制約付き補助関数法による 非負値テンソル因子分解を用いたスポットフォーミングの高速化*

◎綾野翔馬（香川高専），李莉，関翔悟（サイバーエージェント），北村大地（香川高専）

1 はじめに

スポットフォーミングとは観測信号から特定の領域に含まれる目的音源の信号のみを抽出する手法である [1]．スポットフォーミングの手法として，高いモデルの解釈性を持ち正則化の導入が可能な手法として，非負値テンソル因子分解 (nonngative tensor factorization: NTF) に基づくスポットフォーミングを提案した [2]．また，より識別的な学習を促進し，自動でクラスタリングを行うためのアトラクタ正則化を提案した [3]．本稿では，提案したアルゴリズムの高速化を目指して，制約付き補助関数法を用いた更新式の導出を行う．また，アトラクタ正則化に対して特殊なハイパーパラメータを用いることで更に高速な更新式を導出し，その動作を確認する．

2 NTF に基づくスポットフォーミング [3]

2.1 想定するモデル

スポットフォーミングにおいては，Fig. 1 に示すような，各マイクロホンアレイ内では同期録音しているが，マイクロホンアレイ間での同期は必ずしも保証されていない状況を考える．各マイクロホンアレイは，特定の方位の音源を強調するビームフォーミング (beamforming: BF) を用いる．Fig. 1 の場合，各マイクロホンアレイは正面方向に対して，BF を行うことで，目的音源を強調できるが，同一方位上に存在する干渉音源も強調されてしまう．一方で，干渉音源はマイクロホンアレイ毎に異なるため，各 BF の出力信号間で共通する音源成分を抽出できれば，目的音源のみが得られると考えられる．

2.2 定式化

NTF に基づくスポットフォーミングにおいては，各マイクロホンアレイに BF を適用して得られる A 個の振幅スペクトログラムによって構成される 3 次元非負テンソル $\mathsf{C} \in \mathbb{R}_+^{A\times I\times J}$ を入力として扱う．I，J は周波数ビン数および時間フレーム数であり，それぞれの小文字はインデクスを表す．NTF は，3 次元非負テンソルを分配行列 $\boldsymbol{Z} \in \left\{\boldsymbol{Z}\in[0,1]^{A\times K} \middle| \|\boldsymbol{z}_k\|_1 = 1\right\}$，基底行列 $\boldsymbol{T} \in \left\{\boldsymbol{T}\in[0,1]^{I\times K} \middle| \|\boldsymbol{t}_k\|_1 = 1\right\}$，および係数行列 $\boldsymbol{V} \in \mathbb{R}_+^{J\times K}$ を用いて近似する．NTF に基づくスポットフォーミングにおいては，次式に示す分解モデルを用いる．

$$c_{a,i,n} \approx \textstyle\sum_k z_{a,k} t_{i,k} v_{j,k} \tag{1}$$

更に，分解モデルに対して，アトラクタ正則化を導入する．アトラクタ正則化は，分配行列 $\boldsymbol{Z}$ の各列に対して，次式のいずれかのベクトル（これをアトラクタベクトルと呼ぶ）に近づける正則化である．

$$\begin{aligned}
\boldsymbol{p}_1 &:= [1/A, 1/A, \cdots, 1/A]^{\mathrm{T}} \in \{1/A\}^A,\\
\boldsymbol{p}_2 &:= [1, 0, \cdots, 0]^{\mathrm{T}} \in \{0,1\}^A,\\
\boldsymbol{p}_3 &:= [0, 1, \cdots, 0]^{\mathrm{T}} \in \{0,1\}^A,\\
&\ \vdots\\
\boldsymbol{p}_B &:= [0, 0, \cdots, 1]^{\mathrm{T}} \in \{0,1\}^A
\end{aligned} \tag{2}$$

ここで，B はアトラクタベクトルの本数であり，$B = A+1$ である．この正則化により，NTF により推定される基底が全チャネルに含まれる共通成分または単一のチャネルにのみ含まれる固有成分に誘導される．

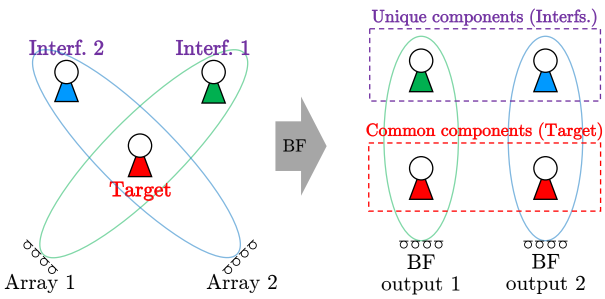

Fig. 1 Situations and signals estimated by BF.

アトラクタ正則化項を含めたコスト関数を次式で定義する．

$$\begin{aligned}
\mathcal{J} &:= \textstyle\sum_{a,i,j} \mathcal{D}(c_{a,i,j} | \sum_k z_{a,k} t_{i,k} v_{j,k})\\
&\quad + \mu \textstyle\sum_k \mathcal{R}(\boldsymbol{p}_{b_k} | \boldsymbol{z}_k)
\end{aligned} \tag{3}$$

ここで，$\mathcal{D}$ は 2 要素間のコスト関数であり，次式で示す一般化 Kullback–Leibler 擬距離を用いる．

$$\mathcal{D}(\gamma|\beta) = \gamma \log \frac{\gamma}{\beta} + \beta - \gamma \tag{4}$$

また，$\mathcal{R}$ はアトラクタ正則化項を表す関数であり，次式で定義される．

$$\mathcal{R}\left(\boldsymbol{p}_{b_k} | \boldsymbol{z}_k\right) := \sum_a \mathcal{D}(p_{a,b_k} | z_{a,k}) \tag{5}$$

$$b_k \in \operatorname*{argmin}_b \sum_a \mathcal{D}(p_{a,b} | z_{a,k}) \tag{6}$$

よって，アトラクタ正則化付き NTF は，次式の最適化問題となる．

$$\begin{aligned}
&\operatorname*{minimize}_{\boldsymbol{Z},\boldsymbol{T},\boldsymbol{V}}\ \mathcal{J}\\
&\text{s.t. } \boldsymbol{Z} \in \left\{\boldsymbol{Z}\in[0,1]^{A\times K} \middle| \|\boldsymbol{z}_k\|_1 = 1\right\},\\
&\qquad \boldsymbol{T} \in \left\{\boldsymbol{T}\in[0,1]^{I\times K} \middle| \|\boldsymbol{t}_k\|_1 = 1\right\}
\end{aligned} \tag{7}$$

*Fast algorithm for spotforming using nonnegative tensor factorization based on constrained-aware majorization-minimization algorithm By Shoma AYANO (NIT Kagawa), Li LI, Shogo SEKI (CyberAgent, Inc.), and Daichi KITAMURA (NIT Kagawa).

アトラクタ正則化付き NTF に基づくスポットフォーミングでは，ハイパーパラメータ μ に大きい値を設定することで一貫して高い性能を得ることが確認されている [3].

2.3 補助関数法に基づく更新式の導出

式 (7) の最適化問題を上界最小化アルゴリズム [4] を用いて解く．はじめに，式 (3) について，Jensen の不等式を用いて，次式に示す上界関数を考える．

$$\mathcal{J} \leq \mathcal{J}^+ \coloneqq \sum_{a,i,j} \left[-c_{a,i,j} \sum_k u_{a,i,j,k} \log \left(\frac{z_{a,k} t_{i,k} v_{j,k}}{u_{a,i,j,k}} \right) + \sum_k z_{a,k} t_{i,k} v_{j,k} \right] + \mu \sum_{a,k} (-p_{a,b_k} \log z_{a,k} + z_{a,k}) + \text{const.} \tag{8}$$

ここで，$u_{a,i,j,k}$ は補助変数であり，これをまとめた集合を U とする．上界関数に対して，以下の最適化問題を考える．

$$\underset{\boldsymbol{Z},\boldsymbol{T},\boldsymbol{V},\mathsf{U}}{\text{minimize}}\ \mathcal{J}^+ \tag{9}$$

式 (8) を各変数について微分し，$\frac{\partial \mathcal{J}^+}{\partial z_{a,k}} = 0$ となる点を探すことで 式 (9) を最小化する．分配行列 $\boldsymbol{Z}$ での微分は次式となる．

$$\sum_{i,j} \left(-c_{a,i,j} u_{a,i,j,k} \frac{1}{z_{a,k}} + t_{i,k} v_{j,k} \right) + \mu \left(-\frac{p_{a,b_k}}{z_{a,k}} + 1 \right) = 0 \tag{10}$$

分配行列 $\boldsymbol{Z}$ について解くと次式を得る．

$$z_{a,k} = \frac{\sum_{i,j} c_{a,i,j} u_{a,i,j,k} + \mu p_{a,b_k}}{\sum_{i,j} t_{i,k} v_{j,k} + \mu} \tag{11}$$

式 (11) により，コスト関数 (8) の最小化を行うことができる．ここで，補助変数に対して等号を満たす変数を代入することで，変数更新後のコスト関数 (8) は変数更新前よりも同値か小さい値となる．式 (8) は，以下の条件を満たす場合に等号が成立する．

$$u_{a,i,j,k} = \frac{z_{a,k} t_{i,k} v_{j,k}}{\sum_{k'} z_{a,k'} t_{i,k'} v_{j,k'}} \tag{12}$$

式 (12) を式 (11) に代入することで，式 (7) の反復更新式を次式として得る．

$$z_{a,k} \leftarrow \frac{z_{a,k} \sum_{i,j} c_{a,i,j} \frac{t_{i,k} v_{j,k}}{\sum_{k'} z_{a,k'} t_{i,k'} v_{j,k'}} + \mu p_{a,b_k}}{\sum_{i,j} t_{i,k} v_{j,k} + \mu} \tag{13}$$

ここで，分配行列 $\boldsymbol{Z}$ が制約を満たすように，次式の正規化を行う．

$$v_{j,k} \leftarrow v_{j,k} \|\boldsymbol{z}_k\|_1,\ z_{a,k} \leftarrow z_{a,k} \|\boldsymbol{z}_k\|_1^{-1} \tag{14}$$

これらの正規化はコスト関数 (3) の値を変えないことに注意する．分配行列 $\boldsymbol{Z}$ の更新により，最も近いアトラクタベクトルが変化することがあるため，分配行列の更新の事前に式 (6) を用いて最も近いアトラクタベクトルを更新する．次に，基底行列 $\boldsymbol{T}$ について同様にして次の更新式を得る．

$$t_{i,k} \leftarrow \frac{t_{i,k} \sum_{a,j} c_{a,i,j} \frac{z_{a,k} v_{j,k}}{\sum_{k'} z_{a,k'} t_{i,k'} v_{j,k'}}}{\sum_{a,j} z_{a,k} v_{j,k}} \tag{15}$$

$$v_{j,k} \leftarrow v_{j,k} \|\boldsymbol{t}_k\|_1,\ t_{i,k} \leftarrow t_{i,k} \|\boldsymbol{t}_k\|_1^{-1} \tag{16}$$

これらの正規化も同様にコスト関数 (3) の値を変えないことに注意する．最後に，係数行列 $\boldsymbol{V}$ の更新式を得る．

$$v_{j,k} \leftarrow \frac{v_{j,k} \sum_{a,i} c_{a,i,j} \frac{z_{a,k} t_{i,k}}{\sum_{k'} z_{a,k'} t_{i,k'} v_{j,k'}}}{\sum_{a,i} z_{a,k} t_{i,k}} \tag{17}$$

式 (6) および式 (13)–(17) を繰り返し更新することでコスト関数 (3) を最小化し，変数行列 $\boldsymbol{Z}$，$\boldsymbol{T}$，および $\boldsymbol{V}$ が推定される．

アトラクタ正則化項の重み係数であるハイパーパラメータ μ を大きくすることで，更に識別的な正則化となる．次節では，制約付き補助関数法および特殊なハイパーパラメータを用いて，より高速な更新式を導出する．

3 制約付き補助関数法に基づく反復更新式の導出

前章では，一般的な補助関数法に基づく NTF の導出を行った．ここで，最適化問題 (9) および分配行列の更新式 (14) に注目すると，上界関数を最小化した後に正則化係数を乗じることで分配行列の制約を満たしていた．一方，変数行列の制約を常に満たすような最適化問題を考えることで，正規化係数を乗じる計算を省略し，反復更新を高速化することができる．

上界関数に対して以下の最適化問題を考える[1]．

$$\underset{\boldsymbol{Z},\boldsymbol{T},\boldsymbol{V},\mathsf{U}}{\text{minimize}}\ \mathcal{J}^+ \quad \text{s.t. } \boldsymbol{Z} \in \left\{ \boldsymbol{Z} \in [0,1]^{A \times K} \,\middle|\, \|\boldsymbol{z}_k\|_1 = 1 \right\} \tag{18}$$

この最適化問題に対して Lagrange の未定乗数法 [6] を用いて解く．分配行列の制約式から，Lagrange 関数は次式で定義される．

$$\mathcal{L}(\boldsymbol{Z}, \boldsymbol{\lambda}) \coloneqq \mathcal{J}^+ + \textstyle\sum_k \lambda_k (\|\boldsymbol{z}_k\|_1 - 1) \tag{19}$$

ここで，λ_k は Lagrangian 乗数であり，これをまとめたベクトルを $\boldsymbol{\lambda} \in \mathbb{R}^K$ とおく．式 (19) について，全ての変数 $\boldsymbol{Z}$ および $\boldsymbol{\lambda}$ で微分し，$\frac{\partial \mathcal{L}}{\partial z_{a,k}} = 0$ および $\frac{\partial \mathcal{L}}{\partial \lambda_k} = 0$ となる点が解となる．式 (19) の各変数での

[1] 文献 [5] においても同様に補助関数の最小化に制約条件を付与しているが，本稿とは解法が異なる．

Table 1　Update rules of algorithms

Method	Update $\boldsymbol{b}$	Update $\boldsymbol{Z}$	Normalize $\boldsymbol{Z}$	Update $\boldsymbol{T}$	Normalize $\boldsymbol{T}$	Update $\boldsymbol{V}$
Naive	(6)	(13)	(14)	(15)	(16)	(17)
Constrained		(25)		(26)		
Constrained ($\mu \to \infty$)	(6) & (28) only once					

微分は次式となる．

$$\sum_{i,j}\left(-c_{a,i,j}u_{a,i,j,k}\frac{1}{z_{a,k}}+t_{i,k}v_{j,k}\right) + \mu\left(-\frac{p_{a,b_k}}{z_{a,k}}+1\right)+\lambda_k=0 \tag{20}$$

$$\|\boldsymbol{z}_k\|_1-1=0 \tag{21}$$

式 (20) を $z_{a,k}$ について解くと次式を得る．

$$z_{a,k}=\frac{\sum_{i,j}\left(c_{a,i,j}u_{a,i,j,k}+\mu p_{a,b_k}\right)}{\sum_{i,j}t_{i,k}v_{j,k}+\mu+\lambda_k} \tag{22}$$

この式に対して，両辺を a に対する和をとり，式 (21) を代入することで次式を得る．

$$\frac{\sum_{a,i,j}\left(c_{a,i,j}u_{a,i,j,k}+\mu p_{a,b_k}\right)}{\sum_{i,j}t_{i,k}v_{j,k}+\mu+\lambda_k}=1 \tag{23}$$

式 (20) および式 (23) より，次式を得る．

$$z_{a,k}=\frac{\sum_{i,j}c_{a,i,j}u_{a,i,j,k}+\mu p_{a,b_k}}{\sum_a\left(\sum_{i,j}c_{a,i,j}u_{a,i,j,k}+\mu p_{a,b_k}\right)} \tag{24}$$

最後に，補助変数に対して等号を満たす変数を代入することで，次式の更新式を得る．

$$z_{a,k}\leftarrow \frac{z_{a,k}\sum_{i,j}c_{a,i,j}\frac{t_{i,k}v_{j,k}}{\sum_{k'}z_{a,k'}t_{i,k'}v_{j,k'}}+\mu p_{a,b_k}}{\sum_{a'}\left(z_{a',k}\sum_{i,j}c_{a',i,j}\frac{t_{i,k}v_{j,k}}{\sum_{k'}z_{a',k'}t_{i,k'}v_{j,k'}}+\mu p_{a',b_k}\right)} \tag{25}$$

同様にして，基底行列 $\boldsymbol{T}$ について同様にして次の更新式を得る．

$$t_{i,k}\leftarrow \frac{t_{i,k}\sum_{a,j}c_{a,i,j}\frac{z_{a,k}v_{j,k}}{\sum_{k'}z_{a,k'}t_{i,k'}v_{j,k'}}}{\sum_{i'}\left(t_{i',k}\sum_{a,j}c_{a,i',j}\frac{z_{a,k}v_{j,k}}{\sum_{k'}z_{a,k'}t_{i',k'}v_{j,k'}}\right)} \tag{26}$$

係数行列 $\boldsymbol{V}$ には制約はないため，式 (17) の更新式を用いる．よって，制約付き補助関数法に基づく更新式では，式 (6)，式 (25)，式 (26)，および式 (17) を繰り返し更新することで最適化問題 (3) を最小化する．通常の更新式との差分は，正規化に伴って係数行列 $\boldsymbol{V}$ にかけていた係数をかけていない点である．また，分子と分母が総和記号を除いて一致することから，分子を計算した後にそれを用いて分母を計算することで，更新式の分母の計算量が小さくなることから，より高速な更新式であるといえる．

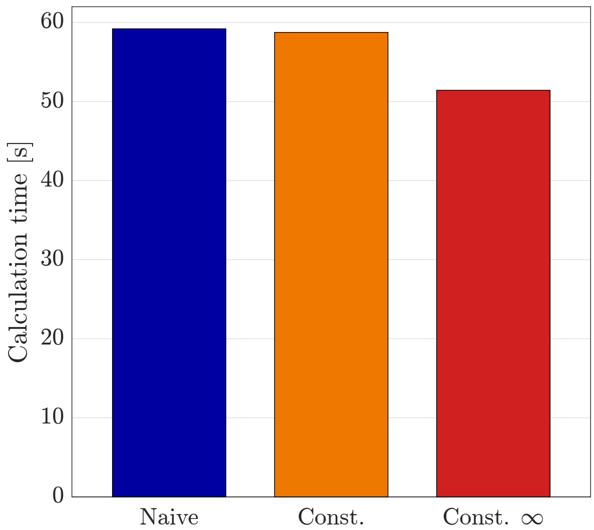

Fig. 2　Calculation time of each update rule.

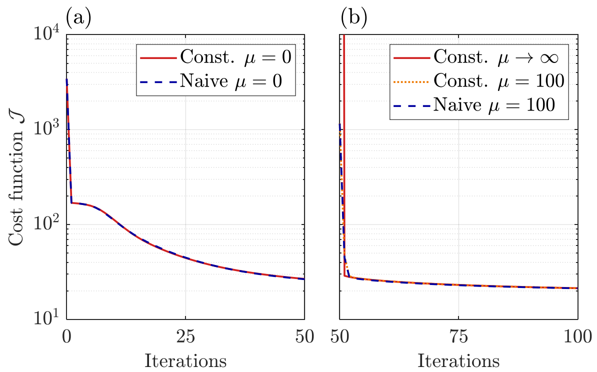

Fig. 3　Cost function behavior: (a) first half and (b) last half iterations.

3.1　ハイパーパラメータの設定による更新式の導出

スポットフォーミングにおいては，アトラクタ正則化のハイパーパラメータ μ を大きい値することで一貫して高い性能を示すことが解っている [3]．そこで，アトラクタ正則化項の重み係数を $\mu \to \infty$ へと近づける場合を考える．分配行列 $\boldsymbol{Z}$ の更新式は次式となる．

$$z_{a,k}\leftarrow \lim_{\mu\to\infty}\left[\frac{z_{a,k}\sum_{i,j}c_{a,i,j}\frac{t_{i,k}v_{j,k}}{\sum_{k'}z_{a,k'}t_{i,k'}v_{j,k'}}+\mu p_{a,b_k}}{\sum_{a'}\left(z_{a',k}\sum_{i,j}c_{a',i,j}\frac{t_{i,k}v_{j,k}}{\sum_{k'}z_{a',k'}t_{i,k'}v_{j,k'}}+\mu p_{a',b_k}\right)}\right] \tag{27}$$

右辺を整理すると，分配行列の各列 $\boldsymbol{z}_k$ は最も近いアトラクタベクトル $\boldsymbol{p}_{b_k}$ と一致する．従って，分配行列

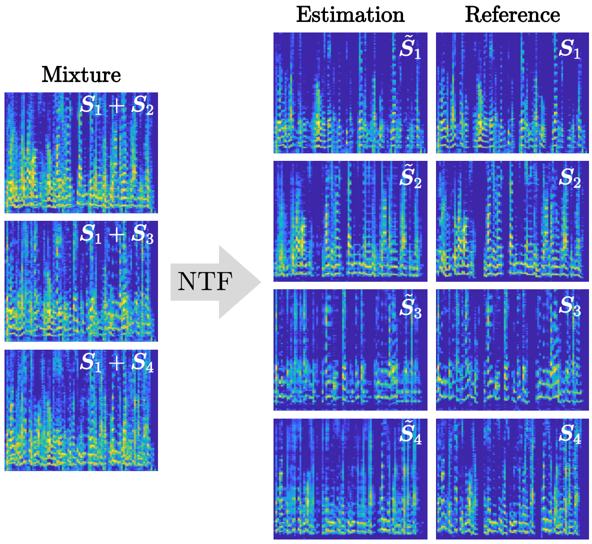

Fig. 4 Result of NTF decomposition ($\mu \to \infty$).

Algorithm 1 Attractor-regularized NTF ($\mu \to \infty$)

Input: Initial value of matrices ($\boldsymbol{Z}$, $\boldsymbol{T}$, and $\boldsymbol{V}$)
Output: $\boldsymbol{b}$, $\boldsymbol{Z}$, $\boldsymbol{T}$, and $\boldsymbol{V}$

for iter = 1 to 100 **do**
  if iter $\leq$ 51 **then**
    Update $\boldsymbol{b}$ by Eq. (6)
  end if
  if iter $\leq$ 50 **then**
    Update $\boldsymbol{Z}$ by Eq. (25)
  else if iter = 51 **then**
    Update $\boldsymbol{Z}$ by Eq. (28)
  end if
  Update $\boldsymbol{T}$ by Eq. (26)
  Update $\boldsymbol{V}$ by Eq. (17)
end for
return $\boldsymbol{b}$, $\boldsymbol{Z}$, $\boldsymbol{T}$, and $\boldsymbol{V}$

$\boldsymbol{Z}$ の更新式は以下の式となる．

$$z_{a,k} \leftarrow p_{a,b_k} \tag{28}$$

この更新式により，分配行列 $\boldsymbol{Z}$ の更新式は式 (28) に示す 1 回のみの代入となり，NTF の動作を更に高速化することができる．ここまで導出した更新式を Table 1 にまとめる．

4 実験

4.1 実験条件

LibriTTS [7] からランダムに選んだ 4 種類の音源に対して，振幅スペクトログラムを得た後，それらを混合して 3 次元テンソルを構成した．構成した 3 次元テンソルに対して，Table 1 の各アルゴリズムを用いたアトラクタ正則化付き NTF を適用した．基底数 K は 100 とし，合計で 100 回の反復更新行い，ハイパーパラメータ μ は最初の 50 反復で 0，最後の 50 反復で 100 または ∞ とした．$\mu \to \infty$ としたときの疑似コードを Algorithm 1 に示す．分配行列の各列 $\boldsymbol{z}_k$ の初期値には $\boldsymbol{p}_1$ を用い，その他の行列の初期値は区間 $[0, 1)$ の一様乱数から生成した．各アルゴリズムの計算時間およびコスト関数の挙動を比較する．また，制約付き補助関数法に基づく更新式 ($\mu \to \infty$) を用いた場合の分解例を示す．

4.2 実験結果

Fig. 2 に各アルゴリズムの計算時間を示す．制約付き補助関数法に基づく更新式が通常の補助関数法に基づく更新式よりもわずかに高速に動作した．特に，$\mu \to \infty$ の更新式は特に高速に動作した．

Fig. 3 に各アルゴリズムのコスト関数の挙動を示す．各アルゴリズムが (コスト関数の切り替わりのタイミングを除いて) 非増加で推移していることが確認できた．全ての更新式で 51 回目の反復でアトラクタ正則化が有効化され，$\mu = 100$ の更新式においては 2 回の反復で大きくコスト関数を減少させている．

Fig. 4 に制約付き補助関数法に基づく更新式 ($\mu \to \infty$) を用いた場合の分解例を示す．$\boldsymbol{S}_1$ は共通成分であり，$\boldsymbol{S}_2$，$\boldsymbol{S}_3$，および $\boldsymbol{S}_4$ は各チャネルの固有成分である．$\mu \to \infty$ としたアトラクタ正則化により，各基底が共通成分および固有成分に自動的に振り分けられ，各成分がよく近似できることが確認できた．なお，文献 [3] の Fig. 6 と同様の実験条件において，$\mu = 1000$(a) では 13.60 dB，(b) では 8.80 dB であったケースにおいて，$\mu \to \infty$ を用いることで (a) では 13.61 dB，(b) では 8.79 dB と，$\mu = 1000$ を含む大きいハイパーパラメータを用いたケースと同等の結果が得られた．

5 おわりに

本稿では，制約付き補助関数法を用いてアトラクタ正則化付き NTF の反復更新式を導出した．制約付き補助関数法により得られた反復更新式について，特に $\mu \to \infty$ とした場合に高速な動作が確認できた．従って，スポットフォーミングにおいては，$\mu \to \infty$ としたアトラクタ正則化 NTF が有効的であるといえる．

参考文献

[1] Y. Kagimoto, K. Itoyama, K. Nishida, and K. Nakadai, “Spotforming by NMF using multiple microphone arrays,” *Proc. IROS*, pp. 9253–9258, 2022.
[2] 綾野翔馬，李莉，関翔悟，北村大地，“非負値テンソル因子分解に基づく分散マイクロホンアレイを用いたスポットフォーミング,” **日本音響学会** *2024* **年春季研究発表会講演論文集**, pp. 137–140, 2024.
[3] S. Ayano, L. Li, S. Seki, and D. Kitamura, “Audio spotforming using nonnegative tensor factorization with attractor-based regularization,” *Proc. EUSIPCO*, pp. 121–125, 2024.
[4] D. R. Hunter and K. Lange, “Quantile regression via an MM algorithm,” *J. Comput. Graph. Stat.*, vol. 9, no. 1, pp. 60–77, 2000.
[5] V. Leplat, N. Gillis, and J. Idier, “Multiplicative updates for NMF with β-divergences under disjoint equality constraints,” *SIAM J. Matrix Anal. Appl.*, vol. 42, no. 2, pp. 730–752, 2021.
[6] D. P. Bertsekas, Constrained optimization and Lagrange multiplier methods, Academic press, 2014.
[7] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, “LibriTTS: a corpus derived from librispeech for text-to-speech,” *Proc. INTERSPEECH*, pp. 1526–1530, 2019.