

# Generalized Blind Deconvolution

Qingyun Sun      Stephen Boyd

April 5, 2017

## 1 Introduction

Blind Deconvolution is the problem of recovering a kernel and a signal from their convolution. This is an ubiquitous problem in fields including signal processing, machine learning, communication, seismology, computer vision, microscopy, neural science.

## 2 Generalized blind deconvolution

### 2.1 Problem formulation

Consider that we have an observation vector  $y \in \mathbf{R}^n$ , and we want to look for a filter  $w \in \mathcal{W} \subset \mathbf{R}^n$  such that the circular convolution  $w * y \in \mathbf{R}^n$  is close to a signal  $x \in \mathcal{X} \subset \mathbf{R}^n$  that is simple under certain measure, for example, sparsity. More precisely, we can write

$$w * y \approx x + \epsilon,$$

where  $\epsilon$  is the noise or error of approximation.

We need to normalize  $w$ , otherwise  $w$  equals zero will result in a trivial signal. We impose that  $\mathcal{W}$  does not contain zero, and if  $w \in \mathcal{W}$ , then  $cw \in \mathcal{W}$  imply  $c = \pm 1$ .

Mathematically, given an observation  $y \in \mathbf{R}^n$ , we want to find a kernel  $w \in \mathcal{W}$ , and a signal  $x \in \mathcal{X}$  such that

$$\begin{aligned} & \text{minimize} && l(x - y * w) + r(x) + s(w) \\ & \text{subject to} && w \in \mathcal{W}, \quad x \in \mathcal{X} \end{aligned} \tag{1}$$

with variable  $w, x$  in subsets  $\mathcal{W}, \mathcal{X}$  of  $\mathbf{R}^n$ .  $l$  is the loss function that measure how well the approximation is.  $r$  is the regularizer on signal,  $s$  is the regularizer on kernel.

We can incorporate the indicator functions of the constraint in the regularizers, and define  $R = r + I_{\mathcal{X}}$ ,  $S = s + I_{\mathcal{W}}$ . Then the problem becomes

$$\text{minimize } l(x - y * w) + R(x) + S(w) \quad (2)$$

This is a discriminative model approach of blind deconvolution. The advantage of this problem is that it is a convex problem if  $l$ ,  $R$  and  $S$  are convex functions.

Even when one of them is not convex, as long as the proximal operator is computable, we can still use ADMM based method to approximately solve the problem.

## 2.2 Examples

Now we enumerate a few different choices for  $l, r, s, \mathcal{X}, \mathcal{W}$ .

**Choice of  $l$ .** For  $l$ , the most common choice is  $l(u) = \|u\|_2^2$ . Other choices are  $l(u) = \|u\|_1$  or Huber function. In general,  $l$  is the negative log-likelihood of the noise distribution.

**Choice of  $r$  and  $s$ .**  $r$  is the simplicity measure of the signal. One family of common examples are  $r(x) = \|x\|_1, \|Dx\|_1$  and  $L_1$  norm for higher order derivative. We can also replace  $L_1$  norm with Huber function and  $\|x\|_2^2$ . Another important family of choices are the higher order cumulants, which is to encourage the non-Gaussian property of the signal if we treat it as a sample of stochastic process.

For  $s$ , we can still choose the simplicity measure as discussed in  $r$ . We can also choose  $s$  to be simplicity measure on Fourier transform of the kernel. If we use Bayesian method and have a prior on the kernel, we can choose  $s$  as the logarithm of the prior distribution.

**Choice of  $\mathcal{X}$  and  $\mathcal{W}$ .** For  $\mathcal{X}$ , one natural example is that the sparsity level of  $x$  is below threshold  $k$ . It is a non-convex constraint but we can compute the projection on this constraint set by hard-thresholding.

For  $\mathcal{W}$ , we must fix the scale of  $w$ , or at least low-bound  $w$  away from *zero* to get a non-trivial solution. As discussed before, a subset  $\mathcal{W}$  fixes scale if it does not contain zero and if  $w \in \mathcal{W}$ , then  $cw \in \mathcal{W}$  imply  $c = \pm 1$ . One natural choice of  $\mathcal{W}$  is to fix some coordinate of  $w$ , for example,  $w_1 = 1$ . In general, we can fix a nonzero vector  $e$ , and fix  $e^T w = 1$ . Another choice is  $\|w\|_2 = 1$ , which is non-convex, but the projection is easy to compute. Besides the equality constraint that fixes scale, we can also constraint the length of  $w$ , for example, we can impose that only the first  $k$  coordinates of  $w$  can be non-zero.

## 2.3 Optimization method

To solve this problem, we can rewrite the problem as

$$\begin{aligned} & \text{minimize} && l(x - z) + R(\tilde{x}) + S(\tilde{w}) \\ & \text{subject to} && z = y * w, \\ & && w = \tilde{w}, \\ & && x = \tilde{x}, \end{aligned} \tag{3}$$

with variables  $w, \tilde{w}, x, \tilde{x}, z \in \mathbf{R}^n$ .

Then we can use alternating direction method of multiplier to solve this problem when the problem is convex. The augmented Lagrangian for this problem is

$$\begin{aligned} \mathcal{L}_{s_1, s_2, s_3}(w, \tilde{w}, x, \tilde{x}, z, u_1, u_2, u_3) &= l(x - z) + R(\tilde{x}) + S(\tilde{w}) + \frac{s_1}{2} \|z - y * w - u_1\|^2 \\ &\quad + \frac{s_2}{2} \|x - \tilde{x} - u_2\|^2 + \frac{s_3}{2} \|w - \tilde{w} - u_3\|^2, \end{aligned}$$

where  $u_1, u_2, u_3$  are the scaled dual variables,  $s_1, s_2, s_3$  are positive.

If we assume  $l, R, S$  are all convex functions, we can define a new variable  $v := (w, \tilde{w}, x, \tilde{x}, z)$ , the objective function  $f(v) = f(w, \tilde{w}, x, \tilde{x}, z) := l(x - z) + R(\tilde{x}) + S(\tilde{w})$ , and write the linear constraints as  $v \in C$ . Then the problem is

$$\begin{aligned} & \text{minimize} && f(v) \\ & \text{subject to} && v \in C, \end{aligned} \tag{4}$$

with variable  $v$ . Then it is a convex problem with linear constraint, we can apply the standard ADMM algorithm from chapter 5 of [?].

$$\begin{aligned} v^{k+1/2} &:= \text{prox}_f(v^k - \tilde{v}^k) \\ v^{k+1} &:= \Pi(v^{k+1/2} + \tilde{v}^k) \\ \tilde{v}^{k+1} &:= \tilde{v}^k + v^{k+1/2} - v^k. \end{aligned} \tag{5}$$

The variable  $k$  is the iteration counter,  $v^{k+1/2}$  and  $v^k$  are primal variables,  $\tilde{v}^k$  are scaled dual variables.

$$\text{prox}_f(v) = \underset{v'}{\operatorname{argmin}} \left( f(v') + (\rho/2) \|v' - v\|_2^2 \right)$$

is the proximal operator of  $f$ , and  $\rho > 0$  is the proximal parameter. To calculate  $\text{prox}_f$ , we only need to calculate the proximal operators  $\text{prox}_l$ ,  $\text{prox}_R$ ,  $\text{prox}_S$ .  $\Pi$  denotes the (Euclidean) projection onto the graph define by the linear constraint  $C$ , it can be explicitly expressed as the projections on the three equations  $z = y * w$ ,  $w = \tilde{w}$ ,  $x = \tilde{x}$ . The last two projections are averaging of  $w$ ,  $\tilde{w}$  and  $x$ ,  $\tilde{x}$ . The first projection is a projection on the graph form  $z = y * w$ , we can apply the graph projection splitting algorithm [?] to this problem. One advantage of convolution is that we can leverage the convolution theorem, and compute the projection of  $(z, w)$  onto  $z = y * w$  using Fourier transform,  $\hat{z} = \hat{y} \circ \hat{w}$ , where  $\circ$  is the Hadamard product. This update can be computed on each coordinate

$$\begin{aligned} \hat{z}^t &= \mathcal{F}z^t, \\ \hat{w}^t &= \mathcal{F}w^t, \\ (\hat{z}_i^{t+1}, \hat{w}_i^{t+1}) &= \Pi_{\hat{y}_i^{t+1}}(\hat{z}_i^{t+1}, \hat{w}_i^{t+1}). \end{aligned}$$

Here the projection is given by

$$\Pi_{\hat{y}_i}(\hat{z}_i, \hat{w}_i) = ((\hat{z}_i + \hat{y}_i \hat{w}_i)/(1 + \hat{y}_i^2), \hat{y}_i(\hat{z}_i + \hat{y}_i \hat{w}_i)/(1 + \hat{y}_i^2)).$$

Using Fast Fourier transform, we can compute this update in  $O(n \log n)$ , therefore, the method is suited for large scale problem. After the coordinate-wise projection, we apply inverse Fourier transform to get  $z^{t+1}, w^{t+1}$ .

## 2.4 Numerical example: Sparse blind deconvolution

As a primary example of the general framework, we study sparse blind deconvolution problem. Assume that we observe  $y \in \mathbf{R}^n$  from the circular convolution of  $w$  and  $x$ , namely,  $y = k * x$ , where the signal  $x \in \mathbf{R}^n$  is I.I.D. series sampled from a random variable  $X$  with probability  $1 - p$  to be zero, and probability  $p$  to be sampled from a standard Gaussian random variable, and the kernel  $k \in \mathbf{R}^n$  has an inverse  $w \in \mathcal{W}_m \subset \mathbf{R}^n$  so that  $w * k = e_1$ , where  $e_1 = (1, 0, \dots, 0) \in \mathbf{R}^n$ .  $\mathcal{W}_m$  is the embedding of  $\mathbf{R}^m$  to the first  $m$

coordinates of  $\mathbf{R}^n$ , it contains all the kernels zero-padded from length  $m$ , and  $\mathcal{X} = \{x \in \mathbf{R}^n | x_1 = 1\}$ , namely the first coordinate of  $x$  is fixed as 1. We choose  $r(x) = \|x\|_1$  as a convex relaxation of the sparsity measure.  $l$  is the indicator function  $I_{x=y*w}$  since the observation is noiseless in this synthetic model. We choose  $s = 0$  since there is not additional information on the kernel. Therefore, we have the following convex optimization problem:

$$\begin{aligned} & \text{minimize} && \|x\|_1 \\ & \text{subject to} && x = y * w, \quad x_1 = 1, \quad w \in \mathcal{W}_m, \end{aligned} \tag{6}$$

with variable  $w \in \mathcal{W}_m, x \in \mathbf{R}^n$ .

In the following numerical experiments, we choose the signal length  $n = 500$ , the kernel length  $m = 80$ ,  $x$  I.I.D. sampled from product of Bernoulli and normal with sparsity level 0.2,  $w$  I.I.D. sampled from standard normal. We run the above algorithm and We can see that the algorithm can perfectly recover the ground truth signal and kernel.

## 3 Extensions and connections

### 3.1 Generative blind deconvolution

Consider that we have an observation vector  $y \in \mathbf{R}^n$ , and we assume that  $y$  is generated by the following model,

$$y = x * k + e,$$

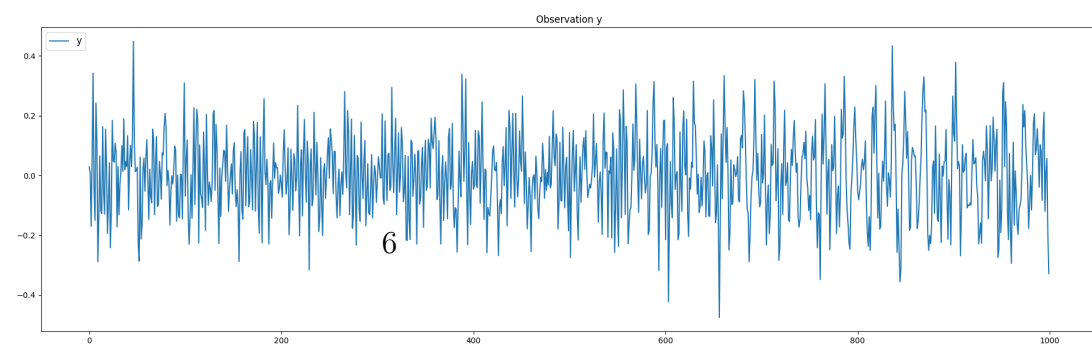
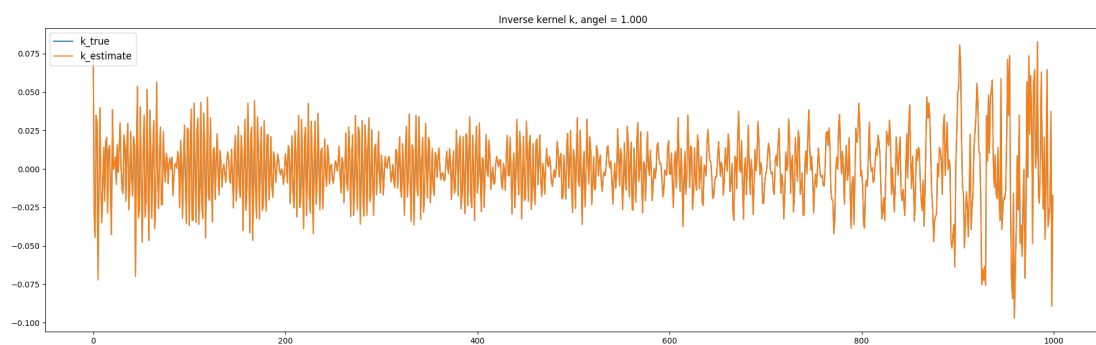
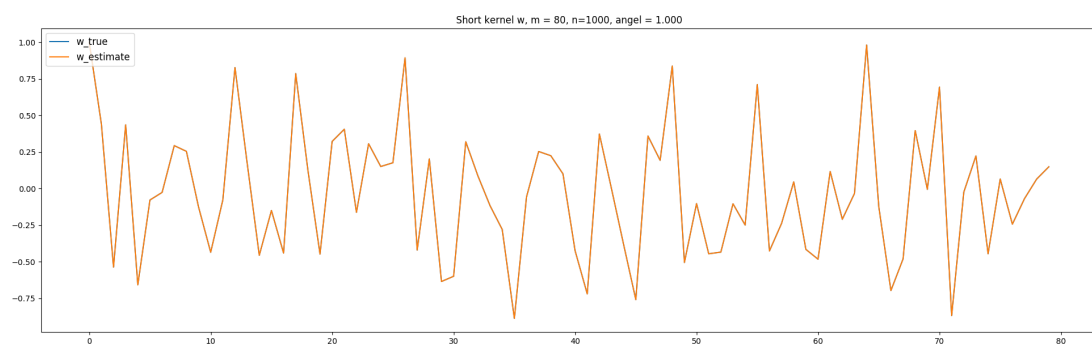
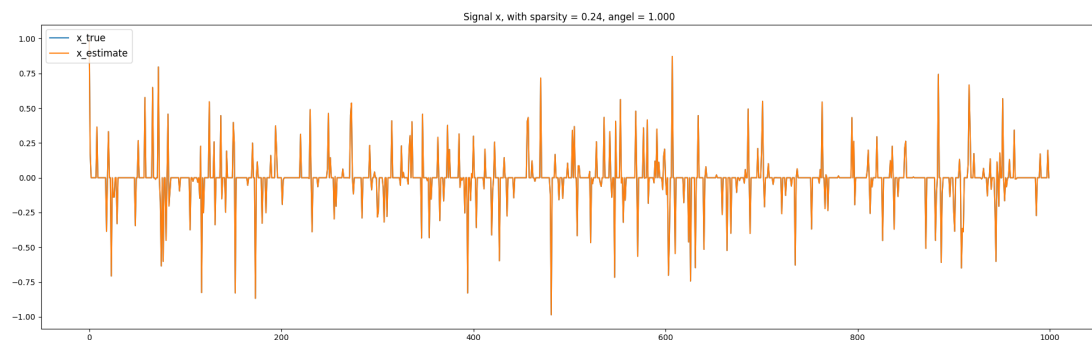
where  $x \in \mathcal{X}$  is a signal and  $k \in \mathcal{K}$  is a filter,  $e$  is the error of the convolutional approximation. As before, we need to normalize  $k$  in the definition of  $\mathcal{K}$  to fix the scale.

Then our optimization problem is

$$\text{minimize} \quad l(y - k * x) + R(x) + S(k) \tag{7}$$

with variable with variables  $x, k \in \mathbf{R}^n$ . As before,  $l$  is the loss function,  $R$  and  $S$  are regularizers that includes the indicator function for the constraints.

This can be viewed as synthesis approach, or generative model approach of blind deconvolution. The problem is non-convex, and the fundamental difficulty is that  $k$  and  $x$  are both unknown in  $k * x$ .



We can rewrite the problem as

$$\begin{aligned}
& \text{minimize} && l(y - z) + R(\tilde{x}) + S(\tilde{k}) \\
& \text{subject to} && z = x * k, \\
& && k = \tilde{k}, \\
& && x = \tilde{x},
\end{aligned} \tag{8}$$

with variables  $x, \tilde{x}, k, \tilde{k}, z$ .

Then we can use alternating direction method of multiplier (ADMM) to solve this problem. The augmented Lagrangian for this problem is

$$\begin{aligned}
\mathcal{L}_{s_1, s_2, s_3}(k, x, \tilde{k}, \tilde{x}, z, u_1, u_2, u_3) = & L(y, z) + R_1(\tilde{x}) + R_2(\tilde{k}) + I_{e^T k - 1 = 0} \\
& + s_1/2 \|z - k * x - u_1\|^2 + s_2/2 \|x - \tilde{x} - u_2\|^2 \\
& + s_3/2 \|k - \tilde{k} - u_3\|^2.
\end{aligned}$$

Here  $u_1, u_2, u_3$  are the scaled dual variables,  $s_1, s_2, s_3$  are positive.

Solving this problem is harder because  $k$  and  $x$  are both variable in  $k * x$ . Therefore, this problem is always non-convex. In practice, we will alternate between the updates on  $k$  and  $x$ .

### 3.2 Connection to independent component analysis

The generalized blind deconvolution model is closely related to independent component analysis (ICA). In ICA, we are given an observation vector  $y \in \mathbf{R}^n$  and trying to solve the problem

$$\begin{aligned}
& \text{minimize} && \|Wy\|_1 \\
& \text{subject to} && WW^T = I
\end{aligned} \tag{9}$$

with variables  $W \in \mathbf{R}^{n \times n}$ .

In the simplest form of sparse blind deconvolution, we are solving the problem

$$\begin{aligned}
& \text{minimize} && \|w * y\|_1 \\
& \text{subject to} && w_1 = 1,
\end{aligned} \tag{10}$$

with variables  $w \in \mathcal{W} \subset \mathbf{R}^n$ . If we use the circular matrix representation of convolution, we can rewrite the problem as

$$\begin{aligned}
& \text{minimize} && \|Wy\|_1 \\
& \text{subject to} && \frac{1}{n} \text{tr}(W) = 1, \\
& && W \text{ is a circulant matrix,}
\end{aligned} \tag{11}$$

with variables  $W \in \mathbf{R}^{n \times n}$ .

We can see that blind deconvolution is replacing the constraint that  $W$  is an orthogonal matrix to that  $W$  is a circular matrix with fixed trace.

### 3.3 Extension to deep deconvolution neural networks

One natural extension of the discriminative blind deconvolution problem is to replace the linear convolution  $w * y$  by a  $m$ -layer convolutional neural network. If we use the Relu function  $\sigma(x) = \max(x, 0)$ , then we can define  $h_0 = y$ , and  $h_t = \sigma(w_t * h_{t-1})$  for  $t = 1, \dots, m$ .

Then the optimization problem is

$$\begin{aligned} & \text{minimize} && l(x - h_m) + R(x) + S(w_1, w_2, \dots, w_m), \\ & \text{subject to} && h_0 = y, \quad h_t = \sigma(w_t * h_{t-1}), \quad t = 1, \dots, m, \end{aligned} \tag{12}$$

with the variables are  $w_1, \dots, w_m, h_1, \dots, h_m, x$ .

From machine learning perspective, blind deconvolution problem is an unsupervised learning problem. Our input is  $y$  and output is  $x$ , since we don't have direct supervision on  $x$ , we only have a regularizer on  $x$  to impose simplicity. This is a really hard problem, and to get a useful result from this problem, we need that the dimension of  $y$  significantly longer than the size of the kernels.

The generalized blind deconvolution model can be viewed an autoencoder using convolutional neural network with one hidden layer. Input of the neural network is  $y$ , and  $k$  is the convolutional kernel to learn,  $x$  is the hidden state,  $z = k * x$  is the output of autoencoder, and  $L(y, z)$  is the reconstruction error in this autoencoder. We can extend the model in similar fashion, then we will get a multi-layer convolutional autoencoder.