Python编程与人工智能实践

# 算法篇：数据降维-PCA

于泓
鲁东大学
信息与电气工程学院
2021.9.24

# 数据降维

- 数据降维是数据挖掘和信号处理任务中，对输入数据进行预处理的常用手段，其目的在于从高维的输入数据中找出能够代表数据特性、能够有利于分类的低维特征

# PCA(Principal Component Analysis) 主成分分析

- PCA是一种使用最广泛的数据降维算法。PCA的主要思想是将n维特征映射到k维上，这k维是全新的正交特征也被称为主成分，是在原有n维特征的基础上重新构造出来的k维特征

将一个维度为n的矢量X分解成k个n维正交矢量$v_i$的线性叠加
$v_i$的系数$[y_1,y_2,y_3,\ldots y_k]$ 就是降维后的特征

$$X_1 = y_{11}v_1 + y_{12}v_2 + \ldots + y_{1k}v_k$$

$$X_2 = y_{21}v_1 + y_{22}v_2 + \ldots + y_{2k}v_k$$

$$X_3 = y_{31}v_1 + y_{32}v_2 + \ldots + y_{3k}v_k$$

例如:

$$\begin{bmatrix}1\\1\\1\end{bmatrix} = 1\begin{bmatrix}1\\0\\0\end{bmatrix} + 1\begin{bmatrix}0\\1\\0\end{bmatrix} + 1\begin{bmatrix}0\\0\\1\end{bmatrix}$$

其中 $v_i^T v_i = 1$

$v_i^T v_j = 0$

$$y_i = X^T v_i$$

投影

将一个维度为n的矢量X分解成k个n维正交矢量$v_i$的线性叠加
$v_i$的系数$[y_1,y_2,y_3,\dots y_k]$ 就是降维后的特征

$$X_1 = \boxed{y_{11}} v_1 + y_{12} v_2 + \dots + y_{1k} v_k$$

$$X_2 = \boxed{y_{21}} v_1 + y_{22} v_2 + \dots + y_{2k} v_k$$

$$X_3 = \boxed{y_{31}} v_1 + y_{32} v_2 + \dots + y_{3k} v_k$$

方差最大

例如:

$$\begin{bmatrix}1\\1\\1\end{bmatrix} = 1\begin{bmatrix}1\\0\\0\end{bmatrix} + 1\begin{bmatrix}0\\1\\0\end{bmatrix} + 1\begin{bmatrix}0\\0\\1\end{bmatrix}$$

其中

$$v_i^T v_i = 1$$

$$v_i^T v_j = 0$$

投影

$$y_i = X^T v_i$$

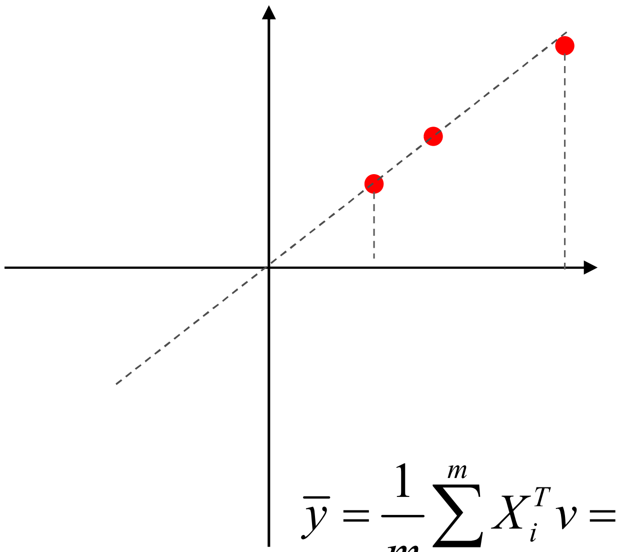
PCA的任务：寻找一组**正交基$v_i$**,使所有样本
沿着$v_i$进行投影后，**方差最大**（**信息量最大**）

$$X_1 = y_{11}v_1 + y_{12}v_2 + \ldots + y_{1k}v_k$$

$$X_2 = y_{21}v_1 + y_{22}v_2 + \ldots + y_{2k}v_k$$

$$X_3 = y_{31}v_1 + y_{32}v_2 + \ldots + y_{3k}v_k$$

方差最大



$$\bar{y} = \frac{1}{m}\sum_{i=1}^{m} X_i^T v = \left(\frac{1}{m}\sum_{i=1}^{m} X_i^T\right) v$$

对X进行中心化处理，则 $\bar{y} = \mathbf{0}$

$$S^2 = \frac{1}{m-1}\sum_{i=1}^{m}(y_i - \bar{y})^2$$

$$= \frac{1}{m-1}\sum_{i=1}^{m}(y_i)^2 = \frac{1}{m-1}\sum_{i=1}^{m}(X_i^T v)^2 = \frac{1}{m-1}(vX^T)(vX^T)^T$$

$$= v\frac{X^T X}{m-1}v^T = vCv^T$$

根据拉格朗日公式

$$F(v) = vCv^T + \lambda(1 - v^T v)$$

求导可得:

$$\frac{\partial F(v)}{\partial v} = 2Cv^T - 2\lambda v^T = 0$$

$$Cv^T = \lambda v^T$$

$\lambda$ 就是$C$的特征值

$v$ 就是特征值所对应的特征矢量

PCA 降维的一般步骤

（1）对输入数据X（m*n，m为样本数目，n为特征维度）进行中心化处理（减均值（1*n））

（2）计算 $C = \frac{X^T X}{m-1}$ C的维度（n*n）

（3）对C 进行特征值分解，并取最大的k个特征值所对应的特征矢量组成降维矩阵V（k*n）

(4) 进行降维 $y = XV^T$

代码实现:

```
# data 输入数据  维度 [N, D]
# n_dim: 降维后的维度
# 返回 [N,n_dim]
def pca(data, n_dim):

    N,D = np.shape(data)

    data = data - np.mean(data, axis = 0, keepdims = True)

    C = np.dot(data.T, data)/(N-1)  # [D,D]

    # 计算特征值和特征向量
    eig_values, eig_vector = np.linalg.eig(C)

    # 将特征值进行排序选取 n_dim 个较大的特征值
    indexs_ = np.argsort(-eig_values)[:n_dim]

    # 选取相应的特征向量组成降维矩阵
    picked_eig_vector = eig_vector[:, indexs_] # [D,n_dim]

    # 对数据进行降维
    data_ndim = np.dot(data, picked_eig_vector)
    return data_ndim, picked_eig_vector
```

```
def draw_pic(datas,labs):
    plt.cla()
    unque_labs = np.unique(labs)
    colors = [plt.cm.Spectral(each)
          for each in np.linspace(0, 1,len(unque_labs))]
    p=[]
    legends = []
    for i in range(len(unque_labs)):
        index = np.where(labs==unque_labs[i])
        pi = plt.scatter(datas[index, 0], datas[index, 1], c =colors[i] )
        p.append(pi)
        legends.append(unque_labs[i])

    plt.legend(p, legends)
    plt.show()


if __name__ == "__main__":

    # 加载数据
    data = np.loadtxt("iris.data",dtype="str",delimiter=',')
    feas = data[:,:-1]
    feas = np.float32(feas)
    labs = data[:,-1]

    # 进行降维
    data_2d, picked_eig_vector= pca(feas, 2)


    #绘图
    draw_pic(data_2d,labs)
```

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.4,3.7,1.5,0.2,Iris-setosa
4.8,3.4,1.6,0.2,Iris-setosa
4.8,3.0,1.4,0.1,Iris-setosa
4.3,3.0,1.1,0.1,Iris-setosa
5.8,4.0,1.2,0.2,Iris-setosa
5.7,4.4,1.5,0.4,Iris-setosa
5.4,3.9,1.3,0.4,Iris-setosa
5.1,3.5,1.4,0.3,Iris-setosa
5.7,3.8,1.7,0.3,Iris-setosa
5.1,3.8,1.5,0.3,Iris-setosa
5.4,3.4,1.7,0.2,Iris-setosa
5.1,3.7,1.5,0.4,Iris-setosa
4.6,3.6,1.0,0.2,Iris-setosa
5.1,3.3,1.7,0.5,Iris-setosa
4.8,3.4,1.9,0.2,Iris-setosa
5.0,3.0,1.6,0.2,Iris-setosa
5.0,3.4,1.6,0.4,Iris-setosa
```