

## 補助情報を用いたテンソル分解

成田敦博<sup>†</sup> 林浩平<sup>††</sup> 富岡亮太<sup>†††</sup> 鹿島久嗣<sup>†††\*</sup>

<sup>†</sup> 東京大学工学部 計数工学科

<sup>††</sup> 東京大学大学院 情報理工学系研究科

<sup>†††</sup> 奈良先端科学技術大学院大学 情報科学研究科

\* JST さきがけ「知の創生と情報社会」研究領域

E-mail: <sup>††</sup>t090594@mail.ecc.u-tokyo.ac.jp, <sup>††</sup>kohei-h@is.naist.jp, <sup>†††</sup>{tomioka,kashima}@mist.i.u-tokyo.ac.jp

あらまし テンソルの未観測部分の補完は、通常、対象となるテンソルが低ランクであることを仮定することによって行われる。しかし、補完するテンソルの未観測部分の割合が高い場合、すなわち疎である場合には補完精度が悪くなることが知られている。本研究では疎なテンソルの補完問題において補完精度を向上させるため、低ランク性の仮定に加えて、データ間の関係性を補助情報として考慮する方法を提案する。テンソルの低ランク分解にグラフラプリアンによる正則化項を導入することによってテンソルの補完にデータ間の補助情報を導入した定式化を提案し、これを解くためのアルゴリズムを示す。人工データと実データを用いた数値実験によって、補助情報を導入することでテンソルが疎な場合における補完精度が実際に向上することを検証する。

キーワード テンソル分解, テンソル補完, 補助情報, グラフラプリアン

## Tensor factorization using auxiliary information

Atsuhiko NARITA<sup>†</sup>, Kohei HAYASHI<sup>††</sup>, Ryota TOMIOKA<sup>†††</sup>, and Hisashi KASHIMA<sup>†††\*</sup>

<sup>†</sup> Department of Mathematical Engineering and Information Physics, School of Engineering, The University of Tokyo

<sup>†††</sup> Graduate School of Information Science and Technology, The University of Tokyo

<sup>††</sup> Graduate School of Information Science, Nara Institute of Science and Technology

\* Basic Research Programs PRESTO, Synthesis of Knowledge for Information Oriented Society

E-mail: <sup>††</sup>t090594@mail.ecc.u-tokyo.ac.jp, <sup>††</sup>kohei-h@is.naist.jp, <sup>†††</sup>{tomioka,kashima}@mist.i.u-tokyo.ac.jp

**Abstract** Most of the existing completion methods of tensors (i.e. multi-way arrays) only assume that tensors to be completed are low rank. However, it is known that accuracy of completion tends to be worse when only limited entries are observed. In this paper, we propose to use relationships among data as auxiliary information in addition to the low-rank assumption to improve accuracy. We introduce two regularization approaches using graph Laplacians induced from the relationships, and design approximation algorithms for the optimization problems. Numerical experiments using synthetic and real datasets show that use of auxiliary information improves completion accuracy over the existing methods based only on the low-rank assumption, especially when observations are sparse.

**Key words** Multi-way arrays; Tensor decomposition; Tensor completion; Auxiliary information; Graph Laplacian

### 1. はじめに

さまざまな出来事が、いくつかの物事どうしのかかわりによって表される。対象が二種類であれば二次元の配列としての行列によってデータを表すことができるが、より多くの対象が同時に関わってくるのなら、高次元の配列、つまりテンソルという形で表すことができる。

誤差が大きい、実験にコストがかかるために一部しか観測で

きないなどといった理由で、データに欠損が生じるといった状況は様々に発生する。欠損の生じたデータはそのままでは解析に適さないといった理由で、多くの場合満足に利用することができず、そのまま無駄になってしまうことになる。このような場合には未知のデータを推定するということが必要になってくる。また欠損したデータを推定することが出来れば、初めから一部しか観測を行わないことでコストや手間を軽減することができる。組み合わせの数が膨大でなためにそもそも網羅的な観

測が不可能な場合であっても、一部の観測から全体像を捉えるといったことも考えられる。そのような目的のために、テンソルを補完する手法が重要になってくる。

テンソルの補完にはいろいろな方法が提案されているが、手がかかりとしてそこに低ランクな構造があることを仮定し、観測されたデータを再現するようなランクの低いテンソルを計算する手法 [7] が代表的であり、近年盛んに研究されている。しかし得られるデータが非常に限られてしまい、ごく僅かな要素しか観測されていないテンソルを補完しなければならない場合には、低ランク性の仮定のみでは十分に精度の良い補完ができないことが考えられる。

ここで、そもそも変数の連続性や、「似ている」であろう対象の情報など、データ間の関係に関する情報を外部から与えて補完のために利用することを考えることができる。テンソルの低ランク性に加えてこれらの情報を利用することで、補完をより良い精度で行えないだろうか。

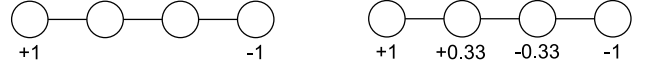
既に行列の場合においては、低ランクな行列の分解に補助情報を取り入れる方法が Li らによって示されている [6]。データ間の関係性を類似度のグラフ構造と考え、行列の低ランク分解に対してグラフラプシアンを用いた正則化を行うことで、低ランク性とデータ間の関係の情報を統一的に扱うものである。

この研究ではその考え方をより一般のテンソルに対して拡張し、テンソルの補完精度を向上させるために外部から補助情報を導入する二種類の手法を提案する。

テンソルの低ランク分解では主に CP 分解と Tucker 分解という二つの方法が存在する [4]。これらの計算は元のテンソルとそれを低ランクで近似したテンソルの距離を最小化するという最適化問題として定式化される。そこへ補助情報を利用した正則化を行うことで、テンソルの低ランク分解に自然な形で補助情報を導入する。具体的には、因子行列に対しての二通りの正則化を提案し、それぞれを解くアルゴリズムを CP 分解と Tucker 分解両方について説明する。

加えて、それらの手法を比較するために、実際にテンソルを補完する数値実験を人工的に生成した低ランクなテンソルと、三次元配列の形を取る三種類の実データに対して実施した。その結果として、全てのデータにおいて補助情報を導入することで補完精度が向上することを確認した。特に未観測要素が非常に高い割合で存在する場合に顕著で、適切にデータ間の補助情報を導入することで、テンソルの補完精度を高めることができるということを示す。

この論文の構成は以下のとおりである。まず 2 章でテンソルの性質と、低ランクなテンソルを分解するための基本的な概念について説明する。次に 3 章において、テンソルの分解に外部からデータの補助情報を加え、テンソルの低ランク性の仮定にデータ間の関係の情報を方法を導入する方法を二種類提案する。4 章では数値実験を行う。EM アルゴリズムによって低ランク分解を用いたテンソルの補完を行うアルゴリズムについて説明し、人工データはと実データに対して要素補完の数値実験を行う。それによって、補助情報を利用することで補完精度を大きく向上させることができることを示す。



(a) 類似度を表すグラフ

(b) 似ているものは近い値を割り当てる

## 2. グラフ構造を用いる補助情報の導入

まずより一般的に、データ間のグラフ構造を用いた正則化の方法について説明する。

何かしら制約条件の元で、似ているデータ同士はなるべく近い値を持っているようにしたい。例えば、次の図 2. のような四つのノードからなるグラフがあり、左右端のノードのみに  $+1, -1$  の値が振られている場合を考える。ここでこのグラフは、枝があるノード同士は互いに「似ている」ことを表すものとする。

似ているノードどうしはなるべく近い値を持つようにしたいので、中ふたつのノードは図 2. のような値を持てばよいということになる。

次にこの考え方を定量的に行う。ある  $D$  次元のデータ列  $\{\mathbf{x}_i\}_{i=1}^n, \mathbf{x} \in \mathbb{R}^D$  が与えられたとし、それぞれのデータがどれだけ似ているか、を表す類似度行列  $A_{ij} (1 \leq i \leq n, 1 \leq j \leq n)$  が設定されているとする。ここで類似度行列は対称である。データ同士の距離をベクトルとして二乗ノルムで考えるとすると、ある添字  $i$  と  $j$  を選んだ時に、

$$\frac{1}{2} A_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \quad (1)$$

が小さくなればよい。

データ列全体に対しての目的関数をつくるために、全てのデータの組に対して足し合わせると、

$$\begin{aligned} l &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n A_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \left[ A_{ij} \sum_{d=1}^D (x_{id} - x_{jd})^2 \right] \\ &= \frac{1}{2} \sum_{d=1}^D \left[ \sum_{i=1}^n \sum_{j=1}^n A_{ij} (x_{id} - x_{jd})^2 \right] \end{aligned}$$

ここで、 $\mathbf{X}$  を  $\{\mathbf{x}_i\}_{i=1}^n$  を横に並べた行列とし、 $\mathbf{X}_{d*}$  を  $\mathbf{X}$  の  $d$  番目の行であるとする、上の式は

$$\frac{1}{2} \sum_{d=1}^D \mathbf{X}_{d*} (\mathbf{D} - \mathbf{A}) \mathbf{X}_{d*}^\top \quad (2)$$

と書くことができる。このとき  $\mathbf{D}$  は

$$\mathbf{D} = \text{diag} \left[ \sum_{i=1}^n A_{1i}, \sum_{i=1}^n A_{2i}, \dots, \sum_{i=1}^n A_{ni} \right]$$

という対角行列であり、グラフにおけるそれぞれのノードの次数を対角成分に持つような行列である。ここで、

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

をグラフラブラシアンと呼び、代数的にグラフを扱う際に非常に重要な役割を持っている。このグラフラブラシアンの性質や応用に関しては [3] などが詳しい。

なお、本研究では

$$\tilde{L} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

のように対角成分を 1 に揃えた正規化ラブラシアンを使用している。

最後に、 $X_{d*}$  を  $X$  の  $d$  番目の行ベクトルとすると、(2) はグラフラブラシアンを使って

$$\frac{1}{2} \sum_{d=1}^D X_{d*} L X_{d*}^\top = \text{tr} (X L X^\top)$$

とできるので、これを使って正則化を行う。

### 3. 補助情報を用いた行列分解 (既存手法)

行列のノルムに加えて補助情報によって正則化をしながら行列を分解する方法 [6] が Li らによって提案されている。ここでは次のような最適化問題を解き、行列の分解にデータ同士の関係の情報を導入している。

$$\min_{U, V} \frac{1}{2} \|X - UV^\top\|_F^2 + \frac{\alpha}{2} (\|U\|^2 + \|V\|^2) + \frac{\beta}{2} \text{tr}(U^\top L_1 U) \quad (3)$$

ここで、

$$X \in \mathbb{R}^{I \times J}, U \in \mathbb{R}^{I \times R}, V \in \mathbb{R}^{J \times R}$$

である。

上の式 (3) では、行列  $X$  を低ランク (ランク  $R$ ) な行列  $UV^\top$  で近似する二乗誤差の項、 $U, V$  の二乗ノルムによる正則化の項に加えて、グラフラブラシアンによる正則化項を加えることで外部からの補助情報を導入している。この項によって、似ているデータはなるべく近い特徴ベクトル ( $U, V$  の行ベクトル) を持つように行列を分解することができる。

### 4. 提案手法 A:Sum

次に、前節における行列の分解に補助情報を用いる考え方をテンソルに発展させ、テンソルの低ランク性という仮定に加えて、要素間の関係の情報を補完に用いるための方法を示す。

#### 4.1 正則化

(3) をテンソルの場合に拡張する。より一般の場合への拡張は容易であるので、ここでテンソルは 3 階であるとし、補助情報は  $U, V, W$  に対応するモードに対してそれぞれ  $L_1, L_2, L_3$  として与えられているものとする。この時、式 (3) からの類推により、補助情報を導入するために必要な正則化項を

$$\frac{\alpha}{2} \text{tr} (U^\top L_1 U + V^\top L_2 V + W^\top L_3 W) \quad (4)$$

として定める。

#### 4.2 CP 分解への補助情報の導入

CP 分解の最小化問題としての定式化に、式 (4) によって正則化を行ったもの考えると、

$$X \in \mathbb{R}^{I \times J \times K}, U \in \mathbb{R}^{I \times R}, V \in \mathbb{R}^{J \times R}, W \in \mathbb{R}^{K \times R}$$

として

$$f(\mathcal{X}, U, V, W) = \frac{1}{2} \|\mathcal{X} - \mathcal{J} \times_1 U \times_2 V \times_3 W\|_F^2 + \frac{\alpha}{2} \text{tr} (U^\top L_1 U + V^\top L_2 V + W^\top L_3 W) \quad (5)$$

という目的関数が得られ、 $\min_{U, V, W} f(U, V, W)$  を解けばよいことになる。ここで、 $\times_n$  は  $n$  番目のモードに対するテンソル積で、 $n$ -モード積と呼ばれる。また、 $\mathcal{J} \in \mathbb{R}^{R \times R \times R}$  は超対角な単位テンソルである。テンソルの演算に関する基本的な性質については Kolda らによる論文 [5] に詳しい。この論文ではその表記に従っている。

#### 4.2.1 変数ごとの最適化

式 (5) は全体としては凸関数にはなっていないが、それぞれの変数  $U, V, W$  については凸関数の形となっている。そこで変数一つずつ繰り返し最適化する方法を取ることにする。

その他の変数の場合も同様に行えるので、ここでは  $V, W$  を固定した状態で (5) を最小化するような  $U$  を求めることを考える。 $X_{(n)}$  をテンソル  $\mathcal{X}$  のモード  $n$  展開であるとする。(5) をモード 1 で展開すると、

$$\begin{aligned} f(\mathcal{X}, U, V, W) &= \frac{1}{2} \|\mathcal{X}_{(1)} - U(W \odot V)^\top\|_F^2 \\ &\quad + \frac{\alpha}{2} \text{tr} (U^\top L_1 U + V^\top L_2 V + W^\top L_3 W) \\ &= \frac{1}{2} \text{tr} \left( (\mathcal{X}_{(1)} - U(W \odot V)^\top)^\top (\mathcal{X}_{(1)} - U(W \odot V)^\top) \right) \\ &\quad + \frac{\alpha}{2} \text{tr} (U^\top L_1 U + V^\top L_2 V + W^\top L_3 W) \end{aligned}$$

ここで  $\odot$  は Khatri-Rao 積である。これを  $U$  について微分すると

$$\begin{aligned} \frac{\partial f}{\partial U} &= \\ &= -\mathcal{X}_{(1)}(W \odot V) + U(W \odot V)^\top(W \odot V) + \alpha L_1 U \end{aligned}$$

となり、左辺に 0 を代入することで

$$\begin{aligned} U(W \odot V)^\top(W \odot V) + \alpha L_1 U &= \\ U(V^\top V * W^\top W) + \alpha L_1 U &= \mathcal{X}_{(1)}(W \odot V) \quad (6) \end{aligned}$$

という形が得られる。これは  $U$  についての Sylvester 方程式の形となっており、これを解くことで  $U$  についての最適化が行える。

#### 4.3 Tucker 分解への補助情報の導入

次に、Tucker 分解へ補助情報を導入することを考えるため、CP 分解の場合と同様に (3) をテンソルに拡張してみる。Tucker 分解の最小化問題としての定式化に式 (4) を正則化項として用いて

$$\begin{aligned} f(U, V, W) &= \|\mathcal{X} - \mathcal{G} \times_1 U \times_2 V \times_3 W\|_F^2 + \\ &\quad \frac{\alpha}{2} \text{tr} (U^\top L_1 U + V^\top L_2 V + W^\top L_3 W) \quad (7) \end{aligned}$$

という目的関数を考え、 $U^T U = I, V^T V = I, W^T W = I$  という条件のもとで

$$\min_{\mathfrak{g}, U, V, W} f(U, V, W)$$

を計算すればよい事になる。

#### 4.3.1 $\mathfrak{g}$ についての最適化

CP 分解の場合と同様に式 (7) は全体として凸関数ではないので、ある 1 つの変数についての最適化を繰り返すことで近似解を求める方法を用いる。

まず、 $U, V, W$  を固定した場合に最適なコアテンソル  $\mathfrak{g}$  を求める。

最適となる  $\mathfrak{g}$  の値は

$$\mathfrak{g} = \mathbf{x} \times_1 U^T \times_2 V^T \times_3 W^T \quad (8)$$

となる。

#### 4.3.2 変数ごとの最適化

次に変数  $U, V, W$  についての最適化を行う。最適な  $\mathfrak{g}$  の値である (8) を (7) に代入すると、

$$\begin{aligned} \|\mathbf{x} - \mathfrak{g} \times_1 U \times_2 V \times_3 W\|_F^2 &= \|\mathbf{x}\|_F^2 - \|\mathfrak{g}\|_F^2 \\ &= \|\mathbf{x}\|^2 - \|\mathbf{x} \times_1 U^T \times_2 V^T \times_3 W^T\|^2 \end{aligned}$$

を得る。つまり (7) の最適化問題は、 $U, V, W$  に関係しない項を無視すると、 $U, V, W$  の列がすべて正規直交という条件のもとで、

$$\|\mathbf{x} \times_1 U^T \times_2 V^T \times_3 W^T\|_F^2 + \frac{\alpha}{2} \text{tr}(U^T L_1 U)$$

を最大化することに等しいということになる。

これを解くため、他の変数を固定した状態で  $U$  について最大化することを考えたい。 $U$  が関わらない項を無視し、モード 1 についてテンソルを展開すると、

$$\|U^T \mathbf{X}_{(1)} (W \otimes V)\|_F^2 + \frac{\alpha}{2} \text{tr}(U^T L_1 U)$$

となる。表記の簡単のため  $S = \mathbf{X}_{(1)} (W \otimes V)$  とおくと、

$$\begin{aligned} &= \|U^T S\|^2 - \frac{\alpha}{2} \text{tr}(U^T L_1 U) \\ &= \text{tr}(U^T S S^T A) - \frac{1}{2} \text{tr}(U^T (\alpha L_1) U) \\ &= \text{tr}(U^T (S S^T - \alpha L_1) U) \end{aligned}$$

となる。直交性の制約を満たしながらこれを最大化するためには、 $S S^T - \alpha L_1$  を固有値分解し、その固有ベクトルを固有値の大きい順に  $U$  の列ベクトルとして設定してやればよい。

またこのとき  $W \otimes V$  を直接計算すると、行列の大きさが非常に大きくなってしまい無駄が生じる。そこで

$$\begin{aligned} \mathfrak{s} &= \mathbf{x} \times_2 V \times_3 W \\ S &= \mathfrak{s}_{(1)} \end{aligned}$$

とすれば、 $S = \mathbf{X}_{(1)} (W \otimes V)$  をより効率的に計算することができる。

## 5. 提案手法 B:Product

### 5.1 クロネッカー積を用いた正則化

次に、正則化項として

$$\text{tr}((W \otimes V \otimes U)^T L (W \otimes V \otimes U)) \quad (9)$$

という形のものを考える。ここで

$$L = D_3 \otimes D_2 \otimes D_1 - A_3 \otimes A_2 \otimes A_1 \quad (10)$$

であり、これらの  $A_1, A_2, A_3, D_1, D_2, D_3$  はそれぞれのモードについての類似度行列と、その次数となる対角行列である。

この正則化項を変形すると、

$$\begin{aligned} &\text{tr}((W \otimes V \otimes U)^T L (W \otimes V \otimes U)) \\ &= \text{tr}((W^T D_3 W) \otimes (V^T D_2 V) \otimes (U^T D_1 U) \\ &\quad - (W^T A_3 W) \otimes (V^T A_2 V) \otimes (U^T A_1 U)) \\ &= \text{tr}(W^T D_3 W) \text{tr}(V^T D_2 V) \text{tr}(U^T D_1 U) \\ &\quad - \text{tr}(W^T A_3 W) \text{tr}(V^T A_2 V) \text{tr}(U^T A_1 U) \end{aligned}$$

と書くことができることに注意する。

### 5.2 CP 分解への補助情報の導入

式 (5) の正則化項を (9) であるように変更すると、 $\min_{U, V, W} f(U, V, W)$  として解くべき目的関数は

$$\begin{aligned} f(U, V, W) &= \frac{1}{2} \|\mathbf{x} - \mathfrak{g} \times_1 U \times_2 V \times_3 W\|_F^2 \\ &\quad + \text{tr}((W \otimes V \otimes U)^T L (W \otimes V \otimes U)) \quad (11) \end{aligned}$$

の形になる。

#### 5.2.1 変数ごとの最適化

同様に 1 変数ごとに最適化を繰り返すことで全体の近似解を得るという方針で考える。 $U$  について解くことにすると、(11) をモード 1 で展開し微分することで、最適な  $U$  の値は

$$\begin{aligned} U(W \otimes V)^T (W \otimes V) + (D_{VW} D_1 - A_{VW} A_1) U \\ = \mathbf{X}_{(1)} (W \otimes V) \end{aligned}$$

という Sylvester 方程式の解として得られるということが分かる。ここで  $D_{VW}, A_{VW}$  は

$$\begin{aligned} D_{VW} &= \text{tr}(W^T D_3 W) \text{tr}(V^T D_2 V) \\ A_{VW} &= \text{tr}(W^T A_3 W) \text{tr}(V^T A_2 V) \end{aligned}$$

として与えられるスカラー値である。

### 5.3 Tucker 分解への補助情報の導入

次に、Tucker 分解を同様の正則化によって行うには

$$\begin{aligned} f(\mathfrak{g}, U, V, W) &= \frac{1}{2} \|\mathbf{x} - \mathfrak{g} \times_1 U \times_2 V \times_3 W\|_F^2 \\ &\quad + \text{tr}((W \otimes V \otimes U)^T L (W \otimes V \otimes U)) \quad (12) \end{aligned}$$

を目的関数として

$$\begin{aligned} & \min_{\mathcal{G}, U, V, W} f(\mathcal{G}, U, V, W) \\ \text{Subject to} & \quad U, V, W \text{ are column-wise orthonormal} \end{aligned}$$

という最適化問題を解けばよい。

これまでの場合と同様に 1 変数ごとの最適化を繰り返すことにより近似解を求めることにする。この時  $U$  についての更新は

$$SS^T A - \alpha(D_{VW}D_1 - A_{VW}A_1) \quad (13)$$

を固有値分解することによって行うことができる。つまり、固有ベクトルに対応する固有値の大きい方から並べて  $U$  に代入すればよい。

## 6. 数値実験

この節では、補助情報を導入したテンソル分解の有効性を示すための数値実験を行う。

これまで示したテンソル分解によって未観測要素を持つテンソルを補完し、その精度を検証する。結果としては、低ランク性の仮定に加えて、適切な補助情報を導入することによって、補完の精度を向上させることができる。

### 6.1 EM アルゴリズムによるテンソルの補完

ここでは観測されていない要素を持つ低ランクなテンソルを補完する手法について述べる。

低ランクなテンソルを分解する方法を使えば、Algorithm 1 のように EM アルゴリズムによって未観測要素を含むテンソルを補完することができる。

---

#### Algorithm 1 テンソルの補完 (EM)

---

```

 $\mathcal{X} \leftarrow \mathcal{X}_0$  (初期化する)
while 適当な収束条件が満たされるまで do
   $x_{ijk} \leftarrow \begin{cases} y_{ijk} & (i, j, k) \in \Omega \\ x_{ijk} & (i, j, k) \notin \Omega \end{cases}$ 
   $\mathcal{G}, U, V, W \leftarrow \mathcal{X}$  を分解する
   $\mathcal{X} \leftarrow \mathcal{G} \times_1 U \times_2 V \times_3 W$ 
end while

```

---

ただし本研究においては上記の単純な EM アルゴリズムではなく、1 ステップごとに変数の更新を 1 度だけ行うことで計算を加速する EM-ALS [2], [8] と呼ばれるより高速なアルゴリズムを用いて実験を行なっている。(Algorithm 2)

最初にテンソルを適当な値を初期化する必要があるが、実験的には初期値依存性はほとんどないことが分かった。そこで本研究においては、観測した要素の平均値を全要素の初期値として用いている。

### 6.2 実験方法

EM アルゴリズムによって補完を行うが、その際のテンソル分解には補助情報を用いない従来の方法がまず考えられる。本研究ではそれに加えて、補助情報の導入しながらテンソルを分解する方法を提案手法 A:Sum と提案手法 B:Product の二種類提案した。また、低ランク分解にも CP 分解と Tucker 分解二種類の手法がある。これらの組み合わせで 6 種類の補完方法が考えられるので、6 種類それぞれを用いてテンソルの補完精度

---

#### Algorithm 2 テンソルの補完 (EM-ALS)

---

```

 $\mathcal{X} \leftarrow \mathcal{X}_0$  (初期化)
 $[\mathcal{G}, U, V, W] \leftarrow [\mathcal{G}_0, U_0, V_0, W_0]$  (初期化)
while 適当な収束条件が満たされるまで do
   $x_{ijk} \leftarrow \begin{cases} y_{ijk} & (i, j, k) \in \Omega \\ x_{ijk} & (i, j, k) \notin \Omega \end{cases}$ 
  for  $i = 0$  to  $N$  ( $N$  は小さい数) do
     $U$  を更新
     $V$  を更新
     $W$  を更新
  end for
   $\mathcal{X} \leftarrow \mathcal{G} \times_1 U \times_2 V \times_3 W$ 
end while

```

---

を比較する。また、未観測要素の設定の方法ごとに二種類の実験を行った。

- 実験 1

テンソル  $\mathcal{X}$  のランダムな要素  $x_{ijk}$  を未観測要素とする。

- 実験 2

テンソル  $\mathcal{X}$  のランダムなファイバー  $x_{*jk}$  を未観測要素とする。ここで  $N$  階テンソルのファイバーとは、行列における行ベクトルや列ベクトルに相当するもので、 $N - 1$  個のモードについて添字を固定し、残る一つのモードにそってテンソルを貫くような要素の集合である。

このような隠し方を行うと、補助情報を導入せずにテンソルの低ランク性の仮定のみで補完する従来の手法ではテンソルの要素を決定するための情報が本質的に足りないため、精度が非常に悪化することが考えられる。このような場合に補助情報を導入するとどの様な効果が現れるかも検証する。上記のような方法によって、補完精度を調べるための数値実験を人工データと実データに対して行う。

## 7. 人工データを用いた評価

まず人工的に生成したテンソルで補完の数値実験を行った。

### 7.1 人工データの生成

この研究を評価するために低ランクな人工データを生成するにあたって、関係の情報を自然に導入できる形で作る必要がある。ここでは一次関数によってなめらかな変化をデータに与え、隣あった要素どうしが類似しているとみなして補助情報を導入することにした。

まずランク  $R$  を決め、行列  $U \in \mathbb{R}^{I \times R}$ ,  $V \in \mathbb{R}^{J \times R}$ ,  $W \in \mathbb{R}^{K \times R}$  を次のように設定する。

$$u_{ir} = i\epsilon_r + \epsilon'_r \quad (1 \leq i \leq I, 1 \leq r \leq R)$$

$$v_{jr} = j\zeta_r + \zeta'_r \quad (1 \leq i \leq J, 1 \leq r \leq R)$$

$$w_{kr} = k\eta_r + \eta'_r \quad (1 \leq i \leq K, 1 \leq r \leq R)$$

ここで、 $\{\epsilon_r, \epsilon'_r, \zeta_r, \zeta'_r, \eta_r, \eta'_r\}_{r=1}^R$  はあらかじめ設定した適当な定数であるとする。この実験では独立な標準正規分布に従って生成した値を用いた。

上のように要素を生成した  $U, V, W$  を CP 分解の意味での

基底とみなし、人工データとなるテンソル  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$  を

$$\mathcal{X} = \mathcal{J} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} \quad (14)$$

のように構成する。

隣り合った要素が類似するようにテンソルを作るので、関係による正則化のために

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots \\ 1 & 0 & 1 & \cdots \\ 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (15)$$

という三重対角な類似度行列  $\mathbf{A}$  を作り、これを用いて  $\mathbf{L} = \mathbf{D} - \mathbf{A}$  としてグラフラプシアンを考えることができる。この方法によって、すべてのモードに対してグラフラプシアン  $L_1, L_2, L_3$  を生成する。この実験では、テンソルのランクは  $R = 2$ 、大きさは  $I = J = K = 30$  と固定して計算を行っている。

実験には上記のように人工的に生成したテンソルを用いる。テンソルからランダムに一定の割合の要素を選択し、それらを未観測として扱うことで補完実験を行った。またこの時、未観測とする要素を変更することで交差検定を行った。

補完に用いたランクは Tucker 分解では  $(2, 2, 2)$ 、CP 分解では真の値である 2 に固定し、また補助情報による正則化項の係数  $\alpha$  に対して、 $\alpha = \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$  の中でハイパーパラメータ最適化を行っている。また実験する未観測率は、実験 1 については  $\{0.75, 0.9, 0.95, 0.99\}$  とし、実験 2 については  $\{0.5, 0.75, 0.9, 0.95\}$  とした。

## 7.2 実験結果

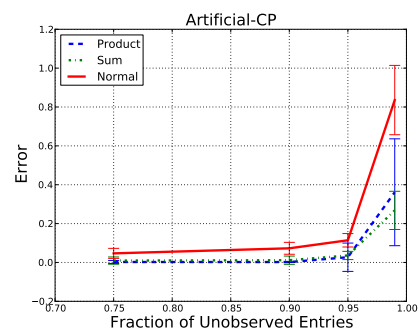
すべての要素をランダムに未観測にする実験 1 について補完精度を示したものが図 1 である。また実験 2 の結果が図 2 である。

どちらの実験でも、補助情報を導入することにより、全体として補完精度が向上していることが分かる。実験 1 の場合、未観測率 0.95 までであれば、補助情報を用いない場合でも安定した精度で補完が来ている。これは補完に用いた人工データが実際に低ランクであり、さらにノイズを含んでいないためであると考えられる。しかし未観測率 0.99 となった場合に、補助情報を使わない手法では急激に精度が悪化しているのに対して、補助情報を導入した手法ではどちらも比較的精度が良い。

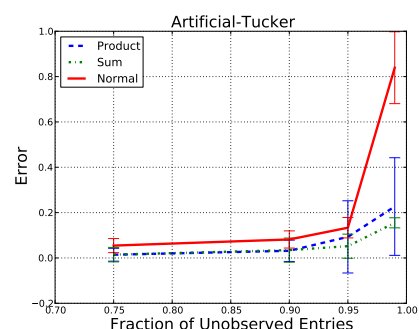
またファイバーごと未観測にする実験 2 では未観測率がそれほど高くない時でも、補助情報なしの手法では高い得られていない。それらの場合においても、補助情報を導入した手法では高い精度を維持していることが確認できる。

## 8. 実データを用いた評価

次に、実データに対してどの程度正確な補完ができるかを検証する。今回は [1] で公開されているデータの中から二種類を選んで実験に使用した。



(c) CP 分解



(d) Tucker 分解

図 1 実験 1:人工データ

## 8.1 使用した実データ

### 8.1.1 データ A: FlowInjection

[1] における “Rank-deficient spectral FIA data” である。組成の異なった 12 種類の化学物質の反応をフローインジェクション分析した観測データであり、12(サンプル数)×100(波長)×89(反応時間) という形の配列になっている。

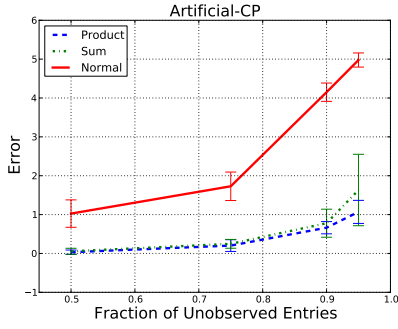
まずサンプル数に関するモードに関してであるが、12 個のサンプルはある化合物の三つの構造異性体<sup>(注1)</sup>の含有量が異なっており、含有量のユークリッド距離の逆数を類似度と考えると補助情報を作った。また波長と反応時間のモードに対して、全体としてはデータは連続的な変化をしていると考えられるので、隣接する要素に対してのみ類似する鎖状のグラフを考えて補助情報とした。上記のように、このデータには三種類のモード全てに補助情報を導入している。

実験では補完のためのランクは Tucker 分解では  $(4, 4, 4)$ 、CP 分解では 4 に固定し、正則化の係数  $\alpha$  は  $\alpha = \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$  の中でクロスバリデーションによってハイパーパラメータ最適化を行った。

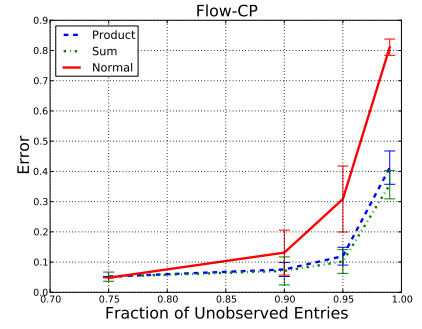
### 8.1.2 データ B: 甘草

[1] における “Three-way electronic nose data” である。甘草の品質を判断するために匂いセンサから得られたデータで、18(サンプル数)×241(反応時間)×12(センサ数) という大きさを持つ三次元配列である。

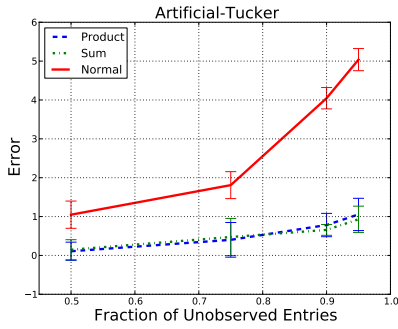
(注1):  $\{2, -3, -4\}$  ヒドロキシベンズアルデヒド



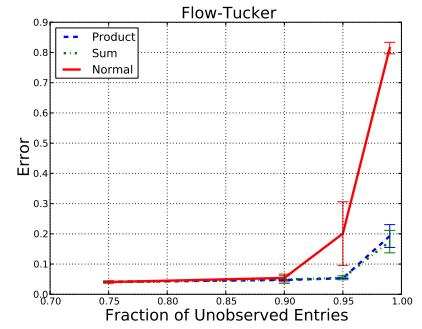
(a) CP 分解



(a) CP 分解



(b) Tucker 分解



(b) Tucker 分解

図 2 実験 2:人工データ

図 3 実験 1: データ A: FlowInjection

反応時間に関しては連続量とみなし、隣接すると類似するとみなした関係情報を導入した。また 18 個のサンプルには品質に関するラベル (“BAD”, “FBAD”, “GOOD”) が付いているので、同じラベルのデータは似ていると考え、それらをつなぐ完全グラフから関係情報を作った。従って、このデータではふたつのモードに対して関係情報を導入する。実験では補完のためのランクは Tucker 分解では (3, 3, 3)、CP 分解では 3 に固定し、 $\alpha = \{10^{-3}, 10^{-2}, 10^{-1}, 10^0\}$  の中でハイパーパラメータ最適化を行った。

## 8.2 実験結果

三種類のデータそれぞれに対して実験 1、実験 2 の二種類の実験の結果を以降に示す。実験 1 の結果を図 3、図 4 実験 2 の結果を図 5、図 6 に示してある。

まず実験 1 では、未観測率を高めていった場合に、精度が悪化するのがどの手法でも人工データと比較して早い。人工データはその作り方からノイズがなく、厳密に低ランクなテンソルであったが、実データはそのような保証はなく、ノイズを含んでいるからである。しかし人工データ同様、どちらの実験でも補助情報を用いた手法が用いない手法よりも精度が向上している事がわかる。テンソルが疎 (未観測率が高い) 場合に特にその差が顕著となっていることも人工データの場合と同様である。

実験 2 でファイバーごと未観測とした場合には、補助情報を用いなければほとんど意味のある補完ができなくなっていることもあるが、補助情報を使った手法では精度の悪化は少ない。

またこの時、全体として提案手法 B:Product が精度が良いことが多いということも分かるが、提案手法 A:Sum の方がよいこともある。データの性質と補助情報の入れ方に左右されることが考えられる。

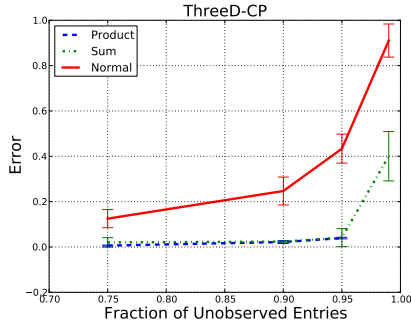
## 9. 結論

本論文では、Li らによる手法 [6] をテンソルに拡張することで、低ランク分解を用いたテンソルの補完にあたってテンソルの低ランク性の仮定に加えて、外部から要素間の関係の情報を用いる手法を提案した。そして人工データと実データでの数値実験によって、適切な補助情報を加えることで補完精度を大きく向上させることができることを示した。

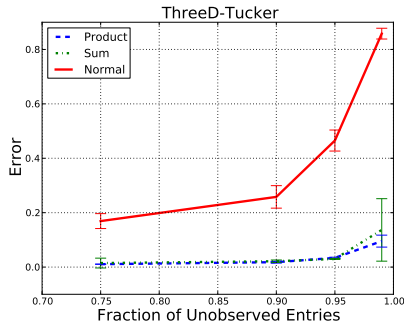
この研究の今後の課題として、使用するメモリ量と計算速度を改善することが挙げられる。テンソルは多次元の配列であるため一般にデータ量が大きく、そのことが計算時間にも影響している。そこで、観測要素が疎である場合に、補助情報であるグラフラプリアンの疎性を生かせるような手法を構築すれば、消費メモリと計算時間と減らすことができると期待される。実際に補助情報のないテンソルの補完では、観測要素の疎性を生かした効率的なアルゴリズムが提案されている [2]。

さらに、テンソルの補完の凸最適化としての定式化に補助情報を導入することが挙げられる。本論文で考えた手法はすべて凸最適化にはなっておらず、逐次的に計算することで実用的に良い解が得られるものの、実際に大域的な最適解が求められる



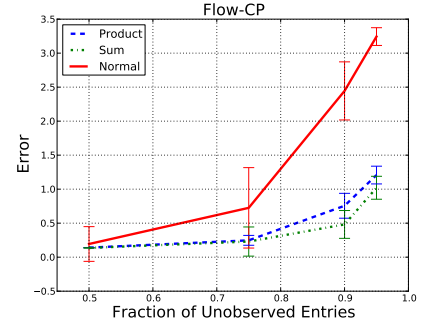


(a) CP 分解

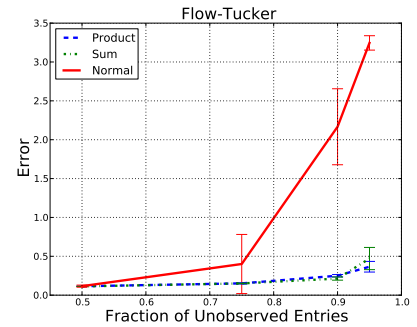


(b) Tucker 分解

図 4 実験 1: データ B: 甘草



(a) CP 分解



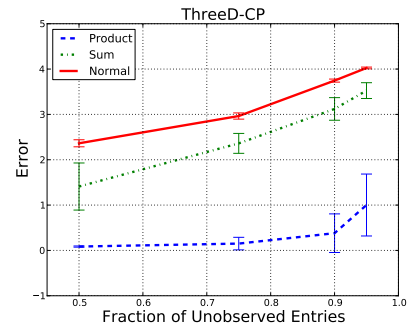
(b) Tucker 分解

図 5 実験 2: データ A: FlowInjection

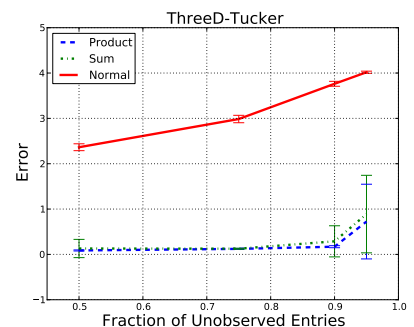
保証はなかった。また補完のためのランクを予め適切に設定する必要があった。トレースノルム最小化といったテンソル補完の凸な定式化 [7] に補助情報を導入し、その効率的なアルゴリズムを作ること、テンソルの補完を更に精度良く高速に求められることが期待される。

### 文献

- [1] Public data sets for multivariate data analysis . <http://www.models.life.ku.dk/datasets>.
- [2] E. Acar, D. Dunlavy, T. Kolda, and M. Mørup. Scalable tensor factorizations with missing data. *Siam Datamining 2010*, pages 701–712.
- [3] F. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.
- [4] T. G. Kolda. Multilinear operators for higher-order decompositions. Technical Report SAND2006-2081, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, April 2006.
- [5] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September 2009.
- [6] W. Li and D. Yeung. Relation regularized matrix factorization. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 1126–1131, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.
- [7] R. Tomioka, K. Hayashi, and H. Kashima. Estimation of low-rank tensors via convex optimization. Technical report, arXiv:1010.0789v2.
- [8] B. Walczak. Dealing with missing data Part I. *Chemo-metrics and Intelligent Laboratory Systems*, 58(1):15–27, September 2001.



(a) CP 分解



(b) Tucker 分解

図 6 実験 2: データ B: 甘草