数理解析研究所講究録 1244 巻 2002 年 119-127

ON-LINE MONTE CARLO 学習

栗原 貴之, 中田 洋平, 用水 邦明, 松本 隆
T. Kurihara, Y. Nakada, K. Yosui and T. Matsumoto
早稲田大学理工学部電気電子情報工学科
Department of Electrical, Electronics and Computer Engineering
Waseda University, 3-4-1 Ohkubo Shinjuku-ku Tokyo, Japan
Phone: +81 3 5286 3377
Fax: +81 3 3702 4735
E-mail: takashi@mse.waseda.ac.jp
Core Research for Evolutional Science and Technology(CREST)

概 要

逐次学習 (on-line learning) は、一括型学習 (batch learning) と対比的に用いられるコンセプトで、学習データが逐次的に与えられる場合を意味する。一括型学習では、システムが時間に対して変化するような問題には、適用の困難が予想される。しかし逐次学習は、学習データが逐次的に与えられるので, 予測しようとする関数が時間的に変化する場合にも適用可能であると考えられる。我々のグループでは、逐次学習の Bayes 的アプローチを試み、主として非線形 dynamical system の再構築と予測を行う。Bayse 的アプローチにおいては、対象となるシステムの予測分布と事後分布の解析的評価が必要であるが、一般にその計算は難しい。ここで用いる実際の手法としては、Importance Resampling を用いた Sequential Monte Carlo を考え、対象とする分布からサンプリングを行い、データフィット用関数族のパラメータを推定するとともに、そのランダムウォークレベルを表すパラメータやシステムのノイズレベルを示すハイパーパラメータの学習・推定も同時に行う。最後に、この手法を時間的に変化する関数に適用し、その有効性を確認するとともに、その精度を検証する。

A Bayesian on-line learning scheme with Sequential Monte Carlo incorporating Importance Resampling is proposed. If environment changes with respect to time, *i. e.*, the input-output relationship to learn varies over time, then on-line learning will be called for. The proposed scheme adjusts not only parameters for data fitting but also adjusts hyperparameters on-line. The scheme is tested against simple examples and is shown to be functional.
keywords : Bayes, online 学習, Importance Resampling, Sequential Monte Carlo, hyperparameter

1 はじめに

逐次学習 (on-line learning) は，一括型学習 (batch learning) と対比的に用いられるコンセプトで，学習データが逐次的に与えられる場合を意味する．逐次学習は，学習・予測の対象が時間と共に変化する場合に対しても対応できるのが特徴のひとつである．

我々のグループでは，一括型学習を階層 Bayes 的枠組みからとらえ，周辺尤度二次近似および MCMC(Markov Chain Monte Carlo) を用い，主として非線形 dynamical system の再構築と予測を行ってきた [1],[2].
この報告では，逐次学習の Bayes 的アプローチを試みる．Bayes 的アプローチの問題のひとつは，パラメータの周辺尤度の計算が困難な事であって，これは一括型学習においても逐次学習でも同様である．より具体的には，パラメータの尤度と先験分布関数の積の周辺化（全空間にわたる積分）を遂行する必要があり，線形・Gauss 雑音の場合以外は解析表示不可能である．Bayes 的手法では，更に，予測分布の計算に，全空間にわたる積分が必要となる．

本稿では，パラメータの周辺化を Sequential Monte Carlo で逐次的に遂行するばかりでなく，ハイパーパラメータも逐次学習する手法を提案し，単純な例に適用して従来法に対する優位性を検証する．学習データが逐次的に与えられるので，予測しようとする (未知) 関数が時間的に変化する場合にも適用可能である．Implementation は Importance Resampling である．

2 定式化

2.1 逐次学習

時刻 t における入力 x_t と出力 y_t が逐次的に与えられる場合，入・出力関係を適当にパラメータ付けされた関数族でフィットし，推定・予測を行う問題を考える．データフィットのための関数族を指定するパラメータを $u_t := (u_{t1}, \cdots, u_{tk_u}) \in \mathbb{R}^{k_u}$ とする時，更新則

$$u_t = \mathbf{G}(u_{t-1}) + \mu_t \tag{1}$$

で，パラメータを学習（推定）する事を試みる．但し，$\mu_t := (\mu_{t1}, \cdots, \mu_{tk_u}) \in \mathbb{R}^{k_u}$ は適当なノイズ過程である．時刻 t までの入・出力データを $D_t := \{x_{t'}, y_{t'}\}_{t'=1}^{t}$ とし，以下の二つの確率密度関数を考える：

$$P(u_t \mid D_{t-1}) = \int P(u_t \mid u_{t-1})\, P(u_{t-1} \mid D_{t-1})\, du_{t-1} \tag{2}$$

$$\begin{aligned} P(u_t \mid D_t) &= \frac{P(y_t \mid u_t)\, P(u_t \mid D_{t-1})}{\int P(y_t \mid u_t)\, P(u_t \mid D_{t-1})\, du_t} \\ &\propto P(y_t \mid u_t)\, P(u_t \mid D_{t-1}) \end{aligned} \tag{3}$$

式 (2) は，時刻 $t-1$ までのデータが与えられたときの，時刻 t の状態 u_t に対する予測分布を表しており，一時刻前の式 (3) を用いて表される．また，式 (3) は，式 (2) を事前分布とし，ベイズ公式を用いて，時刻 t までのデータ D_t が与えられた時の u_t の事後分布となっている．
問題 ： 時刻 $t-1$ までの入・出力データ $D_{t-1} := \{x_{t'}, y_{t'}\}_{t'=1}^{t-1}$ が与えられる時，時刻 t における出力 y_t を予測する逐次予測を行う．但し，x_t は与えられる．
我々は上記の問題を仮定し学習を行う。しかし、更新則 $\mathbf{G}(\cdot)$ の設計とデータフィット用関数族の選択ができたとしても，式 (2), (3) をどの様に計算するかが問題となる．ここでは，Importance Resampling による Sequential Monte Carlo で試みる．

2.1.1 モデル $\mathcal{H}$

データフィットのための関数族として，weight パラメータ w，中間素子数 h，出力関数 $1/(1+\exp(-(\cdot)))$ の三層パーセプトロンを用いるこのモデルを $\mathcal{H}$ と書く．

2.1.2 学習データ

時刻 t における学習データ (x_t, y_t) は，次の確率分布で与えられると仮定する：

$$P(y_t \mid x_t, w_t, \beta_t, \mathcal{H}) = \frac{1}{Z_z(\beta_t)} \exp\left[-\beta_t E_z(x_t, y_t; w_t)\right]$$
$$E_z(x_t, y_t; w_t) = \frac{1}{2}(y_t - f(x_t; w_t))^2$$
$$Z_z(\beta_t) = (\beta_t/2\pi)^{1/2}$$

$f(\cdot; w_t)$ は weight パラメータ $w_t \in \mathbb{R}^k$ をもつパーセプトロンの出力，β_t は時刻 t における観測ノイズ ν_t の不確定性レベルを表すハイパーパラメータである．

2.1.3 weight パラメータ w_t の遷移確率

$$P(w_t \mid w_{t-1}, \gamma_t, \mathcal{H}) = \frac{1}{Z_{w_t}(\gamma_t)} \exp\left[-\gamma_t E_{w_t}(w_t; w_{t-1})\right]$$
$$E_{w_t}(w_t; w_{t-1}) = \frac{1}{2}||w_t - w_{t-1}||^2$$
$$Z_{w_t}(\gamma_t) = (\gamma_t/2\pi)^{k/2}$$

γ_t は時刻 t におけるシステムノイズ μ_t の不確定性レベルを表すハイパーパラメータである．

2.1.4 ハイパーパラメータ (γ_t, β_t) の遷移確率

$$P(\gamma_t \mid \gamma_{t-1}, \mathcal{H}) = \frac{1}{\sqrt{2\pi}\sigma_\gamma \gamma_t} \exp\left[-\frac{(\log\gamma_t - \log\gamma_{t-1})^2}{2\sigma_\gamma^2}\right]$$

$$P(\beta_t \mid \beta_{t-1}, \mathcal{H}) = \frac{1}{\sqrt{2\pi}\sigma_\beta \beta_t} \exp\left[-\frac{(\log\beta_t - \log\beta_{t-1})^2}{2\sigma_\beta^2}\right]$$

$\sigma_\gamma, \sigma_\beta$ は正規分布の標準偏差に相当するパラメータである．また，ハイパーパラメータの時間による変化が余り大きいと学習が不安定になること予想されるため，後に述べられる数値実験においては，σ_γ, σ_β は十分に小さい値としている．

2.1.5 weight パラメータ w_0 の初期分布

weight パラメータの初期分布を以下のような平均 0，分散 α^{-1} の正規分布と仮定する：

$$P(w_0 \mid \mathcal{H}) = \left(\frac{\alpha}{2\pi}\right)^{\frac{k}{2}} \exp\left[-\frac{\alpha}{2}||w_0||^2\right]$$

2.2 事後分布と予測分布

時刻 $t-1$ までのデータ D_{t-1} が与えられた時の $u_t := (w_t, \beta_t, \gamma_t)$ の予測分布, および時刻 t までのデータ D_t が与えられた時の $u_t := (w_t, \beta_t, \gamma_t)$ の事後分布は次のように書ける：

$$P(u_t \mid D_{t-1}, \mathcal{H}) = \int P(u_t \mid u_{t-1}, \mathcal{H})\, P(u_{t-1} \mid D_{t-1}, \mathcal{H})\, du_{t-1} \tag{4}$$

$$P(u_t \mid D_t, \mathcal{H}) = \frac{P(y_t \mid x_t, u_t, \mathcal{H})\, P(u_t \mid D_{t-1}, \mathcal{H})}{\int P(y_t \mid x_t, u_t, \mathcal{H})\, P(u_t \mid D_{t-1}, \mathcal{H})\, du_t} \tag{5}$$

ただし, $P(y_t \mid x_t, u_t, \mathcal{H}) = P(y_t \mid x_t, w_t, \beta_t, \mathcal{H})$,
$P(u_t \mid u_{t-1}, \mathcal{H}) = P(w_t \mid w_{t-1}, \gamma_t, \mathcal{H}) P(\gamma_t \mid \gamma_{t-1}, \mathcal{H}) P(\beta_t \mid \beta_{t-1}, \mathcal{H})$,
$P(u_0 \mid \mathcal{H}) = P(w_0 \mid \mathcal{H}) P(\gamma_0 \mid \mathcal{H}) P(\beta_0 \mid \mathcal{H})$ である.

2.2.1 Sequential Monte Carlo : Importance Resampling

ここでは Importance Resampling を使い, 式 (4) (5) で示す確率分布からサンプルを得る手法を述べる.

まず, 目的とする確率密度関数 $P(u)$ とは別の確率密度関数 $Q(u)$ を用意する. $P(u)$ を target density, $Q(u)$ を proposal density と呼ぶ. ただし, proposal density $Q(u)$ は, $P(u) > 0$ となる u の領域において $Q(u) > 0$ であることと, $Q(u)$ からサンプルを得ることが可能であることが条件である. この二つの条件さえ満たせば, proposal density $Q(u)$ は任意に設定できるが, 精度や効率の点で $P(u)$ に似た形状を持つ $Q(u)$ を用いることが望ましいとされている. ここでは target density を式 (5) で, proposal density を式 (4) で定義する. 初期値としての $Q(u)$ には, 正規分布やガンマ分布などのサンプル発生が容易な分布が用いられることが多い.

式 (4) から適当な方法で N 個のサンプルをとる：

$$s\{u_{t|t-1}^{(l)}\}_{l=1}^{N} \sim P(u_t \mid D_{t-1}, \mathcal{H}) \tag{6}$$

これをもとに importance weight $\Omega_t(u_{t|t-1}^{(l)})$ を計算する：

$$\begin{aligned} \Omega_t(u_{t|t-1}^{(l)}) &= \frac{P_t^*(u_{t|t-1}^{(l)})}{Q_t(u_{t|t-1}^{(l)})} \\ &= P(y_t \mid x_t, u_{t|t-1}^{(l)}, \mathcal{H}) \end{aligned}$$

更に次を計算する：

$$\begin{aligned} \bar{\Omega}_t(u_{t|t-1}^{(l)}) &= \frac{\Omega_t(u_{t|t-1}^{(l)})}{\sum_{l=1}^{N} \Omega_t(u_{t|t-1}^{(l)})} \\ &= \frac{P(y_t \mid x_t, u_{t|t-1}^{(l)}, \mathcal{H})}{\sum_{l=1}^{N} P(y_t \mid x_t, u_{t|t-1}^{(l)}, \mathcal{H})}, \quad (l = 1, \cdots, N) \end{aligned} \tag{7}$$

上記で計算された normalized importance weight $\bar{\Omega}_t(u^{(l)}_{t|t-1})$ を用いて, $Q(u)$ からのサンプル $\{u^{(l)}_{t|t-1}\}^N_{l=1}$ の中から, $P(u)$ の N' 個のサンプルとして $\{u^{(m)}_{t|t}\}^{N'}_{m=1}$ を得る:

$$P(u_t \mid D_t, \mathcal{H}) \approx \sum_{l=1}^{N} \{\bar{\Omega}_t(u^{(l)}_{t|t-1}) \times u^{(l)}_{t|t-1}\}$$

$$\{u^{(m)}_{t|t}\}^{N'}_{m=1} \sim P(u_t \mid D_t, \mathcal{H}), \qquad (N' \leq N)$$

これは, 確率 $\bar{\Omega}_t(u^{(l)}_{t|t-1})$ で $u^{(l)}_{t|t-1}$ を選択し, 選択された $u^{(l)}_{t|t-1}$ を $P(u)$ からサンプル $u^{(m)}_{t|t}$ として扱う作業と見なすこともできる. これらの作業を resampling と呼んでいる.

2.2.2 y_t の予測

式 (4) を用いると時刻 $t-1$ までのデータ D_{t-1} および時刻 t の入力 x_t が与えられた時, y_t の予測分布は以下のようになる.

$$P(y_t \mid x_t, D_{t-1}, \mathcal{H}) = \int P(y_t \mid x_t, u_t, \mathcal{H})\, P(u_t \mid D_{t-1}, \mathcal{H})\, du_t \tag{8}$$

式 (8) からのサンプルを得るため, $u^{(l)}_{t|t-1}$ を用いて $P(y_t \mid x_t, u^{(l)}_t, \mathcal{H})$ からサンプルをとり, y_t を予測する.

3 数値実験

3.1 実験 1

問題 : 観測値 y_t と入力 $x_t \in \mathbb{R}$ の入出力データ $(y_t, x_t), (t = 1, \cdots, 1000)$ が, 以下のように発生しているとする.

$$y_t = \frac{2.6 \sin(2x_t)}{1 + x_t^2} + \exp[-x_t(x_t - 2)] + \nu_t$$

$$x_t \sim \text{i.i.d. } U(-3, 3), \quad \nu_t \sim \text{i.i.d. } N(0, (0.1)^2)$$

このとき, 時刻 t において y_{t+1} を予測する. ただし, y_{t+1} の予測時には, 時刻 $t+1$ における入力 x_{t+1} は与えられているものとする.

3.1.1 条件

- σ_β, σ_γ は共に 0.01 とする.
- β_0, γ_0 は共に 100 とする.
- weight の初期分布の分散の逆数 α は 1.0 とする.
- 比較実験として, ハイパーパラメータ $\beta = 100$ に, $\gamma = 100,\ 500,\ 1000$ の三通りに固定したものを用意する.
- 全ての実験において, サンプル数は 1000, 用いた中間素子数は $h = 10$ とした.

3.1.2 結果

実際の予測値がどのようになっているかを図示する. $\gamma = 100$ で固定した場合, $\sigma_\beta = \sigma_\gamma = 0.01$ の場合の $t = 501$ 以降の予測値を, それぞれ図 1, 図 2 に示す. 図 1 と図 2 より, $\gamma = 100$ に比べると $\sigma_\beta = \sigma_\gamma = 0.01$ の場合のほうが, 真の関数 $f(\cdot)$ の周辺でのぶれが小さくなっており, より適切な予測をしていることが見てとれる.
また, 図 3 は, 二乗誤差の時間による変化を表している. 図 3 から, ハイパーパラメータの学習も行った場合 ($\sigma_\beta = \sigma_\gamma = 0.01$) のほうが, ハイパーパラメータを固定した場合より誤差が少ないことが示されている. このことは, 最初, ハイパーパラメータ (γ_t) が小さいため学習が早く, 次第に学習が進むにつれ, ハイパーパラメータが大きくなる事で予測が落ち着いていると考えられる. このことを示すのが, 次の ハイパーパラメータ (β_t, γ_t) の時間による変化であり, それを図 4 に表す. w_t の学習が十分進み, 出力関数 $f(\cdot \mid w^{(l)}_{t|t-1})$ が真の関数に近似できている状況となったら, それ以上 w_t は変化する必要が無く, γ_t は大きくなることが望ましいため, ハイパーパラメータ γ_t は適切に学習されていると言ってよい.

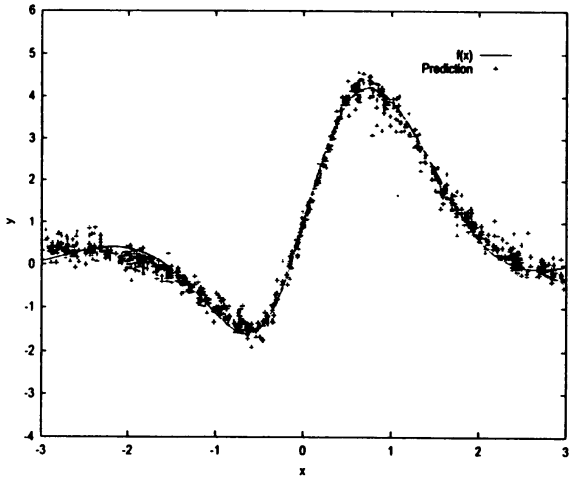

図 1: $t = 501$ 以降の予測値 ($\gamma = 100$)

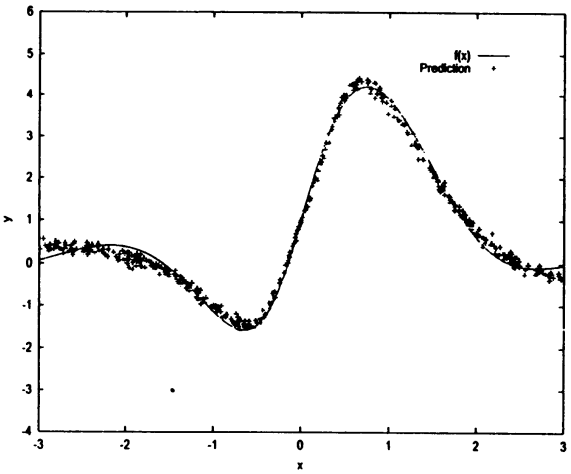

図 2: $t = 501$ 以降の予測値 ($\sigma_\beta = \sigma_\gamma = 0.01$)

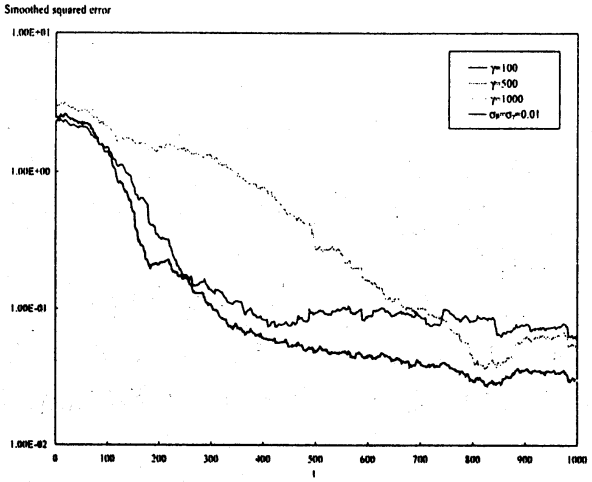

図 3: 二乗誤差の時間的変化

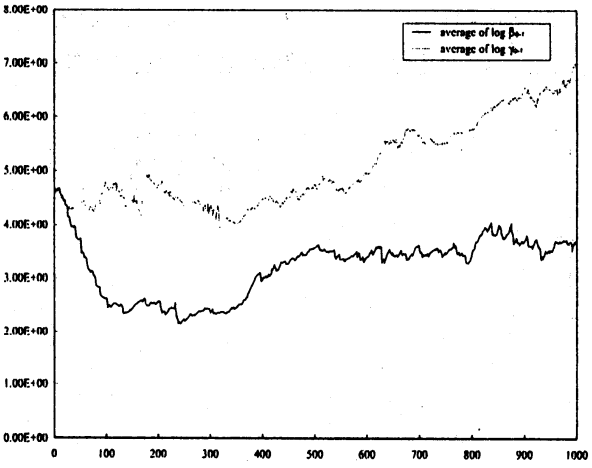

図 4: ハイパーパラメータの変化の様子

3.2 実験 2

今回の実験では, 真の関数が時間により変化しているもとでの予測の精度をみる.

問題 : 観測値 y_t と入力 $x_t \in \mathbb{R}$ の入出力データ $(y_t, x_t), (t = 1, \cdots, 2000)$ が, 以下のように発生しているとする.

$$y_t = \frac{2.6\sin(2x_t)}{1+x_t^2} + \exp[-x_t(x_t-2)]\frac{(\sin(\frac{2\pi t}{500})+1)}{2} + \nu_t,$$

$$x_t \sim i.i.d.\ U(-3,3), \quad \nu_t \sim i.i.d.\ N(0,(0.1)^2)$$

このとき, 時刻 t において y_{t+1} を予測せよ. ただし, y_{t+1} の予測時には, 時刻 $t+1$ における入力変数 x_{t+1} は与えられているものとする.

3.2.1 条件

- サンプル数は 100, 1000 の二通りで行い, 中間素子数は $h = 10$ とした.
- σ_β, σ_γ は共に 0.01 とする.
- β_0, γ_0 は共に 100 とする.
- weight の初期分布の分散の逆数 α は 1.0 とする.
- 比較実験として, ハイパーパラメータ $\beta = 100, \gamma = 100$ に固定したものを用意する.

3.2.2 結果

各実験の二乗誤差を図5に示す. $N = 100$ での予測は, あるレベルまで予測が落ち着くのに時間がかかり, 初めのうちの二乗誤差は減少が遅い. しかし, ハイパーパラメータを動かした $N = 100$ の予測は, 学習が進むにつれてハイパーパラメータを固定した $N = 1000$ の予測の二乗誤差とほとんど変わりない.

次にハイパーパラメータ β_t の動きを図6に, γ_t の動きを図7に示す. 図6からわかることは, β_t がいつまでも $\beta_t = 10 \sim 30$ 付近の値を示している. これはつまり, 実際に真の観測ノイズは分からないが, β_t が一定の値に近づかないことから, 真の関数は常に変化していると推測することができる.

また, γ_t に関しては図7から, $N = 100$ と $N = 1000$ では明らかに違いがある. $N = 100$ では最後まで γ_t が増加しなかったのに対し, $N = 1000$ のほうは, かなり大きい値をとっている. この結果から, 今回の関数の変化に対しては, その動きに十分に対応していることがわかる.

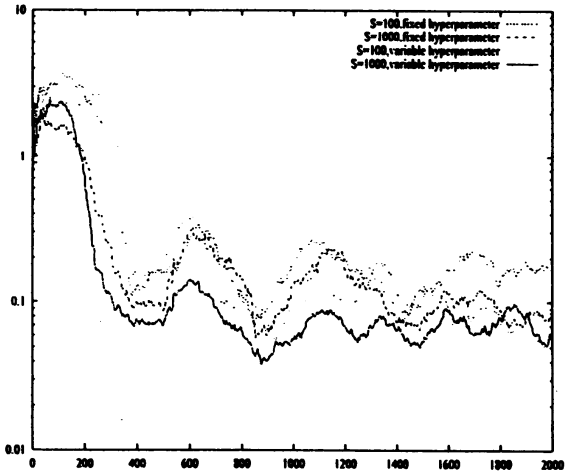

図 5: 二乗誤差の時間的変化 (実験 2)

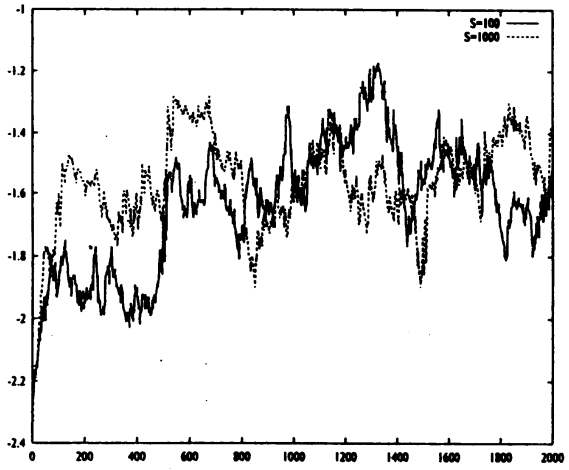

図 6: ハイパーパラメータ β_t の変化の様子

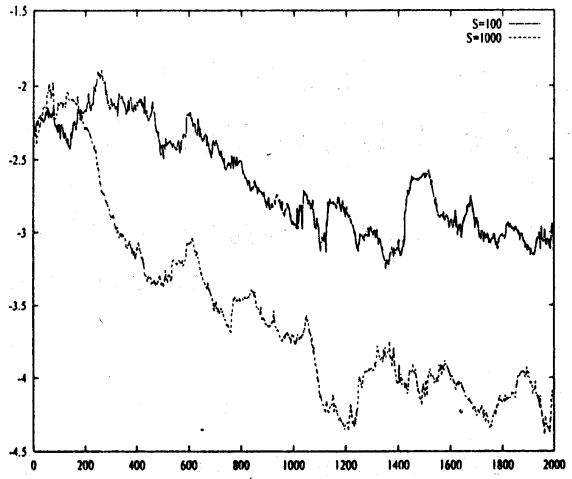

図 7: ハイパーパラメータ γ_t の変化の様子

参考文献

[1] T. Matsumoto, Y. Nakajima, M. Saito, J. Sugi and H. Hamagishi, "Bayesian Reconstructions and Predictions of Nonlinear Dynamical Systems", *IEEE Trans. Signal Processing*, in press

[2] Y. Nakada and T. Matsumoto, "Baysian MCMC nonlinear time series prediction : predictive mean and error bar," *Proc. 10th IEEE Workshop on Neural Networks for Signal Processing*, pp. 155-164, Sydney, Australia, Dec. 2000.

[3] J. F. G. de Freitas, M.Niranjan, A.H.Gee, Hierarchical Bayesian-Kalman models for regularisation and ARD in sequential learning, Technical report CUED/F-INFENG/TR 313, CambridgeUniversity Department of Engineering, 1998.

[4] J. F. G. de Freitas, M. Niranjan, A. H. Gee and A. Doucet, *Sequential Monte Carlo Methods for Optimisation of Neural Network Models*, Technical report CUED/F-INFENG/TR 313, Cambridge University Department of Engineering, 1998.

[5] D. J. C. MacKay, *Bayesian Methods for Adaptive Models*, PhD thesis, Calfornia Inst. Tech. 1991.

[6] R. M. Neal, *Bayesian Learning for Neural Networks*, Lecture Notes in statistics No. 118, Springer-Verlag, New York, 1996.

[7] R. M. Neal, "Annealed importance sampling", *Technical Report No. 9805*, Dept. of Statistics, University of Toronto, 1998.