Máster Universitario en Computación Paralela y Distribuida
Algoritmos Paralelos en Procesamiento de la Señal

Tema 5.
La Descomposición en Valores Singulares.

Curso 2015-2016

Bibliografía para el Tema 4:

"Matrix Computations". G.Golub & C.Van Loan
Capítulos 2 y 7

Lecturas recomendadas para la SVD:
"Matrix Computations". G.Golub & C.Van Loan
Capítulo 2. Punto 2.5 y Capítulo 5. Punto 5.5
Capítulo 8. Punto 8.3

V.Klema, A.Laub
"The Singular Value Decomposition: Its computation and some applications". IEEE Trans. On Automatic Control. Vol. Ac-25 . No. 2. April (1980)

La Descomposición en Valores Singulares (SVD)

Proposición

Sea la matriz real A, mxn, de rango r y sea $p = \min(m,n)$

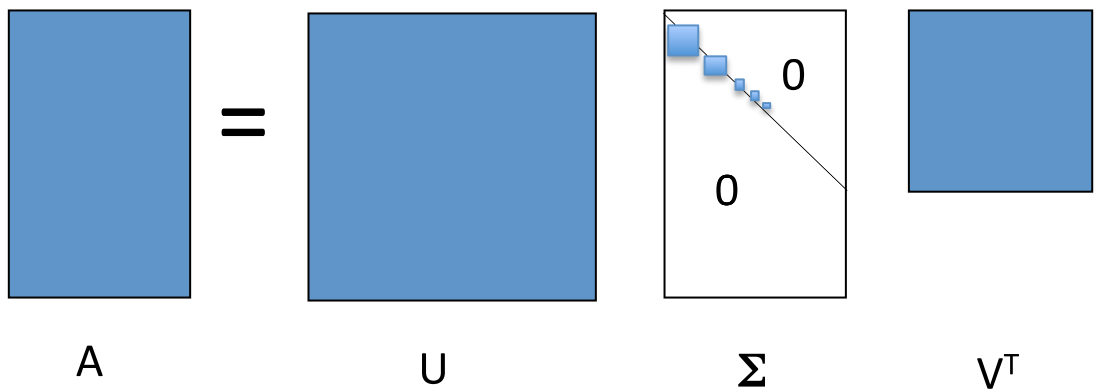
Existen matrices ortogonales U, mxm, y V, nxn, tales que $A = U\Sigma V^T$ donde

$\Sigma = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix}$ $y\ S = diag(\sigma_1, \sigma_2, \ldots, \sigma_r)$, con $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \sigma_{r+2} = \ldots = \sigma_p = 0$

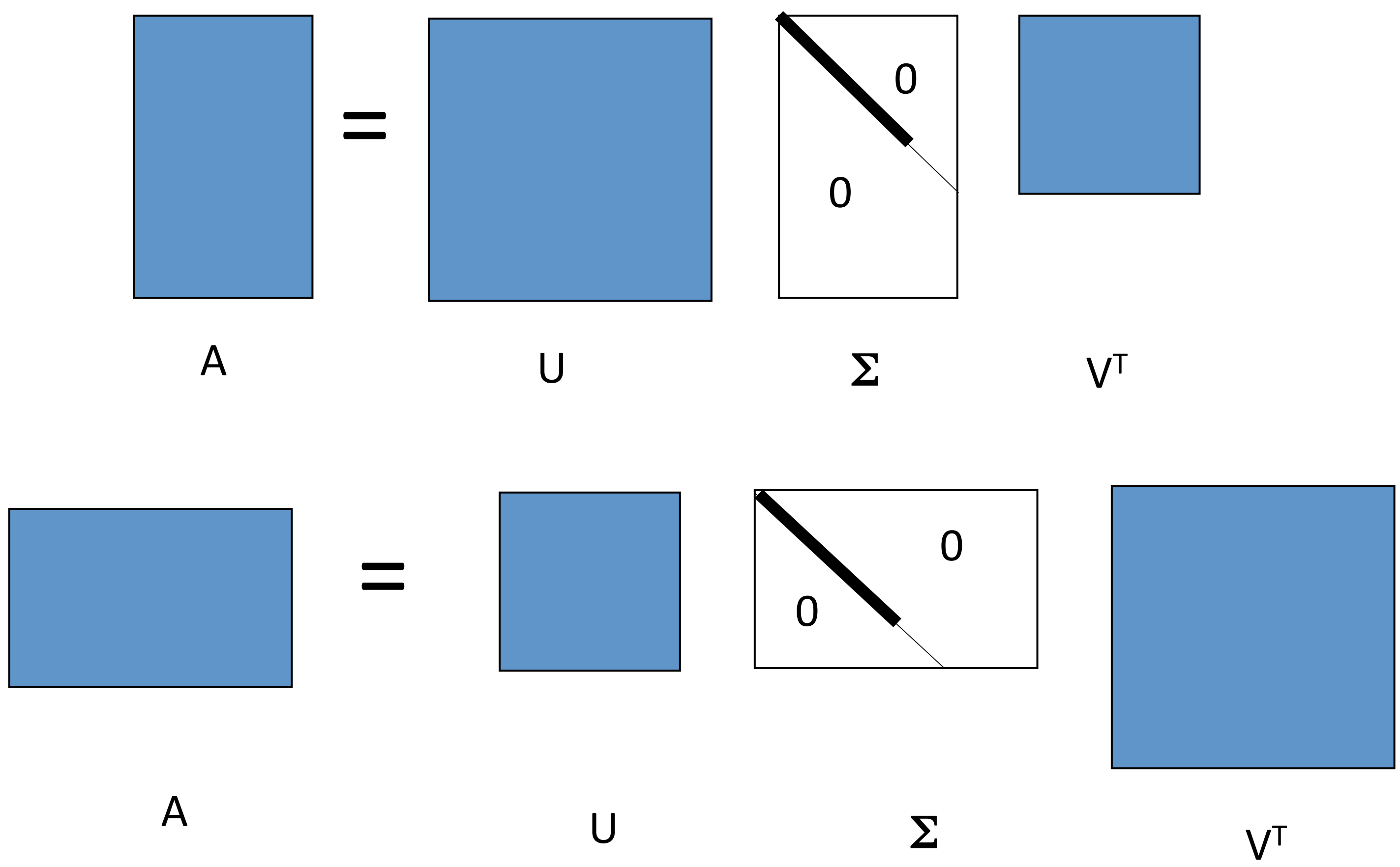
Los elementos $\sigma_1, \sigma_2, \ldots, \sigma_r, \sigma_{r+1}, \sigma_{r+2}, \ldots, \sigma_p$ se denominan valores singulares de A

Las columnas de U son los vectores singulares por la izquierda de A.

Las columnas de V son los vectores singulares por la derecha de A.



La Descomposición en Valores Singulares



Demostración

- Dado que *A* tiene rango *r*, A^TA es una matriz simétrica semidefinida positiva, que tiene r valores propios reales positivos y el resto nulos, y sus vectores propios, v_i, son ortonormales

$$\left(A^T A\right)v_i = \sigma_i^2 v_i, i = 1,2,\ldots,r \text{ con } \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$$

$$\left(A^T A\right)v_i = \sigma_i v_i, i = r+1, r+2,\ldots,n \text{ con } \sigma_i = 0$$

En forma matricial

$$\left(A^T A\right)V_1 = V_1 \begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & .. & \\ & & & \sigma_r^2 \end{bmatrix} = V_1 S^2 \Rightarrow V_1^T A^T A V_1 = S^2 \text{ con } S = diag(\sigma_1, \sigma_2, \ldots, \sigma_r)$$

Y por tanto

$$\left(S^{-1} V_1^T A^T A V_1 S^{-1}\right) = I \Rightarrow \left(A V_1 S^{-1}\right)^T \left(A V_1 S^{-1}\right) = I$$

$$\text{con } V_1 = [v_1, v_2, \ldots, v_r]$$

Además

$$\left(A^T A\right)V_2 = V_2\begin{bmatrix}0 & & & \\ & 0 & & \\ & & .. & \\ & & & 0\end{bmatrix} \Rightarrow V_2{}^T A^T A V_2 = 0 \Rightarrow \left(AV_2\right)^T\left(AV_2\right) = 0 \Rightarrow AV_2 = 0$$

$$\text{con } V_2 = [v_{r+1}, v_{r+2}, \ldots, v_n]$$

Definimos

$$U_1 = AV_1S^{-1} \in \Re^{mxr}, \text{con} \left(U_1\right)^T\left(U_1\right) = I \text{ y } U_1S = AV_1$$

Elegimos

$$U_2 = [u_{r+1}, u_{r+2}, \ldots, u_m] \in \Re^{mx(m-r)}, \text{con} \left(U_2\right)^T\left(U_2\right) = I \text{ y} \left(U_2\right)^T\left(U_1\right) = 0$$

Obsérvese que $U = [U_1\ U_2] \in \Re^{mxm}; V = [V_1\ V_2] \in \Re^{nxn}$

son matrices ortogonales y

$$U^T AV = \begin{bmatrix}U_1{}^T \\ U_2{}^T\end{bmatrix} A[V_1\ V_2] = \begin{bmatrix}U_1{}^T AV_1 & U_1{}^T AV_2 \\ U_2{}^T AV_1 & U_2{}^T AV_2\end{bmatrix} = \begin{bmatrix}S & 0 \\ U_2{}^T U_1 S & 0\end{bmatrix} = \begin{bmatrix}S & 0 \\ 0 & 0\end{bmatrix}$$

o bien $A = U\begin{bmatrix}S & 0 \\ 0 & 0\end{bmatrix}V^T$

Ejemplos

A =

3	4	1
5	2	1
7	3	2
2	1	4

```
>> svd(A)

ans =

   11.0990
    3.2575
    2.2805
```

```
>> [U,S,V]=svd(A)
```

U =

-0.4177	0.2038	-0.8847	0.0364
-0.4852	0.2151	0.3111	0.7884
-0.7048	0.1407	0.3402	-0.6064
-0.3056	-0.9447	-0.0694	0.0970

S =

11.0990	0	0
0	3.2575	0
0	0	2.2805
0	0	0

V =

-0.8310	0.2403	0.5016
-0.4560	0.2220	-0.8619
-0.3185	-0.9450	-0.0749

Ejemplos

```
A =

     2     1     4     6
     3     5    -1     2
     4     1     5     8
```

```
>> svd(A)

ans =

   13.0755
    5.5117
    0.8068
```

```
>> [U,S,V]=svd(A)

U =

   -0.5704    0.1778    0.8019
   -0.2482   -0.9680    0.0380
   -0.7830    0.1774   -0.5962


S =

   13.0755         0         0         0
         0    5.5117         0         0
         0         0    0.8068         0


V =

   -0.3837   -0.3336   -0.8267   -0.2408
   -0.1984   -0.8136    0.4905   -0.2408
   -0.4549    0.4655    0.2336   -0.7223
   -0.7787    0.0998    0.1459    0.6019
```

Invariancia de la SVD frente a transformaciones ortogonales

Los valores singulares de una matriz son invariantes frente a transformaciones ortogonales

Si $A = U \sum V^T$ y P y Q son matrices ortogonales $(P^T = P^{-1}, Q^T = Q^{-1})$ de tamaño adecuado:

- Si σ es un valor singular de A entonces σ es un valor singular de PAQ^T
- Si u es un vector singular por la izquierda de A
  entonces Pu es un vector singular por la izquierda de PAQ^T
- Si v es un vector singular por la derecha de A
  entonces Qv es un vector singular por la derecha de PAQ^T

Propiedades de la SVD

$$\|A\|_2 = \sigma_{\max}$$

$$\|A\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \ldots + \sigma_p^2}, \text{ con } p = \min(m,n).$$

Si A es invertible, $\|A^{-1}\|_2 = 1/\sigma_{\min}$

$$\kappa_2(A) = \frac{\sigma_{\max}}{\sigma_{\min}}$$

Si $A = U\Sigma V^T$, con $A \in \Re^{n\times n}$, entonces $|\det(A)| = \sigma_1 * \sigma_2 * \ldots * \sigma_n$

Propiedades de la SVD

Pseudoinversa de Moore-Penrose de $A \in \Re^{m\times n}$

$A^{+} = \arg\min_{X\in\Re^{n\times m}} \left\| AX - I_m \right\|_2$

Si $A = U\Sigma V^T$, con $rank(A) = r$, $A^{+} = V\Sigma^{+}U^T$, con $\Sigma^{+} = diag(\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_r^{-1}) \in \Re^{n\times m}$

Si $A = U\Sigma V^T$, con $rank(A) = n$, (A es de rango completo), $A^{+} = (A^T A)^{-1} A^T$

Si A es invertible, $A^{+} = A^{-1}$

Producto matriz-vector Ax

Los valores singulares de A son las longitudes de

los semiejes del hiperelipsode definido por $E = \left\{ Ax : \left\| x \right\|_2 = 1 \right\}$

Subespacios de $A \in \Re^{m\times n}$ con $rank(A) = r$, $p = \min(m,n)$

Si $A = U\Sigma V^T$, con $rank(A) = r$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_p = 0$

$null(A) = span\left\{ v_{r+1}, v_{r+2}, \ldots, v_n \right\}$

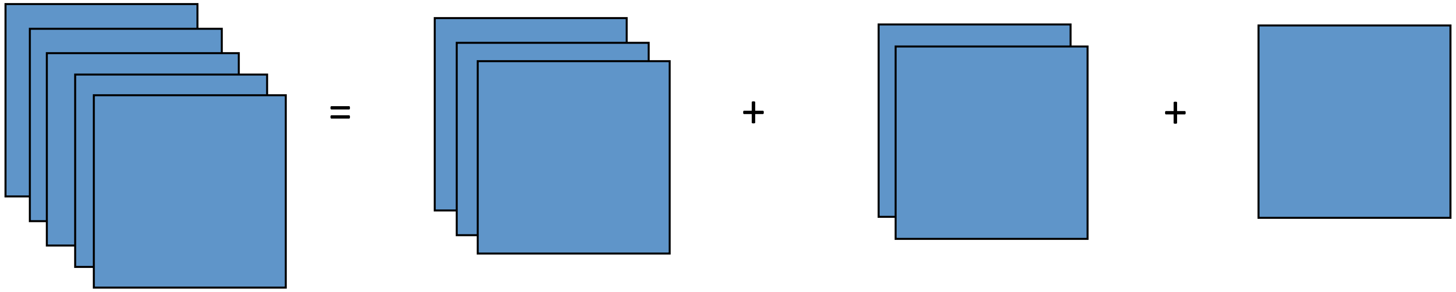
$range(A) = span\left\{ u_1, u_2, \ldots, u_r \right\}$

$$A = U_r \Sigma_r V_r^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

Propiedades de la SVD

Expansión de una matriz mediante la SVD

Si $A = U\sum V^T$, con $A \in \Re^{m\times n}$ y $rank(A) = r$, entonces $A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$



Resolución del Problema Lineal de Mínimos Cuadrados mediante la SVD: $\min_{x\in\Re^n} \|Ax-b\|_2$

$$\|Ax-b\|_2^2 = \|U^T AVV^T x - U^T b\|_2^2 = \|\Sigma y - U^T b\|_2^2 \text{ con } y = V^T x$$

$$\|Ax-b\|_2^2 = \sum_{i=1}^{r}\left(\sigma_i y_i - u_i^T b_i\right)^2 + \sum_{i=r+1}^{m}\left(u_i^T b_i\right)^2$$

Solución de mínimos cuadrados: $y_i = \dfrac{u_i^T b_i}{\sigma_i}, i = 1,2,\ldots,r$

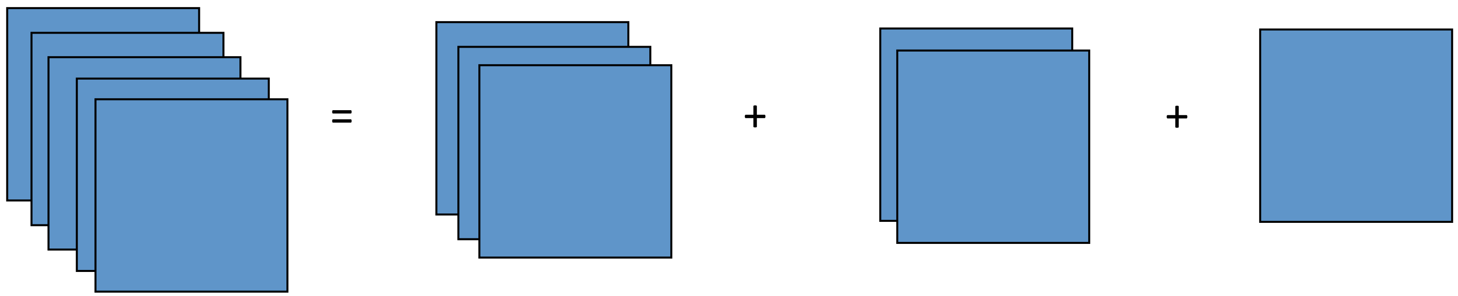
Solución de norma mínima: $y_i = 0,\ i = r+1, r+2,\ldots,n$

$$x_{LS} = Vy = \sum_{i=1}^{r}\frac{u_i^T b_i}{\sigma_i} v_i$$

Propiedades de la SVD

Aproximación de matrices

Si $A = U \sum V^T$, con $A \in \Re^{m \times n}$ y $rank(A) = r$, entonces $A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$

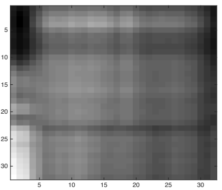
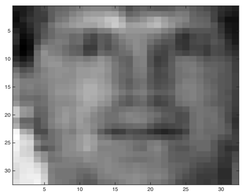
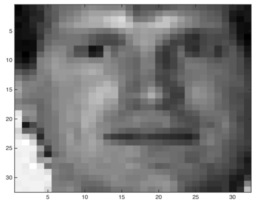
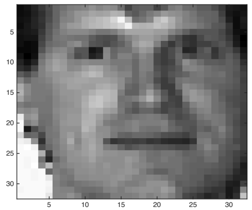
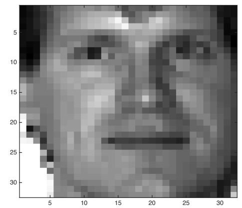


Si $B = \sum_{i=1}^{t} \sigma_i u_i v_i^T$, con $t < r$, entonces $B = \arg \min_{rank(X)=t} \left\| A - X \right\|_2$

Aproximación de matrices

$$A \in \Re^{32\times 32},\ \text{con } A_{ij} \in \{0,1,\ldots,255\},\ \ rank(A) = 32;$$

$$A = \sum_{i=1}^{t} \sigma_i u_i v_i^T,\ \text{con } t = 3,5,10,15,32$$

Propiedades de la SVD

Matrices con rango deficiente y aproximación de matrices

Si $A = U\sum V^T$, con $A \in \Re^{m\times n}$ y $rank(A) = r$, entonces $A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$

$$\boxed{\text{Si } A = U\Sigma V^T,\ \text{con } rank(A) = r,\ \text{y } k < r,\ \text{entonces } \min_{rank(B)=k} \left\| A - B \right\|_2 = \sigma_{k+1}}$$

El valor singular más pequeño de A es la distancia (medida como 2-norma) de A al conjunto de matrices de rango deficiente

Rango numérico de una matriz

Si $A = U\sum V^T$, $\delta > 0$ una cierta tolerancia, y $\sigma_1 \geq \sigma_2 \geq .... \geq \sigma_r > \delta \geq \sigma_{r+1} \geq \sigma_{r+2} \geq ... \geq \sigma_n$ decimos que el rango numérico de A es r

Relevancia de la SVD

- En algunos problemas los valores propios o el determinante no proporciona información útil sobre ciertas propiedades de las matrices.
- Esto es especialmente cierto cuando se trata de propiedades relacionadas con el rango de la matriz.
- Ejemplo:

$$b = \begin{bmatrix} 2^{1-n} \\ 2^{1-n} \\ \dots \\ 2^{1-n} \end{bmatrix}$$

$$A = \begin{bmatrix} -1 & 1 & \dots & 1 \\ & -1 & \dots & 1 \\ & & \dots & \dots \\ & & & -1 \end{bmatrix}$$

$$B = A + be_1^T; \ rank(B) = n-1; \det(B) = 0;$$

$\det(A)=(-1)^n$ $\lambda(A)= -1$ $\sigma_{min}(A)=2^{-n}$

Ejemplos

A =

2	3	1
5	6	1
8	9	1
2	1	-1

```
>> svd(A)
```

ans =

15.0027
1.7085
0.0000

A =

1.0000	0.5000	0.3333	0.2500	0.2000	0.1667
0.5000	0.3333	0.2500	0.2000	0.1667	0.1429
0.3333	0.2500	0.2000	0.1667	0.1429	0.1250
0.2500	0.2000	0.1667	0.1429	0.1250	0.1111
0.2000	0.1667	0.1429	0.1250	0.1111	0.1000
0.1667	0.1429	0.1250	0.1111	0.1000	0.0909

```
>> svd(A)
```

ans =

1.618899858924339
0.242360870575210
0.016321521319876
0.000615748354183
0.000012570757123
0.000000108279948

Relación entre la SVD y los valores propios

Si $A=U\Sigma V^T$, A, nxn, $\Sigma=\text{diag}(\sigma_1,\sigma_2,\ldots,\sigma_n)$

$$(A^T A)V_i = \sigma_i^{\,2} V_i, i = 1,2,\ldots,n$$

$$(AA^T)U_i = \sigma_i^{\,2} U_i, i = 1,2,\ldots,n$$

Cálculo de la SVD

Se podría calcular a partir de:

$$(A^T A)V_i = \sigma_i^2 V_i, i = 1,2,\ldots,n \qquad (AA^T)U_i = \sigma_i^2 U_i, i = 1,2,\ldots,n$$

No es buena idea calcular A^TA o AA^T debido a la pérdida de información que produce esta operación:

$$\text{Si } A = \begin{bmatrix} 1 & 1 \\ 0 & \mu \\ \mu & 0 \end{bmatrix} \text{ con } fl(1+\mu^2) = 1 \Rightarrow (A^T A) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

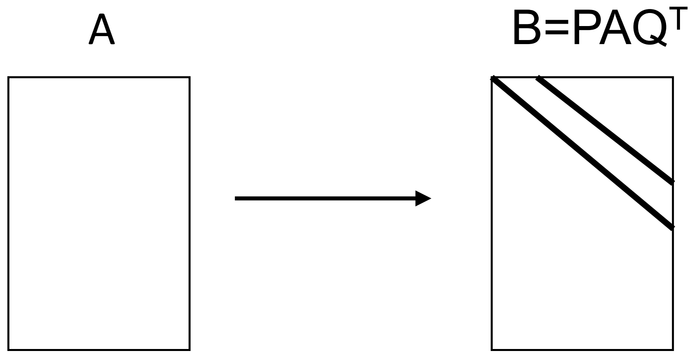
Algoritmos más utilizados:

1. Reducir la matriz A a una forma condensada utilizando transformaciones ortogonales, P y Q: $B=PAQ^T$
2. Calcular la SVD de B mediante un proceso iterativo

Algoritmo de Golub & Reinsch

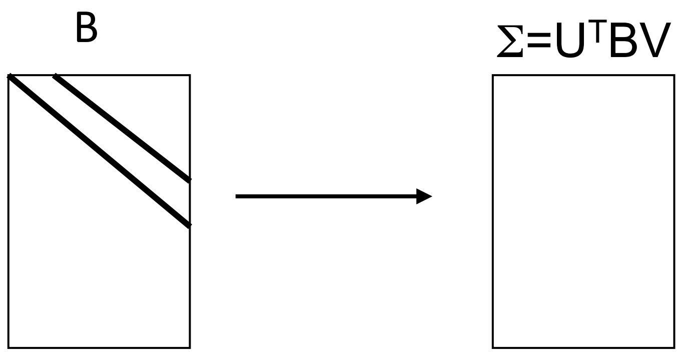
Mejor algoritmo secuencial para calcular la SVD

1. Bidiagonalizar A usando transformaciones ortogonales



$$P_4 \; P_3 \; P_2 \; P_1 \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} Q_1^T \; Q_2^T$$

2. Aplicar el algoritmo iterativo QR a la matriz B^TB para diagonalizarla, pero sin formar explícitamente B^TB



$Z_1 = Z \;\; (Z = B^T B)$

Para $i = 1,2,\ldots,\infty$

$\quad [Q_i, R_i] = qr(Z_i);$

$\quad Z_{i+1} = R_i * Q_i$

Finpara

Algoritmos "Raíz Cuadrada" para el cálculo de la SVD

Muchos algoritmos para calcular valores propios funcionan de forma similar. Ejemplos:

Algoritmo Iterativo QR

$Z_1 = Z$

Para $i = 1,2,\ldots,\infty$

$\quad [Q_i, R_i] = qr(Z_i);$

$\quad Z_{i+1} = R_i * Q_i$

Finpara

Algoritmo de Jacobi

$Z_1 = Z$

Para $i = 1,2,\ldots,\infty$

$\quad$ Elegir un par (r,s)

$\quad$ Calcular Q_i tal que $(Q_i^T Z_i Q_i)_{rs} = 0;$

$\quad Z_{i+1} = Q_i^T Z_i Q_i$

Finpara

Puesto que existe una relación entre la SVD de A y la descomposición en Valores Propios de $A^T A$, podría aprovecharse para establecer una relación entre la sucesión $\{Z_i\}$ y una hipotética sucesión de $\{A_i\}$ de forma que se mantuviera en cada caso la igualdad $Z_i = A_i^T A_i$

Idea básica de los algoritmos "raíz cuadrada" para calcular la SVD de A

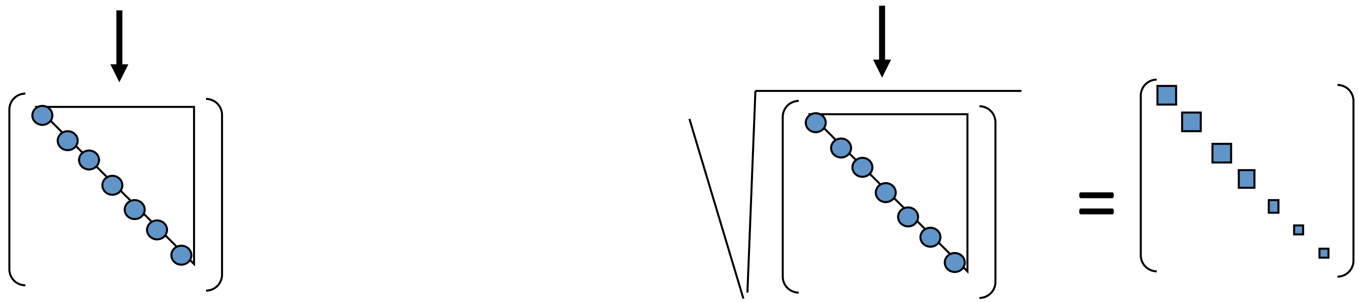
Sea $A_0 = A$ y $Z_0 = A_0^T A_0$

Supongamos que A_0 puede expresarse como $A_0 = f(M_0)$ donde f representa una cierta transformación ortogonal aplicada sobre A_0 y además se verifica que $Z_0 = M_0 M_0^T$

Construimos las sucesiones

Valores propios		SVD
Z_0	=	$M_0 M_0^T$
Z_1	=	$M_1 M_1^T$
$\downarrow$		$\downarrow$
Z_i	=	$M_i M_i^T$
Z_{i+1}	=	$M_{i+1} M_{i+1}^T$

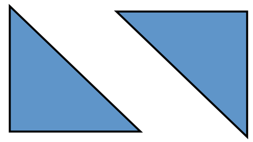
Algoritmo "Raíz Cuadrada"



Algoritmo “Raíz Cuadrada” para el cálculo de la SVD de A basado en el algoritmo iterativo QR

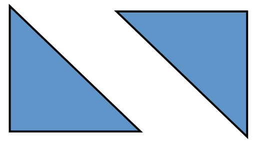
Sea $A = Q_0 R_0$ la descomposición QR de A y

$$Z_0 = A^T A = R_0^{\,T} Q_0^{\,T} Q_0 R_0 = R_0^{\,T} R_0 \longrightarrow$$



Procedamos por inducción. Supongamos que en la **<u>etapa i</u>** del Algoritmo iterativo QR se tiene

$$Z_i = R_i^{\,T} R_i \longrightarrow$$



Si se calcula la descomposición QR de

$R_i^{\,T} = Q_{0i} R_{0i}$ se tiene $Z_i = R_i^{\,T} R_i = Q_{0i} R_{0i} R_i = Q_{0i} R_{0ii}$

es decir $Z_i = Q_{0i} R_{0ii}$ descomposición QR de Z_i

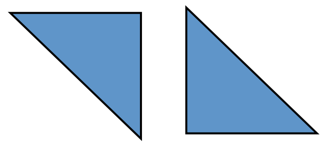
$$Z_i = Q_{0i} R_{0ii}$$

En la **<u>etapa i+1</u>** del Algoritmo iterativo QR se tiene

$$Z_{i+1} = R_{0ii} Q_{0i} = R_{0i} R_i Q_{0i} = R_{0i} R_{0i}^{\,T} Q_{0i}^{\,T} Q_{0i}$$

Es decir, se tiene

$$Z_{i+1} = R_{0i} R_{0i}^{\,T} \longrightarrow$$

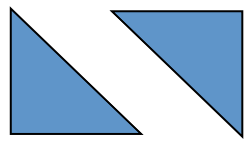


Si calculamos $R_{0i}^{\,T} = Q_{1i} R_{1i}$ se tiene

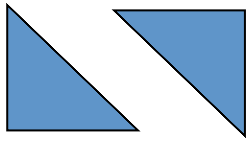
Es decir

$$Z_{i+1} = R_{1i}^{\,T} Q_{1i}^{\,T} Q_{1i} R_{1i} = R_{1i}^{\,T} R_{1i}$$

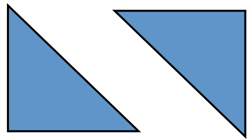
$$Z_{i+1} = R_{1i}^{\,T} R_{1i} \longrightarrow$$



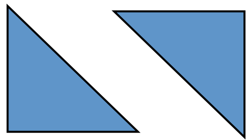
Es decir

$$Z_0 = R_0^{\,T} R_0 \longrightarrow$$ 

$$\downarrow$$

$$Z_i = R_i^{\,T} R_i \longrightarrow$$ 

$$\downarrow$$

$$Z_{i+1} = R_{i+1}^{\,T} R_{i+1} \longrightarrow$$ 

Con R_{i+1} calculado como:

1. Calcular la descomposición QR de $R_i^{\,T} = Q_{0i} R_{0i}$
2. Calcular la descomposición QR de $R_{0i}^{\,T} = Q_{1i} R_{i+1}$

Algoritmo SVD "Raíz Cuadrada"

1. Calcular la descomposición QR de $A = Q_0 R_0$

Para $i = 0,1,2,\ldots$

2.1 Trasponer (R_i, X) (*X = $R_{0i}{}^T$ *)

2.2 Calcular la descomposición QR de $X = Q_i R_{i+1}$

2.3 Trasponer (R_{i+1}, X) (*X = $R_{i+1}{}^T$ *)

2.4 Calcular la descomposición QR de $X = Q_{i+1} R_{i+1}$

Extensiones de la SVD: La Descomposición URV

Proposición:

Supongamos que rank(A)=k en el sentido de que

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k > \sigma_{k+1} \geq \ldots \geq \sigma_n$$

Existen matrices ortogonales U, V tales que

$$A = U \begin{bmatrix} R & F \\ 0 & G \end{bmatrix} V^T$$

con R, G triangulares superiores, $R \in \Re^{k \times k}, U \in \Re^{m \times m}, V \in \Re^{n \times n}$

$$\sigma_{\min}(R) = \sigma_k$$

$$\sqrt{\|F\|_F^2 + \|G\|_F^2} \approx \sqrt{\sigma_{k+1}^2 + \ldots + \sigma_n^2}$$

Proposición:

Las transformaciones ortogonales conservan la Descomposición URV

Ventajas frente a la SVD

Coste computacional más pequeño que la SVD

Permite conocer el rango de A y los subespacios asociados con V

Es más fácil de actualizar (updating) que la SVD

Cálculo de la URV

$$\begin{bmatrix} a & a & a & a \\ & a & a & a \\ & & a & a \\ & & & a \end{bmatrix} Q, \quad Q \text{ ortogonal}$$

$$\begin{bmatrix} \bar{a} & \bar{a} & \bar{a} & 0^3 \\ & \bar{a} & \bar{a} & 0^2 \\ & & \bar{a} & 0^1 \\ \bar{a} & \bar{a} & \bar{a} & \bar{a} \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} \hat{a} & & & \\ \hat{a} & \hat{a} & & \\ \hat{a} & \hat{a} & \hat{a} & \\ \hat{a} & \hat{a} & \hat{a} & \hat{a} \end{bmatrix} \begin{matrix} \longleftarrow \left\| \begin{bmatrix} a & a & a & a \end{bmatrix} \right\|_2 \\ \\ \\ \longleftarrow \left\| \begin{bmatrix} 0 & 0 & 0 & a \end{bmatrix} \right\|_2 \end{matrix}$$

Cálculo de la URV

$$\hat{Q}^T \begin{bmatrix} \hat{a} & 0 & 0 & 0 \\ \hat{a} & \hat{a} & 0 & 0 \\ \hat{a} & \hat{a} & \hat{a} & 0 \\ \hat{a} & \hat{a} & \hat{a} & \hat{a} \end{bmatrix}, \quad \hat{Q} \text{ ortogonal}$$

$$\downarrow$$

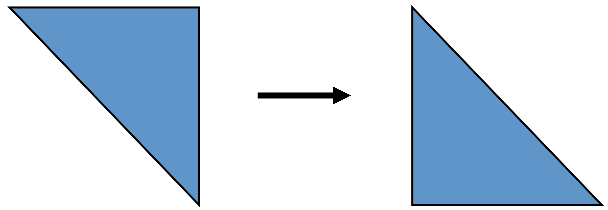
$$\begin{bmatrix} \tilde{\boldsymbol{a}} & \tilde{a} & \tilde{a} & \tilde{a} \\ 0 & \tilde{\boldsymbol{a}} & \tilde{a} & \tilde{a} \\ 0 & 0 & \tilde{\boldsymbol{a}} & \tilde{a} \\ 0 & 0 & 0 & \tilde{\boldsymbol{a}} \end{bmatrix}$$

$$\left\| \begin{bmatrix} \hat{a} \\ \hat{a} \\ \hat{a} \\ \hat{a} \end{bmatrix} \right\|_2 \qquad \left\| \begin{bmatrix} 0 \\ 0 \\ 0 \\ \hat{a} \end{bmatrix} \right\|_2$$

La masa de la matriz tiende a concentrarse en la diagonal

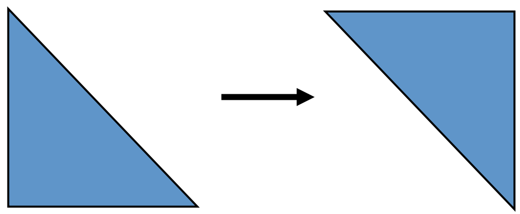
Cálculo de la URV

$$R_0 Q = \hat{R}_0^T$$

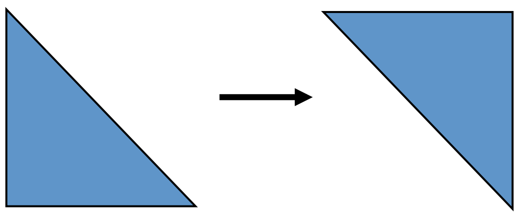


Equivalente a

$$Q^T R_0^T = \hat{R}_0$$



$$R_1 = \hat{Q}^T \hat{R}_0^T$$



La masa de la matriz tiende a concentrarse en la diagonal

Algoritmo para el cálculo de la URV

1. Calcular la descomposición QR de $A = Q_0 R_0$

Para $i = 0,1,2,\ldots$

2.1 Trasponer (R_i, X) (*X = R_i^T *)

2.2 Calcular la descomposición QR de $X = Q_i R_{i+1}$

MATRIZ INICIAL

2.0000000000	3.0000000000	4.0000000000	5.0000000000
3.0000000000	4.0000000000	5.0000000000	6.0000000000
4.0000000000	5.0000000000	6.0000000000	7.0000000000
5.0000000000	6.0000000000	7.0000000000	8.0000000000

-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*

Numero de iteraciones: 1

-2.0954237474E1	9.4531899334E-2	1.8834346660E-10	2.3187629717E-10
2.9103830457E-11	9.5446088288E-1	-1.7740005414E-11	-2.1137647837E-11
-2.6469779602E-22	0.0000000000	-9.1500898891E-12	5.1813526433E-13
4.5494933690E-24	0.0000000000	7.2791893905E-23	7.4855080673E-13

-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*

-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*

Numero de iteraciones: 2

-2.0954451148E1	1.9612712235E-4	-2.5410987881E-21	-3.1763734851E-21
5.6843418861E-14	9.5445115015E-1	5.2216228621E-25	6.5270285965E-25
2.3111159333E-33	0.0000000000	-9.1648458849E-12	3.4509990292E-15
4.8148248610E-35	0.0000000000	5.1698788285E-25	7.4734559140E-13

-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*

-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*

Numero de iteraciones: 3

-2.0954451149E1	4.0690448972E-7	3.3896367021E-32	4.3140830754E-32
1.2490009027E-16	9.5445115011E-1	-1.4450801400E-38	-1.8391929055E-38
0.0000000000	0.0000000000	-9.1648465387E-12	2.2947604916E-17
0.0000000000	0.0000000000	4.0389678347E-27	7.4734553807E-13

-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*

-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*

Numero de iteraciones: 4

-2.0954451148E1	8.4420380826E-10	0.0000000000	0.0000000000
2.4394548881E-19	9.5445115011E-1	0.0000000000	0.0000000000
0.0000000000	0.0000000000	-9.1648465385E-12	1.5259133045E-19
0.0000000000	0.0000000000	2.5243548967E-29	7.4734553807E-13

-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*

-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*

Numero de iteraciones: 5

-2.0954451147E1	1.7514676995E-12	0.0000000000	0.0000000000
5.2939559203E-22	9.5445115011E-1	0.0000000000	0.0000000000
0.0000000000	0.0000000000	-9.1648465383E-12	1.0146642496E-21
0.0000000000	0.0000000000	1.6023737137E-31	7.4734553807E-13

-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*

```
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
Numero de iteraciones: 6
   -2.0954451147E1    3.6337660085E-15      0.0000000000       0.0000000000
  1.1373733423E-24     9.5445115011E-1      0.0000000000       0.0000000000
      0.0000000000        0.0000000000  -9.1648465381E-12   6.7470644367E-24
      0.0000000000        0.0000000000   1.1555579666E-33   7.4734553807E-13
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
Numero de iteraciones: 7
   -2.0954451146E1    7.5389659769E-18      0.0000000000       0.0000000000
  2.2214323091E-27     9.5445115011E-1      0.0000000000       0.0000000000
      0.0000000000        0.0000000000  -9.1648465379E-12   4.4864967457E-26
      0.0000000000        0.0000000000   6.7708474607E-36   7.4734553807E-13
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
Numero de iteraciones: 8
   -2.0954451146E1    1.5641075379E-20      0.0000000000       0.0000000000
  4.7331654313E-30     9.5445115011E-1      0.0000000000       0.0000000000
      0.0000000000        0.0000000000  -9.1648465376E-12   6.5827683646E-37
      0.0000000000        0.0000000000   4.7019774033E-38   7.4734553807E-13
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
Numero de iteraciones: 9
   -2.0954451145E1    3.2450503112E-23      0.0000000000       0.0000000000
  1.0014835711E-32     9.5445115011E-1      0.0000000000       0.0000000000
      0.0000000000        0.0000000000  -9.1648465374E-12       0.0000000000
      0.0000000000        0.0000000000       0.0000000000   7.4734553807E-13
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
Numero de iteraciones: 10
   -2.0954451144E1    6.7324984187E-26      0.0000000000       0.0000000000
  1.9560225998E-35     9.5445115011E-1      0.0000000000       0.0000000000
      0.0000000000        0.0000000000  -9.1648465372E-12       0.0000000000
      0.0000000000        0.0000000000       0.0000000000   7.4734553807E-13
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
```

```
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
Numero de iteraciones: 11
   -2.0954451144E1     1.3967898990E-28       0.0000000000        0.0000000000
   4.1142302279E-38     9.5445115011E-1       0.0000000000        0.0000000000
      0.0000000000         0.0000000000    -9.1648465370E-12      0.0000000000
      0.0000000000         0.0000000000       0.0000000000     7.4734553807E-13
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
Numero de iteraciones: 12
   -2.0954451143E1     2.8979168115E-31       0.0000000000        0.0000000000
      0.0000000000      9.5445115011E-1       0.0000000000        0.0000000000
      0.0000000000         0.0000000000    -9.1648465368E-12      0.0000000000
      0.0000000000         0.0000000000       0.0000000000     7.4734553807E-13
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
Numero de iteraciones: 13
   -2.0954451143E1     6.0123014255E-34       0.0000000000        0.0000000000
      0.0000000000      9.5445115011E-1       0.0000000000        0.0000000000
      0.0000000000         0.0000000000    -9.1648465366E-12      0.0000000000
      0.0000000000         0.0000000000       0.0000000000     7.4734553807E-13
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
Numero de iteraciones: 14
   -2.0954451142E1     1.2473708110E-36       0.0000000000        0.0000000000
      0.0000000000      9.5445115011E-1       0.0000000000        0.0000000000
      0.0000000000         0.0000000000    -9.1648465364E-12      0.0000000000
      0.0000000000         0.0000000000       0.0000000000     7.4734553807E-13
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
Numero de iteraciones: 15
   -2.0954451142E1         0.0000000000       0.0000000000        0.0000000000
      0.0000000000      9.5445115011E-1       0.0000000000        0.0000000000
      0.0000000000         0.0000000000    -9.1648465362E-12      0.0000000000
      0.0000000000         0.0000000000       0.0000000000     7.4734553807E-13
-*-*-*-*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*
   -2.0954451142E1         0.0000000000       0.0000000000        0.0000000000
      0.0000000000      9.5445115011E-1       0.0000000000        0.0000000000
      0.0000000000         0.0000000000    -9.1648465362E-12      0.0000000000
      0.0000000000         0.0000000000       0.0000000000     7.4734553807E-13
```

Cálculo de la SVD en el LAPACK y ScaLAPACK

http://www.netlib.org/lapack/double/

http://www.netlib.org/scalapack/double/

Ejercicios propuestos

1. Escribe una función Matlab que haga ceros en las componentes de un vector utilizando transformaciones ortogonales: [y,Q]=Anula(x), con $y=Qx=ke_1, Q^T=Q^{-1}$
2. Escribe una función Matlab que bidiagonalice una matriz utilizando transformaciones ortogonales: B=Bidiag(A), con $B=PAQ^T$, $P^T=P^{-1}, Q^T=Q^{-1}$
3. Escribe una función Matlab que devuelva los valores singulares de A, s=ValSin(A).
4. Escribe una función Matlab que tome como entrada la diagonal y superdiagonal de una matriz bidiagonal superior B y devuelva los valores singulares de B, s=ValBid(d,e).