Building State-of-the-Art Forecast Systems with the Ensemble Kalman Filter
Jeff Anderson, NCAR/IMAGe
Want to predict where the ball will land.
Building a Forecast System

Want to predict where the ball will land.

Time 0.0 secs
Prediction Model
Prediction Model

For the ball this is simple:

\[ x = x_{\text{initial}} + u_{\text{initial}} t \]
\[ y = y_{\text{initial}} + v_{\text{initial}} t - \frac{1}{2} g t^2 \]
Unsure about release point, velocity, angle… Sample this with an ‘ensemble’ of blue balls.
Unsure about release point, velocity, angle… Sample this with an ‘ensemble’ of blue balls.
Need observations (measurements) of the red ball.

All observations have errors.

Observe position of ball every half second after throw.
Observations of the Red Ball

Time 0.5 secs

Distance

Height
Observations of the Red Ball

Time 1.5 secs

Distance

Height

CAWCR Workshop: 5 December 2016
Building a Forecast System

Prediction Model

Observing System

Data Assimilation

Forecasts

Observations
Building a Forecast System

- Prediction Model
- Observing System

Data Assimilation

Forecasts

Analysis

Observations
Assimilating the First Observation

Make large ensemble of forecasts. Closer to observation => more likely. Fifty likely forecasts are shown (darker blue => more likely).
Fifty balls at time 0.5 are an ensemble analysis. Show uncertainty of best estimate of red ball’s location.
Building a Forecast System

Prediction Model → Observing System

Forecast → Analysis

Initial Conditions
Analysis ensemble are initial conditions for 50 forecasts. Green is weighted mean of ensemble forecast at time 2.0. This is best single forecast given observations at time 0.5.
Building a Forecast System

Prediction Model

Observing System

Data Assimilation

Analysis

Forecasts

Initial Conditions

Observations
Assimilating the Second Observation

Start with forecast at time 1.0 that used observations at time 0.5.
Add information from observation at time 1.0.
Assimilating the Second Observation

New ensemble analysis is initial conditions for 50 forecasts. Green is best single forecast of red ball at time 2.0 given observations at time 0.5 and 1.0.
Next ensemble analysis is initial conditions for 50 forecasts.
Next ensemble analysis is initial conditions for 50 forecasts. Green is best single forecast of red ball at time 2.0 given observations at time 0.5, 1.0 and 1.5.
As forecast lead time gets shorter, forecast improves. Get an estimate of forecast uncertainty.
As forecast lead time gets shorter, forecast improves. Get an estimate of forecast uncertainty.
As forecast lead time gets shorter, forecast improves. Get an estimate of forecast uncertainty.
Also can get ‘reanalysis’ for time 0.5. Reanalysis uses past and future observations to get best possible estimate of where the red ball was. Cannot be done in real time.
Also can get ‘reanalysis’ for time 0.5. Reanalysis uses past and future observations to get best possible estimate of where the red ball was. Cannot be done in real time.
Also can get ‘reanalysis’ for time 0.5. Reanalysis uses past and future observations to get best possible estimate of where the red ball was. Cannot be done in real time.
This thrown ball example is in a 2-dimensional space. Really a 4-dimensional ‘phase’ space including velocity.

Atmosphere, ocean, land, coupled models are BIG.

But they’re still just a ‘ball’ moving in a HUGE phase space.

As many as 100 million dimensions at present.
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:

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

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

\[ v_k \to N(0, R_k) \]  \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:

\[ p(x, t \mid Y_t) \]  \hspace{1cm} Analysis  \hspace{1cm} \[ p(x, t^+ \mid Y_t) \]  \hspace{1cm} 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:

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

Apply Bayes’ rule:

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

Noise is white in time (3), so:

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

Integrate numerator to get normalizing denominator:

\[ p(y_k | Y_{t_{k-1}}) = \int p(y_k | x)p(x, t_k | Y_{t_{k-1}}) \, dx \]  \hspace{1cm} (9)
Probability after new observation:

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

(10)
Methods for Solving the Forecast Problem: Particle Filter

Independent evolving estimates,
Associate probability with each estimate given observations,
Eliminate unlikely estimates,
Duplicate likely estimates,
Can represent arbitrary probability distribution,
Scales very poorly for large problems.
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 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.

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

Covariance:

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

Mean:

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

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

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

Numerator is just product of two gaussians.

Denominator just normalizes posterior to be a PDF.
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)
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)
The Kalman Filter

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

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

Must store and invert covariance matrices.

**Too big** to store for large problems.

**Too costly** to invert, $\geq O(n^2)$. 
The Ensemble Kalman Filter

1. Start with ensemble of forecasts.
The Ensemble Kalman Filter

2. Fit a normal to ensemble.
3. Do standard Kalman filter.
Have continuous posterior; need an ensemble.
4. Can create an ensemble with exact sample mean and covariance of continuous posterior.
One-Dimensional Ensemble Kalman Filter: Assimilating an Observation
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.
Single observed variable, single unobserved variable.

So far, we have a known likelihood for a say temperature at BOM.

Now, suppose the model state has an additional variable, temperature at Melbourne Airport.

How should ensemble members update the additional variable?
Assume that all we know is the prior joint distribution.

One variable is observed.

What should happen to the unobserved variable?
Assume that all we know is the prior joint distribution.

One variable is observed.

Update observed variable with ensemble Kalman filter.
Ensemble filters: Updating additional prior state variables

Assume that all we know is the prior joint distribution.

One variable is observed.

Update observed variable with ensemble Kalman filter.
Ensemble filters: Updating additional prior state variables

Assume that all we know is the prior joint distribution.

One variable is observed.

Update observed variable with ensemble Kalman filter.
Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.
Ensemble filters: Updating additional prior state variables

Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.
Ensemble filters: Updating additional prior state variables

Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.
Ensemble filters: Updating additional prior state variables

Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.
Ensemble filters: Updating additional prior state variables

Assume that all we know is the prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.
Ensemble filters: Updating additional prior state variables

Using only increments guarantees that if observation had no impact on observed variable, the unobserved variable is unchanged.

Highly desirable!
Assume that all we know is the prior joint distribution.

How should the unobserved variable be impacted?

1\textsuperscript{st} choice: least squares

Equivalent to linear regression.

Same as assuming binormal prior.
Ensemble filters: Updating additional prior state variables

Have joint prior distribution of two variables.

How should the unobserved variable be impacted?

1st 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.

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.

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

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

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.

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

Compressed these two.
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.
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.
1. Use model to advance ensemble (3 members here) to time at which next observation becomes available.
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).
4. Find the increments for the prior observation ensemble (this is a scalar problem for uncorrelated observation errors).

Note: Difference between various ensemble filter methods is primarily in observation increment calculation.
5. Use ensemble samples of $y$ and each state variable to linearly regress observation increments onto state variable increments.
How an Ensemble Filter Works for Geophysical Data Assimilation

5. Use ensemble samples of $y$ and each state variable to linearly regress observation increments onto state variable increments.

Theory: impact of observation increments on each state variable can be handled independently!
5. Use ensemble samples of $y$ and each state variable to linearly regress observation increments onto state variable increments.

DART updates all state variables in parallel. Variables randomly assigned to processes for load balancing.
6. When all ensemble members for each state variable are updated, integrate to time of next observation ...
Ensemble Filter for Lorenz-96 40-Variable Model

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’.

![Graph showing state variable over time with 'truth' and 'ensemble' highlighted.](image)
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
Sampling Error: Observations Impact Unrelated State Variables

Plot shows expected absolute value of sample correlation vs. true correlation.

Unrelated obs. reduce spread, increase error.

Attack with localization.

Reduce impact of observation on weakly correlated state variables.

Let weight go to zero for many ‘unrelated’ variables to save on computing.
Lorenz-96 Assimilation with localization of observation impact

Localization from Hierarchical Filter

No Localization
Lorenz-96 Assimilation with localization of observation impact

Localization from Hierarchical Filter

3 Sample Observation Localizations

State Variable
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

dXi / dt = (Xi+1 - Xi-2)Xi-1 - Xi + 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.
Reduce confidence in prior to deal with model error

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.
For linear, gaussian problem:

If, ensemble size $N > N_{\text{crit}}$

Mean and covariance are identical to Kalman Filter,

Else

Diverges.

$N_{\text{crit}}$: Number of positive singular values in SVD of covariance matrix.
(Ensemble) KF optimal for linear model, gaussian likelihood, perfect model.

In KF, only mean and covariance have meaning.

Ensemble allows computation of many other statistics.

What do they mean? Not entirely clear.

What do they mean when there are all sorts of error?
Even less clear.

Must Calibrate and Validate results.
Basic Ensemble Kalman Filter is trivial.

Good ensemble filters require inflation, localization, …

‘Automated’ inflation, localization algorithms exist.

Parallel implementations for 100,000 cores for large models.

Calibration and validation essential.

Hybrids with variational methods.

Other enhancements in the pipeline.
Learn more about DART at:

www.image.ucar.edu/DAReS/DART

dart@ucar.edu