速　報

マクマホン立方体を用いたテンソルデータ処理に関するデータサイエンス教育の試み

山本　直樹[1,*]　石田　明男[2]　大石　信弘[1]　村上　純[1]

Trial of Data Science Education on Tensor Data Processing Using MacMahon's Cubes

Naoki Yamamoto[1,*], Akio Ishida[2], Nobuhiro Oishi[1], Jun Murakami[1]

We developed a new learning material that can be used in a subject of data science offered in our advanced courses. This material is for students to learn tensor data processing related to HOSVD and reflects important points to learn it. A feature of this material is that MacMahon's cubes are used to make the students interested in the data processing. After using this material for self-study in the lecture, we confirmed degree of understanding of the matrix unfolding, the folding, and the *n*-mode product by a written test. As a result, it was found that about 74% of students who took the subject understood all three processes. From this, it can be considered that this material is effective for data science education of the tensor data processing.

キーワード：　テンソルデータ処理、HOSVD、データサイエンス教育、マクマホン立方体、学習教材開発

Keywords：　Tensor data processing, HOSVD, Data science education, MacMahon's cubes, Development of learning materials

１．はじめに

テンソルは、高次元のデータを取り扱う重要なデータ構造の１つであり、テンソル分解などのテンソルデータ処理を適用してデータの低次元化やパターン認識などに利用される(1)。一般に、テンソルに関するデータ処理は複雑であり、これまで我々は、立体パズルを用いてこれらの処理を理解支援するためのツールを開発してきている(2),(3)。文献(2)では、マクマホン立方体(4)を利用したツールを開発しているが、本研究では、この立方体を用いて本校専攻科開講のデータサイエンスのテンソルデータ処理に関する講義で活用できる新たな学習教材を開発する。以下では、2 章でテンソルデータ処理における学習のポイントについて検討し、3 章では、検討された学習ポイントを反映して開発された学習教材とその活用例の説明、教材利用後の受講学生の理解度確認の結果などについて述べる。

２．テンソルデータ処理とその学習ポイント

2.1 講義を行うテンソルデータ処理の概要

テンソルデータ処理の重要なものとして、テンソル分解がある。テンソル分解とは、高次元のテンソルデータを、より低次元のテンソルを用いて積や和で表現するものである。テンソル分解の１つとして、高次特異値分解（Higher-Order Singular Value Decomposition; HOSVD）(5)があるが、本校専攻科開講のデータサイエンスの講義ではこの分解について取り扱っている。ここで、HOSVD とは、行列の特異値分解（Singular Value Decomposition; SVD）を 3 階以上のテンソルの分解に拡張したものである。以下に、N 階テンソルの HOSVD の定義とそのアルゴリズムを示す(5)。

［(定義 1) N 階テンソルの HOSVD］

サイズ$I_1 \times I_2 \times \cdots \times I_N$の実数値をもつ N 階テンソルを$\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$とする。このとき、テンソル$\boldsymbol{\mathcal{A}}$は N 個のサイズ$I_n \times I_n$の正規直交行列$\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n \times I_n}, (n = 1, 2, \dots, N)$と、$\boldsymbol{\mathcal{A}}$ と同じサイズのコアテンソル$\boldsymbol{\mathcal{C}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$との n-モード積（2.2 節の定義 2 参照）により次式として分解され、これを N 階テンソルの HOSVD と呼ぶ。

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{C}} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \cdots \times_N \boldsymbol{U}^{(N)} \qquad (1)$$

ここで、演算子$\times_n, (n = 1, 2, \dots, N)$は n-モード積の演算を表す。

（定義おわり）

［(アルゴリズム 1) N 階テンソルの HOSVD］

入力：サイズ$I_1 \times I_2 \times \cdots \times I_N$の N 階テンソル$\boldsymbol{\mathcal{A}}$

出力：サイズ$I_n \times I_n$の正規直交行列$\boldsymbol{U}^{(n)}, (n = 1, 2, \dots, N)$、$\boldsymbol{\mathcal{A}}$ と同サイズのコアテンソル$\boldsymbol{\mathcal{C}}$

（ステップ 1）入力の N 階テンソル$\boldsymbol{\mathcal{A}}$に n-モード行列展開（2.2 節の定義 3 参照）を適用して、$\boldsymbol{\mathcal{A}}$をサイズ$I_n \times (I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-2} I_{n-1})$の N 通りの行列$\boldsymbol{A}_{(n)} \in$

---

1 電子情報システム工学系
〒861-1102　熊本県合志市須屋 2659-2
Faculty of Electronics and Information Systems Engineering,
2659-2 Suya, Koshi-shi, Kumamoto, Japan 861-1102

2 リベラルアーツ系
〒861-1102　熊本県合志市須屋 2659-2
Faculty of Liberal Arts,
2659-2 Suya, Koshi-shi, Kumamoto, Japan 861-1102

* Corresponding author:
E-mail address: naoki@kumamoto-nct.ac.jp (N. Yamamoto).

$\mathbb{R}^{I_n \times I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-2} I_{n-1}}, (n = 1,2,\dots,N)$に展開する。

（ステップ 2）ステップ 1 で得られた$\boldsymbol{A}_{(n)}$にそれぞれ特異値分解を適用して、行列$\boldsymbol{A}_{(n)}$を次式のように分解する。

$$\boldsymbol{A}_{(n)} = \boldsymbol{U}^{(n)}\, \boldsymbol{\Sigma}^{(n)}\, {\boldsymbol{V}^{(n)}}^{\mathrm{T}}, (n = 1,2,\dots,N) \qquad (2)$$

ここで、$\boldsymbol{U}^{(n)}$および$\boldsymbol{V}^{(n)}$は、それぞれ、左特異行列および右特異行列を表し、$\boldsymbol{\Sigma}^{(n)}$は対角要素に特異値をもつ対角行列である。また、演算子Tは行列の転置を表す。

（ステップ 3）$\boldsymbol{\mathcal{A}}$とステップ 2 で得られた$\boldsymbol{U}^{(n)}$との n-モード積により、次式で与えられるコアテンソル$\boldsymbol{\mathcal{C}}$を計算する。

$$\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}} \times_1 {\boldsymbol{U}^{(1)}}^{\mathrm{T}} \times_2 {\boldsymbol{U}^{(2)}}^{\mathrm{T}} \cdots \times_N {\boldsymbol{U}^{(N)}}^{\mathrm{T}} \qquad (3)$$

（ステップ 4）正規直交行列$\boldsymbol{U}^{(n)}, (n = 1,2,\dots,N)$とコアテンソル$\boldsymbol{\mathcal{C}}$を返す。

（アルゴリズムおわり）

2.2 HOSVD の学習に重要なポイント

2.1 節で述べた定義およびアルゴリズムから、我々は HOSVD の学習に重要なポイントは、以下のものがあると考えている。

（ポイント 1）入力となるテンソルデータの表現
（ポイント 2）n-モード行列展開の計算
（ポイント 3）特異値分解の計算
（ポイント 4）n-モード積の計算

さらに、学習のポイントを探るために、（ポイント 4）の計算について詳しく見ていく。n-モードとはテンソルの n 番目の添字のことで、n-モード積は、この添字に関するテンソルと行列との積の演算であり、次にその定義を示す[5]。

［（定義 2）n-モード積］

サイズ$I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_N$の実数値をもつ N 階テンソルを$\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_N}$とする。このとき、$\boldsymbol{\mathcal{A}}$とサイズ$J_n \times I_n$の行列$\boldsymbol{U}^{(n)} \in \mathbb{R}^{J_n \times I_n}$との n-モード積は次式で定義される。

$$\left(\boldsymbol{\mathcal{A}} \times_n \boldsymbol{U}^{(n)}\right)_{i_1 i_2 \cdots j_n \cdots i_N} = \sum_{i_n=1}^{I_n} a_{i_1 i_2 \cdots i_n \cdots i_N} u_{j_n i_n},$$

$$\begin{pmatrix} i_1 = 1,2,\cdots,I_1; i_2 = 1,2,\cdots,I_2; \cdots; i_n = 1,2,\cdots,I_n; \\ \cdots;\ i_N = 1,2,\cdots,I_N; j_n = 1,2,\cdots,J_n \end{pmatrix} \qquad (4)$$

ここで、$\left(\boldsymbol{\mathcal{A}} \times_n \boldsymbol{U}^{(n)}\right)_{i_1 i_2 \cdots j_n \cdots i_N}$はサイズ$I_1 \times I_2 \times \cdots \times J_n \times \cdots \times I_N$をもつ N 階テンソル$\boldsymbol{\mathcal{A}} \times_n \boldsymbol{U}^{(n)} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times J_n \times \cdots \times I_N}$の$(i_1, i_2, \cdots, j_n, \cdots, i_N)$要素、$a_{i_1 i_2 \cdots i_n \cdots i_N}$は$\boldsymbol{\mathcal{A}}$の$(i_1, i_2, \cdots, i_n, \cdots, i_N)$要素、$u_{j_n i_n}$は$\boldsymbol{U}^{(n)}$の$(j_n, i_n)$要素を表す。

（定義おわり）

以上が、n-モード積の定義であるが、一般に、テンソルの演算はこのように複雑なものとなる。定義 2 に従えば n-モード積の演算はできるが、講義では、この演算の処理を受講学生に分かりやすく捉えてもらうために、以下に示すアルゴリズムを利用した[1]。

［（アルゴリズム 2）n-モード積］

入力：　サイズ$I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_N$の N 階テンソル$\boldsymbol{\mathcal{A}}$、サイズ$J_n \times I_n$の行列$\boldsymbol{U}^{(n)}$

出力：　サイズ$I_1 \times I_2 \times \cdots \times J_n \times \cdots \times I_N$の N 階テンソル$\boldsymbol{\mathcal{A}} \times_n \boldsymbol{U}^{(n)}$

（ステップ 1）$\boldsymbol{\mathcal{A}}$の n-モードに関して行列展開を適用し、$\boldsymbol{\mathcal{A}}$をサイズ$I_n \times (I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-2} I_{n-1})$の行列$\boldsymbol{A}_{(n)} \in \mathbb{R}^{I_n \times I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-2} I_{n-1}}$に展開する。

（ステップ 2）$\boldsymbol{U}^{(n)}$とステップ 1 で得られた$\boldsymbol{A}_{(n)}$の行列積$\boldsymbol{U}^{(n)} \boldsymbol{A}_{(n)} \in \mathbb{R}^{J_n \times I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-2} I_{n-1}}$を計算する。

（ステップ 3）ステップ 2 で得られた行列$\boldsymbol{U}^{(n)} \boldsymbol{A}_{(n)}$を、$n$-モードに関して N 階テンソルへ畳み込んで$\boldsymbol{\mathcal{A}} \times_n \boldsymbol{U}^{(n)}$を得る。

（ステップ 4）N 階テンソル$\boldsymbol{\mathcal{A}} \times_n \boldsymbol{U}^{(n)}$を返す。

（アルゴリズムおわり）

n-モード積の計算にアルゴリズム 2 を利用することにより、（ポイント 4）の計算はさらに以下の点を考慮する必要があると考えられる。

（ポイント 4-a）行列積の計算
（ポイント 4-b）テンソルへの畳み込みの処理
（ポイント 4-c）n-モード積の性質

ここで、（ポイント 4-b）の畳み込みは、（ポイント 2）の行列展開の逆の操作となる。以下に n-モード行列展開と畳み込みの定義を示す[5]。

［（定義 3）n-モード行列展開］

サイズ$I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_N$の実数値をもつ N 階テンソルを$\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_N}$とする。ここで、$N$ 階テンソル$\boldsymbol{\mathcal{A}}$ の $(i_1, i_2, \cdots, i_n, \cdots, i_N)$ 要素を行列 $\boldsymbol{A}_{(n)} \in \mathbb{R}^{I_n \times I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-2} I_{n-1}}, (n = 1,2,\dots,N)$の$(i_n, j_n)$要素に展開することを、$\boldsymbol{\mathcal{A}}$の n-モード行列展開と呼ぶ。ただし、j_nは次式で与えられる。

$$\begin{aligned} j_n = &(i_{n+1} - 1) I_{n+2} \dots I_N I_1 \dots I_{n-2} I_{n-1} \\ &+ (i_{n+2} - 1) I_{n+3} \dots I_N I_1 \dots I_{n-2} I_{n-1} + \cdots \\ &+ (i_N - 1) I_1 \dots I_{n-2} I_{n-1} + (i_1 - 1) I_2 \dots I_{n-2} I_{n-1} + \cdots \\ &+ (i_{n-2} - 1) I_{n-1} + i_{n-1} \end{aligned} \qquad (5)$$

（定義おわり）

［（定義 4）畳み込み］

定義 3 における n-モード行列展開$\boldsymbol{A}_{(n)} \in \mathbb{R}^{I_n \times I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-2} I_{n-1}}, (n = 1,2,\dots,N)$の$(i_n, j_n)$要素を、$N$ 階テンソル$\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_N}$の$(i_1, i_2, \cdots, i_n, \cdots, i_N)$要素に戻すことを、$\boldsymbol{A}_{(n)}$の$\boldsymbol{\mathcal{A}}$への畳み込みと呼ぶ。ただし、$j_n$は式(5)で与えられる。

（定義おわり）

また、n-モード積の演算を行う上で、（ポイント 4-c）の性質は重要となる。以下にこの性質を示す[5]。

［（性質 1）n-モード積］

N 階テンソルを$\boldsymbol{\mathcal{A}}$、２つの行列を$\boldsymbol{U}^{(o)}, \boldsymbol{U}^{(p)}$とし、$o \neq p$のとき、$o$-モード積と p-モード積の演算に関して次が成立する。

$$\begin{aligned} \boldsymbol{\mathcal{A}} \times_o \boldsymbol{U}^{(o)} \times_p \boldsymbol{U}^{(p)} &= \left(\boldsymbol{\mathcal{A}} \times_o \boldsymbol{U}^{(o)}\right) \times_p \boldsymbol{U}^{(p)} \\ &= \left(\boldsymbol{\mathcal{A}} \times_p \boldsymbol{U}^{(p)}\right) \times_o \boldsymbol{U}^{(o)} \end{aligned} \qquad (6)$$

（性質おわり）

以上では、HOSVD の学習に重要なポイントについて検

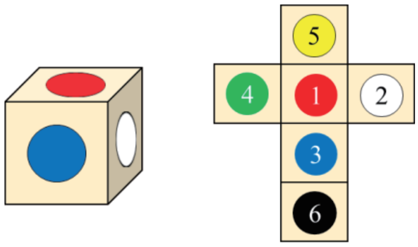

図1　マクマホン立方体

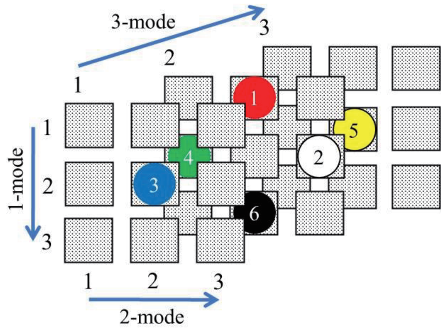

図2　テンソルデータ表現

討した。講義においては、（ポイント1）のテンソルデータ表現、（ポイント2）の行列展開、（ポイント4）の n-モード積、（ポイント4-b）の畳み込み、（ポイント4-c）の性質の学習に焦点を当てた。他に（ポイント3）の特異値分解と（ポイント4-a）の行列積があるが、これらの内容については、受講学生は以前にデータサイエンスや線形代数の講義で学習して予備知識があるものと考えられる。このことから、これら2つのポイント以外に焦点を当てることとした。

3．開発した学習教材と講義での活用

3.1 マクマホン立方体とそのテンソルデータ表現

2.2節で述べた（ポイント1）のテンソルデータ表現の題材として、受講学生にテンソルデータ処理により興味をもってもらう観点から、図1に示すマクマホン立方体[4]を取り上げた。この立方体は、各面に異なる色が塗られているもので、ここで用いた色は、赤(1)、白(2)、青(3)、緑(4)、黄(5)、黒(6)の6色であり、() 内の数字は各色の番号を表す。

図2はこの立方体のテンソルデータ表現を示すが、図1の立方体をサイズ3×3×3の3階テンソル $\mathcal{A} \in \mathbb{R}^{3\times3\times3}$ として、$\mathcal{A}$ の各面の中央の要素には、元の立方体に塗られた色の番号を格納し、それ以外の要素はゼロとしたものである。

3.2 開発した学習教材例

(1)（ポイント2）の行列展開に関する教材

図3に2-モード行列展開に関する問題と解答例を示す。この問題は、R の行列展開を行う関数 unfold[6]に渡された引数から判断して、図3右下の行列展開のセル内に展開される要素値と、行および列に展開されるモードがどのようになるかを学習するものである。これに関する同様な教材として、1-モードと3-モード行列展開の問題も作成されている。

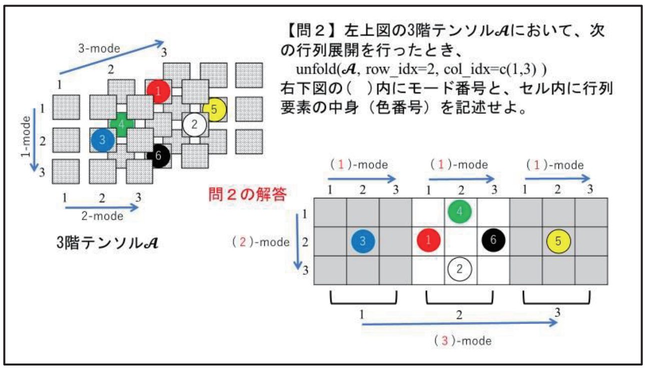

図3　行列展開の問題例と解答例

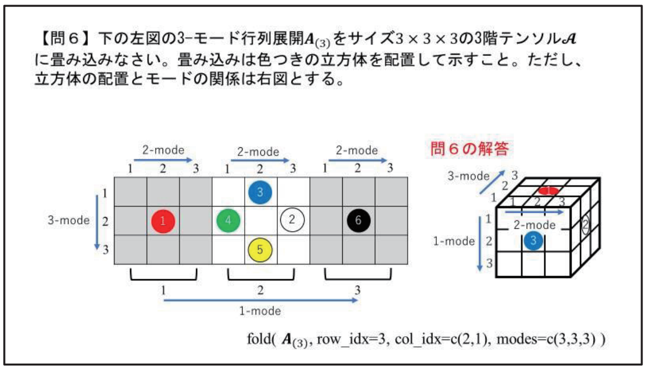

図4　畳み込みの問題例と解答例

(2)（ポイント4-b）の畳み込みに関する教材

図4に畳み込みに関する問題と解答例を示す。ここでは、図4左に示される3-モード行列展開の行列を、図4右の3階テンソル $\mathcal{A}$ に畳み込んで、$\mathcal{A}$ の見える3面のいずれの要素にどの色が格納されるのかを学習するものである。この他の問題として、1-モードと2-モード行列展開の行列を畳み込む問題も作成されている。

(3)（ポイント4）と（ポイント4-c）の n-モード積に関する教材

図5および図6は、それぞれ1-モード積に関する問題および解答例を示す。この教材は、2.2節のアルゴリズム2で述べた n-モード積の計算過程を学習するものである。この他の問題として、2-モード積と3-モード積に関するものも作成されている。図7と図8は、１つのテンソルと３つの行列の n-モード積の演算について学習するものである。テンソルの各要素にどの色が格納されるか、R の n-モード積を行う関数 ttm[6]を用いて記述すると n-モード積の演算を入れ子にして計算できることが分かる。

3.3 講義における本教材の活用と理解度確認の結果

開発された教材は、本校専攻科1年次開講のデータサイエンスのテンソルデータ処理に関する講義で活用された。本講義では、まず、テンソルの概要、R のテンソル演算用パッケージ rTensor[6]の説明、行列展開と n-モード積の処理について R での実装例を示しながら説明を行った。その後で、本教材を受講学生に自学課題として提示し、実施してもらった。学生には、本教材の各問題について R で実行し、各解答を確認する形で利用するように指示した。

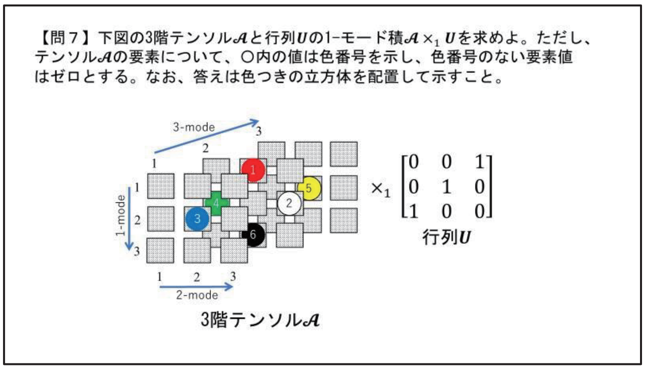

図５　1-モード積の問題例

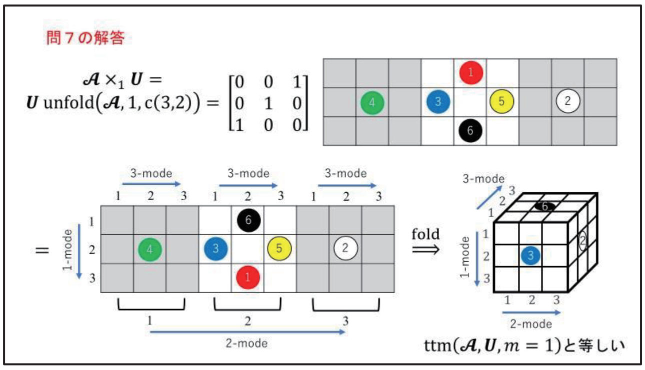

図６　1-モード積の解答例

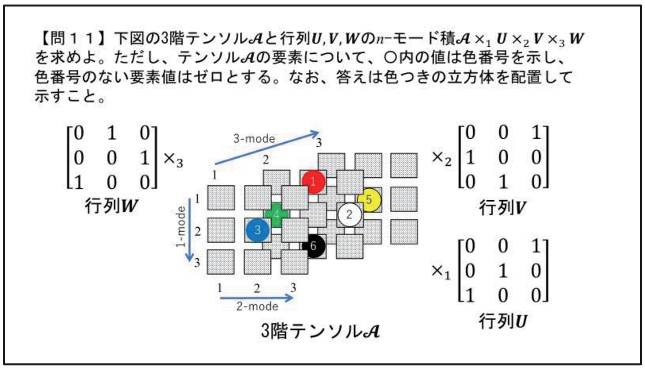

図７　複数の行列を用いた n-モード積の問題例

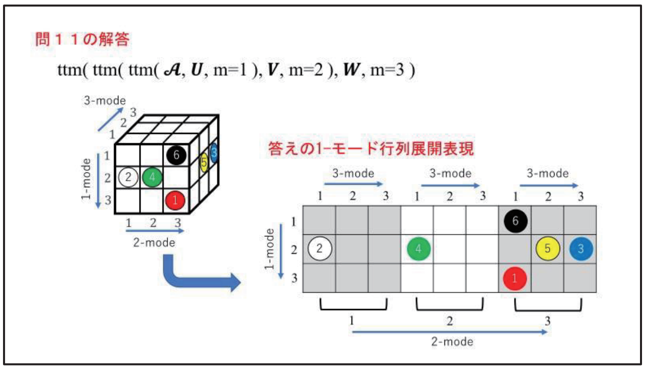

図８　複数の行列を用いた n-モード積の解答例

本教材を自学後に、行列展開、畳み込み、n-モード積について受講学生が理解できているかを確認するために、本教材の問題の中から出題された筆記テストを行った。その結果、行列展開に関しては受講学生全体の約 89%、畳み込みは全体の約 96%、n-モード積は全体の約 82%が全問正解しており、これら３つの処理に関して全問正解したのは全体の約 74%であった。さらに、HOSVD についての講義後に、2.1 節で述べたアルゴリズム 1 を R で実装するレポート課題を受講学生に与えて実施してもらったところ、受講学生全体の約 93%が HOSVD を実装できていた。

４．まとめ

本論文では、テンソルデータ処理のうち、特に HOSVD のテンソル分解に関する学習に重要なポイントについて検討した。その結果、特に重要なものをまとめると以下の通りとなった。

（ポイント 1）テンソルデータの表現
（ポイント 2）n-モード行列展開の計算
（ポイント 4）n-モード積の計算
（ポイント 4-b）畳み込みの処理
（ポイント 4-c）n-モード積の性質

これらの学習ポイントを考慮して、受講学生にテンソルデータ処理に興味をもってもらうために、マクマホン立方体を題材として、これら学習ポイントを自学するための学習教材を開発した。開発した教材は、専攻科開講のデータサイエンスの講義で活用した。

本教材の利用後、受講学生が行列展開、n-モード積、畳み込みについて理解できているかを確認したところ、n-モード積が比較的難しかったが受講学生全体の約 82%が理解しており、３つの処理全てを理解していたのは全体の約 74%であることが分かった。さらに、受講学生全体の約 93%が HOSVD アルゴリズムの実装ができていた。これらのことから、本教材はテンソルデータ処理を学習させるデータサイエンス教育に有用となるものと考えている。

（令和 2 年 9 月 25 日受付）
（令和 2 年 12 月 7 日受理）

参考文献

(1) Lars Eldén : “Matrix Methods in Data Mining and Pattern Recognition”, SIAM, ch. 8 and ch.14 (2007).
(2) N. Yamamoto, A. Ishida, N. Oishi, and J. Murakami : “Development of Teaching Tool for Supporting Understanding of Tensor Decomposition Using MacMahon’s Coloured Cubes”, International Journal of Information and Education Technology, Vol.10, No.1, pp.14-19 (2020).
(3) 山本直樹，村上純，石田明男：｢テンソル分解プログラミングの理解支援のための立体パズルの利用」，SS2019 論文集，pp.114-123 (2019).
(4) J. Köller : MacMahon's Coloured Cubes, http://www.mathematische-basteleien.de/macmahon.htm (Retrieved Aug. 21, 2020).
(5) L. De Lathauwer, B. De Moor, and J. Vandewalle : “A Multilinear Singular Value Decomposition”, SIAM Journal on Matrix Analysis and Applications, Vol.21, No.4, pp.1253-1278 (2000).
(6) J. Li, J. Bien, and M. Wells : Package ‘rTensor’, https://cran.r-project.org/web/packages/rTensor/rTensor.pdf (Retrieved Aug. 21, 2020).