DATA ASSIMILATION WITH MONTE CARLO MIXTURE KALMAN FILTER TOWARD SPACE WEATHER FORECASTING

Tomoyuki Higuchi

The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku, Tokyo 106-8569, Japan

ABSTRACT

Data Assimilation is a technique for a synthesis of information from a dynamic (numerical) model and observation data. It is an emerging area in earth sciences, particularly oceanography, stimulated by recent improvements in computational and modeling capabilities and the increase in the amount of available observations. Past studies for data assimilation employed a linear Gaussian state space model and applied Kalman filter. The Kalman filter based methods, however, do not allow for the strong nonlinear and/or non-Gaussian disturbance behaviors. We develop a new time dependent inversion method, which we call Monte Carlo mixture Kalman filter (MCMKF). A basic idea of MCMKF is as follows: (1) we prepare a finite number of competing state space models, each of which follows a different state space model, (2) we introduce a switching structure among these competing models. We address a capability of applying MCMKF to several topics in space physics.

1. DATA ASSIMILATION

Obtaining an accurate prediction of the upper ocean thermal structure is one of the important issue in modeling studies of the ocean circulation (and Earth 's environment). A numerical ocean model-based experiment, i.e., simulation study has been conducted for this purpose. Dependent variables of the numerical ocean model can be considered as stochastic variables due to the uncertainty in the initial and boundary conditions and the imperfection of the numerical model. A natural idea to compensate for such insufficient information only via simulations is to combine observations with numerical models. Hence, a reasonable way of blending a numerical (physical) model and observation is now becoming a central issue in the earth science community.

Data Assimilation is a technique for a synthesis of information from a dynamic (numerical) model and observation data [1]. In statistical sense, data assimilation supposes two models: system model and observation model. The system and observation model correspond to large-scale numerical model-based simulations and large-scale satellite- and/or ground-based measurement systems, respectively. The data assimilation can be therefore formulated in the state space model (SSM) [2, 3] as follows:

System model

$$\boldsymbol{x}_n = F_n \boldsymbol{x}_{n-1} + \boldsymbol{v}_n, \qquad \boldsymbol{v}_n \sim N(\boldsymbol{0}, Q) \quad (1)$$

Observation model

$$\boldsymbol{y}_n = H_n \boldsymbol{x}_n + \boldsymbol{w}_n, \qquad \boldsymbol{w}_n \sim N(\boldsymbol{0}, R), \quad (2)$$

where $N(\boldsymbol{0}, Q)$ and $N(\boldsymbol{0}, R)$ denote a normal distribution with a mean $\boldsymbol{0}$ and variance-covariance matrix of Q and R, respectively. The state of the ocean at time n is represented by a state vector $\boldsymbol{x}_n$ of which dimension is n_x. $\boldsymbol{y}_n$ is an observation (measurement) vector at time n. Its dimension is denoted by n_y. In data assimilation, n_x is huge such as $n_x \approx 10^5 \sim 10^6$. In contrast, n_y is much smaller than n_x ($n_x \gg n_y$), and then to get an optimal solution of $\boldsymbol{x}_n$ becomes a time-dependent inversion problem. It should be noticed that n_y is extreamely larger than those in the conventional state space models. For example, n_y exceeds a few thousands.

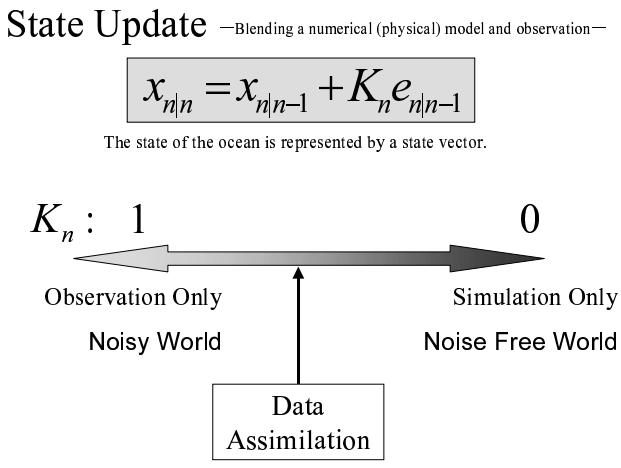

Fig. 1. Schematic Representation

Figure 1 shows a schematic representation of the data assimilation concept with a case of $n_x = n_y = 1$. At each time the state vector is updated according to this scheme (called the filtering step in the Kalman filter). $\boldsymbol{x}_{n|n-1}$ is a

state estimation at time n only via simulations. $\boldsymbol{e}_{n|n-1}$ is a prediction error between an actual observation and predictive value of the observation based on the result of simulations. K_n is a trade off parameter to control how the simulation model accommodates an actual observation. K_n is called the Kalman gain. When $K_n = 0$, an actual observation has no effect on a simulation process. In this case we totally rely on the simulation result. On the other hand, when $K_n = 1$, any discrepancy between the predictive and real values of observations is perfectly adjusted. In this case, it is difficult to identify a dynamics inherent to a simulation model from an estimation of the state vector, because a state vector is highly sensitive to the observation errors.

The SSM has given a platform in nonstationary time series and control studies for three decades after Kalman [3]. The Network Inversion Filter (NIF) [4] which is now recognized as one of the standard techniques in the time-dependent inversion to reveal the whole time history of fault slips event with an analysis of GPS network data, is also defined by the SSM. In the meanwhile, many phenomena in earth sciences have a nonlinearity and come along with non-Gaussian fluctuations. In particular, most of problems arising in space physics have to be discussed in this context, because a media in space, namely, plasma produces a wide variety of nonlinear and non-Gaussian phenomena. Prior to a realization of a space weather forecasting, the nonlinear non-Gaussian data assimilation method is required to evaluate how much the numerical model-based approach can describe the real world phenomena. The motivation of this work is to develop a new technique for non-linear non-Gaussian data assimilation method. Evensen [5] proposed the assimilation method for strongly nonlinear problem, which is usually called the ensemble Kalman filter (EnKF). The Monte Carlo method is used to estimate the variance of the prior probability distribution function (PDF). However, the EnKF also assumes that all the required PDFs are Gaussian. It should be noticed that the EnKF is different from the extended Kalman filter (EKF) which has been applied to weakly nonlinear problems [2].

2. MODEL AND BASIC IDEA

2.1. Conditional dynamic linear model (CDLM)

One of the generalization to develop the non-Gaussian nonlinear data assimlation is to introduce the conditional dynamic linear model (CDLM) [6]. The CDLM can be defined as:

$$\boldsymbol{x}_n = F_n(I_n)\boldsymbol{x}_{n-1} + \boldsymbol{v}_n, \tag{3}$$

$$\boldsymbol{y}_n = H_n(I_n)\boldsymbol{x}_n + \boldsymbol{w}_n \tag{4}$$

where $\boldsymbol{v}_n \sim N(\boldsymbol{0}, Q_n(I_n))$ and $\boldsymbol{w}_n \sim N(\boldsymbol{0}, R_n(I_n))$. The indicator vector I_n is a discrete latent variable which takes an integer value between $1 \sim M$. Usually a number of models treated in the Mixture Kalman filter is about $2 \sim 3$, but we consider a problem of dealing with a large number of models, $M \simeq 100$. Given I_n, F_n, H_n, Q_n, and R_n are known matrices of appropriate dimension. The CDLM is a direct generalization of the dynamic linear model (DLM) and retain a capability of dealing with outliers, sudden jumps, clutters, and other nonlinear features. The CDLM includes other types of generalization of DLM, e.g., Partial non-Gaussian state space model [7], Markov switching state space model, [8] and Dynamic liner models with switching.

2.2. Monte Carlo Mixture Kalman Filter (MCMKF)

We introduce a new filtering scheme and call it as Monte Carlo mixture Kalman filter (MCMKF) that allows us to choose the optimal model from many candidates or to average over many models [9].

The MCMKF algorithm requires a stochastic model which describes a time-dependent structure for I_n. In this study, I_n is assumed to follow a stationary Markov process, i.e.,

$$p(I_n|\boldsymbol{I}_{1:n-1}) = p(I_n|I_{n-1}) \tag{5}$$

where $\boldsymbol{I}_{i:j} = (I_i, I_{i+1}, \cdots, I_j)$ and $p()$ denotes probability density function. An evolution of I_n is realized by Markov switching model with transition probability given by

$$\pi_{ij} = \Pr(I_n = j|I_{n-1} = i) \tag{6}$$

where $\Pr$ denotes realization probability. In the following, we present an algorithm that determines time evolution of I_n.

The MCMKF algorithm consists of two steps. First, temporal variation of the probability distribution of indicator variable I_n is determined. Second, temporal variation of the probability distribution of the state vector $\boldsymbol{x}_n$ is estimated following the history of I_n.

Let $\boldsymbol{y}_{i:j}$ and $\boldsymbol{I}_{i:j}$ be a set of data vectors and indicator variable from time t_i to time t_j, respectively, i.e., $\boldsymbol{y}_{i:j} = (\boldsymbol{y}_i, \boldsymbol{y}_{i+1}, \cdots, \boldsymbol{y}_j)$ and $\boldsymbol{I}_{i:j} = (I_i, I_{i+1}, \cdots, I_j)$. In MCMKF, two conditional joint distributions of $\boldsymbol{I}_{1:n}$: (i) predictive distribution $p(\boldsymbol{I}_{1:n}|\boldsymbol{y}_{1:n-1})$ and (ii) filter distribution $p(\boldsymbol{I}_{1:n}|\boldsymbol{y}_{1:n})$, are approximated by many "particles" that can be considered as independent realizations from each distribution. Let $\boldsymbol{I}^{(j)}_{1:i|k} = (I^{(j)}_{1|k}, I^{(j)}_{2|k}, \cdots I^{(j)}_{i|k})$ be the jth realization of the conditional distribution $p(\boldsymbol{I}_{1:i}|\boldsymbol{y}_{1:k})$. Each distribution is approximated by N_p ($N_p \gg 1$) realizations as follows:

$$\left\{\boldsymbol{I}^{(m)}_{1:n|n-1}\right\}_{m=1}^{N_p} \sim p(\boldsymbol{I}_{1:n}|\boldsymbol{y}_{1:n-1}) \tag{7}$$

$$\left\{\boldsymbol{I}^{(m)}_{1:n|n}\right\}_{m=1}^{N_p} \sim p(\boldsymbol{I}_{1:n}|\boldsymbol{y}_{1:n}) \tag{8}$$

where

$$\Pr\left(\boldsymbol{I}_{1:n} = \boldsymbol{I}^{(j)}_{1:n|n-1}|\boldsymbol{y}_{1:n-1}\right) = \frac{1}{N_p}, \quad (9)$$

$$\Pr\left(\boldsymbol{I}_{1:n} = \boldsymbol{I}^{(j)}_{1:n|n}|\boldsymbol{y}_{1:n}\right) = \frac{1}{N_p}. \quad (10)$$

In this study, we refer to $\{\boldsymbol{I}^{(m)}_{1:n|n-1}\}^{N_p}_{m=1}$ and $\{\boldsymbol{I}^{(m)}_{1:n|n}\}^{N_p}_{m=1}$ as "approximated predictive distribution" and "approximated filter distribution", respectively.

3. RECURSIVE CALCULATION

3.1. Indicator variable estimation

In this subsection, we show that an approximated predictive distribution at time n is obtained from an approximated filter distribution at time $n-1$. We assume that $\{\boldsymbol{I}^{(m)}_{1:n-1|n-1}\}^{N_p}_{m=1}$ and $\boldsymbol{y}_{1:n-1}$ are given. Then the probability $\Pr(\boldsymbol{I}_{1:n} = \boldsymbol{I}^{(j)}_{1:n|n-1}|\ \boldsymbol{y}_{1:n-1})$ is given as

$$\begin{aligned}\Pr(\boldsymbol{I}_{1:n} &= \boldsymbol{I}^{(j)}_{1:n|n-1}|\boldsymbol{y}_{1:n-1}) \\ &= \Pr(I_n = I^{(j)}_{n|n-1}|I_{n-1} = I^{(j)}_{n-1|n-1})\frac{1}{N_p}. \quad (11)\end{aligned}$$

(11) indicates that $\{\boldsymbol{I}^{(m)}_{1:n|n-1}\}^{N_p}_{m=1}$ is obtained by sampling a realization $I^{(j)}_{n|n-1}$ with probability or weight $\Pr(I_n = I^{(j)}_{n|n-1}|I_{n-1} = I^{(j)}_{n-1|n-1})$, and setting $\boldsymbol{I}^{(j)}_{1:n|n-1} = (\boldsymbol{I}^{(j)}_{1:n-1|n-1}, I^{(j)}_{n|n-1})$. Note that $\Pr(I_n = I^{(j)}_{n|n-1}|I_{n-1} = I^{(j)}_{n-1|n-1})$ is given by the Markovian transition probability defined by (6).

Next we show that an approximated filter distribution at time n is obtained from an approximated predictive distribution at time n Given the observation $\boldsymbol{y}_n$, the probability $\Pr(\boldsymbol{I}_{1:n} = \boldsymbol{I}^{(j)}_{1:n|n-1}|\ \boldsymbol{y}_{1:n-1})$ is updated as follows:

$$\Pr(\boldsymbol{I}_{1:n} = \boldsymbol{I}^{(j)}_{1:n|n-1}|\boldsymbol{y}_{1:n}) = \frac{w^{(j)}_n}{\sum^{N_p}_{j=1} w^{(j)}_n} \quad (12)$$

where

$$w^{(j)}_n = p(\boldsymbol{y}_n|\boldsymbol{I}_{1:n} = \boldsymbol{I}^{(j)}_{1:n|n-1}, \boldsymbol{y}_{1:n-1}). \quad (13)$$

Equation (12) means that the filter distribution $p(\boldsymbol{I}_{1:n}|\ \boldsymbol{y}_{1:n})$ is approximated by giving weight proportional to $w^{(j)}_n$ to the jth particle of approximated predictive distribution. For the next prediction step, it is necessary to represent the approximated filter distribution with equally weighted particles $\{\boldsymbol{I}^{(m)}_{1:n|n}\}^{N_p}_{m=1}$. This is achieved by generating N_p particles $\{\boldsymbol{I}^{(m)}_{1:n|n}\}^{N_p}_{m=1}$ by resampling $\{\boldsymbol{I}^{(m)}_{1:n|n-1}\}^{N_p}_{m=1}$ with probability proportional to $\{w^{(m)}_n\}^{N_p}_{m=1}$.

3.2. Kalman filter

Since $\boldsymbol{I}^{(j)}_{1:n-1|n-1}$ is assumed to be given, the CDLM (3) and (4) reduces to a linear Gaussian state space model and thus $\boldsymbol{x}^{(j)}_{n-1|n-1}$ and $V^{(j)}_{n-1|n-1}$ are calculated by Kalman filter. $\boldsymbol{x}^{(j)}_{n|n-1}$ and $V^{(j)}_{n|n-1}$ are also calculated by Kalman filter using $\boldsymbol{x}^{(j)}_{n-1|n-1}$, $V^{(j)}_{n-1|n-1}$ and $I^{(j)}_{n|n-1}$. Then we have

$$\begin{aligned}&p(\boldsymbol{x}_{n-1}|\boldsymbol{I}_{1:n-1} = \boldsymbol{I}^{(j)}_{1:n-1|n-1}, \boldsymbol{y}_{1:n-1}) \\ &\sim N(\boldsymbol{x}^{(j)}_{n-1|n-1}, V^{(j)}_{n-1|n-1}) \quad (14) \\ &p(\boldsymbol{x}_n|\boldsymbol{I}_{1:n} = \boldsymbol{I}^{(j)}_{1:n|n-1}, \boldsymbol{y}_{1:n-1}) \\ &\sim N(\boldsymbol{x}^{(j)}_{n|n-1}, V^{(j)}_{n|n-1}). \quad (15)\end{aligned}$$

Similarly the predictive distribution of data also becomes a Gaussian:

$$\begin{aligned}p(\boldsymbol{y}_n|\boldsymbol{I}_{1:n} &= \boldsymbol{I}^{(j)}_{1:n|n-1}, \boldsymbol{y}_{1:n-1}) \\ &\sim N(\boldsymbol{y}^{(j)}_{n|n-1}, W^{(j)}_{n|n-1}) \quad (16)\end{aligned}$$

where

$$\boldsymbol{y}^{(j)}_{n|n-1} = H_n(I^{(j)}_{n|n-1})\boldsymbol{x}^{(j)}_{n|n-1} \quad (17)$$

$$\begin{aligned}W^{(j)}_{n|n-1} &= H_n(I^{(j)}_{n|n-1})V^{(j)}_{n|n-1}H^T_n(I^{(j)}_{n|n-1}) \\ &\quad + R_n(I^{(j)}_{n|n-1}). \quad (18)\end{aligned}$$

The left hand side of (16) is the weight $w^{(j)}_n$ defined in (13). Thus $w^{(j)}_n$ follows the Gaussian distribution with mean $\boldsymbol{y}^{(j)}_{n|n-1}$ and covariance matrix $W^{(j)}_{n|n-1}$.

3.3. Fixed lag smoother

By using the prediction and the filtering algorithm recursively, we finally obtain N_p particles $\{\boldsymbol{I}^{(m)}_{1:N_e|N_e}\}^{N_p}_{m=1}$ that approximate $p(\boldsymbol{I}_{1:N_e}|\boldsymbol{y}_{1:N_e})$, the posterior distribution of $\boldsymbol{I}_{1:N_e}$ conditioned on all of available data. Here, N_e is the number of observations. $p(\boldsymbol{I}_{1:N_e}|\boldsymbol{y}_{1:N_e})$ is called smoother distribution of $\boldsymbol{I}_{1:N_e}$. A sequence of each particle, $\boldsymbol{I}^{(j)}_{1:N_e|N_e} = [I^{(j)}_{1|N_e}, I^{(j)}_{2|N_e}, \cdots I^{(j)}_{N_e|N_e}]$, is called the trajectory.

In the particle filter, the repetition of resampling gradually decreases the number of different realizations as time passes. Therefore the shape of the distribution of the sate deteriorates as a time passes. Kitagawa [10] showed that this difficulty can be eliminated by employing fixed L-lag smoother rather than fixed interval smoother. Bergman *et al.* [7] presented the other way to avoid this difficulty.

Thus following Kitagawa (1996), we modify the MCMKF filtering algorithm as follows. For fixed L, generate N_p particles $\{\boldsymbol{I}^{(m)}_{n-L:n|n}\}^{N_p}_{m=1}$ by the resampling of $\{\boldsymbol{I}^{(m)}_{n-L:n|n-1}\}^{N_p}_{m=1}$ with probability proportional to $\{w^{(m)}_n\}^{N_p}_{m=1}$ defined in (13).

It is recommended to take L not so large (say, 10 or 20 at the largest 50) [10, 11]. We adopt $L = 20$ in our application. This filtering algorithm is conceptually similar to the storing state vector algorithm in the Monte Carlo filter proposed by Kitagawa [10]. He applied the Monte Carlo approximation directly to the distribution of the state, whereas we apply the approximation to the distribution of the indicator variable.

3.4. State Vector Estimation

We present here an algorithm to estimate the state using all the N_p trajectories $\{\boldsymbol{I}^{(m)}_{1:N_e|N_e}\}^{N_p}_{m=1}$. In this case, $F_n(I^{(j)}_n)$, $Q_n(I^{(j)}_n)$, $H_n(I^{(j)}_n)$ and $R_n(I^{(j)}_n)$ ($j = 1, \cdots, N_p$) in (3) and (4) reduce to sets of known matrices which have different time evolutions corresponding to trajectories. Thus the CDLM defined by (3) and (4) reduces to the conventional linear Gaussian state space model to which Kalman filter is applicable for state estimation. $\{\boldsymbol{x}^{(j)}_{n+1|n}, V^{(j)}_{n+1|n}\}^{N_p}_{j=1}$, $\{\boldsymbol{x}^{(j)}_{n|n}, V^{(j)}_{n|n}\}^{N_p}_{j=1}$ and $\{\boldsymbol{x}^{(j)}_{n|N_e}, V^{(j)}_{n|N_e}\}^{N_p}_{j=1}$ are recursively obtained by Kalman filter. Given $\{\boldsymbol{x}^{(j)}_{n|N_e}, V^{(j)}_{n|N_e}\}^{N_p}_{j=1}$, distribution of the final estimate for $\boldsymbol{x}_n$, $p(\boldsymbol{x}_n|\boldsymbol{y}_{1:N_e})$, is written as

$$p(\boldsymbol{x}_n|\boldsymbol{y}_{1:N_e}) = \frac{1}{N_p}\sum_{j=1}^{N_p} N(\boldsymbol{x}^{(j)}_{n|N_e}, V^{(j)}_{n|N_e}). \qquad (19)$$

4. MCMKF IN SPACE PHYSICS

Many phenomena in space physics tend to be discussed in terms of a complex system in which a nonlinear non-Gaussian fluctuations (disturbances) play an important role. The nonlinear non-Gaussian data assimilation method needs to be developed in an attempt to realize a quantitative prediction of the space environment, i.e., space weather forecasting. In order to cope with this request, we propose the CDLM and apply the MCMKF for its state estimation. Here we give a brief summary of how the MCMKF is applied to problems in space physics.

We prepare several simulation models (simulation schemes) for understanding the specific scientific problem. Each simulation model at time n is specified by an indicator variable I_n. $Q(I_n)$ can be possibly obtained by EnKF. Then we can formulate a system model for each simulation model. The observation model is naturally introduced when the state and observation vectors, $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$, are defined. The problem left to us is to give $Q(I_n)$. One of the easiest way to define $Q(I_n)$ is that we utilize information on statistical fluctuations in each component of the observation vector. We denote each component of $\boldsymbol{y}_n$ by $y_n(j)$ ($j = 1, \ldots, n_y$). To obtain R begins with applying a lowpass filter to each observed time series $y_n(j)$. The smoothed time series is specified by $\widetilde{y}_n(j)$. Then a residual series for each component is determined by $e_n(j) = y_n(j) - \widetilde{y}_n(j)$. Then $Q(I_n)$ is defined by a sample variance-covariance matrix of $e_n(j)$: $Q_{jj'}(I_n) =< e_n(j) \cdot e_n(j') >$, where $Q_{jj'}$ denotes the jj' element of Q.

In conclusion, we address that the MCMKF is designed to deal with the CDLM and then can be applicable to a wide variety of the nonlinear non-Gaussian state space models. The MCMKF allows us to integrate various type of time series models and to generate a flexible time series model automatically.

5. REFERENCES

[1] I. Fukumori, "Data assimilation by models," in *Satellite altimetry and earth sciences: A Handbook of Techniques and Applications*, L.L. Fu and A. Cazenave, Eds., pp. 237–265. Academic Press, 2001.

[2] B.D.O. Anderson and J.B. Moore, *Optimal Filtering*, Prentice-Hall, New Jersey, 1979.

[3] G. Kitagawa and W. Gersch, *Smoothness Priors Analysis of Time Series*, Springer-Verlag, New York, 1996.

[4] P. Segall and M. Matthews, "Time dependent inversion of geodetic data," *Journal of Geophysical Research*, vol. 102, pp. 22391–22409, 1997.

[5] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *Journal of Geophysical Research*, vol. 99, pp. 10143–10162, 1994.

[6] R. Chen and J.S. Liu, "Mixture Kalman filters," *J. R. Statist. Soc. B.*, vol. 62, pp. 493–508, 2000.

[7] N. Bergman, A. Doucet, and N. Gordon, "Optimal estimation and Cramér-Rao bounds for partial non-Gaussian state space models," *Annals of Institute of Statistical Mathematics*, vol. 53, pp. 97–122, 2001.

[8] C.-J. Kim and Ch.R. Nelson, *State space models with regime switching*, MIT press, Cambridge, 1999.

[9] T. Higuchi and J. Fukuda, "Monte Carlo mixture Kalman filter and its application to space-time inversion," in *13th IFAC Symposium on System Identification*, 2003, pp. 1299–1304.

[10] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space model," *Journal of Computational and Graphical Statistics*, vol. 5, pp. 1–25, 1996.

[11] T. Higuchi and G. Kitagawa, "Knowledge discovery and self-organizing state space model," *IEICE Transactions on Information and Systems*, vol. E83-D, pp. 36–43, 2000.