

sklearn因子分析 (python)

原创

guang_mang

于 2017-12-07 21:43:51 发布

33456

收藏

69

版权

分类专栏：


机器学习

 文章标签：

数学

math

python



机器学习 专栏收录该内容

0 订阅 11 篇文章

因子分析

因子分析（Factor Analysis）是指研究从变量群中提取共性因子的统计技术，这里的共性因子指的是不同变量之间内在的隐藏因子。例如，一个学生的英语、数学、语文成绩都很好，那么潜在的共性因子可能是智力水平高。因此，因子分析的过程其实是寻找共性因子和个性因子并得到最优解释的过程。

因子分析有两个核心问题：一是如何构造因子变量，二是如何对因子变量进行命名解释。因子分析有下面4个基本步骤：

过程

1、确定原有若干变量是否适合于因子分析。因子分析的基本逻辑是从原始变量中构造出少数几个具有代表意义的因子变量，这就要求原有变量之间要具有比较强的相关性，否则，因子分析将无法提取变量间的“共性特征”（变量间没有共性还如何提取共性？）。实际应用时，可以使用相关性矩阵进行验证，如果相关系数小于0.3，那么变量间的共性较小，不适合使用因子分析（查看变量间的相关性，也就是共有特征是否有必要提取共性）

```
import pandas as pd
mydata = pd.read_csv(r'D:\PythonDDD\datafiles\data.csv')
mydata.describe()

import math
df = pd.DataFrame(mydata)
df['行驶里程1'] = df['行驶里程'].apply(lambda x: math.log(x))
df.boxplot(column = '行驶里程1', by='车号')#查看车号的行驶里程
df.boxplot(column = '平均时速', by='车号')#查看车号的平均时速

#样本离差矩阵
import numpy as np
mydata = mydata.drop('车号', 1)
mydata = mydata.drop('行驶里程1', 1)
mydata_mean = mydata.mean()
E = np.mat(np.zeros((14, 14)))
for i in range(len(mydata)):
    E += (mydata.iloc[i, :].reshape(14, 1) - mydata_mean.reshape(14, 1)) * (mydata.iloc[i, :].reshape(1, 14) - mydata_mean.reshape(1, 14))

#样本相关性矩阵
R = np.mat(np.zeros((14, 14)))
for i in range(14):
    for j in range(14):
        R[i, j] = E[i, j]/math.sqrt(E[i, i] * E[j, j])
```

或者

```
R1 = mydata.corr() #样本相关性矩阵
```

2、构造因子变量。因子分析中有多种确定因子变量的方法，如基于主成分模型的主成分分析法和基于因子分析模型的主轴因子法、极大似然法、最小二乘法等。其中基于主成分模型的主成分分析法是使用最多的因子分析方法之一。（不同方法不同效果）

```
import numpy.linalg as nlg
eig_value, eigvector = nlg.eig(R)#求矩阵R的全部特征值，构成向量E。
print(eig_value, eigvector)
eig = pd.DataFrame()
```

```
eig['names'] = mydata.columns | eig['eig_value'] = eig_value
eig.sort_values('eig_value', ascending=False, inplace=True)

#求因子模型的因子载荷阵，寻找公共因子个数m
for m in range(1, 14):
    if eig['eig_value'][:m].sum()/eig['eig_value'].sum() >= 0.8:
        print(m)
        break

#因子载荷矩阵，只是因子，
A = np.mat(np.zeros((14, 6)))
A[:,0]=math.sqrt(eig_value[0])*eigvector[:,0]
A[:,1]=math.sqrt(eig_value[1])*eigvector[:,1]
A[:,2]=math.sqrt(eig_value[2])*eigvector[:,2]
A[:,3]=math.sqrt(eig_value[3])*eigvector[:,3]
A[:,4]=math.sqrt(eig_value[4])*eigvector[:,4]
A[:,5]=math.sqrt(eig_value[5])*eigvector[:,5]

a=pd.DataFrame(A)
a.columns=['factor1','factor2','factor3','factor4','factor5','factor6']
```

3、利用旋转使得因子变量更具有可解释性。在实际分析工作中，主要是因子分析得到因子和原变量的关系，从而对新的因子能够进行命名和解释，否则其不具有可解释性的前提下对比PCA就没有明显的可解释价值。

4、计算因子变量的得分。计算因子得分是因子分析的最后一步，因子变量确定以后，对每一样本数据，希望得到它们在不同因子上的具体数据值，这些数值就是因子得分，它和原变量的得分相对应。（生成新的数据）

```
from sklearn.cluster import KMeans

for i in range(2, 10):
    clf = KMeans(n_clusters=i)
    clf.fit(train_data)
    # print(clf.cluster_centers_)#类中心
    print(i, clf.inertia_)#用来评估簇的个数是否合适，距离越小说明簇分的越好，选取临界点的簇个数
```