A Non-sparse Tutorial on Sparse FFTs

Mark Iwen

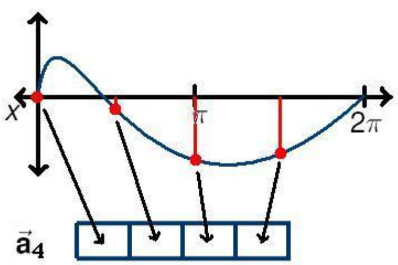
Duke University

February 17, 2013

Problem Setup

Recover $f : [0, 2\pi] \mapsto \mathbb{C}$ consisting of k trigonometric terms

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \; \Omega = \{\omega_1, \dots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

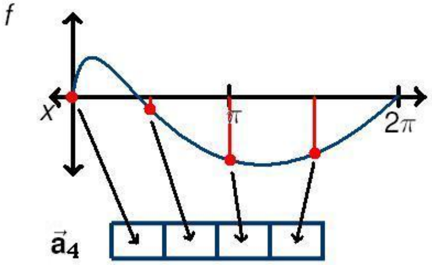


- Approximate $\left\{(\omega_j, C_j) \mid 1 \le j \le k\right\}$ using only $\vec{a}_N$

Problem Setup

Recover $f : [0, 2\pi] \mapsto \mathbb{C}$ consisting of k trigonometric terms

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathrm{e}^{x \cdot \omega_j \cdot \mathrm{i}}, \ \Omega = \{\omega_1, \dots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$



- Approximate $\left\{(\omega_j, C_j) \mid 1 \le j \le k\right\}$ using only $\vec{a}_N$

A Woefully Incomplete History of "Fast" Sparse FFTs

Recover $f : [0, 2\pi] \mapsto \mathbb{C}$ consisting of k trigonometric terms

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \ \Omega = \{\omega_1, \dots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

- The Fast Fourier Transform (FFT) [CT'65] can approximate (ω_j, C_j), $1 \leq j \leq k$, in $O(N \log N)$-time. Efficient FFT implementations that minimize the hidden constants have been developed (e.g., FFTW [FJ' 05)).
- Mansour [M'95]; Akavia, Goldwasser, Safra [AGS' 03]; Gilbert, Guha, Indyk, Muthukrishnan, Strauss [GGIMS' 02] & [GMS' 05]; I., Segal [I'13] & [SI'12]; Hassanieh, Indyk, Katabi, Price [HIKPs'12] & [HIKPst'12]; ... $O(k \log^c N)$-time

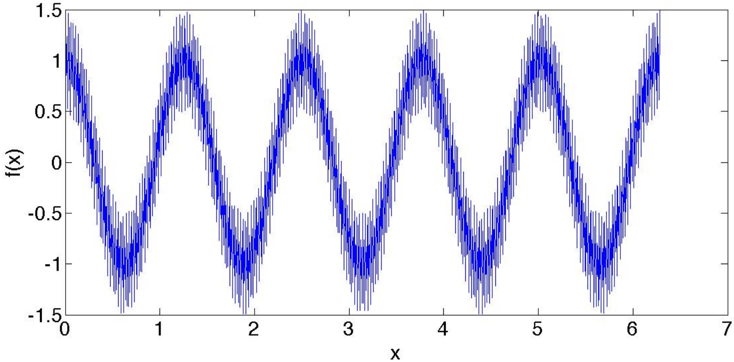
A Woefully Incomplete History of "Fast" Sparse FFTs

Recover $f : [0, 2\pi] \mapsto \mathbb{C}$ consisting of k trigonometric terms

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \; \Omega = \{\omega_1, \dots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

- The Fast Fourier Transform (FFT) [CT'65] can approximate (ω_j, C_j), $1 \leq j \leq k$, in $O(N \log N)$-time. Efficient FFT implementations that minimize the hidden constants have been developed (e.g., FFTW [FJ' 05]).
- Mansour [M'95]; Akavia, Goldwasser, Safra [AGS' 03]; Gilbert, Guha, Indyk, Muthukrishnan, Strauss [GGIMS' 02] & [GMS' 05]; I., Segal [I'13] & [SI'12]; Hassanieh, Indyk, Katabi, Price [HIKPs'12] & [HIKPst'12]; . . . $O(k \log^c N)$-time

Example: $\cos(5x) + .5\cos(400x)$



- $f(x) = (1/4)\mathbb{e}^{-400x\cdot\mathbb{i}} + (1/2)\mathbb{e}^{-5x\cdot\mathbb{i}} + (1/2)\mathbb{e}^{5x\cdot\mathbb{i}} + (1/4)\mathbb{e}^{400x\cdot\mathbb{i}}$
- $\Omega = \{-400, -5, 5, 400\}$
- $C_1 = C_4 = 1/4$, and $C_2 = C_3 = 1/2$

Four Step Approach

Approximate $\left\{(\omega_j, C_j) \,\middle|\, 1 \leq j \leq k\right\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \; \Omega = \{\omega_1, \ldots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

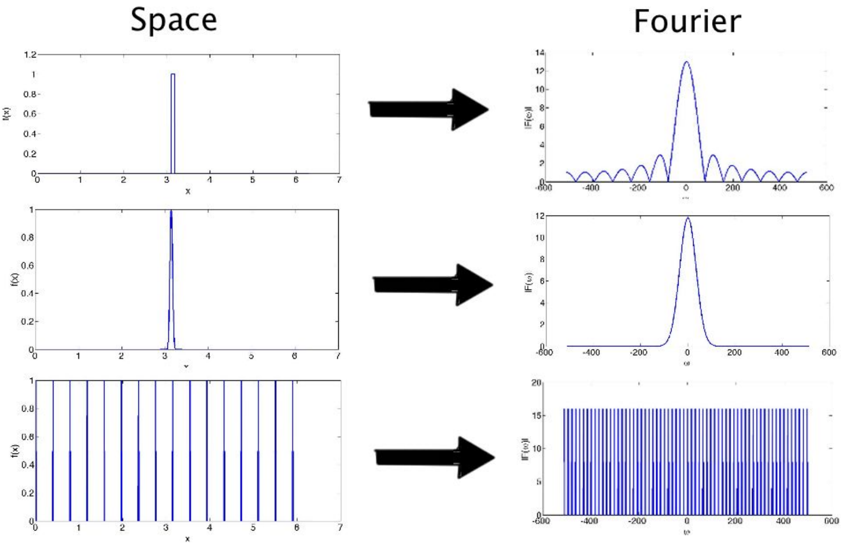
A Sparse Fourier Transform will...

1. Try to isolate each frequency, $\omega_j \in \Omega$, in some

$$f_j(x) = C_j' \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}} \; + \; \epsilon(x)$$

2. $\tilde{\Omega} \leftarrow$ Use $f_j(x)$ to learn all $\omega_j \in \Omega$
3. $\tilde{C}_j \leftarrow$ Estimate C_j for each $\omega_j \in \tilde{\Omega}$
4. Repeat on $f - \sum_{\omega_j \in \tilde{\Omega}} \tilde{C}_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}$, or not...

Design Decision #1: Pick a Filter



Design Decision #1: Pick a Filter

Previous Choices

- (Indicator function,Dirichlet) Pair: [GGIMS' 02] & [GMS' 05]
- (Spike Train,Spike Train) Pair: [I'13] & [SI'12]
- (Conv[Gaussian,Indicator],Gaussian$\times$Dirichlet) Pair[1]: [HIKPs'12] & [HIKPst'12]

We'll use a regular Gaussian today

[1] Also consider Dolph-Chebyshev window function. . .

Design Decision #1: Pick a Filter

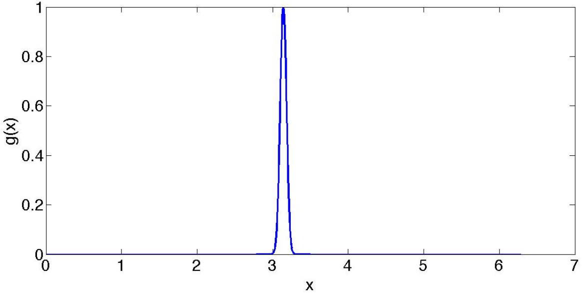
Previous Choices

- (Indicator function,Dirichlet) Pair: [GGIMS' 02] & [GMS' 05]
- (Spike Train,Spike Train) Pair: [I'13] & [SI'12]
- (Conv[Gaussian,Indicator],Gaussian$\times$Dirichlet) Pair[1]: [HIKPs'12] & [HIKPst'12]

We'll use a regular Gaussian today

[1] Also consider Dolph-Chebyshev window function. . .

Gaussian with "Small Support" in Space

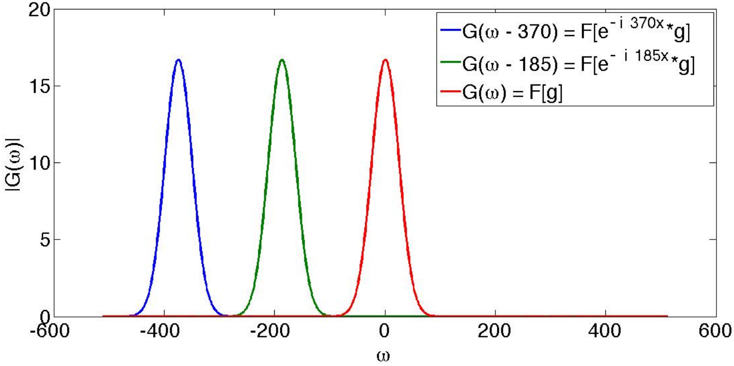


- Supports fast approximate convolutions: $\text{Conv}[g, f](j\Delta x)$ is

$$\sum_{h=0}^{N-1} g(h\Delta x) f\left((j-h)\Delta x\right) \approx \sum_{h=N/2-c}^{N/2+c} g(h\Delta x) f\left((j-h)\Delta x\right).$$

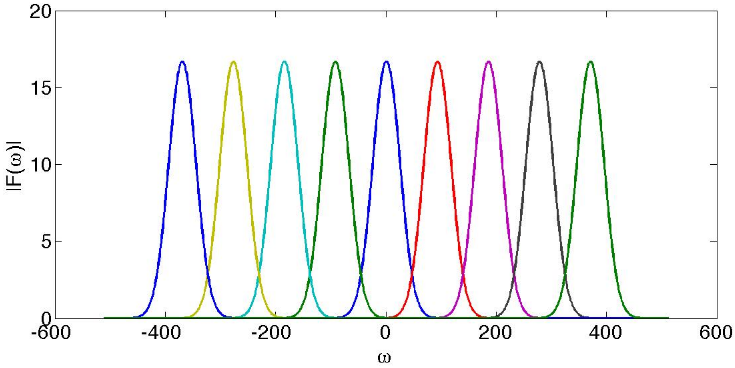
- $\Delta x = 2\pi/N$, c small

Gaussian has “Large Support” in Fourier



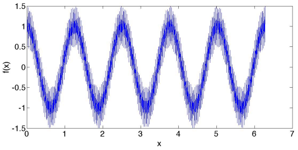
- Modulating the filter, g, a small number of times allows us to bin the Fourier spectrum

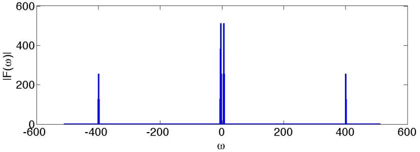
Gaussian has "Large Support" in Fourier



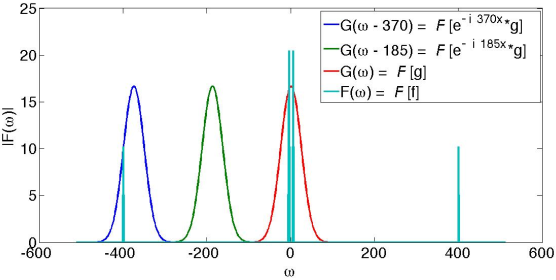
- Modulating the filter, g, a small number of times allows us to bin the Fourier spectrum

Example: Convolutions Bin Fourier Spectrum

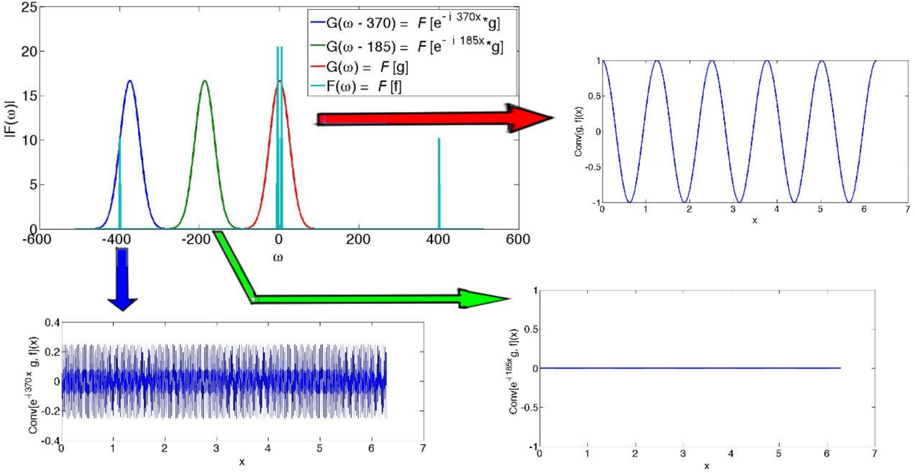




- $\mathcal{F}\left[Conv[g,f](x)\right](\omega) = \mathcal{F}[g](\omega) * \mathcal{F}[f](\omega)$
- Convolving allows us to select parts of *f*'s spectrum



Example: Convolutions Bin Fourier Spectrum



Binning Summary

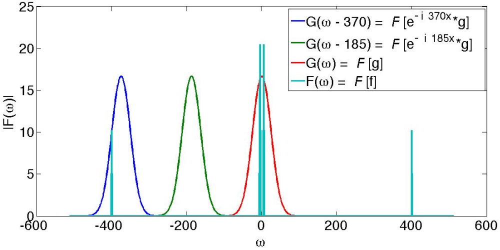
1. Large support in Fourier $\implies$ Need few modulations of g to bin

$$\mathrm{e}^{-\mathrm{i}2ax}g(x),\ \mathrm{e}^{-\mathrm{i}ax}g(x),\ g(x),\ \mathrm{e}^{\mathrm{i}ax}g(x),\ \mathrm{e}^{\mathrm{i}2ax}g(x)$$

2. Small Support in Space $\implies$ Need few samples for convolutions

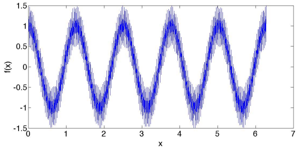
$$\mathrm{Conv}[\mathrm{e}^{-\mathrm{i}ax}g, f](j\Delta x) \approx \sum_{h=\frac{N}{2}-c}^{\frac{N}{2}+c} \mathrm{e}^{-\mathrm{i}ah\Delta x} g(h\Delta x) f\left((j-h)\Delta x\right),\ c \text{ small}$$

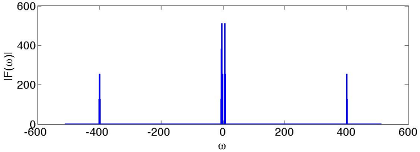
3. Problem: Two frequencies can be binned in the same bucket



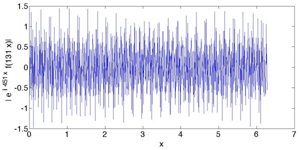
Shift and Spread the Spectrum of f

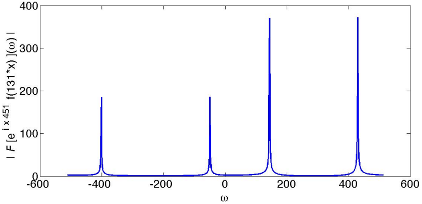
- f $\qquad$ $\mathcal{F}[f](\omega)$



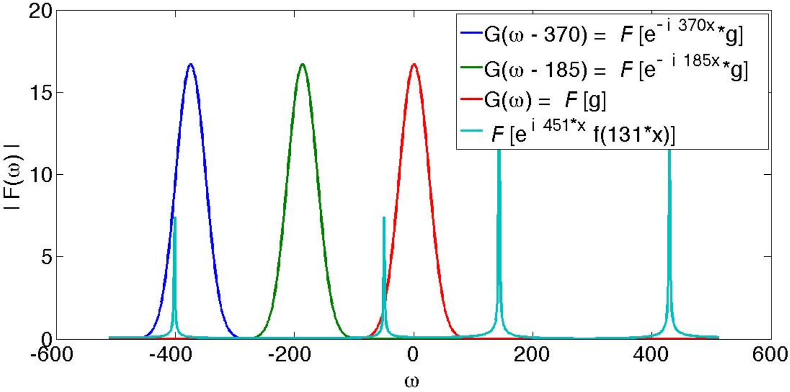


- $\mathbb{e}^{\mathbb{i}451x} f(131 * x)$ $\qquad$ $\mathcal{F}\left[\mathbb{e}^{\mathbb{i}451x} f(131 * x)\right](\omega)$





Frequency Isolation



- We have isolated one of the previously collided frequencies in

$$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}370x}g(x),\ \mathbb{e}^{\mathbb{i}451x}f(131x)](x)$$

Frequency Isolation Summary

1. Choose filter g with small support in space, large support in Fourier
2. Randomly select dilation and modulation pairs, $(d_l, m_l) \in \mathbb{Z}^2$
3. Each energetic frequency in f, $\omega_j \in \Omega$, will have a proxy isolated in

$$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x),\ \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$

   for some n, m_l, d_l triple with high probability.
4. Analyzing probability of isolation is akin to considering tossing balls (frequencies of f) into bins (pass regions of modulated filter)
5. Computing each convolution at a given x of interest is fast since g has small support in space

Frequency Isolation Summary

1. Choose filter g with small support in space, large support in Fourier
2. Randomly select dilation and modulation pairs, $(d_l, m_l) \in \mathbb{Z}^2$
3. Each energetic frequency in f, $\omega_j \in \Omega$, will have a proxy isolated in

$$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x),\ \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$

for some n, m_l, d_l triple with high probability.

4. Analyzing probability of isolation is akin to considering tossing balls (frequencies of f) into bins (pass regions of modulated filter)
5. Computing each convolution at a given x of interest is fast since g has small support in space

Frequency Isolation Summary

1. Choose filter g with small support in space, large support in Fourier

2. Randomly select dilation and modulation pairs, $(d_l, m_l) \in \mathbb{Z}^2$

3. Each energetic frequency in f, $\omega_j \in \Omega$, will have a proxy isolated in

$$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i} nax} g(x),\ \mathbb{e}^{\mathbb{i} m_l x} f(d_l x)](x)$$

for some n, m_l, d_l triple with high probability.

4. Analyzing probability of isolation is akin to considering tossing balls (frequencies of f) into bins (pass regions of modulated filter)

5. Computing each convolution at a given x of interest is fast since g has small support in space

Frequency Isolation Summary

1. Choose filter g with small support in space, large support in Fourier

2. Randomly select dilation and modulation pairs, $(d_l, m_l) \in \mathbb{Z}^2$

3. Each energetic frequency in f, $\omega_j \in \Omega$, will have a proxy isolated in

$$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x),\ \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$

for some n, m_l, d_l triple with high probability.

4. Analyzing probability of isolation is akin to considering tossing balls (frequencies of f) into bins (pass regions of modulated filter)

5. Computing each convolution at a given x of interest is fast since g has small support in space

Frequency Isolation Summary

1. Choose filter g with small support in space, large support in Fourier
2. Randomly select dilation and modulation pairs, $(d_l, m_l) \in \mathbb{Z}^2$
3. Each energetic frequency in f, $\omega_j \in \Omega$, will have a proxy isolated in
$$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x),\ \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$
for some n, m_l, d_l triple with high probability.
4. Analyzing probability of isolation is akin to considering tossing balls (frequencies of f) into bins (pass regions of modulated filter)
5. Computing each convolution at a given x of interest is fast since g has small support in space

Design Decision #2: Frequency Identification

Frequency Isolated in a Convolution

$$f_j(x) := \mathrm{Conv}[\mathbb{e}^{-\mathbb{i} n_j a x} g(x),\ \mathbb{e}^{\mathbb{i} m_{l_j} x} f(d_{l_j} x)](x) = C_j' \cdot \mathbb{e}^{x \cdot \omega_j' \cdot \mathbb{i}} + \epsilon(x)$$

1. Compute the phase of

$$\frac{f_j(h_1 \Delta x)}{f_j(h_1 \Delta x + \pi)} \approx \mathbb{e}^{\pi \mathbb{i} \cdot \omega_j'}$$

2. Perform a modified binary search for ω_j'. A variety of methods exist for making decisions about the set of frequencies ω_j' belongs to at each stage of the search...

Identification Example: One Nonzero Entry

- $M \in \{0,1\}^{5\times 6}$, $\hat{f}_j \in \mathbb{C}^6$ contains 1 nonzero entry.

$$\begin{array}{r} \equiv 0 \bmod 2 \\ \equiv 1 \bmod 2 \\ \equiv 0 \bmod 3 \\ \equiv 1 \bmod 3 \\ \equiv 2 \bmod 3 \end{array} \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- Reconstruct entry index via Chinese Remainder Theorem
- Two estimates of the entry's value

SAVED ONE LINEAR TEST!

Identification Example: One Nonzero Entry

- $M \in \{0,1\}^{5\times 6}$, $\hat{f}_j \in \mathbb{C}^6$ contains 1 nonzero entry.

$$\begin{array}{c} \equiv 0 \bmod 2 \\ \equiv 1 \bmod 2 \\ \equiv 0 \bmod 3 \\ \equiv 1 \bmod 3 \\ \equiv 2 \bmod 3 \end{array} \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- Reconstruct entry index via Chinese Remainder Theorem
- Two estimates of the entry's value

SAVED ONE LINEAR TEST!

Identification Example: One Nonzero Entry

- $M \in \{0,1\}^{5\times 6}$, $\hat{f}_j \in \mathbb{C}^6$ contains 1 nonzero entry.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix} \begin{array}{l} \Leftarrow \text{Index} \equiv 0 \bmod 2 \\ \\ \\ \\ \Leftarrow \text{Index} \equiv 2 \bmod 3 \end{array}$$

- Reconstruct entry index via Chinese Remainder Theorem
- Two estimates of the entry's value

SAVED ONE LINEAR TEST!

Identification Example: One Nonzero Entry

- $M \in \{0,1\}^{5\times 6}$, $\hat{f}_j \in \mathbb{C}^6$ contains 1 nonzero entry.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix} \begin{array}{l} \Leftarrow \text{ Index } \equiv 0 \bmod 2 \\ \\ \\ \\ \Leftarrow \text{ Index } \equiv 2 \bmod 3 \end{array}$$

- Reconstruct entry index via Chinese Remainder Theorem
- Two estimates of the entry's value

SAVED ONE LINEAR TEST!

Identification Example: One Nonzero Entry

- $M \in \{0,1\}^{5\times 6}$, $\hat{f}_j \in \mathbb{C}^6$ contains 1 nonzero entry.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix} \begin{array}{l} \Leftarrow \text{Index } \equiv 0 \bmod 2 \\ \\ \\ \\ \Leftarrow \text{Index } \equiv 2 \bmod 3 \end{array}$$

- Reconstruct entry index via Chinese Remainder Theorem
- Two estimates of the entry's value

SAVED ONE LINEAR TEST!

Identification Example: One Nonzero Entry

- $M \in \{0,1\}^{5\times 6}$, $\hat{f}_j \in \mathbb{C}^6$ contains 1 nonzero entry.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix} \begin{array}{l} \Leftarrow \text{ Index } \equiv 0 \bmod 2 \\ \\ \\ \\ \Leftarrow \text{ Index } \equiv 2 \bmod 3 \end{array}$$

- Reconstruct entry index via Chinese Remainder Theorem
- Two estimates of the entry's value

SAVED ONE LINEAR TEST!

Identification Example: One Fourier Coefficient

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Identification Example: One Fourier Coefficient

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \cdot \mathcal{F}_{6\times 6}\mathcal{F}^{-1}_{6\times 6} \cdot \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Identification Example: One Fourier Coefficient

$$\left(\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \mathcal{F}_{6\times 6} \right) \cdot \left(\mathcal{F}^{-1}_{6\times 6} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Identification Example: One Fourier Coefficient

$$\begin{pmatrix} \sqrt{\frac{3}{2}} & 0 & 0 & \sqrt{\frac{3}{2}} & 0 & 0 \\ \sqrt{\frac{3}{2}} & 0 & 0 & -\sqrt{\frac{3}{2}} & 0 & 0 \\ * & 0 & * & 0 & * & 0 \\ * & 0 & * & 0 & * & 0 \\ * & 0 & * & 0 & * & 0 \end{pmatrix} \cdot \left(\mathcal{F}^{-1}_{6\times 6} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Identification Example: One Fourier Coefficient

$$\begin{pmatrix} \sqrt{\frac{3}{2}} & 0 & 0 & \sqrt{\frac{3}{2}} & 0 & 0 \\ \sqrt{\frac{3}{2}} & 0 & 0 & -\sqrt{\frac{3}{2}} & 0 & 0 \\ * & 0 & * & 0 & * & 0 \\ * & 0 & * & 0 & * & 0 \\ * & 0 & * & 0 & * & 0 \end{pmatrix} \cdot \left(\mathcal{F}^{-1}_{6\times 6} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Identification Example: One Fourier Coefficient

$$\begin{pmatrix} \sqrt{3} \cdot \mathcal{F}_{2\times 2} \cdot \begin{pmatrix} 1 & \color{red}{0} & 0 & 0 & 0 & \color{red}{0} \\ 0 & \color{red}{0} & 0 & 1 & 0 & \color{red}{0} \end{pmatrix} \\ \sqrt{2} \cdot \mathcal{F}_{3\times 3} \cdot \begin{pmatrix} 1 & \color{red}{0} & 0 & 0 & 0 & \color{red}{0} \\ 0 & \color{red}{0} & 1 & 0 & 0 & \color{red}{0} \\ 0 & \color{red}{0} & 0 & 0 & 1 & \color{red}{0} \end{pmatrix} \end{pmatrix} \cdot \left(\mathcal{F}^{-1}_{6\times 6} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Identification Example: One Fourier Coefficient

$$\left(\begin{array}{c} \sqrt{3} \cdot \mathcal{F}_{2\times 2} \cdot \begin{pmatrix} 1 & \color{red}{0} & 0 & 0 & 0 & \color{red}{0} \\ 0 & \color{red}{0} & 0 & 1 & 0 & \color{red}{0} \end{pmatrix} \\ \sqrt{2} \cdot \mathcal{F}_{3\times 3} \cdot \begin{pmatrix} 1 & \color{red}{0} & 0 & 0 & 0 & \color{red}{0} \\ 0 & \color{red}{0} & 1 & 0 & 0 & \color{red}{0} \\ 0 & \color{red}{0} & 0 & 0 & 1 & \color{red}{0} \end{pmatrix} \end{array} \right) \cdot \left(\mathcal{F}^{-1}_{6\times 6} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Identification Example: One Fourier Coefficient

$$\left(\begin{array}{c} \sqrt{3} \cdot \mathcal{F}_{2\times 2} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \\ \sqrt{2} \cdot \mathcal{F}_{3\times 3} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \right) \cdot \left(\mathcal{F}^{-1}_{6\times 6} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Identification Example: One Fourier Coefficient

$$\left(\begin{array}{c} \sqrt{3} \cdot \mathcal{F}_{2\times 2} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \\ \sqrt{2} \cdot \mathcal{F}_{3\times 3} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \right) \cdot \left(\mathcal{F}^{-1}_{6\times 6} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Design Decision #3: Coefficient Estimation

Frequency Isolated in a Convolution

$$f_j(x) := \mathrm{Conv}[\mathrm{e}^{-\mathrm{i} n_j a x} g(x),\ \mathrm{e}^{\mathrm{i} m_{l_j} x} f(d_{l_j} x)](x) = C'_j \cdot \mathrm{e}^{x \cdot \omega'_j \cdot \mathrm{i}} + \epsilon(x)$$

1. Sometimes the procedure for identifying ω'_j automatically provides estimates of $C'_j \ldots$
2. If not, we can compute $C'_j \approx \mathrm{e}^{-x \cdot \omega'_j \cdot \mathrm{i}} f_j(x)$ if $\epsilon(x)$ small
3. Approximate C'_j via (Monte Carlo) integration techniques, e.g.,

$$C'_j \approx \int_0^{2\pi} \mathrm{e}^{-x \cdot \omega'_j \cdot \mathrm{i}}\ f_j(x)\ dx \approx \frac{1}{K} \sum_{h=1}^{K} \mathrm{e}^{-x_h \cdot \omega'_j \cdot \mathrm{i}}\ f_j(x_h)$$

What have we got so far?

Approximate $\left\{(\omega_j, C_j) \;\middle|\; 1 \le j \le k\right\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \ \Omega = \{\omega_1, \dots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

1. We can isolate (a proxy for) each $\omega_j \in \Omega$, in some

$$f_j(x) = \mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x),\ \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$

   for some n, m_l, d_l triple with high probability (w.h.p.).
2. We can identify ω_j by, e.g., doing a binary search on $\hat{f}_j$
3. We can get a good estimate of C_j from $f_j(x)$ once we know ω_j

We have a lot of estimates, $\left\{(\tilde{\omega}_j, \tilde{C}_j) \;\middle|\; 1 \le j \le c_1 k \log^{c_2} N\right\}$, which contain the true Fourier frequency/coefficient pairs.

How do we discard the junk?

What have we got so far?

Approximate $\left\{(\omega_j, C_j) \,\middle|\, 1 \le j \le k\right\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathrm{e}^{x \cdot \omega_j \cdot \mathrm{i}}, \; \Omega = \{\omega_1, \dots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

1. We can isolate (a proxy for) each $\omega_j \in \Omega$, in some

   $$f_j(x) = \mathrm{Conv}[\mathrm{e}^{-\mathrm{i}nax} g(x), \; \mathrm{e}^{\mathrm{i}m_l x} f(d_l x)](x)$$

   for some n, m_l, d_l triple with high probability (w.h.p.).
2. We can identify ω_j by, e.g., doing a binary search on $\hat{f}_j$
3. We can get a good estimate of C_j from $f_j(x)$ once we know ω_j

We have a lot of estimates, $\left\{(\tilde{\omega}_j, \tilde{C}_j) \,\middle|\, 1 \le j \le c_1 k \log^{c_2} N\right\}$, which contain the true Fourier frequency/coefficient pairs.

How do we discard the junk?

What have we got so far?

Approximate $\left\{(\omega_j, C_j) \mid 1 \leq j \leq k\right\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathrm{e}^{x \cdot \omega_j \cdot \mathrm{i}}, \; \Omega = \{\omega_1, \ldots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

1. We can isolate (a proxy for) each $\omega_j \in \Omega$, in some

$$f_j(x) = \mathrm{Conv}[\mathrm{e}^{-\mathrm{i}nax} g(x), \; \mathrm{e}^{\mathrm{i}m_l x} f(d_l x)](x)$$

   for some n, m_l, d_l triple with high probability (w.h.p.).
2. We can identify ω_j by, e.g., doing a binary search on $\hat{f}_j$
3. We can get a good estimate of C_j from $f_j(x)$ once we know ω_j

We have a lot of estimates, $\left\{(\tilde{\omega}_j, \tilde{C}_j) \mid 1 \leq j \leq c_1 k \log^{c_2} N\right\}$, which contain the true Fourier frequency/coefficient pairs.

How do we discard the junk?

What have we got so far?

Approximate $\left\{(\omega_j, C_j) \mid 1 \leq j \leq k\right\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \ \Omega = \{\omega_1, \ldots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

1. We can isolate (a proxy for) each $\omega_j \in \Omega$, in some

$$f_j(x) = \mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x), \ \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$

   for some n, m_l, d_l triple with high probability (w.h.p.).
2. We can identify ω_j by, e.g., doing a binary search on $\hat{f}_j$
3. We can get a good estimate of C_j from $f_j(x)$ once we know ω_j

We have a lot of estimates, $\left\{(\tilde{\omega}_j, \tilde{C}_j) \mid 1 \leq j \leq c_1 k \log^{c_2} N\right\}$, which contain the true Fourier frequency/coefficient pairs.

How do we discard the junk?

Design Decision #4: Iteration?

Approximate $\left\{(\omega_j, C_j) \;\middle|\; 1 \leq j \leq k\right\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \; \Omega = \{\omega_1, \dots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

- Analyzing probability of isolation is akin to considering tossing balls (frequencies of f) into bins (pass regions of modulated filter)

No Iteration: Identification and Estimation Once

Approximate $\{(\omega_j, C_j) \mid 1 \leq j \leq k\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \ \Omega = \{\omega_1, \dots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

1. Tossing the balls (frequencies) into $O(k)$ bins (pass regions) about $T = O(\log N)$-times guarantees that each ball lands in a bin "by itself" on the majority of tosses, w.h.p.
   - Translation: We should identify dominant frequency of

     $$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x), \ \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$

     for $O(\log N)$ random (m_l, d_l)-pairs, $\forall n \in O([-k, k])$.
2. Will identify each $\omega_j \in \Omega$ for $> T/2$ (m_l, d_l)-pairs w.h.p.
3. SO,...we can take medians of real/imaginary parts of C_j estimates for each frequency identified by $> T/2$ (m_l, d_l)-pairs as our final Fourier coefficient estimate for that frequency, and do fine

No Iteration: Identification and Estimation Once

Approximate $\left\{(\omega_j, C_j) \;\middle|\; 1 \le j \le k\right\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \; \Omega = \{\omega_1, \ldots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

1. Tossing the balls (frequencies) into $O(k)$ bins (pass regions) about $T = O(\log N)$-times guarantees that each ball lands in a bin "by itself" on the majority of tosses, w.h.p.
   - Translation: We should identify dominant frequency of

     $$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x), \; \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$

     for $O(\log N)$ random (m_l, d_l)-pairs, $\forall n \in O([-k, k])$.
2. Will identify each $\omega_j \in \Omega$ for $> T/2$ (m_l, d_l)-pairs w.h.p.
3. SO,... we can take medians of real/imaginary parts of C_j estimates for each frequency identified by $> T/2$ (m_l, d_l)-pairs as our final Fourier coefficient estimate for that frequency, and do fine

No Iteration: Identification and Estimation Once

Approximate $\left\{(\omega_j, C_j) \mid 1 \leq j \leq k\right\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \ \Omega = \{\omega_1, \ldots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

1. Tossing the balls (frequencies) into $O(k)$ bins (pass regions) about $T = O(\log N)$-times guarantees that each ball lands in a bin "by itself" on the majority of tosses, w.h.p.
   - Translation: We should identify dominant frequency of

     $$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x), \ \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$

     for $O(\log N)$ random (m_l, d_l)-pairs, $\forall n \in O([-k, k])$.
2. Will identify each $\omega_j \in \Omega$ for $> T/2$ (m_l, d_l)-pairs w.h.p.
3. SO,...we can take medians of real/imaginary parts of C_j estimates for each frequency identified by $> T/2$ (m_l, d_l)-pairs as our final Fourier coefficient estimate for that frequency, and do fine

No Iteration: Identification and Estimation Once

Approximate $\{(\omega_j, C_j) \mid 1 \leq j \leq k\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \ \Omega = \{\omega_1, \ldots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

1. Tossing the balls (frequencies) into $O(k)$ bins (pass regions) about $T = O(\log N)$-times guarantees that each ball lands in a bin "by itself" on the majority of tosses, w.h.p.
   - Translation: We should identify dominant frequency of

     $$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x),\ \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$

     for $O(\log N)$ random (m_l, d_l)-pairs, $\forall n \in O([-k, k])$.
2. Will identify each $\omega_j \in \Omega$ for $> T/2$ (m_l, d_l)-pairs w.h.p.
3. SO,...we can take medians of real/imaginary parts of C_j estimates for each frequency identified by $> T/2$ (m_l, d_l)-pairs as our final Fourier coefficient estimate for that frequency, and do fine

Several rounds of Identification and Estimation

Approximate $\left\{(\omega_j, C_j) \,\middle|\, 1 \leq j \leq k\right\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \; \Omega = \{\omega_1, \dots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

1. Tossing the balls (frequencies) into $O(k)$ bins (pass regions) about $O(T)$-times guarantees that each ball lands in a bin "by itself" at least once with probability $1 - 2^{-T}$
   - Idea: We should identify dominant frequency of

     $$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x),\; \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$

     for $O(1)$ random (m_l, d_l)-pairs, $\forall n \in O([-k, k])$.
   - We can expect to correctly identify a constant fraction of $\omega_1, \dots, \omega_k$
2. Accurately estimating the Fourier coefficients of the identified frequencies is comparatively easy (no binary search required)
3. As long as we estimate the Fourier coefficients of the energetic frequencies "well enough", we've made progress

Several rounds of Identification and Estimation

Approximate $\left\{(\omega_j, C_j) \,\middle|\, 1 \le j \le k\right\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \; \Omega = \{\omega_1, \dots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

1. Tossing the balls (frequencies) into $O(k)$ bins (pass regions) about $O(T)$-times guarantees that each ball lands in a bin "by itself" at least once with probability $1 - 2^{-T}$
   - Idea: We should identify dominant frequency of

     $$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x), \; \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$

     for $O(1)$ random (m_l, d_l)-pairs, $\forall n \in O([-k, k])$.
   - We can expect to correctly identify a constant fraction of $\omega_1, \dots, \omega_k$
2. Accurately estimating the Fourier coefficients of the identified frequencies is comparatively easy (no binary search required)
3. As long as we estimate the Fourier coefficients of the energetic frequencies "well enough", we've made progress

Several rounds of Identification and Estimation

Approximate $\left\{(\omega_j, C_j) \mid 1 \leq j \leq k\right\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathbb{e}^{x \cdot \omega_j \cdot \mathbb{i}}, \; \Omega = \{\omega_1, \ldots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

1. Tossing the balls (frequencies) into $O(k)$ bins (pass regions) about $O(T)$-times guarantees that each ball lands in a bin "by itself" at least once with probability $1 - 2^{-T}$
   - Idea: We should identify dominant frequency of

     $$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x), \; \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$

     for $O(1)$ random (m_l, d_l)-pairs, $\forall n \in O([-k, k])$.
   - We can expect to correctly identify a constant fraction of $\omega_1, \ldots, \omega_k$
2. Accurately estimating the Fourier coefficients of the identified frequencies is comparatively easy (no binary search required)
3. As long as we estimate the Fourier coefficients of the energetic frequencies "well enough", we've made progress

Several rounds of Identification and Estimation

Approximate $\left\{(\omega_j, C_j) \mid 1 \leq j \leq k\right\}$ by sampling

$$f(x) \approx \sum_{j=1}^{k} C_j \cdot \mathrm{e}^{x \cdot \omega_j \cdot \mathrm{i}}, \; \Omega = \{\omega_1, \ldots, \omega_k\} \subset \left(-\frac{N}{2}, \frac{N}{2}\right] \bigcap \mathbb{Z}$$

1. Tossing the balls (frequencies) into $O(k)$ bins (pass regions) about $O(T)$-times guarantees that each ball lands in a bin "by itself" at least once with probability $1 - 2^{-T}$
   - Idea: We should identify dominant frequency of
   $$\mathrm{Conv}[\mathrm{e}^{-\mathrm{i}nax} g(x), \; \mathrm{e}^{\mathrm{i}m_l x} f(d_l x)](x)$$
   for $O(1)$ random (m_l, d_l)-pairs, $\forall n \in O([-k, k])$.
   - We can expect to correctly identify a constant fraction of $\omega_1, \ldots, \omega_k$
2. Accurately estimating the Fourier coefficients of the identified frequencies is comparatively easy (no binary search required)
3. As long as we estimate the Fourier coefficients of the energetic frequencies "well enough", we've made progress

Round 2

1. If we made progress the first time, so we should do it again ...

Implicitly Create a "New Signal"

$$f^2(x) := f(x) - \sum_{j=1}^{O(k)} \tilde{C}_j \cdot \mathbb{e}^{x \cdot \tilde{\omega}_j \cdot \mathbb{i}} \approx \sum_{j=1}^{k/4} C'_j \cdot \mathbb{e}^{x \cdot \omega'_j \cdot \mathbb{i}},$$

where $(\tilde{\omega}_j, \tilde{C}_j)$ where obtained from the last round

2. Sparsity is effectively reduced. Repeat...

Round j

1. Tossing the remaining $k/4^j$ balls (frequencies) into $O(k/4^j)$ bins (pass regions) about $O(j)$-times guarantees that each remaining ball lands in a bin "by itself" at least once with probability $1 - 2^{-j}$
   - We should identify dominant frequencies of
     $$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x),\ \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$
     for $O(j)$ random (m_l, d_l)-pairs, $\forall n \in O([-k/4^j, k/4^j])$.
   - We identify a constant fraction of remaining frequencies, $\omega'_1, \dots, \omega'_{k/4^j}$, with higher probability
2. Estimating Fourier coefficients of identified frequencies can be done more accurately (e.g., w/ relative error $O(2^{-j})$)

$\vdots$

3. We eventually find all of $\omega_1, \dots, \omega_k$ with high probability after $O(\log k)$-rounds. Samples/runtime will be dominated by first round IF. . . .

Round j

1. Tossing the remaining $k/4^j$ balls (frequencies) into $O(k/4^j)$ bins (pass regions) about $O(j)$-times guarantees that each remaining ball lands in a bin "by itself" at least once with probability $1 - 2^{-j}$
   - We should identify dominant frequencies of
   $$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax} g(x),\ \mathbb{e}^{\mathbb{i}m_l x} f(d_l x)](x)$$
   for $O(j)$ random (m_l, d_l)-pairs, $\forall n \in O([-k/4^j, k/4^j])$.
   - We identify a constant fraction of remaining frequencies, $\omega'_1, \ldots, \omega'_{k/4^j}$, with higher probability
2. Estimating Fourier coefficients of identified frequencies can be done more accurately (e.g., w/ relative error $O(2^{-j})$)

$\vdots$

3. We eventually find all of $\omega_1, \ldots, \omega_k$ with high probability after $O(\log k)$-rounds. Samples/runtime will be dominated by first round IF....

Round j

1. Tossing the remaining $k/4^j$ balls (frequencies) into $O(k/4^j)$ bins (pass regions) about $O(j)$-times guarantees that each remaining ball lands in a bin "by itself" at least once with probability $1 - 2^{-j}$
   - We should identify dominant frequencies of
   $$\mathrm{Conv}[\mathrm{e}^{-\mathrm{i}nax} g(x),\ \mathrm{e}^{\mathrm{i}m_l x} f(d_l x)](x)$$
   for $O(j)$ random (m_l, d_l)-pairs, $\forall n \in O([-k/4^j, k/4^j])$.
   - We identify a constant fraction of remaining frequencies, $\omega'_1, \ldots, \omega'_{k/4^j}$, with higher probability
2. Estimating Fourier coefficients of identified frequencies can be done more accurately (e.g., w/ relative error $O(2^{-j})$)

⋮

3. We eventually find all of $\omega_1, \ldots, \omega_k$ with high probability after $O(\log k)$-rounds. Samples/runtime will be dominated by first round IF....

Round j

1. Tossing the remaining $k/4^j$ balls (frequencies) into $O(k/4^j)$ bins (pass regions) about $O(j)$-times guarantees that each remaining ball lands in a bin "by itself" at least once with probability $1 - 2^{-j}$
   - We should identify dominant frequencies of

$$\mathrm{Conv}[\mathbb{e}^{-\mathbb{i}nax}g(x),\ \mathbb{e}^{\mathbb{i}m_l x}f(d_l x)](x)$$

   for $O(j)$ random (m_l, d_l)-pairs, $\forall n \in O([-k/4^j, k/4^j])$.
   - We identify a constant fraction of remaining frequencies, $\omega'_1, \ldots, \omega'_{k/4^j}$, with higher probability
2. Estimating Fourier coefficients of identified frequencies can be done more accurately (e.g., w/ relative error $O(2^{-j})$)

$$\vdots$$

3. We eventually find all of $\omega_1, \ldots, \omega_k$ with high probability after $O(\log k)$-rounds. Samples/runtime will be dominated by first round <u>IF</u>....

We Can Quickly Sample From Residual Signal

The Residual Signal We Need to Sample

$$f^j(x) := f(x) - \sum_{h=1}^{O(k)} \tilde{C}_h \cdot \mathrm{e}^{x \cdot \tilde{\omega}_h \cdot \mathrm{i}} \approx \sum_{h=1}^{k/4^j} C'_h \cdot \mathrm{e}^{x \cdot \omega'_h \cdot \mathrm{i}},$$

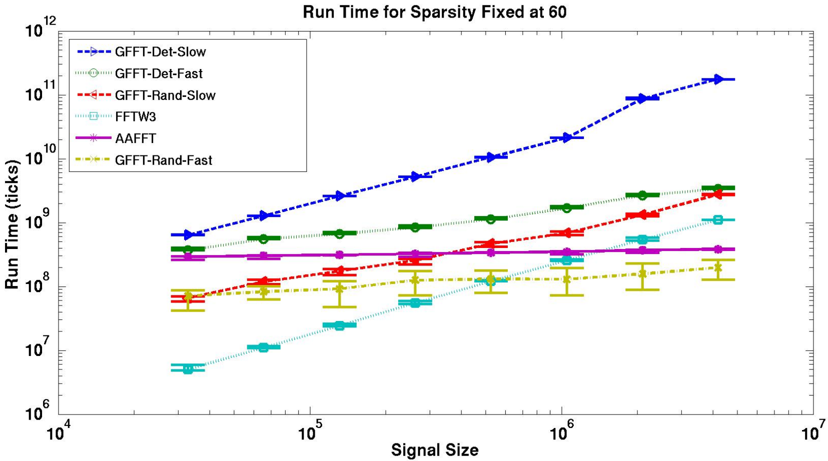
where $(\tilde{\omega}_h, \tilde{C}_h)$ where obtained from the previous rounds

- Subtracting Fourier terms from previous rounds, $(\tilde{\omega}_h, \tilde{C}_h)$, from each "frequency bin" they fall into
  - We know what filter's pass region each $\tilde{\omega}_h$ will fall into (e.g., call it n_h). Subtract $\tilde{C}_h$ from the Fourier transform of

$$\mathrm{Conv}[\mathrm{e}^{-\mathrm{i} n_h a x} g(x),\ \mathrm{e}^{\mathrm{i} m_l x} f(d_l x)](x)$$

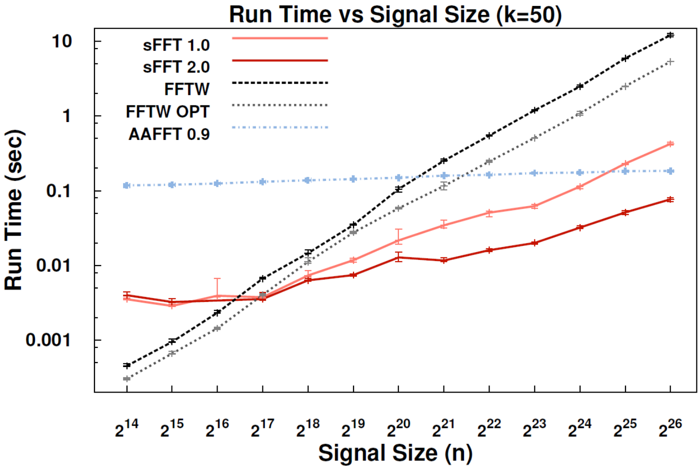
    for each (m_l, d_l)-pair during subsequent rounds.
- Or, we can use nonequispaced FFT ideas (several grids on arithmetic progressions, frequencies nonequispaced).

Publicly Available Codes: FFTW, AAFFT, and GFFT



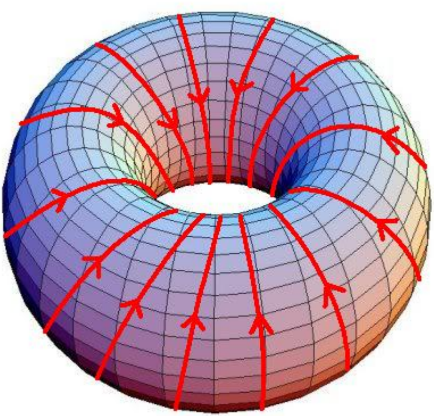
- FFTW: `http://www.fftw.org`
- AAFFT, GFFT: `http://sourceforge.net/projects/gopherfft/`

Publicly Available Codes: SFT 1.0 and 2.0



- `http://groups.csail.mit.edu/netmit/sFFT/code.html`

Extending to Many Dimensions



- Sample $f^{\text{new}}(x) = f\left(x\frac{\tilde{N}}{P_1}, \dots, x\frac{\tilde{N}}{P_D}\right)$, with $\tilde{N} = \prod_{d=1}^{D} P_d > N^D$
- Works because $\mathbb{Z}_{\tilde{N}}$ is isomorphic to $\mathbb{Z}_{P_1} \times \cdots \times \mathbb{Z}_{P_D}$.

Questions?

Thank You!

References

- [CT'65]: J. Cooley, J. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.*, 19, 1965.
- [FJ' 05]: M. Frigo, S. Johnson. The design and implementation of fftw3. *Proceedings of the IEEE*, 93:2, 2005.
- [M' 95]: Y. Mansour. Randomized approxmation and interpolation of sparse polynomials. *SIAM Journal on Computing*, 24:2, 1995.
- [AGS' 03]: A. Akavia, S. Goldwasser, and S. Safra. Proving hard-core predicates using list decoding. In *Annual Symposium on Foundations of Computer Science*, vol. 44, 2003.
- [GGIMS' 02]: A. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss. Near-optimal sparse Fourier estimation via sampling. *ACM STOC*, 2002.
- [GMS' 05]: A. Gilbert, S. Muthukrishnan, and M. Strauss. Improved Time Bounds for Near-Optimal Sparse Fourier Representations, *Proceedings of SPIE Wavelets XI*, 2005.
- [I' 13]: M. Iwen. Improved Approximation Guarantees for Sublinear-Time Fourier Algorithms. *Applied and Computational Harmonic Analysis*, 34:1, 2013.
- [SI' 12]: I. Segal and M. Iwen. Improved sparse fourier approximation results: Faster implementations and stronger guarantees. *Numerical Algorithms*, 2012.
- [HIKPs'12]: H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Simple and Practical Algorithm for Sparse Fourier Transform. *SODA*, 2012.
- [HIKPst'12]: H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Nearly Optimal Sparse Fourier Transform. *STOC*, 2012.