確率分布の事前情報を必要としない粒子フィルタ*

金田 泰昌[†]・入月 康晴[†]Particle Filter without a Prior Information
for Probability Distribution*Yasuaki KANEDA[†] and Yasuharu IRIZUKI[†]

Kalman filter is one of the most famous state estimation methods. Many state estimations based on Kalman filter have been proposed for various conditions of systems and noise. These estimation methods need to know probability density functions of dynamical systems. However, the probability density functions and a prior information of noise in particular are rarely known in practice. On the other hand, in the field of machine learning, probability density estimations have been studied extensively, and conditional probability density estimations were proposed as extensions of the probability density estimations. In this paper, we propose a direct design method of probability density functions for dynamical systems from data by using conditional probability density estimations because the systems are represented as conditional probability density functions. In addition, we apply the method to particle filter, which is one of nonlinear Kalman filters, and propose a new state estimation method without a prior knowledge for the dynamical systems. Numerical simulations demonstrate the effectiveness of the proposed method.

1. はじめに

状態推定は制御工学において古くから研究されている分野の一つであり、現在も精力的に研究が進められている。近年では「ソフトセンサ」に代表されるように、物理量を直接計測するのではなく、別の観測結果からの推定値を用いて間接的に計測する方法が注目されているが、状態推定はこの実現に欠かせない方法である。このように、理論面はもとより、産業応用の側面からも盛んに研究が進められている。

最も有名な状態推定法としてカルマンフィルタ (KF: Kalman Filter) [1] がある。KF は線形ガウス型システムを対象としており、状態の予測分布およびフィルタ分布が解析的に求まることが知られている。非線形システムに対しては KF を拡張した拡張カルマンフィルタ (EKF: Extended KF) やヤコビアン計算を必要としない無香

料カルマンフィルタ (UKF: Unscented KF) [2] などが、また非ガウス系に対する方法として粒子フィルタ (PF: Particle Filter) [3] などが提案されている。PF は任意のノイズ分布に対して適用可能である反面、モンテカルロ計算により確率密度関数を近似するため計算コストが高くなる。そのため、特定の非ガウスノイズに対して計算コストが低い方法もまた提案されている。たとえば、参考文献 [4–6] では、外れ値ノイズとよばれる非ガウスノイズに対する状態推定法としてロバストカルマンフィルタが提案されている。

このように、システムやノイズの条件に応じて KF 型の状態推定法が数多く提案されているが、これらはすべてシステムの確率密度関数を既知として状態の予測分布およびフィルタ分布を Chapman-Kolmogorov-Equation (CKE) やベイズの定理を用いて逐次的に解く方法と解釈できる [7]。しかしながら、現実的にはシステムの確率密度関数の形状やそのパラメータが既知であることはまれである。とくに、通常システムの平均的なふるまいとして状態方程式や観測方程式が用いられ、これらの方程式は第一原理などから得られるが、状態ノイズや観測ノイズを事前に特定することは困難な作業である。この問題に対して、さまざまな仮定のもとでノイズ分布をパラ

* 原稿受付 2018 年 6 月 29 日
第 62 回システム制御情報学会研究発表講演会にて発表 (2018 年 5 月)

[†] 東京都立産業技術研究センター Tokyo Metropolitan Industrial Technology Research Institute; 2-4-10, Aomi, Koutou-ku, Tokyo 135-0064, JAPAN

Key Words: Kalman filter, particle filter, density ratio estimation, data-driven estimation.

メトリックにモデル化し、そのノイズ分布のパラメータをデータから事前に推定する方法や、ノイズ分布のパラメータを状態と同時に推定する方法が提案されている。たとえば、参考文献 [8] ではノイズ分布をガウス分布と仮定し、そのパラメータを変分ベイズ法で推定している。しかしながら、この方法は特定の分布形状に対してパラメータを推定しており、分布形状は既知としている。また、ガウス和フィルタとよばれる任意の分布を扱うことのできる方法に対して、そのパラメータを適応的に決定する方法 [9] が提案されているが、観測ノイズは非ガウス分布を仮定しているものの状態ノイズは既知のガウス分布を仮定している。

状態推定精度がシステムのモデル化誤差に大きく依存する問題に対して Direct Filter (DF) [10] が提案されている。DF はあらかじめ得られたデータから状態推定器をノンパラメトリックに直接設計する方法である。未知システムに対して仮定をおくことなくデータのみから最適フィルタが直接設計できる反面、モデル設計を介さないためモデルを介してシステムを理解することはできない。

機械学習の分野ではデータから確率密度関数を推定する方法が提案されている。たとえば、密度比の直接推定法 [11] が提案されており、その応用として条件付き確率密度関数の直接推定法 [12] が提案されている。状態および観測モデルの確率密度関数は条件付き確率密度関数として与えられる。そこで、条件付き確率密度推定法を用いることでデータから直接システムの確率密度関数を推定できるはずである。そして、この推定した確率密度関数を用いることで、システムの確率密度関数の事前情報が未知の場合でも予測分布およびフィルタ分布が計算できると期待される。また、システムの確率密度関数を推定することで、DF では困難であるモデルを介したシステムの理解も期待される。実際、参考文献 [13] では、データから観測モデルの確率密度関数を直接推定し、事前情報を必要としないフィルタ分布を提案している。

本論文では、状態および観測モデルに関する確率密度関数の事前情報を必要としない状態推定法の開発を目的とする。そのために、機械学習の方法を応用し、データからシステムの確率密度関数をノンパラメトリックに直接設計する。そして、得られた確率密度関数を PF に適用することで、確率分布の事前情報を必要としない状態推定法を提案する。

本論文の構成はつぎの通りである。2 節では問題設定を記述する。3 節ではデータのみから条件付き確率密度関数をノンパラメトリックに直接推定する方法を説明し、4 節ではこの条件付き確率密度推定法を用いた状態推定法を提案する。5 節では数値シミュレーションにて提案法の有効性を示し、6 節で本論文をまとめる。

2. 問題設定

2.1 対象システム

次式の離散時間非線形動的システムに対する状態推定問題を考える。

$$\begin{aligned} \mathbf{x}_{k+1} &\sim p(\mathbf{x}_{k+1}|\mathbf{x}_k) \\ \mathbf{y}_k &\sim p(\mathbf{y}_k|\mathbf{x}_k) \end{aligned}$$

ここで、 $\mathbf{x}_k \in \mathbb{R}^n$ は時刻 k における状態、 $\mathbf{y}_k \in \mathbb{R}^m$ は観測値を表す。また、 $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ は状態遷移に関する確率密度関数で、状態モデルを確率表現したものである。同じく、 $p(\mathbf{y}_k|\mathbf{x}_k)$ は観測モデルを確率表現したものである。本論文では、状態推定器を構築する際、これらの確率密度関数の情報は未知とする。

【例題 1】 たとえば、線形ガウス型システムの状態方程式は次式で与えられる。

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \end{aligned}$$

ここで、 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 、 $\mathbf{C} \in \mathbb{R}^{m \times n}$ である。また、 $\mathbf{w}_k \in \mathbb{R}^n$ は平均 $\mathbf{0}$ 、共分散行列 $\mathbf{Q} \in \mathbb{R}^{n \times n}$ の状態ガウスノイズであり、 $\mathbf{v}_k \in \mathbb{R}^m$ は平均 $\mathbf{0}$ 、共分散行列 $\mathbf{R} \in \mathbb{R}^{m \times m}$ の観測ガウスノイズである。 $\mathbf{w}_k$ と $\mathbf{v}_k$ は無相関とする。このとき、線形ガウス型システムの確率表現は次式で与えられる。

$$\begin{aligned} p(\mathbf{x}_{k+1}|\mathbf{x}_k) &= \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{A}\mathbf{x}_k, \mathbf{Q}) \\ p(\mathbf{y}_k|\mathbf{x}_k) &= \mathcal{N}(\mathbf{y}_k; \mathbf{C}\mathbf{x}_k, \mathbf{R}) \end{aligned}$$

ただし、

$$\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \Sigma) = \frac{1}{(\sqrt{2\pi})^n \sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}\|\mathbf{x} - \bar{\mathbf{x}}\|_{\Sigma^{-1}}^2\right)$$

であり、 $\det(\Sigma)$ は行列 $\Sigma \in \mathbb{R}^{n \times n}$ の行列式を表す。また、 $\|\mathbf{x} - \bar{\mathbf{x}}\|_{\Sigma^{-1}}^2 = (\mathbf{x} - \bar{\mathbf{x}})^T \Sigma^{-1} (\mathbf{x} - \bar{\mathbf{x}})$ である。

(注意 1) システムに入力 $\mathbf{u}_k \in \mathbb{R}^l$ が存在する場合、システムの確率表現は $\mathbf{u}_k$ が与えられたときの確率密度関数として記述できる。たとえば、入力が存在する場合の状態モデルの確率表現は $p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{u}_k)$ と記述できる。以降、簡単のため入力 $\mathbf{u}_k$ を省略して議論するが、 $\mathbf{u}_k$ が存在しても同様の議論ができることを 4.1 項で説明する。

2.2 カルマンフィルタと条件付き確率密度関数

KF 型の状態推定法は時間更新および観測更新の二つのステップより構成されており、それぞれ予測分布およびフィルタ分布とよばれる条件付き確率密度関数をサンプリング周期ごとに逐次的に求めている [7]。いま、 $\mathbf{Y}_k = \{\mathbf{y}_1, \dots, \mathbf{y}_k\}$ とすると、予測分布およびフィルタ分布はそれぞれ次式で与えられる。

$$p(\mathbf{x}_{k+1}|\mathbf{Y}_k) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Y}_k) d\mathbf{x}_k, \quad (1)$$

$$p(\mathbf{x}_k | \mathbf{Y}_k) = \frac{p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Y}_{k-1})}{p(\mathbf{y}_k | \mathbf{Y}_{k-1})} \quad (2)$$

予測分布は1ステップ先の状態遷移計算に利用され、時刻 k から $k+1$ に遷移した状態を $\mathbf{x}_{k+1|k}$ と記述する。また、フィルタ分布は観測値による状態の更新計算に利用され、KF では予測分布で得られた状態 $\mathbf{x}_{k|k-1}$ を観測値 $\mathbf{y}_k$ で更新する際に利用される。時刻 k における更新後の状態を $\mathbf{x}_{k|k}$ と記述する。

【例題 2】 たとえば線形ガウス型システムの場合、初期分布としてガウス分布を仮定することで (1) 式および (2) 式は最終的に次式のように正規分布として与えられる。

$$\begin{aligned} p(\mathbf{x}_{k+1} | \mathbf{Y}_k) &= \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{x}_{k+1|k}, \mathbf{P}_{k+1|k}) \\ p(\mathbf{x}_k | \mathbf{Y}_k) &= \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{k|k}, \mathbf{P}_{k|k}) \end{aligned}$$

ただし、

$$\begin{cases} \mathbf{x}_{k+1|k} = \mathbf{A} \mathbf{x}_{k|k} \\ \mathbf{P}_{k+1|k} = \mathbf{A} \mathbf{P}_{k|k} \mathbf{A}^T + \mathbf{Q} \end{cases}$$

また、

$$\begin{cases} \mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}(\mathbf{y}_k - \mathbf{C} \mathbf{x}_{k|k-1}) \\ \mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K} \mathbf{C}) \mathbf{P}_{k|k-1}, \\ \mathbf{K} = \mathbf{P}_{k|k-1} \mathbf{C}^T (\mathbf{C} \mathbf{P}_{k|k-1} \mathbf{C}^T + \mathbf{R})^{-1} \end{cases}$$

ここで $\mathbf{I}$ は単位行列を表す。つまり、線形 KF は予測分布およびフィルタ分布の平均値および共分散行列を逐次的かつ解析的に計算するアルゴリズムとなっている。

(1) 式および (2) 式を計算するには、条件付き確率密度関数 $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$ および $p(\mathbf{x}_k | \mathbf{y}_k)$ が必要となる。すなわち、確率表現された状態モデルおよび観測モデルが必要となる。問題設定よりこれらは未知である。そこで、本論文ではあらかじめ取得した学習データから確率密度関数を直接推定することを考え、次の条件を仮定する。

(仮定 1) 学習時のみ状態値を取得できる。

この仮定が成り立つ状況の例として、技術的には状態が観測できるハードセンサが存在するものの、高価であったり故障しやすいという理由から、上市する際にはハードセンサを搭載しないという状況が挙げられる。また、オフラインでは状態を観測できるがオンラインでは観測できない状況なども挙げられる。実際、参考文献 [10] では車両のヨーレート推定問題において、仮定 1 のもとで DF が設計されている。

本論文では条件付き確率密度推定法を用いてデータから条件付き確率密度関数を直接推定する。3 節では二種類の条件付き確率密度推定法を説明し、4 節で条件付き確率密度推定法を応用した状態推定法を説明する。

3. データを用いた条件付き確率密度関数の直接推定

3.1 最小二乗法に基づく条件付き確率密度推定

本項では参考文献 [12] の方法を説明する。いま、次式の条件付き確率密度関数を考える。

$$p(\mathbf{t} | \mathbf{s}) = \frac{p(\mathbf{s}, \mathbf{t})}{p(\mathbf{s})} \quad (3)$$

ただし、 $\mathbf{s} \in \mathbb{R}^{n_s}$, $\mathbf{t} \in \mathbb{R}^{n_t}$ とする。いま、(3) 式が次式で近似されるとする。

$$\hat{p}(\mathbf{t} | \mathbf{s}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{s}, \mathbf{t}) \quad (4)$$

ここで、 $\boldsymbol{\phi}(\mathbf{s}, \mathbf{t}) \in \mathbb{R}^N$ は N 個のカーネル関数からなる基底関数ベクトルで、 $\boldsymbol{\theta} \in \mathbb{R}^N$ はその係数ベクトルである。このとき、次式の評価関数の最小化問題を解くことで近似分布 (4) 式の係数を決定する。

$$J_0 = \frac{1}{2} \int \int (p(\mathbf{t} | \mathbf{s}) - \hat{p}(\mathbf{t} | \mathbf{s}))^2 p(\mathbf{s}) d\mathbf{s} d\mathbf{t} \quad (5)$$

いま、(3) 式の関係を利用して (5) 式を展開する。この際、定数項を無視し、また期待値計算を標本平均で置き換えると、(5) 式から次式の評価関数が得られる。

$$J = \frac{1}{2} \boldsymbol{\theta}^T \bar{\mathbf{H}} \boldsymbol{\theta} - \boldsymbol{\theta}^T \bar{\mathbf{h}} \quad (6)$$

ここで、

$$\begin{aligned} \bar{\mathbf{H}} &= \frac{1}{N} \sum_{i=1}^N \boldsymbol{\Phi}_i(\mathbf{s}_i) \\ \bar{\mathbf{h}} &= \frac{1}{N} \sum_{i=1}^N \boldsymbol{\phi}(\mathbf{s}_i, \mathbf{t}_i) \\ \boldsymbol{\Phi}_i(\mathbf{s}_i) &= \int \boldsymbol{\phi}(\mathbf{s}_i, \mathbf{t}_i) \boldsymbol{\phi}^T(\mathbf{s}_i, \mathbf{t}_i) d\mathbf{t}_i \end{aligned} \quad (7)$$

参考文献 [12] では、計算の簡易さを重視し、評価関数 (6) 式に対して L_2 正則化付き最小二乗法を解いた後、次式に従い得られた係数 $\boldsymbol{\theta}$ のうち負になる要素を打ち切ることで確率密度関数の非負性を確保している。

$$\hat{\theta}_i = \max(0, \theta_i), \quad i = 1, \dots, N$$

また、確率密度関数は $\int p(\mathbf{t} | \mathbf{s}) d\mathbf{t} = 1$ を満たす必要がある。そこで、次式のように、最終的には正規化された値を条件付き確率密度関数の推定値として採用する。

$$\hat{p}(\mathbf{t} | \mathbf{s}) = \frac{\hat{\boldsymbol{\theta}}^T \boldsymbol{\phi}(\mathbf{s}, \mathbf{t})}{\eta(\mathbf{s})} \quad (8)$$

$$\eta(\mathbf{s}) = \int \hat{\boldsymbol{\theta}}^T \boldsymbol{\phi}(\mathbf{s}, \mathbf{t}) d\mathbf{t} \quad (9)$$

さらに、基底関数として次式を定義する。

$$\begin{cases} \phi(s, t) = [\phi_1(s, t) \cdots \phi_N(s, t)]^T \\ \phi_i(s, t) = K_{\sigma_s}(s, s^{(i)}) K_{\sigma_t}(t, t^{(i)}) \end{cases} \quad (10)$$

ここで、 $K_{\sigma}(s, s^{(i)})$ はカーネル中心が $s^{(i)}$ 、バンド幅 σ のガウスカーネルで、次式で与えられる。

$$K_{\sigma}(s, s^{(i)}) = \exp\left(-\frac{\|s - s^{(i)}\|^2}{2\sigma^2}\right)$$

このとき (7) 式および (9) 式は解析的に計算できることが示されている。なお、(8) 式の推定精度は正規化パラメータやバンド幅に依存するが、これらは交差検証により決定する。

3.2 KDE を用いた条件付き確率密度推定

3.2.1 Kernel Density Estimation(KDE)

KDE は確率密度関数を推定するためのノンパラメトリック法の一つである。いま、標本 $s^{(i)}$, $i=1, \dots, N$ が与えられ、かつガウスカーネルを用いた場合、KDE による確率密度関数 $p(s)$ は次式で推定される。

$$\hat{p}(s) = \frac{1}{N(\sqrt{2\pi}\sigma)^{n_s}} \sum_{i=1}^N K_{\sigma}(s, s^{(i)})$$

同時分布も同様に考えることができ、標本 $(s^{(i)}, t^{(i)})$ が与えられたときの同時分布 $p(s, t)$ は次式で推定される。

$$\hat{p}(s, t) = \frac{1}{N(\sqrt{2\pi}\sigma_s)^{n_s}(\sqrt{2\pi}\sigma_t)^{n_t}} \sum_{i=1}^N \phi_i(s, t) \quad (11)$$

ここで、 $\phi_i(s, t)$ は (10) 式で与えられる。

3.2.2 KDE を用いたプロパーな条件付き確率密度推定

(3) 式の通り条件付き確率密度関数は密度比で表される。そのため、従来 KDE を用いた条件付き確率密度推定では、 $p(s, t)$ と $p(s)$ をそれぞれ KDE で推定しその比を計算していた。しかし、この方法は推定誤差が増幅され数値的に不安定（インプロパー）な推定値となることが知られている [12]。この問題を解決し、数値的に安定（プロパー）な結果を得られるようにしたのが密度比を直接推定する方法であり、3.1 項の方法である。一方、 $p(s)$ は同時分布 $p(s, t)$ の t に関する周辺分布からも得られる。通常この周辺化の計算は困難であるが、KDE で同時分布を推定した場合、基底関数を (10) 式で与えることで周辺化が解析的に計算できる。そして、このとき得られる条件付き確率密度関数は 3.1 項の結果と同じ構造を持つことを以下で示す。

いま、同時分布が (11) 式で推定されたとき、周辺分布 $\hat{p}(s)$ はガウス積分を用いることで次式として与えられる。

$$\hat{p}(s) = \int \hat{p}(s, t) dt$$

$$= \frac{1}{N(\sqrt{2\pi}\sigma_s)^{n_s}} \sum_{i=1}^N K_{\sigma_s}(s, s^{(i)}) \quad (12)$$

よって、条件付き確率密度関数 $p(t|s)$ の推定値は (3) 式、(11) 式、(12) 式より次式で与えられる。

$$\hat{p}(t|s) = \frac{\sum_{i=1}^N \phi_i(s, t)}{(\sqrt{2\pi}\sigma_t)^{n_t} \sum_{i=1}^N K_{\sigma_s}(s, s^{(i)})} \quad (13)$$

(注意 2) 基底関数が (10) 式で与えられたとき、(8) 式より次式が得られる。

$$\hat{p}(t|s) = \frac{\sum_{i=1}^N \theta_i \phi_i(s, t)}{(\sqrt{2\pi}\sigma_t)^{n_t} \sum_{i=1}^N \theta_i K_{\sigma_s}(s, s^{(i)})}$$

よって、基底関数が (10) 式で与えられたとき、KDE により得られた条件付き確率密度関数 (13) 式は、参考文献 [12] により得られた条件付き確率密度関数 (8) 式の係数を $1/N$ にしたものと解釈できる。つまり、基底関数が (10) 式で与えられたとき、(8) 式と (13) 式はガウスカーネルの加重平均で近似するのかわかるとも算術平均で近似するのかわかるとなり、ともにガウスカーネルの平均値の形で表現される。すなわち、平均値計算という意味で同じであり、数値計算上の安定性の特性は同じになることが期待される。

(注意 3) 条件付き確率密度推定として参考文献 [12] を用いた場合、カーネル関数の重みベクトルがスパースベクトルとして求まるため必要なカーネル関数の数が少なくなる可能性がある。一方、KDE を用いた場合、カーネル関数の数は多くなるが、学習時の最適化プロセスがなく学習が高速であるというメリットがある。

(注意 4) 3.1 節同様、カーネル関数のバンド幅等は交差検証などにて決定する必要がある。

4. 条件付き確率密度推定法を用いた粒子フィルタ

4.1 データによる状態モデルの直接推定と予測分布推定

状態モデルの確率密度関数の推定値 $\hat{p}(x_{k+1}|x_k)$ は前節の結果において s を x_k で、また t を x_{k+1} で置き換えればよい。いま、データセット $(x_0^{(i)}, x_1^{(i)})$, $i=1, \dots, N$ が与えられたとする。ここで、 $x_0^{(i)}$ および $x_1^{(i)}$ との間には $x_1^{(i)} \sim p(x_{k+1}|x_k = x_0^{(i)})$ の関係が成り立つ。つまり、各変数の下添え字は相対離散時刻を意味することに注意する。このとき、たとえば KDE を用いて確率密度関数を推定する場合、状態モデルの確率密度関数は次式で推定される。

$$\hat{p}(x_{k+1}|x_k) = \frac{\sum_{i=1}^N \phi_i(x_k, x_{k+1})}{(\sqrt{2\pi}\sigma_1)^{n_1} \sum_{i=1}^N K_{\sigma_0}(x_k, x_0^{(i)})} \quad (14)$$

ここで、

$$\phi_i(x_k, x_{k+1}) = K_{\sigma_1}(x_{k+1}, x_1^{(i)}) K_{\sigma_0}(x_k, x_0^{(i)})$$

(14) 式を用いて予測分布 (1) 式を計算する。本論文では、PF と同様に M 個の粒子 $\mathbf{x}_{k|k}^{(i)}$ で事前分布を近似する。つまり、事前分布が次式で与えられるとする。

$$p(\mathbf{x}_k|\mathbf{Y}_k) = \frac{1}{M} \sum_{i=1}^M \delta(\mathbf{x}_k - \mathbf{x}_{k|k}^{(i)})$$

ここで、 $\delta(\mathbf{x})$ はディラックのデルタ関数である。このとき、予測分布 (1) 式は次式で表される。

$$\begin{aligned} p(\mathbf{x}_{k+1}|\mathbf{Y}_k) &= \frac{1}{M} \sum_{i=1}^M \hat{p}(\mathbf{x}_{k+1}|\mathbf{x}_k = \mathbf{x}_{k|k}^{(i)}) \\ &= \frac{1}{M} \sum_{i=1}^M \frac{\sum_{j=1}^N \phi_j(\mathbf{x}_{k|k}^{(i)}, \mathbf{x}_{k+1})}{(\sqrt{2\pi}\sigma_1)^n \sum_{l=1}^N K_{\sigma_0}(\mathbf{x}_{k|k}^{(i)}, \mathbf{x}_0^{(l)})} \\ &= \sum_{j=1}^N \alpha_j \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{x}_1^{(j)}, \sigma_1^2 \mathbf{I}) \end{aligned} \quad (15)$$

ここで、

$$\alpha_j = \frac{1}{M} \sum_{i=1}^M \frac{K_{\sigma_0}(\mathbf{x}_{k|k}^{(i)}, \mathbf{x}_0^{(j)})}{\sum_{l=1}^N K_{\sigma_0}(\mathbf{x}_{k|k}^{(i)}, \mathbf{x}_0^{(l)})}$$

いま、 $\sum_{j=1}^N \alpha_j = 1$ であるため、 α_j は確率を意味している。つまり、 j 番目の要素 $\mathcal{N}(\mathbf{x}_{k+1}; \mathbf{x}_1^{(j)}, \sigma_1^2 \mathbf{I})$ が選択される確率が α_j であり、選択された正規分布からサンプリングすることで予測分布のサンプル $\mathbf{x}_{k+1|k}^{(i)}$ が取得できる。 M 個の粒子で予測分布を近似するためには M 回このサンプリングを繰り返せばよい。

(注意 5) (15) 式より、提案法は予測分布を粒子で離散近似するのではなく混合正規分布で近似している。そして、その要素数は粒子数 M ではなくカーネル数 N で決定される。つまり、粒子数が少なくても、カーネル数が多ければ、より広範囲かつ滑らかな予測分布の近似が期待される。

(注意 6) 入力 $\mathbf{u}_k$ が存在する場合、確率密度関数の条件に $\mathbf{u}_k$ を追加し、 $\mathbf{u}_k$ に関するカーネル関数を追加することで同様の議論が展開できる。すなわち、 $\mathbf{u}_k$ に関するカーネル関数を $K_{\sigma_u}(\mathbf{u}_k, \mathbf{u}^{(i)})$ とし、 σ_u はバンド幅とする。簡単のため $\kappa_i(\mathbf{u}_k) := K_{\sigma_u}(\mathbf{u}_k, \mathbf{u}^{(i)})$ とすると、入力が存在する場合の状態モデルの確率密度関数は次式で推定される。

$$\begin{aligned} \hat{p}(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{u}_k) &= \frac{\sum_{i=1}^N \kappa_i(\mathbf{u}_k) \phi_i(\mathbf{x}_k, \mathbf{x}_{k+1})}{(\sqrt{2\pi}\sigma_1)^n \sum_{i=1}^N \kappa_i(\mathbf{u}_k) K_{\sigma_0}(\mathbf{x}_k, \mathbf{x}_0^{(i)})} \end{aligned} \quad (16)$$

つまり、(14) 式に対して (16) 式は入力に依存した重み $\kappa_i(\mathbf{u}_k)$ が考慮された式となっているだけであり、それ以外は入力が存在しない場合と同じ議論展開が可能となる。最終的に、入力が存在する場合の予測分布は次式で与えられる。

$$p(\mathbf{x}_{k+1}|\mathbf{Y}_k, \mathbf{u}_k) = \sum_{j=1}^N \alpha_j(\mathbf{u}_k) \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{x}_1^{(j)}, \sigma_1^2 \mathbf{I})$$

ただし、 $\alpha_j(\mathbf{u}_k)$ は次式で与えられる。

$$\alpha_j(\mathbf{u}_k) = \frac{1}{M} \sum_{i=1}^M \frac{\kappa_j(\mathbf{u}_k) K_{\sigma_0}(\mathbf{x}_{k|k}^{(i)}, \mathbf{x}_0^{(j)})}{\sum_{l=1}^N \kappa_l(\mathbf{u}_k) K_{\sigma_0}(\mathbf{x}_{k|k}^{(i)}, \mathbf{x}_0^{(l)})}$$

4.2 データによる観測モデルの直接推定とフィルタ分布推定

観測モデルの確率密度関数の推定値 $\hat{p}(\mathbf{y}_k|\mathbf{x}_k)$ は前節の結果において $\mathbf{s}$ を $\mathbf{x}_k$ で、 $\mathbf{t}$ を $\mathbf{y}_k$ で置き換えればよい。いま、データセット $(\mathbf{x}_0^{(i)}, \mathbf{y}_0^{(i)})$, $i=1, \dots, N$ が与えられたとする。ここで、 $\mathbf{x}_0^{(i)}$ と $\mathbf{y}_0^{(i)}$ との間には $\mathbf{y}_0^{(i)} \sim p(\mathbf{y}_k|\mathbf{x}_k = \mathbf{x}_0^{(i)})$ の関係が成り立つ。前項同様、ここでも各変数の下添え字は相対離散時刻を意味することに注意する。このとき、たとえば KDE を用いて確率密度関数を推定した場合、観測モデルの確率密度関数は次式で推定される。

$$\hat{p}(\mathbf{y}_k|\mathbf{x}_k) = \frac{\sum_{i=1}^N \phi_i(\mathbf{x}_k, \mathbf{y}_k)}{(\sqrt{2\pi}\sigma_y)^m \sum_{i=1}^N K_{\sigma_x}(\mathbf{x}_k, \mathbf{x}_0^{(i)})} \quad (17)$$

ここで、

$$\phi_i(\mathbf{y}_k, \mathbf{x}_k) = K_{\sigma_x}(\mathbf{x}_k, \mathbf{x}_0^{(i)}) K_{\sigma_y}(\mathbf{y}_k, \mathbf{y}_0^{(i)})$$

フィルタ分布 (2) 式は次式で書き直すことができる。

$$p(\mathbf{x}_k|\mathbf{Y}_k) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Y}_{k-1})}{\int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Y}_{k-1}) d\mathbf{x}_k}$$

よって、事前分布として前項で得られた予測分布と (17) 式を用いることでフィルタ分布をデータから直接計算できる。ここでは前項同様、複数の粒子で事前分布を近似する。いま、事前分布が次式で与えられたとする。

$$p(\mathbf{x}_k|\mathbf{Y}_{k-1}) = \frac{1}{M} \sum_{i=1}^M \delta(\mathbf{x}_k - \mathbf{x}_{k|k-1}^{(i)})$$

ここで、 $\mathbf{x}_{k|k-1}^{(i)}$ は予測分布 (15) 式よりサンプリングされた粒子である。このとき、各パーティクル $\mathbf{x}_{k|k-1}^{(i)}$ に対して次式が成り立つ。

$$p(\mathbf{x}_k = \mathbf{x}_{k|k-1}^{(i)}|\mathbf{Y}_k) = \frac{\hat{p}(\mathbf{y}_k|\mathbf{x}_k = \mathbf{x}_{k|k-1}^{(i)})}{\sum_{i=1}^M \hat{p}(\mathbf{y}_k|\mathbf{x}_k = \mathbf{x}_{k|k-1}^{(i)})} \quad (18)$$

(18) 式は離散的に与えられた状態の条件付き確率でフィルタ分布を近似しており、次ステップの時間更新式でこの分布を用いるためにはサンプリングが必要となる。その際、粒子の多様性が失われフィルタ分布の近似精度が下がる可能性がある点に注意する。なお、この問題の解決法として正則化とよばれる方法がある [7]。最も簡単な方法としては仮想的なシステムノイズを追加する方法や、リサンプリングにより得られた粒子に小さな摂動を

加える方法がある。また、別の方法として Regularized Particle Filter[14] が提案されている。この方法は離散近似されたフィルタ分布 (18) 式を KDE を用いて連続分布として新たに近似し、その近似連続分布からサンプリングすることで新たな粒子を得る方法である。これらにより、粒子の多様性が確保でき、粒子の縮退の問題が回避されることが期待される。

4.3 状態観測時の観測ノイズに関する考察

本論文では、仮定 1 より学習時のみ状態を観測できるとしているが、実際に状態を観測する際には観測ノイズが混入する。そのため、データセットとして利用できる状態のデータは観測ノイズが混入した値 $\mathbf{z}_k$ となる。

いま、状態 $\mathbf{x}_k$ を観測する際に混入する観測ノイズを $\mathbf{v}_k$ とし、 $E[\mathbf{v}_k] = 0$ とする。また、実際に観測できる値 $\mathbf{z}_k$ と状態 $\mathbf{x}_k$ との間に以下の関係が成り立つとする。

$$\mathbf{z}_k = \mathbf{x}_k + \mathbf{v}_k$$

このとき、両辺の期待値をとると、

$$E[\mathbf{z}_k] = E[\mathbf{x}_k]$$

となり、実際に観測できる値 $\mathbf{z}_k$ の平均値は本来取得したい値 $\mathbf{x}_k$ の平均値と一致することがわかる。つまり、実際の学習では予測分布 $p(\mathbf{z}_{k+1}|\mathbf{Y}_k)$ およびフィルタ分布 $p(\mathbf{z}_k|\mathbf{Y}_k)$ が学習されるが、そこから得られるサンプル値を用いて平均値 $\hat{\mathbf{z}}_k$ を計算することで状態の平均値 $\hat{\mathbf{x}}_k$ が求まることになる。

4.4 アルゴリズム

Procedure 1 に提案する状態推定法の処理の流れを示す。提案法は条件付き確率密度関数をデータセットから (8) 式もしくは (13) 式を用いて事前に学習している点が既存の PF との大きな違いである。なお、参考文献 [13] において、基底関数が (10) 式で与えられたとき、(8) 式を用いて学習した場合と (13) 式を用いた場合とで同程度の結果になることが示されている。

また、提案アルゴリズムでは初期粒子群 $\hat{\mathbf{x}}_{1|0}^{(i)}$, $i = 1, \dots, M$ を生成する必要があるが、状態の初期分布が必要となる。状態の初期分布が未知の場合、ベイズ統計モデルでしばしば用いられる無情報事前分布の考えを適用し、分散の大きい正規分布で代用する。

(注意 7) 本論文のアルゴリズムは PF をベースにしているが、分布を粒子で近似するタイプのカルマンフィルタであれば適用可能である。たとえば、アンサンブルカルマンフィルタをベースにすることも可能である。

Procedure 1 Nonlinear Kalman filter using conditional density estimation

- 0) Preliminary: Learning of conditional probability densities
 - a) Obtain sets of data $(\mathbf{x}_0^{(i)}, \mathbf{y}_0^{(i)})$, $i = 1, \dots, N$ and $(\mathbf{x}_0^{(i)}, \mathbf{x}_1^{(i)})$, $i = 1, \dots, N$.
 - b) Learn $\hat{p}(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and $\hat{p}(\mathbf{y}_k|\mathbf{x}_k)$ by Eq. (8) or (13) using the sets of data, and set $k = 1$.
 - c) Generate initial particles: $\mathbf{x}_{1|0}^{(i)} \sim p(\mathbf{x}_0)$, $i = 1, \dots, M$.
 - 1) Measurement update
 - a) Compute Eq. (18) using a measurement $\mathbf{y}_k$, particles of a prior distribution $\mathbf{x}_{k|k-1}^{(i)}$, $i = 1, \dots, M$, and $\hat{p}(\mathbf{y}_k|\mathbf{x}_k)$ learned in advance.
 - b) Resample particles $\mathbf{x}_{k|k}^{(i)}$, $i = 1, \dots, M$ from Eq. (18), and approximate filter distribution using the particles.
 - c) Compute mean of the filter distribution, and adopt the mean as an estimation of state:

$$\hat{\mathbf{x}}_k = \frac{1}{M} \sum_{i=1}^M \mathbf{x}_{k|k}^{(i)}.$$
 - 2) Prediction update
 - a) Compute prediction distribution (15) using the particles $\mathbf{x}_{k|k}^{(i)}$.
 - b) Sample particles $\mathbf{x}_{k+1|k}^{(i)}$, $i = 1, \dots, M$ from the prediction distribution which is gaussian mixture.
 - 3) Set $k = k + 1$, and go back to step 1).
-

5. 数値シミュレーション

5.1 例 1: Van der Pol モデルに対する状態推定

5.1.1 システムおよび条件

リミットサイクルを発生する次式の Van der Pol モデルに対する状態推定問題 [7] を考える。

$$\begin{cases} \dot{x}_1(t) = x_2(t) + w_1(t) \\ \dot{x}_2(t) = \varepsilon(1 - x_1^2(t))x_2(t) - x_1(t) + w_2(t) \end{cases} \quad (19)$$

ただし、 $\varepsilon = 1.0$ とする。また、 $w_1(t) \sim \mathcal{N}(w_1; 0, 0.026)$, $w_2(t) \sim \mathcal{N}(w_2; 0, 0.01)$ である。さらに、観測方程式は 2 番目の状態変数 x_2 が観測されると仮定し、次式として定義する。

$$y(t) = x_2(t) + v(t)$$

提案法の性能を評価するために、観測方程式ならびに観測ノイズの確率密度関数が既知の場合の PF と比較し、次式で定義される二乗平均平方根誤差 (RMSE) で評価

Table 1 Number of dataset

	supervised data	evaluation data
dataset 1	200	200
dataset 2	2000	2000

Table 2 Root mean squared errors of estimates (# of particles is 100)

	$\hat{\mathbf{x}}$	$\hat{x}_2$
PF	0.475	0.334
PF with regularization	0.382	0.242
proposed using dataset 1	0.610	0.321
method using dataset 2	0.365	0.212
no method ($x_2 = y$)	–	0.316

する。

$$\text{RMSE} : \sqrt{\frac{\sum_{k=1}^N \|\hat{\mathbf{x}}_k - \mathbf{x}_k^r\|_2^2}{N}}$$

ここで、 $\mathbf{x}^r$ は $\mathbf{x}$ の真値である。なお、(19) 式に対して PF を適用するために、サンプリング間隔を 0.1s とし (19) 式をオイラー法によって離散化する。また、提案法の学習データ数および粒子数を変化させ、提案法の性能を評価する。

提案法では事前に状態モデルおよび観測モデルを学習する必要がある。本数値例では KDE を用いた方法 (13) 式を用いてこれらのモデルを学習する。その際の学習データ数を Table 1 に示す。カーネル中心は学習データの 1/2 の数だけランダムに選択する。また、学習に必要なパラメータは平均対数尤度を基準とした一個抜き交差検証 (LOOCV) により決定する。

5.1.2 結果

Table 2 に PF および提案法による推定値の RMSE を示す。ここで、 $\hat{\mathbf{x}}$ に関する RMSE に加え、 $\hat{x}_2$ に着目し、観測値をそのまま x_2 の推定値とした場合 (no method) の RMSE も記述する。なお、Table 2 はそれぞれ 100 回のモンテカルロシミュレーションを行った際の平均値を示している。この結果より、学習データが少ない場合は PF に比べて推定誤差が大きいものの、学習データを増やすことで PF よりも推定誤差が小さくなっていることがわかる。また、 $\hat{x}_2$ に着目すると、提案法を用いることで観測値 y をそのまま推定値とするよりも RMSE が小さくなっていることがわかる。つまり、提案法を用いることで観測値 y に印加されているノイズが低減されていることがわかる。

Table 3 に粒子数を変えたときの提案法の RMSE を示す。粒子数を増やすことでまた推定誤差が小さくなっていることがわかる。

Table 4 に粒子数を変えたときの PF の RMSE を示す。粒子数の増加に伴い推定誤差が大きく減少している

Table 3 Root mean squared errors of estimates using the proposed method with dataset 1

# of particles	$\hat{\mathbf{x}}$	$\hat{x}_2$
100	0.610	0.321
500	0.525	0.278
1000	0.511	0.270

Table 4 Root mean squared errors of estimates using particle filter

# of particles	$\hat{\mathbf{x}}$	$\hat{x}_2$
100	0.475	0.334
200	0.346	0.226
300	0.293	0.182

ことがわかる。このことから、Table 2 において、PF は分布を近似するにあたり粒子数が少なかったと考えられる。一方、注意 5 で言及したように提案法の予測分布の近似精度は粒子数とカーネル数に依存する。そのため、Table 2 において、カーネル数が多い場合の提案法の性能は PF を上回ったと考えられる。

5.2 例 2: エノン画像に対する時系列解析

5.2.1 システムおよび条件

カオス時系列の代表であるエノン写像 [15] を例題として、その時系列解析を考える。

エノン写像は次式で示される 2 次元写像であり、状態の 1 番目の要素が観測できるものとする。

$$\begin{cases} x_{1,k+1} = 1 - ax_{1,k}^2 + x_{2,k} \\ x_{2,k+1} = bx_{1,k} \\ y_k = x_{1,k} + v_k \end{cases}$$

ここで、 a および b はパラメータであり、本論文では $a = 1.4$ 、 $b = 0.3$ として議論する。また、 v_k は観測ノイズで、 $v_k \sim \mathcal{N}(v; 0, 0.1)$ とする。このシステムに対して次式の時間遅れ座標系を定義し、観測時系列データを埋め込む。

$$\mathbf{z}_k = \begin{bmatrix} y_k & y_{k-1} & \cdots & y_{k-(d-1)} \end{bmatrix} \in \mathbb{R}^d$$

本項では、この時間遅れ座標系を状態として状態モデルと観測モデルを事前に学習し、観測値の真値を推定する問題を考える。なお、数値例 1 と同様、モデルは KDE を用いた方法 (13) 式を用いて学習する。

提案法の性能を評価するために、前項同様、RMSE を用いて推定値 $\hat{y}_k$ と真値 y_k^r の誤差を評価し、PF と比較する。また、観測値をそのまま推定値とした場合 (no method) の RMSE も評価する。

PF では状態ノイズが存在しない場合、粒子が縮退し、分布の近似精度が劣化する。そこで、小さな分散をもつ仮想的なノイズを導入し、PF の時間更新式を計算する際に仮想的なノイズを追加する。本論文では平均 $\mathbf{0}$ 、共

Table 5 Root mean squared errors of estimates in time series analysis for Henon map

# of particles	proposed method	PF	no method
200	0.261	0.257	0.317
400	0.255	0.243	

分散 $\text{diag}(0.01, 0.01)$ のガウスノイズを仮想的なノイズとして加える。

提案法において、時間遅れ座標系の次数を $d=5$ とする。また、モデルの学習に関して、状態モデルの学習データ数は 2000、観測モデルの学習数は 200 とする。さらに、状態モデルで用いるカーネル中心は学習データと同じものを使用し、観測モデルのカーネル中心は学習データの $1/2$ の数だけランダムに選択する。学習に必要なパラメータは平均対数尤度を基準とした LOOCV により決定する。

5.2.2 結果

Table 5 に PF および提案法による推定値の RMSE を示す。なお、前項同様、本結果は 100 回のモンテカルロシミュレーションを行った際の平均値を示している。この結果より提案法と PF の推定精度がほぼ同等であることがわかる。また、観測値 y に比べて提案法および PF の RMSE が小さくなっていることがわかる。つまり、前項同様、ノイズが低減されていることがわかる。なお、提案法はデータのみから状態推定器を構築していることをここで改めて明記しておく。つまり、提案法は、ノイズの事前情報をはじめ、対象の動特性があらかじめ記述できていなくても PF と同程度の推定精度を実現できることを意味する。

6. おわりに

本論文では、システムの動特性は条件付き確率密度関数で表されることに着目し、条件付き確率密度推定法を用いてデータから直接システムの動特性を推定する方法を提案した。そして、その推定されたモデルを用いて粒子フィルタを構成することで、モデルの事前情報を必要としない状態推定法を提案した。数値シミュレーションではリミットサイクルを発生する系の状態推定問題とカオス時系列解析問題の二つの例を取り上げ、モデルが既知の場合の PF と比較することで提案法の有効性を確認した。

謝 辞

本研究は JSPS 科研費 16K18128 の助成を受けたものです。ここに感謝の意を表します。

参 考 文 献

- [1] R. E. Kalman: A new approach to linear filtering and prediction problems; *J. Basic Eng.*, Vol. 82, No. 1, pp. 35–45 (1960)
- [2] S. Julier, J. Uhlmann and H. F. Durrant-Whyre: A new method for the nonlinear transformation of means and covariances in filters and estimators; *IEEE Trans. Automat. Contr.*, Vol. 45, No. 3, pp. 477–482 (2000)
- [3] 樋口：粒子フィルタ；電子情報通信学会誌, Vol. 88, No. 12, pp. 989–994 (2005)
- [4] G. Agamennoni, J.-I. Nieto and E.-M. Nebot: An outlier robust Kalman filter; *Proc. 2011 IEEE Int. Conf. on Robotics and Automation*, pp. 1551–1558 (2011)
- [5] Y. Kaneda, Y. Irizuki and M. Yamakita: Fast algorithm of robust Kalman filter via l_1 regression by a closed form solution; *Proc. of 19th IFAC World Congress*, pp. 6153–6158 (2014)
- [6] Y. Kaneda, Y. Irizuki and M. Yamakita: Robust unscented Kalman filter via l_1 regression and design method of its parameters; *Proc. of 9th Asian Control Conference*, pp. 1–6 (2013)
- [7] 片山：非線形カルマンフィルタ, 朝倉書店 (2011)
- [8] S. Sarkka and A. Nummenmaa: Recursive noise adaptive Kalman filtering by variational Bayesian approximations; *IEEE Trans. Automat. Contr.*, Vol. 54, No. 3, pp. 596–600 (2009)
- [9] G. Terejanu, P. Singla, T. Singh and P. D. Scott: Adaptive Gaussian sum filter for nonlinear Bayesian estimation; *IEEE Trans. Automat. Contr.*, Vol. 56, No. 9, pp. 2151–2156 (2011)
- [10] C. Novara, F. Ruiz and M. Milanese: Direct filtering: A new approach to optimal filter design for nonlinear systems; *IEEE Trans. Automat. Contr.*, Vol. 58, No. 1, pp. 86–99 (2013)
- [11] T. Kanamori, S. Hido and M. Sugiyama: A least-squares approach to direct importance estimation; *J. Machine Learning Research*, Vol. 10, pp. 1391–1445 (2009)
- [12] M. Sugiyama, I. Takeuchi, T. Suzuki, T. Kanamori, H. Hachiya and D. Okanohara: Least-squares conditional density estimation; *IEICE Trans. Information and Systems*, Vol. E93-D, No. 3, pp. 583–594 (2010)
- [13] 金田, 入月：データを用いたフィルタ分布の直接推定と状態推定への応用；計測自動制御学会論文集, Vol. 53, No. 4, pp. 295–297 (2017)
- [14] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking; *Trans. on IEEE Signal Processing*, Vol. 50, No. 2, pp. 174–188 (2002)
- [15] 合原 編, 池口, 山口, 小室 著：カオス時系列解析の基礎と応用, 産業図書 (2000)

著者略歴

かねだ やすあき
金田 泰昌 (正会員)

2005年東京工業大学大学院理工学研究科機械制御システム専攻修士課程修了。同年(株)リコー入社, 2007年(地独)東京都立産業技術研究センター入所。2014年東京工業大学大学院理工学研究科機械制御システム専攻博士後期課程修了。現在, 東京都立産業技術研究センター情報技術グループ主任研究員。状態推定理論, およびその実装方法や応用に関する研究に従事。博士(工学)。IEEE, 計測自動制御学会, 電気学会, 各会員。

いりづき やすはる
入月 康晴

2002年名古屋大学大学院工学研究科計算理工学専攻博士課程修了。現在, (地独)東京都立産業技術研究センター情報技術グループ長(上席研究員)。知能化技術, 非線形モデリング, 組込みシステムの信頼性向上に関する研究開発に従事。博士(工学)。計測自動制御学会, 電気学会, 日本知能情報ファジィ学会, 各会員。