第17章：特征提取与选择
Feature Extraction and Feature Selection

刘智勇（zhiyong.liu@ia.ac.cn）

助教：杨学行(xhyang@nlpr.ia.ac.cn)
吴一超(yichao.wu@nlpr.ia.ac.cn)

提纲

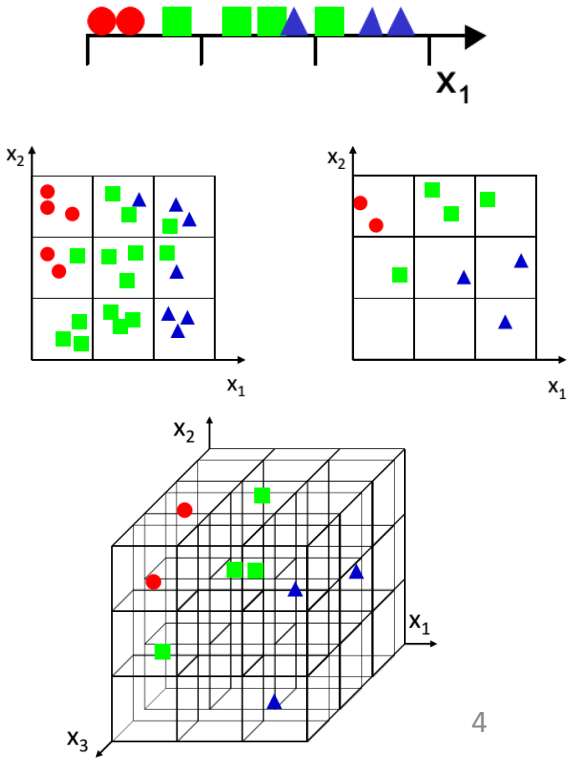
- 维数灾难
- 特征提取
  - 两种指标
  - 线性方法
  - 非线性方法
- 特征选择
  - 目标函数
  - 搜索策略

特征的含义

- A feature of something is an interesting or important part or characteristic of it； a prominent attribute or aspect of something
- In machine learning and pattern recognition, a feature is an individual measurable property of a phenomenon being observed
- 一个不是很恰当的例子：大人，小孩，狗
- 显然，两者有着明显的鸿沟，如何弥补，去伪存真？本节课的主要内容

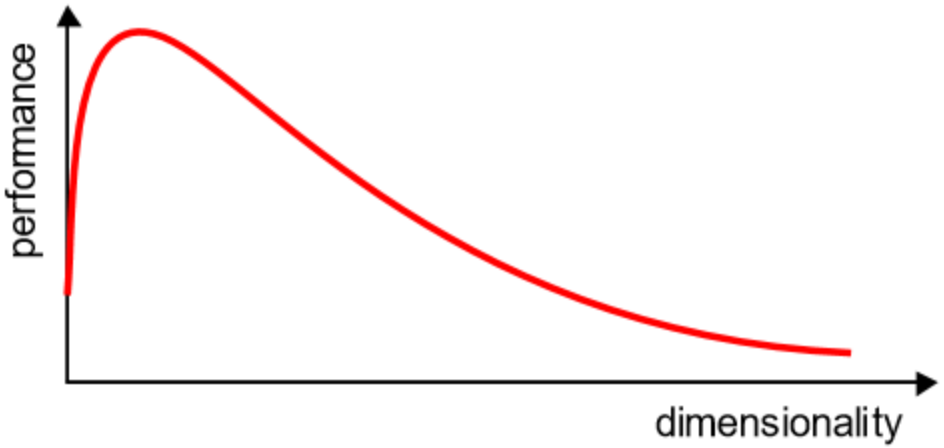
维数灾难（curse of dimensionality）

- 由理查德-贝尔曼在1961年提出，指的是多变量分析中随着维数增加带来的系列问题
- 一个简单的例子：一个分类（3类）问题，随着维数增加
  - 样本的区分度增加，利于分类
  - 保持同样的密度，需要的样本数目指数增加
  - 将特征空间平均划分是不合理的



维数灾难（curse of dimensionality）

- 降维的必要性
  - 估计一个密度分布函数或分类函数所需要的样本数量相对于维数是指数增长的
  - 当样本数目一定的时候，如果样本的维数过大会造成函数缺乏有效估计，导致分布分类的效果其实是下降的
  - 在小样本问题上，通过降低特征维数（牺牲一些特征）其实反而可以提高性能
  - 另一角度，人们只能直观感受3维或以下的模式，对于数据可视化来说也需要降低维度



降维

- 主要有两种方法
  - 特征提取（feature extraction）：通过（线性或非线性变换）组合现有的特征
  - 特征选择（feature selection）：选择较少现有的特征
- 特征提取可以表示为
  - Given a feature space $x\_i \in \Re\text{^}N$ find a mapping $y = f\ (x) : R\text{^}N \rightarrow R\ \text{^}M$ with $M < N$ such that the transformed feature vector $y \in R\text{^}M$ preserves (most of) the information or structure in $R\text{^}N$
- 特征选择可以表示为
  - 给定一组特征$X_N = \{x_i | i = 1 \ldots N\}$，找到它的一个子集$Y_M, M < N$最大化某目标函数

$$Y_M = \{x_{i_1}, x_{i_2}, \ldots x_{i_M}\} = \text{argmax}_{M,i_M} J(Y_M)$$

特征提取（feature extraction）

- 特征提取的两类指标
  - 依据数据本身指标，例如最佳重建（best reconstruction）：PCA，FA，ICA，Local PCA，Nonlinear PCA etc.
  - 依据分类指标：LDA
- 两大类方法
  - 线性变换（子空间subspace）：PCA，FA，ICA，LDA等，输出是输入的线性组合

    $y_i = w_{i,1}x_1 + \ldots w_{i,N}x_N, for\ i = 1,\ldots,M, or$

    $y = Wx, x \in R^N, y \in R^M, W \in R^{M \times N}$
  - 非线性变换（流形manifold）：Local PCA，LLE，各类kernel方法

    $y = F(x)$

特征提取（feature extraction）：PCA

- 主元分析PCA（Principal Component Analysis）- the most principal component
  - 求解线性变换$Y = WX$，使得
    - Y相互是正交的（orthogonal）
    - Y的方差最大

    $w_1 = argmax_{\|w\|=1} Var\{wX\} =$

    $argmax_{\|w\|=1} \|wX\|^2 = argmax_{\|w\|=1} \{wXX^T w^T\} = argmax_{\|w\|=1} \left\{\frac{wXX^T w^T}{ww^T}\right\}$
  - 根据瑞利商Rayleigh quotient，w_1是XX^T（也即X的协方差Σ）的最大特征向量，而$Var\{wX\}$等于Σ的最大特征值
  - 另一种求解，引入拉格朗日算子

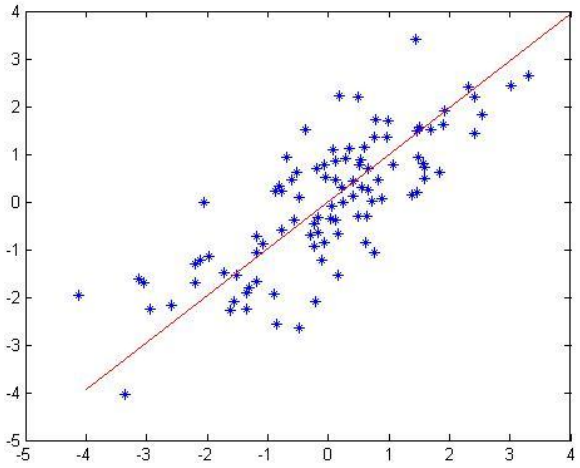
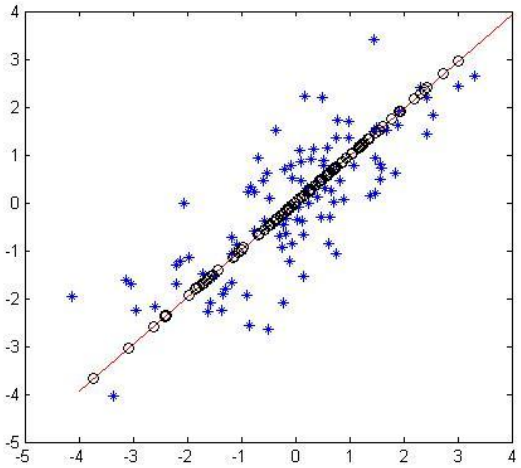
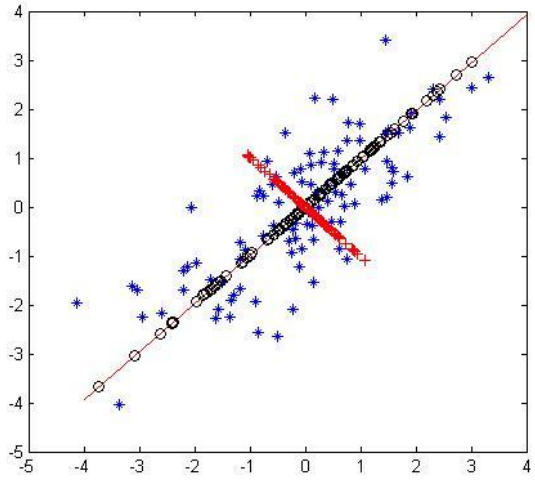
  $L(w) = w\Sigma w^T - \lambda(ww^T - 1)$，对其求导得到$\frac{dL(w)}{dw} = 2w\Sigma\text{-}2\lambda w = 0 \Longrightarrow \Sigma w^T = \lambda w^T$，

  有$Var\{X^T w\} = \frac{wXX^T w^T}{ww^T} = \lambda \frac{ww^T}{ww^T} = \lambda$

特征提取（feature extraction）：PCA

- 主元分析PCA（Principal Component Analysis）- other principal components: 理解一
  - 求第k个PC，那么将原始数据减去已求到的$k-1$个PCs的重建数据（reconstruction）,然后继续刚才的过程

$$X_k = X - \sum_{i=1}^{k-1} w_i^T w_i X = X - W_{k-1}^T W_{k-1} X$$

特征提取（feature extraction）：PCA

- 均方重建误差最小（mean square reconstruction error minimization :MSRE minimization）角度

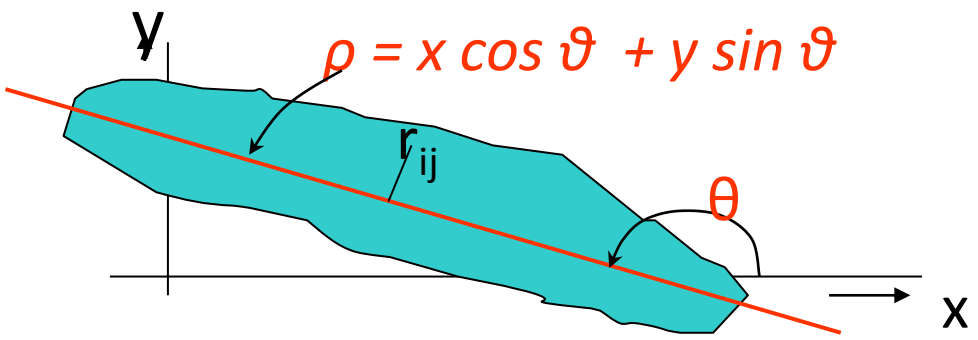
$$W = arg \min_{WW^T=I} E(W,X) =$$
$$arg \min_{WW^T=I} \|X - W^T WX\|^2 =$$
$$arg \min_{WW^T=I} tr\{(X - W^T WX)^T (X - W^T WX)\} =$$
$$arg \min_{WW^T=I} tr\{-X^T W^T WX\} =$$
$$arg \max_{WW^T=I} tr\{X^T W^T WX\} \Longrightarrow W$$包含对应最大的M个特征值的特征向量

特征提取（feature extraction）：PCA

- 再一种解读：数据分布的朝向（主轴），以二维物体为例
  - 可以通过最小化以下能量函数（二阶矩）得到

$$E = \sum_{i=1}^{N}\sum_{j=1}^{M} r_{ij}^2 B[i,j]$$，其中

$B[i,j]$表示坐标（i, j）处的值，r表示坐标（i, j）到线ρ的距离



特征提取（feature extraction）：PCA

PCA求解过程

- batch方法
  - 计算数据的协方差矩阵Σ
  - 对Σ特征值分解，取对应的前M个特征值的特征向量组成变换矩阵W
- Online方法
  - 观察数据是随着时间不断地获取假设数据集，新数据到来时，实时更新主成分
  - 数据量太大，维数太高时，离线PCA失效
  - Online PCA （Oja’s rules）

特征提取（feature extraction）：PCA

Oja's Rule

- Given the sample x_t, the first PC is learned by

$$w(t+1) = w(t) + \eta_t(y_t x_t^T - y_t^2 w(t))$$

$$y_t = w(t)x_t$$

- 当算法收敛时候

$$\Delta w = yx^T - yy^T w = wxx^T - wxx^T w^T w$$

$$= wC - wCw^T w = 0$$

  - wCw^T是一个标量，记为λ，因此得到

$$wC - \lambda w = 0$$

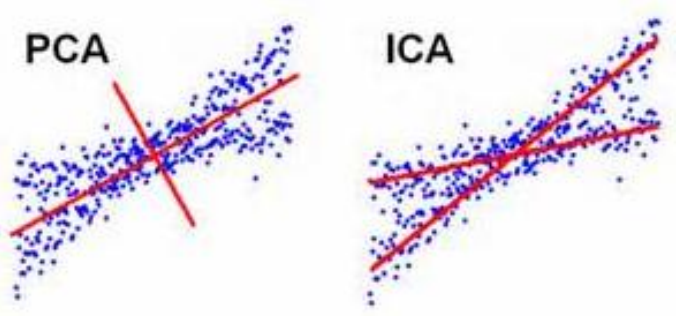
- 对应于最大的特征值和特征向量
- 公式可以直接引申到m个PC's

特征提取（feature extraction）：PCA

- NOTE
  - 利用PCA做图像中的直线检测（线段细化）[Liu et al. EMMCVPR 2000]
  - PCA只是且只能利用至多二阶的统计信息（方差），意味着PCA假设数据为高斯分布，或者说在高斯分布下PCA效果最佳，注意square-error是高斯假设下的一个特例
  - 虽然在最小重建误差的意义下PCA是最优的特征提取工具，但没有保证可以使得分类最优
  - 高维协方差矩阵的特征值分解可以利用迭代法求解，power method
  - PCA有时也被成为K-L变换（Karhunen-Loeve transform），Hotelling变换等
  - PCA是历史最悠久的多元数据分析的工具，最早于1901年由Pearson提出，后来1963年进行了推广

特征提取（feature extraction）：ICA

- 独立元分析ICA（Independent Component Analysis）
  - 求解线性变换$Y = WX$，使得$Y = \{y_1, y_2 \dots y_M\}$的各个分量之间相互独立
  - 和PCA不同，ICA追求的是输出的变量相互独立，而非仅不相关，因此，ICA需要利用数据分布的高阶统计信息而非仅二阶信息
  - 在很多（分类）应用中，ICA提取的特征好于PCA



特征提取（feature extraction）：ICA

- 鸡尾酒会问题（cocktail party problem）
  - 在一个酒会中，人们可以分辨出不同的声音，why?
  - 以两个人为例说明这个过程，它们的声音信号分别表示为$s_1(t), s_2(t)$
    - 声音的混合过程: $\boldsymbol{x} = \boldsymbol{As}$

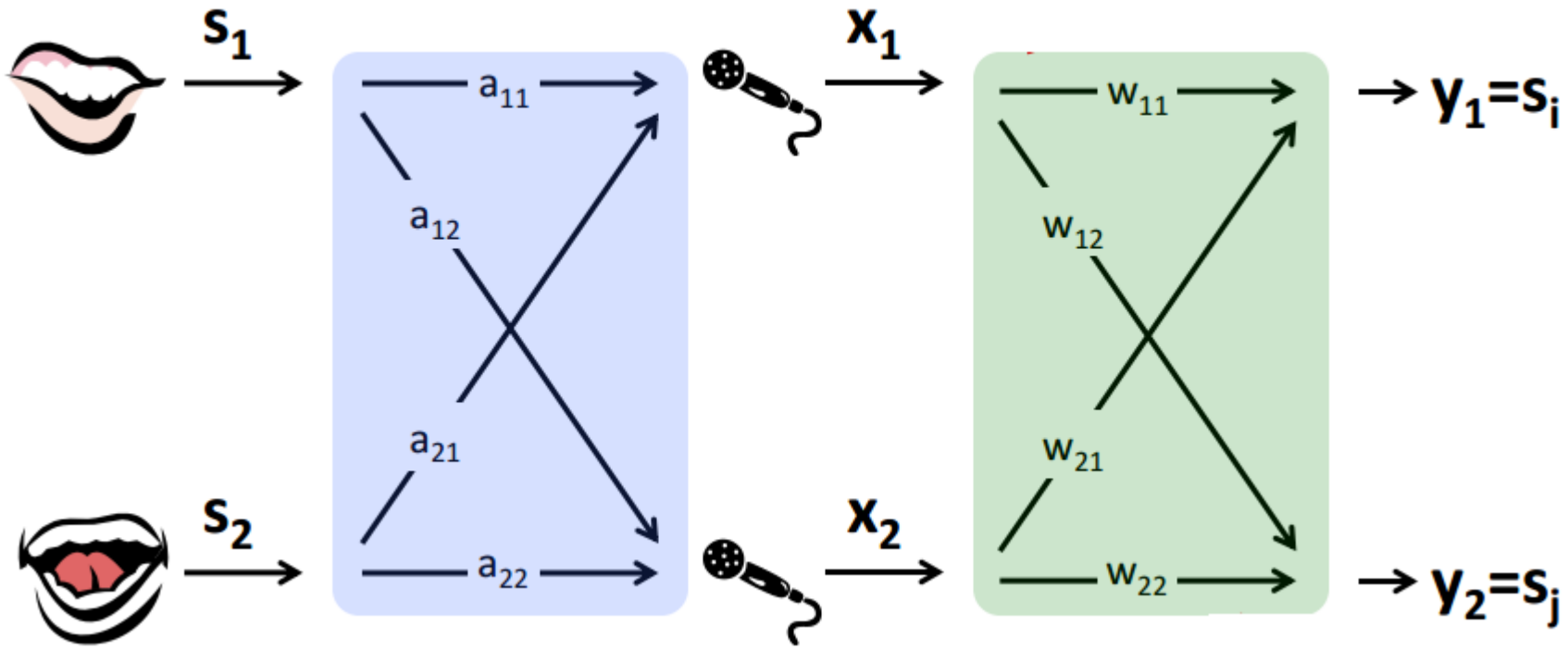
$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t)$$
$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t)$$

    - 将原始声音还原的过程: $\boldsymbol{y} = \boldsymbol{Wx} = \boldsymbol{WAs}$

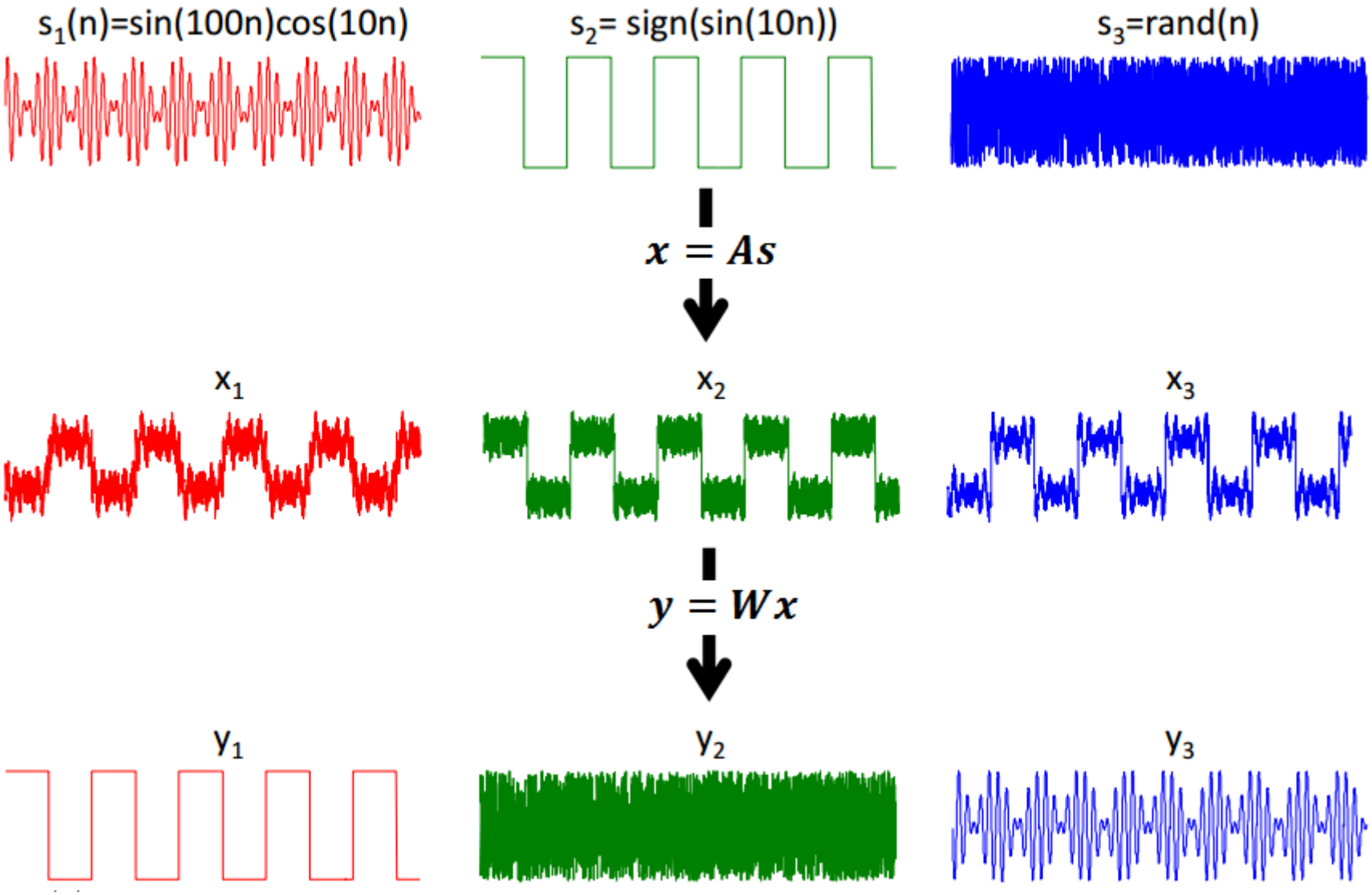
$$y_1(t) = w_{11}x_1(t) + w_{12}x(t)$$
$$y_2(t) = w_{21}x_1(t) + w_{22}x_2(t)$$

特征提取（feature extraction）：ICA

- 鸡尾酒会问题（cocktail party problem）
  - 在一个酒会中，人们可以分辨出不同的声音，



特征提取（feature extraction）：ICA

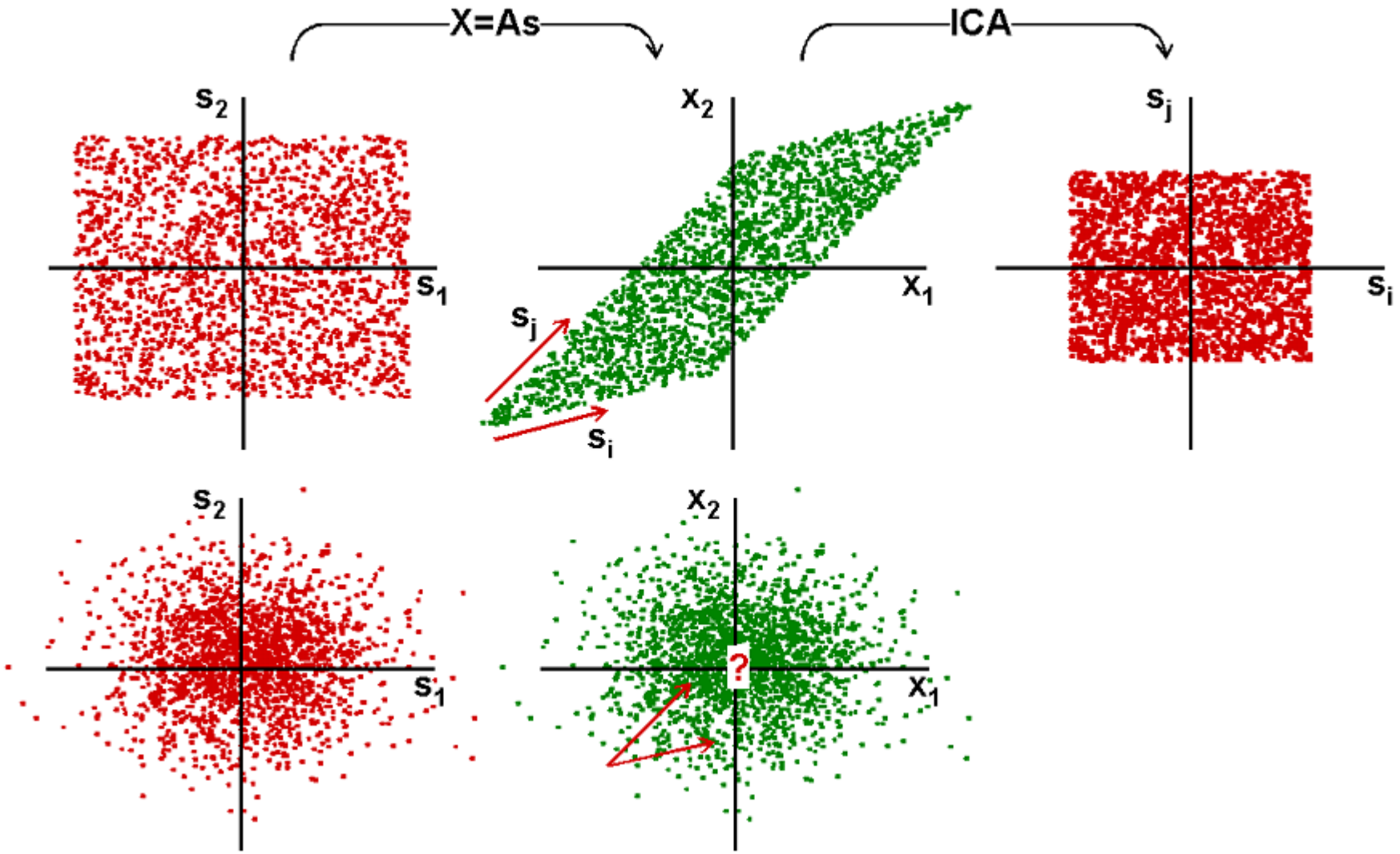


特征提取（feature extraction）：ICA

几个概念：

- 由于*W*和*Y*都是未知的，导致了两个ICA的两类不确定性
  - *Y*的幅值Scale不确定，一般将Y的方差限制为1
  - *Y*的排列顺序不确定
- 独立 V.S. 不相关
  - 独立：$p(y_1, y_2) = p(y_1)p(y_2)$
  - 不相关：$E(y_1 y_2) = 0$
  - 独立意味着不相关，但反之不亦然
  - 高斯分布意味着独立等价不相关
- 信号需是非高斯的
  - 高斯的话可以利用PCA！
  - 高斯分布的对称性导致ICA失效

特征提取（feature extraction）：ICA



特征提取（feature extraction）：ICA

不同的准则

- 最小互信息（MMI: Minimum Mutual Information）

$$I(p,q) = KL(p,q) = \int p(x) log \frac{p(x)}{q(x)} dx \geq 0, \qquad I(p,q) = 0 \Rightarrow p = q$$

$$p \equiv p(y), q \equiv \prod_i p(y_i), I(p,q) = 0 \Rightarrow p(y) = \prod_i p(y_i)$$

- 非高斯最大化（Non-Gaussianity Maximization）
  - $y = Wx = WAs$, 每一维*y*可以看成是*s*的线性组合
  - 根据中心极限定理，*y*要比*s*更趋于高斯分布，而只有当y等于s的时候才具有最低的高斯性
  - 非高斯性一般利用信号的四阶统计量，峭度（kurtosis）来衡量

$$kurt(y) = E(y^4) - 3\left(E(y^4)\right)^2$$

- “一比特”匹配定理[Liu et al. 2004]指出:
  - 在某些合理的假设前提下，**针对于ICA的非高斯的最大化等价于最下互信息**
- 最大似然...

特征提取（feature extraction）：ICA

不同的准则

- 最小互信息（MMI: Minimum Mutual Information）

$$I(p,q) = KL(p,q) = \int p(x) log \frac{p(x)}{q(x)} dx \geq 0, \qquad I(p,q) = 0 \Rightarrow p = q$$

- 非

gaussian

super-gaussian

sub-gaussian

寸候才具

-

- 最大似然...

特征提取（feature extraction）：ICA

算法

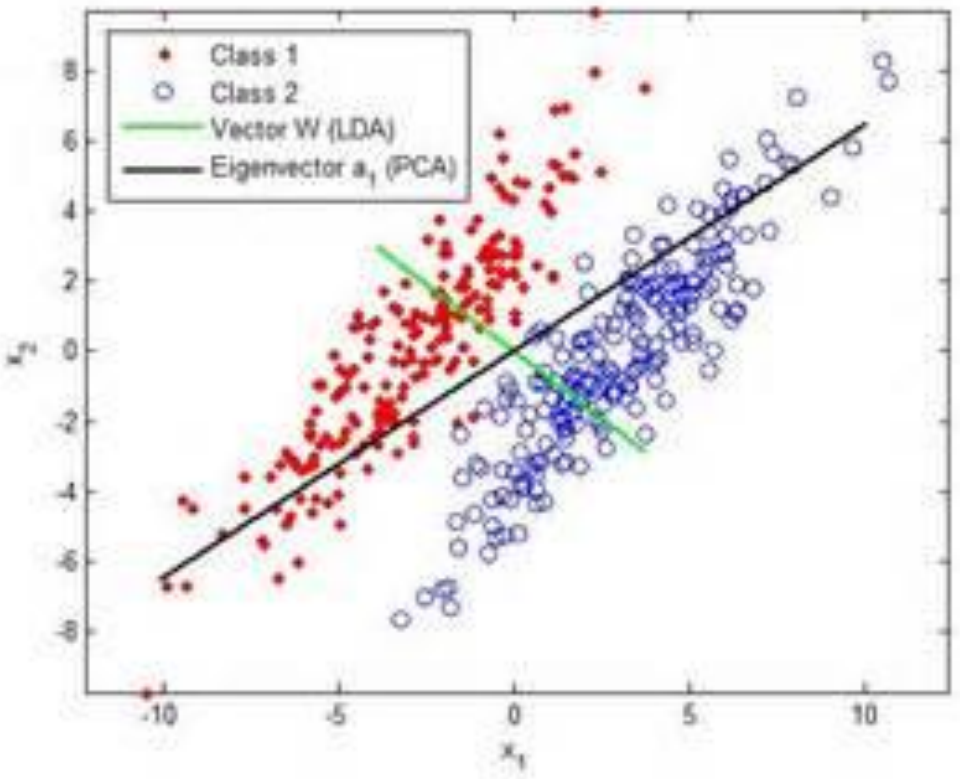
- Natural Gradient[amari. 1998]
  - 梯度下降法的一种，收敛速度较快
- Learned parametric model[Xu.1999]
  - 自动学习了待分离信号是超高斯或亚高斯
- FastICA [Hyvarinen]
  - 基于非高斯最大化的准则，基于最大负熵的非高斯最大化
    - 熵（Entropy）：$H(Y) = -p(y)\log\int p(y)\,dy$,高斯分布具有最大熵
    - 负熵（Negentropy）：$J(y) = H(y_G) - H(y), y_G$是和$y$具有相同方差的高斯分布变量
  - 负熵的近似（需要知道分布！）
    - $J(y) \propto [E\{G(y)\} - E\{G(v)\}]^2, v$是标准高斯分布变量，$G$是一个非二次方程（某种程度上反映了分布函数，非高斯！），但如何选G其实没有很好的标准

特征提取（feature extraction）：LDA

- 线性判别分析LDA（Linear Discriminate Analysis）
  - LDA也称为Fisher Discriminate Analysis，由Fisher 1936年提出，而后由Belhumeur于1996年引入到模式识别领域[Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *ECCV'96*]
  - 和PCA、ICA不同，LDA是一种监督学习方式下的特征提取方式，也可以直接用于分类
  - LDA追求的是在特征提取（降维）的过程中最大程度保持分类的信息

特征提取（feature extraction）：LDA

- 线性判别分析LDA（Linear Discriminate Analysis）
  - LDA
    193(
    式识
    Usir
  - 和P(
    征提
  - LDA
    度保



Fisher
引入到模
ognition
"96]
式下的特
中最大程

特征提取（feature extraction）：LDA

两类问题

- 给定一组n个D维样本$\{x_1, x_2, \dots x_n\}$，其中n_1个样本属于ω_1类，其中n_2个样本属于ω_2类
- 我们需要找出一个点投影$y = w^T x$，使得两类样本得到的y尽可能地分开（如何定义？）
- 利用它们投影均值之间的距离

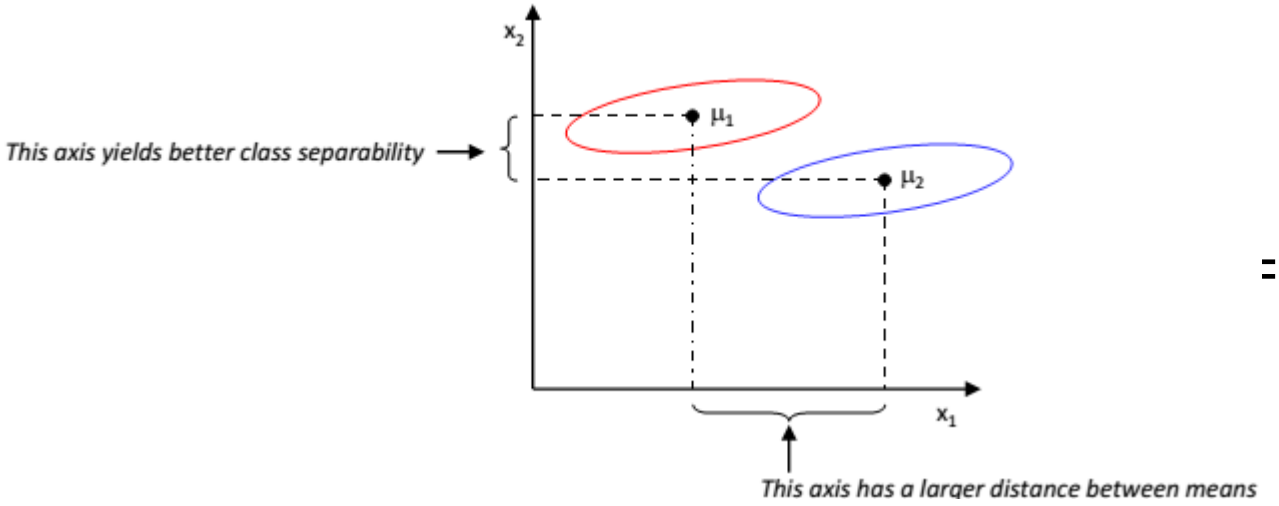
$$\mu_i^y = \frac{1}{n_i}\sum_{y\in\omega_i} y = \frac{1}{n_i}\sum_{x\in\omega_i} w^T x = w^T \frac{1}{n_i}\sum_{x\in\omega_i} x = w^T \mu_i^x$$

$$J(w) = \left|\mu_1^y - \mu_2^y\right| \quad ?$$

特征提取（feature extraction）：LDA

两类问题

- 给定一组n个D维样本$\{x_1, x_2, \ldots x_n\}$，其中n_1个样本属于ω_1类，其中n_2个样本属于ω_2类
- 我们需要找出一个点投影$y = w^T x$，使得两类样本得
- 利



μ_i^y $= w^T \mu_i^x$

特征提取（feature extraction）：LDA

Fisher准则

- 利用投影后样本y的类内分散度（Scatter，也即是方差）来归一化上述距离

$$J(w) = \frac{\left|\mu_1^y - \mu_2^y\right|^2}{s_1^y + s_2^y},$$

$$s_i^y = \frac{1}{n_i}\sum_{y\in\omega_i}(y-\mu_i^y)^2 = w^T\frac{1}{n_i}\sum_{y\in\omega_i}(x-\mu_i^x)(x-\mu_i^x)^T w = w^T S_i^x w$$

$$s_1^y + s_2^y = w^T S_1^x w + w^T S_2^x w = w^T S_W w$$

$$\left|\mu_1^y - \mu_2^y\right|^2 = w^T S_B w, S_B = (\mu_1^x - \mu_2^x)(\mu_1^x - \mu_2^x)^T$$

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

- 我们称S_W为类内方差，S_B类间方差

特征提取（feature extraction）：LDA

问题求解

- $\frac{dJ}{dw} = \frac{d}{dw}\left[\frac{w^T S_B w}{w^T S_W w}\right] = \frac{2S_B w}{w^T S_W w} - \frac{2w^T S_B w S_W w}{(w^T S_W w)^2}$

$$\frac{dJ}{dw} = 0 \Rightarrow S_B w = \frac{w^T S_B w S_W w}{w^T S_W w} = J S_W w \Rightarrow S_W^{-1} S_B w = J w$$

  - 特征值分解问题
- 另一条思路，因为w的幅度变化不影响J，所以我们可以将问题转化为：

$$min. -w^T S_B w, \qquad s.t. w^T S_W w = 1$$

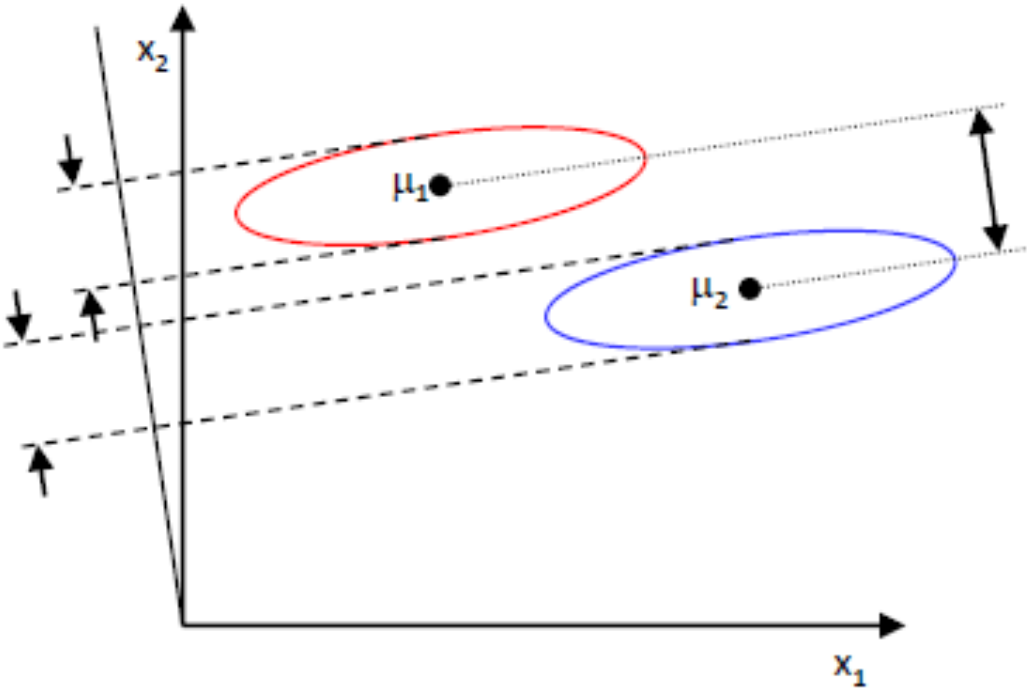
  - 利用拉格朗日乘子，也可以直接得到上述结论（类似前面的PCA求解）

特征提取（feature extraction）：LDA

问题求解

- $\frac{dJ}{dw} = \frac{d}{dw}\left[\frac{w^T S_B w}{w^T S_W w}\right]$ $2S_B w$ $2w^T S_B w S_W w$

$\frac{dJ}{dw} = 0 \Rightarrow$ $_B w = Jw$ 特征值分解



- 另一条 所以我们可以将

  - 利用 论（类似前面

特征提取（feature extraction）：LDA

多类（C类）问题

- 求解$C-1$个投影向量$W=[w_1,w_2,\dots,w_{c-1}]$，从而得到$C-1$个投影$y=[y_1,y_2,\dots,y_{c-1}]$结果

$$y=Wx \text{ 或 } y_i=w_ix$$

- 类似于两类问题
  - 定义类内方差为$S_w=\sum_{i=1}^{C}S_i$
    - $S_i=\frac{1}{n_i}\sum_{x\in\omega_i}(x-\mu_i)\,(x-\mu_i)^T,\ \mu_i=\frac{1}{n_i}\sum_{x\in\omega_i}x$
  - 定义类间方差为$S_B=\frac{1}{n}\sum_{i=1}^{C}n_i(\mu_i-\mu)\,(\mu_i-\mu)^T$
    - $\mu=\frac{1}{n}\sum_{j=1}^{n}x_j$
- 类似地，投影后y的类内和类间方法可以求出等于

$$S_W^y=W^TS_wW,S_B^y=W^TS_BW$$

特征提取（feature extraction）：LDA

多类（C类）问题

- 不再是标量，可以定义指标为

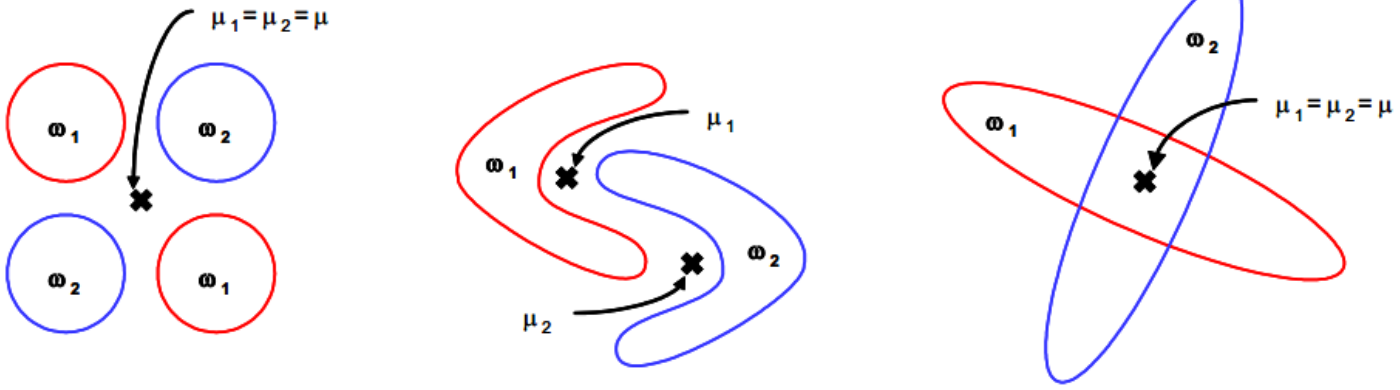
$$J_1(W) = \frac{|W^T S_B W|}{|W^T S_w W|} \text{或} J_2(W) = \frac{tr\{W^T S_B W\}}{tr\{W^T S_w W\}}$$

- $\frac{d}{dW} J_2(W) = \frac{2S_B W}{tr\{W^T S_w W\}} - \frac{2tr\{W^T S_B W\} S_w W}{tr\{W^T S_w W\}^2} = 0 \Rightarrow S_W^{-1} S_B W = J_2 W$
- 对$J_1(W)$操作也能得到同样的结果（课后自己推导试试）
- 可以类似于PCA对样本进行降维
- 对于（C类）分类问题，一般采用1 v.s. all构建C个分类器；或者两两之间构建$\frac{C(C-1)}{2}$个分类器

特征提取（feature extraction）：LDA

LDA小结

- 相对于PCA，LDA利用了类别的信息，在某些情况下可以取得比PCA、ICA等更好的分类效果；并且LDA可以直接用于分类
- LDA产生至多$(C-1)$个特征（因为S_B的秩至多为$C-1$），可能会造成特征不足
- LDA比较适合每类都是单模分布的情况，否则LDA有可能会失效
  - Kernel LDA或者称为kernel Fisher DA
- Otsu's二值化阈值选择利用了一维向点投影的LDA



特征提取（feature extraction）：Local PCA

局部主元分析 Local PCA，或被称为混合PCA模型（mixture of PCA）

- PCA → local PCA 类似于 Gaussian → mixture of Gaussian，也即数据分布呈现多态（multi mode）时，单个分布模型不足以描述数据的结构
  - 只要MoG中高斯的数目足够多，理论上MoG可以描述任意的分布

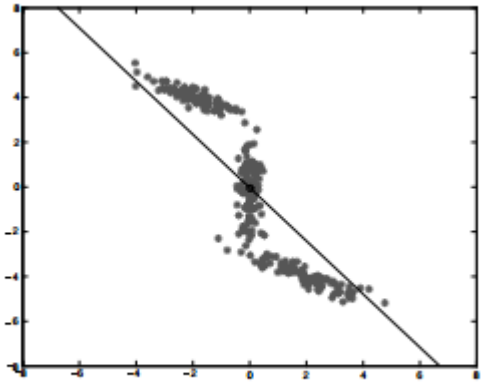


Fig. 1. PCA description.

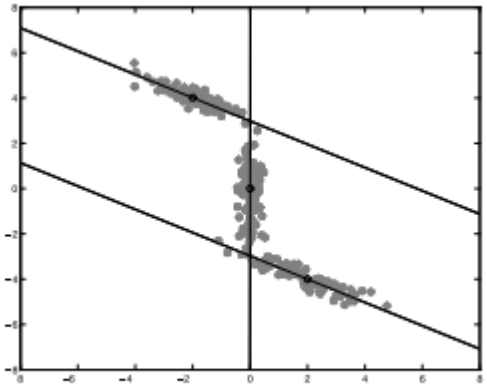


Fig. 2. Local PCA description.

特征提取（feature extraction）：Local PCA

Local PCA算法

- 第一种算法（间接算法）
  1. 估计MoG模型
  2. 对每个高斯分布特征值分解得到PCs
- 优点：简单易懂
- 缺点：分两步走，第一步需要估计数据的协方差矩阵，没有考虑数据结构
  - 协方差矩阵的参数个数为$\frac{n(n+1)}{2}$
  - 如果直接估计PCs，那么参数个数将会大幅减少！（Hinton 1995, Xu 1996）

特征提取（feature extraction）：Local PCA

Local PCA算法

- 第二种算法（直接算法）
  - 将MoG中的协方差矩阵写成分解的形式

$$p(x) = \sum_{i=1}^{k} \alpha_i G(x|\mu_i, \Sigma_i)$$

$G(x|\mu_i, \Sigma_i)$表示高斯分布，μ_i是均值，Σ_i是协方差矩阵，$\sum_{i=1}^{k} \alpha_i = 1$

  - 将 Σ_i写成分解的形式

$$\Sigma_i = \sigma_i I + W_i \psi_i W_i^T$$

I是$n \times n$的单位阵， W_i是$n \times m$矩阵， $n \geq m$,

$\psi_i = diag\{\sigma_{i1}, \sigma_{i2}, .. \sigma_{im}\}, W_i^T W_i = I$, W_i即为PCs

特征提取（feature extraction）：Local PCA

Local PCA算法

- Generalized Expectation Maximization (EM)算法

  For each samplex_t:

  – Find the winner component c by the MAP

  $$c = argmax_i[\alpha_i G(x_t|\mu_i, \Sigma_i)]$$

  – Update $\Theta = \{\alpha_i, \mu_i, \sigma_i, W_i, \psi_i\}_{i=1}^{k}$ by gradient ascend algorithm

  $$\theta_c \leftarrow \theta_c + \eta \nabla_{\theta_c} \log([\alpha_c G(x_t|\mu_c, \sigma_c I + W_c \psi_c W_c^T)]$$

  – 利用Gram-Schmidt正交化方法将W_c正交化（或者直接将W_c在Stiefel manifold上进行优化，略…）

特征提取（feature extraction）：LLE

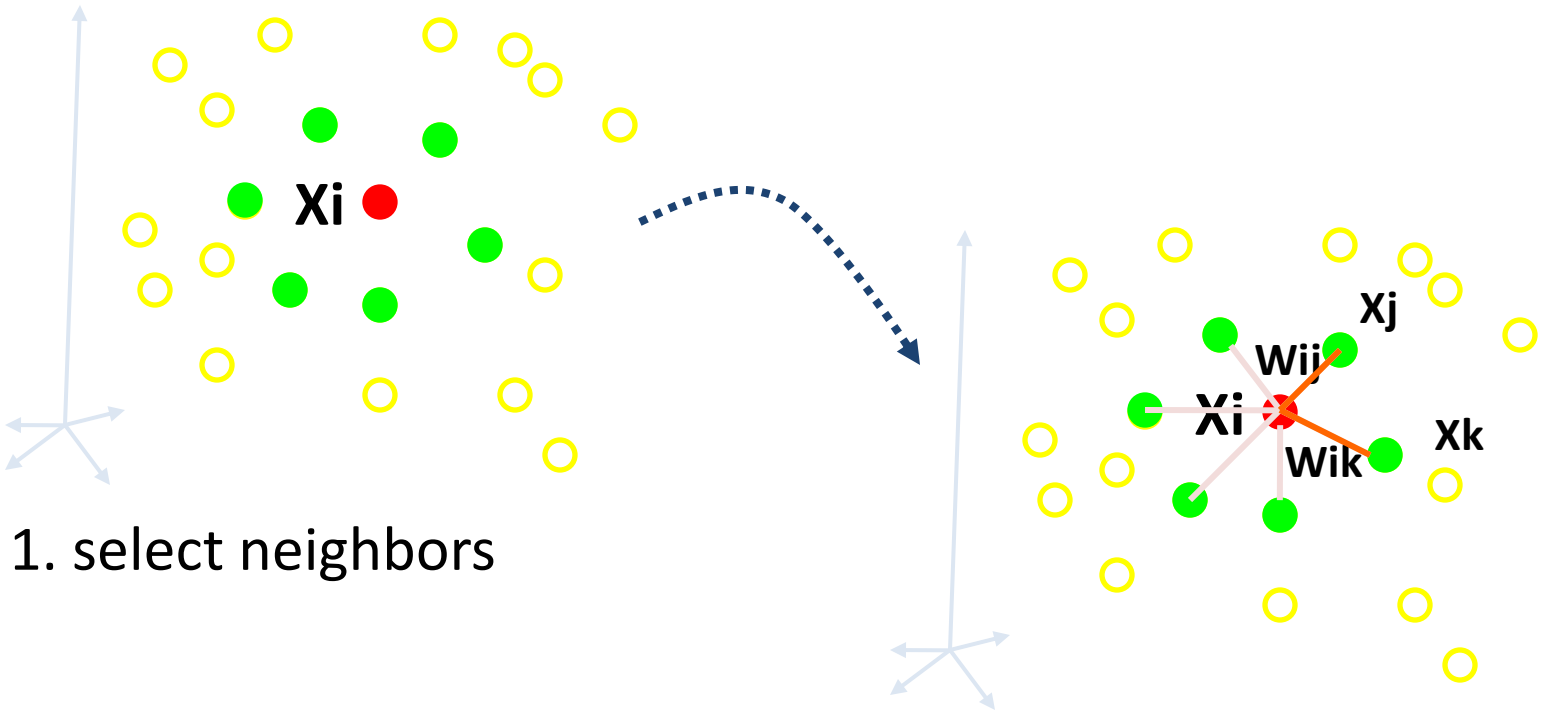
LLE（Locally Linear Embedding）:局部线性嵌入

- PCA：最大程度保持方差
- LLE：最大程度保持点间的距离，流形学习（manifold learning）的一种
  - ✓ 流形：局部具有欧几里得空间性质的空间，局部是“平坦”（flat）的，例如线，圆等
  - ✓ 嵌入：An embedding is a representation of a topological object, manifold, graph, field, etc. in a certain space in such a way that its connectivity or algebraic properties are preserved.
  - ✓ LLE是一种在高维空间中寻找低维流形嵌入的一种方法

特征提取（feature extraction）：LLE

原理

- 假设前提：流形在局部是线性的，那么每个样本将可以利用其周围的样本进行重建



1. select neighbors

2. reconstruct with linear weights

特征提取（feature extraction）：LLE

算法

- Step 1:邻域选择：多种方法，例如样本个数、样本距离等
  - 样本个数k不能小于数据的维度N：dense data
  - k过大容易忽视非线性，过小则容易将流形割裂
- Step 2:计算重建权重（reconstruction weights）

$$min. \left\| x_i - \sum_{j \in J_i} w_{ij} x_j \right\|^2, s.t. \sum_{j \in J_i} w_{ij} = 1$$

等价于：$min. w_i^T Q_i w_i,\ s.t. w_i^T \Gamma = 1$

  - 这里$\Gamma = [1, \dots, 2]^T, Q_i = G_i^T G_i, G_i = [x_{i1} - x_i, \dots, x_{ik} - x_i]$
  - 利用拉格朗日算子，得到

$$w_i^* = \frac{Q_i^{-1}\Gamma}{\Gamma^T Q_i^{-1}\Gamma}$$

特征提取（feature extraction）：LLE

算法

- Step 3:嵌入，最小化局部线性重建误差，得到数据的低维表示

$$\Phi(Y) = \sum_i \left\| y_i - \sum_{j \in J_i} w_{ij}\, y_j \right\|^2$$

  - 固定w，求解Y,等价于

$$\Phi(Y) = \sum_{ij} M_{ij} \left(y_i \cdot y_j \right),$$

$$M_{ij} = \delta_{ij} - w_{ij} - w_{ji} + \sum_k w_{ki} w_{kj}$$

  - The optimal embedding is found by computing the bottom d eigenvector of M, d is the dimension of the embedding

Illustration

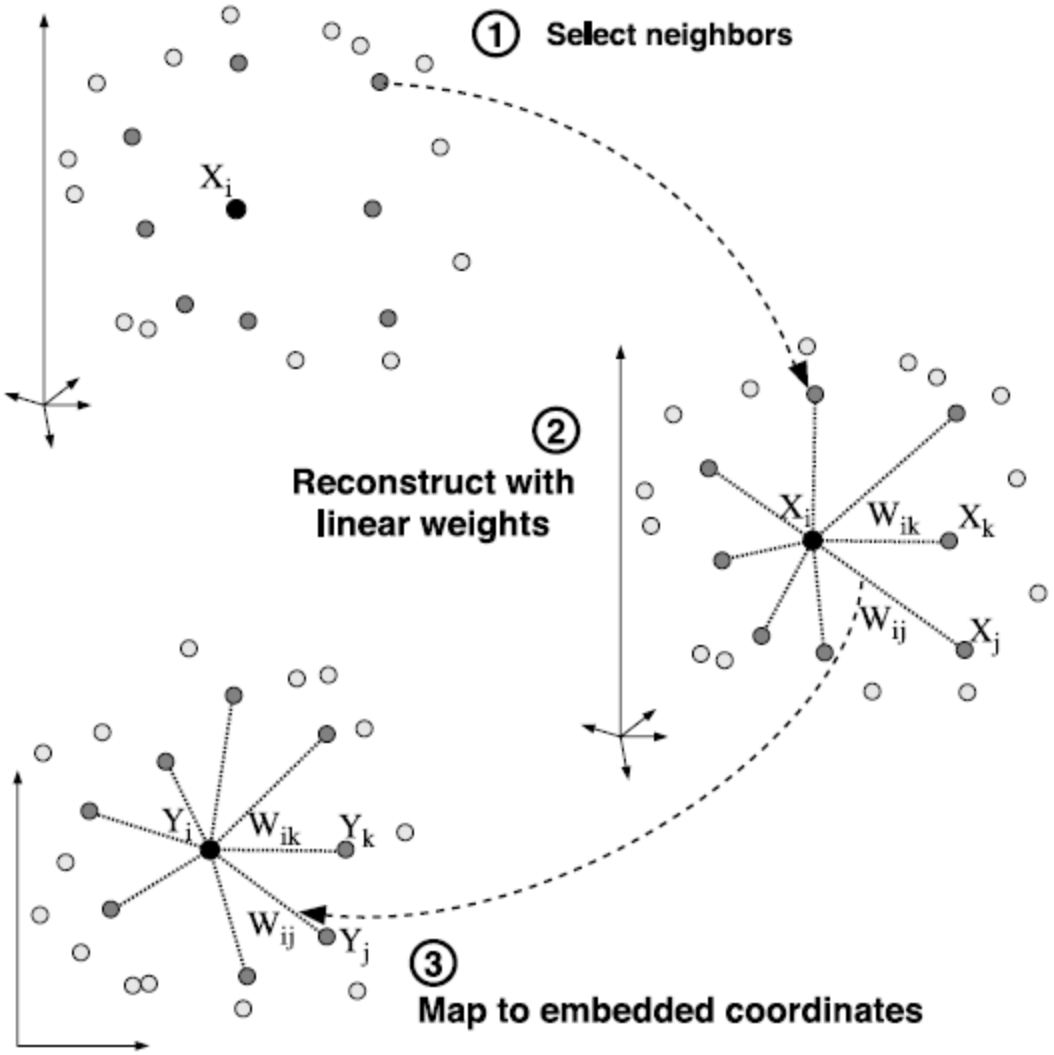


Fig. 2. Steps of locally linear embedding: (1) Assign neighbors to each data point $\vec{X}_i$ (for example by using the K nearest neighbors). (2) Compute the weights W_{ij} that best linearly reconstruct $\vec{X}_i$ from its neighbors, solving the constrained least-squares problem in Eq. 1. (3) Compute the low-dimensional embedding vectors $\vec{Y}_i$ best reconstructed by W_{ij}, minimizing Eq. 2 by finding the smallest eigenmodes of the sparse symmetric matrix in Eq. 3. Although the weights W_{ij} and vectors Y_i are computed by methods in linear algebra, the constraint that points are only reconstructed from neighbors can result in highly nonlinear embeddings.

特征提取（feature extraction）

- 其它一些常见的特征提取方法
  - Multidimensional scaling，MDS
  - Auto-encoder
  - Exploratory Projection Pursuit
  - ISOmap
  - Laplacian eigenmaps，LE
  - ……

特征选择

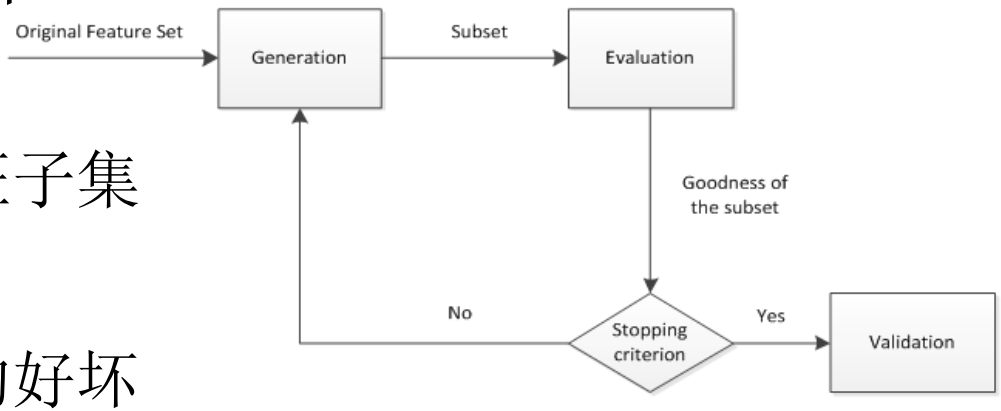
- 给定一组特征$X_N = \{x_i | i = 1 \dots N\}$，找到它的一个子集$Y_M, M < N$最大化某目标函数

  $$Y_M = \{x_{i_1}, x_{i_2}, \dots x_{i_M}\} = \text{argmax}_{M,i_M} J(Y_M)$$

- Why not feature extraction?事实上，特征选择是一类特殊的特征提取（想象一个非常稀疏的线性变换）
  - 某些特征无法变换（譬如非数值型，或者离散型）
  - 特征抽取有时会丢失掉原始特征物理意义（譬如高度、长度等）
  - 正因为它是一类非常特殊的特征抽取，特征选择的方法和特征抽取大相庭径，事实上不涉及变换
  - 尤其适用于那些特征维数很多，但样本数目较少的情况，如DNA microarray分析（几千维特征，几十个数据！）

特征选择

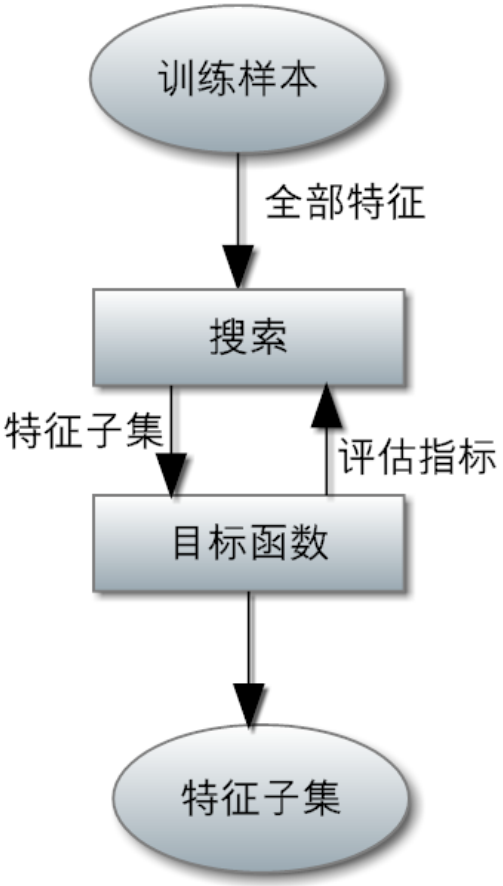
- 特征选择的一般步骤
  - 特征产生
    - 为评价函数提供特征子集
  - 特征评价
    - 评价一个特征子集的好坏
  - 停止策略
    - 一般提前设定一个阈值



(Dash and Liu 1997)

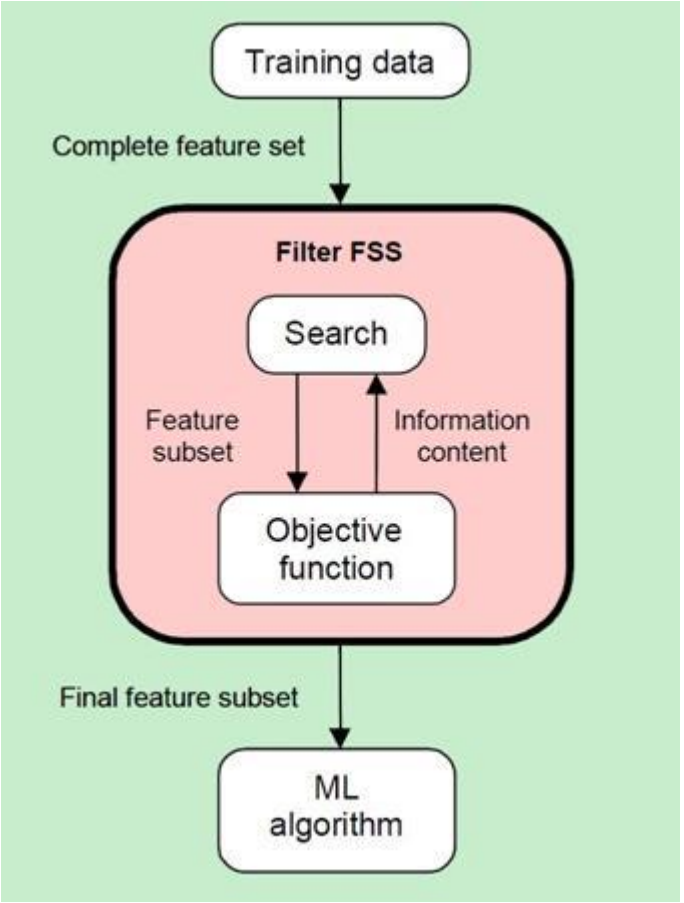
特征选择

- 两大要素
  - 目标函数（评价指标）
  - 搜索策略
- 显然，对于从N个特征中选取出M个特征理论上是一个组合优化问题，复杂度为$\binom{N}{M}$，N=50, M=20 →复杂度为10^{32}
  - 需要一种合理的搜索策略
- 需要有合理的目标函数或评价指标来判断结果，并指导搜索方向，依据不同的评价指标
  - Filter方法：利用数据本身
  - Wrapper方法：利用predictor



特征选择 – 目标函数

- 筛选器（Filter）方法
  - evaluate subsets by their information content, e.g., interclass distance, statistical dependence or information-theoretic measures
  - 相关性、距离、一致性等
  - 针对性差，推广能力较强，计算量较小



(Ricardo Gutierrez-Osuna 2008)

特征选择 – 目标函数

- 相关性
  - 好的特征集包含的特征和类别具有大的相关性（预测能力），互相之间的相关性较小（冗余度小）
  - 线性相关性

$$J(Y_M) = \frac{\sum_{i=1}^{M} \rho_{ic}}{\sum_{i=1}^{M} \sum_{j=i+1}^{M} \rho_{ij}}$$

ρ_{ic}表示特征i和类别标签之间的相关性，ρ_{ij}表示特征i, j之间的相关性

  - 非线性，利用Y_M和C互信息
    - 互信息表示已知Y_M，C减少的不确定性

$$J(Y_M) = I(Y_M; C) = H(C) - H(C|Y_M) = \sum_{c=1}^{C} \int_{Y_M} p(Y_M, \omega_c) log \frac{p(Y_M, \omega_c)}{p(Y_M)p(\omega_c)} dy$$

特征选择 – 目标函数

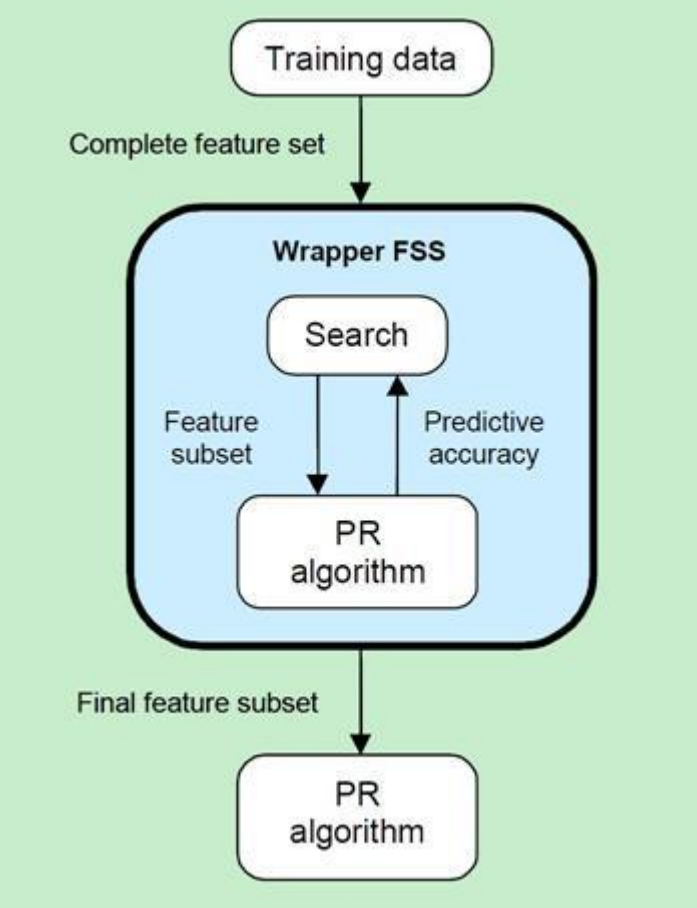
- 非线性（cont.）
  - 需要计算多元密度函数$p(Y_M)$和$p(Y_M, \omega_c)$，实际中一般由下面近似公式代替[Battiti,1994]

$$J(Y_M) = \sum_{m=1}^{M} I(x_{i_m}; C) - \beta \sum_{m=1}^{M} \sum_{n=m+1}^{M} I(x_{i_m}; x_{i_n})$$

- 一致性（consistency）：若样本1与样本2属于不同的分类，但在特征A、B上的取值一样，那么特征子集{A，B}不应该选入特征子集。
- 距离（distance metric）：好的特征子集应该使得属于同一类的样本距离尽可能小，属于不同类的样本之间的距离尽可能远。
  - 常用的距离度量（相似性度量）包括欧氏距离、马氏距离等

特征选择 – 目标函数

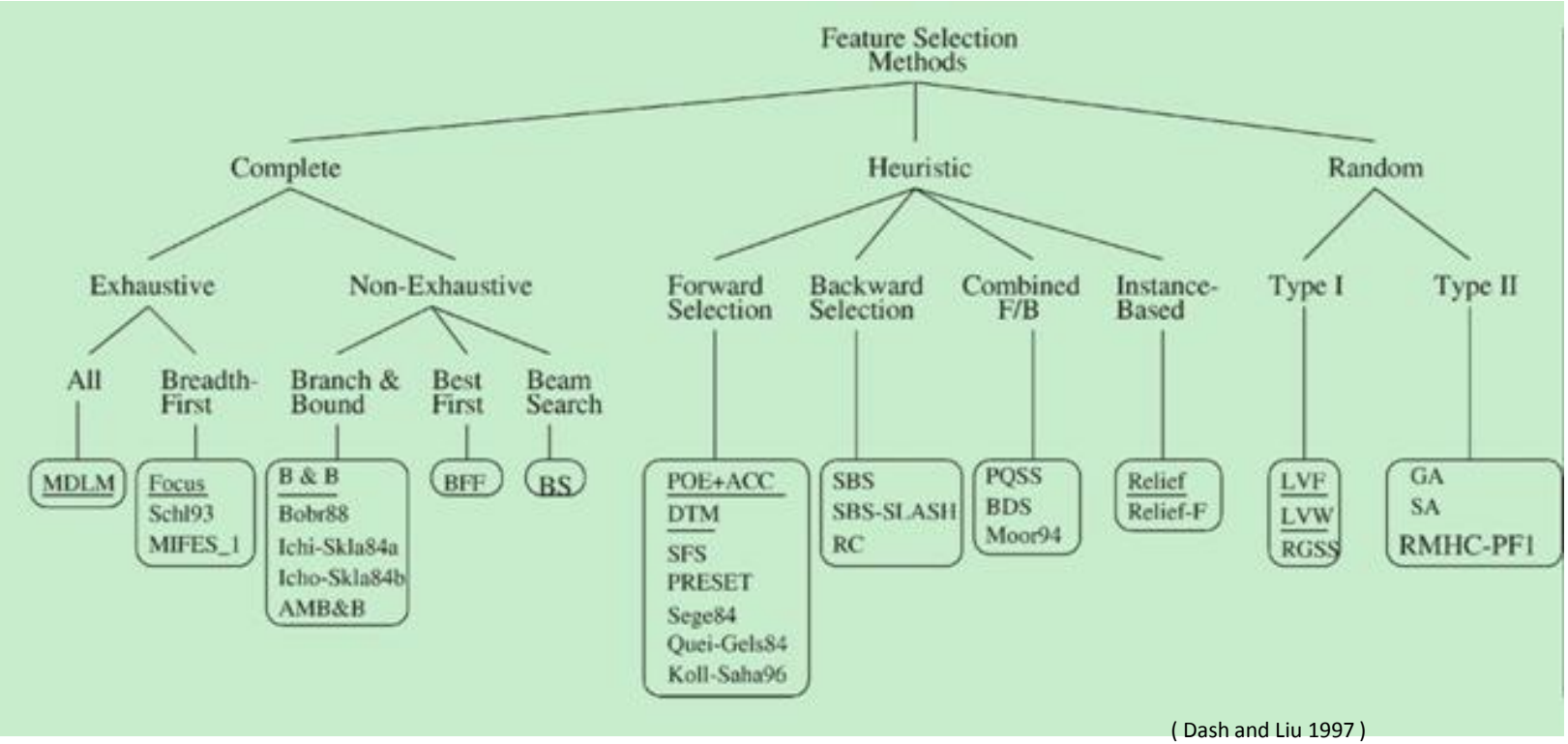
- 封装器（Wrapper）方法
  - use a classifier to evaluate subsets by their predictive accuracy (on test data) by statistical resampling or cross-validation
  - 使用特定的分类器，用给定的特征子集对样本集进行分类，用分类的精度来衡量特征子集的好坏。
  - 针对性强，但推广能力较差，计算量较大



(Ricardo Gutierrez-Osuna 2008)

特征选择 -搜索策略

- 搜索策略



(Dash and Liu 1997)

特征选择 -搜索策略

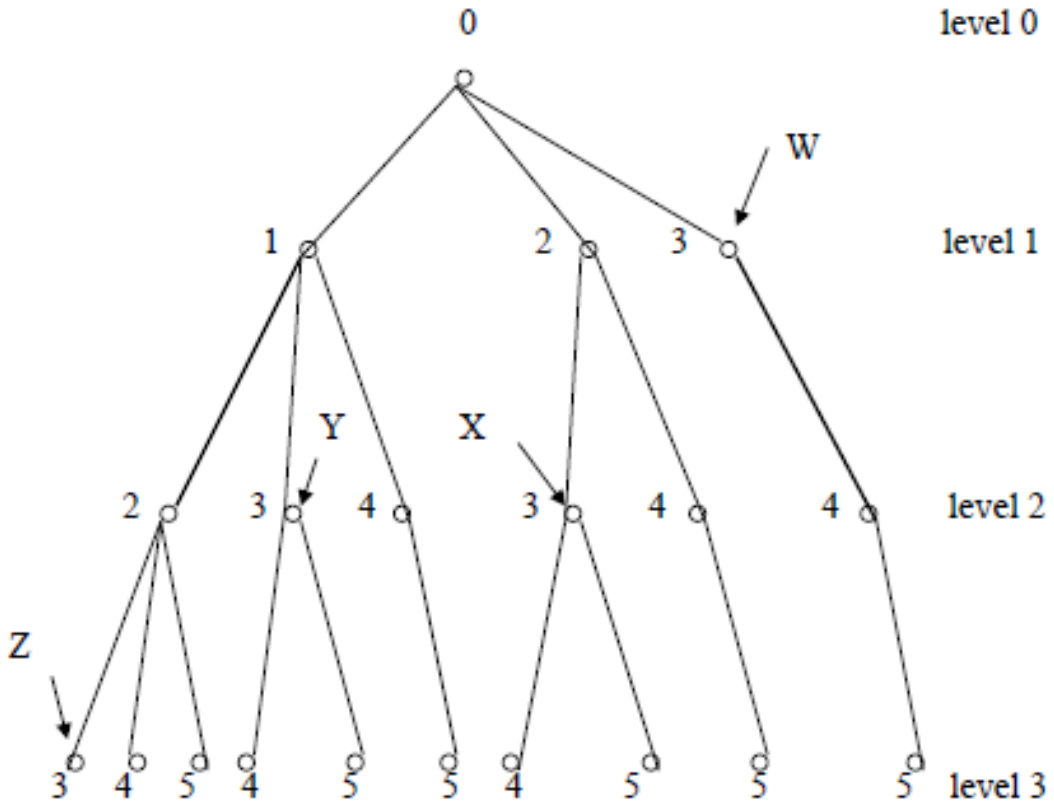
- 搜索策略
  - 完全搜索（exponential algorithms）
    - 完全穷搜：广度优先搜索（复杂度大，不实用）
    - 非完全穷搜：分支定界（将某些不太可能得到比当前解更好解的分支剪除掉）
    - 集束搜索（beam search）: best-first search with breadth-first search
  - 启发式搜索（sequential algorithms）
    - 序列前向选择(SFS, Sequential Forward Selection)
    - 序列后向选择(SBS, Sequential Backward Selection)
    - 双向搜索(BDS, Bidirectional Search)
    - 增L去R选择算法(LRS, Plus-L Minus-R Selection)
    - 序列浮动选择(Sequential Floating Selection)
  - 随机算法（randomized algorithms）
    - 随机产生序列选择算法(RGSS, Random Generation plus Sequential Selection)
    - 模拟退火算法(SA, Simulated Annealing)
    - 遗传算法(GA, Genetic Algorithms)

特征选择 -搜索策略

- 分支定界（branch and bound）
  - 一种相对较快的全局最优法，但最差状态也会堕落为穷举法
  - 用于特征选择时，需要目标函数满足单调性条件，即函数值随着特征数目的增加
    $J(Y_M) < J(Y_{\tilde{M}})\ if\ Y_M\ is\ a\ subset\ of Y_{\tilde{M}}$
  - 定界、减支：把已有搜索到叶子节点的目标函数值作为bound，如果有节点对应的值小于它，那么此节点以下的所有节点则无需搜索（减掉）

特征选择 -搜索策略

- 例子： N=5, M=2 （表示的为待删除特征）



1. Initialize $\alpha = -\infty, k = 1$
2. Generate successors of the current node and store them in $LIST(k)$
3. Select new node
   if $LIST(k)$ is empty
      go to 5
   else
      $i_k = \arg\max_{j \in LIST(k)} J(x_{i_1}, x_{i_2} \dots x_{i_{k-1}}, j)$
      delete i_k from $LIST(k)$
4. Check bound
   if $J(x_{i_1}, x_{i_2} \dots x_{i_k}) < \alpha$
      go to 5
   else if $k = M'$ (we have the desired number of features)
      go to 6
   else
      $k = k + 1$
      go to 2
5. Backtrack to lower level
      $k = k - 1$
      if $k = 0$
         terminate algorithm
      else
         go to 3
6. Last level
      Set $\alpha = J(x_{i_1}, x_{i_2} \dots x_{i_{k-1}}, j)$ and $Y_{M'}^* = \{x_{i_1}, x_{i_2} \dots x_{i_k}\}$
      go to 5

特征选择 -搜索策略

- 加入一个冗余因子，使得BB可以通过某些需要强制剪枝的节点，因此某种程度上放松了单调性的要求，称为Approximate monotonicity B&B (AMB&B)

- 集束搜索（beam search）
  - 最佳优先搜索基础上限制了搜索的范围（例如只是列出来最佳的前几个而非全部）
  - 最佳的状态放在队列的最前端，每次迭代时，BS计算所有的可能的加1个特征的状态
  - 如果不限制队列的长度，变成了穷搜
  - 如果限制队列长度为1，变成了SFS
  - 不能保证搜索到最优解

特征选择 -搜索策略

- 序列前向搜索（Sequential Forward Search: SFS）
  - 从空集开始，逐个地添加特征，使得当前的目标函数最优
  - 算法：

1. Start with the empty set $Y_0 = \{\emptyset\}$
2. Select the next best feature $x^+ = \arg\max_{x \notin Y_k} J(Y_k + x)$
3. Update $Y_{k+1} = Y_k + x^+;\ k = k + 1$
4. Go to 2

  - 较为适合子特征较少的情形
  - 无法删除前期选中但后来无用的特征

特征选择 -搜索策略

- 序列后向搜索（Sequential Backward Search: SFS）
  - 从满集开始，逐个地减少特征，使得当前的目标函数最优
  - 算法：

1. Start with the full set $Y_0 = X$
2. Remove the worst feature $x^- = \arg\max_{x \in Y_k} J(Y_k - x)$
3. Update $Y_{k+1} = Y_k - x^-;\ k = k + 1$
4. Go to 2

  - 较为适合子特征较多的情形
  - 无法找回已删除的特征

特征选择 -搜索策略

- 双向搜索（Bidirectional Search）
  - 集合了SFS和SBS两类搜索策略
  - SFS算法

1. Start with the empty set $Y_0 = \{\emptyset\}$
2. Select the next best feature $x^+ = \arg\max_{x \notin Y_k} J(Y_k + x)$
3. Update $Y_{k+1} = Y_k + x^+;\ k = k + 1$
4. Go to 2

  - SBS算法

1. Start with the full set $Y_0 = X$
2. Remove the worst feature $x^- = \arg\max_{x \in Y_k} J(Y_k - x)$
3. Update $Y_{k+1} = Y_k - x^-;\ k = k + 1$
4. Go to 2

特征选择 -搜索策略

- 随机产生序列选择算法(RGSS, Random Generation plus Sequential Selection)
  - 随机产生一个特征子集，然后在该子集上执行SFS与SBS算法。
  - 可作为SFS与SBS的补充，用于跳出局部最优值。
- 模拟退火算法
  - 本质也是一种贪心算法，但是它的搜索过程引入了随机因素。模拟退火算法**以一定的概率**来接受一个比当前解要差的解，因此**有可能**会跳出这个局部的最优解，达到全局的最优解
  - 一定程度克服了序列搜索算法容易陷入局部最优值的缺点，但是若最优解的区域太小，则难以求解
  - 复杂度一般较大
- 遗传算法
  - 随机产生一批特征子集，并用评价函数给这些特征子集评分，然后通过复制、交叉、突变等操作繁殖出下一代的特征子集，并且评分越高的特征子集被选中参加繁殖的概率越高。这样经过N代的繁殖和优胜劣汰后，种群中就可能产生了评价函数值最高的特征子集
  - 复杂度较大

The END, Thanks!