(Extended) Kalman Filter

Brian Hunt

7 June 2013

Goals of Data Assimilation (DA)

- Estimate the state of a system based on both current and all past observations of the system, using a model for the system dynamics.
- Perform the estimation iteratively: compute the current estimate in terms of a recent past estimate.
- Ideally, quantify the uncertainty in the state estimate.

Terminology and Notation

- Forecast model: a known function M on a vector space of model states.
- Truth: an unknown sequence $\{x_n\}$ of model states to be estimated.
- Model error: $\delta_n = x_{n+1} - M(x_n)$.
- Observations: a sequence $\{y_n\}$ of vectors in observation space (may depend on n).
- Forward operator: a known function H_n from model space to observation space.
- Observation error: $\varepsilon_n = y_n - H_n(x_n)$.

When DA is not Necessary

- If the forward operator H_n is invertible and $\varepsilon_n = 0$ then $x_n = H_n^{-1}(y_n)$.
- If H_n is invertible and the statistics of ε_n are known, then we can compute the pdf of x_n (but data assimilation can improve the estimate).
- Note: the pdf (probability density function) of x gives the relative likelihood of the possible values of x. The maximizer ("mode") of the pdf is the most likely value.

More Terminology

- Background ("first guess"): estimate x_n^b of the current model state x_n given past observations $y_1, \ldots, y_{n-1}$.
- Analysis: estimate x_n^a of x_n given current and past observations $y_1, \ldots, y_n$.
- A data assimilation cycle consists of:
- Analysis step: Determine analysis x_n^a from background x_n^b and observations y_n.
- Forecast step: Typically $x_{n+1}^b = M(x_n^a)$.

Remarks on the Analysis Step

- If the observation error ε_n is zero, we should seek x_n^a close to x_n^b such that $H_n(x_n^a) = y_n$.
- Otherwise, we should just make $H_n(x_n^a)$ closer to y_n than $H_n(x_n^b)$ is.
- How much closer depends on the relative uncertainties of the background estimate x_b^n and the observation y_n.
- The better we understand the uncertainties, the clearer it is how to do the analysis step.

Bayes' Rule

- Definition of conditional probability:

$$P(V|W) = P(V \cap W)/P(W)$$
$$P(V \text{ given } W) = P(V \text{ and } W)/P(W)$$

- Then

$$P(V|W)P(W) = P(V \cap W) = P(V)P(W|V).$$

- Corollary:

$$P(V|W) = P(V)P(W|V)/P(W)$$
$$\text{posterior} = \text{prior} \cdot \text{likelihood}/\text{normalization}$$

Bayesian Data Assimilation

- Assume that the statistics of the model error δ_n and observation error ε_n are known.
- Theoretically, given an analysis pdf $p(x_{n-1}|y_1, \ldots, y_{n-1})$, we can use the forecast model to determine a background ("prior") pdf $p(x_n|y_1, \ldots, y_{n-1})$.
- The forward operator tells us $p(y_n|x_n)$.
- Bayes' rule tells us that the analysis ("posterior") pdf $p(x_n|y_1, \ldots, y_n)$ is proportional to $p(x_n|y_1, \ldots, y_{n-1})p(y_n|x_n)$.

Advantages and Disadvantages

- Advantage: the analysis step is simple – just multiply two functions.
- Disadvantage: the forecast step is generally unfeasible in practice.
- If x is high-dimensional, we can't numerically keep track of an arbitrary pdf for x – too much information!
- We need to make some simplifying assumptions.

Linearity and Gaussianity

- Assume that M and H_n are linear.
- Assume model and observation errors are Gaussian with known covariances and no time correlations: $\delta_n \sim N(0, Q_n)$ and $\varepsilon_n \sim N(0, R_n)$.
- Then in the analysis step, a Gaussian background pdf leads to a Gaussian analysis pdf.
- Gaussian input yields Gaussian output in the forecast step too.
- Let the background pdf have mean x_n^b and covariance P_n^b.

Bayesian DA with Gaussians

- The (unnormalized) background pdf is:

$$\exp[-(x_n - x_n^b)^T (P_n^b)^{-1} (x_n - x_n^b)/2]$$

- The pdf of y_n given x_n is

$$\exp[-(H_n x_n - y_n)^T R_n^{-1} (H_n x_n - y_n)/2]$$

- The analysis pdf is the $\exp(-J_n/2)$ where:

$$\begin{aligned} J_n =& (x_n - x_n^b)^T (P_n^b)^{-1} (x_n - x_n^b) \\ &+ (H_n x_n - y_n)^T R_n^{-1} (H_n x_n - y_n) \end{aligned}$$

- To find the mean and covariance of the analysis pdf, we want to write:

$$J_n = (x_n - x_n^a)^T (P_n^a)^{-1} (x_n - x_n^a) + c$$

The Kalman Filter

[Kalman 1960]

- After some linear algebra, the analysis mean x_n^a and covariance P_n^a are

$$x_n^a = x_n^b + K_n(y_n - H_n x_n^b)$$
$$P_n^a = [(P_n^b)^{-1} + H_n^T R_n^{-1} H_n]^{-1}$$
$$= [I + P_n^b H_n^T R_n^{-1} H_n]^{-1} P_n^b$$

  where $K_n = P_n^a H_n^T R_n^{-1}$ is the Kalman gain matrix.

- The forecast step is $x_{n+1}^b = M x_n^a$ and $P_{n+1}^b = M P_n^b M^T + Q_n$

Observation Space Formulation

- After some further linear algebra, the Kalman filter analysis equations can be written

$$
\begin{aligned}
K_n &= P_n^b H_n^T [H_n P_n^b H_n^T + R_n]^{-1} \\
x_n^a &= x_n^b + K_n(y_n - H_n x_n^b) \\
P_n^a &= (I - K_n H_n) P_n^b
\end{aligned}
$$

- The size of the matrix that must be inverted is determined by the number of (current) observations, not by the number of model state variables.

Example

- Assume that $M = H_n = I$, that x is a scalar, and that $Q_n = 0$ and $R_n = r > 0$.
- We are making independent measurements $y_1, y_2, \ldots$ of a constant-in-time quantity x.
- The analysis equations are:

$$
\begin{aligned}
x_n^a &= x_n^b + P_n^a r^{-1}(y_n - x_n^b) \\
(P_n^a)^{-1} &= (P_n^b)^{-1} + r^{-1}
\end{aligned}
$$

- Start with a uniform "prior" pdf: $(P_1^b)^{-1} = 0$ and x_1^b arbitrary.
- Then by induction, $P_{n+1}^b = P_n^a = r/n$ and $x_{n+1}^b = x_n^a = (y_1 + \cdots + y_n)/n$.

A Least Squares Formulation

- In terms of all the observations $y_1, \ldots, y_n$, what problem did we solve to estimate x_n?
- Assume no model error ($\delta_n = 0$).
- The likelihood of a model trajectory $x_1, \ldots, x_n$ is $\exp(-J_n/2)$ where:

$$J_n = \sum_{i=1}^{n} (H_i x_i - y_i)^T R_i^{-1} (H_i x_i - y_i)$$

- Problem: minimize the cost function $J_n(x_1, \ldots, x_n)$ subject to the constraints $x_{i+1} = M x_i$.

Kalman Filter Revisited

- The Kalman filter expresses the minimizer x_n^a of J_n in terms of the minimizer x_{n-1}^a of J_{n-1} as follows.
- It expresses J_{n-1} as a function of x_{n-1} only.
- It keeps track of an auxiliary matrix P_{n-1}^a that is the 2nd derivative (Hessian) of J_{n-1}.
- Assuming it has done so correctly at time $n-1$, the next slide explains why it does so at time n.

Kalman Filter Revisited

- If x^a_{n-1} minimizes J_{n-1} and P^a_{n-1} is its Hessian, then

$$J_{n-1} = (x_{n-1} - x^a_{n-1})^T (P^a_{n-1})^{-1} (x_{n-1} - x^a_{n-1}) + c_{n-1}$$

- Then substituting $x_n = Mx_{n-1}$, $x^b_n = Mx^a_{n-1}$, and $P^b_n = MP^a_{n-1}M^T$ yields:

$$J_{n-1} = (x_n - x^b_n)^T (P^b_n)^{-1} (x_n - x^b_n) + c_{n-1}$$

- We get the same cost function as before:

$$J_n = J_{n-1} + (Hx_n - y_n)^T R_n^{-1} (Hx_n - y_n)$$

- The KF completes the square as before.

Nonlinear Least Squares

- Now let's eliminate the assumption that M and H_i are linear.
- As before, assume no model error and Gaussian observation errors.
- The maximum likelihood estimate for the true trajectory is the minimizer of:

$$J_n = \sum_{i=1}^{n} (H_i(x_i) - y_i)^T R_i^{-1} (H_i(x_i) - y_i)$$

subject to the constraints $x_{i+1} = M(x_i)$.

Approximate Solution Methods

- Use an approximate solution at time $n - 1$ to find an approximate solution at time n.
- If we track covariances associated with our estimates, we can write:

$$J_{n-1} \approx (x_{n-1} - x_{n-1}^a)^T (P_{n-1}^a)^{-1} (x_{n-1} - x_{n-1}^a) + c$$

- As a further approximation, we can write:

$$J_{n-1} \approx (x_n - x_n^b)^T (P_n^b)^{-1} (x_n - x_n^b) + c$$

- It seems clear that x_n^b should be $M(x_{n-1}^a)$, but what choice of P_n^b is best?

Extended Kalman Filter

- Matching the second derivatives of the two approximate cost functions yields $P^b_n = (DM)P^a_{n-1}(DM)^T$ where DM is the derivative of M at x^a_{n-1}.
- The remaining equations are like the Kalman filter (linearizing H_n near x^b_n).
- Advantage: The approximation error may be smaller than for other methods.
- Disadvantage: For a high-dimensional model, the covariance forecast is computationally expensive.

Extended KF (Square Root Form)

- If M is computed by solving a system of differential equations, then DM is computed by solving the associated tangent linear model (TLM).
- If $P_{n-1}^a = X_{n-1}^a (X_{n-1}^a)^T$, then compute $X_n^b = (DM) X_{n-1}^a$, followed by $P_n^b = X_n^b (X_n^b)^T$.
- This is easier if X_{n-1}^a has (many) fewer columns than rows; the resulting covariance has reduced rank.
- The Kalman covariance update becomes

$$X_n^a = X_n^b (H_n X_n^b)^T [H_n X_n^b (H_n X_n^b)^T + R_n]^{-1/2}.$$

Tangent Linear Model

- Suppose $x_n = x(n)$ where $dx/dt = F(x)$.
- Then for all solutions, $M(x(0)) = x(1)$.
- Consider a family of solutions with $x_\gamma(0) = x_0 + \gamma v$; then $DM(x_0)v = (\partial/\partial\gamma)x_\gamma(1)|_{\gamma=0}$.
- Let $v(t) = (\partial/\partial\gamma)x_\gamma(t)|_{\gamma=0}$.
- Substituting $x_\gamma(t)$ into the ODE and differentiating w.r.t. γ yields

$$dv/dt = DF(x_0(t))v$$

- Compute $v(1)$ with $v(0) = v$ to get $DM(x_0)v$.

Ensemble Kalman Filter

- Use an ensemble of model states whose mean and covariance are transformed according to the Kalman filter equations.
- Forecast each ensemble member separately.
- Advantage: Relatively easy to implement and the analysis step is computationally efficient.
- Disadvantage: Only represents uncertainty in a space whose dimension is bounded by the ensemble size (inherently reduced rank).

3D-Var

- Replace P_n^b with a time-independent background covariance matrix B, determined empirically.
- Numerically minimize the resulting cost function (allowing nonlinear H_n).
- Advantage: The covariance B and associated matrices ($B^{1/2}$ is used in the analysis) only need to be computed once.
- Disadvantage: Ignores time dependence of background uncertainty, which can vary considerably.

(Strong Constraint) 4D-Var

[le Dimet & Talagrand 1985]

- Numerically minimize the cost function

$$J_n = (x_{n-p} - x^b_{n-p})^T B^{-1} (x_{n-p} - x^b_{n-p}) + \sum_{i=n-p}^{n} (H_i(x_i) - y_i)^T R_i^{-1} (H_i(x_i) - y_i)$$

  subject to the constraints $x_{i+1} = M(x_i)$.
- Advantage: Accuracy, especially as p increases.
- Disadvantage: Difficult to implement and computationally expensive.