第十二章 特征选择

§12.1 问题引入

§12.2 典型方法

§12.3 应用举例

§12.1 问题引入

一、问题背景

问题背景

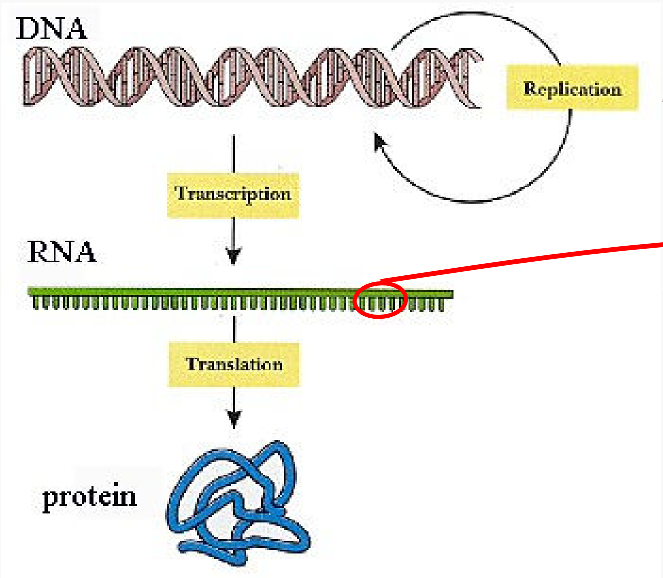
- 维度灾难问题：**文档分类**
  - **任务目标**：将来自不同来源的文档根据内容划分为不同的类别
  - **任务挑战**：单词（term）数量太多，造成文本特征的**维度灾难**

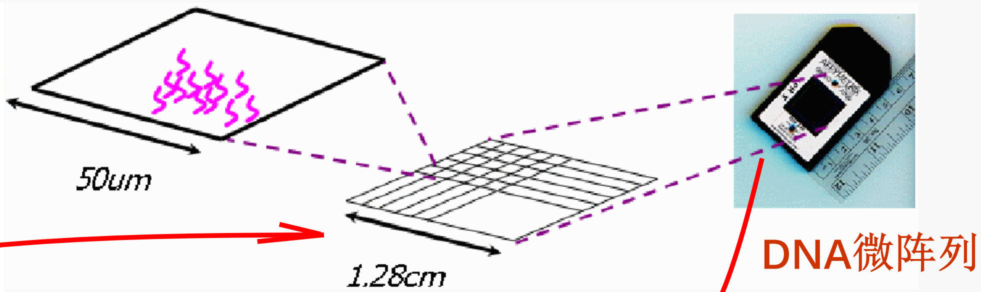
单词

文档	T_1	T_2	………	T_N	类别
D_1	12	0	………	6	体育
D_2	3	10	………	28	旅游
⋮	⋮			⋮	⋮
D_M	0	11	………	16	工作

问题背景

- 维度灾难问题：**基因分类**
  - **任务目标**：根据基因表达微阵列数据将样本进行疾病分类
  - **任务挑战**：基因数量太多，造成基因特征的**维度灾难**





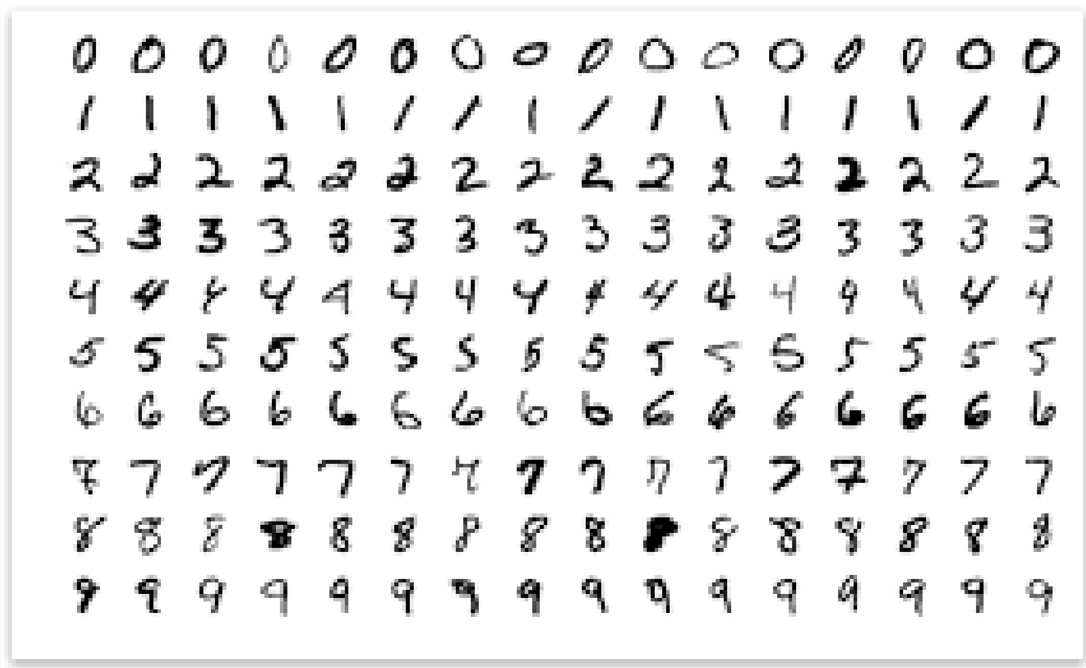
Gene / Sample	M23197_at	U66497_at	M92287_at	...	Class
Sample 1	261	88	4778	...	ALL
Sample 2	101	74	2700	...	ALL
Sample 3	1450	34	498	...	AML
⋮	⋮	⋮	⋮	⋮	⋮

基因表达微阵列数据集

问题背景

- 维度灾难问题：**图像识别**
  - 人脸图像与数字图像：**并非所有维度的特征对分类都有贡献**

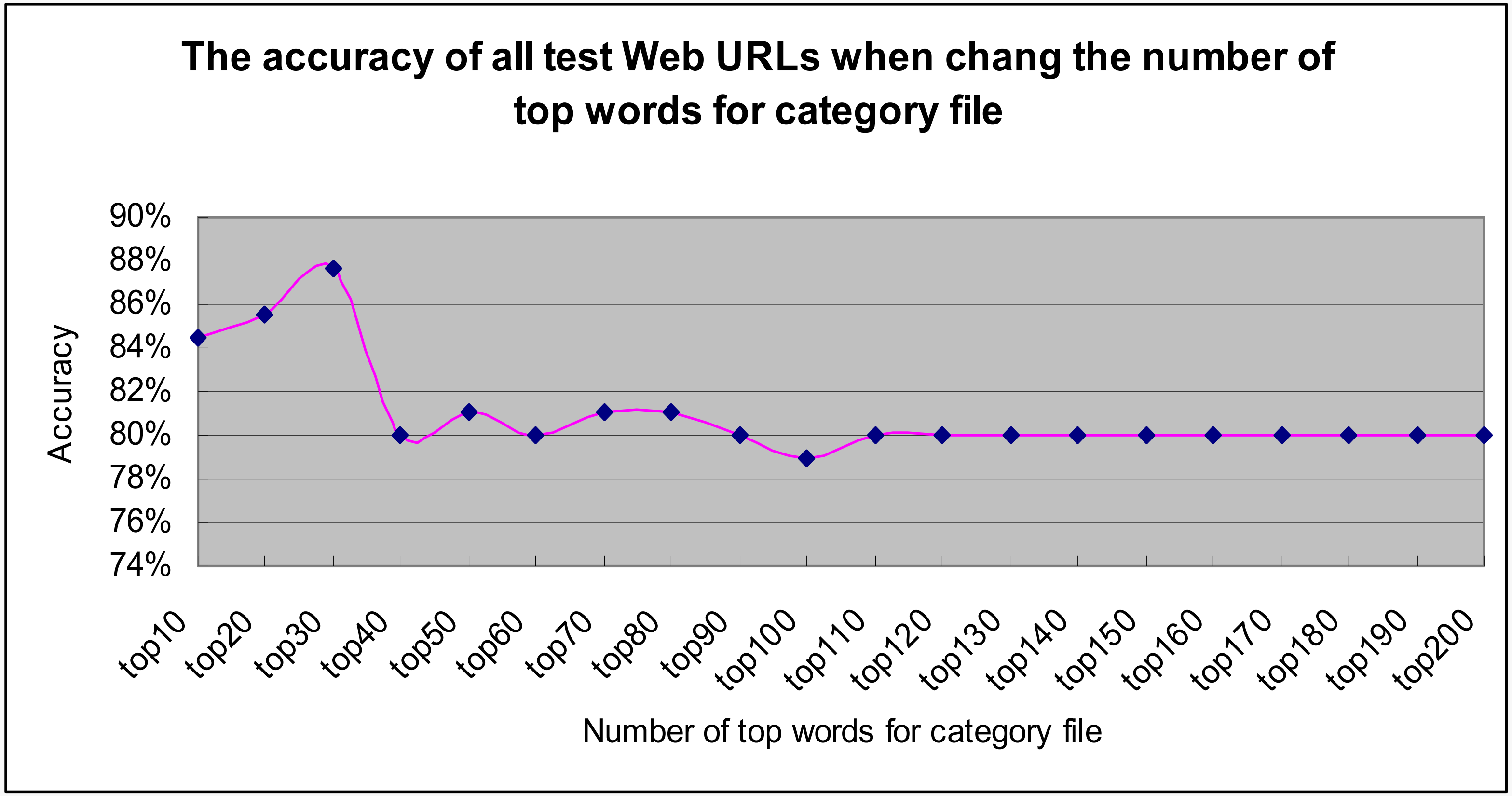
人脸图像



数字图像

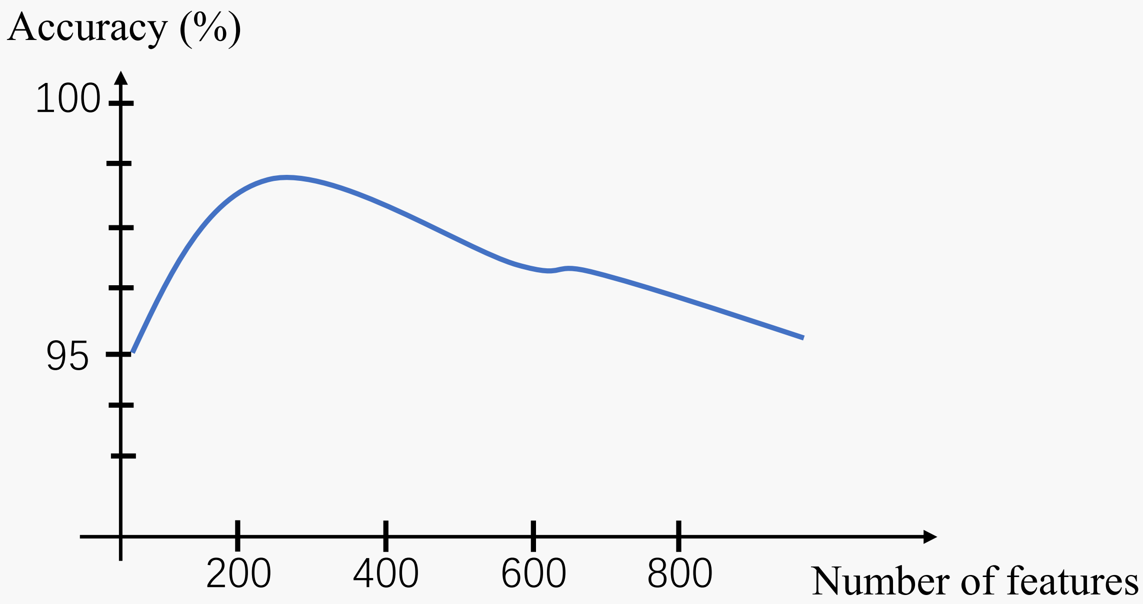
问题背景

- 维数增加导致**任务难度增大**
  - Top words数量**增加**，分类任务准确度反而**下降**



问题背景

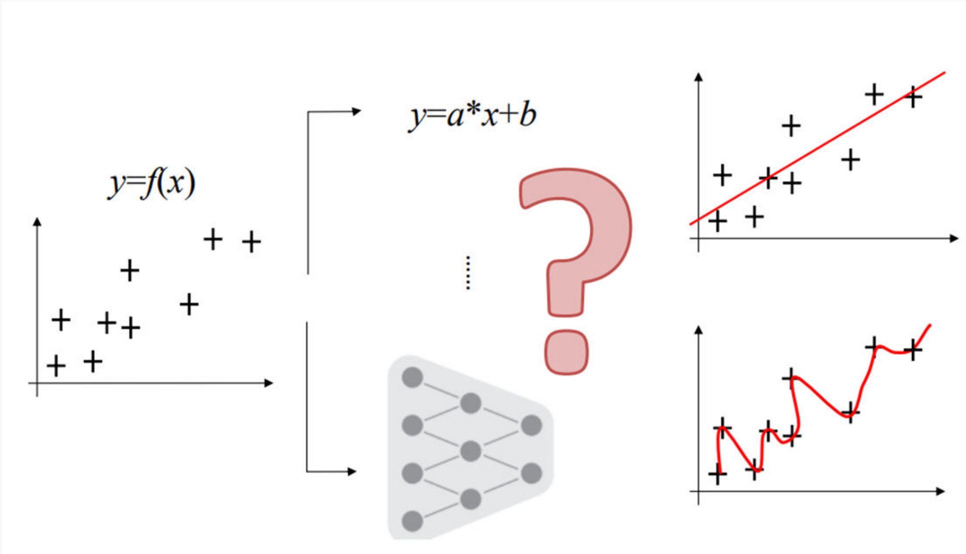
- 维数增加导致**任务难度增大**
  - 特征数量**增加**，垃圾邮件过滤准确度反而**下降**



问题背景

□ 特征选择

- 从数据集每个样本的**所有特征中挑选出与当前学习任务相关的特征子集**，接着再利用数据子集来训练学习器
- 例如：线性回归问题 $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$
  - 当样本数量n远大于样本维度p时，基于最小二乘法的拟合模型具有较小方差
  - 当样本维度p较大而样本数量n较小时，基于最小二乘法的拟合模型具有较大的方差，参数估计不稳定



特征选择

特征选择目标

- 是否输入变量的各个分量都对输出变量的预测有贡献？
  - 有的特征与分类任务之间的**关系不密切**
  - 特征之间存在**冗余性**从而导致高度相关
- 特征太多会影响模型参数估计的**稳定性**

特征选择优点

- 有利于数据的**可视化**与理解
- 降低数据**采集**和**存储**的开销
- 降低模型使用时的**计算**开销
- **避免维数灾难**、提高预测性能

数据降维与特征选择的区别

- 数据降维：把p个特征变为k个新特征
- 特征选择：从p个特征中选出k个，剔除与学习任务无关的属性

§12.2 典型方法

一、子集搜索与评价

二、过滤式特征选择

三、包裹式特征选择

四、嵌入式特征选择

子集搜索与评价

如何获取最优子集

- 直接遍历所有特征子集，当维数过多时**遭遇指数爆炸**

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Gareth James, An Introduction to Statistical Learning with application in R, 2013, Springer

- 从候选特征子集中**不断迭代生成更优候选子集**，则时间复杂度大大减小

子集搜索与评价

- 迭代生成更优候选子集
  - 如何**生成**候选子集
  - 如何**评价**子集好坏
- 搜索候选子集的早期策略
  - 前向搜索
    - 对线性回归模型，即使n<p时也可以用前向算法，在维度为n−1前截止
  - 后向搜索
    - 对线性回归模型，后向算法要求n>p
  - 双向搜索
  - 以贪心算法为例

前向搜索

前向搜索算法思想

- 将每个特征当做一个候选特征子集
- 从当前所有的候选子集中选择出最佳的特征子集
- 在上一轮选出的特征子集中添加一个新的特征，同样地选出最佳特征子集
- 直至选不出比上一轮更好的特征子集

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Gareth James, An Introduction to Statistical Learning with application in R, 2013, Springer

后向搜索

后向搜索算法思想

- 所有特征作为一个候选特征子集
- **尝试去掉上一轮特征子集中的一个特征**并选出当前最优的特征子集
- 直到选不出比上一轮更好的特征子集

Algorithm 6.3 *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Gareth James, An Introduction to Statistical Learning with application in R, 2013, Springer

双向搜索

- 双向搜索算法思想
  - 又称前后向搜索
  - 将前向搜索与后向搜索结合起来，即**在每一轮中既有添加操作也有剔除操作**
  - 在前向算法的基础上允许特征的剔除：
    - 每加入几个特征后，尝试从中去掉部分特征
    - 再继续加入新特征

清華大学

子集搜索与评价

- 给定特征集合$\{a_1, a_2, \cdots, a_d\}$，每个特征都可看作一个候选子集
  - 前向搜索：$\{a_2\} \to \{a_2, a_4\} \to \{a_2, a_4, a_5\} \to \cdots$
  - 后向搜索：$\{a_1, a_2, \cdots, a_d\} \to \{a_1, a_2, \cdots, a_{d-1}\} \to \cdots$
  - 双向搜索：$\{a_1, a_2, \cdots, a_d\} \to \{a_1, a_2, \cdots, a_{d-1}\} \to \cdots$

    （蓝色表示待定，黑色表示确定不会被删除）
- 贪心策略的问题
  - 以前向搜索为例，假如第三轮选择a_5优于a_6，选定集为$\{a_2, a_4, a_5\}$
  - 但在第四轮可能是$\{a_2, a_4, a_6, a_8\}$优于所有的$\{a_2, a_4, a_5, a_i\}$
  - 此时第三轮选择无效
  - 如果不进行穷举搜索，可能陷入局部最优

子集搜索与评价

特征子集的评价

- 基于**信息熵**的判据
  - 熵是不确定性的度量：$I = -\sum_{i=1}^{k} P_i log_2 P_i$
  - 条件熵：特征取值对分类的影响

$$H(x) = -\sum_{i=1}^{k} p(w_i|X = x) log_2 p(w_i|X = x)$$

  - 一个极端的例子：

特征$x=x_1$对分类没有效果

$$p(\omega_i \mid x_1) = \frac{1}{c}$$

熵大（H=lnc）

特征$x=x_2$能够很好地进行区分出第i类

$$p(\omega_i \mid x_2) = 1;\ \ p(\omega_j \mid x_2) = 0\ \ ;i \neq j$$

熵小（H=0）

子集搜索与评价

特征子集的评价

- 基于**信息熵**的判据
  - 给定数据集D，其样本属性均为离散型，假定D类中第i类样本所占比例为$p_i(i=1,2,\cdots,|\mathcal{Y}|)$
  - 对属性子集A, 假定根据其取值将D分成了V个子集$\{D^1,D^2,\cdots,D^V\}$，每个子集样本在A上的取值相同
  - 计算属性子集A的信息增益

$$Gain(A)=Ent(D)-\sum_{v=1}^{V}\frac{|D^v|}{|D|}Ent(D^v)$$

  - 其中信息熵定义为

$$Ent(D)=-\sum_{i=1}^{|\mathcal{Y}|}p_k\log_2 p_k$$

子集搜索与评价

- 特征子集的评价
  - 基于**信息熵**的判据
    - 属性子集A的信息增益

$$Gain(A) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

    - 信息增益$Gain(A)$越大，表明特征子集A包含的有助于分类的信息越多
    - 对于每个候选特征子集，可基于训练数据集D计算信息增益，以此作为评价准则

子集搜索与评价

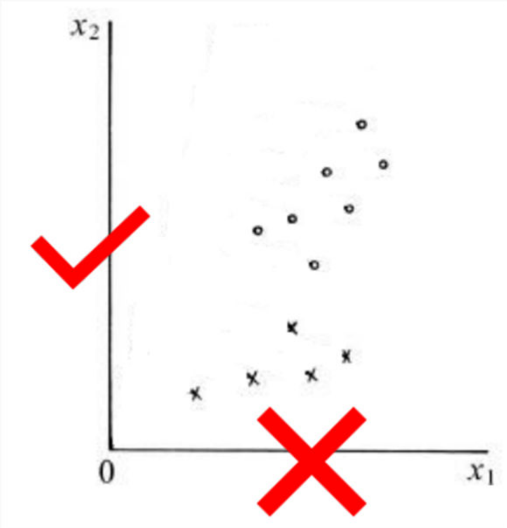
特征子集的评价

- 假设：数据集的属性皆为离散属性，给定一个特征子集，便可以通过这个特征子集的取值将数据集合划分为V个子集
- 例如：A1={男,女}，A2={本科,硕士}，可以将原数据集划分为2*2=4个子集
- 基于类内类间距离的判据：类内散度小，类间散度大

Fisher准则 $J_F(w)=\frac{(\widetilde{\mu_1}-\widetilde{\mu_2})^2}{\widetilde{S_1}^2+\widetilde{S_2}^2}$

类均值 $\widetilde{\mu_i}=\frac{1}{m}\sum_{x_{j\in\gamma}}x_j,i=1$

类内散度 $\widetilde{S_i}^2=\sum_{x_{j\in\gamma}}(x_j-\widetilde{\mu_i})^2,i=1,2$



子集搜索与评价

□ 特征子集的评价

- 基于**随机变量间关联性**的判据：把特征取值X和响应值Y（类别标签）都视为随机变量，衡量二者之间的关联程度
  - 相关系数(线性相关)

$$corr(X,Y)=\frac{\sum_{i=1}^{n}\left(x_i^{(k)}-\bar{x}_i\right)\left(y_i^{(k)}-\bar{y}_i\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i^{(k)}-\bar{x}_i\right)^2\left(y_i^{(k)}-\bar{y}_i\right)^2}}$$

  - 互信息(线性、非线性)

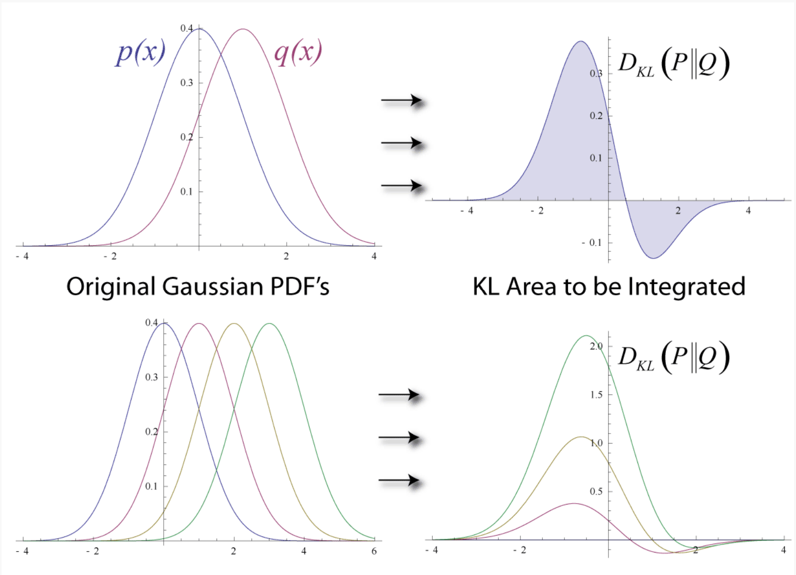
$$I(X,Y)=\sum_{X}\sum_{Y}p(x,y)log\frac{p(x,y)}{p(x)p(y)}$$

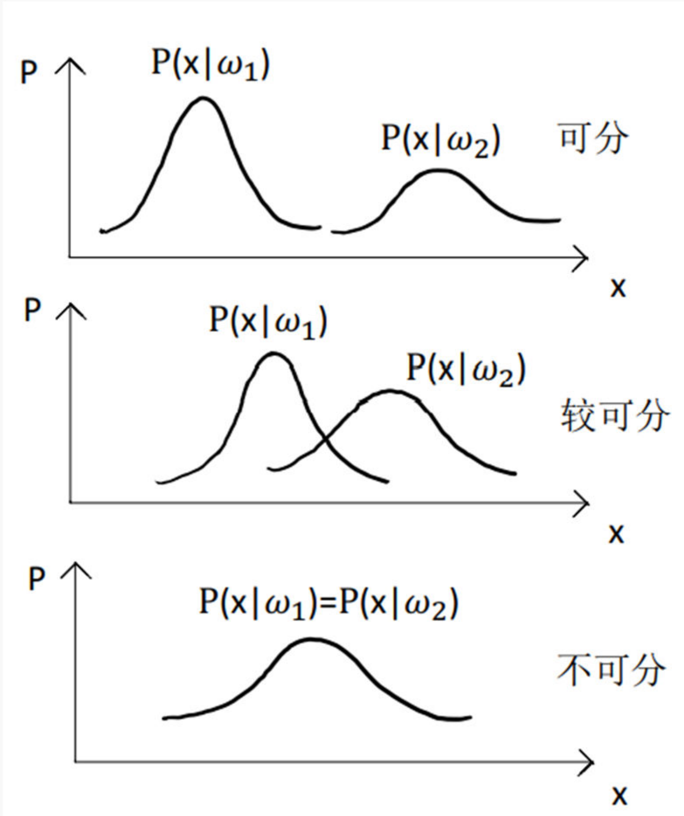
子集搜索与评价

特征子集的评价

- 基于**概率分布可分性**的判据：如何度量两类密度函数之间的相似性？
  - KL散度 (Kullback-Leibler divergence)

$$D_{KL}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$





子集搜索与评价

□ 特征子集的评价

- 基于**统计检验**的判据：选择在两类间有显著统计差异的特征
- 例如：t-test(正态分布假设)

A	B
79.98	80.02
80.04	79.94
80.02	79.98
80.04	79.97
80.03	79.97
80.03	80.03
80.04	79.95
79.97	79.97
80.05	
80.03	
80.02	
80.00	
80.02	

sample 1: $X_1,\cdots,X_n \quad X \sim N(\mu_X,\sigma^2)$

sample 2: $Y_1,\cdots,Y_m \quad Y \sim N(\mu_Y,\sigma^2)$

pooled sample variance $\quad s_P^2 = \dfrac{(n-1)S_X^2+(m-1)S_Y^2}{m+n-2}$

the null hypothesis

$$H_0: \quad \mu_X = \mu_Y$$

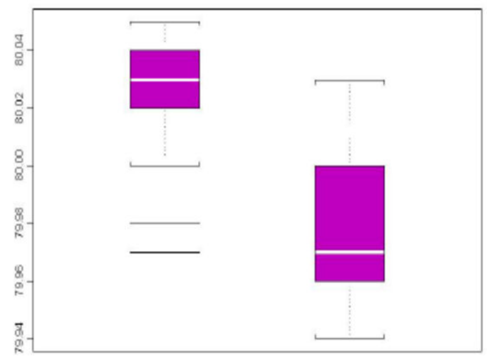
alternative hypotheses

two-sided $\quad H_1: \quad \mu_X \neq \mu_Y$

one-sided $\quad H_2: \quad \mu_X > \mu_Y$

$\quad H_3: \quad \mu_X < \mu_Y$

t-statistic $\quad t = \dfrac{\overline{X}-\overline{Y}}{s_{\overline{X}-\overline{Y}}} = \dfrac{\overline{X}-\overline{Y}}{s_P\sqrt{\dfrac{1}{n}+\dfrac{1}{m}}}$



特征选择的策略

- 过滤特征选择法(filtering)：先选再学
  - 先选特征，再做学习

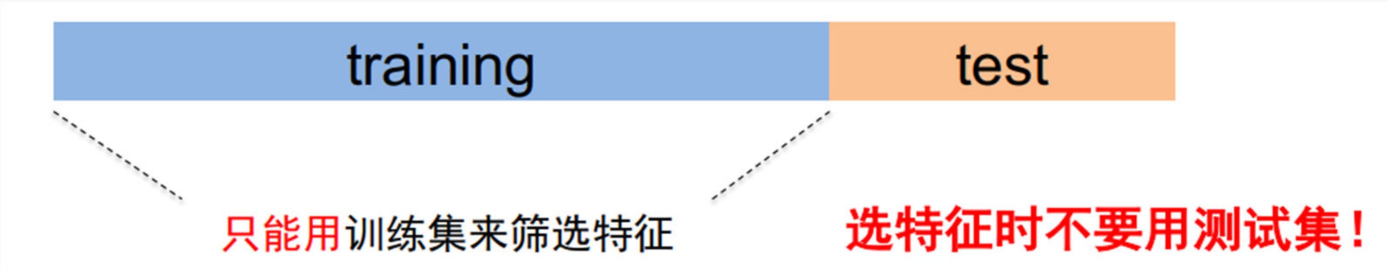
- 包裹特征选择法(wrapper)：边学边选
  - 以最终**分类器的性能作为指标**来选特征
  - 一般要求分类器**能够处理高维特征**
  - 分类器能够在特征维度高但样本数有限时仍能

- 嵌入特征选择法(embedding)：目标函数引入正则项

过滤式特征选择

过滤式特征选择

- 将特征选择与学习器训练相分离的特征选择技术：**首先将相关特征挑选出来，再使用选择出的数据子集来训练学习器**
- 数据划分



经典方法：Relief

- 度量：“相关统计量”，某种统计量来**度量特征X与和响应值Y之间的**“关联程度”，每个分量代表着相应特征的重要性 → **选出其中“关联程度”最高的k个特征，或者高于一定阈值t的所有的特征**

过滤式特征选择

- 过滤式特征选择：Relief
  - 给定训练集$\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$
  - 对于每个样例x_i，Relief先在x_i的**同类样本中寻找其最近邻$x_{i,nh}$**，称为"猜中近邻"(near-hit)
  - 再从x_i的**异类样本中寻找其最近邻$x_{i,nm}$**，称为"猜错近邻"(near-miss)
  - 相关统计量对于属性j的分量为：

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

过滤式特征选择

过滤式特征选择：Relief

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

- 其中x_a^j表示样本x_a在属性j上的取值
- $diff(x_a^j, x_b^j)$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 $diff\left(x_a^j, x_b^j\right) = 0$，否则为1
- 若属性j为连续型，则$diff\left(x_a^j, x_b^j\right) = |x_a^j - x_b^j|$，此处$x_a^j, x_b^j$已规范化到$[0,1]$

过滤式特征选择

- 过滤式特征选择：Relief
  - 从上式可以看出，若x_i与其猜中近邻$x_{i,nh}$在属性j上的距离小于x_i与其猜错近邻$x_{i,nm}$的距离，则说明**属性j对区分同类与异类样本是有益的**，此时增大属性j所对应的统计量分量
  - 若x_i与其猜中近邻$x_{i,nh}$在属性j上的距离大于x_i与其猜错近邻$x_{i,nm}$的距离，则说明**属性j对区分同类与异类样本起负面作用**，此时减小属性j所对应的统计量分量
  - 最后对基于不同样本得到的估计结果进行平均，就得到各属性的相关统计量分量，分量值越大，对应属性的分类能力就越强

过滤式特征选择

过滤式特征选择：Relief

- 实际上，Relief只需要**在数据集的采样上**而不必在整个数据集上估计相关统计量
- 因此Relief的时间开销随采样次数以及原始特征数线性增长，是效率很高的过滤式特征选择算法
- 由于公式中只定义了同类和异类，Relief只能用于二分类问题，其变体Relief-F可以处理多分类问题

	标准Relief算法	拓展变体Relief-F算法
应用	二分类问题	二分类问题
核心	只有一个猜错近邻	有多个猜错近邻

过滤式特征选择

过滤式特征选择：Relief-F

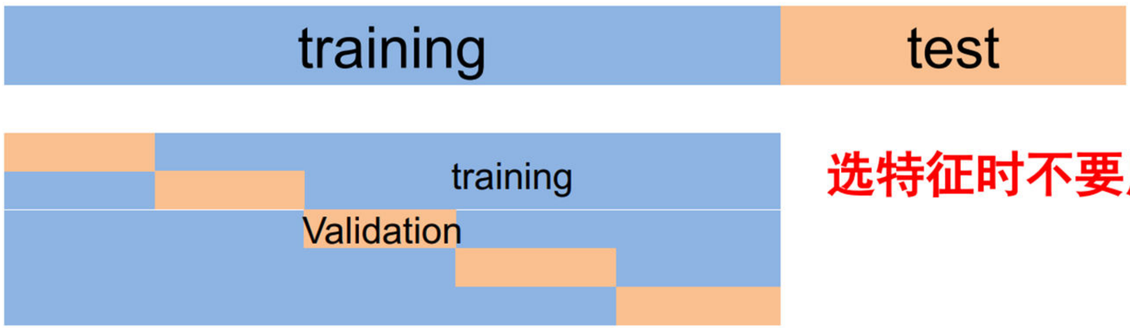
- 假定数据集D中的样本来自$|Y|$个类别
- 对于示例x_i，若它属于第k类，则Relief-F先在第k类的样本中寻找x_i的最近邻示例$x_{i,nh}$并将其作为猜中近邻
- 然后**在第k类之外的每个类中找到一个x_i的最近邻示例作为猜错近邻**，记为$x_{i,l,nm}\ (l=1,2,\ldots,|Y|;\ l\neq k)$
- 则相关统计量对应于属性j的分量为：

$$\delta^j=\sum_i -diff\left(x_i^j,x_{i,nh}^j\right)^2+\sum_{l\neq k}\left(p_l\times diff\left(x_i^j,x_{i,l,nm}^j\right)^2\right)$$

- 其中p_l为第l类样本在数据集D中所占的比例

包裹式特征选择

- 包裹式特征选择
  - 把特征选择与分类器设计集成在一起，利用分类器进行特征选择
  - 数据划分



选特征时不要用测试集!

在训练集上利用交叉验证选择特征

- 经典方法：LVW（Las Vegas Wrapper）

包裹式特征选择

- 经典方法：LVW（Las Vegas Wrapper）
  - 在拉斯维加斯框架下使用随机策略来进行特征子集的搜索
  - 算法流程

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';
  $d' = |A'|$;
  $E' = \mathrm{CrossValidation}(\mathfrak{L}(D^{A'}))$;
  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^*

拉斯维加斯算法：采样越多，越有机会找到最优解，不一定会给出解，且给出的解一定是正确解

包裹式特征选择

- 经典方法：LVW（Las Vegas Wrapper）
  - 算法第8行是**通过在数据集上使用交叉验证法估计学习器的误差**
  - 该误差是在仅考虑特征子集A'时得到的，**若其比在当前特征子集A上的误差更小，或误差相当但A'中包含的特征数更少，则将A'保留下来**
  - 由于LVW算法中特征子集的随机搜索策略需要在每次评价时都训练学习器，计算开销很大，因此**算法设置了停止条件控制参数T**
  - 根据拉斯维加斯方法框架，若初始特征数$|A|$很大、T设置较大，则算法运行时间可能非常长

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：

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: $E'=\mathrm{CrossValidation}(\mathfrak{L}(D^{A'}))$;
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^*

过滤式与包裹式对比

- 两步流程
  - 特征选择（训练集）

    利用某种计量指标筛选，如相关系数、统计检验等
  - 分类（训练集-测试集）

    利用筛选出来的特征构建分类器

- 迭代流程
  - 分类（训练集-验证集）
  - 特征选择

    根据分类效果选择特征
  - 分类（训练集-测试集）

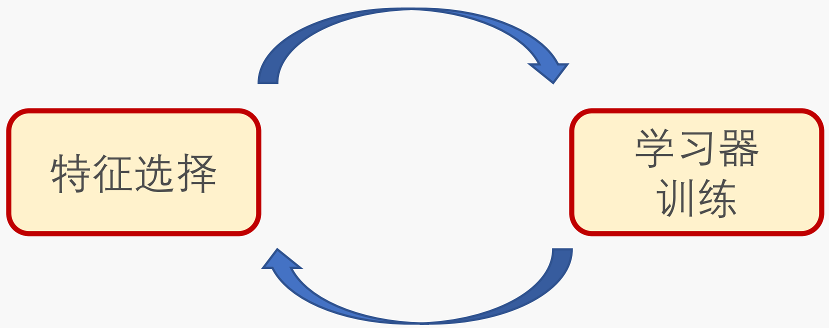
    利用筛选出来的特征构建分类器

嵌入式特征选择

嵌入式特征选择

- 在过滤式和包裹式特征选择方法中，特征选择过程与学习器训练过程有明显的分别
- 嵌入式特征选择是**将特征选择过程与学习器训练过程融为一体**，两者在同一个优化过程中完成，在学习器的训练过程中自动进行了特征选择

过滤式/包裹式
特征选择



嵌入式特征选择

嵌入式特征选择

嵌入式特征选择

- 给定数据集$D=\{(x_1,y_1),(x_2,y_2),\dots,(x_m,y_m)\}$，其中$x\in\mathbb{R}^d, y\in\mathbb{R}$
- 考虑**最简单的线性回归模型**，以平方误差为损失函数，则优化目标为

$$\min_{w}\sum_{i=1}^{m}\left(y_i-w^Tx_i\right)^2$$

- 当样本特征很多，样本数相对较少时，上式容易陷入过拟合
- 为了缓解过拟合问题，可以引入正则化项

嵌入式特征选择

嵌入式特征选择

- 若使用L2范数正则化，则有

$$\min_{w}\sum_{i=1}^{m}\left(y_i-w^Tx_i\right)^2+\lambda||w||_2^2$$

- 其中正则化参数$\lambda>0$，上式称为“岭回归”（ridge regression），**通过引入L2范数正则化，可以显著降低过拟合的风险**
- 也可以将L2范数替换为Lp范数，当采用L1范数时，则有

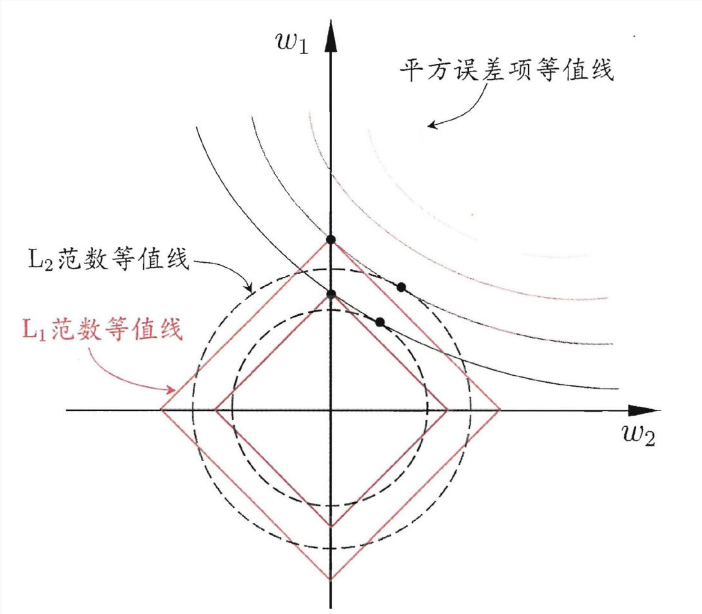
$$\min_{w}\sum_{i=1}^{m}\left(y_i-w^Tx_i\right)^2+\lambda||w||_1$$

- 其中正则化参数$\lambda>0$，上式称为LASSO(Least Absolute Shrinkage and Selection Operator)

嵌入式特征选择

嵌入式特征选择

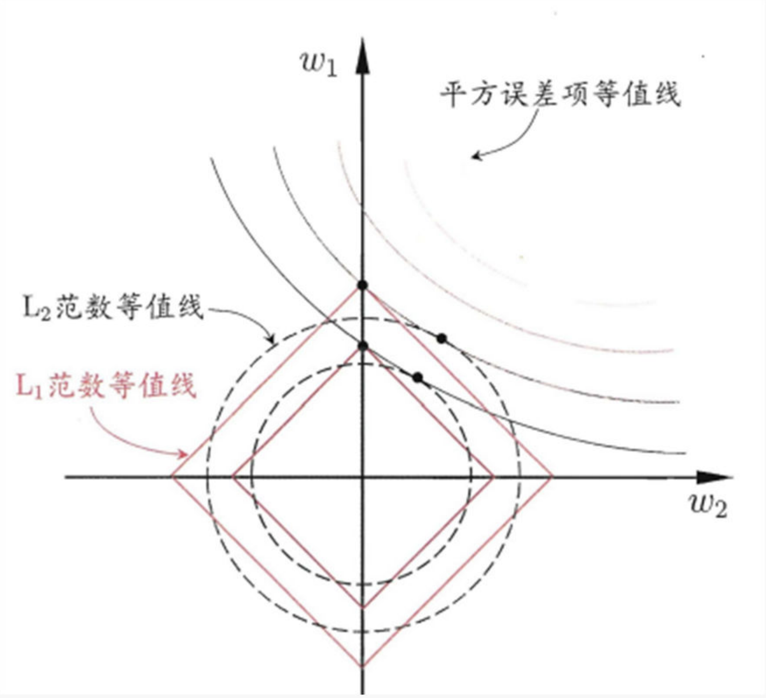
- L1范数和L2范数正则化有助于降低过拟合风险，但前者还会带来一个额外的好处：**它比后者更易于获得“稀疏”解，即解得的w会有更少的非零分量**
- 假定x仅有两个属性，于是解出的w都只有两个分量（w1, w2）
- 将其作为两个坐标轴，并在其中绘制平方误差损失项的“等值线”，即平方误差项取值相同的点的连线
- 绘制出L1和L2范数的等值线



嵌入式特征选择

嵌入式特征选择

- 可以看出**采用L1范数时，等值线的交点更易于落在坐标轴上**，此时可以得到稀疏解
- 注意到w取得稀疏解意味着初始的d个特征中仅有对应着w的非零分量的特征才会出现在最终模型中
- 于是求解L1范数正则化的结果是得到了仅采用一部分初始特征的模型
- 换言之，**基于L1正则化的学习方法就是一种嵌入式特征选择方法**，其特征选择过程与学习器过程融为一体



正则化

正则化

- 与特征选择相关的性质

	L1范数	L2范数
下降速度	按照绝对值函数来下降 能很快地下降到0	按照二次函数来下降 较大可能收敛在0的附近
空间限制	L1范数与L2范数都试图在最小化损失函数的同时，让权值W也尽可能地小	

§12.3 应用举例

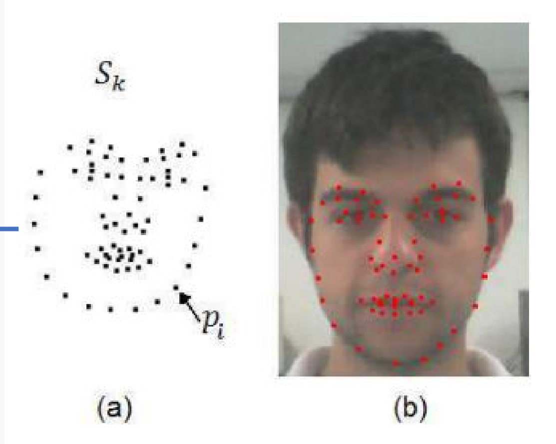
一、人脸识别

二、思考题

应用举例：人脸识别

- 人脸识别
  - **挑战：**不同光照、姿态、表情、年龄、背景等因素的影响
- 基于特征选择的人脸识别
  - 对人脸像素进行特征选择的动机
    - 高清人脸图片带来大量的计算量和存储需求，时延增大
    - 人脸图像像素包含的信息量存在差异

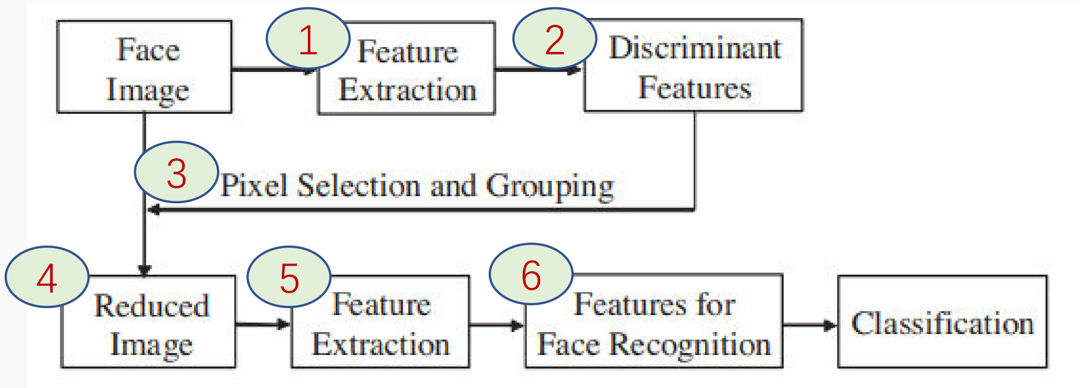
由ASM方法得到的人脸关键点位置



眉眼、鼻子、嘴巴和脸型往往包含更多的身份信息

应用举例：人脸识别

- 特征选择在人脸识别中的应用举例
  - 整体架构



  - 第一步，先对**整个图片提取特征**
    - 使用轻量级的特征提取方法，引入较小计算量
    - 甚至可以仅仅是**灰度化**
  - 第二步，从所有像素的特征中选择**判别能力更强的特征**
    - 特征集合：$\{x_1, x_2, \ldots, x_N\}$，$x_i \in \mathbb{R}^n$表示第 i 个人脸的所有 n 个像素的特征拼接在一起得到的图片级特征

应用举例：人脸识别

- 特征选择在人脸识别中的应用举例
  - 第二步，从所有像素的特征中选择**判别能力更强的特征**
    - 特征集合：$\{x_1, x_2, \ldots, x_N\}$，$x_i \in \mathbb{R}^n$表示第 i 个人脸的所有 n 个像素特征拼接在一起得到的图片级特征
    - 投影矩阵：使用LDA等方法求解上述特征集合的最佳投影矩阵W，其中W的每一列 w_l 表示一个投影方向
    - **像素重要性**：由投影公式

$$\hat{x}_i = W^T x_i$$

w_l 的第 k 个分量的绝对值 $|w_{lk}|$ 反映了 x_i 的第 k 个像素 x_{ik} 的重要性，即

$$Score(\text{第}l\text{个投影方向}, \text{第}k\text{个像素}) = |w_{lk}|$$

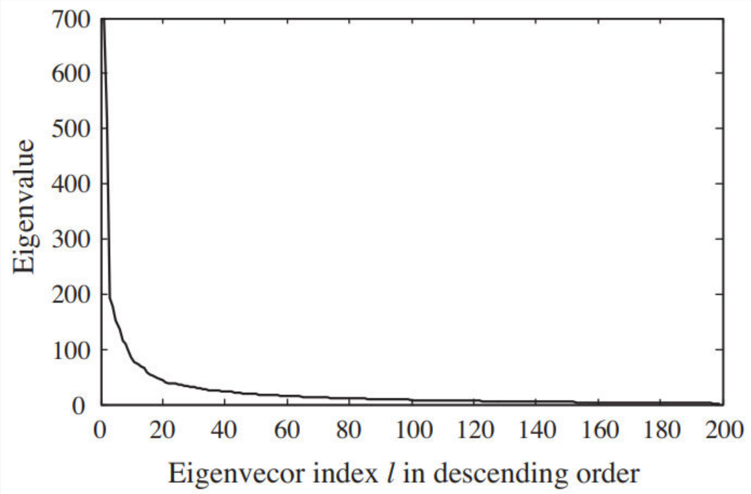
应用举例：人脸识别

- 特征选择在人脸识别中的应用举例
  - 第三步，得到像素的**Mask**，并**滤波**
    - 像素重要性：$Score(\text{第}l\text{个投影方向},\text{第}k\text{个像素}) = |w_{lk}|$
    - **像素重要性排序**：每个投影方向都会得到一个所有像素的重要性排序

$$order(l) = sort(|w_{lk}|)$$

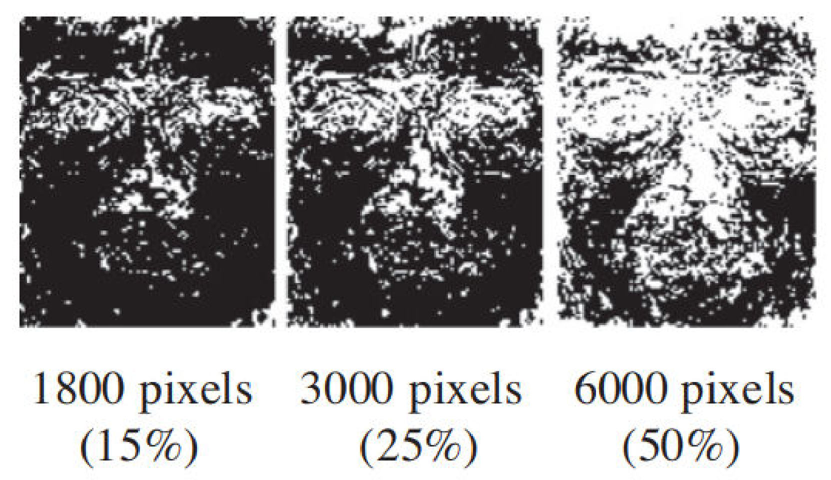
    - **投影方向重要性**：各投影方向的重要性不同
      - 对应的特征值越大，该投影方向越重要
      - 投影方向的加权系数：$\alpha_l = \frac{\lambda_l}{\sum_j \lambda_j}$
    - 最终像素重要性排序：

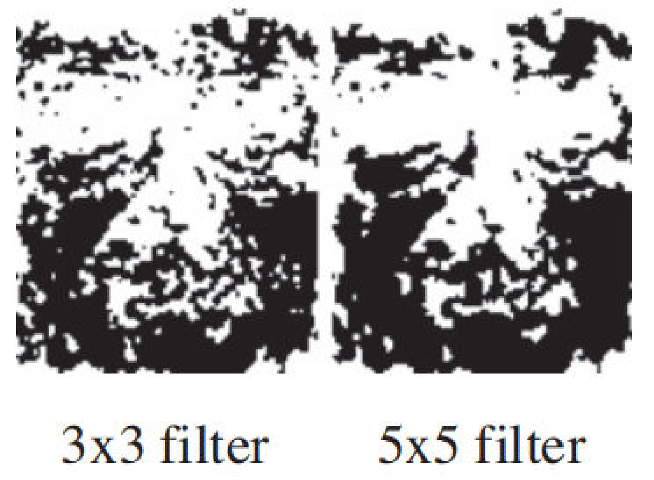
$$order = \sum_l \alpha_l \times order(l)$$



应用举例：人脸识别

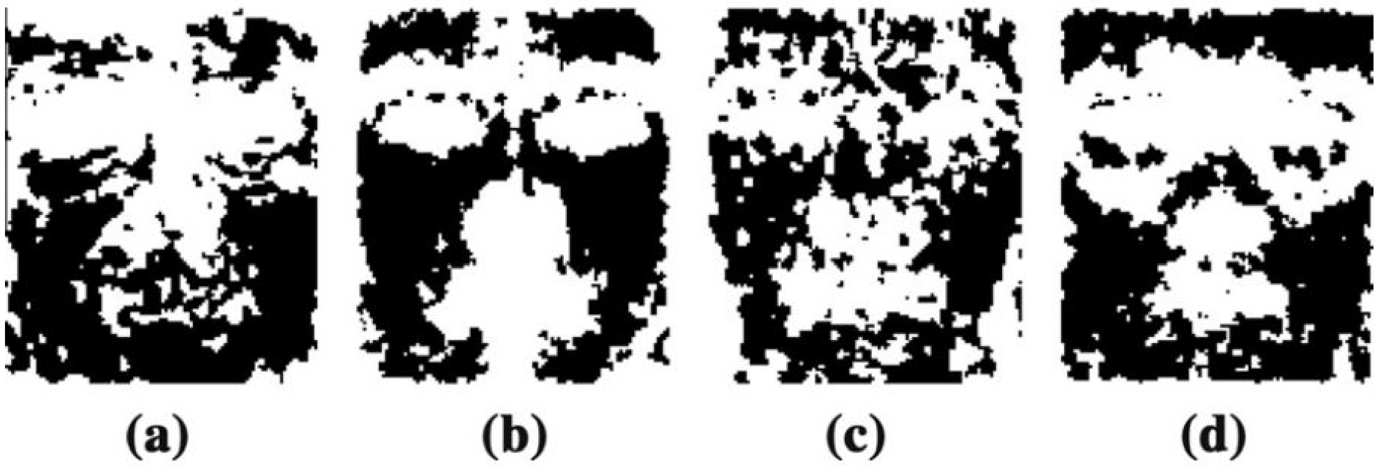
- 特征选择在人脸识别中的应用举例
  - 第三步，得到像素的**Mask**，并**滤波**
    - 最终像素重要性排序：$order = \sum_l \alpha_l \times order(l)$
    - **生成Mask**：根据人为设定的阈值，截取排名前几的像素
    - **滤波**：由于直接mask会得到很多噪点，故使用低通滤波器对mask进行滤波处理





应用举例：人脸识别

- 特征选择在人脸识别中的应用举例
  - 第四步，得到mask后的人脸
  - 第五步，对mask后的人脸提取特征，可只考虑选出的像素
  - 第六步，使用第五步的特征进行后续的人脸识别任务



ig. 11. Masks obtained from (a) FERET database; (b) CMU-PIE database; (c) Yale B database; (d) AR database.

更多滤波后的人脸mask

思考题

- 试从函数拟合的角度分析特征选择的必要性
  - 如果特征组合中包含很多对分类没有贡献的特征，会加剧分类器的过学习，导致分类器测试错误率或者交叉验证错误率的增加
- 试分析分类问题中，特征选择的目标维数对分类器性能的影响
  - 在一个原始特征维数较高的问题中，当逐步选择特征时往往可以看到，随着特征数目的减少，分类器性能逐步提高，但如果特征数目减少到一定程度后继续减少，则分类器性能又会逐渐恶化