Approximating the Fourier Transform with the Discrete Fourier Transform

Nicholas Dwork

The Fourier Transform is a very useful tool. One would hope, and indeed it's the case, that we could use the Discrete Fourier Transform (DFT) to approximate the Fourier Transform. This allows us to use computers to take advantage of Fourier Theory (as has been done for decades).

For this document, we will use the following definitions for the Fourier Transform and inverse Fourier Transform:

$$F(k) = \mathcal{F}\{f\}(k) = \int_{-\infty}^{\infty} f(x)\, \exp(-i\, 2\pi\, k\, x)\, dx$$
$$f(x) = \mathcal{F}^{-1}\{F\}(x) = \int_{-\infty}^{\infty} F(k) \exp(i\, 2\pi\, k\, x)\, dk,$$

where $i = \sqrt{-1}$. The following definitions specify the DFT and inverse DFT for this document:

$$V[m] = \mathsf{DFT}(v)[m] = \sum_{n=0}^{N-1} v[n] \exp\left(-i\, 2\pi\, \frac{m\, n}{N}\right)$$
$$v[n] = \mathsf{DFT}^{-1}(V)[n] = \frac{1}{N} \sum_{k=0}^{N-1} V[m] \exp\left(i\, 2\pi\, \frac{m\, n}{N}\right).$$

1 Riemann Sum Approximation

Approximating the Fourier Transform as a Riemann sum yields the following expression:

$$F(k) \approx \sum_j f(x_j) \exp\left(-i\, 2\pi\, k\, x_j\right) \Delta x_j.$$

Let us assume that the samples $\{f(x_j)\}$ were sampled over a finite domain with a uniform sampling rate of 1 (i.e. $\Delta x_j = 1$); this is a very common convention. Then

$$F(k) \approx \sum_{n=0}^{N-1} f(x_n) \exp\left(-i\, 2\pi\, k\, x_n\right).$$

Let $\boldsymbol{f}$ be the N-tuple $(f(x_0), \cdots, f(x_{N-1}))$. Then

$$F(k) \approx \sum_{n=0}^{N-1} \boldsymbol{f}[n] \exp\left(-i\, 2\pi\, k\, n\right).$$

Suppose we sample the Fourier domain at frequencies $\boldsymbol{k} = \left(0, \frac{1}{N}, \cdots, \frac{N-1}{N}\right)$. Then

$$F(\boldsymbol{k}_m) \approx \sum_{n=0}^{N-1} \boldsymbol{f}[n] \exp\left(-i\, 2\pi\, \frac{m\, n}{N}\right), \qquad \text{for } m \in \{0, 1, \cdots, N-1\}.$$

But the expression on the right is just the DFT! That is, $F(\boldsymbol{k}) \approx \mathsf{DFT}(\boldsymbol{f})$. This shows that we can approximate the Fourier Transform using the DFT.

We will go through the same procedure for the inverse Fourier Transform:

$$f(x) = \int_{-\infty}^{\infty} F(k)\, \exp\left(i\, 2\pi\, k\, x\right)\, dk.$$

Approximating this as a Riemann sum yields

$$f(x) \approx \sum_j F(k_j) \exp\left(i\, 2\pi\, k_j\, x\right)\, \Delta k_j.$$

Assuming we're interested in the values of $\boldsymbol{x} = (0, 1, \cdots, N-1)$, and letting $\boldsymbol{F} = \left(F(0), F\left(\frac{1}{N}\right), \cdots, F\left(\frac{N-1}{N}\right)\right)$,

$$f(x) \approx \frac{1}{N} \sum_{m=0}^{N-1} \boldsymbol{F}[m]\, \exp\left(i\, 2\pi\, \frac{m\, n}{N}\right).$$

But this is just the inverse Discrete Fourier Transform! That is, $f(\boldsymbol{t}) \approx \mathsf{DFT}^{-1}(\boldsymbol{F})$. This shows that we can approximate the inverse Fourier Transform with the inverse DFT.

One might suspect that these approximations are pretty bad. After all, the Fourier Transform and inverse Fourier Transform are integrals from $-\infty$ to ∞, and the DFT and inverse DFT are finite sums. Fortunately, we can quantify these errors and in many cases they aren't very bad. This is the topic of the next section.

2 Sampling

Sampling is the process of attaining a finite subset of the image of a function[1]. A set of N uniformly spaced samples of the function f is the vector $\boldsymbol{f} = (f(0), f(1), \ldots, f(N-1)) \in \mathbb{C}^N$.

A uniformly sampled function over a square subset of the domain is modeled as $f\, Ш\, \Pi_\Delta$ where $Ш$ is the comb (or Shah[2]) function defined as $Ш(x) = \sum_{n=-\infty}^{\infty} \delta(x - n)$ and Π_Δ is the rect function defined as

$$\Pi_\Delta(x) = \begin{cases} 1 & \text{if } |x| \leq \Delta/2 \\ 0 & \text{otherwise} \end{cases}.$$

Consider the Fourier Transform of $f\, Ш\, \Pi_\Delta$. By definition of the Fourier Transform,

$$\begin{aligned} \mathcal{F}\left\{f\, Ш\, \Pi_\Delta\right\}(k) &= \int_{-\infty}^{\infty} f(x) Ш(x) \Pi_\Delta(x) \exp\left(-i\, 2\pi\, k\, x\right) dx \\ &= \int_{-\Delta/2}^{\Delta/2} f(x) Ш(x) \exp\left(-i\, 2\pi\, k\, x\right) dx \\ &= \sum_{n=-N/2}^{N/2} f(n \bmod N) \exp\left(-i\, 2\pi\, k\, n\right) \qquad (1) \\ &= \sum_{n=0}^{N-1} f(n) \exp\left(-i\, 2\pi\, k\, n\right), \end{aligned}$$

where in (1) we assumed that unknown samples of f are related to $\boldsymbol{f}$ through the periodic extension of $\boldsymbol{f}$ defined as $\boldsymbol{f}_{p.e.}(n) = \boldsymbol{f}(n \bmod N)$. Note that in the above expressions, we've assumed that N is even; a similar argument holds when N is odd. When we evaluate the above expression at values of $k = m/N$ we get

$$\mathcal{F}\left\{f\, Ш\, \Pi_\Delta\right\}\left(\frac{m}{N}\right) = \sum_{n=0}^{N-1} f(n) \exp\left(-i\, 2\pi\, \frac{m\, n}{N}\right).$$

But this is just the DFT of $\boldsymbol{f}$!

[1] A real world method of sampling is through the use of an Analog-to-Digital Converter (ADC). A computer will periodically probe (or sample) the ADC, read off values, and store them in memory. Thus, a finite subset of the analog signal is retained.

[2] I have included here for reference the following property of the comb function: $Ш(k/\Delta k) = \Delta k\, Ш_{\Delta k}(k)$ where $Ш_{\Delta k}(k) = \sum_{n=-\infty}^{\infty} \delta(k - n\, \Delta k)$.

Consider again the Fourier Transform of $f\,\text{III}\,\Pi_\Delta$. By the convolution theorem, $\mathcal{F}\left\{f\,\text{III}\,\Pi_\Delta\right\}(k) = (F * \text{III} * b)\,(k)$ where $b(k) = \Delta\,\mathrm{sinc}(k\,\Delta)$. Since both expressions of $\mathcal{F}\left\{f\,\text{III}\,\Pi_\Delta\right\}$ are equal, we arrive at this very important result:

$$\boxed{DFT\,(\boldsymbol{f})\,[m] = (F * \text{III} * b)\left(\frac{m}{N}\right).}$$

So we see that the DFT attains samples of $F(k)$ corrupted by a convolution with III and a convolution with b! The convolution with b results in a blurring of F and convolution with III results in aliasing (where copies of the original function are made and summed together).