CS-GY 6763: Lecture 14
Fast Johnson-Lindenstrauss Transform, Introduction to Sparse Recovery and Compressed Sensing

NYU Tandon School of Engineering, Prof. Christopher Musco

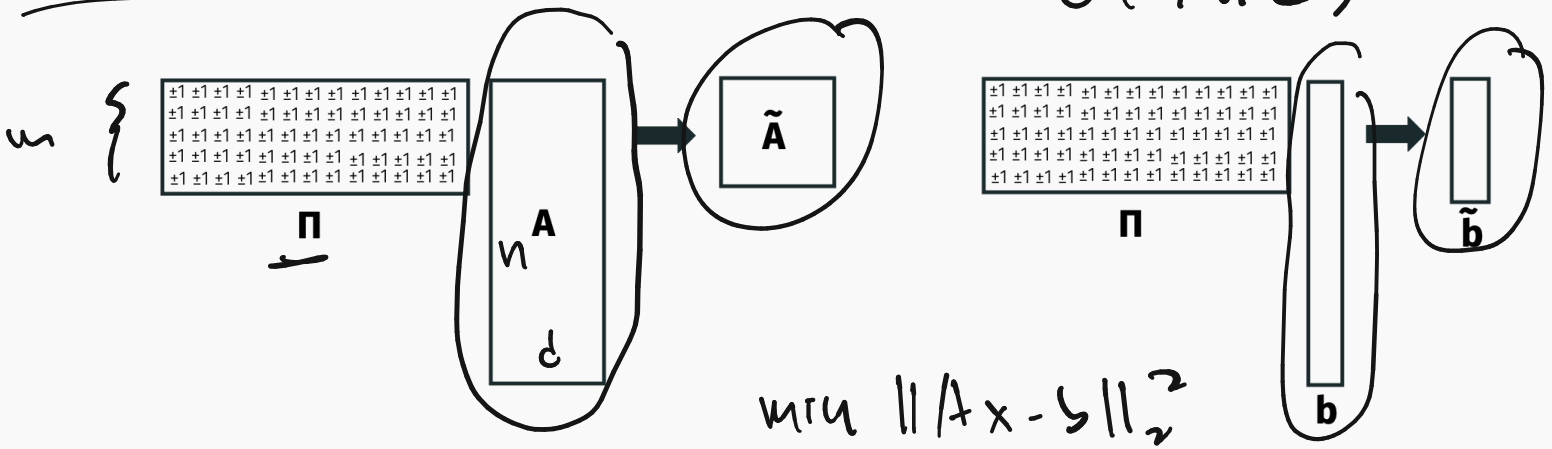
ADMINISTRATIVE

This is our last class!

- Final project due next Friday.
- Exam study guide will be released shortly with practice questions. Same rules as midterm (cheat sheet allowed). will be a 1.5 hour test.
- Solutions for last problem set will be released day after it's due (so no late submissions).

Main idea: Speed up (classical linear algebra problems) using randomization.

$O(mnd) \gg O(nd^2)$



$\min_x \|Ax - b\|_2^2$

$(A^TA)^{-1}A^Tb$

$O(nd^2)$

Input: $\mathbf{A} \in \mathbb{R}^{n \times d}$, $\mathbf{b} \in \mathbb{R}^n$.

Algorithm: Let $\tilde{\mathbf{x}}^* = \arg\min_{\mathbf{x}} \|\mathbf{\Pi A x} - \mathbf{\Pi b}\|_2^2$.

Goal: Want $\|\mathbf{A}\tilde{\mathbf{x}}^* - \mathbf{b}\|_2^2 \leq (1+\epsilon)\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$

$m = O\left(\frac{d}{\epsilon}\right)$

Theorem (Example: Randomized Linear Regression)

Let $\mathbf{\Pi}$ *be a properly scaled JL matrix (random Gaussian, sign, sparse random, etc.) with* $m = O\left(\frac{d}{\epsilon^2}\right)$ *rows. Then with probability* $9/10$*, for any* $\mathbf{A} \in \mathbb{R}^{n \times d}$ *and* $\mathbf{b} \in \mathbb{R}^n$*,*

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2^2 \leq (1+\epsilon)\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|_2^2$$

where $\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{\Pi}\mathbf{A}\mathbf{x} - \mathbf{\Pi}\mathbf{b}\|_2^2$.

Reduce from a $O(nd^2)$ time computation to an $O(d^3)$ time problem.

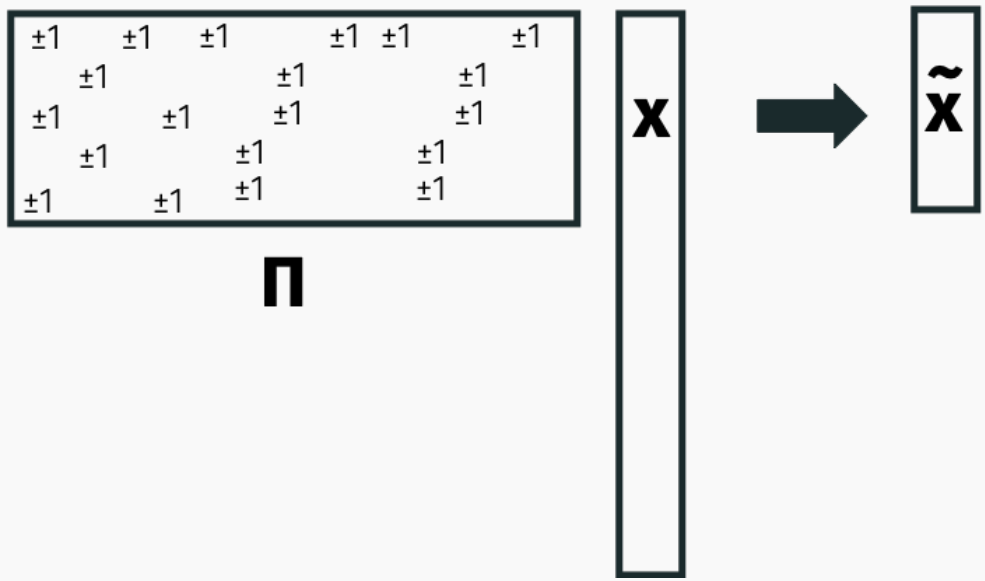
$(\Pi A)^T \Pi A)^{-1} (\Pi A)^T \Pi b \quad O(d^3)$

Issue discussed last time: The dimensionality reduction itself takes $O(nd^2)$ time!

RETURN TO SINGLE VECTOR PROBLEM

Goal: Develop methods that reduce a vector $\mathbf{x} \in \mathbb{R}^n$ down to $m \approx \frac{\log(1/\delta)}{\epsilon^2}$ dimensions in $o(mn)$ time and guarantee: w.p. $1-\delta$

$$(1-\epsilon)\|\mathbf{x}\|_2^2 \leq \|\mathbf{\Pi x}\|_2^2 \leq (1+\epsilon)\|\mathbf{x}\|_2^2$$



$< mn$

Recall that once the bound above is proven, linearity lets us preserve things like $\|\mathbf{y}-\mathbf{z}\|_2^2$ or $\|\mathbf{Ax}-\mathbf{b}\|_2^2$ for all $\mathbf{x}$. I.e., we get Johnson-Lindenstrauss and subspace embeddings for free.

RETURN TO SINGLE VECTOR PROBLEM

Fast embeddings are useful in a lot of other applications.

- (Nearest-neighbor search)(locality sensitive hash functions for for cosine similarity and ℓ_2 starts with JL sketch).
  - Used in FALCONN ANN library, Reformer fast attention architecture, etc.)
- (Key/query vector compression in LLMs (needs to happen very fast).
- Random Fourier features and other methods in machine learning that use JL as a starting point.

Subsampled Randomized Hadamard Transform[1] (SHRT)
(Ailon-Chazelle, 2006)

Theorem (The Fast JL Lemma)

Let $\mathbf{\Pi} = \mathbf{SHD} \in \mathbb{R}^{m \times n}$ *be a subsampled randomized Hadamard transform with* $m = O\left(\frac{\log(n/\delta)\log(1/\delta)}{\epsilon^2}\right)$ *rows. Then for any fixed* $\mathbf{x}$,

$$(1-\epsilon)\|\mathbf{x}\|_2^2 \leq \|\mathbf{\Pi x}\|_2^2 \leq (1+\epsilon)\|\mathbf{x}\|_2^2$$

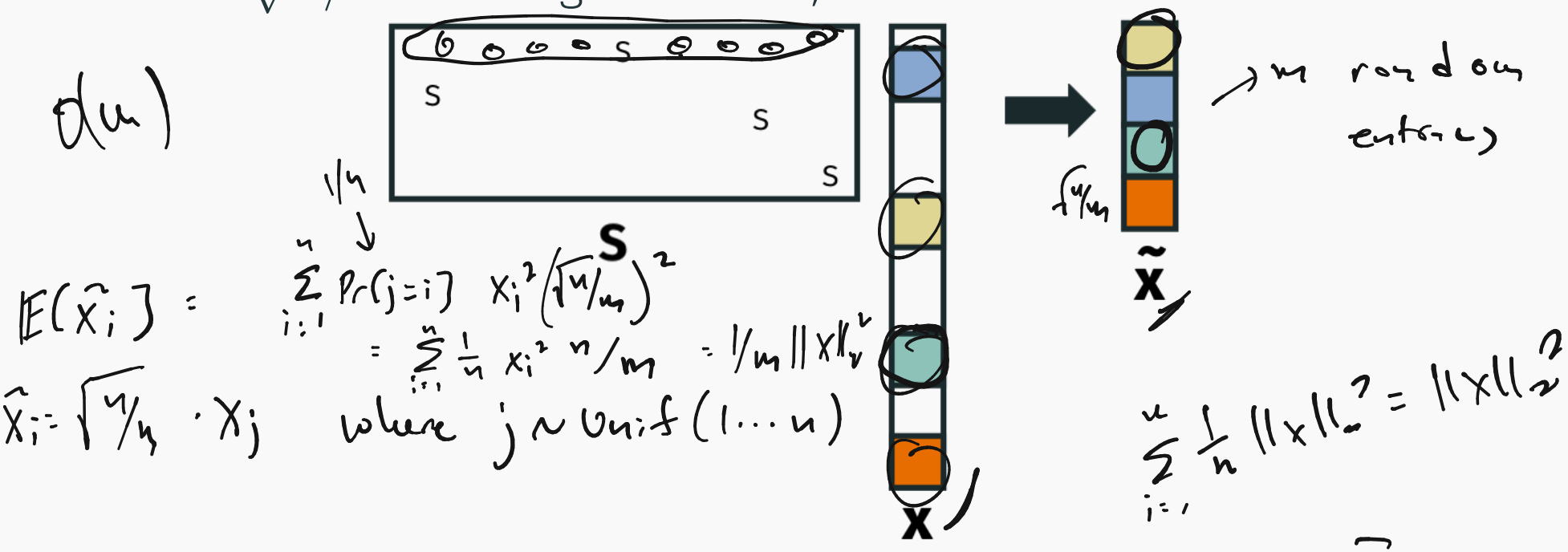
with probability $(1-\delta)$ *and* $\mathbf{\Pi x}$ *can be computed in* $O(n \log n)$ *(nearly linear) time.*

Very little loss in embedding dimension compared to standard JL.

[1] One of my favorite randomized algorithms.

SOLUTION FOR "FLAT" VECTORS

Let $\mathbf{S}$ be a random sampling matrix. Every row contains a value of $s = \sqrt{n/m}$ in a single location, and is zero elsewhere.



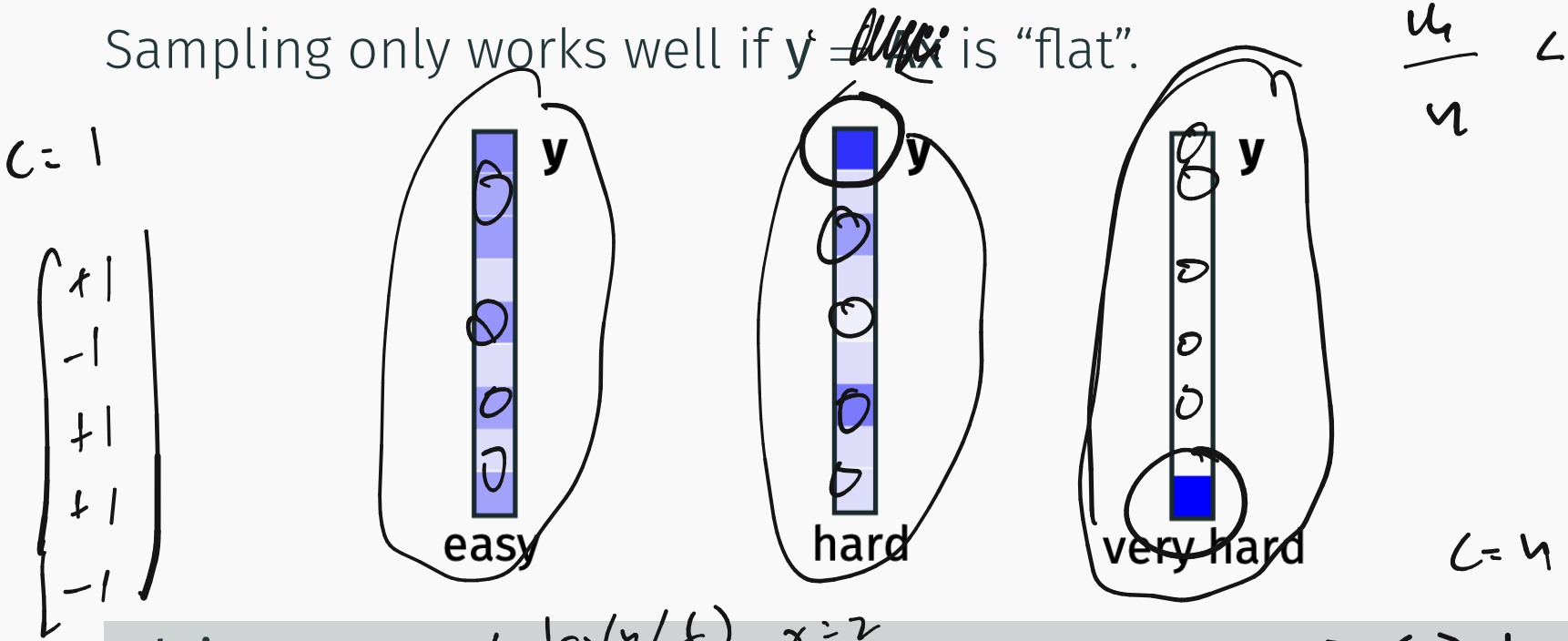
$\tilde{\mathbf{x}}$ can be computed in $O(m)$ time. Woohoo!

$$\mathbb{E}[\|\mathbf{S}\mathbf{x}\|_2^2] =$$

What is the problem with this approach?

VECTOR SAMPLING

Sampling only works well if $\mathbf{y} = \mathbf{Ux}$ is "flat".



Claim

If $\mathbf{x}_i^2 \leq \frac{\alpha}{n}\|\mathbf{x}\|_2^2$ *for all* i *then* $m = O(\alpha \log(1/\delta)/\epsilon^2)$ *samples suffices to ensure the* $(1-\epsilon)\|\mathbf{x}\|_2^2 \leq \|\mathbf{Sx}\|_2^2 \leq (1+\epsilon)\|\mathbf{x}\|_2^2$ *with probability* $1-\delta$.

This just follows from standard Hoeffding inequality.

THE FAST JOHNSON-LINDENSTRAUSS TRANSFORM

Key idea: First multiply $\mathbf{x}$ by a "mixing matrix" $\mathbf{M}$ which ensures it cannot be too concentrated in one place.

$\mathbf{M}$ will have the properties that

$M^T M = I$

1. $\|\mathbf{Mx}\|_2^2 = \|\mathbf{x}\|_2^2$ exactly.
2. Every entry in $\mathbf{Mx}$ is bounded. I.e. $[\mathbf{Mx}]_i^2 \leq \frac{c}{n}\|\mathbf{Mx}\|_2^2$ for some factor c to be determined.
3. We will be able to multiply by $\mathbf{M}$ in $O(n \log n)$ time.

Then we will multiply by a subsampling matrix $\mathbf{S}$ to do the actual dimensionality reduction:

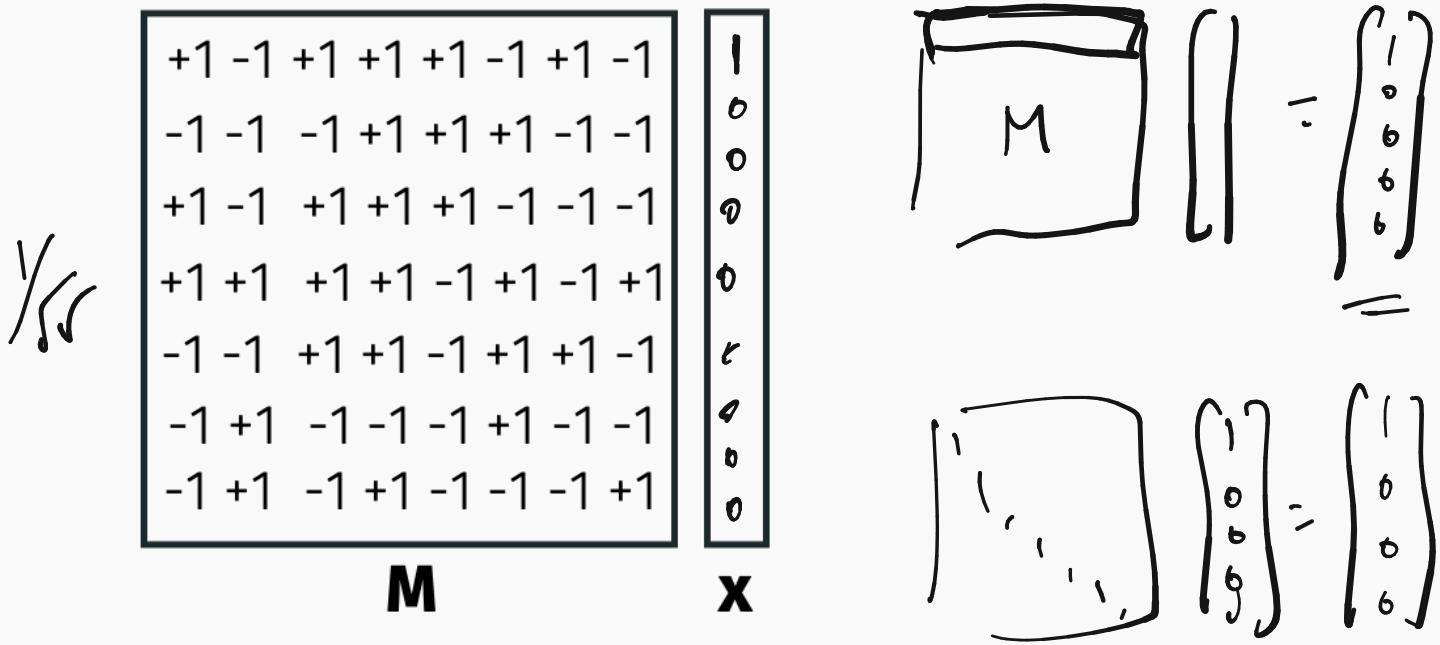
$+ O(n)$ time

$M^T M = I$

$$\mathbf{\Pi x} = \mathbf{SMx}$$

$\frac{\log(1/\delta)}{\epsilon^2}$

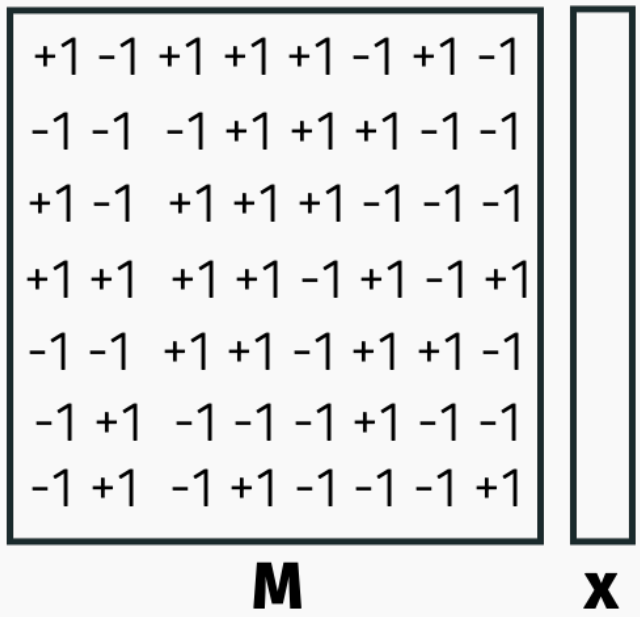
Good mixing matrices should look random:



In fact, I claim to mix any $\mathbf{x}$ with high probability, $\mathbf{M}$ needs to be chosen randomly. Why?

Hint: Recall that $\|\mathbf{Mx}\|_2 = \|\mathbf{x}\|_2$, so $\mathbf{M}$ is orthogonal.

Good mixing matrices should look random:

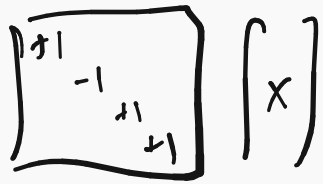


But for this approach to work, we need to be able to compute $\mathbf{Mx}$ very quickly. So we will use a pseudorandom matrix instead.

THE FAST JOHNSON-LINDENSTRAUSS TRANSFORM

(Subsampled)(Randomized Hadamard Transform)

$\mathbf{\Pi} = \mathbf{SM}$ where $\mathbf{M} = \mathbf{HD}$:



Mx in $O(n \log n)$

- $\mathbf{D} \in n \times n$ is a diagonal matrix with each entry uniform ± 1.
- $\mathbf{H} \in n \times n$ is a Hadamard matrix. $H^T H = I$

The Hadarmard matrix is an orthogonal matrix closely related to the discrete Fourier matrix. It has three critical properties:

1. $\|\mathbf{Hv}\|_2^2 = \|\mathbf{v}\|_2^2$ exactly. Thus $\|\mathbf{HDx}\|_2^2 = \|\mathbf{x}\|_2^2$ (M)
2. $\|\mathbf{Hv}\|_2^2$ can be computed in $O(n \log n)$ time.
3. All of the entries in $\mathbf{H}$ have the same magnitude. I.e. the matrix is "flat".

HADAMARD MATRICES RECURSIVE DEFINITION

Assume that n is a power of 2. For $k = 0, 1, \ldots,$ the k^{th} Hadamard matrix $\mathbf{H}_k$ is a $2^k \times 2^k$ matrix defined by:

$$H_0 = 1 \qquad H_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad H_2 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$H_k = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix}$$

The $n \times n$ Hadamard matrix has all entries as $\pm \frac{1}{\sqrt{n}}$.

HADAMARD MATRICES ARE ORTHOGONAL

Property 1: For any $k = 0, 1, \ldots$, we have $\|\mathbf{H}_k \mathbf{v}\|_2^2 = \|\mathbf{v}\|_2^2$ for all $\mathbf{v}$.
I.e., $\mathbf{H}_k$ is orthogonal.

Induction: assume H_{n-1} is orthogonal

$\rightarrow H_{n-1}^T H_{n-1} = I_{2^{n-1}}$

$$H_n^T H_n = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{n-1}^T & H_{n-1}^T \\ H_{n-1}^T & -H_{n-1}^T \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} H_{n-1}^T H_{n-1} + H_{n-1}^T H_{n-1} & H_{n-1}^T H_{n-1} - H_{n-1}^T H_{n-1} \\ H_{n-1}^T H_{n-1} - H_{n-1}^T H_{n-1} & H_{n-1}^T H_{n-1} + H_{n-1}^T H_{n-1} \end{bmatrix}$$

$$= \frac{1}{2} \begin{bmatrix} 2I & 0 \\ 0 & 2I \end{bmatrix} = \boxed{I}.$$

want $= I$

Property 2: Can compute $\Pi x = SHDx$ in $O(n \log n)$ time.

Suffices to show we can compute Hv in $O(n \log n)$ time for any v.

$$H_n v = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{n-1} v_1 + H_{n-1} v_2 \\ H_{n-1} v_1 - H_{n-1} v_2 \end{bmatrix}$$

(2)

1) Multiply $H_{n-1} v_1$
2) Multiply $H_{n-1} v_2$
3) Some additions $O(n)$

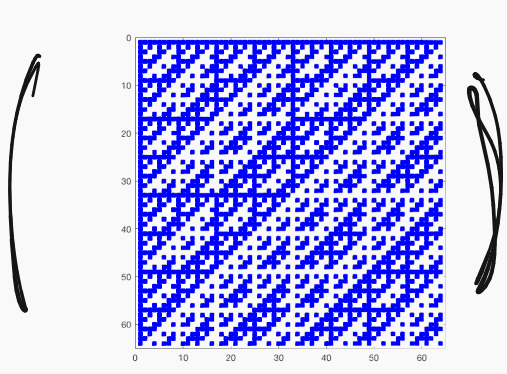
$T(n)$ = cost of multiplying $n \times n$ Hadamard by a vector.

$T(n) = 2 \cdot T(n/2) + cn$

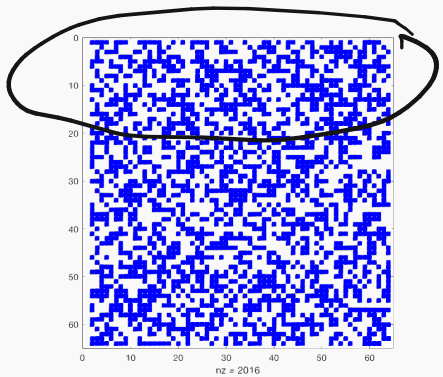
$T(n) = O(n \log(n))$

RANDOMIZED HADAMARD TRANSFORM

Property 3: The randomized Hadamard matrix is a good "mixing matrix" for smoothing out vectors.



Deterministic Hadamard matrix.



Randomized Hadamard **PHD**.

Fully random sign matrix.

Blue squares are $1/\sqrt{n}$'s, white squares are $-1/\sqrt{n}$'s.

Pseudorandom objects like this appear all the time in computer science! Error correcting codes, efficient hash functions, etc.

Lemma (SHRT mixing lemma)

Let $\mathbf{H}$ *be an* $(n \times n)$ *Hadamard matrix and* $\mathbf{D}$ *a random* ± 1 *diagonal matrix. Let* $\mathbf{z} = \mathbf{HDx}$ *for* $\mathbf{x} \in \mathbb{R}^n$*. With probability* $1 - \delta$*, for all* i *simultaneously,*

$$z_i^2 \leq \frac{c \log(n/\delta)}{n} \|\mathbf{z}\|_2^2$$

for some fixed constant c*.*

The vector is very close to uniform with high probability. As we saw earlier, we can thus argue that $\|\mathbf{Sz}\|_2^2 \approx \|\mathbf{z}\|_2^2$. I.e. that:

$$\|\mathbf{\Pi x}\|_2^2 = \|\mathbf{SHDx}\|_2^2 \approx \|\mathbf{x}\|_2^2$$

The main result then follows directly from our sampling result from earlier:

Theorem (The Fast JL Lemma)

Let $\mathbf{\Pi} = \mathbf{SHD} \in \mathbb{R}^{m \times n}$ *be a subsampled randomized Hadamard transform with* $m = O\left(\frac{\log(n/\delta)\log(1/\delta)}{\epsilon^2}\right)$ *rows. Then for any fixed* $\mathbf{x}$*,*

$$(1-\epsilon)\|\mathbf{x}\|_2^2 \leq \|\mathbf{\Pi x}\|_2^2 \leq (1+\epsilon)\|\mathbf{x}\|_2^2$$

with probability $(1-\delta)$*.*

SHRT mixing lemma proof: Need to prove $(z_i)^2 \leq \frac{c \log(n/\delta)}{n} \|\mathbf{z}\|_2^2$.

Recall that here $\mathbf{z} = \mathbf{HDx}$, so this is equivalent to proving:

$$(z_i)^2 \leq \frac{c \log(n/\delta)}{n} \|\mathbf{x}\|_2^2$$

$$|z_i| \leq \sqrt{\frac{c \log(n/\delta)}{n}} \|\mathbf{x}\|_2.$$

(Claim: z_i is a sum of i.i.d. random variables.

Let $\mathbf{h}_i^T$ be the i^{th} row of $\mathbf{H}$. $z_i = \mathbf{h}_i^T \mathbf{D}\mathbf{x}$ where:

$$\mathbf{h}_i^T \mathbf{D} = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & \dots & -1 & -1 \end{bmatrix} \begin{bmatrix} D_1 & & & & \\ & D_2 & & & \\ & & \ddots & & \\ & & & & D_n \end{bmatrix}$$

where $D_1, \dots, D_n$ are random ± 1's.

This is equivalent to

$$\mathbf{h}_i^T \mathbf{D} = \frac{1}{\sqrt{n}} \begin{bmatrix} R_1 & R_2 & \dots & R_n \end{bmatrix}$$

where $R_1, \dots, R_n$ are random ± 1's.

So we have, for all i, $z_i = \mathbf{h}_i^T \mathbf{D}\mathbf{x} = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} R_i x_i$.

- z_i is a random variable with mean 0 and variance $\frac{1}{n}\|\mathbf{x}\|_2^2$, which is a ~~sum of~~ independent random variables.

$$z_i = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} B_i x_i$$

$$\mathbb{E}[z_i] = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} x_i \mathbb{E}[B_i] = 0$$

$$\mathrm{Var}[z_i] = \frac{1}{n} \sum_{i=1}^{n} x_i^2 \underbrace{\mathrm{Var}[B_i]}_{=1} = \frac{1}{n} \sum_{i=1}^{n} x_i^2 = \frac{1}{n} \|x\|_2^2$$

$$z_i \approx \frac{1}{\sqrt{n}} \|x\|_2$$

z_i is a random variable with mean 0 and variance $\frac{1}{n}\|\mathbf{x}\|_2^2$, which is a sum of independent random variables.

- By Central Limit Theorem, we expect that:

$$\Pr[|z_i| \geq t \cdot \frac{\|\mathbf{x}\|_2}{\sqrt{n}}] \leq e^{-O(t^2)}.$$

- Setting $t = \sqrt{\log(n/\delta)}$, we have for constant c,

$$\Pr\left[|z_i| \geq c\sqrt{\frac{\log(n/\delta)}{n}}\|\mathbf{x}\|_2\right] \leq \frac{\delta}{n}$$

.

- Applying a union bound to all n entries of $\mathbf{z}$ gives the SHRT mixing lemma.

Can use Bernstein-type concentration inequality to prove the bound:

Lemma (Rademacher Concentration)

Let $R_1, \ldots, R_n$ be Rademacher random variables (i.e. uniform ± 1's). Then for any vector $\mathbf{a} \in \mathbb{R}^n$,

$$\Pr\left[\sum_{i=1}^{n} R_i a_i \geq t\|\mathbf{a}\|_2\right] \leq e^{-t^2/2}.$$

This is called the Khintchine Inequality. It is specialized to sums of scaled ± 1's, and is a bit tighter and easier to apply than using a generic Bernstein bound.

Recall that $\mathbf{z} = \mathbf{HDx}$.

With probability $1 - \delta$, we have that for all i,

$$|z_i| \leq \sqrt{\frac{c \log(n/\delta)}{n}} \|\mathbf{x}\|_2 = \sqrt{\frac{c \log(n/\delta)}{n}} \|\mathbf{z}\|_2.$$

As shown earlier, we can thus guarantee that:

$$(1-\epsilon)\|\mathbf{z}\|_2^2 \leq \|\mathbf{Sz}\|_2^2 \leq (1+\epsilon)\|\mathbf{z}\|_2^2$$

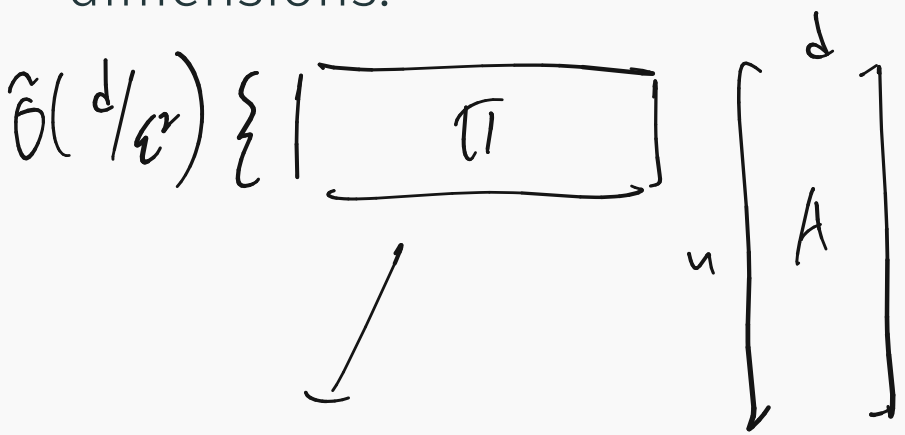
as long as $\mathbf{S} \in \mathbb{R}^{m \times n}$ is a random sampling matrix with

$$m = O\left(\frac{\log(n/\delta)\log(1/\delta)}{\epsilon^2}\right) \text{ rows.}$$

$\|\mathbf{Sz}\|_2^2 = \|\mathbf{SHDx}\|_2^2 = \|\mathbf{\Pi x}\|_2^2$ and $\|\mathbf{z}\|_2^2 = \|\mathbf{x}\|_2^2$, so we are done.

LINEAR REGRESSION WITH SHRTs

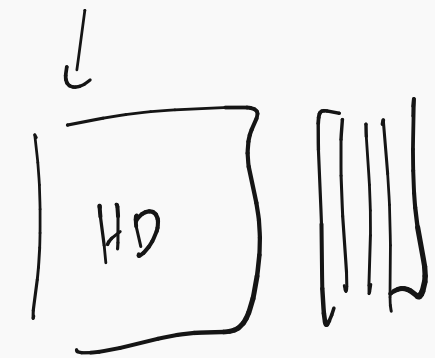
Upshot for regression: Compute $\mathbf{\Pi A}$ in $O(nd \log n)$ time instead of $O(nd^2)$ time. Compress problem down to $\tilde{\mathbf{A}}$ with $O(d^2)$ dimensions.



$$= O(d \cdot n \log(n))$$

$$= O(nd \log(\cdot))$$

$$+ O(d^2)$$



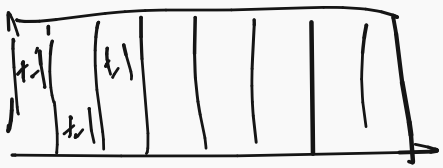
BRIEF COMMENT ON OTHER METHODS

$O(nd \log n)$ is nearly linear in the size of $\mathbf{A}$ when $\mathbf{A}$ is dense.

$O(nd)$

Clarkson-Woodruff 2013, STOC Best Paper: Let $O(\text{nnz}(\mathbf{A}))$ be the number of non-zeros in $\mathbf{A}$. It is possible to compute $\tilde{\mathbf{A}}$ with $\text{poly}(d)$ rows in:

$$O(\text{nnz}(\mathbf{A})) \text{ time.}$$



(Π is chosen to be an ultra-sparse random matrix. Uses totally different techniques (you can't do JL + ϵ-net).

Lead to a whole close of matrix algorithms (for regression, SVD, etc.) which run in time:

$$O(\text{nnz}(\mathbf{A})) + \text{poly}(d, \epsilon).$$

WHAT WERE AILON AND CHAZELLE THINKING?

Simple, inspired algorithm that has been used for accelerating:

- Vector dimensionality reduction
- Linear algebra
- Locality sensitive hashing (SimHash)
- Randomized kernel learning methods.

```
m = 20;
c1 = (2*randi(2,1,n)-3).*y;
c2 = sqrt(n)*fwht(dy);
c3 = c2(randperm(n));
z = sqrt(n/m)*c3(1:m);
```

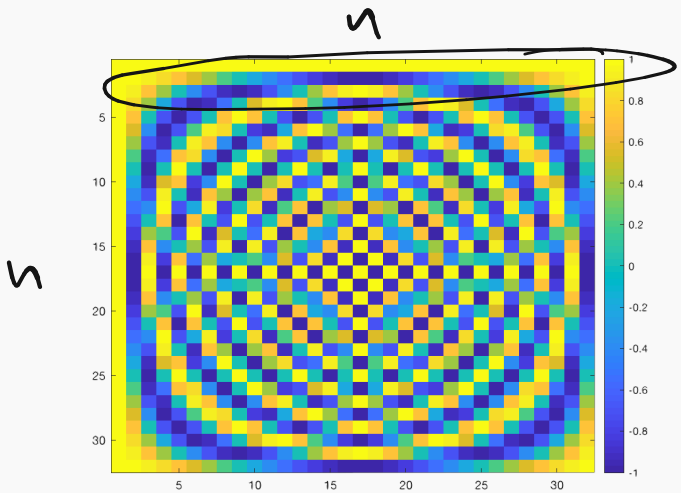
3:25 pm

BREAK

WHAT WERE AILON AND CHAZELLE THINKING?

The Hadamard Transform is closely related to the Discrete Fourier Transform.

$$\mathsf{F}_{j,k} = e^{-2\pi i \frac{j \cdot k}{n}}, \qquad \mathsf{F}^* \mathsf{F} = \mathsf{I}.$$

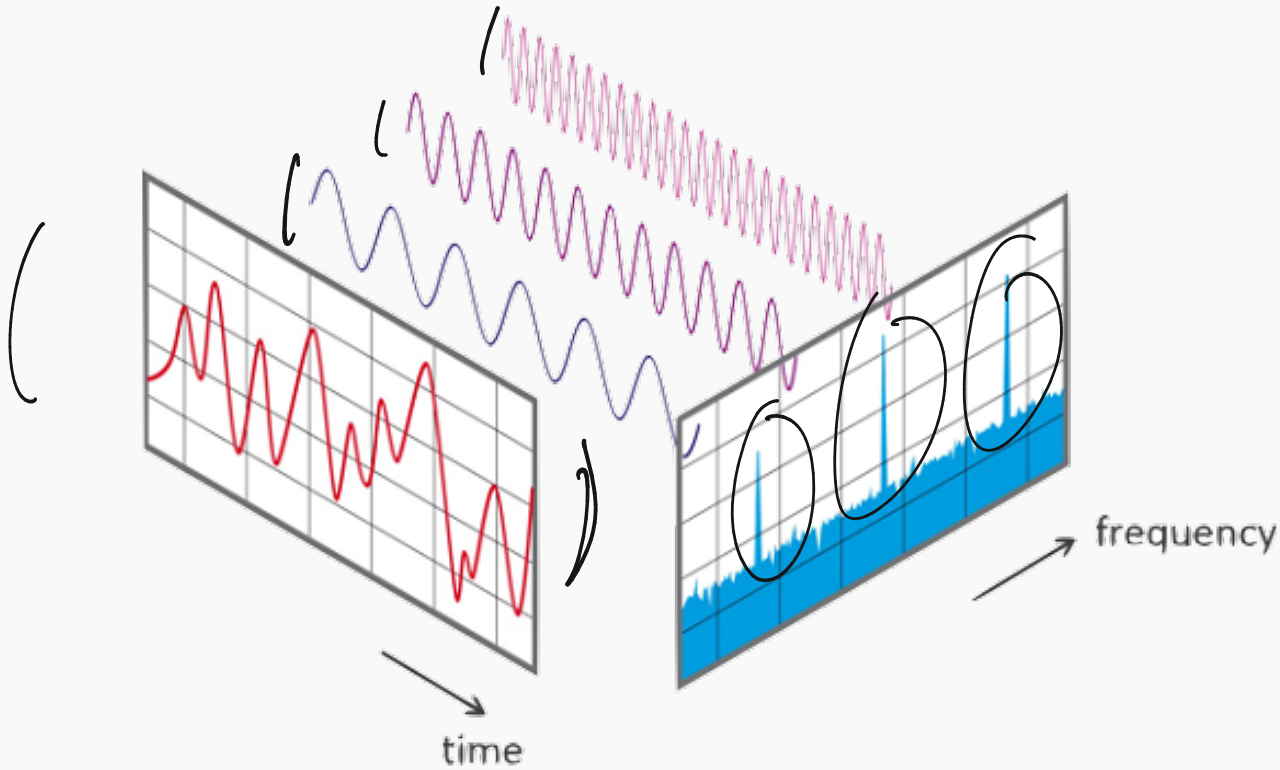


Real part of $\mathsf{F}_{j,k}$.

$\mathbf{Fy}$ computes the Discrete Fourier Transform of the vector $\mathbf{y}$. Can be computed in $O(n \log n)$ time using a divide and conquer algorithm (the Fast Fourier Transform).

The real part of $e^{-2\pi i \frac{j \cdot k}{n}}$ equals $cos(2\pi j \cdot k)$. So, the j^{th} row of $\mathbf{F}$ looks like a cosine wave with frequency $2\pi j$.

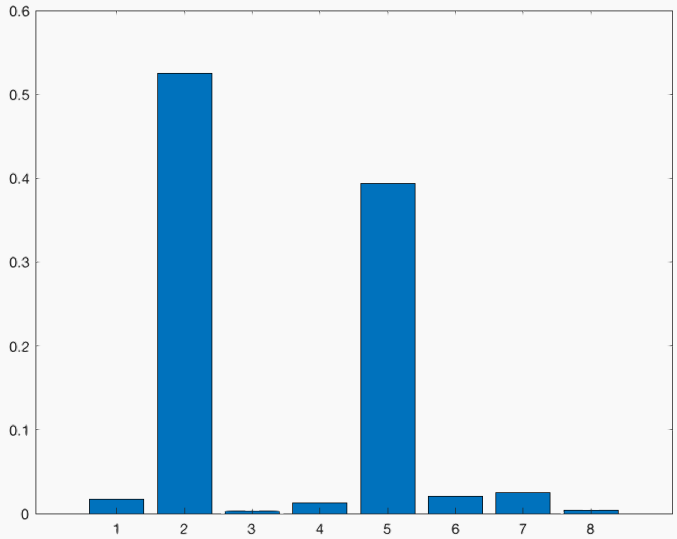
Computing $\mathbf{Fx}$ computes inner products of $\mathbf{x}$ with a bunch of different frequencies, which can be used to decompose the vector into a sum of those frequencies.



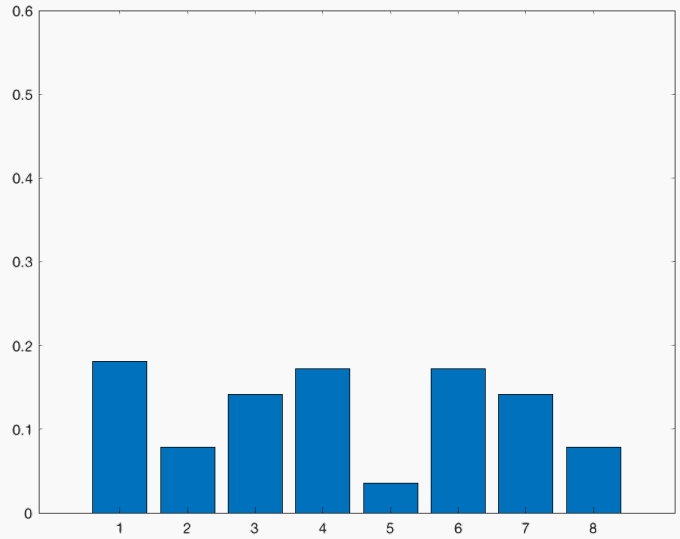
WALSH-HADAMARD TRANSFORM

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

The Uncertainty Principal (informal): A function and it's Fourier transform cannot both be concentrated.



Vector **y**.



Fourier transform **Fy**.

THE UNCERTAINTY PRINCIPAL

Sampling does not preserve norms, i.e. $\|\mathbf{S}\mathbf{y}\|_2 \not\approx \|\mathbf{y}\|_2$ when $\mathbf{y}$ has a few large entries.

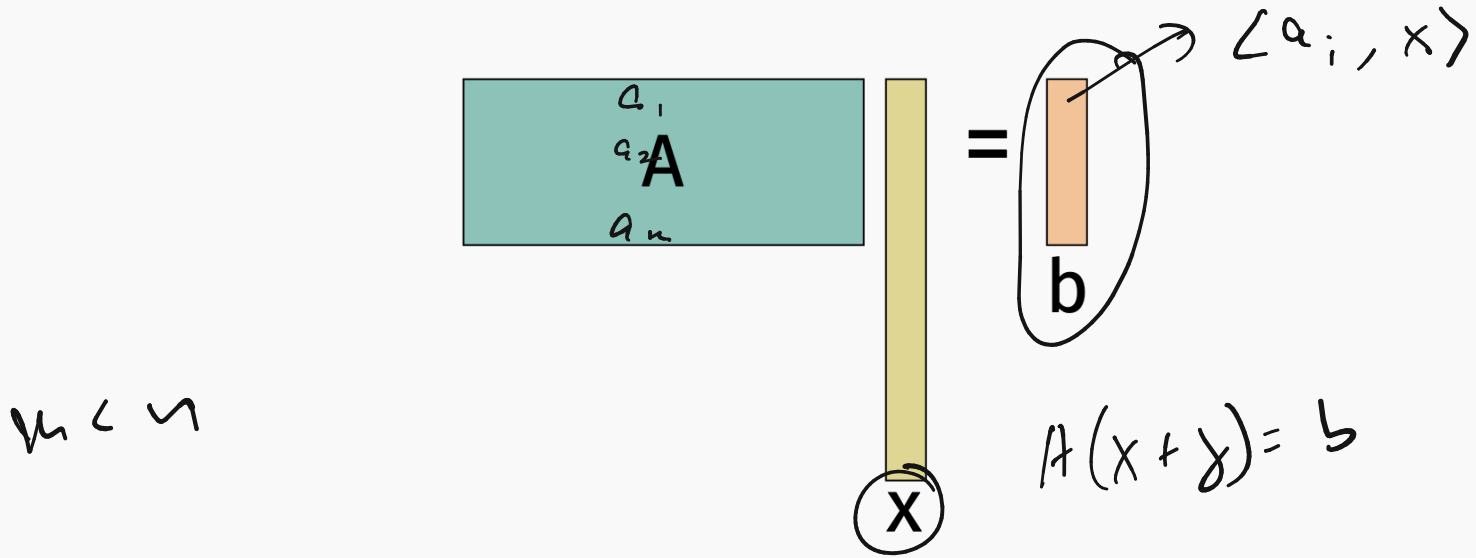
Taking a Fourier transform exactly eliminates this hard case, without changing $\mathbf{y}$'s norm.

One of the central tools in the field of (sparse recovery) aka (compressed sensing).

SPARSE RECOVERY/COMPRESSED SENSING PROBLEM SETUP

Goal: Recover a vector $\mathbf{x}$ from linear measurements.

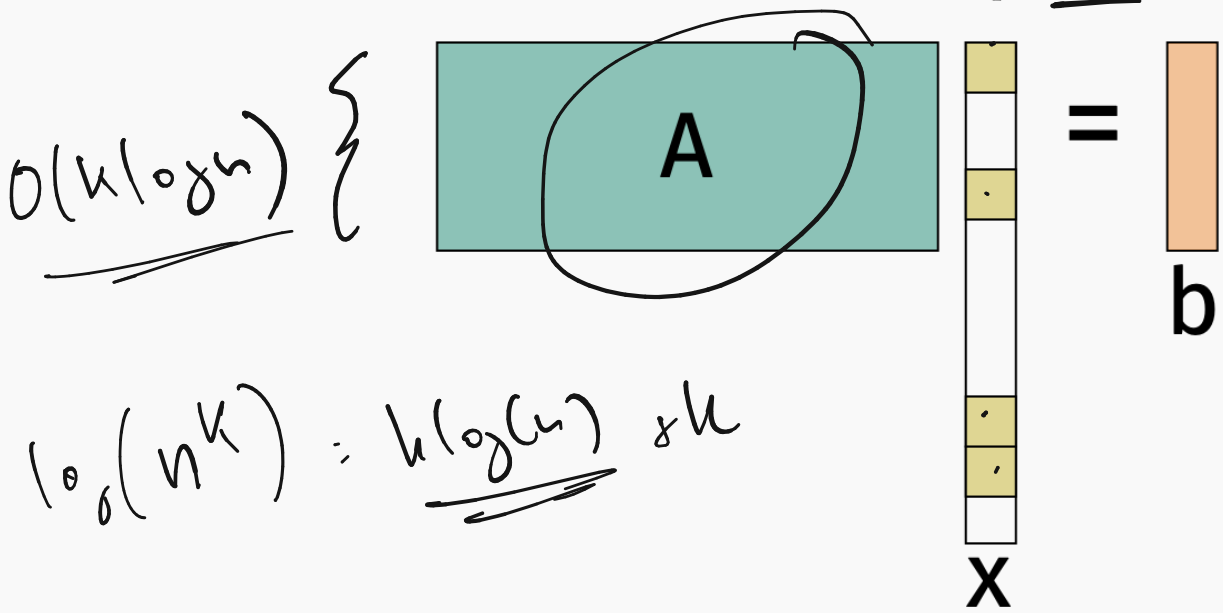
Choose $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m < n$. Assume we can access $\mathbf{b} = \mathbf{Ax}$ via some black-box measurement process. Try to recover $\mathbf{x}$ from the information in $\mathbf{b}$.



- Infinite possible solutions $\mathbf{y}$ to $\mathbf{Ay} = \mathbf{b}$, so in general, it is impossible to recover $\mathbf{x}$ from $\mathbf{b}$.
- Can often be possible if $\mathbf{x}$ has additional structure!

Need to make some assumption to solve the problem. Given $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m < n$, $\mathbf{b} \in \mathbb{R}^m$, want to recover $\mathbf{x}$.

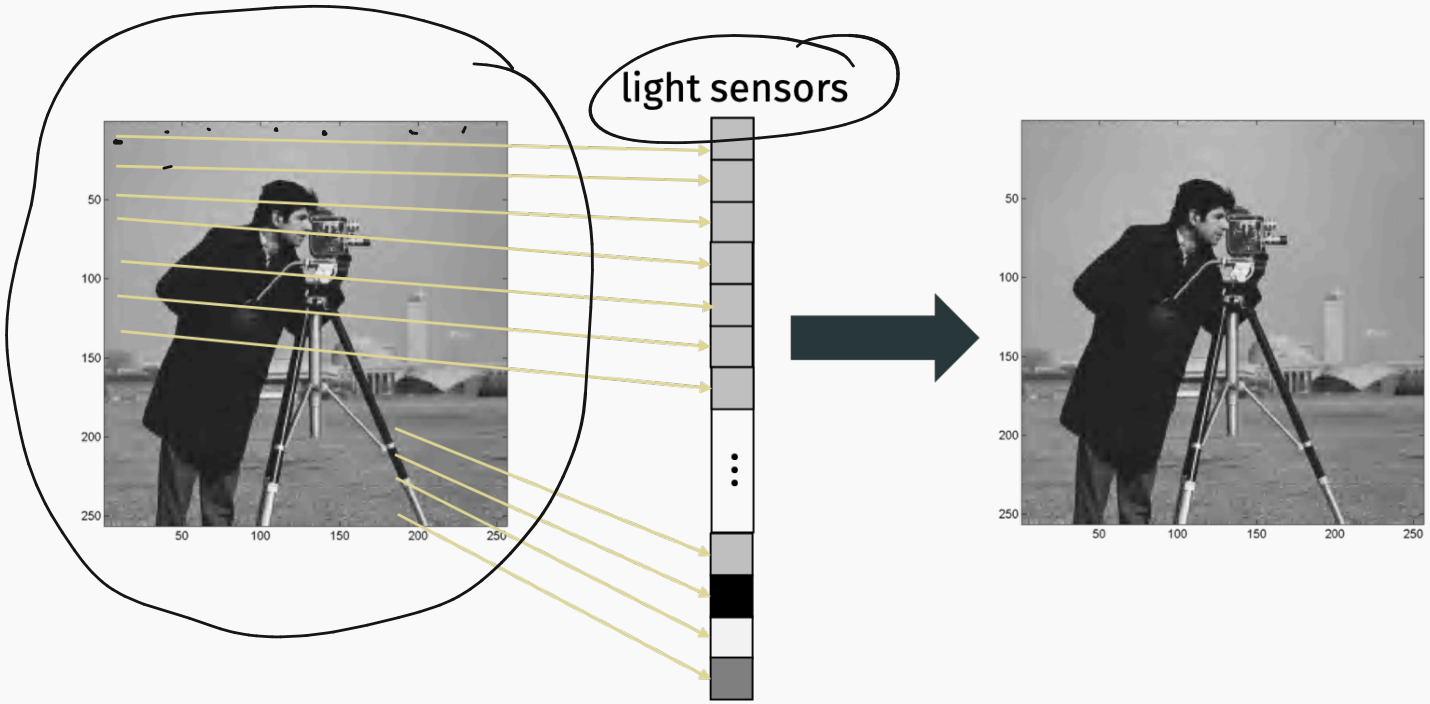
- Assume $\mathbf{x}$ is k-sparse for small k. $\|\mathbf{x}\|_0 \leq k$.



- In many cases can recover $\mathbf{x}$ with $\ll n$ rows. In fact, often $\sim O(k)$ suffice.

EXAMPLE APPLICATION: SINGLE PIXEL CAMERA

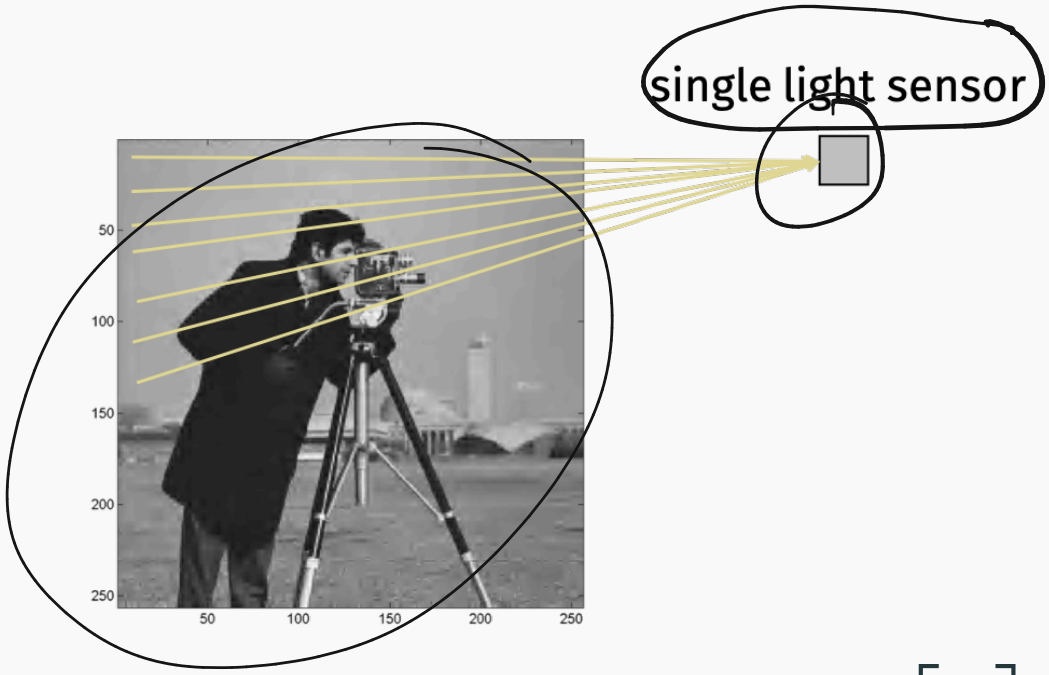
Typical acquisition of image by camera:



Requires one image sensor per pixel captured.

EXAMPLE APPLICATION: SINGLE PIXEL CAMERA

Compressed acquisition of image:

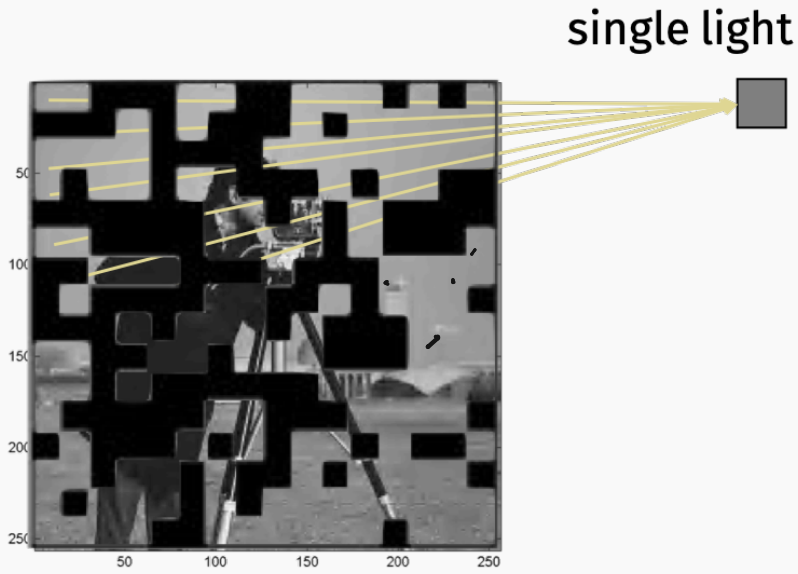


$$b = \frac{1}{n}\sum_{i=1} x_i = \begin{bmatrix} \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Does not provide very much information about the image.

EXAMPLE APPLICATION: SINGLE PIXEL CAMERA

But you can get more information from other linear measurements via masking!



$$b_i = \langle \mathbf{a}_i, \mathbf{x} \rangle = \begin{bmatrix} 0 & 1 & 0 & 0 \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Piece together many of these masked measurements, and can recover the whole image!

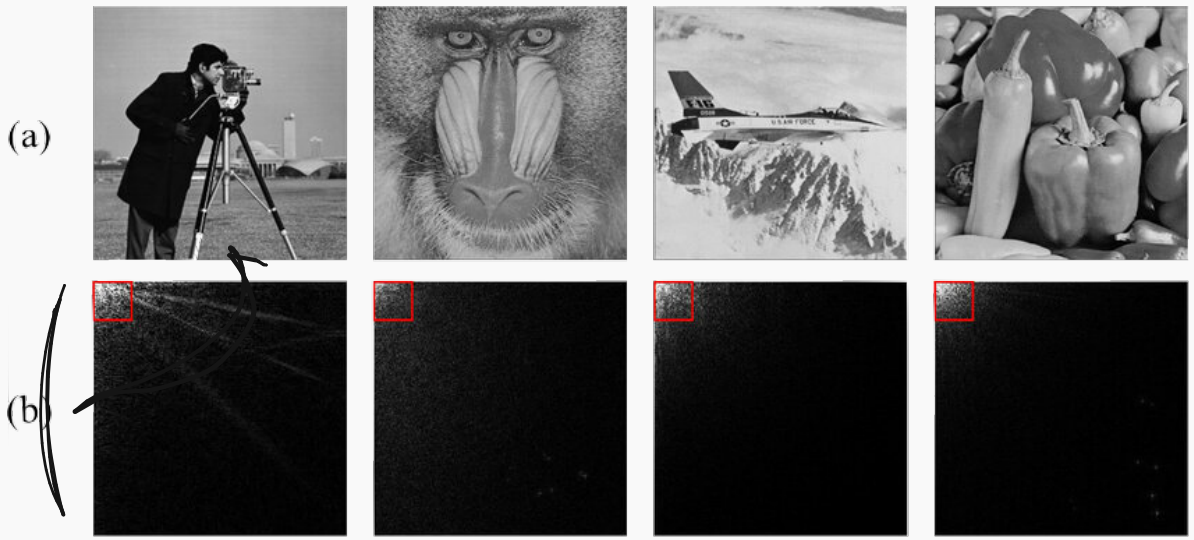
EXAMPLE APPLICATION: SINGLE PIXEL CAMERA

Applications in:

- Imaging outside of the visible spectrum (more expensive sensors).
- Microscopy.
- Other scientific imaging.
- We will discuss other applications shortly.

The theory we will discuss does not exactly describe these problems, but has been very valuable in modeling them.

Is sparsify a reasonable assumption?

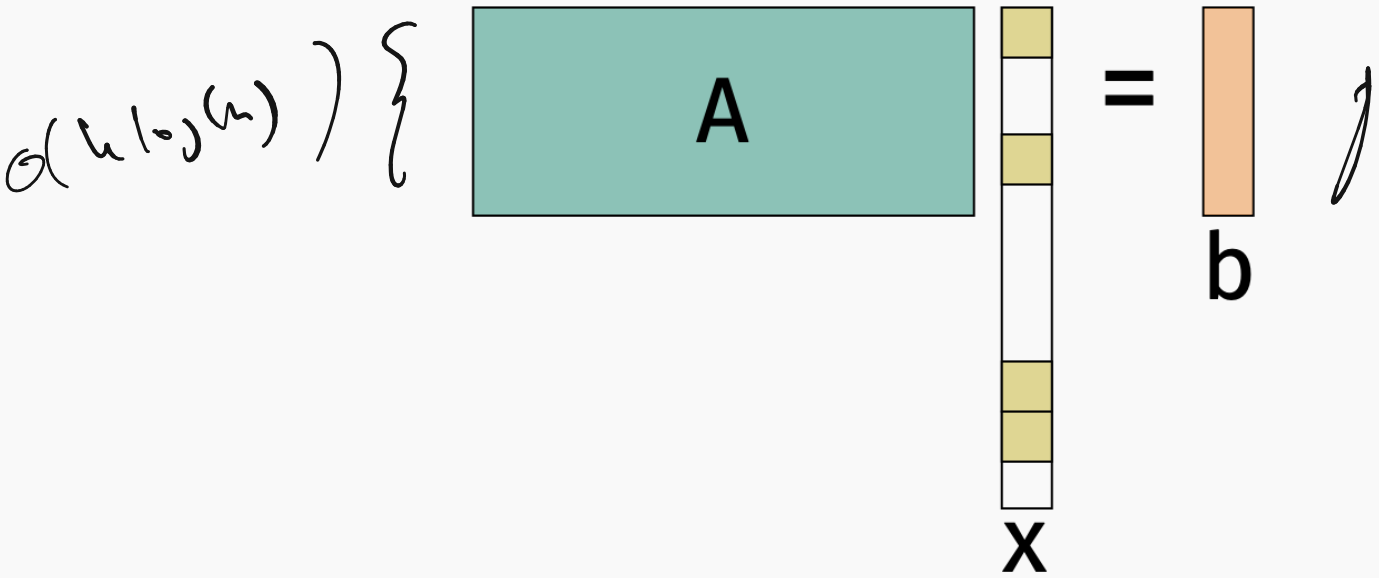


For some of the approachs we will discuss, it suffices to assume that $\mathbf{x}$ is sparse in any fixed (and known) basis. I.e. that $\mathbf{Vx}$ is sparse for some $n \times n$ orthogonal $\mathbf{V}$. E.g. images are sparse in the Discrete Cosine Transform basis.

Sparsity is a starting point for considering other more complex structure.

REQUIREMENTS FOR MEASUREMENT MATRIX

What matrices **A** would definitely not allow us to recover **x**?



Many ways to formalize our intuition

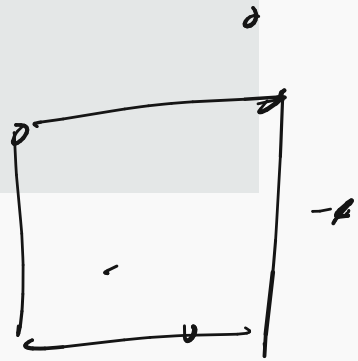
- $\mathbf{A}$ has Kruskal rank r. All sets of r columns in $\mathbf{A}$ are linearly independent.
  - Recover vectors $\mathbf{x}$ with sparsity $k = r/2$.
- $\mathbf{A}$ is μ-incoherent. $|\mathbf{A}_i^T \mathbf{A}_j| \leq \mu \|\mathbf{A}_i\|_2 \|\mathbf{A}_j\|_2$ for all columns $\mathbf{A}_i, \mathbf{A}_j$, $i \neq j$.
  - Recover vectors $\mathbf{x}$ with sparsity $k = 1/\mu$.
- **Focus today**: $\mathbf{A}$ obeys the Restricted Isometry Property.

RESTRICTED ISOMETRY PROPERTY

Definition ((q, ϵ)-Restricted Isometry Property)

A matrix $\mathbf{A}$ satisfies (q, ϵ)-RIP if, for all $\mathbf{x}$ with $\|\mathbf{x}\|_0 \leq q$,

$$(1 - \epsilon)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \epsilon)\|\mathbf{x}\|_2^2.$$



- Johnson-Lindenstrauss type condition.
- $\mathbf{A}$ preserves the norm of all q sparse vectors, instead of the norms of a fixed discrete set of vectors, or all vectors in a subspace (as in subspace embeddings).
- **Preview:** A random matrix $\mathbf{A}$ with $\sim O(q \log(n/q))$ rows satisfies RIP.

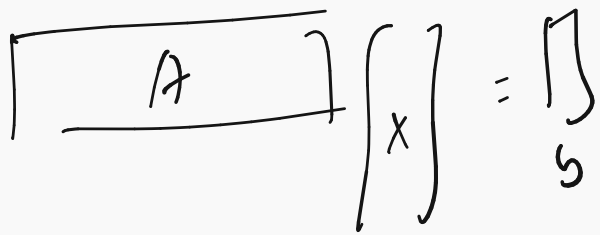
FIRST SPARSE RECOVERY RESULT

Theorem (ℓ_0-minimization)

Suppose we are given $\mathbf{A} \in \mathbb{R}^{m \times n}$ *and* $\mathbf{b} = \mathbf{A}\mathbf{x}$ *for an unknown* k*-sparse* $\mathbf{x} \in \mathbb{R}^n$*. If* $\mathbf{A}$ *is* $(2k, \epsilon)$*-RIP for any* $\epsilon < 1$ *then* $\mathbf{x}$ *is the unique minimizer of:*

$$\min \|\mathbf{z}\|_0 \quad \text{subject to} \quad \mathbf{A}\mathbf{z} = \mathbf{b}.$$

- Establishes that information theoretically we can recover $\mathbf{x}$. Solving the ℓ_0-minimization problem is computationally difficult, requiring $O(n^k)$ time. We will address faster recovery shortly.



FIRST SPARSE RECOVERY RESULT

Claim: If $\mathbf{A}$ is $(2k, \epsilon)$-RIP for any $\epsilon < 1$ then $\mathbf{x}$ is the unique minimizer of $\min_{\mathbf{Az}=\mathbf{b}} \|\mathbf{z}\|_0$.

Proof: By contradiction, assume there is some $\mathbf{y} \neq \mathbf{x}$ such that $\mathbf{Ay} = \mathbf{b}$, $\|\mathbf{y}\|_0 \leq \|\mathbf{x}\|_0$.

Δ

$Ay = Ax \qquad A(x-y) = 0 \qquad \Delta$ is $2 \cdot k$ sparse at most.

$(1-\epsilon)\|\Delta\|_2 \leq \|A\Delta\|_2 \leq (1+\epsilon)\|\Delta\|_2$ contradiction.

$\Delta \neq 0$ $\qquad$ $\Delta \neq 0$

Important note: There are robust versions of this theorem and the others we will discuss. These are much more important practically. Here's a flavor of a robust result:

- Suppose $\mathbf{b} = \mathbf{A}(\mathbf{x} + \mathbf{e})$ where $\mathbf{x}$ is k-sparse and $\mathbf{e}$ is dense but has bounded norm.
- Recover some k-sparse $\tilde{\mathbf{x}}$ such that:

$$\|\tilde{\mathbf{x}} - \mathbf{x}\|_2 \leq \|\mathbf{e}\|_1$$

or even

$$\|\tilde{\mathbf{x}} - \mathbf{x}\|_2 \leq O\left(\frac{1}{\sqrt{k}}\right) \|\mathbf{e}\|_1.$$

ROBUSTNESS

We will not discuss robustness in detail, but along with computational considerations, it is a big part of what has made compressed sensing such an active research area in the last 30 years. Non-robust compressed sensing results have been known for a long time:

Gaspard Riche de Prony, *Essay experimental et analytique: sur les lois de la dilatabilite de fluides elastique et sur celles de la force expansive de la vapeur de l'alcool, a differentes temperatures.* Journal de l'Ecole Polytechnique, 24–76. **1795**.

(r, a)-BIP

What matrices satisfy this property?

- Random Johnson-Lindenstrauss matrices (Gaussian, sign, etc.) with $m = O(\frac{k \log(n/k)}{\epsilon^2})$ rows are (k, ϵ)-RIP.
- ~~Random $m \sim O\left(\frac{k \log^2 k \log n}{\epsilon^2}\right)$ rows of the discrete Fourier matrix **F**~~ a random $m \sim O\left(\frac{k \log^2 k \log n}{\epsilon^2}\right)$ rows of the discrete Fourier matrix **F**.

Odd Bajer

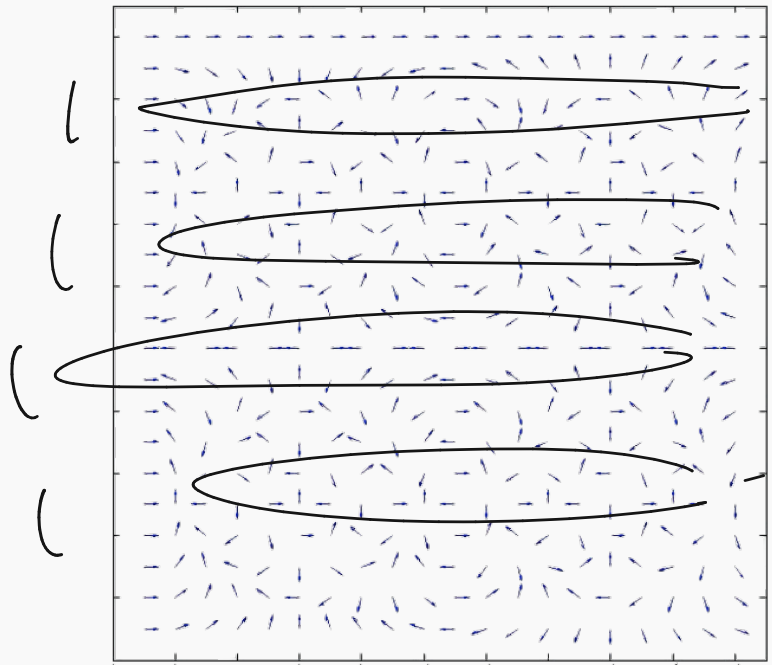
Improves on a long line of work: Candès, Tao, Rudelson, Vershynin, Cheraghchi, Guruswami, Velingker, Bourgain.

THE DISCRETE FOURIER MATRIX

The $n \times n$ discrete Fourier matrix $\mathbf{F}$ is defined:

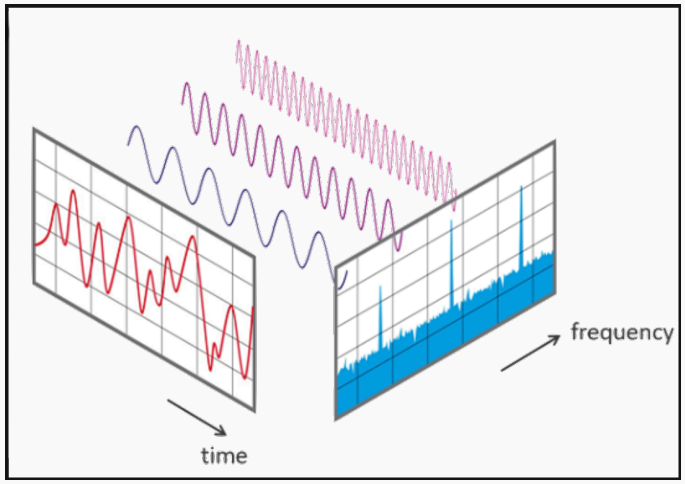
$$F_{j,k} = e^{\frac{-2\pi i}{n} j \cdot k},$$

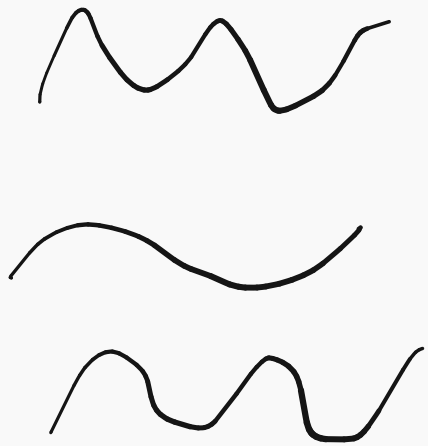
where $i = \sqrt{-1}$. Recall $e^{\frac{-2\pi i}{n} j \cdot k} = cos(2\pi jk/n) - i\,\mathbf{sin}(2\pi jk/n)$.



PSEUDORANDOM RIP MATRICES

In many applications can compute measurements of the form $\mathbf{Ax} = \mathbf{SFx}$, where $\mathbf{F}$ is the Discrete Fourier Transform matrix (what an FFT computes) and $\mathbf{S}$ is a subsampling matrix.





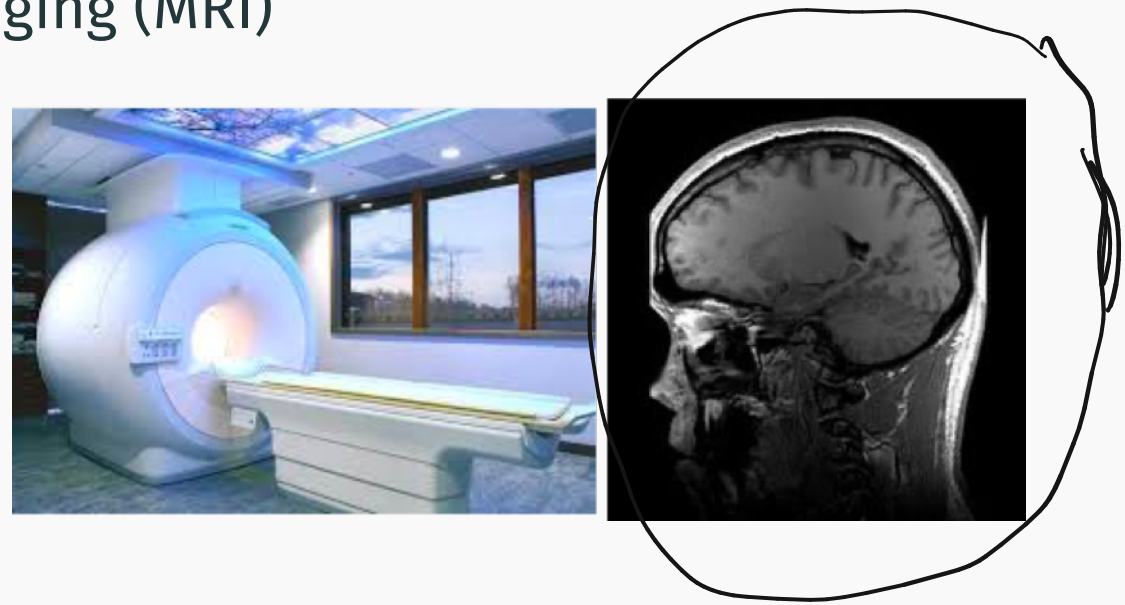
$\mathbf{F}$ decomposes $\mathbf{x}$ into different frequencies: $[\mathbf{Fx}]_j$ is the component with frequency j/n.

If $\mathbf{A} = \mathbf{SF}$ is a subset of rows from $\mathbf{F}$, then $\mathbf{Ax}$ is a subset of random frequency components from $\mathbf{x}$'s discrete Fourier transform.

In many scientific applications, we can collect entries of $\mathbf{Fx}$ one at a time for some unobserved data vector $\mathbf{x}$.

Warning: very cartoonish explanation of very complex problem.

Medical Imaging (MRI)



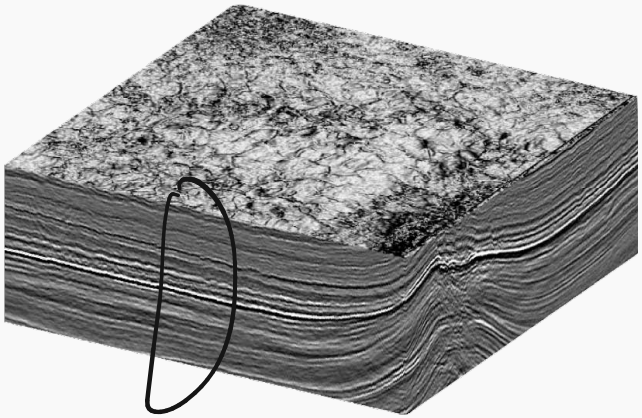
How do we measure entries of Fourier transform **Fx**? Blast the body with sounds waves of varying frequency.

- Using a small number of frequencies is especially important when trying to capture something moving (e.g. lungs, baby, child who can't sit still).
- Can also cut down on high power requirements.

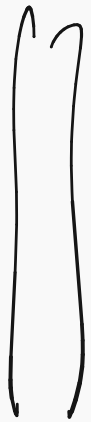
APPLICATION: GEOPHYSICS

Warning: very cartoonish explanation of very complex problem.

Understanding what material is beneath the crust:



Vibrate the earth at different frequencies! And measure the response.



Vibroseis Truck

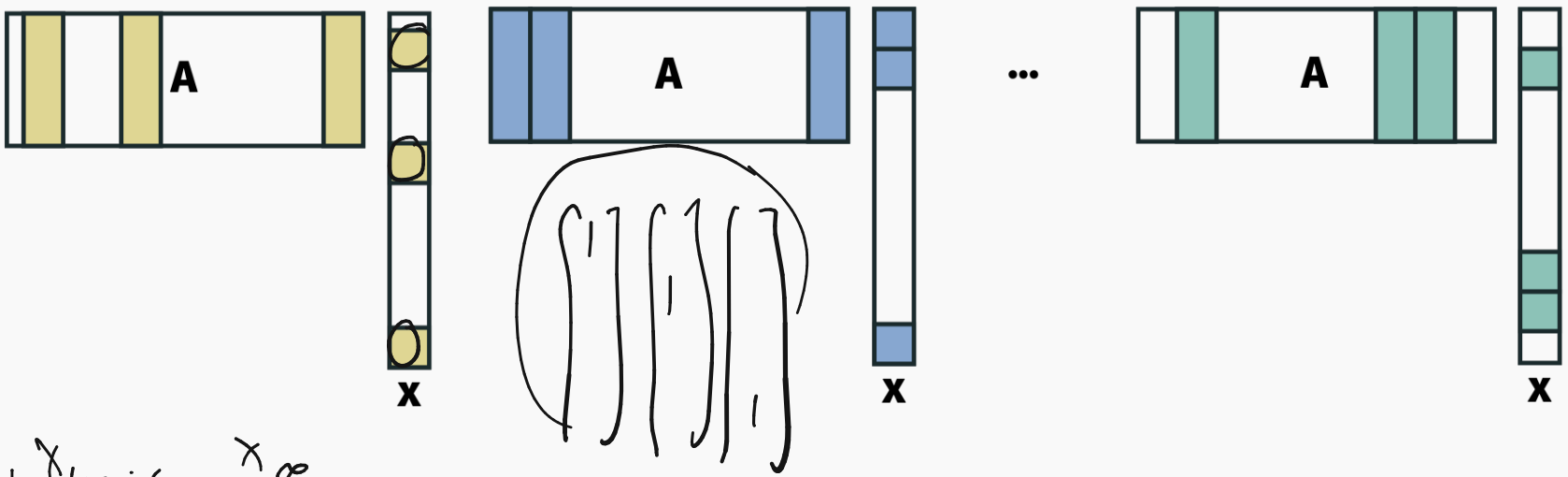
Can also use airguns, controlled explosions, vibrations from drilling, etc. The fewer measurements we need from **Fx**, the cheaper and faster our data acquisition process becomes.

Definition ((q, ϵ)-Restricted Isometry Property – Candes, Tao '05)

A matrix $\mathbf{A}$ satisfies (q, ϵ)-RIP if, for all $\mathbf{x}$ with $\|\mathbf{x}\|_0 \leq q$,

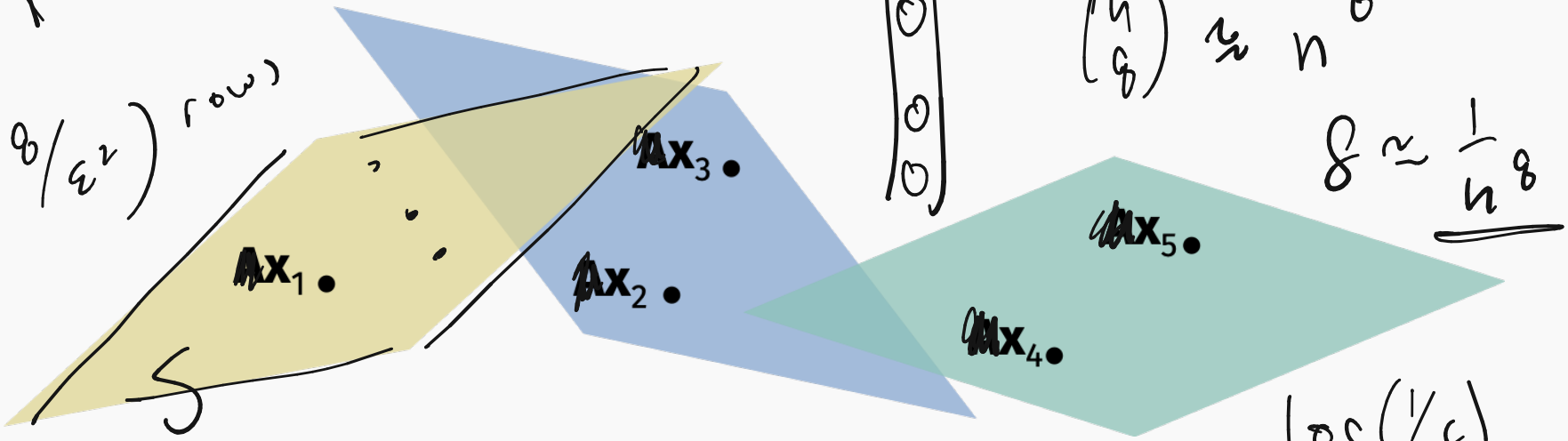
$$(1 - \epsilon)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \epsilon)\|\mathbf{x}\|_2^2.$$

The vectors that can be written as $\mathbf{A}\mathbf{x}$ for q sparse $\mathbf{x}$ lie in a union of q dimensional linear subspaces.



RESTRICTED ISOMETRY PROPERTY

Candes, Tao 2005: A random JL matrix with $O(q \log(n/q)/\epsilon^2)$ rows satisfies (q, ϵ)-RIP with high probability.



Any ideas for how you might prove this? I.e. prove that a random matrix preserves the norm of every $\mathbf{x}$ in this union of subspaces?

Theorem (Subspace Embedding from JL)

Let $\mathcal{U} \subset \mathbb{R}^n$ be a q-dimensional linear subspace in $\mathbb{R}^n$. If $\mathbf{\Pi} \in \mathbb{R}^{m \times n}$ is chosen from any distribution $\mathcal{D}$ satisfying the Distributional JL Lemma, then with probability $1 - \delta$,

$$(1 - \epsilon)\|\mathbf{v}\|_2^2 \leq \|\mathbf{\Pi v}\|_2^2 \leq (1 + \epsilon)\|\mathbf{v}\|_2^2$$

for <u>all</u> $\mathbf{v} \in \mathcal{U}$, as long as $m = O\left(\frac{q + \log(1/\delta)}{\epsilon^2}\right)$.

Quick argument:

FIRST SPARSE RECOVERY RESULT

Theorem (ℓ_0-minimization)

Suppose we are given $\mathbf{A} \in \mathbb{R}^{m \times n}$ *and* $\mathbf{b} = \mathbf{A}\mathbf{x}$ *for an unknown* k*-sparse* $\mathbf{x} \in \mathbb{R}^n$*. If* $\mathbf{A}$ *is* $(2k, \epsilon)$*-RIP for any* $\epsilon < 1$ *then* $\mathbf{x}$ *is the unique minimizer of:*

$$\min \|\mathbf{z}\|_0 \qquad \textit{subject to} \qquad \mathbf{A}\mathbf{z} = \mathbf{b}.$$

Problem: This optimization problem naively takes $O(n^k)$ time to solve.

Convex relaxation of the ℓ_0 minimization problem:

Problem (Basis Pursuit, i.e. ℓ_1 minimization.)

$$\min_{\mathbf{z}} \|\mathbf{z}\|_1 \quad \textit{subject to} \quad \mathbf{Az} = \mathbf{b}.$$

- Objective is convex.
- Optimizing over convex set.

Can be solved in $\mathsf{poly}(n)$ time using a linear program or using e.g. projected gradient descent. Other similar relaxations also work. E.g. Lasso regularization $\min_{\mathbf{z}} \|\mathbf{Az} - \mathbf{b}\|_2 + \lambda\|\mathbf{z}\|_1$.

Theorem

If $\mathbf{A}$ *is* $(3k, \epsilon)$*-RIP for* $\epsilon < .17$ *and* $\|\mathbf{x}\|_0 = k$*, then* $\mathbf{x}$ *is the unique optimal solution of the Basis Pursuit optimization problem.*

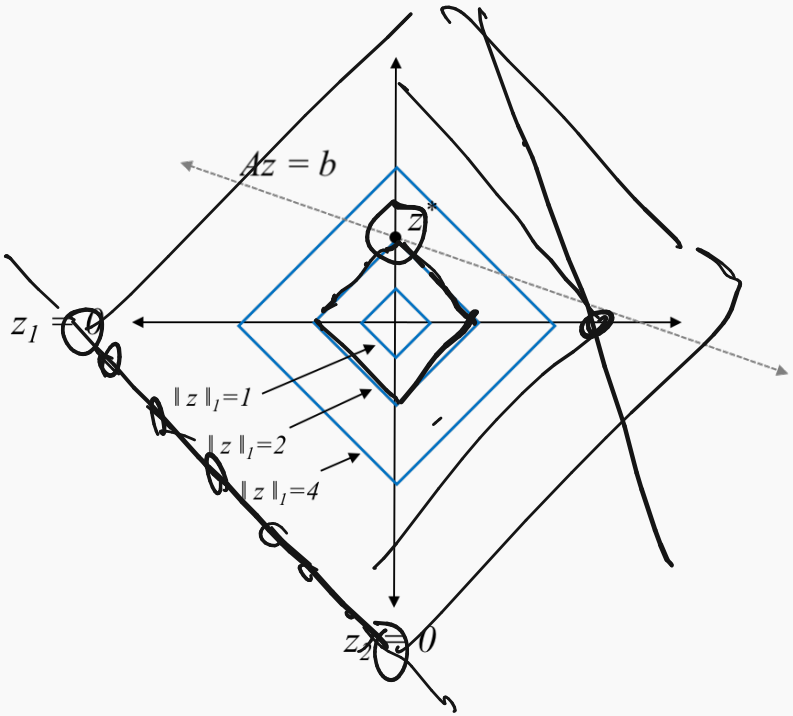
Two surprising things about this result:

- Exponentially improve computational complexity with only a constant factor overhead in measurement complexity.
- Typical "relax-and-round" algorithm, but rounding is not even necessary! Just return the solution of the relaxed problem.

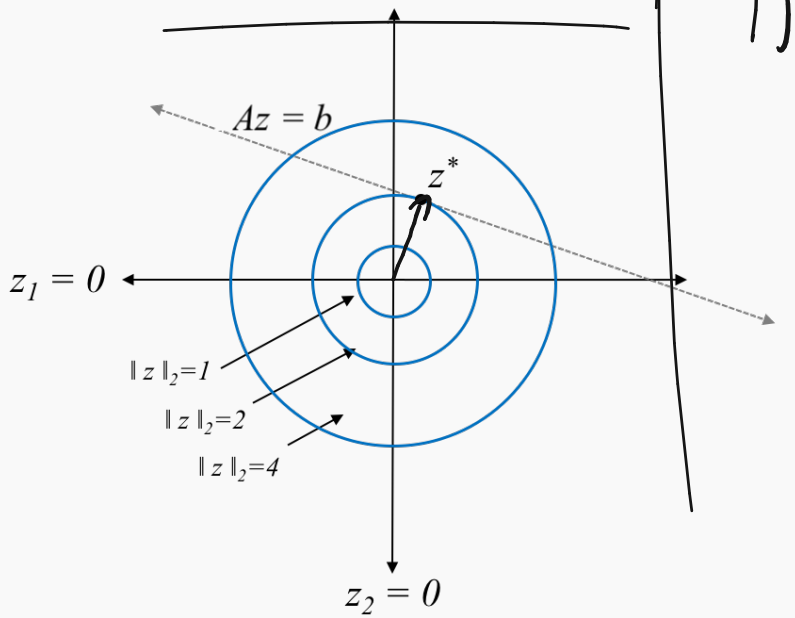
(Why ℓ_1 norm instead of ℓ_2 norm?)

BASIS PURSUIT INTUITION

Suppose $\mathbf{A}$ is 2×1, so $\mathbf{b}$ is just a scalar and $\mathbf{x}$ is a 2-dimensional vector.



Vertices of level sets of ℓ_1 norm correspond to sparse solutions.



This is not the case e.g. for the ℓ_2 norm.

$$\min_{\mathbf{z}} \|\mathbf{z}\|_1 \qquad \text{subject to} \qquad \mathbf{Az} = \mathbf{b}.$$

Theorem

If $\mathbf{A}$ *is* $(3k, \epsilon)$*-RIP for* $\epsilon < .17$ *and* $\|\mathbf{x}\|_0 = k$*, then* $\mathbf{x}$ *is the unique optimal solution of the Basis Pursuit LP).*

Similar proof to ℓ_0 minimization:

- By way of contradiction, assume $\mathbf{x}$ is not the optimal solution. Then there exists some non-zero Δ such that:
  - $\|\mathbf{x} + \Delta\|_1 \leq \|\mathbf{x}\|_1$
  - $\mathbf{A}(\mathbf{x} + \Delta) = \mathbf{A}\mathbf{x}$. I.e. $\mathbf{A}\Delta = 0$.

Difference is that we can no longer assume that Δ is sparse.

We will argue that Δ is "approximately" sparse.

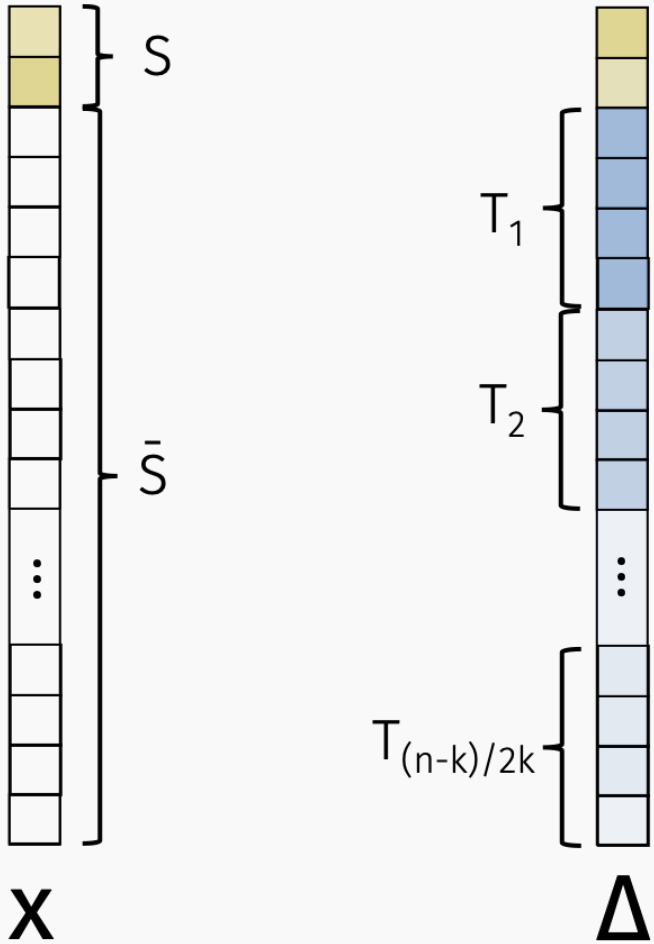
First tool:

For any q-sparse vector $\mathbf{w}$, $\|\mathbf{w}\|_2 \leq \|\mathbf{w}\|_1 \leq \sqrt{q}\|\mathbf{w}\|_2$

Second tool:

For any norm and vectors $\mathbf{a}, \mathbf{b}$, $\|\mathbf{a} + \mathbf{b}\| \geq \|\mathbf{a}\| - \|\mathbf{b}\|$

Some definitions: S is the set of k non-zero indices in $\mathbf{x}$. $\bar{T}_1$ is the set of $2k$ indices not in S with largest magnitude in Δ. $\bar{T}_2$ is the set of $2k$ indices not in S with next largest magnitudes, etc.



Recall: By way of contradiction, if $\mathbf{x}$ is not the minimizer of the ℓ_1 problem, then there is some Δ such that $\mathbf{A}(\mathbf{x} + \Delta) = \mathbf{b}$ and $\|\mathbf{x} + \Delta\|_1 \leq \|\mathbf{x}\|_1$.

Claim 1 (approximate sparsity of Δ): $\|\Delta_S\|_1 \geq \|\Delta_{\bar{S}}\|_1$

Claim 2 (ℓ_2 approximate sparsity): $\|\Delta_S\|_2 \geq \sqrt{2}\sum_{j\geq 2}\|\Delta_{T_j}\|_2$:

We have:

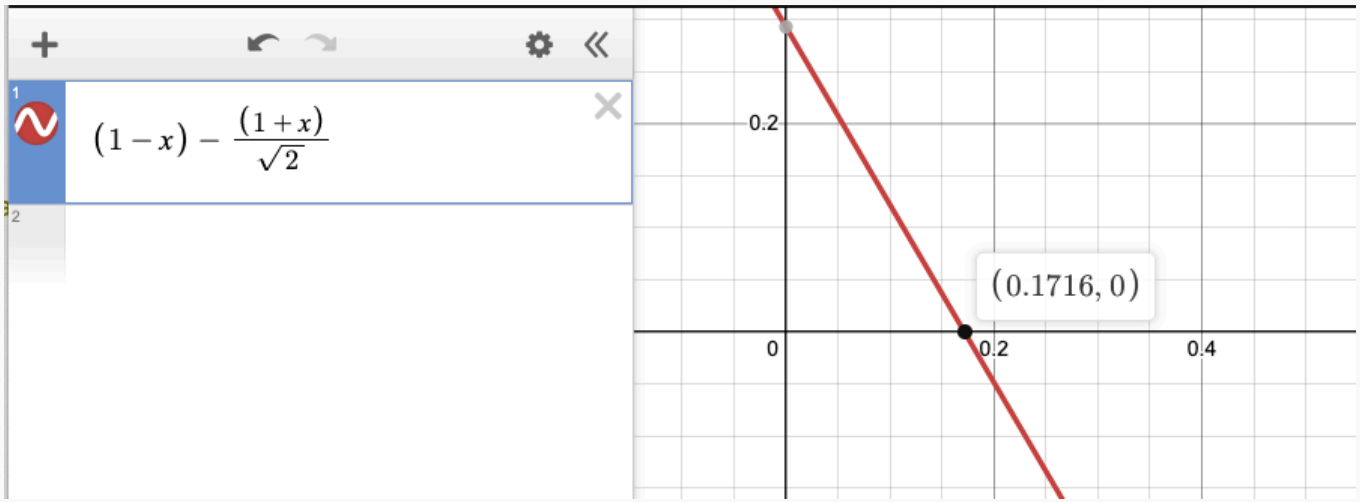
$$\|\Delta_s\|_2 \geq \frac{1}{\sqrt{k}}\|\Delta_S\|_1 \geq \frac{1}{\sqrt{k}}\|\Delta_{\bar{S}}\|_1 = \frac{1}{\sqrt{k}}\sum_{j\geq 1}\|\Delta_{T_j}\|_1.$$

So it suffices to show that: $\|\Delta_{T_j}\|_1 \geq \sqrt{2k}\|\Delta_{T_{j+1}}\|_2$

Finish up proof by contradiction: Recall that $\mathbf{A}$ is assumed to have the $(3k, \epsilon)$ RIP property. And by way of contradiction $\mathbf{A}(\mathbf{x} + \Delta) = \mathbf{b}$.

$$0 = \|\mathbf{A}\Delta\|_2 \geq \|\mathbf{A}\Delta_{S \cup T_1}\|_2 - \sum_{j \geq 2} \|\mathbf{A}\Delta_{T_j}\|_2$$

We have that $(1-\epsilon) - \frac{1+\epsilon}{\sqrt{2}} \geq 0$ whenever $\epsilon < .17$.



Theorem

If $\mathbf{A}$ *is* $(3k, \epsilon)$*-RIP for* $\epsilon < .17$ *and* $\|\mathbf{x}\|_0 = k$*, then* $\mathbf{x}$ *is the unique optimal solution of the Basis Pursuit optimization problem, which can be solved in polynomial time.*

FASTER METHODS

A lot of interest in developing even faster algorithms that avoid using the "heavy hammer" of linear programming, which runs in roughly $O(n^{3.5})$ time.

- **Iterative Hard Thresholding**: Looks a lot like projected gradient descent. Solve $\min_{\mathbf{z}} \|\mathbf{Az} - \mathbf{b}\|$ with gradient descent while continually projecting z back to the set of k-sparse vectors. Runs in time $\sim O(nk \log n)$ for Gaussian measurement matrices and $O(n \log n)$ for subsampled Fourer matrices.
- Other "first order" type methods: Orthogonal Matching Pursuit, CoSaMP, Subspace Pursuit, etc.

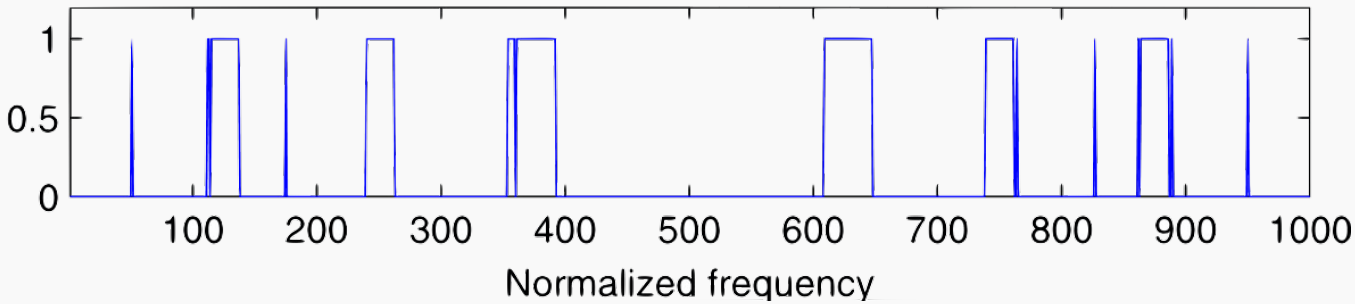
FASTER METHODS

When $\mathbf{A}$ is a subsampled Fourier matrix, there are now methods that run in $O(k \log^c n)$ time [Hassanieh, Indyk, Kapralov, Katabi, Price, Shi, etc. 2012+].

Corollary: When $\mathbf{x}$ is k-sparse, we can compute the inverse Fourier transform $\mathbf{F}^*\mathbf{F}\mathbf{x}$ of $\mathbf{F}\mathbf{x}$ in $O(k \log^c n)$ time!

- Randomly subsample $\mathbf{F}\mathbf{x}$.
- Feed that input into our sparse recovery algorithm to extract $\mathbf{x}$.

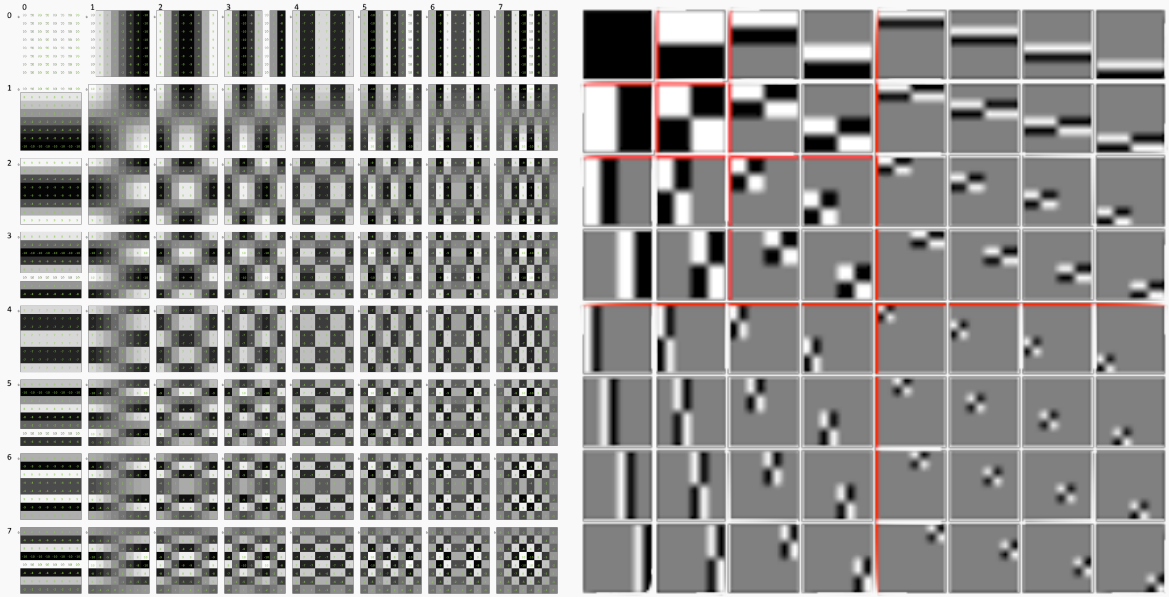
Fourier and inverse Fourier transforms in sublinear time when the output is sparse.



Applications in: Wireless communications, GPS, protein imaging, radio astronomy, etc. etc.

COMPRESSED SENSING FOR IMAGES

Compressed sensing for image data is based on the idea that "natural images" are sparse if some basis. E.g. the DCT or Wavelet basis.



I.e. there is some representation of the image that requires many fewer numbers than explicitly writing down the pixels.

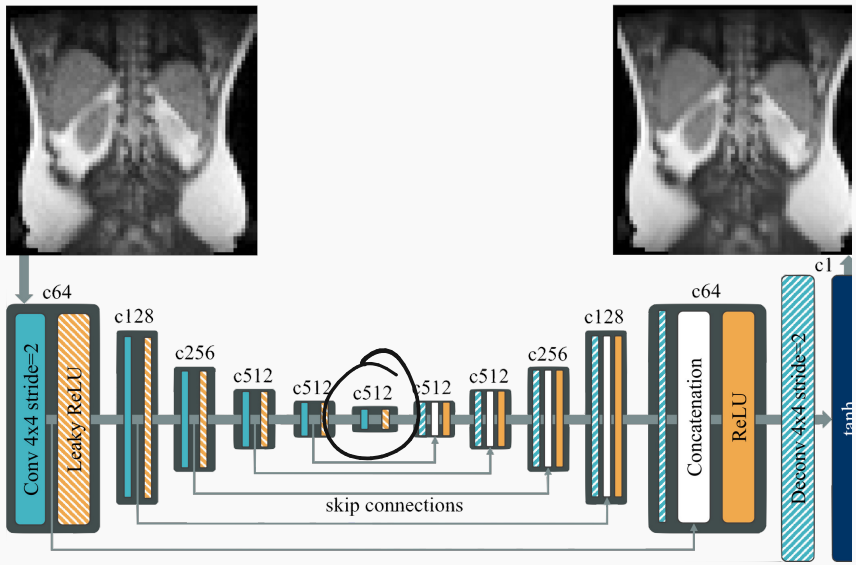
COMPRESSED SENSING RELATED TO MODERN DEEP LEARNING METHOD METHODS

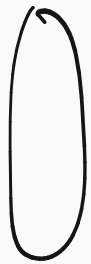
Compressed Sensing using Generative Models

Ashish Bora* Ajil Jalal† Eric Price‡ Alexandros G. Dimakis§

Abstract

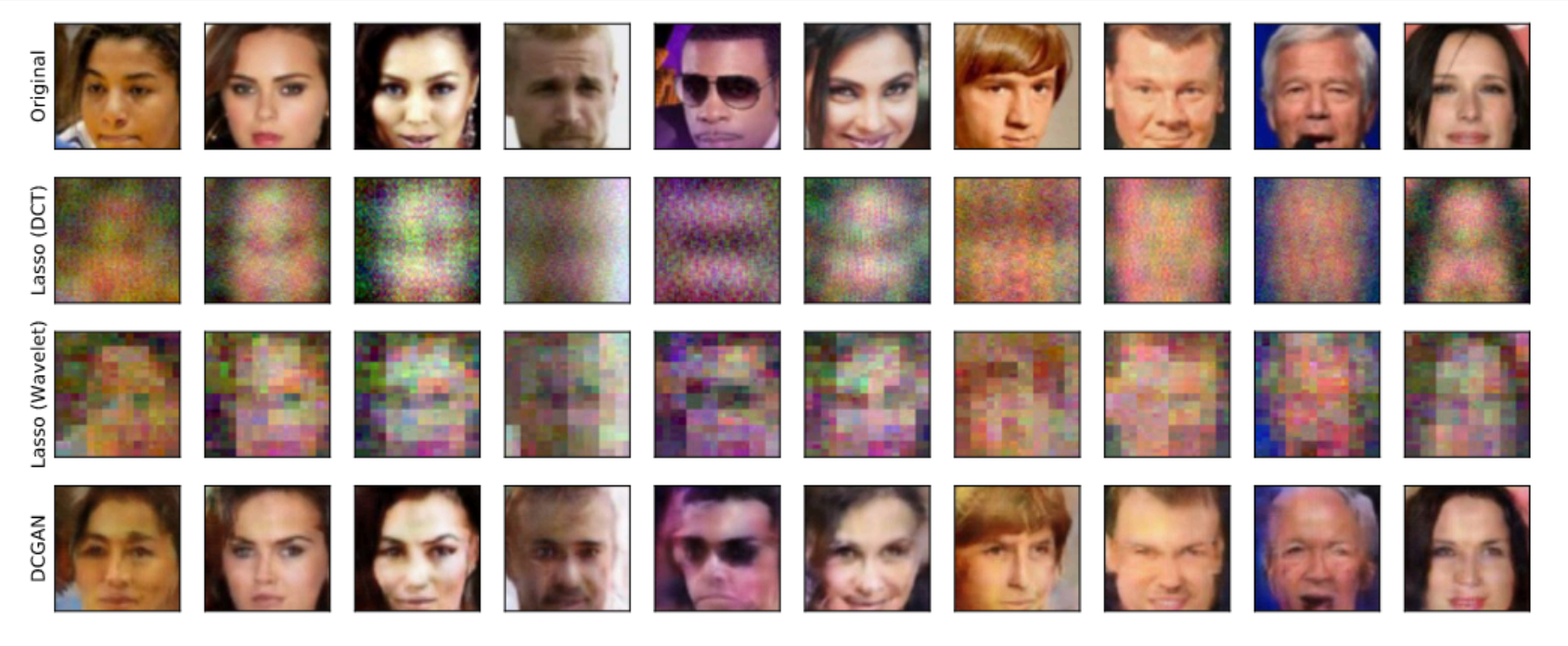
The goal of compressed sensing is to estimate a vector from an underdetermined system of noisy linear measurements, by making use of prior knowledge on the structure of vectors in the relevant domain. For almost all results in this literature, the structure is represented by sparsity in a well-chosen basis. We show how to achieve guarantees similar to standard compressed sensing but without employing sparsity at all. Instead, we suppose that vectors lie near the range of a generative model $G : \mathbb{R}^k \to \mathbb{R}^n$. Our main theorem is that, if G is L-Lipschitz, then roughly $O(k \log L)$ random Gaussian measurements suffice for an ℓ_2/ℓ_2 recovery guarantee. We demonstrate our results using generative models from published variational autoencoder and generative adversarial networks. Our method can use 5-10x fewer measurements than Lasso for the same accuracy.





COMPRESSED SENSING FROM GENERATIVE MODELS

For many generative models (e.g., GANs, diffusion models) output is parameterized by a seed vector $\mathbf{z}$.



Process: measure image $\mathbf{x}$ by computing $\mathbf{b} = \mathbf{A}\mathbf{x}$ for a random matrix $\mathbf{A}$. Use gradient descent to find $\mathbf{z} \in \mathbb{R}^k$ to minimize:

$$\min \|\mathbf{A}\mathcal{G}(\mathbf{z}) - \mathbf{b}\|. \qquad \mathcal{G}(\mathbf{z})$$

Return $\mathcal{G}(\mathbf{z})$.

THANK YOU!

Thank you all for a great course! If you are interested in learning even more, there are several seminars at NYU that you might be interested in attending:

Theoretical Computer Science Seminar: https://csefoundations.engineering.nyu.edu/seminar.html.

Math and Data Seminar: https://mad.cds.nyu.edu/seminar/.

Computational Math and Scientific Computing Seminar: https://cims.nyu.edu/dynamic/calendars/seminars/computational-mathematics-and-scientific-computing-seminar/.