



ÁREA ACADÉMICA DE INGENIERÍA EN COMPUTADORES

CE5201 - PROCESAMIENTO Y ANÁLISIS DE IMÁGENES DIGITALES

Tarea 1: SVD y transformaciones espaciales lineales - Parte
3
Detección de caras

Estudiante:

Gabriel Brenes Vega

Kenneth Hernández Salazar

Isaac Porras Pérez

Carné:

2015127420

2017102682

2017107550

1. Método

La descomposición en valores singulares de una matriz (SVD, por sus siglas en inglés) es, según [1, 2], un tipo de factorización que generaliza para cualquier matriz rectangular el concepto de valores propios, mediante una extensión de la descomposición polar. Respecto a lo expuesto en [3], esta estrategia es similar a la diagonalización de matrices, la descomposición QR o la descomposición LU, pero con la ventaja que existe para cualquier matriz, sin importar sus dimensiones. Parte de su fortaleza recae en que los valores singulares de mayor magnitud se asocian a subespacios propios que condensan la información más significativa de los datos y de esta forma se puede reducir la dimensionalidad de problemas concretos como la compresión de imágenes, en donde es posible controlar la calidad de la compresión, mediante la magnitud de dichos valores.

2. Formulación matemática

Primero se procede a construir un conjunto S con 9 imágenes de 92x112 en escala de grises de la cara para cada una de las 40 personas presentes en la carpeta *Database*:

$$S = [f_1, f_2, \dots, f_{360}]$$

Donde cada f_i representa las imágenes de rostros de cada persona. Es decir, S es una matriz de tamaño 10304×360 , dado que $92 \times 112 = 10304$ y cada columna representa un rostro vectorizado de la base de datos. Después se calcula el rostro promedio \bar{f} del conjunto S :

$$\bar{f} = \frac{1}{N} \sum_{i=1}^N f_i$$

En donde $N=360$ pues representa la cantidad de imágenes totales en la base de datos que deben ser analizadas (40 personas y 9 imágenes de cada persona $9 \times 40 = 360$).

El siguiente paso es tomar nuevamente cada una de las imágenes de la base de datos (las que se encuentran en la carpeta *Database*) y restarles la imagen promedio \bar{f} y cada resultado se almacena en una nueva matriz A :

$$A = [f_1 - \bar{f}, f_2 - \bar{f}, \dots, f_{360} - \bar{f}]$$

Una vez que se tiene la matriz A se obtiene la descomposición en valores singulares de la misma:

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

Donde V es una matriz ortogonal de 360×360 y U es una matriz ortogonal de 10304×10304 que se expresan como:

$$V = [v_1, v_2, \dots, v_{360}]$$
$$U = [u_1, u_2, \dots, u_{10304}]$$

Seguidamente, para cada una de los rostros de entrenamiento, f_i , se tiene que sus coordenadas en el espacio de rostros están dadas por los vectores:

$$x_i = [u_1, u_2, \dots, u_r]^T \cdot (f_i - \bar{f}), \text{ con } r \text{ igual al rango de } A$$

Se elige un $\varepsilon_1=10$ el cual definirá la distancia máxima permitida entre una imagen y el espacio de imágenes para el proceso de reconocimiento.

Para dicho proceso, se tomará una imagen de prueba y se calculará su vector de coordenadas (1), su vector de proyección (2) y finalmente su distancia al espacio de caras (3):

$$x = [u_1, u_2, \dots, u_r]^T \cdot (f_i - \bar{f}) \quad (1)$$

$$f_p = [u_1, u_2, \dots, u_r] \cdot x \quad (2)$$

$$\varepsilon_f = \|(f - \bar{f}) - f_p\|_2 \quad (3)$$

Si $\varepsilon_f > \varepsilon_1$ la imagen no es una cara, en caso contrario, si $\varepsilon_f < \varepsilon_1$ se procede a calcular la distancia ε_i para cada una de las caras en la base de datos junto con la nueva imagen ingresada de la siguiente manera:

$$\varepsilon_i = \|x - x_i\|_2$$

Finalmente, cada uno de esos ε_i se almacenan en un vector para después calcular el valor mínimo del mismo y de acuerdo a la posición en que se encuentre dicho valor en el vector, esa será la posición de la imagen que coincide de mejor manera con la imagen ingresada y de esa manera se logra el reconocimiento de la persona.

3. Pseudocódigo

A continuación se muestra el pseudocódigo para este algoritmo:

Entrada: $\{f_1, f_2, \dots, f_N\}, f, \varepsilon_0$.

Salida: El rostro identificado o un mensaje indicando que el rostro es desconocido.

```

1: Construya  $S = (f_1 \ f_2 \ \dots \ f_N)$ .
2: Calcule  $\bar{f} = \frac{1}{N} \sum_{i=1}^N f_i$ .
3: Construya  $A = (a_1 \ a_2 \ \dots \ a_N)$ , donde  $a_i = f_i - \bar{f}$ .
4: Calcule  $A = U\Sigma V^T$ .
5: Defina  $U_r = (u_1 \ u_2 \ \dots \ u_r)$ , donde  $r = r(A)$ .
6: Para  $i = 1, \dots, N$  calcule  $x_i = (u_1 \ u_2 \ \dots \ u_r)^T \cdot (f_i - \bar{f})$ 
7: Calcule  $w = (u_1 \ u_2 \ \dots \ u_r)^T \cdot (f - \bar{f})$ 
8: Calcule  $\varepsilon = \min_{1 \leq i \leq r} \|w - x_i\|_2 = \|w - x_j\|_2$ , para algún  $j \in \{1, 2, \dots, N\}$ 
9: si  $\varepsilon < \varepsilon_0$  entonces
10:   retornar El rostro corresponde a la imagen  $f_j$ .
11: si no
12:   retornar El rostro es desconocido.
13: fin si

```

Figura 1: Pseudocódigo del algoritmo para reconocimiento de rostros con SVD

4. Aplicaciones del método en el procesamiento de imágenes

De acuerdo a [3], este algoritmo SVD permite representar imágenes mediante la extracción de aquellos valores que tienen una mayor concentración de la imagen y por lo tanto esta podría comprimirse para agilizar los procesos sobre ella. Este algoritmo se utiliza en labores como: compresión de imágenes, modelado de fondo de imágenes para la detección de movimiento en videos, eliminación de ruido de una imagen y reconocimiento facial.

5. Mejora implementada para el algoritmo SVD

La mejora que se implementó para el algoritmo SVD fue el uso del algoritmo de Proyección Bilateral Aleatoria en lugar del uso del rango reducido para SVD.

6. Resultados obtenidos

A continuación se muestran algunos de los resultados que se obtuvieron con el algoritmo de reconocimiento de rostros mediante SVD.



Figura 2: Prueba 1 con SVD utilizando rango reducido

El tiempo que duró para el reconocimiento de la figura 2 fue de 45.731 segundos.

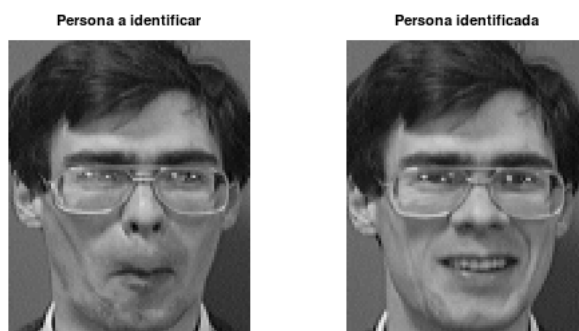


Figura 3: Prueba 2 con SVD utilizando rango reducido

El tiempo que duró para el reconocimiento de la figura 3 fue de 40.158 segundos.

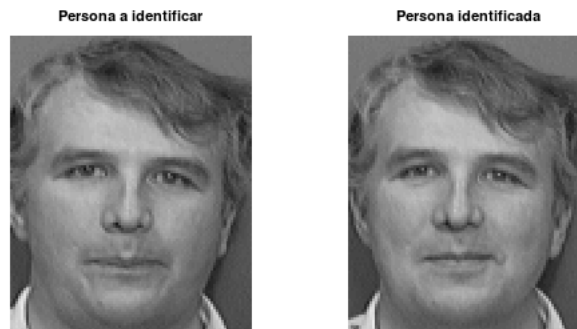


Figura 4: Prueba 3 con SVD utilizando rango reducido

El tiempo que duró para el reconocimiento de la figura 4 fue de 46.237 segundos.

Ahora se muestran algunos de los resultados que se obtuvieron con el algoritmo de reconocimiento de rostros mediante la mejora con Proyección Bilateral Aleatoria.



Figura 5: Prueba 1 con Proyección Bilateral Aleatoria

El tiempo que duró para el reconocimiento de la figura 5 fue de 41.667 segundos.

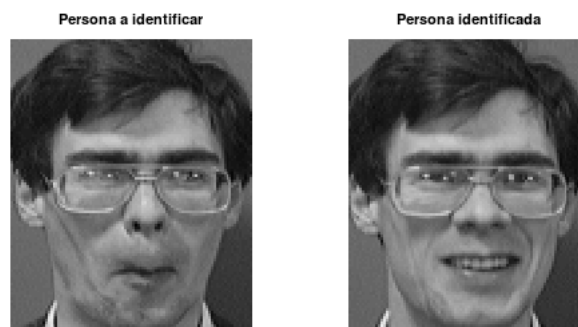


Figura 6: Prueba 2 con Proyección Bilateral Aleatoria

El tiempo que duró para el reconocimiento de la figura 6 fue de 41.442 segundos.

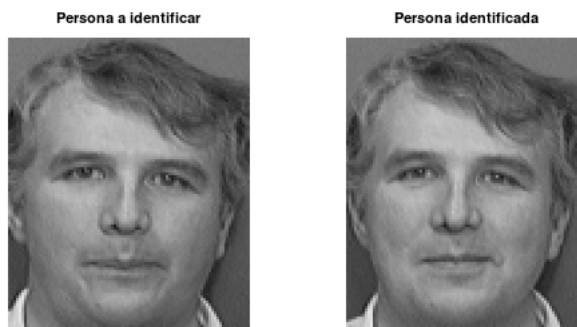


Figura 7: Prueba 3 con Proyección Bilateral Aleatoria

El tiempo que duró para el reconocimiento de la figura 7 fue de 43.143 segundos.

Como se pudo apreciar en los resultados obtenidos, con la implementación del algoritmo de Proyección Bilateral Aleatoria, los tiempos de procesamiento se redujeron en 2 de los casos pero en el otro el tiempo obtenido fue peor en comparación con el mostrado cuando se utilizó el algoritmo SVD con rango reducido. Por lo cual, se puede decir, que la mejora implementada no funcionó en la totalidad de los casos y esto podría deberse a que el procedimiento y los datos sobre los que se trabajaron no necesitaban un cálculo computacional pesado y por lo tanto la diferencia no fue tan marcada. Por otro lado, la iluminación juega un papel fundamental en la capacidad de reconocimiento de rostros de estos algoritmos porque estos se basan en un análisis de los valores de los píxeles de la imagen y por tanto si hay cambios importantes en la iluminación del rostro, aunque se trate de la misma persona no se tomará como tal porque los valores de los píxeles no serán similares.

Finalmente es importante mencionar que algunos resultados no coincidieron de manera correcta, probablemente porque la distribución de los valores de los píxeles son muy similares en algunas de las imágenes, a continuación se muestra un caso de esto.



Figura 8: Error en el reconocimiento

Referencias

- [1] GOLUB, G.H. y VAN LOAN, C.F. (2013): Matrix Computations, Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press. Baltimore.
- [2] HORN, R.A. y JOHNSON, C.R. (1990): Matrix Analysis, Cambridge University Press. Cambridge
- [3] J. J. Fallas Monge, J. Chavarría Molina, and P. Soto Quirós, “Descomposición en valores singulares de una matriz: un repaso por los fundamentos teóricos y sus aplicaciones en el procesamiento de imágenes,” Revista Investigación Operacional, pp. 1–27.