Algunos algoritmos para matrices de Toeplitz

Estos apuntes están escritos por varios estudiantes de la ESFM del IPN, bajo la dirección del profesor Egor Maximenko. Participaron Jocelyn Hernández, Jareth León, Mario Guzmán Silverio, Maria de los Angeles Isidro Pérez, Darío Coutiño Aquino, Rodrigo Gabriel Castillo González, Andrea Alejandra Rendón Peña, Luis Enrique Villanueva López.

En estos apuntes se estudian algunos algoritmos numéricos para las *matrices de Toeplitz*. Por ejemplo, para $n = 4$, son matrices de la forma

$$\begin{bmatrix} a_1 & b_2 & b_3 & b_4 \\ a_2 & a_1 & b_2 & b_3 \\ a_3 & a_2 & a_1 & b_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix}.$$

Índice

1. Construcción de matrices de Toeplitz

Una matriz cuadrada se llama *matriz de Toeplitz* si sus diagonales (paralelas a la diagonal principal) son constantes. Una matriz de Toeplitz se puede definir por su primera columna y su primer renglón.

1 Ejemplo. Las matrices de Toeplitz de orden 4 son de la siguiente forma:

$$\begin{bmatrix} a_1 & b_2 & b_3 & b_4 \\ a_2 & a_1 & b_2 & b_3 \\ a_3 & a_2 & a_1 & b_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix}.$$

Los sistemas de álgebra computacional MATLAB y Wolfram Mathematica tienen funciones que generan matrices de Toeplitz a partir de los vectores a y b. En estos sistemas se supone que el vector b es de la misma longitud que el vector a, con $b_1 = a_1$. Nosotros seguimos el mismo convenio.

2 Definición. Sea $n \in \{1, 2, \ldots, \}$ y sean $a, b \in \mathbb{R}^n$, con $a_1 = b_1$. La matriz de Toeplitz generada por los vectores a y b es una matriz cuadrada A de orden n cuya entrada (j, k) está definida mediante la regla

$$A_{j,k} = \begin{cases} a_{j-k+1} & \text{si } j \geq \ k; \\ b_{k-j+1} & \text{si } k > j. \end{cases}$$

3 Algoritmo (generación de una matriz de Toeplitz en forma completa por su primera columna y su primer renglón)**.** Dados dos vectores $a, b \in \mathbb{R}^n$ la siguiente función construye la matriz de Toeplitz en la forma completa.

```
function [T] = toeplitzmatrix(a, b),
   n = length(a);
   T = zeros(n, n);
   for j = 1 : n,
      for k = 1 : j,
         T(j, k) = a(j - k + 1);
      endfor
      for k = (j + 1) : n,
         T(j, k) = b(k - j + 1);
      endfor
   endfor
endfunction
```

Para evitar ciclos anidados notamos que cada columna de la matriz se compone de una parte del vector a y una parte del vector b invertido:

$$T = \begin{bmatrix} a_1 & b_2 & b_3 & b_4 & b_5 & b_6 \\ a_2 & a_1 & b_2 & b_3 & b_4 & b_5 \\ a_3 & a_2 & a_1 & b_2 & b_3 & b_4 \\ a_4 & a_3 & a_2 & a_1 & b_2 & b_3 \\ a_5 & a_4 & a_3 & a_2 & a_1 & b_2 \\ a_6 & a_5 & a_4 & a_3 & a_2 & a_1 \end{bmatrix}, \quad a(??? :???) = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \quad b(??? : -1 :???) = \begin{bmatrix} b_5 \\ b_4 \\ b_3 \\ b_2 \end{bmatrix}.$$

Escrimos los comandos para rellenar las columnas 3–6 de la matriz T:

```
# column 3 of T:
T(1 : 2, 3) = b(3 : -1 : 2);
T(3 : 6, 3) = a(1 : 4);
# column 4 of T:
T(1 : 3, 4) = b(4 : -1 : 2);
T(4 : 6, 4) = a(??? : ???);
# column 5 of T:
T(1 : 4, 5) = b(??? : -1 : ???);
T(5 : 6, 5) = a(??? : ???);
# column 6 of T:
T(1 : 5, 6) = b(??? : -1 : ???);
T(6 : 6, 6) = a(??? : ???);
```

La misma idea funciona también para las columnas 1 y 2 de la matriz T. En el caso de fragmentos vacíos no se realizan ningunas acciones, tampoco se generan errores. Generalizando los comandos anteriores llegamos a una versión más eficiente del algoritmo anterior:

4 Algoritmo (generación de una matriz de Toeplitz en forma completa por su primera columna y su primer renglón, usando operaciones vectoriales)**.** Dados dos vectores $a, b \in \mathbb{R}^n$ la siguiente función construye la matriz de Toeplitz en la forma completa.

```
function [T] = toeplitzmatrix(a, b),
   n = length(a);
   T = zeros(n, n);
   for k = 1 : n,
      T(??? : ???, k) = a(??? : ???);
      T(??? : ???, k) = b(??? : -1 : ???);
   endfor
endfunction
```

2. Convolución discreta cíclica

5 Ejemplo. Sean $a, b \in \mathbb{C}^3$:

$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}.$$

Entonces su *convolución discreta cíclica* es el vector

$$a * b = \begin{bmatrix} a_1 b_1 + a_2 b_3 + a_3 b_2 \\ ??? \\ ??? \end{bmatrix}.$$

6 Definición (convolución discreta cíclica de dos vectores)**.** Dados dos vectores $a, b \in \mathbb{C}^n$, su *convolución discreta cíclica* es un vector $a * b \in \mathbb{C}^n$ cuya j-ésima componente es

$$(a * b)_j = ???.$$

7 Definición (transformada discreta de Fourier)**.** Denotemos por ω_n al número

$$\omega_n = e^{-\frac{2\pi i}{n}}$$

y definimos la matriz F_n mediante la regla

$$F_n = \left[\omega_n^{(j-1)(k-1)}\right]_{j,k=1}^{n}.$$

El operador lineal $x \mapsto F_n x$ se llama la *transformada discreta de Fourier*.

8 Ejemplo (matriz de la transformada discreta de Fourier de orden 3)**.**

$$F_3 = \begin{bmatrix} \omega_3^{???} & \omega_3^{???} & \omega_3^{???} \\ \omega_3^{???} & \omega_3^{???} & \omega_3^{???} \\ \omega_3^{???} & \omega_3^{???} & \omega_3^{???} \end{bmatrix}.$$

9 Definición (transformada rápida de Fourier)**.** Existe un algoritmo, llamado la *transformada rápida de Fourier*, para calcular $F_n x$ con solamente $Cn \log_2 n$ operaciones, donde C es una constante. En el lenguaje de MATLAB la transformada rápida de Fourier se puede calcular con el comando `fft(x)`, y la transformada inversa se puede calcular con `ifft(x)`.

10 Definición (producto de dos vectores por componentes)**.** Dados $a, b \in \mathbb{C}^n$, denotemos por $a \odot b$ su *producto por componentes*:

$$a \odot b = \left[a_j b_j\right]_{j=1}^{n}.$$

En el lenguaje de MATLAB el producto $a \odot b$ se puede calcular como `a .* b`.

11 Teorema (teorema de la convolución discreta cíclica)**.** *Sean $a, b \in \mathbb{C}^n$. Entonces*

$$F_n(a * b) = (F_n a) \odot (F_n b).$$

3. Matrices circulantes

12 Ejemplo. Sean $a, b \in \mathbb{C}^3$. La *matriz circulante* generada por a es

$$C(a) = \begin{bmatrix} a_1 & a_3 & a_2 \\ a_2 & a_1 & a_3 \\ a_3 & a_2 & a_1 \end{bmatrix}.$$

Notamos que el producto $C(a)b$ es lo mismo que $a * b$:

$$C(a)b = \begin{bmatrix} a_1 & a_3 & a_2 \\ a_2 & a_1 & a_3 \\ a_3 & a_2 & a_1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} ??? \\ ??? \\ ??? \end{bmatrix} = a * b.$$

13 Definición (matriz circulante generada por un vector)**.** Dado un vector $a \in \mathbb{C}^n$, denotamos por $C(a)$ a la *matriz circulante* de tamaño $n \times n$, con la entrada (j, k) definida mediante la siguiente regla:

$$C(a)_{j,k} = ???.$$

1 Proposición (producto de una matriz circulante por un vector)**.** *Sean $a, b \in \mathbb{C}^n$. Entonces*

$$C(a)b = a * b = F_n^{-1}\left(F_n(a) \odot F_n(b)\right).$$

14 Algoritmo (multiplicación rápida de una matriz circulante por un vector)**.** Dados dos vectores $a, b \in \mathbb{C}^n$, la siguiente función calcula el producto $C(a)b$ usando la transformada rápida de Fourier.

```
function [result] = myfastconv(a, b):
   result = ???;
endfunction
```

4. Multiplicación rápida de una matriz de Toeplitz por un vector

Notamos que cada matriz de Toeplitz 4×4 se puede extender a una matriz circulante de tamaño 9×9:

$$\begin{bmatrix} a_1 & b_2 & b_3 & b_4 \\ a_2 & a_1 & b_2 & b_3 \\ a_3 & a_2 & a_1 & b_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix} \mapsto \begin{bmatrix} a_1 & b_2 & b_3 & b_4 & ??? & ??? & ??? & ??? & ??? \\ a_2 & a_1 & b_2 & b_3 & ??? & ??? & ??? & ??? & ??? \\ a_3 & a_2 & a_1 & b_2 & ??? & ??? & ??? & ??? & ??? \\ a_4 & a_3 & a_2 & a_1 & ??? & ??? & ??? & ??? & ??? \\ ??? & ??? & ??? & ??? & ??? & ??? & ??? & ??? & ??? \\ ??? & ??? & ??? & ??? & ??? & ??? & ??? & ??? & ??? \\ ??? & ??? & ??? & ??? & ??? & ??? & ??? & ??? & ??? \\ ??? & ??? & ??? & ??? & ??? & ??? & ??? & ??? & ??? \\ ??? & ??? & ??? & ??? & ??? & ??? & ??? & ??? & ??? \end{bmatrix}.$$

Algunas entradas de la matriz extendida pueden tener valores arbitrarios; en esas entradas se recomienda poner 0. Notamos que la primera columna de la matriz extendida contiene el vector a completo, el vector b casi completo (las entradas de ??? a ??? en el orden invertido), y algunas entradas cero.

Más general, cada matriz de Toeplitz $n \times n$ se puede extender a una matriz circulante de tamaño $m \times m$, si

$$m \geq ??? \text{ (una expresión misteriosa en términos de } n).$$

15 Algoritmo (extensión de una matriz de Toeplitz a una matriz circulante)**.** Sean $a, b \in \mathbb{R}^n$ y sea $m \in \{1, 2, \ldots\}$ tal que $m > 2015n$???. La siguiente función construye un vector $g \in \mathbb{R}^m$ su submatriz ubicada en la intersección de los primeros n renglones y las primeras n columnas de la matriz $C(g)$ coincide con la matriz de Toeplitz asociada a los vectores a y b.

```
function [g] = extendtoeplitztocirculant(a, b, m),
   n = length(a);
   g = ???;
endfunction
```

16 Algoritmo (multiplicación rápida de una matriz de Toeplitz por un vector)**.** Sean $a, b, x \in \mathbb{R}^n$ La siguiente función multiplica la matriz de Toeplitz generada por a y b por el vector x, usando la transformada rápida de Fourier.

```
function [result] = fasttoeplitzbyvector(a, b, x),
   result = ???;
endfunction
```