

# 博士学位論文

テンソル自己組織化写像

平成 28 年 6 月

九州工業大学大学院生命体工学研究科

岩崎 亘



## 論文要旨

本論文は関係データをモデル化し可視化するための解析技術を提案するものである。関係データとは複数要素の組み合わせから観測されるデータであり、たとえばオンラインショップの商品評価データはユーザと商品の組み合わせ（ユーザ × 商品）から得られる関係データである。近年ビッグデータが注目される中で関係データ解析の必要性が高まっており、たとえば電子メールデータ（送信者 × 受信者 × トピック）や脳波データ（チャネル × 周波数 × 時間）、顔画像データ（人物 × 表情 × カメラアングル × 照明）などはすべて関係データとして表される。本論文が提案するテンソル自己組織化写像 (Tensor SOM: TSOM) は、これらの関係データを非線形テンソル分解することを通して、データに内在する本質を可視化する手法である。

第一章では序論として、本研究の背景と目的および本論文の構成について述べる。

第二章では本研究の基礎となる概念や手法について述べる。まずテンソルの表記法や演算について説明し、テンソルの基本的解析手法である線形テンソル分解について概説する。次に TSOM の基盤となる自己組織化写像 (Self-Organizing Map: SOM) について述べ、さらに SOM を線形基底関数モデルで表現する拡張について説明する。またベイズ推論の立場から SOM を理論的に再構築した生成位相写像 (Generative Topographic Mapping: GTM) についても説明する。

第三章では本論文の主題である TSOM について提案する。まず TSOM が解くべき問題設定について説明したのち、TSOM のアルゴリズムについて述べる。また TSOM アルゴリズムの理論的導出も行う。さらに派生形として基底関数型 TSOM およびテンソル生成位相写像 (Tensor GTM: TGTM) についても述べる。

第四章では人工データを用いて TSOM の動作を検証する。このシミュレーションにより TSOM が期待する動作を行うことを示した。さらに計算速度と計算精度についても評価を行い、基底関数型 TSOM を用いれば大規模データであっても高速に計算できることを示した。

第五章では TSOM を利用したさまざまな可視化法を提案する。さらに実データについて応用し、TSOM の実用性を示した。このような多様な可視化法が用いられることも TSOM の長所である。本論文では寿司および飲料の嗜好調査データに対して TSOM を用いてユーザと商品の同時解析を行い、多くの情報が引き出せることを示した。

実データにおいて関係データは欠損値を伴っていたり、付随するデータをもっていたりする。第六章ではこのようなデータに対する TSOM の拡張について述べる。まず欠損値のある関係データに対し TSOM のアルゴリズムを拡張する。これは推薦システムの基盤技術として重要である。次に属性情報とよばれる付加情報が付随する関係データへの TSOM 拡張について述べる。これらの拡張版 TSOM は映画評価データに適用し、その有効性を示した。また電子メールや SNS などのコミュニケーションデータに適した TSOM の拡張を示し、実データに応用した。

第七章では討論として、まずテンソル解析と SOM 双方に関わりのある他手法である高階自己組織化写像 ( $SOM^2$ ) と本手法の比較を行う。このことを通して、TSOM や  $SOM^2$  を包括する理論体系を構築することを試みる。これにより TSOM および  $SOM^2$  はより大きな枠組みのもとで一般化可能なアルゴリズムファミリーを作る可能性を示した。また、本手法の基盤である SOM との関係

についても議論を行い, SOM の一般化という枠組みで TSOM の意義を述べる。

第八章では総括として本論文をまとめる。

以上, 本論文では関係データを非線形テンソル分解によりモデル化し可視化する手法として TSOM を提案した. 本手法は線形テンソル解析にくらべて柔軟な表現能力をもっており, また多様な可視化法を利用することで多くの情報を引き出すことができる. さらに複雑な構造をもつ大規模データや欠損をもつ不完全データにも応用できるため, 幅広い分野での応用が見込める. したがって本研究は知的情報処理を発展させていくうえで重要な基盤技術である.

# 目次

論文要旨	i
<b>第 1 章 序論</b>	1
1.1 本研究の背景・目的 . . . . .	1
1.2 本論文の構成 . . . . .	5
<b>第 2 章 基礎知識</b>	7
2.1 テンソルとテンソル解析手法 . . . . .	7
2.2 自己組織化マップ (Self-Organizing Map: SOM) . . . . .	12
<b>第 3 章 Tensor SOM: TSOM</b>	19
3.1 TSOM-R <sup>2</sup> . . . . .	19
3.2 TSOM-R <sup>M</sup> . . . . .	23
3.3 TSOM-R <sup>M</sup> のアルゴリズム導出 . . . . .	26
3.4 基底関数型 TSOM-R <sup>M</sup> . . . . .	29
3.5 初期化とスケジューリング . . . . .	33
3.6 Tensor GTM: TGTM . . . . .	33
<b>第 4 章 人工データを用いた TSOM の検証</b>	37
4.1 TSOM および TGTM の性能比較 . . . . .	37
4.2 TSOM の動作検証 . . . . .	40
<b>第 5 章 TSOM によるアンケートデータ解析</b>	43
5.1 寿司に関するアンケートデータの解析 . . . . .	43
5.2 飲料に関するアンケートデータの解析 . . . . .	46

---

第 6 章	TSOM のバリエーション	51
6.1	欠損データ推定 . . . . .	51
6.2	属性情報の活用 . . . . .	54
6.3	正方関係データへの応用 . . . . .	55
6.4	まとめ . . . . .	57
第 7 章	討論	59
7.1	TSOM ファミリー . . . . .	59
7.2	TSOM と SOM の関係 . . . . .	66
第 8 章	総括	69
参考文献		71
研究業績リスト		75
謝辞		79

# 第1章

## 序論

### 1.1 本研究の背景・目的

#### 1.1.1 ビッグデータとその特性

近年、大規模で複雑なデータが工学、医学、農学、言語学、気象学などのあらゆる分野で日々蓄積されている。また人々の日常生活においても、パソコンや携帯電話を介したインターネット上での行動履歴や、飲食店や小売店での購入履歴など、多くの活動情報は記録・保存されるようになってきた。これらのデータはセンサー技術の向上や情報化の発達、記録媒体の大容量化に伴って急速に増加している。このようなデータはビッグデータとよばれ、その解析手法の開発が強く求められている。

ビッグデータは、大容量 (Volume), 多様性 (Variety), 高頻度・高速度 (Velocity) という 3 つの特性をもつデータである [1]。ビッグデータを解析する手法を開発する際には、通常のデータ解析手法に対しこれらの特性にあった拡張や改良を行うこととなる。

まず大容量 (Volume) という特性は、扱うデータセットのサイズが非常に大きいことを示しており、ビッグデータの基本となる特性である。解析手法としては、データ量の増加に伴う処理時間の増加を抑える必要がある。すなわちスケーラビリティを向上させる必要がある。具体的には同一計算機内の並列処理や複数の計算機での分散処理ができるよう、アルゴリズム内の計算の依存関係を減らしてそれぞれ独立に算出できるように改良すればよい。

次に、高頻度・高速度 (Velocity) とは、扱うデータの蓄積されていく速度が非常に速く、かつリアルタイムに生成され続けていることを指す。データは絶え間なく生成され続けるため、ストックした後に全データをまとめて解析するというアプローチではなく、流

れてくるデータ（いわゆるストリームデータ）をリアルタイムに解析する必要がある。改良方法としてはデータをまとめて一括に解析するバッチ処理型のアルゴリズムから、オンライン処理・リアルタイム処理型アルゴリズムのようにデータをひとつずつ取り出して逐次処理するか、あるいはミニバッチ処理・マイクロバッチ処理型アルゴリズムのように少量のデータセットを取り出して逐次処理するようアルゴリズムに変更を加えるのが妥当である。

最後に多様性 (Variety) という特性では、扱うデータの構造が单一の数値データベースで表現できる構造に限らないということを表している。ビッグデータは整形された構造化データに加え、複数のセンサーから得られるデータや動画像・文章などの複雑な構造をもつデータを解析の対象とする。ビッグデータの解析手法を考案する際には、取り扱うデータの形式を定める必要があり、このことに大きくかかわるのがこの多様性 (Variety) という特性である。大容量 (Volume), 高頻度・高速度 (Velocity) という 2 つの特性についてはアルゴリズムの改良は必要であるものの、あくまで解析結果を得るまでのアプローチが異なるだけであり、单一のデータの構成も解析結果として得られるものも従来のデータ解析から大きく変化していない。

### 1.1.2 関係データとテンソル

本研究ではこの多様性 (Variety) について、ビッグデータの多くが複数の要素の関係により構成されていることに着目した。例えばオンラインショップの行動履歴はユーザと商品の関係（誰が何を購入したかなど）で表される。このようなデータは一般に関係データと呼ばれている [2]。関係データの用語として、関係を構成する要素数を次数、それぞれの要素のことをモードと呼ぶ。オンラインショップの例では、第 1 モードがユーザ、第 2 モードが商品の次数 2 の関係データである。本論文では簡単のため、このようなデータを (ユーザ)  $\times$  (商品) の関係データあるいは 2 モード関係データと呼ぶこととする。他にも SNS や電子メール上でのコミュニケーション解析は (送信者)  $\times$  (単語)  $\times$  (時間) の 3 モード関係データ [3, 4, 5, 6]、脳波データは (電極位置)  $\times$  (時間)  $\times$  (周波数) の 3 モード関係データ [7, 8]、顔画像データは (人物)  $\times$  (表情)  $\times$  (カメラアングル)  $\times$  (照明) の 4 モード関係データなどで表現できる [9, 10, 11, 12]。このように関係データは多分野で広く取り扱われており、その解析手法の開発は重要である。なお、関係データの成分はしばしばスカラーあるいは 2 値を定義域とするが、本論文中においては  $D$  次元の実数ベクトル  $\mathbb{R}^D$  として考える。

関係データはしばしばテンソルで表現される。テンソルとは行列を一般化したものであり、多次元配列により表される。テンソルを解析する代表的手法はテンソル因子化あるいはテンソル分解と呼ばれるグループを作る [13, 14, 15]。テンソルおよびテンソル解析手法についても次章で改めて説明を行うが、ここで手法について軽く触れる。テンソル分解は行列分解の一般化であり、与えられたテンソルを低次の行列やテンソルの積に分解する。Tucker 分解や CP 分解は代表的なテンソル分解手法である。また特異値分解 (Singular Value Decomposition: SVD) [16]、主成分分析 (Principal Component Analysis: PCA) [17, 18]、独立成分分析 (Independent Component Analysis: ICA) [19, 20]、非負値行列因子分解 (Nonnegative Matrix Factorization: NMF) [21] はすべてテンソルデータが扱えるように拡張されてきた。これらのようにテンソル分解手法は行列分解手法をベースに拡張することで得られる。すなわち関係データ解析手法を開発するとき、関係データから得たい情報に合わせて行列分解手法を選択し、これをベースに拡張することとなる。

### 1.1.3 本研究の目的

本論文では関係データに内在する潜在的性質を可視化するために、自己組織化写像 (Self-Organizing Map: SOM)[22, 23, 24, 25] についてテンソルデータが扱えるように拡張する。これを Tensor SOM (TSOM) とよぶ。

SOM は高次元データの可視化に優れた次元削減法である。SOM は与えられた高次元データ (観測データ) に対し、より低次の内在的な性質を表現する空間 (潜在空間) のデータ (潜在変数) と、潜在空間から観測空間への滑らかな非線形写像によって生成されたと仮定する。そして個々のデータに対応する潜在変数と非線形写像を推定するアルゴリズムである。得られた潜在変数の配置や非線形写像の情報から、観測空間上での現象を可視化することができる。SOM の詳細については次章で述べる。なおベースとなる手法に SOM を用いた理由を以下にまとめる。

- 潜在空間から観測空間への写像が非線形であり、PCA などの類似した線形手法と比べて写像の表現能力が高い。
- 潜在空間から観測空間への写像を直接求めているため、カーネル法を利用した非線形手法 (Kernel PCA など) よりも可視化法が充実している<sup>\*1</sup>。

---

<sup>\*1</sup> ただし SOM は可視化のための低次の潜在空間を仮定しており、写像を表現する上で強い拘束条件となっている。カーネル法では通常低次の潜在空間を仮定しないため、一般にカーネル法を用いた手法のほうが写像の表現能力は高い。

- アルゴリズムがシンプルであり並列化できる箇所が多く、さらに逆行列や固有値などの導出を必要としないため、ビッグデータの Volume の特性に適応した手法である。
- SOM はもともとオンライン型のアルゴリズムであり、ビッグデータの Velocity の特性に対するアプローチが明確である<sup>\*2</sup>。
- SOM は歴史的にはニューラルネットワークの一種であり、この問題の枠組みを確率的生成モデルから再導出した生成位相写像 (Generative Topographic Mapping: GTM)[26, 27] という手法がある。これにより論理的な基盤も併せ持つことが可能となる。なお TSOM にもこの基盤をもたせるため本論文では Tensor GTM (TGTM) もあわせて提案する。

以上を踏まえ、本論文では関係データを解析し可視化するための技術開発を目的とする。具体的には非線形テンソル解析手法として SOM と GTM をテンソル拡張したものを開発することが本研究の目的である。TSOM/TGTM の典型的な応用として、ユーザと商品の 2 モード関係データの解析が挙げられる。ここで必要となる解析は、ユーザ集合の解析、商品集合の解析、およびユーザと商品の関係についての解析である。すなわち関係データのモード内解析とモード間解析モードを実現する必要がある。ここではモード内解析は、各モードごとに潜在空間を設けることで行う（なお SOM の用語として潜在空間のことをマップとも呼ぶことがある。）ユーザと商品の 2 モード関係データの場合、TSOM/TGTM ではユーザマップと商品マップをそれぞれ作成する。ユーザマップ上ではそれぞれのユーザが商品に対しどのくらい似ている嗜好をもつかを表し、商品マップ上ではそれぞれの商品に対するユーザの評価がどのくらい似ているかを表す。またモード間解析においては、2 つのモード間の関係を可視化する必要がある。たとえば、ある商品群に焦点を当てたときのユーザの嗜好の解析や、特定のユーザ集団に焦点を当てたときの商品の評価傾向を解析する必要がある。これもユーザマップと商品マップを利用することで実現する。以上の解析を実現するために、ユーザと商品の内在的性質は低次元の潜在変数として表現されると仮定する。そして観測されるデータは、全モードの潜在変数を入力とする非線形写像で生成されると仮定する。

---

<sup>\*2</sup> なお本論文中では [25] でも推奨されているように安定かつ収束の早いバッチ型のアルゴリズムをベースとしている。

## 1.2 本論文の構成

本論文は以下のように構成されている。第2章では TSOM を理解するうえで必要となる基礎知識について説明する。第3章では TSOM と TGTM の基本形について述べる。第4章では人工データによるシミュレーションを行い、TSOM の動作検証および TSOM と TGTM の比較を行う。第5章は実データを用いたシミュレーションを行い、その解析結果を通じて TSOM のデータ可視化テクニックを紹介する。第6章では TSOM の拡張について述べる。第7章では TSOM の体系化および SOM と TSOM の関連について述べる。第8章で全体をまとめる。



## 第 2 章

### 基礎知識

本章では解析対象である関係データの表現方法であるテンソル、そして本論文で提案する TSOM の基盤アルゴリズムである SOM について述べる。テンソルに関しては表記法や演算について説明し、テンソルの代表的な解析手法について概説する。SOM については与えられるデータと解くべきタスクに触れたのち、基本となるアルゴリズムと基底関数による表現に置き換えたアルゴリズムを記述する。また SOM を確率的生成モデルの観点で解釈しベイズ統計を用いて定式化を行った手法である GTM についても述べる。

#### 2.1 テンソルとテンソル解析手法

本論文におけるテンソルとは各成分がスカラーの多次元配列として表されるデータのことであり、ベクトルや行列を拡張した概念である [13]。 $M$  次元配列で表現されるデータは  $M$  階のテンソルあるいは階数  $M$  のテンソルと呼ばれる。数式上では階数はテンソルの成分を記述する際に必要な添字の総数である。例えば 2 階のテンソルが行列、1 階のテンソルがベクトル、0 階のテンソルがスカラーを表す。0 階から 3 階までのテンソルを図 2.1 に示す。図 2.1 にも表されているように、テンソルの各次元のことをモードと呼ぶ [13]。

##### 2.1.1 スカラー、ベクトル、行列、テンソルの表記法

本論文でスカラーはすべて実数  $\mathbb{R}$  であり、 $x, y$  のようにイタリックで表記する。ただし  $d, i, j, k, l, m, n$  および  $D, I, J, K, L, M, N$  は例外であり、これらはすべて自然数  $\mathbb{N}$  である。これらの例外のうち小文字はベクトル、行列、テンソルの添字に用い、成分を指定

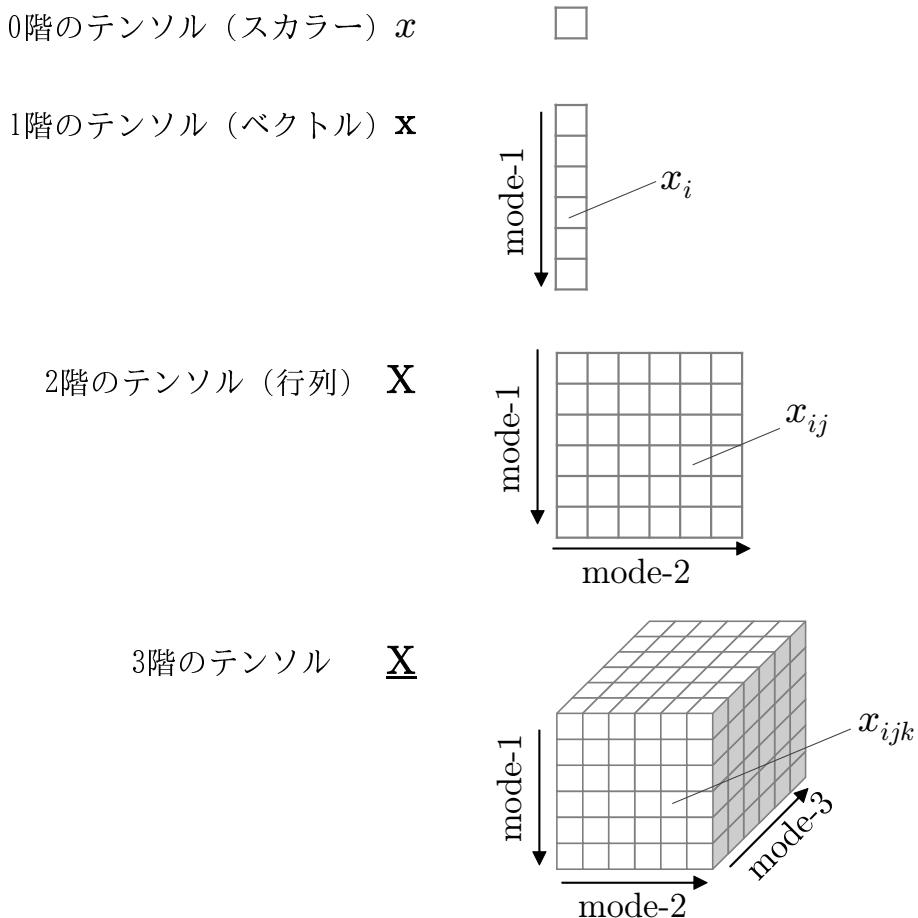


図 2.1 テンソルの構造を示す概念図。各格子内にスカラー値が格納されている。

する際に用いる。また大文字はその上限である。すなわち  $i \in \{1, \dots, I\}$  である。ベクトルはボールド体の小文字で表記する (e.g.,  $\mathbf{x}, \mathbf{y}$ )。なおベクトルは断りのない限り、原則として列ベクトルとする。行列はボールド体の大文字 (e.g.,  $\mathbf{X}, \mathbf{Y}$ )、3 階以上のテンソルはボールド体の大文字に下線をつけて表す (e.g.,  $\underline{\mathbf{X}}, \underline{\mathbf{Y}}$ )。

行列  $\mathbf{A}$  の第  $(i, j)$  成分は  $a_{ij}$  と表記する。同様に、 $a_{ijk}$  は 3 階のテンソル  $\underline{\mathbf{A}}$  の第  $(i, j, k)$  成分を表している。 $M$  階のテンソルの  $\underline{\mathbf{A}}$  の第  $(k_1, k_2, \dots, k_M)$  成分は  $a_{k_1 k_2 \dots k_M}$  であり、 $\mathbf{k} = (k_1, \dots, k_M)$  を用いて  $a_{\mathbf{k}}$  とも書く。

部分配列を表記する際に MATLAB 風のコロン表記を用いる。たとえば  $\mathbf{x}_{i:}$  と  $\mathbf{x}_{:j}$  はそれぞれ行列  $\mathbf{X}$  の第  $i$  行と第  $j$  列の成分から成るベクトルである (どちらも列ベクトルで

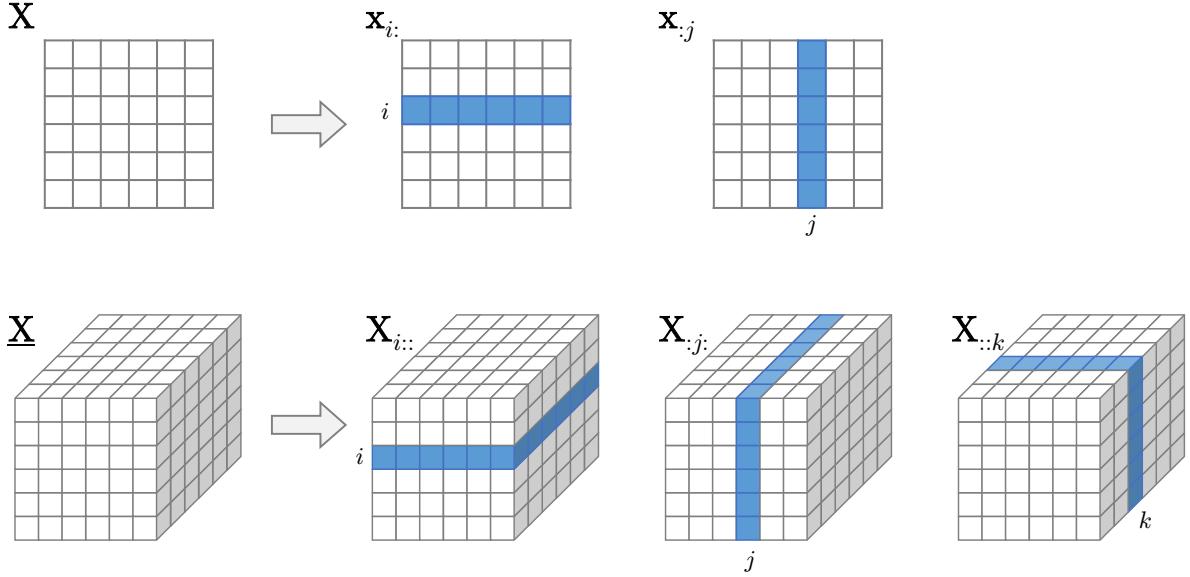


図 2.2 テンソルのスライスを表す概念図。青色で示した範囲が左上の式と対応している。

ある). 同様に,  $\mathbf{x}_{ij:}$  は 3 階のテンソル  $\underline{\mathbf{X}}$  を第 3 モードに沿って切り出したベクトルである.  $M$  階のテンソルに対して,  $(M - 1)$  階のサブテンソルはスライスと呼ぶこととする. たとえば  $\underline{\mathbf{A}}_{::i::}$  は  $\underline{\mathbf{A}}$  の第 3 モードの第  $i$  スライスである. 任意の階数におけるスライスを表すため,  $\underline{\mathbf{A}}$  の第  $m$  モード第  $i$  スライスは  $\underline{\mathbf{A}}_{i(m)}$  とも書く. すなわち  $\underline{\mathbf{A}}_{::i::} = \underline{\mathbf{A}}_{i(3)}$  である. 2 階のテンソルおよび 3 階のテンソルに対するスライスを図 2.2 に示す.

### 2.1.2 テンソル-行列積

テンソル間の積の記号やその表記は, 行列同士の積よりも複雑である [13, 28]. ここでは本論文で重要なテンソル-行列積についてのみ述べておく. 第  $m$  モードに関するテンソル-行列積は  $\times_m$  と表記する. たとえば  $M$  階テンソル  $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$  と行列  $\mathbf{A} \in \mathbb{R}^{J_m \times I_m}$  の第  $m$  モードテンソル-行列積は

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_m \mathbf{A} \quad (2.1)$$

と書く. この積の計算結果を成分で書くと

$$y_{i_1 \dots i_{m-1} j_m i_{m+1} \dots i_M} = \sum_{i_m=1}^{I_m} x_{i_1 \dots i_m \dots i_M} a_{j_m i_m} \quad (2.2)$$

となる。 $\underline{\mathbf{Y}}$  は  $M$  階のテンソルであり、 $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times \cdots \times I_{m-1} \times J_m \times I_{m+1} \times \cdots \times I_M}$  である。

テンソル-行列積は  $M = 2$  の場合、すなわち  $\underline{\mathbf{X}}$  が行列  $\mathbf{X}$  の場合について考えると、行列の積との対応をとることができます。たとえば  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ ,  $\mathbf{A} \in \mathbb{R}^{J_1 \times I_1}$ ,  $\mathbf{B} \in \mathbb{R}^{J_2 \times I_2}$  とすると、

$$(\mathbf{X} \times_1 \mathbf{A})_{j_1 i_2} = (\mathbf{AX})_{j_1 i_2} = \sum_{i_1=1}^{I_1} x_{i_1 i_2} a_{j_1 i_1} \quad (2.3)$$

$$(\mathbf{X} \times_2 \mathbf{B})_{i_1 j_2} = (\mathbf{XB}^T)_{i_1 j_2} = \sum_{i_2=1}^{I_2} x_{i_1 i_2} b_{j_2 i_2} \quad (2.4)$$

となっており、テンソル-行列積は行列の積を一般化したものであることがわかる。

なお、 $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ ,  $\mathbf{A} \in \mathbb{R}^{J_n \times I_n}$ ,  $\mathbf{B} \in \mathbb{R}^{J_m \times I_m}$ (ただし  $n \neq m$ ),  $\mathbf{C} \in \mathbb{R}^{K_n \times J_n}$  とすると、テンソル-行列積は以下の 2 つの性質を満たす [16]。

$$(\underline{\mathbf{X}} \times_n \mathbf{A}) \times_m \mathbf{B} = (\underline{\mathbf{X}} \times_m \mathbf{B}) \times_n \mathbf{A} = \underline{\mathbf{X}} \times_n \mathbf{A} \times_m \mathbf{B} \quad (2.5)$$

$$(\underline{\mathbf{X}} \times_n \mathbf{A}) \times_n \mathbf{C} = \underline{\mathbf{X}} \times_n (\mathbf{CA}) \quad (2.6)$$

(2.3)-(2.5) により、特異値分解をテンソル-行列積を用いて表現することができ、

$$\mathbf{X} = \mathbf{U} \Sigma \mathbf{V}^T = \Sigma \times_1 \mathbf{U} \times_2 \mathbf{V} \quad (2.7)$$

と表される。この表現はテンソルの代表的解析手法である Tucker 分解および CP 分解と密接に関係している。これらのテンソル分解手法については次節で述べる。

また  $M$  個の行列  $\{\mathbf{A}\} = \{\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(M)}\}$  が与えられたとき、 $\underline{\mathbf{X}}$  と  $\{\mathbf{A}\}$  のすべてのモードに関するテンソル-行列積は

$$\underline{\mathbf{X}} \times \{\mathbf{A}\} \triangleq \underline{\mathbf{X}} \times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_M \mathbf{A}^{(M)} \quad (2.8)$$

と表記し、第  $m$  モードを除くすべてのモードに関するテンソル-行列積は

$$\underline{\mathbf{X}} \times_{-m} \{\mathbf{A}\} \triangleq \underline{\mathbf{X}} \times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_{m-1} \mathbf{A}^{(m-1)} \times_{m+1} \mathbf{A}^{(m+1)} \times_{m+2} \cdots \times_M \mathbf{A}^{(M)} \quad (2.9)$$

とする。(2.8),(2.9) を用いることで、任意の階数のテンソルに関するテンソル-行列積が簡潔に記述できるようになる。

### 2.1.3 テンソル解析手法

行列で表現されるデータ(すなわち多変量データ)の特徴抽出や次元削減(または次元縮約)には、特異値分解や非負値行列因子分解などの行列分解が広く利用されている。テン

ソルで表現されるデータ（すなわち関係データ）についても同様に、テンソル分解によって低次なテンソルや行列の積の形でデータを近似表現する。ここでは代表的なテンソル分解手法である Tucker 分解と CP 分解について紹介する<sup>\*1</sup>。

Tucker 分解ではテンソルを 1 つのコアテンソルと各モードに対する行列の積の形で分解する [13, 14, 16, 29]。たとえば 3 階のテンソル  $\underline{\mathbf{X}}$  に対する Tucker 分解は

$$\underline{\mathbf{X}} \simeq \underline{\mathbf{A}} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \quad (2.10)$$

$$= \underline{\mathbf{A}} \times \{\mathbf{U}\} \quad (2.11)$$

となる。ここで  $\underline{\mathbf{A}}$  がコアテンソルと呼ばれる。この式を成分で書くと、

$$x_{k_1 k_2 k_3} \simeq \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \sum_{j_3=1}^{J_3} a_{j_1 j_2 j_3} u_{k_1 j_1}^{(1)} u_{k_2 j_2}^{(2)} u_{k_3 j_3}^{(3)} \quad (2.12)$$

となる。また  $J_m \leq K_m$  である。概念図を図 2.3 に示す。 (2.10) は (2.7) を 3 次テンソルに拡張した式になっており、特に  $\{\mathbf{U}\}$  が直交行列（ユニタリ行列）でありコアテンソル  $\underline{\mathbf{A}}$  が全直交性（all-orthogonality）を持つような Tucker 分解は高階特異値分解（Higher-Order Singular Value Decomposition: HOSVD）とよばれる [16]。また (2.10) は 3 つのモードすべてに対して分解表現するため、このテンソル分解は Tucker3 とも呼ばれる。また 3 階のテンソル  $\underline{\mathbf{X}}$  が 1 つのコアテンソルと 2 つの行列に  $\underline{\mathbf{X}} \simeq \underline{\mathbf{A}} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)}$  と分解されることは Tucker2、1 つのコアテンソルと 1 つの行列に  $\underline{\mathbf{X}} \simeq \underline{\mathbf{A}} \times_1 \mathbf{U}^{(1)}$  と分解されるときは Tucker1 と呼ばれる [13]<sup>\*2</sup>。

もうひとつの代表的なテンソル分解は CP 分解である [30]<sup>\*3</sup>。CP 分解では以下のようにテンソルをランク 1 テンソルの和に分解する。

$$x_{k_1 k_2 k_3} = \sum_{j=1}^J \lambda_j u_{k_1 j}^{(1)} u_{k_2 j}^{(2)} u_{k_3 j}^{(3)}. \quad (2.13)$$

ここで  $\{\mathbf{U}\}$  は正規化された行列である。この式は (2.12) においてコアテンソルが対角テンソルの場合、つまり  $J_1 = J_2 = J_3$  であり  $j_1 = j_2 = j_3$  のときは  $a_{j_1 j_2 j_3} = \lambda_j$ 、それ

<sup>\*1</sup> なお本節では原典の手法にしたがって解析対象を 3 階のテンソルとしているが、自然に  $M$  次テンソルに拡張することができる。

<sup>\*2</sup> Tucker のあとに続く数字は分解表現するモードの総数を表しており、解析対象である  $\underline{\mathbf{X}}$  の階数ではないことに注意が必要である。

<sup>\*3</sup> なお CP 分解は幾度も再発明されており、tensor rank decomposition, canonical polyadic decomposition, parallel factor analysis: PARAFAC, canonical decomposition: CANDECOMP とも呼ばれている。近年では canonical polyadic decomposition あるいは CANDECOMP/PARAFAC decomposition の意味で CP 分解と呼ばれている。

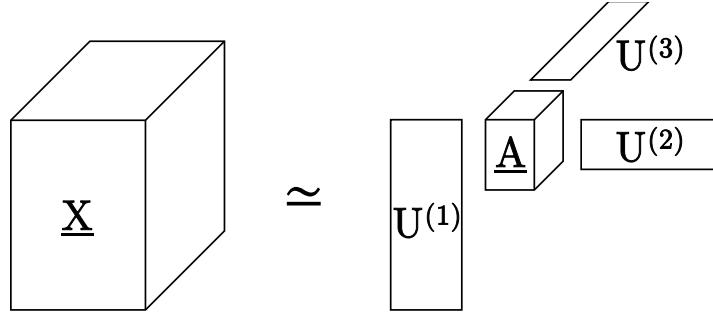


図 2.3 Tucker 分解の概念図.

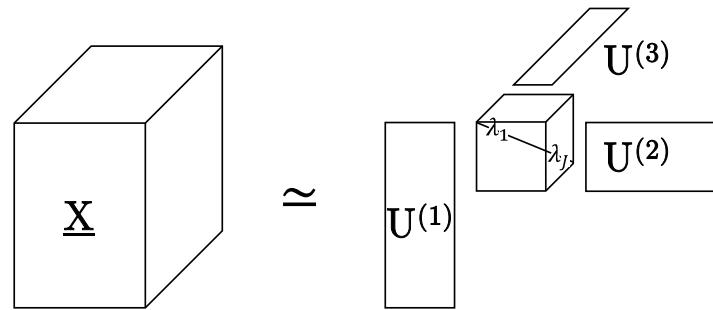


図 2.4 CP 分解の概念図.

以外は  $a_{j_1 j_2 j_3} = 0$  となっている場合に相当する。すなわち CP 分解もまた Tucker 分解に拘束条件を加えたものとみなすことができる。なお CP 分解は HOSVD とは異なる観点から特異値分解をテンソルに拡張したアルゴリズムとなっている。概念図を図 2.4 に示す。

テンソル解析を非線形に拡張する場合、もっともオーソドックスなアプローチはカーネル法を用いることである [31, 32]。また多様体学習を用いる方法も提案されている [33, 34]。

## 2.2 自己組織化マップ (Self-Organizing Map: SOM)

まず SOM で学習する入力データ集合について説明する。今、 $\Omega$  を調べたい対象の母集団とする。そしてサンプル  $\omega \in \Omega$  に対して得られる観測データを  $\mathbf{x}(\omega) \in \mathcal{X}$  とする。こ

ここで  $\mathcal{X}$  は高次元の観測空間であり、本論文では  $D$  次元のユークリッド空間、すなわち  $\mathcal{X} = \mathbb{R}^D$  とする。また  $\Omega_S = \{\omega_1, \dots, \omega_N\} \subset \Omega$  をサンプリングによって得られたインスタンス集合（サンプル集合）とし、インスタンス  $\omega_n$  から得られた観測データを  $\mathbf{x}_{n:}$  とする。すべてのインスタンスに対する観測データをまとめた  $\mathbf{X} = (\mathbf{x}_{n:})$  は  $N \times D$  の行列である。なおこのようなデータセットは多変量データとよばれる。

SOM を生成モデルの観点で見たとき、 $\mathbf{x}_{n:}$  は潜在変数  $\mathbf{z}_{n:} \in \mathcal{Z}$  と滑らかな非線形写像  $f(\mathbf{z}) : \mathcal{Z} \rightarrow \mathcal{X}$  により、 $\mathbf{x}_{n:} = f(\mathbf{z}_{n:}) + \boldsymbol{\epsilon}$  として生成されたと仮定する。ここで  $\boldsymbol{\epsilon}$  は  $D$  次元の等方正規ノイズであり、潜在空間  $\mathcal{Z}$  の次元  $L$  は観測空間の次元  $D$  よりも小さい。SOM のタスクとは、 $\mathbf{x}_{n:} \simeq f(\mathbf{z}_{n:})$  となるように個々のインスタンスに対する潜在変数  $\{\mathbf{z}_{n:}\}$  と非線形写像  $f(\mathbf{z})$  を推定することである。その結果、潜在空間  $\mathcal{Z}$  上にインスタンスが配置され、観測空間での現象を可視化することができる。可視化の関係上、 $\mathcal{Z}$  は低次元空間  $\mathbb{R}^L$  内の閉領域、主に 2 次元正方形領域が用られる。また本論文では  $\mathbf{z}$  の事前分布を  $\mathcal{Z} = [-1, 1]^L \subset \mathbb{R}^L$  ( $L = 1$  or  $2$ ) における一様分布と仮定するが、この仮定は他のケースについても容易に一般化できる。SOM の生成モデルを図 2.5(a) に示す。

通常の SOM では非線形写像  $f(\mathbf{z})$  を表現するために、潜在空間  $\mathcal{Z}$  は  $K$  個の規則的なノードに離散化される。今、 $\zeta_{k:}$  を第  $k$  ノードの潜在空間  $\mathcal{Z}$  における位置ベクトルとする。そして第  $k$  ノードに関する潜在空間から観測空間への写像を  $\mathbf{y}_{k:} \triangleq f(\zeta_{k:})$  とする。 $K$  個の写像を用いて、写像全体は行列  $\mathbf{Y} = (\mathbf{y}_{k:})$  として表せる。 $\mathbf{y}_{k:}$  は第  $k$  ノードに対する観測空間  $\mathcal{X}$  の一点を指し示しているため、参照ベクトルと呼ばれる。SOM のアーキテクチャを図 2.5(b) に示す。

SOM のアルゴリズムは広義の期待値最大化法 (EM アルゴリズム) となっており、E ステップと M ステップを繰り返し実行することで潜在変数と参照ベクトルを推定する [35, 36]。以下に SOM のアルゴリズムを述べるが、テンソルへの拡張を考え、行列表記を用いる。

## E ステップ

SOM の E ステップでは、各データに対して最大事後確率 (maximum a posteriori: MAP) を与えるノード (勝者ノード) を求める。

$$k_n^* = \arg \min_k \|\mathbf{x}_{n:} - \mathbf{y}_{k:}\|^2, \quad (2.14)$$

$$b_{kn} = \delta(k, k_n^*). \quad (2.15)$$

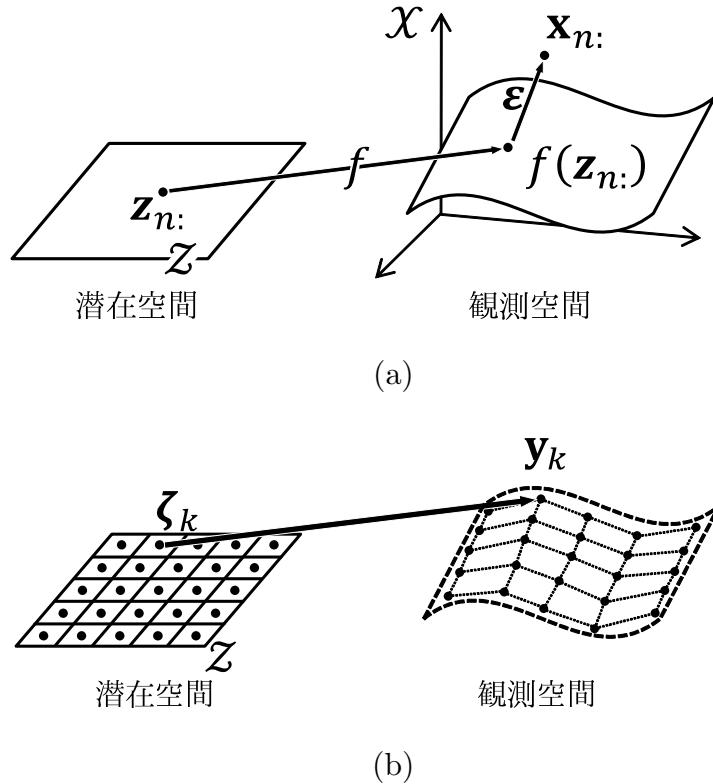


図 2.5 SOM の概念図. (a)SOM の生成モデル的解釈. 潜在変数  $\mathbf{z}_{n:}$  から  $f(\mathbf{z})$  により観測空間に写像され, ノイズ  $\epsilon$  が加わって観測データ  $\mathbf{x}_{n:}$  が生成されたと仮定する. SOM の目的は観測データ  $\mathbf{x}_{n:}$  から写像  $f(\mathbf{z})$  と潜在変数  $\mathbf{z}_{n:}$  を同時推定することである. (b)SOM のアーキテクチャ. 潜在空間に等間隔に配置された各ノードは観測空間への写像をもつ. この参照ベクトル全体で非線形写像を離散近似する.

ここで  $\mathbf{B} = (b_{kn}) \in \mathbb{R}^{K \times N}$  は勝者ノードを表した行列である. また  $\delta$  はクロネッカーのデルタである. 次に  $\mathbf{H} \in \mathbb{R}^{K \times K}$  を次式で定義する近傍行列とする.

$$h_{kk';\sigma(t)} \triangleq \exp \left[ -\frac{1}{2\sigma^2(t)} \|\zeta_{k:} - \zeta_{k':}\|^2 \right]. \quad (2.16)$$

$\sigma(t)$  は近傍半径であり, 学習時刻  $t$ とともに次第に減少させていく. この近傍半径の縮小はシミュレーテッドアニーリングと同様の効果があり, 局所解を回避する上で必要である. 勝者行列  $\mathbf{B}$  と近傍行列  $\mathbf{H}$  を用いて, 負担率行列  $\mathbf{R} \in \mathbb{R}^{K \times N}$  は次のように決定される.

$$\mathbf{R} = \mathbf{HB}. \quad (2.17)$$

以上の計算により潜在変数の期待値をラプラス近似によって近似表現している。また以下の 2 つの行列も M ステップのために計算しておく。

$$\mathbf{G} = \text{diag} \left( \sum_{n=1}^N r_{nk} \right), \quad (2.18)$$

$$\tilde{\mathbf{R}} = \mathbf{G}^{-1} \mathbf{R}. \quad (2.19)$$

$\mathbf{G} \in \mathbb{R}^{K \times K}$  は各ノードの負担率の総和を表す対角行列で、 $\tilde{\mathbf{R}} \in \mathbb{R}^{K \times N}$  は  $\mathbf{G}$  によって規格化した負担率行列である。

### M ステップ

M ステップでは、期待値誤差を最小にするように写像  $\mathbf{Y}$  を更新する。

$$\mathbf{y}_{k:} = \frac{1}{g_{kk}} \sum_{n=1}^N r_{kn} \mathbf{x}_{n:} \quad (2.20)$$

(2.20) を行列表記にすると、

$$\mathbf{Y} = \mathbf{G}^{-1} \mathbf{R} \mathbf{X} = \tilde{\mathbf{R}} \mathbf{X} \quad (2.21)$$

となる。また、テンソル-行列積の表記を用いると、(2.21) は

$$\mathbf{Y} = \mathbf{X} \times_1 \tilde{\mathbf{R}} \quad (2.22)$$

と書ける。ここで、データ行列  $\mathbf{X}$  と写像行列  $\mathbf{Y}$  は 2 階のテンソルとみなしている。以上の 2 ステップを近傍半径を縮小しつつ推定結果が収束するまで繰り返すのが SOM のアルゴリズムである。

多くの先行研究により、SOM の目的関数は次式であることが示されている [35, 36, 37, 38, 39]。

$$F = -\frac{1}{2N} \sum_{n=1}^N \sum_{k=1}^K h_{kk_n^*} \|\mathbf{x}_{n:} - \mathbf{y}_{k:}\|^2 \quad (2.23)$$

M ステップ (2.21) ではこの目的関数を最小にする  $\mathbf{Y}$  を求めている。このことは  $\frac{\partial F}{\partial \mathbf{Y}} = 0$  となる  $\mathbf{Y}$  を求めることで容易に示すことが出来る。また E ステップは若干修正した式が得られる。

$$k_n^* = \arg \min_{k'} \sum_{k=1}^K h_{kk'} \|\mathbf{x}_{n:} - \mathbf{y}_{k:}\|^2. \quad (2.24)$$

オリジナルの SOM における勝者決定 (2.14) は (2.24) の近似式とみることができる。なぜなら写像が滑らかであれば写像先が局所的に平面となり、両者はおおよそ一致するから

である。さらに近傍半径が十分に小さければ両者はほぼ一致する。なお目的関数  $F$  は近傍半径  $\sigma$  の関数であるため、学習を通して  $F$  は徐々に変化することに注意する。

### 2.2.1 基底関数型 SOM

通常の SOM では潜在空間を  $K$  個の離散ノードに分割して非線形関数  $f(\mathbf{z})$  を表現していた。ここでは連続で微分可能な基底関数を用いて、その線形結合で表現することを考える。すなわち、 $\{\phi_1(\mathbf{z}), \dots, \phi_J(\mathbf{z})\}$  を潜在空間  $\mathcal{Z}$  上で定義された適切な基底関数集合とし、非線形関数  $f(\mathbf{z})$  は

$$f(\mathbf{z}) \simeq \sum_{j=1}^J \mathbf{w}_{j:} \phi_j(\mathbf{z}) = \mathbf{W}^T \boldsymbol{\phi}(\mathbf{z}) \quad (2.25)$$

という線形基底関数モデルで十分な精度で表現できるとする。 $\mathbf{W} \in \mathbb{R}^{J \times D}$  は係数行列である。これにより通常の SOM で写像を  $f(\zeta_{k:}) = \mathbf{y}_{k:}$  としていた部分を  $f(\zeta_{k:}) = \mathbf{W}^T \boldsymbol{\phi}(\zeta_{k:})$  として置き換えることができる。

基底関数で表現することの利点は以下の通りである。

- 連続で微分可能な写像表現が可能になり、写像を任意の解像度で表現できる。
- E ステップのノードの総当たりによる勝者決定の代わりに勾配法を用いることができ、計算量がノード数に依存しなくなる。
- 適切な基底を選択することで、境界領域のバイアスが緩和される（これは基底関数の線形和で表現できる写像が制限されていることによる副次的な効果である）。
- 正規直交な基底を選べば、2つの写像間の距離は係数行列の距離と等しくなり、アルゴリズムを Tensor 版に拡張した際に E ステップの高速化ができる。このについては 3 章で述べる。また、正規直交であれば近似的に M ステップの逆行列の計算を避けることができる。

基底関数型 SOM の場合、目的関数は

$$F = -\frac{1}{2N} \sum_{n=1}^N \sum_{k=1}^K h_{kk_n^*} \|\mathbf{x}_{n:} - \mathbf{W}^T \boldsymbol{\phi}(\zeta_{k:})\|^2 \quad (2.26)$$

となる。また SOM のアルゴリズムの E ステップは (2.14) が以下の式に置き換わる。

$$k_n^* = \arg \min_k \|\mathbf{x}_{n:} - \mathbf{W}^T \boldsymbol{\phi}(\zeta_{k:})\|^2. \quad (2.27)$$

そして、SOM と同様に目的関数を最小にするよう M ステップを求める。 $\frac{\partial F}{\partial \mathbf{W}} = 0$  となる  $\mathbf{W}$  を求めると、

$$\mathbf{W} = (\Phi^T \mathbf{G} \Phi)^{-1} \Phi^T \mathbf{R} \mathbf{X} \quad (2.28)$$

$$= (\Phi^T \mathbf{G} \Phi)^{-1} \Phi^T \mathbf{G} \mathbf{G}^{-1} \mathbf{R} \mathbf{X} \quad (2.29)$$

$$= \Phi_{\mathbf{G}}^{\#} \tilde{\mathbf{R}} \mathbf{X}. \quad (2.30)$$

となる。なお、 $\Phi = (\phi(\zeta_{1:}), \dots, \phi(\zeta_{K:}))^T$  であり、 $K \times J$  行列である。また、 $\Phi_{\mathbf{G}}^{\#}$  は  $\mathbf{G}$  を重みとする重み付き一般化逆行列である。すなわち、 $\Phi_{\mathbf{G}}^{\#} \triangleq (\Phi^T \mathbf{G} \Phi)^{-1} \Phi \mathbf{G}$  である。また  $J \times N$  行列  $\tilde{\mathbf{Q}}$  を次式で定義する。

$$\tilde{\mathbf{Q}} \triangleq \Phi_{\mathbf{G}}^{\#} \tilde{\mathbf{R}} = (\Phi^T \mathbf{G} \Phi)^{-1} \Phi^T \mathbf{R}. \quad (2.31)$$

すると (2.30) は

$$\mathbf{W} = \tilde{\mathbf{Q}} \mathbf{X} \quad (2.32)$$

となる。離散ノード表現の更新式 (2.21) と基底関数表現の更新式 (2.32) を見比べると、 $\mathbf{Y}$  と  $\tilde{\mathbf{R}}$  がそれぞれ  $\mathbf{W}$  と  $\tilde{\mathbf{Q}}$  に置き換わったことがわかる。

E ステップは (2.27) のように通常の SOM と同様に  $K$  個の代表点を総当たりで求めてもよいが、連続でありかつ微分可能な基底関数を用いれば、勾配法により勝者位置を求めることもできる。最急降下法の場合、 $E_n(\mathbf{z}) = \|\mathbf{x}_n - \mathbf{W}\phi(\mathbf{z})\|^2$  とすると潜在変数  $\mathbf{z}_{n:} \in \mathbb{R}^L$  は次式で更新できる。

$$\mathbf{z}_{n:}^{\text{new}} := \mathbf{z}_{n:}^{\text{old}} - \eta \frac{\partial E_n(\mathbf{z})}{\partial \mathbf{z}_{n:}^{\text{old}}} \quad (2.33)$$

なお、潜在変数は E ステップごとに 1 回更新するだけでよい。なぜなら学習の収束速度は近傍半径の収束速度に依存しており、各ステップの潜在変数の推定に対してその速度よりも速く収束する計算法を用いても全体の速度に影響を与えないためである。また、勾配法による勝者決定を用いる場合、負担率行列は以下の式になる。

$$r_{kn} = \exp \left[ -\frac{1}{2\sigma^2(t)} \|\zeta_{k:} - \mathbf{z}_{n:}\|^2 \right] \quad (2.34)$$

M ステップ (2.30) では、一般化逆行列  $\Phi_{\mathbf{G}}^{\#}$  を各イテレーションごとに計算しなければならず、計算コストがかかる。ところでサンプルの数が十分であれば、 $k$  番ノードの負担率の総和  $g_{kk}$  と事前分布  $P_k$  の関係は  $\frac{1}{N} g_{kk} \simeq P_k$  となる。なおかつ事前分布が一様分布であるとき、 $\mathbf{G} = \frac{N}{K} \mathbf{I}$  と近似できる。これにより  $(\Phi^T \mathbf{G} \Phi)^{-1} \Phi^T \mathbf{G} \simeq (\Phi^T \Phi)^{-1} \Phi^T$

となり、 $\Phi_{\mathbf{G}}^{\#}$  は通常の一般化逆行列  $\Phi^{\#}$  と近似することができる。 $\Phi^{\#}$  は学習を通して一定であるため、事前に一度計算しておくだけでよい。さらに、正規直交な基底関数を用いると  $\Phi^T \Phi = \mathbf{I}$  により  $\Phi^{\#} = \Phi^T$  となるため、逆行列の計算をする必要がなくなる。

### 2.2.2 生成位相写像 (Generative Topographic Mapping: GTM)

GTM は確率的生成モデルから導出されたアルゴリズムであり、SOM のベイズ理論モデルとみなせる [26, 27, 40]。GTM のアルゴリズムは EM アルゴリズム（もしくは変分ベイズ法）で記述される。E ステップでは、負担率を次式で求める。

$$r_{kn} = \frac{\exp\left[-\frac{\beta}{2}\|\mathbf{x}_{n:} - \mathbf{y}_{k:}\|^2\right]}{\sum_{k'=1}^K \exp\left[-\frac{\beta}{2}\|\mathbf{x}_{n:} - \mathbf{y}_{k':}\|^2\right]}. \quad (2.35)$$

なお  $\beta$  は精度パラメータであり、分散  $\sigma^2$  の逆数である。SOM と異なり、E ステップにおいて近傍関係を持たない。その代わり GTM では M ステップで他の仮定を導入して滑らかな写像を実現する。オリジナルの GTM ではガウス関数による Radial Basis Function(RBF) 基底を用いており [26]、また後にはガウス確率過程 (Gaussian Process: GP) も用いられる [40]。

RBF 基底を用いる場合、M ステップは

$$\mathbf{W} = \left(\Phi^T \mathbf{G} \Phi + \frac{\lambda}{\beta} \mathbf{I}\right)^{-1} \Phi^T \mathbf{R} \mathbf{X}. \quad (2.36)$$

となる。ここで  $\Phi$  は RBF 基底を  $K$  ノードに離散化した行列であり、 $\frac{\lambda}{\beta} \mathbf{I}$  は正則化項、すなわち写像の事前分布である。基底関数型 SOM の場合と同様に  $\tilde{\mathbf{Q}} \triangleq \left(\Phi^T \mathbf{G} \Phi + \frac{\lambda}{\beta} \mathbf{I}\right)^{-1} \Phi^T \mathbf{R}$  とすれば、M ステップは (2.32) と同じく  $\mathbf{W} = \tilde{\mathbf{Q}} \mathbf{X}$  と書ける。

GP を用いる場合の M ステップは

$$\mathbf{Y} = \left(\mathbf{G} + \frac{\lambda}{\beta} \mathbf{H}^{-1}\right)^{-1} \mathbf{R} \mathbf{X} \quad (2.37)$$

となる。ここで近傍行列  $\mathbf{H}$  を GP の共分散関数として用いた。なお基底関数と共に分散関数が  $\Phi \Phi^T = \mathbf{H}$  を満たすとき、RBF 型と GP 型は同じものになる。

非線形写像をガウス過程で表現し、潜在変数を MAP 推定に置き換えると、GTM のアルゴリズムは Gaussian Process Latent Variable Model: GPLVM とほぼ等価となる [41, 42]。また潜在変数を MAP 推定、ガウス RBF 基底（もしくはガウス過程）をガウスカーネル平滑化に置き換えると、GTM は SOM になる。

## 第3章

# Tensor SOM: TSOM

本章では提案手法である Tensor SOM (TSOM) について、解くべきタスクとそのアルゴリズムについて述べる。まず簡潔に説明するために、ユーザと商品の 2 モード関係データを解析する TSOM について記述する。なお本論文では 2 モード関係データを解析する TSOM を TSOM-R<sup>2</sup> (Tensor SOM for 2-mode Relational data) と書く。次にこれを  $M$  モード関係データに一般化して記述する。TSOM-R <sup>$M$</sup>  に関してはアルゴリズムの導出や基底関数での表現についても詳述し、実際にデータ解析に用いる際に必要となる初期化やスケジューリングについても述べる。さらに GTM をベースにした Tensor GTM (TGTM) についても記述する。

### 3.1 TSOM-R<sup>2</sup>

ユーザと商品の 2 モード関係データを解析する場合、解析の対象となるのはユーザ集合  $\Omega^{(\text{user})}$  と商品集合  $\Omega^{(\text{item})}$  である。ユーザ集合と商品集合からサンプルしたインスタンス集合をそれぞれ  $\Omega_S^{(\text{user})} = \{\omega_1^{(\text{user})}, \dots, \omega_{N_1}^{(\text{user})}\}$ ,  $\Omega_S^{(\text{item})} = \{\omega_1^{(\text{item})}, \dots, \omega_{N_2}^{(\text{item})}\}$  とする。そして評価値  $\mathbf{x} \in \mathbb{R}^D$  はユーザと商品の 2 つのインスタンスのペア  $(\omega^{(\text{user})}, \omega^{(\text{item})})$  に対して与えられる。つまり入力データとなるテンソル  $\underline{\mathbf{X}} = (x_{n_1 n_2 d}) \in \mathbb{R}^{N_1 \times N_2 \times D}$  はユーザ、商品、評価項目の 3 階のテンソルとなる。このようなデータが与えられたとき、TSOM-R<sup>2</sup> の目的はそれぞれのモード内における観測対象の関係（すなわちユーザ間の類似性と商品間の類似性）を可視化することと、2 つのモード間の関係（すなわちユーザと商品の関係）を可視化することである。なお、3 階のテンソルのうち 2 つのモードのみを解析対象とするため、データの解析方法としては Tucker2 と同様の形式である。

TSOM-R<sup>2</sup> では 2 モード関係データは 2 つの潜在変数  $\mathbf{z}^{(1)} \in \mathcal{Z}^{(1)}, \mathbf{z}^{(2)} \in \mathcal{Z}^{(2)}$  と滑らかな非線形写像  $f$  により表現されると仮定する。すなわち TSOM-R<sup>2</sup> のタスクは

$$\begin{aligned} f : \mathcal{Z}^{(1)} \times \mathcal{Z}^{(2)} &\longrightarrow \mathcal{Y} \subset \mathcal{X} = \mathbb{R}^D \\ (\mathbf{z}^{(1)}, \mathbf{z}^{(2)}) &\longmapsto \mathbf{y}(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}). \end{aligned} \quad (3.1)$$

となる潜在変数および非線形写像を求めることがある。非線形写像  $f$  を表現するために、それぞれの潜在空間  $\mathcal{Z}^{(1)}, \mathcal{Z}^{(2)}$  を  $K_1, K_2$  のノード点に離散化し、第  $m$  モードの第  $k_m$  ノードの潜在空間における位置ベクトルを  $\zeta_{k_m}^{(m)}$  とする。これによりノードペアに  $(k_1, k_2)$  に対して参照ベクトル  $\mathbf{y}_{k_1 k_2} = f(\zeta_{k_1}^{(1)}, \zeta_{k_2}^{(2)})$  が定義され、非線形写像全体は 3 次テンソル  $\underline{\mathbf{Y}} \in \mathbb{R}^{K_1 \times K_2 \times D}$  で表される。TSOM-R<sup>2</sup> のアーキテクチャを図 3.1(a) に示す。

TSOM-R<sup>2</sup> のアルゴリズムでは、スライス多様体とインスタンス多様体を用いて勝者ノード決定を行う。スライス多様体とは写像  $f$  によって形成される多様体  $\mathcal{Y}$  の部分多様体であり、 $\underline{\mathbf{Y}}$  のスライスとして表現される。たとえばモード 1 の第  $k_1$  スライス多様体は  $\mathcal{Y}_{k_1}^{(1)} = f(\zeta_{k_1}^{(1)}, \mathcal{Z}^{(2)})$  であり、ノード点に離散化することで  $\mathbf{Y}_{k_1 \cdot \cdot} \in \mathbb{R}^{K_2 \times D}$  として表すことができる。つまりスライス多様体はひとつの潜在変数を特定したときのデータ分布を表現するものとみなせる。一方で、インスタンス多様体とは特定のインスタンスに関するデータ分布を表現したものである。たとえばインスタンス  $\omega_{n_1}^{(1)}$  に特定した場合、そのインスタンスに関する観測データ集合  $\{\mathbf{x}_{n_1 1 \cdot \cdot}, \dots, \mathbf{x}_{n_1 N_2 \cdot \cdot}\}$  を作ることができる。そしてこの部分データ集合は  $\mathbf{x}_{n_1 n_2} \simeq f^{(1)}(\mathbf{z}_{n_2}^{(2)} \mid \omega_{n_1}^{(1)})$  として SOM のようにモデル化できるとする。このとき、インスタンス多様体  $\mathcal{U}_{n_1}^{(1)}$  とは  $\mathcal{U}_{n_1}^{(1)} = f^{(1)}(\mathcal{Z}^{(2)} \mid \omega_{n_1}^{(1)})$  のことであり、ユーザと商品の 2 モード関係データの例では「ある特定のユーザの商品評価傾向」を表している。またこの多様体は離散化により  $\mathbf{U}_{n_1 \cdot \cdot} \in \mathbb{R}^{K_2 \times D}$  として表される。ここで  $\mathbf{u}_{n_1 k_2} = f^{(1)}(\zeta_{k_2}^{(2)} \mid \omega_{n_1}^{(1)})$  である。すなわちインスタンス多様体  $\mathbf{U}_{n_1 \cdot \cdot}$  は 3 次テンソル  $\underline{\mathbf{U}}^{(1)} \in \mathbb{R}^{N_1 \times K_2 \times D}$  のモード 1 の第  $n_1$  スライスである。TSOM のアルゴリズムにおいて、インスタンス多様体はすべてのインスタンスに対して求められる。そしてそれらを各インスタンスの特徴ベクトルとみなして勝者ノード（すなわち潜在変数）を決定する。スライス多様体とインスタンス多様体について、図 3.1(b) に示す。

通常の SOM と同様に、TSOM のアルゴリズムも E ステップと M ステップの繰り返しで構成される。以下のアルゴリズム記述では、SOM の式との対応と実装時の容易さを考慮してテンソル表記を用いた式と用いない式を併記する。

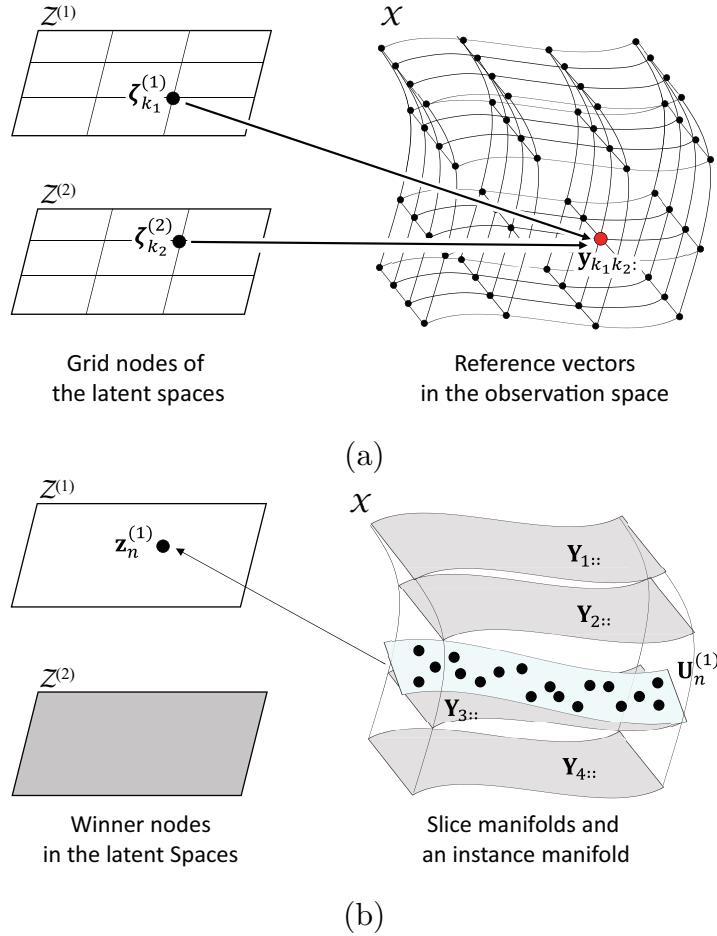


図 3.1 (a) TSOM-R<sup>2</sup> のアーキテクチャ. 潜在空間から観測空間への写像は、2つの潜在空間のノードペア  $(\zeta_{k_1}^{(1)}, \zeta_{k_2}^{(2)})$  から観測空間への参照ベクトル  $\mathbf{y}_{k_1 k_2 :}$  を並べたものとして表現される. 潜在空間が各々 2 次元の場合、非線形写像によって観測空間では 4 次元多様体が作られる（図では非線形にゆがんだ 3 次元立方体として描かれている）. (b) 個々のインスタンス  $\omega_n^{(m)}$  に対してインスタンスマニフェスト  $\{\mathbf{U}_{n(m)}^{(m)}\}$  を推定する. 勝者はインスタンスマニフェストにもっとも近いスライスマニフェストである.

## E ステップ

E ステップでは、各モードの各インスタンスに対して勝者を決定する.

$$k_{n_1}^{*(1)} = \arg \min_{k_1} \left\| \mathbf{U}_{n_1 ::}^{(1)} - \mathbf{Y}_{k_1 ::} \right\|^2 \quad (3.2)$$

$$= \arg \min_{k_1} \sum_{k_2=1}^{K_2} \sum_{d=1}^D \left( u_{n_1 k_2 d}^{(1)} - y_{k_1 k_2 d} \right)^2 \quad (3.3)$$

$$k_{n_2}^{*(2)} = \arg \min_{k_2} \left\| \mathbf{U}_{: n_2 ::}^{(2)} - \mathbf{Y}_{: k_2 ::} \right\|^2 \quad (3.4)$$

$$= \arg \min_{k_2} \sum_{k_1=1}^{K_1} \sum_{d=1}^D \left( u_{k_1 n_2 d}^{(2)} - y_{k_1 k_2 d} \right)^2 \quad (3.5)$$

すなわちインスタンス  $\omega_{n_1}^{(1)}$  に対して、インスタンス多様体  $\mathbf{U}_{n_1::}^{(1)}$  にもっとも近いスライス多様体  $\mathbf{Y}_{k_1::}$  が勝者となる（図 3.1(b)）。（3.2）と（2.14）を比較すると、 $\mathbf{U}_{n_1::}^{(1)}$  と  $\mathbf{Y}_{k_1::}$  があたかもモード 1 に関するデータベクトルと参照ベクトルとして機能していることがわかる。

すべての勝者を決定したのち、勝者行列  $\mathbf{B}^{(1)}, \mathbf{B}^{(2)}$  を得る。これにより負担率行列  $\mathbf{R}^{(1)}, \mathbf{R}^{(2)}$  が次式で得られる。

$$b_{k_1 n_1}^{(1)} = \delta(k_1, k_{n_1}^{*(1)}) \quad (3.6)$$

$$b_{k_2 n_2}^{(2)} = \delta(k_2, k_{n_2}^{*(2)}) \quad (3.7)$$

$$\mathbf{R}^{(1)} = \mathbf{H}^{(1)} \mathbf{B}^{(1)} \quad (3.8)$$

$$\mathbf{R}^{(2)} = \mathbf{H}^{(2)} \mathbf{B}^{(2)}. \quad (3.9)$$

ここで  $\mathbf{H}^{(m)}$  はモード  $m$  についての近傍行列である。成分ごとに書くと以下のとおりである。

$$r_{k_1 n_1}^{(1)} = \exp \left[ -\frac{1}{\sigma^{(1)2}} d^2(k_1, k_{n_1}^{*(1)}) \right] \quad (3.10)$$

$$r_{k_2 n_2}^{(2)} = \exp \left[ -\frac{1}{\sigma^{(2)2}} d^2(k_2, k_{n_2}^{*(2)}) \right] \quad (3.11)$$

ここで  $d(k, k')$  は潜在空間におけるユニット間距離である。（2.18）、（2.19）と同様に、負担率の総和を求めた対角行列  $\mathbf{G}^{(1)}, \mathbf{G}^{(2)}$  および規格化した負担率行列  $\tilde{\mathbf{R}}^{(1)}, \tilde{\mathbf{R}}^{(2)}$  も各モードごとに計算しておく。

$$\mathbf{G}^{(1)} = \text{diag} \left( \sum_{n_1=1}^{N_1} r_{k_1 n_1}^{(1)} \right) \quad (3.12)$$

$$\mathbf{G}^{(2)} = \text{diag} \left( \sum_{n_2=1}^{N_2} r_{k_2 n_2}^{(2)} \right) \quad (3.13)$$

$$\tilde{\mathbf{R}}^{(1)} = \mathbf{G}^{(1)-1} \mathbf{R}^{(1)} \quad (3.14)$$

$$\tilde{\mathbf{R}}^{(2)} = \mathbf{G}^{(2)-1} \mathbf{R}^{(2)} \quad (3.15)$$

Mステップ

Mステップでは、写像を表すテンソル  $\underline{\mathbf{Y}}$  と、インスタンス多様体を表すテンソル  $\underline{\mathbf{U}}^{(1)}$ ,  $\underline{\mathbf{U}}^{(2)}$  を更新する。

$$\underline{\mathbf{U}}^{(1)} = \underline{\mathbf{X}} \times_2 \widetilde{\mathbf{R}}^{(2)} \quad (3.16)$$

$$\underline{\mathbf{U}}^{(2)} = \underline{\mathbf{X}} \times_1 \widetilde{\mathbf{R}}^{(1)} \quad (3.17)$$

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \widetilde{\mathbf{R}}^{(1)} \times_2 \widetilde{\mathbf{R}}^{(2)}. \quad (3.18)$$

上記のテンソル-行列積は、成分ごとに明示的に書くと次のようになる。

$$u_{n_1 k_2 d}^{(1)} = \frac{1}{g_{k_2 k_2}^{(2)}} \sum_{n_2=1}^{N_2} r_{k_2 n_2}^{(2)} x_{n_1 n_2 d} \quad (3.19)$$

$$u_{k_1 n_2 d}^{(2)} = \frac{1}{g_{k_1 k_1}^{(1)}} \sum_{n_1=1}^{N_1} r_{k_1 n_1}^{(1)} x_{n_1 n_2 d} \quad (3.20)$$

$$y_{k_1 k_2 d} = \frac{1}{g_{k_1 k_1}^{(1)} g_{k_2 k_2}^{(2)}} \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} r_{k_1 n_1}^{(1)} r_{k_2 n_2}^{(2)} x_{n_1 n_2 d}. \quad (3.21)$$

なお (3.18) は次のように書くこともできる。

$$\underline{\mathbf{Y}} = \underline{\mathbf{U}}^{(1)} \times_1 \widetilde{\mathbf{R}}^{(1)}. \quad (3.22)$$

あるいは次式のように書いても良い。

$$\underline{\mathbf{Y}} = \underline{\mathbf{U}}^{(2)} \times_2 \widetilde{\mathbf{R}}^{(2)}. \quad (3.23)$$

プログラムを作成する場合は、 $\underline{\mathbf{Y}}$  を直接 (3.18) で更新するよりも、(3.22) もしくは (3.23) で更新するほうが効率的である。

## 3.2 TSOM-R<sup>M</sup>

前節の説明を  $M$  モード関係データに一般化して記述する。今、観測対象の母集団の集合を  $\{\Omega^{(1)}, \dots, \Omega^{(M)}\}$  とし、母集団  $\Omega^{(m)}$  からサンプルした  $N_m$  個のインスタンスの集合を  $\Omega_S^{(m)} = \{\omega_1^{(m)}, \dots, \omega_{N_m}^{(m)}\}$  とする。観測データは  $D$  次元ベクトル  $\mathbf{x}_{n_1 \dots n_M} = \mathbf{x}(\omega_{n_1}^{(1)}, \dots, \omega_{n_M}^{(M)})$  であり、 $M$  個のインスタンスの組み合わせに対して与えられる。さらに観測はすべてのインスタンスの組み合わせに対して行われるとする。すなわち  $M$  個のインスタンス集合の直積集合  $\Omega_S = \prod_{m=1}^M \Omega_S^{(m)}$  の要素すべてに対して観測を行うとする。これにより観測データ全体は  $(M+1)$  階のテンソル  $\underline{\mathbf{X}}$  として表される。

TSOM-R<sup>M</sup> の目的は、このような観測データ  $\underline{\mathbf{X}}$  が与えられたときに、 $M$  個のモードそれぞれのモード内の関係とモード間の関係を可視化することである。TSOM-R<sup>M</sup> では  $M$  個の潜在空間  $\{\mathcal{Z}^{(1)}, \dots, \mathcal{Z}^{(M)}\}$  を仮定しており、観測データは  $M$  個の潜在変数を引数とする非線形関数  $f(\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(M)})$  によって表現されるとする。すなわち、

$$\mathbf{x}_{n_1 \dots n_M} \simeq f(\mathbf{z}_{n_1}^{(1)}, \dots, \mathbf{z}_{n_M}^{(M)}). \quad (3.24)$$

である。このときの潜在変数と非線形写像を同時に推定することが TSOM-R<sup>M</sup> のタスクである。

ちなみに通常の SOM では、観測空間の次元  $D$  は潜在空間の次元  $L$  よりも大きくなければならない。そうでなければ位相同型な写像を作ることができないからである。しかし TSOM においてはこの制約は必要ない。実際に、観測データがスカラー ( $D = 1$ ) の場合でも問題なく解析できる。なぜなら TSOM が満たすべき位相保存性は  $\{\underline{\mathbf{X}}_{n(m)}\}$  と  $\{\mathbf{z}_{n:}^{(m)}\}$  の間であり、 $\underline{\mathbf{X}}_{n(m)}$  の次元は通常  $L$  よりも大きいからである。

さて TSOM-R<sup>M</sup> のアルゴリズムについて述べる。TSOM-R<sup>2</sup> と同様に E ステップと M ステップの繰り返し構成されており、E ステップでは各モードの負担率  $\{\mathbf{R}^{(m)}\}$  を求め、M ステップでは各モードのインスタンス多様体を表すテンソル  $\{\underline{\mathbf{U}}^{(m)}\}$  と非線形写像を表すテンソル  $\underline{\mathbf{Y}}$  を求める。データテンソル  $\underline{\mathbf{X}}$  と写像を表すテンソル  $\underline{\mathbf{Y}}$ 、インスタンス多様体を表すテンソル  $\{\underline{\mathbf{U}}^{(m)}\}$  はいずれも  $(M + 1)$  次テンソルになり、それぞれのサイズは  $(N_1 \times \dots \times N_M \times D)$ ,  $(K_1 \times \dots \times K_M \times D)$  および  $(K_1 \times \dots \times K_{m-1} \times N_m \times K_{m+1} \times \dots \times K_M \times D)$  となる。

### E ステップ

TSOM-R<sup>2</sup> の場合と同様に、インスタンス多様体  $\underline{\mathbf{U}}_{n(m)}^{(m)}$  に対して二乗誤差を最小にするスライス多様体  $\underline{\mathbf{Y}}_{k(m)}$  を勝者とする。すなわち、

$$k_n^{*(m)} = \arg \min_k \left\| \underline{\mathbf{U}}_{n(m)}^{(m)} - \underline{\mathbf{Y}}_{k(m)} \right\|^2 \quad (3.25)$$

$$b_{kn}^{(m)} = \delta(k, k_n^{*(m)}) \quad (3.26)$$

である。勝者決定後、負担率行列  $\mathbf{R}^{(m)}$ 、負担率の総和行列  $\mathbf{G}^{(m)}$ 、そして規格化した負担率行列  $\tilde{\mathbf{R}}^{(m)}$  を各モードについて計算する。

$$\mathbf{R}^{(m)} = \mathbf{H}^{(m)} \mathbf{B}^{(m)} \quad (3.27)$$

$$\mathbf{G}^{(m)} = \text{diag} \left( \sum_{n=1}^{N_m} r_{kn}^{(m)} \right) \quad (3.28)$$

$$\tilde{\mathbf{R}}^{(m)} = \mathbf{G}^{(m)}^{-1} \mathbf{R}^{(m)}. \quad (3.29)$$

### M ステップ

M ステップでは  $\underline{\mathbf{Y}}$  と  $\{\underline{\mathbf{U}}^{(m)}\}$  を次式で更新する。

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times \{\tilde{\mathbf{R}}\} \quad (3.30)$$

$$\underline{\mathbf{U}}^{(m)} = \underline{\mathbf{X}} \times_{-m} \{\tilde{\mathbf{R}}\}. \quad (3.31)$$

$\times_{-m}$  は 2 章の (2.9) で定義したように、第  $m$  モードを除くすべてのモードに関するテンソル-行列積である。以上の E ステップと M ステップを近傍半径を縮小しながら繰り返す。

ここで (3.30) と (3.31) の更新式の高速化について考える。もしこれらの  $(M+1)$  個のテンソルを別個に求めた場合、テンソル-行列積を  $M^2$  回計算する必要がある。たとえば  $M=4$  の場合、(3.30) と (3.31) は

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \tilde{\mathbf{R}}^{(1)} \times_2 \tilde{\mathbf{R}}^{(2)} \times_3 \tilde{\mathbf{R}}^{(3)} \times_4 \tilde{\mathbf{R}}^{(4)} \quad (3.32)$$

$$\underline{\mathbf{U}}^{(1)} = \underline{\mathbf{X}} \times_2 \tilde{\mathbf{R}}^{(2)} \times_3 \tilde{\mathbf{R}}^{(3)} \times_4 \tilde{\mathbf{R}}^{(4)} \quad (3.33)$$

$$\underline{\mathbf{U}}^{(2)} = \underline{\mathbf{X}} \times_1 \tilde{\mathbf{R}}^{(1)} \times_3 \tilde{\mathbf{R}}^{(3)} \times_4 \tilde{\mathbf{R}}^{(4)} \quad (3.34)$$

$$\underline{\mathbf{U}}^{(3)} = \underline{\mathbf{X}} \times_1 \tilde{\mathbf{R}}^{(1)} \times_2 \tilde{\mathbf{R}}^{(2)} \times_4 \tilde{\mathbf{R}}^{(4)} \quad (3.35)$$

$$\underline{\mathbf{U}}^{(4)} = \underline{\mathbf{X}} \times_1 \tilde{\mathbf{R}}^{(1)} \times_2 \tilde{\mathbf{R}}^{(2)} \times_3 \tilde{\mathbf{R}}^{(3)} \quad (3.36)$$

となり、全部で 16 回 ( $= 4^2$ ) のテンソル-行列積となる。しかしながら、以下のように計算順序を工夫することで  $[M \log_2 M + 1]$  回まで減らすことができる (ここで  $[x]$  は ceil 関数であり、小数点以下の切り上げを意味する)。

今、インデックスの集合を  $\mathcal{M} = \{1, \dots, M\}$  とする。そして  $\mathcal{M}$  の部分集合を  $\mathcal{M}'$  とする。ここで、あるテンソル  $\underline{\mathbf{X}}$  と行列の集合  $\{\mathbf{A}\}$  の指定したモードに関するテンソル-行列積は、対象となるモードのインデックスの集合  $\mathcal{M}'$  を用いて  $\underline{\mathbf{X}} \times_{\mathcal{M}'} \{\mathbf{A}\}$  と書くこととする。たとえば  $\underline{\mathbf{X}} \times_2 \mathbf{A}^{(2)} \times_4 \mathbf{A}^{(4)} \equiv \underline{\mathbf{X}} \times_{\{2,4\}} \{\mathbf{A}\}$  である。

さらに TSOM の入力となるデータテンソル  $\underline{\mathbf{X}}$  と規格化した負担率行列集合  $\{\tilde{\mathbf{R}}\}$  の  $\mathcal{M}'$  で指定したモードに関するテンソル-行列積の計算結果を  $\underline{\mathbf{P}}(\mathcal{M}')$  と書くこととする。すなわち

$$\underline{\mathbf{P}}(\mathcal{M}') \triangleq \underline{\mathbf{X}} \times_{\mathcal{M}'} \{\tilde{\mathbf{R}}\} \quad (3.37)$$

と定義する。ここで  $\underline{\mathbf{P}}(\mathcal{M}' + \{m\}) = \underline{\mathbf{P}}(\mathcal{M}') \times_m \tilde{\mathbf{R}}^{(m)}$  であることに注意する（ただし  $m \notin \mathcal{M}'$ ）。また  $\underline{\mathbf{Y}} = \underline{\mathbf{P}}(\mathcal{M})$  であること、および  $\underline{\mathbf{U}}^{(m)} = \underline{\mathbf{P}}(\mathcal{M} \setminus \{m\})$  であることから、(3.32)-(3.36) の計算は次のような手順で実行できる。

$$\underline{\mathbf{P}}(\{1, 2\}) = \underline{\mathbf{X}} \times_1 \tilde{\mathbf{R}}^{(1)} \times_2 \tilde{\mathbf{R}}^{(2)} \quad (3.38)$$

$$\underline{\mathbf{P}}(\{3, 4\}) = \underline{\mathbf{X}} \times_3 \tilde{\mathbf{R}}^{(3)} \times_4 \tilde{\mathbf{R}}^{(4)} \quad (3.39)$$

$$\underline{\mathbf{U}}^{(1)} = \underline{\mathbf{P}}(\{3, 4\}) \times_2 \tilde{\mathbf{R}}^{(2)} \quad (3.40)$$

$$\underline{\mathbf{U}}^{(2)} = \underline{\mathbf{P}}(\{3, 4\}) \times_1 \tilde{\mathbf{R}}^{(1)} \quad (3.41)$$

$$\underline{\mathbf{U}}^{(3)} = \underline{\mathbf{P}}(\{1, 2\}) \times_4 \tilde{\mathbf{R}}^{(4)} \quad (3.42)$$

$$\underline{\mathbf{U}}^{(4)} = \underline{\mathbf{P}}(\{1, 2\}) \times_3 \tilde{\mathbf{R}}^{(3)} \quad (3.43)$$

$$\underline{\mathbf{Y}} = \underline{\mathbf{U}}^{(1)} \times_1 \tilde{\mathbf{R}}^{(1)}. \quad (3.44)$$

すなわち二進木アルゴリズムのように並行して計算することができる。この場合、テンソル-行列積は 16 から 9 ( $= 4 \log_2 4 + 1$ ) 回に減る。

### 3.3 TSOM-R<sup>M</sup> のアルゴリズム導出

本節では、TSOM アルゴリズムの理論的導出について述べる。まず最初に 2 つのテンソルの重みつき二乗誤差の表記を定義する。今、 $(M+1)$  次テンソル  $\underline{\mathbf{X}}$  と  $\underline{\mathbf{Y}}$  があるとする。そして負担率行列を  $\{\mathbf{R}\} = \{\mathbf{R}^{(1)}, \dots, \mathbf{R}^{(M)}\}$  とし、負担率が重みを与えるとする。このとき、 $\underline{\mathbf{X}}$  と  $\underline{\mathbf{Y}}$  の重みつき二乗誤差を次式で定義する。

$$E(\underline{\mathbf{X}}, \underline{\mathbf{Y}}; \{\mathbf{R}\}) \triangleq \sum_{n_1, \dots, n_M} \sum_{k_1, \dots, k_M} \left( \prod_{m=1}^M r_{k_m n_m}^{(m)} \right) \|\mathbf{x}_{n_1 \dots n_M} - \mathbf{y}_{k_1 \dots k_M}\|^2. \quad (3.45)$$

もし  $\mathbf{R}^{(m)}$  が  $\sum_k r_{kn}^{(m)} = 1$  を満たす場合、(3.45) は期待値誤差を表す。さらにモード  $m$  を除いた重みつき二乗誤差を次のように定義する。

$$E_{k_m n_m}^{(-m)}(\underline{\mathbf{X}}, \underline{\mathbf{Y}}; \{\mathbf{R}\}) \triangleq \sum_{\substack{n_1, \dots, n_M \\ \text{expect } n_m \text{ except } k_m}} \sum_{\substack{k_1, \dots, k_M \\ \text{except } m}} \left( \prod_{\substack{m'=1 \\ \text{except } m}}^M r_{k_m n_m}^{(m)} \right) \|\mathbf{x}_{n_1 \dots n_M} - \mathbf{y}_{k_1 \dots k_M}\|^2. \quad (3.46)$$

ここで  $\mathbf{E}^{(-m)}(\underline{\mathbf{X}}, \underline{\mathbf{Y}}; \{\mathbf{R}\})$  は  $K_m \times N_m$  行列になることに注意する。

以上の重みつき二乗誤差の表記を用いると、TSOM の目的関数は次式のように書くことができる。

$$F \triangleq -\frac{1}{2N} E(\underline{\mathbf{X}}, \underline{\mathbf{Y}}; \{\mathbf{R}\}). \quad (3.47)$$

ここで  $N = \prod_m N_m$  であり、また  $r_{kn}^{(m)} = h_{kk_n^{*(m)}}^{(m)}$  である。これは通常の SOM の目的関数 (2.23) をそのまま  $M$  次に拡張した形になっている。ここで (3.47) をモード  $m$  について分解する。ピタゴラスの定理により、

$$F = -\frac{1}{2N_m} \sum_{n=1}^{N_m} \sum_{k=1}^{K_m} r_{kn}^{(m)} \left\{ \frac{1}{K_{-m}} \left\| \underline{\mathbf{U}}_{n(m)}^{(m)} - \underline{\mathbf{Y}}_{k(m)} \right\|^2 + \frac{1}{N_{-m}} E_{k_m n_m}^{(-m)} (\underline{\mathbf{X}}, \underline{\mathbf{U}}^{(m)}; \{\mathbf{R}\}) \right\} \quad (3.48)$$

となる。ここで  $N_{-m} = \prod_{m' \neq m} N_{m'}$ ,  $K_{-m} = \prod_{m' \neq m} K_{m'}$  であり、 $\underline{\mathbf{U}}^{(m)}$  は (3.31) で与えられるインスタンス多様体である。

$M = 3$  でモード 1 について分解したときの (3.47) から (3.48) までの具体的な導出を記述しておく。まず目的関数 (3.47) は

$$F = -\frac{1}{2N_1 N_2 N_3} \sum_{n_1, n_2, n_3} \sum_{k_1, k_2, k_3} r_{k_1 n_1}^{(1)} r_{k_2 n_2}^{(2)} r_{k_3 n_3}^{(3)} \left\| \mathbf{x}_{n_1 n_2 n_3} - \mathbf{y}_{k_1 k_2 k_3} \right\|^2 \quad (3.49)$$

となる。そしてピタゴラスの定理が成り立つよう、

$$F = -\frac{1}{2N_1 N_2 N_3} \sum_{n_1, n_2, n_3} \sum_{k_1, k_2, k_3} r_{k_1 n_1}^{(1)} r_{k_2 n_2}^{(2)} r_{k_3 n_3}^{(3)} \left( \left\| \mathbf{x}_{n_1 n_2 n_3} - \mathbf{u}_{n_1 k_2 k_3}^{(1)} \right\|^2 - \left\| \mathbf{u}_{n_1 k_2 k_3}^{(1)} - \mathbf{y}_{k_1 k_2 k_3} \right\|^2 \right) \quad (3.50)$$

となる  $\mathbf{U}^{(1)}$  を用いる。この式をモード 1 について整理すると、

$$F = -\frac{1}{2N_1} \sum_{n_1} \sum_{k_1} r_{k_1 n_1}^{(1)} \left\{ \frac{1}{N_2 N_3} \sum_{n_2, n_3} \sum_{k_2, k_3} r_{k_2 n_2}^{(2)} r_{k_3 n_3}^{(3)} \left( \left\| \mathbf{u}_{n_1 k_2 k_3}^{(1)} - \mathbf{y}_{k_1 k_2 k_3} \right\|^2 + \left\| \mathbf{x}_{n_1 n_2 n_3} - \mathbf{u}_{n_1 k_2 k_3}^{(1)} \right\|^2 \right) \right\} \quad (3.51)$$

となる. (3.46) より

$$F = -\frac{1}{2N_1} \sum_{n_1} \sum_{k_1} r_{k_1 n_1}^{(1)} \\ \left\{ \frac{1}{N_2 N_3} \sum_{n_2, n_3} \sum_{k_2, k_3} r_{k_2 n_2}^{(2)} r_{k_3 n_3}^{(3)} \left\| \mathbf{u}_{n_1 k_2 k_3}^{(1)} - \mathbf{y}_{k_1 k_2 k_3} \right\|^2 + \frac{1}{N_2 N_3} E_{k_1 n_1}^{(-1)} (\underline{\mathbf{X}}, \underline{\mathbf{U}}^{(1)}; \{\mathbf{R}\}) \right\} \quad (3.52)$$

と書き換えられる. さらに  $\sum_{n_2} r_{k_2 n_2}^{(2)} \simeq \frac{N_2}{K_2}$ ,  $\sum_{n_3} r_{k_3 n_3}^{(3)} \simeq \frac{N_3}{K_3}$  と近似できるとすると,

$$F = -\frac{1}{2N_1} \sum_{n_1} \sum_{k_1} r_{k_1 n_1}^{(1)} \\ \left\{ \frac{1}{N_2 N_3} \sum_{k_2, k_3} \frac{N_2}{K_2} \frac{N_3}{K_3} \left\| \mathbf{u}_{n_1 k_2 k_3}^{(1)} - \mathbf{y}_{k_1 k_2 k_3} \right\|^2 + \frac{1}{N_2 N_3} E_{k_1 n_1}^{(-1)} (\underline{\mathbf{X}}, \underline{\mathbf{U}}^{(1)}; \{\mathbf{R}\}) \right\} \\ = -\frac{1}{2N_1} \sum_{n_1} \sum_{k_1} r_{k_1 n_1}^{(1)} \left\{ \frac{1}{K_{-1}} \left\| \mathbf{U}_{n_1(1)}^{(1)} - \mathbf{Y}_{k_1(1)} \right\|^2 + \frac{1}{N_{-1}} E_{k_1 n_1}^{(-1)} (\underline{\mathbf{X}}, \underline{\mathbf{U}}^{(1)}; \{\mathbf{R}\}) \right\} \quad (3.53)$$

となり, (3.48) が導出される.

さて目的関数 (3.47) をもとにアルゴリズムを導出する. E ステップでは (3.47) を最大化するように勝者  $\{k_n^{*(m)}\}$  を総当たりで求める.

$$k_n^{*(m)} = \arg \min_k \sum_{k'} h_{kk'}^{(m)} E_{k'n_m}^{(-m)} (\underline{\mathbf{X}}, \underline{\mathbf{Y}}; \{\mathbf{R}\}). \quad (3.54)$$

ここで (3.48) を適用する. (3.54) の第二項は勝者決定に関して影響しないため,

$$k_n^{*(m)} = \arg \min_k \sum_{k'} h_{kk'}^{(m)} \left\| \underline{\mathbf{U}}_{n(m)}^{(m)} - \underline{\mathbf{Y}}_{k'(m)} \right\|^2 \quad (3.55)$$

となる. ここで (2.24) から (2.14) への近似と同じことをすれば, (3.54),(3.55) は

$$k_n^{*(m)} = \arg \min_k E_{kn}^{(-m)} (\underline{\mathbf{X}}, \underline{\mathbf{Y}}; \{\mathbf{R}\}) \quad (3.56)$$

$$= \arg \min_k \left\| \underline{\mathbf{U}}_{n(m)}^{(m)} - \underline{\mathbf{Y}}_{k(m)} \right\|^2 \quad (3.57)$$

となる. すなわち (3.25) が得られる. (3.56),(3.57) はどちらもほとんど同じ結果を与えるが, (3.57) のほうが (3.56) よりも速く計算できる.

M ステップでは (3.47) を  $\underline{\mathbf{Y}}$  について最大化することで (3.30) が得られる. これは SOM と同様に  $\frac{\partial F}{\partial \underline{\mathbf{Y}}} = 0$  となる  $\underline{\mathbf{Y}}$  を求めればよい.

### 3.4 基底関数型 TSOM-R<sup>M</sup>

テンソルの次数  $M$  が大きくなると、計算量も莫大になる。各モードのノード数をすべて  $K$  としたとき、TSOM-R<sup>M</sup> が解くべきタスクのサイズはおおまかにいえば  $K^M$  に比例して増大する。計算コストを減らすためには、タスクそのものの縮小とアルゴリズム効率化の双方が必要である。

もっとも簡単なタスクの縮小方法は各モードのノード数  $K$  を減らすことである。しかしながら、単に  $K$  を減少させただけでは写像を表現する参照ベクトルの数も減少してしまい、写像の表現が粗くなってしまう。すなわち潜在空間および写像の解像度が犠牲となり、あまり良い解決法とはいえない。

タスクサイズを減らすもうひとつの方法は基底関数を用いることである。TSOM-R<sup>M</sup> では滑らかな写像を仮定しているため、ここでは比較的少ない基底関数を用いて非線形写像  $f$  を表現できると考える。すなわち各モードそれぞれに  $J$  ( $J < K$ ) 個の基底関数集合  $\left\{ \phi_1^{(m)}(\mathbf{z}^{(m)}), \dots, \phi_J^{(m)}(\mathbf{z}^{(m)}) \right\}$  を用意し、さらに  $K$  個の離散ノードに対する基底関数の値を行列  $\Phi^{(m)} = (\phi_j^{(m)}(\zeta_k^{(m)})) \in \mathbb{R}^{K \times J}$  として写像を  $\underline{\mathbf{Y}} \simeq \underline{\mathbf{W}} \times \{\Phi\}$  と十分な精度で表現できるとする。これにより  $\underline{\mathbf{Y}} \in \mathbb{R}^{K^M \times D}$  というように離散ノードによって表現されていた写像の情報は係数テンソル  $\underline{\mathbf{W}} \in \mathbb{R}^{J^M \times D}$  によって表現できるようになる。したがって  $\underline{\mathbf{Y}}$  は  $\underline{\mathbf{W}}$  へと  $(J/K)^M$  倍に圧縮されたことになる。またこのとき、第  $m$  モードのインスタンス多様体も  $\underline{\mathbf{U}}^{(m)} = \underline{\mathbf{V}}^{(m)} \times_{-m} \{\Phi\}$  となる係数テンソル  $\underline{\mathbf{V}}^{(m)}$  を用いて  $(J/K)^{M-1}$  倍に圧縮表現できる。基底関数を用いた TSOM-R<sup>m</sup> のアルゴリズムの M ステップでは写像  $\underline{\mathbf{Y}}$  とインスタンス多様体  $\{\underline{\mathbf{U}}\}$  を求める必要がなく、その圧縮表現である  $\underline{\mathbf{W}}$  と  $\{\underline{\mathbf{V}}\}$  を求めるだけでよい。

さらに基底関数として正規直交な基底を用いると、そのメリットはさらに増える。正規直交変換は距離を保存するため、2つの写像間の距離は2つの係数行列の距離と等しくなる。このことについて説明する。まず2つの写像を  $f_1(\mathbf{z}), f_2(\mathbf{z})$  とすると、写像間距離は

$$D^2(f_1, f_2) = \int \|f_1(\mathbf{z}) - f_2(\mathbf{z})\|^2 p(\mathbf{z}) d\mathbf{z} \quad (3.58)$$

である。ここで  $p(\mathbf{z})$  が一様分布であり、かつ潜在空間を  $K$  個のノードに離散化して各ノードの写像を  $\mathbf{y}_{1k} \triangleq f_1(\zeta_k), \mathbf{y}_{2k} \triangleq f_2(\zeta_k)$  とすることで写像全体  $f_1, f_2$  をそれぞれ

$\mathbf{Y}_1 = (\mathbf{y}_{1k})$ ,  $\mathbf{Y}_2 = (\mathbf{y}_{2k})$  と表現した場合,

$$D^2(f_1, f_2) \simeq \sum_{k=1}^K \|\mathbf{y}_{1k} - \mathbf{y}_{2k}\|^2 = \|\mathbf{Y}_1 - \mathbf{Y}_2\|^2 \quad (3.59)$$

となる。さらに正規直交基底関数を用いて写像をそれぞれ  $f_1(\zeta_{k:}) = \mathbf{W}_1^T \phi(\zeta_{k:})$ ,  $f_2(\zeta_{k:}) = \mathbf{W}_2^T \phi(\zeta_{k:})$  と圧縮表現できる場合,

$$D^2(f_1, f_2) \simeq \|\mathbf{Y}_1 - \mathbf{Y}_2\|^2 = \|\mathbf{W}_1 - \mathbf{W}_2\|^2 \quad (3.60)$$

となる。(3.60) の右辺は(3.59)の右辺よりも  $(J/K)$  倍の計算で済む。以上の内容を TSOM-R<sup>m</sup> に適用する。具体的には E ステップで用いられるスライス多様体  $\underline{\mathbf{Y}}_{k(m)}$  とインスタンス多様体  $\underline{\mathbf{U}}_{n(m)}^{(m)}$  を正規直交な基底を用いて圧縮表現することで、勝者決定の計算コストを削減する。

今、モード  $m$  に関する微分可能な正規直交系を  $\{\phi_1^{(m)}(\mathbf{z}^{(m)}), \dots, \phi_J^{(m)}(\mathbf{z}^{(m)})\}$  とする。そして第  $m$  モードの  $K$  個の離散ノードに対する基底関数の値を行列  $\Phi^{(m)} = (\phi_j^{(m)}(\zeta_k^{(m)})) \in \mathbb{R}^{K \times J}$  とする。すると  $\underline{\mathbf{Y}}$  は  $\underline{\mathbf{W}}$  から次式で得ることができる。

$$\underline{\mathbf{Y}} \simeq \underline{\mathbf{W}} \times_1 \Phi^{(1)} \times_2 \cdots \times_M \Phi^{(M)} = \underline{\mathbf{W}} \times \{\Phi\}. \quad (3.61)$$

このように圧縮表現されていた  $\underline{\mathbf{W}}$  から離散ノード表現  $\underline{\mathbf{Y}}$  を求める処理をここでは解凍と呼ぶ。第  $m$  モードのスライス多様体の圧縮表現  $\underline{\mathbf{T}}^{(m)}$  は、写像の圧縮表現  $\underline{\mathbf{W}}$  からモード  $m$  についてのみ解凍することで得られる。すなわち

$$\underline{\mathbf{T}}^{(m)} \triangleq \underline{\mathbf{W}} \times_m \Phi^{(m)} \quad (3.62)$$

である。一方モード  $m$  に関するインスタンス多様体の圧縮表現とは,

$$\underline{\mathbf{U}}^{(m)} = \underline{\mathbf{V}}^{(m)} \times_{-m} \{\Phi\} \quad (3.63)$$

となる係数テンソル  $\underline{\mathbf{V}}^{(m)}$  のことである。 $\underline{\mathbf{T}}^{(m)}$  および  $\underline{\mathbf{V}}^{(m)}$  はどちらもモード  $m$  以外すべてについて圧縮しているため、もともとの  $\underline{\mathbf{Y}}_{k(m)}$ ,  $\underline{\mathbf{U}}^{(m)}$  と比べてサイズが  $(J/K)^{M-1}$  倍になっている。それにもかかわらず、 $\{\Phi\}$  が正規直交であるため距離は保存されている。したがって,

$$\left\| \underline{\mathbf{U}}_{n(m)}^{(m)} - \underline{\mathbf{Y}}_{k(m)} \right\|^2 = \left\| \underline{\mathbf{V}}_{n(m)}^{(m)} - \underline{\mathbf{T}}_{k(m)}^{(m)} \right\|^2 \quad (3.64)$$

となる。(3.64)を(3.25)に適用することで、Mステップで求めた  $\underline{\mathbf{W}}$  や  $\{\underline{\mathbf{V}}\}$  を全モードについて解凍することなしに勝者を決定できる。さらに  $K^{M-1} \times D$  次元ベクトル間の距

離を  $J^{M-1} \times D$  次元係数ベクトルの距離で見積もっているため、E ステップにおける計算コストを減らすことができる。以上の正規直交な基底を用いることによる E ステップの効率化は基底関数型 SOM の段階では表れないが、基底関数型 TSOM-R<sup>M</sup> のアルゴリズム効率化においては重要な利点となっている。

アルゴリズムの残りの部分は、各基底関数に対する負担率  $\{\tilde{\mathbf{Q}}\}$  の算出および M ステップの更新式である。モード  $m$  の基底関数に対する負担率  $\tilde{\mathbf{Q}}^{(m)}$  は SOM の (2.31) と同様に次式で定義される。

$$\tilde{\mathbf{Q}}^{(m)} \triangleq \Phi_{\mathbf{G}^{(m)}}^{\#(m)} \tilde{\mathbf{R}}^{(m)} = \left( \Phi^{(m)\top} \mathbf{G}^{(m)} \Phi^{(m)} \right)^{-1} \Phi^{(m)\top} \mathbf{R}^{(m)} \quad (3.65)$$

M ステップでは得られた  $\tilde{\mathbf{Q}}^{(m)}$  を用いて  $\underline{\mathbf{W}}$ ,  $\underline{\mathbf{V}}$  を更新する。つまり基底関数型 TSOM-R<sup>M</sup> では  $\underline{\mathbf{W}}$ ,  $\underline{\mathbf{V}}$  を得るために、計算ループごとにすべてのモードについて的一般化逆行列  $\{\Phi_{\mathbf{G}}^{\#}\}$  を求める必要がある。逆行列は計算コストがかかるため、せっかく正規直交基底を用いたメリットが相殺される恐れがある。しかしながら 2 章でも述べたように、サンプルの数が十分であり事前分布が一様分布であれば  $\mathbf{G}^{(m)}$  は  $\frac{N_m}{K_m} \mathbf{I}$  で近似が可能であり、 $\Phi_{\mathbf{G}^{(m)}}^{\#(m)}$  は通常の一般化逆行列  $\Phi^{\#(m)}$  と近似することができる。そして  $\Phi^{\#(m)}$  は正規直交系の場合、単なる転置行列  $\Phi^{(m)\top}$  となる。したがって逆行列を計算する必要性がまったくなくなる。

また微分可能な基底関数を用いるもうひとつのメリットは、目的関数を潜在変数  $\mathbf{z}$  で微分できるようになるということである。すなわち勝者位置を勾配法などのアルゴリズムにより決定することが可能になる。この段階で離散ノードはもはや完全に不要となり、潜在変数を高い解像度で求めつつ計算コストも削減できることになる<sup>\*1</sup>。

以上をふまえると、基底関数型 TSOM のアルゴリズムは以下のようにかける。

### E ステップ

(3.64) より、勝者はモード  $m$  以外を解凍することなく求めることができる。すなわち

$$k_n^{*(m)} = \arg \min_k \left\| \underline{\mathbf{V}}_{n(m)}^{(m)} - \underline{\mathbf{T}}_{k(m)}^{(m)} \right\|^2 \quad (3.66)$$

---

<sup>\*1</sup> 厳密には負担率を算出する際に区分積分法を用いており、その点では代表点表現は用いている。ここで重要なのは、ニューラルネットワークを起源にもつ SOM には欠かせないノード表現が取り扱われているという点である。

とする。これは総当たり式の勝者決定法を用いた場合である。もし勾配法を使うのであれば更新式は

$$\mathbf{z}_n^{(m)\text{new}} := \mathbf{z}_n^{(m)\text{old}} - \eta \frac{\partial E_n^{(m)}(\mathbf{z}^{(m)})}{\partial \mathbf{z}_n^{(m)\text{old}}} \quad (3.67)$$

となる。ここで  $E_n^{(m)}(\mathbf{z}^{(m)}) \triangleq \left\| \underline{\mathbf{V}}_{n(m)}^{(m)} - \underline{\mathbf{W}} \times_m \boldsymbol{\phi}^{(m)}(\mathbf{z}^{(m)}) \right\|^2$  である。

勝者が決まった後は、負担率行列  $\mathbf{R}^{(m)}$ 、負担率の総和行列  $\mathbf{G}^{(m)}$  および規格化した負担率行列  $\tilde{\mathbf{R}}^{(m)}$  を求める。これは (3.26)-(3.29) と同じである。そして各基底関数に対する負担率行列  $\tilde{\mathbf{Q}}^{(m)}$  を次式で計算する。

$$\tilde{\mathbf{Q}}^{(m)} = \boldsymbol{\Phi}^{(m)\text{T}} \tilde{\mathbf{R}}^{(m)}. \quad (3.68)$$

ここで転置行列  $\boldsymbol{\Phi}^{(m)\text{T}}$  は一般化逆行列  $\boldsymbol{\Phi}_{\mathbf{G}^{(m)}}^{\#(m)}$  の近似として用いている。近似しない厳密な計算は (3.65) である。

### M ステップ

M ステップでは、写像の係数テンソル  $\underline{\mathbf{W}}$  とインスタンス多様体の係数テンソル  $\{\underline{\mathbf{V}}\}$  を更新する。

$$\underline{\mathbf{W}} = \underline{\mathbf{X}} \times \left\{ \tilde{\mathbf{Q}} \right\} \quad (3.69)$$

$$\underline{\mathbf{V}}^{(m)} = \underline{\mathbf{X}} \times_{-m} \left\{ \tilde{\mathbf{Q}} \right\}. \quad (3.70)$$

そして、次の E ステップのためにスライス多様体の係数テンソルを計算しておく（なおこの計算は E ステップで勾配法を用いる場合は不要である。）

$$\underline{\mathbf{T}}^{(m)} = \underline{\mathbf{W}} \times_m \boldsymbol{\Phi}^{(m)}. \quad (3.71)$$

以上が基底関数型 TSOM-R<sup>M</sup> のアルゴリズムである。

基底関数を用いない場合の更新式 (3.30), (3.31) との違いは  $\tilde{\mathbf{R}}^{(m)}$  が  $\tilde{\mathbf{Q}}^{(m)}$  が置き換えたことだけである。よってここでも二進木による同時計算は有効であり、高速化することができる。

4.1 節のシミュレーション結果 (表 4.1, 4.2) では基底関数により計算が高速化されたことを示した。なお本論文では基底関数型 TSOM-R<sup>M</sup> を用いる際、正方形潜在空間の一様分布に対する正規直交基底として正規化 Legendre 多項式を用いている。

## 3.5 初期化とスケジューリング

実際に TSOM-R<sup>M</sup> をアルゴリズムを実装して動作させるときには、SOM と同様に初期化の方法と近傍半径の縮小スケジューリングの方法について決めておく必要がある。

乱数による初期化の場合、写像  $\underline{Y}$  とインスタンス多様体  $\{\underline{U}\}$  を乱数によって初期化する方法と、勝者ノード  $\{k_n^{*(m)}\}$  を乱数によって初期化する方法の 2 つがある。前者の場合は E ステップより学習を開始し、後者は M ステップから学習を開始する。どちらでも大きな違いはないものの、後者はデータ分布のバイアス等に影響されないという利点がある。ただしどちらを用いても乱数による初期値は困難な初期状態であり、困難な初期条件から正しく学習ができるかという頑健性のテストには適しているものの、実際に応用に用いる際には線形手法の結果などを初期値にした方がよい。なお、本論文の TSOM のシミュレーションでは乱数による勝者ノード初期化を用いている。

既存手法を初期値として用いる場合も、勝者ノードの初期化と写像の初期化の 2 つが考えられる。前者の場合、たとえば各モードに関するスライスをベクトルとみなして PCA をかける。すなわちモード  $m$  に関して  $\hat{\mathbf{x}}_n^{(m)} = \text{vec}(\underline{\mathbf{X}}_{n(m)})$  を第  $n$  データベクトルとみなして PCA を実行し、主軸変換した座標を用いて初期の勝者ノードを決めるのである。写像を初期化して始める場合は Tucker 分解のコアテンソルを用いて導出するとよい。

近傍半径の縮小スケジューリングも SOM と同様である。たとえば計算回数  $t$  について、

$$\sigma(t) = \max [\sigma_0(1 - t/\tau), \sigma_\infty] \quad (3.72)$$

のように線形に減少させるか、あるいは

$$\sigma(t) = (\sigma_0 - \sigma_\infty) \exp(-t/\tau) + \sigma_\infty \quad (3.73)$$

のよう指数的に縮小させる。ここで  $\sigma_0$  が初期値、 $\sigma_\infty$  が終値である。また  $\tau$  は縮小スピードを決める時定数である。なお時定数や最小近傍半径などのパラメータはモードごとに別々に設定してもよい。本論文のシミュレーションでは全モードで同一のパラメータを用いており、 $\sigma(t)$  については (3.73) を用いている。

## 3.6 Tensor GTM: TGTM

確率的生成モデルを用いることで、完全にベイズ的アプローチでアルゴリズムを導出することもできる。本論文では SOM のベイズ理論モデルである GTM をベースに導出

し，これを Tensor GTM (TGTM) とよぶ。また  $M$  モード関係データ用の TGTM を TGTM-R $^M$  (Tensor GTM for M-mode Relational data) と書くこととする。

まず TGTM-R $^M$  の扱う関係データの生成モデルを定義する。今， $M$  個の潜在空間  $\{\mathcal{Z}^{(m)}\}$  ( $m = 1, \dots, M$ ) があり，それぞれ  $K_m$  ノードに離散化されているとする。またモード  $m$  の各ノードの潜在空間における位置ベクトルを  $\{\zeta_k^{(m)}\}$  とする。潜在変数の事前分布は  $\text{Prob}(\mathbf{z}^{(m)} = \zeta_k^{(m)}) = p_k^{(m)}$  とするが，以降では  $p_k^{(m)} = 1/K_m$  と仮定する。観測データを得るために，まず潜在変数を各モードについて  $N_m$  回独立に生成する。そして生成した潜在変数を各モードからひとつずつ取り出し，次式の確率モデルにより観測データを得る。

$$\mathbf{x}_{n_1 \dots n_M} = \mathbf{y} \left( \mathbf{z}_{n_1}^{(1)}, \dots, \mathbf{z}_{n_M}^{(M)} \right) + \boldsymbol{\varepsilon}_{n_1 \dots n_M}. \quad (3.74)$$

ここで  $\mathbf{y}$  は潜在空間の直積空間  $\mathcal{Z}^{(1)} \times \dots \times \mathcal{Z}^{(M)}$  から観測空間  $\mathcal{X} \in \mathbb{R}^D$  への非線形写像であり， $\boldsymbol{\varepsilon} \in \mathbb{R}^D$  は  $p(\boldsymbol{\varepsilon}) = \mathcal{N}(\boldsymbol{\varepsilon} | \mathbf{0}, \beta^{-1} \mathbf{I}_D)$  に従う観測ノイズである。この観測は潜在変数のすべての組み合わせ，すなわち  $\prod_m N_m$  通りすべてに対して行う。したがって観測データは  $(M+1)$  階のデータテンソル  $\underline{\mathbf{X}} = (x_{n_1 \dots n_M d})$  として表される。

一方，非線形写像は RBF 基底を用いて表現できるとする。すなわち

$$\underline{\mathbf{Y}} = \underline{\mathbf{W}} \times \{\Phi\} \quad (3.75)$$

とする。ここで  $\Phi^{(m)} \in \mathbb{R}^{K_m \times J_m}$  がモード  $m$  について離散化されたノードの潜在変数  $\{\zeta_k^{(m)}\}$  に対する RBF 基底の値であり， $\underline{\mathbf{W}}$  が係数テンソルである。 $\underline{\mathbf{W}}$  も確率的に生成され，その事前分布は，

$$p\{\underline{\mathbf{W}}\} = \mathcal{N}(\underline{\mathbf{W}} | \underline{\mathbf{O}}^{-1}, \underline{\mathbf{L}}) \quad (3.76)$$

$$\underline{\mathbf{L}} = (\lambda_1 \mathbf{I}_{K_1}) \circ \dots \circ (\lambda_M \mathbf{I}_{K_M}) \quad (3.77)$$

とする。ここで  $\circ$  は外積を表す。たとえば  $\underline{\mathbf{X}} \in \mathbb{R}^{K_1 \times \dots \times K_M}$  と  $\underline{\mathbf{Y}} \in \mathbb{R}^{J_1 \times \dots \times J_N}$  に對して，外積  $\underline{\mathbf{Z}} = \underline{\mathbf{X}} \circ \underline{\mathbf{Y}}$  は  $\underline{\mathbf{Z}} \in \mathbb{R}^{K_1 \times \dots \times K_M \times J_1 \times \dots \times J_N}$  となる。ここで  $z_{\mathbf{k}\mathbf{j}} = x_{\mathbf{k}} y_{\mathbf{j}}$  である。(3.76) より  $\underline{\mathbf{W}}$  の各成分はそれぞれ独立のガウス事前分布  $p(w_{j_1 \dots j_M d}) = \mathcal{N}(w_{j_1 \dots j_M d} | 0, \lambda^{-1})$  に従う。 $M = 2$  の場合の生成モデルを図 3.2(a)，そのグラフィカルモデルを図 3.2(b) に示す。

TGTM-R $^M$  のタスクは係数テンソル  $\underline{\mathbf{W}}$  と潜在変数  $\{\mathbf{z}_{n_m}^{(m)}\}$ ，および潜在ノイズの分散パラメータ  $\beta$  を観測データ  $\underline{\mathbf{X}}$  から推定することである。TGTM-R $^M$  のアルゴリズムは標準的な EM アルゴリズムを適用することで導ける。ここでは  $\underline{\mathbf{W}}$  と  $\beta$  に関して MAP 解を求め，一方潜在変数  $\{\mathbf{z}_{n_m}^{(m)}\}$  について事後分布を求める場合のアルゴリズムを導出す

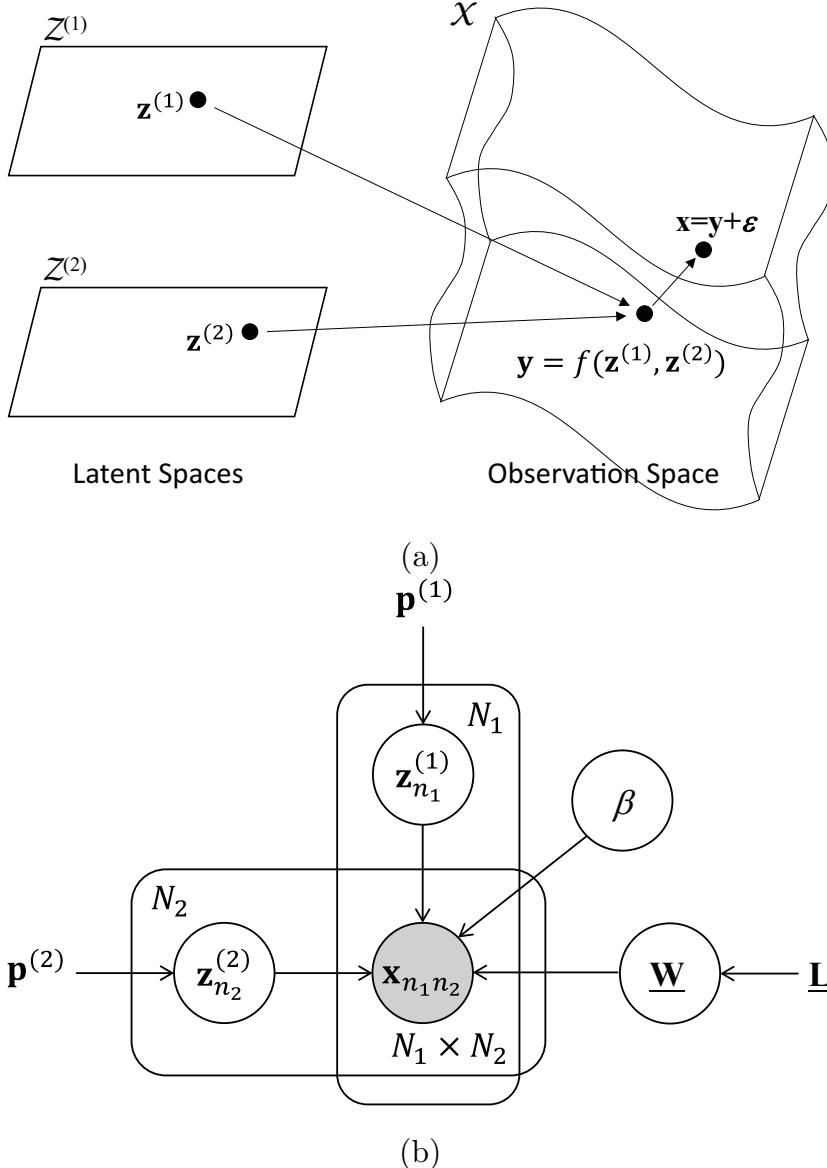


図 3.2 (a) TGTM-R<sup>2</sup> が扱う 2 モード関係データの生成モデル (非線形関数により関数空間に生成される積多様体は 4 次元であるが、図では 3 次元に描いている). (b) 生成モデルのグラフィカルモデル.

る。もし  $\underline{\mathbf{W}}, \beta$  についても事後分布を求めるのであれば、変分ベイズ法を用いることで容易にアルゴリズムを導出できる [40]。生成モデルに変分近似を適用すると、次式の目的関数が得られる。

$$F(\underline{\mathbf{R}}, \underline{\mathbf{W}}, \beta) = -\frac{\beta}{2} E(\underline{\mathbf{X}}, \underline{\mathbf{Y}}(\underline{\mathbf{W}}); \{\underline{\mathbf{R}}\}) - \sum_{\mathbf{k}} \sum_{\mathbf{n}} r_{\mathbf{k}\mathbf{n}} \ln r_{\mathbf{k}\mathbf{n}} \ln p(\underline{\mathbf{W}}) \quad (3.78)$$

ここで  $E(\underline{\mathbf{X}}, \underline{\mathbf{Y}}; \underline{\mathbf{R}})$  は (3.45) で定義される期待値誤差であり、 $\mathbf{R}^{(m)}$  は  $\{\mathbf{z}_{n_m}^{(m)}\}$  に関する

負担率行列である。Eステップでは  $F$  が  $\mathbf{R}^{(m)}$  に関して最大化される。

$$\ln r_{kn}^{(m)} = -\frac{\beta}{2} E_{kn}^{(-m)} (\underline{\mathbf{X}}, \underline{\mathbf{Y}}(\underline{\mathbf{W}}); \{\mathbf{R}\}) + const. \quad (3.79)$$

$M = 3$  の場合、(3.79) は

$$\ln r_{k_1 n_1}^{(1)} = -\frac{\beta}{2} \sum_{k_2, k_3} \sum_{n_2 n_3} r_{k_2 n_2}^{(2)} r_{k_3 n_3}^{(3)} \|\mathbf{y}_{k_1 k_2 k_3} - \mathbf{x}_{n_1 n_2 n_3}\|^2 + const. \quad (3.80)$$

として計算できる。一方 Mステップでは、 $F$  が  $\underline{\mathbf{W}}$  と  $\beta$  に関して最大化される。

$$\underline{\mathbf{W}} = \underline{\mathbf{X}} \times \{\tilde{\mathbf{Q}}\} \quad (3.81)$$

$$\beta^{-1} = \frac{1}{\prod_m N_m} E(\underline{\mathbf{X}}, \underline{\mathbf{Y}}(\underline{\mathbf{W}}); \{\mathbf{R}\}). \quad (3.82)$$

ここで  $\tilde{\mathbf{Q}}^{(m)}$  は

$$\mathbf{G}^{(m)} = \text{diag} \left( \sum_{n_m} r_{k_m n_m}^{(m)} \right) \quad (3.83)$$

$$\tilde{\mathbf{Q}}^{(m)} = \left( \Phi^{(m)\top} \mathbf{G}^{(m)} \Phi^{(m)} + \frac{\lambda_m}{\sqrt[M]{\beta}} \mathbf{I}_{K_m} \right)^{-1} \Phi^{(m)\top} \mathbf{R}^{(m)} \quad (3.84)$$

として計算する。以上が TGTM-R<sup>M</sup> のアルゴリズムである。以上の導出では RBF 基底を用いたが、ガウス過程を用いても同様のアルゴリズムを得ることができる。RBF 基底のかわりにガウスカーネル平滑化を用い、かつ潜在変数を MAP 推定に置き換えると、TGTM-R<sup>M</sup> のアルゴリズムは TSOM-R<sup>M</sup> のアルゴリズムになる。

TGTM-R<sup>M</sup> のアルゴリズムは完全にベイズ的なものであり、この点が TSOM-R<sup>M</sup> に対するアドバンテージとなる。しかしながら TGTM-R<sup>M</sup> の計算コストは TSOM-R<sup>M</sup> と比べてはるかに大きい。このことは 4.1 節のシミュレーション結果でも示している（表 4.1, 4.2）。したがって効率的な計算法を導入しない限り、TGTM-R<sup>M</sup> を実課題に用いるのは現実的ではない。本論文における TGTM-R<sup>M</sup> の意義は、ニューラルネットワークとしての TSOM-R<sup>M</sup> アルゴリズムとベイズ理論に基づくアルゴリズムの理論的橋渡しすることである。すなわち TGTM-R<sup>M</sup> という理論的根拠を与えることで、TSOM-R<sup>M</sup> を幅広い用途に安心して使えるようになるのである。

## 第 4 章

# 人工データを用いた TSOM の検証

本論文で提案したアルゴリズムの性能を調べるためにいくつかの人工データを用いて検証する。まず TSOM-R<sup>2</sup> と TGTM-R<sup>2</sup> について、計算速度や計算精度に関する性能比較を行う。次のシミュレーションとして、通常の TSOM-R<sup>2</sup> にさまざまな人工データを与え、期待する動作が行われているかを検証する。

本章の実験では写像の可視化を容易にする都合上、1 次元潜在空間をもつ 2 モード関係データを生成モデルに基づいて生成する。人工データの具体的な生成手順は以下のとおりである。まず第 1 モード、第 2 モードについてそれぞれ  $N_1$ ,  $N_2$  個の潜在変数  $\{z_1^{(1)}, \dots, z_{N_1}^{(1)}\}$ ,  $\{z_1^{(2)}, \dots, z_{N_2}^{(2)}\}$  を  $[-1, +1]$  上の一様分布から生成する。次に観測データ  $\mathbf{x}_{n_1 n_2}$  を次式により生成する。

$$\mathbf{x}_{n_1 n_2} = f(z_{n_1}^{(1)}, z_{n_2}^{(2)}) + \boldsymbol{\varepsilon}. \quad (4.1)$$

ここで  $\boldsymbol{\varepsilon}$  はガウスノイズ  $p(\boldsymbol{\varepsilon}) = \mathcal{N}(\boldsymbol{\varepsilon} \mid \mathbf{0}, \sigma_{\text{noise}}^2 \mathbf{I})$  である。この生成は  $z_{n_1}^{(1)}$ ,  $z_{n_2}^{(2)}$  の潜在変数のペアすべてについて行う。したがって  $N_1 \times N_2$  個の観測データをもつ人工データが生成される。また本シミュレーションではノイズ強度は  $\sigma_{\text{noise}} = 0.1$  とした。

### 4.1 TSOM および TGTM の性能比較

まずはじめに TSOM-R<sup>2</sup> と TGTM-R<sup>2</sup> の 4 つのバリエーションについて、計算時間と二乗平均平方根誤差 (Root Mean Square Error: RMSE) および得られる写像について比較を行った。比較対象は (1) 通常の TSOM-R<sup>2</sup> (1 モードあたりのノード数 20), (2) 基底関数型 TSOM-R<sup>2</sup> (1 モードあたりの基底の数 4, 基底は正規化 Legendre 基底), (3) TGTM-R<sup>2</sup> (1 モードあたりの基底の数 20, 基底は RBF 基底), (4) TGTM-R<sup>2</sup> (1 モー

表 4.1 人工データによるシミュレーションで使用したパラメータと計算結果

	TSOM-R <sup>2</sup> (20 nodes/mode)	Legendre TSOM-R <sup>2</sup> with 4 bases/mode	TGTM-R <sup>2</sup> with 20 bases/mode	TGTM-R <sup>2</sup> with 4 bases/mode
ノード数 ( $K_1, K_2$ )	$20 \times 20$			
基底の数 ( $J_1, J_2$ )	( $20 \times 20$ )	$4 \times 4$	$20 \times 20$	$4 \times 4$
近傍半径 ( $\sigma_0$ )	2.0	2.0	2.0	2.0
( $\sigma_\infty$ )	0.1	0.1	0.8	0.8
近傍関数の時定数 ( $\tau$ )	50			
データ数 ( $N_1, N_2$ )	$100 \times 100$			
ノイズ強度 ( $\sigma_{\text{noise}}$ )	0.1			
RMSE(20 試行平均)	$0.0775 \pm 0.0037$	$0.0867 \pm 0.0046$	$0.0693 \pm 0.0027$	$0.0777 \pm 0.0019$
計算時間 (sec/100 loops)	$0.45593 \pm 0.00083$	$0.09796 \pm 0.00027$	$15.3294 \pm 0.0057$	$10.9452 \pm 0.0038$

\* Intel Core i7-2600K (3.40GHz), Visual C++ (single thread)

ドあたりの基底の数 4, 基底は RBF 基底) である. 用いたパラメータを表 4.1 に示す. TSOM では勝者を乱数により初期化し, TGTM は写像を乱数により初期化した. なお通常の TSOM-R<sup>2</sup> では総当たり式の勝者決定法を用いたが, 一方基底関数型 TSOM-R<sup>2</sup> では最急降下法により潜在変数を更新した.

経験的に GTM は SOM よりも局所解に陥りやすく, 特に乱数によって初期化したときは顕著である. 本シミュレーションでは局所解を回避するために, SOM の近傍半径と同様に RBF の半径パラメータを徐々に小さくした. このアルゴリズム修正により, TGTM は本シミュレーションに関する限りでは局所解に陥ることがなかった.

近傍半径は時定数  $\tau = 50$  で近傍関数の式 (3.73) のように指数的に縮小させながら EM アルゴリズムを繰り返した. 本シミュレーションでは  $t = 600$  まで学習を進めたが, より少ない繰り返しで学習結果は収束していた. 潜在変数の配置が収束している場合, 近傍関数の式 (3.73) より  $\exp(-t/\tau) \simeq 0$  とみなせると写像の値は変化しなくなるため, 多くとも 300 回程度で十分である. なお近傍関数を線形で縮小させ, 亂数初期化ではなく PCA 初期化などを用いることで繰り返し回数をさらに減らせる.

本実験での人工データセットは, 3 次元観測データを次の関数で生成した.

$$f(z_1, z_2) = \begin{pmatrix} z_1 \cos \pi/4 - z_2 \sin \pi/4 \\ z_1 \sin \pi/4 + z_2 \sin \pi/4 \\ z_1^2 - z_2^2 \end{pmatrix}. \quad (4.2)$$

実際の入力となるデータと期待される写像, 各手法の写像の学習結果を図 4.1 に示す. 比較したすべてのアルゴリズムにおいて期待通りの結果が得られていることがわかる. 初期

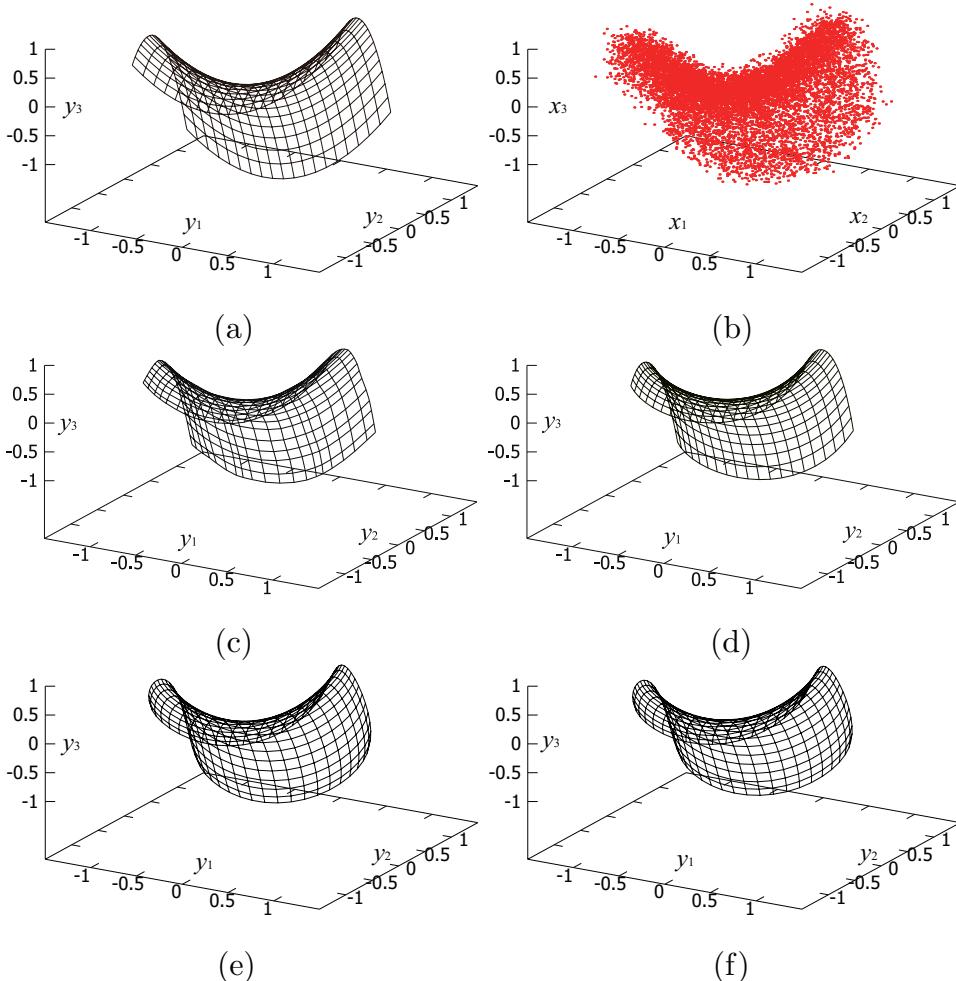


図 4.1 人工データによる学習結果. (a) オリジナルの写像 (期待される結果). (b) 入力となるガウスノイズが付加されたデータ. (c) 通常の TSOM-R<sup>2</sup> の学習結果. (d) 基底関数型 TSOM-R<sup>2</sup>(4×4 の Legendre 基底) の学習結果. (e) TGTM-R<sup>2</sup>(20×20 の RBF 基底) の学習結果. (f) TGTM-R<sup>2</sup>(4×4 の RBF 基底) の学習結果.

状態およびノイズをランダムに変えて 20 回実験を行ったが、結果はすべて一貫しており、かつ安定していた。

このデータセットに対しては、RMSE などのアルゴリズムでも同程度であった。RMSE および計算時間を表 4.1 に示す。比較したアルゴリズムのうち、もっとも計算速度が速かったのは基底関数型 TSOM-R<sup>2</sup> であり、通常の TSOM-R<sup>2</sup> の約 5 倍の速さだった。反対に TGTM-R<sup>2</sup> は基底関数型 TSOM-R<sup>2</sup> の 100 倍以上も遅かった。この速度差はタスクが大きくなるとさらに開く<sup>\*1</sup>。したがって大規模関係データ解析には基底関数型

<sup>\*1</sup> 表 4.2 では関係データのデータ数や潜在空間の解像度などによりタスクを大きくした場合を比較しているが、関係データの次数が増加するとさらに顕著にタスクが増加する。

表 4.2 大きなデータセットに対する計算時間比較

	TSOM-R <sup>2</sup>	基底関数型 TSOM-R <sup>2</sup>	TGTM-R <sup>2</sup>
ノード数 ( $K_1, K_2$ )	$400 \times 400$		
基底の数 ( $J_1, J_2$ )	( $400 \times 400$ )	$16 \times 16$	$16 \times 16$
データ数 ( $N_1, N_2$ )	$1,000 \times 1,000$		
計算時間 (sec/100 loops)	$8.45 \times 10^2$	$3.38 \times 10^1$	$3.91 \times 10^5$

\* Intel Core i7-2600K (3.40GHz), Visual C++ (single thread)

TSOM-R<sup>2</sup> がもっとも良い選択である。

## 4.2 TSOM の動作検証

次のシミュレーションとして、通常の TSOM-R<sup>2</sup> にさまざまな人工データを与えた。図 4.2(a) における赤点は人工データであり、黒の格子線は TSOM-R<sup>2</sup> の写像の学習結果である。このデータは次式により生成した。

$$f(z_1, z_2) = \begin{pmatrix} \cos(0.5\pi z_1 + \pi z_2) \\ \sin(0.5\pi z_2 + \pi z_2) \\ z_2 \end{pmatrix}. \quad (4.3)$$

通常の SOM ではこのような円筒形に巻かれたデータセットを学習するのは困難である。それにもかかわらず TSOM は学習に成功している。これは関係データの方が通常のデータよりも潜在変数の情報をより多くもっているためである。この点については 7 章で検討する。

3.2 節でも述べたように、TSOM は観測データの空間が 1 次元、すなわちスカラーの場合でもマップを推定することができる。図 4.2(b) はスカラーデータセットとその学習結果の例である。ここでデータは次式で生成した。

$$f(z_1, z_2) = \sin\left(\frac{3\pi}{4}(z_1 + z_2)\right). \quad (4.4)$$

この場合、2 次元の直積潜在空間は 1 次元の観測空間に次元が退化する。すなわち異なる潜在変数の組み合わせであってもしばしば同じ出力が得られる。それにもかかわらず TSOM-R<sup>2</sup> は非線形写像と潜在空間の推定に成功していることが図 4.2(b) により示された。

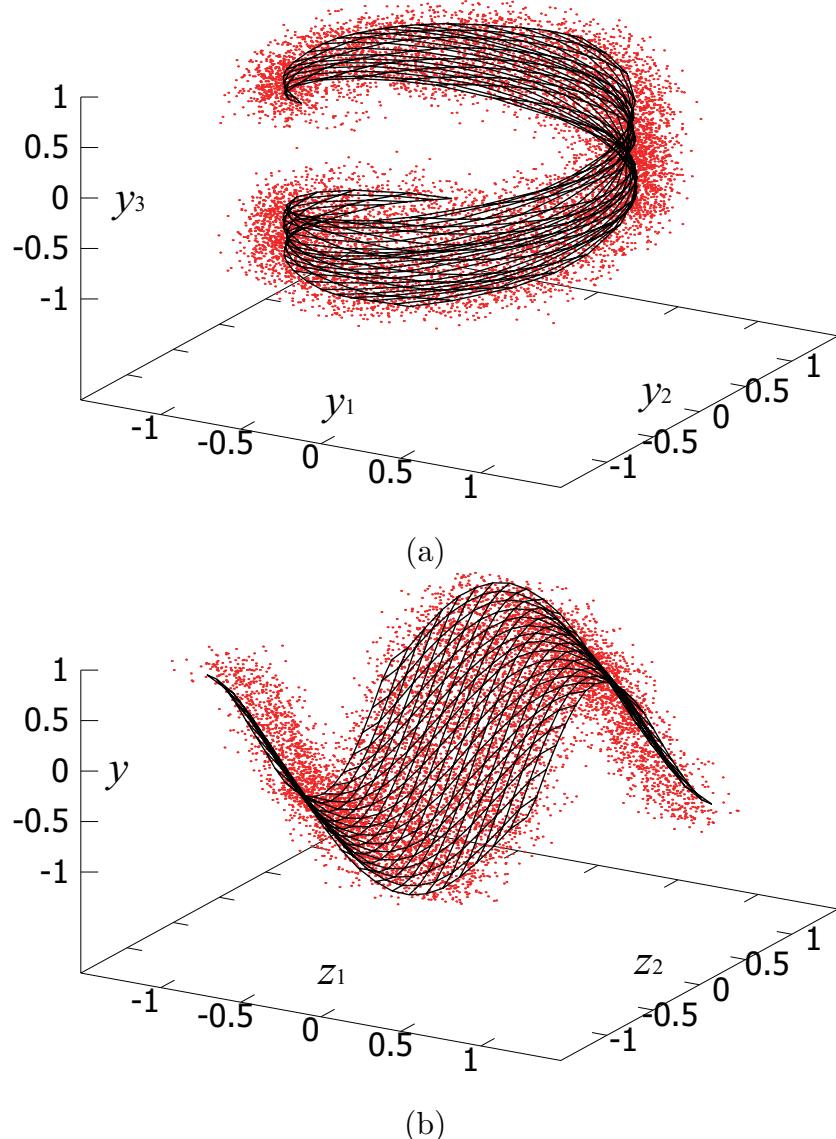


図 4.2 人工データに対して TSOM-R<sup>2</sup> が生成したマップ. データ点は赤いドットで示している. (a) 3 次元データセットと TSOM-R<sup>2</sup> を用いて生成したマップ. 座標軸は観測空間の 3 成分を表している. (b) 1 次元データセットと TSOM-R<sup>2</sup> を用いて生成したマップ. 他の図と異なり, 水平方向の 2 軸は潜在変数を表し, 垂直軸の 1 軸が観測空間を表す.



## 第5章

# TSOMによるアンケートデータ 解析

本章では実応用例として、2つのアンケートデータセットに TSOM-R を応用する。ひとつは寿司嗜好データセット（ユーザと商品の2モード関係データ）、もうひとつは飲料アンケートデータセット（ユーザと商品とコンテキストの3モード関係データ）である。またこれらの学習結果の説明を通して、本手法で関係データがどのように可視化されるかを解説する。なお以下の実験では通常の TSOM-R のみを用いたが、基底関数型 TSOM-R や TGTM-R についても同様の解析・可視化が可能である。

### 5.1 寿司に関するアンケートデータの解析

寿司嗜好データセット<sup>\*1</sup>は 5,000 人の回答者から得た 10 種の寿司のランキングスコアから構成されている [43]。本研究ではこの順序値をスコア値とみなして（1 位の寿司をスコア 10, 10 位の寿司をスコア 1 として順次変換させ），そのデータを用いた。

寿司データの学習結果を図 5.1 に示す。TSOM-R<sup>2</sup>により、ユーザ（回答者）と商品（寿司）の2つのマップが得られた。これらのマップ上での潜在変数の配置は通常の SOM と同様に解釈できる。すなわちマップ上で近いユーザ同士（もしくは寿司同士）は互いに似ており、マップ上で離れた 2 点は互いに似ていないと解釈できる。例えば図 5.1(b) の右上ではエビと鉄火巻が近くに配置されており、エビを好むユーザは鉄火巻を好む（またその逆もなりたつ）という傾向を読み取ることができる。なおユーザマップ上に 5,000 人の

---

<sup>\*1</sup> <http://www.kamishima.net/sushi/>

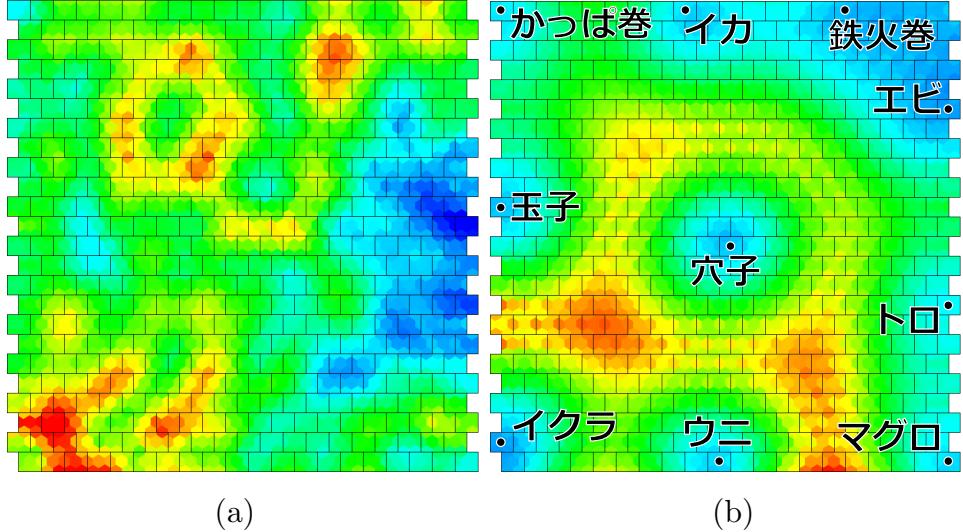


図 5.1 寿司データに対するマップ. (a) (b) ユーザ(回答者)マップおよび商品(寿司)マップ. Marginal U-matrix で彩色しており, 赤い領域はクラスタ境界を意味する.

ユーザラベルを配置すると煩雑になるため, ユーザマップには潜在変数の配置は表示していない.

また TSOM ではマップ上に色を付加することで非線形写像の情報を表すことができる. 今回用いた手法は SOM の色付け手法をベースに TSOM 向けに拡張したものである. 従来の SOM では, 写像の情報は通常は U-matrix もしくは Component Plane による色付けで表される [44, 45]. Component Plane は観測空間の特定のベクトル成分に着目して色付けしたものである. つまり着目した成分を  $d$  としたとき,  $k$  番ノードに  $y_{kd}$  の値を表示したものである. これにより着目成分に関して写像が大きい値を取る領域をマップ上で読み取ることができる. 一方 U-matrix では, 各ノードの写像先が隣接ノードの写像先とどれくらい異なるかに応じて色付けされる. 具体的には隣接するノード  $k$ ,  $k'$  間の距離を  $D(k, k') \triangleq \|y_k - y_{k'}\|$  とし, その値を元に色付けを行う. これにより非線形写像の微分の大きな領域をマップ上で読み取ることができる. その結果, クラスタ境界が可視化される. これら 2 つの可視化法を TSOM に適用する.

図 5.1 に示しているカラー表示は, SOM の可視化手法である U-matrix を TSOM 向けに改良して用いたものである. これを **Marginal U-matrix** とよぶ. TSOM の Marginal U-matrix ではモード  $m$  の潜在空間上で隣接するノード  $k$ ,  $k'$  間の距離を

$$D_m(k, k') \triangleq \|\underline{\mathbf{Y}}_{k(m)} - \underline{\mathbf{Y}}_{k'(m)}\| \quad (5.1)$$

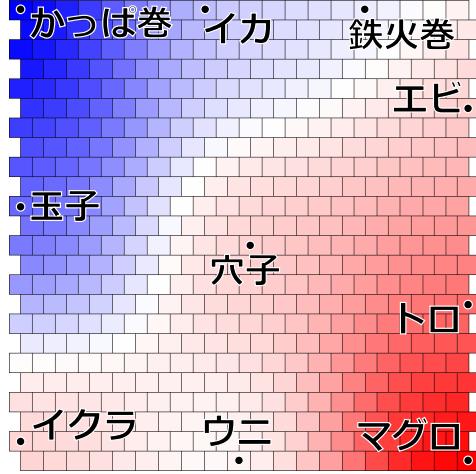


図 5.2 Marginal Component Plane で彩色した商品(寿司)マップ. 赤/青の領域はそれぞれ平均スコアの高/低を表す.

と定義して、これを元に色付けを行う. これにより各マップでそれぞれクラスタ境界を可視化することができる. 例えば図 5.1(b) を見ると左下にイクラとウニの軍艦巻きのクラスタ, 中央に穴子の独立クラスタがあり, 残りの領域でマグロ, トロ, エビ, 鉄火巻, イカ, かつば巻, 玉子と連続的に好みが変化していることが見て取れる.

全回答者の平均的傾向として好まれている寿司, 好まれていない寿司を図 5.2 のように可視化することもできる. これを **Marignal Component Plane** とよぶ. 着目する成分を  $d$  としたとき, 第  $m$  モードの第  $k$  ノードは  $m$  以外のモードに関する平均値によって色付けされる. すなわち,

$$\bar{y}_{kd}^{(m)} = \frac{1}{K-m} \sum_{k_1} \cdots \sum_{k_{m-1}} \sum_{k_{m+1}} \cdots \sum_{k_M} y_{k_1 \dots k_M d}. \quad (5.2)$$

である. 図 5.2 より最高スコアはマグロ, 最低スコアはかつば巻であることや, そのスコアが寿司マップ上で連続的に変化していることが見て取れる. なお本データは観測空間の次元が 1 次元であるため, 着目する成分を指定する必要はない.

以上の 2 つの可視化手法に共通する Marginal という単語は, 着目するモード以外の成分についてすべて周辺化(平均化)することに由来する. これらの手法を用いることで, 各モードのマップを SOM と同じように可視化することができる. これはモード内のインスタンス間の関係を可視化する場合に有効である.

さらに TSOM はモード間の情報も可視化して引き出すことができる. 今, 非線形写像  $\mathbf{y} = f(\mathbf{z}^{(1)}, \mathbf{z}^{(2)})$  をモード 1 に関して可視化したいとする. このときに  $f$  はモード 1 の潜

在変数  $\mathbf{z}^{(1)}$  の関数とみなし、モード 2 の潜在変数  $\mathbf{z}^{(2)}$  は特定の値を指定したものとする。すなわち  $f(\mathbf{z}^{(1)} | \mathbf{z}^{(2)})$  のように、いわば条件付きマップとして考える。そしてこのマップを U-matrix もしくは Component Plane で色付けする。この手法を **Conditional U-matrix, Conditional Component Plane** とよぶ。ここで Conditional とは、解析したいモードのマップを色付けする際に、他のモードに関して着目点を条件指定することを意味する。図 5.3 に寿司データの解析結果に Conditional Component Plane を用いた例を示した。なお Conditional の手法で着目点を連続的に変化させていくと、表示される色付けも連続的に変化する。また Marginal Component Plane を見ても、隣り合うノードの色は近い色となっている。これらの結果から TSOM が滑らかな非線形写像を獲得していることを示すことができる。

## 5.2 飲料に関するアンケートデータの解析

飲料データ<sup>\*2</sup>は 604 人の回答者から 14 種の飲料（オレンジジュース、コーラ等）を 11 の異なる場面（コンテキスト）で好むかを 5 段階評価でアンケート調査したものである。すなわち飲料データは（ユーザ）×（商品）×（コンテキスト）の 3 モード関係データである。

飲料データの学習結果を Marginal U-matrix を用いて図 5.4 に示す。TSOM-R<sup>3</sup> によってユーザ（回答者）マップ、商品（飲料）マップ、コンテキスト（状況）マップの 3 つのマップが得られている。こちらも似たタイプの飲料は商品マップ上で隣り合っており、似た状況も隣り合っていることが見て取れる。たとえば図 5.4(b) を見ると炭酸飲料のクラスタやフルーツジュースのクラスタ、お茶のクラスタがそれぞれ同一の領域に配置されていることが読み取ることができ、図 5.4(c) からは屋内と屋外という状況のクラスタ境界が強く表れているのがわかる。

モード数が 3 以上の場合、前節で述べた Marginal と Conditional の可視化を混在して使うことができる。すなわち第 1 のモードを可視化の対象として、第 2 のモードを条件指定として用い、第 3 のモードは周辺化する、などといったことができる。図 5.5 では、ユーザに関しては周辺化し、残りの 2 つのモードに関しては条件付けしてスコアを可視化した例である。モード数が大きくなると可視化の組み合わせも増大するため、得たい情報に合わせて柔軟に条件を切り替える表示法の開発が必要となる。

PC 上でインタラクティブなユーザインターフェースを使うと、柔軟に条件を切り替え

---

<sup>\*2</sup> <http://www.brain.kyutech.ac.jp/~furukawa/beverage-e/>

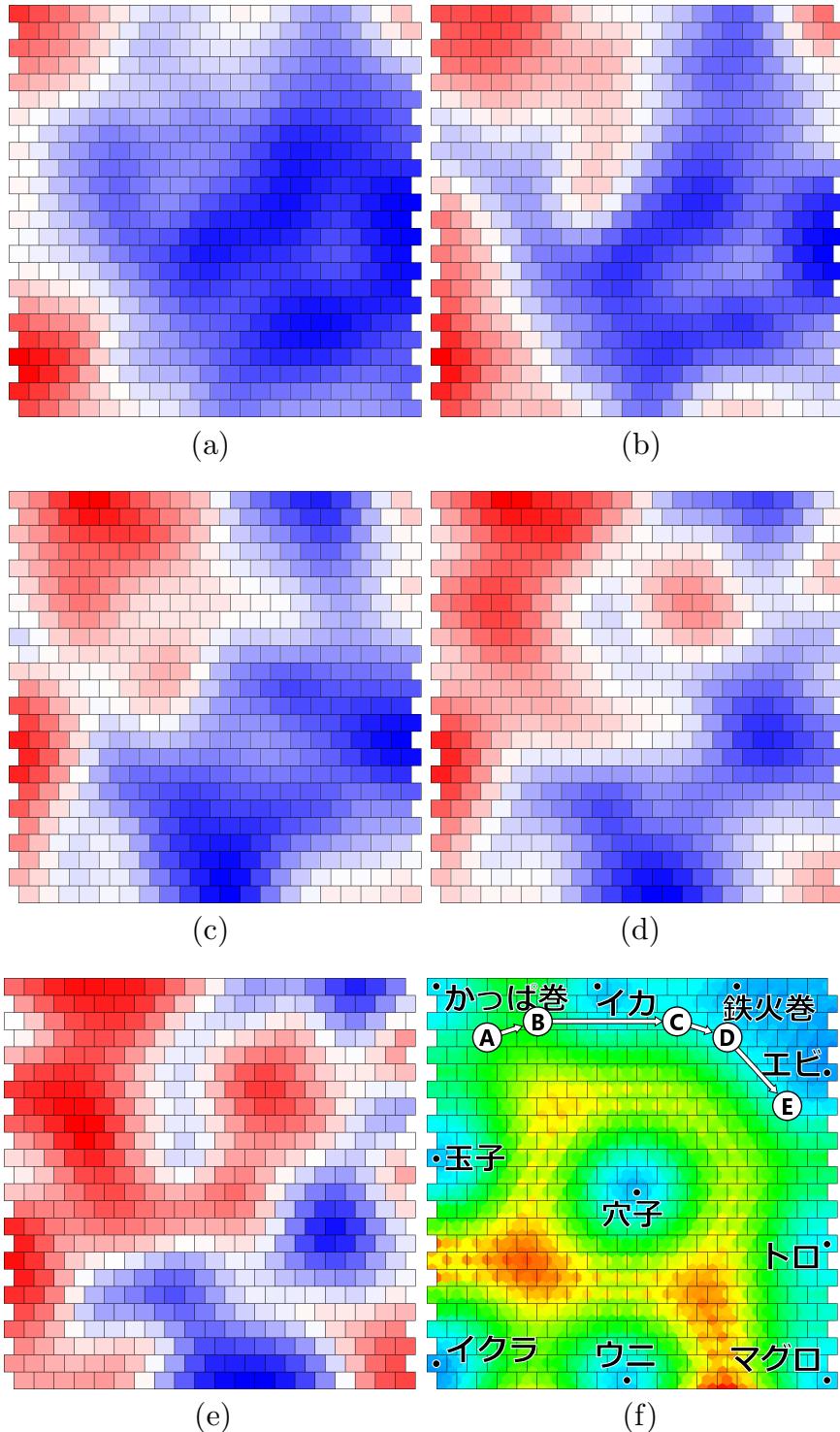


図 5.3 寿司データの解析結果. (a)-(e) ユーザマップ. (f) 寿司マップ. (a)-(e) は Conditional Component Plane で彩色した. それぞれ寿司マップの特定の条件点に対する回答者スコアを示しており, その条件点は寿司マップ (f) 上のラベル A-E の点である. すなわちマップ A における赤/青の領域はそれぞれかつば巻を好む/好まない回答者に対応し, マップ E は各回答者がエビを好む度合いを示す. 寿司マップにおける条件点を A から E へ連続的に動かすと, ユーザマップの Conditional Component Plane の彩色も連続的に変化する.

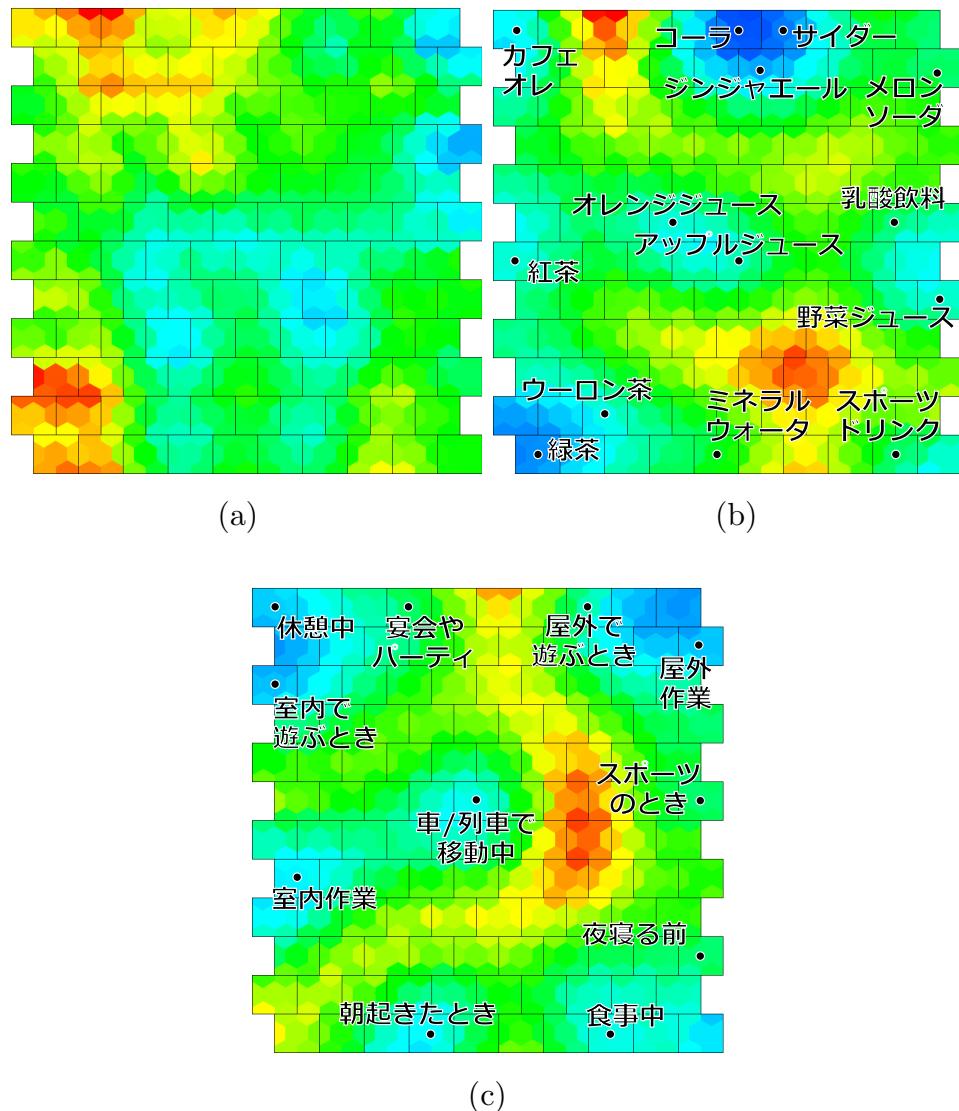


図 5.4 飲料データによるマップ。Marginal U-matrix で彩色した。(a) ユーザ(回答者)マップ (b) 商品(飲料)マップ (c) コンテキスト(状況)マップ

ることができ、Conditional の手法の強みを最大に引き出せる。一方のマップでマウスをクリックすると、その点を着目点とみなし、他方のマップに Conditional U-matrix あるいは Conditional Component Plane が表示されるようにする。条件付けしたマップ上で着目点をマウスで連続的に変化させると、他方のマップでも色付けが連続的に変化する様子が見える。これによってあたかも関係データの中を探索するように、求める情報を次々に可視化することができるようになる。このような直感的解析を可能にする解析ソフトを

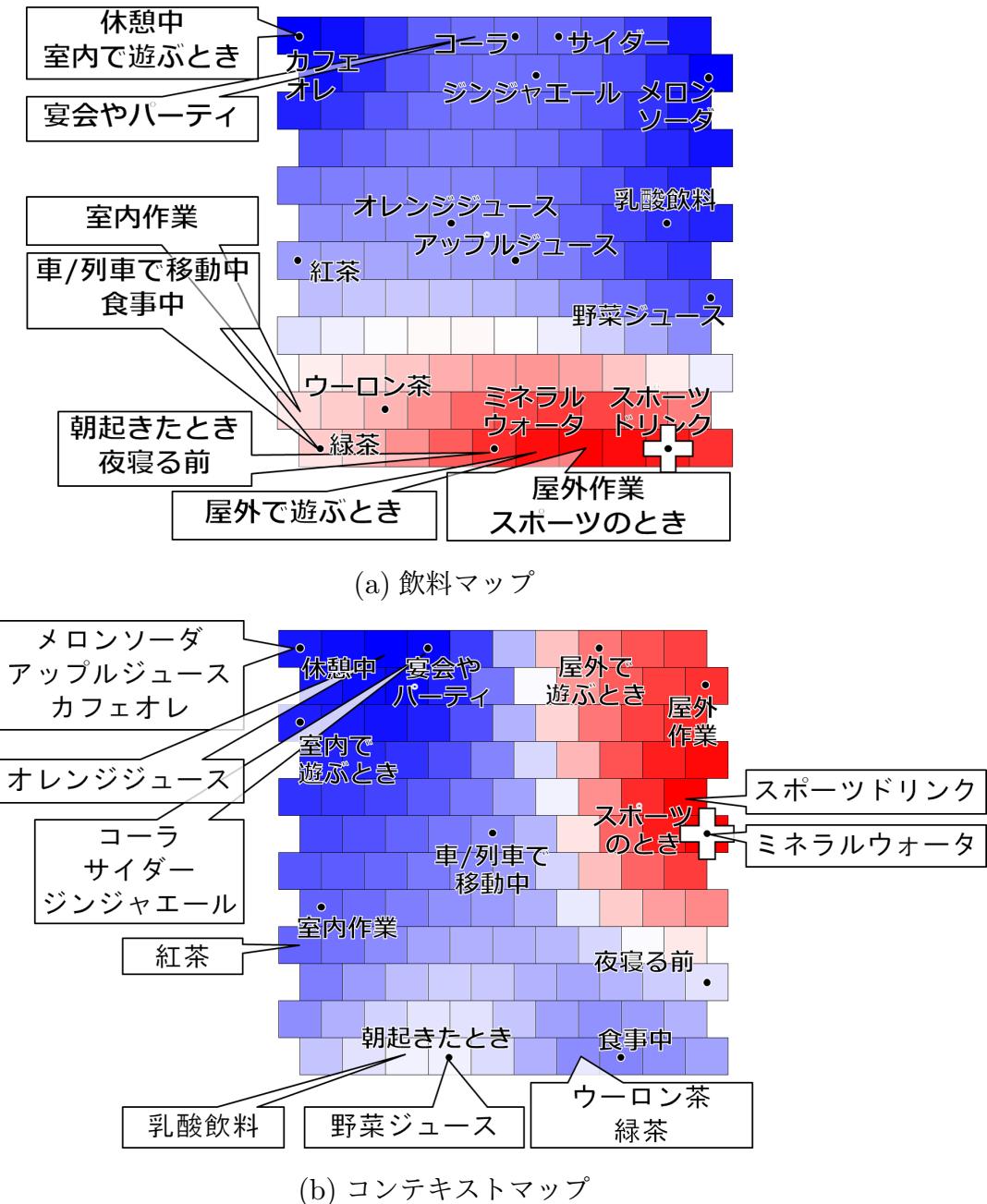


図 5.5 Conditional Component Plane で彩色した飲料データマップ。赤/青の領域はスコアの高/低を表す。また吹き出しが Correspondence overlay である。なおユーザに関しては周辺化してある。(a) 「スポーツのとき」に好まれる飲料を Conditional Component Plane で示した(条件点はコンテキストマップ(b)の+印により示した)。飲料マップの赤い領域(「ミネラルウォーター」と「スポーツドリンク」)が「スポーツのとき」に特に好まれていることを表している。(b) 「スポーツドリンク」が好まれるコンテキスト(条件点は飲料マップ(a)の+印で示した)。コンテキストマップの赤い領域が「スポーツドリンク」の好まれる状況であり、「スポーツのとき」、「屋外作業」、「屋外で遊ぶとき」で好まれることがわかる。またコンテキストマップ(b)上の Correspondence overlay に示した「スポーツドリンク」もこれらのコンテキストに近い位置(もっとも赤い位置)に配置されている。

試作した<sup>\*3</sup>.

しかしインタラクティブな可視化は直感的な解析を可能にするものの、一枚の図としてプリントすることはできない。ここでは静止画として Conditional Component Plane の情報を要約して表示する方法を提案する。またこの方法を **Correspondence overlay** とよぶ。Correspondence overlay の具体例を図 5.5(a) に示す。ここでは飲料マップに対し、コンテキストのラベルが吹き出しとして表示されている。これらのラベルはそれぞれのコンテキストの Conditional Component Plane の最大値の位置に示している。たとえば運動中 (Exercise time) のラベルは、その状況下においてもっとも高いスコアの飲料の位置に配置される。一方、図 5.5(b) はコンテキストマップに飲料ラベルをオーバーレイしている。この場合、オーバーレイされた飲料ラベルは、コンテキストマップにおいてそれぞれの飲料に対するスコアがもっとも高くなる点を表す。わかりやすくかくと、図 5.5(a) の吹き出しは各コンテキストにおいてもっとも好まれる飲料の位置を飲料マップ上に示したものであり、図 5.5(b) は各飲料がもっとも好まれる場面をコンテキストマップ上に示したものである。したがって両者の意味は異なる。

Correspondence overlay は Conditional Component Plane で表される高スコア領域が単一であれば効果的に機能するが、たとえばマップの両端などの複数の領域で高スコア領域が発生した際に、どちらかひとつのみにしかラベルを載せることができず、関係データを理解する上で誤解を生む要因となるので、十分気をつける必要がある。これについては、単純に最大値の位置にオーバーレイするのではなく、たとえば閾値以上の極大値をもつ点すべてにオーバーレイするように改良することで対応できる。しかしそれでも誤解を生む可能性は依然として残っており、データを正確に理解するのが目的であれば Correspondence overlay の多用は避けてインタラクティブに探索するのがよい。Correspondence overlay はあくまで一枚の静止した図として情報を載せる必要がある場合に限り使用すべきである。

---

<sup>\*3</sup> <http://www.brain.kyutech.ac.jp/~furukawa/tsom-e>

## 第 6 章

# TSOM のバリエーション

本章では実際に解析対象となる関係データの典型的な特性に焦点をあて、その特性に合わせた TSOM の拡張を提案する。まず欠損値のある関係データに対し TSOM のアルゴリズムを拡張する。提案したアルゴリズムを用いて、人工データによる検証と実データである映画評価アンケートデータへの応用を行う。次に、属性情報とよばれる付加情報が付随した関係データに対応したアルゴリズムへの拡張を考え、これも同様に映画評価アンケートデータへ応用する。最後に関係データのモードが重複している場合の TSOM アルゴリズムを提案する。たとえば特定のグループ内におけるコミュニケーションのデータは(発信者)×(応答者)×(発信内容)といった関係データとして表すことができる。このデータにおいて、第 1 モードと第 2 モードの母集団は共通している。このような共通の母集団に関する解析に焦点をあててアルゴリズムを拡張し、実際の電子メールトライックデータに対して応用する。以上の応用を通して、さまざまな状況に応じて TSOM が拡張できることを示す。なお本章では TGTM については特に言及しないが、TGTM のバリエーションを同様に作ることも可能である。

### 6.1 欠損データ推定

すでに述べたようにオンラインショップにおけるユーザと商品の 2 モード関係データは TSOM の典型的なアプリケーションである。しかし実際のオンラインショップではすべての商品を購入しているユーザやすべてのユーザに購入されている商品は通常存在しない。あらゆるユーザは商品のごく一部しか購入しておらず、表の大部分は欠損となるのが常である。したがって欠損値推定は実際のデータ解析として応用する上で重要である。ま

た推薦システムを構築する上でも重要な [46, 47].

以下では説明を容易にするため、2モード関係データ ( $M = 2$ ) の場合について述べる。与えられたデータに欠損値がある場合、Mステップ(3.18)は次式になる。

$$y_{k_1 k_2 d} = \frac{1}{g_{k_1 k_2 d}} \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} r_{k_1 n_1}^{(1)} r_{k_2 n_2}^{(2)} \gamma_{n_1 n_2 d} x_{n_1 n_2 d}. \quad (6.1)$$

ここで  $g_{k_1 k_2 d} \triangleq \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} r_{k_1 n_1}^{(1)} r_{k_2 n_2}^{(2)} \gamma_{n_1 n_2 d}$  であり、 $\gamma_{n_1 n_2 d}$  はデータ  $x_{n_1 n_2 d}$  が存在するかどうかを表すバイナリ行列である。すなわち  $x_{n_1 n_2 d}$  が欠損のとき  $\gamma_{n_1 n_2 d} = 0$  となり、存在するときは  $\gamma_{n_1 n_2 d} = 1$  である。インスタンス多様体についても同様に計算できる。

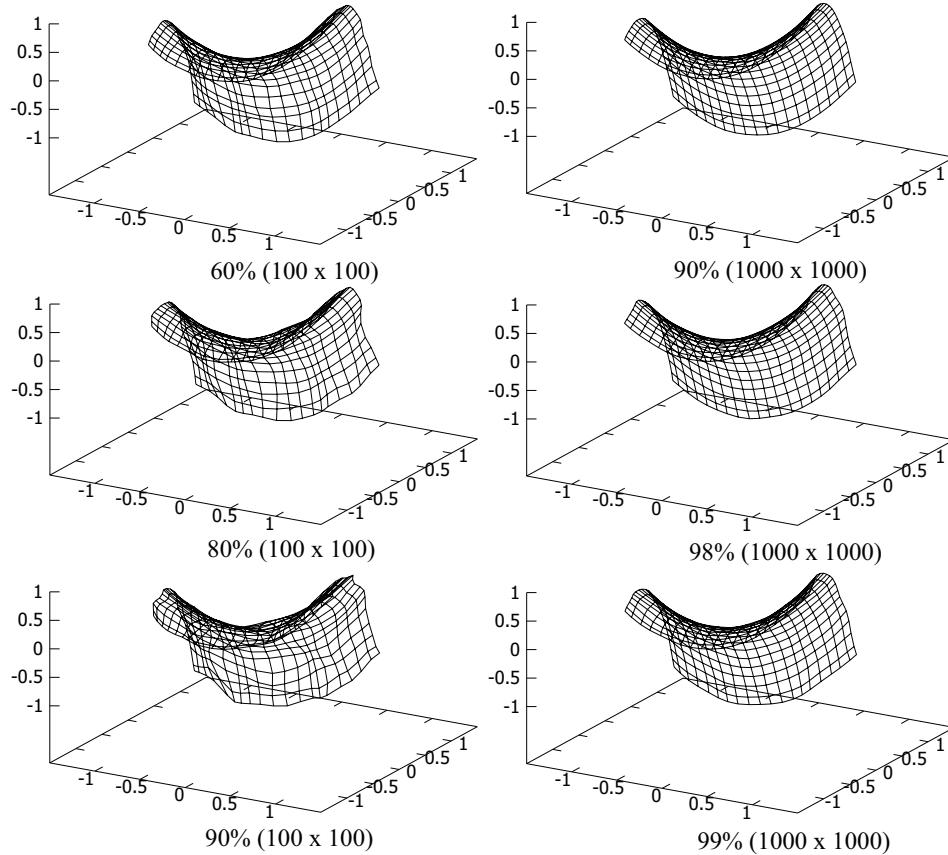
Eステップに関しては2つの方法が考えられる。第一は欠損のない場合(3.57)と同様にインスタンス多様体とスライス多様体を用いて勝者決定する方法である。この方法は勝者が高速に計算できる。ただし欠損率が高くなると各インスタンスにわずかな観測データしか残らないこともある。そのような場合は(3.56)を用いて勝者を直接求めた方が良い。なぜならインスタンス多様体とスライス多様体の距離が欠損により影響を受けて正確に見積もれなくなるからである。学習の完了後、欠損値は  $\tilde{x}_{n_1 n_2 d} = y_{k_{n_1}^{*(1)} k_{n_2}^{*(2)} d}$  として推定できる。

このアルゴリズムを図4.1と同じ人工データを元に検証した。完全なデータセットからランダムに欠損を生じさせ、それを欠損データに対応したTSOMに学習させた<sup>\*1</sup>。このシミュレーションでは勝者を(3.56)で決定した。以上の状況をさまざまな欠損率で検証した。その結果を図6.1に示す。図6.1(a)から欠損があってもTSOMは多様体の形状をうまく捉えており、欠損率が90%でも多様体の形状は崩れなかったことがわかる。またこのときの欠損推定誤差をRMSEで評価した結果が図6.1(b)であり、欠損値に対してロバストであることがわかった。

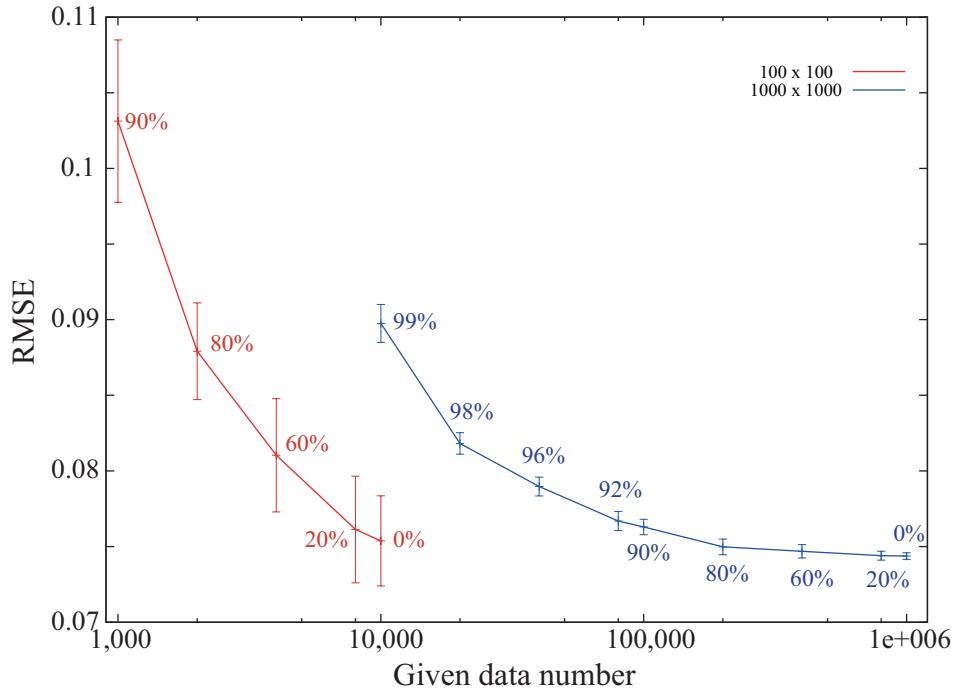
次にこのアルゴリズムをMovieLensデータセット<sup>\*2</sup>に応用した。MovieLensデータセットは映画タイトルに関する、ユーザと商品の2モード関係データである。ここではサブセットであるMovieLens 100kを用いた。このデータには100,000個の評価スコアが5段階評価として入っている。このスコアは943人の回答者が1,682個の映画タイトルを評価したものであり、約94%のデータが欠損している。なおこのデータは回答者の属性および映画タイトルのジャンルも含まれる。しかしそれらは欠損値推定精度を

<sup>\*1</sup> ただし各インスタンスに最低2個はデータが残るようにした。

<sup>\*2</sup> <http://grouplens.org/datasets/movielens/>



(a)



(b)

図 6.1 欠損データの推定結果. (a) さまざまな欠損率下で写像を推定した結果. (b) 欠損値の推定誤差. データサイズが  $100 \times 100$  の場合と,  $1,000 \times 1,000$  の場合について, 学習データサイズと RMSE の関係をプロットした (欠損率も併せて示している).

評価するこの実験では用いなかった。欠損推定精度を評価するため、学習用とテスト用のデータは MLComp<sup>\*3</sup>が提供するデータを用いた。その結果、欠損値推定の RMSE は  $0.9533 \pm 0.0008$  であった。この値は MLComp 内の他の手法によるトップ 10(RMSE は 0.954) とほぼ同程度の性能である。最も精度の良いアルゴリズムは RMSE が 0.923 であり、本手法より 0.03 程度精度が良い。序論でも述べたとおり、TSOM および SOM の主目的は可視化にあり、そのために 2 次元の潜在空間という困難な拘束条件が加わっている。このような状況下であるにも関わらず、単に精度のみを求めた手法と比べて大きく遜色ない結果が得られていることがわかる。

本節では、われわれは欠損値推定を行う暫定的でシンプルなアルゴリズムを紹介した。しかし欠損値推定に EM アルゴリズムを応用することで、欠損値推定を TGTM のアルゴリズムに織り込み、非線形写像の推定等と統合することができるはずである。このような確率論的な欠損値推定法を開発することは、今後の研究の重要なポイントである。

## 6.2 属性情報の活用

実際の関係データ解析の場合、属性情報と呼ばれるデータが与えられることがしばしばある。たとえば MovieLens データセットの場合、回答者の属性情報として年齢、性別、職業などのユーザ属性が与えられている。このような場合、属性情報も関係データと共に考慮してマップを作ることができる。

$\mathbf{x}_n^{\text{s.i.}(m)}$  をモード  $M$  の属性情報とする。このときモード  $m$  の勝者はインスタンス多様体と属性情報の両方を使って決めるこことする。今、 $\mathbf{y}_k^{\text{s.i.}(m)}$  をモード  $m$ 、第  $k$  ノードの属性情報用参照ベクトルとし、属性情報  $\mathbf{x}_n^{\text{s.i.}(m)}$  の次元を  $D^{\text{s.i.}(m)}$  とする。このとき勝者を次式により決定する。

$$k_n^{*(m)} = \arg \min_k \left\{ \frac{1}{K-m} D \left\| \underline{\mathbf{Y}}_{k(m)} - \underline{\mathbf{U}}_{n(m)}^{(m)} \right\|^2 + \frac{\alpha^{(m)}}{D^{\text{s.i.}(m)}} \left\| \mathbf{y}_k^{\text{s.i.}(m)} - \mathbf{x}_n^{\text{s.i.}(m)} \right\|^2 \right\}. \quad (6.2)$$

$\alpha^{(m)}$  は属性情報をどの程度反映させるかを決める重み係数である。 $\alpha^{(m)} = 1$  であれば関係データと属性情報を同程度に扱うことになる<sup>\*4</sup>。M ステップでは  $\mathbf{y}_k^{\text{s.i.}(m)}$  を次式で更新

<sup>\*3</sup> <http://www.mlcomp.org/>

<sup>\*4</sup>  $\alpha^{(m)} = 0$  として、従来どおり関係データ部分のみで勝者を決める方法も考えられる。この場合でも M ステップでは属性情報を含むため、属性の Component Plane を通してさまざまな解析を行うことが可能である。

する。

$$\underline{\mathbf{Y}}^{\text{s.i.}(m)} = \tilde{\mathbf{R}}^{(m)} \mathbf{X}^{\text{s.i.}(m)}. \quad (6.3)$$

すなわち  $M$  ステップでは通常の SOM のように参照ベクトルを更新するだけでよい。

以上のアルゴリズムを MovieLens データセットに応用した。回答者の属性情報（年齢、性別、職業）および映画タイトルの属性情報（ドラマ、コメディ、SF などのジャンル）について、 $\alpha^{(m)} = 1$  として勝者決定に利用した。学習結果を図 6.2 に示す。これによりユーザマップ上では年齢と性別に関するクラスタ境界が見られた。映画ジャンルについても同様に示した。

### 6.3 正方関係データへの応用

通常の TSOM において各モードはそれが解析したい対象集合となっている。したがって  $M$  モード関係データの場合、 $M$  種類の解析対象が存在するのが普通である。しかしながら、2つ以上のモードで対象を共有することもある。すなわち同一の母集団から2個のインスタンスがサンプルされ、そのインスタンスの組み合わせで観測値が決まるような場合である。関係データは非線形写像  $\mathbf{x}_{n_1 n_2} \simeq f(\mathbf{z}_{n_1}, \mathbf{z}_{n_2})$  によってモデル化されるが、 $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}$  が同じ潜在空間  $\mathcal{Z}$  に属することになる。このとき関係データはこの2モードに関して要素数およびラベルが同一となり正方形となるので、本稿では正方形と呼ぶこととする。なおこのような状況は3モード以上が同じ潜在空間を共有する場合にも一般化できる。

典型的な応用事例が同一コミュニティ内のメッセージトライック解析である。たとえばひとつの会社内の電子メール解析などが該当する。この場合、メッセージの送信者集合と受信者集合は同一になる。これを2モード関係データとみなして通常の TSOM で解析すると、送信者マップと受信者マップをそれぞれ別個に作ることとなる。そしてメンバーは両方のマップに1回ずつ（それぞれ異なる場所に）配置される<sup>\*5</sup>。しかしながら、もしメンバー間の双方向的な役割の解析がやりたいことで、そのためにはメンバー全体の人間関係を表すマップがひとつ欲しいということもある。その場合、送信量と受信量の双

---

<sup>\*5</sup> 言い換えると、各メンバーは4次元空間中の1点に写像されることとなる。なおこのほうが各メンバーの送信傾向と受信傾向をそれぞれ別々に解析できるため、都合が良いケースも多い。たとえば社内での役割が送信の立場と受信の立場で異なるようなケースである。正方形とみなして同一の潜在空間として考えることの意味は、送信と受信に関して役割が変化しないという仮定の元で、2次元の潜在空間上に社員の役割を可視化する、ということである。

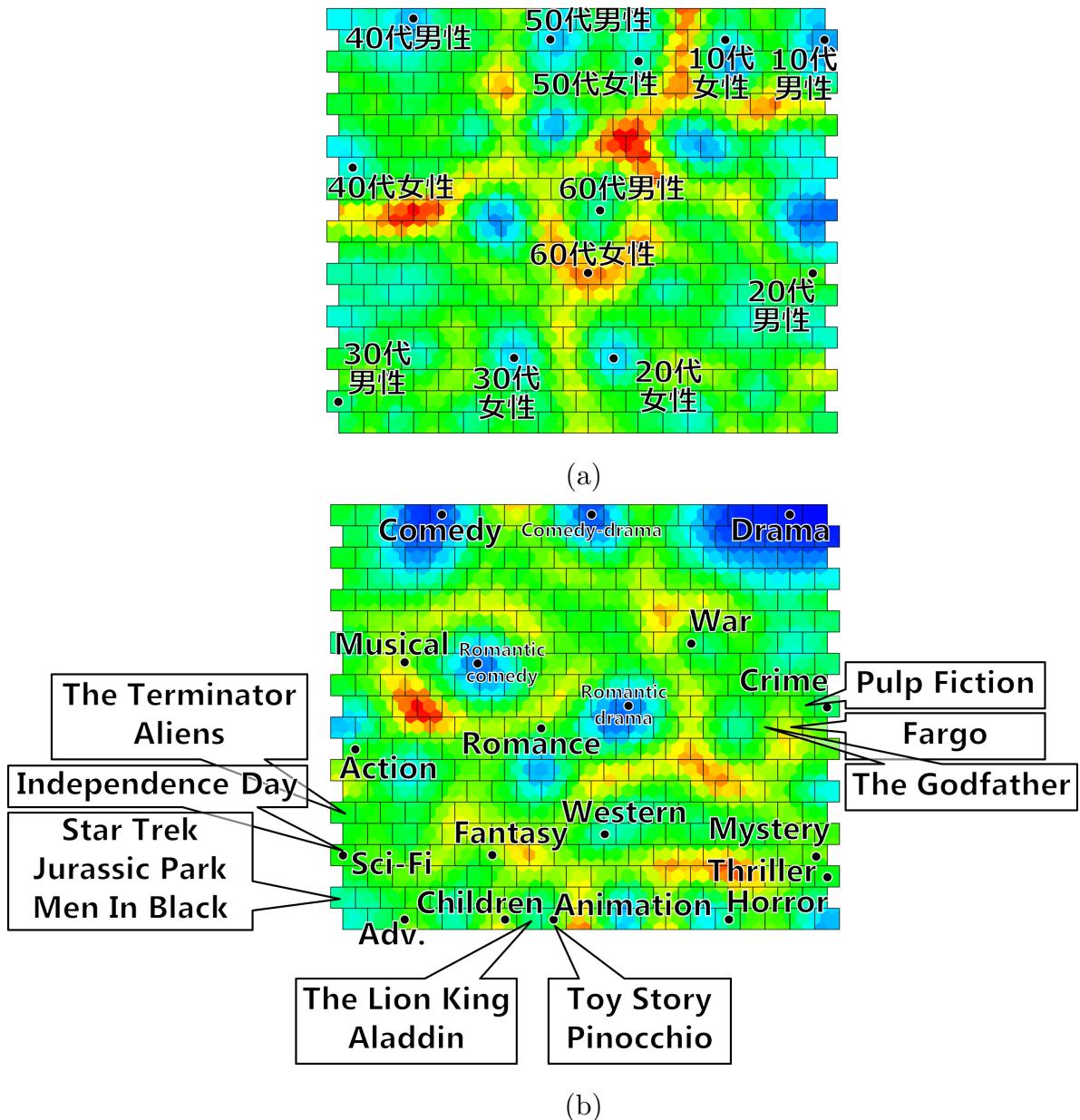


図 6.2 MovieLens データにより生成したマップ. (a) ユーザマップ. 回答者数が 943 人と多いため、ここでは各年代・性別についてもっとも密度の高い点を Correspondence overlay を用いてラベルとして示している. (b) 映画マップ. 映画のジャンルに関しても Correspondence overlay でラベルを示している. またいくつか有名な映画タイトルを吹き出しとして併せて示した.

方を考慮して勝者を決定することで、送信と受信の2つのマップをひとつに統合することができる。すなわち(3.2)-(3.4)の代わりに次式で勝者決定をする。

$$k_n^* = \arg \min_k \left( \left\| \mathbf{Y}_{k::} - \mathbf{U}_{n::}^{(1)} \right\|^2 + \left\| \mathbf{Y}_{::k} - \mathbf{U}_{::n}^{(2)} \right\|^2 \right). \quad (6.4)$$

このアルゴリズムをエンロン E-mail データ<sup>\*6</sup>に応用した[48, 49]。このデータセットはエンロン社の148社員の電子メールデータからなる。実験の手続きは以下の通りである。実験で用いたのは社員から社員への電子メールトラフィックを関係データにしたものであり、メールの‘To:’フィールドで送受信者を特定した。メール数は非常に長い分布の裾をもつため、指數関数を用いておおむね[0, 1]の一様分布になるように変換した。自分自身へのメールについてはデータから除外し、メール数は欠損値扱いとした。

TSOMでの学習結果を図6.3に示す。Marginal U-matrix(図の中央)により、いくつかの社員クラスタが存在することが見て取れる。もっとも明確なクラスタは重役らのものであり、電子メールの送受信パターンが他の社員と大きく異なっていた。マップ上の2点間のメールトラフィックはConditional Component Planeで見ることができる(図の周囲の小さいマップ)。これらの結果から、メールはおもにクラスタ内でやりとりされていることがわかる。このことからクラスタは社内の部署に該当するのではないかと思われる。

## 6.4 まとめ

この章では TSOM の拡張性を示すために実データを用いたデモンストレーションを行った。無論得られたマップがデータマイニング上で意味のあるものになっているかどうかは、TSOM とは別の観点や別の手法も含めて吟味する必要がある。本章で主張したいことは、TSOM のバリエーションを作ることでさまざまな分野に応用が可能になるということ、そして前章で述べた可視化法を組み合わせることで多くの情報を引き出せるということである。

---

<sup>\*6</sup> <https://www.cs.cmu.edu/~./enron/>

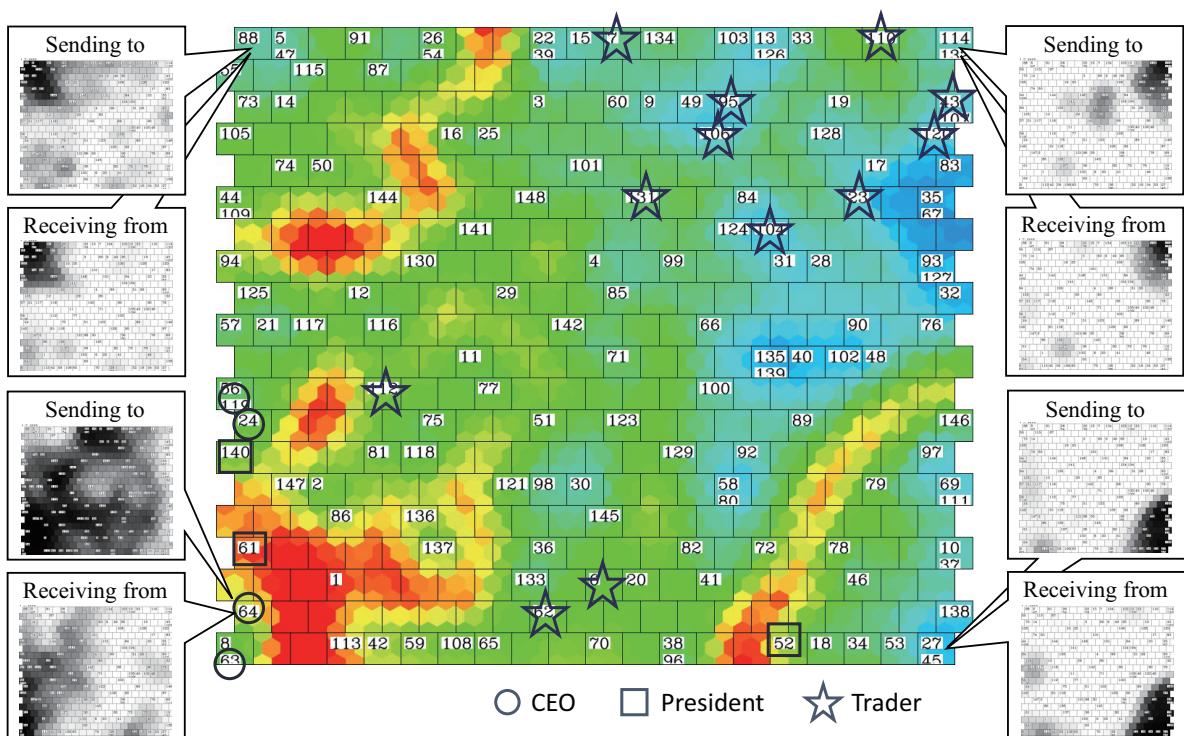


図 6.3 エンロン社の電子メールデータによる社員マップ。中央のマップは Marginal U-matrix で色付けしており、クラスタ境界が赤で示されている。数字はユーザ ID である。周囲の小さなマップは Conditional Component Plane であり、マップの各点におけるユーザが他の社員へ送ったメールおよび受信したメールの分布を表す。

## 第 7 章

# 討論

本章ではテンソル表現に関する SOM の他のバリエーションと TSOM について比較する。この議論は TSOM ファミリーの理論体系を構築するためのひとつの試みでもある。さらに、従来の SOM の一般化という視点で TSOM の意義についても検討する。

### 7.1 TSOM ファミリー

#### 7.1.1 SOM of SOMs ( $SOM^2$ ) との比較

SOM のバリエーションの中で、 $SOM^2$ (もしくは SOM of SOMs) は本手法にもっとも関係の深いものである [50, 51]。TSOM と  $SOM^2$  は非常に似ており、タスクやアルゴリズムに多くの類似点をもつ。第一に、どちらも 2 個もしくはそれ以上の潜在空間から観測空間への非線形写像を推定する。またそのような非線形写像はテンソルとして表現される。第二に、各インスタンスに対する勝者はインスタンス多様体とスライス多様体との距離で決まる点も共通する<sup>\*1</sup>。しかしながら、もっとも本質的な違いは階層構造の有無である。 $SOM^2$  には parent, children と呼ばれる明確な階層性が存在するのに対し、TSOM では階層性が存在しない。この違いはこれらの 2 つのアルゴリズムが取り扱うデータの構造に由来する。すなわち 2 つのアルゴリズムは共通のゴール(すなわち複数の潜在空間から観測空間への非線形写像推定)をもつが、スタート点(すなわち与えられるデータの構造)が異なるのである。この文脈においては、 $SOM^2$  は Tensor SOM for Hierarchical data(TSOM-H)、TSOM は Tensor SOM for Relational data(TSOM-R) と整理するの

---

<sup>\*1</sup>  $SOM^2$  の論文中ではファイバー束の用語を用いて、一方のスライス多様体を **section** と呼び、他方のスライス多様体を **fiber** と呼ぶ [51]。

が適切であると考えられる。TSOM ファミリーの包括的な体系化を行うには、これらのアルゴリズムの類似点と相違点を明確にすることが重要となる。そこで以降、本稿では SOM<sup>2</sup> を TSOM-H<sup>2</sup> と呼び、TSOM ファミリーのメンバーとみなして考える。

### 7.1.2 TSOM-H<sup>2</sup>(SOM<sup>2</sup>) アルゴリズム

TSOM-R<sup>2</sup>(TSOM) と TSOM-H<sup>2</sup>(SOM<sup>2</sup>) の違いはデータ構造の違いにある。そこで、TSOM-H<sup>2</sup>(SOM<sup>2</sup>) の生成モデルから始めることにする。今、2つの解析対象の母集団  $\Omega^{(1)}$  と  $\Omega^{(2)}$  があるとする。また  $\Omega^{(2)}$  がデータ観測に際して上位 (parent) であり、 $\Omega^{(1)}$  が下位 (child) とする。そして同一の上位インスタンスに対し、多くの下位インスタンスからデータが観測されるとする。たとえば新製品が出るたびにマーケティング調査を個別に行う場合がこの状況に相当する。この場合、製品が上位 (parent)、ユーザが下位 (child) になる。このような場合、得られるデータはもはや関係データではない<sup>\*2</sup>。なぜなら新製品が出るたびに異なるユーザをサンプリングして調査を行うので、一人のユーザが他の製品に対してどのように回答するかを知ることができないためである。

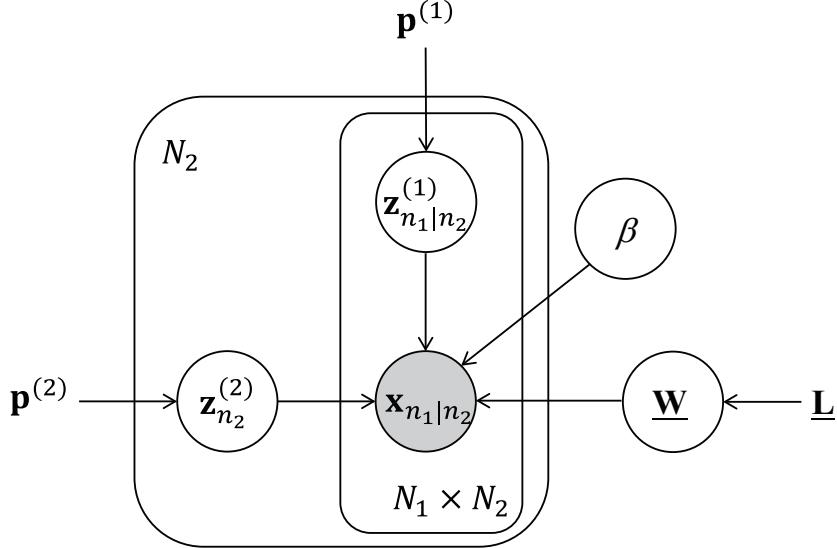
このような状況で、観測データは以下のように得られる。まず上位モードからインスタンスのサンプル集合  $\Omega_S^{(2)} = \{\omega_{n_2}^{(2)}\}$  を得る (個々の新製品に相当する)，そして個々のインスタンス (商品) に対して、独立に下位モードのインスタンス (その製品に関するマーケティング調査の回答者)  $\omega_{n_1}^{(1)}$  をサンプルする。すなわち  $N_2$  個のサンプル集合  $\Omega_{S|\omega_{n_2}^{(2)}}^{(1)} = \{\omega_{n_1|n_2}^{(1)}\}$  をモード 1 について得る。ここで注意すべきことは、 $n_2 \neq n'_2$  ならば  $\omega_{n_1|n_2}^{(1)} \neq \omega_{n_1|n'_2}^{(1)}$  ということである。つまり商品  $n_2$  に関する第  $n_1$  回答者と商品  $n'_2$  に関する第  $n_1$  回答者は別人である。以上より、 $N_1 \times N_2$  個のデータ  $\{\mathbf{x}_{n_1|n_2}\}$  を  $(\omega_{n_1|n_2}^{(1)}, \omega_{n_2}^{(2)})$  のコンビネーションから得ることになる。この生成モデルは図 7.1 のようにグラフィカルモデルとして書くことができる<sup>\*3</sup>。

以上の議論を元に、SOM<sup>2</sup> のアルゴリズムを TSOM-H<sup>2</sup> として以下のように書き換える。以下のアルゴリズムは SOM<sup>2</sup> の論文に書かれているものとは異なるように見えるも

---

<sup>\*2</sup> 関係データとして解釈することも可能ではあるが、特殊な欠損構造をもつ関係データとなる。ここではそういう関係データとしてとらえることはせず、階層構造をもつデータとして議論をすすめる。

<sup>\*3</sup> 厳密にいえば、サンプル  $N_1$  はそれぞれ上位インスタンス  $\omega_{n_2}^{(2)}$  ごとに違って良い。たとえば新製品ごとに異なる人数のマーケティング調査をしてもよい。しかし説明が煩雑になるため、ここでは同数のサンプルをとることとする。

図 7.1 SOM<sup>2</sup>(TSOM-H<sup>2</sup>) の生成モデルのグラフィカルモデル

のの、まったく等価なものである。E ステップでは勝者が次式で決定される。

$$k_{n_1|n_2}^{*(1)} = \arg \min_{k_1} \left\| \mathbf{y}_{k_1 k_{n_2}^{*(2)}:} - \mathbf{x}_{n_1|n_2} \right\|^2 \quad (7.1)$$

$$k_{n_2}^{*(2)} = \arg \min_{k_2} \left\| \mathbf{Y}_{:k_2:}^{(2)} - \mathbf{U}_{:n_2:}^{(2)} \right\|^2. \quad (7.2)$$

(7.1), (7.2) の意味するところは、親モード（ここでは第 2 モード）の勝者はインスタンス多様体  $\mathbf{U}_{:n_2:}^{(2)}$  とスライス多様体  $\mathbf{Y}_{:k_2:}^{(2)}$  の距離で決まり、子モード（ここでは第 1 モード）の勝者は親モードの勝者スライスにおける勝者ユニットとして決定する。

勝者を決定した後、近傍行列  $\mathbf{H}^{(1)}, \mathbf{H}^{(2)}$  および勝者行列  $\mathbf{B}_{n_2}^{(1)}, \mathbf{B}^{(2)}$  を用いて負担率行列  $\mathbf{R}_{n_2}^{(1)}, \mathbf{R}^{(2)}$  を計算する。

$$\mathbf{R}_{n_2}^{(1)} = \mathbf{H}^{(1)} \mathbf{B}_{n_2}^{(1)} \quad (7.3)$$

$$\mathbf{R}^{(2)} = \mathbf{H}^{(2)} \mathbf{B}^{(2)}. \quad (7.4)$$

なお子モードの負担率行列  $\mathbf{R}_{n_2}^{(1)}$  は親モードのインスタンス  $\omega_{n_2}^{(2)}$  に依存するが、親モードの負担率行列  $\mathbf{R}^{(2)}$  は子モードに依存しないことに注意する。

M ステップでは写像を次式で更新する。

$$\mathbf{U}_{:n_2:}^{(2)} = \mathbf{X}_{:n_2:} \times_1 \tilde{\mathbf{R}}_{n_2}^{(1)} \quad (7.5)$$

$$\underline{\mathbf{Y}} = \underline{\mathbf{U}}^{(2)} \times_2 \tilde{\mathbf{R}}^{(2)}. \quad (7.6)$$

そして以上の E,M ステップを繰り返し計算する。

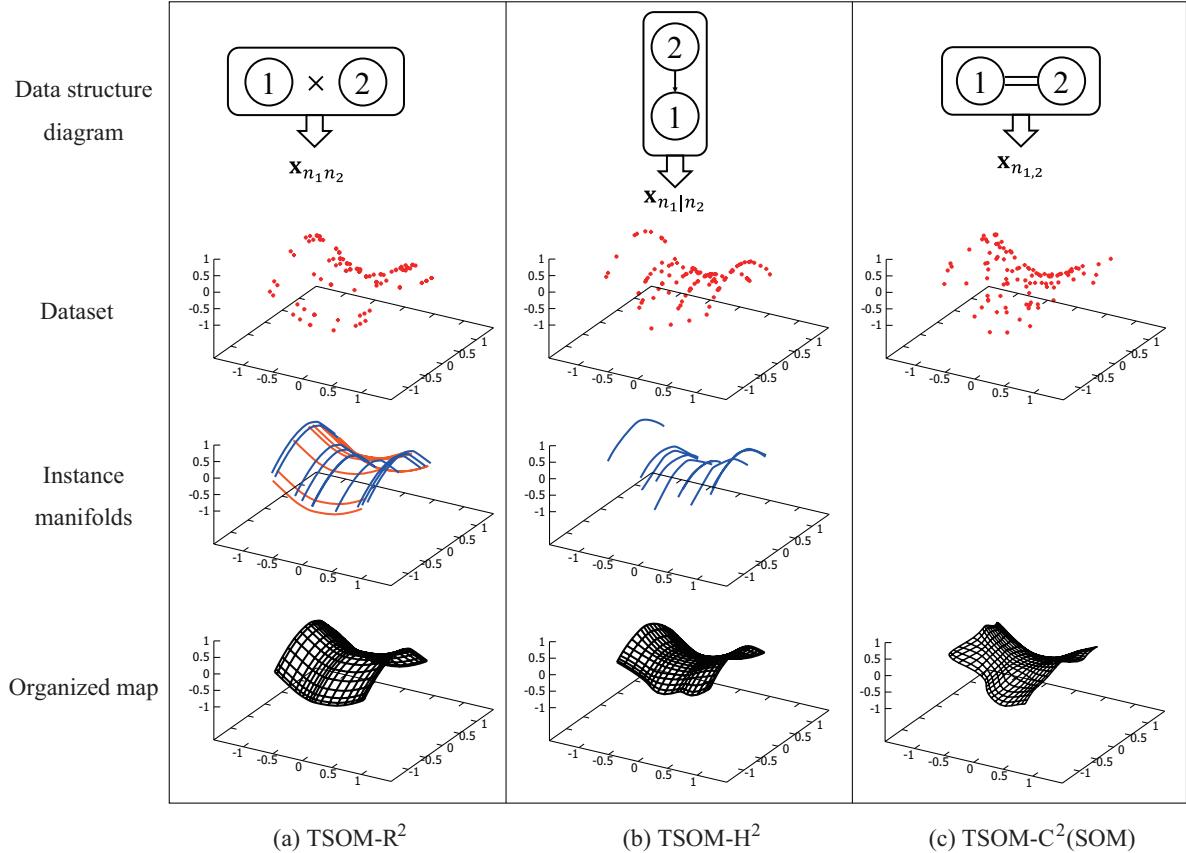


図 7.2 可能なデータ構造と対応するアルゴリズム. (a) 関係データ: TSOM-R<sup>2</sup>, (b) 階層データ: TSOM-H<sup>2</sup>, (c) 連結データ: TSOM-C<sup>2</sup>(従来型 SOM になる). 第1行: データ構造ダイアグラム, 第2行: 観測データ例, 第3行: インスタンス多様体, 第4行: 推定した非線形写像. 関係データ (a) はデータが2モードのどちらにも整列するため, インスタンス多様体も2モードどちらについても作れる. 階層データ (b) は1方向にしか整列しないため, インスタンス多様体は上位モードについてしか作れない. 連結データ (c) の場合はデータの整列が見られないため, インスタンス多様体を推定することができない.

図 7.2 には TSOM-R<sup>2</sup> と TSOM-H<sup>2</sup> がどのように非線形写像を学習するかを示した. TSOM-R<sup>2</sup> ではインスタンス多様体を各モードに関して求めるが, TSOM-H<sup>2</sup> では親モードに関してのみ求める. これが, TSOM-R<sup>2</sup> と TSOM-H<sup>2</sup> の相違点である.

### 7.1.3 データ構造ダイアグラム

各手法の入力データとなるデータ構造を表現するため, 図 7.2 の最上段に示したようなデータ構造ダイアグラムを導入する. このダイアグラムにおいて, 各モードのサンプル集合は円形のノードで示されている. またノード内に書かれた数字はモード番号である. も

し 2 個以上のノードを統合するときはオーバルでまとめる。リンクの存在しないノード同士は、互いに独立にサンプルが行われることを意味する。またシンボル  $\times$  は独立したノード間の直積集合を取ることを意味し、具体的にはすべての要素の組み合わせ集合を作ることを意味する。すなわち関係データとなる。図 7.2(a) は 2 モード関係データの場合であり、TSOM-R<sup>2</sup> が扱うデータ構造である。この場合データベクトル  $\{\mathbf{x}_{n_1 n_2}\}$  はすべての直積サンプル集合  $\Omega_S = \left\{ \left( \omega_{n_1}^{(1)}, \omega_{n_2}^{(2)} \right) \right\}$  の要素に対し行われる。たとえばモード 1 がユーザ、モード 2 が商品ならば、ユーザと商品を独立にサンプリングし、すべてのユーザとすべての商品のペアを作ることを意味する。言い換えれば、すべての商品はすべてのユーザに評価され、またすべてのユーザもすべての商品を評価する。

矢印は階層的サンプリングを意味する。すなわち上位の各要素に対して下位のサンプリングを行い、サンプルペアを作ることを意味する。図 7.2(b) は 2 階層の階層的データの場合であり、TSOM-H<sup>2</sup> が扱うデータ構造である。この場合、データベクトル  $\{\mathbf{x}_{n_1 | n_2}\}$  は集合  $\Omega_S = \left\{ \left( \omega_{n_1 | n_2}^{(1)}, \omega_{n_2}^{(2)} \right) \right\}$  のすべての要素について得られる。たとえば商品のサンプルを行い、各商品に関して独立にユーザをサンプリングし、商品とユーザのペアを作ることを意味する。このとき各商品は（商品ごとに異なる）ユーザ集合に評価されるが、各ユーザは 1 個の商品しか評価しない。このような状況は転移学習やマルチシステム学習において現れる。すなわち上位のモードはタスクやシステムパラメータに該当し、下位のモードは与えられたパラメータ下での状態変数に相当する [52, 53]。

また二重線でつながれたノードは連結サンプリングであり、2 つのモードはサンプリングは同時に行われる。図 7.2(c) がその状況である。2 つのノードはデータ観測において連結される。この場合、インスタンスのペア  $(\omega_n^{(1)}, \omega_n^{(2)})$  に対してデータが観測される。つまりランダムにユーザと商品を同時にサンプルする。したがって各商品はただ一人のユーザに評価され、各ユーザもただひとつ商品しか評価しない。ゆえに商品とユーザの情報を分けることができない。このような連結データに対する TSOM のファミリーを TSOM-C<sup>2</sup> (Tensor SOM for coupled data) と呼ぶこととする。興味深い事に、TSOM-C<sup>2</sup> のアルゴリズムはほとんど従来型 SOM と同じになると考えられる。なぜなら勝者の位置を非線形写像全体から一点選ぶ必要があり、それは従来型 SOM の勝者決定と同じ状況になるからである。この状況はブラインド信号源分離 (Blind Source Separation: BSS) に関係しており、非線形 ICA に相当する。実際、SOM を非線形 ICA に応用できるという報告がある [54, 55]。したがって非線形 ICA の観点から TSOM-C を研究することは重要であり、今後の研究課題である。

これらの3ケースは、それぞれタスクの難易度も異なる。ケース(a)では、 $(N_1 N_2 D)$ 個の観測値から $(N_1 + N_2)$ 個の潜在変数を推定する必要がある。一方(b)では $(N_1 N_2 D)$ 個の観測値から $(N_1 N_2 + N_2)$ 個の潜在変数を推定しなければならない。さらに(c)では $(ND)$ 個の観測値から $(2N)$ 個の潜在変数を推定する必要がある<sup>4</sup>。TSOM-R<sup>2</sup>が従来型SOM(TSOM-C<sup>2</sup>)よりもデータ分布をよく捉えるのはこのような理由からである。

#### 7.1.4 3次のTSOMファミリー

もしモード数が3次以上になると、異なるデータ構造が同時に含まれる場合が生じる。たとえば、まず最初にモード1のサンプル集合を作り、それぞれのインスタンスに対してモード2とモード3のサンプルを得るというデータ構造である<sup>5</sup>。この場合、データベクトルは $\mathbf{x}_{(n_2 n_3) | n_1}$ と表記する。3モードデータの場合、図7.3で示す5通りのデータ構造がある（ただし連結データ構造は除く）。それぞれに対応するアルゴリズムの名称は(a) TSOM-R<sup>3</sup>, (b) TSOM-HR<sup>2</sup>, (c) TSOM-RH<sup>2</sup>, (d) TSOM-H<sup>2</sup>R, (e) TSOM-H<sup>3</sup>となる<sup>6</sup>。これらのアルゴリズムはどれもTSOM-RとTSOM-Hを組み合わせることで実現できる。たとえばTSOM-HR<sup>2</sup>のアルゴリズム（図7.3(b)）は次のようになる。

---

<sup>4</sup>  $N = N_1 N_2$ とすれば、どのデータ構造の場合でも $N_1 N_2 D$ 個の観測値が得られることになる。このとき連結データの場合は $2N_1 N_2$ 個の潜在変数を推定することになり、推定しなければならない未知数がもっとも大きくなる。たとえば $N_1 = N_2 = 100$ ,  $N = 10,000$ とすると、データ構造に関わらず10,000個のD次元データを観測することになる。このとき、関係データ(a)の場合はたった200個の潜在変数を推定すれば良いのに対し、階層データ(b)の場合は10,100個推定する必要がある。さらに連結データの場合は20,000個推定しなければならない。当然推定する潜在変数の数は増えるほどタスクは困難になる。

<sup>5</sup> たとえば都道府県別にユーザと地元商品をランダムサンプリングし、評価値を収集する場合に該当する。この場合、ユーザと商品の双方のサンプリングは都道府県に依存する。

<sup>6</sup> なおこの命名規則はMモードに一般化して記述することができない。任意のモード数に対応可能なシンプルな命名規則を作るのは困難である。この命名規則は3モードまでの状況では簡潔に記述できるため、本稿で用いる。

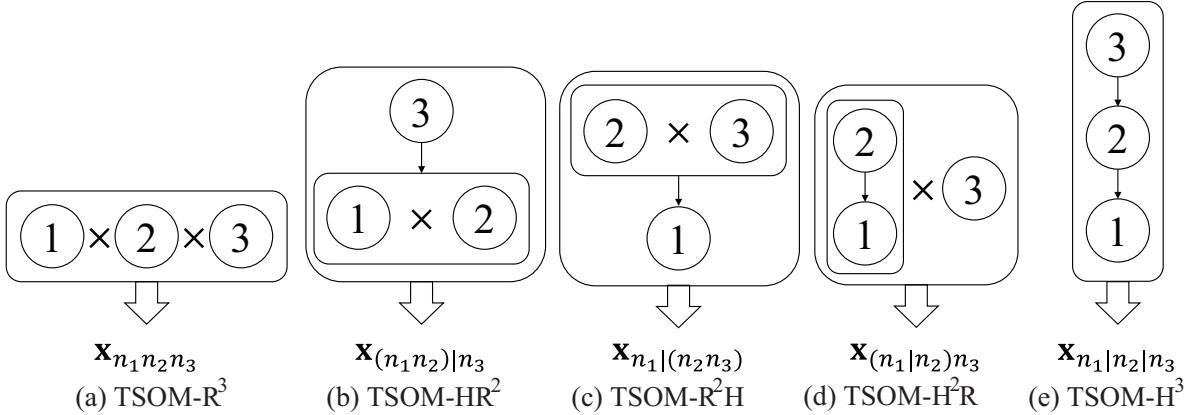


図 7.3 3 次データ構造のダイアグラム

E ステップ

$$k_{n_1 | n_3}^{*(1)} = \arg \min_{k_1} \left\| \underline{\mathbf{Y}}_{k_1 : k_{n_3}^{*(3)} :} - \underline{\mathbf{U}}_{n_1 :: | n_3}^{(1)} \right\|^2 \quad (7.7)$$

$$k_{n_2 | n_3}^{*(2)} = \arg \min_{k_2} \left\| \underline{\mathbf{Y}}_{: k_2 k_{n_3}^{*(3)} :} - \underline{\mathbf{U}}_{: n_2 :: | n_3}^{(2)} \right\|^2 \quad (7.8)$$

$$k_{n_3}^{*(3)} = \arg \min_{k_3} \left\| \underline{\mathbf{Y}}_{:: n_3 :} - \underline{\mathbf{U}}_{:: n_3 :}^{(3)} \right\|^2 \quad (7.9)$$

M ステップ

$$\underline{\mathbf{U}}_{n_1 :: | n_3}^{(1)} = \underline{\mathbf{X}}_{:: n_3 :} \times_2 \widetilde{\mathbf{R}}_{n_3}^{(2)} \quad (7.10)$$

$$\underline{\mathbf{U}}_{: n_2 :: | n_3}^{(2)} = \underline{\mathbf{X}}_{:: n_3 :} \times_1 \widetilde{\mathbf{R}}_{n_3}^{(1)} \quad (7.11)$$

$$\underline{\mathbf{U}}_{:: n_3 :}^{(3)} = \underline{\mathbf{X}}_{:: n_3 :} \times_1 \widetilde{\mathbf{R}}_{n_3}^{(1)} \times_2 \widetilde{\mathbf{R}}_{n_3}^{(2)} \quad (7.12)$$

$$\underline{\mathbf{Y}} = \underline{\mathbf{U}}^{(3)} \times_3 \widetilde{\mathbf{R}}^{(3)} \quad (7.13)$$

すなわち E ステップ, M ステップとともに TSOM-R と TSOM-H を組み合わせた形になる。他のデータ構造の場合も同様にして容易にアルゴリズムを得ることができる。

場合によっては、観測空間の方も 2 個あるいはそれ以上存在することがある。いくつかの例を図 7.4 に示す。この中で (a) のデータケースは 2 つの異なる調査を同一の回答者に対して実施する場合に相当する。また MovieLens のようにサイド情報がある場合は (c) のデータ構造になる。モード数が増加すると、可能なデータ構造の数は指数的に増大す

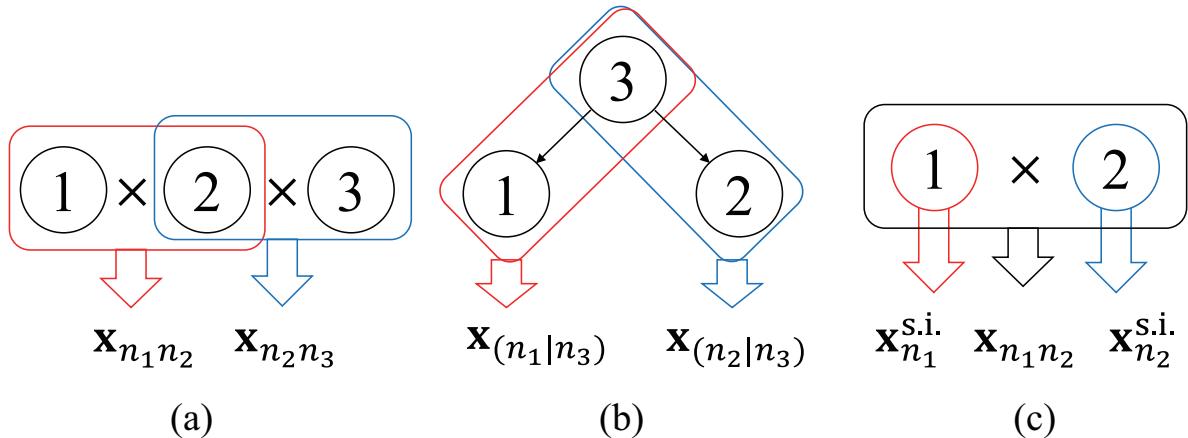


図 7.4 観測空間が複数個ある場合のデータ構造の例. データ構造 (c) はサイド情報つき関係データに該当する.

る。それにもかかわらず TSOM(TGTM) ファミリーはすべてのデータ構造に適応することができる。したがって、これらのアルゴリズムは包括的な理論体系として統合できる。

## 7.2 TSOM と SOM の関係

TSOM は SOM をマルチ潜在空間に一般化したものと考えることができる。すなわち TSOM 側の視点に立てば、オリジナルの SOM は TSOM の特殊な例と見ることもできる。そこで Kohonen のオリジナル SOM について、TSOM の視点から議論する。

オリジナルの SOM では、データベクトルに対してもっとも近い参照ベクトルをもつノードが勝者となる。このとき、データベクトルと参照ベクトルの距離は観測空間における距離で測る。したがって観測空間の距離を測るメトリックが変われば、勝者も変わることになる。通常、データベクトルの各成分はあらかじめ平均ゼロ、分散 1 に規格化しておき、そのうえでユークリッド距離を適用する。この手続きはどの成分も同等に扱うように見えるが、それでもなおメトリックの問題は残る。もしくは複数の成分がまったく同一の情報を扱う場合、それらの重複した成分は他の成分よりも学習に大きく影響する。たとえば生徒の能力を解析する際に、データベクトルの成分がほとんどが身体能力に関するもので、学力に関する成分がほとんどないデータだと、学力はノイズとして無視されてしまう。このようなメトリックの問題は SOM に限らず教師なし学習において本質的に不可避免である。TSOM もまたこの問題を避けることはできないが、好ましくない影響を減らすことはできる可能性がある。すなわちデータベクトル集合を 2 モードの関係データとみなすのである。つまり、TSOM で解析対象のマップ (e.g., 生徒マップ) とデータ成分のマッ

プ (e.g., 測定項目マップ) の 2 つのマップを生成する。もし 2 つのデータ成分がまったく同じならば、それらはデータ成分マップにおいて勝者位置が同じになる。ここで (3.25) 式の勝者決定を用いると TSOM は重複成分の影響力が弱められた結果を出す。このことから、従来は SOM で解析してきたデータであっても TSOM で解析する方が好ましいケースがあると考えられる。

第 2 の視点は SOM の起源に関するものである。SOM はもともと視覚神経系のニューラルネットモデルに起源をもつ。その後、高次元データの次元削減と可視化ツールとして工学応用されるようになった [22]。初期の研究では、網膜から脳への位相保存的神経投射の自己組織化に関する研究 (e.g., [56]) と、視覚的特徴 (たとえば線分の傾きなど) に反応するニューロンの自己組織化に関する研究 (e.g., [57]) の両方が行われていた。一見すると両者の研究は連続的に見えるが、実際にはギャップが存在する。前者は視覚信号ラインの自己組織化であり、後者は視覚信号の特徴の自己組織化である。Kohonen はそれらを type-I, type-II の自己組織化と呼んで区別した [24]。この分類で言えば、SOM は type-II の自己組織化モデルである。そして TSOM の立場で言えば、type-I が成分マップ、type-II が観測対象のマップに相当する。したがって TSOM が 2 つの自己組織化現象を統合するものといえる。

第 3 の視点は潜在空間の座標軸に関するものである。通常、SOM の文脈では潜在空間の座標軸についてはあまり関心が払われない。しかしながら、2 次元正方形の潜在空間を 2 つの 1 次元潜在空間と考えると、SOM は TSOM-C<sup>2</sup> とみなすことができる。すでに議論したように、これは非線形 ICA と関連する。これは TSOM にとって今後の研究が必要な重要なテーマである。



## 第8章

### 総括

本論文ではテンソルデータの位相保存写像のアルゴリズムとして TSOM と TGTM を提案した。TGTM はアルゴリズムの理論的基盤を提供するのに対し、TSOM は実応用可能なアルゴリズムである。理論的には、TSOM と TGTM は関係データの確率的生成モデルに EM アルゴリズムを適用して得られる。いくつかのバリエーションのうち、正規直交基底を用いた TSOM が計算速度も速く解像度も高かった。したがって大規模なタスクには基底関数型 TSOM がもっとも良い選択になる。また本論文では TSOM のさまざまなバリエーションや可視化手法について提案した。これらも TSOM を実応用する上で役に立つ。

一方で SOM<sup>2</sup> と TSOM がひとつのファミリーとして統合されることを示した。したがって本研究はテンソル解析法を提案するのみならず、マルチモードデータを扱うアルゴリズムファミリーの包括的理論体系を確立しようとする試みでもある。



## 参考文献

- [1] D. Laney, 3D Data Management: Controlling Data Volume, Velocity and Variety, META Group, 2001.
- [2] H. Khosravi, B. Bina, A survey on statistical relational learning, in Canadian Conference on Advances in Artificial Intelligence, 2010, pp. 256–268.
- [3] E. Acar, S. A. Çamtepe, M. S. Krishnamoorthy, B. Yener, Modeling and multiway analysis of chatroom tensors, in: Proceedings of the 2005 IEEE International Conference on Intelligence and Security Informatics, ISI'05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 256–268.
- [4] E. Acar, S. A. Camtepe, B. Yener, Collective sampling and analysis of high order tensors for chatroom communications, in: in ISI 2006: IEEE International Conference on Intelligence and Security Informatics, Springer, 2006, pp. 213–224.
- [5] M. W. Berry, M. Browne, Email surveillance using non-negative matrix factorization., Computational & Mathematical Organization Theory 11 (3) (2005) 249–264.
- [6] B. W. Bader, M. W. Berry, M. Browne, Discussion tracking in Enron email using PARAFAC, in: M. W. Berry, M. Castellanos (Eds.), Survey of Text Mining II: Clustering, Classification, and Retrieval, Springer, 2008, Ch. 8, pp. 147–163.
- [7] F. Miwakeichi, E. Martínez-Montes, P. A. Valdés-Sosa, N. Nishiyama, H. Mizuhara, Y. Yamaguchi, Decomposing EEG data into space-time-frequency components using parallel factor analysis, NeuroImage 22 (3) (2004) 1035 – 1045.
- [8] J. Li, L. Zhang, D. Tao, H. Sun, Q. Zhao, A prior neurophysiologic knowledge free Tensor-Based scheme for single trial EEG classification, Neural Systems

- and Rehabilitation Engineering, IEEE Transactions on 17 (2) (2009) 107–115.
- [9] M. A. O. Vasilescu, D. Terzopoulos, Multilinear image analysis for facial recognition, in: ICPR (2), 2002, pp. 511–514.
  - [10] J. Yang, D. Zhang, A. F. Frangi, J.-y. Yang, Two-dimensional PCA: A new approach to appearance-based face representation and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 26 (1) (2004) 131–137.
  - [11] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, H. Zhang, Multilinear discriminant analysis for face recognition, IEEE Trans. on Image Processing 16 (1).
  - [12] N. E. Helwig, S. Hong, J. D. Polk, Parallel factor analysis of gait waveform data: A multimode extension of principal component analysis, Human Movement Science 31 (3) (2012) 630 – 648.
  - [13] T. G. Kolda, B. W. Bader, Tensor decompositions and applications, SIAM REVIEW 51 (3) (2009) 455–500.
  - [14] E. Acar, B. Yener, Unsupervised multiway data analysis: A literature survey, IEEE Transactions on Knowledge and Data Engineering.
  - [15] A. Cichocki, R. Zdunek, A.-H. Phan, S. Amari, Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation, John Wiley & Sons, Ltd, 2009.
  - [16] L. D. Lathauwer, B. D. Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM J. Matrix Anal. Appl 21 (2000) 1253–1278.
  - [17] H. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, MPCA: Multilinear principal component analysis of tensor objects., IEEE Transactions on Neural Networks 19 (1) (2008) 18–39.
  - [18] H. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, A survey of multilinear subspace learning for tensor data, Pattern Recognition 44 (7) (2011) 1540 – 1551.
  - [19] C. F. Beckmann, S. M. Smith, Tensorial extensions of independent component analysis for multisubject FMRI analysis, Neuroimage 25 (1) (2005) 294–311.
  - [20] M. A. O. Vasilescu, D. Terzopoulos, Multilinear independent components analysis, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2005, pp. 547–553.

- [21] M. Welling, M. Weber, Positive tensor factorization., *Pattern Recognition Letters* 22 (12) (2001) 1255–1261.
- [22] T. Kohonen, Self-organized formation of topologically correct feature maps, *Biological Cybernetics* 43 (1) (1982) 59–69.
- [23] T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, Berlin Heidelberg, 2001.
- [24] T. Kohonen, Self-organizing neural projections, *Neural Networks* 19 (6–7) (2006) 723–733.
- [25] T. Kohonen, Essentials of the self-organizing map, *Neural Networks* 37 (3) (2013) 52–65.
- [26] C. M. Bishop, M. Svensen, C. K. I. Williams, GTM: The generative topographic mapping, *Neural Computation* 10 (1998) 215–234.
- [27] C. M. Bishop, M. Svensen, C. K. I. Williams, Developments of the generative topographic mapping., *Neurocomputing* 21 (1-3) (1998) 203–224.
- [28] B. W. Bader, T. G. Kolda, Algorithm 862: MATLAB tensor classes for fast algorithm prototyping, *ACM Transactions on Mathematical Software*, 32 (2006) 635–653.
- [29] L. R. Tucker, Implications of factor analysis of three-way matrices for measurement of change, in: C. W. Harris (Ed.), *Problems in Measuring Change*, University of Wisconsin Press, 1963, pp. 122–137.
- [30] R. A. Harshman, Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multi-modal factor analysis, *UCLA Working Papers in Phonetics*, 16 (1) (1970) 1–84.
- [31] M. Signoretto, L. D. Lathauwer, J. A. Suykens, A kernel-based framework to tensorial data analysis, *Neural Networks* 24 (8) (2011) 861 – 874, *artificial Neural Networks: Selected Papers from ICANN 2010*.
- [32] Q. Zhao, G. Zhou, T. Adali, L. Zhang, A. Cichocki, Kernel-based tensor partial least squares for reconstruction of limb movements, in: *ICASSP’13*, 2013, pp. 3577–3581.
- [33] C. S. Lee, A. Elgammal, Modeling view and posture manifolds for tracking, *IEEE International Conference on Computer Vision* (2007) 1–8.
- [34] X. Gao, C. Tian, Multi-view face recognition based on tensor subspace analysis

- and view manifold modeling, *Neurocomput.* 72 (16-18) (2009) 3742–3750.
- [35] T. Heskes, J.-J. Spanjers, W. Wiegerinck, Em algorithms for self-organizing maps., in: *IJCNN* (6), 2000, pp. 9–14.
  - [36] J. Verbeek, N. Vlassis, B. Krose, Self-organizing mixture models, *Neurocomputing / EEG Neurocomputing* 63 (2005) 99–123.
  - [37] S. P. Luttrell, Derivation of a class of training algorithms, *IEEE Transactions on Neural Networks* 1 (2) (1990) 229–232.
  - [38] Y. Cheng, Convergence and ordering of kohonen’s batch map, *Neural Comput.* 9 (8) (1997) 1667–1676.
  - [39] T. Graepel, M. Burger, K. Obermayer, Self-organizing maps: Generalizations and new optimization techniques, *Neurocomputing* 21 (1998) 173–190.
  - [40] I. Olier, A. Vellido, Variational bayesian generative topographic mapping., *J. Math. Model. Algorithms* 7 (4) (2008) 371–387.
  - [41] N. Lawrence, Gaussian process latent variable models for visualization of high dimensional data, In *Advances in Neural Information Processing Systems* 16 (2003) 329–336.
  - [42] N. Lawrence, Probabilistic non-linear principal component analysis with gaussian process latent variable models, *Journal of Machine Learning Research* 6 (2005) 1783–1816.
  - [43] T. Kamishima, H. Kazawa, S. Akaho, A survey and empirical comparison of object ranking methods, in: J. Fürnkranz, E. Hüllermeier (Eds.), *Preference Learning*, Springer, 2010, pp. 181–201.
  - [44] A. Ultsch, H. P. Siemon, Kohonen’s self organizing feature maps for exploratory data analysis., in: *Proc. INNC’90, Int. Neural Network Conf.*, 1990, pp. 305–308.
  - [45] P. Stefanovic, O. Kurasova, Visual analysis of self-organizing maps, *Nonlinear Analysis: Modelling and Control* 16 (4) (2011) 488–504.
  - [46] G. Ricci, M. de Gemmis, G. Semeraro, Matrix and tensor factorization techniques applied to recommender systems: a survey, *International Journal of Computer and Information Technology* 1 (2012) 94–98.
  - [47] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender

- systems, Computer 42 (8) (2009) 30–37.
- [48] B. Klimt, Y. Yang, The Enron corpus: A new dataset for email classification research, in: Machine Learning: ECML 2004, 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004, Proceedings, 2004, pp. 217–226.
  - [49] A. McCallum, X. Wang, A. Corrada-Emmanuel, Topic and role discovery in social networks with experiments on enron and academic email, J. Artif. Int. Res. 30 (1) (2007) 249–272.
  - [50] T. Furukawa, SOM of SOMs: Self-organizing map which maps a group of self-organizing maps, in: W. Duch, J. Kacprzyk, E. Oja, S. Zadrożny (Eds.), Artificial Neural Networks: Biological Inspirations, Vol. 3696 of Lecture Notes in Computer Science, Springer, 2005, pp. 391–396.
  - [51] T. Furukawa, SOM of SOMs, Neural Networks 22 (4) (2009) 463–478.
  - [52] T. Ohkubo, T. Furukawa, K. Tokunaga, Requirements for the learning of multiple dynamics, in: J. Laaksonen, T. Honkela (Eds.), Advances in Self-Organizing Maps, Vol. 6731 of Lecture Notes in Computer Science, Springer, 2011, pp. 101–110.
  - [53] T. Furukawa, K. Natsume, T. Ohkubo, Research on multi-system learning theory: A case study of brain-inspired system research, in: Proc. of SCIS-ISIS 2012, 2012, pp. 311–314.
  - [54] P. Pajunen, A. Hyvarinen, J. Karhunen, Nonlinear blind source separation by self-organizing maps, in: In Proc. Int. Conf. on Neural Information Processing, 1996, pp. 1207–1210.
  - [55] M. Haritopoulos, H. Yin, N. M. Allinson, Image denoising using self-organizing map-based nonlinear independent component analysis., Neural Networks 15 (8-9) (2002) 1085–1098.
  - [56] S. Amari, Topographic organization of nerve fields, Bulletin of Mathematical Biology 42 (3) (1980) 339–364.
  - [57] C. von der Malsburg, Self-organization of orientation sensitive cells in the striate cortex, Kybernetik 14 (2) (1973) 85–100.



# 研究業績リスト

## I. 学術論文（国際・査読あり）

1. T. Iwasaki, T. Furukawa, “Tensor SOM and Tensor GTM: Nonlinear Tensor Analysis by Topographic Mappings,” Neural Networks 2016 (*accepted*)

## II. 国内学会

(口頭発表・査読なし)

1. 岩崎 亘, 古川 徹生, “テンソル SOM による関係データの可視化,” 第 31 回ファジィシステムシンポジウム, CD-ROM, 2015.
2. 岩崎 亘, 伊達 洋祐, 古川 徹生, “テンソル SOM による企業内チーム分析,” 第 25 回インテリジェント・システム・シンポジウム, 2015.
3. 石橋 英明, 岩崎 亘, 堀尾 恵一, 難波 秀行, 古川 徹生, “クラス推定型高階 SOM によるライフパターンの可視化～ユーザの環境を考慮した解析手法～,” 電子情報通信学会研究報告, Vol.113, No.148, pp29-34, 2013.
4. 伊達 洋祐, 脇田 靖弘, 岩崎 亘, 古川 徹生, “Tensor SOM による MovieLens dataset の解析,” 電子情報通信学会研究報告, Vol.113, No.500, pp63-68, 2014.
5. 脇田 靖弘, 岩崎 亘, 古川 徹生, “欠損データに対する Tensor SOM のロバスト性,” 電子情報通信学会研究報告, Vol.114, No.437, pp21-26, 2015.
6. 伊達 洋祐, 岩崎 亘, 古川 徹生, “テンソル SOM による自家用車インパネの印象評価,” 第 31 回ファジィシステムシンポジウム, CD-ROM, 2015.



## 謝辞

本研究の遂行にあたり以下の方々にお世話になりました。ここに感謝の意を表します。

九州工業大学大学院生命体工学研究科人間知能システム工学専攻の古川徹生教授には5年間の研究生活を通して公私ともに終始温かいご指導、ご鞭撻を賜りました。物事にたいして深く洞察を行いその本質を見定める姿勢は、結論を急ぐ自分にとって学ぶことが多くありました。本論文の主体となるTSOMについては、理論構築を中心としてアルゴリズムの詳細やその拡張について広く議論していただき、その中で多くのご助言とご指導を賜りました。また学会聴講や産学連携などの機会をたくさんいただき、研究の見聞を広める上で多大なご助力をいただきました。古川教授には本当に感謝の念が絶えません。心よりお礼申し上げます。

同専攻・堀尾恵一准教授、吉田香准教授には本研究の質を高めるための貴重なご助言をいただきました。堀尾准教授にはアルゴリズムの記述に関して説明不足な点を丁寧にご指摘いただき、本論文の内容をより正確に伝えるための貴重な議論をしていただきました。吉田准教授には本研究の主張をより明快かつ魅力的に伝えるためのご指摘、ご助言を賜りました。厚くお礼申し上げます。

古川研究室のメンバーには公私にわたって多くのご助力を賜りました。日々の活発な議論から得られる心地よい刺激は、博士後期課程進学について考える大きなきっかけとなりました。薬師寺翔氏をはじめとする諸先輩方は研究や勉強に取り組む真摯な姿勢を見せてくださいり、現役の先輩方の居ない博士後期課程の期間中はそのときの姿勢を手がかりにして歩みを進めることができました。5年間の研究生活をともに過ごした石橋英朗氏および同輩、後輩の方はそれぞれに学んだことや疑問点をいつでも議論して共有し、高めあうことのできる貴重な存在でした。学生の方々には研究に限らずさまざまな面でお世話になりました。深く感謝いたします。

そして長い学生生活を心身ともに支え続けてくれた家族、いつも心の支えとなってくれた友人達に心から感謝いたします。

