EE16B DFT & Polynomials

compiled by Alvin Wan

Contents

1 Properties and Theorems **3**

1.1 Property 1 3

1.1.1 Proof 3

1.2 Property 2 4

1.2.1 Proof 4

2 Lagrange Interpolation **5**

2.1 Designing Lagrange 5

2.2 Example 5

3 Linear Algebra **7**

3.1 Recovering the Polynomial, Using Matrices 7

3.2 Vandermonde Matrix 8

3.3 Exercise 9

4 Sampling **10**

4.1 Recovering Signals 10

4.2 Shannon-Nyquist Sampling Theorem 11

4.3 Up-sampling 13

4.4 Link To DFT 13

Chapter 1

Properties and Theorems

A polynomial of degree n, if $a_n \neq 0$. If $a_n = 0$, then this is at most degree $n-1$.

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

We can rewrite this polynomial in the form

$$a((x) = \sum a_k x^k$$

Question: What is the degree of polynomial $c(x) = a(x) + b(x)$, where both a and b are at most degree n?

Answer: It is *at most* n.

$$\text{DEG}(a + B) = \text{MAX}(deg(a), deg(b))$$

To multiply two polynomials, a and b,

$$a(x)b(x) = \sum c_k x^k$$

$$c_k = \sum_{j=0}^{k} a_j b_{k-j}$$

1.1 Property 1

A polynomial of degree n has *at most* n roots

1.1.1 Proof

We can additionally divide polynomials. *Google "long division". Baruch went over this, but I can't LaTeX it fast enough.*. This allows you to show the following, where $d(x)$ is the divisor and $r(x)$ is the remainder.

$$p(x) = q(x)d(x) + r(x)$$

$$\text{DEG}(r) < \text{DEG}(d)$$

We claim that this gives us the following. Since a is a root, $p(a) = 0$.

$$\frac{p(x)}{x-a} = p(a)$$
$$\frac{p(x)}{x-a} = 0$$

Consider the following.

$$p(x) = q(x)(x-a) + r$$
$$p(a) = q(a)(x-a) + r$$

Since $q(a)(x-a)$ is 0, $p(a) = r$. Let us define what a root is. a is a root is $p(a) = 0$. Additionally, if a is a root, then $p(x) = q(x)(x-a)$. We can only divide this by at most n, since each division reduces the degree of the polynomial by at most one. QED.

1.2 Property 2

If two polynomials $a(x)$, $b(x)$ of degree n agree at $n+1$ points $(\forall k \in \{0, 1 \cdots\}, a(x_k) = b(x_k))$, then $a(x) = b(x)$.

1.2.1 Proof

If $a(x)$ and $b(x)$ agree at all x_k, let us consider

$$c(x_k) = a(x_k) - b(x_k)$$

Since $a(x_k) - b(x_k)$ is 0, x_k is a root of $c(x)$. Thus, $c(x)$ would intersect the x-axis $n+1$ times, as there are $n+1$ c_k. We know $c(x)$ has degree n and thus cannot have $n+1$ roots, thus it is the 0 polynomial. This means that $a(x) = b(x)$.

$$c(x) = 0 \Rightarrow a(x) = b(x)$$

Chapter 2

Lagrange Interpolation

2.1 Designing Lagrange

We would like to find a polynomial that passes through a given set of n points. First, we design some $L_k(x_j)$ according to the following guidelines.

$$L_k(x_j) = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases}$$

Using L_k, how could we construct $a(x)$?

$$a(x) = y_0 L_o(x) + y_1 L_1(x) + \cdots + y_n L_n(x)$$

Let us consider a particular set of coordinates, (x_0, y_0).

$$a(x_0) = y_0 \cdot 1 + y_1 \cdot 0 + \cdots + y_n \cdot 0 = y_0$$

Thus, $\forall k, a(x_k) = y_k$. This means that our polynomial would indeed pass through all k points. So, we proceed to design L_k. Note that we can make this 0, if we simply consider all $(x - x_k)$, where x_k are all roots.

$$L_k(x) = (x - x_1)(x - x_2) \cdots (x - x_n)$$

This can be rewritten to be

$$L_k(x) = \Pi_{j \neq k}(x - x_j)$$

When $j \neq k$, we have 0. However, when $j = k$, we have some value other than 1. Since $x = x_k$, we can divide by x_k. Our final L_k is the following.

$$L_k(x) = \frac{\Pi_{j \neq k}(x - x_j)}{\Pi_{j \neq k}(x_k - x_j)}$$

2.2 Example

Let us consider an example for lagrange interpolation., with $(0, 2), (1, 1), (2, 4)$.

$$\begin{aligned}
L_0 &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} \\
&= \frac{(x-1)(x-2)}{(0-1)(0-2)} \\
&= \frac{1}{2}(x^2-3x+2)
\end{aligned}$$

$$\begin{aligned}
L_1 &= \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} \\
&= \frac{(x-0)(x-2)}{(1-0)(1-2)} \\
&= -(x^2-2x)
\end{aligned}$$

$$\begin{aligned}
L_2 &= \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \\
&= \frac{(x-0)(x-1)}{(2-0)(2-1)} \\
&= \frac{1}{2}(x^2-x)
\end{aligned}$$

Finally, we can compute $a(x)$.

$$\begin{aligned}
a(x) &= L_0(x)y_0 + L_1(x)y_1 + L_2(x)y_2 \\
&= 2(\frac{1}{2}(x^2-3x+2)) + -(x^2-2x) + 4(\frac{1}{2}(x^2-x)) \\
&= x^2-3x+2-x^2+2x+2x^2-2x \\
&= 2x^2-3x+2
\end{aligned}$$

Let us now verify that $a(x)$ passes through all three provided points.

$$\begin{aligned}
a(0) &= 2 \\
a(1) &= 2-3_2-1 \\
a(2) &= 2\cdot 4-6+2=4
\end{aligned}$$

Chapter 3

Linear Algebra

3.1 Recovering the Polynomial, Using Matrices

Let us consider the matrix formulation of a matrix. Consider a polynomial $a(x)$ evaluated at points $x_0, x_1 \cdots x_n$

$$a(x_0) = a_0 + a_1 x_0 + a_2 x_0^2 + \cdots a_n x_0^{n-1}$$
$$a(x_1) = a_0 + a_1 x_1 + a_2 x_1^2 + \cdots a_n x_1^{n-1}$$

Rewrite these equations in matrix form.

$$V \cdot \vec{a} = \vec{y}$$

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_n^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

This is called the **Vandermonde Matrix**. However let us consider another interpretation of lagrange interpolation in matrix form, where each of the L_k below are polynomials we found in the last chapter.

$$L \cdot \vec{y} = \vec{a}$$

$$\begin{bmatrix} L_0 & L_1 & \cdots & L_n \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

Note something interesting. Above have

$$L \cdot \vec{y} = \vec{a}$$

So, we also have the following.

$$\vec{y} = L^{-1} \cdot \vec{a}$$
$$L^{-1} \cdot \vec{a} = \vec{y}$$

From the previous part with the Vandermonde matrix, we also have the following.

$$V \cdot \vec{a} = \vec{y}$$

This necessarily means that $L = V^{-1}$! Due to the existence of lagrange interpolation, we thus know that the Vandermonde matrix *always* has an inverse. Additionally, that inverse will recover the polynomial's coefficients.

Consider the example from the last section. We will reconstruct the L matrix and show that it indeed recovers our polynomial. Recall our points are the following. $(0, 2), (1, 1), (2, 4)$

$$\begin{aligned}
L_0 &= \frac{1}{2}(x^2 - 3x + 2) \\
L_1 &= -(x^2 - 2x) \\
L_2 &= \frac{1}{2}(x^2 - x)
\end{aligned}$$

We can construct the matrix.

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -3/2 & 2 & -1/2 \\ 1/2 & -1 & 1/2 \end{bmatrix}$$

We now construct $\vec{y}$ and evaluate.

$$\begin{bmatrix} 1 & 0 & 0 \\ -3/2 & 2 & -1/2 \\ 1/2 & -1 & 1/2 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \\ 2 \end{bmatrix}$$

We thus recover our polynomial $2x^2 - 3x + 2$! Algebraically, we were reperforming the *exact* same procedure. On paper, it was simply in a different form.

3.2 Vandermonde Matrix

We see that the Vandermonde matrix is a basis. Every one of the column vectors in the following matrix are linearly independent.

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \end{bmatrix}$$

We could also use a point-value basis. There are a variety of different bases we can use. We can now switch between point-value pairs and polynomial coefficients, using lagrange.

3.3 Exercise

Consider the following. We have t and wish to recover q.

$$p(x) = x^t q(x)$$

If $q(x)$ has degree $\leq n$. This means $\textsc{Deg}(p) \leq n + t$. (Insert Baruch's joke about something being "pointless".) We can then rearrange.

$$\frac{p(x_0)}{x_0^2}\frac{p(x_1)}{x_1^t}\cdots\frac{p(x_{n-1})}{x_{n-1}^t}$$

$$q(x_0)q(x_1)\cdots q(x_{n-1})$$

We know how to interpolate and thus recover p.

In sum, we can translate coefficients into point-values by V. We can also translate from point-values back to coefficients using lagrange.

Chapter 4

Sampling

Consider a radio, receiving a specific set of signals. We know where we start and where we end. In other words, the bandwidth is limited. We need an efficient way to sample. First, we will discuss sampling. Then, connect this with polynomials at the end. Note that this is a different from the approach used in lecture.

4.1 Recovering Signals

We have some base frequency ω_0, which can be re-expressed as $2\pi f_0$. Given,

$$L(x) = \sum_{p=n_1}^{n_1} \alpha_p \cos(\omega_0 pt + \theta_p)$$

we would like to find α_p and θ_p. Once we have all α_p and θ_p, we can reconstruct the signal. For example, we can receive radio signals and then play audio. Or, we can receive satellite signals for a GPS. You could also consider this as imaging. We connect a sensor, but here, we are reconstructing a function of *space*. This way, we can up-sample the image. We can begin with a simpler version of the problem, where we work from $j = 0$ to $\lceil \frac{n}{2} \rceil - 1$.

$$L(x) = \sum_{p=0}^{\lceil \frac{n}{2} \rceil - 1} \alpha_p \cos(\omega_0 pt + \theta_p)$$

We take n samples every $T = \frac{1}{nf_0} = \frac{2\pi}{n\omega_0}$. Call the signal that represents our signal, x. We will sample at every T to take the system from continuous time to discrete time.

$$\begin{aligned} x(t) &= x(kT) \\ &= \sum \alpha_p \cos(\omega_0 pk \cdot T + \theta_p) \\ &= \sum \alpha_p \cos(\omega_0 pk \frac{2\pi}{n\omega_0} + \theta_p) \\ &= \sum_{p=0}^{\lceil \frac{n}{2} \rceil - 1} \alpha_p \cos(\frac{2\pi}{n} pk + \theta_p) \end{aligned}$$

This is reminiscent of the DFT! We can thus recover α_p and θ_p using DFT.

$$X = U^* x == \alpha_p \frac{\sqrt{n}}{2} \begin{bmatrix} 0 \\ \vdots \\ e^{j\theta} \\ \vdots \\ e^{-j\theta} \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ r_p e^{j\theta_p} \\ \vdots \\ r_p e^{-j\theta_p} \\ \vdots \\ 0 \end{bmatrix}$$

The non-zero elements are in the p and $-p$th positions, and

$$r_p = \alpha_p \frac{\sqrt{n}}{2}$$

This means that

$$\alpha_p = \frac{2r_p}{\sqrt{n}}$$

Thus, we can substitute r_p in to obtain a final expression for α_p.

$$\alpha_p = \frac{2|X[p]|}{\sqrt{n}}$$

Note that the definition for θ_p falls out.

$$\theta_p = X[p] = -X[-p]$$

Note that a sum of vectors in the DFT basis is the DFT of a sum of vectors, so taking the DFT of our time-domain signal will yield a sum of DFT basis vectors. This because a change of basis is a linear operation.

4.2 Shannon-Nyquist Sampling Theorem

There is no difference between $\cos(\frac{2\pi}{n} pk + \theta)$ with $p = \frac{n}{2}$ and $p = 2$. The **Shannon-Nyquist Theorem** generalizes this issue, stating that the maximum frequency we can represent with n samples over a period T with frequency $f_0 = \frac{1}{T}$ is $\text{MAX}(f_s) = 2f_0$.

Consider a signal in the time domain.

$$s(t) = \cos(2\pi t)$$

If we sample the signal at regular intervals, then for some integer k, t is always a multiple of our sampling period T_s, $t = kT_s$. Expand it using Euler's.

$$\begin{aligned} s(t) &= \cos(2\pi t) \\ s(kT_s) &= \cos(2\pi f k T_s) \\ &= \frac{1}{2}(e^{i2\pi f k T_s} + e^{-i2\pi k T_s}) \end{aligned}$$

We are interested knowing which DFT basis elements the exponentials correspond to. Specifically, we wish to find the indices of the elements, p and $-p$. First, we write the general form of a given DFT basis element.

$$e^{i\frac{2\pi}{N}pk}$$

We will now equate the two to find p, the index of our DFT basis element.

$$e^{i\frac{2\pi}{N}pk} = e^{i2\pi f k T_s}$$

Log both sides and then apply algebra.

$$\begin{aligned} i\frac{2\pi}{N}pk &= i2\pi f k T_s \\ \frac{p}{N} &= fT_s \\ f &= \frac{p}{NT_s} \end{aligned}$$

Consider the following table, which maps p to the continuous-time frequency. We know that the maximum p is $\lceil \frac{n}{2} \rceil - 1$. Why? Because after that, we have negative indices that wrap around the DFT.

DISCRETE	CONTINUOUS
0	0
1	$\frac{1}{nT_s}$
2	$\frac{1}{nT_s}$
$\vdots$	$\vdots$
p	$\frac{p}{nT_s}$
$\vdots$	$\vdots$
$\lceil \frac{n}{2} \rceil - 1$	$\frac{\frac{n}{2}-1}{nT_s}$

Let us simplify that last frequency.

$$\begin{aligned}\text{MAX}(f_0) &= \frac{\frac{n}{2}-1}{nT_s}\\ &\sim \frac{\frac{n}{2}}{nT_s}\\ &= \frac{1}{2T_s}\\ &= \frac{f_s}{2}\end{aligned}$$

In other words, we know that $f_0 < \frac{f_s}{2}$. We can rearrange to now get the results of Shannon-Nyquist.

$$2f_0 < f_s$$

This means that our sampling frequency is strictly *greater* than double our signal's frequency.

4.3 Up-sampling

We would like to pad a signal, and increase its signal. If we take more samples in the same time period, we get the same exact frequency. If we increase time period with the same sampling rate, the discrete signal decreases frequency. DFT does not change the norm, so as we increase the number of samples, we need to multiply by

$$\frac{\sqrt{N}}{\sqrt{n}}$$

Or, with an increase in sampling by a factor of b, we have

$$\frac{\sqrt{nb}}{\sqrt{n}} = \sqrt{b}$$

To down-sample an image: take a signal, remove all signals higher than $2n+1$; this is called a low-pass or anti-aliasing filter. Then, re-sample at the lower rate. To do this in code, zero out the middle of the vector, provided the vector gives us frequencies $0...n-1$ to get rid of higher frequencies.

4.4 Link To DFT

$$f(t) = \sum_{p=0}^{n-1} \alpha_p e^{j\frac{\omega_s}{n}pt}$$

We take n samples at kT_s.

$$f(kT_s) = \sum_{p=0}^{n-1} \alpha_p e^{j\frac{2\pi}{nT_s}kpT_s}$$

$$= \sum_{p=0}^{n-1} \alpha_p e^{j\frac{2\pi}{n}kp}$$

$$= \sum_{p=0}^{n-1} \alpha_p (e^{j\frac{2\pi}{n}k})^p$$

Let $x = e^{j\frac{2\pi}{n}k}$.

$$= \sum_{p=0}^{n-1} \alpha_p x^p$$

Note that this is the structure of a polynomial!

$$p(x) = \sum_{p=0}^{n-1} \alpha_p x^p$$

Thus, sampling the signal at n times is the same as taken n points for a polynomial, where the sampling points are the roots of unity.

$$p(e^{j\frac{2\pi}{n}k}) = f(kT_s)$$