

Análisis factorial

Daniel Czarniewicz

Análisis Factorial

Se parte de una matriz de datos de la forma:

$$\mathbf{X}_{I \times J} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1J} \\ x_{21} & x_{22} & \dots & x_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ x_{I1} & x_{I2} & \dots & x_{IJ} \end{bmatrix}$$

A partir de ella se definen dos espacios:

1. El espacio definido por la nube de las N_I filas, el cual está incluido en \mathbb{R}^J (dado que cada fila constituye un vector con J componentes).
2. El espacio definido por la nube de las N_J columnas, el cual está incluido en \mathbb{R}^I (dado que cada columna constituye un vector con I componentes).

El objetivo principal de un análisis factorial es eliminar la información redundante (reducción de dimensionalidad). Los resultados de un análisis factorial son: los ejes de inercia, y las coordenadas de los puntos sobre dichos ejes (llamados factores).

Desarrollo por N_I

Trabajamos primero con la nube de puntos N_I , definida por las filas de la matriz $\mathbf{X}_{I \times J}$. Cada individuo está representado por un vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iJ})' \in \mathbb{R}^J$. El objetivo es encontrar el conjunto de ejes ortonormados que maximicen la inercia proyectada sobre ellos. Al conjunto de dichas coordenadas sobre un eje de inercia se le llama *factor*.

Se definen las matrices diagonales $\mathbf{M}_{J \times J}$ y $\mathbf{D}_{I \times I}$ tales que:

$$\mathbf{M}_{J \times J} = \begin{bmatrix} m_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & m_J \end{bmatrix} \quad \mathbf{D}_{I \times I} = \begin{bmatrix} p_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & p_I \end{bmatrix}$$

Estas matrices actuarán de pesos o métricas en cada análisis según la siguiente tabla:

	Espacio	Métrica	Pesos
Nube de filas	\mathbb{R}^J	\mathbf{M}	\mathbf{D}
Nube de columnas	\mathbb{R}^I	\mathbf{D}	\mathbf{M}

Dado que \mathbf{M} es diagonal, la distancia entre dos puntos i y k de N_I se calcula como:

$$d^2(i, k) = \sum_{j=1}^J (x_{ij} - x_{kj})^2 m_j$$

Sea \mathbf{u}_s un vector director de un eje cualquiera de \mathbb{R}^J . Definimos el vector de coordenadas proyectadas sobre \mathbf{u}_s , al que llamaremos $\mathbf{F}_s(i)$, como:

$$\mathbf{F}_s(i) = \mathbf{x}'_i \mathbf{M} \mathbf{u}_s$$

Nótese que \mathbf{x}'_i es de forma $1 \times J$, \mathbf{M} es una matriz de forma $J \times J$, \mathbf{u}_s es de forma $J \times 1$. $\mathbf{F}_s(i)$ es un escalar. Este número representa la proyección del i -ésimo individuo sobre el eje de inercia \mathbf{u}_s . Visto en forma matricial:

$$\mathbf{F}_s = \mathbf{X} \mathbf{M} \mathbf{u}_s \quad \text{cons} = 1, \dots, J$$

Nótenes que omitimos la mención al i -ésimo individuo en \mathbf{F}_s . Esto se debe a que \mathbf{X} es una matriz de tamaño $I \times J$, por lo que \mathbf{F}_s es un vector de dimensión $I \times 1$, donde su i -ésima entrada es la proyección del i -ésimo individuo sobre el eje de inercia \mathbf{u}_s .

Podemos entonces definir la *inercia de la nube proyectada* como:

$$\text{inercia} = \mathbf{F}'_s \mathbf{D} \mathbf{F}_s$$

la cual podemos escribir, utilizando la definición de \mathbf{F}_s , como:

$$\text{inercia} = \mathbf{u}'_s \mathbf{M}' \mathbf{X}' \mathbf{D} \mathbf{X} \mathbf{M} \mathbf{u}_s$$

El objetivo, tal como fuera planteado, es hallar el eje $\mathbf{u} \in \mathbb{R}^J$ unitario en la métrica \mathbf{M} , que maximice la inercia. Es decir,

$$\max_u \{\text{inercia}\} = \max_u \{\mathbf{u}'_s \mathbf{M}' \mathbf{X}' \mathbf{D} \mathbf{X} \mathbf{M} \mathbf{u}_s\}$$

Si la métrica es euclídea, \mathbf{M} es la matriz identidad, y el problema se reduce a hallar un vector \mathbf{u} tal que $\mathbf{u}'\mathbf{u} = 1$ que maximice la inercia:

$$\max_u \{\text{inercia}\} = \max_u \{\mathbf{u}' \mathbf{X}' \mathbf{D} \mathbf{X} \mathbf{u}\}$$

Dado que $\mathbf{X}' \mathbf{D} \mathbf{X}$ es simétrica:

- $\mathbf{X}'\mathbf{D}\mathbf{X}$ es diagonalizable.
- $\exists \mathbf{U}$ matriz ortogonal (es decir, $\mathbf{U}'\mathbf{U} = \mathbf{U}\mathbf{U}' = \mathbf{I}$).
- $\mathbf{U} = ((u_j))$ son los vectores propios asociados al valor propio λ_j .
- se define $\mathbf{\Lambda}$ como la matriz diagonal de valores propios: $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_J)$ tales que

$$\mathbf{U}'\mathbf{X}'\mathbf{D}\mathbf{X}\mathbf{U} = \mathbf{\Lambda} \Rightarrow \mathbf{X}'\mathbf{D}\mathbf{X} = \mathbf{U}'\mathbf{\Lambda}\mathbf{U}$$

- sus vectores propios forman una base ortonormal en \mathbf{R}^J .

Desarrollo por N_J

Interpretación de resultados

Análisis de Componentes Principales

Partimos de una matriz de datos donde a n individuos se le miden p variables cuantitativas. Cada individuo puede entonces ser representado por un vector en \mathbb{R}^p , y cada variable puede ser representada por un vector en \mathbb{R}^n . El objetivo es reducir la cantidad de dimensiones necesarias para representar cada individuo, sin perder información (i.e. varianza) sobre ellos.

Construcción de los ejes factoriales

Maximizar la inercia capturada

En esta visión del método, lo que se quiere es encontrar las combinaciones lineales de las variables originales que representan los datos de forma correcta. Se utilizan tantos ejes direccionales (nuevas variables) como sea necesario (máximo p). El primero de ellos se encuentra resolviendo el sistema de ecuaciones

$$\mathbf{z}_1 = \mathbf{X}\phi_1$$

de forma de maximizar la varianza capturada por él. Es decir, de todas las posibles combinaciones lineales, buscamos aquella sobre la cual las proyecciones de los puntos originales estén más separadas. A los coeficientes $\phi_1 = (\phi_{11}, \phi_{21}, \dots, \phi_{p1})'$ les llamamos *loadings* de la primer componente principal, y les imponemos que $\sum_{j=1}^p \phi_j^2 = 1$. Esto se debe a que, de no hacerlo, podríamos aumentarlos arbitrariamente, para aumentar así la varianza.

Para calcular los loadings, primero debemos estandarizar las variables. Luego debemos resolver el problema

$$\max_{\phi_1} = \left\{ \frac{1}{n} \sum_{i=1}^n z_{i1}^2 \right\} = \max_{\phi_1} = \left\{ \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^p \phi_{j1} x_{ij} \right)^2 \right\}$$

$$\text{sujeto a } \sum_{j=1}^p \phi_j^2 = 1$$

Al vector $\mathbf{z}_1 = (z_{11}, z_{21}, \dots, z_{n1})'$ le llamamos vector de *scores* de la primera componente principal. El score es el valor que la i -ésima observación toma en la proyección sobre el eje encontrado.

Para hallar la segunda componente principal, debemos hallar los scores que maximizan la varianza, siendo el nuevo vector ortogonal al vector \mathbf{z}_1 .

$$\max_{\phi_2} = \left\{ \frac{1}{n} \sum_{i=1}^n z_{i2}^2 \right\} = \max_{\phi_2} = \left\{ \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^p \phi_{j2} x_{ij} \right)^2 \right\}$$

$$\text{sujeto a } \sum_{j=1}^p \phi_j^2 = 1 \quad \text{y} \quad \langle \mathbf{z}_1, \mathbf{z}_2 \rangle = 0$$

Se procede de esta forma hasta hallar todos los ejes \mathbf{z}_j , hasta un máximo de p ejes.

Minimizar distancias al eje de proyección

Otra forma de resolver el problema es minimizando la suma de las distancias al cuadrado entre los puntos originales y sus proyecciones sobre los ejes. Cuando hablamos de distancias, en este caso, nos referimos a las proyecciones (en sentido algebraico). Es decir, la distancia medida de forma tal que esta sea perpendicular al eje de inercia, no al eje x_1 o x_2 . Si llamamos r_i a la distancia del individuo i al eje de proyección a_1 , utilizamos el teorema de Pitágoras para ver que:

$$r_i^2 = \|\mathbf{x}_i\|^2 - \|z_i \mathbf{a}_1\|^2 = \mathbf{x}_i' \mathbf{x}_i - (z_i)^2 \|\mathbf{a}_1\|^2 = \mathbf{x}_i' \mathbf{x}_i - z_i^2$$

Nuestro problema se puede plantear entonces como:

$$\min \left\{ \sum_{i=1}^n r_i^2 \right\} = \min \left\{ \sum_{i=1}^n \mathbf{x}_i' \mathbf{x}_i - \sum_{i=1}^n z_i^2 \right\} = \sum_{i=1}^n \mathbf{x}_i' \mathbf{x}_i - \max \left\{ \sum_{i=1}^n z_i^2 \right\}$$

donde utilizamos que $\sum_{i=1}^n \mathbf{x}_i' \mathbf{x}_i$ está fija.

Dado que la matriz de datos está centrada, $\sum_{i=1}^n z_i^2$ es la varianza muestral de los datos proyectados. Por lo tanto, queremos hallar los ejes incorrelados que resuelven:

$$\mathbf{z}_j = \mathbf{X} \mathbf{a}_j$$

Nuevamente, impondremos que $\mathbf{a}_j' \mathbf{a}_j = 1$.

Por último, notemos también que, dado que la matriz de datos está centrada,

$$\begin{aligned}
 \mathbf{Var}(\mathbf{z}_1) &= \mathbf{Var}(\mathbf{X}\mathbf{a}_1) \\
 &= \frac{1}{n} \mathbf{z}'_1 \mathbf{z}_1 \\
 &= \frac{1}{n} (\mathbf{X}\mathbf{a}_1)' (\mathbf{X}\mathbf{a}_1) \\
 &= \frac{1}{n} \mathbf{a}'_1 \mathbf{X}' \mathbf{X} \mathbf{a}_1 \\
 &= \mathbf{a}'_1 \left(\frac{1}{n} \mathbf{X}' \mathbf{X} \right) \mathbf{a}_1 \\
 &= \mathbf{a}'_1 \boldsymbol{\Sigma}_{\mathbf{X}} \mathbf{a}_1
 \end{aligned}$$

Por lo tanto, el problema de maximización lo podemos resolver con el siguiente Lagrangeano:

$$\mathcal{L}_1(\mathbf{a}_1) = \mathbf{a}'_1 \boldsymbol{\Sigma}_{\mathbf{X}} \mathbf{a}_1 - \lambda_1 (\mathbf{a}'_1 \mathbf{a}_1 - 1)$$

Cuya CPO está dada por:

$$\frac{\partial \mathcal{L}_1(\mathbf{a}_1)}{\partial \mathbf{a}'_1} = 2\boldsymbol{\Sigma}_{\mathbf{X}} \mathbf{a}_1 - 2\lambda_1 \mathbf{I}_p \mathbf{a}_1 = \mathbf{0}_p \Rightarrow (\boldsymbol{\Sigma}_{\mathbf{X}} - \lambda_1 \mathbf{I}_p) \mathbf{a}_1 = \mathbf{0}_p$$

Dado que lo que estamos buscando es un $\mathbf{a}_1 \neq \mathbf{0}_p$, resolver la CPO equivale a resolver

$$\det(\boldsymbol{\Sigma}_{\mathbf{X}} - \lambda_1 \mathbf{I}_p) = 0$$

por lo tanto, λ es el valor propio de $\boldsymbol{\Sigma}_{\mathbf{X}}$ asociado al vector propio \mathbf{a}_1 . Volviendo al análisis de la varianza, tenemos que:

$$\mathbf{Var}(\mathbf{z}_1) = \mathbf{a}'_1 \boldsymbol{\Sigma}_{\mathbf{X}} \mathbf{a}_1 = \mathbf{a}'_1 \lambda_1 \mathbf{a}_1 = \lambda_1 \mathbf{a}'_1 \mathbf{a}_1 = \lambda_1$$

Concluimos entonces que, para maximizar la varianza (o minimizar las distancias r_i), debemos tomar el mayor valor propio asociado a $\boldsymbol{\Sigma}_{\mathbf{X}}$, λ_1 , y su correspondiente vector propio asociado, \mathbf{a}_1 .

Luego, para hallar \mathbf{z}_2 , procedemos de forma análoga, pero agregando la restricción adicional $\mathbf{Cov}(\mathbf{a}_1, \mathbf{a}_2) = 0$. Por lo tanto, el Lagrangeano asociado al problema será:

$$\mathcal{L}_2(\mathbf{a}_2) = \mathbf{a}'_2 \boldsymbol{\Sigma}_{\mathbf{X}} \mathbf{a}_2 - \lambda_2 (\mathbf{a}'_2 \mathbf{a}_2 - 1) - \delta (\mathbf{a}'_2 \mathbf{a}_1 - 0)$$

Su CPO será:

$$\begin{aligned}
 \frac{\partial \mathcal{L}_2(\mathbf{a}_2)}{\partial \mathbf{a}_2'} &= 2\Sigma_{\mathbf{X}}\mathbf{a}_2 - 2\lambda_2\mathbf{a}_2 - \delta\mathbf{a}_1 \\
 &= 2\mathbf{a}_1'\Sigma_{\mathbf{X}}\mathbf{a}_2 - 2\lambda_2\mathbf{a}_1'\mathbf{a}_2 - \delta\mathbf{a}_1'\mathbf{a}_1 \\
 &= 2\mathbf{a}_1'\Sigma_{\mathbf{X}}\mathbf{a}_2 - 2\lambda_2\underbrace{\mathbf{a}_1'\mathbf{a}_2}_{=0} - \delta\underbrace{\mathbf{a}_1'\mathbf{a}_1}_{=1} \\
 &= 2\mathbf{a}_1'\Sigma_{\mathbf{X}}\mathbf{a}_2 - \delta = \mathbf{0}_p
 \end{aligned}$$

Tenemos entonces que el máximo se halla cuando

$$\delta = 2\mathbf{a}_1'\Sigma_{\mathbf{X}}\mathbf{a}_2 = 2\mathbf{a}_2'\Sigma_{\mathbf{X}}\mathbf{a}_1$$

por lo tanto,

$$\frac{\partial \mathcal{L}_2(\mathbf{a}_2)}{\partial \mathbf{a}_2'} = 2\Sigma_{\mathbf{X}}\mathbf{a}_2 - 2\lambda_2\mathbf{I}_p\mathbf{a}_2 = \mathbf{0}_p \Rightarrow (\Sigma_{\mathbf{X}} - \lambda_2\mathbf{I}_p)\mathbf{a}_2 = \mathbf{0}_p$$

lo cual implica elegir el segundo mayor valor propio de $\Sigma_{\mathbf{X}}$, y \mathbf{a}_2 será el vector propio asociado a dicho valor propio.

Este procedimiento se repite p veces, restringiendo a que cada nuevo vector \mathbf{z}_j sea ortogonal al anterior. Se obtiene así la matriz ortogonal $\mathbf{A}_{p \times p} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p)$.

Relaciones entre ejes

Elección de la cantidad de ejes

Análisis de Correspondencias Simples

Análisis de Correspondencias Múltiple

Referencias

Beygelzimer, Alina, Sham Kakadet, John Langford, Sunil Arya, David Mount, and Shengqiao Li. 2018. *FNN: Fast Nearest Neighbor Search Algorithms and Applications*. <https://CRAN.R-project.org/package=FNN>.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 112. Springer.

R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.

Rencher, Alvin C. 1998. *Multivariate Statistical Inference and Applications*. Wiley New York.

Wasserman, Larry. 2007. *All of Nonparametric Statistics*. Springer, New York.

Wickham, Hadley. 2017. *Tidyverse: Easily Install and Load the 'Tidyverse'*. <https://CRAN.R-project.org/package=tidyverse>.