S1_LSC1_PCA

October 3, 2021

ESPACIO PARA BANNER DE LA MAESTRIA -

1 Análisis de Componentes Principales. Fundamentos Teóricos

En este *cuaderno* estudiaremos el Análisis de Componentes Principales y como podemos utilizarlos para reducir la dimensionalidad de datos. El objetivo de este *cuaderno* es que aprenda que son los componentes principales, que sea capaz de reconocer las características y el funcionamiento del algoritmo de componentes principales, y como construirlo e implementarlo.

NO es necesario editar el archivo o hacer una entrega. Los ejemplos contienen celdas con código ejecutable (`en gris`), que podés modificarlas libremente. Esta puede ser una buena forma de aprender nuevas funcionalidades del *cuaderno*, o experimentar variaciones en los códigos de ejemplo.

1.1 Introducción

Los grandes conjuntos de datos son cada vez más comunes y, a menudo, difíciles de interpretar. Para interpretar tales conjuntos de datos, se requieren métodos para reducir drásticamente su dimensionalidad de una manera interpretable, de modo que se conserve la mayor parte de la información de los datos.

El análisis de componentes principales, o PCA por sus siglas en inglés, es una técnica de aprendizaje no supervisado que permite reducir la dimensionalidad de tales conjuntos de datos, aumentando la interpretabilidad, pero al mismo tiempo minimizando la pérdida de información. Se han desarrollado muchas técnicas para este propósito, pero el análisis de componentes principales es uno de los más antiguos y más utilizados. Su idea es simple: reducir la dimensionalidad de un conjunto de datos, mientras se preserva la mayor "variabilidad" posible.

A modo de ejemplo, supongamos que disponemos de una base de datos de 100 clientes de un banco que cuenta con información sobre 10 variables que miden riesgo crediticio de los clientes. Estas variables contienen información sobre transacciones bancarias, historia crediticia, etc. La pregunta que queremos responder es si es posible construir una suerte de índice que permita condensar todas estas dimensiones en una sola que resuma el riesgo crediticio.

La primer respuesta obvia sería construir un índice compuesto por el promedio de estas 10 variables. Sin embargo, todas estas variables pueden contener información redundante o no tener la misma importancia. El problema entonces se puede pensar como cuál es la mejor forma de construir este índice combinando linealmente estas variables.

1.2 ¿Qué son los componentes principales?

Supongamos que queremos visualizar n observaciones de las cuales tenemos k variables o atributos, representadas por $x_1, x_2, \ldots, x_k$ como parte de un análisis descriptivo. Siguiendo el ejemplo anterior n serian los 100 clientes y k las 10 variables que miden el riesgo crediticio. Una forma de hacer el análisis descriptivo es haciendo diagramas de dispersión para las n observaciones examinando dos variables por vez. El problema acá se hace evidente, si tenemos 10 variables tendríamos que examinar 45 gráficas $\left(\begin{pmatrix} 10 \\ 2 \end{pmatrix} = \frac{10 \times 9}{2} = 45 \right)$

Si k es muy grande, como suele serlo en estas aplicaciones, sería imposible verlos a todos. Necesitamos un método que nos permita visualizar cuando la dimensión k es muy grande. El análisis de componentes principales es una herramienta que nos permite alcanzar este objetivo.

Intuitivamente, PCA plantea que cada observación vive en un espacio $k - dimensional$, pero que no todas estas dimensiones son igualmente informativas. PCA busca representar los datos es un espacio de menor dimensión, reteniendo la mayor cantidad de información posible. Entonces estas nuevas dimensiones encontradas por PCA, llamadas componentes, es una combinación lineal de las variables originales.

El primer componente de un conjunto de variables $X_1, X_2, \ldots, X_k$ es

$$f_1 = \delta_{11}x_1 + \delta_{12}x_2 + \cdots + \delta_{1k}x_k$$

Donde f_1 denota al primer componente principal y δ_{ij} se conocen como pesos o "loadings" del primer componente principal. Esta ecuación ilustra también el hecho de que el primer componente principal es una combinación lineal de las variables originales. La pregunta que naturalmente surge es: ¿Cómo se calculan estos componentes de forma tal que preserven la mayor información posible?

1.3 Cálculo de los componentes

1.3.1 Cálculo del primer componente

Formalmente, supongamos que X es una matriz $n \times k$ que contiene los datos, es decir, las n observaciones de las k variables. Asumimos que cada una de las variables en X estan centradas para tener media cero, y tiene una matriz de covarianza ($Var(X)$) denotada con S, que por definición es una matriz cuadrada de orden k.

La tarea del primer componente principal es encontrar la combinación lineal de las variables originales que maximiza la varianza, es decir, preservan la mayor información posible. El objetivo es crear un índice f_1 que tiene la siguiente forma:

$$f_1 = X\delta_1' \tag{1}$$
$$= \delta_{11}x_1 + \delta_{12}x_2 + \cdots + \delta_{1k}x_k \tag{2}$$

en donde δ_1 es un vector de K números reales ($\delta_1 = \delta_{11}, \ldots, \delta_{1k}$). El problema consiste en elegir δ_1 óptimamente. Por lo que, este indice va a ser la "mejor" combinación lineal de $x_1, x_2, \ldots, x_K$. Donde definimos como "mejor" a aquella combinación que maximiza la varianza. Dicho de otra

forma, vamos a buscar maximizar la varianza de forma tal que podamos reproducir de la mejor manera posible la variabilidad (información) original de las variables x_j.

Notando que

$$\begin{aligned} Var(f_1) &= Var(X\delta_1') && (3)\\ &= \delta_1 Var(X)\delta_1' && (4)\\ &= \delta_1 S\delta_1' && (5) \end{aligned}$$

El problema se reduce a elegir δ_1 de forma que maximice $Var(X\delta_1)$. Maximizar $\delta_1 S\delta_1'$ tiene como solución trivial llevar δ_1 a infinito ($\delta_1 \to \infty$). Para que esta maximización tenga solución en la practica se impone una restricción adicional que normaliza δ_1:

$$\delta_1\delta_1' = 1 \tag{6}$$

Esto restringe a que la suma del cuadrado de los pesos o "loadings" se igual a uno, ya que si lo restringimos a que sea un número arbitrariamente grande esto podría resultar en una varianza arbitrariamente grande.

El problema queda definido de la siguiente manera:

$$\begin{aligned} &\max_{\delta_1} \ \delta_1 S\delta_1' && (7)\\ &\text{sujeto a} && (8)\\ &\delta_1\delta_1' = 1 && (9) \end{aligned}$$

maximiza $\delta_1 S\delta_1'$ restringiendo que $\delta_1\delta_1' = 1$. Escribiendo el Lagrangiano,

$$\mathcal{L} = \delta_1 S\delta_1' + \lambda_1(1 - \delta_1\delta_1') \tag{10}$$

y maximizaremos esta expresión de la forma habitual derivando respecto a δ_1 e igualando a cero:

$$\frac{\partial \mathcal{L}}{\partial \delta_1} = S\delta_1' - \lambda_1\delta_1' = 0 \tag{11}$$

Reordenando:

$$S\delta_1' = \lambda_1\delta_1' \tag{12}$$

En el óptimo, δ_1 es el eigenvector correspondiente al eigenvalor λ. Premultiplicando la ecuación anterior por δ_1 y usando la restricción $\delta_1\delta_1' = 1$:

$$\delta_1 S \delta_1' = \lambda_1 \tag{13}$$

para maximizar $\delta_1 S \delta_1'$ debemos elegir λ_1 igual al máximo autovalor de S y δ_1 igual al eigenvalor correspondiente. Notando además que $\delta_1 S \delta_1' = Var(X\delta_1')$, el problema de encontrar la mejor combinación lineal que reproduce la variabilidad en X se reduce a encontrar el mayor autovalor de S y su correspondiente autovector.

Ejemplo cálculo primer componente Ilustremos en detalle el cálculo del primer componente principal en `Python` utilizando datos de la Encuesta Nacional de Presupuestos de los Hogares (ENPH) de Colombia, realizada por el DANE en 2017. Los datos contienen información estandarizada de los gastos promedio en Salud, Transporte, y Educación para 38 ciudades colombianas.

[1]:
```
#Cargamos las librerias a utilizar
import pandas as pd
import numpy as np

# Cargamos y visualizamos la primeras observaciones de los datos
gasto = pd.read_csv('Data/gasto_col_2017_norm.csv')
gasto = gasto.set_index("Ciudad")
gasto.head()
```

[1]:

Ciudad	Salud	Transporte	Educación
Arauca	-0.125062	-1.344088	-1.028321
Armenia	0.434314	0.691669	0.528711
Barrancabermeja	-0.752697	-0.093582	0.966586
Barranquilla	-0.859420	0.275332	0.503810
Bogotá	2.745217	2.313395	3.943969

La matriz de varianzas es:

[2]:
```
S = gasto.cov()
S
```

[2]:

	Salud	Transporte	Educación
Salud	1.000000	0.687662	0.644640
Transporte	0.687662	1.000000	0.541518
Educación	0.644640	0.541518	1.000000

Los valores propios son las raíces de la ecuación:

$$|S - \lambda I| = 0$$

y los obtenemos en `Python` con:

[3]:
```
eig_val, eig_vec = np.linalg.eig(S)
eig_val
```

[3]:
```
array([2.25116235, 0.28709315, 0.4617445 ])
```

El máximo eigenvalor es el primero y el vector propio asociado a este eigenvalor nos dara los pesos o "loadings" del primer componente principal:

[4]:
```
eig_vec=eig_vec.T
eig_vec[[0]]
```

[4]:
```
array([[0.60164121, 0.57198642, 0.55754766]])
```

El primer componente principal es:

$$f_1 = 0.60 \times Salud + 0.27 \times Transporte + 0.55 \times Educacin \tag{14}$$

Calculando este para cada una de las ciudades tenemos entonces un índice que reduce las tres variables en una:

[5]:
```
f1=gasto.dot(eig_vec[[0]].T) #calculamos f1
f1.columns = ['CP1'] #nombramos la columna del primer componente como CP1
f1.head()
```

[5]:

Ciudad	CP1
Arauca	-1.417381
Armenia	0.951708
Barrancabermeja	0.032537
Barranquilla	-0.078678
Bogotá	5.173817

Si ordenamos las ciudades según esta nueva variable construida (PCA), se puede notar que las ciudades quedan prácticamente ordenadas por su ingreso. La primera componente principal entonces esta redescubriendo el ingreso de estas ciudades.

[6]:
```
f1.sort_values(by='CP1', ascending=False).head()
```

[6]:

Ciudad	CP1
Bogotá	5.173817
Tunja	2.408399
Medellín y A.M.	2.213589
Manizales y A.M.	1.849532
Cali	1.638936

1.3.2 Cálculo del segundo componente principal

Luego de calcular el primer componente principal f_1, podemos encontrar también el segundo componente principal, f_2:

$$f_2 = X\delta_2' \tag{15}$$
$$= \delta_{21}x_1 + \delta_{22}x_2 + \cdots + \delta_{2k}x_k \tag{16}$$

El segundo componente principal, será la combinación lineal que tiene la máxima varianza de todas las combinaciones lineales ortogonales a f_1. En otras palabras, responde a la pregunda: ¿Cuál es la mejor combinación lineal de las variables $x_1, x_2, \ldots, x_k$ no correlacionada al primer componente principal? Intuitivamente, esta es la "segunda mejor" combinacion lineal de $x_1, x_2, \ldots, x_k$, que no esta contenida en el primer componente.

El cálculo del segundo componente entonces responde al siguiente problema:

$$\max_{\delta_2} \ \delta_2 S \delta_2' \tag{17}$$
$$\text{sujeto a} \tag{18}$$
$$\delta_2\delta_2' = 1y \tag{19}$$
$$\delta_2\delta_1' = 0 \tag{20}$$

Donde el objetivo a maximizar y la primera restricción son similares al del problema del primer componente. La diferencia surge en la segunda restricción donde imponemos que $\delta_1\delta_2' = 0$ que asegura que los componentes no esten correlacionados (sean ortogonales). Escribiendo el Lagrangiano,

$$\mathcal{L} = \delta_2 S \delta_2' + \lambda_2(1 - \delta_2\delta_2' + \lambda_{21}(0 - \delta_1\delta_2') \tag{21}$$

y maximizaremos esta expresión de la forma habitual derivando respecto a δ_2 e igualando a cero:

$$\frac{\partial \mathcal{L}}{\partial \delta_2} = -2S\delta_2' - \lambda_2 2\delta_2' - \lambda_{21}\delta_1' = 0 \tag{22}$$

Premultiplicando por δ_1

$$-2\delta_1 S\delta_2' - \lambda_2 2\delta_1\delta_2' - \lambda_{21}\delta_1\delta_1' = 0 \tag{23}$$

Notemos que $\delta_1\delta_1' = 1$ por lo tanto $\lambda_{21} = 0$, y que $\$\delta\_1\delta'\_2 = 0\$.Usandoestasecuacionestenemos$:

$$S\delta_2' = \lambda_2\delta_2' \tag{24}$$
$$\delta_2 S\delta_2' = \lambda_2 \tag{25}$$

De forma que δ_2 es un eigenvector de S, y puesto que queremos maximizar la varianza, deberiamos elegir el eigenvector asociado al eigenvalue mas grande, pero este ya lo utilizamos para el primer componente. Entonces δ_2 es el eigenvector asociado al segundo eigenvalue mas grande.

Siguiendo esta lógica, utilizando inducción es posible seguir calculando componentes cada uno ortogonal entre si y decrecientes en importancia. En general, para una matriz X con n observaciones y k variables tiene al menos el mínimo entre el número de observaciones menos 1 $(n-1)$ y el número de variables (k), i.e. $min(n-1,k)$, componentes principales distintos.

Ejemplo cálculo PCA (cont.) Continuando con el ejemplo anterior, el segundo componente principal es aquel que esta asociado al segundo eigenvalue mayor, que es 0.4617445.

[7]:
```
eig_val
```

[7]:
```
array([2.25116235, 0.28709315, 0.4617445 ])
```

El eigenvector asociado a este eigenvalue:

[8]:
```
eig_vec[[2]]
```

[8]:
```
array([[-0.10616542, -0.63455599,  0.76555052]])
```

El segundo componente principal es:

$$f_2 = -0.10 \times Salud - 0.63 \times Transporte + 0.76 \times Educacin \tag{26}$$

Calculando este para cada una de las ciudades tenemos entonces un índice que reduce las tres variables en una:

[9]:
```
f2=gasto.dot(eig_vec[[2]].T) #calculamos f2
f2.columns = ['CP2'] #nombramos la columna del primer componente como CP2
f2.sort_values(by='CP2', ascending=False).head()
```

[9]:

Ciudad	CP2
Inírida	2.122875
Bogotá	1.259882
Leticia	1.217096
San José del Guaviare	0.911059
Barrancabermeja	0.879264

1.4 Vizualiación de los componentes

Una vez que hemos calculado los componentes principales, podemos graficarlos entre sí para producir una representación de dimensión menor de los datos. Comenzamos uniendo las bases con los PCAs calculados:

[10]:
```
pcas = pd.concat([f1,f2], axis=1, join="inner")
pcas.head()
```

[10]:

Ciudad	CP1	CP2
Arauca	-1.417381	0.078945
Armenia	0.951708	-0.080257
Barrancabermeja	0.032537	0.879264
Barranquilla	-0.078678	0.302219
Bogotá	5.173817	1.259882

Por ejemplo, podemos graficar los primeros dos componentes principales. Geométricamente, esto equivale a proyectar los datos originales en el subespacio generado por δ_1 y δ_2, y graficar los puntos proyectados.

[11]:
```
import plotly.express as px # Cargamos el paquete plotly para hacer gráficas␣
 ↪interactivas.
fig = px.scatter(pcas, x = "CP1", y = "CP2",
    text = pcas.index.values)
fig.update_traces(textposition = 'top center')
```

1.5 Referencias

- Ahumada, H. A., Gabrielli, M. F., Herrera Gomez, M. H., & Sosa Escudero, W. (2018). Una nueva econometría: Automatización, big data, econometría espacial y estructural.
- DANE (29 de Septiembre de 2020). Encuesta nacional de presupuestos de los hogares (ENPH). Anexos: 32 ciudades y 6 ciudades intermedias. https://www.dane.gov.co/files/investigaciones/boletines/enph/ciudades-enph-2017.xls
- Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). Mathematics for machine learning. Cambridge University Press.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
- Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.
- Peña, D. (2002). Análisis de datos multivariantes (Vol. 24). Madrid: McGraw-hill.