

# グラフを用いたNMFの地域分散高速化

越塚 毅<sup>1</sup> 竹内 孝<sup>2</sup> 松林 達史<sup>2</sup> 澤田 宏<sup>2</sup>

**概要** : 集計データに対する教師なしのパターン認識技術として、非負値行列因子分解 (Non-negative Matrix Factorization: NMF) は広く使われている。特に Non-negative Multiple Matrix Factorization (NMMF) では、複数のデータから共通する項目を共通因子として扱い、効果的に同時分解を行う。本研究では共通因子に加え、地域性などの物理的に関係性をもつ集計データに焦点を当て、因子分解を行う rNMF (Regional Non-negative Matrix Factorization) を提案する。rNMF は、物理的に距離の近い地域のデータは同様の特徴空間で表現し、分析結果をより直感的にわかりやすいものとする。さらに分析対象のデータが大規模な行列であっても、分散システム上で高速に処理を行える。rNMF は、地域ごとの観測行列  $\mathbf{Y}_i$  を入力とし、それぞれ観測行列を二つの因子行列  $\mathbf{H}_i, \mathbf{U}_i$  へと分解する。ただし、 $i$  は地域の識別番号を表す。そして、再構成誤差の最小化と同時に、物理的に距離の近い地域の行列からは同様の基底行列が得られるよう、対象の基底行列間の距離を最小化する。図2では、再構成誤差の最小化を  $\mathbf{Y}_i \simeq \mathbf{H}_i \mathbf{U}_i$  と表し、基底行列間の距離の最小化を  $\mathbf{H}_i \simeq \mathbf{H}_j$  と表している。また、地域の位置関係はグラフによって与える。さらにグラフの彩色問題をヒューリスティック的に解くことで、分散システム上で並列に処理することが可能である。本稿では、まず rNMF を NMF の拡張アルゴリズムとして定式化を行う。そして、再構成誤差と基底行列間誤差に Frobenius ノルムと一般化 KL ダイバージェンスを用いた場合のパラメータ更新法を示す。地域ごとに集計された実データを用いた実験では、rNMF を用いることで、隣接した地域データを共通した特徴空間で表現できること、従来の NMF に対して汎化性能が悪化しないこと、分散システム上で高速に動作することを示す。

キーワード : 行列分解, 並列化, 分散処理

## Graph-based regional NMF for distributed computing

Koshizuka Takeshi<sup>1</sup> Koh Takeuchi<sup>2</sup> Tatsushi Matsubayashi<sup>2</sup> Hiroshi Sawada<sup>2</sup>

### 1. はじめに

NMF (Non-negative Matrix Factorization: 非負値行列因子分解) [1] [2] は、非負制約下で観測データを指定した基底数に分解し、低ランク近似表現を得る行列分解手法である。文章データ [5]、音源データ [6]、顧客データなどの非負値の行列で表現されるデータに対して適用し、データの圧縮や潜在的特徴の可視化を目的とする。低ランク表現を得る手法として、他に主成分分析や特異値分解などがあるが、NMF は非負値行列のみを扱い、分解後の行列も非負値行列に制限される。これによって、より直感的な潜在的特徴を抽出することが可能とされている [7]。近年は扱えるデータが増加しており、NMF を用いた分析が可能な場面が増えてい

る。中でも、地域ごとに集計されたデータに対して NMF を適用する場面は多く考えられる。例えば地域ごとに集計された顧客データに対して NMF を適用し、潜在的特徴を抽出することで、地域ごとの販売戦略に生かす場面などが考えられる。このとき、地域ごとのデータから得られる潜在的特徴を、より直感的にわかりやすいものとするため、物理的に距離の近い地域のデータは同様の特徴空間で表現したいという要求がある。しかしデータが大規模である場合には、各地域のデータを結合して処理を行うことは現実的でなく、地域ごとのデータを分散的に処理する必要がある。本稿では、以上の要求を解決する rNMF (Region Non-negative Matrix Factorization) を提案する。rNMF は、地域ごとの観測行列  $\mathbf{Y}^* = \{\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_I\}$  を入力とし、それぞれ観測行列を二つの因子行列  $\{\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_I\}, \{\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_I\}$  へと分解する。この時再構成誤差の最小化と同時に、物理

<sup>1</sup> 東京理科大学理工学部情報科学科

<sup>2</sup> 日本電信電話 (株) NTT コミュニケーション科学基礎研究所

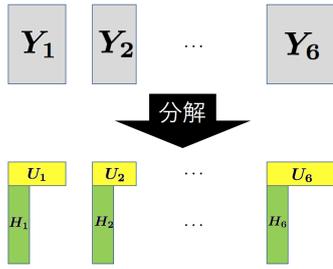


図 1: 提案手法の直感的理解

Fig. 1 an intuitive understanding of proposed method

的に距離の近い地域の行列からは同様の基底行列が得られるよう、対象の基底行列間の距離を最小化する。また、地域の位置関係はグラフによって与える。ただし、ここでのグラフは平面グラフに近いグラフを想定しており、完全グラフのような頂点に対して辺数が極端に多いものは想定していない。グラフにおいて、頂点  $i, j$  間に辺が存在しない時は、行列  $H_i, H_j$  の更新を非同期的に並列に行うことができる。そして、非同期的に並列実行可能な頂点集合を求める問題は、グラフの彩色問題として解くことができ、本稿では Welsh-Powell アルゴリズムを用いて解くことで、効率的な並列化を行う。なお、図 1 は  $I$  が 6 の場合における提案手法の概要を図式化したものである。以上より rNMF は、地域ごとの観測行列からより直感的にわかりやすい因子行列を得ることや、計算の並列分散化による高速化が期待できる。

## 2. 関連研究

NMF(Non-negative Matrix Factorization: 非負値行列因子分解) [1] [2] は、行列が持つ潜在的要素を分析するために用いる低ランク近似手法の一つであり、非負値行列を 2 つの非負値行列に分解する行列分解手法である。NMF は因子行列の非負値制約により、解釈可能なパターンが抽出されやすい特徴がある。NMF の拡張として、スパースな行列に対して有効な手法 [10] や、非負テンソルを解析する手法 [2]、MapReduce の枠組みを用いて処理の分散高速化を行う手法 [8] [9] などが提案されている。中でも本研究と関連の深い NMF の拡張として、グラフ構造を用いた正則化項を加える手法 (GNMF) [4] や辞書学習を用いて行列分解を最適化する手法 (Sparse Coding and Sparse Dictionary Learning) [11]、複数の行列を同時に因子分解する手法 [13] [14] がある。以下では、従来の NMF と GNMF の定式化を述べる。

### 2.1 NMF

非負値の分解前の行列  $\mathbf{Y} = [\mathbf{y}_1 \ \mathbf{y}_2 \ \cdots \ \mathbf{y}_M] \in \mathbb{R}_{\geq 0}^{N \times M}$  が与えられる。基底数を  $K$  と定めると、分解後の行列は基底行列  $\mathbf{H} \in \mathbb{R}_{\geq 0}^{N \times K}$  と重み行列  $\mathbf{U} \in \mathbb{R}_{\geq 0}^{K \times M}$  となる。

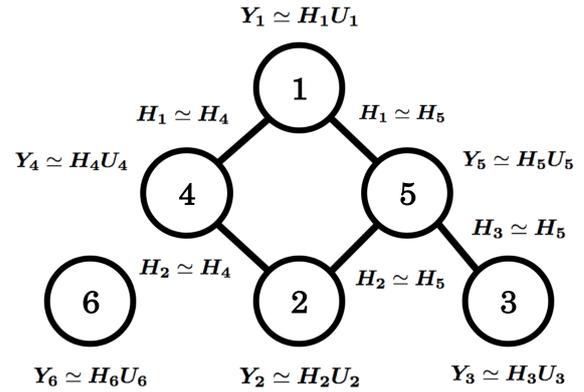


図 2: グラフの例

Fig. 2 an example of graph

$\mathbf{H}, \mathbf{U}$  の積  $\hat{\mathbf{Y}}$  が元の行列  $\mathbf{Y}$  の近似となり、 $\hat{\mathbf{Y}}$  と  $\mathbf{Y}$  の距離関数を最小化問題を解く。一般的に Frobenius ノルムや一般化 KL ダイバージェンス、板倉齊藤距離が距離規範としてよく用いられる [2]。この距離関数最小化問題にはいくつかの数学的解法が提案されている。乗法的更新ルール [2] や ALS (Alternating Least Squares) [3] などが提案されている。ここでは、Frobenius ノルムの最小化問題を乗法的更新ルールを用いて解く例を示す。乗法的更新ルールでは、まず行列  $\mathbf{H}, \mathbf{U}$  を非負値で初期化する。その後式 (3) を繰り返し用いて、行列  $\mathbf{H}, \mathbf{U}$  の値を更新する。そして更新を規定回数行う、または更新前後の変化がある一定値以下となることで終了する。

$$\text{minimize} \quad \|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2 \quad (\hat{\mathbf{Y}} = \mathbf{H}\mathbf{U}) \quad (1)$$

$$\text{subject to} \quad \mathbf{H} \geq 0, \mathbf{U} \geq 0 \quad (2)$$

$$H_{n,k} \leftarrow H_{n,k} \frac{[\mathbf{Y}\mathbf{U}^T]_{n,k}}{[\hat{\mathbf{Y}}\mathbf{U}^T]_{n,k}}, U_{k,m} \leftarrow U_{k,m} \frac{[\mathbf{H}^T\mathbf{Y}]_{k,m}}{[\mathbf{H}^T\hat{\mathbf{Y}}]_{k,m}} \quad (3)$$

### 2.2 GNMF

$\mathbf{Y} = [\mathbf{y}_1 \ \mathbf{y}_2 \ \cdots \ \mathbf{y}_M] \in \mathbb{R}_{\geq 0}^{N \times M}$  が与えられる。基底数を  $K$  と定め、因子行列  $\mathbf{H} \in \mathbb{R}_{\geq 0}^{N \times K}$  と  $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_M] \in \mathbb{R}_{\geq 0}^{K \times M}$  へと分解する。次に、 $\mathbf{y}_j$  と  $\mathbf{u}_l$  の近さを表す隣接行列  $\mathbf{W} = (w_{j,l})$  を定義する。この時、目的関数を以下のように定める。

$$\text{minimize} \quad \sum_i^I D_R(\mathbf{Y}_i | \mathbf{H}_i \mathbf{U}_i) + \frac{\lambda}{2} \sum_{j,l=1}^N D_G(\mathbf{u}_i | \mathbf{u}_j) w_{jl} \quad (4)$$

$$\text{subject to} \quad \mathbf{H}, \mathbf{U} \geq 0$$

ただし、 $D_R, D_G$  は距離関数、 $\lambda$  はハイパーパラメータである。距離関数を  $D_R, D_G$  とともに Frobenius ノルムとした場合、乗法的更新ルールを用いると以下の更新式が導かれる。

$$H_{n,k} \leftarrow H_{n,k} \frac{[\mathbf{Y}\mathbf{U}^T]_{n,k}}{[\hat{\mathbf{Y}}\mathbf{U}^T]_{n,k}} \quad (5)$$

$$U_{k,m} \leftarrow U_{k,m} \frac{[\mathbf{Y}^T\mathbf{H}]_{k,m} + \lambda[\mathbf{U}^T\mathbf{W}]_{k,m}}{[\mathbf{H}^T\hat{\mathbf{Y}}^T]_{k,m} + \lambda[\mathbf{U}^T\mathbf{D}]_{k,m}} \quad (6)$$

$$\text{ただし, } \mathbf{D} = (d_{j,j}) = \sum_l w_{jl} \quad (7)$$

GNMF では、観測行列の特定の列ベクトル間に潜在的な共通の構造を想定し、対応する基底行列の列ベクトル間の距離を近づけている。一方提案手法は、観測行列の区分行列単位での潜在的な共通構造を想定している点で異なる。さらに、提案手法では分散処理による高速化の手法について提案している点においても異なる。

### 3. 提案手法

本稿では、複数の非負値行列が与えられた時に、同様な基底行列を用いた低ランク表現を得る手法について述べる。図1のように地域ごとの行列  $\{\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_I\}$  が与えられる。各区分行列をそれぞれ基底行列  $\{\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_I\}$  と重み行列  $\{\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_I\}$  に分解し、この時同時に特定の基底行列  $\mathbf{H}_i, \mathbf{H}_j$  の距離を近づける。グラフによって近づける基底行列の情報を与えることで、柔軟に基底行列を扱うことが可能となっている。

#### 3.1 定式化

##### 3.1.1 入力

複数の非負値行列  $\mathbf{Y}^* = \{\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_I\}$  ( $\mathbf{Y}_i \geq 0$ ) が与えられる。

##### 3.1.2 出力

複数の非負値行列  $\mathbf{Y}^*$  から、基底行列  $\mathbf{H}^* = \{\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_I\}$  と重み行列  $\mathbf{U}^* = \{\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_I\}$  ( $\mathbf{H}_i, \mathbf{U}_i \geq 0$ ) に分解される。

##### 3.1.3 条件

出力で得る  $\mathbf{H}^*$  のうち、 $\mathbf{H}_i, \mathbf{H}_j$  を同様の行列としたい場合に、辺  $(i, j) \in E$  となる無向グラフ  $G(V = \{1, 2, \dots, I\}, E)$  を与える。このグラフの隣接行列を  $\mathbf{A}(G) = (a_{i,j})$  とする。例として、 $I = 6$  の場合のグラフの例を図2に示す。

##### 3.1.4 目的関数

$$\text{minimize } \sum_{i=1}^I [D_R(\mathbf{Y}_i|\mathbf{H}_i\mathbf{U}_i) + \frac{\alpha}{2|E_i|} \sum_{(i,j) \in E_i} D_G(\mathbf{H}_i|\mathbf{H}_j)]$$

$$\text{ただし, } E_i \text{ は頂点 } i \text{ に接続する辺の集合} \quad (8)$$

$$\text{subject to } \mathbf{H}_i, \mathbf{U}_i \geq 0 \quad (\forall i \in \{1, 2, \dots, I\}),$$

ただし、 $D_R, D_G$  は距離関数、 $\alpha$  はハイパーパラメータである。この目的関数について述べる。左項は従来のNMFでも扱う再構成誤差であり、右項が提案手法で導入した項である。この項によって、最終的に得られる基底行

列  $\mathbf{H}_i, \mathbf{H}_j$  が同様のものとなる狙いがある。図2のグラフの例では、シグマ記号の添え字  $i$  の値が1のとき、 $D_R(\mathbf{Y}_1|\mathbf{H}_1\mathbf{U}_1) + \frac{\alpha}{2} \{D_G(\mathbf{H}_1|\mathbf{H}_2) + D_G(\mathbf{H}_1|\mathbf{H}_3)\}$  を計算する。この項が存在することで、 $\mathbf{H}_1$  と  $\mathbf{H}_2, \mathbf{H}_1$  と  $\mathbf{H}_3$  の距離が小さい結果が得られると期待できる。

##### 3.1.5 モデル推定

距離関数を  $D_R, D_G$  ともに Frobenius ノルムとした場合、以下の式9を用いて、目的関数を最小化する最適な  $\mathbf{H}^*, \mathbf{U}^*$  が求められる。

$$H_{i,n,k} \leftarrow H_{i,n,k} \frac{[\mathbf{Y}_i\mathbf{U}_i^T]_{n,k} + \frac{\alpha}{|E_i|} \sum_{(i,j) \in E_i} H_{j,n,k}}{[\hat{\mathbf{Y}}_i\mathbf{U}_i^T]_{n,k} + \frac{\alpha}{|E_i|} \sum_{(i,j) \in E_i} H_{i,n,k}} \quad (9)$$

$$U_{i,k,m} \leftarrow U_{i,k,m} \frac{[\mathbf{H}_i^T\mathbf{Y}_i]_{k,m}}{[\mathbf{H}_i^T\hat{\mathbf{Y}}_i]_{k,m}} \quad (\text{ただし, } \hat{\mathbf{Y}}_i = \mathbf{H}_i\mathbf{U}_i)$$

距離関数を  $D_G$  を一般化KLダイバージェンスとすると、式10の目的関数を乗法的更新ルールを用いて最小化することはできない。これは、 $\mathbf{H}_i$  の更新式が非負性を保った更新式でなくなるためである。そこで、 $D_G$  に一般化KLダイバージェンスを用いる時は、目的関数を以下のように定義する。

$$\text{minimize } \sum_{i=1}^I [D_R(\mathbf{Y}_i|\mathbf{H}_i\mathbf{U}_i) + \frac{\alpha}{|E_i|} \sum_{(i,j) \in E_i} D_{KL}(\mathbf{H}_i + \mathbf{1}|\mathbf{H}_j + \mathbf{1})] \quad (10)$$

$$\text{ただし, } E_i = \{j \mid (i,j) \in E_i\}, \mathbf{1}: \text{全ての要素が1の行列}$$

$$\text{subject to } \mathbf{H}_i, \mathbf{U}_i \geq 0 \quad (\forall i \in \{1, 2, \dots, I\}),$$

距離関数を  $D_R, D_G$  をともに一般化KLダイバージェンスとすると、式11を用いて、目的関数を最小化する最適な  $\mathbf{H}^*, \mathbf{U}^*$  が求められる。

$$A = [(\mathbf{Y}_i \circ \hat{\mathbf{Y}}_i)\mathbf{U}_i^T]_{n,k} + \frac{\alpha}{|E_i|} \sum_{(i,j) \in E_i} \log(H_{j,n,k} + 1) + \frac{(H_{j,n,k} + 1)}{(H_{i,n,k} + 1)}$$

$$B = \sum_{m=1}^M [\mathbf{U}_i]_{k,m} + \frac{\alpha}{|E_i|} \sum_{(i,j) \in E_i} (\log(H_{i,n,k} + 1) + 1)$$

$$H_{i,n,k} \leftarrow H_{i,n,k} \frac{A}{B} \quad (11)$$

$$U_{i,k,m} \leftarrow \frac{[\mathbf{H}_i^T(\mathbf{Y}_i \circ \hat{\mathbf{Y}}_i)]_{k,m}}{\sum_{n=1}^N [\mathbf{H}_i]_{n,k}}$$

$$\text{ただし, } \circ \text{ は要素ごとの除算を表す。}$$

さらに、 $D_R, D_G$  に異なる距離関数を用いることもでき、 $H_{i,n,k}$  の更新式は以下のように整理できる。

$$H_{i,n,k} \leftarrow H_{i,n,k} \frac{NL + \frac{\alpha}{|E_i|} \sum_{(i,j) \in E_i} NR}{DL + \frac{\alpha}{|E_i|} \sum_{(i,j) \in E_i} DR} \quad (12)$$

$D_R$ : Frobenius ノルム

$$NL = [\mathbf{Y}_i \mathbf{U}_i^T]_{n,k}, DL = [\hat{\mathbf{Y}}_i \mathbf{U}_i^T]_{n,k}$$

$D_R$ : 一般化 KL ダイバージェンス

$$NL = [(\mathbf{Y}_i \circ \hat{\mathbf{Y}}_i) \mathbf{U}_i^T]_{n,k}, DL = \sum_{m=1}^M [\mathbf{U}_i]_{k,m}$$

$D_G$ : Frobenius ノルム

$$NR = H_{j,n,k}, DR = H_{i,n,k}$$

$D_G$ : 一般化 KL ダイバージェンス

$$NR = \log(H_{j,n,k} + 1) + \frac{(H_{j,n,k} + 1)}{(H_{i,n,k} + 1)}$$

$$DR = \log(H_{i,n,k} + 1) + 1$$

なお、式 9、式 11 の導出方法の詳細は付録に示す。

### 3.2 欠損値のある行列に対する NMF

$T$  は添え字全体の集合、 $J$  は欠損値の添え字の集合とし、行列を定義する。

$$\Omega = (\omega_{i,j}) = \begin{cases} 1 & ((i,j) \in T \setminus J) \\ 0 & ((i,j) \in J) \end{cases} \quad (13)$$

欠損値のある行列に対する従来の NMF の更新式を示す。

$$H_{i,n,k} \leftarrow H_{i,n,k} \frac{[(\Omega \circ \mathbf{Y}_i) \mathbf{U}_i^T]_{n,k}}{[(\Omega \circ \hat{\mathbf{Y}}_i) \mathbf{U}_i^T]_{n,k}}$$

$$U_{i,k,m} \leftarrow U_{i,k,m} \frac{[\mathbf{H}_i^T (\Omega \circ \mathbf{Y}_i)]_{k,m}}{[\mathbf{H}_i^T (\Omega \circ \hat{\mathbf{Y}}_i)]_{k,m}}$$

ただし、 $\circ$  はアダマール積を表す。

### 3.3 効率的な並列化

与えられるグラフ  $G$  において、頂点  $i, j$  間に辺が存在しない時は、行列  $\mathbf{H}_i, \mathbf{H}_j$  の更新を非同期的に並列に行うことができる。そこで、全体のグラフの頂点集合  $V = \{1, 2, \dots, I\}$  をグループ分けし、部分頂点集合の分割  $\{V_1, V_2, \dots, V_G\}$  を求める。ただし、各部分頂点集合  $V_g$  は、 $V_g$  に含まれる任意の 2 つの頂点間に辺が存在しない頂点集合とする。この時、各部分頂点集合  $V_g$  に含まれる頂点に対応する地域の行列は、並列に更新を行うことができる。さらに分割する集合の数  $G$  を最小化することで並列数を高めることができる。 $G$  を最小化する問題は、グラフの頂点彩色問題であり、最小の彩色数を求める問題は NP 困難クラスの問題である。そこでヒューリスティックで近似的に彩色数を求める。一つのヒューリスティックとして、Welsh-Powell アルゴリズム [12] を用いる。このアルゴリズムを用いて頂点を彩色した例を図 3 に示す。

### 3.4 アルゴリズム

距離関数を  $D_R, D_G$  とともに Frobenius ノルムとした場合のアルゴリズムを Algorithm1 に示す。

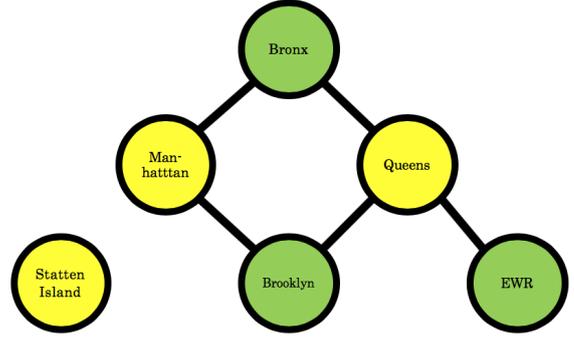


図 3: 頂点彩色の例

Fig. 3 an example of vertex coloring

#### Algorithm 1 rNMF

**Require:**  $\mathbf{Y}^* = \{\mathbf{Y}_1 \geq 0, \mathbf{Y}_2 \geq 0, \dots, \mathbf{Y}_I \geq 0\}, K, \alpha, Iteration$   
 一様分布からのサンプリングにより初期化:  $\mathbf{H}_i, \mathbf{U}_i$  ( $1 \leq i \leq I$ )  
 頂点集合を Welsh-Powell によって  $\{V_1, V_2, \dots, V_G\}$  へ分解:  
 $\{V_1, V_2, \dots, V_G\} \leftarrow \text{Welsh-Powell}(\{1, 2, \dots, I\}, \text{graph})$   
**for**  $iter = 0$  to  $Iteration - 1$  **do**  
   **for**  $g = 1$  to  $G$  **do**  
   **for each**  $i \in V_g$  **do in parallel**  
      $\mathbf{H}_i \leftarrow H_{i,n,k} \frac{[\mathbf{Y}_i \mathbf{U}_i^T]_{n,k} + \frac{\alpha}{|E_i|} \sum_{(i,j) \in E_i} H_{j,n,k}}{[\hat{\mathbf{Y}}_i \mathbf{U}_i^T]_{n,k} + \frac{\alpha}{|E_i|} \sum_{(i,j) \in E_i} H_{i,n,k}}$   
      $\mathbf{U}_i \leftarrow U_{i,k,m} \frac{[\mathbf{H}_i^T \mathbf{Y}_i]_{k,m}}{[\mathbf{H}_i^T \hat{\mathbf{Y}}_i]_{k,m}}$   
   **end for**  
**end for**  
**end for**

#### 3.4.1 時間計算量

地域ごとの行列  $\mathbf{Y}^* = \{\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_I\}$  を入力とし、各行列のサイズを  $(N \times M_i)$  とする。 $\mathbf{Y}^*$  を以下のように結合し、1 つの行列として従来手法を適用し、基底  $K$  で分解を行うとき、イテレーション回数を  $T$  とすれば、時間計算量のオーダーは  $O(\sum_{i=1}^I M_i N K T)$  である。

$$\begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ \mathbf{Y}_1 & \mathbf{Y}_2 & \dots & \mathbf{Y}_I \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

$\mathbf{Y}^*$  に対し提案手法を適用し、他は同様の条件で分解を行う。Welsh-Powell アルゴリズムによって、全体のグラフの頂点集合  $V = \{1, 2, \dots, I\}$  から集合族  $\{V_1, V_2, \dots, V_G\}$  を求める。ただし、各部分頂点集合  $V_g \subset V$  は  $V_g$  内の頂点間に辺が存在しない集合とする。この時、時間計算量のオーダーは  $O(\sum_{g=1}^G \max_{i \in V_g} M_i N K T)$  となる。よって、理論上提案手法は従来手法より高速に動作する。また、与えられるグラフが平面グラフかつ単純グラフである場合、 $G \leq 6$  となり、十分に高速に動作する。

## 4. 実験

### 4.1 実験概要

実験では、基底行列と重み行列の更新に式3を用いる従来手法と、式9を用いる提案手法の比較を行った。問題設定としては、まず地域ごとの行列  $Y_1, Y_2, \dots, Y_I$  が与えられる。特に断りのない限り、一台のCPUでマルチプロセス実行を行うプログラムとし、従来手法を元の行列  $Y$  に適用する場合は逐次実行を行うプログラムとした。

### 4.2 使用データ

全ての評価実験でNewYork市のタクシー乗車数のデータを用い、サイズの異なる2種類の行列を元に実験を行った。

- 行列データ1:  $Y_{small}^* = \{Y_1, Y_2, \dots, Y_6\}$

行列データ1は、NewYork市のタクシー乗車数をNewYorkの6地域ごとに集計したデータである。各行列  $Y_i$  ( $i = 1, 2, \dots, 6$ ) のサイズは  $(24 \times 365)$  であり、 $Y_i = (y_{n,m}^{(i)})$  は地域番号  $i$  における  $m$  日の  $n$  時から  $n+1$  時までのタクシー乗車数を表す。 $Y_{small}^*$  を列方向に結合した行列は  $Y_{small}$  ( $24 \times 2190$ ) であり、行列  $Y_{small} = (y_{n,m})$  は、地域番号  $\lceil \frac{m}{6} \rceil$  における  $m \pmod{6}$  日の  $n$  時から  $n+1$  時までのタクシー乗車数を表す。なお、地域番号と実際の地域との対応は以下のようになっている。

- 1 Bronx
- 2 Brooklyn
- 3 EWR
- 4 Manhattan
- 5 Queens
- 6 Staten Island

- 行列データ2:  $Y_{large}^* = \{Y_1, Y_2, \dots, Y_{234}\}$

行列データ2は、NewYork市のタクシー乗車数をNewYorkの234地域ごとに集計したデータである。行列データ1では、6地域ごとに集計されていたが、さらに細かく地域を分けて集計したものとなっている。各行列  $Y_i$  ( $i = 1, 2, \dots, 234$ ) のサイズは  $(144 \times 365)$  であり、 $Y_i = (y_{n,m}^{(i)})$  は地域番号  $i$  における  $m$  日の  $10n$  分から  $10(n+1)$  分までのタクシー乗車数を表す。 $Y_{large}^*$  を列方向に結合した行列は  $Y_{large}$  ( $144 \times 85410$ ) であり、行列  $Y_{large} = (y_{n,m})$  は、地域番号  $\lceil \frac{m}{234} \rceil$  における  $m \pmod{234}$  日の  $10n$  分から  $10(n+1)$  分までのタクシー乗車数を表す。

### 4.3 グラフの構築

グラフの頂点はNewYorkの各地域に対応するように構築した。つまり行列データ1を用いる時はグラフの頂点数は6とし、行列データ2を用いる時はグラフの頂点数は234

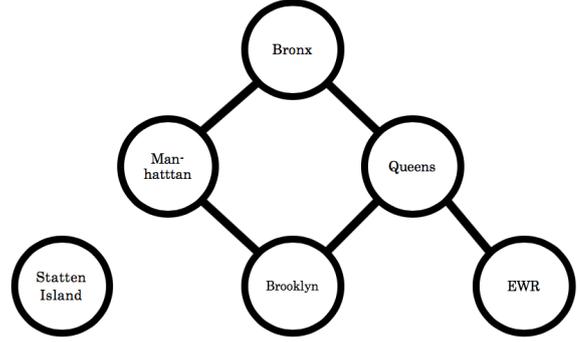


図4: 地域数6のデータにおけるグラフ  $G$

Fig. 4 a graph  $G$  of 6 region

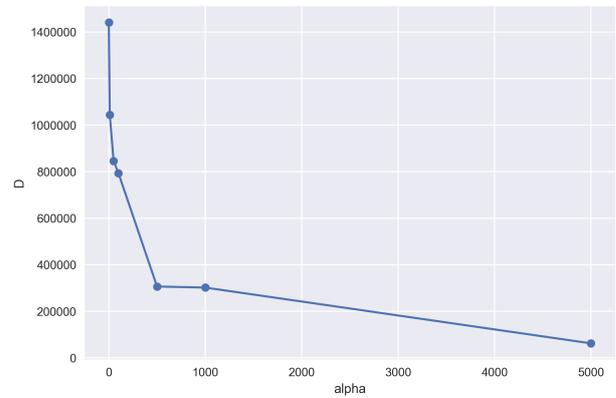


図5: 基底行列間の距離

Fig. 5 Distance of basic matrix

とした。グラフの辺は、実際の隣接している地域間に作り、重みは1とした。図4は、行列データ1を用いた実験で与えたグラフの図であり、NewYork市の6地域の隣接関係が表された図となっている。以下は、図4のグラフの隣接行列である。

$$A(G) = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (14)$$

### 4.4 基底行列の評価

#### 4.4.1 評価手法

行列データ1を用いて基底行列間の距離が近づいているかの評価実験を行った。まず基底行列をヒートマップを用いて可視化し、従来手法を  $Y_{small}^*$  に適用する場合、提案手法を  $Y_{small}^*$  に適用する場合で比較を行った。従来手法適用時の再構成誤差、 $D_R, D_G$  は全て Frobenius ノルムとし、

$\alpha = 500$  として実験を行った。

次に評価値を  $D = \sum_{(i,j) \in E_i} D_G(\mathbf{H}_i | \mathbf{H}_j)$  として求め、定量的な評価を行った。評価値  $D$  の値は、小さいほど同様の特徴空間で隣接地域のデータを表現できていると言える。提案手法において指定するハイパーパラメータ  $\alpha$  の値を変化させ、評価値  $D$  を求めた結果を表 5 に示した。なお、 $\alpha = 0$  の時が従来手法を  $\mathbf{Y}_{small}^*$  に適用した場合と等しい。提案手法適用時の  $D_R, D_G$  は全て Frobenius ノルムとした。実験の詳細条件を以下で述べる。特徴数  $K = 5$ 、イテレーション数  $Iteration = 500$  とし、 $\mathbf{H}_i, \mathbf{U}_i$  の初期化には、値域が  $[0.0, \sqrt{\frac{4.0 \sum_{n,m} Y_{i,n,m}}{624365}}]$  の一様乱数を用いた。この乱数を用いることで、初期値の期待値が  $E[[\hat{\mathbf{Y}}_i]_{n,m}] = \frac{\sum_{(n,m)} Y_{i,n,m}}{nm}$  を満たす。

#### 4.4.2 結果

まず、ヒートマップを用いた実験結果について述べる。図 6, 7 を比較すると、従来手法では全てが異なっているのに対し、提案手法では Bronx, Brooklyn, EWR, Manhattan, Queens の基底行列が同様な行列になっていることがわかる。すなわち提案手法では、グラフで連結な頂点に対応する基底行列の距離を小さくすることができている。さらにグラフで連結成分を持たなかった Staten Island の基底行列は、他の地域とは別の結果を得ることができている。このことからグラフを用いて、共通の基底を用いて表現する行列を選択できる点で、従来の NMF よりも柔軟性の高い分析が可能であると言える。次に  $D$  の値を用いた定量的な評価実験の結果について述べる。グラフ 5 から、 $\alpha$  の値が大きくなるほど  $D$  の値が小さくなっているとわかる。よって、 $\alpha$  の値を大きくするほど、近い特徴空間で隣接地域のデータを表現できると言える。

### 4.5 汎化性能の評価

#### 4.5.1 評価手法

行列データ 1 を用いて、従来手法を  $\mathbf{Y}_{small}$  に適用する場合、提案手法を  $\mathbf{Y}_{small}^*$  に適用する場合、 $\mathbf{Y}_{small}^*$  を 3 階テンソル  $\mathbf{Y}_{small}^{(t)}$  として扱い、非負値テンソル因子分解 (Non-negative Tensor Factorization: NTF) を適用する場合で比較を行った。なお 3 階テンソル  $\mathbf{Y}_{small}^{(t)}$  のサイズは、 $(6 \times 24 \times 365)$  である。評価手法としては、まず与えられた観測行列の約 5 パーセントの要素をマスクし、欠損のある行列とみなして式 13 を用いて行列分解を行う。すなわちマスクする添え字の集合を  $J$  とすれば、入力行列  $[\mathbf{Y}]_{i,j}$  ( $i, j \in J$ ) は欠損値として扱う。そして因子行列から再構成した行列  $\hat{\mathbf{Y}}$  を得て、汎化性能の評価値を  $L = \sum_{(n,m) \in J} \log L([\mathbf{Y}]_{(n,m)} | [\hat{\mathbf{Y}}]_{(n,m)})$  として算出する。ただし、 $L$  は元の観測行列  $\mathbf{Y}$  の従う分布の対数尤度関数である。NMF の統計的解釈によると、 $D_R$  を Frobenius ノルムとした時、 $Y_{i,n,m}$  は正規分布に従う仮定を置いている。実

験では、 $D_R, D_G$  をともに Frobenius ノルムとし、 $k$  の値を 3, 4, 5, 6 と変化させ、性能評価値を比較した。実験の詳細条件を以下で述べる。イテレーション数  $Iteration = 500$  とし、 $\mathbf{H}_i, \mathbf{U}_i$  の初期化には、値域が  $[0.0, \sqrt{\frac{4.0 \sum_{n,m} Y_{i,n,m}}{624365}}]$  の一様乱数を用いた。この乱数を用いることで、初期値の期待値が  $E[[\mathbf{H}_i \mathbf{U}_i]_{n,m}] = \frac{\sum_{(n,m)} Y_{i,n,m}}{nm}$  を満たす。提案手法の更新式では、 $\alpha = 500$  を用いた。

#### 4.5.2 結果

実験結果を図 8 に示した。rNMF と NMF を比較すると大きな差はなく、汎化性能において提案手法は従来手法の NMF に対して悪化していない。NTF と比較すると、rNMF と従来の NMF は汎化性能において劣っている。

### 4.6 計算時間の評価 (実験 1)

#### 4.6.1 評価手法

行列データ 1 を用いて、従来手法を  $\mathbf{Y}_{small}$  に適用する場合、提案手法を  $\mathbf{Y}_{small}^*$  に適用する場合で比較を行った。提案手法の実用上の工夫として、パラメータ  $\mathbf{H}_i, \mathbf{U}_i$  の更新を数回まとめて行うことで、高速化する方法が考えられる。本実験では、提案手法は逐次実行、マルチプロセス実行、10 回の更新をまとめて行うマルチプロセス実行の三通りの異なる実装を行い、実行時間を比較した。なお  $D_R, D_G$  はともに Frobenius ノルムとし、 $k$  の値は 5, 10, 50, 100 と変化させた。実験の詳細条件を以下で述べる。イテレーション数  $Iteration = 500$  とし、 $\mathbf{H}_i, \mathbf{U}_i$  の初期化には、値域が  $[0.0, \sqrt{\frac{4.0 \sum_{n,m} Y_{i,n,m}}{624365}}]$  の一様乱数を用いた。この乱数を用いることで、初期値の期待値が  $E[[\mathbf{H}_i \mathbf{U}_i]_{n,m}] = \frac{\sum_{(n,m)} Y_{i,n,m}}{nm}$  を満たす。提案手法の更新式では、 $\alpha = 500$  を用いた。

#### 4.6.2 結果

実験結果を図 9 に示した。マルチプロセス実行を行う提案手法と従来手法では、マルチプロセス実行を行う提案手法の方が計算時間が速く、優位性があると言える。逐次実行を行う提案手法と従来手法では計算時間に差はなく、並列化によって高速化が実現できていると言える。10 回の更新をまとめて行う提案手法と通常マルチプロセス実行を行う提案手法とでは、10 回の更新をまとめて行った方が実行時間が速い。ただし、まとめて更新を行う回数を増やすぎると本来の分析結果と大きく異なった結果が得られてしまう恐れがある。目的関数の値の変化が大きい序盤の更新は、通常マルチプロセス実行の提案手法で行い、終盤は数回の更新をまとめて行う提案手法を用いるという実用上の工夫が考えられる。

### 4.7 計算時間の評価 (実験 2)

#### 4.7.1 評価手法

サイズの大きい行列データ 2 を用いて評価実験を行った。従来手法を全体行列  $\mathbf{Y}_{large}$  に適用する場合、提案手

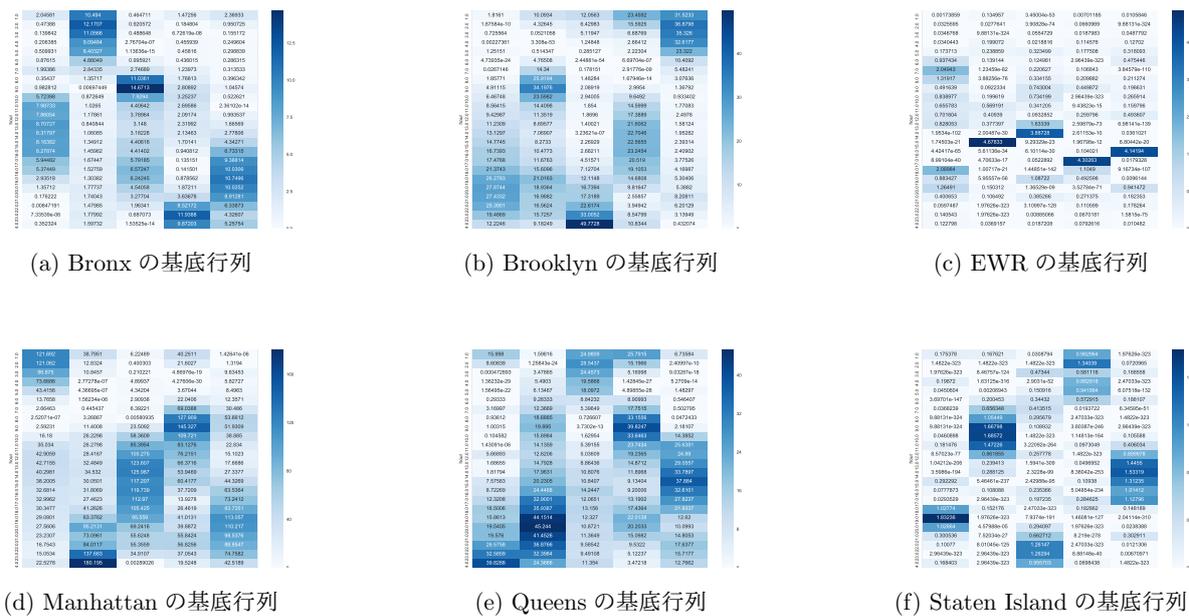


(a) Bronx の基底行列 (b) Brooklyn の基底行列 (c) EWR の基底行列



(d) Manhattan の基底行列 (e) Queens の基底行列 (f) Staten Island の基底行列

図 6: 提案手法を複数の区分行列  $Y_{small}^*$  に適用した場合の基底行列  
 Fig. 6 Base Matrix of regional data by proposed method



(a) Manhattan の基底行列 (b) Brooklyn の基底行列 (c) EWR の基底行列  
 (d) Manhattan の基底行列 (e) Queens の基底行列 (f) Staten Island の基底行列

図 7: 従来手法を複数の区分行列  $Y_{small}^*$  に適用した場合の基底行列  
 Fig. 7 Base Matrix of regional data by conventional method

法を  $Y_{large}^*$  に適用する場合で比較を行った。提案手法はマルチプロセス実行, 10 回の更新をまとめて行うマルチプロセス実行の二通りの異なる実装を行い, 実行時間を比較した。なお  $D_R, D_G$  はともに Frobenius ノルムとし,  $k$  の値は 5, 10, 50, 100 と変化させた。実験の詳細条件を以下で述べる。イテレーション数  $Iteration = 100$  とし,  $H_i, U_i$  の初期化には, 値域が  $[0.0, \sqrt{\frac{4.0 \sum_{n,m} Y_{i,n,m}}{624365}}]$  の一様乱数を用いた。この乱数を用いることで, 初期値の期待

値が  $E[[H_i U_i]_{n,m}] = \frac{\sum_{n,m} Y_{i,n,m}}{nm}$  を満たす。提案手法の更新式では,  $\alpha = 500$  を用いた。

### 4.7.2 結果

実験結果を図 10 に示した。計算時間の順位関係は, 実験 1 と同様な結果となった。行列サイズが大きい実験 2 の方がより計算時間の差が明らかとなっており, サイズの大きい行列に対する提案手法の優位性を強く示せた

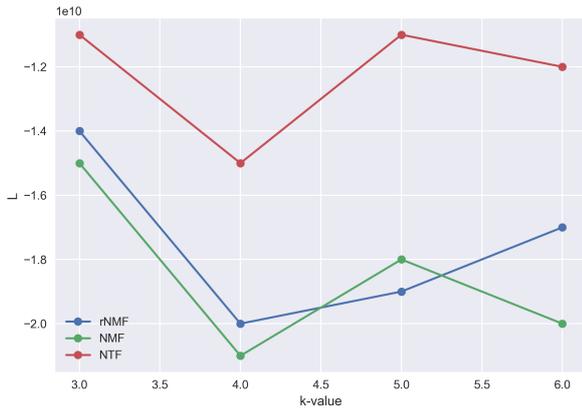


図 8: 汎化性能 (縦軸 L が大きいほど良い)

Fig. 8 generalization performance

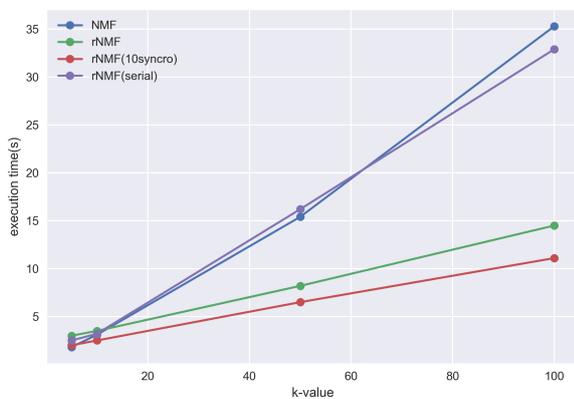


図 9: 計算時間の評価 (実験 1)

Fig. 9 evaluation of execution time (the experiment 1)

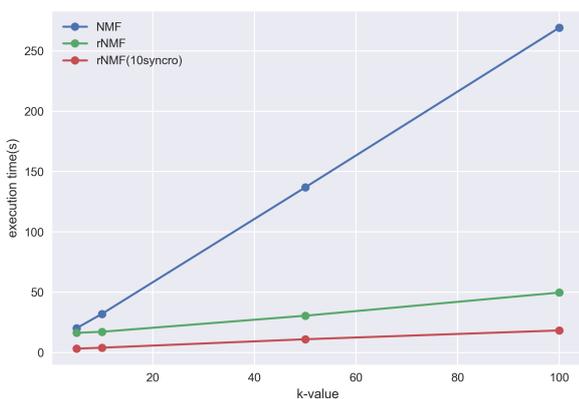


図 10: 計算時間の評価 (実験 2)

Fig. 10 evaluation of execution time (the experiment 2)

と言える。

## 5. 考察

本稿では、主に地域ごとのデータに対して適用する rNMF

を提案した。rNMF は、隣接した地域データを共通した特徴空間で表現し、なおかつ分散システム上で高速に動作することを目的とした手法である。本稿では、まず本手法で扱う問題の定式化を行ったのち、距離関数として Frobenius ノルムを用いた場合と、一般化 KL ダイバージェンスを用いた場合の問題の解法を示した。その後 NewYork 市のタクシー乗車数のデータを用いて実験を行い、指定した基底行列間の距離が小さくなっていること、従来手法と比較して汎化性能が劣らないことを示した。計算時間は、二つのサイズの行列データに対して適用し、従来手法と提案手法の比較を行なった。サイズの小さい行列に適用した場合もサイズの大きい行列データに適用した場合も提案手法が従来手法に対して優位性を示した。また、10 回の更新をまとめて行う提案手法と通常のマルチプロセス実行を行う提案手法とでは、10 回の更新をまとめて行った方が実行時間が速く、実用上の工夫として有効であることを示した。提案した手法が持つ課題や拡張の余地としては、良い分析結果を得るためのグラフ構築手法の定式化、非負値テンソル因子分解の高速化への応用などが考えられる。

謝辞

## 参考文献

- [1] D.D. Lee and H.S. Seung, "Learning the parts of objects with nonnegative matrix factorization," Nature, vol. 401, pp. 788-791, 1999.
- [2] A. Cichocki, R. Zdunek, A.H. Phan, and S. Amari. Non-negative Matrix and Tensor Factorizations : Applications to Exploratory Multi-way Data Analysis and Blind Source Separation, Wiley, 2009.
- [3] Cichocki, A., Zdunek, R. and Amari, S.: Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization, Proc. ICA (2007).
- [4] Cai, D., He, X. and Han, J.: Graph regularized non-negative matrix factorization for data representation, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.33, pp.1548-1560 (2011).
- [5] Xu, W., Liu, X. and Gong, Y.: Document clustering based on non-negative matrix factorization, Proc. SIGIR (2003).
- [6] P. Smaragdis and J.C. Brown, "Non-negative matrix factorization for polyphonic music transcription," In Proc. WASPAA 2003, pp. 177-180, Oct. 2003.
- [7] Eggert, J. and Korner, E.: Sparse coding and nmf, Proc. IJCNN (2004).
- [8] Liu, C., Yang, H., Fan, J., He, L. and Wang, Y.: Distributed nonnegative matrix factorization for web-scale dyadic data analysis on MapReduce, Proc. WWW (2010).
- [9] S. Liu, P. Flach, and N. Cristianini, "Generic Multiplicative Methods for Implementing Machine Learning Algorithms on MapReduce," CoRR, 2011.
- [10] P.O. Hoyer. Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research, 5:1457-1469, 2004.
- [11] V. Sindhwani and A. Ghoting, "Large-scale distributed nonnegative sparse coding and sparse dictionary learning," Proceedings of the 18th ACM SIGKDD interna-

表 A-1: 記法  
Table A-1 notation

記号	意味
$\mathbf{Y}$	入力行列
$\mathbf{H}$	$\mathbf{Y}$ の因子行列 (基底)
$\mathbf{U}$	$\mathbf{Y}$ の因子行列 (重み)
$\hat{\mathbf{Y}}$	因子行列の積 $\mathbf{H}\mathbf{U}$
$N$	入力行列の行数
$K$	因子行列の基底数.
$M$	入力行列の列数
$\mathbf{Y}^*$	入力行列の小行列の集合
$\mathbf{H}^*$	$\mathbf{Y}^*$ の小行列から得られる因子行列 (基底) の集合
$\mathbf{U}^*$	$\mathbf{Y}^*$ の小行列から得られる因子行列 (重み) の集合
$I$	入力行列を分割した小行列の数
$\mathbf{Y}_i$	入力行列を分割した $i$ 番目の小行列
$\mathbf{H}_i$	$\mathbf{Y}_i$ から得られる因子行列 (基底)
$\mathbf{U}_i$	$\mathbf{Y}_i$ から得られる因子行列 (重み)
$\hat{\mathbf{Y}}_i$	因子行列の積 $\mathbf{H}_i\mathbf{U}_i$
$D_R$	再構成誤差
$D_G$	基底行列間の距離関数
$\alpha$	ハイパーパラメータ
$E$	グラフの辺の集合
$E_i$	頂点 $i$ と連結な頂点集合
$\mathbf{Y}_{small}$	実験で用いたサイズが小さい入力行列
$\mathbf{Y}_{small}^*$	入力行列 $\mathbf{Y}_{small}$ の小行列の集合
$\mathbf{Y}_{large}$	実験で用いたサイズが大きい入力行列
$\mathbf{Y}_{large}^*$	入力行列 $\mathbf{Y}_{large}$ の小行列の集合. 分割単位が小さい.
$\mathbf{Y}_{large}^{**}$	入力行列 $\mathbf{Y}_{large}$ の小行列の集合. 分割単位が大きい.

tional conference on Knowledge discovery and data mining - KDD ' 12, p.489,2012.

- [12] Welsh, D. J. A.; Powell, M. B., "An upper bound for the chromatic number of a graph and its application to timetabling problems", The Computer Journal 10 (1): 85-86, 1967.
- [13] 竹内 孝, 石黒 勝彦, 木村 昭悟, 澤田 宏, 「非負値制約下における複合行列分解とそのソーシャルメディア解析への応用」 Multiple Matrix Factorization under the Non-negative Constraints and its Applications for Social Media Analysis, 2013.
- [14] 幸島 匡宏, 松林 達史, 澤田 宏, 「属性情報を考慮した消費者行動パターン抽出のための非負値多重行列因子分解法」 Non-negative Multiple Matrix Factorization for Consumer Behavior Pattern Extraction by Considering Attribution Information, 2015.

## 付 録

### A.1 記法

### A.2 更新式の導出

距離関数を  $D_R, D_G$  とともに Frobenius ノルムとした場合, この問題は乗法的更新ルールを用いて解くことが可能である. 更新式の導出を以下に示す.

$$f(\mathbf{H}^*, \mathbf{U}^*) = \sum_{i=1}^I \{ \|\mathbf{Y}_i - \mathbf{H}_i\mathbf{U}_i\|_F^2 + \frac{\alpha}{2|E_i|} \sum_{(i,j) \in E_i} \|\mathbf{H}_i - \mathbf{H}_j\|_F^2 \}$$

$$= \sum_{i=1}^I [ \sum_{(n,m)} |Y_{i,n,m}|^2 - 2Y_{i,n,m}[\mathbf{H}_i\mathbf{U}_i]_{n,m} + |[\mathbf{H}_i\mathbf{U}_i]_{n,m}|^2 + \frac{\alpha}{2|E_i|} \sum_{(i,j) \in E_i} \sum_{(n,k)} |H_{i,n,k} - H_{j,n,k}|^2 ]$$

Jensen の不等式より,

$$\begin{aligned} &\leq \sum_{i=1}^I [ \sum_{(n,m)} |Y_{i,n,m}|^2 - 2Y_{i,n,m}[\mathbf{H}_i\mathbf{U}_i]_{n,m} \\ &+ \sum_{t=1}^K \frac{H_{i,n,t}^2 U_{i,t,m}^2}{\lambda_{i,t,n,m}} + \frac{\alpha}{2|E_i|} \sum_{(i,j) \in E_i} \sum_{(n,k)} |H_{i,n,k} - H_{j,n,k}|^2 ] \\ &= G(\mathbf{H}^*, \mathbf{U}^*) \\ &\quad \left( \sum_{t=1}^K \lambda_{i,t,n,m} = 1, \lambda_{i,t,n,m} \geq 0 \right) \end{aligned}$$

$G(\mathbf{H}^*, \mathbf{U}^*)$  を最小化することで, 元の目的関数  $f(\mathbf{H}^*, \mathbf{U}^*)$  が最小化される.

$$\lambda_{i,t,n,m} \leftarrow \arg \min_{\lambda_{i,t,n,m}} \frac{H_{i,n,t} U_{i,t,m}}{\sum_{t'} H_{i,n,t'} U_{i,t',m}}$$

を用いると以下のように偏微分できる.

$$\begin{aligned} \frac{\partial G(\mathbf{H}^*, \mathbf{U}^*)}{\partial H_{i,n,k}} &= -2[\mathbf{Y}_i\mathbf{U}_i^T]_{n,k} + 2 \sum_{m=1}^M \frac{[\mathbf{H}_i]_{n,k} [\mathbf{U}_i]_{k,m}^2}{\lambda_{i,k,n,m}} \\ &+ 2 \frac{\alpha}{|E_i|} \sum_{(i,j) \in E_i} (H_{i,n,k} - H_{j,n,k}) = 0 \end{aligned}$$

$$\frac{\partial G(\mathbf{H}^*, \mathbf{U}^*)}{\partial U_{i,k,m}} = -2[\mathbf{H}_i^T\mathbf{Y}_i]_{k,m} + 2 \sum_{n=1}^N \frac{[\mathbf{H}_i]_{n,k}^2 [\mathbf{U}_i]_{k,m}}{\lambda_{i,k,n,m}} = 0$$

これを解けば, 式 9 の更新式が得られる.