Kalman Filtering: Part I

Instructor: Istvan Szunyogh

Class: February 23, 2007

Recommended Readings:

- Geir Evensen, 2006: *Data Assimilation: The Ensemble Kalman Filter,* Springer, 280 pages, is a nice handbook that also provides a good summary of the history. Cautionary notes
  - There have been many important developments since book has been completed (most likely in early 2005)
  - When considering the computational cost of the alternative computational algorithms, the book does not really consider that the algorithms are usually implemented on parallel computers (this is the case for an operational NWP model)
  - A little too much credit is claimed by the author–this limits the value of the book only as a source on history
- Brian Hunt, Eric Kostelich and Istvan Szunyogh, 2007: Efficient Data Assimilation for Spatiotemporal Chaos: a Local Ensemble Kalman Filter. *Physica D*. Available from the Weather-Chaos web page.

Mathematical Formulation, following Brian Hunt (the most elegant formulation I am aware of)

The Analysis Problem:

- Consider a system governed by the ordinary differential equation
$$\frac{d\mathbf{x}}{dt} = F(t, \mathbf{x}), \tag{1}$$
where $\mathbf{x}$ is an m-dimensional vector representing the state of the system at a given time.

- Suppose we are given a set of (noisy) observations of the system made at various times.

- We want to determine which trajectory $\{\mathbf{x}(t)\}$ of (1) "best" fits the observations. For any given t, this trajectory gives an estimate of the system state at time t.

Notation

- Let us assume that the observations are the result of measuring quantities that depend on the system state in a known way, with Gaussian measurement errors.

- An observation at time t_j is a triple $(\mathbf{y}_j^o, H_j, \mathbf{R}_j)$, where $\mathbf{y}_j^o$ is a vector of observed values, and H_j and $\mathbf{R}_j$ describe the relationship between $\mathbf{y}_j^o$ and $\mathbf{x}(t_j)$:

$$\mathbf{y}_j^o = H_j(\mathbf{x}(t_j)) + \varepsilon_j,$$

  where ε_j is a Gaussian random variable with mean $\mathbf{0}$ and covariance matrix $\mathbf{R}_j$.

- Here, a perfect model is assumed: the observations are based on a trajectory of (1), and our problem is simply to infer which trajectory produced the observations. In a real application, the observations come from a trajectory of the physical system for which (1) is only a model.

The maximum likelihood estimate for the trajectory that best fits the observations at times $t_1 < t_2 < \cdots < t_n$.

- The likelihood of a trajectory $\mathbf{x}(t)$ is proportional to

$$\prod_{j=1}^{n} \text{exp}(-[\mathbf{y}_j^o - H_j(\mathbf{x}(t_j))]^T \mathbf{R}_j^{-1} [\mathbf{y}_j^o - H_j(\mathbf{x}(t_j))]),$$

  since the observational errors are normally distributed and are assumed to be independent at the different observation times. The most likely trajectory is the one that maximizes this expression.

- Equivalently, the most likely trajectory is the one that minimizes the "cost function"

$$J^o(\{\mathbf{x}(t)\}) = \sum_{j=1}^{n} [\mathbf{y}_j^o - H_j(\mathbf{x}(t_j))]^T \mathbf{R}_j^{-1} [\mathbf{y}_j^o - H_j(\mathbf{x}(t_j))]. \quad (2)$$

  Thus, the "most likely" trajectory is also the one that best fits the observations in a least square sense.

Replacing the Trajectory with the State at a Particular Time

- (2) expresses the cost J^o as a function of the trajectory $\{\mathbf{x}(t)\}$. To minimize the cost, it is more convenient to write it as a function of the system state at a particular time t.

- Let $M_{t,t'}$ be the map that propagates a solution of (1) from time t to time t'. Then

$$J_t^o(\mathbf{x}) = \sum_{j=1}^{n} [\mathbf{y}_j^o - H_j(M_{t,t_j}(\mathbf{x}))]^T \mathbf{R}_j^{-1} [\mathbf{y}_j^o - H_j(M_{t,t_j}(\mathbf{x}))] \quad (3)$$

  expresses the cost in terms of the system state $\mathbf{x}$ at time t.

- To estimate the state at time t, we attempt to minimize J_t^o.

Remarks

- In practice the observations do not have to be all collected at t_n. In a typical implementation, at t_n we assimilate all observations that were collected at times t in the window $t_n - \Delta t/2 < t < t_n + \Delta t/2$, where $\Delta t = t_j - t_{j-1}$, $j = 2, \ldots, n$.

- For a nonlinear model, there is no guarantee that a unique minimum exists.

- Even if a minimum exist, evaluating J_t^o is apt to be computationally expensive, and minimizing it may be impractical.

- But, if both the model and the observation operators H_j are linear, the minimization is quite tractable, because J_t^o is then quadratic. Furthermore, one can compute the minimum by an iterative method, namely the Kalman Filter (Kalman 1960; Kalman and Bucy 1961).

Linear Scenario: the Kalman Filter

- In the linear scenario, we can write $M_{t,t'}(\mathbf{x}) = \mathbf{M}_{t,t'}\mathbf{x}$ and $H_j(\mathbf{x}) = \mathbf{H}_j\mathbf{x}$ where $\mathbf{M}_{t,t'}$ and $\mathbf{H}_j$ are matrices.

- We now describe how to perform
  - a forecast step from time t_{n-1} to time t_n
  - followed by an analysis step at time t_n,
  - in such a way that if we start with the most likely system state, given the observations up to time t_{n-1}, we end up with the most likely state given the observations up to time t_n.

The estimate of the state and the uncertainty at t_{n-1}

- Suppose the analysis at time t_{n-1} has produced a state estimate $\bar{\mathbf{x}}^a_{n-1}$ and an associated covariance matrix $\mathbf{P}^a_{n-1}$. In probabilistic terms, $\bar{\mathbf{x}}^a_{n-1}$ and $\mathbf{P}^a_{n-1}$ represent the mean and covariance of a Gaussian probability distribution that represents the relative likelihood of the possible system states given the observations from time t_1 to t_{n-1}.

- Algebraically, what we assume is that for some constant c,

$$\sum_{j=1}^{n-1}[\mathbf{y}^o_j - \mathbf{H}_j\mathbf{M}_{t_{n-1},t_j}\mathbf{x}]^T\mathbf{R}_j^{-1}[\mathbf{y}^o_j - \mathbf{H}_j\mathbf{M}_{t_{n-1},t_j}\mathbf{x}] = \tag{4}$$

$$= [\mathbf{x} - \bar{\mathbf{x}}^a_{n-1}]^T(\mathbf{P}^a_{n-1})^{-1}[\mathbf{x} - \bar{\mathbf{x}}^a_{n-1}] + c.$$

  In other words, the analysis at time t_{n-1} has "completed the square" to express the part of the quadratic cost function $J^o_{t_{n-1}}$ that depends on the observations up to that time as a single quadratic form plus a constant.

- The Kalman Filter determines $\bar{\mathbf{x}}^a_n$ and $\mathbf{P}^a_n$ such that an analogous equation holds at time t_n.

The Kalman Filter I

- We propagate the analysis state estimate $\bar{\mathbf{x}}^a_{n-1}$ and its covariance matrix $\mathbf{P}^a_{n-1}$ using the forecast model to produce a background state estimate $\bar{\mathbf{x}}^b_n$ and covariance $\mathbf{P}^b_n$ for the next analysis:

$$\bar{\mathbf{x}}^b_n = \mathbf{M}_{t_{n-1},t_n}\bar{\mathbf{x}}^a_{n-1}, \tag{5}$$

$$\mathbf{P}^b_n = \mathbf{M}_{t_{n-1},t_n}\mathbf{P}^a_{n-1}\mathbf{M}^T_{t_{n-1},t_n}. \tag{6}$$

- Under a linear model, a Gaussian distribution of states at one time propagates to a Gaussian distribution at any other time, and the equations above describe how the model propagates the mean and covariance of such a distribution.

The Kalman Filter II

- Next, we want to rewrite the cost function $J^o_{t_n}$ given by (3) in terms of the background state estimate and the observations at time t_n. (This step is often formulated as applying Bayes' Rule to the corresponding probability density functions.) In (4), $\mathbf{x}$ represents a system state at time t_{n-1}. In our expression for $J^o_{t_n}$, we want $\mathbf{x}$ to represent a system state at time t_n

- Using (5) and (6) yields that part of the cost function at t_n that reflects the effect of observations collected up to t_n

$$\sum_{j=1}^{n-1}[\mathbf{y}^o_j-\mathbf{H}_j\mathbf{M}_{t_n,t_j}\mathbf{x}]^T\mathbf{R}_j^{-1}[\mathbf{y}^o_j-\mathbf{H}_j\mathbf{M}_{t_n,t_j}\mathbf{x}] = [\mathbf{x}-\bar{\mathbf{x}}^b_n]^T(\mathbf{P}^b_n)^{-1}[\mathbf{x}-\mathbf{x}^b_n]+c.$$

- It follows that the total cost function at t_n is

$$J^o_{t_n}(\mathbf{x}) = [\mathbf{x}-\bar{\mathbf{x}}^b_n]^T(\mathbf{P}^b_n)^{-1}[\mathbf{x}-\bar{\mathbf{x}}^b_n]+[\mathbf{y}^o_n-\mathbf{H}_n\mathbf{x}]^T\mathbf{R}_n^{-1}[\mathbf{y}^o_n-\mathbf{H}_n\mathbf{x}]+c. \quad (7)$$

  where the second term reflects the effects of observations collected at t_n

The Kalman Filter III

- To complete the data assimilation cycle, we determine the state estimate $\bar{\mathbf{x}}_n^a$ and its covariance $\mathbf{P}_n^a$ so that

$$J_{t_n}^o(\mathbf{x}) = [\mathbf{x} - \bar{\mathbf{x}}_n^a]^T (\mathbf{P}_n^a)^{-1} [\mathbf{x} - \bar{\mathbf{x}}_n^a] + c'$$

  for some constant c'.

- Equating the terms of degree 2 in $\mathbf{x}$, we get

$$\mathbf{P}_n^a = \left[(\mathbf{P}_n^b)^{-1} + \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{H}_n \right]^{-1}. \tag{8}$$

- Equating the terms of degree 1, we get

$$\bar{\mathbf{x}}_n^a = \mathbf{P}_n^a \left[(\mathbf{P}_n^b)^{-1} \bar{\mathbf{x}}_n^b + \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{y}_n^o \right]. \tag{9}$$

- The last equation in some sense (consider, for example, the case where $\mathbf{H}_n$ is the identity matrix) expresses the analysis state estimate as a weighted average of the background state estimate and the observations, weighted according to the inverse covariance of each.

The Kalman Filter IV

Equations (8) and (9) can be written in many different but equivalent forms

- Using (8) to eliminate $(\mathbf{P}_n^b)^{-1}$ from (9) yields

$$\bar{\mathbf{x}}_n^a = \bar{\mathbf{x}}_n^b + \mathbf{P}_n^a \mathbf{H}_n^T \mathbf{R}_n^{-1} (\mathbf{y}_n^o - \mathbf{H}_n \bar{\mathbf{x}}_n^b) = \bar{\mathbf{x}}_n^b + \mathbf{K}(\mathbf{y}_n^o - \mathbf{H}_n \bar{\mathbf{x}}_n^b) \quad (10)$$

- The matrix $\mathbf{K} = \mathbf{P}_n^a \mathbf{H}_n^T \mathbf{R}_n^{-1}$ is called the Kalman gain. It multiplies the difference between the observations at time t_n and the values predicted by the background state estimate to yield the increment between the background and analysis state estimates.

- Rearranging (8) yields

$$\mathbf{P}_n^a = (\mathbf{I} + \mathbf{P}_n^b \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{H}_n)^{-1} \mathbf{P}_n^b = (\mathbf{I} - \mathbf{K}\mathbf{H}) \mathbf{P}_n^b. \quad (11)$$

  This expression is better than the previous one from a practical point of view, since it does not require inverting $\mathbf{P}_n^b$.

The Nonlinear Scenario: The Extended Kalman Filter

- Many approaches to data assimilation for nonlinear problems are based on the Kalman Filter, or at least on minimizing a cost function similar to (7).

- At a minimum, a nonlinear model forces a change in the forecast equations (5) and (6), while nonlinear observation operators H_n force a change in the analysis equations (10) and (11)

- The Extended Kalman Filter (see, for example, Jazwinski 1970) computes $\bar{\mathbf{x}}_n^b = M_{t_{n-1},t_n}(\bar{\mathbf{x}}_{n-1}^a)$ using the nonlinear model, but computes $\mathbf{P}_n^b$ using the linearization $\mathbf{M}_{t_{n-1},t_n}$ of M_{t_{n-1},t_n} around $\bar{\mathbf{x}}_{n-1}^a$. The analysis then uses the linearization $\mathbf{H}_n$ of H_n around $\bar{\mathbf{x}}_n^b$.

Difficulties with the Implementation of the Extended Kalman Filter

- It is not easy to linearize the dynamics for a complex, high-dimensional model, such as a global weather prediction model.

- The number of model variables m is several million, and as a result the $m \times m$ matrix inverse required by the analysis cannot be performed in a reasonable amount of time.

- The use of the linear evolution equations can lead to an unbounded linear instability (see chapter 4.2.3 in Evensen 2006).

Practical Implementations at the NWP Centers

- Approaches used in operational weather forecasting generally eliminate for pragmatic reasons the time iteration of the Kalman Filter.

- NCEP/NWS: data assimilation is done every 6 hours with a 3D-VAR method, in which the background covariance $\mathbf{P}_n^b$ is replaced by a constant matrix $\mathbf{B}$ The 3D-VAR cost function also includes a nonlinear observation operator H_n, and is minimized numerically to produce the analysis state estimate $\mathbf{x}_n^a$.

- The 4D-VAR method (e.g., Le Dimet and Talagrand 1986; Talagrand and Courtier 1987) used by the European Centre for Medium-Range Weather Forecasts uses a cost function that includes a constant-covariance background term as in 3D-VAR together with a sum like (2) accounting for the observations collected over a 12 hour time span.