CS-GY 6763: Lecture 14
Finish Sparse Recovery and Compressed Sensing, Introduction to Leverage Score Sampling

NYU Tandon School of Engineering, Prof. Christopher Musco

This is our last class!

- Final project due next Tuesday.
- Exam study guide was released. Same rules as midterm (cheat sheet allowed). will be a 1.5 hour test.
- Solutions for last problem sets will be released tonight.

COURSE FEEDBACK

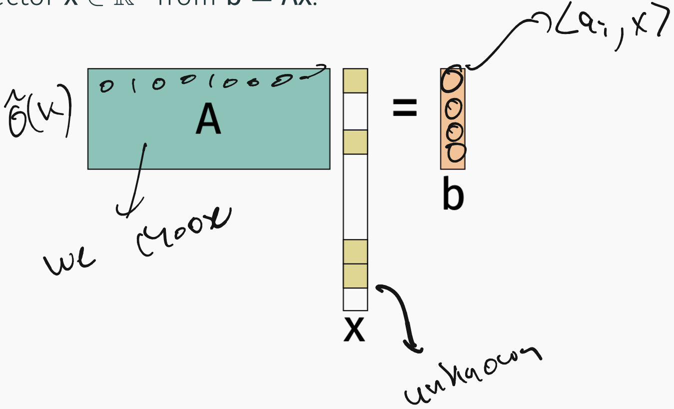
This course is taught every year and is now one of the primary ways of filling the theory breadth requirement for Ph.D. students, so it is important that we keep improving it.

Graduate Section

COURSE FEEDBACK

Undergraduate Section

Design $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m < n$ rows so that we can recover k sparse vector $\mathbf{x} \in \mathbb{R}^n$ from $\mathbf{b} = \mathbf{A}\mathbf{x}$.



RESTRICTED ISOMETRY PROPERTY

$\frac{\log(1/\delta)}{\epsilon^2}$ for a single

sparsity

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

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

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

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

Can argue this property holds for random JL matrice with $m = O\left(\frac{q\log(n/q)}{\epsilon^2}\right)$ rows.

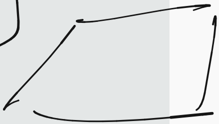
$q = O(k)$

$\sim \tilde{O}(q \log(n))$

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 $1-\delta$,

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

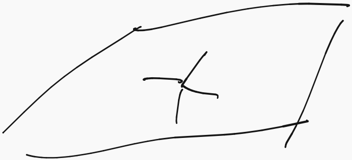


for all $\mathbf{v} \in \mathcal{U}$, as long as $m = O\left(\frac{q+\log(1/\delta)}{\epsilon^2}\right)$.

We will use union bound to apply this theorem to a collection of linear subspaces.

2^8

$2^8 \binom{n}{8} << 2^4$

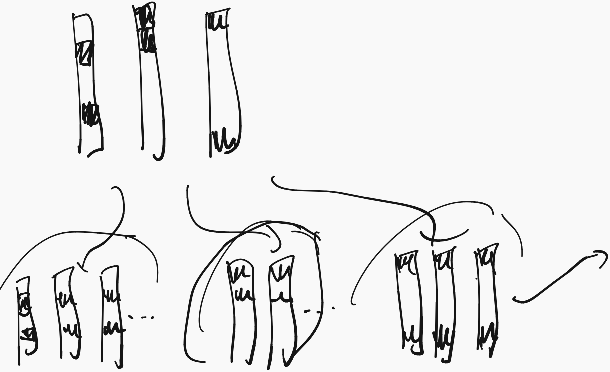


Let $\mathcal{S}_k = \{\mathbf{x} : \|\mathbf{x}\|_0 \leq q\}$ be the collection of all q sparse vectors.

$$\mathcal{S} = \mathcal{U}_1 \cup \ldots \cup \mathcal{U}_T,$$

where $T = \binom{n}{q}$.

want to show:
for all $x \in S_k$
$\|Ax\|_2^2 \approx (1 \pm \varepsilon)\|x\|_2^2$



$c_1 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + c_2 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

Theorem (Subspace Embedding from JL) $\left(\frac{n}{q}\right)^q \le \binom{n}{q} \le \left(\frac{en}{q}\right)^q$

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 $1-\delta$, $T \le \left(\frac{en}{q}\right)^q$

$$(1-\epsilon)\|\mathbf{v}\|_2^2 \le \|\mathbf{\Pi}\mathbf{v}\|_2^2 \le (1+\epsilon)\|\mathbf{v}\|_2^2$$ $\log(T) \le q\log\left(\frac{en}{q}\right)$

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

As long as we take a JL matrix with $O(\frac{q+\log(T)}{\epsilon^2})$ rows then it will preserve the norm of all vectors in $\mathcal{S} = \mathcal{U}_1 \cup \ldots \cup \mathcal{U}_T$ with high probability.

$$\log(T) = \log\binom{n}{q} = \frac{n\cdot(n-1)\cdots(n-q)}{q(q-1)\cdots 1} \approx \left(\frac{n}{q}\right)^q$$

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:*

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

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

Suppose $\exists\, y : \|y\|_0 \leq \|x\|_0 = k$

and $Ay = b$

$\|A(x-y)\|_2 \geq (1-\epsilon)\|x-y\|_2$

$= 0$

Convex relaxation of the ℓ_0 minimization problem:

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

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

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

- Objective is convex.
- Optimizing over convex set. $O(n^{3.5})$

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

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.*

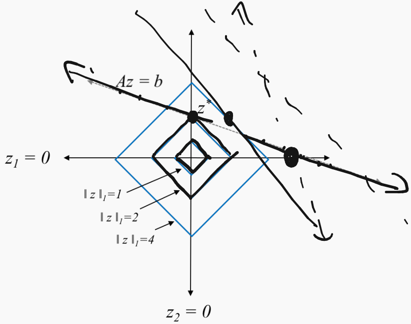
Two surprising things about this result:

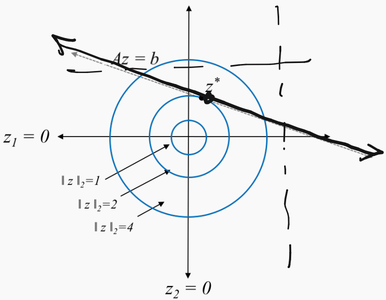
- Exponentially improve computational complexity with only a constant factor overhead in measurement complexity.
- Typical "relax-and-round" algorithm, but rounding is not even necessary! Just return the solution of the relaxed problem.

Why ℓ_1 norm instead of ℓ_2 norm?

$A_1 x_1 + A_2 x_2 = b$

Suppose **A** is ~~2 × 1~~ 1 × 2, 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.

$(\min_z \|z\|_2)$

$$\min_{\mathbf{z}} \|\mathbf{z}\|_1 \quad \text{subject to} \quad \mathbf{A}\mathbf{z} = \mathbf{b}.$$

Theorem

$y = x + \Delta \quad \Delta = y - x$

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).*

suppose $\exists\, y$:

$\|y\|_2 \leq \|x\|_2$ and $Ay = b$
$= Ax$

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. $\mathbf{A}\Delta = 0$.

$Ax = Ax + A\Delta \quad A\Delta = 0.$

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

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \rightarrow \|w\|_2 = \sqrt{\sum_{i=1}^{q} w_i} = \sqrt{q} = \sqrt{q}$$

$$\|w\|_1 = \sum_{i=1}^{q} |w_i| = q$$

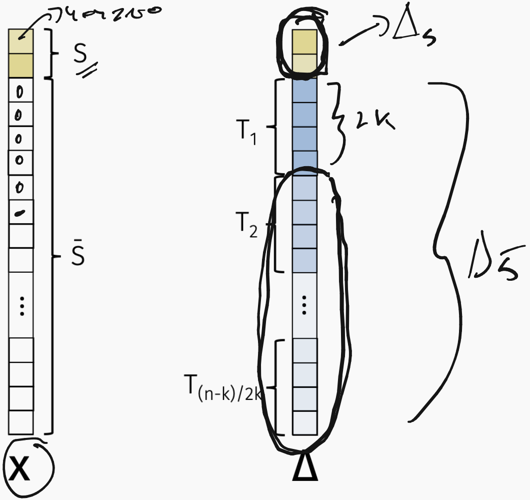
Second tool:

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

Triangle inequality. $\|a+b\| \geq \|b\| - \|a\|$

$$\underbrace{\|a - (a+b)\|}_{= \|b\|} \leq \|a\| + \|a+b\|$$

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.



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

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

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

$$\|\Delta_S\|_1 \geq \|\Delta_{\bar{S}}\|_1$$

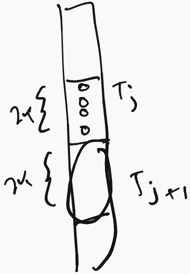
Claim 2 (ℓ_2 approximate sparsity): $\|\Delta_S\|_2 \geq \sqrt{2}\sum_{j\geq 2}\|\Delta_{T_j}\|_2$.

We have:

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

$\bar{S} = T_1 \cup T_2 \ldots \cup T_z$

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



$\|\Delta_{T_j}\|_1 \geq 2k \cdot \max(\Delta_{T_{j+1}})$

$\|T_{j+1}\|_2 \leq \sqrt{2k} \cdot \max(\Delta_{T_{j+1}})$

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

$A\Delta = 0$

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

$$A\Delta = A\Delta_{S \cup T_1} + A\Delta_{T_2} + \dots A\Delta_{T_2}$$

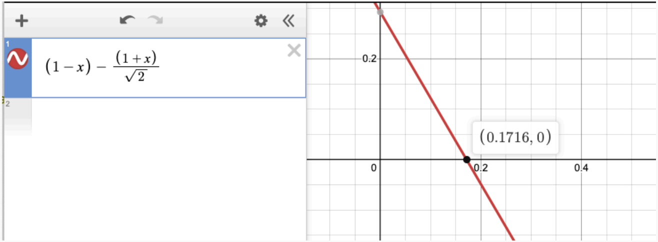
by RIP

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

$$\geq (1-\epsilon)\|\Delta_S\|_2 - \frac{(1+\epsilon)}{\sqrt{2}}\|\Delta_S\|_2$$

$$= \left((1-\epsilon) - \frac{(1+\epsilon)}{\sqrt{2}}\right)\|\Delta_S\|_2$$

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, which can be solved in polynomial time.*

$O(k \log(n/k))$

FASTER METHODS

$\min_z \|Az - b\|$ s.t. $\|z\|_0 \leq k$

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

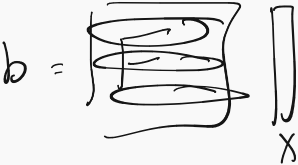
- **Iterative Hard Thresholding**: Looks a lot like projected gradient descent. Solve $\min_{\mathbf{z}} \|\mathbf{Az} - \mathbf{b}\|$ with gradient descent while continually projecting z back to the set of k-sparse vectors. Runs in time $\sim O(nk \log n)$ for Gaussian measurement matrices and $O(n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

FASTER METHODS

sublinear in 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+].

Wait a minute...



SPARSE FOURIER TRANSFORM

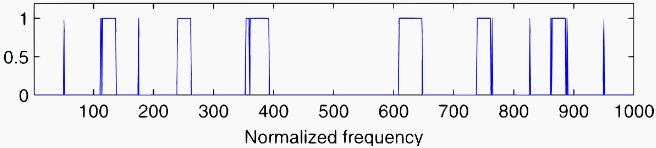
Corollary: When $\mathbf{x}$ is k-sparse, we can compute the inverse Fourier transform $\mathbf{F}^*\mathbf{F}\mathbf{x}$ of $\mathbf{F}\mathbf{x}$ in $O(k \log^c n)$ time!

F^*

$y = F x$

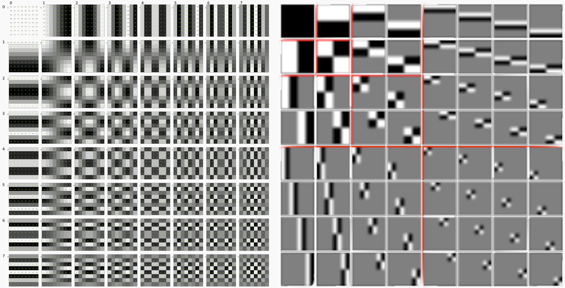
- Randomly subsample $\mathbf{F}\mathbf{x}$.
- Feed that input into our sparse recovery algorithm to extract $\mathbf{x}$.

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



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

ω

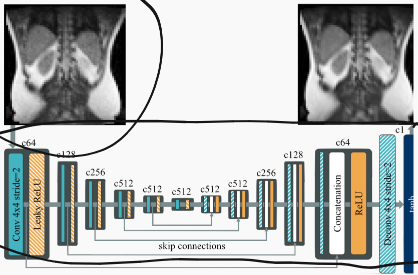
x

Compressed Sensing using Generative Models

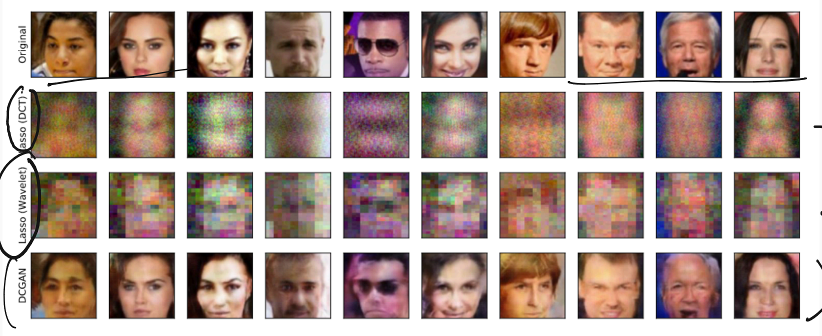
Ashish Bora* Ajil Jalal† Eric Price‡ Alexandros G. Dimakis§

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.



For most generative models (e.g., GANs) output is parameterized by a short seed vector $\mathbf{z}$.



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)

Ax

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 $1 - \delta$,

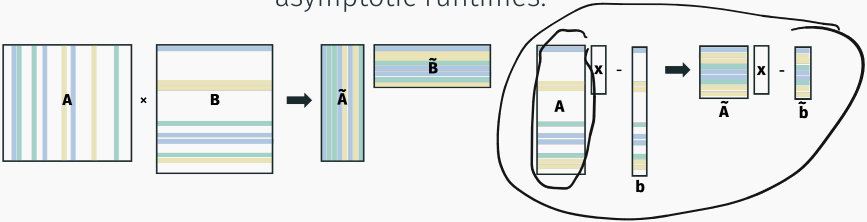
$$(1-\epsilon)\|\mathbf{A}\mathbf{x}\|_2^2 \leq \|\mathbf{\Pi}\mathbf{A}\mathbf{x}\|_2^2 \leq (1+\epsilon)\|\mathbf{A}\mathbf{x}\|_2^2$$

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 $\mathbf{B}$.

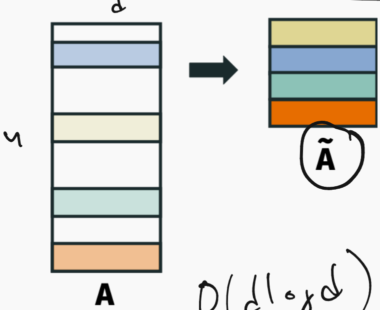
Recurring research interest: Replace random projection methods with random sampling methods. Prove that for essentially all problems of interest, can obtain same asymptotic runtimes.



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:

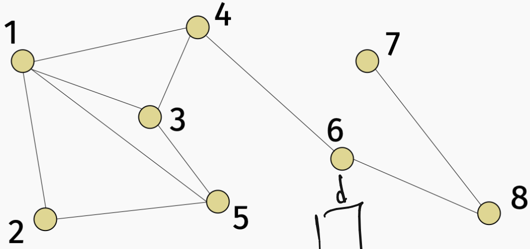
$$(1-\epsilon)\|\mathbf{A}\mathbf{x}\|_2^2 \leq \|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 \leq (1+\epsilon)\|\mathbf{A}\mathbf{x}\|_2^2.$$



$O(d \log d)$

EXAMPLE WHERE STRUCTURE MATTERS

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



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

$\mathbf{B}$

d

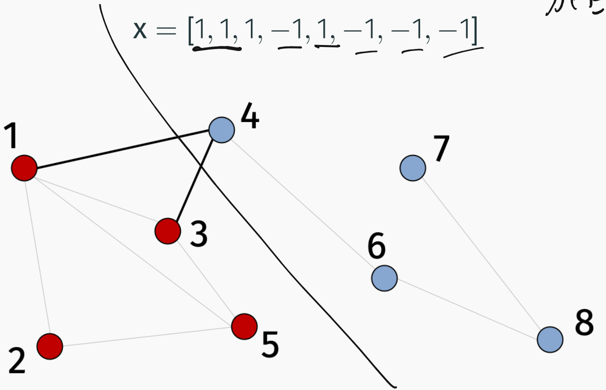
d^2

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

LINEAR ALGEBRAIC VIEW OF CUTS

$\|Bx\|_2^2 = \sum_{i,j \in E} (x(i) - x(j))^2$

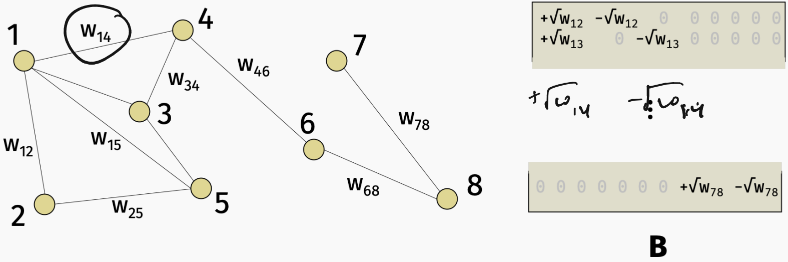
$x = [1, 1, 1, -1, 1, -1, -1, -1]$



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

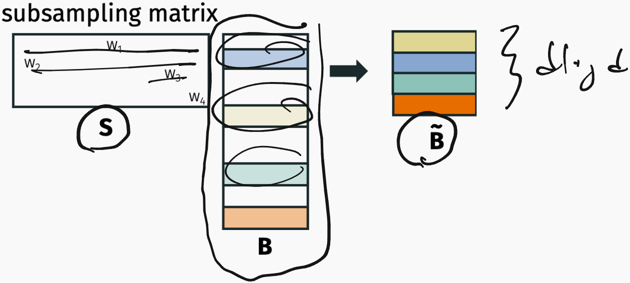
$\|\tilde{B}x\|_2^2$

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 <u>cut indicator vector</u> for a cut S in the graph, then $\frac{1}{4}\|\mathbf{Bx}\|_2^2 = \mathbf{cut}(S, V \setminus S)$.

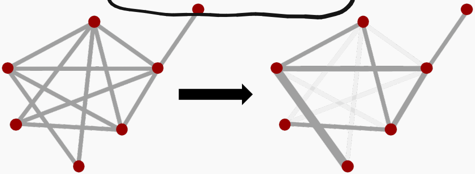
Goal: Approximate $\mathbf{B}$ by a weighted subsample. I.e. by $\tilde{\mathbf{B}}$ with $m \ll |E|$ rows, each of which is a scaled copy of a row from $\mathbf{B}$.



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 $\mathbf{x}$,

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

$\tilde{\mathbf{B}}$ 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,

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

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

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

HISTORY OF SPECTRAL SPARSIFICATION

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.

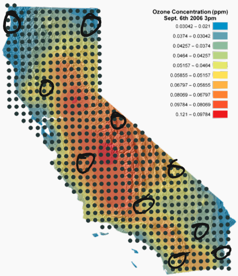
Rest of class: Learn about an important random sampling algorithm for constructing spectral sparsifiers, and subspace embeddings for matrices more generally.

ANOTHER APPLICATION: ACTIVE REGRESSION

In many applications, computational costs are second order to data collection costs. 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:

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

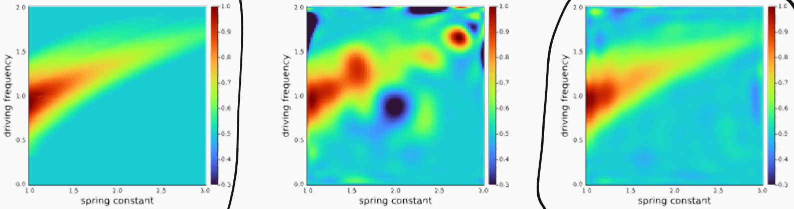
Want to do so after observing as few $b_1, \ldots, b_n$ as possible. Applications include healthcare, environmental science, etc.



- Tons of applications in computational science (e.g. we have a DOE award on learning based methods for parametric PDEs).

$F(\theta)$

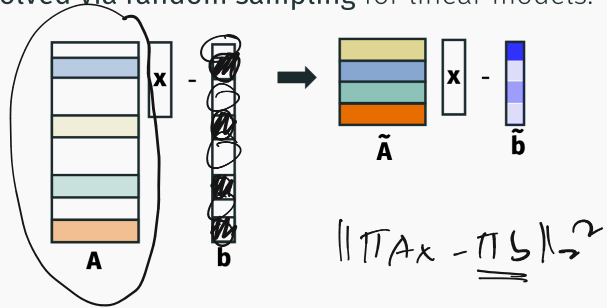
- How you collect samples really matters!



(a) Target Function (b) Bernoulli Sampling (c) Pivotal Sampling (our method)

ANOTHER APPLICATION: ACTIVE REGRESSION

Can be solved via random sampling for linear models.



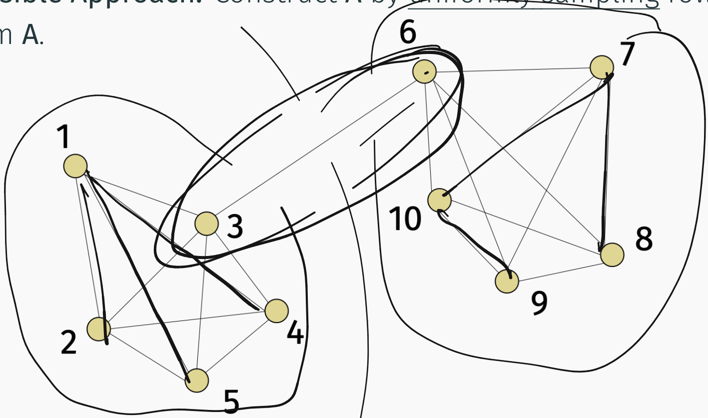
Claim: Let $\tilde{\mathbf{A}}$ is an $O(1)$-factor subspace embedding for $\mathbf{A}$ (obtained via leverage score sampling). Then $\tilde{\mathbf{x}} = \arg\min \|\tilde{\mathbf{A}}\mathbf{x} - \tilde{\mathbf{b}}\|_2^2$ satisfies:

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

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

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

$\sum_{i=1}^{n} \langle \frac{1}{\sqrt{p_i}} a_i, x \rangle^2 \cdot \mathbb{1}_i \quad = \mathbb{E}\left[\sum_{i=1}^{n} \langle a_i, x \rangle^2 \left(\frac{1}{p_i} \cdot \mathbb{1}_i\right)\right] = \|Ax\|_2^2$

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

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

For $i = 1, \ldots, n$, define the statistical leverage score as:

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

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.*

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

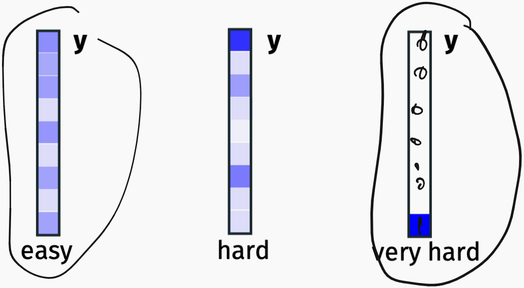
As usual, consider a single vector $\mathbf{x}$ and understand how to sample to preserve norm of $\mathbf{y} = \mathbf{A}\mathbf{x}$:

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

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

VECTOR SAMPLING

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



$\|S\mathbf{y}\|_2^2 \approx (1 \pm \epsilon)\|\mathbf{y}\|$

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

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

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

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:

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

$\|Sy\|_2^2$

The number of samples we take in expectation is:

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

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 any $\mathbf{x}$.

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

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

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

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

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

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

Leverage score sampling:

- For $i = 1, \ldots, n$,
  - Compute $\tau_i = \mathbf{a}_i^T(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{a}_i$.
  - Set $p_i = \frac{c\log(1/\delta)}{\epsilon^2} \cdot \tau_i$.
  - Add row $\mathbf{a}_i$ to $\tilde{\mathbf{A}}$ with probability p_i and reweight by $\frac{1}{\sqrt{p_i}}$.

For any fixed $\mathbf{x}$, we will have that
$(1-\epsilon)\|\mathbf{A}\mathbf{x}\|_2^2 \leq \|\tilde{\mathbf{A}}\mathbf{x}\|_2^2 \leq (1+\epsilon)\|\mathbf{A}\mathbf{x}\|_2^2$ with probability $(1-\delta)$.

Two remaining concerns:

1) How do we extend from <u>any</u> $\mathbf{x}$ to <u>all</u> $\mathbf{x}$?

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

d

SUM OF LEVERAGE SCORES

Claim: No matter how large n is, $\sum_{i=1}^{n} \tau_i = d$ for a matrix $\mathbf{A} \in \mathbb{R}^d$.

$$\sum_{i=1}^{n} a_i^T (A^T A)^{-1} a_i$$

$$= \sum_{i=1}^{n} \mathrm{tr}\left(a_i^T (A^T A)^{-1} a_i \, a_i^T\right)$$

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

"Zero-sum" law for the importance of matrix rows.

Naive ϵ-net argument leads to d^2 dependence since we need to set $\delta = c^d$. Getting the right $d \log d$ dependence below requires a standard "matrix Chernoff bound" (see e.g. Tropp 2015).

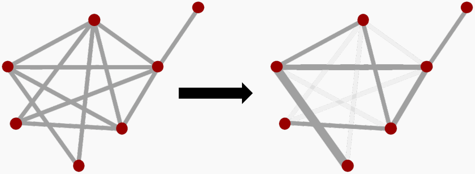
Theorem (Subspace Embedding from Subsampling)

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.*

For any graph G with d nodes, there exists a graph $\tilde{G}$ with $O(d \log d/\epsilon^2)$ edges such that, for all $\mathbf{x}$, $\|\tilde{\mathbf{B}}\mathbf{x}\|_2^2 = (1 \pm \epsilon)\|\mathbf{B}\mathbf{x}\|_2^2$.



As a result, the value of any cut in $\tilde{G}$ is within a $(1 \pm \epsilon)$ factor of the value in G, the Laplacian eigenvalues are with a $(1 \pm \epsilon)$ factors, etc.

THANK YOU!

Thank you all for a great course! If you are interested in learning even more, there are several seminars at NYU that you might be interested in attending:

Theoretical Computer Science Seminar: https://csefoundations.engineering.nyu.edu/seminar.html.

Math and Data Seminar: https://mad.cds.nyu.edu/seminar/.

Computational Math and Scientific Computing Seminar: https://cims.nyu.edu/dynamic/calendars/seminars/computational-mathematics-and-scientific-computing-seminar/.