

4) 启动 Agent 的 shell 操作。



- **网络爬虫**，自动地抓取 Web 信息的程序或者脚本，为搜索引擎和大数据分析提供数据来源。



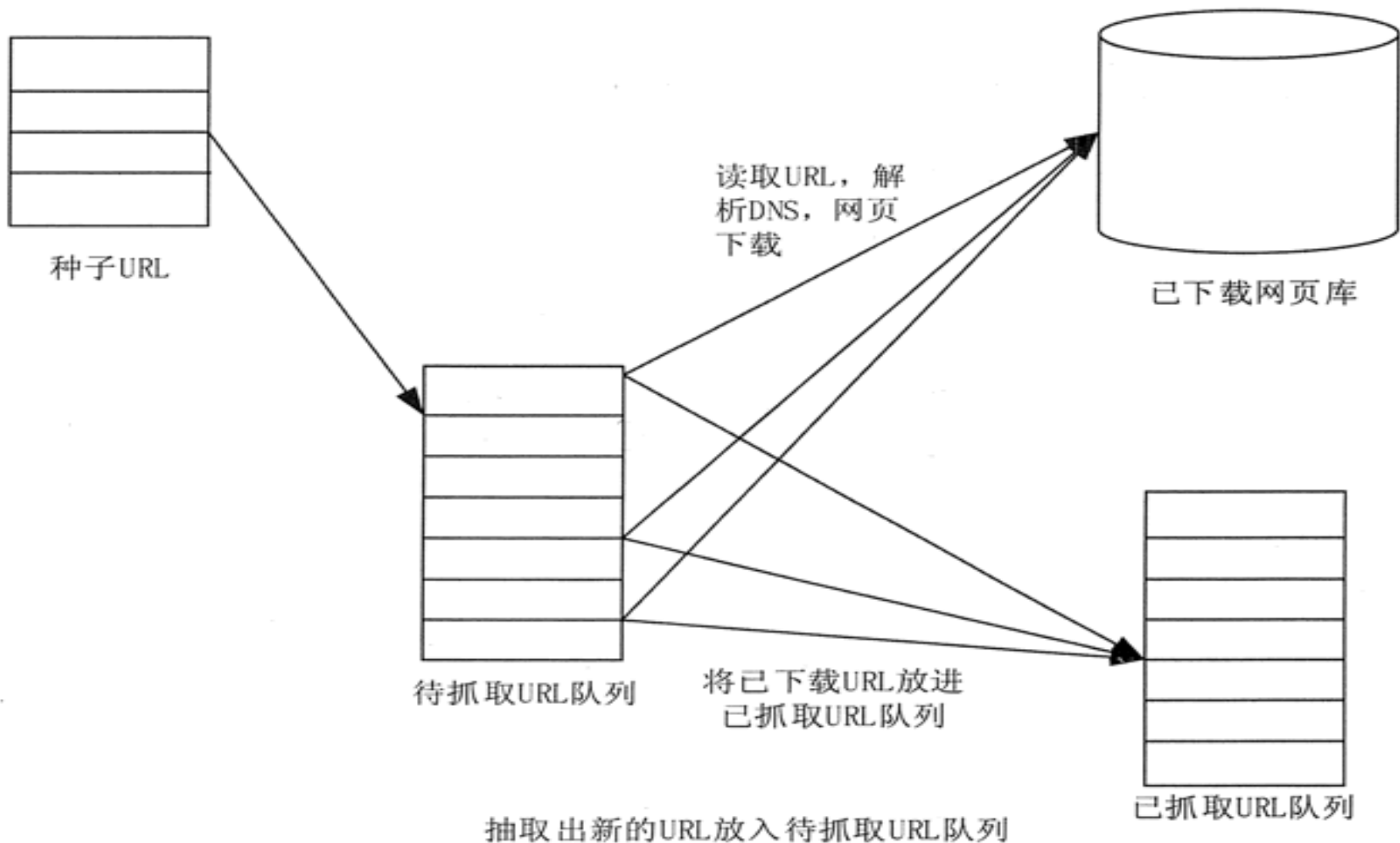


# 网络爬虫工作流程



- 1) 首先选取一部分种子 URL。
- 2) 将这些 URL 放入待抓取 URL 队列。
- 3) 从待抓取 URL 队列中取出待抓取 URL，解析 DNS，得到主机的 IP 地址，并将 URL 对应的网页下载下来，存储到已下载网页库中。
- 4) 分析已抓取 URL 队列中的 URL，分析其中的其他 URL，并且将这些 URL 放入待抓取 URL 队列，从而进入下一个循环。

# 网络爬虫工作流程





- KDD是一个多步骤的处理过程：
  - 1、问题定义
  - 2、数据采集
  - 3、数据预处理
  - 4、数据挖掘
  - 5、模式评估



- KDD是一个多步骤的处理过程，一般分为
  - 问题定义、
  - 数据采集、
  - 数据预处理（清洗、转换、描述、选择、抽取）
  - 数据挖掘、
  - 模式评估



- 数据预处理（清洗、转换、描述、选择、抽取）
- 数据清洗的主要任务：
  - 1、数据缺失的处理
  - 2、噪音数据的处理
  - 3、数据不一致的识别与处理

## 第二章 知识发现过程与应用结构



### ■ 数据清洗的主要任务

#### ■ 1、数据缺失的处理：删除、平均值、分类、拟合……

客户编号	年龄（岁）	性别	年收入(万元)	婚姻	豪华车
1	<30	女	86	已婚	否
2	<30	男	65	单身	否
3	<30	男	90	离异	否
4	<30	女		已婚	否
5	30~50	女	82	已婚	是
6	30~50	男	91	已婚	是
7	30~50	女	200	离异	是
8		女	40	单身	否
9	30~50	男	20	离异	否
10	>50	女	96	离异	否
11	>50	女	80	单身	否
12	>50	男	50	单身	是
13	>50	女	80	离异	否
14	>50	男	92	离异	是



- 将缺失值的对象信息删除 → 损失有用信息
- 将缺失值用同一个值替代 → 左右分析结果

由类型为“否”的客户  
的年收入的均值来填补：

$$(86+65+90+40+20+96+80+80)/8=58$$

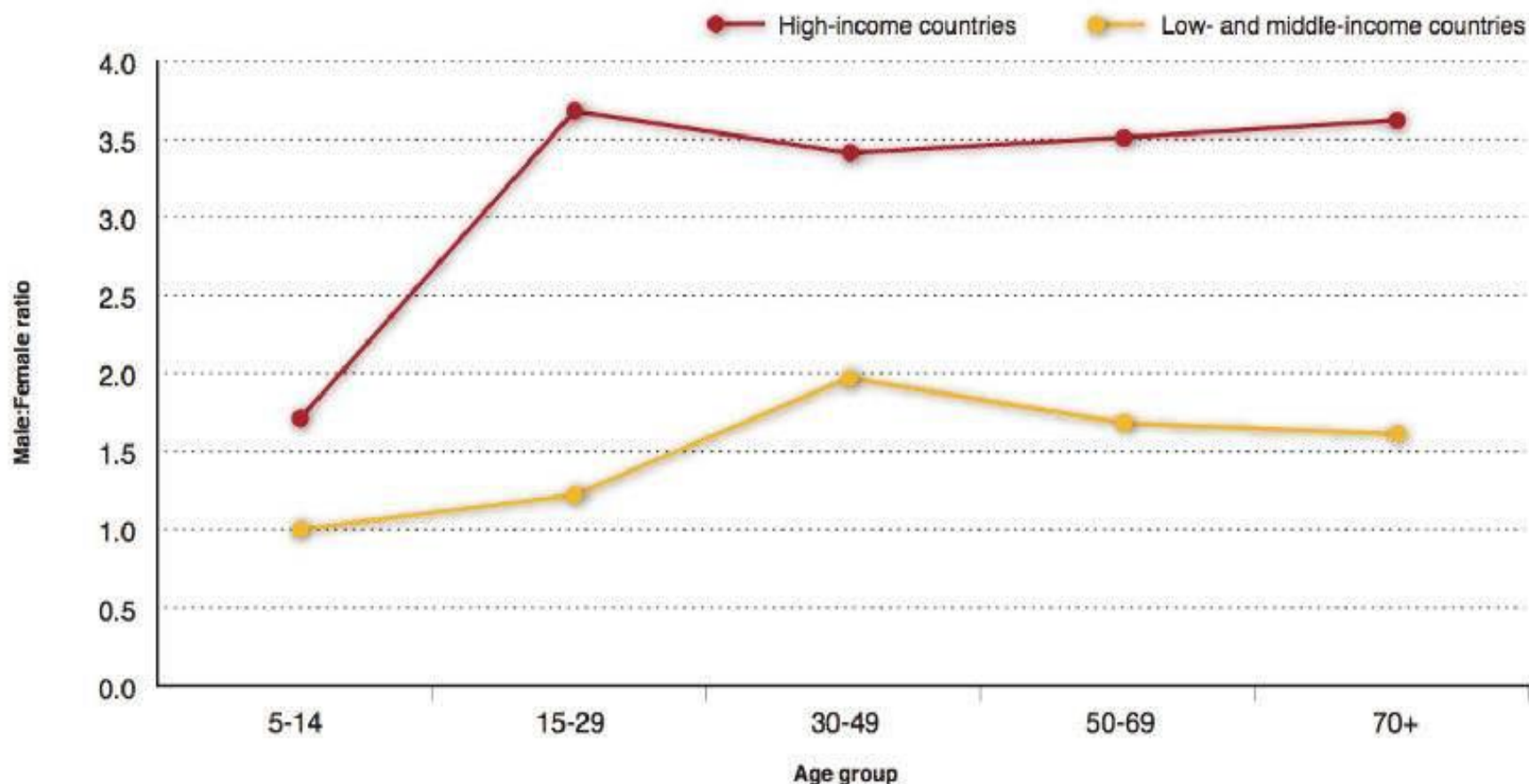
对于离散型属性问题或定性属性，  
用众数代替均值。





预测第4个客户的年收入:

- 将年收入作为目标属性, 利用其他对象构建预测模型。







## ■ 缺失数据的处理：利用分箱法

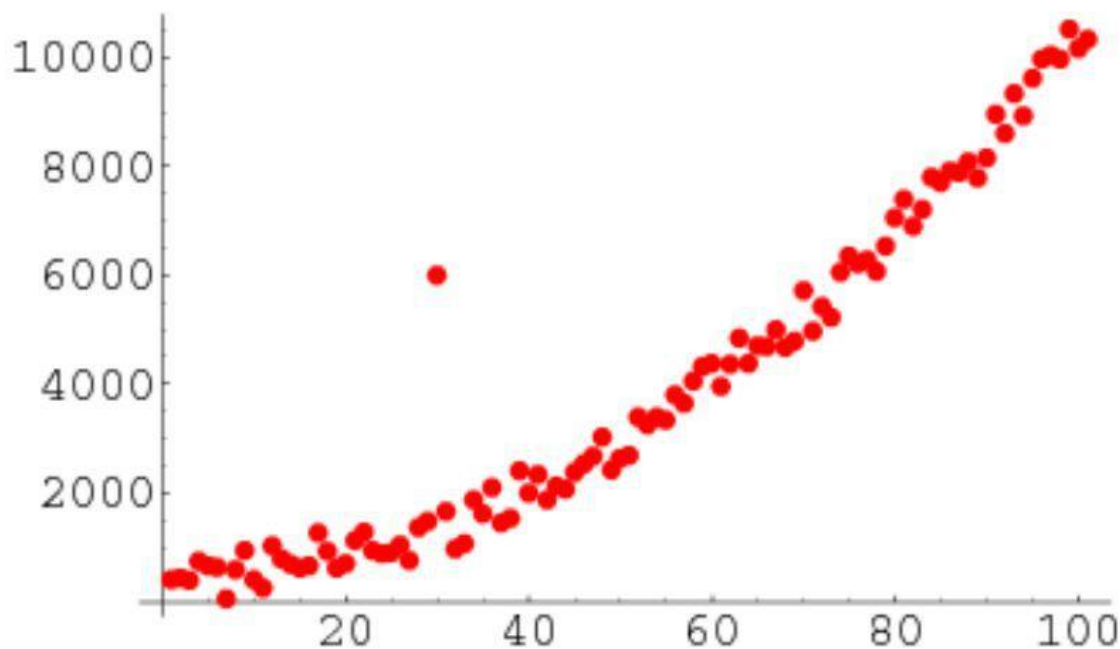
客户编号	年龄（岁）	性别	年收入(万元)	婚姻	豪华车
1	<30	女	86	已婚	否
2	<30	男	65	单身	否
3	<30	男	90	离异	否
4	<30	女		已婚	否
5	30~50	女	82	已婚	是
6	30~50	男	91	已婚	是
7	30~50	女	200	离异	是
8		女	40	单身	否
9	30~50	男	20	离异	否
10	>50	女	96	离异	否
11	>50	女	80	单身	否
12	>50	男	50	单身	是
13	>50	女	80	离异	否
14	>50	男	92	离异	是



- 将取值进行排序: 20, 40, 50, 58, 65, 80, 80, 82, 86, 90, 91, 92, 96, 200。
- 利用等频率分箱, 每箱3个值: 4个箱分别为 [20, 40, 50], [58, 65, 80, 80], [82, 86, 90], [91, 92, 96, 200]。
- 每个箱中的值可以用均值或者中位数代替: 若用中位数代替, 则变为 [40, 40, 40], [73, 73, 73, 73], [86, 86, 86], [94, 94, 94, 94]。



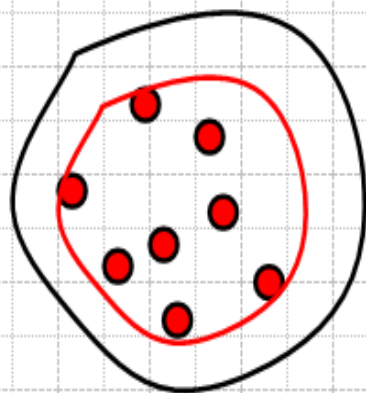
- 数据清洗的主要任务
- 1、数据缺失的处理：删除、平均值、分类、拟合……
- 2、噪声的处理：识别出噪声、利用其他非噪音数据降低噪音的影响，起到平滑的作用



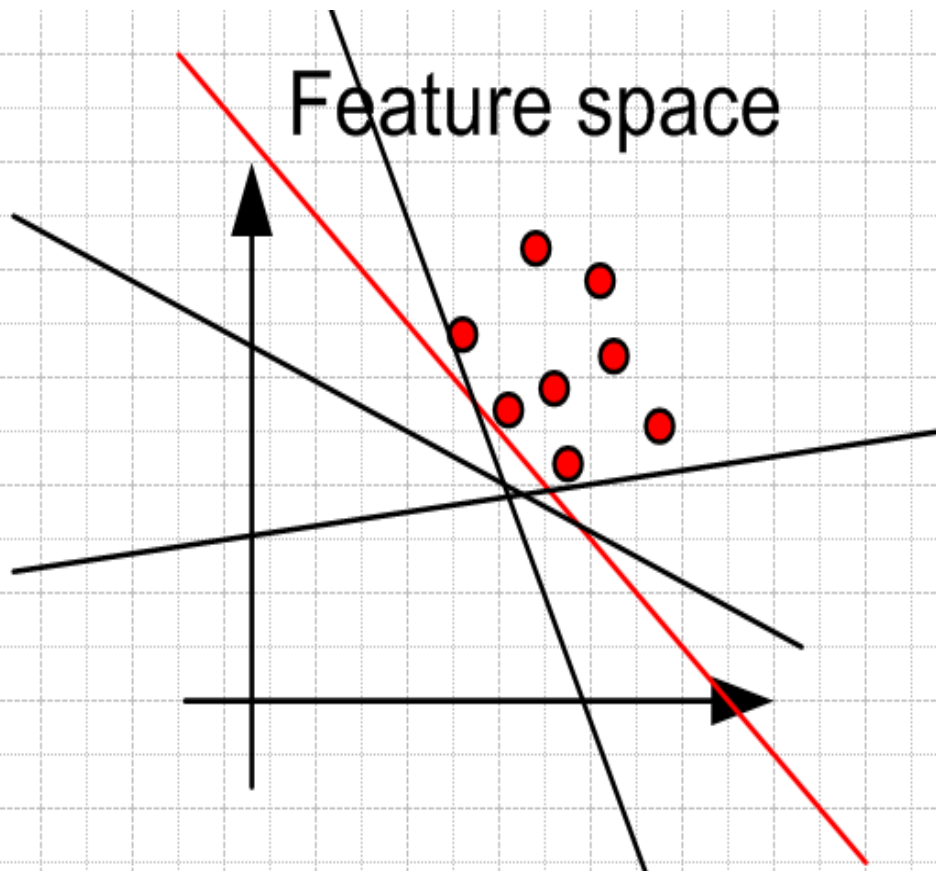


- 噪声数据：可采用OneClassSVM、Isolation Forest、Local Outlier Factor (LOF)等方法识别噪声。

Original space

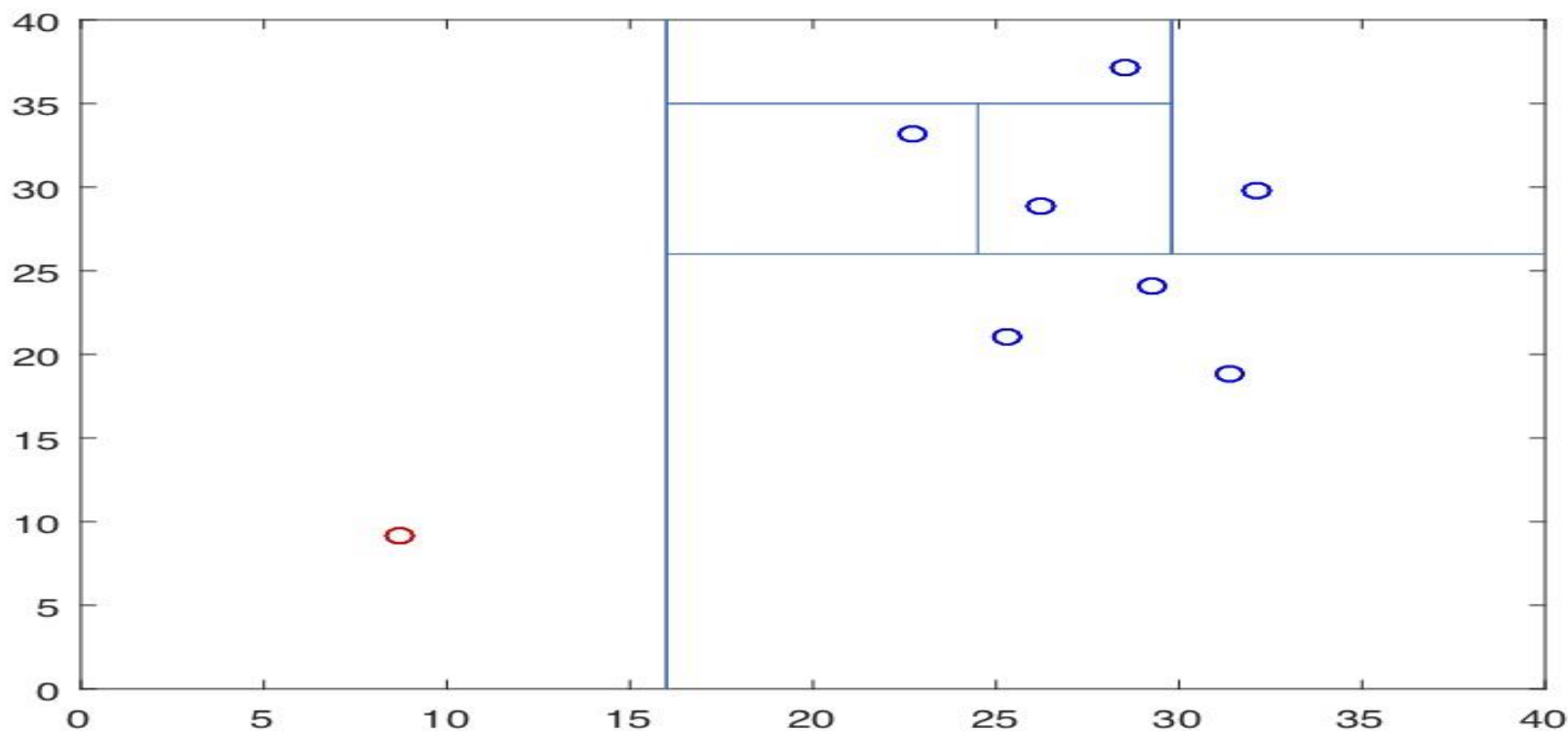


Feature space





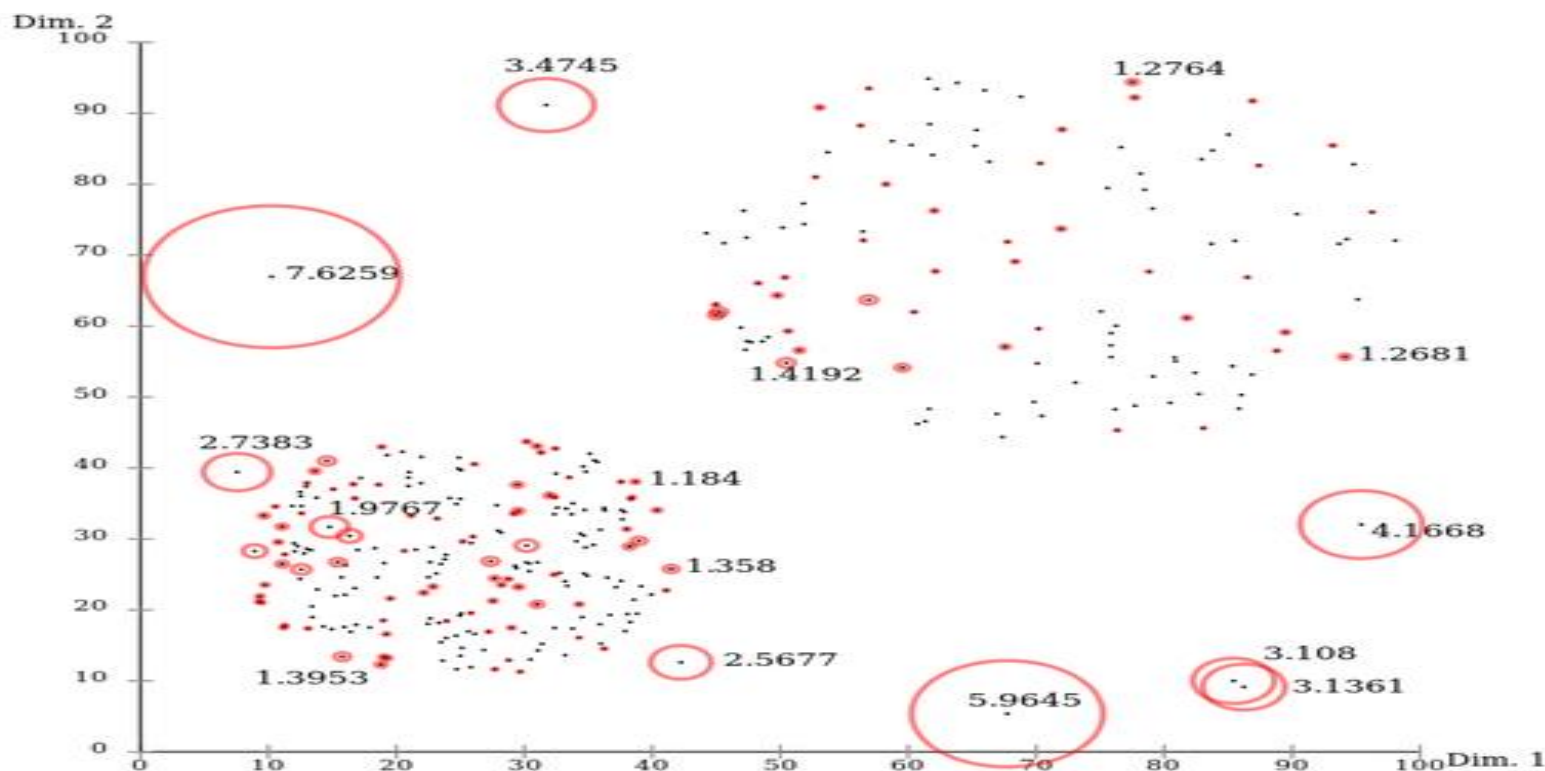
- 噪声数据：是指数据中存在着错误或异常（偏离期望值）的数据。可采用OneClassSVM、Isolation Forest、Local Outlier Factor (LOF)等方法识别噪声。







- 噪声数据：是指数据中存在着错误或异常（偏离期望值）的数据。可采用OneClassSVM、Isolation Forest、Local Outlier Factor (LOF)等方法识别噪声。





## ■ 噪声数据的处理：利用分箱法平滑噪声。

1. Bin 方法：通过利用应被平滑数据点的周围点（近邻），对一组排序数据进行平滑。排序后的数据被分配到若干桶（称为 Bins）中。

- 排序后价格：4, 8, 15, 21, 21, 24, 25, 28, 34
- 划分为等高度Bin：
  - Bin1: 4, 8, 15
  - Bin2: 21, 21, 24
  - Bin3: 25, 28, 34
- 根据Bin均值进行平滑：
  - Bin1: 9, 9, 9
  - Bin2: 22, 22, 22
  - Bin3: 29, 29, 29
- 根据Bin边界进行平滑：
  - Bin1: 4, 4, 15
  - Bin2: 21, 21, 24
  - Bin3: 25, 25, 34



- 数据预处理（清洗、**转换**、描述、选择、抽取）
- 数据描述的主要任务：
  - 1、数据立方体聚集、
  - 2、归约
  - 3、数据压缩
  - 4、数值压缩
  - 5、离散化





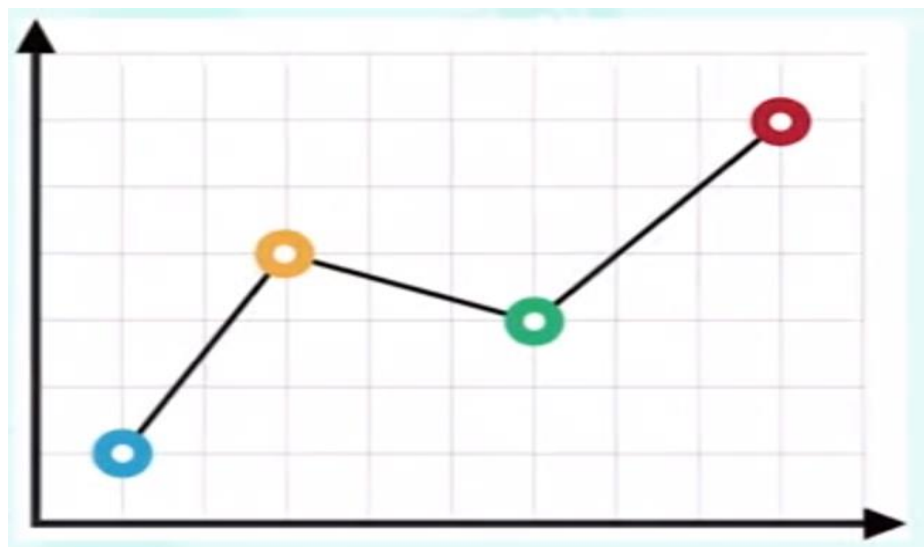
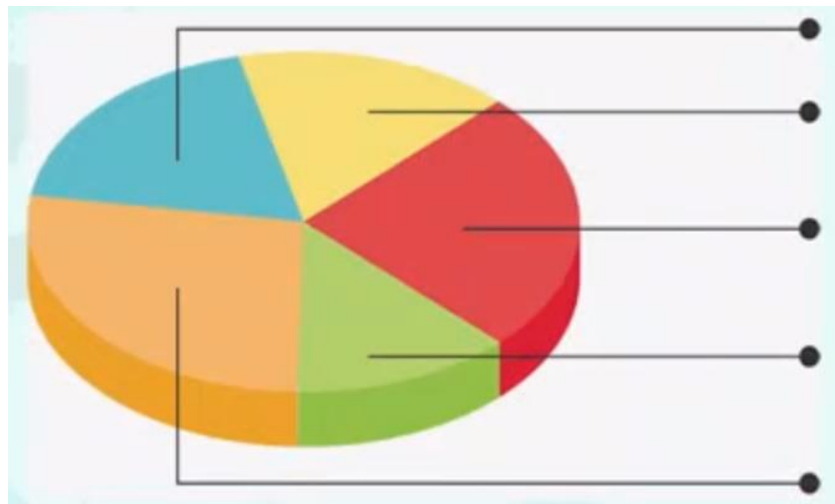
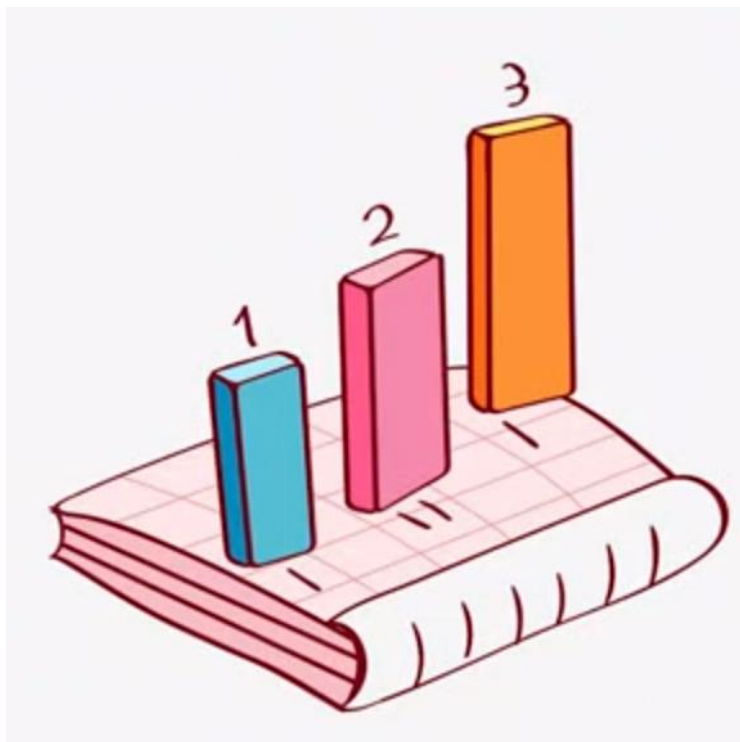
- **数据转换**就是将数据进行转换或归并，从而构成一个适合数据处理的描述形式。
- 数据转换包含以下处理内容：
  - 1) **合计处理**：对数据进行总结或合计操作。例如，每天的数据经过合计操作可以获得每月或每年的总额。这一操作常用于构造数据立方或对数据进行多粒度的分析。
  - 2) **数据泛化处理**：用更抽象（更高层次）的概念来取代低层次或数据层的数据对象。例如，街道属性可以泛化到城市、国家，数值型的属性，如年龄属性，可以映射到更高层次的概念

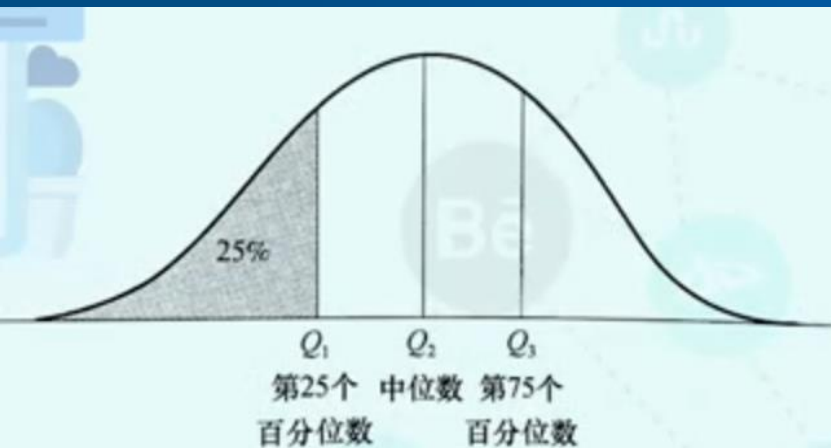


- 3) **规格化处理**: 将有关属性数据按比例投射到特定的小范围之中。例如, 将工资收入属性值映射到 0 到 1 范围内。
- 4) **属性构造处理**: 根据已有属性集构造新的属性, 以帮助数据处理过程。
- 三种规格化方法:
  - 1. **最大最小规格化方法**:  
$$\frac{(\text{待转换属性值} - \text{属性最小值})}{(\text{属性最大值} - \text{属性最小值})} * (\text{映射区间最大值} - \text{映射区间最小值}) + \text{映射区间最小值}$$
  - 2. **零均值规格化方法**:  
$$(\text{待转换属性值} - \text{属性平均值}) / \text{属性方差}$$
  - 3. **十基数变换规格化方法**: 科学计数

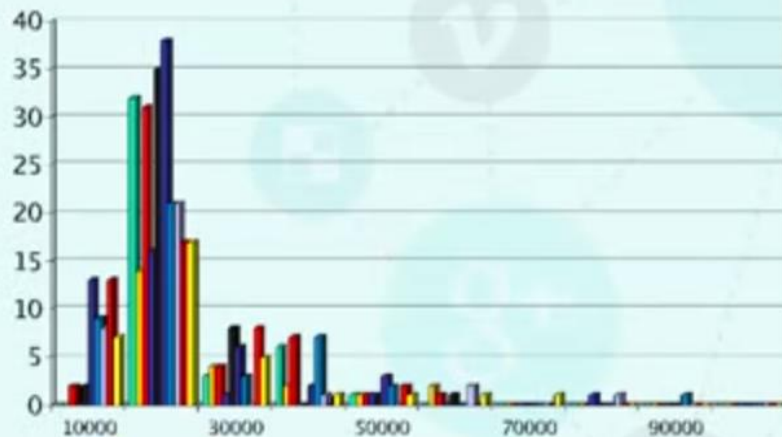


- 数据预处理（清洗、转换、描述、选择、抽取）
- 数据描述的主要任务：
  - 1、数据的基本统计
  - 2、把握数据的全貌
  - 3、识别噪声或离群数据点
  - 4、对数据进行可视化。

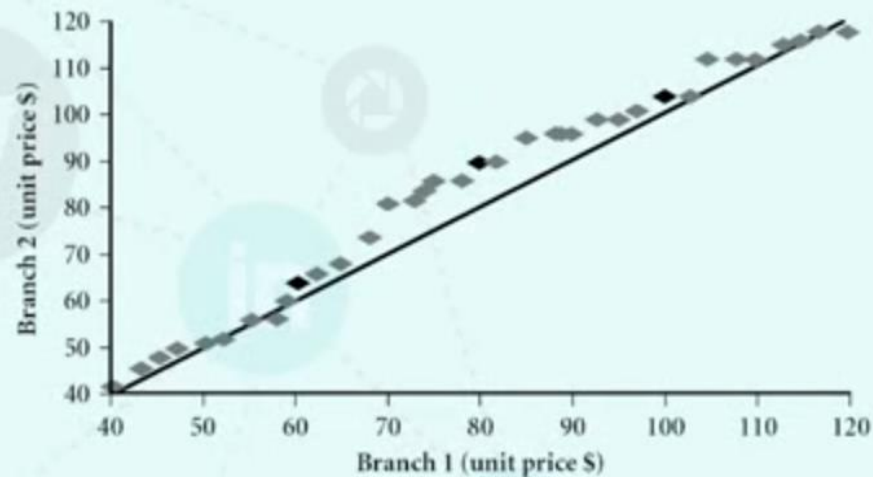




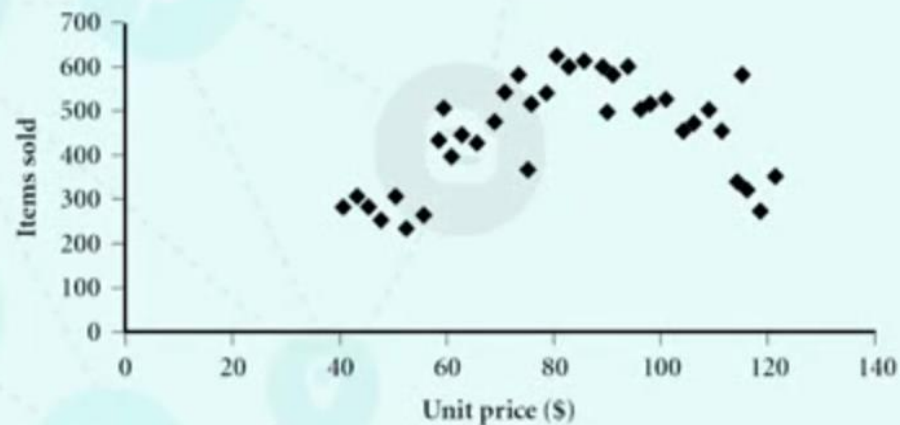
## 分位数图



## 直方图



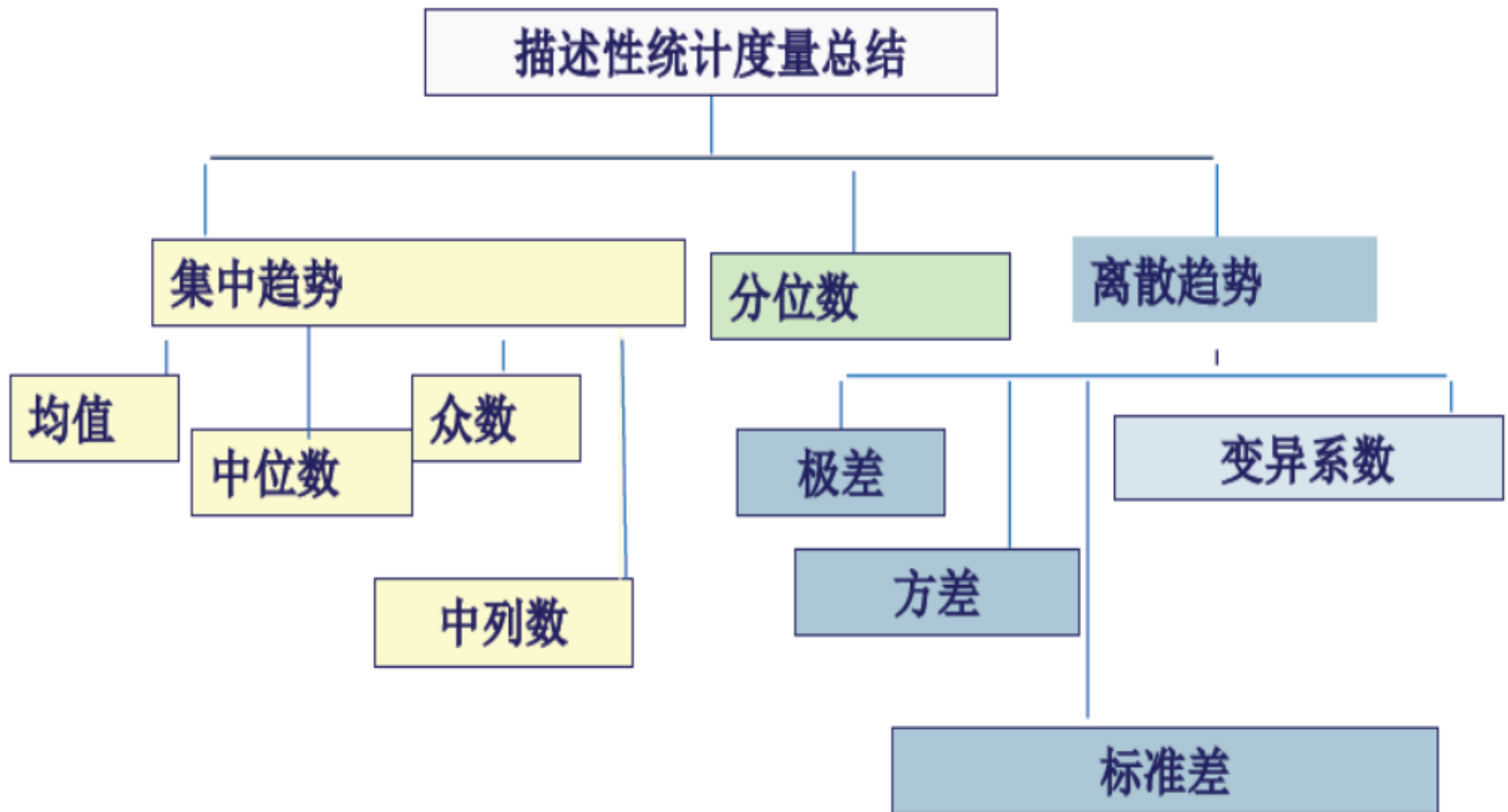
## 分位数-分位数图



## 散点图



## ■ 数据的基本统计描述







## 均值

· 令  $x_1, x_2, \dots, x_N$  为某数值属性x(如salary)的N个观测值。

· 计算公式: x的平均值=  $\bar{x} = \frac{\sum_{i=1}^N x_i}{N}$

假设我们有工资水平 (salary) 的如下值(以千美元为单位), 按递增次序显示:

· 30,31,47,50,52,52,56,60,63,70,70,110。

则, 均值=(30+31+47+50+52+52+56+60+63+70+70+110)/12=58



## 拓展1

- 有的时候，属性x的每个观测值的权重是不一样的。
- 权重反映它们所依附的对应值的意义、重要性或出现的频率。

## 加权平均值

· 权计算公式：加权平均值 = 
$$\frac{\sum_{i=1}^N x_i \cdot w_i}{\sum_{i=1}^N w_i}$$





### 拓展2

抵消少数极端值的影响，可以使用截尾均值(trimmedmean)，有时也翻译成修剪的平均数。

我们可以对工资的观测值排序，并计算均值之前去掉高端和低端的2%。

注意：不要在两端截去太多(如20%),因为这可能导致丢失有价值的信息。



# 中位数

- 地位：对于倾斜(非对称)数据，数据中心的更好度量是中位数(median)
- 定义：中位数是有序数据值的中间值。它是把数据较高的一半与较低的一半分开的值。



# 中位数

计算方法：

假设给定某属性 $X$ 的 $N$ 个值按递增序排序

- 如果 $N$ 是奇数，中位数是该有序集的中间值；
- 如果 $N$ 是偶数，中位数是最中间的两个值和它们之间的任意值。在 $X$ 是数值属性的情况下，中位数可取作最中间两个值的平均值。



### 中位数

假设1: 我们有工资水平 (salary) 的如下值(以千美元为单位), 按递增次序显示:

· 30,31,47,50,52,52,56,60,63,70,70,110。

则, 中位数 $= (52+56)/2=54$





- 众数：mode，是集合中出现最频繁的值

假设1：我们有工资水平（salary）的如下值(以千美元为单位)，按递增次序显示：

· 30,31,47,50,52,52,56,60,63,70,70,110。

众数=52,70



### 计算方法:

- 具有一个、两个、三个众数的数据集合分别称为单峰、双峰和三峰。
- 具有两个或更多众数的数据集是多峰。如果每个数据值仅出现一次，则它没有众数。



### 中列数

- 用途：评估数值数据的中心趋势。
- 定义：最大和最小值的平均值。

假设1：我们有工资水平（salary）的如下值(以千美元为单位)，按递增次序显示：

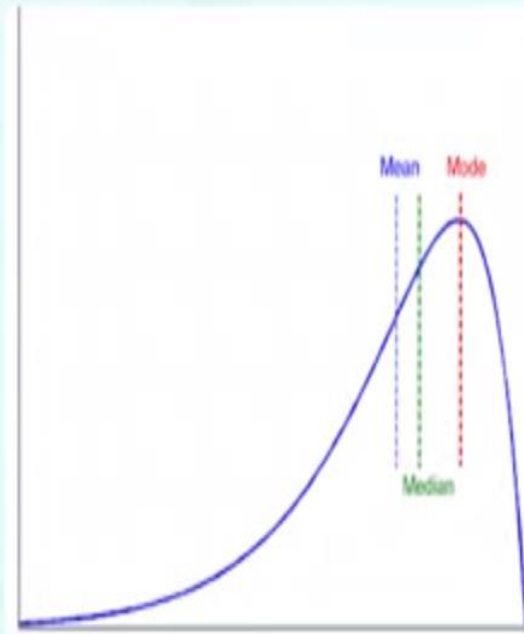
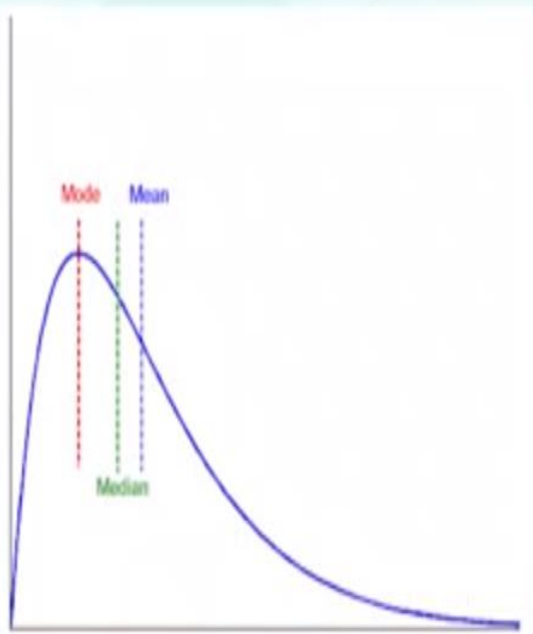
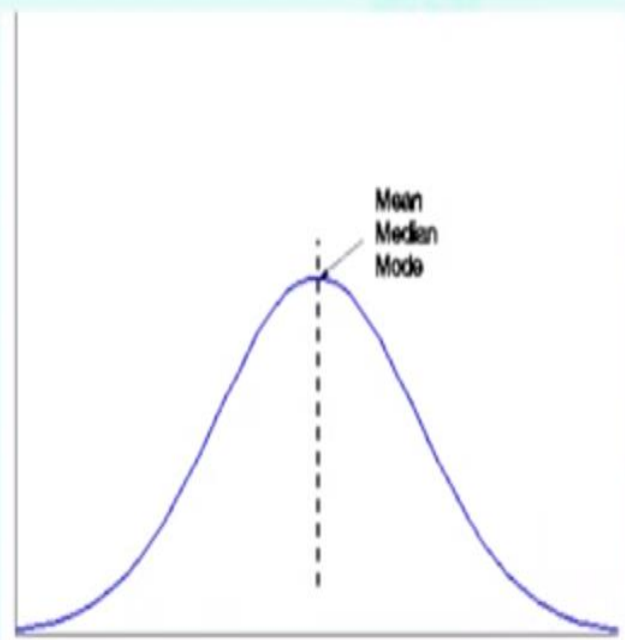
- 30,31,47,50,52,52,56,60,63,70,70,110。

$$(30+110) / 2 = 70$$

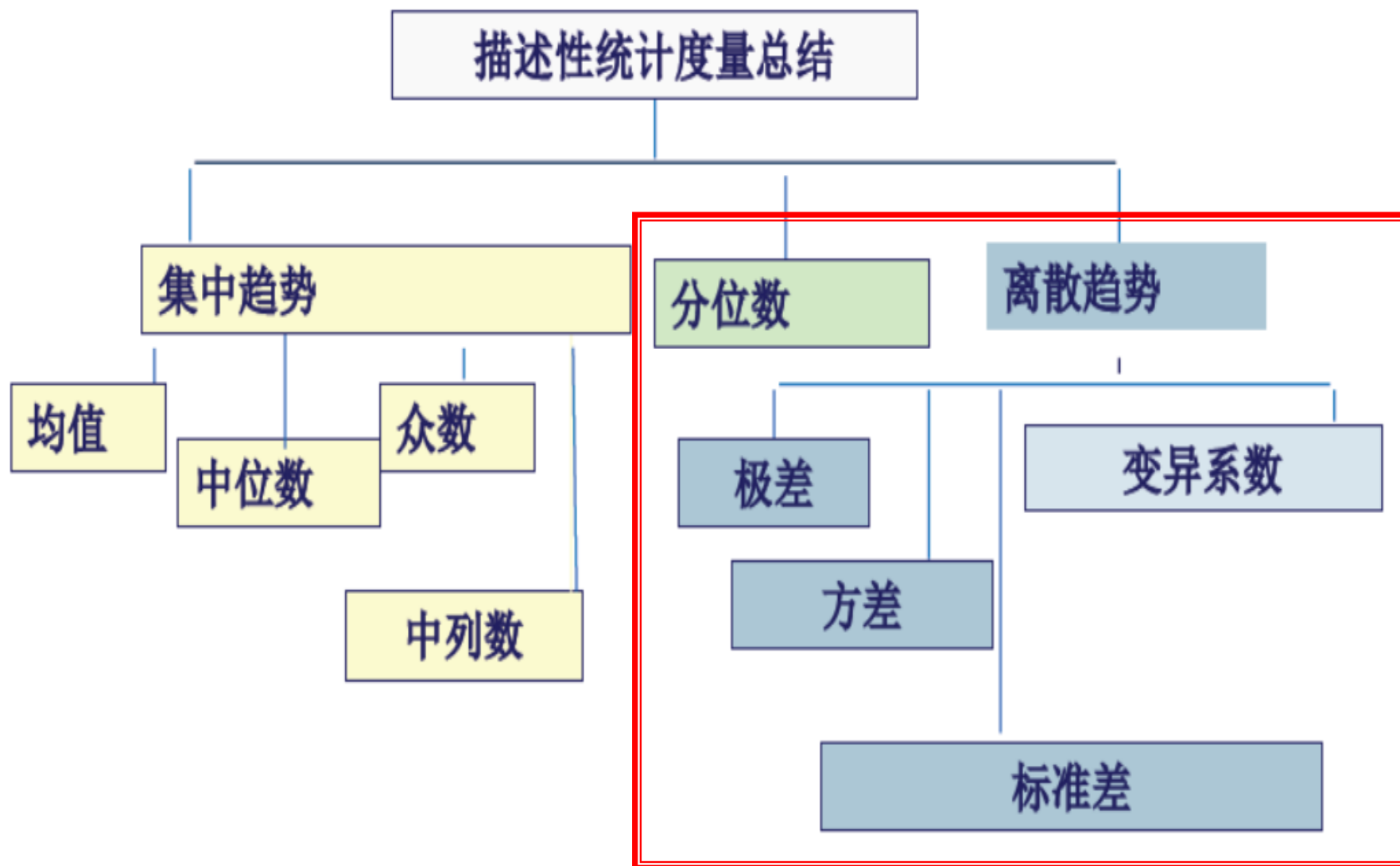


### 众数对称/偏斜数据

- 中位数，均值，众数：对称，正倾斜和负倾斜数据









- 数据的离散趋势：
  - 1、极差、
  - 2、分位数、
  - 3、分位数极差
  - 4、五数概括
  - 5、箱线图
  - 6、方差
  - 7、标准差
  - 8、变异系数



## ■ 极差 (range) : 也叫全距, 整个距离的意思

设  $x_1, x_2, \dots, x_N$  设是某数值属性X上的观测的集合。  
该集合的极差是最大值与最小值之差。

$$\text{Range}(X) = \text{Max}(X) - \text{Min}(X)$$

- 极差是数据中最大与最小间的差距;
- 是衡量数据变异程度最简单的描述;
- 极差对最大与最小数据的值的敏感性很强。



## 第P个百分位数

定义：至少有P%的数据项小于或等于这个值，且至少有(100-P)%的数据项大于或等于这个值。

计算方法：

- 递增排序；
- 计算位置的指数  $i = (p/100)n$ ；
- 如果i不是整数，将其向上取整；
- 如果i是整数，则p分位数为第i项与第i+1项的数据的平均值。



## 中位数

Step1: 将数据集按递增顺序排列

Step2: 判断

$$m_e = \begin{cases} x_{(\frac{n+1}{2})}, & \text{当属性数值为奇数} \\ \frac{1}{2} (x_{(\frac{n}{2})} + x_{(\frac{n}{2}+1)}), & \text{当属性数值为偶数} \end{cases}$$

中位数是分位数的一个特例。





## 案例：房屋租金

425	430	430	435	435	435	435	435	440	440
440	440	440	445	445	445	445	445	450	450
450	450	450	450	450	460	460	460	465	465
465	470	470	472	475	475	475	480	480	480
480	485	490	490	490	500	500	500	500	510
510	515	525	525	525	535	549	550	570	570
575	575	580	590	600	600	600	600	615	615

90%分位数

- 先对整个序列进行排序，再找中位数



425	430	430	435	435	435	435	435	440	440
440	440	440	445	445	445	445	445	450	450
450	450	450	450	450	460	460	460	465	465
465	470	470	472	475	475	475	480	480	480
480	485	490	490	490	500	500	500	500	510
510	515	525	525	525	535	549	550	570	570
575	575	580	590	600	600	600	600	615	615

## 90%分位数

- $i = (p/100)n = (90/100)70 = 63$
- 90th Percentile =  $(580 + 590)/2 = 585$



# 四分位数

- 特定的百分位数;
- 第一个四分位数为25%百分位数;
- 第二个四分位数为50%百分位数, 即中位数;
- 第三个四分位数为75%百分位数。





## 案例：房屋租金

425	430	430	435	435	435	435	435	440	440
440	440	440	445	445	445	445	445	450	450
450	450	450	450	450	460	460	460	465	465
465	470	470	472	475	475	475	480	480	480
480	485	490	490	490	500	500	500	500	510
510	515	525	525	525	535	549	550	570	570
575	575	580	590	600	600	600	600	615	615

### 第三个四分位数

- Third quartile = 75th percentile

- $i = (p/100)n = (75/100)70 = 52.5 \approx 53$



## 案例：房屋租金

425	430	430	435	435	435	435	435	440	440
440	440	440	445	445	445	445	445	450	450
450	450	450	450	450	460	460	460	465	465
465	470	470	472	475	475	475	480	480	480
480	485	490	490	490	500	500	500	500	510
510	515	525	525	525	535	549	550	570	570
575	575	580	590	600	600	600	600	615	615

### 第三个四分位数

- Third quartile = 75th percentile
- $i = (p/100)n = (75/100)70 = 52.5 \approx 53$
- Third quartile = 525



## 分布的五数概括(five-number summary)

- 包含：中位数( $Q_2$ )、四分位数 $Q_1$ 和 $Q_3$ 、最小和最大观测值；
- 排序：按次序 Minimum,  $Q_1$ , Median,  $Q_3$ , Maximum写出；
- 特点：最能反映数据重要特征的5个数。



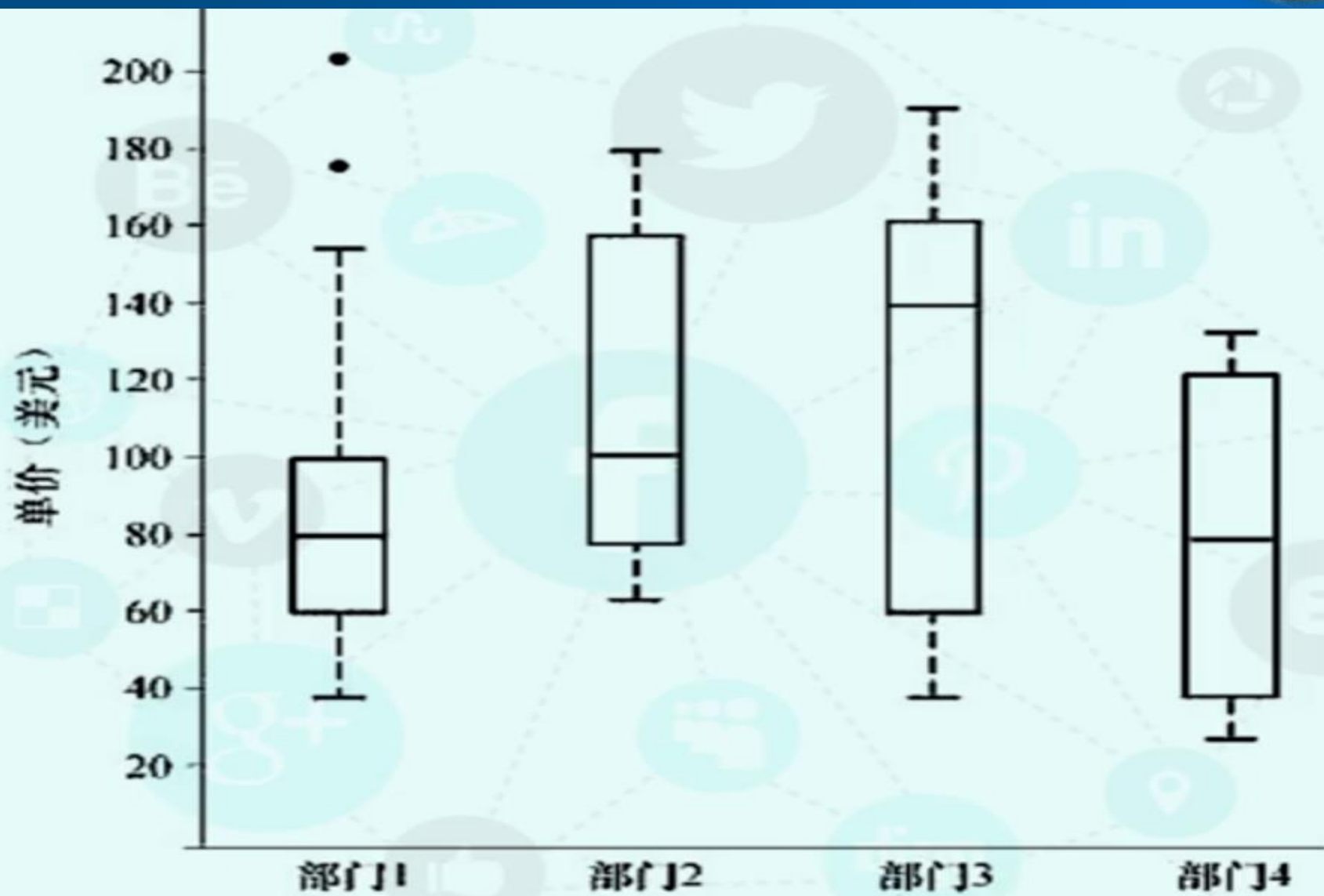
# 箱线图，又称盒图

作用：体现五数概括

特征：

- 在盒图中，第一个四分位数和第三个四分位数确定了盒子的底部和顶部；
- 盒子中间的粗线就是中位数所在的位置；
- 由盒子向上向下伸出的垂直部分称为触须，表示数据的散布范围，通常最远点是 $1.5IQR$ 。



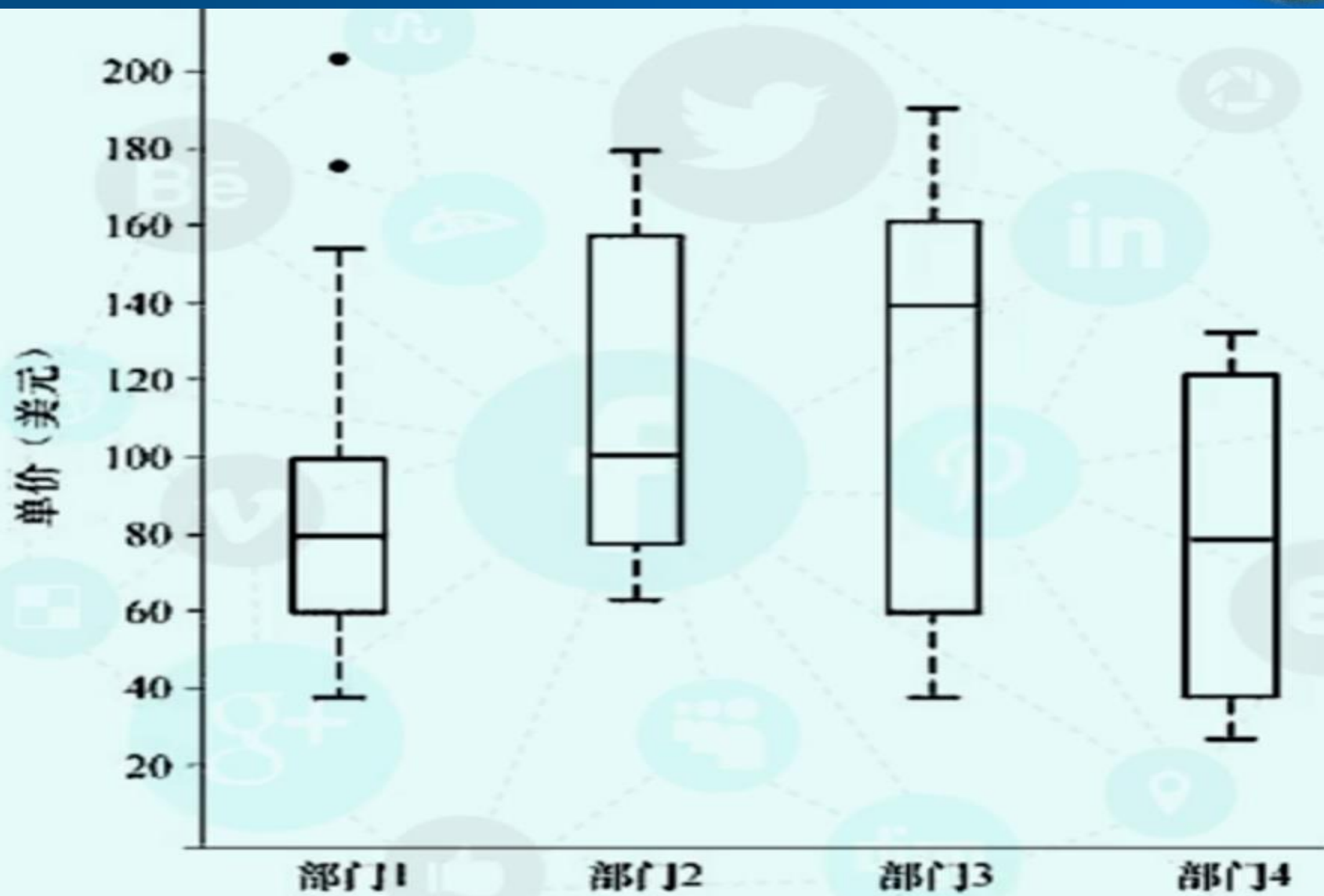




## 四分位极差

- 定义：第1个和第3个四分位数之间的距离；
- 公式： $IQR = Q3 - Q1$ ；
- 特点：该距离是散布的一种简单度量，能够克服极端值的影响；

■ 识别可疑离群点的通常规则：如果数值落在第3个四分位数之上或第1个四分位数之下至少  $1.5 * IQR$  处的值，则被看作是可疑的离群点。







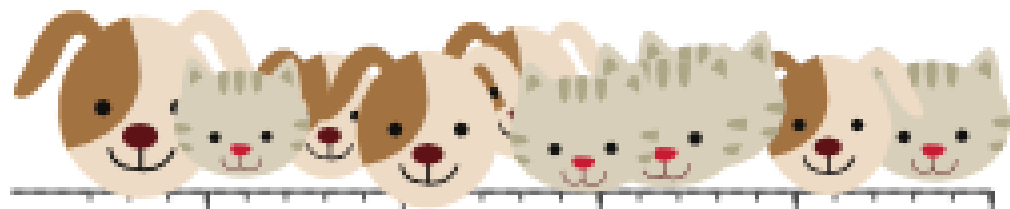
- KDD是一个多步骤的处理过程，一般分为
- 问题定义、
- 数据采集、
- 数据预处理（清洗、转换、描述、**选择、抽取**）
- 数据挖掘、
- 模式评估、
- 等基本阶段。



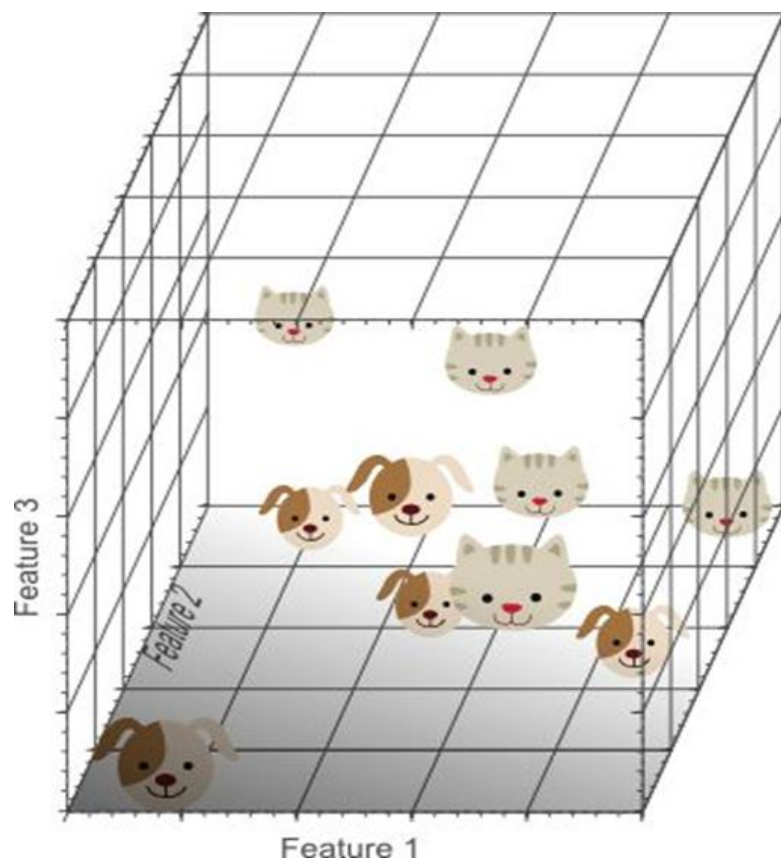
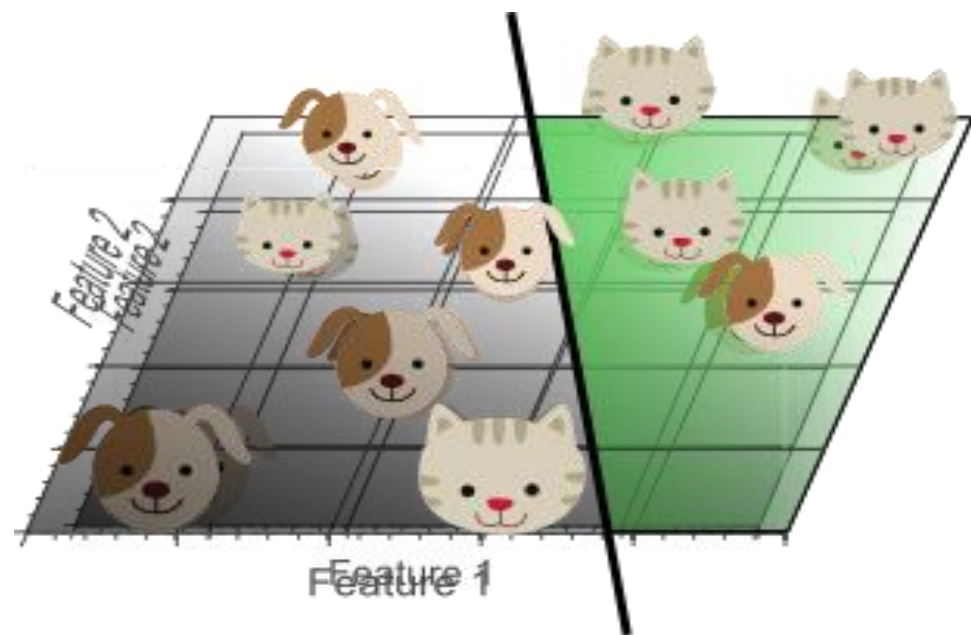
### ■ 为什么要做特征选择：

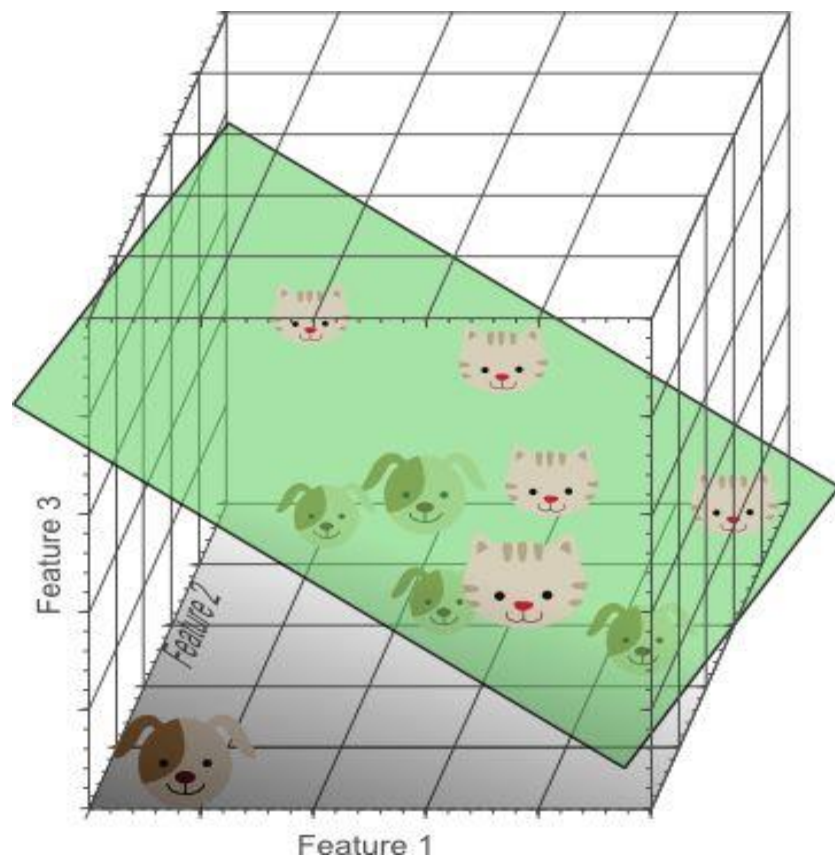
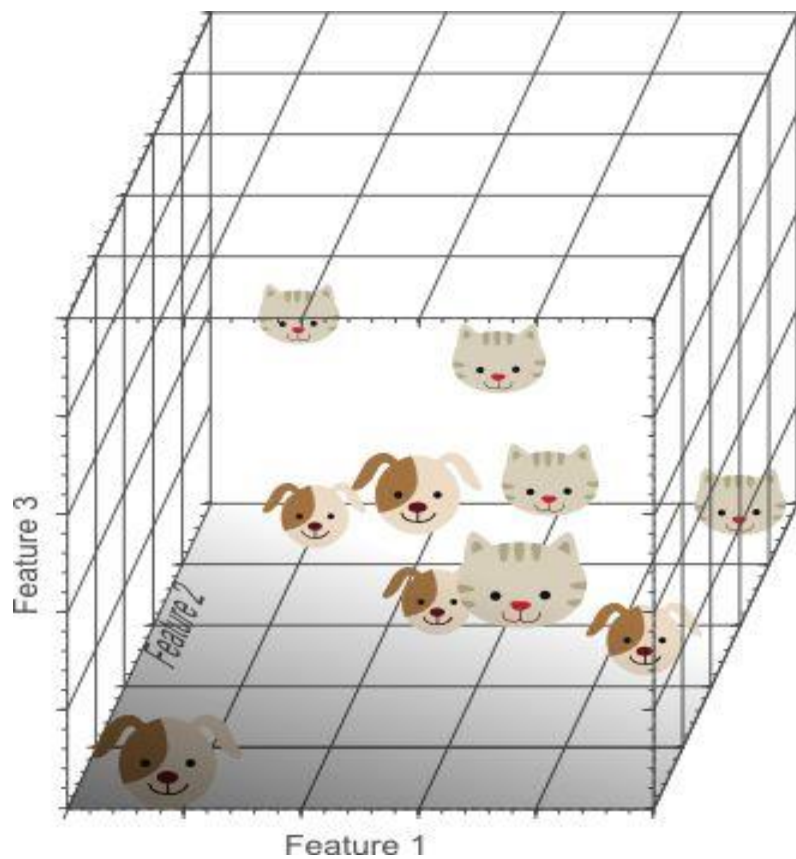
- 特征数量可能较多
- 可能存在不相关的特征
- 特征之间存在相关性，但特征个数越多，分析特征、训练模型所需的时间就越长，模型也会越复杂。
- 特征个数越多，容易引起“维度灾难”，其推广能力会下降。

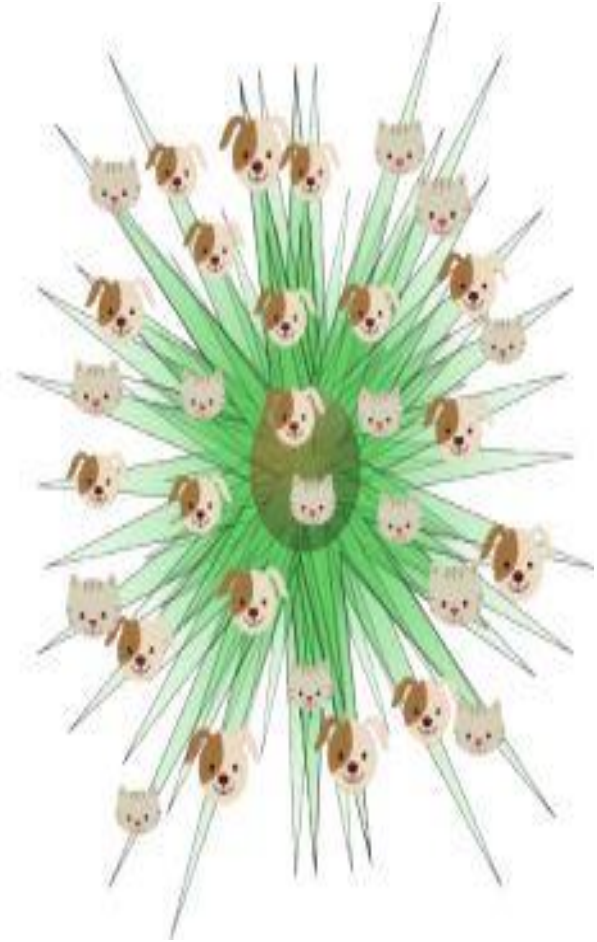
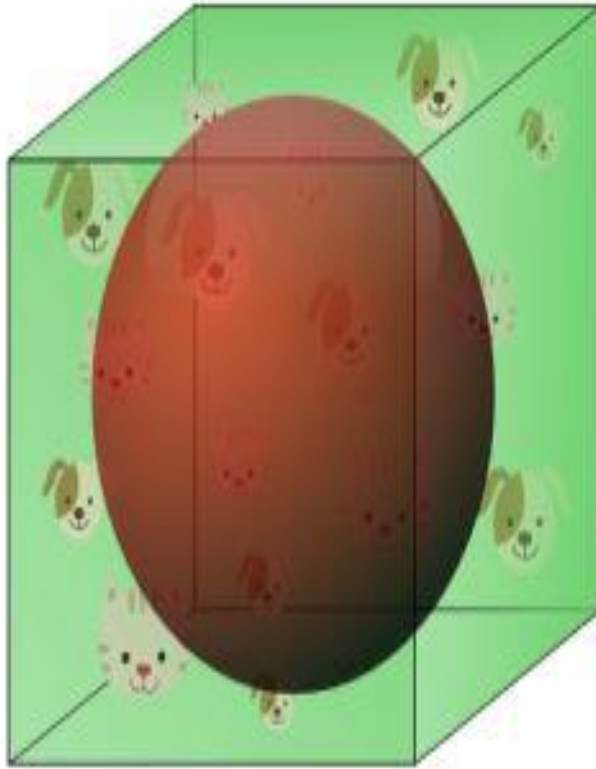
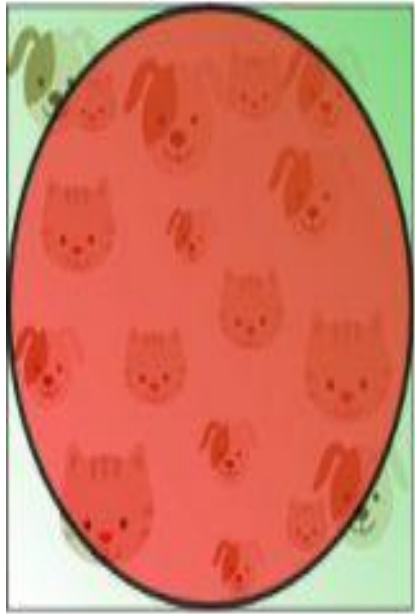
# 特征选取



Feature 1









- 数据特征**选择**：从属性集合中选择那些重要的、与分析任务相关的子集的过程
- 数据特征**提取**：对属性进行重新组合，获得一组反映事物本质的少量新属性的过程
- 有效的数据特征选择：
  - 降维
  - 降低学习任务的难度
  - 提升模型的效率





- 主成分分析（Principal components analysis，以下简称PCA）是一种通过降维技术把多个变量化为少数几个主成分的统计方法，是最重要的降维方法之一。
- 通过正交变换将一组可能存在相关性的变量转换为一组线性不相关的变量，转换后的这组变量叫主成分。PCA的思想是将 $n$ 维特征映射到 $k$ 维上（ $k < n$ ），这 $k$ 维是全新的正交特征。这 $k$ 维特征称为主成分，是重新构造出来的 $k$ 维特征，而不是简单地从 $n$ 维特征中去除其余 $n-k$ 维特征。



## 第二章 知识发现过程与应用结构



随着人工智能的崛起，一个叫ImageNET视觉识别的挑战赛在近几年里备受瞩目。

这个挑战赛要求参赛团队使用 ImageNet——全球最大的图像识别数据库，测试他们系统的运行情况。



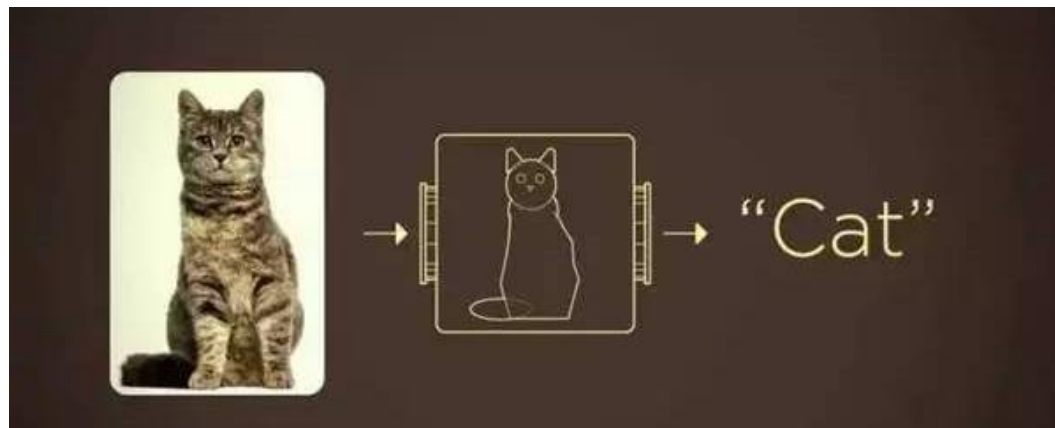
**ImageNet Large Scale Visual Recognition Challenges**



## 第二章 知识发现过程与应用结构



李飞飞，现为美国斯坦福大学教授、斯坦福大学人工智能实验室与视觉实验室负责人、谷歌云人工智能和机器学习首席科学家，斯坦福以人为本人工智能研究院共同院长。





## 第二章 知识发现过程与应用结构



随着人工智能的崛起，一个叫ImageNET视觉识别的挑战赛在近几年里备受瞩目。

这个挑战赛要求参赛团队使用 ImageNet——全球最大的图像识别数据库，测试他们系统的运行情况。



**ImageNet Large Scale Visual Recognition Challenges**





- KDD是一个多步骤的处理过程：
- 1、问题定义、
- 2、数据采集、
- 3、数据预处理（清洗、转换、描述、选择、抽取）
- 4、数据挖掘、
- 5、模式评估



- KDD是一个多步骤的处理过程：
- 1、问题定义、
- 2、数据采集、
- 3、数据预处理（清洗、转换、描述、选择、抽取）
- 4、数据挖掘（十大经典算法、机器学习、NN…）
- 5、模式评估



实施这样的项目不仅需要充足的资金，而且需要有良好的技术和人员储备。在整个的知识发现过程中，需要有不同专长的技术人员支持。

1、业务分析人员：要求精通业务，能够解释业务对象，并根据各业务对象确定出用于数据定义和挖掘算法的业务需求。

2、数据分析人员：精通数据分析技术，并对统计学有较熟练的掌握，有能力把业务需求转化为知识发现的各步操作，并为每步操作选择合适的模型或工具。

3、数据管理人员：精通数据管理技术，并负责从数据库或数据仓库中收集数据。





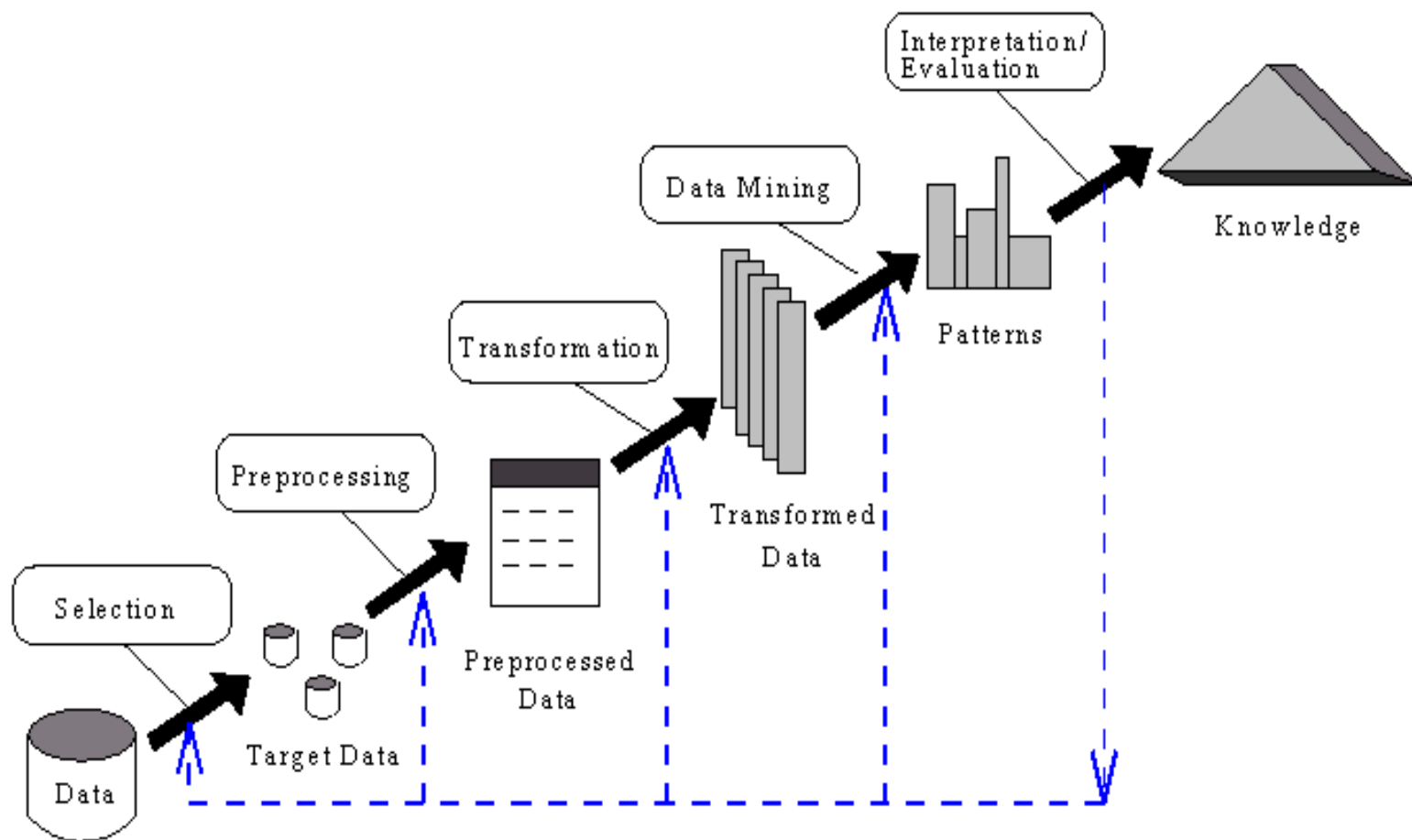
# 知识评估阶段的功能

- 数据挖掘阶段发现出来的模式，经过评估，可能存在冗余或无关的模式，这时需要将其剔除；也有可能模式不满足用户要求，这时则需要整个发现过程**回退到前续阶段**，如重新选取数据、采用新的数据变换方法、设定新的参数值，甚至换一种算法等等。
- KDD由于最终是面向人类用户的，因此可能要对发现的模式进行**可视化**，或者把结果转换为用户易懂的另一种表示。所以知识评估阶段是KDD一个重要的必不可少的阶段，它不仅担负着将KDD系统发现的知识以用户能了解的方式呈现，而且根据需要进行知识评价，如果和用户的挖掘目标不一致就需要返回前面相应的步骤进行螺旋式处理以最终获得可用的知识。



# 阶梯处理过程模型

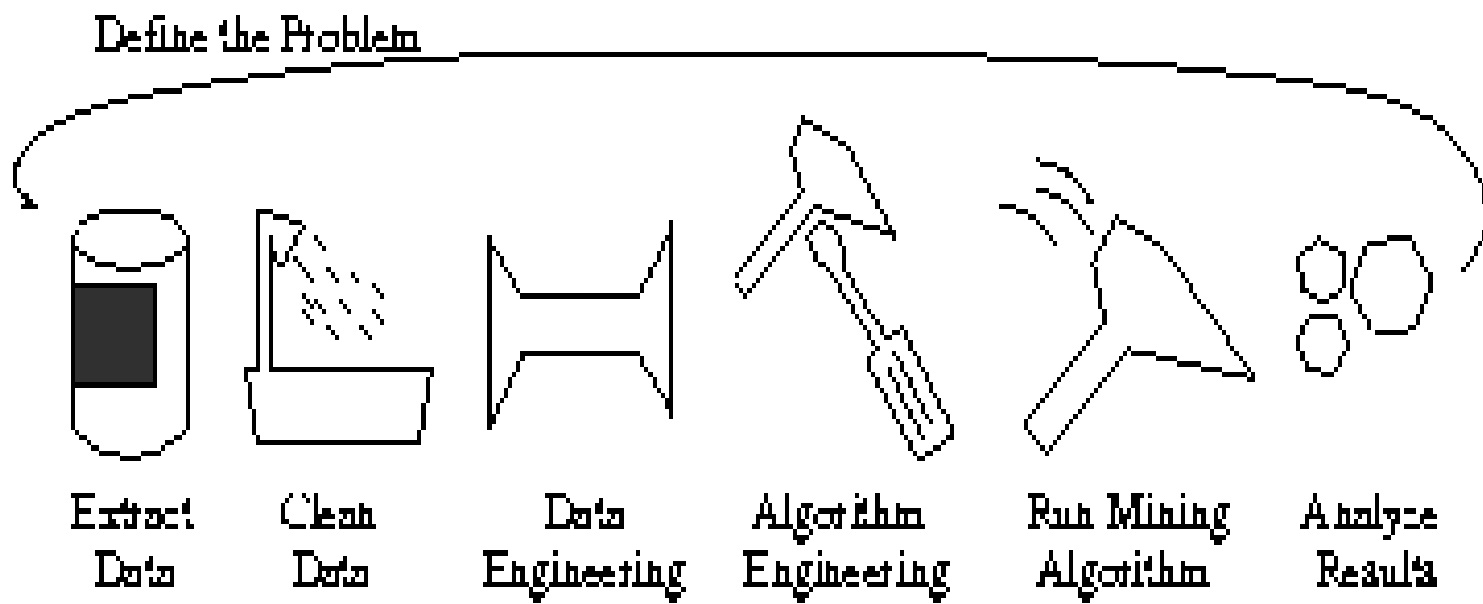
## ■ 多阶段流水处理模型：





# 螺旋处理过程模型

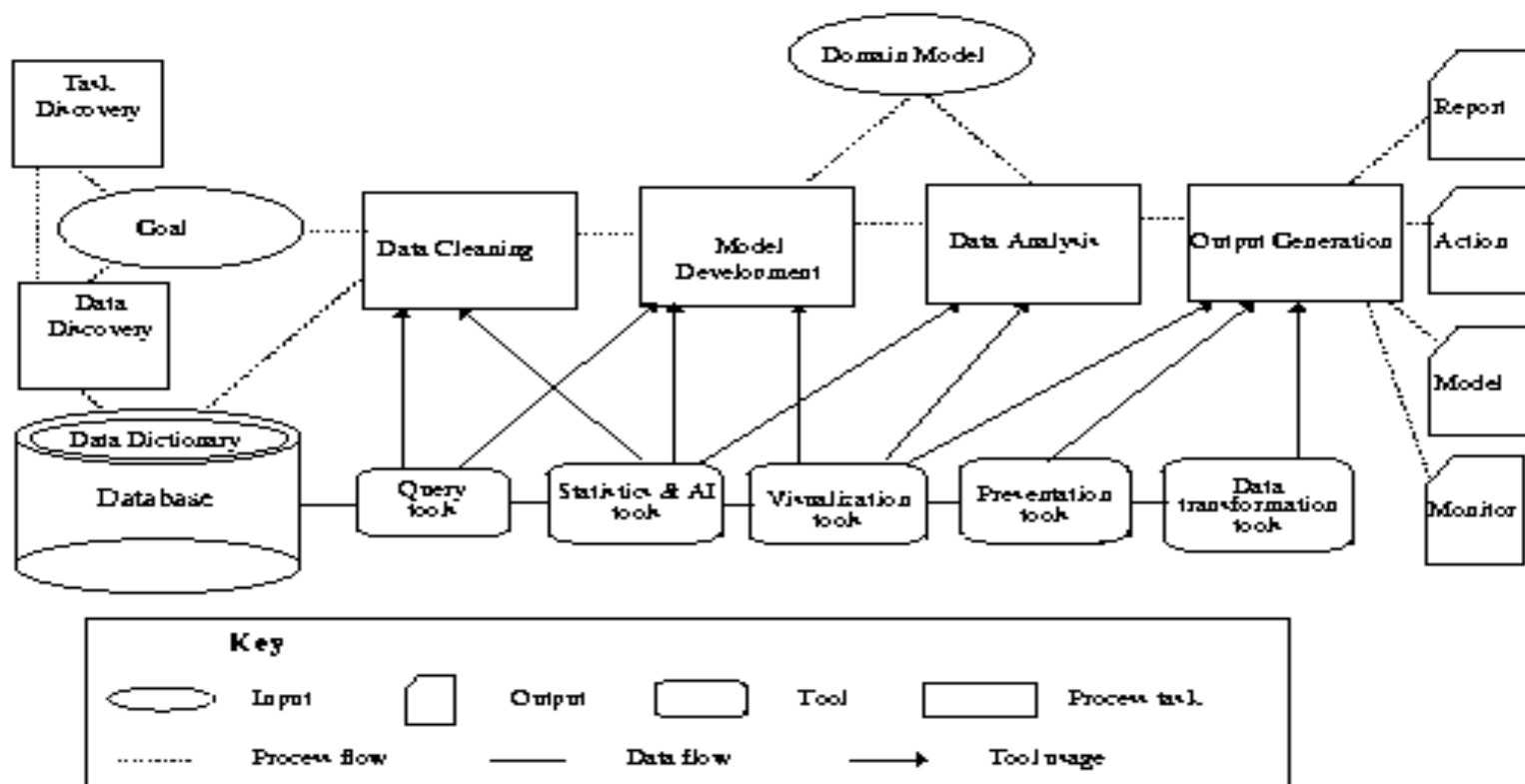
- 它强调领域专家参与的重要性，并以问题的定义为中心循环评测挖掘的结果。当结果不令人满意时，就需要重新定义问题，开始新的处理循环。每次循环都使问题更清晰，结果更准确，因此是一个螺旋式上升过程。





# 以用户为中心的处理模型

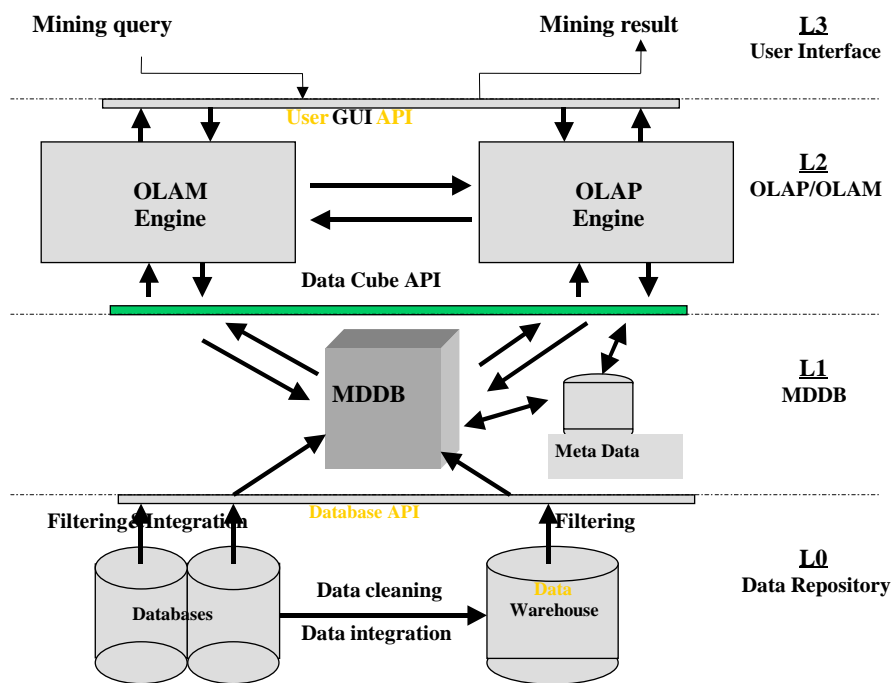
- Brachman和Anand从用户的角度对KDD处理过程进行了分析。他们认为数据库中的知识发现应该更着重于对用户进行知识发现的整个过程的支持，而不是仅仅限于在数据挖掘的一个阶段上。该模型强调对**用户与数据库交互**的支持。





# 联机KDD模型

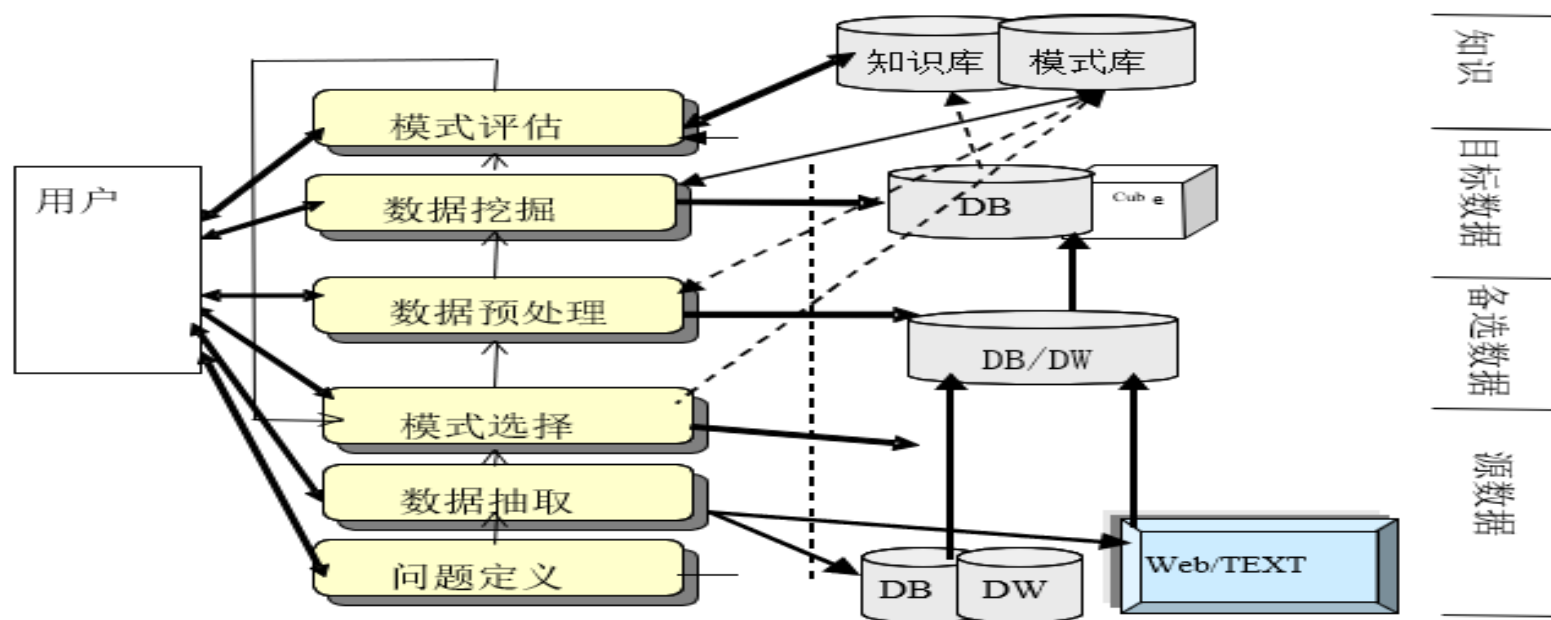
- 实现联机交互式KDD需要可视化技术支撑。这种可视化需要从数据挖掘过程可视化、数据可视化、模型可视化和算法可视化等方面来理解。
- OLAM (On Line Analytical Mining: 联机分析挖掘) 的概念是OLAP的发展。





# 支持多数据源多知识模式的KDD处理模型

- 数据与方法相对独立。数据不是针对某一特定知识模式，而是针对某一类问题来抽取。经过预处理后，这些数据对于某些挖掘算法来说可能存在属性冗余、与目标无关等问题，因此在后面的阶段再进行相关的数据清洗和选择工作，这样使得解决同一类问题的**不同算法可以在统一的KDD平台上完成。**





# 第二章 知识发现过程与应用结构

## 内容提要

- 知识发现的基本过程
- 数据库中的知识发现处理过程模型
- 知识发现软件或工具的发展
- 知识发现项目的过程化管理
- 数据挖掘语言介绍





# 知识发现软件的发展

- 虽然市场上已经有许多所谓的知识发现系统或工具，但是，这些工具只能用来辅助技术人员进行设计和开发，而且知识发现软件本身也正处于发展阶段，仍然存在各种各样需要解决的问题。
- 粗略地说，知识发现软件或工具的发展经历了独立的知识发现软件、横向的知识发现工具集和纵向的知识发现解决方案三个主要阶段，其中后面两种反映了目前知识发现软件的两个主要发展方向。



# 独立的知识发现软件

- 独立的知识发现软件出现在数据挖掘和知识发现技术研究的早期。当研究人员开发出一种新型的数据挖掘算法后，就在此基础上形成软件原型。这些原型系统经过完善被尝试使用。
- 这类软件要求用户必须对具体的数据挖掘技术和算法有相当的了解，还要手工负责大量的数据预处理工作。



# 横向的知识发现工具

- 集成化的知识发现辅助工具集，属于通用辅助工具范畴，可以帮助用户快速完成知识发现的不同阶段处理工作。
- 一些有代表性的原型系统或工具介绍。

名称	研究机构或公司	主要特点
DBMiner[1] 等多模式。	Simon Fraser	以OLAM引擎为核心的联机挖掘原型系统；包含多特征/序列/关联
Quest[75]	IBM Almaden	面向大数据集的多模式（关联规则/分类等）挖掘工具。
IBM Intelligent Miner[76]	IBM	包含多种技术（神经网络/统计分析/聚类等）的辅助挖掘工具集。
Darwin[76]	Thinking Machines	基于神经网络的辅助挖掘工具。
ReMind	Cognitive System	基于实例推理和归纳逻辑的辅助挖掘工具。



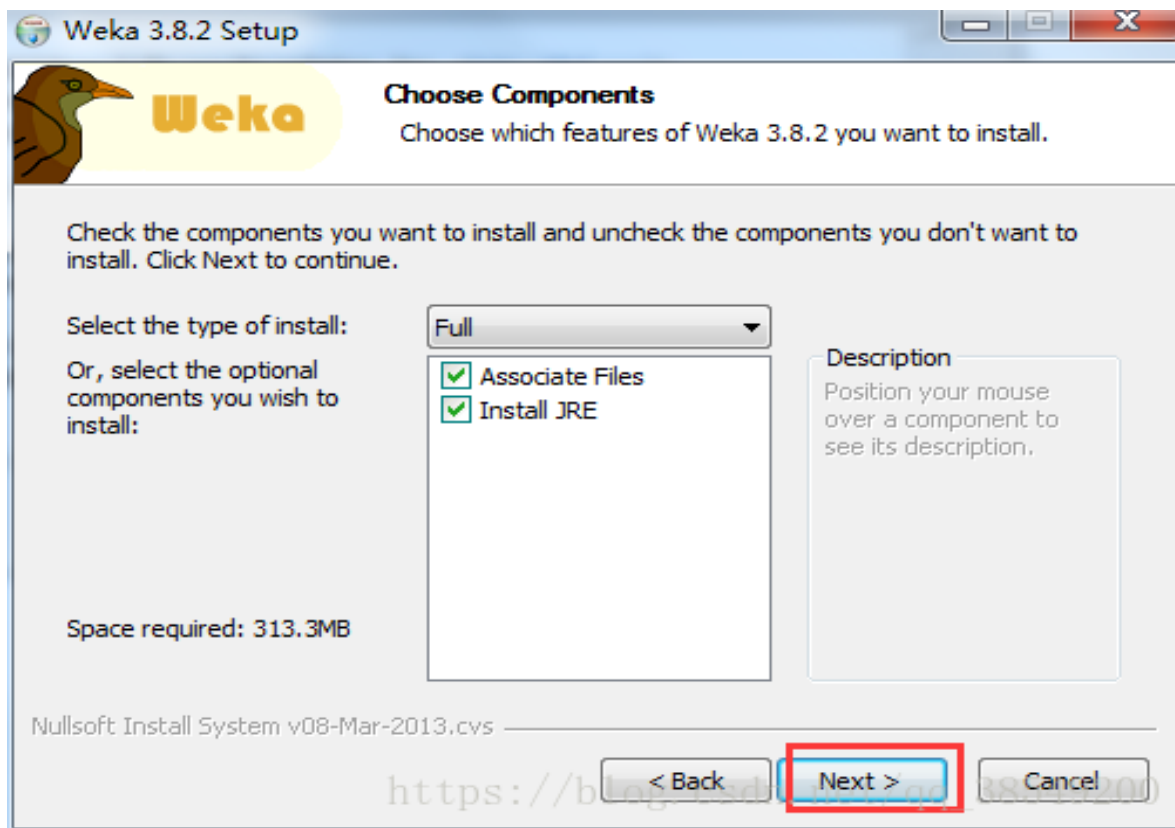
WEKA的全名是怀卡托智能分析环境（Waikato Environment for Knowledge Analysis），weka也是新西兰的一种鸟名





## ■ WEKA的官方地址是

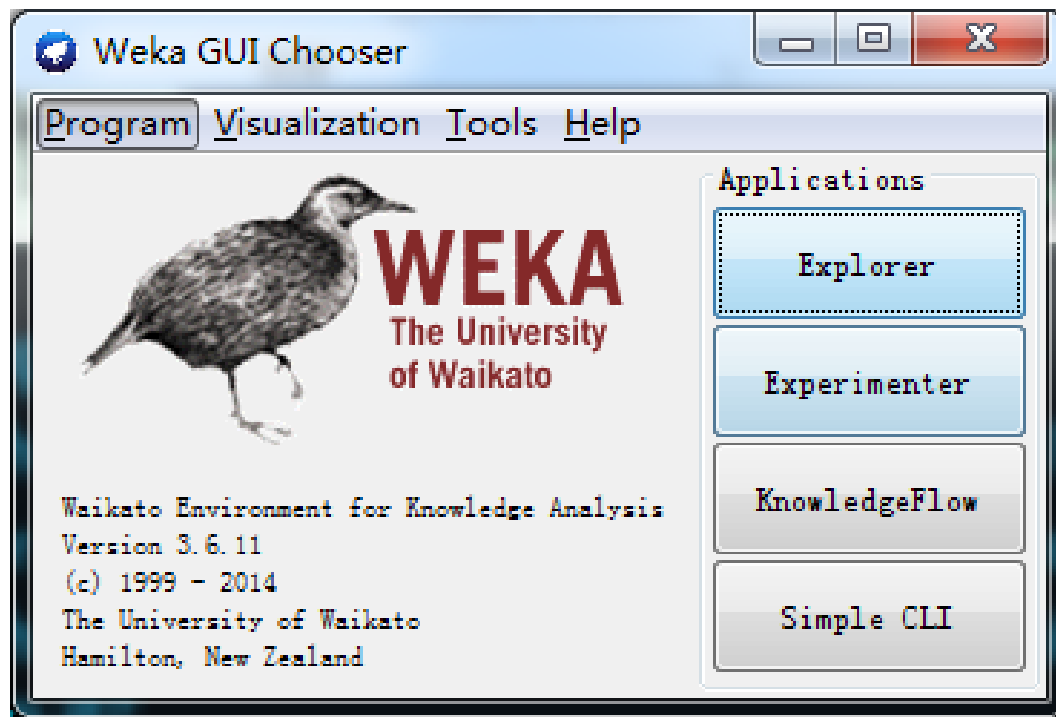
<http://www.cs.waikato.ac.nz/ml/weka/downloading.html>。里面有windows, mas os, linux等平台下的版本。





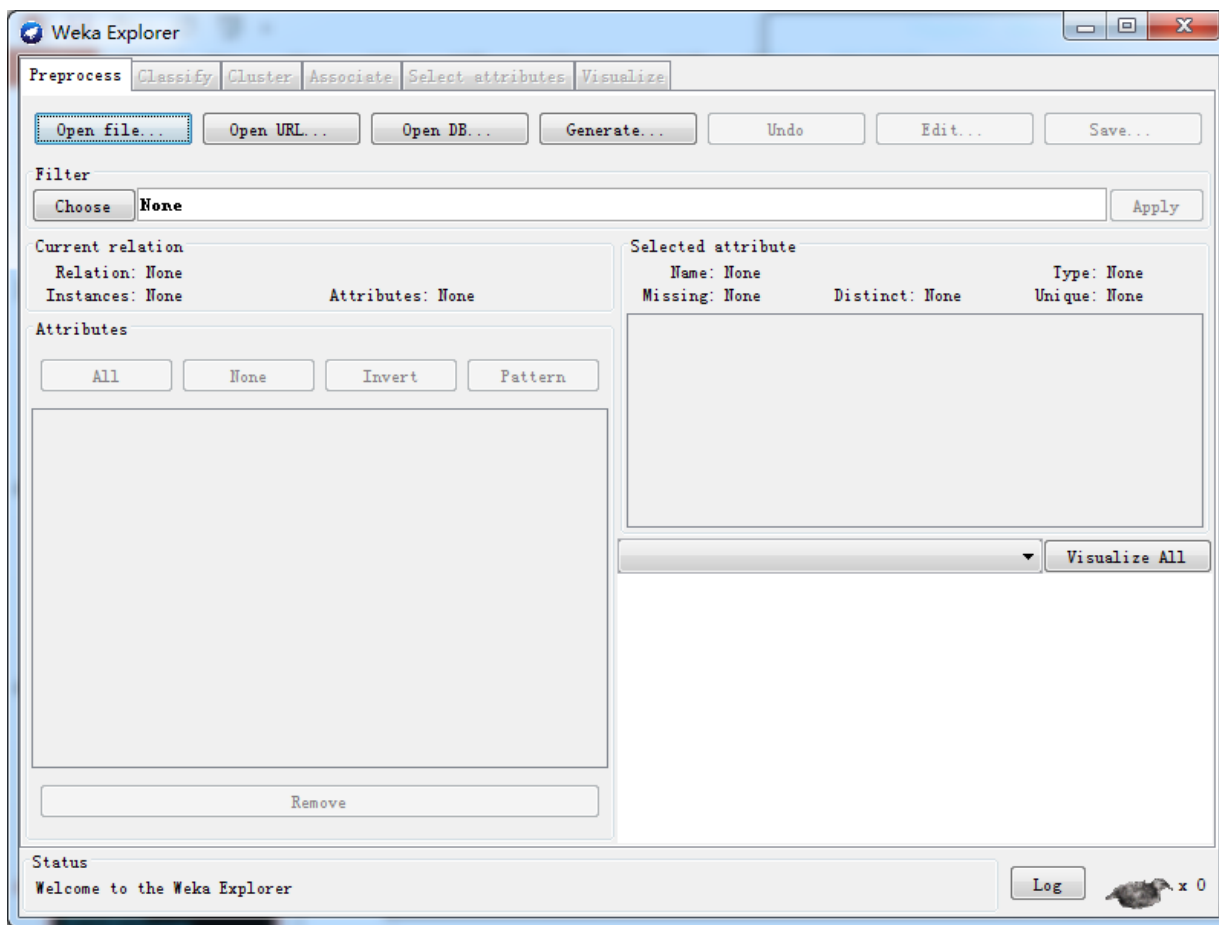


- WEKA作为一个公开的数据挖掘工作平台，集合了大量能承担数据挖掘任务的机器学习算法，包括对数据进行预处理，分类，回归、聚类、关联规则以及在新的交互式界面上的可视化。





- 点开erxplore, 打开数据文件 (\*.arff), 多观察看看各种属性和标签按钮





## ■ 多观察看看各种属性和标签按钮

**Weka Explorer**

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter: Choose **None** Apply

Current relation:  
Relation: weather.symbolic  
Instances: 14 Attributes: 5

Attributes:  
All None Invert **Pattern**

No.	Name
1	<input checked="" type="checkbox"/> outlook
2	<input type="checkbox"/> temperature
3	<input type="checkbox"/> humidity
4	<input type="checkbox"/> windy
5	<input type="checkbox"/> play

Remove

Selected attribute:  
Name: outlook  
Missing: 0 (0%) Distinct: 3 Type: Nominal Unique: 0 (0%)

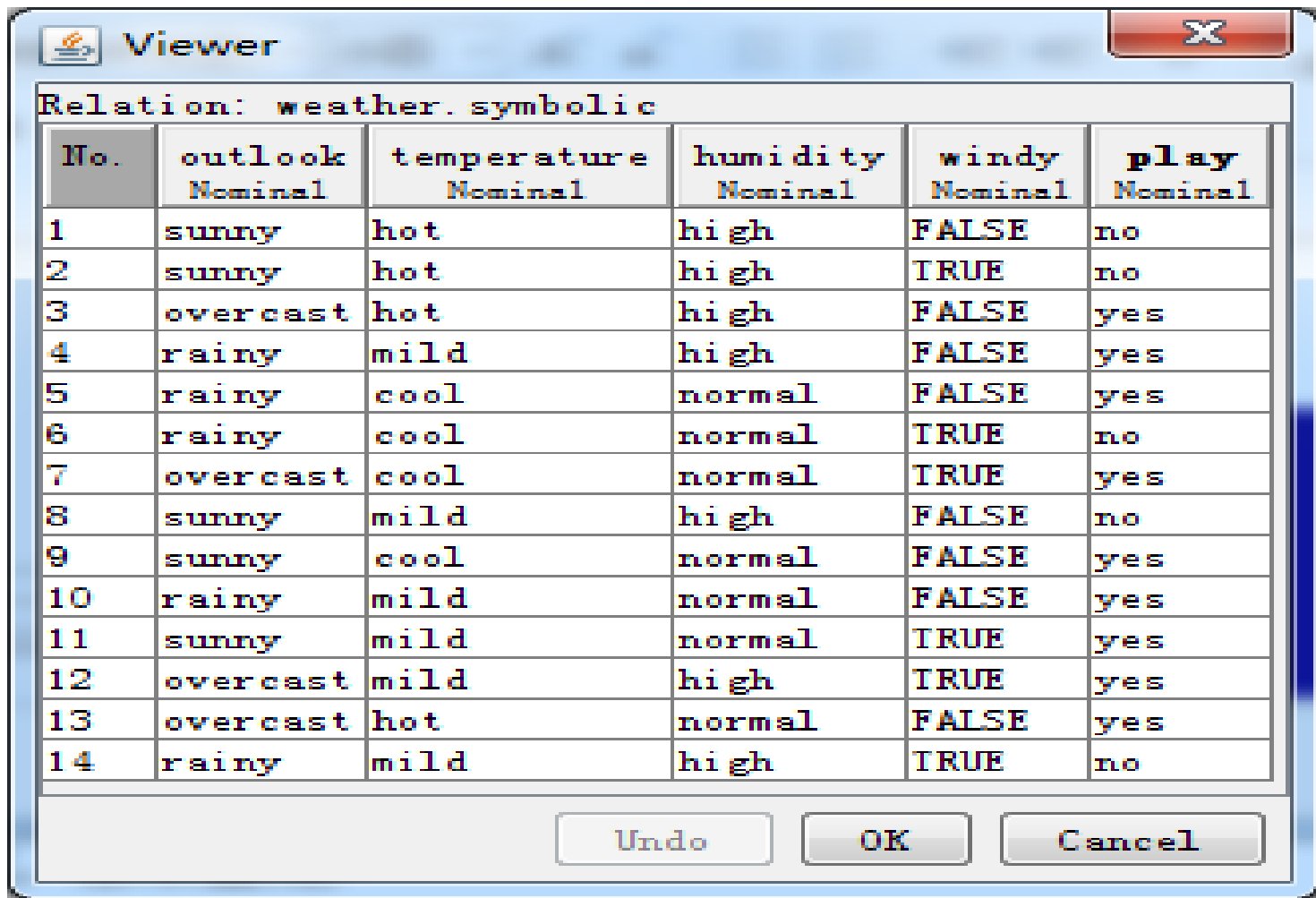
No.	Label	Count
1	sunny	5
2	overcast	4
3	rainy	5

Class: play (Nom) Visualize All

Status: OK Log x 0



- 点击edit按钮，查看数据的表格形式，非常直观



Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	play Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Undo OK Cancel



- 选择一个算法训练这组数据（比如：决策树，从tree里选择j48，再选交叉验证方法，再点start，可以从右边看到结果）

Weka Explorer

Preprocess | **Classify** | Cluster | Associate | Select attributes | Visualize

Classifier: Choose J48 -C 0.25 -M 2

Test options

- ☐ Use training set
- ☐ Supplied test set Set...
- ☒ Cross-validation Folds 10
- ☐ Percentage split % 66

More options...

(Nom) play

Start Stop

Result list (right-click for options)

- 17:09:58 - trees.J48
- 17:10:55 - trees.J48
- 17:10:59 - trees.J48

Classifier output

Correctly Classified Instances 7 50 %  
Incorrectly Classified Instances 7 50 %  
Kappa statistic -0.0426  
Mean absolute error 0.4167  
Root mean squared error 0.5984  
Relative absolute error 87.5 %  
Root relative squared error 121.2987 %  
Total Number of Instances 14

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC AUC
Weighted Avg.	0.556	0.6	0.625	0.556	0.588	0.4

=== Confusion Matrix ===

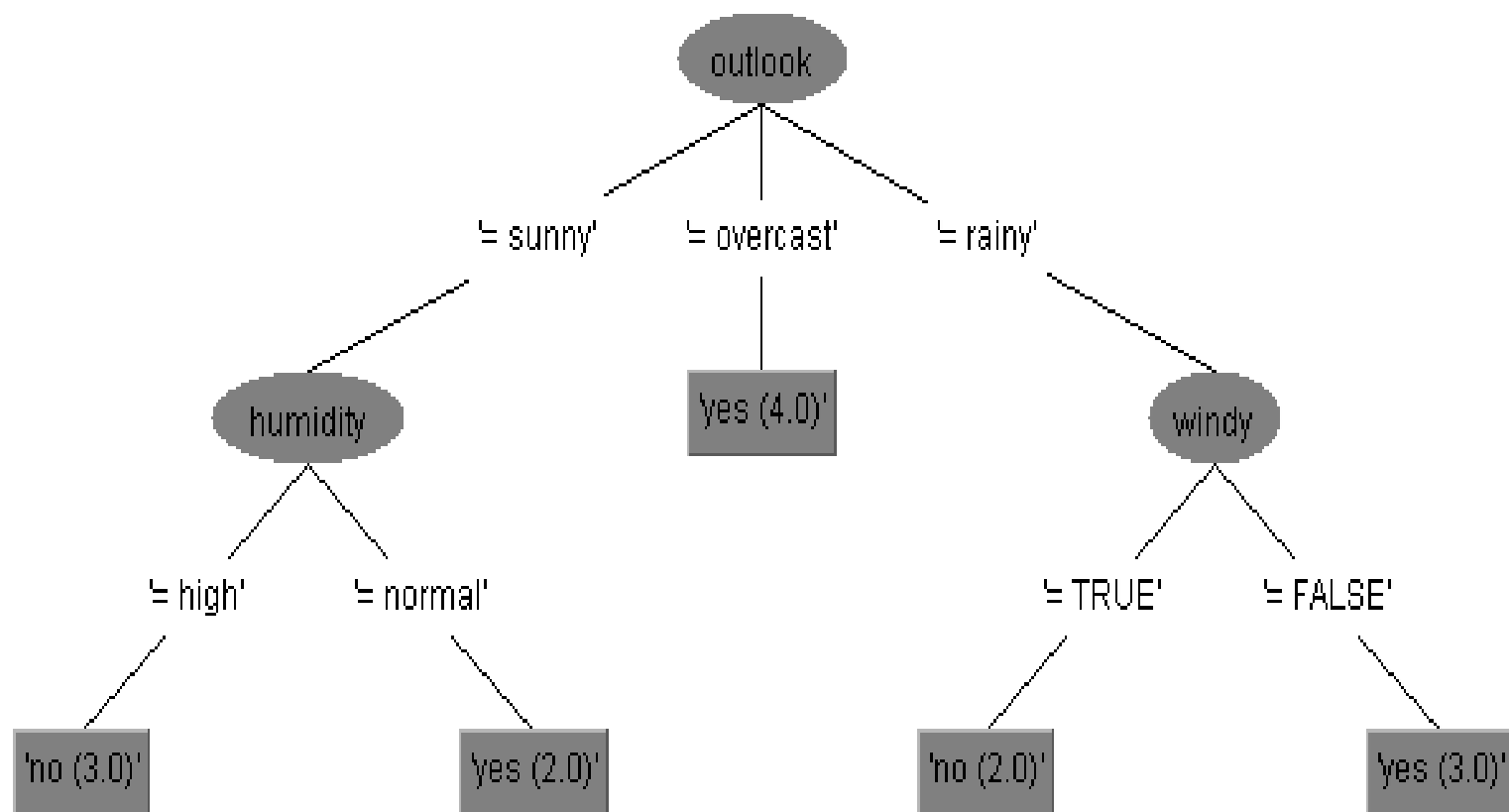
a b <-- classified as  
5 4 | a = yes  
3 2 | b = no

Status OK

Log x 0



- 对着上图选中的那次实验，鼠标右键，然后选择 visualize tree





## Classifier output

=== Classifier model (full training set) ===

J48 pruned tree

-----

outlook = sunny

| humidity = high: no (3.0)

| humidity = normal: yes (2.0)

outlook = overcast: yes (4.0)

outlook = rainy

| windy = TRUE: no (2.0)

| windy = FALSE: yes (3.0)

Number of Leaves : 5

Size of the tree : 8

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	7	50	%
Incorrectly Classified Instances	7	50	%
Kappa statistic	-0.0426		
Mean absolute error	0.4167		
Root mean squared error	0.5984		
Relative absolute error	87.5	%	
Root relative squared error	121.2987	%	
Total Number of Instances	14		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.556	0.600	0.625	0.556	0.588	-0.043	0.633	0.758	yes
	0.400	0.444	0.333	0.400	0.364	-0.043	0.633	0.457	no
Weighted Avg.	0.500	0.544	0.521	0.500	0.508	-0.043	0.633	0.650	

=== Confusion Matrix ===

a b <-- classified as

5 4 | a = yes

3 2 | b = no



=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.556	0.400	0.714	0.556	0.625	0.149	0.656	0.743	yes
	0.600	0.444	0.429	0.600	0.500	0.149	0.656	0.513	no
Weighted Avg.	0.571	0.416	0.612	0.571	0.580	0.149	0.656	0.661	

=== Confusion Matrix ===

a b <-- classified as

5 4 | a = yes

2 3 | b = no

# 第二章 知识发现过程与应用结构

## 内容提要

- 知识发现的基本过程
- 数据库中的知识发现处理过程模型
- 知识发现软件或工具的发展
- 知识发现项目的过程化管理
- 数据挖掘语言介绍





# 知识发现项目的过程化管理问题

- 开发一个数据挖掘和知识发现项目需要各方面协同合作而且极易出现问题，因此它的质量管理问题的讨论是重要而困难的。
- 近几年，有一些针对数据挖掘和知识发现项目的过程化管理所开展的工作，其中一个典型的模型三被称作强度挖掘（Intension Mining）的I-MIN过程模型。
- I-MIN过程模型把KDD过程分成IM1、IM2、...、IM6等步骤处理，在每个步骤里，集中讨论几个问题，并按一定的质量标准来控制项目的实施。



# IM1的任务与目的

- 它是KDD项目的计划阶段，需要确定企业的挖掘目标，选择知识发现模式，编译知识发现模式得到的元数据。其目的是将企业的挖掘目标嵌入到对应的知识模式中。
- 对数据挖掘研究人员来说，往往把主要精力用在改进现有算法和探索新算法上。但是在真正调用挖掘算法之前，必须对企业的决策机制和流程进行充分调研，理解企业急需解决的问题。需要准确地确定挖掘目标和可交付系统的指标等。



# IM2的任务与目标

- 它是KDD的预处理阶段，可以用IM2a、IM2b、IM2c等分别对应于数据清洗、数据选择和数据转换等阶段。其目的是生成高质量的目标数据。
- 知识发现项目的数据预处理是一个费时费力的工作。事实上，数据挖掘的成功与否，数据预处理起到了至关重要的作用。只有好的预处理，才能避免Garbage in, Garbage out (GIGO: 垃圾进垃圾出) 的现象发生。



# IM3的任务与目标

- 它是KDD的挖掘准备阶段，数据挖掘工程师进行挖掘实验，反复测试和验证模型的有效性。其目的是通过实验和训练得到浓缩知识(Knowledge Concentrate)，为最终用户提供可使用的模型。





# IM4的任务与目标

- 它是KDD的数据挖掘阶段，用户通过指定数据挖掘算法得到对应的知识。



# IM5的任务与目标

- 它是KDD的知识表示阶段，按指定要求形成规格化的知识。



# IM6的任务与目标

- 它是KDD的知识解释与使用阶段，其目的是根据用户要求直观地输出知识或集成到企业的知识库中。

# 第二章 知识发现过程与应用结构

## 内容提要

- 知识发现的基本过程
- 数据库中的知识发现处理过程模型
- 知识发现软件或工具的发展
- 知识发现项目的过程化管理
- 数据挖掘语言介绍





- 设计理想的数据挖掘语言是一个巨大的挑战。这是因为数据挖掘覆盖的任务宽、包含知识形式广（如数据特征化、关联规则、数据分类、聚集等等）。每个任务都有不同的需求，每种知识表示形式都有不同内涵。一个有效的数据挖掘语言设计需要对各种不同的数据挖掘任务的能力、约束以及运行机制有深入地理解。
- 众所周知，关系查询语言的标准化，发生在关系型数据库开发的早期阶段。经过不懈的努力，以SQL为代表的关系型数据库查询语言的标准化被成功解决。同样，一个好的数据挖掘语言可以有助于数据挖掘系统平台的标准化进程，甚至可以象HTML推动Internet的发展一样，推动数据挖掘行业的开发和发展。
- 数据挖掘语言的发展大致经历了两个阶段：第一个阶段是研究单位和公司自行研究和开发阶段；第二阶段是研究单位和公司组成联盟，研制和开发数据挖掘语言标准化的阶段。



- 根据功能和侧重点不同，数据挖掘语言可以分为三种类型：
  - 数据挖掘查询语言：希望以一种像SQL这样的数据库查询语言完成数据挖掘的任务。
  - 数据挖掘建模语言：对数据挖掘模型进行描述和定义的语言，设计一种标准的数据挖掘建模语言，使得数据挖掘系统在模型定义和描述方面有标准可以遵循。
  - 通用数据挖掘语言：通用数据挖掘语言合并了上述两种语言的特点，既具有定义模型的功能，又能作为查询语言与数据挖掘系统通信，进行交互式挖掘。通用数据挖掘语言的标准化是目前解决数据挖掘行业出现问题的颇具吸引力的研究方向。



- J. W. Han等开发的数据挖掘系统DBMiner中数据挖掘查询语言DMQL (Data Mining Query Language) 是这类挖掘语言的典型代表。数据挖掘查询语言DMQL由数据挖掘原语组成，数据挖掘原语用来定义一个数据挖掘任务。用户使用数据挖掘原语与数据挖掘系统通信，使得知识发现更有效。
- 这些原语有以下几个种类：
  - 数据库部分以及用户感兴趣的数据集（包括感兴趣的数据库属性或数据仓库的维度）；
  - 挖掘知识的种类；在指导挖掘过程中有用的背景知识；
  - 模式估值的兴趣度测量；挖掘出的知识如何可视化表示。
- 数据挖掘查询的基本单位是数据挖掘任务，通过数据挖掘查询语言，数据挖掘任务可以通过查询的形式输入到数据挖掘系统中。一个数据挖掘查询由五种基本的数据挖掘原语定义。





- 数据挖掘建模语言是对数据挖掘模型进行描述和定义的语言。
- 预言模型标记语言” (Predictive Model Markup Language, PMML) 被一个称作数据挖掘协会 (The Data Mining Group, DMG) 的组织开发。PMML是一种基于XML的语言, 用来定义预言模型。PMML允许应用程序和联机分析处理 (OLAP) 工具能从数据挖掘系统获得模型, 而不用独自开发数据挖掘模块。
- PMML的模型定义由以下几部分组成:
  - 头文件 (Header);
  - 数据模式 (Data Schema);
  - 数据挖掘模式 (Data Mining Schema);
  - 预言模型模式 (Predictive Model Schema);
  - 预言模型定义 (Definitions for Predictive Models);
  - 全体模型定义 (Definitions for Ensembles of Models);
  - 选择和联合模型和全体模型的规则 (Rules for Selecting and Combining Models and Ensembles of Models);
  - 异常处理的规则 (Rules for Exception Handling)



- 通用数据挖掘语言合并了上述两种语言的特点，既具有定义模型的功能，又能作为查询语言与数据挖掘系统通信，进行交互式挖掘。通用数据挖掘语言的标准化是目前解决数据挖掘行业出现问题的颇具吸引力的研究方向。
- 2000年3月，微软公司推出了一个数据挖掘语言，称作OLE DB for Data Mining (DM)，是通用数据挖掘语言中最具代表性的尝试。微软此举的目的是为数据挖掘提供行业标准。只要符合这个标准，都能容易地嵌入应用程序中。
- OLE DB for DM支持多种流行的数据挖掘算法。使用OLE DB for DM，数据挖掘应用能够通过OLE DB生产者接进任何表格式的数据源。



# DMQL挖掘查询语言介绍

## ■ DMQL语言的顶层语法

{DMQL} ::= <DMQL\_Statement> ; {<DMQL\_Statement>}

<DMQL\_Statement> ::= <Data\_Mining\_Statement>

          | <Concept\_Hierarchy\_Definition\_Statement>

          | <Visualization\_and\_Presentation>

## ■ 数据挖掘声明 (Data\_Mining\_Statement) 语句相关项说明

<Data\_Mining\_Statement> ::= use database <database\_name>

          | use data warehouse <data\_warehouse\_name>

          {use hierarchy <hierarch\_name> for  
          <attribute\_or\_dimension>}

          from <relation(s)/cube(s)> [where <condition>]

          in relevance to <attribute\_or\_dimension\_list>

          [order by <order\_list>]

          [group by <grouping\_list>]

          [having <condition>]

## ■ 例子：

*use database* AllElectronics\_db

*in relevance to* I.name, I.price, C.income, C.age

*from* customer C, item I, purchases P, items\_sold S

*where* I.item\_ID=S.item\_ID and S.trans\_ID=P.trans\_ID and P.cust\_ID=C.cust\_ID and  
          C.country="Canada"

*group by* P.date;



# DMQL挖掘查询语言介绍(续)

## ■ 挖掘知识指定 (Mine\_Knowledge\_Specification) 语句相关项说明

$\langle \text{Mine\_Knowledge\_Specification} \rangle ::= \langle \text{Mine\_Char} \rangle \mid \langle \text{Mine\_Discr} \rangle \mid \langle \text{Mine\_Assoc} \rangle \mid \langle \text{Mine\_Class} \rangle$

$\langle \text{Mine\_Char} \rangle ::= \text{mine characteristics [as } \langle \text{pattern\_name} \rangle \text{] analyze } \langle \text{measure(s)} \rangle$

$\langle \text{Mine\_Discr} \rangle ::= \text{mine comparison [as } \langle \text{pattern\_name} \rangle \text{]}$   
for  $\langle \text{target\_class} \rangle$  where  $\langle \text{target\_condition} \rangle$   
{versus  $\langle \text{contrast\_class\_i} \rangle$  where  
 $\langle \text{contrast\_condition\_i} \rangle$ }  
analyze  $\langle \text{measure(s)} \rangle$

$\langle \text{Mine\_Assoc} \rangle ::= \text{mine associations [as } \langle \text{pattern\_name} \rangle \text{]}$   
[matching  $\langle \text{metapattern} \rangle$ ]

$\langle \text{Mine\_Class} \rangle ::= \text{mine classification [as } \langle \text{pattern\_name} \rangle \text{]}$   
analyze  $\langle \text{classifying\_attribute\_or\_dimension} \rangle$



# DMQL挖掘查询语言介绍(续)

## ■ 概念分层声明 (Concept\_Hierarchy\_Definition\_Statement) 相关项说明

```
<Concept_Hierarchy_Definition_Statement> ::= define hierarchy  
    <hierarchy_name> [for <attribute_or_dimension>]  
    on <relation_or_cube_or_hierarchy>  
    as <hierarchy_description>  
    [where <condition>]
```

## ■ 例子:

```
define hierarchy age_hierarchy for age on customer as  
    level1: {young, middle_aged, senior} < level0: all  
    level2: {20, ..., 39} < level1: young  
    level2: {40, ..., 59} < level1: middle_aged  
    level2: {60, ..., 89} < level1: senior;  
define hierarchy profit_margin_hierarchy on item as  
    level1: low_profit_margin < level_0: all  
    if (price - cost) < $50  
    level1: medium-profit_margin < level_0: all  
    if ((price - cost) > $50) and ((price - cost) <= $250))  
    level1: high_profit_margin < level_0: all  
    if (price - cost) > $250;
```



# DMQL挖掘查询语言介绍(续)

## ■ 模式表示和可视化说明的语法

$\langle \text{Visualization\_and\_Presentation} \rangle ::= \text{display as}$

$\langle \text{result\_form} \rangle \mid \{ \langle \text{Multilevel\_Manipulation} \rangle \};$

$\langle \text{Multilevel\_Manipulation} \rangle ::= \text{roll up on } \langle \text{attribute\_or\_dimension} \rangle$

$\mid \text{drill down on } \langle \text{attribute\_or\_dimension} \rangle$

$\mid \text{add } \langle \text{attribute\_or\_dimension} \rangle$

$\mid \text{drop } \langle \text{attribute\_or\_dimension} \rangle;$

其中 $\langle \text{result\_form} \rangle$ 可以是规则、表、交叉表、饼图或条图、判定树、立方体、曲线或曲面等

# 第二章 知识发现过程与应用结构

## 内容提要

- 知识发现的基本过程
- 数据库中的知识发现处理过程模型
- 知识发现软件或工具的发展
- 知识发现项目的过程化管理
- 数据挖掘语言介绍











# 其他常用的数据采集工具



- 不同的数据源使用不同的采集技术和工具

- 1) 日志文件

日志文件常用的采集工具有Flume、Logstash、FileBeat等

- 2) 数据库

数据库数据同步常用的工具有Sqoop和Kettle。

- 3) 网页和APP

网页和APP数据的采集技术一般采用埋点实现。开源的网页埋点工具有Piwik，只需在页面中嵌入一段js代码即可实现数据的采集和传输，后台支持插件开发，对于采集字段做额外处理，自带可视化展现工具，数据从采集到展示的时效性很高。



- KDD是一个多步骤的处理过程：
  - 1、问题定义
  - 2、数据采集
  - 3、数据预处理
  - 4、数据挖掘
  - 5、模式评估