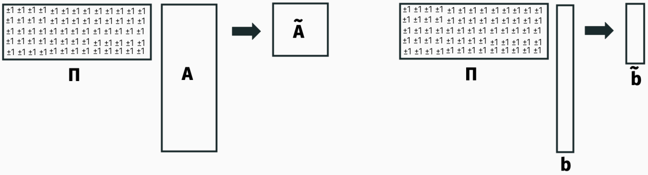
CS-GY 6763: Lecture 13
Fast Johnson-Lindenstrauss Transform, Sparse Recovery and Compressed Sensing

NYU Tandon School of Engineering, Prof. Christopher Musco

RANDOMIZED NUMERICAL LINEAR ALGEBRA

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



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 (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.

Theorem (Second Example: Randomized Low-Rank Approximation[1])

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

$$\|\mathbf{A} - \mathbf{A}\tilde{\mathbf{V}}_k\tilde{\mathbf{V}}_k^T\|_2^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_2^2$$

where $\tilde{\mathbf{V}}_k$ *contains the top k right singular vectors of* $\tilde{\mathbf{A}}$*.*

Reduce from a $O(ndk)$ time computation to an $O(dk^2)$ time problem.

[1] See e.g. Sarlos, 2006 or Halko, Martinson, Tropp, 2011.

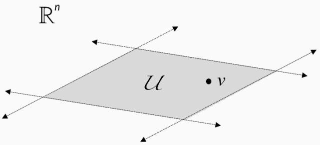
Key Ingredient:

Theorem (Subspace Embedding JL)

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

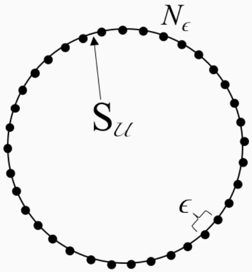
$$(1-\epsilon)\|\mathbf{v}\|_2^2 \leq \|\mathbf{\Pi}\mathbf{v}\|_2^2 \leq (1+\epsilon)\|\mathbf{v}\|_2^2$$

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



Proof idea: Construct ϵ-net, N_ϵ, for the unit sphere, S.

1. Prove that $\|\mathbf{\Pi}\mathbf{w}\|_2^2 = (1 \pm \epsilon)\|\mathbf{w}\|_2^2$ for all $\mathbf{w} \in N_\epsilon$ using union bound.
2. Use a direct argument to extend to the rest of sphere.



Lemma (ϵ-net for the sphere)

Let S be a d dimensional union sphere. For any $\epsilon \leq 1$, there exists a set $N_\epsilon \subset S$ with $|N_\epsilon| = \left(\frac{3}{\epsilon}\right)^d$ such that $\forall \mathbf{v} \in S$,

$$\min_{\mathbf{w} \in N_\epsilon} \|\mathbf{v} - \mathbf{w}\|_2 \leq \epsilon.$$

We skipped the proof of this last time.

We will prove it using a common technique known as a "volume" argument.

Lemma (ϵ-net for the sphere)

Let S be a d dimensional union sphere. For any $\epsilon \leq 1$, there exists a set $N_\epsilon \subset S$ with $|N_\epsilon| = \left(\frac{3}{\epsilon}\right)^d$ such that $\forall \mathbf{v} \in S$,

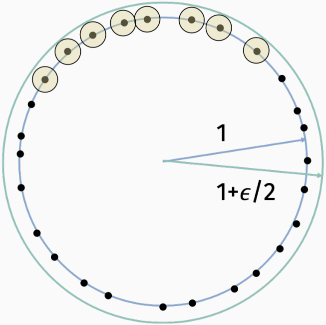
$$\min_{\mathbf{w} \in N_\epsilon} \|\mathbf{v} - \mathbf{w}\|_2 \leq \epsilon.$$

Imaginary algorithm for constructing N_ϵ:

- Set $N_\epsilon = \{\}$
- While such a point exists, choose an arbitrary point $\mathbf{v} \in S$ where there is no $\mathbf{w} \in N_\epsilon$ with $\|\mathbf{v} - \mathbf{w}\| \leq \epsilon$.
- Add $\mathbf{v}$ to N_ϵ.

After running this procedure, we have $N_\epsilon = \{\mathbf{w}_1, \ldots, \mathbf{w}_{|N_\epsilon|}\}$ and $\min_{\mathbf{w} \in N_\epsilon} \|\mathbf{v} - \mathbf{w}\| \leq \epsilon$ for all $\mathbf{v} \in S$ as desired.

How many steps does this procedure take?



Can place a ball of radius $\epsilon/2$ around each $\mathbf{w}_i$ without intersecting any other balls. All of these balls live in a ball of radius $1 + \epsilon/2$.

Volume of d dimensional ball of radius r is

$$\text{vol}(d, r) = c \cdot r^d,$$

where c is a constant that depends on d, but not r. From previous slide we have:

$$\begin{aligned}
\text{vol}(d, \epsilon/2) \cdot |N_\epsilon| &\leq \text{vol}(d, 1 + \epsilon/2) \\
|N_\epsilon| &\leq \frac{\text{vol}(d, 1 + \epsilon/2)}{\text{vol}(d, \epsilon/2)} \\
&\leq \left(\frac{1 + \epsilon/2}{\epsilon/2}\right)^d \leq \left(\frac{3}{\epsilon}\right)^d
\end{aligned}$$

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* $\mathbf{b} \in \mathbb{R}^n$*,*

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

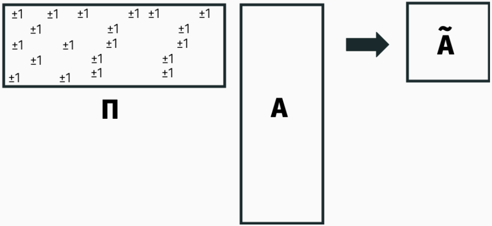
where $\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{\Pi}\mathbf{A}\mathbf{x} - \mathbf{\Pi}\mathbf{b}\|_2^2$.

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.

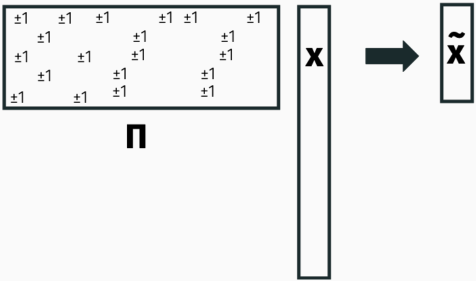


Typically using sparse or structured matrices instead of fully random JL matrices.

Useful in many other applications two. For example, faster methods are often used in LSH systems to implement SimHash.

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:

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



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 $\mathbf{x}$.

Subsampled Randomized Hadamard Transform[2] (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* $\mathbf{x}$*,*

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

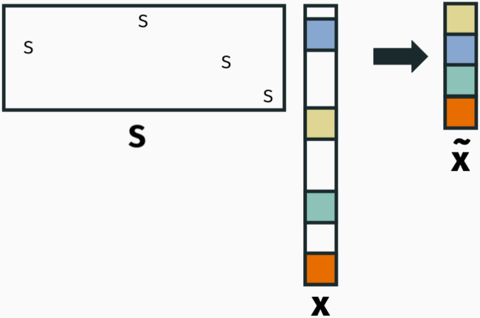
with probability $(1-\delta)$ *and* $\mathbf{\Pi x}$ *can be computed in* $O(n \log n)$ *(nearly linear) time.*

Very little loss in embedding dimension compared to standard JL.

[2] One of my favorite randomized algorithms.

SOLUTION FOR "FLAT" VECTORS

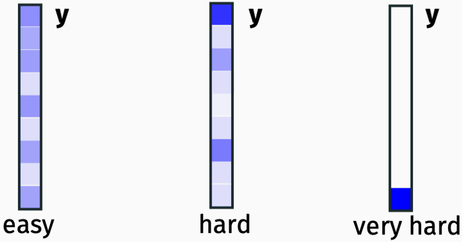
Let $\mathbf{S}$ be a random sampling matrix. Every row contains a value of $s = \sqrt{n/m}$ in a single location, and is zero elsewhere.



If we take m samples, $\tilde{\mathbf{x}}$ can be computed in $O(m)$ time. Woohoo!

What is the problem with this approach?

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



Claim

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

This just follows from standard Hoeffding inequality.

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

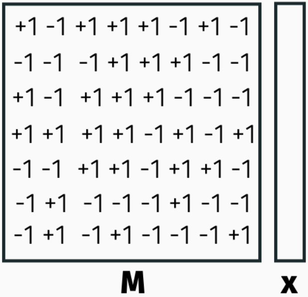
$\mathbf{M}$ will have the properties that

1. $\|\mathbf{M}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$ exactly.
2. Every entry in $\mathbf{M}\mathbf{x}$ is bounded. I.e. $[\mathbf{M}\mathbf{x}]_i^2 \leq \frac{c}{n}\|\mathbf{M}\mathbf{x}\|_2^2$ 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:

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

Good mixing matrices should look random:



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

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

Good mixing matrices should look random:

$$\underbrace{\begin{bmatrix} +1 & -1 & +1 & +1 & +1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 & +1 & +1 & -1 & -1 \\ +1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 \\ +1 & +1 & +1 & +1 & -1 & +1 & -1 & +1 \\ -1 & -1 & +1 & +1 & -1 & +1 & +1 & -1 \\ -1 & +1 & -1 & -1 & -1 & +1 & -1 & -1 \\ -1 & +1 & -1 & +1 & -1 & -1 & -1 & +1 \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \end{bmatrix}}_{\mathbf{x}}$$

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

Subsampled Randomized Hadamard Transform

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

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

The Hadarmard matrix is an 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 $\|\mathbf{HDx}\|_2^2 = \|\mathbf{x}\|_2^2$
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"/

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:

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

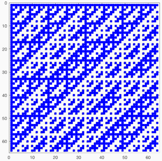
$$H_k = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix}$$

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

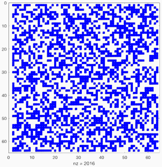
Property 1: For any $k = 0, 1, \ldots$, we have $\|\mathbf{H}_k \mathbf{v}\|_2^2 = \|\mathbf{v}\|_2^2$ for all $\mathbf{v}$. I.e., $\mathbf{H}_k$ is orthogonal.

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.

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 $\boldsymbol{\Pi} = \mathbf{SHD} \in \mathbb{R}^{m \times n}$ *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* $\mathbf{x}$*,*

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

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

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

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

- 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 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,

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

.

- 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.

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: 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:

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

$\mathbf{\Pi}$ 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:

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

Simple, inspired algorithm that has been used for accelerating:

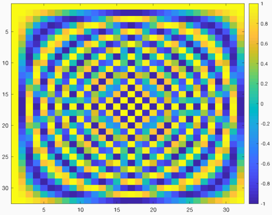
- Vector dimensionality reduction
- Linear algebra
- Locality sensitive hashing (SimHash)
- Randomized kernel learning methods.

```
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);
```

BREAK

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

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

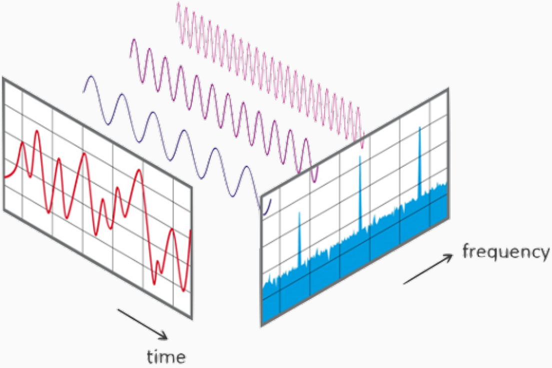


Real part of $\mathbf{F}_{j,k}$.

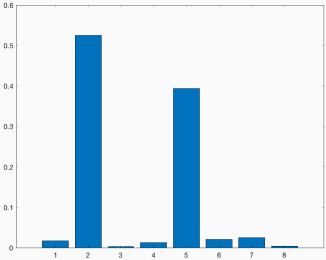
$\mathbf{F}\mathbf{y}$ computes the Discrete Fourier Transform of the vector $\mathbf{y}$. Can be computed in $O(n \log n)$ time using a divide and conquer algorithm (the Fast Fourier Transform).

The 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 **F** looks like a cosine wave with frequency $2\pi j$.

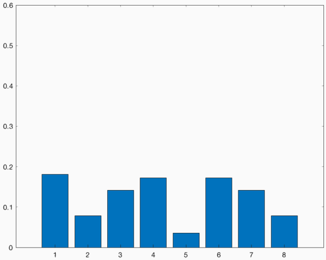
Computing **Fx** computes inner products of **x** with a bunch of different frequencies, which can be used to decompose the vector into a sum of those frequencies.



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



Vector **y**.



Fourier transform **Fy**.

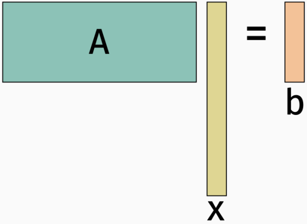
Sampling does not preserve norms, i.e. $\|\mathbf{S}\mathbf{y}\|_2 \not\approx \|\mathbf{y}\|_2$ when $\mathbf{y}$ has a few large entries.

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.

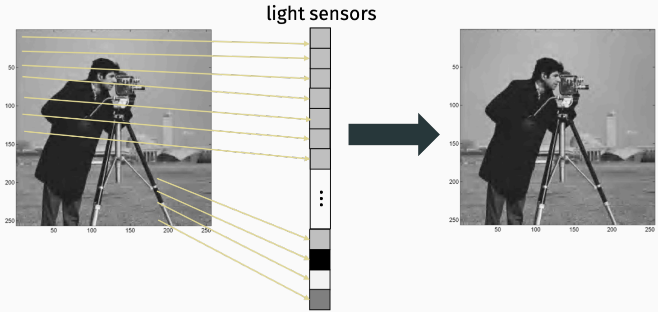
Goal: Recover a vector $\mathbf{x}$ from linear measurements.

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



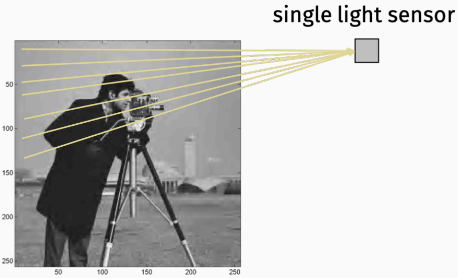
- Infinite possible solutions $\mathbf{y}$ to $\mathbf{A}\mathbf{y} = \mathbf{b}$, so in general, it is impossible to recover $\mathbf{x}$ from $\mathbf{b}$.
- Can often be possible if $\mathbf{x}$ has additional structure!

Typical acquisition of image by camera:



Requires one image sensor per pixel captured.

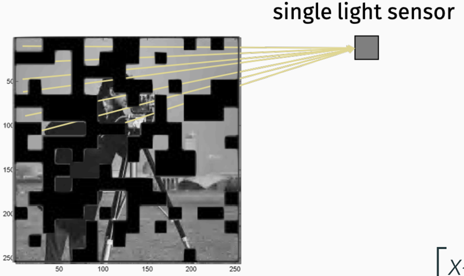
Compressed acquisition of image:



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

Does not provide very much information about the image.

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



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

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

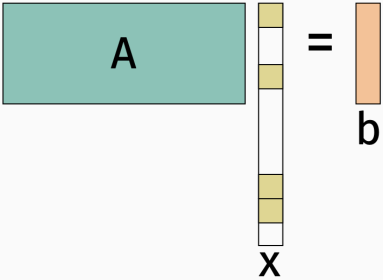
Applications in:

- 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.

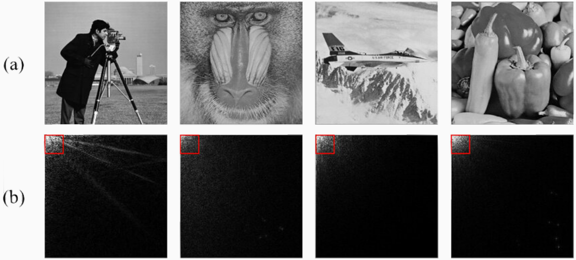
Need to make some assumption to solve the problem. Given $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m < n$, $\mathbf{b} \in \mathbb{R}^{m}$, want to recover $\mathbf{x}$.

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



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

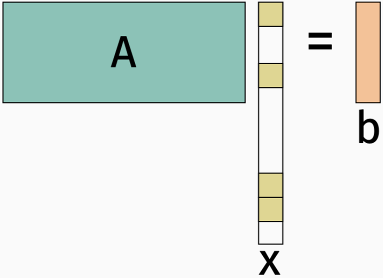
Is sparsify a reasonable assumption?



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 $\mathbf{V}\mathbf{x}$ 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.

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.

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.

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 shortly.

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

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

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

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

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 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?

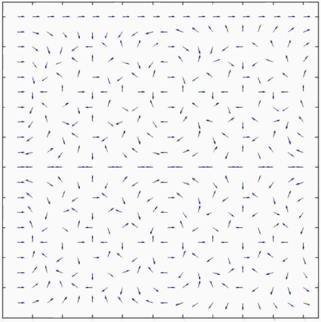
- Random Johnson-Lindenstrauss matrices (Gaussian, sign, etc.) with $m = O(\frac{k \log(n/k)}{\epsilon^2})$ rows are (k, ϵ)-RIP.

Some real world data may look random, but this is also a useful observation algorithmically when we want to design $\mathbf{A}$.

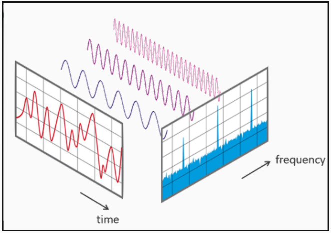
The $n \times n$ discrete Fourier matrix $\mathbf{F}$ is defined:

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

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



In many applications can compute measurements of the form $\mathbf{A}\mathbf{x} = \mathbf{S}\mathbf{F}\mathbf{x}$, 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: $[\mathbf{F}\mathbf{x}]_j$ is the component with frequency j/n.

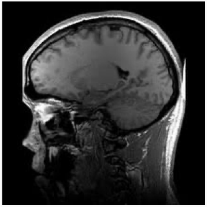
If $\mathbf{A} = \mathbf{SF}$ is a subset of rows from $\mathbf{F}$, then $\mathbf{Ax}$ is a subset of random frequency components from $\mathbf{x}$'s discrete Fourier transform.

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

APPLICATION: MEDICAL IMAGING

Warning: very cartoonish explanation of very complex problem.

Medical Imaging (MRI)

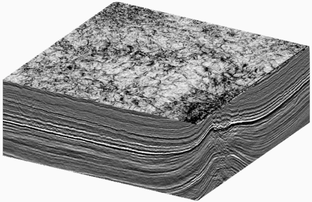



How do we measure entries of Fourier transform $\mathbf{F}\mathbf{x}$? Blast the body with sounds waves of varying frequency.

- 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.

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 explorations, vibrations from drilling, etc. The fewer measurements we need from $\mathbf{F}\mathbf{x}$, the cheaper and faster our data acquisition process becomes.

Setting **A** to contain a random $m \sim O\left(\frac{k \log^2 k \log n}{\epsilon^2}\right)$ rows of the discrete Fourier matrix **F** yields a matrix that with high probability satisfies (k, ϵ)-RIP. [Haviv, Regev, 2016].

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

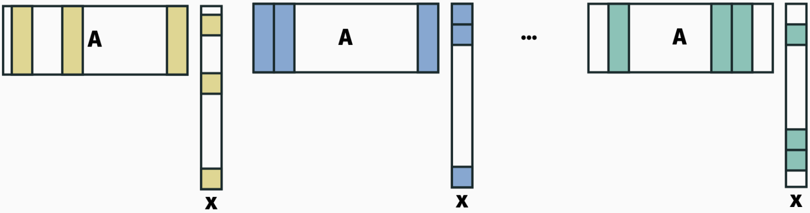
Proving this requires similar tools to analyzing subsampled Hadamard transforms!

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

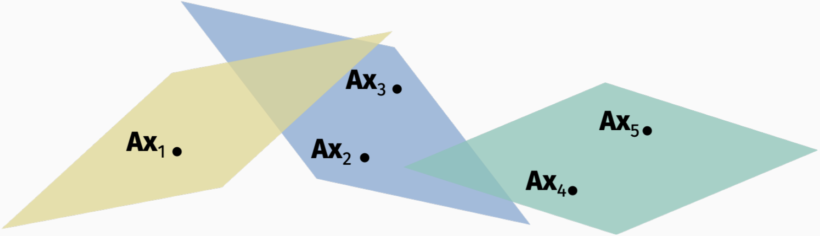
A matrix $\mathbf{A}$ satisfies (q, ϵ)-RIP if, for all $\mathbf{x}$ with $\|\mathbf{x}\|_0 \leq q$,

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

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:



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 $1 - \delta$,

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

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: