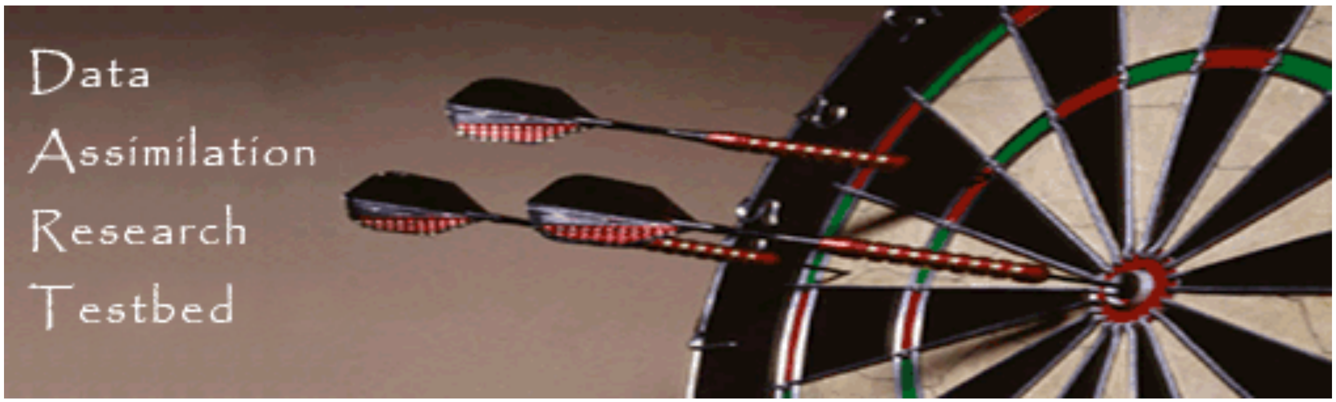
Parallel Implementation of Sequential Ensemble Kalman Filters



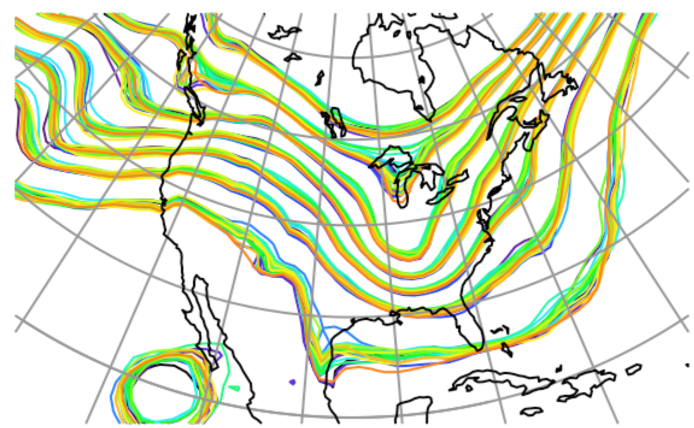
Jeffrey Anderson

NCAR Institute for Math Applied to Geophysics

What is Data Assimilation?

Observations combined with a Model forecast…

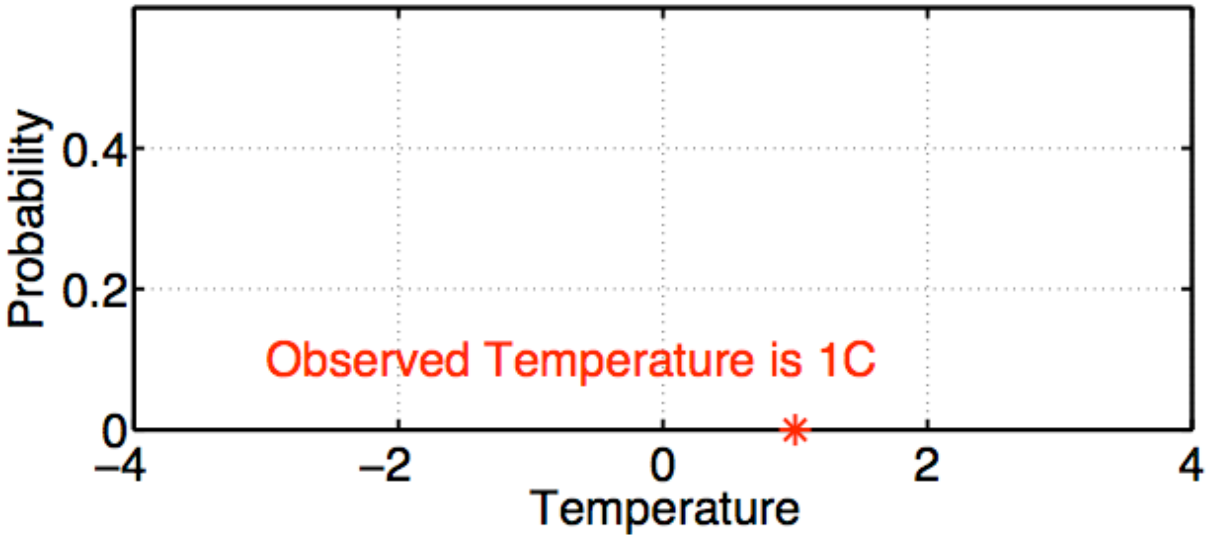
+



…to produce an analysis
(best possible estimate).

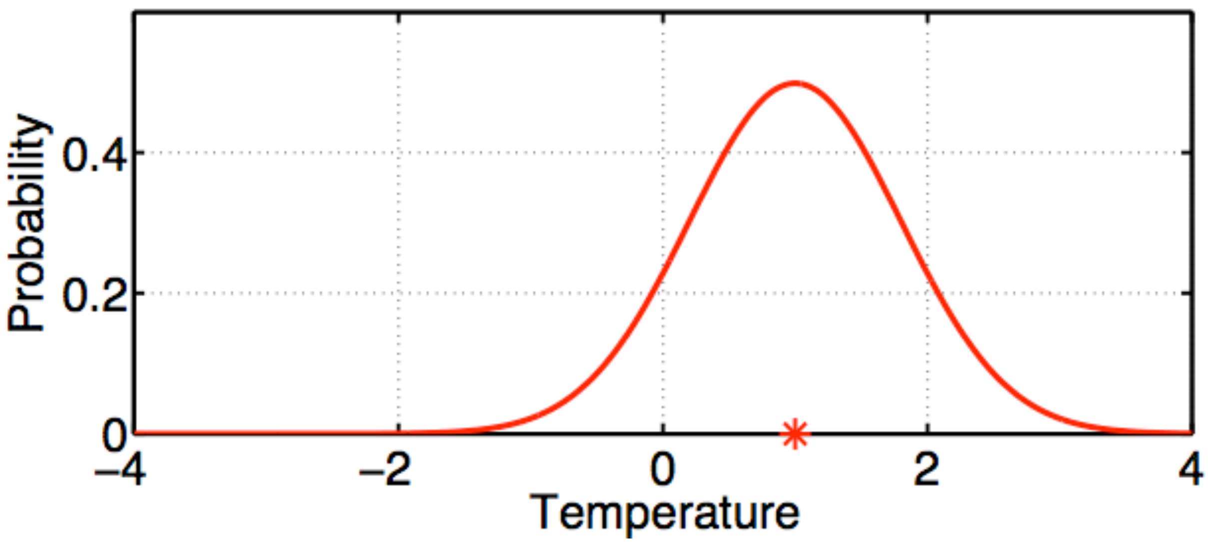
Example: Estimating the Temperature Outside

An observation has a value (*),



Example: Estimating the Temperature Outside

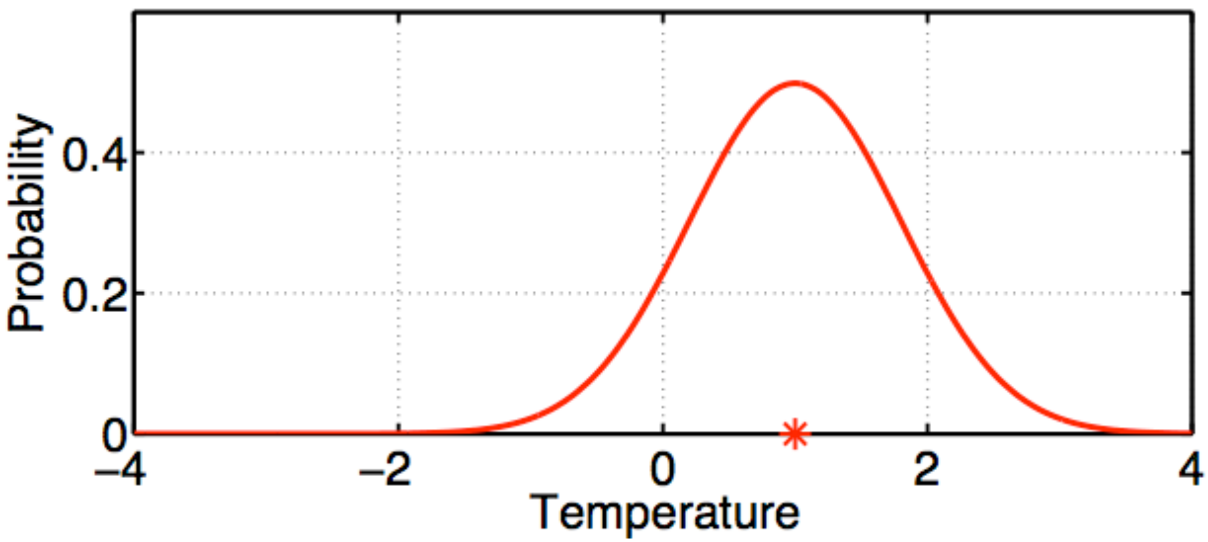
An observation has a value (*),



and an error distribution (red curve) that is associated with the instrument.

Example: Estimating the Temperature Outside

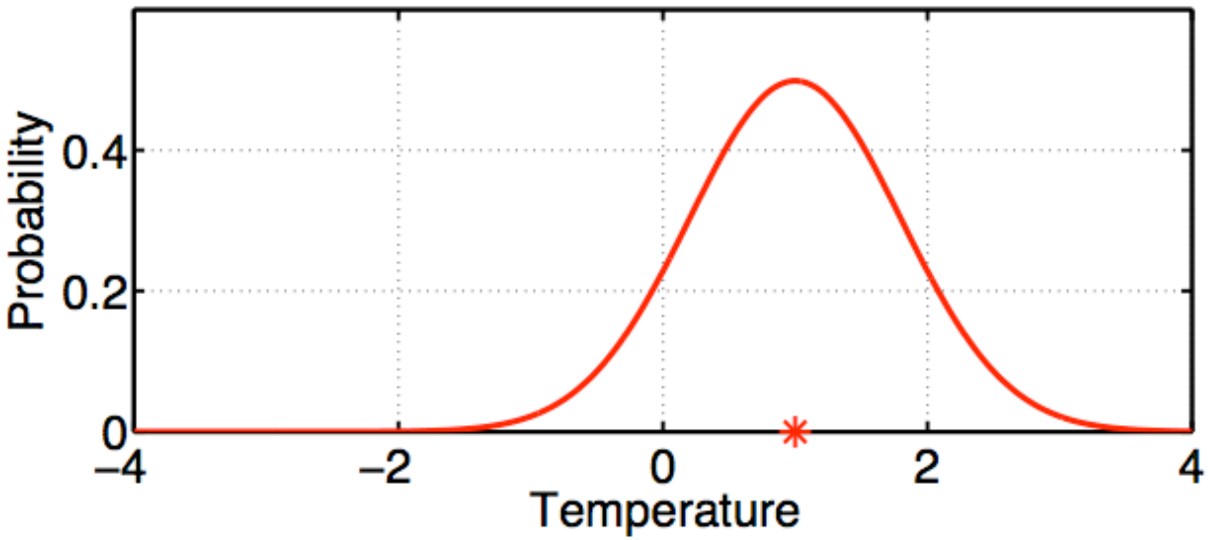
Thermometer outside measures 1C.



Instrument builder says thermometer is unbiased with +/- 0.8C gaussian error.

Example: Estimating the Temperature Outside

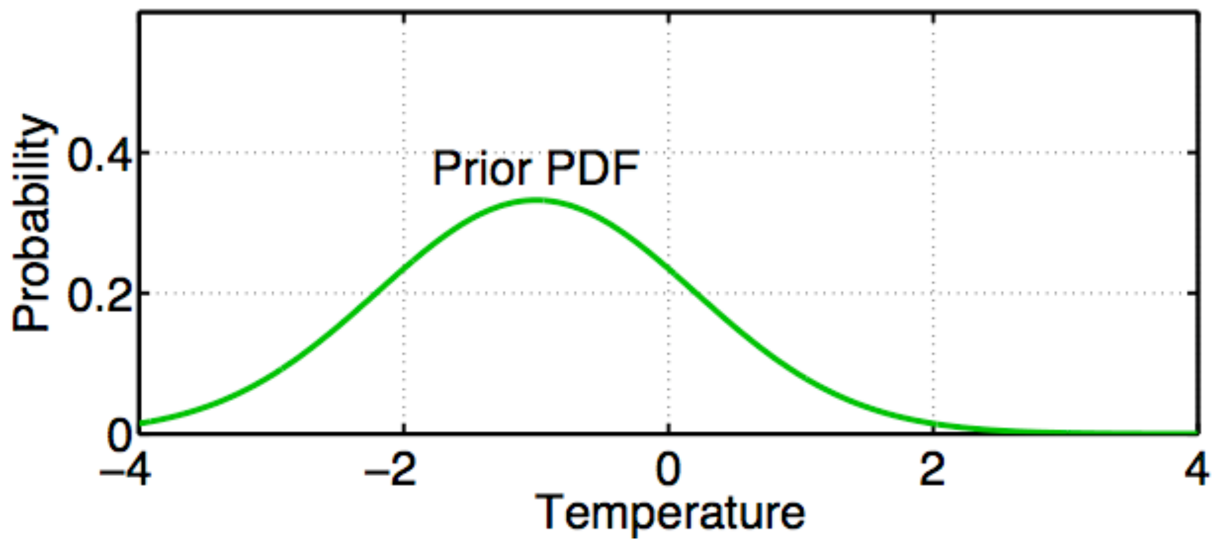
Thermometer outside measures 1C.



The red plot is $P(T \mid T_o)$, probability of temperature given that T_o was observed.

Example: Estimating the Temperature Outside

We also have a prior estimate of temperature.



The green curve is $P(T \mid C)$; probability of temperature given all available prior information C.

Example: Estimating the Temperature Outside

Prior information C can include:

1. Observations of things besides T;
2. Model forecast made using observations at earlier times;
3. *A priori* physical constraints (T > -273.15C);
4. Climatological constraints (-30C < T < 40C).

Combining the Prior Estimate and Observation

Bayes Theorem: $$P(T \mid T_o, C) = \frac{P(T_o \mid T, C)P(T \mid C)}{P(T_o \mid C)}$$

Prior

Posterior: Probability of T given observations and Prior. Also called update or analysis.

Likelihood: Probability that T_o is observed if T is true value and given prior information C.

Combining the Prior Estimate and Observation

Rewrite Bayes as:

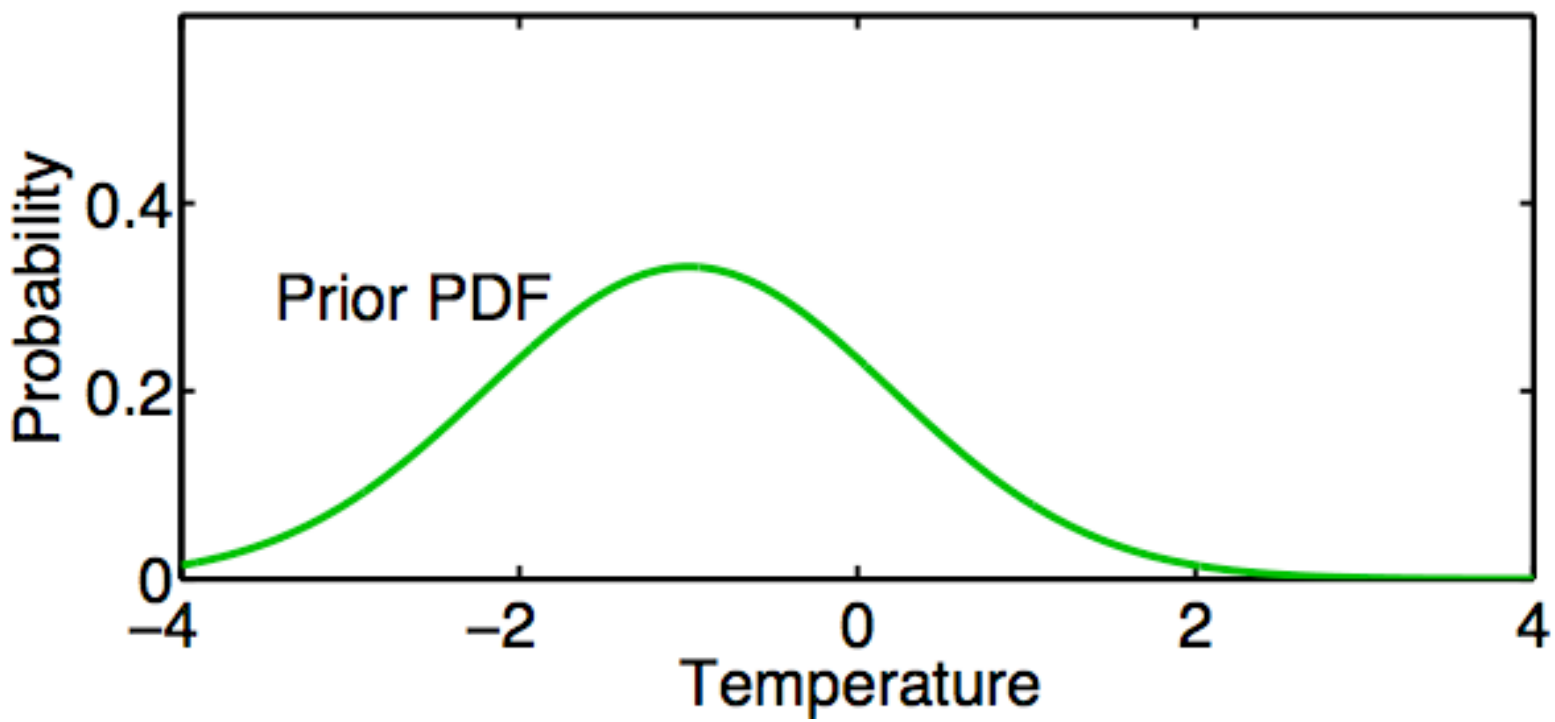
$$\frac{P(T_o \mid T,C)P(T \mid C)}{P(T_o \mid C)} = \frac{P(T_o \mid T,C)P(T \mid C)}{\int P(T_o \mid x)P(x \mid C)dx}$$

$$= \frac{P(T_o \mid T,C)P(T \mid C)}{normalization}$$

Denominator normalizes so Posterior is PDF.

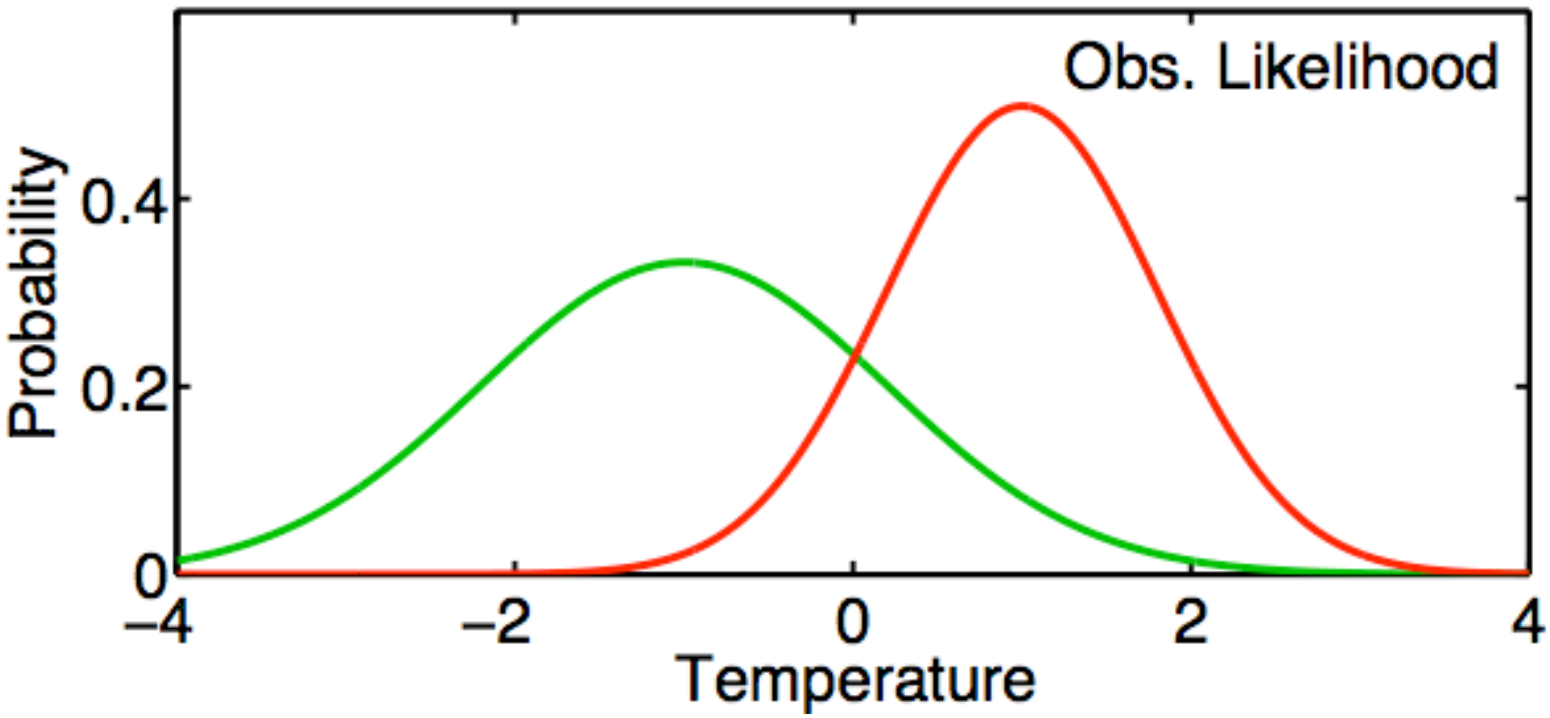
Combining the Prior Estimate and Observation

$$P(T \mid T_o, C) = \frac{P(T_o \mid T, C)\,P(T \mid C)}{normalization}$$



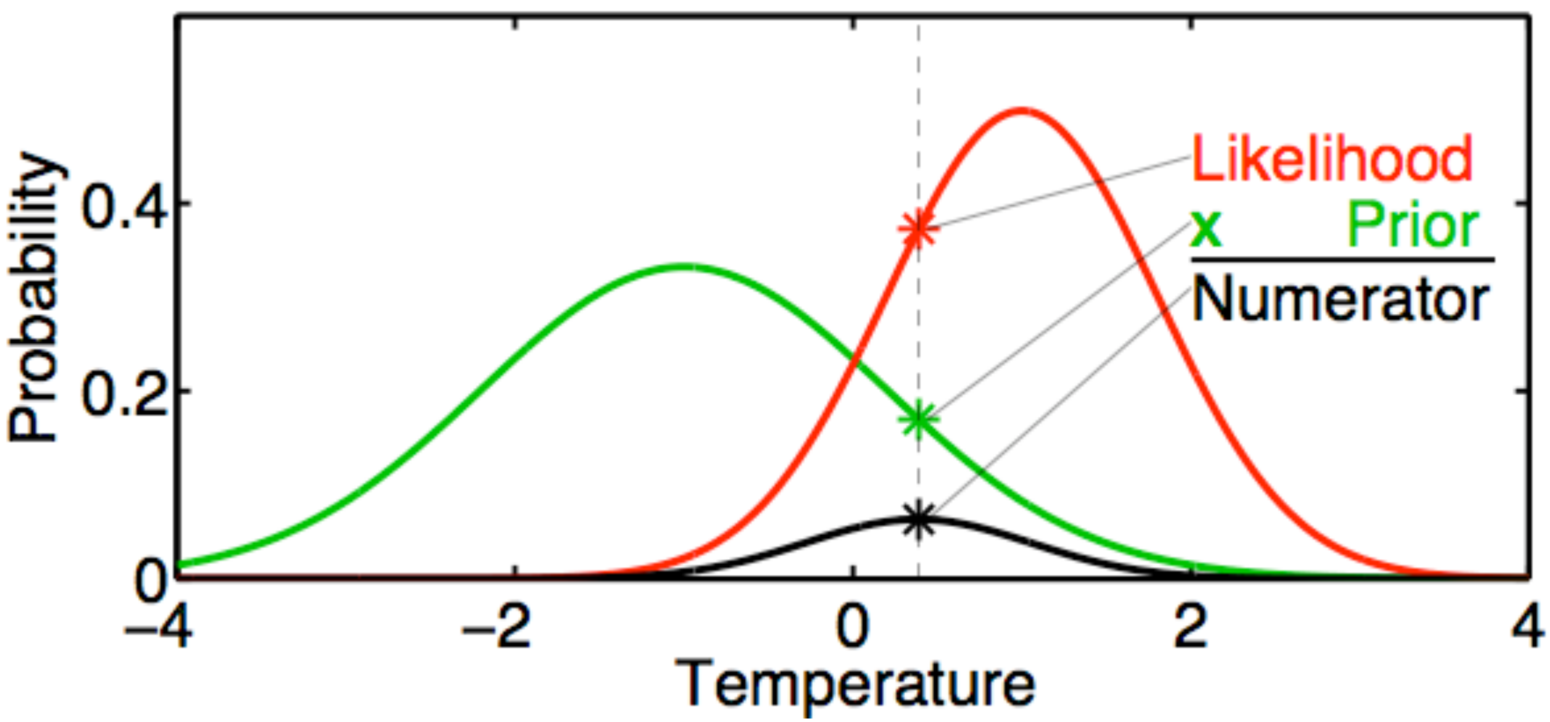
Combining the Prior Estimate and Observation

$$P(T \mid T_o, C) = \frac{P(T_o \mid T, C)\,P(T \mid C)}{normalization}$$



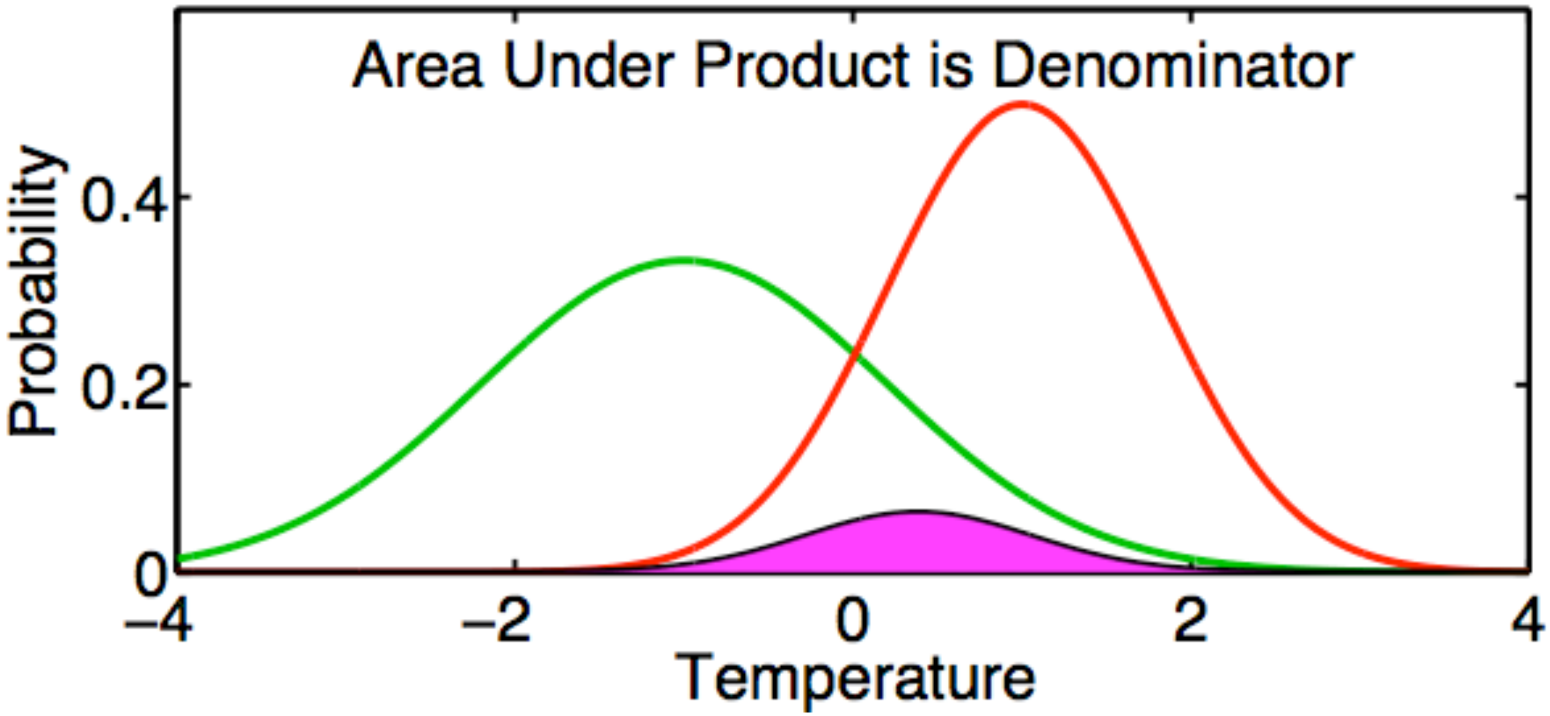
Combining the Prior Estimate and Observation

$$P(T \mid T_o, C) = \frac{P(T_o \mid T, C)\, P(T \mid C)}{normalization}$$



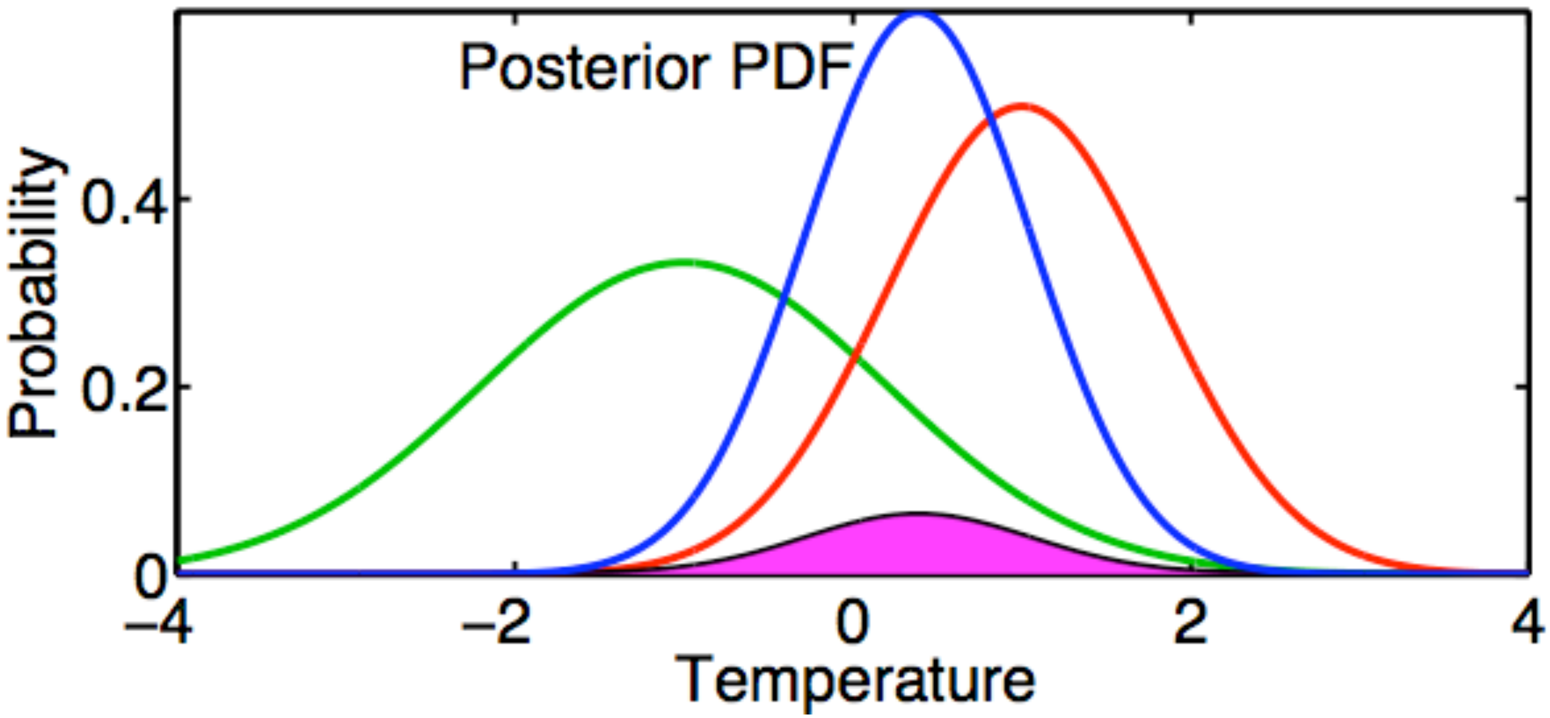
Combining the Prior Estimate and Observation

$$P(T \mid T_o, C) = \frac{P(T_o \mid T, C)\,P(T \mid C)}{normalization}$$



Combining the Prior Estimate and Observation

$$P(T \mid T_o, C) = \frac{P(T_o \mid T, C)\, P(T \mid C)}{normalization}$$



Consistent Color Scheme Throughout Tutorial

Green = Prior

Red = Observation

Blue = Posterior

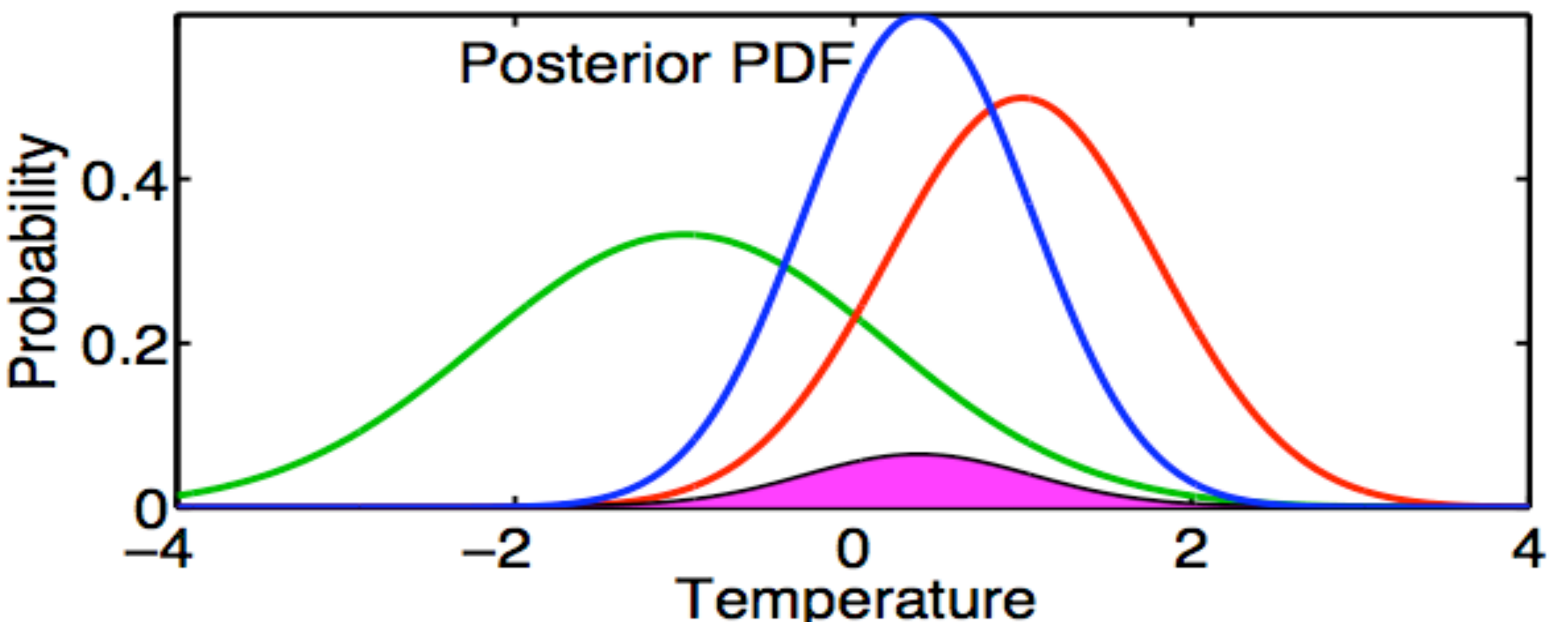
Black = Truth

(truth available only for ‘perfect model’ examples)

Combining the Prior Estimate and Observation

$$P(T \mid T_o, C) = \frac{P(T_o \mid T, C)\,P(T \mid C)}{normalization}$$

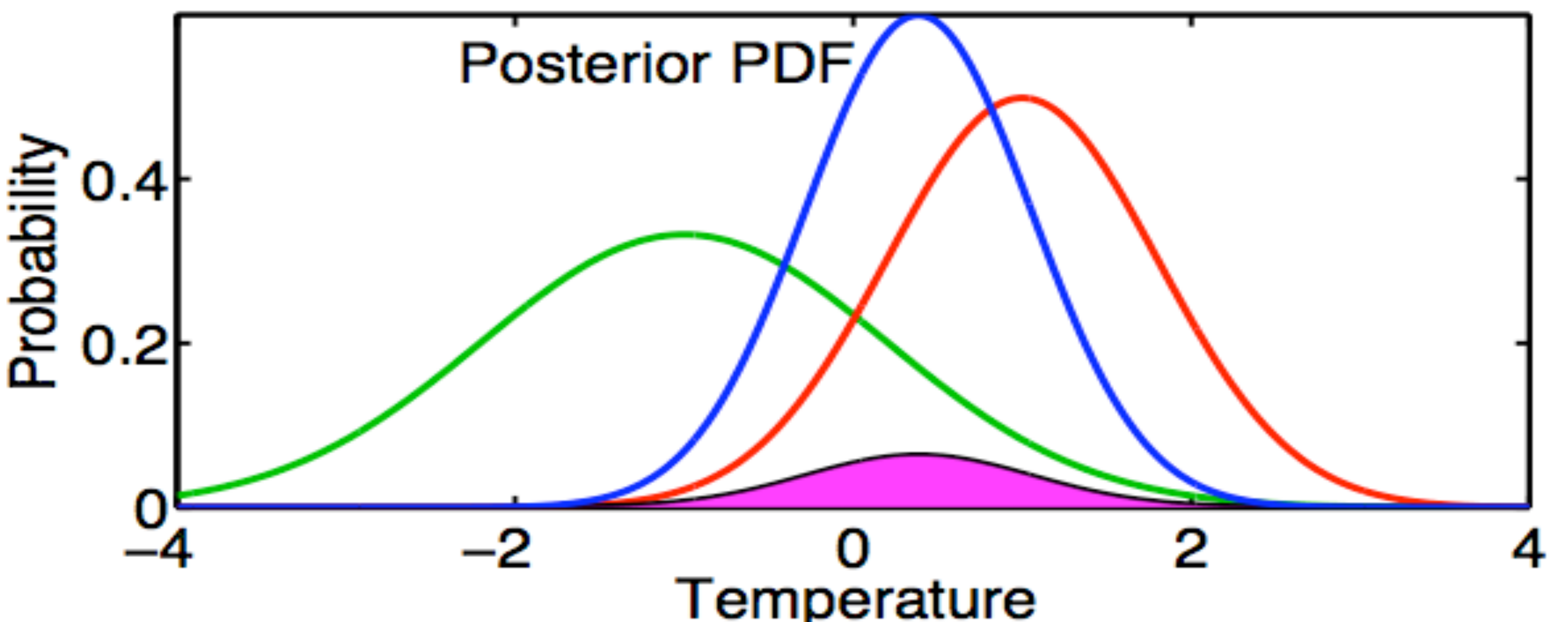
Generally no analytic solution for Posterior.



Combining the Prior Estimate and Observation

$$P(T \mid T_o, C) = \frac{P(T_o \mid T, C)\, P(T \mid C)}{normalization}$$

Gaussian Prior and Likelihood -> Gaussian Posterior



Combining the Prior Estimate and Observation

For Gaussian prior and likelihood…

Prior $$P(T \mid C) = Normal(T_p, \sigma_p)$$

Likelihood $$P(T_o \mid T, C) = Normal(T_o, \sigma_o)$$

Then, Posterior $$P(T \mid T_o, C) = Normal(T_u, \sigma_u)$$

With

$$\sigma_u = \sqrt{\left(\sigma_p^{-2} + \sigma_o^{-2}\right)^{-1}}$$

$$T_u = \sigma_u^2\left[\sigma_p^{-2} T_p + \sigma_o^{-2} T_o\right]$$

The One-Dimensional Kalman Filter

1. Suppose we have a linear forecast model L

   A. If temperature at time $t_1 = T_1$, then temperature at $t_2 = t_1 + \Delta t$ is $T_2 = L(T_1)$

   B. Example: $T_2 = T_1 + \Delta t T_1$

The One-Dimensional Kalman Filter

1. Suppose we have a linear forecast model L.

   A. If temperature at time $t_1 = T_{1,}$ then
      temperature at $t_2 = t_1 + \Delta t$ is $T_2 = L(T_1)$.

   B. Example: $T_2 = T_1 + \Delta t T_1$.

2. If posterior estimate at time t_1 is *Normal*($T_{u,1}$, $\sigma_{u,1}$) then
   prior at t_2 is *Normal*($T_{p,2}$, $\sigma_{p,2}$).

   $T_{p,2} = T_{u,1}, + \Delta t T_{u,1}$

   $\sigma_{p,2} = (\Delta t + 1)\ \sigma_{u,1}$

The One-Dimensional Kalman Filter

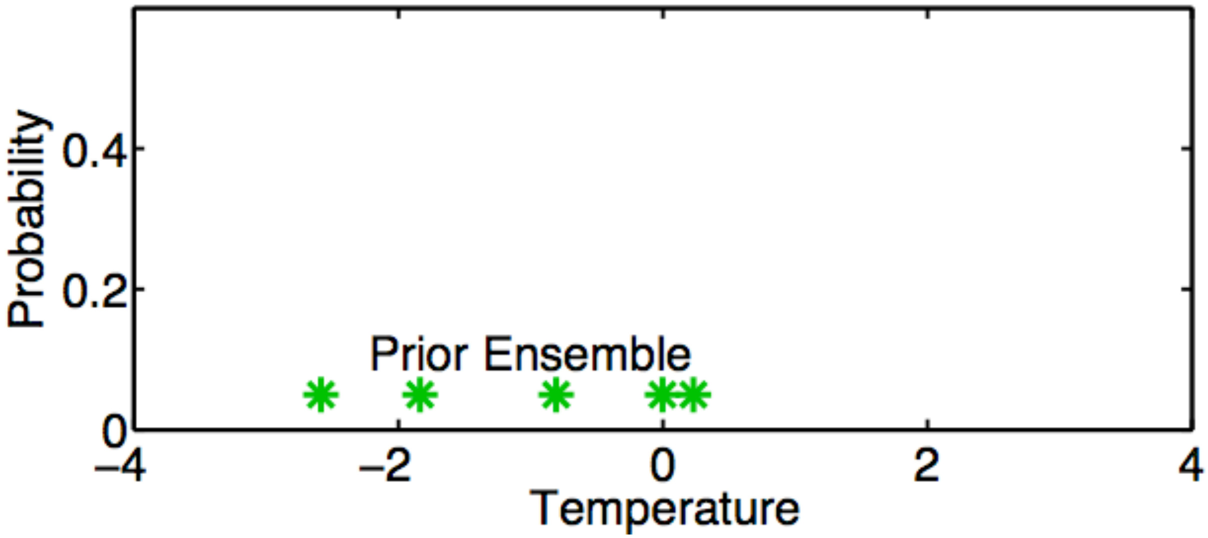
1. Suppose we have a linear forecast model L.
   A. If temperature at time $t_1 = T_1$, then
      temperature at $t_2 = t_1 + \Delta t$ is $T_2 = L(T_1)$.
   B. Example: $T_2 = T_1 + \Delta t T_1$.
2. If posterior estimate at time t_1 is *Normal*($T_{u,1}$, $\sigma_{u,1}$) then
   prior at t_2 is *Normal*($T_{p,2}$, $\sigma_{p,2}$).
3. Given an observation at t_2 with distribution *Normal*(t_o, σ_o)
   the likelihood is also *Normal*(t_o, σ_o).

The One-Dimensional Kalman Filter

1. Suppose we have a linear forecast model L.
   A. If temperature at time $t_1 = T_{1,}$ then temperature at $t_2 = t_1 + \Delta t$ is $T_2 = L(T_1)$.
   B. Example: $T_2 = T_1 + \Delta t T_1$.
2. If posterior estimate at time t_1 is *Normal*($T_{u,1}$, $\sigma_{u,1}$) then prior at t_2 is *Normal*($T_{p,2}$, $\sigma_{p,2}$).
3. Given an observation at t_2 with distribution *Normal*(t_o, σ_o) the likelihood is also *Normal*(t_o, σ_o).
4. The posterior at t_2 is *Normal*($T_{u,2}$, $\sigma_{u,2}$) where $T_{u,2}$ and $\sigma_{u,2}$ come from page 19.

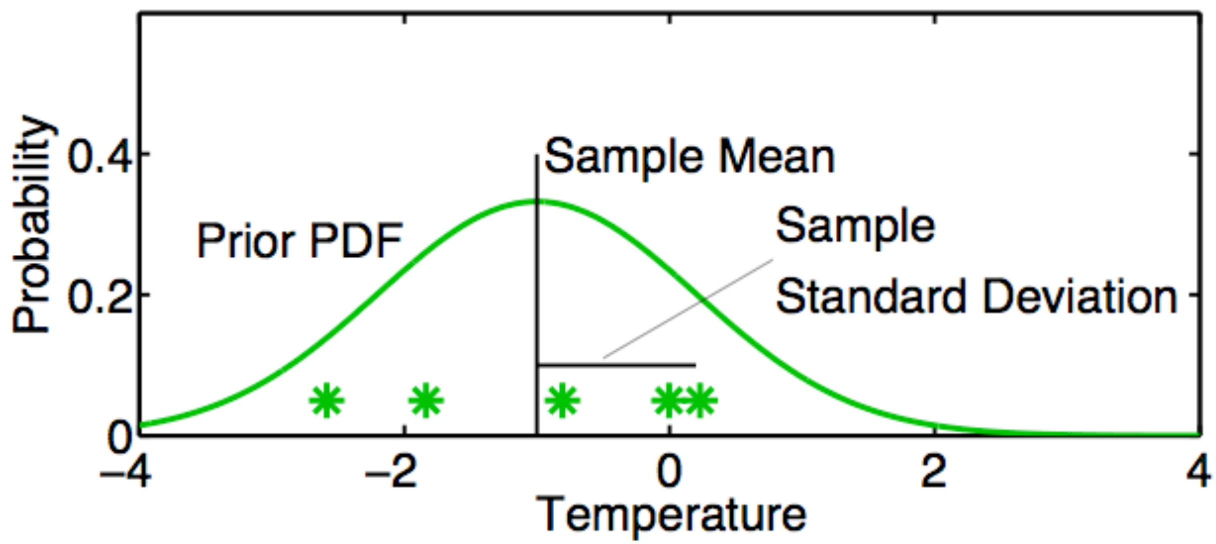
A One-Dimensional Ensemble Kalman Filter

Represent a prior pdf by a sample (ensemble) of N values:



A One-Dimensional Ensemble Kalman Filter

Represent a prior pdf by a sample (ensemble) of N values:



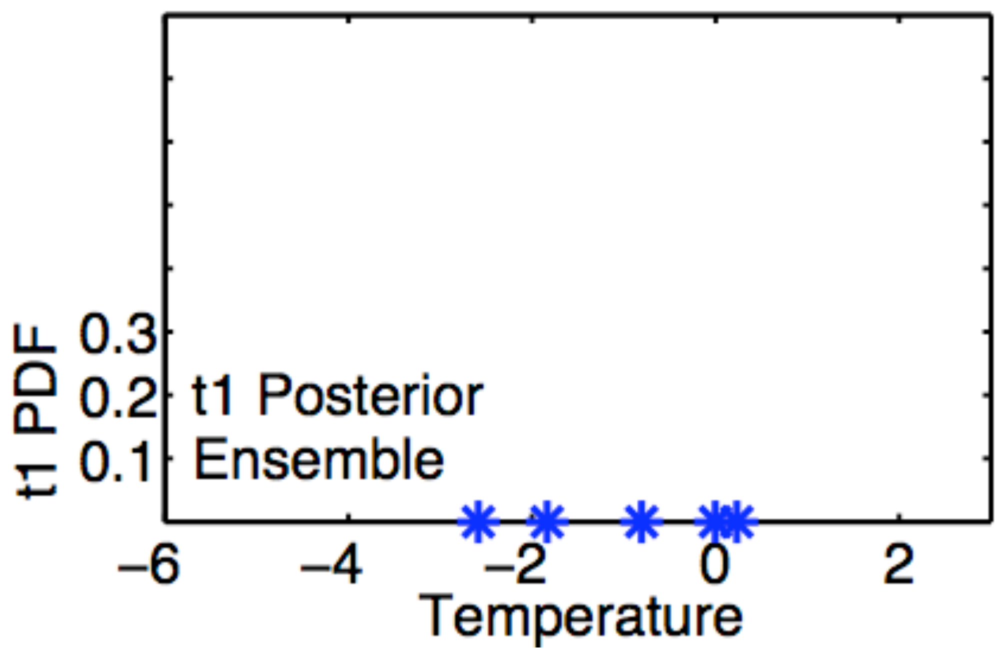
Use sample mean $\overline{T} = \sum_{n=1}^{N} T_n / N$

and sample standard deviation $\sigma_T = \sqrt{\sum_{n=1}^{N} \left(T_n - \overline{T}\right)^2 / (N-1)}$

to determine a corresponding continuous distribution $Normal\left(\overline{T}, \sigma_T\right)$

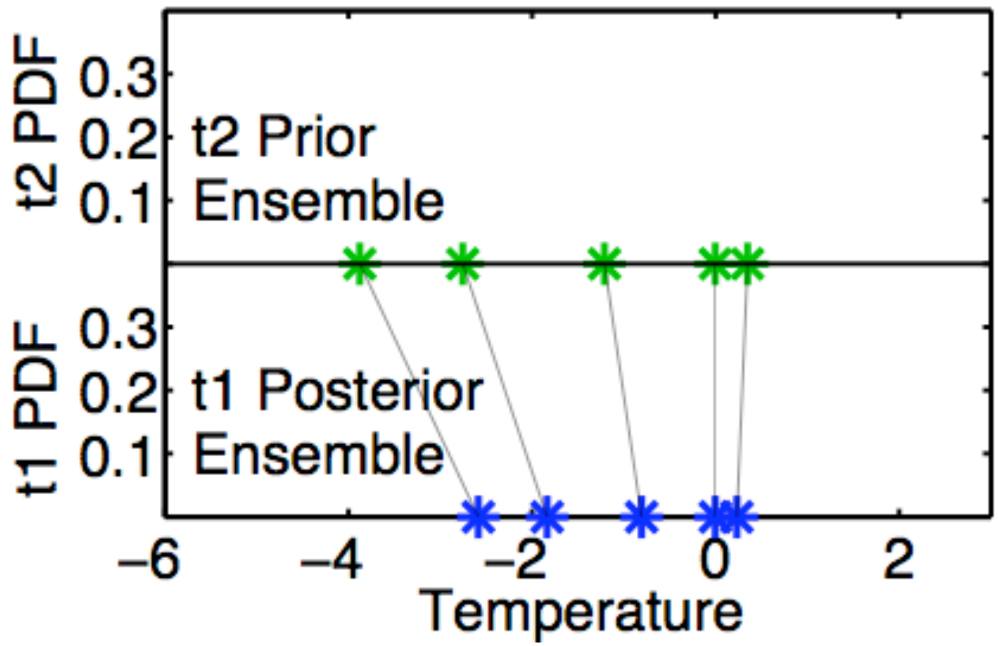
A One-Dimensional Ensemble Kalman Filter: Model Advance

If posterior ensemble at time t_1 is $T_{1,n}$, $n = 1, \ldots, N$



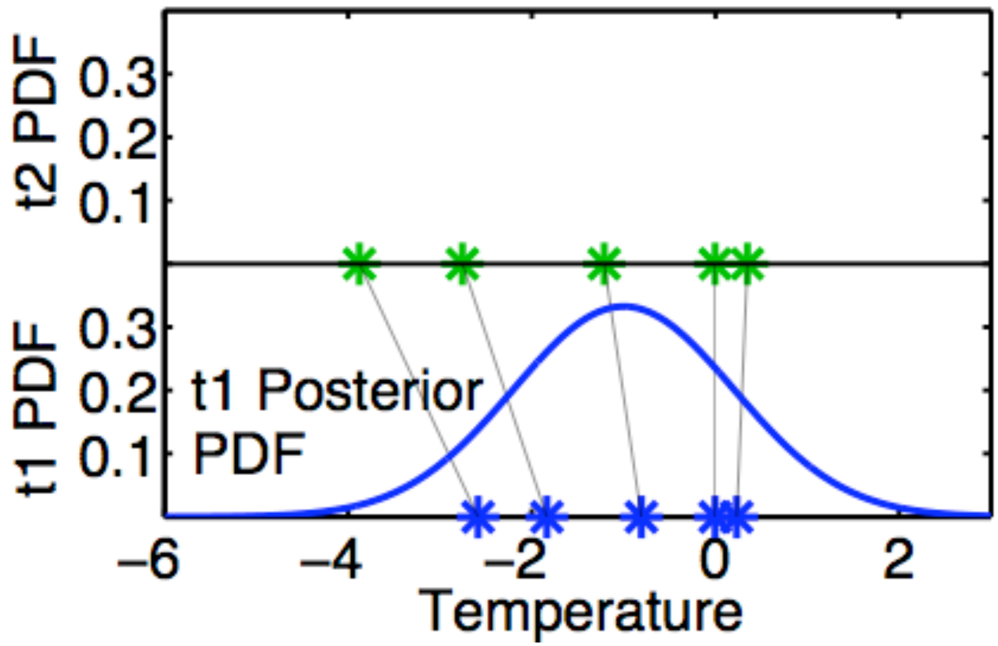
A One-Dimensional Ensemble Kalman Filter: Model Advance

If posterior ensemble at time t_1 is $T_{1,n}$, $n = 1, \ldots N$,
advance each member to time t_2 with model, $T_{2,n} = L(T_{1,n})$ $n = 1, \ldots, N$.



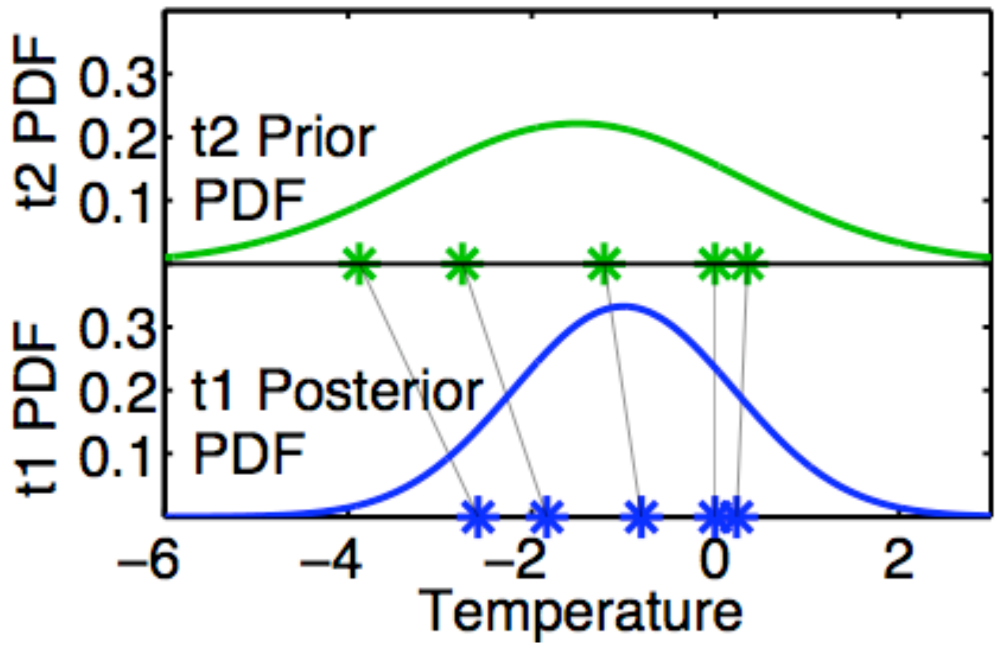
A One-Dimensional Ensemble Kalman Filter: Model Advance

Same as advancing continuous pdf at time t_1 …

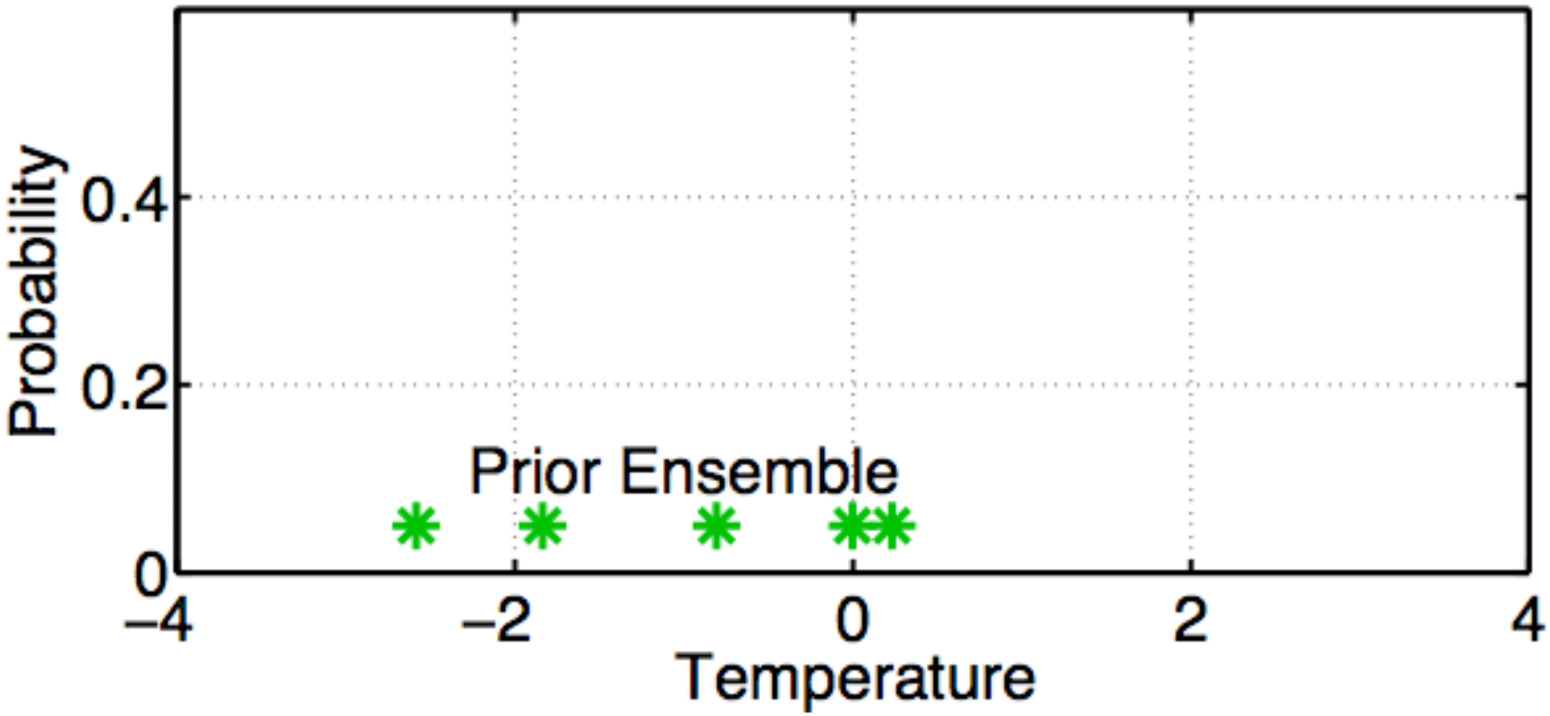


A One-Dimensional Ensemble Kalman Filter: Model Advance

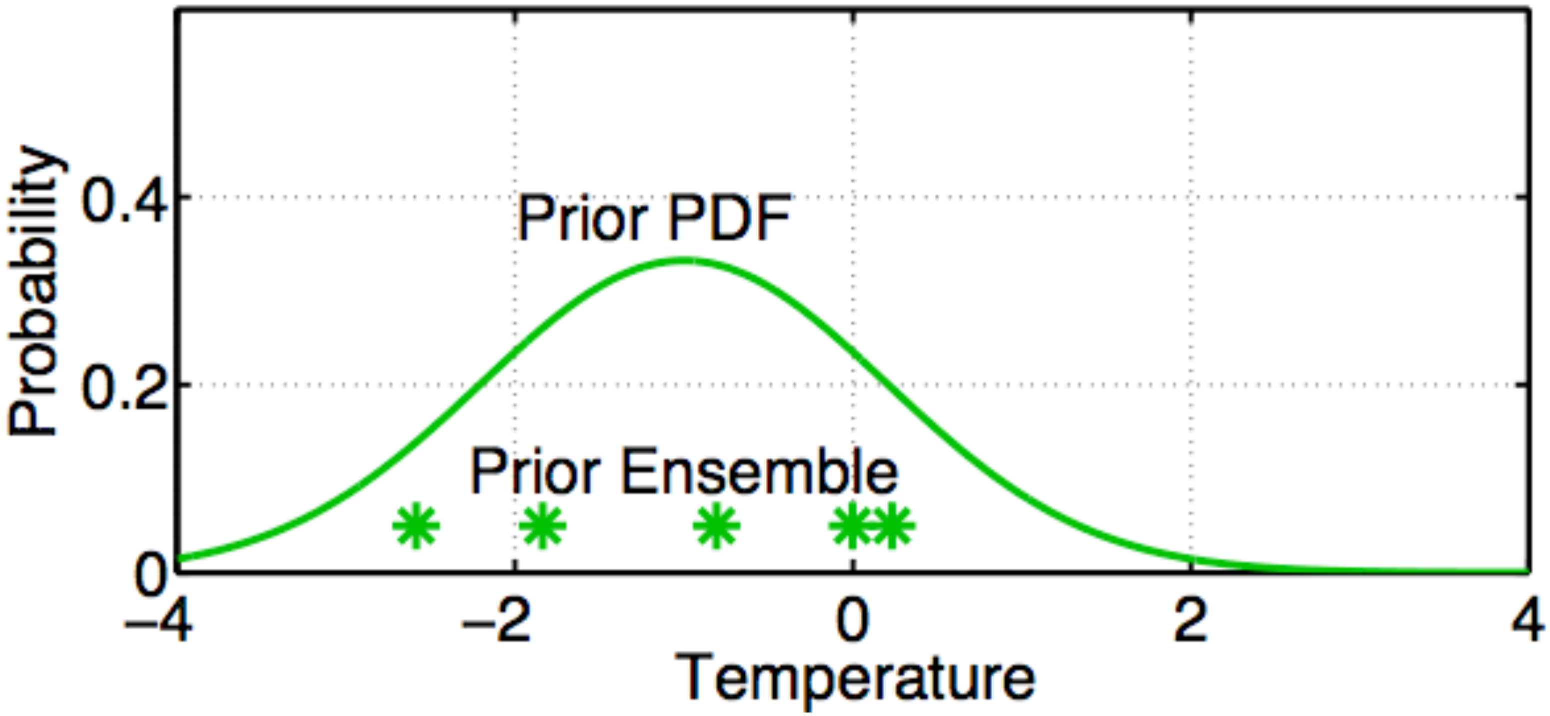
Same as advancing continuous pdf at time t_1 to time t_2 with model L.



A One-Dimensional Ensemble Kalman Filter:
Assimilating an Observation

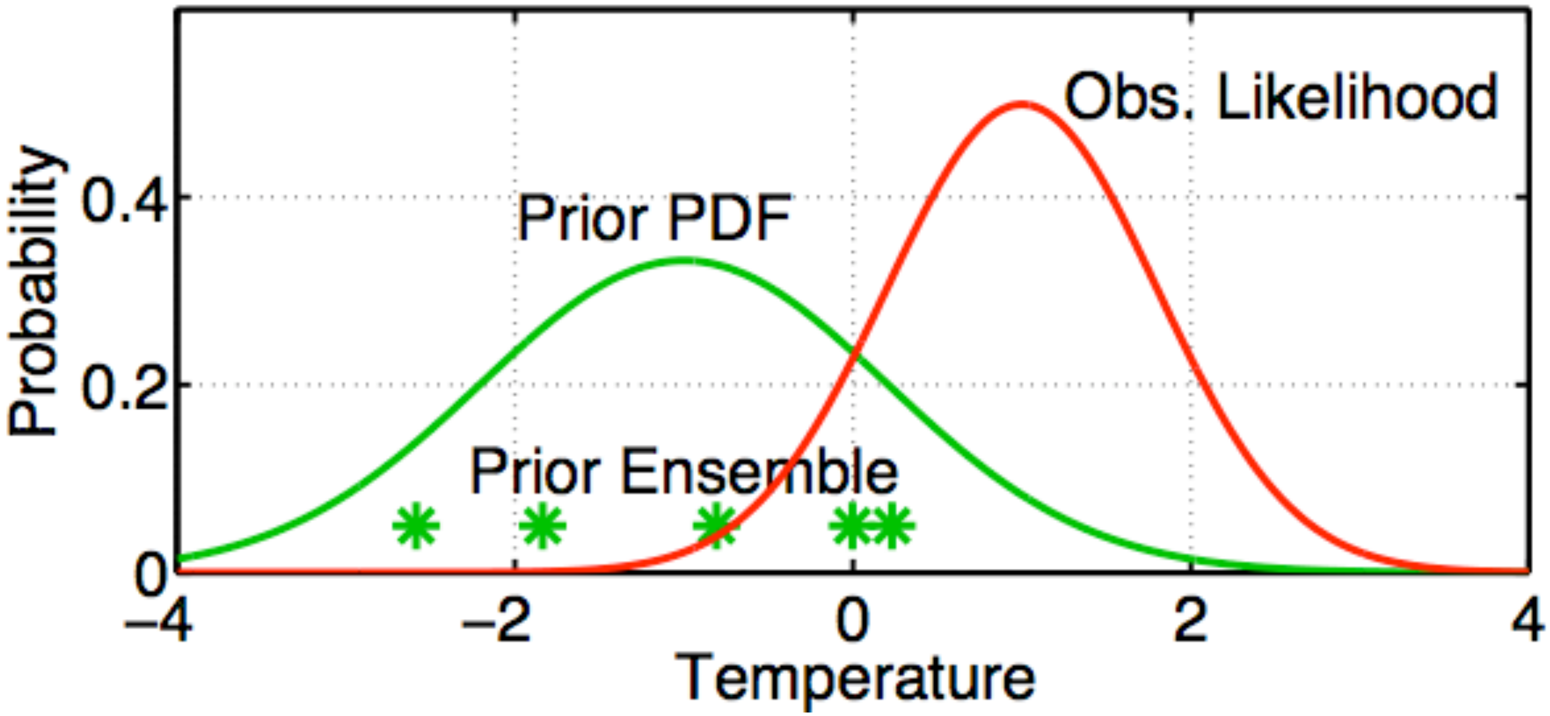


A One-Dimensional Ensemble Kalman Filter: Assimilating an Observation



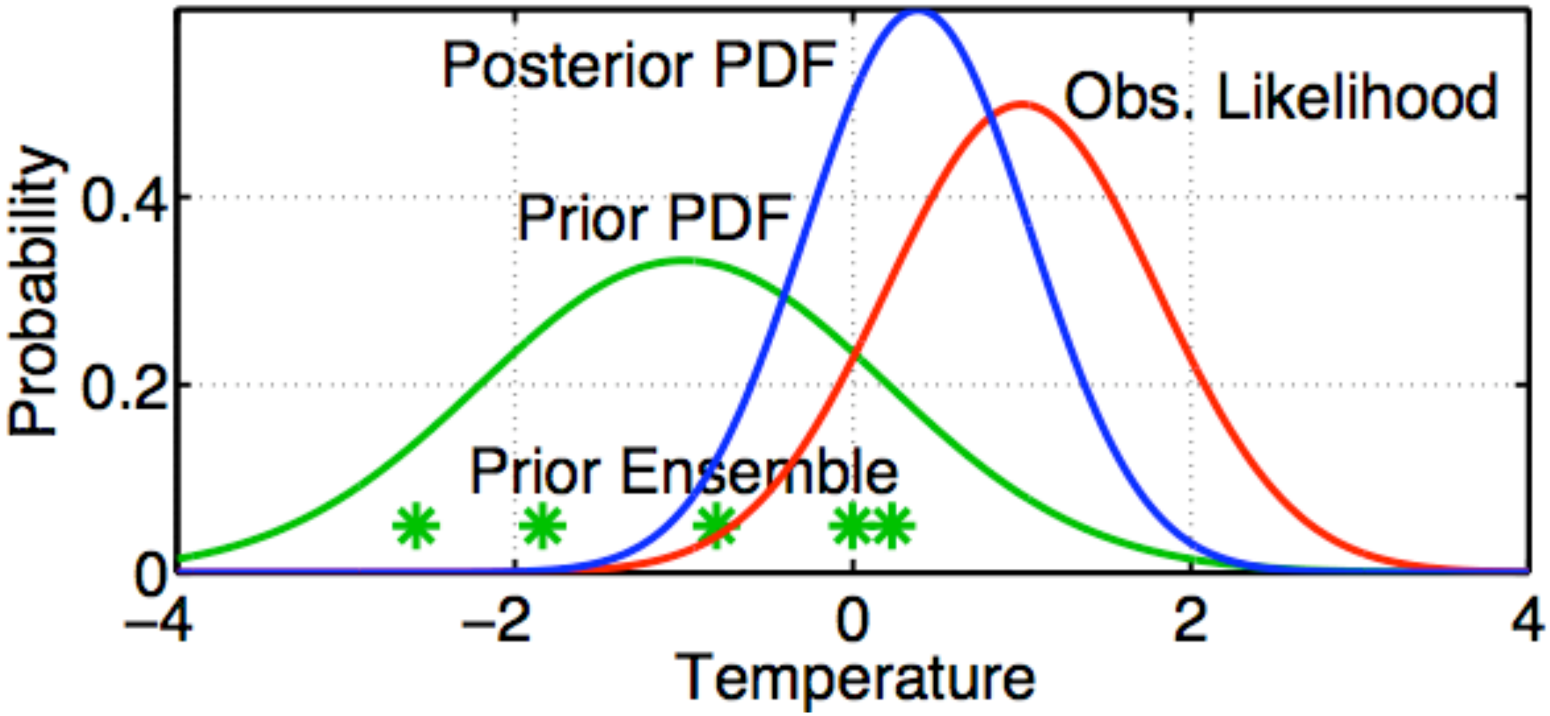
Fit a Gaussian to the sample.

A One-Dimensional Ensemble Kalman Filter: Assimilating an Observation



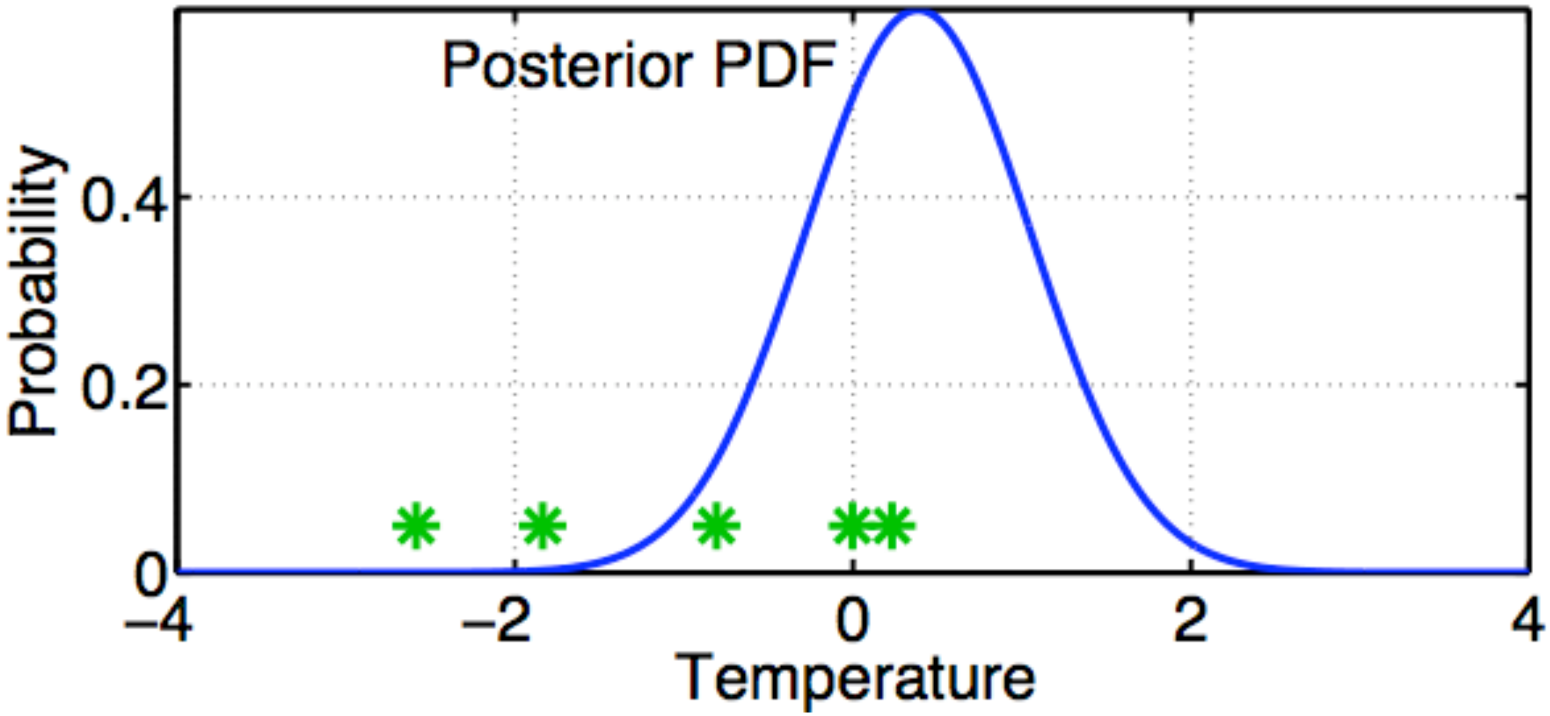
Get the observation likelihood.

A One-Dimensional Ensemble Kalman Filter:
Assimilating an Observation



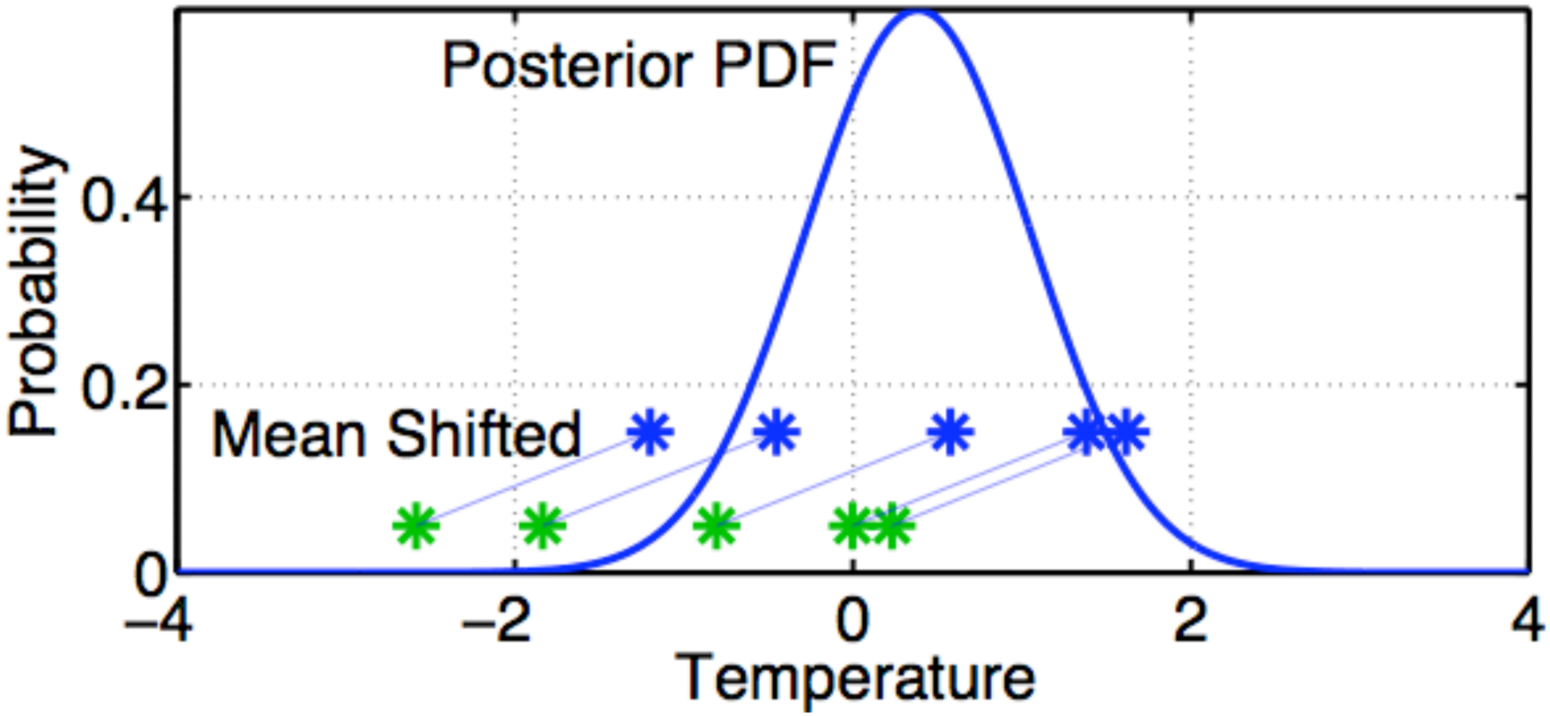
Compute the continuous posterior PDF.

A One-Dimensional Ensemble Kalman Filter: Assimilating an Observation



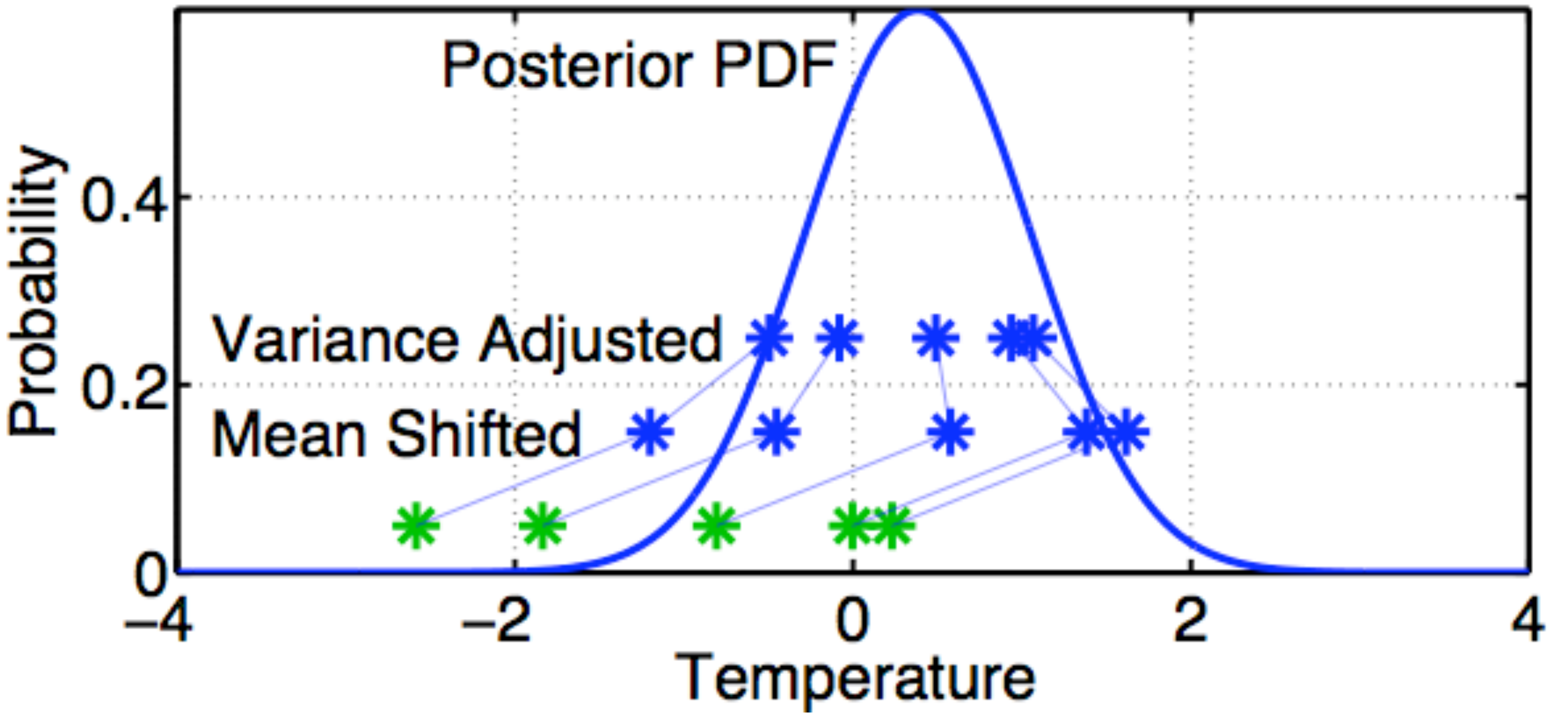
Use a deterministic algorithm to 'adjust' the ensemble.

A One-Dimensional Ensemble Kalman Filter: Assimilating an Observation



First, 'shift' the ensemble to have the exact mean of the posterior.

A One-Dimensional Ensemble Kalman Filter: Assimilating an Observation

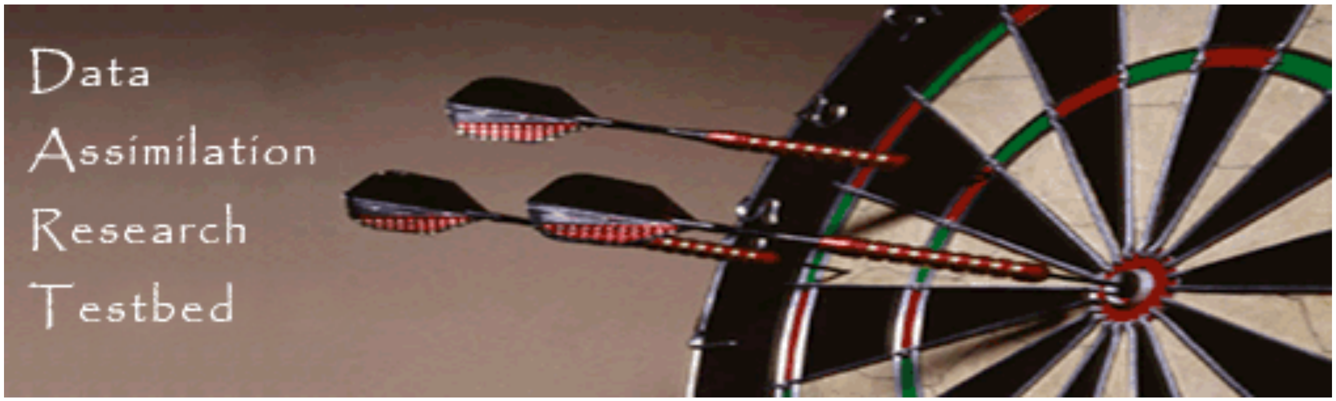


First, 'shift' the ensemble to have the exact mean of the posterior.

Second, linearly contract to have the exact variance of the posterior.

Sample statistics are identical to Kalman filter.

We now know how to assimilate a single observed variable.



Section 2: How should observations of one state variable impact an unobserved state variable?

Single observed variable, single unobserved variable

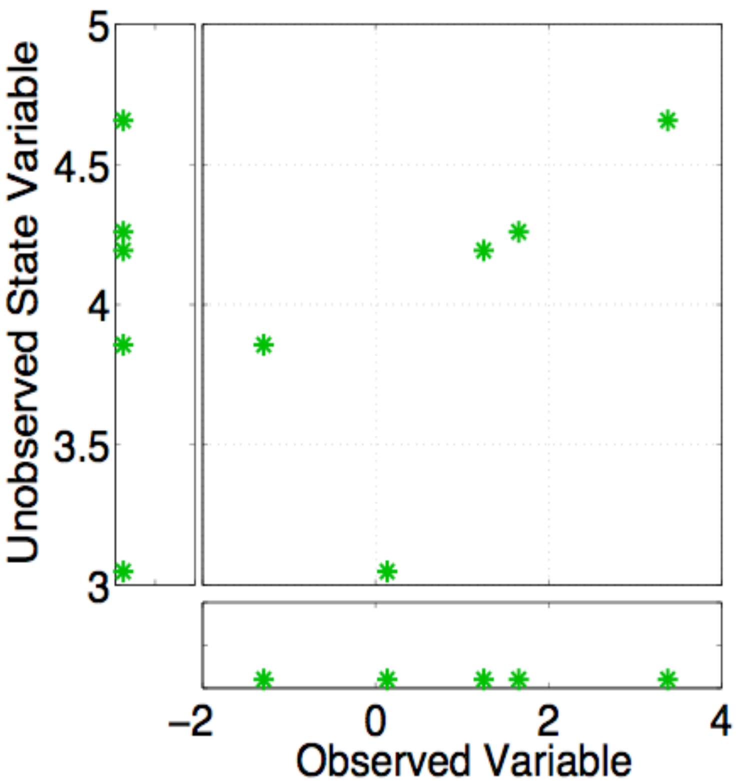
So far, we have a known observation likelihood for single variable.

Now, suppose the prior has an additional variable.

Examine how ensemble members update the additional variable.

Basic method generalizes to any number of additional variables.

Ensemble filters: Updating additional prior state variables

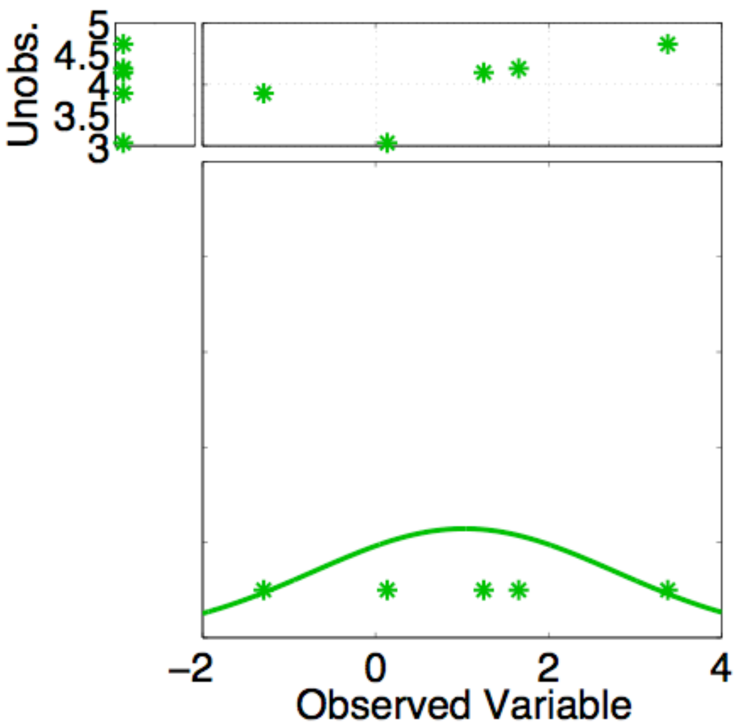


Assume that all we know is prior joint distribution.

One variable is observed, temperature at Reading.

What should happen to an unobserved variable, like temperature at London?

Ensemble filters: Updating additional prior state variables

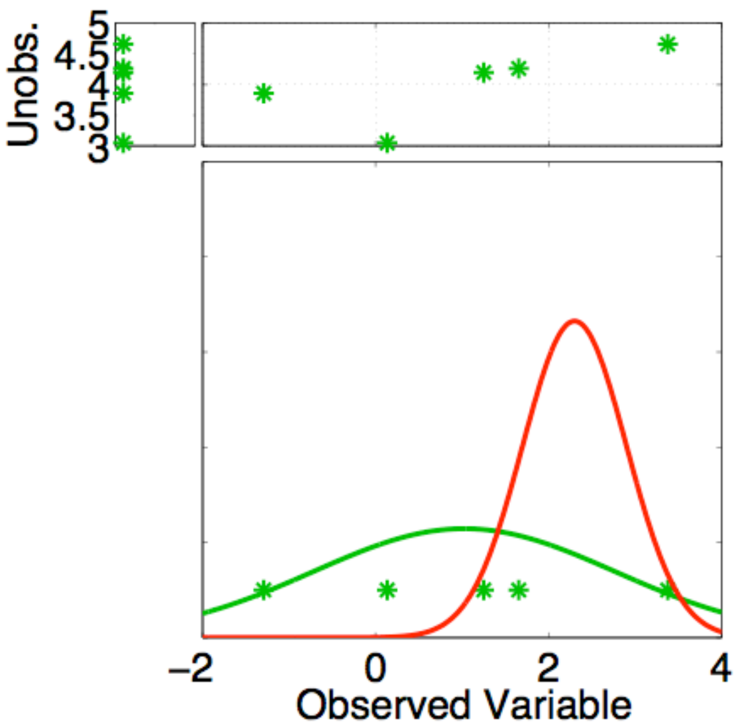


Assume that all we know is prior joint distribution.

One variable is observed.

Update observed variable as in previous section.

Ensemble filters: Updating additional prior state variables

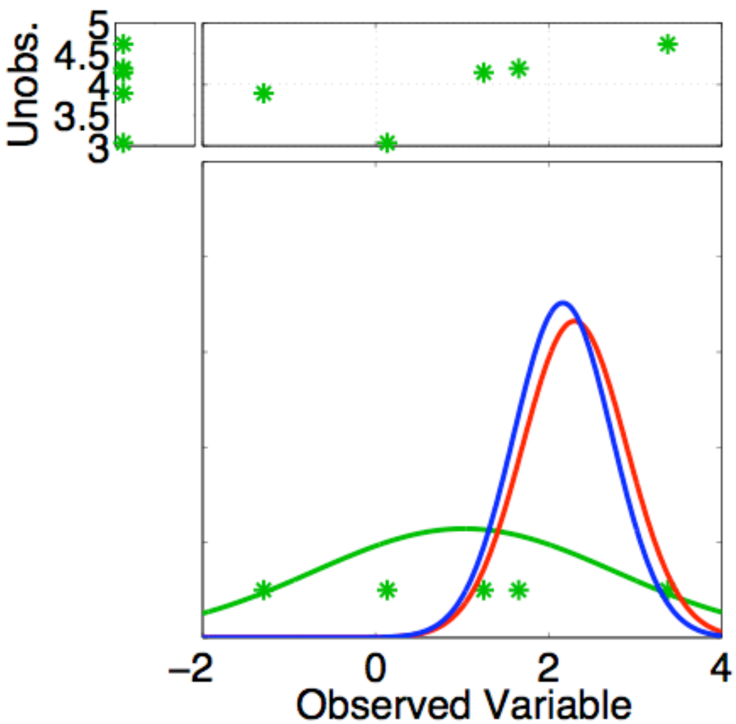


Assume that all we know is prior joint distribution.

One variable is observed.

Update observed variable as in previous section.

Ensemble filters: Updating additional prior state variables

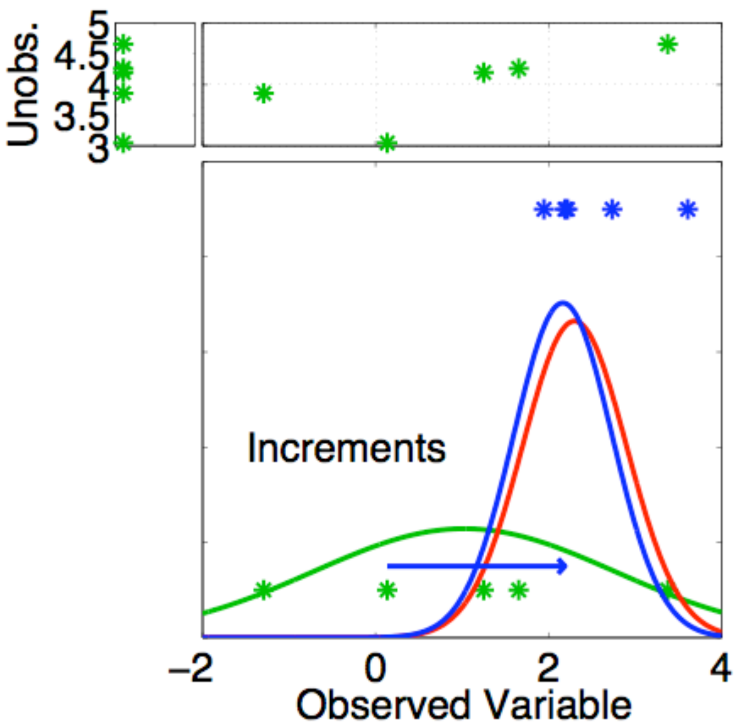


Assume that all we know is prior joint distribution.

One variable is observed.

Update observed variable as in previous section.

Ensemble filters: Updating additional prior state variables

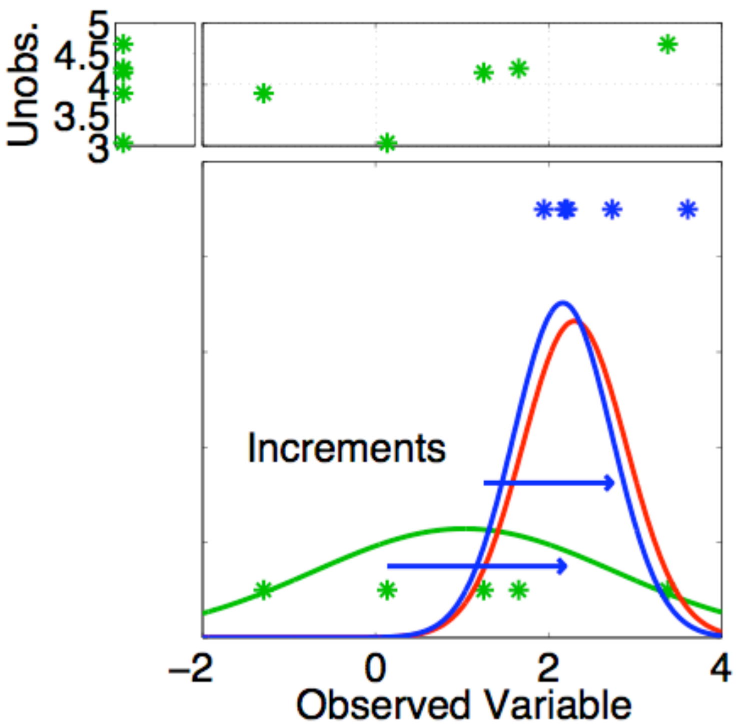


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

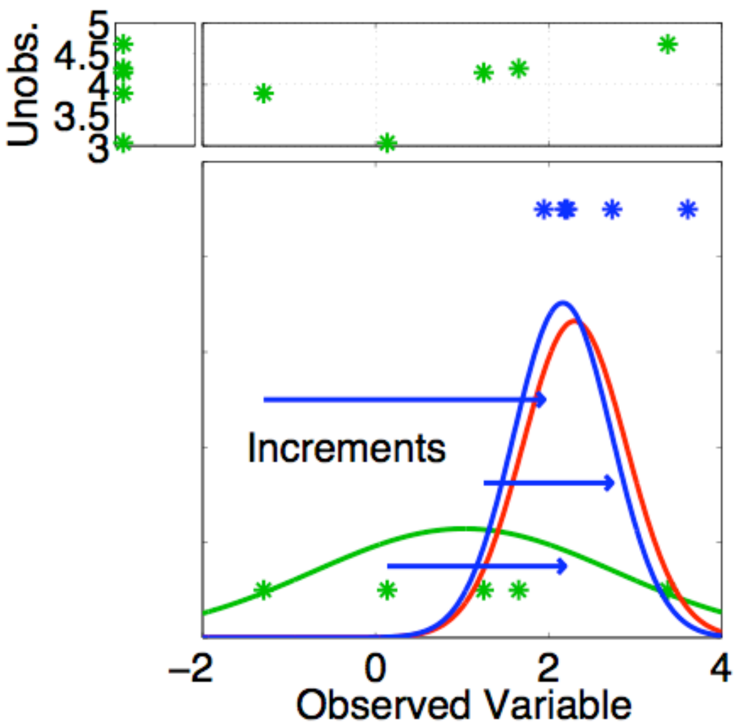


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

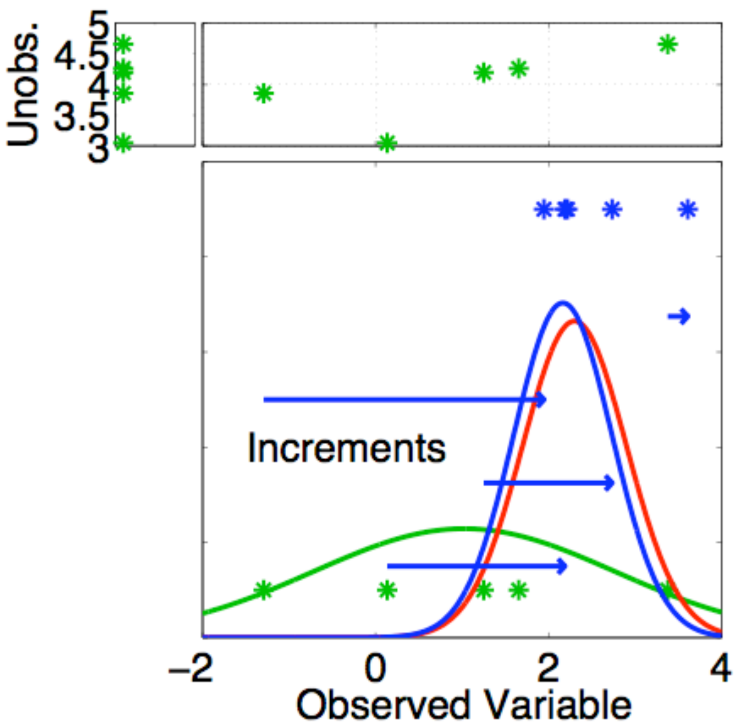


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

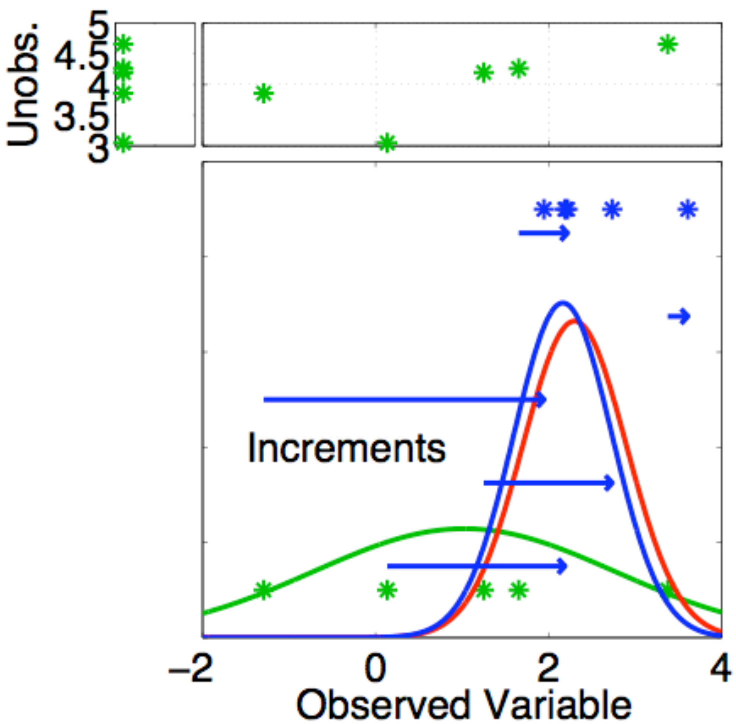


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

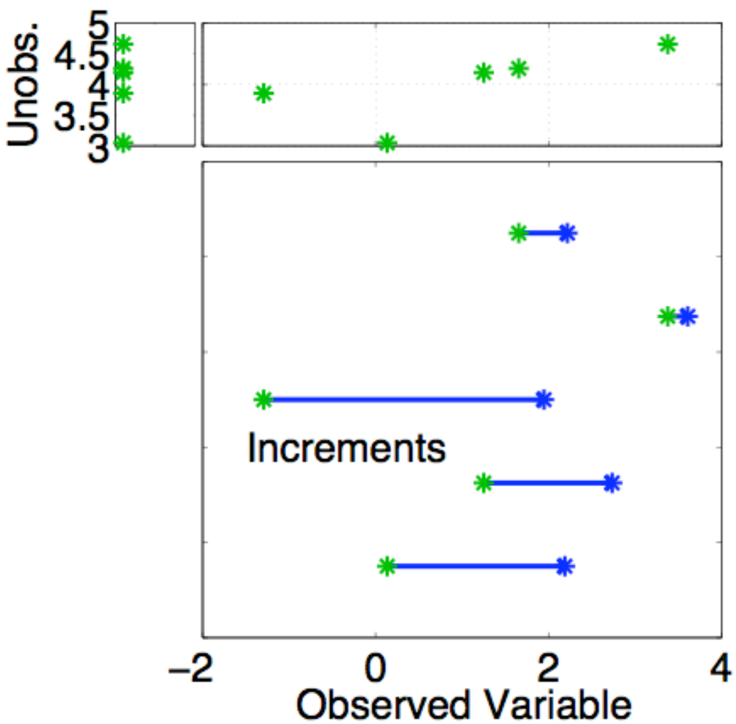


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

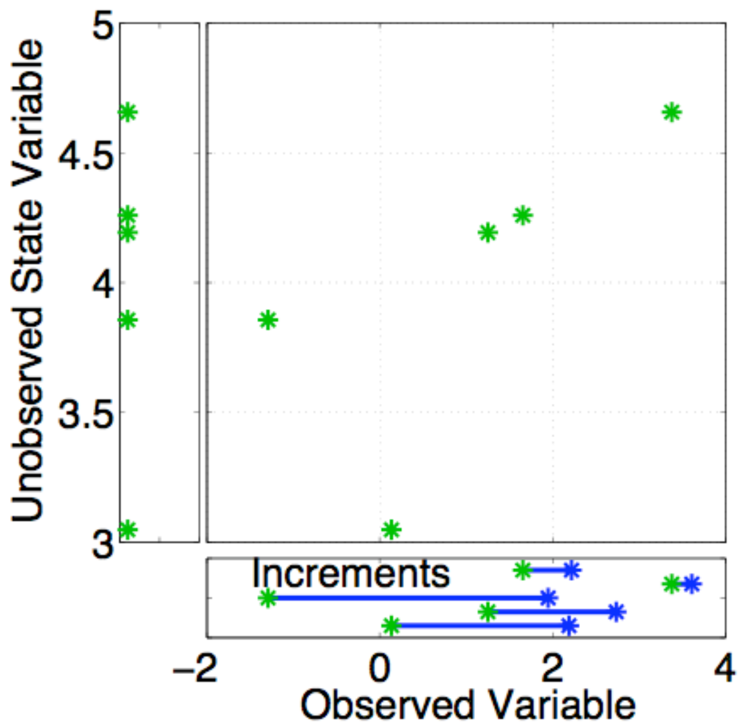


Assume that all we know is prior joint distribution.

One variable is observed.

Using only increments guarantees that if observation had no impact on observed variable, unobserved variable is unchanged (highly desirable).

Ensemble filters: Updating additional prior state variables



Assume that all we know is prior joint distribution.

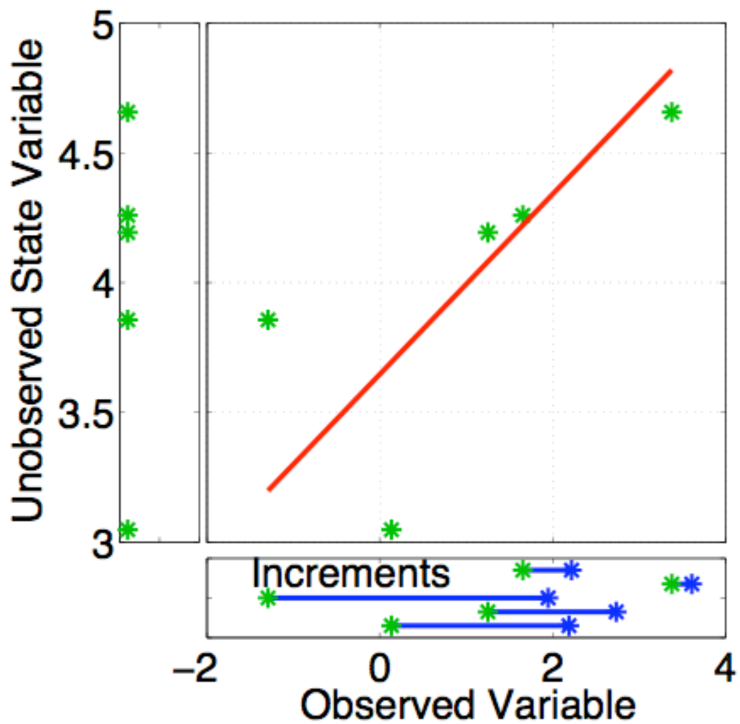
How should the unobserved variable be impacted?

First choice: least squares.

Equivalent to linear regression.

Same as assuming binormal prior.

Ensemble filters: Updating additional prior state variables



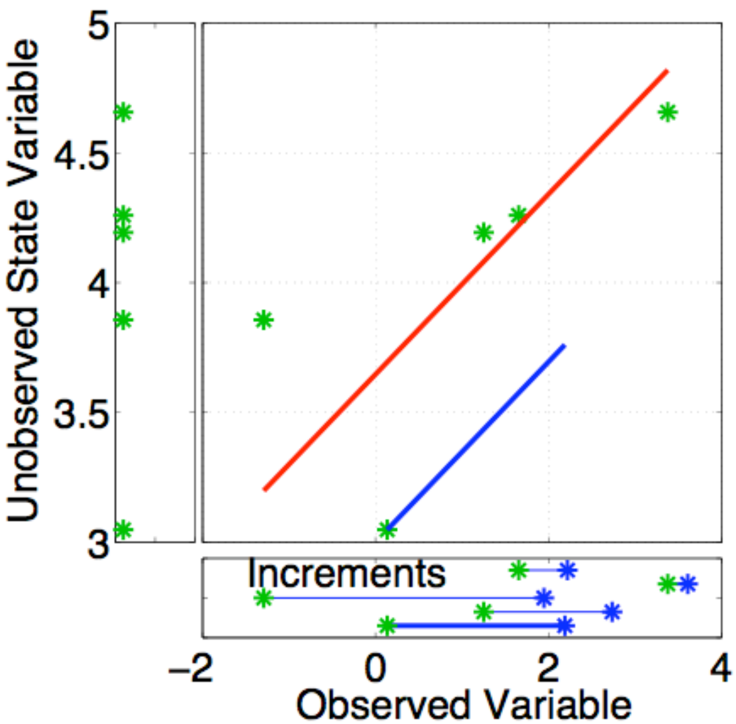
Have joint prior distribution of two variables.

How should the unobserved variable be impacted?

First choice: least squares.

Begin by finding least squares fit.

Ensemble filters: Updating additional prior state variables

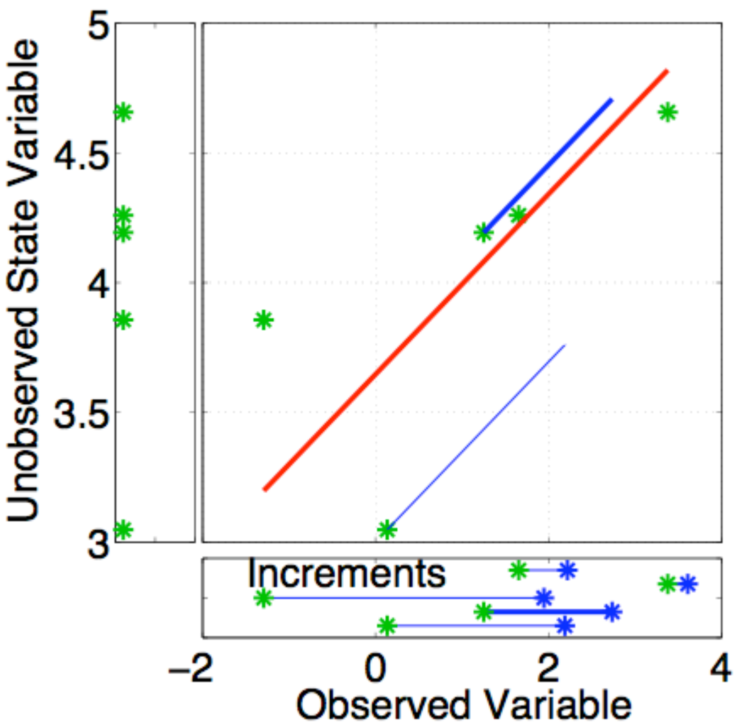


Have joint prior distribution of two variables.

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

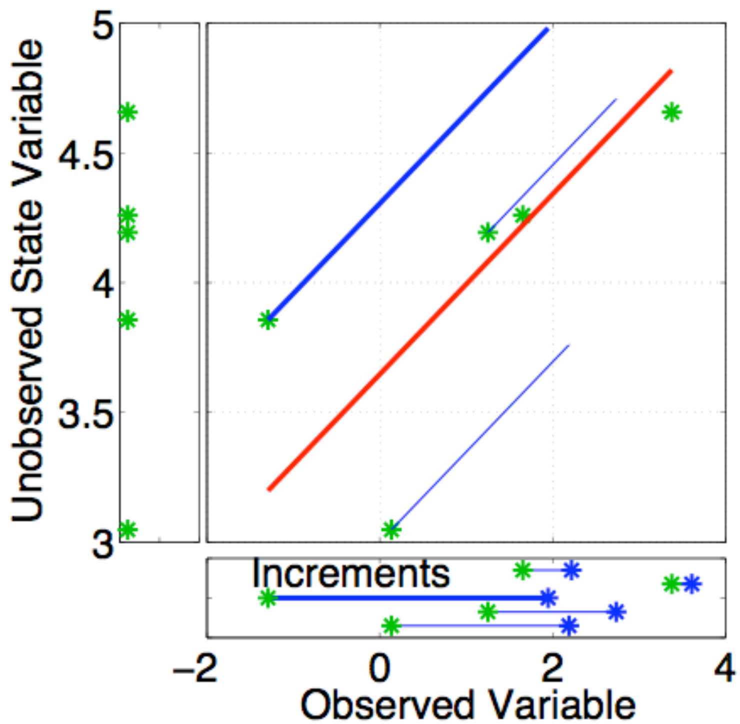


Have joint prior distribution of two variables.

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

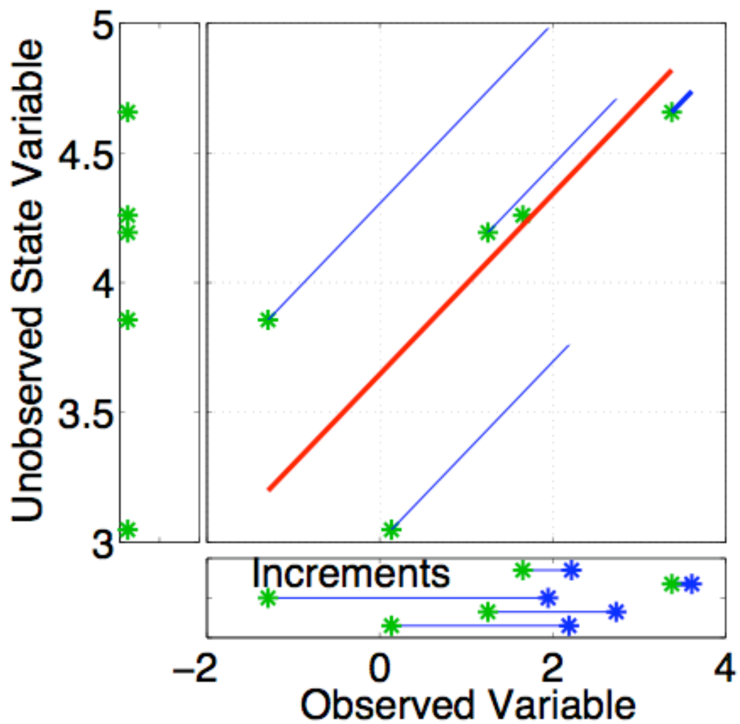


Have joint prior distribution of two variables.

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

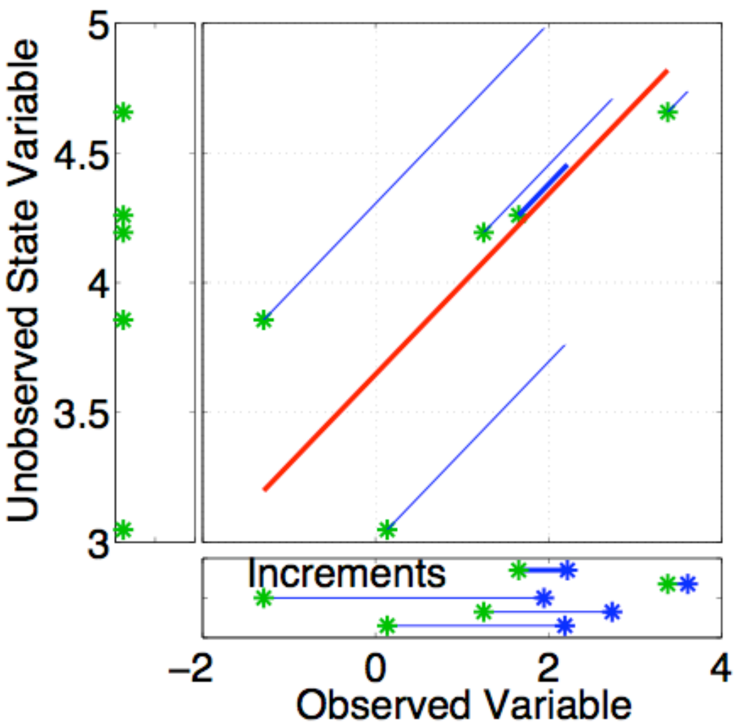


Have joint prior distribution of two variables.

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

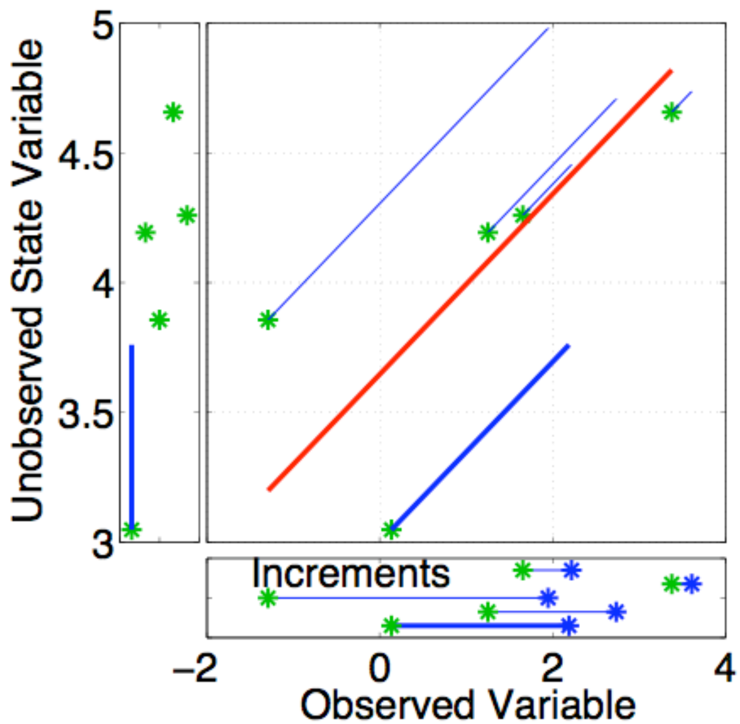


Have joint prior distribution of two variables.

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

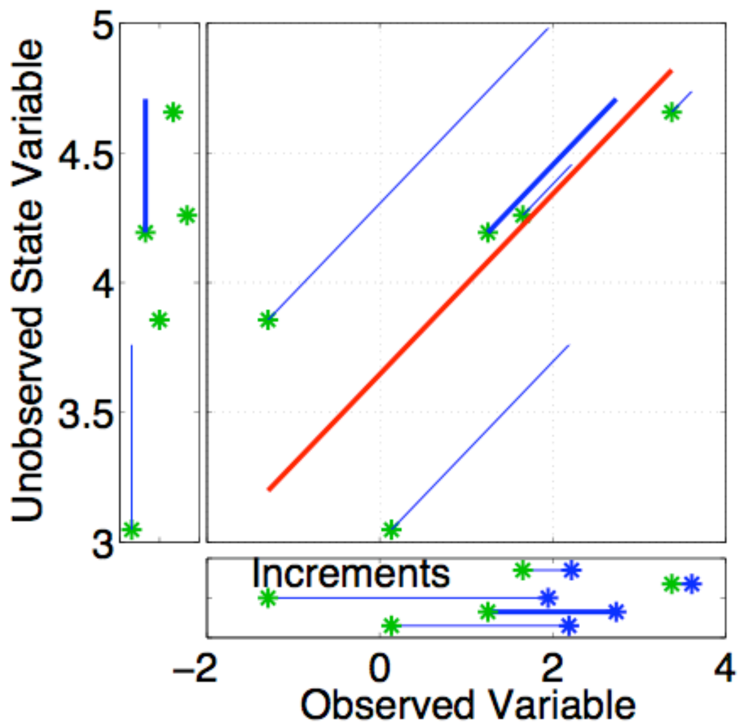


Have joint prior distribution of two variables.

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

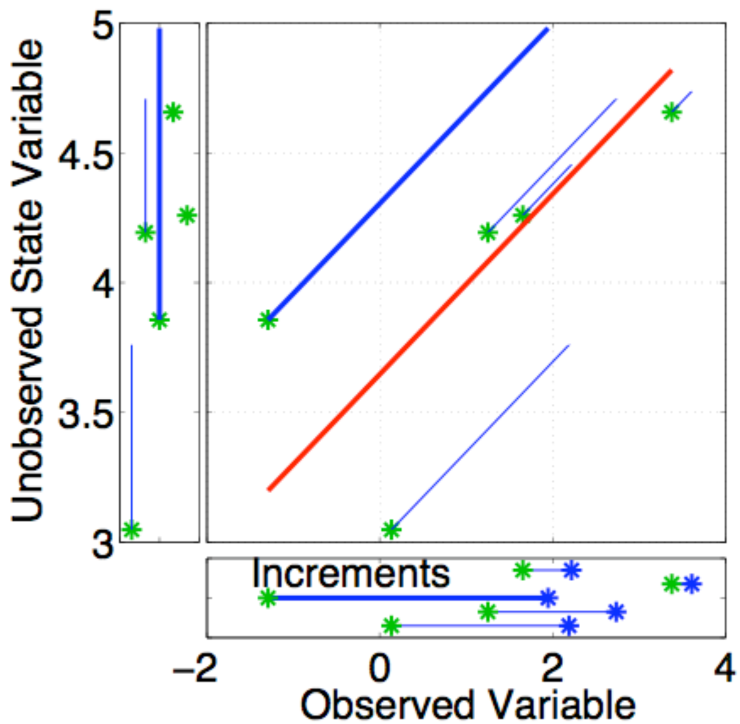


Have joint prior distribution of two variables.

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

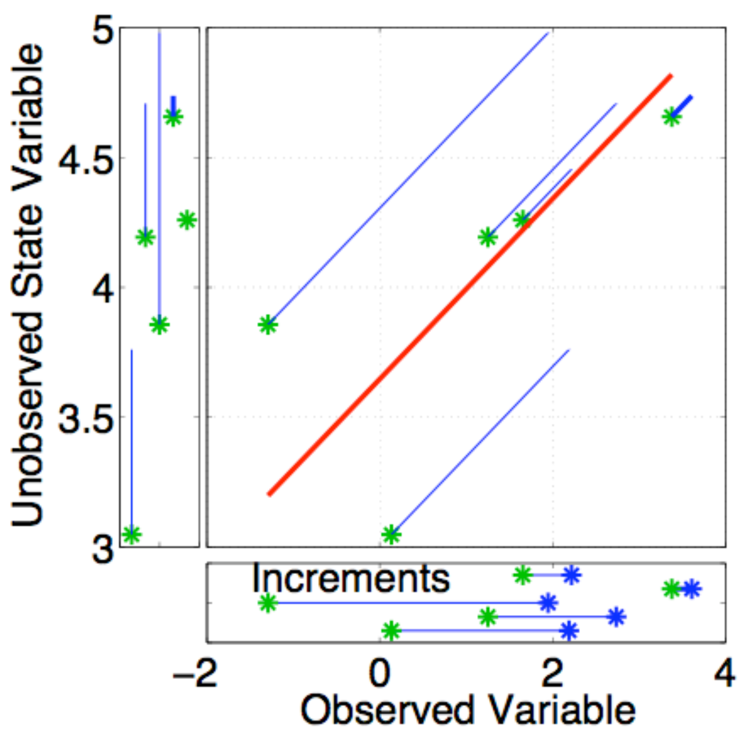


Have joint prior distribution of two variables.

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

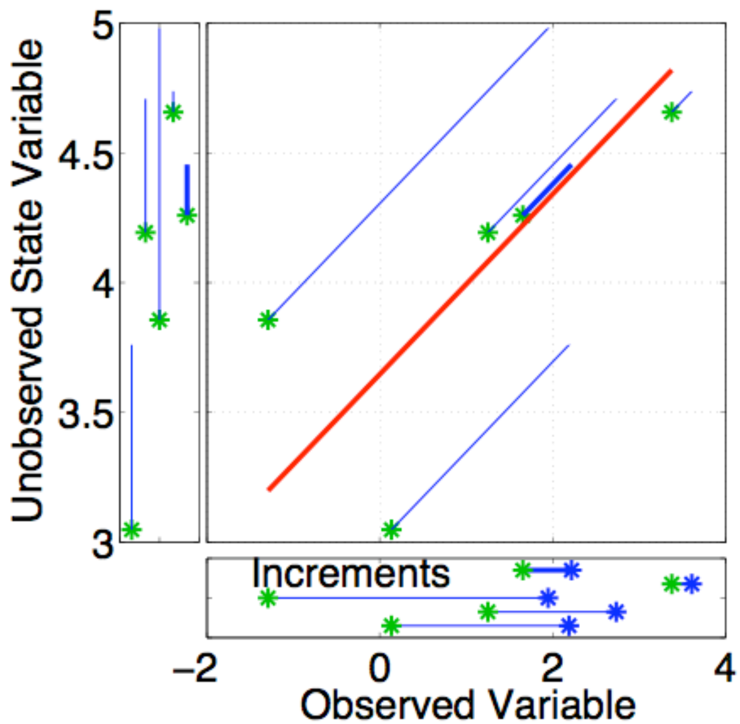


Have joint prior distribution of two variables.

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

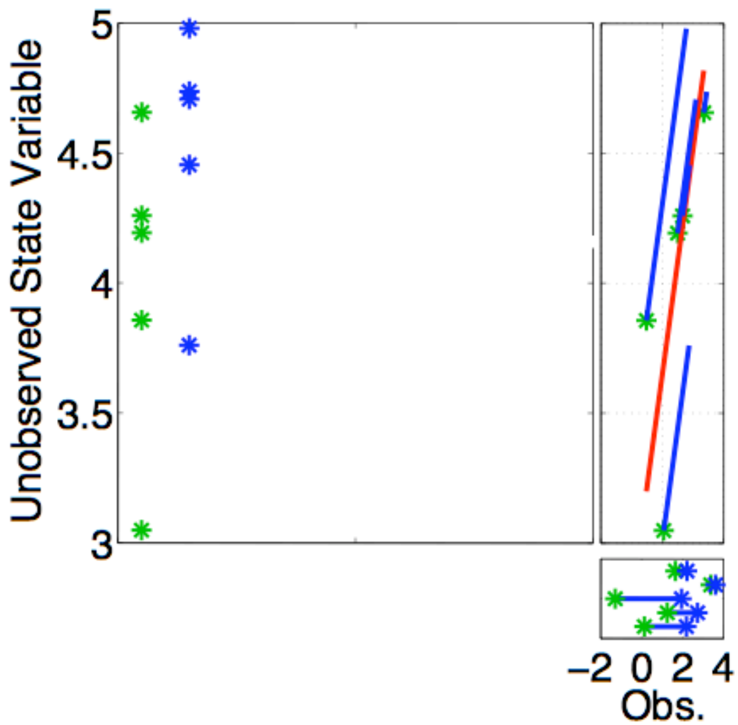


Have joint prior distribution of two variables.

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

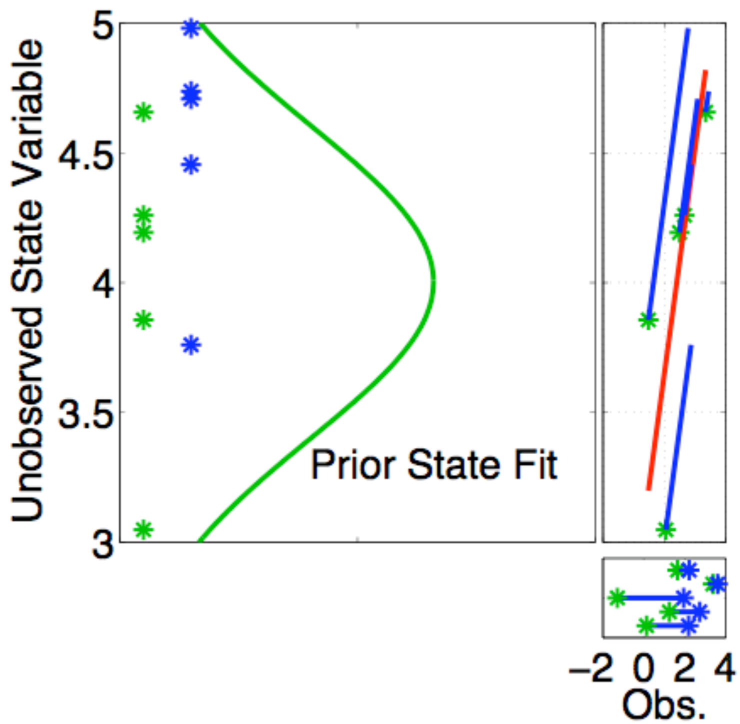
Then projecting from joint space onto unobserved priors.

Ensemble filters: Updating additional prior state variables



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

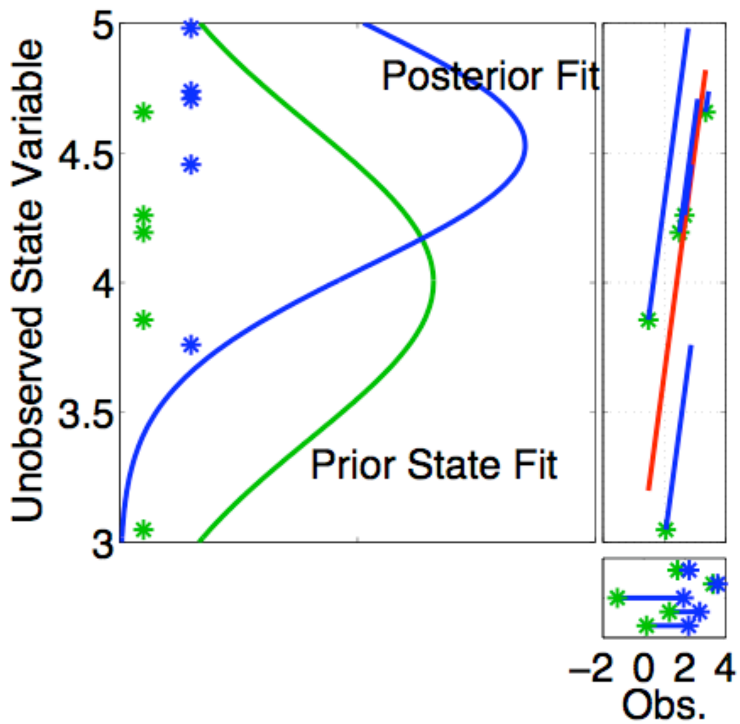
Ensemble filters: Updating additional prior state variables



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.

Ensemble Filter for Large Geophysical Models

1. Use model to advance ensemble (3 members here) to time at which next observation becomes available.

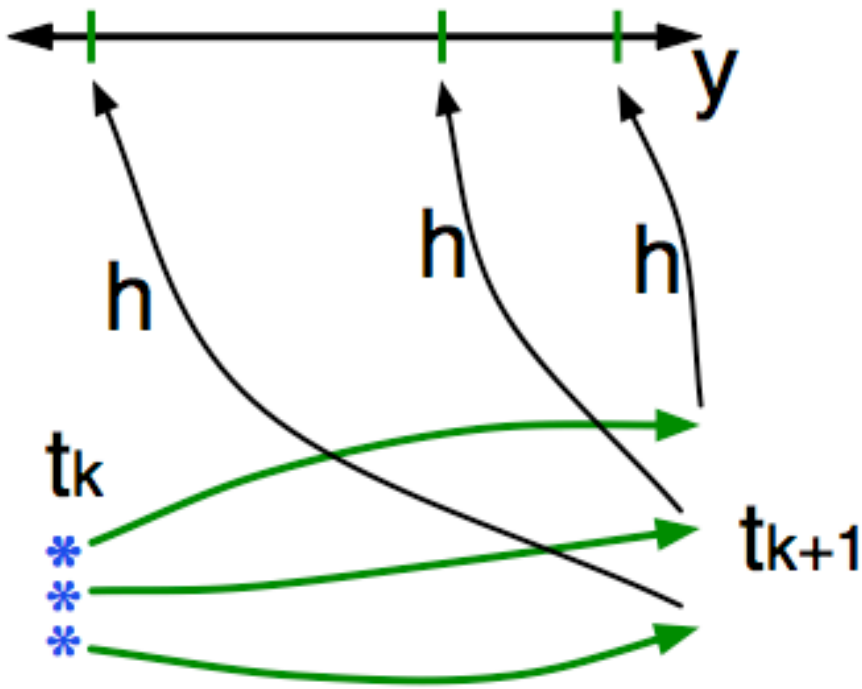
Ensemble state estimate after using previous observation (analysis)

Ensemble state at time of next observation (prior)

t_{k+1}

Ensemble Filter for Large Geophysical Models

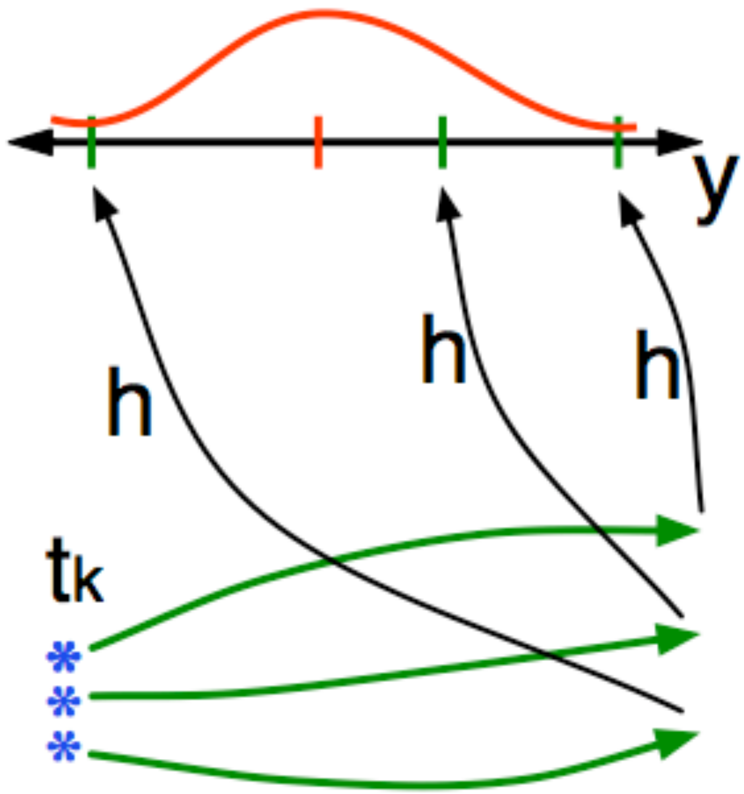
2. Get prior ensemble sample of observation, $y = h(x)$, by applying forward operator **h** to each ensemble member.



Theory: observations from instruments with uncorrelated errors can be done sequentially.

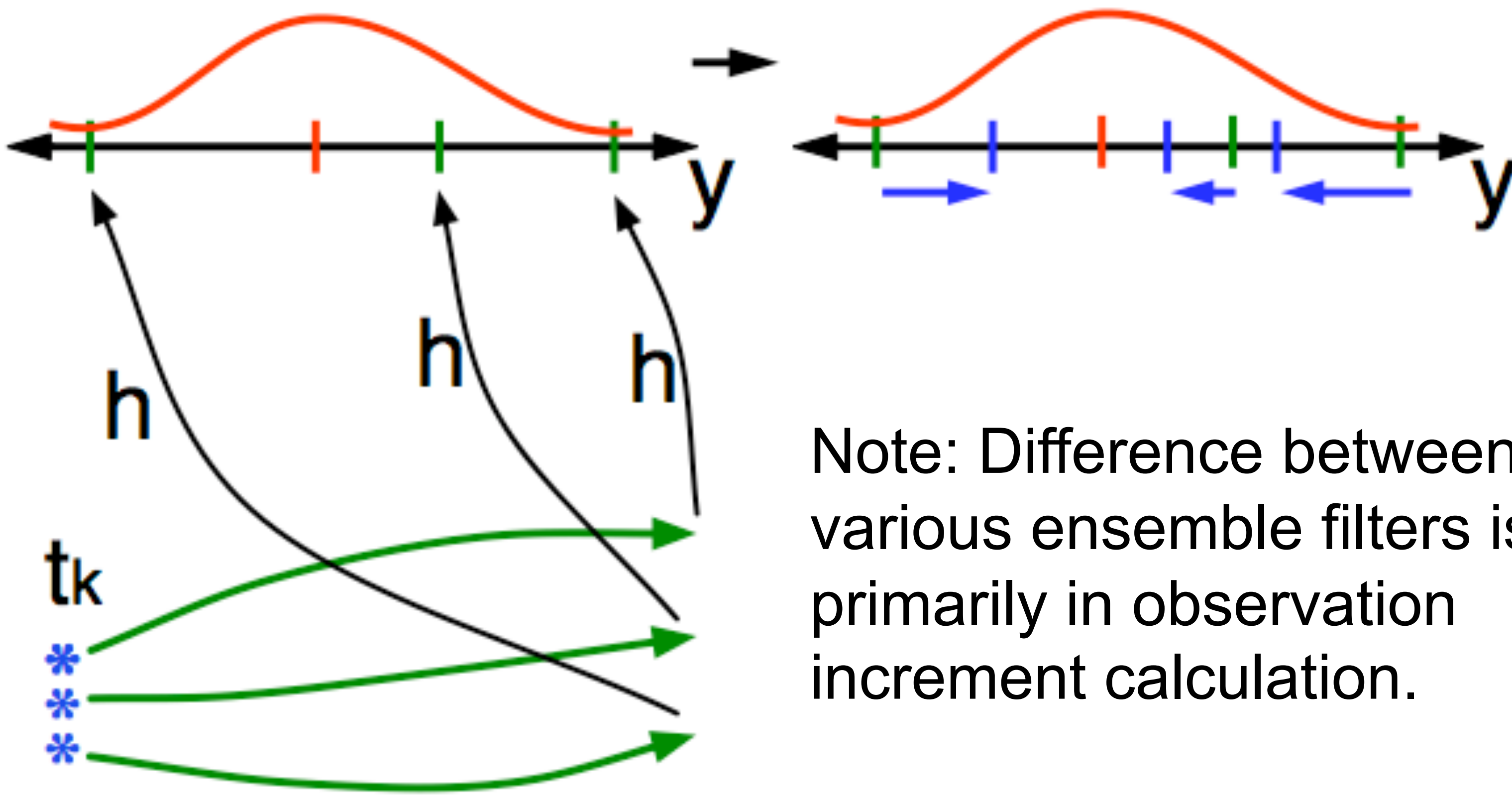
Ensemble Filter for Large Geophysical Models

3. Get observed value and observational error distribution from observing system.



Ensemble Filter for Large Geophysical Models

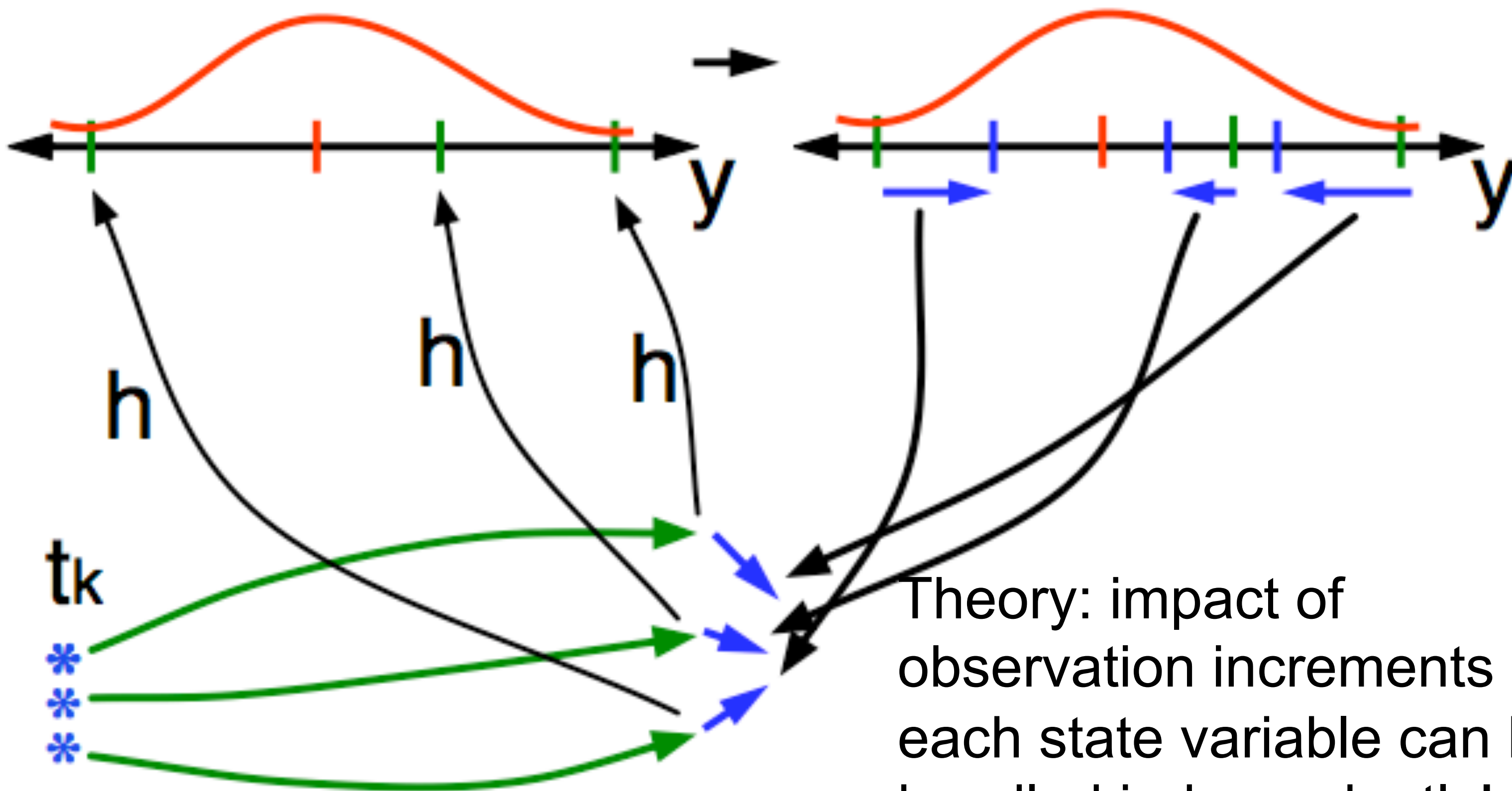
4. Find the increments for the prior observation ensemble (this is a scalar problem for uncorrelated observation errors).



Note: Difference between various ensemble filters is primarily in observation increment calculation.

Ensemble Filter for Large Geophysical Models

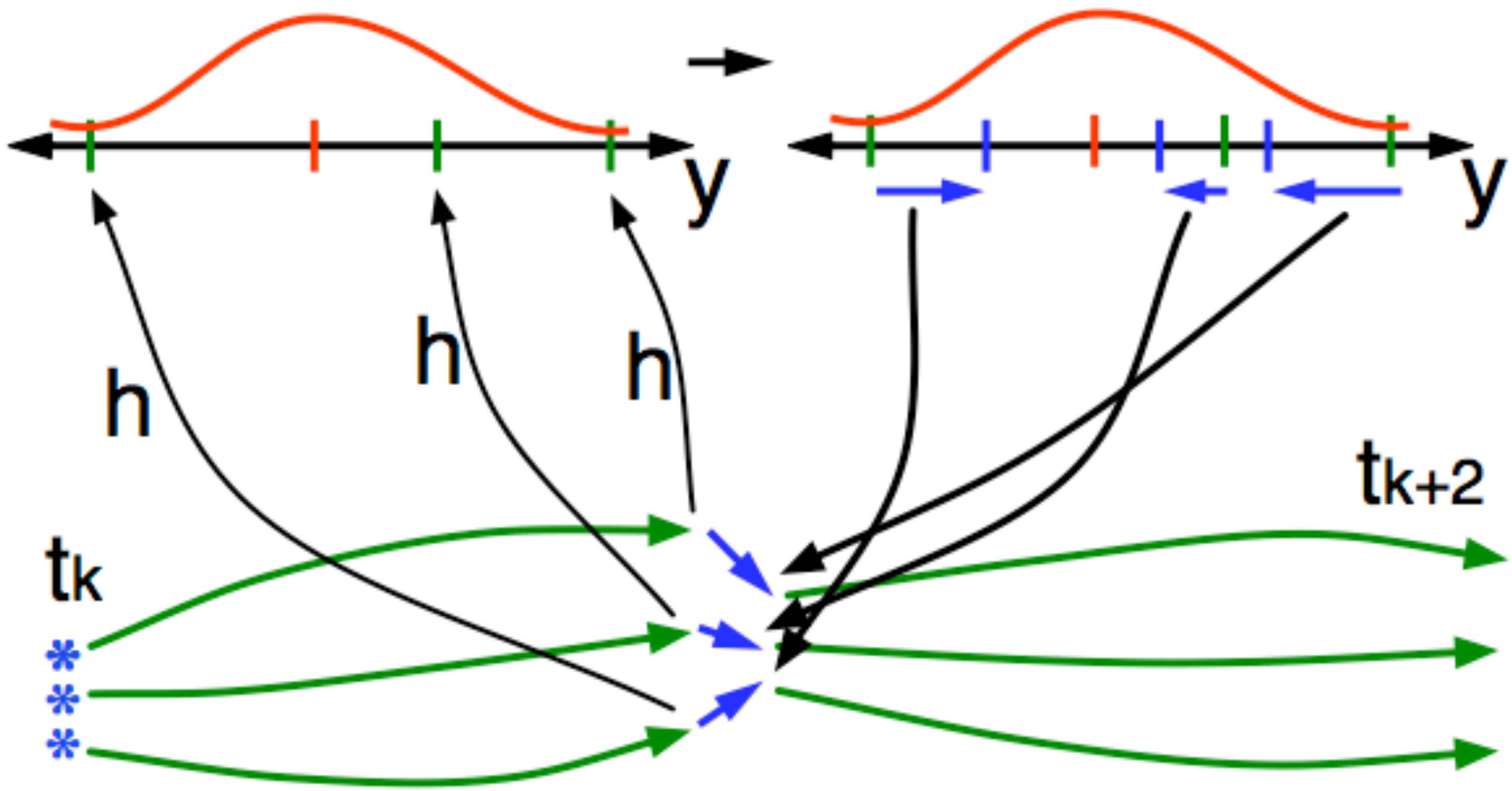
5. Use ensemble samples of **y** and each state variable to linearly regress observation increments onto state variable increments.



Theory: impact of observation increments on each state variable can be handled independently!

Ensemble Filter for Large Geophysical Models

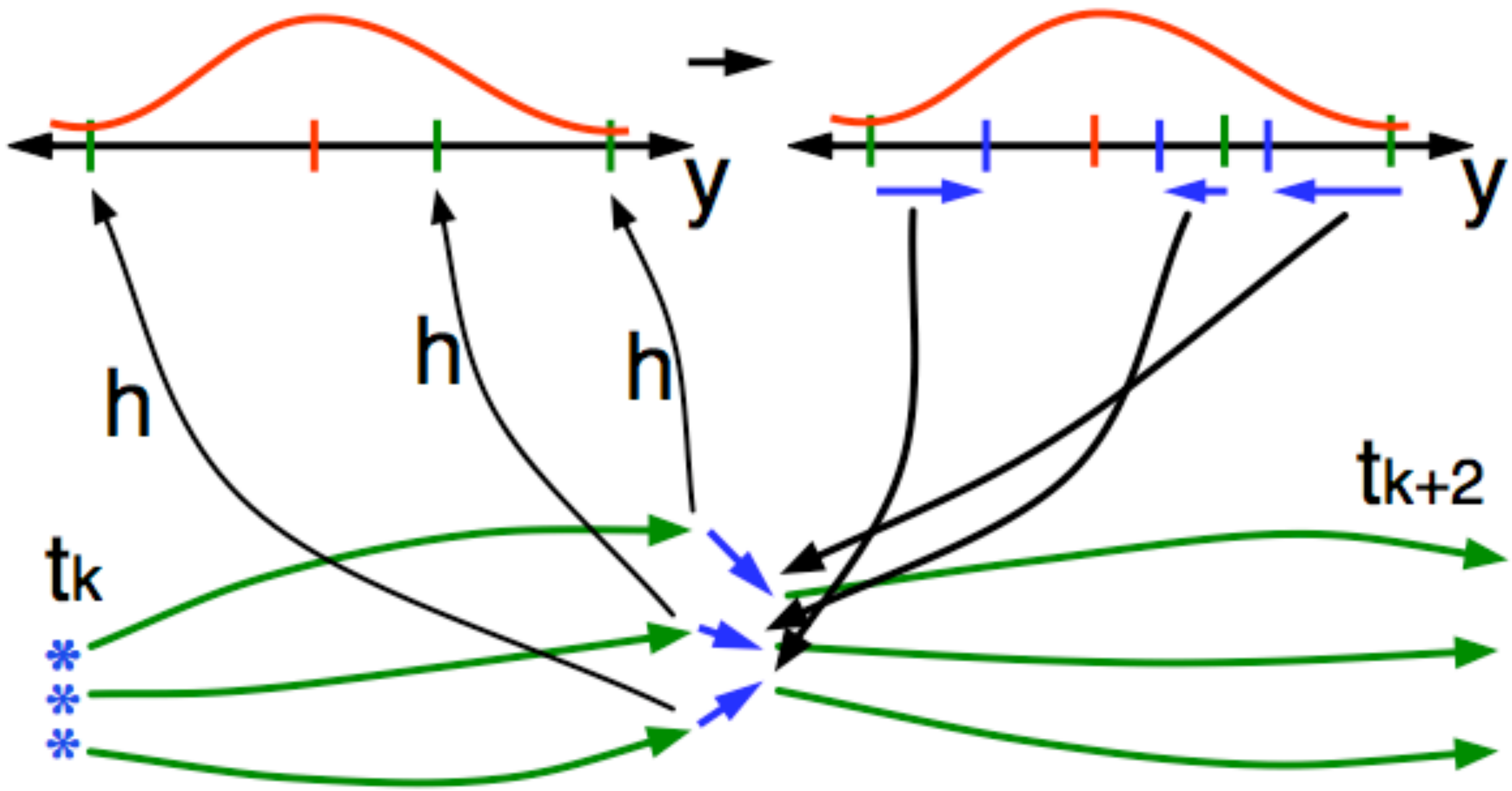
6. When all ensemble members for each state variable are updated, there is a new analysis. Integrate to time of next observation …



Ensemble Filter for Large Geophysical Models

A generic ensemble filter system like DART just needs:

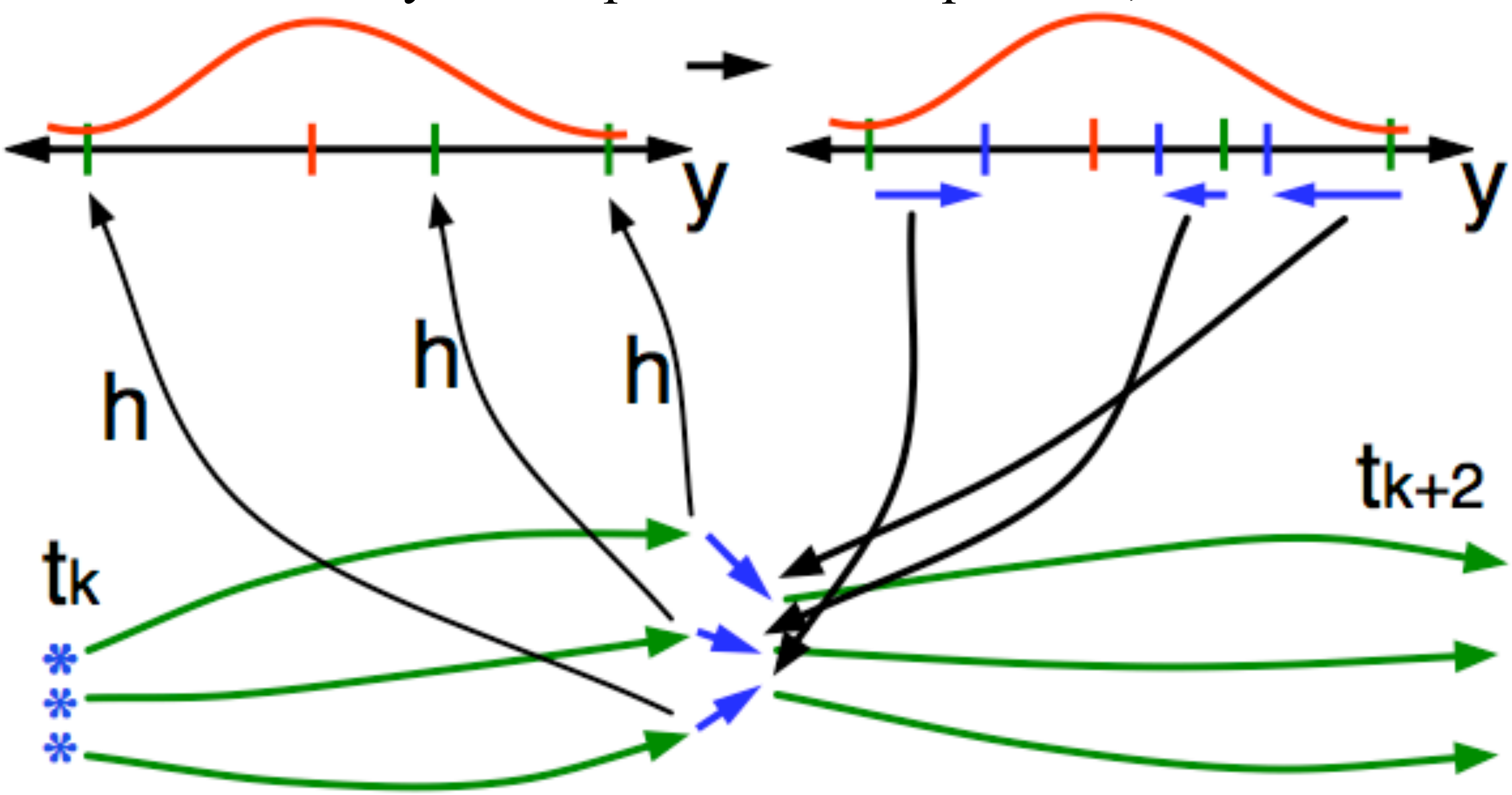
1. A way to make model forecasts;



Ensemble Filter for Large Geophysical Models

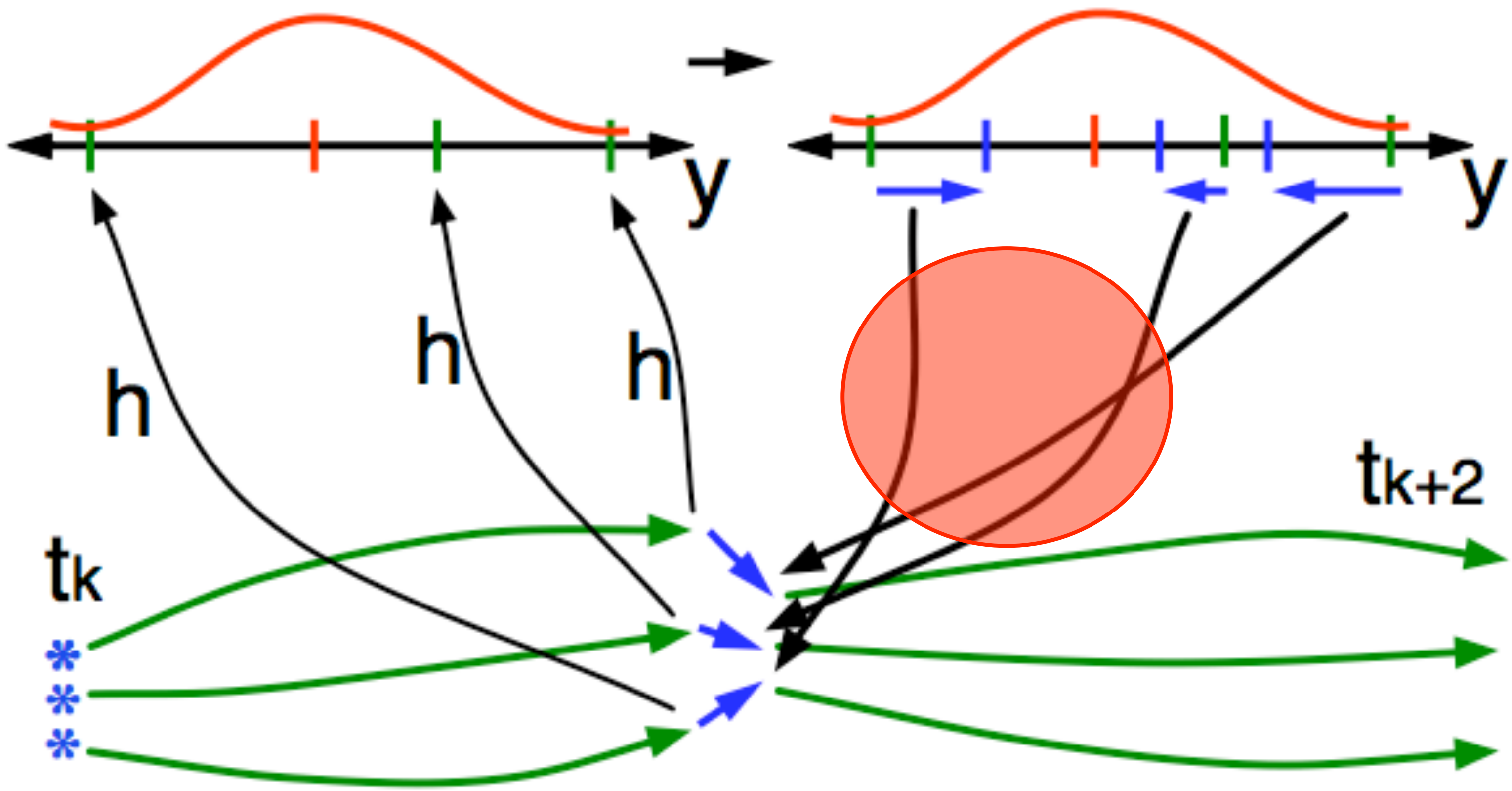
A generic ensemble filter system like DART just needs:

1. A way to make model forecasts;
2. A way to compute forward operators, h.



Parallel Implementation of Sequential Filter

For large models, regression of increments onto each state variable dominates time.



Parallel Implementation of Sequential Filter

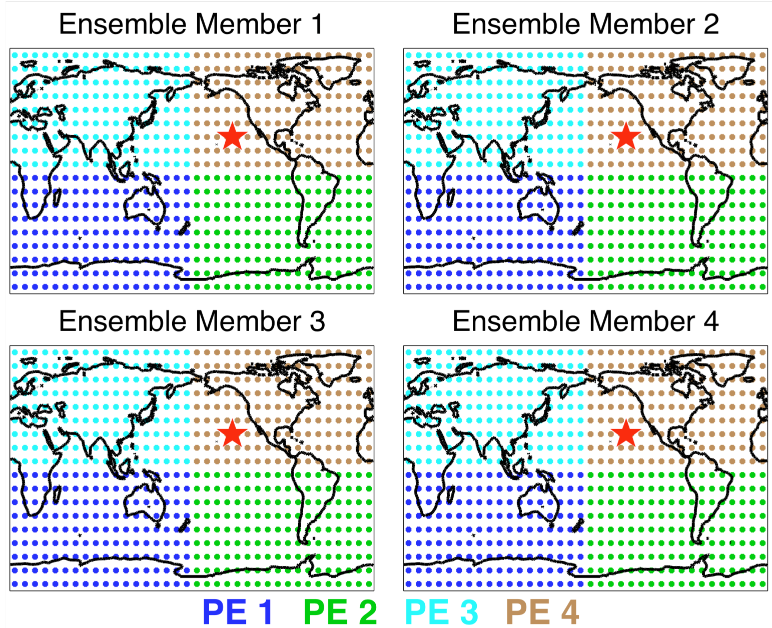
For large models, regression of increments onto each state variable dominates time.

Simple example:

4 Ensemble members;

4 PEs (colors).

Observation shown by red star.



Parallel Implementation of Sequential Filter

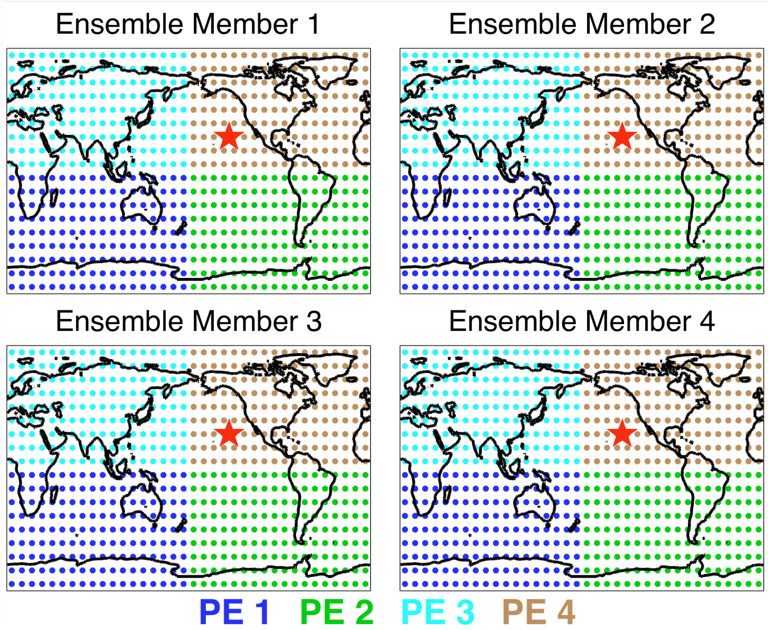
For large models, regression of increments onto each state variable dominates time.

One PE broadcasts obs. increments.

All ensemble members for each state variable are on one PE.

Can compute mean, variance without communication.

All state increments computed in parallel.

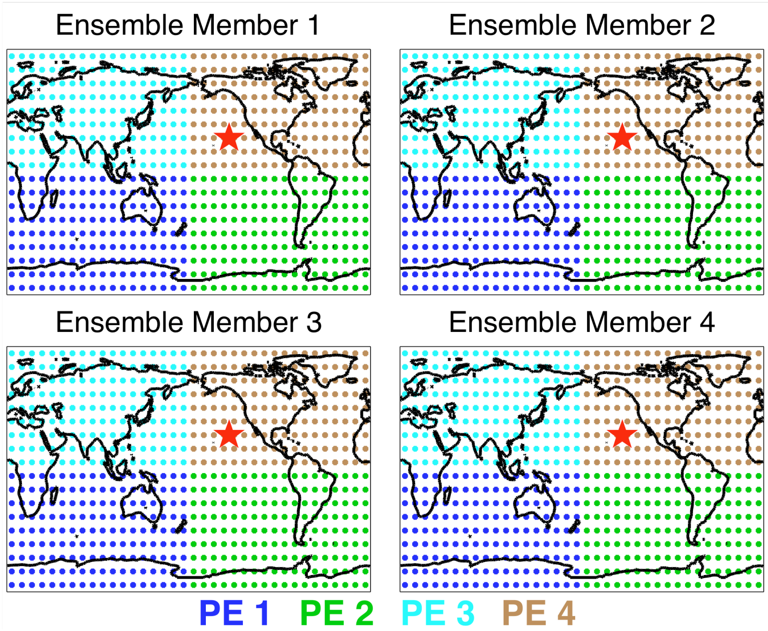


Parallel Implementation of Sequential Filter

For large models, regression of increments onto each state variable dominates time.

Computing forward operator, h, is usually local interpolation.

Most obs. require no communication.

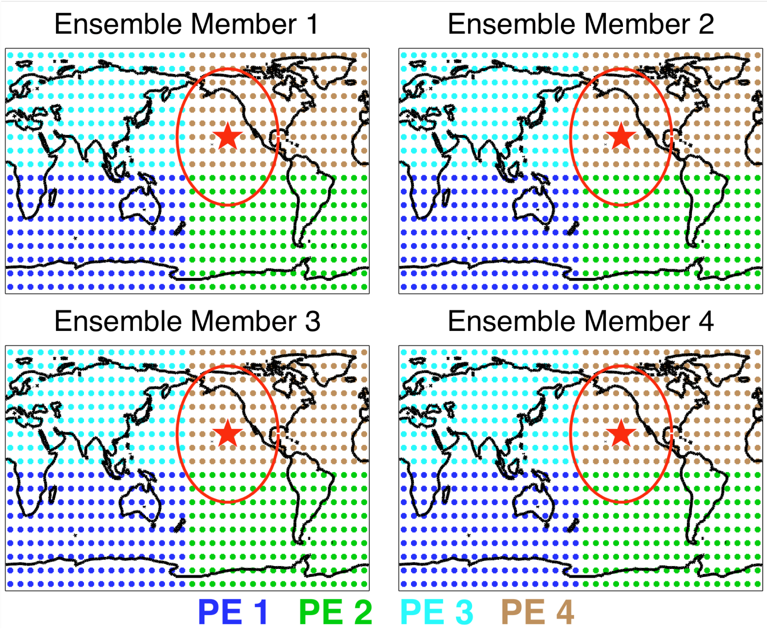


Parallel Implementation of Sequential Filter

For large models, regression of increments onto each state variable dominates time.

Observation impact usually localized.
- ➢ Reduces sampling error.
- ➢ Observation in N. Pacific not expected to change Antarctic state.

Now have a load balancing problem.

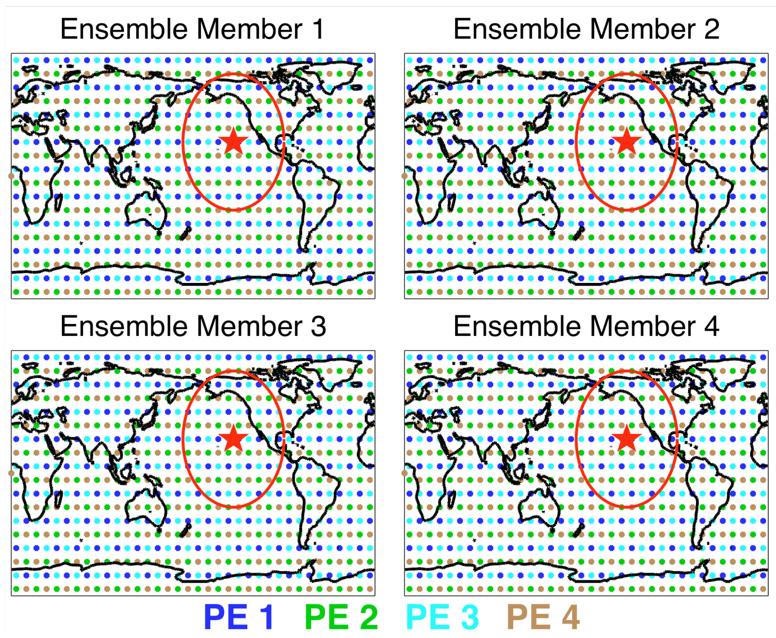


Parallel Implementation of Sequential Filter

For large models, regression of increments onto each state variable dominates time.

Can balance load by ‘randomly’ assigning state ensembles to PEs.

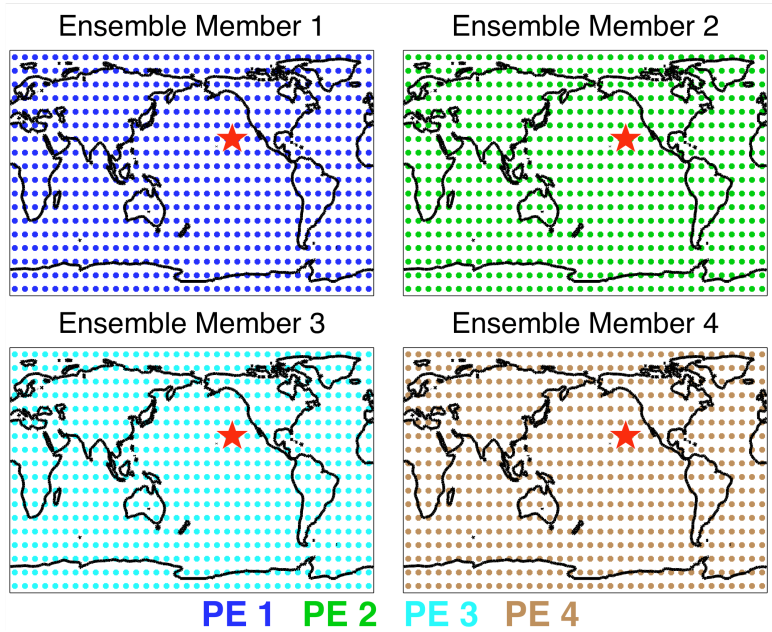
Now computing forward operators, h, requires communication.



Parallel Implementation of Sequential Filter

For large models, regression of increments onto each state variable dominates time.

If each PE has a complete ensemble, forward operators require no communication.

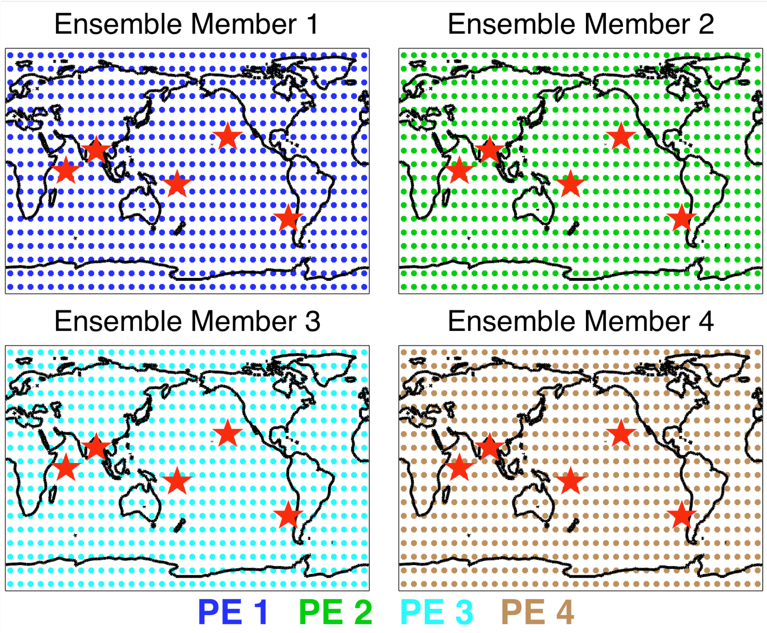


Parallel Implementation of Sequential Filter

For large models, regression of increments onto each state variable dominates time.

If each PE has a complete ensemble, forward operators require no communication.

Can do many forward operators in parallel.

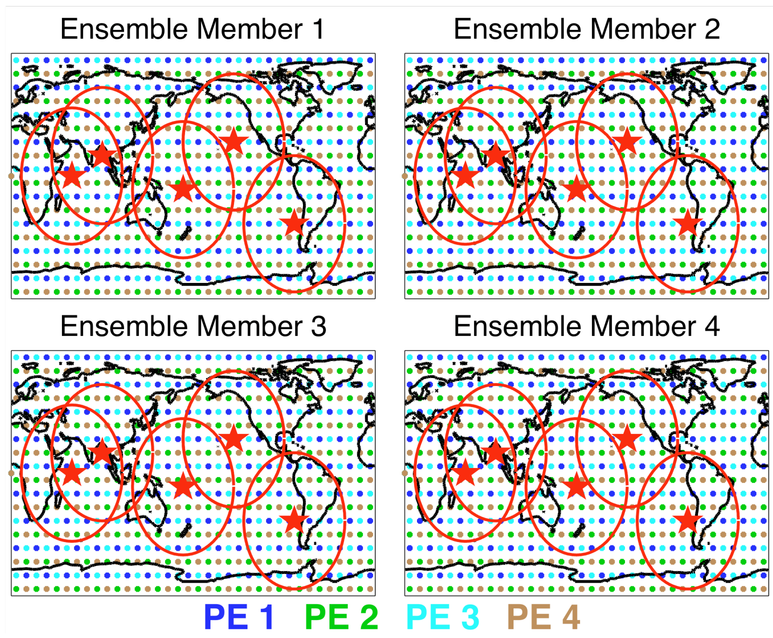


Parallel Implementation of Sequential Filter

For large models, regression of increments onto each state variable dominates time.

Do a data transpose, using all to all communication.

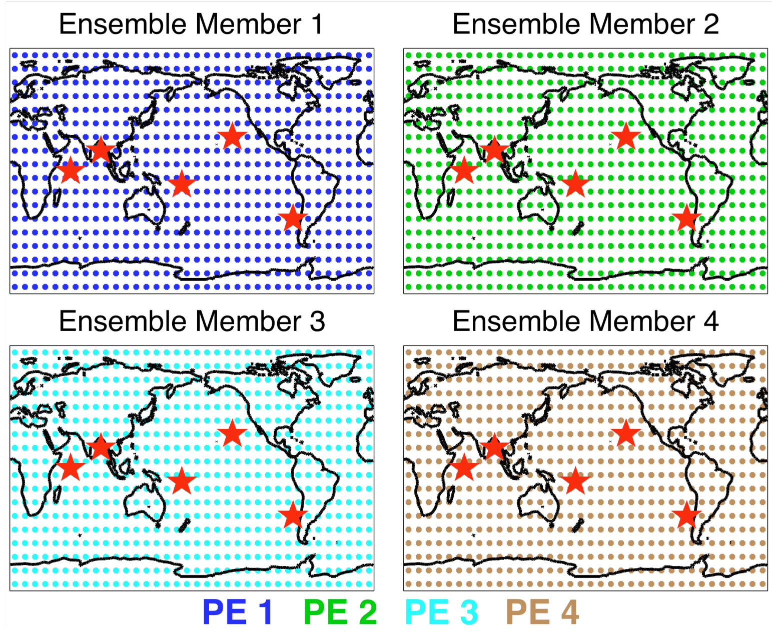
Can do state increments for many obs in parallel for extra cost $O(n^2)$
(n is number of obs)



Parallel Implementation of Sequential Filter

For large models, regression of increments onto each state variable dominates time.

Then transpose back to do more forward operators or advance model.



Parallel Implementation of Sequential Filter

Algorithm can be tuned for problem size, # of PEs;

Number of observations per transpose;
Selection of subsets of obs. to do in parallel;

How to assign state variables to PEs to:

1). Minimize transpose cost;
2). Minimize forward operator cost;
3). Minimize communication for updates.

Really fun for heterogeneous communication paths!

Parallel Implementation of Sequential Filter

Scaling for large atmospheric models:

Naïve random algorithm scales to O(100) PEs for mid-size climate / regional prediction models.

Expect modern NWP model to scale to O(1000).

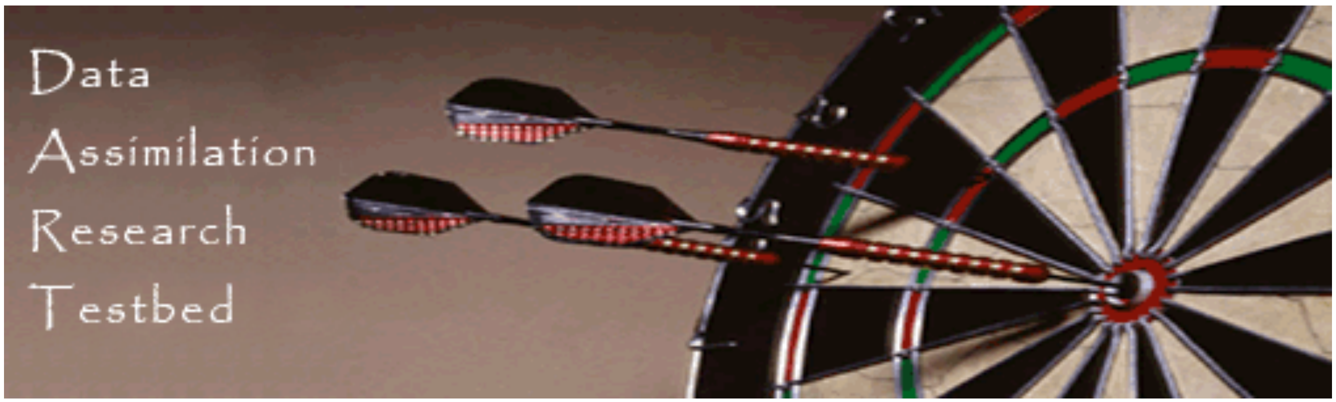
O(10,000) seems viable with custom algorithm design.

Conclusions

1. Ensemble filters are trivial to implement for arbitrary models and observations. We've done atmospheric and ocean GCMs in less than a person month.

2. Sequential ensemble filter algorithms promise to scale well to O(1,000), and probably O(10,000) PEs with limited model specific tuning.

Code to implement all of the parallel filter algorithms discussed are freely available from:



http://www.image.ucar.edu/DAReS/DART/