特征值与特征向量，特征值计算

2025年3月17日　　16:29

特征向量与特征空间

在机器学习中，数据通常以表格（或称矩阵）的形式表示。表格中的每一列称为特征（Feature），每一行称为数据点（Data Point）或样本（Sample）。**特征向量**是将这些特征组合在一起的向量，其中每个元素表示数据点的一个特征值。

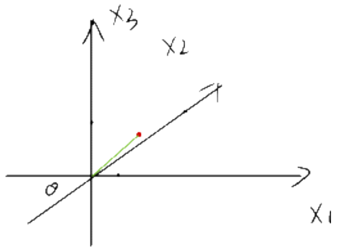
对以下表格，有以下特征：

- 身高（Height）
- 体重（Weight）
- 年龄（Age）

对于张三，我们可以将这些特征值组合在一起，形成一个特征向量：

$$\vec{x}=[x1, x2, x3]=[1.80, 75, 25]$$

其中，x1 表示身高，x2 表示体重，x3表示年龄。



特征

	身高	体重	年龄
张三	1.8	75	25
李四	1.75	70	30
王五	1.78	73	22

→特征向量

特征空间（Feature Space）是一个抽象的多维空间，其中每个维度对应于数据点的一个特征。特征空间的维数等于特征的数量。

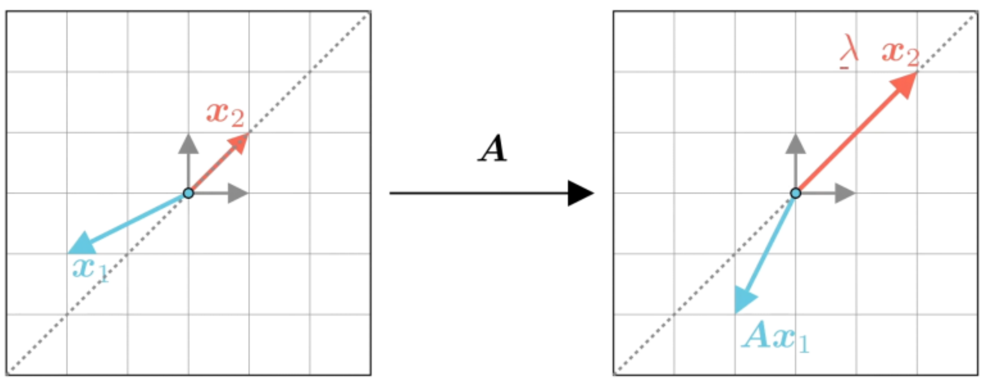
在上面的例子中，我们有三个特征，因此特征空间的维数为3。在这个空间中，我们可以将一个人表示为一个点：

$$\vec{p}=[1.80, 75, 25]$$

这个点代表了这个人在特征空间中的位置。

1.什么是特征值?

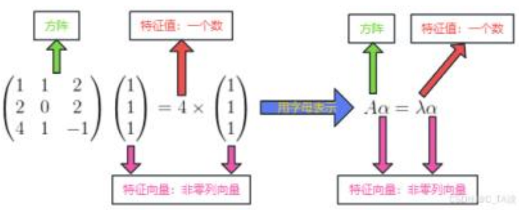
先从几何意义上理解：



向量x1经过矩阵A映射——>Ax1——>x1与Ax1不在一条直线上——>不构成特征向量

向量x2经过矩阵A映射——>Ax2——>x2与Ax2在一条直线上——>构成特征向量——>矩阵A只对x2起到了伸缩作用

因此：



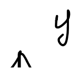
特征向量（要求非0）

$$AX=\lambda X$$

映射矩阵(nxn)　　特征值

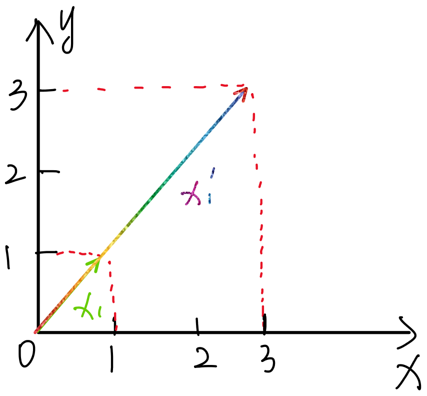
注：

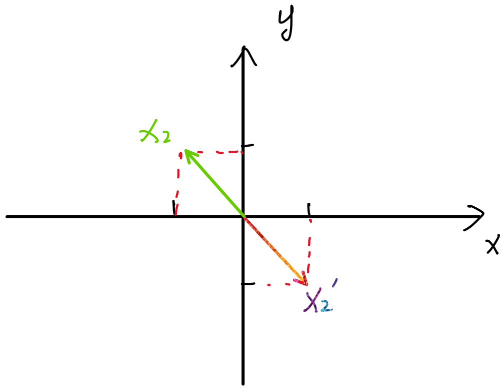
如果特征值是**负数**，那说明了矩阵不但把向量拉长（缩短）了，而且让向量指向了相反的方向。



注:

如果特征值是**负数**，那说明了矩阵不但把向量拉长（缩短）了，而且让向量指向了相反的方向。





$$\underbrace{[1\ \ 1]}_{x_1} \times \underbrace{\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}}_{A} = \underbrace{3}_{\lambda_1}\underbrace{[1\ \ 1]}_{x_1}$$

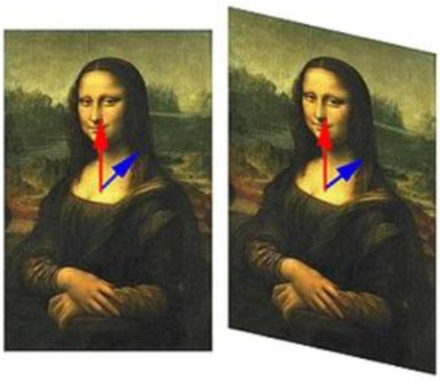
$$\underbrace{[-1\ \ 1]}_{x_2} \times \underbrace{\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}}_{A} = \underbrace{-1}_{\lambda_2}\underbrace{[-1\ \ 1]}_{x_2}$$

同向放大　　　　同模反向

[【熟肉】线性代数的本质 - 10 - 特征向量与特征值_哔哩哔哩_bilibili]

简而言之——>特征向量就是在线性变化当中**不变**的向量

在线性代数中，“特征” 就是一种更抽象的描述。矩阵乘法对应了一个变换，是把任意一个向量变成另一个方向或长度都大多不同的新向量。在这个变换的过程中，原向量主要发生旋转、伸缩的变化。如果矩阵对某一个向量或某些向量只发生伸缩（尺度）变换，而**没有产生旋转**的效果（也就意味着张成的子空间没有发生改变），这样的向量就认为是特征向量。



在这个仿射变换中，蒙娜丽莎的图像被变形，但是中心的纵轴在变换下保持不变。（注意：角落在右边的图像中被裁掉了。）蓝色的向量，从胸部到肩膀，其方向改变了，但是红色的向量，从胸部到下巴，其方向不变。因此红色向量是该变换的一个特征向量，而蓝色的不是。**因为红色向量既没有被拉伸又没有被压缩，其特征值为1。所有沿着垂直线的向量也都是特征向量，它们的特征值相等。它们构成这个特征值的特征空间。**

2.特征值计算方法

$$A\alpha = \lambda\alpha \quad \Rightarrow \quad (\lambda E - A)\alpha = \theta,$$

即要求齐次线性方程组 $(\lambda E - A)x = \theta$ 有非零解，

即方程 $|\lambda E - A| = 0$ 的根就是矩阵A的特征值，

相应**非零解**即为特征向量。

例题:

设 $A=\begin{pmatrix}-2 & 1 & 1\\ 0 & 2 & 0\\ -4 & 1 & 3\end{pmatrix}$，求$A$的特征值与特征向量。

1、求特征方程$|\lambda E-A|=0$的全部根，即为矩阵 A 的全部特征值；

$$|\lambda E-A|=\begin{vmatrix}\lambda+2 & -1 & -1\\ 0 & \lambda-2 & 0\\ 4 & -1 & \lambda-3\end{vmatrix}$$

降阶：$(\lambda-2)\begin{vmatrix}\lambda+2 & -1\\ 4 & \lambda-3\end{vmatrix}$

$=(\lambda-2)[(\lambda+2)(\lambda-3)-4]$

$=(\lambda-2)[\lambda^2-\lambda+2]$

$=(\lambda-2)^2(\lambda+1)$

$$=(\lambda-2)^2(\lambda+1)=0,$$

所以A的特征值为 $\lambda_1=2$(二重根), $\lambda_2=-1$.

2、对每一特征值λ_i，求出齐次线性方程组

$$(\lambda_i E-A)x=\theta$$

的一个基础解系$\eta_1,\eta_2,\Lambda\ ,\eta_{n-r}$，其中 r 为$\lambda_i E-A$ 的秩；

则 A 的属于特征值 λ_i 的全部特征向量为

$$k_1\eta_1+k_2\eta_2+\Lambda\ +k_{n-r}\eta_{n-r},$$

其中$k_1,k_2,\Lambda\ ,k_{n-r}$为不全为零的任意数。

对 $\lambda_1=2$，$2E-A=\begin{pmatrix}4 & -1 & -1\\ 0 & 0 & 0\\ 4 & -1 & -1\end{pmatrix}\to\begin{pmatrix}4 & -1 & -1\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix}$,

相应齐次线性方程组的基础解系为

$$\alpha_1=(1,0,4)^T,\quad \alpha_2=(0,1,-1)^T,$$

因此属于特征值 $\lambda_1=2$ 的全部特征向量为

$$k_1\alpha_1+k_2\alpha_2\ (k_1,k_2\text{不全为零});$$

对 $\lambda_2=-1$，$-E-A=\begin{pmatrix}1&-1&-1\\0&-3&0\\4&-1&-4\end{pmatrix}\rightarrow\begin{pmatrix}1&-1&-1\\0&1&0\\0&0&0\end{pmatrix}$，

相应齐次线性方程组的基础解系为

$$\alpha_3=(1,0,1)^T,$$

因此属于特征值 $\lambda_2=-1$ 的全部特征向量为 $k_3\alpha_3(k_3\neq 0)$.

用python中的numpy库计算特征值与特征向量

代码:

```
import numpy as np
a = np.matrix('-2 1 1;0 2 0;-4 1 3')
b=np.linalg.eig(a)
print(b)
```

结果:

```
EigResult(eigenvalues=array([-1.,  2.,  2.]), eigenvectors=matrix([[-0.70710678, -0.24253563,  0.30151134],
        [ 0.        ,  0.        ,  0.90453403],
        [-0.70710678, -0.9701425 ,  0.30151134]]))
```

其中：特征值EigResult(eigenvalues=array([-1., 2., 2.])

特征向量：eigenvectors=matrix
([[-0.70710678, -0.24253563, 0.30151134],
[0. , 0. , 0.90453403],
[-0.70710678, -0.9701425 , 0.30151134]]))

numpy 计算出的特征向量是归一化的，即每个特征向量的模长为 1

- 第一列特征向量的模长：

$$\sqrt{(-0.70710678)^2+0^2+(-0.70710678)^2}=\sqrt{0.5+0+0.5}=\sqrt{1}=1$$

- 第二列特征向量的模长：

$$\sqrt{(-0.24253563)^2+0^2+(-0.9701425)^2}=\sqrt{0.0588+0+0.9412}=\sqrt{1}=1$$

- 第三列特征向量的模长：

$$\sqrt{(0.30151134)^2+(0.90453403)^2+(0.30151134)^2}=\sqrt{0.0909+0.8182+0.0909}=$$

- 特征值 -1 对应的特征向量 $\begin{pmatrix}-0.70710678\\0\\-0.70710678\end{pmatrix}$ 可以表示为 $-\frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\1\end{pmatrix}$，与手动计算的 $\begin{pmatrix}1\\0\\1\end{pmatrix}$ 方向一致。

通过一个简单的线性回归示例来展示如何使用Python的scikit-learn库实现特征向量和特征空间的计算。

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
# 生成一组随机数据
np.random.seed(42)
X = np.random.rand(100, 2)
y = 3 * X[:, 0] + 2 * X[:, 1] + np.random.randn(100)
# 标准化特征向量
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
# 创建线性回归模型
model = LinearRegression()
# 训练模型
model.fit(X_scaled, y)
# 预测目标变量
y_pred = model.predict(X_scaled)
# 计算预测误差
error = np.mean((y_pred - y) ** 2)
print("预测误差:", error)
# 查看特征向量
print("特征向量:", X_scaled)
# 查看特征空间的维数
print("特征空间的维数:", X_scaled.shape[1])
```

在这个示例中，首先生成了一组随机数据，其中包含两个特征。然后使用标准化器（StandardScaler）对特征向量进行了标准化，以便在训练模型时避免梯度下降算法的收敛问题。接下来，我们创建了一个线性回归模型，并使用训练数据集来训练模型。最后使用训练好的模型来预测目标变量，并计算预测误差