

# モンテカルロ・フィルタおよび平滑化について

統計数理研究所 北 川 源 四 郎

(1995年11月 受付)

## 1. はじめに

1960年にKalmanにより提案された状態空間モデルとカルマンフィルタは工学の諸分野では広く利用されてきたが、統計科学において積極的な利用が開始されるようになったのは比較的に最近のことである。ARMAモデルを用いた統計的システム同定 (Akaike (1974)), Dynamic linear model (Harrison and Stevens (1976)) などがそのはしりと考えられるが、本格的に統計学者に用いられるのは、ようやく1980年代に入ってから非定常時系列のモデリングが重要な課題となってからと思われる。

しかし1980年代中ごろからは、統計科学の研究者達は従来の線形・ガウス型の状態空間モデルだけでは取扱えない問題に着目し、カルマンフィルタの拡張やモデルの改良を積極的に試みる様になってきた。例えば、West et al. (1985) や Smith and Miller (1986), Fahrmeir (1992) は一般化動的線形モデルとそのフィルタリングの方法を提案している。また、Sage and Mersa (1971), Meinhold and Singpurwalla (1989), Fahrmeir and Kaufmann (1991) や Schnatter (1992) などではさまざまな形のカルマンフィルタの拡張が、また Sage and Mersa (1971) や Masreliz (1975) では非ガウス型フィルタの近似が試みられている。一方、Kitagawa (1987) は状態空間モデルを非ガウス型に拡張し、逐次型のフィルタおよび平滑化のアルゴリズムを導き、数値計算によりほぼ厳密に計算する方法を提案した。時系列解析におけるさまざまな困難な問題、すなわち、パラメータの突然の変化、異常値、非ガウス型分布に従う時系列モデルによる解析、非線形モデルなどが非ガウス型フィルタおよび平滑化のアルゴリズムにより厳密に取り扱うことができることが示された (Kitagawa (1987, 1991))。

しかしながらこの数値的方法は、メモリに関してもまた計算時間に関しても計算機の集約的利用に基づいており、したがって高次元の問題への適用には限界がある。Hodges and Hale (1993), Tanizaki (1993) などにより計算量の軽減に関しても多くの研究が行われているが、本質的な解決にはなっていないように思われる。Kitagawa (1989, 1994) は低次元の問題に対して Alspach and Sorenson (1972) により開発されたガウス和フィルタを改良して非ガウス型季節調整法の開発を行なった。この方法により非ガウス型季節調整モデルを実用化することはできたが、アルゴリズムの複雑さから広範なモデルへの適用には限度があった。

この様な計算上の困難を解決し、多様な非線形・非ガウス型しかも高次元の状態空間モデルのフィルタリングおよび平滑化のためにモンテカルロ・フィルタと呼ばれる新しいアルゴリズムが開発された (Kitagawa (1993, 1996), 樋口 (1994, 1996))。この方法では、確率分布を多数の実現値で近似し、それぞれの粒子がモデルにしたがってどのように変化するかをシミュレートすることにより予測分布の計算を実現する。フィルタの段階では、予測分布に従って得られた粒子を、それぞれの粒子の“尤度”を重みとしてリサンプリング (復元抽出) することによりフィルタの分布を近似する粒子を求める。この方法は、モンテカルロ法に基づくものであるが Carlin et al. (1992) や Frühwirth-Schnatter (1994) によって開発されたギブス・サンプラーに基づく

く方法とは全く別のものである。なお、本稿作成後 Gordon et al. (1993) によりブートストラップフィルタと呼ばれる同様の方法が提案されていることが判明したので参考文献に追加しておく。ただし、この文献には平滑化の方法は与えられていない。

この解説では厳密な計算が可能な簡単な例を用いながら、モンテカルロ・フィルタおよび平滑化のアルゴリズムを詳しく説明する。

## 2. 非線形・非ガウス型の状態空間モデルと状態推定

時系列  $y_n$  に対して以下のような非線形・非ガウス型の状態空間モデル

$$(2.1) \quad x_n = F(x_{n-1}, v_n)$$

$$(2.2) \quad y_n = H(x_n, w_n),$$

を考えることにする。ただし  $x_n$  は  $k$  次元の状態ベクトルで、(2.1) はシステムモデルあるいは状態モデル、(2.2) は観測モデルと呼ばれる。システムノイズ  $v_n$  と観測ノイズ  $w_n$  は、それぞれ密度関数  $q(v)$  および  $r(w)$  に従う  $\ell$  次元および 1 次元の白色雑音とする。関数  $F$  および  $H$  は  $\mathbb{R}^k \times \mathbb{R}^\ell \rightarrow \mathbb{R}^k$  および  $\mathbb{R}^k \times \mathbb{R} \rightarrow \mathbb{R}$  の一般には非線形の関数である。ただし、状態  $x_n$  と観測値  $y_n$  が与えられると観測ノイズ  $w_n$  が一意に決まり、 $y_n$  に関して微分可能な関数  $G$  を用いて  $w_n = G(y_n, x_n)$  で与えられるものとする。また、初期状態ベクトル  $x_0$  は密度関数  $p_0(x)$  に従うものとする。この非線形・非ガウス型の状態空間モデルでは以下の例のように多様な線形あるいは非線形の、また、ガウス型あるいは非ガウス型の時系列モデルを統一的に取り扱うことができる。

1.  $F(x, v) = Fx + Gv$  および  $H(x, w) = Hx + w$  とすると、通常の線形状態空間モデルが得られる。この場合、 $w$  は  $G(y, x) = y - Hx$  により与えられ、 $\partial G / \partial y = 1$  となる。 $v$  および  $w$  はガウス型でも非ガウス型でもよい。
2.  $F(x, v) = F_1(x) + v$  および  $H(x, w) = H_1(x) + w$  とすると、状態とノイズが分離した非線形状態空間モデルが得られる。この場合  $G(y, x) = y - H_1(x)$  である。
3.  $F(x, v) = x + v$  および  $H(x, w) = e^x w$  とすると、 $e^x$  は観測ノイズのスケールパラメータとなり、時間とともに分散が変化する時系列のモデルが得られる。この場合、 $G(y, x) = ye^{-x}$ 、 $\partial G / \partial y = e^{-x}$  である。

状態空間モデルにおける最も重要な問題は状態  $x_n$  の推定である。パラメータ推定のための尤度計算、時系列の予測、欠測値の補間、非定常時系列の分解などの時系列解析の重要な問題がすべて状態推定の問題として統一的に取り扱うことができる。

状態推定の問題は条件つき分布  $p(x_n | Y_t)$  を求める問題として定式化することができる。ただし、 $Y_t \equiv \{y_1, \dots, y_t\}$  は時刻  $t$  までに得られた観測値の集合である。以下の三つの場合、すなわち  $n > t$ ,  $n = t$ ,  $n < t$  に対応して状態  $x_n$  の条件つき分布  $p(x_n | Y_t)$  はそれぞれ予測、フィルタおよび平滑化の分布と呼ばれる。

通常の線形ガウス型の状態空間モデルに対してはこれらの分布は正規分布となるので、その平均と分散共分散行列だけを求めればよく、カルマンフィルタや固定区間平滑化アルゴリズムにより求めることができる (Anderson and Moore (1979), 片山 (1983))。

しかしながら、非ガウス型状態空間モデルの場合は条件つき分布  $p(x_n | Y_t)$  は非ガウス型となるので平均と分散共分散行列だけでは、分布を特定することができない。一般的な分布を自由に扱うためには何らかの形で数値的表現を用いたフィルタや平滑化のアルゴリズムの開発が必要となる。代表的な例としては各時点ごとにひとつの正規分布で近似する拡張カルマンフィル

タ (Anderson and Moore (1979)), 多数の正規分布の混合で近似するガウス和フィルタ (Alspach and Sorenson (1972), Kitagawa (1989)), 少数のパラメータで表現される分布とその共役事前分布を用いる動的一般線形モデル (West et al. (1985)), 一般の分布を階段関数や折れ線などで近似する非ガウス型フィルタおよび平滑化 (Kitagawa (1987), Hodges and Hale (1993), Tanizaki (1993)) などがある。

図1は非ガウス型の分布をそれぞれの方法がどのように近似するかを示したものである。(a)は仮定した真の分布で二山からなっている。(b)はこれをひとつの正規分布で近似した場合で、拡張カルマンフィルタやモデルを線形近似した場合などに相当する。真の分布が二山以上の場合にはよい近似は得られない。(c)は分点の数が40の折れ線で近似した場合、(d)は階段関数で近似した

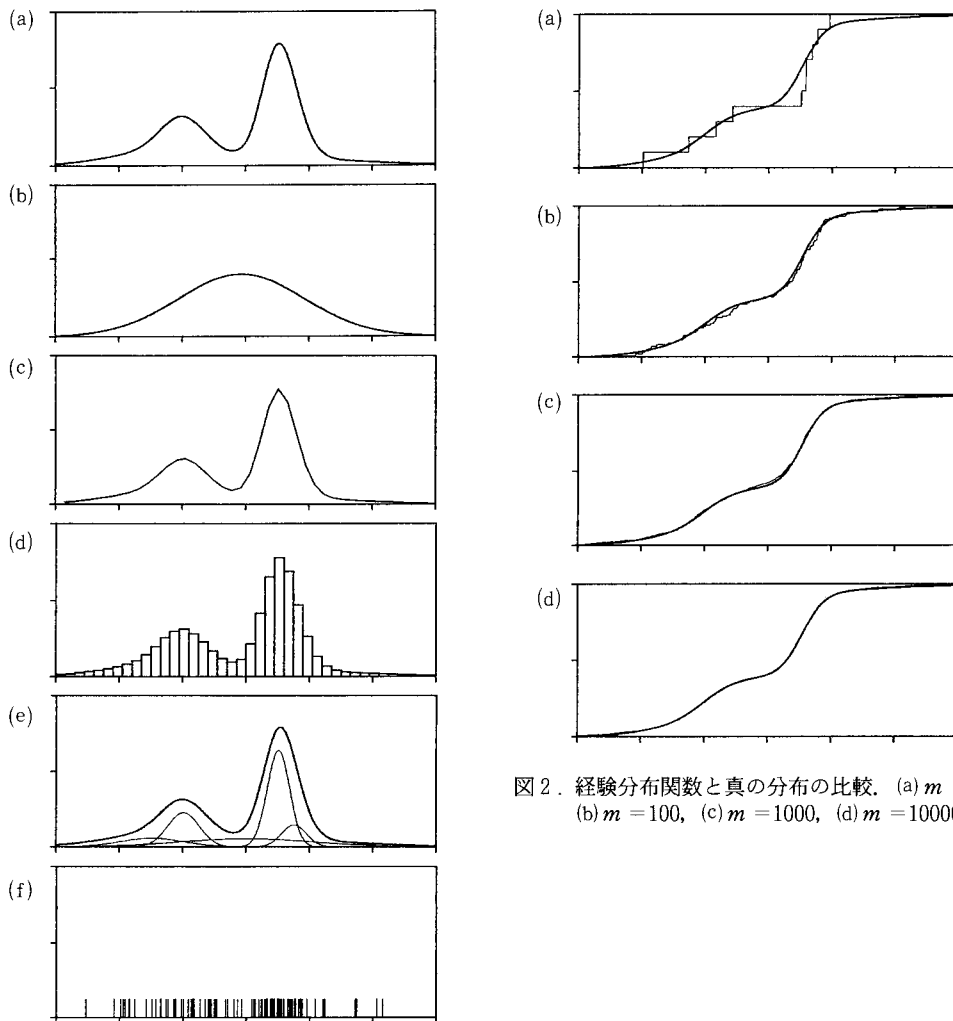


図2. 経験分布関数と真の分布の比較. (a)  $m = 10$ , (b)  $m = 100$ , (c)  $m = 1000$ , (d)  $m = 10000$ .

図1. 非ガウス型分布の近似法. (a)仮定した真の分布, (b)正規近似, (c)折れ線近似, (d)階段関数近似, (e)混合正規近似, (f)実現値による近似.

場合である。分点の数を大きく(数百程度に)とれば、任意の分布をかなりの精度で近似できる。非ガウス型フィルタ・平滑化は(c)や(d)の形の近似を用いている。(e)は5個の正規分布の混合で近似した場合である。細線はそれぞれの正規分布の寄与部分、太線はそれらの和として得られる混合正規分布を示す。ガウス和フィルタ・平滑化はこのような近似を用いている。

本稿で説明するモンテカルロ・フィルタの方法はこれらの近似法とは異なり、分布を多数の実現値を用いて表現する。図1(f)は100個の実現値による(a)の分布の近似である。それぞれの短い縦線が実現値の位置を表す。(a)の密度関数の高さが高いところには多くの実現値が集中している。図2はこのような実現値から作成した経験分布関数と真の分布関数の比較を行ったものである。(a), (b), (c), (d)の順に  $m = 10, 100, 1000, 10000$  の場合を示す。太線は真の分布関数、細線は経験分布関数を示す。 $m = 10$  のようにサンプル数が少ない場合でもある程度よく近似できていること、 $m$  の増加とともに真の分布に収束していることがわかる。

モンテカルロ・フィルタでは予測、フィルタ、平滑化の分布を  $m$  個(例えば、1000から10000個程度)の“粒子”を用いて以下のように近似する。

$$\begin{aligned} \{p_n^{(1)}, \dots, p_n^{(m)}\} &\sim p(x_n | Y_{n-1}) && \text{予測分布} \\ \{f_n^{(1)}, \dots, f_n^{(m)}\} &\sim p(x_n | Y_n) && \text{フィルタ分布} \\ \{s_{n|N}^{(1)}, \dots, s_{n|N}^{(m)}\} &\sim p(x_n | Y_N) && \text{平滑化分布} \\ \{v_n^{(1)}, \dots, v_n^{(m)}\} &\sim p(v_n) && \text{システムノイズの分布} \end{aligned}$$

これは、それぞれの分布関数を“粒子”から得られる経験分布関数によって近似することに相当する。以下の節では、これらの実現値がカルマンフィルタや非ガウス型フィルタと同様に逐次的に計算できることを示す。

### 3. モンテカルロ・フィルタ

本節では、予測およびフィルタの分布を近似する粒子を逐次的に生成するためのモンテカルロ・フィルタのアルゴリズムを示す。モンテカルロ・フィルタによれば予測分布に従う粒子  $\{p_n^{(1)}, \dots, p_n^{(m)}\}$  は一期前のフィルタの実現値  $\{f_{n-1}^{(1)}, \dots, f_{n-1}^{(m)}\}$  から生成でき、フィルタの実現値  $\{f_n^{(1)}, \dots, f_n^{(m)}\}$  は予測分布の実現値  $\{p_n^{(1)}, \dots, p_n^{(m)}\}$  のリサンプリングで生成することができる。

#### 3.1 一期先予測

状態  $x_{n-1}$  の条件つき分布  $p(x_{n-1} | Y_{n-1})$  の実現値とみなせる  $m$  個の粒子  $\{f_{n-1}^{(1)}, \dots, f_{n-1}^{(m)}\}$  およびシステムノイズ  $v_n$  の独立な実現値とみなせる粒子  $\{v_n^{(1)}, \dots, v_n^{(m)}\}$  が与えられているものとする。すなわち、 $i = 1, \dots, m$  について

$$(3.1) \quad f_{n-1}^{(i)} \sim p(x_{n-1} | Y_{n-1}), \quad v_n^{(i)} \sim q(v)$$

がなりたつものとする。このとき、 $x_n$  の一期先予測分布は

$$(3.2) \quad p(x_n | Y_{n-1}) = \iint p(x_n, x_{n-1}, v_n | Y_{n-1}) dx_{n-1} dv_n$$

$$(3.3) \quad = \int \int p(x_n | x_{n-1}, v_n, Y_{n-1}) p(v_n | x_{n-1}, Y_{n-1}) p(x_{n-1} | Y_{n-1}) dx_{n-1} dv_n$$

と表せるが、システムノイズはそれ以前の状態や観測値と独立であることから  $p(v_n | x_{n-1}, Y_{n-1})$

$= p(v_n)$ , また,  $x_n$  は  $x_{n-1}$  と  $v_n$  だけに依存することから  $p(x_n | x_{n-1}, v_n, Y_{n-1}) = p(x_n | x_{n-1}, v_n)$   
 $= \delta(x_n - F(x_{n-1}, v_n))$  がなりたつので

$$(3.4) \quad p(x_n | Y_{n-1}) = \int \int \delta(x_n - F(x_{n-1}, v_n)) p(v_n) p(x_{n-1} | Y_{n-1}) dx_{n-1} dv_n$$

が得られる. したがって,  $p(v_n)$  と  $p(x_{n-1} | Y_{n-1})$  の実現値  $\{v_n^{(j)}\}$  および  $\{x_{n-1}^{(j)}\}$  が得られている  
 ときには  $p(x_n | Y_{n-1})$  の実現値  $\{p_n^{(j)}\}$  が

$$(3.5) \quad p_n^{(j)} = F(f_{n-1}^{(j)}, v_n^{(j)})$$

により得られる.

### 3.2 フィルタ

次にフィルタのステップでは, まず観測値  $y_n$  に基づく粒子  $p_n^{(j)}$  の, “尤度”  $\alpha_n^{(j)}$  を計算する.  
 すなわち,  $j = 1, \dots, m$  について

$$\alpha_n^{(j)} = p(y_n | p_n^{(j)}) = r(G(y_n, p_n^{(j)})) \left| \frac{\partial G}{\partial y_n} \right|$$

とする. ただし, 2 節に示されているように  $G$  は関数  $H$  の逆関数,  $r$  は観測ノイズ  $w$  の密度関数である.  $\alpha_n^{(j)}$  は粒子  $p_n^{(j)}$  の重要度を表す重み係数と考えることができる.

次に,  $p_n^{(1)}, \dots, p_n^{(m)}$  を “尤度”  $\alpha_n^{(1)}, \dots, \alpha_n^{(m)}$  に比例した確率でリサンプリングを行うことにより  $m$  個の粒子  $f_n^{(1)}, \dots, f_n^{(m)}$  を求める. すなわち, 新しい粒子  $f_n^{(j)}$ , ( $j = 1, \dots, m$ ) を

$$(3.6) \quad f_n^{(j)} = \begin{cases} p_n^{(1)} & \text{確率 } \alpha_n^{(1)} / (\alpha_n^{(1)} + \dots + \alpha_n^{(m)}) \\ \vdots & \vdots \\ p_n^{(m)} & \text{確率 } \alpha_n^{(m)} / (\alpha_n^{(1)} + \dots + \alpha_n^{(m)}) \end{cases}$$

の復元抽出により求める. このとき以下に示すように  $\{f_n^{(1)}, \dots, f_n^{(m)}\}$  はフィルタ分布  $p(x_n | Y_n)$  からの実現値とみなすことができる.

分布  $p(x_n | Y_{n-1})$  の  $m$  個の独立な実現値  $p_n^{(1)}, \dots, p_n^{(m)}$  が与えられているとき, 分布  $p(x_n | Y_{n-1})$  は経験分布関数

$$(3.7) \quad P_n(x) = \frac{1}{m} \sum_{i=1}^m I(x, p_n^{(i)}),$$

により近似していることに相当する. ただし,  $I(x, a)$  は  $x < a$  のとき  $I(x, a) = 0$ , その他のとき  $I(x, a) = 1$  により定義される関数である. これは予測分布  $p(x_n | Y_{n-1})$  が確率関数

$$(3.8) \quad \Pr(x_n = p_n^{(j)} | Y_{n-1}) = \frac{1}{m}, \quad j = 1, \dots, m \quad \text{について}$$

により近似されることを意味している. このとき, 観測値  $y_n$  が与えられると, 事後分布が

$$(3.9) \quad \begin{aligned} \Pr(x_n = p_n^{(j)} | Y_n) &= \Pr(x_n = p_n^{(j)} | Y_{n-1}, y_n) \\ &= \lim_{\Delta y \rightarrow 0} \frac{\Pr(x_n = p_n^{(j)}, y_n \leq y \leq y_n + \Delta y | Y_{n-1})}{\Pr(y_n \leq y \leq y_n + \Delta y | Y_{n-1})} \end{aligned}$$

$$\begin{aligned}
&= \frac{p(y_n | p_n^{(j)}) \Pr(x_n = p_n^{(j)} | Y_{n-1})}{\sum_{i=1}^m p(y_n | p_n^{(i)}) \Pr(x_n = p_n^{(i)} | Y_{n-1})} \\
&= \frac{\alpha_n^{(j)} \cdot \frac{1}{m}}{\sum_{i=1}^m \alpha_n^{(i)} \cdot \frac{1}{m}} = \frac{\alpha_n^{(j)}}{\sum_{i=1}^m \alpha_n^{(i)}}
\end{aligned}$$

で求められる. この  $\Pr(x_n = p_n^{(j)} | Y_n)$  に対応する累積分布関数

$$(3.10) \quad \frac{1}{\sum_{i=1}^m \alpha_n^{(i)}} \sum_{i=1}^m \alpha_n^{(i)} I(x, p_n^{(i)})$$

は  $m$  個の点  $p_n^{(1)}, \dots, p_n^{(m)}$  だけで  $\alpha_n^{(1)}, \dots, \alpha_n^{(m)}$  に比例する量 (すなわち (3.9) 式の右辺) のジャンプをもつ.

この (3.10) 式によりフィルタの分布の近似が得られたことになるが, 次の時点で予測のステップの計算 (3.5) を行うためには, 等確率の  $m$  個の粒子  $f_n^{(1)}, \dots, f_n^{(m)}$  で表現しなおす必要がある. これは (3.10) の分布を経験分布関数

$$(3.11) \quad \frac{1}{m} \sum_{i=1}^m I(x, f_n^{(i)})$$

により表現することに相当するが,  $m$  個の実現値  $\{p_n^{(1)}, \dots, p_n^{(m)}\}$  を確率

$$(3.12) \quad \Pr(f_n^{(j)} = p_n^{(i)} | Y_n) = \frac{\alpha_n^{(i)}}{\alpha_n^{(1)} + \dots + \alpha_n^{(m)}}, \quad j = 1, \dots, m \quad \text{について}$$

でリサンプリングすることにより  $\{f_n^{(1)}, \dots, f_n^{(m)}\}$  を求めることができる.

### 3.3 リサンプリング法について

本節ではフィルタリングのステップで必要となるリサンプリングの方法について検討する. まず, ランダムサンプリングに基づく基本的なアルゴリズムを示す.

$j = 1, \dots, m$  について以下の(a)-(c)を繰り返す.

- (a) 一様乱数  $u_n^{(j)} \in U[0, 1]$  を生成する.
- (b)  $\frac{1}{C} \sum_{\ell=1}^{i-1} \alpha_n^{(\ell)} < u_n^{(j)} \leq \frac{1}{C} \sum_{\ell=1}^i \alpha_n^{(\ell)}$  をみたす  $i$  を探す. ただし,  $C = \sum_{\ell=1}^m \alpha_n^{(\ell)}$  である.
- (c) フィルタの粒子を  $f_n^{(j)} = p_n^{(i)}$  により求める.

ただし, 上記のリサンプリングの目的は (3.10) で定義された分布関数を, 重みが等しい経験分布関数の形で表現しなおすことなので, 厳密なランダムサンプリングを行なうことは必ずしも必要ではない. この点を考慮すると, リサンプリング法についてはソーティングと乱数の生成法に関して様々な変形が考えられる. ひとつの改良の方法は以下のように層化抽出を行なうことである. まず, 区間  $(0, 1]$  を  $m'$  個の同じ長さの小区間に分割する.  $m'$  の大きさをどうするかが問題となるが, 通常は  $m' = m$  とする. この場合, ステップ(a)は以下のふたつのいずれかで置き換えることになる.

- (a-S) 一様乱数  $u_n^{(j)} \sim U\left(\frac{j-1}{m}, \frac{j}{m}\right]$  を生成する.

(a-D) 固定した  $\alpha \in (0, 1]$  について  $u_n^{(j)} = \frac{j-\alpha}{m}$  とする.

$\alpha$  としては  $1/2$  を用いたり乱数で求めることが考えられる. 文字どおり, 層化抽出を行うためには, まず  $p_n^{(1)}, \dots, p_n^{(m)}$  を大きさの順に並べなおし,  $\alpha_n^{(1)}, \dots, \alpha_n^{(m)}$  もそれに対応して入れかえたあと上記の抽出を行う必要がある. しかしながら, ソーティングは, とくに粒子数  $m$  が大きいときモンテカルロ・フィルタの中で最も時間がかかる演算となる. HEAPSORT アルゴリズム (Press et al. (1992)) でも約 40MFLOPS の計算機で  $m = 10, 100, 1000$  および 10000 のときそれぞれ, 0.0001, 0.0005, 0.020, 1.736 秒を要した. さらにこの方法は多次元の場合には適用できないこと, および, 層化抽出の目的を重みの合計が  $1/m$  の粒子のグループからちょうど 1 個の粒子が抽出されるようにするためと考えると, 必ずしも順序による並べかえを行う必要はないように思われる.

Kitagawa (1996) ではいろいろなサンプリング法を用いた場合の近似の良さを評価している. この結果によれば, 忠実にランダムサンプリングを行うよりも, 層化を行った方がはるかに良い近似が得られる. 層化を行う二つの方法の中では, (a-D) の確定的な方法が層化抽出法 (a-S) よりも少し良い結果が得られている. また, ソートが行われた場合には, どちらの方法でも  $O(m^{-2})$  の収束が見られるが, ソートを行わない場合には  $O(m^{-1})$  の収束であるように見える. したがって, ソートを行なって確定的抽出を行った場合が最も優れている. ただし, ソートには多大な計算時間がかかることと,  $\{p_n^{(j)}, \alpha_n^{(j)}\}$  で決まる分布自体がある程度の近似誤差を含んでいるので, それをあまり忠実に再現しても意味がないことを考慮するとソートを行わない確定的抽出法が最も実用的であると思われる (Higuchi (1995a)).

### 3.4 モンテカルロ・フィルタのアルゴリズム

以上の結果をまとめると以下のようなモンテカルロ・フィルタのアルゴリズムが得られる (図 3 参照).

1.  $j = 1, \dots, m$  について  $k$  次元の乱数  $f_0^{(j)} \sim p_0(x)$  を生成する.
2.  $n = 1, \dots, N$  について以下のステップを実行する.
  - (a)  $j = 1, \dots, m$  について  $\ell$  次元の乱数  $v_n^{(j)} \sim q(v)$  を生成する.
  - (b)  $j = 1, \dots, m$  について  $p_n^{(j)} = F(f_{n-1}^{(j)}, v_n^{(j)})$  を計算する.

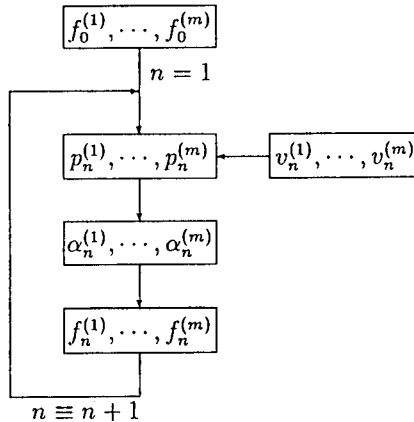


図 3. モンテカルロ・フィルタのアルゴリズム.

- (c)  $j = 1, \dots, m$  について  $\alpha_n^{(j)} = r(G(y_n, p_n^{(j)})) |\partial G / \partial y_n|$  を計算する.  
 (d)  $j = 1, \dots, m$  について  $p_n^{(1)}, \dots, p_n^{(m)}$  のリサンプリングにより  $f_n^{(j)} \sim (\sum_{i=1}^m \alpha_n^{(i)})^{-1} \sum_{i=1}^m \alpha_n^{(i)} I(x, p_n^{(i)})$  を生成する.

### 3.5 モデルの尤度

状態空間モデル (2.1), (2.2) は通常, ノイズの分散や  $F$  や  $H$  の係数など未知のパラメータを含んでいる. それらのパラメータを  $\theta$  と表すことにする. 観測値  $y_1, \dots, y_N$  が与えられるとき, パラメータ  $\theta$  の尤度は以下の式により求めることができる.

$$(3.13) \quad L(\theta) = p(y_1, \dots, y_N | \theta) = \prod_{n=1}^N p(y_n | y_1, \dots, y_{n-1}, \theta) = \prod_{n=1}^N p(y_n | Y_{n-1}),$$

ただし,  $p(y_1 | Y_0) = p_0(y_1)$  とする. したがって, 以下の近似

$$(3.14) \quad \begin{aligned} p(y_n | Y_{n-1}) &= \int p(y_n | x_n) p(x_n | Y_{n-1}) dx_n \\ &\cong \frac{1}{m} \sum_{j=1}^m p(y_n | p_n^{(j)}) \\ &= \frac{1}{m} \sum_{j=1}^m \alpha_n^{(j)}, \end{aligned}$$

を用いると対数尤度は

$$(3.15) \quad \ell(\theta) = \sum_{n=1}^N \log p(y_n | Y_{n-1}) \cong \sum_{n=1}^N \log \left( \frac{1}{m} \sum_{j=1}^m \alpha_n^{(j)} \right) - N \log m.$$

と近似することができる. パラメータの最尤推定値  $\hat{\theta}$  はこの対数尤度を数値的に最大化することによって求めることができる.

ただし, モンテカルロ・フィルタで求めた対数尤度にはモンテカルロ法に由来する近似誤差が存在することに注意する必要がある. Higuchi (1995a) には, この誤差の分布などの報告がある. この問題の解決法としては, パラメータ  $\theta$  も状態ベクトルとみなして同時に推定する自己組織型のモデリング (Kitagawa (1995), Higuchi (1995b)) が提案されている.

### 3.6 数値例

一期先の予測分布とフィルタ分布が“粒子”によりどの様に近似されるかを示すために, 1 ステップだけのモンテカルロ・フィルタの結果 ( $n = 1$ ) を示す. 以下のような1次の線形・非ガウス型モデルを考えることにする.

$$(3.16) \quad x_n = x_{n-1} + v_n$$

$$(3.17) \quad y_n = x_n + w_n,$$

ただし,  $v_n$  と  $w_n$  はそれぞれ, 密度関数が  $q(v) = 0.1\pi^{-1}(v^2 + 0.01)^{-1}$  で与えられるコーシー分布および標準正規分布  $N(0, 1)$  に従う白色雑音である. また, 初期分布  $p_0(x_0)$  も標準正規分布に従うものと仮定する. いうまでもなく, モンテカルロ・フィルタでは正規性の仮定は本質的ではない. 以上の条件のもとで, 予測分布  $p(x_1 | Y_0)$  およびフィルタ分布  $p(x_1 | Y_1)$  を求めた. 粒子数は  $m = 100$  とした.  $m = 100$  は図を見やすくするためであり, 通常の計算ではもっと大きな値を用いる.

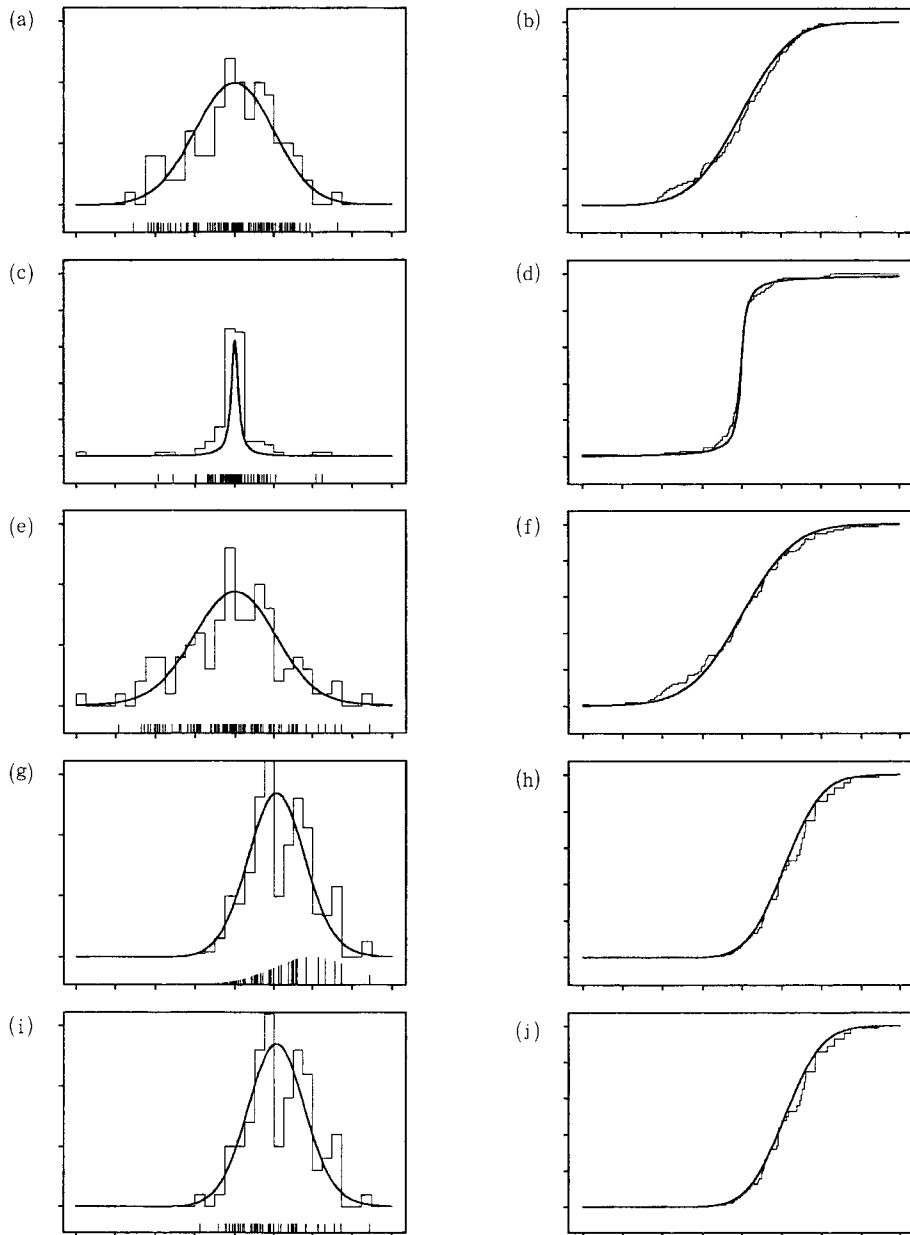


図4. 1ステップのモンテカルロ・フィルタ. (a)初期分布の密度関数, 実現値およびヒストグラム, (b)初期分布の分布関数および経験分布関数, (c)システムノイズの密度関数, 実現値およびヒストグラム, (d)システムノイズの分布関数および経験分布関数, (e)予測分布の密度関数, 実現値およびヒストグラム, (f)予測分布の分布関数および経験分布関数, (g)フィルタ分布の密度関数と重み付きの実現値とヒストグラム, (h)フィルタ分布の分布関数および経験分布関数, (i)リサンプリング後の実現値とヒストグラム, (j)リサンプリング後の経験分布関数.

図4(a)の実線は仮定した初期分布 $p_0(x_0)$ を示す。一方、たて棒はこの分布から生成した100個の実現値の場所、棒状の関数はそのヒストグラムを表す。一方(b)はこれに対応して太線は真の分布関数、細線は(a)の実現値から得られた経験分布関数を示す。同様に、(c)、(d)はシステムノイズ $v_1$ の密度関数、実現値などを示す。

(e)、(f)は予測分布 $p(x_1|Y_0)$ に関するものである。(e)の曲線は(a)と(c)の密度関数から数値積分により求めた“真”の密度関数である。(f)の実線はこれをさらに積分して求めた“真”の分布関数である。一方、(e)のたて棒は(a)、(c)の実現値を(3.5)式に代入して求めた $p_1^{(j)}$ の位置を示す。そのヒストグラムは真の関数の近似となっている。経験分布関数と真の分布関数との関係は(f)に示す。

(g)の曲線は $y_1 = 2$ という観測値が得られたと仮定したときのフィルタの分布で、非ガウス型フィルタの式 (Kitagawa (1987)) に従って求めたものである。(g)の粒子の位置は(e)と同一であるが直線の高さは尤度 $\alpha_n^{(j)}$ に比例している。したがって、ステップ幅が $\alpha_n^{(j)}$ に比例した(3.10)で定義される(h)の累積分布関数は、ステップの場所が(f)と同一であるにもかかわらずステップ幅が一定の(f)とは全く異なりフィルタ分布を近似している。(i)はリサンプリング後のヒストグラムと粒子の位置および厳密なフィルタ分布を表す。(i)の密度関数および(j)の経験分布関数はともに、それぞれ(g)、(h)のよい近似となっていることがわかる。

以下の例は図5(a)の人工的に作成したデータ (北川 (1993)) に対して1次のトレンドモデル

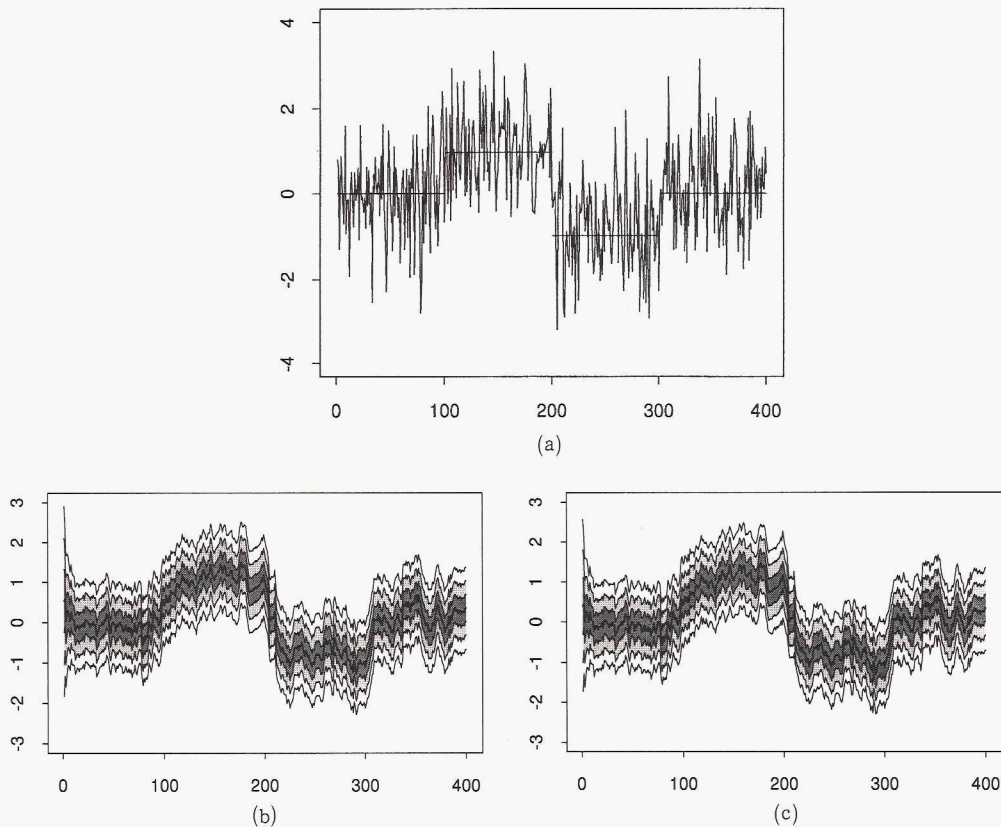


図5. モンテカルロ・フィルタの結果. (a)テストデータ, (b)カルマンフィルタによる厳密な値, (c)モンテカルロ・フィルタによる結果 ( $m = 10000$ ).

$$(3.18) \quad x_n = x_{n-1} + v_n$$

$$(3.19) \quad y_n = x_n + w_n$$

を想定してモンテカルロ・フィルタにより、状態推定を行った場合の結果である。 $w_n$  は平均0、分散 $\sigma^2$ の正規分布と仮定するが、システムノイズについては正規分布の場合とコーシー分布の二つの場合を考える。

図5(b)はシステムノイズが正規分布の場合にカルマンフィルタで求めた厳密なフィルタ分布を示す。中心の太線は分布の平均値、二つのグレイゾーンはそれぞれ $\pm 1\sigma$ 、 $\pm 2\sigma$ の幅を示す。一方、(c)は $m = 10000$ としてモンテカルロ・フィルタで求めたものである。厳密なものとの区別がつかないほどよい近似が得られている。

#### 4. 平滑化

モンテカルロ・フィルタの方法は原理的には簡単に平滑化の場合に拡張することができる。本節では2つの方法を示す。

##### 4.1 過去の粒子を保存する方法

まず、前節のモンテカルロ・フィルタのアルゴリズムの簡単な修正により、平滑化が実現できることを示す。本節では、 $(s_{1|n}^{(j)}, \dots, s_{n|n}^{(j)})^t$  は同時分布 $p(x_1, \dots, x_n | Y_n)$ の $j$ 番目の実現値を示すものとする。

$\Pr(x_1 = s_{1|n-1}^{(j)}, \dots, x_{n-1} = s_{n-1|n-1}^{(j)} | Y_{n-1}) = 1/m$  および  $v_n^{(j)} \sim q(v)$  がなりたつものとし、 $(p_{1|n-1}^{(j)}, \dots, p_{n|n-1}^{(j)})$  を以下のように定義する。

$$(4.1) \quad p_{i|n-1}^{(j)} = \begin{cases} s_{i|n-1}^{(j)}, & i = 1, \dots, n-1 \text{ について} \\ F(s_{n-1|n-1}^{(j)}, v_n^{(j)}), & i = n \text{ について} \end{cases}$$

このとき $(p_{1|n-1}^{(j)}, \dots, p_{n|n-1}^{(j)})$  は観測値 $Y_{n-1}$  が与えられたときの $(x_1, \dots, x_n)$ の同時分布の実現値とみなすことができる。

次に観測値 $y_n$  が与えられるとき、分布 $\Pr(x_1 = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)} | Y_{n-1})$  は次のように更新される。

$$\begin{aligned} (4.2) \quad \Pr(x_1 = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)} | Y_n) \\ &= \Pr(x_1 = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)} | Y_{n-1}, y_n) \\ &= \frac{p(y_n | x_1 = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)}) \Pr(x_1 = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)} | Y_{n-1})}{p(y_n | Y_{n-1})} \\ &= \frac{p(y_n | p_{n|n-1}^{(j)}) \Pr(x_1 = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)} | Y_{n-1})}{p(y_n | Y_{n-1})}. \end{aligned}$$

$p_{n|n-1}^{(j)}$  はフィルタのアルゴリズム (3.7) の $p_n^{(j)}$  と同一なので、平滑化の分布 $p(x_1, \dots, x_n | Y_n)$  は $m$  個の $n$  次元ベクトル $(p_{1|n-1}^{(j)}, \dots, p_{n|n-1}^{(j)})^t$ ,  $j = 1, \dots, m$  をフィルタの場合と同じ重みでリサンプリングすればよいことになる。

したがって、平滑化のためには3.4節のアルゴリズムのステップ2(d)を以下のように変更するだけでよい。

(d-S)  $j = 1, \dots, m$  について  $(s_{1|n-1}^{(j)}, \dots, s_{n-1|n-1}^{(j)}, p_n^{(j)})'$  のリサンプリングにより  $(s_{1|n}^{(j)}, \dots, s_{n-1|n}^{(j)}, s_{n|n}^{(j)})'$  を生成する.

この修正では  $\{(s_{1|n-1}^{(j)}, \dots, s_{n-1|n-1}^{(j)}, p_n^{(j)})', j = 1, \dots, m\}$  を3.4節のステップ2(d)の重みでリサンプリングする. これだけで非線形・非ガウス型モデルの固定区間平滑化を実現することができる. ただ

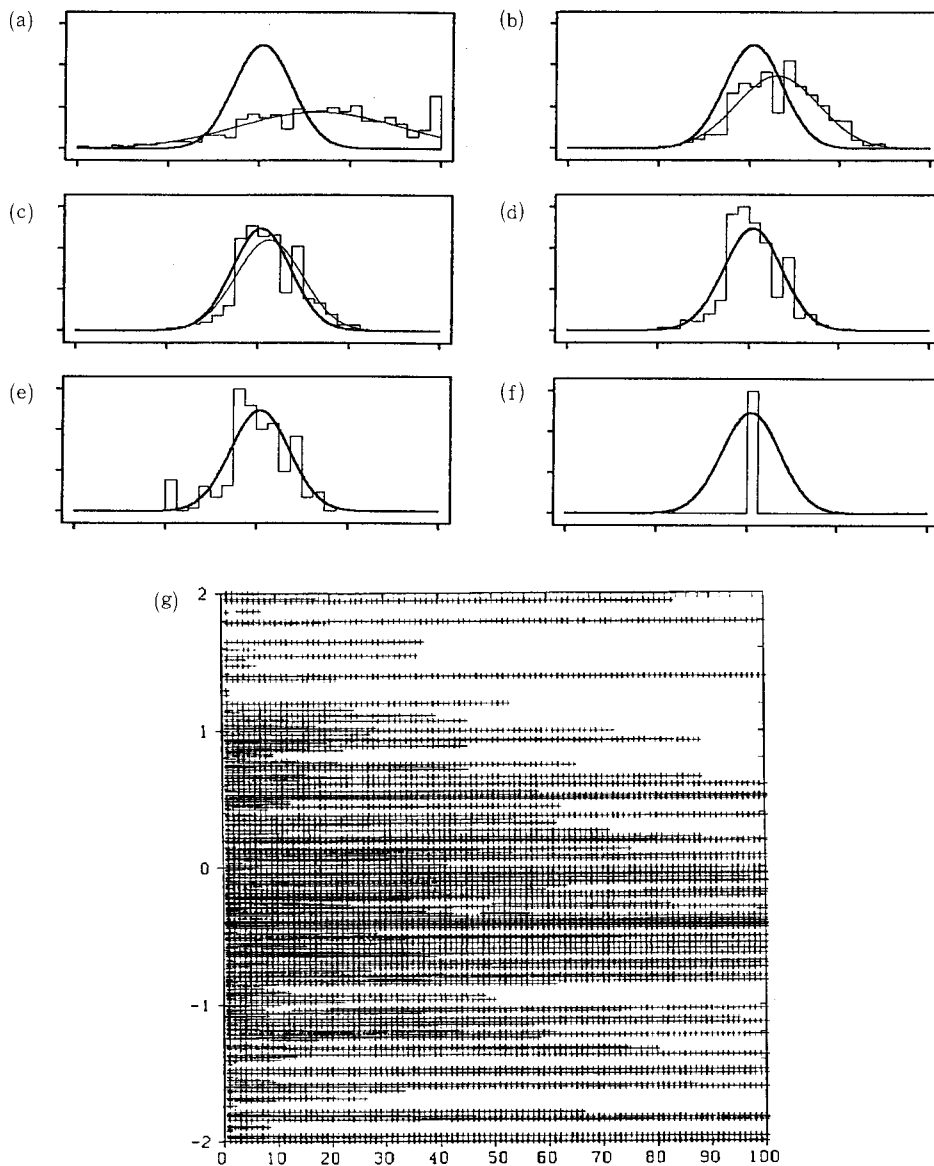


図6. モンテカルロ・平滑化: 細線:  $p(x_t|Y_n)$ , 太線:  $p(x_t|Y_{500})$  およびヒストグラム.  
(a)  $n = 1$ , (b)  $n = 5$ , (c)  $n = 10$ , (d)  $n = 20$ , (e)  $n = 100$ , (f)  $n = 500$ , (g)  $n$  の増加による粒子の変化.

し、以下に示すように、実際の計算は有限個（ $m$  個）のサンプルを用いて計算するために、実際に異なる粒子の数は単調非増加となり、リサンプリングを繰り返していくと、少数の粒子が繰り返し出現し、やがて分布の形が損なわれることになる。

図6は図5と同じデータに対して平滑化を行った結果である。 $v_n$ は正規分布に従うと仮定している。(a)-(f)に $n = 1, 5, 10, 20, 100$ および500の場合カルマンフィルタと平滑化により求めた正確な固定点平滑化分布 $p(x_1|Y_n)$ （細線）と最終的な固定区間平滑化分布 $p(x_1|Y_{500})$ （太線）を示す。 $n$ が増加するとき $p(x_1|Y_n)$ がすばやく $p(x_1|Y_{500})$ に収束していることがわかる。一方、ヒストグラムは1000個の粒子 $\{x_{1|n}^{(1)}, \dots, x_{1|n}^{(1000)}\}$ から得られたものである。(a)の $n = 1$ の場合はヒストグラムは $p(x_1|Y_1)$ のよい近似であるが $p(x_1|Y_{500})$ からは遠く隔たっている。一方 $n = 100$ ではヒストグラムは $p(x_1|Y_n)$ とやや異なっており、さらに $n = 500$ では中心の区間に集中している。実際、 $m = 1000$ のときの計算例では $n = 50$ では84個、 $n = 100$ では51個の粒子に集中していた。(g)は $p(x_1|Y_n)$ を近似する粒子の位置を表示したものである。横軸は $n$ の値を示す。 $n$ の増加とともに急速に減少していることがわかる。

以上の結果を考慮すると平滑化のアルゴリズムで(d-S)を以下のように変更することが考えられる。

(d-L)  $j = 1, \dots, m$  について  $(s_{n-L|n-1}^{(j)}, \dots, s_{n-1|n-1}^{(j)}, p_n^{(j)})$  のリサンプリングにより  $(s_{n-L|n}^{(j)}, \dots, s_{n-1|n}^{(j)})$  を生成する。ただし、 $f_n^{(j)} = s_{n|n}^{(j)}$  で  $L$  は固定した値とする。

興味深いことに、この修正したアルゴリズムは、 $L$  ラグの固定ラグ平滑化を行っていることに相当する。図6に示したように、 $L$  を大きくすると、 $p(x_n|Y_{n+L})$  は  $p(x_n|Y_n)$  をよく近似するが、 $\{x_{n|n+L}^{(1)}, \dots, x_{n|n+L}^{(m)}\}$  の分布は  $p(x_n|Y_{n+L})$  からかえって離れていく。したがって、 $L$  はあまり大きくないほうがよいことになる。通常は20程度、最大でも50以下にするほうがよい (Kitagawa (1996))。

この固定ラグ平滑化のアルゴリズムは計算機のメモリの点でも有利である。もともとのアルゴリズムでは  $m \times k \times n$  のメモリを要するが、後者では  $m \times k \times L$  だけでよい。例えば、 $m = 10000$ 、 $k = 10$ 、 $n = 1000$ 、 $L = 20$  の場合、前者では100メガとなるが、後者では2メガだけでよい。

図7(a)と(b)に非ガウス型平滑化により厳密に求めた  $p(x_n|Y_n)$  の分布を示す。(a)は  $v_n$  に正規分布を用いた場合、(b)はコーシー分布を用いた場合である。(c)、(d)、(e)にはそれぞれ、ガウス型モデルで  $m = 100, 1000, 10000$  とした場合の平滑化分布を示す。 $L = 25$  とし、図の太線は平均、濃いグレイゾーンは  $\pm 1\sigma$  の範囲、淡いグレイゾーンは  $\pm 2\sigma$  の範囲を示している。(c)の  $m = 100$  の場合には平均は(a)の比較的よい近似となっているが、分布自体の近似はよくないことがわかる。(d)の  $m = 1000$  の場合にはかなりよい近似が得られている。(e)の  $m = 10000$  の場合には  $\pm 3\sigma$  の範囲まで示している。 $\pm 3\sigma$  の点は少し変動が大きい、その他の点は極めてよい近似が得られている。(f)にはコーシー分布モデルで  $m = 10000$  とした場合の結果を示す。以上から分布の50%点はかなり小さな  $m$  でも(a)、(b)の厳密な値の比較的よい近似値が得られているが他のパーセント点を推定するには  $m$  を大きくとらないといけないことがわかる。

この平滑化分布の近似をよくする方法の一つとして、以下のような修正が考えられる。これまでのアルゴリズムでは、予測分布の計算において各  $f_n^{(j)}$  に対して、1個のシステムノイズ  $v_n^{(j)}$  を用いて粒子  $p_n^{(j)}$  を1個だけ発生させていた。これに対して、 $d$  個のノイズ  $v_n^{(j,i)} \sim q(v)$ 、 $i = 1, \dots, d$  を発生させ  $d$  個の粒子  $p_n^{(j,i)}$  を求めることもできる。このとき  $m \times d$  個の粒子で予測分布を近似することになる (樋口 (1994))。フィルタの段階ではリサンプリングで  $m$  個の粒子を作ればよい。

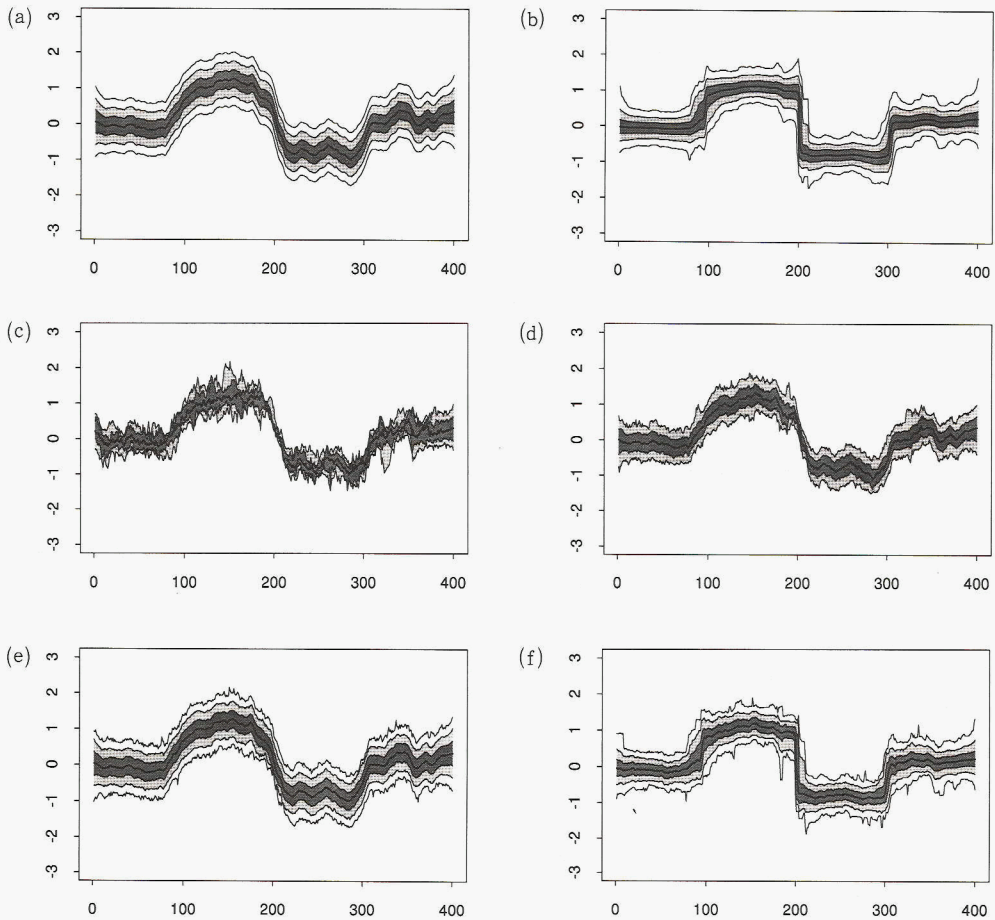


図7. モンテカルロ平滑化. (a)厳密な平滑値 (ガウスモデルの場合), (b)厳密な平滑値 (コーシーモデルの場合), (c)モンテカルロ平滑化 (ガウスモデル,  $m = 100$ ), (d)モンテカルロ平滑化 (ガウスモデル,  $m = 1000$ ), (e)モンテカルロ平滑化 (ガウスモデル,  $m = 10000$ ), (f)モンテカルロ平滑化 (コーシーモデル,  $m = 10000$ ).

#### 4.2 二方向フィルタによる平滑化

もう一つの平滑化の方法はKitagawa (1994) でも用いられた二方向フィルタを利用するものである. この方法では, 前向きと後向きのフィルタを行い, この二つの結果を結合して平滑化を行う. おもしろいことに, 後向きフィルタで得られる  $p(Y_N^n | x_{n-1})$  は通常のフィルタリングにおける状態の尤度  $p(y_n | x_n)$  と同様の役割を果たす.

$Y_N^n = \{y_n, \dots, y_N\}$  とする. このとき  $Y_N = Y_{n-1} \cup Y_N^n$  がなりたつ. したがって平滑化分布は以下の式により求められる.

$$\begin{aligned}
 (4.3) \quad p(x_n | Y_N) &= p(x_n | Y_{n-1}, Y_N^n) \\
 &\propto p(x_n, Y_N^n | Y_{n-1}) \\
 &= p(Y_N^n | x_n) p(x_n | Y_{n-1}).
 \end{aligned}$$

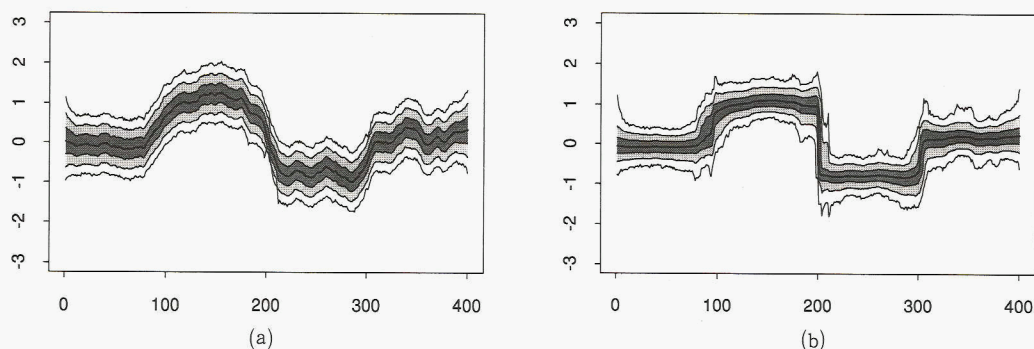


図8. 二方向フィルタによるモンテカルロ平滑化. (a)ガウス型モデル,  $m = 10000$ ,  
(b) コーシーモデル,  $m = 10000$ .

ここで  $p(Y_N^n | x_n)$  は以下により評価することができる.

$$\begin{aligned}
 p(Y_N^n | x_N) &= p(y_N | x_N) \\
 (4.4) \quad p(Y_N^{n+1} | x_n) &= \int p(Y_N^{n+1} | x_{n+1}) p(x_{n+1} | x_n) dx_{n+1} \\
 p(Y_N^n | x_n) &= p(y_n | x_n) p(Y_N^{n+1} | x_n).
 \end{aligned}$$

後向きのフィルタの公式は少なくとも形式的には前向きのフィルタ公式と同様であるので、モンテカルロ法にもとづいて二方向フィルタ公式による平滑化を行うことができる. すなわち、モンテカルロ・フィルタの方法により前向きと後向きのフィルタに対応する粒子を生成し、(4.3)式に相当する計算を行えばよい. 詳しくは Kitagawa (1996) を参照.

図8(a)と(b)は  $m = 10000$  とし、二方向フィルタによって求めた平滑化分布である. 図7の粒子を保存する方法により求めた結果よりはるかに滑らかで、厳密な値(a), (b)により近い図が得られている.

このように一次元状態空間モデルの場合には二方向フィルタの方法では極めてよい結果が得られるが、二次元以上の状態空間モデルへの適用には問題があるようであり、今後の研究が必要である. 現在のところ高次元状態空間モデルに対しては過去の粒子を保存する方法が用いられている.

## 5. まとめ

分布を多数の粒子で近似することにより、任意の非線形・非ガウス型状態空間モデルに対するフィルタおよび平滑化のアルゴリズムを構成することができる. この方法は乱数を用いた方法なので非ガウス型フィルタおよび平滑化のアルゴリズムほど厳密な分布は求められないが、高次元のシステムに適用でき、しかも複雑な非線形モデルでも簡単に適用できるところに特徴がある.

本稿はカルマンフィルタや非ガウス型フィルタ、平滑化のアルゴリズムによれば厳密な分布が得られる簡単な例を利用して、モンテカルロ・フィルタがどのように働くかを詳しく解説した.

## 謝 辞

本研究の過程で貴重なコメントをいただいたハワイ大学の Prof. Will Gersch および統計数理研究所の樋口知之氏に感謝する.

## 参 考 文 献

- Akaike, H. (1974). Markovian representation of stochastic processes and its application to the analysis of autoregressive moving average processes, *Ann. Inst. Statist. Math.*, **26**, 367-387.
- Alspach, D.L. and Sorenson, H.W. (1972). Nonlinear Bayesian estimation using Gaussian sum approximations, *IEEE Transactions on Automatic Control*, **17**, 439-448.
- Anderson, B. D. O. and Moore, J. B. (1979). *Optimal Filtering*, Prentice-Hall, New Jersey.
- Carlin, B.P., Polson, N.G. and Stoffer, D.S. (1992). A Monte Carlo approach to nonnormal and nonlinear state-space modeling, *J. Amer. Statist. Assoc.*, **87**(418), 439-500.
- Fahrmeir, L. (1992). Posterior mode estimation by extended Kalman filtering for multivariate dynamic generalized linear models, *J. Amer. Statist. Assoc.*, **87**(418), 501-509.
- Fahrmeir, L. and Kaufmann, H. (1991). On Kalman filtering, posterior mode estimation and Fisher scoring in dynamic exponential family regression, *Metrika*, **38**, 37-60.
- Frühwirth-Schnatter, S. (1994). Data augmentation and dynamic linear models, *J. Time Ser. Anal.*, **15**, 183-202.
- Gordon, N.J., Salmond, D.J. and Smith, A.F.M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEE Proceedings-F*, **140**(2), 107-113.
- Harrison, P.J. and Stevens, C.F. (1976). Bayesian forecasting (with discussion), *J. Roy. Statist. Soc. Ser. B*, **34**, 1-41.
- 樋口知之 (1994). モンテカルロフィルタ (MCF) におけるランダムサンプリング, ハイパフォーマンスコンピューティング, **54**(2), 7-14.
- 樋口知之 (1996). 遺伝的アルゴリズムとモンテカルロフィルタ, 統計数理, **44**, 19-30.
- Higuchi, T. (1995a). On the resampling scheme in the filtering procedure of the Monte Carlo filter, *Research Memo.*, No. 556, The Institute of Statistical Mathematics, Tokyo.
- Higuchi, T. (1995b). An analysis of time-varying frequency wave by using Self-ORGanizing Time Series Model (SORTism), 統計数理研究所共同研究レポート79, 「時系列解析の理論と応用」, 37-50.
- Hodges, P.E. and Hale, D.F. (1993). A computational method for estimating densities of non-Gaussian nonstationary univariate time series, *J. Time Ser. Anal.*, **14**(2), 163-178.
- 片山 徹 (1983). 『応用カルマンフィルタ』, 朝倉書店, 東京.
- Kitagawa, G. (1987). Non-Gaussian state-space modeling of nonstationary time series (with discussion), *J. Amer. Statist. Assoc.*, **82**, 1032-1063.
- Kitagawa, G. (1989). Non-Gaussian seasonal adjustment, *Comput. Math. Appl.*, **18**(617), 503-514.
- Kitagawa, G. (1991). A nonlinear smoothing method for time series analysis, *Statistica Sinica*, **1**(2), 371-388.
- 北川源四郎 (1993). 『FORTRAN77時系列解析プログラミング』, 岩波書店, 東京.
- Kitagawa, G. (1993). A Monte Carlo filtering and smoothing method for non-Gaussian nonlinear state space models, *Proceedings of the 2nd U.S.-Japan Joint Seminar on Statistical Time Series Analysis*, 110-131, Honolulu, Hawaii, January 25-29.
- Kitagawa, G. (1994). The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother, *Ann. Inst. Statist. Math.*, **46**(4), 605-623.
- Kitagawa, G. (1995). Self-organizing smoother for nonlinear non-Gaussian state space model, *Research Memo.*, No. 578, The Institute of Statistical Mathematics, Tokyo.
- Kitagawa, G. (1996). Monte Carlo filtering and smoothing method for nonlinear non-Gaussian state space models, *Journal of Computational Graphical Statistics*, **5**(1), 1-25.
- Masreliz, C.J. (1975). Approximate non-Gaussian filtering with linear state and observation relations, *IEEE Trans. Automat. Control*, **AC-20**, 107-110.
- Meinhold, R.J. and Singpurwalla, N.D. (1989). Robustification of Kalman filter models, *J. Amer. Statist. Assoc.*, **84**(406), 479-486.
- Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1992). *Numerical Recipes in FORTRAN, The Art of Scientific Computing*, 2nd ed., Cambridge University Press, Cambridge.
- Sage, A.P. and Mersa, J.L. (1971). *Estimation Theory with Application to Communication and Control*, McGraw-Hill, New York.
- Schnatter, S. (1992). Integration-based Kalman-filtering for a dynamic generalized linear trend model, *Comput. Statist. Data Anal.*, **13**, 447-459.
- Smith, R.L. and Miller, J.E. (1986). A non-Gaussian state space model and application to prediction of

- records, *J. Roy. Statist. Soc. Ser. B*, **48**(1), 79-88.
- Tanizaki, H. (1993). Nonlinear Filters, *Lecture Notes in Econom. and Math. Systems*, **400**, Springer, Berlin.
- West, M., Harrison, P.J. and Migon, H.S. (1985). Dynamic generalized linear models and Bayesian forecasting (with discussion), *J. Amer. Statist. Assoc.*, **80**, 73-97.

## On Monte Carlo Filter and Smoother

Genshiro Kitagawa

(The Institute of Statistical Mathematics)

New algorithms for filtering and smoothing of general nonlinear non-Gaussian state space models are explained. The algorithms are based on the approximations of the predictive, filter, smoothed and noise distributions by many realizations generated from these distributions. This method can be applied to a broad class of nonlinear non-Gaussian high-dimensional state space models providing that the dimension of the system noise is not so high.

---

Key words: Fixed interval smoothing, non-Gaussian state space model, nonstationary time series, recursive filtering, time series modeling.