情報処理学会論文誌 Vol.62 No.1 387–396 (Jan. 2021)

推薦論文

グラフを用いたNMFの地域分散高速化

越塚 毅[1,a)] 竹内 孝[2,†1] 松林 達史[3,†2] 澤田 宏[2]

受付日 2020年4月2日, 採録日 2020年10月6日

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，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な関係性を持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分かりやすいものとする．なお，地域の位置関係はグラフによって与える．さらに分析対象のデータが大規模な行列であっても，グラフの彩色問題をヒューリスティックに解くことで，分散システム上で高速に処理を行える．本稿では，まず rNMF を NMF の拡張として定式化を行い，パラメータ更新法も示す．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[1,a)] Koh Takeuchi[2,†1] Tatsushi Matsubayashi[3,†2] Hiroshi Sawada[2]

Received: April 2, 2020, Accepted: October 6, 2020

Abstract: Non-negative Matrix Factorization (NMF) is a popular unsupervised pattern recognition technique for the analysis of aggregated data. In particular, Non-negative Multiple Matrix Factorization (NMMF) treats common elements from multiple data as common factors, and execute simultaneous decomposition effectively. In this study, we propose a novel matrix factorization method called regional Non-negative Matrix Factorization (rNMF), which factorises multiple matrics simultaneously, focusing on physical relation between aggregated data such as regional characteristics in addition to common factors. rNMF expresses data of physically close areas in a similar feature space, and extracts intuitively interpretable bases and coefficients from multiple matrics. The information of regional location is given by a graph. Furthermore, by solving the graph coloring problem heuristically, rNMF works at high speed on a distributed system even if the analyzed data are large matrices. In this paper, we formulate rNMF as an extended version of NMF and derive multiplicative update rules for parameter estimation. We performed experiment with real data, which were aggregated by region, in order to verify that rNMF can expresses adjacent regional data in a common feature, rNMF attained similar generalization performance as the original NMF, and rNMF works at high speed on a distributed system.

Keywords: matrix factorization, parallelization, distributed processing

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

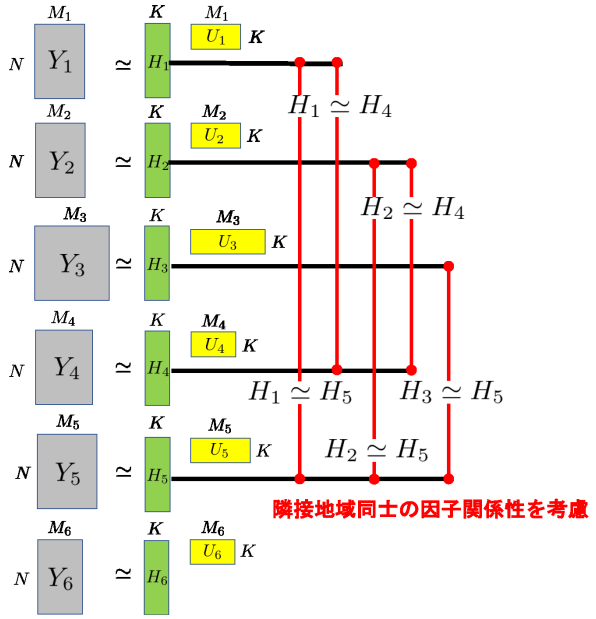
[1] 東京理科大学理工学部情報科学科
Department of Information Sciences, Tokyo University of Science, Noda, Chiba 278–8510, Japan
[2] 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories, Soraku-gun, Kyoto 619–0237, Japan
[3] 日本電信電話株式会社 NTT サービスエボリューション研究所
NTT Service Evolution Laboratories, Yokosuka, Kanagawa 239–0847, Japan
[†1] 現在，京都大学大学院情報学研究科
Presently with Graduate School of Informatics, Kyoto University
[†2] 現在，株式会社 ALBERT
Presently with ALBERT Inc.

[a)] touhoucrisis7@gmail.com

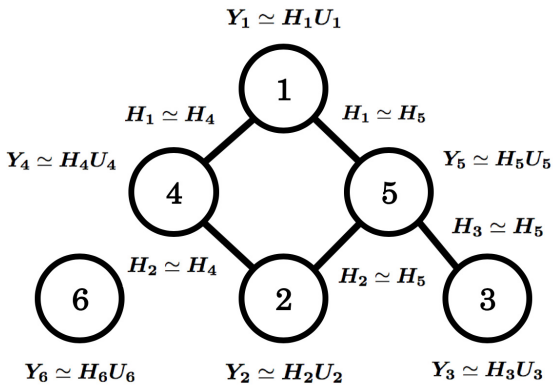
本論文の内容は 2019 年 7 月のマルチメディア，分散，協調とモバイル（DICOMO2019）シンポジウムで報告され，マルチメディア通信と分散処理研究会主査により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

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，ほか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．なかでも，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．たと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，潜在的特徴を抽出し，分析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，データの構造や特徴を正しく理解するため，データから得られる潜在的特徴がより直感的に分かりやすいことが必要となる．特に地域性のあるデータは，地域性を考慮することで潜在的特徴が直感的に分かりやすくなる場合がある．地域性を考慮する 1 つの方法として，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するというものが考えられる．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．



(a) 行数の等しい複数地域にまたがる行列データを，地域間の関係性を考慮し，地域ごとに因子分解を行う．



(b) 隣接地域の関係性を表すグラフの例．

図 1 提案手法の概要図
Fig. 1 A schematic image of the proposed method.

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 $\{\boldsymbol{Y}_i\}_{i=1}^{I}$ を入力とし，観測行列 $\boldsymbol{Y}_i$ を 2 つの因子行列 $\boldsymbol{H}_i$, $\boldsymbol{U}_i$ へ分解し，$\{\boldsymbol{H}_i\}_{i=1}^{I}$, $\{\boldsymbol{U}_i\}_{i=1}^{I}$ を得る．このとき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特徴空間が得られるよう，対象の基底行列 $\boldsymbol{H}_i$, $\boldsymbol{H}_j$（$1 \leq i,\ j \leq I$）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．具体的には地域をグラフの頂点に対応させ，隣接関係をグラフの辺に対応させる．作られるグラフは最大次数の小さいグラフや平面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ときは，行列 $\boldsymbol{H}_i$, $\boldsymbol{H}_j$ の更新を非同期的に並列に行うことができ，地域ごとのデータを分散的に処理することができる．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分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 1 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．なか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 [14], [15] がある．以下では，従来の NMF の定式化を述べる．

非負値の分解前の行列 $\boldsymbol{Y} \in \mathbb{R}_{+}^{N \times M}$ 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 $\boldsymbol{H} \in \mathbb{R}_{+}^{N \times K}$

と重み行列 $\boldsymbol{U} \in \mathbb{R}_+^{K\times M}$ となる．$\boldsymbol{H}$, $\boldsymbol{U}$ の積 $\hat{\boldsymbol{Y}}$ が元の行列 $\boldsymbol{Y}$ の近似となり，$\hat{\boldsymbol{Y}}$ と $\boldsymbol{Y}$ の距離関数を最小化する問題を解く．一般的に Frobenius ノルムや一般化 KL ダイバージェンス，板倉斉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，式 (1) に示した 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 $\boldsymbol{H}$, $\boldsymbol{U}$ を非負値で初期化する．その後式 (2) を繰り返し用いて，行列 $\boldsymbol{H}$, $\boldsymbol{U}$ 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．

$$\begin{aligned} &\text{minimize} \quad ||\boldsymbol{Y}-\hat{\boldsymbol{Y}}||_F^2 \quad (\hat{\boldsymbol{Y}}=\boldsymbol{H}\boldsymbol{U}) \qquad (1)\\ &\text{subject to} \quad \boldsymbol{H}\in\mathbb{R}_+^{N\times K},\ \boldsymbol{U}\in\mathbb{R}_+^{K\times M} \end{aligned}$$

$$H_{n,k} \leftarrow H_{n,k}\frac{\left[\boldsymbol{Y}\boldsymbol{U}^{\mathsf{T}}\right]_{n,k}}{[\hat{\boldsymbol{Y}}\boldsymbol{U}^{\mathsf{T}}]_{n,k}},\ U_{k,m} \leftarrow U_{k,m}\frac{\left[\boldsymbol{H}^{\mathsf{T}}\boldsymbol{Y}\right]_{k,m}}{[\boldsymbol{H}^{\mathsf{T}}\hat{\boldsymbol{Y}}]_{k,m}} \qquad (2)$$

ただし $[\boldsymbol{A}]_{i,j}$ は行列 $\boldsymbol{A}$ の (i,j) 成分を表す．

3. 提案手法

本稿では，複数の非負値行列が与えられたときに，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 に示すとおり，地域ごとの行列 $\{\boldsymbol{Y}_i\}_{i=1}^I$ ($\boldsymbol{Y}_i \in \mathbb{R}_+^{N\times M_i}$) が与えられる．各行列をそれぞれ基底行列 $\{\boldsymbol{H}_i\}_{i=1}^I$ ($\boldsymbol{H}_i \in \mathbb{R}_+^{N\times K}$) と重み行列 $\{\boldsymbol{U}_i\}_{i=1}^I$ ($\boldsymbol{U}_i \in \mathbb{R}_+^{K\times M_i}$) に分解し，このとき同時に特定の基底行列 $\boldsymbol{H}_i$, $\boldsymbol{H}_j$ 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

本節では，提案手法の定式化を行う．与えられる入力と得られる出力，仮定する条件，最小化すべき目的関数，最適化するパラメータの更新方法について述べる．

入力

複数の非負値行列 $\{\boldsymbol{Y}_i\}_{i=1}^I$（$\boldsymbol{Y}_i \in \mathbb{R}_+^{N\times M_i}$）が与えられる．図 1 に示すとおり，各行列の行数は等しく，列数は異なってもよい．

出力

複数の非負値行列 $\{\boldsymbol{Y}_i\}_{i=1}^I$ から，基底行列 $\{\boldsymbol{H}_i\}_{i=1}^I$（$\boldsymbol{H}_i \in \mathbb{R}_+^{N\times K}$）と重み行列 $\{\boldsymbol{U}_i\}_{i=1}^I$（$\boldsymbol{U}_i \in \mathbb{R}_+^{K\times M_i}$）に分解される．

条件

出力で得る $\{\boldsymbol{H}_i\}_{i=1}^I$ のうち，$\boldsymbol{H}_i$, $\boldsymbol{H}_j$ 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 Gragh（$V=\{1,\cdots,I\}$, E）を与える．例として，$I=6$ の場合のグラフの例を図 1 に示す．

目的関数

最小化すべき目的関数の一般的な形を述べる．

$$\begin{aligned} &\text{minimize}\\ &\sum_{i=1}^{I}\left[D_R(\boldsymbol{Y}_i|\boldsymbol{H}_i\boldsymbol{U}_i)+\frac{\alpha}{|E_i|}\sum_{(i,j)\in E_i}D_G(\boldsymbol{H}_i|\boldsymbol{H}_j)\right] \qquad (3)\\ &\text{subject to } \boldsymbol{H}_i\in\mathbb{R}_+^{N\times K},\ \boldsymbol{U}_i\in\mathbb{R}_+^{K\times M_i} \quad (\forall i: 1\le i\le I) \end{aligned}$$

ただし，E_i は頂点 i に接続する辺の集合，D_R, D_G は距離関数，α はハイパーパラメータである．式 (3) の左項は従来の 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 $\boldsymbol{H}_i$, $\boldsymbol{H}_j$ が同様のものとなる狙いがある．

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 (4) を用いて，目的関数を最小化する最適な $\{\boldsymbol{H}_i\}_{i=1}^I$, $\{\boldsymbol{U}_i\}_{i=1}^I$ が求められる．

$$\begin{aligned} &[\boldsymbol{H}_i]_{n,k} \leftarrow [\boldsymbol{H}_i]_{n,k}\frac{\left[\boldsymbol{Y}_i\boldsymbol{U}_i^{\mathsf{T}}\right]_{n,k}+\frac{\alpha}{|E_i|}\sum_{(i,j)\in E_i}[\boldsymbol{H}_j]_{n,k}}{[\hat{\boldsymbol{Y}}_i\boldsymbol{U}_i^{\mathsf{T}}]_{n,k}+\frac{\alpha}{|E_i|}\sum_{(i,j)\in E_i}[\boldsymbol{H}_i]_{n,k}}\\ &[\boldsymbol{U}_i]_{k,m} \leftarrow [\boldsymbol{U}_i]_{k,m}\frac{\left[\boldsymbol{H}_i^{\mathsf{T}}\boldsymbol{Y}_i\right]_{k,m}}{[\boldsymbol{H}_i^{\mathsf{T}}\hat{\boldsymbol{Y}}_i]_{k,m}} \quad (\hat{\boldsymbol{Y}}_i=\boldsymbol{H}_i\boldsymbol{U}_i) \qquad (4) \end{aligned}$$

距離関数を D_G を一般化 KL ダイバージェンスとするとき，式 (3) 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，$\boldsymbol{H}_i$ 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ときは，目的関数を式 (5) のように定義する．

$$\begin{aligned} &\text{minimize}\ \sum_{i=1}^{I}D_R(\boldsymbol{Y}_i|\boldsymbol{H}_i\boldsymbol{U}_i)\\ &\qquad +\frac{\alpha}{|E_i|}\sum_{(i,j)\in E_i}D_{KL}(\boldsymbol{H}_i+\boldsymbol{e}|\boldsymbol{H}_j+\boldsymbol{e}) \qquad (5)\\ &\text{subject to } \boldsymbol{H}_i\in\mathbb{R}_+^{N\times K},\ \boldsymbol{U}_i\in\mathbb{R}_+^{K\times M_i} \quad (\forall i: 1\le i\le I) \end{aligned}$$

ただし，$\boldsymbol{e}$ はすべての要素が 1 の行列とする．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き，式 (6) を用いて，目的関数を最小化する最適な $\{\boldsymbol{H}_i\}_{i=1}^I$, $\{\boldsymbol{U}_i\}_{i=1}^I$ が求められる．

$$\begin{aligned} &[\boldsymbol{H}_i]_{n,k} \leftarrow [\boldsymbol{H}_i]_{n,k}\frac{NL+\frac{\alpha}{|E_i|}\sum_{(i,j)\in E_i}NR}{DL+\frac{\alpha}{|E_i|}\sum_{(i,j)\in E_i}DR}\\ &[\boldsymbol{U}_i]_{k,m} \leftarrow \frac{[\boldsymbol{H}_i^{\mathsf{T}}(\boldsymbol{Y}_i\oslash\hat{\boldsymbol{Y}}_i)]_{k,m}}{\sum_{n=1}^{N}[\boldsymbol{H}_i]_{n,k}} \end{aligned} \qquad (6)$$

式 (6) 内の項 NL, NR, DL, DR にはそれぞれ式 (9), 式 (11) を入れる．ただし，$\oslash$ 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，$[\boldsymbol{H}_i]_{n,k}$ の更新式は式 (7) のように整理できる．式 (4) で

示した D_R，D_G をともに Frobenius ノルムとする場合は，NL，NR，DL，DR にそれぞれ式 (8)，式 (10) を入れることに対応する．

$$[\boldsymbol{H}_i]_{n,k} \leftarrow [\boldsymbol{H}_i]_{n,k} \frac{NL + \frac{\alpha}{|E_i|}\sum_{(i,j)\in E_i} NR}{DL + \frac{\alpha}{|E_i|}\sum_{(i,j)\in E_i} DR} \tag{7}$$

- D_R：Frobenius ノルム

$$NL = \left[\boldsymbol{Y}_i \boldsymbol{U}_i^{\mathsf{T}}\right]_{n,k},\ DL = [\hat{\boldsymbol{Y}}_i \boldsymbol{U}_i^{\mathsf{T}}]_{n,k} \tag{8}$$

- D_R：一般化 KL ダイバージェンス

$$NL = [(\boldsymbol{Y}_i \oslash \hat{\boldsymbol{Y}}_i)\boldsymbol{U}_i^{\mathsf{T}}]_{n,k},\ DL = \sum_{m=1}^{M_i} [\boldsymbol{U}_i]_{k,m} \tag{9}$$

- D_G：Frobenius ノルム

$$NR = [\boldsymbol{H}_j]_{n,k},\ DR = [\boldsymbol{H}_i]_{n,k} \tag{10}$$

- D_G：一般化 KL ダイバージェンス

$$\begin{aligned} NR &= \log\left([\boldsymbol{H}_j]_{n,k} + 1\right) + \frac{([\boldsymbol{H}_j]_{n,k} + 1)}{([\boldsymbol{H}_i]_{n,k} + 1)} \\ DR &= \log\left([\boldsymbol{H}_i]_{n,k} + 1\right) + 1 \end{aligned} \tag{11}$$

なお，式 (4) 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

Λ は添え字全体の集合，Γ は欠損値の添え字の集合とし，行列を定義する．

$$\boldsymbol{\Omega} = (\boldsymbol{\Omega}_{n,m}) = \begin{cases} 1 & ((n,m) \in \Lambda \backslash \Gamma) \\ 0 & ((n,m) \in \Gamma) \end{cases} \tag{12}$$

欠損値のある行列に対する従来の NMF の更新式を示す．

$$[\boldsymbol{H}_i]_{n,k} \leftarrow [\boldsymbol{H}_i]_{n,k} \frac{\left[(\boldsymbol{\Omega} \circ \boldsymbol{Y}_i)\boldsymbol{U}_i^{\mathsf{T}}\right]_{n,k}}{[(\boldsymbol{\Omega} \circ \hat{\boldsymbol{Y}}_i)\boldsymbol{U}_i^{\mathsf{T}}]_{n,k}}$$

$$[\boldsymbol{U}_i]_{k,m} \leftarrow [\boldsymbol{U}_i]_{k,m} \frac{\left[\boldsymbol{H}_i^{\mathsf{T}}(\boldsymbol{\Omega} \circ \boldsymbol{Y}_i)\right]_{k,m}}{[\boldsymbol{H}_i^{\mathsf{T}}(\boldsymbol{\Omega} \circ \hat{\boldsymbol{Y}}_i)]_{k,m}}$$

ただし，$\circ$ はアダマール積を表す．

3.3 効率的な並列化

与えられるグラフにおいて，頂点 i，j 間に辺が存在しないときは，行列 $\boldsymbol{H}_i$，$\boldsymbol{H}_j$ 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 $V = \{1, \cdots, I\}$ をグループ分けし，部分頂点集合の分割 V_g $(1 \le g \le G)$ 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．このとき，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．1 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 2 に示す．

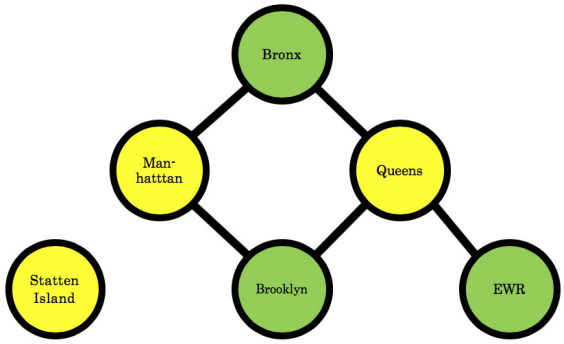


図 2 地域数 6 のデータのグラフと頂点彩色の例
Fig. 2 A graph G of 6 region and an example of vertex coloring.

Algorithm 1 rNMF

Require: $\{\boldsymbol{Y}_i\}_{i=1}^{I}$ $(\boldsymbol{Y}_i \in \mathbb{R}_{+}^{N \times M_i})$, K, α, T
一様分布からのサンプリングにより初期化：$\boldsymbol{H}_i, \boldsymbol{U}_i$ $(1 \le i \le I)$
Welsh-Powell によって頂点集合の分割 $\{V_1, \cdots, V_G\}$ を得る：
$\{V_1, \cdots, V_G\} \leftarrow \mathrm{WelshPowell}(V, E)$
for $iter = 0$ to $T - 1$ **do**
  for $g = 1$ to G **do**
    for each $i \in \mathcal{V}_g$ **do in parallel**

$$[\boldsymbol{H}_i]_{n,k} \leftarrow [\boldsymbol{H}_i]_{n,k} \frac{\left[\boldsymbol{Y}_i \boldsymbol{U}_i^{\mathsf{T}}\right]_{n,k} + \frac{\alpha}{|E_i|}\sum_{(i,j)\in E_i} [\boldsymbol{H}_j]_{n,k}}{[\hat{\boldsymbol{Y}}_i \boldsymbol{U}_i^{\mathsf{T}}]_{n,k} + \frac{\alpha}{|E_i|}\sum_{(i,j)\in E_i} [\boldsymbol{H}_i]_{n,k}}$$

$$[\boldsymbol{U}_i]_{k,m} \leftarrow [\boldsymbol{U}_i]_{k,m} \frac{\left[\boldsymbol{H}_i^{\mathsf{T}} \boldsymbol{Y}_i\right]_{k,m}}{[\boldsymbol{H}_i^{\mathsf{T}} \hat{\boldsymbol{Y}}_i]_{k,m}}$$

    end for
  end for
end for

3.4 アルゴリズム

提案手法を用いて，観測行列 $\boldsymbol{Y}_i$ $(1 \le i \le I)$ から因子行列 $\boldsymbol{H}_i$，$\boldsymbol{U}_i$ $(1 \le i \le I)$ を求めるアルゴリズムを Algorithm 1 に示す．ただし距離関数を D_R，D_G ともに Frobenius ノルムとする．

3.4.1 時間計算量

地域ごとの行列 $\{\boldsymbol{Y}_i\}_{i=1}^{I}$ を入力とし，各行列のサイズを $(N \times M_i)$ とする．$\{\boldsymbol{Y}_i\}_{i=1}^{I}$ を列方向に結合し，1 つの行列として従来手法を適用する．基底数 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は $O(NKT\sum_{i=1}^{I} M_i)$ である．

地域ごとの行列 $\{\boldsymbol{Y}_i\}_{i=1}^{I}$ の各行列に対し，並列に従来手法を適用する．ほかは同様の条件で分解を行う．このとき，時間計算量のオーダは $O(NKT \max_{1\le i\le I} M_i)$ である．

地域ごとの行列 $\{\boldsymbol{Y}_i\}_{i=1}^{I}$ に対し，並列に提案手法を適用する．ほかは同様の条件で分解を行う．前段階として

Welsh-Powell アルゴリズムによって，全体のグラフの頂点集合 $V = \{1, \cdots, I\}$ から頂点集合の分割 $\{V_1, \cdots, V_G\}$ 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である．Welsh-Powell アルゴリズムの時間計算量のオーダは $O\left(I^2\right)$ である．行列分解の計算量と合わせると，提案手法の時間計算量のオーダは $O(I^2 + NKT\sum_{g=1}^{G}\max_{i\in V_g} M_i)$ である．分割数 G は $G \leq I$ より，結合した 1 つの行列に従来手法を逐次的に適用する場合と，地域ごとの行列に提案手法を並列に適用する場合では，後者が高速に動作すると考えられる．また，与えられるグラフの最大次数 Δ(Graph) が小さい場合，$G \leq \Delta(\text{Graph}) + 1$ より，十分に高速に動作する．本稿では実装の簡単さから Welsh-Powell アルゴリズムを用いたが，平面グラフに対して $O(I)$ で 5 色以下の彩色が行えるアルゴリズム [13] が提案されている．このアルゴリズムを用いれば，与えられるグラフが平面グラフならば，$G \leq 5$ で高速に動作する．一方，地域ごとの行列に従来手法を並列に適用する場合と，地域ごとの行列に提案手法を並列に適用する場合では，前者の方が高速に動作すると考えられる．これは，地域ごとの基底行列 $\boldsymbol{H}_i$ を独立に求めることができるためである．一方で，地域性を考慮した基底行列を得ることはできない．

4. 実験

実験では提案手法を用いることで，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．

4.1 実験概要

実験では，基底行列と重み行列の更新に式 (2) を用いる従来手法と，式 (4) 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 $\{\boldsymbol{Y}_i\}_{i=1}^{I}$ が与えられる．特に断りのない限り，1 台の 4 コア 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 $\boldsymbol{Y}$ 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

すべての評価実験で 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：$\boldsymbol{Y}^*_{\text{small}} = \{\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_6\}$
  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 $\boldsymbol{Y}_i = (y^{(i)}_{nm})$ 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．$I = 6$，$N = 24$，$M_i = 365$（$i = 1 \cdots I$）である．$\boldsymbol{Y}^*_{\text{small}}$ を列方向に結合した行列は $\boldsymbol{Y}_{\text{small}}$ (24×2190) であり，行列 $\boldsymbol{Y}_{\text{small}} = (y_{nm})$ は，地域番号 $\lceil \frac{m}{6} \rceil$ における m $(mod\ 6)$ 日の n 時から $n+1$ 時までのタクシー乗車数を表す．地域番号と実際の地域との対応は，1：Bronx, 2：Brooklyn, 3：EWR, 4：Manhattan, 5：Queeens, 6：Staten Island とする．
- 行列データ 2：$\boldsymbol{Y}^*_{\text{large}} = \{\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_{234}\}$
  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 $\boldsymbol{Y}_i = (y^{(i)}_{nm})$ 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．$I = 234$，$N = 144$，$M_i = 365$（$i = 1 \cdots I$）である．$\boldsymbol{Y}^*_{\text{large}}$ を列方向に結合した行列は $\boldsymbol{Y}_{\text{large}}$ (144×85410) であり，行列 $\boldsymbol{Y}_{\text{large}} = (y_{nm})$ は，地域番号 $\lceil \frac{m}{234} \rceil$ における m $(mod\ 234)$ 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 を用いるときはグラフの頂点数は 6 とし，行列データ 2 を用いるときはグラフの頂点数は 234 とした．図 2 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．

4.4 基底行列の評価

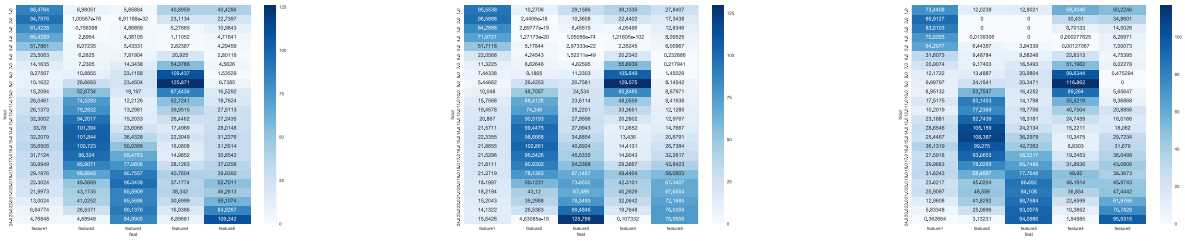
4.4.1 評価手法

従来手法で求めた基底行列間の距離と比較して，提案手法で求めた基底行列間の距離が近づいているか評価する実験を行った．行列データ 1 を用いた．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 $\boldsymbol{Y}^*_{\text{small}}$ に適用する場合，提案手法を $\boldsymbol{Y}^*_{\text{small}}$ 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と，提案手法適用時の D_R, D_G はすべて Frobenius ノルムとした．

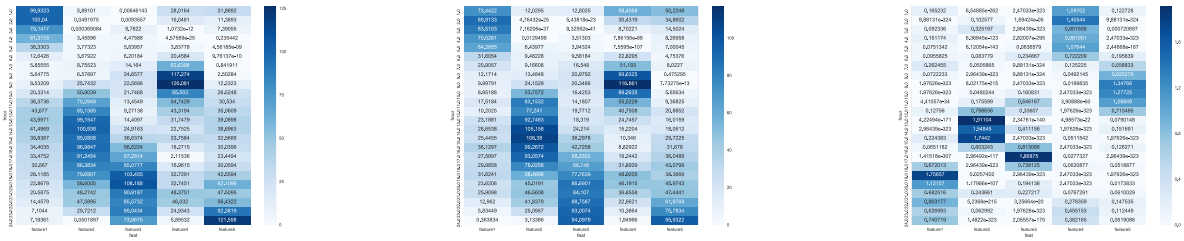
実験の詳細条件を以下で述べる．基底数 $K = 5$，イテレーション数 $T = 500$ とし，$\boldsymbol{H}_i$，$\boldsymbol{U}_i$ の初期化には，値域が $[0.0, \sqrt{\frac{4.0\sum_{(n,m)\in\Lambda_i}[\boldsymbol{Y}_i]_{n,m}}{6\times24\times365}}]$ の一様乱数を用いた．この乱数を用いることで，初期値の期待値が $E[[\boldsymbol{H}_i\boldsymbol{U}_i]_{n,m}] = \frac{\sum_{(n,m)\in\Lambda_i}[\boldsymbol{Y}_i]_{n,m}}{NM}$ を満たす．提案手法の更新式では，すべての実験で $\alpha = 250$ を用いた．

4.4.2 結果

図 3，**図 4** を比較すると，従来手法ではすべ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分かる．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

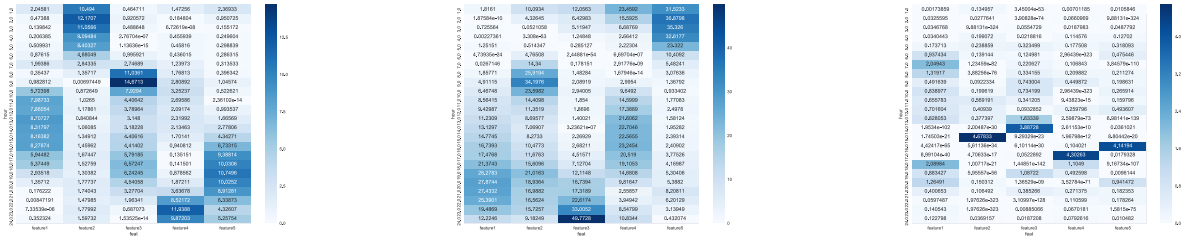


(a) Bronx の基底行列　(b) Brooklyn の基底行列　(c) EWR の基底行列

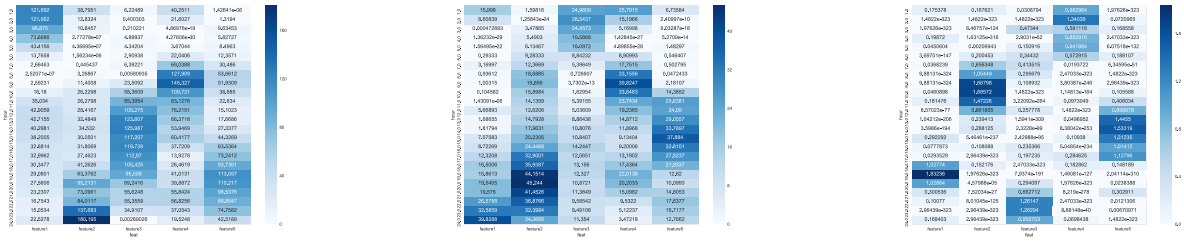


(d) Manhattan の基底行列　(e) Queens の基底行列　(f) Staten Island の基底行列

図 3　提案手法を複数の区分行列 $\boldsymbol{Y}^*_{\text{small}}$ に適用した場合の基底行列
Fig. 3　Base Matrix of regional data by proposed method.



(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 4　従来手法を複数の区分行列 $\boldsymbol{Y}^*_{\text{small}}$ に適用した場合の基底行列
Fig. 4　Base Matrix of regional data by conventional method.

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．隣接した地域のタクシーの乗車数の傾向は類似すると考えられるが，それをふまえたうえでは，図 4 の基底行列よりも図 3 の基底行列の方が直感的に理解しやすい．そして，データの構造や特徴を正しく理解するうえで，直感的に分かりやすい潜在的特徴が得られることは必要である．よって，提案手法は物理的に距離の近い地域のデータを同様の特徴空間で表現することができ，地域性を考慮した直感的に分かりやすい潜在的特徴を得ることができる利点があるといえる．

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 $\boldsymbol{Y}_{\text{small}}$ に適用する場合，提案手法を $\boldsymbol{Y}^*_{\text{small}}$ に適用する場合，$\boldsymbol{Y}^*_{\text{small}}$ を 3 階テンソルとして扱い，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，欠損のある行列と見なして式 (12) 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 Γ_i（$i = 1 \cdots I$）とすれば，入力行列

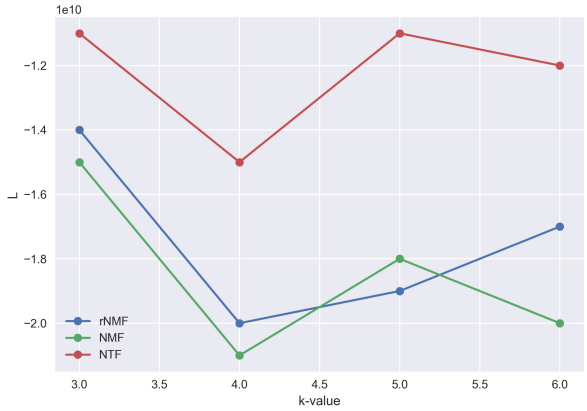


図 5 汎化性能（縦軸 L が大きいほど良い）
Fig. 5 Generalization performance.

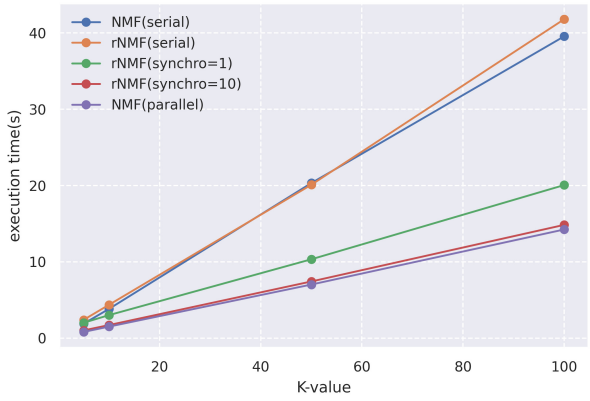


図 6 計算時間の評価（実験 1）
Fig. 6 Evaluation of execution time (the experiment 1).

$[\boldsymbol{Y}_i]_{n,m}$ $(n,m) \in \Gamma_i$ 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 $\hat{\boldsymbol{Y}}_i$ を得て，汎化性能の評価値を $L = \sum_{i=1}^{I} \sum_{(n,m)\in\Gamma_i} \log L([\boldsymbol{Y}_i]_{n,m} \mid [\hat{\boldsymbol{Y}}_i]_{n,m})$ として算出する．ただし，L は元の観測行列 $\boldsymbol{Y}_i$ 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 ノルムとしたとき，$[\boldsymbol{Y}_i]_{n,m}$ 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，基底数 K 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 $T = 500$ とし，$\boldsymbol{H}_i$，$\boldsymbol{U}_i$ の初期化には，値域が $[0.0, \sqrt{\frac{4.0\sum_{(n,m)\in\Lambda_i}[\boldsymbol{Y}_i]_{n,m}}{6\times24\times365}}]$ 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 $E[[\boldsymbol{H}_i\boldsymbol{U}_i]_{n,m}] = \frac{\sum_{(n,m)\in\Lambda_i}[\boldsymbol{Y}_i]_{n,m}}{NM}$ を満たす．実験結果を**図 5** 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 $\boldsymbol{Y}_{\text{small}}$ に適用する場合，従来手法を $\boldsymbol{Y}^*_{\text{small}}$ に適用する場合，提案手法を $\boldsymbol{Y}^*_{\text{small}}$ に適用する場合で比較を行った．従来手法を $\boldsymbol{Y}_{\text{small}}$ に適用する場合は逐次実行，従来手法を $\boldsymbol{Y}^*_{\text{small}}$ に適用する場合はマルチプロセス実行での計算時間を比較した．提案手法の実用上の工夫として，パラメータ $\boldsymbol{H}_i$，$\boldsymbol{U}_i$ 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 3 通りの異なる実装を行い，計算時間を比較した．なお D_R，D_G はともに Frobenius ノルムとし，基底数 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 $T = 500$ とし，$\boldsymbol{H}_i$，$\boldsymbol{U}_i$ の初期化には，値域が $[0.0, \sqrt{\frac{4.0\sum_{(n,m)\in\Lambda_i}[\boldsymbol{Y}_i]_{n,m}}{6\times24\times365}}]$ 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 $E[[\boldsymbol{H}_i\boldsymbol{U}_i]_{n,m}] = \frac{\sum_{(n,m)\in\Lambda_i}[\boldsymbol{Y}_i]_{n,m}}{NM}$ を満たす．

4.6.2 結果

実験結果を**図 6** に示した．横軸が基底数 K，縦軸が計算時間である．図 6 中の，NMF（serial）は従来手法を $\boldsymbol{Y}_{\text{small}}$ に適用した場合，rNMF（serial）は逐次実行で提案手法を $\boldsymbol{Y}^*_{\text{small}}$ に適用した場合，rNMF（synchro=1）はマルチプロセス実行で提案手法を $\boldsymbol{Y}^*_{\text{small}}$ に適用した場合，rNMF（synchro=10）はマルチプロセス実行で提案手法を $\boldsymbol{Y}^*_{\text{small}}$ に適用し，10 回の更新をまとめて行う場合，NMF（parallel）はマルチプロセス実行で従来手法を $\boldsymbol{Y}^*_{\text{small}}$ に適用した場合をそれぞれ示す．マルチプロセス実行で提案手法を $\boldsymbol{Y}^*_{\text{small}}$ に適用する場合と，マルチプロセス実行で従来手法を $\boldsymbol{Y}^*_{\text{small}}$ に適用する場合では，後者の方が計算時間が短い．しかし前者と後者の計算時間の差は $K = 100$ の場合でも 6 秒ほどであり，地域性を考慮することによる提案手法の計算量の増加は小さいといえる．マルチプロセス実行で提案手法を $\boldsymbol{Y}^*_{\text{small}}$ に適用する場合と，従来手法を $\boldsymbol{Y}_{\text{small}}$ に適用する場合では，前者の方が計算時間が短い．逐次実行で提案手法を $\boldsymbol{Y}^*_{\text{small}}$ に適用する場合と従来手法を $\boldsymbol{Y}_{\text{small}}$ に適用する場合では計算時間に差はなく，並列化による高速化が実現できているといえる．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計算時間が短い．ただし，まとめて更新を行う回数を増やし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，通常の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．

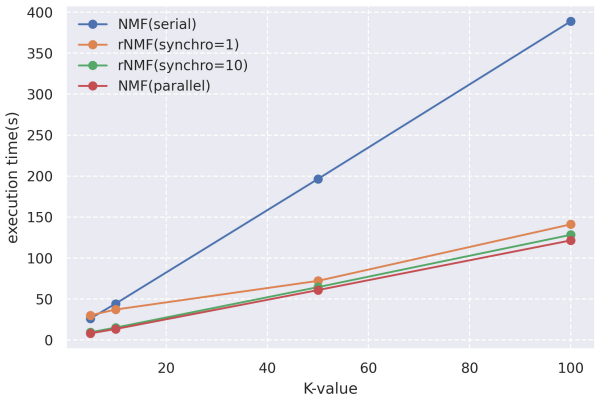


図 7 計算時間の評価（実験 2）
Fig. 7 Evaluation of execution time (the experiment 2).

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 $\boldsymbol{Y}_{\text{large}}$ に適用する場合，従来手法を $\boldsymbol{Y}^*_{\text{large}}$ に適用する場合，提案手法を $\boldsymbol{Y}^*_{\text{large}}$ 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 2 通りの異なる実装を行い，計算時間を比較した．なお D_R，D_G はともに Frobenius ノルムとし，基底数 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 $T = 100$ とし，$\boldsymbol{H}_i$, $\boldsymbol{U}_i$ の初期化には，値域が $[0.0, \sqrt{\frac{4.0 \sum_{(n,m)\in\Lambda_i}[\boldsymbol{Y}_i]_{n,m}}{6\times 24\times 365}}]$ 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 $E[[\boldsymbol{H}_i\boldsymbol{U}_i]_{n,m}] = \frac{\sum_{(n,m)\in\Lambda_i}[\boldsymbol{Y}_i]_{n,m}}{NM}$ を満たす．

4.7.2 結果

実験結果を**図 7** に示した．横軸が基底数 K，縦軸が計算時間である．図 7 中の，NMF（serial）は従来手法を $\boldsymbol{Y}_{\text{large}}$ に適用した場合，rNMF（synchro=1）はマルチプロセス実行で提案手法を $\boldsymbol{Y}^*_{\text{large}}$ に適用した場合，rNMF（synchro=10）はマルチプロセス実行で提案手法を $\boldsymbol{Y}^*_{\text{large}}$ に適用し，10 回の更新をまとめて行う場合，NMF（parallel）はマルチプロセス実行で従来手法を $\boldsymbol{Y}^*_{\text{large}}$ に適用した場合をそれぞれ示す．計算時間の優位性の順序関係は，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，提案手法を $\boldsymbol{Y}^*_{\text{large}}$ に適用する場合と，従来手法を $\boldsymbol{Y}_{\text{large}}$ に適用する場合で，より計算時間の差が大きくなっている．たとえば $K = 100$ で従来手法は提案手法に対して，実験 1 は約 1.95 倍であったが，実験 2 は約 2.75 倍になっている．よって，分解を行う行列サイズが大きいとき並列化がより高速化に有効であるといえる．

5. おわりに

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，2 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計算時間が速く，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．

参考文献

[1] Lee, D.D. and Seung, H.S.: Learning the parts of objects with nonnegative matrix factorization, *Nature*, Vol.401, pp.788–791 (1999).
[2] Cichocki, A., Zdunek, R., Phan, A.H., et al.: Nonnegative 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 nonnegative 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] Smaragdis, P. and Brown, J.C.: Non-negative matrix factorization for polyphonic music transcription, *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., et al.: Distributed nonnegative matrix factorization for web- scale dyadic data analysis on MapReduce, *Proc. WWW* (2010).
[9] Liu, S., Flach, P. and Cristianini, N.: Generic Multiplicative Methods for Implementing Machine Learning Algorithms on MapReduce, *CoRR* (2011).
[10] Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints, *Journal of Machine Learning Research*, Vol.5, pp.1457–1469 (2004).
[11] Sindhwani, V. and Ghoting, A.: Large-scale distributed nonnegative sparse coding and sparse dictionary learning, *Proc. 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD'12*, p.489 (2012).
[12] Welsh, D.J.A. and Powell, M.B.: An upper bound for the chromatic number of a graph and its application to timetabling problems, *The Computer Journal*, Vol.10,

No.1, pp.85–86 (1967).
[13] Chiba, N., Nishizeki, T. and Saito, N.: A linear algorithm for five-coloring a planer graph, *Graph Theory and Algorithms*, Lecture Notes in Computer Science, Vol.108, pp.9–19, Springer (1981).
[14] 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，情報処理学会論文誌数理モデル化と応用，Vol.7, No.1, pp.71–83 (2013).
[15] 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，人工知能学会論文誌，Vol.30, No.6, pp.745–754 (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．

$$f\left(\{\boldsymbol{H}_i\}_{i=1}^{I}, \{\boldsymbol{U}_i\}_{i=1}^{I}\right)$$

表 A·1 記法
Table A·1 notation.

記号	意味
$\boldsymbol{Y}$	入力行列
$\boldsymbol{H}$	$\boldsymbol{Y}$ の因子行列（基底）
$\boldsymbol{U}$	$\boldsymbol{Y}$ の因子行列（重み）
$\hat{\boldsymbol{Y}}$	因子行列の積 $\boldsymbol{HU}$
N	入力行列の行数
K	因子行列の基底数.
M_i	入力行列 Y_i の列数
I	入力行列を分割した小行列の数
$\boldsymbol{Y}_i$	入力行列を分割した i 番目の小行列
$\boldsymbol{H}_i$	$\boldsymbol{Y}_i$ から得られる因子行列（基底）
$\boldsymbol{U}_i$	$\boldsymbol{Y}_i$ から得られる因子行列（重み）
$\hat{\boldsymbol{Y}}_i$	因子行列の積 $\boldsymbol{H}_i\boldsymbol{U}_i$
$[\boldsymbol{A}]_{i,j}$	行列 $\boldsymbol{A}$ の (i,j) 成分
D_R	再構成誤差
D_G	基底行列間の距離関数
α	ハイパーパラメータ
T	イテレーション数
V	グラフの頂点集合
E	グラフの辺の集合
E_i	頂点 i と連結な頂点集合
Λ_i	添字の集合 $\{(n,m) \mid 1 \le n \le N,\ 1 \le m \le M_i\}$
Γ_i	欠損値の添字の集合 $\Gamma_i \subset \Lambda_i$
$\boldsymbol{Y}_{\text{small}}$	実験で用いたサイズが小さい入力行列
$\boldsymbol{Y}^*_{\text{small}}$	入力行列 $\boldsymbol{Y}_{\text{small}}$ の小行列の集合
$\boldsymbol{Y}_{\text{large}}$	実験で用いたサイズが大きい入力行列
$\boldsymbol{Y}^*_{\text{large}}$	入力行列 $\boldsymbol{Y}_{\text{large}}$ の小行列の集合．分割単位が小さい．

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

Jensen の不等式より，

$$\le \sum_{i=1}^{I}\sum_{(n,m)\in\Lambda_i} |\,[\boldsymbol{Y}_i]_{n,m}\,|^2 - 2\,[\boldsymbol{Y}_i]_{n,m}\,[\boldsymbol{H}_i\boldsymbol{U}_i]_{n,m} + \sum_{t=1}^{K}\frac{[\boldsymbol{H}_i]_{n,t}^2\,[\boldsymbol{U}_i]_{t,m}^2}{\lambda_{i,t,n,m}} + \frac{\alpha}{|E_i|}\sum_{(i,j)\in E_i}\sum_{(n,k)} |\,[\boldsymbol{H}_i]_{n,k} - [\boldsymbol{H}_j]_{n,k}\,|^2$$
$$= G\left(\{\boldsymbol{H}_i\}_{i=1}^{I}, \{\boldsymbol{U}_i\}_{i=1}^{I}\right)$$

ただし，$\sum_{t=1}^{K}\lambda_{i,t,n,m} = 1$, $\lambda_{i,t,n,m} \ge 0$ であり，G を最小化することで，元の目的関数 f が最小化される．

$$\lambda_{i,t,n,m} \leftarrow \underset{\lambda_{i,t,n,m}}{\arg\min}\ \frac{[\boldsymbol{H}_i]_{n,t}\,[\boldsymbol{U}_i]_{t,m}}{\sum_{t'}[\boldsymbol{H}_i]_{n,t'}\,[\boldsymbol{U}_i]_{t',m}}$$

を用いると以下のように偏微分できる．

$$\frac{\partial G}{\partial\,[\boldsymbol{H}_i]_{n,k}} = -2\left[\boldsymbol{Y}_i\boldsymbol{U}_i^{\mathsf{T}}\right]_{n,k} + 2\sum_{m=1}^{M_i}\frac{[\boldsymbol{H}_i]_{n,k}\,[\boldsymbol{U}_i]_{k,m}^2}{\lambda_{i,k,n,m}} + 2\frac{\alpha}{|E_i|}\sum_{(i,j)\in E_i}\left([\boldsymbol{H}_i]_{n,k} - [\boldsymbol{H}_j]_{n,k}\right) = 0$$

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

これを解けば，式 (4) の更新式が得られる．

推薦文

本論文は DICOMO2019 の DPS 研究会関連セッションで発表された論文のうち，座長と評価委員 2 名による評価において最も高いスコアを獲得した．地域性などの物理的な関係性を持つ集計データにたいして非負値行列因子分解を行う方法の提案であり，大規模な行列であっても分散システム上で処理可能となることを実現している．性能評価として既存の非負値行列因子分解と比較し，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確認していることも評価できる．

（マルチメディア通信と分散処理研究会主査 田上 敦士）

越塚 毅

2017 年東京理科大学理工学部学士課程入学．2018 年日本電信電話株式会社 NTT コミュニケーション科学基礎研究所のインターンシップに参加．

竹内 孝

2011 年早稲田大学大学院先進理工学研究科修士課程修了．2019 年京都大学大学院情報学研究科博士後期課程修了．博士（情報学）．2011 年日本電信電話株式会社 NTT コミュニケーション科学基礎研究所，2020 年より京都大学大学院情報学研究科助教，機械学習およびデータマイニングの研究に従事．

松林 達史 （正会員）

2000 年京都大学理学部物理学科卒業．2002 年理化学研究所非常勤研究員．2005 年東京工業大学大学院理工学研究科地球惑星科学専攻博士課程修了．同年日本電信電話株式会社 NTT コミュニケーション科学基礎研究所，2010 年同社 NTT サービスエボリューション研究所．2015 年電気通信大学大学院情報理工学研究科客員准教授．2020 年株式会社 ALBERT にて，機械学習を用いた画像認識およびロボティクスの研究開発に従事．博士（理学）．

澤田 宏

1991 年京都大学工学部情報工学科卒業．1993 年同大学院工学研究科情報工学専攻修士課程修了．2001 年博士（情報学，京都大学）．1993 年日本電信電話株式会社（NTT）入社．以来，VLSI 設計技術，計算機アーキテクチャ，信号処理，機械学習の研究に従事．現在，NTT コミュニケーション科学基礎研究所協創情報研究部 部長/上席特別研究員．IEEE Fellow，電子情報通信学会フェロー，日本音響学会会員．