数据挖掘与机器学习

潘斌
panbin@nankai.edu.cn
范孙楼227

上节回顾

- 什么是数据挖掘
  - 数据挖掘是在大型数据存储库中，自动的发现有用信息的过程
- 什么是机器学习
  - 可自动发现有用信息的手段即为机器学习算法
- 什么是大数据
  - 大数据具有**4V**特征
- 数据的特点
  - 数据属性

本节提要

- 数据的特点
  - 数据集的一般特性
  - 数据质量
  - 数据预处理
- 特征学习
  - 特征提取
  - 特征选择
- 概念学习
  - 总体、目标、样本、假设

数据探索（DATA EXPLORATION）

- 拿到数据的第一件事
- 对数据初步研究，以更好理解其特殊性质
  - 有助于选择合适的数据预处理技术和数据分析技术
  - 可以处理一些通常由数据挖掘解决的问题
    - 如，特征的设计
  - 理解和解释数据挖掘的结果

鸢尾花（IRIS）数据集

- 包含150种鸢尾花信息
- 取自三个物种
  - 山鸢尾（Setosa）；维吉尼亚鸢尾（Virginica）；变色鸢尾（Versicolour）
- 特征用五种属性描述
  - 萼片长度（cm）；萼片宽度（cm）；花瓣长度（cm）；花瓣宽度（cm）；类（属种）

萼片长度	萼片宽度	花瓣长度	花瓣宽度	类
5.1	3.5	1.4	0.2	*setosa*
4.9	3	1.4	0.2	*setosa*
……				
5.7	2.9	4.2	1.3	*versicolor*
5.7	2.8	4.1	1.3	*versicolor*
……				
5.8	2.7	5.1	1.9	*virginica*
7.1	3	5.9	2.1	*virginica*
……				

鸢尾花数据集（部分）

汇总统计（SUMMARY STATISTICS）

- 用单个数或数的小集合捕获可能很大的值集的各种特征
- 频率：给定一个在$\{v_1,v_2,....,v_k\}$取值的分类属性x和m个对象的集合，值v_i的频率定义为 $\text{frequency}(v_i) = \frac{\text{具有属性值}v_i\text{的对象数}}{m}$
- 众数：具有最高频率的值

一所假想大学中各年级学生人数

年级	人数	频率
一年级	200	0.33
二年级	160	0.27
三年级	130	0.22
四年级	110	0.18

则年级属性的众数为“一年级”。

汇总统计

- 众数（统计量）的分辨率问题
  - 对于连续属性，按照目前的定义，众数通常没有用。
  - 以毫米为单位，20个人的身高通常不会重复，但如果以分米为单位，则某些人很可能具有相同的身高。
  - 众数可以用来估计缺失值。

汇总统计

- 百分位数
  - xp：对应一个x值，使得x的p%观测值小于xp

萼片长度、萼片宽度、花瓣长度和花瓣宽度的百分位数（所有的值都以厘米为单位）

百分位数	萼片长度	萼片宽度	花瓣长度	花瓣宽度
0	4.3	2.0	1.0	0.1
10	4.8	2.5	1.4	0.2
20	5.0	2.7	1.5	0.2
30	5.2	2.8	1.7	0.4
40	5.6	3.0	3.9	1.2
50	5.8	3.0	4.4	1.3
60	6.1	3.1	4.6	1.5
70	6.3	3.2	5.0	1.8
80	6.6	3.4	5.4	1.9
90	6.9	3.6	5.8	2.2
100	7.9	4.4	6.9	2.5

汇总统计

- 位置度量：均值和中位数

$$\text{mean}(x) = \overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

- $$\text{median}(x) = \begin{cases} x_{(r+1)} & \text{if } m \text{ is odd, i.e., } m = 2r+1 \\ \frac{1}{2}(x_{(r)} + x_{(r+1)}) & \text{if } m \text{ is even, i.e., } m = 2r \end{cases}$$

萼片长度、萼片宽度、花瓣长度和花瓣宽度的均值、中位数和截断均值（所有值都以厘米为单位）

度量	萼片长度	萼片宽度	花瓣长度	花瓣宽度
均值	5.84	3.05	3.76	1.20
中位数	5.80	3.00	4.35	1.30
截断均值（20%）	5.79	3.02	3.72	1.12

汇总统计

- 散布度量：极差和方差
  - 极差： $\text{range}(x) = \max(x) - \min(x) = x_{(m)} - x_{(1)}$
  - 方差： $\text{variance}(x) = s_x^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \overline{x})^2$
  - 绝对平均偏差：（absolute average deviation）

    $$\text{AAD}(x) = \frac{1}{m}\sum_{i=1}^{m}|x_i - \overline{x}|$$

  - 中位数绝对偏差（median absolute deviation）

    $$\text{MAD}(x) = median\Big(\{|x_1 - \overline{x}|, \ldots, |x_m - \overline{x}|\}\Big)$$

  - 四分位数极差（interquartile range）

    $$\text{interquartile range}(x) = x_{75\%} - x_{25\%}$$

汇总统计

萼片长度、萼片宽度、花瓣长度和花瓣宽度的极差、标准差（std）、绝对平均偏差（AAD）、中位绝对偏差（MAD）和中间四分位数极差（IQR）（所有值都以厘米为单位）

度量	萼片长度	萼片宽度	花瓣长度	花瓣宽度
极差	3.6	2.4	5.9	2.4
std	0.8	0.4	1.8	0.8
AAD	0.7	0.3	1.6	0.6
MAD	0.7	0.3	1.2	0.7
IQR	1.3	0.5	3.5	1.5

数据集的一般特性

- 维度（Dimensionality）
  - 数据集中的对象具有的属性数目
  - 维数灾难（Curse of Dimensionality）
- 稀疏性（Sparsity）
  - 一个对象的大部分属性上的值都为0
- 分辨率（Resolution）
  - 不同分辨率下数据的性质不同
  - 模式依赖于分辨率水平

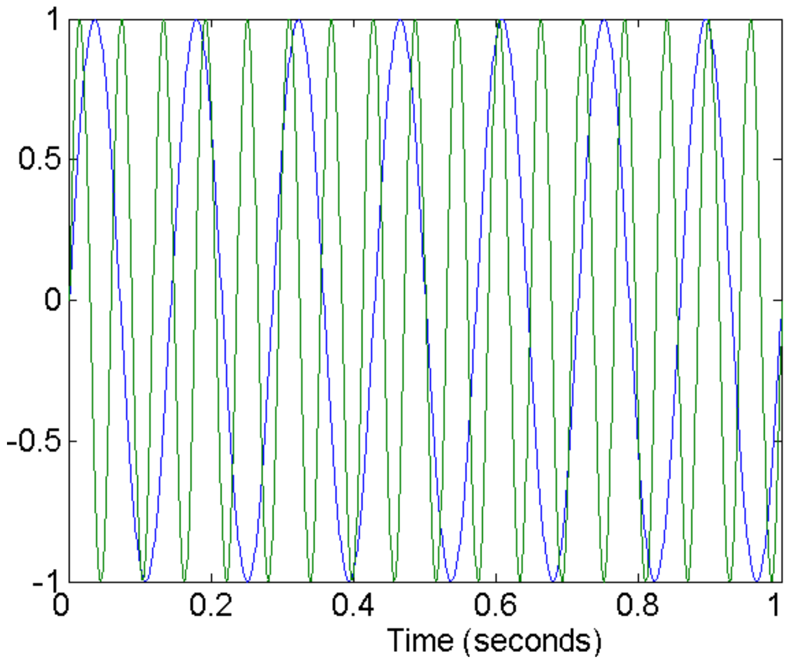
数据质量

- 数据并非完美
  - 人为错误
  - 设备限制
  - 搜集漏洞
- 无法避免
- 两个方面
  - 数据质量问题的检测和纠正（预处理）
  - 使用可以容忍低质量数据的算法（鲁棒）

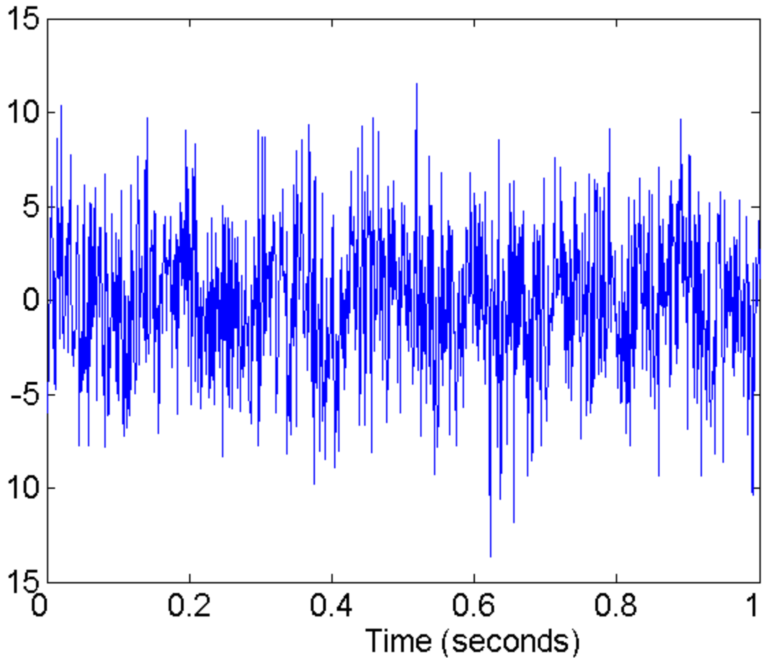
数据质量

- 噪声：测量误差的随机部分
  - 可通过使用信号或图像处理技术降低噪声，很难完全消除
  - Robust algorithm 的使用能产生可以接受的结果
  - 例：

数据质量



Two Sine Waves

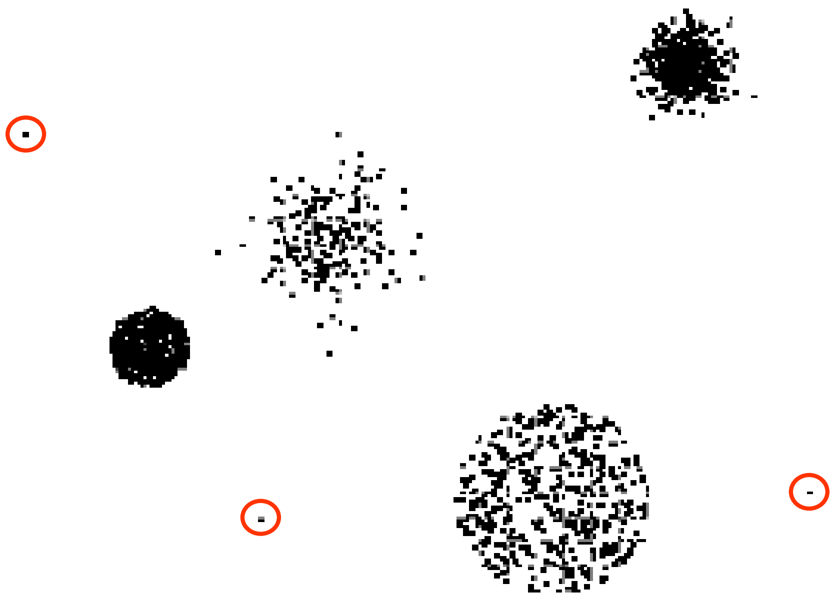


Two Sine Waves + Noise

数据质量

- 离群点（Outlier）
  - 在某种意义上具有不同于数据集中其他大部分数据对象的特征，或相对于该属性典型值来说不寻常的特征
  - 异常（anomalous）对象，异常值
  - 可以是合法的数据对象或值
  - 有时是感兴趣的对象

数据质量



数据质量

- 缺失值（Missing Values）
  - 信息搜集不全
  - 某些属性并不能用于所有对象
- 处理策略
  - 删除数据对象或属性
  - 估计缺失值
  - 忽略缺失值
  - 用所有可能值代替（可能性为权重）

数据质量

- 重复数据（Duplicate Data）
  - 数据集可能包含重复或几乎重复的数据
  - 如，重复邮件
- 不一致的值
  - 如，地址字段列出了邮政编码和城市名，但有的邮政编码区域并不包含在对应的城市中
  - 负数的身高

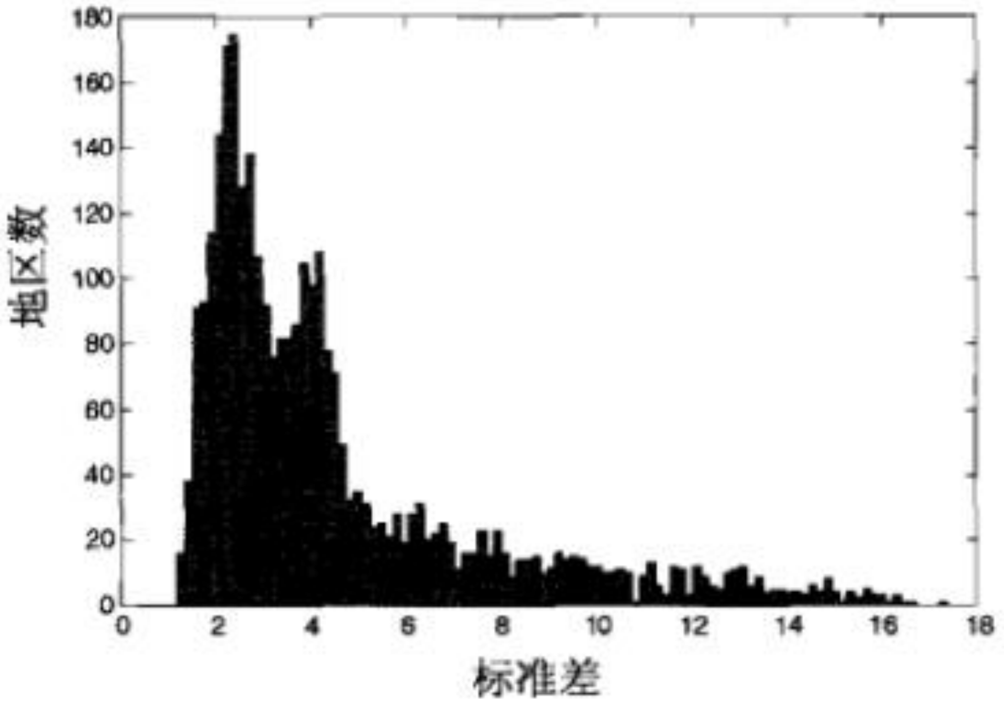
数据预处理

- 聚集（Aggregation）
- 抽样（Sampling）
- 维归约（Dimensionality Reduction）
- 特征子集选择（Feature subset selection）
- 特征创建（Feature creation）
- 离散化和二元化（Discretization and Binarization）
- 属性变换（Attribute Transformation）

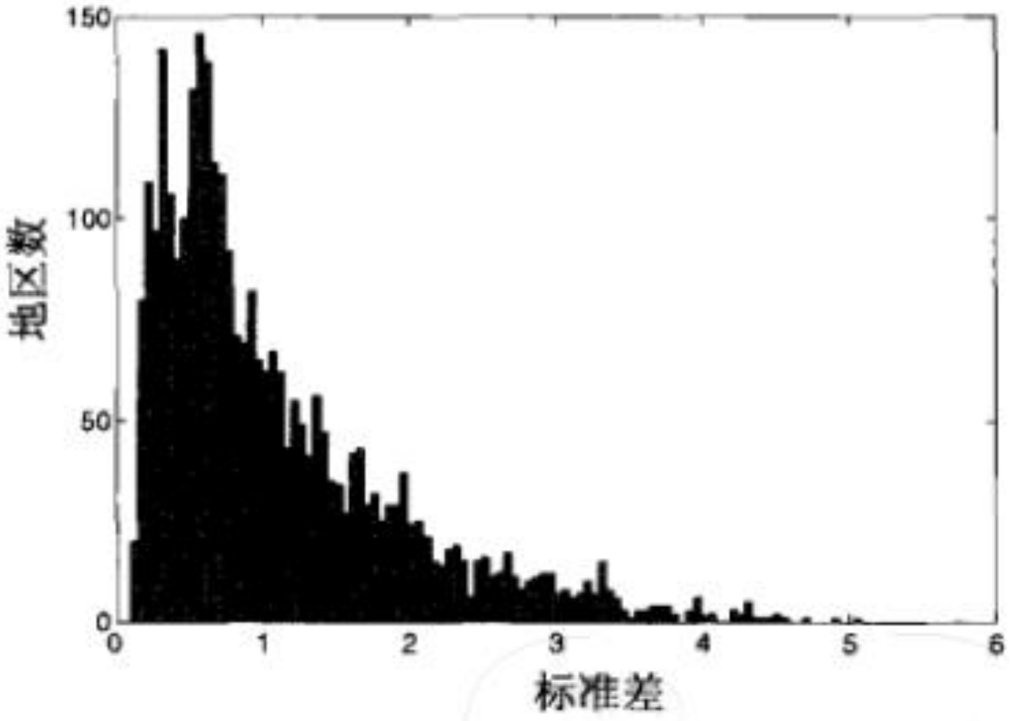
数据预处理

- 聚集
  - 将两个或多个对象合并为单个对象
  - 范围或标度的转换
    - 城市聚集为地区、州、国家等
    - 数据由按天记录聚集为按月记录
  - 对象或属性群的行为通常比单个对象或属性的行为更加稳定（stable）
    - 平均值、总数等的变异性（variability）较小
    - 可能丢失有趣的细节

- 澳大利亚降水量（1982-1993）



(a) 平均月降水量标准差的直方图

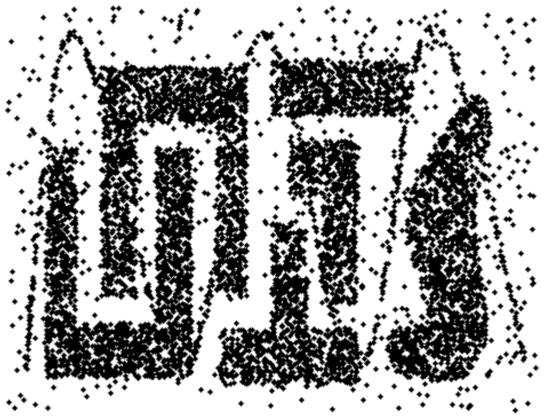


(b) 平均年降水量标准差的直方图

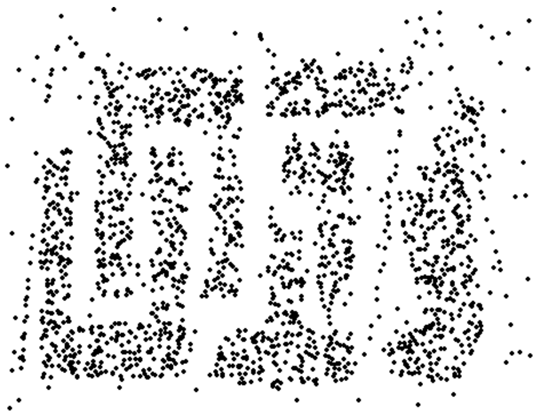
数据预处理

- 抽样
  - 选择数据对象子集进行分析的常用方法
    - 统计学中常用
  - 二者抽样动机不同
    - 统计学：得到感兴趣的整个数据集费用太高、太费时间
    - 数据挖掘：处理所有数据的费用太高、太费时间

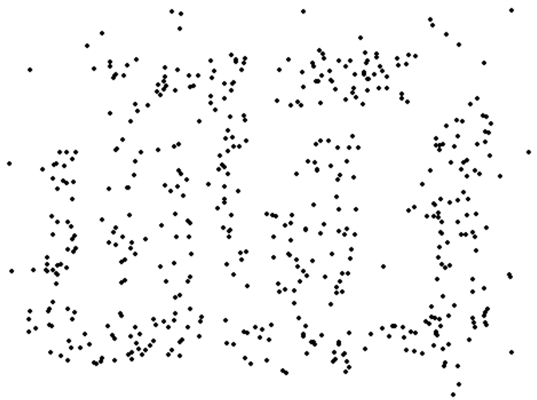
抽样与信息损失



8000 points



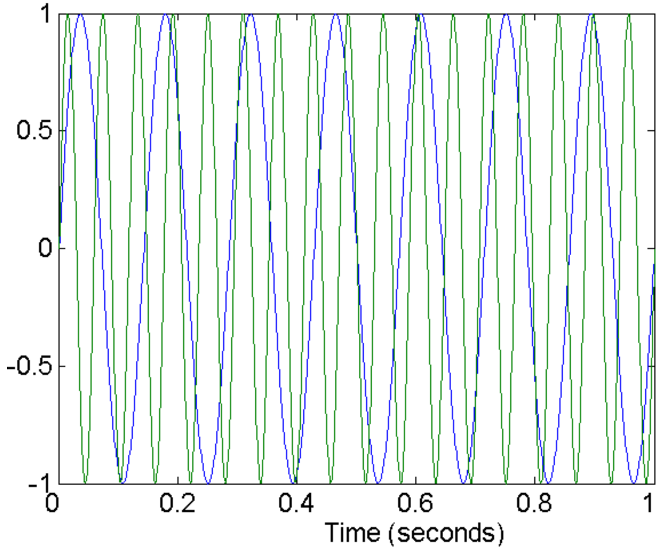
2000 Points



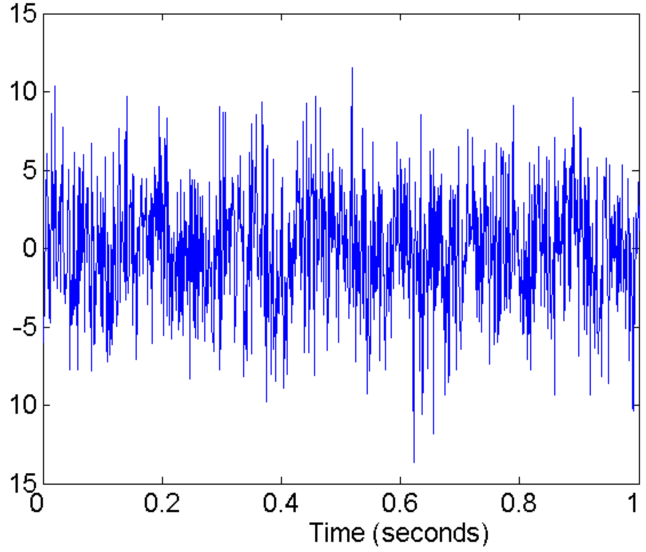
500 Points

数据预处理

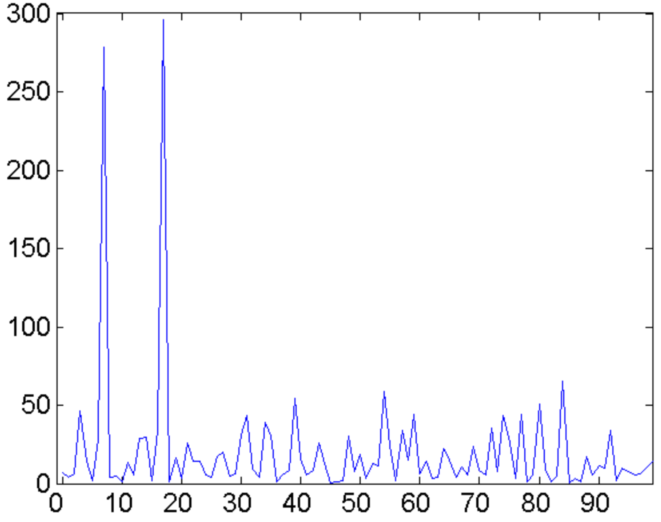
- 特征创建
  - 由原来的属性创建新的属性集，更有效的捕获数据集中的重要信息
  - 常用方法
    - 特征提取（Feature Extraction）
      - 针对具体领域，如图像处理
    - 特征构造（ Feature Construction）
      - 常用专家意见构造特征
    - 映射到新的空间
      - 如傅立叶变换检测时间序列数据的周期模式



Two Sine Waves



Two Sine Waves + Noise



Frequency

数据预处理

- 离散化和二元化
  - 将连续属性变换成分类属性（离散化）
  - 离散和连续属性可能都需要变换成一个或多个二元属性
  - 二元化方法
    - 如有m个分类值，将每个原始值唯一地赋予区间[0,m-1]中的一个整数；如属性是有序的，则赋值必须保持序关系；将这m个整数的每一个都变成一个二进制数
    - 需要$n=[\log_2 m]$个二进位表示这些整数，因此需要n个二元属性表示这些二进制数

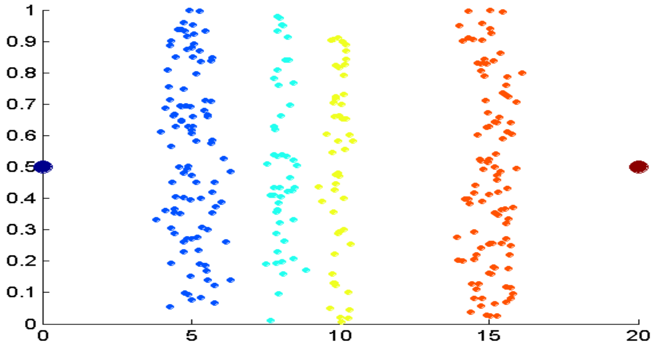
数据预处理

- 例，具有五个值{awful,poor,ok,good,great}的分类变量需要三个二元变量
- One-hot

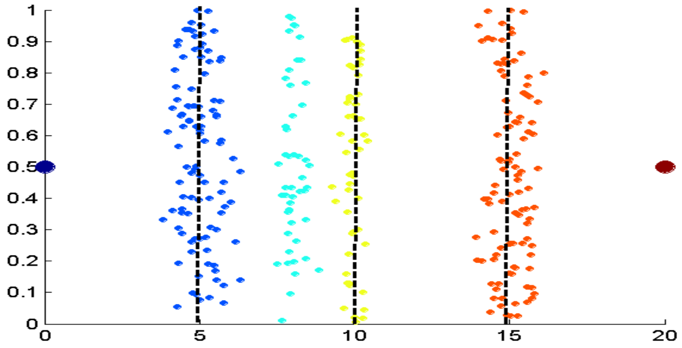
分类值	整数值	x_1	x_2	x_3
awful	0	0	0	0
poor	1	0	0	1
OK	2	0	1	0
good	3	0	1	1
great	4	1	0	0

数据预处理

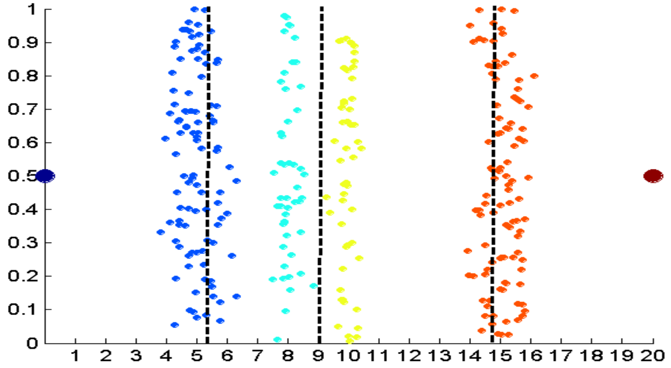
- 离散化方法（非监督）



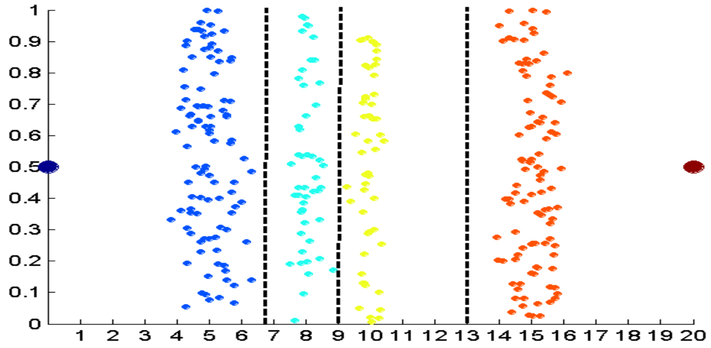
原始数据



等宽离散化



等频率离散化



K-means离散化

数据预处理

- 离散化方法（监督）
  - 通常能够产生更好的结果
  - 基于熵的方法
    - 将初始值切分成两部分（候选分割点可以是每个值），让两个结果区间产生最小的熵
    - 取具有最大熵的区间继续分割
    - 重复此分割过程，直到满足终止条件

数据预处理

设 k 是不同的类标号数，m_i 是某划分的第 i 个区间中值的个数，而 m_{ij} 是区间 i 中类 j 的值的个数。第 i 个区间的熵 e_i 由如下等式给出

$$e_i = -\sum_{j=1}^{k} p_{ij} \log_2 p_{ij}$$

其中，$p_{ij} = m_{ij}/m_i$ 是第 i 个区间中类 j 的概率（值的比例）。

该划分的总熵 e 是每个区间的熵的加权平均，即

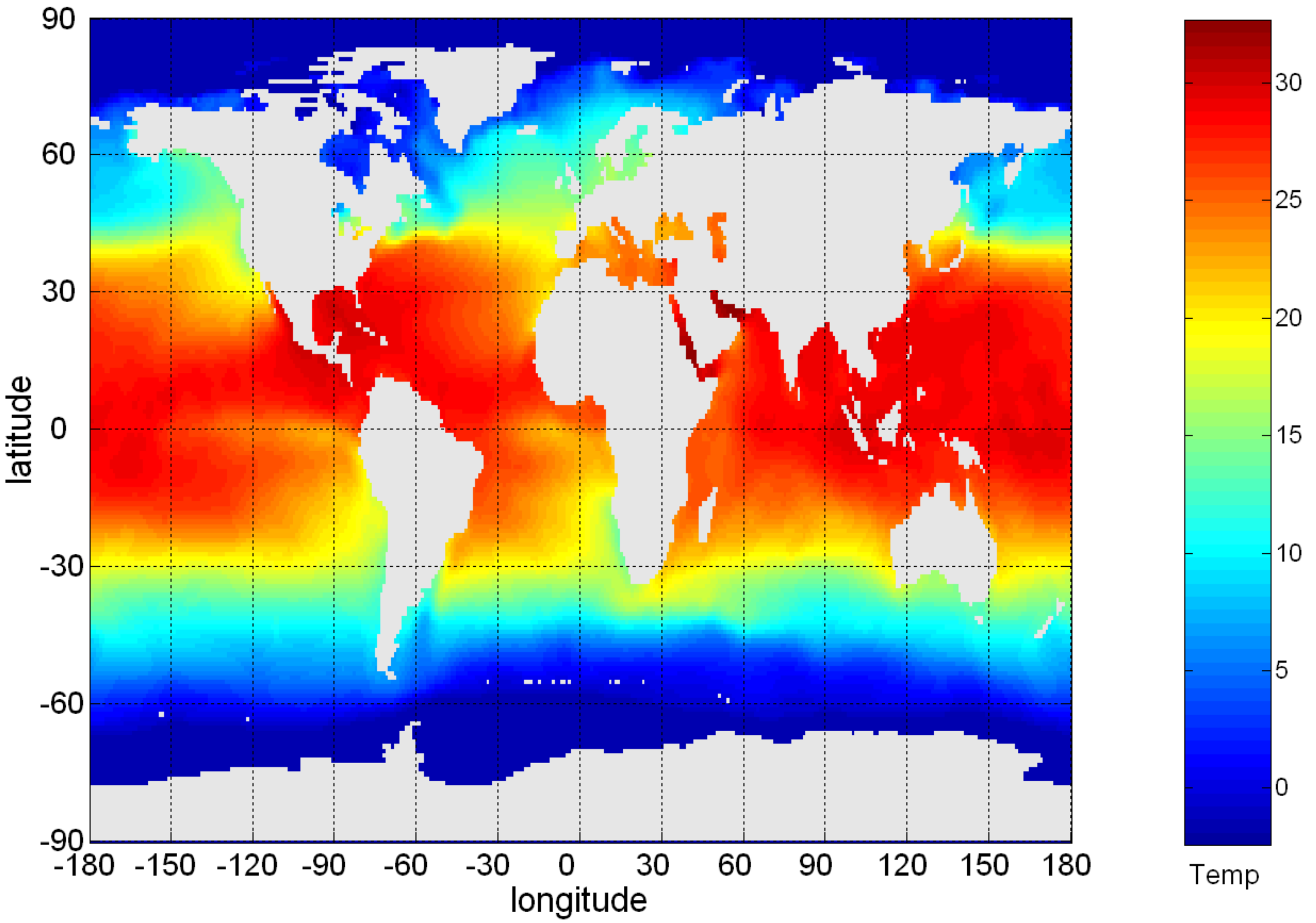
$$e = \sum_{i=1}^{n} w_i e_i$$

其中，m 是值的个数，$w_i = m_i/m$ 是第 i 个区间的值的比例，而 n 是区间个数。

数据预处理

- 变量变换
  - 用于变量的所有值的变换
  - 简单函数
    - x^k, $\log(x)$, e^x, $|x|$, $\sqrt{x}$, $1/x$, $\sin x$
  - 标准化（Standardization）或规范化（Normalization）或归一化
    - 利用均值和标准差
    - 有时用中位数取代均值，用绝对标准差取代标准差
    - 目的：保持数据分布稳定（如神经网络）；排除数据测度的影响

可视化（VISUALIZATION）



- 颜色
- 标尺

可视化

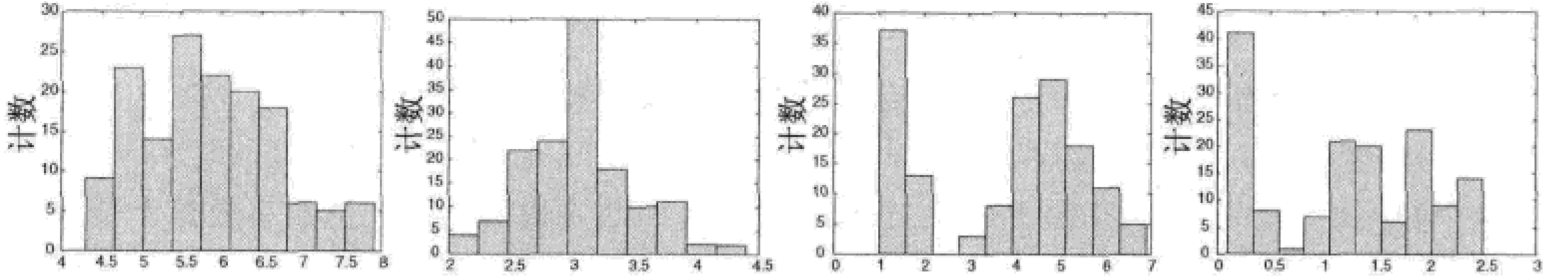
- 重新安排数据的重要性

	1	2	3	4	5	6
1	0	1	0	1	1	0
2	1	0	1	0	0	1
3	0	1	0	1	1	0
4	1	0	1	0	0	1
5	0	1	0	1	1	0
6	1	0	1	0	0	1
7	0	1	0	1	1	0
8	1	0	1	0	0	1
9	0	1	0	1	1	0

	6	1	3	2	5	4
4	1	1	1	0	0	0
2	1	1	1	0	0	0
6	1	1	1	0	0	0
8	1	1	1	0	0	0
5	0	0	0	1	1	1
3	0	0	0	1	1	1
9	0	0	0	1	1	1
1	0	0	0	1	1	1
7	0	0	0	1	1	1

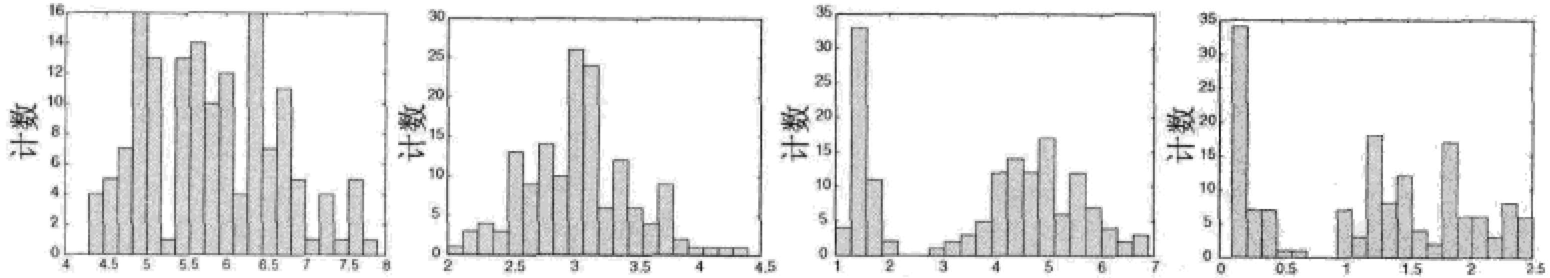
可视化

- 直方图（ Histogram ）



(a) 萼片长度　(b) 萼片宽度　(c) 花瓣长度　(d) 花瓣宽度

四个鸢尾花属性的直方图（10 个箱）

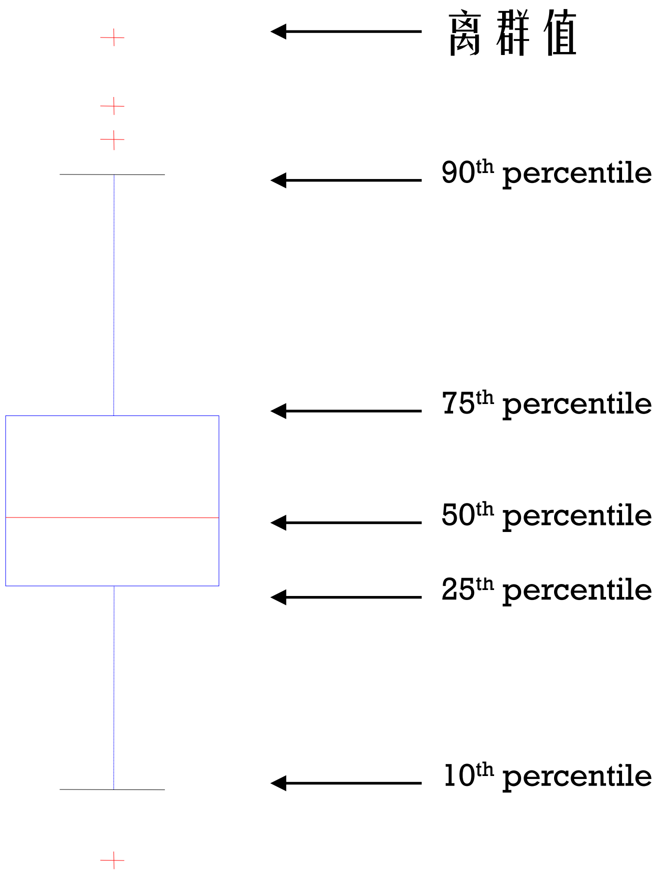


(a) 萼片长度　(b) 萼片宽度　(c) 花瓣长度　(d) 花瓣宽度

四个鸢尾花属性的直方图（20 个箱）

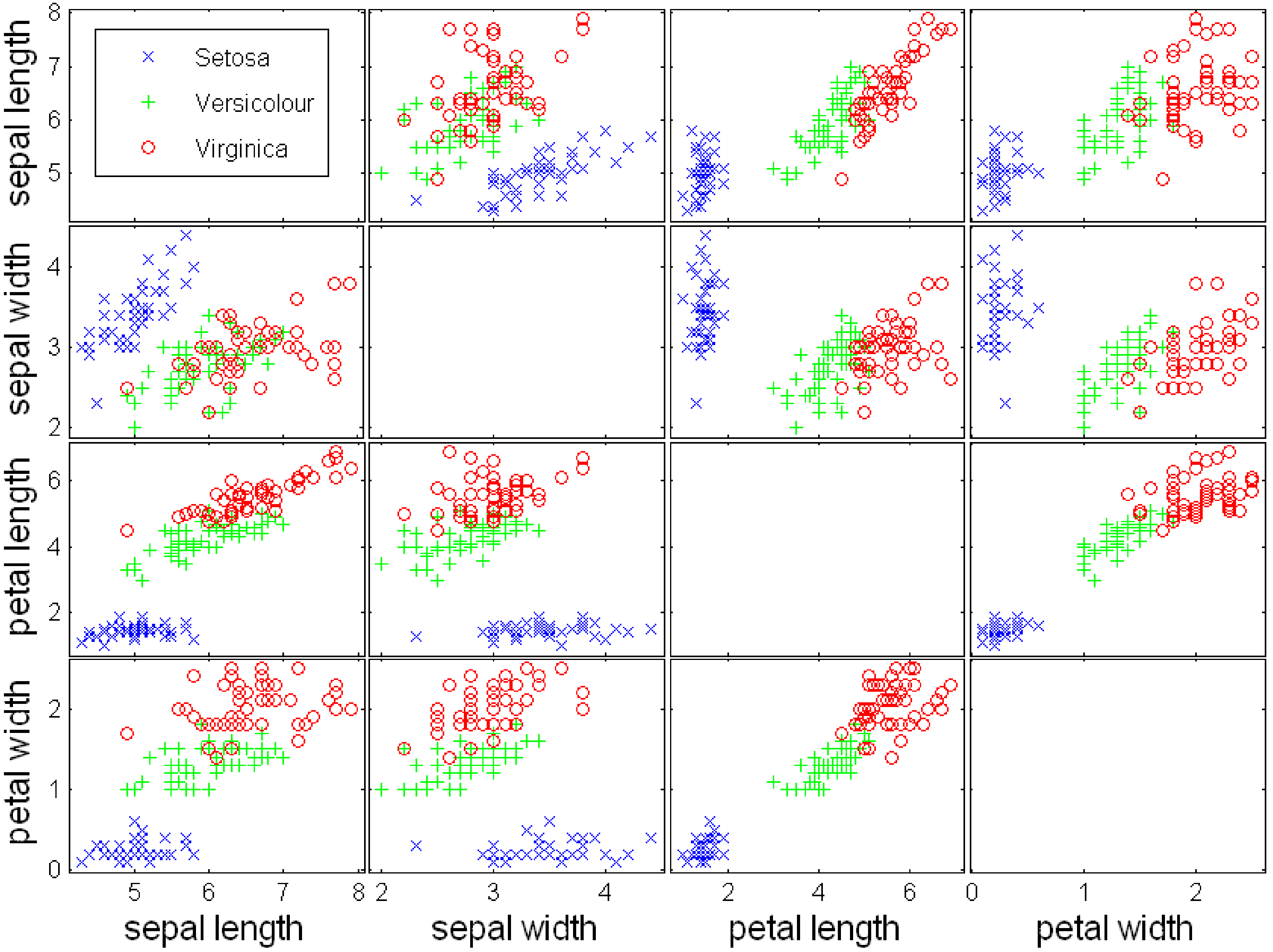
可视化

- 盒状图（box plot）



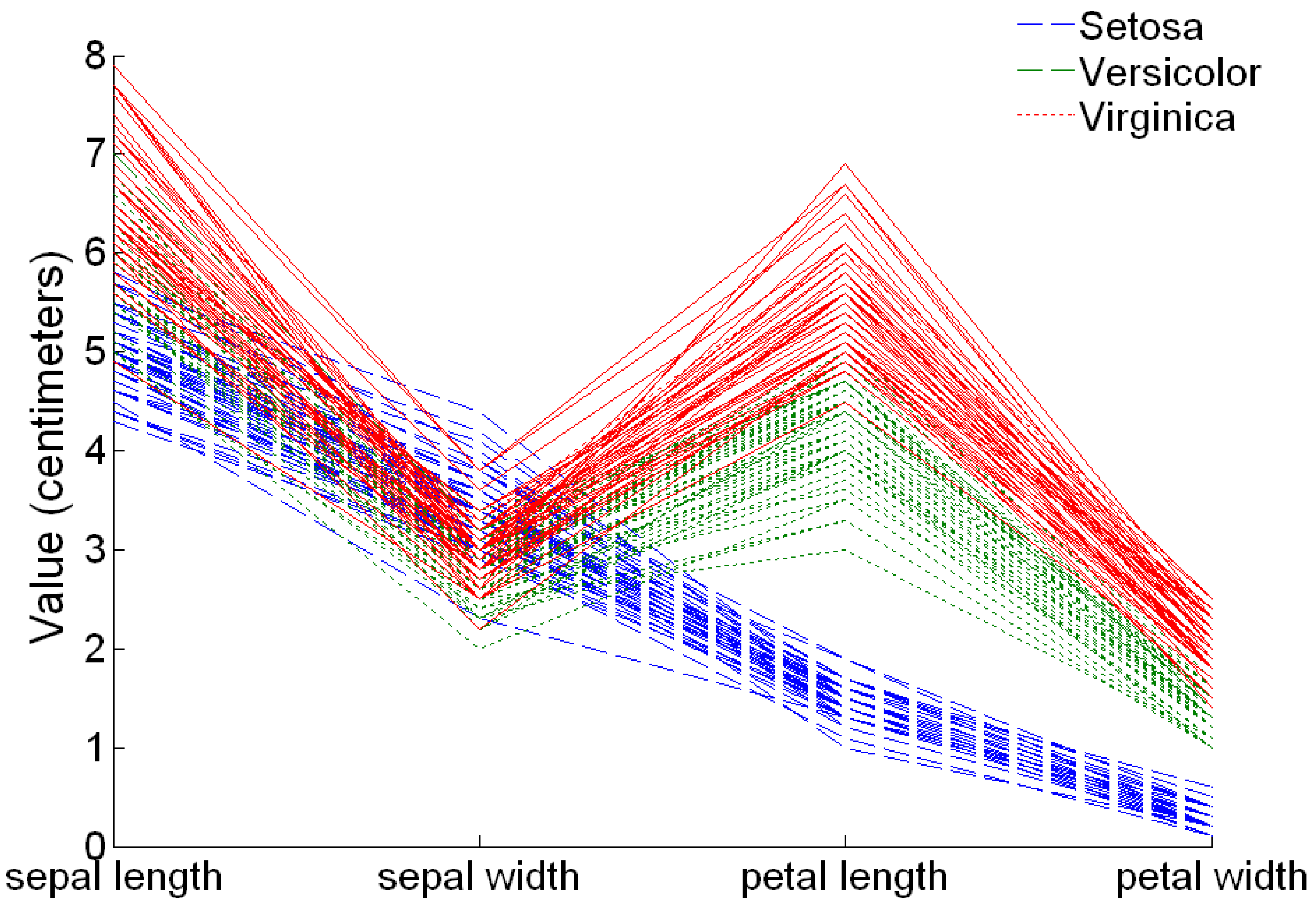
可视化

- 散布图（ Scatter plots ）
  - 图形化显示二属性之间的关系
  - 当类标号给出时，考察二属性将类分开的程度



可视化

- 平行坐标系（ Parallel Coordinates ）



41

特征

The Feature

数据预处理中的特征提取

- 维归约（特征提取）
  - 维度较低时，许多数据挖掘算法的效果会更好
    - 删除不相关的特征，降低噪声，避免维数灾难
    - 降低了算法的时间和内存需求
    - 模型更易理解，容易让数据可视化
  - 维归约常用方法
    - 主成分分析（PCA）
    - 线性判别分析（LDA)

- 特征形成与计算
  - 根据应用领域相关知识决定采用哪些特征，称为原始特征
  - 例如细胞图像大小256 x 256，如果全部采用的话，原始特征即为65536维
  - 如果改为计算细胞的面积、周长、形状、纹理、核浆比，则特征维数变为5维

- 特征提取的原因
  - 机器学习系统的成败，首先取决于所采用的特征是否较好的反映模式的特性以及模式的分类问题
  - 原始特征依赖于具体应用问题和相关专业知识（文字识别和图像识别）
  - 希望在保证分类效果前提下，采用尽可能少的特征完成分类

- 原始特征的问题
  - 有很多特征可能与要解决的分类问题关系不大，但却在后续分类器设计中影响分类器性能
  - 即使很多特征与分类问题关系密切，但特征过多导致计算量大、推广能力差。当样本数有限时容易出现病态矩阵等问题

- 特征提取问题
  - 已知给定的M个原始特征
  - 经过数学变换得到m个特征（$m < M$）

PCA

- 主成分分析
  - Principal Component Analysis (PCA)

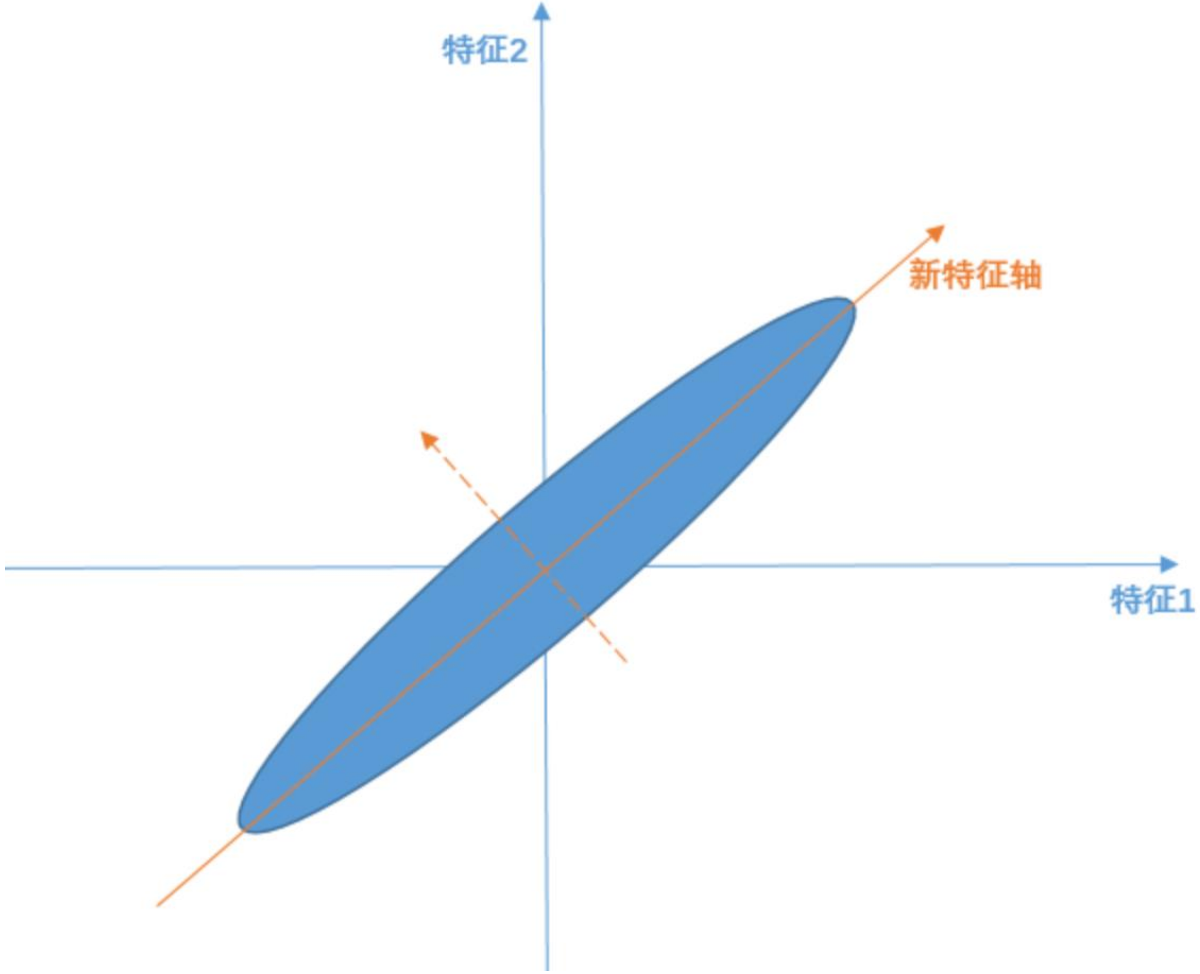
  - 已知给定的M个原始特征
  - 经过线性组合（正交）变换，得到一组按“重要性”从大到小排列的特征
  - 取前m个特征

PCA

- **PCA问题**
  - 设原始特征向量x，维数为M，线性组合的新特征向量y
    - y的各分量 $y_i = \sum_{j=1}^{m} a_{ij} x_j = a_i^T x$
    - 基本约束条件 $a_i^T a_i = 1$
    - 矩阵形式 $y = A^T x$
  - 求最佳的正交变换矩阵A，使得新特征的方差达到极值

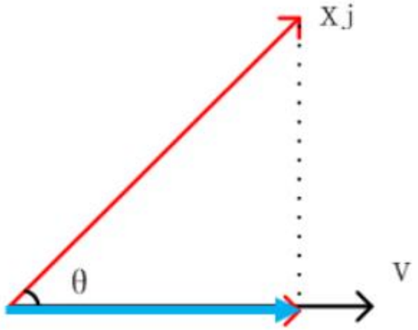
PCA

主轴方向方差大



PCA

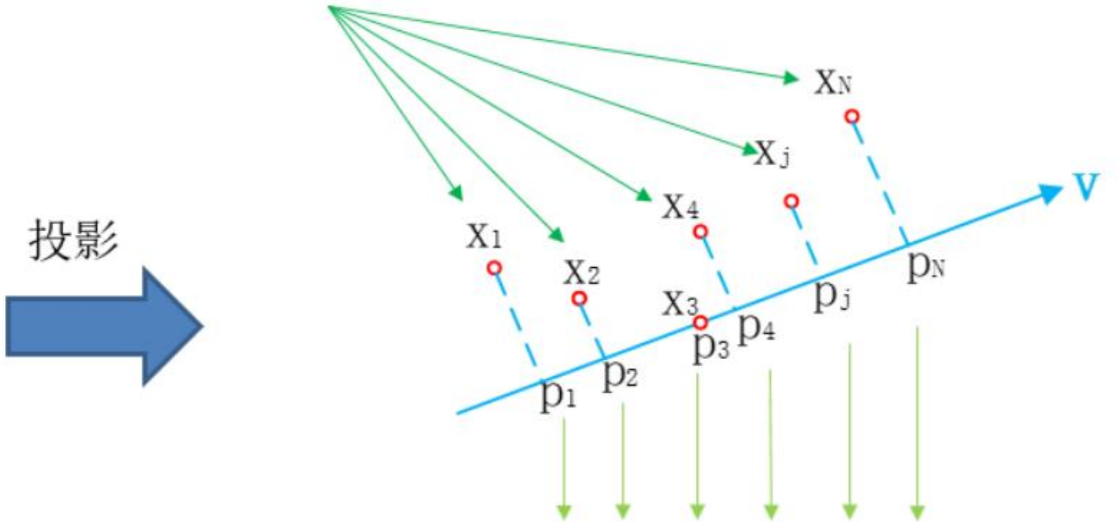
◆ 投影数据



◆xj的投影长度

$$\|x_j\| \cdot \cos\theta = \|x_j\| \cdot \frac{\langle x_j, v\rangle}{\|x_j\| \cdot \|v\|} = \frac{\langle x_j, v\rangle}{\|v\|} = \langle x_j, v\rangle$$

◆样本数据，n维 $x_j = [x_{j1}, x_{j2}, \cdots x_{jn}]^T$

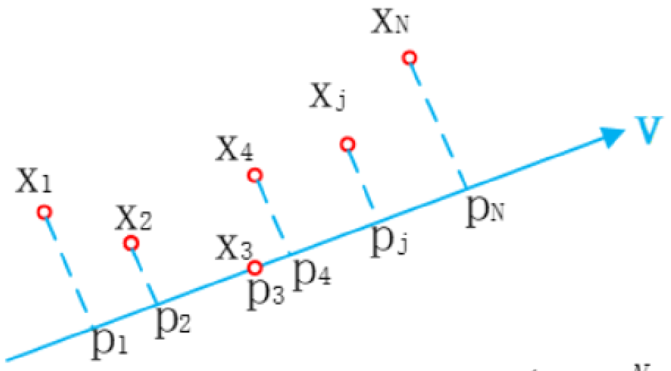


向量内积 → $p_1, p_2, \cdots, p_N$

◆投影数据，N个

PCA

◆ 表征原始数据的信息大小——投影数据的方差



投影数据去除均值

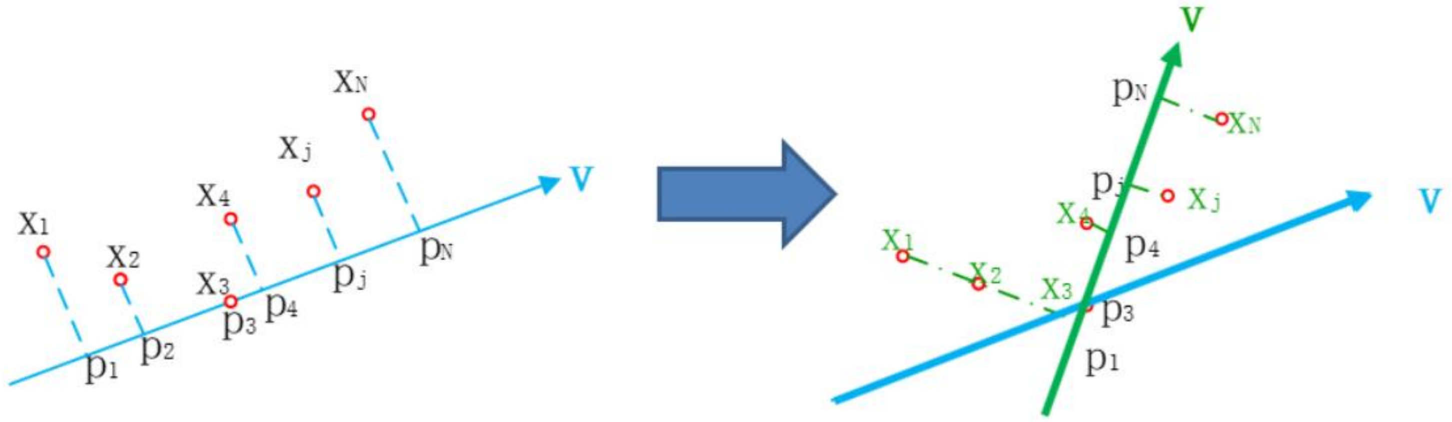
$$\sigma^2 = \frac{1}{N}\cdot\sum_{i=1}^{N}\left(v^T x_j - 0\right)^2 = \frac{1}{N}\cdot\sum_{i=1}^{N}\left(v^T x_j \cdot v^T x_j\right)$$

$$= \frac{1}{N}\cdot\sum_{i=1}^{N}\left(v^T x_j \cdot x_j^T v\right) = v^T \left(\boxed{\frac{1}{N}\cdot\sum_{i=1}^{N}\left(x_j \cdot x_j^T\right)}\right) v$$

$$= \boxed{v^T \left(\boxed{C_x}\right) v}$$

PCA

- 重要：PCA用投影数据的方差来表征原始数据的信息大小——投影数据的方差越大，表明数据分散程度越大，其中包含的信息量越大。
- 考虑到，沿不同的向量v进行投影，将得到不同的投影数据，那么投影数据方差也将发生变化，即其所表征的信息大小也将发生变化。



PCA

- 优化问题建模
- 为了尽可能大地反映样本数据的信息，我们需要确定某个单位向量v，使得样本数据在其上的投影数据具有最大方差。建模如下：

$$\begin{cases} \max\limits_{v} \sigma^2 = \max v^T C_x v \\ s.t. \ \|v\| = 1 \end{cases}$$

PCA

- 优化问题求解
- 利用“拉格朗日方程”求解该最优化：

$$f(v,\lambda) = v^T C_x v - \lambda\left(\|v\| - 1\right) = v^T C_x v - \lambda\left(v^T v - 1\right)$$

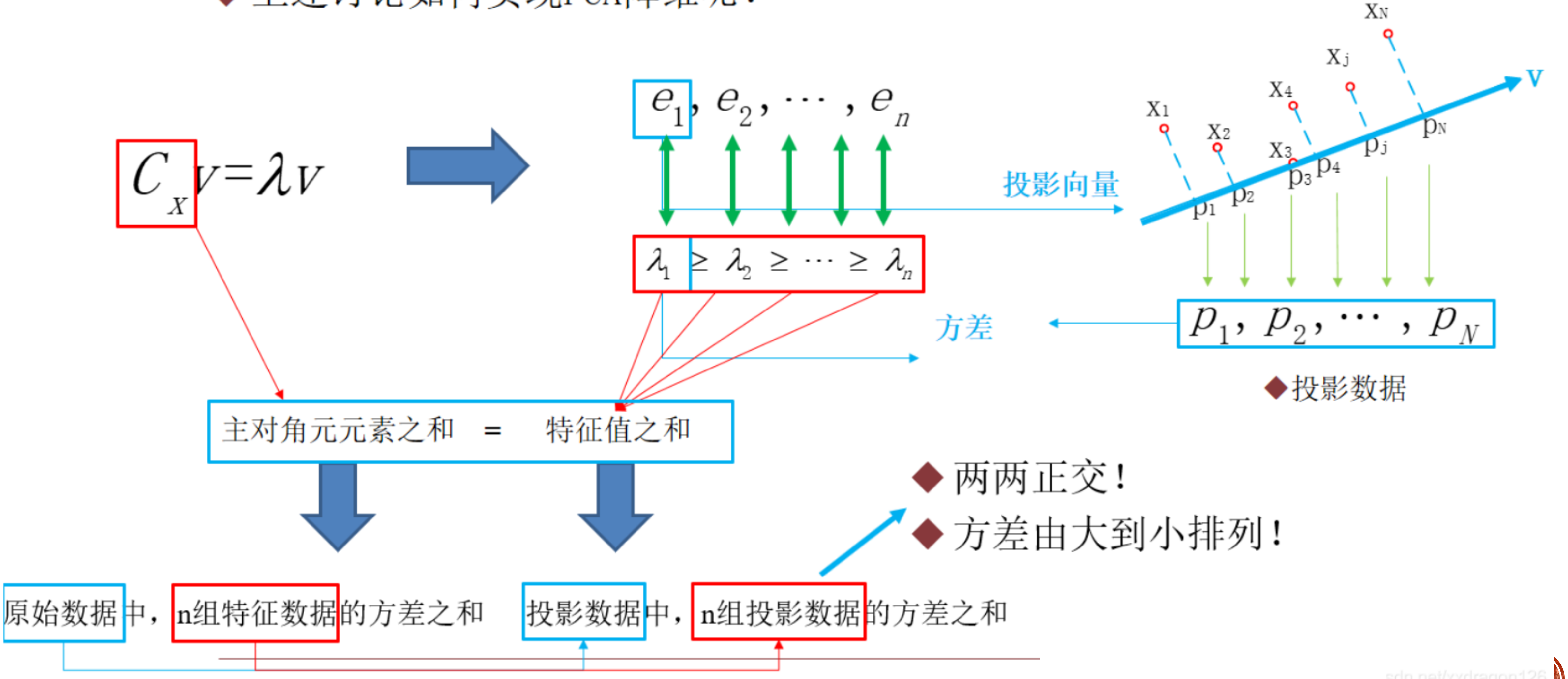
$$\begin{cases} \dfrac{\partial}{\partial v} f(v,\lambda) = 2C_x v - 2\lambda v = 0 \\ \dfrac{\partial}{\partial \lambda} f(v,\lambda) = v^T v - 1 = 0 \end{cases}$$

$$\begin{cases} \boxed{C_x v = \lambda v} \\ v^T v = 1 \end{cases} \Rightarrow$$ v是Cx的特征向量

$$\max \sigma^2 = \max v^T C_x v = \max v^T \lambda v = \max \lambda \Rightarrow$$ 方差是Cx的特征向量

PCA

- 上述讨论如何实现PCA降维呢？

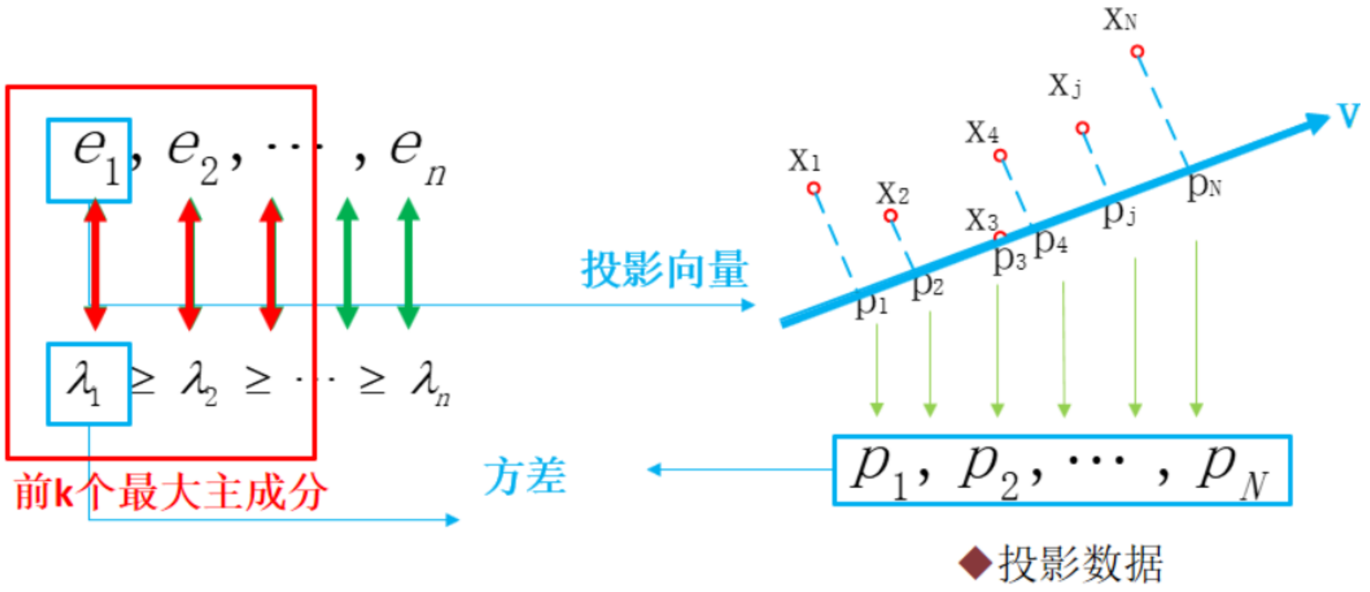


PCA

◆ 小结PCA降维过程

$$r=\frac{\sum_{j=1}^{k}\lambda_j}{\boxed{\sum_{j=1}^{n}\lambda_j}}$$

方差占比（贡献率）



◆ 即，这k组投影数据对原始数据的表征能力为r(称为“贡献率”)。

◆ 特别地，当k取n时，r=1，意味着这k个主成分能够完全表征原始数据。

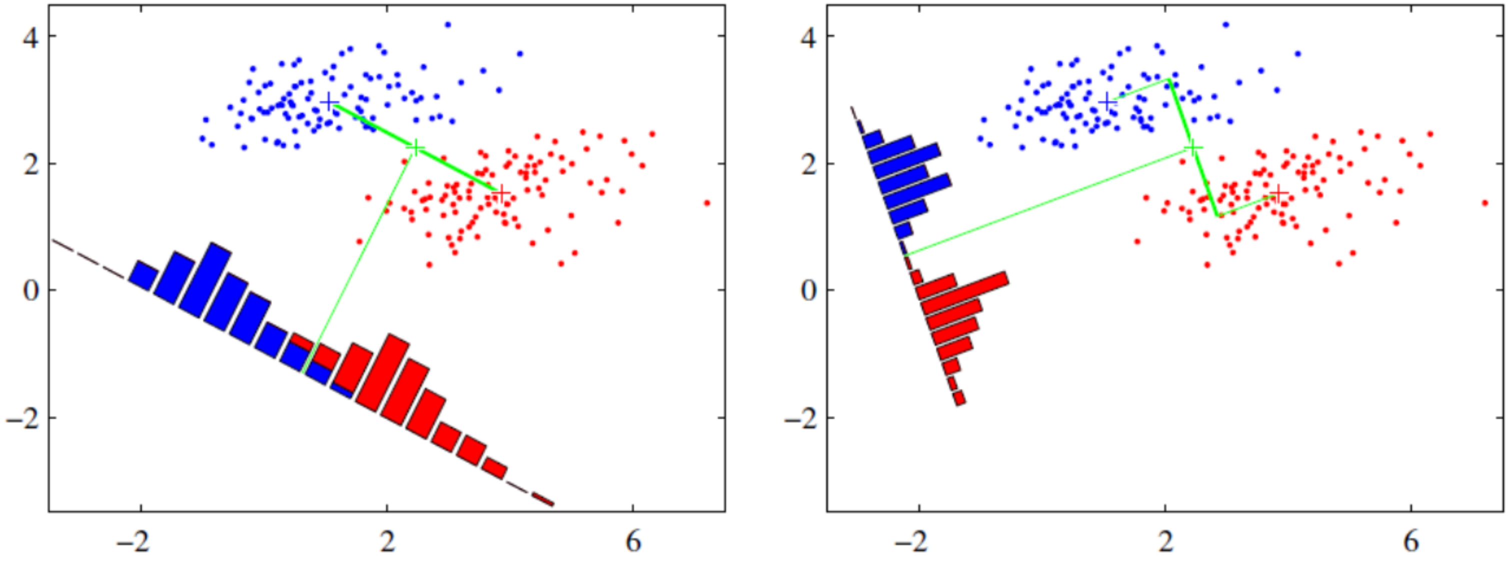
此时即完成了将原始n维数据降维成新的k(k≤n)组特征数据。

LDA

- **Fisher投影准则**
  - 已知给定的M个原始特征
  - 经过数学投影得到1个特征
  - 求最佳投影向量p*

$$\max\ J_F(p) = \frac{p^T S_B p}{p^T S_W p}$$

同类尽可能近，不同类尽可能远

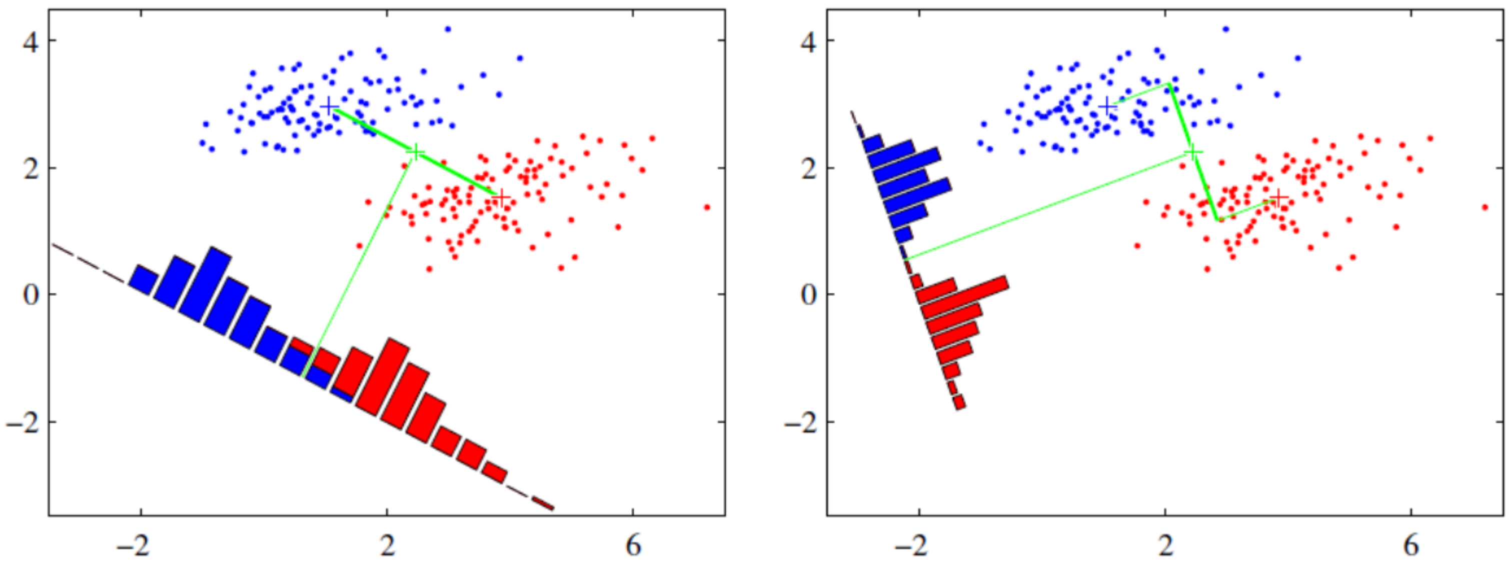


LDA

- **LDA投影准则**
  - 已知给定的M个原始特征
  - 经过数学投影得到m个特征
  - 求最佳投影矩阵P*

$$\max J_F(p) = \frac{p^T S_B p}{p^T S_W p}$$

同类尽可能近，不同类尽可能远



LDA

- 优化准则

$$\mathbf{S}_w = \mathbf{\Sigma}_0 + \mathbf{\Sigma}_1 = \sum_{\boldsymbol{x}\in X_0} (\boldsymbol{x}-\boldsymbol{\mu}_0)(\boldsymbol{x}-\boldsymbol{\mu}_0)^{\mathrm{T}} + \sum_{\boldsymbol{x}\in X_1} (\boldsymbol{x}-\boldsymbol{\mu}_1)(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}$$

$$\mathbf{S}_b = (\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}$$

令分母为1

$$\min_{\boldsymbol{w}} \quad -\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}$$
$$\text{s.t.} \quad \boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w} = 1 .$$

拉格朗日对偶乘子法

数据预处理中的特征选择

- 特征子集选择（特征选择）
  - 降低维度的又一方法
  - 冗余特征（Redundant features ）
    - 重复包含在一个或多个其他属性中的许多或所有信息
      - 一种产品的购买价格和所支付的销售税额
  - 不相关特征（Irrelevant features）
    - 包含对于相关数据挖掘任务几乎完全没用的信息
      - 学生的ID号码对于预测学生总平均成绩

几种常见的子集选择方法

- 暴力（Brute-force ）方法
  - 将所有可能的特征子集作为感兴趣的数据挖掘算法的输入，然后选择产生最好结果的子集
- 过滤（Filter）方法
  - 在数据挖掘算法运行前进行特征选择
- 包装（Wrapper）方法
  - 将数据挖掘算法作为黑盒子找到最好的属性子集，通常并不枚举
- 嵌入（Embedded）方法
  - 特征选择自然的作为数据挖掘算法的一部分，算法本身决定使用哪些属性和忽略哪些属性

- 过滤算法
  - Relief(Kira &Rendell,1992)
    - 设计一个“相关统计量”来度量特征的重要性（如：单特征分类正确率）
      - 一个向量，每个分量对应一个初始特征
      - 特征子集的重要性由相应相关统计量分量之和来决定
    - 选择方法
      - 指定一个阈值，选择比其大的分量对应的特征
      - 指定子集中特征的个数k，选择最大的k个特征

- 确定相关统计量

给定训练集 $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$，对每个示例 $\boldsymbol{x}_i$，

"猜中近邻" (near-hit)：$\boldsymbol{x}_i$ 的同类样本中的最近邻 $\boldsymbol{x}_{i,\mathrm{nh}}$

"猜错近邻" (near-miss)：$\boldsymbol{x}_i$ 的异类样本中的最近邻 $\boldsymbol{x}_{i,\mathrm{nm}}$

相关统计量对应于属性 j 的分量为

$$\delta^j = \sum_i -\mathrm{diff}(x_i^j, x_{i,\mathrm{nh}}^j)^2 + \mathrm{diff}(x_i^j, x_{i,\mathrm{nm}}^j)^2 ,$$

其中 x_a^j 表示样本 $\boldsymbol{x}_a$ 在属性 j 上的取值，$\mathrm{diff}(x_a^j, x_b^j)$ 取决于属性 j 的类型：若属性 j 为离散型，则 $x_a^j = x_b^j$ 时 $\mathrm{diff}(x_a^j, x_b^j) = 0$，否则为 1；若属性 j 为连续型，则 $\mathrm{diff}(x_a^j, x_b^j) = |x_a^j - x_b^j|$，注意 x_a^j, x_b^j 已规范化到 $[0, 1]$ 区间.

包装算法

- LVW(Las Vegas Wrapper, Liu & Setiono,1996)

- 本质上是**对特征**集进行有放回**采样**，找到所有采样中的最优值

输入: 数据集 D;
特征集 A;
学习算法 $\mathfrak{L}$;
停止条件控制参数 T.

过程:

$E = \infty$;
$d = |A|$;
$A^* = A$;
$t = 0$;
while $t < T$ **do**
  随机产生特征子集 A';
  $d' = |A'|$;
  $E' = \text{CrossValidation}(\mathfrak{L}(D^{A'}))$;
  if $(E' < E) \vee ((E' = E) \wedge (d' < d))$ **then**
    $t = 0$;
    $E = E'$;
    $d = d'$;
    $A^* = A'$
  else
    $t = t + 1$
  end if
end while

输出: 特征子集 A^*

LVW 算法描述

初始化

在特征子集A'上通过交叉验证估计学习器误差

嵌入算法

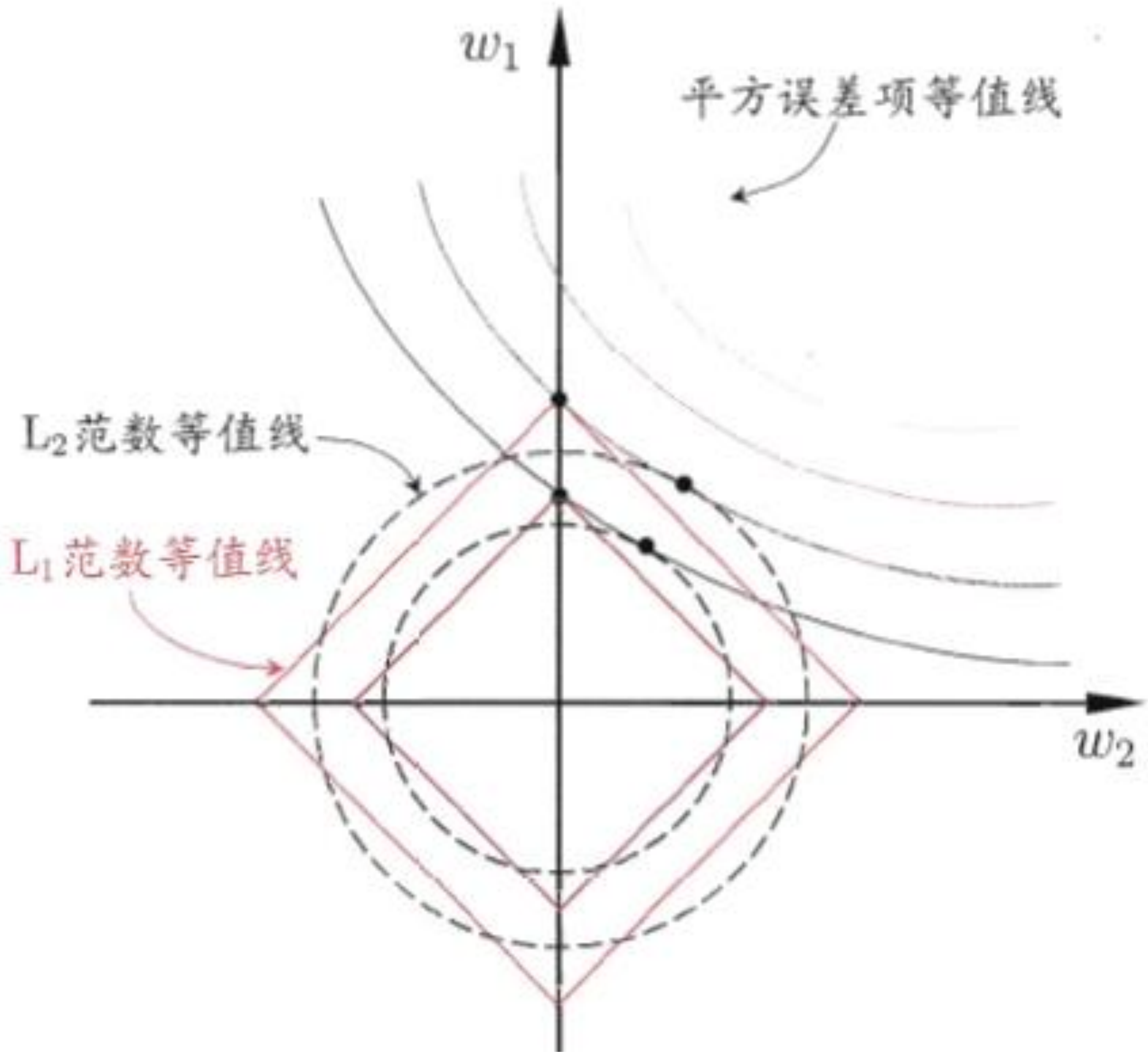
给定数据集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$, 其中 $\boldsymbol{x} \in \mathbb{R}^d, y \in \mathbb{R}$. 考虑最简单的线性回归模型, 优化目标为

$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} (y_i - \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i)^2 \ .$$

采用 L_1 范数, 则有

$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} (y_i - \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i)^2 + \lambda \|\boldsymbol{w}\|_1 \ .$$

其中正则化参数 $\lambda > 0$. 称为 LASSO (Least Absolute Shrinkage and Selection Operator) [Tibshirani, 1996]).



- L1 范 数 切 点 在 轴 上，故 稀 疏
- L2 范 数 切 点 在 象 限 内，故 无 稀 疏
- L0 范 数 在 哪？

概念学习

Concept learning

- 概念（concept）：一个对象或事件集合，它是从更大的集合中选取的子集，或者是在这个较大集合中定义的布尔函数。
  - 如，从动物的集合中选取鸟类
    - 在动物集合中定义的函数，它对鸟类产生true并对其他动物产生false
- 概念学习：从样例中逼近布尔值函数。是指从有关某个布尔函数的输入输出训练样例中，推断出该布尔函数。

- 一个概念学习的例子：Aldo Enjoy Sport

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

- 输入某天的各种属性Sky、AirTemp、Humidity…等条件，Aldo是否进行水上运动?

- 术语定义
  - 概念定义在一个实例（instance）集合之上，这个集合表示为X。
    - 例中，X是所有可能的日子，每个日子由Sky、AirTemp、Humidity、Wind、Water和Forecast六个属性表示。
  - 待学习的概念或函数称为目标概念（target concept），记作c。
    - c可以是定义在实例X上的任意布尔函数，即c:X→{0, 1}。
    - 例中，目标概念对应于属性EnjoySport的值，当EnjoySport=Yes时c(x)=1，当EnjoySport=No时c(x)=0。

- 训练样例（training examples）：X中的一个实例x以及它的目标概念值c(x)。
  - 用序偶<x,c(x)>来描述训练样例
  - 符号D用来表示训练样例的集合。
- 正例（positive example）：对于c(x)=1的实例，也称为目标概念的成员。
- 反例（negative example）：对于c(x)=0的实例，也称为非目标概念成员。

- 给定目标概念c的训练样例集，学习器面临的问题是假设或估计c。
- 使用符号H来表示所有可能假设的集合，称之为假设空间。
  - 通常H依设计者所选择的假设表示而定。
  - H中每个假设h表示X上定义的布尔函数，即h:X→{0,1}。
  - 目标：寻找H中的假设h，使对于X中的所有x，h(x)=c(x)。

- 获取h的方法：归纳学习
- 归纳学习：从特殊的样例得到普遍的规律
  - 归纳学习算法最多只能保证输出的假设与训练样例（特殊）相拟合。
- 归纳学习假设（The Inductive Learning Hypothesis）：任一假设如果在足够大的训练样例集中很好地逼近目标函数，它也能在未见实例中很好地逼近目标函数。（数据同分布）

- 表示假设（Representing Hypotheses）
  - 很多种可能的表示方法
  - 一个简单的形式：实例的各属性约束的合取式∧（Conjunction）
  - EnjoySport例中，令每个假设为6个约束（或变量）的向量，每个约束对应一个属性可取值范围，为
    - 由“?”表示任意值（如，AirTemp=? ）
    - 明确指定的属性值（如，AirTemp=Warm）
    - 由“Ø”表示不接受任何值（如，AirTemp=Ø）
    - 如，
      - < Sunny , ?, ?, Strong , ? , Same >
      - <?, ?, ?, ?, ?, ?>　　所有的样例都是正例（最一般，任取皆正）
      - < Ø, Ø, Ø, Ø, Ø, Ø >　所有的样例都是反例（最特殊，任取皆零）

- 任何概念学习任务能被描述为
  - 实例的集合（总体）
  - 实例集合上的目标函数（目标）
  - 训练样例的集合（样本）
  - 候选假设的集合（假设）
- EnjoySport 概念学习任务

EnjoySport 概念学习任务

- 已知：
  - 实例集 *X*：可能的日子，每个日子由下面的属性描述：
    - *Sky*（可取值为 *Sunny*，*Cloudy* 和 *Rainy*）
    - *AirTemp*（可取值为 *Warm* 和 *Cold*）
    - *Humidity*（可取值为 *Normal* 和 *High*）
    - *Wind*（可取值为 *Strong* 和 *Weak*）
    - *Water*（可取值为 *Warm* 和 *Cool*）
    - *Forecast*（可取值为 *Same* 和 *Change*）
  - 假设集 *H*：每个假设描述为 6 个属性 *Sky*，*AirTemp*，*Humidity*，*Wind*，*Water* 和 *Forecast* 的值约束的合取。约束可以为“?”（表示接受任意值），“∅”（表示拒绝所有值），或一特定值。
  - 目标概念 *c*: *EnjoySport*: $X \rightarrow \{0, 1\}$
  - 训练样例集 *D*：目标函数的正例和反例
- 求解：
  - *H* 中的一假设 *h*，使对于 *X* 中任意 *x*，$h(x)=c(x)$。

- 当假设的表示形式选定后，那么就隐含地为学习算法确定了所有假设的空间
  - 这些假设是学习程序所能表示的
  - 也是它能够学习的
  - 例，Enjoy-Sport的假设空间
- 概念学习可以看作一个搜索的过程
  - 搜索范围：假设的表示所隐含定义的整个空间
  - 搜索目标：能够最好地拟合训练样例的假设

- 假设的一般到特殊序关系
  - 考虑下面两个假设
    - h1=<sunny, ?, ?, Strong, ?, ?>
    - h2=<Sunny, ?, ?, ?, ?, ?>
  - 任何被h1划分为正例的实例都会被h2划分为正例，因此h2比h1更一般。
- 利用这个关系，无需列举所有假设，就能在无限的假设空间中进行较彻底的搜索

- more-general-than-or-equal-to（更一般或相等）
- 定义：令h_j和h_k为在X上定义的布尔函数。定义一个more-general-than-or-equal-to关系，记做$\geq_g$。称$h_j \geq_g h_k$当且仅当

$$(\forall x \in X)[(h_k(x)=1) \rightarrow (h_j(x)=1)]$$

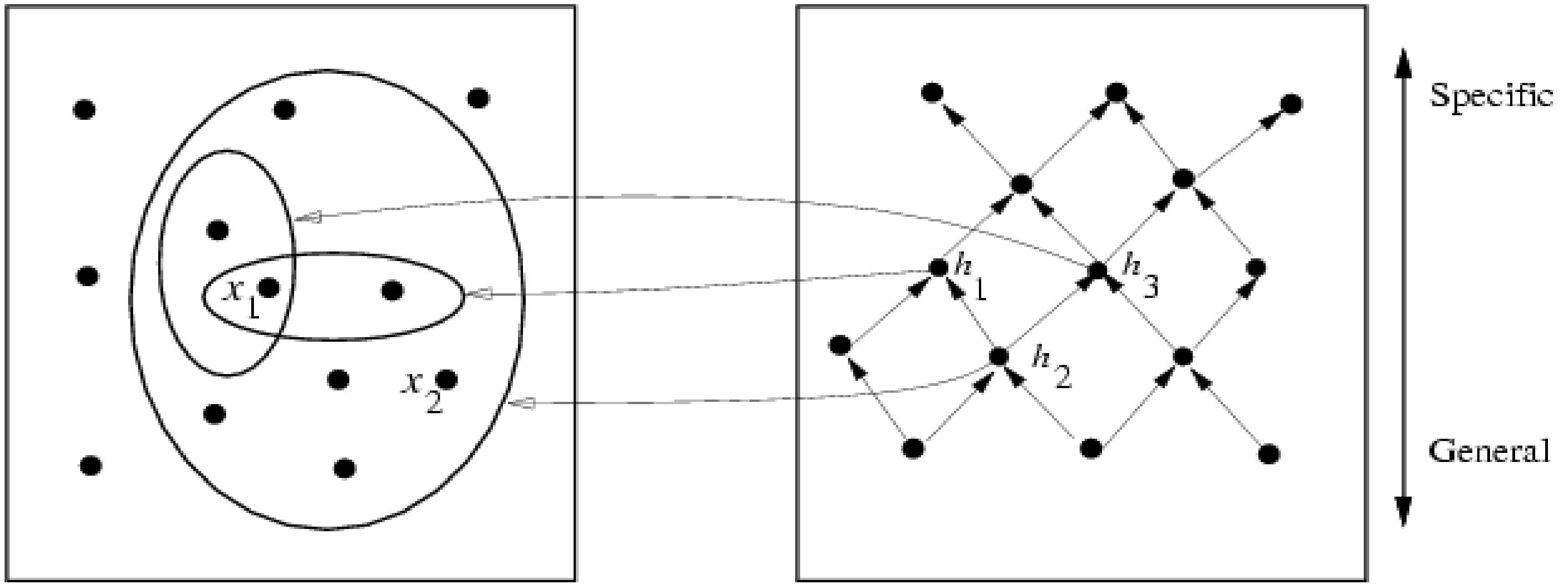
- 对X中任意实例x和H中任意假设h，我们说x满足h当且仅当h(x)=1。
- 给定假设h_j和h_k，h_j more-general-than-or-equal-to h_k，当且仅当任意一个满足h_k的实例同时也满足h_j。
- h_j严格的more-general-than h_k（写作$h_j >_g h_k$），当且仅当$(h_j \geq_g h_k) \wedge \neg(h_k \geq_g h_j)$。
- 逆向的关系"比……更特殊"：h_j more-specific-than h_k，当h_k more-general-than h_j。

- Instance, Hypotheses, and More-General-Than

$$h_1 = \langle Sunny, ?, ?, Strong, ?, ? \rangle$$
$$h_2 = \langle Sunny, ?, ?, ?, ?, ? \rangle$$
$$h_3 = \langle Sunny, ?, ?, ?, Cool, ? \rangle$$

$$x_1 = \langle Sunny, Warm, High, Strong, Cool, Same \rangle$$
$$x_2 = \langle Sunny, Warm, High, Light, Warm, Same \rangle$$

Instances X   *Hypotheses H*



x_1 = <Sunny, Warm, High, Strong, Cool, Same>
x_2 = <Sunny, Warm, High, Light, Warm, Same>

h_1 = <Sunny, ?, ?, Strong, ?, ?>
h_2 = <Sunny, ?, ?, ?, ?, ?>
h_3 = <Sunny, ?, ?, ?, Cool, ?>

- 练习：给出下列假设的偏序关系。

h1: <Sunny, Warm, ?, Strong, ?, ?>

h2: <Sunny, ?, ?, Strong, ?, ?>

h3: <Sunny, Warm, ?, ?, ?, ?>

h4: <?, Warm, ?, Strong, ?, ?>

h5: <Sunny, ?, ?, ?, ?, ?>

h6: <?, Warm, ?, ?, ?, ?>

- 一致（Consistent）
- 定义：一个假设h与训练样例集合D一致，当且仅当对D中每一个样例<x,c(x)>，h(x)=c(x)。

$$Consistent(h,D) \equiv (\forall \langle x,c(x)\rangle \in D)\ h(x)=c(x)$$

  - 与“满足”不同
    - 一个样例x在h(x)=1时称为满足假设h，不论x是目标概念的正例还是反例。
    - 这一样例是否与h一致与目标概念有关，即是否h(x)=c(x)。

THE END!