EE451: SRE Presentation

Fourier Transform of PsuedoRandom Sequences

Badal Varshney
19D070015

Guided by- Prof. Virendra R. Sule
Department of Electrical Engineering
IIT Bombay

Nov. 27, 2022

Introduction

Discrete Fourier Transform (DFT):

- DFT translates the finite length time-domain discrete sequence into its frequency domain counterpart having the same length (or greater) so that no spectral information is lost and perfect reconstruction is possible
- Suppose x(n) is a sequence of finite length 'N' then its DFT is

$$y(k) = \sum_{n=0}^{N-1} x(n).e^{-i2nk\pi/N} \quad k \in \{0, 1, 2, ..., N-1\} \tag{1}$$

- The time complexity of calculating DFT results to O(N^2).
- Further, Introduction of a FFT by Cooley-Tukey, Radix-2 algorithm was developed with time complexity O(NlogN).
- Our aim is to further reduce the time complexity for large sequences.

Methodology

- Minimal Polynomial
- Henkel Matrix of Sequences
- Primitive Nth root of unity
- Fourier Transform matrix (T)
- Formation of Ω-Matrix
- DFT of sequences (F)
- Future Work

Minimal Polynomial

A sequence S $= (s_0, s_1, s_2, ..., s_m, ...s_{M-1})$ with terms in a finite field GF(q) with q elements is called a linear recurring sequence over GF(q) with minimal polynomial

$$X^m = a_0 + a_1 X + ... + a_{m-1} X^{m-1} \quad \epsilon \quad GF(q)[x] \tag{2}$$

If,

$$a_0 s_i + a_1 s_{i+1} + ... + a_m s_{i+m} = 0 \quad for \quad any \quad i \geq 1. \tag{3}$$

Take the coefficient of minimal as column vector 'α'-

$$\alpha = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{m-1} \end{pmatrix} \tag{4}$$

Henkel Matrix of Sequences

- Henkel Matrix of sequence $(s_0, s_1, s_2, ...s_{M-1})$ as defined as-

$$H(m) = \begin{pmatrix} s_0 & s_1 & \cdots & s_{m-1} \\ s_1 & s_2 & \cdots & s_m \\ \vdots & \vdots & \ddots & \vdots \\ s_{m-1} & s_{m-2} & \cdots & s_{2m-1} \end{pmatrix} \tag{5}$$

- Column vector h(m+1) is defined as

$$h(m+1) = H(m) * \alpha \tag{6}$$

Henkel Matrix Contd..

- The shifted sequence vector of our original sequence looks like-

$$h(m+1) = \begin{pmatrix} s_m \\ s_{m+1} \\ \vdots \\ s_{2m-1} \end{pmatrix} \tag{7}$$

- come up with results m $= \frac{M}{2}$
- H(m) have to satisfy this relation-

$$RankH(m) = RankH(m+j), \quad j = \{1, 2, ...\} \tag{8}$$

- 'm' is the degree of maximum rank minimum polynomial.

Primitive Nth root of unity

Let n be a positive integer. A primitive n^{th} root of unity is an n^{th} root of unity that is not a k^{th} root of unity for any positive k < n. That is, ω is a primitive n^{th} root of unity if and only if

$$\omega^n = 1, \text{ and } \omega^k \neq 1 \text{ for any positive integer } k < n. \tag{9}$$

- Here, Take the primitive n^{th} root of unity as follows-

$$\omega^n = 1 = e^{j2\pi} = cos(2\pi) + jsin(2\pi)$$
$$\implies \omega = e^{\frac{j2\pi}{n}} = cos(\frac{2\pi}{n}) + jsin(\frac{2\pi}{n}) \tag{10}$$

Fourier Transform matrix (T)

- Fourier Transform matrix (T) of DFT of length m is defined as

$$T = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^3 & \cdots & \omega^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix} \tag{11}$$

- 'w' is primitive Nth root of unity Primitive nth root of unity.
- Convert the binary sequence to real sequence of period n

$$c = (-1)^a, \text{where a is sequence element} \tag{12}$$

- c $= [(-1)^{s_0}, (-1)^{s_1}, \ldots.. , (-1)^{s_{n-1}}]$

Formation of $\Omega - Matrix$

- It is defined as identity matrix with first column embedded with zero vector and last row is embedded with minimal polynomial coefficient

$$\Omega = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_0 & a_1 & a_2 & \cdots & a_{m-1} \end{pmatrix} \tag{13}$$

- a_0, a_1, a_2, ..., a_{m-1} are the coefficients of minimal polynomial
- Ω matrix is used to get the shifted subsequences which satisfy these relation-

$$x(k+1) = \Omega * x(k), \text{where k} = \{0,1,2,....\} \tag{14}$$

and,

$$x(k) = \Omega^k * x(0) \tag{15}$$

DFT of sequences (F)

- Here, we find the DFT of sub-sequences 'c', which is defined as

$$F = T * c^t$$

$$\implies F = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^3 & \cdots & \omega^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix} \begin{pmatrix} (-1)^{s_0} \\ (-1)^{s_1} \\ (-1)^{s_2} \\ \vdots \\ (-1)^{s_{n-1}} \end{pmatrix} \tag{16}$$

- we defined the matrix X(f) as

$$X(f) = T * x(0) \tag{17}$$

DFT of sequences Contd...

- from the above relation, we get-

$$x(0) = T^{-1} * x(f) \tag{18}$$

- Combining the equation (15) and (18), we get

$$x(k) = \Omega^k * T^{-1} * x(f) \tag{19}$$

- DFT of x(k) as defined as

$$\text{DFT of x(k)} = T * x(k)$$

$$\implies T * x(k) = T * (\Omega^k * T^{-1} * x(f))$$

$$\implies T * x(k) = (T * \Omega^k * T^{-1}) * x(f) \tag{20}$$

- Now, we come up with DFT of all sub-sequences.

Future Work

- Find the way to merge all the DFT of subsequences to determine the DFT of large sequences to reduce the complexity of DFT
- Verify the property of this DFT techniques like linearity, time-shifting, frequency-shifting, convolution etc.

Thank You

References

- Prof. R. Fateman, "The (finite field) Fast Fourier Transform", https://people.eecs.berkeley.edu/ fateman/282/readings/fftnotes.pdf
- Zhi-Han Gaoa, Fang-Wei Fu, "The minimal polynomial of a sequence obtained from the componentwise linear transformation of a linear recurring sequence", Theor Comput Sci, 2010, vol 411: 3883–3893
- James L Massey and Shirlei Scrconek, "Linear Complexity of Periodic Sequences: A General Theory", N. Koblitz (Ed.): Advances in Cryptology - CRYPTO '96, LNCS 1109, pp. 358-371, 1996
- Graham H. Norton, "Minimal Polynomial Algorithms for Finite Sequences", arXiv:0911.0130v3 [cs.IT]
- Graham H. Norton. "The Berlekamp-Massey Algorithm via Minimal Polynomials", http://arXiv.org, 1001.1597.
- Howard W. Levinson and Vadim A. Markel, "Binary Discrete Fourier Transform and its Inversion", arXiv:2011.10130v2 [math.NA]

References

- Ianqin Zhou, "A fast algorithm for determining the linear complexity of periodic sequences", arXiv:cs/0512040v1 [cs.CR]
- Blackburn,S.R., A generalization of the discrete Fourier transform: determining the minimal polynomial of a periodic sequence[J]. IEEE Trans on Information Theory, 1994, 40(5): 1702-1704.
- R. Jha, R. Prasad, R. Khemka and A. Mandpura, "A Faster DFT Algorithm for Specific Binary Pulse Sequences," 2020 International Conference on Communication and Signal Processing (ICCSP), 2020, pp. 1459-1461, doi: 10.1109/ICCSP48568.2020.9182054.
- S. V. Fedorenko, "A Method for Computation of the Discrete Fourier Transform over a Finite Field", Probl Inf Transm 42, 139–151 (2006). https://doi.org/10.1134/S0032946006020074
- J.W. Cooley and J. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series" Math Comp. 19 29–301 (1965)