6.3. Uso de la SVD para determinar la estructura de una matriz

Primero definiremos algunas características de matrices.

Rango de una matriz:
Sea A una matriz m x n se entonces su rango es definido como el número de columnas(o filas) linealmente independientes que tiene dicha matriz. Una matriz se dice que es de rango completo por columnas si todas sus columnas son linealmente independientes. Una matriz se dice que es de rango completo por filas si todas sus filas son linealmente independientes.. Una matriz es de rango completo si es de rango completo por filas o por columnas.
Propiedades del rango de una matriz

a) rango(A)=rango(A')
b) rango(AB)≥rango(A) + rango(B)-n donde B es n x p.
c) rango(AB)≤min(rango(A),rango(B))
d) rango(A+B)≤rango(A)+rango(B)
e) Si A es una matriz cuadrada y de rango completo entonces su inversa existe

Matlab tiene una función **rank** que estima el rango de una matriz (ver más adelante, lo referente a rango numérico.

Normas de vectores y matrices

Sea $x=(x_1,x_2,\ldots x_n)$ un vector n dimensional entonces se pueden definir las siguientes normas vectoriales

a) $||x||_1=|x_1|+|x_2|+\ldots\ldots|x_n|$
b) $||x||_2=\sqrt{x_1^2+x_2^2+\ldots.+x_2^n}$ (norma Euclideana)
c) $||x||_\infty=\max_i|x_i|$

Sea A una matriz m x n entonces se pueden definir las siguientes normas

a) $||A||_1=\max_{1\le j\le n}\sum_{i=1}^{m}|a_{ij}|$, es la mayor de la suma de los valores absolutos de las columnas

b) $||A||_\infty=\max_{1\le i\le m}\sum_{j=1}^{n}|a_{ij}|$, es la mayor de la suma de los valores absolutos de las filas

c) $||A||_2=\max_{x\neq 0}\frac{||Ax||_2}{||x||_2}$, llamada la norma espectral. Se puede demostrar que $||A||_2=\sqrt{\max \quad eigenvalue \quad AA'}$

d) $||A||_F=\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij})^2}$, llamada la norma de Frobenius. Se puede mostrar que $||A||_F^2$ =traza(AA').

La función **norm** de MATLAB calcula la norma de una matriz.
En R hay una librería Matriz, donde hay una funcion norm que calcula la norma de una matriz.

Número Condición:
Un problema se dice que está mal condicionado si pequeña perturbaciones en los datos causa un gran aumento en el error relativo de la solución. El número condición indica si el problema está mal condicionado o no. Para un sistema de ecuaciones Ax=b, el número condición es $||A||||A^{-1}||$. Si el número condición es grande entonces, digamos mayor que 100, hay mal condicionamiento. Frecuentemente, lo que se hace es estimar el valor del número condición y no hay necesidad de calcular la inversa explicitamente.

Ejemplo Consideremos el sistema Hx=b, donde H es la matriz de Hilbert de orden 5 (H[i,j]=1/(i+j-1)) y b=(2.2833,1.4500,1.0929,0.8845,0.7456) cuya solución exacta es (1,1,1,1,1)'. Si la entrada (5,1) que es .2 se perturba a .20001 entonces la solución cambia a

(0.9937,1.1252,0.4365,1.8765,0.5618)'.

En R,
```
> library(Matrix)
> Hilbert(5)
5 x 5 Matrix of class "dpoMatrix"
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.5000000 0.3333333 0.2500000 0.2000000
[2,] 0.5000000 0.3333333 0.2500000 0.2000000 0.1666667
[3,] 0.3333333 0.2500000 0.2000000 0.1666667 0.1428571
[4,] 0.2500000 0.2000000 0.1666667 0.1428571 0.1250000
[5,] 0.2000000 0.1666667 0.1428571 0.1250000 0.1111111
> # Norma-1
>norm(Hilbert(5))
[1] 2.283333
># Norma Frobenius
> norm(Hilbert(5),type="F")
[1] 1.580906
> norm(Hilbert(5),type="M")
[1] 1
> norm(Hilbert(5),type="I")
[1] 2.283333
> rcond(Hilbert(5))
[1] 1.059708e-06
> # numero condicion
```

```
>1/rcond(Hilbert(5))
[1] 943656
```

En Matlab,
la función hilb(n) define una matriz de Hilbert de orden 5.
La funcion cond(A) da el número condición de una matriz

```
» hilb(5)

ans =

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

» x=hilb(5)*ones(5,1)

x =

    2.2833
    1.4500
    1.0929
    0.8845
    0.7456
```

```
» A=hilb(5)
» A(5,1)=.20001

A =

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

» inv(A)*x

ans =

    0.9937
    1.1252
    0.4365
```

1.8765
0.5618

```
» cond(hilb(5))

ans =

  4.7661e+005

»
```

La SVD puede ser usada para calcular el rango de una matriz, algunas de sus normas y el número condición. Todas ellas de alguna manera definen la estructura de una matriz. Así, sea σ_i el i-ésimo valor singular de A

1. $||A||_2=\sigma_{max}$. En efecto, $||A||_2=||USV'||_2=||S||_2=\sigma_{max}$
2. $||A||_F=\sqrt{\sigma_1^2+\sigma_2^2+......+\sigma_n^2}$
3. $||A^{-1}||_2=1/\sigma_{min}$.
4. $Cond_2(A)=||A||_2||A^{-1}||_2=\sigma_{max}/\sigma_{min}$.

Rango Numérico de una matriz

El número de valores singulares distintos de ceros determina el rango de una matriz. El problema es cuando aceptar que un valor singular calculado con computadoras sea cero. Una regla práctica es la siguiente:
Aceptar un valor singular como cero si es menor que $\delta=10^{-t}||A||_\infty$, asumiendo son correctas hasta t digitos.
Luego el rango numérico de una matriz se determina contando el número de valores singulares que son mayores que δ

6.4. Calculando la SVD

Como se ha visto anteriormente los valores singulares son las raices positivas de los valores propios de la matriz simétrica A'A. Sin embargo, ésta no es la forma más eficiente de calcular la SVD debido a que pueden haber errores de redondeo en el cálculo de A'A.
Ejemplo. Sea la matriz

$$A=\begin{pmatrix} 1.0001 & 1.0000 \\ 1.0000 & 1.0001 \end{pmatrix}$$

Los valores singulares de A son 2.00010 y 0.0001
Por otro lado

$$A'A=\begin{pmatrix} 2.0002 & 2.0002 \\ 2.0002 & 2.0002 \end{pmatrix}$$

Los valores propios de A'A son 0 y 4.0004, por lo tanto sus raices cuadradas no darian Los valores singulares.

Existen varios algoritmos para calcular la SVD de una matriz pero la mas usada es el Algoritno de Golub, Kahan y Reinsh (1970). El algoritmo consiste de dos etapas.
Etapa 1. La matriz A de orden m x n es transformada en una matriz bidiagonal superior B usando transformaciones ortogonales, usualmente se usa 2n-2 matrices Householder. Más especificamente,

$$U'AV=\begin{pmatrix} B \\ 0 \end{pmatrix}$$

Donde B es una matriz bidiagonal superior de la forma

$$B=\begin{pmatrix} * & * & 0 & ... & 0 \\ 0 & * & * & 0 & 0 \\ 0 & 0 & * & * & 0 \\ ... & ... & .. & * & * \\ 0 & 0 & 0 & 0 & * \end{pmatrix}$$

U consiste del producto de matrices Householder y V consiste del producto de n-2 Householder.
Etapa 2. La matriz Bidiagonal B es reducida a la matriz diagonal S usando un algoritmo iterativo implícito basado en QR con matrices Givens. Al final se obtendrá

$$U_1'BV_1 = S$$

donde S es la matriz que contiene los valores singulares de A.
En cada paso del proceso iterativo se obtiene una matriz bidiagonal que tiene valores en la diagonal que son menores que la del paso anterior y cada vez se aproximan a cero. Esto se obtiene aplicando QR a la matriz tridiagonal y simétrica B'B y donde para acelerar la convergencia se aplica una factor de traslación a cada elemento de B'B. Este factor de corrección es llamado "Wilkinson Shift" y es igual al valor propio de la submatriz 2 x 2 localizada en la parte inferior de B'B que resulta en cada paso.