

## ADMM アルゴリズムを用いた優決定ブラインド音源分離\*

☆ 渡會博子, 山田宏樹, 矢田部浩平 (農工大)

## 1 まえがき

優決定ブラインド音源分離の手法として, 主-双対近接分離法 (PDS) を用いてコスト関数の最小化問題を解き, 分離フィルタを最適化するフレームワークが提案されている [1]. しかし, PDS は計量を歪めるため [2], 収束に多くの反復を要する場合がある.

経験的に PDS よりも収束が速いアルゴリズムとして交互方向乗数法 (ADMM) が知られている. 本稿では, ADMM を用い, PDS よりも少ない反復回数で収束するアルゴリズムを提案する.

## 2 PDS を用いたブラインド音源分離

優決定ブラインド音源分離は, 分離フィルタ  $\mathbf{W}[f] \in \mathbb{C}^{N \times M}$  の推定問題として定式化される. 適切な分離行列が推定できれば, 時間周波数領域における  $M$  チャンネルの観測信号  $\mathbf{x}[t, f] \in \mathbb{C}^M$  から,  $N$  音源の分離音  $\mathbf{y}[t, f] \in \mathbb{C}^N$  を線形な演算

$$\mathbf{y}[t, f] = \mathbf{W}[f] \mathbf{x}[t, f] \quad (1)$$

で得られる. 特に, 音源数  $N$  とマイクロホン数  $M$  の間に  $N \leq M$  が成り立つ場合を優決定と呼ぶ.

従来法として, PDS アルゴリズムを用いたブラインド音源分離フレームワークが提案されている [1]. このフレームワークは従来の音源モデルを

$$\text{Minimize}_{\mathbf{W}[f]} \mathcal{P}(\mathbf{W}[f] \mathbf{x}[t, f]) - \sum_{f=1}^F \log |\det(\mathbf{W}[f])| \quad (2)$$

の形式で一般化して扱う. ただし,  $\mathcal{P}$  は音源モデルごとのペナルティ項である.  $\mathcal{P}$  が  $l_1$  ノルムや  $l_{2,1}$  混合ノルムである場合, それぞれ周波数領域独立成分分析 (FDICA) と独立ベクトル分析 (IVA) に対応する. 式 (2) は, PDS によって最適化可能な形式の問題として次のようにも定式化できる.

$$\text{Minimize}_{\mathbf{w}} \mathcal{I}(\mathbf{w}) + \mathcal{P}(\mathbf{X}\mathbf{w}) \quad (3)$$

ここで,  $\mathbf{w} \in \mathbb{C}^{NM}$  はすべての周波数について  $\mathbf{W}[f]$  をベクトル化し並べたもの,  $\mathbf{X} \in \mathbb{C}^{NTF \times NMF}$  は観測信号からなるブロック対角行列であり,  $\mathcal{I}(\mathbf{w})$  は  $\mathcal{I}(\mathbf{w}) = -\sum_{f=1}^F \sum_{n=1}^N \log \sigma_n(\text{mat}(\mathbf{w}[f]))$  である. ただし,  $\sigma_n$  は行列の  $n$  番目の特異値を得る関数であり,  $\text{mat}(\cdot)$  はベクトルを行列化する作用素である. PDS を用いれば, 行列  $\mathbf{X}$  およびその共役転置, そして後述する各項の近接作用素のみを用いて最適化できる [1].

## Algorithm 1 Proposed Method

Input:  $\mathbf{X}, \mathbf{w}^{[1]}, \mathbf{z}_1^{[1]}, \mathbf{z}_2^{[1]}, \mathbf{u}_1^{[1]}, \mathbf{u}_2^{[1]}, \rho, \alpha$ Output:  $\mathbf{w}^{[K+1]}$ 


---

```

1: for  $k = 1, \dots, K$  do
2:    $\mathbf{w}^{[k+1]} = (\mathbf{I} + \mathbf{X}^H \mathbf{X})^{-1} \left( (\mathbf{z}_1^{[k]} - \mathbf{u}_1^{[k]}) + \mathbf{X}^H (\mathbf{z}_2^{[k]} - \mathbf{u}_2^{[k]}) \right)$ 
3:    $\mathbf{y}_1 = \alpha \mathbf{w}^{[k+1]} + (1 - \alpha) \mathbf{z}_1^{[k]}$ 
4:    $\mathbf{y}_2 = \alpha \mathbf{X} \mathbf{w}^{[k+1]} + (1 - \alpha) \mathbf{z}_2^{[k]}$ 
5:    $\mathbf{z}_1^{[k+1]} = \text{prox}_{\frac{\rho}{2}} [\mathbf{y}_1 + \mathbf{u}_1^{[k]}]$ 
6:    $\mathbf{z}_2^{[k+1]} = \text{prox}_{\frac{\rho}{2}} [\mathbf{y}_2 + \mathbf{u}_2^{[k]}]$ 
7:    $\mathbf{u}_1^{[k+1]} = \mathbf{u}_1^{[k]} + \mathbf{y}_1 - \mathbf{z}_1^{[k+1]}$ 
8:    $\mathbf{u}_2^{[k+1]} = \mathbf{u}_2^{[k]} + \mathbf{y}_2 - \mathbf{z}_2^{[k+1]}$ 
9: end for

```

---

## 3 提案手法

PDS よりも経験的に収束が速いことが知られているアルゴリズムとして, ADMM がある. そこで本稿では, ADMM を用いた音源分離を提案する. 提案手法を Alg. 1 に示す. 以下にその導出を記す.

## 3.1 再定式化と ADMM アルゴリズム

式 (3) を ADMM によって最適化可能な形式に再定式化すると,

$$\begin{aligned} & \text{Minimize}_{\mathbf{w}} \quad \mathcal{I}(\mathbf{z}_1) + \mathcal{P}(\mathbf{z}_2) \\ & \text{subject to} \quad \mathbf{z}_1 = \mathbf{w}, \quad \mathbf{z}_2 = \mathbf{X}\mathbf{w} \end{aligned} \quad (4)$$

となる. ここで,  $\mathbf{z}_1, \mathbf{z}_2$  は補助変数である. 式 (4) は制約付き最小化問題であり,

$$\begin{cases} \mathbf{w}^{[k+1]} = \arg \min_{\mathbf{w}} \left( \left\| \begin{bmatrix} \mathbf{I} \\ \mathbf{X} \end{bmatrix} \mathbf{w}^{[k]} - \begin{bmatrix} \mathbf{z}_1^{[k]} \\ \mathbf{z}_2^{[k]} \end{bmatrix} + \begin{bmatrix} \mathbf{u}_1^{[k]} \\ \mathbf{u}_2^{[k]} \end{bmatrix} \right\|_2^2 \right) \\ \mathbf{y}_1^{[k+1]} = \alpha \mathbf{w}^{[k+1]} + (1 - \alpha) \mathbf{z}_1^{[k]} \\ \mathbf{y}_2^{[k+1]} = \alpha \mathbf{X} \mathbf{w}^{[k+1]} + (1 - \alpha) \mathbf{z}_2^{[k]} \\ \mathbf{z}_1^{[k+1]} = \arg \min_{\mathbf{z}_1} \left( \mathcal{I}(\mathbf{z}_1) + \frac{\rho}{2} \|\mathbf{y}_1^{[k+1]} - \mathbf{z}_1^{[k]} + \mathbf{u}_1^{[k]}\|_2^2 \right) \\ \mathbf{z}_2^{[k+1]} = \arg \min_{\mathbf{z}_2} \left( \mathcal{P}(\mathbf{z}_2) + \frac{\rho}{2} \|\mathbf{y}_2^{[k+1]} - \mathbf{z}_2^{[k]} + \mathbf{u}_2^{[k]}\|_2^2 \right) \\ \mathbf{u}_1^{[k+1]} = \mathbf{u}_1^{[k]} + \mathbf{y}_1^{[k+1]} - \mathbf{z}_1^{[k+1]} \\ \mathbf{u}_2^{[k+1]} = \mathbf{u}_2^{[k]} + \mathbf{y}_2^{[k+1]} - \mathbf{z}_2^{[k+1]} \end{cases} \quad (5)$$

を反復的に適用することで解くことができる. ただし,  $\mathbf{y}_1, \mathbf{y}_2$  の更新式はアルゴリズムの安定性を向上させるための緩和操作であり,  $\alpha \in (0, 2)$ ,  $\rho > 0$  である. また,  $\mathbf{u}_1, \mathbf{u}_2$  は双対変数である [3].

\* Determined blind source separation using ADMM algorithm. By Hiroko WATARAI, Koki YAMADA and Kohei YATABE (Tokyo University of Agriculture and Technology)

### 3.2 w の更新

式 (5) 中の  $\mathbf{w}$  の更新式は微分可能な関数の最小化問題である。その勾配を求めると、

$$\begin{aligned} \nabla_{\mathbf{w}} \left\| \begin{bmatrix} \mathbf{I} \\ \mathbf{X} \end{bmatrix} \mathbf{w} - \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \right\|_2^2 \\ = \begin{bmatrix} \mathbf{I} \\ \mathbf{X} \end{bmatrix}^H \begin{bmatrix} \mathbf{I} \\ \mathbf{X} \end{bmatrix} \mathbf{w} - \begin{bmatrix} \mathbf{I} \\ \mathbf{X} \end{bmatrix}^H \left( \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \right) \end{aligned} \quad (6)$$

となる。一次の最適性条件より勾配を  $\mathbf{0}$  と置き、これを解けば  $\mathbf{w}$  の更新式が以下のように得られる。

$$\begin{aligned} \mathbf{w} &= \left( \begin{bmatrix} \mathbf{I} \\ \mathbf{X} \end{bmatrix}^H \begin{bmatrix} \mathbf{I} \\ \mathbf{X} \end{bmatrix} \right)^{-1} \left( \begin{bmatrix} \mathbf{I} \\ \mathbf{X} \end{bmatrix}^H \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{I} \\ \mathbf{X} \end{bmatrix}^H \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \right) \\ &= (\mathbf{I} + \mathbf{X}^H \mathbf{X})^{-1} ((\mathbf{z}_1 - \mathbf{u}_1) + \mathbf{X}^H (\mathbf{z}_2 - \mathbf{u}_2)) \end{aligned} \quad (7)$$

### 3.3 z の更新

$\mathbf{z}_1, \mathbf{z}_2$  の更新式は、二次の近接項を含む最小化問題である。関数  $\mathcal{F}$  の近接作用素  $\text{prox}_{\lambda\mathcal{F}}$  は

$$\text{prox}_{\lambda\mathcal{F}}(\mathbf{v}) = \arg \min_{\mathbf{x}} \left( \mathcal{F}(\mathbf{x}) + \frac{1}{2\lambda} \|\mathbf{x} - \mathbf{v}\|_2^2 \right) \quad (8)$$

と定義される。したがって、 $\mathbf{z}_1, \mathbf{z}_2$  の更新式は近接作用素の計算と見なすことができる。近接作用素が解析的に書けるコスト関数は数多く知られている [4]。

## 4 実験

提案手法を用いた ADMM による IVA を、従来法である AuxIVA、および PDS による IVA (PDS-IVA) と比較する実験を行った。提案手法では、Alg. 1 中の近接作用素  $\text{prox}_{\lambda\mathcal{F}}$  として、 $l_{2,1}$  混合ノルムの近接作用素であるグループソフト閾値作用素を用いることで IVA を実現した。データセットには SiSEC 2011 の dev1 を用いた。残響時間は 130 ms, 250 ms, マイク間距離は 1 m, 5 cm とし、女声男声それぞれ 3 組を用いて 2 チャンネルの 2 話者混合音を作成した。視認性のために、得られた 24 組の混合音のうち、AuxIVA を含む多くの手法で分離に失敗した条件を除き、最終的なデータ数は 16 組となった。提案手法における最適化パラメータは  $\rho$  を 2.0, 3.0, 4.0,  $\alpha$  を 1.5, 1.6, 1.7 とした。反復回数はすべての手法で 100 回とし、評価指標には  $\Delta\text{SDR}$  を用いた。

5 反復ごとの  $\Delta\text{SDR}$  の平均値の推移を図-1 に示す。この結果から、提案手法は PDS-IVA より少ない反復回数で収束することが確認できた。特に、 $\rho = 3.0$  の時に安定して高速に収束することが分かった。

最終的な  $\Delta\text{SDR}$  の値を図-2 に示す。提案手法では、AuxIVA や PDS-IVA と同等またはそれ以上の性

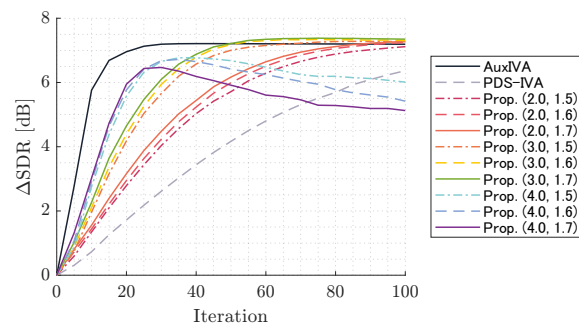


図-1 反復に伴う  $\Delta\text{SDR}$  の推移。提案手法における最適化パラメータを括弧内に  $(\rho, \alpha)$  の順で記載している。

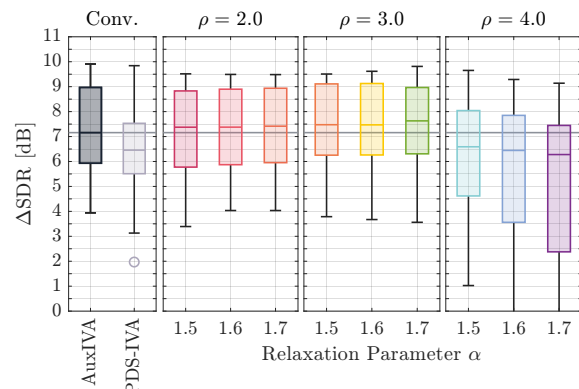


図-2 最終的な分離性能。1 列目は従来法の結果である。それ以降は提案手法の結果であり、それぞれの  $\rho$  について各  $\alpha$  ごとの性能を示している。 $\rho$  が 3.0 以下の時、提案手法は従来法と同等またはそれ以上の性能が得られた。

能が得られ、特に  $\rho = 3.0$  の時は従来法よりも分離性能が優れることが分かった。

以上より、適切にパラメータを設定することで、PDS-IVA より高速に収束し、AuxIVA と同等またはそれ以上の分離性能が得られることが確認できた。

## 5 むすび

本稿では、ADMM による優決定ブラインド音源分離手法を提案し、その有効性が収束速度と分離性能の両面で確認できた。今後の展望として、先行研究で提案されている、PDS アルゴリズム中の近接作用素を時間周波数マスキングで置き換える手法 [1] について、ADMM を用いて高速化することが挙げられる。

### 参考文献

- [1] K. Yatabe and D. Kitamura, "Determined BSS Based on Time-Frequency Masking and Its Application to Harmonic Vector Analysis," *IEEE/ACM Trans. on Audio Speech Lang. Process.*, **29**, 1609–1625, (2021).
- [2] L. Condat, D. Kitahara, A. Contreras and A. Hirabayashi, "Proximal Splitting Algorithms for Convex Optimization: A Tour of Recent Advances, with New Twists," *SIAM Rev.*, **65**(2), 375–435, (2023).
- [3] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Found. Trends Mach. Learn.*, **3**(1), 1–122, (2011).
- [4] N. Parikh and S. Boyd, "Proximal Algorithms," *Found. Trends Optim.*, **1**(3), 123–231, (2013).