

EE4.66 Topics in Large Dimensional Data Processing

Wei Dai

Imperial College London (ICL)

2018

Basic Information

Instructor: Dr. Wei Dai

GTA: Mohamed Suliman and Yifan Ran

Prerequisites: Basic knowledge of algorithm design, linear algebra, and probability.

Textbook: No textbook is required. You can rely on lecture notes.

Lectures: 9:00-11:00 Thursdays (10/11/2018-12/13/2018) Room 509B

Assessment: Exam (75%) and coursework (25%).

Section 1

Introduction

A Paradigm Shift

Data analysis: learn an unknown function f from training data \mathbf{x} such that

$$y_i \approx f(\mathbf{x}_i),$$

- ▶ $\mathbf{x}_i \in \mathbb{R}^n, \quad i = 1, 2, \dots, m.$

Classical data processing

- ▶ A small number n of parameters.
- ▶ A large number m of observations.

In many modern applications

- ▶ A large number n of parameters.
- ▶ A relatively small sample size m ($m \approx n$ or $m < n$).

Image Classification Example

- ▶ THE MNIST database of handwritten digits.
 - ▶ $28 \times 28 = 784$ pixels.
 - ▶ 10 categories, 60,000 training samples, 10,000 test samples.
- ▶ Caltech 256: Pictures of objects belonging to 256 categories.
 - ▶ Pictures of various sizes: normally $100 \times 100 = 10,000$ pixels.
 - ▶ 256 categories, 30,607 images in total.
- ▶ Modern database
 - ▶ Including Imagenet, Labelme, etc.
 - ▶ Pictures of various sizes, including HD ones.
 - ▶ Less training images per category in general.

Another Example: The Game of Go

Cited from <http://www.theatlantic.com>:

'The rules of Go are simple and take only a few minutes to learn, but the possibilities are seemingly endless. The number of potential legal board positions is:

208,168,199,381,979,984,699,478,633,344,862,770,286,522,
453,884,530,548,425,639,456,820,927,419,612,738,015,378,
525,648,451,698,519,643,907,259,916,015,628,128,546,089,
888,314,427, 129,715,319,317,557,736,620,397,247,064,840,
935.

That number—which is greater than the number of atoms in the universe—was only determined in early 2016.'

Example Applications

- ▶ Biotech data
 - ▶ DNA microarray: tens of thousands of genes.
 - ▶ Proteomics: thousands of proteins.
 - ▶ Relatively small number of “individuals” (at most in hundreds).
- ▶ Images and videos
 - ▶ Millions of pixels.
 - ▶ Number of patients in cohort study (medical imaging).
- ▶ Business data
 - ▶ Huge amount of internal and external data.
- ▶ Recommendation Engine (Netflix problem)
 - ▶ Large number of users and movies.
 - ▶ Relatively small number of ratings.

Curse of Dimensionality

Curse of dimensionality:

- ▶ The computational difficulty
- ▶ The intrinsic statistical difficulty
 - ▶ Data points are isolated.
 - ▶ False structures.
 - ▶ Overfitting (the inferred model describes the noise instead of the underlying relationship).

To address it: low dimensional structure.

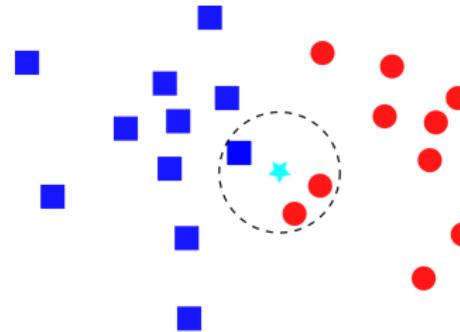
Example 1: K-Nearest Neighbours Algorithm

Training data: blue squares and red dots.

For a given test sample (cyan star), the K -NN algorithm can be used

- ▶ For classification: majority vote using K -nearest neighbors.
- ▶ For regression: average value of the K -nearest neighbors.

The performance is decided by how dense the training points are.

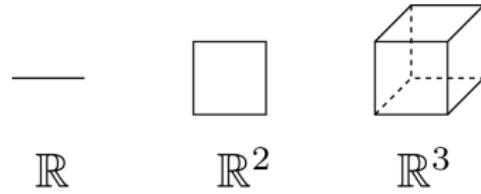


KNN with High Dimensional Data

Question: In the n -dimensional space, how many training samples are needed so that

for any given test sample, there must exist one training sample less than distance 1 away?

Mathematically, how many unit balls are needed to cover the whole space the hypertube $[0, 1]^n$?



k -NN: Isolated Data Points

Cover the hypertube $[0, 1]^n$ by unit balls:

- The volume of $V_n(r)$ of a n -dimensional ball of radius $r > 0$:

$$V_n(r) = \frac{\pi^{n/2}}{\Gamma(n/2 + 1)} r^n \underset{n \rightarrow \infty}{\sim} \left(\frac{2\pi e r^2}{n} \right)^{n/2} (n\pi)^{-1/2}.$$

- To cover the hypertube $[0, 1]^n$ by unit balls, one must have

$$[0, 1]^n \subset \bigcup_{i=1}^m B_n(\mathbf{x}^{(i)}, 1).$$

- That is, $1 \leq m V_n(1)$, or

$$m \geq \frac{\Gamma(n/2 + 1)}{\pi^{n/2}} \underset{n \rightarrow \infty}{\sim} \left(\frac{n}{2\pi e} \right)^{n/2} \sqrt{n\pi}.$$

k -NN: Isolated Data Points

Required number of data points for covering:

n	20	30	50	100	150
m	39	45630	5.7×10^{12}	42×10^{39}	1.28×10^{72}

Example 2: (False Structures) Empirical Covariance

The problem: given samples $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)} \in \mathbb{R}^n$, we want to estimate the covariance matrix

$$\Sigma_x := \mathbb{E} [\mathbf{X}\mathbf{X}^T].$$

Solution: the empirical covariance matrix

$$\begin{aligned}\hat{\Sigma} &:= \frac{1}{m} \sum_{i=1}^m \mathbf{x}^{(i)} \left(\mathbf{x}^{(i)} \right)^T \\ &= \frac{1}{m} \tilde{\mathbf{X}} \tilde{\mathbf{X}}^T,\end{aligned}\tag{1}$$

where $\tilde{\mathbf{X}} = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}]$ is the sample matrix.

Rationale: By the *Law of Large Numbers*, if n is fixed and $m \rightarrow \infty$,

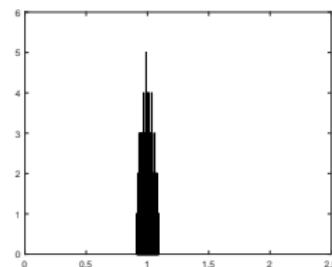
$$\hat{\Sigma} \rightarrow \mathbf{I}.$$

Empirical Covariance

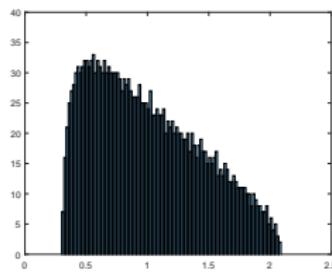
Let $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)} \in \mathbb{R}^n$ and $\mathbf{x}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Compute the eigenvalues of $\hat{\Sigma}$ in (1).

- $n = 200$ and $m = 10^5$ ($m \gg n$).



- $n = 2000$ and $m = 10^4$ ($m \gtrapprox n$).



Asymptotic Behavior of Empirical Covariance

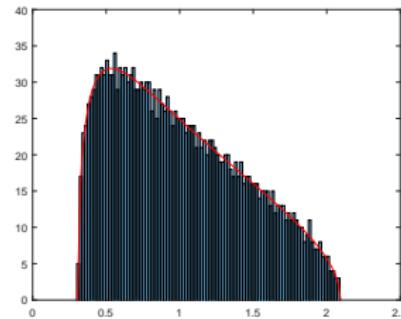
If $n, m \rightarrow \infty$ proportionally ($n/m \rightarrow r \in \mathbb{R}^+$):

the distribution of eigenvalues of empirical covariance matrix $\hat{\Sigma}$ converges to **Marchenko-Pastur** distribution

$$f(\lambda) = \begin{cases} \left(1 - \frac{1}{r}\right) \delta_{\lambda=0} + \frac{\sqrt{(\lambda^+ - \lambda)(\lambda - \lambda^-)}}{r\lambda} 1_{\lambda \in [\lambda^-, \lambda^+]} & \text{if } r \geq 1, \\ \frac{1}{2\pi} \frac{\sqrt{(\lambda^+ - \lambda)(\lambda - \lambda^-)}}{r\lambda} 1_{\lambda \in [\lambda^-, \lambda^+]} & \text{if } r \in (0, 1), \end{cases}$$

where $\lambda_{\pm} = (1 \pm \sqrt{r})^2$.

Quite different from the identity matrix.



Example 3: Linear Regression

Task: Given training samples (\mathbf{x}_i, y_i) , $i = 1, \dots, m$, want to estimate a linear function represented by $\boldsymbol{\alpha} \in \mathbb{R}^n$ s.t. $y_i \approx \langle \mathbf{x}_i, \boldsymbol{\alpha} \rangle$.

Solution: Let $\mathbf{y} = [y_1, \dots, y_m]^T$, $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^T$, and $\mathbf{e} = [e_1, \dots, e_m]$. Write

$$\mathbf{y} = \mathbf{X}\boldsymbol{\alpha} + \mathbf{e},$$

Then solve for $\boldsymbol{\alpha}$.

Issue: when $m < n$, there are infinite many valid solutions to $\mathbf{y} = \mathbf{X}\boldsymbol{\alpha}$.

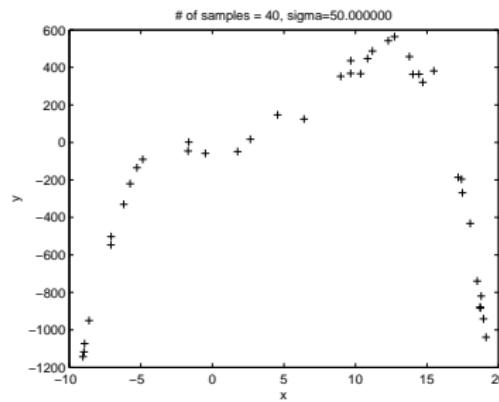
Linear Regression to Learn Nonlinear Function

Task: Learn an unknown *nonlinear* function f based on the input-output pairs (\mathbf{x}_i, y_i) , $i = 1, 2, \dots, m$, so that $y_i \approx f(\mathbf{x}_i)$.

Polynomial approximation - scalar case ($x_i \in \mathbb{R}$): Suppose that f can be approximated by a degree S polynomial:

$$f(x) = \sum_{s=0}^S \alpha_s x^s.$$

Example:



Good News

Fact 1.1

Given m distinct samples, \exists a polynomial of degree $m - 1$ to match the data perfectly.

Proof.

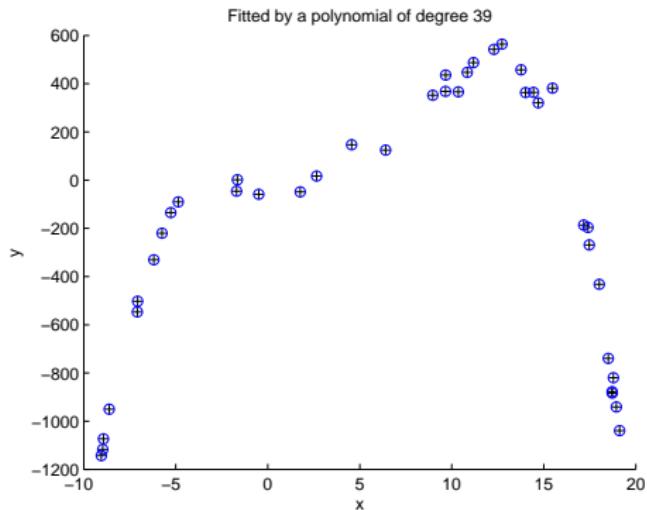
Since

$$\sum_{\ell=0}^{m-1} \alpha_\ell x_i^\ell = y_i, \quad i = 1, 2, \dots, m,$$

one can find f by computing α from

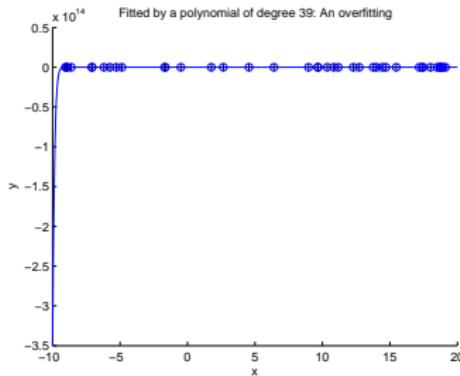
$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}_y = \underbrace{\begin{bmatrix} 1 & x_1 & \cdots & x_1^{m-1} \\ 1 & x_2 & \cdots & x_2^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \cdots & x_m^{m-1} \end{bmatrix}}_X \underbrace{\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{m-1} \end{bmatrix}}_\alpha.$$

A Solution that Looks Perfect

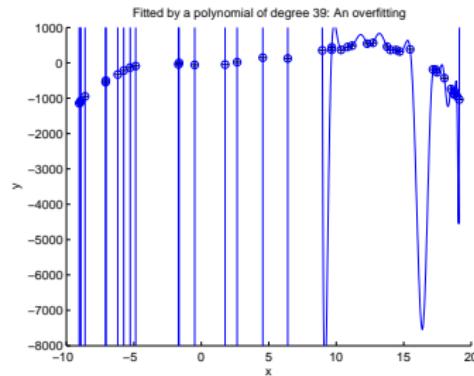


Bad News: Overfitting

Poor prediction performance



$$f(x)$$



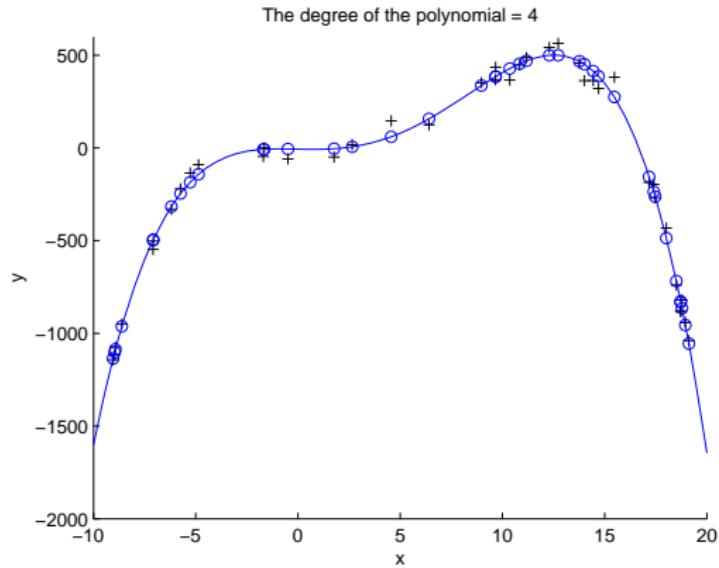
$$f(x) \text{ zoomed in}$$

Note that

$$\mathbf{y} = \mathbf{X}\boldsymbol{\alpha}_0 + \mathbf{w} \quad \Rightarrow \quad \hat{\boldsymbol{\alpha}} = \mathbf{X}^{-1}\mathbf{y} = \boldsymbol{\alpha}_0 + \mathbf{X}^{-1}\mathbf{w}.$$

The estimate $\hat{\boldsymbol{\alpha}}$ may overfit the noise.

A Sparse Approximation



α is sparse (only a few nonzeros): force $\alpha_5 = \alpha_6 = \dots = \alpha_{39} = 0$.

A More Realistic Example: Vector Input

Assume $y = f(\mathbf{x}) + w$, $\mathbf{x} \in \mathbb{R}^d$, f is a polynomial with $\deg(f) \leq 2$.

$$\begin{aligned}f(\mathbf{x}) = & \alpha_0 + \alpha_1 x_1 + \cdots + \alpha_d x_d \\& + \alpha_{d+1} x_1^2 + \alpha_{d+2} x_1 x_2 + \cdots + \alpha_{2d} x_1 x_d \\& + \alpha_{2d+1} x_2^2 + \cdots + \alpha_{n-2} x_{d-1} x_d \\& + \alpha_{n-1} x_d^2.\end{aligned}$$

Task: Given $(\mathbf{x}(j), y(j))$, $j = 1, 2, \dots, m$, try to find
 $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \dots, \alpha_{n-1}]$.

To Find the Polynomial

$$\underbrace{\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(m) \end{bmatrix}}_y = \underbrace{\begin{bmatrix} 1 & x_1(1) & \cdots & x_d^2(1) \\ 1 & x_1(2) & \cdots & x_d^2(2) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1(m) & \cdots & x_d^2(m) \end{bmatrix}}_X \underbrace{\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{bmatrix}}_\alpha.$$

Two issues:

- ▶ No sufficient data:
 - ▶ Number of terms $n = O(d^2) \gg m$.
- ▶ Even when data is abundant: need to avoid overfitting.

Sparsity plays a key role: Enforce most of the elements of α to be zero.

Circumvent the curse of dimensionality:

In most cases, the data have an intrinsic low-dimensional structure.

What is this Course About

- ▶ Programming X
- ▶ Computer architecture X
- ▶ Concepts and mechanisms ✓
 - ▶ Tools ✓
 - ▶ Sparse regression.
 - ▶ Convex Optimization (include SVM).
 - ▶ Statistical modeling and methods.
 - ▶ Elementary graph theory.
 - ▶ Applications that are good illustrations ✓
 - ▶ Denoising.
 - ▶ Face recognition with block occlusion.
 - ▶ Video foreground/background separation.
 - ▶ Recommendation engine: Netflix problem.
 - ▶ Community detection in social graph.

Image Denoising

Original



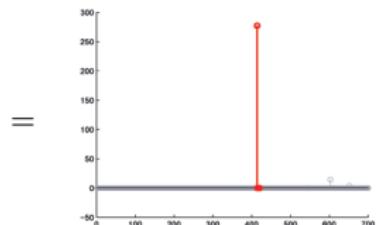
Noisy



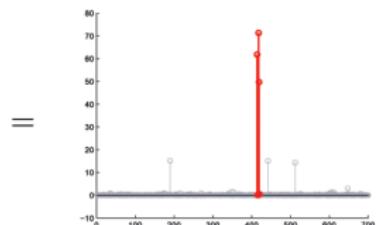
Denoised



Face Recognition with Block Occlusion [Wright et al., 2009]



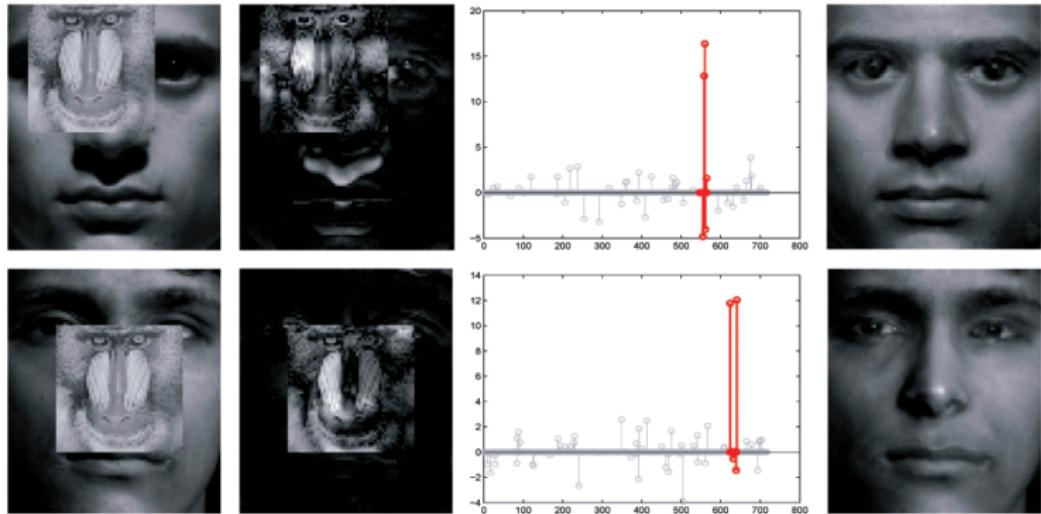
\times



\times



Face Recognition with Block Occlusion [Wright et al., 2009]



Netflix Problem

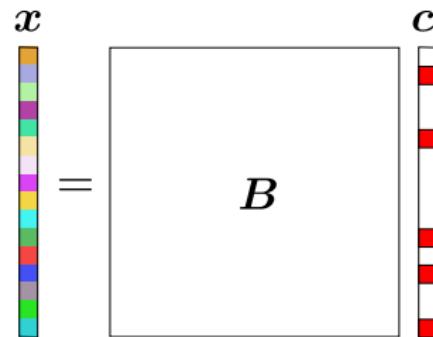
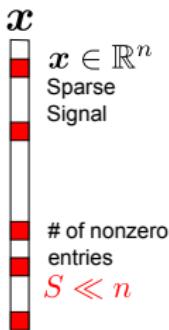


Section 2

Sparse Regression: Basics

Definition: Sparse Signals

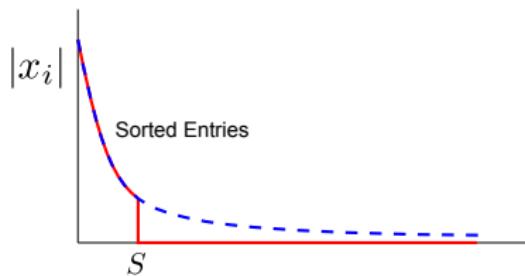
- *S-sparse*



\boldsymbol{B} is a given transform or dictionary.

Definition: Compressible Signals

- **Compressible signals:** can be well approximated by S -sparse signals.



- Natural pictures are compressible under DCT/Wavelet transform.
- Communication signals are often compressible under Fourier transform.
- In function approximation, it is typically assumed that the unknown function can be well approximated by a few 'kernel' functions.

A Mathematical Example

- ▶ Let \mathbf{x} be a vector. Suppose that the entries of \mathbf{x} obey a power law

$$|x_k| \leq c \cdot k^{-r}, \quad k = 1, 2, \dots$$

with a given $r > 1$.

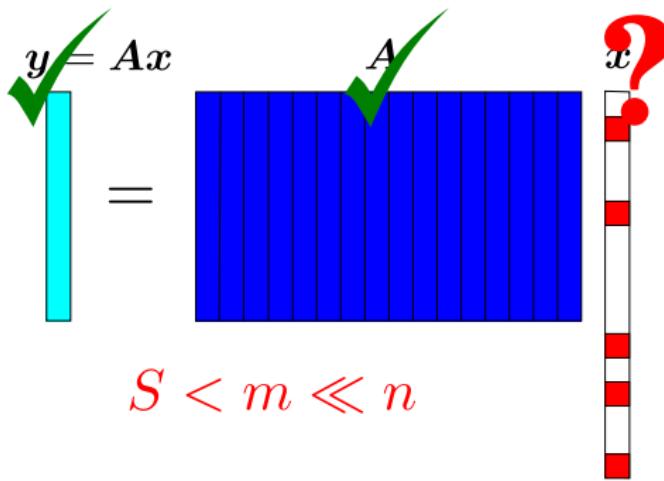
- ▶ Now use the leading S -term sub-vector to approximate \mathbf{x} , i.e.,

$$x_{S,k} = \begin{cases} x_k, & \text{if } 1 \leq k \leq S, \\ 0, & \text{if } k > S. \end{cases}$$

Then the best S -term approximation gives a distortion

$$\|\mathbf{x} - \mathbf{x}_S\|_2 \leq c' \cdot S^{-r+1/2}.$$

The Sparse Regression Problem



Given the observations \mathbf{y} and the dictionary \mathbf{A} , try to find a sparse \mathbf{x} such that $\mathbf{y} \approx \mathbf{Ax}$.

- ▶ Machine learning.
- ▶ Compressed sensing.

Compressed Sensing: Reducing the Number of Samples

Large and expensive sensors: reduce the cost/time of sensing

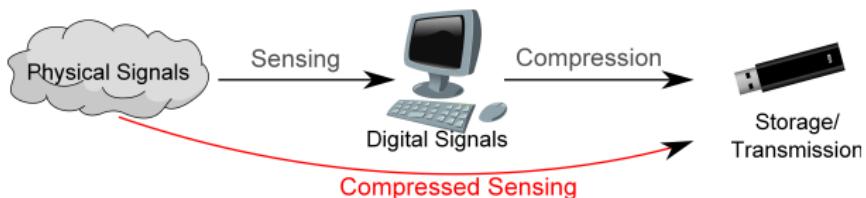
- ▶ Magnetic Resonance Imaging (MRI):



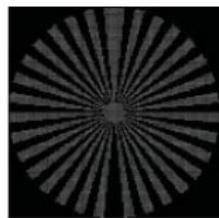
- ▶ Infrared sensing:



The Paradigm Shift: Compressed Sensing



An example in MRI



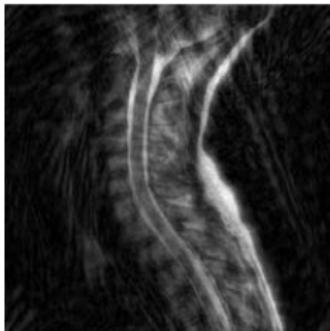
Trzasko, Manduca, Borisch (Mayo Clinic)

Sampling Pattern in Fourier domain

Fast Magnetic Resonance Imaging



Fully sampled



$6 \times$ undersampled
classical



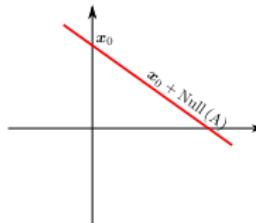
$6 \times$ undersampled
CS

Trzasko, Manduca, Borisch (Mayo Clinic)

The Solutions of the Problem

Problem: Find a sparse x such that $y = Ax$ where $A \in \mathbb{R}^{m \times n}$.

- ▶ Typically $m < n$.
Infinitely many solutions.



- ▶ Want a sparse solution, but
 - ▶ Do not know how many nonzero entries are there.
 - ▶ Do not know where the nonzero entries are.

The Least Squared Solution

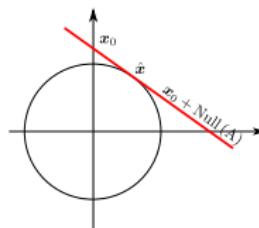
The least squared solution:

$$\hat{\mathbf{x}} = \mathbf{A}^\dagger \mathbf{y}$$

where \mathbf{A}^\dagger is the pseudo-inverse.

Or equivalently, choose $\hat{\mathbf{x}}$ to be the solution of the optimization problem:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_2, \quad \text{s.t. } \mathbf{y} = \mathbf{A}\mathbf{x}. \quad (2)$$



- ▶ Closed form solution.
- ▶ Not sparse. (Not what we want.)

Seeking for a Sparse Solution

Definition 2.1

The ℓ_0 **pseudo**-norm is defined as

$$\|\mathbf{x}\|_0 = \text{number of nonzero entries in } \mathbf{x}.$$

To find a sparse solution:

Noise-free case (our focus):

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \mathbf{y} = \mathbf{A}\mathbf{x}. \quad (3)$$

Noisy case (will not be discussed in details):

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon, \quad \text{or}$$

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0,$$

$\lambda \rightarrow 0$: enforce $\mathbf{y} = \mathbf{A}\mathbf{x}$.

$\lambda \rightarrow \infty$: the data consistency constraint does not matter.

Property of ℓ_0 Pseudo-norm

- The ℓ_0 pseudo-norm is discontinuous and nonconvex.
A demonstration of the discontinuity.

Let $e_1 = [1, 0, \dots, 0]^T$, $e_2 = [0, 1, 0, \dots, 0]^T$, \dots .

Then

$$\|e_1\|_0 = 1, \quad \text{but } \|e_1 + \epsilon e_2\|_0 = 2,$$

no matter how small $\epsilon \neq 0$ is.

- Solving (3) usually means an exhaustive search.
 - Prohibitive complexity $O(n^S)$.

Definition: Support Set and Truncation

Definition 2.2

Let $\boldsymbol{x} \in \mathbb{R}^n$. Its support set is defined as

$$\text{supp}(\boldsymbol{x}) = \{i : x_i \neq 0\}.$$

$$\boldsymbol{x} = \begin{bmatrix} 0.1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} \Rightarrow \mathcal{I} = \{1, 4\}.$$

Truncation

Definition 2.3

Let $\mathcal{I} \subset \{1, 2, \dots, n\}$ be an index set. Let $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$.

- ▶ $\mathbf{A}_{\mathcal{I}}$: a matrix formed by columns of \mathbf{A} indexed by \mathcal{I} .
 - ▶ $\mathbf{x}_{\mathcal{I}}$: a vector formed by entries of \mathbf{A} indexed by \mathcal{I} .

Example: Let $\mathcal{I} = \text{supp}(x)$. Then $y = Ax = A_{\mathcal{I}}x_{\mathcal{I}}$.

$$y = Ax \quad y = A_{\mathcal{I}}x_{\mathcal{I}}$$

Solving ℓ_0 -Minimization: Exhaustive Search

For $s = 1, 2, \dots$

Try all $\mathcal{I} \subset [n] \triangleq \{1, 2, \dots, n\}$ s.t. $|\mathcal{I}| = s$

Let $\hat{\mathbf{x}}_{\mathcal{I}} = \mathbf{A}_{\mathcal{I}}^\dagger \mathbf{y}$.

If $\mathbf{y} = \mathbf{A}_{\mathcal{I}} \hat{\mathbf{x}}_{\mathcal{I}}$, then terminate the search. Otherwise, continue.

End

Set $\mathbf{x}_{\mathcal{I}} = \mathbf{A}_{\mathcal{I}}^\dagger \mathbf{y}$ and $\mathbf{x}_{\mathcal{I}^c} = \mathbf{0}$.

Computational Complexity

Suppose that the exhaustive search terminates when $S^\# = \|\boldsymbol{x}\|_0$.
The computational complexity is approximately

$$\sum_{s=1}^{S^\#} \binom{n}{s} \geq \binom{n}{S^\#} = \frac{n!}{S^\#!(n - S^\#)!}.$$

Stirling approximation: $n! \approx \sqrt{2\pi} n^{n+1/2} e^{-n}$.

As a result, complexity is $O(n^{S^\#})$.

Conclusion: ℓ_0 -minimization is not practical for large n .

Feasible Ways?

- ▶ Greedy algorithms.
- ▶ Convex optimization.

Section 3

Linear Algebra

Linear Inverse Problem and Its Solutions

Given a system of linear equations

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w},$$

where $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$ are given and $\mathbf{w} \in \mathbb{R}^m$ is the noise, the task is to find the unknown vector $\mathbf{x} \in \mathbb{R}^n$.

- ▶ If $m = n$ and \mathbf{A} is invertible, then we typically compute $\hat{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{y}$.
- ▶ How about $m > n$, i.e., \mathbf{A} is a tall matrix?
- ▶ How about $m < n$, i.e., \mathbf{A} is a flat matrix?

Linear Independence and Dependence

Vectors $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{R}^m$ are **linearly independent** if
$$\sum \lambda_i \mathbf{v}_i = \mathbf{0} \Rightarrow \lambda_i = 0, \forall i.$$

In matrix format,

$$[\mathbf{v}_1, \dots, \mathbf{v}_n] \boldsymbol{\lambda} = \mathbf{0} \in \mathbb{R}^m \Rightarrow \boldsymbol{\lambda} = \mathbf{0} \in \mathbb{R}^n.$$

Vectors $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{R}^m$ are **linearly dependent** if $\exists \boldsymbol{\lambda} \neq \mathbf{0}$ such that
$$[\mathbf{v}_1, \dots, \mathbf{v}_n] \boldsymbol{\lambda} = \mathbf{0}.$$

Rank and Matrix Inverse

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a given matrix.

Column rank: the maximum number of linearly independent columns.

Row rank: the maximum number of linearly independent rows.

Rank: For every matrix, column rank = row rank = rank.

Definition 3.1 (Matrix Inverse and Pseudoinverse)

- ▶ A matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is invertible if there exists a $\mathbf{B} \in \mathbb{R}^{n \times n}$ such that

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I}.$$

- ▶ For a matrix \mathbf{A} , its pseudoinverse \mathbf{A}^\dagger is defined to satisfy
 - ▶ $\mathbf{AA}^\dagger \mathbf{A} = \mathbf{A}$; $\mathbf{A}^\dagger \mathbf{AA}^\dagger = \mathbf{A}^\dagger$.
 - ▶ $(\mathbf{AA}^\dagger)^T = \mathbf{AA}^\dagger$; $(\mathbf{A}^\dagger \mathbf{A})^T = \mathbf{A}^\dagger \mathbf{A}$.
- ▶ \mathbf{A} is invertible $\Leftrightarrow \mathbf{A}$ is of full rank.

Examples

```
A = [2 0; 0 4]
```

```
A =
```

$$\begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$$

```
B = inv(A)
```

```
B =
```

$$\begin{bmatrix} 0.5000 & 0 \\ 0 & 0.2500 \end{bmatrix}$$

```
A = [2 0; 0 0]
```

```
A =
```

$$\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

```
B = inv(A)
```

```
[Warning: Matrix is singular to working precision.]
```

```
B =
```

$$\begin{bmatrix} \text{Inf} & \text{Inf} \\ \text{Inf} & \text{Inf} \end{bmatrix}$$

```
C = pinv(A)
```

```
C =
```

$$\begin{bmatrix} 0.5000 & 0 \\ 0 & 0 \end{bmatrix}$$

Eigen-decomposition

Definition 3.2 (**Eigendecomposition**, spectral decomposition)

A non-zero vector $\mathbf{v} \in \mathbb{R}^n$ is an eigenvector of a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ if there is a constant λ such that

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \quad (4)$$

where λ is called the **eigenvalue** corresponding to \mathbf{v} .

If matrix \mathbf{A} can be eigendecomposed and if none of its eigenvalues are zero, then \mathbf{A} is invertible (nonsingular) and its inverse is given by

$$\mathbf{A}^{-1} = \mathbf{Q}\boldsymbol{\Lambda}^{-1}\mathbf{Q}^{-1}, \quad (5)$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix with

$$\boldsymbol{\Lambda}_{i,i} = \lambda_i.$$

An Example

```
A = [1 2; 1 3] M1 = [A*V(:,1) - D(1,1)*V(:,1), A*V(:,2) - D(2,2)*V(:,2)]  
A = M1 =  
1 2 1.0e-16 *  
1 3 0.5551 0  
[V,D] = eig(A) -0.8327 0  
  
V = M2 = V*inv(D)*inv(V)*A  
-0.9391 -0.5907  
0.3437 -0.8069  
M2 =  
1.0000 0.0000  
0.0000 1.0000  
D =  
0.2679 0  
0 3.7321
```

Homework (Examples)

- ▶ Show that the definition in (5) satisfies $\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$.
- ▶ Use the definition (4), compute the eigendecomposition of
 - ▶ $\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, and $\mathbf{C} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$.
 - ▶ Compare your results with those given by Matlab.
 - ▶ Find their inverse and pseudo-inverse.

Singular Value Decomposition

SVD: For an arbitrary matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, there exists a factorization of the form

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T,$$

where $\mathbf{U} \in \mathbb{R}^{m \times m}$ is unitary (contains m orthonormal columns), $\Sigma \in \mathbb{R}^{m \times n}$ is diagonal matrix with non-negative diagonal entries, and $\mathbf{V} \in \mathbb{R}^{n \times n}$ is unitary.

The m columns of \mathbf{U} and the n columns of \mathbf{V} are called the **left-singular vectors** and **right-singular vectors** of \mathbf{M} , respectively. The diagonal entries σ_i of Σ are known as the **singular values** of \mathbf{M} .

A convention is to list the singular values in descending order, that is, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m$ (assuming $m \leq n$). In this case, the diagonal matrix Σ is uniquely determined.

An Example

```
A = [1 2; 3 4; 5 6; 7 8]
```

```
A =
```

```
1      2  
3      4  
5      6  
7      8
```

```
[U,S,V]=svd(A)
```

```
U =
```

```
-0.1525   -0.8226   -0.3945   -0.3800  
-0.3499   -0.4214    0.2428    0.8007  
-0.5474   -0.0201    0.6979   -0.4614  
-0.7448    0.3812   -0.5462    0.0407
```

```
S =
```

```
14.2691      0  
    0     0.6268  
    0      0  
    0      0
```

```
V =
```

```
-0.6414    0.7672  
-0.7672   -0.6414
```

Compact SVD

Compact SVD only shows r columns of \mathbf{U} and r rows of \mathbf{V}^T corresponding to r nonzero singular values $\Sigma \in \mathbb{R}^{r \times r}$.

```
[U,S,V]=svd(A,0)
```

$\mathbf{U} =$

```
-0.1525 -0.8226
-0.3499 -0.4214
-0.5474 -0.0201
-0.7448 0.3812
```

$\mathbf{S} =$

```
14.2691 0
0 0.6268
```

$\mathbf{V} =$

```
-0.6414 0.7672
-0.7672 -0.6414
```

Compact SVD: Flat Matrices

$$\begin{matrix} M \\ \boxed{} \end{matrix} = \begin{matrix} U \\ \boxed{} \end{matrix} \begin{matrix} \Sigma \\ \boxed{} \end{matrix} \begin{matrix} V^T \\ \boxed{} \end{matrix}$$
$$= \begin{matrix} U \\ \boxed{} \end{matrix} \begin{matrix} \Sigma_r \\ \boxed{} \end{matrix} \begin{matrix} V_r^T \\ \boxed{} \end{matrix}$$

Compact SVD: Tall Matrices

$$\begin{matrix} M \\ \boxed{} \end{matrix} = \begin{matrix} U \\ \boxed{} \end{matrix} \begin{matrix} \Sigma \\ \boxed{} \end{matrix} \begin{matrix} V^T \\ \boxed{} \end{matrix}$$
$$= \begin{matrix} U_r \\ \boxed{} \end{matrix} \begin{matrix} \Sigma_r \\ \boxed{} \end{matrix} \begin{matrix} V^T \\ \boxed{} \end{matrix}$$

SVD and ED

Let

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T.$$

Then

$$\mathbf{M}\mathbf{M}^T = \mathbf{U}\Sigma^2\mathbf{U}^T; \quad \text{and } \mathbf{M}^T\mathbf{M} = \mathbf{V}\Sigma^2\mathbf{V}^T$$

- ▶ The left-singular vectors \mathbf{u}_i 's of \mathbf{M} are eigenvectors of $\mathbf{M}\mathbf{M}^T$.
- ▶ The right-singular vectors \mathbf{v}_i 's of \mathbf{M} are eigenvectors of $\mathbf{M}^T\mathbf{M}$.
- ▶ The singular values σ_i 's of \mathbf{M} are the square roots of the eigenvalues of both $\mathbf{M}\mathbf{M}^T$ and $\mathbf{M}^T\mathbf{M}$. That is, $\lambda_i = \sigma_i^2$.

Pseudoinverse and SVD

Consider the compact SVD

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T.$$

Then it holds that

$$\mathbf{A}^\dagger = \mathbf{V}\Sigma^\dagger\mathbf{U}^T.$$

- ▶ $\mathbf{A}\mathbf{A}^\dagger = \mathbf{U}_r\mathbf{U}_r^T$ and $\mathbf{A}\mathbf{A}^\dagger\mathbf{A} = \mathbf{A}$.
 - ▶ \mathbf{U}_r contains the first r columns of \mathbf{U} where r is the rank.
- ▶ Similarly, $\mathbf{A}^\dagger\mathbf{A} = \mathbf{V}_r\mathbf{V}_r^T$ and $\mathbf{A}^\dagger\mathbf{A}\mathbf{A}^\dagger = \mathbf{A}^\dagger$.

Pseudoinverse: An Illustration

$$M \quad M^\dagger = \begin{matrix} \boxed{} & \boxed{} & = & \boxed{} & \boxed{} & \boxed{} \end{matrix} \neq I$$

The diagram illustrates the non-commutativity of matrix multiplication involving the pseudoinverse. On the left, M and M^\dagger are multiplied. The result is a 3x3 matrix where the middle column contains a diagonal line from top-right to bottom-left, indicating it is not the identity matrix I . Dashed lines separate the columns.

$$M^\dagger \quad M = \begin{matrix} \boxed{} & \boxed{} & = & \boxed{} & \boxed{} & \boxed{} \end{matrix} = I$$

The diagram illustrates the commutativity of matrix multiplication involving the pseudoinverse. On the left, M^\dagger and M are multiplied. The result is the identity matrix I , represented by three separate boxes.

Linear Subspace and Basis

Definition 3.3 (Linear subspace)

Let $\mathcal{B} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\} \subset \mathbb{R}^m$ containing linearly independent vectors.

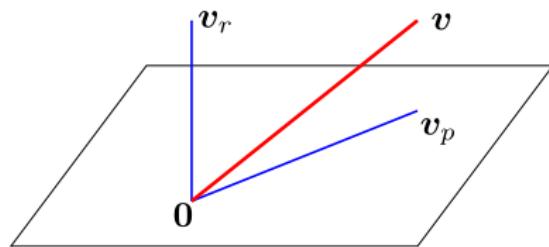
The **linear span** of \mathcal{B} is defined as

$$\text{span}(\mathcal{B}) = \left\{ \sum_{i=1}^n \lambda_i \mathbf{v}_i : \lambda_i \in \mathbb{R} \right\}.$$

It is a **linear subspace** of \mathbb{R}^m .

- ▶ The set \mathcal{B} is a **basis** for the linear subspace $\mathcal{S} = \langle \mathcal{B} \rangle$.
 - ▶ The basis \mathcal{B} **may not be unique**, but its dimension is.
 - ▶ **$\dim(\mathcal{S}) = n$** : the # of vectors in a basis.
- ▶ \mathcal{B} is orthonormal if $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0$ for $i \neq j$ and $\langle \mathbf{v}_i, \mathbf{v}_i \rangle = 1$.
- ▶ For convenience, we use $\text{span}(\mathcal{B})$ and $\text{span}(\mathbf{B})$ interchangeably where $\mathbf{B} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$.

Projection



Definition 3.4 (Projection)

The **projection** of $\mathbf{x} \in \mathbb{R}^n$ onto the subspace $\text{span}(\mathbf{A})$ is defined as

$$\mathbf{x}_p = \text{proj}(\mathbf{x}, \mathbf{A}) = \mathbf{A}\mathbf{A}^\dagger\mathbf{x}.$$

And the **projection residue** is given by

$$\mathbf{x}_r = \text{resid}(\mathbf{x}, \mathbf{A}) = \mathbf{x} - \mathbf{x}_p.$$

Projection Viewed in SVD

Consider an n -d subspace in \mathbb{R}^m with $m > n$.

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be its basis matrix. Clearly \mathbf{A} is a tall matrix.

Consider the compact SVD $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$. Then

$$\mathbf{A}\mathbf{A}^\dagger = \mathbf{U}\mathbf{U}^T.$$

And

$$\mathbf{x}_p = \mathbf{U}\mathbf{U}^T\mathbf{x} = \mathbf{U}\mathbf{w}_x,$$

where \mathbf{U} is an orthonormal basis for $\text{span}(\mathbf{A})$ and \mathbf{w}_x is the *projection coefficients*.

Projection Residue Vector

Corollary 3.5

Suppose that $\mathbf{x}_r = \text{resid}(\mathbf{x}, \mathbf{A}) \neq \mathbf{0}$. Then \mathbf{x}_r is orthogonal to \mathbf{A} , i.e., $\mathbf{A}^T \mathbf{x}_r = \mathbf{0}$.

Proof:

$$\begin{aligned}\mathbf{A}^T \mathbf{x}_r &= \mathbf{A}^T (\mathbf{x} - \mathbf{A}\mathbf{A}^\dagger \mathbf{x}) \\ &= \mathbf{V}\Sigma\mathbf{U}^T (\mathbf{x} - \mathbf{U}\mathbf{U}^T \mathbf{x}) \\ &= \mathbf{V}\Sigma\mathbf{U}^T \mathbf{x} - \mathbf{V}\Sigma\mathbf{U}^T \mathbf{x} = \mathbf{0}\end{aligned}$$

Back to Linear Inverse Problem

Given

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w},$$

an estimation of \mathbf{x} is given by

$$\hat{\mathbf{x}} = \mathbf{A}^\dagger \mathbf{y} = \begin{cases} \mathbf{x} + \mathbf{A}^\dagger \mathbf{w}, & \text{if } m \geq n, \\ \text{proj}(\mathbf{x}, \mathbf{A}^T) + \mathbf{A}^\dagger \mathbf{w}, & \text{if } m < n. \end{cases}$$

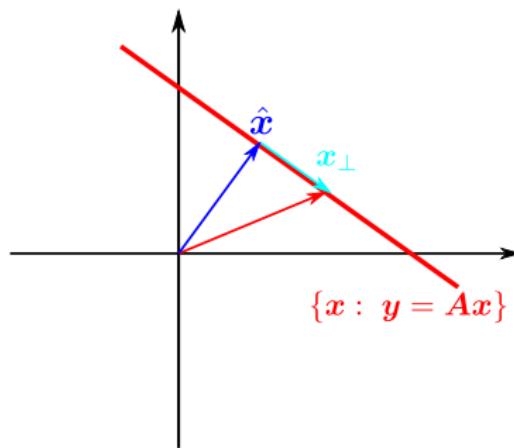
- ▶ The case $m < n$:

Consider the compact SVD of \mathbf{A} : $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}_m^T$. Clearly

$$\mathbf{A}^\dagger = \mathbf{V}_m \mathbf{\Lambda}^{-1} \mathbf{U}^T, \text{ and } \mathbf{A}^\dagger \mathbf{A}\mathbf{x} = \mathbf{V}_m \mathbf{V}_m^T \mathbf{x},$$

which is $\text{proj}(\mathbf{x}, \mathbf{V}_m) = \text{proj}(\mathbf{x}, \mathbf{A}^T)$.

A Geometric Picture



$$\mathcal{X} := \{x : y = Ax\} = \text{span}(V_{\perp}) + \hat{x},$$

where $V_{\perp} \in \mathbb{R}^{n \times (n-m)}$ is the orthogonal complement of V_m .

Link to the Least Squared Problem (2)

$\hat{\mathbf{x}} := \mathbf{A}^\dagger \mathbf{y}$ is the solution of

$$\min_{\mathbf{x}} \|\mathbf{x}\|_2, \quad \text{s.t. } \mathbf{y} = \mathbf{A}\mathbf{x}.$$

- ▶ For all $\mathbf{x} \in \mathcal{X}$,

$$\begin{aligned}\mathbf{x} &= \mathbf{I}\mathbf{x} = \mathbf{V}\mathbf{V}^T\mathbf{x} = [\mathbf{V}_m \mathbf{V}_\perp] \begin{bmatrix} \mathbf{V}_m^T \\ \mathbf{V}_\perp^T \end{bmatrix} \mathbf{x} \\ &= \underbrace{\mathbf{V}_m \mathbf{V}_m^T \mathbf{x}}_{\hat{\mathbf{x}}} + \underbrace{\mathbf{V}_\perp \mathbf{V}_\perp^T \mathbf{x}}_{\mathbf{x}_\perp}.\end{aligned}$$



$$\begin{aligned}\|\mathbf{x}\|_2^2 &= \langle \hat{\mathbf{x}} + \mathbf{x}_\perp, \hat{\mathbf{x}} + \mathbf{x}_\perp \rangle = \hat{\mathbf{x}}^T \hat{\mathbf{x}} + 2\hat{\mathbf{x}}^T \mathbf{x}_\perp + \mathbf{x}_\perp^T \mathbf{x}_\perp \\ &= \|\hat{\mathbf{x}}\|_2^2 + \|\mathbf{x}_\perp\|_2^2 \geq \|\hat{\mathbf{x}}\|_2^2.\end{aligned}$$

Section 4

Greedy Algorithms

Greedy Algorithms: the Approach

Recall: $\|\mathbf{x}\|_0 = \text{number of nonzero entries in } \mathbf{x}$.

- ▶ When we roughly know the sparsity $\|\mathbf{x}\|_0$,

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \text{ s.t. } \|\mathbf{x}\|_0 \leq S.$$

- ▶ Otherwise if we roughly know the noise energy,

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon.$$

Major Greedy Algorithms

- ▶ Orthogonal matching pursuit (OMP)
- ▶ Subspace pursuit (SP)
- ▶ Compressive sampling matching pursuit (CoSaMP)
- ▶ Iterative hard thresholding (IHT)

Intuition

$$\mathbf{y} = \mathbf{A}\mathbf{x} \quad \Rightarrow \quad \mathbf{A}^T\mathbf{y} = \mathbf{A}^T\mathbf{A}\mathbf{x}.$$

Under the assumption that

- ▶ Columns of \mathbf{A} are normalised.
- ▶ Columns of \mathbf{A} are near orthogonal.

$\mathbf{A}^T\mathbf{y}$ “looks like” \mathbf{x} .

From now on, we assume that columns of \mathbf{A} are normalised.

Intuition: When $S = 1$

When $S = 1$: The location of the nonzero entry is given by

$$i^* = \arg \max_i |\mathbf{a}_i^T \mathbf{y}|$$

Once i^* is found,

$$x_{i^*} = \mathbf{a}_{i^*}^\dagger \mathbf{y}, \quad x_j = 0, \quad \forall j \neq i^*.$$

Intuition: $S = 2$

Suppose that we knew $S = 2$ and the location of one nonzero entry, i.e. the support set $\mathcal{I} = \{i_1, ?\}$.

- ▶ Cancel the effect from i_1 :

$$\mathbf{y}_r := \mathbf{y} - \mathbf{a}_{i_1} \mathbf{a}_{i_1}^\dagger \mathbf{y} = \mathbf{y} - \mathbf{a}_{i_1} \mathbf{a}_{i_1}^T \mathbf{y}.$$

- ▶ Choose i_2 via

$$i_2 = \arg \max_i |\langle \mathbf{a}_i, \mathbf{y}_r \rangle|.$$

Remark: It holds that $i_2 \neq i_1$. We get two locations indeed.

Proof: Clearly \mathbf{y}_r is orthogonal to \mathbf{a}_{i_1} , i.e. $\langle \mathbf{y}_r, \mathbf{a}_{i_1} \rangle = 0$.

Intuition: $S = 3$

Suppose that we knew $S = 3$ and the locations of two nonzero entries, i.e. the support set $\mathcal{I} = \{i_1, i_2, ?\}$.

- ▶ Cancel the effect from i_1 and i_2 : Let $\mathcal{I}_2 = \{i_1, i_2\}$.

$$\mathbf{y}_r := \mathbf{y} - \mathbf{A}_{\mathcal{I}_2} \mathbf{A}_{\mathcal{I}_2}^\dagger \mathbf{y}.$$

- ▶ Choose i_2 via

$$i_3 = \arg \max_i |\langle \mathbf{a}_i, \mathbf{y}_r \rangle|.$$

Remark: It holds that $i_3 \notin \mathcal{I}_2$. We get three locations.

The Orthogonal Matching Pursuit (OMP) Algorithm

Input: S , \mathbf{A} , \mathbf{y} .

Initialization:

$\mathbf{x} = \mathbf{0}$, $\mathcal{T}^\ell = \phi$, and $\mathbf{y}_r = \mathbf{y}$.

Iteration: $\ell = 1, 2, \dots, S$

1. Let $i_\ell = \arg \max_j |\langle \mathbf{a}_j, \mathbf{y}_r \rangle|$

2. $\mathcal{T}^\ell = \mathcal{T}^{\ell-1} \cup \{i_\ell\}$. (Add one index)

3. $\mathbf{x}_{\mathcal{T}^\ell} = \mathbf{A}_{\mathcal{T}^\ell}^\dagger \mathbf{y}$. (Estimate ℓ -sparse signal)

4. $\mathbf{y}_r = \mathbf{y} - \mathbf{A}\mathbf{x}$. (Compute estimation error)

Performance?

Suppose that

$$\mathbf{y} = \mathbf{A}\mathbf{x}_0 + \mathbf{w},$$

where the signal \mathbf{x}_0 is S -sparse and the noise satisfies $\|\mathbf{w}\|_2 \leq \epsilon$.

The question is

$$\|\hat{\mathbf{x}} - \mathbf{x}_0\|_2 \leq ?.$$

- ▶ Noise free case ($\epsilon = 0$): when $\hat{\mathbf{x}} = \mathbf{x}_0$?
- ▶ Noisy case ($\epsilon > 0$):
 - ▶ How the recovery error $\|\hat{\mathbf{x}} - \mathbf{x}_0\|_2$ behaves with ϵ .
- ▶ Approximately sparse case:
 - ▶ Let $\mathbf{x}_{0,S}$ be the best S -term approximation of \mathbf{x}_0 .
 - ▶ How the recovery error $\|\hat{\mathbf{x}} - \mathbf{x}_0\|_2$ behaves with
 - ▶ ϵ , and
 - ▶ $\|\mathbf{x}_0 - \mathbf{x}_{0,S}\|_2$.

Performance Guarantee of OMP: Mutual Coherence

Definition 4.1 (Mutual coherence)

The mutual coherence of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, denoted by $\mu(\mathbf{A})$, is the maximal correlation (in magnitude) between two (normalized) columns.

$$\mu(\mathbf{A}) = \max_{i \neq j} \frac{|\langle \mathbf{a}_i, \mathbf{a}_j \rangle|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2}.$$

When $\|\mathbf{a}_i\|_2 = 1$, $\forall i \in [n]$, then $\mu(\mathbf{A}) = \max_{i \neq j} |\langle \mathbf{a}_i, \mathbf{a}_j \rangle|$.

Performance Guarantee of OMP

Theorem 4.2

Suppose that \mathbf{A} satisfies that

$$\mu < \frac{1}{2S}.$$

Then the OMP algorithm is guaranteed to exactly recover all S -sparse \mathbf{x} from \mathbf{y} .

The key for the proof: To show $\hat{\mathbf{x}} = \mathbf{x}_0$:

- ▶ Want to show that $\text{supp}(\hat{\mathbf{x}}) = \text{supp}(\mathbf{x}_0)$.
- ▶ Or show that at the ℓ -th iteration of OMP, the chosen index $i_\ell \in \mathcal{T}_0 := \text{supp}(\mathbf{x}_0)$.

The proof needs Cauchy–Schwartz Inequality in Theorem 4.9 in Appendix.

The First Iteration of OMP (1)

Want to show that $i_1 := \arg \max_i |\langle \mathbf{a}_i, \mathbf{y} \rangle| \in \mathcal{T}_0$.

- ▶ $\forall i, |\langle \mathbf{a}_i, \mathbf{y} \rangle| = \left| \left\langle \mathbf{a}_i, \sum_{j \in \mathcal{T}_0} \mathbf{a}_j x_{0,j} \right\rangle \right| = \left| \sum_{j \in \mathcal{T}_0} x_{0,j} \langle \mathbf{a}_i, \mathbf{a}_j \rangle \right|.$
- ▶ For all $i \notin \mathcal{T}_0$:

$$\begin{aligned} |\langle \mathbf{a}_i, \mathbf{y} \rangle| &= \left| \sum_{j \in \mathcal{T}_0} x_{0,j} \langle \mathbf{a}_i, \mathbf{a}_j \rangle \right| \leq \sum_{j \in \mathcal{T}_0} |x_{0,j}| |\langle \mathbf{a}_i, \mathbf{a}_j \rangle| \\ &\leq \mu \sum_{j \in \mathcal{T}_0} |x_{0,j}| \stackrel{(a)}{\leq} \mu \sqrt{S} \|\mathbf{x}\|_2 \end{aligned}$$

where (a) follows from Cauchy–Schwartz Inequality (Theorem 4.9).

- ▶ Hence,

$$\max_{i \notin \mathcal{T}_0} |\langle \mathbf{a}_i, \mathbf{y} \rangle| \leq \mu \sqrt{S} \|\mathbf{x}\|_2. \quad (6)$$

The First Iteration of OMP (2)

- For all $i \in \mathcal{T}_0$:

$$\begin{aligned} |\langle \mathbf{a}_i, \mathbf{y} \rangle| &= \left| \sum_{j \in \mathcal{T}_0} x_{0,j} \langle \mathbf{a}_i, \mathbf{a}_j \rangle \right| \geq |\mathbf{x}_{0,i} \langle \mathbf{a}_i, \mathbf{a}_i \rangle| - \left| \sum_{j \neq i} x_{0,j} \langle \mathbf{a}_i, \mathbf{a}_j \rangle \right| \\ &\geq |x_{0,i}| - \mu \sum_{j \neq i} |x_{0,j}| \stackrel{(a)}{\geq} |x_{0,i}| - \mu \sqrt{S} \|\mathbf{x}\|_2, \end{aligned}$$

where (a) follows from Cauchy–Schwartz Inequality.

-
- $$\max_{i \in \mathcal{T}_0} |\langle \mathbf{a}_i, \mathbf{y} \rangle| \geq \frac{1}{\sqrt{S}} \|\mathbf{x}\|_2 - \mu \sqrt{S} \|\mathbf{x}\|_2,$$

where we have used the fact that

$$\frac{1}{\sqrt{S}} \|\mathbf{x}\|_2 = \frac{(\sum x_i^2)^{\frac{1}{2}}}{\sqrt{S}} \leq \frac{\left(\sum (\max_i |x_i|)^2 \right)^{\frac{1}{2}}}{\sqrt{S}} = \max_{i \in \mathcal{T}_0} |x_i|. \quad (7)$$

The First Iteration of OMP (3)

- ▶ Now suppose that $\mu < \frac{1}{2S}$ (the assumption in Theorem 4.2). Then

$$\frac{1}{\sqrt{S}} \|\mathbf{x}\|_2 > 2\mu\sqrt{S} \|\mathbf{x}\|_2,$$

- ▶ Or equivalently,

$$\max_{i \in \mathcal{T}_0} |\langle \mathbf{a}_i, \mathbf{y} \rangle| \geq \frac{1}{\sqrt{S}} \|\mathbf{x}\|_2 - \mu\sqrt{S} \|\mathbf{x}\|_2 > \mu\sqrt{S} \|\mathbf{x}\|_2 \geq \max_{i \notin \mathcal{T}_0} |\langle \mathbf{a}_i, \mathbf{y} \rangle|.$$

- ▶ One concludes that

$$i_1 \in \mathcal{T}_0.$$

The ℓ^{th} Iteration: Mathematical Induction

- ▶ Let $i_1, \dots, i_{\ell-1}$ be the indices chosen in the first $\ell - 1$ iterations.
Let $\mathcal{T}^{\ell-1} = \{i_1, \dots, i_{\ell-1}\}$. Assume that $\mathcal{T}^{\ell-1} \subset \mathcal{T}_0$.
- ▶ Then

$$\mathbf{y}_r = \mathbf{y} - \mathbf{A}_{\mathcal{T}^{\ell-1}} \mathbf{A}_{\mathcal{T}^{\ell-1}}^\dagger \mathbf{y} = \mathbf{y} - \mathbf{A}_{\mathcal{T}^{\ell-1}} \tilde{\mathbf{y}}_{\ell-1} \in \text{span}(\mathbf{A}_{\mathcal{T}_0}).$$

Or

$$\mathbf{y}_r = \mathbf{A}_{\mathcal{T}_0} \tilde{\mathbf{v}}_{\mathcal{T}_0}.$$

for some $\tilde{\mathbf{v}}_{\mathcal{T}_0}$.

- ▶ Use the same arguments as before, $i_\ell \in \mathcal{T}_0$.
At the same time, $\mathbf{A}_{\mathcal{T}^{\ell-1}}^T \mathbf{y}_r = \mathbf{0}$ and hence $i_\ell \notin \mathcal{T}^{\ell-1}$.
 $|\mathcal{T}^\ell| = \ell$.
- ▶ OMP algorithm needs S iterations to recover S -sparse signals.

Hard Thresholding Function

Hard thresholding function $H_S(\mathbf{a})$:

Set all but the largest (in magnitude) S elements of \mathbf{a} to zero.

Example:

$$\mathbf{a} = [3, -4, 1] \Rightarrow H_1(\mathbf{a}) = [0, -4, 0] \text{ & } H_2(\mathbf{a}) = [3, -4, 0].$$

$\text{supp}(\mathbf{a})$: Index set of nonzero entries in \mathbf{a} .

$$\text{supp}(H_1(\mathbf{a})) = \arg \max_i |a_i|.$$

$$\text{supp}(H_S(\mathbf{a})) = \{S \text{ indices of the largest magnitude entries in } \mathbf{a}\}.$$

In the following greedy algorithms:

$$\text{supp}(H_1(\mathbf{A}^T \mathbf{y})) = \arg \max_j |\langle \mathbf{y}, \mathbf{a}_j \rangle|.$$

$$\text{supp}(H_S(\mathbf{A}^T \mathbf{y})) = \{S \text{ indices corr. to the } S \text{ largest } |\langle \mathbf{y}, \mathbf{a}_j \rangle|\}.$$

The Subspace Pursuit (SP) Algorithm

Input: S , \mathbf{A} , \mathbf{y} .

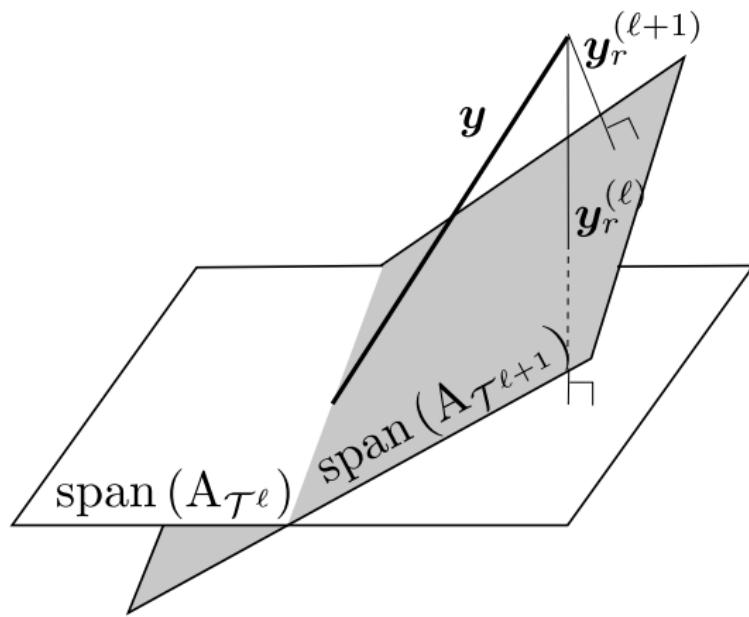
Initialization:

1. $\mathcal{T}^0 = \text{supp}(H_S(\mathbf{A}^T \mathbf{y}))$.
2. $\mathbf{y}_r = \text{resid}(\mathbf{y}, \mathbf{A}_{\mathcal{T}^0})$.

Iteration: $\ell = 1, 2, \dots$ until exit criteria are true.

1. $\tilde{\mathcal{T}}^\ell = \mathcal{T}^{\ell-1} \cup \text{supp}(H_S(\mathbf{A}^T \mathbf{y}_r))$. (Expand support)
2. Let $\mathbf{b}_{\tilde{\mathcal{T}}^\ell} = \mathbf{A}_{\tilde{\mathcal{T}}^\ell}^\dagger \mathbf{y}$ and $\mathbf{b}_{(\tilde{\mathcal{T}}^\ell)^c} = \mathbf{0}$. (Estimate $2S$ -sparse signal)
3. Set $\mathcal{T}^\ell = \text{supp}(\mathbf{b})$. (Shrink support)
4. Let $\mathbf{x}_{\mathcal{T}^\ell}^\ell = \mathbf{A}_{\mathcal{T}^\ell}^\dagger \mathbf{y}$ and $\mathbf{x}_{(\mathcal{T}^\ell)^c}^\ell = \mathbf{0}$. (Estimate S -sparse signal)
5. Let $\mathbf{y}_r = \mathbf{y} - \mathbf{A}\mathbf{x}^\ell$. (Compute estimation error)

Geometric Interpretation



The Compressive Sampling Matching Pursuit (CoSaMP) Algorithm

Input: $S, \mathbf{A}, \mathbf{y}$.

Initialization:

$\mathbf{x}^0 = \mathbf{0}$, and $\mathbf{y}_r = \mathbf{y}$.

Iteration: $\ell = 1, 2, \dots$ until exit criterion true.

1. $\tilde{\mathcal{T}}^\ell = \mathcal{T}^{\ell-1} \cup \text{supp}(H_{2S}(\mathbf{A}^T \mathbf{y}_r))$. (Expand support)
2. Let $\mathbf{b}_{\tilde{\mathcal{T}}^\ell} = \mathbf{A}_{\tilde{\mathcal{T}}^\ell}^\dagger \mathbf{y}$ and $\mathbf{b}_{(\tilde{\mathcal{T}}^\ell)^c} = \mathbf{0}$. (Estimate $3S$ -sparse signal)
3. $\mathbf{x}^\ell = H_S(\mathbf{b})$. ($\mathcal{T}^\ell = \text{supp}(H_S(\mathbf{b}))$). (Shrink support)
4. $\mathbf{y}_r = \mathbf{y} - \mathbf{A}\mathbf{x}^\ell$. (Update estimation error)

The Iterative Hard Thresholding (IHT) Algorithm

Input: S , \mathbf{A} , \mathbf{y} .

Initialization:

$$\mathbf{x}^0 = \mathbf{0}.$$

Iteration: $\ell = 1, 2, \dots$ until exit criterion true.

$$\mathbf{x}^\ell = H_S \left(\mathbf{x}^{\ell-1} + \mathbf{A}^T \left(\mathbf{y} - \mathbf{A}\mathbf{x}^{\ell-1} \right) \right).$$

A more general form: for some $\mu > 0$.

$$\mathbf{x}^\ell = H_S \left(\mathbf{x}^{\ell-1} + \mu \mathbf{A}^T \left(\mathbf{y} - \mathbf{A}\mathbf{x}^{\ell-1} \right) \right).$$

Comments

History

- ▶ MP: Friedman and Stuetzle, 1981; Mallat and Zhang, 1993; Qian and Chen, 1994.
- ▶ OMP: Chen, et al., 1989; Pati, et al., 1993; Davis, et al., 1994.
Analysed by Tropp, 2004.
- ▶ SP: Dai and Milenkovic, 2009. (Online available 06/03/2008)
CoSaMP: Needell and Tropp, 2009. (Online available 17/03/2008)
IHT: Blumensath and Davies, 2009. (Online available 05/05/2008)

Comparison:

	# of measurements	# of iterations
Exhaustive Search	$2S + 1$	$\binom{n}{S} = O(n^S)$
OMP	$O(S^2 \log n)$	S
SP, CoSaMP, IHT	$O(S \cdot \log \frac{n}{S})$	Typically $O(\log S)$, at most S

of measurements is based on random Gaussian matrices.

Restricted Isometry Property (RIP)

Definition 4.3 (Restricted isometry property (RIP) and restricted isometry constant (RIC))

A matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is said to satisfy the **RIP** with parameters (K, δ) , if for all $\mathcal{T} \subset [n]$ such that $|\mathcal{T}| \leq K$ and for all $\mathbf{q} \in \mathbb{R}^{|\mathcal{T}|}$, it holds that

$$(1 - \delta) \|\mathbf{q}\|_2^2 \leq \|\mathbf{A}_{\mathcal{T}} \mathbf{q}\|_2^2 \leq (1 + \delta) \|\mathbf{q}\|_2^2.$$

The **RIC** δ_K is defined as the smallest constant δ for which the K -RIP holds, i.e.,

$$\delta_K = \inf \left\{ \delta : (1 - \delta) \|\mathbf{q}\|_2^2 \leq \|\mathbf{A}_{\mathcal{T}} \mathbf{q}\|_2^2 \leq (1 + \delta) \|\mathbf{q}\|_2^2, \forall |\mathcal{T}| \leq K, \forall \mathbf{q} \in \mathbb{R}^{|\mathcal{T}|} \right\}.$$

RIP, Eigenvalues and Singular Values

Let $\mathbf{B} \in \mathbb{R}^{m \times K}$ be a tall matrix, i.e. $m \geq K$. Then the following statements are equivalent.

- ▶ For all $\mathbf{q} \in \mathbb{R}^K$,

$$(1 - \delta) \|\mathbf{q}\|_2^2 \leq \|\mathbf{B}\mathbf{q}\|_2^2 \leq (1 + \delta) \|\mathbf{q}\|_2^2.$$

- ▶
- ▶ $1 - \delta_K \leq \lambda_{\min}(\mathbf{B}^T \mathbf{B}) \leq \lambda_{\max}(\mathbf{B}^T \mathbf{B}) \leq 1 + \delta_K.$
- ▶ $\sqrt{1 - \delta_K} \leq \sigma_{\min}(\mathbf{B}) \leq \sigma_{\max}(\mathbf{B}) \leq \sqrt{1 + \delta_K}.$

RIP, Eigenvalues and Singular Values: Proof

- ▶ Let $\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^T$ be the compact SVD.



$$\begin{aligned}\|\mathbf{B}\mathbf{q}\|_2^2 &= \|\mathbf{U}\Sigma\mathbf{V}^T\mathbf{q}\|_2^2 = \mathbf{q}^T\mathbf{V}\Sigma\mathbf{U}^T\mathbf{U}\Sigma\mathbf{V}^T\mathbf{q} \\ &= \mathbf{q}^T\mathbf{V}\Sigma^2\mathbf{V}^T\mathbf{q} \\ &= \sum_{i=1}^K \sigma_i^2 c_i^2,\end{aligned}$$

where $c_i := \mathbf{v}_i^T \mathbf{q}$.

- ▶ $\sum_{i=1}^K c_i^2 = \|\mathbf{q}\|_2^2$. This follows from $\|\mathbf{V}^T\mathbf{q}\|_2^2 = \|\mathbf{q}\|_2^2$.



$$\sum_{i=1}^K \sigma_i^2 c_i^2 \leq \sigma_{\max}^2 \sum_{i=1}^K c_i^2 = \sigma_{\max}^2 \|\mathbf{q}\|_2^2,$$

$$\sum_{i=1}^K \sigma_i^2 c_i^2 \geq \sigma_{\min}^2 \sum_{i=1}^K c_i^2 = \sigma_{\min}^2 \|\mathbf{q}\|_2^2.$$

Monotonicity of RIC

Theorem 4.4

$\delta_1 \leq \delta_2 \leq \delta_3 \leq \dots (\delta_K \leq \delta_{K'} \text{ for all } K \leq K').$

Proof: Let $\mathcal{Q}_K = \{\mathbf{q} \in \mathbb{R}^n : \|\mathbf{q}\|_0 \leq K, \|\mathbf{q}\|_2 \leq 1\}$. It is clear that $\mathcal{Q}_K \subset \mathcal{Q}_{K'}$ if $K \leq K'$.

Then it holds that

$$\delta_K := \sup_{\mathbf{q} \in \mathcal{Q}_K} \left(\|\mathbf{A}\mathbf{q}\|_2^2 - 1 \right) \leq \sup_{\mathbf{q} \in \mathcal{Q}_{K'}} \left(\|\mathbf{A}\mathbf{q}\|_2^2 - 1 \right) =: \delta_{K'}.$$

Near Orthogonality of the Columns

Theorem 4.5

Let $\mathcal{I}, \mathcal{J} \subset [n]$ be two disjoint sets, i.e., $\mathcal{I} \cap \mathcal{J} = \emptyset$. For all $\mathbf{a} \in \mathbb{R}^{|\mathcal{I}|}$ and $\mathbf{b} \in \mathbb{R}^{|\mathcal{J}|}$,

$$|\langle \mathbf{A}_{\mathcal{I}} \mathbf{a}, \mathbf{A}_{\mathcal{J}} \mathbf{b} \rangle| \leq \delta_{|\mathcal{I}|+|\mathcal{J}|} \|\mathbf{a}\|_2 \|\mathbf{b}\|_2, \quad (8)$$

and

$$\|\mathbf{A}_{\mathcal{I}}^T \mathbf{A}_{\mathcal{J}} \mathbf{b}\|_2 \leq \delta_{|\mathcal{I}|+|\mathcal{J}|} \|\mathbf{b}\|_2. \quad (9)$$

Proof: From (8) to (9):

$$\begin{aligned} \|\mathbf{A}_{\mathcal{I}}^* \mathbf{A}_{\mathcal{J}} \mathbf{b}\|_2 &= \max_{\mathbf{q}: \|\mathbf{q}\|_2=1} |\langle \mathbf{q}, \mathbf{A}_{\mathcal{I}}^T \mathbf{A}_{\mathcal{J}} \mathbf{b} \rangle| = \max_{\mathbf{q}: \|\mathbf{q}\|_2=1} |\mathbf{q}^T \mathbf{A}_{\mathcal{I}}^T \mathbf{A}_{\mathcal{J}} \mathbf{b}| \\ &\leq \max_{\mathbf{q}: \|\mathbf{q}\|_2=1} \delta_{|\mathcal{I}|+|\mathcal{J}|} \|\mathbf{q}\|_2 \|\mathbf{b}\|_2 \\ &= \delta_{|\mathcal{I}|+|\mathcal{J}|} \|\mathbf{b}\|_2 \end{aligned}$$

Proof of (8)

(8) obviously holds when either \mathbf{a} or \mathbf{b} is zero. Assume $\mathbf{a} \neq \mathbf{0}$ and $\mathbf{b} \neq \mathbf{0}$. Define

$$\begin{aligned}\mathbf{a}' &= \mathbf{a} / \|\mathbf{a}\|_2, & \mathbf{b}' &= \mathbf{b} / \|\mathbf{b}\|_2, \\ \mathbf{x}' &= \mathbf{A}_{\mathcal{I}} \mathbf{a}', & \mathbf{y}' &= \mathbf{A}_{\mathcal{J}} \mathbf{b}'.\end{aligned}$$

Then RIP implies that

$$2(1 - \delta_{|\mathcal{I}|+|\mathcal{J}|}) \leq \|\mathbf{x}' + \mathbf{y}'\|_2^2 = \left\| [\mathbf{A}_{\mathcal{I}} \mathbf{A}_{\mathcal{J}}] \begin{bmatrix} \mathbf{a}' \\ \mathbf{b}' \end{bmatrix} \right\|_2^2 \leq 2(1 + \delta_{|\mathcal{I}|+|\mathcal{J}|}),$$

$$2(1 - \delta_{|\mathcal{I}|+|\mathcal{J}|}) \leq \|\mathbf{x}' - \mathbf{y}'\|_2^2 = \left\| [\mathbf{A}_{\mathcal{I}} \mathbf{A}_{\mathcal{J}}] \begin{bmatrix} \mathbf{a}' \\ -\mathbf{b}' \end{bmatrix} \right\|_2^2 \leq 2(1 + \delta_{|\mathcal{I}|+|\mathcal{J}|}).$$

Thus

$$\begin{aligned}\langle \mathbf{x}', \mathbf{y}' \rangle &= \frac{\|\mathbf{x}' + \mathbf{y}'\|_2^2 - \|\mathbf{x}' - \mathbf{y}'\|_2^2}{4} \leq \delta_{|\mathcal{I}|+|\mathcal{J}|} \\ -\langle \mathbf{x}', \mathbf{y}' \rangle &= \frac{\|\mathbf{x}' - \mathbf{y}'\|_2^2 - \|\mathbf{x}' + \mathbf{y}'\|_2^2}{4} \leq \delta_{|\mathcal{I}|+|\mathcal{J}|}\end{aligned}$$

Therefore,

$$\frac{|\langle \mathbf{A}_{\mathcal{I}} \mathbf{a}, \mathbf{A}_{\mathcal{J}} \mathbf{b} \rangle|}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2} = |\langle \mathbf{x}', \mathbf{y}' \rangle| \leq \delta_{|\mathcal{I}|+|\mathcal{J}|}.$$

Why RIP

In OMP, we need near-orthogonality between columns.

- $|\langle \mathbf{a}_i, \mathbf{a}_j \rangle|$ is small.

In other greedy algorithms, we need near-orthogonality between submatrices.

- $\|\mathbf{A}_{\mathcal{I}}^T \mathbf{A}_{\mathcal{J}} \mathbf{b}\|_2 \leq \delta_{|\mathcal{I}|+|\mathcal{J}|} \|\mathbf{b}\|_2$ means $\sigma_{\max}(\mathbf{A}_{\mathcal{I}}^T \mathbf{A}_{\mathcal{J}})$ is small.

Example: near-orthogonality of columns does not mean near-orthogonality of submatrices.

Suppose that $\mathbf{A}_{\mathcal{I}}^T \mathbf{A}_{\mathcal{J}} = \begin{bmatrix} \frac{1}{\ell} & \frac{1}{\ell} & \cdots & \frac{1}{\ell} \\ \frac{1}{\ell} & \frac{1}{\ell} & \cdots & \frac{1}{\ell} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\ell} & \frac{1}{\ell} & \cdots & \frac{1}{\ell} \end{bmatrix} \in \mathbb{R}^{\ell \times \ell}$.

Then $\sigma(\mathbf{A}_{\mathcal{I}}^T \mathbf{A}_{\mathcal{J}}) = 1, 0, \dots, 0$.

IHT Performance: A Sufficient Condition

Theorem 4.6

Suppose that \mathbf{A} satisfies the RIP with $\delta_{3S} < 1/\sqrt{32}$, then the k^{th} iteration of IHT obeys

$$\|\mathbf{x}_0 - \mathbf{x}^k\|_2 \leq 2^{-k} \|\mathbf{x}_0\|_2 + 5 \|\mathbf{w}\|_2.$$

Consequence: IHT estimates \mathbf{x} with accuracy

$$\|\mathbf{x}_0 - \mathbf{x}^k\|_2 \leq 6 \|\mathbf{w}\|_2, \quad \text{if } k > k^* = \left\lceil \log_2 \left(\frac{\|\mathbf{x}_0\|_2}{\|\mathbf{w}\|_2} \right) \right\rceil.$$

Optimality

Claim: No recovery method can perform fundamentally better.

Suppose that an oracle tells us the support \mathcal{T}_0 of \mathbf{x}_0 . Then

$$\hat{\mathbf{x}} = \begin{cases} (\mathbf{A}_{\mathcal{T}_0}^T \mathbf{A}_{\mathcal{T}_0})^{-1} \mathbf{A}_{\mathcal{T}_0}^T \mathbf{y} & \text{on } \mathcal{T}_0, \\ \mathbf{0} & \text{elsewhere.} \end{cases}$$

Thus, $\hat{\mathbf{x}} - \mathbf{x}_0 = \mathbf{0}$ on \mathcal{T}_0^c , while on \mathcal{T}_0

$$\hat{\mathbf{x}} - \mathbf{x}_0 = (\mathbf{A}_{\mathcal{T}_0}^T \mathbf{A}_{\mathcal{T}_0})^{-1} \mathbf{A}_{\mathcal{T}_0}^T \mathbf{w}.$$

By the RIP property,

$$\frac{1}{\sqrt{1 + \delta_S}} \|\mathbf{w}\|_2 \leq \|\hat{\mathbf{x}} - \mathbf{x}_0\|_2 \leq \frac{1}{\sqrt{1 - \delta_S}} \|\mathbf{w}\|_2.$$

Proof Idea

Let $\mathbf{r}^k := \mathbf{x}_0 - \mathbf{x}^k$ ($\mathbf{r}^0 = \mathbf{x}_0$). The key is to show that

$$\|\mathbf{r}^{k+1}\|_2 \leq \sqrt{8\delta_{3S}} \|\mathbf{r}^k\|_2 + 2\sqrt{1+\delta_S} \|\mathbf{w}\|_2.$$

In particular, if $\delta_{3S} < 1/\sqrt{32}$,

$$\|\mathbf{r}^{k+1}\|_2 \leq 0.5 \|\mathbf{r}^k\|_2 + 2.17 \|\mathbf{w}\|_2.$$

Back to the main result:

$$\begin{aligned}\|\mathbf{r}^k\|_2 &\leq \frac{1}{2} \|\mathbf{r}^{k-1}\|_2 + 2.17 \|\mathbf{w}\|_2 \\ &\leq \frac{1}{4} \|\mathbf{r}^{k-2}\|_2 + 2.17 \left(1 + \frac{1}{2}\right) \|\mathbf{w}\|_2 \\ &\cdots < \frac{1}{2^k} \|\mathbf{r}^0\|_2 + 4.34 \|\mathbf{w}\|_2.\end{aligned}$$

Detailed Proof

Recall that

$$\mathbf{x}^{k+1} = H_S \left(\mathbf{x}^k + \mathbf{A}^T \left(\mathbf{y} - \mathbf{A}\mathbf{x}^k \right) \right).$$

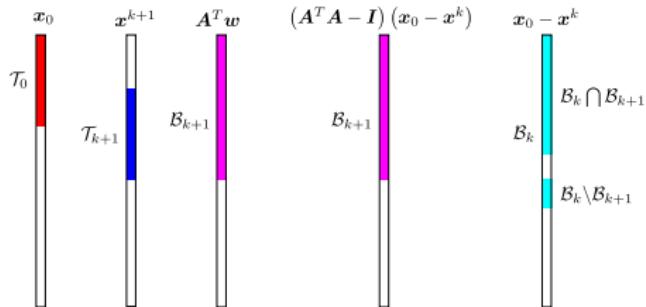
Define

$$\begin{aligned}\mathbf{a}^{k+1} &:= \mathbf{x}^k + \mathbf{A}^T \left(\mathbf{y} - \mathbf{A}\mathbf{x}^k \right) \\ &= \mathbf{x}_0 - \mathbf{x}_0 + \mathbf{x}^k + \mathbf{A}^T \left(\mathbf{A}\mathbf{x}_0 + \mathbf{w} - \mathbf{A}\mathbf{x}^k \right) \\ &= \mathbf{x}_0 + (\mathbf{A}^T \mathbf{A} - \mathbf{I}) \left(\mathbf{x}_0 - \mathbf{x}^k \right) + \mathbf{A}^T \mathbf{w} \\ &= \mathbf{x}_0 + (\mathbf{A}^T \mathbf{A} - \mathbf{I}) \mathbf{r}^k + \mathbf{A}^T \mathbf{w}. \end{aligned} \tag{10}$$

Then

$$\mathbf{x}^{k+1} = H_S \left(\mathbf{x}_0 + (\mathbf{A}^T \mathbf{A} - \mathbf{I}) \mathbf{r}^k + \mathbf{A}^T \mathbf{w} \right).$$

Detailed Proof (Continued)



$$\mathbf{x}^{k+1} = H_S \left(\mathbf{x}_0 + (\mathbf{A}^T \mathbf{A} - \mathbf{I}) \mathbf{r}^k + \mathbf{A}^T \mathbf{w} \right).$$

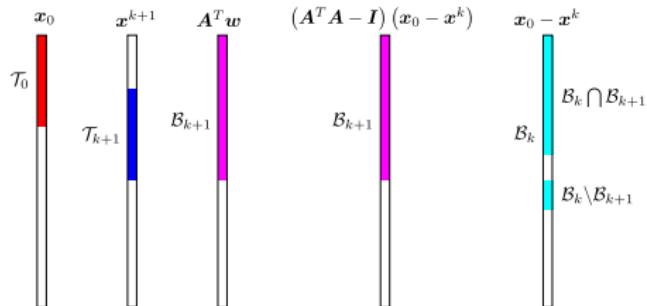
Let $\mathcal{T}_0 = \text{supp}(\mathbf{x}_0)$, $\mathcal{T}^k = \text{supp}(\mathbf{x}^k)$, and $\mathcal{B}^k = \mathcal{T}_0 \cup \mathcal{T}^k$.

- ▶ $\mathbf{r}^{k+1} = \mathbf{x}_0 - \mathbf{x}^{k+1}$ is supported on \mathcal{B}^{k+1}
- ▶ $\mathbf{r}^k = \mathbf{x}_0 - \mathbf{x}^k$ is supported on \mathcal{B}^k .

Want to show that $\|\mathbf{r}^{k+1}\|_2$ is small.

- ▶ Both $(\mathbf{A}^T \mathbf{A} - \mathbf{I}) \mathbf{r}^k$ and $\mathbf{A}^T \mathbf{w}$ are small.

Detailed Proof (Continued)



Focus on the set \mathcal{B}^{k+1} :

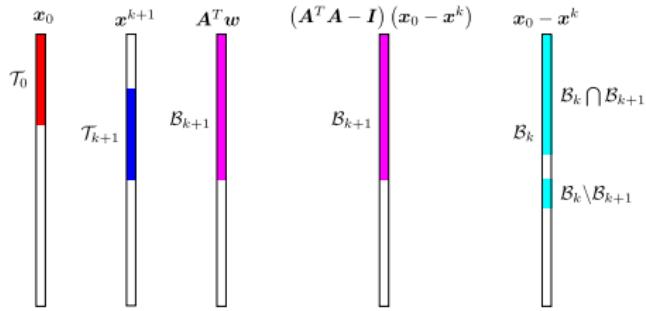
$$\begin{aligned}
 \|r^{k+1}\|_2 &= \left\| \mathbf{x}_{0, \mathcal{B}^{k+1}} - \mathbf{x}_{\mathcal{B}^{k+1}}^{k+1} \right\|_2 \\
 &= \left\| \mathbf{x}_{0, \mathcal{B}^{k+1}} - \mathbf{a}_{\mathcal{B}^{k+1}}^{k+1} + \mathbf{a}_{\mathcal{B}^{k+1}}^{k+1} - \mathbf{x}_{\mathcal{B}^{k+1}}^{k+1} \right\|_2 \\
 &\stackrel{(a)}{\leq} \left\| \mathbf{x}_{0, \mathcal{B}^{k+1}} - \mathbf{a}_{\mathcal{B}^{k+1}}^{k+1} \right\|_2 + \left\| \mathbf{a}_{\mathcal{B}^{k+1}}^{k+1} - \mathbf{x}_{\mathcal{B}^{k+1}}^{k+1} \right\|_2 \\
 &\stackrel{(b)}{\leq} 2 \left\| \mathbf{x}_{0, \mathcal{B}^{k+1}} - \mathbf{a}_{\mathcal{B}^{k+1}}^{k+1} \right\|_2,
 \end{aligned} \tag{11}$$

where

(a) has used triangle inequality, and

(b) follows from that $\mathbf{x}_{\mathcal{B}^{k+1}}^{k+1}$ is the best s -term approximation to $\mathbf{a}_{\mathcal{B}^{k+1}}^{k+1}$.

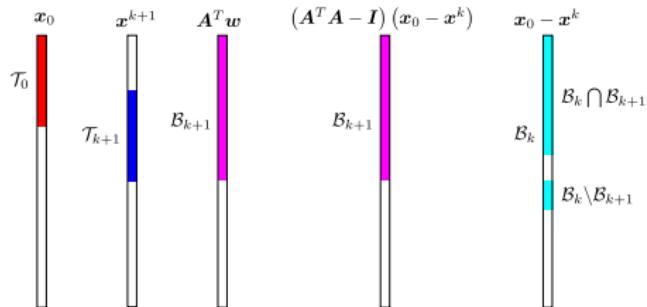
Detailed Proof (Continued)



The noise term: $A^T w$.

$$\left\| (A^T w)_{B^{k+1}} \right\|_2 = \left\| A_{B^{k+1}}^T w \right\|_2 \leq \sqrt{1 + \delta_{2S}} \|w\|_2.$$

Detailed Proof (Continued)



$$\begin{aligned}
& \left((\mathbf{I} - \mathbf{A}^T \mathbf{A}) \mathbf{r}^k \right)_{\mathcal{B}^{k+1}} = \mathbf{r}_{\mathcal{B}^{k+1}}^k - \mathbf{A}_{\mathcal{B}^{k+1}}^T \mathbf{A} \mathbf{r}^k \\
&= \mathbf{r}_{\mathcal{B}^{k+1}}^k - \mathbf{A}_{\mathcal{B}^{k+1}}^T \mathbf{A}_{\mathcal{B}^{k+1}} \cdot \mathbf{r}_{\mathcal{B}^{k+1}}^k - \mathbf{A}_{\mathcal{B}^{k+1}}^T \mathbf{A}_{\mathcal{B}^k \setminus \mathcal{B}^{k+1}} \cdot \mathbf{r}_{\mathcal{B}^k \setminus \mathcal{B}^{k+1}}^k \\
&= (\mathbf{I} - \mathbf{A}_{\mathcal{B}^{k+1}}^T \mathbf{A}_{\mathcal{B}^{k+1}}) \mathbf{r}_{\mathcal{B}^{k+1}}^k - \mathbf{A}_{\mathcal{B}^{k+1}}^T \mathbf{A}_{\mathcal{B}^k \setminus \mathcal{B}^{k+1}} \cdot \mathbf{r}_{\mathcal{B}^k \setminus \mathcal{B}^{k+1}}^k.
\end{aligned}$$

Hence

$$\|\cdots\|_2 \leq \delta_{2S} \left\| \mathbf{r}_{\mathcal{B}^{k+1}}^k \right\|_2 + \delta_{3S} \left\| \mathbf{r}_{\mathcal{B}^k \setminus \mathcal{B}^{k+1}}^k \right\|_2 \leq \sqrt{2} \delta_{3S} \left\| \mathbf{r}^k \right\|_2,$$

Detailed Proof (Continued)

where

- ▶ The 1st term follows from $|\mathcal{B}^{k+1}| \leq 2S$ and RIP.
- ▶ The 2nd term follows from Theorem 4.5.
- ▶ The last term uses $\delta_{2S} \leq \delta_{3S}$ (Theorem 4.4) and Cauchy-Schwartz Inequality

$$\begin{aligned}& \left\| \mathbf{r}_{\mathcal{B}^{k+1}}^k \right\|_2 + \left\| \mathbf{r}_{\mathcal{B}^k \setminus \mathcal{B}^{k+1}}^k \right\|_2 \\& \leq \sqrt{2} \left(\left\| \mathbf{r}_{\mathcal{B}^{k+1}}^k \right\|_2^2 + \left\| \mathbf{r}_{\mathcal{B}^k \setminus \mathcal{B}^{k+1}}^k \right\|_2^2 \right)^{1/2} \\& = \sqrt{2} \left\| \mathbf{r}_{\mathcal{B}^k \cup \mathcal{B}^{k+1}}^k \right\|_2 = \sqrt{2} \left\| \mathbf{r}^k \right\|_2.\end{aligned}$$

Finally,

$$\left\| \mathbf{r}^{k+1} \right\|_2 \leq 2 \left\| \mathbf{x}_{0, \mathcal{B}^{k+1}} - \mathbf{a}_{\mathcal{B}^{k+1}}^{k+1} \right\|_2 \leq \sqrt{8} \delta_{3S} \left\| \mathbf{r}^k \right\|_2 + \sqrt{1 + \delta_{3S}} \left\| \mathbf{w} \right\|_2.$$

ℓ_p -Norm

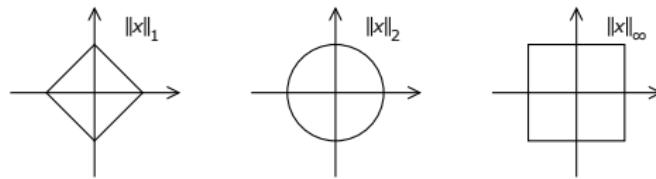
Definition 4.7 (ℓ_p -norm)

For a real number $p \geq 1$, the ℓ_p -norm of $\mathbf{x} \in \mathbb{R}^n$ is given by

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}.$$

Examples

- ▶ ℓ_1 -norm (Manhattan distance): $\|\mathbf{x}\|_1 = \sum |x_i|$.
- ▶ ℓ_2 -norm (Euclidean norm): $\|\mathbf{x}\| = \sqrt{\sum x_i^2}$.
- ▶ ℓ_∞ -norm: $\|\mathbf{x}\|_\infty = \max(|x_1|, \dots, |x_n|)$.



The Hölder's Inequality

Theorem 4.8 (The Hölder's inequality)

Let $p, q \in [1, \infty]$ with $1/p + 1/q = 1$.

For all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, it holds that

$$\begin{aligned} \sum_{i=1}^n |x_i \cdot y_i| &\leq \|\mathbf{x}\|_p \|\mathbf{y}\|_q \\ &= \left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \cdot \left(\sum_{i=1}^n |y_i|^q \right)^{1/q}. \end{aligned}$$

The equality holds iff $|x|^p$ and $|y|^q$ are linear dependent, i.e.,
 $\alpha |x_i|^p = \beta |y_i|^q, \forall i$.

(Proof is omitted.)

The Cauchy–Schwartz Inequality

Theorem 4.9 (The Cauchy–Schwartz Inequality)

A special case of the Hölder's inequality is when $p = q = 2$.

$$\sum_{i=1}^n |x_i \cdot y_i| \leq \|\mathbf{x}\|_2 \cdot \|\mathbf{y}\|_2.$$

In particular, for all $\mathbf{x} \in \mathbb{R}^n$,

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i| \leq \sqrt{n} \|\mathbf{x}\|_2,$$

where the equality holds iff $|x_i| = |x_j|$.