01月16日 進捗報告

Kalman Filter によるパラメータ推定

機械学習に特有のパラメータ推定法？

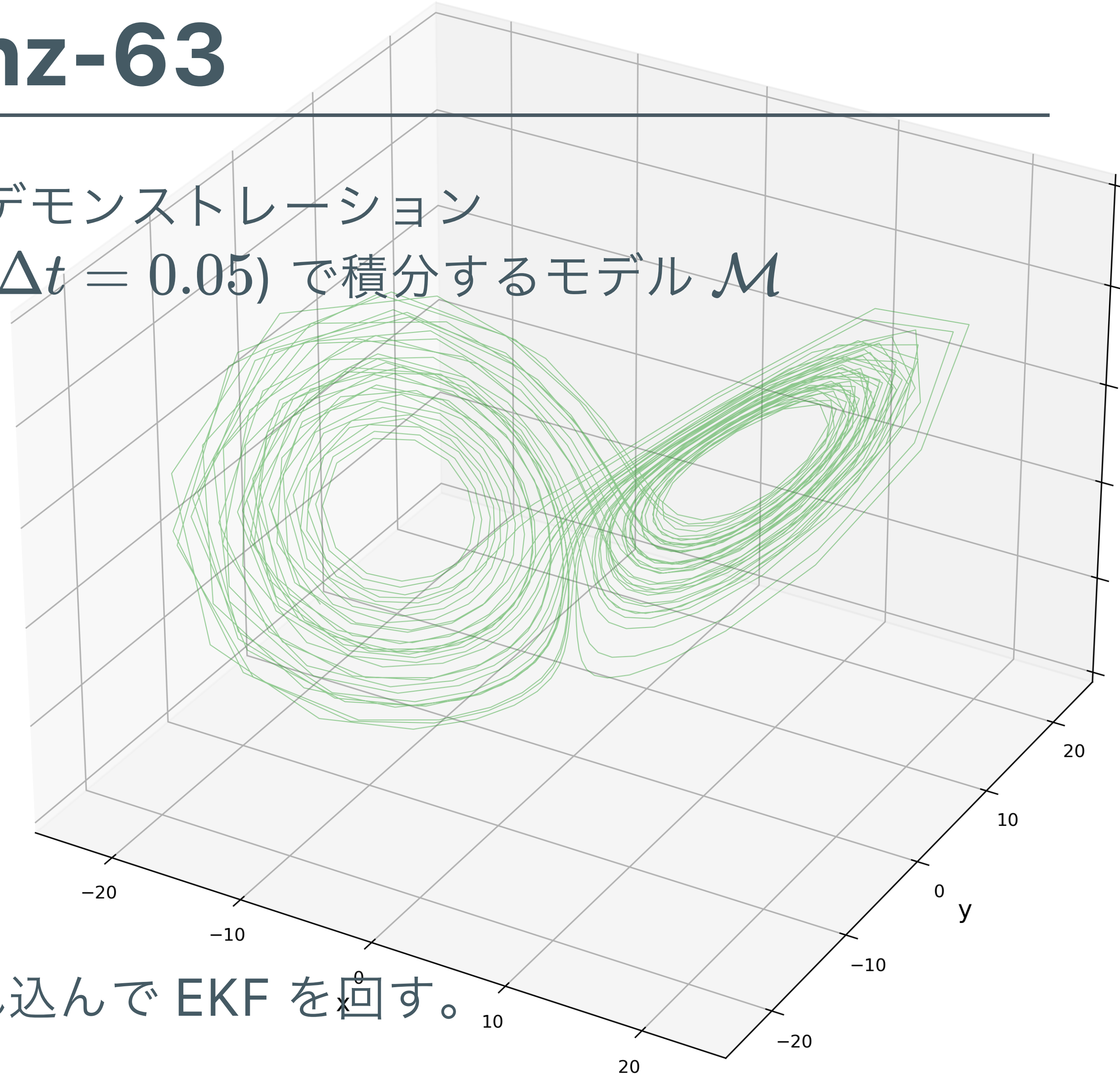
0500-32-7354,　佐藤 匠

前回 (1/9) のふりかえり

[ふり返り] 問題設定｜Lorenz-63

「Kalman Filter によるパラメータ推定」 のデモンストレーション
Lorenz-63 モデルを 4次の Runge-Kutta 法 ($\Delta t = 0.05$) で積分するモデル $\mathcal{M}$

$$\frac{\mathrm{d}\vec{x}}{\mathrm{d}t} = \begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x), \\ \dfrac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y \\ \dfrac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z. \end{cases}$$



パラメータは $\sigma,\ \rho,\ \beta$ の3つ。
推定するパラメータを状態変数ベクトルに入れ込んで EKF を回す。
例： $\vec{x} = (x, y, z, \sigma)^T$

[ふり返り] Inf付パラメータ推定実験｜実験の設定

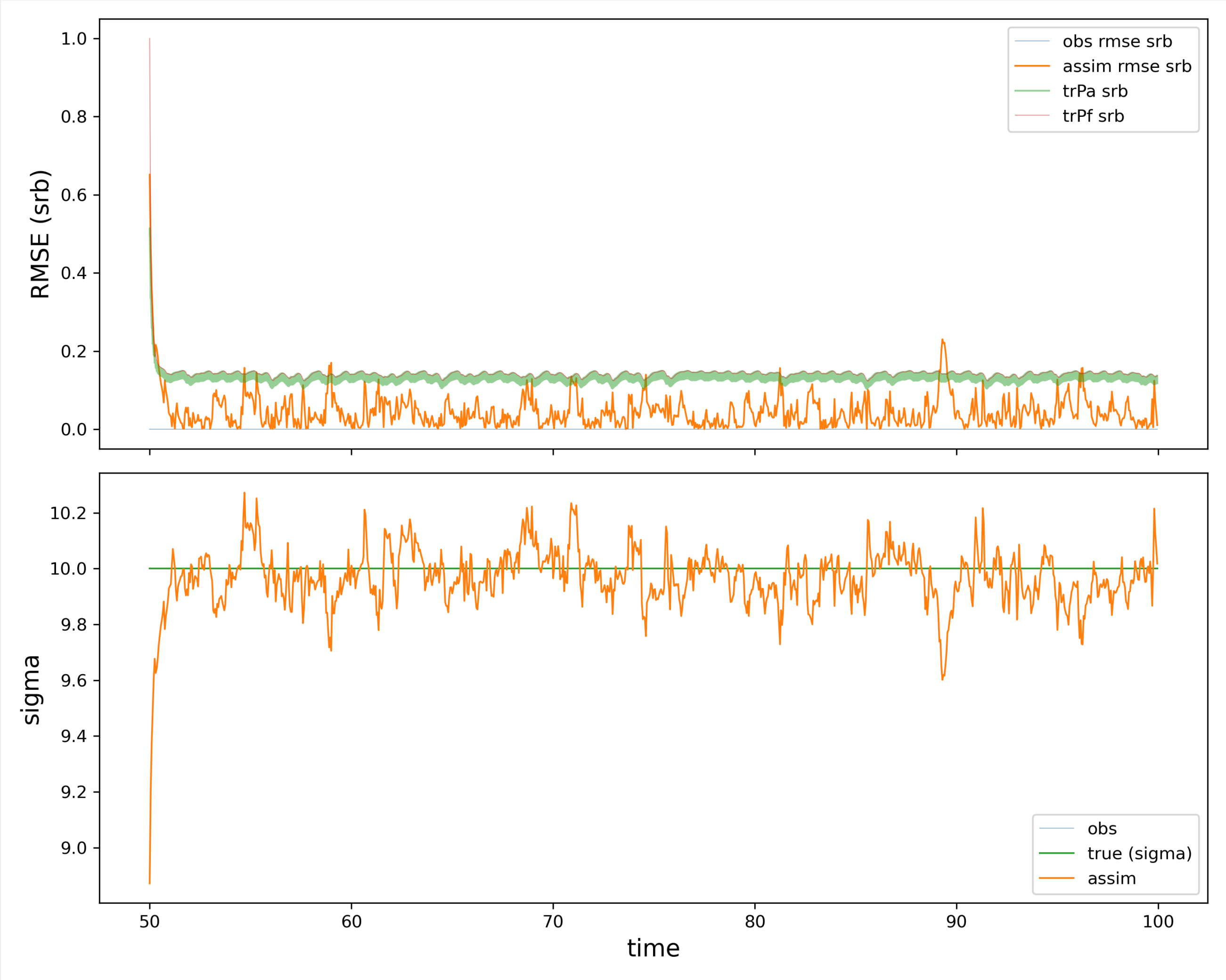
状態変数空間とパラメータ空間でインフレーションを変えやすいAdditive Inflation を採用。

$$P^a_{\mathrm{inf}} = P^a_{\mathrm{orig}} + P^{\mathrm{add}}$$

$$P^{\mathrm{add}} = \begin{pmatrix} \alpha_{xyz} & 0 & 0 & 0 \\ 0 & \alpha_{xyz} & 0 & 0 \\ 0 & 0 & \alpha_{xyz} & 0 \\ 0 & 0 & 0 & \alpha_\sigma \end{pmatrix}$$

調整パラメータは $\alpha_{xyz}, \alpha_\sigma$ の二つ。

[ふり返り] Inf付パラメータ推定実験｜(x, y, z, σ)



正解は $\sigma = 10$
σ の初期値は適当に 8 とする。

$\alpha_{xyz} = 0.0$
$\alpha_\sigma = 0.01$

正解の周りを上下に振動
→ うまく推定できている。

RNN に向けて

$dx = f(x,t)dt$ ではなく $x_i = f(x_{i-1})$

AR や RNN は 漸化式のようなモデル

AR(2) モデル

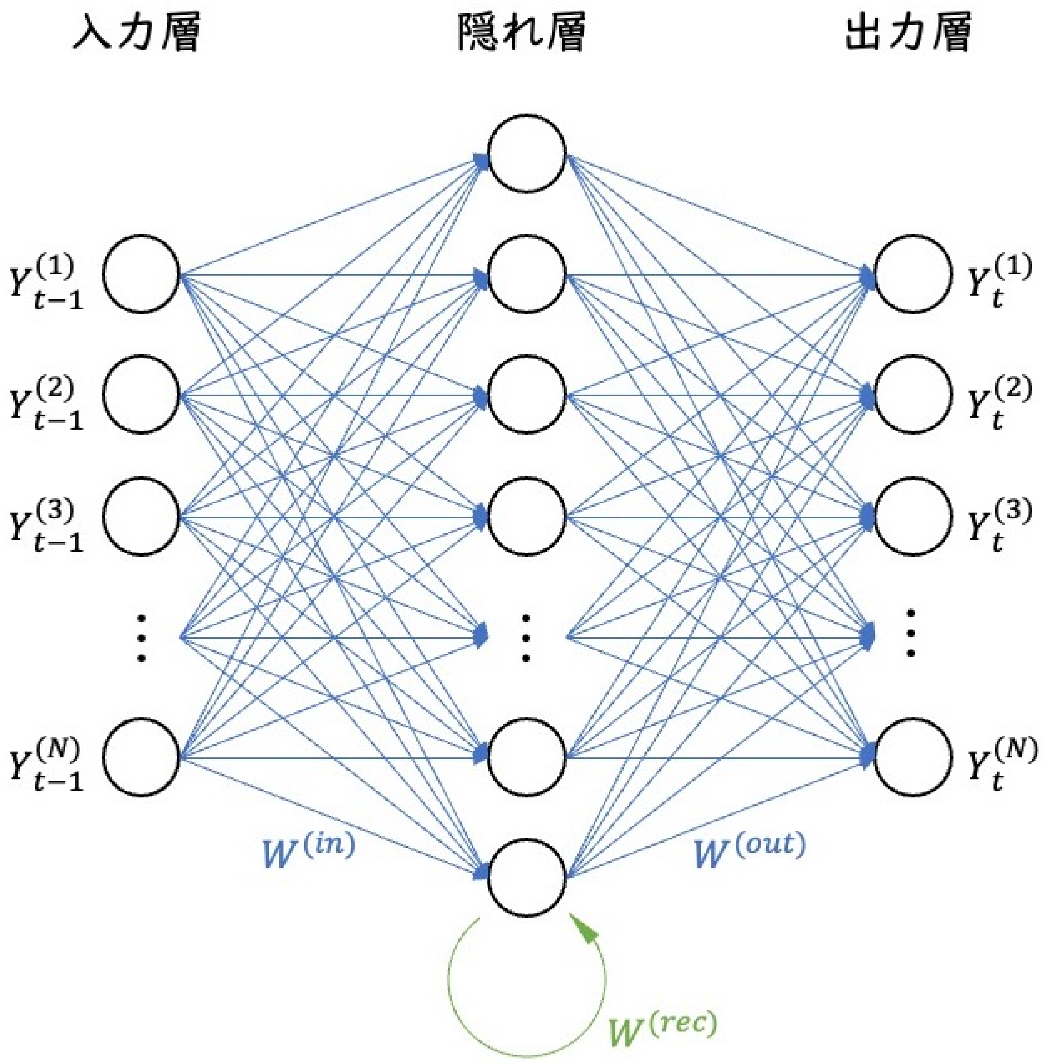
$$Z_t = \varepsilon_t + c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2}$$

$$Z_t = f(Y_{t-1}, Y_{t-2})$$

（c, ϕ_i がパラメータ）

直前の値を入れると、
次ステップの値が返ってくる。

RNN →



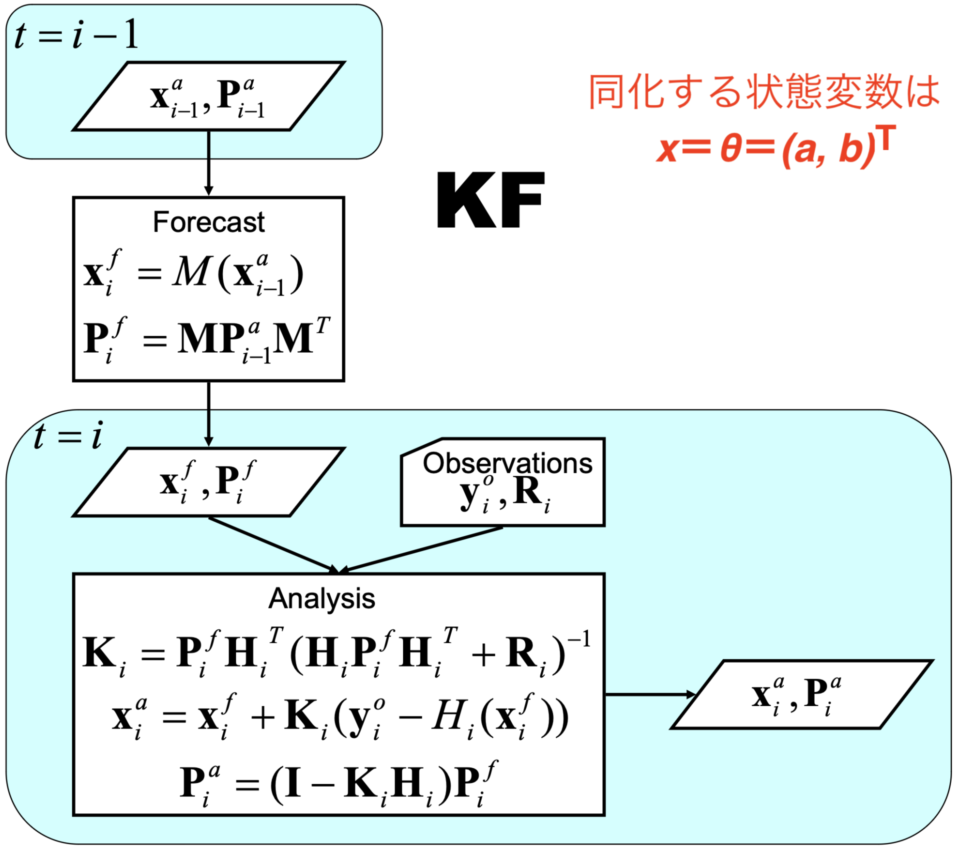
機械学習 "的な" パラメータ推定

Kalman Neuro Computing における機械学習 (パラメータ推定) は、講義で習った推定法**ではない！**

これまで試していた推定法

推定するパラメータを状態変数ベクトルに入れ込んで EKF を回す。

例： $\vec{x} = (x, y, z, \sigma)^T$



KF

同化する状態変数は
$x = \theta = (a, b)^T$

Kalman Neuro Computing での推定法

機械学習モデルは時間発展モデル $x_{i+1} = \mathcal{M}(x_i),\ (x_i = y_i,\ H = I)$ ではなく、

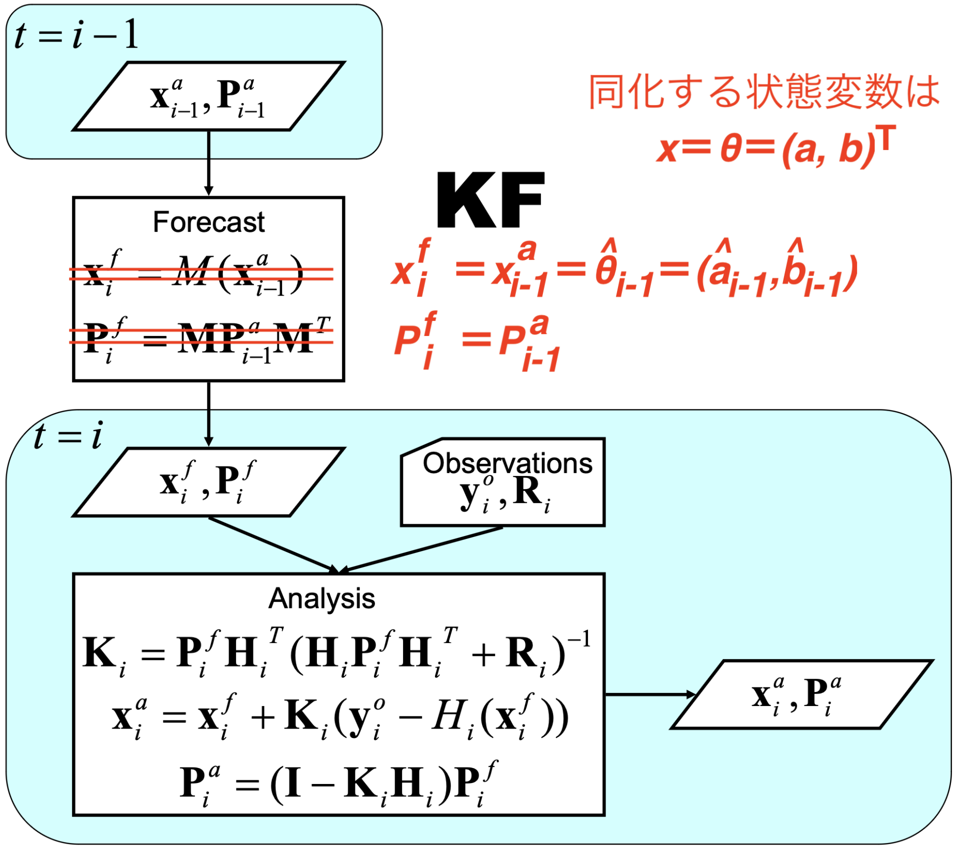
観測演算子 $y^b_i = \mathcal{H}(x^f_i) = z^{out}(y_{i-1}|\theta = \hat{\theta}_{i-1})$ で $M = I$

機械学習 "的な" パラメータ推定

Kalman Neuro Computing における
機械学習 (パラメータ推定) は、講義
で習った推定法**ではない！**

これまで試していた推定法

推定するパラメータを状態変数ベクトルに
入れ込んで EKF を回す。
例： $\vec{x} = (x, y, z, \sigma)^T$



同化する状態変数は
$x = \theta = (a, b)^T$

KF

$x^f_i = x^a_{i-1} = \hat{\theta}_{i-1} = (\hat{a}_{i-1}, \hat{b}_{i-1})$

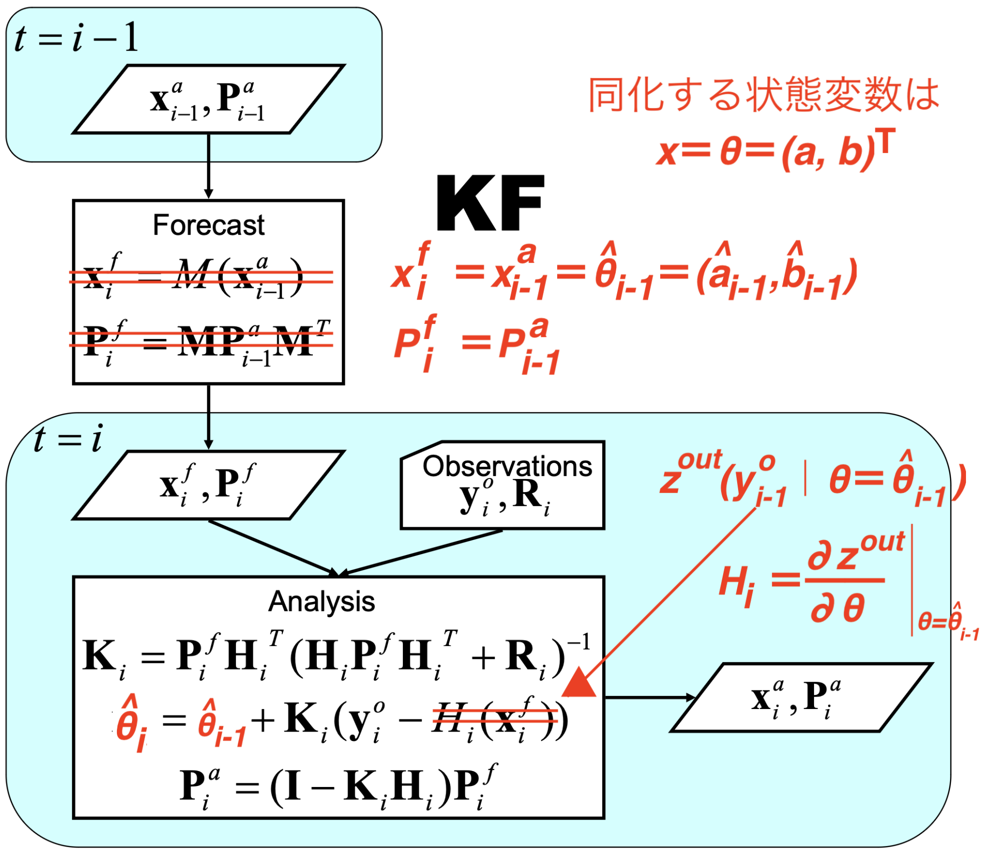
$P^f_i = P^a_{i-1}$

Kalman Neuro Computing での推定法

機械学習モデルは時間発展モデル $x_{i+1} = \mathcal{M}(x_i)$, $(x_i = y_i,\ H = I)$ ではなく、
観測演算子 $y^b_i = \mathcal{H}(x^f_i) = z^{out}(y_{i-1} | \theta = \hat{\theta}_{i-1})$ で $M = I$

機械学習 "的な" パラメータ推定

Kalman Neuro Computing における
機械学習 (パラメータ推定) は、講義
で習った推定法**ではない！**



これまで試していた推定法

推定するパラメータを状態変数ベクトルに
入れ込んで EKF を回す。
例： $\vec{x} = (x, y, z, \sigma)^T$

Kalman Neuro Computing での推定法

機械学習モデルは時間発展モデル $x_{i+1} = \mathcal{M}(x_i)$, $(x_i = y_i,\ H = I)$ ではなく、
観測演算子 $y^b_i = \mathcal{H}(x^f_i) = z^{out}(y_{i-1} | \theta = \hat{\theta}_{i-1})$ で $M = I$

TLM を H として使う

$$\begin{pmatrix} \delta x_i \\ \delta y_i \\ \delta a \\ \delta b \end{pmatrix} = \begin{pmatrix} \frac{\partial x_i}{\partial x_{i-1}} & \frac{\partial x_i}{\partial y_{i-1}} & \frac{\partial x_i}{\partial a} & \frac{\partial x_i}{\partial b} \\ \frac{\partial y_i}{\partial x_{i-1}} & \frac{\partial y_i}{\partial y_{i-1}} & \frac{\partial y_i}{\partial a} & \frac{\partial y_i}{\partial b} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \delta x_{i-1} \\ \delta y_{i-1} \\ \delta a \\ \delta b \end{pmatrix}$$

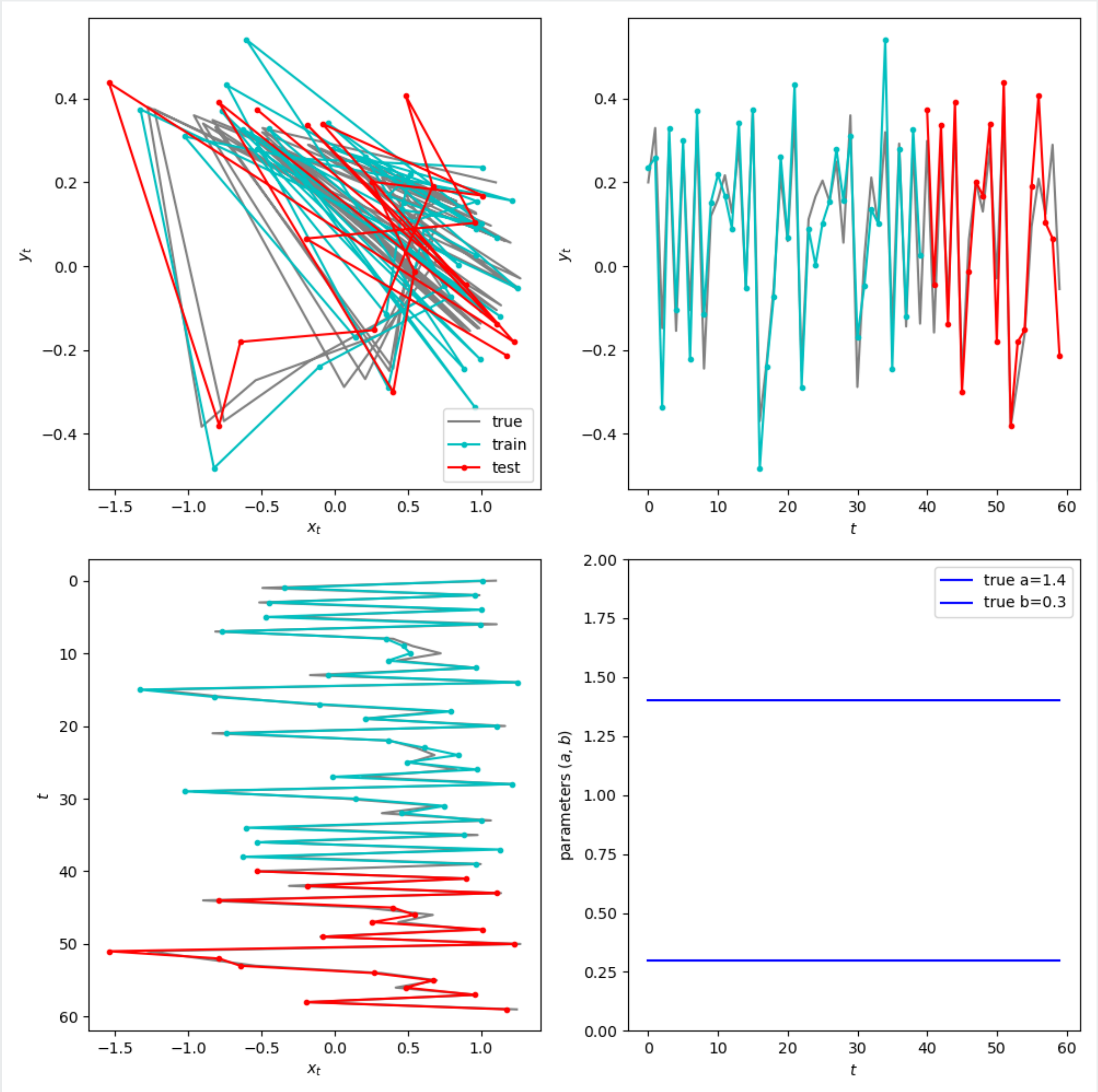
Simultaneous estimation で使う Henon 写像の TLM のうち、右上の 2x2 が

$$H = \frac{\partial z^{out}}{\partial \theta}$$

RNN に向けて

機械学習"的な"パラメータ推定を試す

KNC っぽい推定 | Henon 写像



離散的なモデル。

$$\begin{cases} x_{i+1} = 1 - a x_i^2 + y_i \\ y_{i+1} = b x_i \end{cases}$$

パラメータは $a,\ b$

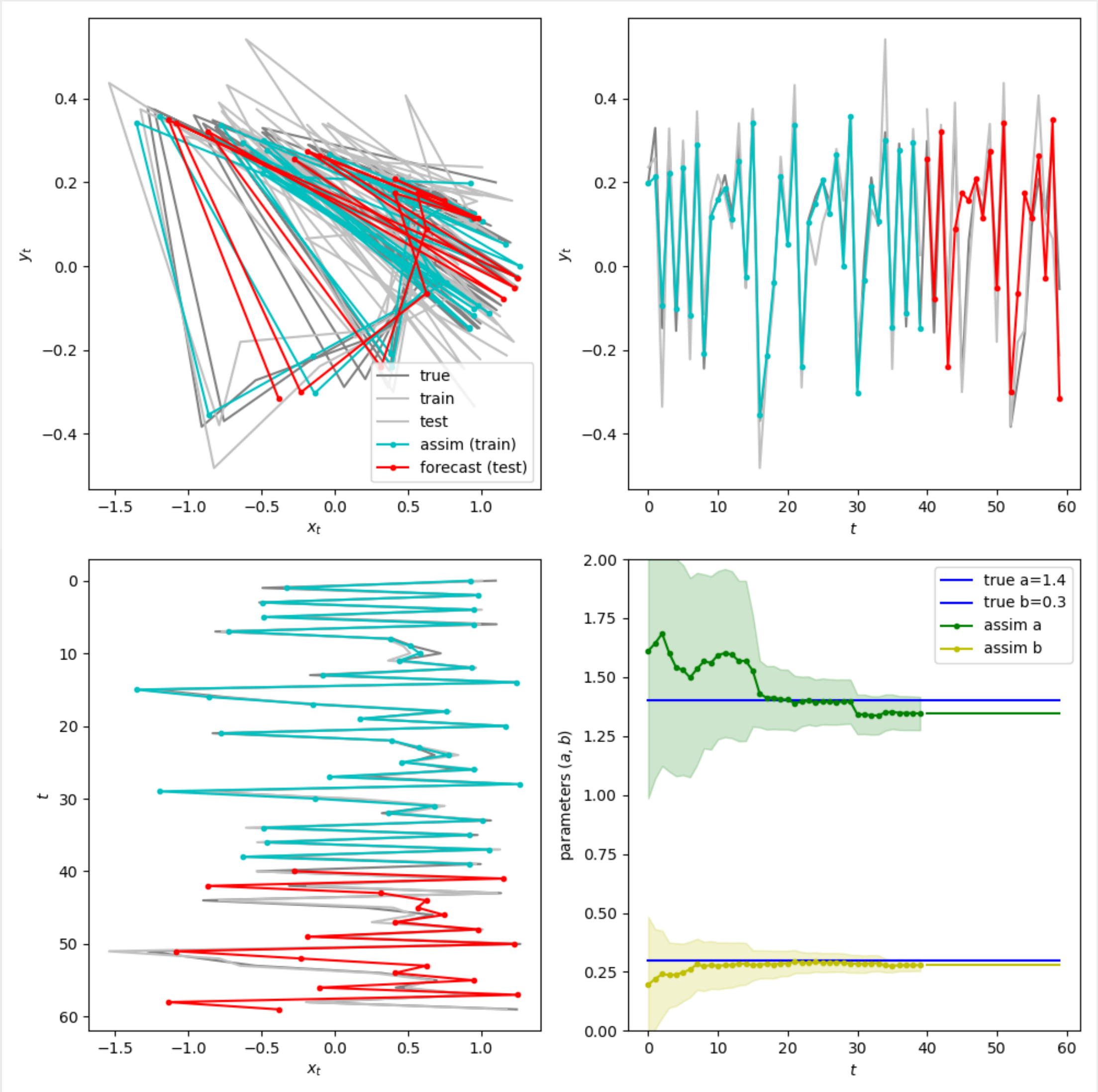
正解の値は
$a = 1.4, \quad b = 0.3$
とする。

初期値
$\vec{Y}_0 = (1.1,\ 0.2)^T$
として 60 時点のデータを作成し、

- 訓練:
  前 40 点学習
- 検証:
  後 20 点予測

観測誤差は半径 0.1 の正規分布

KNC っぽい推定｜慣れ親しんだ推定法 (inf無, 軌道)



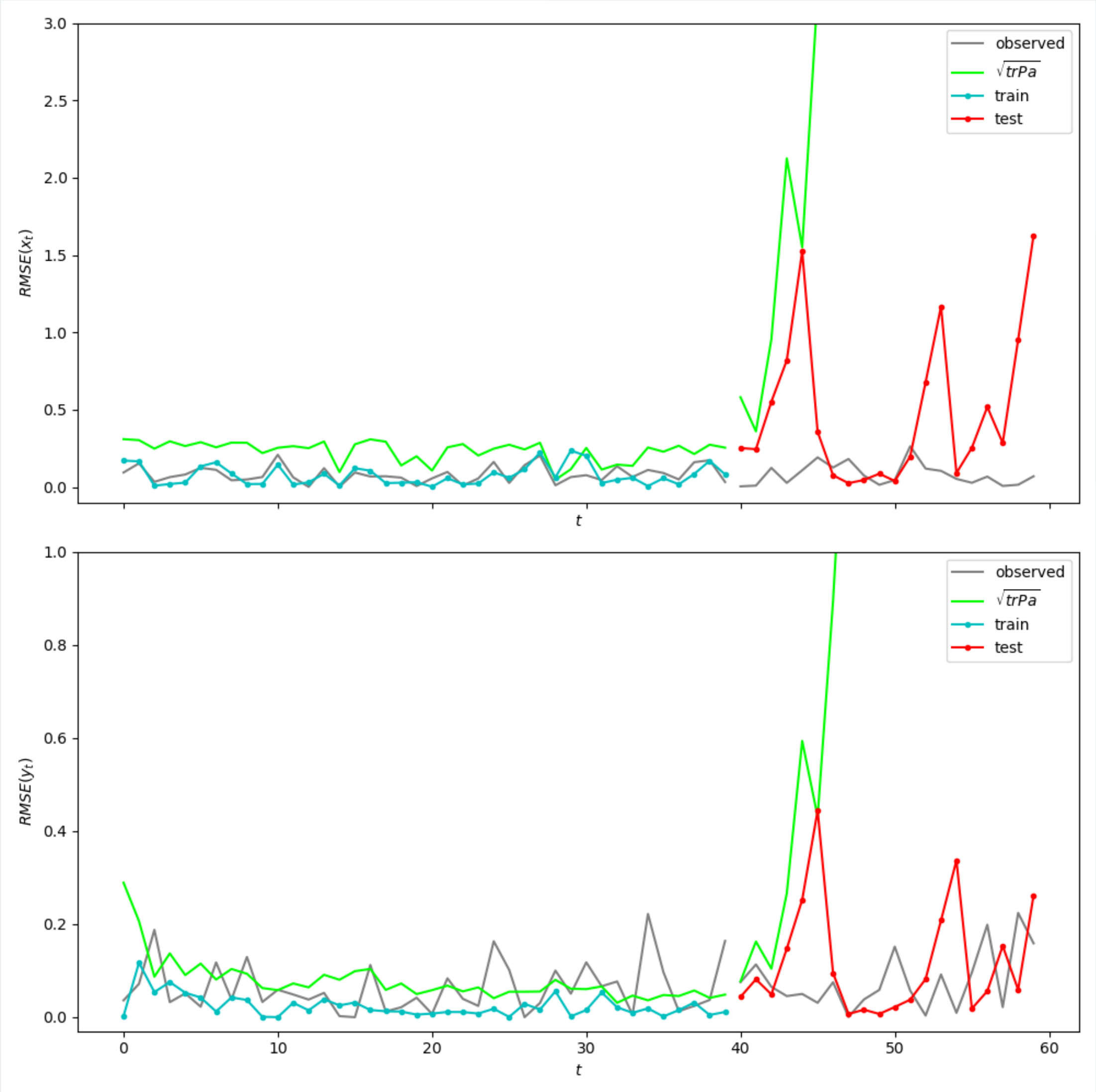
状態変数ベクトルを$\vec{x} = (x, y, a, b)^T$ として同化（Inflation なし）。

つまり、Henon写像を
時間発展モデル $x_{i+1} = \mathcal{M}(x_i)$
として EKF に入力している。

初期値は $(a, b) = (2.0, 0.0)$,
(x, y) は最初の観測を鵜呑み、
P^a は $P^a_{xy} = 0.1$, $P^a_{ab} = 0.5$

パラメータ空間でフィルター発散。

KNC っぽい推定｜慣れ親しんだ推定法 (inf無, 誤差)



状態変数ベクトルを

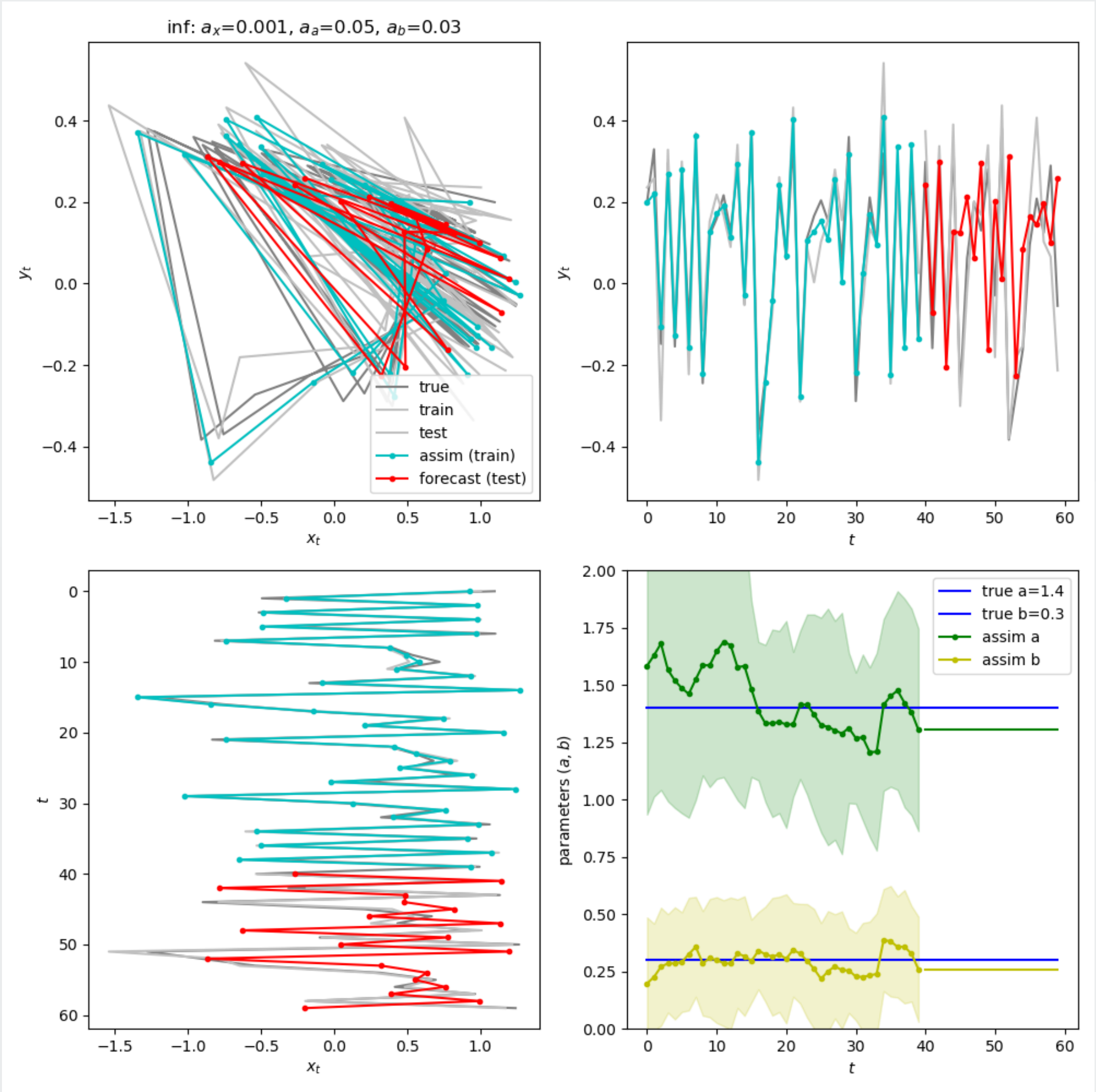
$\vec{x} = (x, y, a, b)^T$

として同化（Inflation なし）。

x, y の予測だけを見ると
うまくいっている。

予測期間で $P_i^f = MP_{i-1}^f M^T$
とすると P^a **の値は爆発する。**
（Q. 予報のRMSEから外れすぎ？）

KNC っぽい推定｜慣れ親しんだ推定法 (inf有, 軌道)



状態変数ベクトルを

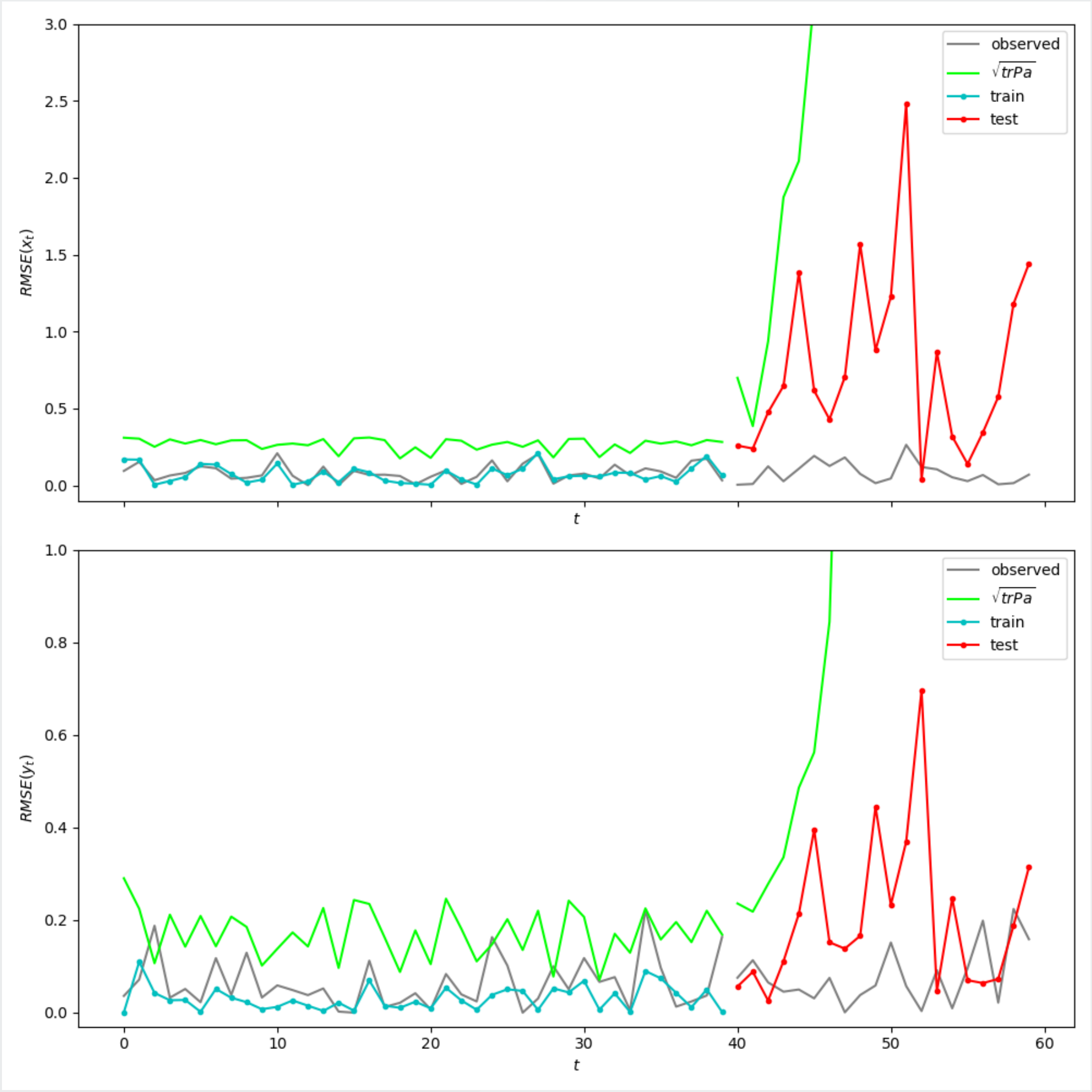
$\vec{x} = (x, y, a, b)^T$

として同化（Inflation あり）。

Additive Inflation は P^a の
x, y に それぞれ +0.001
a に +0.05
b に +0.03

動いたが a の P^a が大きい。

KNC っぽい推定｜慣れ親しんだ推定法 (inf有, 誤差)



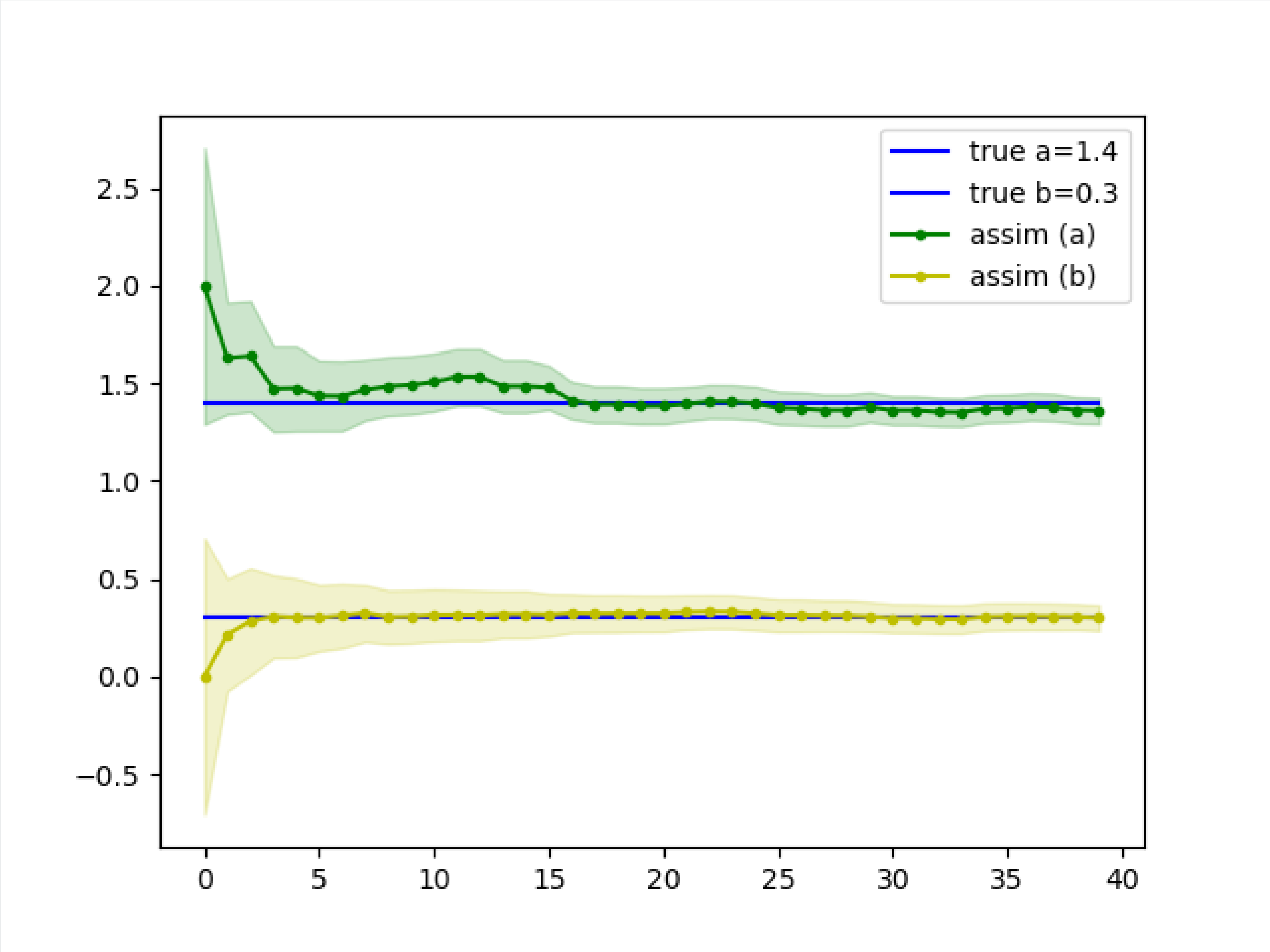
状態変数ベクトルを

$\vec{x} = (x, y, a, b)^T$

として同化（Inflation あり）。

x, y の予報の RMSE が悪化した。

KNC っぽい推定｜新しい推定法 (infなし)

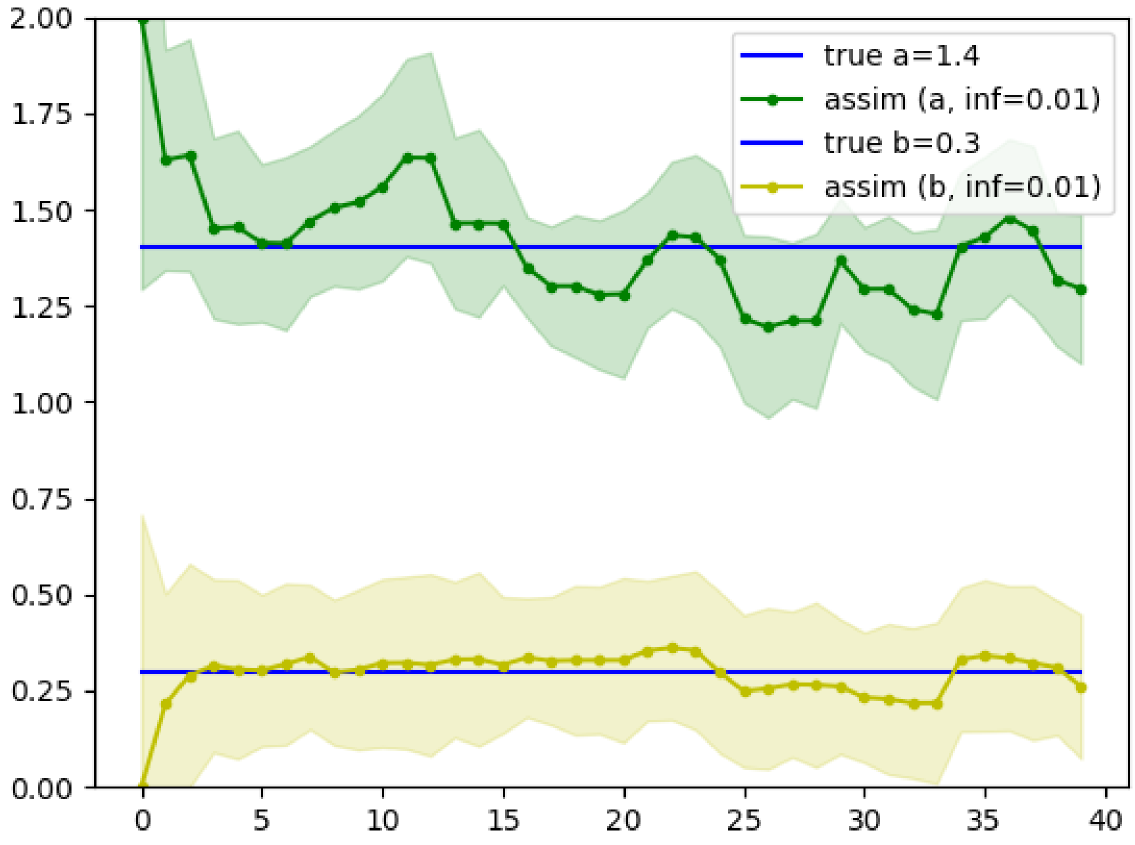


状態変数ベクトルを $\vec{x} = (a, b)^T$ として同化（Inflation なし）。

つまり、Henon写像を観測演算子 $z_{i+1} = \mathcal{H}(y_i|a, b)$ として EKF に入力している。

Inflation 無しでも動く時と、動かない時がある。

KNC っぽい推定｜新しい推定法 (infあり)



状態変数ベクトルを $\vec{x} = (a, b)^T$ として同化（Inflation なし）。

Additive Inflation は P^a の a, b にそれぞれ +0.01

少ない Inflation で理想的な動き。

Q.
x, yは同化していないので、予測誤差の予測 P^f_{xy} は計算不可？