

# 粒子フィルタとプライシングによる時系列区分化の手法を用いた非線形時系列生成モデル変化の推定

時永 祥三<sup>1</sup> 池田 欽一<sup>2</sup>

**概要:** まず、時系列が非線形状態方程式により生成され、その観測データから粒子フィルタ (PF) により状態を推定する問題を定義し、尤度の時間変化を用いて時系列を区分化し、時系列生成モデルの時間変化を推定する方法を示す。この場合、区分化を行うための時間軸は、プライシングと呼ばれる、ある離散系カオス変動により特徴付けられることを用いている。これにより、ランダムな探索と比較して効率が向上することを示す。次に非線形モデルの時間変化をオンラインで行う 1 つの方法として、現在の関数形状が遺伝的操作により変形されるとする仮定を用いて、遺伝的プログラミング (Genetic Programming: GP) の手法により、変化の発生と変化後の関数を推定する方法を提案する。応用例として人工的なデータを用いて、本報告の手法の有効性を確認するとともに、現実の時系列データに適用して、モデル変化の後の状態方程式推定を議論する。

## 1. はじめに

観測された時系列に基づいて、時系列を生成するパラメータの時間変化を推定する場合に、時系列を区分化をする手法を用いる研究が示されている [1]。また、線形システムのモデルにおいても、記述するパラメータの時間的な変化を推定する手法が示されている [2]。しかしながら、非線形システムとして記述されるモデルから生成された時系列 (以下では時系列生成モデルと呼ぶことにする) に対して、これらの方法を直接的に用いることはできないため、別の方法による時系列区分化と生成モデル変化の推定手法を用いる必要がある。これまでの研究で、統計的手法として、観測された時系列データに対する非線形状態方程式により得られる尤度を用いて、システムの状態方程式の変化を抽出する方法が示されているが、ある時点でのシステムの変化だけを議論している限界がある [3][4]。非線形時系列生成モデルの時間変化推定には、これまで多くの分野で適用されてきた、粒子フィルタ (Particle Filter: PF) 手法が有効である [4][5][6]。本報告では PF と時系列区分化の手法を用いた非線形時系列生成モデルの時間変化の推定について述べる。

本報告ではまず、観測される時系列が非線形状態方程式により生成され、その観測データから PF により状態を推定する問題を定義し、尤度を用いて、いくつかの候補となる非線形システム記述の、どれが最適かを判断する方法を適

用する。この場合、区分化を行うための時間軸は、プライシングと呼ばれる、ある離散系カオス変動により特徴付けられることを仮定として用いている。すなわち非線形システムが時間的に変化する背景にある、決定論にしがうメカニズムが存在すると仮定する [7][8][9]。これにより、ランダム探索を行う場合と比較して効率が向上することを示す。このような時系列の区分化の方法は、まとまったデータに対してオフラインで行うに適した方法であるが、一方では瞬間的に発生するシステムの変化抽出には適さない。したがって、次に、非線形モデルの時間変化をオンラインで行う 1 つの方法として、変化後の関数形状を遺伝的プログラミング (Genetic Programming: GP) の手法により推定する方法を提案する [10]。応用例として人工的なデータを用いて、本報告の手法の有効性を確認するとともに、現実の時系列データに適用して、モデル変化の後の状態方程式推定を議論する。

## 2. 非線形時系列生成モデルの時間変化の推定

### 2.1 非線形時系列生成モデルと時間変化

以下ではまず非線形状態方程式により生成される時系列モデルを記述しておく [4][6]。本報告では時系列生成を記述するモデルは 1 つではなく、時間的に変化することを仮定している。時系列を記述する時間は同じ長さ 1 の区間に分割されており、また、いくつかの区間にわたり同じ時系列生成モデルが適用され、 $j$  番目のセグメントを  $s_j$  と呼んでおく。1 つのセグメントでは 1 つの非線形時系列生成モデルが適用 (選択) され、選択されるモデルを区別する記号 (クラ

<sup>1</sup> 九州大学大学院  
Kyushu Univ.

<sup>2</sup> 北九州市立大学  
The Univ. of Kitakyushu

ス) を  $z_k, k = 1, 2, \dots, K$  としておく (総数が  $K$  個).

いま 1 つのセグメントにおける非線形時系列生成モデルは、次のように記述できる (適用されるモデルのクラスを  $z_k$  としておく).

$$\begin{aligned} x(t+1) &= F[x(t)]^{(z_k)} + u(t)^{(z_k)}, \\ x(t) &= [x_1(t), x_2(t), \dots, x_m(t)], \\ u(t) &= [u_1(t), u_2(t), \dots, u_m(t)]. \end{aligned} \quad (1)$$

$$y(t) = G[x(t)]^{(z_k)} + v(t)^{(z_k)}. \quad (2)$$

ここで、変数  $y(t)$  は時刻  $t$  における観測データを意味するスカラー (1次元ベクトル) であり、変数  $x(t)$  は推定されるべき  $m$  次元の状態変数ベクトルである。また  $u(t)^{(z_k)}, v(t)$  は  $m$  次元、1次元のシステムノイズである。 $F[\cdot]^{(z_j)}, G(\cdot)^{(z_k)}$  は、変数  $x(t)$  についての関数である。ここで、 $G(\cdot)^{(z_k)}$  は既知の非線形関数であり、関数のセットから選択されると仮定する。

## 2.2 PF による状態推定とモデルの尤度

記号として、 $x_i(t), i = 1, 2, \dots, m$  の  $j$  番目の粒子を  $x_i^{(j)}(t)$  としておく。

(1) ステップ 1: 粒子の初期値の生成

時刻 0 における変数  $x_i^{(j)}(t)$  の初期値  $x_i^{(j)}(0), i = 1, 2, \dots, m$  について、あらかじめ定めた確率分布に従う乱数により  $N_P$  セットの粒子の集合として生成する。

(2) ステップ 2: システムノイズの生成

それぞれの粒子について、変数に対する粒子  $x_i^{(j)}(0)$  と同様に、システムノイズである  $u_i^{(j)}(0)$  および  $v^{(j)}(0)$  について、あらかじめ与えた確率分布に従った乱数によって  $N_P$  セットの粒子として生成する。

(3) ステップ 3:  $x_{k,t}^{(j)}(t)$  の 1 時刻先予測

$x_i^{(j)}(t)$  の 1 時刻先予測が、次により得られる。

$$x_i^{(j)}(t+1) = F[x_i^{(j)}(t)] + v^{(j)}(t). \quad (3)$$

ただし上の式はベクトル  $x^{(j)}(t+1)$  のそれぞれの要素  $x_i^{(j)}(t+1)$  について計算することを意味する。

(4) ステップ 4: 重みの計算

次に、それぞれ粒子  $j$  の重み  $w_t^{(j)}$  を計算する。

$$w_t^{(j)} = R_t(y(t+1)|x^{(j)}(t+1)). \quad (4)$$

ここで  $R_t(\cdot)$  は  $x^{(j)}(t+1)$  が与えられた場合の  $y(t+1)$  の条件付分布である。これらの重みを、集計しておく。

$$W_t = \sum_{j=1}^{N_P} w_t^{(j)}. \quad (5)$$

(5) ステップ 5: 粒子のリサンプリング

それぞれの粒子を、重みに従って、リサンプリングする。すなわち  $i$  番目の状態変数については、粒子の集合  $X_{t+1} = (x_i^{(1)}(t+1), x_i^{(2)}(t+1), \dots, x_i^{(N_P)}(t+1))$  から粒子を  $w_t^{(j)}/W_t$  の確率に従って復元抽出して、生成する。

## 3. 時系列区分化と非線形時系列生成モデル変化

### 3.1 プライシングを用いた時系列区分化手法

サービスを処理する窓口をノードと考え、ノードでのサービスの処理能力 (容量) を  $\mu$  としておく [7][8][9]。ノードに流入するフロー  $\lambda(t)$  は、ノードでの待ち時間や処理コストの予測値  $\pi(t)$  (pricing: プライシングと呼ばれる) に関連して調整される。フロー  $\lambda(t)$  はノードに対してすべてが供給されるものではなく、この部分が供給されるモデルとなる。 $\lambda(t)$  は、次の式により与えられる ( $a > d$ )。

$$\lambda(t) = \begin{cases} 1, & (0 \leq \pi(t) \leq d), \\ (a - \pi(t))/(a - d), & (d < \pi(t) < a), \\ 0, & (\pi(t) \geq a). \end{cases} \quad (6)$$

プライシングの値は、指数平滑和により与えられる ( $0 < \omega < 1$ )。

$$\pi(t+1) = \omega\pi(t) + (1 - \omega)p(t), p(t) = Q_\mu(\lambda(t)). \quad (7)$$

$$Q_\mu(\lambda(t)) = \frac{1}{(1 - \lambda(t)/\mu)^\mu}. \quad (8)$$

関数  $Q_\mu(\lambda(t))$  は、いわゆる待ちシステムにおいて入力フローが到着率  $\lambda$  のポアソン到着、処理時間の分布が指数分布である場合の平均待ち時間である。

図 1 には、プライシングとして生成されるカオス時系列の例を示している。図 1 に示すように、時系列は間歇的に大きな値をとり、この時点で時系列モデルの変化が発生したとみなす。また分岐図は省略するが、パラメータが  $0.2 < \mu < 0.9$  となる範囲では変数  $\pi(t)$  はカオス変動にしたがった挙動となる。したがって  $\mu$  の値を調整することにより、間歇的に大きな値をとる時刻 (ジャンプの発生時刻) の間隔を調整することができる。

なお図 1 に示すように、プライシング時系列が大きな値をとる間隔は 10~20 であり、これをそのまま非線形時系列の生成における変化発生の時間間隔に用いると、時系列変化が激しくなり、推定が難しいケースが生じる。そのため本報告では、プライシングにおける時間間隔  $\Delta_P$  と、非線形時系列のモデル変化における時間間隔である  $\Delta_T$  の関係を、 $\Delta_P = 5\Delta_T$  としておく。これにより観測される非線形時系列の変化は、緩やかになる。

### 3.2 システム構成の概要

本報告の手法において推定すべきパラメータは、次のものである。プライシング、すなわちセグメントの最適分割に関しては、容量  $\mu$ 、パラメータ変化時点を与える閾値  $P_J$ 、プライシング時系列の初期値  $\pi(0)$  の推定を行う必要がある。これらのパラメータの推定には、よく知られている遺伝的アルゴリズム (Genetic Algorithm: GA) 手法を適用す

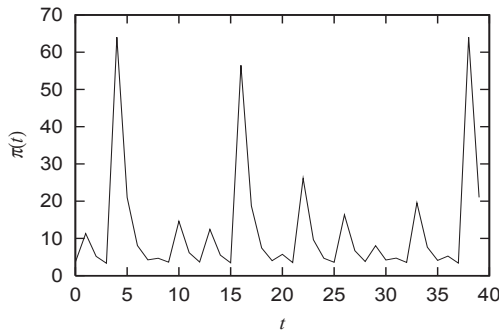


図 1 プライシング時系列の例

Fig. 1 An example of pricing time series

る。一方、これらのパラメータの推定と並行して、それぞれのセグメントに割り当てる非線形時系列生成の状態方程式のクラス  $z_k$  の推定を行う必要がある。これら 2つの手順は、同時には進められないので、ある時刻では一方を固定しておいて、他方を最適化する手法を用いる。

(ステップ 1) プライシングの初期値 (初期個体) の設定

プライシングの特性を推定する場合には、カオス変動を規定するパラメータである待ち行列の容量  $\mu$ 、変数の初期値  $\pi(0)$  のほかに、プライシング閾値  $P_J$  を推定対象とする必要がある。まず GA 手法の適用にあたり  $\pi(0)$ 、 $\mu$ 、 $P_J$  の値の候補となる個体 (初期個体) を複数個ランダムに生成しておく。

(ステップ 2) 個体からのプライシング時系列の生成とセグメントへの分割

それぞれの個体に対応して、プライシング時系列の生成と時間軸のセグメントへの分割を行う。具体的には、それぞれの個体が与える  $\mu$ 、 $P_J$ 、 $\pi(0)$  の設定が最適であると仮定して、式 (7)(8)(9) にしたがって 1つの時系列をプライシング時系列として生成し、時系列が閾値  $P_J$  を超える時点を検出し、時間軸をいくつかのセグメントへと分割する。

(ステップ 3) セグメント  $s_j$  へのクラス  $z_k$  の割り当て

次に、個体から生成されるこれらのそれぞれのセグメント  $s_j$  において、尤度が最大となるようにセグメントに割り当てる時系列生成モデルのクラス  $z_k$  を決定する。このような計算によって得られる時系列全体についての尤度を、この個体の適合度とする。

(ステップ 4) 個体への遺伝的操作

それぞれの個体の適合度が得られたあとで、GA における遺伝的操作を用いて個体の性能を改善する。

(ステップ 5) 手順の繰り返し

推定のレベルがあらかじめ設定した基準に達したら、繰り返しを停止する。そうでない場合には、ステップ 2に戻る。

なお本報告においては、これ以降、プライシングにおけるパラメータについて、次のように設定して、シミュレーションを実施する。この条件を、プライシングによるパラメータ設定とする (パラメータ設定 P と呼ぶことにする)。以下では

この記号 P により、シミュレーション条件を引用する。 $\mu$  は  $\mu = 0.35$  などの特定の値に設定する場合もあるので、このときには説明文において明示する。

パラメータ設定 P:

$P_J = 10 \sim 30, \mu = 0.3 \sim 0.7$

$\pi(0)$ : 0.5 ~ 15 の間の一様乱数

## 4. 応用例

### 4.1 4つの非線形システムにおける時間変化

以下のように定義される 4つの非線形システム (Case I から Case IV としておく) を分析の対象とする。最初に、これらのシステムの概要についてまとめておく。なお説明において、クラスを  $z_1, z_2, \dots$  として表記し、それぞれのクラスにおける非線形システムを区別している。

Case I: 離散系カオス変動から導出された方程式

離散系カオス変動のシステムとして、ロジスティック写像、エノン写像の 3つを用いる ( $K = 3$  となる)。しかし、これらの写像そのものを用いると、カオス変動だけが強調されるため、本報告で行うような分析には適切ではない、不自然な時系列になる。したがって、カオスである制限を緩めるため係数を少し修正 (変形) し、しかもランダム変数  $u(t), v(t)$  の影響を導入した、以下の 3つのシステム方程式に変換する。なお以下の 3つのシステムにおいて観測データは  $y(t) = x(t) + v(t)$ 、すなわち  $G(x(t))^{(z_k)} = x(t)$  により生成されると仮定する。

$z_1$ : ロジスティック写像の変形

$$F(x(t)) = 3x(t)(1 - x(t)). \quad (9)$$

$z_2$ : ベルヌーイシフト写像の変形

$$F(x(t)) = \begin{cases} 1.5x(t), & (0 < x(t) < 0.5), \\ 1.5x(t) - 0.5, & (0.5 \leq x(t) < 1). \end{cases} \quad (10)$$

$z_3$ : テント写像の変形

$$F(x(t)) = \begin{cases} 1.5x(t), & (0 \leq x(t) \leq 0.5), \\ 1.5 - 1.5x(t), & (0.5 < x(t) \leq 1). \end{cases} \quad (11)$$

3つのシステムについて  $\sigma_u = 0.01, \sigma_v = 0.01$  とする。

Case II: プライシング時系列

もともとプライシング時系列は、本報告で示すモデル推定の方法論においては、システムの変化を記述するためのジャンプ系列の生成に用いられている。しかしながら、容量  $\mu$  を変化させると、ジャンプの発生間隔の平均値が異なる時系列となるので、この性質を利用した非線形時系列の生成システムを定義する。具体的には、式 (8) に示すプライシング時系列の生成モデルにおいて、容量を  $\mu = 0.35, 0.55, 0.75$  の 3つに変化させた場合を 3種類のカオス変動として定義する ( $K = 3$  となる)。また分析対象としては、特にカオス時系列である必要はないので、ランダム変数を加えた表現としている。  $\sigma_u = \sigma_v = 0.1$  とする。

Case III:製品の発注の時系列

Case III においては、時系列  $y(t)$  が小売業者から製造業者へと伝達される製品製造の数量を表現する場合に、小売業者が用いている需要予測を推定する問題を取り扱う [11]. 記号を次のように定義する. $x(t)$ :小売業者の推定する需要,  $y(t)$ :小売業者が製造業者に出す注文, 関数  $G(x(t))$  で変形される. 小売業者は市場の需要を  $x(t)$  にしたがって予測し, これをもとに製造業者に対して製造数量を伝える. しながら小売業者は, そのままの最適な数量ではなく, 自身が修正を加えた数量  $y(t)$  を, 製造業者に伝達する (3つの方法で適切な需要は変形されるので,  $K = 3$  となる). 小売業者がとるべき最適な行動を含めて, この詳細をここで述べるのは適切ではないので, 付録 B にまとめておく.

変数のプロセス  $x(t)$  は 1 次の自己回帰モデルにしたがって次のように生成される.

$$x(t+1) = d + \rho x(t) + u(t). \quad (12)$$

パラメータ  $d, \rho$  はある範囲で自由に選ぶことはできるがシミュレーションの結果には大きな影響を与えないので, 以下では  $d = 3, \rho = 0.5, \sigma_u = 1, \sigma_v = 0.5$  としておく. 時間遅れの導入のための変数を  $s(t) = x(t-1)$  としておく. 小売業者が注文すべき最適な製品数量  $r(t)$  は決定されるが, これが小売業者の意思により関数を介して変形される. 小売業者が何らかの変更を加えない場合には, 数量  $y(t) = r(t)$  として製造業者に注文される ( $l$  は小売業者があらかじめ販売を予測する期間であり, リードタイムと呼ばれる).

$$r(t) = x(t) + c_R[x(t) - x(t-1)], c_R = \frac{\rho(1 - \rho^{l+1})}{1 - \rho}. \quad (13)$$

しながら小売業者は, 自身の利益を考慮して, 最適な製造数量を修正する. この場合の行動は, 次に示す変換関数  $G(z(t))$  として記述できると仮定する. このような変換は, 次のような単調増加関数からランダムに選択され, 状態変数から観測時系列が生成される.

$$\begin{aligned} z_1: G(r)^{(z_1)} &= r \\ z_2: G(r)^{(z_2)} &= 6 - r^{1.2}/2.0 (0 \leq r \leq 6), \\ &G(r)^{(z_2)} = 6 + r^{1.2}/2 (r > 6) \\ z_3: G(r)^{(z_3)} &= 0 (\sum_{d=D_L}^0 x(t-d) < 40), \\ &G(r)^{(z_3)} = r (r > 40) \end{aligned}$$

クラス  $z_1$  は予測される需要を変更しないで発注する場合であり, クラス  $z_2$  は予測される需要を 6 を境界として 6 より大きい (小さい) ときに, より多く (少なく) 発注する処理を行う場合である. クラス  $z_3$  は需要の予測量を保留しておいて, 前回発注を行った時刻 (時間  $D_L$  前としておく) から需要予測の累計をとり, その累計が一定の値 (40 としておく) を超過した時点で, 一気に注文を出す場合を示している.

Case IV:構造変化検証のためのモデル

Case IV において用いる非線形システムは, これまで非

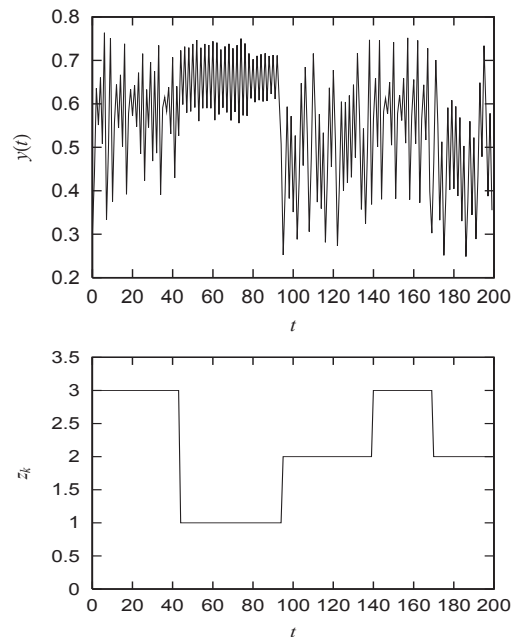


図 2 プライシングにしたがって生成された観測時系列とクラス  $z_k$  の例 (Case I)

Fig. 2 An example of generated time series  $y(t)$  and class  $z_k$  governed by pricing (Case I).

線形システムにおける構造変化の推定問題において, 例題として用いられたものである. システムの挙動は, 以下の式により記述される. 本報告では, このシステム記述において, パラメータ  $b$  の値を 3 つに設定した場合の時系列を用いる.

$$x(t) = ax(t-1) + b \frac{x(t-1)}{(1+x(t-1))^2} + 8 \cos[1.2(t-1)] + u_{t-1}. \quad (14)$$

$$y(t) = 0.05x(t)^2 + v_t. \quad (15)$$

それぞれのクラスにおけるパラメータの設定は, 以下のようにする.  $z_1 : b = 25, z_2 : b = 12.5, z_3 : b = 1.2. \sigma_u = 0.5, \sigma_v = 0.5$  とする.

4.2 4つのケースについての時間変化と時系列

これまで定義した 4 つの非線形システムにおける時間変化について, あらかじめ定められたプライシングにしたがってシステム変化の境界を設定し, これらの時点においてクラスの切り替え (ランダムにクラスを選択する) を行った場合の, 観測時系列  $y(t)$  の事例を示しておく (スペースの関係で Case I についてのみ). 以下に示す図 2 の上の部分は Case I に対応する観測時系列  $y(t)$  であり, 図の下部分にはクラス  $z_k$  の時間変化を示している. 話を簡単にするために, クラス  $z_k$  を数値の  $k$  に対応させて, 示している. これらの図より分かるように, 時間的に非線形システムの切り替えが行われ, プライシングにしたがって生成される境界を挟んで, 様相が異なる時系列が出現している.

表 1 パラメータ  $\pi(0), \mu, P_J$  の推定結果 (相対誤差), (パラメータ設定 P を利用).

Table 1 Result of estimation of parameters  $\pi(0), \mu, P_J$ (relative error) (by using parameter setting P).

| Case I   |       |         |       | Case II |       |         |       |
|----------|-------|---------|-------|---------|-------|---------|-------|
| $\mu$    | $R_I$ | $R_\mu$ | $R_J$ | $\mu$   | $R_I$ | $R_\mu$ | $R_J$ |
| 0.3      | 0.08  | 0.06    | 0.07  | 0.3     | 0.08  | 0.08    | 0.08  |
| 0.5      | 0.08  | 0.07    | 0.07  | 0.5     | 0.09  | 0.08    | 0.08  |
| 0.7      | 0.08  | 0.07    | 0.08  | 0.7     | 0.09  | 0.09    | 0.09  |
| Case III |       |         |       | Case VI |       |         |       |
| $\mu$    | $R_I$ | $R_\mu$ | $R_J$ | $\mu$   | $R_I$ | $R_\mu$ | $R_J$ |
| 0.4      | 0.08  | 0.06    | 0.07  | 0.4     | 0.08  | 0.08    | 0.08  |
| 0.55     | 0.08  | 0.07    | 0.07  | 0.55    | 0.09  | 0.08    | 0.08  |
| 0.7      | 0.08  | 0.07    | 0.08  | 0.7     | 0.09  | 0.09    | 0.09  |

#### 4.3 エンジンデータを用いたパラメータ変化の推定

シミュレーションの条件を、以下のように設定する  
容量であるパラメータ:  $\mu = 0.3, 0.5, 0.7$

GA 個体における変数のストリング長: 8 ビット

表 1 には、パラメータ  $\pi(0), \mu, P_J$  の値を既知として与えた場合の推定結果 ( $\hat{\pi}(0), \hat{\mu}, \hat{P}_J$  としておく) について、容量  $\mu$  をいくつかの値に設定した場合ごとに示している。Case I ~ Case IV のそれぞれについて、プライシングの初期値を変えながら、それぞれのクラスが 7 個程度含まれるような非線形時系列を生成し、シミュレーションの結果を求めている。シミュレーションでは、単独プライシングにおけるパラメータ設定 P を用いる。なお、シミュレーションにおいては、最初にプライシングを生成するモデルのパラメータの値を既知として与えて推定を実施するので、その推定結果がどれくらい正しいかは、ただちに計算することができる。表 1 においては、相対誤差である  $R_I = |\pi(0) - \hat{\pi}(0)|/\pi(0), R_\mu = |\mu - \hat{\mu}|/\mu, R_J = |P_J - \hat{P}_J|/P_J$  の平均値を示している。この結果から分かるように、パラメータの推定誤差は小さく抑えられており、GA 手法による推定は良好であると言える。

#### 4.4 セグメントごとの最適なクラスの推定

次に、本報告の手法を用いて非線形時系列生成モデルをセグメントごとに推定した結果について整理する。シミュレーションにおいては、あらかじめ初期設定において、セグメントごとに割り当てている関数のクラスが分かっているので、本報告の手法により推定したセグメントごとのクラス推定値と比較することにより、性能を評価することができる。すでに述べたように、容量  $\mu$  をいくつかの値に設定した場合ごとに Case I ~ Case IV のそれぞれについて、プライシングの初期値を変えながら、それぞれのクラスが 7 個程度含まれるような非線形時系列を生成し、シミュレーションの結果を求めている。シミュレーションでは、単独プライシングにおけるパラメータ設定 P を用いる。

表 2 には、それぞれのケースについてクラスごとに、セグ

表 2 セグメントに割り当てたクラスの推定結果 (パラメータ設定 P を利用).

Table 2 Result of estimation of classes for segments (by using parameter setting P).

| Case I   |       |       |       | Case II |       |       |       |
|----------|-------|-------|-------|---------|-------|-------|-------|
| $\mu$    | $z_1$ | $z_2$ | $z_3$ | $\mu$   | $z_1$ | $z_2$ | $z_3$ |
| 0.3      | 0.94  | 0.80  | 0.92  | 0.3     | 0.90  | 0.92  | 0.92  |
| 0.5      | 0.95  | 0.81  | 0.93  | 0.5     | 0.90  | 0.92  | 0.92  |
| 0.7      | 0.95  | 0.82  | 0.94  | 0.7     | 0.92  | 0.92  | 0.93  |
| Case III |       |       |       | Case VI |       |       |       |
| $\mu$    | $z_1$ | $z_2$ | $z_3$ | $\mu$   | $z_1$ | $z_2$ | $z_3$ |
| 0.4      | 0.90  | 0.91  | 0.91  | 0.4     | 0.89  | 0.91  | 0.80  |
| 0.55     | 0.90  | 0.91  | 0.92  | 0.55    | 0.90  | 0.92  | 0.81  |
| 0.7      | 0.89  | 0.91  | 0.92  | 0.7     | 0.90  | 0.93  | 0.81  |

メントが正しく検出された割合を示している。セグメントの真のクラスは既知であるので、時刻  $t$  ごとに推定されたセグメントのクラスと、既知のクラスとの比較を行うことで、正しい検出の割合を求めている。この表から分かるように、セグメントの識別は Case I ~ Case IV ごとに多少の変化はみられるが、ほぼ 0.8 ~ 0.95 の範囲に収まっており、本報告の手法の妥当性を示すものとなっている。

#### 4.5 現実データの時系列生成モデル変化の実時間推定

現実に観測される時系列データとして、次の 2 種類を用いる。

Case A: 2 つの製品の実際の需要データと発注データ

ここで用いるのは文献に示されている、実際に観測された製品の需要および発注データであるが、ここでは少し加工を行っている。元の文献では、需要予測をする小売業者の推定する理想的な需要時系列  $y_1(t)$  と、これを小売業者が独自に判断で修正して製造業者に伝達する時系列  $y_2(t)$  の 2 つが示されている。これらの時系列においては、 $y_1(t)$  は比較的緩やかな傾向を示すのに対して、時系列  $y_2(t)$  は、小売業者のより有利に商品を調達したい小売業者の行動が反映され、変動の大きなものになっている。本報告では、これらの 2 種類の時系列を  $y_1(t), y_2(t)$  の順番で接続して、1 つの時系列とみなして、これらの接続時点の推定と、非線形モデルの関数形状の推定がなされるかを同時に検証する。

Case B: 対個人・対事業所サービス業界データ

日本における 2000 年から 2013 年までの対個人・対事業所サービス業界のデータであり、対個人サービスではテーマパークなど 8 つの業種のデータ (Case B, Data I としておく) を、対事業所サービスではリース業など 15 の業種における来客数 (会社にサービスを依頼する件数) の月次データ (Case B, Data II としておく) として集計されている [12]。

次に、GP による非線形システムの推定について述べる。国の経済のサイクルは、1 年を通じて、極めて明確なサイクルが存在することが知られており、例えばデパートの売り上げは、毎年 12 月に大きなピークがある。これを季節性と

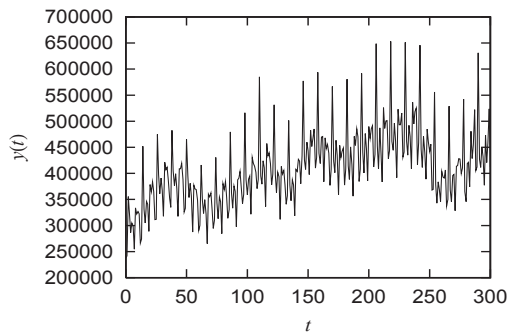


図 3 Case B における時系列の例

Fig. 3 An example of time series in Case B

呼んでいる ( $S(t)$  としておく), また, 長年にわたって時系列のベースとなる数値が緩やかに変化するトレンドと呼ばれるものが存在する ( $T(t)$  としておく). これらの 2 つを, 観測時系列から除去したあとには, 不規則な変動を示す時系列が残る (不規則成分と呼ばれ,  $R(x(t))$  としておく). このような成分への分解を行うことにより, GP による関数形状の推定も, やや単純な問題に帰着させることができる.

本報告では, 季節性への分解において, 成分の積で時系列を表現する乗法モデルを用いる. これまで述べた非線形時系列の生成モデルを, この乗法モデルで表現すると, 以下のようになる.

$$F(x)^{(z_k)} = R(x), G(x)^{(z_k)} = S(t)T(t)F(x)^{(z_k)}. \quad (16)$$

季節性  $S(t)$  は明確な周期性をもっているので, 周期関数により近似する.  $R(x(t))$  は不規則な動きをするので, 通常行われる近似方法にしたがって, 1 次の自己回帰 (Auto-regressive:AR) モデルを適用して近似する. トレンド  $T(t)$  は単調な線形, あるいは低次の多項式の関数で近似する.

時系列の長さは人工データと比較して相対的に短いので, 識別されるセグメント個数とクラス数は同じになる. 検出されるセグメントの個数  $N_S$  は, 時系列の長さに関連しているが, それほど長い時系列ではないので,  $N_S = 2, N_S = 3$  に分布している. 推定された関数形状は,  $S(t)$  については周期関数,  $T(t)$  については緩やかに変化する単調な関数となっている. また  $R(x(t))$  は, 通常経済時系列を近似するモデルとして用いられている 1 次の自己回帰モデルが, 適切であることが示されている. スペースの関係で, 関数形状は省略する.

図 3 には, Case B における時系列の例を示している.

## 5. むすび

本報告では, PF とプライシングによる時系列区分化の手法を用いた非線形時系列生成モデル変化の推定について述べた. 時系列を区分化する場合に, プライシングを用いて効率を向上させた. 現実データに対して, 変化の発生と変化後の関数を推定する方法を提案し, 応用例として人工的

なデータを用いた分析と, 現実データからの状態方程式推定を議論した.

今後, より複雑な関数形のシステムへの適用可能性を検討する予定である.

## 参考文献

- [1] N.Dobigeon, J.Y.Tourneret and J.D.Scargle, "Joint segmentation of multivariate astronomical time series:Bayesian sampling with a hierarchical model," IEEE Transactions on Signal Processing, vol.55, no.2, pp.414-423, 2007.
- [2] E.Fox, E.B.Sudderth, M.I.Jordan and A.S.Willsky, "Bayesian nonparametric inference of switching dynamic linear models," IEEE Transactions on Signal Processing, vol.59, no.4, pp.1569-1585, 2011.
- [3] N.Vaswani, "Additive change detection in nonlinear systems with unknown change parameters," IEEE Transactions on Signal Processing, vol.55, no.3, pp.859-872, 2007.
- [4] V. V. Kadiramanathan, P. Li, M. H. Jaward and S. G. Farbi, "Particle filtering based fault detection in nonlinear stochastic systems," International Journal of System Science, vol.33, pp.259-265, 2002.
- [5] A.Doucet, N.J.Gordon and V.Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," IEEE Transactions on Signal Processing, vol.49, no.3, pp.613-624, 2001.
- [6] 時永祥三, 譚 康融, "遺伝的プログラミングによる方程式近似に基づく粒子フィルタを用いた時系列からの状態推定とその変動抑制への応用," 信学論, vol.J93-A, no.11, pp.739-755, 2010.
- [7] 時永祥三, "3つのネットワーク結合離散系カオス変動モデルにおける edge snapping による同期化とその応用," 情報処理学会論文誌, 数理モデルとその応用, vol.6, no.2, pp.36-52, 2013.
- [8] C.M.Rump and S. Stidham, Jr., "Stability and chaos in input pricing for a service facility with adaptive customer response to congestion," Management Science, vol.44, no.2, pp.246-261, 1998.
- [9] 池田欽一, 陳曉榮, 時永祥三, "GP による学習を基礎としたマルチエージェント・システムによるプライシング時系列のカオス性分析とその応用," 信学論, vol.J89-A, no.4, pp.298-307, 2006.
- [10] 池田欽一, 時永祥三, "ネットワーク構成されたノードにおけるフロー入力調整と退去を含むプライシング時系列のカオス解析とその抑制," 信学論, vol.J93-A, no.1, pp.1-10, 2010.
- [11] 池田欽一, 時永祥三, "ノードへのフロー入力調整を含むネットワークにおけるプライシングのカオス性変動の解析とその抑制," 情報処理学会論文誌, TOM0202004, vol.2, no.2, pp.22-37, 2009.
- [12] 時永祥三, 池田欽一, "局所的交流による行動決定と状態遷移を行うマルチエージェントからなる平面上の エージェント・クラス形成分析," 情報処理学会論文誌, TOM, vol.4, no.4, pp.19-36, 2011.
- [13] H.L.Lee, V.Padmanabhan and S.Whang, "Information distortion in a supply chain:The bullwhip effect," Management Science, vol.43, no.4, pp.546-558, 1997.
- [14] 総務省統計局 データ, [http://www.meti.go.jp/statistics/tyo/tokusabido/result/result\\_1.html](http://www.meti.go.jp/statistics/tyo/tokusabido/result/result_1.html)