主成分分析 (principle component ananlysis)

Guangyao Zhao

2022-05-10

Contents

样本表示：

$$\mathrm{X} = (x_1; x_2; ...; x_3) = \begin{pmatrix} x_{11} & x_{11} & \cdots & x_{1m} \\ x_{21} & x_{21} & \cdots & x_{2m} \\ \vdots & & & \\ x_{n1} & x_{n1} & \cdots & x_{nm} \end{pmatrix}$$

其中 n 个样本，m 个维度。

样本均值

$$\bar{\mathrm{X}} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}(x_1, x_2, ..., x_n) \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}_n = \frac{1}{n}\mathrm{X}^{\mathrm{T}} 1_n$$

其中：

$$1_n = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}_n$$

样本方差矩阵

$$\begin{aligned} S &= \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^{\mathrm{T}} \\ &= \frac{1}{n}\left(x_1 - \bar{x}, x_2 - \bar{x}, ..., x_n - \bar{x}\right) \begin{pmatrix} (x_1 - \bar{x})^{\mathrm{T}} \\ (x_2 - \bar{x})^{\mathrm{T}} \\ \vdots \\ (x_n - \bar{x})^{\mathrm{T}} \end{pmatrix} \\ &= \frac{1}{n}\mathrm{X}^{\mathrm{T}}\left(\mathrm{I}_n - \bar{\mathrm{X}}1_n 1_n^{\mathrm{T}}\right)\left(\mathrm{X}^{\mathrm{T}}\left(\mathrm{I}_n - \bar{\mathrm{X}}1_n 1_n^{\mathrm{T}}\right)\right)^{\mathrm{T}} \end{aligned}$$

将 $\bar{\mathrm{X}} = \frac{1}{n}\mathrm{X}^{\mathrm{T}}1_n$ 代入上式子：

$$\begin{aligned} S &= \frac{1}{n}\left[\mathrm{X}^{\mathrm{T}}\left(\mathrm{I}_n - \frac{1}{n}1_n 1_n^{\mathrm{T}}\right)\right]\left[\mathrm{X}^{\mathrm{T}}\left(\mathrm{I}_n - \frac{1}{n}1_n 1_n^{\mathrm{T}}\right)\right]^{\mathrm{T}} \\ &= \frac{1}{n}\mathrm{X}^{\mathrm{T}}\mathrm{H}\mathrm{H}^{\mathrm{T}}\mathrm{X} \\ &= \frac{1}{n}\mathrm{X}^{\mathrm{T}}\mathrm{H}\mathrm{X} \end{aligned}$$

其中：$\mathrm{H} = \mathrm{I}_n - \frac{1}{n}1_n 1_n^{\mathrm{T}}$。H 为中心矩阵。

由以上可证明协方差矩阵 S 是对称矩阵，可对角化。

最大投影差

- 一个中心：将一组可能线性相关的变量，变成一组线性无关的变量。即，对原始特征的重构。
- 两个基本点：最大投影方差；最小重构距离。

向量 a 在向量 u 方向的投影计算公式为：

$$d = \frac{au}{u}$$

其中 u 为单位向量，即 $u^{\mathrm{T}}u = 1$。

PCA 的思想是将原有空间的样本尽可能分散地映射到一个子空间，即各维度应该最大。且尽可能的各维度线性无关，即使基向量正交。根据线性代数，我们可以知道同一元素的协方差就表示该元素的方差，不同元素之间的协方差就表示它们的相关性。

此时就可将问题转化为（首先需要中心化）：

$$
\begin{aligned}
J &= \sum_{i=1}^{n} \left((x_i - \bar{x})^{\mathrm{T}} u_1 \right)^2, \\
&= \sum_{i=1}^{n} u_1^{\mathrm{T}} (x_i - \bar{x})(x_i - \bar{x})^{\mathrm{T}} u_1 \\
&= u_1^{\mathrm{T}} \left(\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^{\mathrm{T}} \right) u_1 \\
&= u_1^{\mathrm{T}} \mathrm{S} u_1
\end{aligned}
$$

求 J 最大值，将其转化为优化问题：

$$
u_1^* = \underset{u_1}{\operatorname{argmax}}\, u_1^{\mathrm{T}} \mathrm{S} u_1
$$

上式子满足：$s.t. u^{\mathrm{T}} u = 1$。引入拉格朗日乘子法：

$$
\begin{aligned}
L(u_1, \lambda) &= u_1^{\mathrm{T}} \mathrm{S} u_1 + \lambda (1 - u_1^{\mathrm{T}} u_1) \\
\frac{\partial L(u_1, \lambda)}{\partial u_1} &= 2 S u_1 - 2 \lambda u_1 = 0
\end{aligned}
$$

即：$S u_1 = \lambda u_1$。其中 λ 为特征值，u_1 为特征向量。

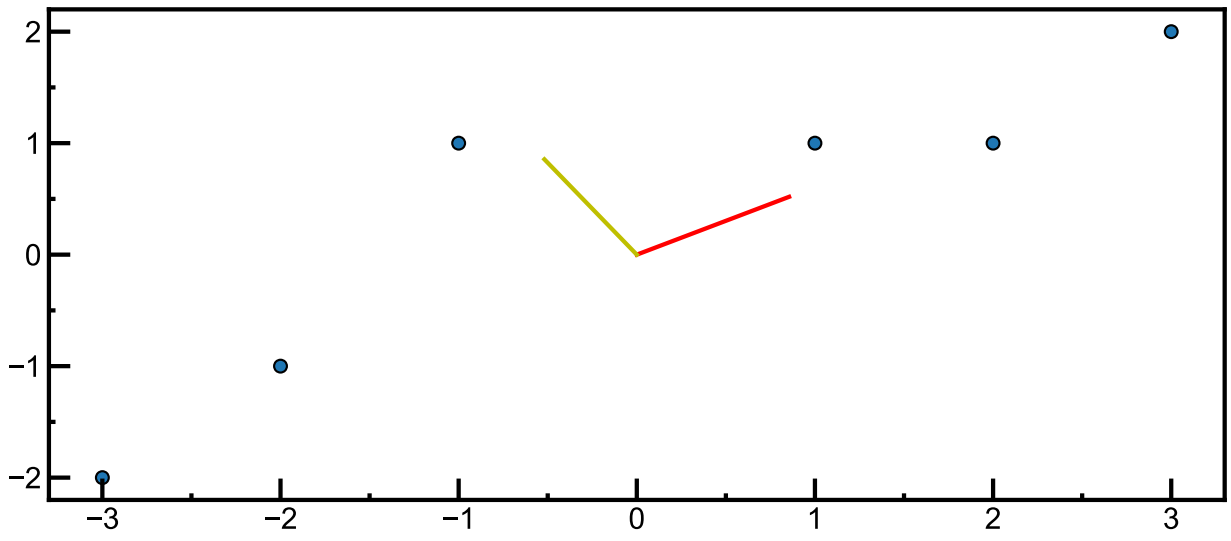
以上推导过程说明：信息量保存能力最大的基向量一定是样本矩阵 X 的协方差矩阵的特征向量，并且这个特征向量保存的信息量就是它对应的特征值的绝对值。这个推导过程就解释了为什么 PCA 算法要利用样本协方差的特征向量矩阵来降维。

代码

```python
##Python 实现 PCA
import numpy as np
import matplotlib.pyplot as plt

def pca(X, k):

```

```
    m = X.shape[1]
    norm_X = X - np.mean(X, axis=0)  # 中心化
    scatter_matrix = np.dot(np.transpose(norm_X), norm_X)  # 协方差矩阵

    #Calculate the eigenvectors and eigenvalues
    eig_val, eig_vec = np.linalg.eig(scatter_matrix)
    eig_pairs = [(np.abs(eig_val[i]), eig_vec[:, i]) for i in range(m)]
    print('eig_vec:', eig_vec)
    print('eig_pairs: ', eig_pairs)

    eig_pairs.sort(reverse=True)  # 按照特征值从大到小排序
    feature = np.array([ele[1] for ele in eig_pairs[:k]])  # 选择最大的 k 个特征向量

    #get new data
    data = np.dot(norm_X, feature.T)  # 降维

    # 绘图
    x = X[:, 0].flatten()
    y = X[:, 1].flatten()
    plt.scatter(x, y)
    plt.plot([eig_vec[:, 0][0], 0], [eig_vec[:, 0][1], 0],
             color='red')  # 特征向量 1
    plt.plot([eig_vec[:, 1][0], 0], [eig_vec[:, 1][1], 0], color='y')  # 特征向量 2
    plt.show()
    return data

X = np.array([[-1, 1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])
print(pca(X, 1))
```

```
eig_vec: [[ 0.8549662  -0.51868371]
 [ 0.51868371  0.8549662 ]]
eig_pairs: [(37.70674550364641, array([0.8549662 , 0.51868371])), (1.6265878296869225, array([-
0.51868371,  0.8549662 ]))]


[[-0.50917706]
 [-2.40151069]
```




```
[-3.7751606 ]
[ 1.20075534]
[ 2.05572155]
[ 3.42937146]]
```