ECE 133A - Homework 4

Wednesday, November 1, 2023 3:39 PM

6.9 *Circulant matrices and discrete Fourier transform.* A *circulant matrix* is a square matrix of the form

$$T(a) = \begin{bmatrix} a_1 & a_n & a_{n-1} & \cdots & a_3 & a_2 \\ a_2 & a_1 & a_n & \cdots & a_4 & a_3 \\ a_3 & a_2 & a_1 & \cdots & a_5 & a_4 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1} & a_{n-2} & a_{n-3} & \cdots & a_1 & a_n \\ a_n & a_{n-1} & a_{n-2} & \cdots & a_2 & a_1 \end{bmatrix}. \quad (9)$$

We use the notation $T(a)$ for this matrix, where $a = (a_1, a_2, \ldots, a_n)$ is the n-vector in the first column. Each of the other columns is obtained by a circular downward shift of the previous column. In matrix notation,

$$T(a) = \begin{bmatrix} a & Sa & S^2a & \cdots & S^{n-1}a \end{bmatrix}$$

where S is the $n \times n$ *circular shift* matrix

$$S = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ I_{n-1} & 0 \end{bmatrix}.$$

(a) Let W be the $n \times n$ DFT matrix:

$$W = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)(n-1)} \end{bmatrix} \quad (10)$$

where $\omega = e^{2\pi j/n}$. Verify that

$$WS^{k-1} = \mathbf{diag}(We_k)W, \quad k = 1, \ldots, n,$$

where S^i is the ith power of S (with $S^0 = I$), e_k is the kth unit vector (hence, We_k is column k of W), and $\mathbf{diag}(We_k)$ is the $n \times n$ diagonal matrix with We_k on its diagonal.

We_k = 1 where column is k

$\therefore We_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$

$$\begin{bmatrix} 1 & \vdots & & \\ \vdots & \vdots & & \\ 1 & 0 & \cdots & 0 \end{bmatrix}$$

$\therefore$ $\mathrm{diag}(We_k)$ means 0 where $j \neq k$

$$WS^{k-1} = W\begin{bmatrix} 0 & 1 \\ I_{k-2} & 0 \end{bmatrix}$$

(b) The inverse of W is $W^{-1} = (1/n)W^H$. The expression in part (a) can therefore be written as

$$S^{k-1} = \frac{1}{n} W^H \,\mathbf{diag}(We_k) W, \quad k = 1, \ldots, n.$$

Use this to show that $T(a)$ can be factored as a product of three matrices:

$$T(a) = \frac{1}{n} W^H \,\mathbf{diag}(Wa) W. \tag{11}$$

$$T(a) = [a \;\; Sa \;\; S^2a \;\ldots\; S^{n-1}a] = [a \;\; \tfrac{1}{n}W^H \mathrm{diag}(We_2)Wa \;\ldots\; \tfrac{1}{n}W^H \mathrm{diag}(We_n)Wa]$$

$$= [\underset{S_0}{I} \;\; \underset{S_1}{\tfrac{1}{n}W^H \mathrm{diag}(We_2)W} \;\ldots\; \underset{S_n}{\tfrac{1}{n}W^H \mathrm{diag}(We_n)W}]\, a$$

$$Wa = \left.\begin{matrix} a_n \\ a_i \sum_{i=1}^{n} w^{-(i-1)} \\ \vdots \\ a \sum_{j=1}^{n} w^{-(j-1)(i-1)} \end{matrix}\right\}$$ looks like We_k $\therefore$ each element of $T(a)$ being multiplied seems commutative

$$\therefore \; T(a) = \frac{1}{n} W^H \mathrm{diag}(Wa) W$$

(c) The matrix-vector product $y = Wx$ is the discrete Fourier transform of x. The matrix-vector product $y = W^{-1}x = (1/n)W^H x$ is the inverse discrete Fourier transform. The DFT and its inverse can be computed in order $n \log n$ operations using the Fast Fourier Transform algorithm.

Use the factorization (11) to formulate a fast algorithm, with a complexity of order $n \log n$, for computing matrix-vector products $T(a)x$. The product $T(a)x$ is known as the *circular convolution* of the vectors a and x.

$T(a) = \frac{1}{n} W^H \operatorname{diag}(Wa) W$

- we can see that solving the upper half of W gives us both halves ∴ $\log n$
- multiplying by a gives $\operatorname{diag}(Wa)$ → $n \log n$

$T(a)$ is essentially $n \log n$ to solve

multiply $T(a)$ with x where x is $n \times 1$ → $n \log n$

(d) Use the factorization in part (b) to formulate a fast algorithm, with an order $n \log n$ complexity, for solving a set of linear equations $Ax = b$ with variable x and coefficient matrix $A = T(a)$ (assuming $T(a)$ is nonsingular). Compare the speed of your algorithm with the standard method (`A \ b`), for randomly generated a and b. You can use the following code to generate a and b, and construct the circulant matrix $A = T(a)$.

```
a = randn(n, 1);
b = randn(n, 1);
A = toeplitz(a, [a(1), flipud(a(2:n))']);
```

Use the MATLAB functions `fft` and `ifft` to implement the fast algorithm. `y = fft(x)` evaluates $y = Wx$ using the Fast Fourier Transform algorithm; `y = ifft(x)` evaluates $y = W^{-1}x$.

Julia users will need to add the `FFTW` package to compute discrete Fourier transforms. The function `fft(x)` returns the DFT Wx of a vector x; `ifft(x)` returns the inverse DFT $W^{-1}x$. The Julia equivalent of the MATLAB code above is

```
a = randn(n,1);
b = randn(n,1);
A = hcat( [ circshift(a,k) for k=0:n-1 ]... );
```

below

6.10 Refer to the factorization (11) of a circulant matrix (9) with the n-vector a as its first column. In (11), W is the $n \times n$ discrete Fourier transform matrix and $\mathbf{diag}(Wa)$ is the diagonal matrix with the vector Wa (the discrete Fourier transform of a) on its diagonal.

(a) Suppose $T(a)$ is nonsingular. Show that its inverse $T(a)^{-1}$ is a circulant matrix. Give a fast method for computing the vector b that satisfies $T(b) = T(a)^{-1}$.

if nonsingular $\Rightarrow$ invertible

$$a = [1\ 1\ 0] \Rightarrow T(a) = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Z^{-1}

$$\begin{bmatrix} 1 & & 0 \\ & 1 & \\ 0 & & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} = \begin{bmatrix} A+G & B+H & C+I \\ A+D & B+E & C+F \\ D+G & E+H & F+I \end{bmatrix}$$

$A = 1-G \quad B = -H \quad C = -I \quad A = -D \quad B = 1-E \quad C = F \quad D = G \quad E = -H \quad F = 1-I$

$A = 1 - A \Rightarrow A = \frac{1}{2},\ G = \frac{1}{2},\ D = -\frac{1}{2}$

$B = \frac{1}{2},\ H = -\frac{1}{2},\ E = \frac{1}{2} \quad , \quad F = \frac{1}{2},\ I = \frac{1}{2},\ C = -\frac{1}{2}$

$$\therefore Z^{-1} = \frac{1}{2} \begin{bmatrix} 1 & 1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \end{bmatrix}$$

we can see it is circular

$z = \frac{1}{2}[1\ \ -1\ \ 1]$

$$\mathrm{diag}(I) = T(a)T^{-1}(a) = \begin{bmatrix} a_1 & a_n & \cdots & a_2 \\ a_2 & a_1 & & \vdots \\ \vdots & \vdots & & \\ a_n & a_{n-1} & \cdots & a_1 \end{bmatrix} \begin{bmatrix} A_1 & B_1 & \cdots & N_1 \\ \vdots & \vdots & & \vdots \\ A_n & B_n & \cdots & N_n \end{bmatrix}$$

$A_1 a_1 + A_2 a_n + \dots A_n a_2 = 1$

$B_1 a_2 + B_2 a_1 + \dots B_n a_3 = 1$

$\vdots$

we can see that the if $A_i = B_{i+1}$ then they are paired with the same a_k, by doing this across all elements $\Rightarrow$

T(a) has to be circulant

$$T(a) = \frac{1}{n} W^H \operatorname{diag}(Wa) W$$

$$T(b) = \left(\frac{1}{n} W^H \operatorname{diag}(Wa) W\right)^{-1} = \frac{1}{n} W^{-1} \operatorname{diag}(Wa)^{-1} W^{-H}$$

(b) Let a and b be two n-vectors. Show that the product $T(a)T(b)$ is a circulant matrix. Give a fast method for computing the vector c that satisfies $T(c) = T(a)T(b)$.

given that $T(a)^{-1} = T(b) \Rightarrow T(c) = I = T(a)T(b)$

$$= I = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1_n \end{bmatrix}$$

by looking at each column & thinking of moving the 1 down at each step, the result is a circulant matrix

homework4

Homework 4

A6.9d

```
n = 100
```

```
n = 100
```

```
a = randn(n, 1);
b = randn(n, 1);
A = toeplitz(a, [a(1), flipud(a(2:n))']);

tic
x = A \ b
```

```
x = 100×1
   -0.0307
```

```
   -0.0986
   -0.2964
   -0.0385
   -0.0552
    0.0717
    0.0408
   -0.4898
    0.2229
```

```
toc
```

```
Elapsed time is 0.070788 seconds.
```

```
tic
trans = fft(x)
```

```
trans = 100×1 complex
   2.7931 + 0.0000i
  -0.3835 + 1.0354i
  -1.7540 + 0.1027i
   0.5661 + 4.8063i
   2.5113 + 1.0524i
  -0.4171 + 0.4440i
  -0.0286 + 0.0081i
   0.9798 - 0.4015i
  -1.9451 + 1.4526i
   1.3681 - 0.5175i
```

```
toc
```

```
Elapsed time is 0.099894 seconds.
```

```
tic
ifft(trans)
```

```
ans = 100×1
   -0.0907
   -0.0618
   -0.0986
   -0.2984
   -0.0385
   -0.0552
    0.0717
    0.0408
   -0.4898
    0.2229
```

```
toc
```

```
Elapsed time is 0.017922 seconds.
```