6. 子空间学习

黄晟

huangsheng@cqu.edu.cn

办公室：信息大楼B701

6. 1 主成分分析

黄晟

huangsheng@cqu.edu.cn

办公室：信息大楼B701

数据表征(Data Representation)

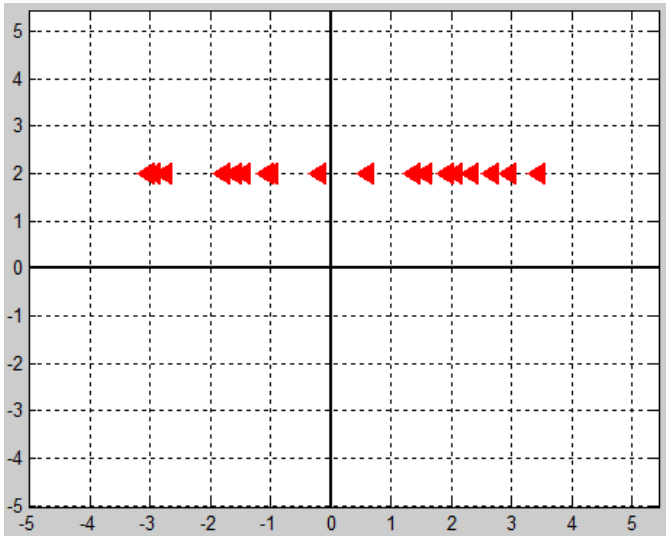
例子1

利用X描述表Y

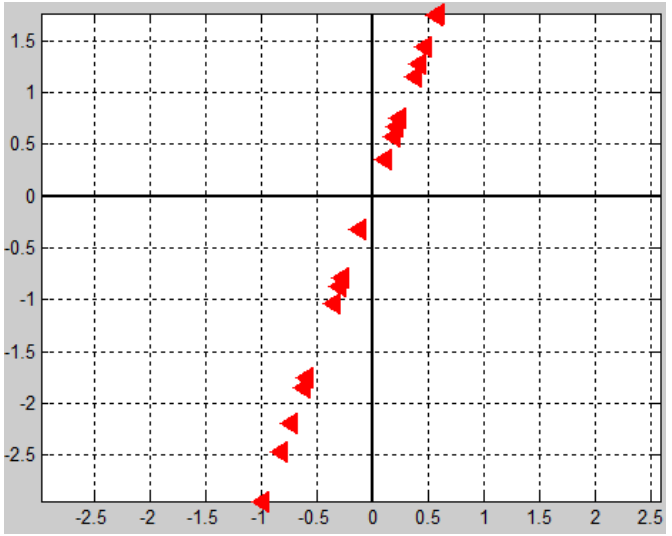
样本属性: $X = \{x_1, x_2\}$		标签: Y
千米/每小时	英里/每小时	交通工具
10	6.2137119	自行车
80	49.7096954	汽车
200	124.2742384	高铁
700	434.9598346	飞机

样本维度为2

例子2



例子3



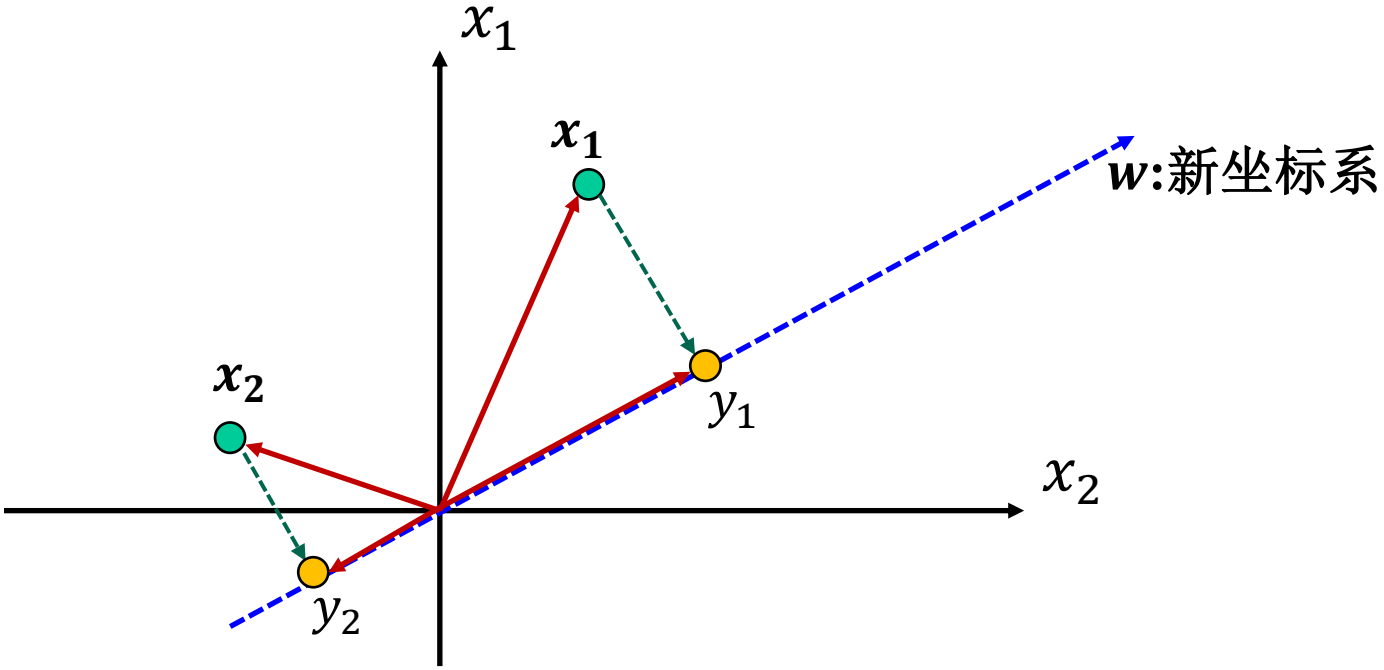
原数据表达有冗余！

降维（Dimension Reduction, DR）

- 数据冗余的弊端——维度灾难
  - 增加计算机开销：
    - 浪费更多数据存储空间
    - 需要更多计算资源（机器学习中频繁用到距离运算）
  - 采样困难
- 解决手段——降维（维度约简）
  - 寻找一组映射对样本进行重新表示(representation)
    - 原样本：$\boldsymbol{x} = [x_1; x_2; \cdots; x_d] \in \Re^d$
    - 新表示：$\mathbf{y} = [y_1; y_2; \cdots; x_m] \in \Re^m, m \ll d$
    - 降维目标：学习映射$\phi(\boldsymbol{x}) = \boldsymbol{y}$

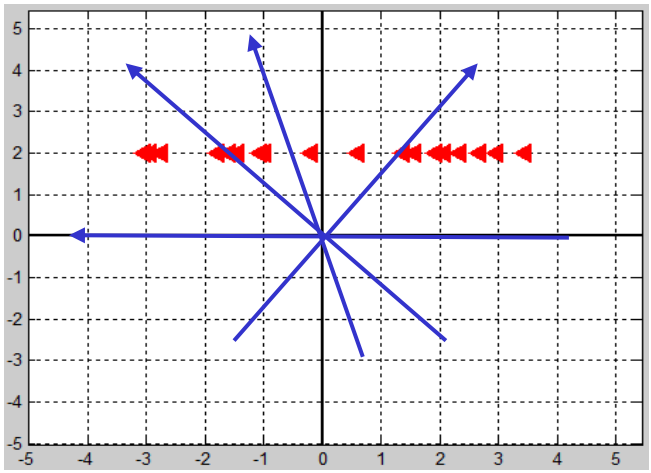
主成分分析

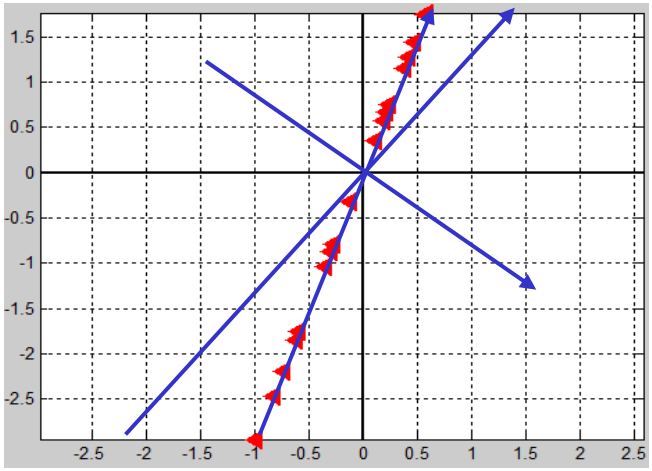
- 考虑最简单的情况：$\phi(\cdot)$为一个线性映射
  - $\phi(\cdot)$：$W^T\boldsymbol{x} = \boldsymbol{y} \in \Re^m, W = [w_1, \cdots, w_m] \in \Re^{\boldsymbol{d}\times m}$
  - **不妨假设$\boldsymbol{y}$为1维的情况：$\boldsymbol{w}^T\boldsymbol{x} = y \in \Re^1$**
- 线性映射几何解释：$\boldsymbol{w}$向量实质定义一个新坐标

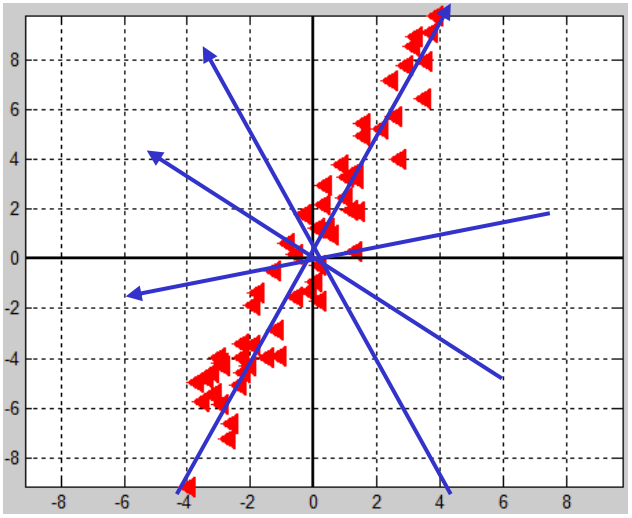


主成分分析

- 哪个坐标$\boldsymbol{w}$能保留更多信息？

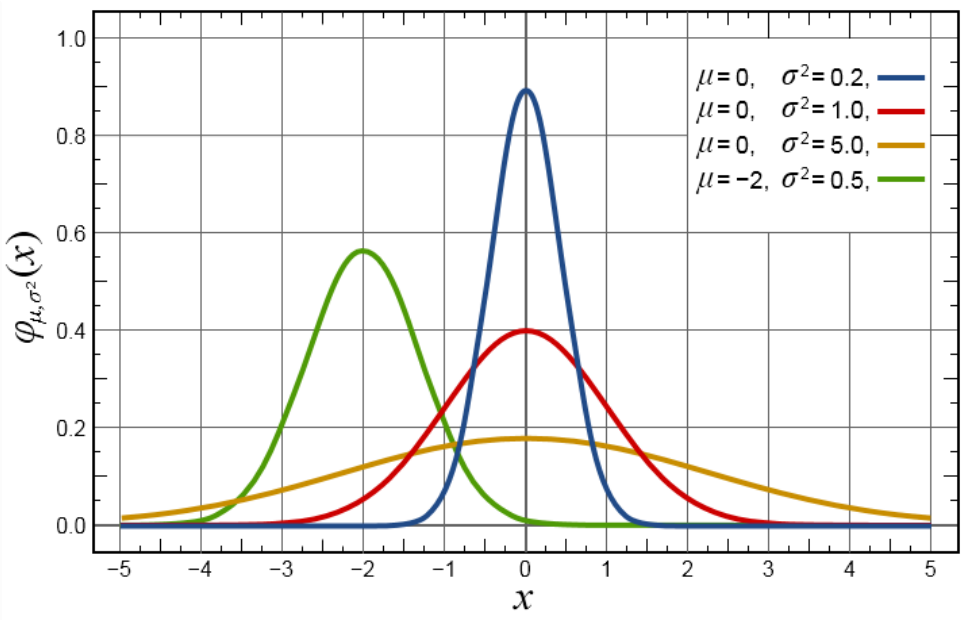






主成分分析

- 最大方差理论（信号处理领域，Signal Processing）
  - 在信号处理中认为信号具有较大的方差，噪声有较小的方差。
  - 即新坐标系上数据方差应越大越好。
- 概率解释：



概率密度函数：

$$p(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

信息熵：

$$E(x)=\int -p(x)\log(p(x))$$

方差$\sigma\uparrow$，则$E(x)\uparrow$

主成分分析

- 主成分分析(Principal Component Analysis, PCA)目标函数——新坐标系最大化数据新表征的方差:

$$\max_{\boldsymbol{w}} var(y) \coloneqq \sum_{i=1}^{n} (y_i - \bar{y})^2 \,, \text{s.t.}\, y_i = \boldsymbol{w}^T \boldsymbol{x}_i, ||\text{w}|| = 1$$

约束$||\text{w}|| = 1$为了坐标系标准化

- 矩阵化推广：选择多个坐标轴构建新坐标系.

$$\max_{W} \sum_{i=1}^{n} \|\boldsymbol{y}_i - \bar{\boldsymbol{y}}\|^2 \,, \text{s.t.}\, \boldsymbol{y}_i = W^T \boldsymbol{x}_i, W^T W = I$$

$W = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m] \in \Re^{\boldsymbol{d} \times m}$, 约束$W^T W = I$保证坐标系标准正交坐标系。

PCA化简

$$f(W)=\sum_{i=1}^{n}\|\boldsymbol{y}_i-\bar{\boldsymbol{y}}\|^2=Tr(\sum_{i=1}^{n}(\boldsymbol{y}_i-\bar{\boldsymbol{y}})^T(\boldsymbol{y}_i-\bar{\boldsymbol{y}}))=Tr(\sum_{i=1}^{n}(\boldsymbol{y}_i-\bar{\boldsymbol{y}})(\boldsymbol{y}_i-\bar{\boldsymbol{y}})^T)$$

$$=Tr\left(\sum_{i=1}^{n}(\boldsymbol{y}_i-\bar{\boldsymbol{y}})(\boldsymbol{y}_i-\bar{\boldsymbol{y}})^T\right)=Tr(\sum_{i=1}^{n}(W^T\boldsymbol{x}_i-W^T\bar{\boldsymbol{x}})\left(W^T\boldsymbol{x}_i-W^T\bar{\boldsymbol{x}}\right)^T)$$

$$=Tr\left(\sum_{i=1}^{n}W^T(\boldsymbol{x}_i-\bar{\boldsymbol{x}})(\boldsymbol{x}_i-\bar{\boldsymbol{x}})^TW\right)=Tr\left(W^T(\sum_{i=1}^{n}(\boldsymbol{x}_i-\bar{\boldsymbol{x}})(\boldsymbol{x}_i-\bar{\boldsymbol{x}})^T)W\right)$$

$$=Tr\left(W^T(X-\bar{X})(X-\bar{X})^TW\right)=Tr(W^T(n-1)COV(X)W)$$

$COV(X)$为样本X的协方差矩阵(Covariance Matrix)，协方差矩阵本质就是方差的推广。

PCA化简

$$\mathrm{tr}\{\sum_{i=1}^{n}(\boldsymbol{x}_i-\bar{\boldsymbol{x}})(\boldsymbol{x}_i-\bar{\boldsymbol{x}})^T\}=tr\{(X-\bar{X})(X-\bar{X})^T\}$$

$$\boldsymbol{x}_i=[x_{i(1)},\cdots,x_{i(j)},\cdots,x_{i(d)}]^T\in\Re^{d\times 1},$$
$$\bar{\boldsymbol{x}}=[\bar{x}_{(1)},\cdots,\bar{x}_{(j)},\cdots,\bar{x}_{(d)}]^T\in\Re^{d\times 1}$$
$$X=[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_i,\cdots,\boldsymbol{x}_n]\ \in\Re^{d\times n},$$
$$\bar{X}=[\bar{\boldsymbol{x}},\cdots,\bar{\boldsymbol{x}},\cdots,\bar{\boldsymbol{x}}]\ \in\Re^{d\times n},$$
$$X-\bar{X}=[\boldsymbol{x}_1-\bar{\boldsymbol{x}},\cdots,\boldsymbol{x}_i-\bar{\boldsymbol{x}},\cdots,\boldsymbol{x}_n-\bar{\boldsymbol{x}}]\ \in\Re^{d\times n}$$

PCA化简——讨论左式

$$(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T =$$

$$\begin{bmatrix} (x_{i(1)} - \bar{x}_{(1)})^2 & \cdots & (x_{i(1)} - \bar{x}_{(1)})(x_{i(d)} - \bar{x}_{(d)})^T \\ \vdots & \ddots & \vdots \\ (x_{i(d)} - \bar{x}_{(d)})(x_{i(1)} - \bar{x}_{(1)})^T & \cdots & (x_{i(d)} - \bar{x}_{(d)})^2 \end{bmatrix}$$

$$\sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T =$$

$$\begin{bmatrix} \sum_i (x_{i(1)} - \bar{x}_{(1)})^2 & \cdots & \sum_i (x_{i(1)} - \bar{x}_{(1)})(x_{i(d)} - \bar{x}_{(d)})^T \\ \vdots & \ddots & \vdots \\ \sum_i (x_{i(d)} - \bar{x}_{(d)})(x_{i(1)} - \bar{x}_{(1)})^T & \cdots & \sum_i (x_{i(d)} - \bar{x}_{(d)})^2 \end{bmatrix}$$

$$\left(\sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \right)_{jt} = \sum_i (x_{i(j)} - \bar{x}_{(j)})(x_{i(t)} - \bar{x}_{(t)})^T$$

PCA化简——讨论右式

$$令A=(X-\bar{X}),则AA^T=(X-\bar{X})(X-\bar{X})^T$$

$$A_{j.}=[x_{1(j)}-\bar{x}_{(j)},\cdots,x_{i(j)}-\bar{x}_{(j)}\cdots,x_{n(j)}-\bar{x}_{(j)}]$$

$$A_{.t}^T=(A_{t.})^T=[x_{1(t)}-\bar{x}_{(t)},\cdots,x_{i(t)}-\bar{x}_{(t)}\cdots,x_{n(t)}-\bar{x}_{(t)}]^T$$

$$AA^T{}_{jt}=A_{j.}A_{.t}^T=\sum_i(x_{i(j)}-\bar{x}_{(j)})(x_{i(t)}-\bar{x}_{(t)})^T$$

$$=\left(\sum_i(\boldsymbol{x}_i-\bar{\boldsymbol{x}})(\boldsymbol{x}_i-\bar{\boldsymbol{x}})^T\right)_{jt}$$

$$故可证明\sum_{i=1}^{n}(\boldsymbol{x}_i-\bar{\boldsymbol{x}})(\boldsymbol{x}_i-\bar{\boldsymbol{x}})^T=(X-\bar{X})(X-\bar{X})^T$$

$$COV(X)_{jt}=\mathrm{cov}(x_{(j)},x_{(t)})=E\left(\left(x_{(j)}-E(x_{(j)})\right)\left(x_{(t)}-E(x_{(t)})\right)\right)$$

$$=\frac{1}{n-1}\sum_i(x_{i(j)}-\bar{x}_{(j)})(x_{i(t)}-\bar{x}_{(t)})^T$$

PCA问题

- 带约束优化问题

$$\max_{W} tr(W^T(X-\bar{X})(X-\bar{X})^T W), \text{ s.t. } W^T W = I$$

- 利用拉格朗日乘子法转化为对偶问题

$$\min_{W} \mathcal{L}(W,\lambda) := -tr(W^T(X-\bar{X})(X-\bar{X})^T W) + \lambda tr(W^T W - I), \quad \text{s.t. } \lambda \geq 0$$

- 问题求解：$\frac{\partial \mathcal{L}}{\partial W} = 0$

$$(X-\bar{X})(X-\bar{X})^T W = \lambda W$$

特征值分解

- 典型特征值分解(Eigendecomposition)问题

$$CW = \lambda W,\ C = (X - \bar{X})(X - \bar{X})^T$$

对$(X - \bar{X})(X - \bar{X})^T$进行特征值分解：

$$(X - \bar{X})(X - \bar{X})^T = V\Lambda V^T = \sum_i \lambda_i v_i v_i^T$$

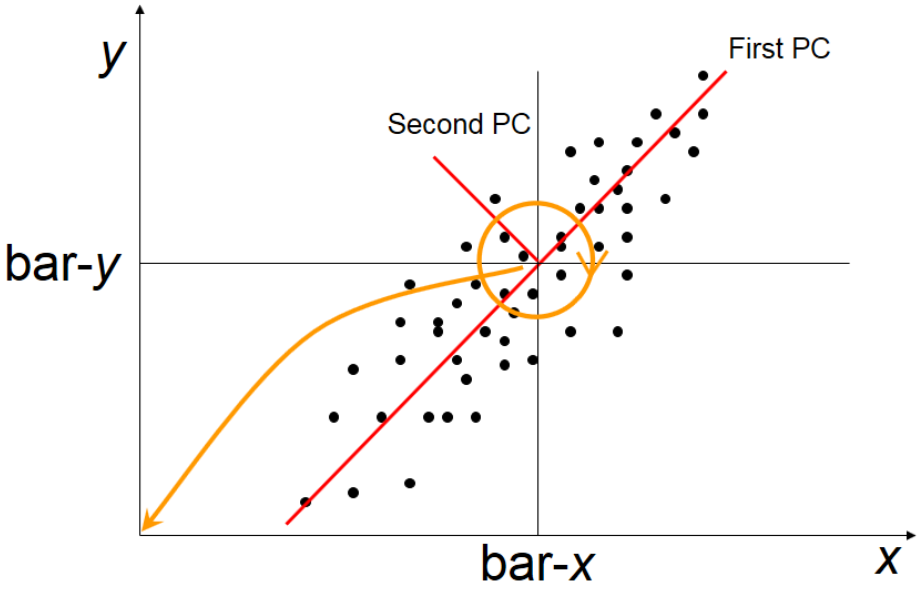
$$\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_d),\ V = [v_1, v_2, \cdots, v_d]$$

可得$(X - \bar{X})(X - \bar{X})^T$矩阵的$d$个特征根，对其进行排序$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$，并选取**前$m$个最大特征值**所对应的特征向量即组成了最优的投影矩阵$W = [v_1, \cdots, v_m]$。

- 数据降维：给定新样本矩阵$\tilde{X}$，可求其降维后的数据表征$\tilde{Y} = W^T\tilde{X}$。

符号释义

- $W = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m]$投影矩阵，新数据表征的坐标系集合，我们可以说$W$在样本空间中张(Span)了一个子空间(Subspace)。
- 向量$\boldsymbol{w}_i$定义一个样本的主成分(Principal Component, PC), 也定义新坐标系中第i个坐标轴，数学上称为W张成的子空间的一个基(basis)。



- $\boldsymbol{y}_i = W^T \boldsymbol{x}_i$是样本$\boldsymbol{x}_i$在$W$张成的子空间的一个低维嵌入(embedding).

PCA算法

- 中心化：

$$A = X - \bar{X},\ \ \bar{X} = \frac{1}{n}\mathbf{1}\mathbf{1}^{T}X = [\bar{\boldsymbol{x}}, \cdots, \bar{\boldsymbol{x}}, \cdots, \bar{\boldsymbol{x}}] \ \in \Re^{d \times n},$$

- 计算协方差矩阵

$$C = AA^{T} = (X - \bar{X})(X - \bar{X})^{T}$$

- 特征值分解

$$C = V\Lambda V^{T}$$

- 根据特征值选取特征向量构建投影矩阵

$$W = [v_1, \cdots, v_m]$$

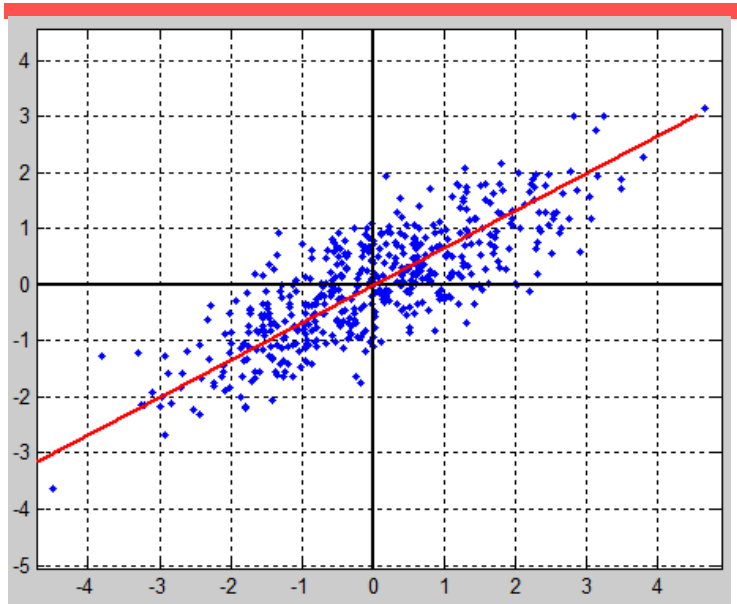
- 输入原样本投影至新子空间完成降维

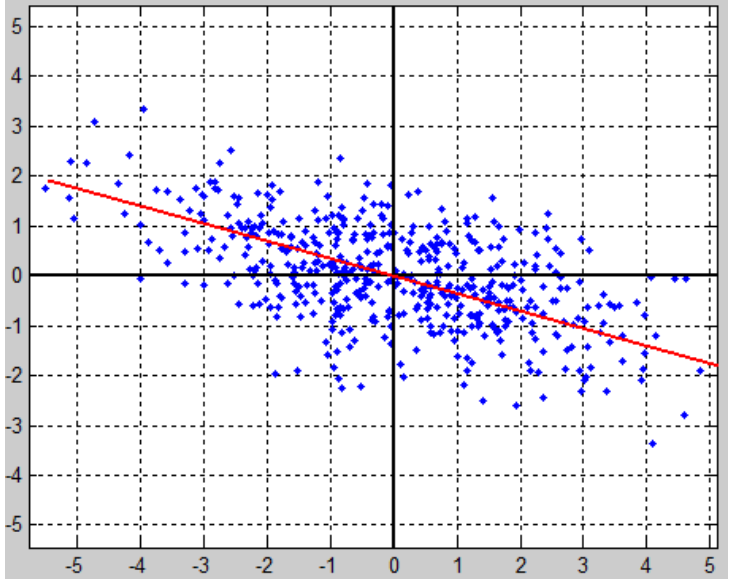
$$\tilde{Y} = W^{T}\tilde{X}$$

PCA算法实现（in matlab）

```
% 生成500个服从高斯分布2维样本，高斯分布的为参数[均值, 协方差矩阵]
X = mvnrnd([3, 3],[2 1.15; 1.15 1], 500);%每行为一个样本,nxd的矩阵
sample_num=size(X,1);%样本数目
Xbar=repmat(mean(X)',1,sample_num);%计算机dxn维度的均值矩阵
X=X';%dxn样本矩阵保证每个列向量为样本
Cov=(X-Xbar)*(X-Xbar)'/(sample_num-1);%计算协方差矩阵
[vs,lam]=eigs(Cov);%利用特征值分解求解 矩阵V
[lambda,index]=sort(diag(lam),'descend');%对特征根进行降序排序
presdim=1;%想保留的维度
vs=vs(:,index);%根据特征根大小对特征向量排序
W=vs(:,1:presdim);%选择前m个最大特征根对应的特征向量组成投影向量W
X=X-Xbar;%中心化
y=W'*X;
```

实验效果





```
>> Cov

Cov =

    1.9427    1.1168
    1.1168    1.0075

vs =

   -0.8325    0.5540
   -0.5540   -0.8325

lam =

    2.6859         0
         0    0.2643

>> w'

ans =

   -0.8325   -0.5540

Cov =

    3.6486   -1.0518
   -1.0518    1.0138

vs =

   -0.9438   -0.3305
    0.3305   -0.9438

lam =

    4.0169         0
         0    0.6454

ans =

   -0.9438    0.3305
```

PCA算法实现（in matlab）简洁版

```
% Data Generation
  X = mvnrnd([5, 5],[1 1.5; 1.5 3], 100);

% Step 1: Centralization
  Xhat = X - ones(n,1)*mean(X)/n;

% Step 2: Covariance matrix
  Cov = cov(Xhat);

% Step 3: Eigenvalue Decomposition
  [PC, variances, explained] = pcacov(Cov);
```

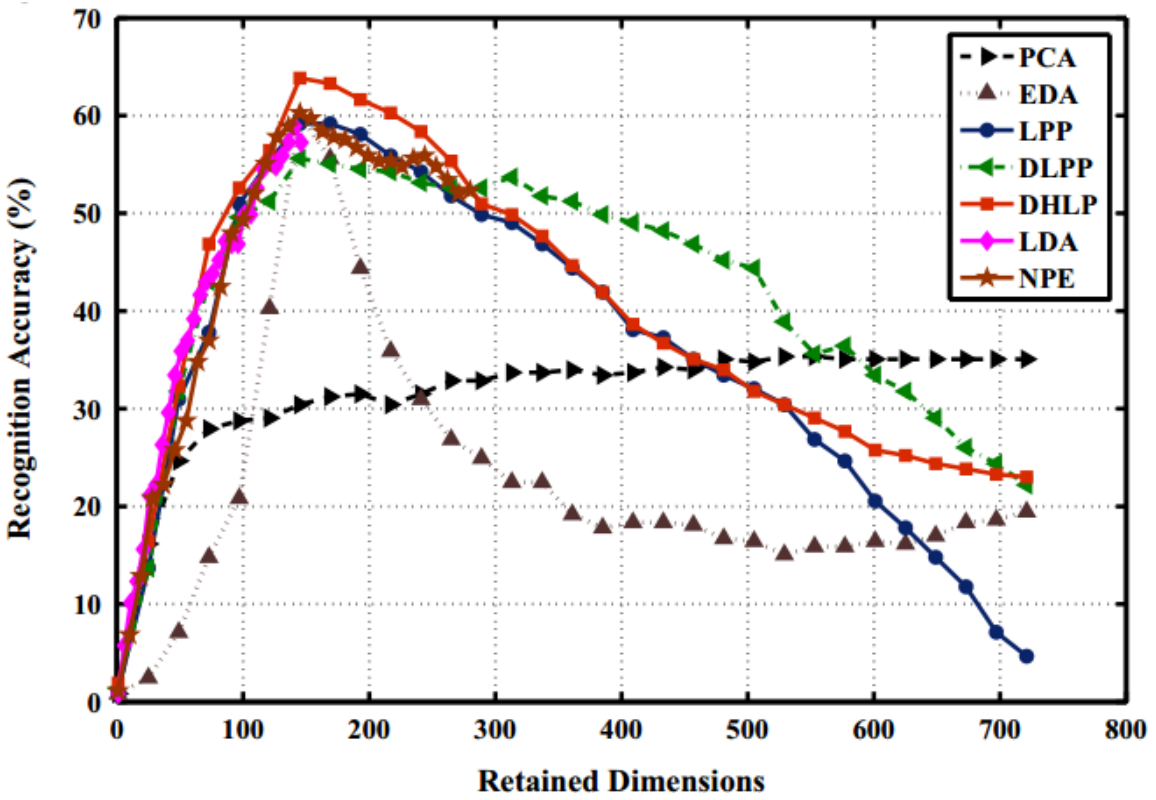
Retained Dimensionality

- 如何embedding——y的维度如何确定？
  1. 固定值（分类问题多设为类别数目-1）
  2. trial by error
  3. 特征值比重

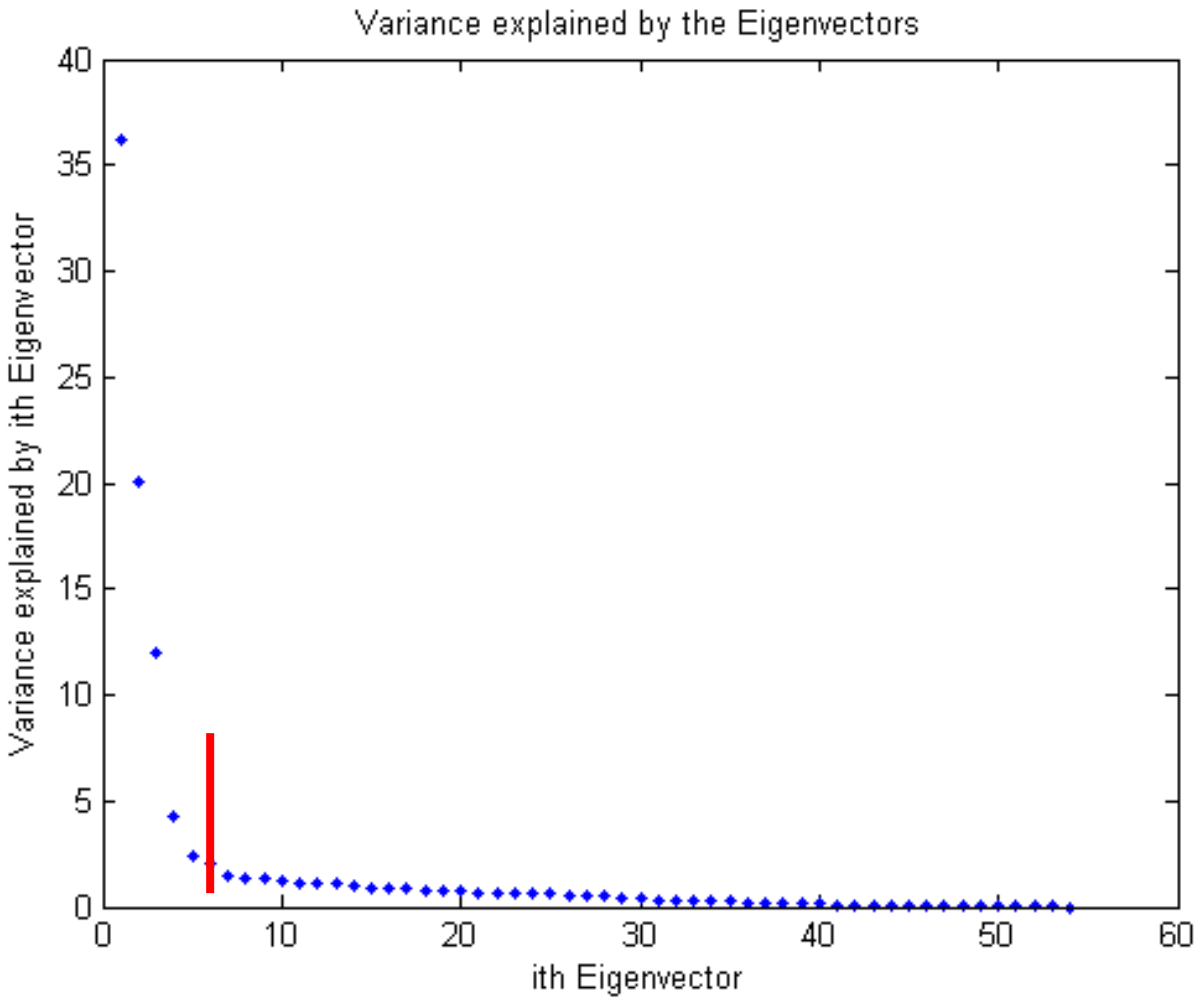
$$\max_m \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n} \lambda_i} < h$$

h为0-1之间的一个阈值，称为eigenvalue ratio.

  4. 根据拐点



Retained Dimensionality



Limitations of PCA

- Expensive Cost of Storage and Computation
  - The storage of Covariance Matrix is $O(d^2)$.
  - The computation complexity is $O(d^3)$.
  - Consider the $n \ll d$ case?
- Have not exploited the supervised information (i.e. Category label)
  - Preserving more information or preserving more discriminative information?

奇异值分解

- 解决第一个局限的利器——奇异值分解（Singular Value Decomposition，SVD）

$$\mathbf{A}_{[d \times n]} = \mathbf{U}_{[d \times r]} \mathbf{\Sigma}_{[r \times r]} (\mathbf{V}_{[n \times r]})^T$$

✓ **A**: Input data matrix——$d \times n$ matrix

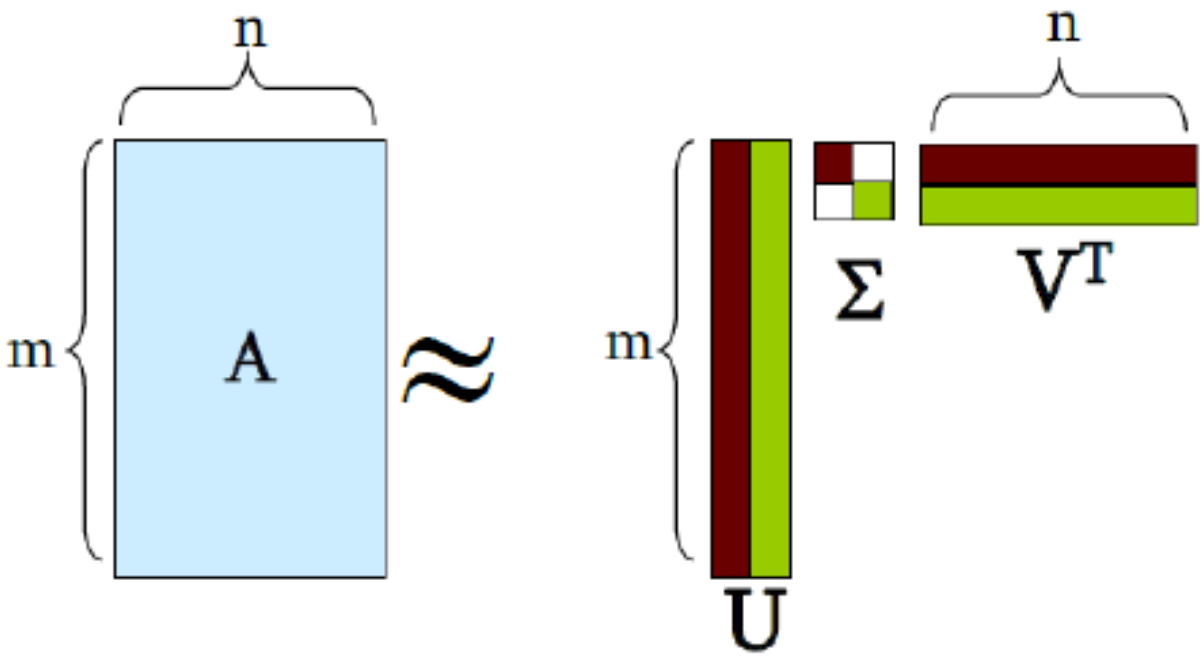
✓ **U**: Left singular vectors——$d \times r$ matrix：$\boldsymbol{U^T U = I}$

✓ **Σ** : Singular values——$r \times r$ diagonal matrix , r is the rank of the matrix.

✓ **V**: Right singular vectors——$n \times r$ matrix : $\boldsymbol{V^T V = I}$

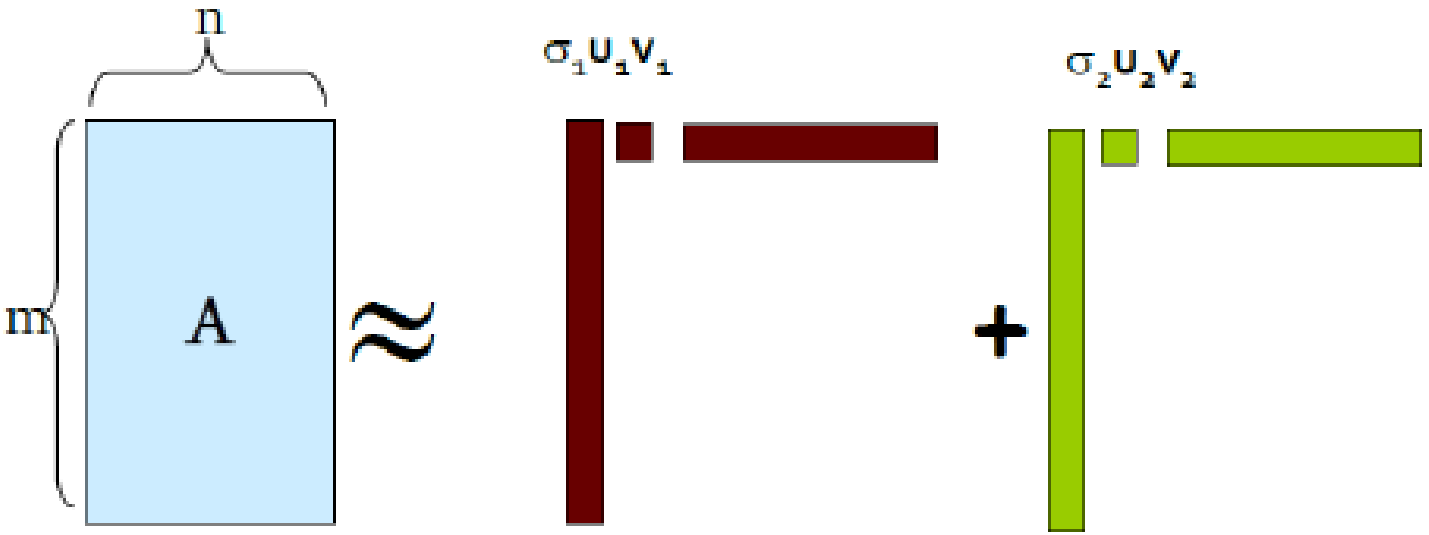
SVD - Definition

$$\mathbf{A} \approx \mathbf{U}\,\Sigma\,\mathbf{V}^{\mathsf{T}}$$



SVD - Definition

$$\mathbf{A} \approx \mathbf{U}\,\Sigma\,\mathbf{V}^{\mathsf{T}} = \sum_i \sigma_i u_i v_i^{\mathsf{T}}$$



PCA与SVD的解一致

- 令A= $(X - \bar{X})$; A=VΣU^T

$$C = AA^T = (X - \bar{X})(X - \bar{X})^T$$

$$= V\Sigma U^T (V\Sigma U^T)^T = V\Sigma U^T U\Sigma V^T$$

$$= V\Sigma\Sigma V^T = V\Lambda V^T$$

其中$\lambda_i = \sigma_i^2$

SVD vs PCA

Computational Complexity:

- SVD – O(nd^2) or O(dn^2) , even less
  - If we just want singular values
  - If we want first k singular vectors
  - If the matrix is sparse
- PCA – O(d^3)

Storage:

- SVD – O(nd) (e.g., original data matrix)
- PCA – O(d^2) (e.g., covariance matrix)

SVD and PCA

- Test on a random dataset [n,d] = [100,4000]
- Running time of PCA is 69.6601s
- Running time of SVD is 0.3414s

6.2 线性判别分析

黄晟

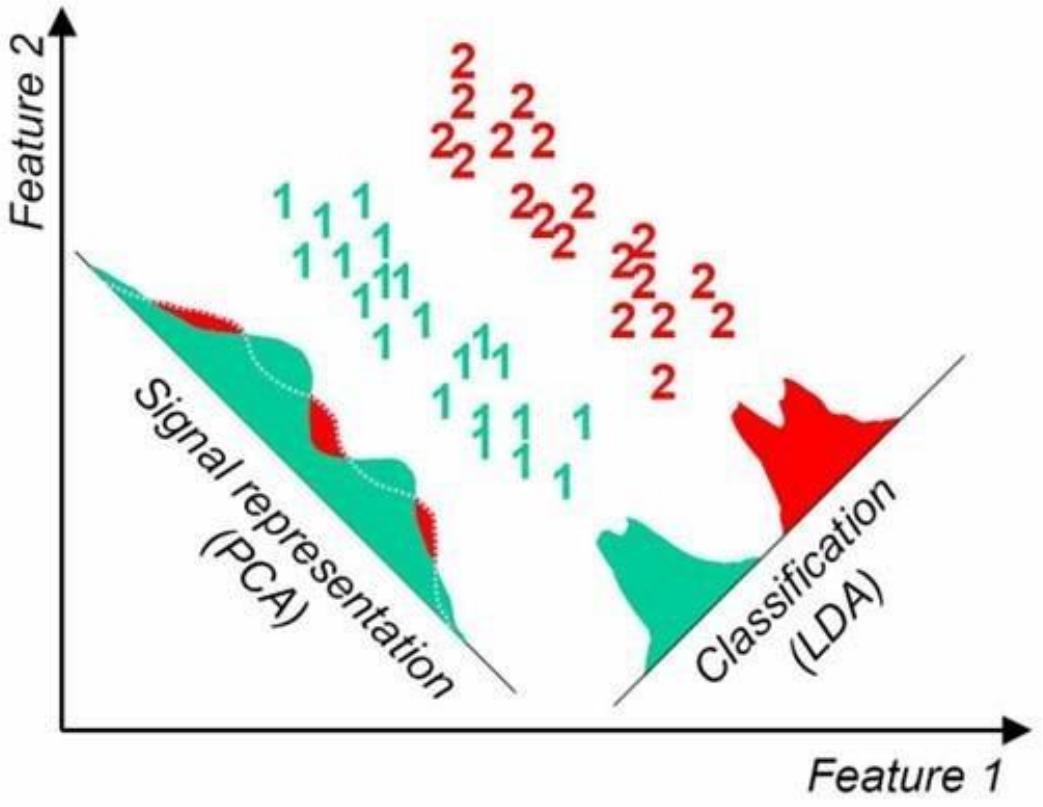
huangsheng@cqu.edu.cn

办公室：信息大楼B701

Linear Discriminant Analysis

- PCA在考虑数据降维时候希望在新空间的信息越多越好，即低维嵌入(embedding)的方差(协方差矩阵的迹)越大。
- PCA是一个**无监督降维方法**（Unsupervised DR），并没有运用任何有监督信息（Supervised Information），这也是PCA算法一个弊端。
- 对于一个分类问题，保留信息越多分类效果一定会更好么？
  - 不一定，分类效果完全取决样本在新的子空间下的可区分度。

线性判别分析



Fisher Criteria:

同类样本应该尽量聚合在一起，而不同类样本之间应该尽量扩散。

如何定义聚合程度和扩散程度？

方差(协方差矩阵)不是刚好就能够度量一组样本的扩散程度与聚合程度么！

$\mathrm{Var}(\boldsymbol{X})\uparrow, \mathrm{Scatter}(\boldsymbol{X})\uparrow, \mathrm{Compact}(\boldsymbol{X})\downarrow$

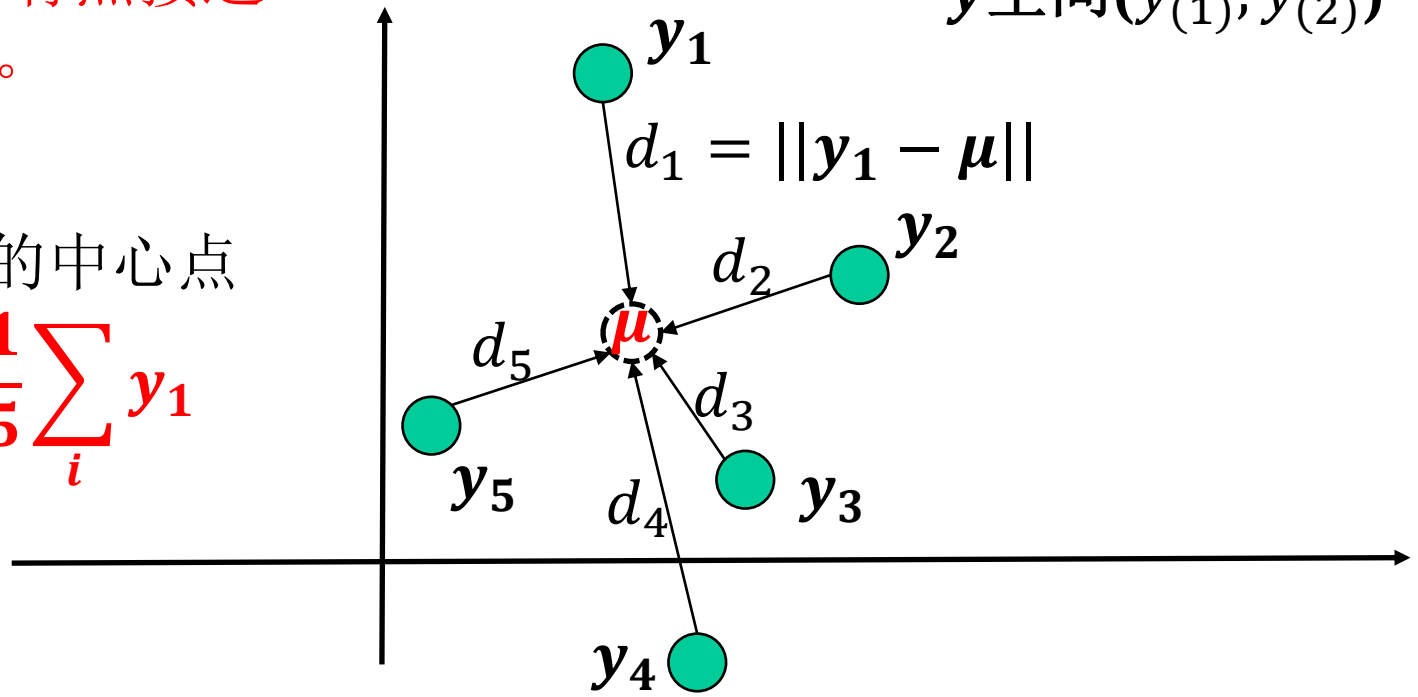
由于线性判别分析用到**fisher criteria**,所以又称费舍儿判别分析(**Fisher Discriminant Analysis, FDA**)!

如何保证样本点尽量聚合？

聚合:所有点接近中心点。

所有$\boldsymbol{y}$的中心点

$$\boldsymbol{\mu} = \frac{\mathbf{1}}{\mathbf{5}}\sum_{\boldsymbol{i}} \boldsymbol{y_1}$$



5个点凝聚度越高，则显然所有点到中心点的距离越近，

即$\sum_{i=1}^{5} d_i = \sum_{i=1}^{5} ||\boldsymbol{y_i} - \boldsymbol{\mu}|| \downarrow$。

$\sum_{i=1}^{5} ||\boldsymbol{y_i} - \boldsymbol{\mu}|| \downarrow$ 等价于 $\sum_{i=1}^{5} ||\boldsymbol{y_i} - \boldsymbol{\mu}||^2 \downarrow$ 等价于 $4 \times \mathrm{tr}(Cov(Y)) \downarrow$

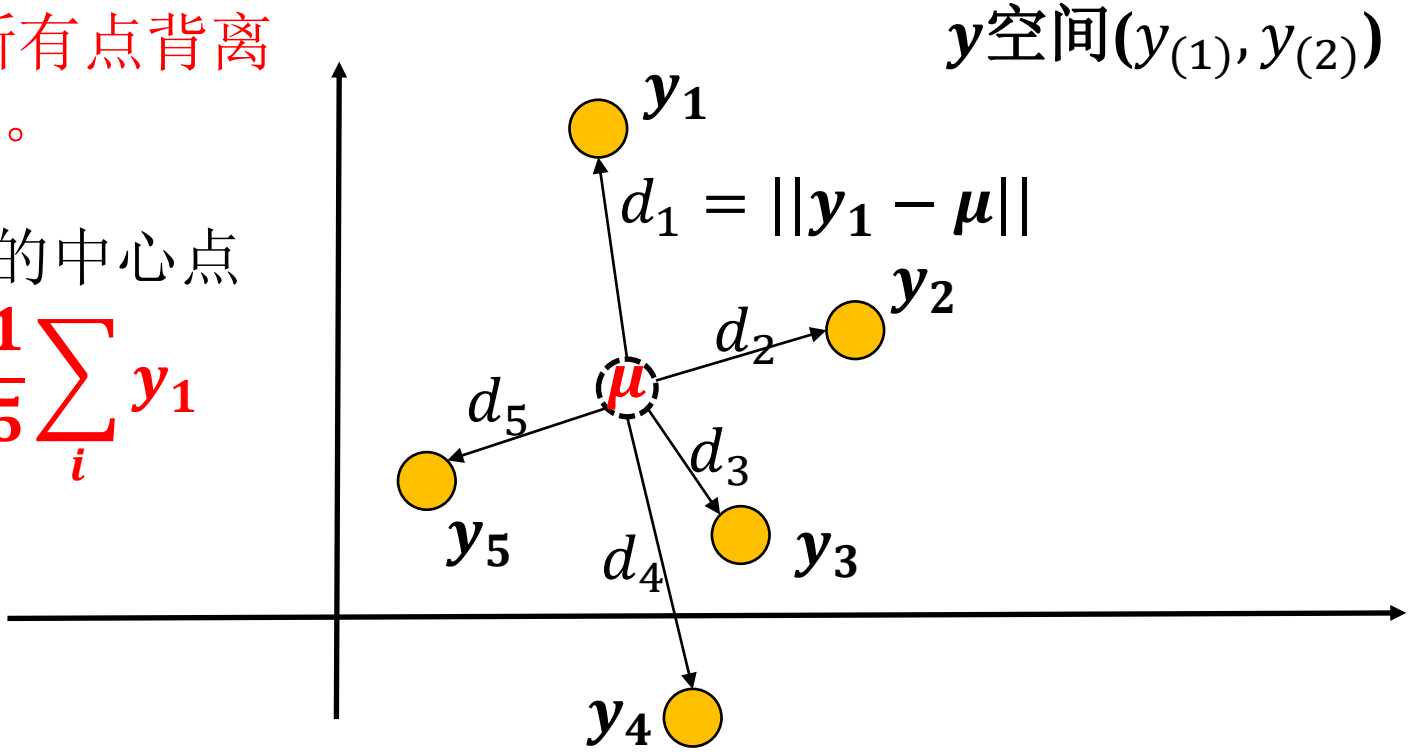
回顾PCA公式中的推导，$\mathrm{tr}\big(Cov(Y)\big) = \frac{1}{4}\sum_{i=1}^{5} ||\boldsymbol{y_i} - \boldsymbol{\mu}||^2$

如何保证类别尽量扩散？

扩散:所有点背离中心点。

所有$\boldsymbol{y}$的中心点

$$\boldsymbol{\mu} = \frac{1}{5}\sum_{i} \boldsymbol{y_1}$$



5个点扩散程度越高，则显然所有点到中心点距离越远，

即$\sum_{i=1}^{5} d_i = \sum_{i=1}^{5} ||\boldsymbol{y_i} - \boldsymbol{\mu}||$ ↑。

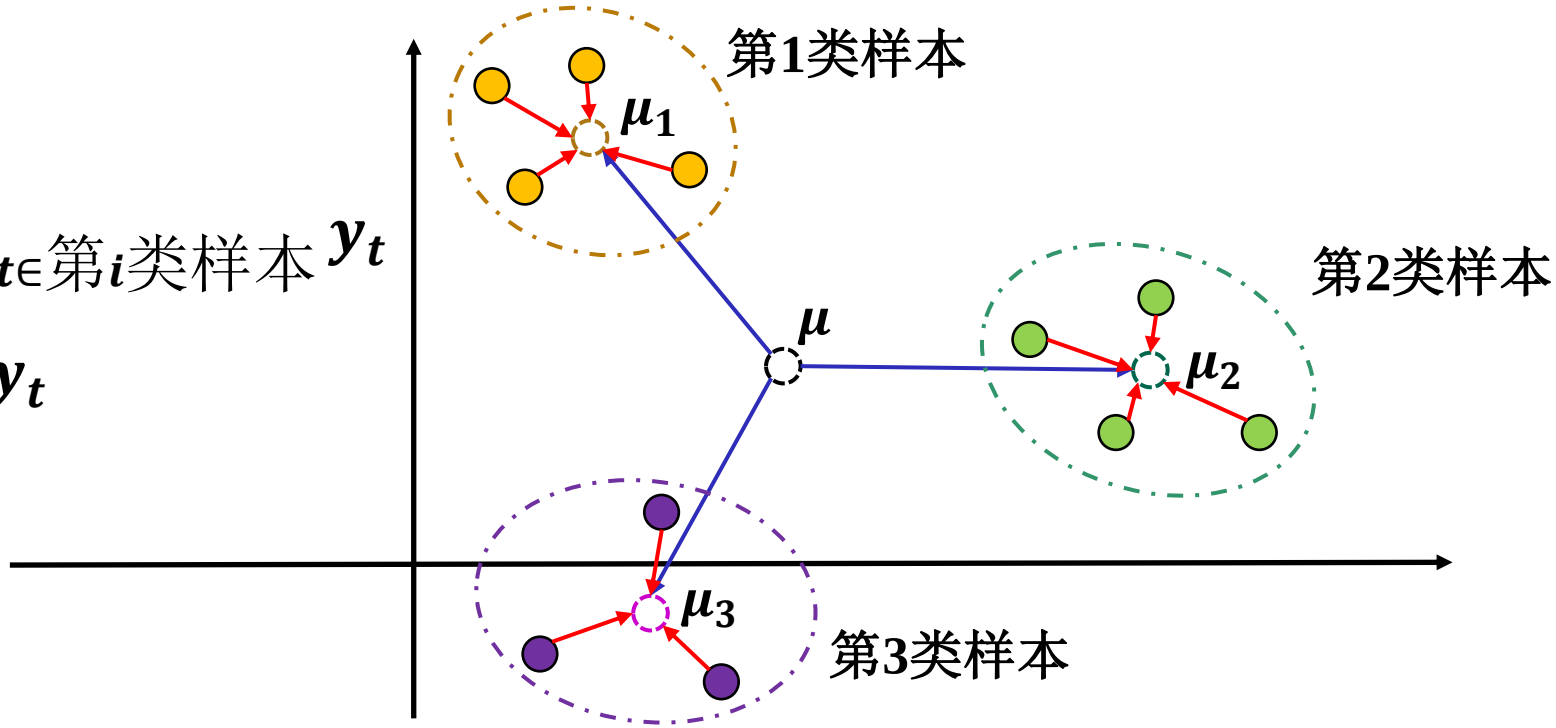
$\sum_{i=1}^{5} ||\boldsymbol{y_i} - \boldsymbol{\mu}||$ ↑ 等价于 $\sum_{i=1}^{5} ||\boldsymbol{y_i} - \boldsymbol{\mu}||^2$ ↑ 等价于$4 \times \mathrm{tr}(Cov(Y))$ ↑

如何便于分类？

- Fisher Criteria:
  - 同类样本应具有较好的聚合度，而类别之间应该具有较好的扩散度。

$$\boldsymbol{\mu_i} = \frac{1}{n_i}\sum_{t\in\text{第}i\text{类样本}} \boldsymbol{y_t}$$

$$\boldsymbol{\mu} = \frac{1}{n}\sum_t \boldsymbol{y_t}$$



同类样本扩散度尽量低(所有红色线段尽量短)，不同类扩散度尽量高(所有蓝色线段尽量长)。

目标函数

- 度量第$c \in [1, \cdots, c, \cdots, C]$类样本的扩散度程度（聚合程度）：

一维的情况（方差）：

$$s_w(c) = \sum_{i \in \text{class } c} (y_i - \mu_c)^2 = (n_c - 1) \times var(y_{i \in \text{class } c}),$$

$$\mu_c = \frac{1}{n_c} \sum_{i \in \text{class } c} y_i$$

多维推广(协方差矩阵——本质上是高维样本的方差)：

$$s_w(c) = tr\{ \sum_{i \in \text{class } c} (\boldsymbol{y}_i - \boldsymbol{\mu}_c)(\boldsymbol{y}_i - \boldsymbol{\mu}_c)^T \} = (n_c - 1) \times tr\{COV(Y_c)\}$$

$$\boldsymbol{\mu}_c = \frac{1}{n_c} \sum_{i \in \text{class } c} \boldsymbol{y}_i$$

目标函数

- 度量类别之间的扩散程度：

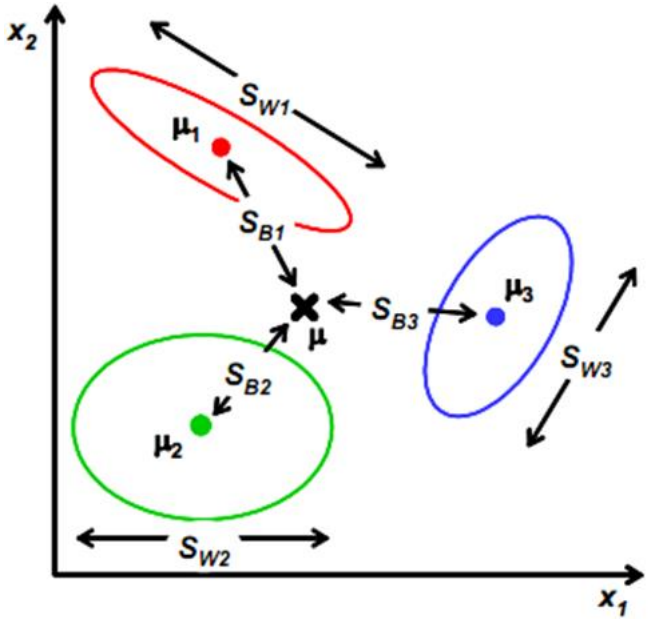
不同类权重不同

$$s_b = \sum_{c=1}^{C} n_c tr\{(\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T\}, \boldsymbol{\mu} = \frac{1}{n}\sum_i \boldsymbol{y}_i$$

- 根据Fisher Criteria， $\boldsymbol{y}$在一个理想的坐标系下，应该具有如下特性，
  - $\forall c,\ s_w(c)\downarrow$ 且 $s_b\uparrow$

故满足Fisher准则的目标函数可构建如下：

$$\min \frac{\sum_c s_w(c)}{s_b} \text{ 或} \max \frac{s_b}{\sum_c s_w(c)}$$



目标函数

- 假设低维嵌入$\boldsymbol{y}$是样本$\boldsymbol{x}$通过投影矩阵W投影到LDA子空间的坐标，即
  - $\boldsymbol{y} = W^T\boldsymbol{x}, \boldsymbol{y} \in \Re^{m\times 1}, \boldsymbol{x} \in \Re^{d\times 1}, W \in \Re^{d\times m}$
- 把$\boldsymbol{y} = W^T\boldsymbol{x}$代入目标函数

$$\min_W \frac{\sum_c s_w(c)}{s_b} := \frac{tr\{\sum_{c=1}^{C}\sum_{i\in \text{class } c}(\boldsymbol{y}_i - \boldsymbol{\mu}_c)(\boldsymbol{y}_i - \boldsymbol{\mu}_c)^T\}}{tr\{\sum_{c=1}^{C} n_c(\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T\}}$$

$$= \frac{tr\{W^T\sum_{c=1}^{C}\sum_{i\in \text{class } c}(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_c)(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_c)^T W\}}{tr\{W^T\sum_{c=1}^{C} n_c(\overline{\boldsymbol{x}}_c - \overline{\boldsymbol{x}})(\overline{\boldsymbol{x}}_c - \overline{\boldsymbol{x}})^T W\}}$$

- 注意——原来中心点在新坐标系下仍然是中心点
  - $\boldsymbol{\mu_c} = \frac{1}{n_c}\sum_{i\in \text{class } c}\boldsymbol{y_i} = \frac{1}{n_c}\sum_{i\in \text{class } c} W^T\boldsymbol{x_i} =$

$$W^T\left(\frac{1}{n_c}\sum_{i\in \text{class } c}\boldsymbol{x_i}\right) = W^T\overline{\boldsymbol{x}}_c \in \Re^{m\times 1}$$

目标函数

- 令$S_w = \sum_{c=1}^{C} \sum_{i \in \text{class } c} (\boldsymbol{x}_i - \overline{\boldsymbol{x}}_c)(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_c)^T \in \Re^{d \times d}$，
  $S_b = \sum_{c=1}^{C} n_c (\overline{\boldsymbol{x}}_c - \overline{\boldsymbol{x}})(\overline{\boldsymbol{x}}_c - \overline{\boldsymbol{x}})^T \in \Re^{d \times d}$
- 则目标函数可化简如下：

$$\min_W \frac{\sum_c s_w(c)}{s_b} := \frac{tr\{\sum_{c=1}^{C} \sum_{i \in \text{class } c} (\boldsymbol{y}_i - \boldsymbol{\mu}_c)(\boldsymbol{y}_i - \boldsymbol{\mu}_c)^T\}}{tr\{\sum_{c=1}^{C} n_c (\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T\}} =$$

$$= \frac{tr(W^T S_w W)}{tr(W^T S_b W)} = tr\left(\frac{W^T S_w W}{W^T S_b W}\right)$$

S_w被称为类内散度矩阵(within-class scatter matrix)

S_b被称为类间散度矩阵(between-class scatter matrix)

$S_t = S_w + S_b = \sum_{i=1}^{n} (\boldsymbol{x_i} - \overline{\boldsymbol{x}})(\boldsymbol{x_i} - \overline{\boldsymbol{x}})^T \in \Re^{d \times d}$被称为全局散度矩阵(total scatter matrix)。

目标函数

- 线性判别分析(LDA)目标函数:

$$\min_W tr\left(\frac{W^T S_w W}{W^T S_b W}\right) \text{或} \max_W tr\left(\frac{W^T S_b W}{W^T S_w W}\right)$$

- 该目标函数为广义瑞利商（generalized Rayleigh Quotient）, 最优解为广义特征值问题。此模型的最优解有无穷多个（对最优解w进行任意缩放）
- 问题等价于

$$\min_W tr\left(W^T S_w W\right), \text{s.t. } W^T S_b W = I$$

或

$$\min_W -tr\left(W^T S_b W\right), \text{s.t. } W^T S_w W = I$$

目标函数求解

- 对优化问题构建拉格朗日函数$\mathcal{L}$，直接求偏导赋零：$\frac{\delta \mathcal{L}}{\delta W} = 0$

$$\mathcal{L} = tr\{W^T S_w W + \lambda(I - W^T S_b W)\}$$

- 最后化简可得：

$$S_w W = \lambda S_b W \rightarrow S_b^{-1} S_w W = \lambda W$$

- 典型特征值分解问题，

W为$S_b^{-1} S_w$矩阵前m个**最小非零特征根**对于特征向量组成。

LDA算法步骤

- 计算各类样本的中心与所有样本的中心：

$$\bar{\boldsymbol{x}}_{\boldsymbol{c}} = \frac{1}{n_c}\sum_{i\in \text{class } c} \boldsymbol{x}_{\boldsymbol{i}}, \bar{\boldsymbol{x}} = \frac{1}{n}\sum \boldsymbol{x}_{\boldsymbol{i}}$$

- 计算类间与类内散度矩阵

$$S_w = \sum_{c=1}^{C}\sum_{i\in \text{class } c}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_c)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_c)^T \in \Re^{d\times d},$$

$$S_b = \sum_{c=1}^{C} n_c(\bar{\boldsymbol{x}}_c - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_c - \bar{\boldsymbol{x}})^T \in \Re^{d\times d}$$

- 特征值分解

$$S_b^{-1}S_w = V\Lambda V^T$$

- 根据特征值选取特征向量构建投影矩阵

$$W = [v_1, \cdots, v_m]$$

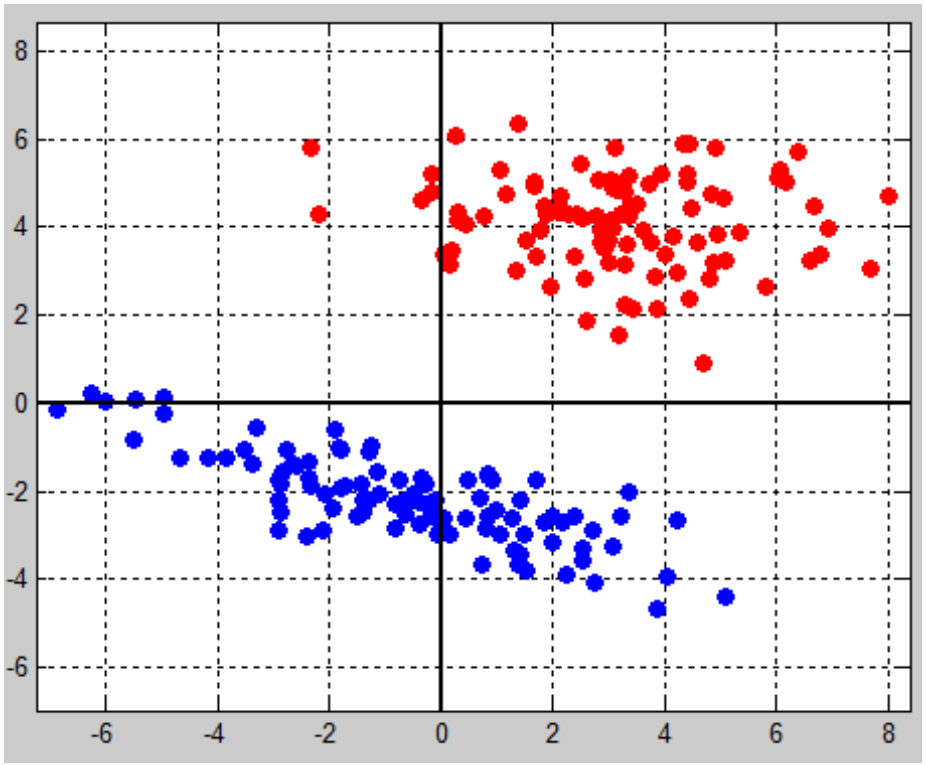
- 输入原样本投影至新子空间完成降维

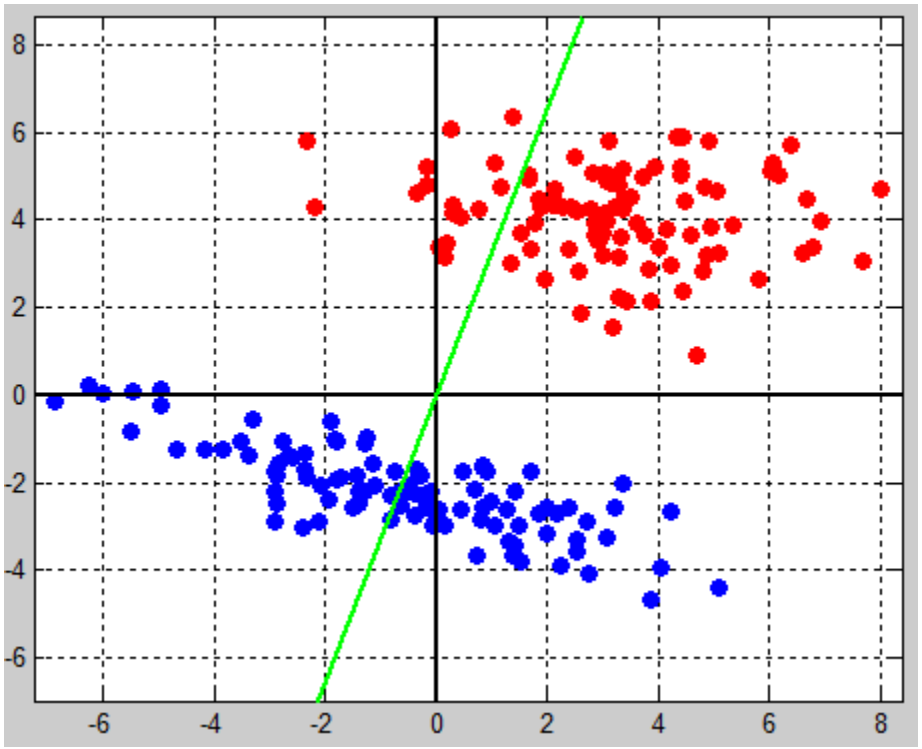
$$\tilde{Y} = W^T\tilde{X}$$

LDA算法实现（matlab版本）

```
X1= mvnrnd([3, 4],[5 -0.35; -0.35 1], 100);%生成服从高斯分布的正样本
X2 = mvnrnd([-1, -2],[6 -1.85; -1.85 1], 100); %生成服从高斯分布的负样本
X=[X1;X2];%偷懒，把正样本与负样本合并，便于后期调用均值函数求样本中心
X1=X1';
X2=X2';
x1_bar=mean(X1')';%正样本中心
x2_bar=mean(X2')';%负样本中心
x_bar=mean(X')';%所有样本的中心
n1=size(X1,2);%正样本数目
n2=size(X2,2);%负样本数目
n=n1+n2;%样本总数
sw_1=n1*(x1_bar-x_bar)*(x1_bar-x_bar)';%构建正样本散度矩阵
sw_2=n2*(x2_bar-x_bar)*(x2_bar-x_bar)';%构建负样本散度矩阵
Sw=sw_1+sw_2;%类内散度矩阵
MX1=X1-repmat(x1_bar,1,n1);%正样本中心化
MX2=X2-repmat(x2_bar,1,n2);
Sb=MX1*MX1'+MX2*MX2';%类间散度矩阵
[vs,lam]=eigs(Sw,Sb);%fisher criterion, minimize Sw/Sb, 特征值分解
lam=diag(lam);%去除小于等于零的特征值对应的特征向量，属于退化解
ind=find(lam<=0);
lam(ind)=[];
vs(:,ind)=[];
[lambda,index]=sort(lam,'ascend');%对特征值降序排序
presdim=1;
vs=vs(:,index);
w=vs(:,1:presdim);%LDA投影矩阵，LDA基矩阵
```

LDA算法结果





lambda =

0.0602

Inf

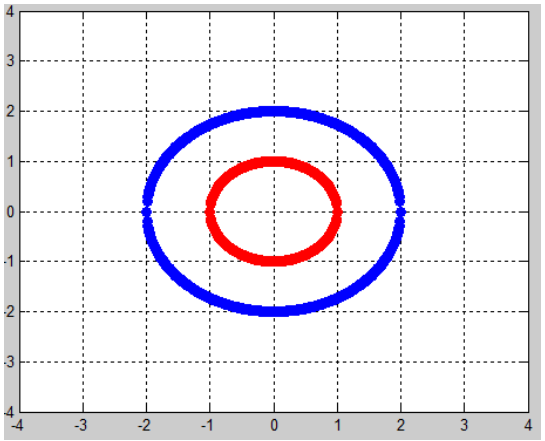
w =

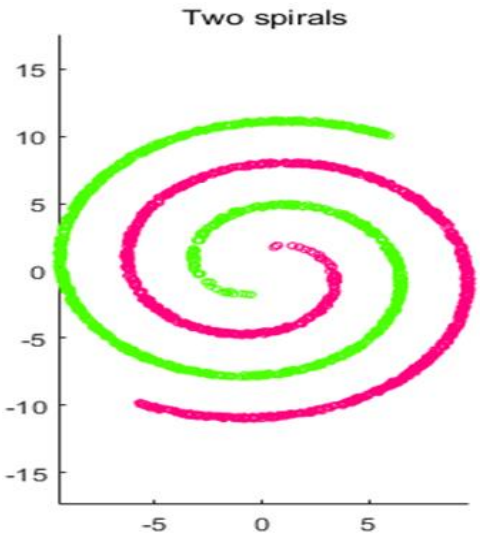
-0.3057

-1.0000

LDA Limitations

- 小样本问题（Small Sample Size, SSS）:每个类都只有一个样本的情况，$\boldsymbol{x}_i - \overline{\boldsymbol{x}}_c = \mathbf{0}, S_w = \mathbf{0}$
- **PCA算法类似**
  - The storage of Scatter Matrix is $O(d^2)$.
  - The computation complexity is $O(d^3)$.
  - Cannot handle the nonlinear case.



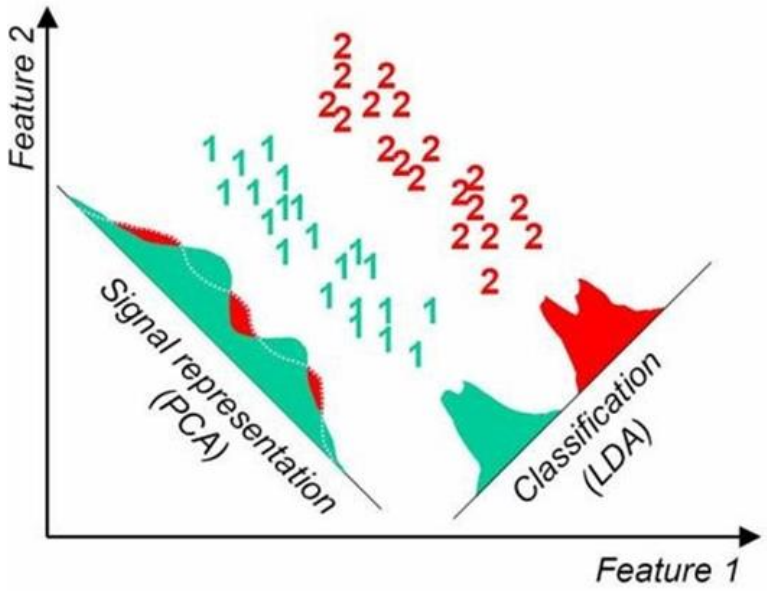


LDA Limitations

- S_b不满秩，没有矩阵逆，**LDA求解不稳定。**
- **解决方案？**
  - 正则化
  - **先PCA后LDA**

PCA与LDA对比-不同点

- 思想上：
  - PCA旨在寻找一组子坐标系（定义一个子空间）使得样本点的方差最大，即信息量保留越多。
  - LDA旨在寻找一组子坐标系（定义一个子空间）使得样本点类内散度越小，类间散度越大（Fisher Criteria）。
- 监督性：
  - PCA是无监督学习方法。
  - LDA是有监督学习方法。
- 算法效率
  - PCA效率更胜一筹。



PCA与LDA对比-相同点

- 子空间学习（Subspace Learning）角度：
  - PCA与LDA都属于线性子空间学习算法（Linear Subspace Learning）。目标都是学习一个投影矩阵$W = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m]$，使得样本在新坐标系上的表示具有相应特性（PCA——样本方差最大，LDA——同类样本高聚合度，不同类样本高扩散度）。在样本空间定义一个新的子坐标系（即子空间），其每个列向量定义一个坐标轴，故此类算法均称为子空间学习算法。
- 降维（Dimension Reduction）角度：
  - 坐标轴数目少，维度也少了。
- 特征提取（Feature Extraction）角度：
  - 样本在新坐标系下的坐标相当于样本的新特征（Feature，or Representation）。

6.3 子空间学习的核化

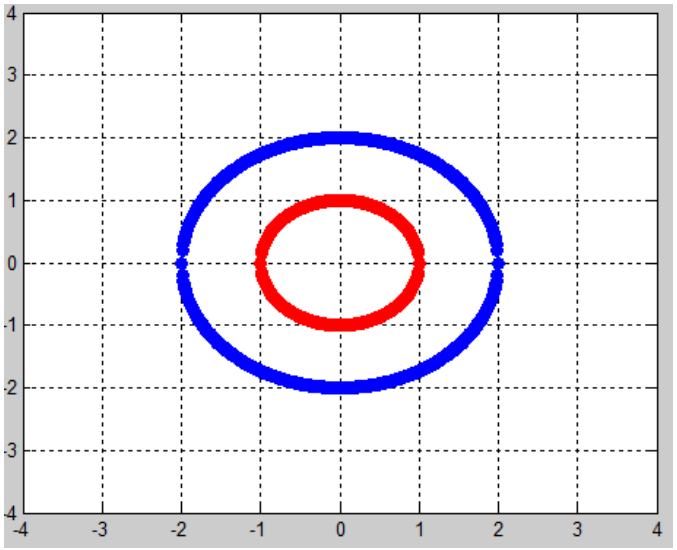
黄晟

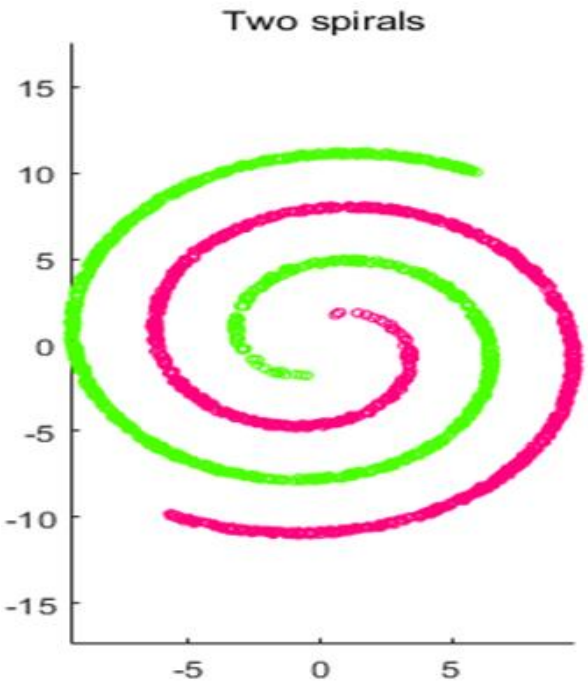
huangsheng@cqu.edu.cn

办公室：信息大楼B701

Kernelized Subspace learning

- 样本分布不是线性的。

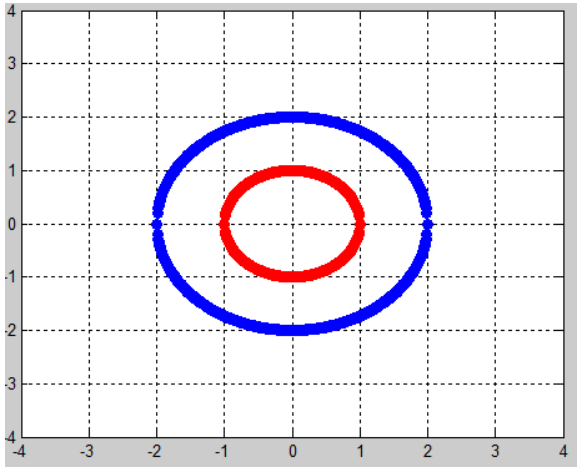


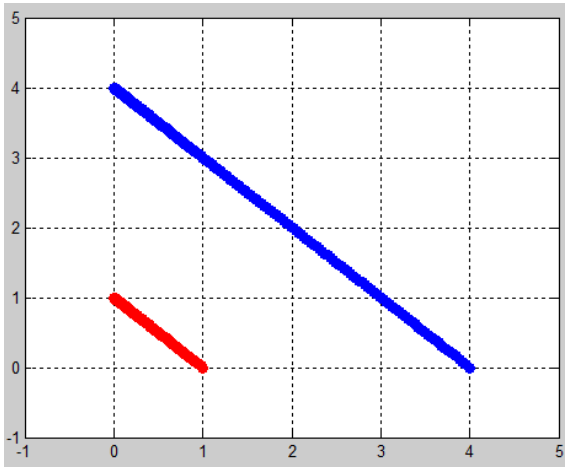


- 哪个方向是样本主成分，哪个方向具有最好的判别性？

非线性映射

- 思路：
  - 把样本利用非线性映射投射到更好维度，使得在高维非线性空间，可以使用线性子空间学习进行数据分析。





$(x_{(1)}, x_{(2)}) \longrightarrow (x_{(1)}^2, x_{(2)}^2, x_{(1)}x_{(2)}, x_{(1)}, x_{(2)})$

非线性映射$\varphi(\cdot)$

2维样本空间　　5维多项式空间

核技巧(Kernel trick)

- 然而，对于任意给定数据寻找合适的非线性映射φ是不现实的。
- 定义核函数（Kernel Function），回顾核化SVM:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_i)$$

- φ可以隐式地由核函数表示出来，核函数很好找只要满足Mercer 条件即是一个核函数，而且内积的计算也非常方便。

核函数

- 常见核函数（核矩阵对称半正定——Mercer条件）

名称	表达式
线性核	$K(x_i, x_j) = x_i^{\mathrm{T}} x_j$
多项式核	$K(x_i, x_j) = (x_i^{\mathrm{T}} x_j)^d$
高斯核	$K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$
拉普拉斯核	$K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|}{\sigma})$
Sigmoid 核	$K(x_i, x_j) = \tanh(\beta x_i^T x_j + \theta)$

核化子空间学习

- 外积的迹转为内积：$\mathrm{tr}(\boldsymbol{x}_i\boldsymbol{x}_i^T) = \boldsymbol{x}_i^T\boldsymbol{x}_i$
- PCA

$$f(W) = W^T\left(\sum_{i=1}^{n}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T(\boldsymbol{x}_i - \overline{\boldsymbol{x}})\right)W$$

$$K(\boldsymbol{x}_i - \overline{\boldsymbol{x}}, \boldsymbol{x}_i - \overline{\boldsymbol{x}})$$

- LDA

$$f(W) = \frac{W^T\sum_{c=1}^{C}\sum_{i\in\text{class }c}(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_c)^T(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_c)W}{W^T\sum_{c=1}^{C}n_c(\overline{\boldsymbol{x}}_c - \overline{\boldsymbol{x}})^T(\overline{\boldsymbol{x}}_c - \overline{\boldsymbol{x}})W}$$

$$K(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_c, \boldsymbol{x}_i - \overline{\boldsymbol{x}}_c)$$

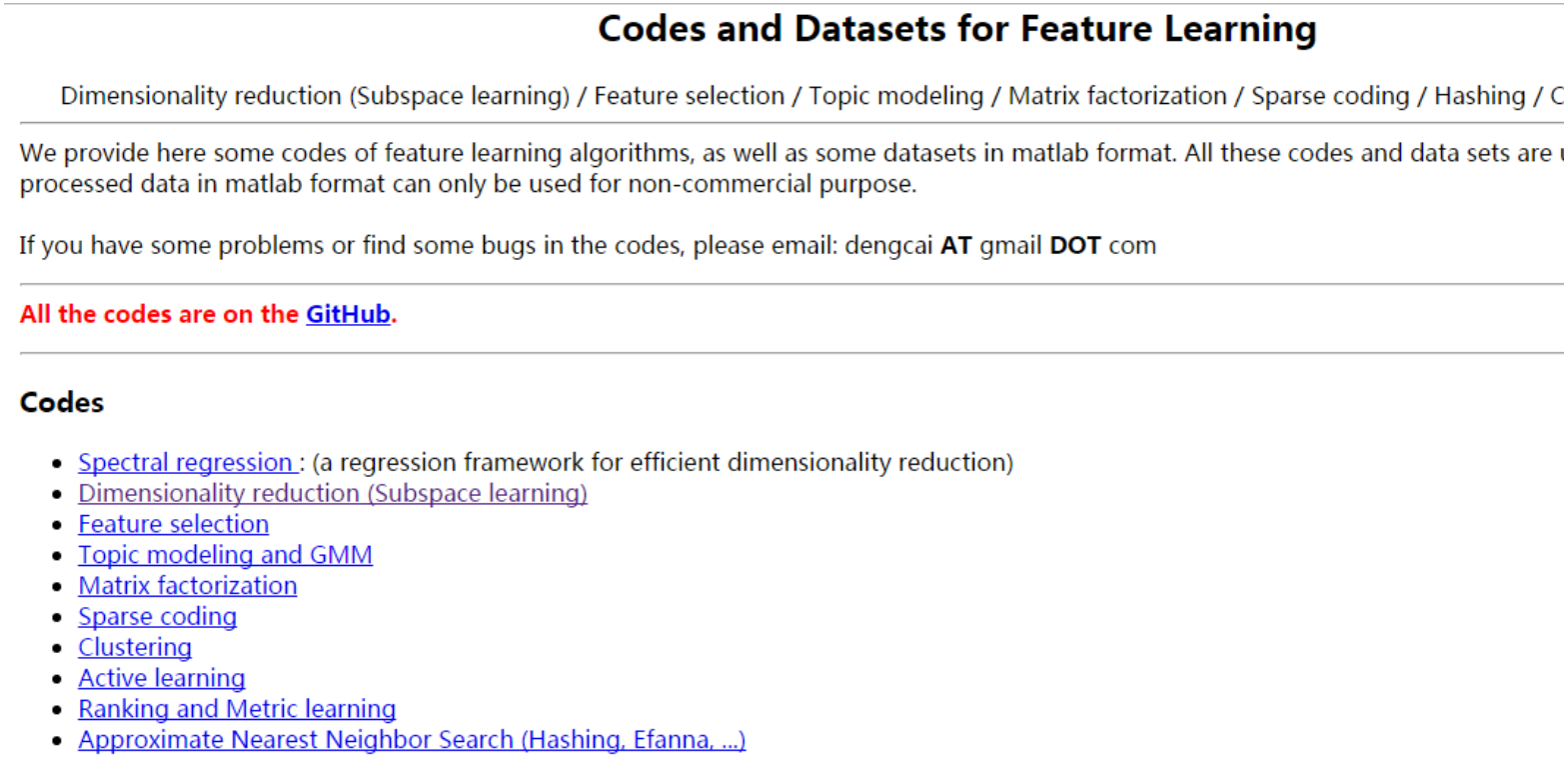
$$K(\overline{\boldsymbol{x}}_c - \overline{\boldsymbol{x}}, \overline{\boldsymbol{x}}_c - \overline{\boldsymbol{x}})$$

子空间学习总结

- PCA
- SVD
- LDA
- Kernelized Subspace Learning

子空间学习的资料

- 各类子空间学习matlab代码+数据库
  - 浙江大学蔡登教授主页
  - http://www.cad.zju.edu.cn/home/dengcai/Data/data.html



Codes and Datasets for Feature Learning

Dimensionality reduction (Subspace learning) / Feature selection / Topic modeling / Matrix factorization / Sparse coding / Hashing / C

We provide here some codes of feature learning algorithms, as well as some datasets in matlab format. All these codes and data sets are u
processed data in matlab format can only be used for non-commercial purpose.

If you have some problems or find some bugs in the codes, please email: dengcai **AT** gmail **DOT** com

All the codes are on the GitHub.

Codes

- Spectral regression : (a regression framework for efficient dimensionality reduction)
- Dimensionality reduction (Subspace learning)
- Feature selection
- Topic modeling and GMM
- Matrix factorization
- Sparse coding
- Clustering
- Active learning
- Ranking and Metric learning
- Approximate Nearest Neighbor Search (Hashing, Efanna, ...)

6.4 子空间学习在人脸识别中的应用

黄晟

huangsheng@cqu.edu.cn

办公室：信息大楼B701

Appearance-based face recognition

- 子空间学习曾是解决人脸识别问题的经典方法，基于子空间学习的人脸识别方法又称基于表观的方法（Appearance-based approach）。
  - PCA又称为Eigenface
  - LDA又称为Fisherface
- Methodology：

人脸图片(灰度图)➔PCA/LDA➔KNN➔标签

特征提取　　分类

K近邻算法

- K近邻算法（K-Nearest Neighbor, KNN）是非常经典一个分类算法，也是机器学习中最简单的方法之一。
- 核心思想：物以类聚，人以群分。
  - 如果一个样本在特征空间中的k个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别。
- KNN不需要训练，属于典型懒惰学习(Lazy Learning)方法。
- KNN是有监督学习算法，因为测试阶段使用了标签。

K近邻算法

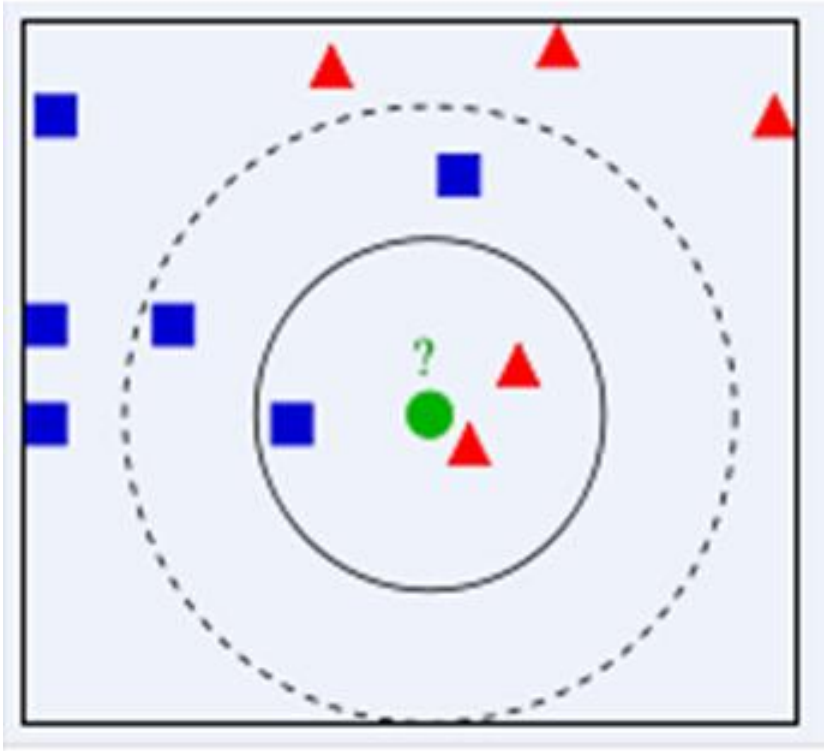
- KNN算法步骤
  - ① 算距离：给定测试对象，计算它与训练集中的每个对象的距离。
  - ② 找邻居：圈定距离最近的k个训练对象，作为测试对象的近邻。
  - ③ 做分类：根据这k个近邻归属的主要类别，来对测试对象分类。

K近邻算法

- 距离度量：
  - 明可夫斯基距离(Minkowski Distance):
    $D(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sqrt[p]{\sum_t (x_{i(t)} - x_{j(t)})^p}$
  - $p = 1$,曼哈顿距离（Manhattan Distance):
    $D(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sum_t |x_{i(t)} - x_{j(t)}|$
  - $p = 2$, 欧式距离（Euclidean Distance):
    $D(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sqrt{\sum_t (x_{i(t)} - x_{j(t)})^2}$
  - $p = \infty$,切比雪夫距离(Chebyshev Distance):
    $D(\boldsymbol{x_i}, \boldsymbol{x_j}) = \max |x_{i(t)} - x_{j(t)}|$
  - Cosine距离，马氏距离等。

K近邻算法

- K的取值?



- $1 \leq k \leq \sqrt{n}$， 选择策略 trial by error,
- 一般常用$k = 1$，即最近邻分类器

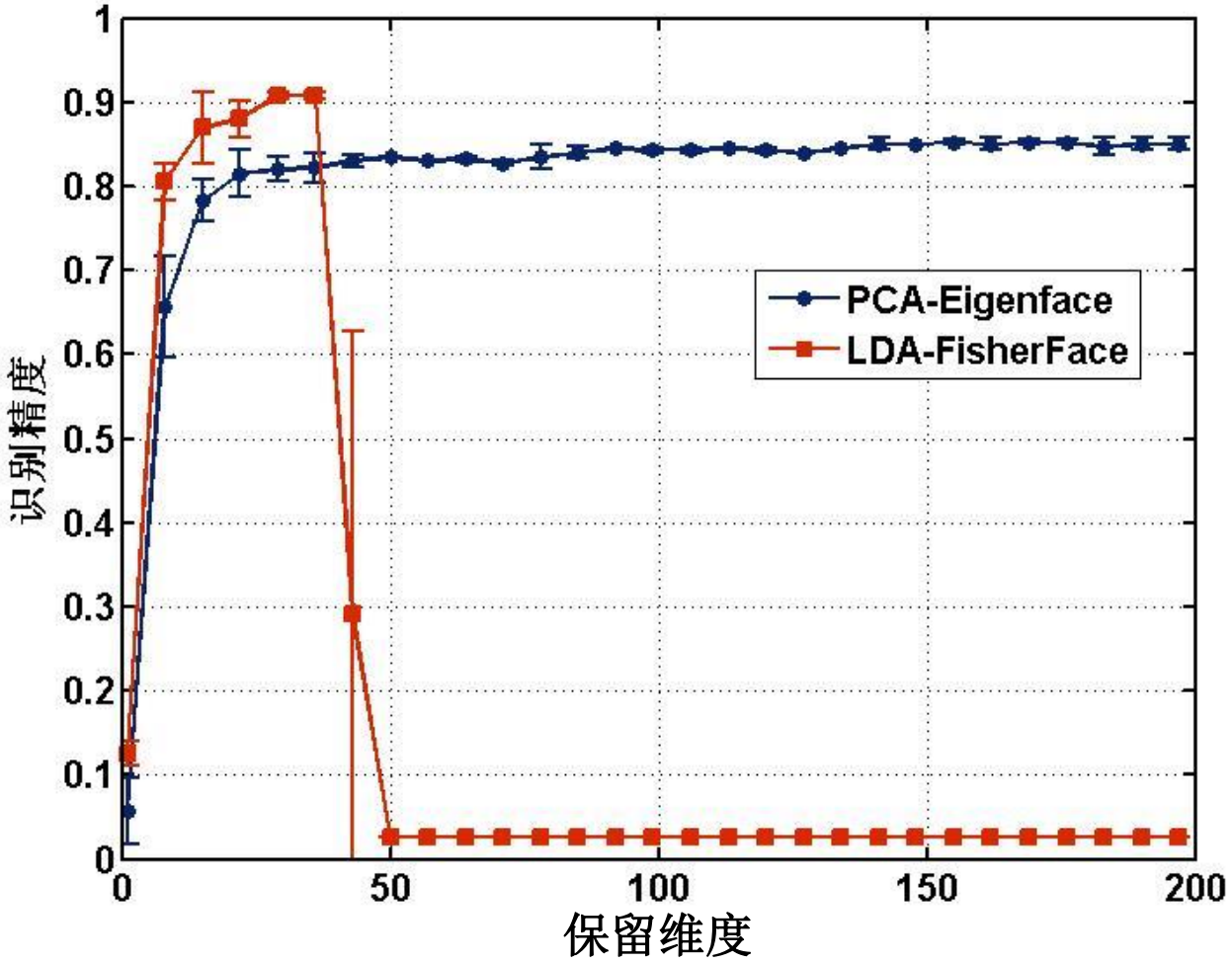
实验

- ORL数据库：
  - 32×32 灰度人脸图片 400张 (40类×10张人脸)
  - 验证策略：二折交叉验证（每次200样本训练，200样本测试）
  - 分类器：最近邻分类器

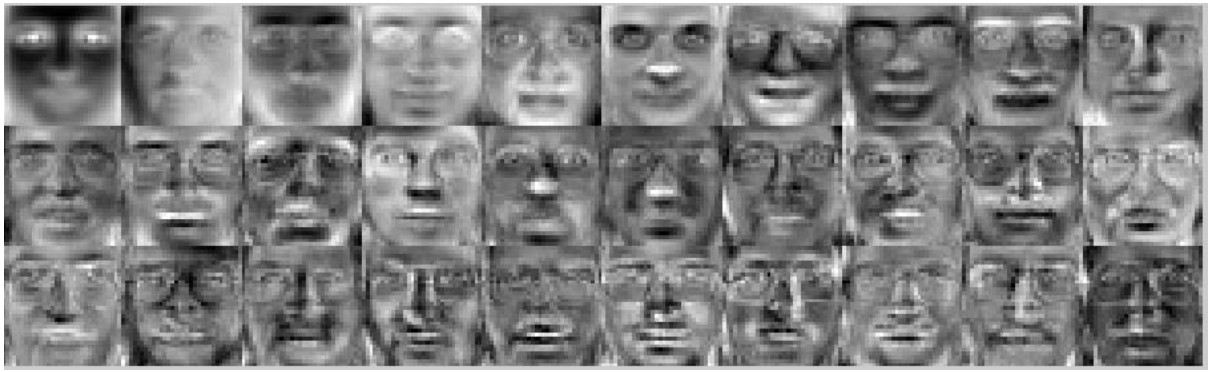
Eigenfaces Vs Fisherfaces

最优识别精度(%)：

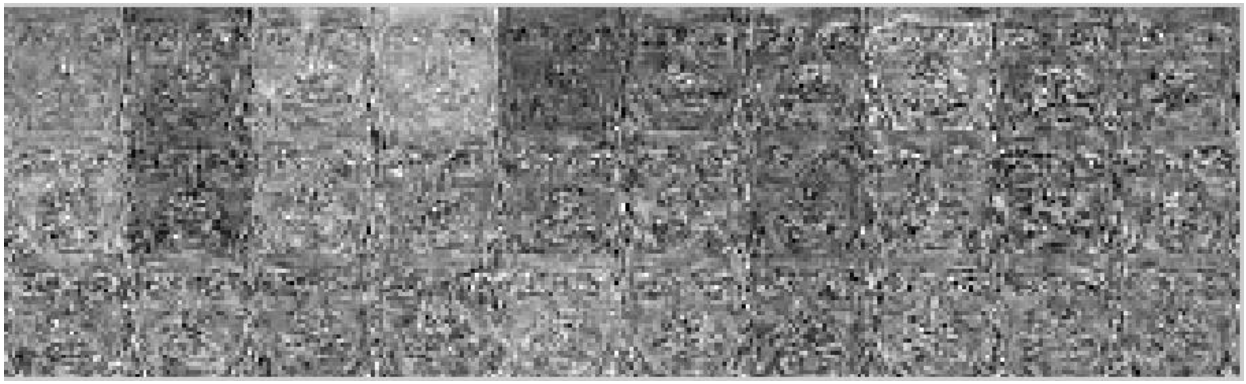
LDA+NN：91.50 ± 1.41 (31), PCA+NN: 85.25 ± 0.35 (153)



Eigenfaces and Fisherfaces on ORL dataset



从ORL数据库训练得到的前30个Eigenfaces（PCA投影矩阵前30个基）



从ORL数据库训练得到的前30个Fisherfaces（LDA投影矩阵前30个基

作业

- 任选一门语言，实现PCA与LDA算法，用于解决人脸识别问题，并在ORL数据库上进行测试。（不需要上交）