# ECE 273 Project: Blind Deconvolution

1st Wenyuan Zhao
*Dept. Electrical and Computer Engineering*
*University of California San Diego*
La Jolla, US
wez030@ucsd.edu

***Abstract*—We study the question of recovering two signals $\mathbf{w}$ and $\mathbf{x}$ from their convolution $\mathbf{y} = \mathbf{w} * \mathbf{x}$. Generally, the solution to this blind deconvolution problem is non-unique and non-convex. But with assumptions on sparsity, subspace structure and transformed variable, we can convert the non-convex nuclear norm into a convex problem by "dual-dual" relaxation. In this project, we also implement the convex algorithm proposed in *Blind Deconvolution Using Convex Programming*, and compare its performance with non-blind and non-convex algorithms. Moreover, the evaluation shows that the convex algorithm is robust against sparsity violation, but sensitive to low-rank condition. At last, we try to extend the algorithm to 2D deconvolution by recovering a blurred image. But the result on 2D deconvolution still need improvement.**

***Index Terms*—Blind deconvolution, convex programming, nuclear norm minimization, low-rank matrix**

## I. Project Objective

This project aims to recover two signals given the output of their convolution under suitable conditions based on convex algorithms proposed in [1], and evaluate theoretical guarantees and robustness of the proposed method.

## II. Background and Motivation

Blind deconvolution problem, arises in many areas of science and technology, including astronomy, medical imaging, optics, and communications engineering [3, 6, 7]. How to find a fast and reliable algorithm for blind deconvolution has confounded researchers for many decades. There exist many methods to solve this problem [1, 2, 5]. Inspired by what we learnt in ECE273 – Convex Optimization, we want to explore a convex programming method to solve blind deconvolution problem.

## III. Problem Definition

The general statement of the problem is as follows. Assume we observe the circular convolution of signals $\mathbf{w}$ and $\mathbf{w}$, which is

$$\mathbf{y} = \mathbf{w} * \mathbf{x}, \text{ or } y[l] = \sum_{l'=1}^{L} w[l']\, x[l - l' + 1]. \tag{1}$$

In general, the solution is not unique. Given one observation $\mathbf{y}$, there can be multiple convolution vector combinations. For example, $\mathbf{y} = \mathbf{w} * \mathbf{x} = \alpha\mathbf{w} * \alpha^{-1}\mathbf{x}$. We can choose any nonzero $\alpha$ to recover different solutions. On the other hand, the convolution $\mathbf{y}$ remains the same after $\mathbf{w}$ and $\mathbf{x}$ are circularly shifted, which will cause different solutions as well.

To make the solution unique, we have to make several assumptions on the structure of $\mathbf{w}$ and $\mathbf{x}$:

1) *Subspace Structure:* Suppose $\mathbf{w}$ and $\mathbf{x}$ has dimension $K$, $N$, respectively. Then $\mathbf{w}$ lives in a fixed subspace, while $\mathbf{x}$ lives in a "generic" subspace chosen at random. That is,
$$\begin{aligned} \mathbf{w} &= \mathbf{B}\mathbf{h},\ \mathbf{h} \in \mathbb{R}^K, \\ \mathbf{x} &= \mathbf{C}\mathbf{m},\ \mathbf{m} \in \mathbb{R}^N. \end{aligned} \tag{2}$$
Here $\mathbf{B}$ is a $L \times K$ matrix, and $\mathbf{C}$ is a $L \times N$ matrix. Then the original problem is equivalent to recovering $\mathbf{h}$ and $\mathbf{m}$, which lives in two subspaces of $\mathbb{R}^L$, respectively. Current results on recovering low-rank matrices from underdetermined linear observations show that $\mathbf{w}$ and $\mathbf{x}$ can be deconvoluted exactly.
2) *Sparsity:* The two signals $\mathbf{w}$ and $\mathbf{x}$ are sparse over the subspaces.
3) *Normalization:* Without loss of generality, let $\|\mathbf{h}\|_2 = \|\mathbf{m}\|_2 = 1$.

With the above constraints, the shifted or scalar-multiplied signals will not live in these subspaces, and then the problem can permit unique solutions. To recover the signals, we can break apart the convolution in (1) by expanding $\mathbf{x}$ as a linear combination of the columns $\mathbf{C}_1, ..., \mathbf{C}_N$ of $\mathbf{C}$,

$$\begin{aligned} \mathbf{y} &= m_1\mathbf{w} * \mathbf{C}_1 + ... + m_N\mathbf{w} * \mathbf{C}_N \\ &= \begin{bmatrix} \text{circ}(\mathbf{C}_1) & \cdots & \text{circ}(\mathbf{C}_N) \end{bmatrix} \begin{bmatrix} m_1\mathbf{w} \\ \vdots \\ m_N\mathbf{w} \end{bmatrix}. \end{aligned} \tag{3}$$

Similarly, by expanding $\mathbf{w}$ as a linear combination of the columns of $\mathbf{B}$, this becomes

$$\mathbf{y} = \begin{bmatrix} \text{circ}(\mathbf{C}_1)\mathbf{B} & \cdots & \text{circ}(\mathbf{C}_N)\mathbf{B} \end{bmatrix} \begin{bmatrix} m_1\mathbf{h} \\ \vdots \\ m_N\mathbf{h} \end{bmatrix}. \tag{4}$$

If we have the knowledge that $\mathbf{w}$ and $\mathbf{x}$ lie in the span of the columns of $\mathbf{B}$ and $\mathbf{C}$ respectively, they can be uniquely separated in many situations. For matrices $\mathbf{B}$ and $\mathbf{C}$, let the columns of these matrices provide bases for the subspaces where $\mathbf{w}$ and $\mathbf{x}$ live. Then recovering $\mathbf{h}$ and $\mathbf{m}$ is equivalent to recovering $\mathbf{w}$ and $\mathbf{x}$.

$R(B)$ and $R(C)$ are different with high probability, if $K$ and $N$ are much smaller than $L$m. And therefore we are more likely to separated them uniquely. More extremely, if $R(B) \cap R(C)$ is nullspace, we can easily separate them.

## IV. Method

### A. Fourier Transform

It is convenient to solve (4) in the Fourier domain. We leverage the L points discrete Fourier (DFT) matrix

$$\mathbf{F}(w,l) = \frac{1}{\sqrt{L}} e^{-j2\pi(w-1)(l-1)/L},\ 1 \leqslant w,l \leqslant L \tag{5}$$

to convert $\mathbf{y}$ into the Fourier domain

$$\hat{\mathbf{y}} = \mathbf{F}\mathbf{y} = \begin{bmatrix} \triangle_1 \hat{\mathbf{B}} & \cdots & \triangle_N \hat{\mathbf{B}} \end{bmatrix} \begin{bmatrix} m_1 \mathbf{h} \\ \vdots \\ m_N \mathbf{h} \end{bmatrix}. \tag{6}$$

Here $\hat{\mathbf{C}} = \mathbf{F}\mathbf{C}$, $\hat{\mathbf{B}} = \mathbf{F}\mathbf{B}$, $\text{circ}(\mathbf{C}_n) = \mathbf{F}^* \triangle_n \mathbf{F}$, and $\triangle_n = \text{diag}\left(\sqrt{L}\hat{\mathbf{C}}_\mathbf{n}\right)$. Hence, while $\mathbf{y}$ is a nonlinear combination of the coefficients $\mathbf{h}$ and $\mathbf{m}$, it is a linear combination of the entries of their outer product $\mathbf{X}_0 = \mathbf{h}\mathbf{m}^*$. Clearly, recovering $\mathbf{m}$ and $\mathbf{h}$ from $\hat{\mathbf{y}}$ is the same as recovering $\mathbf{x}$ and $\mathbf{w}$ from $\mathbf{y}$.

### B. Transformed Variable

We can express $\hat{\mathbf{y}}$ as a linear transformation $\mathcal{A}$ from $\mathbb{R}^{K\times N}$ to $\mathbb{R}^L$, which is $\hat{\mathbf{y}} = \mathcal{A}(\mathbf{X_0})$. To ensure $\mathcal{A}$ is invertible over all matrices, we need at least as many observations as unknowns, $L \geqslant NK$. With the assumption that $\mathbf{X}_0$ has the rank of 1, we will be able to recover it from $L \ll NK$ under certain conditions on $\mathcal{A}$. Let $\hat{\mathbf{b}}_l \in \mathbb{C}^K$ be the $L$th column of $\hat{\mathbf{B}}^*$, $\hat{\mathbf{c}}_l \in \mathbb{C}^N$ be the $l$th row of $\sqrt{L}\hat{\mathbf{C}}$. Then the entry of $\hat{\mathbf{y}}$ can be translated as trace inner products of different $K \times N$ matrices against $\mathbf{h}\mathbf{m}^*$,

$$\begin{aligned} \hat{\mathbf{y}}(l) &= \hat{c}_l(1)m(1)\left\langle \mathbf{h}, \hat{\mathbf{b}}_l \right\rangle + \ldots + \hat{c}_l(N)m(N)\left\langle \mathbf{h}, \hat{\mathbf{b}}_l \right\rangle \\ &= \langle \hat{\mathbf{c}}_l, \mathbf{m} \rangle \left\langle \mathbf{h}, \hat{\mathbf{b}}_l \right\rangle \\ &= \text{trace}\left(\mathbf{A}_l^*(\mathbf{h}\mathbf{m}^*)\right),\ \text{where } \mathbf{A}_l = \hat{\mathbf{b}}_l \hat{\mathbf{c}}_l^*. \end{aligned} \tag{7}$$

Now the deconvolution problem is transformed as a linear inverse problem over the rank-1 matrices. The transformed variable makes the problem linear. However, it is still a non-convex problem yet. We will introduce the convex relaxation to make the problem convex.

### C. Convex Relaxation

A natural way to recover $\mathbf{h}$ and $\mathbf{m}$ from measurements of form (6) is nuclear norm minimization, the sum of the singular values of a matrix, as a proxy for rank. Given $\hat{\mathbf{y}} \in \mathbb{C}^L$, our goal is to find $\mathbf{h} \in \mathbb{R}^K$, $\mathbf{m} \in \mathbb{R}^N$. We want to solve

$$\begin{aligned} &\min_{\mathbf{u},\mathbf{v}} \|\mathbf{u}\|_2^2 + \|\mathbf{v}\|_2^2 \\ &\text{subject to } \hat{\mathbf{y}}(l) = \langle \hat{\mathbf{c}}_l, \mathbf{u} \rangle \left\langle \mathbf{v}, \hat{\mathbf{b}}_l \right\rangle,\ l = 1, \ldots, L. \end{aligned} \tag{8}$$

This is a non-convex quadratic optimization problem. Although the cost function is convex, but the quadratic equality constraints mean that the feasible set is non-convex. With a convex objective but a non-convex constraint set, this is still a non-convex problem.

A standard approach to solving constraint quadratic is to use duality, which is the semi-definite program (SDP):

$$\begin{aligned} &\max_{\lambda} \Re\langle \hat{\mathbf{y}}, \lambda \rangle \\ &\text{subject to } \begin{bmatrix} \mathbf{I} & \sum_{l=1}^L \lambda(l)\mathbf{A}_l \\ \sum_{l=1}^L \lambda(l)^* \mathbf{A}_l^* & \mathbf{I} \end{bmatrix} \succcurlyeq 0, \end{aligned} \tag{9}$$

where $\mathbf{A}_l = \hat{\mathbf{b}}_l \hat{\mathbf{c}}_l^*$ defined as in previous section. Taking the dual again, we have the dual SDP of (9)

$$\begin{aligned} &\min_{\mathbf{W}_1, \mathbf{W}_2, X} \frac{1}{2}\text{trace}\left(\mathbf{W}_1\right) + \frac{1}{2}\text{trace}\left(\mathbf{W}_2\right) \\ &\text{subject to } \begin{bmatrix} \mathbf{W}_1 & \mathbf{X} \\ \mathbf{X}^* & \mathbf{W}_2 \end{bmatrix} \succcurlyeq 0,\ \hat{\mathbf{y}} = \mathcal{A}(\mathbf{X}), \end{aligned} \tag{10}$$

which is equivalent to

$$\begin{aligned} &\min \|\mathbf{X}\|_* \\ &\text{subject to } \hat{\mathbf{y}} = \mathcal{A}(\mathbf{X}). \end{aligned} \tag{11}$$

Now we have converted the non-convex least-squares estimation problem into a convex problem by the "dual-dual" relaxation of a nuclear norm. By optimizing $\|\mathbf{X}\|_*$, we can recover $\mathbf{h}$ and $\mathbf{m}$ from $\mathbf{X}_0$.

**Algorithm 1:** Blind Deconvolution Using Convexity

**Input:** $\mathbf{y}$, $\mathbf{B}$, $\mathbf{C}$
**Output:** $\hat{\mathbf{w}}$, $\hat{\mathbf{x}}$
Calculate Fourier transform $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$
**for** $l$ *in* $1:L$ **do**
  $\mathbf{A}_l = \hat{\mathbf{b}}_l \hat{\mathbf{c}}_l^*$, $\mathbf{X} = \sum_{l=1}^L \lambda(l)\mathbf{A}_l$
**end**
**cvx_begin**
  $\min \|\mathbf{X}\|_*$
  subject to $\hat{\mathbf{y}} = \mathcal{A}(\mathbf{X})$
**cvx_end**
$[U, \Sigma, V] = svd(\mathbf{X})$
**Return:** $\hat{\mathbf{w}} = \hat{\mathbf{B}}\mathbf{h}$, $\hat{\mathbf{x}} = \hat{\mathbf{C}}\mathbf{m}$

### D. Obtain the Constituent Signals

By solving (11), we can obtain $\mathbf{X}_0 = \mathbf{h}\mathbf{m}^*$. To recover $\mathbf{h}$ and $\mathbf{m}$ from $\mathbf{X}_0$, we can follow the outer product form of singular value decomposition (SVD) proposed in [4]. Recall that the SVD of matrix $\mathbf{A}$ can be expressed using partitioned

matrices as follows:

$$\begin{aligned}\mathbf{A} &= [u_1 \ \cdots \ u_k \ u_{k+1} \ \cdots \ u_m] \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_k & \\ & 0 & & 0 \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_k^T \\ v_{k+1}^T \\ \vdots \\ v_n^T \end{bmatrix} \\ &= \begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_k^T \end{bmatrix} \\ &+ \begin{bmatrix} u_{k+1} & \cdots & u_m \end{bmatrix} [0] \begin{bmatrix} v_{k+1}^T \\ \vdots \\ v_n^T \end{bmatrix} \\ &= \begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_k^T \end{bmatrix}. \end{aligned} \tag{12}$$

Hence, $\mathbf{A} = \mathbf{XY}$ can be expressed as an outer product expansion: $\mathbf{A} = \sum_{i=1}^{k} \sigma_i u_i v_i^T$. Therefore, given $\mathbf{X}_0 = \mathbf{hm}^*$, we can recover $\mathbf{h}$ and $\mathbf{m}$ by doing singular value decomposition of $\mathbf{X}_0 = \mathbf{U\Sigma V^H}$. Then recovered $\mathbf{h}$ and $\mathbf{m}$ are the first column of $\mathbf{U}$ and $\mathbf{V}$, respectively.

### E. Theoretical Guarantees

Basically, the convex relaxation in IV-C is not deterministic. We can guarantee the convex algorithm for a relatively large subspace dimensions $K$ and $N$ when $\mathbf{B}$ is incoherent in the Fourier domain, and when $\mathbf{C}$ is generic. Before clarifying our guarantees, the convex algorithm are affected by several conditions:

1) $\mathbf{w}$ is time limited to $Q$ ($K \leqslant Q \leqslant L$), which implies that the last $L - Q$ rows of $B$ are zero.
2) *Incoherence parameter:* Without loss of generality, columns of $\mathbf{B}$ are orthonormal, that is,

$$\mathbf{B}^*\mathbf{B} = \hat{\mathbf{B}}^*\hat{\mathbf{B}} = \sum_{l=1}^{L} \hat{\mathbf{b}}_l \hat{\mathbf{b}}_l^* = \mathbf{I}. \tag{13}$$

Then incoherence parameter $\mu_{\min}$ and $\mu_{\max}$ are defined as the degree to which the columns of $\mathbf{B}$ are jointly concentrated in the Fourier domain:

$$\mu_{\min}^2 = \frac{L}{K} \min_{1 \leqslant l \leqslant L} \|\hat{\mathbf{b}}_l\|_2^2, \ \mu_{\max}^2 = \frac{L}{K} \max_{1 \leqslant l \leqslant L} \|\hat{\mathbf{b}}_l\|_2^2.$$

We will always have $0 \leqslant \mu_{\min}^2 \leqslant 1$ and $\mu_{\min}^2 \leqslant \mu_{\max}^2$.
3) *Diffusion:* Our analytic results depend on how diffuse the particular signal we are trying to recover $\mathbf{w} = \mathbf{Bh}$ is in the Fourier domain. The diffusion parameter $\mu_h$ is given by

$$\mu_h^2 = L \max_{1 \leqslant l \leqslant L} |\hat{w}(l)|^2 = L \max_{1 \leqslant l \leqslant L} \left| \left\langle \mathbf{h}, \hat{\mathbf{b}}_l \right\rangle \right|^2. \tag{14}$$

Applying Cauchy-Schwartz inequality, and by the assumption $\|\mathbf{h}\|_2 = 1$ in III, we have

$$\mu_h^2 \leqslant L \max_{1 \leqslant l \leqslant L} \|\hat{\mathbf{b}}_l\|_2^2 \|\mathbf{h}\|_2^2 \leqslant K\mu_{\max}^2. \tag{15}$$

Meanwhile, take a summation over $l$ on both sides:

$$\sum_{l=1}^{L} \mu_h^2 = L \sum_{l=1}^{L} \max_{1 \leqslant l \leqslant L} \left| \left\langle \mathbf{h}, \hat{\mathbf{b}}_l \right\rangle \right|^2, \tag{16}$$

which implies $\mu_h^2 \geqslant \sum_{l=1}^{L} \left| \left\langle \mathbf{h}, \hat{\mathbf{b}}_l \right\rangle \right|^2 = \|\mathbf{h}\|_2^2 = 1$. Therefore, the diffusion parameter $\mu_h$ is bounded by

$$1 \leqslant \mu_h^2 \leqslant K\mu_{\max}^2.$$

4) *Generic subspace:* We will take the entries of $\mathbf{C}$ to be independent and identically distributed random variables $C[l, n] \sim \text{Normal}\left(0, L^{-1}\right)$, and the rows $\hat{\mathbf{c}}_l$ of $\mathbf{C}$ are *independent* – this allows us to apply recently developed tools for estimating the spectral norm of a sum of independent random linear operators.

Now we can give the theoretical guarantees of the convex algorithm: Take two random normalized vectors $\mathbf{h} \in \mathbb{R}^K$ and $\mathbf{m} \in \mathbb{R}^N$. Let $\mathbf{B}$, $\mathbf{C}$, $\mu_{\max}^2$ and $\mu_h^2$ satisfy the above conditions. Fix $\alpha > 1$. Let $\mathbf{B}$ is a deterministic $L \times K$ matrix satisfying (13) and the last $L - Q$ rows of $\mathbf{B}$ are zero based on the assumptions. Then there exists a constant $C_\alpha = O(\alpha)$ depending only on $\alpha$, such that

$$\max\left(\mu_{\max}^2 K, \mu_h^2 N\right) \leqslant \frac{L}{C_\alpha \log^3 L}. \tag{17}$$

Then $\mathbf{X}_0 = \mathbf{hm}^*$ is the unique solution to (11) with probability $1 - O(L^{-\alpha+1})$, and we can recover both $\mathbf{w}$ and $\mathbf{x}$ (within a scalar multiple) from $\mathbf{y} = \mathbf{w} * \mathbf{x}$. Therefore, this is a *probabilistic* guarantee.

## V. Experiments and Results

An important guarantee in theory is $L/\log L \gtrsim N + K$. Hence, it is necessary to explore the probability of successful recovery against the number of measurements $L/(K + N)$.

### A. Convex Programming

We implement the convex algorithm and recreate the results in [1]. Fig. 1 shows the success rate of the convex algorithm. Due to the limitation of time and computational capacity, we reduce $L$ to $2^8(256)$, and $K, N$ range from 5 to 125 with step-size of 5.

We can observe a sharp region around which the probability changes from close to zero, to close to 1. In both cases, we are able to deconvolve this signals with a high rate of success when $L \gtrsim 2.5(K + N)$, which is close to the theoretical guarantee $L \gtrsim \log L(K + N) = 2.4(K + N)$.

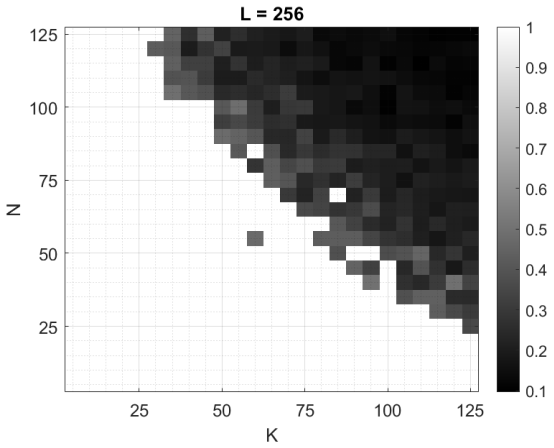


Fig. 1. Empirical success rate for the deconvolution of two vectors $\mathbf{w}$ and $\mathbf{x}$. In this experiment, $\mathbf{x}$ is a random vector in the subspace spanned by the columns of an $L \times N$ matrix whose entries are independent and identically distributed Gaussian random variables. $\mathbf{w}$ is a generic sparse vector, with support and nonzero entries chosen randomly.

### B. Comparison: Blind vs. Non-blind

Non-blind deconvolution refers to the case that one of the constituent signals is known. Then the problem can be expressed as least-square:

$$\arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{w} * \mathbf{x}\|_2 \tag{18}$$

We compare performance of the blind deconvolution algorithm against non-blind deconvolution by evaluating their success rate with different number of measurements. Fig. 2 shows the performance of blind and non-blind deconvolution. When the observed signal $\mathbf{y}$ is small, the non-blind algorithm has best performance. When the observed signal is larger, the blind algorithm shows better performance to recover $\mathbf{w}$ and $\mathbf{x}$ correctly. If $L \geqslant 2.1(K+N)$, we better choose blind algorithm even if we had known one of the signals.

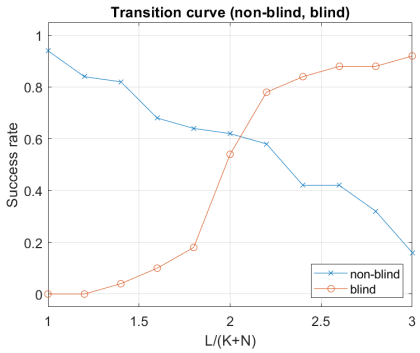


Fig. 2. Calculate the probability of successful recovery under different $L/(K+N)$. In this experiment, fix $K = N = 16$, and let the length $L$ varies from $K+N$ to $3(K+N)$.

### C. Comparison: Convex vs. Non-convex

We compared the convex algorithm against the non-convex algorithm in [5]. The non-convex algorithm follows the same thought that convert the problem into Fourier domain. Chosen linear operator $\mathcal{A}(\mathbf{X}) = \{\mathbf{b}_l^* \mathbf{X} \mathbf{a}_l\}_{l=1}^L$, the non-convex algorithm turns the problem as a least square problem:

$$\min_{\mathbf{h},\mathbf{x}} \|\mathcal{A}(\mathbf{h}\mathbf{x}^* - \mathbf{h}_0\mathbf{x}_0^*)\|^2 \tag{19}$$

The problem (19) can be solved by Wirtinger gradient descent method which avoids local minima. Implementation can follow Alg. 2 in [5].

Fig. 3 shows the probability of successful recovery using convex and non-convex algorithm. Generally, the non-convex algorithm has higher success rate given specific $L/(K+N)$. However, the non-convex algorithm costs more time to recover the constituent signals.

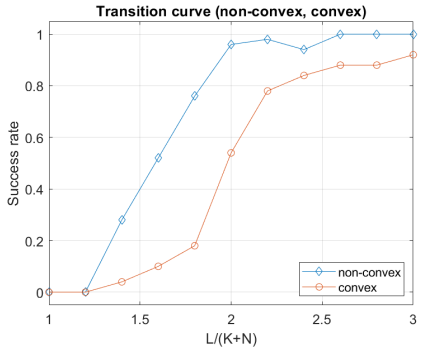


Fig. 3. Calculate the probability of successful recovery under different $L/(K+N)$. In this experiment, fix $K = N = 16$, and let the length $L$ varies from $K+N$ to $3(K+N)$.

### D. Robustness against Sparsity

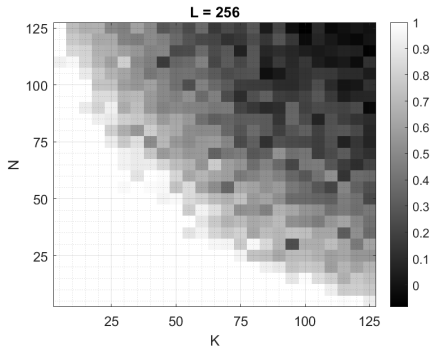


Fig. 4. Empirical success rate for the deconvolution of two vectors $\mathbf{w}$ and $\mathbf{x}$. In this experiment, $\mathbf{x}$ is a random vector in the subspace spanned by the columns of an $L \times N$ matrix whose entries are independent and identically distributed Gaussian random variables. $\mathbf{w}$ is a generic dense vector, with support and nonzero entries chosen randomly.

First, we violate one condition: sparsity. Fig. 4 shows the empirical success rate for the deconvolution of two dense vectors $\mathbf{w}$ and $\mathbf{x}$. Generally, the success rate is not heavily affected. To better illustrate the robustness of the convex algorithm against sparsity, we run the same experiment under four different conditions: dense $\mathbf{B}$ and $\mathbf{C}$, dense $\mathbf{B}$ and sparse $\mathbf{C}$, sparse $\mathbf{B}$ and $\mathbf{C}$, sparse $\mathbf{B}$ and dense $\mathbf{C}$.

Fig. 5 illustrates the success rate when violating sparsity conditions in four cases. It shows that the success rate will not be affected too much by the violation. When the observation is not large enough ($L/(K+N) \leqslant 2.5$), the performance is better on sparse $\mathbf{B}$ and $\mathbf{C}$. But with large enough observations, the sparsity condition does not affect the performance of convex algorithm. This also consistent with our theoretical guarantee. From the above results, we can say that the convex algorithm is robust against sparsity.

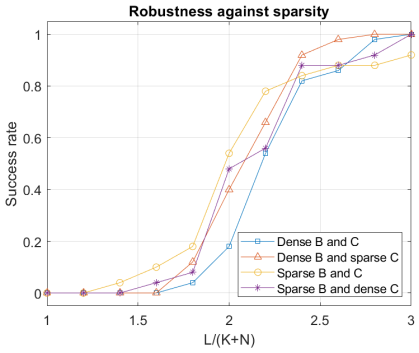


Fig. 5. Calculate the probability of successful recovery under different $L/(K+N)$. In this experiment, fix $K = N = 16$, and let the length $L$ varies from $K+N$ to $3(K+N)$.

### E. Robustness against Low-rank

In our problem set up, we claim that $\mathbf{X_0} = \mathbf{hm}^*$ has low-rand, namely rank-1. In this experiment, Fig. 1 can also show the robustness against low-rank condition. Given an observation $\mathbf{y}$ in length $L$, the probability of successful recovery is higher when the dimension of $\mathbf{w}$ and $\mathbf{x}$ is lower. When the dimension of $\mathbf{w}$ and $\mathbf{x}$ grows, i.e. $X_0$ more likely violates low-rank, the success rate decreases sharply. In addition, there is a sharp region around which the probability changes from close to zero, to close to 1. However, if $L/(K+N)$ is fixed, the algorithm will show robustness no matter how $K$ and $N$ are changed. Hence, the convex algorithm is not so robust against low-rank. A reliable blind deconvolution using convex algorithm still requires strict low-rank condition.

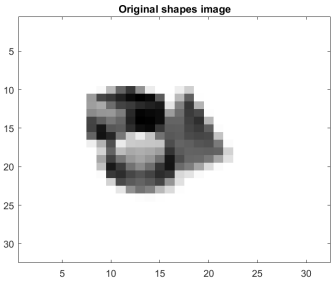


Fig. 6. Original 32x32 shapes image.

## VI. Extra: Blind Deconvolution in 2D

We extend the algorithm to perform blind deconvolution in 2D. We selected the same image in [1] and blurred it with a suitable kernel. Since my laptop is only equipped with 8GB memory, I reduce the size of the raw image to 32x32 as shown in Figure 6. Other cases will cause "out of memory" problem.

The generation of kernel and image blurring follows the work provided by CACTuS-AI. Fig. 7 and 8 shows the shape of the blur kernel and the output image after being blurred.

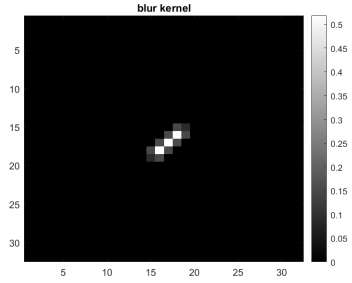


Fig. 7. Selected kernel to blur the image.

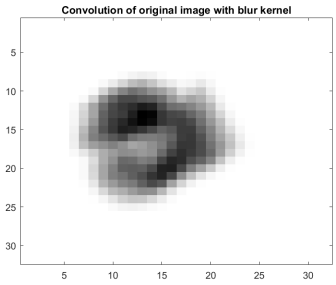


Fig. 8. Blurred image.

2D deconvolution using CVX is similar to 1D deconvolution. Inspired from CACTuS-AI, we find the wavelet coefficients of the target from the blurred image. However, compared with Fig. 9, the recovered image (Fig. 10 from my program is not so good.

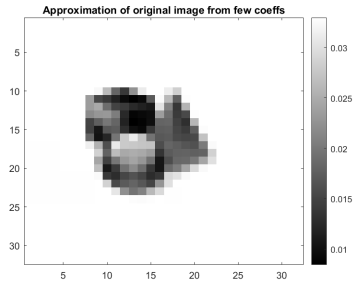


Fig. 9. Approximation of original image from few coefficients.

## VII. Conclusion

In conclusion, we implement the convex algorithm proposed in [1] and compare it with non-blind and non-convex approaches. Compared with non-blind deconvolution, convex algorithm performs better when the number of measurements is

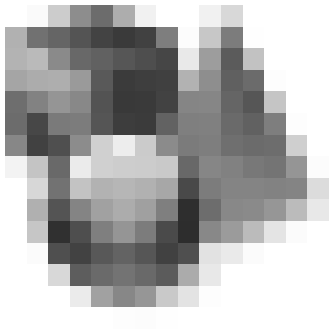

Fig. 10. Deconvolution in 2D using convex programming.

larger, while the non-blind deconvolution approach could stuck in local minima. Compared with the nonconvex approach, convex algorithm requires a larger number measurements than non-convex approach with the same accuracy. However, convex algorithm converges faster. Based on our evaluations, convex algorithm is robust against sparsity violation, but is sensitive to low-rank conditions. At last, we try to extend the algorithm to 2D deconvolution. But experimental results are not so good on recovering blurred image. Due to the limitations in computational capacity, we cannot implement the algorithm in a large scale. Hence, there are still a lot of things to explore and improve.

## DATA SET

The experiment in 1D does not require any data set. The data is generated randomly. The experiment in 2D uses the same picture in [1]. To reduce our computation burden, we change the size of the picture to 32x32.

## REFERENCES


[1] Ali Ahmed, Benjamin Recht, and Justin Romberg. Blind deconvolution using convex programming. *IEEE Transactions on Information Theory*, 60(3):1711–1732, 2013.

[2] Horacio E Fortunato and Manuel M Oliveira. Fast high-quality non-blind deconvolution using sparse adaptive priors. *The Visual Computer*, 30(6):661–671, 2014.

[3] Stuart M Jefferies and Julian C Christou. Restoration of astronomical images by iterative blind deconvolution. *The Astrophysical Journal*, 415:862, 1993.

[4] Dan Kalman. A singularly valuable decomposition: the svd of a matrix. *The college mathematics journal*, 27(1):2–23, 1996.

[5] Xiaodong Li, Shuyang Ling, Thomas Strohmer, and Ke Wei. Rapid, robust, and reliable blind deconvolution via nonconvex optimization. *Applied and computational harmonic analysis*, 47(3):893–934, 2019.

[6] Lang Tong, Guanghan Xu, and Thomas Kailath. Blind identification and equalization based on second-order statistics: A time domain approach. *IEEE Transactions on information Theory*, 40(2):340–349, 1994.

[7] Gerhard Wunder, Holger Boche, Thomas Strohmer, and Peter Jung. Sparse signal processing concepts for efficient 5g system design. *IEEE Access*, 3:195–208, 2015.