Universidad de La Coruña

Práctica 4 - Reconstrucción de Bases de Datos

Problemas inversos y Reconstrucción de imágenes

Miriam Gutiérrez Serrano

Índice

1. Introducción

En muchas aplicaciones científicas y tecnológicas a menudo las bases de datos están parcialmente destruidas o tienen huecos ('gaps'), es decir, falta parte de la información.

Afortunadamente, los datos que provienen de simulaciones numéricas se pueden aprovechar matemáticamente. En este proyecto, se explora una técnica de reconstrucción basada en Descomposición en Valores singulares (SVD), que nos permite identificar correlaciones en los datos y completar la información que falta.

Un ejemplo muy sencillo es el que se ha comentado en clase, unas imágenes con ruido, a las que gracias a la restauración podemos eliminar el ruido y mantener los detalles principales. O el del enunciado, donde las nubes dejan 'huecos' en los datos grabados por satélites de detección remota.

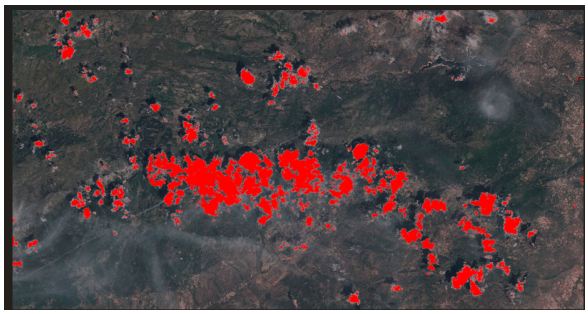

Figura 1: Imagen con huecos debido a las nubes.

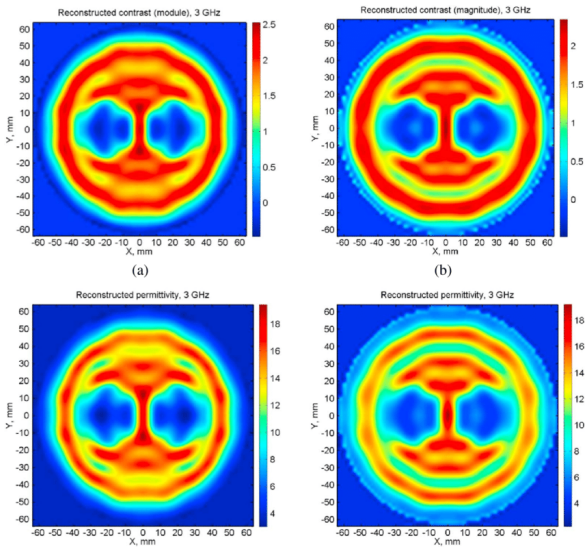

Figura 2: Reconstrucción de imágenes en Tomografía de Microondas.

Además de la teoría descrita en clase, cuando aplicamos SVD para reconstruir bases de datos con información faltante, también es necesario medir como de precisa es la construcción. Para esto, tenemos en cuenta los siguientes errores:

$$\text{RMSE} = \frac{1}{N}||\mathcal{A}^{\text{original}} - \mathcal{A}^{\text{reconstruida}}||_2, \quad \text{MaxE} = \max_{ij} |\mathcal{A}_{ij}^{\text{original}} - \mathcal{A}_{ij}^{\text{reconstruida}}|$$

donde $\mathcal{A}^{\text{original}}$ representa la base de datos original ('non-gappy') y $\mathcal{A}^{\text{reconstruida}}$ corresponde a su reconstrucción.

El error cuadrático medio (RMSE) mide la diferencia promedio entre los valores originales y los reconstruidos y MaxE hace referencia al error máximo cometido teniendo en cuenta todo elemento de la matriz.

El objetivo principal del proyecto es reconstruir bases de datos 'gappy', bases de datos a las que hemos eliminado de maneria aleatoria un porcentaje de elementos.

2. Construcción de bases de datos

Primero generamos las bases de datos a partir de una función definida. Se considera la función

$$f(x,y,z) = x^2[\sin(5\pi y) + 3\ln(x^3 + y^2 + z + \pi^2) - 1]^2 - 4x^2y^3(1-z)^{3/2} \\ +(x+z-1)(2y-z)\cos(30(x+z))\ln(6+x^2y^2+z^3)$$

y la transformación $T : \mathbb{R} \longrightarrow \mathbb{R}$ definida por

$$F(x,y,z) = T(f(x,y,z)) = 2(f(x,y,z) - f_{\min})/(f_{\max} - f_{\min}) - 1$$

con $0 \leq x, y, z \leq 1$. Ahora, para un valor fijo $z = z* \in [0,1]$ podemos construir la base de datos definiendo la matriz de la siguiente forma:

$$A_{ij} = F(x_i, y_j, z*)$$

donde $x_i = (i-1)\Delta x$, $y_j = (j-1)\Delta x$, con $i, j = 1, \ldots, N$ y $\Delta x = 1/(N-1)$.

Podemos contruir entonces, las matrices pedidas A_1, A_2, A_3 con $z = 0,0{,}5,1$ respectivamente. A continuación, tomando un cierto porcentaje w de elementos creamos las 3 bases de datos 'gappy', es decir, eliminamos un porcentaje de las matrices aleatoriamente. Se obtienes las siguientes matrices:

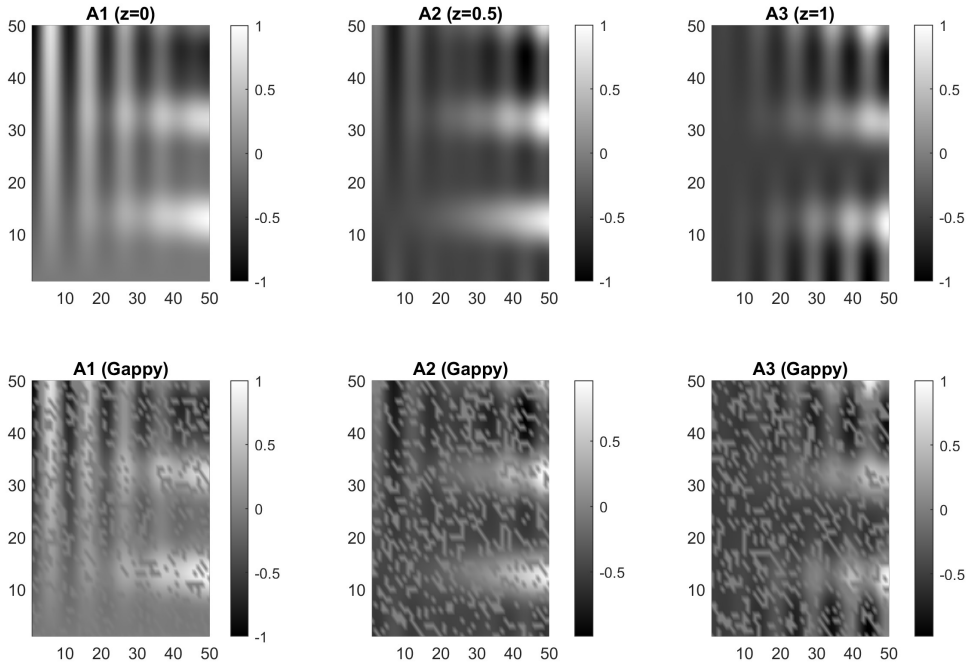

Figura 3: Matrices A_1, A_2, A_3 y matrices 'gappy'$\tilde{A}_1, \tilde{A}_2, \tilde{A}_3$

3. Algoritmo de reconstrucción

Sea A_1 la matriz construida posteriormente. Vamos a reconstruir la matriz entera aplicando el siguiente algoritmo iterativo.

- *Paso* $s = 0$. Define una reconstrucción inicial $\mathcal{A}^0$.
- *Paso* $s \geq 1$. Aplica una reconstrucción SVD (truncada, con un número fijo m de modos) a la matriz modificada $\hat{\mathcal{A}}^s = \mathcal{A}^{s-1}$ en los puntos 'gappy' (aquí $\mathcal{A}^{s-1}$ corresponde a la matriz reconstruida mediante SVD en la iteración $s-1$) y $\hat{\mathcal{A}}^s = \mathcal{A}$ en los puntos 'non-gappy'.

Para la realización del algoritmo iterativo hemos fijado un número maximo de iteraciones y un umbral de toleracia mayor que el error RMSE. Además, se han definido dos reconstrucciones de $\mathcal{A}^0$ iniciales: una que usa la media de los valores conocidos y otra que usa valores aleatorio entre -1 y 1 en los huecos. A continuación, muestro el código correspondiente.

```
     %Reconstrucción SVD truncada, con un número fijo m de modos

function A_rec = truncated_svd (A_gappy, m, max_iter, tol)
    %Algoritmo iterativo para reconstrucción de matrices 'gappy' usando SVD

    %Parámetros:
    %   -A_gappy: Matriz con un cierto porcentaje de elementos vacíos
    %   -m: Número de modos fijos
    %   -tol: tolerancia para la convergencia
    %   -max_iter: número maximo de iteraciones

    [~,N] = size(A_gappy);


    %Elementos eliminados de la matriz, valores 'gappy'
    elems = isnan(A_gappy);

    %Reconstrucción inicial A_0
    A_0 = A_gappy;

    % Opción 1: Usar la media de los valores conocidos (actual)
    valor_medio = mean(A_gappy(~elems), 'all');
    A_0(elems) = valor_medio; %Rellenamos los huecos con la media para que no sean Nal

    % Opción 2: Usar valores aleatorios en los huecos
   %A_0(elems) = rand(size(A_gappy(elems))) * 2 - 1; % Números entre -1 y 1

    A_rec = A_0;
    iter = 0;
    RMSE_prev = inf;

    while iter < max_iter
        iter = iter + 1;

        %Aplicamos SVD a la matriz inicial
        [U, S, V] = svd(A_rec);
```

```
        % Extraer los valores singulares y ordenarlos
        valores_singulares = diag(S);
        [m_valores, idx] = sort(valores_singulares, 'descend');

        % Tomar los m valores singulares más grandes
        if m < length(m_valores)
            m_valores(m+1:end) = 0;
        end

        % Construir nueva matriz S con solo los m valores singulares más grandes
        S_trunc = zeros(size(S));
        for i = 1:min(m, length(m_valores))
            S_trunc(idx(i), idx(i)) = m_valores(i);
        end

        %Reconstruccion de los m modos principales
        A_s = U* S_trunc* V';

        %Restaurar los puntos 'non-gappy'
        A_s(elems) = A_gappy(elems);

         %Calculo error
        RMSE = (1/N) *norm(A_rec - A_s, 'fro'); %Norma frobenius, ||.||_2

        %Criterio de parada
        if RMSE < tol
            break;
        end
        RMSE_prev = RMSE;

        %Matriz reconstruida
        A_rec = A_s;
    end
end
```

4. Resultados

Finalmente, vamos a analizar la calidad de las reconstrucciones en función del porcentaje w de elementos 'gappy', del número m de modos retenidos y de la reconstrucción inicial $\mathcal{A}^0$.

Primero analicemos según el porcetaje de elementos eliminados. Fijando el número de modos para $m = 5$ hemos obtenido las siguiente matrices para $w = 0{,}1, 0{,}3, 0{,}7$

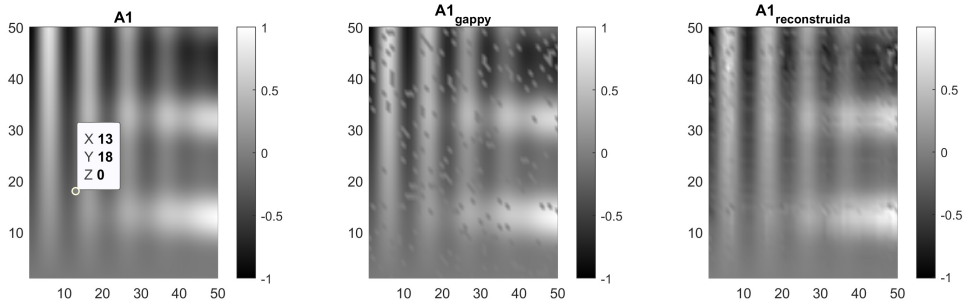

Figura 4: $w = 0{,}1$, RMSE $= 0{,}07595$, MaxE $= 0{,}98997$

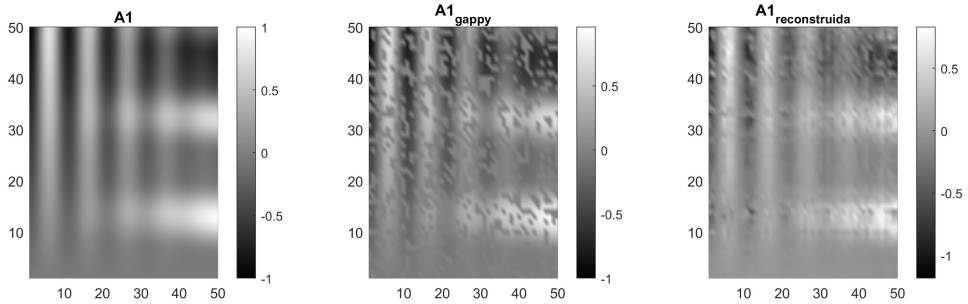

Figura 5: $w = 0{,}3$, RMSE $= 0{,}13667$, MaxE $= 0{,}9494$

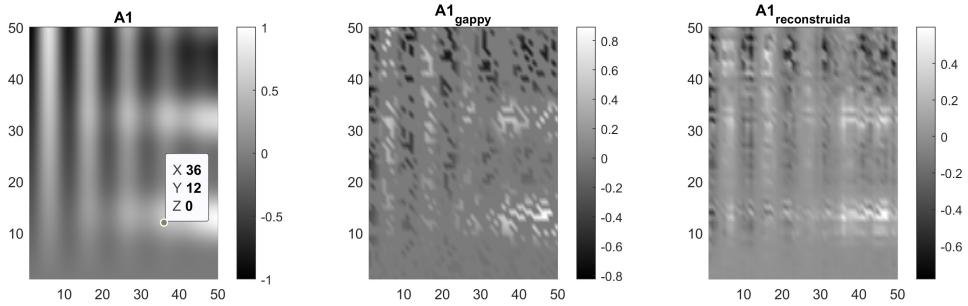

Figura 6: Evolución del error medio respecto w

A continuación, muestro una gráfica que representa la evolución del error en función de los valores de w en el intervalo $[0{,}1, 0{,}9]$.

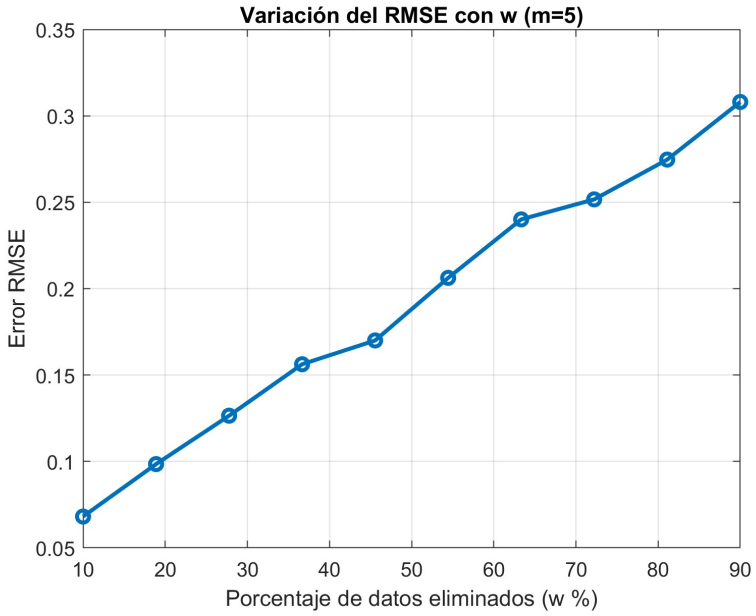

Figura 7

Como cabe esperar, cuanto más pequeño es el porcentaje w, la reconstrucción es más precisa, se puede observar también en la representación de las matrices. Por otra parte, a medida que w crece, la calidad disminuye porque el algoritmo tiene menos información con la que trabajar.

Estudiemos ahora como afecta el número de modos retenidos m, es decir, la cantidad de valores singulares que usamos en la reconstrucción. Fijamos $w = 0{,}3$ y reprensentamos las reconstrucciones para $m = 2, 8, 20$

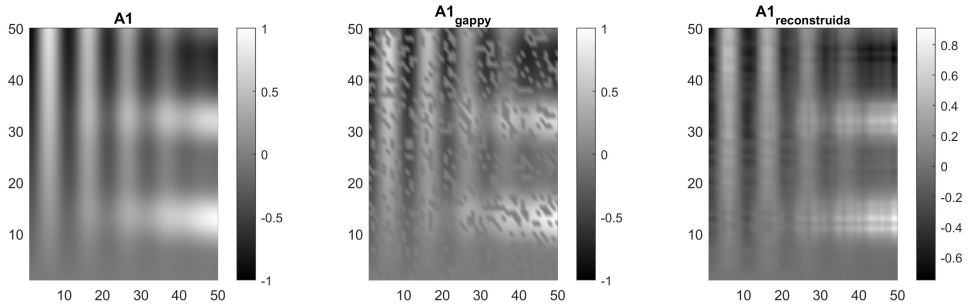

Figura 8: $m = 2$, RMSE $= 0{,}13667$, MaxE $= 0{,}9494$

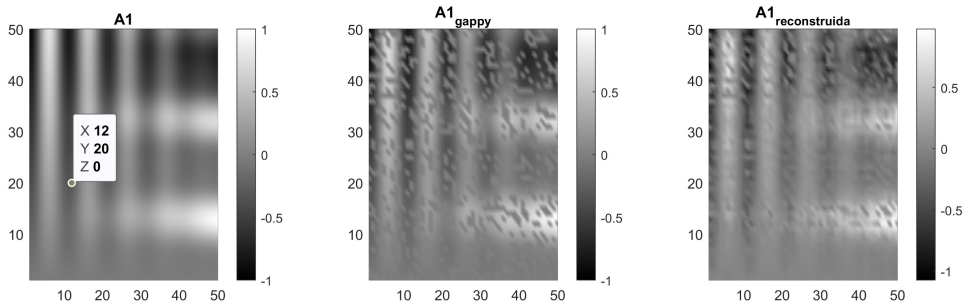

Figura 9: $m = 8$, RMSE $= 0{,}14932$, MaxE $= 1{,}0524$

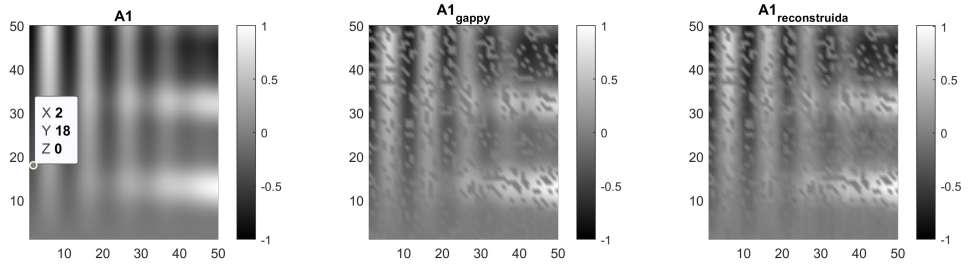

Figura 10: $m = 20$, RMSE $= 0{,}17274$, MaxE $= 1{,}0325$

A continuación, muestro una gráfica que representa la evolución del error en función de los valores de $m = 1, 2, \ldots, 9$

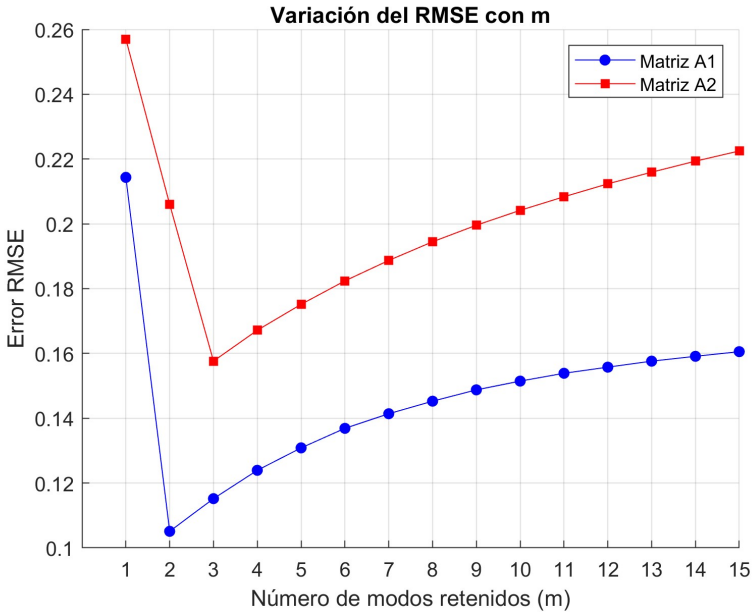

Figura 11

Para el caso del estudio de modos retenidos he optado por representar también el caso de la matriz A_2, para observar mejor que tanto valores muy pequeños de m como muy grandes proporcionan errores más notables. Podemos concluir, por lo tanto, que el algoritmo proporciona menos error para valores de m entre 3 y 5.

Por último, evaluamos la calidad de las reconstrucciones en función de la elección de la reconstrucción inicial.

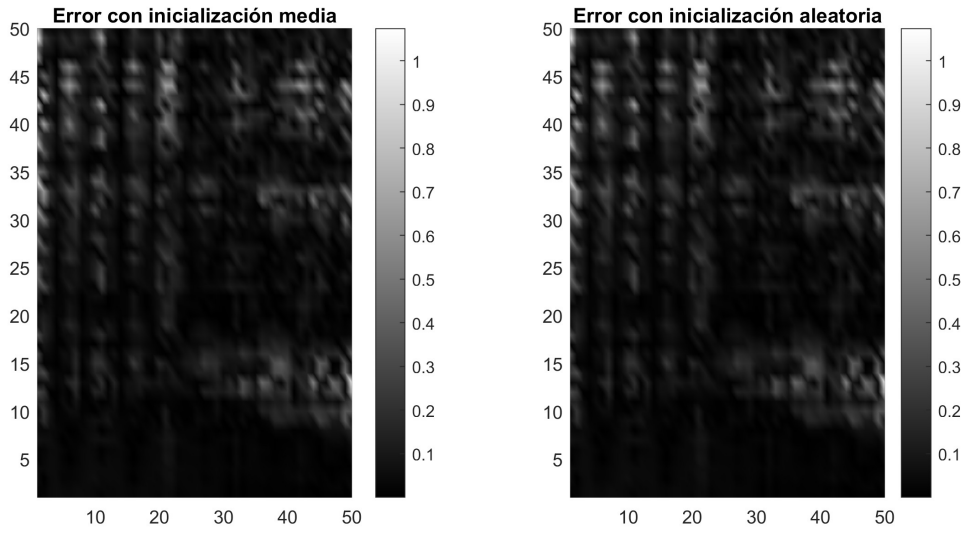

Figura 12: Izquierda: reconstrucción opción 1, Derecha: reconstrucción opción 2

En ambos casos, el error obtenido es el mismo:

$$\text{RMSE} = 0{,}1967$$

Además, a simple vista, las matrices reconstruidas no presentan diferencias significativas. Por lo tanto, podemos concluir que la elección de la reconstrucción inicial no influye de manera notable en el resultado final.

5. Conclusiones

En este proyecto, hemos analizado la reconstrucción de bases de datos mediante Descomposición de valores singulares. A lo largo del estudio, hemos evaluado distintos parámetros que afectan a la calidad de la reconstrucción, como el número de modos retenidos, el porcentaje de datos y la elección de la reconstrucción inicial.

Los resultados muestran que :

- La calidad de la reconstrucción disminuye a medida que aumenta el porcentaje de datos faltantes.
- Existe un número óptimo de modos retenidos m, ya que valores demasiado bajos o demasiado altos pueden afectar negativamente la reconstrucción.
- La elección de la reconstrucción inicial no tiene un impacto significativo en el error final, aunque sí puede influir en la velocidad de convergencia del algoritmo iterativo.

A. Anexo

A continuación, se presentan los códigos utilizados en este trabajo, los cuales se adjuntan como archivos separados. De todas formas se recogen todos los scripts y gráficas generadas en: Proyecto4.mlx.

A.1. BasesDeDatos_gappy.m

Este script genera las bases de datos originales y las versiones con datos faltantes.

A.2. truncated_svd.m

Implementa la reconstrucción de matrices incompletas mediante **SVD truncado**.

A.3. BaseReconstruida.m

Aplica el algoritmo y compara gráficamente las matrices **original, incompleta y reconstruida**.

A.4. truncated_svd2.m

Variante del algoritmo que **almacena el error RMSE** en cada iteración para evaluar la evolución de la reconstrucción.

A.5. variacion_A_0

Genera las gráficas correspondientes a cada reconstrucción y muestra el error.

A.6. variacion_m

Genera la gráfica correspondiente al error frente al numero de modos.

A.7. variacion_w

Genera la gráfica correspondiente al error frente porcentaje de elementos eliminados.

Los archivos correspondientes se adjuntan junto con este informe para su consulta y ejecución.