

数据挖掘与机器学习

潘斌

panbin@nankai.edu.cn

范孙楼227

1

上节回顾

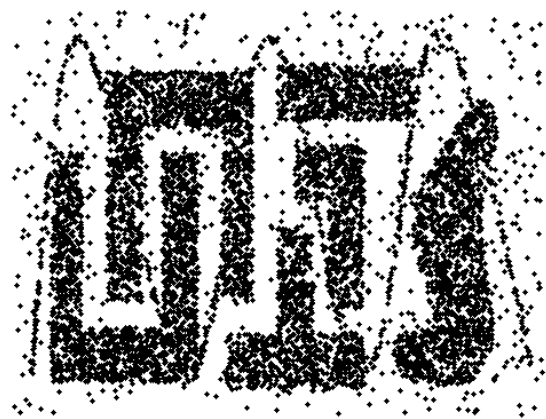
- 什么是数据挖掘
 - 数据挖掘是在大型数据存储库中，自动的发现有用信息的过程
- 什么是机器学习
 - 可自动发现有用信息的手段即为机器学习算法
- 数据的特点
 - 数据属性
 - 数据质量
 - 数据预处理



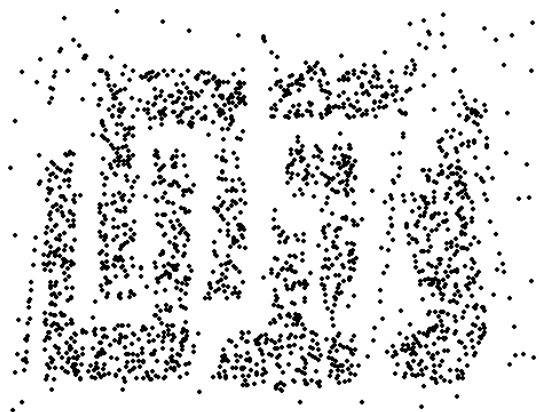
数据预处理（接上节）

- 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

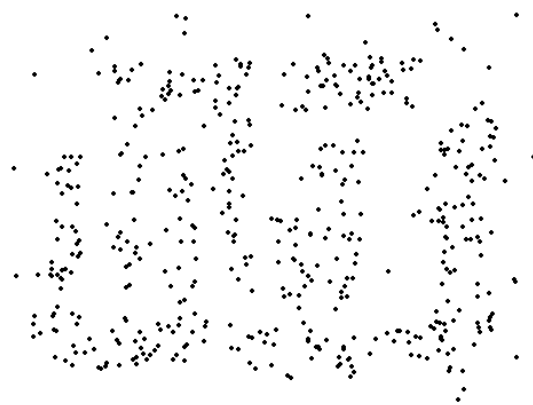
抽样与信息损失



8000 points



2000 Points



500 Points



数据预处理

- 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的（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 \mathbf{x} ，维数为 M ，线性组合的新特征向量 \mathbf{y}

- \mathbf{y} 的各分量
$$y_i = \sum_{j=1}^m a_{ij} x_j = \mathbf{a}_i^T \mathbf{x}$$

- 基本约束条件
$$\mathbf{a}_i^T \mathbf{a}_i = 1$$

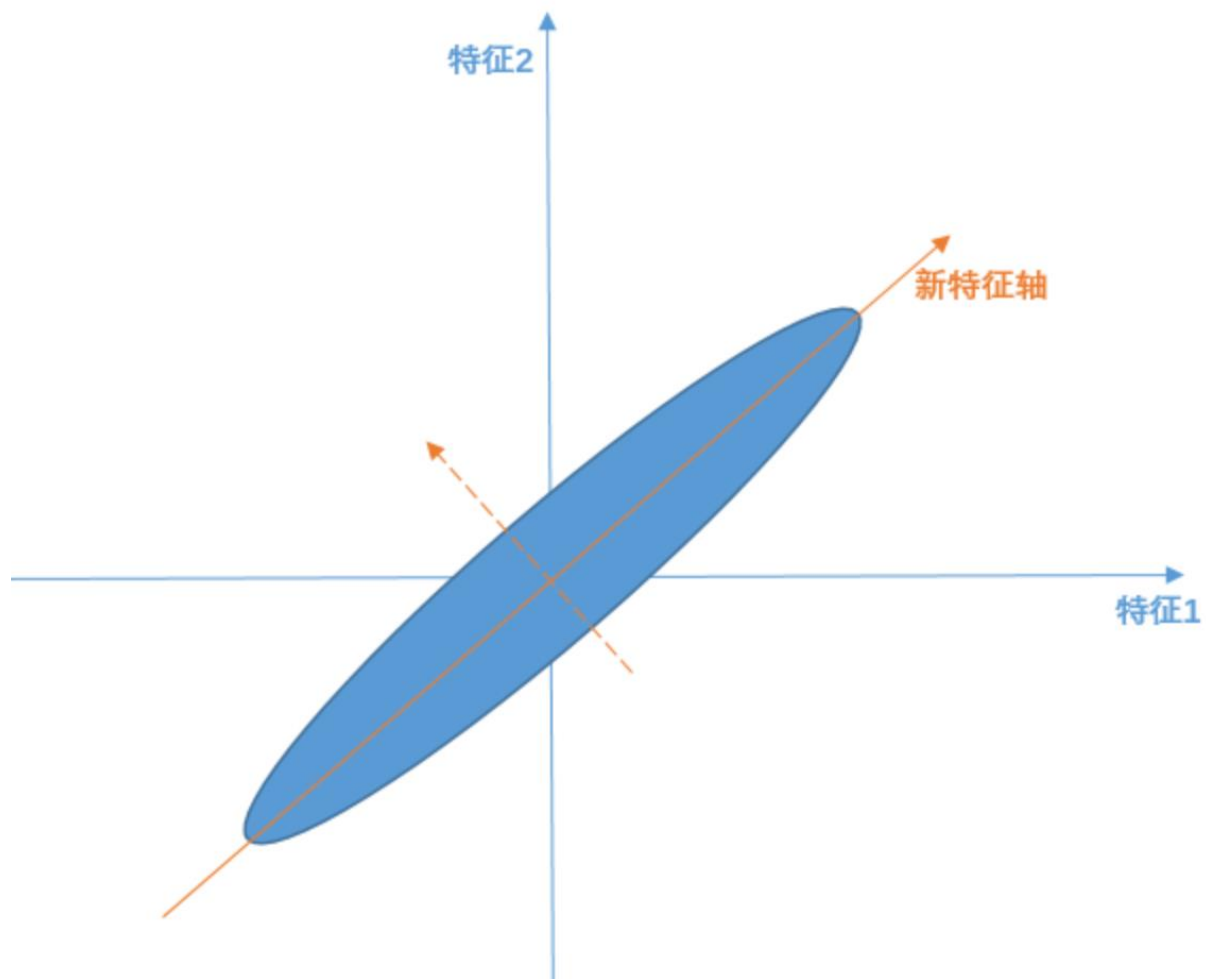
- 矩阵形式
$$\mathbf{y} = \mathbf{A}^T \mathbf{x}$$

- 求最佳的正交变换矩阵 \mathbf{A} ，使得新特征的方差达到极值



PCA

主轴方向方差大



PCA

- PCA求解

- 原理：方差越大，特征越“重要”

- 方差

$$\Sigma = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T$$

- a_1 约束条件

$$a_1^T a_1 = 1$$

- a_2 约束条件

- y_2 与 y_1 不相关

- 即 a_2 与 a_1 正交

$$a_2^T a_2 = 1$$

$$a_2^T a_1 = 0$$

- a_3 约束条件.....



PCA

- **PCA求解**

- 求解 a_1 (约束优化问题)

$$L(a_1, \lambda_1) = a_1^T \Sigma a_1 - \lambda_1 (a_1^T a_1 - 1)$$

$$\Sigma a_1 = \lambda_1 a_1$$

$$\text{var}(y_1) = a_1^T \Sigma a_1 = \lambda_1$$

- 得到协方差阵 Σ 的第一本征值和第一本征向量
- 然后依次求解后续本征值和本征向量即可



PCA

- **PCA**特点

- 求解协方差阵 Σ 的本征值对应的本征向量
- 新特征按方差大小顺序排列——“重要性”
- 新特征之间不相关——正交变换
- 不考虑样本的类别——非监督方法



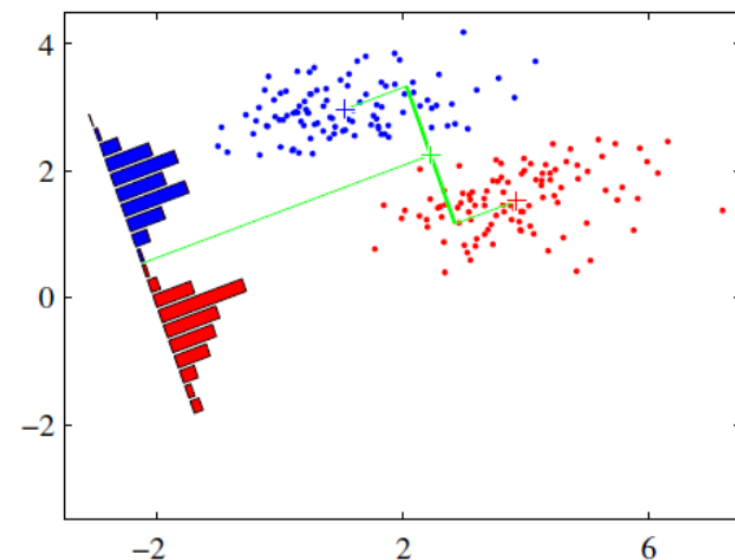
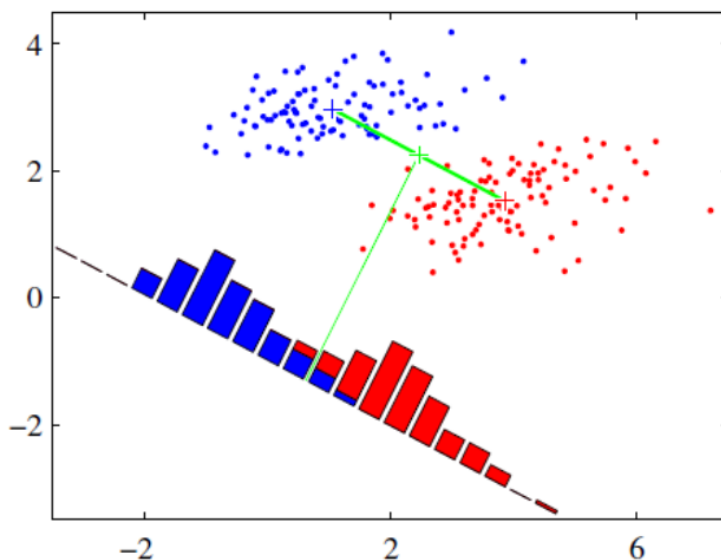
LDA

- **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**



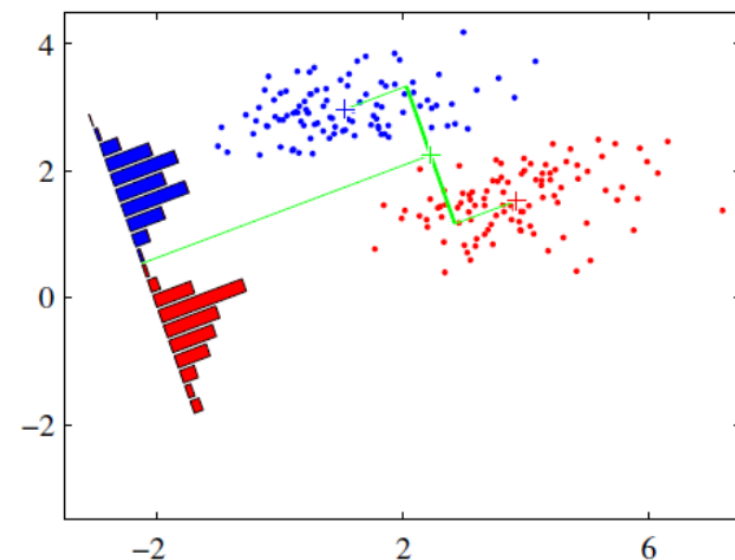
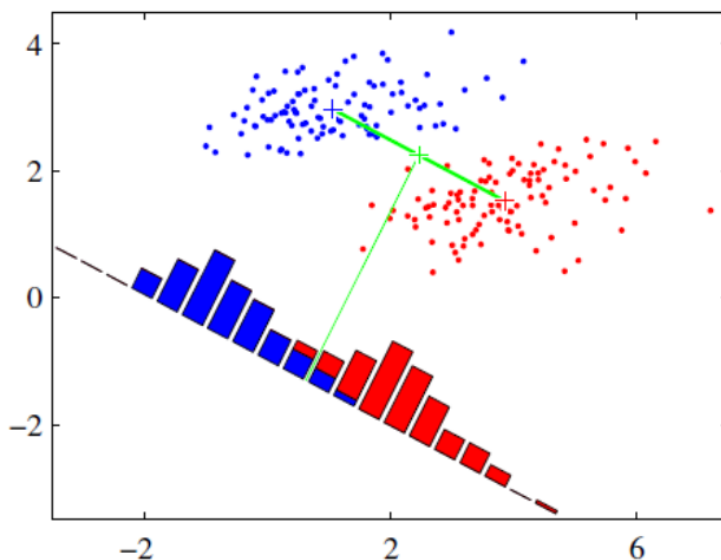
LDA

- 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

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

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

令分母为1

$$\begin{aligned} \min_{\mathbf{w}} \quad & -\mathbf{w}^T \mathbf{S}_b \mathbf{w} \\ \text{s.t.} \quad & \mathbf{w}^T \mathbf{S}_w \mathbf{w} = 1 \end{aligned}$$

拉格朗日对偶乘子法



数据预处理

- 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 $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$, 对每个示例 \mathbf{x}_i ,

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

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

相关统计量对应于属性 j 的分量为

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

其中 x_a^j 表示样本 \mathbf{x}_a 在属性 j 上的取值, $\text{diff}(x_a^j, x_b^j)$ 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 $\text{diff}(x_a^j, x_b^j) = 0$, 否则为 1; 若属性 j 为连续型, 则 $\text{diff}(x_a^j, x_b^j) = |x_a^j - x_b^j|$, 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)

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

过程:

```
1:  $E = \infty$ ;  
2:  $d = |A|$ ;  
3:  $A^* = A$ ;  
4:  $t = 0$ ;  
5: while  $t < T$  do  
6:   随机产生特征子集  $A'$ ;  
7:    $d' = |A'|$ ;  
8:    $E' = \text{CrossValidation}(\mathcal{L}(D^{A'}))$ ;  
9:   if  $(E' < E) \vee ((E' = E) \wedge (d' < d))$  then  
10:     $t = 0$ ;  
11:     $E = E'$ ;  
12:     $d = d'$ ;  
13:     $A^* = A'$   
14:   else  
15:     $t = t + 1$   
16:   end if  
17: end while  
输出: 特征子集  $A^*$ .
```

初始化

在特征子集 A' 上
通过交叉验证估计
学习器误差

■ 本质上是**对特征集**进行
所有放回采样, 找到
所有采样中的最优值



■ 嵌入算法

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

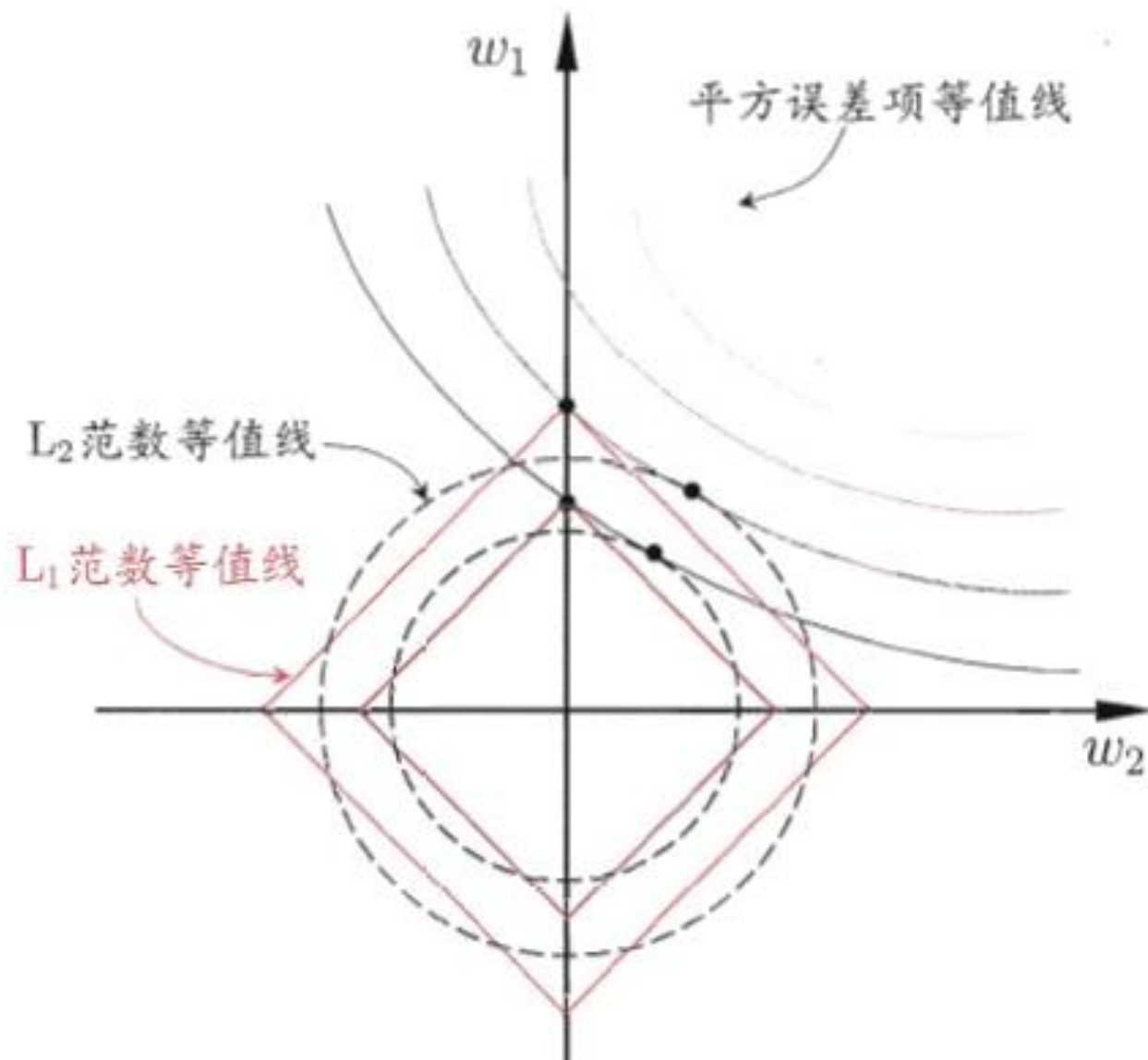
$$\min_{\mathbf{w}} \sum_{i=1}^m (y_i - \mathbf{w}^T \mathbf{x}_i)^2 .$$

采用 L_1 范数, 则有

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

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？

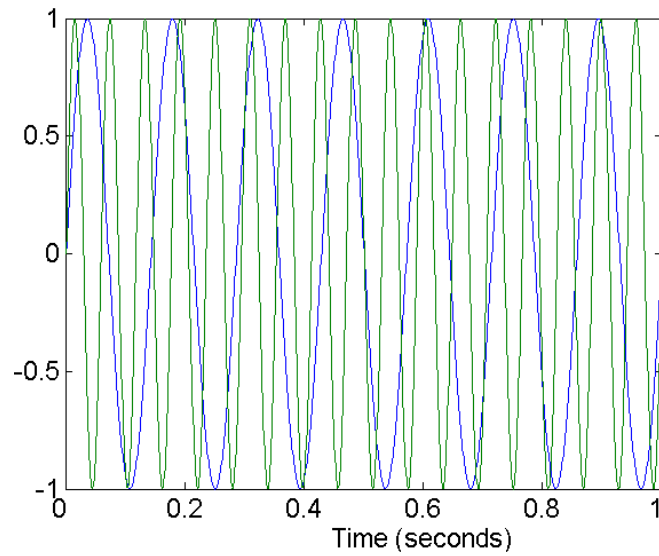


数据预处理

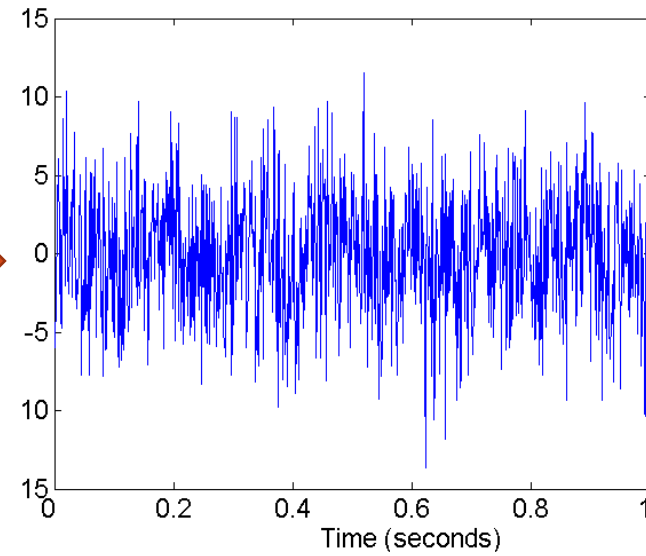
■ 特征创建

- 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

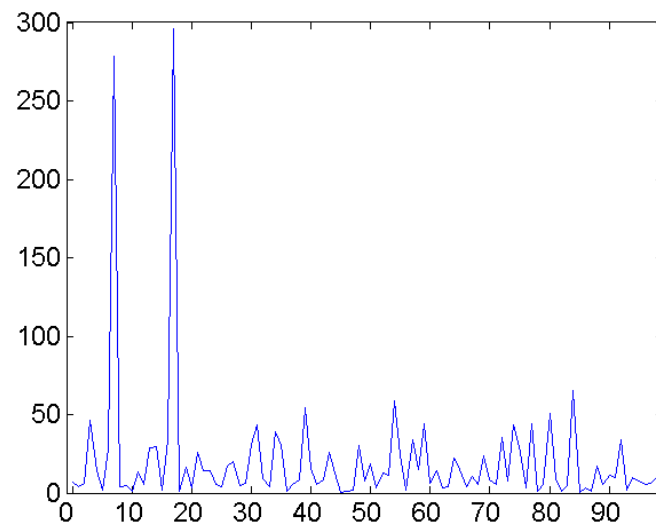




Two Sine Waves



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 $n = \lceil \log_2 m \rceil$ 个二进制位表示这些整数，因此需要 n 个二元属性表示这些二进制数



数据预处理

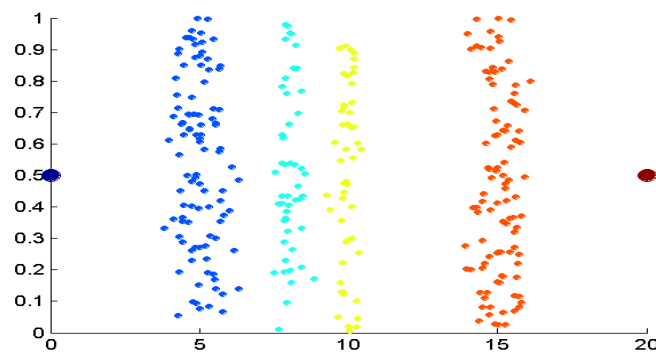
- 例，具有五个值 {awful, poor, ok, good, great} 的分类变量需要三个二元变量

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

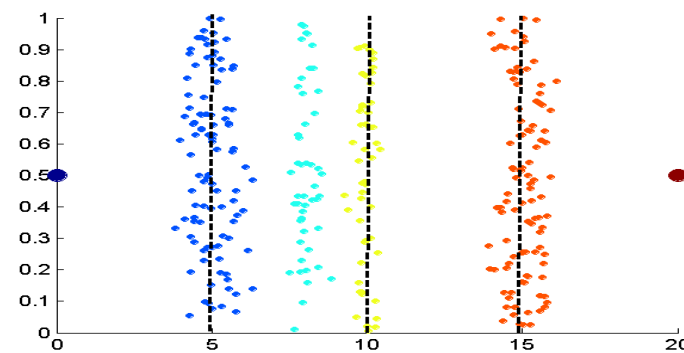


数据预处理

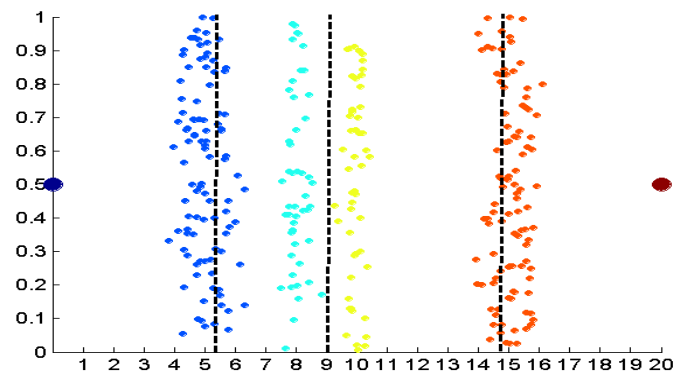
■ 离散化方法（非监督）



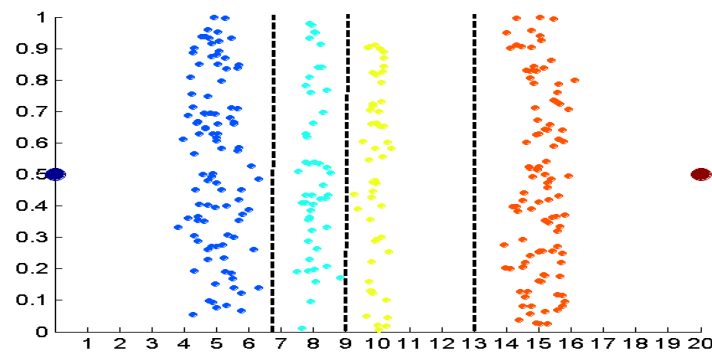
原始数据



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



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	<i>setosa</i>
4.9	3	1.4	0.2	<i>setosa</i>
.....				
5.7	2.9	4.2	1.3	<i>versicolor</i>
5.7	2.8	4.1	1.3	<i>versicolor</i>
.....				
5.8	2.7	5.1	1.9	<i>virginica</i>
7.1	3	5.9	2.1	<i>virginica</i>
.....				

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 $\{v_1, v_2, \dots, v_k\}$ 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

- x_p : 对应一个 x 值, 使得 x 的 $p\%$ 观测值小于 x_p

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

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

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

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： $\text{variance}(x) = s_x^2 = \frac{1}{m-1} \sum_{i=1}^m (x_i - \bar{x})^2$

- 绝对平均偏差：（ absolute average deviation ）

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

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

$$\text{MAD}(x) = \text{median} \left(\{|x_1 - \bar{x}|, \dots, |x_m - \bar{x}|\} \right)$$

- 四分位数极差（ interquartile range ）

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



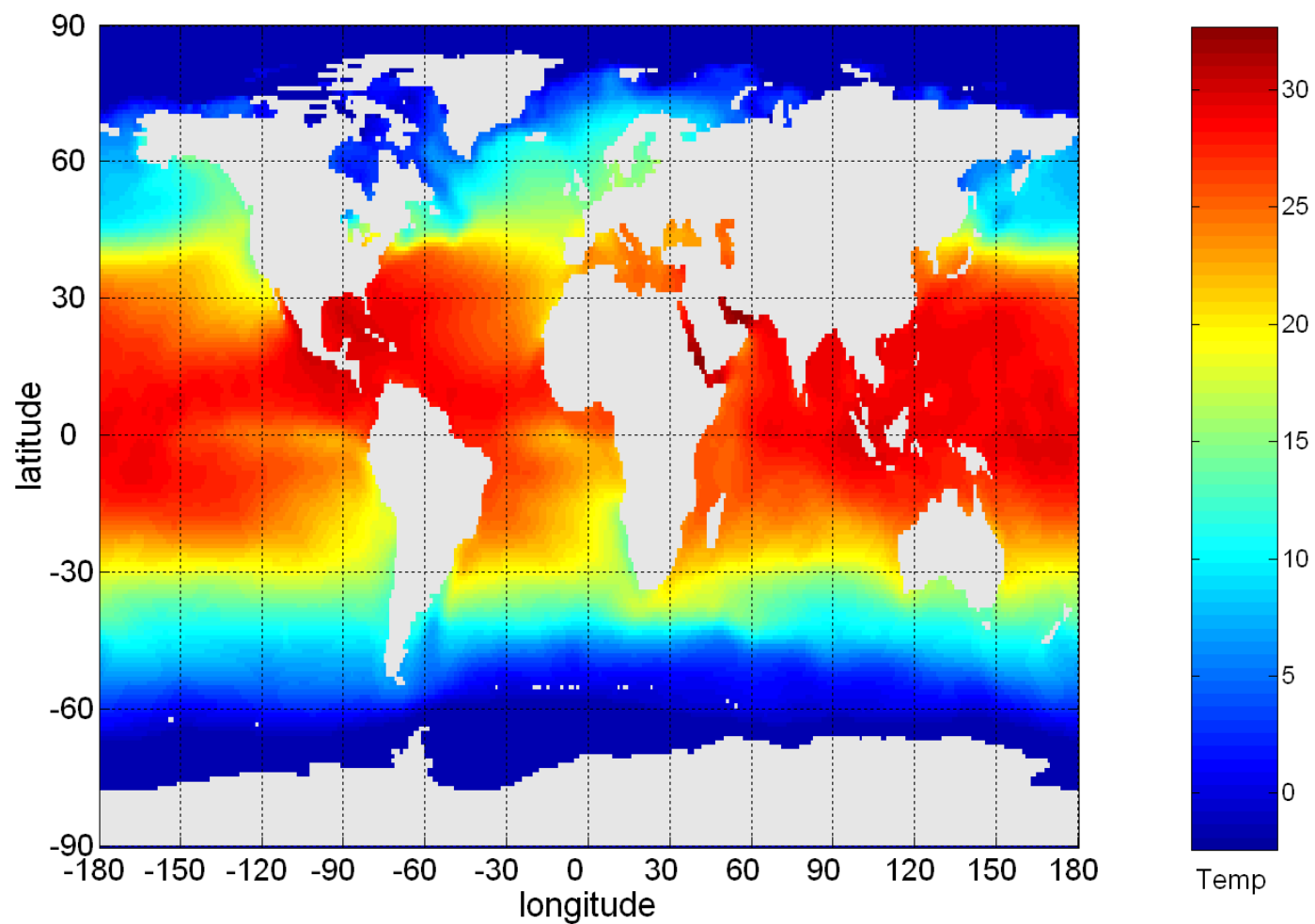
汇总统计

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



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

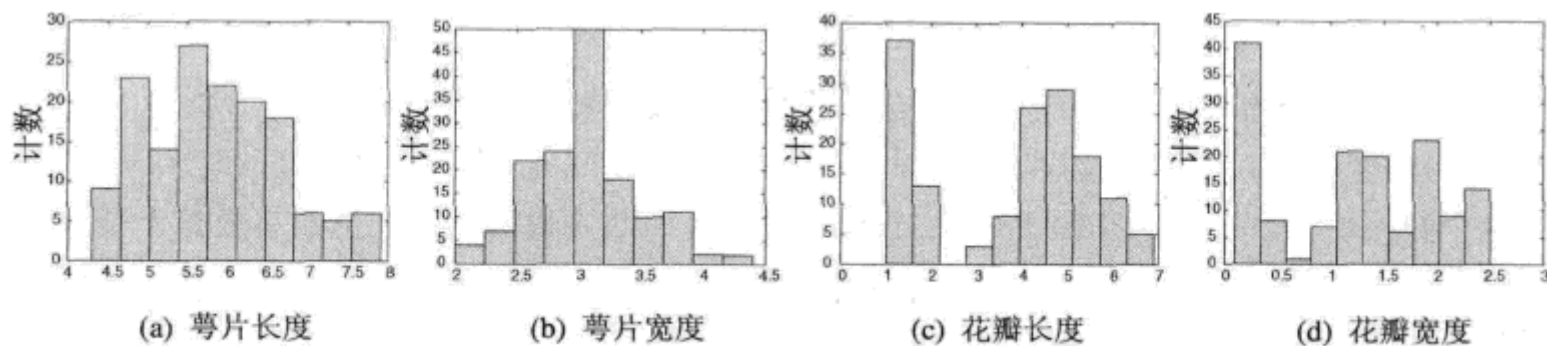


	6	1	3	2	5	4
4	1	1	1	0	0	0
2	1	1	1	0	0	0
6	1	1	1	0	0	0
8	1	1	1	0	0	0
5	0	0	0	1	1	1
3	0	0	0	1	1	1
9	0	0	0	1	1	1
1	0	0	0	1	1	1
7	0	0	0	1	1	1

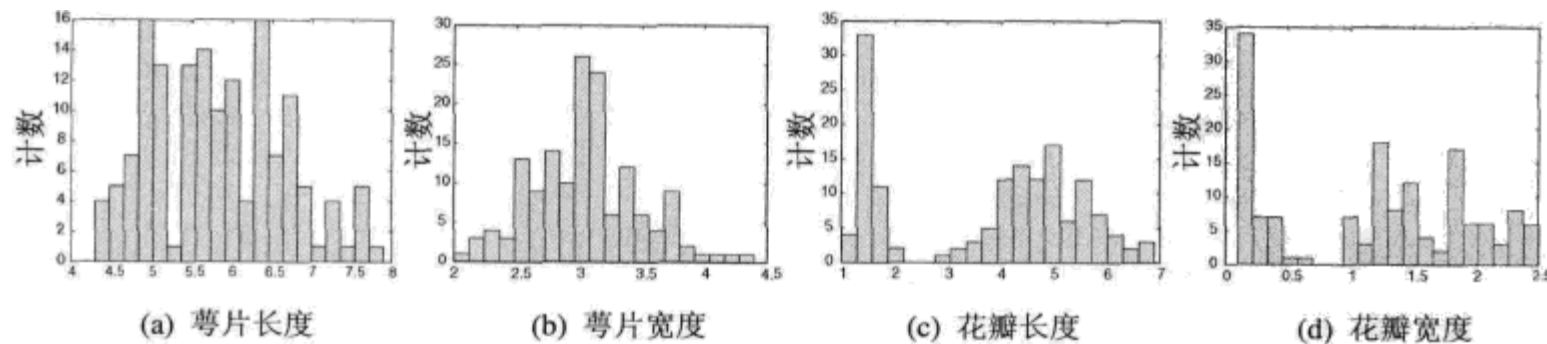


可视化

■ 直方图 (Histogram)



四个鸢尾花属性的直方图 (10 个箱)

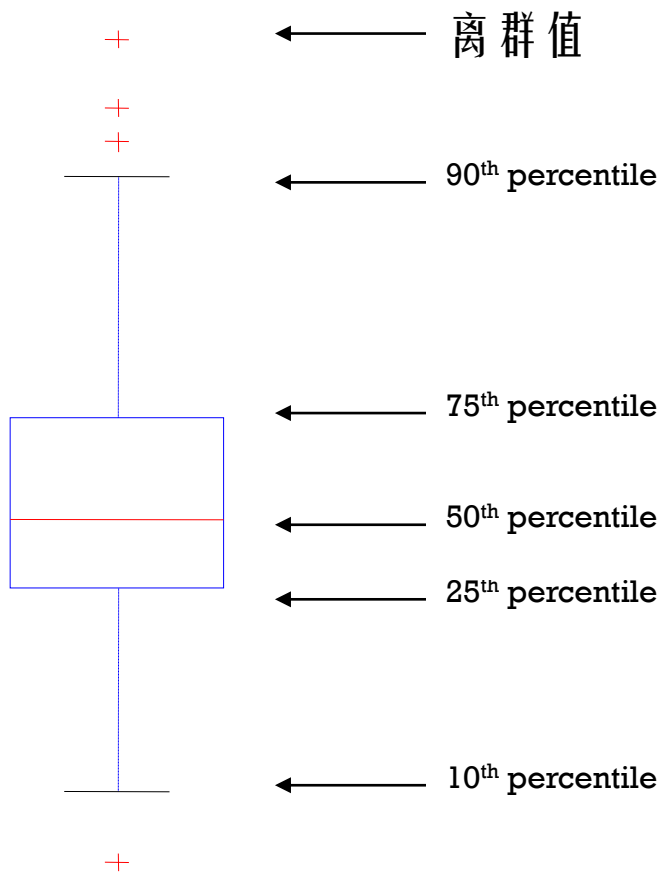


四个鸢尾花属性的直方图 (20 个箱)



可视化

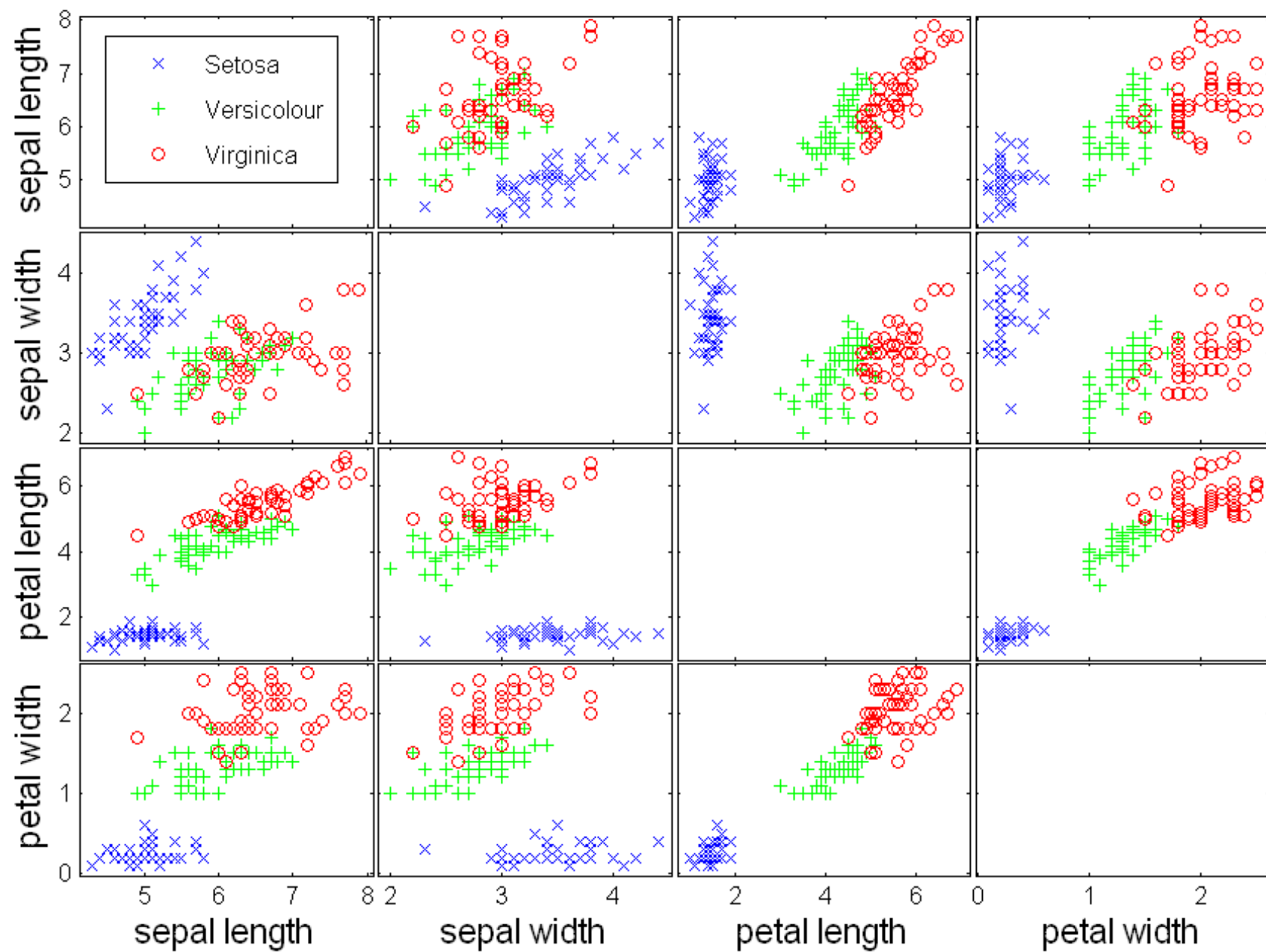
■ 盒状图 (box plot)



可 视 化

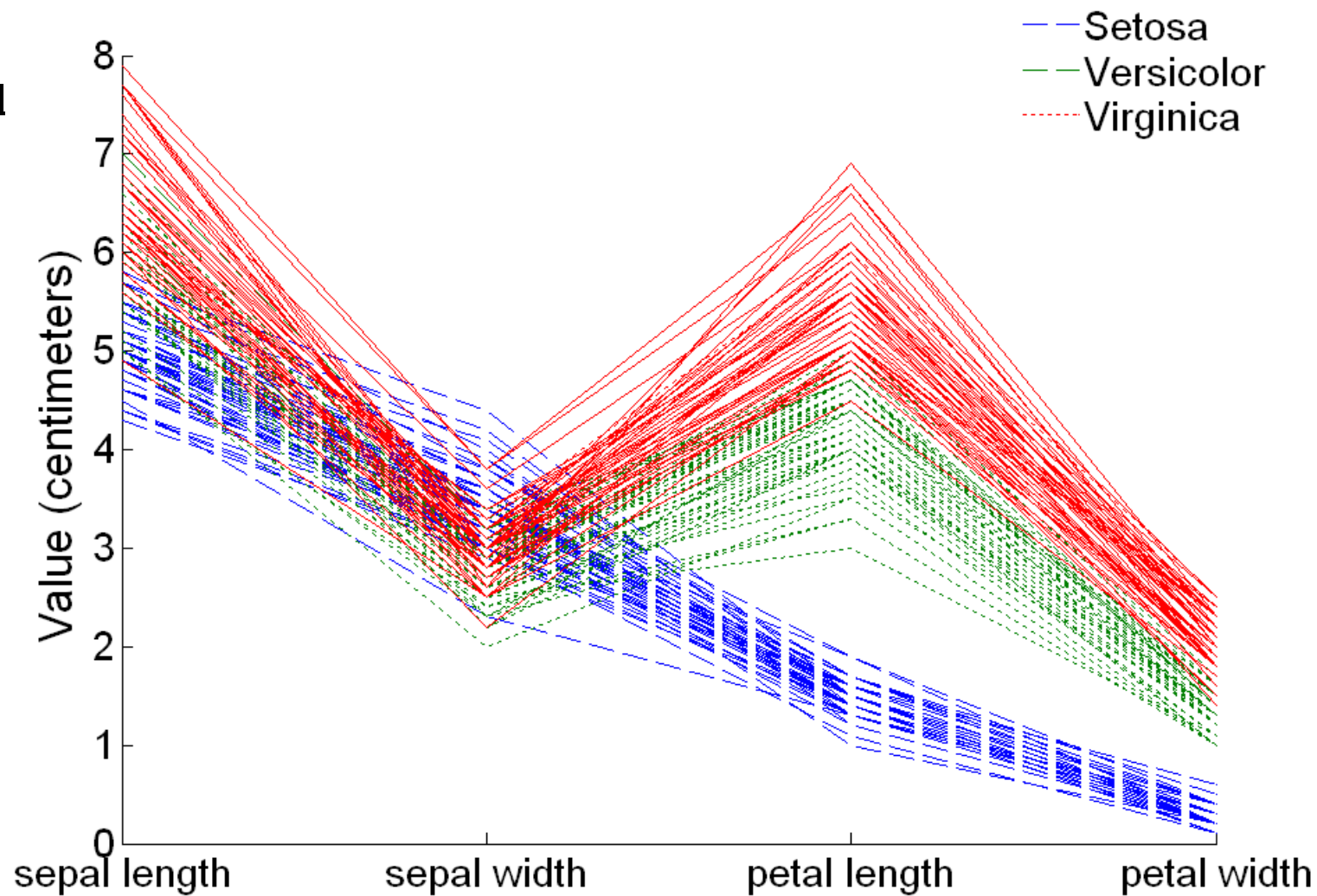
- 散布图 (Scatter plots)
 - 图形化显示二属性之间的关系
 - 当类标号给出时，考察二属性将类分开的程度

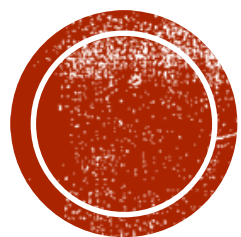




可视化

- 平行坐标系 (Parallel





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	<i>Sky</i>	<u><i>AirTemp</i></u>	<i>Humidity</i>	<i>Wind</i>	<i>Water</i>	<i>Forecast</i>	<u><i>EnjoySport</i></u>
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 $c:X \rightarrow \{0, 1\}$ 。
 - 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 $\langle x, c(x) \rangle$ 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 $h:X \rightarrow \{0,1\}$ 。
 - 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 “ \emptyset ” 表示不接受任何值 (如，AirTemp= \emptyset)
 - 如，
 - < Sunny , ?, ?, Strong , ? , Same >
 - <?, ?, ?, ?, ?, ?> 所有的样例都是正例 (最一般，任取皆正)
 - < \emptyset , \emptyset , \emptyset , \emptyset , \emptyset , \emptyset > 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), “ \emptyset ” (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
 - $h1 = \langle \text{sunny}, ?, ?, \text{Strong}, ?, ? \rangle$
 - $h2 = \langle \text{Sunny}, ?, ?, ?, ?, ? \rangle$
 - 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

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



- 对 \mathbf{X} 中任意实例 \mathbf{x} 和 \mathbf{H} 中任意假设 h ，我们说 \mathbf{x} 满足 h 当且仅当 $h(\mathbf{x})=1$ 。
- 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

$h_1 = \langle \text{Sunny}, ?, ?, \text{Strong}, ?, ? \rangle$

$h_2 = \langle \text{Sunny}, ?, ?, ?, ?, ? \rangle$

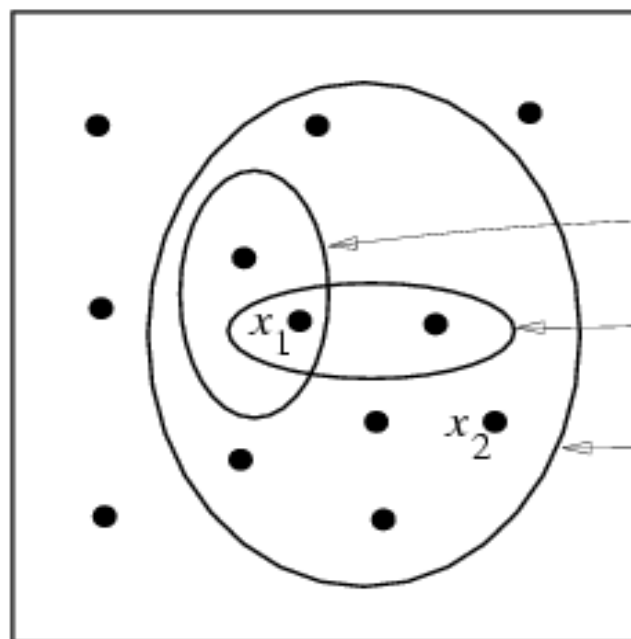
$h_3 = \langle \text{Sunny}, ?, ?, ?, \text{Cool}, ? \rangle$

$x_1 = \langle \text{Sunny}, \text{Warm}, \text{High}, \text{Strong}, \text{Cool}, \text{Same} \rangle$

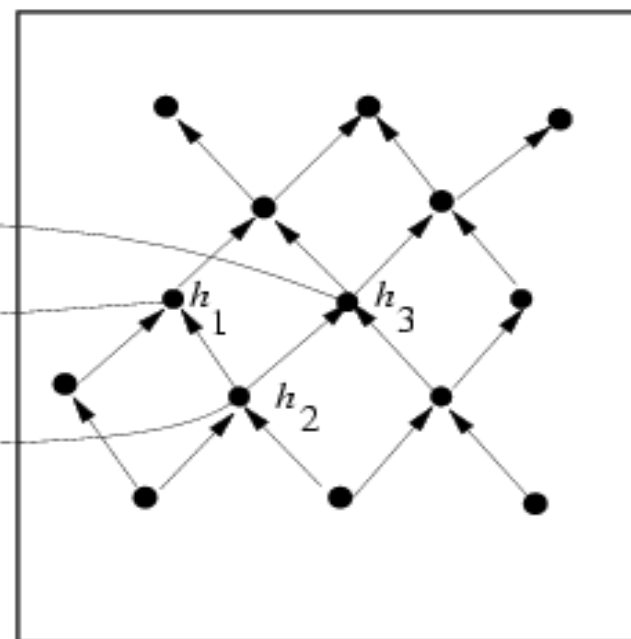
$x_2 = \langle \text{Sunny}, \text{Warm}, \text{High}, \text{Light}, \text{Warm}, \text{Same} \rangle$



Instances X



Hypotheses H



Specific

General

$x_1 = \langle \text{Sunny, Warm, High, Strong, Cool, Same} \rangle$

$x_2 = \langle \text{Sunny, Warm, High, Light, Warm, Same} \rangle$

$h_1 = \langle \text{Sunny, ?, ?, Strong, ?, ?} \rangle$

$h_2 = \langle \text{Sunny, ?, ?, ?, ?, ?} \rangle$

$h_3 = \langle \text{Sunny, ?, ?, ?, Cool, ?} \rangle$



■ 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 $\langle x, c(x) \rangle$ ， $h(x) = c(x)$ 。

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

- 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 $h(x) = c(x)$ 。



■ 求解 h 的算法

■ Find-S Algorithm

■ 基本思想：

- 使用more_general_than偏序的搜索算法
- 沿着偏序链，从较特殊的假设逐渐转移到较一般的假设。
- 在每一步，假设只在需要覆盖新的正例时被泛化。
- 每一步得到的假设，都是在那一点上与训练样例一致的最特殊的假设。



■ Find-S Algorithm

- 将 h 初始化为 H 中最特殊假设
- 对每个正例 x
 - 对 h 的每个属性约束 a_i
 - 如果 x 满足 a_i
 - 那么不做任何事
 - 否则，将 h 中 a_i 替换为 x 满足的紧邻的更一般约束
- 输出假设 h



■ Find-S 算法实例 $h_0 \leftarrow \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$

$x_1 = \langle \text{Sunny Warm Normal Strong Warm Same} \rangle, +$

$h_1 \leftarrow \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle$

$x_2 = \langle \text{Sunny Warm High Strong Warm Same} \rangle, +$

$h_2 \leftarrow \langle \text{Sunny, Warm, ?, Strong, Warm, Same} \rangle$

$x_3 = \langle \text{Rainy Cold High Strong Warm Change} \rangle, -$

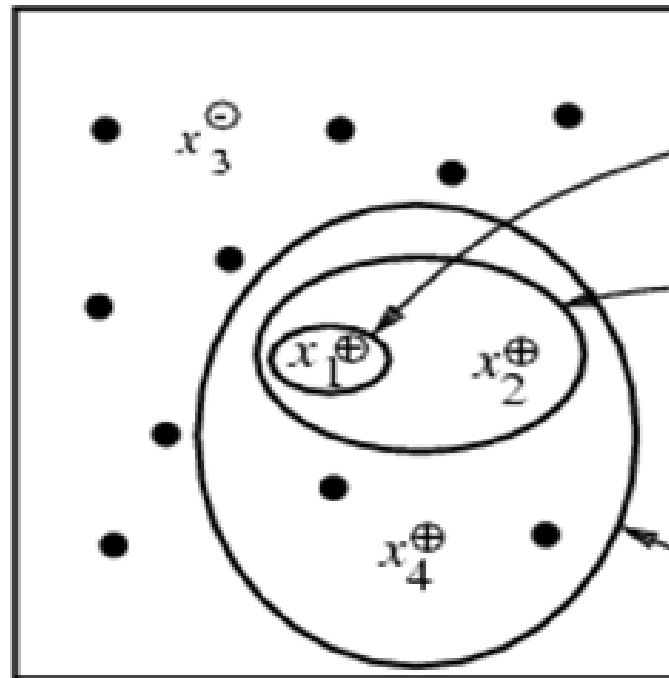
$h_3 \leftarrow \langle \text{Sunny, Warm, ?, Strong, Warm, Same} \rangle$

$x_4 = \langle \text{Sunny Warm High Strong Cool Change} \rangle, +$

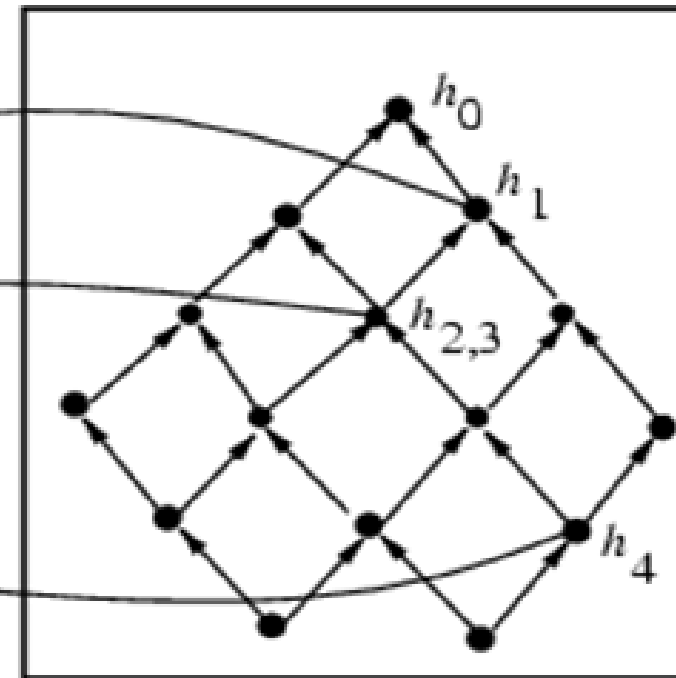
$h_4 \leftarrow \langle \text{Sunny, Warm, ?, Strong, ?, ?} \rangle$



Instances X



Hypotheses H



Specific
↑
↓
General



■ 练习

Outlook	Temperature	Humidity	Wind	PlayTennis
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Mild	High	Strong	No
Sunny	Cool	Normal	Weak	Yes
Overcast	Hot	Normal	Weak	Yes



■ Find-S 学习算法的特点及不足

- Find-S 的重要特点：对以属性约束的合取式（conjunction）描述的假设空间 H ，保证输出为 H 中与正例一致的最特殊的假设。
- 存在的问题（反例；容错性）



- 变型空间（Version space，版本空间）

- 定义：关于假设空间 H 和训练样例集 D 的变型空间，标记为 $VS_{H,D}$ ，是 H 中与训练样例集 D 一致的所有假设构成的子集。

$$VS_{H,D} = \{h \in H \mid \text{Consistent}(h,D)\}$$

- 与训练样例集一致的所有假设组成的集合
- 包含的是目标概念的所有合理的变型



■ 求解 h 的算法

- 列表后消除算法 (The List-Then-Eliminate Algorithm)
 - 变型空间 $\text{VersionSpace} \leftarrow$ 包含 H 中所有假设的列表
 - 对每个训练样例 $\langle \mathbf{x}, c(\mathbf{x}) \rangle$ 从变型空间中 **移除所有 $h(\mathbf{x}) \neq c(\mathbf{x})$ 的假设 h**
 - 输出 VersionSpace 中的假设列表
 - 只要假设空间是 **有限的**，就可使用
 - 保证得到 **所有** 与训练数据一致的假设
 - 非常繁琐地列出 H 中的所有假设，大多数实际的假设空间 **无法做到**



■ 变型空间举例

■ EnjoySport

Example	<i>Sky</i>	<i>AirTemp</i>	<i>Humidity</i>	<i>Wind</i>	<i>Water</i>	<i>Forecast</i>	<i>EnjoySport</i>
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

结果如→

每来一个样本，压
缩一次假设空间

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

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

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

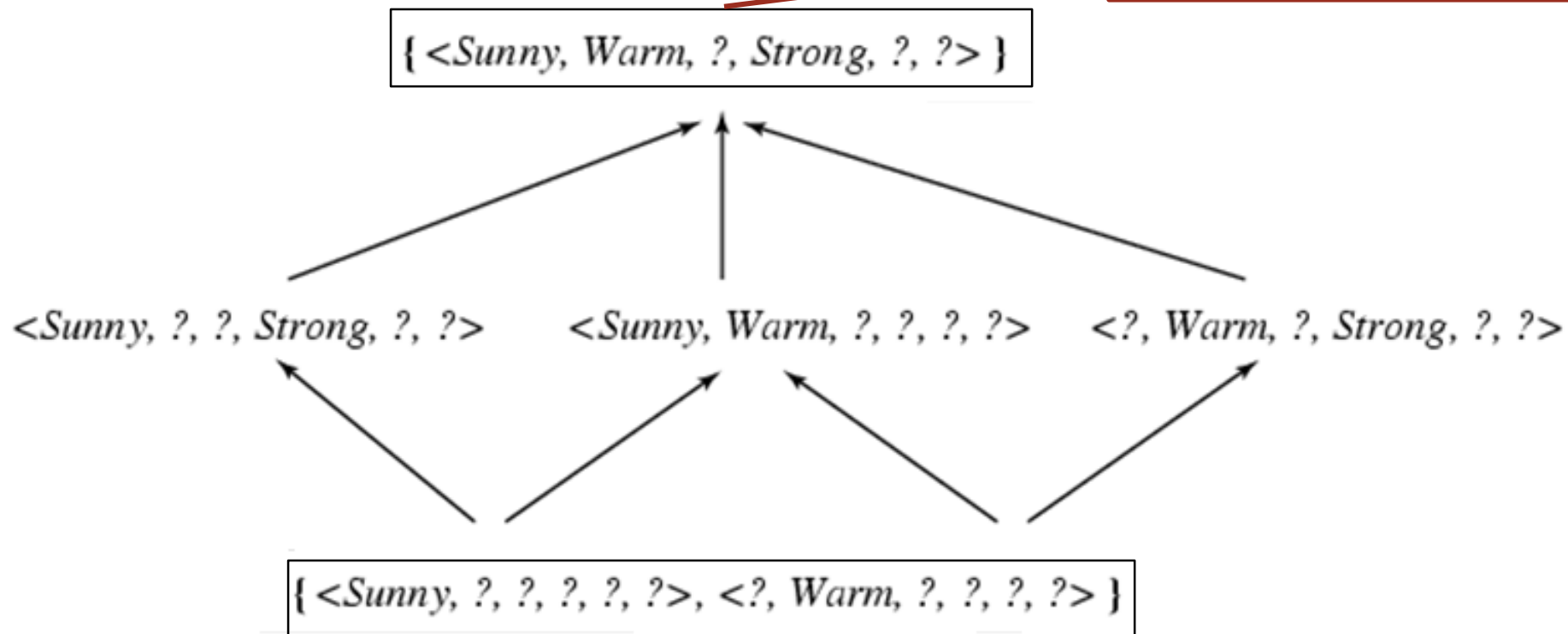
h4: <?, Warm, ?, Strong, ?, ?>

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

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



Find-S输出的假设



- 求解**h**的算法： 候选消除算法

- 变型空间的两个边界定义

- 一般边界（ General Boundary ） **G**：

- **H**中与**D**相一致的极大一般成员的集合。

$$G \equiv \{ g \in H \mid \text{Consistent}(g, D) \wedge (\neg \exists g' \in H)[(g' >_g g) \wedge \text{Consistent}(g', D)] \}$$

- 特殊边界（ Specific Boundary ） **S**：

- 在**H**中与**D**相一致的极大特殊成员的集合。

$$S \equiv \{ s \in H \mid \text{Consistent}(s, D) \wedge (\neg \exists s' \in H)[(s >_g s') \wedge \text{Consistent}(s', D)] \}$$



■ 求解 h 的算法

将 G 集合初始化为 H 中极大一般假设

■ 候选消除算法

将 S 集合初始化为 H 中极大特殊假设

对每个训练例 d ，进行以下操作：

- 如果 d 是一正例

- 从 G 中移去所有与 d 不一致的假设

- 对 S 中每个与 d 不一致的假设 s

- 从 S 中移去 s

- 把 s 的所有的极小一般化式 h 加入到 S 中，其中 h 满足

- h 与 d 一致，而且 G 的某个成员比 h 更一般

- 从 S 中移去所有这样的假设：它比 S 中另一假设更一般

- 如果 d 是一个反例

- 从 S 中移去所有 d 不一致的假设

- 对 G 中每个与 d 不一致的假设 g

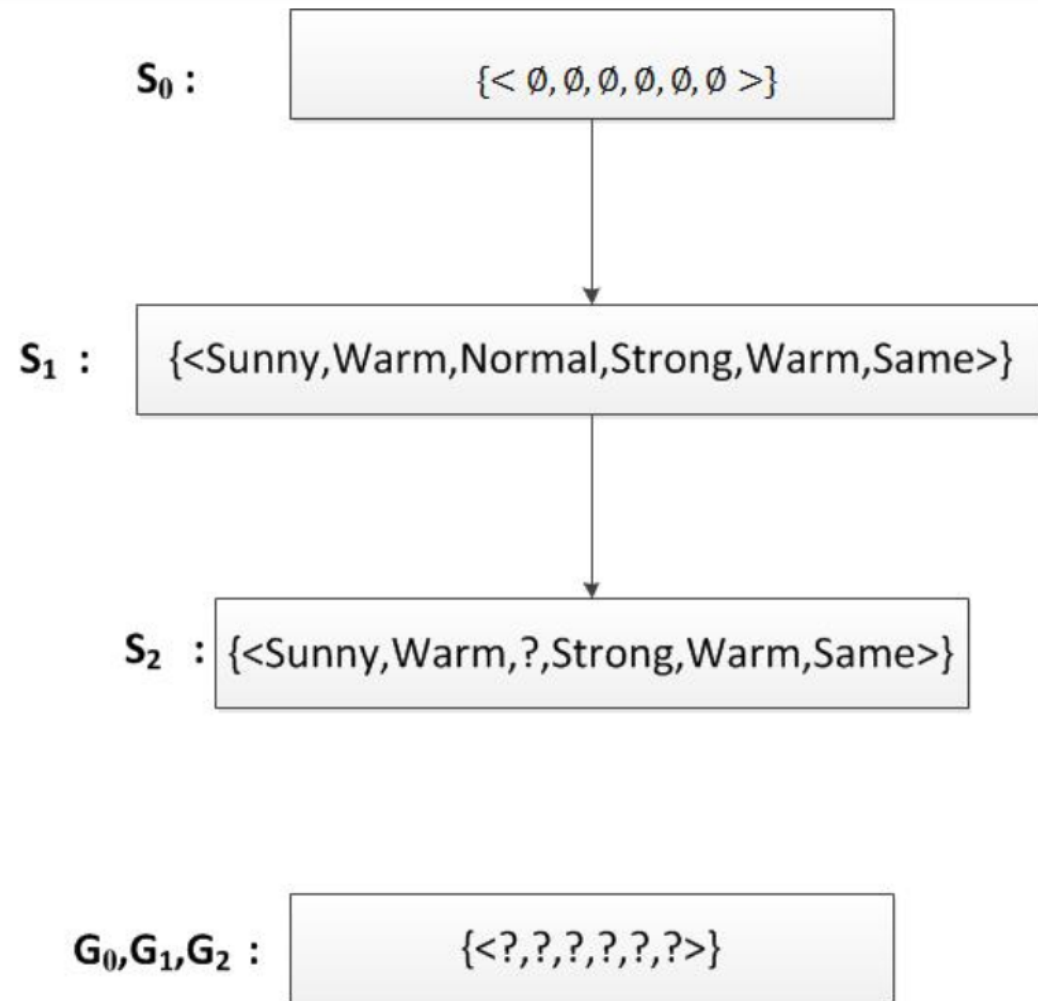
- 从 G 中移去 g

- 把 g 的所有的极小特殊化式 h 加入到 G 中，其中 h 满足

- h 与 d 一致，而且 S 的某个成员比 h 更特殊

- 从 G 中移去所有这样的假设：它比 G 中另一假设更特殊





▪ 分别看是否满足S, G

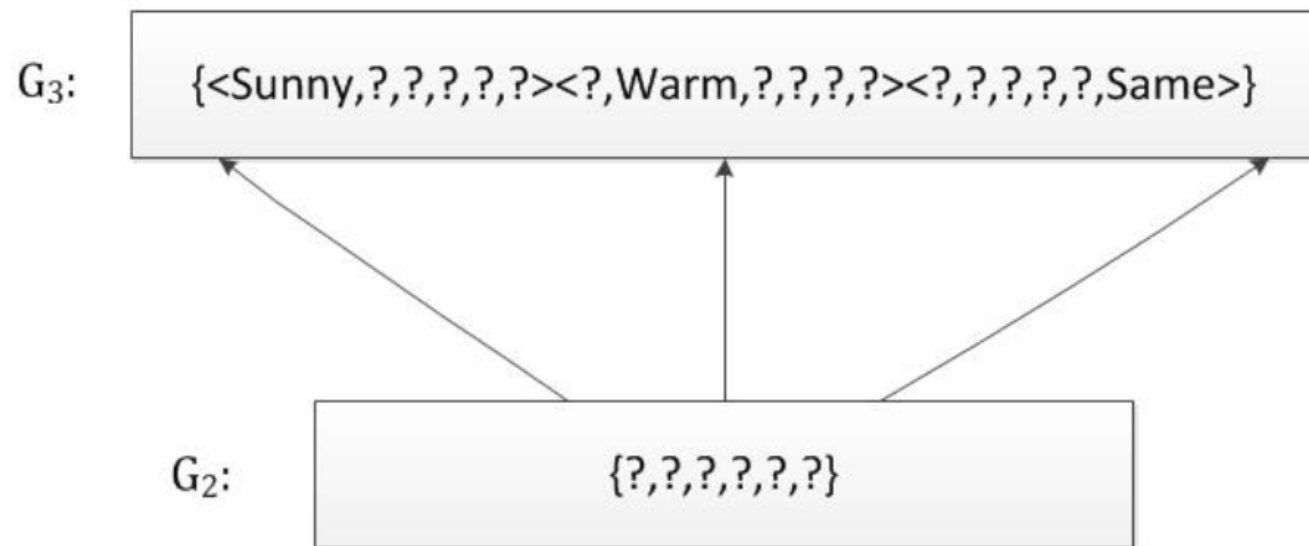
训练样例：

1. $\langle \text{Sunny}, \text{Warm}, \text{Normal}, \text{Strong}, \text{Warm}, \text{Same} \rangle, \text{EnjoySport} = \text{Yes}$

2. $\langle \text{Sunny}, \text{Warm}, \text{High}, \text{Strong}, \text{Warm}, \text{Same} \rangle, \text{EnjoySport} = \text{Yes}$



S_2, S_3 {<Sunny, Warm, ?, Strong, Warm, Same>}

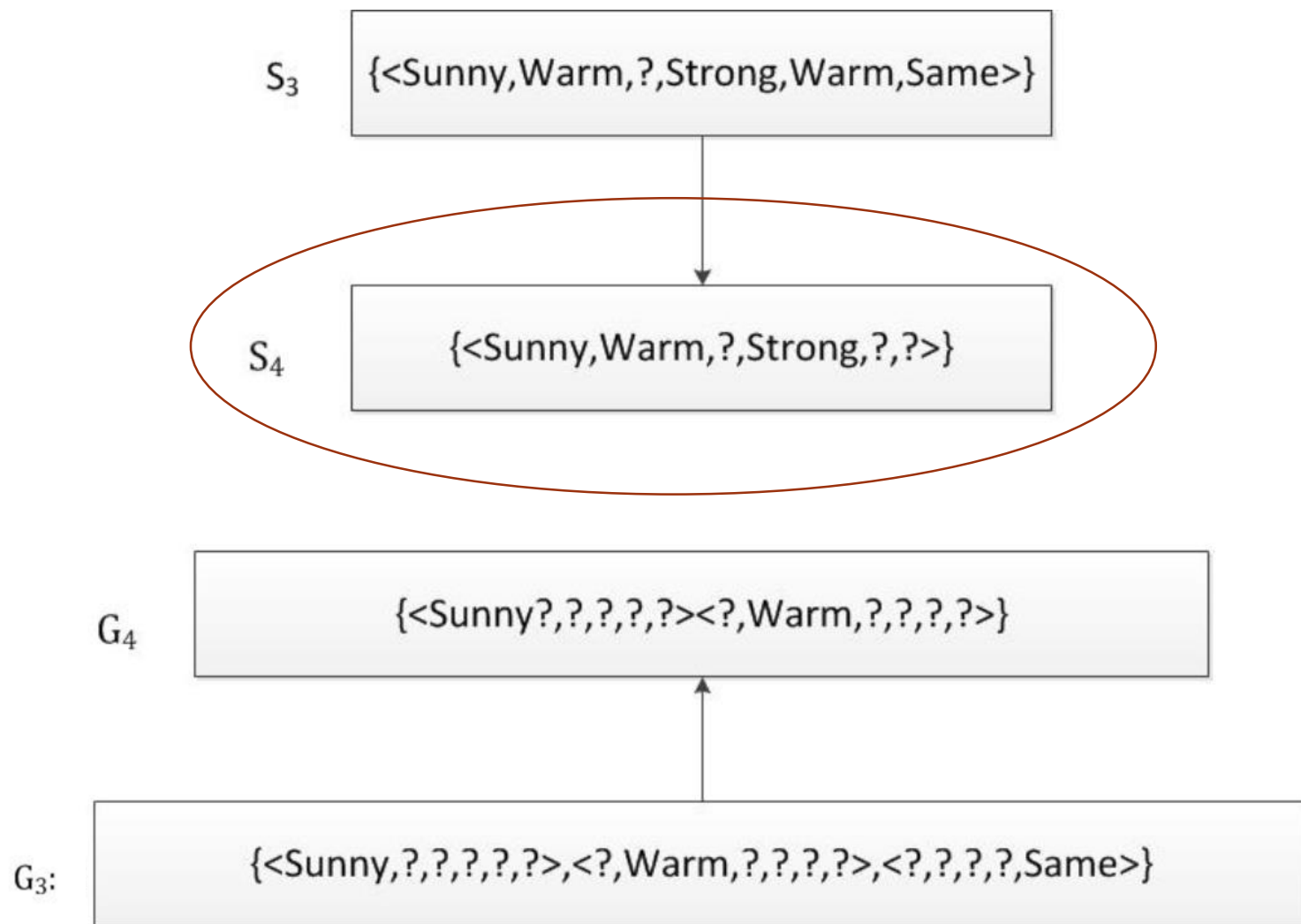


- 分别看是否满足S, G
- G_3 中6选3
- G_3 中有多个可选的极大一般假设

训练样例:

3.<Rainy, Cold, High, Strong, Warm, Change>, EnjoySport=No





- 分别看是否满足S, G
- 红框为最终结果

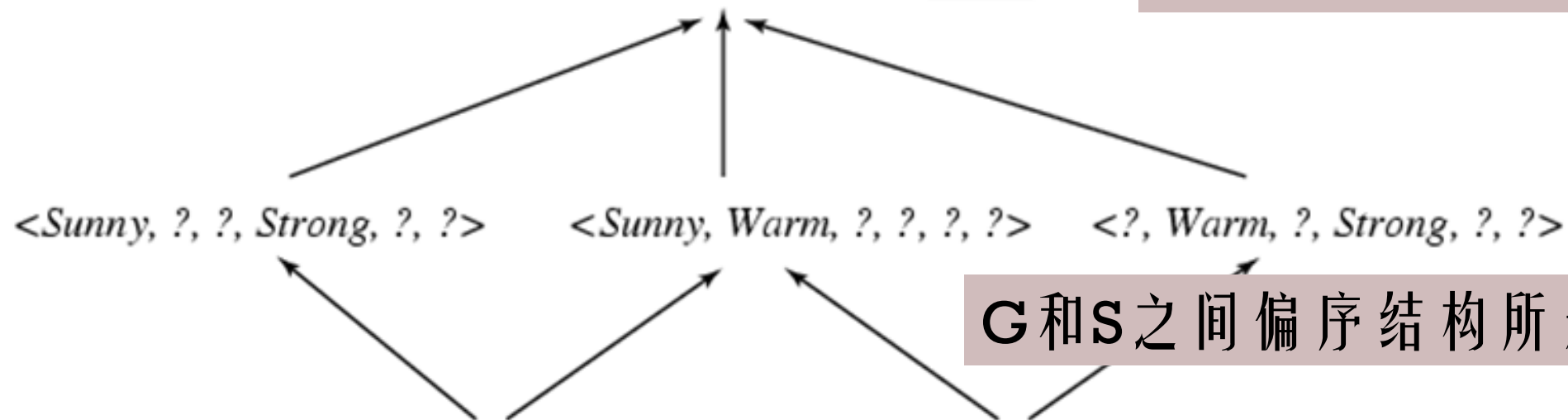
训练样例:

4.<Sunny,Warm,High,Storage,Cool,Change>,EnjoySport=Yes



S { <Sunny, Warm, ?, Strong, ?, ?> }

S中包含的假设集



G和S之间偏序结构所规定的假设

G { <Sunny, ?, ?, ?, ?, ?>, <?, Warm, ?, ?, ?, ?> }

G中包含的假设集



■ 变型空间表示定理

- 令 X 为一任意的实例集合， H 为 X 上定义的布尔假设的集合。令 $c: X \rightarrow \{0,1\}$ 为 X 上定义的任一目标概念，并令 D 为任一训练样例集合 $\{ \langle x, c(x) \rangle \}$ 。对所有的 X, H, c, D 以及定义好的 S 和 G ，变型空间表示如下：

$$VS_{H,D} = \{ h \in H \mid (\exists s \in S)(\exists g \in G)(g \geq_g h \geq_g s) \}$$

- 上述定理表明：

- 变型空间由 G 、 S 、 G 和 S 之间的偏序结构所规定的假设 h 组成



■ 候选消除算法讨论

- 候选消除算法输出与训练样例一致的所有假设的集合
- 候选消除算法在描述这一集合时不需要明确列举所有成员
- 利用 `more_general_than` 偏序结构，可以得到一个一致假设集合的简洁表示
- 候选消除算法的缺点：同 **Find-S** 一样，容错性能差



- 候选消除算法收敛到正确目标概念的条件
 - 训练样例中没有错误
 - H 中确实包含描述目标概念的正确假设



■ 一个有偏的假设空间

- 在EnjoySport这个例子中，假设空间限制为只包含属性值的合取（同时发生，交集）。（有偏）
- 这一限制，导致假设空间不能够表示最简单的析取（发生一件，并集）形式的目标概念。

Example	<i>Sky</i>	<i>AirTemp</i>	<i>Humidity</i>	<i>Wind</i>	<i>Water</i>	<i>Forecast</i>	<i>EnjoySport</i>
1	Sunny	Warm	Normal	Strong	Cool	Change	Yes
2	Cloudy	Warm	Normal	Strong	Cool	Change	Yes
3	Rainy	Warm	Normal	Strong	Cool	Change	No

“Sky=Sunny或Sky=Cloudy”



- 归纳推理的一个重要基本属性：
 - 学习器如果不对目标概念的形式做预先的假定，那么它从根本上无法对未见的样本（实例）进行分类。
- 归纳偏置（ **Inductive Bias** ）：
 - 对归纳学习进行的某种形式的预先假定



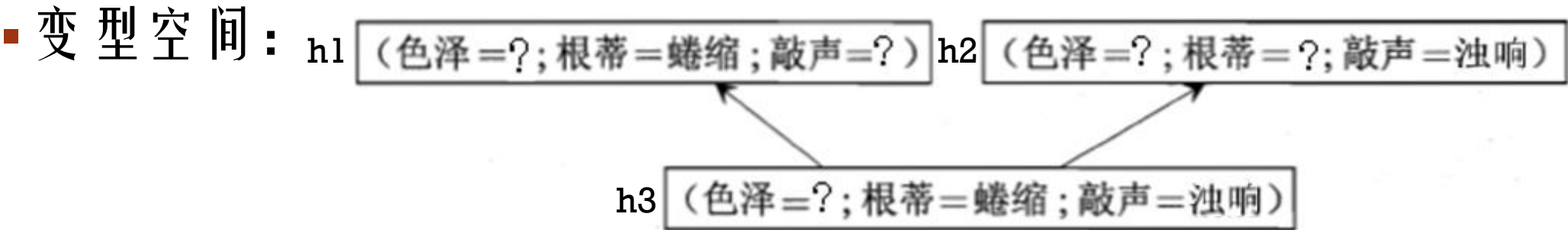
■ 归纳偏好：

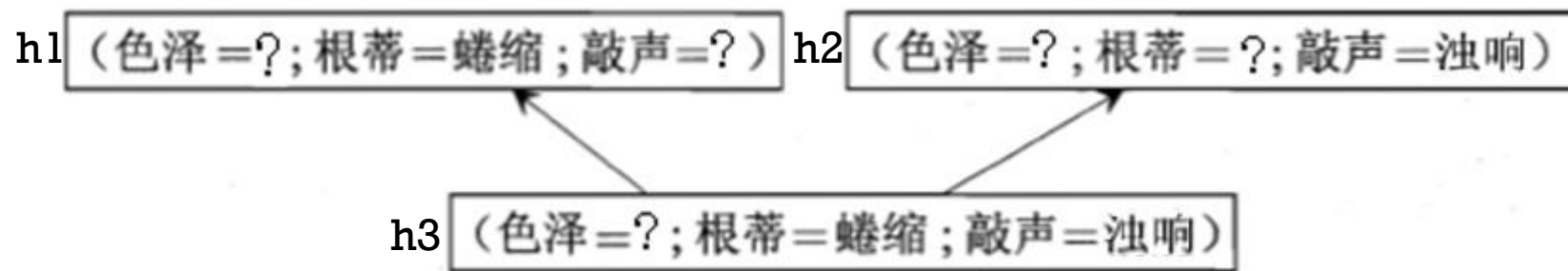
- 假如通过不同学习算法得到三个与训练集一致的假设，但是他们对应的模型在遇到相同的问题时，会产生不同的预测结果。那么，应该选择哪种模型？我们无法通过训练模型得知哪个模型“更好”。这时，学习算法本身的“偏好”就会起到决定性作用。机器学习算法在学习过程中对某种类型假设的偏好，称为：“归纳偏好”。



■ 西瓜数据集：学习概念“好瓜”

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否





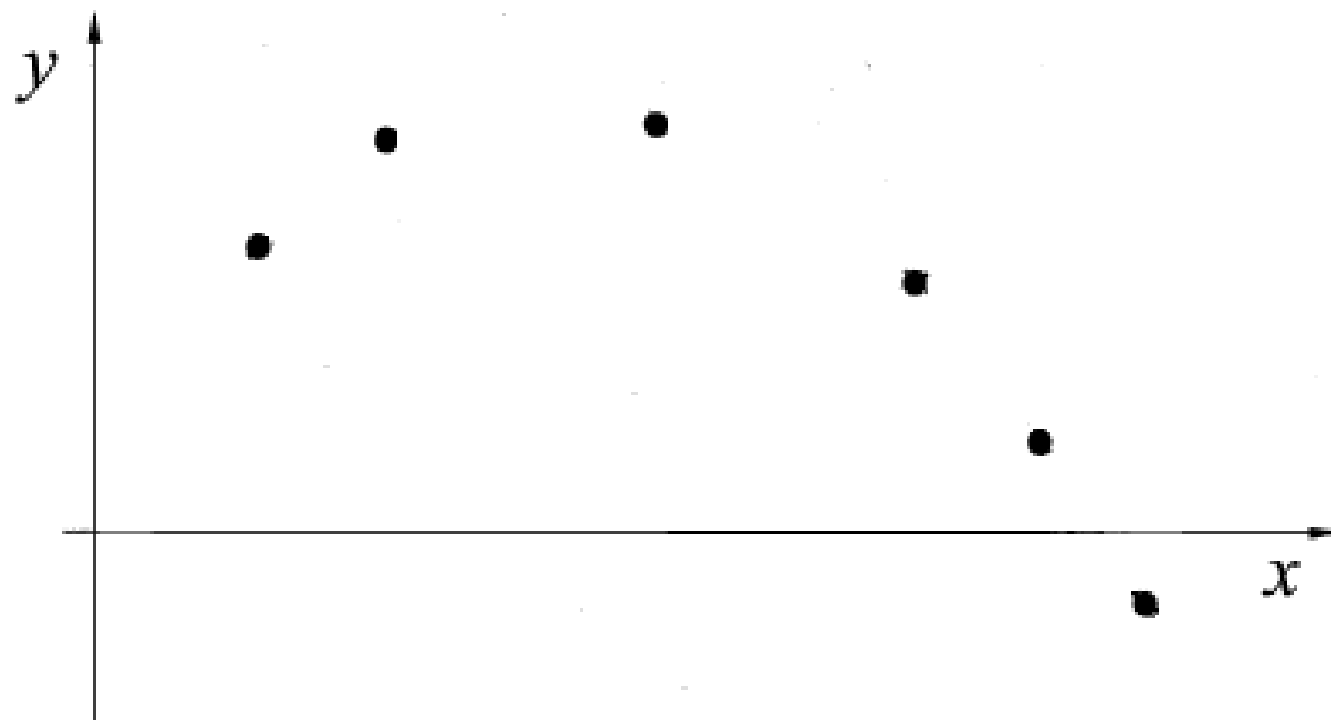
- 新瓜（好瓜）：色泽 = 青绿；根蒂 = 蜷缩；敲声 = 清脆
 - 算法偏好尽可能特殊的模型：否
 - 算法偏好尽可能一般的模型，且由于某种原因它更“相信”根蒂：是
 - 算法偏好多数表决：否

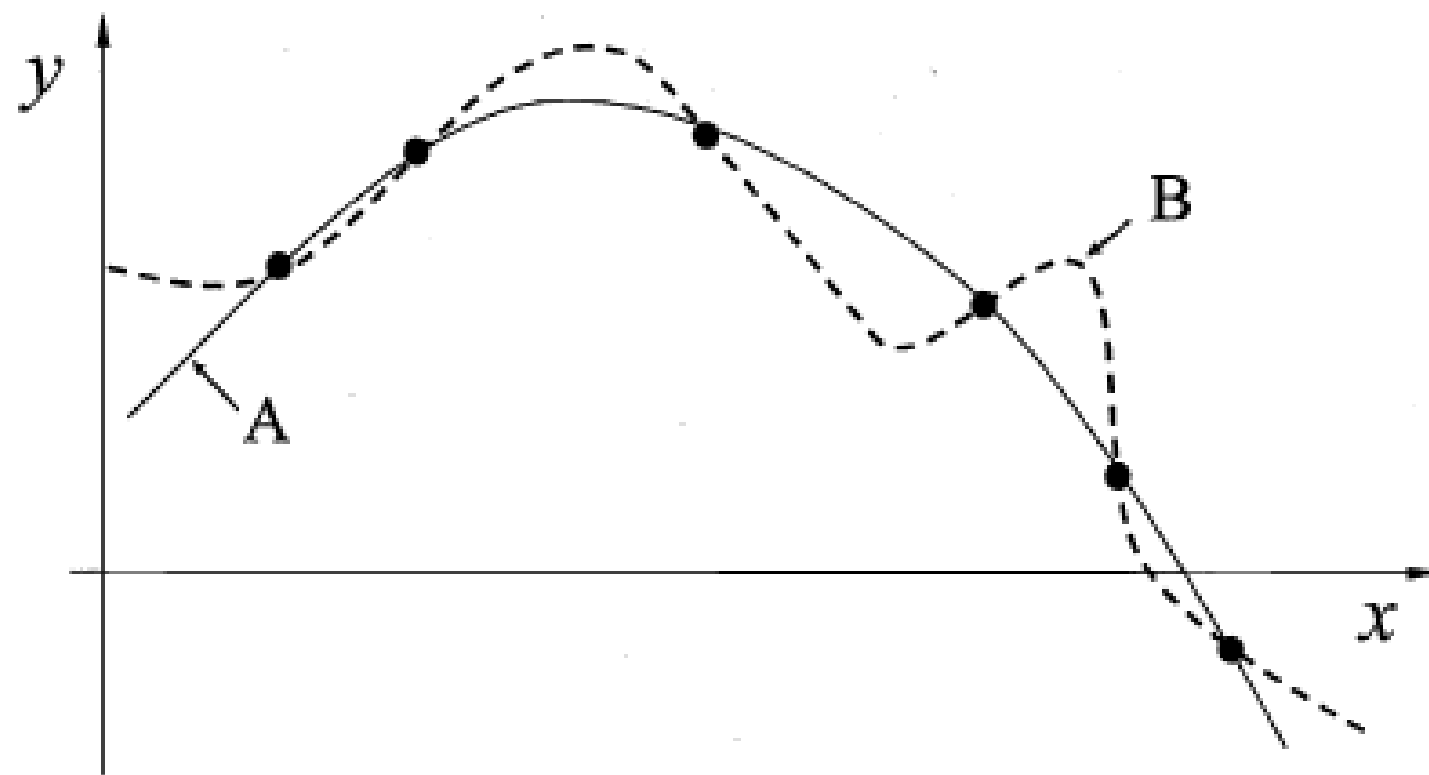


- 归纳偏好常表现为正则化项
- 可依据“奥卡姆剃刀”原理进行选择：若有多个假设与观察一致，则选择最简单的那个



■ 考察一个回归学习

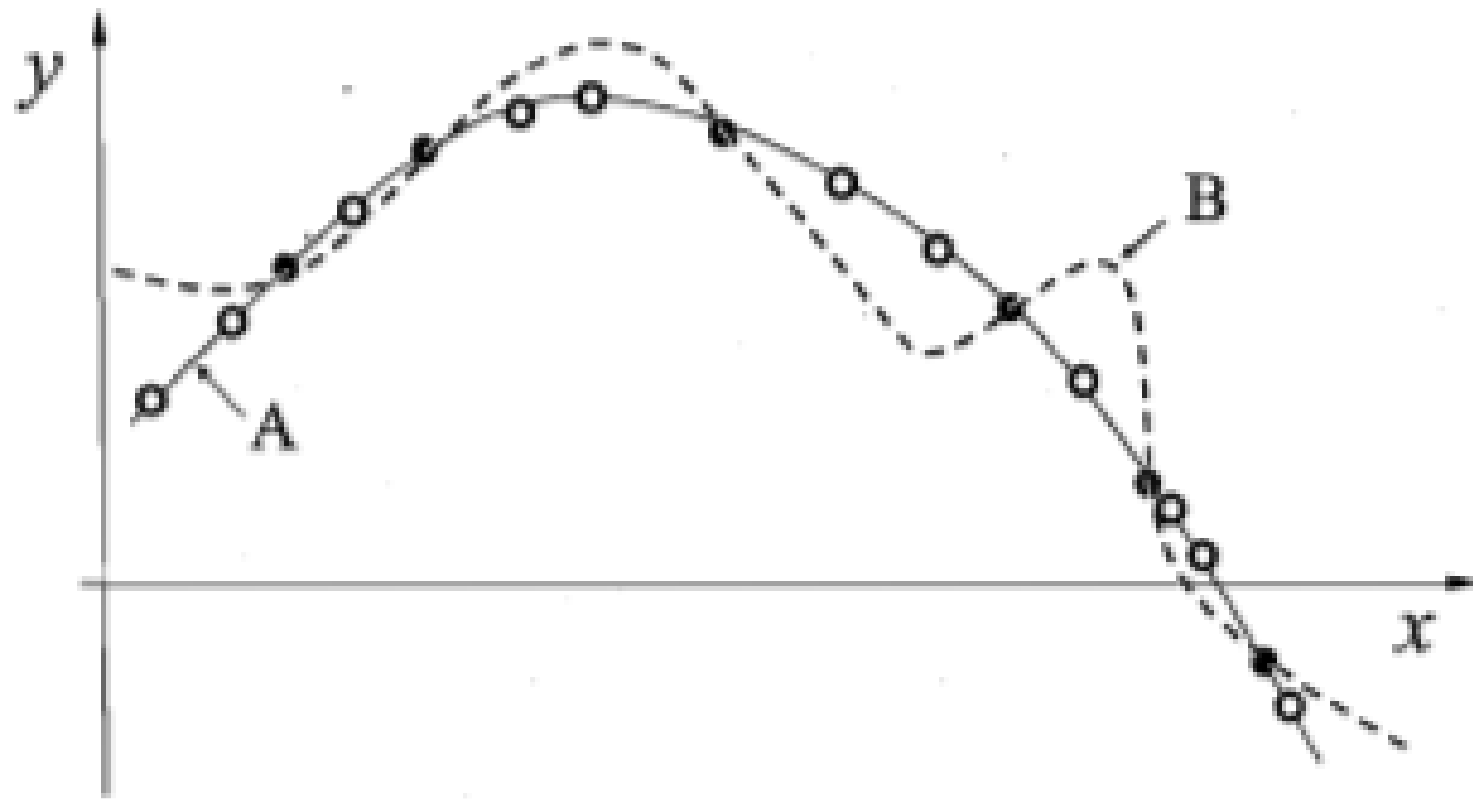




存在多条曲线与有限样本训练集一致

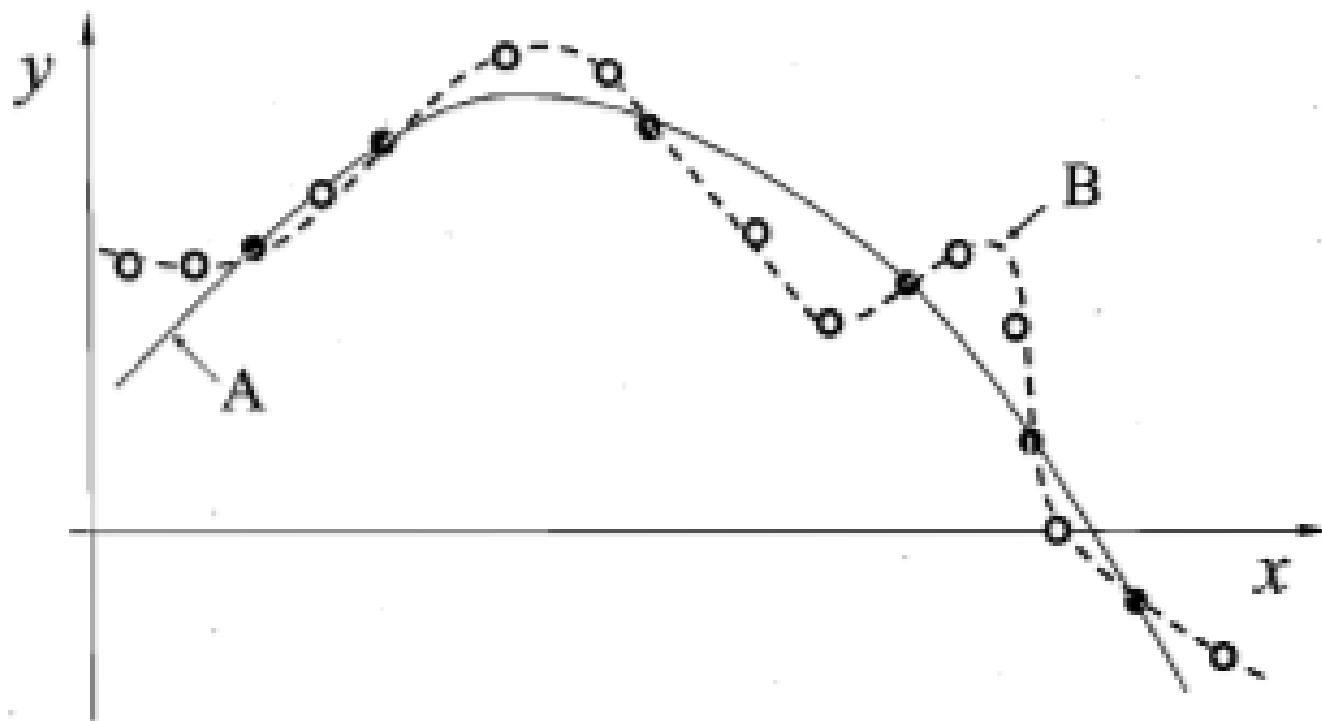
- 通过学习算法得到A、B两条拟合曲线





- 根据“奥卡姆剃刀”原理，A优于B





■ 但B优于A的情况也是完全可能存在的

对于一个学习算法fa，若它在某些问题上比学习算法fb好，则必然存在在另一些问题上，fb比fa好。这个结论对任何算法均成立。“**没有免费的午餐**”定理证实，无论学习算法fa多聪明、学习算法fb多笨拙，它们的期望性能竟然相同（训练集外误差）。



为简单起见, 假设样本空间 \mathcal{X} 和假设空间 \mathcal{H} 都是离散的. 令 $P(h|X, \mathcal{L}_a)$ 代表算法 \mathcal{L}_a 基于训练数据 X 产生假设 h 的概率, 再令 f 代表我们希望学习的真实目标函数. \mathcal{L}_a 的“训练集外误差”, 即 \mathcal{L}_a 在训练集之外的所有样本上的误差为

$$E_{ote}(\mathcal{L}_a|X, f) = \sum_h \sum_{\mathbf{x} \in \mathcal{X} - X} P(\mathbf{x}) \mathbb{I}(h(\mathbf{x}) \neq f(\mathbf{x})) P(h | X, \mathcal{L}_a) ,$$

其中 $\mathbb{I}(\cdot)$ 是指示函数, 若 \cdot 为真则取值 1, 否则取值 0.



若 f 均匀分布, 则有一半的 f 对 \mathbf{x} 的预测与 $h(\mathbf{x})$ 不一致.

对所有可能的 f 按均匀分布对误差求和, 有

$$\begin{aligned}\sum_f E_{ote}(\mathcal{L}_a | X, f) &= \sum_f \sum_h \sum_{\mathbf{x} \in \mathcal{X} - X} P(\mathbf{x}) \mathbb{I}(h(\mathbf{x}) \neq f(\mathbf{x})) P(h | X, \mathcal{L}_a) \\&= \sum_{\mathbf{x} \in \mathcal{X} - X} P(\mathbf{x}) \sum_h P(h | X, \mathcal{L}_a) \sum_f \mathbb{I}(h(\mathbf{x}) \neq f(\mathbf{x})) \\&= \sum_{\mathbf{x} \in \mathcal{X} - X} P(\mathbf{x}) \sum_h P(h | X, \mathcal{L}_a) \frac{1}{2} 2^{|\mathcal{X}|} \\&= \frac{1}{2} 2^{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X} - X} P(\mathbf{x}) \sum_h P(h | X, \mathcal{L}_a) \\&= 2^{|\mathcal{X}|-1} \sum_{\mathbf{x} \in \mathcal{X} - X} P(\mathbf{x}) \cdot 1.\end{aligned}$$

总误差竟然与学习算法无关!



对于任意两个学习算法 \mathcal{L}_a 和 \mathcal{L}_b ,

$$\sum_f E_{ote}(\mathcal{L}_a|X, f) = \sum_f E_{ote}(\mathcal{L}_b|X, f)$$

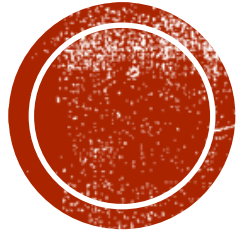
这就是“没有免费的午餐”定理 (No Free Lunch Theorem, 简称 NFL 定理)
[Wolpert, 1996; Wolpert and Macready, 1995].

- 优化算法的等价性
- 任何优化算法都不比穷举法好
- 为什么还要研究最优化和机器学习算法呢?



- **NFL**定理有一个重要前提：所有“问题”出现的机会相同、或**所有问题同等重要**。但实际情形并不是这样的。很多时候，我们**只关注自己正在试图解决的问题**，希望为它找到一个解决方案，至于这个解决方案在别的问题上是否为好方案，我们并不关心。
- 脱离具体问题，空泛的谈论“什么学习算法更好”
毫无意义
- 收敛速度





THE END !

