

IDENTIFICACIÓN APROXIMADA DE SEÑALES EN RECONOCIMIENTO DE PATRONES

WILLIAM FUNEZ

RESUMEN. En este artículo estudiamos un sistema de identificación y clasificación de señales que utiliza el algoritmo de descomposición en valores singulares (SVD) y análisis de componentes principales (PCA). El algoritmo se basa en un enfoque de “eigenfaces” para los primeros experimentos realizados, que representa un método de PCA en el que se utiliza un pequeño conjunto de características significativas para describir la variación entre las imágenes. Se muestran resultados experimentales para diferentes números de “eigenfaces” para verificar la viabilidad del método propuesto.

ABSTRACT. In this article we study a signal identification and classification system which uses the Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) algorithm. The algorithm is based on an eigenfaces approach for the first performed experiments, which represents a PCA method in which a small set of significant features is used to describe the variation between images. Experimental results are shown for different numbers of eigenfaces to verify the feasibility of the proposed method.

1. INTRODUCCIÓN

Una de las demostraciones más llamativas de la descomposición en valores singulares (SVD) y análisis de componentes principales (PCA) es la identificación o clasificación de imágenes. En este problema, PCA (es decir, SVD en datos de resta de la media) se aplica a un conjunto de imágenes para extraer las correlaciones más dominantes entre las imágenes. El resultado de esta descomposición es un conjunto que se le llama “eigenfaces” que definen un nuevo sistema de coordenadas donde las imágenes pueden representarse en estas coordenadas tomando el producto escalar con cada una de las principales componentes [9]. El problema llamado “eigenface” fue estudiado por primera vez por Sirovich y Kirby en [11]. Su aplicación al reconocimiento facial automatizado fue presentado por Turk y Pentland en [10]. Este problema se resuelve mediante PCA, la cual es una técnica de proyección que encuentra un conjunto de vectores de proyección diseñados para que los datos proyectados retengan la mayor parte de la información sobre los datos originales. Los vectores más representativos son los autovectores correspondientes a los autovalores más altos de la matriz de covarianza [9]. Este método reduce la dimensionalidad del espacio de datos al proyectar datos desde el espacio de dimensión M al espacio de dimensión N , donde $N \ll M$. En la siguiente sección se presenta la justificación

Fecha: Abril 19, 2021.

Palabras y frases clave. Descomposición en valores singulares, análisis de componentes principales, identificación, eigenfaces.

Asesor Dr. Fredy Vides.

del proyecto, donde se dará la importancia y líneas de investigación relacionadas con el proyecto.

2. JUSTIFICACIÓN

La importancia de este trabajo recae en la creciente intersección de los métodos basados en datos, optimización, estadística y otros temas de la matemática. La mayoría de los programas de pregrado en ingeniería y los campos de la ciencia en la UNAH, tienen poca exposición a los métodos de datos y optimización, el objetivo es proporcionar una introducción a la ciencia de datos. Se ha elegido SVD y PCA por su relevancia, simplicidad y generalidad, esto ya que cada vez más los investigadores están utilizando enfoques basados en datos.

El descubrimiento basado en datos está revolucionando actualmente la forma en que modelamos, predecimos y controlamos sistemas complejos. Los problemas científicos y de ingeniería más apremiantes de la era moderna no son basados en principios básicos, esto hace que cada vez más los investigadores recurran a enfoques basados en datos para una amplia gama de sistemas complejos como, la turbulencia, el cerebro, el clima, la epidemiología, las finanzas, la robótica, la autonomía, entre otros [9]. Estos sistemas son típicamente no lineales, dinámicos, de escala múltiple en espacio y tiempo, de alta dimensión, con patrones dominantes que deben ser modelados para el objetivo eventual de detección, predicción, estimación y control [2, 7].

Las aplicaciones que se presentarán en este trabajo será la identificación y detección de imágenes, ya que uno de los enfoques de PCA más simples y efectivos utilizados en los sistemas de identificación de imágenes es el llamado enfoque de “eigenfaces”. Este enfoque transforma los rostros en un pequeño conjunto de características esenciales, las “eigenfaces”, que son los componentes principales del conjunto inicial de imágenes de aprendizaje (conjunto de formación), en la sección 5 se brinda más detalles sobre la idea principal de lo son las “eigenfaces”.

La identificación se realiza proyectando una nueva imagen en el subespacio de “eigenfaces”, después de lo cual la persona se clasifica comparando su posición en el espacio de eigenfaces con la posición de los individuos conocidos [11]. La ventaja de este enfoque sobre otros sistemas de identificación es su simplicidad, velocidad e insensibilidad a los cambios pequeños o graduales en el rostro. El problema se limita a los archivos que se pueden utilizar para identificar el rostro. Es decir, las imágenes deben ser vistas de manera frontal y vertical. También se utiliza la modificación de los algoritmos en [3] y un resultado presentado en [1].

La teoría de aproximación esparcida (representación esparcida) trata con soluciones esparcidas para sistemas de ecuaciones lineales, las técnicas para encontrar éstas soluciones y explotarlas en aplicaciones han encontrado un amplio uso en el procesamiento de imágenes en [4], en este caso se presentará una aplicación mediante SVD para clasificación de imágenes. Para mejorar ciertas debilidades de este método se desarrollará análisis robusto de componentes principales (RPCA), que busca descomponer una matriz de datos A en una matriz estructurada L de “low-rank” y una matriz esparcida S [5]. A continuación se mostrará la líneas de investigación entre las cuales este proyecto está relacionado.

La Universidad Nacional Autónoma de Honduras (UNAH), estipula como uno de sus ejes de investigación el estudio de la Población, dentro de éste se define como

tema prioritario de investigación, entre otros, ciencia, de esta manera, el presente estudio se enmarca en este eje prioritario. Considerando que éste artículo cubre el análisis de datos, este es un tema de interés enfocado en la línea de investigación de optimización matemática, análisis numérico de la Maestría en Matemática con Orientación a Ingeniería de la UNAH, ya que los sistemas suelen ser no lineales, dinámicos y de alta dimensión, con patrones dominantes que deben caracterizarse y modelarse para el objetivo eventual de detección, predicción, estimación y control. En la siguiente sección se expone una perspectiva histórica sobre como se ha desarrollado la temática del artículo desde sus inicios hasta como se logró llegar a los conceptos mas modernos.

3. ANTECEDENTES

La SVD tiene una larga y rica historia, que va desde los primeros trabajos de desarrollo de los fundamentos teóricos hasta el trabajo moderno sobre estabilidad y eficiencia computacional. Hay una excelente revisión histórica de Stewart en [6], que proporciona contexto y muchos detalles importantes. La revisión se centra en los primeros trabajos teóricos de Beltrami y Jordan (1873), son los progenitores de la SVD, Beltrami en virtud de la primera publicación y Jordan por la integridad y elegancia de su tratamiento. La contribución de Beltrami apareció en el *Journal of Mathematics for the Use of the Students of the Italian Universities* y su propósito era alentar a los estudiantes a familiarizarse con las formas bilineales. Sylvester (1889), escribió dos documentos sobre el tema de la SVD, también una nota al pie de página que aparece al final de un artículo en *The Messenger of Mathematics*, en el documento Sylvester describe un algoritmo iterativo para reducir una forma cuadrática a una forma diagonal, en la nota señala que se puede usar una iteración análoga para diagonalizar una forma bilineal. Schmidt (1907), logra que la historia ahora pase del dominio del álgebra lineal a las ecuaciones integrales, uno de los temas candentes de las primeras décadas del siglo XX, en el tratamiento de ecuaciones integrales con núcleos asimétricos, Erhard Schmidt (de la fama de Gram-Schmidt y un estudiante de Hilbert) introdujo el análogo de dimensión infinita de la SVD. Pero fue más allá de la mera existencia de la descomposición al mostrar cómo se puede utilizar para obtener aproximaciones óptimas de low-rank a un operador, al hacerlo, transformó la SVD de una curiosidad matemática a una importante herramienta teórica y computacional. Autonne (1913), extendió la descomposición a matrices complejas. Hotelling (1933), como alternativa al análisis factorial (técnica estadística de reducción de datos usada para explicar las correlaciones entre las variables observadas en términos de un número menor de variables no observadas llamadas factores). Eckart (1936) y Young (1939) lo extendieron a matrices rectangulares y redescubrieron el teorema de aproximación de Schmidt, que a menudo (e incorrectamente) se llama el teorema de Eckart-Young. La SVD fue introducida en el análisis numérico por Golub y Kahan (1965). En las siguientes secciones se dará una breve descripción de SVD y propiedades que se utilizarán para el desarrollo del proyecto.

4. DESCRIPCIÓN DE SVD

Como en este proyecto la información que necesitamos requiere ser almacenada en una matriz A , que a su vez el propósito principal es representar e identificar la

información de dicha matriz, se hará uso de la técnica de SVD. En términos generales la SVD es una factorización matricial que nos proporciona ciertas características de la matriz A , también es una generalización del teorema espectral, ya que en este caso podemos hacer la descomposición para cualquier matriz rectangular y no simétrica, nos permite describir el mapeo lineal asociado a la matriz A mediante el sistema $Ax = b$.

En términos generales para cualquier matriz $A \in \mathbb{R}^{n \times m}$ se puede obtener una factorización de la forma $A = U\Sigma V^T$, donde $U \in \mathbb{R}^{n \times n}$, $V \in \mathbb{R}^{m \times m}$ son matrices ortogonales y $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$ son los valores singulares de A , se tiene que $r = \text{rank}A$, esto equivale al Teorema 2.6.3 de [1]. Una vez obtenida la SVD es posible acceder a una gran cantidad de información de la matriz A , por ejemplo el rango de la matriz, la norma espectral y el número de condición [1, 2]. También podemos resolver sistemas de ecuaciones que consideren a la matriz A como la matriz de coeficientes y es importante en algunos problemas de optimización, estos problemas nos ayudarán a obtener una mejor aproximación de “low-rank”.

Otra propiedad que se puede extraer de la SVD es la asociada con sus normas, ya que para la SVD los valores singulares juegan un papel muy importante, se tendrá que hacer uso de lo siguiente [1]:

1. Norma Espectral: $\|A\|_2 = \sigma_1$.
2. Norma de Frobenius: $\|A\|_F^2 = \sum_{i=1}^n \sigma_i^2$.

En la siguiente sección se presenta la idea principal detrás de las “eigenfaces”, su cálculo y como se representan en una base.

5. IDEA PRINCIPAL DETRÁS DE LAS EIGENFACES

Suponga que Γ es un vector de tamaño $N^2 \times 1$, correspondiente a una imagen I de tamaño $N \times N$, donde la imagen tiene N píxeles de alto y N píxeles de ancho. La idea es representar Γ ($\Phi = \Gamma - \text{cara media}$) en un espacio de baja dimensión:

$$(5.1) \quad \tilde{\Phi} - \text{media} = \sum_{i=1}^K w_i u_i$$

Cálculo de las Eigenfaces. La ecuación (5.1) representa las caras sobre una base se da más detalle de esta ecuación al final de esta sección, a continuación se muestra el proceso de como realizar el cálculo de las “eigenfaces”:

Paso 1: Se obtendrá las imágenes de las caras I_1, I_2, \dots, I_M .

*Obs. Las imágenes de la cara deben estar centradas y deben tener el mismo tamaño [4].

Paso 2: Representar cada imagen I_i como un vector Γ_i .

Paso 3: Calcular el vector de cara promedio Ψ :

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i$$

Paso 4: Restar la cara media

$$\Phi_i = \Gamma_i - \Psi.$$

Paso 5: Calcular la matriz de covarianza C :

$$C = \frac{1}{M} \sum_{i=1}^M \Phi \Phi^T = AA^T, \quad (\text{Matrix de } N^2 \times N^2),$$

$$\text{donde } A = [\Phi_1 \cdots \Phi_M], \quad (\text{Matrix de } N^2 \times M).$$

Paso 6: Calcular los autovectores u_i de AA^T

Paso 6.1: Considerando la matriz $A^T A$ (de tamaño $M \times M$).

Paso 6.2: Calcular los autovectores v_i de $A^T A$.

$$A^T A v_i = \mu_i v_i$$

$$AA^T A v_i = \mu_i A v_i$$

$$C A v_i = \mu_i u_i \text{ donde } u_i = A v_i.$$

Por tanto, AA^T y $A^T A$ tienen los mismos valores propios y sus vectores propios están relacionados de la siguiente manera: $u_i = A v_i$.

Obs. 1. AA^T puede tener hasta N^2 autovalores y autovectores.

Obs. 2. $A^T A$ puede tener hasta M autovalores y autovectores.

Obs. 3. Los M autovalores de $A^T A$ (junto con sus correspondientes autovectores) corresponden a los M autovalores más grandes de AA^T (junto con sus correspondientes autovectores).

Paso 6.3: Calcular los M mejores vectores propios de AA^T : $u_i = A v_i$.

Obs. Normalizar u_i tal que $\|u_i\| = 1$

Paso 7: Mantener solo K autovectores (correspondientes a los K autovalores más grandes)

Representar las caras sobre una base. Cada cara (menos la media) Φ_i en el conjunto se puede representar como una combinación lineal de los mejores vectores propios de K :

$$\tilde{\Phi}_i - \text{media} = \sum_{i=1}^K w_i u_i, \quad (w_i = u_i^T \Phi_i),$$

cada cara normalizada Φ_i está representada en una base por un vector

$$\Omega_i = [w_1^i \cdots w_K^i]^T, \quad i = 1, \dots, M.$$

En la siguiente sección se presenta el uso de “eigenfaces” para identificación y detección de imágenes.

6. IDENTIFICACIÓN Y DETECCIÓN

El análisis de componentes principales (PCA) transforma un conjunto de datos obtenidos de variables posiblemente correlacionadas en un conjunto de valores de variables no correlacionadas llamadas componentes principales [9]. El número de componentes puede ser menor o igual al número de variables originales. La primera componente principal tiene la varianza más alta posible, y cada uno de las componentes siguientes tiene la varianza más alta posible bajo la restricción de que debe ser ortogonal a la componente anterior. Se desea encontrar las componentes principales, en este caso vectores propios de la matriz de covarianza de las imágenes. Lo primero que debemos hacer es crear un conjunto de datos de entrenamiento. La imagen I_i se puede representar como un vector concatenando filas [10]. Lo antes

mencionado es lo que se mostró en la sección 5, a continuación se presentará el uso de “eigenfaces” para identificación y detección de imágenes con sus correspondientes pasos.

Algorithm 1 Identificación.

Input: Dada una imagen desconocida Γ (centrada y del mismo tamaño que el conjunto ingresado al inicio), se debe seguir lo siguiente:

- 1: Normalizar Γ : $\Phi = \Gamma - \Psi$
- 2: Proyectar en el eigenspace

$$\tilde{\Phi}_i - \text{media} = \sum_{i=1}^K w_i u_i, \quad (w_i = u_i^T \Phi_i)$$

- 3: Representar Φ como: $\Omega = [w_1 \cdots w_K]^T$
 - 4: Encontrar $e_r = \min_l \|\Omega - \Omega^l\|$
 - 5: Si $e_r < T_r$, entonces Γ se reconoce como la cara l del conjunto.
-

Algorithm 2 Detección.

Input: Dada una imagen desconocida Γ (centrada y del mismo tamaño que el conjunto ingresado al inicio), se debe seguir lo siguiente:

- 1: Calcular $\Phi = \Gamma - \Psi$
- 2: Calcular

$$\tilde{\Phi}_i - \text{media} = \sum_{i=1}^K w_i u_i, \quad (w_i = u_i^T \Phi_i)$$

- 3: Calcular $e_d = \|\Phi_i - \tilde{\Phi}_i\|$
 - 4: Si $e_d < T_d$, entonces Γ es la cara.
-

En la siguiente sección se presenta un experimento numérico que muestra como resolver la identificación y detección de imágenes.

7. EXPERIMENTO NUMÉRICO

El experimento numérico muestra como resolver la identificación de imágenes, se toma un grupo de imágenes de un hombre en este caso del actor Sylvester Stallone y otro de una mujer en este caso la cantante Taylor Swift se toma 20 imágenes de cada persona (ver figura 1), las cuales fueron extraídas de internet, recordar que todas las imágenes deben estar centradas y tener el mismo tamaño, cada imagen tiene 200 píxeles de alto y 175 píxeles de ancho donde cada una de las imágenes que se utiliza se ha reformado en un vector columna con $200 \times 175 = 35,000$ elementos, a continuación se presenta el algoritmo 1 el cual nos permite identificar las imágenes en dos grupos (ver figura 4), mediante las técnicas descritas en las secciones anteriores, también nos permite detectar una imagen ingresada después del conjunto inicial, para lo antes mencionado se utiliza GNU Octave con técnicas computacionales utilizadas en [12].



FIGURA 1. Algunas imágenes del conjunto de las 20 imágenes que se cargan para cada grupo para realizar el experimento.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% ALGORITMO 1  
% Identificacion  
% Deteccion
```



FIGURA 2. Proyecta la imagen sobre una base.



(A) Imagen de una mujer la cual no es Taylor Swift, además cuenta con daños en su rostro, notaremos que se aleja en la identificación.



(B) Imagen de Stallone, solo que está imagen cuenta con rasgos distintos al grupo de 20 imágenes que se ingresó al inicio.

FIGURA 3. Imágenes que se desean detectar, para cada uno de los distintos grupos.

```
clear all
close all
clc

% Tamaño de cada imagen
m = 200;
n = 175;

% Numero de imagenes (muestra)
N = 20;

avg = zeros(m*n,1); % la cara promedio
```

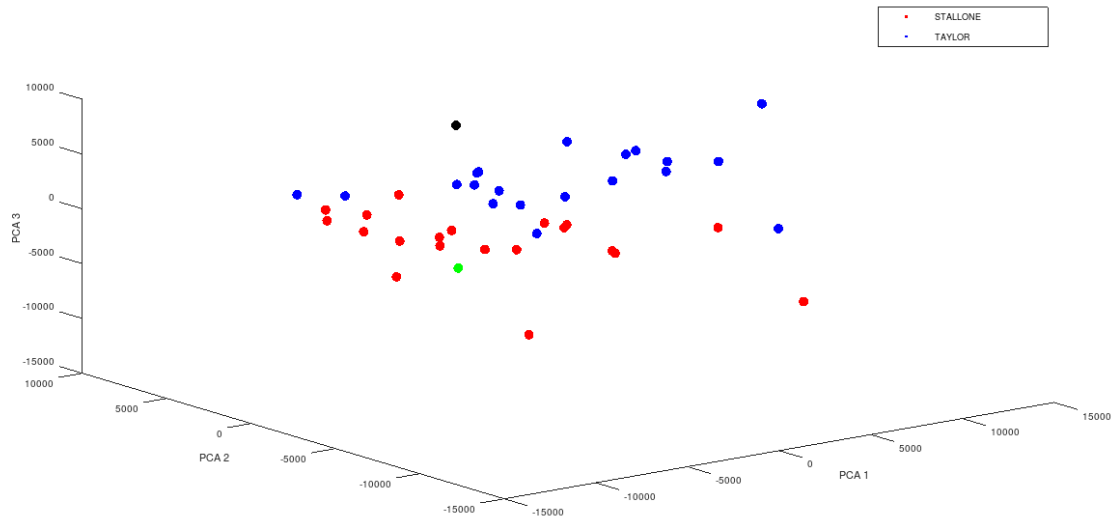



FIGURA 4. En el siguiente gráfico se muestra la identificación de los dos grupos, los puntos de color rojo son los correspondientes a Stallone y los de color azul a Taylor Swift, donde las imágenes ingresadas al final están de colores distintos para la imagen de Stallone con rasgo distinto se tiene en color verde la cual se mantiene en el grupo, así que detecta bien las características principales del rostro, para la imagen de la mujer con daños en el rostro que no es Taylor Swift se tiene en color negro podemos ver que se aleja del grupo, lo cual está bien, ya que no corresponde al grupo de imágenes de Taylor Swift.

```
A = [];
```

```
% Cargar imagenes de Sylvester Stallone
count = 0;
for j = 1:N
    figure(1)
    ff = ['faces/stallone',num2str(j,'%02d'),' .jpg'];
    u = imread(ff); % Leer la imagen en una matriz
    imshow(u)
    if(size(u,3)==1)
        M=double(u);
    else
        M=double(rgb2gray(u));
    end
    pause(0.1);
    R = reshape(M,m*n,1);
    A = [A,R];
    avg = avg + R;
    count = count + 1;
```

```

end
%% Cargar imagenes de Taylor Swift
for j = 1:N
    figure(1)
    ff = ['faces/taylor',num2str(j,'%02d'),' .jpg'];
    u = imread(ff); % Leer la imagen en una matriz
    imshow(u)
    M = double(u(:,:,1));
    R = reshape(M,m*n,1);
    A = [A,R];
    pause(0.1);
    avg = avg + R;
    count = count + 1
end
avg = avg /count;

%% Calcular el "promedio" de cada cara
avgTS = uint8(reshape(avg,m,n));
figure(1), imshow(avgTS);

%% Centrar las imagenes de la muestra en el "origen"
figure(1)
for j = 1:2*N
    A(:,j) = A(:,j) - avg;
    R = reshape(A(:,j),m,n);
    imshow(R);
    set(gcf,'Position',[100 900 500 500])
    pause(.1);
end

%% Calculando la SVD
[U,S,V] = svd(A,'econ');
Phi = U(:,1:2*N);
Phi(:,1) = -1*Phi(:,1);
figure(2)
count = 1;
for i=1:3
    for j=1:3
        subplot(3,3,count)
        imshow(uint8(25000*reshape(Phi(:,count),m,n)));
        count = count + 1;
    end
end

%% Proyectar cada imagen sobre la base
for j = 1:N

```

```

        imvec = A(:,j);
        STAL(:,j) = imvec'*Phi(:,1:3);
    end
for j = 1:N
    imvec = A(:,N+j);
    TLR(:,j) = imvec'*Phi(:,1:3);
end

figure(3)

plot3(STAL(1,:),STAL(2,:),STAL(3,:), 'r.', 'MarkerSize',30)
hold on
plot3(TLR(1,:),TLR(2,:),TLR(3,:), 'b.', 'MarkerSize',30)
xlabel('PCA 1')
ylabel('PCA 2')
zlabel('PCA 3')
legend('STALLONE', 'TAYLOR')

%% Agrega algunas imagenes fuera del conjunto
u = imread('faces/pruebastallone1.jpg');
figure(4)
subplot(1,2,1)
imshow(u);
u = double(rgb2gray(u));
ustal = reshape(u,m*n,1)-avg;
stalpts = ustal'*Phi(:,1:3);
v = imread('faces/pruebaconjunto2.jpg');
subplot(1,2,2)
imshow(v);
v = double(rgb2gray(v));
vterm = reshape(v,m*n,1)-avg;
termpts = vterm'*Phi(:,1:3);
%%
figure(3)
plot3(stalpts(1),stalpts(2),stalpts(3), 'g.', 'MarkerSize',30)
plot3(termpts(1),termpts(2),termpts(3), 'k.', 'MarkerSize',30)

```

En la siguiente sección se presenta las conclusiones del trabajo realizado para resolver la identificación y detección de imágenes, utilizando el análisis de componentes principales.

8. CONCLUSIÓN

Se utilizó una base de datos que contiene 40 imágenes de 2 personas diferentes (20 imágenes por persona), utilizando el análisis de componentes principales para realizar este proceso, de los resultados obtenidos mediante el experimento numérico podemos notar que la identificación y detección de imágenes es aceptable, pero se puede realizar una mejora a este método, haciendo una pequeña observación también que la tasa de identificación aumenta con el número de imágenes por persona. Es obvio que si la distancia mínima entre la imagen de prueba y otras

imágenes es cero, la imagen de prueba coincide completamente con la imagen de la base. Si la distancia es mayor que cero, pero menor que cierto umbral, es una persona conocida con otra expresión facial, de lo contrario es una persona desconocida.

9. TRABAJO A FUTURO

Para mejorar el sistema de identificación de señales utilizado en este trabajo, en esta mejora se utilizará el algoritmo de descomposición en modo dinámico (DMD), donde esta técnica se basa en la descomposición ortogonal adecuada (POD), que utiliza la (SVD) computacionalmente eficiente de modo que escala bien para proporcionar reducción de dimensionalidad en sistemas de alta dimensión. La DMD es una técnica que permite extraer las características de flujo dinámicamente relevantes de datos experimentales o numéricos. En esta técnica, un mapeo lineal se construye a partir de una serie de datos resueltos en el tiempo, para este tipo de experimento se considerará la descomposición de Koopman en relación con la DMD, considerando la ecuación no lineal de Schrödinger (NLS).

REFERENCIAS

1. Horn, R. A. (1986). *Topics in Matrix Analysis*. USA: Cambridge University Press.
2. Golub, G. H. (1996). *Matrix Computation*. USA: Johns Hopkins University Press.
3. Liberty, E., Woolfe F., Tygert M., Martinsson P., Rokhlin V. (2007). Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences - PNAS*. 104(51): 20167-20172. doi.org/10.1073/pnas.0709640104
4. Yang, A., Ganesh, A., Wright, J., Sastry, S., Ma, Y. (2009). Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 31(2):210-27. doi.org/10.1109/TPAMI.2008.79
5. Candès, E. J., Wright, J., Li, X., Ma, Y. (2011). Robust principal component analysis?. *Journal of the ACM*. 58(3):1-37. doi.org/10.1145/1970392.1970395
6. Stewart, G. W. (1993). On the early history of the singular value decomposition. *SIAM Review*. 35(4):551-566. doi.org/10.1137/1035134
7. Trefethen, L. (1997). *Numerical Linear Algebra*. USA: Society for Industrial and Applied Mathematics (SIAM).
8. Ayubi, J. (2014). Lossy Color Image Compression Based on Singular Value Decomposition and GNU GZIP. *ACSIJ Advances in Computer Science: an International Journal*. 3(3): 2322-5157.
9. Nathan Kutz, J. (2013). *Data-Driven Modeling Scientific Computation. Methods for Complex Systems Big Data*. Oxford University Press.
10. Pentland, A., Turk, M. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*. 3(1):71-86. doi.org/10.1162/jocn.1991.3.1.71.
11. Sirovich, L., Kirby M. (1987). Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*. 4(3):519-524. doi.org/10.1364/JOSAA.4.000519
12. Vides, F. (2021). *Métodos Numéricos y Modelación Computacional*. <https://caddslab.github.io/HPCMMModeling.html>.

UNIVERSIDAD NACIONAL AUTÓNOMA DE HONDURAS - UNAH, ESCUELA DE MATEMÁTICA Y CIENCIAS DE LA COMPUTACIÓN, MAESTRIA EN MATEMÁTICA

Dirección actual: Tegucigalpa M.D.C, Honduras

Dirección de correo electrónico: william.funez@unah.hn