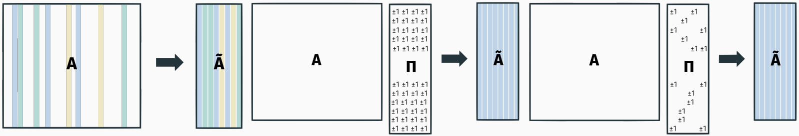
CS-GY 9223 D: Lecture 12
Fast Johnson-Lindenstrauss Transform, Start on Sparse Recovery and Compressed Sensing

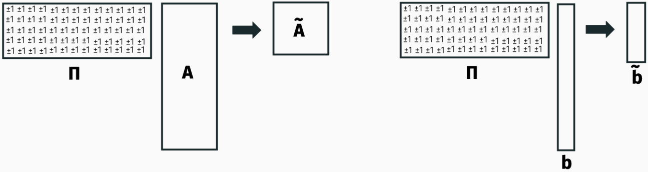
NYU Tandon School of Engineering, Prof. Christopher Musco

Main idea: If you want to compute singular vectors or eigenvectors, multiply two matrices, solve a regression problem, etc.:

1. Compress your matrices using a randomized method.
2. Solve the problem on the smaller or sparser matrix.
    - $\tilde{\mathbf{A}}$ called a "sketch" or "coreset" for $\mathbf{A}$.



Randomized approximate regression using a Johnson-Lindenstrauss Matrix:



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

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

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

Theorem (Randomized Linear Regression)

Let $\mathbf{\Pi}$ *be a properly scaled JL matrix (random Gaussian, sign, sparse random, etc.) with* $m = \tilde{O}\left(\frac{d}{\epsilon^2}\right)$ *rows. Then with probability* $(1-\delta)$*, for any* $\mathbf{A} \in \mathbb{R}^{n \times d}$ *and* $\mathbf{b} \in \mathbb{R}^n$*,*

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

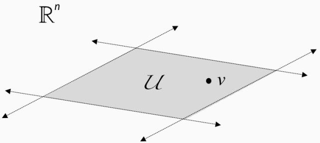
where $\tilde{\mathbf{x}}^* = \arg\min_{\mathbf{x}} \|\mathbf{\Pi}\mathbf{A}\mathbf{x} - \mathbf{\Pi}\mathbf{b}\|_2^2$.

Theorem (Subspace Embedding)

Let $\mathcal{U} \subset \mathbb{R}^n$ be a d-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 $1 - \delta$,

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

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



Theorem (Subspace Embedding)

Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ *be a matrix. If* $\mathbf{\Pi} \in \mathbb{R}^{m \times n}$ *is chosen from any distribution* $\mathcal{D}$ *satisfying the Distributional JL Lemma, then with probability* $1 - \delta$,

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

for all $\mathbf{x} \in \mathbb{R}^d$, *as long as* $m = O\left(\frac{d+\log(1/\delta)}{\epsilon^2}\right)$.

Implies regression result, and more.

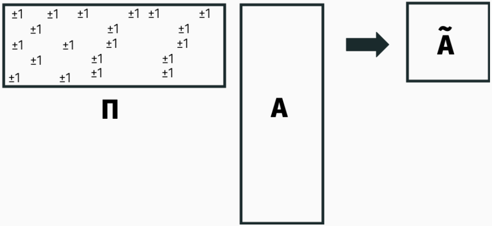
Example: The top singular value $\tilde{\sigma}_1^2$ of $\mathbf{\Pi}\mathbf{A}$ is a $(1 \pm \epsilon)$ approximation to the true top singular value σ_1^2. Do you see why?

For $\epsilon, \delta = O(1)$, we need $\mathbf{\Pi}$ to have $m = O(d)$ rows.

- Cost to solve $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$:
  - $O(nd^2)$ time for direct method. Need to compute $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$.
  - $O(nd) \cdot$ (# of iterations) time for iterative method (GD, AGD, conjugate gradient method).
- Cost to solve $\|\mathbf{\Pi}\mathbf{A}\mathbf{x} - \mathbf{\Pi}\mathbf{b}\|_2^2$:
  - $O(d^3)$ time for direct method.
  - $O(d^2) \cdot$ (# of iterations) time for iterative method.

But time to compute $\mathbf{\Pi A}$ is an $(m \times n) \times (n \times d)$ matrix multiply: $O(mnd) = O(nd^2)$ time.

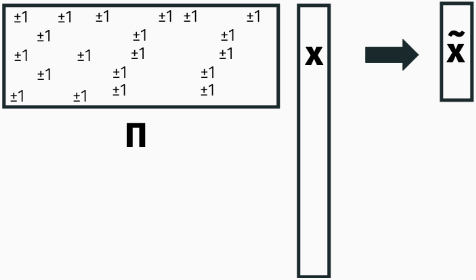
Goal: Develop faster Johnson-Lindenstrauss projections.



Typically using sparse or structured matrices instead of fully random JL matrices.

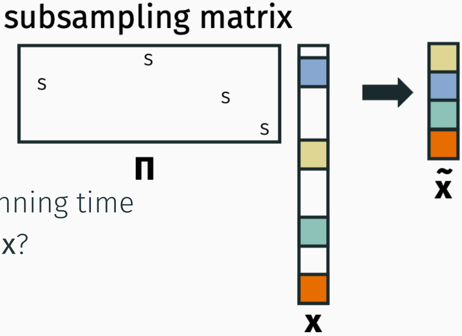
Goal: Develop methods that reduce a vector $\mathbf{x} \in \mathbb{R}^n$ down to $m \approx \frac{\log(1/\delta)}{\epsilon^2}$ dimensions in $o(mn)$ time and guarantee:

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



We will learn about a truly brilliant method that runs in $O(n \log n)$ time. **Preview:** Will involve Fast Fourier Transform in disguise.

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

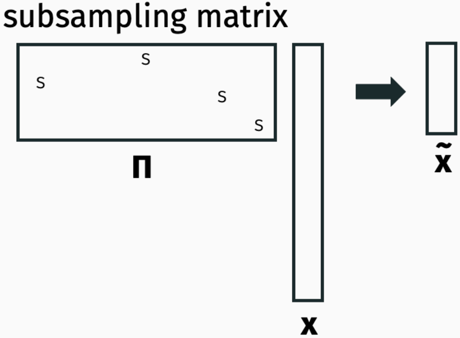


What's the running time tp compute $\mathbf{\Pi}\mathbf{x}$?

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

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

So $\mathbb{E}\|\mathbf{\Pi x}\|_2^2 = \|\mathbf{x}\|_2^2$ in expectation. To show it is close with high probability we would need to apply a concentration inequality. How do you think this will work out?



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

$\sigma^2 = \text{Var}[\|\mathbf{\Pi x}\|_2^2] =$

Recall Chebyshev's Inequality:

$$\Pr\left[\left|\|\mathbf{\Pi x}\|_2^2 - \|\mathbf{x}\|_2^2\right| \leq \frac{1}{10} \cdot \sigma\right] \leq \frac{1}{100}$$

We want additive error $\left|\|\mathbf{\Pi x}\|_2^2 - \|\mathbf{x}\|_2^2\right| \leq \epsilon \|\mathbf{x}\|_2^2$

We need to choose m so that:

$$\frac{1}{10}\sqrt{\frac{n}{m}}\|\mathbf{x}\|_4^2 \leq \epsilon\|\mathbf{x}\|_2^2.$$

How do these two two norms compare?

$$\|\mathbf{x}\|_4^2 = \left(\sum_{i=1}^{n} x_i^4\right)^{1/2} \qquad \|\mathbf{x}\|_2^2 = \sum_{i=1}^{n} x_i^2$$

Consider 2 extreme cases:

$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

We need to choose m so that:

$$\frac{1}{10}\sqrt{\frac{n}{m}}\|\mathbf{x}\|_4^2 \leq \epsilon\|\mathbf{x}\|_2^2.$$

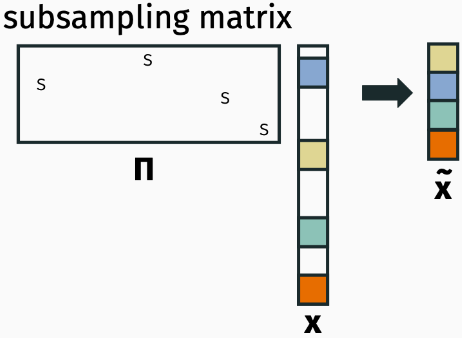
Suppose $\mathbf{x}$ is very evenly distributed. I.e., for all $i \in 1, \ldots, n$,

$$x_i^2 \leq \frac{c}{n}\sum_{i=1}^{n} x_i^2 = \frac{c}{n}\|\mathbf{x}\|_2^2$$

Claim: $\|\mathbf{x}\|_4^2 \leq \sqrt{\frac{c}{n}}\|\mathbf{x}\|_2^2$. So $m = O(c/\epsilon^2)$ samples suffices.[1]

[1]Using the right Bernstein bound we can prove $m = O(c\log(1/\delta)/\epsilon^2)$ suffices for failure probability δ.

So sampling does work to preserve the norm of $\mathbf{x}$, but only when the vector is relatively “smooth” (not concentrated). Do we expect to see such vectors in the wild?



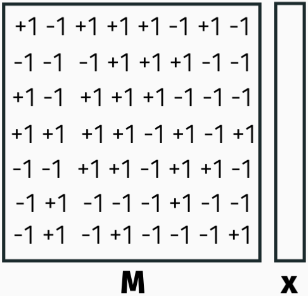
Subsampled Randomized Hadamard Transform (SHRT) (Ailon-Chazelle, 2006)

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

$\mathbf{M}$ should have the property that $\|\mathbf{M}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$ exactly, or is very close. Then we will multiply by a subsampling matrix $\mathbf{S}$ to do the actual dimensionality reduction:

$$\mathbf{\Pi}\mathbf{x} = \mathbf{S}\mathbf{M}\mathbf{x}$$

Good mixing matrices should look random:



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

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

$\mathbf{\Pi} = \mathbf{S}\mathbf{M}$ where $\mathbf{M} = \mathbf{H}\mathbf{D}$:

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

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

1. $\|\mathbf{H}\mathbf{v}\|_2^2 = \|\mathbf{v}\|_2^2$ exactly. Thus $\|\mathbf{H}\mathbf{D}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$
2. $\|\mathbf{H}\mathbf{v}\|_2^2$ can be computed in $O(n \log n)$ time.

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:

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

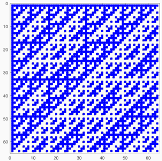
$$H_k = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix}$$

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

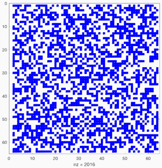
Property 1: For any $k = 0, 1, \ldots$, we have $\|\mathbf{H}_k \mathbf{v}\|_2^2 = \|\mathbf{v}\|_2^2$ for all $\mathbf{v}$. I.e., $\mathbf{H}_k$ is orthogonal.

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

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.

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* $1 - \delta$*,*

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

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:

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

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* $\mathbf{x}$,

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

with probability $(1-\delta)$.

Very little loss in embedding dimension compared to full random matrix, and $\mathbf{\Pi}$ can be multiplied by $\mathbf{x}$ in $O(n \log n)$ (nearly linear) time.

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

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

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

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

This is equivalent to

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

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

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

- $\mathbf{z}_i$ 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:

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

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

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

.

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

Formally, need to 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 call 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.

With probability $1-\delta$, we have that all $\mathbf{z}_i \leq \sqrt{\frac{c\log(n/\delta)}{n}}\|\mathbf{c}\|_2$.

As shown earlier, we can thus guarantee that:

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

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

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

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

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 $(1-\delta)$.

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.

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

Clarkson-Woodruff 2013, STOC Best Paper: Possible to compute $\tilde{\mathbf{A}}$ with poly(d) rows in:

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

$\mathbf{\Pi}$ 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:

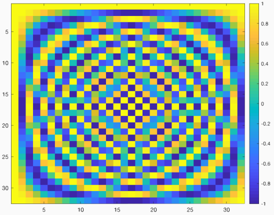
- Vector dimensionality reduction
- Linear algebra
- Locality sensitive hashing (SimHash)
- Randomized kernel learning methods (we will discuss after Thanksgiving)

```
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);
```

WHAT WERE AILON AND CHAZELLE THINKING?

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

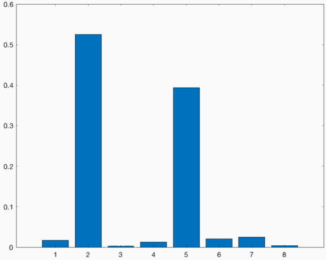
$$\mathbf{F}_{j,k} = e^{-2\pi i \frac{j \cdot k}{n}}, \qquad \mathbf{F}^*\mathbf{F} = \mathbf{I}.$$



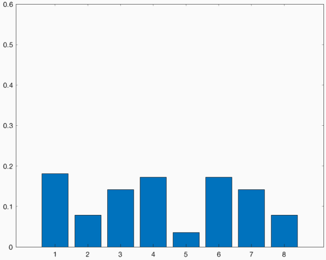
Real part of $\mathbf{F}_{j,k}$.

$\mathbf{F}\mathbf{y}$ 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 Uncertainty Principal (informal): A function and it's Fourier transform cannot both be concentrated.



Vector **y**.



Fourier transform **Fy**.

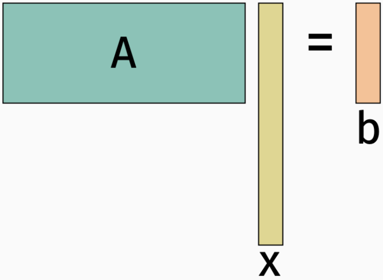
What do we know?

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.

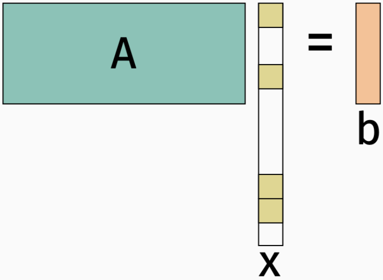
Underdetermined linear regression: Given $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m < n$, $\mathbf{b} \in \mathbb{R}^m$. Assume $\mathbf{b} = \mathbf{A}\mathbf{x}$ for some $\mathbf{x} \in \mathbb{R}^n$.



- Infinite possible solutions $\mathbf{y}$ to $\mathbf{A}\mathbf{y} = \mathbf{b}$, so in general, it is impossible to recover parameter vector $\mathbf{x}$ from the data $\mathbf{A}, \mathbf{b}$.

Underdetermined linear regression: Given $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m < n$, $\mathbf{b} \in \mathbb{R}^m$. Solve $\mathbf{Ax} = \mathbf{b}$ for $\mathbf{x}$.

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



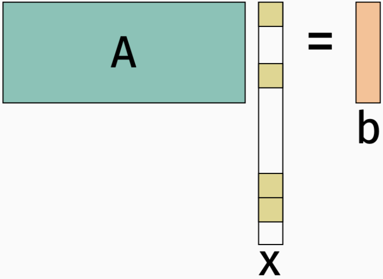
- In many cases can recover $\mathbf{x}$ with $\ll n$ rows. In fact, often $\sim O(k)$ suffice.
- Need additional assumptions about $\mathbf{A}$!

- In statistics and machine learning, we often think about **A**'s rows as data drawn from some universe/distribution:

	bedrooms	bathrooms	sq.ft.	floors	list price	sale price
home 1	2	2	1800	2	200,000	195,000
home 2	4	2.5	2700	1	300,000	310,000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
home n	5	3.5	3600	3	450,000	450,000

- In many othersettings, we will get to choose **A**'s rows. I.e. each $b_i = \mathbf{x}^T \mathbf{a}_i$ for some vector $\mathbf{a}_i$ that we select.
- In this setting, we often call b_i a linear measurement of $\mathbf{x}$ and we call **A** a measurement matrix.

When should this problem be difficult?



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 $\mathbf{A}_i, \mathbf{A}_j$.
  - Recover vectors $\mathbf{x}$ with sparsity $k = 1/\mu$.
- **Focus today**: $\mathbf{A}$ obeys the Restricted Isometry Property.

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

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

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

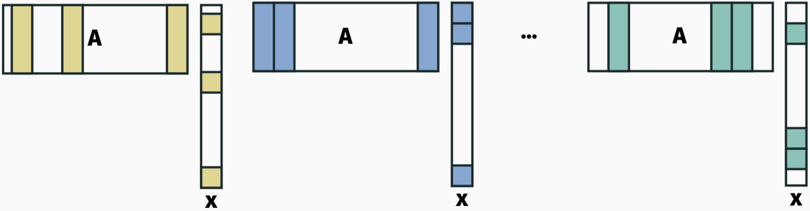
- Johnson-Lindenstrauss type condition.
- $\mathbf{A}$ preserves the norm of all q sparse vectors, instead of the norms of a fixed discrete set of vectors, or all vectors in a subspace (as in subspace embeddings).

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

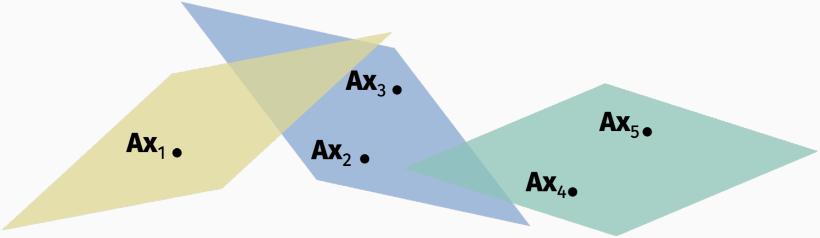
A matrix $\mathbf{A}$ satisfies (q, ϵ)-RIP if, for all $\mathbf{x}$ with $\|\mathbf{x}\|_0 \leq q$,

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

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



Any ideas for how you might prove a random JL matrix with $O(k \log n/\epsilon^2)$ rows satisfies (q, ϵ)-RIP?



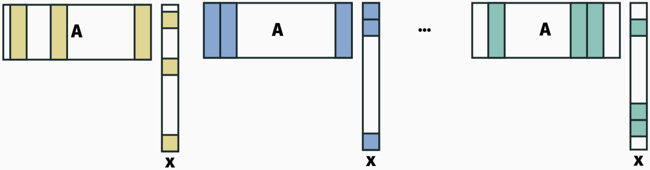
I.e. prove that that random matrix preserves the norm of every $\mathbf{x}$ in this union of subspaces?

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

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

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

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



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:*

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

- 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 next lecture.

Proof:

Important note: Robust versions of this theorem and the others we will discuss exist. 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

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

We will not discuss robustness in detail, but it is a big part of what has made compressed sensing such an active research area in the last 20 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**.

What matrices satisfy this property?

- Random Johnson-Lindenstrauss matrices (Gaussian, sign, etc.) with $m = O(\frac{k\log(n/k)}{\epsilon^2})$ rows are $(O(k), \epsilon)$-RIP.

Some real world data may look random, but this is also a useful observation algorithmically when we want to design **A**.

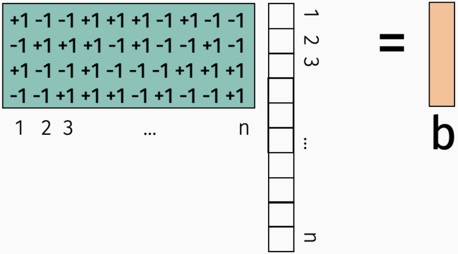
Suppose you view a stream of numbers in $1, \ldots, n$:

$$4, 18, 4, 1, 2, 24, 6, 4, 3, 18, 18, \ldots$$

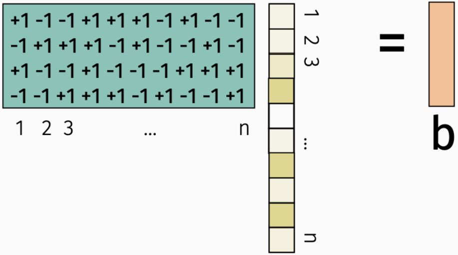
After some time, you want to report which k items appeared most frequently in the stream.

E.g. Amazon is monitoring web-logs to see which product pages people view. They want to figure out which products are viewed most frequently. $n \approx 500$ million.

How can you do this quickly in small space?

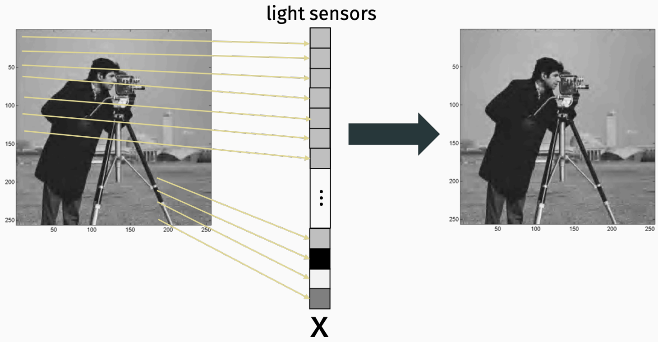


- Every time we receive a number i in the stream, add column A_i to $\mathbf{b}$.



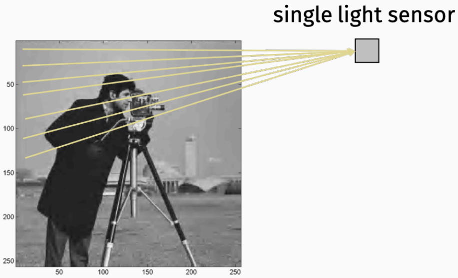
- At the end $\mathbf{b} = \mathbf{A}\mathbf{x}$ for an approximately sparse $\mathbf{x}$ if there were only a few "heavy hitters". Recover $\mathbf{x}$ from $\mathbf{b}$ using a sparse recovery method (like ℓ_0 minimization).

Typical acquisition of image by camera:



Requires one image sensor per pixel captured.

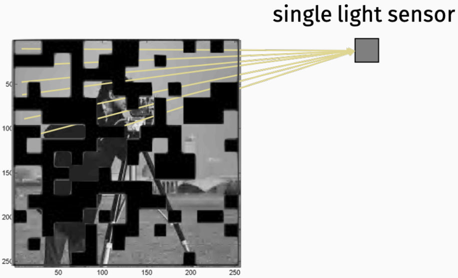
Compressed acquisition of image:



$$p = \sum_{i=1} x_i = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Does not provide very much information about the image.

But several random linear measurements do!



$$p = \sum_{i=1} R_i x_i = \begin{bmatrix} 0 & 1 & 0 & 0 \ldots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Applications in:

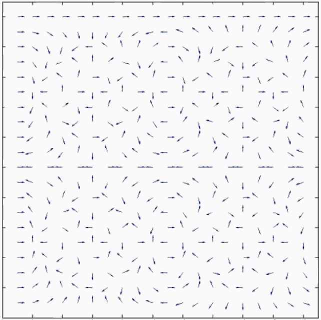
- Imaging outside of the visible spectrum (more expensive sensors).
- Microscopy.
- Other scientific imaging.

Compressed sensing theory does not exactly describe the problem, but has been very valuable in modeling it.

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

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

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



Set $\mathbf{A}$ to contain a random $\approx \tilde{O}(k \log n)$ random rows of this matrix.

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

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

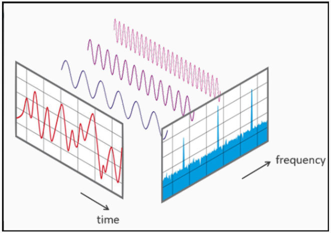
$$(1-\epsilon)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1+\epsilon)\|\mathbf{x}\|_2^2.$$

Uniformly subsampled Discrete Fourier matrices with $m \sim O\left(\frac{k \log^2 k \log n}{\epsilon^2}\right)$ rows $(O(k), \epsilon)$-RIP. [Haviv, Regev, 2016].

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

Might be believable based on our analysis of the subsampled Hadamard matrix, which is closely related ot the Discrete Fourier matrix.

$\mathbf{Fx}$ is the Discrete Fourier Transform of the vector $\mathbf{x}$ (what an FFT computes).



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

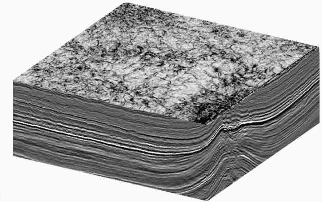
Because $\mathbf{F}^*\mathbf{F} = \mathbf{I}$, $\mathbf{F}^*\mathbf{Fx} = \mathbf{x}$, so we can recover $\mathbf{x}$ if we have access to its DFT, $\mathbf{Fx}$.

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

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

Warning: very cartoonish explanation of very complex problem.

Understanding what material is beneath the crust:



Think of vector $\mathbf{x}$ as scalar values of the density/reflectivity in a single vertical core of the earth.

How do we measure entries of Fourier transform $\mathbf{F}\mathbf{x}$?

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

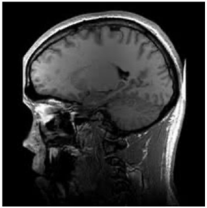
Vibroseis Truck

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

Killer app: Oil Exploration.

Warning: very cartoonish explanation of very complex problem.

Medical Imaging (MRI)



Vector $\mathbf{x}$ here is a 2D image. Everything works with 2D Fourier transforms.

How do we measure entries of Fourier transform $\mathbf{F}\mathbf{x}$?

Blast body with sounds waves waves of varying frequencies.

The fewer measurements we need from Fx, the faster we can acquire and image.

- Especially important when trying to capture something moving (e.g. lungs, baby, child who can't sit still).
- Can also cut down on power requirements (which for MRI machines are huge).

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

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

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

Lots of other random matrices satisfy RIP as well.

One major theoretical question is if we can deterministically construct good RIP matrices. Interestingly, if we want $(O(k), O(1))$ RIP, we can only do so with $O(k^2)$ rows (now very slightly better – thanks to Bourgain et al.).

Whether or not a linear dependence on k is possible with a deterministic construction is unknown.

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* $\mathbf{x}$*. If* $\mathbf{A}$ *is* $(2k, \epsilon)$*-RIP for any* $\epsilon < 1$ *then* $\mathbf{x}$ *is the unique minimizer of:*

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

Algorithm question: Can we recover $\mathbf{x}$ using a faster method? Ideally in polynomial time.

Convex relaxation of the ℓ_0 minimization problem:

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

$$\min_{\mathbf{z}} \|\mathbf{z}\|_1 \qquad \textit{subject to} \qquad \mathbf{A}\mathbf{z} = \mathbf{b}.$$

- Objective is convex:
- Optimizing over convex set:

What is one method we know for solving this problem?

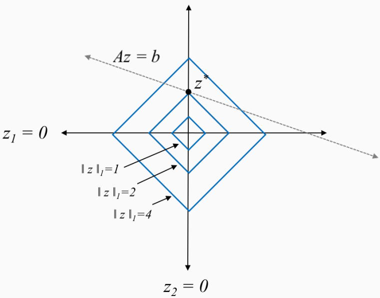
Equivalent formulation:

Problem (Basis Pursuit Linear Program.)

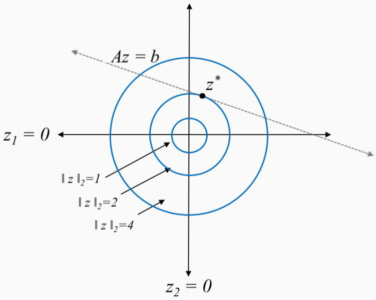
$$\min_{\mathbf{w},\mathbf{z}} \mathbf{1}^T\mathbf{w} \qquad \textit{subject to} \qquad \mathbf{A}\mathbf{z} = \mathbf{b}, -\mathbf{w} \leq \mathbf{z} \leq \mathbf{w}.$$

Can be solved using any algorithm for linear programming. An Interior Point Method will run in at worst $\sim O(n^{3.5})$ time.

Suppose 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.

Theorem

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

Similar proof to ℓ_0 minimization:

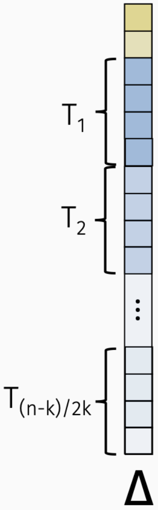
- By way of contradiction, assume $\mathbf{x}$ is not the optimal solution. Then there exists some non-zero Δ such that:
  - $\|\mathbf{x} + \Delta\|_1 \leq \|\mathbf{x}\|_1$
  - $\mathbf{A}(\mathbf{x} + \Delta) = \mathbf{A}\mathbf{x}$. I.e. $\mathbf{A}\Delta = 0$.

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

Only one tool needed:

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

Some definitions:



Claim 1: $\|\Delta_S\|_1 \geq \|\Delta_{\bar{S}}\|_1$

Claim 2: $\|\Delta_S\|_2 \geq \sqrt{2}\sum_{j\geq 2}\|T_j\|_2$:

Finish up proof by contradiction:

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

- **Iterative Hard Thresholding**: Looks a lot like projected gradient descent. Solve $\min_{\mathbf{z}} \|\mathbf{A}\mathbf{z} - \mathbf{b}\|$ 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.

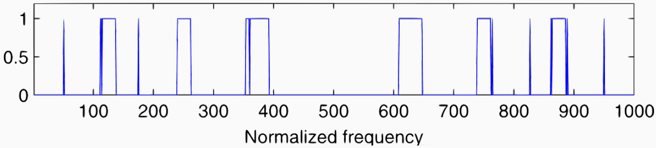
When $\mathbf{A}$ is a subsampled Fourier matrix, there are now methods that run in $O(k \log^c n)$ time [Hassanieh, Indyk, Kapralov, Katabi, Price, Shi, etc. 2012+].

Hold up...

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.