Análisis de datos

Clase 6 - Reducción de dimensiones

Métodos de proyección

Estos métodos de reducción de dimensiones se basan en aplicar transformaciones, en principio lineales, a los datos originales. De este modo esperamos capturar las direcciones de mayor importancia en los datos.

En esta materia vamos a estudiar tres métodos

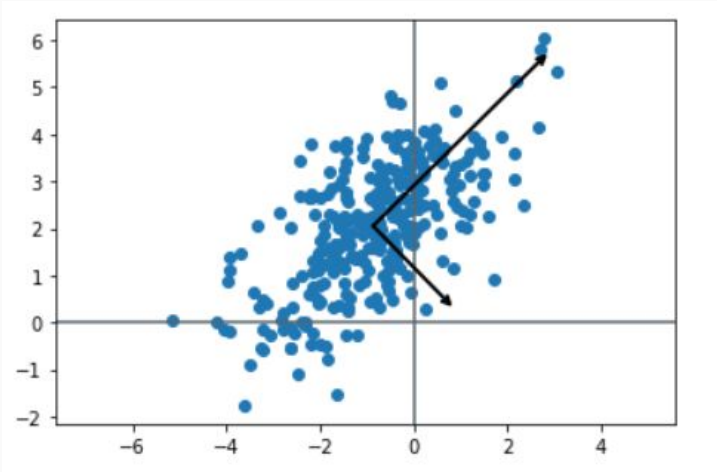
- PCA
- ICA
- SVD

pero existen muchos otros como KPCA, FA, etc.

Análisis de componentes principales (PCA)

Motivación

PCA busca proyectar los datos en un espacio lineal (de menor dimensión), llamado *subespacio principal*, tal que la varianza en los datos proyectados sea máxima.



Análisis de componentes principales (PCA)

Cómo hallar las direcciones de máxima varianza

Supongamos que tenemos una matriz $\mathbf{X} \in \mathbb{R}^{n \times p}$ una matriz de n observaciones de p variables . Buscamos hallar las $m \leq p$direcciones que maximicen la varianza de las muestras.

Primero hallamos $\boldsymbol{\alpha_1} \in \mathbb{R}^n$ tq $\boldsymbol{\alpha}_1^T \mathbf{X}$ sea máximo sujeto a $\boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_1 = 1$

Luego buscamos $\boldsymbol{\alpha_2} \in \mathbb{R}^n$ tq $\boldsymbol{\alpha}_2^T \mathbf{X}$ sea máximo sujeto a $\boldsymbol{\alpha}_2^T \boldsymbol{\alpha}_2 = 1$ y además $\boldsymbol{\alpha}_2^T \mathbf{X}$ esté descorrelacionado con $\boldsymbol{\alpha}_1^T \mathbf{X}$ i.e . $\boldsymbol{\alpha_2} \perp \boldsymbol{\alpha_1}$.

Se prosigue de la misma forma hasta hallar los m vectores $\boldsymbol{\alpha_1}, \ldots, \boldsymbol{\alpha_m}$

Notar que por definición la matriz $\boldsymbol{\alpha}_m = \left[\boldsymbol{\alpha_1} \ \ldots \ \boldsymbol{\alpha_m}\right]$ es ortonormal, y por lo tanto define una matriz de proyección.

Análisis de componentes principales (PCA)

Vinculación con los autovectores

Definiendo $\tilde{\Sigma} = (\mathbf{X} - \bar{\mathbf{X}})^T(\mathbf{X} - \bar{\mathbf{X}})$, la matriz de covarianza muestral, los $\boldsymbol{\alpha_i}$ están asociados a los m primeros autovectores de $\tilde{\Sigma}$.

Al ser $\tilde{\Sigma}$ simétrica, $\tilde{\Sigma} = \mathbf{V}\mathbf{S}\mathbf{V}^T$,

donde $\mathbf{S} = diag(\lambda_1, \ldots, \lambda_p)$, $\lambda_1 \geq \ldots \geq \lambda_p$ son los autovalores de $\tilde{\Sigma}$, y $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_p]$ con $\mathbf{v}_i$ el autovector asociado a λ_i.

Se concluye que $\boldsymbol{\alpha}_m = \mathbf{V}_m$ donde $\mathbf{V}_m = [\mathbf{v}_1, \ldots, \mathbf{v}_m]$

Los datos transformados resultan $\hat{\mathbf{X}} = \mathbf{X}\mathbf{V}_m$

Análisis de componentes principales (PCA)

Criterios de selección de orden

Una pregunta importante es cómo elijo la cantidad m de features a retener. Existen dos enfoques comúnmente usados:

- Busco explicar el k% de la varianza de los datos: m es tal que $\frac{\sum_{i=1}^{m} \sigma_i}{\sum_{i=1}^{p} \sigma_i} 100 > k$

- Método del codo ('elbow'). Grafico los λ_i y tomo m en el punto de inflexión de la curva.

Análisis de componentes principales (PCA)

Comentarios finales

- **Ventajas:**
  - Obtengo features descorrelacionados
  - No "tiro" información de ninguna variable
  - Explicable en términos de la matriz de correlación
  - No supervisado (sirve para mayor cantidad de problemas)
- **Desventajas:**
  - Si el dataset es muy grande puede ser muy costoso de computar
  - Pierdo explicabilidad de los features (ahora son una c.l. de las mediciones)
- **Observaciones:**
  - Las direcciones de los componentes principales se pueden ver afectadas por las unidades de medida, por ejemplo un feature es la altura en metros de una persona y otra el peso en gramos). Una práctica común estandarizar las variables para que tengan media 0 y varianza 1 antes de aplicar PCA.

Independent Component Analysis (ICA)

ICA, también conocido como *“blind source separation”* o *“Cocktail party problem”*, busca transformar el dataset en columnas independientes.

El modelo asume que cada señal se puede modelar como una combinación lineal de componentes independientes. Sean $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_k$ las k fuentes independientes, luego cada señal se modela como}

$$\boldsymbol{x} = \alpha_1 \boldsymbol{s}_1 + \ldots + \alpha_k \boldsymbol{s}_k$$

Si definimos $X = \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix}$, podemos escribirlo como $X = A \begin{bmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_k \end{bmatrix} = AS$

ICA

Suposiciones de ICA:

- El proceso de mezcla es lineal
- Todas las fuentes de señal son independientes
- Las señales **no** son gaussianas

Nota: a diferencia de lo que ocurre en PCA, el orden de las componentes no significa nada.

ICA

Observación:

ICA no es exactamente una técnica de reducción de dimensiones, sin embargo puede usarse para tal fin. Supongamos que se tiene un dataset de n observaciones, con q variables cada Luego, si modelamos las observaciones como provenientes de $m<q$ fuentes, obtenemos una reducción en las dimensiones del problema.

```
import numpy as np
from sklearn.decomposition import FastICA

# Generate random Data of size (n x 5).
X = np.random.uniform(low=0, high=100, size=(20, 5))

# Number of sources wanted. The resulted sources are (n x 3).
ica = FastICA(n_components=3)
sources = ica.fit_transform(X)
```

ICA - Preprocesamiento

- **Centrado:** restamos la media de todas las señales

$$D = X - \mu = \begin{bmatrix} \boldsymbol{d}_1 \\ \vdots \\ \boldsymbol{d}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1 - \mu \\ \vdots \\ \boldsymbol{x}_n - \mu \end{bmatrix}$$

- **Blanqueado:** Buscamos descorrelacionar las variables y estandarizar las a varianza unitaria. Esto puede hacerse con PCA.

  Observación: Si las señales fueran gaussianas, entonces con PCA alcanza para obtener señales independientes. Por este motivo se pide que las señales no tengan distribución gaussiana para aplicar ICA.

¿Cómo funciona ICA?

El objetivo de ICA es a partir de las señales X, encontrar las componentes S, es decir que buscamos la matriz W tal que $S = WX$. A esto se lo conoce como *unmixing problem*.

Una vez hallada W, se proyectan los datos blanqueados sobre esa matriz para hallar las componentes independientes.

ICA: criterios de cómputo

Existen tres principales criterios de independencia que llevar a distintas :

1. Basados en la no-gaussianidad. Esto puede medirse usando medidas como ***negentropy*** o kurtosis. El objetivo es hallar las componentes que maximicen la no-gaussianidad.
2. Minimizando la información mutua entre las componentes
3. Usando estimación de máxima verosimilitud.

El preprocesamiento se calcula directo de los datos, pero la matriz W se obtiene por aproximación numérica, mediante métodos de optimización. La solución óptima es difícil de hallar debido a la presencia de extremos locales en la función objetivo.

ICA: implementación de Scikit-Learn

La implementación de Scikit-Learn se basa en el algoritmo FastIca, basado en la negentropy.

Se define la negentropy como $J(y)=H(y_{gauss})-H(y)$. Esta definición se basa en el hecho que la para un nivel de varianza constante, las variables gaussianas con las que tienen máxima entropía.

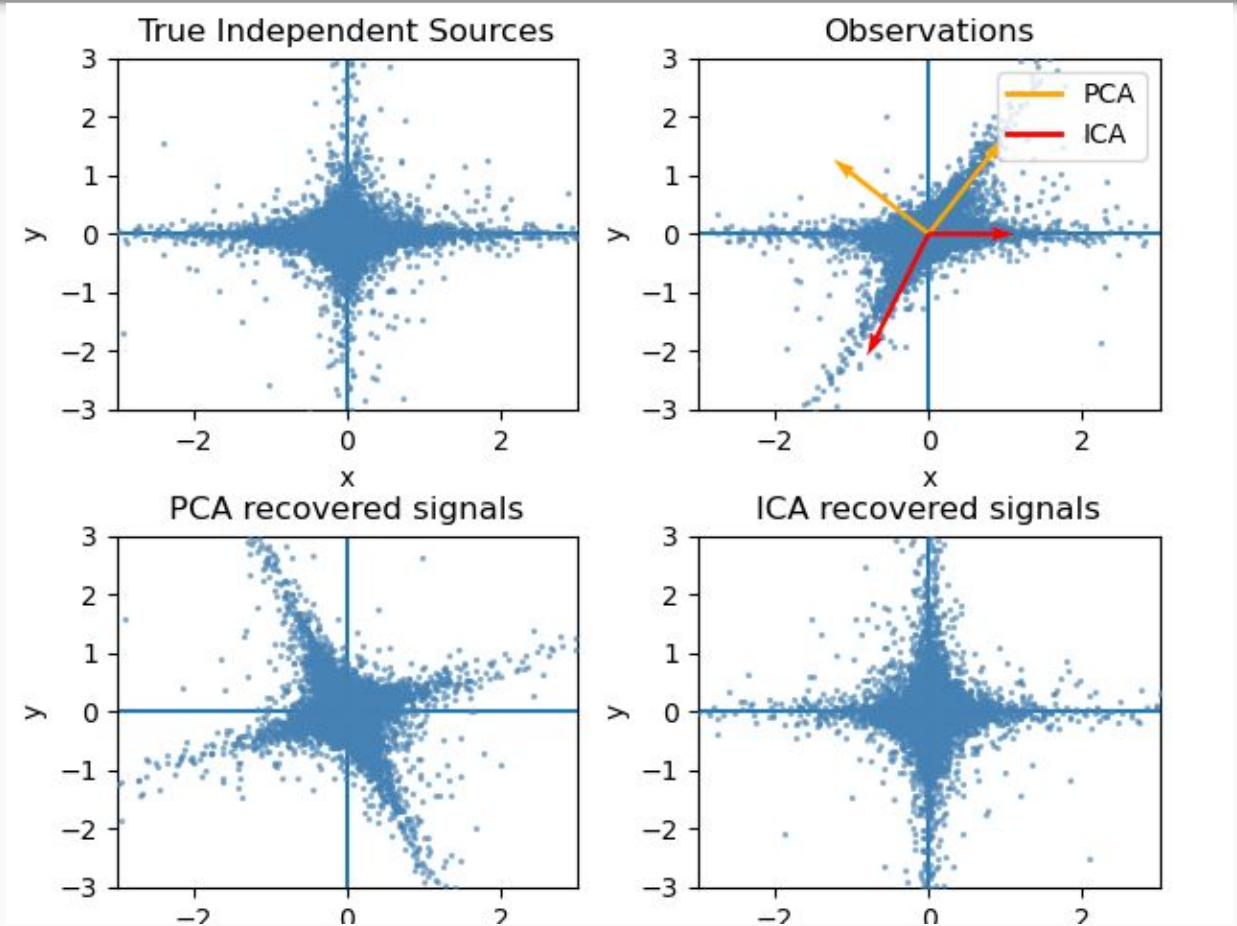
Se utiliza $J(y)\propto[E\{G(y)\}-E\{G(\nu)\}]^2$ para estimar J(y).

ICA: implementación de Scikit-Learn

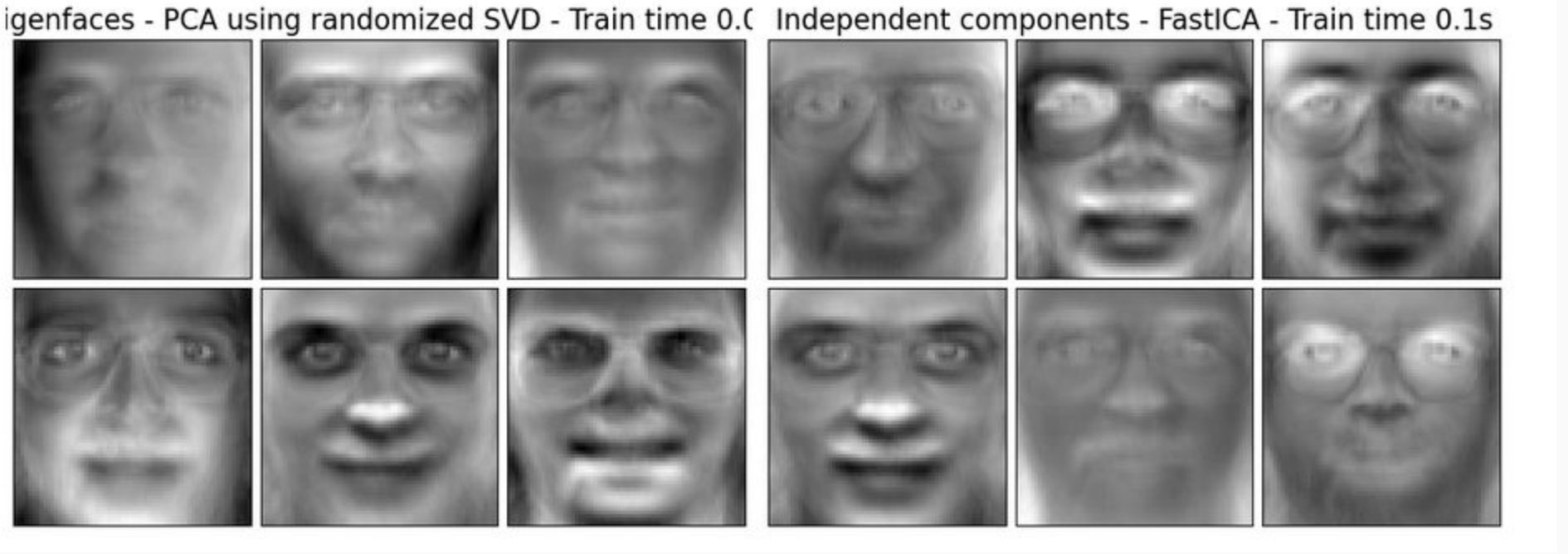
FastICA algorithm is as follows:

1. Choose an initial (e.g. random) weight vector $\mathbf{w}$.
2. Let $\mathbf{w}^{+}=E\{\mathbf{x}g(\mathbf{w}^{\mathrm{T}}\mathbf{x})\}-E\{g'(\mathbf{w}^{\mathrm{T}}\mathbf{x})\}\mathbf{w}$
3. Let $\mathbf{w}=\mathbf{w}^{+}/\|\mathbf{w}^{+}\|$
4. If not converged, go back to 2.

ICA vs PCA: ejemplo 1



ICA vs. PCA - ejemplo 2



Descomposición en valores singulares (SVD)

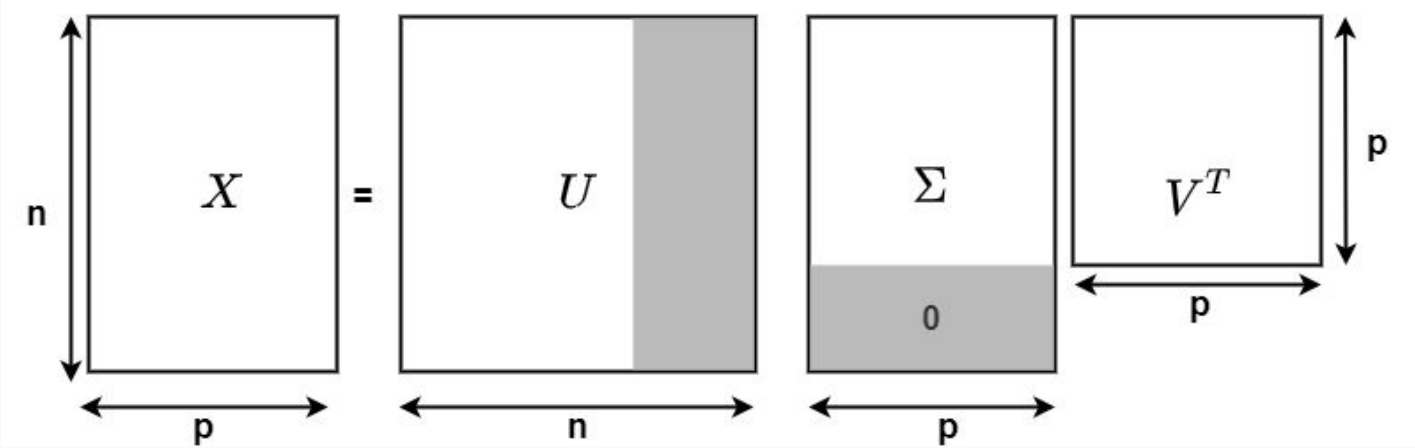
La SVD es muy popular para reducir dimensiones para datos *sparse*.

Sea $\mathbf{X} \in \mathbb{R}^{n \times p}$ una matriz de n observaciones de p variables. Luego, $\mathbf{X}$ se puede descomponer como

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

donde $\mathbf{U} \in \mathbb{R}^{n \times n}$, $\mathbf{V} \in \mathbb{R}^{p \times p}$ y $\mathbf{\Sigma} \in \mathbb{R}^{n \times p}$ es una matriz "diagonal"

Descomposición en valores singulares (SVD)



Finalmente, los datos transformados resultan $\hat{\mathbf{X}} = \mathbf{X}\mathbf{V}_\mathbf{k}$,

donde $\mathbf{V}_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$

Descomposición en valores singulares (SVD)

¿Qué representan las matrices **U** y **V**?

- $\mathbf{U}$ se corresponde con los autovectores de $\mathbf{X}\mathbf{X}^T$ (correlación empírica entre muestras)
- $\mathbf{V}$ está asociada a los autovectores de $\mathbf{X}^T\mathbf{X}$ (correlación empírica de los features)
- $\mathbf{\Sigma}$ es la raíz cuadrada de lo autovalores de ambas matrices.

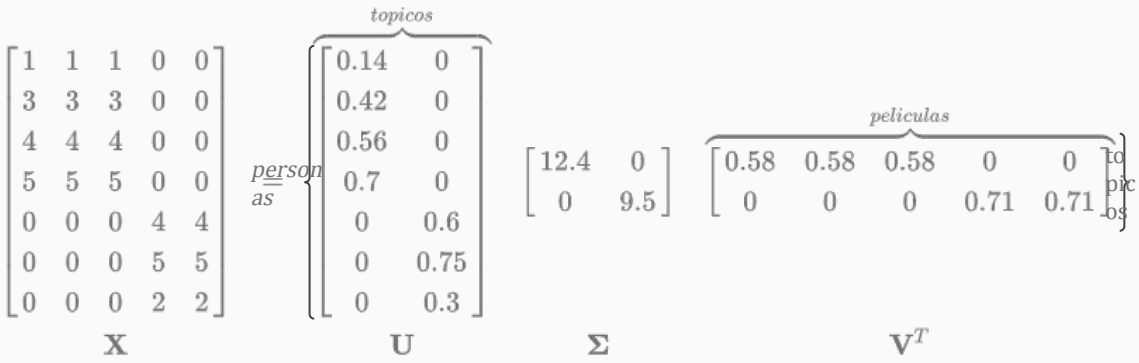
Descomposición en valores singulares (SVD)

¿Qué representan las matrices **U** y **V**? Ejemplo

Puntuación de distintos usuarios a 5 películas[1]

	Matrix	Alien	Star Wars	Casablanca	Titanic
Joe	1	1	1	0	0
Jim	3	3	3	0	0
John	4	4	4	0	0
Jack	5	5	5	0	0
Jill	0	0	0	4	4
Jenny	0	0	0	5	5
Jane	0	0	0	2	2

SVD



[1] http://infolab.stanford.edu/~ullman/mmds/ch11.pdf

Descomposición en valores singulares (SVD)

Comentarios finales

- Realizar la descomposición SVD sobre los datos centrados es equivalente a hacer PCA
- SVD funciona sobre datos sparse, sin necesidad de "redensificarlos" que puede ocupar mucha memoria
- Según el problema, las matrices $\mathbf{U}$ y $\mathbf{V}$ de la SVD pueden dar información útil acerca de las correlaciones entre las muestras y variables.

Bibliografía

- "Mining of Massive Datasets", Leskovec J, Rajaraman A., Ullman J.D., Stanford University. Capítulo 11. http://infolab.stanford.edu/~ullman/mmds/ch11.pdf
- https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c
- "Pattern Recognition and Machine Learning", Bishop, Christopher M. New York, Springer, 2006
- A. Hyvarinen and E. Oja, Independent Component Analysis: Algorithms and Applications, Neural Networks, 13(4-5), 2000, pp. 411-430
- Independent component analysis: An introduction