九州大学学術情報リポジトリ
Kyushu University Institutional Repository

粒子フィルタおよび遺伝的プログラミングを用いた非線形時系列モデルにおける構造変化の推定

時永，祥三
九州大学大学院経済学研究院

池田，欽一
北九州市立大学経済学部

https://doi.org/10.15017/21836

出版情報：經濟學研究. 78 (5/6), pp.1-15, 2012-03-30. 九州大学経済学会
バージョン：
権利関係：

KYUSHU UNIVERSITY

粒子フィルタおよび遺伝的プログラミングを用いた非線形時系列モデルにおける構造変化の推定

時　永　祥　三
池　田　欽　一

1　まえがき

非線形システムにおいて観測データから内部状態を推定する手法として, 粒子フィルタ (Particle Filter： PF) が適用され, 移動体の同定などに応用されている [1, 2, 3, 4, 5]. また, 状態方程式が未知である場合に, 関数推定の手法と PF とを結合する方法も提案されている [6, 7]. 一方, 従来より観測された時系列データを用いて, 予測誤差の統計処理によりシステムの状態方程式の変化 (以下では構造変化と呼ぶことにする) を抽出する方法が適用され, 最近では非線形システムに拡張されている [8, 9, 10, 11, 12, 13, 14]. しかしながら, 構造変化の後でシステムを記述する状態方程式が, どのように変化しているかを推定することは重要であると思われるが, あまり検討はなされていない. これまで粒子フィルタを用いて観測データから状態推定を行う方法において, 状態変数にマルコフ遷移の状態を加えて, この中のどの状態にあるかを推定することが行われているが, 線形システムに限定されており, またあらかじめ状態遷移する複数の状態方程式が既知である制限がある [1, 2, 3]. 本論文では,PF および遺伝的プログラミング (Genetic Programming:GP) を用いた非線形時系列モデルにおける構造変化の推定とその応用について述べる [15].

本論文では, 構造変化の時点から, どのような方向にシステム関数が変化するかを推定することを目的としている. すなわち PF における非線形方程式が変形され, 現在の関数 (C とする) から派生すると仮定し,GP 手法における交叉処理によりこの変形後の複数の関数 ($A_1, A_2, ...$ とする) を推定する方法を適用する [7, 16, 17, 18, 19, 20, 21, 22, 23, 24]. 関数 C を元のシステム関数とすると,GP 手法における個体プールにはランダムに生成された関数の集合があり, これらの関数と元の関数 C を部分的に交換する交叉処理をほどこすことにより, 関数 C から変形された複数の関数形を個体プールの個体の個数だけ求めることができる. この生成された関数 (A_i とする) のすべてに対して PF を適用することにより, 構造変化後の関数として誤差が最小となりモデルの尤度が最大となる関数 A_m を見出すことができる. 応用例として人工的なデータを用いて, 本論文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, 構造変化後の状態方程式推定を議論する.

以下では, 2. では PF による構造変化抽出と関数推定の基本を説明する. 3. では GP 手法と関数変形による方程式推定について述べる, 4. では応用例を示す.

2　粒子フィルタによる構造変化抽出と関数推定

2.1　GP と粒子フィルタを用いた構造変化の推定

以下ではまず最初に, 観測された時系列から GP 手法と粒子フィルタ (PF) により構造変化 (状態方程式の変化) を抽出する方法について概要を述べておく. 状態方程式で記述される系の内部状態推定の際, 一般的にはこの状態方程式の形状は時間変化しないと仮定されている. しかし現実の多くの事例で, 状態方程式の構造変化が見られ, このような構造変化を観察する指標として, モデルの尤度が用いられている. すなわち, 構造変化が発生した場合には, 構造変化を仮定しないモデルを用いて状態推定などを行った場合に, モデルの尤度が極端に低下することが予想される. この尤度の時間変化や, これに類似した指標は構造変化を検出する手法として有用である. しかしながら一方では, 構造変化が発生した場合において, 発生を検出することと同時に, どのような構造変化が発生したかを推定することも重要である. これまでの研究において, 構造変化を仮定しないで信号の異常などを検出する方法論は検討されているが, 構造変化の後の状態方程式を推定することは行われていない [9, 10, 11, 12, 13, 14]. 本論文では, この構造変化の後の状態方程式を GP 手法による推定することを提案する [15]. また一般的な非線形システムを取り扱うために, PF による状態推定を用いることとする.

このような方法論をまとめると, 以下のようになる.

(1) 尤度の時間変化の観測

PF 手法の適用において求められる尤度を計算し, その時間変化を観測する. この尤度の値が一定の基準より低下すると, 構造変化が発生したと判断する

(2) 構造変化の後の状態方程式推定

構造変化が発生したと判断した時点以降のデータを用いて構造変化を近似する方向にシステム方程式を推定する. 関数近似手法として確率的関数、あるいはプログラム探索手法の遺伝的プログラミング (GP) を適用するが, 初期の解（GP において関数を表す個体）をまったくランダムに作成せずに, 構造変化をする前の状態方程式の形状も, 参考にすることとする. これにより構造変化前後の関数系に類似性がある場合, 解（構造変化後の状態方程式）を効率的に探索することができることとなる.

(3) 構造変化の後の状態推定の適用

構造変化を検出することだけが目的とされる場合のほかに, 構造変化の前後においても, 継続して観測できない内部状態推定をする必要があるケースも存在する. これを考慮して, 本論文では構造変化の前後における状態推定の性質についても議論する.

2.2　粒子フィルタによる状態推定の基本

以下では粒子フィルタによる状態推定の基本についてまとめておくが, 本論文で適用する分野に限定して簡潔に述べることとする. 非線形状態方程式により生成される時系列モデルを, 次のように記述しておく [1, 2, 3, 4, 5, 6, 7].

$$x_t = F_t(x_{t-1}) + v_t. \tag{1}$$

$$y_t = H_t(x_t) + w_t. \tag{2}$$

ここで, 変数 y_t は時刻 t ($t = 1, 2, \ldots, T$) 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 $x_t = (x_{1,t}, x_{2,t}, \ldots, x_{n,t})$ は推定されるべき n 次元の状態変数ベクトルである. また, ベクトル $v_t = (v_{1,t}, v_{2,t}, \ldots, v_{n,t})$ は n 次元のシステムノイズ, w_t は 1 次元の観測ノイズである.$F_t(.)$ は x_{t-1} を引数とする関数ベクトルである. ここで,$F_t(.), H_t(.)$ はそれぞれ状態方程式, 観測方程式となる非線形関数であり, 一般の粒子フィルタの適用では, 両方程式は既知であると仮定するが, 本論文では構造変化前後の方程式が未知である場合があることを仮定し GP により推定することとする. 説明の都合上, これらの状態方程式の形状は, 以下では既知, あるいは GP により既に推定されていると仮定しておく.

いま,$F_t(.), H_t(.)$ が行列を用いた線形表現の形式で与えられ ($x_{t+1} = A_t x_t + v_t, y = B_t x_t + w_t$ の形式, 行列 A_t, B_t は時間 t のみの関数), v_t, w_t が正規乱数に従う場合には, 従来からある手法であるカルマンフィルタを適用して, 状態を推定することができる. しかし, 方程式が非線形の場合 ($F_t(.), H_t(.)$ が x_t を含む非線形関数である場合) には, カルマンフィルタが適用できないため粒子フィルタが援用され, 有効性が示されている. 粒子フィルタ適用の概要は, 以下のようにまとめられる. いま, 観測データとして時系列 y_t が与えられた場合に, 式 (1)(2) に示すモデルを仮定して, $x_t = (x_{1,t}, x_{2,t}, \ldots, x_{n,t})$ を推定する. 粒子フィルタでは状態変数の分布を多数の粒子により近似するが, 記号として, 状態 $x_{k,t}, k = 1, 2, \ldots, n$ の i 番目の粒子を $x_{k,t}^{(i)}$ と表すこととする.

ステップ 1:粒子の初期値の生成

時刻 0 における変数 $x_{k,t}^{(i)}$ の初期値 $x_{k,0}^{(i)}, i = 1, 2, ..., N$ について, あらかじめ定めた確率分布に従がう乱数により N セットの粒子の集合として生成する.

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, システムノイズである $v_{k,t}^{(i)}, k = 1, .2, ..., n$ および $w_t^{(i)}$ を, あらかじめ与えた確率分布に従った乱数によって N セットの粒子として生成する.

ステップ 3:$x_{k,t}^{(i)}$ の 1 時刻先予測

状態方程式により $x_{k,t}^{(i)}$ の 1 時刻先予測が次によりられる.

$$x_{t+1}^{(i)} = F_t(x_t^{(i)}, v_t^{(i)}), \quad v_t^{(i)} = (v_{1,t}^{(i)}, v_{2,t}^{(i)}, \ldots, v_{n,t}^{(i)}). \tag{3}$$

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

ステップ 4:重みの計算

次に, それぞれ粒子 i の重み $w_t^{(i)}$ を計算する.

$$w_t^{(i)} = R_t(y_{t+1}|x_{t+1}^{(i)}). \tag{4}$$

ここで $R_t(.)$ は $x_{t+1}^{(i)}$ が与えられた場合の y_{t+1} の条件付分布であり, これらの重みを次のように集計しておく.

$$W_t = \sum_{i=1}^{N} w_t^{(i)}. \tag{5}$$

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. すなわち粒子の集合 $X_{t+1} = (x_{t+1}^{(1)}, x_{t+1}^{(2)}, ..., x_{t+1}^{(N)})$ から粒子を $w_t^{(i)}/W_t$ の確率に従って復元抽出して, 時刻 $t+1$ での状態の分布を表す粒子を生成す

ることができる. この結果として, 変数 x_t の確率分布を推定することができる. 時刻 t における状態の点推定値は, 集合 X_t の平均として与えられる.

ステップ 6:時刻 $t = 2, 3, \ldots, T$ について繰り返し

ステップ 2〜ステップ 5 の操作を, 時刻 $t = 2, 3, \ldots, T$ について繰り返す. この結果として, 時刻 $t = 2, 3, \ldots, T$ についても状態変数 x_t の確率分布を推定することができる. 最終的に得られるモデルの対数尤度は, 次により与えられる.

$$L_m = \sum_{t=1}^{T} l_t - T \log N, \quad l_t = \log W_t. \tag{6}$$

構造変化抽出の指標については, L_m を構成する時間要素である $l_t = \log W_t$ を直接用いることにする. すなわち, 時間軸にそって計算される l_t の値が, ある閾値より小さくなると, 構造変化が発生したと判断する.

3　GP 手法と関数変形による方程式推定

3.1　GP 手法による関数変形

これまでの議論では, 状態方程式に含まれる関数である $F_t(.), H_t(.)$ は既知であるとして進めてきた. しかし, 状態方程式に構造変化が生じた場合には, この関数 $F_t(.), H_t(.)$ の形状は不明であり近似しながら, 状態推定を進めていくことが必要となる. この推定に GP を用いることとする.

GP はこれまで関数近似などの分野に応用され, 有効性が確認されている.GP 手法ついてはこれまで多くの記述があるので, ここでは概要を述べるにとどめておく [7, 16, 17, 18, 19, 20, 21, 22, 23, 24].

GP 手法を関数近似に適用する問題においては, 近似するための非線形関数の候補を多数の GP 個体として準備しておく. 非線形関数は四則演算など演算子を節（ノード）, 用いる変数, 定数など非演算子を葉（ターミナル）とした木構造により表される. これらの関数の近似度に対応する適合度が相対的に大きな個体どうしに対して部分的に木を交換する交叉処理を適用し, 新たに生成された個体を, 個体プールの中で適合度が相対的に小さい個体と置き換える. このような操作を繰り返すことにより, 個体の関数近似の精度を向上させることができる. また個体集合の単純な収束を防ぐためにランダムに個体を書きかえる突然変異処理も実施する.

ステップ 1：初期個体発生

乱数を用いて被演算子, 演算記号からなる木構造を発生させ, 初期個体のプール P-S を構成する.

ステップ 2：適合度計算

個体に表現された関数をもとに, それぞれの個体により得られる予測値を求める. これをもとにして, 個体の適合度（評価値）S_i を求める.

ステップ 3：交叉処理

適合度に比例する確率に応じて, 2 つの個体を選択し, あらかじめ設定した交叉確率に応じて, この 2 つの個体に対して関数を表す木を部分的に交換する遺伝的操作を行い, 生成された新しい個体を個体のプールである P-B に格納しておく. このような新しい個体の生成を, 規定回数繰り返す. 新規個体の生成が終了したら, プール P-A(最初の繰り返しではこれはプール P-S に等しい) の個体の

中で, 相対的に適合度の低い個体を, プール P-B の個体により置き換える. この結果, プール P-A は, もとから存在する適合度の高い個体と, 生成された適合度が高い個体から構成されることになる.

ステップ 4：突然変異処理

ステップ 3 の交叉処理を適用し, 新しい個体のプールを作成したあとに, 設定した突然変異確率に応じて突然変異を実施する. 突然変異は任意に個体を選択して, この個体の被演算子, 演算記号の部分を, ランダムに選択した被演算子, 演算記号により置き換える.

ステップ 5：繰返し

ステップ 2 からステップ 4 までの操作を規定回数, あるいは適合度に変化がなくなるまで繰り返す.

一般的な GP 手法は上記のような方法となるが, 本論文で議論するような構造変化をした後のシステムの状態方程式を推定する場合には, やや異なる手順で GP 手法における交叉処理を利用し, 関数推定の効率化を実施する. すなわち, 現在までシステムの挙動を記述している関数 C が, ある時刻において別の関数に変化 (構造変化) することを仮定するが, この関数の形状は現在の関数 C から派生し, 類似する側面をもつと仮定し,GP 手法により, この変形後の関数形を推定する方法を適用する. すなわち GP 手法における初期個体発生時に個体プールはランダムに生成された関数の集合であるが, 最初の交叉処理の実施の際, このランダムに生成された関数と関数 C を交叉のペアとし, 関数 C の性質を引き継いだ個体（関数) を生成することとする. これにより, 関数 C から変形された複数の関数形 A_i を個体プールの個体の個数だけ求めることができる. このように生成された関数 (個体) のすべてに対して粒子フィルタを適用することにより, 予測誤差が最小となる関数 A_m を見出すことができる.

3.2 粒子フィルタと GP との並行的最適化

本論文では通常の粒子フィルタとは異なり, 状態方程式の変化を GP により近似する方法を適用している. このため, 粒子フィルタにおける対数尤度 L_m (個体 i に対して得られる L_m を $L_{m,i}$ とする) と, GP における適合度 (個体 i を用いた場合の関数の推定誤差の絶対値の時刻 t における平均を e_i とし, この逆数を個体の適合度 $f_i = 1/e_i$ として定義する) の両方を用いて, 粒子フィルタおよび GP において並行的に最適化を進める必要がある. 本来の対数尤度には時系列長 T と粒子数 N だけに関係する項 $T \log N$ があるが, これらはシミュレーションを通じて固定するので, これを除外しておく. これを行うために一般的には 2 つの最適化尺度 $(f_i, L_{m,i})$ に重みをつけ, GP 適用における本来の適合度の代わりに用い, 最適化の進行につれて, これらの比率を変化させる方法を用いている [7]. すなわち, 個体そのものの適合度 f_i ではなく, 次に示す拡張された適合度 G_i を用いる.

$$G_i = L_{m,i} + I_G \beta f_i. \tag{7}$$

このように粒子フィルタを適用したあとに, 拡張された適合度を用いて遺伝的操作により GP 個体の適合度を改善することを 1 回の操作とする. この式においては, このような毎回の操作を繰り返す回数 (世代) を I_G とする (β の大きさは問題に応じて変える). この尺度は GP, 粒子フィルタ適用の回数 I_G に応じて,f_i の重みがより大きくなるように設定しておく. すなわち粒子フィルタ, GP 適用の最初の段階では GP 手法の適用による関数の近似の精度よりは, 粒子フィルタにおける粒子選択の方が重要であると思われるので, GP 適用においても $L_{m,i}$ の重みを相対的に大きくしておく.

しかし次第に粒子フィルタおよび GP によるシステムの推定が進行するにしたがって, GP 手法による関数近似が重要となると思われるので, f_i の重みを相対的に大きくしておく. 尺度 G_i は GP における個体選択に用いられる相対的な値であるので, このような重みの変更による絶対値の変化は, 最適化に影響しない.

4 応用例

4.1 人工データを用いた構造変化の抽出

本論文で提案する状態方程式変化の抽出について, 最初に簡単な人工データを用いて性能を考察する [7, 25, 26]. この事例は製品の製造販売において小売業者 (エージェント) が製造業者に注文を出す場合に, 本来行うべき注文を自己の都合で変形するケースである. この例題では構造変化後の観測方程式が未知であるケースとなる. 以下の例題に用いる記号の定義を示しておく.

- D_t : 時刻 t におけるエージェントが見込む需要.
- J_t : 時刻 t におけるエージェントが見込む理想的な在庫数量.
- B_t : 時刻 $t, t-1$ におけるエージェントが見込む理想的な在庫数量の差.
- S_t : 時刻 t におけるエージェントの過剰在庫数量.
- A_t : 理論的に最適な時刻 t におけるエージェントの注文数量
- O_t : 時刻 t におけるエージェントが実際に注文する数量.
- $h_t(.)$: エージェントの行動を記述する非線形関数.

基本モデルを以下に示す [26].

$$D_{t+1} = d_i + \rho D_t + \varepsilon_t^D. \tag{8}$$

$$B_t = J_t - J_{t-1}, \quad J_t = m_t + k\sigma^D\sqrt{n}, \quad S_{t+1} = O_t - A_t. \tag{9}$$

$$O_t = h_t(A_t), \quad A_t = D_t + B_t - S_t. \tag{10}$$

ここで A_t, B_t, J_t, S_t は, 単なる定義式 (中間変数) である. また m_t, n は以下のように与えられる.

$$\begin{aligned} m_t &= \frac{d}{1-\rho}[(l+1) - \sum_{j=1}^{l+1}\rho^j] + \frac{\rho(1-\rho^{l+1})}{1-\rho}D_t, \\ n &= \frac{1}{(1-\rho)^2}\sum_{j=1}^{l+1}(1-\rho^j)^2, \\ k &= \Phi^{-1}[p_2/(p_1+p_2)]. \end{aligned} \tag{11}$$

p_1, p_2 は商品の貯蔵コスト, 在庫切れコストであり関数 $\Phi(.)$ は標準正規分布の分布関数である. 本論文では $p_1 = 2, p_2 = 50$ としておく.

観測データは次の式により与えられる.

$$y_t = O_t + \varepsilon_t^O. \tag{12}$$

ただし, D_t, O_t にはそれぞれ, 正規分布する平均がゼロで標準偏差が σ^D, σ^O であるノイズ $\varepsilon_t^D, \varepsilon_t^O$ が含まれることを仮定している. 関数 $h_t(.)$ が A_t について線形なら通常のカルマンフィルタによる推定が可能であるが, エージェントは次の非線形式で表される 3 つの行動をとり, 理想的な注文数量 A_t を修正して注文を出すと仮定する. なおここに示す例題では, ある時刻にエージェントが用いる関数 $h_t(.)$ は 1 種類であると仮定する. 以下では文献 [25] に整理されている代表的な小売業者 (エージェント) の行動を, 関数で表現することをこころみる. すなわち小売業者は, 1) 生産者が価格をつりあげているのではないかと疑い買い控えをする行動 (price variations) 2) 注文数量が一定のレベルになるまで注文しないでまとめて注文する行動 (order batching) , および 3) 売れそうな商品を早めに確保する行動 (rationing game), をとることがあるがあると述べられている. これらの 3 つの行動を元に, ケースとして以下に仮定する.

- **Case 1**: 一定量を超えると 1 次以上の関数にしたがって注文を増やす
- **Case 2**: 一定数量になるまで注文をしないで, あとでまとめて注文する
- **Case 3**: 一定量を超えるまでは 1 次以上の関数にしたがって増やすが, その後は緩やかに増加する

これらのケースでは理想的な注文数量を x とした場合に, それぞれ Case 1:大きい x の部分が高めに評価される, Case 2:x についてまとめて注文を行う, Case 3:小さい x の部分が高めに評価されるに対応する. ただし, 前に整理した本来の 1)〜3) の小売業者の行動を完全に表現したものではないので, 擬似的に関数で表現したものである限界はある. 人工的なデータを用いたシミュレーションの条件を以下に示す. なおエージェントは, それぞれの Case 1〜Case 3 において, 次のような関数形状 $h_t(.)$ を用いて決定を行うと仮定しておく.
Case 1: $h_t = (A_t/1.2)^{1.2}$
Case 2: $h_t = r(A_t - 3.5)$
Case 3: $h_t = 3\log(1 + 3A_t)$

なお Case 0 として, 理想的な注文数量 A_t を修正しないで注文するケースを定義しておく.

構造変化の時刻については, 時刻 $t = 1 \sim T_1$ までを変化の前段として, 時刻 $t = T_1 + 1 \sim T_2$ を構造変化の後の時刻とする. これまで説明したモデルに対して人工的なデータを与えて, GP および粒子フィルタによる構造変化推定と変化後の観測方程式推定のシミュレーションを行った結果を示す. 本節ではパラメータは, 次のように設定している.
$d = 2, \rho = 0.5$, $l = 5, \sigma^D = 0.5, \sigma^O = 0.4$, T_1, T_2 の設定:$T_1 = 25, T_2 = 35$, 粒子フィルタにおける粒子数:10000, GP 個体数:1000, GP 個体における最大配列長:30, GP における交叉処理および突然変異の確率:0.2 および 0.01.

図 1 には GP 手法適用により方程式の変化を推定する効果を見るために, 構造変化の前後における尤度変化の例を示している. 図においては構造変化を GP 手法により推定して状態方程式を変化させた場合 (with-updata) に得られる尤度と同時に, このような方程式の変化を行わない場

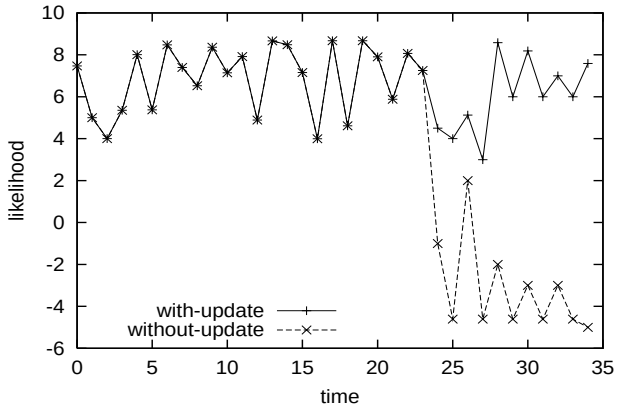

図 1: 人工データにおける構造変化の前後での尤度の時間変化の例

合 (without-updata) の尤度を表示している. なお,GP による推定を用いるケース (with-updata) では構造変化検出後, システム方程式を推定し, 尤度を再計算している. この図から分かるように with-updata の場合には構造変化の後 ($T = 26$ 以降) では尤度は構造変化の前 ($T = 25$ 以前) と比較してやや低下するが, その値は without-updata の場合と比較して十分に大きなものとなっている.

このように GP 手法により構造変化のあとの関数を推定し, これを用いる有効性を調べるために, 構造変化の前および構造変化後の関数の形状 (それぞれ C と A とする) を Case 0〜Case 3 から選択し, 関数 A を GP により推定した際に得られるモデルの尤度 l_t の平均値 (L1) を求めておく.L1 は関数推定が適切であれば大きくなる. これに対して関数を変化後の関数をそのまま C し, 変更しない場合のモデルの尤度 l_t の平均値 (L0) を求めておく. これらの 2 つの尤度の差は構造変化により低下する尤度の平均値を表し, 変化後の方程式を推定したことによる有効性であると考えることができる. これをまとめたものが表 1 である. 表 1 においては, 縦軸 (横軸) が構造変化の前 (後) における関数の形状に相当するケース (Case 0〜Case 3) を表しており, 欄の中には構造変化の検出と GP による関数形状の推定を行ったあとでのモデル推定の尤度の差異 (L1-L0)) を示している. この結果から分かるように, GP 手法および粒子フィルタを適用することによる構造変化の検出と, 観測方程式の変化の推定が正しく行われている.

次に, 表 2 には GP 手法により推定された構造変化後の関数の形状 $h_t(.)$ を示している. 表 2 の結果においては, もとの関数の形状に加えて変数のべき乗の関数などが加わった形になっているが, 本来含まれているべき関数部分は推定された形状には加わっていることが確認できる. これらの結果より分かるように, GP により推定された構造変化後の方程式と, これに基づく粒子フィルタによる状態推定は適切なレベルには十分であると思われる.

図 2 には構造変化の前後における状態 D_t の推定の例を示す. これらのモデルにおいてはモデルとして生成された状態の値は未知として粒子フィルタで推定するがモデルから実際の値を得ることもできるので, 図においては推定された状態 $\hat{x}_t$(構造変化後関数推定あり:with-update, 構造変化後関数推定なし:without-update) と, モデルにより計算される D_t(original) とを同時に示し比較した例を示している. これより構造変化後の関数推定なし (without-update) では推定された状態が構造

表 1: GP による構造変化推定を考慮した場合のモデル尤度の向上

pre/post	Case 0	Case 1	Case 2	Case 3
Case 0	-	4.2	12.3	7.2
Case 1	4.0	-	10.2	5.0
Case 2	10.0	8.0	-	12.1
Case 3	4.1	3.1	8.0	-

表 2: GP 手法による関数 $f_t(.)$ の推定の例

case change	$\hat{h}_t(.)$
Case 0→Case 2	$u(x-4.0)*4.5+0.2x^2+0.01x$
Case 0→Case 1	$0.12x+0.01x^2+0.001x^3+0.001\log(1+2x^2)$
Case 0→Case 3	$0.9x+0.01x^2+1.2\log(1+3x^2)$
Case 2→Case 0	$0.1u(x-4.5)+2.0x+0.01x^2+0.001x^3$
Case 2→Case 1	$0.2u(x-4.5)+0.9x+0.001x^3$

変化後 ($t=26$ 以降) 本来の状態の値と外れているが, これと比較し関数推定あり (with-update) では元の状態と近い値が推定されていることが分かる.

また, 構造変化を GP 手法で推定する方法の有効性を確認するために, 構造変化の前後において計算された状態の推定 $\hat{x}_t$ と実際のデータである x_t との間の平方 2 乗誤差を, ノイズの標準偏差 σ^D に係数をかけた形で求めてみる. 以下ではこのような比較分析の結果だけを簡単に整理しておく. まず, 構造変化の前における状態変数の推定誤差の平均は $0.3\sigma^D$ 程度である. 一方, 構造変化のあとで GP 手法で推定された状態方程式をもとに推定した状態変数の推定誤差の平均は $0.42\sigma^D$ 程度である. なお推定誤差の大きさを比較する例として, 情報関連機器の需要分析として影響する複数の状態変数 (既知として与えられている) を含む線形モデルにおいてカルマンフィルタを用いた状態推定があるので, これを参考にする [27]. これらの例題では, 既知である状態を観測データから推定した場合, その推定誤差は $0.01\sigma^D$〜$0.45\sigma^D$ の範囲にあり, 平均は $0.25\sigma_i^D$ 程度であることが示されている. したがって本論文の手法による粒子フィルタによる状態推定は, カルマンフィルタにおける推定より精度は低下しているが, 構造変化のあとにおいても極端な推定の悪化は見られないといえる.

4.2 経済モデルにおける状態方程式変化の抽出

次の応用例として, 米国と日本の経済構造の変化を抽出する問題へ本論文で示す手法を適用して考察する [28]. 具体的には, 経済活動を代表する指標を時系列として収集すると同時に, 一般的によく知られている経済構造の変化を, これらの時系列を用いた状態方程式の構造変化として検出できるかを考察する. 経済構造の変化として具体的には, 米国では 1979 年におけるインフレの進行

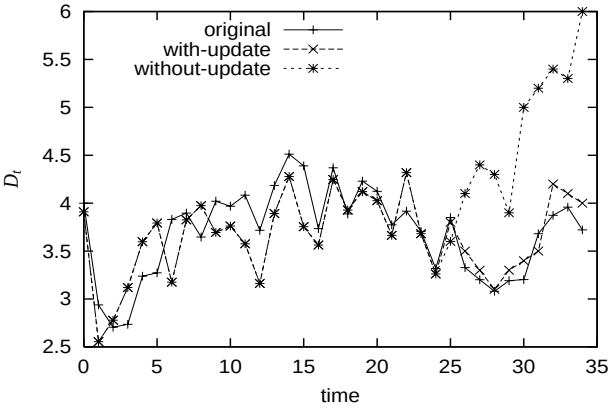

図 2: 状態 D_t の推定の例

(Model U) であり, 日本では 2001 年の IT バブル景気の崩壊 (Model J) と仮定する. これらモデルは構造変化前後の状態方程式が未知で観測方程式が既知であるケースである.

用いるデータは米国と日本の経済に関するデータであり, 年 4 回観測される (いわゆる 4 半期データ). 観測期間は構造変化の前後に分けられ, 以下のように区別する. 米国:前半は 1957 年〜1979 年, 後半は 1980 年〜2005 年. 日本:前半は 1980 年〜2000 年, 後半は 2001 年〜2006 年.

観測時系列を y:実質利子率とし, 状態ベクトルを x_t のそれぞれの要素 $x_{1,t}, x_{2,t}, \ldots$ は次のように定義している. $x_{1,t}$:国内総生産, $x_{2,t}$:労働時間, $x_{3,t}$:名目利子率, $x_{4,t}$:物価指数, $x_{5,t}$:個人消費.

なお, 状態方程式は前節で述べた人工的なデータを用いた応用例とは異なり既知ではないので, GP 手法により推定することになる. 話を簡単にするために, 式 (1) に示す変数に関する状態方程式について, 観測期間の前半のデータを用いて, 変数 $x_{i,t}$ に関する非線形関数 $F_t(x_{t-1}) = (f_{1,t}(x_{t-1}), f_{2,t}(x_{t-1}), \ldots)$ を, それぞれ GP 手法により推定しておく (いわゆる非線形の自己回帰モデルの推定). また観測方程式は既知で固定されたものと仮定し, 実質利子率 y_t を名目利子率 x_3 と物価指数の x_4 変数により説明するものとして, 以下のように簡単に定義する.

$$y_t = x_{3,t} - x_{4,t} + v_t. \tag{13}$$

GP パラメータなどシミュレーションの条件は, 前節で述べた人工データを用いた場合と同じである. なおこの例だいでは構造変化前の状態方程式も未知であり, 観測データを用いて予め状態方程式を推定する必要があるので, ここでは, 分析開始の起点を Model J, Model U についてそれぞれ $t_0 = 52, 64$ として, ここをあらためて $t = 1$ とし, 構造変化の前後 $t = 1 \sim T_1, t = T_1 + 1 = T_2$ を次のように設定している.T_1, T_2 の設定:$T_1 = 25, T_2 = 35$.

図 3 には構造変化の前後において, GP 手法により状態方程式の推定を行って粒子フィルタを適用した場合 (with-updata) と, 特にこのような構造変化への対応を行わなかった場合 (without-updata) の, 尤度の時間変化を Model J について示している. なお, 構造変化後の状態方程式を推定する場合においては尤度は推定された方程式を用いて再計算している. この図 3 より分かるように, 構造変化を検出して GP により状態方程式の推定と状態方程式の切り替えを行った場合には尤度の極端な

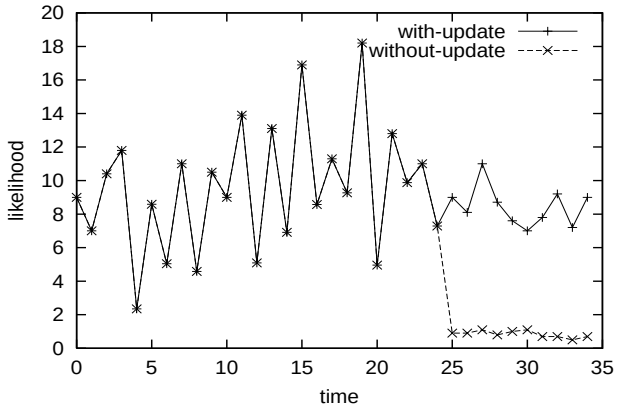

図 3: 経済モデルにおける構造変化前後での尤度の時間変化の例

低下は見られず, その効果は, 明らかである. 構造変化を考慮し関数を推定した場合に得られるモデルの尤度 l_t の平均値 (L1) と, 特に関数を推定しない (変化させない) 場合のモデルの尤度 l_t の平均値 (L0) との差 $L_D = L1 - L0$ は次のようになる.Model J:$L_D = 8.0$, Model U:$L_D = 8.4$.

次に, 表 3 には GP 手法により推定された関数の形状 $t = 1$〜T_1 および $t = T_1 + 1$〜T_2 における $f_{t,i}$ の形状の例 (Model J) を示している ($f_i(T_1)$ などとして区別, 時刻 t は省略している). 表 3 から分かるように, 構造変化の後において推定された関数の形状は, 変化の前の関数形状に別の変数項が付加された表現が多く見られる.

4.3 定数記述にバイアスが加わるケース

これまで非線形時系列生成モデルにおける構造変化を議論した研究では, 状態方程式を記述する式の定数において, ある時刻 T_1+1 以降にバイアスが加わる事例が用いられている [9, 13, 29, 30]. そこで以下では次のような 3 つのケースについて考察する. 構造変化の前後 $t = 1$〜$T_1, t = T_1 + 1 = T_2$ を次のように設定している.

T_1, T_2 の設定:$T_1 = 15, T_2 = 25$

最初の 2 つのケースの状態方程式は, 次に示される.

$$\begin{aligned} x_t &= ax_{t-1} + b + c \times 25 \frac{x_{t-1}}{(1 + x_{t-1}^2)} \\ &\quad +8cos[1.2(t-1)] + v_{t-1}. \end{aligned} \tag{14}$$

$$y = 0.05x_t^2 + w_t. \tag{15}$$

Case A:時刻 T_1 までは $a = 0.5, b = 0, c = 1$ であり, 時刻 $T_1 + 1$ 以降において $b = 2$ となる.

Case B:時刻 T_1 までは $a = 0.5, b = 0, c = 1$ であり, 時刻 $T_1 + 1$ 以降において $c = 0.33$ となる.

3 番目のケースの状態方程式は次のように仮定する.

$$x_{1,t} = 5 + d + 0.3x_{1,t-1} + v_{1,t},$$

表 3: GP 手法による関数 $f_t(.)$ の推定の例

$f_i(.)$	functions
$f_1(T_1)$	$\lvert 1.2+x_3+3x_1-2x_4+\lvert x_2+1.2\rvert+1.3(1.2-\lvert x_2+x_1\rvert)\rvert$
$f_1(T_2)$	$\lvert x_3(0.8+x_2)+1.7+\lvert 2x_4-0.6\rvert\rvert$
$f_2(T_1)$	$\lvert 2.4+x_3+2x_1-x_4-1.2(1.3-\lvert x_1+x_2\rvert)$
$f_2(T_2)$	$\lvert x_3(0.8+x_2)+1.7+\lvert 2x_4-0.6\rvert\rvert$
$f_3(T_1)$	$\lvert 1.2+x_3+x_4-0.06\rvert$
$f_3(T_2)$	$\lvert 1.2-x_1-x_2+2x_3-x_4-x_3x_5\rvert$
$f_4(T_1)$	$\lvert 1.2+x_3+2x_4+0.05-\lvert x_1+x_2-0.05\rvert-x_4-2x_2\rvert$
$f_4(T_2)$	$\lvert 1.2+x_2+2x_4+0.05+(\lvert x_1+2x_2)+x_4\rvert+x_2\rvert$
$f_5(T_1)$	$\lvert -0.46+2x_3-x_1^2\rvert-0.17(0.17-x_2)$
$f_5(T_2)$	$\lvert 0.5-x_3+\lvert 1.2+x_3\rvert-2x_1+x_4-1.2)\rvert$

$$x_{2,t} \quad = \quad 5+0.3x_{2,t-1}+v_{2,t}. \tag{16}$$

$$y_t = \tan(x_{2,t}/x_{1,t})+w_t. \tag{17}$$

Case C:時刻 T_1 までは $d=0$ であり, 時刻 T_1+1 以降において $d=1$ となる.

シミュレーションを通じて $v_{i,t}, w_t$ は平均がゼロで標準偏差がそれぞれ 0.3, 0.5 である正規乱数であると仮定しておく. なおこれらの事例においては, 構造変化への追従を中心的に議論するので, 状態推定の結果については省略する.

図 4 には構造変化の前後において, GP 手法により状態方程式の推定を行って粒子フィルタを適用した場合 (with-updata) と, 特にこのような構造変化への対応を行わなかった場合 (without-updata) の, 尤度の時間変化を示している. この図 4 より分かるように, 構造変化を検出して GP により状態方程式の推定と状態方程式の切り替えを行った場合には, 極端な尤度の低下は見られない. 構造変化を考慮し関数推定をした場合に得られるモデルの尤度 l_t の平均値 (L1) と, 特に関数推定をしない場合のモデルの尤度 l_t の平均値 (L0) との差 L_D は次のようになる.

Case A:$L_D=3.52$, **Case B**:$L_D=3.49$, **Case C**:$L_D=3.9$.

これらのケースにおいては, 定数項や係数のバイアスの推定であるので, GP 手法により推定された関数の形状についての結果は, 簡単に以下のようにまとめておく. 時刻 t_c+1 以降の時系列を用いて推定された $a, b, \ldots$ の値 $\hat{a}, \hat{b}, \ldots$ は次のようになる.

Case A:$\hat{a}=0.99, \hat{b}=1.95$, **Case B**:$\hat{c}=0.38$,**Case C**:$\hat{d}=1.31$.

これらの結果から分かるように, 構造変化の後において推定された定数の値は, 変化の前の関数から変形された形状に近いことがわかる.

5　むすび

本論文では, 粒子フィルタおよび GP を用いた構造変化の推定とその応用について述べた. 非線形方程式が現在の関数形から派生すると仮定し,GP 手法における交叉処理によりこの変形後の関

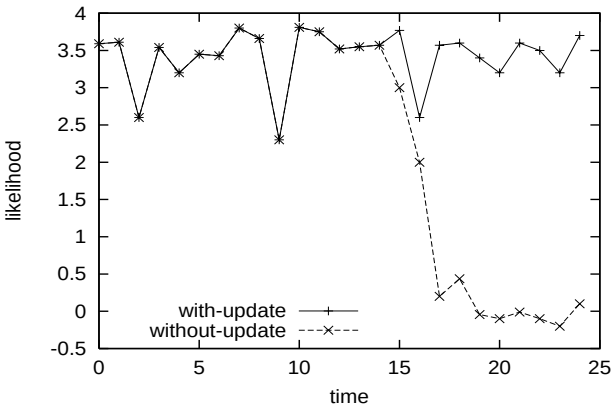

図 4: 定数記述にバイアスを含む構造変化の前後での尤度の時間変化の例

数形を推定する方法を示した. 応用例として人工的なデータを用いて, 関数を推定するための条件を検討し, 同時に現実のデータに対する適用可能性を検証した.

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] 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.

[2] A. Doucet, N. de Freitas and N. Gordon, *Sequential Monte Carlo Methods in Practice*, New York, Splinger Verlag, 2001.

[3] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson and P. Nordlund, “Particle filters for positioning, navigation, and tracking,” *IEEE Trans. Signal Processing*, vol.50, no.2, pp.425–437, 2002.

[4] J. H. Kotecha and P. M. Djuric, “Gaussian sum particle filtering,” *IEEE Trans. Signal Processing*, vol.51, no.10. pp.2602–2612, 2003.

[5] T. Schon, F. Gustafsson and P. J. Nordlind, “Marginalized particle filters for mixed linear/nonlinear state-space models,” *IEEE Trans. Signal Processing*, vol.53, no.7, pp.2279–2289, 2005.

[6] A. Zia, T. Kirubrajan, J. P. Reilly, D. Yee, K. Punithakumar and S. Shirani, “An EM algorithm for nonlinear state estimation with model uncertainty,” *IEEE Trans. Signal Processing*, vol.56, no.3, pp.921–936, 2008.

[7] 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.

[8] G. C. Chow, “Tests of equality between sets of coefficients in two linear ergressions,” *Econometrica*, vol.28, no.3, pp.591–605, 1960.

[9] 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.

[10] N. Vaswani, A. RoyChowdhury and R. Chellapa, “Shape activity: A continuous state HMM for moving/deforming shapes with application to abnormal activity detection,” *IEEE Trans. Image Processing*, vol.14, no.10, pp.1603–1616, 2005.

[11] A. Andreiu, A Doucet, S. S. Singh and B. Tadic, “Particle methods for change detection, system identification and control,” *Proc. IEEE*, vol.93, pp.423–438, 2004.

[12] V. V. Kadirkamanathan, 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.

[13] P. Li and V. Kadirkamanathan, “Particle filters based likelihood ratio approach to fault diagnosis in nonlinear stochastic systems,” *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol.31, no.3, pp.337–343, 2001.

[14] H. Hammouri, M. Kinnaert and E.Yaagoubi, ’Observer-based approach to fault detection and isolation for nonlinear systems,” *IEEE Trans. Automatic Control*, vol.44, no.10, pp.1879–1884, 1999.

[15] 時永祥三, 池田欽一, “非線形時系列モデルにおける粒子フィルタおよび遺伝的プログラミングを用いた構造変化の推定,” 信学技報, NLP–110–465, pp.117-122, 2011.

[16] 陳暁栄, 時永祥三, “共進化 GP を用いたマルチエージェントシステムの構成とその人工市場分析への応用,” 信学論, volJ86–A, no.10, pp.1038–1048, 2003.

[17] Y. Ikeda and S. Tokinaga, “Chaoticity and fractality analysis of an artificial stock market by the multi-agent systems based on the co-evulutionary Genetic Programming,” *IEICE Trans. Fundamentals*, vol.E87–A, no.9, pp.2387–2394, 2004.

[18] 池田欽一, 陳暁栄, 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.

[19] 池田欽一, 時永祥三, 呂建軍, “遺伝的プログラミングと遅延とモグララフィを用いたネットワーク構成の同定と内部遅延時間の推定,” 情報処理学会論文誌, vol.47, No.SIG 1(TOM 14), pp.12–18, 2006.

[20] Y. Ikeda and S. Tokinaga, “Multi-fractality Analysis of time series in artificial stock market generated by multi-agent systems based on the Genetic Programming and its applications,” *IEICE Trans. Fundamentals*, vol.E90–A, no.9, pp.2212–2222, 2007.

[21] Y. Ikada and S. Tokinaga, “Discriminant analyses of stock prices by using multifractality of time series generated via multi- agent systems and interpolation based on wavelet transforms,” *Trans. IEICE Fundamentals*, vol.E93–A, no.10, pp.12–29, 2010.

[22] 時永祥三, 岸川善紀, “遺伝的プログラミングと多段ファジイ推論に基づくジャンプ過程を含む時系列生成モデルの推定,” 信学論, vol.J93–A, no.5, pp.365-374, 2010.

[23] 時永祥三, 池田欽一, “局所的交流をする平面上のエージェント特性で生成される信号のカオス・マルチフラクタル分析,” 信学論, vol.J94–A, No.6, pp.389–404, 2011.

[24] J. R. Koza, *Genetic Programming*, MIT Press, 1992.

[25] 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.

[26] H. L. Lee, K. C. So and C. S. Tang, “The value of information sharing in a two-level supply chain,” *Management Science*, vol.46, no.5, pp.626–643, 2000.

[27] 杉原敏夫, 適応的モデルによる経済時系列分析, pp.48–80, 工学図書, 1994.

[28] E. Mertens, “Structure shocks and the comovements between output and interest rates,” *Journal of Economic Dynamics and Control* , vol.34, pp.1171–1186, 2010.

[29] N. J. Gordon, D. J. Salmond and F. M. Smith, “Novel approach to nonlinear/nongaussian baysian state estimation,” *Proc. IEEE*, vol.140, no.2, pp.107–113, 1993.

[30] R. Karlsson and F. Gustaffson, “Recursive bayesian estimation-bearing-only applications,” *Proc. IEEE*, vol.152, no.5, pp.305–313, 1989.

時永　祥三〔九州大学大学院経済学研究院　教授〕
池田　欽一〔北九州市立大学経済学部　准教授〕