

## [5] 研究成果物

### ① 知見・成果物・知的財産権等

サブテーマ4：人間・社会の視覚情報データ解析（人間・社会関連）

1. Monte Carlo Dynamic Classifier ツール（添付資料2）

### ② 成果発表等

<論文発表>

[学術論文]

サブテーマ1：バイオイメージング・インフォーマティクス（医療・生物関連）

1. T. Tokunaga, O. Hirose, S. Kawaguchi, Y. Toyoshima, T. Teramoto, H. Ikebata, S. Kuge, T. Ishihara, Y. Iino, R. Yoshida, Automated detection and tracking of many cells by using 4D live-cell imaging data, *Bioinformatics*, 2014, in press.
  2. 徳永旭将, 吉田亮, 岩崎唯史, データ同化によって線虫の神経系をまるごと読み解く現状と課題, *シミュレーション*, 3(4):31-38, 2014.
- サブテーマ2：オプティカルイメージングデータを用いた脳の動作原理の解明（医療・生物関連）
3. Fujiki, Y., Yokoota, S., Okada, Y., Oku, Y., Tamura, Y., Ishiguro, M., Miwakeichi, F., Standardization of Size, Shape and Internal Structure of Spinal Cord Images: Comparison of Three Transformation Methods, *PLOS ONE*, 8(11), e76415, 2013 (IF 3.73) (査読付き学術論文).
  4. Matsumoto, K., Kikuchi, S., Miwakeichi, F., Yamauchi, Y., Ishiguro, M., Kato, S., A Sensitive Indicator of Hemodynamic Changes in The Lateral Prefrontal Cortex Using a Modified Version of "Rock, Paper, Scissors" as a Task Load, *Journal of Medical and Biological Engineering*, 33, pp97-94, 2013 (IF 0.9) (査読付き学術論文).
  5. Yamauchi, Y., Kikuchi, S., Miwakeichi, F., Matsumoto, K., Nishida, M., Ishiguro, M., Watanabe, E., Kato, S., Relation between Parametric Change of the Workload and Prefrontal Cortex Activity during a Modified Version of the 'Rock, Paper, Scissors' Task, *Neuropsychobiology*, 68, pp24-33 2013 (査読付き学術論文).
  6. 三分一 史和, 生体イメージングデータ解析のための時空間フィルタリング方法, *認知神経科学*, 15-1, pp25-32, 2013 (査読付き学術論文).
  7. Inoue, H., Shimizu, S., Nara, H., Tsuruga, T., Miwakeichi, F., Hirai, N., Kikuchi, S., Watanabe, E., Kato, S., Basic Study for New Assistive Technology Based on Brain Activity during Car Driving, *Journal of Robotics and Mechatronics*, accepted (査読付き学術論文).
  8. Miwakeichi, F., Extraction of Neural Activation from Biological Spatio-temporal Imaging Data using Autoregressive Model-based Filtering Technique, *GLOBAL HEALTH 2013, The Second International Conference on Global Health Challenges*, 2013 (査読付きプロシードィング).
  9. Shimizu, S., Inoue, H., Nara, H., Miwakeichi, F., Hirai, N., Kikuchi, S., Kato, S., Watanabe, E., Basic Study for Human Spatial Cognition Based on Brain Activity During Car Driving, *COGNITIVE 2013, The Fifth International Conference on Advanced Cognitive Technologies and Applications*, 2013 (査読付きプロシードィング).
  10. Inoue, H., Shimizu, S., Nara, H., Miwakeichi, F., Tsuruga, T., Hirai, N., Kikuchi, S., Kato, S., Watanabe, E., Consideration for Evaluation Method of Human Behavior Based on Brain Activity, *COGNITIVE 2013, The Fifth International Conference on Advanced Cognitive Technologies and Applications*, 2013 (査読付きプロシードィング).

[データベース]

特になし。

[著書等]

特になし。

[解説・総説]

サブテーマ3：医用画像処理における統計的手法に関する研究（医療・生物関連）

1. 池田思朗, 計測技術におけるスペースモデリングの応用について, MEDICAL IMAGING TECHNOLOGY (MIT誌), 32巻3号特集「統計的画像処理の研究動向」, 印刷中.

[その他]

特になし。

<会議発表等>

[招待講演]

サブテーマ1：バイオイメージング・インフォーマティクス（医療・生物関連）

【国内】

1. 徳永旭将, "4次元動態計測データからの神経細胞活動度の自動定量化", 第一回腫瘍分子生物学・生命情報共同セミナー, 金沢大学, 2014年3月.
2. 吉田亮, "ライフサイエンスにおけるベイズ統計学の戦略的応用分野の開拓", 第一回腫瘍分子生物学・生命情報共同セミナー, 金沢大学, 2014年3月.
3. 吉田亮, "Bayesian method for data assimilation", 株式会社 国際電気通信基礎技術研究所 特別セミナー, 国際電気通信基礎技術研究所, 2013年12月.
4. 吉田亮, "ベイズ統計学入門：システムズバイオロジー、分子設計、バイオイメージングの応用例を中心に", 第23回 大阪大学生命機能数理モデル検討会, 大阪大学免疫学フロンティア研究センター, 2013年12月.
5. 吉田亮, "ベイズ統計学とバイオサイエンスの協働", JST・CREST「生命動態の理解と制御のための基盤技術の創出」第2回領域会議 産総研セミナー・数理デザイン道場, 日本科学未来館, 2013年11月.

【国際】

6. R.Yoshida, "Bayesian statistics for data assimilation", Cooperation of Computation Materials Science and Mathematics toward Smart Materials Design, Tokyo, 2014年1月.
7. R. Yoshida, "Bayesian statistics for designing systems, molecules and others", Forum "Math-for-Industry"2013, Fukuoka, 2013年11月.

サブテーマ2：オプティカルイメージングデータを用いた脳の動作原理の解明（医療・生物関連）

【国内】

8. Miwakeichi, F., Evolution of the analysis of brain signal (招待講演), Research Center for Applied Perceptual Science Kickoff Symposium, Between Perception and Language, Fukuoka, Japan, 2013.04.01.

[一般講演]

サブテーマ1：バイオイメージング・インフォーマティクス（医療・生物関連）

【国内】

1. 徳永旭将, 広瀬修, 池端久貴, 川口翔太郎, 豊島有, 佐藤博文, 寺本孝行, 久下小百合, 石原健, 飯野雄一, 吉田亮, "逐次的カーネル密度推定法を用いた神経細胞カルシウムイオン分布の定量化", 2013 年度統計関連学会連合大会, 大阪 (大阪大学豊中キャンパス), 2013 年 9 月.

<受賞>

サブテーマ1：バイオイメージング・インフォーマティクス（医療・生物関連）

1. 2013 年度統計関連学会連合大会 最優秀報告賞（徳永旭将）。

③ その他の成果発表

特になし。

## Automated detection and tracking of many cells by using 4D live-cell imaging data

T. Tokunaga, O. Hirose, S. Kawaguchi, Y. Toyoshima, T. Teramoto, H. Ikebata, S. Kuge, T. Ishihara, Y. Iino, and R. Yoshida

*Bioinformatics*, 2014, in press

### 概要

線虫 *C. elegans* の神経系は、302 個の神経細胞で構成され、神経回路網の配線図が全て明らかになっている。近年のカルシウムイオンイメージング技術の進歩により、全神経細胞の約半分に相当する約 150 個の神経細胞の活動状態を同時計測できるようになった。これにより、生きた線虫の神経ネットワークの情報処理の様子をリアルタイムに観察できるようになる。データは、空間 3 次元プラス時間軸の立体動画である。オブジェクトの形状（細胞核）は、細胞ごとに多少の違いはあるが、視覚的に違いを識別することは非常に難しい。また、頭部にとりわけ多数の神経細胞が密集しており、計測中に線虫が不規則に動くことで細胞の位置は時間変化する。したがって、データから各細胞の活動状態を推定するには、細胞数の数え上げ、位置検出、セグメンテーション、そして形状均一性が高い密集した 100~200 個の細胞集団を追跡するための画像解析技術が不可欠となる。本研究では、ベイズ統計学と最適化のテクニックにもとづき個々の問題に対する要素技術を開発し、パイプライン化を行った。細胞の数え上げと位置検出のタスクでは、カーネル密度推定と Repulsive Parallel Hill-Climbing 法という独自の最適化アルゴリズムを組み合わせ、解析手法の設計を行った。細胞のトラッキングでは、マルコフ確率場で細胞位置の時空間変化をモデリングし、トッラカーの置換や併合を回避しながら、約 120 個の細胞の動きを追跡することに成功した。最後に、各時刻のトラッカーの位置情報から細胞を分画し、分画領域の輝度情報から活動状態の推定を行った。

### 論文の解説

カルシウムイオンイメージングでは、神経細胞の核を可視化するカルシウムイオン濃度インディケータの時間変化が、共焦点顕微鏡の立体動画として記録される。実験中、線虫はマイクロ流体デバイスに挿入された状態で顕微鏡のポードに固定され、流路から線虫の鼻先にいくつかの匂いや化学刺激が与えられる。図 1 に、共焦点レーザー顕微鏡で撮影された線虫頭部の神経細胞のカルシウムイオン濃度の 3 次元グレイスケール画像を示す。この画像には、およそ 150 個の神経細胞が含まれている。このような 3 次元画像が時々刻々撮影される。

## Ca<sup>2+</sup> imaging data

Resolution (x,y,z): 512 × 256 × 20 (pixel) Time interval: 0.36 sec

Number of objects (cells): 50-200

Appearance of objects: Ellipsoidal or little bit more general

A worm is immobilized to a thin plastic tube, but the body posture can slightly move

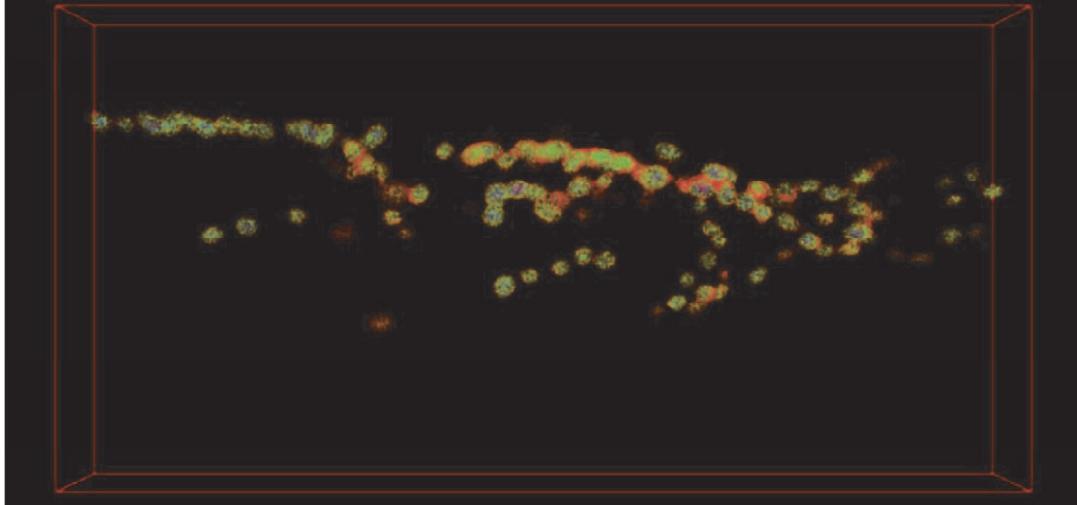


図 1 カルシウムイオンイメージングにより計測された線虫頭部の神経細胞

図 1 のような立体動画から、神経細胞ごとのカルシウムイオン濃度分布の時間変化を定量化するための第一ステップは、神経細胞の個数を数え上げ、位置および細胞領域を同定することである。また、線虫は生きたままチューブに挿入されているが、完全に固定することができないため、計測中の線虫は電子顕微鏡の視野の中で微妙に動いてしまう。そのため、フレームごとに各神経細胞の位置を補正する必要がある。さらに、計測されたカルシウムイオンデータにはアノテーションがない。そのため、計測された画像に映っている細胞の名前を、レファレンス画像から決定する必要がある。まとめると、計測されたカルシウムイオン画像から神経細胞ごとの活動度を定量化するには、画像中のどの領域にいくつの神経細胞が存在するかを決定し、各細胞の位置と境界を求め、さらに位置の時間変化を補正しなくてはならない。また、定量化したデータが、どの神経細胞に相当するかを、レファレンス画像との比較から決定する必要がある。処理の流れとしては、図 2 に示すようなパイプラインとなる。

- Step 1. カーネル密度推定を行い、画像を滑らかな関数に変換
- Step 2. 細胞数と位置の自動検出（ヒル・クライミングでローカルモードを発見）
- Step 3. トラッキング（時間変化するローカルモードを追跡）
- Step 4. セグメンテーション

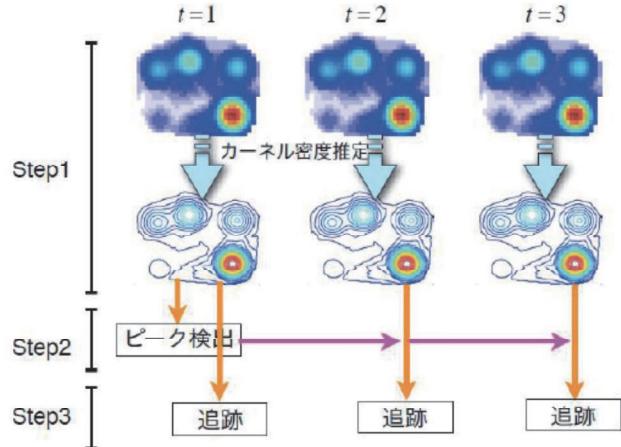


図 2 画像解析パイプライン

本研究ではパイプライン全体の内、細胞の数え上げ、位置検出、追跡、セグメンテーションの要素技術を開発した。具体的には、ベイズ統計学と最適化のテクニックにもとづき個々の問題に対する要素技術を開発した。細胞の数え上げと位置検出のタスクでは、カーネル密度推定と Repulsive Parallel Hill-Climbing 法という独自の最適化アルゴリズムを組み合わせ、解析手法の設計を行った。細胞のトラッキングでは、マルコフ確率場で細胞位置の時空間変化をモデリングし、トッパカーの置換や併合を回避しながら、約 120 個の細胞の動きを追跡することに成功した。最後に、各時刻のトッパカーの位置情報から細胞を分画し、分画領域の輝度情報から活動状態の推定を行った。

### 細胞検出

画像中の細胞を数え上げ、その位置を決定するために、離散的な画像データを連続な関数で近似する。細胞体は基本的に球状であるため、画像を連続な関数で近似することができれば、細胞の中心位置はその連続な関数の局所最大点として定義できる。画像データでは、3 次元ボクセル座標上に輝度が与えられている。輝度は常に非負であることから、画像は輝度のヒストグラムとみなすことができる。よって 3 次元画像の平滑化は、3 次元の確率密度関数を推定する問題に置き換えることが可能である。

細胞体の形状は基本的に球状であることから、確率密度関数としてガウス混合モデルを当てはめることが考えられる。しかしながら、カルシウムイオン画像中の細胞体は、完全な楕円体から逸脱したものもあるため、ガウス混合モデルでは細胞が密集した領域を不自然に分解してしまう、BIC などによる成分数決定法がうまく機能しないなど、いくつかの弊害が生じる。従って、カルシウムイオン画像の平滑化には、ノンパラメトリックな手法を用いることが望ましい。本研究では、ガウスカーネルを用いて各立体画像をノンパラメトリックな密度関数に変換した。

目的: 細胞数のカウントと細胞体の位置の同定

仮定: 細胞体は「カーネル密度関数のローカルモードの周辺に存在する」

方法:

- i. 3D画像をカーネル密度関数に変換
- ii. ピークの数・位置  $\Leftarrow$  細胞体の個数・位置
- iii. ヒル・クライミングで「全ての」ローカルモードを発見

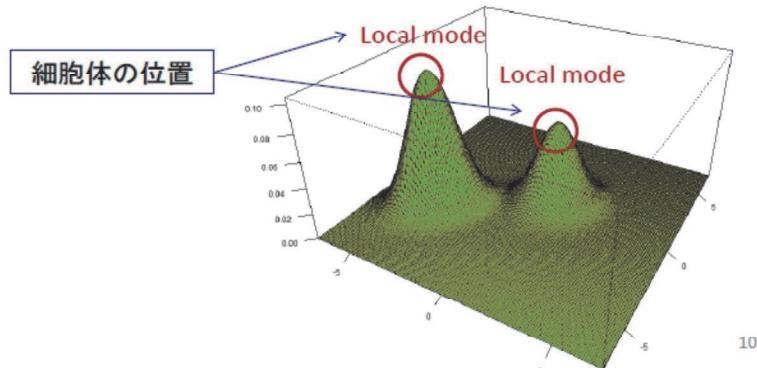
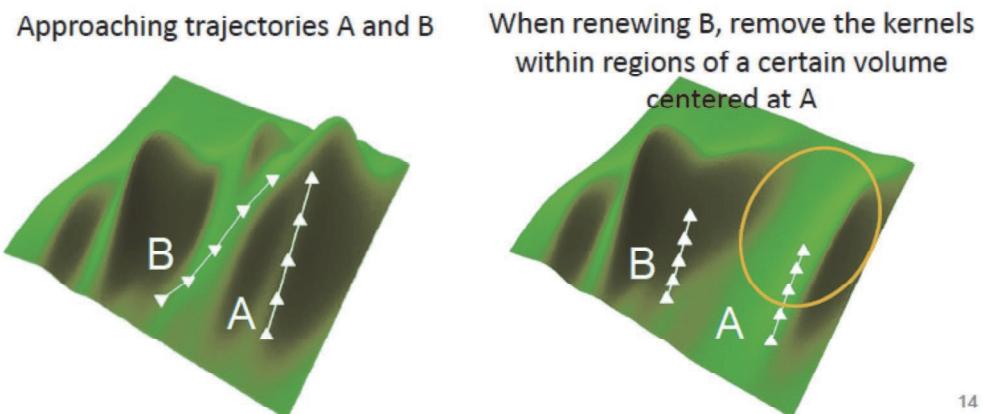


図 3 カーネル密度推定による細胞中心位置の検出

カルシウムイオン画像中の細胞を数え上げ、その位置を検出するには、画像中の球状物体を事前情報なしに検出可能な画像処理手法の開発が必要となる。いま、離散的な入力画像がカーネル密度推定法によって連続な関数へと近似されている。その場合、球状物体を検出する問題は、その局所最大点を検出する問題と等価である(図3)。局所最大点を検出するには、任意の初期点を設定し、EMアルゴリズムなど再帰的な最適化手法により近傍の山を登らせていいけば良い。一例としては、次に示すEMアルゴリズムに類似した再帰的なアルゴリズムが考えられる。EMアルゴリズムは、最近傍の局所最大点に到達するまで単調に山を登ることが保証されている。しかしながら、この種の最適化手法では、質量密度の大きな局所最大点へ収束し易い性質があるために、暗い物体は検知できずに見逃されてしまう可能性がある。そのため、すべての細胞を検出するには、非常に多くの初期点から探索を行う必要がある。

この問題を克服するには、複数の最適化を並列に走らせ、各探索点間に相互作用をさせることで、同じ局所最大点に収束するのを抑止する方法を考案した(Repulsive Parallel Hill-Climbingアルゴリズム; 図4)。この並列型の最適化計算は、GPU(Graphics Processing Unit)やMIC(Many Integrated Core)など、高データ並列処理が得意な計算機の活用により、比較的容易な計算効率の向上が可能である。



14

図 4 探索点に相互作用を持たせる最適化の概念図。左図:通常の最適化。右図:2つの探索点に反発を作用させる最適化。反発により、探索点Bは軌道が逸らされて、結果的に別の山を登る。

## 多細胞追跡

現状の4次元カルシウムイオンイメージングシステムでは、動態計測中の被検体（線虫）を完全に固定することができない。そのため、神経活動度の時間変化を計測するには、細胞の位置の動きを追跡する必要がある。このようなオブジェクトトラッキング技術としては、粒子フィルタや最近傍マッチング法などが有名である。しかしながら、カルシウムイメージングデータのように追跡対象としている物体が複数存在し、しかも密に分布している場合、単体物体の追跡よりも格段に問題が難しくなる。具体的には、物体が密に分布している画像では、オブジェクトを追跡していた複数のトラッカーが一つの物体に集中されてしまう「併合」や、ある物体を追跡していたトラッカーが別の物体に遷移してしまう「置換」が頻繁に発生してしまう。

複数物体の追跡で併合や置換の発生を抑止するには、物体の動きに対し何らかの拘束条件を加える必要がある。神経細胞の場合、細胞同士の相対的な位置関係が保存されることから、マルコフ確率場により動きをモデル化することが可能である。本研究では、マルコフ確率場で細胞位置の時空間変化をモデリングし、トラッカーの置換や併合を回避しながら、約120個の細胞の動きを追跡することに成功した（図5）。画像中の物体追跡はCV（computer vision）の分野で古くから研究されてきたが、100個以上の物体の一つ一つを正確に追跡できる方法は存在しない。この多物体追跡法は、神経科学のみならず様々な問題に応用されることが期待される。

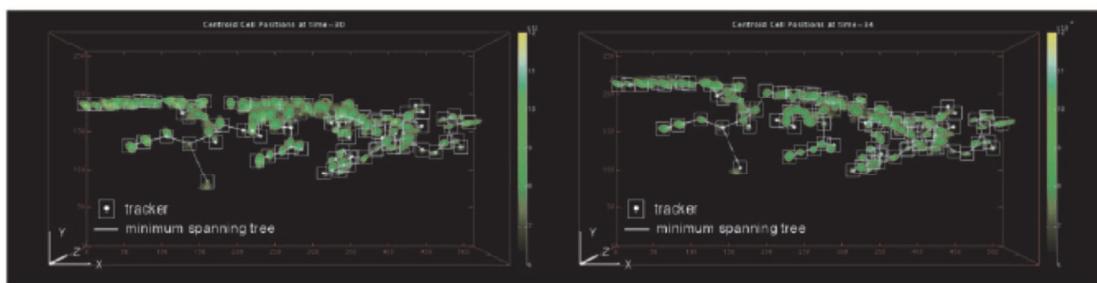


図5 マルコフ確率場によって動きをモデル化した複数細胞の同時追跡

## セグメンテーション

一つの細胞内のカルシウムイオン濃度の分布を定量化するために、どのボクセルまでが同一の細胞に属するROI（region of interest）を決定する必要がある。一般に、画像中の特定のROIを切り出す処理はセグメンテーション（分画）と呼ばれ、既存手法としてはwatershed法などが有名である。神経細胞ごとに活動度を定量化する場合、セグメンテーションが適切でないと複数の神経細胞活動度が混在する形で定量化されてしまうため、セグメンテーションには細心の注意を払う必要がある。

カルシウムイオン画像のように、複数のオブジェクトが密に分布し、背景とのコントラストが低い画像では、watershed法は過剰に分画する傾向ある。本研究では、低コントラストかつ物体が密に分布する画像データを適切に分画することができるセグメンテーション法を開発することに成功した（図6）。

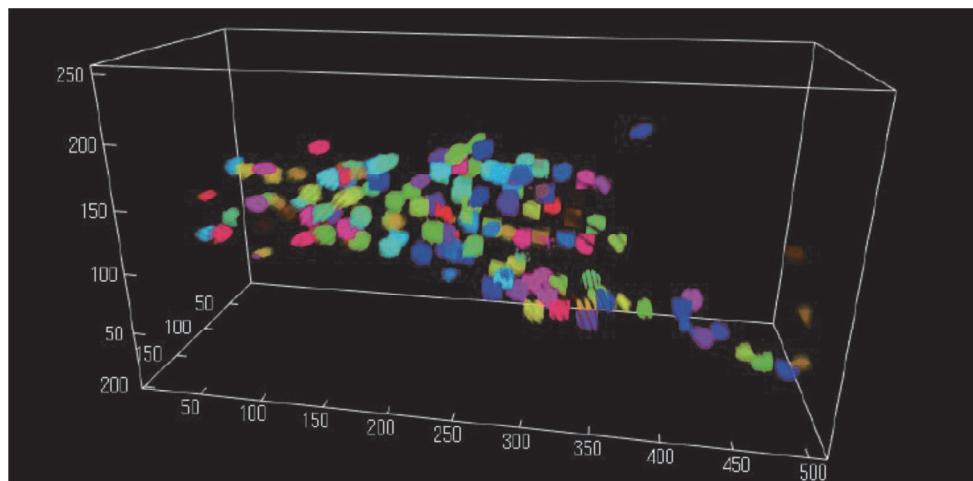


図6 ベイズアロケーション法による細胞核のセグメンテーション

## Monte Carlo Dynamic Classifier

### プログラム実行手順

# 1. はじめに

---

Monte Carlo Dynamic Classifier (MCDC) ツールのプログラム実行手順を説明する。このツールは、任意の観測データ系列のモデル推定、およびその推定モデルによる状態系列の推定を行うプログラムである。推定されたモデルは、異なる観測データ系列に適用してモデルの尤度を計算することにより、観測データ系列のクラス分類などに応用できる。MCDC ツールは、以下のプログラム群から構成される。

## MCDCTrain

モデル推定プログラム

## MCDCTest

モデルの尤度計算プログラム

## Graphs

推定されたモデルのグラフ描画関数群

以下の各章では、これらのプログラムの実行方法と実行例（第 2 章）、プログラム構成（第 3 章）について説明する。

## 2. プログラムの実行

MCDC ツールでは、観測データ系列に対してモデル推定を行う MCDCTrain と、推定モデルを用いて未知の観測データ系列の状態系列の推定を行う MCDCTest が提供する。

このほかに、推定モデルをグラフ描画する関数群を集めたツールとして Graphs クラスが提供する。本章ではこれらのプログラムの実行方法について説明する。

### 2.1. MCDCTrain

MCDCTrain 関数は、観測データ系列に対してモデル推定を行うプログラムである。

#### 2.1.1. 実行方法

MCDCTrain 関数は、以下のように実行する。

```
[ IDX, SKP, OKP, FV, GV, XE, YE, loglik ] = MCDCTrain ( ...  
    algorithm, ...  
    grids, ...  
    stateKernelGens, ...  
    obsKernelGens, ...  
    stateMeanFuncs, ...  
    obsMeanFuncs, ...  
    gridDimForGM, ...  
    splineHandle, ...  
    x0, ...  
    xaux, ...  
    u, ...  
    y', ...  
    N, ...  
    J, ...  
    K, ...  
    aspect ...  
) ;
```

また、以前の実行で出力されたファイルを読み込み、反復実行を継続することもできる。この場合は、MCDCTrain 関数を以下のように実行する。継続実行の際には、指定された matfile からパラメータを読み込み、前回と同じ設定で実行を再開する。ただし、Name, Value の組を指定することで、前回の設定を上書きして実行できる。

```
[ IDX, SKP, OKP, FV, GV, XE, YE, loglik ] = MCDCTrain ( matfile, aspect, Name, Value, ... )
```

#### 2.1.2. パラメータ

MCDCTrain 関数のパラメータは以下の通りである。

パラメータ	データ型	内容
algorithm	Algorithm	アルゴリズム種別
xGrids	G double[] cell	状態空間上に構成される格子点の座標。各次元の値を double

パラメータ	データ型	内容
		配列として格納したセル配列として表す
stateKernelGens	Dx handle cell	状態遷移関数のカーネル事前分布。Dx は状態空間のうち推定対象とされる次元数とする
obsKernelGens	P handle cell	観測関数のカーネル事前分布
gridDimForGramMatrix	int	グラム行列作成に用いる次元
splineHandle	handle	スプライン補間に用いるアルゴリズム。現在は GenericSpline のハンドルに固定
x0	Dx*1 double	状態変数の初期状態
xaux	Da*T double	追加状態データ
u	D*T double	制御データ
z	P*T double	観測データ
N	int	Particle filter 実行時の粒子数
J	int	MCMC iteration の全体の反復回数
K	int	観測オフセット
aspect	Aspect	システム設定情報 (ログ出力先等)

algorithm パラメータには、以下のいずれかのクラスを選択できる。

クラス名	アルゴリズム内容
GPPF	状態遷移関数と観測関数をガウス過程からのランダムサンプリングで生成する。パラメータ推定は行わない
PMMH	状態遷移関数と観測関数をガウス過程からのランダムサンプリングで生成する。パラメータ推定に Particle marginal Metropolis-Hastings 法を用いる
PMMH2	状態遷移関数と観測関数をガウス過程からのランダムサンプリングで生成する。パラメータ推定に Particle marginal Metropolis-Hastings 法を用いる。状態空間の特定の一次元のみを共分散関数に用いる

アルゴリズムとして PMMH2 を利用する場合には、さらに、平均値関数と共分散関数の学習方式、共分散関数に用いるカーネルパラメータの学習方式を以下から選択できる。

クラス名	アルゴリズム内容
MCDCStrategyChoice1	平均値関数を学習せず、常にアンカーモデルを利用する。共分散関数にカーネル関数を用いる
MCDCStrategyChoice2	現在の GP surface を平均値関数として関数をサンプリングする。共分散関数にカーネル関数を用いる
MCDCStrategyChoice3	現在の GP surface を平均値関数として関数をサンプリングする。共分散関数にカーネル関数を用いず、固定された共分散行列を用いる
RBFKernelGeneratorStrategyChoice1	RBF カーネルのパラメータを事前分布からランダムサンプリングする
RBFKernelGeneratorStrategyChoice2	RBF カーネルのパラメータを現在値からのランダムウォークによって生成

### 2.1.3. 戻り値

MCDCTrain 関数の実行結果として戻される値は、以下の通りである。

値	データ型	内容
IDX	A*1 double	j 回目の反復までの累積受理回数
SKP	A*D*2 double	第 a 受理反復における、状態遷移関数のカーネルパラメータ (sigma, l)
OKP	A*P*2 double	第 a 受理反復における、観測関数のカーネルパラメータ (sigma, l)
FV	A*D*G double	第 a 受理反復における、状態遷移関数による格子点上の値。 G は格子点のサイズに一致する多次元配列
GV	A*P*G double	第 a 受理反復における、観測関数による格子点上の値。 G は格子点のサイズに一致する多次元配列
XE	A*T*D double	第 a 受理反復における、時刻 t の状態変数の推定平均
YE	A*T*P double	第 a 受理反復における、時刻 t の観測変数の推定平均
loglik	A*1 double	第 a 受理反復における、particle filter による推定の対数尤度

### 2.1.4. 中間ファイル

実行時には、aspect の設定にしたがって中間状態が保存される。これは.mat 形式のダンプファイルとして出力される。ファイルに格納されているデータは以下の通りである。

値	データ型	内容
algorithm	Algorithm	MCDCTrain のパラメータに渡された algorithm
in	MCDCInput	MCDCTrain のパラメータのうち、algorithm と aspect を除くすべて
out	MCDCOutput	MCDCTrain の戻り値のすべて（途中状態）および、現在の反復回数 j

### 2.1.5. ログファイル

実行中には、aspect の設定にしたがってログファイルが出力される。これはテキスト形式のファイルになる。ログファイルは以下のような形式になる。

```
2014/04/26 11:29:36 - Iteration 18 / 200
2014/04/26 11:29:36 - StateKernel[1]: [ Sigma=8.932992, L=7.409991 ]
2014/04/26 11:29:36 - StateKernel[2]: [ Sigma=4.410584, L=2.304895 ]
2014/04/26 11:29:36 - ObsKernel[1]: [ Sigma=1.711191, L=9.248074 ]
2014/04/26 11:29:36 - ObsKernel[2]: [ Sigma=9.867704, L=9.599577 ]
2014/04/26 11:29:36 - ObsKernel[3]: [ Sigma=7.700470, L=8.679293 ]
2014/04/26 11:29:36 - Creating GramMatrix using 1 dim...
2014/04/26 11:29:36 - StateKernel[1] Done
2014/04/26 11:29:36 - StateKernel[2] Done
2014/04/26 11:29:36 - ObsKernel[1] Done
2014/04/26 11:29:36 - ObsKernel[2] Done
2014/04/26 11:29:36 - ObsKernel[3] Done
2014/04/26 11:29:36 - Drawing GP surface...
```

```

2014/04/26 11:29:36 - Estimating using particle filter... (N=500)
2014/04/26 11:30:20 - Acceptance log probability = 232745.510426
2014/04/26 11:30:20 - logLH = -5347068.232758, accepted
2014/04/26 11:30:20 - Elapsed time is 44.324939 seconds.
2014/04/26 11:30:20 - j: 8 bytes
2014/04/26 11:30:20 - IDX: 1600 bytes
2014/04/26 11:30:20 - SKP: 128 bytes
2014/04/26 11:30:20 - OKP: 192 bytes
2014/04/26 11:30:20 - FV: 61504 bytes
2014/04/26 11:30:20 - GV: 92256 bytes
2014/04/26 11:30:20 - XE: 122752 bytes
2014/04/26 11:30:20 - YE: 184128 bytes
2014/04/26 11:30:20 - loglik: 32 bytes

```

## 2.2. MCDCTest

MCDCTest 関数は、MCDCTrain によって得られたモデルを用いて未知のデータの状態を推定する。複数の異なるモデルを用いて状態推定を行い、それぞれの尤度を比較することで、クラス分類問題への応用も可能である。

### 2.2.1. 実行方法

MCDCTest 関数は、以下のように実行する。

```
[ result, FnState, FnObs ] = MCDCTest (u, y, N, modelFile)
```

### 2.2.2. パラメータ

MCDCTest 関数のパラメータは以下の通りである。

パラメータ	データ型	内容
u	D*T double	制御データ
y	P*T double	観測データ
N	int	Particle filter 実行時の粒子数
modelFile	chars	モデルファイル

### 2.2.3. 戻り値

MCDCTest 関数の実行結果として戻される値は、以下の通りである。

値	データ型	内容
result	ParticleFilter	j回目の反復までの累積受理回数
FnState	handle	状態遷移関数
FnObs	handle	観測関数

戻り値の result に、particle filter による状態推定の結果が含まれる。これは以下の構造を持つ。

値	データ型	内容
Particles	N*T*D double	時刻 t における各粒子の座標
Weights	N*T double	時刻 t における各粒子の重み
Loglik	double	推定された状態の対数尤度

## 2.3. Graphs

MCDCTrain によって推定されたモデルは、Graph クラスに定義された関数群を用いて PDF ファイルに出力することができる。以下のグラフを出力可能である。

### 2.3.1. Graphs.YE

観測データ系列と、推定モデルを用いた粒子フィルタによって観測データを追跡した結果の時間推移グラフを出力する。MCMC iteration の中で、指定された特定の反復における推定モデルを用いた結果を出力する。観測データ系列の次元ごとにグラフが出力する。以下のように実行する。

```
Graphs.YE ( ...  
    outputFileNamePrefix, ...  
    matFileName, ...  
    iterations, ...  
    times ...  
);
```

パラメータは以下の通りである。

パラメータ	データ型	内容
outputFileNamePrefix	char[]	出力グラフファイルのファイル名接頭辞。ファイル名は、指定された接頭辞に、次元数と拡張子.pdf を付け加えたもの
matFileName	char[]	推定モデルを含む Matlab データファイル名
iterations	int[]	プロットする推定値の MCMC iteration を列挙した配列
times	int[]	プロットするするデータ系列の範囲。省略時はデータ系列の全体を出力

### 2.3.2. Graphs.XE

推定モデルを用いた粒子フィルタによる状態系列推定値の時間推移グラフを出力する。MCMC iteration の中で、指定された特定の反復における推定モデルを用いた結果を出力する。状態系列の次元ごとにグラフが出力される。以下のように実行する。

```
Graphs.XE ( ...  
    outputFileNamePrefix, ...  
    matFileName, ...  
    iterations, ...  
    times ...  
);
```

### 2.3.3. Graphs.YEMean

観測データ系列と、推定モデルを用いた粒子フィルタによって観測データを追跡した結果の時間推移グラフを出力する。MCMC iteration の全体を平均した推定モデルを用いる。観測データ系列の次元ごとにグラフが出力される。以下のように実行する。

```
Graphs.YEMean ( ...  
    outputFileNamePrefix, ...  
    matFileName, ...  
    times ...  
);
```

パラメータは以下の通りである。

パラメータ	データ型	内容
outputFileNamePrefix	char[]	出力グラフファイルのファイル名接頭辞。ファイル名は、指定された接頭辞に、次元数と拡張子.pdfを付け加えたもの
matFileName	char[]	推定モデルを含む Matlab データファイル名
times	int[]	プロットするデータ系列の範囲。省略時はデータ系列の全体を出力

#### 2.3.4. Graphs.XEMean

推定モデルを用いた粒子フィルタによる状態系列推定値の時間推移グラフを出力する。MCMC iteration の全体を平均した推定モデルを用いる。状態系列の次元ごとにグラフが出力される。以下のように実行する。

```
Graphs.XEMean ( ...
    outputFileNamePrefix, ...
    matFileName, ...
    times ...
);
```

#### 2.3.5. Graphs.Loglik

モデル推定において、MCMC iteration ごとに推定されたモデルの対数尤度を出力する。

```
Graphs.Loglik ( ...
    outputFileNamePrefix, ...
    matFileName1, ...
    matFileName2, ...
    ...
);
```

パラメータは以下の通りである。

パラメータ	データ型	内容
outputFileNamePrefix	char[]	出力グラフファイルのファイル名接頭辞。ファイル名は、指定された接頭辞に.pdfを付け加えたもの
matFileName1, 2, ...	char[]	推定モデルを含む Matlab データファイル名。複数ファイルを指定した場合は、それぞれを一系列としてグラフを描画

#### 2.3.6. Graphs.Rmse

モデル推定において、各 MCMC iteration までの推定モデルを用いて観測データ系列を推定した結果の平均二乗誤差を出力する。以下のように実行する。

```
Graphs.Rmse ( ...
    outputFileNamePrefix, ...
    matFileName ...
);
```

パラメータは以下の通りである。

パラメータ	データ型	内容
outputFileNamePrefix	char[]	出力グラフファイルのファイル名接頭辞。ファイル名は、指定された接頭辞に.pdfを付け加えたもの
matFileName	char[]	推定モデルを含む Matlab データファイル名

### 3. プログラム構成

MCDC ツールのプログラムファイルの一覧は以下の通りである。

ファイル名	内容
Algorithm.m	モデル推定に用いるアルゴリズムを表す抽象基底クラス
Aspect.m	プログラム動作を定める設定クラス
BoundedNormalDistribution.m	正の範囲に限定された正規分布
CheckGridTransformation.m	グリッドの写像に対する範囲チェック関数
Distribution.m	確率分布を表す抽象基底クラス
GPPF.m	モデル推定アルゴリズム。状態遷移関数と観測関数をガウス過程からのランダムサンプリングで生成する。パラメータ推定は行わない
GPSurface.m	グリッドの写像を求める関数
GenericSpline.m	多次元スプライン補間関数。spapi 関数を用いる
Graphs.m	グラフ描画関数群
GridData.m	状態空間におけるグリッド構造
Kernel.m	カーネル関数（抽象基底クラス）
KernelGenerator.m	カーネル関数生成器（抽象基底クラス）
LogMvnPdf.m	多変量正規分布の確率密度関数の計算
MCDCInput.m	モデル推定の入力データ
MCDCMatFile.m	モデル推定の中間ファイル出力クラス
MCDCOuput.m	モデル推定の出力結果
MCDCStrategy.m	MCMC の動作を定める抽象基底クラス
MCDCStrategyChoice1.m	MCMC の動作を定めるクラス。平均値関数を学習せず、常にアンカーモデルを利用する
MCDCStrategyChoice1_GPPF.m	MCMC の動作を定めるクラス。平均値関数を学習せず、常にアンカーモデルを利用する
MCDCStrategyChoice2.m	MCMC の動作を定めるクラス。現在の GP surface を平均値関数として関数をサンプリングする。
MCDCStrategyChoice3.m	MCMC の動作を定めるクラス。現在の GP surface を平均値関数として関数をサンプリングする。共分散関数を学習せず、固定された共分散行列を用いる
MCDCTest.m	推定モデルの尤度計算プログラム
MCDCTrain.m	モデル推定プログラム
ModelFunctions.m	グリッドの写像に対するスプライン関数
ModelFunctions2.m	グリッドの写像に対するスプライン関数
NormalDistribution.m	正規分布
PMMH.m	モデル推定アルゴリズム。状態遷移関数と観測関数をガウス過程からのランダムサンプリングで生成する。

ファイル名	内容
	パラメータ推定に Particle marginal Metropolis-Hastings 法を用いる
PMMH2.m	モデル推定アルゴリズム。状態遷移関数と観測関数をガウス過程からのランダムサンプリングで生成する。 パラメータ推定に Particle marginal Metropolis-Hastings 法を用いる。状態空間の特定の一次元のみを共分散関数に用いる
PMMHParticleFilter.m	粒子フィルタ。PMMH 法で利用するために Ancestor sampling を行う
ParticleFilter.m	粒子フィルタ
PlotGraph.m	グラフ描画サブルーチン。Graphs.m から利用される
RBFKernel.m	RBF カーネル。ガウス過程の共分散関数として利用する
RBFKernelGenerator.m	RBF カーネル関数生成器
RBFKernelGeneratorStrategy.m	RBF カーネル関数の生成方法を定める抽象基底クラス
RBFKernelGeneratorStrategyChoice1.m	RBF カーネル関数生成アルゴリズム。カーネルパラメータを事前分布からランダムサンプリングする
RBFKernelGeneratorStrategyChoice2.m	RBF カーネル関数生成アルゴリズム。カーネルパラメータを現在値からのランダムウォークによって生成する
UniformDistribution.m	一様分布
VectorValuedFunction.m	各次元のスカラー値関数をベクトル値関数にまとめるルーチン

