Jeff Anderson is part of the Data Assimilation Research Section, Located at the National Center for Atmospheric Research in Boulder, Colorado. It’s an awfully nice place to visit, and we love hosting visitors for collaboration! Someday the world will be back to normal.
Removing the Kalman from the Ensemble Kalman Filter

Jeff Anderson, NCAR/DAReS

The National Center for Atmospheric Research is sponsored by the National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

©UCAR 2014
Building a Forecast System

- Prediction Model
- Observing System
- Data Assimilation
- Analysis

Forecasts

Initial Conditions

Observations
A system governed by (stochastic) Difference Equation:

\[ dx_t = f(x_t, t) + G(x_t, t) d\beta_t, \quad t \geq 0 \]  \hspace{1cm} (1)

Observations at discrete times:

\[ y_k = h(x_k, t_k) + \nu_k; \quad k = 1, 2, \ldots; \quad t_{k+1} > t_k \geq t_0 \]  \hspace{1cm} (2)

Observational error white in time and Gaussian (nice, not essential).

\[ \nu_k \rightarrow N(0, R_k) \]  \hspace{1cm} (3)

Complete history of observations is:

\[ Y_\tau = \{y_l; t_l \leq \tau\} \]  \hspace{1cm} (4)

Goal: Find probability distribution for state:

\[ p(x, t \mid Y_t) \quad \text{Analysis} \quad p(x, t^+ \mid Y_t) \quad \text{Forecast} \]  \hspace{1cm} (5)
A General Description of the Forecast Problem

State between observation times obtained from Difference Equation. Need to update state given new observations:

\[ p(x,t_k \mid Y_{t_k}) = p(x,t_k \mid y_k,Y_{t_{k-1}}) \]  \hspace{1cm} (6)

Apply Bayes’ rule:

\[ p(x,t_k \mid Y_{t_k}) = \frac{p(y_k \mid x_k,Y_{t_{k-1}}) p(x,t_k \mid Y_{t_{k-1}})}{p(y_k \mid Y_{t_{k-1}})} \]  \hspace{1cm} (7)

Noise is white in time (3), so:

\[ p(y_k \mid x_k,Y_{t_{k-1}}) = p(y_k \mid x_k) \]  \hspace{1cm} (8)

Integrate numerator to get normalizing denominator:

\[ p(y_k \mid Y_{t_{k-1}}) = \int p(y_k \mid x) p(x,t_k \mid Y_{t_{k-1}}) \, dx \]  \hspace{1cm} (9)
A General Description of the Forecast Problem

Probability after new observation:

\[
p(x, t_k | Y_{t_k}) = \frac{p(y_k | x) p(x, t_k | Y_{t_{k-1}})}{\int p(y_k | \xi) p(\xi, t_k | Y_{t_{k-1}}) d\xi}
\]

(10)

Prior (forecast)

Likelihood

Posterior (analysis).

Denominator just normalization.
Assumes:

- Linear model
- Gaussian noise
- Gaussian state
- Linear forward operator,
- Gaussian observation error

\[ dx_t = f(x_t, t) + G(x_t, t) \, d\beta_t, \quad t \geq 0 \]

\[ y_k = h(x_k, t_k) + v_k; \quad k = 1, 2, \ldots; \quad t_{k+1} > t_k \geq t_0 \]
Product of of Two Gaussians

Product of d-dimensional normals with means $\mu_1$ and $\mu_2$ and covariance matrices $\Sigma_1$ and $\Sigma_2$ is normal.

$$N(\mu_1, \Sigma_1)N(\mu_2, \Sigma_2) = cN(\mu, \Sigma)$$
Product of Two Gaussians

Product of d-dimensional normals with means $\mu_1$ and $\mu_2$ and covariance matrices $\Sigma_1$ and $\Sigma_2$ is normal.

$$N(\mu_1, \Sigma_1)N(\mu_2, \Sigma_2) = cN(\mu, \Sigma)$$

Covariance:

$$\Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}$$

Mean:

$$\mu = \Sigma(\Sigma_1^{-1} \mu_1 + \Sigma_2^{-1} \mu_2)$$
Product of Two Gaussians

Product of $d$-dimensional normals with means $\mu_1$ and $\mu_2$ and covariance matrices $\Sigma_1$ and $\Sigma_2$ is normal.

\[
N(\mu_1, \Sigma_1) N(\mu_2, \Sigma_2) = c N(\mu, \Sigma)
\]

Covariance:

\[
\Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}
\]

Mean:

\[
\mu = \Sigma (\Sigma_1^{-1} \mu_1 + \Sigma_2^{-1} \mu_2)
\]

Weight:

\[
c = \frac{1}{(2 \pi)^{d/2} |\Sigma_1 + \Sigma_2|^{1/2}} \exp \left\{ - \frac{1}{2} \left[ (\mu_2 - \mu_1)^T (\Sigma_1 + \Sigma_2)^{-1} (\mu_2 - \mu_1) \right] \right\}
\]

We’ll ignore the weight since we immediately normalize products to be PDFs.
The Kalman Filter

\[ p(x, t_k \mid Y_{t_k}) = \frac{p(y_k \mid x)p(x, t_k \mid Y_{t_{k-1}})}{\int p(y_k \mid \xi)p(\xi, t_k \mid Y_{t_{k-1}}) \, d\xi} \] (10)

Numerator is just product of two gaussians.

Denominator just normalizes posterior to be a PDF.
\[ p(x, t_k \mid Y_{t_k}) = \frac{p(y_k \mid x) p(x, t_k \mid Y_{t_k-1})}{\int p(y_k \mid \xi) p(\xi, t_k \mid Y_{t_k-1}) d\xi} \]
The Kalman Filter

\[
p(x, t_k | Y_{t_k}) = \frac{p(y_k | x) p(x, t_k | Y_{t_k-1})}{\int p(y_k | \xi) p(\xi, t_k | Y_{t_k-1}) d\xi}
\]
The Kalman Filter

\[ p(x, t_k | Y_{t_k}) = \frac{p(y_k | x) p(x, t_k | Y_{t_k-1})}{\int p(y_k | \xi) p(\xi, t_k | Y_{t_k-1}) d\xi} \]  

(10)
Product of d-dimensional normals with means $\mu_1$ and $\mu_2$ and covariance matrices $\Sigma_1$ and $\Sigma_2$ is normal.

$N(\mu_1, \Sigma_1)N(\mu_2, \Sigma_2) = cN(\mu, \Sigma)$

Covariance:

$$\Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}$$

Mean:

$$\mu = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} (\Sigma_1^{-1} \mu_1 + \Sigma_2^{-1} \mu_2)$$

Must store and invert covariance matrices.  
**Too big** to store for large problems.  
**Too costly** to invert, $> O(n^2)$.  

Kalman Filter: Cost Challenges
1. Start with ensemble of forecasts.
2. Fit a normal to ensemble.
3. Do standard Kalman filter.
The Ensemble Kalman Filter

Have continuous posterior; need an ensemble.
4. Can create an ensemble with exact sample mean and covariance of continuous posterior.
1. No need for linear model to advance covariance estimate.
Without loss of generality (for Kalman filter)…

Can assimilate observations serially, one at a time.
Fit a Gaussian to the sample.
Get the observation likelihood.
Compute the continuous posterior PDF.
Use a deterministic algorithm to ‘adjust’ the ensemble.
First, ‘shift’ the ensemble to have the exact mean of the posterior.
First, ‘shift’ the ensemble to have the exact mean of the posterior. Second, linearly contract to have the exact variance of the posterior. Sample statistics are identical to Kalman filter.
Without loss of generality (for Kalman filter)…

Can compute impact of observation on each state variable independently.
Assume that all we know is the prior joint distribution.

One variable is observed.

What should happen to the unobserved variable?
Ensemble filters: Updating additional prior state variables

Assume that all we know is the prior joint distribution.

How should the unobserved variable be impacted?

Least squares.

Equivalent to linear regression.

Same as assuming bi-Gaussian prior.
Ensemble filters: Updating additional prior state variables

Have joint prior distribution of two variables.

How should the unobserved variable be impacted?

1\textsuperscript{st} choice: least squares

Begin by finding least squares fit.
Ensemble filters: Updating additional prior state variables

Have joint prior distribution of two variables.

Next, regress the observed variable increments onto increments for the unobserved variable.

Equivalent to first finding image of increment in joint space.
Ensemble filters: Updating additional prior state variables

Have joint prior distribution of two variables.

Next, regress the observed variable increments onto increments for the unobserved variable.

Equivalent to first finding image of increment in joint space.
Ensemble filters: Updating additional prior state variables

Have joint prior distribution of two variables.

Regression: Equivalent to first finding image of increment in joint space.

Then projecting from joint space onto unobserved priors.
Ensemble filters: Updating additional prior state variables

Have joint prior distribution of two variables.

Regression: Equivalent to first finding image of increment in joint space.

Then projecting from joint space onto unobserved priors.
Ensemble filters: Updating additional prior state variables

Now have an updated (posterior) ensemble for the unobserved variable.

Unobserved State Variable

We’ve expanded this plot. Same information as previous slides.

Compressed these two.

Jeff Anderson, LMU Colloquium in Data Science
Ensemble filters: Updating additional prior state variables

Now have an updated (posterior) ensemble for the unobserved variable.

Fitting Gaussians shows that mean and variance have changed.

Other features of the prior distribution may also have changed.
Without loss of generality (for ensemble Kalman filter)…

Can do data assimilation in the following way.
1. Use model to advance ensemble (3 members here) to time at which next observation becomes available.

Ensemble state estimate after using previous observation (analysis)

Ensemble state at time of next observation (prior)
2. Get prior ensemble sample of observation, $y = h(x)$, by applying forward operator $h$ to each ensemble member.

Theory: observations from instruments with uncorrelated errors can be done sequentially.
3. Get **observed value** and **observational error distribution** from observing system.
4. Find the **increments** for the prior observation ensemble (this is a scalar problem for uncorrelated observation errors).
5. Use ensemble samples of $y$ and each state variable to linearly regress observation increments onto state variable increments.
6. When all ensemble members for each state variable are updated, integrate to time of next observation ...
1. No need for linear model to advance covariance estimate.

2. No need for linear forward operator.
40 state variables: $X_1, X_2, \ldots, X_{40}$.
\[ \frac{dX_i}{dt} = (X_{i+1} - X_{i-2})X_{i-1} - X_i + F. \]
Acts ‘something’ like weather around a latitude band.
Lorenz-96 is sensitive to small perturbations

Introduce 20 ‘ensemble’ state estimates. Each is perturbed for each of the 40-variables at time 0. Refer to unperturbed control integration as ‘truth’.
Assimilate ‘observations’ from 40 random locations.

Interpolate truth to station location.
Simulate observational error:
   Add random draw from $N(0, 16)$ to each.
Start from ‘climatological’ 20-member ensemble.
Some Error Sources in Ensemble Filters

1. Model error
2. Obs. operator error; Representativeness
3. Observation error
4. Sampling Error; Gaussian Assumption
5. Sampling Error; Assuming Linear Statistical Relation
Lorenz-96 Assimilation with localization of observation impact

Localization from Hierarchical Filter

3 Sample Observation Localizations

Jeff Anderson, LMU Colloquium in Data Science
Lorenz-96 Assimilation with localization of observation impact

Localization from Hierarchical Filter

No Localization
1. No need for linear model to advance covariance estimate.

2. No need for linear forward operator.

3. No need for unbiased estimate of covariance.
Some Error Sources in Ensemble Filters

1. Model error

2. Obs. operator error; Representativeness

3. Observation error

4. Sampling Error; Gaussian Assumption

5. Sampling Error; Assuming Linear Statistical Relation
Assimilating in the presence of simulated model error

\[
\frac{dX_i}{dt} = (X_{i+1} - X_{i-2})X_{i-1} - X_i + F.
\]

For truth, use \( F = 8 \).
In assimilating model, use \( F = 6 \).

Time evolution for first state variable shown. Assimilating model quickly diverges from ‘true’ model.
Assimilating in the presence of simulated model error

dXi / dt = (Xi+1 - Xi-2)Xi-1 - Xi + F.
For truth, use F = 8.
In assimilating model, use F = 6.
Use inflation.
Simply increase prior ensemble variance for each state variable. Adaptive algorithms use observations to guide this.
Inflation is a function of state variable and time. Automatically selected by adaptive inflation algorithm.
1. No need for linear model to advance covariance estimate.

2. No need for linear forward operator.

3. No need for unbiased estimate of covariance.

4. No need for unbiased model prior.
Kalman assimilation algorithms assume Gaussians. May be okay for quantity like temperature.

\[ P(x_{t_k} | Y_{t_k}) = \frac{P(y_k | x) P(x_{t_k} | Y_{t_{k-1}})}{\text{Normalization}} \]
Kalman assimilation algorithms assume Gaussians. Tracer concentration is bounded. Gaussian a poor choice.

\[
P(x_{t_k} | Y_{t_k}) = \frac{P(y_k | x) P(x_{t_k} | Y_{t_{k-1}})}{\text{Normalization}}
\]
Bayes Rule (1D example in ‘observation space’)  

Can fit any prior and posterior pdfs, if we can get posterior ensemble.

\[
P(x_{t_k} | Y_{t_k}) = \frac{P(y_k | x) P(x_{t_k} | Y_{t_{k-1}})}{\text{Normalization}}
\]
Step 1: Get continuous prior distribution density.
- Place \((\text{ens}\_\text{size} + 1)^{-1}\) mass between adjacent ensemble members.
- Reminiscent of rank histogram evaluation method.
Observation-Space Rank Histogram Filter

Step 1: Get continuous prior distribution density.

- Place \((\text{ens}_\text{size} + 1)^{-1}\) mass between adjacent ensemble members.
- Reminiscent of rank histogram evaluation method.
Observation-Space Rank Histogram Filter

Step 1: Get continuous prior distribution density.
- Place \((\text{ens}_\text{size} + 1)^{-1}\) mass between adjacent ensemble members.
- Reminiscent of rank histogram evaluation method.
Step 1: Get continuous prior distribution density.

- Place \((\text{ens}_\text{size} + 1)^{-1}\) mass between adjacent ensemble members.
- Reminiscent of rank histogram evaluation method.
Observation-Space Rank Histogram Filter

Step 1: Get continuous prior distribution density.

- Partial gaussian kernels on tails, N(tail\_mean, ens\_sd).
- *tail\_mean selected so that (ens\_size + 1)\(^{-1}\) mass is in tail.*
Observation-Space Rank Histogram Filter

Step 2: Use **likelihood** to compute weight for each ensemble member.
- Analogous to classical particle filter.
- Can be extended to non-gaussian obs. likelihoods.
Step 2: Use **likelihood** to compute weight for each ensemble member.

- Can approximate interior likelihood with linear fit; for efficiency.
Step 3: Compute continuous posterior distribution.

- Approximate likelihood with trapezoidal quadrature, take product.
  
  (Displayed product normalized to make posterior a PDF).
Step 3: Compute continuous posterior distribution.

- Approximate likelihood with trapezoidal quadrature, take product.
  (Displayed product normalized to make posterior a PDF).
Step 3: Compute continuous posterior distribution.

- Approximate likelihood with trapezoidal quadrature, take product.

(Displayed product normalized to make posterior a PDF).
Step 3: Compute continuous posterior distribution.
- Approximate likelihood with trapezoidal quadrature, take product.
  (Displayed product normalized to make posterior a PDF).
Step 3: Compute continuous posterior distribution.
- Product of prior gaussian kernel with likelihood for tails.
- Easy for gaussian likelihood.
Step 4: Compute posterior ensemble members:

- \((\text{ens}_\text{size} + 1)^{-1}\) of posterior mass between each ensemble pair.
- \((\text{ens}_\text{size} + 1)^{-1}\) in each tail.
Compare to standard Ensemble Adjustment Filter (EAKF). Nearly gaussian case, differences are small.
Rank Histogram gets rid of outlier that is clearly inconsistent with obs.

EAKF can’t get rid of outlier.

Large prior variance from outlier causes EAKF to shift all members too much towards observation.
1. No need for linear model to advance covariance estimate.

2. No need for linear forward operator.

3. No need for unbiased estimate of covariance.

4. No need for unbiased model prior.

5. (Almost) no Gaussian assumed for prior.
Step 1: Get continuous prior distribution density (same).
- Partial gaussian kernels on tails, $N(\text{tail\_mean}, \text{ens\_sd})$.
- $\text{tail\_mean}$ selected so that $(\text{ens\_size} + 1)^{-1}$ mass is in tail.
Step 2: Use **likelihood** to compute weight for each ensemble member (same).
Step 3: Compute continuous posterior distribution.

- Approximate likelihood with trapezoidal quadrature.
- Uniform likelihood tails! (Different). No Gaussian assumption left.
Step 3: Compute continuous posterior distribution (same).
- Really simple with uniform likelihood tails.
Step 4: Compute updated ensemble members (same):

- $(\text{ens}\_\text{size} + 1)^{-1}$ of posterior mass between each ensemble pair.
- $(\text{ens}\_\text{size} + 1)^{-1}$ in each tail.
1. No need for linear model to advance covariance estimate.

2. No need for linear forward operator.

3. No need for unbiased estimate of covariance.

4. No need for unbiased model prior.

5. (Almost) no need for Gaussian prior.

6. No need for Gaussian likelihood.

7. Reduced need for linear regression for state increments.
What Kalman assumptions are left?

Still need information from regression for state increments. (I haven’t told you why, see the MARHF paper).

Assumes bivariate information is sufficient.

Not sure how to go further unless…
   Just go to the particle filter.

Lots of fun still left merging ensemble and particle filters!
Learn more about DART at:

www.image.ucar.edu/DAReS/DART

dart@ucar.edu