

曖昧不規則データを用いた 1 階自己回帰モデルのパラメータ推定に 関するシミュレーション実験研究*

福 田 得 夫

概 要

本論文では、ある種の曖昧不規則データからシステム内に含まれる未知パラメータの曖昧推定値を求める手法についてシミュレーション実験により数値的に考察する。本論文で対象とする曖昧不規則データは本質的に曖昧性と不規則性を同時に有するデータであり、通常の不規則過程の見本過程を曖昧に識別することで得られるものである。

まず、ファジィ集合の集合表現を用いて、ファジィ確率過程およびその2次までの統計学的モーメントの合理的な定義法を、次に曖昧不規則データのみを用いたファジィ確率過程の統計学的モーメントの推定法を概観する。最後に、時系列モデルのうち最も簡単と思われる、1階の自己回帰モデル内に含まれる未知パラメータを、曖昧不規則データのみから推定する手法を導出し、プログラムパッケージ化することを試みる。さらに構築したプログラムパッケージを用いてシミュレーション実験をおこない、推定法ならびにプログラムパッケージの有効性を数値的に検証する。

キーワード：曖昧不規則データ、ファジィ確率過程、ファジィ集合の集合表現

* 本研究の一部は文部省科学研究費補助金（基盤研究(C)(2) No. 11680461）の補助を受けた。

1. 曖昧不規則過程

1.1 曖昧ベクトル

通常の実数値 $x_0 \in \mathbb{R}$ の曖昧識別として与えられるファジィ集合 $\tilde{x}$ は三つ組

$$\tilde{x} = (\mathbb{R}, \{\tilde{x}_\alpha \mid \alpha \in (0, 1)\}, s_{\tilde{x}}) \quad (1.1)$$

で定義される。ここで、 $s_{\tilde{x}}$ は $x \in \mathbb{R}$ の各要素に命題

$$s_{\tilde{x}}(x) = \{x \text{ は } x_0 \in \mathbb{R} \text{ に一致する}\} \quad (1.2)$$

を与える述語であり、 $\{\tilde{x}_\alpha \mid \alpha \in (0, 1)\}$ は任意の $\alpha \in (0, 1)$ に対して

$$L_\alpha \tilde{x} \subseteq \tilde{x}_\alpha \subseteq L_{\tilde{x}} \tilde{x} \quad (1.3)$$

を満たす $\tilde{x}$ の集合表現である。また、(1.3) の $L_\alpha \tilde{x}$ と $L_{\tilde{x}} \tilde{x}$ は、各々 $\tilde{x}$ のレベル α での強カット、レベル集合と呼ばれるものである [1]。

次のような性質を有するファジィ集合の集まりを G_c で表すことにする。
すなわち

- (i) $\bigcap_{\alpha \in (0, 1)} \tilde{x}_\alpha \neq \emptyset$
- (ii) 各 $\alpha \in (0, 1)$ に対して $\tilde{x}_\alpha$ は閉区間、
- (iii) $\text{supp. } \tilde{x} = \text{cl. } \bigcup_{\alpha \in (0, 1)} \tilde{x}_\alpha$, で与えられる $\tilde{x}$ のサポート $\text{supp. } \tilde{x}$ も閉区間。

G_c に属するファジィ集合を曖昧数またはファジィ数と呼ぶことにする。

1.2 ファジィ確率過程

文献 [1] で定義されているファジィ確率ベクトルを拡張することにより、ファジィ確率過程を定義する。まず、次のような確率差分方程式で記

述される確率システムを導入する：

$$x_k = f(x_{k-1}, \dots, x_{k-n}; e_{k_1}, \dots, e_{k-m}; k; \theta). \quad (1.4)$$

ここで $\{x_k\}$ はシステム出力である確率過程, $\{e_k\}$ は確率空間 $(\Omega, \mathfrak{A}, P)$ 上の同分布かつ独立な確率変数系列でモデル化される観測不可能な不規則外乱, θ は対象とするシステム内の未知パラメータをベクトル表現したものである。曖昧不規則データ $\{\tilde{x}_k; k=1, 2, \dots\}$ は, 出力過程 $\{x_k\}$ の曖昧識別として得られるものと考えことにする。

ところで, $\mathfrak{F}$ を曖昧識別による情報欠落を表す $\mathfrak{A}$ の部分 σ 代数とすると, 曖昧不規則データは次のように定義されるファジィ確率過程の各時刻における実現値の系列と考えるのが妥当であると思われる：

定義1.1. 通常の確率過程 $\{x_k\}$ の曖昧識別として得られるファジィ確率過程 $\{\tilde{x}_k; k=1, 2, \dots\}$ は

$$\tilde{x}_k: \Omega \rightarrow \mathbb{G}_c \quad (k=1, 2, \dots) \quad (1.5a)$$

によって与えられる。ここで,

$$\tilde{x}_k = (R, \{\tilde{x}_{k,\alpha} \mid \alpha \in (0, 1)\}, s_{x_k}) \quad (1.5b)$$

であり, $\{\tilde{x}_k(\omega)\}$ の集合表現の系列 $\{\tilde{x}_{k,\alpha}(\omega) \mid \alpha \in (0, 1)\} (k=1, 2, \dots)$ は見本空間 Ω から $\mathfrak{B}(\mathfrak{R}_c)$ への $\mathfrak{F}$ -可測な対応 (correspondence) である。ただし, $\mathfrak{R}_c$ は空集合を除く R 上の閉区間の族, $\mathfrak{B}(\mathfrak{R}_c)$ は $\mathfrak{R}_c$ の開集合から生成される最小の σ 代数である¹⁾。また, ファジィ確率過程を与える元になる確率過程 $\{x_k\}$ をオリジナル確率過程と呼ぶことにする。さらに,

1) $\mathfrak{R}_c$ の位相は Hausdorff 距離によって誘導される。

オリジナル確率過程の候補となりうるすべての確率過程の集合を、オリジナル確率過程の許容集合と呼び $\mathfrak{V}$ で表示することにする。

本論文では、オリジナル確率過程の許容集合 $\mathfrak{V}$ は次の性質を満たすものと仮定する：

(A-1) $\{x_k\}$ (オリジナル確率過程) $\in \mathfrak{V}$

(A-2) $x_k \in \text{supp. } \tilde{x}_k$ a.s. ($k=1, 2, \dots$)

(A-3) $\xi_k \in \bigcap_{\alpha \in (0, 1)} \tilde{x}_{k, \alpha}$ a.s. ($k=1, 2, \dots$) を満たす $\{\xi_k\} \in \mathfrak{V}$ が存在する。

(1.4) でモデル化されるシステムの出力過程を曖昧に識別することによりファジィ確率過程が得られるという立場においては、(A-1) および (A-2) は自然な条件であると思われる。また、(A-3) は $\tilde{x}_{k, \alpha}$ が各時刻ステップ $k=1, 2, \dots$, ならびに任意のレベル $\alpha \in (0, 1)$ において、対応 $\tilde{x}_{k, \alpha}$ が少なくとも一つのセレクションを確率 1 で持つことを保証している。

ファジィ確率過程のオリジナル確率過程の候補は無数にあるのが普通である。したがって、次のように定義することは合理的であると思われる：

定義1.2. オリジナル確率過程の許容集合 $\mathfrak{V}$ 内のすべての確率過程が独立過程であるとき、ファジィ確率過程 $\{x_k \in G_c; k=1, 2, \dots\}$ は独立であるという。また、 $\mathfrak{V}$ 内のすべての確率過程が定常過程であるとき、 $\{\tilde{x}_k\}$ を定常過程と呼ぶことにする。

2. ファジィ確率過程の統計学的モーメント

確率過程 $\{\xi_k\}$ をファジィ確率過程 $\{\tilde{x}_k\}$ のオリジナル過程の候補、すなわち、 $\mathfrak{V}$ の要素とする。このとき、 $\{\tilde{x}_k\}$ の統計学的モーメントを考察するために、まず最初に

$$s_{E\{\bar{x}_k\}}(x) = \{x \text{ は } x_k \text{ の期待値 } E\{x_k\} \text{ に一致する}\} \quad (2.1)$$

で与えられる命題を考察する。ただし、 $\{x_k\}$ は $\{\bar{x}_k\}$ のオリジナル確率過程である。そうすると (2.1) で与えられる命題は、複合命題

$$s_{E\{\bar{x}_k\}}(x) = \bigvee_{\xi \in \mathfrak{U}} \{(x = E\{\xi_k\}) \wedge (\xi_k = x_k \text{ a.s.})\} \quad (2.2)$$

で書き直すことができる。ここで、 $\xi = \{\xi_k; k=1, 2, \dots\}$ と略記した。さらに、(2.2) 内の命題 $(\xi_k = x_k \text{ a.s.})$ は

$$(\xi_k = x_k \text{ a.s.}) = \bigwedge_{\omega \in \Omega \setminus N_0} s_{\bar{x}_k}(\xi_k, \omega) \quad (N_0 \text{ は零集合}) \quad (2.3)$$

および

$$s_{\bar{x}_k}(\xi_k, \omega) = \{\xi_k(\omega) \text{ は } x_k(\omega) \text{ に一致する}\} \quad (2.4)$$

で与えられる。そうすると、(2.2) は最終的には

$$s_{E\{\bar{x}_k\}}(x) = \bigvee_{\xi \in \mathfrak{U}} \left\{ (x = E\{\xi_k\}) \wedge \left(\bigwedge_{\omega \in \Omega \setminus N_0} s_{\bar{x}_k}(\xi_k, \omega) \right) \right\} \quad (2.5)$$

と表すことができる。ここで、Zadeh が提案したファジィ集合に関する拡張原理 [2] の考え方を採用すると、 $s_{E\{\bar{x}_k\}}(x)$ の真理値は

$$t(s_{E\{\bar{x}_k\}}(x)) = \sup_{\xi \in \mathfrak{U}} \left\{ \text{essinf}_{\omega \in \Omega} t(s_{\bar{x}_k}(\xi_k, \omega)) \mid x = E\{\xi_k\} \right\} \quad (2.6)$$

と評価するのが妥当であると思われる。ただし、ここで $\text{essinf}_{\omega \in \Omega} t(s_{\bar{x}_k}(\xi_k, \omega))$ は

$$t(s_{\bar{x}_k}(\xi_k, \omega)) = (\bar{x}_k(\omega))(\xi_k) \geq a \text{ a.s.}$$

を満たす a の上限である。

ところで、文献 [1] で定義された擬 Aumann 積分を用いると、 $\bar{x}_{k,a}$ の期待値は

$$E\{\tilde{x}_{k,\alpha}\} = \{x \mid x = E\{\xi_k\}, \xi_k \in S(\tilde{x}_{k,\alpha})\} \quad (2.7a)$$

で与えられる。ただし,

$$S(\tilde{x}_{k,\alpha}) = \{\xi_k \mid \xi_k \in \mathfrak{Y}; \xi_k(\omega) \in \tilde{x}_{k,\alpha}(\omega) \text{ a.s.}\} \quad (2.7b)$$

である。また, オリジナル確率過程の許容集合 $\mathfrak{Y}$ が条件 (A-3) を満たす可積分確率過程で構成されているとき, 各 $k=1, 2, \dots$ における任意の $\alpha \in (0, 1)$ に対して

$$\{x \mid t(s_{E\{\xi_k\}}(x)) > \alpha\} \subseteq E\{\tilde{x}_{k,\alpha}\} \subseteq \{x \mid t(s_{E\{\xi_k\}}(x)) \geq \alpha\} \quad (2.8)$$

であることを容易に示すことができる。したがって, ファジィ確率過程 $\{\tilde{x}_k\}$ の平均過程を次のように定義することが合理的であると思われる:

定義2.1. $\{\tilde{x}_k; k=1, 2, \dots\}$ をファジィ確率過程, $\{\tilde{x}_{k,\alpha} \mid \alpha \in (0, 1)\}$ を $\tilde{x}_k$ の集合表現であるとする。オリジナル確率過程の許容集合 $\mathfrak{Y}$ の各要素が可積分な確率過程であるとき, $\{\tilde{x}_k\}$ の平均過程は

$$E\{\tilde{x}_k\} = (R, \{E\{\tilde{x}_{k,\alpha}\} \mid \alpha \in (0, 1)\}, s_{E\{\xi_k\}}) \quad k=1, 2, \dots \quad (2.9)$$

で与えられる。ここで $s_{E\{\xi_k\}}$ は (2.1) の命題を与える述語である。また, $\{E\{\tilde{x}_{k,\alpha}\} \mid \alpha \in (0, 1)\}$ は (2.7a) で与えられる $E\{\tilde{x}_k\}$ の集合表現である。

ファジィ確率過程 $\{\tilde{x}_k; k=1, 2, \dots\}$ が定常であるとき, $E\{\tilde{x}_k\}$ は一定のファジィ集合, すなわち

$$E\{\tilde{x}_k\} = m_x \quad (k=1, 2, \dots) \quad (2.10)$$

であることは明らかである。

ファジィ確率過程の2次の統計学的モーメントについては, 平均過程の場合と同様な議論により次のような定義が妥当であると思われる:

定義2.2. $\{\tilde{x}_k; k=1, 2, \dots\}$ をファジィ確率過程, $\{\tilde{x}_{k,\alpha} \mid \alpha \in (0, 1)\}$ を $\tilde{x}_k$ の集合表現であるとする。オリジナル確率過程の許容集合 $\mathfrak{Y}$ の各要素が自乗可積分な確率過程であるとき, $\{\tilde{x}_k\}$ の2次モーメント $\phi_x(k, k+\tau)$ は

$$\phi_x(k, k+\tau) = (R, \{\phi_{\tilde{x}_\alpha}(k, k+\tau) \mid \alpha \in (0, 1)\}, s_{\phi_{\tilde{x}}(k, k+\tau)}) \quad (2.11)$$

で与えられる。ここで, $s_{\phi_{\tilde{x}}(k, k+\tau)}$ は命題

$$s_{\phi_{\tilde{x}}(k, k+\tau)}(x) = \{x \text{ は } \phi_x(k, k+\tau) \text{ に一致する}\} \quad (2.12)$$

を与える述語である。ただし, $\phi_x(k, k+\tau) = E\{x_k x_{k+\tau}\}$ である。さらに, $\{\phi_{\tilde{x}_\alpha}(k, k+\tau) \mid \alpha \in (0, 1)\}$ は $\phi_{\tilde{x}}(k, k+\tau)$ の集合表現で (2.7b) で定義された $S(\tilde{x}_{k,\alpha})$ を用いると

$$\phi_{\tilde{x}_\alpha}(k, k+\tau) = \{x \mid x = \phi_{\xi}(k, k+\tau), \xi_k \in S(\tilde{x}_{k,\alpha})\} \quad (2.13)$$

で与えられる。同様に, $\{\tilde{x}_k\}$ の自己共分散関数 $\phi_x(k, k+\tau)$ については

$$\phi_x(k, k+\tau) = (R, \{\phi_{\tilde{x}_\alpha}(k, k+\tau) \mid \alpha \in (0, 1)\}, s_{\phi_{\tilde{x}}(k, k+\tau)}) \quad (2.14)$$

で与えられる。ただし, $s_{\phi_{\tilde{x}}(k, k+\tau)}$ は命題

$$s_{\phi_{\tilde{x}}(k, k+\tau)}(x) = \{x \text{ は } \phi_x(k, k+\tau) \text{ に一致する}\} \quad (2.15)$$

を与える述語である。ここで

$$\phi_x(k, k+\tau) = E\{(x_k - E\{x_k\})(x_{k+\tau} - E\{x_{k+\tau}\})\},$$

および

$$\phi_{\tilde{x}_\alpha}(k, k+\tau) = \{x \mid x = \phi_{\xi}(k, k+\tau), \xi_k \in S(\tilde{x}_{k,\alpha})\} \quad (2.16)$$

である。

$\{\tilde{x}_k\}$ が定常ファジィ確率過程のとき、その 2 次モーメント $\phi_{\tilde{x}}(k, k+\tau)$ と自己共分散関数 $\phi_{\tilde{x}}(k, k+\tau)$ は時間差 τ のみに依存する。したがって、それらを各々

$$\phi_{\tilde{x}}(\tau) = \phi_{\tilde{x}}(k, k+\tau) \quad (2.17a)$$

および

$$\phi_{\tilde{x}}(\tau) = \phi_{\tilde{x}}(k, k+\tau) \quad (2.17b)$$

と表現することにする。

3. パラメータ推定

3.1 統計学的モーメントの推定値

$\{\tilde{x}_k; k=1, 2, \dots\}$ を定常ファジィ確率過程とする。そうすると、期待値 $E\{\tilde{x}_k\}$ の推定値としてファジィ集合

$$\hat{m}_{\tilde{x}}(N) = (R, \{(\hat{m}_{\tilde{x}}(N))_{\alpha} \mid \alpha \in (0, 1)\}, s_{\hat{m}_{\tilde{x}}(N)}) \quad (3.1)$$

を与えるのが自然であると思われる。ここで、述語 $s_{\hat{m}_{\tilde{x}}(N)}$ は命題

$$s_{\hat{m}_{\tilde{x}}(N)}(v) = \left\{ v \text{ は } \frac{1}{N} \sum_{k=1}^N x_k \text{ に一致する} \right\} \quad (3.2)$$

を与える。そうすると、任意の $\alpha \in (0, 1)$ において

$$(\hat{m}_{\tilde{x}}(N))_{\alpha} = \frac{1}{N} \sum_{k=1}^N \tilde{x}_{k, \alpha}, \quad (3.3a)$$

また、任意の $\alpha \in [0, 1)$ において

$$L_{\alpha} \hat{m}_{\tilde{x}}(N) = \frac{1}{N} \sum_{k=1}^N L_{\alpha} \tilde{x}_k, \quad (3.3b)$$

さらに任意の $\alpha \in (0, 1)$ において

$$L_{\bar{\alpha}} \hat{m}_x(N) = \frac{1}{N} \sum_{k=1}^N L_{\bar{\alpha}} \tilde{x}_k \quad (3.3c)$$

であることがわかる。したがって、

$$\hat{m}_x(N) = \frac{1}{N} \sum_{k=1}^N \tilde{x}_k \quad (3.4)$$

と表現することが妥当であると思われる。

2次モーメント $\hat{\phi}_x(\tau)$ の推定値については、

$$\hat{\phi}_{x, N}(\tau) = \frac{1}{N-\tau} \sum_{k=1}^{N-\tau} \tilde{x}_k \tilde{x}_{k+\tau}, \quad (3.5)$$

および

$$(\hat{\phi}_{x, N}(\tau))_{\alpha} = \frac{1}{N-\tau} \sum_{k=1}^{N-\tau} \tilde{x}_{k, \alpha} \tilde{x}_{k+\tau, \alpha} \quad (\alpha \in (0, 1)) \quad (3.6a)$$

$$L_{\alpha}(\hat{\phi}_{x, N}(\tau)) = \frac{1}{N-\tau} \sum_{k=1}^{N-\tau} (L_{\alpha} \tilde{x}_k)(L_{\alpha} \tilde{x}_{k+\tau}) \quad (\alpha \in [0, 1)) \quad (3.6b)$$

$$L_{\bar{\alpha}}(\hat{\phi}_{x, N}(\tau)) = \frac{1}{N-\tau} \sum_{k=1}^{N-\tau} (L_{\bar{\alpha}} \tilde{x}_k)(L_{\bar{\alpha}} \tilde{x}_{k+\tau}) \quad (\alpha \in (0, 1]) \quad (3.6c)$$

と与えるのが適当であることを示すことができる。さらに、ファジィ確率過程 $\{\tilde{x}_k\}$ の自己共分散関数の推定値は、発見論的に

$$\hat{\phi}_{x, N}(\tau) = \hat{\phi}_{x, N}(\tau) - (\hat{m}_x(N))^2 \quad (3.7)$$

で与えることにする。ただし、(3.4)～(3.7)での四則演算はファジィ数に関する拡張原理に基づくものを採用する。

3.2 自己回帰モデルからの曖昧不規則データ

システムモデル (1.4) が n 階自己回帰 (AR(n)) モデル

$$x_k = -\theta_1 x_{k-1} \cdots -\theta_n x_{k-n} + e_k \quad (3.8)$$

で与えられるものとする。ここで、 $\hat{\theta}=(\hat{\theta}_1, \dots, \hat{\theta}_n)'$ は未知パラメータをベクトル化したものである。また、対象とするシステムの安定性を仮定すると次の条件が満たされている：

$$(B-1) \quad z^n + \sum_{i=1}^n \hat{\theta}_i z^{n-i} = 0 \quad \text{は単位円内に全ての根をもつ。}$$

第2節で述べたように、曖昧不規則データ $\{x_k\}$ はオリジナル確率過程を曖昧に識別することによって得られる。したがって、オリジナル確率過程の許容集合 $\mathfrak{Y}$ は n 階自己回帰モデルの出力過程によって構成されていると考えることは適当であると思われる：

(B-2) $\{\xi_k\} \in \mathfrak{Y}$ の各構成要素は

$$\xi_k = -\theta_1 \xi_{k-1} \cdots -\theta_n \xi_{k-n} + e_k \quad (3.9)$$

で記述される。ここで、 $\theta=(\theta_1, \dots, \theta_n)'$ は (B-1) において $\hat{\theta}$ を θ で置き換えた条件を満たす。

(3.8) で記述されるオリジナル確率過程 $\{x_k\}$ は、次の等式を満たすことを容易に確認することができる：

$$\phi_x(\tau) + \sum_{i=1}^n \hat{\theta}_i \phi_x(\tau-i) = 0 \quad (\tau=0, 1, \dots, n) \quad (3.10a)$$

$$\sigma_e^2 = \phi_x(0) + \sum_{i=1}^n \hat{\theta}_i \phi_x(i) \quad (3.10b)$$

$$m_e = \left(1 + \sum_{i=1}^n \hat{\theta}_i\right) m_x. \quad (3.10c)$$

ここで、 $m_x = E\{x_k\}$ および

$$\begin{aligned} \phi_x(\tau) &= E\{(y_k - m_x)(y_{k+\tau} - m_x)\} \\ &= E\{x_k x_{k+\tau}\} - (m_x)^2 \end{aligned} \quad (3.10d)$$

である。また m_e と σ_e^2 は各々 $\{x_k\}$ の平均および分散である。それゆえ、(3.8) から直接観測データが得られるものと仮定すると、 $E\{x_k\}$ と $\phi_x(\tau)$

$(\tau=0, 1, \dots, n)$ の推定値の関数として未知パラメータの推定が得られる。
すなわち, $\hat{\theta}$, $\hat{m}_e$ と $\hat{\sigma}_e^2$ の推定値は

$$\hat{\phi}_{x,N}(\tau) + \sum_{i=1}^n \hat{\theta}_i \hat{\phi}_{x,N}(\tau-i) = 0 \quad (\tau=1, 2, \dots, n) \quad (3.11a)$$

$$\hat{\sigma}_e^2 = \hat{\phi}_{x,N}(0) + \sum_{i=1}^n \hat{\theta}_i \hat{\phi}_{x,N}(i) \quad (3.11b)$$

$$\hat{m}_e(N) = \left(1 + \sum_{i=1}^n \hat{\theta}_i\right) \hat{m}_x(N), \quad (3.11c)$$

を解くことにより得ることができる。

上述の議論から発見論的に $\hat{\theta}$, $\hat{m}_e$ および $\hat{\sigma}_e^2$ の曖昧推定値は

$$\hat{\phi}_{\hat{x},N}(\tau) + \sum_{i=1}^n \hat{\Theta}_{i,N} \hat{\phi}_{\hat{x},N}(\tau-i) = 0 \quad (\tau=1, 2, \dots, n) \quad (3.12a)$$

$$\hat{\Sigma}_{e,N}^2 = \hat{\phi}_{\hat{x},N}(0) + \sum_{i=1}^n \hat{\Theta}_{i,N} \hat{\phi}_{\hat{x},N}(i) \quad (3.12b)$$

$$\hat{M}_e(N) = \left(1 + \sum_{i=1}^n \hat{\Theta}_{i,N}\right) \hat{m}_{\hat{x}}(N) \quad (3.12c)$$

を制約条件 (B-1) と (B-2) の下で $\hat{\Theta}_{i,N} (i=1, 2, \dots, n)$, $\hat{M}_{e,N}$ および $\hat{\Sigma}_e^2$ に
関して解くことにより得ることにする。

4. シミュレーション実験

4.1 擬似曖昧不規則データ

$\{x_k; k=1, 2, \dots\}$ をファジィ確率過程 $\{\hat{x}_k; k=1, 2, \dots\}$ のオリジナル
確率過程とする。このとき, シミュレーション実験においては擬似曖昧不
規則データを,

$$(i-1)a + b < x_k \leq (i+1)a - b \quad (a/2 < b < a) \quad (4.1)$$

のときは

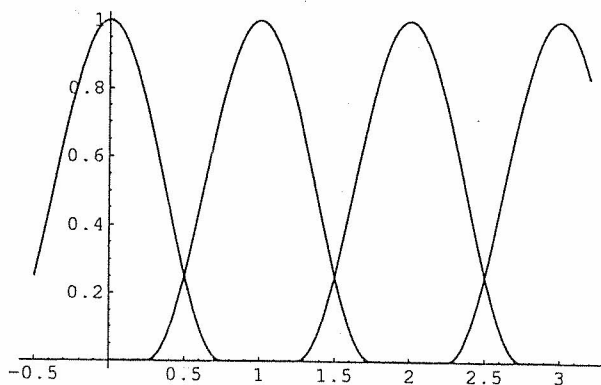


図4.1 $\bar{x}(i)$ のメンバーシップ関数 ($a=1, b=0.6$)

$$\bar{x}_k = \bar{x}(i) \quad (4.2)$$

と, また

$$(i+1)a - b < x_k \leq ia + b \quad (4.3)$$

のときは同確率で

$$\bar{x}_k = \bar{x}(i) \quad \text{または} \quad \bar{x}_k = \bar{x}(i+1) \quad (4.4)$$

をとるものと仮定する。ここで, $\bar{x}(i) (i=0, \pm 1, \dots)$ は「約 ia 」を表す曖昧数であり, そのメンバーシップ関数は図4.1に示したようないわゆる釣鐘型の形状を有するものとする。すなわち

$$\bar{x}(i)(x) = \begin{cases} \frac{\cos((\pi/b)(x-a)) + 1}{2} & (ia - b < x < ia + b \text{ のとき}) \\ 0 & (\text{その他の場合}) \end{cases} \quad (4.5)$$

である。したがって, 本論文で採用しているファジィ集合の集合表現を用いると, $\bar{x}(i)$ は

$$\tilde{x}(i) = (R, \{\tilde{x}_{i,\alpha} \mid \alpha \in (0, 1)\}, s_{\tilde{x}(i)}) \quad (4.6)$$

と表現できる。ただし、ここで $s_{\tilde{x}(i)}$ は

$$s_{\tilde{x}(i)}(x) = \{x \text{ は } ia \text{ に一致する}\} \quad (4.7a)$$

を与える述語であり、集合表現の各要素 $\tilde{x}_{i,\alpha}$ は閉区間

$$\tilde{x}_{i,\alpha} = \left[ia - \frac{b}{\pi} \cos^{-1}(2\alpha - 1), ia + \frac{b}{\pi} \cos^{-1}(2\alpha - 1) \right] \quad (4.7b)$$

である。

4.2 AR(1) モデル

簡単のために、もっとも簡単な1階自己回帰モデル

$$x_k = -\hat{\theta} x_{k-1} + e_k \quad (4.8)$$

で記述されるシステムをシミュレーション実験の対象とする。ここで、 $\hat{\theta}$ は未知パラメータであり、条件 (B-1) を満たすようにするために $|\hat{\theta}| < 1$ と仮定する。また、 $\{e_k\}$ は観測不可能なシステム外乱系列であり、平均 m_e 、分散 σ_e^2 の独立な正規分布を持つ確率変数列でモデル化されるものとする。第4.1節で述べた手法で生成された擬似曖昧不規則データから、その平均と2次モーメント $\phi_{\tilde{x}}(\tau)$ は (3.4) と (3.5) を用いて推定することができる。また、自己共分散の曖昧推定値は (3.7) より推定することができる。さらに、 $m = 1$ のときの (3.12a), (3.12b) および (3.12c) より

$$\hat{\phi}_{\tilde{x},N}(1) + \hat{\theta}_N \hat{\phi}_{\tilde{x},N}(0) = 0 \quad (4.9a)$$

$$\hat{\Sigma}_{\tilde{x},N}^2 = \hat{\phi}_{\tilde{x},N}(0) + \hat{\theta}_N \hat{\phi}_{\tilde{x},N}(1) \quad (4.9b)$$

$$\hat{M}_e(N) = (1 + \hat{\theta}_N) \hat{m}_{\tilde{x}}(N) \quad (4.9c)$$

を得ることができる。したがって、 $\hat{\theta}$, σ_e^2 および m_e の曖昧推定値は各々

$$\hat{\Theta}_N = - \frac{\hat{\phi}_{\hat{x}, N}(1)}{\hat{\phi}_{\hat{x}, N}(0)} \quad (4.10)$$

と (4.9b) さらには (4.9c) より得ることとする。ただし、(4.10) においても演算は拡張原理に基づいておこなうことにする。

以上の設定の下で、シミュレーション実験用のプログラムパッケージを試作した。実際の開発にあたっては、パーソナルコンピュータ (CPU Pentium II 450 MHz×2, メモリ 512 MB, 使用 OS Linux v.2.2) 上で、いわゆるフリーソフトウェアの C++ である GNU G++ (egcs-2.91.66) を開発言語として採用した。また、プログラムの一部は文献 [3] で示したプログラムパッケージを G++ 用に修正したものを再利用した。さらに、数値実験結果を可視化 (図形化) するために、Linux 用 Mathematica (v. 3.0) を用いた。付録 1 および 2 にそれぞれのプログラムを示す (再利用部分を除く)。また、システムパラメータ $\hat{\theta}$, 入力外乱 $\{e_k\}$ の平均 m_e および分散 σ_e^2 を各々 $\theta = 0.65$, $m_e = 0.0$, $\sigma_e^2 = 100$ とし、曖昧数 $\hat{x}(i)$ のメンバーシップ関数内の係数 a と b を $a = 1$ および $b = 0.6$ と設定した場合の、曖昧不規則データであるファジィ確率過程の典型的な見本過程を、そのメンバーシップ関数列によって示したのが図 4. 2 である。このようなメンバーシップ関数列を持つ曖昧不規則データのみから推定した $\hat{\Theta}_N$, $\hat{M}_e(N)$ および $\hat{\Sigma}_e^2(N)$ の見本過程のメンバーシップ関数列を、時間ステップ $N = 500$ まで、各々、図 4. 3, 図 4. 4 および図 4. 5 に示す。

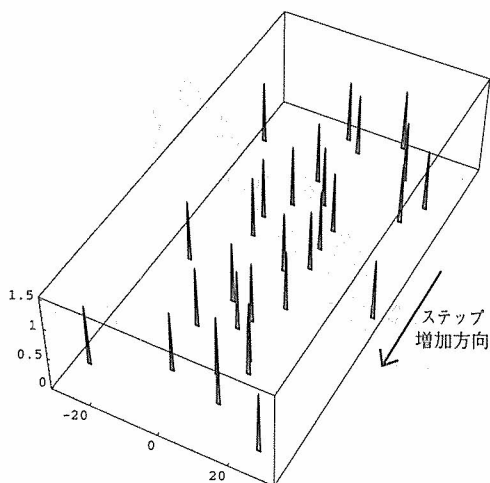
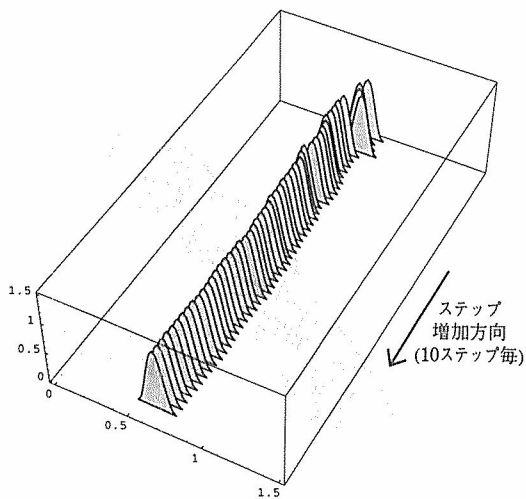


図4.2 曖昧不規則データのメンバーシップ関数の見本系列

図4.3 θ_N のメンバーシップ関数の見本系列

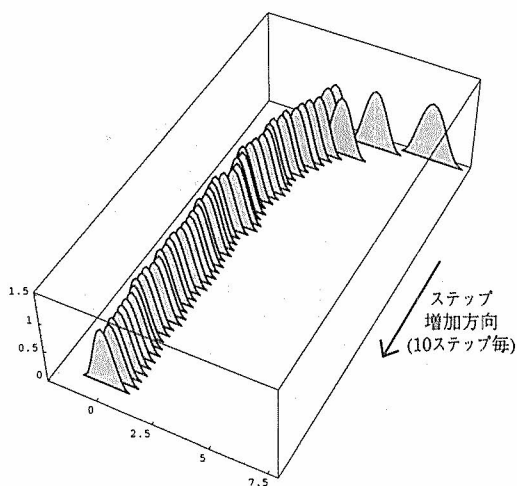


図4. 4 $\hat{M}_0(N)$ のメンバーシップ関数の見本係列

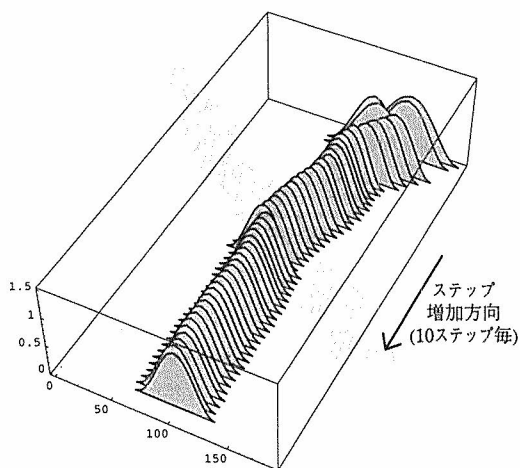


図4. 5 $\hat{\Sigma}_{e,N}^2$ のメンバーシップ関数の見本係列

5. 結 言

本論文では、最も簡単な1階自己回帰モデル内に含まれる未知パラメータの曖昧推定値を求める手法を導出し、その数値実験用プログラムパッケージを開発することを試みた。また、実際にシミュレーション実験を実施することにより、提案した手法の有効性を数値的に検証した。より一般的なシステムモデル内の未知パラメータの推定法およびそのプログラムパッケージについては、現在鋭意研究・開発中であり追って報告する予定である。

参 考 文 献

- [1] 福田得夫. ある種のファジィ確率ベクトルについての一考察. 日本ファジィ学会誌, 10(3): 499-505 (1998)
- [2] L. A. Zadeh. The concept of a linguistic variables and its application in approximate reasoning, parts 1, 2 and 3. *Information Sciences*, 8 and 9: 199-249, 301-357, 43-80, respectively, 1975, 76.
- [3] 福田得夫. ファジィベクトルの二項演算用プログラムパッケージの開発. 追手門経済論集, 29(2): 72-109 (1995)

Appendix 1

Page 1

```

// *****
// estimate.cc: 主プログラム
// *****

#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <iostream.h>
#include "f_num.h"
#include "f_data.h"

char itoa1(int i);
char *itoa3(int i);
void file_operation(char status[40], char datatype[40]);
void results_writing(int time_step,
    FuzzyNumbers &auto0_h, FuzzyNumbers &auto1_h,
    FuzzyNumbers &phi0_h, FuzzyNumbers &phi1_h,
    FuzzyNumbers &sigma_2_h, FuzzyNumbers &m_e_h,
    FuzzyNumbers &mean_h, FuzzyNumbers &Theta_h);

void estimate(char datatype[40],
    const int print_interval, const int Number_of_Sample_Paths,
    const int Number_of_Data, double theta, double m_e,
    double var_e);

char file_number[4];
FILE *file_mean_h, *file_auto0_h, *file_auto1_h,
    *file_phi0_h, *file_phi1_h, *file_data,
    *file_theta_h, *file_m_e_h, *file_sigma_2_h;

// *****
int main() {
// *****
    int itemp, print_interval=100; // per time step
    char datatype[]="ar_1.data";
    int Number_of_Sample_Paths=1,
        Number_of_Data= 10000;
    double theta=0.7, m_e=0.0, var_e=100.0;

    printf("\nNumber of Sample Paths= "); scanf("%d", &Number_of_Sample_Paths);
    printf("\nNumber of Data= "); scanf("%d", &Number_of_Data);
    printf("\nprint interval= "); scanf("%d", &print_interval);

    printf("\nWhat kind of data type do you need\n");
    printf("\nAR(1) --> 1, MA(1) --> 2. ");
    printf("Input the type of data --> "); scanf("%d", &itemp);
    if (itemp==1) {datatype="ar_1.data";} else {datatype="ma_1.data";}
    printf("\ntrue value of theta= "); scanf("%lf", &theta);
    printf("\ntrue value of m_e= "); scanf("%lf", &m_e);
    printf("\ntrue value of var_e= "); scanf("%lf", &var_e);

    estimate(datatype, print_interval, Number_of_Sample_Paths,
        Number_of_Data, theta, m_e, var_e);

    return(0);
}

char *itoa3(int i) {
    static char c[4];
    int itemp;
    itemp= i/100;
    c[0]= itoa1(itemp);
    itemp= i%100;
    c[1]=itoa1(itemp/10);
    c[2]=itoa1(itemp % 10);
    c[3]='\0';
}

```

Appendix 1

Appendix 1

Page 2

```

// printf("c is %c, %c, %c, \n",c[0],c[1],c[2]);
return c;
}

// *****
char itoal(int i){
// *****
char c;
if (i>=0 && i<=9)
switch(i)
{
case 0:      c=0x30; break;
case 1:      c=0x30+1; break;
case 2:      c=0x30+2; break;
case 3:      c=0x30+3; break;
case 4:      c=0x30+4; break;
case 5:      c=0x30+5; break;
case 6:      c=0x30+6; break;
case 7:      c=0x30+7; break;
case 8:      c=0x30+8; break;
case 9:      c=0x30+9; break;
}
else
printf("\n** something wrong in itoal **\n");
return c;
}

// *****
void file_operation(char status[40], char datatype[40]){
// *****
char ctemp[20];
if (status=="open"){
strcpy(ctemp,"mean_h.");
strcat(ctemp, file_number);
file_mean_h=fopen(ctemp, "w");

strcpy(ctemp,"auto0_h.");
strcat(ctemp, file_number);
file_auto0_h=fopen(ctemp, "w");

strcpy(ctemp,"autol_h.");
strcat(ctemp, file_number);
file_autol_h=fopen(ctemp, "w");

strcpy(ctemp,"phi0_h.");
strcat(ctemp, file_number);
file_phi0_h=fopen(ctemp, "w");

strcpy(ctemp,"phil_h.");
strcat(ctemp, file_number);
file_phil_h=fopen(ctemp, "w");

strcpy(ctemp,"theta_h.");
strcat(ctemp, file_number);
file_theta_h=fopen(ctemp, "w");

strcpy(ctemp,"m_e_h.");
strcat(ctemp, file_number);
file_m_e_h=fopen(ctemp, "w");

strcpy(ctemp,"sigma_2_h.");
strcat(ctemp, file_number);
file_sigma_2_h=fopen(ctemp, "w");

file_data=fopen(datatype, "r");
}
}

```

Appendix 1

Appendix 1

Page 3

```

if (status=="close"){
    fclose(file_data);
    fclose(file_mean_h);
    fclose(file_auto0_h);
    fclose(file_autol_h);
    fclose(file_phi0_h);
    fclose(file_phil_h);
    fclose(file_theta_h);
    fclose(file_me_h);
    fclose(file_sigma_2_h);
}
}

// *****
void results_writing(int time_step,
                    FuzzyNumbers &auto0_h,    FuzzyNumbers &autol_h,
                    FuzzyNumbers &phi0_h,    FuzzyNumbers &phil_h,
                    FuzzyNumbers &sigma_2_h, FuzzyNumbers &m_e_h,
                    FuzzyNumbers &mean_h,    FuzzyNumbers &Theta_h){
// *****
    fprintf(file_mean_h,"time_step= %7d\n", time_step);
    mean_h.WriteToFile(file_mean_h);
    fprintf(file_mean_h,"\n");

    fprintf(file_auto0_h,"time_step= %7d\n", time_step);
    auto0_h.WriteToFile(file_auto0_h);
    fprintf(file_auto0_h,"\n");

    fprintf(file_autol_h,"time_step= %7d\n", time_step);
    autol_h.WriteToFile(file_autol_h);
    fprintf(file_autol_h,"\n");

    fprintf(file_phi0_h,"time_step= %7d\n", time_step);
    phi0_h.WriteToFile(file_phi0_h);
    fprintf(file_phi0_h,"\n");

    fprintf(file_phil_h,"time_step= %7d\n", time_step);
    phil_h.WriteToFile(file_phil_h);
    fprintf(file_phil_h,"\n");

    fprintf(file_theta_h,"time_step= %7d\n", time_step);
    Theta_h.WriteToFile(file_theta_h);
    fprintf(file_theta_h,"\n");

    fprintf(file_me_h,"time_step= %7d\n", time_step);
    m_e_h.WriteToFile(file_me_h);
    fprintf(file_me_h,"\n");

    fprintf(file_sigma_2_h,"time_step= %7d\n", time_step);
    sigma_2_h.WriteToFile(file_sigma_2_h);
    fprintf(file_sigma_2_h,"\n");

    printf("time_step= %7d\n", time_step);
}

// *****
void estimate(char datatype[40],
             const int print_interval,
             const int Number_of_Sample_Pathes,
             const int Number_of_Data, double theta,
             double m_e, double var_e){
// *****
    char          c, ctemp[20];
    int           time_step, i;
    // double      xmin, xmax, tmin, tmax;
    Intervals     Itemp;

```

Appendix 1

Appendix 1

Page 4

```

FuzzyNumbers    auto0_h, autol_h, phi0_h, phil_h, data1, data2,
                  sigma_2_h, m_e_h, templ, temp2, mean_h, Theta_h;
for(i=1;i<=Number_of_Sample_Pathes;i++){
    strcpy(file_number, itoa3(i));
    fuzzy_data(Number_of_Data+1, theta, m_e, var_e, datatype);
    file_operation("open", datatype);
    mean_h.PutCrispValue(0.0);
    auto0_h.PutCrispValue(0.0);
    autol_h.PutCrispValue(0.0);
    fscanf(file_data, "%s %d", ctemp, &time_step);
    data1.ReadFromFile(file_data);
    c=fgetc(file_data);
    while(!feof(file_data)!=1){
        mean_h= (1.0-1.0/time_step)*mean_h+(1.0/time_step)*data1;
        //      auto0_h= (1.0-1.0/time_step)*auto0_h+(1.0/time_step)*data1*data1;
        auto0_h= (1.0-1.0/time_step)*auto0_h+(1.0/time_step)*data1.Square();
        fscanf(file_data, "%s %d", ctemp, &time_step);
        data2.ReadFromFile(file_data);
        c=fgetc(file_data);
        autol_h= (1.0-1.0/time_step)*autol_h+(1.0/time_step)*data1*data2;
        data1= data2;
        if((time_step % print_interval) == 0) {
            //      templ = mean_h*mean_h;
            templ = mean_h.Square();
            phi0_h= auto0_h-templ;
            phil_h= autol_h-templ;
            if (strcmp(datatype, "ar_1.data", 9) == 0) {
                Theta_h= -phil_h/phi0_h;
                sigma_2_h= phi0_h+Theta_h*phil_h;
                m_e_h=(1+Theta_h)*mean_h;
            }
            else {
                printf("\nnot available at this time \n\n");
            }
            results_writing(time_step, auto0_h, autol_h, phi0_h, phil_h,
                           sigma_2_h, m_e_h, mean_h, Theta_h);
        }
    }
    file_operation("close", datatype);
}
}

// *****
//      f_data.h: ヘッダーファイル
// *****

#define PI          3.14159

void fuzzy_data(int Number_of_Data, double theta,
                double m, double variance, char file_name[40]);
double normal_random_numbers(double m, double var);
FuzzyNumbers fuzzy_data_generation(double& crisp_data);

// *****
//      f_data.cc: プログラムリスト
// *****

// *****
//      Generation of fuzzy random data
//      Original process is the output process of a first order
//      autoregressive model or moving average model.
// *****

#include <stdlib.h>
#include <iostream.h>
#include <math.h>

```

Appendix 1

Appendix 1

Page 5

```
#include <string.h>
#include "f_num.h"
#include "f_data.h"

// *****
void fuzzy_data(int Number_of_Data, double theta,
                double m, double variance, char file_name[40]){
// *****
    int i;
    double x, e_k, e_k_l;
    FuzzyNumbers data;
    FILE *ofile;

    ofile= fopen(file_name,"w");
    if (strncmp(file_name,"ar_1.data",9)==0) {
        x=0.0;
        for(i=1;i<=Number_of_Data;i++){
            e_k= normal_random_numbers(m,variance);
            x= -theta*x + e_k;
            data=fuzzy_data_generation(x);
            fprintf(ofile,"time_step= %7d\n", i);
            data.WriteToFile(ofile);
            fprintf(ofile,"\n");
        }
    }
    else {
        for(i=1;i<=Number_of_Data;i++){
            e_k= normal_random_numbers(m,variance);
            x= e_k + theta*e_k_l;
            data=fuzzy_data_generation(x);
            fprintf(ofile,"time_step= %7d\n", i);
            data.WriteToFile(ofile);
            fprintf(ofile,"\n");
            e_k_l= e_k;
        }
        fclose(ofile);
    }
}

// *****
double normal_random_numbers(double m, double variance){
// *****
    int i;
    double normal= 0.0;
    for(i=1;i<=12;i++) normal= normal + rand()/(double) RAND_MAX;
    normal= normal-6.0;
    normal= sqrt(variance)*normal+m;
    return normal;
}

// *****
FuzzyNumbers fuzzy_data_generation(double &crisp_data){
// *****
    const double a= 1.0, // center of membership functions
                b= 0.6; // half width of membership function/2
    int i, position;
    FuzzyNumbers Result;
    i=-10000;
    while(crisp_data > i*a+b) i++;
    if ((crisp_data >= (i+1)*a-b) && (rand()/(double) RAND_MAX >= 0.5))
        position= i+1;
    else
        position= i;
    Result.PutFuzzyNumber(position*a, b);
    return Result;
}
```

Appendix 1

Appendix 2

Page 1

```

In[1]:=<<Graphics`Graphics3D`
In[2]:=templ=(Word, Number, Table[(Number, Number, Number), {i, 0, 20}]);
In[3]:=data=ReadList["ar_1.data", templ];
graphdata={};
NumberOfLevels=Length[data[{1}][{3}]];
Do[
  temp2=Table[
    {data[{1}][{3}][{j}][{2}],
     data[{1}][{3}][{j}][{1}]}, {j, 1, NumberOfLevels-1}];
  temp3=Table[
    {data[{1}][{3}][{NumberOfLevels+1-j}][{3}],
     data[{1}][{3}][{NumberOfLevels+1-j}][{1}]},
    {j, 1, NumberOfLevels}];
  temp4=Join[temp2, temp3];
  g=Graphics[{GrayLevel[0.7], Polygon[temp4]}];
  AppendTo[graphdata, g]; {i, 1, 15};
  Show[StackGraphics[Reverse[graphdata]],
    BoxRatios->{10, 20, 2}, PlotRange->{0, 1.5},
    ViewPoint->{1.104, -2.139, 2.379}, Axes->{True, False, True},
    Boxed->True];
In[4]:=files={"theta_h.001", "m_e_h.001", "sigma_2_h.001", "auto0_h.001",
  "autol_h.001", "phi0_h.001", "phi1_h.001", "mean_h.001"}
In[6]:=Do[readfile=files[{k}];
  data=ReadList[readfile, templ];
  graphdata={}; NumberOfData=Length[data];
  NumberOfLevels=Length[data[{1}][{3}]];
  Do[
    temp2=Table[
      {data[{1}][{3}][{j}][{2}],
       data[{1}][{3}][{j}][{1}]}, {j, 1, NumberOfLevels-1}];
    temp3=Table[
      {data[{1}][{3}][{NumberOfLevels+1-j}][{3}],
       data[{1}][{3}][{NumberOfLevels+1-j}][{1}]},
      {j, 1, NumberOfLevels}];
    temp4=Join[temp2, temp3];
    g=Graphics[{GrayLevel[0.7], Polygon[temp4]}];
    AppendTo[graphdata, g]; {i, 5, NumberOfData};
    Show[StackGraphics[Reverse[graphdata]],
      BoxRatios->{5, 15, 2}, PlotRange->All,
      ViewPoint->{1.104, -2.139, 2.379}, Axes->{True, False, True},
      PlotRange->{0, 1.5},
      Boxed->True, PlotLabel->FontForm[readfile, {"Times", 18}];
    Show[graphdata[{NumberOfData-5}], Axes->True];
    , {k, 1, 8}]

```

Appendix 2