CS-GY 6763: Lecture 14
Finish Sparse Recovery and Compressed Sensing, Introduction to Spectral Sparsification

NYU Tandon School of Engineering, Prof. Christopher Musco

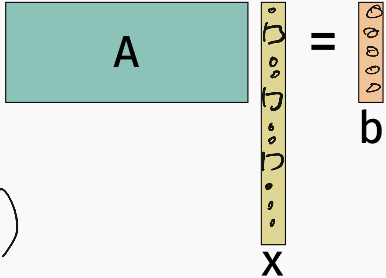
- Final project due next Wednesday, same day as final exam.
- Exam study guide will be released tonight.
- Solutions for last problem sets will be reviewed in office hours.

SPARSE RECOVERY/COMPRESSED SENSING PROBLEM SETUP

- Design a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m < n$, $\mathbf{b} \in \mathbb{R}^m$.
- "Measure" $\mathbf{b} = \mathbf{A}\mathbf{x}$ for some <u>k-sparse</u> $\mathbf{x} \in \mathbb{R}^n$.

$k \ll n$

m

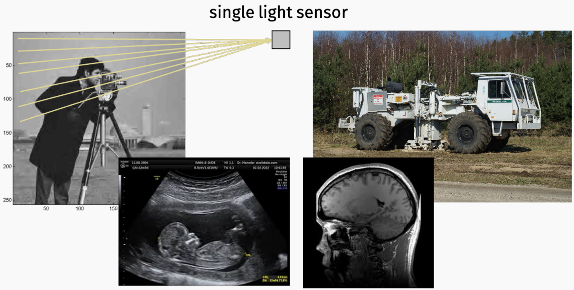


$O(k \log(n))$

- Recover $\mathbf{x}$ from $\mathbf{b}$.

Dual goals: Minimize <u>sample complexity</u> (number of rows in $\mathbf{A}$ and <u>computational complexity</u> to recover $\mathbf{x}$ from $\mathbf{A}, \mathbf{b}$.

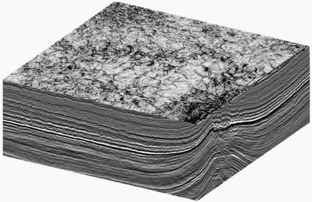
This simple to state problem models a lot of important real-world applications!



- **Sample complexity** usually corresponds to some application-dependent cost (e.g. length of time to acquire MRI, number of experiments needed to image below the earths surface).

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 **Fx**, the cheaper and faster our data acquisition process becomes.

Typically design $\mathbf{A}$ with as few rows as possible that fulfills some desired property.

- $\mathbf{A}$ 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 $k = 1/\mu$.
- $\mathbf{A}$ obeys the (q, ϵ)-Restricted Isometry Property.
  - Recover vectors $\mathbf{x}$ with sparsity $k = O(q)$.

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

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

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

Can argue this property holds for random matrices (JL matrices) and ~~subsampled Fourier matrices~~ with roughly $m = O\left(\frac{q \log n}{\epsilon^2}\right)$ rows. $= O(k \log n)$

$q = O(k)$ $\quad$ $q = 2k, 3k$

$\epsilon = O(1)$

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:

$\Delta = y - x$ $A\Delta = Ay - Ax$

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

$= b - b$

$= 0$

x is k-sparse $Ax = b$. y $Ax = b$ and y was $\leq k$ sparse

- Establishes that information theoretically we can recover $\mathbf{x}$ in $O(n^k)$ time from $O(k \log n)$ measurements.

Proof: Suppose (by way contradiction) $\exists$ y, s.t.

1) $Ay = b$ 2) $\|y\|_0 \leq \|x\|_0$. Then A cannot be $(2k, \epsilon)$-RIP.

If A was RIP,

$y = x + \Delta$ $\|A\Delta\|_2 = 0$ Δ is $2k$ sparse. $\frac{1}{2}\|\Delta\|_2 \leq \|A\Delta\|_2$

for some nonzero Δ. $\leq 1.5\|\Delta\|_2$

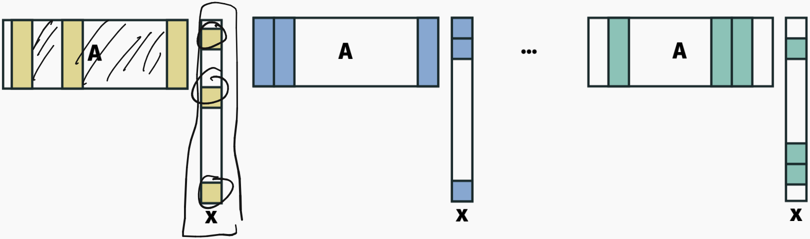
Definition ((q, ϵ)-Restricted Isometry Property – Candes, Tao '05) Donoho

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

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

$O\left(\frac{q \log(n)}{\epsilon^2}\right)$ rows

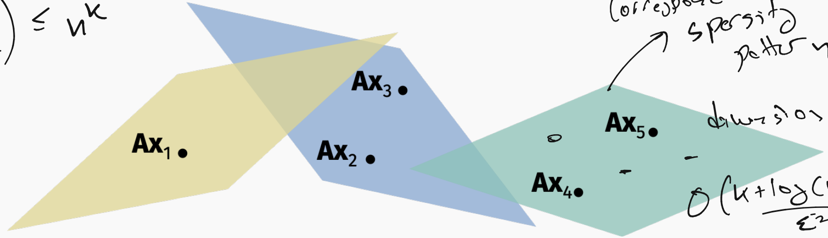
The vectors that can be written as $\mathbf{A}\mathbf{x}$ for q sparse $\mathbf{x}$ lie in a union of q dimensional linear subspaces:



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

$\binom{n}{k} \leq n^k$

correspond to sparsity pattern



dimension k

$O\left(\frac{k+\log(1/\delta)}{\epsilon^2}\right)$

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?

$\delta \approx \frac{\delta'}{N_S}$ → number of subspace

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 $1-\delta$,

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

$$\binom{n}{k} \sim \left(\frac{n}{k}\right)^k$$

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

Quick argument: $\binom{n}{k} \leq n^k$ subspaces

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

$$\delta = \frac{1}{n^c}$$

$$\frac{k+\log(n^k)}{\epsilon^2} = \frac{k+k\log n}{\epsilon^2} = O\left(\frac{k\log n}{\epsilon^2}\right)$$

Theorem (ℓ_0-minimization)

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

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

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

Convex relaxation of the ℓ_0 minimization problem:

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

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

$$\sum_{i=1}^{n} |z_i| = \|z\|_1$$

$$\sum_{i=1}^{n} \mathbb{1}[z_i \neq 0] = \|z\|_0$$

- Objective is convex.
- Optimizing over convex set.

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

Theorem $(2k, \varepsilon)$

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

Two surprising things about this result:

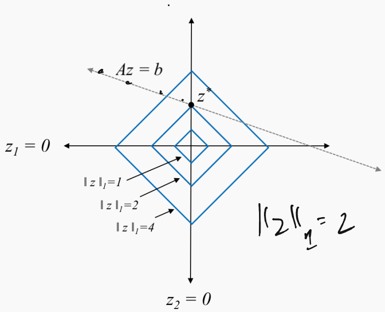
- Exponentially improve computational complexity with only a constant factor overhead in measurement complexity.
- Typical "relax-and-round" algorithm, but rounding is not even necessary! Just return the solution of the relaxed problem.

Why ℓ_1 norm instead of ℓ_2 norm?

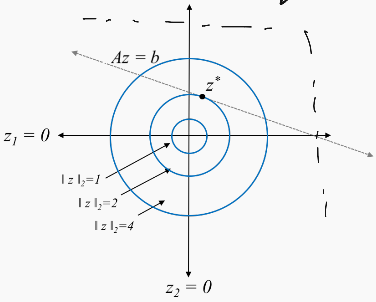
BASIS PURSUIT INTUITION

Suppose **A** is 2×1, so **b** is just a scalar and **x** is a 2-dimensional vector.

$Az = b$ $\|z\|_1$ $\|z\|_2$



$\|z\|_1 = 2$



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

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

$\min \|z\|_2 \quad \text{s.t.} \quad Az = b$

Theorem

If $\mathbf{A}$ *is* $(3k, \epsilon)$*-RIP for* $\epsilon < .17$ *and* $\|\mathbf{x}\|_0 = k$*, then* $\mathbf{x}$ *is the unique optimal solution of the Basis Pursuit LP).* → ℓ_1 minimization

Similar proof to ℓ_0 minimization:

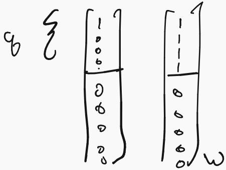
- By way of contradiction, assume $\mathbf{x}$ is not the optimal solution. Then there exists some non-zero Δ such that:
  - $\|\mathbf{x} + \Delta\|_1 \leq \|\mathbf{x}\|_1$
  - $\underbrace{\mathbf{A}(\mathbf{x} + \Delta)}_{= b} = \underbrace{\mathbf{A}\mathbf{x}}_{= b}$. I.e. $\mathbf{A}\Delta = 0$.

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 $\mathbf{w}$, $\|\mathbf{w}\|_2 \leq \|\mathbf{w}\|_1 \leq \sqrt{q}\|\mathbf{w}\|_2$



$\|w\|_2 = q$

$\|w\|_2 = \sqrt{q}$

Second tool:

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

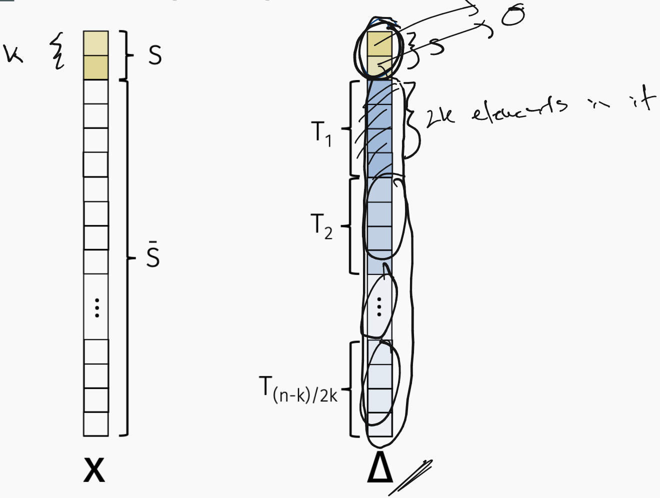
$\|a\| = \|a+b-b\| \leq \|a+b\| + \|-b\|$

$\|a\| \leq \|a+b\| + \|b\|$

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

$x_S = [\;]$

x_i

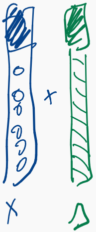


Recall: By way of contradiction, if $\mathbf{x}$ is not the minimizer of the ℓ_1 problem, then there is some Δ such that $\mathbf{A}(\mathbf{x}+\Delta) = \mathbf{b}$ and $\|\mathbf{x}+\Delta\|_1 \leq \|\mathbf{x}\|_1$.

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

$$\cancel{\|x\|_1} \geq \|x+\Delta\|_1 = \underbrace{\|x_S+\Delta_S\|_1} + \|\Delta_{\bar S}\|_1$$

$$\geq \cancel{\|x_S\|_1} - \|\Delta_S\|_1 + \|\Delta_{\bar S}\|_1$$



$$\sum_{i=1}^{n} |(x+\Delta)_i| = \sum_{i\in S} |(x+\Delta)_i| + \sum_{i\notin S} |(x+\Delta)_i|$$

$$= \sum_{i\in S} |(x+\Delta)_i| + \sum_{i\notin S} |\Delta_i|$$

$$\hookrightarrow = \|\Delta_{\bar S}\|_1$$

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

We have:

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

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

$m = \max_{i \in T_{j+1}} \Delta_i$

$\Delta_{T_j} \to \|\Delta_{T_j}\|_1$

$\Delta_{T_{j+1}} \to \|\Delta_{T_{j+1}}\|_2$

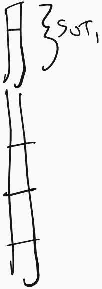
$\|\Delta_{T_j}\|_2 \geq 2k \cdot m$

$\|\Delta_{T_{j+1}}\|_2 \leq \sqrt{2k \cdot m^2} = \sqrt{2k} \cdot m$

$\sum$

6^{7}

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



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

$$\geq (1-\varepsilon)\|\Delta_{S\cup T_1}\|_2 - (1+\varepsilon)\sum_{j\geq 2}\|\Delta_{T_j}\|_2$$

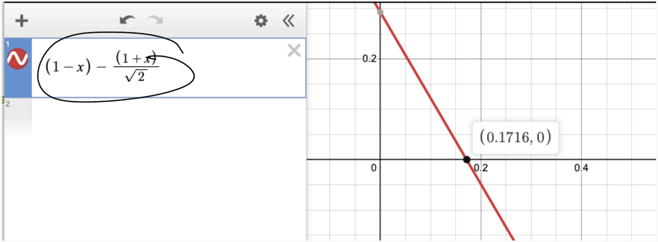
$$\geq (1-\varepsilon)\|\Delta_{S\cup T_1}\|_2 - \frac{(1+\varepsilon)}{\sqrt{2}}\|\Delta_S\|_2$$

$$\geq (1-\varepsilon)\|\Delta_{S\cup T_1}\|_2 - \frac{(1+\varepsilon)}{\sqrt{2}}\|\Delta_{S\cup T_1}\|_2$$

positive

$$= \left((1-\varepsilon) - \frac{(1+\varepsilon)}{\sqrt{2}}\right)\|\Delta_{S\cup T_1}\|_2$$

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



Theorem

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

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 $\min_{\mathbf{z}} \|\mathbf{A}\mathbf{z} - \mathbf{b}\|$ 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.

$O(k \log^2 n)$

When **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+].

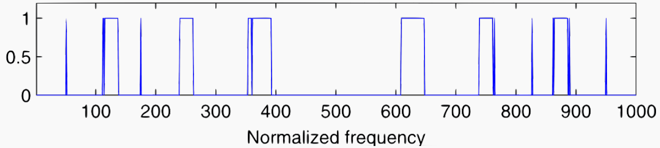
$O(n \log n)$

Wait a minute...

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 $\mathbf{x}$.

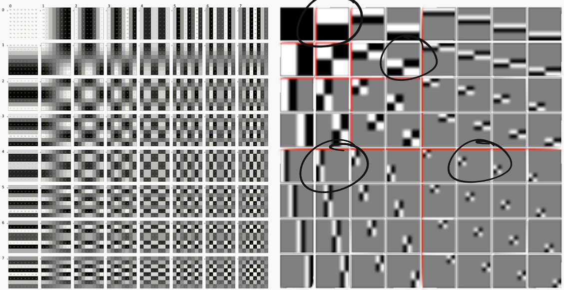
Fourier and inverse Fourier transforms in <u>sublinear time</u> when the output is sparse.



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

COMPRESSED SENSING FOR IMAGES

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



2 , c_1
10 , c_2
15 , c_3
18 , c_4

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

COMPRESSED SENSING RELATED TO MODERN DEEP LEARNING METHOD METHODS

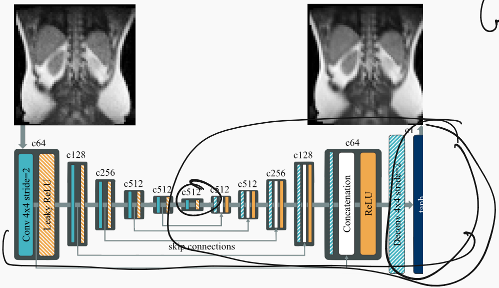
Compressed Sensing using Generative Models

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

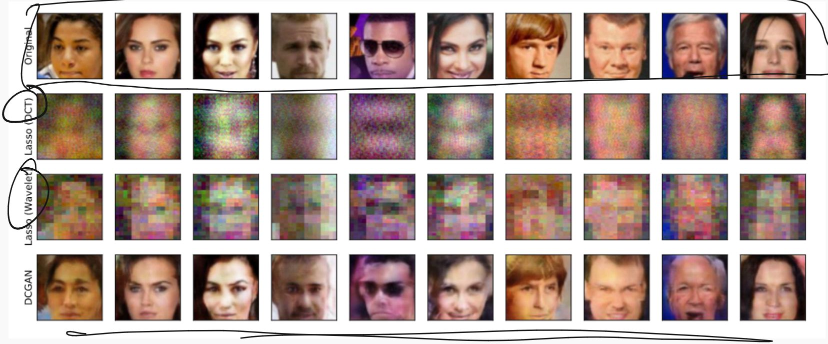
Abstract

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

$G(z)$



Reconstruction using the same number of samples. Last row is method based on a GAN generative model.



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

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

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

A LITTLE ABOUT MY RESEARCH

Theorem (Subspace Embedding)

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

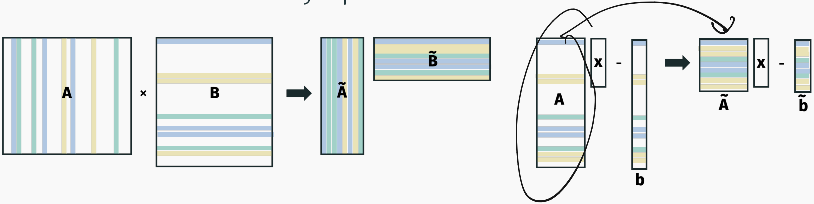
$$(1-\epsilon)\|\mathbf{A}\mathbf{x}\|_2^2 \leq \|\mathbf{\Pi}\mathbf{A}\mathbf{x}\|_2^2 \leq (1+\epsilon)\|\mathbf{A}\mathbf{x}\|_2^2$$

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

Implies regression result, and more.

Example: Any singular value $\tilde{\sigma}_i$ of $\mathbf{\Pi}\mathbf{A}$ is a $(1 \pm \epsilon)$ approximation to the true singular value σ_i of $\mathbf{B}$.

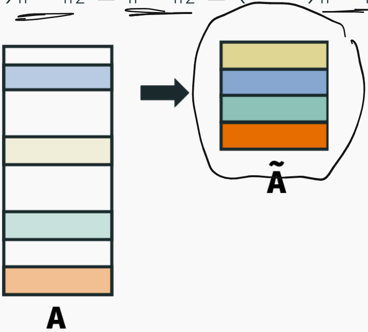
Recurring research interest: Replace random projection methods with random sampling methods. Prove that for essentially all problems of interest, can obtain same asymptotic runtimes.



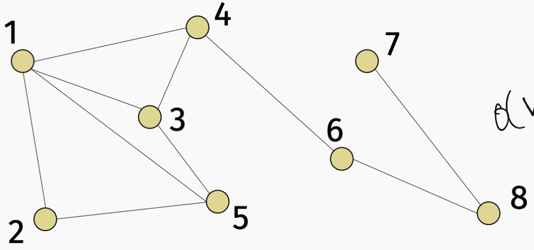
Sampling has the added benefit of preserving matrix sparsity or structure, and can be applied in a wider variety of settings where random projections are too expensive.

Goal: Can we use sampling to obtain subspace embeddings? I.e. for a given $\mathbf{A}$ find $\tilde{\mathbf{A}}$ whose rows are a (weighted) subset of rows in $\mathbf{A}$ and:

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



Let $\mathbf{B}$ be the edge-vertex incidence matrix of a graph G with vertex set V, $|V| = d$. Recall that $\mathbf{B}^T\mathbf{B} = \mathbf{L}$.



n

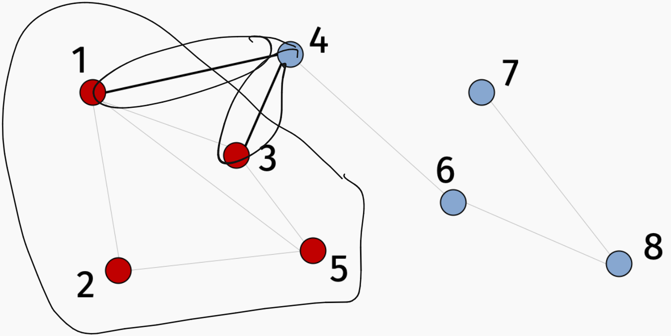
$O(n^2)$

$$\begin{bmatrix} +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ +1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ +1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ +1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & +1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & +1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 \\ & & & \vdots & & & & \\ 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 \end{bmatrix}$$

B

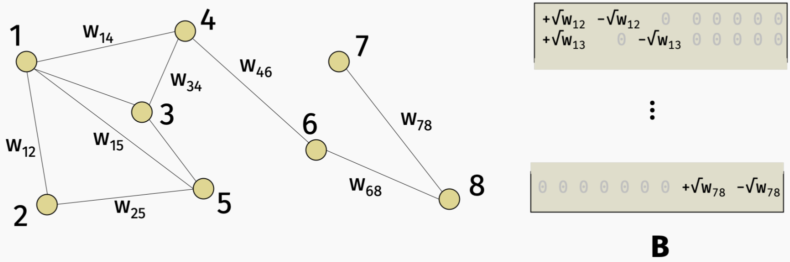
Recall that if $\mathbf{x} \in \{-1, 1\}^n$ is the cut indicator vector for a cut S in the graph, then $\frac{1}{4}\|\mathbf{B}\mathbf{x}\|_2^2 = \text{cut}(S, V \setminus S)$.

$$\mathbf{x} = [1, 1, 1, -1, 1, -1, -1, -1]$$



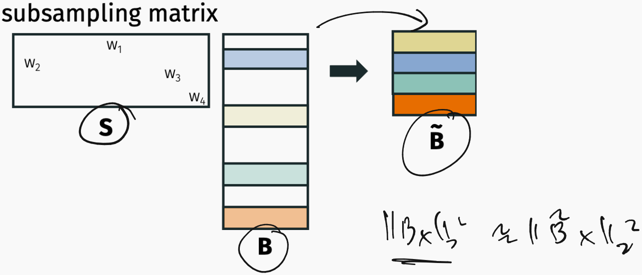
$\mathbf{x} \in \{-1, 1\}^d$ is the cut indicator vector for a cut S in the graph, then $\frac{1}{4}\|\mathbf{B}\mathbf{x}\|_2^2 = \mathbf{cut}(S, V \setminus S)$

Extends to weighted graphs, as long as square root of weights is included in $\mathbf{B}$. Still have the $\mathbf{B}^T\mathbf{B} = \mathbf{L}$.



And still have that if $\mathbf{x} \in \{-1, 1\}^d$ is the cut indicator vector for a cut S in the graph, then $\frac{1}{4}\|\mathbf{B}\mathbf{x}\|_2^2 = \mathbf{cut}(S, V \setminus S)$.

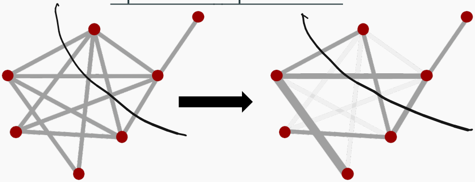
Goal: Approximate $\mathbf{B}$ by a weighted subsample. I.e. by $\tilde{\mathbf{B}}$ with $m \ll |E|$ rows, each of which is a scaled copy of a row from $\mathbf{B}$.



Natural goal: $\tilde{\mathbf{B}}$ is a subspace embedding for $\mathbf{B}$. In other words, $\tilde{\mathbf{B}}$ has $\approx O(d)$ rows and for all $\mathbf{x}$,

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

$\tilde{\mathbf{B}}$ is itself an edge-vertex incidence matrix for some sparser graph $\tilde{G}$! $\tilde{G}$ is called a spectral sparsifier for G.



For example, we have that for any set S,

$$(1-\epsilon)\,\mathrm{cut}_G(S, V \setminus S) \leq \mathrm{cut}_{\tilde{G}}(S, V \setminus S) \leq (1+\epsilon)\,\mathrm{cut}_G(S, V \setminus S).$$

So $\tilde{G}$ can be used in place of G in solving e.g. max/min cut problems, balanced cut problems, etc.

In contrast $\mathbf{\Pi B}$ would look nothing like an edge-vertex incidence matrix if $\mathbf{\Pi}$ is a JL matrix.

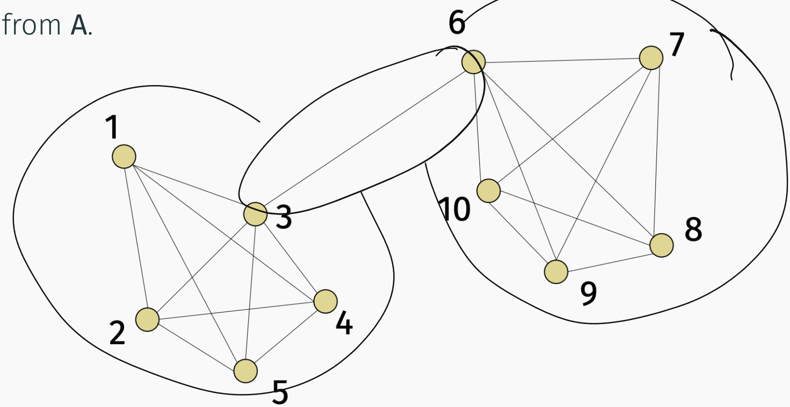
Spectral sparsifiers were introduced in 2004 by Spielman and Teng in an influential paper on faster algorithms for solving Laplacian linear systems.

- Generalize the cut sparsifiers of Benczur, Karger '96.
- Further developed in work by Spielman, Srivastava + Batson, '08.
- Have had huge influence in algorithms, and other areas of mathematics – this line of work lead to the 2013 resolution of the Kadison-Singer problem in functional analysis by Marcus, Spielman, Srivastava.

Rest of class: Learn about an important random sampling algorithm for constructing spectral sparsifiers, and subspace embeddings for matrices more generally.

Goal: Find $\tilde{\mathbf{A}}$ such that $\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\|_2^2$ for all $\mathbf{x}$.

Possible Approach: Construct $\tilde{\mathbf{A}}$ by uniformly sampling rows from $\mathbf{A}$.



Can check that this approach fails even for the special case of a graph vertex-edge incidence matrix.

Key idea: Importance sampling. Select some rows with higher probability.

Suppose $\mathbf{A}$ has n rows $\mathbf{a}_1 \ldots, \mathbf{a}_n$. Let $p_1, \ldots, p_n \in [0,1]$ be sampling probabilities. Construct $\tilde{\mathbf{A}}$ as follows:

- For $i = 1, \ldots, n$
  - Select $\mathbf{a}_i$ with probability p_i.
  - If $\mathbf{a}_i$ is selected, add the scaled row $\frac{1}{\sqrt{p_i}}\mathbf{a}_i$ to $\tilde{\mathbf{A}}$.

Remember, ultimately want that $\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 = (1 \pm \epsilon)\|\mathbf{A}\mathbf{x}\|_2^2$ for all $\mathbf{x}$.

Claim 1: $\mathbb{E}[\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2] = \|\mathbf{A}\mathbf{x}\|_2^2$.

Claim 2: Expected number of rows in $\tilde{\mathbf{A}}$ is $\sum_{i=1}^n p_i$.

How should we choose the probabilities $p_1, \ldots, p_n$?

τ_i = ith "statistical leverage score" of A

For $i = 1, \ldots, n$,

$$\tau_i = \mathbf{a}_i^T(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{a}_i.$$

Theorem (Subspace Embedding from Subsampling)

For each i, and fixed constant c, let $p_i = \min\left(1, \frac{c\log d}{\epsilon^2}\cdot\tau_i\right)$. *Let* $\tilde{\mathbf{A}}$ *have rows sampled from* $\mathbf{A}$ *with probabilities* $p_1, \ldots, p_n$. *With probability* $9/10$,

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

and $\tilde{\mathbf{A}}$ *has* $O(d\log d/\epsilon^2)$ *rows in expectation.*

VECTOR SAMPLING

$\|Ax\|$

How should we choose the probabilities $p_1, \ldots, p_n$?

As usual, consider a single vector $\mathbf{x}$ and understand how to sample to preserve norm of $\mathbf{y} = \mathbf{A}\mathbf{x}$:

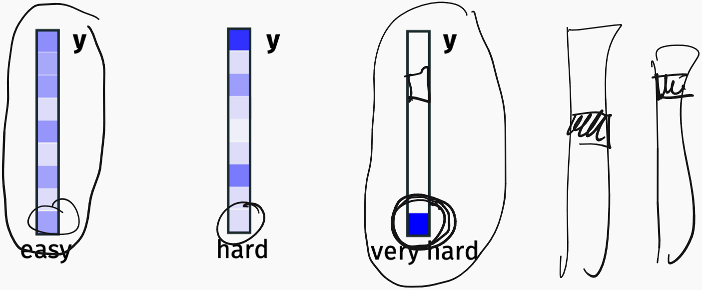
$$\|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 = \|\mathbf{S}\mathbf{A}\mathbf{x}\|_2^2 = \|\mathbf{S}\mathbf{y}\|_2^2 \approx \|\mathbf{y}\|_2^2 = \|\mathbf{A}\mathbf{x}\|_2^2.$$

Then we can union bound over an ϵ-net to extend to all $\mathbf{x}$.

VECTOR SAMPLING

$y := Ax$

As discussed a few lectures ago, uniform sampling only works well if $\mathbf{y} = \mathbf{A}\mathbf{x}$ is "flat".



Instead consider sampling with probabilities at least proportional to the magnitude of $\mathbf{y}$'s entries:

$$p_i > c \cdot \frac{y_i^2}{\|y\|_2^2} \text{ for constant } c \text{ to be determined.}$$

Using a Bernstein bound (or Chebyshev's inequality if you don't care about the δ dependence) we have that if $c = \frac{\log(1/\delta)}{\epsilon^2}$ then:

$$\Pr[\left|\|\tilde{\mathbf{y}}\|_2^2 - \|\mathbf{y}\|_2^2\right| \geq \epsilon\|\mathbf{y}\|_2^2] \leq \delta.$$

The number of samples we take in expectation is:

$$\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} c \cdot \frac{y_i^2}{\|y_i\|_2^2} = \frac{\log(1/\delta)}{\epsilon^2}.$$

MAJOR CAVEAT!

We don't know $y_1, \ldots, y_n$! And in fact, these values aren't fixed. We wanted to prove a bound for $\mathbf{y} = \mathbf{A}\mathbf{x}$ for <u>any</u> $\mathbf{x}$.

Idea behind leverage scores: Sample row i from $\mathbf{A}$ using the <u>worst case</u> (largest necessary) sampling probability:

$$\tau_i = \max_{\mathbf{x}} \frac{y_i^2}{\|\mathbf{y}\|_2^2} \qquad \text{where} \qquad \underline{\mathbf{y} = \mathbf{A}\mathbf{x}}.$$

If we sample with probability $p_i = \frac{1}{\epsilon^2} \cdot \tau_i$, then we will be sampling by at least $\frac{1}{\epsilon^2} \cdot \frac{y_i^2}{\|\mathbf{y}\|_2^2}$, <u>no matter what $\mathbf{y}$ is</u>.

CLOSED FORM EXPRESSION FOR LEVERAGE SCORES

$y^* = Ax^*$

$y_i^* = a_i^T (A^T A^{-1}) a_i$

$= a_i^T x$

$$\tau_i = \max_{\mathbf{x}} \frac{y_i^2}{\|\mathbf{y}\|_2^2} \qquad \text{where} \qquad \mathbf{y} = \mathbf{A}\mathbf{x}.$$

A little messy algebra shows that $\mathbf{x}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{a}_i$.

$$\tau_i = \frac{(a_i^T (A^T A)^{-1} a_i)^2}{\|A (A^T A)^{-1} a_i\|_2^2} = \frac{\cdots}{a_i^T (A^T A)^{-1} \cancel{A^T} A \cancel{(A^T A)^{-1}} a_i}$$

$$= \frac{(a_i^T (A^T A)^{-1} a_i)^2}{(a_i^T (A^T A)^{-1} a_i)}$$

$\|z\|_2^2 \;\; z^T z$

$$\tau_i = a_i (A^T A)^{-1} a_i$$

Two concerns:

1) How to efficiently compute $\tau_1, \ldots, \tau_n$?

2) The number of samples we take will be roughly $\underbrace{\sum_{i=1}^{n} \tau_i}_{\leq d}$. How do we bound this?

Topic for another day!

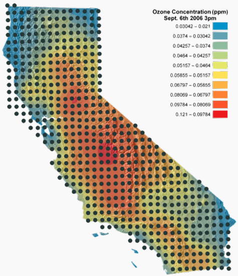
$$\sum a_i^T (A^T A)^{-1} a_i = tr\left(A (A^T A)^{-1} A^T\right)$$

$$= tr\left(I_{d \times d}\right) = d$$

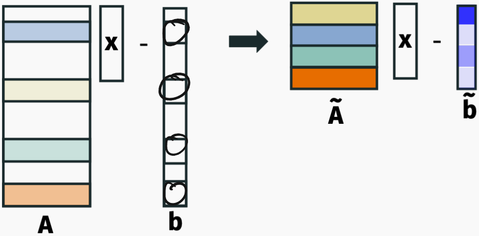
In many applications, computational costs are second order to <u>data collection costs.</u> We have a huge range of possible data points $\mathbf{a}_1, \ldots, \mathbf{a}_n$ that we can collect labels/values $b_1, \ldots, b_n$ for. Goal is to learn $\mathbf{x}$ such that:

$$\mathbf{a}_i^T \mathbf{x} \approx b_i.$$

Want to do so after observing as few $b_1, \ldots, b_n$ as possible. Applications include healthcare, environmental science, etc.



Can be solved via random sampling for linear models.



Claim: Let $\tilde{\mathbf{A}}$ is an $O(1)$-factor subspace embedding for $\mathbf{A}$ (obtained via leverage score sampling). Then $\tilde{\mathbf{x}} = \arg\min \|\tilde{\mathbf{A}}\mathbf{x} - \tilde{\mathbf{b}}\|_2^2$ satisfies:

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

Computing $\tilde{\mathbf{x}}$ only requires collecting $O(d \log d)$ labels.

Lots of applications:

$\sum_{i=1}^{k} \sigma_i^2$

- Robust bandlimited and multiband interpolation [STOC 2019].
- Active learning for Gaussian process regression [NeurIPS 2020].
- Active learning beyond the ℓ_2 norm [FOCS 2022]
- Active learning for polynomial regression [SODA 2023]
- Active learning for 1 layer neural nets [NeurIPS 2023]
- DOE Grant on "learning based" algorithms for solving parametric partial differential equations.

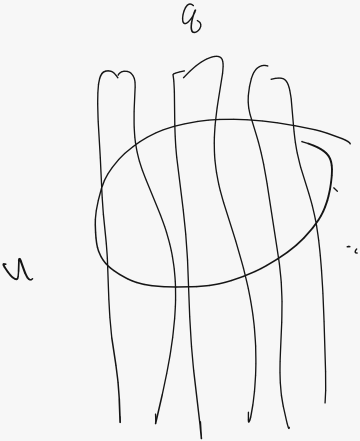
U^T

$\|X\|_F^2 = \sum_{i=1}^{d} \sigma_i^2$

$X = U \Sigma V^T$

$\|X\|_F^2 = \|XV\|_F^2$

$\|X\|_F^2 = \|U^T X V\|_F^2 = \|\Sigma\|_F^2$



$v = Q w$

$v = \cancel{P} (A^T A) g$