

遺伝的手法と粒子フィルタを用いた動的ベイズネットワーク記述における状態変化の推定

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

<https://doi.org/10.15017/25877>

出版情報 : 経済学研究. 79 (4), pp.25-41, 2012-12-20. 九州大学経済学会
バージョン :
権利関係 :

遺伝的手法と粒子フィルタを用いた動的ベイズネットワーク記述における状態変化の推定

時 永 祥 三
池 田 欽 一

1 まえがき

ベイズネットワーク (Bayesian Network:BN) はランダム変数 (確率変数) の間の結合の静的表現を, 方向性非巡回グラフ (Directed Acyclic Graph:DAG) を用いて図式的に表現する方法であり, さまざまな分野で応用されている [1]-[6]. 近年 DAG において, 時間的な遷移を導入した動的ベイズネットワーク (Dynamic BN:DBN) が提案され, 応用されている [7]-[12]. 同時に状態遷移に注目して DBN において観測されたデータから状態を推定する問題に関して, 粒子フィルタ (Particle Filter:PF) を適用する方法が提案されている [7][13]-[17]. しかしながら従来の DBN 手法においては DAG のとりうる範囲に関して, あらかじめ形状が既知であるグラフに限定され, 状態の存在確率も既知のパターンに属するとされており, 未知の DAG 形状および状態変化などには適用できない問題がある [7]. 本論文では, 遺伝的手法と粒子フィルタを用いた DBN 記述における状態変化の推定について述べる [18].

本論文ではまず最初に, DBN における基本的なモデルと DAG の形状が変化する場合の関係式を整理する [7]-[12]. 同時に, 遺伝的手法により変化したあとの DAG の形状変化および変数の状態遷移を推定する方法を述べる [5][18]. この場合, DAG の形状変化推定においては遺伝的プログラミング (Genetic Programming:GP) において用いられる交叉処理を含む一般的な手法全体ではなく, DAG の枝の方向を変えるなどの突然変異を基礎とした方法を用いて, 処理における整合性を保っている (文献 [5] における呼称に従い Evolutionary Programming:EP と呼ぶ)[5][18]. また状態遷移を表現するダイナミクス記述において関数 $f_k(z)$ を導入し, この形状を推定する場合に GP を適用する [19]-[24]. 次に状態推定の問題に関して, DBN においては状態変数の同時分布が時刻 t から時刻 $t+1$ にかけて変化することに注目し, 非線形状態方程式における状態推定に適用される PF を用いた状態推定の手法について述べる.PF による状態変化の推定においては DBN の構造および状態遷移を表現する粒子は複数個与えられるので, これらの粒子により表現される DAG の構造がケース表から得られる確率分布をどれくらい近似しているかを用いて粒子の重みとする. 応用例として, まず DAG の構造変化と状態変化が既知である人工データに対して本論文の手法を適用し, 推定を行った結果を評価することにより性能を検証するとともに, 現実のデータへの適用を考察する.

以下では, 2. において問題の定式化と EP・GP による DBN 記述の変化推定を述べ, 3. では PF による状態変化の推定について説明する.4. では応用例を示す.

2 問題の定式化と EP・GP による DBN 記述の変化推定

2.1 DBN と DAG

いま、確率変数のベクトル $X = (X_1, X_2, \dots, X_n)$ に関する観測データがあると仮定する。このようなデータベースを以下ではケース表 (table of cases) と呼んでおく [2]。BN は確率的なグラフモデルにより確率変数 X_1, X_2, \dots, X_n の間の関係を表現することが目的であり、その場合、関係表現から得られた確率分布と観測されたケース表から得られる確率分布とが一致する必要がある。このグラフにおけるノードは変数に対応しており、これらのノードの結合関係である DAG により変数の間の依存性を表現する。変数 X_k は変数の位置より先行する親の変数、すなわち DAG において変数 X_k に対応するノードに流入する方向で結合されているノードに対応する変数 X_j を用いた条件付確率により記述される。すなわち $P(X_k|Pa(X_k))$ として定義され、ここで $Pa(X_k)$ は変数 X_k の親の変数の集合を意味する。これにより確率変数 X_1, X_2, \dots, X_n の同時分布は、次のように表現できることが分かる [1]-[6]。

$$P(X) = \prod_{k=1}^n P(X_k|Pa(X_k)), X = (X_1, X_2, \dots, X_n). \quad (1)$$

このような最適な DAG を推定する問題は、ある DAG を仮定した場合に得られる同時分布が、与えられたケース表から得られる確率分布に等しくなるように DAG を求めることであり、遺伝的手法をはじめとしてさまざまな手法が提案されている [2]-[6]。しかしながら本論文では、現在の DAG 形状からの変化を推定することが課題であり、初期状態からスタートして最適な DAG を求める方法の詳細は省略する。

次にこのような BN について、状態変数の時間変化と DAG の時間変化を導入したモデル (DBN) へと拡張する。ただし、この時間変化はマルコフ過程に基づく遷移であると仮定し、時刻 t におけるベイズ推定の式には、時刻 $t-1$ における状態変数と DAG の形状変化が含まれると仮定する。確率変数のベクトルは時間 t を導入して表示し、 $X(t) = (X_1(t), X_2(t), \dots, X_n(t))$ としておく。ここで状態変数の遷移に関して、次のようなダイナミクスによる記述を用いることとし、そのための非線形関数 $f_k(\cdot)$ を導入する

$$X_k(t+1) = f_k(X_k(t)) + v_k(t). \quad (2)$$

ここで $v_k(t)$ は関数の近似誤差であり、平均がゼロで分散 σ_k^2 がである正規乱数とする。このようなダイナミクスを導入する理由としては、多くの状態推定問題において、観測される変数の値が真の値ではなく、何らかの変換 (歪み) を受けたものである場合が多いことがある [17][25]。また一般的には、ケース表の大きさが十分大きいと保証されない場合には、統計的な分布を考慮する必要があることも理由の 1 つである。

状態遷移を用いた式 (1) にならって表現すると $P(X(t)) = \prod_{k=1}^n P(X_k(t)|Pa(X_k(t)))$ となるが、この場合 $Pa(X_k(t))$ 、すなわち $X_k(t)$ に対応するノードへ流入するリンクのものとノードには、時刻 t における変数と同時に、時刻 $t-1$ における変数のいくつかが含まれることになる。更に DBN においては、時刻 t における $Pa(X_k(t))$ を表現するグラフを $G(t)$ とした場合には、この形状は $G(t-1)$ とは異なる可能性があることを仮定している。

一般的に、動的システムにおける状態推定の問題においては、観測データから変数 $X(t)$ の状態を推定することが行われるが、われわれは応用分野を考慮して、状態の遷移に用いられる関数 $f_k(z)$ を

推定する問題を対象としている. 同時に, DBN を表現するグラフの構造 $G(t)$ を推定する問題となる. したがって, 推定すべき対象 (状態) である $s(t)$ は次のようになる.

$$s(t) = [f(z), G(t)], f(z) = (f_1(z), f_2(z), \dots, f_m(z)). \quad (3)$$

推定に用いる観測データは, 変数 $X(t)$ の時刻 t にいたるまでの観測値である.

2.2 EP 適用による DAG 変化の推定

すでに述べたように観測されたデータからこれを表現する BN を推定する問題は, 本論文では直接的には取り扱わない. 本論文では, これらの BN における DAG 推定の方法の 1 つとして提案されている突然変異に基づく方法を用いて, DAG の変化を推定することを行う [5][6]. 遺伝的手法の代表的なものとして GP があり, 一般的に関数近似などに適用される [19]-[24]. しかしここでは, この一部のみを用いている. すなわち, GP 手法を適用する場合には個体の一部を置き換える突然変異のほかに関数を表現する 2 つの個体を途中で切断して, 交互の交叉させて, 子供 (offspring) を生成する方法を適用するが, ここでは突然変異だけを用いる.

EP 手法における突然変異の方法については, 文献 [5] に示されている方法を用いることにする. 現在の DAG の構造を表現するデータ (例えば接続行列など) を, 一般の GP 手法と同様に個体と呼んでおく. EP 手法は, ある時刻の個体の集合に対して遺伝的処理を行い, 適合度の大きい個体を次の処理に残す手法である. 適合度の定義は後で述べるが, その基本となるものは, ある個体に対応する DAG 構造を仮定して, 式 (1) の同時分布の確率を計算した場合に, ケース表から求めた確率をどの程度近似しているかにより定義される.

EP 手法は, 以下のような 4 つの手順としてまとめられる. 以下ではここにまとめられる手順を, あらためて EP 手法と呼ぶことにする.

手順 M1: 枝の追加

現在の DAG において任意に選択した 2 つのノード N_i, N_j の間に枝を挿入する. もしすでに存在している場合には, この操作は実施しない.

手順 M2: 枝の方向転換

現在の DAG において任意に選択した 2 つのノード N_i, N_j の間に存在する枝の方向を転換する.

手順 M3: 親の変数の変換

任意に選択したノード N_i へ流入する枝のもとにあるノードを, 別のノードに切り替える.

手順 M4: バイパスとなる枝の削除

EP 手法を適用している途中の段階では, ノードの間に冗長な枝が形成されることがある. 具体的には, 3 つのノード A, B, C がある場合に, A-B-C を経由するリンクのほかに, ノード A からノード C へと直接伸びる枝が形成されるケースである. この場合には, ノード A からノード C へと直接伸びる枝は構造を推定する場合には悪い影響を及ぼすので, この枝を削除する.

なお手順 M4 はデータを整理する場合にだけ用いるので, 突然変異の手順には加えない.

表 1: 確率近似を用いた EP 選択処理の効率化

$M - 1$	2 値	3 値	4 値
2	0.03	0.04	0.05
3	0.04	0.04	0.05
4	0.04	0.04	0.06
5	0.05	0.05	0.06

2.3 確率の近似を用いた EP 適用の効率化

時刻 $t - 1$ から t へのグラフ変化 ($G(t - 1)$ から $G(t)$ への変化) を推定する場合に, EP を適用してランダムに検索をする方法を用いることも可能であるが, 本論文では, 条件付確率に関する近似的な方法を利用して効率化をはかることにする. 具体的には, 時刻 $t - 1$ における変数 $X_k(t - 1)$ に関する親の変数の集合 $Pa(X_k(t - 1))$ から変数ノード (これをノード j としておく) を削除した場合に, 近似的に次のような関係がなりたつことを用いる.

$$P(X_k(t)|Pa(X_k(t))) \sim P(X_k(t - 1)|Pa(X_k(t))). \quad (4)$$

すなわち, 時刻 t における変数の親のノードに対応する変数の集合 ($Pa(X_k(t - 1))$ からノード j を削除する) を, 仮に時刻 $t - 1$ における変数の親のノードの変数の集合として用いた場合の条件付確率によって, 確率 $P(X_k(t)|Pa(X_k(t)))$ を近似することができる性質を用いる.

この近似を確認するために, 以下のようなシミュレーションを行ってみる (DBN においては, 以下のような部分的な形状が複数結合されているので, この部分を検証すれば特性を分析できる). いま $k = 1$ として, この変数 $X_1(t)$ の親のノードに対応する変数を $X_2(t), \dots, X_M(t)$ としておき, ノードの親のノードの集合からノード 2 を削除した場合に計算される, 式 (4) における近似度を求めてみる. 近似度の定義として, 式の左辺の近似 (これを $\hat{P}(X_k(t)|Pa(X_k(t)))$ としておく) と真の値 $P(X_k(t)|Pa(X_k(t)))$ との誤差 $r_P = (\hat{P}(X_k(t)|Pa(X_k(t))) - P(X_k(t)|Pa(X_k(t)))) / P(X_k(t)|Pa(X_k(t)))$ を用いる. また変数は, 離散値をとるケースを仮定しておく. 表 1 には相対誤差 r_P の絶対値の平均値を示している. 表の縦軸はノードに対する親のノードの個数 $M - 1$ であり横軸は変数のとる離散値の区別である. この表より分かるように, 確率 $P(X_k(t)|Pa(X_k(t)))$ を近似を用いて推定する方法は有用である. なお変数に状態遷移がともなう場合においても, 遷移が観測あるいは推定されているので条件は同じとなるので, 表 1 に類似した結果となる.

2.4 GP 手法による関数 $f_k(z)$ の変形

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いて, 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. なお GP 手法に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]-[23].

(ステップ 1)

乱数を用いて被演算子, 演算記号の並びからなる初期個体のプールを構成する.

(ステップ 2)

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, 個体における適合度を求める.

(ステップ 3)

適合度に比例する確率に応じて選択された 2 つの個体に対して遺伝的操作を行い, 新しい個体を生成し, 相対的に適合度の低い個体を, プールの個体により置き換える. 任意に個体を選択して被演算子, 演算記号の部分を用いたものに置き換える突然変異を適用する.

しかしながら, 本論文で議論するような変化をしたあと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することを仮定するが, この関数の形状は現在の関数 C の形状に類似する側面をもち, C 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, 関数 C を変形する相手の関数であるとみなすことができる. 現在の関数形 C に対して, ランダムに生成された GP の個体プールのそれぞれの個体に対して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 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 PF 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.

シミュレーションにより得られる結果として, 一般的な GP 手法による関数近似の場合とは異なり, 本論文で取り扱う問題においては, 変形を受ける関数 C が 1 つであり, この関数に対する交叉処理により変形後の関数 A_i を推定するので, 誤差の小さい近似関数が得られるまでの GP 手法適用の適用回数は, 極めて小さくなる.

確率的な状態遷移への対応

一方では状態変数の値は, 時刻 t から $t+1$ まで同レベルであることと同時に, 上方や下方へと確率的に遷移する可能性もある. このような動きをすべて単独の関数で表現する場合には, 関数 $f_k(z)$ の形状は極めて複雑なものになることが予想される. これを回避するためにここでは, 関数は (1) 線形, (2) 凸形, (3) 凹形の 3 つの形状からなり, この 3 つの形状からもっとも適した形状が選択されると仮定する (これらは $f_k(z)^{(A)}$, $f_k(z)^{(B)}$, $f_k(z)^{(C)}$ として区別されるが, 特に区別が必要ない場合には 3 つを含むとして, 単に $f_k(z)$ としておく). これらの 3 つの関数は, そのままの形で状態遷移の関数として用いられるほかに, すでに説明した GP 手法を適用して変形を受ける. この変形を施す原始関数は, 四則演算, 多項式のほかに対数関数, 指数関数を含み, さらに次のように定義される区分別線形関数がある (ただし, 変数 z について正規化した形を示す).

$$u(z) = 0, z \leq 0, u(z) = 1, z > 0$$

$$r(z) = 0, z \leq 0, r(z) = z, z > 0$$

通常の GP 処理では適合度が低下した個体は次の時刻では個体プールから削除されるが, 上に述べたような, 時間をかけて再び個体を使用される可能性もあることから, 修正を加える. すなわち, 3 種類の関数ごとにプールを作成して, 個体を使用されない場合でも, 次に使用される機会には個体プールが再現されるようにする.

3 PFによる状態変化の推定

3.1 PF手法の基本

本論文では、時刻 $t, t-1$ における遷移をモデルに取り入れている DBN を取り扱っているため、一般的な BN において実施される内部状態の推定をそのまま用いることはできない。したがって、非線形ダイナミクスをモデル化し、その状態を推定する方法として有効性が確認されている PF を適用する [7][13]-[18]。式 (3) に示される時刻 $t-1$ と時刻 t における状態変数や DAG の間の関係式は、その対象が確率変数であることが一般の非線形状態方程式とは異なっているが、時刻 $t-1, t$ の間の関係式とみなすと、同様な処理が可能である。

説明を分かりやすくするため、まず最初に、一般的な連続変数に対する PF の概要を整理しておく。システムを記述するダイナミクスを、次のように仮定しておく。

$$x_k(t+1) = F(x(t)) + v_k(t), y(t) = H(x(t)) + u(t), k = 1 \sim n. \quad (5)$$

ここで $x(t) = (x_1(t), x_2(t), \dots, x_n(t))$ は状態変数からなる n 次元ベクトルであり、 $y(t)$ は 1 次元の観測データである。 $v_k(t), u(t)$ はそれぞれ変数に付随するランダムノイズである。観測データである $y(t)$ から、連続値をとる変数 $x_i(t)$ のベクトル $x(t) = (x_1(t), x_2(t), \dots, x_n(t))$ を推定する問題である。PF 適用の概要は、以下のようにまとめられる。記号として、 $x_k(t), k = 1, 2, \dots, n$ の i 番目の粒子を $x_k(t)^{(i)}$ 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 $x_k(t)^{(i)}$ の初期値 $x_k(0)^{(i)}, i = 1, 2, \dots, N_P$ 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、システムノイズである $v_k(t)^{(i)}, k = 1, 2, \dots, n$ および $u(t)^{(i)}$ 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: $x_k(t)^{(i)}$ の 1 時刻先予測

$x_k(t)^{(i)}$ の 1 時刻先予測が次により与えられる。

$$x_k(t+1)^{(i)} = F(x(t)^{(i)}) + v_k(t)^{(i)}. \quad (6)$$

(4) 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。

$$w(t)^{(i)} = R_t(y(t+1)|x(t+1)^{(i)}). \quad (7)$$

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

$$W(t) = \sum_{i=1}^{N_P} w(t)^{(i)}. \quad (8)$$

(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 $\Omega_{t+1} = (x(t+1)^{(1)}, x(t+1)^{(2)}, \dots, x(t+1)^{(N_P)})$ から粒子 $x(t+1)^{(i)}$ を $w(t+1)^{(i)}/W(t+1)$ の確率に従って復元抽出して、生成する.

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

ステップ 2~ ステップ 5 の操作を、時刻 $t = 2, 3, \dots, T$ について繰り返す. この結果として、時刻 $t = 2, 3, \dots, T$ についても状態変数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.

$$L_m = \sum_{t=1}^T l_t - T \log N_P, l_t = \log W(t). \quad (9)$$

3.2 PF と EP・GP による DAG 変化と状態遷移推定

本論文では通常の PF とは異なり、DAG の変化と状態遷移を EP と GP により近似する方法を適用していることを考慮して、以下のような簡素化をはかっている.

(1) 状態変数を離散化する

一般的な PF による状態推定の問題においては、変数は連続的な値をとることを仮定している. しかしながら、このような連続量の導入により、記述すべきシステムの次元は極めて大きなものになる. 本論文においては変数の状態遷移と同時に DAG の変化も推定する必要があるので、変数 (確率変数) $X_k(t)$ は離散的な値をとると仮定しておく.

(2) EP により生成される DAG 区分の離散化と適合度 (重み)

次に EP により生成される DAG についても、複数のパターンを時刻 t における可能性のある DAG 形状として区分し、これを離散的な番号により管理することとする. 本論文では現在の DAG の形状が部分的に変化したものが、時刻 $t+1$ における DAG であると仮定しており、この変化後の DAG の形状を、番号 i により区分しておく. 時刻 t における変化後の DAG 形状として、存在する複数のパターンを $G(t)^{(i)}, i = 1, 2, \dots$ としておく. このパターンは PF における DAG 形状の推定においては、粒子に対応させられて用いられる.

(3) 変数の状態遷移 (ダイナミックス) の区分化

すでに述べたように本論文では、変数 (確率変数) $X_k(t)$ は関数 $f_k(z)$ を用いたダイナミックスにより記述されるが、この関数の連続的な変化を仮定すると、システムの次元が増大する. これを回避するために、関数の GP 手法による変形において生成される個体の性質を利用した区分化 (パターン化) を行う. いま時刻 t における関数 $f_k(z)$ は、この形状を基にしながら GP 処理により時刻 $t+1$ において用いられる可能性のある形状として変形される. この変形された関数は個体に対応しているので、適合度の大きい個体から有限個を用いる. 具体的には、この複数のパターンの中の i 番目のパターンを $f_k(z)^{(i)}$ としておく. このそれぞれのパターンは、PF における粒子の対応させられて推定に用いられる.

以上のような仮定のもとで、PF と EP・GP による DAG 変化と状態遷移推定の手順は次のようにまとめられる.

(1) ステップ 1:変数についてのケース表と DAG 構成の初期値を与える

観測されたデータから時刻 0 における状態変数に関するケース表の初期値が与えられていると仮定する. 同時に, このケース表を表現する DAG の構造が推定されており, これが初期値 $G(0)$ として与えられていると仮定する.

(2) ステップ 2: 次の時刻 $t+1$ における粒子の生成

次の時刻 $t+1$ における変数の状態遷移確率の i 番目のパターンに対応する粒子 $f_k(z)^{(i)}$, $k = 1, 2, \dots, n, i = 1, 2, \dots, N_k$, および時刻 $t+1$ における変化後の可能なパターンに対応する粒子 $G(t)^{(i)}$ について, それぞれ N_k, N_G セットの粒子の集合として生成する. この場合, 変数の状態遷移確率と変化後の DAG を示す粒子とを組み合わせたもの (2 種類つの粒子からなる対 $(f_k(z)^{(i)}, G(t+1)^{(i)})$ の集合) を, あらためて粒子としておく (区別するために, 複合粒子と呼んでおく).

(3) ステップ 3: 重みの計算

次に, それぞれ複合粒子 i の重み $w(t+1)^{(i)}$ を計算する. 本論文では, 時刻 $t+1$ におけるケース表は観測データとして与えられていると仮定しているため, 同時確率の近似度合いに応じて, 複合粒子の重みを計算する. いまケース表から得られる変数の同時確率を $P(X(t+1))$ としておく. 複合粒子 i により仮定されている遷移確率と DAG 構造のもとで得られる同時確率の計算値を, $\hat{P}(X(t+1))$ としておき, この数値が式 (7) におけると同様に, $P(X(t+1))$ を与えた場合の条件付確率にしたがうと仮定して重み $w(t+1)^{(i)}$ を求める.

(4) ステップ 4: 複合粒子のリサンプリング

それぞれの複合粒子を, 重みに従って, リサンプリングする. すなわち複合粒子の集合 Ω_{t+1} (対 $(f_k(z)^{(i)}, G(t+1)^{(i)})$ の集合) から粒子を $w(t+1)^{(i)}/W(t+1)$ の確率に従って復元抽出して, 生成する.

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

ステップ 2~ ステップ 5 の操作を, 時刻 $t = 2, 3, \dots, T$ について繰り返す.

4 応用例

4.1 人工的に生成された DBN における推定

本章で提案する EP・GP 手法と PF を用いた DBN における DAG 構造と状態の推定を現実のデータに適用する前に, 以下ではまず, 人工的なデータに適用して方法の性能を評価する. なお, シミュレーションの全体を通じてケース表の件数は 5000~20000 とし, DAG に含まれる 1 時刻前の変数の個数は全体の 1/3 程度としておく.

EP 手法適用の仮定

ある DAG を生成しこれを DAG-1 とした場合に, この DAG-1 においてランダムに選択したノードに対して EP 手法の手順 M1~M3 をランダムに適用して得られる DAG 構成 (一般には複数) を DAG-2 としておく. シミュレーションを行う手順において, 以下のことを仮定しておく.

仮定 1: 手順 M1~M3 の適用回数の制限

DAG-1 から DAG-2 を得るための手順 M1~M3 (M4 は用いない) の適用は, 最大でも 5 回に限定しておく. 適用回数 m は乱数により選択することとし, 1~5 回を適用する確率をはほぼ同じ値とし

ておく. この 1 回の手順適用において適用される手順の種類は, ランダムに選択する.

仮定 2:用いるケース表

構造変化のあとで得られる DAG-2 に対して, 観測データとしてケース表が与えられると仮定する. すなわち構造変化のあとでも, DAG 構造を推定するための観測データが得られていると仮定する. なおケース表そのものは, 推定の対象となる真の DAG(これを DAG-T とする) をもとにして乱数を用いて生成する. 人工データの場合には DAG-T は既知であるので, このようなデータは生成可能である.

仮定 3:状態変数のとる離散値の範囲

ノードに対応する状態変数は離散値をとり, その離散値の範囲は 2~4 としておく.

仮定 1 は, 本論文の対象が一般的な DAG の推定ではなく, すでに与えられている DAG-1 から有限の回数だけ変化した DAG-T の構造を推定する問題に限定していることに対応している. すなわち手順 M1~M3 の適用回数 m が多くなれば, ランダムな構造をもった DAG-2 の初期値からスタートして, 構造変化後の構成に最も近い形状の DAG-2 を推定する問題となるため, 既存の方法と大きな差異はなくなる. 次に, 仮定 2 については, 構造変化後においても観測データが利用可能であることを仮定しており, EP 手法により DAG-2 に対応する個体の適合度を計算する場合に用いる条件付確率 $P(X_k|Pa(X_k))$ の計算を, 随時行えることを仮定している.

推定すべき対象の生成

次に, 推定すべき対象である DAG および $f_k(z)$ の定義について述べる. EP 手法は DAG 形状の推定に用いられると同時に, シミュレーションにおいては, 推定対象である DAG をあらかじめ生成するためにも用いられる. 上に述べた DAG の生成の過程はすべて既知であるので, 与えられた問題は次のようになる. DAG-1 から手順 M1~M3 を有限回適用して得られる 1 つの DAG を DAG-T としておき, これを構造変化後の真の DAG であるとする. EP 手順を用いて DAG-1 から生成される DAG-2 の中から, 真の DAG-T を近似する DAG-2 を逐次的に推定し, 最終的に DAG-T と一致する DAG-2 を求める. このままでは 1 つの DAG だけの推定問題となるので, ある時刻 t においてこの DAG-T を, 別の DAG-T に置き換え推定対象を切り替える.

次に, ある $f_k(z)$ の基本形 ($f_k(z)^{(A)}, f_k(z)^{(B)}, f_k(z)^{(C)}$) に対して GP 手法を適用して, 複数の関数を生成しておき, ある時間間隔ごとに, この複数の関数形状を $f_k(z)$ として切り替えながら状態遷移関数として用い, 観測データの生成を行う. このように関数の基本形から派生した関数を $f_k(z)$ として用いて, しかも時間的に切り替えることにより, 本論文で提案すると PF 手法と EP・GP 手法を用いた DBN 記述推定を検証することが可能となる.

図 1 には DAG-1, DAG-T の例を示している. この図では構造変化手順が 3 回適用されている. 図 2 には, この場合の EP による DAG 構造変化推定の過程を示している. この図において横軸は時間であり, 縦軸は推定された時刻 t における真のグラフ形状 $G(t)^{(i)}$ に対応する粒子に与えられる PF による推定の過程で得られる重み $w_i(t)$ の, 全体の重み合計 $W(t)$ に対する比率 $r = w_i(t)/W(t)$ を示している. 図 2 から分かるように, 多少の変動は見られるが, 推定対象である真のグラフ形状に与えられる PF 処理における重みは 0.5 以上となっており, 識別が可能となっている.

次に図 3 には, 推定された状態遷移関数の変化について推定の過程を示している. この図において横軸は時間であり, 縦軸は推定された時刻における関数 $f_k(z)$ の形状の推定値の正しさが示され

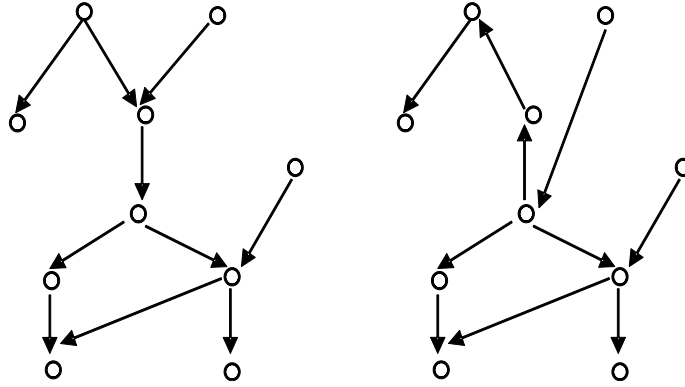


図 1: DAG 変形の例 (左:DAG-T, 右:DAG-1)

ている. EP と PF による形状の推定の場合と同様に, シミュレーションに用いる観測データを生成するため, ある時刻で適用された状態遷移の関数は分かっているので, この形状を GP・PF による推定により特定できているかを求める. 具体的には A, B, C の 3 つのグループのいずれかから GP 操作により派生した状態遷移関数を適用しているため, 推定された関数に対応する PF における粒子の重みを求めることで, 推定の正しさを検証している. 例えば真の状態遷移関数がグループ A に属すると仮定して, 推定された関数に対応する粒子に与えられる処理における重みが $w_i(t)$ であるとする, 粒子全体の重み合計に対する比を求める. 図 3 から分かるように, 推定対象である真の状態遷移関数に与えられる PF 処理における重みは 0.5 以上となっており, 識別が可能となっている.

次に手法により推定された同時確率 $\hat{P}(X(t))$ の関数と, ケース表より与えられた確率 $P(X(t))$ の差異を求め, この差異を $P(X(t))$ で割った相対誤差を計算する. 結果は相対誤差がほとんどの時刻でゼロになる単純なものであるため, 図示することは省略する. 推定された同時確率の推定誤差が発生するのは DAG 形状が切り替わったあとの数時刻であるが, その値は約 0.05 程度の極めて小さいものとなっている.

4.2 GP・EP と PF による状態変化推定の特性

次に DAG を記述するパラメータを変化させながら, GP・EP と PF による状態変化の推定を行った場合の特性について, シミュレーション結果をまとめながら考察する. シミュレーションの条件を, 以下に示す.

(1) DAG に含まれるノードの個数 N

$N = 5, 10, 15, 20$ のいずれかとする.

(2) DAG-T 生成のための手順 M1~M3 適用回数 m

$m = 1 \sim 5$ のうちいずれかとする. なおグラフのノード数と m との関係を考えて, $N = 5, N = 10$ の場合は, それぞれ $m = 1 \sim 3, m = 1 \sim 4$ に制限をしておく.

これらのシミュレーションの条件のもとで, $N = 5, 10, 15, 20$ のそれぞれに対して乱数を用いて 50 個の DAG-1 をランダムに生成し, 更にこれらの DAG-1 のそれぞれに対して形状変化手順を適

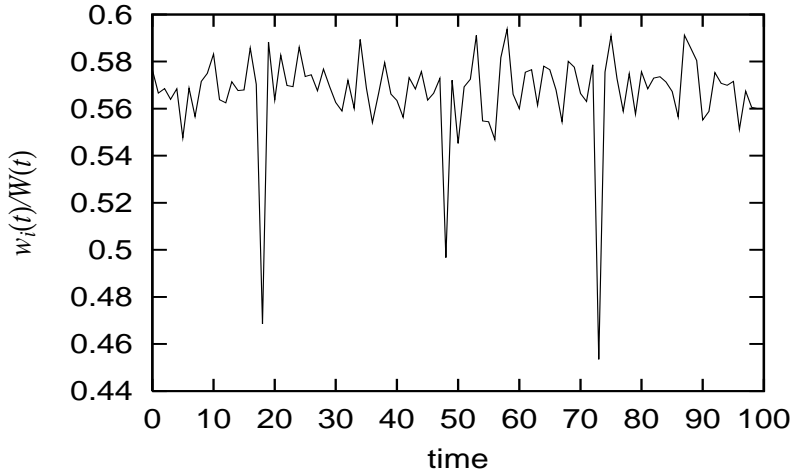


図 2: EP・PF による DAG 形状変化推定の例 (真の DAG に対応する粒子の PF における重みの比)

用して, DAG-T を求めている. なお, これらのシミュレーションにおける DAG 形状変化推定での EP 適用回数と推定誤差の関係や, 状態推定における真の値と推定値との関係は, すでに例として示した図 2~3 における事例と大きな差異はないので, ここでは図示することは省略する.

DAG 形状推定の特徴

表 2 には $N = 5, 10, 15, 20$ のケースごとに, DAG 形状変化適用回数 m と EP による DAG の形状推定における EP 適用回数の最小値 N_{GP} との関係を示している. なお表において m は DAG-T を得るための手順 M1~M3 の適用回数であり, DAG-T を推定する場合の EP 手法における手順 M1~M3 の適用回数ではない. この表 2 から分かるように, N の値が大きくなるにしたがって EP 適用回数の最小値 N_{EP} は大きくなっている. また同様に, 形状変化適用回数 m の増加にもなると N_{EP} の値も増加している. 形状の推定に必要な適用回数は $m \leq 2$ の場合には多くても 6 回程度であり, 形状が未知である場合に, ランダムな初期値からスタートして形状を推定する場合より, 極めて小さな値となっている. しかしながら $m > 2$ となる場合には, m が大きくなるにつれて N_{EP} も大きな値になっており, 多少の計算時間の増加は避けられないものとなっている.

次に表 3 には, 推定された $G(t)$ の形状が, あらかじめ与えている形状とどれくらい一致しているかを示している. すでに述べたように, PF 処理においては形状 $G(t)$ に対応して複数の粒子 $G(t)^{(i)}$ が生成され, それぞれが重み $w_i(t)$ をもっている. この重みの合計 $W(t)$ に占める比率が, 最も大きなもの (例えば $w_k(t)$ とする) が推定された DAG の形状であると判断される. 表 3 には, この重みの比率 $r = w_k(t)/W(t)$ を表現している. この結果から分かるように, 真の $G(t)$ 形状に対応する形状を与える DAG 形状の PF における粒子の重みの比率は 0.5 以上となっており, 他の粒子と区別して特定するには, 十分大きな数値となっている.

状態遷移推定の特徴

次に表 4 には, 推定された $f_k(z)$ が, あらかじめ与えている $f_k(z)$ とどれくらい一致しているかを示している. すでに述べた $G(t)$ に対応する粒子 $G(t)^{(i)}$ の重み $w_i(t)$ により形状推定を評価した方

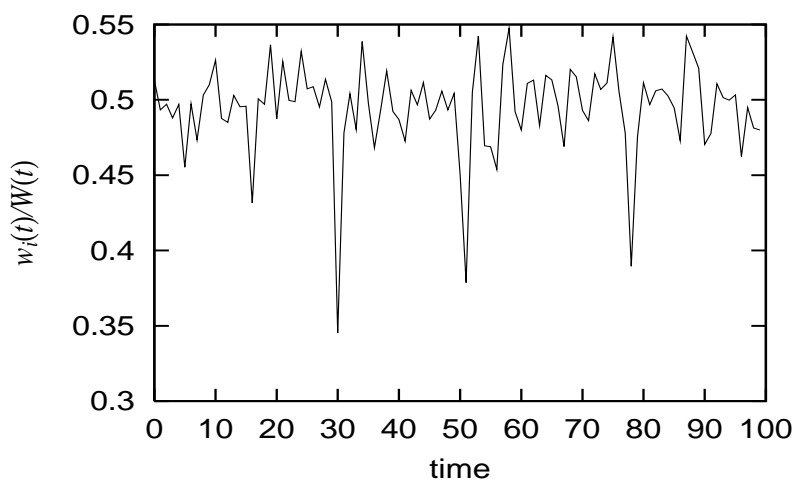


図 3: 状態遷移関数の推定の例 (真の関数に対応する粒子の PF における重みの比)

表 2: DAG 形状変化推定における EP 適用回数 N_{EP}

N	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
5	1.2	2.1	11.2	-	-
10	2.1	3.8	13.3	22.2	-
15	2.3	4.2	15.1	24.2	28.2
20	3.1	4.9	18.2	33.4	43.3

法と同様に、関数の推定を評価する。具体的には、推定された関数に対応する PF における粒子の重みを求めることで、推定の正しさを検証している。表 4 には、この重みの比率 $r = w_k(t)/W(t)$ を表現している。この結果から分かるように、真の $f_k(z)$ に対応する粒子の重みの比率は 0.5 以上となっており、他の粒子と区別して特定するには、十分大きな数値となっている。

なお、同時確率 $P(X(t))$ の推定結果も求めることができるが、ケース表から得られる $P(X(t))$ との相対誤差は極めて小さな値となることが確認できるので、ここでは詳細は省略する。

4.3 債券格付データへの適用

ここでは、本論文で示す DBN の DAG 形状および状態遷移推定の手法を現実のデータに対して応用する事例として、各国が発行する債券に関する格付データへの適用をとりあげる。債券格付は、企業や国が発行する債券について、その価値をいくつかのランクとして示すものであり、格付機関のほか、金融機関により実施されている。格付は同時に、市場での取引の価値（返済の可能性）を示しており、その時間的な変化を観測すると同時に、その原因を推定することも重要になっている。しかしながら、ここでは格付の詳細を説明することは適切ではないので、以下ではシミュレーションに

表 3: 推定された DAG 形状に対応する粒子の重みの比率

N	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
5	0.52	0.55	0.60	-	-
10	0.58	0.59	0.62	0.63	-
15	0.61	0.63	0.65	0.68	0.68
20	0.65	0.67	0.67	0.70	0.70

表 4: 推定された $f_k(z)$ に対応する粒子の重みの比率

N	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
5	0.57	0.56	0.55	-	-
10	0.58	0.59	0.59	0.55	-
15	0.62	0.63	0.63	0.65	0.65
20	0.65	0.68	0.67	0.67	0.67

よる分析で最低限必要な事項について整理し、問題を簡単化している。格付の値 (ランク) は格付機関により公表されているが、その方法論の詳細は公開されていないので、ここでは、密接に関連しているデータを用い DBN として関連性を推定する問題を考察する。観測された格付データの概要は以下のとおりである

観測年, 国:1990 年 ~2010 年における OECD 加盟国を主要として 37 カ国であり月次データを収集する。なお、それぞれの月次データを 1 つの分析対象とすると、十分な大きさのケース表とはならないので、1 年分をまとめて 1 つのケース表として仮定する。

観測変数:次のように与えている: 格付ランク, 財政赤字, 格付ランク,GDP, 内需, 最終消費支出, 総資本形成, 総固定資本形成, 在庫品増加, 総輸出, 総輸入, 雇用者報酬, 失業率, 為替レート。数値は、国民あたりの数値に変換されているものもある。

なお公表される財務指標の個数は極めて多く、そのまま用いることは適切ではないので、多変量解析の 1 つの手法である因子分析を用いて、以下のような 5 個の主要な変数だけを抽出して分析に用いている。

X_1 :格付, X_2 :GDP, X_3 :輸出, X_4 :失業率, X_5 :財政赤字

シミュレーションを実施するための仮定として、以下のことを用いている

(1) 変数の値の離散化

人工データに対するシミュレーションの場合と同様に、観測される変数の値は 3 段階として離散化して用いる。

(2) 変数変化のダイナミクス導入の制限

すべての変数に変数変化のダイナミクス導入すると、システムの次元が極めて大きなものとなるので、シミュレーションにおいては、変数 X_1 だけにダイナミクス記述をするように制限している。

なお人工データを用いたシミュレーションの場合とは異なり、現実の格付データ分析については、

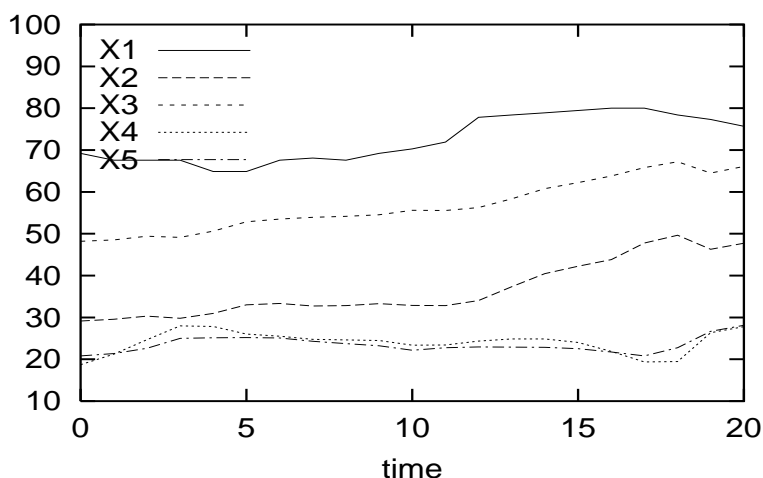


図 4: 変数の平均値の時間変化

初期の DAG-1 は与えられていないので、形状を推定する必要がある。与えられた観測データ（例えばケース表）から、適切な DAG を推定する方法は多数提案されているが、本論文ではこのような形状推定が直接の目的ではないので、やや簡単な方法で DAG-1 を推定することにする。具体的には、DAG の形状を乱数を用いてランダムに与えておき、この DAG-1 に対して形状変化手順を繰り返し適用する方法である。この方法では、特別な時間短縮の工夫はなされてはいないが、文献 [6] の事例にあるように、確実に DAG の形状を推定することができる。

図 4 には変数の平均値の時間変化を示す。また図 5 には推定された DAG の例を示す。現実のデータに対して本論文の DBN 記述の推定手法を適用したシミュレーション結果を、以下のように (1) DAG と $f_k(z)$ の変化回数、(2) $P(X(t))$ の推定誤差、(3) 推定された $f_k(z)$ について整理している。

(1) DAG と $f_k(z)$ の変化回数の推定

DAG, $f_k(z)$ の形状は、次の時刻において数回変化することが確認できる。

DAG:1995 年,2003 年,2009 年

$f_k(z)$:1991 年,1997 年,2003 年,2009 年

(2) $P(X(t))$ の推定誤差

人工データの場合とは異なり、現実のデータを用いたシミュレーションにおいては、ケース表から得られる同時確率 $P(X(t))$ の値と、DAG を仮定した場合の推定値 $\hat{P}(X(t))$ とは一致しないため相対誤差が発生する。この分析結果を図として示すことは省略するが、その範囲は 0.03~0.30 であり、平均値は 0.12 である。

(3) 推定された $f_k(z)$

表 5 には推定された $f_k(z)$ の事例を示している。格付は急激には変化していないことから、この関数の形状は比較的滑らかなものとなっている。

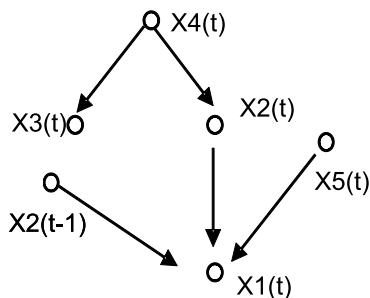


図 5: 推定された DAG の事例

表 5: 推定された $f_k(z)$ の例

$f_k(z) = 0.02z^2 + 0.7122z - 0.0056 - 0.01 \log(1 + z)$ $f_k(z) = -0.0326z^2 + 1.3661z + 0.0052 + 0.03 \exp(-0.02z)$

5 むすび

本論文では GP・EP と PF を用いた DBN ネットワーク記述における状態変化の推定について述べた。EP により DAG 形状変化を推定すると同時に、遷移関数を GP により推定し、これらの選択を PF における粒子の重みにより評価した。応用例として、人工的に生成された DBN に対して本論文の手法を適用するとともに、現実のデータへの適用を考察した。

今後の課題として、状態変数のダイナミクスをより一般的にしたケースへの適用可能性の考察があり、今後検討を進める予定である。

参考文献

- [1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems : Network of Plausible Inference*, Morgan Kaufmann, San Francisco, CA, 1988.
- [2] G. F. Cooper, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol.9, pp.309–347, 1992.
- [3] R. Etxeberria, P. Larranga and J. M. Picaza, "Analysis of behavior of genetic algorithm when learning bayesian network structure from data," *Pattern Recognition Letters*, vol.18, no.11–13, pp.1269–1273, 1997.
- [4] M. Wong, S. Y. Lee and K. S. Lenug, "A hybrid data-mining approach to discover bayesian networks using evolutionary programming," *Proceeding of the Genetic and Evolutionary Computation Conference*, pp.214–222, 2002.

- [5] M. Wong, S. W. Lam and K. S. Lenug, "Using evolutionary programming and minimum description length principle for data mining of bayesian networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.21, no.2, pp.174–178, 1999.
- [6] W. Lam, "Bayesian network refinement via machine learning approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.21, no.2, pp.174–178, 1999.
- [7] Z. Wang, E. E. Kuruoglu, X. Yang, Y. Xu and T. S. Huang, "Time varying dynamics Bayesian network for nonstationary events modeling and online inference," *IEEE Transaction on Signal Processing*, vol.59, no.4, pp.1553–1568, 2011.
- [8] K. Wang, J. Zhang, F. Shen and L. Shi, "Adaptive learning of dynamic Bayesian networks with changing structure by detecting geometric structure of time series," *Knowledge Intelligent System*, vol.17, no.1, pp.121–133, 2008.
- [9] S. H. Nielsen and T. D. Nielsen, "Adaptive Bayesian network structures to non-stationary domains," *International Journal of Approximate Reasoning*, vol.49, no.2, pp.379–397, 2008.
- [10] E. Punsakaya, C. Andrieu, A. Doucet and W. Fitzgerald, "Bayesian curve fitting using MCMC with applications to signal segmentation," *IEEE Transaction on Signal Processing*, vol.50, no.3, pp.747–758, 2002.
- [11] P. Fearnhead, "Exact and efficient Bayesian inference for multiple changepoint problems," *Statistical Computation*, vo.16, no.2, pp.203–213, 2006.
- [12] N. Friedman, K. Murphy and S. Russel, "Learning the structure of dynamic probabilistic networks," *Proc 14th Annual Conference of Uncertainty and Artificial Intelligence*, pp.139–214, 1998.
- [13] A. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Baysian tracking," *IEEE Transaction on Signal Processing*, vol.50, no.2, pp.174–189, 2002.
- [14] 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.
- [15] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson and P. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol.50, no.2, pp.425–437, 2002.
- [16] 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.
- [17] 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.

- [18] 時永祥三, 池田欽一, “遺伝的手法と粒子フィルタを用いた動的ベイズネットワーク記述における構造・状態推定,” 信学技報, SIS2011-46, pp.37-42, 2011.
- [19] 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.
- [20] Y. Ikeda and S. Tokinaga, “Analysis of price changes in artificial double auction markets consisting of multi-agents using Genetic Programming for learning and its applications,” *IEICE Trans. Fundamentals*, vol.E90-A, no.9, pp.2203-2211, 2007.
- [21] 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.
- [22] 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.
- [23] 呂建軍, 時永祥三, “遺伝的プログラミングによるルール生成を用いたクラスタ特徴記述システムの構成とその応用,” 信学論, vol.J89-A, no.12, pp.1142-1152, 2006.
- [24] 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.
- [25] M. W. Korolkiewicz and R. J. Elliott, “A hidden Markov model of credit quality,” *Journal of Economic Dynamics & Control*, vo.32, pp.3807-3819, 2008.

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]