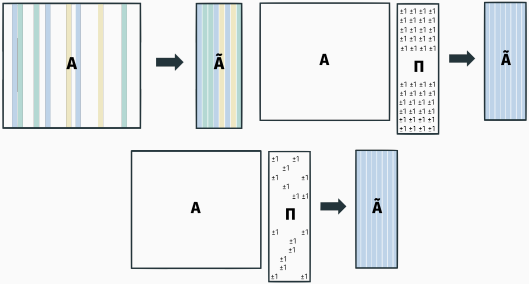
CS-GY 6763: Lecture 12
Fast Johnson-Lindenstrauss Transform, Sparse Recovery and Compressed Sensing

NYU Tandon School of Engineering, Prof. Christopher Musco

RANDOMIZED NUMERICAL LINEAR ALGEBRA

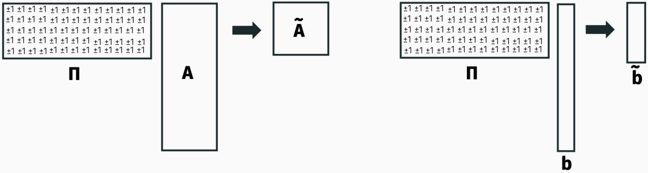
Main idea: If you want to compute singular vectors or eigenvectors, multiply two matrices, solve a regression problem, etc.:

1. Compress your matrices using a randomized method.
2. Solve the problem on the smaller or sparser matrix.
    - Ã called a "sketch" or "coreset" for A.



SKETCHED REGRESSION

Randomized approximate regression using a Johnson-Lindenstrauss Matrix:



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

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

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

main result from last class

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

PROOF APPROACH

- Showed that for <u>all</u> $\mathbf{x}$, $\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\|_2^2 = (1 \pm \epsilon) \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$
- Easy to prove for a single $\mathbf{x}$ using JL lemma.
- To extend to all $\mathbf{x}$ (an finite set) used an ϵ-net argument.

RUNTIME CONSIDERATION

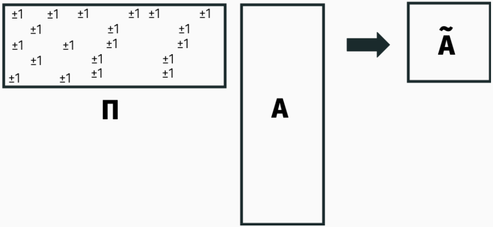
For $\epsilon, \delta = O(1)$, we need $\mathbf{\Pi}$ to have $m = O(d)$ rows.

- 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.

RUNTIME CONSIDERATION

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.



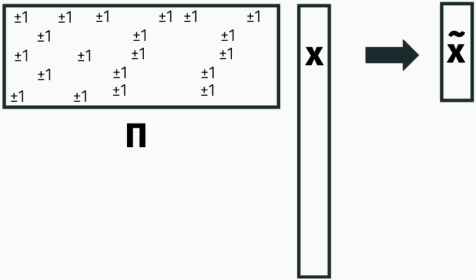
Typically using sparse or structured matrices instead of fully random JL matrices.

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

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:

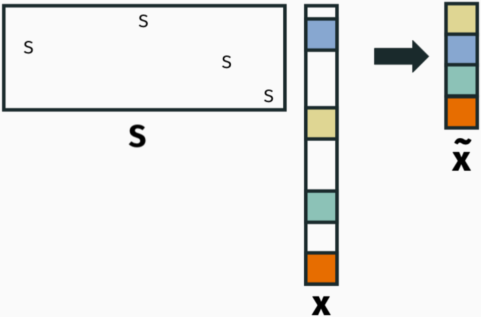
$$(1-\epsilon)\|\mathbf{x}\|_2^2 \leq \|\mathbf{\Pi x}\|_2^2 \leq (1+\epsilon)\|\mathbf{x}\|_2^2$$



Recall that once the bound above is proven, linearity lets use preserve things like $\|\mathbf{y}-\mathbf{z}\|_2^2$ or $\|\mathbf{Aw}-\mathbf{b}\|_2^2$.

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.

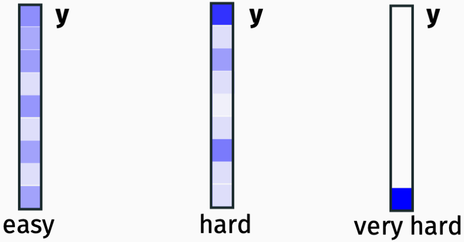


If we take m samples, $\tilde{\mathbf{x}}$ can be computed in $O(m)$ time. Woohoo!

What is the problem with this approach?

VECTOR SAMPLING

Uniform 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.

EXTENSION TO GENERAL VECTORS

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

Theorem (The Fast JL Lemma)

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

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

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

Very little loss in embedding dimension compared to standard JL.
Leverages the simple sampling result from above.

[1] One of my top 3 favorite randomized algorithms.

THE FAST JOHNSON-LINDENSTRAUSS TRANSFORM

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

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

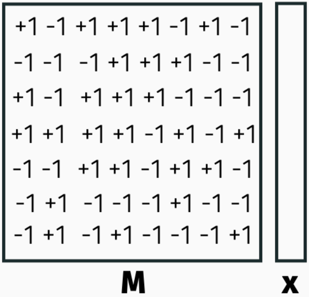
1. $\|\mathbf{M}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$ exactly.
2. Every entry in $\mathbf{M}\mathbf{x}$ is bounded. I.e. $[\mathbf{M}\mathbf{x}]_i^2 \leq \frac{c}{n}\|\mathbf{M}\mathbf{x}\|_2^2$ for some factor c to be determined.
3. We will be able to multiply by $\mathbf{M}$ in $O(n \log n)$ time.

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

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

THE FAST JOHNSON-LINDENSTRAUSS TRANSFORM

Good mixing matrices should look random:



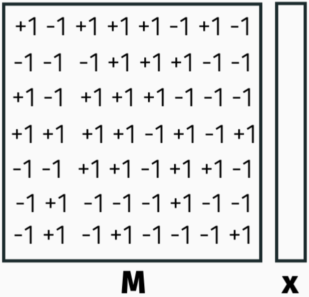
$$\begin{matrix} +1 & -1 & +1 & +1 & +1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 & +1 & +1 & -1 & -1 \\ +1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 \\ +1 & +1 & +1 & +1 & -1 & +1 & -1 & +1 \\ -1 & -1 & +1 & +1 & -1 & +1 & +1 & -1 \\ -1 & +1 & -1 & -1 & -1 & +1 & -1 & -1 \\ -1 & +1 & -1 & +1 & -1 & -1 & -1 & +1 \end{matrix}$$

I claim to mix any x with high probability, M needs to be chosen randomly. Why?

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

THE FAST JOHNSON-LINDENSTRAUSS TRANSFORM

Good mixing matrices should look random:



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

But for this approach to work, we need to be able to compute **Mx** very quickly. So we will use a pseudorandom matrix instead.

Subsampled Randomized Hadamard Transform

$\mathbf{\Pi} = \mathbf{S}\mathbf{M}$ 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 orthogonal 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.

HADAMARD MATRICES RECURSIVE DEFINITION

Assume that n is a power of 2. For $k = 0, 1, \ldots,$ the k^{th} Hadamard matrix 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

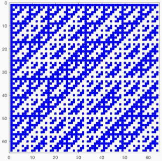
Property 1: For any $k = 0, 1, \ldots$, we have $\|\mathbf{H}_k \mathbf{v}\|_2^2 = \|\mathbf{v}\|_2^2$ for all $\mathbf{v}$. I.e., $\mathbf{H}_k$ is orthogonal.

HADAMARD MATRICES

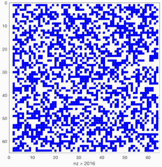
Property 2: Can compute $\mathbf{\Pi x} = \mathbf{SHDx}$ in $O(n \log n)$ time.

RANDOMIZED HADAMARD TRANSFORM

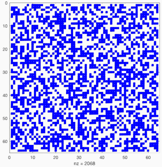
Property 3: The randomized Hadamard matrix is a good "mixing matrix" for smoothing out vectors.



Deterministic Hadamard matrix.



Randomized Hadamard **PHD**.



Fully random sign matrix.

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

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:

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

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

Theorem (The Fast JL Lemma)

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

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

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

RANDOMIZED HADAMARD ANALYSIS

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.

RANDOMIZED HADAMARD ANALYSIS

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.

RANDOMIZED HADAMARD ANALYSIS

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.

RADEMACHER CONCENTRATION

Could use Hoeffding or Bernstein inequality, or a shift, 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.

FINISHING UP

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

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.

linear regression with SHRTs

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

BRIEF COMMENT ON OTHER METHODS

$O(nd \log n)$ 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). Related to Danrong's reading group presentation.

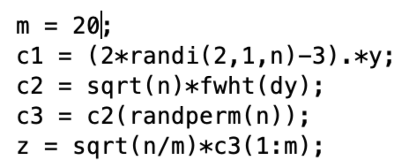
Lead to a whole close of matrix algorithms (for regression, SVD, etc.) which run in time:

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

WHAT WERE AILON AND CHAZELLE THINKING?

Simple, inspired algorithm that has been used for accelerating:

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

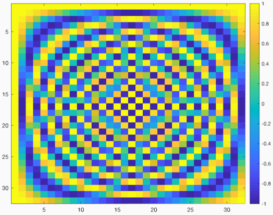


```
m = 20;
c1 = (2*randi(2,1,n)-3).*y;
c2 = sqrt(n)*fwht(dy);
c3 = c2(randperm(n));
z = sqrt(n/m)*c3(1:m);
```

WHAT WERE AILON AND CHAZELLE THINKING?

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

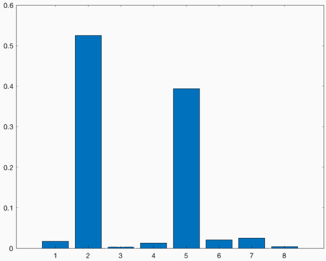
$$\mathbf{F}_{j,k} = e^{-2\pi i \frac{j\cdot k}{n}}, \qquad\qquad \mathbf{F}^*\mathbf{F} = \mathbf{I}.$$



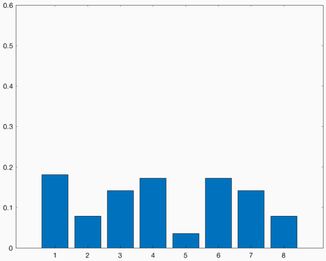
Real part of $\mathbf{F}_{j,k}$.

$\mathbf{F}\mathbf{y}$ 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**.

sparse recovery/compressed sensing

What do we know?

THE UNCERTAINTY PRINCIPAL

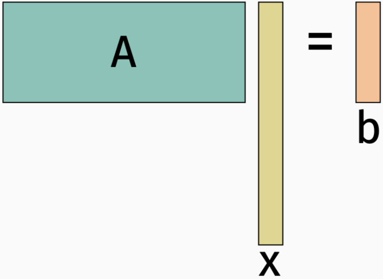
Sampling does not preserve norms, i.e. $\|\mathbf{S}\mathbf{y}\|_2 \not\approx \|\mathbf{y}\|_2$ when $\mathbf{y}$ has a few large entries.

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

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

sparse recovery/compressed sensing problem setup

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 $\mathbf{x} \in \mathbb{R}^{n}$.

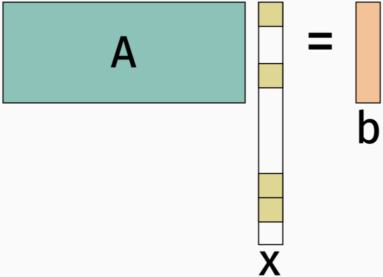


- Infinite possible solutions $\mathbf{y}$ to $\mathbf{A}\mathbf{y} = \mathbf{b}$, so in general, it is impossible to recover parameter vector $\mathbf{x}$ from the data $\mathbf{A}, \mathbf{b}$.

SPARSITY RECOVERY/COMPRESSED SENSING

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

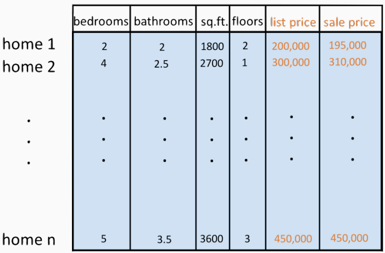
- Assume $\mathbf{x}$ is k-sparse for small k. $\|\mathbf{x}\|_0 = k$.



- 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:

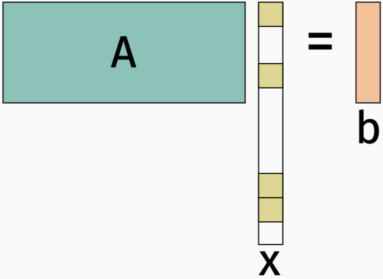


	bedrooms	bathrooms	sq.ft.	floors	list price	sale price
home 1	2	2	1800	2	200,000	195,000
home 2	4	2.5	2700	1	300,000	310,000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
home n	5	3.5	3600	3	450,000	450,000

- 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.

ASSUMPTIONS ON MEASUREMENT MATRIX

When should this problem be difficult?



ASSUMPTIONS ON MEASUREMENT MATRIX

Many ways to formalize our intuition

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

RESTRICTED ISOMETRY PROPERTY

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

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

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

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

first sparse recovery result

Theorem (ℓ_0-minimization)

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

$$\min \|\mathsf{z}\|_0 \qquad \text{subject to} \qquad \mathsf{A}\mathsf{z} = \mathsf{b}.$$

- Establishes that information theoretically we can recover 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 $\min_{\mathbf{A}\mathbf{z}=\mathbf{b}} \|\mathbf{z}\|_0$.

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

ROBUSTNESS

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

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

ROBUSTNESS

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

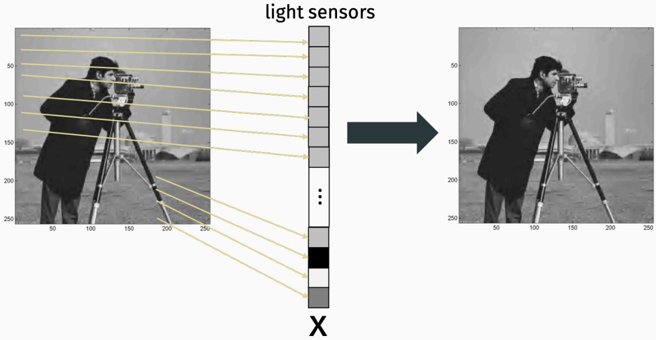
What matrices satisfy this property?

- 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 $\mathbf{A}$.

APPLICATION: SINGLE PIXEL CAMERA

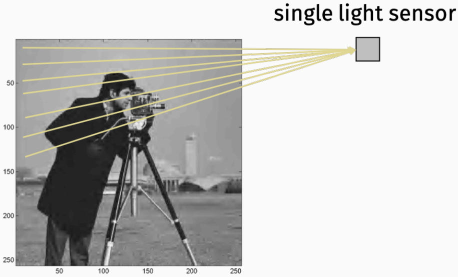
Typical acquisition of image by camera:



Requires one image sensor per pixel captured.

APPLICATION: SINGLE PIXEL CAMERA

Compressed acquisition of image:

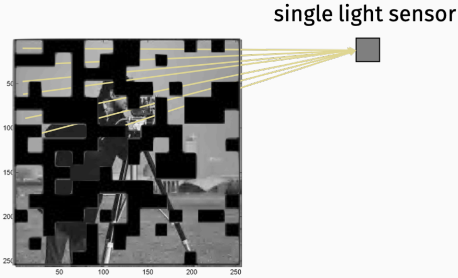


$$p = \sum_{i=1} x_i = \begin{bmatrix} \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Does not provide very much information about the image.

APPLICATION: SINGLE PIXEL CAMERA

But several random linear measurements do!



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

APPLICATION: SINGLE PIXEL CAMERA

Applications in:

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

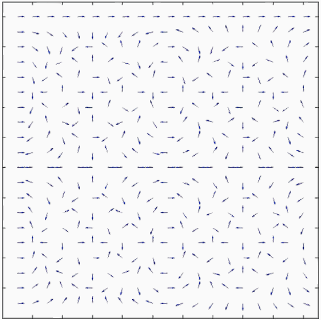
Compressed sensing theory does not exactly describe these problems, but has been very valuable in modeling them.

THE DISCRETE FOURIER MATRIX

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

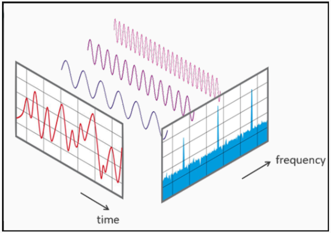
$$F_{j,k} = e^{\frac{-2\pi i}{n} j \cdot k},$$

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



PSEUDORANDOM RIP MATRICES

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.

THE DISCRETE FOURIER MATRIX

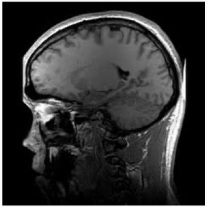
If $\mathsf{A} = \mathsf{SF}$ is a subset of rows from F, then Ax is a subset of random frequency components from x's discrete Fourier transform.

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

APPLICATION: MEDICAL IMAGING

Warning: very cartoonish explanation of very complex problem.

Medical Imaging (MRI)



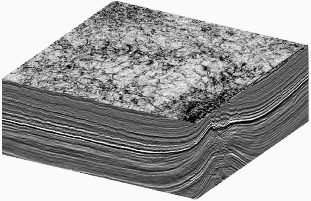
How do we measure entries of Fourier transform $\mathbf{F}\mathbf{x}$? Blast the body with sounds waves of varying frequency.

- 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.

APPLICATION: GEOPHYSICS

Warning: very cartoonish explanation of very complex problem.

Understanding what material is beneath the crust:



APPLICATION: GEOPHYSICS

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.

RESTRICTED ISOMETRY PROPERTY

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!

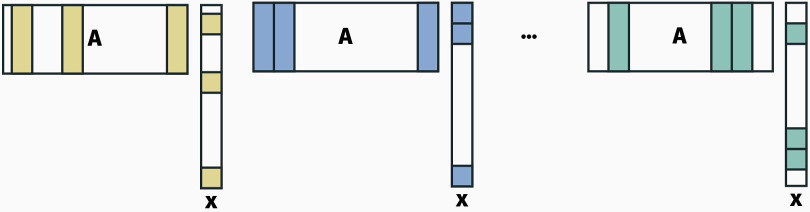
RESTRICTED ISOMETRY PROPERTY

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

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

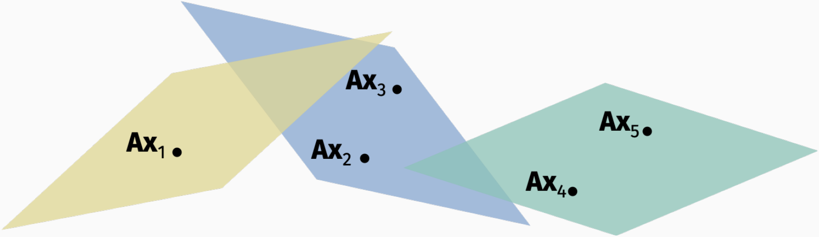
$$(1 - \epsilon)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \epsilon)\|\mathbf{x}\|_2^2.$$

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 $\mathbf{x}$ in this union of subspaces?

RESTRICTED ISOMETRY PROPERTY FROM JL

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:

RESTRICTED ISOMETRY PROPERTY

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

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

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

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:*

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

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

BASIS PURSUIT

Convex relaxation of the ℓ_0 minimization problem:

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

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

- Objective is convex.
- Optimizing over convex set.

What is one method we know for solving this problem?

basis pursuit linear program

Equivalent formulation:

Problem (Basis Pursuit Linear Program.)

$$\min_{\mathbf{w},\mathbf{z}} \mathbf{1}^T\mathbf{w} \qquad \textit{subject to} \qquad \mathbf{A}\mathbf{z} = \mathbf{b}, \mathbf{w} \geq 0, -\mathbf{w} \leq \mathbf{z} \leq \mathbf{w}.$$

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

BASIS PURSUIT ANALYSIS

Theorem

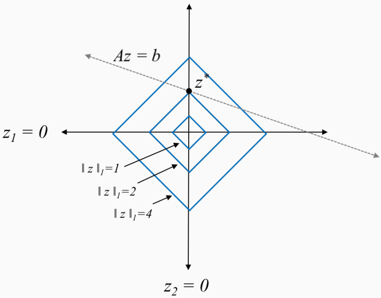
If A *is* $(3k, \epsilon)$*-RIP for* $\epsilon < .17$ *and* $\|\mathbf{x}\|_0 = k$*, then* x *is the unique optimal solution of the Basis Pursuit LP).*

Two surprising things about this result:

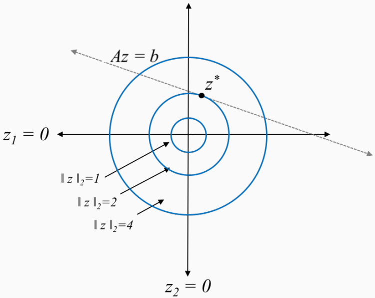
- Exponentially improve computational complexity with only a constant factor overhead in measurement complexity.
- Typical "relax-and-round" algorithm, but rounding is not even necessary! Just return the solution of the relaxed problem.

BASIS PURSUIT INTUITION

Suppose A is 2×1, so **b** is just a scalar and **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.

BASIS PURSUIT ANALYSIS

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.

TOOLS NEEDED

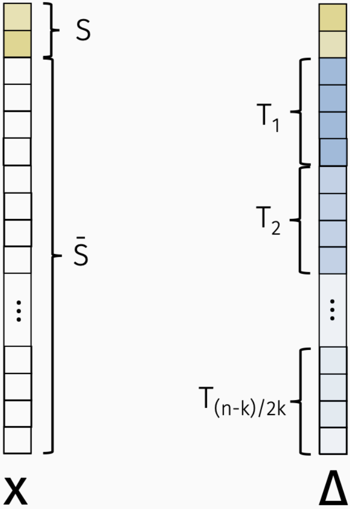
First tool:

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:



BASIS PURSUIT ANALYSIS

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

BASIS PURSUIT ANALYSIS

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

$$\|\boldsymbol{\Delta}_s\|_2 \geq \frac{1}{\sqrt{k}}\|\boldsymbol{\Delta}_S\|_1 \geq \frac{1}{\sqrt{k}}\|\boldsymbol{\Delta}_{\bar{S}}\|_1 = \frac{1}{\sqrt{k}}\sum_{j\geq 1}\|\boldsymbol{\Delta}_{T_j}\|_1.$$

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

BASIS PURSUIT ANALYSIS

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

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

FASTER METHODS

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

FASTER METHODS

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

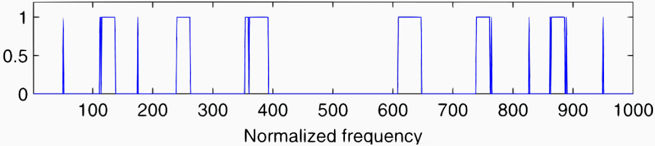
Hold up...

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!

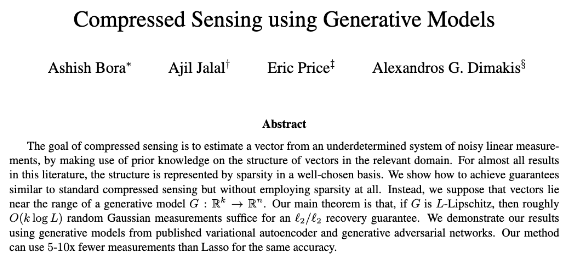
- Randomly subsample $\mathbf{F}\mathbf{x}$.
- Feed that input into our sparse recovery algorithm to extract $\mathbf{x}$.

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



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

COMPRESSED SENSING FROM GENERATIVE MODELS

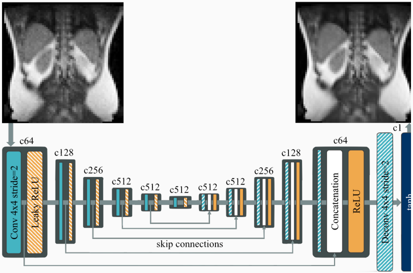


Compressed Sensing using Generative Models

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

Abstract

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



COMPRESSED SENSING FROM GENERATIVE MODELS