



- 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，一般分为
 - 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
- 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、



- **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

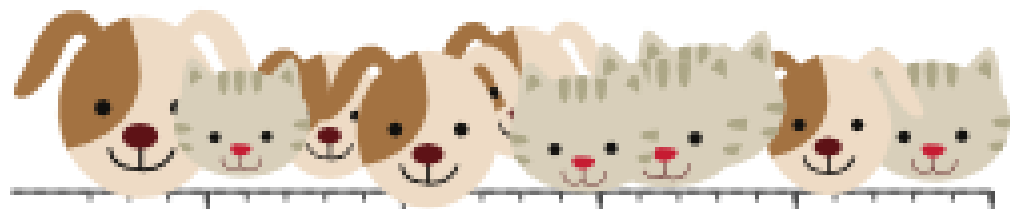


- 3) **规格化处理**: 将有关属性数据按比例投射到特定的小范围之中。例如, 将工资收入属性值映射到 0 到 1 范围内。
- 4) **属性构造处理**: 根据已有属性集构造新的属性, 以帮助数据处理过程。
- 三种规格化方法:
 - 1. **最大最小规格化方法**:
$$\frac{(\text{待转换属性值} - \text{属性最小值})}{(\text{属性最大值} - \text{属性最小值})} * (\text{映射区间最大值} - \text{映射区间最小值}) + \text{映射区间最小值}$$
 - 2. **零均值规格化方法**:
$$\frac{(\text{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

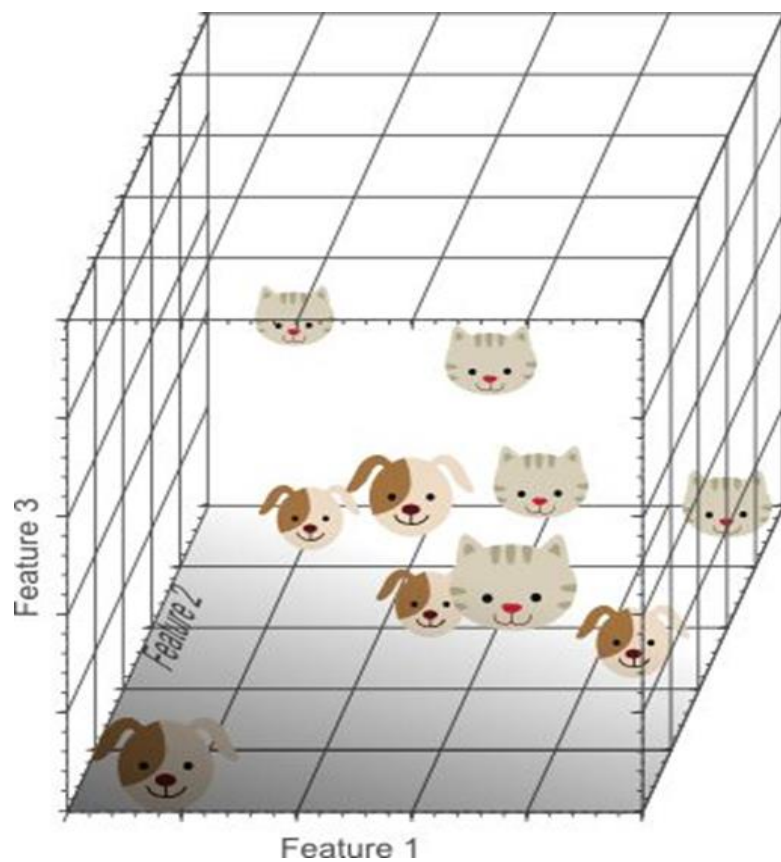
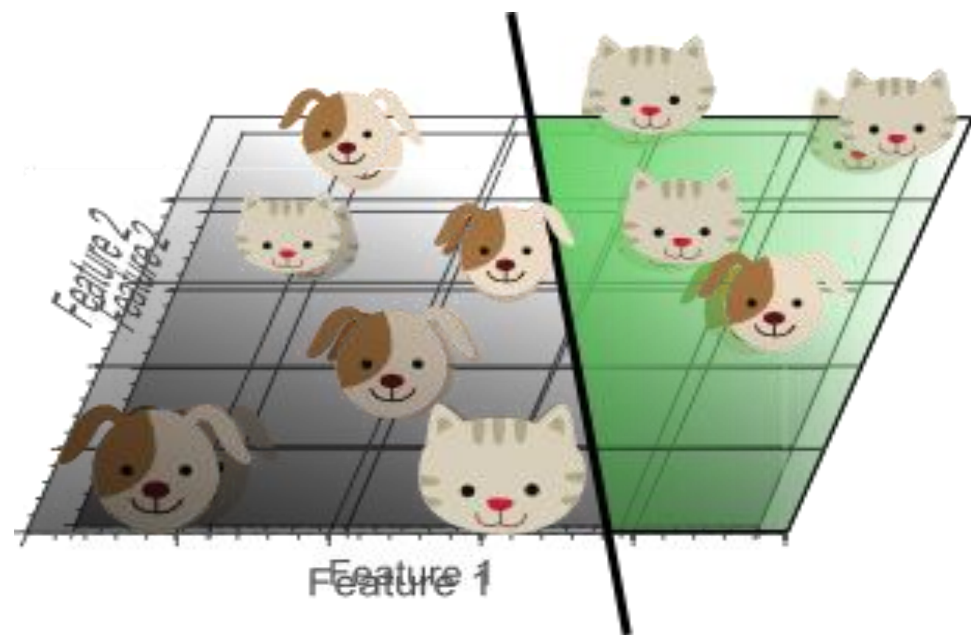


- KDD是一个多步骤的处理过程，一般分为
- 1、问题定义、
- 2、数据采集、
- 3、数据预处理（清洗、转换、描述、**选择、抽取**）
- 4、数据挖掘、
- 5、模式评估、

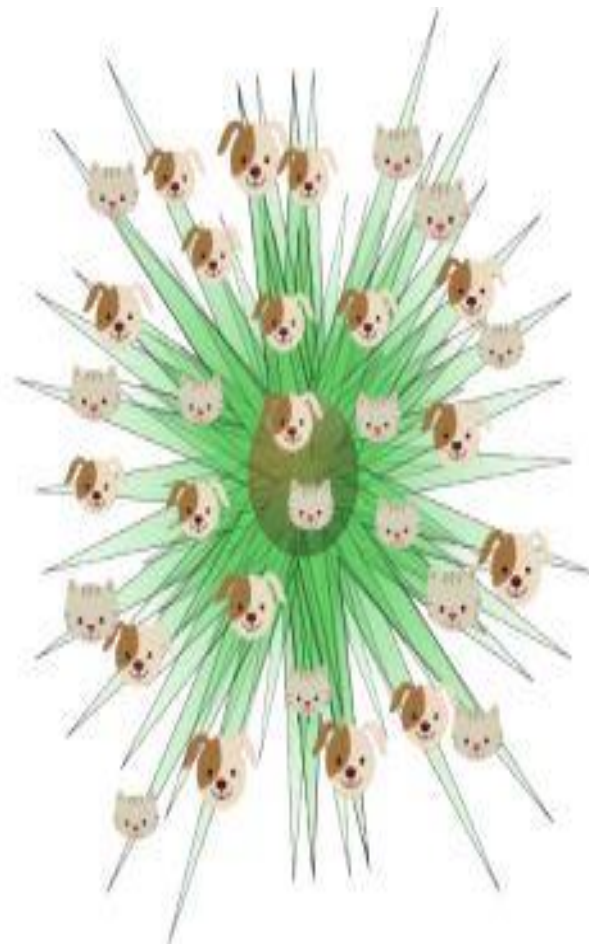
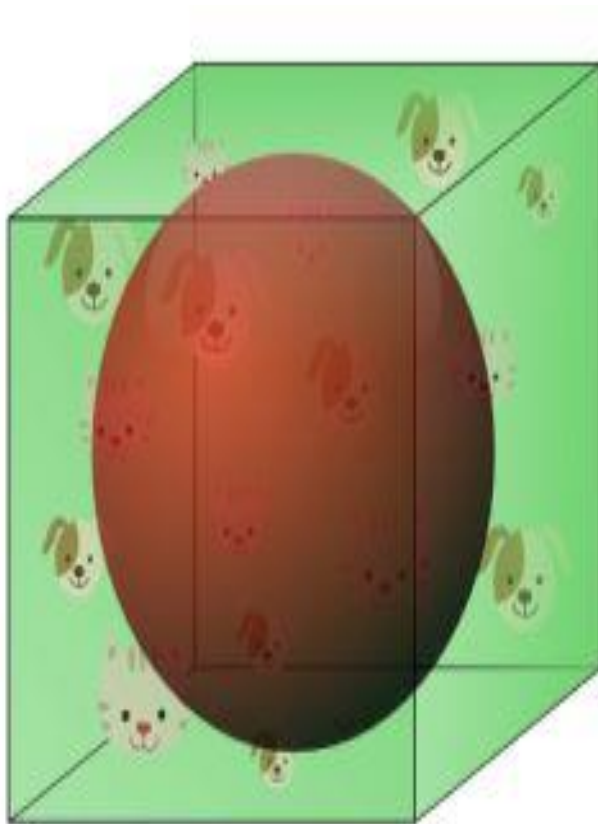
特征选取



Feature 1



为什么要做特征选择





根据性价比挑选房源：

	房价(百万元)	面积(百平米)
<i>a</i>	10	10
<i>b</i>	2	2
<i>c</i>	1	1
<i>d</i>	7	7
<i>e</i>	3	3

可以看出两者**完全正相关**，有一列其实是多余的



- 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

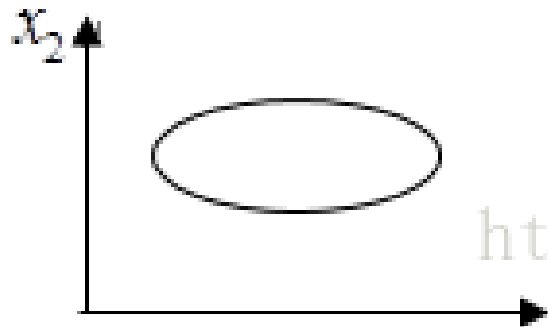


图 1

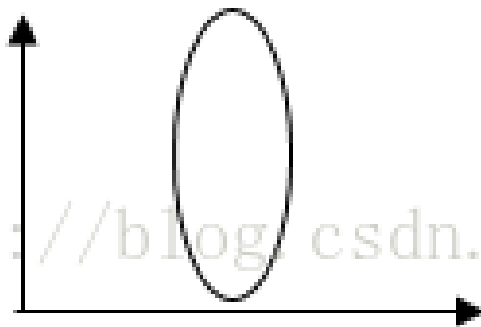


图 2

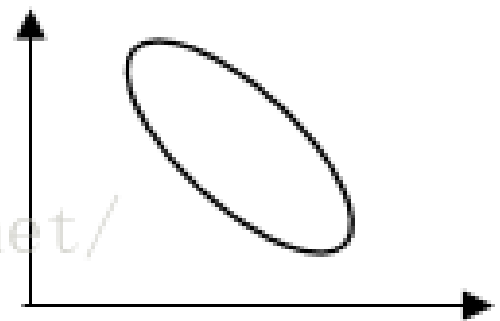


图 3

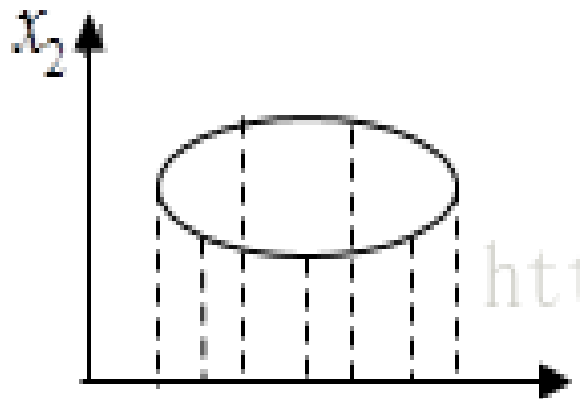


图 1

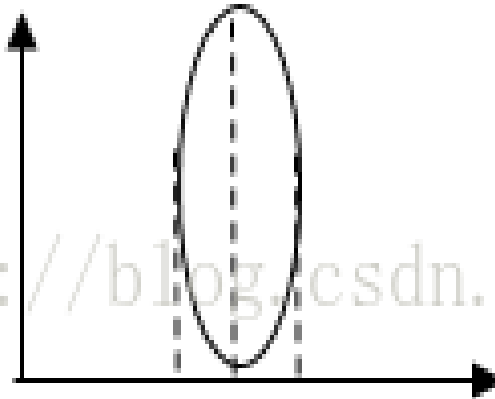


图 2

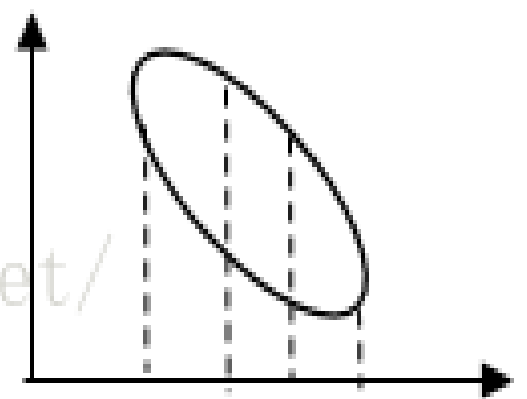
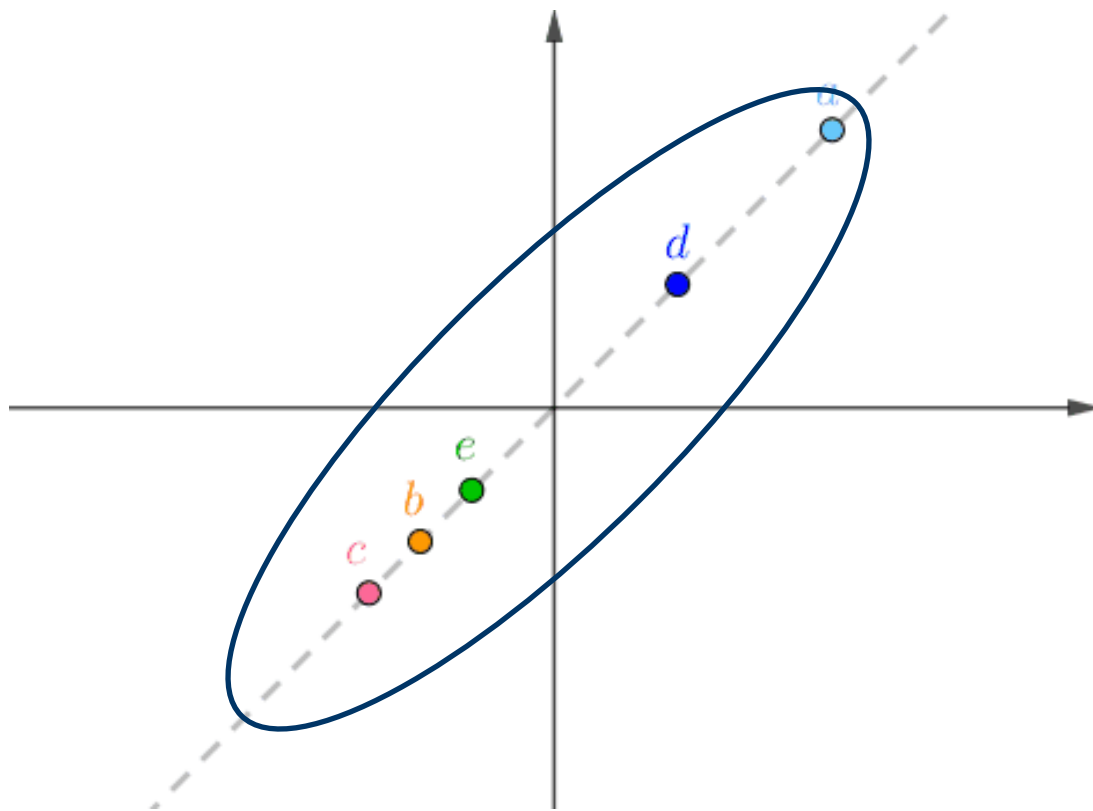


图 3



移动及旋转坐标系，去寻找主元成分





重新构造挑选房源的主元成分：

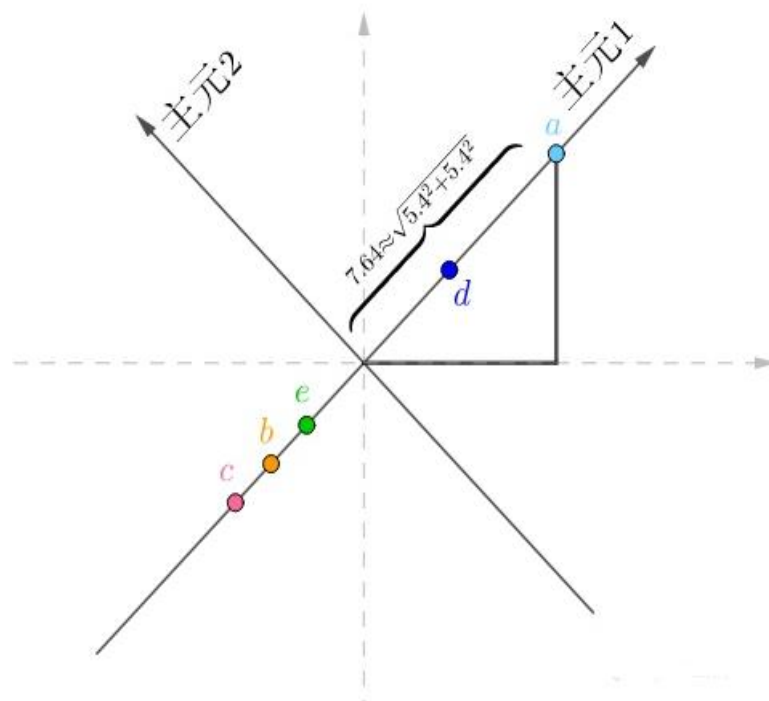
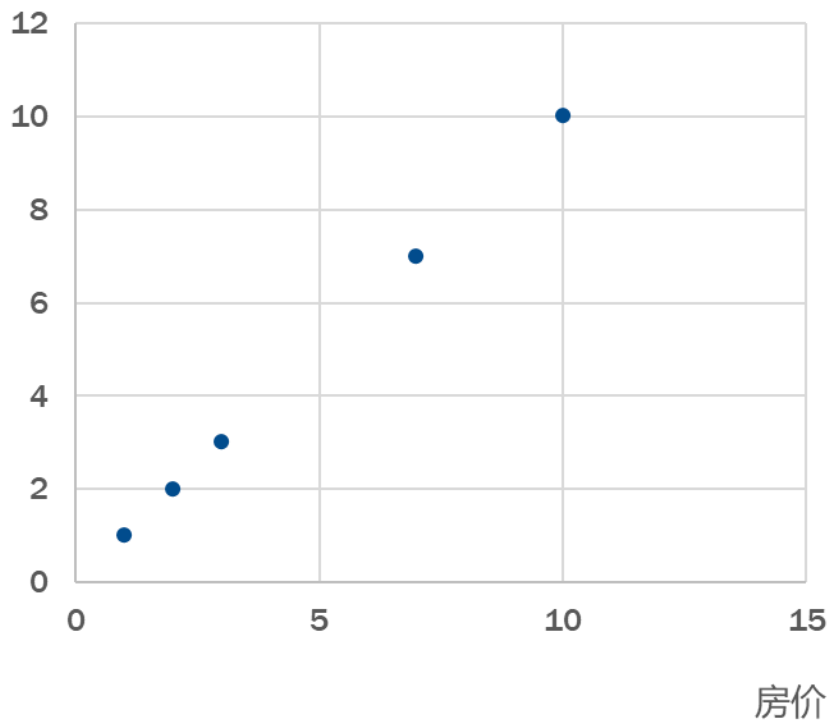
	房价(百万元)	面积(百平米)
<i>a</i>	10	10
<i>b</i>	2	2
<i>c</i>	1	1
<i>d</i>	7	7
<i>e</i>	3	3

主成分分析



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。（方差尽可能的大）

面积





把所有的房价换算到新的坐标系上：

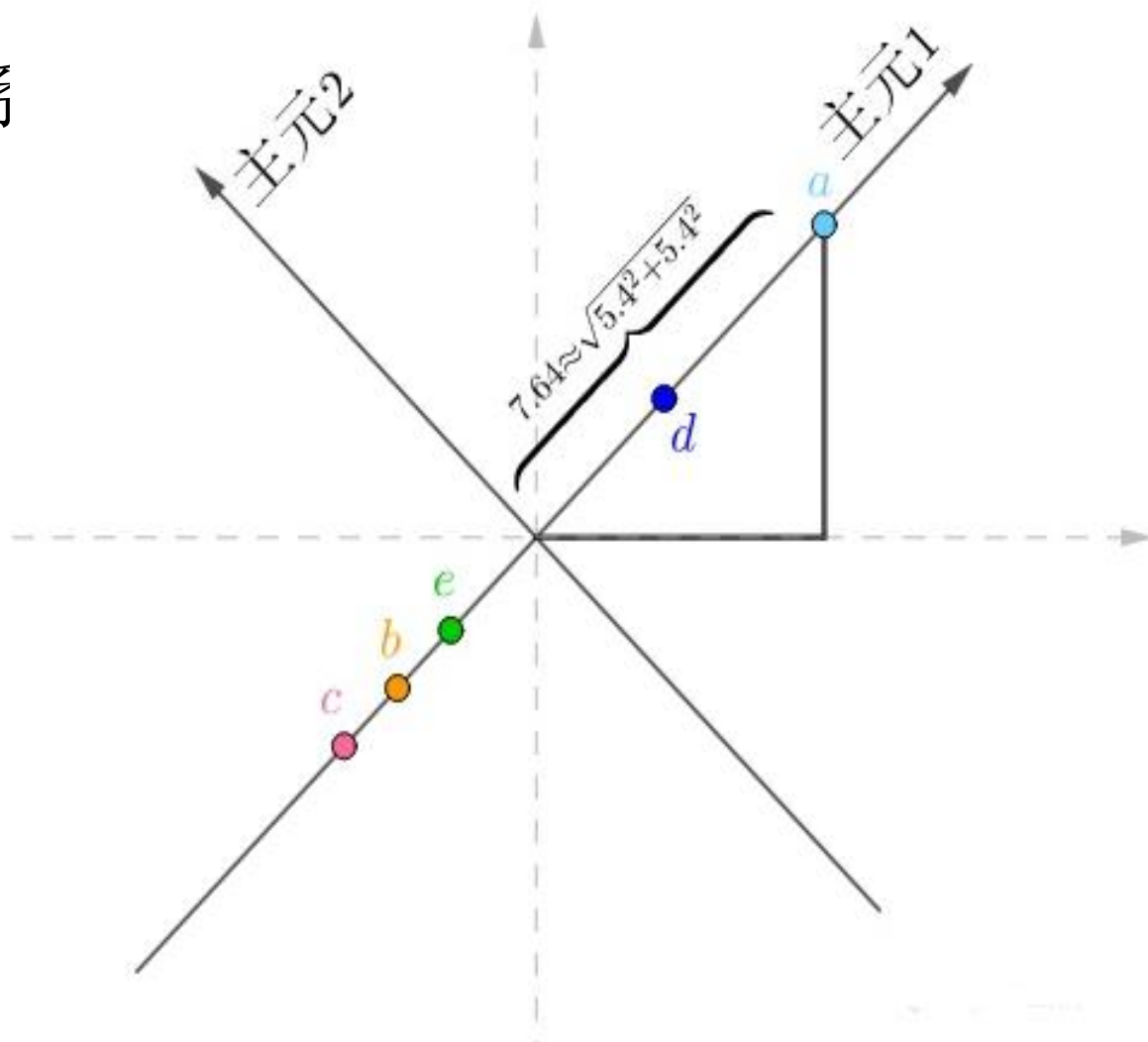
	房价(百万元)	面积(百平米)
<i>a</i>	10	10
<i>b</i>	2	2
<i>c</i>	1	1
<i>d</i>	7	7
<i>e</i>	3	3

	主元1	主元2
<i>a</i>	7.64	0
<i>b</i>	-3.68	0
<i>c</i>	-5.09	0
<i>d</i>	3.39	0
<i>e</i>	-2.26	0



旋转后的坐标系，横纵坐标不再代表“房价”、“面积”了，而是两者的混合，称作“主元1”。

用勾股定理计算出来，比如a在“主元1”的坐标值为





降维的方法有很多，而最为常用的就是PCA(主成分分析)。PCA 是将数据从原来的坐标系转换到新的坐标系，新的坐标系的选择是由数据本身决定的。

第一个新坐标轴选择的是原始数据中方差最大的方向，第二个新坐标轴的选择和第一个坐标轴正交且方差最大的方向。然后该过程一直重复，重复次数为原始数据中的特征数量。最后会发现大部分方差都包含在最前面几个新坐标轴中，因此可以忽略剩下的坐标轴，从而达到降维的目的。



将数据转换成 n 个主成分步骤如下：

- 1、去中心化（去除均值）
- 2、计算协方差矩阵
- 3、计算协方差矩阵的特征值和特征向量
- 4、将特征值从大到小排序
- 5、保留最上面的 n 个特征向量
- 6、将数据转换到上述 n 个特征向量构建的新空间中



去中心化：意思是将数据中每个维度上的均值变成 0。那为什么要这样做呢？**PCA** 实质上是找方差最大的方向，而方差的公式如下(其中 μ 为均值)：

$$Var(x) = \frac{1}{n} \sum_{i=1}^n (x - \mu)^2$$

如果将均值变成 0，那么方差计算起来就更加方便，如下：

$$Var(x) = \frac{1}{n} \sum_{i=1}^n (x)^2$$



均值为：

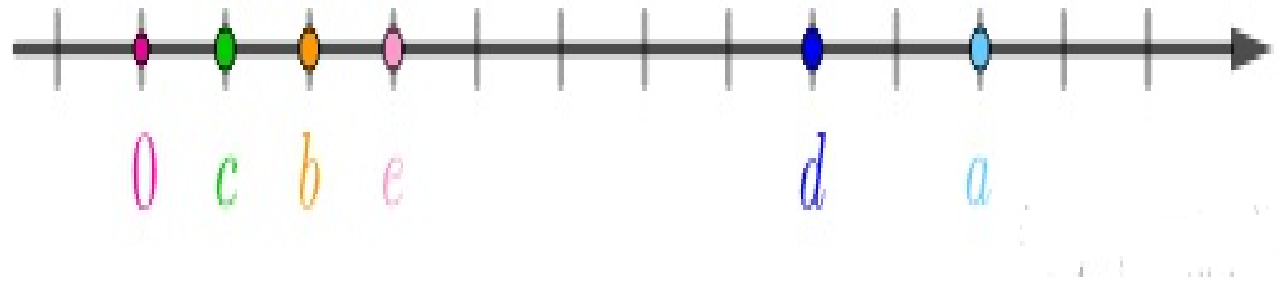
$$\bar{X} = \frac{X_1 + X_2 + X_3 + X_4 + X_5}{5} = \frac{10 + 2 + 1 + 7 + 3}{5} = 4.6$$

对数据进行“中
心化”处理

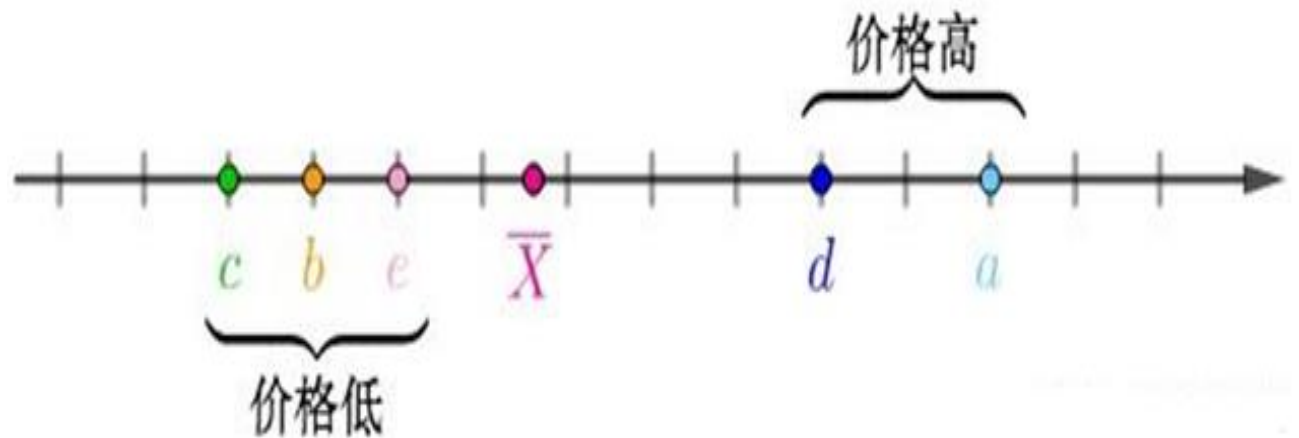
	房价(百万元)
<i>a</i>	$10 - \bar{X} = 5.4$
<i>b</i>	$2 - \bar{X} = -2.6$
<i>c</i>	$1 - \bar{X} = -3.6$
<i>d</i>	$7 - \bar{X} = 2.4$
<i>e</i>	$3 - \bar{X} = -1.6$



数据直接放在实数轴上:



中心化后放在实数轴上:



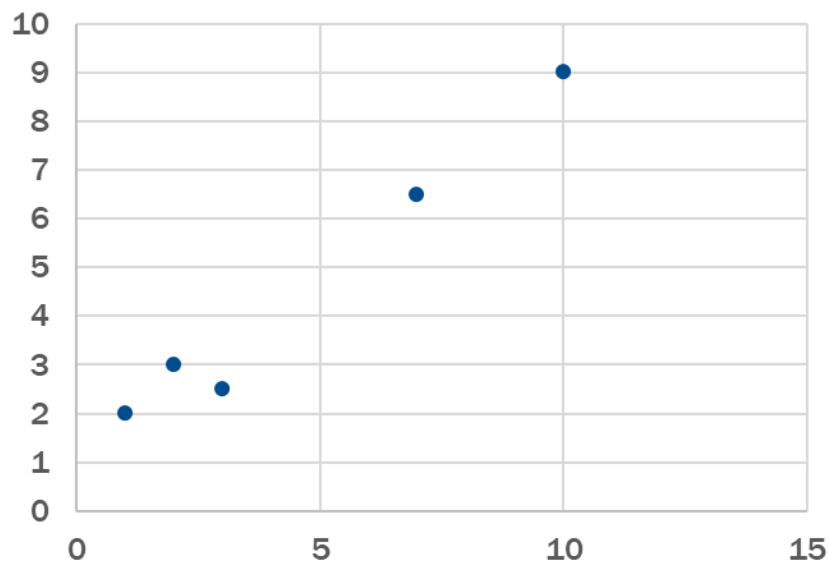
非理想情况如何降维?



房价和面积**正相关**（不是正交）

	房价(百万元)	面积(百平米)
<i>a</i>	10	9
<i>b</i>	2	3
<i>c</i>	1	2
<i>d</i>	7	6.5
<i>e</i>	3	2.5

面积



房价

非理想情况如何降维?



房价和面积正相关——中心化

	房价(百万元)	面积(百平米)
<i>a</i>	10	9
<i>b</i>	2	3
<i>c</i>	1	2
<i>d</i>	7	6.5
<i>e</i>	3	2.5

中心化
→

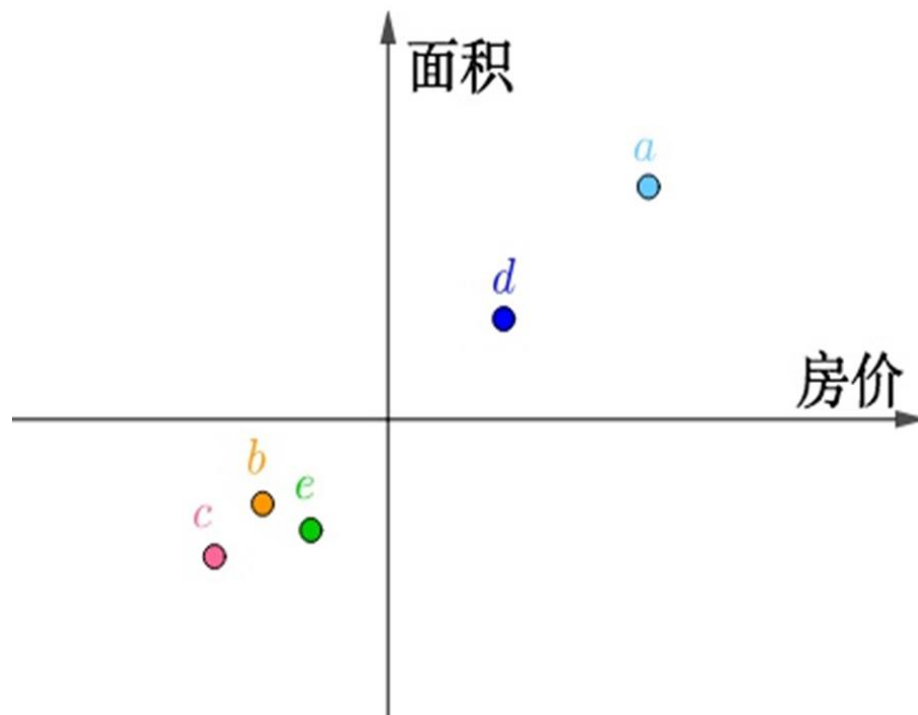
	房价(百万元)	面积(百平米)
<i>a</i>	5.4	4.4
<i>b</i>	-2.6	-1.6
<i>c</i>	-3.6	-2.6
<i>d</i>	2.4	1.9
<i>e</i>	-1.6	-2.1

非理想情况如何降维?



房价和面积正相关

	房价(百万元)	面积(百平米)
<i>a</i>	5.4	4.4
<i>b</i>	-2.6	-1.6
<i>c</i>	-3.6	-2.6
<i>d</i>	2.4	1.9
<i>e</i>	-1.6	-2.1





将数据转换成 n 个主成分步骤如下：

- 1、去中心化（去除均值）
- 2、计算协方差矩阵
- 3、计算协方差矩阵的特征值和特征向量
- 4、将特征值从大到小排序
- 5、保留最上面的 n 个特征向量
- 6、将数据转换到上述 n 个特征向量构建的新空间中

协方差描述的是两个特征之间的相关性，当协方差为正时，两个特征呈正相关关系（同增同减）；当协方差为负时，两个特征呈负相关关系（一增一减）；当协方差为0时，两个特征之间没有任何相关关系。



协方差的数学定义如下(假设样本有 x 和 y 两种特征, 而 X 就是包含所有样本的 x 特征的集合, Y 就是包含所有样本的 y 特征的集合):

$$\text{conv}(X, Y) = \frac{\sum_{i=1}^n (x_i - \mu_x) \sum_{i=1}^n (y_i - \mu_y)}{n - 1}$$

如果在算协方差之前做了 `demean` 操作, 那么公式则为:

$$\text{conv}(X, Y) = \frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n - 1}$$



假设样本只有 X 和 Y 这两个特征，现在把 X 与 X，X 与 Y，Y 与 X，Y 与 Y 的协方差组成矩阵，那么就构成了协方差矩阵。而协方差矩阵反应的就是特征与特征之间的相关关系。

	X	Y
X	<code>cov(X,X)</code>	<code>cov(X,Y)</code>
Y	<code>cov(Y,X)</code>	<code>cov(Y,Y)</code>

NumPy 提供了计算协方差矩阵的函数 `cov`，示例代码如下：

```
1. import numpy as np
2.
3. # 计算after_demean的协方差矩阵
4. # after_demean的行数为样本个数，列数为特征个数
5. # 由于cov函数的输入希望是行代表特征，列代表数据的矩阵，所以要转置
6. cov = np.cov(after_demean.T)
```



特征值与特征向量

特征值与特征向量的数学定义：如果向量 \mathbf{v} 与矩阵 \mathbf{A} 满足 $\mathbf{A}\mathbf{v}=\lambda\mathbf{v}$ ，则称向量 \mathbf{v} 是矩阵 \mathbf{A} 的一个特征向量， λ 是相应的特征值。

因为协方差矩阵为方阵，所以我们可以计算协方差矩阵的特征向量和特征值。其实这里的特征值从某种意义上来说体现了方差的大小，特征值越大方差就越大。而特征值所对应的特征向量就代表将原始数据进行坐标轴转换之后的数据。

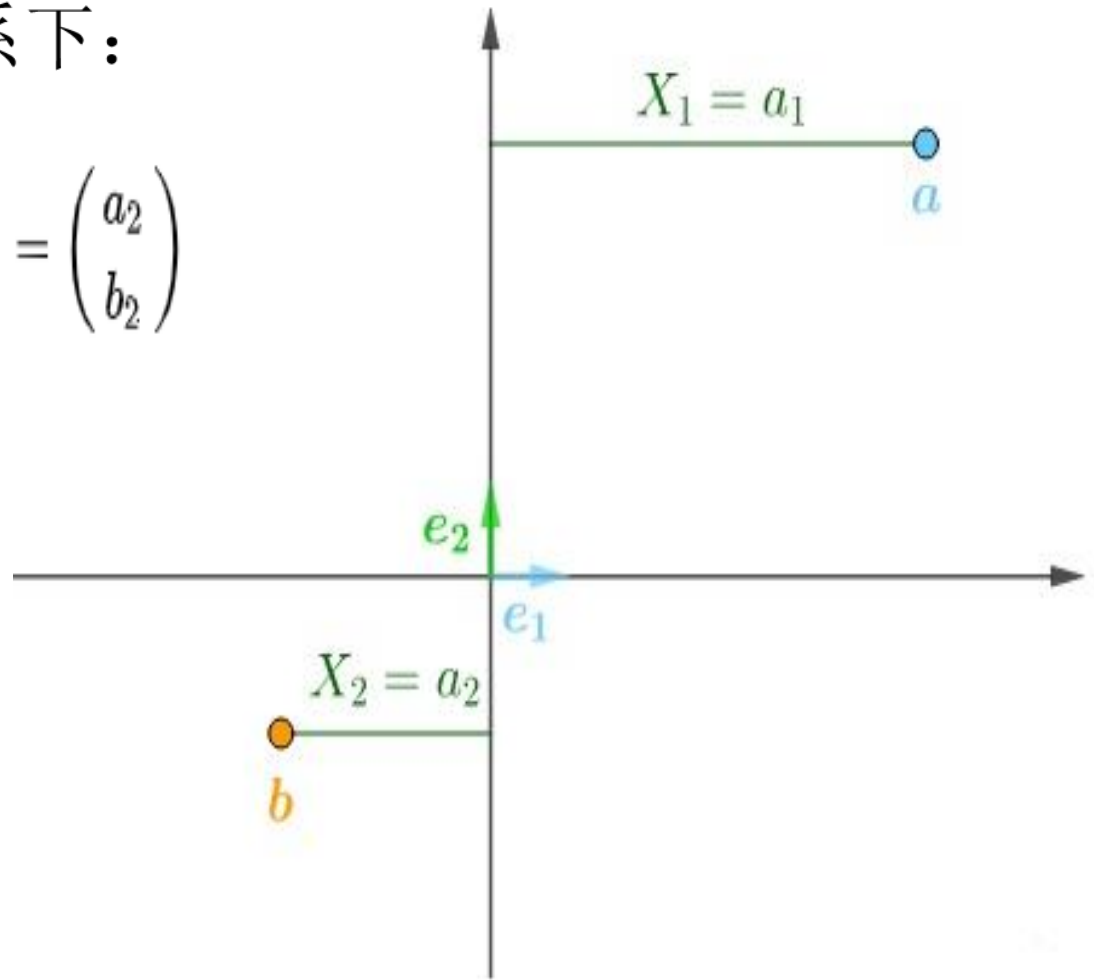
numpy 为我们提供了计算特征值与特征向量的接口 `eig`，示例代码如下：

```
1. import numpy as np
2.
3. #eig函数为计算特征值与特征向量的函数
4. #cov为矩阵，value为特征值，vector为特征向量
5. value, vector = np.linalg.eig(cov)
```



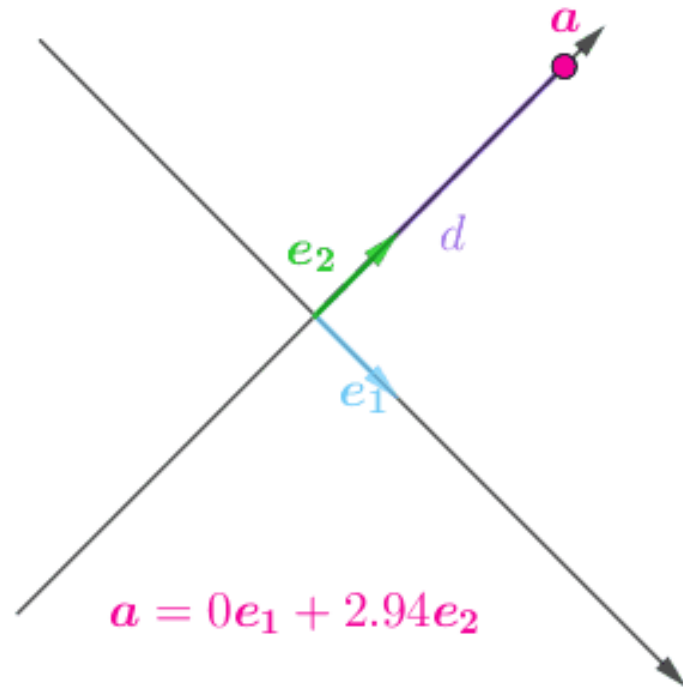
这两个点在初始坐标系下：

$$\mathbf{a} = \begin{pmatrix} X_1 \\ Y_1 \end{pmatrix} = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} X_2 \\ Y_2 \end{pmatrix} = \begin{pmatrix} a_2 \\ b_2 \end{pmatrix}$$





随着坐标系的不同， x_1 、 x_2 的值会不断变化：





将数据转换成 n 个主成分步骤如下：

- 1、中心化（去除均值）
- 2、计算协方差矩阵
- 3、计算协方差矩阵的特征值和特征向量
- 4、将特征值从大到小排序
- 5、保留最上面的 n 个特征向量
- 6、将数据转换到上述 n 个特征向量构建的新空间中



- 协方差矩阵:

$$X = \begin{pmatrix} a_1 & a_2 & \dots & a_m \\ b_1 & b_2 & \dots & b_m \end{pmatrix}$$

$$Cov = \frac{1}{m} X X^T = \begin{pmatrix} \frac{1}{m} \sum_{i=1}^m a_i^2 & \frac{1}{m} \sum_{i=1}^m a_i b_i \\ \frac{1}{m} \sum_{i=1}^m a_i b_i & \frac{1}{m} \sum_{i=1}^m b_i^2 \end{pmatrix}$$

- 协方差对角化:

$$P C P^T = \begin{bmatrix} \lambda_1 & & & & \\ & \lambda_{12} & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & \lambda_n \end{bmatrix}$$



将数据转换成 n 个主成分步骤如下：

- 1、中心化（去除均值）
- 2、计算协方差矩阵
- 3、计算协方差矩阵的特征值和特征向量
- 4、将特征值从大到小排序
- 5、保留最上面的 n 个特征向量
- 6、将数据转换到上述 n 个特征向量构建的新空间中



- 计算协方差矩阵 C 的特征根和主成分矩阵，保留前 q 个最大的特征根及对应的特征向量，其中最大特征根对应的特征向量称为第一主成分，第二大特征根对应的是第二主成分，... $\lambda_1 \geq \lambda_2 \geq \dots, \geq \lambda_m \geq 0$
- 构造主成分矩阵 P ，其中其列向量 p_i 是第 i 个主成分
- 假设降序排列的特征根为 $\frac{\sum_{i=1}^q \lambda_i}{\sum_{k=1}^m \lambda_k}$ ，第 i 个主成分的贡献率的计算如下：
$$\frac{\lambda_i}{\sum_{k=1}^m \lambda_k} \quad (i=1, 2, \dots, m)$$
- 计算最终降维后的数据集 Y ， $Y=XP$ ，其中 P 是主成分矩阵， X 是步骤 1 中得到的矩阵。



要想尽量多分配给x1、x2，借鉴最小二乘法就是让：

$$X_1^2 + X_2^2 = \sum_{i=0}^2 X_i^2 \text{ 最大}$$

$$\mathbf{e}_1 = \begin{pmatrix} e_{11} \\ e_{12} \end{pmatrix} \quad \mathbf{e}_2 = \begin{pmatrix} e_{21} \\ e_{22} \end{pmatrix}$$

x1、x2可以表示为：

$$X_1 = \mathbf{a} \cdot \mathbf{e}_1 = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \cdot \begin{pmatrix} e_{11} \\ e_{12} \end{pmatrix} = a_1 e_{11} + b_1 e_{12}$$

$$X_2 = \mathbf{b} \cdot \mathbf{e}_1 = \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} \cdot \begin{pmatrix} e_{11} \\ e_{12} \end{pmatrix} = a_2 e_{11} + b_2 e_{12}$$



$$X_1 = \mathbf{a} \cdot \mathbf{e}_1 = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \cdot \begin{pmatrix} e_{11} \\ e_{12} \end{pmatrix} = a_1 e_{11} + b_1 e_{12}$$

$$X_2 = \mathbf{b} \cdot \mathbf{e}_1 = \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} \cdot \begin{pmatrix} e_{11} \\ e_{12} \end{pmatrix} = a_2 e_{11} + b_2 e_{12}$$

要想尽量多分配给 x_1 、 x_2 ， $X_1^2 + X_2^2 = \sum_{i=1}^2 X_i^2$ 最大

$$\begin{aligned} X_1^2 + X_2^2 &= (a_1 e_{11} + b_1 e_{12})^2 + (a_2 e_{11} + b_2 e_{12})^2 \\ &= a_1^2 e_{11}^2 + 2a_1 b_1 e_{11} e_{12} + b_1^2 e_{12}^2 + a_2^2 e_{11}^2 + 2a_2 b_2 e_{11} e_{12} + b_2^2 e_{12}^2 \\ &= (a_1^2 + a_2^2) e_{11}^2 + 2(a_1 b_1 + a_2 b_2) e_{11} e_{12} + (b_1^2 + b_2^2) e_{12}^2 \end{aligned}$$



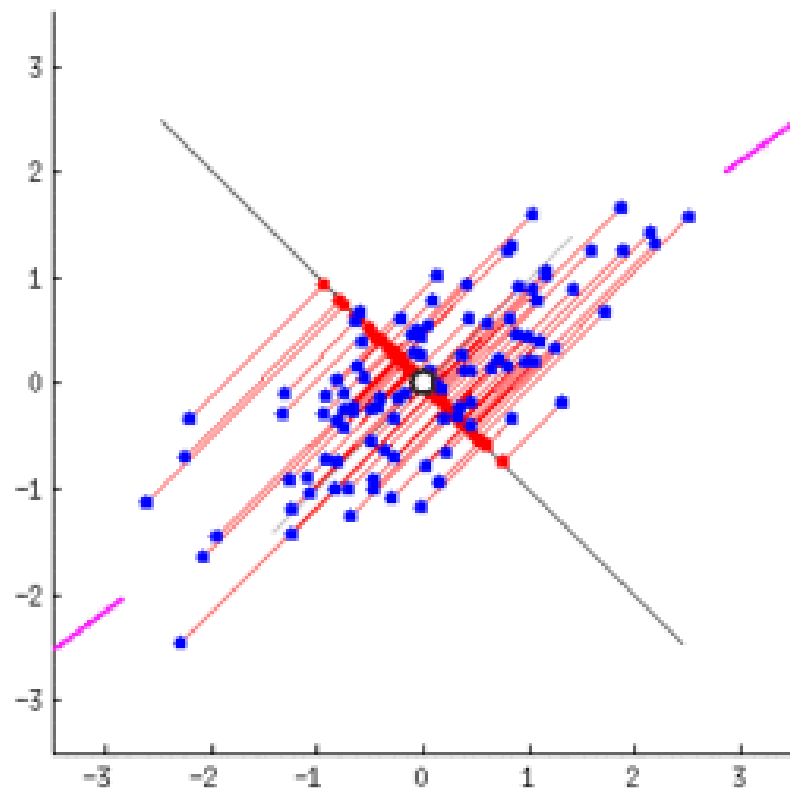
$$\begin{aligned}X_1^2 + X_2^2 &= (a_1 e_{11} + b_1 e_{12})^2 + (a_2 e_{11} + b_2 e_{12})^2 \\&= a_1^2 e_{11}^2 + 2a_1 b_1 e_{11} e_{12} + b_1^2 e_{12}^2 + a_2^2 e_{11}^2 + 2a_2 b_2 e_{11} e_{12} + b_2^2 e_{12}^2 \\&= (a_1^2 + a_2^2) e_{11}^2 + 2(a_1 b_1 + a_2 b_2) e_{11} e_{12} + (b_1^2 + b_2^2) e_{12}^2\end{aligned}$$

二次型：

$$X_1^2 + X_2^2 = \mathbf{e}_1^T \underbrace{\begin{pmatrix} a_1^2 + a_2^2 & a_1 b_1 + a_2 b_2 \\ a_1 b_1 + a_2 b_2 & b_1^2 + b_2^2 \end{pmatrix}}_P \mathbf{e}_1 = \mathbf{e}_1^T P \mathbf{e}_1$$



如果是两个点 $a = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, b = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$,



为了降维，应该选择尽量多分配给 x_1 、 x_2 ，少分配给 y_1 、 y_2 的坐标系。



- **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, 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。

主成分分析步骤



- 中心化数据集, 使得每个变换后的属性的均值为零。

$$X_{ij} = X_{ij} - A_j$$

- 计算协方差矩阵C, 元素Cij是属性Ai和Aj之间的协方差:

$$C_{ij} = \text{cov}(A_i, A_j)$$

- 计算协方差矩阵C的特征根和主成分矩阵, 保留前q个最大的特征根及对应的特征向量, 构造主成分矩阵P, 其中其列向量pi是第i个主成分。

$$\frac{\lambda_i}{\sum_{k=1}^m \lambda_k} (i = 1, 2, \dots, m)$$

$$\frac{\sum_{i=1}^q \lambda_i}{\sum_{k=1}^m \lambda_k}$$

取累计贡献率达到85%~95%的前q个特征值对应的特征向量为q个主成分。

- 计算最终降维后的数据集Y, $Y = XP$ 。



1、原始数据：
$$\begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix}$$

2、计算协方差矩阵：

$$C = \frac{1}{5} \begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} -1 & -2 \\ -1 & 0 \\ 0 & 0 \\ 2 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{6}{5} & \frac{4}{5} \\ \frac{4}{5} & \frac{6}{5} \end{pmatrix}$$



3、获取特征值： $\lambda_1 = 2, \lambda_2 = 2/5$

4、获取特征向量： $C_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix}, C_2 \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

5、特征向量一定能使协方差矩阵对角化：

$$PCP^T = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 6/5 & 4/5 \\ 4/5 & 6/5 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2/5 \end{pmatrix}$$



6、将特征向量进行标准化，然后降维

$$Y = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} -3/\sqrt{2} & -1/\sqrt{2} & 0 & 3/\sqrt{2} & -1/\sqrt{2} \end{pmatrix}$$



■ PCA性能分析

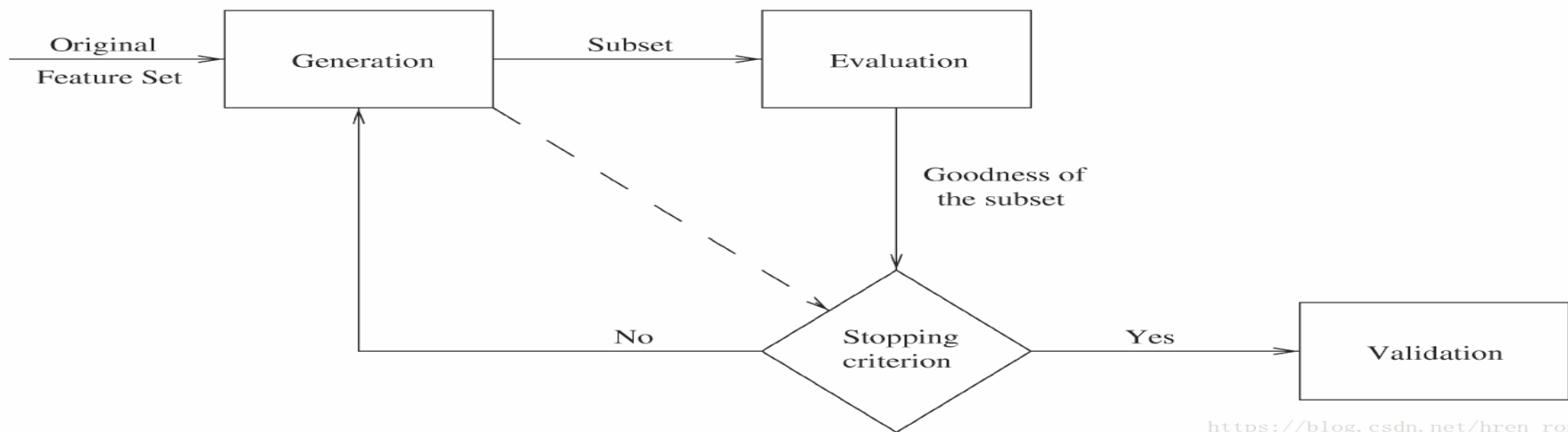
- 优点：降低数据的复杂性，识别最重要的多个特征
- 缺点：不一定需要，且可能损失有用信息
- 适用数据类型：数值型数据

第二章 知识发现过程与应用结构



■ 如何做特征选择

- 1、生成过程：生成候选的特征子集；
- 2、评价函数：评价特征子集的好坏；
- 3、停止条件：决定什么时候该停止；
- 4、验证过程：特征子集是否有效；



https://blog.csdn.net/hren_ron

选做作业：Iris 鸢尾花数据集预处理



Iris 鸢尾花数据集是一个经典数据集，在统计学习和机器学习领域都经常被用作示例。数据集内包含 3 类共 150 条记录，每类各 50 个数据，每条记录都有 4 项特征：花萼长度、花萼宽度、花瓣长度、花瓣宽度，可以通过这4个特征预测鸢尾花卉属于（iris-setosa, iris-versicolour, iris-virginica）中的哪一品种。



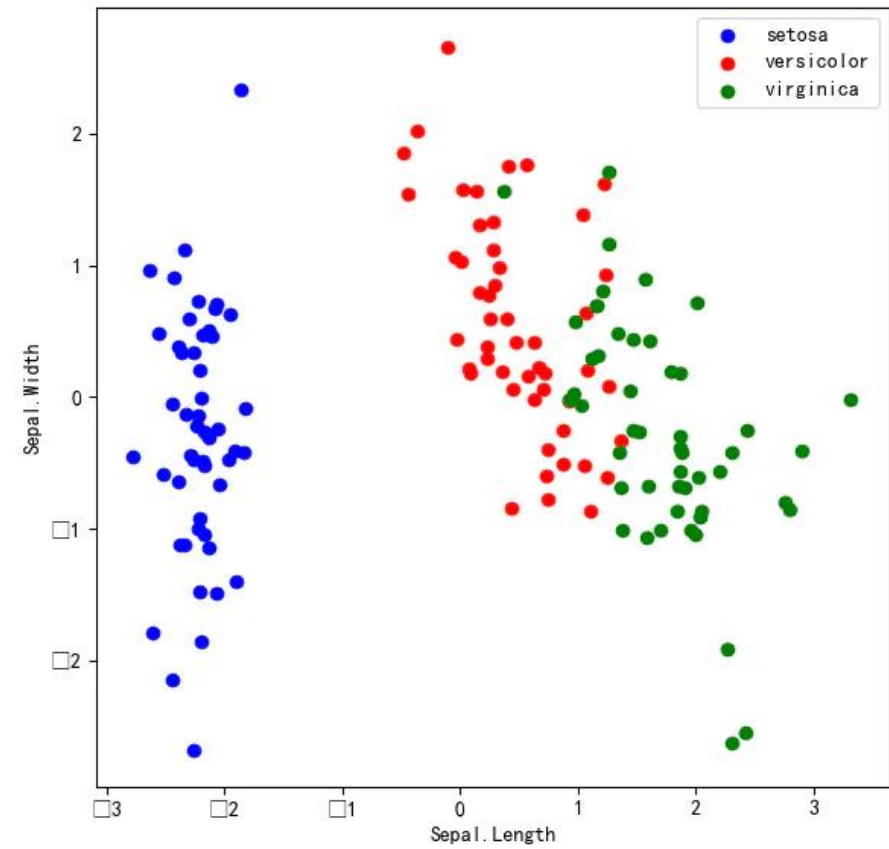
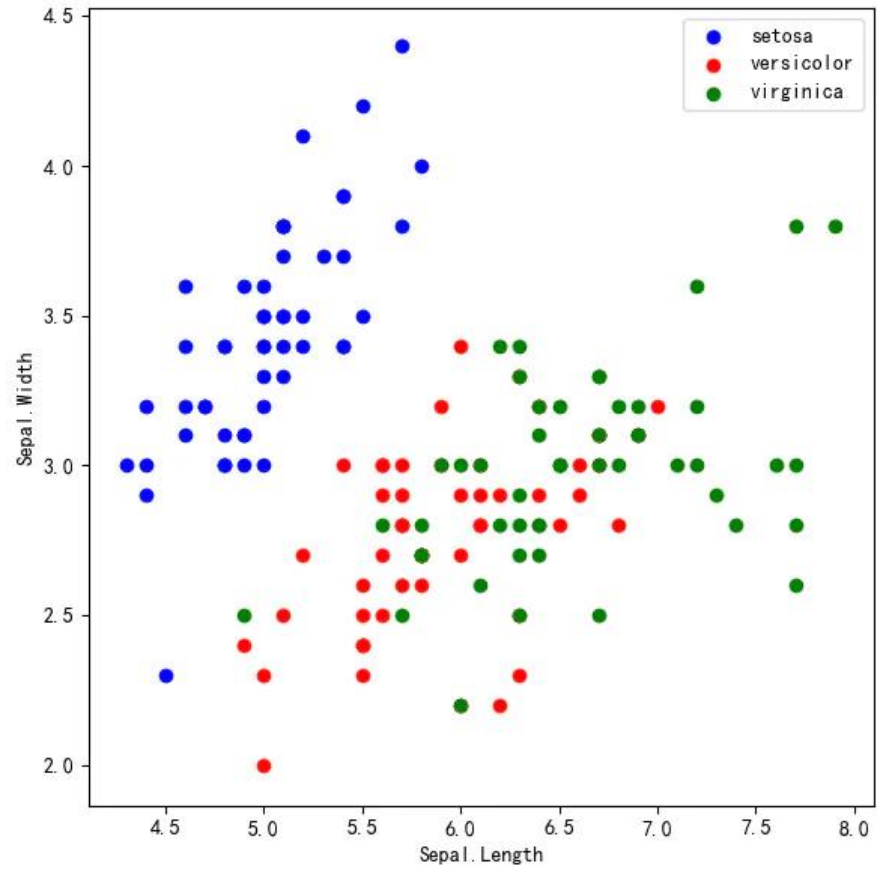


中文名	特征描述
无附属物亚属	根状茎明显，根绳索状，不为纺锤形；花被管明显；外花被裂片倒卵形，基部逐渐过渡成狭窄的爪，中脉上无附属物，少数种生有单细胞的纤毛；蒴果大多无喙；种子梨形、半圆形或圆形，有时压扁，通常无翼或沿边缘有狭窄的翼状突起。
无附属物组	花柱分裂大多至基部；花凋谢后花被管不残存在果实上。
紫苞鸢尾组	根状茎纤细；花茎上有1朵花；种子光亮，受潮后附属物变粘。
单苞鸢尾组	植株冬季常绿，夏季枯萎；根状茎粗壮，近地表处膨大成球形；苞片1枚。
琴瓣鸢尾亚属	根的内皮层细胞沿辐射方向明显的伸长；外花被裂片提琴形；蒴果有喙和石条明显而突出的棱，每两条棱成对靠近；种子具膜质而膨起的种皮。
尼泊尔鸢尾亚属	根状茎小，被毛发状的枯叶残留纤维；根肉质，肥厚，纺锤形。

作业：对鸢尾花数据进行主成分分析



```
1  def pca():
2      """
3      主成分分析进行特征降维
4      :return:
5      """
6      #读取数据集
7      data = pd.read_csv('../..数据集/机器学习/分类算法/鸢尾花数据集/iris.csv')
8      #划分特征值与目标值
9      x = data.iloc[:,0:4].values
10     y = data.iloc[:,4].values
11     print(x,y)
12     return None
13
14 if __name__=="__main__":
15     pca()
```



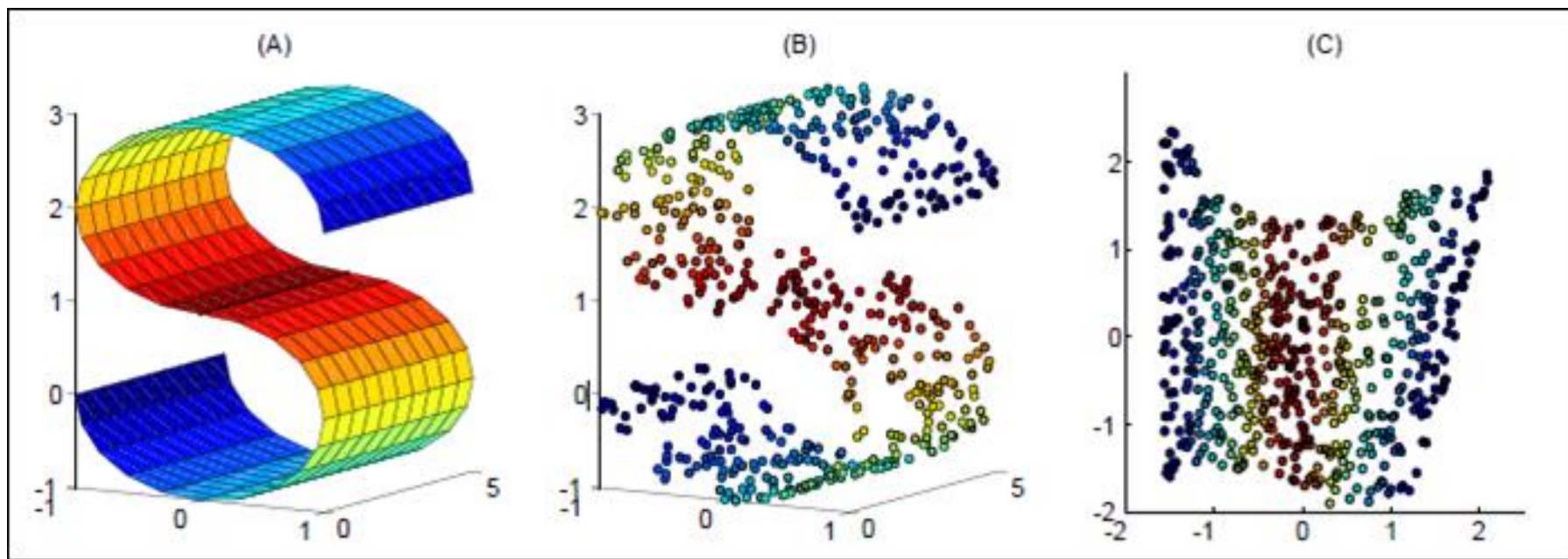


- Linear Discriminant Analysis(也有叫做Fisher Linear Discriminant)是一种有监督的 (supervised) 线性降维算法。
 - 1、同类的数据点尽可能的接近 (within class)
 - 2、不同类的数据点尽可能的分开 (between class)
- PCA保持数据信息不同, LDA是为了使得降维后的数据点尽可能地容易被区分!

局部线性嵌入 (LLE)



- 局部线性嵌入：Locally linear embedding，是一种非线性降维算法，它能够使降维后的数据较好地保持原有流形结构。



Laplacian Eigenmaps 拉普拉斯特征映射



拉普拉斯特征映射：是从局部近似的角度去构建数据之间的关系。将要降维的数据构建成图，图中的每个节点和距离它最近的 K 个节点建立边关系。然后图中相连的点（原始空间中相互靠近的点）在降维后的空间中也尽可能地靠近，从而在降维后仍能保持原有的局部结构关系。

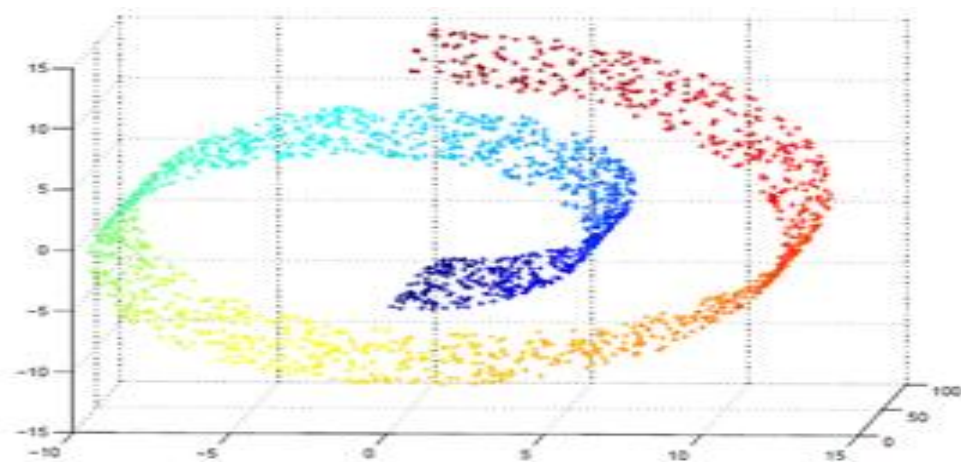
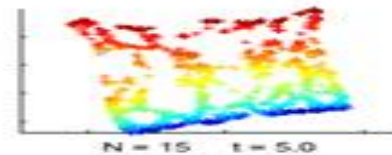
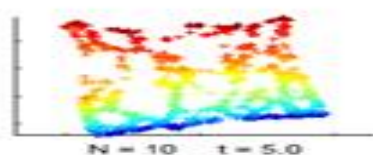
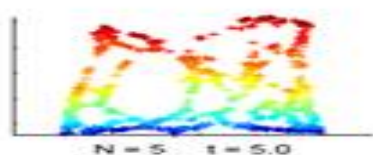


Figure 1: 2000 random data points on the "swiss roll".





- 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、PCA、NN、DL…）
- 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，需要有不同的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。



3主成分分析 --- PCA

提交中

未开启补交

[详情](#) [立即截止](#) [进入学习](#) [更多](#)

陶洁 已开始做题 10 人 未开始做题 188 人 已完成做题 1 人 提交剩余时间: 12 天 1 小时 22 分



4使用PyTorch训练MNIST数据集上的卷积神经网络

提交中

未开启补交

[详情](#) [立即截止](#) [进入学习](#) [更多](#)

陶洁 已开始做题 0 人 未开始做题 198 人 已完成做题 0 人 提交剩余时间: 21 天 0 小时 22 分



PyTorch简介

进行深度学习研究离不开深度学习框架。目前主流的深度学习框架包括 Google 发布的 TensorFlow、Facebook 发布的 PyTorch 和 Amazon 发布的 MXNet 等。

根据深度学习框架的运行方式，通常可以分为静态图设计和动态图设计两种。在静态图设计中，计算图的构建在网络模型的计算之前，对于用户定义的网络模型，深度学习框架会先进行一个“编译”的过程，将网络模型转化成计算图，之后的计算都根据这个计算图来进行。

使用PyTorch建立TinyNet网络模型

在本实训中，你需要使用 PyTorch 实现一个用于 MNIST 手写数字识别任务的 TinyNet 网络模型。MNIST 数据集包含 60000 张训练图片和 10000 张测试图片，每张图片是一个 28×28 的黑白图像，每张包含一个手写数字。因为数字是从 0-9，所以这个任务可以看作是一个有 10 个类别的分类任务。因为是黑白图像，所以输入图片只有一个通道，是一个 $(B, 1, 28, 28)$ 的 Tensor，其中 B 是 batch size。下图展示了一部分 MNIST 数据集中的数据。





序号	类型	参数	输出特征图大小
0	输入	(B, 1, 20, 20)	-
1	卷积层	输出通道32, 卷积核大小3x3, 步长1, 填充1	(B, 32, 20, 20)
2	激活函数	ReLU	(B, 32, 20, 20)
3	卷积层	输出通道64, 卷积核大小3x3, 步长1, 填充1	(B, 64, 20, 20)
4	激活函数	ReLU	(B, 64, 20, 20)
5	池化层	最大值池化, 池化窗口2x2, 步长2, 填充0	(B, 64, 10, 10)
6	Dropout	p=0.25	(B, 64, 10, 10)
7	全连接层	输出神经元128	(B, 128)
4	激活函数	ReLU	(B, 128)
6	Dropout	p=0.5	(B, 128)



2模型评估、选择与验证

已截止

[详情](#) [更多](#)

陶洁 已开始做题 180 人 未开始做题 18 人 已完成做题 141 人 评阅剩余时间: 92 天 10 小时 50 分



1数据挖掘 --- 绪论

已截止

[详情](#) [更多](#)

陶洁 已开始做题 187 人 未开始做题 11 人 已完成做题 186 人 评阅剩余时间: 92 天 10 小时 50 分

请输入姓名或者学号搜索



11个检索结果 (11个学生)

序号	姓名	学号	作业状态	实训总耗时	最新完成关卡	结束前完成关卡	关卡得分
1	李伟峰	190501...	未开启	--	0/3	0/3	--
2	曹凯	190501...	未开启	--	0/3	0/3	--
3	唐湘豫	190505...	未开启	--	0/3	0/3	--
4	潘汇贤	190501...	未开启	--	0/3	0/3	--
5	蒋鸿宇	190501...	未开启	--	0/3	0/3	--
6	xujie	190505...	未开启	--	0/3	0/3	--
7	胡宏飞	190501...	未开启	--	0/3	0/3	--
8	郑涛	190505...	未开启	--	0/3	0/3	--
9	李沛灏	190501...	未开启	--	0/3	0/3	--
10	危星宇	190204...	未开启	--	0/3	0/3	--
11	徐安德	190505...	未开启	--	0/3	0/3	--

第二章 知识发现过程与应用结构

内容提要

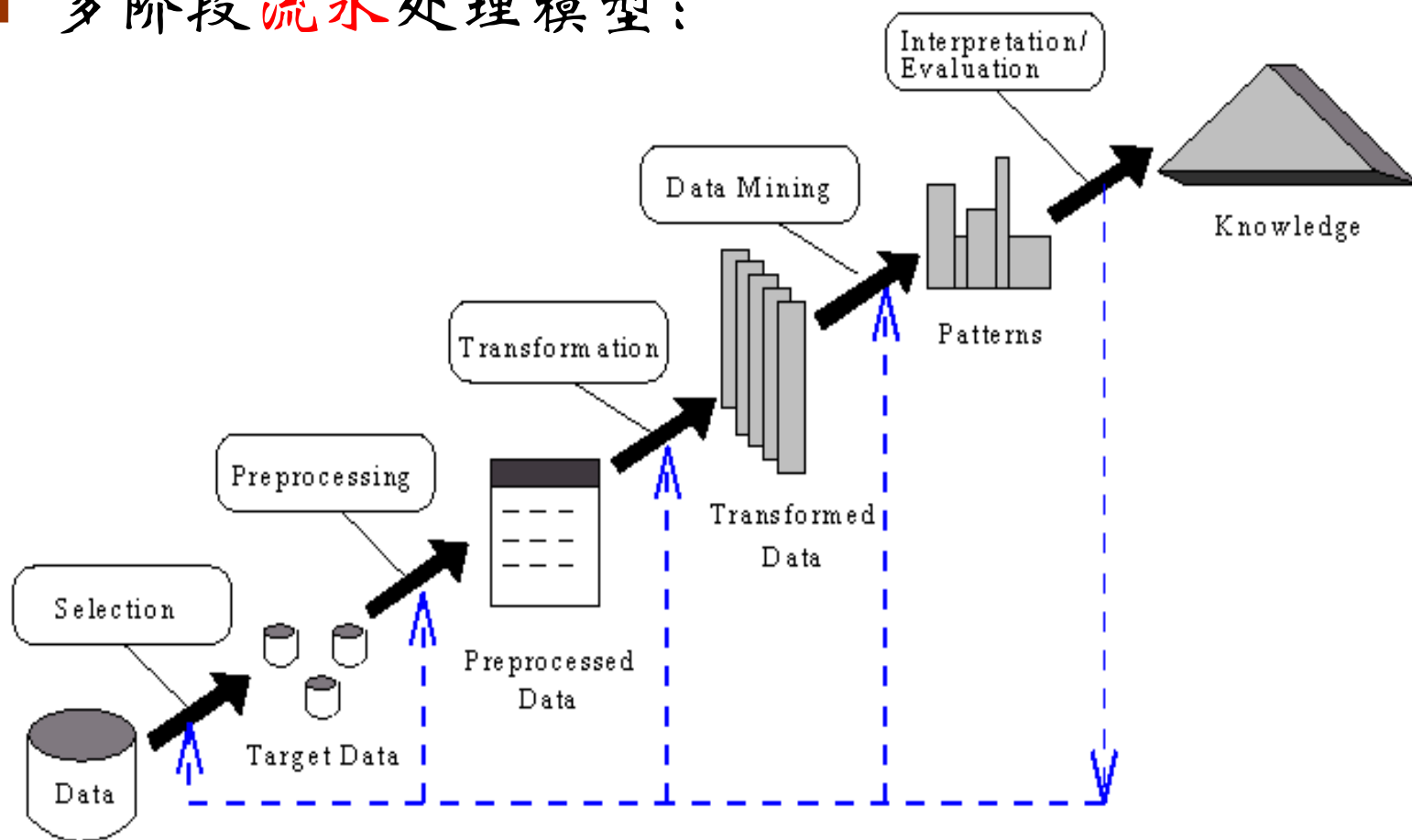
- 知识发现的基本过程
- 数据库中的知识发现处理过程模型
- 知识发现软件或工具的发展
- 知识发现项目的过程化管理
- 数据挖掘语言介绍



小结——KDD基本过程



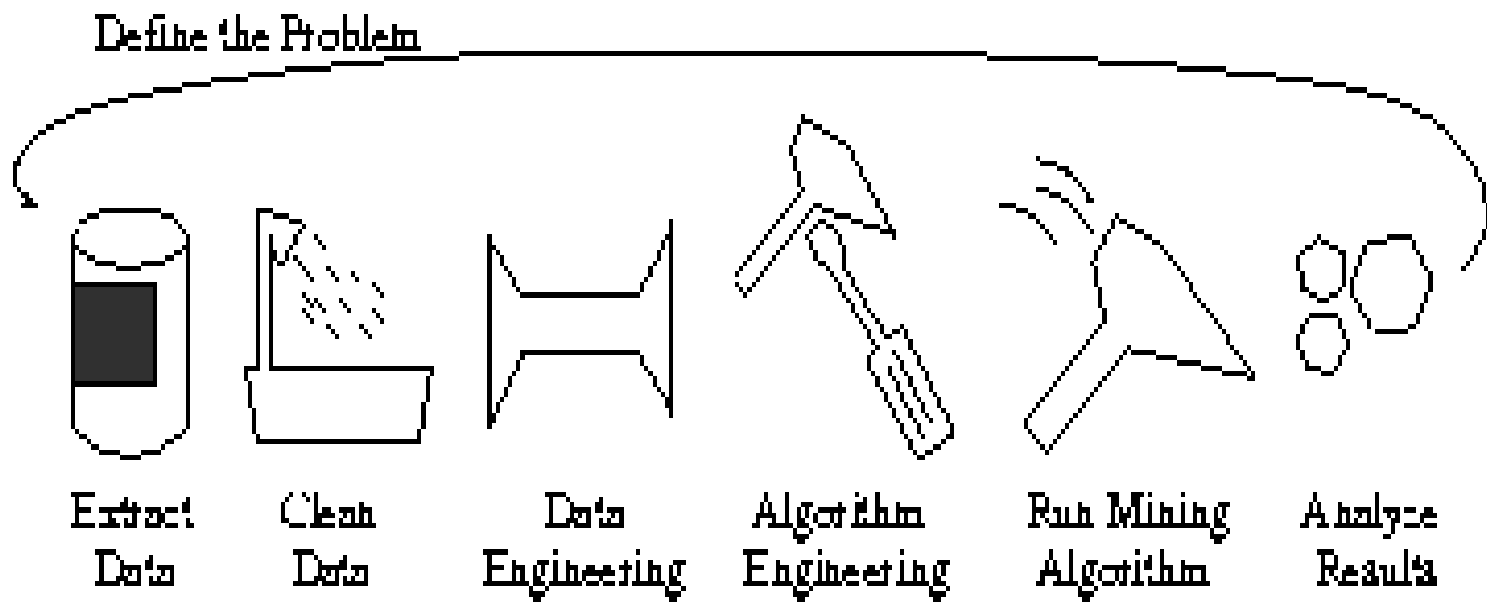
■ 多阶段流水处理模型：





螺旋处理过程模型

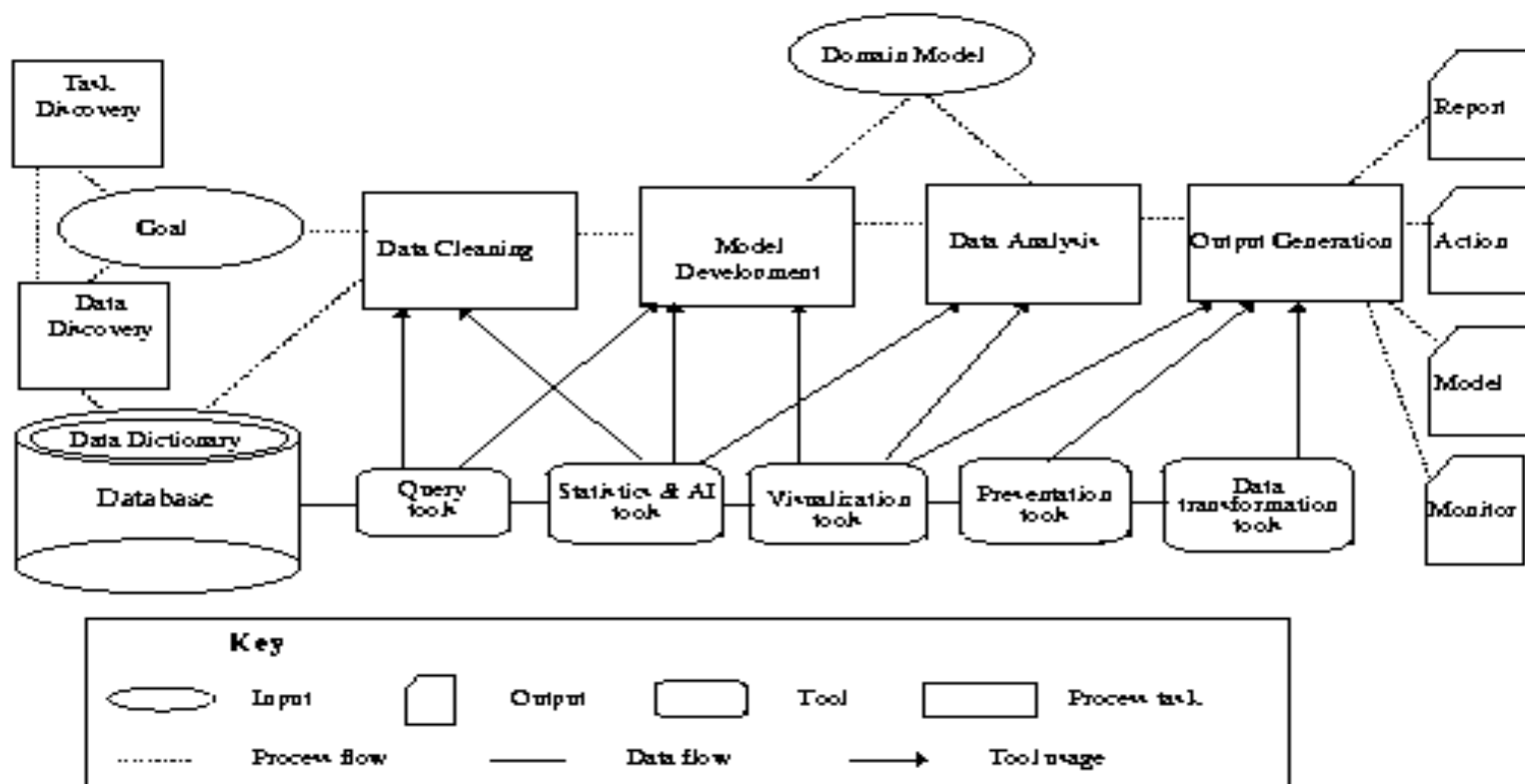
- 它强调领域专家参与的重要性，并以问题的定义为中心循环评测挖掘的结果。当结果不令人满意时，就需要重新定义问题，开始新的处理循环。每次循环都使问题更清晰，结果更准确，因此是一个螺旋式上升过程。





以用户为中心的处理模型

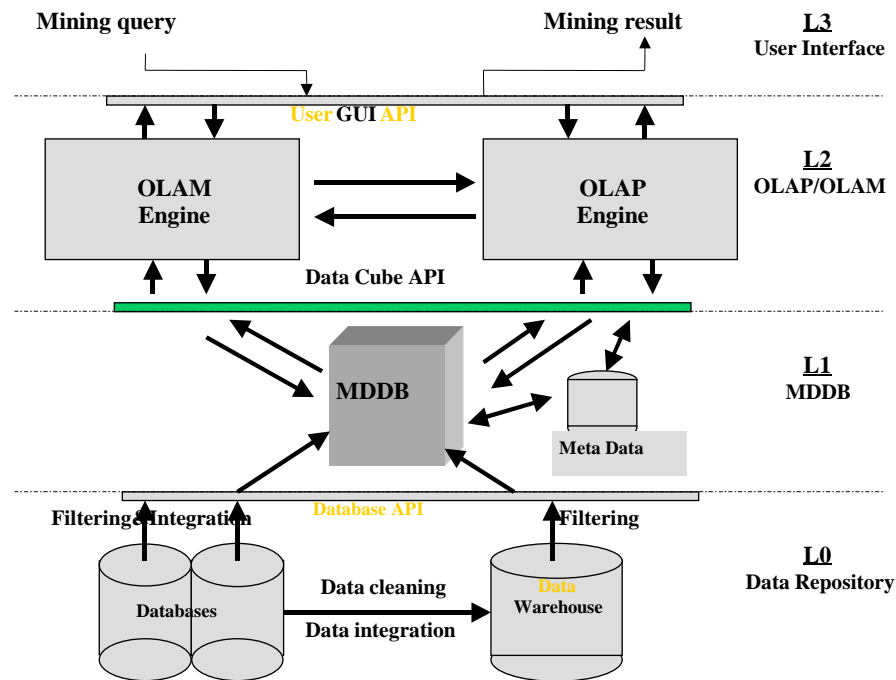
- Brachman和Anand从用户的角度对KDD处理过程进行了分析。他们认为数据库中的知识发现应该更着重于对用户进行知识发现的整个过程的支持，而不是仅仅限于在数据挖掘的一个阶段上。该模型强调对**用户与数据库交互**的支持。





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。





-
- The diagram illustrates the architecture of a data mining system. On the left, a 'User' box interacts with a sequence of six yellow process boxes: '问题定义' (Problem Definition), '数据抽取' (Data Extraction), '模式选择' (Model Selection), '数据预处理' (Data Preprocessing), '数据挖掘' (Data Mining), and '模式评估' (Model Evaluation). These processes are connected by upward arrows. On the right, data sources are categorized into four levels: '源数据' (Source Data) includes 'Web/TEXT' (blue box); '备选数据' (Candidate Data) includes 'DB/DW' (grey cylinder); '目标数据' (Target Data) includes 'DB' (grey cylinder) and 'Cube' (grey box); and '知识' (Knowledge) includes '知识库' (Knowledge Base) and '模式库' (Model Base) (grey cylinders). A vertical dashed line separates the processes from the data sources. Arrows show data flow: 'Web/TEXT' feeds into 'DB/DW'; 'DB' and 'DW' (small cylinders) also feed into 'DB/DW'; 'DB/DW' feeds into 'DB' and 'Cube'; 'DB' and 'Cube' feed into '数据挖掘'; '知识库' and '模式库' feed into '模式评估'. A feedback loop arrow goes from '模式评估' back to '用户'.

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

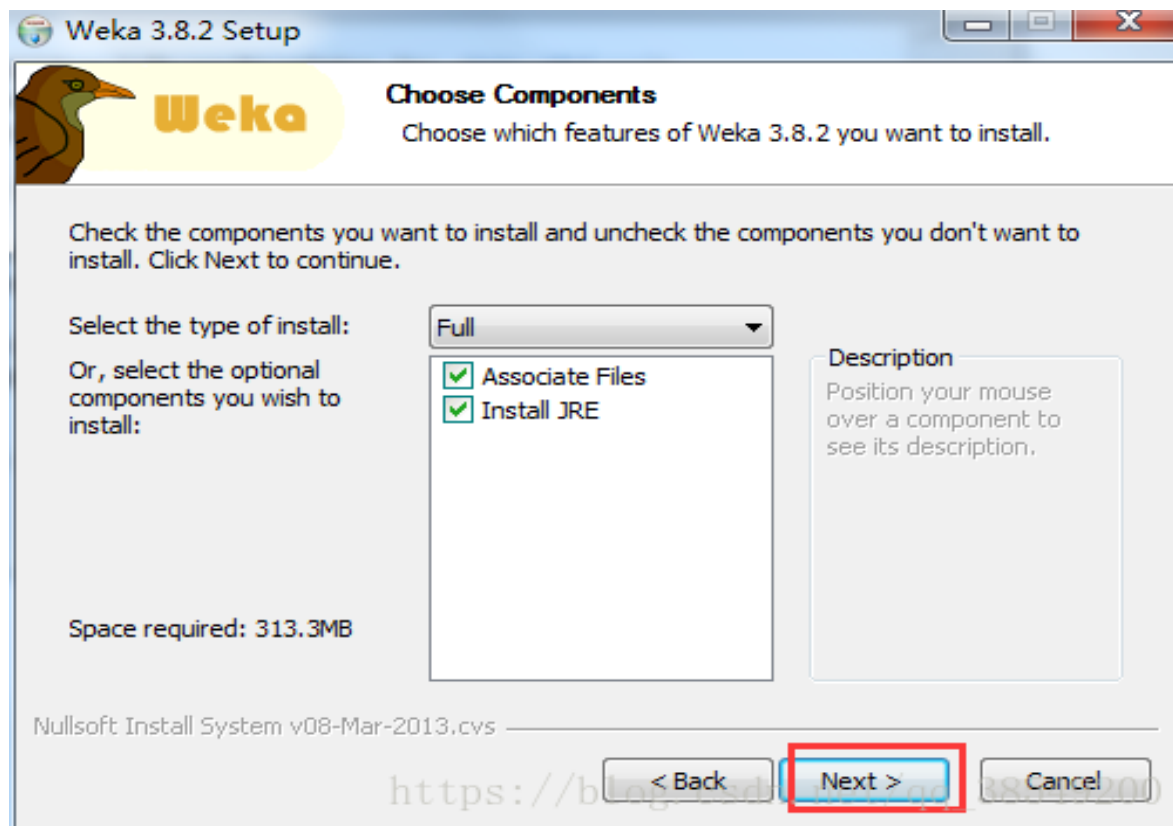




Weka的安装

- WEKA的官方地址是

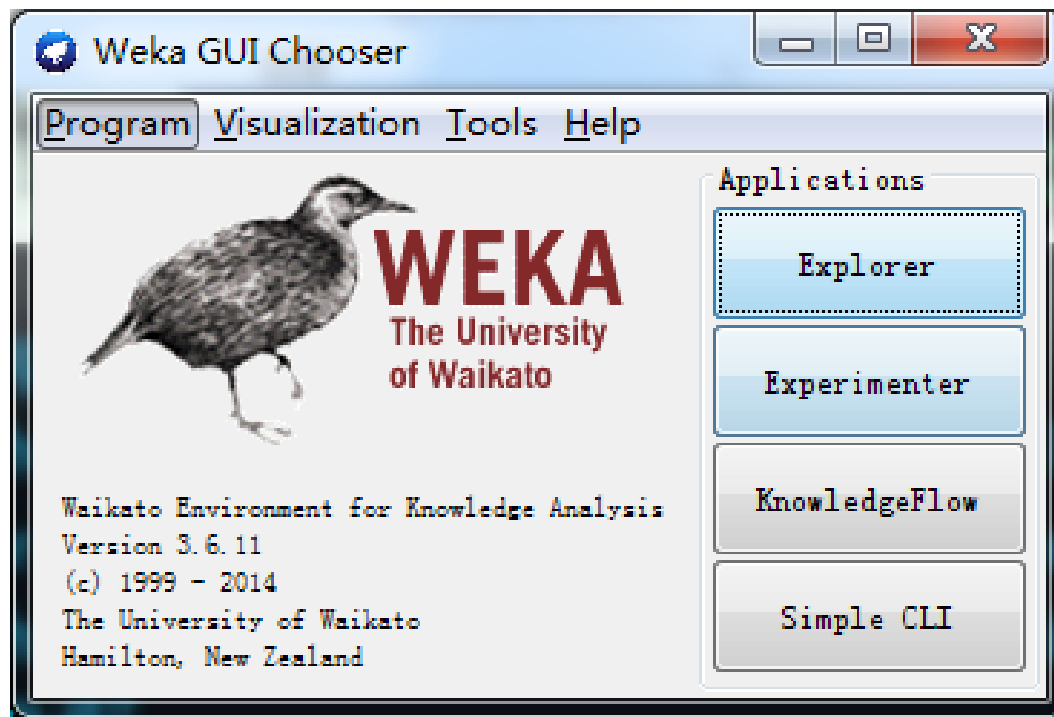
<http://www.cs.waikato.ac.nz/ml/weka/downloading.html>。里面有windows, mas os, linux等平台下的版本。





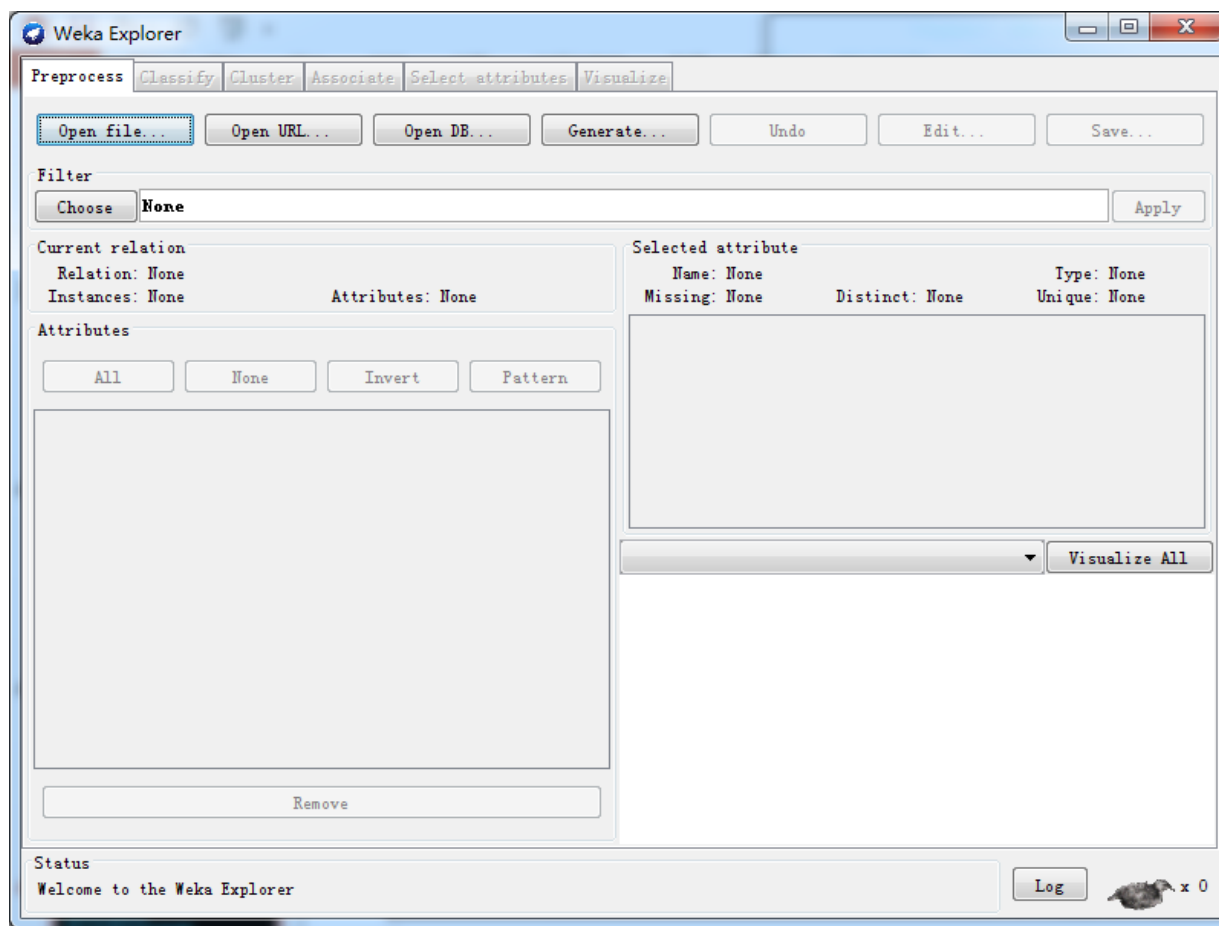
实验目的

- 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%)

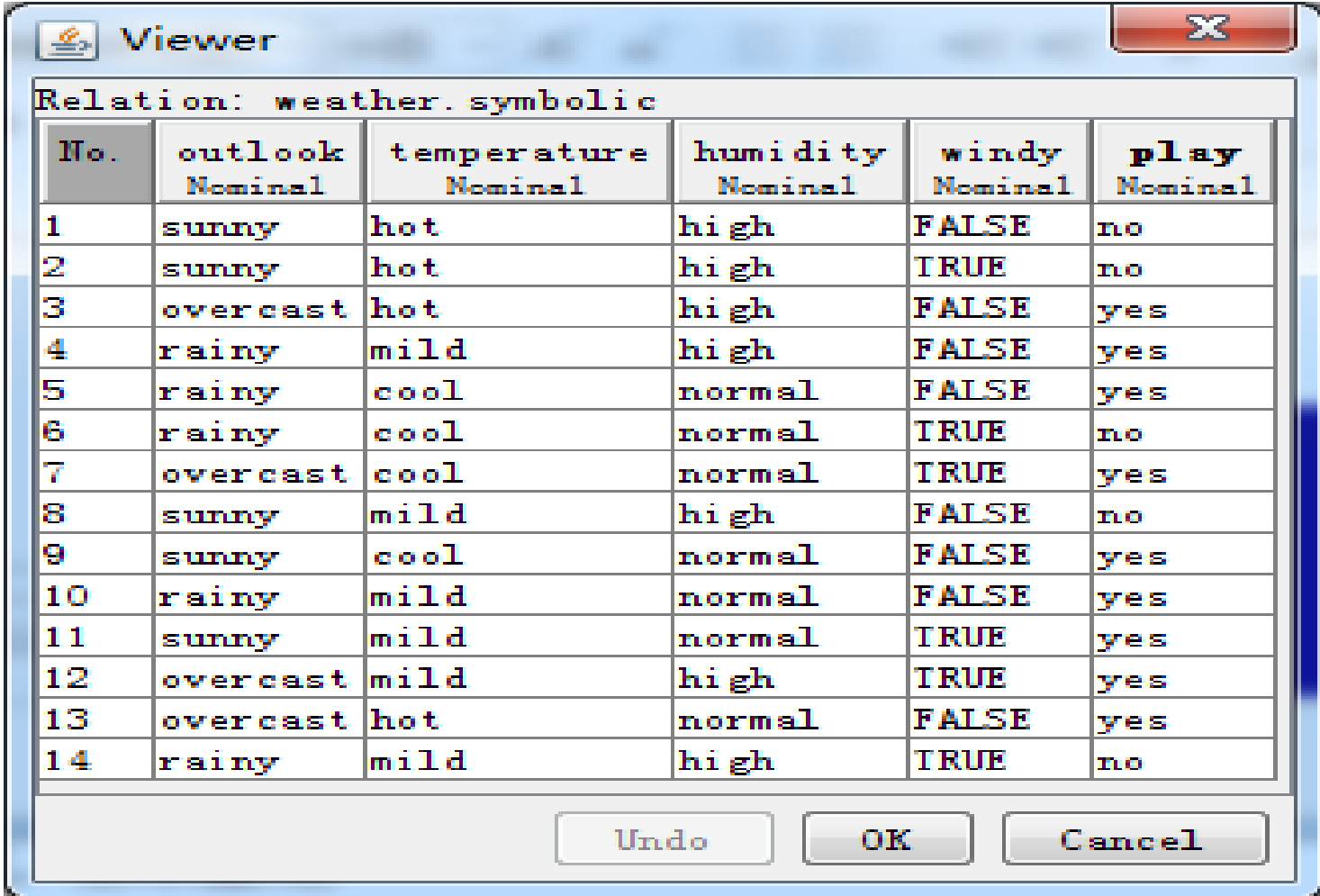
No.	Label	Count
1	sunny	5
2	overcast	4
3	rainy	5

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 ===

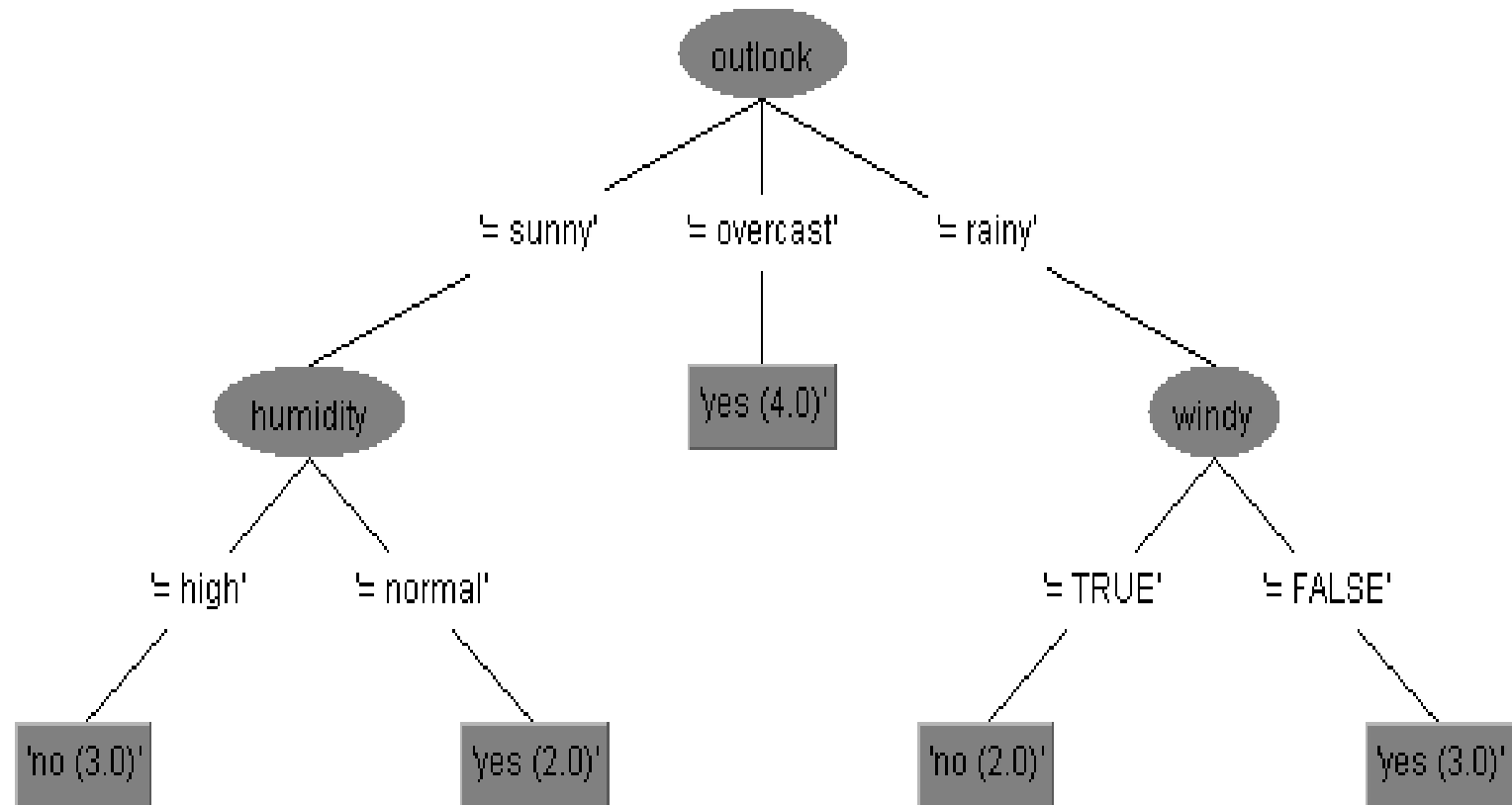
a b <-- classified as
5 4 | a = yes
3 2 | b = no

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



例如

=== 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





数据挖掘语言概述

- 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。



数据挖掘查询语言

- 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

- 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

- 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

`<Mine_Knowledge_Specification> ::= <Mine_Char> | <Mine_Discr> | <Mine_Assoc> | <Mine_Class>`

`<Mine_Char> ::= mine characteristics [as <pattern_name>] analyze
 <measure(s)>`

`<Mine_Discr> ::= mine comparison [as <pattern_name>]
 for <target_class> where <target_condition>
 {versus <contrast_class_i> where
 <contrast_condition_i>}
 analyze <measure(s)>`

`<Mine_Assoc> ::= mine associations [as <pattern_name>]
 [matching <metapattern>]`

`<Mine_Class> ::= mine classification [as <pattern_name>]
 analyze <classifying_attribute_or_dimension>`



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

