Linear Algebra

Vector Spaces, Linear Transformations and Innver Product Spaces.

Definitions, theorems and exercises from the fourth edition of the book *Linear Algebra Done Right* by Sheldon Axler.

§1 Subespacios invariantes

Definition 1.1 (Subespacio invariante)

Suponiendo $T \in \mathcal{L}(V)$. Un subespacio U de V es llamado **invariante** bajo T si $u \in U$ implica $Tu \in U$.

En la búsqueda del subespacio no trivial más simple posible (1-dimensional) nos encontramos con un U definido como

$$U = \{\lambda v : \lambda \in \mathbb{F}\} = \text{span}(v)$$

Vemos que si U es invariante bajo un operador $T \in \mathcal{L}(V)$ entonces $Tv \in U$ y por tanto hay un escalar $\lambda \in \mathbb{F}$ que cumple

$$Tv = \lambda v$$

Esta ecuación es tan importante que el vector v y el valor λ reciben su propio nombre.

§2 Vectores y valores propios

Definition 2.1 (Valor Propio o Eigenvalue)

Suponiendo $T \in \mathcal{L}(V)$. Un número $\lambda \in \mathbb{F}$ es llamado **valor propio** de T si existe $v \in V$ tal que $v \neq 0$ y $Tv = \lambda v$.

Es condición indispensable que $v \neq 0$ porque cualquier escalar $\lambda \in \mathbb{F}$ cumple $T0 = \lambda 0$.

Definition 2.2 (Vector Propio o Eigenvector)

Suponiendo $T \in \mathcal{L}(V)$ y $\lambda \in \mathbb{F}$ es un valor propio de T. Un vector $v \in V$ es llamado **vector propio** de T correspondiente a λ si $v \neq 0$ y $Tv = \lambda v$.

Teorema 2.3 (Una lista de vectores propios es linealmente independiente)

Sea $T \in \mathcal{L}(V)$. Supón $\lambda_1, \ldots, \lambda_m$ son distintos valores propios de T y $v_1, \ldots, v_m$ son los correspondientes vectores propios. Entonces $v_1, \ldots, v_m$ es linealmente independiente.

Proof. Suponeos que $v_1, \ldots, v_m$ es linealmente dependiente. Siendo k el entero positivo más pequeño tal que

$$v_k \in span(v_1, \ldots, v_{k-1}); \tag{5.11}$$

la existencia de k con esta propiedad se sigue del *Lema de Dependencia Lineal* (2.21). Por tanto existe $a_1, \ldots, a_{k-1} \in \mathbb{F}$ tal que

$$v_k = a_1 v_1 + \cdots + a_{k-1} v_{k-1}. \tag{5.12}$$

Applicando T a ambos lados de la ecuación obtenemos

$$\lambda_k v_k = a_1 \lambda_1 v_1 + \cdots + a_{k-1} \lambda_{k-1} v_{k-1}.$$

Multiplicando ambos lados de 5.12 por λ_k y luego restando la ecuación de arriba obtenemos

$$0 = a_1(\lambda_k - \lambda_1) v_1 + \cdots + a_{k-1}(\lambda_k - \lambda_{k-1}) v_{k-1}.$$

Dado que definimos k como el menor entero positivo que satisface 5.11, $v_1, \ldots, v_{k-1}$ es linealmente independiente. Por tanto la ecuación de arriba implica que todas las a's son 0. Sin embargo, esto significa que v_k es igual a 0, contradiciendo nuestra hipotesis de que v_k es un vector propio. Por tanto nuestra asunción de que $v_1, \ldots, v_m$ es linealmente dependiente es falsa. □

Teorema 2.4 (máximo de valores propios)

Suponiendo V finito-dimensional. Cada operador en V tiene como mucho $\dim V$ valores propios distintos.

§2.1 Definiciones clave para el calculo de valores propios

Definition 2.5

Las siguientes afirmaciones para un operador $T \in \mathcal{L}(V)$, con V de dimensión finita, y un escalar $\lambda \in \mathbb{F}$ son equivalentes:

(a) λ es un valor propio de T;

(b) $T - \lambda I$ no es inyectivo;

(c) $T - \lambda I$ no es sobreyectivo;

(d) $T - \lambda I$ no es invertible.

Teorema 2.6 (Teorema multiplos de 3)

Para todo $n \in \mathbb{Z}$ se cumple que al menos uno de los factores de la expresión $n(n+1)(n+2)$ es divisible por 3.

Proof. Vamos a completar la prueba por inducción, es fácil ver que el teorema se cumple para el caso $n = 1$, $1 \cdot 2 \cdot 3 = 3 \cdot (2)$. Ahora suponiendo que se cumple para n demostraremos que lo hace también para $n+1$. Con la expresión

$$(n+1)(n+2)(n+3) = 3k$$

Para cierto $k \in \mathbb{Z}$, desarrollando la expresión obtenemos

$$\begin{aligned}(n+1)(n+2)(n+3) &= \frac{3k}{n}(n+3) \\ &= 3 \cdot \frac{k}{n}(n+3)\end{aligned}$$

Donde la primera igualdad se sostiene de la supocisión inductiva. Vemos que si $\frac{k(n+3)}{n}$ es un entero entonces hemos terminado la prueba y sabemos que es un entero ya que de la suposición inductiva sabemos que

$$\begin{aligned}\frac{k(n+3)}{n} &= \frac{kn+3k}{n} \\ &= \frac{kn+n(n+1)(n+2)}{n} \\ &= k+(n+1)(n+2)\end{aligned}$$

Por tanto $\frac{k(n+3)}{n}$ es un entero completando la prueba □

§3 Singular Value Decomposition (SVD)

Definition 3.1 (SVD)

Suppose $T \in \mathcal{L}(V,W)$ and the positive singular values of T are $s_1, \ldots, s_m$. Then there exist orthonormal lists $e_1, \ldots, e_m$ in V and $f_1, \ldots, f_m$ in W such that

$$Tv = s_1\langle v, e_1\rangle f_1 + \cdots + s_m\langle v, e_m\rangle f_m \tag{3.11}$$

for every $v \in V$.

Proof. Let $s_1, \ldots, s_m$ denote the singular values of T (thus $n = dimV$). Because T^*T is a positive operator, the spectral theorem implies that there exists an orthonormal basis $e_1, \ldots, e_n$ of V with

$$T^*Te_k = s_k^2 e_k \tag{3.12}$$

for each $k = 1, \ldots, n$.

For each $k = 1, \ldots, m$, let

$$f_k = \frac{Te_k}{s_k}. \tag{3.13}$$

If $j, k \in 1, \ldots, m$, then

$$\langle f_j, f_k\rangle = \frac{1}{s_j s_k}\langle Te_j, Te_k\rangle = \frac{1}{s_j s_k}\langle e_j, T^*Te_k\rangle = \frac{s_k}{s_j}\langle e_j, e_k\rangle = \begin{cases} 0 & if\ j \neq k, \\ 1 & if\ j = k. \end{cases}$$

Thus $f_1, \ldots, f_m$ is an orthonormal list in W.

If $k \in 1, \ldots, n$ and $k > m$, then $s_k = 0$ and hence $T^*Te_k = 0$ (by 3.12), which implies that $Te_k = 0$.

Suppose $v \in V$. Then

$$\begin{aligned} Tv &= T\left(\langle v, e_1\rangle e_1 + \cdots + \langle v, e_n\rangle e_n\right) \\ &= \langle v, e_1\rangle Te_1 + \cdots + \langle v, e_m\rangle Te_m \\ &= s_1\langle v, e_1\rangle f_1 + \cdots + s_m\langle v, e_m\rangle f_m, \end{aligned}$$

where the last index in the first line switched from n ot m in the secod line because $Te_k = 0$ if $k > m$ (as noted in the paragraph above) and the third line follows from 3.13. The equation above is our desired result. □

With the tool presented above we can arrive to a very useful concept in copression theory and computation, wich is the appoximation by linear maps with lower-dimensional range.

Definition 3.2 (best approximation by linear map whose range has dimension $\leq k$)
Suppose $T \in \mathcal{L}(V, W)$ and $s_1 \geq \cdots \geq s_m$ are the positive singular values of T.

Suppose $1 \leq k < m$. Then

$$min\{\|T - S\| : S \in \mathcal{L}(V, W) \text{ and dim range } S \leq k\} = s_{k+1}.$$

Furthermore, if

$$Tv = s_1\langle v, e_1\rangle f_1 + \cdots + s_m\langle v, e_m\rangle f_m$$

is a singular value decomposition of T and $T_k \in \mathcal{L}(V, W)$ is defined by

$$T_k v = s_1\langle v, e_1\rangle f_1 + \cdots + s_k\langle v, e_k\rangle f_k$$

for each $v \in V$, then dim range $T_k = k$ and $\|T - T_k\| = s_{k+1}$.

Proof. If $v \in V$ then

$$\begin{aligned} \|(T - T_k)v\|^2 &= \|s_{k+1}\langle v, e_{k+1}\rangle f_{k+1} + \cdots + s_m\langle v, e_m\rangle f_m\|^2 \\ &= s_{k+1}^2|\langle v, e_{k+1}\rangle|^2 + \cdots + s_m^2|\langle v, e_m\rangle|^2 \\ &\leq s_{k+1}^2\left(|\langle v, e_{k+1}\rangle|^2 + \cdots + |\langle v, e_m\rangle|^2\right) \\ &\leq s_{k+1}^2\|v\|^2. \end{aligned}$$

Thus $\|T - T_k\| \leq s_{k+1}$. The equation $(T - T_k)e_{k+1} = s_{k+1}f_{k+1}$ now shows that $\|T - T_k\| \leq s_{k+1}$.

Suppose $S \in \mathcal{L}(V, W)$ and dim range $S \leq k$. Thus $Se_1, \ldots, Se_{k+1}$, which is a list of length $k+1$, is linearly dependent. Hence there exist $a_1, \ldots, a_{k+1} \in \mathbb{F}$, not all 0, shut that

$$a_1 Se_1 + \cdots + a_{k+1} Se_{k+1} = 0.$$

Now $a_1 Se_1 + \cdots + a_{k+1} Se_{k+1} \neq 0$ because $a_1, \ldots, a_{k+1}$ are not 0. We have

$$\begin{aligned} \|(T-S)(a_1 e_1 + \cdots + a_{k+1} e_{k+1})\|^2 &= \|T(a_1 e_1 + \cdots + a_{k+1} e_{k+1})\|^2 \\ &= \|s_1 a_1 f_1 + \cdots + s_{k+1} a_{k+1} f_{k+1}\|^2 \\ &= s_1^2 |a_1|^2 + \cdots + s_{k+1}^2 |a_{k+1}|^2 \\ &\geq s_{k+1}^2 (|a_1|^2 + \cdots + |a_{k+1}|^2) \\ &= s_{k+1}^2 \|a_1 e_1 + \cdots + a_{k+1} e_{k+1}\|^2. \end{aligned}$$

Because $a_1 e_1 + \cdots + a_{k+1} e_{k+1} \neq 0$, the inequality above implies that

$$\|T - S\| \geq s_{k+1}.$$

Thus $S = T_k$ minimizes $\|T - S\|$ among $S \in \mathcal{L}(V, W)$ with dim range $S \leq k$. $\square$

Problem 3.3. Fix $u, x \in V$ with $u \neq 0$. Define $T \in \mathcal{L}(V)$ by $Tv = \langle v, u \rangle x$ for every $v \in V$.

Prove that

$$\sqrt{T^* T} v = \frac{\|x\|}{\|u\|} \langle v, u \rangle u$$

for every $v \in V$.

§4 QR Decomposition and Householder reflections

Definition 4.1 (Householder operator)

The Householder operator H in a inner product space V represents a reflectionof a given vector x over a plane perpendicular to a unitary vector u, wich means $\|u\| = 1$. so H applied to x has de form:

$$H(x) = x - 2\text{proj}_u(x)$$

Knowing the norm of u is 1 and that u^* represents the conjugate transpose which in the real field $\mathbb{R}$ equals the transpose u^T we end up with

$$H(x) = x - 2u^*xu = (I - 2u^*u)x$$

With this we see that the Householder matrix H representing a reflection over a normal vectoro u has the form

$$H = I - 2uu^*$$

Definition 4.2 (QR Decomposition)

Given a matrix A with m rows and n columns. Then there exists a unique way of decomposing A in a combination of a unitary matrix named Q and an upper triangular matrix R, so A can be expressed in the form

$$A = QR$$

Proof. Suppose A is formed with the list $v_1, \ldots, v_n$. By the Gram-Schmidt prodedure we can write any vector v of the vector space V in the form

$$v = \langle v_1, e_1\rangle e_1 + \cdots + \langle v_n, e_n\rangle e_n$$

where $e_1, \ldots, e_n$ is an orthonormal list that spans V.

Constructing Q with the column vectors of this orthonormal basis say

$$Q = \begin{pmatrix} e_1 & e_2 & \cdots & e_n \end{pmatrix}$$

This Q sqare matrix is unitary by construction with dimensions $n \times n$.

Then, multiplying Q by R upper triangular with dimensions $n \times m$ formed by the inner products of each component of v with its coresponding orthonormal vector the multiplication QR gives us the Gram-Schmidt procedure. □

Now we can see that the unitary matrix Q can be formed multiplying a series of Householder operators say H_k with the objetive of transforming a matrix A into a triangular matrix by a series of this Householder reflections in the form

$$H_k \cdots H_1 A = R$$

and therefore

$$Q = (H_k \cdots H_1)^{-1} = H_1^* \cdots H_k^*$$

The second equality holds because heach H_j is itself unitary and the composition of unitary matrices gives a unitary matrix in this case Q.

Now in order to construct the desired H_k matrix for reflecting an arbitrary vector x on some other vector choosen in an specyfic way that the reflextion lands on a vector of the orthonormal base we have to follow some steps.

First choose the first normal column vector of the canonical base with the right dimension say $e_1 = (1, 0, \dots, 0)^T$, next define a vector

$$v = x + sign(x_1)\|x\|e_1,$$

then we normalize this vector

$$u = \frac{v}{\|v\|}$$

and lastly the matrix is constructed like we defined it $H = I - 2uu^T$.

Example 4.3 (Basic reflection of a given vector example)
Given a vector say $x = (1, 2, 3)^T$ on $\mathbb{R}^3$ so the canonical basis vector we watn to choose is $e_1 = (1, 0, 0)^T$, we will obtain v as explained:

$$\begin{aligned} v &= x - \operatorname{sign}(x_1)\|x\|e_1 \\ &= x + \sqrt{1^2 + 2^2 + 3^2}e_1 \\ &= (1, 2, 3)^T + \sqrt{14}e_1 \\ &= (1 + \sqrt{14}, 2, 3)^T \end{aligned}$$

now to normalize v

$$u = \frac{v}{\|v\|} = \frac{(1 + \sqrt{14}, 2, 3)^T}{\sqrt{(1 + \sqrt{14})^2 + 13}}$$

The reflection matrix H will have the form

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - 2 \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} (u_1, u_2, u_3)$$

And with this we se that

$$Hx = \begin{pmatrix} -\sqrt{14} \\ 0 \\ 0 \end{pmatrix}$$

which is a scalar multiple of a canonical vector as we desired.

This vector reflection can be applied to not just a single vector but to an entire matrix as it is formed with vectors allowing us to transform a given matrix into a lower triangular matrix.

In the Golub and Van Loan 'Matrix Computations' book there is an implementation of this reflection that ensures no overflow neither underflow and excelent numerical stability without sacrificing that much eficiency. In asimpthotic notation whis algorithm runs in $\mathcal{O}(n)$ of time worst case, leaving the computation of the entire Householder matrix a complexity of $\mathcal{O}(n^3)$. The speudocode algorithm looks like this:

Algorithm 1 Householder reflection algorithm from 'Matrix Computations'

Require: Input vector x
Ensure: Transformed vector v and β scalar

$m = length(x),\ \sigma = x[2:m]^T x[2:m],\ v = \begin{pmatrix} 1 \\ x[2:m] \end{pmatrix}$

if $\sigma = 0$ and $x_0 \geq 0$ **then**
  $\beta = 0$
else if $\sigma = 0$ and $x_0 < 0$ **then**
  $\beta = -2$
else
  $\mu = \sqrt{x_0^2 + \sigma}$
  if $x_0 \leq 0$ **then**
    $v_0 = x_0 - \mu$
  else
    $v_0 = -\sigma/(x_0 + \mu)$
  end if
end if
return v, β

Once the Householder vector is obtained the matrix H_k is applied to the original matrix A making the kth column be all zeros bellow the diagonal like so:

$$H_1 H_2 A = \begin{pmatrix} \times & \times & \times & \times & & \times \\ 0 & \times & \times & \times & & \times \\ 0 & 0 & \times & \times & \cdots & \times \\ 0 & 0 & \times & \times & & \times \\ & & \vdots & & \ddots & \vdots \\ 0 & 0 & \times & \times & \cdots & \times \end{pmatrix}$$

With enough H operators the matrix A will transform into the desired upper diagonal R matrix now on how to compute the matrix multiplications presented above we follow the methods from Golub book.

Now, instead of computing an H_k for each Householder vector we'll use the blocking method which consists on dividing the original matrix A into blocks, computing the Householder vectors of each block and instead of forming the H matrix as we know by the formula

$$H = I - 2uu^T$$

we will acumulate those transformed vectors on clusters of size r and form the matrices

$$H = I - YTY^T$$

With T upper triangular (in Golub's book is presented as $I - WY^T$)

The full algorithm that computes the householder matrices using blocking looks like this

Algorithm 2 Block Householder QR Factorization without Explicit Q

Require: $A \in \mathbb{R}^{m\times n}$, block size r
Ensure: Upper triangular matrix R, and block reflectors $\{W_k, Y_k\}$ such that $A = QR$, $Q = \prod_k (I - W_k Y_k^T)$

```
a ← 1
k ← 0
while a ≤ n do
   t ← min(a + r − 1, n)
   k ← k + 1
   {Panel factorization (Householder QR on columns a to t)}
   for j = a to t do
      [v, β] ← House(A(j : m, j))
      A(j : m, j : t) ← (I − βvv^T)A(j : m, j : t)
      if j < m then
         A(j + 1 : m, j) ← v(2 : end)
      end if
      Store v, β for j
   end for
   {Build compact WY representation: I − W_k Y_k^T}
   Initialize Y_k, W_k as empty matrices
   for j = 1 to t − a + 1 do
      Reconstruct v_j and β_j from storage
      if j = 1 then
         Y_k(:, 1) ← v_j
         W_k(:, 1) ← β_j v_j
      else
         z ← β_j (I − W_k(:, 1 : j − 1) Y_k(:, 1 : j − 1)^T) v_j
         W_k(:, j) ← z
         Y_k(:, j) ← v_j
      end if
   end for
   {Apply block reflector to trailing matrix}
   A(a : m, t + 1 : n) ← A(a : m, t + 1 : n) − W_k(Y_k^T A(a : m, t + 1 : n))
   a ← t + 1
end while
R ← upper triangular part of modified A
```

§5 Wilkinson Shift

In the Wilkinson shift we don't compute the QR decomposition over the hermitian matrix M but over its shifted version $M - \mu I$, where μ is obtained with the following expression:

$$\mu = b - \frac{\operatorname{sign}(\delta)}{|\delta| + \sqrt{\delta^2 + \beta^2}} \cdot \beta^2$$

Here b is the last elemt of the diagonal of the M matrix, so $b = M_{n-1,n-1}$, then

$$\delta = \frac{M_{n-2,n-2} + M_{n-1,n-1}}{2}$$

and $\beta = M_{n-2,n-1}$.

§6 Why column-major?

The complete implementation in C of the SVD algorithm wich you can visit at my Github repository utilizes a *column-major* order for storing the matrices, which means is stored prioricing columns which means the matrix:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \text{ is stored as } \quad \{1, 4, 7, 2, 5, 8, 3, 6, 9\}$$

Note that this is not the normal way C stores matrices which is the *row-major* orther, but it can be seen as storing the transposed matrix.

The reason for bothering in changing the normal way of storing the data is not a minor thing, in fact many linear algebra libraries such as LAPACK use this *column-major* the reason for this is that the data is visited vertically as the program works with the matrices, that is because the vectors are represented as columns for building the matrices so we prevent unnecessary cache misses that will slow the calculations by improving the *spatial locality*. With this we obtain an improvement on the CPU cache usage on the algorithms were operations are made column by column, in our case the QR decomposition and the SVD.