

---

# 降维任务

陈飞宇

[fchen@cqu.edu.cn](mailto:fchen@cqu.edu.cn)

办公室：软件学院529

# 降维

---

## 为什么要降维？

- 去除不相关的特征（去噪、特征提取）
- 储存与计算
- 可视化
- 数据本身具有低维特点

# Dimension Reduction Algorithms

---

## 线性降维方法:

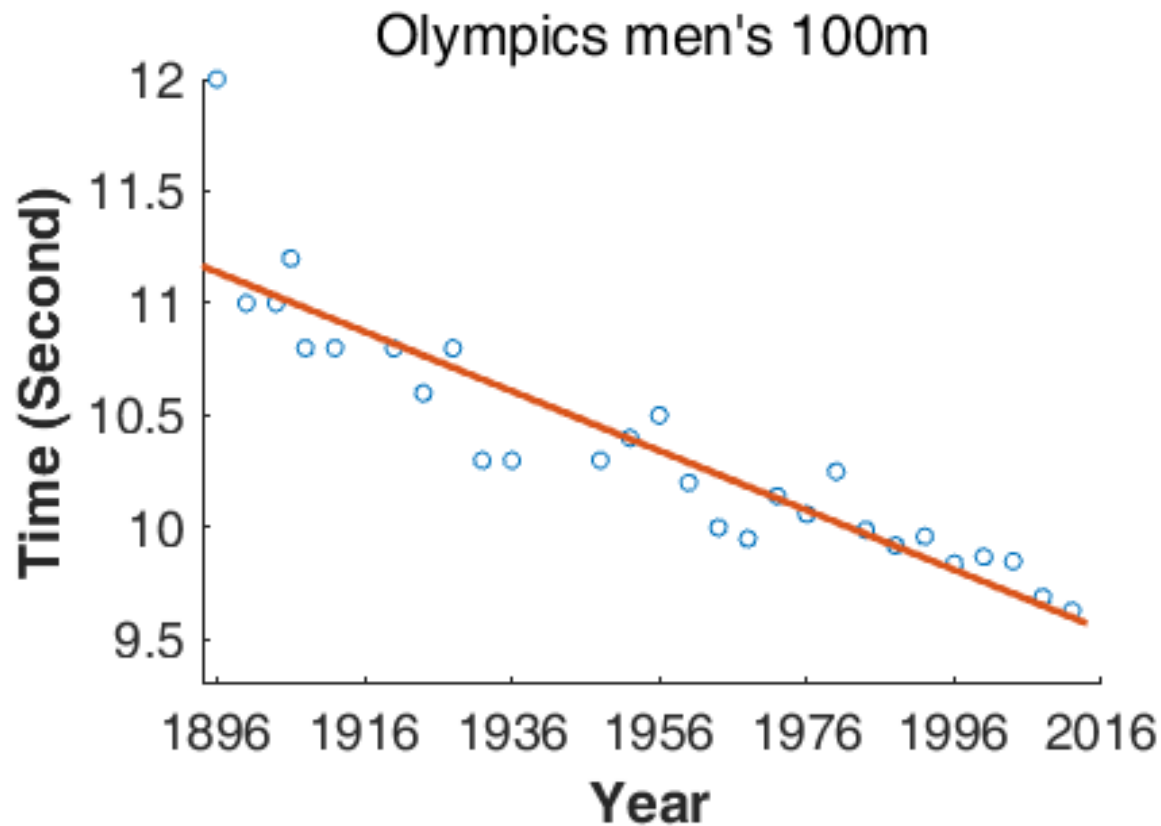
- PCA (Pearson, 1901)
- LDA (Fisher, 1936)
- SVD (Golub and Van Loan, 1983)
- NMF(Lee and Seung, 1999)

## 非线性降维方法:

- Kernel PCA (Scholkopf et al., 1998)
- Isomap (Tenenbaum et al., 2000)
- MDS (Cox and Cox 2001)

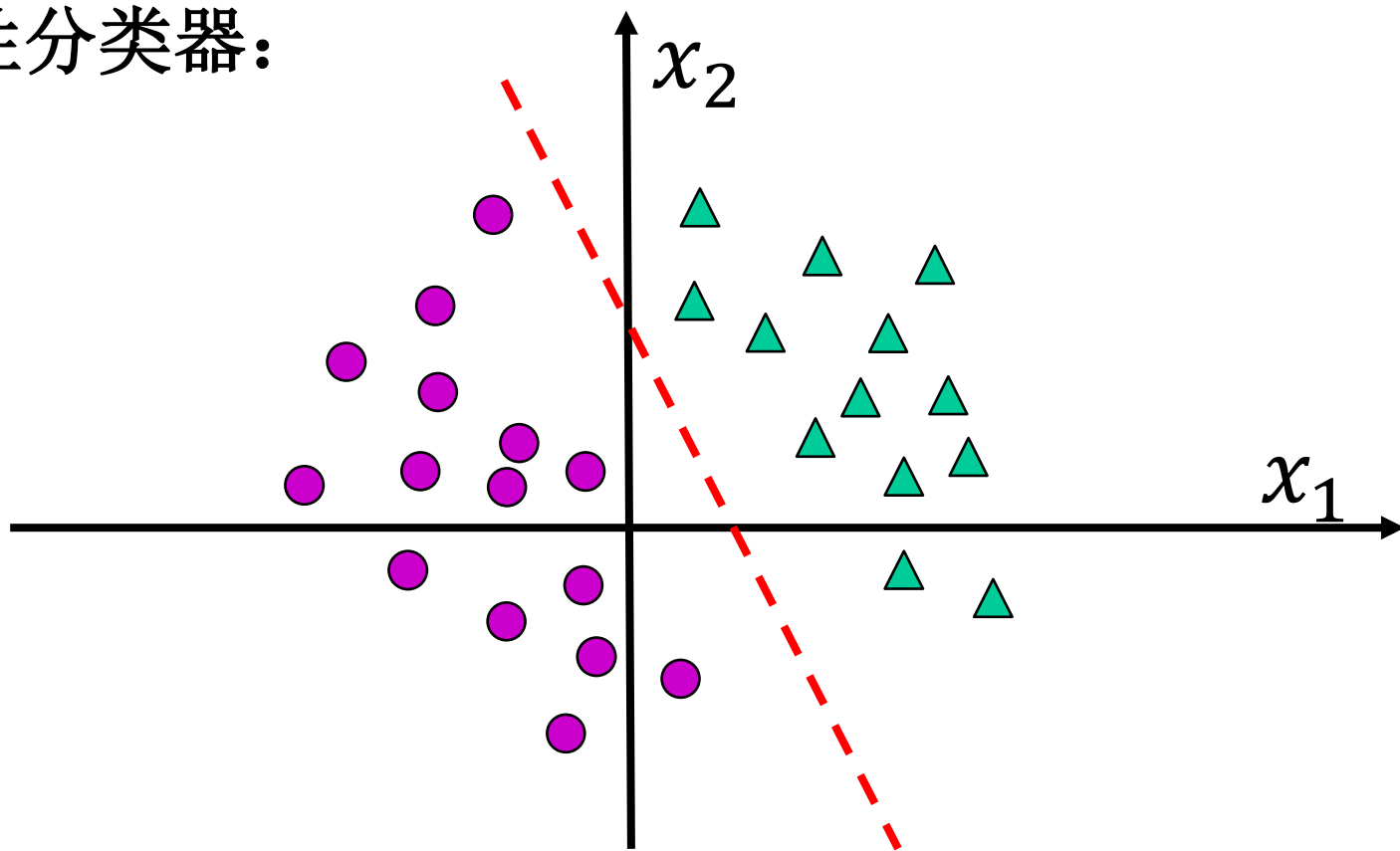
# 回归任务—直线

线性回归：



# 分类任务---直线

线性分类器:



---

# 6.1 主成分分析

陈飞宇

[fchen@cqu.edu.cn](mailto:fchen@cqu.edu.cn)

办公室：软件学院529

# 基础知识

方差 (variance) :

$$\text{Var}(X) = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

其中  $E[X]$  表示随机变量  $X$  的期望。

协方差 (covariance) :

$$\text{cov}(X, Y) = E[(X - E[X])(Y - E[Y])^T]$$

特别地,  $\text{cov}(X) = \text{Var}(X) = \frac{1}{n-1} XX^T$

# 基础知识

方差 (variance) :

$$\text{Var}(X) = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

其中  $E[X]$  表示随机变量  $X$  的期望。

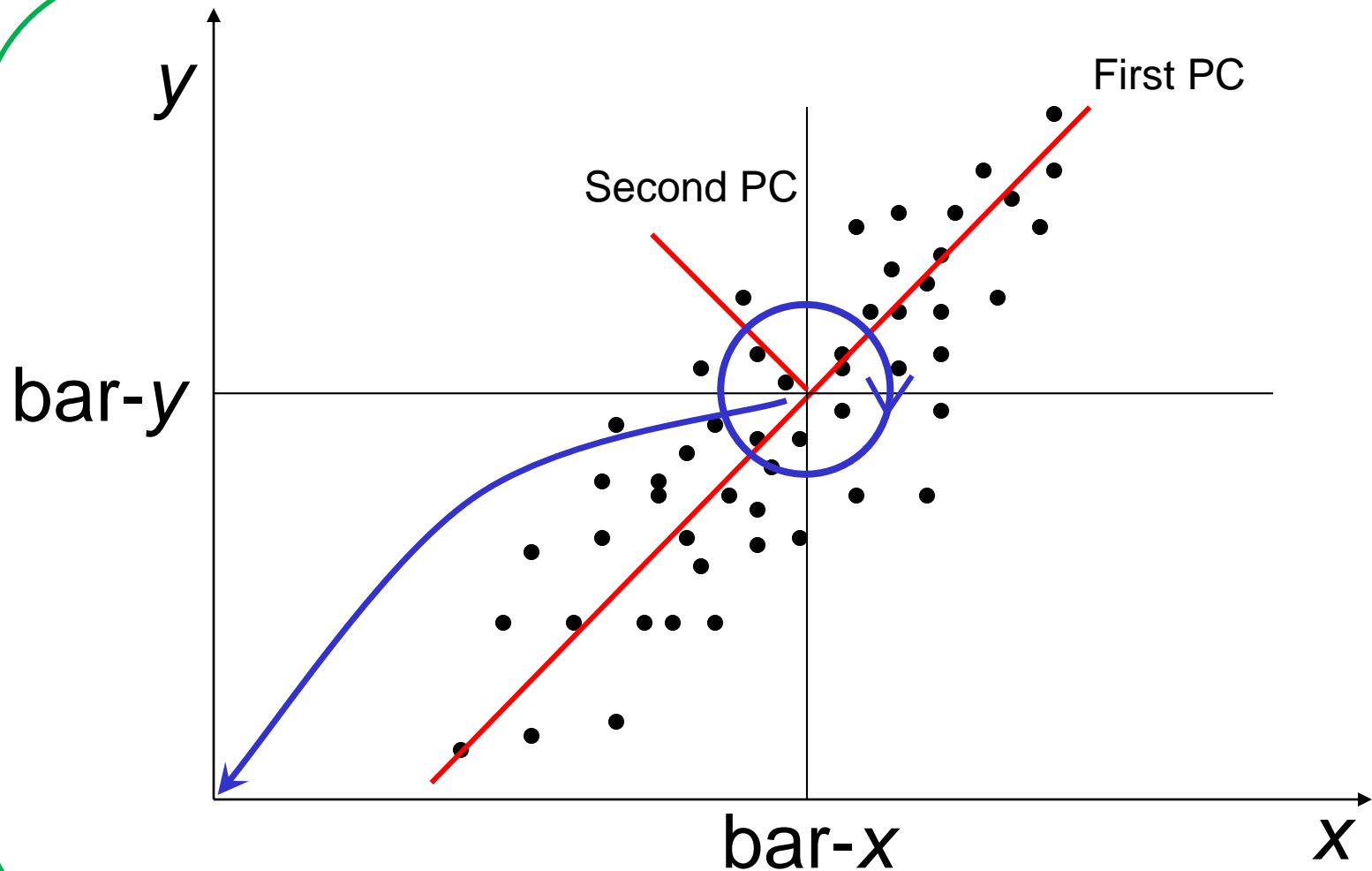
协方差 (covariance) :

$$\text{cov}(X, Y) = E[(X - E[X])(Y - E[Y])^T]$$

特别地,  $\text{cov}(X) = \text{Var}(X) = \frac{1}{n-1} XX^T$

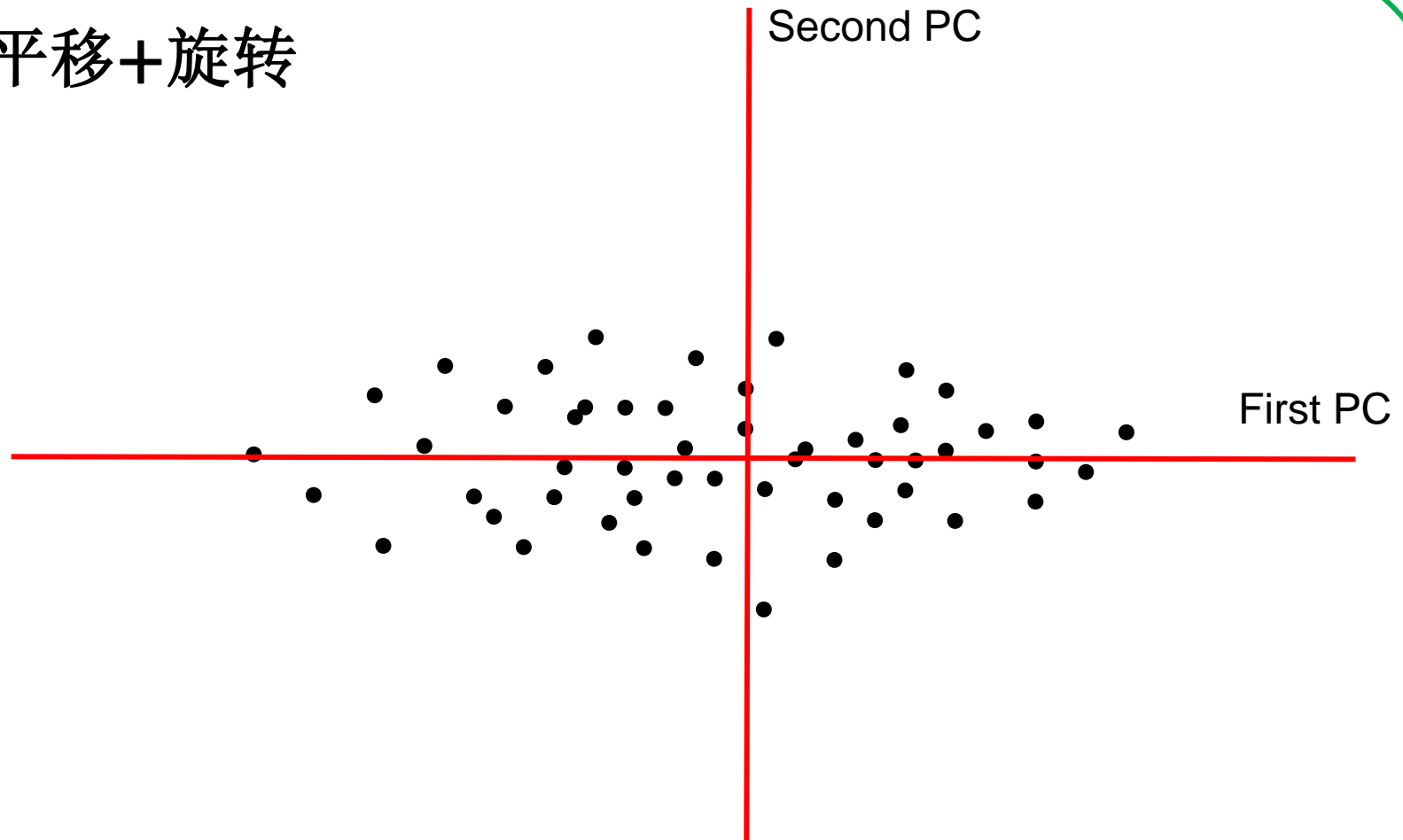


# 主成分分析示例



# 主成分分析示例

平移+旋转



# 主成分分析

精确描述：

- 1<sup>st</sup> pc 包含了样本方差的最大方向。
- 2<sup>nd</sup> pc 与第一个主成分**不相关**（夹角90度），包含了剩余样本方差的最大方向。
- 前几个主成分包含了样本的绝大部分信息，以至于我们可以忽略后面的主成分。

**核心思想：**寻找合适的正交投影矩阵  $U$ ，使得投影之后的样本  $Y = U^T X$  方差达到最大。

$$\max \text{Var}(Y)$$

$$\text{s.t. } Y = U^T X, U^T U = I$$

# 主成分分析

假设  $Y = U^T X$  是经过变换之后的数据,

$$\begin{aligned} & \text{Var}(Y) \\ &= E \left( (Y - E[Y])(Y - E[Y])^T \right) \\ &= E \left( (U^T X - E[U^T X])(U^T X - E[U^T X])^T \right) \\ &= U^T E \left( (X - E[X])(X - E[X])^T \right) U \\ &= U^T \text{Var}(X) U \end{aligned}$$

# 主成分分析

因此，PCA模型：

$$\begin{aligned} \max \quad & \text{Var}(Y) \\ \text{s. t.} \quad & Y = U^T X, \quad U^T U = I \end{aligned}$$

等价于

$$\begin{aligned} \max \quad & U^T \text{Var}(X) U \\ \text{s. t.} \quad & U^T U = I \end{aligned}$$

此模型的最优解是矩阵 $\text{Var}(X)$ 的前 $k$ 个特征向量

# 拉格朗日乘子法

主成分分析模型：

$$\max \quad U^T \text{Var}(X) U \quad \text{s.t.} \quad U^T U = I$$

拉格朗日函数：

$$\mathcal{L}(U, \Lambda) = -U^T \text{Var}(X) U + \Lambda(U^T U - I)$$

$\mathcal{L}$  对变量  $U$  求偏导：

$$\frac{\partial \mathcal{L}(U, \Lambda)}{\partial U} = -2\text{Var}(X)U + 2\Lambda U$$

极大极小问题（对偶问题）：

$$\max_{\Lambda} \min_U \mathcal{L}(U, \Lambda)$$

# 主成分分析

极小问题:

$$\begin{aligned}\min_U \mathcal{L}(U, \Lambda) &= -U^T \mathbf{Var}(X) U + \Lambda(U^T U - I) \\ &= -U^T \Lambda U + \Lambda(U^T U - I) = \Lambda\end{aligned}$$

极大极小问题:

$$\max_{\Lambda} \Lambda \quad \text{s.t.} \quad \mathbf{Var}(X) U = \Lambda U$$

主成分分析模型:

$$\max U^T \mathbf{Var}(X) U \quad \text{s.t.} \quad U^T U = I$$

等价于  $\mathbf{Var}(X) U = \Lambda U$  最大  $k$  个特征值对应的特征向量

# PCA:最大可分性

主成分分析模型:

$$\max \quad U^T \text{Var}(X) U$$

$$\text{s. t.} \quad U^T U = I$$

- 此模型的最优解是对称半正定矩阵  $\text{Var}(X)$  的前  $k$  个特征值所对应的特征向量。
- 此模型的目标是最大化投影后数据的方差，这个性质通常被称为PCA的 最大可分性 。



# 主成分分析

**PCA 算法流程:**

**Step 1: 中心化**  $\bar{X} = X - E[X]$

**Step 2: 计算协方差矩阵**  $C = \text{cov}(\bar{X}) = \frac{1}{n-1} \bar{X} \bar{X}^T$

**Step 3: 特征值分解**  $C = U \Lambda U^T$

**Step 4: 降维**  $Y = U_k^T \bar{X}$ , 其中  $U_k^T$  为  $U^T$  的前  $k$  行。

# 主成分分析

中心化前：

$$\mathbf{cov}(X) = E[(X - E[X])(X - E[X])^T] = E[\bar{X}\bar{X}^T]$$

中心化后：

$$\mathbf{cov}(\bar{X}) = E[(\bar{X} - E[\bar{X}])(\bar{X} - E[\bar{X}])^T] = E[\bar{X}\bar{X}^T]$$

因此，

$$\mathbf{cov}(X) = \mathbf{cov}(\bar{X}) = \frac{1}{n-1} \bar{X}\bar{X}^T$$

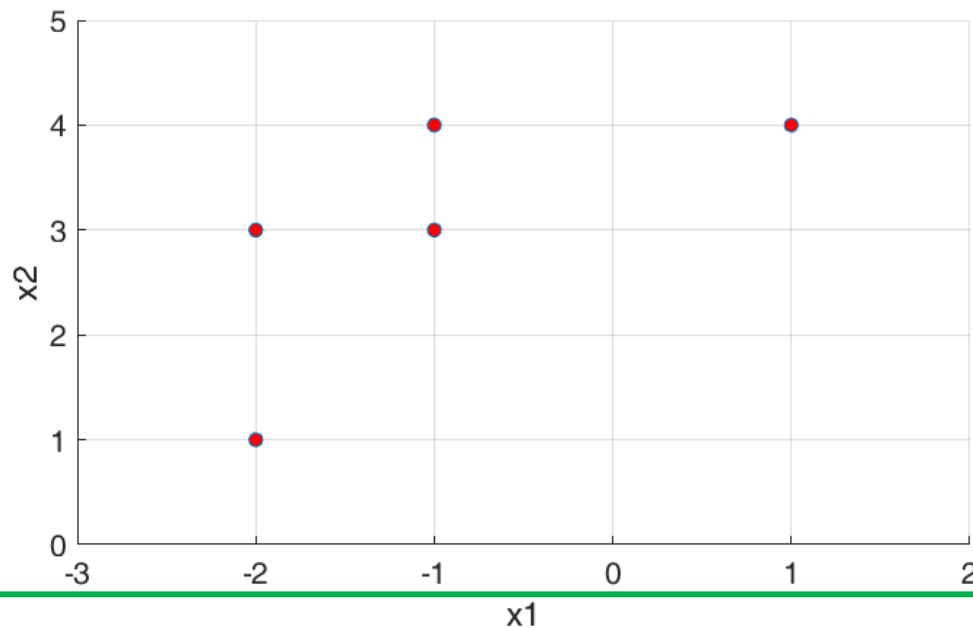
问题：中心化的意义？

# 例子

假设二维数据点集为

$$X = \begin{pmatrix} -2 & -2 & -1 & 1 & -1 \\ 1 & 3 & 3 & 4 & 4 \end{pmatrix}$$

试利用主成分分析算法对该数据进行降维。



# 例子

解:  $X = \begin{pmatrix} -2 & -2 & -1 & 1 & -1 \\ 1 & 3 & 3 & 4 & 4 \end{pmatrix}$

Step 1: 中心化

$$\bar{X} = X - E[X] = \begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Step 2: 计算协方差矩阵

$$C = \text{cov}(\bar{X}) = \frac{1}{n-1} \bar{X} \bar{X}^T = \frac{1}{4} \begin{pmatrix} 6 & 4 \\ 4 & 6 \end{pmatrix}$$

# 例子

## Step 3: 特征值分解

$$\begin{aligned} C &= U \Lambda U^T \\ &= \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 5/2 & 0 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \end{aligned}$$

## Step 4: 投影、降维

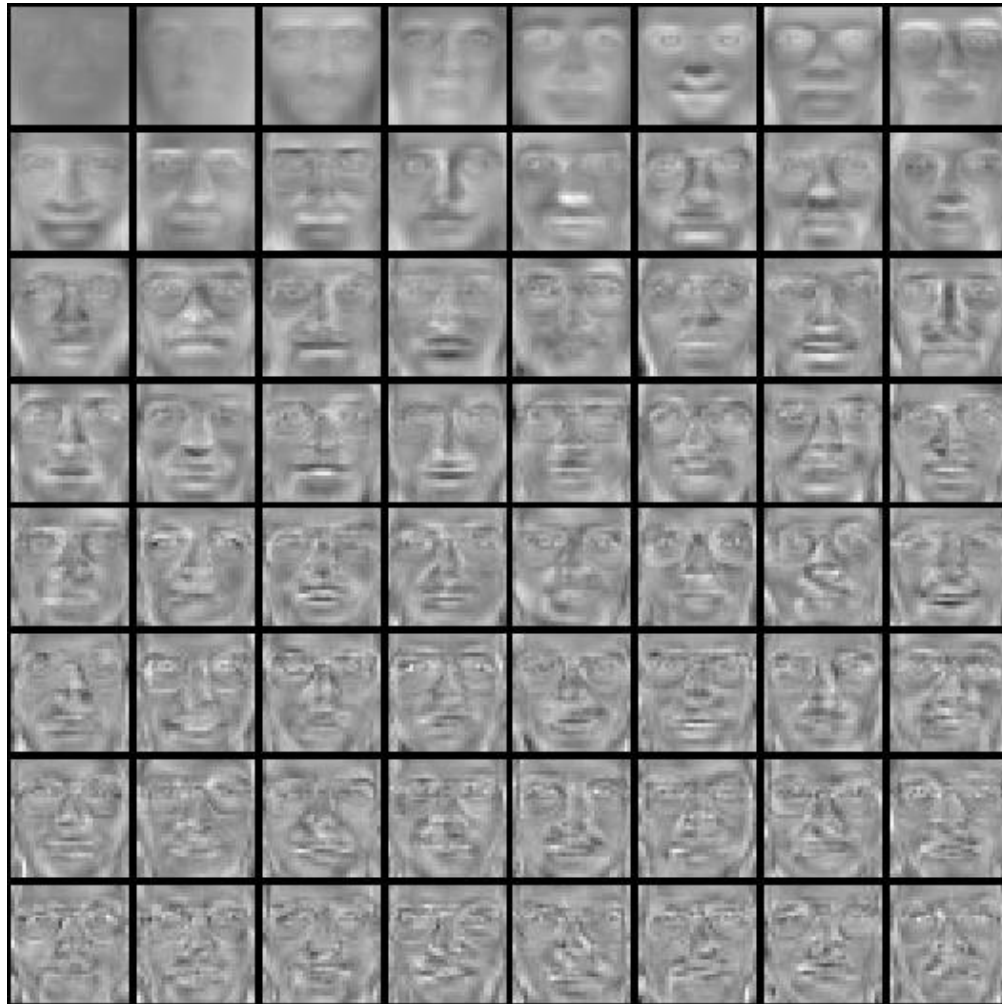
$$\begin{aligned} Y &= U_1^T \bar{X} \\ &= \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix} \\ &= \begin{pmatrix} -3/\sqrt{2} & -1/\sqrt{2} & 0 & 3/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \end{aligned}$$

# PCA应用—人脸识别



# PCA应用—Eigenface

---



# PCA局限性

---

What if very high dimensional data?

- e.g., HD image ( $d \geq 10^4$ )

Problem:

- Covariance matrix  $C$  is size ( $d^2$ )
- $d = 10^4 \rightarrow |C| = 10^8$

Speedup:

Singular Value Decomposition (SVD)



---

## 6.2 奇异值分解

陈飞宇

[fchen@cqu.edu.cn](mailto:fchen@cqu.edu.cn)

办公室：软件学院529

# Singular Value Decomposition

$$A_{m \times n} = U_{m \times r} \Sigma_{r \times r} V_{n \times r}^T$$

- **A: 数据矩阵**

$m \times n$  matrix (e.g.,  $m$  documents,  $n$  terms)

- **U: 左奇异值向量:  $U^T U = I$**

$m \times r$  matrix ( $m$  documents,  $r$  concepts)

- **$\Sigma$ : 奇异值,  $r$  为矩阵A的秩**

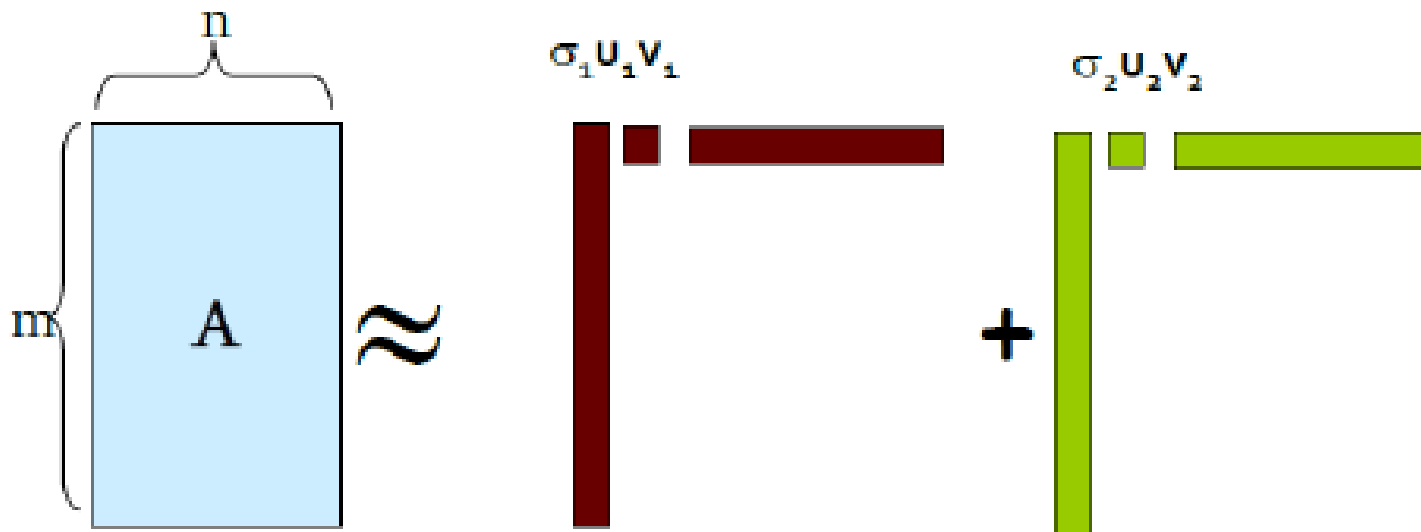
$r \times r$  diagonal matrix (strength of ‘concept’)

- **V: 右奇异值向量:  $V^T V = I$**

$n \times r$  matrix ( $n$  terms,  $r$  concepts)

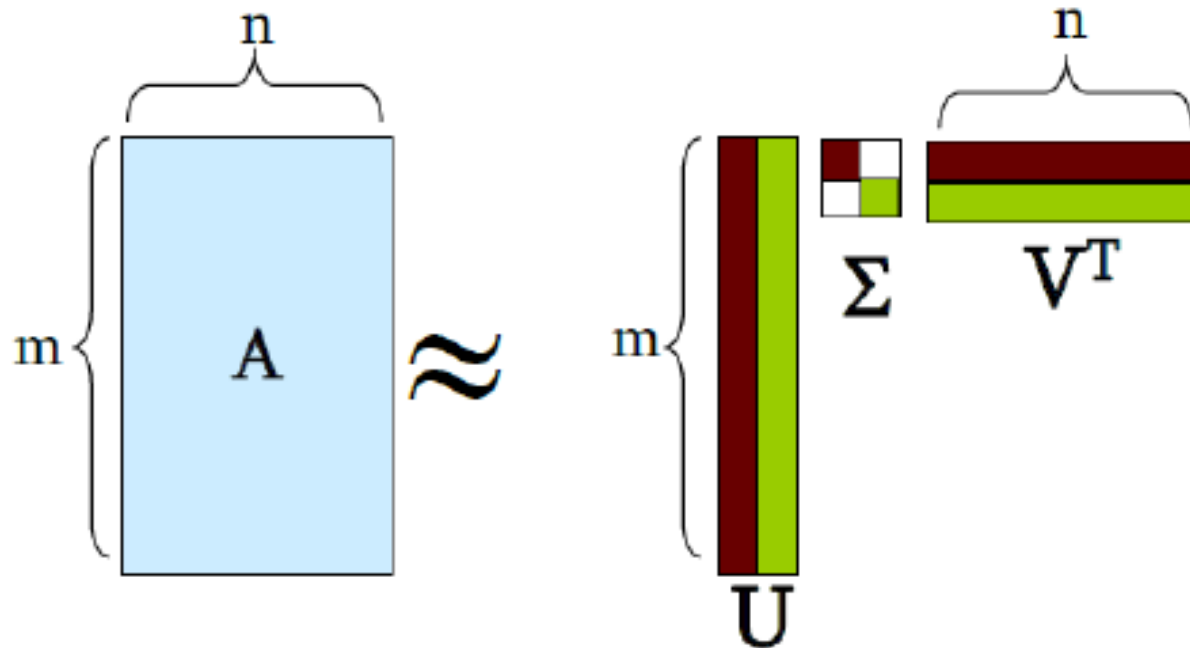
# Singular Value Decomposition

$$A \approx U_{m \times k} \Sigma_{k \times k} V_{n \times k}^T = \sum_{i=1}^k \sigma_i u_i v_i^T$$



# Singular Value Decomposition

$$A_{m \times n} \approx U_{m \times k} \Sigma_{k \times k} V^T_{n \times k}$$



# SVD 例子

Problem:

- #1: Find concepts in text
- #2: Reduce dimensionality

term document	data	information	retrieval	brain	lung
CS-TR1	1	1	1	0	0
CS-TR2	2	2	2	0	0
CS-TR3	1	1	1	0	0
CS-TR4	5	5	5	0	0
MED-TR1	0	0	0	2	2
MED-TR2	0	0	0	3	3
MED-TR3	0	0	0	1	1

# SVD 解释

---

‘documents’, ‘terms’ and ‘concepts’:

- $\mathbf{U}$ : document-to-concept similarity matrix
- $\mathbf{V}$ : term-to-concept similarity matrix
- $\Sigma$ : its diagonal elements: ‘strength’ of each concept

# SVD 例子

$$\begin{array}{c} \uparrow \\ \text{CS} \\ \downarrow \\ \uparrow \\ \text{MD} \\ \downarrow \end{array}
 \begin{bmatrix} & \text{data} & \text{information} & \text{retrieval} & \text{brain} & \text{lung} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} & = & \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} & \times & \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} & \times & \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}
 \end{array}$$

CS-concept

MD-concept

doc-to-concept

# SVD 例子

CS

↑

↓

MD

↑

↓

$$\begin{bmatrix}
 \text{data} & \text{information} & \text{retrieval} & \text{brain} & \text{lung} \\
 1 & 1 & 1 & 0 & 0 \\
 2 & 2 & 2 & 0 & 0 \\
 1 & 1 & 1 & 0 & 0 \\
 5 & 5 & 5 & 0 & 0 \\
 0 & 0 & 0 & 2 & 2 \\
 0 & 0 & 0 & 3 & 3 \\
 0 & 0 & 0 & 1 & 1
 \end{bmatrix} = \begin{bmatrix}
 0.18 & 0 \\
 0.36 & 0 \\
 0.18 & 0 \\
 0.90 & 0 \\
 0 & 0.53 \\
 0 & 0.80 \\
 0 & 0.27
 \end{bmatrix} \times \begin{bmatrix}
 9.64 & 0 \\
 0 & 5.29
 \end{bmatrix} \times \begin{bmatrix}
 0.58 & 0.58 & 0.58 & 0 & 0 \\
 0 & 0 & 0 & 0.71 & 0.71
 \end{bmatrix}$$

‘strength’ of CS-concept

‘strength’ of MD-concept



# SVD 例子

Diagram illustrating the SVD decomposition of a matrix and the resulting components.

The matrix is defined by the following structure:

	data	inf.	retrieval	brain	lung
CS	1	1	1	0	0
	2	2	2	0	0
	1	1	1	0	0
	5	5	5	0	0
	0	0	0	2	2
	0	0	0	3	3
	0	0	0	1	1

The matrix is decomposed into three components:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

Labels and arrows indicate the components:

- MD-concept**: Points to the first matrix (left).
- CS-concept**: Points to the second matrix (middle).
- term-to-concept**: Points to the third matrix (right).
- CS**: Points to the first matrix (left).
- MD**: Points to the first matrix (left).

# SVD 例子

降维：另较小的奇异值为0.

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

# SVD 例子

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \approx \begin{bmatrix} 0.18 \\ 0.36 \\ 0.18 \\ 0.90 \\ 0 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} 9.64 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \end{bmatrix}$$

# SVD 例子

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# 奇异值分解

---

- 对于任意矩阵来说，它的奇异值分解存在唯一，且所有奇异值都是正数。
- 利用奇异值分解进行降维处理，其意义是最小化重构误差。

# 奇异值分解

**SVD 算法流程:**

**Step 1: 中心化  $\bar{X} = X - E[X]$**

**Step 2: 奇异值分解  $\bar{X} = U\Sigma V^T$**

**Step 3: 降维  $Y = U_k^T \bar{X}$ , 其中  $U_k^T$  为  $U^T$  的前k行**

。

# PCA vs SVD

- **SVD** –  $\bar{X} = U\Sigma V^T$
- **PCA** –  $\bar{X}\bar{X}^T = U\Sigma V^T (U\Sigma V^T)^T = U\Sigma^2 U^T$   
 $\bar{X}^T \bar{X} = (U\Sigma V^T)^T U\Sigma V^T = V\Sigma^2 V^T$

SVD on  $\bar{X}$



PCA on  $\bar{X}$

# PCA vs SVD

## Computational Complexity:

- **SVD** –  $O(mn^2)$  or  $O(nm^2)$  或者更少
  - 只需要奇异值
  - 或者只需要前几个奇异向量
  - 或者矩阵是稀疏的
- **PCA** –  $O(n^3)$

## Storage:

- **SVD** –  $O(mn)$  (e.g., 数据矩阵)
- **PCA** –  $O(n^2)$  (e.g., 协方差矩阵)



# PCA vs SVD

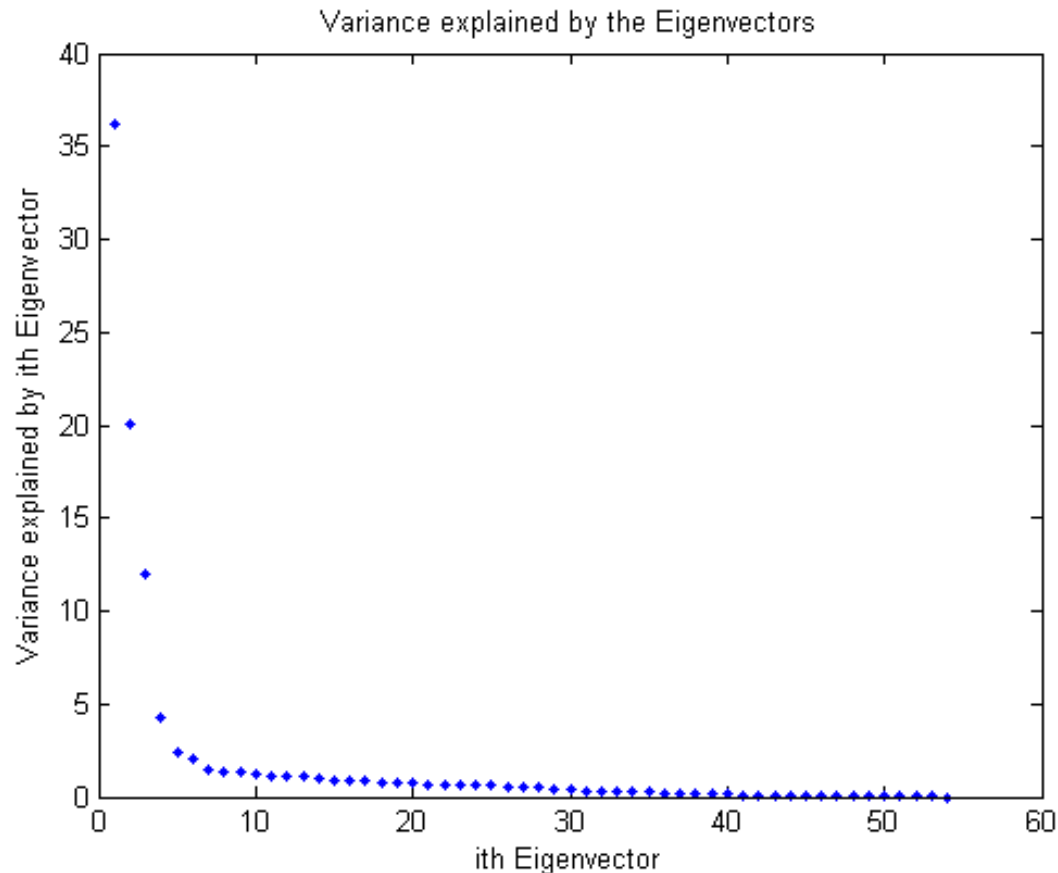
---

## Matlab 对比试验

- 随机数据  $[m,n] = [100,4000]$
- PCA运行时间: 69.6601s
- SVD 运行时间: 0.3414s

# 主成分个数选取？

观察特征值或奇异值的分布情况来选取合适的 $k$ 。



# 主成分分析小结

- 主成分分析（PCA）是最常用的降维技术。
- 主成分分析最大化投影后数据的方差，同时主成分分析还最小化重建误差。
- 从数值上看，奇异值分解可以大幅度地提高主成分分析算法的计算速度。

---

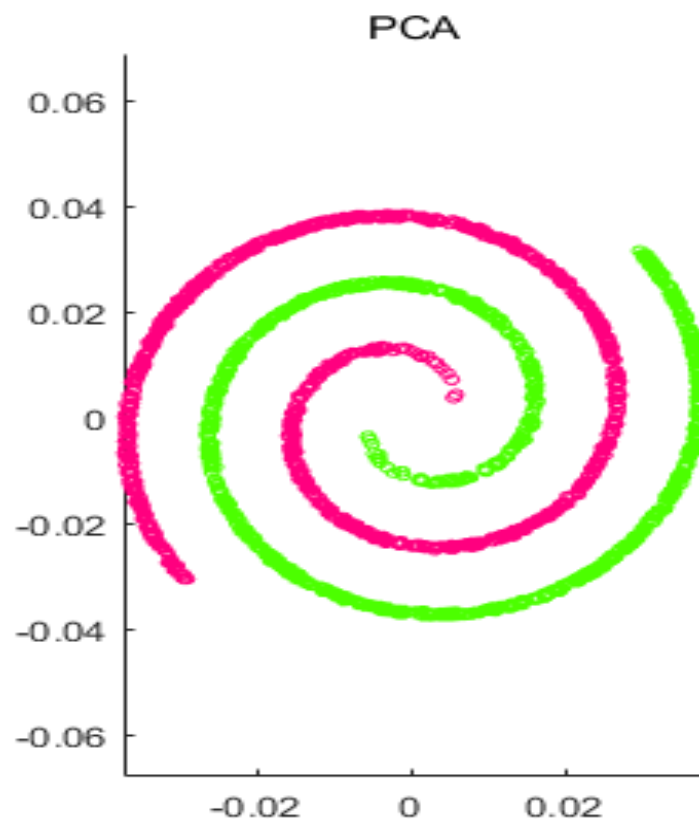
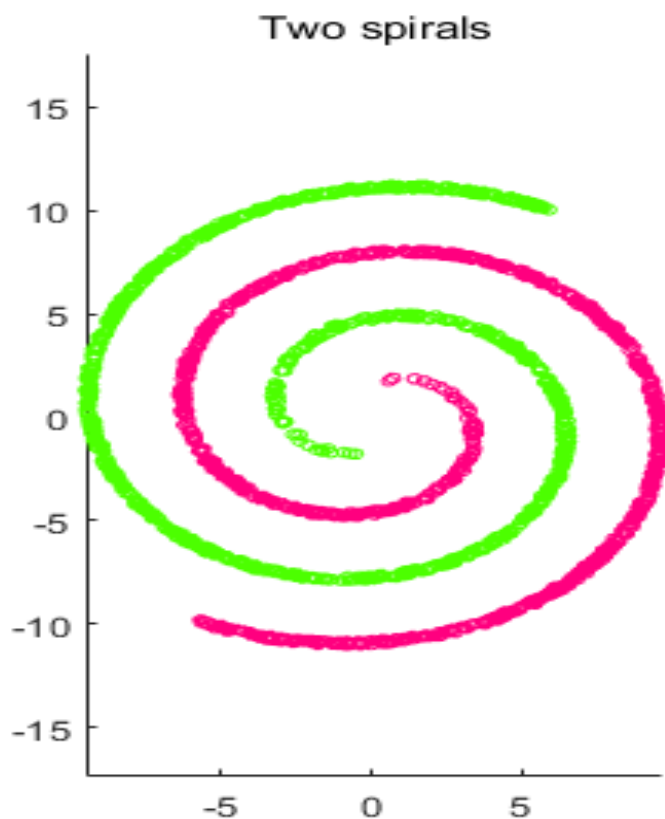
## 6.3 核化PCA

陈飞宇

[fchen@cqu.edu.cn](mailto:fchen@cqu.edu.cn)

办公室：软件学院529

# 非球状数据降维



# 解决方案

对于非线性可分的数据 $\{x_i\}_{i=1}^m$ ，尝试找到一个非线性映射  $\phi$ ，使得数据 $\{\phi(x_i)\}_{i=1}^n$ 在新的空间（通常为高维空间）是线性可分的，然后再使用PCA进行降维。

椭圆方程:  $w_1(x^{(1)})^2 + w_2(x^{(2)})^2 + b = 0$

直线方程:  $w_1x^{(1)} + w_2x^{(2)} + b = 0$

# 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  $\phi$  是不现实的。

定义核函数 (Kernel Function) :

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

这样， $\phi$  可以隐式地由核函数表示出来，而且内积的计算也变得更加容易。

# 核函数

常见的核函数（核矩阵对称半正定）：

名称	表达式
线性核	$K(x_i, x_j) = x_i^T x_j$
多项式核	$K(x_i, x_j) = (x_i^T x_j)^d$
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	$K(x_i, x_j) = \tanh(\beta x_i^T x_j + \theta)$



# 核化主成分分析

---

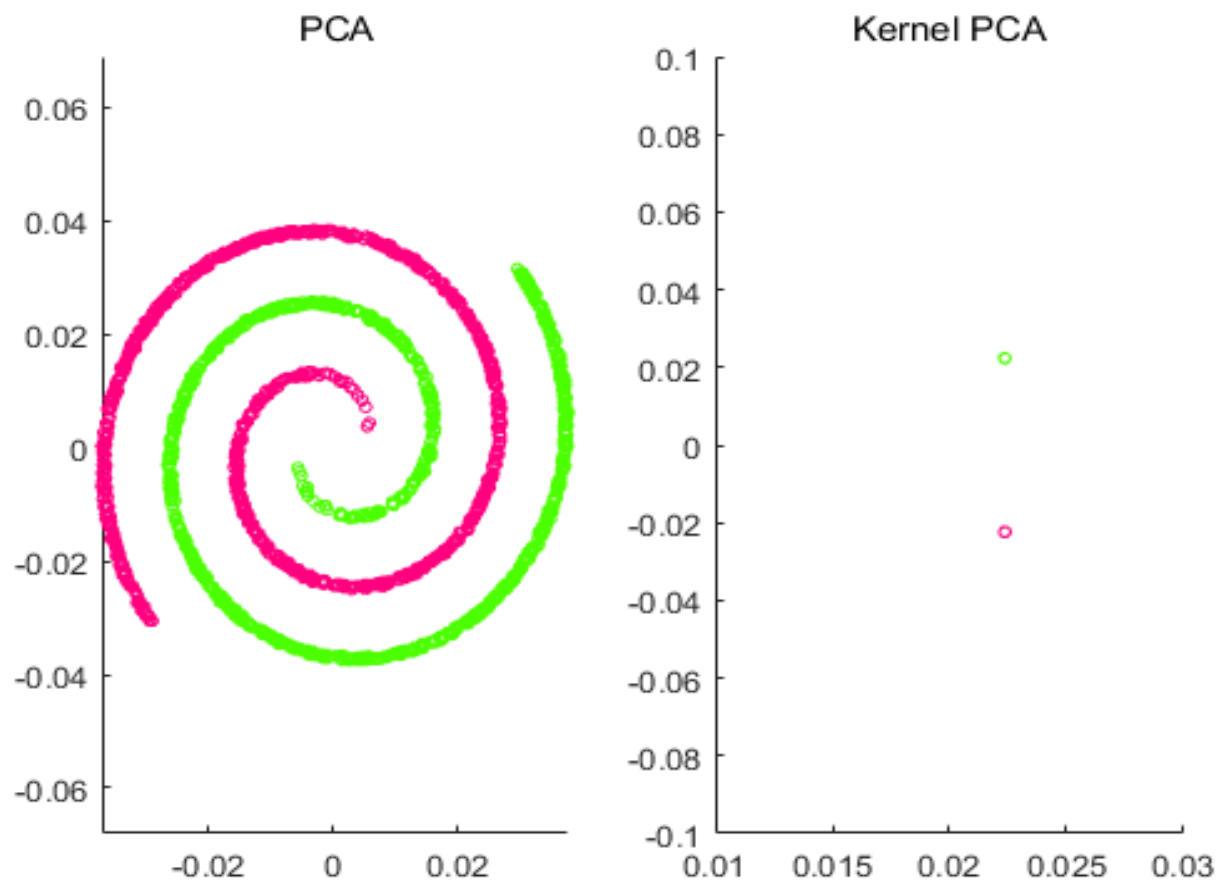
**KPCA 算法流程:**

**Step 1:** 选取合适的核函数, 求出矩阵  $K$

**Step 2:** 特征值分解  $K = U\Lambda U^T$

**Step 3:** 降维  $Y = U_k^T \Lambda^{1/2}$ 。

# 核化主成分分析



# 核化主成分分析小结

---

- 核化主成分分析（**KPCA**）是非线性降维方法，是**PCA**的直接推广。
- **KPCA** 可以对非球状数据进行降维。
- 和**PCA**类似，当数据量较大时，**KPCA**的计算效率低。

---

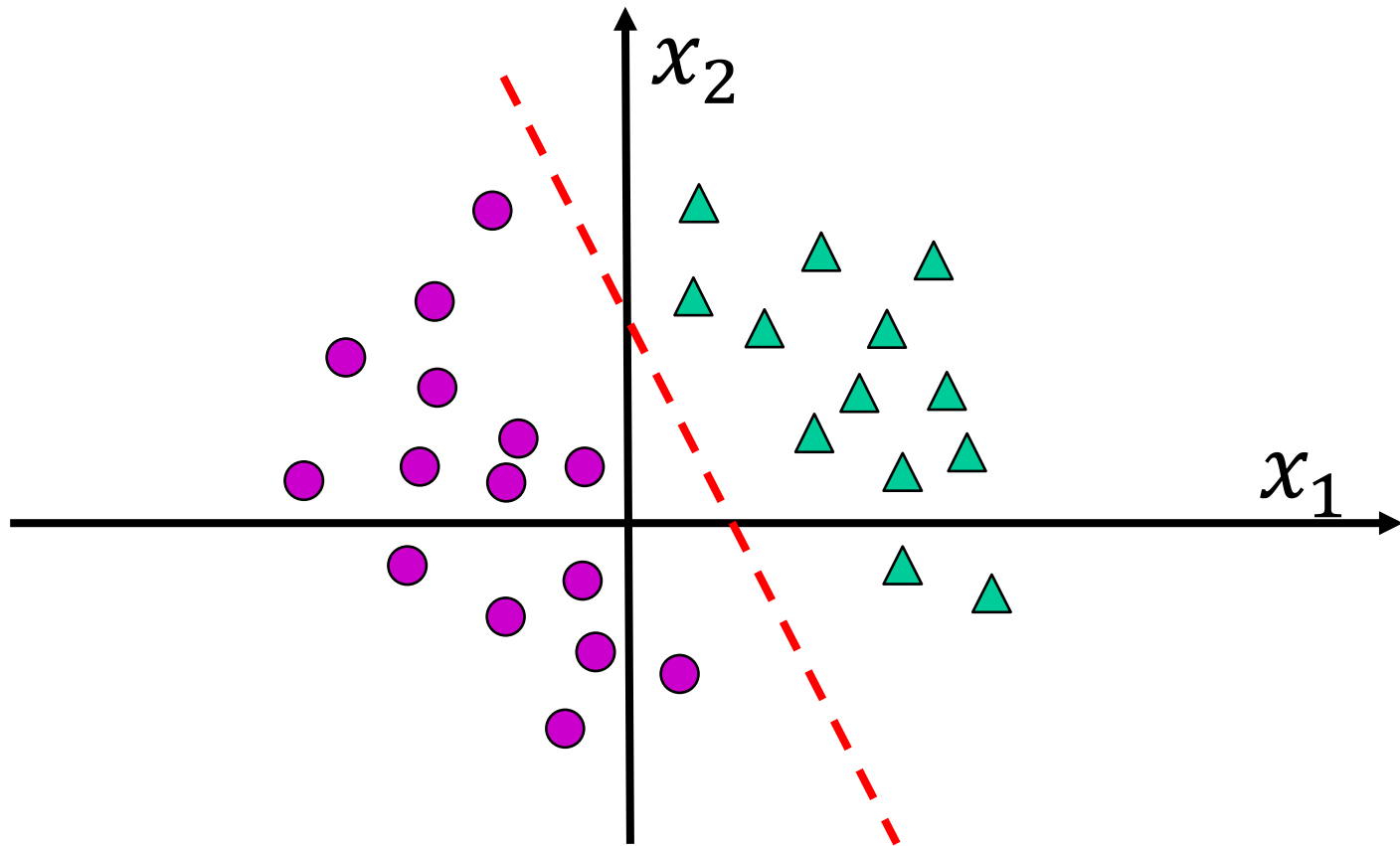
## 6.4 线性判别分析

陈飞宇

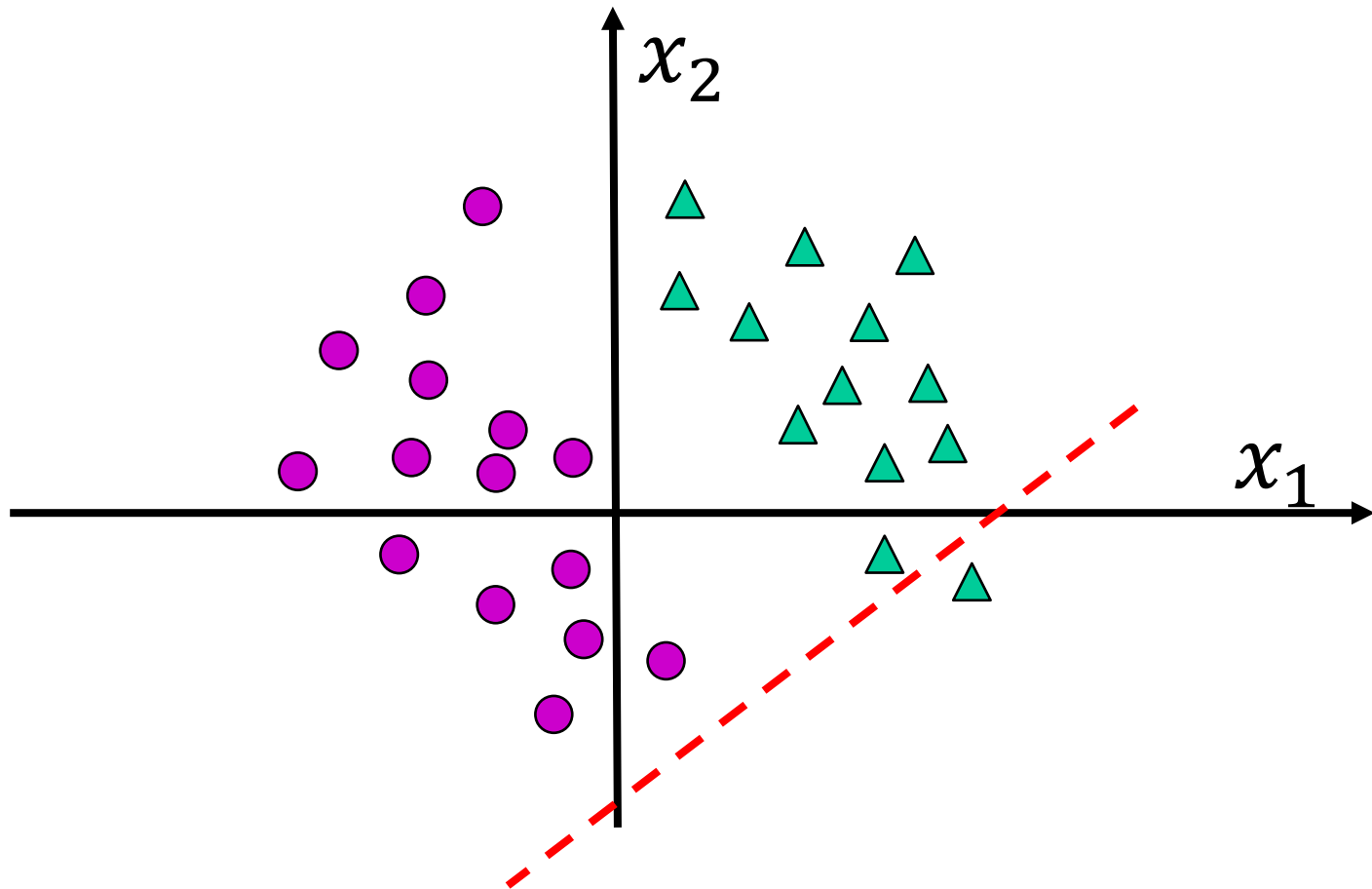
[fchen@cqu.edu.cn](mailto:fchen@cqu.edu.cn)

办公室：软件学院529

# 降维+分类？



# 降维+分类？



# 降维+分类？

---

思想：假设降维的目的是解决维度灾难，在低维空间进行分类。那么我们希望降维后的样本保持一定的**可分性**，即经过投影之后，**同类样本**的投影点尽可能**接近**，**异类样本**的投影点尽可能**远离**。

解决方案：让投影后的类中心之间的距离尽可能大，从而增强样本的可分性。

# 降维+分类?

对于二分类问题，给定训练集  $\{(x_i, y_i)\}_{i=1}^n$ ，其中  $y_i \in \{0, 1\}$ 。令  $D_i$ ,  $\mu_i$ ,  $\Sigma_i$  分别表示  $i \in \{0, 1\}$  类样本的集合、均值向量和协方差矩阵。

假设投影矩阵是  $w$ , 则投影后的中心为:

$$\frac{1}{|D_i|} \sum_{x \in D_i} w^T x = w^T \left( \frac{1}{|D_i|} \sum_{x \in D_i} x \right) = w^T \mu_i$$

因此投影后的类中心之间的距离为:

$$\|w^T \mu_0 - w^T \mu_1\|_2^2 = w^T (\mu_0 - \mu_1) (\mu_0 - \mu_1)^T w$$



# 降维+分类？

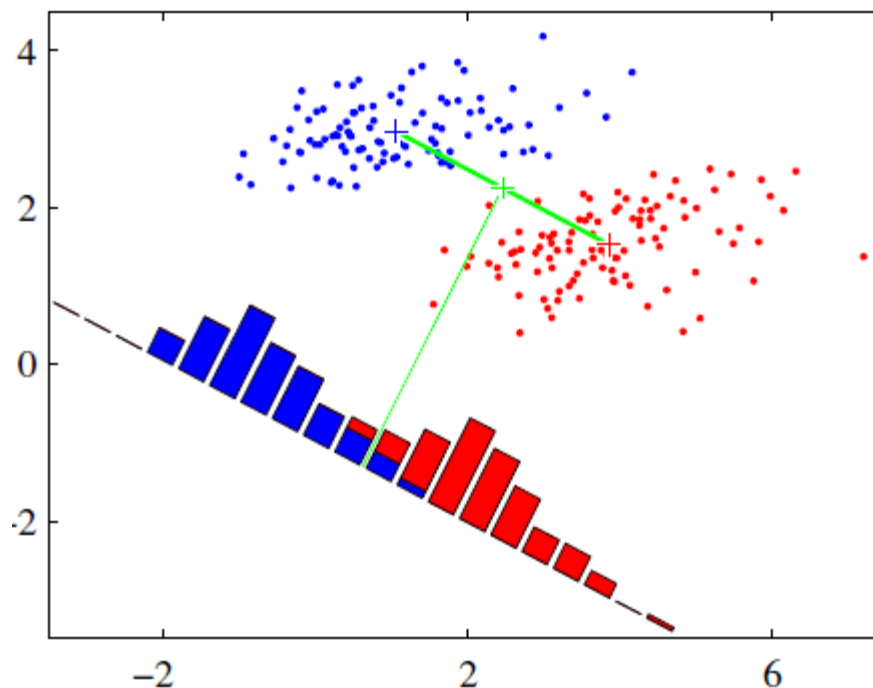
类间散度矩阵（between-class scatter matrix）：

$$S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T$$

最大化类间散度？

$$\max_w w^T S_b w$$

投影后有重合区域！



# 降维+分类?

投影后类内的样本不够‘紧凑’，即协方差过大，希望投影后类内样本的协方差小。

$$\min_w w^T \Sigma_0 w + w^T \Sigma_1 w$$

类内散度矩阵（within-class scatter matrix）：

$$\begin{aligned} S_w &= \Sigma_0 + \Sigma_1 \\ &= \sum_{x \in D_0} (x - \mu_0)(x - \mu_0)^T + \sum_{x \in D_1} (x - \mu_1)(x - \mu_1)^T \end{aligned}$$

# 线性判别分析

对于二维数据来说，线性判别分析（Linear Discriminant Analysis）同时最大化类间散度矩阵，最小化类内散度矩阵。

$$\max_w \frac{w^T S_b w}{w^T S_w w}$$

目标函数通常被成为广义瑞利商（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）。

# 线性判别分析

线性判别分析(LDA):

$$\max \mathbf{w}^T \mathbf{S}_b \mathbf{w} \quad \text{s.t.} \quad \mathbf{w}^T \mathbf{S}_w \mathbf{w} = 1$$

拉格朗日函数:

$$\mathcal{L}(\mathbf{w}, \lambda) = -\mathbf{w}^T \mathbf{S}_b \mathbf{w} + \lambda(\mathbf{w}^T \mathbf{S}_w \mathbf{w} - 1)$$

$\mathcal{L}$  对变量  $\mathbf{w}$  求偏导:

$$\frac{\partial \mathcal{L}(\mathbf{w}, \lambda)}{\partial \mathbf{w}} = -2\mathbf{S}_b \mathbf{w} + 2\lambda \mathbf{S}_w \mathbf{w}$$

极大极小问题（对偶问题）：

$$\max_{\lambda} \min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \lambda)$$

# 线性判别分析

极小问题:

$$\begin{aligned}\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \lambda) &= -\mathbf{w}^T \mathbf{S}_b \mathbf{w} + \lambda(\mathbf{w}^T \mathbf{S}_w \mathbf{w} - 1) \\ &= -\mathbf{w}^T \lambda \mathbf{S}_w \mathbf{w} + \lambda(\mathbf{w}^T \mathbf{S}_w \mathbf{w} - 1) = \lambda\end{aligned}$$

极大极小问题:

$$\max_{\lambda} \lambda \quad \text{s. t.} \quad \mathbf{S}_b \mathbf{w} = \lambda \mathbf{S}_w \mathbf{w}$$

线性判别分析(LDA):

$$\max \mathbf{w}^T \mathbf{S}_b \mathbf{w} \quad \text{s. t.} \quad \mathbf{w}^T \mathbf{S}_w \mathbf{w} = 1$$

等价于  $\mathbf{S}_b \mathbf{w} = \lambda \mathbf{S}_w \mathbf{w}$  的最大特征值。

# 线性判别分析

线性判别分析(LDA):

$$S_b w = \lambda S_w w$$

常规方法是求矩阵  $S_w^{-1} S_b$  的特征值问题!

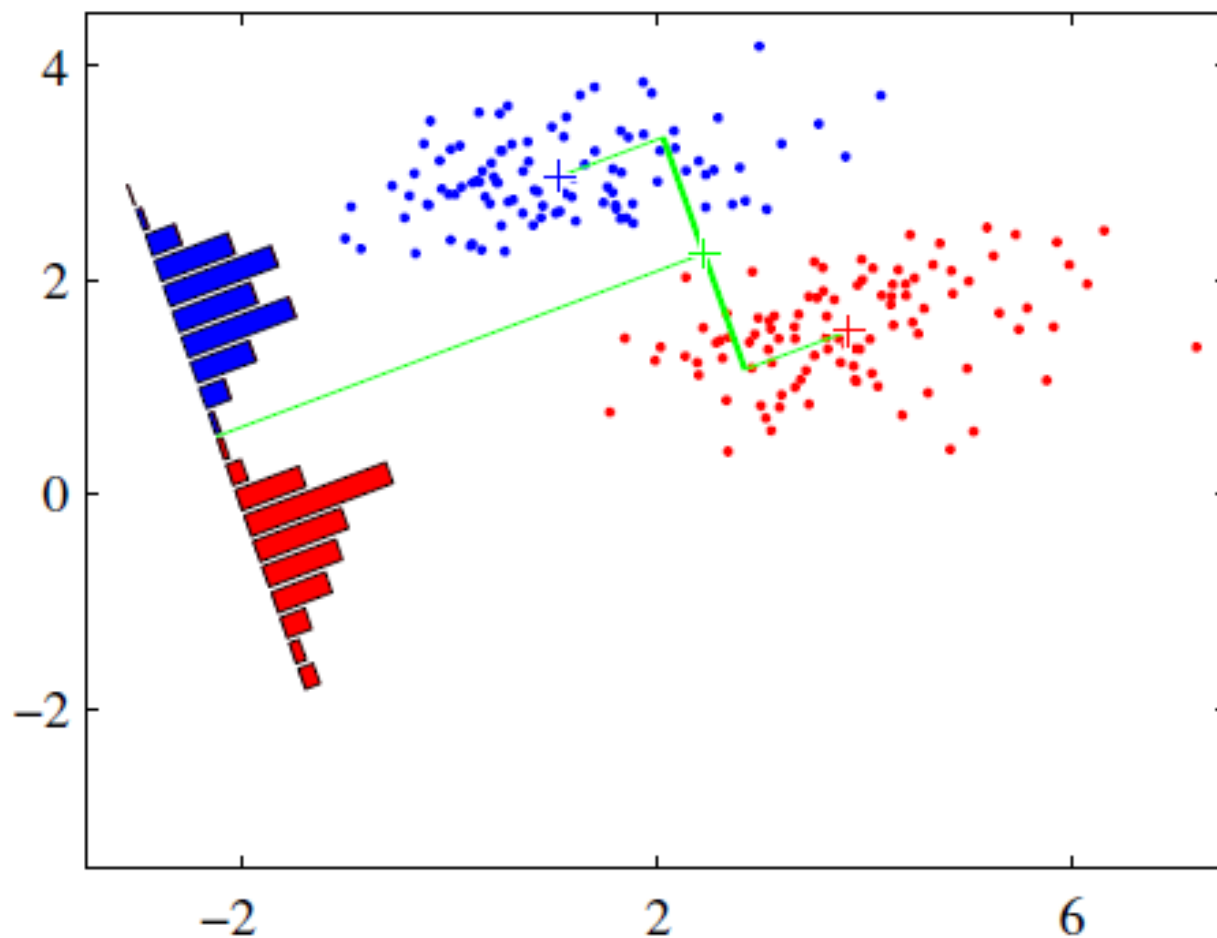
然而  $S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T$ , 我们有

$$\begin{aligned} S_b w &= (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T w \\ &= ((\mu_0 - \mu_1)^T w)(\mu_0 - \mu_1) \end{aligned}$$

原问题等价于求解:

$$S_w w = \mu_0 - \mu_1$$

# LDA降维示意图



# LDA vs PCA

---

相同点：

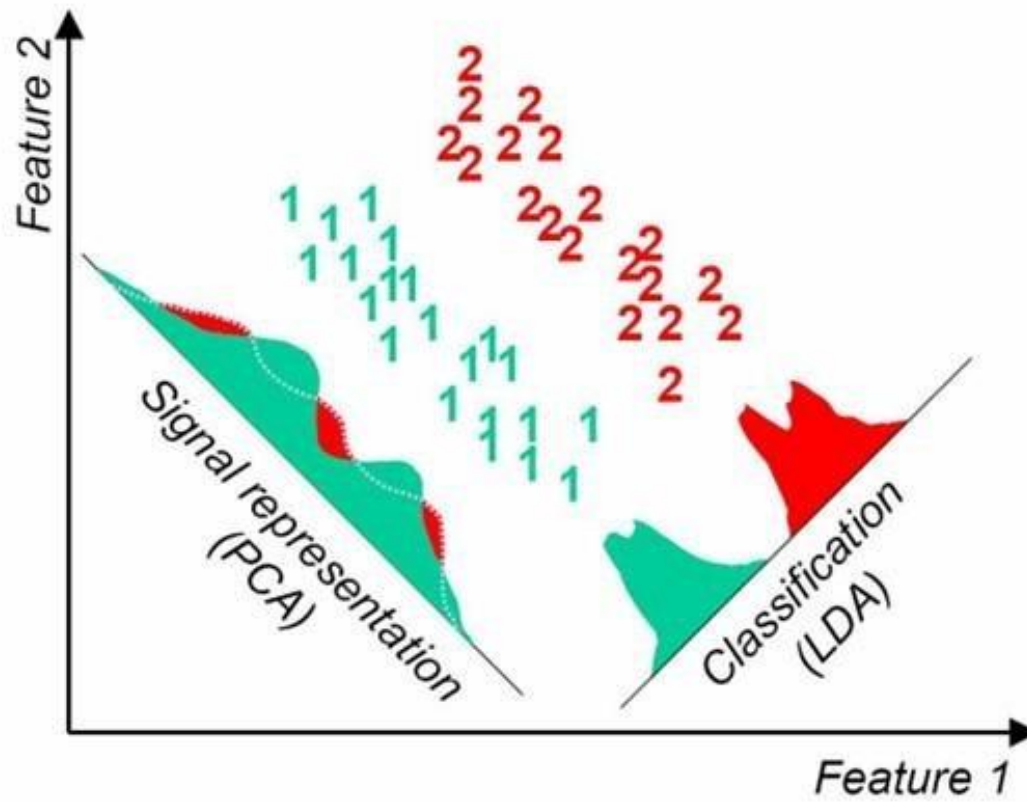
- 都属于降维方法。
- 都转化为求解特征值问题。
- 都可以使用核化技巧。

不同点：

- PCA是非监督降维，LDA是监督降维。



# LDA vs PCA



# LDA --最小二乘法

当  $x \in X_i$  时, 令  $y = (-1)^i n/|D_i|$ , 则

$$\sum_{i=1}^n y_i = |D_0| \times n/|D_0| - |D_1| \times n/|D_1| = 0$$

最小二乘法:

$$\operatorname{argmin}_{w,b} \frac{1}{n} \sum_i (y_i - (w^T x_i + b))^2$$

# LDA --最小二乘法

目标函数对  $w$  和  $b$  求偏导得,

$$\frac{1}{n} \sum_i (w^T x_i + b - y_i) = 0, \quad \sum_i (w^T x_i + b - y_i) x_i = 0$$

对第一个式子展开,

$$b = -\frac{1}{n} \sum_i w^T x_i = -w^T \mu = -\frac{1}{n} (|D_0| \mu_0 + |D_1| \mu_1)$$

把  $b$  代入第二个式子, 并展开有

$$(S_w + \frac{|D_0| |D_1|}{n}) w = n(\mu_0 - \mu_1)$$

---

## 6.5 多分类LDA

陈飞宇

[fchen@cqu.edu.cn](mailto:fchen@cqu.edu.cn)

办公室：软件学院529

# 多分类LDA

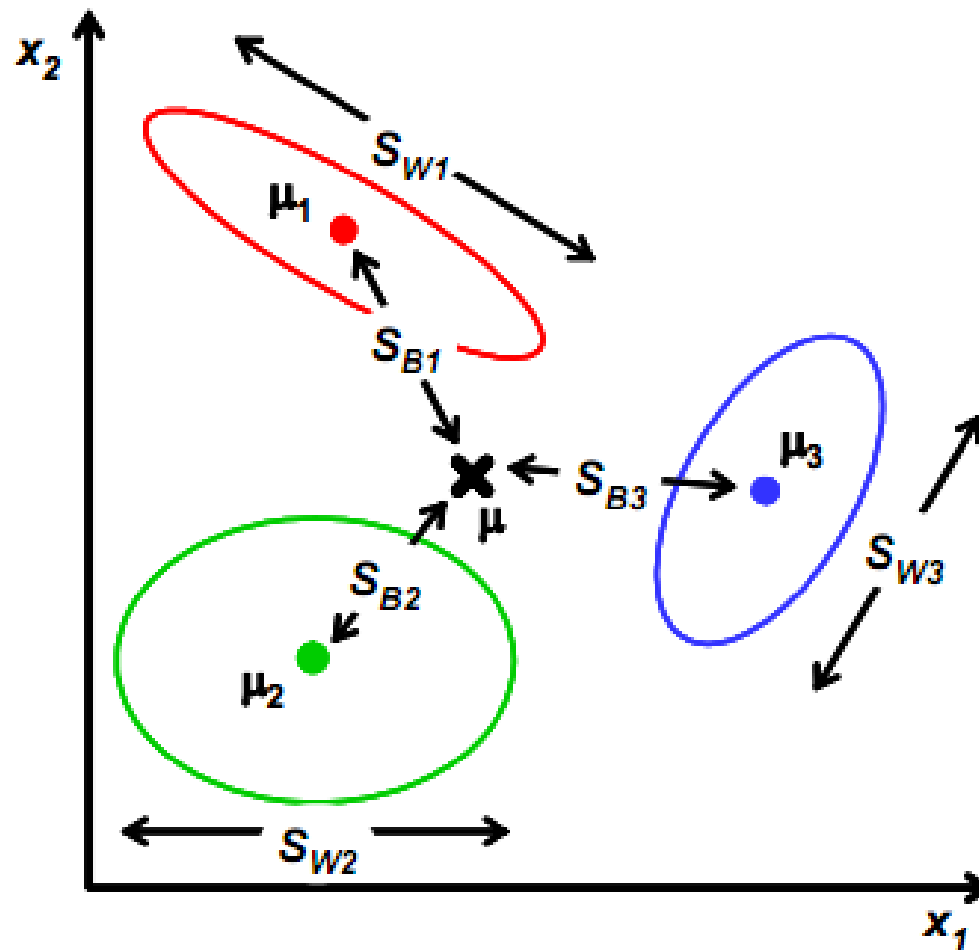
假设存在 $k$ 个类，令 $D_j, |D_j|, \mu_j, \Sigma_j$ 分别表示第 $j$ 类样本的集合、均值向量和协方差矩阵。

类内散度矩阵（within-class scatter matrix）：

$$S_w = \sum_{j=1}^k S_{wj} = \sum_{j=1}^k \sum_{x \in D_j} (x - \mu_j)(x - \mu_j)^T$$

中心变成 $k$ 个之后，如何定义类间散度矩阵？

# 多分类LDA



# 多分类LDA

类间散度矩阵 (within-class scatter matrix) :

$$S_b = \sum_{j=1}^k |D_j| (\mu - \mu_j)(\mu - \mu_j)^T$$

全局散度矩阵 (total scatter matrix) :

$$S_t = S_w + S_b = \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T$$

# 多分类LDA

多分类LDA需要求解投影矩阵 $W$ ，同时最大化类间散度矩阵，最小化类内散度矩阵。

$$\max_W \frac{\text{tr}(W^T S_b W)}{\text{tr}(W^T S_w W)} \quad \text{s. t.} \quad \mathbf{W}^T \mathbf{W} = \mathbf{I}$$

其中， $\text{tr}$ 函数表示矩阵主对角线上的元素之和，即  $\text{tr}(A) = \sum_{i=1}^n A_{ii}$ ，正交约束是因为每个投影方向互相垂直！

此模型的最优解为矩阵 $S_w^{-1} S_b$ 的前 $d$ 个最大特征值所对应的特征向量组成的矩阵。



---

## 6.6 非负矩阵分解

陈飞宇

[fchen@cqu.edu.cn](mailto:fchen@cqu.edu.cn)

办公室：软件学院529

# PCA和LDA的局限性

PCA和LDA的投影矩阵都是正交的，投影之后结果很可能出现负数，不可解释！

PCA



×



=



# 非负矩阵分解

非负矩阵分解模型（NMF）：

$$\min \quad \|X - WH\|_F^2$$

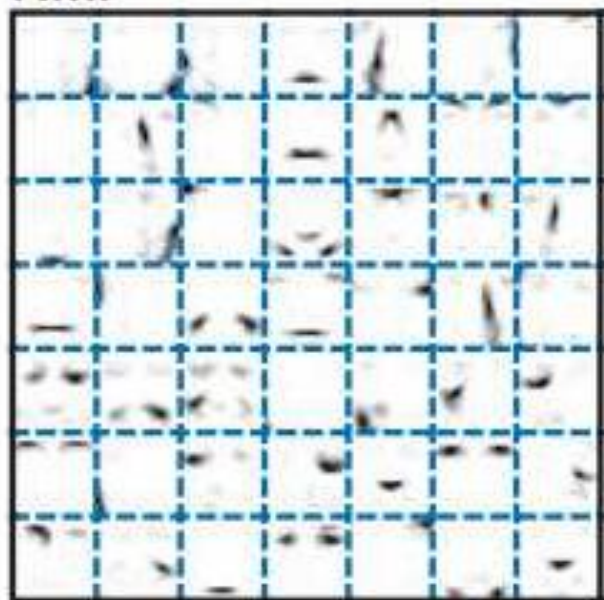
$$\text{s. t. } W \geq 0, H \geq 0$$

- 把数据矩阵  $X$  分解成两个小规模矩阵的乘积，本质上说是寻找数据  $X$  的低秩逼近。
- 基矩阵和系数矩阵的非负约束使得模型更有意义。
- NMF是非凸的模型，目标函数是非凸的。

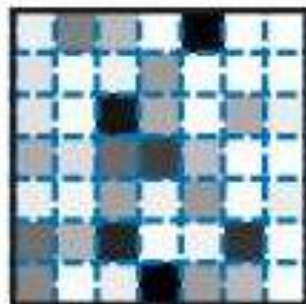
# 非负矩阵分解

非负矩阵分解模型（NMF）：

NMF

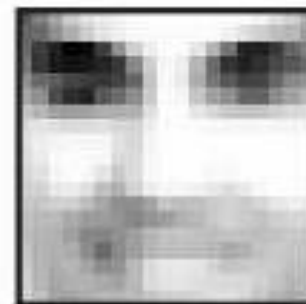
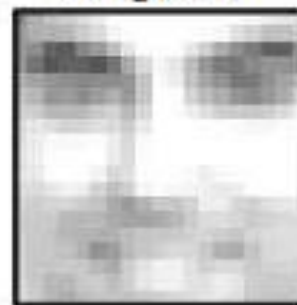


×



=

Original



# 交替最小二乘法

非负矩阵分解模型（NMF）：

$$\min \quad \|X - WH\|_F^2$$

$$\text{s. t.} \quad W \geq 0, H \geq 0$$

迭代公式：

$$W^+ = \max(0, (W^T W)^{-1} W^T X)$$

$$H^+ = \max(0, X H^T (H H^T)^{-1})$$

# 乘积更新算法

非负矩阵分解模型（NMF）：

$$\min \quad \|X - WH\|_F^2$$

$$\text{s. t. } W \geq 0, H \geq 0$$

迭代公式：

$$W^+ = W \odot [XH^T \oslash (WHH^T + \varepsilon)]$$

$$H^+ = H \odot [W^T X \oslash (W^T W H + \varepsilon)]$$