

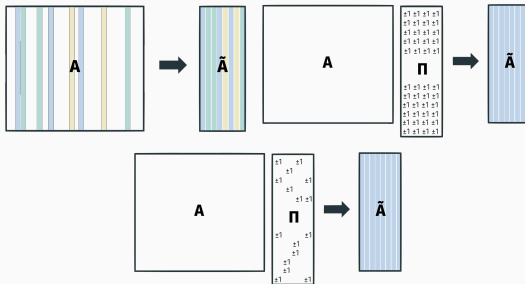
CS-GY 6763: Lecture 12

Fast Johnson-Lindenstrauss Transform, Sparse Recovery and Compressed Sensing

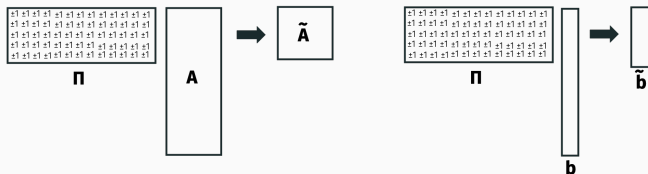
NYU Tandon School of Engineering, Prof. Christopher Musco

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{A} called a “sketch” or “coreset” for A .



Randomized approximate regression using a Johnson-Lindenstrauss Matrix:



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

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

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

Theorem (Randomized Linear Regression)

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

RUNTIME CONSIDERATION

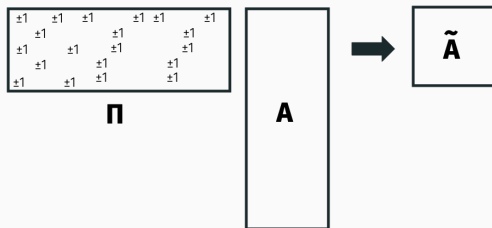
For $\epsilon, \delta = O(1)$, we need $\mathbf{\Pi}$ to have $m = O(d)$ rows.

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

RUNTIME CONSIDERATION

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

Goal: Develop faster Johnson-Lindenstrauss projections.

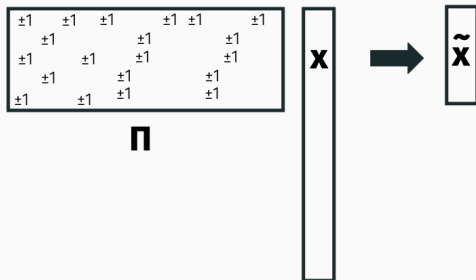


Typically using sparse or structured matrices instead of fully random JL matrices.

RETURN TO SINGLE VECTOR PROBLEM

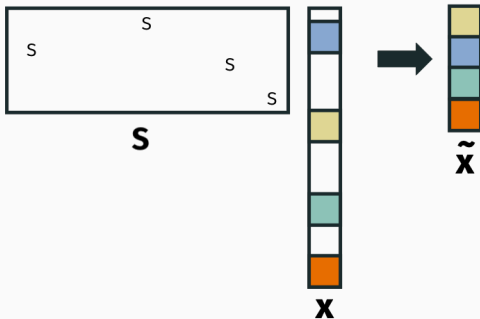
Goal: Develop methods that reduce a vector $\mathbf{x} \in \mathbb{R}^n$ down to $m \approx \frac{\log(1/\delta)}{\epsilon^2}$ dimensions in $o(mn)$ time and guarantee:

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



SOLUTION FOR “FLAT” VECTORS

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



Claim

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

Subsampled Randomized Hadamard Transform (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. **Leverages the simple sampling result from above.**

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

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

THE FAST JOHNSON-LINDENSTRAUSS TRANSFORM

Good mixing matrices should look random:

$$\begin{array}{|c|} \hline +1 & -1 & +1 & +1 & +1 & -1 & +1 & -1 \\ \hline -1 & -1 & -1 & +1 & +1 & +1 & -1 & -1 \\ \hline +1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 \\ \hline +1 & +1 & +1 & +1 & -1 & +1 & -1 & +1 \\ \hline -1 & -1 & +1 & +1 & -1 & +1 & +1 & -1 \\ \hline -1 & +1 & -1 & -1 & -1 & +1 & -1 & -1 \\ \hline -1 & +1 & -1 & +1 & -1 & -1 & -1 & +1 \\ \hline \end{array} \quad \begin{array}{|c|} \hline \\ \hline \\ \hline \\ \hline \\ \hline \\ \hline \\ \hline \end{array}$$

M **x**

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

$\Pi = SM$ where $M = HD$:

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

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

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

HADAMARD MATRICES RECURSIVE DEFINITION

Assume that n is a power of 2. For $k = 0, 1, \dots$, 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 $\pm \frac{1}{\sqrt{n}}$.

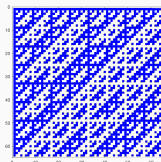
HADAMARD MATRICES ARE ORTHOGONAL

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

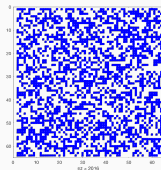
Property 2: Can compute $\Pi x = SHDx$ in $O(n \log n)$ time.

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.

Lemma (SHRT mixing lemma)

Let \mathbf{H} be an $(n \times n)$ Hadamard matrix and \mathbf{D} a random ± 1 diagonal matrix. Let $\mathbf{z} = \mathbf{H}\mathbf{D}\mathbf{x}$ 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 $\|\mathbf{S}\mathbf{z}\|_2^2 \approx \|\mathbf{z}\|_2^2$. I.e. that:

$$\|\mathbf{I}\mathbf{x}\|_2^2 = \|\mathbf{S}\mathbf{H}\mathbf{D}\mathbf{x}\|_2^2 \approx \|\mathbf{x}\|_2^2$$

Our main results then follows directly from our sampling result from earlier:

Theorem (The Fast JL Lemma)

Let $\Pi = \text{SHD} \in \mathbb{R}^{m \times n}$ 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} ,

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

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:

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

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 , $\mathbf{z}_i = \mathbf{h}_i^T \mathbf{D}\mathbf{x} = \frac{1}{\sqrt{n}} \sum_{j=1}^n R_j x_j$.

- \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 ,

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

- 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, \dots, 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 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:

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

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

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

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

Upshot for regression: Compute ΠA in $O(nd \log n)$ time instead of $O(nd^2)$ time. Compress problem down to \tilde{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 class 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:

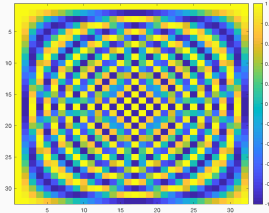
- Vector dimensionality reduction
- Linear algebra
- Locality sensitive hashing (SimHash)
- Randomized kernel learning methods.

```
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);
```

WHAT WERE AILON AND CHAZELLE THINKING?

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

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

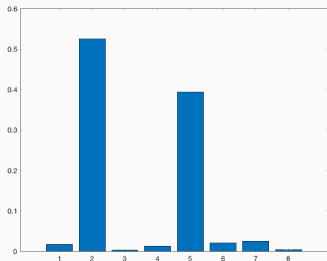


Real part of $F_{j,k}$.

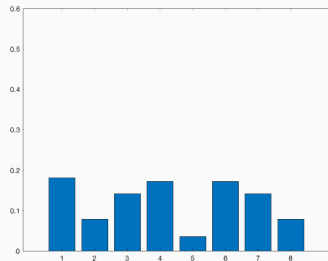
Fy computes the Discrete Fourier Transform of the vector y .
Can be computed in $O(n \log n)$ time using a divide and conquer algorithm (the Fast Fourier Transform).

THE UNCERTAINTY PRINCIPAL

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



Vector y .



Fourier transform Fy .

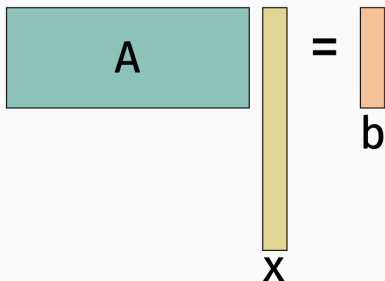
What do we know?

Sampling does not preserve norms, i.e. $\|\mathbf{S}\mathbf{y}\|_2 \neq \|\mathbf{y}\|_2$ 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 $\mathbf{x} \in \mathbb{R}^n$.

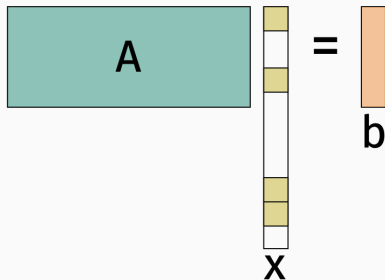


The diagram illustrates the equation $\mathbf{A}\mathbf{x} = \mathbf{b}$. On the left, a teal rectangular box labeled \mathbf{A} is positioned to the left of a tall, thin yellow vertical bar labeled \mathbf{x} . To the right of the yellow bar is an equals sign, followed by a shorter, thin orange vertical bar labeled \mathbf{b} . The yellow bar is taller than the orange bar, visually representing the fact that $n > m$.

- 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} !

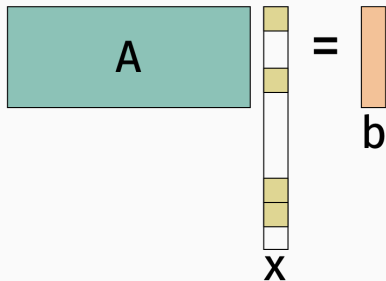
QUICK ASIDE

- In statistics and machine learning, we often think about \mathbf{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 other settings, we will get to choose \mathbf{A} 's rows. I.e. each $b_i = \mathbf{x}^T \mathbf{a}_i$ for some vector \mathbf{a}_i that we select.
- In the later case, we often call b_i a linear measurement of \mathbf{x} and we call \mathbf{A} a measurement matrix.

When should this problem be difficult?



Many ways to formalize our intuition

- **A** has Kruskal rank r . All sets of r columns in **A** are linearly independent.
 - Recover vectors **x** with sparsity $k = r/2$.
- **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 **x** with sparsity $k = 1/\mu$.
- **Focus today:** **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 $\|\mathbf{x}\|_0 \leq q$,

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

- 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).

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 $\mathbf{Ay} = \mathbf{b}$, $\|\mathbf{y}\|_0 \leq \|\mathbf{x}\|_0$.

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 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 $m = O\left(\frac{k \log(n/k)}{\epsilon^2}\right)$ 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 A.

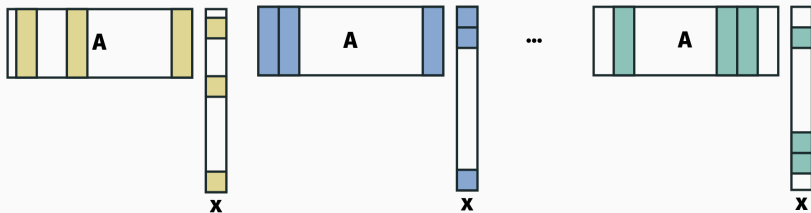
RESTRICTED ISOMETRY PROPERTY

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

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

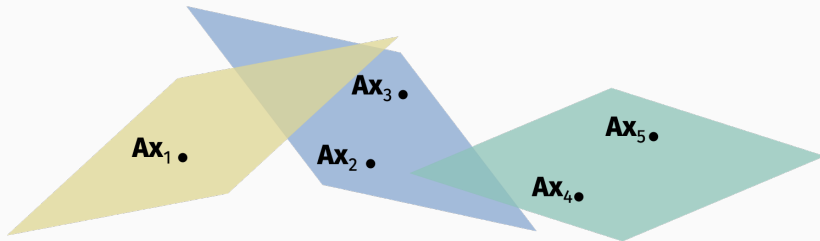
$$(1 - \epsilon)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \epsilon)\|\mathbf{x}\|_2^2.$$

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

Quick argument:

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

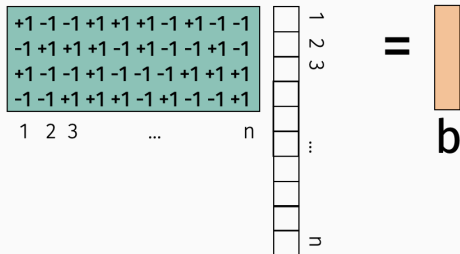
4, 18, 4, 1, 2, 24, 6, 4, 3, 18, 18, ...

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.

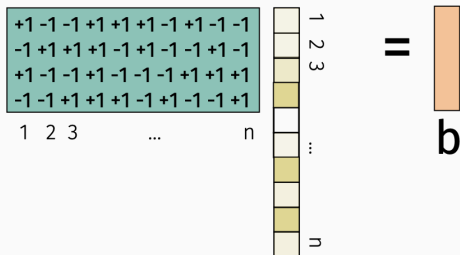
How can you do this quickly in small space?

APPLICATION: HEAVY HITTERS IN DATA STREAMS



- Every time we receive a number i in the stream, add column A_i to b .

APPLICATION: HEAVY HITTERS IN DATA STREAMS



- At the end $b = Ax$ for an approximately sparse x if there were only a few “heavy hitters”. Recover x from b using a sparse recovery method (like ℓ_0 minimization).

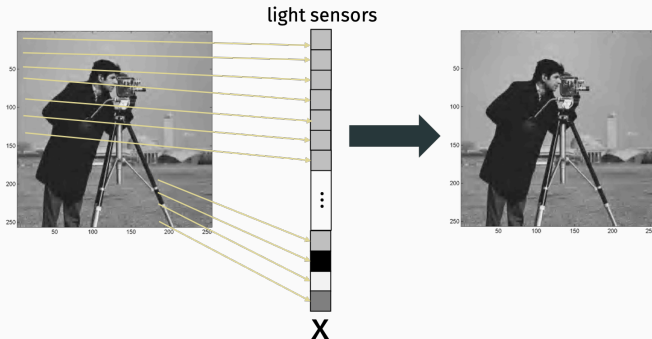
In contrast to the standard implementations of CountMin and related methods, sparse recovery based methods naturally handles both insertions or deletions.

insert(4), insert(18), remove(4), insert(1), insert(2), remove(2) . . .

E.g. Amazon is monitoring what products people add to their “wishlist” and wants a list of most tagged products. Wishlists can be changed over time, including by removing items.

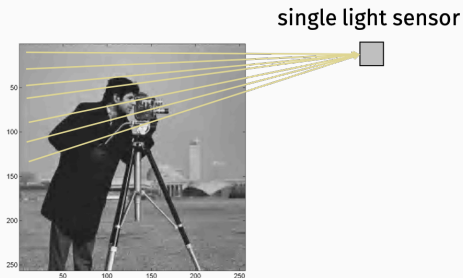
APPLICATION: SINGLE PIXEL CAMERA

Typical acquisition of image by camera:



Requires one image sensor per pixel captured.

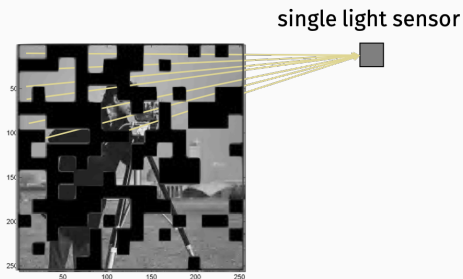
Compressed acquisition of image:



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

Does not provide very much information about the image.

But several random linear measurements do!



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

Applications in:

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

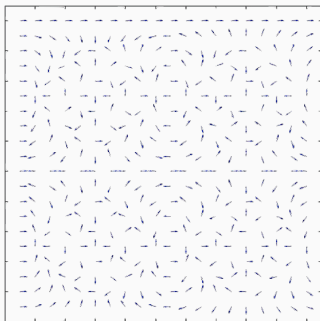
Compressed sensing theory does not exactly describe these problems, but has been very valuable in modeling them.

THE DISCRETE FOURIER MATRIX

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

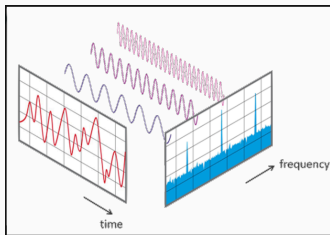
$$F_{j,k} = e^{\frac{-2\pi i}{n}j \cdot k},$$

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



THE DISCRETE FOURIER MATRIX

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



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

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

Setting \mathbf{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 \mathbf{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.

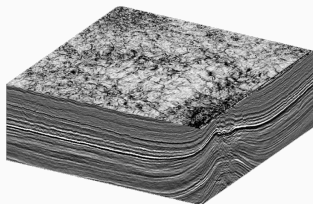
Proving this requires similar tools to analyzing subsampled Hadamard transforms!

If \mathbf{A} is a subset of q 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.

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{Fx} ?

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

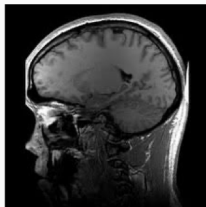


Vibroseis Truck

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

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 $\mathbf{F}\mathbf{x}$?

APPLICATION: GEOPHYSICS

Blast the body with sounds waves of varying frequency.



The fewer measurements we need from F_x , the faster we can acquire an 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 $((q, \epsilon)$ -Restricted Isometry Property)

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

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

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:

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

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.)

$$\min_z \|z\|_1 \quad \text{subject to} \quad Az = b.$$

- Objective is convex.
- Optimizing over convex set.

What is one method we know for solving this problem?

Equivalent formulation:

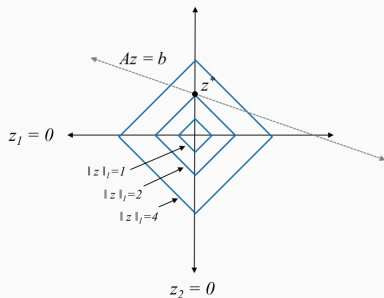
Problem (Basis Pursuit Linear Program.)

$$\min_{w,z} \mathbf{1}^T w \quad \text{subject to} \quad Az = \mathbf{b}, w \geq 0, -w \leq z \leq w.$$

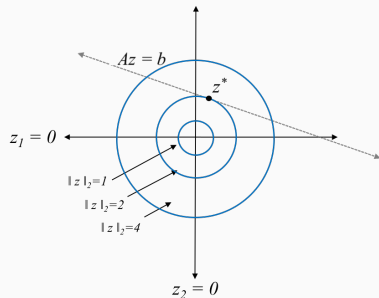
Can be solved using any algorithm for linear programming. An Interior Point Method will run in $\sim O(n^{3.5})$ time.

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.

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.

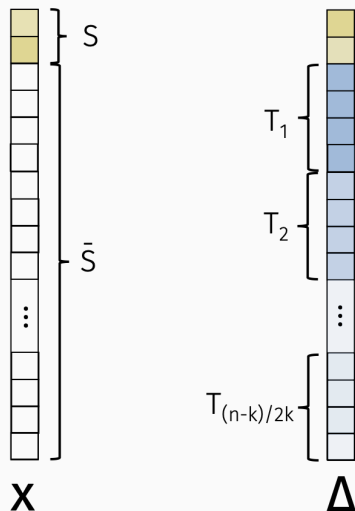
First tool:

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:



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

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

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

Claim: $\|\Delta_{T_j}\|_1 \geq \sqrt{2k} \|\Delta_{T_{j+1}}\|_2$

Finish up proof by contradiction: Recall that \mathbf{A} is assumed to have the $(3k, \epsilon)$ RIP property.

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

A 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 $\min_z \|\mathbf{Az} - \mathbf{b}\|$ 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 Fourier 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+].

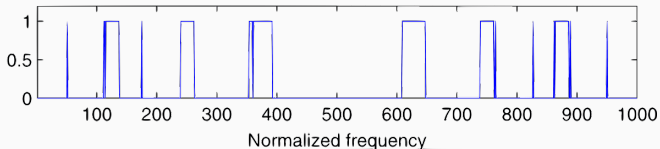
Hold up...

SPARSE FOURIER TRANSFORM

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.