統計数理(2008)
第 56 巻 第 2 号 225–234

特集「データマイニングと統計数理」
[研究ノート]

Merging Particle Filter とその特性

中野 慎也[1,2] ・上野 玄太[1,2] ・中村 和幸[1,2] ・樋口 知之[1,2]

(受付 2008 年 1 月 4 日;改訂 2008 年 3 月 31 日)

要　旨

逐次データ同化への応用が検討されつつある手法の一つに particle filter があるが，この手法はアンサンブルの縮退という問題が起こるために高次元の問題に対してはあまり有効に機能しない．そこでこの問題を回避するために merging particle filter (MPF) という手法が提案された．MPF において，フィルタ分布を表現するアンサンブルの構成粒子は，予測分布を表現するアンサンブルから抽出した複数のサンプルの重みつき和によって生成する．このとき，フィルタ分布の平均と共分散が保存されるよう適切に重みを与える必要があるが，その与え方には任意性がある．本研究では，重みの与え方によって MPF の性質がどう変化するかを調べるために，2 種類の重みの与え方のもとでデータ同化実験を行った．その結果，低次元のモデルに対しては，重みのうちの一つを 1 に近い値に，その他を 0 に近い値にした方が基本的に精確な推定ができるが，比較的次元の高いモデルに対しては，そのような重みの設定では，アンサンブルを構成する粒子の数を多くしないとよい推定が得られないことがわかった．

キーワード： データ同化，particle filter，merging particle filter.

1. はじめに

データ同化(data assimilation) (e.g., Kalnay, 2003; 中村 他, 2005; Evensen, 2006) とは，地球物理学的現象のモデリングの際に用いられる考え方で，物理法則を記述する数値シミュレーションモデルに実際の観測データの情報を組み込むことで，実際の現象をよく再現する信頼性の高いモデルを構築しようとすることを言う．データ同化は，すでに気象予報の精度向上などの目的で応用されているほか，さらに様々な目的での応用が検討されており，地球物理学における重要な研究トピックとなっている．一口にデータ同化と言っても様々なアプローチがあるが，中でもシステムの時間発展をシミュレーションモデルで計算しながら順次観測の情報を取り込んでいくアプローチは逐次データ同化と呼ばれ，地球物理学的現象のダイナミクスをモデリングするための強力な方法として盛んに研究がなされている．

逐次データ同化を行うためのアルゴリズムには様々なものがあり，状況に応じて使い分けられる．まず，システムのダイナミクスが線型で記述でき，かつシステムの状態の確率分布が Gauss 分布で記述できる場合には，Kalman filter を用いて逐次データ同化を行うことができる．しかし，線型の方程式で記述できるような地球物理現象はさほど多くはなく，Kalman filter を適用できる状況は限られてくる．システムのダイナミクスが非線型であっても，システム方程式を線型近似して状態の共分散行列を計算する extended Kalman filter が使えるが，不安定性

[1] 統計数理研究所：〒106–8569 東京都港区南麻布 4–6–7
[2] 科学技術振興機構，戦略的創造研究推進事業

の強い非線型システムを扱う際には有効に機能しない(e.g., Evensen, 1992)うえ，行列の計算を多く含むために高次元のモデルでは著しく計算時間が掛かるという問題もある．計算時間の問題については，singular evolutive extended Kalman (SEEK) filter (Pham et al., 1998b)によって解決できるが，これも線型近似を用いており，不安定性の強いシステムには適用できない．

不安定性の強い非線型システムのモデリングには，確率分布を多数の実現値で構成されるアンサンブルで近似するのが有効である．Ensemble Kalman filter (EnKF) (Evensen, 1994; Burgers et al., 1998)は，確率分布をアンサンブルで近似して一期先予測に利用するアルゴリズムの一つであり，非線型モデルに対するデータ同化によく用いられている．ただ，EnKF もシステムの状態と観測との間に線型の関係が成り立つことを仮定しており，その関係が成り立たない場合には妥当な結果が得られない場合もある．また，Kalman filter や extended Kalman filter と同様，行列の計算を多く含むために高次元のモデルでは著しく計算時間が掛かるという問題もある．アンサンブル近似に基づく別の手法として，SEEK filter の変形である singular evolutive interpolated Kalman (SEIK) filter (Pham et al., 1998a)という方法も提案されている．これは EnKF よりも計算効率がよいと言われている(Nerger et al., 2005)が，システムの状態と観測との間に線型の関係が成り立たない場合は，EnKF 同様，あまり有効ではない．

そこで，システムの状態と観測との間の関係が線型であることを仮定しない方法として挙げられるのが，particle filter (PF) (Gordon et al., 1993; Kitagawa, 1993, 1996; Kitagawa and Gersch, 1996; Higuchi and Kitagawa, 2000; van Leeuwen, 2003)である．PF も確率分布のアンサンブル近似に基づくアルゴリズムの一つであるが，この方法は，システム，観測の線型性の仮定や Gauss 分布の仮定を必要としないため，広く様々なデータ同化問題に適用可能である．PF では，アンサンブルを構成するサンプル(粒子と呼ばれる)がどれだけデータに適合しているかを尤度によって評価し，尤度の比に応じて粒子の複製をつくることで，フィルタ分布を近似するアンサンブルを得る．このとき，尤度がきわめて小さい粒子は破棄されることになる．しかし，尤度が小さい粒子が破棄されて他の粒子の複製に置き換えられることで，フィルタリングを何度も繰り返すうちにアンサンブルを構成する粒子の多くが同一もしくは互いにきわめて近い値を取るようになり，状態の確率分布がうまく表現できなくなってしまう場合があるという問題がある．これは EnKF などのアルゴリズムにはみられない問題で，アンサンブルの縮退と呼ばれる．アンサンブルの縮退は，原理的にはアンサンブルに含まれる粒子の数を十分多くすることで回避可能なのだが，一期先予測に多大な計算コストがかかるような問題の場合には粒子数を増やすこと自体が困難であり，実際に PF の応用できる範囲が極めて限定されてしまう大きな原因となっている．観測の線型性を仮定する必要がなく，かつアンサンブルの縮退が起こらないアルゴリズムとして，事後分布を一旦 Gauss 分布で近似し，その Gauss 分布から粒子をサンプリングし直す方法(Kotecha and Djurić, 2003)，各粒子の示す実現値を中心とする Gauss 分布の重ね合わせで事後分布を表現し，サンプリングし直す kernel filter などと呼ばれる方法(Hürzeler and Künsch, 1998; Anderson and Anderson, 1999; Musso et al., 2001)など，PF から派生した手法も幾つか提案されているが，高次元のモデルでは適切な共分散を持つ Gauss 分布からのサンプリング自体に多大な計算コストを要する場合があり，これらの方法もあまり有効とは言えない．

上記のような背景から，PF の利点を活かしつつも，アンサンブルの縮退が起こりにくく，またモデルが高次元になっても計算コストがあまり増大しない方法として提案されたのが merging particle filter (MPF) (Nakano et al., 2007)である．MPF では，フィルタ分布を表現するアンサンブルの各構成粒子を，予測分布を表現するアンサンブルから複数のサンプルを抽出した上で，その重みつき和を取ることによって生成する．この重みつき和を取る際に，重みを適切に調整することで，フィルタ分布の平均と共分散の情報がアンサンブルで保持されるようにする．以

下では，まず第 2 節において MPF のアルゴリズムについて説明した上で，第 3 節で幾つかのデータ同化実験を行ってその結果を示し，第 4 節でその結果を踏まえてこのアルゴリズムの特性についての考察を加える．

2. Merging particle filter

まず，

(2.1a) $$\boldsymbol{x}_k = F_k(\boldsymbol{x}_{k-1}, \boldsymbol{v}_k)$$

(2.1b) $$\boldsymbol{y}_k = H_k(\boldsymbol{x}_k, \boldsymbol{w}_k)$$

のような状態空間モデルを考える．ここで，$\boldsymbol{x}_k$，$\boldsymbol{y}_k$ はそれぞれ時刻 $T=t_k$ $(k=1,\ldots)$ における状態ベクトル，観測ベクトルであり，$\boldsymbol{v}_k$，$\boldsymbol{w}_k$ はそれぞれシステムノイズ，観測ノイズを示す．演算子 F_k はシステムの時間発展を記述し，H_k はある状態ベクトル，観測ノイズに対して得られるべき観測を与える．

PF を適用する場合と同様，時刻 $T=t_{k-1}$ でのフィルタ分布 $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})$ を，アンサンブル $\{\boldsymbol{x}^{(i)}_{k-1|k-1}\}_{i=1}^N$ を用いて

(2.2) $$p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1}) \approx \frac{1}{N}\sum_{i=1}^{N}\delta\left(\boldsymbol{x}_{k-1}-\boldsymbol{x}^{(i)}_{k-1|k-1}\right)$$

のように近似する．ここで，δ は Dirac のデルタ関数であり，N はアンサンブルを構成するサンプルの数である．ここから，次のタイムステップ $T=t_k$ での予測分布のアンサンブル近似が

(2.3) $$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) \approx \frac{1}{N}\sum_{i=1}^{N}\delta\left(\boldsymbol{x}_k-\boldsymbol{x}^{(i)}_{k|k-1}\right)$$

というように得られる．ただし，予測分布のアンサンブルを構成する各粒子 $\boldsymbol{x}^{(i)}_{k|k-1}$ は，アンサンブル $\{\boldsymbol{x}^{(i)}_{k-1|k-1}\}_{i=1}^N$ を構成する各粒子にシステム方程式を $F_k(\boldsymbol{x}^{(i)}_{k-1|k-1}, \boldsymbol{v}^{(i)}_k)$ のように適用することで得られる．

MPF では，フィルタ分布のアンサンブル近似 $\{\boldsymbol{x}^{(i)}_{k|k}\}_{i=1}^N$ を構成する各粒子 $\boldsymbol{x}^{(i)}_{k|k}$ を生成するために，アンサンブル $\{\boldsymbol{x}^{(i)}_{k|k-1}\}_{i=1}^N$ から復元抽出によって得た複数個の粒子を用いる．したがって，もし粒子 $\boldsymbol{x}^{(i)}_{k|k}$ を生成するのに n 個の粒子を用いるとすると，N 個の粒子のアンサンブル $\{\boldsymbol{x}^{(i)}_{k|k}\}_{i=1}^N$ を生成するために，$n\times N$ 個のサンプル $\{\hat{\boldsymbol{x}}^{(1,1)}_{k|k},\ldots,\hat{\boldsymbol{x}}^{(n,1)}_{k|k},\ldots,\hat{\boldsymbol{x}}^{(1,N)}_{k|k},\ldots,\hat{\boldsymbol{x}}^{(n,N)}_{k|k}\}$ を予測分布アンサンブル $\{\boldsymbol{x}^{(i)}_{k|k-1}\}_{i=1}^N$ から復元抽出することになる．各粒子 $\boldsymbol{x}^{(i)}_{k|k}$ は，この $n\times N$ 個のサンプル $\{\hat{\boldsymbol{x}}^{(i,j)}_{k|k}\}_{i,j}$ のうちの n 個の粒子からなるサブセット $\{\hat{\boldsymbol{x}}^{(1,i)}_{k|k},\ldots,\hat{\boldsymbol{x}}^{(n,i)}_{k|k}\}$ の重みつき和

(2.4) $$\boldsymbol{x}^{(i)}_{k|k} = \sum_{j=1}^{n}\alpha_j\hat{\boldsymbol{x}}^{(j,i)}_{k|k}$$

によって生成する．このとき，重み α_j を

(2.5a) $$\sum_{j=1}^{n}\alpha_j = 1$$

(2.5b) $$\sum_{j=1}^{n}\alpha_j^2 = 1$$

を満たすように与えれば，新たに得られるアンサンブル $\{\boldsymbol{x}_{k|k}^{(i)}\}_{i=1}^N$ がフィルタ分布 $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ と近似的に等しい平均・共分散を持つことになる(Nakano et al., 2007 参照)．そこで MPF では，このアンサンブルを 2 次のモーメントまでを保持したフィルタ分布の近似と見なし，次の一期先予測のステップに進む．

なお，MPF によって得られるアンサンブル $\{\boldsymbol{x}_{k|k}^{(i)}\}_{i=1}^N$ には，フィルタ分布の 2 次のモーメントまでの情報が近似的に保存されるが，3 次以上のモーメントの情報は保存されない．したがって，MPF ではフィルタ分布の形状についての情報が一般には失われてしまうことになる．2 次までのモーメントを保存し，かつ 3 次以上のモーメントも保存できるような実数の重みの組み合わせ $\{\alpha_j\}_{j=1}^n$ が存在しないことは，以下のようにして確認できる．簡単のため確率変数 x_k を 1 次元として，m 次のモーメントを考えると，

$$
\begin{aligned}
(2.6)\quad & \int (x_k-\mu_{k|k})^m\, p(x_k|y_{1:k})\, dx_k \\
& \approx \int (x_k-\mu_{k|k})^m \frac{1}{N}\sum_{i=1}^N \delta\left(x_k - x_{k|k}^{(i)}\right) dx_k \\
& = \frac{1}{N}\sum_{i=1}^N (x_{k|k}^{(i)}-\mu_{k|k})^m = \frac{1}{N}\sum_{i=1}^N\prod_{l=1}^m\left(\sum_{j_l=1}^n \alpha_{j_l}\hat{x}_{k|k}^{(j_l,i)}-\mu_{k|k}\right) \\
& \approx \frac{1}{N}\sum_{i=1}^N\sum_{j=1}^n \alpha_j^m\left(\hat{x}_{k|k}^{(j,i)}-\mu_{k|k}\right)^m = \sum_{j=1}^n \alpha_j^m \int (x_k-\mu_{k|k})^m \frac{1}{N}\sum_{i=1}^N \delta\left(x_k-\hat{x}_{k|k}^{(j,i)}\right)\, dx_k \\
& \approx \sum_{j=1}^n \alpha_j^m \int (x_k-\mu_{k|k})^m\, p(x_k|y_{1:k})\, dx_k
\end{aligned}
$$

となるから，m 次のモーメントが保存されるためには，

$$(2.7)\qquad \sum_{j=1}^n \alpha_j^m = 1$$

が成り立つ必要がある．ただし，式(2.6)の式変形の過程で

$$\frac{1}{N}\sum_{i=1}^N\prod_{l=1}^m\left(\hat{x}_{k|k}^{(j_l,i)}-\mu_{k|k}\right)\approx 0 \quad (\text{unless } j_1=j_2=\cdots=j_m)$$

と近似した．一方，0 でない実数の組 $\{\alpha_j\}$ が式(2.5b)を満足するためには，$0<|\alpha_j|<1$ がすべての j に対して成り立たなくてはならないから，2 より大きい整数 m に対して，

$$(2.8)\qquad |\alpha_j|^2 > |\alpha_j|^m$$

が言える．α_j が実数なら $\alpha_j^2>0$ だから，$\alpha_j>0$，$\alpha_j<0$ に関わらず

$$(2.9)\qquad \alpha_j^2 > \alpha_j^m.$$

これがすべての j について言えるので，

$$(2.10)\qquad \sum_{j=1}^n \alpha_j^2 > \sum_{j=1}^n \alpha_j^m.$$

したがって，0 でない実数の組 $\{\alpha_j\}$ が式(2.5b)を満足するとき，$m>2$ であれば式(2.10)から

$$(2.11)\qquad \sum_{j=1}^n \alpha_j^m < 1$$

となり，式(2.7)の m 次のモーメントが保存されるための条件が成り立たなくなることがわかる．つまり，2 次のモーメントを保存するような重みで 3 次以上のモーメントが保存されることはない．ここでは確率変数 x_k を 1 次元としたが，$\boldsymbol{x}_k$ が多次元の場合でも m 次のモーメントが保存されないことが同様にして確認できる．

重みつき和を取る粒子の数 n は 3 以上の数にする必要がある．もし $n=1$ ならば通常の PF そのものとなり，また $n=2$ の場合も，式(2.5a), (2.5b)を満足するには片方の重みを 1，もう片方を 0 に取るしかなく，やはり通常の PF と同じものになってしまうからである．一方，$n \geq 3$ であれば，式(2.5a), (2.5b)を満たすような重み $\{\alpha_1, \ldots, \alpha_n\}$ の与え方は無数に存在する．基本的には，アンサンブルの多様性が維持されるように，すべての重みを互いに異なる値に設定するのが望ましいと考えられるが，重みをどのように与えるべきかについての具体的な指針は今のところ特に提案されていない．ただし，重みの与え方によって MPF の性質は少し変化する．以下ではそのことを示すために 2 種類の重みの与え方のもとでデータ同化実験を行い，その結果を比較していく．

3. データ同化実験

3.1 Lorenz 63 モデルによる実験

まず，Lorenz (1963)によって発表された 3 変数の簡単なカオスのモデル(以下 Lorenz 63 モデルと呼ぶ)を使ってデータ同化の実験を行う．Lorenz 63 モデルは，以下のような方程式によって記述される．

$$\frac{dx}{dt} = -s(x-y) \tag{3.1a}$$

$$\frac{dy}{dt} = rx - y - xz \tag{3.1b}$$

$$\frac{dz}{dt} = xy - bz. \tag{3.1c}$$

ここでは，パラメータを元の論文にしたがって $s=10$，$r=28$，$b=8/3$ と与える．時間の 1 ステップは 0.01 とする．同化するデータとしては，Lorenz 63 モデルにある初期値を与えて得られた結果にノイズを加えることで人工的に生成したデータを用いた．データは 20 ステップごとに 1 回，x，y，z の 3 成分すべてについて得られるものとし，各成分に加わるノイズは標準偏差 2 の Gaussian ノイズとした．また，MPF でも通常の PF でも，フィルタリングの際，観測 $\boldsymbol{y}_k$ が得られたときの予測ベクトル $\boldsymbol{x}_{k|k-1} = (x_{k|k-1}, y_{k|k-1}, z_{k|k-1})$ の尤度 $p(\boldsymbol{y}_k|\boldsymbol{x}_{k|k-1})$ を計算する必要があるが，ここでは，観測ノイズ $\boldsymbol{w}_k$ が平均 0，共分散行列が $\mathrm{diag}(\sigma^2, \sigma^2, \sigma^2)$ となるような Gauss 分布に従うものとして，

$$p(\boldsymbol{y}_k|\boldsymbol{x}_{k|k-1}) = \frac{1}{(\sqrt{2\pi}\sigma)^3} \exp\left[-\frac{||\boldsymbol{y}_k - \boldsymbol{x}_{k|k-1}||^2}{2\sigma^2}\right] \tag{3.2}$$

のように尤度を計算する．ただし，$\sigma=3$ とする．すなわち，尤度の計算の際の観測ノイズの標準偏差は，人工データに加えられた観測ノイズの標準偏差とは異なる値としている．システムノイズに関しては，平均 0，共分散行列が $\mathrm{diag}(0.01, 0.01, 0.01)$ の Gaussian ノイズを仮定する．時間 $T=t_1$ における $\boldsymbol{x}_{1|1}$ の分布は，そのときの観測データを平均とする共分散行列が $\mathrm{diag}(16, 16, 16)$ の Gauss 分布になるものとした．

MPF を適用する際の重みについては，和を取る粒子の数 n を 3 としたうえで重みをそれぞれ

$$\alpha_1 = \frac{3}{4}, \qquad \alpha_2 = \frac{\sqrt{13}+1}{8}, \qquad \alpha_3 = -\frac{\sqrt{13}-1}{8} \tag{3.3}$$

表 1. Root-mean-square deviations from the true state over 50,000 time steps for an experiment using the Lorenz 63 model.

	MPF 1	MPF 2	PF
$N = 64$	0.97	0.93	4.79
$N = 128$	0.93	0.88	4.07
$N = 256$	0.91	0.88	0.87
$N = 512$	0.91	0.88	0.87
$N = 1024$	0.91	0.87	0.85
$N = 2048$	0.90	0.87	0.85
$N = 4096$	0.90	0.87	0.84

と与えた場合，及び，それぞれ

$$\alpha_1 = \frac{19}{20}, \qquad \alpha_2 = \frac{\sqrt{77}+1}{40}, \qquad \alpha_3 = -\frac{\sqrt{77}-1}{40} \tag{3.4}$$

と与えた場合の 2 通りを考える．式(3.4)の重みは，式(3.3)の重みに比べて α_1 が 1 に近くなっており，通常の PF に近い重みづけになっている．これら 2 種類の MPF と通常の PF を加えた計 3 つの方法について，粒子数 N を様々な値に変化させながらデータ同化実験を行い，それぞれどのくらいの粒子数で縮退の影響を受けるのかを調べた．

表 1 は，データ同化の結果得られた状態の推定値を人工データを生成したシミュレーションの上での実際の値と比較したものであり，推定値と実際の状態の値との差を，50,000 ステップ分について root-mean-square をとったもので示している．MPF 1 は式(3.3)で重みを定めたもの，MPF 2 は式(3.4)で重みを定めたものである．N が大きい場合には PF の方が精度がよいが，N が小さい場合には概して PF よりも MPF の方が精度のよい推定ができるということが分かる．また，MPF のうちでも，より PF に近い式(3.4)の重みづけの方が，式(3.3)の重みづけよりもよい精度で推定が得られることが分かる．

3.2 Lorenz 96 モデルによる数値実験

次に，Lorenz and Emanuel (1998)のより次元の大きいモデル(以下 Lorenz 96 モデルと呼ぶ)を使った実験を行う．Lorenz 96 モデルの各成分 $x_j (j=1,\ldots,J)$ の時間発展は，以下のような方程式によって記述される

$$\frac{dx_j}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + f. \tag{3.5}$$

但し，$x_{-1} = x_{J-1}$，$x_0 = x_J$，$x_{J+1} = x_1$ とし，$f = 8$ とする．また，モデルの次元 J は 40，時間の 1 ステップは 0.005 とした．同化するデータは，この Lorenz 96 モデル自体を走らせて得られた結果に観測ノイズを加えることで人工的に生成した．データの取得は，まずモデルを初期値

$$x_j = 8.0 \qquad (\text{for } j \neq 20) \tag{3.6a}$$

$$x_j = 8.008 \qquad (\text{for } j = 20) \tag{3.6b}$$

から擾乱を十分発達させるために 2,000 ステップ走らせた上で，そこから 10 タイムステップずつ行った．但し，このモデルの変数 x_j のうちの j が偶数のもののみが観測されるものとし，かつ観測で得られるのは x_j の絶対値のみであるものとした．データに加えた観測ノイズには，平均 0，標準偏差 1.5 の Gauss 分布にしたがうノイズを用いた．

表 2. Root-mean-square deviations from the true state from time step 3,000 to time step 20,000 for an experiment using the Lorenz 96 model.

	MPF 1	MPF 2	PF
$N = 256$	2.47	3.72	4.01
$N = 512$	1.50	2.76	3.66
$N = 1024$	1.20	1.55	3.70
$N = 2048$	1.19	1.28	3.15
$N = 4096$	1.14	1.12	2.65
$N = 8192$	1.14	1.08	2.07
$N = 16384$	1.13	1.05	1.80
$N = 32768$	1.13	1.03	1.23

このようにして生成したデータを同化する際，システムノイズは平均 0，共分散 diag(0.25,...,0.25) の Gauss 分布にしたがうものと仮定し，また尤度は

$$p(\boldsymbol{y}_k|\boldsymbol{x}_{k|k-1}) = \frac{1}{(\sqrt{2\pi}\sigma)^{20}} \exp\left[-\frac{||\boldsymbol{y}_k - H(\boldsymbol{x}_{k|k-1})||^2}{2\sigma^2}\right] \tag{3.7}$$

とする．ただし，観測演算子 H は，$H(\boldsymbol{x}_{k|k-1}) = (|x_{2,k|k-1}|\ |x_{4,k|k-1}|\ \ldots\ |x_{40,k|k-1}|)^T$ となるように定義されたものであり，また $\sigma = 3$ とする．先の Lorenz 63 モデルでの実験と同様，MPF を適用するにあたっては重みが式(3.3)で与えられる場合と式(3.4)で与えられる場合との 2 通りを考え，これに通常の PF を加えた 3 種類の方法について実験を行い，比較した．

表 2 は，データ同化の結果得られた状態の推定値を人工データを生成したシミュレーションの上での実際の値と比較したものであり，推定値と実際の状態の値との差を，最初の観測から 5,000 ステップ目から 20,000 ステップ目までについて root-mean-square をとったもので示している．表 1 と同様，MPF 1 は式(3.3)で重みを定めたもの，MPF 2 は式(3.4)で重みを定めたものである．PF を用いた場合，N を 32768 まで増やしても，MPF ほど精度のよい推定はできない．一方，MPF については，2 種類を比較すると，N が大きい場合には，より PF に近い式(3.4)の重みづけの方が，式(3.3)の重みづけよりもよい精度で推定が得られるが，$N = 1024$ もしくは $N = 2048$ では，式(3.3)の重みを使った場合の方が式(3.4)の重みの場合よりもよい推定結果が得られている．

4. 考察

Nakano et al. (2007) で議論されたように，MPF は PF と比べて少ない粒子数で精度よい状態の推定ができるが，前節で示されたように MPF の特性はその重みの与え方によっても変化する．MPF で重みつき和を取る際，仮に重みのうちの 1 つ α_1 を 1 とし，残りの重みを 0 とすると通常の PF と同値になるが，そこから重みの値を少し動かして，α_1 を 1 に近い値にとり，残りは 0 に近い値となるようにすると，通常の PF で得られるアンサンブル $\{\hat{\boldsymbol{x}}_{k|k}^{(1,i)}\}_{i=1}^N$ の各粒子の場所にアンサンブル $(\sum_{j=2}^n \alpha_j \hat{\boldsymbol{x}}_{k|k}^{(j,i)})_{i=1}^N$ で表現される幅の小さい分布を配置した形になる．つまりこれは kernel filter (Hürzeler and Künsch, 1998; Anderson and Anderson, 1999; Musso et al., 2001) と類似のものを MPF の文脈で実現したものと見なすことができる．MPF では，第 2 節でも述べたように一般にはフィルタ分布の形状が保存されないが，このように α_1 を 1 に近い値にとれば，kernel filter 同様，フィルタ分布の形状が大きく歪められることはないものと考えられる．

Lorenz 63 モデルの実験では，MPF の重みのうち，α_1 を 1 に近い値にとった方がよい推定

結果が得られるという傾向が現れた．これは，Lorenz 63 モデルでは，強い非線型性のために状態の確率分布が Gauss 分布とは大きく異なる形になってしまう場合が多く，フィルタ分布の形状を歪めずに扱った方が精確な状態の推定ができることを示しているものと考えられる．実際に地球物理モデルにデータ同化を適用する際にも，しばしば状態の確率分布が Gauss 分布から大きく外れてくる場合があるが，そのような場合には，こうした α_1 を 1 に近い値にとるような重みづけが有効となるだろう．

一方，Lorenz 96 モデルの実験では，粒子数 N を少なくした場合，α_1 を 1 に近づけない方がよい推定結果を得た．このことは，高次元のモデルにおいては，粒子数 N を少なくすると，フィルタ分布の形状を保存するよりもアンサンブルの縮退の方が問題になってくるためと考えられる．但し，粒子数 N を多く取れば，α_1 を 1 に近づけた場合の方がより精確な推定値を得ることができた．これは，十分な数の粒子を使うことができるならば，フィルタ分布の形状をうまく表現できる手法を用いた方がより精確な推定ができるということを示している．勿論，α_1 をあまり 1 に近づけ過ぎると，通常の PF とほとんど同じになってしまって縮退の影響が出やすくなると考えられるので，粒子数 N を多く取れる場合でも，α_1 に縮退の影響が出にくい程度の適切な値を選ぶことが大切であろう．

謝　辞

本研究は科学技術振興機構・戦略的基礎研究推進事業（JST/CREST）ならびに科学研究費補助金（19740306）の助成を受けたものである．

参 考 文 献

Anderson, J. L. and Anderson, S. L. (1999). A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts, *Monthly Weather Review*, **127**, 2741–2758.

Burgers, G., van Leeuwen, P. J. and Evensen, G. (1998). Analysis scheme in the ensemble Kalman filter, *Monthly Weather Review*, **126**, 1719–1724.

Evensen, G. (1992). Using the extended Kalman filter with a multilayer quasi-geostrophic model, *Journal of Geophysical Research*, **97**(C11), 17905–17924.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, *Journal of Geophysical Research*, **99**(C5), 10143–10162.

Evensen, G. (2006). *Data Assimilation: The Ensemble Kalman Filter*, Springer-Verlag, Berlin and Heidelberg.

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**, 107–113.

Higuchi, T. and Kitagawa, G. (2000). Knowledge discovery and self-organizing state space model, *IEICE Transactions on Information and Systems*, **E83-D**, 36–43.

Hürzeler, M. and Künsch, H. R. (1998). Monte Carlo approximations for general state space models, *Journal of Computational Graphical Statistics*, **7**, 175–193.

Kalnay, E. (2003). *Atmospheric Modeling, Data Assimilation and Predictability*, Cambridge University Press, Cambridge.

Kitagawa, G. (1993). A Monte Carlo filtering and smoothing method for non-Gaussian nonlinear state space models, Research Memorandum, No. 462, The Institute of Statistical Mathematics, Tokyo.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, *Journal of Computational Graphical Statistics*, **5**, 1–25.

Kitagawa, G. and Gersch, W. (1996). *Smoothness Priors Analysis of Time Series*, Chapter 6, Springer-Verlag, New York.

Kotecha, J. H. and Djurić, P. M. (2003). Gaussian particle filtering, *IEEE Transactions on Signal Processing*, **51**, 2592–2601.

Lorenz, E. N. (1963). Deterministic nonperiodic flow, *Journal of the Atmospheric Sciences*, **20**, 130–141.

Lorenz, E. N. and Emanuel, K. A. (1998). Optimal sites for supplementary weather observations: Simulations with a small model, *Journal of the Atmospheric Sciences*, **55**, 399–414.

Musso, C., Oudjane, N. and Le Gland, F. (2001). Improving regularized particle filters, *Sequential Monte Carlo Methods in Practice* (eds. A. Doucet, N. de Freitas and N. Gordon), Chapter 12, 247–271, Springer-Verlag, New York.

中村和幸, 上野玄太, 樋口知之(2005). データ同化：その概念とアルゴリズム, 統計数理, **53**, 211–229.

Nakano, S., Ueno, G. and Higuchi, T. (2007). Merging particle filter for sequential data assimilation, *Nonlinear Processes in Geophysics*, **14**, 395–408.

Nerger, L., Hiller, W. and Schröter, J. (2005). A comparison of error subspace Kalman filters, *Tellus*, **57A**, 715–735.

Pham, D. T., Verron, J. and Gourdeau, L. (1998a). Filtres de Kalman singuliers évolutifs pour l'assimilation de données en océanographie, *Comptes rendus de l'Académie des sciences, Série 2: Sciences de la Terre et des Planètes*, **326**, 255–260.

Pham, D. T., Verron, J. and Roubaud, M. C. (1998b). A singular evolutive extended Kalman filter for data assimilation in oceanography, *Journal of Marine Systems*, **16**, 323–340.

van Leeuwen, P. J. (2003). A variance-minimizing filter for large-scale applications, *Monthly Weather Review*, **131**, 2071–2084.

Merging Particle Filter and Its Characteristics

Shin'ya Nakano, Genta Ueno, Kazuyuki Nakamura and Tomoyuki Higuchi

The Institute of Statistical Mathematics;
Japan Science and Technology Agency

A significant problem with the basic particle filter algorithm is degeneration. The merging particle filter algorithm has recently been proposed to overcome this problem at a reasonable computational cost. In an MPF, each member of a filtered ensemble is generated from a weighted sum of multiple samples from the forecast ensemble such that the mean and covariance of the filtered distribution are approximately preserved. In this study, we performed data assimilation experiments using an MPF with two different sets of merging weights. When one merging weight is set to near 1 and the other weights are set small, better estimates are obtained for data assimilation into a low dimensional model. For data assimilation into a relatively large dimensional model, such a weight set requires a large ensemble size.

Key words: Data assimilation, particle filter, merging particle filter.