第8章 成分分析

刘家锋

哈尔滨工业大学

第8章　成分分析

8.0 引言和基础知识

引言

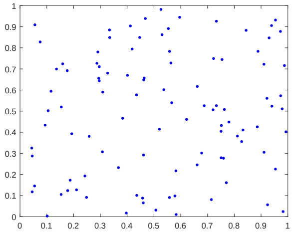
- **特征映射**
  - 在分类器设计过程中，经常需要对输入的特征做某种变换
  - 将输入的d维特征矢量$\mathbf{x}$映射为新的d'维矢量$\mathbf{y}$

$$\mathbf{y} = \Phi(\mathbf{x}), \qquad R^d \rightarrow R^{d'}$$

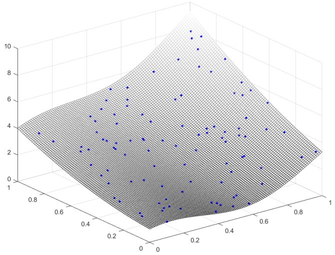
- **成分分析：降维($d' < d$)**
  - 去除原始特征中的冗余信息
  - 降低分类器的复杂度，提高泛化能力

特征映射

- 升维
  - 连续映射Φ将输入$\mathbf{x}$映射到高维空间的一个曲面(流形)上
  - 在高维空间中，可以实现样本的线性分类



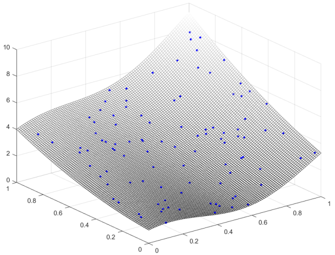
输入$\mathbf{x} \in R^d$



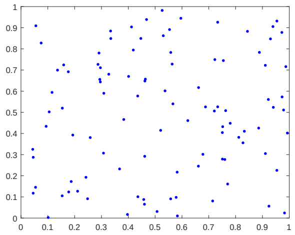
$\mathbf{y} = \Phi(\mathbf{x}) \in R^{d'}$

特征映射

- **降维(流形学习)**
  - 输入数据采样于嵌入在高维空间的低维流形
  - 在流形上建立坐标系，以低维矢量表示输入数据



输入$\mathbf{x} \in R^d$



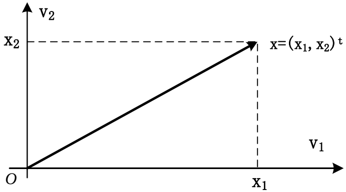
$\mathbf{y} = \Phi(\mathbf{x}) \in R^{d'}$

矢量与坐标系

- **坐标系**
  - 坐标系由原点O和一组基矢量$\{\mathbf{v}_1, \cdots, \mathbf{v}_d\}$构成
  - 基矢量一般是标准正交的

$$\mathbf{v}_i^t \mathbf{v}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad \Rightarrow \quad \|\mathbf{v}_i\| = 1, \mathbf{v}_i \perp \mathbf{v}_j$$

  - 给定坐标系下，矢量可以用坐标表示：$\mathbf{x} = (x_1, \cdots, x_d)^t$



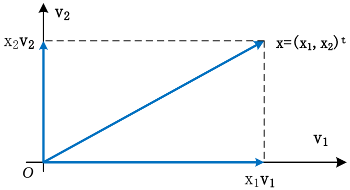
矢量与坐标系

- 矢量
  - 矢量$\mathbf{x}$可以表示为基矢量的线性组合

$$\mathbf{x} = \sum_{i=1}^{d} x_i \mathbf{v}_i$$

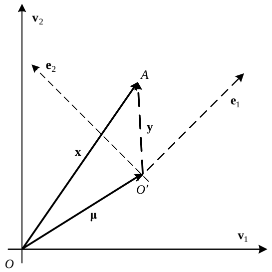
  - 由基矢量的标准正交性

$$\mathbf{x}^t \mathbf{v}_j = \sum_{i=1}^{d} x_i \mathbf{v}_i^t \mathbf{v}_j = x_j$$



坐标变换

- **坐标系1**
  - 基矢量$\{\mathbf{v}_1, \cdots, \mathbf{v}_d\}$，原点$O$
  - A点对应矢量：$\mathbf{x} = (x_1, \cdots, x_d)^t$
- **坐标系2**
  - 基矢量$\{\mathbf{e}_1, \cdots, \mathbf{e}_d\}$，原点$O'$在坐标系1的对应矢量$\boldsymbol{\mu}$
  - A点对应矢量：$\mathbf{y} = (y_1, \cdots, y_d)^t$



坐标变换

- **在坐标系1下**
  - 矢量之间的关系

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{y} = \boldsymbol{\mu} + \sum_{i=1}^{d} y_i \mathbf{e}_i$$

- **在坐标系2下**
  - 计算矢量的坐标

$$\mathbf{y} = \mathbf{x} - \boldsymbol{\mu} \quad \Rightarrow \quad y_i = \mathbf{e}_i^t(\mathbf{x} - \boldsymbol{\mu})$$

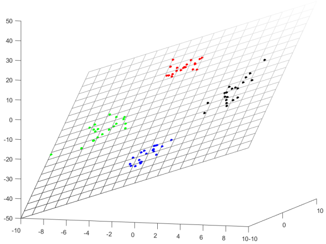
  - 以$\{\mathbf{e}_1, \cdots, \mathbf{e}_d\}$在坐标系1下的坐标为列矢量表示变换矩阵$E = (\mathbf{e}_1, \cdots, \mathbf{e}_d)$
  - 矢量$\mathbf{y}$在坐标系2下的坐标

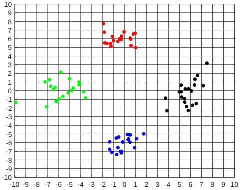
$$\mathbf{y} = E^t(\mathbf{x} - \boldsymbol{\mu})$$

8.1 主成分分析

PCA：Principal Component Analysis

- **主成分分析**
  - 主成分分析是一种最经典的线性降维方法
  - 基本假设是高维数据分布在嵌入的低维线性流形上
  - 目标是要找到一个线性映射，在低维空间表示高维数据

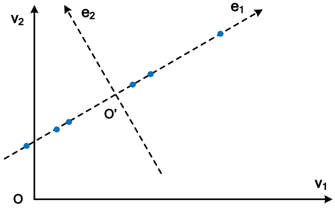




$$\mathbf{x} \in R^3 \quad \overset{PCA}{\Longrightarrow} \quad \mathbf{y} \in R^2$$

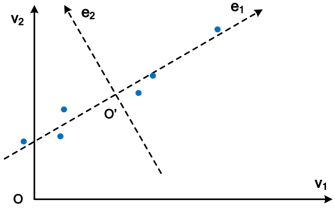
主成分分析

- **PCA的目标**
  - 建立新的坐标系，用更少的坐标重新表示数据
  - 理想情况：新的坐标表示可以完美地恢复数据



主成分分析

- **PCA的目标**
  - 建立新的坐标系，用更少的坐标重新表示数据
  - 理想情况：新的坐标表示可以完美地恢复数据
  - 噪声情况：恢复数据的误差(损失)最小



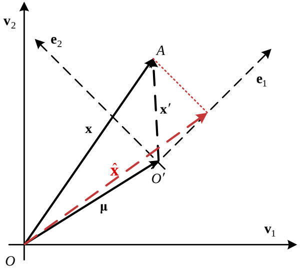
主成分分析

- 矢量的坐标表示
  - A点在$\{\mathbf{e}_i\}$坐标系下对应矢量：$\mathbf{x}' = (a_1, \cdots, a_d)^t$

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{x}' = \boldsymbol{\mu} + \sum_{i=1}^{d} a_i \mathbf{e}_i$$

  - 只保留$d' < d$个特征，恢复原坐标系矢量存在误差

$$\hat{\mathbf{x}} = \boldsymbol{\mu} + \sum_{i=1}^{d'} a_i \mathbf{e}_i \quad \Rightarrow \quad \mathbf{x} - \hat{\mathbf{x}} = \sum_{i=d'+1}^{d} a_i \mathbf{e}_i$$



主成分分析

- **PCA的优化目标**
  - 样本集$D = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}, \mathbf{x}_k \in R^d$
  - 建立一个新的坐标系，以样本均值$\boldsymbol{\mu}$为原点，$\{\mathbf{e}_1, \cdots, \mathbf{e}_d\}$为基矢量
  - 新的坐标系下只用前d'个特征表示样本
  - 恢复原坐标系矢量$\{\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_n\}$的平均误差最小

$$\min_{\mathbf{e}_1, \cdots, \mathbf{e}_d} J(\mathbf{e}_1, \cdots, \mathbf{e}_d) = \frac{1}{n} \sum_{k=1}^{n} \|\mathbf{x}_k - \hat{\mathbf{x}}_k\|^2$$

subject to

$$\mathbf{e}_i^t \mathbf{e}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

PCA的优化

展开优化目标函数

$$
\begin{aligned}
J(\mathbf{e}_1, \cdots, \mathbf{e}_d) &= \frac{1}{n}\sum_{k=1}^{n}\left\|\mathbf{x}_k - \hat{\mathbf{x}}_k\right\|^2 \\
&= \frac{1}{n}\sum_{k=1}^{n}\left\|\sum_{i=d'+1}^{d} a_{ki}\mathbf{e}_i\right\|^2 \\
&= \frac{1}{n}\sum_{k=1}^{n}\left(\sum_{i=d'+1}^{d} a_{ki}\mathbf{e}_i\right)^t\left(\sum_{i=d'+1}^{d} a_{ki}\mathbf{e}_i\right) \\
&= \frac{1}{n}\sum_{k=1}^{n}\sum_{i=d'+1}^{d} a_{ki}^2 \quad \text{(基矢量的单位正交性)}
\end{aligned}
$$

PCA的优化

$\mathbf{x}_k$在新坐标系的第i个坐标$a_{ki} = (\mathbf{x}_k - \boldsymbol{\mu})^t \mathbf{e}_i$为标量，因此：

$$
\begin{aligned}
J(\mathbf{e}_1, \cdots, \mathbf{e}_d) &= \frac{1}{n} \sum_{k=1}^{n} \sum_{i=d'+1}^{d} a_{ki}^2 \\
&= \frac{1}{n} \sum_{k=1}^{n} \sum_{i=d'+1}^{d} [(\mathbf{x}_k - \boldsymbol{\mu})^t \mathbf{e}_i]^t [(\mathbf{x}_k - \boldsymbol{\mu})^t \mathbf{e}_i] \\
&= \frac{1}{n} \sum_{k=1}^{n} \sum_{i=d'+1}^{d} [\mathbf{e}_i^t (\mathbf{x}_k - \boldsymbol{\mu})][(\mathbf{x}_k - \boldsymbol{\mu})^t \mathbf{e}_i] \\
&= \sum_{i=d'+1}^{d} \mathbf{e}_i^t \left[\frac{1}{n} \sum_{k=1}^{n} (\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^t \right] \mathbf{e}_i
\end{aligned}
$$

PCA的优化

- **PCA的优化问题**
  - 定义样本集D的协方差矩阵

$$S = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^t$$

  - 则PCA的优化问题可以表示为

$$\min_{\mathbf{e}_1,\cdots,\mathbf{e}_d} J(\mathbf{e}_1,\cdots,\mathbf{e}_d) = \sum_{i=d'+1}^{d} \mathbf{e}_i^t S \mathbf{e}_i$$

subject to

$$\mathbf{e}_i^t \mathbf{e}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

PCA的优化

- **Lagrange函数**
  - 构造Lagrange函数（暂时忽略正交约束）

$$L(\mathbf{e}_1,\cdots,\mathbf{e}_d,\boldsymbol{\lambda}) = \sum_{i=d'+1}^{d} \mathbf{e}_i^t S \mathbf{e}_i - \sum_{i=1}^{d} \lambda_i (\mathbf{e}_i^t \mathbf{e}_i - 1)$$

    其中，$\lambda_1,\cdots,\lambda_d$为Lagrange系数
  - 计算Lagrange函数的极值点

$$\frac{\partial L(\mathbf{e}_1,\cdots,\mathbf{e}_d,\boldsymbol{\lambda})}{\partial \mathbf{e}_j} = 2S\mathbf{e}_j - 2\lambda_j \mathbf{e}_j = \mathbf{0}$$

    得到：

$$S\mathbf{e}_j = \lambda_j \mathbf{e}_j, \qquad j = 1,\cdots,d$$

PCA的优化

- **基矢量的求解**
  - 基矢量$\mathbf{e}_i$是S的特征矢量，Lagrange系数λ_i是相应的特征值
  - 协方差矩阵S为$d \times d$实对称矩阵，有d个特征值和特征矢量
  - 正交性：实对称矩阵的特征矢量之间相互正交
- **基矢量的选择**
  - 将$S\mathbf{e}_i = \lambda_i \mathbf{e}_i$代入目标函数

$$J(\mathbf{e}_1, \cdots, \mathbf{e}_d) = \sum_{i=d'+1}^{d} \mathbf{e}_i^t S \mathbf{e}_i = \sum_{i=d'+1}^{d} \lambda_i \mathbf{e}_i^t \mathbf{e}_i = \sum_{i=d'+1}^{d} \lambda_i$$

  - 矢量恢复的平方误差由丢弃的基矢量对应的特征值决定
  - 新坐标的基矢量$\{\mathbf{e}_1, \cdots, \mathbf{e}_{d'}\}$，应该选择协方差矩阵$S$最大的$d'$个特征值对应的特征矢量

PCA算法

Algorithm 1 PCA: Principal Component Analysis

Input: 样本集$D = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}, \mathbf{x}_i \in R^d$

Output: 降维样本集$\{\mathbf{y}_1, \cdots, \mathbf{y}_n\}, \mathbf{y}_i \in R^{d'}$

1: 计算样本集D的均值$\boldsymbol{\mu}$和协方差矩阵S

2: 计算矩阵S的特征值，并由大到小排序

3: 选择前d'个特征值对应特征矢量作为列矢量，构造变换矩阵$E = (\mathbf{e}_1, \cdots, \mathbf{e}_{d'})$

4: 计算降维样本集

$$\mathbf{y}_i = E^t(\mathbf{x}_i - \boldsymbol{\mu}), \quad i = 1, \cdots, n$$

PCA的讨论

- **不相关性**
  - 在新坐标系下特征之间是不相关的，协方差矩阵为对角阵
  - 证明：

$$\boldsymbol{\mu}_{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i = \frac{1}{n}\sum_{i=1}^{n}E^t(\mathbf{x}_i - \boldsymbol{\mu}) = E^t\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i - \boldsymbol{\mu}\right) = \mathbf{0}$$

$$\begin{aligned} S_y &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i\mathbf{y}_i^t = \frac{1}{n}\sum_{i=1}^{n}E^t(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^t E \\ &= E^t S E = \begin{pmatrix} \mathbf{e}_1^t \\ \vdots \\ \mathbf{e}_{d'}^t \end{pmatrix} \times S \times (\mathbf{e}_1, \cdots, \mathbf{e}_{d'}) \\ &= \begin{pmatrix} \mathbf{e}_1^t S\mathbf{e}_1 & \cdots & \mathbf{e}_1^t S\mathbf{e}_{d'} \\ \vdots & \ddots & \vdots \\ \mathbf{e}_{d'}^t S\mathbf{e}_1 & \cdots & \mathbf{e}_{d'}^t S\mathbf{e}_{d'} \end{pmatrix} \end{aligned}$$

PCA的讨论

证明(续):

由于

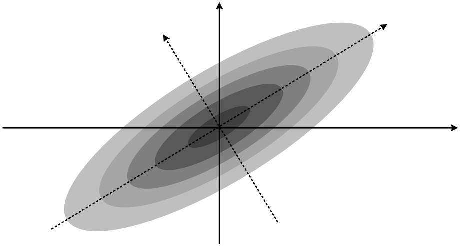
$$\mathbf{e}_i^t S \mathbf{e}_j = \lambda_j \mathbf{e}_i^t \mathbf{e}_j = \begin{cases} \lambda_j, & i = j \\ 0, & i \neq j \end{cases}$$

因此

$$S_y = \begin{pmatrix} \mathbf{e}_1^t S \mathbf{e}_1 & \cdots & \mathbf{e}_1^t S \mathbf{e}_{d'} \\ \vdots & \ddots & \vdots \\ \mathbf{e}_{d'}^t S \mathbf{e}_1 & \cdots & \mathbf{e}_{d'}^t S \mathbf{e}_{d'} \end{pmatrix} = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_{d'} \end{pmatrix}$$

PCA的讨论

- **特征值、特征矢量的含义**
  - 特征值：对应变换之后各维特征分布的方差
  - 特征矢量：对应分布的主轴方向



例8.1 PCA降维

有训练集样本：

$$D=\left\{\begin{pmatrix}-5\\-4\end{pmatrix},\begin{pmatrix}-4\\-5\end{pmatrix},\begin{pmatrix}-5\\-6\end{pmatrix},\begin{pmatrix}-6\\-5\end{pmatrix},\begin{pmatrix}5\\4\end{pmatrix},\begin{pmatrix}4\\5\end{pmatrix},\begin{pmatrix}5\\6\end{pmatrix},\begin{pmatrix}6\\5\end{pmatrix}\right\}$$

使用PCA方法将2维特征降为1维特征。

例8.1 PCA降维

计算样本集的均值：

$$\begin{aligned}\boldsymbol{\mu} &= \frac{1}{8}\left\{\begin{pmatrix}-5\\-4\end{pmatrix}+\begin{pmatrix}-4\\-5\end{pmatrix}+\begin{pmatrix}-5\\-6\end{pmatrix}+\begin{pmatrix}-6\\-5\end{pmatrix}+\begin{pmatrix}5\\4\end{pmatrix}+\begin{pmatrix}4\\5\end{pmatrix}+\begin{pmatrix}5\\6\end{pmatrix}+\begin{pmatrix}6\\5\end{pmatrix}\right\}\\ &= \begin{pmatrix}0\\0\end{pmatrix}\end{aligned}$$

计算协方差矩阵：

$$\begin{aligned}S &= \frac{1}{8}\sum_{i=1}^{8}(\mathbf{x}_k-\boldsymbol{\mu})(\mathbf{x}_k-\boldsymbol{\mu})^t\\ &= \frac{1}{8}\left\{\begin{pmatrix}25 & 20\\20 & 16\end{pmatrix}+\begin{pmatrix}16 & 20\\20 & 25\end{pmatrix}+\begin{pmatrix}25 & 30\\30 & 36\end{pmatrix}+\begin{pmatrix}36 & 30\\30 & 25\end{pmatrix}\right.\\ &\quad\left.+\begin{pmatrix}25 & 20\\20 & 16\end{pmatrix}+\begin{pmatrix}16 & 20\\20 & 25\end{pmatrix}+\begin{pmatrix}25 & 30\\30 & 36\end{pmatrix}+\begin{pmatrix}36 & 30\\30 & 25\end{pmatrix}\right\}\\ &= \begin{pmatrix}25.5 & 25\\25 & 25.5\end{pmatrix}\end{aligned}$$

例8.1 PCA降维

协方差矩阵的特征值方程：

$$S\mathbf{e} = \lambda\mathbf{e} \qquad \Rightarrow \qquad (S - \lambda I)\mathbf{e} = \mathbf{0}$$

其中I为单位矩阵，特征值方程为齐次线性方程组，存在非0解的条件是系数矩阵的行列式值为0：

$$|S - \lambda I| = \begin{vmatrix} 25.5 - \lambda & 25 \\ 25 & 25.5 - \lambda \end{vmatrix} = 0$$

得到：

$$(25.5 - \lambda)^2 - 25^2 = 0 \qquad \Rightarrow \qquad 25.5 - \lambda = \pm 25$$

协方差矩阵的两个特征值为：

$$\lambda_1 = 50.5, \qquad \lambda_2 = 0.5$$

例8.1 PCA降维

选择最大特征值，计算对应的特征向量：

$$S\mathbf{e} = \lambda_1 \mathbf{e}$$

即：

$$\begin{pmatrix} 25.5 & 25 \\ 25 & 25.5 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = 50.5 \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}$$

可以得到：

$$25.5e_1 + 25e_2 = 50.5e_1 \qquad \Rightarrow \qquad e_1 = e_2$$

令$e_1 = e_2 = 1$，得到λ_1对应的特征向量$\mathbf{e}_1$，并归一化向量长度：

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \Rightarrow \qquad \mathbf{e}_1 = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right)^t$$

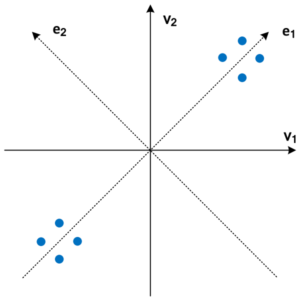
同理可得λ_2对应的特征向量$\mathbf{e}_2$：

$$\mathbf{e}_2 = \left(\frac{-\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right)^t$$

例8.1 PCA降维

样本集及降维前后的坐标系如右图，计算样本在坐标轴$\mathbf{e}_1$上的投影：

$$y_1 = \mathbf{e}_1^t(\mathbf{x}_1 - \boldsymbol{\mu}) = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)\begin{pmatrix}-5\\-4\end{pmatrix} = -\frac{9}{2}\sqrt{2}$$



同理得到：

$$y_2 = -\frac{9}{2}\sqrt{2}, \quad y_3 = -\frac{11}{2}\sqrt{2}$$
$$y_4 = -\frac{11}{2}\sqrt{2}$$
$$y_5 = \frac{9}{2}\sqrt{2}, \quad y_6 = \frac{9}{2}\sqrt{2}$$
$$y_7 = \frac{11}{2}\sqrt{2}, \quad y_8 = \frac{11}{2}\sqrt{2}$$

例8.1 代码

```python
import numpy as np
from sklearn.decomposition import PCA

X = np.array([[-5,-4],[-4,-5],[-5,-6],[-6,-5],
              [+5,+4],[+4,+5],[+5,+6],[+6,+5]])

pca = PCA(n_components=2).fit(X)
n = pca.n_samples_

print("Mean of samples:", pca.mean_)
print("Eigen Vectors:\n", pca.components_)
print("Eigen Values of unbiased:", pca.explained_variance_)
print("Eigen Values of biased:", pca.explained_variance_*(n-1)/n)
```

```
Mean of samples: [0. 0.]
Eigen Vectors:
 [[-0.70710678 -0.70710678]
 [-0.70710678  0.70710678]]
Eigen Values of unbiased: [57.71428571  0.57142857]
Eigen Values of biased: [50.5  0.5]
```

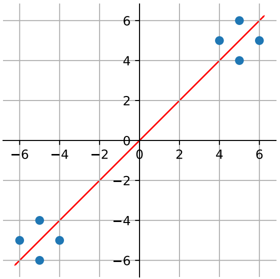
例8.1 代码

```python
import matplotlib.pyplot as plt

X_pca = pca.transform(X)
print("Transformed Samples:\n", X_pca)

plt.plot(X[:,0],X[:,1],'o')
'''省略显示部分'''
plt.show()
```

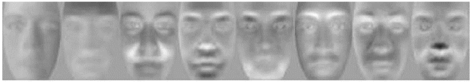
```
Transformed Samples:
 [[ 6.36396103  0.70710678]
 [ 6.36396103 -0.70710678]
 [ 7.77817459 -0.70710678]
 [ 7.77817459  0.70710678]
 [-6.36396103 -0.70710678]
 [-6.36396103  0.70710678]
 [-7.77817459  0.70710678]
 [-7.77817459 -0.70710678]]
```



特征人脸(EigenFace)

- **人脸图像的PCA分解**

人脸图像库



基矢量（EigenFaces）

例8.2 PCA分解与重构

```python
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA

data = pd.read_csv("MNIST_test.csv")
X_test = data.iloc[:,1:785].to_numpy()
y_test = data.iloc[:,0].to_numpy()

pca = PCA(n_components=200).fit(X_test)
X_pca = pca.transform(X_test)

print("Original Shape:", X_test.shape)
print("Transformed Shape:", X_pca.shape)
```

```
Original Shape: (10000, 784)
Transformed Shape: (10000, 200)
```

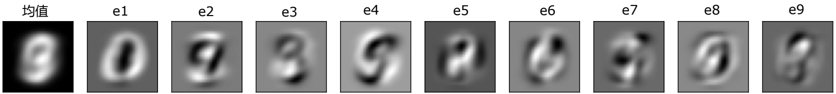
MINST数据PCA均值、特征向量

```
import matplotlib.pyplot as plt

fix, axes = plt.subplots(1,10,figsize=(16,6),subplot_kw={'xticks':(),'yticks':()})
mu = pca.mean_; image = mu.reshape([28,28])
axes[0].imshow(image,cmap='gray')

for i, ax in zip(range(9),axes[1:].ravel()):
    e = pca.components_[i,:]
    image = e.reshape([28,28])
    ax.imshow(image,cmap='gray')

plt.plot()
```

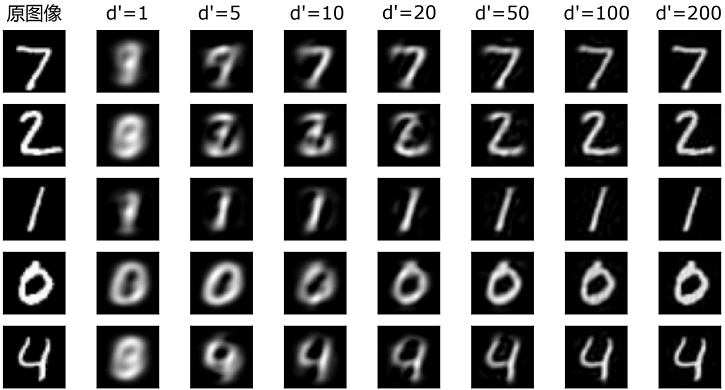


MINST数据PCA重构

```
fix, axes = plt.subplots(5,8,figsize=(10,5),subplot_kw={'xticks':(), 'yticks':()})
for i in range(5):
    x = X_test[i,:]; image = x.reshape([28,28])
    axes[i,0].imshow(image, cmap="gray")
    for n_components, ax in zip([1,5,10,20,50,100,200], axes[i,1:].ravel()):
        E = pca.components_[0:n_components,:]
        x_recons = np.matmul(X_pca[i,0:n_components], E) + pca.mean_
        x_recons[x_recons<0] = 0
        image = np.floor(x_recons.reshape([28,28]))
        ax.imshow(image,cmap='gray')

plt.show()
```

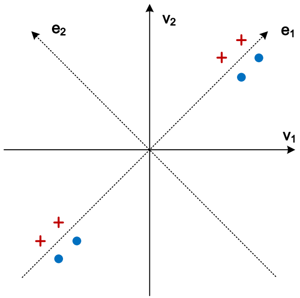
MINST数据PCA重构



8.2 线性判别分析

无监督的成分分析

- **PCA是无监督的成分分析**
  - 只考虑了样本集的整体分布，没有使用样本的类别信息
  - 丢弃的特征有可能恰恰包含了重要的可分性信息



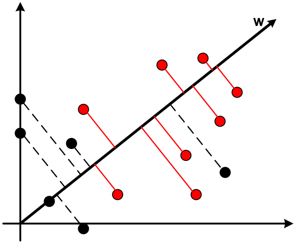
线性判别分析

- **LDA:Linear Discriminant Analysis**
  - LDA是有监督的成分分析方法，寻找可分性最大意义下的最优线性映射
  - 充分保留样本的类别可分性信息
  - 也称为FDA：Fisher Discriminant Analysis
- **Fisher线性判别准则**
  - 将c个类别的样本集$D_1, \cdots, D_c$向矢量$\mathbf{w}$的方向上投影
  - 使得投影之后
    - 同类别样本之间的距离近
    - 不同类别样本之间的距离远

LDA的推导

- **两类样本的投影**
  - 样本$\mathbf{x}$在矢量$\mathbf{w}$方向上的投影是一个标量：$y = \mathbf{w}^t\mathbf{x}$
  - 包含n_1和n_2个样本的两类别样本集D_1和D_2，在矢量$\mathbf{w}$方向上的投影为$\mathcal{Y}_1, \mathcal{Y}_2$
  - 投影前后样本集的均值

$$\boldsymbol{\mu}_i = \frac{1}{n_i}\sum_{\mathbf{x}\in D_i}\mathbf{x}, \qquad \tilde{\mu}_i = \frac{1}{n_i}\sum_{y\in\mathcal{Y}_i} y = \frac{1}{n_i}\sum_{\mathbf{x}\in D_i}\mathbf{w}^t\mathbf{x} = \mathbf{w}^t\boldsymbol{\mu}_i$$



LDA的推导

- **类内的散布**
  - 定义ω_i类样本投影之后的散布

$$\tilde{s}_i^2 = \sum_{y \in \mathcal{Y}_i} (y - \tilde{\mu}_i)^2$$
$$= \sum_{\mathbf{x} \in D_i} \left(\mathbf{w}^t \mathbf{x} - \mathbf{w}^t \boldsymbol{\mu}_i\right)^2$$
$$= \sum_{\mathbf{x} \in D_i} \mathbf{w}^t (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^t \mathbf{w} = \mathbf{w}^t S_i \mathbf{w}$$

  - 两类样本总的散布，描述了同类样本的分散程度

$$\tilde{s}_1^2 + \tilde{s}_2^2 = \mathbf{w}^t S_1 \mathbf{w} + \mathbf{w}^t S_2 \mathbf{w} = \mathbf{w}^t (S_1 + S_2) \mathbf{w} = \mathbf{w}^t S_w \mathbf{w}$$

  - 类内散布矩阵

$$S_w = S_1 + S_2 = \sum_{i=1}^{2} \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^t$$

LDA的推导

- **类间的散布**
  - 用两个类样本投影之后均值的差异，描述不同类样本的分散程度

$$\begin{aligned}(\tilde{\mu}_1 - \tilde{\mu}_2)^2 &= \left(\mathbf{w}^t \boldsymbol{\mu}_1 - \mathbf{w}^t \boldsymbol{\mu}_2\right)^2 \\ &= \mathbf{w}^t (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t \mathbf{w} \\ &= \mathbf{w}^t S_b \mathbf{w}\end{aligned}$$

  - 类间散布矩阵

$$S_b = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t$$

LDA的推导

- **Fisher准则**
  - 定义Fisher准则

$$J(\mathbf{w}) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{\mathbf{w}^t S_b \mathbf{w}}{\mathbf{w}^t S_w \mathbf{w}}$$

    $J(\mathbf{w})$为Rayleigh商，越大则样本集投影之后的可分性越强；
  - 简单地最大化$J(\mathbf{w})$是不适定的，因为如果$\mathbf{w}^*$为最优解的话，$\alpha\mathbf{w}^*$同样是最优解
  - 转化为约束优化问题

$$\max_{\mathbf{w}} \mathbf{w}^t S_b \mathbf{w}$$

subject to

$$\mathbf{w}^t S_w \mathbf{w} = C$$

C为任意常数

LDA的推导

- **优化求解**
  - 构造Lagrangre函数

$$L(\mathbf{w}, \lambda) = \mathbf{w}^t S_b \mathbf{w} - \lambda(\mathbf{w}^t S_w \mathbf{w} - C)$$

  - 计算极值点

$$\frac{\partial L(\mathbf{w}, \lambda)}{\partial \mathbf{w}} = 2S_b\mathbf{w} - 2\lambda S_w \mathbf{w} = \mathbf{0}$$

因此有：

$$S_b \mathbf{w} = \lambda S_w \mathbf{w}$$

LDA的推导

- **LDA的解**
  - Lagrange系数λ是S_b相对于S_w的广义特征值，$\mathbf{w}$是相应的广义特征矢量
  - 在S_w非奇异的条件下
$$S_w^{-1}S_b\mathbf{w}=\lambda\mathbf{w}$$
  - 将第i个广义特征值和特征矢量，代入Fisher准则函数
$$J(\mathbf{w}_i)=\frac{\mathbf{w}_i^tS_b\mathbf{w}_i}{\mathbf{w}_i^tS_w\mathbf{w}_i}=\frac{\lambda_i\mathbf{w}_i^tS_w\mathbf{w}_i}{\mathbf{w}_i^tS_w\mathbf{w}_i}=\lambda_i$$
  - 显然，LDA应该选择最大广义特征值对应的特征矢量作为最优的投影方向

线性判别分析

- **多类别的LDA**
  - c个类别的训练样本集：$D_1, \cdots, D_c$
  - 类内散布矩阵

$$S_w = \sum_{i=1}^{c} S_i = \sum_{i=1}^{c} \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^t$$

  - 类间散布矩阵

$$S_b = \sum_{i=1}^{c} n_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^t$$

其中，$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{c} \sum_{\mathbf{x} \in D_i} \mathbf{x}$，为所有样本的均值

LDA算法

Algorithm 2 LDA: Linear Discriminant Analysis

Input: c个类别的样本集$D_1, \cdots, D_c$

Output: 降维样本集$\mathcal{Y}_1, \cdots, \mathcal{Y}_c$

1: 计算类内散布矩阵S_w和类间散布矩阵S_b

2: 计算矩阵$S_w^{-1}S_b$的特征值和特征矢量

3: 选择非0的$c-1$个特征值对应特征矢量作为列矢量

4: 构造变换矩阵$W = (\mathbf{w}_1, \cdots, \mathbf{w}_{c-1})$

5: 计算降维样本集：

$$\mathbf{y} = W^t\mathbf{x}, \quad \mathbf{x} \in D_1, \cdots, D_c$$

LDA的讨论

- **LDA的特点**
  - 非正交：LDA的基矢量不构成一个正交的坐标系
  - 特征维数：新的坐标维数最多为$c-1$，c为类别数
  - 解的存在性：样本数多时，才能保证矩阵S_w是非奇异的
- **PCA与LDA的结合**
  - 对样本集$D = D_1 \cup \cdots \cup D_c$计算PCA
  - 在PCA的基矢量$\{\mathbf{w}_1, \cdots, \mathbf{w}_d\}$中，选择使得Fisher准则$J(\mathbf{w})$最大的一组基矢量

8.3 非线性的成分分析

核PCA

Algorithm 3 KPCA: Kernel Principal Component Analysis

Input: 样本集$D = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}, \mathbf{x}_i \in R^d$，核函数$k(\mathbf{x}, \mathbf{y})$

Output: 降维样本集$\{\mathbf{y}_1, \cdots, \mathbf{y}_n\}$

1: 计算$n \times n$维核矩阵K，元素$k_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = k(\mathbf{x}_i, \mathbf{x}_j)$

2: 计算K的特征值$\{\lambda_k\}$和特征矢量$\{\boldsymbol{\alpha}_k\}$，保留前$d'$个特征值

3: 特征空间中第k个基矢量

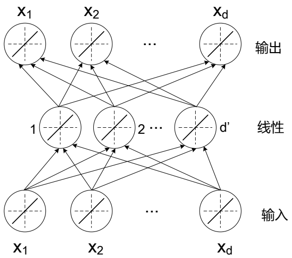
$$\mathbf{v}^{(k)} = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^{n} \alpha_i^{(k)} \phi(\mathbf{x}_i), \quad \boldsymbol{\alpha}_k = \left(\alpha_1^{(k)}, \cdots, \alpha_n^{(k)} \right)^t$$

4: 输入矢量$\mathbf{x}$在特征空间第k个基矢量上的投影

$$y_k = \left\langle \phi(\mathbf{x}), \mathbf{v}^{(k)} \right\rangle = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^{n} \alpha_i^{(k)} k(\mathbf{x}, \mathbf{x}_i)$$

AE：Auto-Encoder

- **PCA的神经网络实现**
  - 输入层：d个神经元
  - 隐含层：d'个神经元，线性激活函数
  - 输出层：d个神经元，线性激活函数

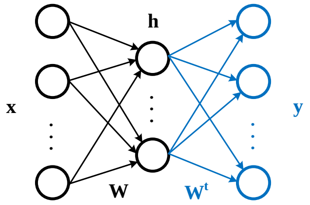


AE：Auto-Encoder

- **线性Auto-Encoder**
  - 隐含层与输出层的权值是相同的（互为转置，偏置不同），输出分别为

$$\mathbf{h} = W\mathbf{x} + \mathbf{b}, \quad \mathbf{y} = W^t(W\mathbf{x} + \mathbf{b}) + \mathbf{c}$$

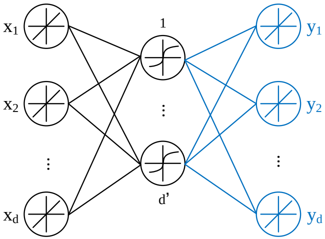
  - AE的学习：样本集$D = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$，既作为训练样本输入，也作为期望输出
  - 可以证明，隐含层神经元的权值为样本集协方差矩阵的前d'个特征矢量



AE：Auto-Encoder

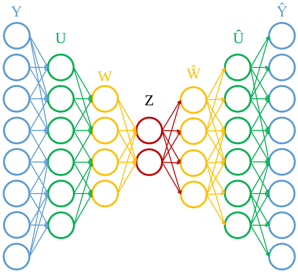
- **非线性AE**
  - 隐含层使用Sigmoid激活函数
  - 优化平方误差函数：

$$\min_{W,\mathbf{b},\mathbf{c}} J(W,\mathbf{b},\mathbf{c}) = \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{y}_i - \mathbf{x}_i\|^2$$



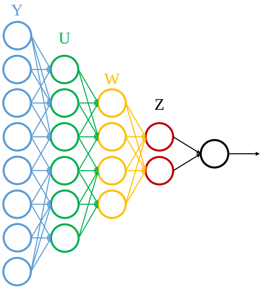
深度的自编码器

- **SAE：Stacked Auto-Encoder**
  - 增加隐含层的数量，可以提高网络的映射能力
  - 希望在中间的“瓶颈层”发现一些有意义的语义概念

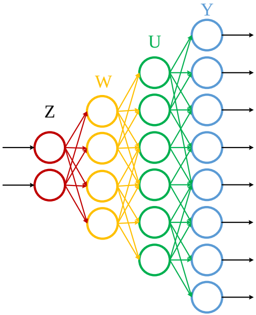


SAE的应用

- **构成判别网络**
  - 网络增加类别层
  - 模型化：$P(y|\mathbf{x})$

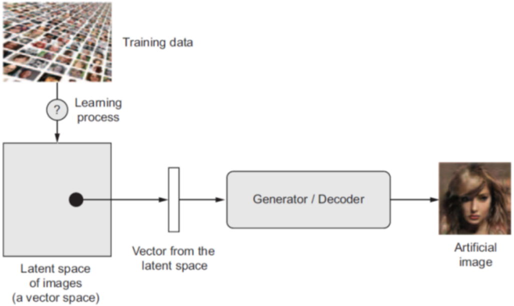


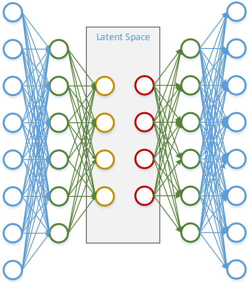
- **构成生成网络**
  - 网络以噪声输入
  - 生成分布$p(\mathbf{x})$的样本



SAE的应用

- **SAE生成人脸**
  - 训练样本集学习SAE
  - 在低维的Latent Space上抽样，生成输出





SAE的应用

- **图像的语义概念编辑**
  - 在Latent Space上，不同维度代表不同的语义概念
  - 例如，某个神经元的输出代表Smile
  - 输入人脸图像映射到Latent Space
  - 编辑Latent Space相应的维度，生成输出人脸图像