Bilinear Inverse Problems: Theory, Algorithms, and Applications in Imaging Science and Signal Processing

Shuyang Ling

Department of Mathematics, UC Davis

May 31, 2017

Acknowledgements

Research in collaboration with:

- Prof.Xiaodong Li (UC Davis)
- Prof.Thomas Strohmer (UC Davis)
- Dr.Ke Wei (UC Davis)

This work is sponsored by NSF-DMS and DARPA.

Outline

(a) Part I: self-calibration and biconvex compressive sensing

- Application in array signal processing
- SparseLift: a convex approach towards biconvex compressive sensing

(b) Part II: blind deconvolution

- Applications in image deblurring and wireless communication
- Mathematical models and convex approach
- A nonconvex optimization approach towards blind deconvolution
- Extended to joint blind deconvolution and blind demixing

Part I

Part I: self-calibration and biconvex compressive sensing

Linear inverse problem

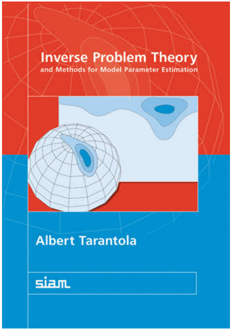
Inverse problem: to infer the values or parameters that characterize/describe the system from the obversations.

Many inverse problems involve solving a linear system:

$$y = \underbrace{A}_{\text{perfectly known}} \underbrace{x}_{\text{signal of interests}} + w.$$

Find x when y and A are given:

- A is overdetermined $\Longrightarrow$ linear least squares
- A is underdetermined: we need regularization, e.g., Tikhonov regularization and ℓ_1 regularization (sparsity and compressive sensing)



Linear inverse problem

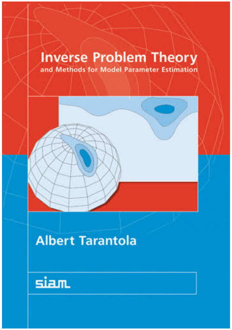
Inverse problem: to infer the values or parameters that characterize/describe the system from the obversations.

Many inverse problems involve solving a linear system:

$$\boldsymbol{y} = \underbrace{\boldsymbol{A}}_{\text{perfectly known}} \underbrace{\boldsymbol{x}}_{\text{signal of interests}} + \boldsymbol{w}.$$

Find $\boldsymbol{x}$ when $\boldsymbol{y}$ and $\boldsymbol{A}$ are given:

- $\boldsymbol{A}$ is overdetermined $\Longrightarrow$ linear least squares
- $\boldsymbol{A}$ is underdetermined: we need regularization, e.g., Tikhonov regularization and ℓ_1 regularization (sparsity and compressive sensing)



Linear inverse problem

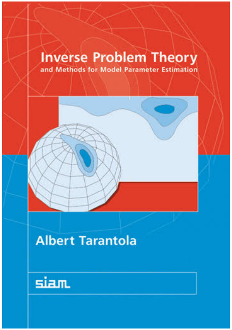
Inverse problem: to infer the values or parameters that characterize/describe the system from the obversations.

Many inverse problems involve solving a linear system:

$$\boldsymbol{y} = \underbrace{\boldsymbol{A}}_{\text{perfectly known}} \underbrace{\boldsymbol{x}}_{\text{signal of interests}} + \boldsymbol{w}.$$

Find $\boldsymbol{x}$ when $\boldsymbol{y}$ and $\boldsymbol{A}$ are given:

- $\boldsymbol{A}$ is overdetermined $\Longrightarrow$ linear least squares
- $\boldsymbol{A}$ is underdetermined: we need regularization, e.g., Tikhonov regularization and ℓ_1 regularization (sparsity and compressive sensing)



Calibration

However, the sensing matrix $\boldsymbol{A}$ may not be perfectly known.

Calibration issue:

- Calibration is to adjust one device with the standard one.
- Why? To reduce or eliminate bias and inaccuracy.
- Difficult or even impossible to calibrate high-performance hardware.
- Self-calibration: Equip sensors with a smart algorithm which takes care of calibration automatically.

Calibration

However, the sensing matrix $\boldsymbol{A}$ may not be perfectly known.

Calibration issue:

- Calibration is to adjust one device with the standard one.
- Why? To reduce or eliminate bias and inaccuracy.
- Difficult or even impossible to calibrate high-performance hardware.
- Self-calibration: Equip sensors with a smart algorithm which takes care of calibration automatically.

However, the sensing matrix $\boldsymbol{A}$ may not be perfectly known.

Calibration issue:

- Calibration is to adjust one device with the standard one.
- Why? To reduce or eliminate bias and inaccuracy.
- Difficult or even impossible to calibrate high-performance hardware.
- Self-calibration: Equip sensors with a smart algorithm which takes care of calibration automatically.

Calibration realized by machine?

Uncalibrated devices leads to imperfect sensing

We encounter imperfect sensing all the time: the sensing matrix $\boldsymbol{A}(\boldsymbol{h})$ depending on an unknown calibration parameter $\boldsymbol{h}$,

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{h})\boldsymbol{x} + \boldsymbol{w}.$$

This is too general to solve for $\boldsymbol{h}$ and $\boldsymbol{x}$ jointly.

Examples:

- Phase retrieval problem: $\boldsymbol{h}$ is the unknown phase of the Fourier transform of $\boldsymbol{x}$.
- Cryo-electron microscopy images: $\boldsymbol{h}$ can be the unknown orientation of a protein molecule and $\boldsymbol{x}$ is the particle.

Calibration realized by machine?

Uncalibrated devices leads to imperfect sensing

We encounter imperfect sensing all the time: the sensing matrix $\boldsymbol{A}(\boldsymbol{h})$ depending on an unknown calibration parameter $\boldsymbol{h}$,

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{h})\boldsymbol{x} + \boldsymbol{w}.$$

This is too general to solve for $\boldsymbol{h}$ and $\boldsymbol{x}$ jointly.

Examples:

- Phase retrieval problem: $\boldsymbol{h}$ is the unknown phase of the Fourier transform of $\boldsymbol{x}$.
- Cryo-electron microscopy images: $\boldsymbol{h}$ can be the unknown orientation of a protein molecule and $\boldsymbol{x}$ is the particle.

A simplified but important model

Our focus:

One special case is to assume $\boldsymbol{A}(\boldsymbol{h})$ to be of the form

$$\boldsymbol{A}(\boldsymbol{h}) = \boldsymbol{D}(\boldsymbol{h})\boldsymbol{A}$$

where $\boldsymbol{D}(\boldsymbol{h})$ is an unknown diagonal matrix.

However, this seemingly simple model is very useful and mathematically nontrivial to analyze.

- Phase and gain calibration in array signal processing
- Blind deconvolution

A simplified but important model

Our focus:

One special case is to assume $\boldsymbol{A}(\boldsymbol{h})$ to be of the form

$$\boldsymbol{A}(\boldsymbol{h}) = \boldsymbol{D}(\boldsymbol{h})\boldsymbol{A}$$

where $\boldsymbol{D}(\boldsymbol{h})$ is an unknown diagonal matrix.

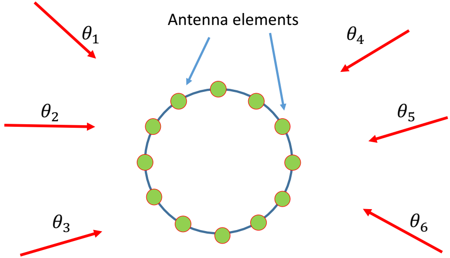
However, this seemingly simple model is very useful and mathematically nontrivial to analyze.

- Phase and gain calibration in array signal processing
- Blind deconvolution

Self-calibration in array signal processing

Calibration in the DOA (direction of arrival estimation)

One calibration issue comes from the unknown gains of the antennae caused by temperature or humidity.



Consider s signals impinging on an array of L antennae.

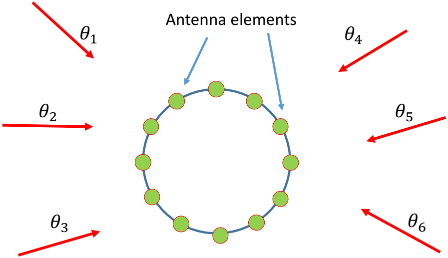
$$\boldsymbol{y} = \sum_{k=1}^{s} \boldsymbol{D}\boldsymbol{A}(\bar{\theta}_k)x_k + \boldsymbol{w}$$

where $\boldsymbol{D}$ is an unknown diagonal matrix and d_{ii} is the unknown gain for i-th sensor. $\boldsymbol{A}(\theta)$: array manifold. $\bar{\theta}_k$: unknown direction of arrival. $\{x_k\}_{k=1}^{s}$ are the impinging signals.

Self-calibration in array signal processing

Calibration in the DOA (direction of arrival estimation)

One calibration issue comes from the unknown gains of the antennae caused by temperature or humidity.



Consider s signals impinging on an array of L antennae.

$$\boldsymbol{y} = \sum_{k=1}^{s} \boldsymbol{D}\boldsymbol{A}(\bar{\theta}_k)x_k + \boldsymbol{w}$$

where $\boldsymbol{D}$ is an unknown diagonal matrix and d_{ii} is the unknown gain for i-th sensor. $\boldsymbol{A}(\theta)$: array manifold. $\bar{\theta}_k$: unknown direction of arrival. $\{x_k\}_{k=1}^{s}$ are the impinging signals.

How is it related to compressive sensing?

Discretize the manifold function $\boldsymbol{A}(\theta)$ over $[-\pi \leq \theta < \pi]$ on N grid points.

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

where

$$\boldsymbol{A} = \begin{bmatrix} | & \cdots & | \\ \boldsymbol{A}(\theta_1) & \cdots & \boldsymbol{A}(\theta_N) \\ | & \cdots & | \end{bmatrix} \in \mathbb{C}^{L \times N}$$

- To achieve high resolution, we usually have $L \leq N$.
- $\boldsymbol{x} \in \mathbb{C}^{N\times 1}$ is s-sparse. Its s nonzero entries correspond to the directions of signals. Moreover, we don't know the locations of nonzero entries.
- Subspace constraint: assume $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{B}\boldsymbol{h})$ where $\boldsymbol{B}$ is a known $L \times K$ matrix and $K < L$.
- Number of constraints: L; number of unknowns: $K + s$.

Self-calibration and biconvex compressive sensing

Goal: Find $(\boldsymbol{h}, \boldsymbol{x})$ s.t. $\boldsymbol{y} = \operatorname{diag}(\boldsymbol{B}\boldsymbol{h})\boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$ and $\boldsymbol{x}$ is sparse.

Biconvex compressive sensing

We are solving a biconvex (not convex) optimization problem to recover sparse signal $\boldsymbol{x}$ and calibrating parameter $\boldsymbol{h}$.

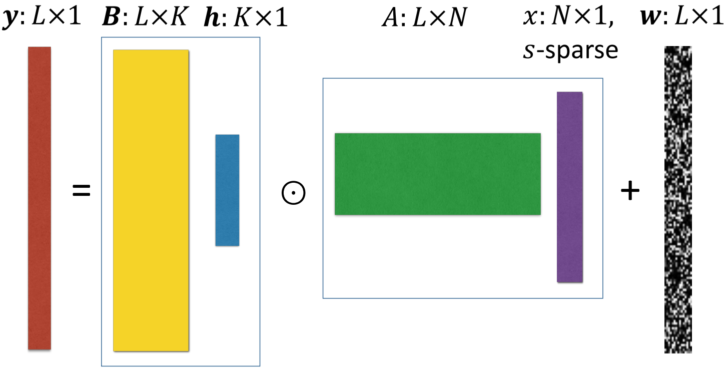
$$\min_{\boldsymbol{h},\boldsymbol{x}} \|\operatorname{diag}(\boldsymbol{B}\boldsymbol{h})\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 + \lambda\|\boldsymbol{x}\|_1$$

$\boldsymbol{A} \in \mathbb{C}^{L\times N}$ and $\boldsymbol{B} \in \mathbb{C}^{L\times K}$ are known. $\boldsymbol{h} \in \mathbb{C}^{K\times 1}$ and $\boldsymbol{x} \in \mathbb{C}^{N\times 1}$ are unknown. $\boldsymbol{x}$ is sparse.

Remark: If $\boldsymbol{h}$ is known, $\boldsymbol{x}$ can be recovered; if $\boldsymbol{x}$ is known, we can find $\boldsymbol{h}$ as well. Regarding identifiability issue, See [Lee, Bresler, etc. 15].

Biconvex compressive sensing

Goal: we want to find $\boldsymbol{h}$ and a sparse $\boldsymbol{x}$ from $\boldsymbol{y}$, $\boldsymbol{B}$ and $\boldsymbol{A}$.



Convex approach and lifting

Two-step convex approach

(a) Lifting: convert bilinear to linear constraints

(b) Solving a convex relaxation to recover $\boldsymbol{h}_0\boldsymbol{x}_0^*$.

Step 1: lifting

Let $\boldsymbol{a}_i$ be the i-th column of $\boldsymbol{A}^*$ and $\boldsymbol{b}_i$ be the i-th column of $\boldsymbol{B}^*$.

$$y_i = (\boldsymbol{B}\boldsymbol{h}_0)_i \boldsymbol{x}_0^* \boldsymbol{a}_i + w_i = \boldsymbol{b}_i^* \boldsymbol{h}_0 \boldsymbol{x}_0^* \boldsymbol{a}_i + w_i.$$

Let $\boldsymbol{X}_0 := \boldsymbol{h}_0\boldsymbol{x}_0^*$ and define the linear operator $\mathcal{A} : \mathbb{C}^{K\times N} \to \mathbb{C}^L$ as,

$$\mathcal{A}(\boldsymbol{Z}) := \{\boldsymbol{b}_i^* \boldsymbol{Z} \boldsymbol{a}_i\}_{i=1}^L = \{\langle \boldsymbol{Z}, \boldsymbol{b}_i \boldsymbol{a}_i^* \rangle\}_{i=1}^L.$$

Then, there holds

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{X}_0) + \boldsymbol{w}.$$

In this way, $\mathcal{A}^*(\boldsymbol{z}) = \sum_{i=1}^L z_i \boldsymbol{b}_i \boldsymbol{a}_i^* : \mathbb{C}^L \to \mathbb{C}^{K\times N}$.

Convex approach and lifting

Two-step convex approach

(a) Lifting: convert bilinear to linear constraints

(b) Solving a convex relaxation to recover $\boldsymbol{h}_0\boldsymbol{x}_0^*$.

Step 1: lifting

Let $\boldsymbol{a}_i$ be the i-th column of $\boldsymbol{A}^*$ and $\boldsymbol{b}_i$ be the i-th column of $\boldsymbol{B}^*$.

$$y_i = (\boldsymbol{B}\boldsymbol{h}_0)_i \boldsymbol{x}_0^* \boldsymbol{a}_i + w_i = \boldsymbol{b}_i^* \boldsymbol{h}_0 \boldsymbol{x}_0^* \boldsymbol{a}_i + w_i.$$

Let $\boldsymbol{X}_0 := \boldsymbol{h}_0 \boldsymbol{x}_0^*$ and define the linear operator $\mathcal{A} : \mathbb{C}^{K\times N} \to \mathbb{C}^L$ as,

$$\mathcal{A}(\boldsymbol{Z}) := \{\boldsymbol{b}_i^* \boldsymbol{Z} \boldsymbol{a}_i\}_{i=1}^L = \{\langle \boldsymbol{Z}, \boldsymbol{b}_i \boldsymbol{a}_i^* \rangle\}_{i=1}^L.$$

Then, there holds

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{X}_0) + \boldsymbol{w}.$$

In this way, $\mathcal{A}^*(\boldsymbol{z}) = \sum_{i=1}^L z_i \boldsymbol{b}_i \boldsymbol{a}_i^* : \mathbb{C}^L \to \mathbb{C}^{K\times N}$.

Rank-1 matrix recovery

Lifting: recovery of a rank - 1 and row-sparse matrix

$$\text{Find } \boldsymbol{Z} \quad \text{s.t.} \quad \text{rank}(\boldsymbol{Z}) = 1$$
$$\mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0)$$
$$\boldsymbol{Z} \text{ has sparse rows}$$

- $\|\boldsymbol{X}_0\|_0 = Ks$ where $\boldsymbol{X}_0 = \boldsymbol{h}_0 \boldsymbol{x}_0^*$, $\boldsymbol{h}_0 \in \mathbb{C}^K$ and $\boldsymbol{x}_0 \in \mathbb{C}^N$ with $\|\boldsymbol{x}_0\|_0 = s$.

$$\boldsymbol{Z} = \begin{bmatrix} 0 & 0 & h_1 x_{i_1} & 0 & \cdots & 0 & h_1 x_{i_s} & 0 & \cdots & 0 \\ 0 & 0 & h_2 x_{i_1} & 0 & \cdots & 0 & h_2 x_{i_s} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & h_K x_{i_1} & 0 & \cdots & 0 & h_K x_{i_s} & 0 & \cdots & 0 \end{bmatrix}_{K \times N}$$

- An NP-hard problem to find such a rank-1 and sparse matrix.

Rank-1 matrix recovery

Lifting: recovery of a rank - 1 and row-sparse matrix

$$\text{Find } \boldsymbol{Z} \quad \text{s.t.} \quad \begin{aligned} &\text{rank}(\boldsymbol{Z}) = 1 \\ &\mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0) \\ &\boldsymbol{Z} \text{ has sparse rows} \end{aligned}$$

- $\|\boldsymbol{X}_0\|_0 = Ks$ where $\boldsymbol{X}_0 = \boldsymbol{h}_0\boldsymbol{x}_0^*$, $\boldsymbol{h}_0 \in \mathbb{C}^K$ and $\boldsymbol{x}_0 \in \mathbb{C}^N$ with $\|\boldsymbol{x}_0\|_0 = s$.

$$\boldsymbol{Z} = \begin{bmatrix} 0 & 0 & h_1 x_{i_1} & 0 & \cdots & 0 & h_1 x_{i_s} & 0 & \cdots & 0 \\ 0 & 0 & h_2 x_{i_1} & 0 & \cdots & 0 & h_2 x_{i_s} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & h_K x_{i_1} & 0 & \cdots & 0 & h_K x_{i_s} & 0 & \cdots & 0 \end{bmatrix}_{K \times N}$$

- An NP-hard problem to find such a rank-1 and sparse matrix.

SparseLift

$\|\boldsymbol{Z}\|_*$: nuclear norm and $\|\boldsymbol{Z}\|_1$: ℓ_1-norm of vectorized $\boldsymbol{Z}$.

A popular way: nuclear norm + ℓ_1- minimization

$$\min \|\boldsymbol{Z}\|_1 + \lambda \|\boldsymbol{Z}\|_* \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0), \quad \lambda \geq 0.$$

However, combination of multiple norms may not do any better. [Oymak, Jalali, Fazel, Eldar and Hassibi 12].

SparseLift

$$\min \|\boldsymbol{Z}\|_1 \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0).$$

Idea: Lift the recovery problem of two unknown vectors to a matrix-valued problem and exploit sparsity through ℓ_1-minimization.

SparseLift

$\|\boldsymbol{Z}\|_*$: nuclear norm and $\|\boldsymbol{Z}\|_1$: ℓ_1-norm of vectorized $\boldsymbol{Z}$.

A popular way: nuclear norm + ℓ_1- minimization

$$\min \|\boldsymbol{Z}\|_1 + \lambda \|\boldsymbol{Z}\|_* \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0), \quad \lambda \geq 0.$$

However, combination of multiple norms may not do any better. [Oymak, Jalali, Fazel, Eldar and Hassibi 12].

SparseLift

$$\min \|\boldsymbol{Z}\|_1 \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0).$$

Idea: Lift the recovery problem of two unknown vectors to a matrix-valued problem and exploit sparsity through ℓ_1-minimization.

Main theorem

Theorem: [Ling-Strohmer, 2015]

Recall the model:

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{A}\boldsymbol{x}, \quad \boldsymbol{D} = \operatorname{diag}(\boldsymbol{B}\boldsymbol{h}),$$

where

(a) $\boldsymbol{B}$ is an $L \times K$ DFT tall matrix with $\boldsymbol{B}^*\boldsymbol{B} = \boldsymbol{I}_K$

(b) $\boldsymbol{A}$ is an $L \times N$ real Gaussian random matrix or a random Fourier matrix.

Then SparseLift recovers $\boldsymbol{X}_0$ exactly with high probability if

$$L = \mathcal{O}(\underbrace{K}_{\text{dimension of } \boldsymbol{h}} \underbrace{s}_{\text{level of sparsity}} \log^2 L)$$

where $Ks = \|\boldsymbol{X}_0\|_0$.

Comments

- $\min \|\boldsymbol{X}\|_*$ fails if $L < N$.

$\min \|\boldsymbol{X}\|_*$	$L = \mathcal{O}(K + N)$
$\min \|\boldsymbol{X}\|_1$	$L = \mathcal{O}(\mathbf{Ks} \log KN)$

- Solving ℓ_1-minimization is easier and cheaper than solving SDP.
- Compared with Compressive Sensing

Compressive Sensing	$L = \mathcal{O}(\mathbf{s} \log N)$
Our Case	$L = \mathcal{O}(K\mathbf{s} \log KN)$

- Believed to be optimal if one uses the 'Lifting' technique. It is unknown whether any algorithm would work for $L = \mathcal{O}(K + s)$.

Phase transition: SparseLift vs. $\|\cdot\|_1 + \lambda\|\cdot\|_*$

$\min \|\cdot\|_1 + \lambda\|\cdot\|_*$ does not do any better than $\min \|\cdot\|_1$.
White: Success, Black: Failure

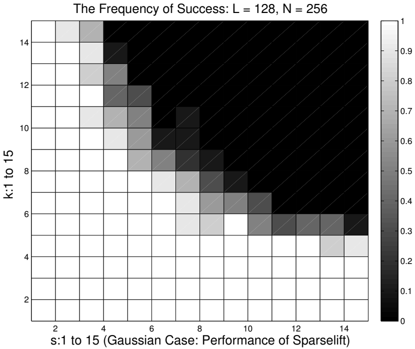


Figure: SparseLift

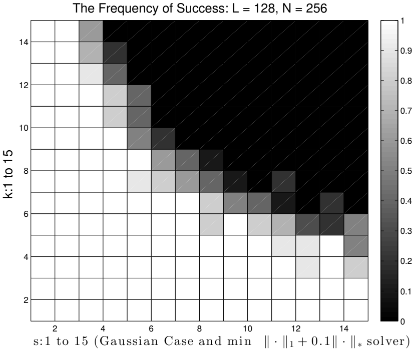


Figure: $\min \|\cdot\|_1 + 0.1\|\cdot\|_*$

$L = 128, N = 256$. $\boldsymbol{A}$: Gaussian and $\boldsymbol{B}$: Non-random partial Fourier matrix. 10 experiments for each pair (K, s), $1 \le K, s \le 15$.

Minimal L is nearly proportional to Ks

L : 10 to 400; $N = 512$; $\boldsymbol{A}$: Gaussian random matrices;
$\boldsymbol{B}$: first K columns of a DFT matrix.

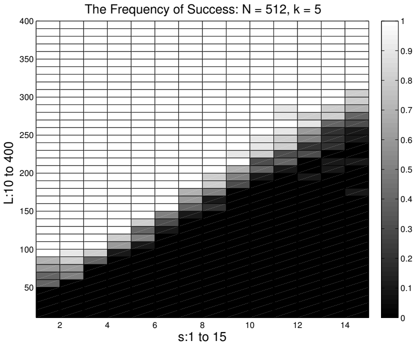


Figure: Fix $K = 5$

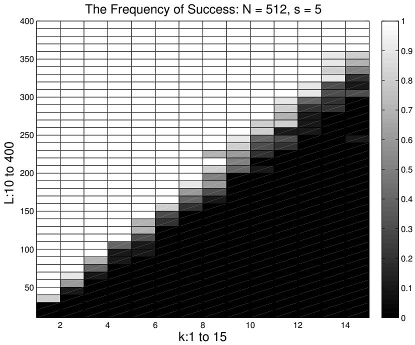


Figure: Fix $s = 5$

Stability theory

Assume that $\boldsymbol{y}$ is contaminated by noise, namely, $\boldsymbol{y} = \mathcal{A}(\boldsymbol{X}_0) + \boldsymbol{w}$ with $\|\boldsymbol{w}\| \leq \eta$, we solve the following program to recover $\boldsymbol{X}_0$,

$$\min \|\boldsymbol{Z}\|_1 \quad \text{s.t. } \|\mathcal{A}(\boldsymbol{Z}) - \boldsymbol{y}\| \leq \eta.$$

Theorem

If $\boldsymbol{A}$ is either a Gaussian random matrix or a random Fourier matrix,

$$\|\hat{\boldsymbol{X}} - \boldsymbol{X}_0\|_F \leq (C_0 + C_1\sqrt{Ks})\eta$$

with high probability. L satisfies the condition in the noiseless case. Both C_0 and C_1 are constants.

Stability theory

Assume that $\boldsymbol{y}$ is contaminated by noise, namely, $\boldsymbol{y} = \mathcal{A}(\boldsymbol{X}_0) + \boldsymbol{w}$ with $\|\boldsymbol{w}\| \leq \eta$, we solve the following program to recover $\boldsymbol{X}_0$,

$$\min \|\boldsymbol{Z}\|_1 \quad \text{s.t. } \|\mathcal{A}(\boldsymbol{Z}) - \boldsymbol{y}\| \leq \eta.$$

Theorem

If $\boldsymbol{A}$ is either a Gaussian random matrix or a random Fourier matrix,

$$\|\hat{\boldsymbol{X}} - \boldsymbol{X}_0\|_F \leq (C_0 + C_1\sqrt{Ks})\eta$$

with high probability. L satisfies the condition in the noiseless case. Both C_0 and C_1 are constants.

Numerical example: relative error vs SNR

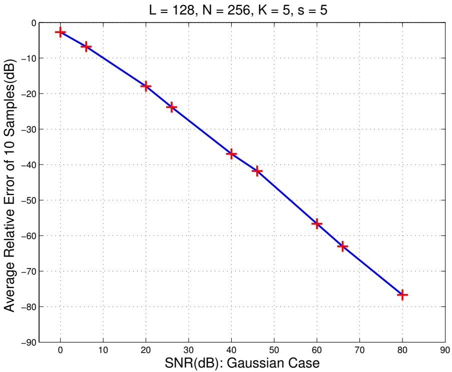


Figure: $\boldsymbol{A}$: Gaussian matrix

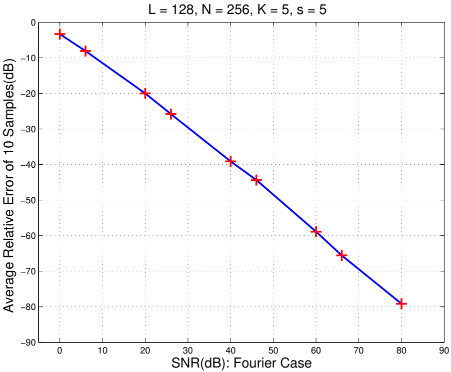


Figure: $\boldsymbol{A}$: random Fourier matrix

Remarks: $L = 128, N = 256, K = s = 5$.

Part II: Blind deconvolution and nonconvex optimization

What is blind deconvolution?

What is blind deconvolution?

Suppose we observe a function $\boldsymbol{y}$ which consists of the convolution of two unknown functions, the blurring function $\boldsymbol{f}$ and the signal of interest $\boldsymbol{g}$, plus noise $\boldsymbol{w}$. How to reconstruct $\boldsymbol{f}$ and $\boldsymbol{g}$ from $\boldsymbol{y}$?

$$\boldsymbol{y} = \boldsymbol{f} * \boldsymbol{g} + \boldsymbol{w}.$$

It is obviously a highly ill-posed bilinear inverse problem...

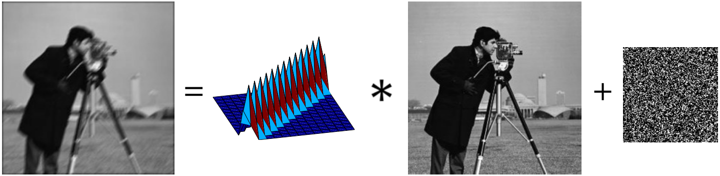
- Much more difficult than ordinary deconvolution...but have important applications in various fields.
- Solvability? What conditions on $\boldsymbol{f}$ and $\boldsymbol{g}$ make this problem solvable?
- How? What algorithms shall we use to recover $\boldsymbol{f}$ and $\boldsymbol{g}$?

Image deblurring

Let $\boldsymbol{f}$ be the blurring kernel and $\boldsymbol{g}$ be the original image, then $\boldsymbol{y} = \boldsymbol{f} * \boldsymbol{g}$ is the blurred image.

Question: how to reconstruct $\boldsymbol{f}$ and $\boldsymbol{g}$ from $\boldsymbol{y}$

$$\underset{\text{blurred image}}{\boldsymbol{y}} = \underset{\text{blurring kernel}}{\boldsymbol{f}} * \underset{\text{original image}}{\boldsymbol{g}} + \underset{\text{noise}}{\boldsymbol{w}}$$

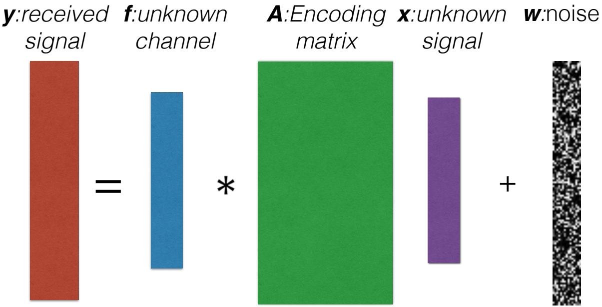


Why do we care about blind deconvolution?

Joint channel and signal estimation in wireless communication

Suppose that a signal $\boldsymbol{x}$, encoded by $\boldsymbol{A}$, is transmitted through an unknown channel $\boldsymbol{f}$. How to reconstruct $\boldsymbol{f}$ and $\boldsymbol{x}$ from $\boldsymbol{y}$?

$$\boldsymbol{y} = \boldsymbol{f} * \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}.$$



We start from the original model

$$\boldsymbol{y} = \boldsymbol{f} * \boldsymbol{g} + \boldsymbol{w}.$$

As mentioned before, it is an ill-posed problem. Phase retrieval is actually a special case if $\boldsymbol{g}(-x) = \bar{\boldsymbol{f}}(x)$. Hence, this problem is unsolvable without further assumptions...

Subspace assumption

Both $\boldsymbol{f}$ and $\boldsymbol{g}$ belong to known subspaces: there exist known tall matrices $\widetilde{\boldsymbol{B}} \in \mathbb{C}^{L\times K}$ and $\widetilde{\boldsymbol{A}} \in \mathbb{C}^{L\times N}$ such that

$$\boldsymbol{f} = \widetilde{\boldsymbol{B}}\boldsymbol{h}_0, \quad \boldsymbol{g} = \widetilde{\boldsymbol{A}}\boldsymbol{x}_0,$$

for some unknown vectors $\boldsymbol{h}_0 \in \mathbb{C}^K$ and $\boldsymbol{x}_0 \in \mathbb{C}^N$. Here $\boldsymbol{x}_0$ is not necessarily sparse.

Subspace assumptions

We start from the original model

$$\boldsymbol{y} = \boldsymbol{f} * \boldsymbol{g} + \boldsymbol{w}.$$

As mentioned before, it is an ill-posed problem. Phase retrieval is actually a special case if $\boldsymbol{g}(-x) = \bar{\boldsymbol{f}}(x)$. Hence, this problem is unsolvable without further assumptions...

Subspace assumption

Both $\boldsymbol{f}$ and $\boldsymbol{g}$ belong to known subspaces: there exist known tall matrices $\widetilde{\boldsymbol{B}} \in \mathbb{C}^{L\times K}$ and $\widetilde{\boldsymbol{A}} \in \mathbb{C}^{L\times N}$ such that

$$\boldsymbol{f} = \widetilde{\boldsymbol{B}}\boldsymbol{h}_0, \quad \boldsymbol{g} = \widetilde{\boldsymbol{A}}\boldsymbol{x}_0,$$

for some unknown vectors $\boldsymbol{h}_0 \in \mathbb{C}^K$ and $\boldsymbol{x}_0 \in \mathbb{C}^N$. Here $\boldsymbol{x}_0$ is not necessarily sparse.

Examples for subspace assumption:

Subspace assumption

Both $\boldsymbol{f}$ and $\boldsymbol{g}$ belong to known subspaces: there exist known tall matrices $\widetilde{\boldsymbol{B}} \in \mathbb{C}^{L\times K}$ and $\widetilde{\boldsymbol{A}} \in \mathbb{C}^{L\times N}$ such that

$$\boldsymbol{f} = \widetilde{\boldsymbol{B}}\boldsymbol{h}_0, \quad \boldsymbol{g} = \widetilde{\boldsymbol{A}}\boldsymbol{x}_0,$$

for some unknown vectors $\boldsymbol{h}_0 \in \mathbb{C}^K$ and $\boldsymbol{x}_0 \in \mathbb{C}^N$.

Useful examples:

- In image deblurring, $\widetilde{\boldsymbol{B}}$ can be the support of the blurring kernel; $\widetilde{\boldsymbol{A}}$ is a wavelet basis.
- In wireless communication, $\widetilde{\boldsymbol{B}}$ is related to the maximum delay spread and $\widetilde{\boldsymbol{A}}$ is an encoding matrix.

Model under subspace assumption

After taking Fourier transform, circular convolution becomes entrywise multiplication:

$$\boldsymbol{y} = (\widetilde{\boldsymbol{B}}\boldsymbol{h}_0) * (\widetilde{\boldsymbol{A}}\boldsymbol{x}_0) + \boldsymbol{w} \Longrightarrow \hat{\boldsymbol{y}} = \operatorname{diag}(\boldsymbol{B}\boldsymbol{h}_0)\boldsymbol{A}\boldsymbol{x}_0 + \hat{\boldsymbol{w}},$$

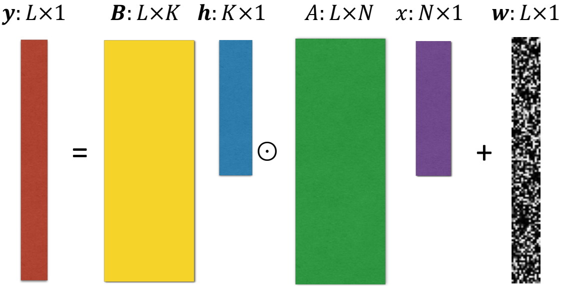
where

$$\hat{\boldsymbol{y}} = \boldsymbol{F}\boldsymbol{y} \in \mathbb{C}^L, \quad \boldsymbol{B} = \boldsymbol{F}\widetilde{\boldsymbol{B}}, \quad \boldsymbol{A} = \boldsymbol{F}\widetilde{\boldsymbol{A}}$$

and $\boldsymbol{F}$ is the $L \times L$ DFT matrix.

Goal: recover $\boldsymbol{h}_0, \boldsymbol{x}_0$ from $\boldsymbol{B}$, $\boldsymbol{A}$, and $\hat{\boldsymbol{y}}$.

More on subspace assumption



Since we don't assume $\boldsymbol{x}$ to be sparse, the degree of freedom for unknowns is $K + N$; number of constraints: L.

Mathematical model

$$\boldsymbol{y} = \text{diag}(\boldsymbol{B}\boldsymbol{h}_0)\boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{w},$$

where $\frac{\boldsymbol{w}}{d_0} \sim \frac{1}{\sqrt{2}}\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_L) + i\frac{1}{\sqrt{2}}\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_L)$ and $d_0 = \|\boldsymbol{h}_0\|\|\boldsymbol{x}_0\|$.
One might want to solve the following nonlinear least squares problem,

$$\min F(\boldsymbol{h}, \boldsymbol{x}) := \|\text{diag}(\boldsymbol{B}\boldsymbol{h})\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2.$$

Difficulties:

1. **Nonconvexity:** F is a nonconvex function; algorithms (such as gradient descent) are likely to get trapped at local minima.
2. **No performance guarantees.**

Mathematical model

$$\boldsymbol{y} = \mathrm{diag}(\boldsymbol{B}\boldsymbol{h}_0)\boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{w},$$

where $\frac{\boldsymbol{w}}{d_0} \sim \frac{1}{\sqrt{2}}\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_L) + i\frac{1}{\sqrt{2}}\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_L)$ and $d_0 = \|\boldsymbol{h}_0\|\|\boldsymbol{x}_0\|$.
One might want to solve the following nonlinear least squares problem,

$$\min F(\boldsymbol{h}, \boldsymbol{x}) := \|\,\mathrm{diag}(\boldsymbol{B}\boldsymbol{h})\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2.$$

Difficulties:

1. **Nonconvexity:** F is a nonconvex function; algorithms (such as gradient descent) are likely to get trapped at local minima.
2. **No performance guarantees.**

Convex relaxation and state of the art

Nuclear norm minimization

Consider the convex envelop of rank($\boldsymbol{Z}$): nuclear norm $\|\boldsymbol{Z}\|_* = \sum \sigma_i(\boldsymbol{Z})$.

$$\min \|\boldsymbol{Z}\|_* \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0)$$

where $\boldsymbol{X}_0 = \boldsymbol{h}_0 \boldsymbol{x}_0^*$.

Theorem [Ahmed-Recht-Romberg 11]

Assume $\boldsymbol{y} = \text{diag}(\boldsymbol{B}\boldsymbol{h}_0)\boldsymbol{A}\boldsymbol{x}_0$, $\boldsymbol{A} : L \times N$ is a complex Gaussian random matrix,

$$\boldsymbol{B}^*\boldsymbol{B} = \boldsymbol{I}_K, \quad \|\boldsymbol{b}_i\|^2 \leq \frac{\mu_{\max}^2 K}{L}, \quad L\|\boldsymbol{B}\boldsymbol{h}_0\|_\infty^2 \leq \mu_h^2,$$

the above convex relaxation recovers $\boldsymbol{X} = \boldsymbol{h}_0 \boldsymbol{x}_0^*$ exactly with high probability if

$$C_0(K + \mu_h^2 N) \leq \frac{L}{\log^3 L}.$$

Convex relaxation and state of the art

Nuclear norm minimization

Consider the convex envelop of rank($\boldsymbol{Z}$): nuclear norm $\|\boldsymbol{Z}\|_* = \sum \sigma_i(\boldsymbol{Z})$.

$$\min \|\boldsymbol{Z}\|_* \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0)$$

where $\boldsymbol{X}_0 = \boldsymbol{h}_0 \boldsymbol{x}_0^*$.

Theorem [Ahmed-Recht-Romberg 11]

Assume $\boldsymbol{y} = \operatorname{diag}(\boldsymbol{B}\boldsymbol{h}_0)\boldsymbol{A}\boldsymbol{x}_0$, $\boldsymbol{A} : L \times N$ is a complex Gaussian random matrix,

$$\boldsymbol{B}^*\boldsymbol{B} = \boldsymbol{I}_K, \quad \|\boldsymbol{b}_i\|^2 \le \frac{\mu_{\max}^2 K}{L}, \quad L\|\boldsymbol{B}\boldsymbol{h}_0\|_\infty^2 \le \mu_h^2,$$

the above convex relaxation recovers $\boldsymbol{X} = \boldsymbol{h}_0 \boldsymbol{x}_0^*$ exactly with high probability if

$$C_0(K + \mu_h^2 N) \le \frac{L}{\log^3 L}.$$

Pros and Cons of Convex Approach

Pros and Cons

- Pros: Simple, efficient and comes with theoretic guarantees
- Cons: Computationally too expensive to solve SDP

Our Goal: **rapid, robust, reliable nonconvex approach**

- Rapid: linear convergence
- Robust: stable to noise
- Reliable: provable and comes with theoretic guarantees; number of measurement close to information-theoretic limits.

A nonconvex optimization approach?

An increasing list of nonconvex approach to various problems:

- Phase retrieval: by Candès, Li, Soltanolkotabi, Chen, etc...
- Matrix completion: by Sun, Luo, Montanari, etc...
- Various problems: by Wainwright, Recht, Constantine, etc...

Two-step philosophy for provable nonconvex optimization

(a) Use spectral initialization to construct a starting point inside *"the* ***basin of attraction***";

(b) Simple gradient descent method.

The key is to build up "the basin of attraction".

A nonconvex optimization approach?

An increasing list of nonconvex approach to various problems:

- Phase retrieval: by Candès, Li, Soltanolkotabi, Chen, etc...
- Matrix completion: by Sun, Luo, Montanari, etc...
- Various problems: by Wainwright, Recht, Constantine, etc...

Two-step philosophy for provable nonconvex optimization

(a) Use spectral initialization to construct a starting point inside *"the* ***basin of attraction****"*;

(b) Simple gradient descent method.

The key is to build up "the basin of attraction".

Building *"the basin of attraction"*

The basin of the attraction relies on the following three observations.

Observation 1: Unboundedness of solution

- If the pair $(\boldsymbol{h}_0, \boldsymbol{x}_0)$ is a solution to $\boldsymbol{y} = \text{diag}(\boldsymbol{B}\boldsymbol{h}_0)\boldsymbol{A}\boldsymbol{x}_0$, then so is the pair $(\alpha\boldsymbol{h}_0, \alpha^{-1}\boldsymbol{x}_0)$ for any $\alpha \neq 0$.
- Thus the blind deconvolution problem always has infinitely many solutions of this type. We can recover $(\boldsymbol{h}_0, \boldsymbol{x}_0)$ only up to a scalar.
- It is possible that $\|\boldsymbol{h}\| \gg \|\boldsymbol{x}\|$ (vice versa) while $\|\boldsymbol{h}\| \cdot \|\boldsymbol{x}\| = d_0$. Hence we define $\mathcal{N}_{d_0}$ to balance $\|\boldsymbol{h}\|$ and $\|\boldsymbol{x}\|$:

$$\mathcal{N}_{d_0} := \{(\boldsymbol{h}, \boldsymbol{x}) : \|\boldsymbol{h}\| \le 2\sqrt{d_0}, \|\boldsymbol{x}\| \le 2\sqrt{d_0}\}.$$

Building *"the basin of attraction"*

Observation 2: Incoherence

How much $\boldsymbol{b}_l$ and $\boldsymbol{h}_0$ are aligned matters:

$$\mu_h^2 := \frac{L\|\boldsymbol{B}\boldsymbol{h}_0\|_\infty^2}{\|\boldsymbol{h}_0\|^2} = L\frac{\max_i |\boldsymbol{b}_i^*\boldsymbol{h}_0|^2}{\|\boldsymbol{h}_0\|^2}, \quad \textbf{the smaller } \mu_h\textbf{, the better.}$$

Therefore, we introduce the $\mathcal{N}_\mu$ to control the incoherence:

$$\mathcal{N}_\mu := \{\boldsymbol{h} : \sqrt{L}\|\boldsymbol{B}\boldsymbol{h}\|_\infty \leq 4\mu\sqrt{d_0}\}.$$

"Incoherence" is not a new idea. In matrix completion, we also require the left and right singular vectors of the ground truth cannot be too "aligned" with those of measurement matrices $\{\boldsymbol{b}_i\boldsymbol{a}_i^*\}_{1\leq i\leq L}$. The same philosophy applies here.

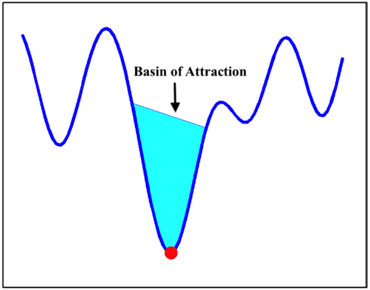
Building *"the basin of attraction"*

Observation 3: "Close" to the ground truth

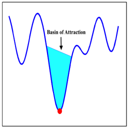
We define $\mathcal{N}_\varepsilon$ to quantify closeness of $(\boldsymbol{h}, \boldsymbol{x})$ to true solution, i.e.,

$$\mathcal{N}_\varepsilon := \{(\boldsymbol{h}, \boldsymbol{x}) : \|\boldsymbol{h}\boldsymbol{x}^* - \boldsymbol{h}_0\boldsymbol{x}_0^*\|_F \leq \varepsilon d_0\}.$$

We want to find an initial guess close to $(\boldsymbol{h}_0, \boldsymbol{x}_0)$.



Based on the three observations above, we define the three neighborhoods (denoting $d_0 = \|h_0\|\|x_0\|$):



$$
\begin{aligned}
\mathcal{N}_{d_0} &:= \{(\boldsymbol{h},\boldsymbol{x}) : \|\boldsymbol{h}\| \leq 2\sqrt{d_0}, \|\boldsymbol{x}\| \leq 2\sqrt{d_0}\} \\
\mathcal{N}_{\mu} &:= \{\boldsymbol{h} : \sqrt{L}\|\boldsymbol{B}\boldsymbol{h}\|_\infty \leq 4\mu\sqrt{d_0}\} \\
\mathcal{N}_{\varepsilon} &:= \{(\boldsymbol{h},\boldsymbol{x}) : \|\boldsymbol{h}\boldsymbol{x}^* - \boldsymbol{h}_0\boldsymbol{x}_0^*\|_F \leq \varepsilon d_0\}.
\end{aligned}
$$

where $\varepsilon < \frac{1}{15}$. We first obtain a good initial guess $(\boldsymbol{u}_0, \boldsymbol{v}_0) \in \mathcal{N}_{d_0} \cap \mathcal{N}_{\mu} \cap \mathcal{N}_{\varepsilon}$, which is followed by regularized gradient descent.

Objective function: a variant of projected gradient descent

The objective function $\widetilde{F}$ consists of two parts: F and G:

$$\min_{(\boldsymbol{h},\boldsymbol{x})} \quad \widetilde{F}(\boldsymbol{h},\boldsymbol{x}) := F(\boldsymbol{h},\boldsymbol{x}) + G(\boldsymbol{h},\boldsymbol{x}),$$

where $F(\boldsymbol{h},\boldsymbol{x}) = \|\mathcal{A}(\boldsymbol{h}\boldsymbol{x}^*) - \boldsymbol{y}\|^2 = \|\operatorname{diag}(\boldsymbol{B}\boldsymbol{h})\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2$ and

$$G(\boldsymbol{h},\boldsymbol{x}) := \rho\Big[\underbrace{G_0\left(\frac{\|\boldsymbol{h}\|^2}{2d}\right) + G_0\left(\frac{\|\boldsymbol{x}\|^2}{2d}\right)}_{\mathcal{N}_{d_0}} + \underbrace{\sum_{l=1}^{L} G_0\left(\frac{L|\boldsymbol{b}_l^*\boldsymbol{h}|^2}{8d\mu^2}\right)}_{\mathcal{N}_{\mu}}\Big].$$

Here $G_0(z) = \max\{z-1, 0\}^2$, $\rho \approx d^2$, $d \approx d_0$ and $\mu \geq \mu_h$.

Objective function: a variant of projected gradient descent

The objective function $\widetilde{F}$ consists of two parts: F and G:

$$\min_{(\boldsymbol{h},\boldsymbol{x})} \quad \widetilde{F}(\boldsymbol{h},\boldsymbol{x}) := F(\boldsymbol{h},\boldsymbol{x}) + G(\boldsymbol{h},\boldsymbol{x})$$

We refer F and G as

- F : least squares term, i.e., impose the measurement equations
- G : regularization term, i.e., regularization forces iterates $(\boldsymbol{u}_t, \boldsymbol{v}_t)$ inside $\mathcal{N}_{d_0} \cap \mathcal{N}_{\mu} \cap \mathcal{N}_{\varepsilon}$.

Algorithm: Wirtinger Gradient Descent

Step 1: Initialization via spectral method and projection:

1: Compute $\mathcal{A}^*(\boldsymbol{y})$, (since $\mathbb{E}(\mathcal{A}^*(\boldsymbol{y})) = \boldsymbol{h}_0\boldsymbol{x}_0^*$);
2: Find the leading singular value, left and right singular vectors of $\mathcal{A}^*(\boldsymbol{y})$, denoted by $(d, \hat{\boldsymbol{h}}_0, \hat{\boldsymbol{x}}_0)$ respectively;
3: $\boldsymbol{u}^{(0)} := \mathcal{P}_{\mathcal{N}_\mu}(\sqrt{d}\hat{\boldsymbol{h}}_0)$ and $\boldsymbol{v}^{(0)} := \sqrt{d}\hat{\boldsymbol{x}}_0$;
4: Output: $(\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)})$.

Step 2: Gradient descent with constant stepsize η:

1: **Initialization:** obtain $(\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)})$ via Algorithm 1.
2: **for** $t = 1, 2, \dots,$ **do**
3: $\boldsymbol{u}^{(t)} = \boldsymbol{u}^{(t-1)} - \eta\nabla\widetilde{F}_{\boldsymbol{h}}(\boldsymbol{u}^{(t-1)}, \boldsymbol{v}^{(t-1)})$
4: $\boldsymbol{v}^{(t)} = \boldsymbol{v}^{(t-1)} - \eta\nabla\widetilde{F}_{\boldsymbol{x}}(\boldsymbol{u}^{(t-1)}, \boldsymbol{v}^{(t-1)})$
5: **end for**

Algorithm: Wirtinger Gradient Descent

Step 1: Initialization via spectral method and projection:

1: Compute $\mathcal{A}^*(\boldsymbol{y})$, (since $\mathbb{E}(\mathcal{A}^*(\boldsymbol{y})) = \boldsymbol{h}_0\boldsymbol{x}_0^*$);
2: Find the leading singular value, left and right singular vectors of $\mathcal{A}^*(\boldsymbol{y})$, denoted by $(d, \hat{\boldsymbol{h}}_0, \hat{\boldsymbol{x}}_0)$ respectively;
3: $\boldsymbol{u}^{(0)} := \mathcal{P}_{\mathcal{N}_\mu}(\sqrt{d}\hat{\boldsymbol{h}}_0)$ and $\boldsymbol{v}^{(0)} := \sqrt{d}\hat{\boldsymbol{x}}_0$;
4: Output: $(\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)})$.

Step 2: Gradient descent with constant stepsize η:

1: **Initialization:** obtain $(\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)})$ via Algorithm 1.
2: **for** $t = 1, 2, \dots,$ **do**
3: $\quad \boldsymbol{u}^{(t)} = \boldsymbol{u}^{(t-1)} - \eta\nabla\widetilde{F}_{\boldsymbol{h}}(\boldsymbol{u}^{(t-1)}, \boldsymbol{v}^{(t-1)})$
4: $\quad \boldsymbol{v}^{(t)} = \boldsymbol{v}^{(t-1)} - \eta\nabla\widetilde{F}_{\boldsymbol{x}}(\boldsymbol{u}^{(t-1)}, \boldsymbol{v}^{(t-1)})$
5: **end for**

Main theorem

Theorem: [Li-Ling-Strohmer-Wei, 2016]

Let $\boldsymbol{B}$ be a tall partial DFT matrix and $\boldsymbol{A}$ be a complex Gaussian random matrix. If the number of measurements satisfies

$$L \geq C(\mu_h^2 + \sigma^2)(K + N)\log^2(L)/\varepsilon^2,$$

(i) then the initialization $(\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)}) \in \frac{1}{\sqrt{3}}\mathcal{N}_{d_0} \bigcap \frac{1}{\sqrt{3}}\mathcal{N}_{\mu} \bigcap \mathcal{N}_{\frac{2}{5}\varepsilon}$;
(ii) the regularized gradient descent algorithm creates a sequence $(\boldsymbol{u}^{(t)}, \boldsymbol{v}^{(t)})$ in $\mathcal{N}_{d_0} \cap \mathcal{N}_{\mu} \cap \mathcal{N}_{\varepsilon}$ satisfying

$$\|\boldsymbol{u}^{(t)}(\boldsymbol{v}^{(t)})^* - \boldsymbol{h}_0\boldsymbol{x}_0^*\|_F \leq (1-\alpha)^t \varepsilon d_0 + c_0\|\mathcal{A}^*(\boldsymbol{w})\|$$

with high probability where $\alpha = \mathcal{O}(\frac{1}{(1+\sigma^2)(K+N)\log^2 L})$

Remarks

(a) If $\boldsymbol{w} = \mathbf{0}$, $(\boldsymbol{u}^{(t)}, \boldsymbol{v}^{(t)})$ converges to $(\boldsymbol{h}_0, \boldsymbol{x}_0)$ linearly.

$$\|\boldsymbol{u}^{(t)}(\boldsymbol{v}^{(t)})^* - \boldsymbol{h}_0\boldsymbol{x}_0^*\|_F \leq (1-\alpha)^t \varepsilon d_0 \to 0, \text{ as } t \to \infty$$

(b) If $\boldsymbol{w} \neq \mathbf{0}$, $(\boldsymbol{u}^{(t)}, \boldsymbol{v}^{(t)})$ converges to a small neighborhood of $(\boldsymbol{h}_0, \boldsymbol{x}_0)$ linearly.

$$\|\boldsymbol{u}^{(t)}(\boldsymbol{v}^{(t)})^* - \boldsymbol{h}_0\boldsymbol{x}_0^*\|_F \to c_0\|\mathcal{A}^*(\boldsymbol{w})\|, \text{ as } t \to \infty$$

where

$$\|\mathcal{A}^*(\boldsymbol{w})\| = \mathcal{O}\left(\sigma d_0 \sqrt{\frac{(K+N)\log L}{L}}\right) \to 0, \text{ if } L \to \infty.$$

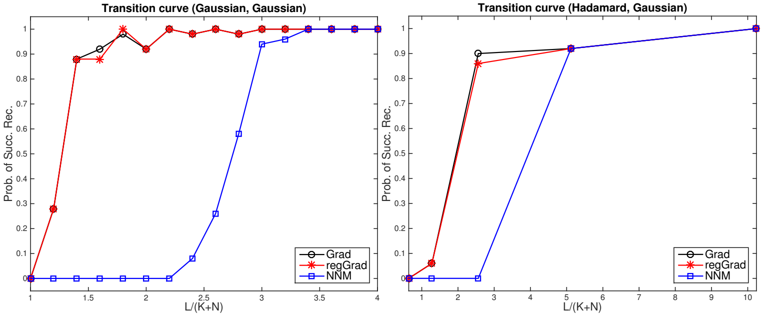
As L is becoming larger and larger, the effect of noise diminishes. (Recall linear least squares.)

Numerical experiments

Nonconvex approach v.s. convex approach:

$$\min_{(\boldsymbol{h},\boldsymbol{x})} \widetilde{F}(\boldsymbol{h},\boldsymbol{x}) \quad \text{v.s.} \quad \min \|\boldsymbol{Z}\|_* \quad s.t.\|\mathcal{A}(\boldsymbol{Z}) - \boldsymbol{y}\| \le \eta.$$

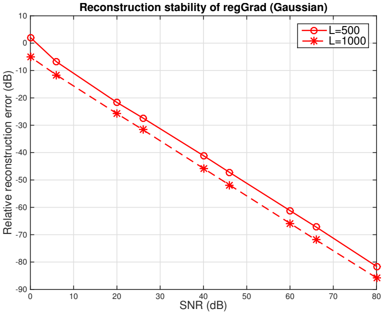
Nonconvex method requires fewer measurements to achieve exact recovery than convex method. Moreover, if $\boldsymbol{A}$ is a partial Hadamard matrix, our algorithm still gives satisfactory performance.



$K = N = 50$, $\boldsymbol{B}$ is a low-frequency DFT matrix.

Stability

Our algorithm yields stable recovery if the observation is noisy.

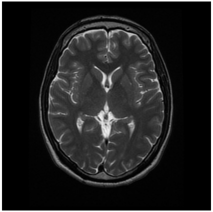


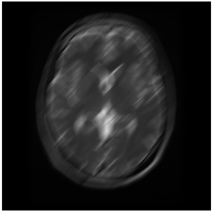
Here $K = N = 100$.

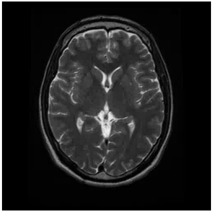
MRI image deblurring:

Here $\boldsymbol{B}$ is a partial DFT matrix and $\boldsymbol{A}$ is a partial wavelet matrix.

When the subspace $\boldsymbol{B}, (K = 65)$ or support of blurring kernel is known: $\boldsymbol{g} \approx \boldsymbol{Ax}$: image of 512×512; $\boldsymbol{A}$: wavelet subspace corresponding to the $N = 20000$ largest Haar wavelet coefficients of $\boldsymbol{g}$.

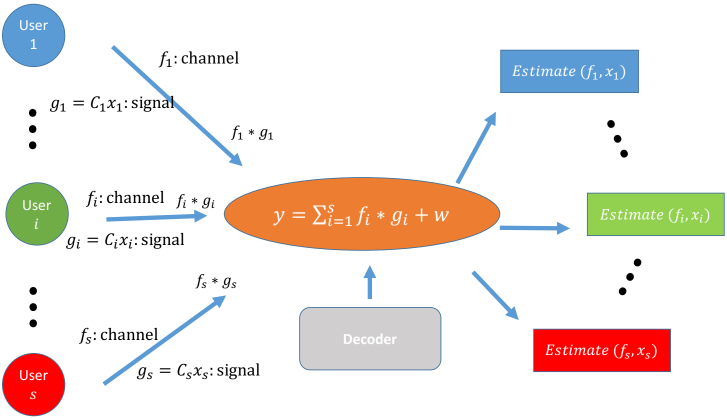






Extended to joint blind deconvolution and blind demixing

Suppose there are s users and each of them sends a message $\boldsymbol{x}_i$, which is encoded by $\boldsymbol{C}_i$, to a common receiver. Each encoded message $\boldsymbol{g}_i = \boldsymbol{C}_i \boldsymbol{x}_i$ is convolved with an unknown impulse response function $\boldsymbol{f}_i$.



Suppose that

- Each impulse response $\boldsymbol{f}_i$ has maximum delay spread K (compact support):
$$\boldsymbol{f}_i(n) = 0, \quad \text{for } n > K.$$
- $\boldsymbol{g}_i := \boldsymbol{C}_i \boldsymbol{x}_i$ is the signal $\boldsymbol{x}_i \in \mathbb{C}^N$ encoded by $\boldsymbol{C}_i \in \mathbb{C}^{L \times N}$ with $L > N$.

Mathematical model

Let $\boldsymbol{B}$ be the first K columns of the DFT matrix and $\boldsymbol{A}_i = \boldsymbol{F}\boldsymbol{C}_i$,

$$\boldsymbol{y} = \sum_{i=1}^{s} \operatorname{diag}(\boldsymbol{B}\boldsymbol{h}_i)\boldsymbol{A}_i\boldsymbol{x}_i + \boldsymbol{w}.$$

Goal: We want to recover $\{(\boldsymbol{h}_i, \boldsymbol{x}_i)\}_{i=1}^{s}$ from $(\boldsymbol{y}, \boldsymbol{B}, \{\boldsymbol{A}_i\}_{i=1}^{s})$.
The degree of freedom for unknowns: $s(K+N)$; number of constraints: L.

Objective function: a variant of projected gradient descent

The objective function $\widetilde{F}$ consists of two parts: F and G,

$$\min_{(\boldsymbol{h},\boldsymbol{x})} \quad \widetilde{F}(\boldsymbol{h},\boldsymbol{x}) := \underbrace{F(\boldsymbol{h},\boldsymbol{x})}_{\text{least squares term}} + \underbrace{G(\boldsymbol{h},\boldsymbol{x})}_{\text{regularization term}}$$

where $F(\boldsymbol{h},\boldsymbol{x}) := \| \sum_{i=1}^{s} \operatorname{diag}(\boldsymbol{B}\boldsymbol{h}_i)\boldsymbol{A}_i\boldsymbol{x}_i - \boldsymbol{y}\|^2$ and

$$G(\boldsymbol{h},\boldsymbol{x}) := \rho \sum_{i=1}^{s} \Bigg[\underbrace{G_0\left(\frac{\|\boldsymbol{h}_i\|^2}{2d_i}\right) + G_0\left(\frac{\|\boldsymbol{x}_i\|^2}{2d_i}\right)}_{\mathcal{N}_{d_0}\text{: balance } \|\boldsymbol{h}_i\| \text{ and } \|\boldsymbol{x}_i\|} + \underbrace{\sum_{l=1}^{L} G_0\left(\frac{L|\boldsymbol{b}_l^* \boldsymbol{h}_i|^2}{8d_i\mu^2}\right)}_{\mathcal{N}_{\mu}\text{: impose incoherence}} \Bigg].$$

Algorithm:

- Spectral initialization
- Apply gradient descent to $\widetilde{F}$

Main results

Theorem [Ling-Strohmer 17]

Assume $\boldsymbol{w} \sim \mathcal{CN}(0, \sigma^2 d_0^2/L)$ and $\boldsymbol{A}_i$ as a complex Gaussian matrix. Starting with the initial value

$$(\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)}) \in \frac{1}{\sqrt{3}}\mathcal{N}_{d_0} \bigcap \frac{1}{\sqrt{3}}\mathcal{N}_{\mu} \bigcap \mathcal{N}_{\frac{2\varepsilon}{5\sqrt{s}\kappa}},$$

$(\boldsymbol{u}^{(t)}, \boldsymbol{v}^{(t)})$ converges to the global minima linearly,

$$\sqrt{\sum_{i=1}^{s} \|\boldsymbol{u}_i^{(t)}(\boldsymbol{v}_i^{(t)})^* - \boldsymbol{h}_{i0}\boldsymbol{x}_{i0}^*\|_F^2} \leq \underbrace{(1-\alpha)^t \varepsilon d_0}_{\text{linear convergence}} + \underbrace{c_0\|\mathcal{A}^*(\boldsymbol{w})\|}_{\text{error term}}$$

with probability at least $1 - L^{-\gamma+1}$ and $\alpha = \mathcal{O}((s(K+N)\log^2 L)^{-1})$ if

$$L \geq C_\gamma(\mu_h^2 + \sigma^2)s^2\kappa^4(K+N)\log^2 L \log s/\varepsilon^2.$$

Numerics: Does L scale linearly with s?

Let each $\boldsymbol{A}_i$ be a complex Gaussian matrix. The number of measurement scales linearly with the number of sources s if K and N are fixed. Approximately, $L \approx 1.5s(K+N)$ yields exact recovery.

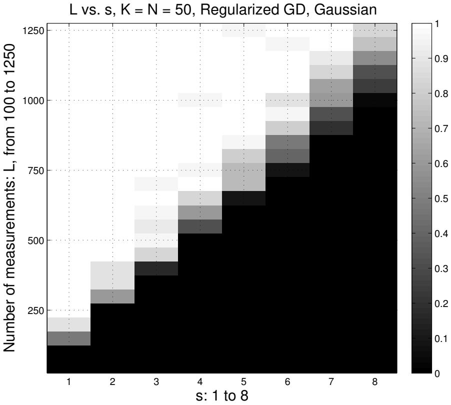
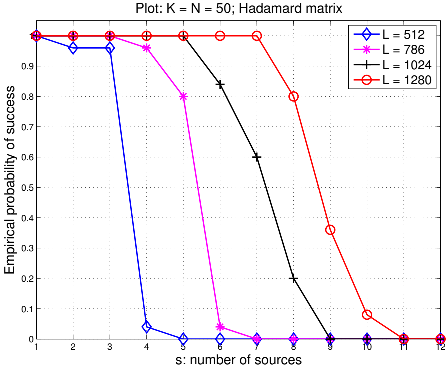


Figure: Black: failure; white: success

A communication example

A more practical and useful choice of encoding matrix $\boldsymbol{C}_i$: $\boldsymbol{C}_i = \boldsymbol{D}_i\boldsymbol{H}$ (i.e., $\boldsymbol{A}_i = \boldsymbol{F}\boldsymbol{D}_i\boldsymbol{H}$) where $\boldsymbol{D}_i$ is a diagonal random binary ± 1 matrix and $\boldsymbol{H}$ is an $L \times N$ deterministic partial Hadamard matrix. With this setting, our approach can demix many users **without** performing channel estimation.



$L \approx 1.5s(K + N)$ yields exact recovery.

Important ingredients of proof

The first three conditions hold over "the basin of attraction" $\mathcal{N}_{d_0} \cap \mathcal{N}_{\mu} \cap \mathcal{N}_{\varepsilon}$.

Condition 1: Local Regularity Condition

Guarantee sufficient decrease in each iterate and linear convergence of $\widetilde{F}$:

$$\|\nabla \widetilde{F}(\boldsymbol{h}, \boldsymbol{x})\|^2 \geq \omega \widetilde{F}(\boldsymbol{h}, \boldsymbol{x})$$

where $\omega > 0$ and $(\boldsymbol{h}, \boldsymbol{x}) \in \mathcal{N}_{d_0} \cap \mathcal{N}_{\mu} \cap \mathcal{N}_{\varepsilon}$.

Condition 2: Local Smoothness Condition

Governs rate of convergence. Let $\boldsymbol{z} = (\boldsymbol{h}, \boldsymbol{x})$. There exists a constant C_L (Lipschitz constant of gradient) such that

$$\|\nabla \widetilde{F}(\boldsymbol{z} + t\Delta \boldsymbol{z}) - \nabla \widetilde{F}(\boldsymbol{z})\| \leq C_L t \|\Delta \boldsymbol{z}\|, \quad \forall 0 \leq t \leq 1,$$

for all $\{(\boldsymbol{z}, \Delta \boldsymbol{z}) : \boldsymbol{z} + t\Delta \boldsymbol{z} \in \mathcal{N}_{d_0} \cap \mathcal{N}_{\mu} \cap \mathcal{N}_{\varepsilon}, \forall 0 \leq t \leq 1\}$.

Important ingredients of proof

Condition 3: Local Restricted Isometry Property

Transfer convergence of objective function to convergence of iterates.

$$\frac{2}{3}\|\boldsymbol{h}\boldsymbol{x}^* - \boldsymbol{h}_0\boldsymbol{x}_0^*\|_F^2 \le \|\mathcal{A}(\boldsymbol{h}\boldsymbol{x}^* - \boldsymbol{h}_0\boldsymbol{x}_0^*)\|^2 \le \frac{3}{2}\|\boldsymbol{h}\boldsymbol{x}^* - \boldsymbol{h}_0\boldsymbol{x}_0^*\|_F^2$$

holds uniformly for all $(\boldsymbol{h}, \boldsymbol{x}) \in \mathcal{N}_{d_0} \cap \mathcal{N}_\mu \cap \mathcal{N}_\varepsilon$.

Condition 4: Robustness Condition

Provide stability against noise.

$$\|\mathcal{A}^*(\boldsymbol{w})\| \le \frac{\varepsilon d_0}{10\sqrt{2}}.$$

where $\mathcal{A}^*(\boldsymbol{w}) = \sum_{l=1}^{L} w_l \boldsymbol{b}_l \boldsymbol{a}_l^*$ is a sum of L rank-1 random matrices. It concentrates around $\mathbf{0}$.

Outlook and Conclusion

Conclusion: The proposed algorithm is arguably the first nonconvex blind deconvolution/demixing algorithm with rigorous recovery guarantees. We also propose a convex approach (sub-optimal) to solve a self-calibration problem related to biconvex compressive sensing.

- Can we show if similar result holds for other types of $\boldsymbol{A}$?
- What if $\boldsymbol{x}$ or $\boldsymbol{h}$ is sparse/both of them are sparse?
- **See details**:
  1. Self-calibration and biconvex compressive sensing. *Inverse Problems* 31 (11), 115002
  2. Blind deconvolution meets blind demixing: algorithms and performance bounds, *To appear in IEEE Trans on Information Theory*
  3. Rapid, robust, and reliable blind deconvolution via nonconvex optimization, *arXiv:1606.04933*.
  4. Regularized gradient descent: a nonconvex recipe for fast joint blind deconvolution and demixing *arXiv:1703.08642*.

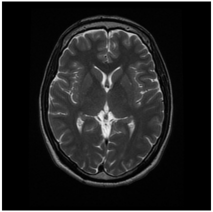
Outlook and Conclusion

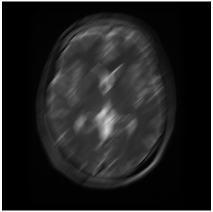
Conclusion: The proposed algorithm is arguably the first nonconvex blind deconvolution/demixing algorithm with rigorous recovery guarantees. We also propose a convex approach (sub-optimal) to solve a self-calibration problem related to biconvex compressive sensing.

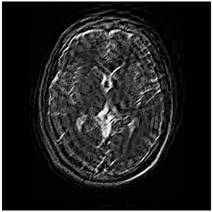
- Can we show if similar result holds for other types of $\boldsymbol{A}$?
- What if $\boldsymbol{x}$ or $\boldsymbol{h}$ is sparse/both of them are sparse?
- **See details**:
  1. Self-calibration and biconvex compressive sensing. *Inverse Problems* 31 (11), 115002
  2. Blind deconvolution meets blind demixing: algorithms and performance bounds, *To appear in IEEE Trans on Information Theory*
  3. Rapid, robust, and reliable blind deconvolution via nonconvex optimization, *arXiv:1606.04933*.
  4. Regularized gradient descent: a nonconvex recipe for fast joint blind deconvolution and demixing *arXiv:1703.08642*.

MRI imaging deblurring:

When the subspace $\boldsymbol{B}$ or support of blurring kernel is unknown:
we assume the support of blurring kernel is contained in a small box;
$N = 35000$.







Two-page proof

Condition 1 + 2 $\Longrightarrow$ Linear convergence of $\widetilde{F}$

Proof.

Let $\mathbf{z}_{t+1} = \mathbf{z}_t - \eta\nabla\widetilde{F}(\mathbf{z}_t)$ with $\eta \leq \frac{1}{C_L}$. By using modified descent lemma,

$$\begin{aligned}\widetilde{F}(\mathbf{z}_t + \eta\nabla\widetilde{F}(\mathbf{z}_t)) &\leq \widetilde{F}(\mathbf{z}_t) - (2\eta + C_L\eta^2)\|\nabla\widetilde{F}(\mathbf{z}_t)\|^2 \\ &\leq \widetilde{F}(\mathbf{z}_t) - \eta\omega\widetilde{F}(\mathbf{z}_t)\end{aligned}$$

which gives $\widetilde{F}(\mathbf{z}_{t+1}) \leq (1-\eta\omega)^t\widetilde{F}(\mathbf{z}_0)$. $\square$

Two-page proof: continued

Condition 3 $\Longrightarrow$ Linear convergence of $\|\boldsymbol{u}_t\boldsymbol{v}_t^* - \boldsymbol{h}_0\boldsymbol{x}_0^*\|_F$.

It follows from $\widetilde{F}(\boldsymbol{z}_t) \geq F(\boldsymbol{z}_t) \geq \frac{3}{4}\|\boldsymbol{u}_t\boldsymbol{v}_t^* - \boldsymbol{h}_0\boldsymbol{x}_0^*\|_F^2$. Hence, linear convergence of objective function also implies linear convergence of iterates.

Condition 4 $\Longrightarrow$ Proof of stability theory

If L is sufficiently large, $\mathcal{A}^*(\boldsymbol{w})$ is small since $\|\mathcal{A}^*(\boldsymbol{w})\| \to 0$. There holds

$$\|\mathcal{A}(\boldsymbol{h}\boldsymbol{x}^* - \boldsymbol{h}_0\boldsymbol{x}_0^*) - \boldsymbol{w}\|^2 \approx \|\mathcal{A}(\boldsymbol{h}\boldsymbol{x}^* - \boldsymbol{h}_0\boldsymbol{x}_0^*)\|^2 + \sigma^2 d_0^2.$$

Hence, the objective function behaves "almost like" $\|\mathcal{A}(\boldsymbol{h}\boldsymbol{x}^* - \boldsymbol{h}_0\boldsymbol{x}_0^*)\|^2$, the noiseless version of F if the sample size is sufficiently large.