主成分分析与因子分析

主成分分析

1.基本原理

设 $F_i(i=1,2,\ldots,m)$ 表示第 i 个主成分，且

$$F_i=c_{i1}x_1+c_{i2}x_2+\cdots+c_{im}x_m,$$

其中 $\sum\limits_{j=1}^{m}c_{ij}^2=1$ ，$C_i=[c_{i1},\ c_{i2},\ \ldots,\ c_{im}]^T$ 使 $Var(F_i)$ 达到最大，且 C_i 之间相互正交。

设有 n 个研究对象， m 个指标变量 $x_1,x_2,\ldots,x_m$ ，第 i 个对象关于第 j 个指标取值为 a_{ij} ，构造数据矩阵 $A=(a_{ij})_{n\times m}$ ，
1）对原来的 m 个指标进行标准化，得到标准化指标变量 $y_j(j=1,2,\ldots,m)$ ，对应的，得到标准化的数据矩阵 $B=(b_{ij})_{n\times m}$ ；
2）根据标准化的数据矩阵 B 求出相关系数矩阵 $R=(r_{ij})_{m\times m}$ ；
3）计算相关系数矩阵 R 的特征值 $\lambda_1\ge\lambda_2\ge\cdots\ge\lambda_m$ ，及对应的标准化正交化特征向量 $u_1,u_2,\ldots,u_m$ ，其中 $u_j=[\ u_1,\ u_2,\ \ldots,\ u_m\ ]^T$ ，由特征向量组成 p 个新的指标变量

$$\begin{cases}F_1=u_{11}y_1+u_{21}y_2+\cdots+u_{m1}y_m,\\F_2=u_{12}y_1+u_{22}y_2+\cdots+u_{m2}y_m,\\\qquad\qquad\vdots\\F_m=u_{1m}y_1+u_{2m}y_2+\cdots+u_{mm}y_m,\end{cases}$$

其中 F_j 为第 j 主成分；
4）计算主成分贡献率及累计贡献率，主成分 F_j 的贡献率为

$$w_j=\frac{\lambda_j}{\sum\limits_{k=1}^{m}\lambda_k},\quad j=1,2,\ldots,m,$$

前 i 个主成分的累计贡献率为

$$\frac{\sum\limits_{k=1}^{i}\lambda_k}{\sum\limits_{k=1}^{m}\lambda_k}$$

一般取累计贡献率达 85% 以上的特征值对应的主成分。

2.Python求解

```
from sklearn.decomposition import PCA
from scipy.stats import zscore
import numpy as np

a = np.loadtxt('data.txt')
b = zscore(a, ddof=1)  # 数据标准化
md = PCA().fit(b)  # 主成分分析
md.explained_variance_  # 提取特征值
md.explained_variance_raito_  # 提取各主成分贡献率
xs1 = md.components_  # 提取各主成分系数
check = xs1.sum(axis=1, keepdims=True)  # 计算各个主成分系数的和
xs2 = xs1 * np.sign(check)  # 调整主成分系数，和为负时乘以-1
```

因子分析

1.基本理论

设有 n 个研究对象， m 个指标变量 $x_1, x_2, \ldots, x_m$ ，将指标变量分解为

$$\begin{cases} x_1 = \mu_1 + a_{11}f_1 + a_{12}f_2 + \cdots + a_{1p}f_p + \varepsilon_1, \\ x_2 = \mu_2 + a_{21}f_1 + a_{22}f_2 + \cdots + a_{2p}f_p + \varepsilon_2, \\ \qquad\qquad\qquad \vdots \\ x_m = \mu_m + a_{m1}f_1 + a_{m2}f_2 + \cdots + a_{mp}f_p + \varepsilon_m, \end{cases}$$

简记为

$$x = \mu + Af + \varepsilon,$$

其中 $x = [\, x_1,\ x_2,\ \ldots,\ x_m \,]^T$ 为指标变量，$\mu = [\, \mu_1,\ \mu_2,\ \ldots,\ \mu_m \,]^T$ 为 x 的期望向量；
$f = [\, f_1,\ f_2,\ \ldots\ f_p \,]^T$ 为公共因子向量，$\varepsilon = [\, \varepsilon_1,\ \varepsilon_2,\ \ldots,\ \varepsilon_m \,]^T$ 为特殊因子向量；
$A = (a_{ij})_{m \times p}$ 为因子载荷矩阵， a_{ij} 是变量 x_i 在公共因子 f_j 上的载荷，反应 f_j 对 x_i 的重要程度。

设 x 为标准化变量，其相关系数矩阵为 $R = (r_{ij})_{m \times m}$ ， $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ 为其特征值，$u_1, u_2, \ldots, u_m$ 为对应的标准正交化特征向量，$p < m$ ，则因子载荷矩阵为

$$A = [\, \sqrt{\lambda_1}u_1,\ \sqrt{\lambda_2}u_2,\ \ldots,\ \sqrt{\lambda_p}u_p \,]$$

f_j 的贡献率为 λ_j / m ，其中 λ_j 为相关系数矩阵的第 j 大特征值。

2.Python求解

```python
from factor_analyzer import FactorAnalyzer as FA

# n_factors为因子数量，rotation为因子旋转方法
fa = FA(n_factors=3, rotation=None).fit(b)
A = fa.loading_  # 提取载荷矩阵
```