EE4.66 Topics in Large Dimensional Data Processing

Wei Dai

Imperial College London (ICL)

2019

Basic Information

Instructor: Dr. Wei Dai
GTA: Yifan Ran and Xi Yao

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 (10/10/2019-12/12/2019) Room 509B
Assessment: Exam (75%) and coursework (25%).

Section 1
Introduction

A Paradigm Shift

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

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

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

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).

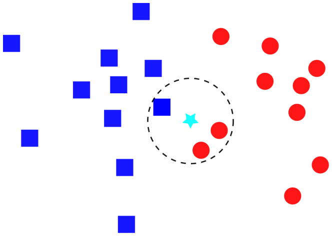
To address it: low dimensional structure.

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.

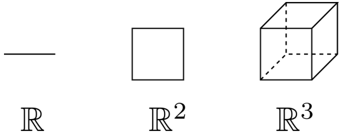


KNN with High Dimensional Data

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 $r > 0$:

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

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

$$\underbrace{[0,1]^n}_{\text{union space}} \subset \underbrace{\bigcup_{i=1}^{m} B_n\left(\boldsymbol{x}^{(i)}, 1\right)}_{\text{covered by unit balls}}.$$

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

$$m \ge \frac{\Gamma(n/2+1)}{\pi^{n/2}} \overset{n\to\infty}{\sim} \left(\frac{n}{2\pi e}\right)^{n/2} \sqrt{n\pi}.$$

k-NN: Isolated Data Points

Required number of data points for covering:

dimension

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

samples

Example 2: (False Structures) Empirical Covariance

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

$$\boldsymbol{\Sigma}_x := \mathrm{E}\left[\boldsymbol{X}\boldsymbol{X}^T\right].$$

Solution: the empirical covariance matrix

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

where $\tilde{\boldsymbol{X}} = \left[\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(m)}\right]$ is the sample matrix.

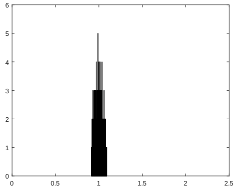
Rationale: By the *Law of Large Numbers*, if n is fixed and $m \to \infty$,

$$\hat{\boldsymbol{\Sigma}} \to \boldsymbol{I}.$$

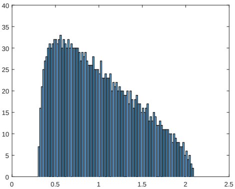
Empirical Covariance

Let $\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(m)} \in \mathbb{R}^n$ and $\boldsymbol{x}^{(i)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
Compute the eigenvalues of $\hat{\boldsymbol{\Sigma}}$ in (1).

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



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

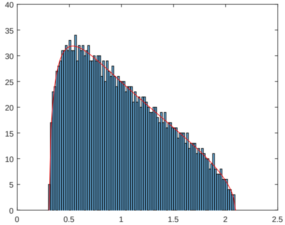


Asymptotic Behavior of Empirical Covariance

If $n, m \to \infty$ proportionally $(n/m \to r \in \mathbb{R}^+)$:
the distribution of eigenvalues of empirical covariance matrix $\hat{\boldsymbol{\Sigma}}$ 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 $(\boldsymbol{x}_i, y_i)$, $i = 1, \cdots, m$, want to estimate a linear function represented by $\boldsymbol{\alpha} \in \mathbb{R}^n$ s.t. $y_i \approx \langle \boldsymbol{x}_i, \boldsymbol{\alpha} \rangle$.

Solution: Let $\boldsymbol{y} = [y_i, \cdots, y_m]^T$, $\boldsymbol{X} = [\boldsymbol{x}_i, \cdots, \boldsymbol{x}_m]^T$, and $\boldsymbol{e} = [e_1, \cdots, e_m]$. Write

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

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

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

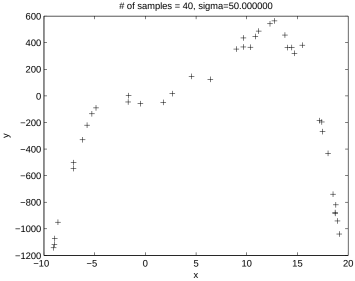
Linear Regression to Learn Nonlinear Function

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

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.

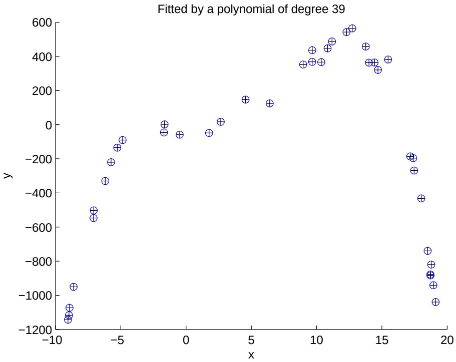
Since

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

one can find f by computing $\boldsymbol{\alpha}$ from

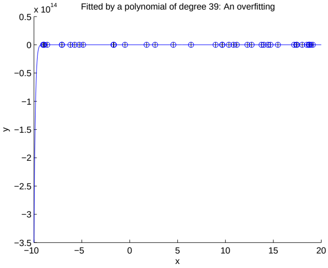
$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}_{\boldsymbol{y}} = \underbrace{\begin{bmatrix} 1 & x_1 & \cdots & x_1^{m-1} \\ 1 & x_2 & \cdots & x_2^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \cdots & x_m^{m-1} \end{bmatrix}}_{\boldsymbol{X}} \underbrace{\begin{bmatrix}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

A Solution that Looks Perfect

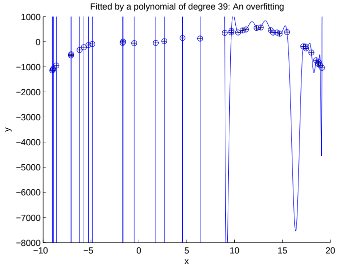


Bad News: Overfitting

Poor prediction performance



$f(x)$



$f(x)$ zoomed in

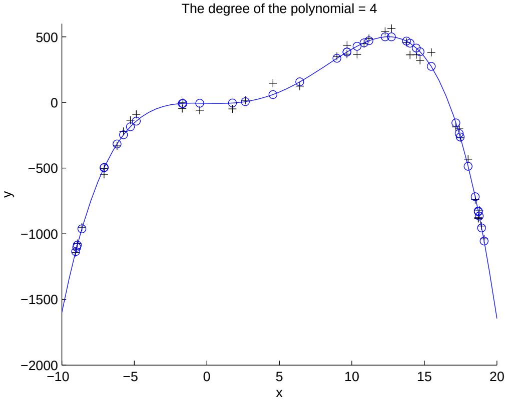
Note that

amplify noise

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

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

A Sparse Approximation



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

A More Realistic Example: Vector Input

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

$$\begin{aligned} f(\boldsymbol{x}) = & \alpha_0 + \underbrace{\alpha_1 x_1 + \cdots + \alpha_d x_d}_{d \text{ 1st-order terms}} \\ & + \alpha_{d+1} x_1^2 + \alpha_{d+2} x_1 x_2 + \cdots + \alpha_{2d} x_1 x_d \\ & + \alpha_{2d+1} x_2^2 + \cdots + \alpha_{n-2} x_{d-1} x_d \\ & + \alpha_{n-1} x_d^2. \end{aligned}$$

$(n-1-d)$ 2nd-order terms

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

To Find the Polynomial

$$\text{samples } (m)\underbrace{\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(m) \end{bmatrix}}_{\boldsymbol{y}} = \underbrace{\overbrace{\begin{bmatrix} 1 & x_1(1) & \cdots & x_d^2(1) \\ 1 & x_1(2) & \cdots & x_d^2(2) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1(m) & \cdots & x_d^2(m) \end{bmatrix}}^{\text{dimension } (n)}}_{\boldsymbol{X}} \underbrace{\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{bmatrix}}_{\boldsymbol{\alpha}}.$$

Two issues:

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

Sparsity plays a key role: Enforce most of the elements of $\boldsymbol{\alpha}$ to be zero.

Circumvent the curse of dimensionality: via transform
In most cases, the data have an intrinsic low-dimensional structure.

What is this Course About

- Programming ✗
- Computer architecture ✗
- 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.

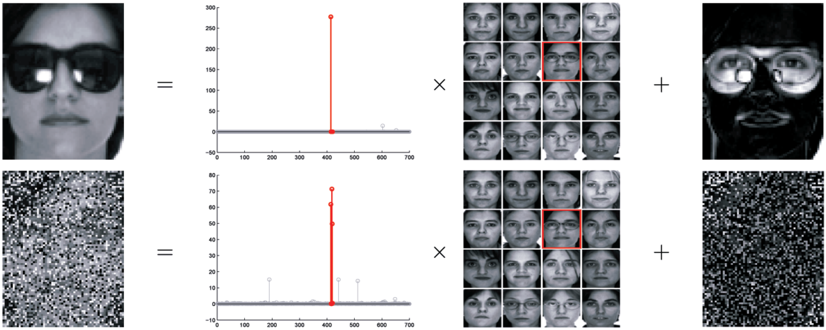
Image Denoising

Original

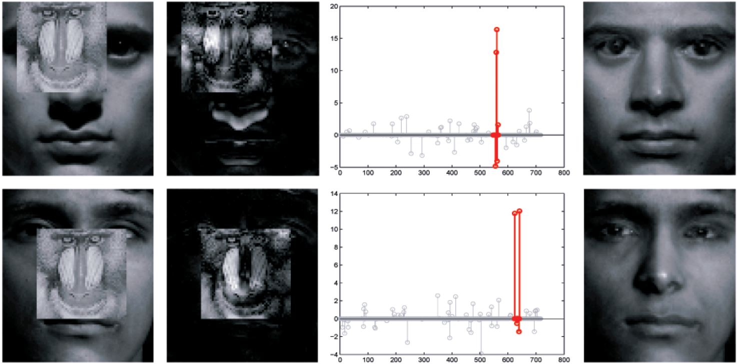
Noisy

Denoised

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



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



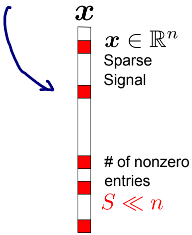
Netflix Problem

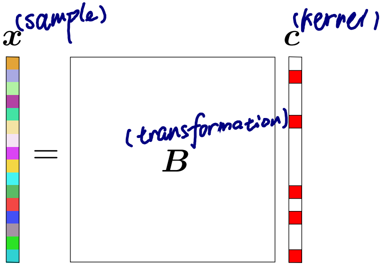
	Black Swan	Titanic	True Grit	The King's Speech
J. Cameron		★★★★★	★★★☆☆	
C. Eastwood	★★★★☆		★★★★★	
P. Jackson		★★★☆☆		★★★★☆
Roman Polanski	★★★★★			★★★★★

Section 2
Sparse Regression: Basics

Definition: Sparse Signals

- S-sparse

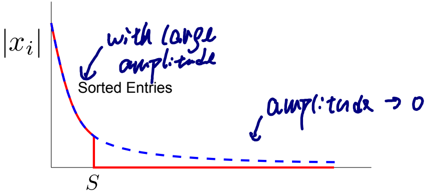




$\boldsymbol{B}$ is a given transform or dictionary.

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 $\boldsymbol{x}$ be a vector. Suppose that the entries of $\boldsymbol{x}$ obey a power law

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

with a given $r > 1$.

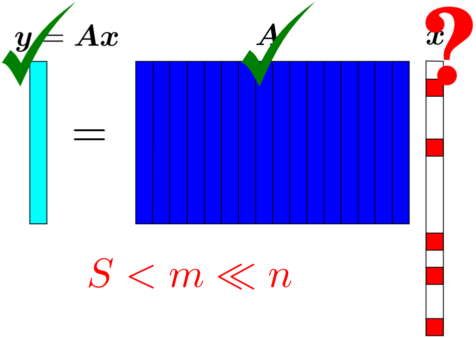
- Now use the leading S-term sub-vector to approximate $\boldsymbol{x}$, i.e.,

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

Then the best S-term approximation gives a distortion

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

The Sparse Regression Problem



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

- Machine learning.
- Compressed sensing.

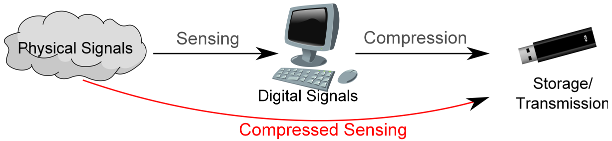
Compressed Sensing: Reducing the Number of Samples

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

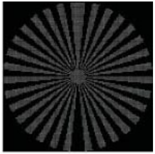
- Magnetic Resonance Imaging (MRI):

- Infrared sensing:

The Paradigm Shift: Compressed Sensing



An example in MRI

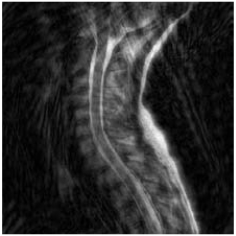


Trzasko, Manduca, Borisch (Mayo Clinic)

Sampling Pattern in Fourier domain

Fast Magnetic Resonance Imaging

Fully sampled



6 $\times$ undersampled classical

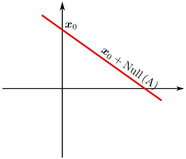
6 $\times$ undersampled CS

Trzasko, Manduca, Borisch (Mayo Clinic)

The Solutions of the Problem

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

- 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

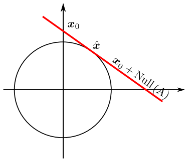
The least squared solution:

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

least squared solution.
· closed form
· not sparse (although min norm)

where $\boldsymbol{A}^{\dagger}$ is the pseudo-inverse.
Or equivalently, choose $\hat{\boldsymbol{x}}$ to be the solution of the optimization problem:

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_2, \quad \text{s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \tag{2}$$



- 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

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

To find a sparse solution:

Noise-free case (our focus):

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \text{ s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \tag{3}$$

Noisy case (will not be discussed in details):

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

λ: weight on sparsity

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

$\lambda \to 0$: enforce $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$.

$\lambda \to \infty$: 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 $\boldsymbol{e}_1 = [1, 0, \cdots, 0]^T$, $\boldsymbol{e}_2 = [0, 1, 0, \cdots, 0]^T$, $\cdots$.
  Then
  $$\|\boldsymbol{e}_1\|_0 = 1, \quad \text{but } \|\boldsymbol{e}_1 + \epsilon \boldsymbol{e}_2\|_0 = 2,$$
  no matter how small $\epsilon \neq 0$ is.
- Solving (3) min $\|x\|_0$ s.t $y = Ax$ usually means an exhaustive search.
  - Prohibitive complexity $O\left(n^S\right)$.

Definition: Support Set and Truncation

Definition 2.2

sparse index set

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

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

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

Truncation

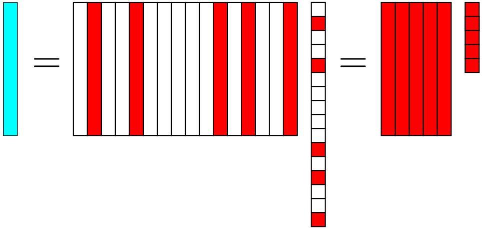
Definition 2.3

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

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

Example: Let $\mathcal{I} = \text{supp}(\boldsymbol{x})$. Then $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}_{\mathcal{I}}\boldsymbol{x}_{\mathcal{I}}$. !

$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ $\qquad\qquad$ $\boldsymbol{y} = \boldsymbol{A}_{\mathcal{I}}\boldsymbol{x}_{\mathcal{I}}$



Solving ℓ_0-Minimization: Exhaustive Search

$\min \|x\|_0 \quad \text{s.t. } y = Ax$

For $s = 1, 2, \cdots$

C_n^s

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

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

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

End

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

Computational Complexity

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

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

$$= \frac{n^{n+\frac{1}{2}} e^{-n}}{S^{\#}!\,(n-S^{\#})^{n-S^{\#}+\frac{1}{2}} e^{-n+S^{\#}}}$$

$$\approx n^{S^{\#}} \cdots$$

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

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

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

m: samples (observations)
n: dimension (unknowns)

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

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

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

Linear Independence and Dependence

Vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n \in \mathbb{R}^m$ are linearly independent if

$$\sum \lambda_i \boldsymbol{v}_i = \mathbf{0} \ \Rightarrow \ \lambda_i = 0, \ \forall i.$$

In matrix format,

$$\underset{m \times n}{[\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n]} \underset{n \times 1}{\boldsymbol{\lambda}} = \mathbf{0} \in \mathbb{R}^m \ \Rightarrow \ \boldsymbol{\lambda} = \mathbf{0} \in \mathbb{R}^n.$$

linearly independent

Vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n \in \mathbb{R}^m$ are linearly dependent if $\exists \boldsymbol{\lambda} \neq \mathbf{0}$ such that

$$[\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n] \boldsymbol{\lambda} = \mathbf{0}.$$

linearly dependent.

Rank and Matrix Inverse

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ 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 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is invertible if there exists a $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ such that

$$\boldsymbol{A}\boldsymbol{B} = \boldsymbol{B}\boldsymbol{A} = \boldsymbol{I}.$$

- For a matrix $\boldsymbol{A}$, its pseudoinverse $\boldsymbol{A}^\dagger$ is defined to satisfy
  - $\boldsymbol{A}\boldsymbol{A}^\dagger\boldsymbol{A} = \boldsymbol{A}$; $\boldsymbol{A}^\dagger\boldsymbol{A}\boldsymbol{A}^\dagger = \boldsymbol{A}^\dagger$.
  - $\left(\boldsymbol{A}\boldsymbol{A}^\dagger\right)^T = \boldsymbol{A}\boldsymbol{A}^\dagger$; $\left(\boldsymbol{A}^\dagger\boldsymbol{A}\right)^T = \boldsymbol{A}^\dagger\boldsymbol{A}$.

- $\boldsymbol{A}$ is invertible $\Leftrightarrow$ $\boldsymbol{A}$ is of full rank.

Examples

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

A =

     2     0
     0     4

B = inv(A)

B =

    0.5000         0
         0    0.2500
```

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

A =

     2     0
     0     0

B = inv(A)
[Warning: Matrix is singular to working precision.]

B =

   Inf   Inf
   Inf   Inf

C = pinv(A)

C =

    0.5000         0
         0         0
```

Eigen-decomposition

Definition 3.2 (Eigendecomposition, spectral decomposition)

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

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}, \tag{4}$$

where λ is called the eigenvalue corresponding to $\boldsymbol{v}$.

invertible = eigenvalues all nonzero

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

$A = Q\Lambda Q^{-1}$

$$\boldsymbol{A}^{-1} = \boldsymbol{Q}\boldsymbol{\Lambda}^{-1}\boldsymbol{Q}^{-1}, \tag{5}$$

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

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

An Example $A \in \mathbb{R}^{n\times n}$

$\Downarrow$

$Av = \lambda v$

$\Downarrow$

$A^{-1} = Q\Lambda^{-1}Q^{-1}$

```
A = [1 2; 1 3]

A =

     1     2
     1     3

[V,D] = eig(A)

V =

   -0.9391   -0.5907
    0.3437   -0.8069


D =

    0.2679         0
         0    3.7321
```

```
M1 = [A*V(:,1) - D(1,1)*V(:,1), A*V(:,2) - D(2,2)*V(:,2)]

M1 =

   1.0e-16 *

    0.5551         0
   -0.8327         0

M2 = V*inv(D)*inv(V)*A

M2 =

    1.0000    0.0000
    0.0000    1.0000
```

Homework (Examples)

$(5): A^{-1} = Q\Lambda^{-1}Q^{-1}$

$A^{-1}A = Q\Lambda^{-1}Q^{-1}Q\Lambda Q^{-1} = I$

$AA^{-1} = Q\Lambda Q^{-1}Q\Lambda^{-1}Q^{-1} = I$

$(4)\ Av = \lambda v$

$\Rightarrow (A-\lambda I)v = 0$

$\Rightarrow \boxed{|A-\lambda I| = 0}$

③ $(1-\lambda)(4-\lambda) - 4 = 0 \Rightarrow \lambda = 0, 5 \Rightarrow \Lambda = \begin{bmatrix} 5 & \\ & 0 \end{bmatrix}$

$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = 5\begin{bmatrix} a \\ b \end{bmatrix} \Rightarrow b = 2a \Rightarrow v = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \Rightarrow Q = \begin{bmatrix} 1 & 1 \\ 2 & -0.5 \end{bmatrix}$

$= 0\begin{bmatrix} a \\ b \end{bmatrix} \Rightarrow b = -\frac{a}{2} \Rightarrow v = \begin{bmatrix} 1 \\ -0.5 \end{bmatrix}$

A^{-1} does not exist

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

① $(1-\lambda)(2-\lambda) - 1 = 0 \Rightarrow 2 - 3\lambda + \lambda^2 - 1 = 0 \Rightarrow \lambda = \frac{3 \pm \sqrt{5}}{2} \Rightarrow \Lambda = \begin{bmatrix} \frac{3+\sqrt{5}}{2} & \\ & \frac{3-\sqrt{5}}{2} \end{bmatrix}$

$\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = \frac{3 \pm \sqrt{5}}{2}\begin{bmatrix} a \\ b \end{bmatrix} \Rightarrow b = \frac{1 \pm \sqrt{5}}{2} a \Rightarrow V = \begin{bmatrix} 1 \\ \frac{1 \pm \sqrt{5}}{2} \end{bmatrix}\ Q = \begin{bmatrix} 1 & 1 \\ \frac{1+\sqrt{5}}{2} & \frac{1-\sqrt{5}}{2} \end{bmatrix}$

$A^{-1} = \frac{1}{2-1}\begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}.\ A^{+} = A^{-1}$

② $(1-\lambda)^2 - 1 = 0 \Rightarrow \lambda = 0, 2 \Rightarrow \Lambda = \begin{bmatrix} 2 & \\ & 0 \end{bmatrix}$

$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = 2\begin{bmatrix} a \\ b \end{bmatrix} \Rightarrow b = a \Rightarrow v = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow Q = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

$= 0\begin{bmatrix} a \\ b \end{bmatrix} \Rightarrow b = -a \Rightarrow v = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

A^{-1} does not exist

$A^{+} = (A^{T}A)^{-1}A^{T}$

Singular Value Decomposition

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

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

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

$V^{-1} = V^*$

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

A convention is to list the singular values in descending order, that is, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m$ (assuming $m \leq n$). In this case, the diagonal matrix $\boldsymbol{\Sigma}$ 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
```

$m \times m$

```
S =

   14.2691         0
         0    0.6268
         0         0
         0         0
```

$m \times n$

```
V =

   -0.6414    0.7672
   -0.7672   -0.6414
```

$n \times n$

Compact SVD

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

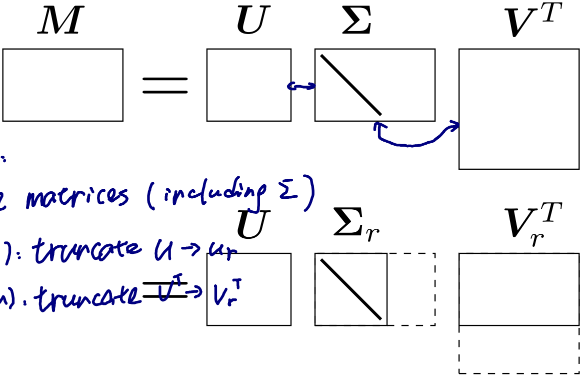
```
[U,S,V]=svd(A,0)

U =

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


S =

   14.2691         0
         0    0.6268


V =

   -0.6414    0.7672
   -0.7672   -0.6414
```

M x N

N x N

N x N

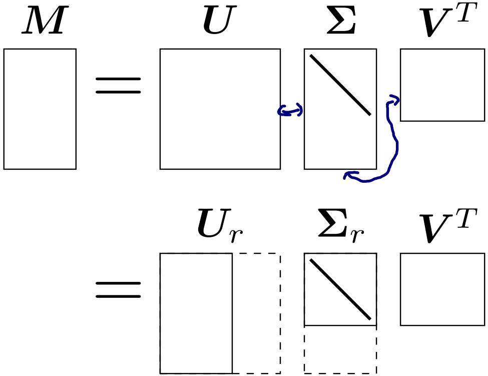
Compact SVD: Flat Matrices



Compact SVD:

- $2\times$ square matrices (including Σ)
- thin $(m>n)$: truncate $U \to U_r$
- flat $(m<n)$: truncate $V^T \to V_r^T$

Compact SVD: Tall Matrices



SVD and ED

Let

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

eigenvector

Then

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

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

$\Lambda = \Sigma^2$

Pseudoinverse and SVD

Consider the compact SVD

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

Then it holds that

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

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

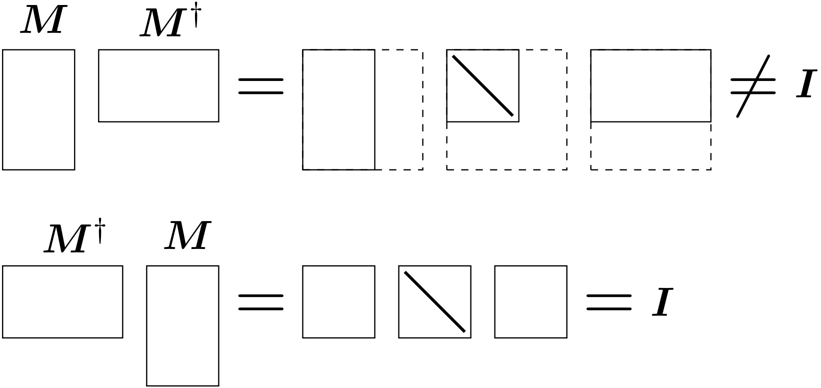
$AA^{\dagger} = U\Sigma V^T \cdot V\Sigma^{\dagger}U^T = U\Sigma\Sigma^{\dagger}U^T = U_rU_r^T$

$A^{\dagger}A = V\Sigma^{\dagger}U^T \cdot U\Sigma V^T = V\Sigma^{\dagger}\Sigma V^T = V_rV_r^T$

Pseudoinverse: An Illustration

$M = U\Sigma V^T$

$M^\dagger = V\Sigma^\dagger U^T$



Linear Subspace and Basis

Definition 3.3 (Linear subspace)

m samples in each dimension

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

n dimensions in each sample

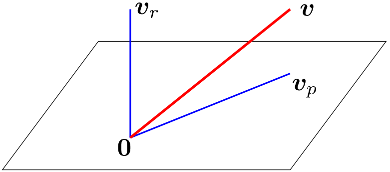
The linear span of $\mathcal{B}$ is defined as

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

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. (dimension)
- $\mathcal{B}$ is orthonormal if $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0$ for $i \neq j$ and $\langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle = 1$.
- For convenience, we use $\text{span}(\mathcal{B})$ and $\text{span}(\boldsymbol{B})$ interchangeably where $\boldsymbol{B} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n]$.

Projection



Definition 3.4 (Projection)

The projection of $\boldsymbol{x} \in \mathbb{R}^n$ onto the subspace $\mathrm{span}\,(\boldsymbol{A})$ is defined as

$$\boldsymbol{x}_p = \mathrm{proj}\,(\boldsymbol{x}, \boldsymbol{A}) = \boldsymbol{A}\boldsymbol{A}^\dagger \boldsymbol{x}.$$

$AA^+ = U_r U_r^T$

$U_r \in \mathbb{R}^n$

$\downarrow$ denote

$U \in \mathrm{span}(A)$

And the projection residue is given by

$$\boldsymbol{x}_r = \mathrm{resid}\,(\boldsymbol{x}, \boldsymbol{A}) = \boldsymbol{x} - \boldsymbol{x}_p.$$

Projection Viewed in SVD

Consider an n-d subspace in $\mathbb{R}^m$ with $m > n$.
Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ be its basis matrix. Clearly $\boldsymbol{A}$ is a tall matrix.
Consider the compact SVD $\boldsymbol{A} = \boldsymbol{U\Sigma V}^T$. Then

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

$m > n \Rightarrow U_n = U$

(first $r = n$ columns)

$aa^+ = aa^T$

And

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

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

Projection Residue Vector

$A^T x_r = A^T (x - AA^\dagger x)$
$= V\Sigma U^T (x - UU^T x)$
$= V\Sigma U^T x - V\Sigma U^T x = 0$

$AA^\dagger = U\Sigma V^T V \Sigma^\dagger U^T = U U_r^T$
$m > n \Rightarrow U_r = U$

Corollary 3.5

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

Proof:

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

.

Back to Linear Inverse Problem

$$\begin{cases} proj(x, A) = AA^{+}x = U_m U_m^T x \\ proj(x, A^T) = A^{+}Ax = V_m V_m^T x \end{cases}$$

Given

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

an estimation of $\boldsymbol{x}$ is given by $\hat{x} = A^{+}y = A^{+}(Ax + w)$
$= A^{+}Ax + A^{+}w$

$$A^{+}A = \begin{cases} I, & m \geq n \\ proj(x, A^T), & m < n \end{cases}$$

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

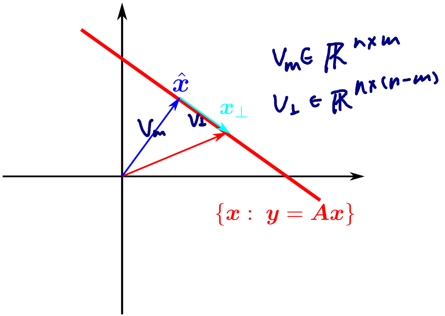
$= A^{+}Ax_{n \times 1}$, $n \times m \; m \times n$

- The case $m < n$:
  Consider the compact SVD of $\boldsymbol{A}$: $\boldsymbol{A} = \boldsymbol{U}\overset{\text{truncated } \Sigma}{\boldsymbol{\Lambda}}\boldsymbol{V}_m^T$. Clearly

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

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

A Geometric Picture



$$\mathcal{X} := \{\boldsymbol{x}:\ \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}\} = \operatorname{span}(\boldsymbol{V}_\perp) + \hat{\boldsymbol{x}},$$

where $\boldsymbol{V}_\perp \in \mathbb{R}^{n\times(n-m)}$ is the orthogonal complement of $\boldsymbol{V}_m$.

Link to the Least Squared Problem (2)

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

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

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

unitary

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

-

$$\|\boldsymbol{x}\|_2^2 = \langle \hat{\boldsymbol{x}} + \boldsymbol{x}_\perp, \hat{\boldsymbol{x}} + \boldsymbol{x}_\perp \rangle = \hat{\boldsymbol{x}}^T \hat{\boldsymbol{x}} + 2\hat{\boldsymbol{x}}^T \boldsymbol{x}_\perp + \boldsymbol{x}_\perp^T \boldsymbol{x}_\perp$$
$$= \|\hat{\boldsymbol{x}}\|_2^2 + \|\boldsymbol{x}_\perp\|_2^2 \geq \|\hat{\boldsymbol{x}}\|_2^2 .$$

Section 4
Greedy Algorithms

Greedy Algorithms: the Approach

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

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

sparsity vs consistency

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

- Otherwise if we roughly know the noise energy,

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

Major Greedy Algorithms

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

Intuition

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \quad \Rightarrow \quad \boldsymbol{A}^T\boldsymbol{y} = \underbrace{\boldsymbol{A}^T\boldsymbol{A}}_{I}\boldsymbol{x}.$$

Under the assumption that

- Columns of $\boldsymbol{A}$ are normalised.
- Columns of $\boldsymbol{A}$ are near orthogonal. → irrelevant features

$\boldsymbol{A}^T\boldsymbol{y}$ "looks like" $\boldsymbol{x}$.

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

Intuition: When $S = 1$

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

$$i^* = \arg\max_i \left| \boldsymbol{a}_i^T \boldsymbol{y} \right|$$

$A^T y \sim x$

$a_i^T y \sim x_i$

Once i^* is found,

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

i^*-column of A

Intuition: $S = 2$

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

projection of y to a_i

- Cancel the effect from i_1:

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

- Choose i_2 via

$$i_2 = \arg\max_{i} \ |\langle \boldsymbol{a}_i, \boldsymbol{y}_r \rangle| \,.$$

Remark: It holds that $i_2 \neq i_1$. We get two locations indeed.
Proof: Clearly $\boldsymbol{y}_r$ is orthogonal to $\boldsymbol{a}_{i_1}$, i.e. $\langle \boldsymbol{y}_r, \boldsymbol{a}_{i_1} \rangle = 0$.

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 $\mathcal{I}_2 = \{i_1, i_2\}$.

projection of y to $A_{\mathcal{I}_2}$

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

- Choose i_2 via

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

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

The Orthogonal Matching Pursuit (OMP) Algorithm

Input: S, $\boldsymbol{A}$, $\boldsymbol{y}$.
Initialization:
$\boldsymbol{x} = \boldsymbol{0}$, $\mathcal{T}^{\ell} = \phi$, and $\boldsymbol{y}_r = \boldsymbol{y}$.
Iteration: $\ell = 1, 2, \cdots, S$

1. Let $i_\ell = \arg \max_j \left|\langle \boldsymbol{a}_j, \boldsymbol{y}_r \rangle\right|$ pick max contribution
2. $\mathcal{T}^{\ell} = \mathcal{T}^{\ell-1} \bigcup \{i_\ell\}$. update index set (Add one index)
3. $\boldsymbol{x}_{\mathcal{T}^\ell} = \boldsymbol{A}^{\dagger}_{\mathcal{T}^\ell} \boldsymbol{y}$. estimate via sparse coefficients (Estimate ℓ-sparse signal)
4. $\boldsymbol{y}_r = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}$. check remaining parts (Compute estimation error)

Performance?

Suppose that

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

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

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

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

Performance Guarantee of OMP: Mutual Coherence

mutual coherence $\Leftrightarrow$ maximal correlation $\Leftrightarrow$ independency between features (dimensions)

Definition 4.1 (Mutual coherence)

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

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

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

Performance Guarantee of OMP

Theorem 4.2

Suppose that $\boldsymbol{A}$ satisfies that

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

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

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

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

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

The First Iteration of OMP (1)

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

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

$$\mu \triangleq \max_{i \neq j} |\langle a_i, a_j \rangle|$$

$$|\langle \boldsymbol{a}_i, \boldsymbol{y} \rangle| = \left| \sum_{j \in \mathcal{T}_0} x_{0,j} \langle \boldsymbol{a}_i, \boldsymbol{a}_j \rangle \right| \leq \sum_{j \in \mathcal{T}_0} |x_{0,j}| \, |\langle \underset{i \notin T_0}{\boldsymbol{a}_i}, \underset{j \in T_0}{\boldsymbol{a}_j} \rangle|$$

$$\|x\|_1 \leq \sqrt{S} \|x\|_2$$

$$\leq \mu \sum_{j \in \mathcal{T}_0} |x_{0,j}| \overset{(a)}{\leq} \mu \sqrt{S} \|\boldsymbol{x}\|_2$$

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

- Hence,

$$\max_{i \notin \mathcal{T}_0} |\langle \boldsymbol{a}_i, \boldsymbol{y} \rangle| \leq \mu \sqrt{S} \|\boldsymbol{x}\|_2 \,. \quad ① \tag{6}$$

The First Iteration of OMP (2)

$\|x\|_1 \le \sqrt{S}\,\|x\|_2$

$\max_{i \in T_0} |x_i| \ge \frac{1}{\sqrt{S}} \|x\|_2$

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

$$|\langle \boldsymbol{a}_i, \boldsymbol{y} \rangle| = \left| \sum_{j \in \mathcal{T}_0} x_{0,j} \langle \boldsymbol{a}_i, \boldsymbol{a}_j \rangle \right| \ge |x_{0,i} \langle \boldsymbol{a}_i, \boldsymbol{a}_i \rangle| - \left| \sum_{j \ne i} x_{0,j} \langle \boldsymbol{a}_i, \boldsymbol{a}_j \rangle \right|$$

$$\ge |x_{0,i}| - \mu \sum_{j \ne i} |x_{0,j}| \overset{(a)}{\ge} |x_{0,i}| - \mu \sqrt{S} \|\boldsymbol{x}\|_2 \, ,$$

$\max_{i \in T_0} |x_i| \ge \frac{1}{\sqrt{S}} \|x\|_2$

where (a) follows from Cauchy–Schwartz Inequality.

-

$$\max_{i \in \mathcal{T}_0} |\langle \boldsymbol{a}_i, \boldsymbol{y} \rangle| \ge \frac{1}{\sqrt{S}} \|\boldsymbol{x}\|_2 - \mu \sqrt{S} \|\boldsymbol{x}\|_2 \, ,$$ ②

where we have used the fact that

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

The First Iteration of OMP (3)

- Now suppose that $\mu < \overset{3}{\frac{1}{2S}}$ (the assumption in Theorem 4.2). Then
$$\frac{1}{\sqrt{S}} \|\boldsymbol{x}\|_2 > 2\mu\sqrt{S} \|\boldsymbol{x}\|_2 ,$$
- Or equivalently,
$$\underbrace{\max_{i \in \mathcal{T}_0} |\langle \boldsymbol{a}_i, \boldsymbol{y} \rangle| \geq \frac{1}{\sqrt{S}} \|\boldsymbol{x}\|_2 - \mu\sqrt{S} \|\boldsymbol{x}\|_2}_{2} > \underbrace{\mu\sqrt{S} \|\boldsymbol{x}\|_2 \geq \max_{i \notin \mathcal{T}_0} |\langle \boldsymbol{a}_i, \boldsymbol{y} \rangle|}_{1}.$$
- One concludes that
$$i_1 \in \mathcal{T}_0.$$

$\max\limits_{i \in T_0} |\langle a_i, y \rangle| \geq \max\limits_{i \notin T_0} |\langle a_i, y \rangle|$

The ℓ^{th} Iteration: Mathematical Induction

- Let $i_1, \cdots, i_{\ell-1}$ be the indices chosen in the first $\ell - 1$ iterations. Let $\mathcal{T}^{\ell-1} = \{i_1, \cdots, i_{\ell-1}\}$. Assume that $\mathcal{T}^{\ell-1} \subset \mathcal{T}_0$.
  to prove $T^{l} \in T_0$
- Then

$$\boldsymbol{y}_r = \boldsymbol{y} - \boldsymbol{A}_{\mathcal{T}^{\ell-1}} \boldsymbol{A}^{\dagger}_{\mathcal{T}^{\ell-1}} \boldsymbol{y} = \boldsymbol{y} - \boldsymbol{A}_{\mathcal{T}^{\ell-1}} \hat{\boldsymbol{x}}_{\ell-1} \in \operatorname{span}\left(\boldsymbol{A}_{\mathcal{T}_0}\right).$$

Or

$$\boldsymbol{y}_r = \boldsymbol{A}_{\mathcal{T}_0} \tilde{\boldsymbol{v}}_{\mathcal{T}_0}. \quad \begin{cases} i_l \in T_0 & \text{valid} \\ i_l \notin T^{l-1} & \text{unique} \\ |T^l| = l & \text{new} \end{cases}$$

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

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

Other Greedy Algorithm

OMP: One index is added per iteration.

$$\Downarrow$$

SP, CoSaMP, IHT: Multiple indices are updated per iteration.

Analysis:
Near orthogonality of all pairs of columns

$$\Downarrow$$

Near orthogonality of all disjoint sub-matrices.

Hard Thresholding Function

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

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

Example:

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

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

$$\operatorname{supp}(H_1(\boldsymbol{a})) = \arg\max_i |a_i|.$$
$$\operatorname{supp}(H_S(\boldsymbol{a})) = \{S \text{ indices of the largest magnitude entries in } \boldsymbol{a}\}.$$

In greedy algorithms:

$$\operatorname{supp}\left(H_1\left(\boldsymbol{A}^T\boldsymbol{y}\right)\right) = \arg\max_j |\langle \boldsymbol{y}, \boldsymbol{a}_j \rangle|.$$
$$\operatorname{supp}\left(H_S\left(\boldsymbol{A}^T\boldsymbol{y}\right)\right) = \{S \text{ indices corr. to the } S \text{ largest } |\langle \boldsymbol{y}, \boldsymbol{a}_j \rangle|\}.$$

The Subspace Pursuit (SP) Algorithm

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

Initialization:

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

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

previous S + residue S

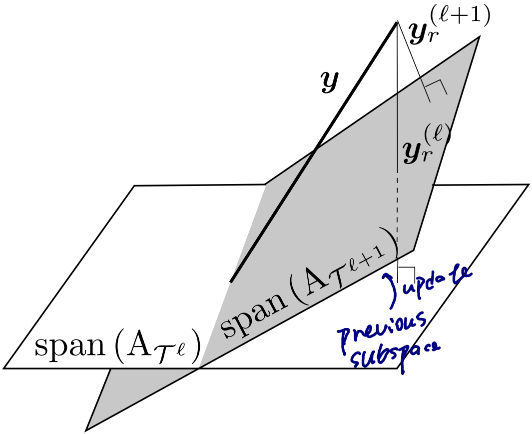
1. $\tilde{\mathcal{T}}^\ell = \mathcal{T}^{\ell-1} \bigcup \text{supp}\left(H_S\left(\boldsymbol{A}^T\boldsymbol{y}_r\right)\right)$. (Expand support)

   2S-sparse
2. Let $\boldsymbol{b}_{\tilde{\mathcal{T}}^\ell} = \boldsymbol{A}^\dagger_{\tilde{\mathcal{T}}^\ell}\boldsymbol{y}$ and $\boldsymbol{b}_{\left(\tilde{\mathcal{T}}^\ell\right)^c} = \boldsymbol{0}$. (Estimate $2S$-sparse signal)

   update sparse support: 2S → S
3. Set $\mathcal{T}^\ell = \text{supp}\left(H_S\left(\boldsymbol{b}\right)\right)$. (Shrink support)

   S-sparse
4. Let $\boldsymbol{x}^\ell_{\mathcal{T}^\ell} = \boldsymbol{A}^\dagger_{\mathcal{T}^\ell}\boldsymbol{y}$ and $\boldsymbol{x}^\ell_{\left(\mathcal{T}^\ell\right)^c} = \boldsymbol{0}$. (Estimate S-sparse signal)
5. Let $\boldsymbol{y}_r = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^\ell$. (Compute estimation error)

Geometric Interpretation



The Compressive Sampling Matching Pursuit (CoSaMP) Algorithm

CoSaMP: 3S-subspace

Input: S, $\boldsymbol{A}$, $\boldsymbol{y}$.
Initialization:
$\boldsymbol{x}^0 = \boldsymbol{0}$, and $\boldsymbol{y}_r = \boldsymbol{y}$.
Iteration: $\ell = 1, 2, \cdots$ until exit criterion true.

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

The Iterative Hard Thresholding (IHT) Algorithm

Input: S, $\boldsymbol{A}$, $\boldsymbol{y}$.
Initialization:
$\boldsymbol{x}^0 = \boldsymbol{0}$.
Iteration: $\ell = 1, 2, \cdots$ until exit criterion true.

$$\boldsymbol{x}^\ell = H_S\left(\boldsymbol{x}^{\ell-1} + \overbrace{\boldsymbol{A}^T\left(\underbrace{\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{\ell-1}}_{\text{y-residue}}\right)}^{\text{estimation error}}\right).$$

reform S candidates in each iteration.

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

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

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	$2S+1$	$\binom{n}{S} = O\left(n^S\right)$
OMP	$O\left(S^2 \log n\right)$	S
SP, CoSaMP, IHT	$O\left(S \cdot \log \frac{n}{S}\right)$	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 $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is said to satisfy the RIP with parameters (K, δ), if for all $\mathcal{T} \subset [n]$ such that $|\mathcal{T}| \le K$ and for all $\boldsymbol{q} \in \mathbb{R}^{|\mathcal{T}|}$, it holds that

$\mathcal{T}$: index set

$|\mathcal{T}|$: support

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

submatrix: randomly picked from A, with at most K columns.

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

preserve $\|q\|_2$ as long as q is at most K-sparse.

[1, 5]

minimum maximum

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

$(1, 5) \to [1+\varepsilon, 5-\varepsilon]$

infimum supremum

RIP, Eigenvalues and Singular Values

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

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

$$(1-\delta_K)\|\boldsymbol{q}\|_2^2 \leq \|\boldsymbol{Bq}\|_2^2 \leq (1+\delta_K)\|\boldsymbol{q}\|_2^2 .$$

-

$$1-\delta_K \leq \lambda_{\min}\left(\boldsymbol{B}^T\boldsymbol{B}\right) \leq \lambda_{\max}\left(\boldsymbol{B}^T\boldsymbol{B}\right) \leq 1+\delta_K .$$

-

$$\sqrt{1-\delta_K} \leq \sigma_{\min}(\boldsymbol{B}) \leq \sigma_{\max}(\boldsymbol{B}) \leq \sqrt{1+\delta_K} .$$

RIP, Eigenvalues and Singular Values: Proof

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

$$\|\boldsymbol{Bq}\|_2^2 = \left\|\boldsymbol{U\Sigma V}^T\boldsymbol{q}\right\|_2^2 = \boldsymbol{q}^T\boldsymbol{V\Sigma U}^T\boldsymbol{U\Sigma V}^T\boldsymbol{q}$$
$$= \boldsymbol{q}^T\boldsymbol{V\Sigma}^2\boldsymbol{V}^T\boldsymbol{q} = \boldsymbol{q}'^T\boldsymbol{\Sigma}^2\boldsymbol{q}' \quad \text{definition}$$
$$= \sum_{i=1}^{K} \sigma_i^2 q_i'^2,$$

$$\underbrace{\sigma_{\max}^2 \|q\|_2^2 \geq \|Bq\|_2^2}_{\text{SVD}} \geq (1-\delta)\|q\|_2^2$$
$$\sigma_{\min}^2 \|q\|_2^2 \leq \|Bq\|_2^2 \leq (1+\delta)\|q\|_2^2$$

where $\boldsymbol{q}' := \boldsymbol{V}^T\boldsymbol{q}$.

- It is clear that $\|\boldsymbol{q}'\|_2^2 = \boldsymbol{q}^T\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{q} = \|\boldsymbol{q}\|_2^2$. $(1+\delta)\|q\|_2^2 \geq \|Bq\|_2^2$ $= \sum_i \sigma_i^2 q_i'^2$

- $\sigma_{\max}^2 \|q\|_2^2 \geq \|Bq\|_2^2 \geq \sigma_{\min}^2 \|q\|_2^2$

$$\sum_{i=1}^{K} \sigma_i^2 q_i'^2 \leq \sigma_{\max}^2 \sum_{i=1}^{K} q_i'^2 = \sigma_{\max}^2 \|\boldsymbol{q}\|_2^2, \quad \geq \sigma_{\min}^2 \|q\|_2^2$$
$$\geq \sigma_{\min}^2 \sum_{i=1}^{K} q_i'^2 = \sigma_{\min}^2 \|\boldsymbol{q}\|_2^2.$$

Monotonicity of RIC

Theorem 4.4

RIC is a monotonically increasing sequence.

$\delta_1 \le \delta_2 \le \delta_3 \le \cdots$ ($\delta_K \le \delta_{K'}$ for all $K \le K'$).

Q_K: a set composed of q with no more than k nonzero elements.

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

$Q_{K'}$ has larger DoF

Then it holds that

$$1+\delta_K = \sup_{q \in Q_K} \|Aq\|_2^2 \le \sup_{q \in Q_{K'}} \|Aq\|_2^2 = 1+\delta_{K'}$$

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

Near Orthogonality of the Columns

Theorem 4.5

no overlapping

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

↳ any vectors from $A_\mathcal{I}$ $A_\mathcal{J}$ are orthogonal.

$$\left|\langle \boldsymbol{A}_\mathcal{I}\boldsymbol{a}, \boldsymbol{A}_\mathcal{J}\boldsymbol{b}\rangle\right| \le \delta_{|\mathcal{I}|+|\mathcal{J}|} \left\|\boldsymbol{a}\right\|_2 \left\|\boldsymbol{b}\right\|_2, \tag{8}$$

$$\frac{|\langle A_\mathcal{I} a, A_\mathcal{J} b\rangle|}{\|a\|_2 \|b\|_2} \le \delta_{|\mathcal{I}|+|\mathcal{J}|}$$

and

$$\left\|\boldsymbol{A}_\mathcal{I}^T \boldsymbol{A}_\mathcal{J}\boldsymbol{b}\right\|_2 \le \delta_{|\mathcal{I}|+|\mathcal{J}|} \left\|\boldsymbol{b}\right\|_2. \tag{9}$$

$$\frac{\|A_\mathcal{I}^T A_\mathcal{J} b\|_2}{\|b\|_2} \le \delta_{|\mathcal{I}|+|\mathcal{J}|}$$

↳ columns of $A_\mathcal{I}$, $A_\mathcal{J}$ are nearly orthogonal (small correlation between submatrices)

Proof: From (8) to (9):

(ℓ_2-norm : maximum inner product)

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

Proof of (8)

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

$$\boldsymbol{a}' = \boldsymbol{a} / \left\|\boldsymbol{a}\right\|_2, \quad \boldsymbol{b}' = \boldsymbol{b} / \left\|\boldsymbol{b}\right\|_2,$$
$$\boldsymbol{x}' = \boldsymbol{A}_{\mathcal{I}}\boldsymbol{a}', \qquad \boldsymbol{y}' = \boldsymbol{A}_{\mathcal{J}}\boldsymbol{b}'.$$

Then RIP implies that

$(1-\delta_{|\mathcal{I}|}) + (1-\delta_{|\mathcal{J}|})$ $\quad$ a_1 $\quad$ $(1+\delta_{|\mathcal{I}|}) + (1+\delta_{|\mathcal{J}|})$

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

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

a_2

Thus

max $\quad$ min

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

Therefore,

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

small correlation between submatrices.

Why RIP

In OMP, we need near-orthogonality between columns.

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

stronger (but require less)

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

$\sup_{b} \|A_{\mathcal{I}}^T A_{\mathcal{J}} b\|_2 = \sigma_{\max}(A_{\mathcal{I}}^T A_{\mathcal{J}})$: align b with dominant singular vector of $A_{\mathcal{I}}^T A_{\mathcal{J}}$.

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

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

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

near-orthogonality between columns

Then $\sigma\left(\boldsymbol{A}_{\mathcal{I}}^T \boldsymbol{A}_{\mathcal{J}} \right) = 1, \underbrace{0, \cdots, 0}$.

strong correlation between submatrices.

IHT Performance: A Sufficient Condition

Theorem 4.6

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

$$\left\| \boldsymbol{x}_0 - \boldsymbol{x}^k \right\|_2 \leq 2^{-k} \left\| \boldsymbol{x}_0 \right\|_2 + 5 \underbrace{\left\| \boldsymbol{w} \right\|_2}_{\text{noise}}.$$

Consequence: IHT estimates $\boldsymbol{x}$ with accuracy

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

Optimality

Claim: No recovery method can perform fundamentally better. (in the order of magnitude)

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

$$\hat{\boldsymbol{x}} = \begin{cases} \underbrace{\left(\boldsymbol{A}_{\mathcal{T}_0}^T \boldsymbol{A}_{\mathcal{T}_0}\right)^{-1} \boldsymbol{A}_{\mathcal{T}_0}^T}_{A_{T_0}^{\dagger}} \boldsymbol{y} & \text{on } \mathcal{T}_0, \\ \boldsymbol{0} & \text{elsewhere.} \end{cases}$$

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

Even if we know the support we can't perform better than IHT.

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

By the RIP property,

constant

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

Proof Idea

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

$$\left\|\boldsymbol{r}^{k+1}\right\|_2 \le \sqrt{8}\delta_{3S}\left\|\boldsymbol{r}^k\right\|_2 + 2\sqrt{1+\delta_S}\left\|\boldsymbol{w}\right\|_2.$$

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

$$\left\|\boldsymbol{r}^{k+1}\right\|_2 \le 0.5\left\|\boldsymbol{r}^k\right\|_2 + 2.17\left\|\boldsymbol{w}\right\|_2.$$

Back to the main result:

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

Detailed Proof

Recall that

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

Define

diagonal = 0
off-diagonal = tiny
only 2S columns matter.

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

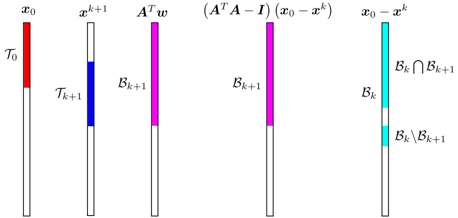
at most 2S nonzero entries

S · 2S

Then

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

Detailed Proof (Continued)



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

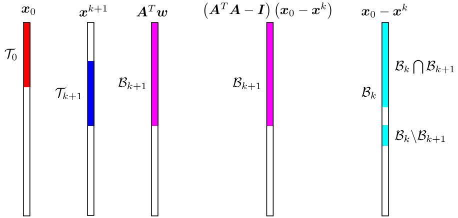
Let $\mathcal{T}_0 = \mathrm{supp}\left(\boldsymbol{x}_0\right)$, $\mathcal{T}^k = \mathrm{supp}\left(\boldsymbol{x}^k\right)$, and $\mathcal{B}^k = \mathcal{T}_0 \bigcup \mathcal{T}^k$.

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

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

- Both $\left(\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{I}\right)\boldsymbol{r}^k$ and $\boldsymbol{A}^T\boldsymbol{w}$ are small.

Detailed Proof (Continued)

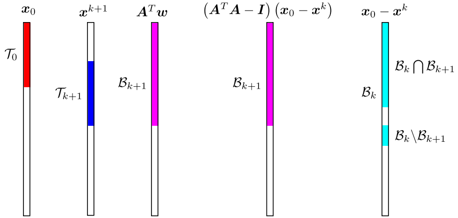


Focus on the set $\mathcal{B}^{k+1}$:

$$\begin{aligned}
\left\|\boldsymbol{r}^{k+1}\right\|_2 &= \left\|\boldsymbol{x}_{0,\mathcal{B}^{k+1}} - \boldsymbol{x}^{k+1}_{\mathcal{B}^{k+1}}\right\|_2 \\
&= \left\|\boldsymbol{x}_{0,\mathcal{B}^{k+1}} - \boldsymbol{a}^{k+1}_{\mathcal{B}^{k+1}} + \boldsymbol{a}^{k+1}_{\mathcal{B}^{k+1}} - \boldsymbol{x}^{k+1}_{\mathcal{B}^{k+1}}\right\|_2 \\
&\overset{(a)}{\le} \left\|\boldsymbol{x}_{0,\mathcal{B}^{k+1}} - \boldsymbol{a}^{k+1}_{\mathcal{B}^{k+1}}\right\|_2 + \left\|\boldsymbol{a}^{k+1}_{\mathcal{B}^{k+1}} - \boldsymbol{x}^{k+1}_{\mathcal{B}^{k+1}}\right\|_2 \\
&\overset{(b)}{\le} 2\left\|\boldsymbol{x}_{0,\mathcal{B}^{k+1}} - \boldsymbol{a}^{k+1}_{\mathcal{B}^{k+1}}\right\|_2, \qquad (11)
\end{aligned}$$

where
(a) has used triangle inequality, and
(b) follows from that $\boldsymbol{x}^{k+1}_{\mathcal{B}^{k+1}}$ is the best s-term approximation to $\boldsymbol{a}^{k+1}_{\mathcal{B}^{k+1}}$.

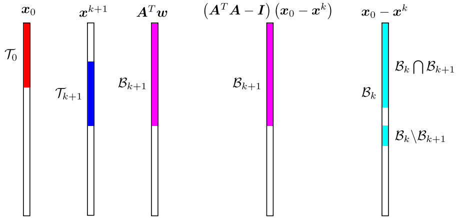
Detailed Proof (Continued)



The noise term: $\boldsymbol{A}^T\boldsymbol{w}$.

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

Detailed Proof (Continued)



$$\left(\left(\boldsymbol{I}-\boldsymbol{A}^T\boldsymbol{A}\right)\boldsymbol{r}^k\right)_{\mathcal{B}^{k+1}} = \boldsymbol{r}^k_{\mathcal{B}^{k+1}} - \boldsymbol{A}^T_{\mathcal{B}^{k+1}}\boldsymbol{A}\boldsymbol{r}^k$$

$$= \boldsymbol{r}^k_{\mathcal{B}^{k+1}} - \boldsymbol{A}^T_{\mathcal{B}^{k+1}}\boldsymbol{A}_{\mathcal{B}^{k+1}}\cdot\boldsymbol{r}^k_{\mathcal{B}^{k+1}} - \boldsymbol{A}^T_{\mathcal{B}^{k+1}}\boldsymbol{A}_{\mathcal{B}^k\backslash\mathcal{B}^{k+1}}\cdot\boldsymbol{r}^k_{\mathcal{B}^k\backslash\mathcal{B}^{k+1}}$$

$$= \left(\boldsymbol{I}-\boldsymbol{A}^T_{\mathcal{B}^{k+1}}\boldsymbol{A}_{\mathcal{B}^{k+1}}\right)\boldsymbol{r}^k_{\mathcal{B}^{k+1}} - \boldsymbol{A}^T_{\mathcal{B}^{k+1}}\boldsymbol{A}_{\mathcal{B}^k\backslash\mathcal{B}^{k+1}}\cdot\boldsymbol{r}^k_{\mathcal{B}^k\backslash\mathcal{B}^{k+1}}.$$

Hence

$$\left\|\cdots\right\|_2 \le \delta_{2S}\left\|\boldsymbol{r}^k_{\mathcal{B}^{k+1}}\right\|_2 + \delta_{3S}\left\|\boldsymbol{r}^k_{\mathcal{B}^k\backslash\mathcal{B}^{k+1}}\right\|_2 \le \sqrt{2}\delta_{3S}\left\|\boldsymbol{r}^k\right\|_2,$$

Detailed Proof (Continued)

where

- The 1st term follows from $\left|\mathcal{B}^{k+1}\right| \leq 2S$ 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

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

Finally,

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

ℓ_p-Norm

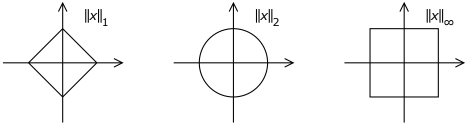
Definition 4.7 (ℓ_p-norm)

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

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

Examples

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



The Hölder's Inequality

Theorem 4.8 (The Hölder's inequality)

$p=1 \quad q=\infty$

Let $p, q \in [1, \infty]$ *with* $1/p + 1/q = 1$. $\sum_i |x_i y_i| \le \sum_i |x_i| \cdot \max(|y_1| \cdots |y_n|)$
For all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, *it holds that*

$$\|x\|_{p^*} = \sup_{y \neq 0} \frac{y^T x}{\|x\|_p} = \|y\|_q$$

$$\sum_{i=1}^{n} |x_i \cdot y_i| \le \|\boldsymbol{x}\|_p \|\boldsymbol{y}\|_q$$

$$= \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \cdot \left(\sum_{i=1}^{n} |y_i|^q \right)^{1/q}.$$

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 $p = q = 2$.

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

$y_i = 1$
$\|y\|_2 = \sqrt{n}$

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

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

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