因子分析

TUTU

因子分析基本概念

♣ 基本思想：

根据相关性大小把变量分组，使得同组内的变量之间相关性较高，不同组的变量相关性较低。每组变量代表一个基本结构，这个基本结构称为因子。

♣ 与其他方法的不同：

- 与回归分析：因子分析中的因子是一个比较抽象的概念，而回归因子有非常明确的实际意义
- 与主成分分析：主成分分析仅仅是变量变换，而因子分析需要构造因子模型
  - 主成分分析：原始变量的线性组合表示新的综合变量，即主成分
  - 因子分析：潜在的假想变量和随机影响变量的线性组合表示原始变量

因子分析模型

♣ 设 $\boldsymbol{X}=(X_1,X_2,\cdots,X_p)^{\mathrm{T}}$ 是 p 维随机向量，则称 $\boldsymbol{X}$ 是有 $m(m\leq p)$ 个公因子的模型，若 $\boldsymbol{X}$ 能表示为

$$X_i=\sum_{j=1}^{m}a_{ij}f_j+\varepsilon_i,(i=1,2,\cdots,p)$$

也可写成矩阵形式：$\boldsymbol{X}=\boldsymbol{A}\boldsymbol{f}+\boldsymbol{\varepsilon}$

a_{ij} 为因子载荷；$\boldsymbol{A}$ 为因子载荷矩阵；$\boldsymbol{f}$ 为公因子向量；$\boldsymbol{\varepsilon}$ 为特殊因子向量

♣ 因子模型满足：

- $E(\boldsymbol{f})=\mathbf{0}$；$\mathrm{Cov}(\boldsymbol{f})=\boldsymbol{I}_m$；$\mathrm{Cov}(\boldsymbol{f},\boldsymbol{\varepsilon})=\mathbf{0}$
- $E(\boldsymbol{\varepsilon})=\mathbf{0}$；$\mathrm{Cov}(\boldsymbol{\varepsilon})=\mathrm{diag}(\sigma_1^2,\cdots,\sigma_p^2)$
- $\boldsymbol{\varepsilon}\sim N(0,\sigma_i^2)$

参数的统计意义

♣ 参数的统计意义：

- $\mathrm{Cov}(X_i, f_j) = a_{ij}$ (X_i 在第 j 个公因子上的权)
- 因子载荷不是唯一的
- 变量 X_i 的共同度 (对 $\boldsymbol{f}$ 的共同依赖程度)：$h_i^2 = \sum_{j=1}^{m} a_{ij}^2$ (行元素平方和，h_i^2 越大，模型效果越好)
- $1 = h_i^2 + \sigma_i^2$；$\hat{\sigma}_i^2 = s_{ii} - \sum_{j=1}^{m} a_{ij}^2$
- f_j 对变量 $\boldsymbol{X}$ 的方差贡献和：$g_j^2 = \sum_{i=1}^{p} a_{ij}^2$ (列元素平方和)
- f_j 的方差贡献率：$\dfrac{g_j^2}{\sum_{i=1}^{p} \mathrm{Var}(X_i)}$
- $f_1, f_2, \cdots, f_m$ 的累计方差贡献率：$\dfrac{\sum_{j=1}^{m} g_j^2}{\sum_{i=1}^{p} \mathrm{Var}(X_i)}$

因子载荷矩阵的估计

♣ 设 $\boldsymbol{X}$ 的协差阵 $\boldsymbol{\Sigma}=(s_{ij})_{p\times p}$(正定) 的特征根 $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_p>0$,

对应的正交单位化特征向量为 $\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_p$，其中

$\boldsymbol{e}_i=(e_{1i},e_{2i},\cdots,e_{pi})^{\mathrm{T}}$，取累计贡献率 ≥ 0.85 的前 m 个公因子 (对应

的因子载荷阵为 $\hat{\boldsymbol{A}}_{p\times m}$)，则 $\boldsymbol{\Sigma}\approx\hat{\boldsymbol{A}}\hat{\boldsymbol{A}}^{\mathrm{T}}+\boldsymbol{\Psi}_{\boldsymbol{\varepsilon}}$,

其中 $\hat{\boldsymbol{A}}=\left(\sqrt{\lambda_1}\boldsymbol{e}_1,\sqrt{\lambda_2}\boldsymbol{e}_2,\cdots,\sqrt{\lambda_m}\boldsymbol{e}_m\right)$,

$\boldsymbol{\Psi}_{\boldsymbol{\varepsilon}}$ 的估计为 $\hat{\boldsymbol{\Psi}}_{\boldsymbol{\varepsilon}}=\mathrm{diag}(\hat{\psi}_{11},\hat{\psi}_{22},\cdots,\hat{\psi}_{pp})=\mathrm{diag}(\sigma_1^2,\cdots,\sigma_p^2)$,

且 $\hat{\sigma}_i=s_{ii}-\sum_{j=1}^{m}a_{ij}^2$

因子旋转

♣ 目的：

使因子载荷阵的结构简化，使载荷矩阵每列或行的元素平方值向 0 和 1 两极分化

♣ 旋转方法：

- 正交旋转：由因子载荷矩阵 A 右乘一正交阵而得到，经过旋转后的新的公因子仍然保持彼此独立的性质；方法：方差最大法和四次方最大法
- 斜交旋转：放弃了因子之间彼此独立这个限制，可达到更简洁的形式，实际意义也更容易解释

因子旋转

♣ 方差最大法：

1. 第一轮旋转，每次取两个，全部配对旋转，变换共需进行$\dfrac{m(m-1)}{2}$次
   - 旋转矩阵$T=\begin{pmatrix}\cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi\end{pmatrix}$，$B=AT$
   - 各列元素方差总和：
     $$V=\left[\frac{1}{p}\sum_{i=1}^{p}(b_{i1}^2)^2-\left(\frac{1}{p}\sum_{i=1}^{p}b_{i1}^2\right)\right]+\left[\frac{1}{p}\sum_{i=1}^{p}(b_{i2}^2)^2-\left(\frac{1}{p}\sum_{i=1}^{p}b_{i2}^2\right)\right]$$
   - $\dfrac{\partial V}{\partial\varphi}=0\Rightarrow\varphi$
2. 对第一轮旋转所得结果用上述方法继续进行旋转，得到第二轮旋转结果。每一次旋转后，矩阵各列平方的相对方差之和总会比上一次有所增加
3. 当总方差的改变不大时，就可以停止旋转

因子旋转

♣ 四次方最大旋转法：

- 从简化载荷矩阵的行出发，通过旋转初始因子，使每个变量只在一个因子上有较高的载荷，而在其它的因子上尽可能低的载荷
- 使因子载荷矩阵中每一行的因子载荷平方的方差达到最大
- 简化准则：$\max Q = \sum_{i=1}^{p} \sum_{j=1}^{m} b_{ij}^{4}$

♣ 因子旋转的特征：

- 旋转后因子的共同度没有发生变化
- 旋转后公共因子的方差贡献发生了变化

因子得分

♣ 因子得分函数:

$$F_j = \beta_{j1} X_1 + \cdots + \beta_{jp} X_p, j = 1, \cdots, m$$

♣ 回归法估计:

- $\hat{F}_j = b_j' X$
- $a_{ij} = \mathrm{Cov}(X_i, f_j) = (r_{i1}, r_{i2}, \cdots, r_{ip}) \cdot \begin{pmatrix} b_{j1} \\ b_{j2} \\ \vdots \\ b_{jp} \end{pmatrix}$
- $\boldsymbol{R}\boldsymbol{b}_j = \boldsymbol{a}_j \Rightarrow \boldsymbol{b}_j = \boldsymbol{R}^{-1}\boldsymbol{a}_j$
- $F = A^{\mathrm{T}} R^{-1} X$

因子分析

♣ 因子得分的基本步骤：

1. 选择分析的变量
2. 计算所选原始变量的相关系数矩阵
3. 提取公共因子
4. 因子旋转
5. 计算因子得分
6. 用因子分析方法进行综合评价：权为 $\alpha_j = \dfrac{g_j^2}{p}$

$$F = \alpha_1 F_1 + \alpha_2 F_2 + \cdots + \alpha_m F_m$$

因子分析 SAS 代码

♣ SAS 代码：

```
/*因子分析，选定5个因子，r=v是方差最大，r=q是四次方最大*/
proc factor data=yourdata method=prin r=v n=5 out=a1 outstat=
    stat1 reorder;
run;

/*计算因子得分*/
data a2;
set a1;
f=(5.6327*factor1+2.7072*factor2+2.2692*factor3+1.3137*factor4
    +1.0431*factor5)/15;
keep f factor1 factor2 factor3 factor4 factor5;
run;

/*按综合因子得分降序排列*/
proc sort data=a2 out=a3;
by descending f;
run;
```