Deconvolution ISTA: A solver for multidimensional convolution problems with low computational complexity

Masanori Gocho1a)
1 National Institute of Information and Communications Technology
4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795 Japan
a) gocho@nict.go.jp

Abstract: In this report, we employ the iterative shrinkage-thresholding algorithm (ISTA), which is one of the sparse reconstruction methods, to solve multidimensional circular convolution problems. The novelties of this work are as follows: the derivation of the subgradient and the Lipschitz constant of a multidimensional deconvolution problem; the construction of a sparse reconstruction algorithm to solve the problem; the evaluation of the qualitative ability of the algorithm, especially computational complexities. The proposed method can not only solve the convolution problems but also achieve low computational complexity.

Keywords: ISTA, Convolution problem, Sparse reconstruction, FFT, Computational complexity

Classification: Sensing

References

1 Introduction

Convolution often appears in the formulation of various problems [1].

• In sensing using an active sensor, the convolution of the transmitted signal with spatial information gives the observation signal.

• In linear system theory, the convolution of the input of a system with the impulse response gives the output of the system.

• In image processing, filtering is a two-dimensional convolution of an image and a filter.

The inverse problem of the convolution is called deconvolution, the purpose of which is to estimate the signal before convolution by using the known observation signal and the convolved signal.

In this report, we propose a deconvolution solver that uses sparse reconstruction. Specifically, we optimize the iterative shrinkage-thresholding algorithm (ISTA) [2, 3] to deconvolution problems [4]; it is the most common sparse reconstruction method and can solve linear problems. Sparse reconstruction is used to reconstruct signals under the assumption that the reconstructed signal is sparse, where sparse means that the signal has many zero components. The prior information that the estimated signal is sparse allows us to achieve higher performance in terms of resolution and noise immunity than by conventional methods such as matched filtering and Wiener filtering [5].

The novelties of this work are as follows:

• the derivation of the subgradient and the Lipschitz constant of a multidimensional deconvolution problem;

• the construction of a sparse reconstruction algorithm to solve the multidimensional deconvolution problem;

© IEICE 2022
• the evaluation of the qualitative ability of the algorithm, especially computational complexities.

Another novelty is that the algorithm is derived without any conversion such as vectorization, which has been used in some related studies [6, 7, 8]. To simplify the deconvolution problem, we often use vectorization for the multidimensional array. However, the vectorization for the convolution problem gives unnecessary degrees of freedom, causing high computational complexity and mathematical ambiguity.

The rest of the paper is organized as follows. In sec. 2, we define and describe multidimensional convolution problems. In sec. 3, we propose a sparse solver for the multidimensional convolution problem. In sec. 4, we show the performances of the proposed algorithm. Finally in sec. 5, we conclude this paper.

2 Convolution/Deconvolution Problems

In a linear system, the observed signal of the system holds the convolutional relationship

\[ S = A \ast \cdots \ast X + C, \]  

where \( S \in \mathbb{C}^{N_1 \times \cdots \times N_D} \) is the signal array of the system; \( A \in \mathbb{C}^{N_1 \times \cdots \times N_D} \) is the impulse response of the system; \( X \in \mathbb{C}^{N_1 \times \cdots \times N_D} \) is an unknown signal array to be estimated; \( C \in \mathbb{C}^{N_1 \times \cdots \times N_D} \) is an additive noise uncorrelated with the signal; \( D \in \mathbb{N}_+ \) is the number of dimensions of the signal; and \( N_d \in \mathbb{N}_+, d = 1, \ldots, D \) is the size of the \( d \)-th dimension of the signal. The binary operator \( \ast \cdots \ast \) denotes the \( D \)-dimensional circular convolution with period \( N_d, d = 1, \ldots, D \). In the case of \( H = F \ast \cdots \ast G \), it can be defined as

\[ H[n] = \sum_m F[m] G[n - m \mod N], \]  

where \( F, G, \) and \( H \) are same-size arrays; vectors \( m = [m_1, \ldots, m_D], (m_d = 0, 1, \ldots, N_d - 1) \) and \( n = [n_1, \ldots, n_D], (n_d = 0, 1, \ldots, N_d - 1) \) are indices of these arrays; \( F[n], G[n], \) and \( H[n] \) are the \( n \)-th element of the arrays \( F, G, \) and \( H \), respectively; and the binary operator \( n \mod N \) is the (elementwise) modulo operation. Specifically, in the case of a two-dimensional problem, Eq. (2) can be written as

\[ H[n_1, n_2] = \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} F[m_1, m_2] G[n_1 - m_1 \mod N_1, n_2 - m_2 \mod N_2]. \]  

Note that Eq. (1) includes a one-dimensional convolution problem such as \( s = a \ast x + c \in \mathbb{C}^N \). The deconvolution problem corresponding to Eq. (1) is the inverse problem of estimating \( X \) using known \( A \) and \( S \). Figure 1 shows an example of the convolution/deconvolution problem.

One solution of Eq. (1) is expressed as

\[ X_{LS} = \arg \min_Z \frac{1}{2} \| S - A \ast \cdots \ast Z \|_2^2, \]  

© IEICE 2022
Fig. 1. System model of a convolution/deconvolution problem.

where $\|\cdot\|_p$ denotes the $p$ norm, which is defined as

$$
\|V\|_p = \left( \sum_{v \in V} |v|^p \right)^{\frac{1}{p}}
$$

(5)

for any array $V$. The solution of Eq. (4), which is well known as the least squares (LS) fitting, has a limit in the resolution because the bandwidth corresponds to $A$.

3 Sparse Deconvolution using ISTA

One solution of Eq. (1) based on the least absolute shrinkage and selection operator (LASSO) can be expressed as

$$
X_{\text{LASSO}} = \arg \min_Z \left( \frac{1}{2} \| S - A * \cdots * Z \|_2^2 + \lambda \| Z \|_1 \right),
$$

(6)

where $\lambda$ is the regularization parameter of the LASSO and controls the sparsity of the solution.

The ISTA [2] is a solver for Eq. (6) and can iteratively search for the optimized value $X_{\text{ISTA}}$ as follows:

$$
X_{\text{ISTA}}^{(\text{new})} = S_{\frac{1}{L}} \left( X_{\text{ISTA}} - \frac{1}{L} \nabla f (X_{\text{ISTA}}) \right),
$$

(7)

$$
f (Z) = \frac{1}{2} \| S - A * \cdots * Z \|_2^2,
$$

(8)

$$
S_\alpha (v) = \max \left( |v| - \alpha, 0 \right) e^{j \arg v},
$$

(9)

where $\nabla$ is the gradient operator; $L$ is the Lipschitz constant of the function $\nabla f$; and $S_\alpha$ is the soft thresholding operator with the amplitude parameter $\alpha$. Note that Eq. (8) is not holomorphic, so $\nabla f$ used in Eq. (7) is actually a subgradient at $X_{\text{ISTA}}$. © IEICE 2022
Since
\[ \nabla f (Z) = -A_{\text{reverse}} \ast \cdots \ast (S - A \ast \cdots \ast Z), \] (10)
we can obtain
\[ X_{\text{ISTA}} - \frac{1}{L} \nabla f (X_{\text{ISTA}}) = Y_{\text{ISTA}} + X_0, \] (11)
\[ Y_{\text{ISTA}} = X_{\text{ISTA}} - \frac{1}{L} A_{\text{reverse}} \ast \cdots \ast A \ast \cdots \ast X_{\text{ISTA}}, \] (12)
\[ X_0 = \frac{1}{L} A_{\text{reverse}} \ast \cdots \ast S, \] (13)
where \( V \) (over-line) and \( V_{\text{reverse}} \) (subscript “reverse”) are the complex conjugate and the array whose element order is reversed. We can simplify Eqs. (12) and (13) to
\[ Y_{\text{ISTA}} = X_{\text{ISTA}} - \frac{1}{L} F^{-1} \left( F(A) \circ F(X_{\text{ISTA}}) \right), \] (14)
\[ X_0 = \frac{1}{L} F^{-1} \left( F(A) \circ F(S) \right), \] (15)
\[ W = 1_{N_1 \times \cdots \times N_D} - \frac{P(A)}{L}, \] (16)
using the convolution theorem
\[ F(F \ast \cdots \ast G) = F(F) \circ F(G), \] (17)
where \( F \) is the discrete Fourier transform (DFT), which is implemented by the fast Fourier transform (FFT); \( P \) is the power spectrum function, which is defined as \( P(V) = F(V) \circ F(V) \); \( 1_{\text{size}} \in \mathbb{N}^{\text{size}} \) is an array in which all elements are one; and the binary operator \( \circ \) is the Hadamard product. We also used the following property of the Fourier transform:
\[ F(V_{\text{reverse}}) = \overline{F(V)}. \] (18)

\( L \) is defined as
\[ L = \sup_{\Delta X \neq 0} \frac{\| \nabla f (X + \Delta X) - \nabla f (X) \|_2}{\| \Delta X \|_2}, \] (19)
where \( \Delta X \) is any array with the same size as \( X \). From Eq. (10), we can obtain
\[ \| \nabla f (X + \Delta X) - \nabla f (X) \|_2 = \left\| F^{-1} (P(A) \circ F(\Delta X)) \right\|_2, \] (20)
\[ \| \Delta X \|_2 = \left\| F(\Delta X) \right\|_2, \] (21)
where we used the linearity and isometry of the DFT Eq. (21). Thus,
\[ L = \sup_{\Delta X \neq 0} \frac{\| P(A) \circ F(\Delta X) \|_2}{\| F(\Delta X) \|_2}. \]
Algorithm 1 Deconvolution ISTA

Input: \( S \in \mathbb{C}^{N_1 \times \cdots \times N_D}, A \in \mathbb{C}^{N_1 \times \cdots \times N_D}, \lambda \in \mathbb{R}_+ \)

Output: \( X \in \mathbb{C}^{N_1 \times \cdots \times N_D} \)

1: \( X \leftarrow 0_{N_1 \times \cdots \times N_D} \)

2: \( L \leftarrow \max \mathcal{P}(A) \)

3: \( W \leftarrow 1_{N_1 \times \cdots \times N_D} - \frac{\mathcal{P}(A)}{L} \)

4: \( X_0 \leftarrow \frac{1}{L} \mathcal{F}^{-1} \left( \mathcal{F}(A) \circ \mathcal{F}(S) \right) \)

5: \( \text{repeat} \)

6: \( X \leftarrow S_{\frac{\lambda}{L}} \left( \mathcal{F}^{-1} (W \circ \mathcal{F}(X)) + X_0 \right) \)

7: \( \text{until } X \text{ is converged} \)

\[ = \max \mathcal{P}(A), \quad (22) \]

because Eq. (22) is the spectral norm of \( \mathcal{F}(\Delta X) \).

By summarizing Eqs. (7), (11), (14), (15), (16), and (22), we can obtain algorithm 1. The algorithm is constructed of two blocks: the initialization block 1–4 and the iterative updating block 5–7. In the updating block, the desired array \( X \) is updated until it is sufficiently converged. We should use the exact convergence condition as the minimization function Eq. (6), but actually, it is sufficient to monitor the relative change of \( X \).

4 Qualitative Performance

4.1 Estimation performance

There are many indicators of the algorithm performance, but the estimation performance of the proposed method is the same as that of the normal ISTA except for the computational complexity. Here, the normal ISTA refers to a sparse solver of the matrix equation \( s = Ax + c \) for \( x \) [2]. Specifically, the proposed method inherits the following features from the normal ISTA: resolution, robustness to noise, and characteristics of convergence. For the performance of the ISTA, refer to reports such as ref. [3].

We can easily confirm the above similarity by converting Eq. (1) to a matrix equation,

\[ s = Bx + c, \quad (23) \]

where

\[ s = \text{vec}(S) \in \mathbb{C}^N, \quad (24) \]

\[ B = [a_1, a_2, \ldots, a_N] \in \mathbb{C}^{N \times N}, \quad (25) \]

\[ x = \text{vec}(X) \in \mathbb{C}^N, \quad (26) \]

\[ c = \text{vec}(C) \in \mathbb{C}^N, \quad (27) \]

\[ N = \prod_{d=1}^{D} N_d, \quad (28) \]
Table I. Comparison of computational complexities.

<table>
<thead>
<tr>
<th></th>
<th>Normal ISTA</th>
<th>Deconvolution ISTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{T}(N)$</td>
<td>$O(T \cdot N^2)$</td>
<td>$O(T \cdot N \log N)$</td>
</tr>
<tr>
<td>$\mathcal{M}(N)$</td>
<td>$O(N^2)$</td>
<td>$O(N)$</td>
</tr>
</tbody>
</table>

and $a_n \in \mathbb{C}^N, n = 1, \ldots, N$ is a vector calculated using $A$ corresponding to the $n$-th estimation point. The vectorization conversion shown above has been used in refs [6, 7, 8]. In the case of one-dimensional problems, $s = a^* x + c \leftrightarrow s = Bx + c$, where $B$ is the circulant matrix of $a$. The equivalence of this conversion shows that the proposed method and the normal ISTA have the same estimation performance.

4.2 Computational complexity

There are two indicators of complexity: time complexity $\mathcal{T}(N)$ and space complexity $\mathcal{M}(N)$, where $N$ is the input length of the algorithm. In the case of algorithms using multi-dimensional arrays, the total number of elements, $N = \prod_{d=1}^{D} N_d$, is usually used.

The proposed method consists of elementwise operators and the FFT. The time complexity of the proposed method $\mathcal{T}(N)$ depends on the number of iterations, $T$, and the time complexity of the FFT. Thus, the time complexity $\mathcal{T}(N)$ is

$$\mathcal{T}(N) = O(T \cdot \mathcal{T}_{FFT}(N)) = O(T \cdot N \log N), \quad (29)$$

where $\mathcal{T}_{FFT}(N)$ is the time complexity of the FFT, which is the well-known equation $\mathcal{T}_{FFT}(N) = O(N \log N)$ [9]. On the other hand, the proposed method uses some scalar numbers and the same-size arrays as $N_1 \times N_2 \times \cdots \times N_D$. Thus, the space complexity of the proposed method is

$$\mathcal{M}(N) = O(N) + \mathcal{M}_{FFT}(N) = O(N), \quad (30)$$

where $\mathcal{M}_{FFT}(N) = O(1)$ is the space complexity of the FFT [9]. Therefore, the proposed method depends on the log-linear time and linear memory space. Table I shows the differences between the normal ISTA and the proposed method in terms of the computational complexities. We can confirm that both computational complexities are reduced thanks to the FFT. In multi-dimensional problems, the proposed method is useful because $N$ increases exponentially.

Note that $T$ is not negligibly small. However, we can easily reduce $T$ to employ the Nesterov gradient accelerator; the ISTA employing it is called the Fast-ISTA (FISTA) [10]. The proposed method has the same extensibility as the conventional ISTA.
5 Conclusion

In this paper, we proposed a sparse solver using the ISTA for the deconvolution problem Eq. (1). The proposed algorithm 1 can not only solve the convolution problem with sparsity but also run in quasi-linear time and linear memory consumption.