Práctica 4: Eigenfaces

Reconocimiento de Patrones - 0757

Facultad de Ingeniería
Universidad Nacional Autónoma de México

Ceballos Equihua C. N.
Ingeniería en Computación
Facultad de Ingeniería, UNAM
Ciudad de México, México
ceballos.equihua@gmail.com

Muñoz Marbán J.
Ingeniería en Mecatrónica
Facultad de Ingeniería, UNAM
Ciudad de México, México
juanmzmb@gmail.com

Murrieta Villegas A.
Ingeniería en Computación
Facultad de Ingeniería, UNAM
Ciudad de México, México
alfonsomvmx@gmail.com

Solano González F. J.
Ingeniería en Mecatrónica
Facultad de Ingeniería, UNAM
Ciudad de México, México
felipe.solano.gos@gmail.com

Profesores:
Dr. Boris Escalante Ramírez
Dra. Olveres Montiel Jimena
I.I.M.A.S. - UNAM

Resumen— **Esta práctica consiste en la correcta implementación de la técnica llamada Eigenfaces para realizar reconocimiento facial de un conjunto de imágenes de diversos rostros.**

I. Objetivo

Implementar el análisis por componentes principales para realizar reconocimiento facial de un conjunto de imágenes utilizando la técnica de Eigenfaces.

II. Introducción

Para comenzar se debe mencionar una de las principales herramientas utilizadas para el desarrollo de la práctica:

Análisis de componentes principales

El análisis de componentes principales (ACP) consiste en expresar un conjunto de variables en un conjunto de combinaciones lineales de factores no correlacionados entre sí, estos factores dando cuenta una fracción cada vez más débil de la variabilidad de los datos. Este método permite representar los datos originales (individuos y variables) en un espacio de dimensión inferior del espacio original, mientras limite al máximo la pérdida de información.

Una forma intuitiva de entender el proceso de PCA consiste en interpretar las componentes principales desde un punto de vista geométrico. Supóngase un conjunto de observaciones para las que se dispone de dos variables (X1, X2). El vector que define la primera componente principal (Z1) sigue la dirección en la que las observaciones varían más (linea roja). La proyección de cada observación sobre esa dirección equivale al valor de la primera componente para dicha observación (principal component scores, zi1).

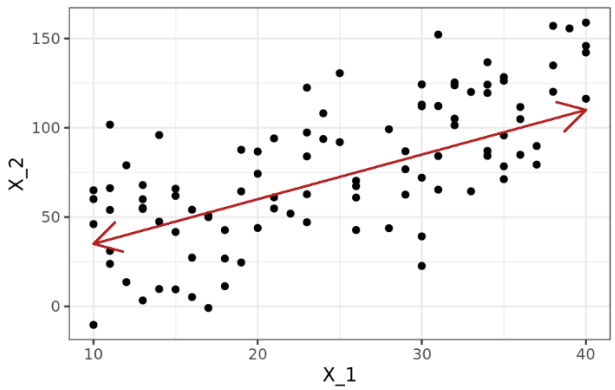

Figura 1. PCA – Primera proyección

La segunda componente (Z2) sigue la segunda dirección en la que los datos muestran mayor varianza y que no está correlacionada con la primera componente. La condición de no correlación entre componentes principales equivale a decir que sus direcciones son perpendiculares/ortogonales.

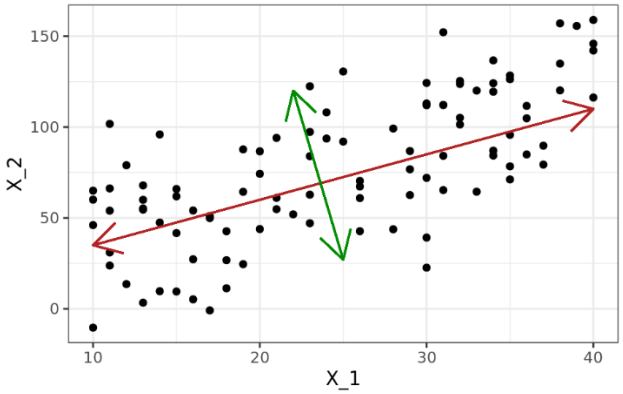

Figura 2. PCA – Segunda proyección

El análisis de componentes principales no se limita a un numero determinado de componentes.

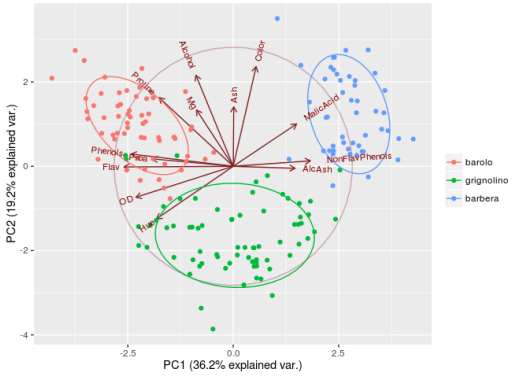

Figura 3. PCA

Eigenfaces

Los eigenfaces se refieren a un enfoque sobre el reconocimiento de rostros que busca capturar la variación en una colección de imágenes de rostros y utilizar esta información para codificar y comparar imágenes de rostros individuales de una manera holística (en contraposición a una basada en partes).

Figura 4. Imágenes de rostros

Las eigenfaces son los componentes invariables de una serie de caras, o lo que es lo mismo, los vectores propios de la matriz de covarianza del conjunto de imágenes de caras que forman una base de datos.

Esto quiere decir, que extraen la información facial relevante, que puede o no estar directamente relacionada con la intuición humana de rasgos faciales como los ojos, la nariz y los labios. Una forma de hacerlo es capturar la variación estadística entre las imágenes faciales. Los eigenfaces pueden considerarse como un conjunto de rasgos que capturan la variación global entre las imágenes faciales.

La idea de utilizar componentes invariables para representar rostros humanos fue desarrollada por Sirovich y Kirby (Sirovich y Kirby 1987) y utilizada por Turk y Pentland (Turk y Pentland 1991) para la detección y el reconocimiento de rostros. El enfoque de Eigenface es considerado por muchos como la primera tecnología de reconocimiento facial que funciona, y sirvió de base para uno de los principales productos comerciales de tecnología de reconocimiento facial.

Desde su desarrollo y publicación iniciales, ha habido muchas extensiones del método original y muchos nuevos desarrollos en los sistemas de reconocimiento facial automático. Eigenfaces se sigue considerando a menudo como un método de comparación de referencia para demostrar el rendimiento mínimo esperado de un sistema de este tipo.

Figura 5. Eigenfaces

III. Desarrollo

El desarrollo de la práctica consistió en una base de datos del Proyecto de Rostros del MIT que consiste en un conjunto de imágenes de rostros. De ese conjunto se obtuvieron las imágenes que tuvieran un cierto porcentaje de información necesaria, es decir, que tuvieran un rostro que analizar.

El preprocesamiento de la práctica fue obtener el tamaño mínimo de las imágenes ya que varían de tamaño, por lo tanto, las imágenes de tamaño mayor se redimensionan al tamaño mínimo para tener homogeneidad.

```
min_rows, min_cols = sys.maxsize, sys.maxsize
max_rows, max_cols = 0, 0
for (i, image) in enumerate(all_images):
    r, c = image.shape[0], image.shape[1]
    min_rows = min(min_rows, r)
    max_rows = max(max_rows, r)
    min_cols = min(min_cols, c)
    max_cols = max(max_cols, c)

print("\n==> Tamaño de la imagen más pequeña:", min_rows, "x", min_cols, "píxeles")
print("==> Tamaño de la imagen más grande:", max_rows, "x", max_cols, "píxeles")
```

Figura 6. Obtención del tamaño mínimo y máximo de las imágenes.

```
# Se redimensionan las imágenes mayores al tamaño mínimo

resized_images = []
i = 0
for image in all_images:
  img = image
  if image.shape != (min_rows, min_cols):
    print(i)
    print("Imagen original:")
    cv2_imshow(image)
    img = cv2.resize(img, (min_rows, min_cols), interpolation = cv2.INTER_AREA
    print("Imagen redimensionada:")
    cv2_imshow(img)
  i += 1
  resized_images.append(img)
```

Figura 7. Redimensión de las imágenes.

Además, para poder hacer el análisis PCA se creó una matriz de entrenamiento con la lista que contiene las imágenes, de tal forma que se obtuvo una matriz de M x D, cuyo valor M es el número de imágenes de entrenamiento y D es el número de píxeles.

La biblioteca de NumPy cuenta con una función para descomponer una matriz de componentes singulares, cuya salida será una matriz U correspondiente a las filas, un vector cuyos valores corresponden a los mismos de la matriz diagonal S, y una matriz V correspondiente a las columnas.

El PCA se requiere antes obtener el valor promedio de los vectores, posterior a cada vector de entrenamiento se le resta dicho valor promedio. El resultado es una matriz que llamaremos M.

Para obtener la matriz de covarianza, la cual se obtiene con:

$$C = M\,M^T$$

El resultado será una matriz de dimensión TxT donde T es uno de los dalos de la imagen cuadrada, elevado al cuadrado. Por lo tanto, es una matriz enorme como resultado.

Para esto calculamos los vectores característicos que puedan representar la matriz de covarianza, pero utilizando la cantidad más representativa. Obteniendo como resultado 3956 características que nos representan el 99% de la varianza.

```
min_components = np.where(pca.explained_variance_ratio_.cumsum() > 0.99)[0][0]
print("Mínimo número de componentes para obtener una suma acumulada del 99% en la varianza: ", str(min_components))
```

```
Mínimo número de componentes para obtener una suma acumulada del 99% en la varianza:  695
```

```
print("Componentes totales: ", str(pca.n_components_))
```

```
Componentes totales:  3956
```

Figura 8. Obtención del número de componentes principales.

Para reconstruir imágenes utilizamos una regresión lineal que nos devolviera un vector de coeficientes para poder representar dicha imagen como una combinación lineal de los vectores catacterísticos. Después sumamos el vector promedio o cara promedio.

Figura 9. Imágenes de prueba.

Se fue reconstruyendo las imágenes de prueba con diferente número de eigenvalores o eigenfaces. Donde podemos ver en las siguientes imágenes que con 695 eigenfaces es suficiente para ver el rostro de la imagen real.

Figura 10. Reconstruyendo imágenes.

Finalmente, se hizo una sección de Face Morphing que es transformar un rostro a otro.

Figura 11. Face Morphing.

IV. Conclusión

En la presente práctica aprendimos acerca de análisis de componentes principales, así como su debida implementación. En términos generales, nos permite reducir el número de características minimizando la perdida de información. Los resultados obtenidos a pesar de tener menos de la mitad de las componentes principales son buenos, demostrando la efectividad de este método. Respecto a la técnica de Eigenfaces, se comprendió tanto de manera conceptual como práctica la implementación de esta técnica, como se puede observar en los resultados obtenidos se pudo realizar la reconstrucción y la deformación de diferentes rostros. En conclusión, se puede mencionar que los objetivos de la práctica se cumplen debido a que se realizó un análisis por componentes principales y se realizo reconocimiento facial por medio de la técnica de Eigenfaces.

Referencias

[1] S. Dey, «EigenFaces and A Simple Face Detector with PCA/SVD in Python,» 2018. [En línea]. Available: https://sandipanweb.wordpress.com/2018/01/06/eigenfaces-and-a-simple-face-detector-with-pca-svd-in-python/. [Último acceso: 27 de noviembre del 2021].

[2] J. P. Labonia, «IDIS,» 14 Julio 1991. [En línea]. Available: https://proyectoidis.org/eigenface/. [Último acceso: 25 11 2021].

[3] J. A. Rodrigo, «Análisis de Componentes Principales (Principal Component Analysis, PCA) y t-SNE,» Junio 2017. [En línea]. Available: https://www.cienciadedatos.net/documentos/35_principal_component_analysis. [Último acceso: 25 11 2021].

[4] «OpenCV Eigenfaces for Face Recognition,» [En línea].

[5] A. Rosebrock, «OpenCV Eigenfaces for Face Recognition,» 10 Mayo 2021. [En línea]. Available: https://www.pyimagesearch.com/2021/05/10/opencv-eigenfaces-for-face-recognition/. [Último acceso: 25 11 2021].