

アンサンブルを用いた変分法データ同化と 計数データのための拡張

中野 慎也^{1,2,3}

(受付 2022 年 1 月 3 日 ; 改訂 6 月 10 日 ; 採択 6 月 13 日)

要 旨

4次元変分法の問題を解く方法の一つとして、シミュレーションを様々な設定で実行するアンサンブルシミュレーションの結果を利用した手法が提案されている。こうしたアンサンブルによる変分法アルゴリズムは、シミュレーションモデルを完全にブラックボックスとして扱うことができるため、広く使われるアジョイント法と比較して、非常に実装しやすいのが利点となっている。但し、現状のアンサンブルによる手法は、システムの状態が与えられた下で観測のしたがる条件付き分布が正規分布になることを仮定してアルゴリズムが導出されており、そのままでは正規分布以外の分布にしたがる観測を利用できない。本研究では、システムの状態が与えられた下で観測データがポアソン分布にしたがる場合を考え、シミュレーションモデルをブラックボックス化可能なアンサンブル変分法のアルゴリズムを導出する。

キーワード：データ同化、4次元アンサンブル変分法、ガウス・ニュートン法、ポアソン分布。

1. はじめに

データ同化の代表的なアプローチの一つである4次元変分法は、観測データの時系列が与えられた下で事後確率密度を最大化する解(maximum a posteriori 推定値; MAP 推定値)を求める方法である。4次元変分法の問題を解く代表的なアルゴリズムはアジョイント法(adjoint method; 例えば、淡路 他, 2009)であり、通常、4次元変分法と言えばアジョイント法のことを指す。但し、本稿において「4次元変分法」はMAP推定値を求める方法全般を指す言葉として用い、アジョイント法は4次元変分法の問題を解くアルゴリズムの一つとして区別する。アジョイント法は、シミュレーションモデルによる時間積分を行った後、シミュレーションモデルの随伴作用素に相当するアジョイントモデルで時間を遡る計算を行い、対数事後確率密度関数の勾配を効率的に求める手法である。比較的小さい計算コストで大規模なデータ同化を行うことができるのが利点であるが、一方でアジョイントモデルの開発に手間が掛かるという欠点がある。また、元のシミュレーションモデルに修正を加えた場合は、アジョイントモデルも一緒に更新する必要があるため、維持にも手間が掛かる。

一方、アンサンブルシミュレーションの結果を利用することで、アジョイントモデルを使

¹ 統計数理研究所：〒190-8562 東京都立川市緑町 10-3

² データサイエンス共同利用基盤施設 データ同化研究支援センター：〒190-8562 東京都立川市緑町 10-3

³ 総合研究大学院大学 複合科学研究科統計科学専攻：〒190-8562 東京都立川市緑町 10-3

わずに4次元変分法の問題を解く方法がLiu et al. (2008)によって提案されている。アンサンブルシミュレーションとは、シミュレーションを別々の設定で実行することを意味する。Liu et al. (2008)の4次元アンサンブル変分法は、Lorenz (2003)やZupanski (2005)が提案したアンサンブルシミュレーションによる対数事後確率密度関数の近似を用いており、近似された対数事後確率密度を最大化する値を勾配法で求める。Bocquet and Sakov (2013, 2014)は、同様の近似を用いたガウス・ニュートン法(Gauss-Newton method; 以下GN法)による4次元変分法を提案している。この方法は、手続きの上ではアンサンブルカルマン smoother (ensemble Kalman smoother) (van Leeuwen and Evensen, 1996; Evensen and van Leeuwen, 2000)を繰り返し適用する形になることから、iterative ensemble Kalman smoother (IEnKS)と呼ばれているが、平滑化分布を求める方法ではなく、MAP推定値を近似的に求める方法であり、アンサンブル変分法の一つと見なしてよいと思われる。アンサンブルを用いてGN法の近似アルゴリズムは他にも研究がなされている (Gu and Oliver, 2007; Evensen, 2018)。また、Nakano (2021)でも高次元の問題を念頭に類似したアルゴリズムを議論している。

これらのアンサンブル変分法は、アンサンブルシミュレーションを用いるために計算コストが掛かる上、アンサンブル近似によって精度も落ちる可能性があるが、シミュレーションモデルを完全にブラックボックスとして扱えるため、簡単に実装できるという利点がある。実際、筆者が最近参加した研究 (Minami et al., 2020)においても、アンサンブル変分法を採用することにより、シミュレーションモデルの中身に触れることなくデータ同化を実現している。一方、現状のアンサンブル変分法は、システムの状態が与えられた下で観測のしつがう条件付き分布(尤度)が正規分布になることを仮定してアルゴリズムが導出されており、尤度が正規分布以外の分布で与えられた場合に適用できるか否か自明ではない。現実には、正規分布以外の分布を想定すべき場合もあり、特にポアソン分布を仮定すべき状況は少なからず存在する。例えば、宇宙科学分野では、微弱な紫外線を遠隔から観測したデータを用いることがある。Nakano et al. (2014)では、正規分布による近似を用いて紫外線観測からの推定を行ったが、1回の観測でセンサーによって検知される光子の個数は数個にとどまる場合があり、本来このような場合はポアソン分布で計数データとして扱うのが適切である。他にも、感染症伝播シミュレーションにおいて与えられた状態の下での観測事例数をポアソン分布で表現する場合もあるなど (Saito et al., 2018)、様々な分野でポアソン分布を用いた観測のモデリングが行われる。そこで本研究では、尤度がポアソン分布で記述される場合に、アンサンブル変分法と同様のアルゴリズムが得られるかを検討する。以下では、2節で4次元変分法の問題を定式化した後、3節ではNakano (2021)に基づいたアンサンブルによる変分法のアルゴリズムを導入する。4節ではアルゴリズムのGN法との関係について述べる。5節では4節の考え方を応用し、ポアソン分布にしたがう観測を利用可能なアンサンブルによる変分法アルゴリズムを導出する。最後に6節では、5節のアルゴリズムの有効性を簡単な数値実験で検証する。

2. 4次元変分法

時刻 t_k におけるシステムの状態を $\mathbf{x}_k \in \mathbb{R}^d$ 、 \mathbf{x}_k から得られる観測を $\mathbf{y}_k \in \mathbb{R}^n$ と表す。ここでは、 $k = 1, \dots, K$ に対して、数値モデルによる時刻 t_{k-1} から時刻 t_k の間の状態遷移が関数 f_k で

$$(2.1) \quad \mathbf{x}_k = f_k(\mathbf{x}_{k-1})$$

のように書けるものとする。このような仮定を強拘束と呼ぶ。強拘束のもとでは、初期値 \mathbf{x}_0 が与えられれば、任意の k に対する \mathbf{x}_k が、

$$(2.2) \quad \mathbf{x}_k = \mathbf{f}_k \circ \mathbf{f}_{k-1} \circ \cdots \circ \mathbf{f}_1(\mathbf{x}_0)$$

のように定まる。但し、記号 \circ は合成関数を表す。

シミュレーションモデルから時刻 t_k における状態 \mathbf{x}_k が得られたとき、観測 \mathbf{y}_k の予測値を \mathbf{x}_k の関数 $\mathbf{h}_k(\mathbf{x}_k)$ で表すことにすると、 \mathbf{y}_k に基づく尤度 $p(\mathbf{y}_k|\mathbf{x}_0)$ は、

$$(2.3) \quad p(\mathbf{y}_k|\mathbf{x}_0) = p(\mathbf{y}_k|\mathbf{x}_k) \propto \exp\left(-\frac{1}{2}[\mathbf{y}_k - \mathbf{h}_k(\mathbf{x}_k)]^\top \mathbf{R}^{-1}[\mathbf{y}_k - \mathbf{h}_k(\mathbf{x}_k)]\right)$$

を満たす。 \mathbf{x}_0 の事前分布を

$$(2.4) \quad p(\mathbf{x}_0) = \frac{1}{\sqrt{(2\pi)^d |\mathbf{P}_b|}} \exp\left(-\frac{1}{2}[\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b}]^\top \mathbf{P}_b^{-1}[\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b}]\right)$$

とすると、観測の時系列 $\mathbf{y}_1, \dots, \mathbf{y}_K$ が与えられた下での \mathbf{x}_0 の事後分布は

$$(2.5) \quad p(\mathbf{x}_0|\mathbf{y}_1, \dots, \mathbf{y}_K) \propto \exp\left(-\frac{1}{2}[\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b}]^\top \mathbf{P}_b^{-1}[\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b}]\right) \\ \times \prod_{k=1}^K \exp\left(-\frac{1}{2}[\mathbf{y}_k - \mathbf{h}_k(\mathbf{x}_k)]^\top \mathbf{R}^{-1}[\mathbf{y}_k - \mathbf{h}_k(\mathbf{x}_k)]\right)$$

を満たす。強拘束の4次元変分法によるデータ同化では、 $p(\mathbf{x}_0|\mathbf{y}_1, \dots, \mathbf{y}_K)$ の対数を取った対数事後確率密度の最大化を考える。すなわち、以下の目的関数の最小化を行うことになる：

$$(2.6) \quad J(\mathbf{x}_0) = \frac{1}{2}(\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b})^\top \mathbf{P}_b^{-1}(\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b}) + \frac{1}{2} \sum_{k=1}^K [\mathbf{y}_k - \mathbf{h}_k(\mathbf{x}_k)]^\top \mathbf{R}^{-1}[\mathbf{y}_k - \mathbf{h}_k(\mathbf{x}_k)].$$

ここで、今後の議論で便利のように、

$$(2.7) \quad \mathbf{g}_k(\mathbf{x}_0) = \mathbf{h}_k \circ \mathbf{f}_k \circ \mathbf{f}_{k-1} \circ \cdots \circ \mathbf{f}_1(\mathbf{x}_0)$$

のように定義される関数 \mathbf{g}_k を導入しておく。このとき、 \mathbf{x}_k が式(2.2)を満たすことから、

$$(2.8) \quad \mathbf{h}_k(\mathbf{x}_k) = \mathbf{g}_k(\mathbf{x}_0)$$

が成り立つ。したがって、式(2.6)の J は

$$(2.9) \quad J(\mathbf{x}_0) = \frac{1}{2}(\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b})^\top \mathbf{P}_b^{-1}(\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b}) + \frac{1}{2} \sum_{k=1}^K [\mathbf{y}_k - \mathbf{g}_k(\mathbf{x}_0)]^\top \mathbf{R}^{-1}[\mathbf{y}_k - \mathbf{g}_k(\mathbf{x}_0)]$$

のように書き直せる。

3. アンサンブルを用いた変分法アルゴリズム

以下では、反復法で J の最小化を行うことを考え、 m 回目のイテレーションでの \mathbf{x}_0 の推定値を $\hat{\mathbf{x}}_{0,m}$ と置くことにする。 N 個の異なる初期値 $\{\mathbf{x}_{0,m}^{(1)}, \dots, \mathbf{x}_{0,m}^{(N)}\}$ を用意し、それぞれシミュレーションを実行する。このとき、 N 個の初期値は平均が $\hat{\mathbf{x}}_{0,m}$ となるように与える。すなわち

$$(3.1) \quad \frac{1}{N} \sum_{i=1}^N \mathbf{x}_{0,m}^{(i)} = \hat{\mathbf{x}}_{0,m}.$$

これにより N 種類のシミュレーション結果 $\{\mathbf{x}_{0:K,m}^{(1)}, \dots, \mathbf{x}_{0:K,m}^{(N)}\}$ が得られる。但し、 $\mathbf{x}_{0:K} = (\mathbf{x}_0^\top \cdots \mathbf{x}_K^\top)^\top$ であり、すなわち $\mathbf{x}_{0:K}$ は時刻 t_0 から t_K までの時間ステップごと

の状態を表している．ここで， \mathbf{x}_0 をアンサンブルシミュレーションの初期値 $\{\mathbf{x}_{0,m}^{(1)}, \dots, \mathbf{x}_{0,m}^{(N)}\}$ の線型和で表すものとする．通常， \mathbf{x}_0 の次元は N よりもはるかに大きく，このような制約を課すことによって， $(m+1)$ 回目のイテレーションで得られる \mathbf{x}_0 の推定値 $\hat{\mathbf{x}}_{0,m+1}$ はアンサンブルメンバーの張る $(N-1)$ 次元の部分空間に制限されることになる．しかし，イテレーションごとに異なる部分空間を張るアンサンブルメンバーを用いれば， $(N-1)$ 次元よりも高次元の空間で解を求めることができる．さて， \mathbf{x}_0 がアンサンブルメンバーの線型結合で表せるという制約の下で J の最小化を考える．このとき， $\hat{\mathbf{x}}_{0,m}$ が式(3.1)を満たすことに注意すれば，適当に $w_{m+1}^{(1)}, \dots, w_{m+1}^{(N)}$ を置くことにより，

$$(3.2) \quad \mathbf{x}_0 = \hat{\mathbf{x}}_{0,m} + \sum_{i=1}^N w_{m+1}^{(i)} (\mathbf{x}_{0,m}^{(i)} - \hat{\mathbf{x}}_{0,m})$$

と書ける．各アンサンブルメンバーの $\hat{\mathbf{x}}_{0,m}$ からの差を横に並べた $d \times N$ 行列

$$(3.3) \quad \mathbf{X}_m = (\mathbf{x}_{0,m}^{(1)} - \hat{\mathbf{x}}_{0,m} \quad \cdots \quad \mathbf{x}_{0,m}^{(N)} - \hat{\mathbf{x}}_{0,m})$$

を定義し， $\mathbf{w}_{m+1} = (w_{m+1}^{(1)} \quad \cdots \quad w_{m+1}^{(N)})^\top$ と置くと， \mathbf{x}_0 は以下のように書ける：

$$(3.4) \quad \mathbf{x}_0 = \hat{\mathbf{x}}_{0,m} + \mathbf{X}_m \mathbf{w}_{m+1}.$$

次に，アンサンブルシミュレーションの結果を利用しやすくするために，関数 $\mathbf{g}_k(\mathbf{x}_0)$ を

$$(3.5) \quad \mathbf{g}_k(\mathbf{x}_0) \approx \mathbf{g}_k(\hat{\mathbf{x}}_{0,m}) + \mathbf{G}_{k,m}(\mathbf{x}_0 - \hat{\mathbf{x}}_{0,m})$$

のように一次近似する．但し， $\mathbf{G}_{k,m}$ は関数 \mathbf{g}_k の $\mathbf{x}_0 = \hat{\mathbf{x}}_{0,m}$ におけるヤコビ行列である．これに式(3.4)を代入すると，

$$(3.6) \quad \mathbf{g}_k(\mathbf{x}_0) \approx \mathbf{g}_k(\hat{\mathbf{x}}_{0,m}) + \mathbf{G}_{k,m} \mathbf{X}_m \mathbf{w}_{m+1}.$$

関数 \mathbf{g}_k は，シミュレーションを k ステップ実行した後，関数 \mathbf{h}_k を適用する操作に相当するため，そのヤコビ行列 $\mathbf{G}_{k,m}$ を求めるのは通常極めて難しい．しかし，式(3.5)の近似を用いると，初期値 $\mathbf{x}_{0,m}^{(i)}$ に対する \mathbf{g}_k の出力が

$$(3.7) \quad \mathbf{g}_k(\mathbf{x}_{0,m}^{(i)}) \approx \mathbf{g}_k(\hat{\mathbf{x}}_{0,m}) + \mathbf{G}_{k,m}(\mathbf{x}_{0,m}^{(i)} - \hat{\mathbf{x}}_{0,m})$$

と書けるので，

$$(3.8) \quad \Gamma_{k,m} = (\mathbf{g}_k(\mathbf{x}_{0,m}^{(1)}) - \mathbf{g}_k(\hat{\mathbf{x}}_{0,m}) \quad \cdots \quad \mathbf{g}_k(\mathbf{x}_{0,m}^{(N)}) - \mathbf{g}_k(\hat{\mathbf{x}}_{0,m}))$$

のように定義される行列 $\Gamma_{k,m}$ を導入すると，

$$(3.9) \quad \mathbf{G}_{k,m} \mathbf{X}_m \approx \Gamma_{k,m}$$

という近似が成り立つ．これを式(3.6)に代入すれば，

$$(3.10) \quad \mathbf{g}_k(\mathbf{x}_0) \approx \mathbf{g}_k(\hat{\mathbf{x}}_{0,m}) + \Gamma_{k,m} \mathbf{w}_{m+1}$$

となる．行列 $\Gamma_{k,m}$ はアンサンブルシミュレーションの結果 $\{\mathbf{g}_{0:k}(\mathbf{x}_0^{(i)})\}_{i=1}^N$ から構成できるので，式(3.10)で \mathbf{g}_k を近似すれば，ヤコビ行列 $\mathbf{G}_{k,m}$ を計算する必要がなく都合がよい．なお，式(3.9)の近似の妥当性を担保するためには，各アンサンブルメンバー $\mathbf{x}_{0,m}^{(i)}$ と $\hat{\mathbf{x}}_{0,m}$ の距離 $\|\mathbf{x}_{0,m}^{(i)} - \hat{\mathbf{x}}_{0,m}\|$ が十分に小さくなるように，すなわち，アンサンブルの $\hat{\mathbf{x}}_{0,m}$ からのばらつきが十分に小さくなるようにするのが望ましい (Nakano, 2021)．式(3.10)と式(3.4)を式(2.9)に代入すると，目的関数 $J(\mathbf{x}_0)$ を近似する関数 $\tilde{J}_{m+1}(\mathbf{w}_{m+1})$ が得られる：

$$(3.11) \quad \check{J}_{m+1} = \frac{1}{2}(\hat{\boldsymbol{x}}_{0,m} - \bar{\boldsymbol{x}}_{0,b} + \mathbf{X}_m \boldsymbol{w}_{m+1})^\top \mathbf{P}_b^{-1}(\hat{\boldsymbol{x}}_{0,m} - \bar{\boldsymbol{x}}_{0,b} + \mathbf{X}_m \boldsymbol{w}_{m+1}) \\ + \frac{1}{2} \sum_{k=1}^K [\boldsymbol{y}_k - \boldsymbol{g}_k(\hat{\boldsymbol{x}}_{0,m}) - \boldsymbol{\Gamma}_{k,m} \boldsymbol{w}_{m+1}]^\top \mathbf{R}^{-1} [\boldsymbol{y}_k - \boldsymbol{g}_k(\hat{\boldsymbol{x}}_{0,m}) - \boldsymbol{\Gamma}_{k,m} \boldsymbol{w}_{m+1}].$$

\check{J}_{m+1} を最小化するために \boldsymbol{w}_{m+1} による勾配を取ると,

$$(3.12) \quad \nabla \check{J}_{m+1} = \mathbf{X}_m^\top \mathbf{P}_b^{-1}(\hat{\boldsymbol{x}}_{0,m} - \bar{\boldsymbol{x}}_{0,b} + \mathbf{X}_m \boldsymbol{w}_{m+1}) - \sum_{k=1}^K \boldsymbol{\Gamma}_{k,m}^\top \mathbf{R}^{-1} [\boldsymbol{y}_k - \boldsymbol{g}_k(\hat{\boldsymbol{x}}_{0,m}) - \boldsymbol{\Gamma}_{k,m} \boldsymbol{w}_{m+1}].$$

\check{J}_{m+1} を最小化する $\check{\boldsymbol{w}}_{m+1}$ は $\nabla \check{J}_{m+1} = 0$ の条件から以下のように求めることができる。

$$(3.13) \quad \check{\boldsymbol{w}}_{m+1} = - \left(\mathbf{X}_m^\top \mathbf{P}_b^{-1} \mathbf{X}_m + \sum_{k=1}^K \boldsymbol{\Gamma}_{k,m}^\top \mathbf{R}^{-1} \boldsymbol{\Gamma}_{k,m} \right)^{-1} \\ \times \left(\mathbf{X}_m^\top \mathbf{P}_b^{-1} [\hat{\boldsymbol{x}}_{0,m} - \bar{\boldsymbol{x}}_{0,b}] - \sum_{k=1}^K \boldsymbol{\Gamma}_{k,m}^\top \mathbf{R}^{-1} [\boldsymbol{y}_k - \boldsymbol{g}_k(\hat{\boldsymbol{x}}_{0,m})] \right).$$

これを式(3.4)に代入すると, $\check{\boldsymbol{x}}_0$ の推定値

$$(3.14) \quad \check{\boldsymbol{x}}_{0,m+1} = \hat{\boldsymbol{x}}_{0,m} + \mathbf{X}_m \check{\boldsymbol{w}}_{m+1}$$

が得られる。ただ, $\check{\boldsymbol{x}}_{0,m+1}$ を用いた場合, 元の目的関数 J に対して

$$(3.15) \quad J(\check{\boldsymbol{x}}_{0,m+1}) \leq J(\hat{\boldsymbol{x}}_{0,m})$$

が必ずしも成り立たないという問題がある。つまり, 推定値 $\check{\boldsymbol{x}}_{0,m+1}$ は $\hat{\boldsymbol{x}}_{0,m}$ よりもよいという保証がないことになる。そこで, ある実数 α に対し, $\|\boldsymbol{w}_{m+1}\|^2 < \alpha$ の領域で

$$(3.16) \quad \hat{J}_{m+1}(\boldsymbol{w}_{m+1}) = \check{J}_{m+1}(\boldsymbol{w}_{m+1}) + \frac{\sigma_{m+1}^2}{2} \boldsymbol{w}_{m+1}^\top \boldsymbol{w}_{m+1} \geq J(\boldsymbol{x}_0)$$

が成り立つように σ_{m+1} を選び, \check{J}_{m+1} の代わりに \hat{J}_{m+1} を最小化する $\hat{\boldsymbol{w}}_{m+1}$ を求める。このようにすると, $\|\hat{\boldsymbol{w}}_{m+1}\|^2 < \alpha$ を満たしていれば, MM アルゴリズム (majorize-minimization algorithm) (Lange, 2016) の論理により, $\hat{\boldsymbol{x}}_{0,m+1} = \hat{\boldsymbol{x}}_{0,m} + \mathbf{X}_m \hat{\boldsymbol{w}}_{m+1}$ に対して

$$(3.17) \quad J(\hat{\boldsymbol{x}}_{0,m+1}) \leq \hat{J}_{m+1}(\hat{\boldsymbol{w}}_{m+1}) \leq \hat{J}_{m+1}(\mathbf{0}) = J(\hat{\boldsymbol{x}}_{0,m})$$

が成り立つ (Nakano, 2021)。式(3.16)の $\check{J}_{m+1}(\boldsymbol{w}_m)$ を省略せずに書くと

$$(3.18) \quad \hat{J}_{m+1}(\boldsymbol{w}_{m+1}) = \frac{\sigma_{m+1}^2}{2} \boldsymbol{w}_{m+1}^\top \boldsymbol{w}_{m+1} \\ + \frac{1}{2}(\hat{\boldsymbol{x}}_{0,m} - \bar{\boldsymbol{x}}_{0,b} + \mathbf{X}_m \boldsymbol{w}_{m+1})^\top \mathbf{P}_b^{-1}(\hat{\boldsymbol{x}}_{0,m} - \bar{\boldsymbol{x}}_{0,b} + \mathbf{X}_m \boldsymbol{w}_{m+1}) \\ + \frac{1}{2} \sum_{k=1}^K [\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_{0,b}) - \boldsymbol{\Gamma}_{k,m} \boldsymbol{w}_{m+1}]^\top \mathbf{R}^{-1} [\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_{0,b}) - \boldsymbol{\Gamma}_{k,m} \boldsymbol{w}_{m+1}].$$

\hat{J}_{m+1} の勾配は,

$$(3.19) \quad \nabla \hat{J}_{m+1} = \sigma_{m+1}^2 \boldsymbol{w}_{m+1} + \mathbf{X}_m^\top \mathbf{P}_b^{-1}(\hat{\boldsymbol{x}}_{0,m} - \bar{\boldsymbol{x}}_{0,b} + \mathbf{X}_m \boldsymbol{w}_{m+1}) \\ - \sum_{k=1}^K \boldsymbol{\Gamma}_{k,m}^\top \mathbf{R}^{-1} [\boldsymbol{y}_k - \boldsymbol{g}_k(\hat{\boldsymbol{x}}_{0,m}) - \boldsymbol{\Gamma}_{k,m} \boldsymbol{w}_{m+1}]$$

となるので, \hat{J}_{m+1} は I_N を N 次の単位行列として,

$$(3.20) \quad \hat{\boldsymbol{w}}_{m+1} = - \left(\sigma^2 I_N + \mathbf{X}_m^T \mathbf{P}_b^{-1} \mathbf{X}_m + \sum_{k=1}^K \Gamma_{k,m}^T \mathbf{R}^{-1} \Gamma_{k,m} \right)^{-1} \\ \times \left(\mathbf{X}_m^T \mathbf{P}_b^{-1} [\hat{\boldsymbol{x}}_{0,m} - \bar{\boldsymbol{x}}_{0,b}] - \sum_{k=1}^K \Gamma_{k,m}^T \mathbf{R}^{-1} [\mathbf{y}_k - \mathbf{g}_k(\hat{\boldsymbol{x}}_{0,m})] \right)$$

で最小となる. したがって,

$$(3.21) \quad \hat{\boldsymbol{x}}_{0,m+1} = \hat{\boldsymbol{x}}_{0,m} + \mathbf{X}_m \hat{\boldsymbol{w}}_{m+1}$$

を $(m+1)$ 回目の推定値とすればよい. アルゴリズムをまとめると, Algorithm 1 のようになる.

Algorithm 1 アンサンブルを用いた変分法アルゴリズム.

$m := 0$ とする.

初期推定値 $\hat{\boldsymbol{x}}_{0,0}$ を設定する.

while unconverged **do**

式(3.1)を満たすようにアンサンブル $\{\boldsymbol{x}_{0,m}^{(1)}, \dots, \boldsymbol{x}_{0,m}^{(N)}\}$ を生成する.

式(3.8)により, 行列 $\Gamma_{k,m}$ を得る.

式(3.21)の $\hat{\boldsymbol{w}}_{m+1}$ を求める.

式(3.21)に従って, $(m+1)$ 回目の推定値 $\hat{\boldsymbol{x}}_{0,m+1}$ を得る.

$m := m+1$ とする.

end while

4. ガウス・ニュートン法との関係

Algorithm 1 は, ガウス・ニュートン法 (Gauss–Newton method; 以下 GN 法) あるいはその修正版であるレーベンバーク・マーカート法 (Levenberg–Marquardt method; 以下 LM 法) (Nocedal and Wright, 2006; 金森 他, 2016) のアンサンブルによる近似と見なすこともできる. GN 法や LM 法がよく用いられるのは, 以下のような非線型の最小二乗問題

$$(4.1) \quad J_l = \frac{1}{2} [\mathbf{y} - \mathbf{g}(\boldsymbol{x})]^T \mathbf{R}^{-1} [\mathbf{y} - \mathbf{g}(\boldsymbol{x})]$$

において, 観測 \mathbf{y} が与えられたときに J_l を最小化する \boldsymbol{x} を求める問題である. GN 法では, m ステップ目における最適解の推定値を $\hat{\boldsymbol{x}}_m$ として, $\hat{\boldsymbol{x}}_m$ の周りで \mathbf{g} の 1 次近似

$$(4.2) \quad \mathbf{g}(\boldsymbol{x}) \approx \mathbf{g}(\hat{\boldsymbol{x}}_m) + \mathbf{G}_m(\boldsymbol{x} - \hat{\boldsymbol{x}}_m)$$

を考える. 但し, \mathbf{G}_m は \mathbf{g} の $\hat{\boldsymbol{x}}_m$ におけるヤコビ行列である. 式(4.2)で式(4.1)を近似すると, 目的関数 J_l の近似

$$(4.3) \quad \check{J}_l = \frac{1}{2} [\mathbf{y} - \mathbf{g}(\hat{\boldsymbol{x}}_m) - \mathbf{G}_m(\boldsymbol{x} - \hat{\boldsymbol{x}}_m)]^T \mathbf{R}^{-1} [\mathbf{y} - \mathbf{g}(\hat{\boldsymbol{x}}_m) - \mathbf{G}_m(\boldsymbol{x} - \hat{\boldsymbol{x}}_m)]$$

が得られる. このとき, $\boldsymbol{x} = \hat{\boldsymbol{x}}_m$ における \check{J}_l の勾配, ヘッセ行列はそれぞれ,

$$(4.4) \quad \nabla \check{J}_l = -\mathbf{G}_m^T \mathbf{R}^{-1} [\mathbf{y} - \mathbf{g}(\hat{\boldsymbol{x}}_m)],$$

$$(4.5) \quad \nabla^2 \check{J}_l = \mathbf{G}_m^T \mathbf{R}^{-1} \mathbf{G}_m$$

となる．GN法は，式(4.4)の勾配，式(4.5)のヘッセ行列を使い，ニュートン・ラフソン法(Newton-Raphson method)の式を用いて最適解の推定値を更新する．すなわち，GN法の $(m+1)$ ステップ目の推定値 $\hat{\boldsymbol{x}}_{m+1}$ は，

$$(4.6) \quad \check{\boldsymbol{x}}_{m+1} = \hat{\boldsymbol{x}}_m + [\mathbf{G}_m^T \mathbf{R}^{-1} \mathbf{G}_m]^{-1} \mathbf{G}_m^T \mathbf{R}^{-1} [\mathbf{y} - \mathbf{g}(\hat{\boldsymbol{x}}_m)]$$

となる．LM法は式(4.3)の \check{J}_l に正則化項を加えた

$$(4.7) \quad \hat{J}_l = \frac{\lambda_m^2}{2} \mathbf{x}^T \mathbf{x} + \frac{1}{2} [\mathbf{y} - \mathbf{g}(\hat{\boldsymbol{x}}_m) - \mathbf{G}_m(\mathbf{x} - \hat{\boldsymbol{x}}_m)]^T \mathbf{R}^{-1} [\mathbf{y} - \mathbf{g}(\hat{\boldsymbol{x}}_m) - \mathbf{G}_m(\mathbf{x} - \hat{\boldsymbol{x}}_m)]$$

を考え，これを最小化する．この場合， $m+1$ ステップ目の推定値 $\hat{\boldsymbol{x}}_{m+1}$ は，

$$(4.8) \quad \hat{\boldsymbol{x}}_{m+1} = \hat{\boldsymbol{x}}_m + [\lambda_m^2 \mathbf{I}_d + \mathbf{G}_m^T \mathbf{R}^{-1} \mathbf{G}_m]^{-1} \mathbf{G}_m^T \mathbf{R}^{-1} [\mathbf{y} - \mathbf{g}(\hat{\boldsymbol{x}}_m)]$$

ようになる．式(4.1)の非線形最小二乗問題は，式(2.9)において右辺第一項の事前分布に相当する項を落とした場合に相当する．この場合，式(3.21)は

$$(4.9) \quad \hat{\boldsymbol{w}}_m = \left(\sigma^2 \mathbf{I}_N + \sum_{k=1}^K \mathbf{\Gamma}_m^T \mathbf{R}^{-1} \mathbf{\Gamma}_m \right)^{-1} \sum_{k=1}^K \mathbf{\Gamma}_m^T \mathbf{R}^{-1} [\mathbf{y}_k - \mathbf{g}_k(\hat{\boldsymbol{x}}_{0,m})]$$

となる．つまり，Algorithm 1は，各イテレーションにおいて関数 \mathbf{g}_k のヤコビ行列を求める代わりに $\mathbf{\Gamma}_m$ を用いた近似を行い，その近似に基づいてLM法を実行していると解釈することもできる．

Bocquet and Sakov (2013, 2014)では，GN法に基づいてAlgorithm 1と類似のアルゴリズムを導いており，これをiterative ensemble Kalman smoother (IEnKS)と呼んでいる．但し，IEnKSでは，事前分布の分散共分散行列 \mathbf{P}_b を初期アンサンブルの標本分散共分散行列

$$(4.10) \quad \mathbf{P}_b = \frac{1}{N-1} \mathbf{X}_0 \mathbf{X}_0^T$$

で与えており，また， \mathbf{x}_0 が

$$(4.11) \quad \mathbf{x}_0 = \bar{\mathbf{x}}_{0,b} + \mathbf{X}_0 \mathbf{w}^*$$

の形で書けると仮定している．通常，アンサンブルメンバー数 N は状態ベクトル \mathbf{x}_k の次元よりもはるかに小さい値となる．事前分布の分散共分散行列を式(4.10)で与えた場合， \mathbf{P}_b は非正則となるので，事前分布はアンサンブルメンバーの張る部分空間に分布し，

$$(4.12) \quad p(\mathbf{x}_0) \propto \exp \left(-\frac{1}{2} [\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b}]^T \mathbf{P}_b^- [\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b}] \right)$$

を満たす．但し， \mathbf{P}_b^- は \mathbf{P}_b のMoore-Penrose一般逆行列である．したがって，解くべき目的関数は

$$(4.13) \quad J(\mathbf{x}_0) = \frac{1}{2} (\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b})^T \mathbf{P}_b^- (\mathbf{x}_0 - \bar{\mathbf{x}}_{0,b}) + \frac{1}{2} \sum_{k=1}^K [\mathbf{y}_k - \mathbf{g}_k(\mathbf{x}_0)]^T \mathbf{R}^{-1} [\mathbf{y}_k - \mathbf{g}_k(\mathbf{x}_0)]$$

となる．式(4.10)が成り立つとき， \mathbf{X}_0 の特異値分解 $\mathbf{X}_0 = \mathbf{U}_0 \mathbf{\Lambda}_0 \mathbf{V}_0^T$ を考えると，

$$(4.14) \quad \mathbf{P}_b = \frac{1}{N-1} \mathbf{U}_0 \mathbf{\Lambda}_0^2 \mathbf{U}_0^T$$

なので，

$$(4.15) \quad \mathbf{X}_0^T \mathbf{P}_b^- \mathbf{X}_0 = (N-1) \mathbf{V}_0 \mathbf{V}_0^T = (N-1) \mathbf{I}_N$$

が成り立つ. これと式(3.10)の近似を組み合わせると, 式(4.13)は $\hat{\mathbf{w}}_m^*$ を m 回目のイテレーションにおける \mathbf{w}^* の推定値として,

$$(4.16) \quad \check{J}^*(\mathbf{w}^*) = \frac{N-1}{2} \mathbf{w}^{*\top} \mathbf{w}^* + \frac{1}{2} \sum_{k=1}^K [\mathbf{y}_k - \mathbf{g}_k(\hat{\mathbf{x}}_{0,m}) - \Gamma_{k,m}(\mathbf{w}^* - \hat{\mathbf{w}}_m^*)]^\top \mathbf{R}^{-1} [\mathbf{y}_k - \mathbf{g}_k(\hat{\mathbf{x}}_{0,m}) - \Gamma_{k,m}(\mathbf{w}^* - \hat{\mathbf{w}}_m^*)]$$

となる. これの $\mathbf{w}^* = \hat{\mathbf{w}}_m^*$ における勾配, ヘッセ行列はそれぞれ

$$(4.17) \quad \nabla \check{J}_m^* = (N-1) \hat{\mathbf{w}}_m^* - \sum_{k=1}^K \Gamma_{k,m}^\top \mathbf{R}^{-1} [\mathbf{y}_k - \mathbf{g}_k(\hat{\mathbf{x}}_{0,m})],$$

$$(4.18) \quad \nabla^2 \check{J}_m^* = (N-1) \mathbf{I}_N + \sum_{k=1}^K \Gamma_{k,m}^\top \mathbf{R}^{-1} \Gamma_{k,m}.$$

したがって, GN法にしたがって J の最小化を行う場合, 各イテレーションで

$$(4.19) \quad \mathbf{w}_{m+1}^* = \mathbf{w}_m^* - \left[(N-1) \mathbf{I}_N + \sum_{k=1}^K \Gamma_{k,m}^\top \mathbf{R}^{-1} \Gamma_{k,m} \right]^{-1} \left[(N-1) \mathbf{w}_m^* - \sum_{k=1}^K \Gamma_{k,m}^\top \mathbf{R}^{-1} [\mathbf{y}_k - \mathbf{g}_k(\hat{\mathbf{x}}_{0,m})] \right]$$

を行えばよく, IEnKSではこの式を用いる. 但しこの方法を用いた場合, 式(4.11)の仮定から解は初期アンサンブルメンバーの張る部分空間に制限されることになる.

5. 計数観測データのための拡張

次に, 時刻 t_k の状態 \mathbf{x}_k が与えられた下で観測 \mathbf{y}_k の各要素 y_{kj} ($j = 1, \dots, n$) がポアソン分布にしたがう場合について, アンサンブルシミュレーションの結果を用いた変分法アルゴリズムの導出を試みる. この場合の尤度は, \mathbf{x}_k に基づく y_{kj} の予測値を $h_{kj}(\mathbf{x}_k)$ として,

$$(5.1) \quad P(\mathbf{y}_k | \mathbf{x}_k) = \prod_{j=1}^n \frac{h_{kj}(\mathbf{x}_k)^{y_{kj}}}{y_{kj}!} \exp[-h_{kj}(\mathbf{x}_k)].$$

このとき, $h_{kj}(\mathbf{x}_k) \geq 0$ が各 j について成り立つ必要がある. ただ, 実際の応用においては, ポアソン分布にしたがう観測が非負の状態変数と結びついていることが多い. 例えば, 1節で触れた宇宙科学分野の紫外線観測の例では, 観測される光子の数の期待値が発光源となる流体の密度と比例し, 状態変数である流体密度が非負であることから, 観測の期待値も非負となる. このような場合, 観測と対応付けられる状態変数が非負になっており, 状態変数が非負であれば, 観測の期待値 $h_{kj}(\mathbf{x}_k)$ は自ずと非負になる状況になっている. 以下では, 話を簡単にするために状態ベクトル \mathbf{x}_k の全要素が非負であると仮定する. 関数 h_k は, \mathbf{x}_k の全要素が非負であれば必ず $h_{kj}(\mathbf{x}_k)$ の全要素が非負になるような関数であると仮定して, $h_{kj}(\mathbf{x}_k)$ が非負になることを保証しておく. 初期値 \mathbf{x}_0 については, 各要素 $x_{0\ell}$ ($\ell = 1, \dots, d$) が正の値を取るように

$$(5.2) \quad \xi_\ell = \log x_{0\ell}$$

と置き, $\boldsymbol{\xi} = (\xi_1 \cdots \xi_d)^\top$ を推定することで, \mathbf{x}_0 の推定を行うことにする. $\boldsymbol{\xi}$ の事前分布は, 正規分布

$$(5.3) \quad p(\boldsymbol{\xi}) = \frac{1}{\sqrt{(2\pi)^d |\mathbf{P}_\xi|}} \exp\left(-\frac{1}{2}[\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}_b]^\top \mathbf{P}_\xi^{-1} [\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}_b]\right)$$

で与える。このとき、観測の時系列 $\mathbf{y}_1, \dots, \mathbf{y}_K$ が与えられた下での $\boldsymbol{\xi}_0$ の事後分布は

$$(5.4) \quad p(\boldsymbol{\xi} | \mathbf{y}_1, \dots, \mathbf{y}_K) \propto \frac{1}{\sqrt{(2\pi)^d |\mathbf{P}_\xi|}} \exp\left(-\frac{1}{2}[\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}_b]^\top \mathbf{P}_\xi^{-1} [\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}_b]\right) \\ \times \prod_{k=1}^K \prod_{j=1}^n \frac{h_{kj}(\mathbf{x}_k)^{y_{kj}}}{y_{kj}!} \exp[-h_{kj}(\mathbf{x}_k)]$$

を満たし、対数事後確率密度は C を定数として

$$(5.5) \quad \log p(\boldsymbol{\xi} | \mathbf{y}_1, \dots, \mathbf{y}_K) \\ = -\frac{1}{2}[\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}_b]^\top \mathbf{P}_\xi^{-1} [\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}_b] + \sum_{k=1}^K \sum_{j=1}^n [y_{kj} \log h_{kj}(\mathbf{x}_k) - \log y_{kj}! - h_{kj}(\mathbf{x}_k)] + C$$

と書ける。対数事後確率密度 $\log p(\boldsymbol{\xi} | \mathbf{y}_1, \dots, \mathbf{y}_K)$ を最大化するには、以下の目的関数を最小化すればよい：

$$(5.6) \quad J_P(\boldsymbol{\xi}) = \frac{1}{2}[\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}_b]^\top \mathbf{P}_\xi^{-1} [\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}_b] + \sum_{k=1}^K \sum_{j=1}^n [\log y_{kj}! + h_{kj}(\mathbf{x}_k) - y_{kj} \log h_{kj}(\mathbf{x}_k)].$$

ここで、式(2.8)と同様に、

$$(5.7) \quad \mathbf{h}_k(\mathbf{x}_k) = \mathbf{z}_k(\mathbf{z}_0)$$

を満たす関数 \mathbf{z}_k を定義する。 \mathbf{z}_k は式(2.7)のような合成関数の形で書くと、

$$(5.8) \quad \mathbf{z}_k(\boldsymbol{\xi}) = \mathbf{h}_k \circ \mathbf{f}_k \circ \mathbf{f}_{k-1} \circ \dots \circ \mathbf{f}_1 \circ \mathbf{e}(\boldsymbol{\xi})$$

となる。但し、関数 \mathbf{e} は

$$(5.9) \quad \mathbf{e}_k(\boldsymbol{\xi}) = \begin{pmatrix} \exp(\xi_1) \\ \vdots \\ \exp(\xi_d) \end{pmatrix}$$

となる d 次元関数である。 $\mathbf{z}_k(\boldsymbol{\xi})$ の j 番目の要素を $z_{kj}(\boldsymbol{\xi})$ と書くことにすると、目的関数 J_P は

$$(5.10) \quad J_P(\boldsymbol{\xi}) = \frac{1}{2}[\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}_b]^\top \mathbf{P}_\xi^{-1} [\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}_b] + \sum_{k,j} [\log y_{kj}! + z_{kj}(\boldsymbol{\xi}) - y_{kj} \log z_{kj}(\boldsymbol{\xi})]$$

と書き直せる。

$(m+1)$ 回目のイテレーションでのアンサンブルシミュレーションを実行するために、 $\boldsymbol{\xi}$ について N 個の異なる値 $\{\boldsymbol{\xi}_m^{(1)}, \dots, \boldsymbol{\xi}_m^{(N)}\}$ を用意する。但し、 $\boldsymbol{\xi}$ の推定値を $\hat{\boldsymbol{\xi}}_m$ として、式(3.1)と同様に、

$$(5.11) \quad \frac{1}{N} \sum_{i=1}^N \boldsymbol{\xi}_m^{(i)} = \hat{\boldsymbol{\xi}}_m$$

が成り立つようにそれぞれの $\boldsymbol{\xi}_m^{(i)}$ を生成するものとする。 $\boldsymbol{\xi}$ については、各アンサンブルメンバーの $\hat{\boldsymbol{\xi}}_m$ からの差を並べた $d \times N$ 行列

$$(5.12) \quad \Xi_m = (\xi_m^{(1)} - \hat{\xi}_m \quad \cdots \quad \xi_m^{(N)} - \hat{\xi}_m)$$

を定義したとき、式(3.4)と同様に

$$(5.13) \quad \xi = \hat{\xi}_m + \Xi_m \mathbf{w}_{m+1}$$

の形で書けるものとする。アンサンブルシミュレーションの結果を用いて N 次元の行ベクトル $\zeta_{k,j,m}^T$ を

$$(5.14) \quad \zeta_{k,j,m}^T = (z_{kj}(\xi_m^{(1)}) - z_{kj}(\hat{\xi}_m) \quad \cdots \quad z_{kj}(\xi_m^{(N)}) - z_{kj}(\hat{\xi}_m))$$

のように定義し、式(3.10)と同様に、関数 z_{kj} を

$$(5.15) \quad z_{kj}(\xi) \approx z_{kj}(\bar{\xi}_b) + \zeta_{k,j,m}^T \mathbf{w}_{m+1}$$

で近似すると、式(5.10)は、

$$(5.16) \quad \begin{aligned} \check{J}_P = & \frac{1}{2} (\hat{\xi}_m - \bar{\xi}_b + \Xi_m \mathbf{w}_{m+1})^T \mathbf{P}_\xi^{-1} (\hat{\xi}_m - \bar{\xi}_b + \Xi_m \mathbf{w}_{m+1}) \\ & + \sum_{k,j} [\log y_{kj}! + z_{kj}(\bar{\xi}_b) + \zeta_{k,j,m}^T \mathbf{w}_{m+1} - y_{kj} \log(z_{kj}(\bar{\xi}_b) + \zeta_{k,j,m}^T \mathbf{w}_{m+1})]. \end{aligned}$$

各イテレーションで解が改善されるようにするために、式(3.19)と同じく 2 次の項を追加すると、

$$(5.17) \quad \begin{aligned} \hat{J}_P = & \frac{\sigma_{m+1}^2}{2} \mathbf{w}_{m+1}^T \mathbf{w}_{m+1} + \frac{1}{2} (\hat{\xi}_m - \bar{\xi}_b + \Xi_m \mathbf{w}_{m+1})^T \mathbf{P}_\xi^{-1} (\hat{\xi}_m - \bar{\xi}_b + \Xi_m \mathbf{w}_{m+1}) \\ & + \sum_{k,j} [\log y_{kj}! + z_{kj}(\bar{\xi}_b) + \zeta_{k,j,m}^T \mathbf{w}_{m+1} - y_{kj} \log(z_{kj}(\bar{\xi}_b) + \zeta_{k,j,m}^T \mathbf{w}_{m+1})]. \end{aligned}$$

\hat{J}_P の勾配は、

$$(5.18) \quad \nabla \hat{J}_P = \Xi_m^T \mathbf{P}_\xi^{-1} [\hat{\xi}_m - \bar{\xi}_b] + \sum_{k,j} \left[\zeta_{k,j,m} - \frac{y_{kj}}{z_{kj}(\bar{\xi}_b)} \zeta_{k,j,m} \right],$$

ヘッセ行列は、

$$(5.19) \quad \nabla^2 \hat{J}_P = \sigma_{m+1}^2 \mathbf{I}_N + \Xi_m^T \mathbf{P}_\xi^{-1} \Xi_m + \sum_{k,j} \left[\frac{y_{kj}}{z_{kj}(\bar{\xi}_b)^2} \zeta_{k,j,m} \zeta_{k,j,m}^T \right].$$

したがって、4 節の考え方にに基づき、ニュートン・ラフソン法で \mathbf{w}_{m+1} の推定値 $\hat{\mathbf{w}}_{m+1}$ を求めると、

$$(5.20) \quad \begin{aligned} \hat{\mathbf{w}}_{m+1} = & - \left(\sigma_{m+1}^2 \mathbf{I}_N + \Xi_m^T \mathbf{P}_\xi^{-1} \Xi_m + \sum_{k,j} \left[\frac{y_{kj}}{z_{kj}(\bar{\xi}_b)^2} \zeta_{k,j,m} \zeta_{k,j,m}^T \right] \right)^{-1} \\ & \times \left(\Xi_m^T \mathbf{P}_\xi^{-1} [\hat{\xi}_m - \bar{\xi}_b] + \sum_{k,j} \left[\zeta_{k,j,m} - \frac{y_{kj}}{z_{kj}(\bar{\xi}_b)} \zeta_{k,j,m} \right] \right). \end{aligned}$$

あとは、式(3.21)と同様に

$$(5.21) \quad \hat{\xi}_{m+1} = \hat{\xi}_m + \Xi_m \hat{\mathbf{w}}_{m+1}$$

で $\hat{\xi}_{m+1}$ を求めれば、 $\hat{\mathbf{x}}_{0,m+1} = e(\hat{\xi}_{m+1})$ が \mathbf{x}_0 の $(m+1)$ 回目の推定値となる。以上のように、

尤度がポアソン分布で与えられた場合であっても、関数 z_k の各要素 z_{kj} を式 (5.15) で近似すれば、勾配、ヘッセ行列を近似的に求めることができるので、アンサンブルを用いた変分法で \mathbf{x}_0 の推定ができると考えられる。

6. 1次元移流拡散モデルによる実験

前節で議論したアルゴリズムの有効性を確認するために、1次元の流体をポアソンノイズ付きで観測するという簡単な設定で数値実験を行った。流体をポアソンノイズ付きで観測するというのは、宇宙プラズマの撮像観測で現れる問題を単純化したものである。1次元流体の移流拡散方程式は、

$$(6.1) \quad \frac{\partial \rho}{\partial t} = -u \frac{\partial \rho}{\partial x} + \nu \frac{\partial^2 \rho}{\partial x^2}$$

で与えられる。但し、 ρ は流体の密度、 t は時間、 x は位置、 u は速度、 ν は拡散係数である。 x 方向の範囲は $0 \leq x < 200$ とし、 $\rho(x=200) = \rho(x=0)$ の周期境界条件を設定する。また、 $u=2$ 、 $\nu=2$ で固定することとし、 ρ の初期値のみ未知とする。式 (6.1) は、 $\Delta x=2$ 、 $\Delta t=0.2$ の間隔で離散化し、Lax-Wendroff スキーム (例えば、藤井, 1994) で解く。

推定に用いる擬似データは、初期値を

$$(6.2) \quad \rho_0(x) = \begin{cases} -\frac{20}{900}(x-20)(x-80), & (\text{when } 20 < x < 80), \\ 0, & (\text{otherwise}), \end{cases}$$

と与えた上でモデルを実行し、生成した。つまり、これが推定すべき「真の初期値」ということになる。上述のように、 $0 \leq x < 200$ の範囲を $\Delta x=2$ の間隔で離散化しているので、実際に推定すべき未知変数の数は 100 ということになる。観測は $2\Delta x$ おきの空間間隔で、時間間隔 $20\Delta t$ おきに $t=100\Delta t=20$ まで取得されるものとし、各観測にはポアソン分布にしたがうノイズが付加されるものとする。

推定に当たっては、 ρ_0 の各格子点における値 $\rho_{0\ell}$ の対数 $\xi_\ell = \log \rho_{0\ell}$ を 100 個並べた

$$(6.3) \quad \boldsymbol{\xi} = (\xi_1 \quad \cdots \quad \xi_{100})^\top$$

について、事前分布を平均 $\mathbf{0}$ の正規分布

$$(6.4) \quad p(\boldsymbol{\xi}) = \frac{1}{\sqrt{(2\pi)^d |\mathbf{P}_\xi|}} \exp\left(-\frac{1}{2} \boldsymbol{\xi}^\top \mathbf{P}_\xi^{-1} \boldsymbol{\xi}\right)$$

で与える。分散共分散行列 \mathbf{P}_ξ は

$$(6.5) \quad \boldsymbol{\xi}^\top \mathbf{P}_\xi^{-1} \boldsymbol{\xi} = \frac{1}{\alpha^2} \left[\varepsilon^2 \boldsymbol{\xi}^\top \boldsymbol{\xi} + \sum_{\ell=1}^{100} \left(\xi_\ell - \frac{\xi_{\ell-1} + \xi_{\ell+1}}{2} \right)^2 \right]$$

を満たすように設定する。但し、 $\xi_0 = \xi_{100}$ 、 $\xi_{101} = \xi_1$ であり、また本研究では $\varepsilon=0.01$ 、 $\alpha=0.02$ と置いている。また、 $(m+1)$ 回目のイテレーションで用いるアンサンブルは、 m 回目のイテレーションで得られた推定値 $\hat{\boldsymbol{\xi}}_m$ を平均とする正規分布

$$(6.6) \quad \mathcal{N}(\hat{\boldsymbol{\xi}}_m, \delta^2 \mathbf{P}_\xi) = \frac{1}{\sqrt{(2\pi)^d |\delta^2 \mathbf{P}_\xi|}} \exp\left(-\frac{1}{2\delta^2} [\boldsymbol{\xi} - \hat{\boldsymbol{\xi}}_m]^\top \mathbf{P}_\xi^{-1} [\boldsymbol{\xi} - \hat{\boldsymbol{\xi}}_m]\right)$$

から生成する。

以上のような設定の下、5節で導入したアルゴリズムで初期値 ρ_0 の推定を行った。図1は、

イテレーションごとの推定値の対数尤度 $\sum_k \log p(\mathbf{y}_k | \hat{\xi}_m)$ を示したもので、実線がアンサンプルメンバー数を $N = 100$ としたとき、点線が $N = 50$ としたときの結果を示している。ここで示している対数尤度とは、式(5.10)の目的関数 J_P そのものではなく、式(5.10)右辺第2項の符号を反転させたものに対応する。アンサンプルメンバー数に関わらず、対数尤度はイテレーションごとに増大しており、アルゴリズムが期待どおりに動いていることが確認できる。

図1を詳しく見ると、 $N = 100$ の場合には6回程度のイテレーションでほぼ収束しており、 $N = 50$ の場合は8回程度のイテレーションで収束している。計算コストのほとんどがアンサンプルシミュレーションの実行に因るとすると、 $N = 50$ とした場合、 $N = 100$ とした場合と比べ、1回のイテレーションに必要な計算コストは半分になる。収束に至るまでに要したイテレーションの回数は、 $N = 100$ の場合、 $N = 50$ とした場合の3/4倍程度であったため、今回の実験では、未知変数の数(100)よりも少ないアンサンプルメンバー数($N = 50$)の場合の方が効率的だったということになる。

図2は、6回のイテレーションによる初期値の推定結果であり、黒の実線、黒の点線がそれ

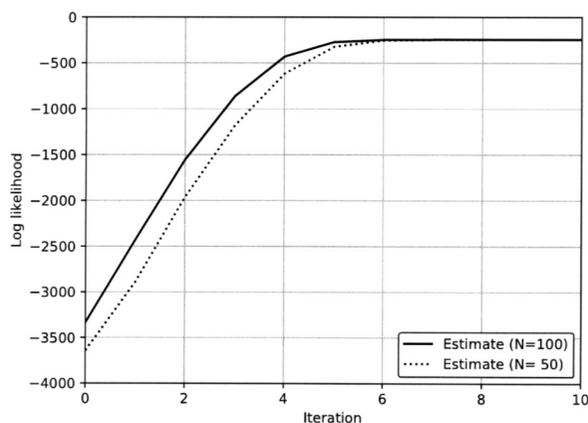


図1. イテレーションごとの対数尤度の変化. $N = 100$ の場合の結果を実線, $N = 50$ の場合の結果を点線で示す.

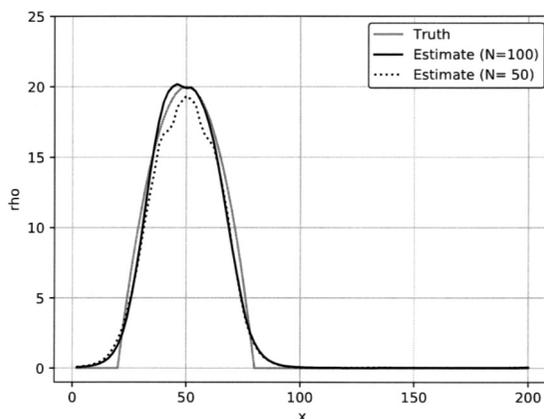


図2. 6回のイテレーションによる初期値の推定結果. $N = 100$ の場合の結果を黒の実線, $N = 50$ の場合の結果を黒の点線, 真の初期値を灰色の実線で示す.

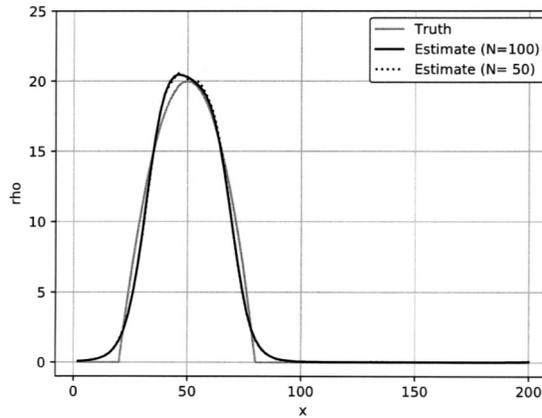


図 3. 10 回のイテレーションによる初期値の推定結果. 各線の意味は図 2 に準じる.

ぞれ $N = 100$, $N = 50$ の場合の結果を示している. また, 灰色の実線は真の初期値である. $N = 100$ の場合, 真の値に近い推定結果が得られている. $N = 50$ の結果を見ると, $N = 100$ の場合と比較して真の値との差異が大きいが, 大まかな構造は捉えられており, 期待どおりアルゴリズムが動いていると言える. 図 3 は, イテレーション 10 回の時点での推定結果で, それぞれの線の意味は図 2 と同じである. $N = 100$ の場合は, イテレーション 6 回の結果とほぼ変わらない. 一方, $N = 50$ の場合, イテレーション 6 回では収束に至っていなかったが, 10 回のイテレーションで $N = 100$ の場合とほぼ同じ値に収束していることが確認できる. $N = 100$ の場合も, $N = 50$ の場合も同じ目的関数の最小化を行っているが, 目的関数が同じであれば N に関わらず同じ結果が得られており, このこともアルゴリズムが正しく動いていることを示している.

比較のために, 観測モデルを正規分布とした場合の実験も行った. 目的関数は

$$(6.7) \quad J_G(\xi) = \frac{1}{2}[\xi - \bar{\xi}_b]^T P_\xi^{-1}[\xi - \bar{\xi}_b] + \frac{1}{2} \sum_{k=1}^K [y_k - z_k(\xi)]^T R^{-1}[y_k - z_k(\xi)]$$

のように設定し, 同じ擬似観測データを与えた上で, 3 節のアルゴリズムを用いて目的関数 J_G を最小化する ξ を求め, 初期値 ρ_0 を推定した. ポアソン分布の場合と異なり, 観測誤差共分散行列 R を設定する必要があるが, R の全要素を推定するのは困難であるため, $R = \text{diag}(s^2, s^2, \dots, s^2)$ のようにすべての対角要素が同じ値を取る対角行列を仮定した. R の対角要素は $s = 0.5, 1.0, 1.5, \dots$ と s を増やして ξ の推定を繰り返し, 収束後の対数尤度 $\sum_k \log p(y_k | \hat{\xi})$ が最大となった値 ($s = 2$ つまり $s^2 = 4$) を選択した. 図 4 が, イテレーションごとの対数尤度 $\sum_k \log p(y_k | \hat{\xi}_m)$ であり, 図 1 と同様に実線がアンサンブルメンバー数を $N = 100$ としたとき, 点線が $N = 50$ としたときの結果である. 観測データがポアソン分布にしたがっているため, 図 4 の正規分布に基づく対数尤度は, 図 1 と結果と比較すると, 計算が収束しても値が小さくなる. また, 収束もやや遅くなっているように見える. 図 5 は, 10 回のイテレーションによる初期値の推定結果である. 大まかな構造は捉えられているが, 図 3 の結果と比較するとピークの付近で ρ の値を過大評価している. これは, 本来ポアソン分布である $p(y_k | x_k)$ を無理に正規分布と仮定して推定したことによるバイアスが現れているものと考えられる. このように, システムの状態が与えられた下でポアソン分布にしたがう計数データを, 正規分布の仮定の下で同化すると必ずしもよい推定結果が得られない. 本稿で導入したアルゴ

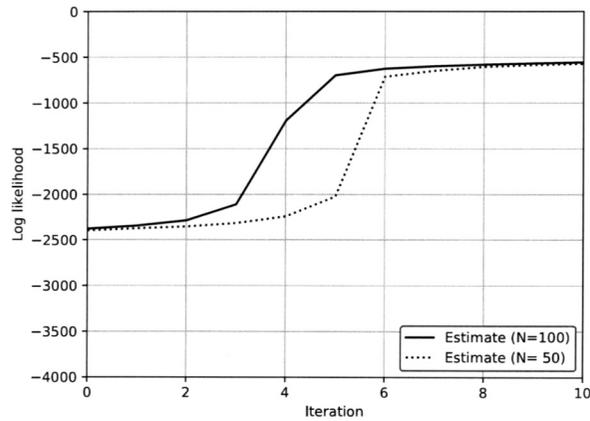


図 4. 観測モデルを正規分布と設定した場合のイテレーションごとの対数尤度の変化。
 $N = 100$ の場合の結果を実線, $N = 50$ の場合の結果を点線で示す。

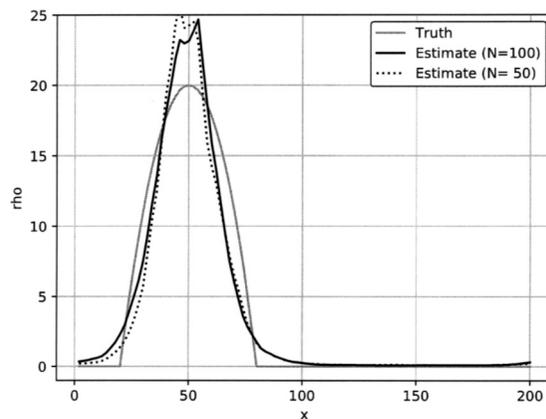


図 5. 観測モデルを正規分布と設定した場合の 10 回のイテレーションによる初期値の推定結果。各線の意味は図 2 に準じる。

リズムは, このような場合に有効であると考えられる。

7. おわりに

本稿では, アンサンブルシミュレーションの結果を利用して 4 次元変分法の問題を解くアプローチが, 尤度が正規分布で与えられる場合だけでなく, ポアソン分布で与えられる場合にも適用可能であることを示した. 正規分布の場合と同じく, ポアソン分布の場合においても, 対数事後確率密度関数から導出される目的関数の勾配, ヘッセ行列をアンサンブルシミュレーションの結果を利用して近似すれば, ニュートン・ラフソン法の式に基づいて入力(初期値)を求めることができる. 本稿ではポアソン分布のみについて議論したが, 同様のアルゴリズムは, 正規分布, ポアソン分布以外にも様々な分布に対して導出できる可能性がある. アンサンブルによる変分法アルゴリズムは, シミュレーションモデルをブラックボックスとして扱うことができ, 実装も容易なので, 他の分布にも適用できれば応用範囲は大きく広がると考えられる。

謝 辞

本研究を進めるにあたっては、科学研究費補助金基盤研究 B (課題番号: 17H01704) および基盤研究 A (課題番号: 17H00728) の助成を受けた。ここに感謝の意を表する。

参 考 文 献

- 淡路敏之, 蒲地政文, 池田元美, 石川洋一 (2009). 『データ同化-観測・実験とモデルを融合するイノベーション』, 京都大学学術出版会, 京都.
- Bocquet, M. and Sakov, P. (2013). Joint state and parameter estimation with iterative ensemble Kalman smoother, *Nonlinear Processes in Geophysics*, **20**, 803–818, <http://dx.doi.org/10.5194/npg-20-803-2013>.
- Bocquet, M. and Sakov, P. (2014). An iterative ensemble Kalman smoother, *Quarterly Journal of the Royal Meteorological Society*, **140**, 1521–1535, <http://dx.doi.org/10.1002/qj.2236>.
- Evensen, G. (2018). Analysis of iterative ensemble smoothers for solving inverse problems, *Computational Geosciences*, **22**, 885–908, <http://dx.doi.org/10.1007/s10596-018-9731-y>.
- Evensen, G. and van Leeuwen, P. J. (2000). An ensemble Kalman smoother for nonlinear dynamics, *Monthly Weather Review*, **128**, 1852–1867.
- 藤井孝蔵 (1994). 『流体力学の数値計算法』, 東京大学出版会, 東京.
- Gu, Y. and Oliver, D. S. (2007). An iterative ensemble Kalman filter for multiphase fluid flow data assimilation, *SPE Journal*, **12**, 438–446.
- 金森敬文, 鈴木大慈, 竹内一郎, 佐藤一誠 (2016). 『機械学習のための連続最適化』, 講談社, 東京.
- Lange, K. (2016). *MM Optimization Algorithms*, SIAM, Philadelphia.
- Liu, C., Xiao, Q. and Wang, B. (2008). An ensemble-based four-dimensional variational data assimilation scheme. Part I: Technical formulation and preliminary test, *Monthly Weather Review*, **136**, 3363–3373.
- Lorenc, A. C. (2003). The potential of the ensemble Kalman filter for NWP—a comparison with 4D-Var, *Quarterly Journal of the Royal Meteorological Society*, **129**, 3183–3203.
- Minami, T., Nakano, S., Lesur, V., Takahashi, F., Matsushima, M., Shimizu, H., Nakashima, R., Taniguchi, H. and Toh, H. (2020). A candidate secular variation model for IGRF-13 based on MHD dynamo simulation and 4D-EnVar data assimilation, *Earth Planets and Space*, **72**, p.136, <http://dx.doi.org/10.1186/s40623-020-01253-8>.
- Nakano, S. (2021). Behavior of the iterative ensemble-based variational method in nonlinear problems, *Nonlinear Processes in Geophysics*, **28**, 93–109, <http://dx.doi.org/10.5194/npg-28-93-2021>.
- Nakano, S., Fok, M.-C., Brandt, P. C. and Higuchi, T. (2014). Estimation of temporal evolution of the helium plasmasphere based on a sequence of IMAGE/EUV images, *Journal of Geophysical Research*, **119**, 3708–3723, <http://dx.doi.org/10.1002/2013JA019734>.
- Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*, 2nd ed., Springer, New York.
- Saito, M. M., Nishiura, H. and Higuchi, T. (2018). Reconstructing the transmission dynamics of rubella in Japan, 2012–2013, *PLoS ONE*, **13**, p. e0205889, <http://dx.doi.org/10.1371/journal.pone.0205889>.
- van Leeuwen, P. J. and Evensen, G. (1996). Data assimilation and inverse methods in terms of a probabilistic formulation, *Monthly Weather Review*, **124**, 2898–2913.
- Zupanski, M. (2005). Maximum likelihood ensemble filter: Theoretical aspects, *Monthly Weather Review*, **133**, 1710–1726.

Ensemble-based Variational Data Assimilation Approach and Its Extension for Count Data

Shin'ya Nakano^{1,2,3}

¹The Institute of Statistical Mathematics

²Center for Data Assimilation Research and Applications,
Joint Support-Center for Data Science Research

³Department of Statistical Science, School of Multidisciplinary Sciences,
The Graduate University for Advanced Studies, SOKENDAI

Ensemble-based variational approaches are a class of data assimilation methods which solves four-dimensional variational data assimilation problems by using ensemble simulations under various initial conditions and parameter settings. In contrast with the adjoint method, which is usually employed for four-dimensional variational data assimilation, these ensemble-based methods can easily be implemented without editing a simulation code allowing the model to be treated as a black box. A limitation of existing ensemble-based methods is that they are derived assuming that the conditional distributions of observations given the system states are Gaussian. Hence, they are not immediately applicable to observations that obey other distributions. This study derives an ensemble-based algorithm applicable for data assimilation into a black-box simulation model when the conditional distributions of observations are given by Poisson distributions.