机器学习

第 7 章非监督学习-聚类和降维

欧阳毅

浙江工商大学
管理工程与电子商务学院

2023

目录

1 聚类
- EM-KMeans
- 混合高斯模型 GMM
- 谱聚类

2 降维处理
- 主成分分析

KMeans

1. 确定参数 K.
2. 初始化 K 个聚类中心（初始化可用随机方式分派）.
3. 通过比较数据与质心的距离，决定 N 个数据的类标签

$$\hat{z}_k = \arg\min_j ||x^{(i)} - \mu_j||^2$$

4. 对分配相同标签数据，重新估计 K 个聚类中心.

$$\hat{\mu}_k = \frac{1}{C_k} \sum_{i \in C_k} x_i$$

if N 个数据在最后一次迭代未变化 **then**
  退出
else
  返回第 3 步.
end if

```python
def kmeans(data,k):
    rarray = np.random.randint(len(data),size=k)
    center = data[rarray]
    n=len(data)
    z = np.zeros((n), np.int)
    run = True
    while run:
        for i in range(n):
            dist = np.square(data[i] - center)
            dist = np.sum(dist, axis=1)
            z[i] = np.argmin(dist)
        run = False
        for i in range(k):
            cluster = data[z==i]
            newcenter = np.mean(cluster, axis=0)
            ss = np.abs(center[i]-newcenter)
            if np.sum(ss, axis=0) > 1e-4:
                center[i] = newcenter
                run = True
        print('new center=', center)
```

KMeans

1. 确定参数 K.
2. 初始化 K 个聚类中心（初始化可用随机方式分派）.
3. 通过比较数据与质心的距离，决定 N 个数据的类标签

$$\hat{z}_k = \arg\min_j ||x^{(i)} - \mu_j||^2$$

4. 对分配相同标签数据，重新估计 K 个聚类中心.

$$\hat{\mu}_k = \frac{1}{C_k} \sum_{i \in C_k} x_i$$

if N 个数据在最后一次迭代未变化 **then**
  退出
else
  返回第 3 步.
end if

KMeans

$$z_n^{(t)} = \arg\min_k (x - \mu_k^{(t)})^T \Sigma_k^{-1(t)} (x_n - \mu_k^{(t)})$$

$$\hat{\mu}_t = \frac{1}{C_k} \sum_{i \in C_k} x_i$$

- KMeans 是 EM 算法的一种特例，对于样本所属类，属于硬分派 (hard).
- 即均值 μ_k 为每轮按照概率最大的被 hard 分配到第 j 类簇的所有样本的均值.

期望最大化 (EM) 算法

1. 选择参数 Θ_d 的一个初始设置.
2. E 步，计算后验概率, (也就是标签的期望)

$$< Z_n^k >= P(Z|X, \Theta_d)$$

3. M 步，通过最大似然计算新的参数:

$$\Theta_{new} = \arg\max_{\Theta} \log \sum_z P(X, Z|\Theta_d)$$

$$= \arg\max_{\Theta} \log \sum_z P(Z|\Theta_z) P(X|Z, \Theta_d)$$

4. 检查参数值或者对数似然的收敛性，如果不满足收敛准则，则

$$\Theta_d \leftarrow \Theta_{new}$$

，回到第 2 步继续

Z 在 Kmeans 中为类标签，Θ 为质心

目录

1 聚类
- EM-KMeans
- 混合高斯模型 GMM
- 谱聚类

2 降维处理
- 主成分分析

混合高斯模型 GMM

$$P(X|\Theta) = \sum_{i=1}^{k} w_i f(X|\Theta_i), \quad \sum_{i=1}^{k} w_i = 1$$

$$f(X|\theta_j) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_j|}} exp(-\frac{1}{2}(X-\mu_j)^T \Sigma_j^{-1} (X-\mu_j))$$

- GMM 属于软分派 (soft).

混合高斯模型 GMM I

E-Step: 最大化期望来获取标签，对于每个样本 $x^{(i)}$ 的第 j 个类别计算:

$$z_j^{(i)} = P(z^{(i)} = j|x^{(i)}; w_j, \mu, \Sigma)$$

$$\hat{z}_{ij} = E_{P(z^{(i)}=j|X,\theta)}[z_{ij}] = \frac{w_j f(x|\theta)}{\sum_{j=1}^{k} w_j f(x|\theta)}$$

由于都是高斯分布，乘起来还是高斯分布
M-Step: 最大化似然度函数来获得参数

$$P(X, z|\theta) = \prod_{i=1}^{n}\prod_{j=1}^{k}(w_j f(x_i|\theta_j))^{z_{ij}}$$

$$\log P(X, z|\theta) = \sum_{i=1}^{n}\sum_{j=1}^{k} z_{ij}(\log w_j f(x_i|\theta_j))$$

混合高斯模型 GMM II

$$\max_{\theta,\alpha} \sum_{i=1}^{n} \sum_{j=1}^{k} z_{ij} (\log w_j + \log f(x_i|\theta_j)) \quad s.t. \sum_{j=1}^{k} w_j = 1$$

希望最大化期望函数来获得参数

$$L = \sum_{i=1}^{n} \sum_{j=1}^{k} z_{ij} (\log w_j + \log f(x_i|\theta_j)) + \lambda (\sum_{j=1}^{k} w_j - 1)$$

$$\frac{\partial L}{\partial w_j} = \sum_{i=1}^{n} \frac{z_{ij}}{w_j} - \lambda = 0$$

$$w_j^* = \frac{\sum_{i=1}^{n} z_{ij}}{\lambda}$$

混合高斯模型 GMM III

又因为 $\sum_{j=1}^{k} w_{.j} = 1$ 所以有

$$\sum_{j=1}^{k} \frac{\sum_{i=1}^{n} z_{ij}}{\lambda} = 1$$

$$\lambda = n \rightarrow w_j^* = \frac{\sum_{i=1}^{n} z_{ij}}{\sum_{j=1}^{k} \sum_{i=1}^{n} z_{ij}} = \frac{\sum_{i=1}^{n} z_{ij}}{n}$$

$$\frac{\partial L}{\partial \mu_j} = \sum_{i=1}^{n} z_{ij}(\Sigma_j^{-1}(x_i - \mu_j)) = 0$$

j 列的方差跟 i 没关系，可以消去

$$\mu_j = \frac{\sum_{i=1}^{n} z_{ij} x_i}{\sum_{i=1}^{n} z_{ij}}$$

期望最大化 (GMM) 算法 I

E-Step: 对于每个样本 $x^{(i)}$ 的第 j 个类别计算:

$$z_j^{(i)} = P(z^{(i)} = j|x^{(i)}; w_j, \mu, \Sigma)$$

$$\hat{z}_{ij} = E_{P(z^{(i)}=j|X,\theta)}[z_{ij}] = \frac{w_j f(x|\theta)}{\sum_{j=1}^{k} w_j f(x|\theta)}$$

M 步，通过最大似然计算新的参数:

$$w_j = \frac{\sum_{i=1}^{n} z_{ij}}{n}$$

$$\mu_j = \frac{\sum_{i=1}^{n} z_{ij} x_i}{\sum_{i=1}^{n} z_{ij}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{n} z_{ij}(x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{n} z_{ij}}$$

期望最大化 (GMM) 算法 II

4. 检查参数值或者对数似然的收敛性，如果不满足收敛准则，则重复 E-Step

```
from scipy import stats
w1, w2 = 0.5, 0.5
mu1, mu2 = 170, 160
std1, std2 = 10, 5
n = len(Data)  # 样本长度
zij=np.zeros([n,2])
for t in range(10):
    # E-step
    z1_up = w1 * stats.norm(mu1,std1).pdf(Data)
    z2_up = w2*stats.norm(mu2, std2).pdf(Data)
    z_all = (w1*stats.norm(mu1,std1).pdf(Data)+
             w2*stats.norm(mu2,std2).pdf(Data))+0.001

    rz1 = z1_up/z_all        # 为男分布的概率
    rz2 = z2_up/z_all        # 为女分布的概率
    # M-step
    mu1 = np.sum(rz1*Data)/np.sum(rz1)
    mu2 = np.sum(rz2*Data)/np.sum(rz2)
    std1 = np.sum(rz1*np.square(Data-mu1))/np.sum(rz1)
```

```
    std2 = np.sum(rz2*np.square(Data-mu2))/np.sum(rz2)
    w1 = np.sum(rz1)/n
    w2 = np.sum(rz2)/n
for i in range(n):
    zij[i][0] = rz1[i]/(rz1[i]+rz2[i])
    zij[i][1] = rz2[i]/(rz1[i]+rz2[i])
```

混合高斯模型 GMM I

例

使用混合高斯模型 GMM，计算如下数据点的聚类过程，

```
Data = np.array([1,2,6,7])
均值初值为:
mu1, mu2 = 1, 5
权重初值为:
w1, w2 = 0.5, 0.5
方差:
std1, std2 = 1, 1
K = 2
```

- 10 次迭代后数据的聚类标签是多少?

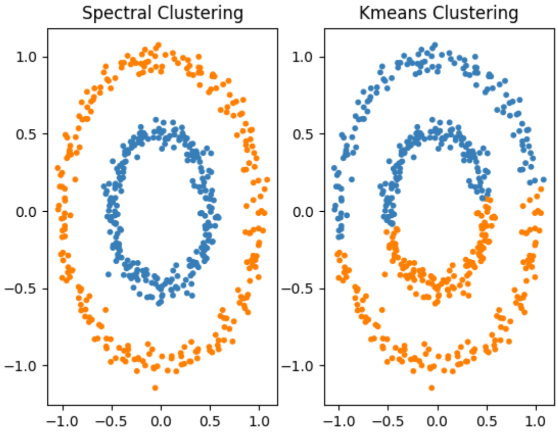
目录

1 聚类
- EM-KMeans
- 混合高斯模型 GMM
- 谱聚类

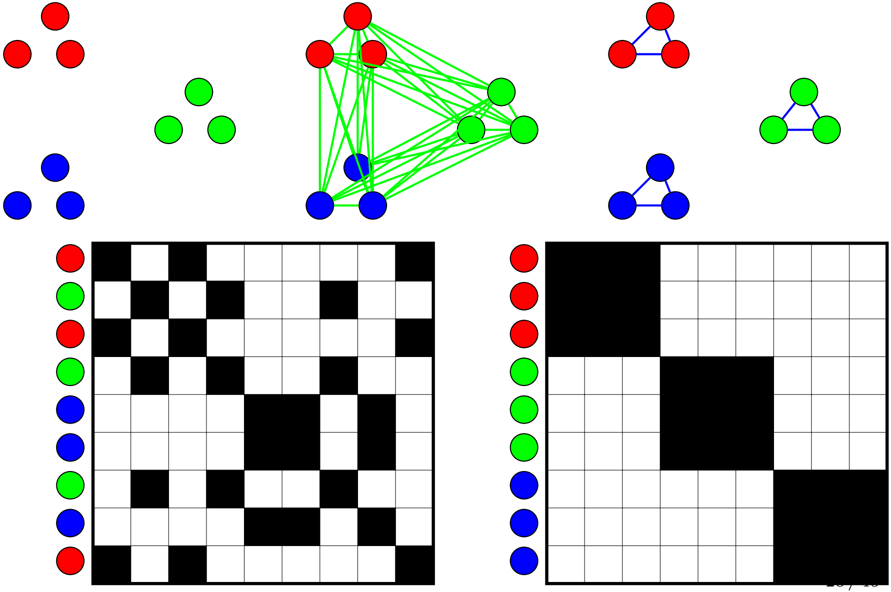
2 降维处理
- 主成分分析

谱聚类-Spectral Clustering

- 图的邻接矩阵的按照升序排列的特征值对应的特征向量就叫做图的谱



谱聚类-Spectral Clustering



谱聚类-Spectral Clustering

- 成对相似度矩阵

$$W_{i,j} = e^{\frac{-||x_i - x_j||_2^2}{\sigma^2}}, W_{ii} = 0$$

- 度矩阵

$$D(i,i) = \sum_j W_{ij}$$

- 拉普拉斯矩阵 Laplacian：

$$L = D - W$$

- 关联矩阵:

$$As(A,B) = \sum_{i \in A} \sum_{j \in B} W_{ij}$$

图割 Cuts

- 割（边）= 去掉后使得图不再连通的一组边集合 ß
- 割的权重

$$cut(A,B) = \sum_{i \in A} \sum_{j \in B} As(A,B)$$

- 规范化割:

$$Ncut(A,B) = \frac{cut(A,B)}{d_A} + \frac{cut(A,B)}{d_B}$$

- 转换 Ncut 为矩阵形式 (Shi &Malik 2000):

$$\min_x Ncut(x) = \min_y \frac{y^T(D-W)y}{y^TDy}, \quad s.t. y \in \{1,-b\}^n, y^TD1 = 0$$

y 是待求解的最小特征值对应的特征向量

图割 Cuts

- 等价与求其特征值及对应的特征向量

$$\min_x Ncut(x) = \arg\min_P tr(P^T LP), \quad s.t. P^T DP = 1$$

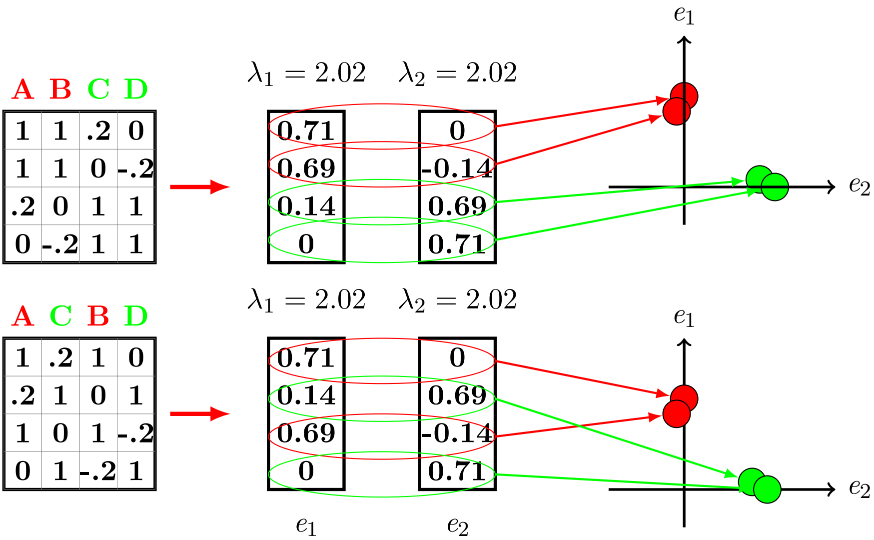
- P 是待求解的特征值对应的特征向量

谱聚类-Spectral Clustering

$$\begin{array}{cccc} A & B & C & D \\ \end{array}$$
$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \xrightarrow{\text{求解特征值}} \quad \lambda_1 = 2: \begin{bmatrix} 0.71 \\ 0.71 \\ 0 \\ 0 \end{bmatrix} \quad \lambda_2 = 2: \begin{bmatrix} 0 \\ 0 \\ 0.71 \\ 0.71 \end{bmatrix} \quad \lambda_3 = \lambda_4 = 0$$

$$\begin{bmatrix} 1 & 1 & .2 & 0 \\ 1 & 1 & 0 & -.2 \\ .2 & 0 & 1 & 1 \\ 0 & -.2 & 1 & 1 \end{bmatrix} \xrightarrow{\text{求解特征值}} \quad \lambda_1 = 2.02: \begin{bmatrix} 0.71 \\ 0.69 \\ 0.14 \\ 0 \end{bmatrix} \quad \lambda_2 = 2.02: \begin{bmatrix} 0 \\ -0.14 \\ 0.69 \\ 0.71 \end{bmatrix} \quad \lambda_3 = -0.02 \quad \lambda_4 = -0.02$$

谱聚类-Spectral Clustering



谱聚类-Spectral Clustering

1. 计算距离矩阵（例如欧氏距离）
2. 利用 KNN 计算相似度 W
3. 由 W 计算度矩阵 D 和拉普拉斯矩阵 L
标准化 $L \leftarrow D^{-1/2}LD^{-1/2}$
4. 对矩阵 L 进行特征值分解，得到特征向量 P
5. 将 P 当成样本送入 Kmeans 聚类获得聚类结果

$$C = (C_1, C_2, ..., C_k)$$

```
def W_Matrix(X):
  S = np.zeros((len(X), len(X)))
  sigma=0.2
  for i in range(len(X)):
    for j in range(i+1, len(X)):
      S[i][j] = np.exp(-np.linalg.norm(X[i]-X[j])/sigma)
      S[j][i] = S[i][j]
    return S
def calLaplacianMatrix(A):
    D = np.sum(A, axis=1)
    L = np.diag(D) - A
    sqrtD = np.diag(1.0 / (D ** (0.5)))
    return np.dot(np.dot(sqrtD, L), sqrtD)

W = W_Matrix(data)
Adjacent = KNN(W, k=10)
Laplacian = calLaplacianMatrix(Adjacent)
lambdax, P = np.linalg.eig(Laplacian)
x = zip(lambdax, range(len(lambdax)))
x = sorted(x, key=lambda x:x[0])
H = np.vstack([P[:,i] for (v, i) in x[:500]]).T
sp_kmeans = KMeans(n_clusters=2).fit(H)
```

瑞利商 Rayleigh quotient

定义

定义瑞利商（Rayleigh quotient）为:

$$R(A, x) = \frac{x^T A x}{x^T x}$$

其中，x 为非零向量，A 为 $n \times n$ Hermitian Matrix(共轭转置和自身相等的矩阵)，A 的特征向量即是函数 R(A,x) 的驻点 (critical point)，特征向量相对应的特征值即为函数在该驻点的值。

当所有元素均为实数。厄米特矩阵 A 是对称的，其特征值一定是实数。

定义

设 A 是 p 阶对称矩阵，其特征值依次是 $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$，，相应的一组正交特征向量是 $T = \{t_1, .., t_p\}$，则

$$max_{x \neq 0} \frac{x^T A x}{x^T x} = \lambda_1, if \quad x = t_1 \tag{1}$$

$$\max_{x^T t_k = 0, k=1,..,i-1, x \neq 0} \frac{x^T A x}{x^T x} = \lambda_i, if \quad x = t_i \tag{2}$$

由此，我们可知 R(A,x) 的最大值等于矩阵 A 最大的特征值，而最小值等于矩阵 A 的最小的特征值

$$\lambda_{min} \leq \frac{x^T A x}{x^T x} \leq \lambda_{max}$$

当向量 x 是标准正交基时，即满足 $x^T x = 1$，瑞利商（Rayleigh quotient）为:

$$R(A, x) = x^T A x$$

瑞利商定理 Rayleigh quotient I

简单证明 (1)
可以转化为拉格朗日乘子问题,c 为常数:

$$J(x) = x^T Ax - \lambda(x^T x - c)$$

$\frac{\partial J(x)}{\partial x} = 0 \rightarrow Ax = \lambda x$

$$R(A, x) = \frac{x^T Ax}{x^T x} = \frac{x^T \lambda x}{x^T x} = \lambda$$

目录

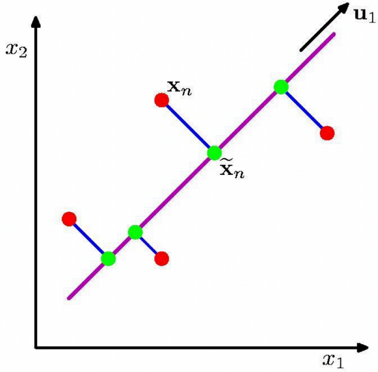
1 聚类
- EM-KMeans
- 混合高斯模型 GMM
- 谱聚类

2 降维处理
- 主成分分析

主成分分析

Principal Component Analysis

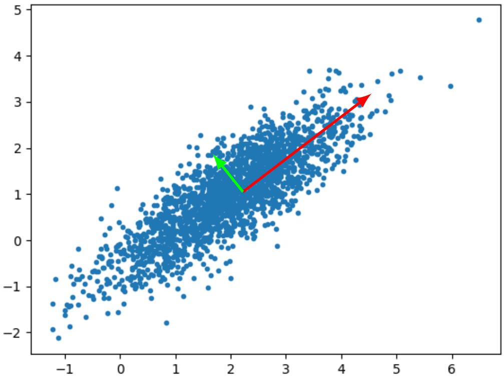
- 正交投影数据到低维度线性空间。
- 最大化投影数据的方差
- 最小化数据点和投影的均方距离



主成分分析

Principal Component Analysis

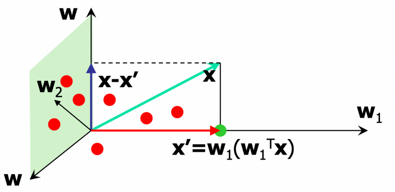
- 正交投影数据到低维度线性空间。
- 最大化投影数据的方差
- 最小化数据点和投影的均方距离



主成分分析

Principal Component Analysis

- PCA 向量源于质心
- 主成分 1 给出了最大方差方向
- 后续每个主成分都是:
  - 同上一个主成分正交
  - 残差空间中最大方差方向



主成分分析的推导 I

基于最小投影距离

- 中心化:$\sum_{i=1}^{m} x^{(i)} = 0$
- 经过投影变换后的新坐标系为:$\{w_1, ..., w_n\}, ||w||_2 = 1$ 且为正交基向量
- 如果我们将数据从 n 维降到 n' 维，则新的坐标系为 $\{w_1, w_2, ..., w_{n'}\}$, 样本点 $x^{(i)}$ 在 n' 维坐标系中的投影为 $z^{(i)} = (z_1^i ..., z_{n'}^i), z_j^{(i)} = w_j^T x(i) = (W^T X)_{ji}$
- 再将 $z^{(i)}$ 恢复原始数据 $x^{(i)}$, 则得到的 $\hat{x}^{(i)} = \sum_{j=1}^{n'} w_j z_j^{(i)} = W z^{(i)}$

主成分分析的推导 II

基于最小投影距离

- 我们希望所有的样本到这个新坐标系，并恢复回来的差异距离足够近，即最小化下式

$$L = \sum_{i=1}^{m} ||\hat{x}^{(i)} - x^{(i)}||_2^2$$

主成分分析的推导 III

基于最小投影距离

$$
\begin{aligned}
L &= \sum_{i=1}^{m} || Wz^{(i)} - x^{(i)} ||_2^2 & (3)\\
&= \sum_{i=1}^{m} (Wz^{(i)})^T Wz^{(i)} - 2\sum_{i=1}^{m} (Wz^{(i)})^T x^{(i)} + \sum_{i=1}^{m} (x^{(i)})^T x^{(i)} & (4)\\
&= \sum_{i=1}^{m} (z^{(i)})^T z^{(i)} - 2\sum_{i=1}^{m} (z^{(i)})^T W^T x^{(i)} + \sum_{i=1}^{m} (x^{(i)})^T x^{(i)} & (5)\\
&= \sum_{i=1}^{m} (z^{(i)})^T z^{(i)} - 2\sum_{i=1}^{m} (z^{(i)})^T z^{(i)} + \sum_{i=1}^{m} (x^{(i)})^T x^{(i)} & (6)\\
&= \sum_{i=1}^{m} (z^{(i)})^T z^{(i)} + (x^{(i)})^T x^{(i)} & (7)\\
&= -tr(W^T (\sum_{i=1}^{m} x^{(i)} (x^{(i)})^T) W) + \sum_{i=1}^{m} (x^{(i)})^T x^{(i)} & (8)
\end{aligned}
$$

主成分分析的推导 I

- 由于 $\sum_{i=1}^{m} x^{(i)}(x^{(i)})^T$ 是常量, 最小化上式等价于:

$$\arg\min_{W} -tr(W^T X X^T W) \quad s.t. \quad W^T W = I$$

- 利用拉格朗日函数可以得到

$$J(W) = -tr(W^T X X^T W) + \lambda(W^T W - I)$$

- 对 W 求导有 $-XX^T W + \lambda W = 0$, 整理下即为:

$$XX^T W = \lambda W$$

- W 为 $\frac{1}{m}XX^T$ 的 n' 个特征向量组成的矩阵; 而 λ 为 XX^T 的若干特征值组成的矩阵，特征值在主对角线上，其余位置为 0

主成分分析

Principal Component Analysis

- 由三个总体构成的 3 维随机变量（X，Y，Z）样本的协方差矩阵 XX^T

定义

样本的协方差矩阵

$$\begin{bmatrix} \frac{1}{n-1}\sum_{i=1}^{n}(X_i-\overline{X})^2 & \frac{1}{n-1}\sum_{i=1}^{n}(X_i-\overline{X})(Y_i-\overline{Y}) & \frac{1}{n-1}\sum_{i=1}^{n}(X_i-\overline{X})(Z_i-\overline{Z}) \\ \frac{1}{n-1}\sum_{i=1}^{n}(Y_i-\overline{Y})(X_i-\overline{X}) & \frac{1}{n-1}\sum_{i=1}^{n}(Y_i-\overline{Y})^2 & \frac{1}{n-1}\sum_{i=1}^{n}(Y_i-\overline{Y})(Z_i-\overline{Z}) \\ \frac{1}{n-1}\sum_{i=1}^{n}(Z_i-\overline{Z})(X_i-\overline{X}) & \frac{1}{n-1}\sum_{i=1}^{n}(Z_i-\overline{Z})(Y_i-\overline{Y}) & \frac{1}{n-1}\sum_{i=1}^{n}(Z_i-\overline{Z})^2 \end{bmatrix}$$

```python
def pca(X, d):
    # Centralization
    means = np.mean(X, 0)
    X = X - means
    # Covariance Matrix
    M=len(X)
    X=np.mat(X)
    #covM1 =(X.T * X)/(M-1)
    covM2=np.cov(X.T)
    eigval, eigvec = np.linalg.eig(covM2)
    indexes = np.argsort(eigval)[-d:]
    W = eigvec[:, indexes]
    return X*W
```

PCA 算法

1. 对所有的样本进行中心化：

$$x^{(i)} = x^{(i)} - \frac{1}{m}\sum_{j=1}^{m} x^{(j)}$$

2. 计算样本的协方差矩阵 XX^T
3. 对矩阵 XX^T 进行特征值分解
4. 取出最大的 n' 个特征值对应的特征向量 $(w_1, ..., w_{n'})$, 将所有的特征向量标准化后，组成特征向量矩阵 W。
5. 对样本集中的每一个样本 $x^{(i)}$, 转化为新的样本 $z^{(i)} = W^T x^{(i)}$
6. 得到输出样本集 $D = (z^{(1)}, ...z^{(m)})$

实例分析 I

假设我们的数据集有 10 个二维数据, 需要用 PCA 降到 1 维特征。

```
[[ 1.95  1.33]
 [ 1.7   0.18]
 [ 2.69  1.81]
 [ 3.4   1.88]
 [ 0.41 -0.05]
 [ 1.92  1.81]
 [ 3.47  2.08]
 [ 0.92  0.18]
 [ 1.66  1.33]
 [ 2.08  1.64]]
```

- Step1: 对样本中心化

实例分析 II

```
array([[-0.249,  0.309],
[ 0.011,  0.889],
[ 0.661,  0.449],
[ 0.901, -0.541],
[ 0.511,  0.659],
[-1.089, -1.571],
[-0.879, -0.611],
[ 0.231,  0.599],
[-0.829, -0.481],
[ 0.731,  0.299]])
```

- Step2. 求样本的协方差矩阵
  XX^T=matrix([[0.53394333, 0.34250111], [0.34250111, 0.59281]])

实例分析 III

- 求出特征值
  array([0.21961318, 0.90714015])
- 对应的特征向量
  t1=[-0.7367567   0.67615794]^T
  t2=[- 0.67615794   -0.7367567]^T
- 由于 0.90714015 最大，k=1 的特征向量为:
  w_1=[-0.67615794   -0.7367567]^T
- W=[-0.67615794   -0.7367567]^T
- $z^{(i)} = W^T x^{(i)}$, 降维后的数据为

实例分析 IV

```
[[-0.0592945 ],
[-0.66241445],
[-0.77774416],
[-0.21063293],
[-0.83103938],
[ 1.89378078],
[ 1.04450117],
[-0.59750975],
[ 0.91491491],
[-0.71456171]]
```

作业 I

假设我们的数据集有 10 个 3 维数据, 需要用 PCA 降到 2 维特征。

```
array([[ 3.25,  1.85, -1.29],
       [ 3.06,  1.25, -0.18],
       [ 3.46,  2.68,  0.64],
       [ 0.3 , -0.1 , -0.79],
       [ 0.83, -0.21, -0.88],
       [ 1.82,  0.99,  0.16],
       [ 2.78,  1.75,  0.51],
       [ 2.08,  1.5 , -1.06],
       [ 2.62,  1.23,  0.04],
       [ 0.83, -0.69, -0.61]])
```

- 给出求解过程