

UNIVERSIDAD POLITÉCNICA DE VALENCIA
DEPARTAMENTO DE MATEMÁTICA APLICADA
MÁSTER EN INVESTIGACIÓN MATEMÁTICA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

TRABAJO FIN DE MÁSTER

**”Descomposición en valores singulares y análisis
semántico latente”**

Realizado por: Maribel Mamani Roque

Dirigido por:

Jose Manuel Calabuig Rodríguez

Luis M. García Raffi

Septiembre, 2018

Índice

1. Introducción	4
1.1. Una breve reseña Histórica	5
2. La descomposición en valores singulares. Resultados teóricos	8
2.1. Definición del DVS	8
2.2. El teorema de descomposición	9
2.3. Teorema del rango aproximado	11
2.4. Ejemplo matemático	12
3. Aplicaciones de Descomposición en valores singulares	16
3.1. Aplicaciones matemáticas de la DVS	16
3.1.1. Problema lineal de mínimos cuadrados	16
3.1.2. Pseudoinversa	16
3.2. Algunas aplicaciones de la descomposición en valores singulares en tecnología	19
3.2.1. Compresión de imágenes	19
3.2.2. Compresión y filtrado de electrocardiograma	20
3.2.3. Reconocimiento de caras (eigenfaces)	21
3.2.4. Cocktail party problem	22
3.2.5. Sistema de recomendación	23
3.2.6. Análisis componentes principales	24
4. Aplicación de la DVS al análisis semántico latente	25
4.1. Análisis semántico Latente (ASL)	25
4.2. Matriz término-documento	25
4.3. Ejemplo de aplicación de DVS al ASL	25
5. Bibliografía	37
Apéndices	38
Apéndice A. Álgebra lineal	38
Apéndice B. Ortogonalidad y diagonalización	41
Apéndice C. Diagonalización ortogonal	43

Agradecimiento

En primer lugar quiero agradecerle al director académico del Máster, José Alberto Conejero, el gran apoyo intelectual y moral que me ha prestado, incluso en los momentos bajos en los que se hacía difícil continuar. Gracias a su capacidad de motivar y de reunir personas interesadas en una misma temática, ha salido esta tesis adelante. En segundo lugar, quisiera también agradecer a mis tutores Jose Manuel Calabuig Rodriguez y Luis M. Garcia Raffi por las aportaciones tan valiosas que me ha hecho. Sus consejos y comentarios han sido enormemente útiles y me han allanado mucho mi labor investigadora.

Por último, dedico esta tesis a mi hija Naomi, por entenderme la ausencia y la distancia durante el año académico, también a mi madre Rosario por su apoyo incondicional.

Resumen

La descomposición en valores singulares tiene numerosas aplicaciones tanto el campo de la matemática (aproximación lineal mediante mínimos cuadrados, cálculo de la matriz psudeoinversa), como en el de la ingeniería (compresión y filtrado de imágenes, de electrocardiogramas, análisis de componentes principales, reconocimiento facial entre otros). En este trabajo se estudia la aplicación de esta técnica en el conocido como análisis semántico latente. Bajo la idea de que las palabras que se utilizan en un mismo contexto tiene significados similares el análisis semántico latente permite identificar patrones entre palabras y títulos en una colección de textos. Esta técnica ayuda a resolver, matemáticamente, problemas lingüísticos como la sinonimia y la polisemia que aparecen en la indexación de los buscadores de términos.

PALABRAS CLAVES: Álgebra lineal, Descomposición en valores singulares (DVS), Análisis semántico latente (ASL)

1. Introducción

Desde un punto de vista matemático, la *Descomposición en valores singulares* (DVS), es una técnica de descomposición matricial que permite descomponer una matriz arbitraria (en nuestro caso con coeficientes reales, aunque la factorización es también posible en el cuerpo de los números complejos) como producto de tres matrices. Este tipo de factorizaciones tiene una gran cantidad de aplicaciones no sólo de carácter matemático (o estadístico) sino también en el mundo de la tecnología.

Muchos métodos de búsqueda de textos en las páginas web de Internet dependen de un emparejamiento de palabras “al pie de la letra” entre las palabras que busca el usuario y las que existen en las páginas web. Se plantean dos situaciones antitéticas: por un lado, el buscador encuentra términos homónimos y, por otro, los sinónimos le pasan desapercibidos. En el primer caso, son recuperadas páginas cuyo tema no nos interesa; en el segundo, son ignoradas páginas en las que estaríamos interesados.

Dado que los ordenadores no saben nada de lenguas ni de idiomas, se trata de dotarlos de un método matemático que “aprenda” el significado de las palabras que “viven” en el universo semántico constituido por todas las páginas web que existen en un momento dado, dispersas por el mundo. Es posible lograr este aprendizaje mediante *el análisis semántico latente* (en inglés “latent semantic index” o LSI) que desvela el significado oculto dado a las palabras en las páginas de la Red.

El objeto de estudio es resolver matemáticamente el problema de la *sinonimia* y de la *polisemia* en los métodos de indexación de textos. La idea central se basa en la agrupación de textos por conceptos (en lugar de hacerlo por palabras): palabras que se utilizan en un mismo contexto suelen tener significados similares y específicamente el desarrollo del caso de uso de aplicación de la descomposición en valores singulares al análisis semántico latente.

El trabajo fin de Máster está estructurado de la siguiente manera:

- **Capítulo 1.-** Se explica detalladamente la problemática y el objeto de estudio.
- **Capítulo 2.-** A continuación se incluyen los resultados teóricos más relevantes relacionados con dicha descomposición.
- **Capítulo 3.-** Se recogen las siguientes aplicaciones de la descomposición en valores singulares (DVS): Problema lineal de mínimos cuadrado (PLMC), pseudoinversa, compresión de imágenes, comprensión y filtrado de electrocardiograma (ECG), reconocimiento de caras (RC), cocktail party problem y sistema de recomendación

(SR) y análisis componentes principal (ACP).

- **Capítulo 4.-** Finalmente, el último capítulo presenta la aplicación de DVS al ASL con el programa en MATLAB que implementan los algoritmos paso a paso y la visualización gráfica.

Al final del documento se incluyen una serie de apéndices con definiciones y resultados de álgebra lineal.

1.1. Una breve reseña Histórica

El origen de los valores singulares se encuentra en el intento de los geómetras del siglo XIX por conseguir, en lenguaje actual, la reducción de una forma cuadrática a forma diagonal mediante cambios de base ortogonales. La primera contribución en este sentido parece ser de Eugene Beltrami, un geómetra diferencial italiano, que intentando promover entre sus estudiantes el gusto por el estudio de la formas bilineales escribió un artículo en la revista *Giornale di Matematiche ad Uso degli Studenti Delle Università*, [4].

Beltrami comienza con una forma bilineal $f(x, y) = x^T Ay$, donde A es una matriz real de orden n , y viene a demostrar que existen matrices ortogonales Q_1 y Q_2 , $n \times n$, tales que.

$$\Sigma = \text{Diag}(\sigma_1, \dots, \sigma_n),$$



Eugene Beltrami
(1835–1899)

es una matriz diagonal de números reales positivos, los cuales son las raíces cuadradas de los valores propios de $A^T A$ (o de AA^T , dado que estas dos matrices simétricas son semejantes). En realidad, Beltrami propone, más que demuestra, la descomposición anterior cuando los elementos diagonales son todos distintos entre sí. En otras palabras, se puede considerar que Beltrami es el descubridor de lo que hoy en día llamaríamos la descomposición en valores singulares de matrices reales cuadradas no singulares con valores singulares distintos. Omite las posibles situaciones de degeneración que, como veremos, tienen importantes consecuencias.

Esta omisión no es debida a un intento de simplificar las cosas para sus estudiantes, sino que, tal y como se comprueba por las ideas que aporta, posiblemente sea consecuencia de no plantearse el problema en su completa generalidad.

Camille Jordan publicó un par de trabajos sobre formas bilineales, [10, 11], un año más tarde que Beltrami, por lo que se le puede considerar codescubridor de los valores

singulares.

Su aproximación es completamente diferente. De hecho, en el primero de los artículos citados, su preocupación es la de buscar el máximo y mínimo de la forma bilineal.

$$P = x^T A y$$

condicionados a que $\|x\|_2 = \|y\|_2 = 1$. Usando la condición de la anulación de la diferencial en los máximos y mínimos relativos, Jordan prueba que dicho máximo es la raíz cuadrada del mayor valor propio de la matriz

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \quad (1)$$



Camille Jordan
(1831–1921)

Es posiblemente la primera vez que se usa el hoy ampliamente utilizado proceso de *deflacción*, que consiste en reducir un problema sobre matrices al mismo problema con una matriz de orden una unidad inferior. La caracterización de los valores singulares de A como las raíces cuadradas positivas de los valores propios de la matriz en (2) es un hecho muy conocido y popular, aunque habitualmente atribuido a Wielandt (ver [3]) y Lanczos [12].

Desconocedor, aparentemente, de los trabajos de Beltrami y Jordan, J.J. Sylvester escribió una nota a pie de página y dos artículos sobre la diagonalización de formas bilineales mediante sustituciones ortogonales.



John Joseph Sylvester
(1814–1897)

La nota a pie de página apareció en un artículo de la revista *The Messenger of Mathematics* [13]. El artículo lleva por título “A new proof that a general quadric may be reduced to its canonical form (that is, a linear function of squares) by means of a real orthogonal substitution”, y en él Sylvester propone un algoritmo para reducir una forma cuadrática a forma diagonal. En la nota a pie de página señala que una iteración similar es posible para diagonalizar una forma bilineal y dice que lo ha enviado para su publicación en las *Comptes Rendues* [20]. Esto nos da una idea de que no conocía el artículo original de Jordan publicado más de una década antes.

En palabras de Stewart estos artículos de Sylvester “no se leen fácilmente. El estilo es opaco y Sylvester pontifica sin probar, dejando demasiados detalles al lector”. Aparentemente el autor estima sus resultados como importantes, pero, sin poner en duda que lo son, hoy en día se ven más como un redescubrimiento de resultados ya obtenidos por otros. En cualquier caso, Sylvester es el primero en dar nombre a los números reales positivos

que aparecen en la forma canónica diagonal: *multiplicadores canónicos*.

Aunque hay nuevos resultados en el terreno del álgebra lineal ligados a los valores singulares, damos un salto en el tiempo para redescubrirlos en un área de la matemática bien distinta: las ecuaciones integrales, uno de los temas que recibió más atención durante las primeras décadas del siglo XX.

En 1907, **Erhard Schmidt** (autor del famoso procedimiento de Gram-Schmidt) publicó, [21], una teoría general de ecuaciones integrales reales con núcleos simétricos y no simétricos. Introduce los conceptos de valor propio y de función propia de un núcleo $K(x, y)$ continuo en $[a, b] \times [a, b]$ mediante las ecuaciones integrales.

$$u(x) = \lambda \int_a^b K(x, y) dy$$



Erhard Schmidt
(1876–1959)

y

$$v(y) = \lambda \int_a^b K(x, y) u(x) dx.$$

Y muestra, a continuación, que λ debe ser real porque λ^2 es un valor propio del núcleo simétrico definido positivo

$$H(x, y) = \int_a^b K(x, t) K(t, y) dt.$$

Si pensamos en $K(x, y)$ como lo análogo a la matriz A , entonces $H(x, y)$ es análogo a $A^T A$. De hecho, Schmidt transita, en dimensión infinita, por senderos similares a los de Beltrami.

Un poco después de la publicación del artículo de Schmidt, Bateman, [14], introduce la notación de valores singulares para referirse a unos números que son esencialmente los recíprocos de los valores propios definidos por Schmidt. Y Picard, [15], hace notar que para los núcleos simétricos los valores propios de Schmidt son reales y en este caso (pero no en general) los llama valores singulares.



Hermann Weyl (1885–1955)

A pesar de todo, la denominación no se estabiliza, y entre los muchos matemáticos notables que han trabajado en las propiedades de los valores singulares, debemos nombrar a *Hermann Weyl* con importantes contribuciones a la teoría de perturbación de los mismos. Pero en cuanto a nomenclatura se refiere, todavía en su artículo [15] habla de “dos clases de valores propios” y en una traducción al inglés de un tratado ruso sobre operadores no autoadjuntos, Gohberg y Krein, [5], se refieren a los valores singulares como “s-numeros” de un operador.

La fuerte implantación de los ordenadores y el consecuente apogeo del análisis numérico ha contribuido a que finalmente la denominación de valores singulares haya arraigado en la comunidad matemática.

2. La descomposición en valores singulares. Resultados teóricos

En esta sección inicialmente se presenta los valores singulares, seguidamente la definición y los teoremas que muestran cómo la DVS da información útil sobre la estructura de una matriz.

2.1. Definición del DVS

La Descomposición en Valores Singulares (*en inglés Singular Value Decomposition, SVD*) [7] es una técnica de factorización de matrices que permite descomponer una matriz A en otras tres matrices U , Σ , y V^T de la siguiente manera:

$$SVD(A) = U \times \Sigma \times V^t$$

Es decir, que el producto matricial de la matriz U por Σ por V^t da como resultado la matriz inicial A .

Respecto a las dimensiones de las matrices tenemos que la matriz A va a tener unas dimensiones de $(m \times n)$ es decir m filas y n columnas. La matriz U va a tener dimensión $(m \times m)$, la matriz Σ tendrá dimensión $(m \times n)$ y por último la matriz V tendrá dimensión $(n \times n)$. Al ser V una matriz cuadrada las dimensiones de la matriz traspuesta serán las mismas que la matriz original. A continuación mostramos como serían las dimensiones de las matrices para esta descomposición:

$$SVD(A_{m \times n}) = U_{m \times m} \times \Sigma_{m \times n} \times V_{n \times n}^t$$

Recordemos que U y V son matrices ortogonales si:

$$U^t \times U = U \times U^t = I_{n \times n}$$

$$V^t \times V = V \times V^t = I_{m \times m}$$

La matriz Σ ha de ser una matriz diagonal, la cual tendrá en su diagonal lo que se denominan *Valores Singulares*, y estos han de estar puestos en orden decreciente (y tendrán siempre valores mayores o iguales a cero), es decir que la matriz Σ tendrá la siguiente forma:

$$\Sigma = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \lambda_n \end{bmatrix}, \text{ donde } \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \quad (2)$$

En el apéndice recordamos la definición de lo que son los autovalores, autovectores y que es el polinomio característico de una matriz ya que son conceptos que hemos de manejar para la obtención de estas matrices.

2.2. El teorema de descomposición

Los siguientes resultados demuestran que toda matriz tiene una descomposición en valores singulares como la descrita anteriormente.

Teorema 2.1 *Dada una matriz $A \in \mathbb{R}^{m \times n}$ de rango r , existen números reales $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ y bases ortonormales $\{u_1, u_2, \dots, u_m\}$ de \mathbb{R}^m y $\{v_1, v_2, \dots, v_n\}$ de \mathbb{R}^n tales que*

$$\begin{aligned} Av_i &= \sigma_i u_i, & i &= 1, 2, \dots, r, & A^T u_i &= \sigma_i v_i, & i &= 1, \dots, r, \\ Av_i &= 0, & i &= r + 1, \dots, n, & A^T u_i &= 0, & i &= r + 1, \dots, m. \end{aligned}$$

Estas ecuaciones implican que v_1, v_2, \dots, v_n son autovectores de $A^T A$, u_1, u_2, \dots, u_m son autovectores de AA^T y $\sigma_1^2, \dots, \sigma_r^2$ son los autovalores no nulos de $A^T A$ y AA^T .

Demostración 1 *Como $A^T A$ es simétrica, diagonaliza ortogonalmente y sus autovalores son reales. Sea (v_1, \dots, v_n) una base ortonormal de \mathbb{R}^n formada de autovectores de $A^T A$ y sean $\lambda_1, \dots, \lambda_n$ los autovalores asociados. Como $A^T A$ es semidefinida positiva, es decir no negativos, todos sus autovalores son mayores o iguales que 0. Sin pérdida de generalidad consideramos v_1, \dots, v_n ordenados de modo que $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$. Por ser r el rango de A y, por lo tanto, el rango de $A^T A$ debe cumplirse que $\lambda_r > 0$ y $\lambda_{r+1} = \lambda_{r+2} = \dots = \lambda_n = 0$. Por lo tanto v_1, \dots, v_n están en el núcleo de $A^T A$ y, por tanto, en el núcleo de A , es decir, $Av_i = 0$ para $i = r + 1, \dots, n$. Para $1 \leq i \leq r$ definimos σ_i y u_i de la siguiente manera*

$$\sigma_i = \sqrt{\lambda_i}, \quad u_i = \frac{1}{\sigma_i} Av_i. \quad (3)$$

Por la forma en que hemos definido u_i se tiene que $Av_i = \sigma_i u_i$, $i = 1, \dots, r$, y por la definición de σ_i obtenemos que

$$\|u_i\| = \sqrt{u_i^T u_i} = \frac{1}{\sigma_i} \sqrt{(Av_i)^T (Av_i)} = \frac{1}{\sigma_i} \sqrt{(v_i)^T (A^T A) v_i} = \frac{\sigma_i}{\sigma_i} \sqrt{v_i^T v_i} = 1, \quad i = 1, \dots, r.$$

Por otro lado dados dos vectores ortogonales $v_i, v_j, i \neq j$, se tiene que Av_i y Av_j también son ortogonales, puesto que

$$(Av_i)^T (Av_j) = v_i^T A^T Av_j = v_i^T \sigma_j^2 v_j = \sigma_j^2 v_i^T v_j = 0$$

Esto implica que por la forma en que hemos definido $u_i, i = 1, \dots, r$, los vectores u_1, \dots, u_r son ortogonales. Además $\sigma_i^2 = \lambda_i, i = 1, \dots, r$. Ahora multiplicando por A^T en la segunda igualdad de (1) se tiene para $1 \leq i \leq r$.

$$A^T u_i = \frac{1}{\sigma_i} A^T Av_i = \frac{1}{\sigma_i} \lambda_i v_i = \sigma_i v_i. \quad (4)$$

Falta ahora definir u_{r+1}, \dots, u_m en el caso en que $r < m$. Los vectores u_1, \dots, u_r son autovectores de AA^T asociados a los autovalores no nulos $\lambda_1, \dots, \lambda_r$ pues

$$AA^T u_i = A \sigma_i v_i = \sigma_i Av_i = \sigma_i^2 u_i = \lambda_i u_i.$$

Como $AA^T \in \mathbb{R}^{m \times m}$ y tiene rango r , la dimensión del subespacio $\text{Ker}(AA^T)$ es $m - r$. Tomamos u_{r+1}, \dots, u_m vectores ortonormales de una base de $\text{Ker}(AA^T)$. Dado que u_{r+1}, \dots, u_m son autovectores de AA^T asociados al autovalor 0 y que AA^T es simétrica, los vectores u_{r+1}, \dots, u_m son ortogonales a u_1, \dots, u_r . Por tanto, u_1, \dots, u_m es una base ortonormal de \mathbb{R}^m formada por autovectores de AA^T . Además, como el núcleo de AA^T es el mismo que el núcleo de A^T se tiene que $A^T u_i = 0$ para $i = r + 1, \dots, m$.

Teorema 2.2 Dada una matriz $A \in \mathbb{R}^{m \times n}$ de rango r , existen matrices ortogonales $U \in \mathbb{R}^{m \times m}$ y $V \in \mathbb{R}^{n \times n}$, y una matriz Σ de la forma

$$\Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \\ & & & 0 & 0 \end{bmatrix} \in \mathbb{R}^{m \times n}$$

Con $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$, (el símbolo 0 denota matrices nulas de las dimensiones adecuadas), tales que

$$A = U \Sigma V^T \quad (5)$$

Demostración 2 En virtud del Teorema 1, existen bases ortogonales $\{v_1, \dots, v_n\}$ de \mathbb{R}^n y $\{u_1, \dots, u_m\}$ de \mathbb{R}^m tales que

$$Av_i = \begin{cases} \sigma_i u_i, & i = 1, \dots, r, \\ 0 & i = r + 1, \dots, n. \end{cases} \quad (6)$$

Tomando U y V las matrices con columnas igual a los vectores de dichas bases ortonormales se tiene que $U = [u_1, \dots, u_m] \in \mathbb{R}^{m \times m}$ y $V = [v_1, \dots, v_n] \in \mathbb{R}^{n \times n}$ son matrices ortogonales y las ecuaciones (7) pueden ser escritas en forma matricial como

$$[v_1, \dots, v_r | v_{r+1}, \dots, v_n] = [u_1, \dots, u_r | u_{r+1}, \dots, u_m] \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \\ & & & 0 & 0 \end{bmatrix}$$

$AV = U\Sigma$, donde los vectores u_{r+1}, \dots, u_m no intervienen expresamente al ir siempre multiplicados por ceros.

Como V es ortogonal, $VV^T = I$ y, por tanto,

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

Los $\sigma_i, i = 1, 2, \dots, r$, son los llamados valores singulares de A , con vectores singulares asociados por la derecha v_1, \dots, v_r y por la izquierda u_1, \dots, u_r . Teniendo en cuenta la estructura de ceros de la matriz Σ y a la vista de (6), A se puede escribir como una suma de r matrices de rango 1

$$A = \sum_{j=1}^r \sigma_j u_j v_j^T : \quad (7)$$

2.3. Teorema del rango aproximado

Un resultado fundamental para las aplicaciones de la SVD es el siguiente teorema de aproximación, cuya demostración puede verse en [8]. Supongamos que, a partir del cálculo de la DVS, tenemos A (de rango r) expresada como

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_r u_r v_r^T.$$

Entonces, la mejor aproximación de A (en la norma espectral) mediante matrices de su mismo tamaño y rango entonces $k < r$ es la matriz

$$A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_k u_k v_k^T,$$

que se obtiene tomando solamente los k valores singulares más grandes.

La norma espectral de una matriz A viene dada por

$$\|A\|_2 = \sigma_1,$$

es decir es igual al mayor de sus valores singulares. En relación con la mejor aproximación mediante matrices de rango $\leq k < r$ se tiene

$$\|A - A_k\|_2 = \sigma_{k+1}$$

2.4. Ejemplo matemático

En esta sección se va a poner un ejemplo práctico de como obtener las matrices U, S y Σ , es decir, como hacer el SVD a partir de la siguiente matriz A de dimensiones 2×3 :

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 3 & 0 \end{bmatrix} \quad (8)$$

Cálculo de la Matriz U

El primer paso que debemos de dar es calcular la matriz U . Es indiferente empezar calculando la matriz U o la V , pero comenzaremos por la U con el fin de mantener el orden que se ha seguido a la hora de explicar este método. Lo primero que tenemos que hacer es calcular la matriz cuadrada (que llamaremos C_1) de la que obtendremos los autovalores y autovectores. Para ello hacemos el siguiente cálculo

$$C_1 = A \times A^t = A = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 3 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 3 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 13 \end{bmatrix} \quad (9)$$

Una vez que tenemos la matriz cuadrada C_1 , pasamos a calcular los autovalores igualando a cero el polinomio característico de la matriz C_1

$$|C_1| = \begin{vmatrix} 2 - \lambda & 2 \\ 2 & 13 - \lambda \end{vmatrix} = (2 - \lambda)(13 - \lambda) - 4 = \lambda^2 - 15\lambda + 22 \quad (10)$$

El siguiente paso es despejar λ para que nos de los autovalores. En este caso tenemos que resolver una ecuación de segundo grado:

$$\lambda = \frac{15 \pm \sqrt{15^2 - 4 \cdot 1 \cdot 22}}{2 \cdot 1} \rightarrow \lambda_1 = 13,35 \quad y \quad \lambda_2 = 1,65 \quad (11)$$

Con este cálculo ya hemos obtenido los autovalores de la matriz C_1 que son $\lambda_1 = 13,35$ y $\lambda_2 = 1,65$.

El siguiente paso que dar es el de calcular los autovectores asociados a cada uno de los autovalores obtenidos. Esto lo hacemos de la siguiente forma:

$$C_1 v_1 = \lambda_1 v_1; \quad \begin{bmatrix} 2 & 2 \\ 2 & 13 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 13,35 \begin{bmatrix} x \\ y \end{bmatrix} \quad (12)$$

$$C_1 v_2 = \lambda_2 v_2; \quad \begin{bmatrix} 2 & 2 \\ 2 & 13 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 1,65 \begin{bmatrix} x \\ y \end{bmatrix} \quad (13)$$

El sistema de ecuaciones resultante de la ecuación (13) es el siguiente:

$$\begin{aligned} 2x + 2y &= 13,35x \\ 2 + 13y &= 13,35y \end{aligned} \quad (14)$$

Lo resolvemos de la siguiente manera:

$$\begin{aligned} -11,35 + 2y &= 0; \\ \text{si } x &= 1 \\ y &= \frac{11,35}{2} = 5,675 \end{aligned} \quad (15)$$

Por tanto tenemos que $x = 1$ e $y = 5,675$ pero estos valores no forman todavía el autovector ya que el autovector tiene que tener norma '1', por lo tanto hay que normalizarlo:

$$|v_1| = \sqrt{1^2 + 5,675^2} = 5,848 \quad (16)$$

Por lo tanto, el primer autovector va a ser el siguiente:

$$|v_1| = \left[\frac{1}{5,848}, \frac{5,675}{5,848} \right] = [0,17, 0,98] \quad (17)$$

Para obtener el segundo autovector tengo que realizar los mismo pasos pero en este caso tengo que resolver el siguiente sistema de ecuaciones:

$$\begin{aligned} 2x + 2y &= 1,65x \\ 2 + 13y &= 1,65y \end{aligned} \quad (18)$$

Resolviendo este sistema de ecuaciones obtengo el segundo autovector:

$$v_2 = [0,98 \quad -0,17] \quad (19)$$

Una vez que tengo estos todos los autovectores ya puedo construir la matriz U que estará compuesta por los dos autovectores de la siguiente forma:

$$U = \begin{bmatrix} 0,17 & 0,98 \\ 0,98 & -0,17 \end{bmatrix} \quad (20)$$

Para comprobar que hemos calculado bien esta matriz podemos comprobar que cumple las propiedades descritas anteriormente, que decían se si un autovector lo multiplicamos por si mismo (producto escalar) da como resultado 1 y si multiplicamos un autovector con otro da como resultado 0 :

$$v_1 v_1^T = [0,17 \quad 0,98] \begin{bmatrix} 0,17 \\ 0,98 \end{bmatrix} = 1 \quad (21)$$

$$v_1 v_2^T = [0, 17 \quad 0, 98] \begin{bmatrix} 0, 98 \\ -0, 17 \end{bmatrix} = 0 \quad (22)$$

Cálculo de la Matriz V

Para el cálculo de la matriz en este caso obtendremos los autovalores y autovectores de la matriz cuadrada resultante de multiplicar $A^t \times A$.

$$C_2 = A^t \times A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 2 & 3 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 6 & 1 \\ 6 & 9 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad (23)$$

El siguiente paso es el de calcular los autovalores igualando a cero en polinomio característico de la matriz C_2 :

$$|C_2| = \begin{vmatrix} 5 - \lambda & 6 & 1 \\ 6 & 9 - \lambda & 0 \\ 1 & 0 & 1 - \lambda \end{vmatrix} = -\lambda^3 + 15\lambda^2 - 22\lambda \quad (24)$$

Para obtener los autovalores tenemos que despejar λ en esta ecuación de tercer grado. Como ya se comentó los autovalores deben ser los mismos que los obtenidos en el cálculo de la matriz U y la diferencia de autovalores tendrán valor 0, así que es fácil comprobar que para los valores de $\lambda_1 = 13, 35$, $\lambda_2 = 1, 65$ y $\lambda_3 = 0$ se cumple la siguiente igualdad y por lo tanto son estos los autovalores de C_2 :

$$-\lambda^3 + 15\lambda^2 - 22\lambda = 0 \quad (25)$$

Para el cálculo de los autovectores, tendríamos que resolver los siguientes sistemas de ecuaciones (27), (28) y (29)

$$C_2 v_1 = \lambda_1 v_1; \quad \begin{bmatrix} 5 & 6 & 1 \\ 6 & 9 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 13, 35 \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (26)$$

$$C_2 v_3 = \lambda_3 v_3; \quad \begin{bmatrix} 5 & 6 & 1 \\ 6 & 9 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 0 \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (27)$$

Resolviendo estos sistemas de ecuaciones obtenemos los siguientes autovectores:

$$v_1 = [0, 58 \quad 0, 49 \quad -0, 64] \quad (28)$$

$$v_2 = [0, 81 \quad -0, 41 \quad -0, 43] \quad (29)$$

$$v_3 = [0,05 \quad 0,77 \quad 0,64] \quad (30)$$

Con el cálculo de estos autovectores ya podemos obtener la matriz V que mostramos a continuación:

$$V = \begin{bmatrix} 0,58 & 0,49 & -0,64 \\ 0,81 & -0,41 & -0,43 \\ 0,05 & 0,77 & 0,64 \end{bmatrix} \quad (31)$$

Al igual que en el cálculo de la matriz U , podemos comprobar que se ha calculado correctamente haciendo el producto escalar de cada una de las filas consigo misma y el producto escalar de una fila con otra y nos deberá dar como resultados 1 y 0 respectivamente.

Nota:

Otra forma de calcular la matriz V será la siguiente. Para $i = 1, 2, \dots, r$ tomamos

$$v_i = \frac{1}{\sqrt{\sigma_i}} Au_i$$

Para $i = r + 1, \dots, m$ tomamos una b.o.n de $\mathcal{L}(v_1, \dots, v_r)^\perp$.

Cálculo de la Matriz Σ

El último paso es calcular la matriz Σ y este es el cálculo más sencillo ya que prácticamente lo tenemos hecho. Lo primero que hay que ver es la dimensión que tiene la matriz original A que es de (2×3) para construir la matriz Σ con la misma dimensión, por tanto la matriz σ también tendrá dimensión (2×3) .

El siguiente paso es poner en su diagonal, la raíz cuadrada de los autovalores calculados en el cálculo de la matriz U y V que serán los mismos salvo los autovalores que han dado valor cero, es decir que la matriz Σ tendrá la siguiente forma:

$$\Sigma = \begin{bmatrix} \sqrt{\lambda_1} & 0 & 0 \\ 0 & \sqrt{\lambda_2} & 0 \end{bmatrix} = \begin{bmatrix} \sqrt{13,35} & 0 & 0 \\ 0 & \sqrt{1,65} & 0 \end{bmatrix} \quad (32)$$

Por tanto concluimos que la matriz S tendrá el siguiente valor:

$$\Sigma = \begin{bmatrix} 3,65 & 0 & 0 \\ 0 & 1,28 & 0 \end{bmatrix} \quad (33)$$

Una vez que se tiene hecho el cálculo de las tres matrices U , S y Σ es muy fácil comprobar que el producto matricial de estas tres matrices (haciendo el producto matricial con la matriz V traspuesta) es igual a la matriz A original:

$$\begin{bmatrix} 1 & 0 & 1 \\ 2 & 3 & 0 \end{bmatrix} = \begin{bmatrix} 0,17 & 0,98 \\ 0,98 & -0,17 \end{bmatrix} \begin{bmatrix} 3,65 & 0 & 0 \\ 0 & 1,28 & 0 \end{bmatrix} \begin{bmatrix} 0,58 & 0,81 & 0,05 \\ 0,49 & -0,41 & 0,77 \\ -0,64 & 0,43 & 0,64 \end{bmatrix} \quad (34)$$

3. Aplicaciones de Descomposición en valores singulares

3.1. Aplicaciones matemáticas de la DVS

3.1.1. Problema lineal de mínimos cuadrados

El problema lineal de mínimos cuadrados consiste en encontrar la solución de un sistema lineal $Ax = b$ con $x \in \mathbb{R}^n$ el vector de incógnitas. Si $m > n$, el sistema es sobredeterminado y podrá no tener solución. En ese caso buscaremos un x tal que $\|b - Ax\|_2$ sea mínima. Hallar dicho x es lo que conocemos como resolver el problema lineal de mínimos cuadrados. En el caso en el que $m \geq n$ y $\text{rango}(A) = n$ el problema lineal de mínimos cuadrados tiene solución única. Si $\text{rango}(A) < n$, la solución en el sentido de mínimos cuadrados no es única y existen varios $x \in \mathbb{R}^n$ para los cuales $\|b - Ax\|_2$ es mínima. Incluso en el caso $m < n$ puede ocurrir que $Ax = b$ no tenga solución.

Como la solución del problema lineal de mínimos cuadrados puede no ser única, consideramos el siguiente problema que sí va a tener solución única: *de todos los $x \in \mathbb{R}^n$ que minimizan $\|b - Ax\|_2$, encontrar aquel para el cual $\|x\|_2$ sea lo más pequeña posible.*

Supongamos A y b conocidos, y que tenemos la descomposición en valores singulares de A , $A = U\Sigma V^T$, donde $U \in \mathbb{R}^{m \times m}$ y $V \in \mathbb{R}^{n \times n}$ son ortogonales y Σ es una matriz diagonal. Como U es ortogonal,

$$\|b - Ax\|_2 = \|U^T(b - Ax)\|_2 = \|U^Tb - U^T Ax\|_2 = \|U^Tb - \Sigma(V^T x)\|_2.$$

Haciendo $c = U^Tb$ e $y = V^T x$, se tiene

$$\|b - Ax\|_2^2 = \|c - \Sigma y\|_2^2 = \sum_{i=1}^r |c_i - \sigma_i y_i|^2 + \sum_{i=r+1}^m |c_i|^2. \quad (35)$$

El mínimo se alcanza si, y solo si, $y_i = \frac{c_i}{\sigma_i}$, $i = 1, \dots, r$, y sale $\sum_{i=r+1}^m |c_i|^2$.

Cuando $r < n$, y_{r+1}, \dots, y_n no aparecen explícitamente y de todas las soluciones así obtenidas tiene $\|y\|_2$ mínima la que satisface $y_{r+1} = \dots = y_n = 0$. Como $x = Vy$ y V es ortogonal, $\|x\|_2 = \|y\|_2$. Entonces $\|x\|_2$ es mínima si, y solo si, $\|y\|_2$ lo es. Esto prueba que el problema lineal de mínimos cuadrados tiene exactamente una solución de norma mínima.

3.1.2. Pseudoinversa

Ahora la definición de pseudoinversa de una matriz que adoptamos en este trabajo, que está íntimamente relacionada con la resolución del problema lineal de mínimos cuadrados. La Pseudoinversa A^+ de una matriz $A \in \mathbb{R}^{m \times n}$ es una generalización de la matriz inversa [1]. El tipo de matriz pseudoinversa más conocida es la llamada *Pseudoinversa de Moore-Penrose*, que fue descrita independientemente por E. H. Moore [13] en 1920. Podemos dar una expresión para A^+ en algunos casos particulares.

- Si $m = n$ y A es regular, $x = A^{-1}b$ es la única solución del sistema lineal $Ax = b$ y, por tanto, en este caso $A^+ = A^{-1}$ (la pseudoinversa coincide con la inversa).
- Si $m > n$ y el rango de A es n , ya hemos comentado que existe una única solución en el sentido de mínimos cuadrados del sistema lineal $Ax = b$, que no es otra que la solución de las correspondientes ecuaciones normales, que al ser $A^T A$ regular viene dada por $x = (A^T A)^{-1} A^T b$. En este caso, entonces, $A^+ = (A^T A)^{-1} A^T$.

En la descripción que hemos hecho del problema lineal de mínimos cuadrados hemos visto que si $c = U^T b = \begin{pmatrix} \hat{c} \\ d \end{pmatrix}$ e $y = \begin{pmatrix} \hat{y} \\ 0 \end{pmatrix}$ con \hat{c} y $\hat{y} \in \mathbb{R}^r$, la solución de norma mínima viene dada por $x = Vy$ donde además

$$x = Vy = V \begin{pmatrix} \hat{y} \\ 0 \end{pmatrix} = V \begin{pmatrix} \Sigma^{-1} \hat{c} \\ 0 \end{pmatrix} = V \begin{pmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{c} \\ d \end{pmatrix} = V \begin{pmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{pmatrix} c = V \begin{pmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T b,$$

de donde, con la definición que hemos dado de pseudoinversa, se deduce que

$$A^+ = V \begin{pmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T$$

De la igualdad anterior se deduce también que el rango de A^+ y el de A coinciden, que u_1, \dots, u_r y v_1, \dots, v_r son vectores singulares de A^+ por la derecha y por la izquierda, respectivamente y que $\sigma_1^{-1}, \dots, \sigma_r^{-1}$ valores singulares de A^+ . Además, la matriz $\begin{pmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{pmatrix}$ no es otra que Σ^+ , la pseudoinversa de Σ . Para ver esto, notemos que si nos planteamos el sistema lineal $\Sigma x = b$ o, equivalentemente,

$$\begin{aligned} \sigma_i x_i &= b_i, \quad i = 1, \dots, r. \\ 0 &= b_i, \quad i = r + 1, \dots, m. \end{aligned}$$

su solución de norma mínima en el sentido de mínimos cuadrados viene dada por $x_i = \frac{b_i}{\sigma_i}$ para $i = 1, \dots, r$, y $x_i = 0$, para $i = r + 1, \dots, n$, que puede expresarse como

$$x = \begin{pmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{pmatrix} b \Rightarrow \Sigma^+ = \begin{pmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{n \times m}$$

Podemos escribir entonces

$$A^+ = V \Sigma^+ U^T \tag{36}$$

o incluso $A^+ = \sum_{j=1}^r \frac{1}{\sigma_j} v_j u_j^T$ que es la descomposición en valores singulares de A^+ , análoga a (4) para A .

Veamos ahora que la pseudoinversa, tal y como se ha definido en esta subsección coincide con la inversa generalizada de Moore-Penrose.

Más precisamente, si $A \in \mathbb{R}^{m \times n}$ es una matriz dada y $A^+ \in \mathbb{R}^{n \times m}$ denota su pseudoinversa se trata de ver que se satisfacen las siguientes condiciones:

1. $AA^+A = A$.
2. $A^+AA^+ = A^+$.
3. $(AA^+)^T = AA^+$.
4. $(A^+A)^T = A^+A$.

Demostración 3 *En todos los casos vamos a utilizar la descomposición en valores singulares de A y A^+ obtenidas en (3) y (7) respectivamente.*

(1) Si

$$AA^+A = (U\Sigma V^T)(V\Sigma^+U^T)(U\Sigma V^T) = U\Sigma\Sigma^+\Sigma V^T$$

Teniendo en cuenta que $\Sigma\Sigma^+$ es una matriz diagonal $m \times m$ con los elementos $(i, i) = 1$, $i = 1, \dots, r$ e $(i, i) = 0$, $i = r + 1, \dots, m$, es fácil concluir que $\Sigma\Sigma^+ = \Sigma$ y, por tanto, que

$$AA^+A = U\Sigma\Sigma^+\Sigma V^T = U\Sigma V^T = A.$$

(2) de forma análoga a lo anterior, tenemos

$$A^+AA^+ = (V\Sigma^+U^T)(U\Sigma V^T)(V\Sigma^+U^T) = V\Sigma^+\Sigma\Sigma^+U^T$$

Teniendo en cuenta ahora que $\Sigma^+\Sigma$ es una matriz diagonal $n \times n$ con los elementos $(i, i) = 1$, $i = 1, \dots, r$ e $(i, i) = 0$, $i = r + 1, \dots, n$, es fácil ver que $\Sigma^+\Sigma\Sigma^+ = \Sigma^+$ y, por tanto, concluir que

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

(3) se tiene que

$$AA^+ = (U\Sigma V^T)(V\Sigma^+U^T) = U\Sigma\Sigma^+U^T.$$

Trasponiendo obtenemos

$$(AA^+)^T = (U\Sigma\Sigma^+U^T)^T = U(\Sigma\Sigma^+)^T U^T = U\Sigma\Sigma^+U^T,$$

donde la última igualdad es cierta por ser $\Sigma\Sigma^+$ matriz diagonal.

(4) razonamos de forma análoga al caso anterior,

$$A^+A = (V\Sigma^+U^T)(U\Sigma V^T) = V\Sigma^+\Sigma V^T.$$

Calculando ahora la traspuesta de A^+A se tiene que

$$(A^+A)^T = (V\Sigma^+\Sigma V^T)^T = V(\Sigma^+\Sigma)^T V^T = V\Sigma^+\Sigma V^T,$$

siendo la última igualdad cierta por ser $\Sigma^+\Sigma$ matriz diagonal.

3.2. Algunas aplicaciones de la descomposición en valores singulares en tecnología

3.2.1. Compresión de imágenes

Una importante aplicación de la DVS es la compresión de imágenes digitales, que es precisamente la primera de las aplicaciones que destaca G. Strang en [19]. El fundamento teórico de esta aplicación es el resultado de aproximación en la norma de Frobenius, que consiste en aproximar cada coeficiente de una matriz por el correspondiente coeficiente de la matriz que se obtiene al considerar su aproximación óptima por matrices de un rango determinado.

Dada una matriz A de datos de tamaño $m \times n$, la transmisión de dicha matriz completa supone la transmisión de, mn valores reales. Si consideramos la DVS de $A = \sum_{i=1}^r \sigma_i u_i v_i^T$, el teorema de aproximación en la norma de Frobenius permite de igual forma escribir la matriz de rango k que mejor aproxima a A , como $A_k = \sum_{i=1}^k \sigma_i u_i v_i^T$. Incrementar en una unidad de rango de la matriz con la que aproximamos a la matriz original supone la transmisión de $(m+n)$ números adicionales, puesto que se puede multiplicar previamente cada vector singular por la derecha por el correspondiente valor singular. Si en vez de transmitir A , transmitimos la información necesaria para generar la aproximación A_k , es suficiente transmitir $k(m+n)$ datos, en lugar de los mn que forman la matriz completa.

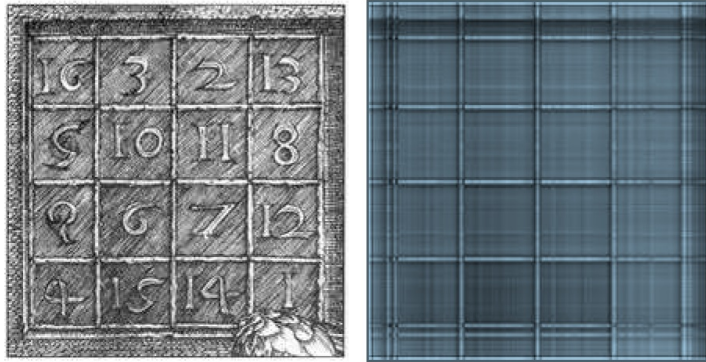


Figura 1: La figura de la izquierda muestra la imagen original con un total de 50380 datos. La figura de la derecha utiliza 898 datos, el 2% del total.

El procedimiento tendrá interés sólo si se puede tomar $k \ll \frac{mn}{m+n}$. En caso contrario, además de tener que transmitir más elementos que el número de píxeles de la imagen original habríamos tenido que calcular la DVS de la que, como hemos visto, requiere el orden de m^2n operaciones, al ser necesarios también los valores singulares. El teorema de aproximación permite disponer de una primera imagen no exacta sin necesidad de disponer de los valores exactos de todos los píxeles.

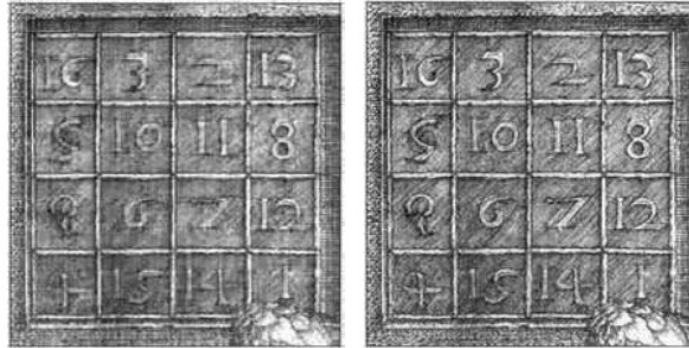
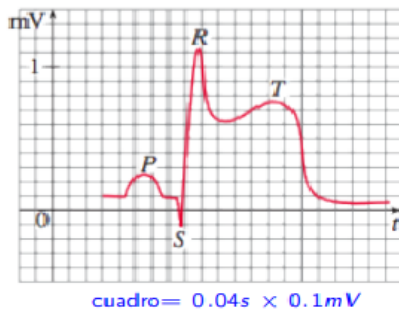


Figura 2: La figura de la izquierda muestra la imagen con 12572 datos que representa el 25 % del total. La figura de la derecha utiliza el 50 % del total de los datos, 25144.

3.2.2. Compresión y filtrado de electrocardiograma



Un electrocardiograma es la gráfica de la función potencial eléctrico $V(t)$ (medido en minivoltios) en función del tiempo (medido en segundos) en una determinada dirección correspondiente a una parte del corazón. Los cardiólogos utilizan los extremos relativos (P,R,S,T) para determinar posibles problemas del corazón.

La descomposición en valores singulares (DVS) es una transformación matemática basada en la correlación entre señales, en este caso las obtenidas del ECG estándar. Por lo general, se aplica DVS a las ocho derivaciones mutuamente independientes ($I, II, V_1 \dots V_6$) del ECG y luego se reconstruye dicha información en un espacio ortogonal óptimo de ocho pseudoderivaciones ($S_1 \dots S_8$) como se muestra en la Figura . En ese espacio, S_1 contendrá la máxima energía o autovalor (λ_1) en esa dirección, S_2 contendrá la máxima energía (λ_2) perpendicular a S_1 , S_3 contendrá el máximo λ_3 perpendicular a las dos primeras pseudoderivaciones y así sucesivamente. $S_1S_2S_3$ contienen alrededor del 98 % de la energía total del ECG, denominada componente dipolar ($\lambda_1\lambda_2\lambda_3$), y $S_4 \dots S_8$ el 2 % restante, denominada

componente no dipolar ($\lambda_4 \dots \lambda_8$).

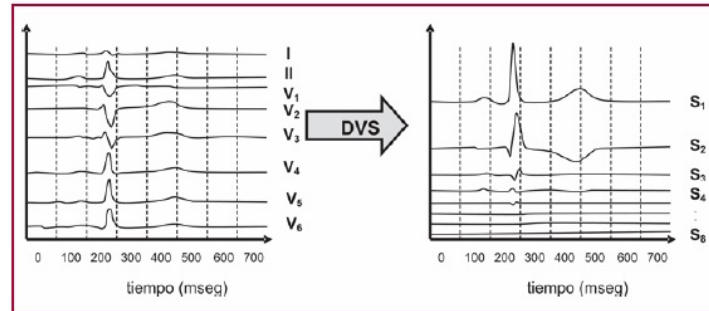


Figura 3: *ECG estándar (panel izquierdo) y las ocho pseudoderivaciones obtenidas por aplicación DVS (panel derecho). Obsérvese la máxima cantidad de energía concentrada en las tres primeras direcciones $S_1S_2S_3$ luego de aplicar DVS.*

3.2.3. Reconocimiento de caras (eigenfaces)

El reconocimiento de caras es un método biométrico para la identificación de individuos mediante características faciales. Existen diversas aplicaciones prácticas, abarcando numerosas disciplinas, donde puede utilizarse el reconocimiento de caras. Dichas aplicaciones pueden categorizarse en aplicaciones de seguridad o comerciales. Dentro de las aplicaciones relacionadas a la seguridad se encuentra el reconocimiento sobre retratos policiales, vigilancia de video, etc.

Para el reconocimiento de caras se utiliza la DVS y “umbralizando” los autovalores así obtenidos a algún valor mayor que el del umbral. Utiliza un análisis proyectivo con el conjunto de entrenamiento de las imágenes sirviendo como el espacio de proyección.

Si $A \in R^{m \times n}$ representando la imagen, y $A_j^{(i)}$ representa la j -ésima cara de la persona i , entonces la imagen promedio para la persona i se encuentra dada por:

$$\frac{1}{N} \sum_{j=1}^N A_j^{(i)}$$

Los autovalores y autovectores se determinan por medio de esta imagen promedio utilizando DVS. Una imagen prueba se proyecta sobre el espacio determinado por los autovectores, la norma de Frobenius se utiliza como un criterio para determinar que persona pertenece a la imagen prueba.

Su uso hoy en día, es muy amplio de lo que creemos, por ejemplo, en las redes sociales

como el *Instagram* utilizan el reconocimiento facial con sus famosos filtros. *Facebook*, permite a las personas etiquetar a otros usuarios en las fotos que subimos por el análisis de sus caras. Vivimos en un ambiente tecnológico en el que nuestros móviles saben mucho de nosotros.

3.2.4. Cocktail party problem

A continuación la aplicación más efectiva para separar las señales, conocido como *the Cocktail Party Problem*, es un ejemplo clásico de Separación de Fuentes Ciegas (BSS), la separación de un conjunto de observaciones en el subyacente señales de *fuentes*. En la figura 2 muestra que cada uno de los J voces se puede escuchar en una fiesta son grabados por N micrófonos, las grabaciones serán una matriz compuesta por un conjunto de N vectores, cada uno de los cuales es una superposición lineal de los J voces. Para un conjunto discreto de M muestras, podemos denotar las fuentes por una matriz $J \times M$, Z y las N grabaciones por una matriz $N \times M$ por matriz X . Por lo tanto, Z se transforma, en observables X , (a través de la propagación de ondas sonoras a través de la habitación). La figura 2 ilustra este paradigma en el que el $J = 3$ independiente hablando (z_1, Z_2 y z_3 izquierda) son superpuesto (centro), y se registran como tres vectores de fuente mixta con ligeras diferencias fases a tres micrófonos separados espacialmente pero por lo demás idénticos.

Para que podamos escoger una voz de un conjunto de voces en una sala llena de gente, debemos debe realizar algún tipo de BSS para recuperar las fuentes originales de la mezcla observada. Matemáticamente, queremos encontrar una matriz de multiplicación M , que al multiplicarla por la variable X de las grabaciones, produzca una estimación Y de las fuentes Z . Por lo tanto, W es un conjunto de pesos (aproximadamente) igual para A^{-1} . Se conoce uno de los métodos clave para realizar el BSS como Análisis Independiente de Componentes (ICA), donde tomamos ventaja de (un supuesto) independencia lineal entre las fuentes

Un excelente ejemplo interactivo The Cocktail Party Problem es:

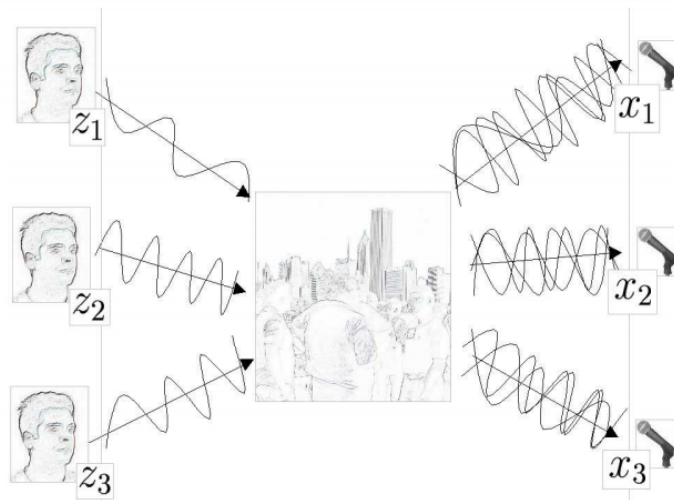


Figura 4: the Cocktail Party Problem

Las ondas sonoras si $J = 3$ con altavoces independiente (z_1, z_2 y z_3 por la izquierda) estan superpuesto en un cocktail party (centro), y se registran como tres vectores fuente mixtos, (x_1, x_2 y x_3 si $N = 3$ por la derecha). Las grabaciones $J \times M$, de fuentes subadyacentes en X, Z es una mezcla lineal de fuentes, por lo tanto $X = AZ$, donde A es una matriz de mezcla lineal. Un estimado Y , de los $J \times M$ fuentes Z , se realiza calculando una matriz de desmezcla W , que actúa sobre X tal como que $Y = WX = \hat{Z}$ y $W \approx A^{-1}$.

3.2.5. Sistema de recomendación

Un sistema de recomendación es un sistema inteligente que proporciona a los usuarios una serie de sugerencias personalizadas (recomendaciones) sobre un determinado tipo de elementos (items). Los sistemas de recomendación estudian las características de cada *usuario* y mediante un procesamiento de los datos, encuentra un subconjunto de *items* que pueden resultar de interés para el usuario.

Si consideramos la matriz de votos como la matriz A (*usuarios* \times *items*) sobre la que realizamos el DVS, obtenemos las tres matrices U, σ y V con sus dimensiones correspondientes. Si reducimos las dimensiones de la matriz σ a k dimensiones, tendremos que la matriz U tendrá dimensiones de (*usuarios* \times k) y la matriz V tendrá dimensiones de (*items* \times k). En este punto se tiene que la matriz U va a tener k -factores que representarán a cada uno de los usuarios y la matriz V va a tener los mismos k -factores que representaran a cada uno de los items.

Supongamos un Sistema de Recomendación de películas en la que tenemos de 5 películas (*items*) y 4 *usuarios*, tal y como se muestra en la siguiente figura 3. Tenemos la matriz de votos de los 4 usuarios sobre las películas de Terminator, Resacon en las Vegas, Ti-

tanic, Toy Story y El Padrino, es decir que tenemos películas de géneros muy diferentes (acción, comedia, romance, etc.). Con la matriz de votos, realizamos el DVS y obtenemos los siguientes tres factores para los *usuarios e items* que se muestran en la siguiente figura:



Figura 5: Ejemplo de factores de los *usuarios e items* obtenidos con el método DVS .

Si observamos bien las votaciones que han realizado los *usuarios* y los valores de los factores de los *usuarios* y de los *items*, podemos sacar en conclusión que el factor $F1$ esta asociado al género cinematográfico de la *Acción*, el factor $F2$ esta asociado a la *Comedia* y el factor $F3$ esta asociado al género *Romántico*. Si vemos los valores de estos factores podemos ver el gusto de cada usuario sobre ese genero y la pertenencia de cada película a ese género.

3.2.6. Análisis componentes principales

Si A representa una matriz de datos de tamaño $m \times n$ con m observaciones de n variables, la matriz $A^T A$ es conocida como *matriz de autocorrelación* y expresa las relaciones estadísticas existentes entre las n variables. Básicamente, la técnica PCA consiste en diagonalizar la matriz de correlación $A^T A$ o la matriz de covarianza $(A - \bar{A})^T (A - \bar{A})$ de los datos A .

La relación entre DVS y PCA se pone de manifiesto al escribir la descomposición DVS de la matriz A en la forma $A = RV^T$, donde $R = US$ es conocida como *matriz de resultados* y a la matriz de vectores singulares por la derecha V^T se le conoce como *matriz de cargas*.

Ejemplo, supongamos que disponemos de un conjunto de datos con información sobre los jugadores de fútbol de primera división. Teniendo en cuenta que la liga española dispone de 20 equipos y que cada uno consta de 25 jugadores con licencia federativa, dispondremos de un total de $m = 500$ observaciones. Para cada jugador disponemos de

tres variables con la media de pases, goles y balones robados por partido respectivamente ($n = 3$ variables, la matriz de datos tendrá un tamaño $m = 500 \times 3$). Los datos, representados en el sistema de ejes tridimensional de las variables, adoptan un aspecto similar al de una nube de puntos en el espacio. La aplicación permite determinar los ejes principales de la distribución de la nube de puntos.

4. Aplicación de la DVS al análisis semántico latente

4.1. Análisis semántico Latente (ASL)

El ASL fue patentado en 1988 por Scott Deerwester, Susan Dumais, George Furnas, Richard Harshman, Thomas Landauer, Karen Lochbaum y Lynn Streeter.

Análisis semántico latente (ASL) llamado también Indexación Semántica Latente (ISL) es la aplicación de una técnica matemática particular, llamada Descomposición en Valores Singulares (DVS por sus siglas en inglés), a una matriz de término-documento basado en la semántica de los documentos, que intenta superar los dos principales problemas que tienen los esquemas de indexación tradicionales de coincidencias léxicas: *polisemia* y *sinonimia*. El primero tiene que ver con que una palabra puede tener múltiples significados y por tanto las palabras de una consulta pueden no coincidir en significado con las de los documentos; el segundo, significa que varios términos pueden significar lo mismo y de allí que las palabras usadas en consultas pueden coincidir con documentos no relevantes.

La característica fundamental del ASL es su habilidad para extraer el contenido conceptual de un documento, estableciendo asociaciones entre aquellos términos que ocurran en contextos similares.

4.2. Matriz término-documento

Para el análisis ASL primero se construye una matriz A donde las filas representan los términos y las columnas los documentos, esta matriz establece las relaciones término documento por lo que cada elemento x_{ij} representa el peso del término i en el documento j , por lo tanto, cada elemento A_{ij} , $1 \leq i \leq t$, $1 \leq j \leq d$.

4.3. Ejemplo de aplicación de DVS al ASL

En esta última sección presentamos los códigos en MatLab para hacer un estudio de la relación entre un listado de títulos de tesis mediante unas palabras claves utilizando la DVS.

Para mejorar la compresión y visualización para nuestro ejemplo trabajaremos con una base de datos formada por una muestra de 20 títulos y de 70 palabras. Para ello generamos el script BaseReducida:

```
%% Titulos de las Tesis
titulos= {'matrices combinadas de algunos tipos de matrices'
'diseño, implementacion y convergencia de metodos iterativos para resolver
ecuaciones y sistemas no lineales utilizando funciones peso'
'desarrollo y analisis de algoritmos probabilisticos para la
reconstruccion de modelos metabolicos a escala genomica'
'modelo socio-demografico dinamico para el estudio de la
sostenibilidad demografica desde factores de calidad de vida'
'algunos resultados sobre b-matrices y matrices con inversa positiva'
'estudio de la
clase de matrices {k,s+1}-potentes'
'metodos iterativos eficientes para la resolucion de sistemas no lineales'
'estructura de grupos finitos y propiedades aritmeticas de los tamaños
de clase de conjugacion'
'metricas fuzzy. aplicaciones al filtrado de imagenes en color'
'svd para la transmision progresiva de imagenes y la codificacion de video
digital'
'aplicaciones de la transformada wavelet al reconocimiento de patrones en la
señal electrocardiografica'
'aplicacion de la transformada wavelet al filtrado de señales
electrocardiograficas'
'analisis estrategico de los estudios tic en la universidad politecnica
de valencia'
'problemas de completacion de matrices parciales'
'modelizacion matematica de la transmision de calor en el proceso
del rectificadado industrial plano'
'indice de alcanzabilidad local de sistemas 2d positivos'
'modelos matematicos en transporte de solutos'
'modelizacion y simulacion de dispositivos micrometricos basados
en estructuras espaciales de solitones opticos'
'algunos problemas de tipo isoperimetrico'
'soluciones numericas de ecuaciones en derivadas parciales
generadas a partir de esquemas semi-implicitos y el metodo de
autofunciones discreto'
};
%% palabras que contienen los titulos de las tesis
palabras={'convergencia'; 'implementacion'; 'combinadas';
'desarrollo'; 'analisis'; 'algoritmos'; 'reconstruccion';
```

```

'modelos'; 'escala'; 'modelo'; 'dinamico'; 'estudio'; 'sostenibilidad';
'demografica'; 'factores'; 'calidad'; 'vida'; 'resultados'; 'matrices';
'inversa'; 'positiva'; 'clase'; 'potentes'; 'metodos';
'iterativos'; 'eficientes'; 'resolucion'; 'sistemas'; 'lineales';
'estructura'; 'grupos'; 'finitos'; 'propiedades'; 'aritmeticas';
'tamaños'; 'conjugacion'; 'metricas'; 'aplicaciones'; 'filtrado';
'color'; 'transmision'; 'progresiva'; 'imagenes';
'codificacion'; 'digital'; 'reconocimiento'; 'patrones';
'señal'; 'electrocardiografica'; 'aplicacion'; 'transformada';
'señales'; 'estrategico'; 'estudios'; 'universidad'; 'problemas';
'completacion'; 'parciales'; 'modelizacion'; 'matematica';
'calor'; 'proceso'; 'rectificado'; 'industrial'; 'plano'; 'indice';
'alcanzabilidad'; 'local'; 'positivos';
'matematicos'};

```

Matriz de incidencia

A continuación construimos la *matriz de incidencia*. Dicha matriz (que llamaremos *ejem*) indica la cantidad de veces que aparece la palabra i en el título j .

```

TitulosTotales=length(titulos); % Cantidad de titulos de la BD.
PalabrasTotales=length(palabras);% Cantidad de palabras de la BD.
ejem=zeros(PalabrasTotales,TitulosTotales);
for i=1:TitulosTotales
    for idx=1:PalabrasTotales
        T=char(titulos(i));
        P=char(palabras(idx));
        ejem(idx,i)=length(strfind(T,P));
    end
end
end

```

Si multiplicáramos la matriz de incidencia por un vector columna con todas sus componentes nulas excepto la componente k (cuyo valor será un 1) obtenemos un vector columna con la cantidad de veces que aparece cada palabra de la base de datos en el título k . Esto lo hacemos mediante el *script*:

```

posTit=1; % posicion del titulo seleccionado
v=zeros(TitulosTotales,1);
v(posTit)=1;
T=titulos(posTit); % ttulo seleccionado de la BD
N=ejem*v;
idx=1;
for i=1:PalabrasTotales

```

```

    if N(i)~= 0
        paltit(idx)=i;
        idx=idx+1;
    end
end
end
P=palabras(paltit); % palabras de título seleccionado

```

Por ejemplo, si el título seleccionado de la BD es:

- matrices combinadas de algunos tipos de matrices
- las palabras que contienen dicho título son:
- combinadas
 - matrices

Análogamente, en el caso de multiplicar la matriz de incidencia por un vector fila de 70 componentes nulas excepto una componente p obtenemos un vector fila de 1x20 que nos indica la cantidad de veces que aparece la palabra p en cada uno de los títulos. El *script* en este caso sería:

```

posPal=19; % posicion del titulo seleccionado
v=zeros(1, PalabrasTotales);
v(posPal)=1;
P=palabras(posPal); % palabra seleccionada de la BD
N=v*ejem;
idx=1;
for i=1:TitulosTotales
    if N(i)~= 0
        titpal(idx)=i;
        idx=idx+1;
    end
end
end
T=titulos(titpal); % títulos que contienen la palabra

```

En este caso, si la palabra seleccionada de la base de datos es:

- matrices
- los títulos que contienen la palabra (matrices) serían:
- matrices combinadas de algunos tipos de matrices
 - algunos resultados sobre b-matrices y matrices con inversa positiva
 - estudio de la clase de matrices $k,s+1$ -potentes
 - problemas de completacion de matrices parciales

Descomposición en valores singulares

Hacemos ahora la DVS de la matriz `ejem` usando la instrucción:

```
[ejemU,ejemS,ejemV]=svd(ejem)
```

Teorema del rango aproximado

Para obtener una primera representación gráfica en dos dimensiones aplicamos en Teorema del rango aproximado con $k = 2$. Para ello usamos primero el *script*:

```
for k=1:2
    ejemS2(k,:)=ejemS(k,:);
    ejemU2(:,k)=ejemU(:,k);
    ejemV2(:,k)=ejemV(:,k);
end
ejemS2=diag(diag(ejemS2));
ejem2=ejemU2*ejemS2*ejemV2';
```

Esto nos permite hacer algunas visualizaciones:

(1) En primer lugar representamos los vectores que contienen las matrices `ejemU2` y `ejemV2` en la misma gráfica usando el *script*:

```
figure('name','Representación Palabras - Títulos');
plot(ejemU2(:,1),ejemU2(:,2),'ro');
hold on
plot(ejemV2(:,1),ejemV2(:,2),'g*');
title('Relación entre Titulos y Relación entee Palabras ',
      'FontWeight','bold','FontSize',14);
xlabel('Correlación','FontWeight','bold');
ylabel('Correlación','FontWeight','bold');
grid on
```

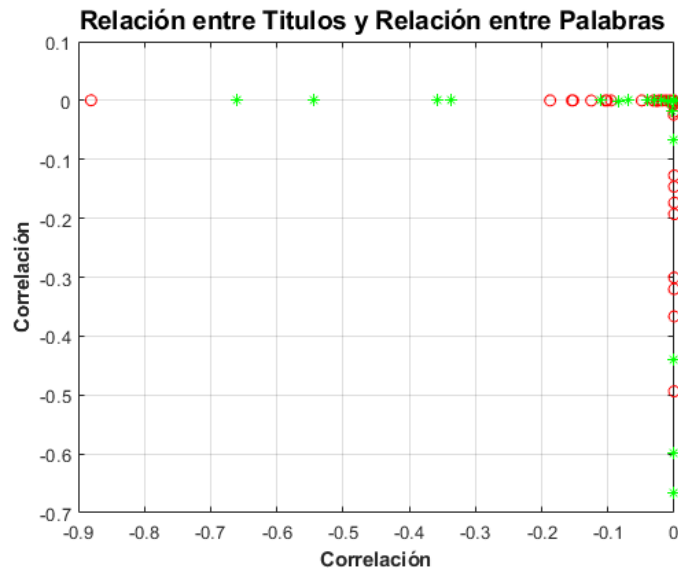


Figura 6: Representación EjemU2 y EjemV2

(2) Para la siguiente visualización ordenamos primero las matrices `ejemV2` y `ejemU2` y los almacenamos en otras variables eliminando además sus filas nulas. Los vectores `F1` y `F2` son las palabras o títulos de la lista sin los títulos o palabras eliminados de las otras dos matrices

```
[pal,F1]=creaMatriz3D(PalabrasTotales,ejemU2,palabras);
[tit,F2]=creaMatriz3D(TitulosTotales,ejemV2,titulos);
```

Ejecutamos entonces el siguiente *script*

```
figure('name','Representación 3D Palabras')
plot3(pal(:,1),pal(:,2),pal(:,3),'ro')
d=num2str(pal(:,3));
text(pal(:,1),pal(:,2),pal(:,3),d,'BackgroundColor',[1 1 .6])
title('Representación de la Relación entre Palabras
','FontWeight','bold','FontSize',14);
grid on
```

que nos proporciona la gráfica:

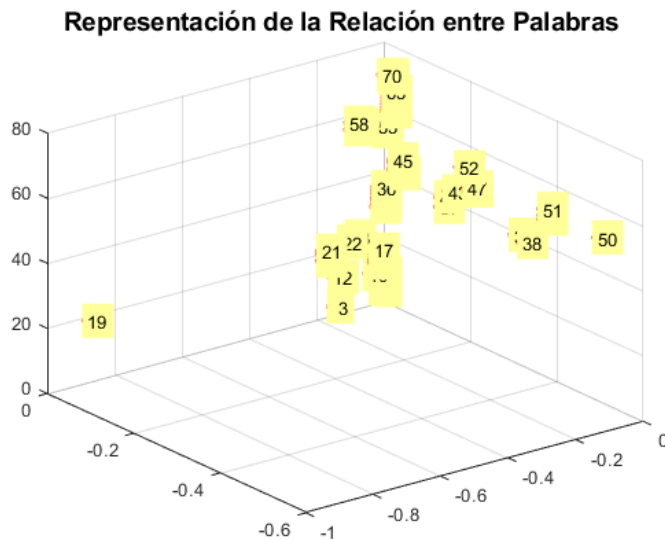


Figura 7: Representación 3D Palabras

En esta representación, en la que el tercer eje añadido indica la posición de cada una de las palabras en la base de datos podemos apreciar como se han ordenado. Deducimos que palabras o vectores con un valor de y semejante tienen cierta relación entre ellos (aunque no podamos medir claramente esta relación) al igual que vectores con un valor parecido de x están también relacionados.

(3) Finalmente podemos hacer una visualización similar teniendo ahora en cuenta los títulos. Nuevamente el eje Z nos indica únicamente la posición del título en la base de datos, evitando que estos se solapen como ocurría con la gráfica en XY.

```
figure('name','Representación 3D Títulos')
plot3(tit(:,1),tit(:,2),tit(:,3),'ro')
d=num2str(tit(:,3));
text(tit(:,1),tit(:,2),tit(:,3),d,'BackgroundColor',[1 1 .6])
title('Representación de la Relación entre Titulos
','FontWeight','bold','FontSize',14);
grid on
```

La gráfica obtenida es ahora:

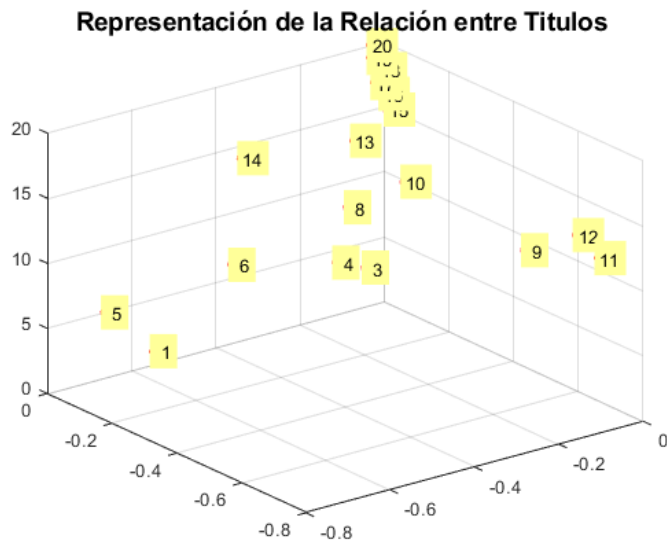


Figura 8: Representación 3D Títulos

Análisis comparativo

(A) La forma más sencilla es usar la matriz de incidencia. Podemos hacer diferentes comparaciones:

(a) Palabras con títulos. Este es el caso más sencillo porque se trata simplemente de usar la matriz de incidencia (como ya hemos visto anteriormente).

(b) Palabras con palabras. En este caso construimos una nueva matriz `pal2` comparando palabra a palabra:

```
for i=1:PalabrasTotales
    for j=1:PalabrasTotales
        pal2(i,j)=ejem2(i,:)*ejem2(j,:)' ;
        pal2(i,j)=pal2(i,j)/(norm(ejem2(i,:))*norm(ejem2(j,:)'));
    end
end
```

(c) Títulos con títulos. Usamos ahora el *script*:

```
for i=1:TitulosTotales
    for j=1:TitulosTotales
        tit2(i,j)=ejem2(:,i)'*ejem2(:,j);
        tit2(i,j)=tit2(i,j)/(norm(ejem2(:,i))'*norm(ejem2(:,j))));
    end
end
```

(B) Otra posibilidad es comparar títulos mediante búsqueda de palabras. En este método se realiza una búsqueda a partir de un vector formado por unas palabras determinadas. Este vector se compara con la matriz de incidencia proporcionando cuyos valores están entre 0 y 1 siendo 1 una alta afinidad de resultado. Fijando un valor del error podemos cuantificar la relación de nuestra búsqueda. Para ello usamos la función `Metodo1`:

```
function [PalabrasBusqueda, TitulosRelacionados,
valors]=Metodo1(y,ejem,error,palabras,titulos)
x=find(y==1);
PalabrasBusqueda=palabras(x);
comparacion=normal(y,ejem,length(titulos));
x1=find(comparacion>error);
indicador=0;
for j=1:length(titulos)
    if isnan(comparacion(1,j))==1
        comparacion(1,j)=0;
    end
    if comparacion(1,j)>error
        indicador=1;
    end
end
if indicador==0
    fprintf('La busqueda no ha devuelto ningún resultado\n')
else
    comparacion=comparacion';
    [~,C]=sort(comparacion,'descend');
    valors=[comparacion(C,:),C];
    TitulosRelacionados=titulos(C(1:length(x1)));
end
end
```

Para dicha función hemos usado también el *script*:

```
function comparativa=normal(y,MI,TitulosTotales)
ynorm=y/norm(y);
for i=1:TitulosTotales
    MInorm(:,i)=MI(:,i)/(norm(MI(:,i)));
end
comparativa=ynorm*MInorm;
end
```

Además para introducir el vector de palabras usamos también:

```

function vectory=creavectory(vectorpalabras,palabras)
tam=length(palabras);
vectory=zeros(1,tam);
vectory(vectorpalabras)=1;
end

```

Ilustramos el uso mediante dos ejemplos:

(1) Si usamos:

```

[PalabrasBusqueda, TitulosRelacionados,valor]=Metodo1(y,ejem,error,palabras,
titulos);
% Ejemplo 1
fprintf('Ejemplo 1:\n')
fprintf('Las palabras buscadas para el método 1 son:\n');
for idx=1:length(PalabrasBusqueda)
    PalBusc=PalabrasBusqueda{idx};
    fprintf('- %s \n',PalBusc)
end
fprintf('\n');
fprintf('El título %d es el que está mas relacionado con la búsqueda con un valor de:
%d\n',valor(1,2), valor(1,1))
fprintf('Los títulos relacionados ordenados son:\n')
for idx=1:length(TitulosRelacionados)
    Ti=TitulosRelacionados{idx};
    fprintf('- %s \n',Ti)
end
fprintf('\n');
u=find(valor(:,1)>error);
ejemplo1valores=[valor(u,2),valor(u,1)];

```

y ejecutamos:

```

% Ejemplo 1:
% buscamos las palabras: matrices y combinadas
y=creavectory([3 19],palabras);
% con un error de 0.2
error=0.2;

```

obtenemos como resultado que para las palabras buscadas para el método 1 (combinadas y matrices) el título 1 es el que está mas relacionado con la búsqueda con un valor de: $9,486833e - 01$ siendo además los títulos relacionados (ordenados):

- matrices combinadas de algunos tipos de matrices
- algunos resultados sobre b-matrices y matrices con inversa positiva

- estudio de la clase de matrices $k,s+1$ -potentes
- problemas de completacion de matrices parciales

(2) Si usamos:

```
y1=creavectory([5 12 65],palabras);
```

obtenemos que para las palabras buscadas para el método 1 (que son ahora analisis, estudio y plano) el título 13 es el que está mas relacionado con la búsqueda con un valor de: $5,163978e - 01$. Además Los títulos relacionados ordenados son:

- analisis estrategico de los estudios tic en la universidad politecnica de valencia
- estudio de la clase de matrices $k,s+1$ -potentes
- desarrollo y analisis de algoritmos probabilisticos para la reconstruccion de modelos metabolicos a escala genomica
- modelo socio-demografico dinamico para el estudio de la sostenibilidad demografica desde factores de calidad de vida
- modelizacion matematica de la transmision de calor en el proceso del rectificado industrial plano.

Un estudio similar se podría hacer mediante el uso de la DVS juntocon el Teorema del rango aproximado (en este caso hemos tomado $k = 20$ obteniéndose el mismo resultado. El *script* correspondiente al método 2 sería:

```
function [PalabrasBusqueda, TitulosRelacionados, valors]=Metodo2(y,ejem,
error,palabras, titulos)
x=find(y==1);
PalabrasBusqueda=palabras(x);
% Teorema del rango aproximado para K=20
[ejemU,ejemS,ejemV]=svd(ejem);
rango=rank(ejem);
ejemSr=ejemS(1:rango,1:rango);
ejemUr=ejemU(:,1:rango);
ejemVr=ejemV(:,1:rango);
yU=y*ejemUr;
SV=ejemSr*ejemVr';
comparacion=normal(yU,SV,length(titulos));
x1=find(comparacion>error);
indicador=0;
for j=1:length(titulos)
    if isnan(comparacion(1,j))==1
        comparacion(1,j)=0;
    end
    if comparacion(1,j)>error
        indicador=1;
    end
end
```

```

        end
    end
    if indicador==0
        fprintf('La busqueda no ha devuelto ningún resultado\n')
    else
        comparacion=comparacion';
        [~,C]=sort(comparacion,'descend');
        valors=[comparacion(C,:),C];
        TitulosRelacionados=titulos(C(1:length(x1)));
    end
end
end

```

(C) Finalmente, en este último método se corrige el vector búsqueda de forma que se proyecta sobre el subespacio generado por las columnas de nuestra matriz de incidencia. A continuación se realizan los dos métodos anteriores pero con el nuevo vector de búsqueda (que hemos llamado ahora *yy* para el ejemplo 1 e *yy1* para el 2). Si bien esto debería devolver unos resultados más precisos realmente esto no es así por tratarse de una base de datos no muy grande. En las pruebas realizadas obtenemos los mismos títulos, aunque su valor de relación con la búsqueda devuelve un valor mas elevado. Cabe recordar que nuestra base de datos dista mucho de ser un buen ejemplo para realizar este tipo de pruebas. Así, haciendo la corrección para el ejemplo 1 (es decir cambiando el vector *y* por el *yy*):

```

rango=rank(ejem);
ejemUr=ejemU(:,1:rango);
x=find(y==1);
PalabrasBusqueda=palabras(x);
yy=ejemUr*ejemUr'*y';
yy=yy';

```

o bien (cambiando el vector *y1* por el *yy1*):

```

rango=rank(ejem);
ejemUr=ejemU(:,1:rango);
x=find(y1==1);
PalabrasBusqueda=palabras(x);
yy1=ejemUr*ejemUr'*y1';
yy1=yy1';

```

para el ejemplo 2, obtendríamos los mismos resultados.

5. Bibliografía

Referencias

- [1] Ben-Israel, Adi; Thomas N.E. Greville (2003). *Generalized Inverses*. Springer-Verlag.
- [2] Castro, J. (2012). *Un nuevo modelo ponderado para Sistemas de Recomendación Basados en Contenido con medidas de contingencia y entropía. (Trabajo tutelado)*. Jaén: Universidad de Jaén.
- [3] K. Fan, A. J. Hoffman, *Some metric inequalities in the space of matrices*, Proceedings of the American Mathematical Society 6, 111-116, 1955.
- [4] E. Beltrami, Sulle funzioni bilineari, *Giornale di Matematiche ad Uso degli Studenti Delle Università 11*, 98-106, 1873. Traducido al inglés por D. Boley: Technical Report, University of Minnesota, Department of Computer Science, 90- 37, 1990.
- [5] I. C. Gohberg, M. G. Krein, *Introduction to the Theory of Linear Nonselfadjoint Operators*. American Mathematical Society, Providence, Rhode Island, 1969.
- [6] Campos, L., Fernández, J., Huete J. y Rueda, M. *Uso de conocimiento estructurado en un sistema de recomendación basado en Contenido. Granada: Universidad de Granada*.
- [7] Funk, S. *Netflix Update: Try This at Home*. <http://sifter.org/simon/journal/20061211.html>, Diciembre 2006.
- [8] G. H. Golub, C. Van Loan, *Matrix Computations*, 3rd edition, Johns Hopkins University Press, Baltimore, 1996.
- [9] Tom M. Apostol. *Calculus*. Ed. Reverte, 1973.
- [10] C. Jordan, Mémoire sur les formes bilinéaires, *Journal de Mathématiques Pures et Appliquées* 19, 35-54, 1874.
- [11] C. Jordan, Sur la réduction des formes bilinéaires, *Comptes Rendues de l'Académie des Sciences* 78, 614-617, 1874.
- [12] C. Lanczos, *Linear systems in self-adjoint form*, *American Mathematical Monthly* 65, 665-679, 1948.
- [13] Moore, E. H. (1920). *On the reciprocal of the general algebraic matrix*. *Bulletin of the American Mathematical Society* 26 (9): 394-395.
- [14] H. Nateman, A formula for the solving function of a certain integral equation of the second kind. *Transactions of the Cambridge Philosophical Society*, 20, 179-187, 1908.

- [15] E. Picard, Sur un théorème général relatif aux intégrales de première espèce et sur quelques problèmes de physique mathématique. *Rendiconti del Circolo Matematico di Palermo*, 25, 79–97, 1910.
- [16] J. Shlens, *A Tutorial on Principal Component Analysis*. Center for Neural Science, NYU y Systems Neurology Laboratory, Salk Institute for Biological Studies La Jolla. 2009.
- [17] H. Zou, T. Hastie y R. Tibshirani, "Sparse Principal Component Analysis". *Journal of Computational and Graphical Statistics*, 15, pp.265-286. 2006.
- [18] Bronkhorst, Adelbert W. (2000). «*The Cocktail Party Phenomenon: A Review on Speech Intelligibility in Multiple-Talker Conditions*» (pdf). *Acta Acustica united with Acustica* 86: 117-128. Consultado el 18 de abril de 2010.
- [19] G. Strang, *Linear Algebra and Its Applications*, 3rd edition, Harcourt BraceJovanovich, 1988.
- [20] J.J. Sylvester, *Sur la réduction biorthogonale d'une forme linéo-linéaire à sa forme canonique*, *Comptes Rendues de l'Academie des Sciences* 108, 651-653.
- [21] E. Schmidt, Zur Theorie der linearen und nichtlinearen Integralgleichungen. I Teil. Entwicklung willkürlichen Funktionen nach System vorgeschriebener. *Mathematische Annalen*, 63, 433–476, 1907.
- [22] Wood, N; Cowan, N (enero de 1995). «*The cocktail party phenomenon revisited: how frequent are attention shifts to one's name in an irrelevant auditory channel?*». *J Exp Psychol Learn Mem Cogn* 21 (1): 255-60.
- [23] I. Zaballa, *Valores singulares. ¿Qué son?. ¿Para qué sirven?*, Departamento de Matemática Aplicada y EIO, Universidad del País Vasco.

Apéndices

A. Álgebra lineal

- (1) **Ecuación lineal en \mathbb{R}^n** . Conjunto de la forma $\{(x_1, \dots, x_n) \in \mathbb{R}^n : a_1x_1 + \dots + a_nx_n = b \text{ con } a_1, \dots, a_n \in \mathbb{R}\}$.
- (2) **Sistema de ecuaciones lineales en \mathbb{R}^n** . Conjunto formado por una cantidad finita de ecuaciones lineales de \mathbb{R}^n .

- (3) **Matriz de tamaño** $m \times n$. Colección formada por $m \cdot n$ números reales ordenados en la forma

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

- (4) **Rango de una matriz.** Número de pivotes de su forma escalonada (o de unos principales de su forma escalonada reducida).
- (5) **Aplicación lineal.** Una aplicación $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ se dice lineal si existe una matriz A de tamaño $m \times n$ tal que

$$T(\mathbf{x}) = A \cdot \mathbf{x} \text{ para todo } \mathbf{x} \in \mathbb{R}^n.$$

- (6) **Aplicación lineal invertible.** Aplicación $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ que verifica la propiedad de que

$$\text{para cada } \mathbf{y} \in \mathbb{R}^n \text{ existe un } \mathbf{x} \in \mathbb{R}^n \text{ tal que } T(\mathbf{x}) = \mathbf{y}.$$

- (7) **Aplicación lineal inversa.** La aplicación inversa de la aplicación lineal $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ se denota por $T^{-1} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ y se define como $T^{-1}(\mathbf{y}) = \mathbf{x}$ siempre que $T(\mathbf{x}) = \mathbf{y}$.

- (8) **Matriz invertible.** Matriz cuadrada A de forma que la aplicación lineal $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ definida por $T(\mathbf{x}) = A \cdot \mathbf{x}$ es una aplicación lineal invertible.

- (9) **Matriz inversa.** Si A es una matriz cuadrada de forma que la aplicación lineal $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ definida por $T(\mathbf{x}) = A \cdot \mathbf{x}$ es una aplicación invertible entonces la aplicación lineal inversa siempre es lineal y la matriz que la define se llama matriz inversa de A . Se denota por A^{-1} .

- (10) **Subespacio vectorial de** \mathbb{R}^n . Es un subconjunto no vacío X de \mathbb{R}^n que verifica:

- si $\mathbf{x}, \mathbf{y} \in X$ entonces $\mathbf{x} + \mathbf{y} \in X$,
- si $\mathbf{x} \in X$ entonces $\alpha \mathbf{x} \in X$ para todo $\alpha \in \mathbb{R}$.

- (11) **Núcleo de una aplicación lineal.** Dada una aplicación lineal $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ el núcleo de T es el conjunto

$$\ker(T) = \{\mathbf{x} \in \mathbb{R}^n : T(\mathbf{x}) = \mathbf{0}\}.$$

- (12) **Imagen de una aplicación lineal.** Dada una aplicación lineal $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ la imagen de T es el conjunto

$$\text{Im}(T) = \{\mathbf{y} \in \mathbb{R}^m : \exists \mathbf{x} \in \mathbb{R}^n \text{ tal que } T(\mathbf{x}) = \mathbf{y}\}.$$

- (13) **Combinación lineal.** Una combinación lineal de los vectores $\mathbf{x}_1, \dots, \mathbf{x}_k$ es una expresión de la forma

$$\alpha_1 \mathbf{x}_1 + \dots + \alpha_k \mathbf{x}_k \text{ para ciertos } \alpha_1, \dots, \alpha_k \in \mathbb{R}.$$

- (14) **Envoltura lineal.** La envoltura lineal del conjunto de vectores $\{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ es el conjunto formado por todas las combinaciones lineales de los vectores del conjunto. Se denota por $\mathcal{L}\{\mathbf{x}_1, \dots, \mathbf{x}_k\}$.
- (15) **Sistema generador de un subespacio vectorial X de \mathbb{R}^n .** Decimos que un conjunto de vectores $\{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ es un sistema generador de un subespacio vectorial X de \mathbb{R}^n si $\mathcal{L}\{\mathbf{x}_1, \dots, \mathbf{x}_k\} = X$.
- (16) **Conjunto linealmente independiente o libre.** Es un conjunto de vectores $\{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ de \mathbb{R}^n que verifica la propiedad: si $\alpha_1 \mathbf{x}_1 + \dots + \alpha_k \mathbf{x}_k = \mathbf{0}$ entonces $\alpha_1 = \dots = \alpha_k = 0$.
- (17) **Conjunto linealmente dependiente o ligado.** Es un conjunto de vectores de \mathbb{R}^n que no es linealmente independiente.
- (18) **Base de un subespacio vectorial X de \mathbb{R}^n .** Es un conjunto de vectores de \mathbb{R}^n que es un sistema generador de X y además es linealmente independiente.
- (19) **Dimensión de un subespacio vectorial X de \mathbb{R}^n .** Es el número de vectores que tiene una base de X . Se denota por $\dim(X)$.
- (20) **Coordenadas de un vector respecto de una base.** Las coordenadas de un vector $\mathbf{x} \in \mathbb{R}^n$ respecto de una base $\mathcal{B} = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ de un subespacio X de \mathbb{R}^n es otro vector $(\alpha_1, \dots, \alpha_k) \in \mathbb{R}^k$ tal que $\mathbf{x} = \alpha_1 \mathbf{x}_1 + \dots + \alpha_k \mathbf{x}_k$.
- (21) **Teorema de caracterización de aplicaciones lineales.** Una aplicación $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ es lineal si, y sólo si,
- $T(\mathbf{x} + \mathbf{y}) = T(\mathbf{x}) + T(\mathbf{y})$ para todo $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$,
 - $T(\alpha \mathbf{x}) = \alpha T(\mathbf{x})$ para todo $\mathbf{x} \in \mathbb{R}^n$ y todo $\alpha \in \mathbb{R}$.
- (22) **Teorema de caracterización de aplicaciones lineales invertibles.** Sea $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ la aplicación lineal definida por $T(\mathbf{x}) = A \cdot \mathbf{x}$. Son equivalentes:
- (a) T es lineal, (b) El sistema $A \cdot \mathbf{x} = \mathbf{b}$ es compatible determinado para todo $\mathbf{b} \in \mathbb{R}^n$,
 - (c) La forma escalonada reducida de A es I_n , (d) $\rho(A) = n$, (e) $\det(A) \neq 0$.
- (23) **Teorema de la base.** Todas las bases de un subespacio vectorial de \mathbb{R}^n tienen el mismo número de vectores.

(24) **Teorema de las dimensiones.** Sea $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ una aplicación lineal. Entonces

$$\dim(\ker(T)) + \dim(\text{Im}(T)) = n.$$

B. Ortogonalidad y diagonalización

1. **Valor propio y vector propio.** Decimos que $\alpha \in \mathbb{R}$ es un valor propio de A si existe un vector $x \in \mathbb{R}^n$ no nulo tal que $Ax = \alpha x$. En este caso decimos que x es vector propio de A asociado al valor propio α .
2. **Polinomio característico.** Es el polinomio de grado n dado por $d_A(\lambda) = \det(A - \lambda I_n)$.
3. **Multiplicidad algebraica de un valor propio α .** Es el número de veces que aparece repetido α como raíz del polinomio característico. Se denota por α .
4. **Autoespacio asociado a un autovalor α .** Es el subespacio vectorial de \mathbb{R}^n , $E(\alpha) = \ker(A - \alpha I_n)$.
5. **Multiplicidad geométrica de un valor propio α .** Es la dimensión de $E(\alpha)$. Se denota por $MG(\alpha)$.
6. **Matriz diagonalizable.** A es diagonalizable si existe una matriz P real de tamaño n invertible y una matriz D real de tamaño n diagonal de forma que $A \cdot P = P \cdot D$, es decir si $A = P \cdot D \cdot P^{-1}$.
7. **Matriz simétrica.** Si $A^T = A$.
8. **Matriz ortogonal.** Si $A^T A = I_n$, es decir si $A^{-1} = A^T$.
9. **Matriz diagonalizable ortogonalmente.** A es diagonalizable ortogonalmente si existe una matriz P de tamaño n ortogonal e y una matriz D de tamaño n diagonal de forma que $A = P \cdot D \cdot P^T$.
10. **Teorema caracterización diagonalización.** A es diagonalizable si, y sólo si,
 - a) todos sus autovalores son reales,
 - b) para cada autovalor $\alpha \in \mathbb{R}$ de A , se cumple que $MA(\alpha) = MG(\alpha)$.
11. **Teorema caracterización diagonalización ortogonal.** A es diagonalizable ortogonalmente si, y sólo si, A es simétrica.

Sea E un espacio vectorial **real**.

1. **Producto escalar en E .** Es una aplicación $(\cdot|\cdot) : E \times E \rightarrow \mathbb{R}$ que verifica:
 - a) es **bilineal**, es decir es lineal en cada componente,
 - b) es **simétrica**, es decir $(\mathbf{x}|\mathbf{y}) = (\mathbf{y}|\mathbf{x})$, y
 - c) es **definida positiva**, es decir $(\mathbf{x}|\mathbf{x}) \geq 0$ y $(\mathbf{x}|\mathbf{x}) = 0$ si, y sólo si, $\mathbf{x} = \mathbf{0}$.
2. **Norma asociada a un producto escalar.** Viene determinada por $\|\cdot\| = +\sqrt{(\cdot|\cdot)}$.
3. **Vectores ortogonales.** Se dice que $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ son ortogonales si $(\mathbf{x}|\mathbf{y}) = 0$.
4. **Vectores unitarios.** Decimos que $\mathbf{x} \in \mathbb{R}^n$ es unitario si $(\mathbf{x}|\mathbf{x}) = 1$, es decir si, $\|\mathbf{x}\| = 1$.
5. **Conjunto ortogonal.** Familia de vectores ortogonales dos a dos.
6. **Conjunto ortonormal.** Familia de vectores unitarios ortogonales dos a dos.
7. **ángulo formado por dos vectores.** El ángulo formado por $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ es el único $\theta \in [0, \pi]$ tal que

$$\cos(\theta) = \frac{(\mathbf{x}|\mathbf{y})}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}.$$

8. **Proyección ortogonal de un vector sobre otro.** La proyección ortogonal de un vector $\mathbf{x} \in \mathbb{R}^n$ sobre un vector no nulo $\mathbf{y} \in \mathbb{R}^n$ es el vector \mathbb{R}^n definido por

$$\mathbf{proy}_{\mathbf{y}}(\mathbf{x}) = \frac{(\mathbf{x}|\mathbf{y})}{(\mathbf{y}|\mathbf{y})}\mathbf{y}.$$

9. **Proyección ortogonal de un vector sobre un subespacio vectorial de E .** La proyección ortogonal de un vector $\mathbf{x} \in \mathbb{R}^n$ sobre un subespacio vectorial U de E es

$$\mathbf{proy}_{\mathbf{y}}(\mathbf{x}) = \mathbf{proy}_{\mathbf{y}_1}(\mathbf{x}) + \mathbf{proy}_{\mathbf{y}_2}(\mathbf{x}) + \dots + \mathbf{proy}_{\mathbf{y}_k}(\mathbf{x}),$$

siendo $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k\}$ es una base ortogonal de U .

10. **Subespacio ortogonal.** El subespacio ortogonal de un subespacio vectorial U de E es

$$U^\perp = \{\mathbf{x} \in \mathbb{R}^n : (\mathbf{x}|\mathbf{y}) = 0, \text{ para todo } \mathbf{y} \in U\}.$$

C. Diagonalización ortogonal

PASO 1 Se puede calcular de matrices A cuadradas y simétricas de tamaño n .
Cálculo de D .

Calculamos los valores propios de A .

Como la matriz es diagonalizable (por ser simétrica) las multiplicidades algebraicas y geométricas de todos los autovalores coinciden.

Ordenamos los valores propios (como queramos, mi consejo es primero los positivos, luego los negativos y al final los nulos).

La matriz D es la que tiene en la diagonal a los valores propios repetidos tantas veces como sus multiplicidades.

PASO 2 **Cálculo de P .**

Para cada valor propio λ calculamos una base del subespacio propio asociado

$$E(\lambda) = \ker(A - \lambda I_n).$$

Como del paso anterior ya sabemos la multiplicidad geométrica (pues coincide con la algebraica) entonces para calcular una base basta con tener un sistema generador de $E(\lambda)$ formado por la misma cantidad de vectores que la multiplicidad.

Si la base de $E(\lambda)$ no es ortonormal aplicamos el MOGS.

La matriz P es la que tiene como columnas las bases ortonormales de cada $E(\lambda)$ ordenadas según el orden que hayamos elegido para D .

PASO 3 **Cálculo de la diagonalización ortogonal.**

La diagonalización ortogonal de A es

$$A = P \cdot D \cdot P^T.$$