

Tarea 2

Aprendizaje de Máquina I, 2020

Eigenfaces

En esta tarea, vamos a implementar el tradicional método de *eigenfaces* con dos conjuntos de datos: AT&T Faces y MNIST. El conjunto AT&T Faces se descarga en Python de la siguiente forma:

```
from sklearn.datasets import fetch_olivetti_faces
data, targets = fetch_olivetti_faces(return_X_y=True)
```

El conjunto MNIST puede descargarse en Python de la siguiente forma:

```
from keras.datasets import mnist
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

Ejercicio 1. Implemente en Python un algoritmo que haga PCA, o equivalentemente, la descomposición SVD de una matriz de datos $\mathbf{X} \in \mathbb{R}^{n \times d}$, donde $n \geq 1$ es el número de datos, y $d \geq 1$ la dimensión.

Su algoritmo debe estar en una función que reciba como parámetros, a lo mínimo, la matriz de datos \mathbf{X} , y el número k de componentes principales a extraer. Como resultado, su función debe devolver, al menos los siguientes:

- la matriz de datos reducidos (proyectados) a dimensión k ,
- el conjunto de direcciones principales,
- el conjunto de valores singulares (o los autovalores).

Ejercicio 2 (MNIST). Considere la base de datos MNIST. Utilice únicamente el conjunto de 10,000 imágenes de prueba \mathbf{X}_{test} .

- Haga una descomposición SVD o PCA de los datos. Elabore una gráfica que explique el porcentaje de variabilidad explicada en función del número k de componentes incluidas en un análisis.
- Haga una descomposición PCA de los datos, para diferentes valores de k . Por ejemplo, use $k = 10, 50, 100, 200$ y 300 . Haga una gráfica de los *eigenfaces*. Para cada uno de los valores de k :
 - Reconstruya las imágenes y calcule el error de la reconstrucción.
 - Muestre un gráfico comparativo de algunas (al menos 10) imágenes originales contra sus reconstrucciones para los diferentes valores de k .
- Compare en una gráfica o tabla los valores del error obtenido contra k . Discuta sus resultados.

Ejercicio 3 (AT&T Faces). Repita el ejercicio anterior para el conjunto de datos *Olivetti Faces*.

Observación. Puede utilizar herramientas que calculen PCA ya contenidas en librerías de Python únicamente para fines comparativos, no como su tarea. Sí puede usar `np.linalg.svd`.