机器学习丨PCA降维的经典算法

爱数据原统计网 5天前

以下文章来源于赵小洛洛洛，作者小洛同学

赵小洛洛洛

一枚来自北京的互联网行业的数据分析师，主要分享互联网数据分析、产品、运营相关...

文末扫码领【数据分析实战宝典】

作者：饭小米
转自：赵小洛洛洛

在机器学习的领域中，我们对原始数据进行特征提取，经常会得到高维度的特征向量。在这些多特征的高维空间中，会包含一些冗余和噪声。所以我们希望通过降维的方式来寻找数据内部的特性，提升特征表达能力，降低模型的训练成本。PCA是一种降维的经典算法，属于**线性、非监督、全局**的降维方法。

01 PCA原理

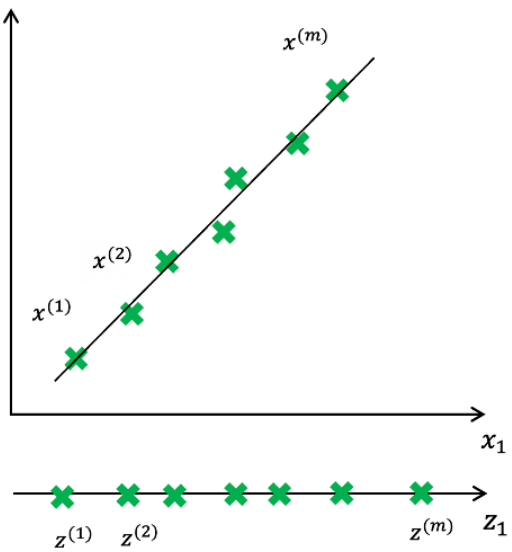
PCA的原理是线性映射，简单的说就是将高维空间数据投影到低维空间上，然后将数据包含信息量大的主成分保留下来，忽略掉对数据描述不重要的次要信息。

而对于正交属性空间中的样本，如何用一个**超平面**对所有样本进行恰当合适的表达呢？若存在这样的超平面，应该具有两种性质：

- 所有样本点到超平面的距离最近
- 样本点在这个超平面的投影尽可能分开

以上两种性质便是主成分分析的两种等价的推导，即PCA最小平方误差理论和PCA最大方差理论，本篇主要为大家介绍**最大方差理论。**

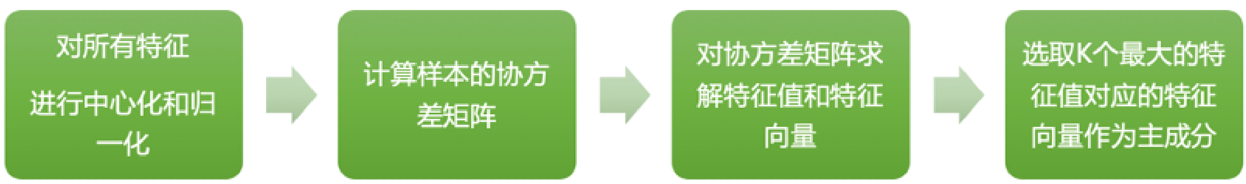
PCA的降维操作是选取数据**离散程度最大**的方向(方差最大的方向)作为第一主成分，第二主成分选择方差次大的方向，并且与第一个主成分**正交**。不算重复这个过程直到找到k个主成分。



数据点分布在主成分方向上的离散程度最大，且主成分向量彼此之间正交；

02

PCA算法实现步骤



1、对所有数据特征进行中心化和归一化

对样本进行平移使其重心在原点，并且消除不同特征数值大小的影响，转换为统一量纲：

假设训练集的样本为一维变量 $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$

样本均值 $\mu = \sum_{i=1}^{n} x^{(i)}$

中心化 $x^{(i)} = x^{(i)} - \mu$

归一化 $x^{(i)} = \frac{x^{(i)} - \mu}{\sigma}$

2、计算样本的协方差矩阵

协方差是对两个随机变量联合分布线性相关程度的一种度量；

$$X = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{pmatrix} \qquad X^T = \begin{pmatrix} x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(m)} \\ x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ x_n^{(1)} & x_n^{(2)} & \cdots & x_n^{(m)} \end{pmatrix}$$

$$\text{协方差矩阵 } \Sigma = \frac{1}{m} X^T X = \frac{1}{m} \Sigma_{i=1}^{m} x^{(i)} (x^{(i)})^T = \begin{pmatrix} var(x_1) & cov(x_1, x_2) & \cdots & cov(x_1, x_n) \\ cov(x_2, x_1) & var(x_2) & \cdots & cov(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ cov(x_n, x_1) & cov(x_n, x_2) & \cdots & var(x_n) \end{pmatrix}$$

3、对协方差矩阵求解特征值和特征向量

$$\begin{pmatrix} var(x_1) & cov(x_1, x_2) & \cdots & cov(x_1, x_n) \\ cov(x_2, x_1) & var(x_2) & \cdots & cov(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ cov(x_n, x_1) & cov(x_n, x_2) & \cdots & var(x_n) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \lambda_i \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

所有数据点在主特征u_i上投影点的方差$\sigma_1^2 = \lambda_1$

注意点：

① 对称矩阵的特征向量**相互正交**，其点乘为0

② **数据点在特征向量上投影的方差，为对应的特征值**，选择特征值大的特征向量，就是选择点投影方差大的方向，即是具有高信息量的主成分；次佳投影方向位于最佳投影方向的正交空间，是第二大特征值对应的特征向量，以此类推；

4、选取k个最大大特征值对应的特征向量，即是k个主成分

$$U^{(k)} = \begin{pmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \cdots & u^{(k)} \\ | & | & & | \end{pmatrix}$$

U是协方差矩阵所有的特征向量构成的矩阵，对应的特征值满足：λ1>λ2>⋯>λn，同时使其满足在主成分向量上投影的方差和占总方差的99%或者95%以上，即确定了k的选取。

03
降维Python实现

1、配置环境，导入相关包

```
import pandas as pd
import numpy as np
# import warnings
from matplotlib import pyplot as plt
# from pylab import mpl
# import seaborn as sns
# from IPython.display import Image
from mpl_toolkits.mplot3d import Axes3D
# from numpy import random
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
# %matplotlib inline
```

2、读取数据

```
from sklearn.datasets import load_iris
load_iris=load_iris()
pima_df = pd.read_csv('/Users/zhaoluo/opt/anaconda3/lib/python3.8/site-packages/sklearn/datasets/data/iris.csv')
pima_df.head()
```

	150	4	setosa	versicolor	virginica
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

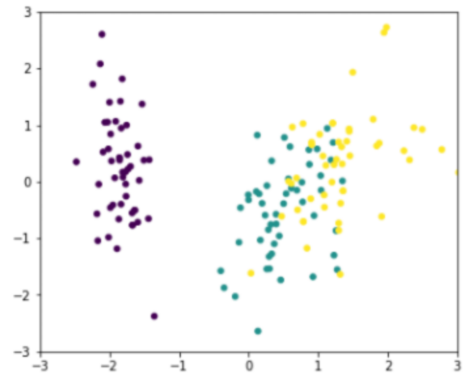
3、读取特征、标签列，并进行中心化归一化，选取主成分个数，前2个主成分的方差和>95%

```
# 提取特征列
X = pima_df.iloc[:, 0:3]
# 提取标签列
Y = pima_df.iloc[:, 4]
#中心化归一化
scaler = StandardScaler()
scaler.fit(X)
X_scaled = scaler.transform(X)
# n_components=2表示将特征降低到2维
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
print(pca.explained_variance_ratio_)
```

```
[0.67380995 0.30247819]
```

4、将降维后特征可视化，横纵坐标代表两个主成分，颜色代表结果标签分类，即可根据主成分进行后续分析、建模

```
fig1 = plt.figure(figsize=(6,5))
plt.scatter(X_pca[:,0], X_pca[:,1],c=Y,s=20)
plt.xlim(-3, 3)
plt.ylim(-3, 3)
plt.xticks()
plt.yticks()
plt.show()
```



以上PCA主成分分析就讲完了，本文进行了样本点在超平面的投影尽可能分开的推导原理阐述，大家感兴趣的可以研究另一种等价推导，即样本点到超平面的距离最近。

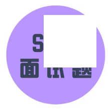
推荐阅读

实战丨你还不会用聚类模型(k-means)做数据分析?

2021-01-12

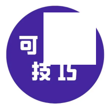
SQL面试题：如何分析平台业务?

2021-01-11



教你做超惊艳的南丁格尔玫瑰图

2021-01-10



爱数据原统计网推荐搜索

数据分析丨机器学习丨可视化丨Python

- END -