Actividades 6

Prácticas de Cálculo Numérico I (doble grado)

20 de abril de 2021

1. SVD y optimización

Recuerda de la teoría que la descomposición en valores singulares (SVD por *Singular Value Decomposition*) de una matriz A es la factorización $A = UDV^\dagger$ donde U y V son matrices unitarias (su inversa coincide con su traspuesta conjugada, indicada con $\dagger$) y D es una matriz cuyos únicos elementos no nulos son $d_{ii} \geq 0$, estos son los llamados *valores singulares* que dan nombre a la descomposición. En `matlab/octave`, el comando `svd`, utilizado como en el siguiente ejemplo, permite obtener la SVD.

```
A = rand(10,7);
[U,D,V] = svd(A);
norm(A-U*D*V','fro')
```

Como es evidente por la primera línea, la descomposición no requiere que la matriz sea cuadrada. La última línea es la comprobación de que el error es pequeño. Se ha escogido la norma de Frobenius solo para recordar el comando.

Si se llama a `svd` sin recuperar la salida con tres matrices, se obtiene solo el vector de valores singulares. Así las líneas 4 y 7 del siguiente código muestran el mismo resultado.

```
1 A = rand(3,5);
2 [U,D,V] = svd(A);
3 disp('Completa')
4 disp( diag(D) )
5 disp('Solo valores singulares')
6 v = svd(A);
7 disp(v)
```

Parte del interés de la SVD es que permite resolver algunos problemas de optimización. En particular, en la teoría has visto que la solución de mínimos cuadrados de un sistema incompatible $A\vec{x} = \vec{b}$ viene dada por

$$\vec{x} = \sum_{k=1}^{r} \sigma_k^{-1} \langle \vec{b}, U^{(k)} \rangle V^{(k)}$$

donde r es el rango, el superíndice (k) indica la columna k-ésima y σ_k son los valores principales. El producto escalar es el usual, definido en el caso complejo con el convenio del álgebra lineal. Si $A \in \mathcal{M}_{n\times m}(\mathbb{C})$ con $r = m < n$, la situación típica en la que nos habíamos fijado, esto es lo mismo que $VE(U_m)^\dagger \vec{b}$ donde E es la matriz diagonal $m \times m$ que tiene $e_{ii} = \sigma_i^{-1}$ y U_m indica que nos quedamos solo con las primeras m columnas. Veamos en un ejemplo que se obtiene lo mismo que ya obtuvimos con QR o con la barra de `matlab/octave`.

```
A = [1,2; 1,1; 3,1];
b = [2;8;6];
[n,m] = size(A);
[U,D,V] = svd(A);
E = diag( diag(D) );
V*inv(E)*U(:,1:m)'*b
A\b
```

Las dos últimas líneas producen una salida idéntica, $(2,1)^t$. Invertir `E` no le cuesta esfuerzo a `matlab/octave` porque sabe que es una matriz diagonal. Si trabajáramos con un lenguaje de programación sin esta estructura de datos, lo eficiente será multiplicar las columnas de V con los σ_k^{-1}.

Actividad 6.1.1. *Comprueba si lo anterior da el mismo resultado que* `A\b` *también cuando trabajamos con números complejos. Para ello, parte de una matriz* `A` *aleatoria compleja* `2N` $\times$ `N` *y de un vector* `b` *de dimensión* `N` *y estudia si ambos resultados tienen una diferencia despreciable.*

Otra aplicación de la SVD relacionada con la optimización es el *teorema de Eckart-Young* (a veces se añade el nombre de Mirsky) que afirma que si tenemos una matriz A, la mejor aproximación A_r de rango $r \le \mathrm{rg}(A)$, en el sentido de que la norma de Frobenius de $A - A_r$ es mínima, se obtiene mediante $A_r = \sum_{k=1}^{r} \sigma_k U^{(k)} (V^{(k)})^\dagger$. Esto es lo mismo que anular en la SVD de A todos los valores singulares de índice mayor que r. Normalmente el resultado se enuncia para matrices reales pero el caso complejo es similar y está esencialmente hecho en [Mir60].

El siguiente código muestra un ejemplo:

```
A = [1,2,3; 1,1,2; 3,1,4; 3,1,4];

r = 2;
[U,D,V] = svd(A);
for k=r+1:min(size(A))
```

```
6         D(k,k)=0;
7 end
8
9 Ar = U*D*V';
10
11 disp(Ar)
12 disp(norm(A-Ar,'fro'))
```

Como en este ejemplo la matriz de partida tiene rango 2, se obtiene `Ar` igual a `A`. Si cambiamos la tercera línea por `r = 1`, obtendremos una matriz con todas sus filas proporcionales.

Actividad 6.1.2. *Para una matriz real aleatoria de tamaño* 100×200 *fijada, dibuja una gráfica que muestre cómo cambia la norma de Frobenius de* $A - A_r$ *cuando* $1 \leq r < 100$.

Un tema relacionado es que, en este mismo sentido, si $A = UDV^\dagger$ es la SVD de una matriz cuadrada A, entonces $UV^\dagger$ es la matriz unitaria que mejor aproxima a A, de nuevo en el sentido de la norma de Frobenius. El siguiente código ejemplifica esto gráficamente. Se dibuja la circunferencia unidad centrada en $(2, 0)$, se le aplica una transformación lineal que la convierte en una elipse y $UV^\dagger$ es un giro que superpone la circunferencia y la elipse.

```
A = [0.546, -0.722; 0.783, 0.427];

t = linspace(0,2*pi,300);
x = [2+cos(t); sin(t)];
y = A*x;

[U,D,V] = svd(A);

z = U*V'*x;

figure(1)
plot(x(1,:), x(2,:))
hold on
plot(y(1,:), y(2,:))
hold on
plot(z(1,:), z(2,:))
hold off
axis('equal')
```

Actividad 6.1.3. *Modifica el código anterior para que la figura de partida sea el cuadrado definido por la frontera de* $[2, 3] \times [0, 1]$.

En la teoría no has visto ningún algoritmo eficiente para calcular la SVD que podamos implementar (si tienes curiosidad, mira [SB93, §6.7]). Sin embargo, sí te han contado un método para hallarla "a mano" a partir de la diagonalización de $A^\dagger A$, aunque no es muy eficiente desde el punto de vista numérico. En resumen, la diagonalización da V y U es una matriz unitaria que cumple $AV = UD$.

ACTIVIDAD 6.1.4. *Implementa en* `matlab/octave` *el algoritmo anterior usando que* `[V,D] = eig(S)` *da la diagonalización* $S = VDV^\dagger$ *con* V *unitaria para una matriz hermítica* S*. Halla* U *y* D *con* `qr` *y trata de justificar por qué sabemos que la* D *hallada de esta forma debe tener ceros fuera* d_{ii}*, lo cual va más allá de ser triangular (en rigor, trapezoidal) superior.*

Referencias

[Mir60] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *Quart. J. Math. Oxford Ser. (2)*, 11:50–59, 1960.

[SB93] J. Stoer and R. Bulirsch. *Introduction to numerical analysis*, volume 12 of *Texts in Applied Mathematics*. Springer-Verlag, New York, second edition, 1993. Translated from the German by R. Bartels, W. Gautschi and C. Witzgall.