# PRÁCTICA 3 - BIOMETRÍA

LDA - FisherFaces

Cristian Villarroya Sánchez

01-05-2021

# Índice

# 1. Descripción de la tarea

Se disponen de 10 imágenes de 40 individuos diferentes, siendo las 5 primeras las que se usarán en entrenamiento y las 5 últimas en test.

El objetivo es implementar fisherfaces utilizando el vecino más cercano y obtener las curvas de error variando el valor de la dimensionalidad reducida d'.

# 2. Carga de las imágenes

Las imágenes vienen dadas en formato PGM. Este formato es un formato de archivo en escala de grises de mínimo denominador común. El dataset está organizado de la siguiente manera:

- \data
  - \Test
    - \s1
      - 6.pgm
      - …
    - \s2
    - …
    -
  - \Train
    - \s1
      - 1.pgm
      - …
    - \s2
    - …

Lo primero que se ha hecho es un método (read_images) mediante el cual, se accede a la carpeta raíz y se buscan todas las carpetas dentro de dicha carpeta raíz. Esto se hace para hacer al algoritmo invariante de tener 40 o 50 carpetas con imágenes de usuarios, facilitando de esta manera la adicción de datos en un futuro si fuese necesario.

Para cada una de las subcarpetas de la carpeta raíz, busca todos los ficheros, que serán los ficheros PGM de las imágenes de caras.

Para leer el fichero PGM, se ha hecho uso del método descrito en la página Intellipaat [1] (read_pgm). Este método devuelve un conjunto de 3 valores, el primero es un numpy array de 1xn con los datos, el segundo una tupla con la longitud y profundidad y el tercero el número de niveles de gris.

Cada una de las imágenes obtenidas las guarda en un array. Finalmente se tienen dos arrays, uno con las imágenes de entrenamiento y otro con las de test.

## 3. Cálculo de LDA

LDA también es un método de proyección lineal, pero a diferencia de PCA, si es discriminante ya que sí tiene en cuenta las etiquetas de las clases. Tiene una limitación y es que solamente puede proyectar desde 1 a c-1 dimensiones, siendo c el número de clases.

Para calcular la matriz de proyección se define por un lado la "Between-class scatter matrix", que mide, con respecto a la media, cómo de divergentes son las medias de las clases con respecto a la media global, es decir, la diferencia entre la media de cada una de las clases y la media global. También se tiene la "Within-class scatter mattrix", que calcula la varianza de los puntos clase por clase, es decir, dentro de cada clase, cuanto de "separadas" están las muestras de dicha clase. LDA funcionará solamente si las clases tienen una representación Gaussiana unimodal.

Si la dimensionalidad es muy alta puede que la matriz within-class no tenga inversa y no se pueda resolver LDA de esta manera, por lo que, antes de aplicar LDA se ha aplicado PCA, evitando así singularidades. Se ha proyectado mediante PCA a un nuevo espacio de 40 dimensiones. Se ha elegido 40 dimensiones por que se tienen imágenes de 40 individuos, además, valores mayores de 41, hacen que el script no funcione y valores más pequeños de 40, empeoran la precisión final del clasificador. Para calcular PCA se ha usado el algoritmo descrito en la práctica anterior.

Para el calculo de LDA, se ha definido un método (calculateLDA) en el que, en primer lugar, se calcula la media de la matriz devuelta por PCA. Después de calculan las matrices $S_b$ y $S_w$ de la siguiente manera:

$$S_b = \sum_{c=1}^{C} n_c(\mu_c - \mu)(\mu_c - \mu)^t, \quad S_w = \sum_{c=1}^{C} \sum_{\mathbf{x} \in c} (\mathbf{x} - \mu_c)(\mathbf{x} - \mu_c)^t,$$

Para $S_b$, para cada clase, se multiplica el numero de muestras de dicha clase por la resta de la media de dicha clase con la media global multiplicada por la resta de la media de dicha clase con la media global traspuesta.

Por otra parte, $S_w$ se calcula multiplicando la resta de la matriz de clase menos la media de dicha clase multiplicada por la resta de la matriz de clase menos la media de clase traspuesta.

Una vez se tienen las matrices $S_b$ y $S_w$ se calcula la matriz A como la inversa de $S_w$ por $S_b$.

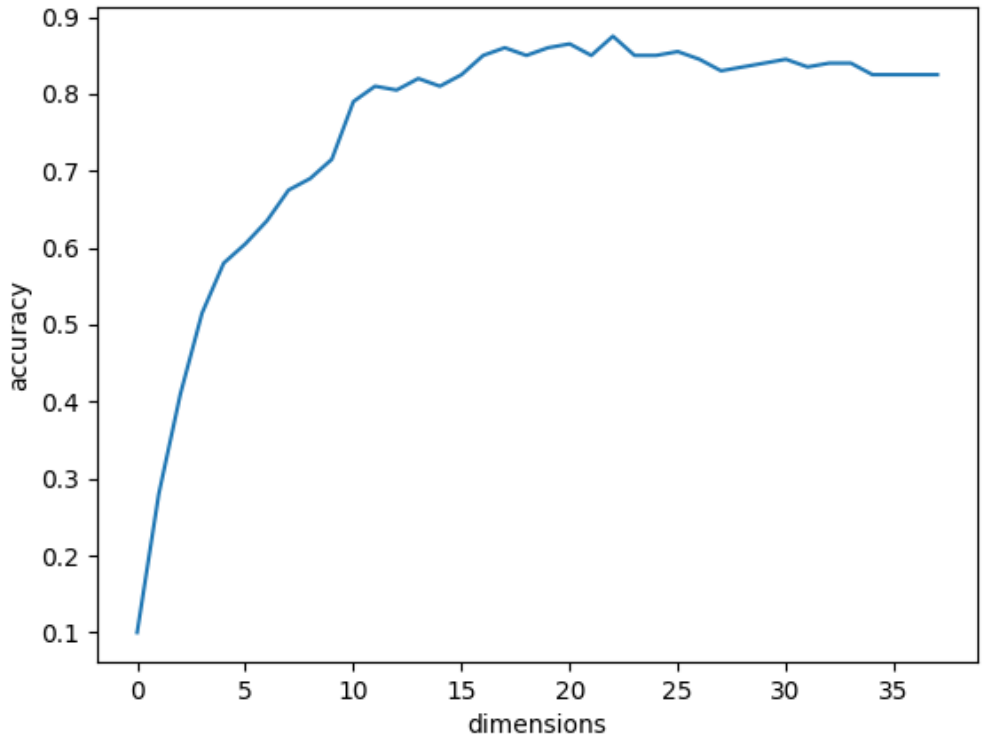
Dicha matriz A se diagonaliza para sacar los eigenvectors y eigenvalues, después se ordenan para guardar aquellos con máxima varianza y se devuelven la matriz de proyección B, conteniendo los eigenvectors

Con esa matriz de proyección se proyectan las imágenes de train y test (previamente reducidas con PCA) y se utiliza un clasificador por el vecino más cercano para la clasificación.

Ejecutando este script, reduciendo a 40 dimensiones con PCA y a 20 con LDA, se ha obtenido una precisión del 86%.

## 4. Variación de la dimensionalidad

Se ha realizado un estudio de la influencia de la dimensión en la precisión, para ello, se ha variado el rango desde 1 a 40, obteniendo los siguientes resultados:




Como puede verse, hay dimensiones que no aportan nada y podrían ser eliminadas. A partir de 25 dimensiones el resultado no solo no mejora, si n que parece empeorar. El mejor resultado ha sido obtenido con 22 dimensiones y una precisión del 87.5%.

## 5. Conclusiones

Se ha implementado un script que calcula las fisherfaces. Se ha visto que las imágenes pueden contener dimensiones que no aportan varianza y, por tanto, no tienen influencia en la precisión del clasificador, por eso, se pueden eliminar haciendo LDA.

Además, se ha visto la problemática de que es posible que una la matriz $S_w$ no tenga inversa y complique el calculo de LDA. Por tanto, es conveniente aplicar PCA antes de aplicar LDA evitando así singularidades que den lugar a este problema.

## 6. Bibliografía

[1] Intellipaat. How to read pgm p2 image in python, 2019