Lecture 26: Complex matrices; fast Fourier transform

Matrices with all real entries can have complex eigenvalues! So we can't avoid working with complex numbers. In this lecture we learn to work with complex vectors and matrices.

The most important complex matrix is the Fourier matrix F_n, which is used for Fourier transforms. Normally, multiplication by F_n would require n^2 multiplications. The fast Fourier transform (FFT) reduces this to roughly $n \log_2 n$ multiplications, a revolutionary improvement.

Complex vectors

Length

Given a vector $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \in \mathbb{C}^n$ with complex entries, how do we find its length? Our old definition:

$$\mathbf{z}^T\mathbf{z} = \begin{bmatrix} z_1 & z_2 & \cdots & z_n \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}$$

is no good; this quantity isn't always positive. For example:

$$\begin{bmatrix} 1 & i \end{bmatrix} \begin{bmatrix} 1 \\ i \end{bmatrix} = 0.$$

We don't want to define the length of $\begin{bmatrix} 1 \\ i \end{bmatrix}$ to be 0. The correct definition is: $|\mathbf{z}|^2 = \overline{\mathbf{z}}^T\mathbf{z} = |z_1|^2 + |z_2|^2 + \cdots + |z_n|^2$. Then we have:

$$\left(\text{length} \begin{bmatrix} 1 \\ i \end{bmatrix}\right)^2 = \begin{bmatrix} 1 & -i \end{bmatrix} \begin{bmatrix} 1 \\ i \end{bmatrix} = 2.$$

To simplify our notation we write $|\mathbf{z}|^2 = \mathbf{z}^H\mathbf{z}$, where $\mathbf{z}^H = \overline{\mathbf{z}}^T$. The H comes from the name Hermite, and $\mathbf{z}^H\mathbf{z}$ is read "$\mathbf{z}$ Hermitian $\mathbf{z}$".

Inner product

Similarly, the inner or dot product of two complex vectors is not just $\mathbf{y}^T\mathbf{x}$. We must also take the complex conjugate of $\mathbf{y}$:

$$\mathbf{y}^H\mathbf{x} = \overline{\mathbf{y}}^T\mathbf{x} = \overline{y}_1x_1 + \overline{y}_2x_2 + \cdots + \overline{y}_nx_n.$$

Complex matrices

Hermitian matrices

Symmetric matrices are real valued matrices for which $A^T = A$. If A is complex, a nicer property is $\overline{A}^T = A$; such a matrix is called *Hermitian* and we abbreviate $\overline{A}^T$ as A^H. Note that the diagonal entries of a Hermitian matrix must be real. For example,

$$\overline{A}^T = A = \begin{bmatrix} 2 & 3+i \\ 3-i & 5 \end{bmatrix}.$$

Similar to symmetric matrices, Hermitian matrices have real eigenvalues and perpendicular eigenvectors.

Unitary matrices

What does it mean for complex vectors $\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_n$ to be perpendicular (or orthonormal)? We must use our new definition of the inner product. For a collection of $\mathbf{q}_j$ in complex space to be orthonormal, we require:

$$\overline{\mathbf{q}}_j \mathbf{q}_k = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}$$

We can again define $Q = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{bmatrix}$, and then $Q^H Q = I$. Just as "Hermitian" is the complex equivalent of "symmetric", the term "*unitary*" is analogous to "orthogonal". A *unitary matrix* is a square matrix with perpendicular columns of unit length.

Discrete Fourier transform

A *Fourier series* is a way of writing a periodic function or *signal* as a sum of functions of different frequencies:

$$f(x) = a_0 + a_1 \cos x + b_1 \sin x + a_2 \cos 2x + b_2 \sin 2x + \cdots .$$

When working with finite data sets, the *discrete Fourier transform* is the key to this decomposition.

In electrical engineering and computer science, the rows and columns of a matrix are numbered starting with 0, not 1 (and ending with $n-1$, not n). We'll follow this convention when discussing the Fourier matrix:

$$F_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & & w^{n-1} \\ 1 & w^2 & w^4 & & w^{2(n-1)} \\ \vdots & & & \ddots & \vdots \\ 1 & w^{n-1} & w^{2(n-1)} & \cdots & w^{(n-1)^2} \end{bmatrix}.$$

Notice that $F_n = F_n^T$ and $(F_n)_{jk} = w^{jk}$, where $j, k = 0, 1, ..., n-1$ and the complex number w is $w = e^{i \cdot 2\pi/n}$ (so $w^n = 1$). The columns of this matrix are orthogonal.

All the entries of F_n are on the unit circle in the complex plane, and raising each one to the nth power gives 1. We could write $w = \cos(2\pi/n) + i\sin(2\pi/n)$, but that would just make it harder to compute w^{jk}.

Because $w^4 = 1$ and $w = e^{2\pi i/4} = i$, our best example of a Fourier matrix is:

$$F_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & i^2 & i^3 \\ 1 & i^2 & i^4 & i^6 \\ 1 & i^3 & i^6 & i^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}.$$

To find the Fourier transform of a vector with four components (four data points) we multiply by F_4.

It's easy to check that the columns of F_4 are orthogonal, as long as we remember to conjugate when computing the inner product. However, F_4 is not quite unitary because each column has length 2. We could divide each entry by 2 to get a matrix whose columns are orthonormal:

$$\frac{1}{4} F_4^H F_4 = I.$$

An example

The signal corresponding to a single impulse at time zero is (roughly) described by $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$. To find the Fourier transform of this signal we compute:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

A single impulse has all frequencies in equal amounts.

If we multiply by F_4 again we almost get back to $(1, 0, 0, 0)$:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 4 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Because $\frac{1}{\sqrt{n}} F_n$ is unitary, multiplying by F_n and dividing by the scalar n inverts the transform.

Fast Fourier transform

Fourier matrices can be broken down into chunks with lots of zero entries; Fourier probably didn't notice this. Gauss did, but didn't realize how significant a discovery this was.

There's a nice relationship between F_n and F_{2n} related to the fact that $w_{2n}^2 = w_n$:

$$F_{2n} = \begin{bmatrix} I & D \\ I & -D \end{bmatrix} \begin{bmatrix} F_n & 0 \\ 0 & F_n \end{bmatrix} P,$$

where D is a diagonal matrix and P is a $2n$ by $2n$ permutation matrix:

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

So, a $2n$ sized Fourier transform F times $\mathbf{x}$ which we might think would require $(2n)^2 = 4n^2$ operations can instead be performed using two size n Fourier transforms ($2n^2$ operations) plus two very simple matrix multiplications which require on the order of n multiplications. The matrix P picks out the even components x_0, x_2, x_4, ... of a vector first, and then the odd ones – this calculation can be done very quickly.

Thus we can do a Fourier transform of size 64 on a vector by separating the vector into its odd and even components, performing a size 32 Fourier transform on each half of its components, then recombining the two halves through a process which involves multiplication by the diagonal matrix D.

$$D = \begin{bmatrix} 1 & & & & & \\ & w & & & & \\ & & w^2 & & & \\ & & & \ddots & & \\ & & & & w^{n-1} \end{bmatrix}.$$

Of course we can break each of those copies of F_{32} down into two copies of F_{16}, and so on. In the end, instead of using n^2 operations to multiply by F_n we get the same result using about $\frac{1}{2} n \log_2 n$ operations.

A typical case is $n = 1024 = 2^{10}$. Simply multiplying by F_n requires over a million calculations. The fast Fourier transform can be completed with only $\frac{1}{2} n \log_2 n = 5 \cdot 1024$ calculations. This is 200 times faster!

This is only possible because Fourier matrices are special matrices with orthogonal columns. In the next lecture we'll return to dealing exclusively with real numbers and will learn about positive definite matrices, which are the matrices most often seen in applications.