Frobenius ノルム規準の非負値行列因子分解における乗法更新式に関する一考察*

○亀岡弘和，ルルージョナトン

日本電信電話 (株) NTT コミュニケーション科学基礎研究所

1 序 論

非負値行列因子分解 (以後，NMF) とは，与えられた非負値行列 $\mathbf{Y} = (y_{m,n})_{M\times N} \in \mathbb{R}^{\geq 0, M\times N}$ に対し，

$$\mathbf{Y} \simeq \mathbf{HU} \tag{1}$$

となるような非負値行列 $\mathbf{H} = (h_{m,k})_{M\times K} \in \mathbb{R}^{\geq 0, M\times K}$, $\mathbf{U} = (u_{k,n})_{K\times N} \in \mathbb{R}^{\geq 0, K\times N}$ を決定することをいい，非負信号のブラインド信号分離への応用技術として脳・画像・音響信号処理などの分野で注目されている。

式 (1) の解法として，$\mathbf{Y}$ と $\mathbf{HU}$ の近さを誤差行列 $\mathbf{Y}-\mathbf{HU}$ の Frobenius ノルムで測った場合の最適化問題

$$\begin{array}{ll} \text{minimize} & f(\mathbf{H},\mathbf{U}) = \|\mathbf{Y}-\mathbf{HU}\|_F^2 \\ \text{subject to} & \mathbf{H} \in \mathbb{R}^{\geq 0, M\times K},\ \mathbf{U} \in \mathbb{R}^{\geq 0, K\times N} \end{array} \tag{2}$$

を解く効率的な反復アルゴリズムが Lee らによって与えられている [1]。このアルゴリズムは，

$$h_{m,k} \leftarrow h_{m,k} \frac{\left[\mathbf{YU}^{\mathrm{T}}\right]_{m,k}}{\left[\mathbf{HUU}^{\mathrm{T}}\right]_{m,k}} \tag{3}$$

$$u_{k,n} \leftarrow u_{k,n} \frac{\left[\mathbf{H}^{\mathrm{T}}\mathbf{Y}\right]_{k,n}}{\left[\mathbf{H}^{\mathrm{T}}\mathbf{HU}\right]_{k,n}} \tag{4}$$

のような乗算型の更新式からなるため，乗法更新法と呼ばれることがある。$[\cdot]_i$, $[\cdot]_{i,j}$ はそれぞれベクトルの i 番目の要素，行列の $\{i,j\}$ 番目の要素を表すものとする。

本報告では，Lee らによって導かれたこの反復アルゴリズム (以後，Lee-Seung(LS) アルゴリズム)[1] がなぜ効果的であるかについて一考察を与え，これをもとに，より効率的な反復アルゴリズムが得られる可能性について議論する。また，この議論を通して，複素 NMF[2] における最適化アルゴリズムの効率化の可能性についても言及する。

2 LS アルゴリズム [1]

以下，LS アルゴリズムの導出の概要を示す。$\mathbf{H}$, $\mathbf{U}$ はともに変数であるため，式 (2) は非負制約つき非線形最適化問題であり，解析的に解くことはできない。これに対し，LS アルゴリズムは，制約を満たしながら目的関数を反復的に降下する手法であり，次に示す補助関数法の原理に基づく。

目的関数 $G(z)$ に対し，

$$G(z) = \min_{\bar{z}} G^{+}(z,\bar{z}) \tag{5}$$

が成り立つとき，$G^{+}(z,\bar{z})$ を $G(z)$ の**補助関数**，$\bar{z}$ を**補助変数**と定義する。このとき，次の定理が成り立つ。

定理 1 (補助関数法)**.** 補助関数 $G^{+}(z,\bar{z})$ を，$\bar{z}$ に関して最小化するステップと，z に関して最小化するステップを繰り返すと，目的関数値は単調収束する。

証明： $\bar{z}_{t+1} = \mathrm{argmin}_{\bar{z}}\, G^{+}(z_t,\bar{z})$ とし，$z_{t+1} = \mathrm{argmin}_{z}\, G^{+}(z,\bar{z}_{t+1})$ とする。ただし，t は反復計算のステップ数を表す。$z = z_t$, $\bar{z} = \bar{z}_t$ から $z = z_{t+1}$, $\bar{z} = \bar{z}_{t+1}$ に更新されたときに，$G(z)$ が増加しないことを示す。明らかに $G(z_t) = G^{+}(z_t,\bar{z}_{t+1})$ であり，$z_{t+1} = \mathrm{argmin}_{z}\, G^{+}(z,\bar{z}_{t+1})$ より，$G^{+}(z_t,\bar{z}_{t+1}) \geq G^{+}(z_{t+1},\bar{z}_{t+1})$ である．更に補助関数の定義より，$G^{+}(z_{t+1},\bar{z}_{t+1}) \geq G(z_{t+1})$ なのだから，結局，$G(z_t) \geq G(z_{t+1})$ である。 □

以上を踏まえ，式 (4) を導出する。ここで，$\mathbf{y}_n := (y_{1,n},\cdots,y_{M,n})^{\mathrm{T}}$ と $\mathbf{H}$ が与えられたとき，

$$g(\mathbf{u}_n) := \|\mathbf{y}_n - \mathbf{Hu}_n\|_2^2 \tag{6}$$

を $\mathbf{u}_n := (u_{1,n},\cdots,u_{K,n})^{\mathrm{T}}$ に関して最小化する問題を考えよう。式 (6) を直接最小化する $\mathbf{u}_n$ は $M \geq K$ のとき $\mathbf{u}_n = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{y}_n$ で与えられるが，この場合 $\mathbf{u}_n$ の各要素が非負性を満たすかどうかは定かではない。LS アルゴリズムのポイントは，$g(\mathbf{u}_n)$ に対し要素同士の交差項が存在しない二次形式の補助関数を立てる点にあり，これにより各要素について非負値制約に注意しながら独立に更新式を求めることができる。このような補助関数は次のようにして立てられる。まず，

$$\begin{aligned} g^{+}(\mathbf{u}_n,\mathbf{u}'_n) :=& g(\mathbf{u}'_n) + (\mathbf{u}_n - \mathbf{u}'_n)^{\mathrm{T}}\nabla g(\mathbf{u}'_n) \\ &+ \frac{1}{2}(\mathbf{u}_n - \mathbf{u}'_n)^{\mathrm{T}}\mathbf{W}(\mathbf{u}_n - \mathbf{u}'_n) \end{aligned} \tag{7}$$

と置く。ここで，$\mathbf{W}$ が

$$\mathbf{W} = \mathrm{diag}(w_1,\cdots,w_K) \tag{8}$$

$$w_k = 2\left[\mathbf{H}^{\mathrm{T}}\mathbf{Hu}'_n\right]_k / u'_{k,n} \tag{9}$$

*Study on Multiplicative Update Formula for Nonnegative Matrix Factorization with Frobenius Norm Criterion. by Hirokazu Kameoka, Jonathan Le Roux (NTT Communication Science Laboratories)

で与えられるとき，

$$g(\mathbf{u}_n) \leq g^+(\mathbf{u}_n, \mathbf{u}'_n) \tag{10}$$

が言える ($\mathbf{W}-2\mathbf{H}^{\mathrm{T}}\mathbf{H}$ が半正定値であることによる [1])。明らかに $\mathbf{u}'_n = \mathbf{u}_n$ のとき $g(\mathbf{u}_n) = g^+(\mathbf{u}_n, \mathbf{u}'_n)$ であるため，$g^+(\mathbf{u}_n, \mathbf{u}'_n)$ は $g(\mathbf{u}_n)$ の補助関数としての定義を満たす。従って，定理 1 より，$g^+(\mathbf{u}_n, \mathbf{u}'_n)$ を $\mathbf{u}'_n$ と $\mathbf{u}_n$ に関してそれぞれ最小化するステップ

$$\mathbf{u}'_n \leftarrow \mathbf{u}_n \tag{11}$$

$$\mathbf{u}_n \leftarrow \mathbf{u}'_n - \mathbf{W}^{-1}\nabla g(\mathbf{u}'_n) \tag{12}$$

を行えば $g(\mathbf{u}_n)$ を降下させることができる。式 (11), (12) をまとめて，さらに要素ごとの表現に書き直すと，

$$u_{k,n} \leftarrow u_{k,n}\frac{\left[\mathbf{H}^{\mathrm{T}}\mathbf{y}_n\right]_k}{\left[\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{u}_n\right]_k} \tag{13}$$

となり，式 (4) を得る。$g^+(\mathbf{u}_n, \mathbf{u}'_n)$ は $\mathbf{u}_n$ の要素 $u_{k,n}$ ごとの二次関数に分離されており，式 (13) の更新値はその放物線の頂点の座標に該当する。従って，式 (13) がもし負であれば $u_{k,n} \geq 0$ の制約下で $g^+(\mathbf{u}_n, \mathbf{u}'_n)$ が最小となるのは $u_{k,n} = 0$ のときである。このため，正確には

$$u_{k,n} \leftarrow \max\left\{u_{k,n}\frac{\left[\mathbf{H}^{\mathrm{T}}\mathbf{y}_n\right]_k}{\left[\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{u}_n\right]_k}, 0\right\} \tag{14}$$

と書かれるべきであるが，$\mathbf{H}$, $\mathbf{y}_n$, $\mathbf{u}_n$ の要素がすべて非負値であれば式 (13) の更新では $\mathbf{u}_n$ は必ず非負値に更新されるため，$\mathbf{H}$ と $\mathbf{U}$ の初期値が非負であることが前提となっている場合は更新式の表現は式 (13) のままで十分である。このように更新値の非負性が逐次的に保証される点は LS アルゴリズムの特徴の一つである。式 (3) も以上と同様の考え方により得られる。

3　一般化 LS アルゴリズム

以下，同様に $\mathbf{u}_n$ に関する更新法についてのみ議論する。また，表記の簡単化のため n は以後省略する。

以上で見てきたように，LS アルゴリズムのポイントは，$g(\mathbf{u})$ を直接最小化しようとするのではなく，$\mathbf{u}$ の要素同士の交差項が存在せず要素ごとの二次関数に分離された形態をもつ補助関数を設計することにあった。実は，以上のような性質の補助関数は，

$$\sum_k \alpha_m^{(k)} = 1,\ \sum_k \beta_m^{(k)} = 1,\ \beta_m^{(k)} > 0 \tag{15}$$

を満たす任意の $\alpha_m^{(k)}$, $\beta_m^{(k)}$ のもとで成り立つ不等式

$$g(\mathbf{u}) \leq \sum_{m,k}\frac{1}{\beta_m^{(k)}}\left(\alpha_m^{(k)}y_m - h_{m,k}u_k\right)^2 \tag{16}$$

を用いて無数に作ることができる。上記不等式の証明については [2] を参照されたい。この不等式は，$\beta_m^{(k)}$ の値に依らず $\alpha_m^{(k)}$ が

$$\alpha_m^{(k)} = \frac{1}{y_m}\left[h_{m,k}u_k + \beta_m^{(k)}\left(y_m - \left[\mathbf{H}\mathbf{u}\right]_m\right)\right] \tag{17}$$

のとき等号成立 [2] するため，式 (16) の右辺

$$\mathcal{G}^+(\mathbf{u}, \alpha) := \sum_{m,k}\frac{1}{\beta_m^{(k)}}\left(\alpha_m^{(k)}y_m - h_{m,k}u_k\right)^2 \tag{18}$$

は $g(\mathbf{u})$ の補助関数としての定義を満たし，$\alpha_m^{(k)}$ が補助変数となる。$\mathcal{G}^+(\mathbf{u}, \alpha)$ はベクトル要素 u_k 同士の交差項のない二次関数の和になっており，このような性質の補助関数が $\beta_m^{(k)}$ の任意性の分，無数に定義できたことになる。ここで注目すべきは，2 章で立てた補助関数が，

$$\beta_m^{(k)} = \frac{h_{m,k}u_k}{\left[\mathbf{H}\mathbf{u}\right]_m} \tag{19}$$

と置いた場合の $\mathcal{G}^+(\mathbf{u}, \alpha)$ の特殊形に相当している点である。これを以下で確認しよう。

式 (17) に式 (19) を代入すると

$$\alpha_m^{(k)} = \frac{h_{m,k}u_k}{\left[\mathbf{H}\mathbf{u}\right]_m} \tag{20}$$

が得られ，$\beta_m^{(k)} = \alpha_m^{(k)}$ となる。式 (19), (20) での u_k は式 (18) の u_k と区別する必要があるため u'_k と置いた上で式 (19), (20) を式 (18) に代入すると

$$\mathcal{G}^+(\mathbf{u}, \alpha) = \sum_{m,k}\frac{h_{m,k}u'_k}{\left[\mathbf{H}\mathbf{u}'\right]_m}\left(y_m - \frac{u_k}{u'_k}\left[\mathbf{H}\mathbf{u}'\right]_m\right)^2 \tag{21}$$

を得る。これは，式 (7) を要素ごとの表記に書き直したものと等しい。なお，ここでは $\mathbf{H}$ を定数と見なしているので α を補助変数と見なすことは $\mathbf{u}'$ を補助変数と見なすことと同じである。以上より，$\mathcal{G}^+(\mathbf{u}, \alpha)$ は $g^+(\mathbf{u}, \mathbf{u}')$ を包含する一般形であることが分かった。

さて，定理 1 より，

$$\alpha_m^{(k)} \leftarrow \underset{\alpha_m^{(k)}}{\operatorname{argmin}}\, \mathcal{G}^+(\mathbf{u}, \alpha) \tag{22}$$

$$u_k \leftarrow \underset{u_k}{\operatorname{argmin}}\, \mathcal{G}^+(\mathbf{u}, \alpha) \tag{23}$$

を行えば $g(\mathbf{u})$ を降下させることができる。ここで，式 (22) は式 (17) で与えられ，式 (23) は

$$u_k = \frac{\sum_m \alpha_m^{(k)}y_m h_{m,k}/\beta_m^{(k)}}{\sum_m h_{m,k}^2/\beta_m^{(k)}} \tag{24}$$

で与えられる。$h_{m,k}$ の更新方法の一般形についても以上と同様にして得られる。$h_{m,k}$ の更新式を含めて式 (22), (23) により構成される反復アルゴリズムを一般化 LS アルゴリズムと呼ぶことにする。

4 LS アルゴリズムの正当性について

3 章では LS アルゴリズムは一般化 LS アルゴリズムにおいて $\beta_m^{(k)}$ を式 (19) と置いた特殊ケースに相当することを示した。ここでは，その合理性について考察する。

まず，$\beta_m^{(k)}$ の役割について考えよう。式 (18) を見ると明らかなように，$\beta_m^{(k)}$ は，どの $\mathbf{u}$ においても $g(\mathbf{u})$ を決して下回らない $\mathcal{G}^+(\mathbf{u},\alpha)$ の選び方の中で，u_k ごとの二次関数の頂点の座標を調節する変数と見ることができる。従って，式 (23) において $\mathcal{G}^+(\mathbf{u},\alpha)$ をどれだけ降下できるかは，$\beta_m^{(k)}$ の選び方次第となる。以上より，u_k の二次関数の頂点の値ができるだけ小さくなるように $\beta_m^{(k)}$ を決めることができれば，反復アルゴリズムを効率化できるはずである。

以上の観点に立ち，各ステップで $\beta_m^{(k)}$ を最適に決定する問題を定式化する。$\mathcal{G}^+(\mathbf{u},\alpha)$ は，

$$\mathcal{G}^+(\mathbf{u},\alpha) = \sum_k \left\{ \left(\sum_m \frac{{h_{m,k}}^2}{\beta_m^{(k)}} \right) {u_k}^2 - 2\left(\sum_m \frac{\alpha_m^{(k)} y_m h_{m,k}}{\beta_m^{(k)}} \right) u_k + \sum_m \frac{\alpha_m^{(k)2} {y_m}^2}{\beta_m^{(k)}} \right\}$$

と書けるので，$\mathcal{G}^+(\mathbf{u},\alpha)$ の $\mathbf{u}$ に関して最小となる値は $\beta_m^{(k)}$ の関数となり，

$$\mathcal{F}(\beta) := \sum_k \left\{ \sum_m \frac{\alpha_m^{(k)2} {y_m}^2}{\beta_m^{(k)}} - \left(\sum_m \frac{\alpha_m^{(k)} y_m h_{m,k}}{\beta_m^{(k)}} \right)^2 \Big/ \left(\sum_m \frac{{h_{m,k}}^2}{\beta_m^{(k)}} \right) \right\} \tag{25}$$

で与えられる。$\beta_m^{(k)}$ には式 (15) の制約があるため，

$$\begin{array}{ll} \text{minimize} & \mathcal{F}(\beta) \\ \text{subject to} & \sum_k \beta_m^{(k)} = 1 \ \text{ for all } m \end{array} \tag{26}$$

を解く問題に帰着する。この最適化問題は解析的に解くことはできないが，実は式 (19) の結果は，$\mathcal{F}(\beta)$ の第一項を最小化する解となっている。これを以下で確認しよう。式 (15) の制約に対応するラグランジュ未定乗数 λ を導入し，ラグランジュ関数を

$$\mathcal{L}(\beta,\lambda) := \sum_k \sum_m \frac{\alpha_m^{(k)2} {y_m}^2}{\beta_m^{(k)}} + \sum_m \lambda_m \left(\sum_k \beta_m^{(k)} - 1 \right)$$

と定義する。これを $\beta_m^{(k)}$ に関して偏微分し，0 と置くと

$$\beta_m^{(k)} = \frac{\alpha_m^{(k)} y_m}{\sqrt{\lambda_m}} \tag{27}$$

が得られるので，式 (15) に代入し，

$$\sum_k \frac{\alpha_m^{(k)} y_m}{\sqrt{\lambda_m}} = 1 \ \Rightarrow \ \lambda_m = {y_m}^2 \left(\sum_k \alpha_m^{(k)} \right)^2 \tag{28}$$

を式 (27) に代入すると

$$\beta_m^{(k)} = \frac{\alpha_m^{(k)}}{\sum_k \alpha_m^{(k)}} = \alpha_m^{(k)} \tag{29}$$

となる。これを式 (17) に代入すると，最終的に

$$\alpha_m^{(k)} = \beta_m^{(k)} = \frac{h_{m,k} u_k}{\left[\mathbf{Hu}\right]_m} \tag{30}$$

が導かれる。

以上より，LS アルゴリズムは，一般化 LS アルゴリズムの枠組から見たとき，各ステップにおいて目的関数の降下量を最大にする $\beta_m^{(k)}$ が選択されているわけではないが，式 (26) のある種の近似解が選択されていることが明らかになった。このことは，LS アルゴリズムが効率的に動作することに対する一つの正当性を示すものである。一方で，LS アルゴリズムはどの場面でも効率的であるとは限らず，より安定した効率性を備えた反復アルゴリズムを検討する余地が残されていることが分かった。

5 改良 LS アルゴリズムの可能性について

以上の考察をヒントにし，ここでは LS アルゴリズムの改良の可能性について議論する。

前章で示したとおり，各ステップで目的関数の降下量を最大にする $\beta_m^{(k)}$ は解析的に求めることはできないが，最適な $\beta_m^{(k)}$ を数値的に探索することはできる。すなわち，一般化 LS アルゴリズムにおいて，式 (22) の後段で行われる式 (23) の更新を $\mathbf{u}$, β に関する最適化問題と捉えるわけである。ここで，$\mathcal{G}^+(\mathbf{u},\alpha)$ を $\mathbf{u}$ に関して最小化するステップと β に関して最小化するステップを交互に繰り返すアプローチをとることにすると，反復アルゴリズムは表 1 のとおりとなる。なお，式 (31) は $\mathbf{u}$ を固定した状態で式 (15) の制約下において $\mathcal{G}^+(\mathbf{u},\alpha)$ を最小にする β の解である。以下に示したアルゴリズムはあくまで改良案であり，真に改良アルゴリズムとなっているかどうかは数値実験で評価する必要がある。これは今後の検討課題である。

6 複素 NMF アルゴリズム [2] の効率化

本稿での議論は，Frobenius ノルム規準の NMF に限らず，我々が以前提案した複素 NMF[2] の問題設定にも当てはまる。以下，複素 NMF の概要を示す。

複素 NMF とは，観測信号の時間周波数成分 $Y := (Y_{\omega,t})_{\Omega\times T} \in \mathbb{C}^{\Omega\times T}$ に対し，以下に示すモデル

$$F_{\omega,t} = \sum_k H_\omega^{(k)} U_t^{(k)} e^{\mathrm{j}\phi_{\omega,t}^{(k)}} \tag{32}$$

表 1　改良 LS アルゴリズム (案)

1. $\mathbf{H}$, $\mathbf{U}$ を初期設定する。

2. β を以下のとおりに初期設定する。

$$\beta_{m,n}^{(k)} = \frac{h_{m,k}u_{k,n}}{\left[\mathbf{HU}\right]_{m,n}}$$

3. α を以下の式に従って更新する。

$$\alpha_{m,n}^{(k)} \leftarrow \frac{1}{y_{m,n}}\Big[h_{m,k}u_{k,n} + \beta_{m,n}^{(k)}\Big(y_{m,n} - \left[\mathbf{HU}\right]_{m,n}\Big)\Big]$$

4. 以下の 2 ステップを収束するまで繰り返す。

   (a) $\mathbf{H}$, $\mathbf{U}$ の行列要素を以下の式に従って更新する。

$$h_{m,k} \leftarrow \frac{\sum_n \alpha_{m,n}^{(k)} y_{m,n} u_{k,n}/\beta_{m,n}^{(k)}}{\sum_n {u_{k,n}}^2/\beta_{m,n}^{(k)}}$$

$$u_{k,n} \leftarrow \frac{\sum_n \alpha_{m,n}^{(k)} y_{m,n} h_{m,k}/\beta_{m,n}^{(k)}}{\sum_n {h_{m,k}}^2/\beta_{m,n}^{(k)}}$$

   (b) β を以下の式に従って更新する。

$$\beta_{m,n}^{(k)} \leftarrow \frac{\left|\alpha_{m,n}^{(k)} y_{m,n} - h_{m,k}u_{k,n}\right|}{\sum_k \left|\alpha_{m,n}^{(k)} y_{m,n} - h_{m,k}u_{k,n}\right|} \tag{31}$$

   収束後，2. に戻る。

を，U ができるだけスパースになるようにフィッティングするものである。各パラメータの説明については [2] を参照されたい。複素 NMF では，具体的には，

$$f(H,U,\phi) := \sum_{\omega,t} \left|Y_{\omega,t} - F_{\omega,t}\right|^2 + 2\lambda \sum_{k,t} |U_t^{(k)}|^p \tag{33}$$

を目的関数として

$$\begin{array}{ll} \text{minimize} & f(H,U,\phi) \\ \text{subject to} & \sum_x H_\omega^{(k)} = 1 \quad (k = 1, \cdots, K) \end{array} \tag{34}$$

のような最適化問題を解くことになる。[2] ではこの最適化問題の解法として，補助関数

$$\begin{aligned} & f^+(H,U,\phi,\alpha,V) \\ & := \sum_{\omega,t}\sum_k \frac{1}{\beta_{\omega,t}^{(k)}} \left|\alpha_{\omega,t}^{(k)} Y_{\omega,t} - H_\omega^{(k)} U_t^{(k)} e^{\mathrm{j}\phi_{\omega,t}^{(k)}}\right|^2 \\ & \quad + \lambda \sum_{k,t} \left\{p|V_t^{(k)}|^{p-2} {U_t^{(k)}}^2 + (2-p)|V_t^{(k)}|^p\right\} \end{aligned}$$

をもとにした反復アルゴリズムを提案したが，β をどのように決定すべきかが重要課題として残されていた。ここでは本稿の考察に従ってこの課題を以下で議論する。

複素 NMF における β の決定方法において，5 章までの議論との唯一の違いは複素数を扱っている点である。この点に注意して，H, U, ϕ の更新による目的関数の降下量を最大にする β の最適化問題を 3 章と同様に定式化する。例えば U の更新においては，β の目的関数は

$$\begin{aligned} \mathcal{F}(\beta) := & \sum_{k,t} \Bigg\{ \sum_\omega \frac{\left|\alpha_m^{(k)}\right|^2 \left|Y_{\omega,t}\right|^2}{\beta_{\omega,t}^{(k)}} \\ & - \left(\sum_\omega \frac{H_\omega^{(k)}}{\beta_{\omega,t}^{(k)}} \mathrm{Re}\Big[\alpha_{\omega,t}^{(k)*} Y_{\omega,t}^* e^{\mathrm{j}\phi_{\omega,t}^{(k)}}\Big]\right)^2 \Bigg/ \left(\sum_\omega \frac{{H_\omega^{(k)}}^2}{\beta_{\omega,t}^{(k)}}\right) \Bigg\} \end{aligned}$$

となる。$\mathcal{F}(\beta)$ を最小化する β は解析的に求めることはできないが，4 章と同様 $\mathcal{F}(\beta)$ の第一項を最小化するには

$$\beta_{\omega,t}^{(k)} = \frac{\left|\alpha_{\omega,t}^{(k)}\right|}{\sum_k \left|\alpha_{\omega,t}^{(k)}\right|} \tag{35}$$

のように更新すれば良いことが示される。この β の更新則は，[2] で採用した更新則と異なるため，今後それぞれの更新則を採用した場合のアルゴリズムの振る舞いを比較検討したい。

7　まとめ

本報告では，Frobenius ノルム規準の非負値行列因子分解 (NMF) のための反復アルゴリズム (LS アルゴリズム)[1] を包含するアルゴリズムの一般形 (一般化 LS アルゴリズム) を示し，それにより導かれる解釈をもとに，より効率的なアルゴリズム (改良 LS アルゴリズム) を実現できる可能性について考察した。また，その考察をもとに，一般化 LS アルゴリズムと類似のアプローチによりパラメータの最適化を行う複素 NMF [2] における計算の効率化に向けて，その展望について議論した。

参考文献

[1] D.D. Lee and H.S. Seung, “Algorithms for non-negative matrix factorization,” in Advances in Neural Information Processing Systems 13 (Proc. NIPS*2000), pp. 556–562, 2000.

[2] H. Kameoka, N. Ono, K. Kashino, S. Sagayama, ”Complex NMF: A New Sparse Representation for Acoustic Signals,” In Proc. 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2009), pp. 3437-3440, 2009.