

# 机器学习

## Machine learning

## 第七章 降维与特征选择

### Feature Reduction & Selection

授课人：周晓飞  
zhouxiaofei@iie.ac.cn  
2024-11-15

课件放映 → PDF 视图 → 全屏模式

# 第七章 降维与特征选择

## 7.1 概述

## 7.2 特征选择

## 7.3 特征降维

# 第七章 降维与特征选择

## 7.1 概述

## 7.2 特征选择

## 7.3 特征提取

# 概述

## 内容概要

机器学习算法的**有效性和计算复杂度**是敏感于数据的**特征表达和维度**。

本章介绍数据的降维表示方法，主要包括**特征选择**和**特征提取**两种方法。

## 为什么要进行特征选择和提取？

### 例子 1. 两类问题：男性、女性

数据特征：身高、体重、音频、头发长短、**出生日期、家庭住址、籍贯、专业**

所有特征都有用吗？

## 为什么要进行特征选择和提取？

### 例子 2. 高维数据的稀疏情况

数据 1: 1, 0, 0, 0, 0, 0, 1, 3, 0, 0, 0, 0, 0, 0

数据 2: 1, 0, 0, 0, 0, 0, 3, 3, 0, 0, 0, 0, 0, 0

数据 3: 1, 0, 1, 0, 0, 0, 5, 3, 0, 2, 0, 0, 0, 0

样本中出现许多 0 值的属性，这些属性特征有用吗？

## 为什么要进行特征选择和提取？

### 例子 2. 高维数据的稀疏情况

数据 1: 1, 0, 0, 0, 0, 0, 1, 3, 0, 0, 0, 0, 0, 0  
数据 2: 1, 0, 0, 0, 0, 0, 3, 3, 0, 0, 0, 0, 0, 0  
数据 3: 1, 0, 1, 0, 0, 0, 5, 3, 0, 2, 0, 0, 0, 0

样本中出现许多 0 值的属性，这些属性特征有用吗？

## 为什么要进行特征选择和提取?

### 例子 3. 降低数据尺寸





## 为什么要进行特征选择和提取？

### 特征降维的意义

#### (1) 数据压缩：

简化数据表示，加快数据通信传输、节省存储资源、...

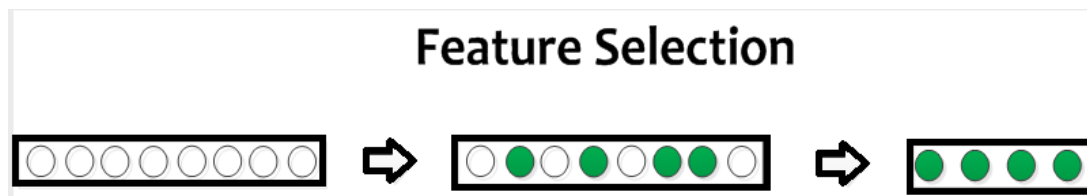
#### (2) 学习算法效率：

- 计算上，简化计算，加快速度
- 性能上，提升精确度
- 可理解性，发现数据的潜在本质特征

## 特征选择和特征提取的区别？

特征选择：

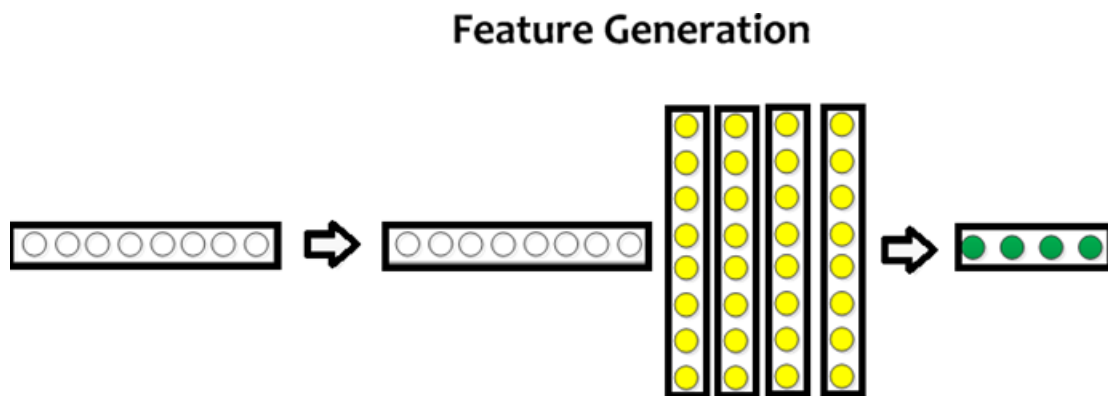
从  $D$  个特征中选择  $d$  个，来表达模式。



## 特征选择和特征提取的区别？

### 特征提取：

采用**特征变换**的方法，生成  $d$  个新的特征



# 第七章 降维与特征选择

7.1 概述

7.2 特征选择

7.3 特征提取

# 特征选择

## 特征选择框架

### 特征选择问题:

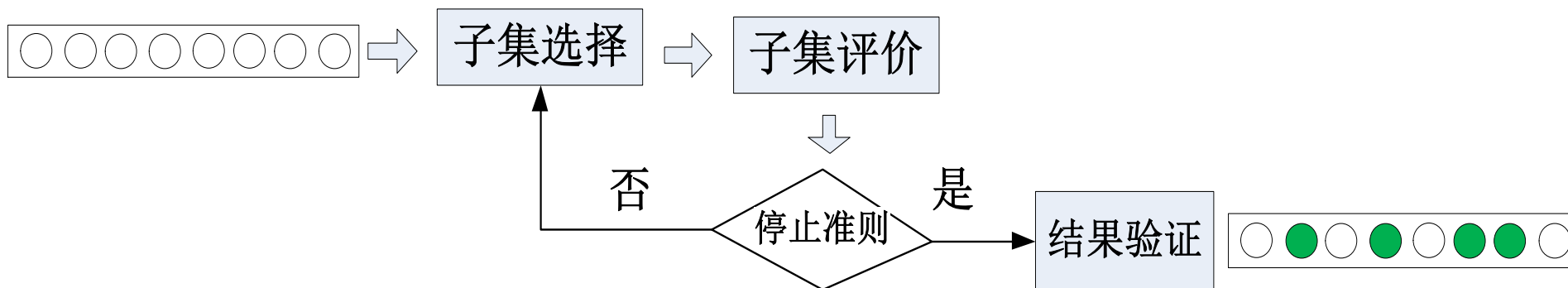
从  $D$  维特征中选择  $d$  维 ( $d < D$ ) 特征子集

- 使数据的压缩率高
- 使学习机预测性能最佳
- 使学习机学习速度加快

# 特征选择

## 特征选择框架

特征选择的处理过程：



- 特征子集生成
- 特征评价准则
- 特征选择的框架

# 特征选择

## 特征子集生成

特征子集生成问题:

D 维特征中, 选择 d 维 ( $d < D$ ) 特征子集, 子集个数?

$$C_D^d = \frac{D!}{(D-d)!d!}$$

例如:  $X=(x_1, x_2, x_3, x_4, x_5)$  降到 2 维

可能的特征表示:

$(x_1, x_2)$ 、 $(x_1, x_3)$ 、 $(x_1, x_4)$ 、 $(x_1, x_5)$ 、 $(x_2, x_3)$ 、  
 $(x_2, x_4)$ 、 $(x_2, x_5)$ 、 $(x_3, x_4)$ 、 $(x_3, x_5)$ 、 $(x_4, x_5)$

# 特征选择

## 特征子集生成

### 1. 穷举（最优子集搜索）

计算特征的所有可能组合，并逐一评价。

选择的特征数为  $d$ , 可能的组合数:

$$\text{组合数 } C_D^d = \frac{D!}{(D-d)!d!}$$

$$D = 100, d = 2, C = 4950$$

$$D = 100, d = 3, C = 161700$$

$$D = 100, d = 10, C = 1.73103 \times 10^{13}$$

$$D = 100, d = 50, C = 1.00891 \times 10^{29}$$

$$D = 1000, d = 2, C = 499500$$

$$D = 10000, d = 2, C = 4.9995 \times 10^7$$

$d=1 \sim D$ , 总的穷举总数:

$$\sum_{d=1}^D C_D^d = \sum_{d=1}^D \frac{D!}{(D-d)!d!} = 2^D - 1$$

计算量大，适合于  $D$  较小的情况。



# 特征选择

## 特征子集生成

### 2. 单独最优特征组合

对每个特征分别评估，找前  $d$  个单独最优特征

- 方法描述：

对特征集  $X = \{x_1, x_2, \dots, x_i, \dots, x_D\}$

根据给定的样本，分别计算  $J(x_1)$ 、 $J(x_2)$ 、 $\dots$ 、 $J(x_D)$

按最优顺序，排序  $J(x^*_1) > J(x^*_2) > \dots > J(x^*_D)$

选择前  $d$  个特征，作为选择特征集

优点：算法简单

缺点：没有考虑特征之间的关系，存在特征冗余

# 特征选择

## 特征子集生成

### 3. SFS (Sequential forward selection, 前向序贯)

每次加入一个特征，该特征使得新的特征组合最优。

每次寻优后特征集：

$$X_d = X_{d1} + x_j$$

$$J_j = \max_{x_j} J(X_{d1} + x_j)$$

其中， $x_j \notin X_{d1}$

特点：一旦增加，无法删除

# 特征选择

## 特征子集生成

### 4. GSFS (广义 SFS)

每次加入  $k$  个特征，使加入特征后的组合最优。

每次寻优后特征集：

$$X_d = X_{d1} + \{x_j\}_k$$

$$J_j = \max_{\{x_j\}_k} J(X_{d1} + \{x_j\}_k)$$

$\{x_j\}_k$  表示  $k$  个样本组成的集合， $\{x_j\}_k \notin X_{d1}$ 。

计算量比 SFS 大

# 特征选择

## 特征子集生成

### 5. SBS (Sequential backward selection, 后向序贯)

每次减掉一个特征，使剩余特征组合最优。

每次寻优后特征集：

$$X_d = X_{d1} - x_j$$

$$J_j = \max_{x_j} J(X_{d1} - x_j)$$

其中， $x_j \in X_{d1}$

特点：一旦删除，无法增加

# 特征选择

## 特征子集生成

### 6. GSBS (广义 SBS):

每次减  $k$  个特征，使剩余特征组合最优。

每次寻优后特征集：

$$X_d = X_{d1} - \{x_j\}_k$$

$$J_j = \max_{\{x_j\}_k} J(X_{d1} - \{x_j\}_k)$$

其中， $\{x_j\}_k \in X_{d1}$ 。

计算量比 SBS 大

# 特征选择

## 特征子集生成

### 7. L-R 法 (增加 L 个, 减 R 个)

每次增加 L 个再减 R 个, ( $L > R$ );

或减 R 个增加 L 个 ( $L < R$ )

先 L-后 R : 每次寻优后特征集:

$$X_d = X_{d1} + \{x_j\}_L - \{x_i\}_R$$

$$J_{j1} = \max_{\{x_j\}_L} J(X_{d1} + \{x_j\}_L)$$

$$J_{j2} = \max_{\{x_i\}_R} J(X_{d1} + \{x_j\}_L - \{x_i\}_R)$$

其中,  $\{x_j\}_L \in X_{d1}$ ,  $\{x_i\}_R \in X_{d1} + \{x_j\}_L$ .

# 特征选择

## 特征子集生成

先 R-后 L，每次寻优后特征集：

$$X_d = X_{d1} + \{x_i\}_R - \{x_j\}_L$$

$$J_{j1} = \max_{\{x_i\}_R} J(X_{d1} + \{x_i\}_R)$$

$$J_{j2} = \max_{\{x_i\}_R} J(X_{d1} + \{x_i\}_R - \{x_j\}_L)$$

其中， $\{x_i\}_R \in X_{d1}$ ， $\{x_j\}_L \in X_{d1} + \{x_i\}_R$ 。

# 特征选择

## 特征子集生成

### 8. 广义的 L-R (ZL, ZR)

增 L 和减 R 分 Z 步进行

先 L-后 R, 每次寻优后特征集:

$$X_d = X_{d1} + \{x_{j1}\}_{L/z} + \dots + \{x_{jk}\}_{L/z} - \{x_{i1}\}_{R/z} - \dots - \{x_{il}\}_{R/z}$$

$$J_{j1} = \max_{\{x_{j1}\}_{L/z}} J(X_{d1} + \{x_{j1}\}_{L/z}), \dots$$

$$J_{jz} = \max_{\{x_{j1}\}_{L/z}} J(X_{d1} + \{x_{j1}\}_{L/z} \dots + \{x_{jz}\}_{L/z})$$

$$J_{i1} = \max_{\{x_{i1}\}_{R/z}} J(X_{d1} + \{x_{j1}\}_{L/z} \dots + \{x_{jz}\}_{L/z} - \{x_{i1}\}_{R/z}), \dots$$

$$J_{iz} = \max_{\{x_{i1}\}_{R/z}} J(X_{d1} + \{x_{j1}\}_{L/z} \dots + \{x_{jz}\}_{L/z} - \{x_{i1}\}_{R/z} - \dots - \{x_{iz}\}_{R/z})$$



# 特征选择

## 特征评价准则

### (1) 可分性度量:

在选择的特征集下，采用类别可分性的程度，评价特征选择的好与坏。  
常用于 Filter 框架下。

### (2) 学习算法精度的度量:

在选择的特征集下，通过学习算法的精确度，评价特征选择的好与坏。  
常用于 wrapper 框架下。

# 特征选择

## 特征评价准则

### (1) 可分性度量:

在选择的特征集下，采用类别可分性的程度，评价特征选择的好与坏。  
常用于 Filter 框架下。

评价准则：距离准则、概率可分、熵可分准则。

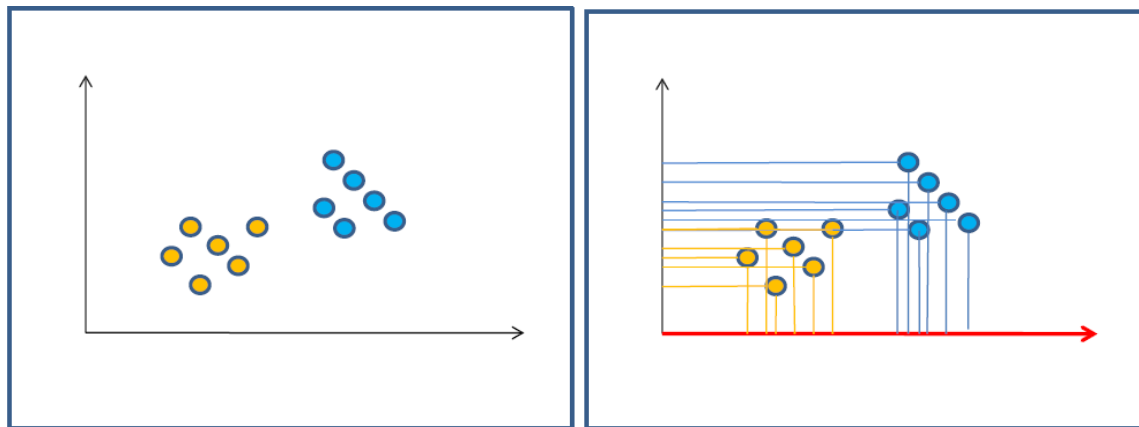
### (2) 学习算法精度的度量:

在选择的特征集下，通过学习算法的精确度，评价特征选择的好与坏。  
常用于 wrapper 框架下。

# 特征选择

## 特征评价准则 1. 基于距离的可分性判据

通常依赖于**类内类间**的距离度量（可回顾第三章）；（前提是数据具有类别标签）  
可分性评估是在选择的特征子集维度上计算数据统计量。



### 距离的可分性判据的特点:

- 容易理解和实现;
- 与错误率无直接关系, 不敏感于数据交叠情况;
- 常用于 Filter 特征选择框架下。

# 特征选择

## 特征评价准则 1. 基于距离的可分性判据

- 常用的数据统计量

(1) 类内散度矩阵

$$S_i = \frac{1}{n} \sum_{j=1}^n (\mathbf{x}'_j - \mathbf{m}_i)(\mathbf{x}'_j - \mathbf{m}_i)^T$$

$\mathbf{x}'_j$  是特征选择后的向量,  $\mathbf{m}_i$  是特征选择后第  $i$  类的平均

(2) 总类内离散度矩阵

$$S_w = \sum_{i=1}^c P_i S_i$$

(3) 类间散度矩阵

$$S_b = \sum_{i=1}^c (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$$

# 特征选择

## 特征评价准则 1. 基于距离的可分性判据

- 常用的可分性判据

对一维特征可采用：（两类）  $J_f = \frac{(m_1 - m_2)^2}{S_1^2 + S_2^2}$  ,      （多类）  $J_f = \frac{S_b}{S_w}$

对多维选择特征可采用：

$$J_1 = \text{tr}(S_w + S_b)$$

$$J_2 = \text{tr}(S_w^{-1} S_b)$$

$$J_3 = \ln \frac{|S_b|}{|S_w|}$$

$$J_4 = \frac{\text{tr}(S_b)}{\text{tr}(S_w)}$$

$$J_5 = \frac{|S_b - S_w|}{|S_w|}$$

## 特征评价准则 2. 基于概率分布的可分性判据

从类别概率密度的角度，讨论两个类别的交叠程度。

- 密度函数之间的距离：

$$J(.) = \int g[p(x|\omega_1), p(x|\omega_2), p_1, p_2] dx$$

满足三个条件：

1.  $J(.) \geq 0$
2. 若  $p(x|\omega_1)p(x|\omega_2) = 0, \forall x$

$$J = J_{\max} \quad \text{完全不重叠}$$

3. 若  $p(x|\omega_1) = p(x|\omega_2), \forall x$

$$J = 0 \quad \text{完全重叠}$$

# 特征选择

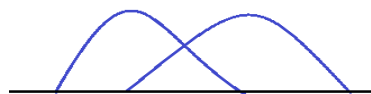
## 特征评价准则 2. 基于概率分布的可分性判据

可分

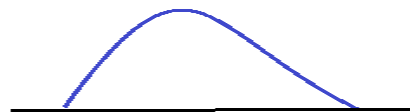


$$p(x|\omega_1)p(x|\omega_2)=0, \forall x$$

不可分



完全重叠



$$p(x|\omega_1)=p(x|\omega_2), \forall x$$

## 特征评价准则 2. 基于概率分布的可分性判据

- 常见的概率距离准则 (J)

(1) Bhattacharyya 距离:

$$J = -\ln \int [p(x|\omega_1)p(x|\omega_2)]^{1/2} dx$$

两类完全重合时,  $J=0$

两类完全不交时,  $J = \text{无穷大}$



## 特征评价准则 2. 基于概率分布的可分性判据

### (2) Chernoff:

$$J = -\ln \int p^s(x|\omega_1) p^{1-s}(x|\omega_2) dx$$

当  $s=0.5$  时, 与 Bhattacharyya 距离相同

### (3) Matusita:

$$J = \left[ \int \left( \sqrt{p(x|\omega_1)} - \sqrt{p(x|\omega_2)} \right)^2 dx \right]^{1/2}$$

## 特征评价准则 2. 基于概率分布的可分性判据

### (4) Patrick-Fisher

$$J = \left[ \int_x (p(x|\omega_1)P_1 - p(x|\omega_2)P_2)^2 dx \right]^{1/2}$$

### (5) Lissack-Fu

$$J = \int_x |p(x|\omega_1) - p(x|\omega_2)|^s p(x)^{1-s} dx$$

## 特征评价准则 2. 基于概率分布的可分性判据

### (6) Komolgorov

$$J = \int_x |p(x|\omega_1) - p(x|\omega_2)| dx$$

### (7) 散度

$$J = \int_x [p(x|\omega_1) - p(x|\omega_2)] \ln \frac{p(x|\omega_1)}{p(x|\omega_2)} dx$$

# 特征选择

## 特征评价准则 3. 熵可分性判据

- 互信息

$$I(X;Y) = \sum_{x \in S_x} \sum_{y \in S_y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

互信息：两个变量之间相互依存关系的强弱

- 条件互信息

$$I(X;Y | Z) = \sum_{x \in S_x} \sum_{y \in S_y} \sum_{z \in S_z} p(x, y, z) \log \frac{p(x, y | z)}{p(x | z)p(y | z)}$$

在 Z 已知的条件下，X 和 Y 的统计依存度

# 特征选择

## 特征选择方法

**特征选择通常包括三种方案：Filter、Wrapper、Embedded**

- **Filtering approaches**
  - Filter out features with small predictive potential
  - done before classification; typically uses univariate analysis
- **Wrapper approaches**
  - Select features that directly optimize the accuracy of the multivariate classifier
- **Embedded methods**
  - Feature selection and learning closely tied in the method

# 特征选择

## 特征选择方法---1. Filter 方法

- 不依赖于学习算法（如分类器）的结果，直接由数据构建评估函数，对选择的特征子集进行评估。
- 通常方法：根据特征评价准则进行评估，选择最优的特征子集。  
评价准则：距离准则、概率可分、熵可分准则。

**优点：**计算复杂度低，效率高。

**缺点：**选择的特征之间存在冗余信息。

# 特征选择

## 特征选择方法---1. Filter 方法

### 例 1: Filter 框架 + 特征子集生成 + 距离评价

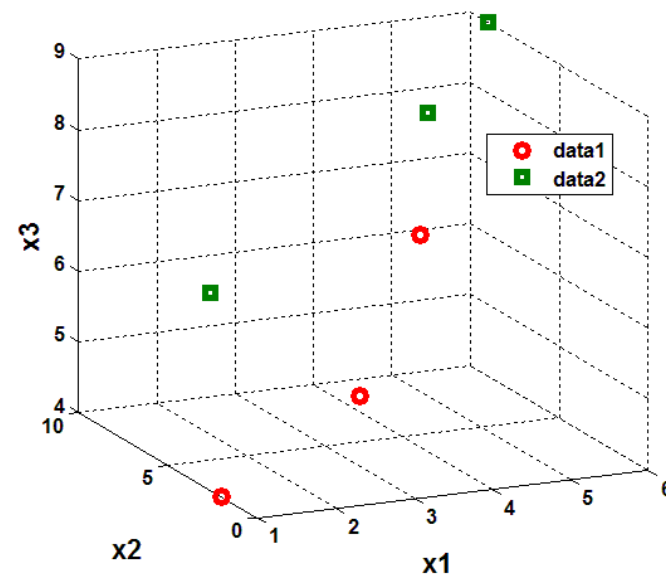
两类 3 维数据（每类各 3 个）每行是一个样本

W1 =

1	2	4
3	3	5
4	4	7

W2 =

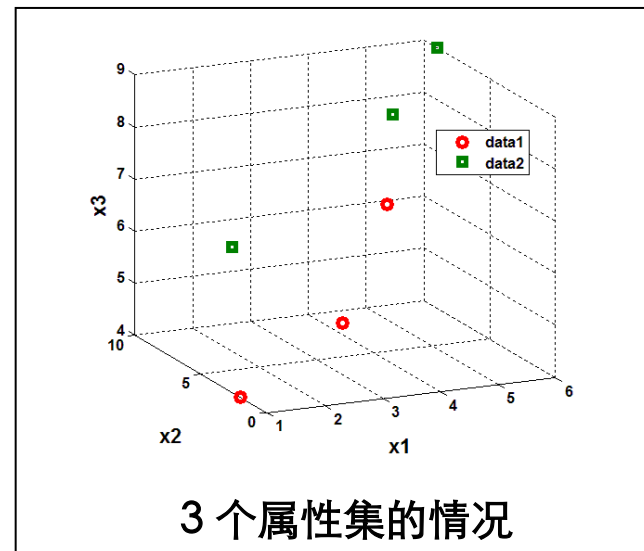
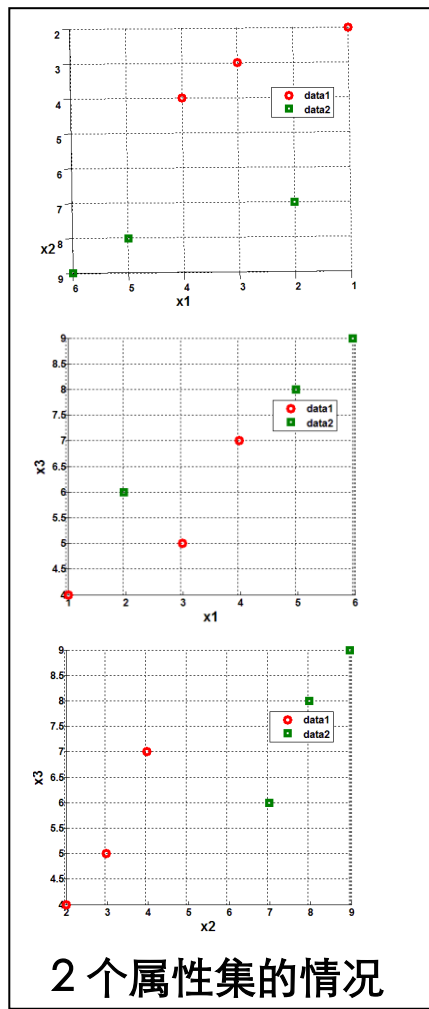
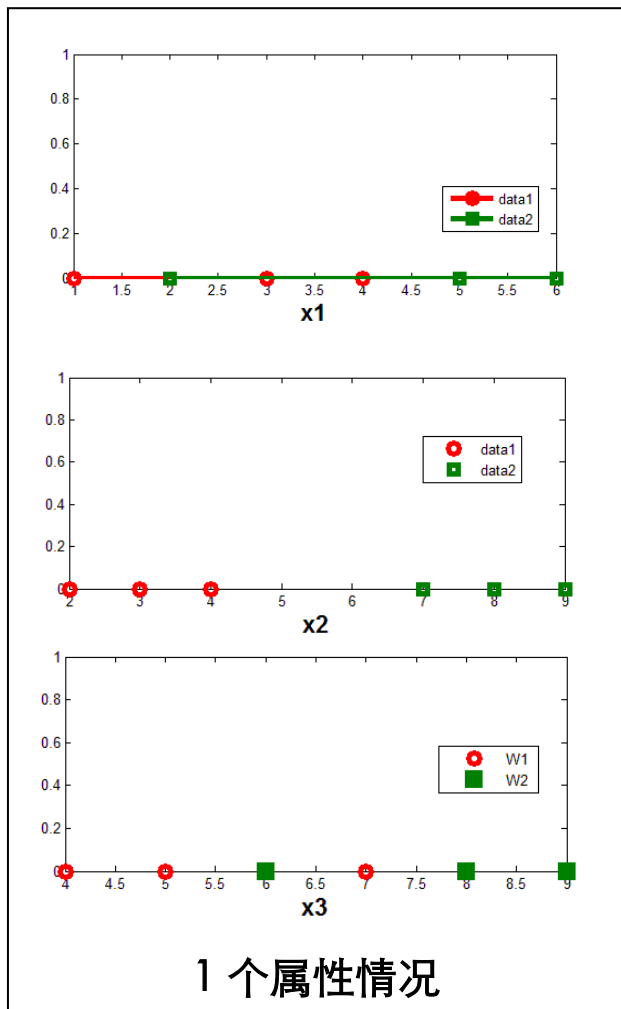
2	7	6
5	8	8
6	9	9



W1 红色, W2 绿色

# 特征选择

## 特征选择方法---1. Filter 方法





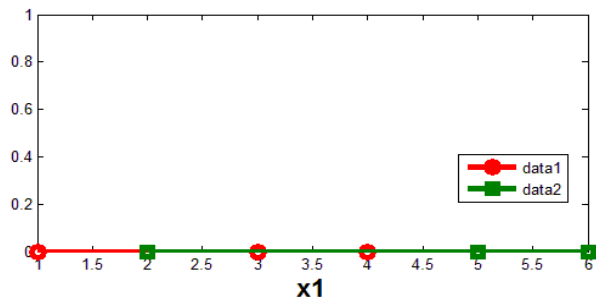
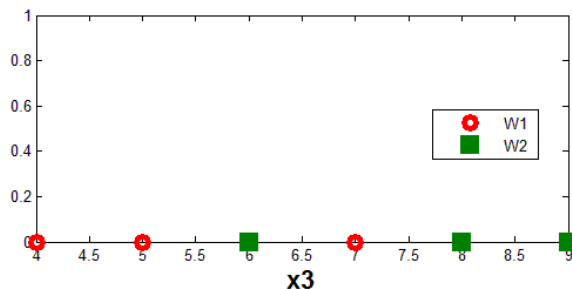
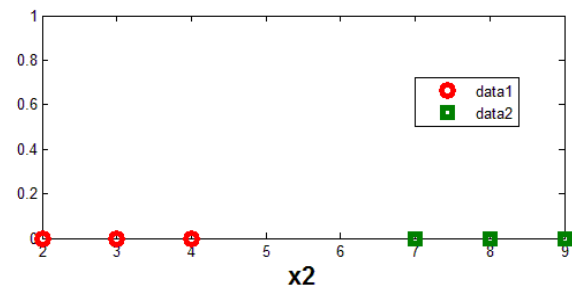
# 特征选择

## 特征选择方法---1. Filter 方法

### 选择 1 维特征时

每个特征计算  $J_f = \frac{(m_1 - m_2)^2}{S_1^2 + S_2^2}$ , 并排序

结果:  $J(x_2) > J(x_3) > J(x_1)$



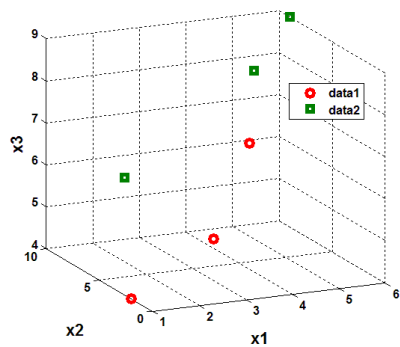
# 特征选择

## 特征选择方法---1. Filter 方法

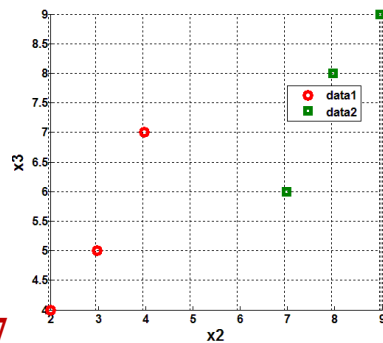
### 选择最优子集

每个子集计算

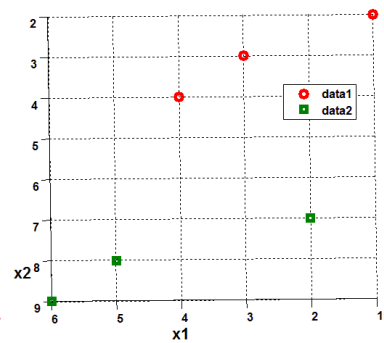
$$J_2 = \text{tr}(S_w^{-1} S_b)$$



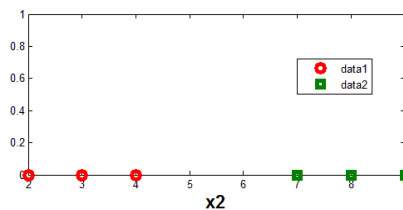
53.2917



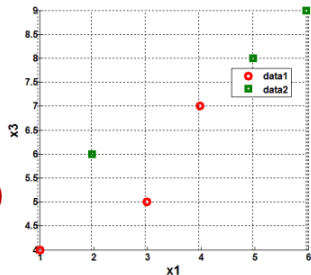
43.1667



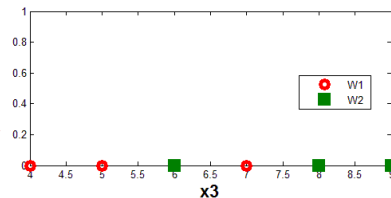
26.2821



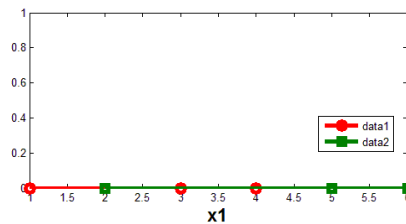
3.1250



0.7292



0.2917



0.1042

# 特征选择

## 特征选择方法---1. Filter 方法

### 例 2: 经典特征选择算法: Relief 算法 (1992)

(1) 两个类别的样本集, 分别为  $S^+$  和  $S^-$ ; 初始化权重向量  $w=(0,0,...,0)$ , 该向量用来衡量特征的重要性。

(2) For  $i=1\sim m$  (循环  $m$  次)

- 随机抽取样本  $x$ , 分别计算  $S^+$  和  $S^-$  中离  $x$  最近的点:

$$z^+ = \arg \min_{z_i \in S^+} \|x - z_i\|^2, \quad z^- = \arg \min_{z_i \in S^-} \|x - z_i\|^2$$

如果  $x \in S^+$ ,  $hit = z^+$ ,  $miss = z^-$ ;

如果  $x \in S^-$ ,  $hit = z^-$ ,  $miss = z^+$ ;

- 更新权重, For  $\forall w_i$ ,  
$$w_i = w_i - d(x_i, hit_i)^2 + d(x_i, miss_i)^2$$

分析: 如果  $d(x_i, hit_i)^2 < d(x_i, miss_i)^2$ , 该特征支持 “ $x$  与  $hit$  同类”, 则  $w_i$  变大;

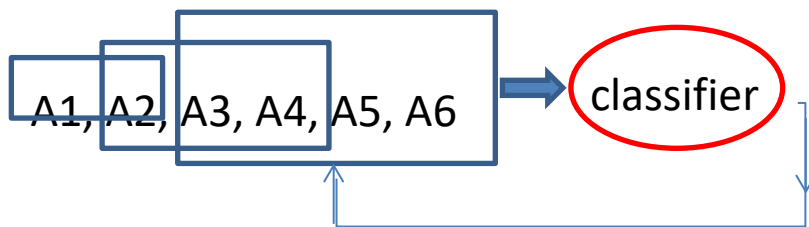
如果  $d(x_i, hit_i)^2 > d(x_i, miss_i)^2$ , 该特征不支持 “ $x$  与  $hit$  同类”, 则  $w_i$  变小;

(3) 特征选择: 设定阈值  $t$ , 如果  $w_i > t$ , 则特征  $f_i$  被选择。

# 特征选择

## 特征选择方法---2. Wrapper 方法

**原理：**通过学习算法（如分类器），对选择的特征子集进行评估。



原理图

**优点：**选择的特征可以支持学习算法。

**缺点：**算法的计算复杂度高。

# 特征选择

## 特征选择方法---2. Wrapper 方法

### 算法举例：LVW方法

#### 停止条件：错误率

LVW (Las Vegas Wrapper) [Liu and 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  $A^*$ .
```

---

# 特征选择

## 特征选择方法---3. Embedded 方法

**原理：**特征选择过程在学习算法中完成，目标是完成学习过程。

**特点：**不是专门的特征选择过程

**缺点：**计算复杂度高。

# 特征选择

## 特征选择方法---3. Embedded 方法

岭回归：引入正则项，避免过拟合

$$\min_{\mathbf{w}} \sum_{i=1}^m (y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_2^2$$

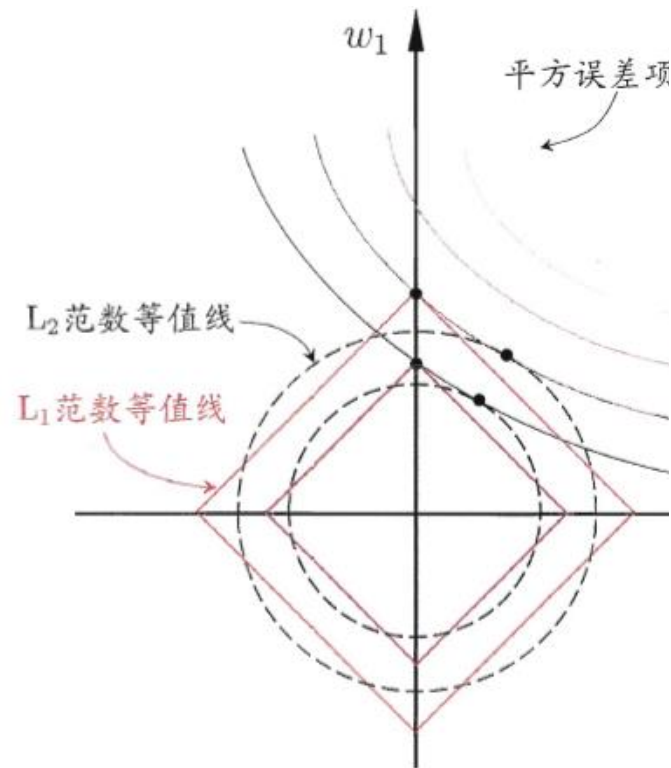
**LASSO** (Least Absolute Shrinkage and Selection Operator),

目标优化的同时，学习具有稀疏性的特征：

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

采用近端梯度下降 (Proximal Gradient Descent, PGD)。

(请参看《机器学习》)



## 参考文献

1. 周志华，机器学习，清华大学出版社，2016.
2. Duda, R.O. et al. Pattern classification. 2nd, 2003.
3. 边肇祺，张学工等编著，模式识别(第二版)，清华大学，1999。
4. Chris Bishop. Pattern recognition and Machine Learning. Springer, 2006. (PR&ML)



