Análisis de Componentes Principales

Con Python

Jose Luis Paniagua Jaramillo
jlpaniagua@uao.edu.co

Agenda

Agenda

Reducción de Dimensionalidad

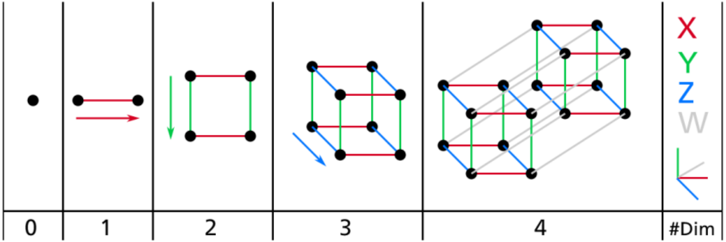


Figura: [1]

El problema de la dimensionalidad

- alta probabilidad de presentarse sobre entrenamiento (overfitting).
- dificultades para la visualización.
- afecta la velocidad en el proceso de entrenamiento.

Agenda

1. Reducción de Dimensionalidad
   - PCA
2. Implementacion en Python
   - Desde cero
   - Usando la librería scikit-learn
3. Referencias

Principal Component Analysis - PCA I

original 3D dataset

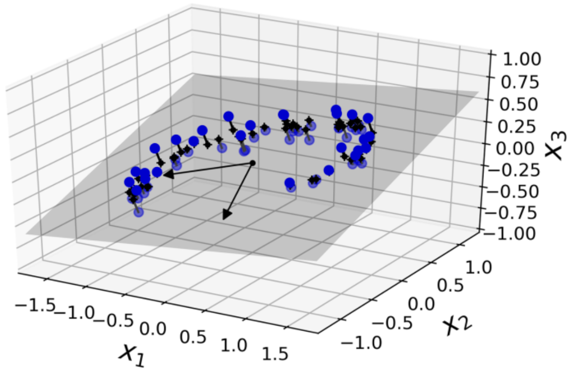


Figura: [1]

2D dataset after projection

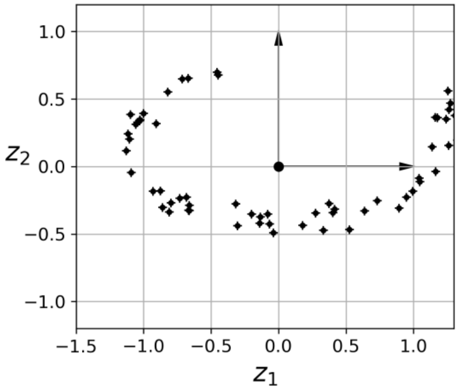
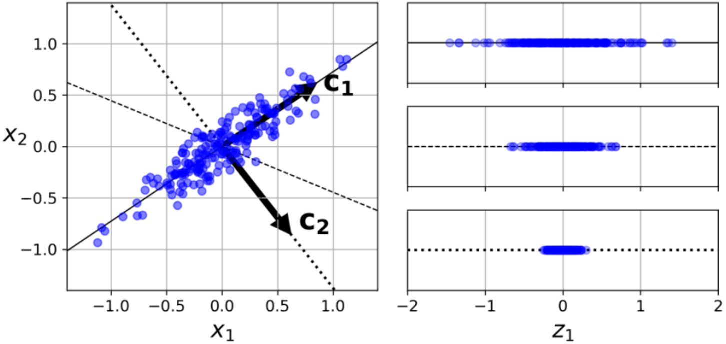


Figura: [1]

Principal Component Analysis - PCA II



- se debe seleccionar el eje (hiperplano) que preserve la mayor varianza.
- mayor varianza, implica menos perdida de información.
- el obejtivo es proyectar sobre los ejes que minimicen la distancia entre los datos originales y su proyección.

Agenda

1. Reducción de Dimensionalidad
   - PCA
2. Implementacion en Python
   - Desde cero
   - Usando la librería scikit-learn
3. Referencias

Agenda

Calculo de los Componentes Principales I

Singular Value Decomposition (SVD)

SVD es una técnica estándar de factorización de matrices.

$$X = U\Sigma V^T$$

- U y V son matrices unitarias.
- Σ es una matriz diagonal con $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$

Calculo de los Componentes Principales II

¿Qué imagen enviaría?

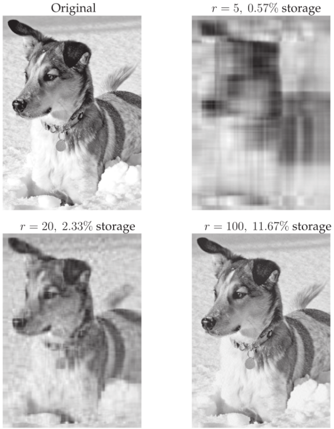


Figura: source[2]

Calculo de los Componentes Principales III

The Singular Value Decomposition (SVD) separa cualquier matriz en piezas sencillas (escalado y rotación)

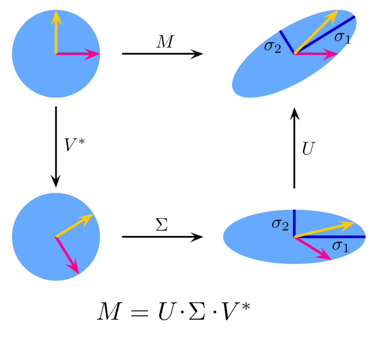
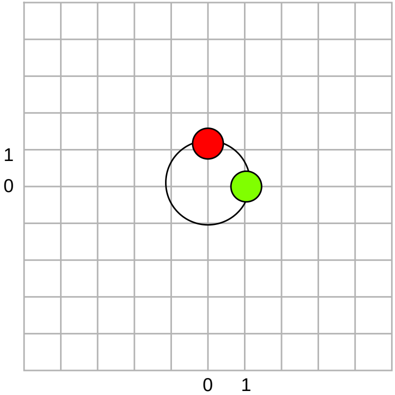


Figura: source: wikipedia

Calculo de los Componentes Principales IV

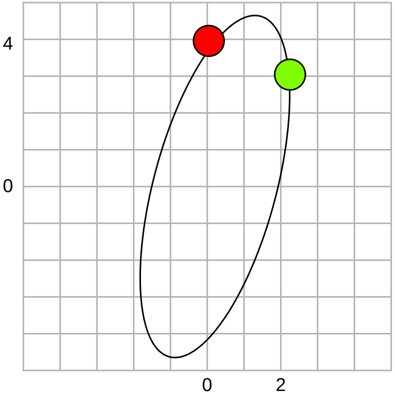
Transformación Lineal = Matriz



$$M = \begin{bmatrix} 2 & 0 \\ 3 & 4 \end{bmatrix}$$

$(1,0)$ $(2,3)$

$(1,0)$ $(2,3)$



Calculo de los Componentes Principales V

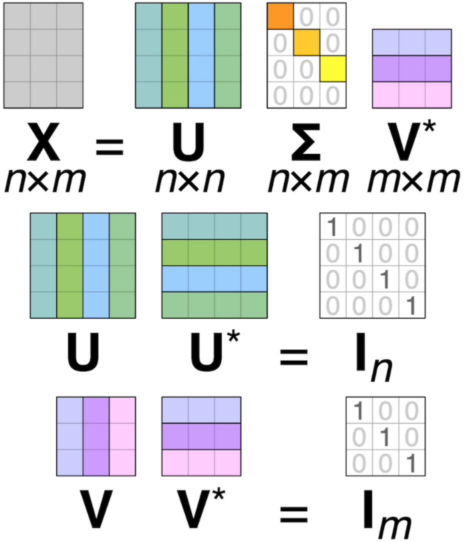
SVD

$$M = U\Sigma V'$$

$$\begin{bmatrix} 2 & 0 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$$

- U and V' are unitary.
- Σ is diagonal with $\sigma_1 \geq \sigma_2 \geq \sigma_3 \cdots \geq 0$

Calculo de los Componentes Principales VI



Calculo de los Componentes Principales VII

$$
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
=
\begin{bmatrix}
1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1
\end{bmatrix}
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
$$

Calculo de los Componentes Principales VIII

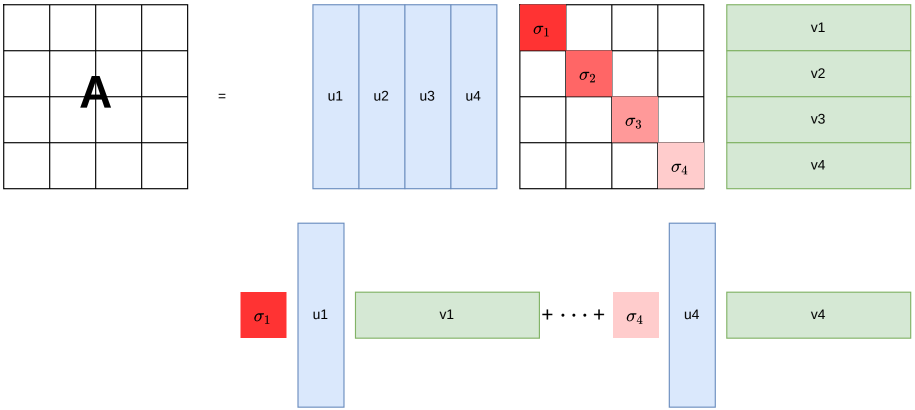
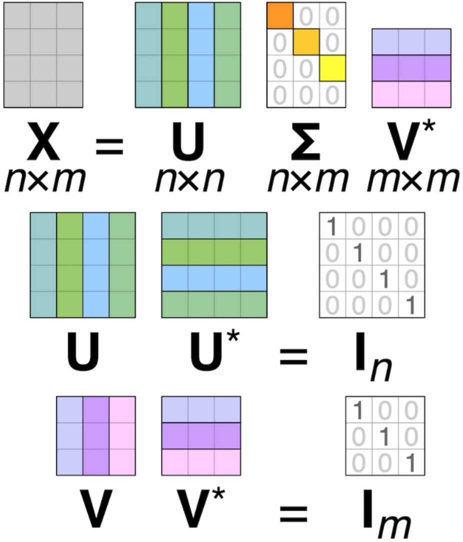


Figura: source:[3]

Calculo de los Componentes Principales IX



V contiene los vectores unitarios que definen los componentes principales.

$$V = \begin{bmatrix} | & | & & | \\ c_1 & c_2 & \dots & c_n \\ | & | & & | \end{bmatrix}$$

Proyección

Reduccion de Dimensionalidad

Después de identificar los componentes principales, es posible realizar la reducción de dimensional proyectando la matriz X en el hiperplano definido por las primeros d componentes principales.

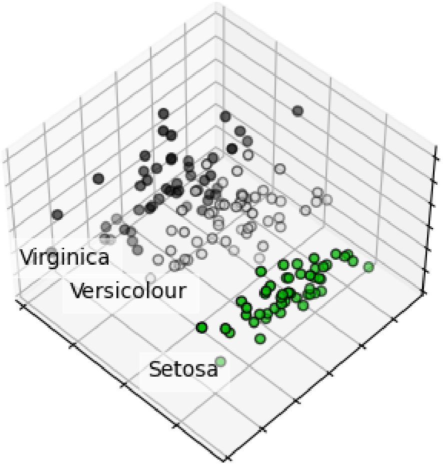
$$X_{d-proj} = XW_d$$

Ejemplo: iris dataset

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Agenda

Ejemplo: iris dataset



Agenda

1. Reducción de Dimensionalidad
   - PCA
2. Implementacion en Python
   - Desde cero
   - Usando la librería scikit-learn
3. Referencias

Referencias

Aurélien Géron.
Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.
O'Reilly Media, 2019.

Steven L Brunton and J Nathan Kutz.
Data-driven science and engineering: Machine learning, dynamical systems, and control.
Cambridge University Press, 2019.

Luis Serrano.
Grokking Machine Learning.
Simon and Schuster, 2021.

https://towardsdatascience.com/principal-component-analysis-pca-from-scratch-in-python-7f3e2a540c51
https://scikit-learn.org/stable/index.html