Joint Blind Deconvolution and Blind Demixing via Nonconvex Optimization

Shuyang Ling

Department of Mathematics, UC Davis

July 19, 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) Blind deconvolution meets blind demixing: applications in image processing and wireless communication

(b) Mathematical models and convex approach

(c) A nonconvex optimization approach towards joint blind deconvolution and blind demixing

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 $\boldsymbol{g}$?

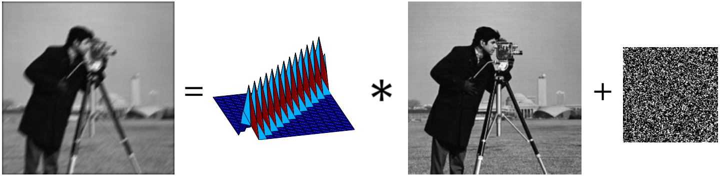
Why do we care about blind deconvolution?

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.

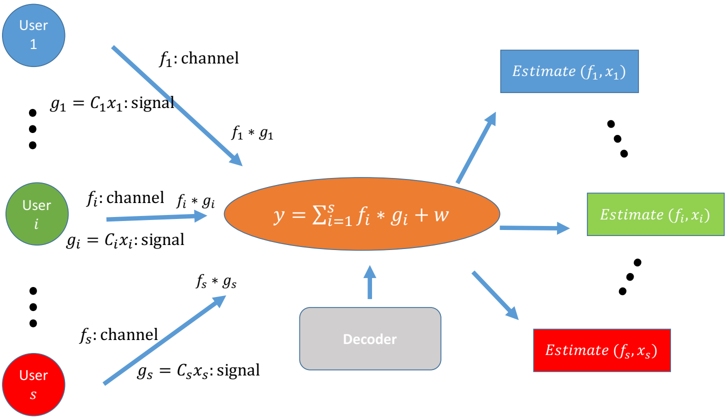
Question: how to reconstruct $\boldsymbol{f}$ and $\boldsymbol{g}$ from $\boldsymbol{y}$

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



Blind deconvolution meets 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 $\boldsymbol{f}_i$.



Blind deconvolution and blind demixing

Consider the model:

$$\boldsymbol{y} = \sum_{i=1}^{s} \boldsymbol{f}_i * \boldsymbol{g}_i + \boldsymbol{w}.$$

This is even more difficult than blind deconvolution ($s = 1$), since this is a "mixture" of blind deconvolution problems. It also includes phase retrieval as a special case if $s = 1$ and $\bar{\boldsymbol{g}}_i = \boldsymbol{f}_i$.

More assumptions

- Each impulse response $\boldsymbol{f}_i$ has maximum delay spread K (compact support):

$$\boldsymbol{f}_i(n) = 0, \quad \text{for } n > K, \quad \boldsymbol{f}_i = \begin{bmatrix} \boldsymbol{h}_i \\ 0 \end{bmatrix}.$$

- Let $\boldsymbol{g}_i := \boldsymbol{C}_i \boldsymbol{x}_i$ be the signal $\boldsymbol{x}_i \in \mathbb{C}^N$ encoded by $\boldsymbol{C}_i \in \mathbb{C}^{L \times N}$ with $L > N$. We also require $\boldsymbol{C}_i$ to be mutually incoherent by imposing randomness.

Mathematical model

Subspace assumption on the frequency domain

Denote $\boldsymbol{F}$ as the $L \times L$ DFT matrix.

- Let $\boldsymbol{h}_i \in \mathbb{C}^K$ be the first K nonzero entries of $\boldsymbol{f}_i$ and $\boldsymbol{B}$ be a low-frequency DFT matrix. There holds, $\hat{\boldsymbol{f}}_i = \boldsymbol{F}\boldsymbol{f}_i = \boldsymbol{B}\boldsymbol{h}_i$.
- Let $\hat{\boldsymbol{g}}_i := \boldsymbol{A}_i \boldsymbol{x}_i$ where $\boldsymbol{A}_i := \boldsymbol{F}\boldsymbol{C}_i$ and $\boldsymbol{x}_i \in \mathbb{C}^N$.

Mathematical model

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

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

Mathematical model

Subspace assumption on the frequency domain

Denote $\boldsymbol{F}$ as the $L \times L$ DFT matrix.

- Let $\boldsymbol{h}_i \in \mathbb{C}^K$ be the first K nonzero entries of $\boldsymbol{f}_i$ and $\boldsymbol{B}$ be a low-frequency DFT matrix. There holds, $\hat{\boldsymbol{f}}_i = \boldsymbol{F}\boldsymbol{f}_i = \boldsymbol{B}\boldsymbol{h}_i$.
- Let $\hat{\boldsymbol{g}}_i := \boldsymbol{A}_i\boldsymbol{x}_i$ where $\boldsymbol{A}_i := \boldsymbol{F}\boldsymbol{C}_i$ and $\boldsymbol{x}_i \in \mathbb{C}^N$.

Mathematical model

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

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

Naive approach

Nonlinear least squares

We may want to try nonlinear least squares approach:

$$\min_{(\boldsymbol{h}_i,\boldsymbol{x}_i)} \underbrace{\left\| \sum_{i=1}^{s} \operatorname{diag}(\boldsymbol{B}\boldsymbol{h}_i)\boldsymbol{A}_i\boldsymbol{x}_i - \boldsymbol{y} \right\|^2}_{F(\boldsymbol{h}_i,\boldsymbol{x}_i)}.$$

- The objective function is highly nonconvex and more complicated than blind deconvolution ($s = 1$).
- Gradient descent might get stuck at local minima.
- No guarantees for recoverability.

Naive approach

Nonlinear least squares

We may want to try nonlinear least squares approach:

$$\min_{(\boldsymbol{h}_i,\boldsymbol{x}_i)} \underbrace{\left\| \sum_{i=1}^{s} \operatorname{diag}(\boldsymbol{B}\boldsymbol{h}_i)\boldsymbol{A}_i\boldsymbol{x}_i - \boldsymbol{y} \right\|^2}_{F(\boldsymbol{h}_i,\boldsymbol{x}_i)}.$$

- The objective function is highly nonconvex and more complicated than blind deconvolution ($s = 1$).
- Gradient descent might get stuck at local minima.
- No guarantees for recoverability.

Convex relaxation and low-rank matrix recovery

Lifting

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

$$y_l = \sum_{i=1}^{s} (\boldsymbol{B}\boldsymbol{h}_i)_l \cdot (\boldsymbol{A}_l \boldsymbol{x}_i)_l = \sum_{i=1}^{s} \boldsymbol{b}_l^* \underbrace{\boldsymbol{h}_i \boldsymbol{x}_i^*}_{\text{rank-1}} \boldsymbol{a}_{i,l}.$$

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

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

Then, there holds $\boldsymbol{y} = \sum_{i=1}^{s} \mathcal{A}_i(\boldsymbol{X}_i) + \boldsymbol{w}$.

See [Candès-Strohmer-Voroninski 13], [Ahmed-Recht-Romberg, 14].

Convex relaxation and low-rank matrix recovery

Rank-s matrix recovery

We rewrite $\boldsymbol{y} = \sum_{i=1}^{s} \operatorname{diag}(\boldsymbol{B}\boldsymbol{h}_i)\boldsymbol{A}_i\boldsymbol{x}_i$ as

$$y_l = \left\langle \underbrace{\begin{bmatrix} \boldsymbol{h}_1\boldsymbol{x}_1^* & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{h}_2\boldsymbol{x}_2^* & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{h}_s\boldsymbol{x}_s^* \end{bmatrix}}_{\text{rank-}s\text{ matrix}}, \begin{bmatrix} \boldsymbol{b}_l\boldsymbol{a}_{1,l}^* & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{b}_l\boldsymbol{a}_{2,l}^* & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{b}_l\boldsymbol{a}_{s,l}^* \end{bmatrix} \right\rangle$$

- Recover a rank-s block diagonal matrix satisfying convex constraints.
- Finding such a rank-s matrix is generally an NP-hard problem.

Nuclear norm minimization

The ground truth is a rank-s block-diagonal matrix. It is natural to recover the solution via solving

$$\min \sum_{i=1}^{s} \|\boldsymbol{Z}_i\|_* \quad \text{subject to} \quad \sum_{i=1}^{s} \mathcal{A}_i(\boldsymbol{Z}_i) = \boldsymbol{y}$$

where $\sum_{i=1}^{s} \|\boldsymbol{Z}_i\|_*$ is the nuclear norm of $\mathrm{blkdiag}(\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_s)$.

Question: Can we recover $\{\boldsymbol{h}_{i0}\boldsymbol{x}_{i0}^*\}_{i=1}^{s}$ exactly?

Convex approach

Theorem

Assume that

- Let $\boldsymbol{B} \in \mathbb{C}^{L\times K}$ be a partial DFT matrix with $\boldsymbol{B}^*\boldsymbol{B} = \boldsymbol{I}_K$;
- Each $\boldsymbol{A}_i$ is a Gaussian random matrix.

The SDP relaxation is able to recover $\{(\boldsymbol{h}_{i0}, \boldsymbol{x}_{i0})\}_{i=1}^s$ exactly with probability at least $1 - \mathcal{O}(L^{-\gamma})$. Here the number of measurements L satifies

- [Ling-Strohmer 15] $L \geq C_\gamma s^2(K + \mu_h^2 N)\log^3 L$;
- [Jung-Krahmer-Stöger 17] $L \geq C_\gamma(s(K + \mu_h^2 N))\log^3 L$

where $\mu_h^2 = L\max_{1\leq i\leq s} \frac{\|\boldsymbol{B}\boldsymbol{h}_{i0}\|_\infty^2}{\|\boldsymbol{h}_{i0}\|^2}$.

- We can jointly estimate the channels and signals for s users with one simple convex program.
- SDP is able to recover $\{(\boldsymbol{h}_i, \boldsymbol{x}_i)\}_{i=1}^s$ but it is computationally expensive.

Convex approach

Theorem

Assume that

- Let $\boldsymbol{B} \in \mathbb{C}^{L\times K}$ be a partial DFT matrix with $\boldsymbol{B}^*\boldsymbol{B} = \boldsymbol{I}_K$;
- Each $\boldsymbol{A}_i$ is a Gaussian random matrix.

The SDP relaxation is able to recover $\{(\boldsymbol{h}_{i0}, \boldsymbol{x}_{i0})\}_{i=1}^s$ exactly with probability at least $1 - \mathcal{O}(L^{-\gamma})$. Here the number of measurements L satifies

- [Ling-Strohmer 15] $L \geq C_\gamma s^2 (K + \mu_h^2 N)\log^3 L$;
- [Jung-Krahmer-Stöger 17] $L \geq C_\gamma (s(K + \mu_h^2 N))\log^3 L$

where $\mu_h^2 = L \max_{1\leq i\leq s} \frac{\|\boldsymbol{B}\boldsymbol{h}_{i0}\|_\infty^2}{\|\boldsymbol{h}_{i0}\|^2}$.

- We can jointly estimate the channels and signals for s users with one simple convex program.
- SDP is able to recover $\{(\boldsymbol{h}_i, \boldsymbol{x}_i)\}_{i=1}^s$ but it is computationally expensive.

A nonconvex optimization approach?

An increasing list of nonconvex approaches to various problems in machine learning and signal processing:

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

Two-step philosophy for provable nonconvex optimization

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

(b) Run gradient descent method.

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

Building *"the basin of attraction"*

The basin of attraction relies on the following three observations.

Observation 1: Unboundedness of solution

- If the pair $(\boldsymbol{h}_{i0}, \boldsymbol{x}_{i0})$ is a solution to $\boldsymbol{y} = \sum_{i=1}^{s} \mathrm{diag}(\boldsymbol{B}\boldsymbol{h}_{i0})\boldsymbol{A}_i\boldsymbol{x}_{i0}$, then so is the pair $(\alpha_i \boldsymbol{h}_{i0}, \alpha_i^{-1}\boldsymbol{x}_{i0})$ for any $\alpha_i \neq 0$.

- Thus the blind deconvolution problem always has infinitely many solutions of this type. We can recover $(\boldsymbol{h}_{i0}, \boldsymbol{x}_{i0})$ only up to a scalar.

- It is possible that $\|\boldsymbol{h}_i\| \gg \|\boldsymbol{x}_i\|$ (vice versa) while $\|\boldsymbol{h}_i\| \cdot \|\boldsymbol{x}_i\|$ is fixed. Hence we define $\mathcal{N}_{d_0}$ to balance $\|\boldsymbol{h}_i\|$ and $\|\boldsymbol{x}_i\|$:

$$\mathcal{N}_{d_0} := \{\{(\boldsymbol{h}_i, \boldsymbol{x}_i)\}_{i=1}^{s} : \|\boldsymbol{h}_i\| \leq 2\sqrt{d_{i0}}, \|\boldsymbol{x}_i\| \leq 2\sqrt{d_{i0}}\}.$$

where $d_{i0} = \|\boldsymbol{h}_{i0}\|\|\boldsymbol{x}_{i0}\|$.

Building *"the basin of attraction"*

Observation 2: Incoherence

Our numerical experiments have shown that the algorithm's performance depends on how much $\boldsymbol{b}_l$ (the rows of $\boldsymbol{B}$) and $\boldsymbol{h}_{i0}$ are correlated.

$$\mu_h^2 := \max_{1 \le i \le s} \frac{L\|\boldsymbol{B}\boldsymbol{h}_{i0}\|_\infty^2}{\|\boldsymbol{h}_{i0}\|^2}, \quad \textbf{the smaller } \mu_h\textbf{, the better.}$$

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

$$\mathcal{N}_\mu := \{\{\boldsymbol{h}_i\}_{i=1}^s : \sqrt{L}\|\boldsymbol{B}\boldsymbol{h}_i\|_\infty \le 4\mu\sqrt{d_{i0}}\}.$$

"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 $\{\boldsymbol{b}_l \boldsymbol{a}_{i,l}^*\}_{1 \le l \le L}$.

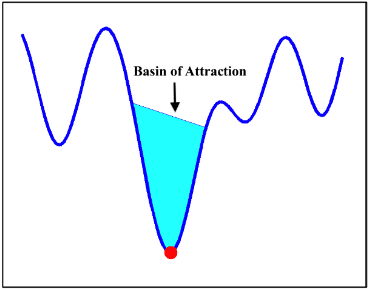
Building *"the basin of attraction"*

Observation 3: "Close" to the ground truth

We define $\mathcal{N}_\varepsilon$ to quantify closeness of $\{(\boldsymbol{h}_i, \boldsymbol{x}_i)\}_{i=1}^s$ to true solution, i.e.,

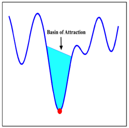
$$\mathcal{N}_\varepsilon := \{\{(\boldsymbol{h}_i, \boldsymbol{x}_i)\}_{i=1}^s : \|\boldsymbol{h}_i \boldsymbol{x}_i^* - \boldsymbol{h}_{i0} \boldsymbol{x}_{i0}^*\|_F \le \varepsilon d_{i0}\}.$$

We want to find an initial guess close to $\{(\boldsymbol{h}_{i0}, \boldsymbol{x}_{i0})\}_{i=1}^s$.



Building *"the basin of attraction"*

Based on the three observations above, we define the three neighborhoods:



The basin of attraction

The basin of attraction is the intersection of the following three sets $\mathcal{N}_{d_0} \cap \mathcal{N}_{\mu} \cap \mathcal{N}_{\varepsilon}$:

$$\mathcal{N}_{d_0} := \{\{(\boldsymbol{h}_i, \boldsymbol{x}_i)\}_{i=1}^{s} : \|\boldsymbol{h}_i\| \leq 2\sqrt{d_{i0}}, \|\boldsymbol{x}_i\| \leq 2\sqrt{d_{i0}}, 1 \leq i \leq s\}$$

$$\mathcal{N}_{\mu} := \{ \quad \{\boldsymbol{h}_i\}_{i=1}^{s} : \quad \sqrt{L}\|\boldsymbol{B}\boldsymbol{h}_i\|_\infty \leq 4\sqrt{d_{i0}}\mu, 1 \leq i \leq s\}$$

$$\mathcal{N}_{\varepsilon} := \left\{\{(\boldsymbol{h}_i, \boldsymbol{x}_i)\}_{i=1}^{s} : \frac{\|\boldsymbol{h}_i \boldsymbol{x}_i^* - \boldsymbol{h}_{i0}\boldsymbol{x}_{i0}^*\|_F}{d_{i0}} \leq \varepsilon, 1 \leq i \leq s\right\}$$

where $d_{i0} = \|\boldsymbol{h}_{i0}\|\|\boldsymbol{x}_{i0}\|$, μ is a parameter and $\mu \geq \mu_h$ and ε is a predetermined parameter in $(0, \frac{1}{15}]$.

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

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

Here $G_0(z) = \max\{z-1, 0\}^2$, $\rho \approx d^2$, $d \approx d_0$, $d_i \approx d_{i0}$ and $\mu \geq \mu_h$.

Note that

$$\mathcal{A}_i^*(\boldsymbol{y}) = \underbrace{\mathcal{A}_i^*\mathcal{A}_i(\boldsymbol{h}_{i0}\boldsymbol{x}_{i0}^*)}_{\mathbb{E}(\mathcal{A}_i^*\mathcal{A}_i(\boldsymbol{h}_{i0}\boldsymbol{x}_{i0}^*))=\boldsymbol{h}_{i0}\boldsymbol{x}_{i0}^*} + \underbrace{\mathcal{A}_i^*\left(\sum_{j\neq i}\mathcal{A}_j(\boldsymbol{h}_{j0}\boldsymbol{x}_{j0}^*)\right)}_{\text{with mean 0}}$$

The leading singular vectors of $\mathcal{A}_i^*(\boldsymbol{y})$ can approximate $(\boldsymbol{h}_{i0}, \boldsymbol{x}_{i0})$.

Step 1: Initialization via spectral method and projection:

1: **for** $i = 1, 2, \dots, s$ **do**
2: Compute $\mathcal{A}_i^*(\boldsymbol{y})$, (since $\mathbb{E}(\mathcal{A}_i^*(\boldsymbol{y})) = \boldsymbol{h}_{i0}\boldsymbol{x}_{i0}^*$);
3: $(d, \hat{\boldsymbol{h}}_{i0}, \hat{\boldsymbol{x}}_{i0}) = \text{svds}(\mathcal{A}_i^*(\boldsymbol{y}))$;
4: $\boldsymbol{u}_i^{(0)} := \mathcal{P}_{\mathcal{N}_\mu}(\sqrt{d_i}\hat{\boldsymbol{h}}_{i0})$ and $\boldsymbol{v}_i^{(0)} := \sqrt{d_i}\hat{\boldsymbol{x}}_{i0}$.
5: **end for**

Algorithm: Wirtinger gradient descent

Step 2: Gradient descent with constant stepsize η:

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

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. There hold:

- the initial guess $(\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)}) \in \frac{1}{\sqrt{3}}\mathcal{N}_{d_0} \bigcap \frac{1}{\sqrt{3}}\mathcal{N}_{\mu} \bigcap \mathcal{N}_{\frac{2\varepsilon}{5\sqrt{s}\kappa}}$,
- $\sqrt{\sum_{i=1}^{s} \|\boldsymbol{u}_i^{(t)}(\boldsymbol{v}_i^{(t)})^* - \boldsymbol{h}_{i0}\boldsymbol{x}_{i0}^*\|_F^2} \leq (1-\alpha)^t \varepsilon d_0 + c_0\sqrt{s}\|\mathcal{A}^*(\boldsymbol{w})\|$

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

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

where $\kappa = \frac{\max d_{i0}}{\min d_{i0}}$.

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. There hold:

- the initial guess $(\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)}) \in \frac{1}{\sqrt{3}}\mathcal{N}_{d_0} \bigcap \frac{1}{\sqrt{3}}\mathcal{N}_{\mu} \bigcap \mathcal{N}_{\frac{2\varepsilon}{5\sqrt{s}\kappa}}$,
- $\sqrt{\sum_{i=1}^{s} \|\boldsymbol{u}_i^{(t)}(\boldsymbol{v}_i^{(t)})^* - \boldsymbol{h}_{i0}\boldsymbol{x}_{i0}^*\|_F^2} \leq \underbrace{(1-\alpha)^t \varepsilon d_0}_{\text{linear convergence}} + \underbrace{c_0\sqrt{s}\|\mathcal{A}^*(\boldsymbol{w})\|}_{\text{error term}}$

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

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

where $\kappa = \frac{\max d_{i0}}{\min d_{i0}}$.

Remark

- The iterates $(\boldsymbol{u}_i^{(t)}, \boldsymbol{v}_i^{(t)})$ converges linearly to $(\boldsymbol{h}_{i0}, \boldsymbol{x}_{i0})$:

$$\|\boldsymbol{u}_i^{(\infty)}(\boldsymbol{v}_i^{(\infty)})^* - \boldsymbol{h}_{i0}\boldsymbol{x}_{i0}^*\|_F \leq c_0\sqrt{s}\|\mathcal{A}^*(\boldsymbol{w})\|$$

- $\|\mathcal{A}^*(\boldsymbol{w})\|$ converges to 0 with the rate of $\mathcal{O}(L^{-1/2})$:

$$\|\mathcal{A}^*(\boldsymbol{w})\| \leq C_0\sigma d_0\sqrt{\frac{s(K+N)(\log^2 L)}{L}}$$

Therefore, $(\boldsymbol{u}_i^{(\infty)}, \boldsymbol{v}_i^{(\infty)})$ is a consistent estimator of $(\boldsymbol{h}_{i0}, \boldsymbol{x}_{i0})$.

- Challenges: s^2 is not optimal. The optimal scaling should be $L = \mathcal{O}(s(K+N))$ instead of $L = \mathcal{O}(s^2(K+N))$.

Remark

- The iterates $(\boldsymbol{u}_i^{(t)}, \boldsymbol{v}_i^{(t)})$ converges linearly to $(\boldsymbol{h}_{i0}, \boldsymbol{x}_{i0})$:

$$\|\boldsymbol{u}_i^{(\infty)}(\boldsymbol{v}_i^{(\infty)})^* - \boldsymbol{h}_{i0}\boldsymbol{x}_{i0}^*\|_F \le c_0\sqrt{s}\|\mathcal{A}^*(\boldsymbol{w})\|$$

- $\|\mathcal{A}^*(\boldsymbol{w})\|$ converges to 0 with the rate of $\mathcal{O}(L^{-1/2})$:

$$\|\mathcal{A}^*(\boldsymbol{w})\| \le C_0 \sigma d_0 \sqrt{\frac{s(K+N)(\log^2 L)}{L}}$$

  Therefore, $(\boldsymbol{u}_i^{(\infty)}, \boldsymbol{v}_i^{(\infty)})$ is a consistent estimator of $(\boldsymbol{h}_{i0}, \boldsymbol{x}_{i0})$.

- Challenges: s^2 is not optimal. The optimal scaling should be $L = \mathcal{O}(s(K+N))$ instead of $L = \mathcal{O}(s^2(K+N))$.

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.

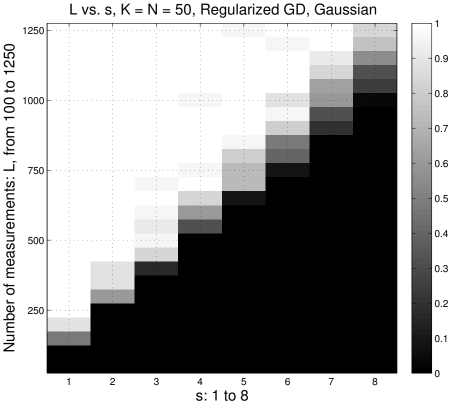
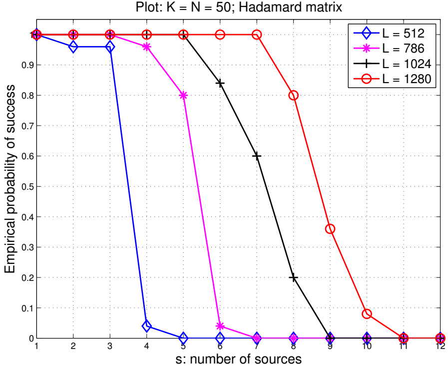


Figure: Black: failure; white: success

Back to the communication example

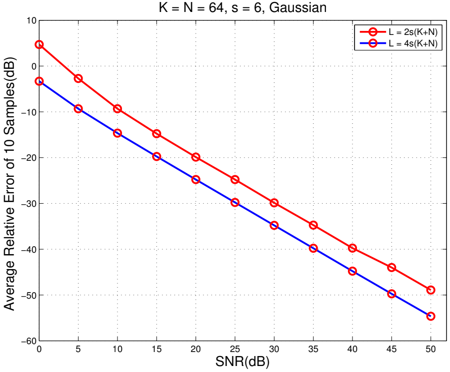
A more practical and useful choice of encoding matrix $\boldsymbol{C}_i$: $\boldsymbol{C}_i = \boldsymbol{D}_i \boldsymbol{H}$ (i.e., $\boldsymbol{A}_i = \boldsymbol{F}\boldsymbol{D}_i\boldsymbol{H}$) where $\boldsymbol{D}_i$ is a diagonal random binary ± 1 matrix and $\boldsymbol{H}$ is an $L \times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

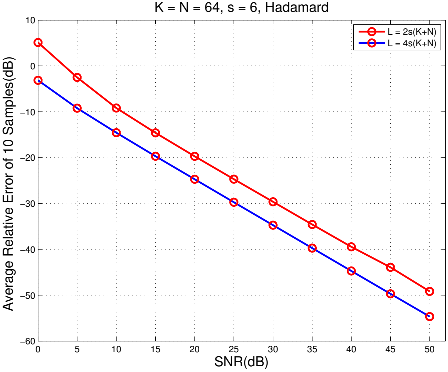


$L \approx 1.5s(K+N)$ yields exact recovery.

Numerics: robustness

We see that the relative error is linearly correlated with the noise in dB. Approximately, 10 units of increase in SNR leads to the *same* amount of decrease in relative error (in dB).





Outlook and Conclusion

Conclusion: The proposed algorithm is arguably the first blind deconvolution/blind demixing algorithm that is numerically efficient, robust against noise and comes with rigorous recovery guarantees under subspace conditions.

- Open problem: Does similar result hold for other types of $\boldsymbol{A}_i$?
- Open problem: what if either $\boldsymbol{h}_i$ or $\boldsymbol{x}_i$ is sparse?
- Major open problem in nonconvex optimization:

  How to remove the s^2-dependence for rank-s matrix recovery?

Outlook and Conclusion

Conclusion: The proposed algorithm is arguably the first blind deconvolution/blind demixing algorithm that is numerically efficient, robust against noise and comes with rigorous recovery guarantees under subspace conditions.

- Open problem: Does similar result hold for other types of $\boldsymbol{A}_i$?
- Open problem: what if either $\boldsymbol{h}_i$ or $\boldsymbol{x}_i$ is sparse?
- Major open problem in nonconvex optimization:

  How to remove the s^2-dependence for rank-s matrix recovery?