Generalized Blind Deconvolution

Samsung Global Research Outreach Program 2017
Progress Report

Boyd Group, Stanford University

1 Proposal Identification

Title of Proposal
Generalized Blind Deconvolution

Subject Theme Machine Learning and Recognition

Subject Title Other

Principal Investigator
Stephen P. Boyd
Samsung Professor in the School of Engineering
Professor, Information Systems Laboratory, Department of Electrical Engineering
Professor (by courtesy), Department of Management Science and Engineering
Professor (by courtesy), Department of Computer Science
Institute for Computational and Mathematical Engineering
Stanford University
Packard 254, Stanford, CA 94305
Tel: (650) 723-0002
Fax: (650) 723-8473
`boyd@stanford.edu`
`http://www.stanford.edu/~boyd/`

Co-PI Information None.

Joint Proposal No.

2 Generalized blind deconvolution

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

We have an observation vector $y \in \mathbf{R}^n$, and we seek a filter $w \in \mathbf{R}^n$ such that the (circular) convolution $w * y \in \mathbf{R}^n$ is close to a signal $x \in \mathbf{R}^n$ that is simple under a given measure. We formulate this as the problem

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

with variable w and x. Here $l : \mathbf{R}^n \to \mathbf{R}$ is a loss function that measures the approximation error, $r : \mathbf{R}^n \to \mathbf{R}$ is a regularizer on the signal x, $s : \mathbf{R}^n \to \mathbf{R}$ is a regularizer on the kernel, and $\mathcal{W}$ and $\mathcal{X}$ are constraint sets. The constraint sets $\mathcal{W}$ and $\mathcal{X}$ always contain a *normalization constraint* on w, to preclude the trivial solution $w = x = 0$.

We can incorporate the indicator functions of the constraints in the regularizers, by defining $R = r + I_{\mathcal{X}}$, $S = s + I_{\mathcal{W}}$. Then the problem can be expressed as

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

with variables x and w. The problem is convex if l, r, s, $\mathcal{W}$, and $\mathcal{X}$ are convex.

Loss. Common choices for the loss function are $l(u) = \|u\|_2^2$, $l(u) = \|u\|_1$. For the exact case, $l = I_0$, we have a equality constraint $x = y * w$. If we have a model of the form $y = w * x + e$, where e is a noise with density p, we can take $l(u) = -\log p(u)$, the negative log-likelihood.

Signal regularizer. Common convex choices include $r(x) = \|x\|_1$, total variation $\|Dx\|_1$, or Laplacian $\|Dx\|_2^2$, where D is the circular difference matrix. Common nonconvex choices include $\|x\|_0$, the cardinality of the support of x, or $\|x\|_p^p$ with $0 < p < 1$.

Kernel regularizer. Common choices include the same choices as signal regularizer, such as $s(w) = \|w\|_1$, total variation $\|Dw\|_1$.

Signal and kernel set. The signal set $\mathcal{X}$ can specify simplicity, for example by limiting the cardinality of the support of x. It must also include a normalization such as $\|x\| = 1$ for some norm $\|\cdot\|$ or $c^T x = 1$ for some nonzero $c \in \mathbf{R}^n$. The same choices can be made for $\mathcal{W}$.

3 Algorithm

We first rewrite the problem as

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

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

Then we can use the alternating direction method of multipliers (ADMM) to solve this problem, when it is convex, or approximately solve the problem, otherwise. The augmented Lagrangian for this problem is

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

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

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

$$\begin{array}{ll} \text{minimize} & f(v) \\ \text{subject to} & v \in C, \end{array}$$

with variable v. Then it is a convex problem with linear constraint, we can apply the standard ADMM algorithm.

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

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

The proximal operator of f is

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

where $\rho > 0$ is the proximal parameter. Here prox_f equals the sum of the proximal operators $\mathrm{prox}_l, \mathrm{prox}_R, \mathrm{prox}_S$.

The projection Π projects the variables onto the graph of the linear constraint C define by the three equations $z = y * w, w = \tilde{w}, x = \tilde{x}$.

To compute the projection on the graph form $z = y * w$, we can leverage the convolution theorem, and compute the projection of (z, w) onto $z = y * w$ using its 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 $O(n \log n)$. After the coordinate-wise projection, we apply inverse Fourier transform to get z^{t+1}, w^{t+1}.

4 Numerical example

As an example, we study sparse blind deconvolution problem, where $r(x) = \|x\|_1$ as a convex relaxation of the sparsity measure, $\mathcal{X} = \{x \in \mathbf{R}^n | x_1 = 1\}$, $\mathcal{W}$ is the embedding of $\mathbf{R}^m$ to the first m coordinates of $\mathbf{R}^n$, $s = 0$, $l = I_0$ is the indicator function to impose equality.

We generate the observation $y \in \mathbf{R}^n$ from the circular convolution of w and x, namely, $y = k * x$, where the signal $x \in \mathbf{R}^n$ is I.I.D. sampled from the product of a Bernoulli random variable with probability p and 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 $e_1 = (1, 0, \ldots, 0) \in \mathbf{R}^n$. In the following numerical experiments, we choose the signal length $n = 500$, the kernel length $m = 80$, x has expected sparsity level $p = 0.2$. We run the above algorithm and our algorithm exactly recover the ground truth.

5 Extensions and connections

5.1 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,

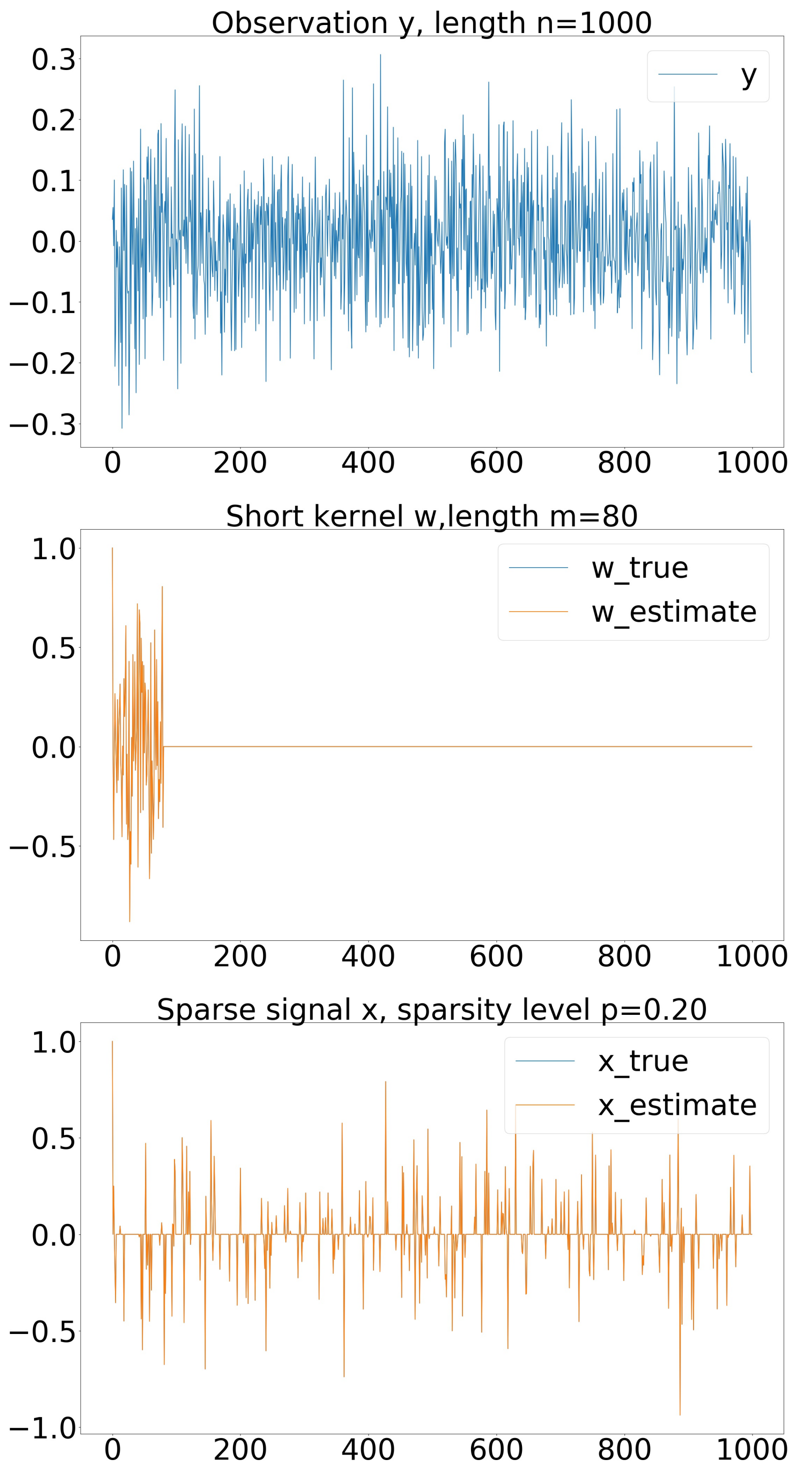
$$y = x * k + e,$$

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, the constraint sets $\mathcal{X}, \mathcal{K}$ always contain a *normalization constraint* on k because of the symmetry in $k * x$.

Our optimization problem is

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

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.


Observation y, length n=1000
y
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0
200
400
600
800
1000
Short kernel w,length m=80
w_true
w_estimate
1.0
0.5
0.0
−0.5
0
200
400
600
800
1000
Sparse signal x, sparsity level p=0.20
x_true
x_estimate
1.0
0.5
0.0
−0.5
−1.0
0
200
400
600
800
1000

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

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

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

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

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.

5.2 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

$$\begin{array}{ll} \text{minimize} & \|Wy\|_1 \\ \text{subject to} & WW^T = I \end{array} \tag{2}$$

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

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

$$\begin{array}{ll} \text{minimize} & \|w * y\|_1 \\ \text{subject to} & w_1 = 1, \end{array} \tag{3}$$

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

$$\begin{array}{ll} \text{minimize} & \|Wy\|_1 \\ \text{subject to} & \frac{1}{n}\mathrm{tr}(W) = 1, \\ & W \text{ is a circulant matrix,} \end{array} \tag{4}$$

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.

5.3 Extension to deep deconvolution

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 $t = 1, \ldots, m$.

Then the optimization problem is

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

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

We call this problem deep deconvolution. 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.

6 Samsung contacts