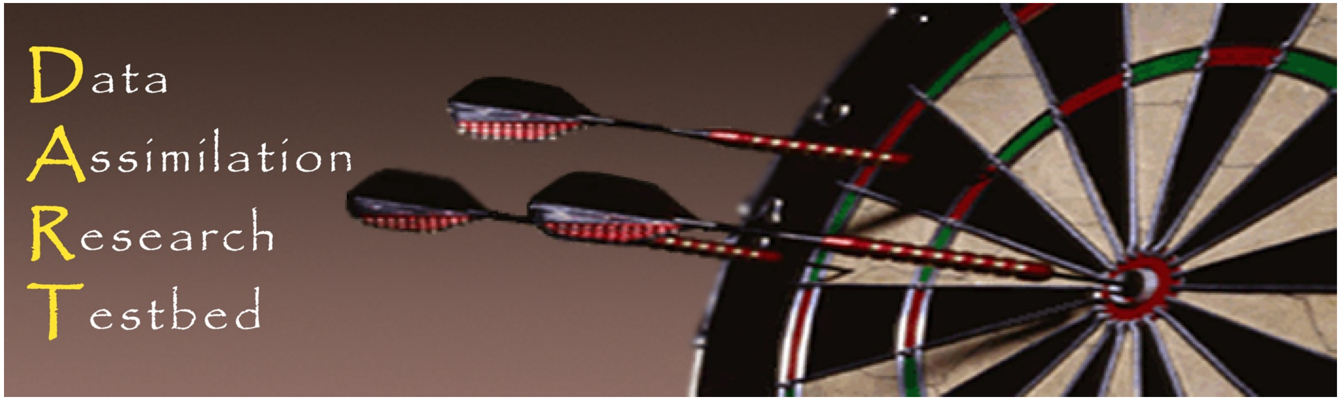
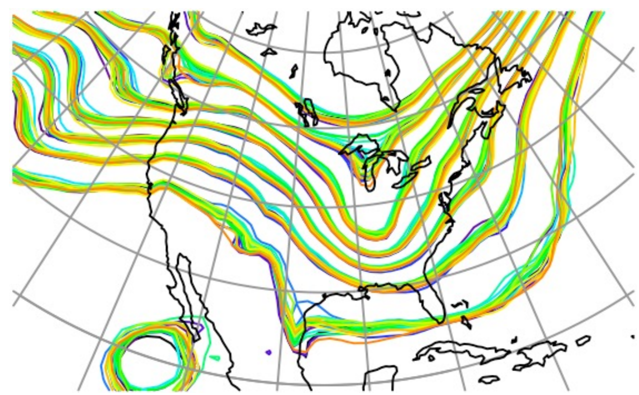
Jeff Anderson is part of the Data Assimilation Research Section,
Located at the National Center for Atmospheric Research in Boulder, Colorado.
It's an awfully nice place to visit, and we love hosting visitors for collaboration!
Someday the world will be back to normal.



Removing the Kalman from the Ensemble Kalman Filter

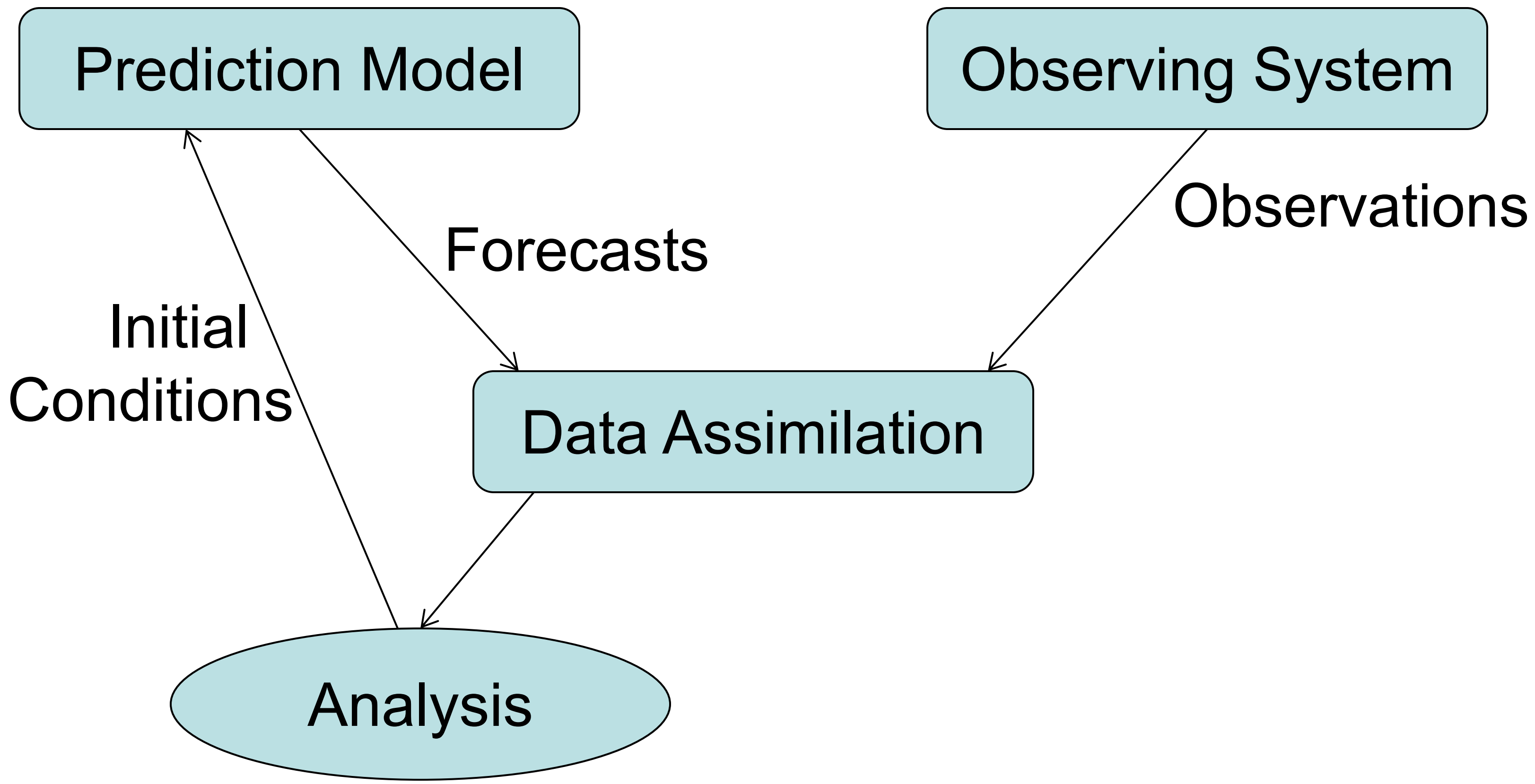
Jeff Anderson, NCAR/DAReS



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.

NCAR | National Center for
UCAR | Atmospheric Research

Building a Forecast System



A General Description of the Forecast Problem

A system governed by (stochastic) Difference Equation:

$$dx_t = f(x_t, t) + G(x_t, t)\, d\beta_t, \qquad t \ge 0 \tag{1}$$

Observations at discrete times:

$$y_k = h(x_k, t_k) + v_k; \qquad k = 1, 2, \ldots; \qquad t_{k+1} > t_k \ge t_0 \tag{2}$$

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

$$v_k \to N(0, R_k) \tag{3}$$

Complete history of observations is:

$$Y_\tau = \{ y_l; t_l \le \tau \} \tag{4}$$

Goal: Find probability distribution for state:

$$p(x, t \mid Y_t) \quad \text{Analysis} \qquad\qquad p(x, t^+ \mid Y_t) \quad \text{Forecast} \tag{5}$$

A General Description of the Forecast Problem

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

$$p\left(x,t_k \mid Y_{t_k}\right) = p\left(x,t_k \mid y_k, Y_{t_{k-1}}\right) \tag{6}$$

Apply Bayes' rule:

$$p\left(x,t_k \mid Y_{t_k}\right) = \frac{p(y_k \mid x_k, Y_{t_{k-1}})\,p(x,t_k \mid Y_{t_{k-1}})}{p(y_k \mid Y_{t_{k-1}})} \tag{7}$$

Noise is white in time (3), so:

$$p\left(y_k \mid x_k, Y_{t_{k-1}}\right) = p\left(y_k \mid x_k\right) \tag{8}$$

Integrate numerator to get normalizing denominator:

$$p(y_k \mid Y_{t_{k-1}}) = \int p(y_k \mid x)\,p(x,t_k \mid Y_{t_{k-1}})\,dx \tag{9}$$

A General Description of the Forecast Problem

Probability after new observation:

Prior (forecast)

Likelihood

$$p\left(x,t_k \mid Y_{t_k}\right) = \frac{p\left(y_k \mid x\right) p\left(x,t_k \mid Y_{t_{k-1}}\right)}{\int p(y_k \mid \xi) p(\xi,t_k \mid Y_{t_{k-1}}) d\xi} \qquad (10)$$

Posterior (analysis).

Denominator just normalization.

Methods for Solving the Forecast Problem: Kalman Filter

Assumes:

linear model

Gaussian noise

$$dx_t = f(x_t, t) + G(x_t, t)\, d\beta_t, \qquad t \geq 0$$

Gaussian state

linear forward operator,

$$y_k = h(x_k, t_k) + v_k; \qquad k = 1, 2, \ldots; \qquad t_{k+1} > t_k \geq t_0$$

Gaussian observation error

Product of Two Gaussians

Product of d-dimensional normals with means μ_1 and μ_2 and covariance matrices Σ_1 and Σ_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 μ_1 and μ_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 μ_1 and μ_2 and covariance matrices Σ_1 and Σ_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: $$c = \frac{1}{(2\Pi)^{d/2}\left|\Sigma_1 + \Sigma_2\right|^{1/2}} \exp\left\{-\frac{1}{2}\left[(\mu_2 - \mu_1)^T (\Sigma_1 + \Sigma_2)^{-1} (\mu_2 - \mu_1)\right]\right\}$$

We'll ignore the weight since we immediately normalize products to be PDFs.

The Kalman Filter

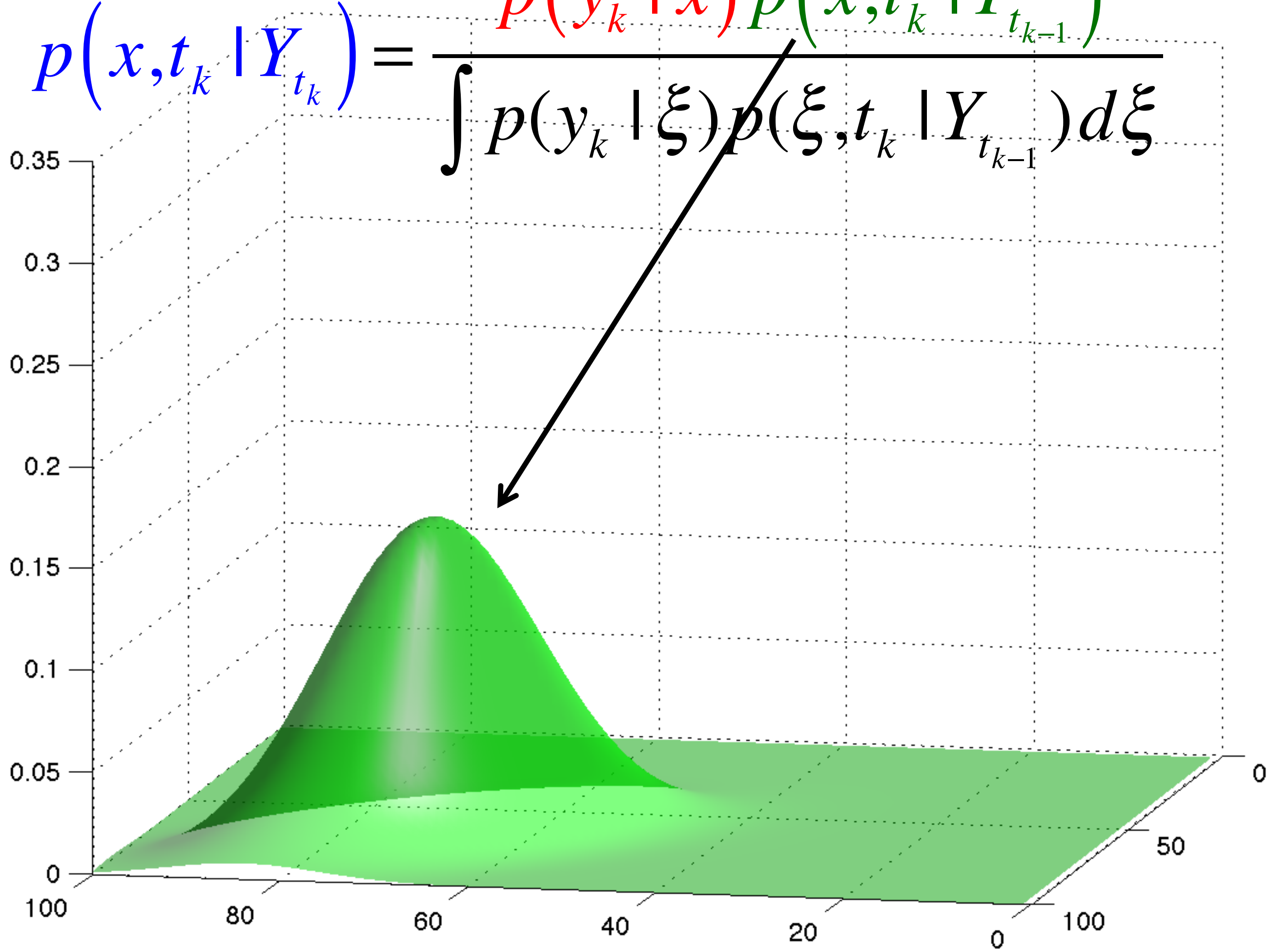
$$p\left(x, t_k \mid Y_{t_k}\right) = \frac{p\left(y_k \mid x\right) p\left(x, t_k \mid Y_{t_{k-1}}\right)}{\int p(y_k \mid \xi) p(\xi, t_k \mid Y_{t_{k-1}}) d\xi} \tag{10}$$

Numerator is just product of two gaussians.

Denominator just normalizes posterior to be a PDF.

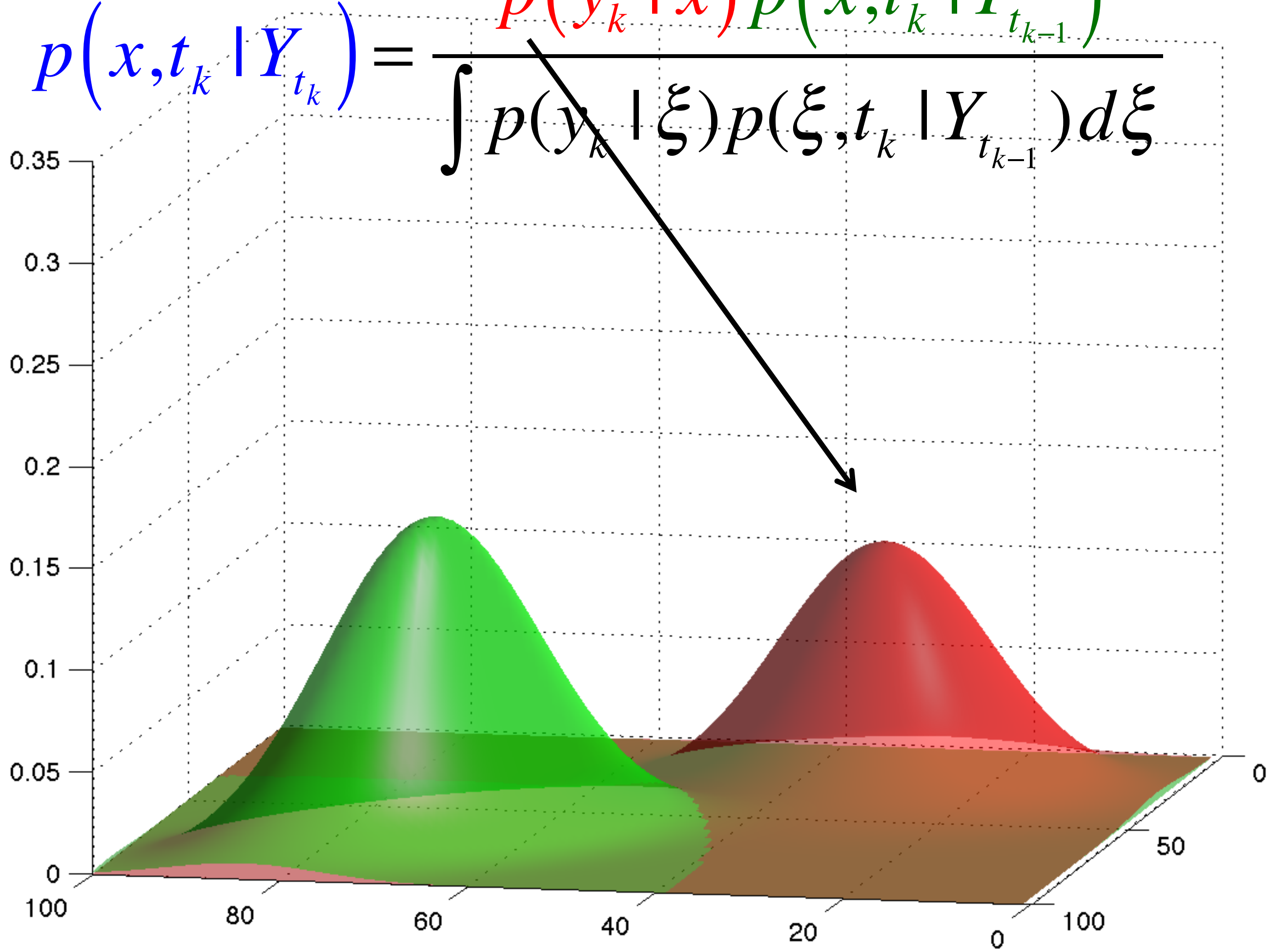
The Kalman Filter

$$p\left(x,t_k \mid Y_{t_k}\right) = \frac{p\left(y_k \mid x\right) p\left(x,t_k \mid Y_{t_{k-1}}\right)}{\int p(y_k \mid \xi) p(\xi,t_k \mid Y_{t_{k-1}}) d\xi} \qquad (10)$$



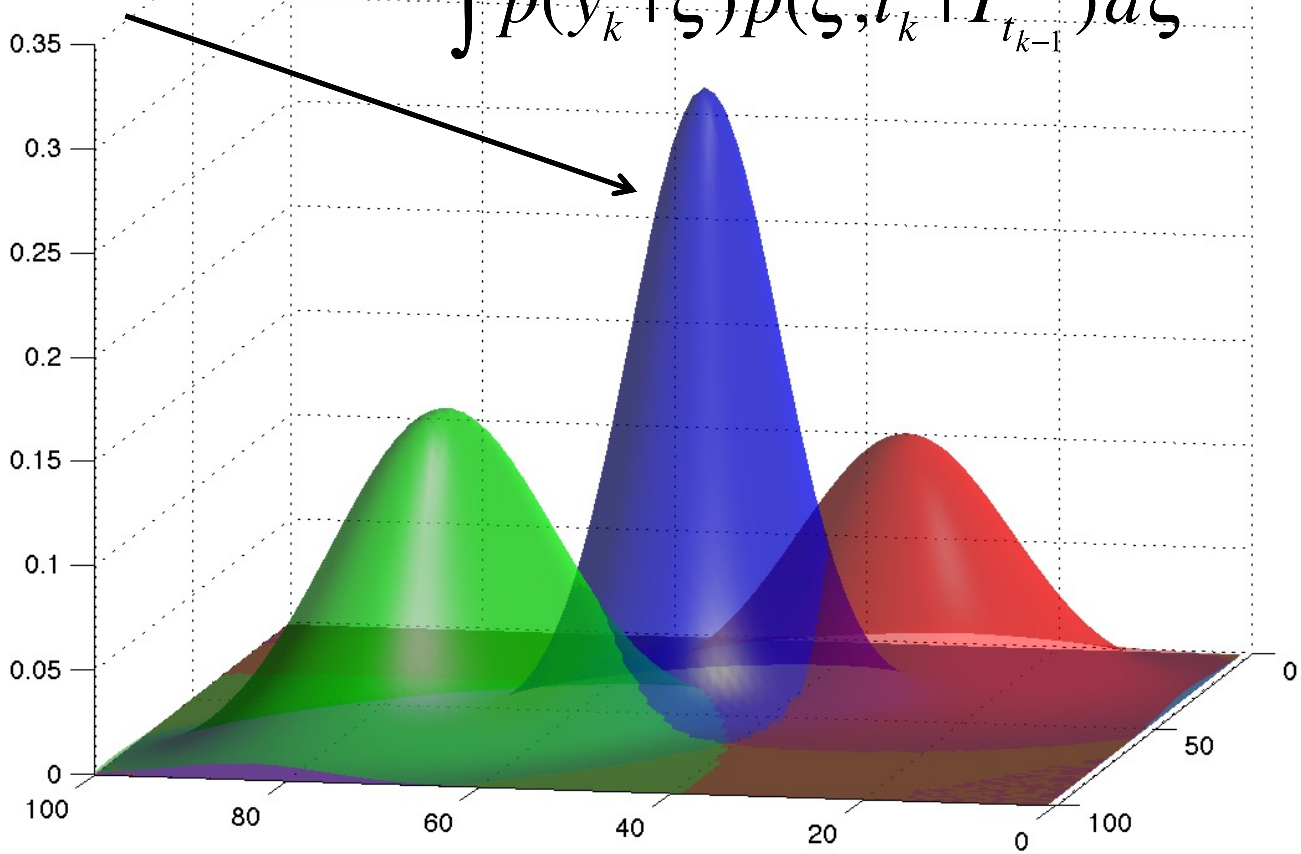
The Kalman Filter

$$p\left(x,t_k \mid Y_{t_k}\right) = \frac{p\left(y_k \mid x\right) p\left(x,t_k \mid Y_{t_{k-1}}\right)}{\int p(y_k \mid \xi) p(\xi,t_k \mid Y_{t_{k-1}}) d\xi} \qquad (10)$$



The Kalman Filter

$$p\left(x,t_k \mid Y_{t_k}\right) = \frac{p\left(y_k \mid x\right) p\left(x,t_k \mid Y_{t_{k-1}}\right)}{\int p(y_k \mid \xi) p(\xi,t_k \mid Y_{t_{k-1}}) d\xi} \tag{10}$$



Kalman Filter: Cost Challenges

Product of d-dimensional normals with means μ_1 and μ_2 and covariance matrices $\Sigma_1$and Σ_2 is normal.

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

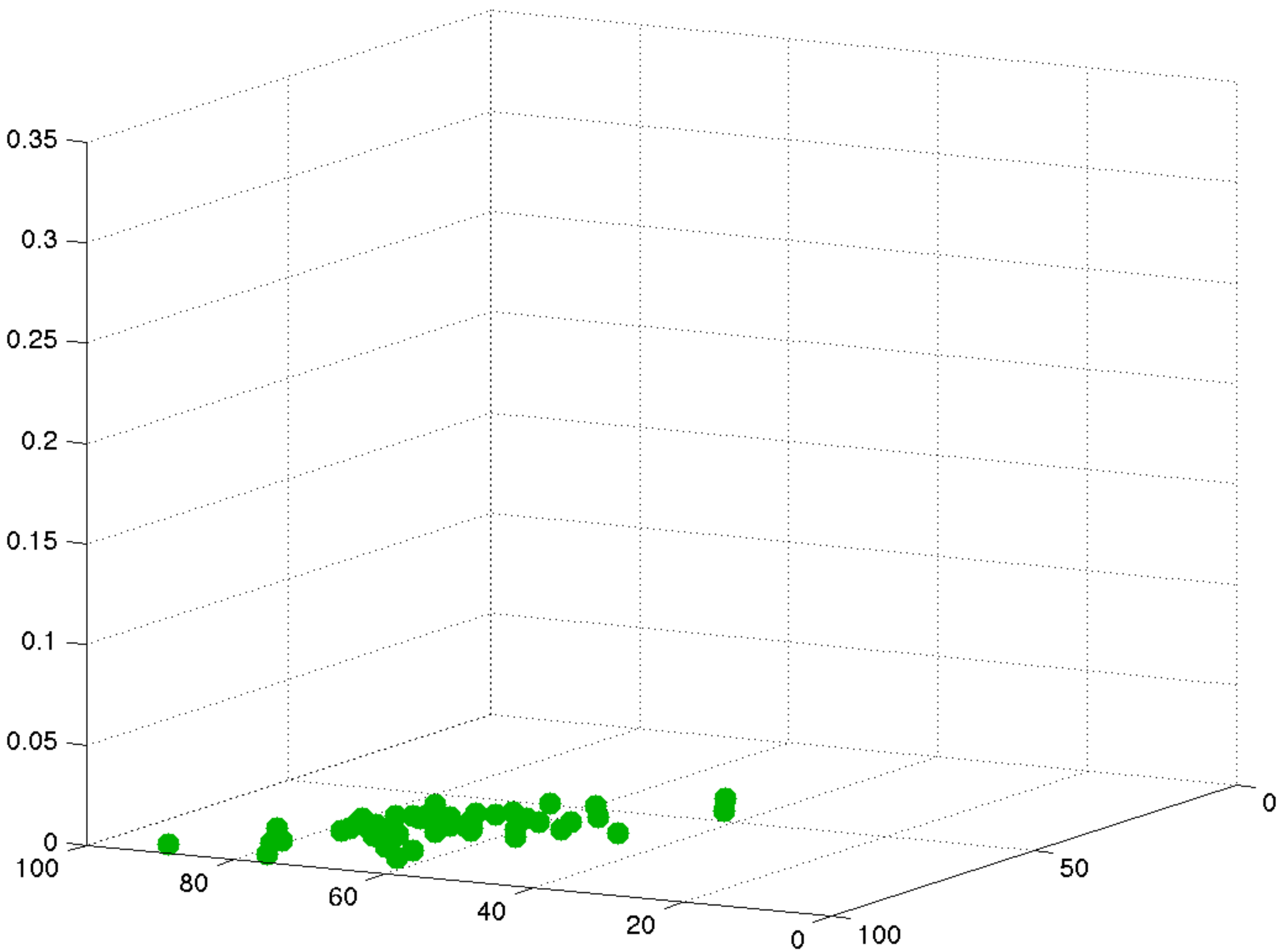
Covariance: $$\Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}$$

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, > $O(n^2)$.

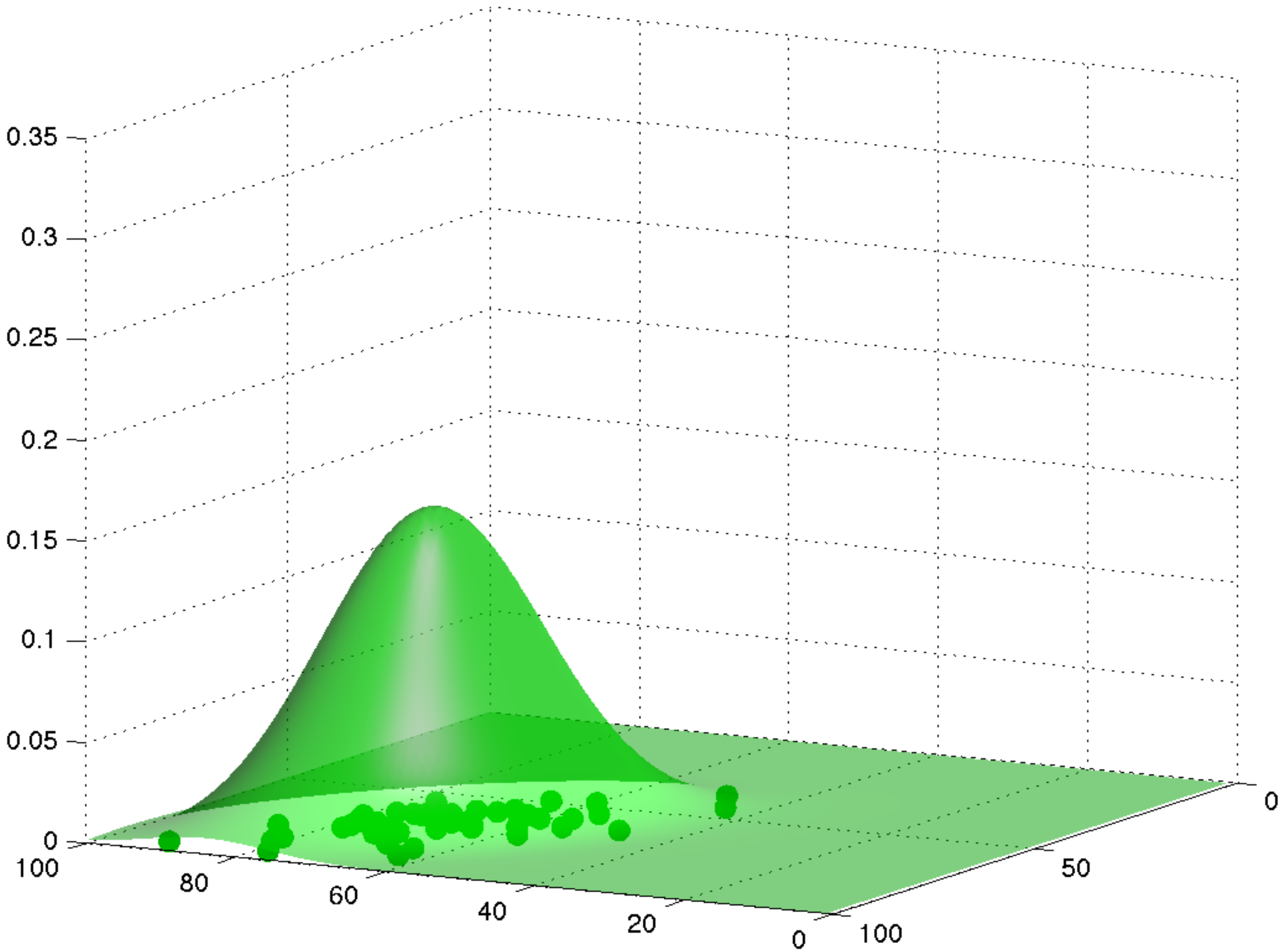
The Ensemble Kalman Filter

1. Start with ensemble of forecasts.



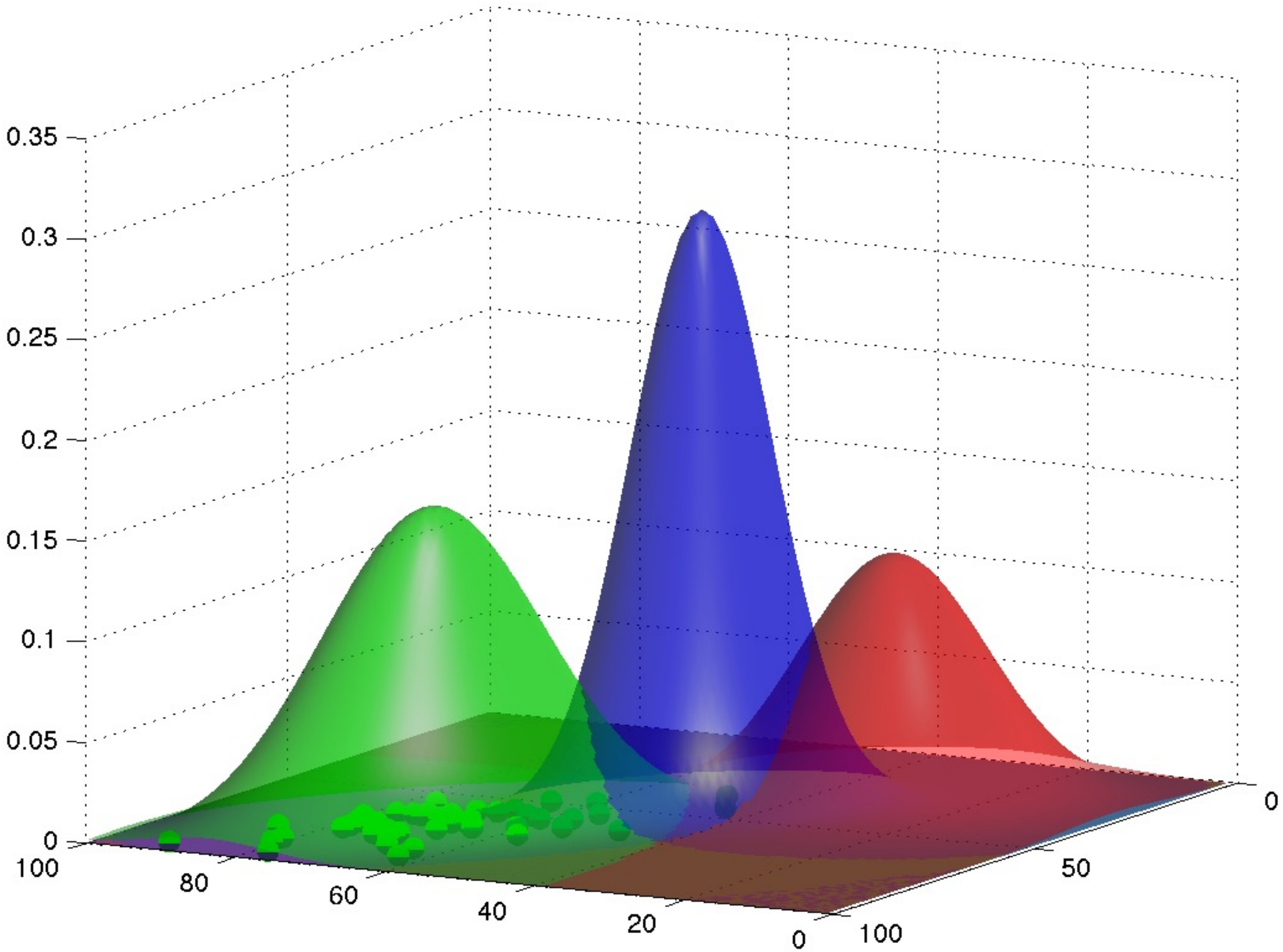
The Ensemble Kalman Filter

2. Fit a normal to ensemble.



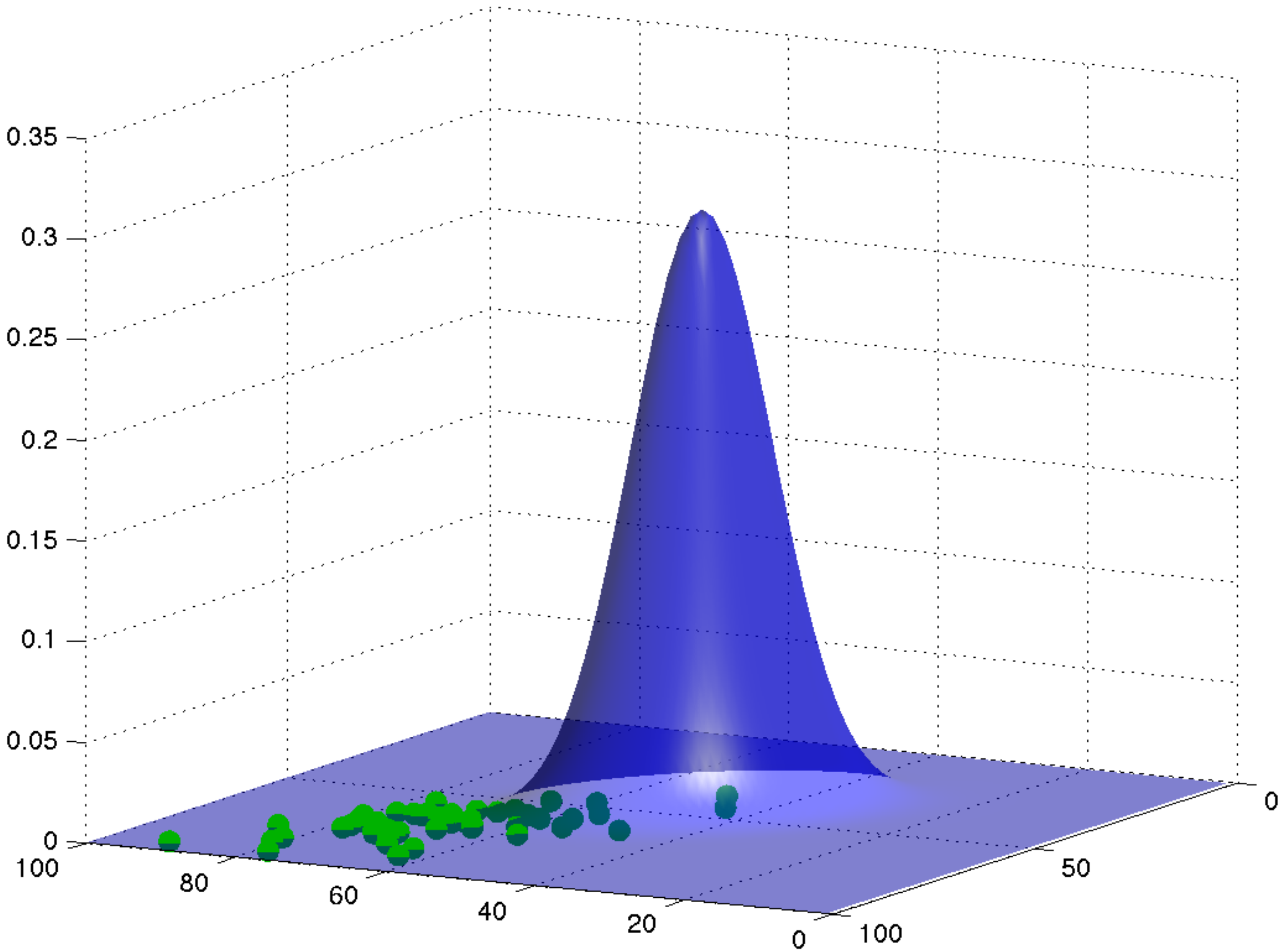
The Ensemble Kalman Filter

3. Do standard Kalman filter.



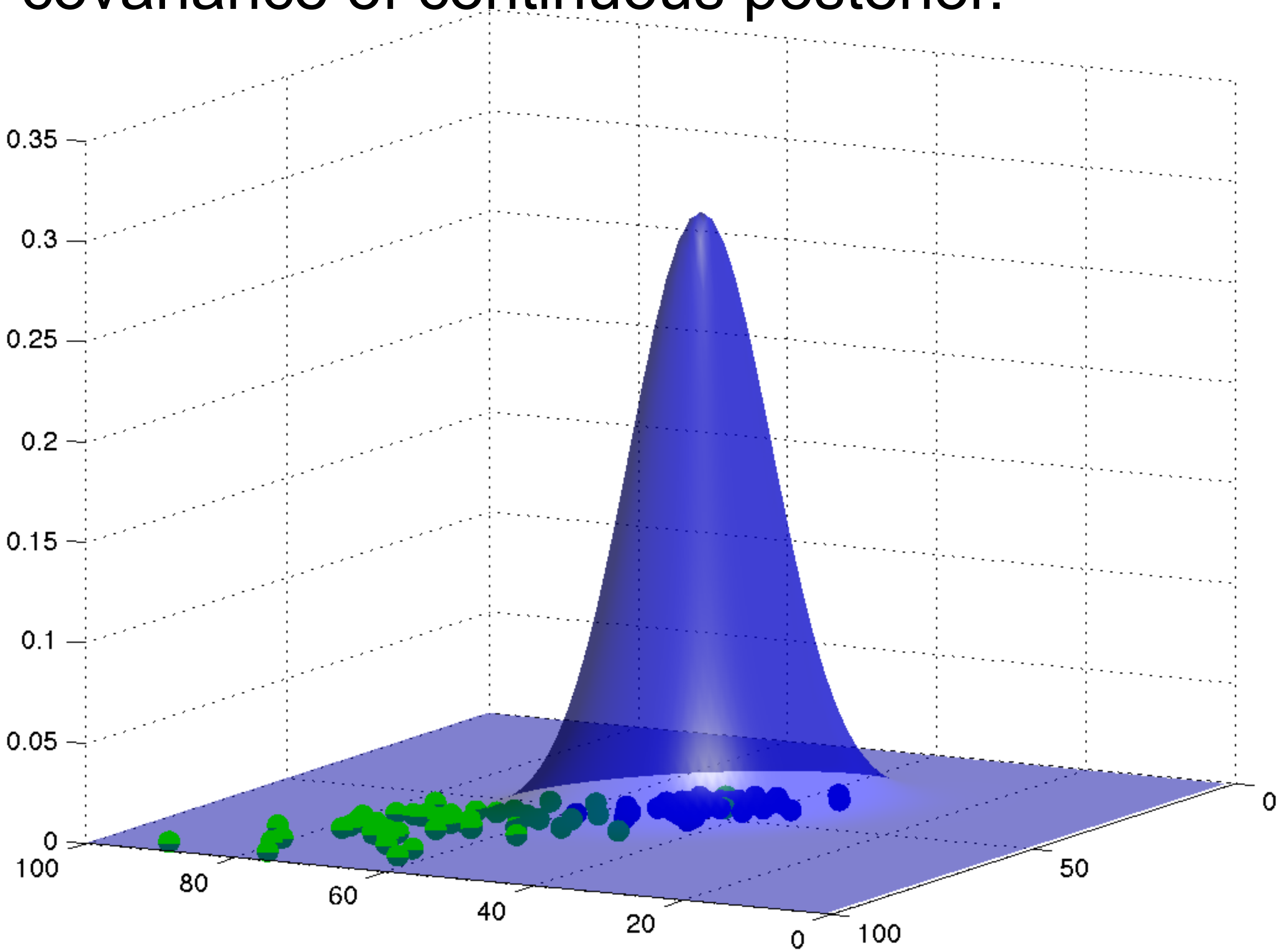
The Ensemble Kalman Filter

Have continuous posterior; need an ensemble.



The Ensemble Kalman Filter

4. Can create an ensemble with exact sample mean and covariance of continuous posterior.



Removing the Kalman from the Ensemble Kalman Filter

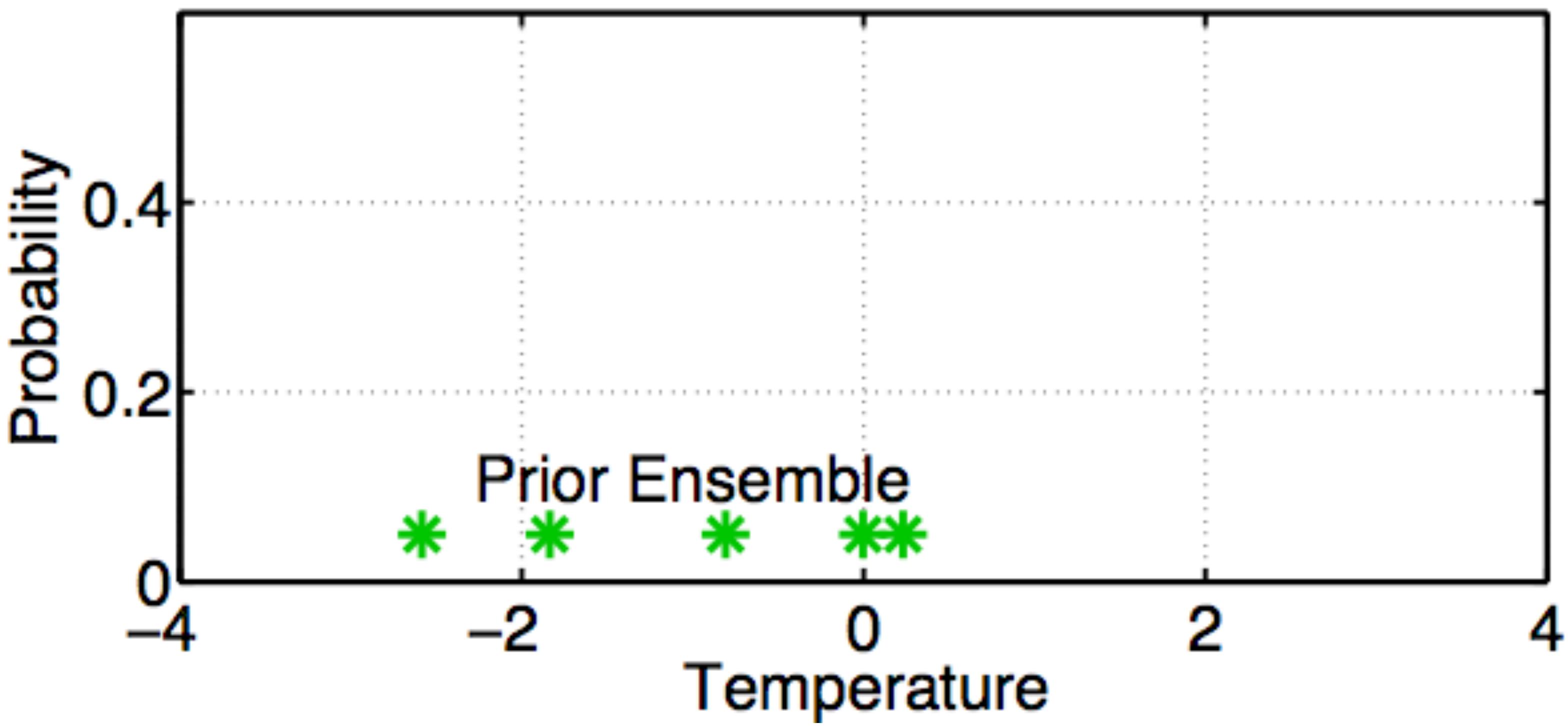
1. No need for linear model to advance covariance estimate.

One-Dimensional Ensemble Kalman Filter: Assimilating an Observation

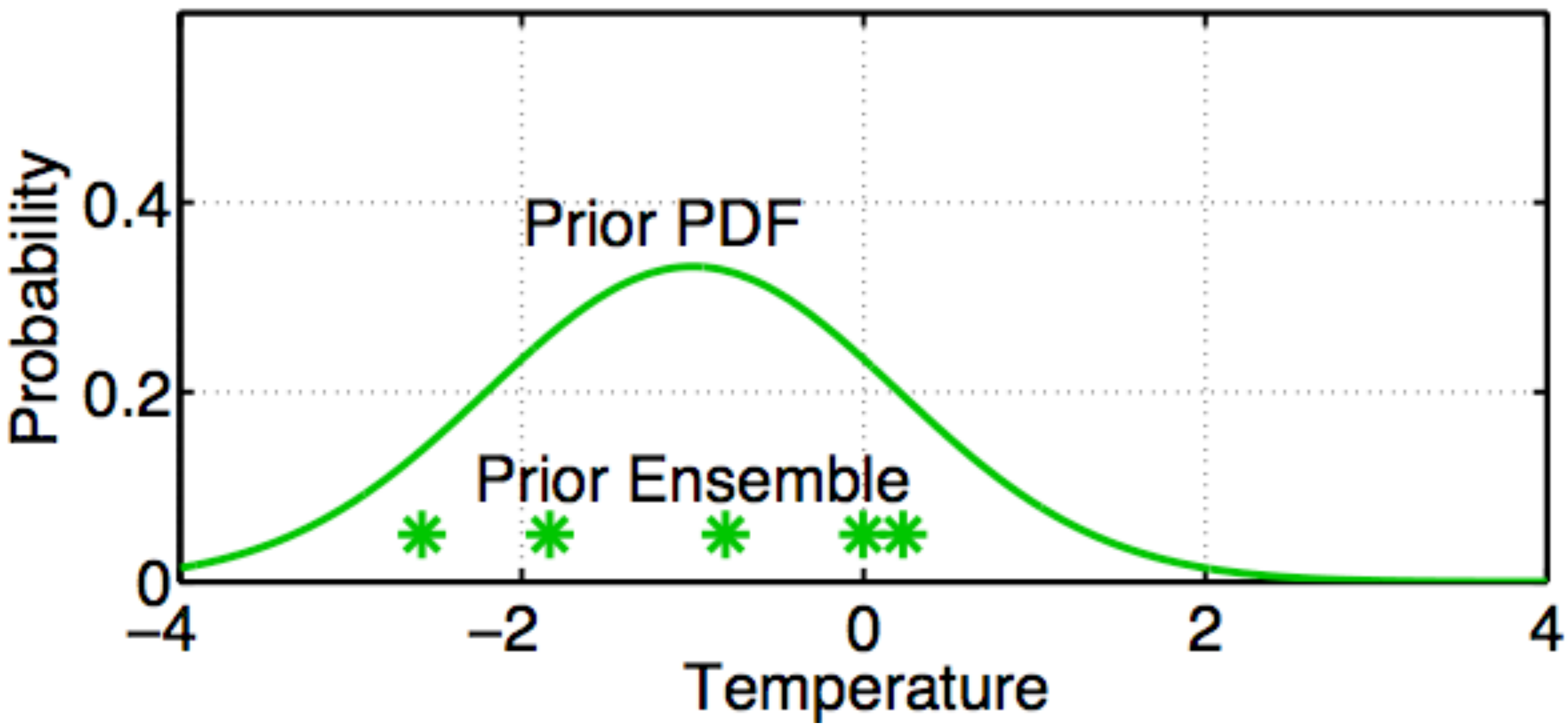
Without loss of generality (for Kalman filter)…

Can assimilate observations serially, one at a time.

One-Dimensional Ensemble Kalman Filter: Assimilating an Observation

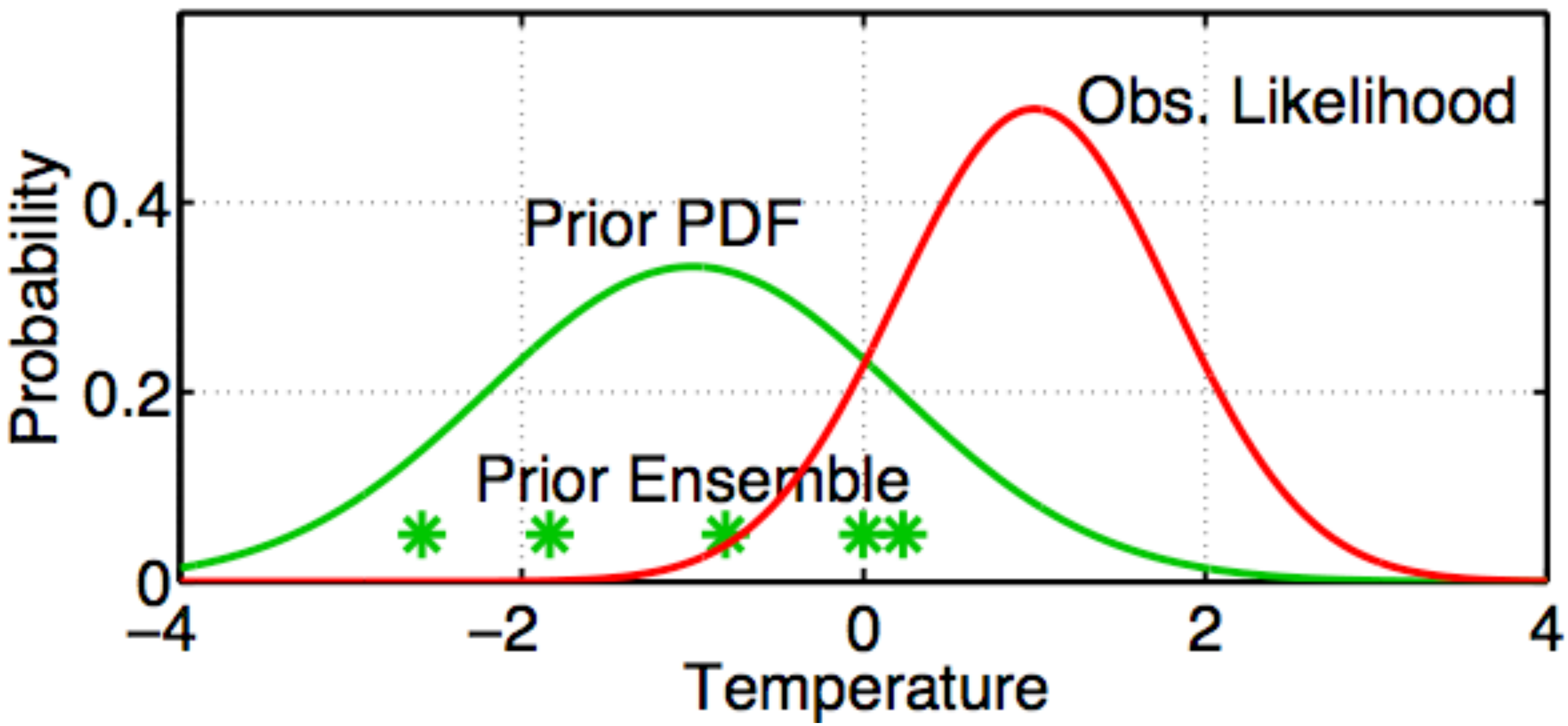


One-Dimensional Ensemble Kalman Filter: Assimilating an Observation



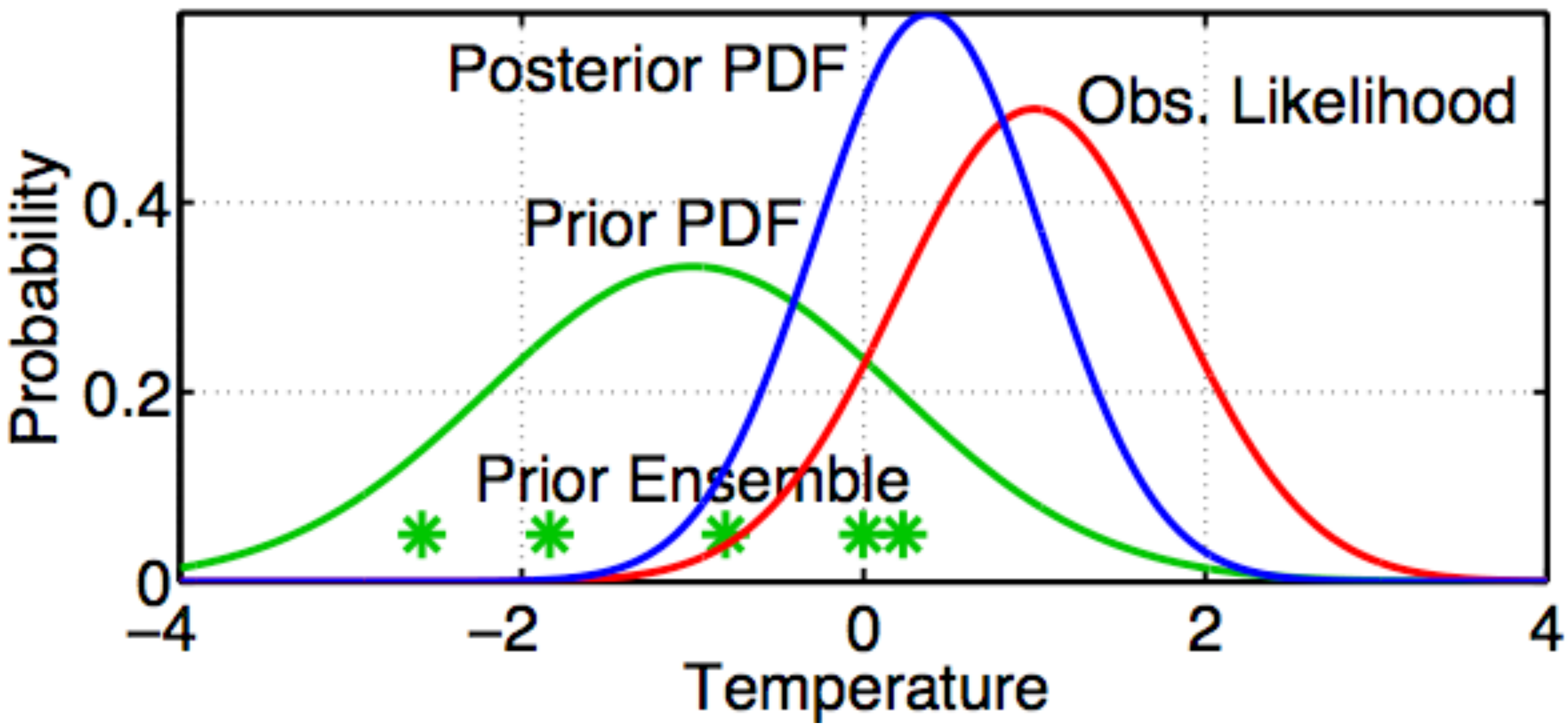
Fit a Gaussian to the sample.

One-Dimensional Ensemble Kalman Filter: Assimilating an Observation



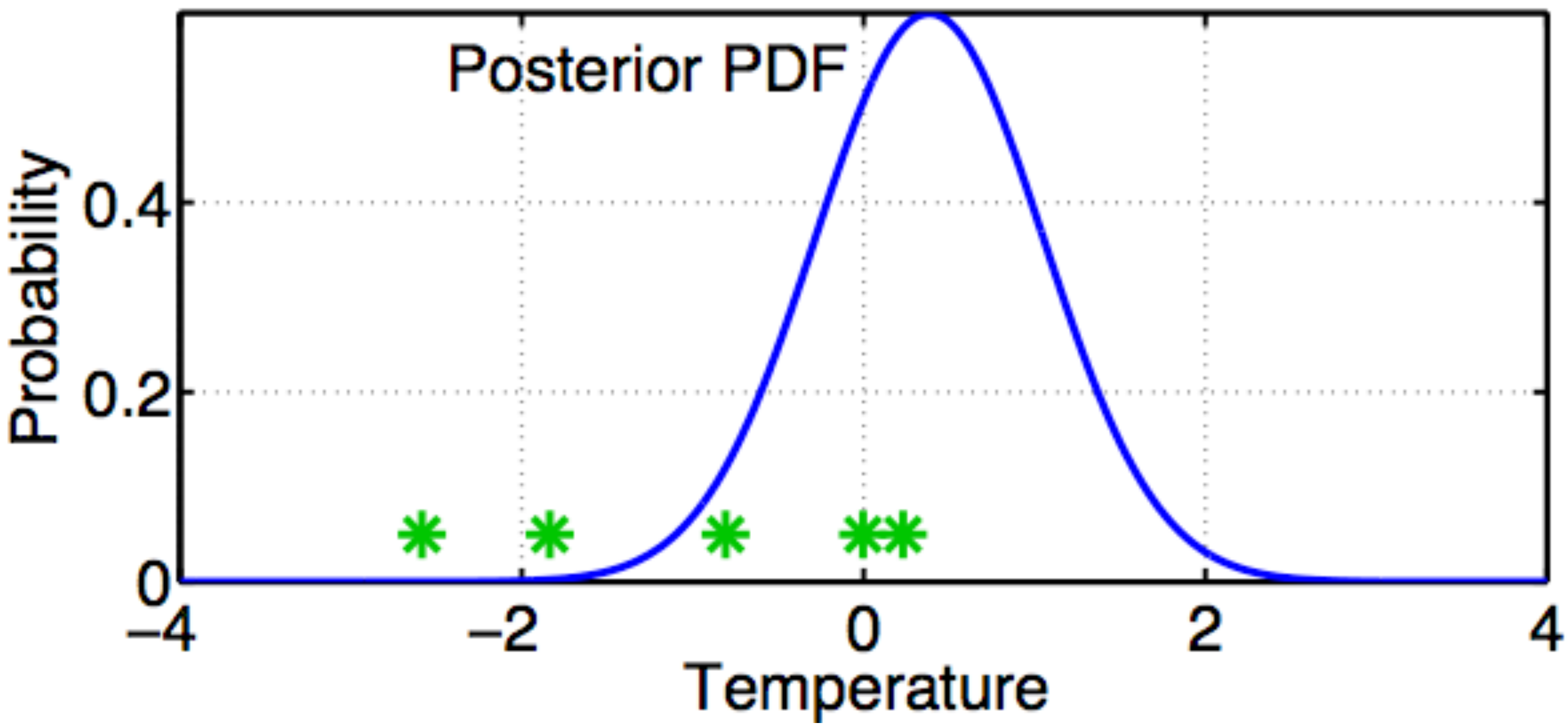
Get the observation likelihood.

One-Dimensional Ensemble Kalman Filter: Assimilating an Observation



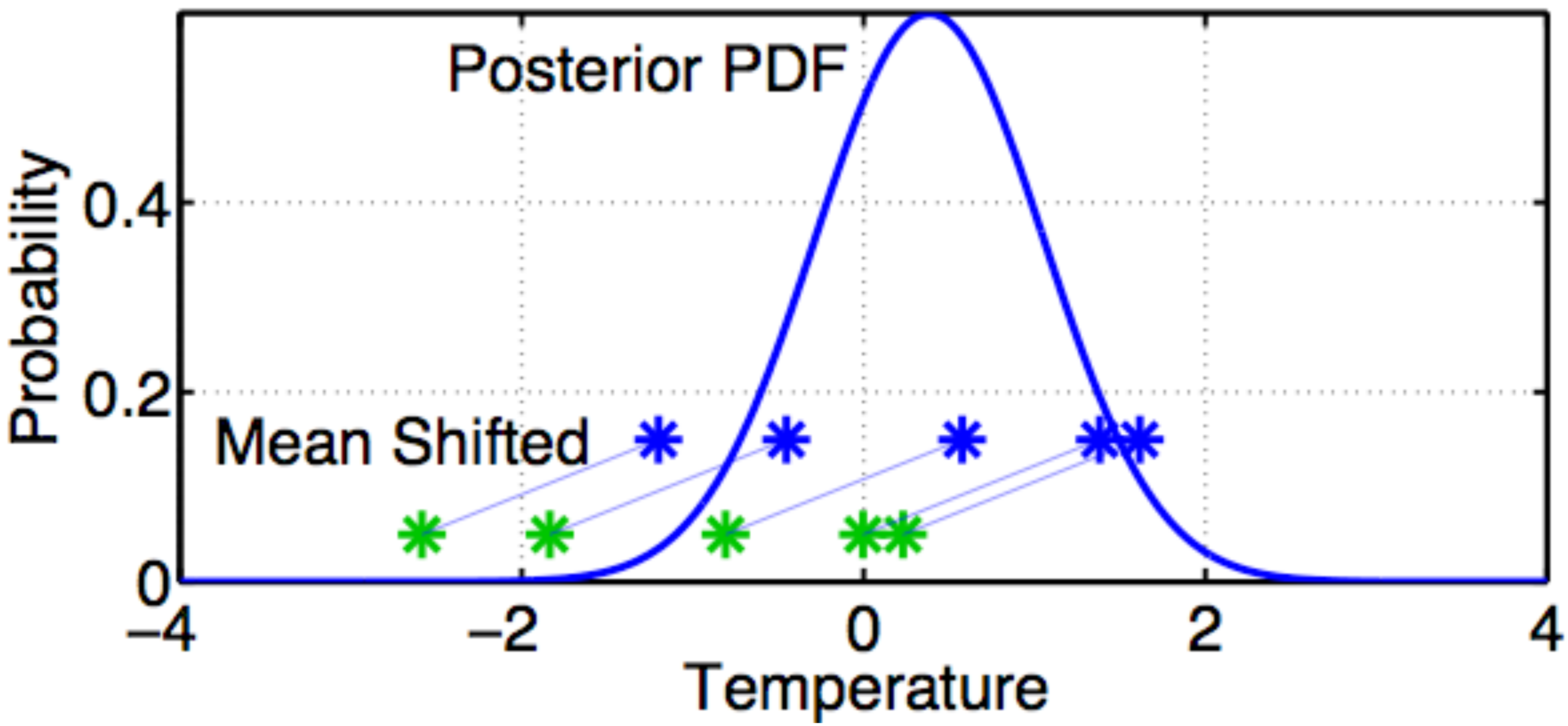
Compute the continuous posterior PDF.

One-Dimensional Ensemble Kalman Filter: Assimilating an Observation



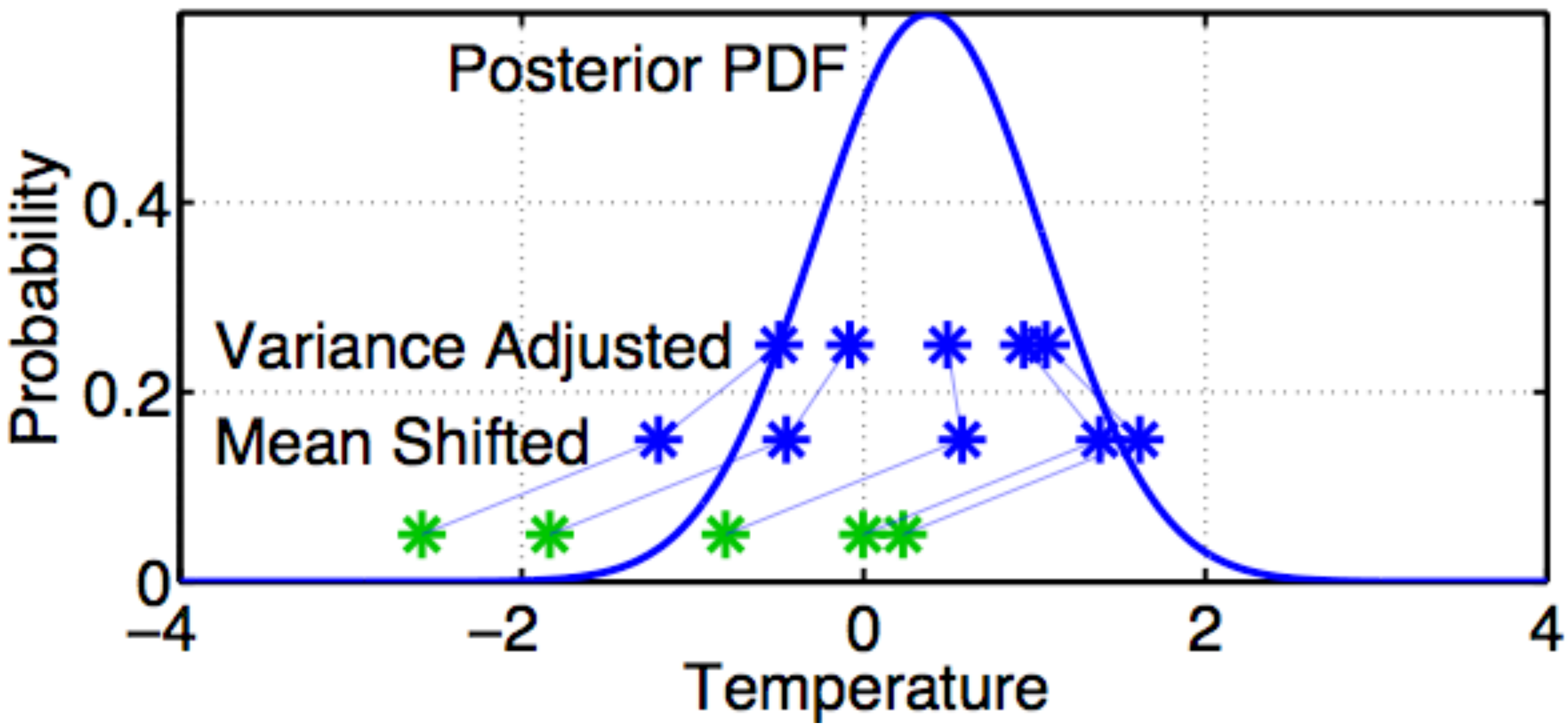
Use a deterministic algorithm to ‘adjust’ the ensemble.

One-Dimensional Ensemble Kalman Filter: Assimilating an Observation



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

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.

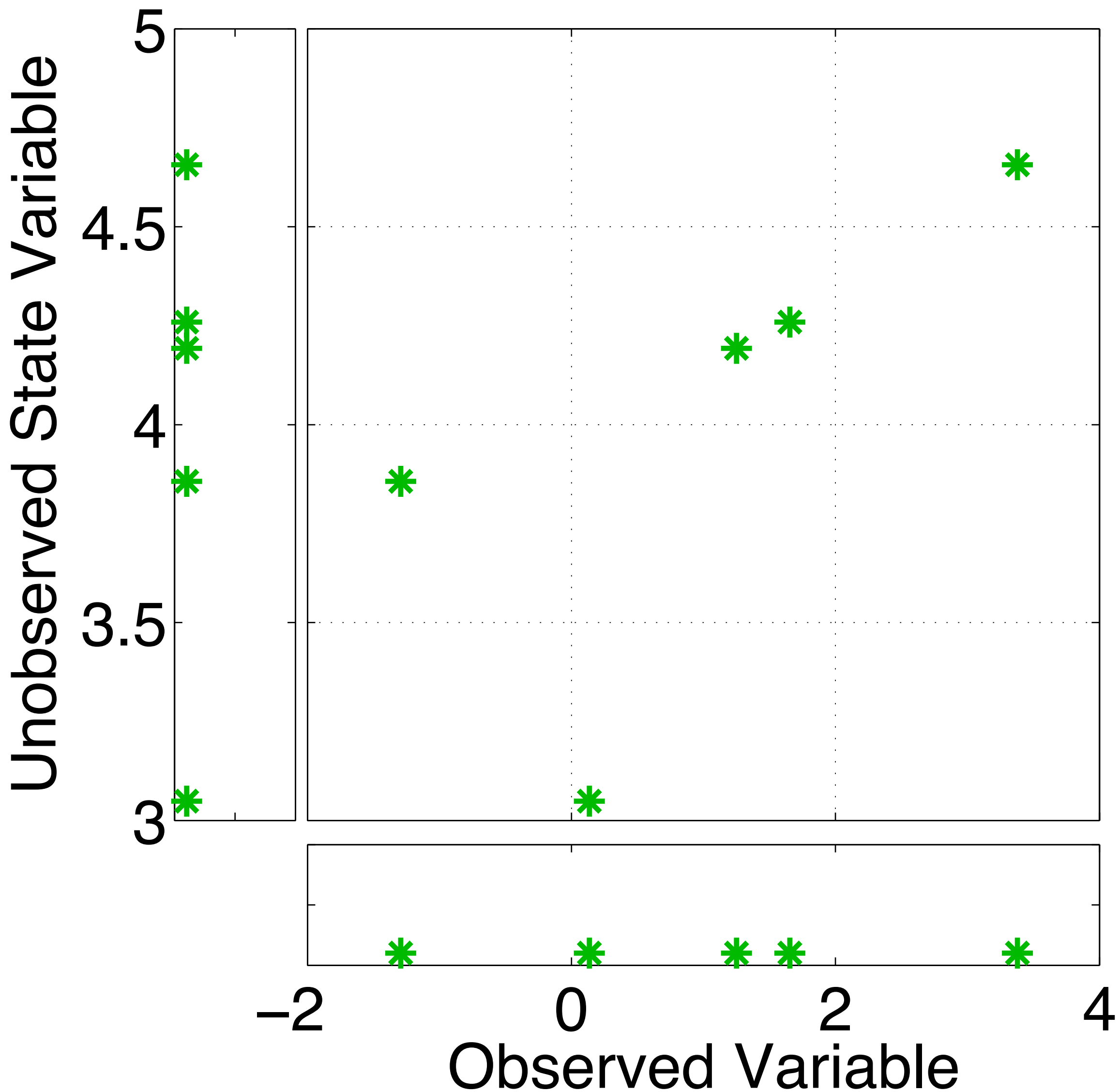
Sample statistics are identical to Kalman filter.

Ensemble filters: Updating additional prior state variables

Without loss of generality (for Kalman filter)…

Can compute impact of observation on each state variable independently.

Ensemble filters: Updating additional prior state variables

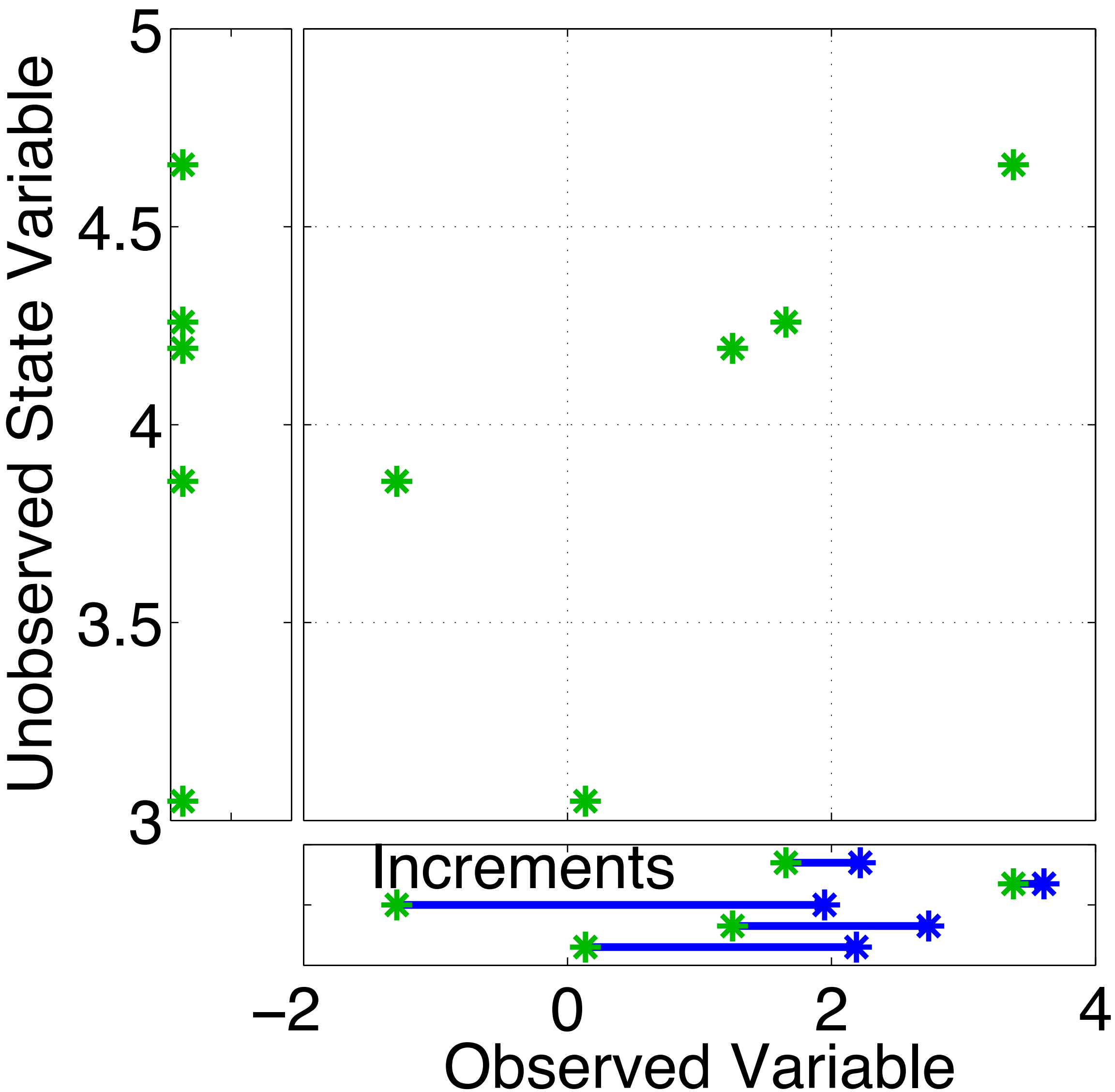


Assume that all we know is the prior joint distribution.

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.

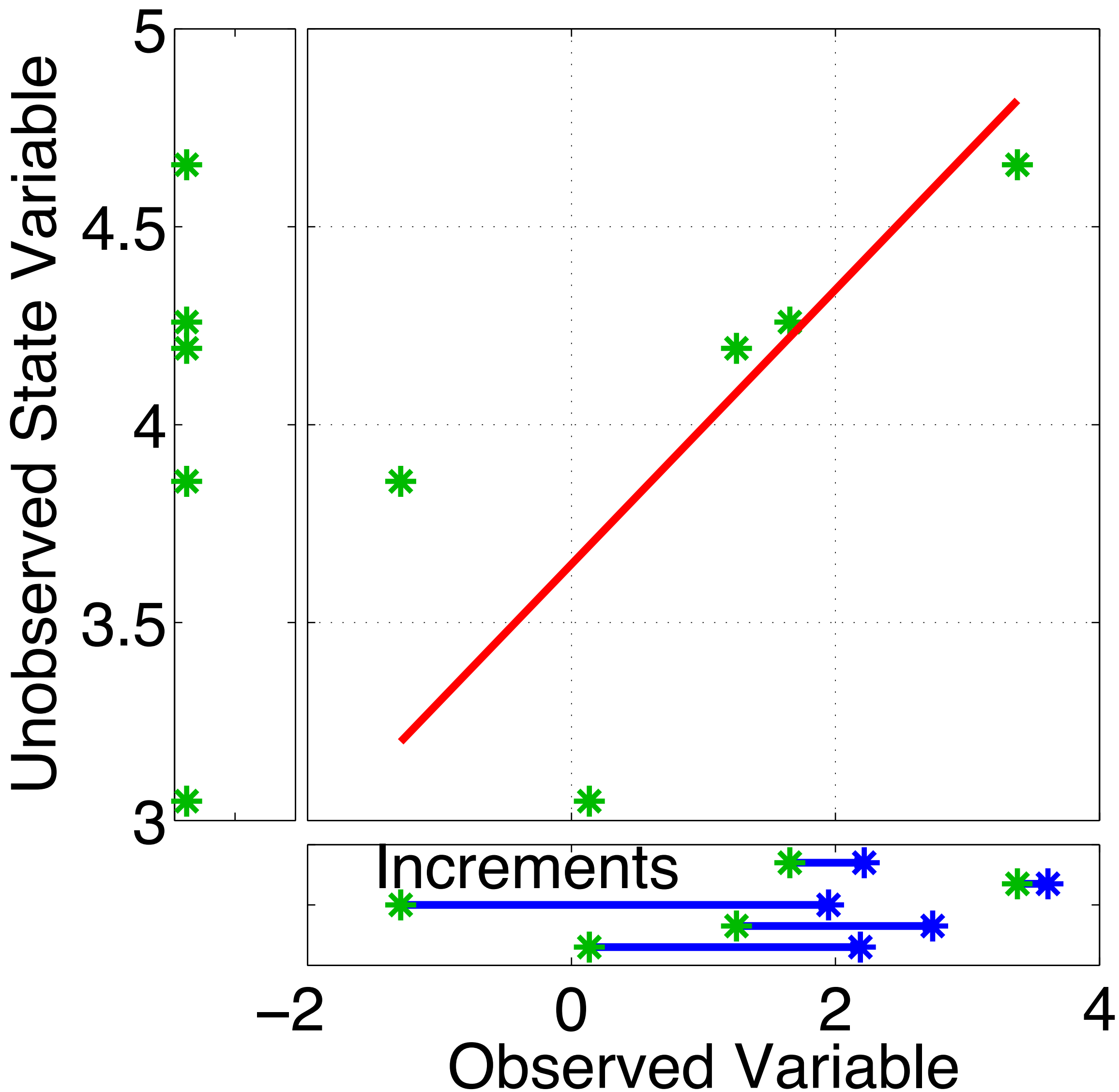
How should the unobserved variable be impacted?

Least squares.

Equivalent to linear regression.

Same as assuming bi-Gaussian prior.

Ensemble filters: Updating additional prior state variables

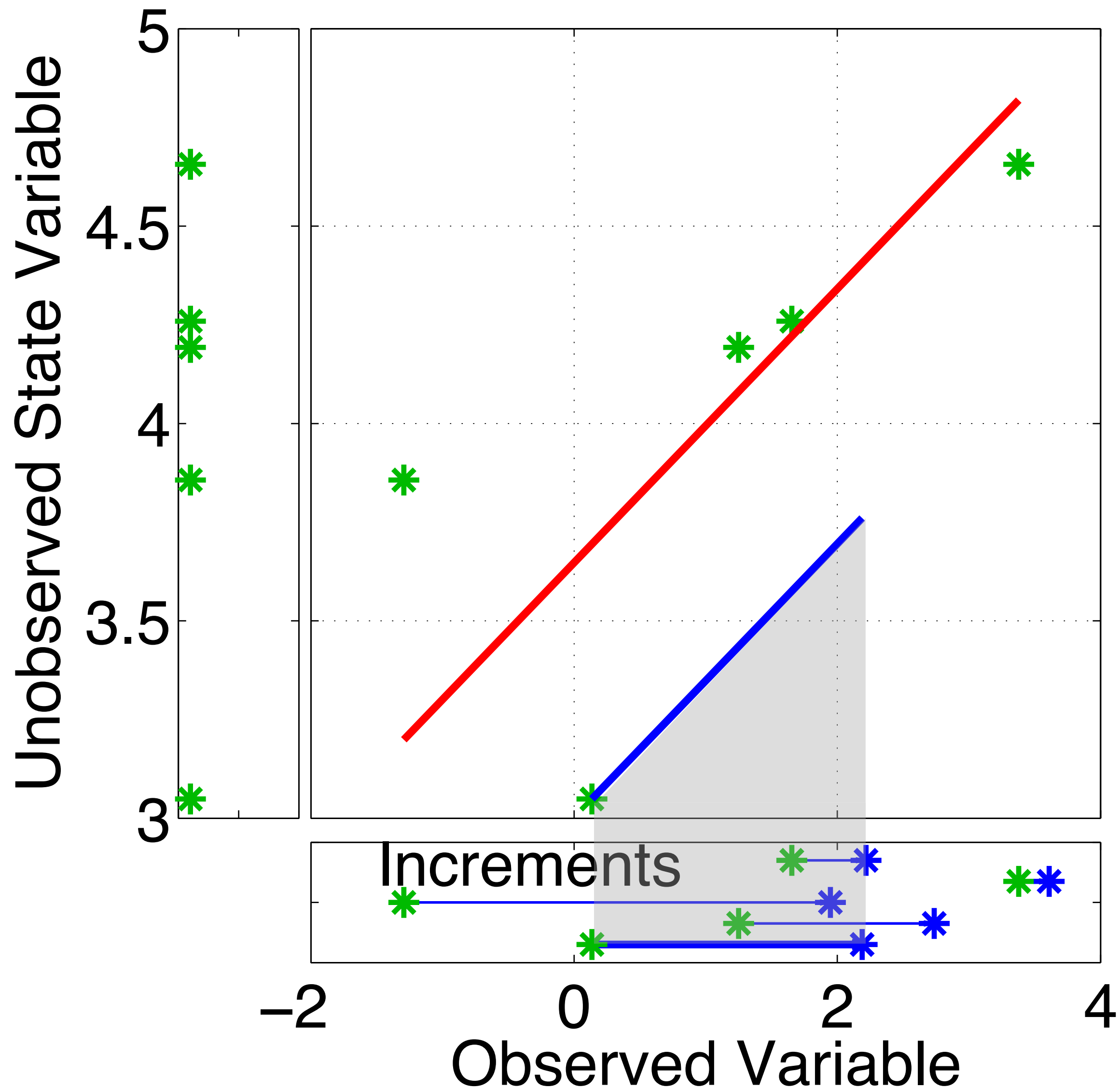


Have joint prior distribution of two variables.

How should the unobserved variable be impacted?

1st choice: least squares

Begin by finding least squares fit.

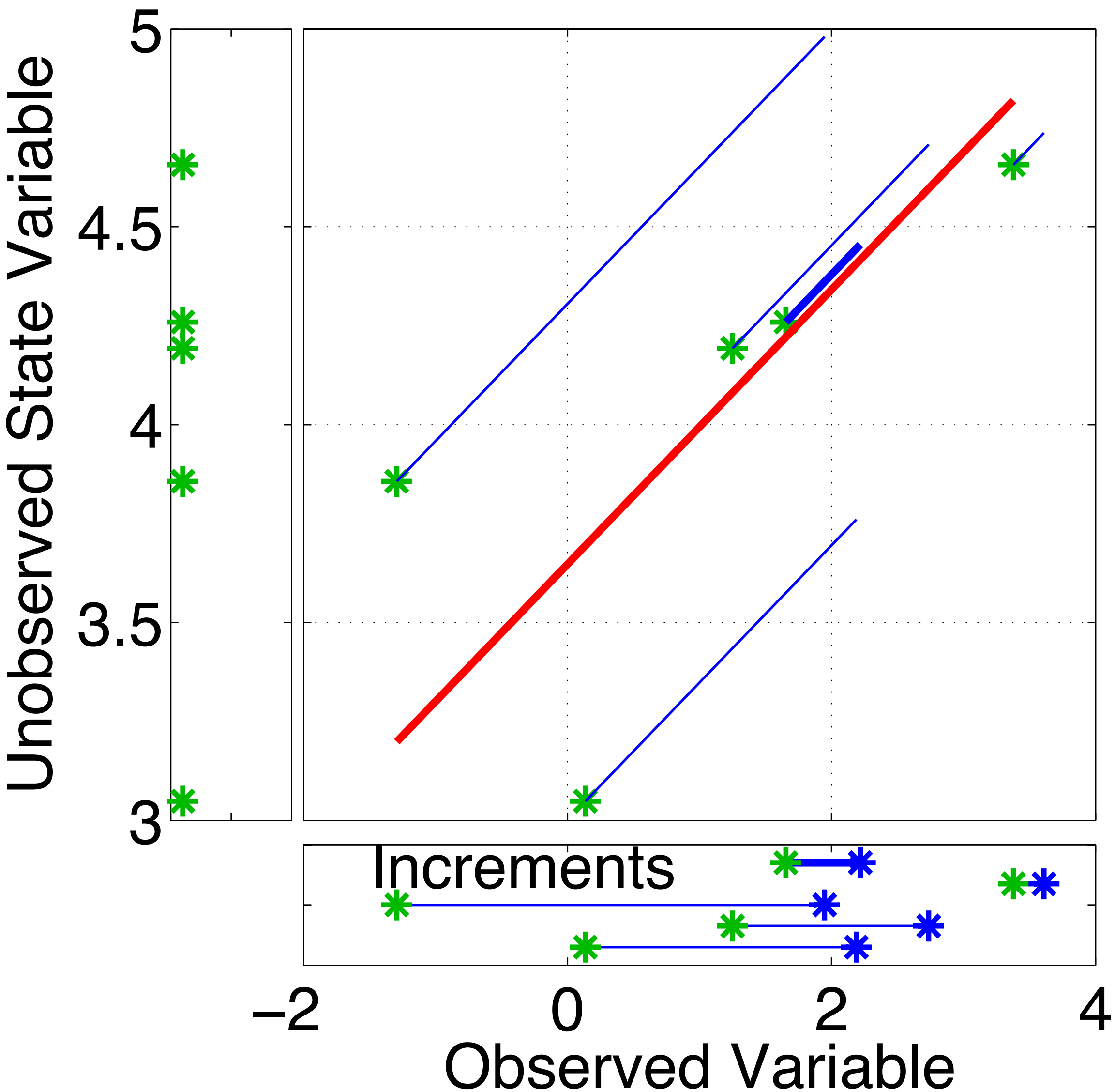


Have joint prior distribution of two variables.

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

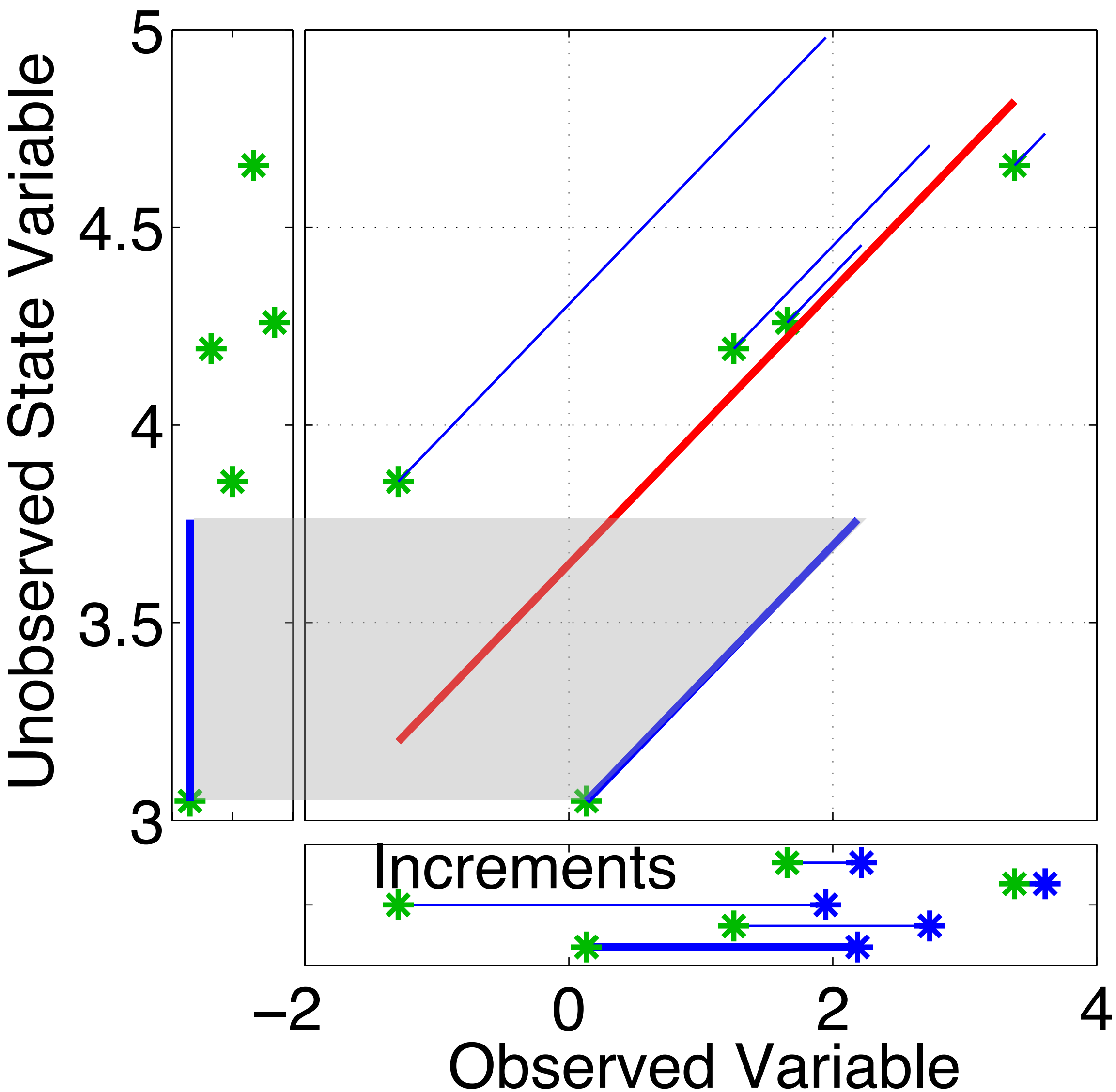


Have joint prior distribution of two variables.

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

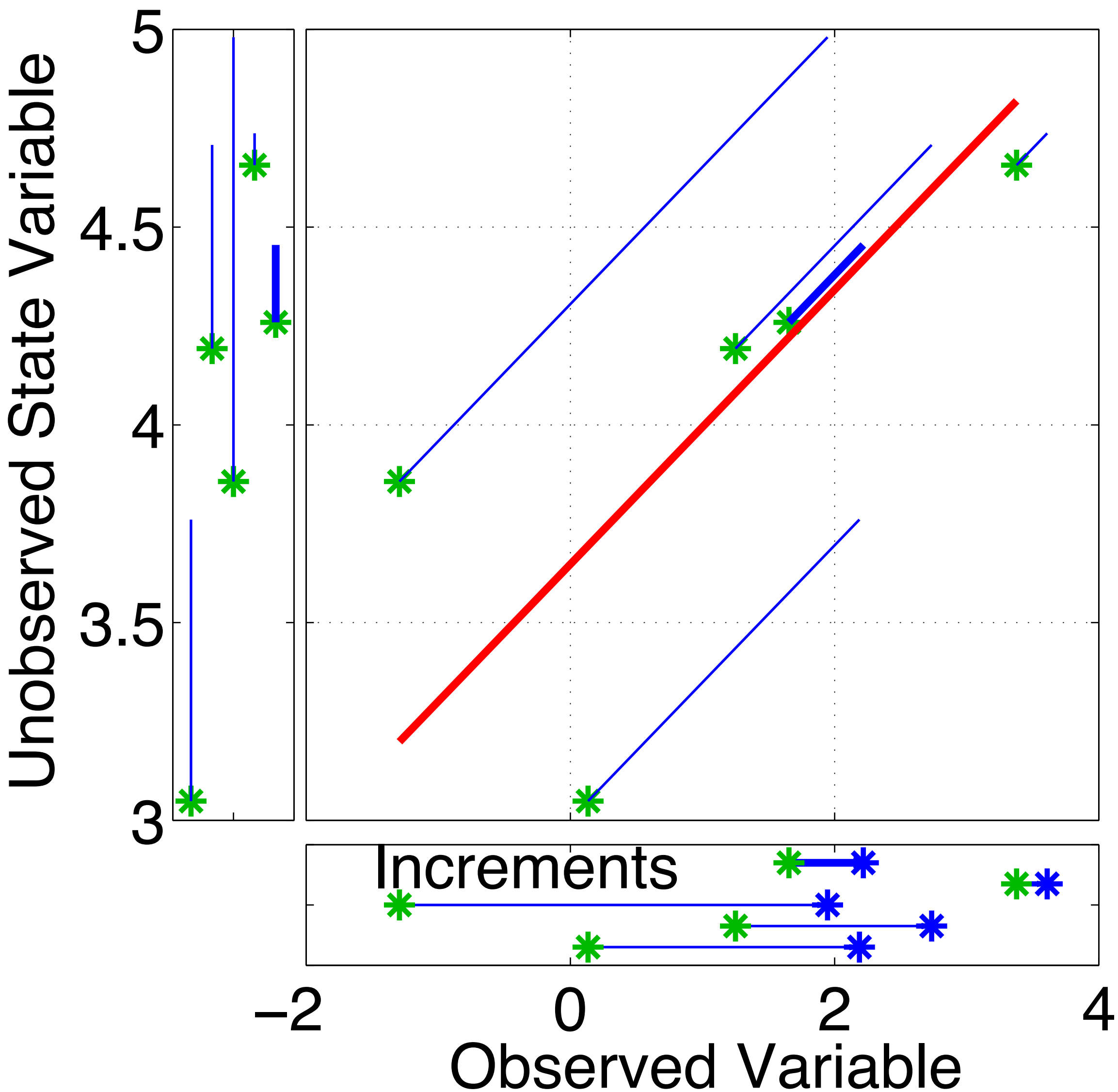


Have joint prior distribution of two variables.

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

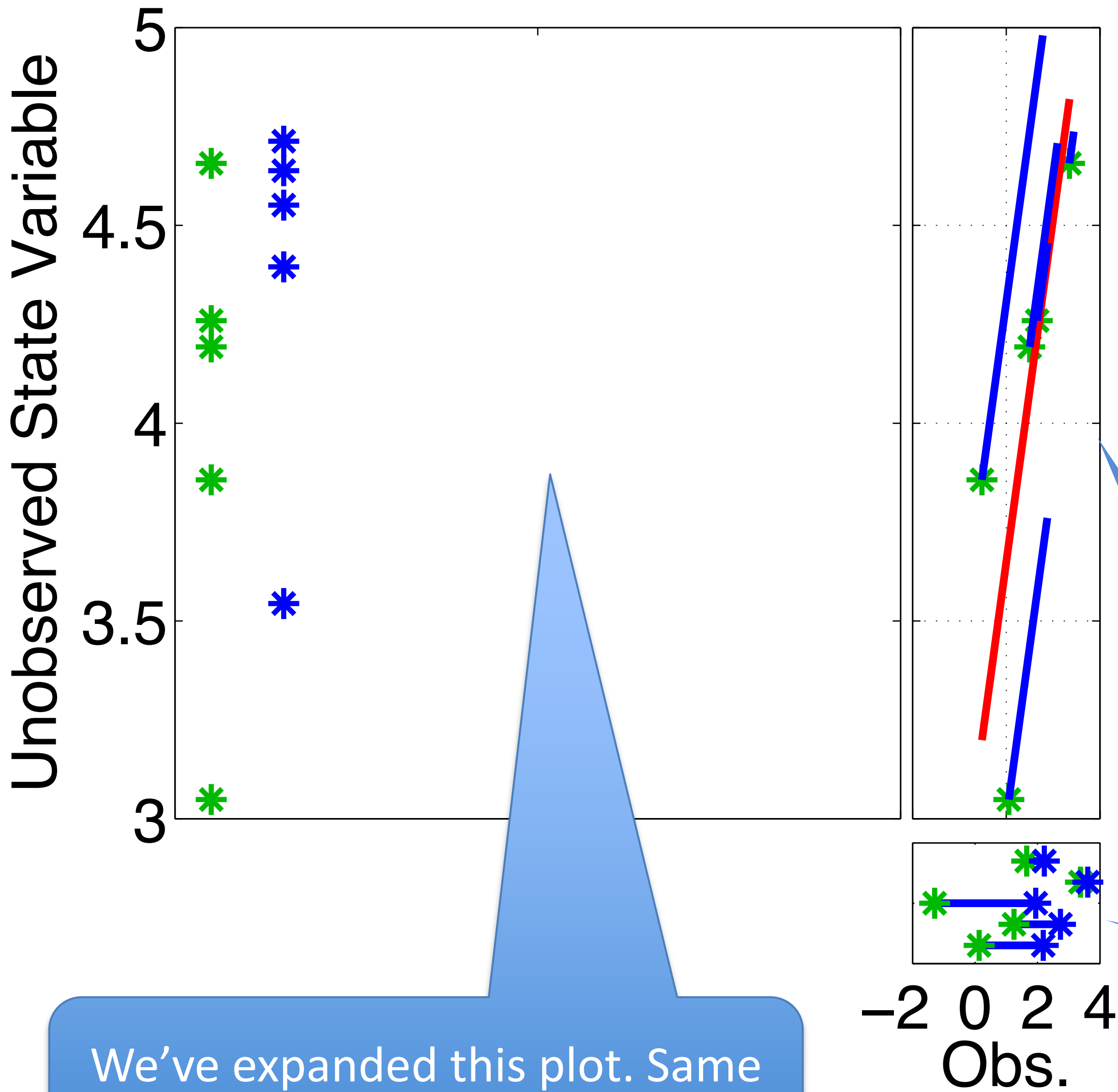


Have joint prior distribution of two variables.

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

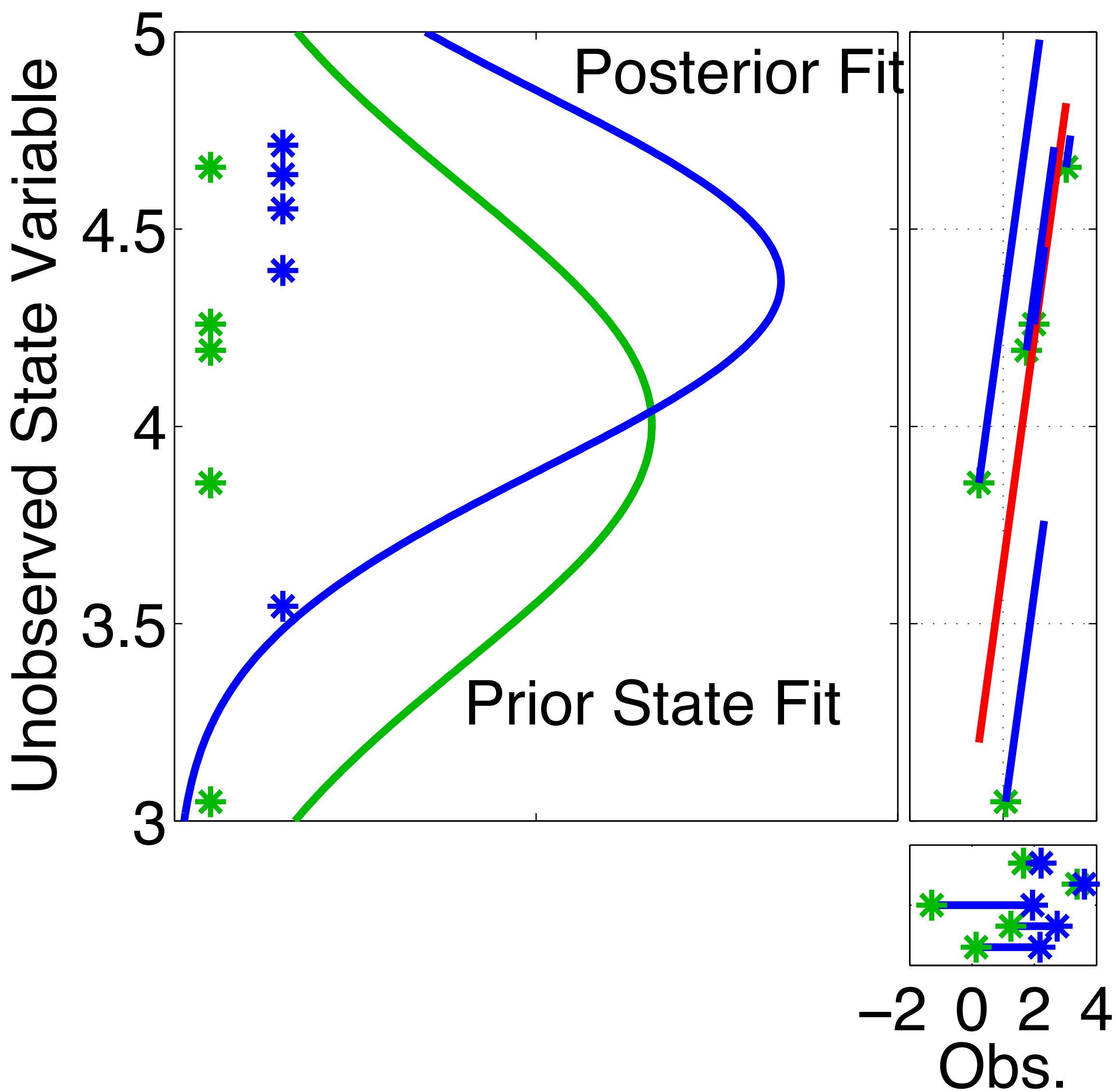


Now have an updated (posterior) ensemble for the unobserved variable.

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.

Other features of the prior distribution may also have changed.

How an Ensemble Filter Works for Geophysical Data Assimilation

Without loss of generality (for ensemble Kalman filter)…

Can do data assimilation in the following way.

How an Ensemble Filter Works for Geophysical Data Assimilation

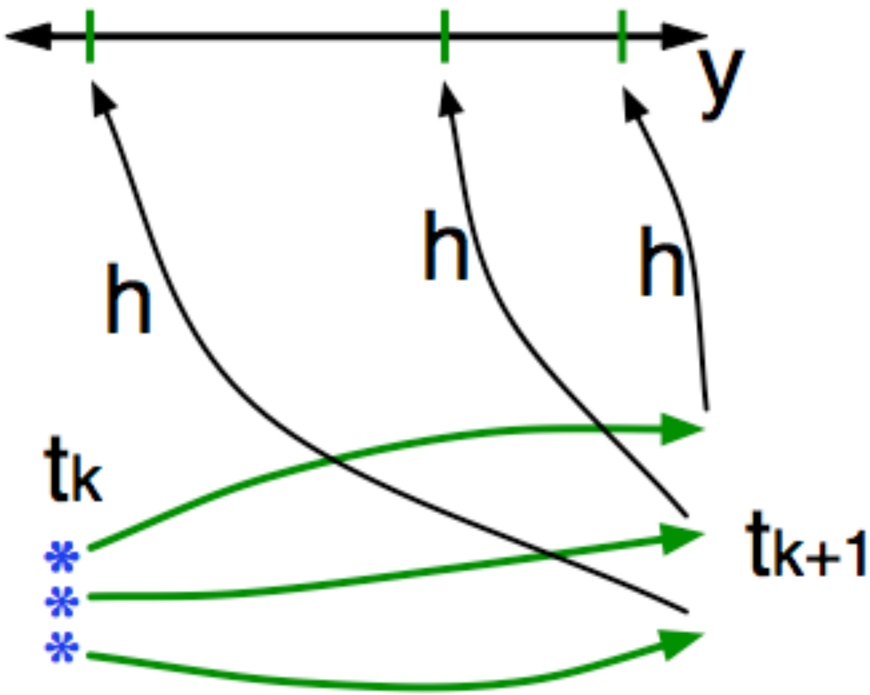
1. Use model to advance ensemble (3 members here) to time at which next observation becomes available.

Ensemble state estimate after using previous observation (analysis)

Ensemble state at time of next observation (prior)

How an Ensemble Filter Works for Geophysical Data Assimilation

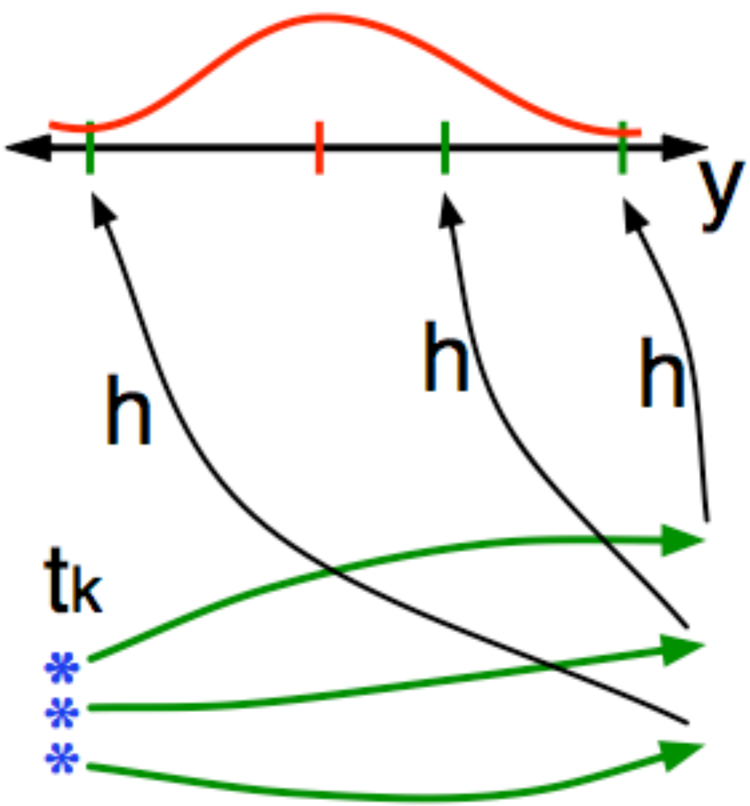
2. Get prior ensemble sample of observation, $y = h(x)$, by applying forward operator **h** to each ensemble member.



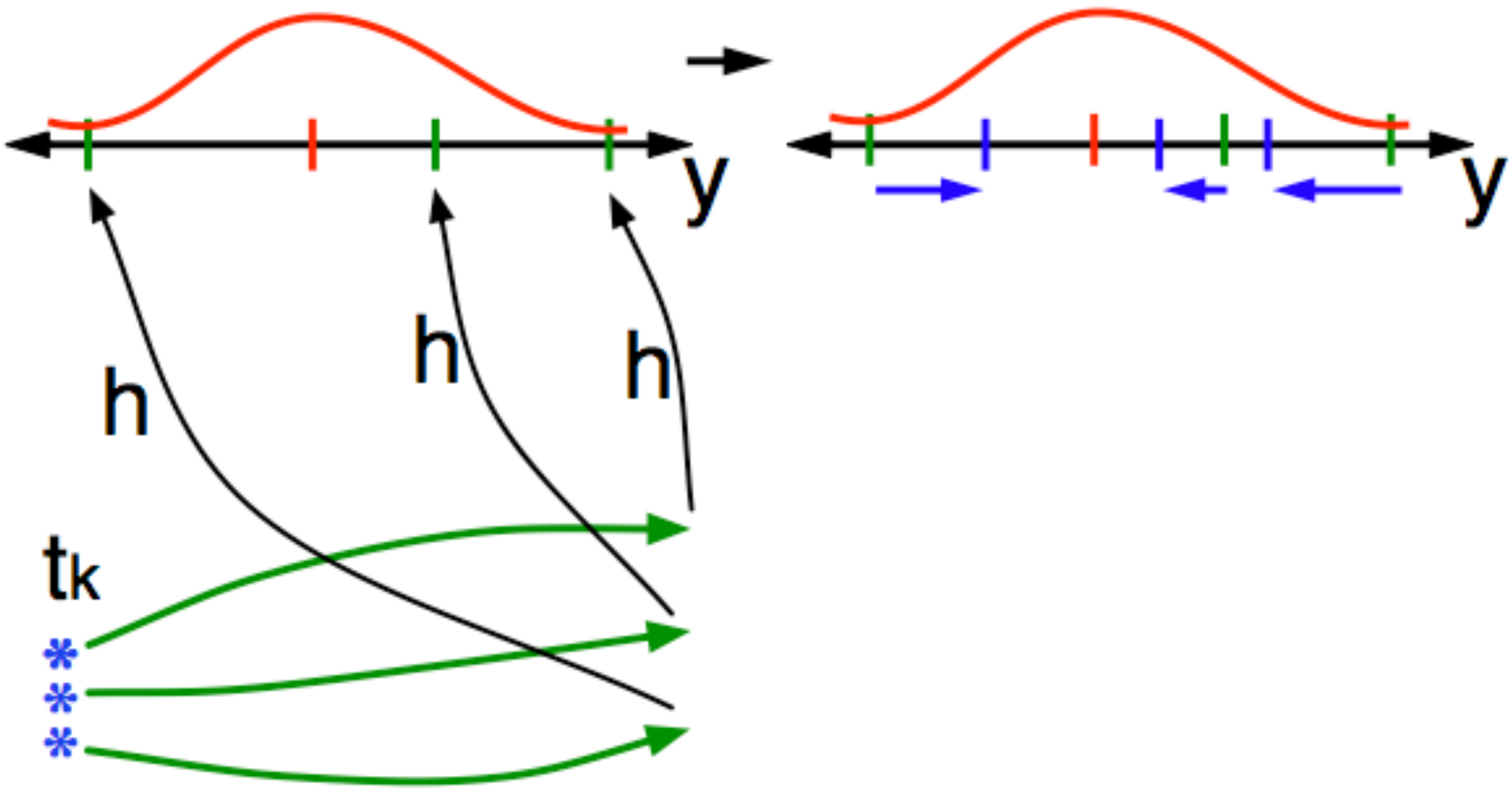
Theory: observations from instruments with uncorrelated errors can be done sequentially.

How an Ensemble Filter Works for Geophysical Data Assimilation

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

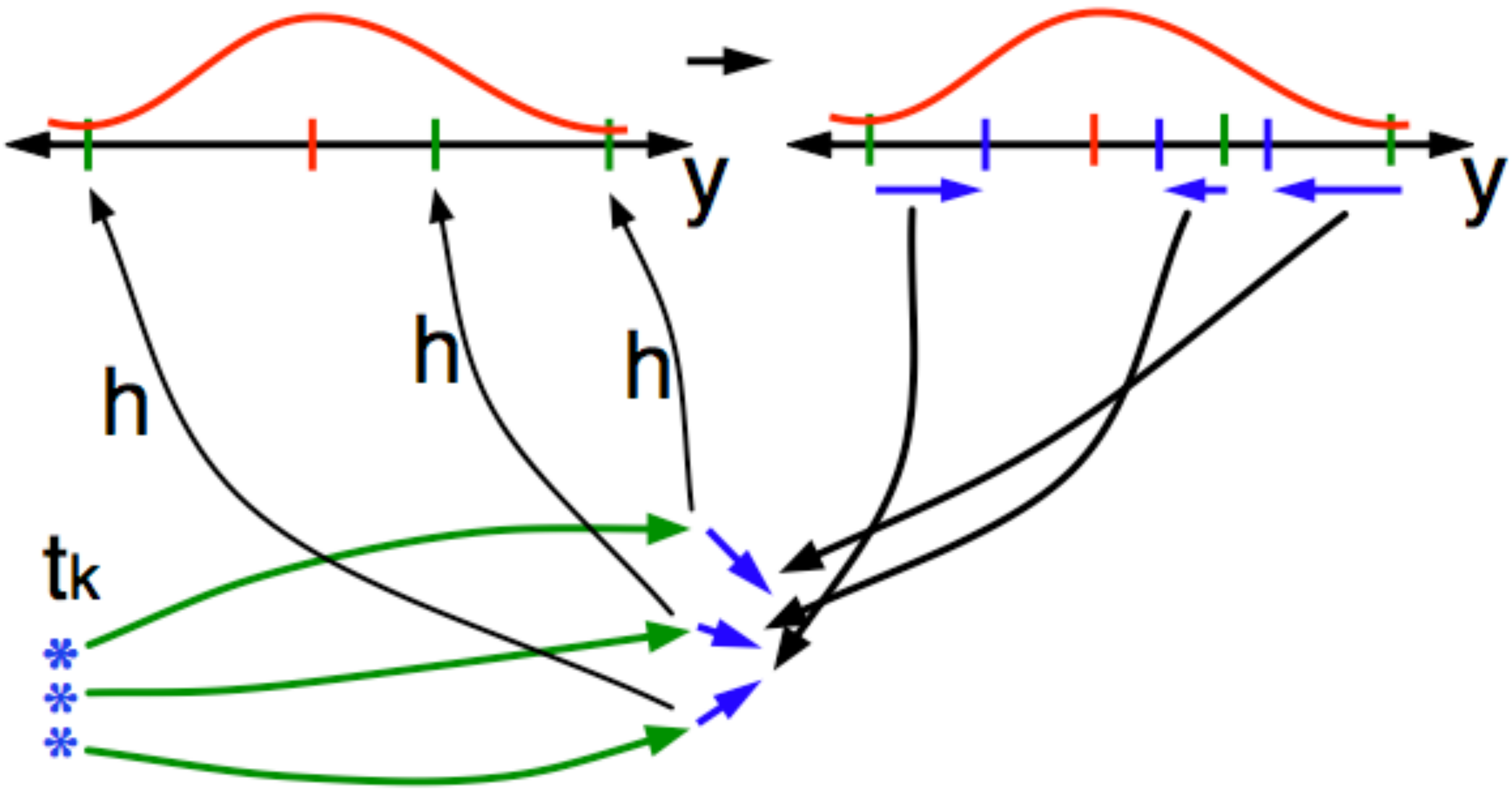


4. Find the increments for the prior observation ensemble (this is a scalar problem for uncorrelated observation errors).

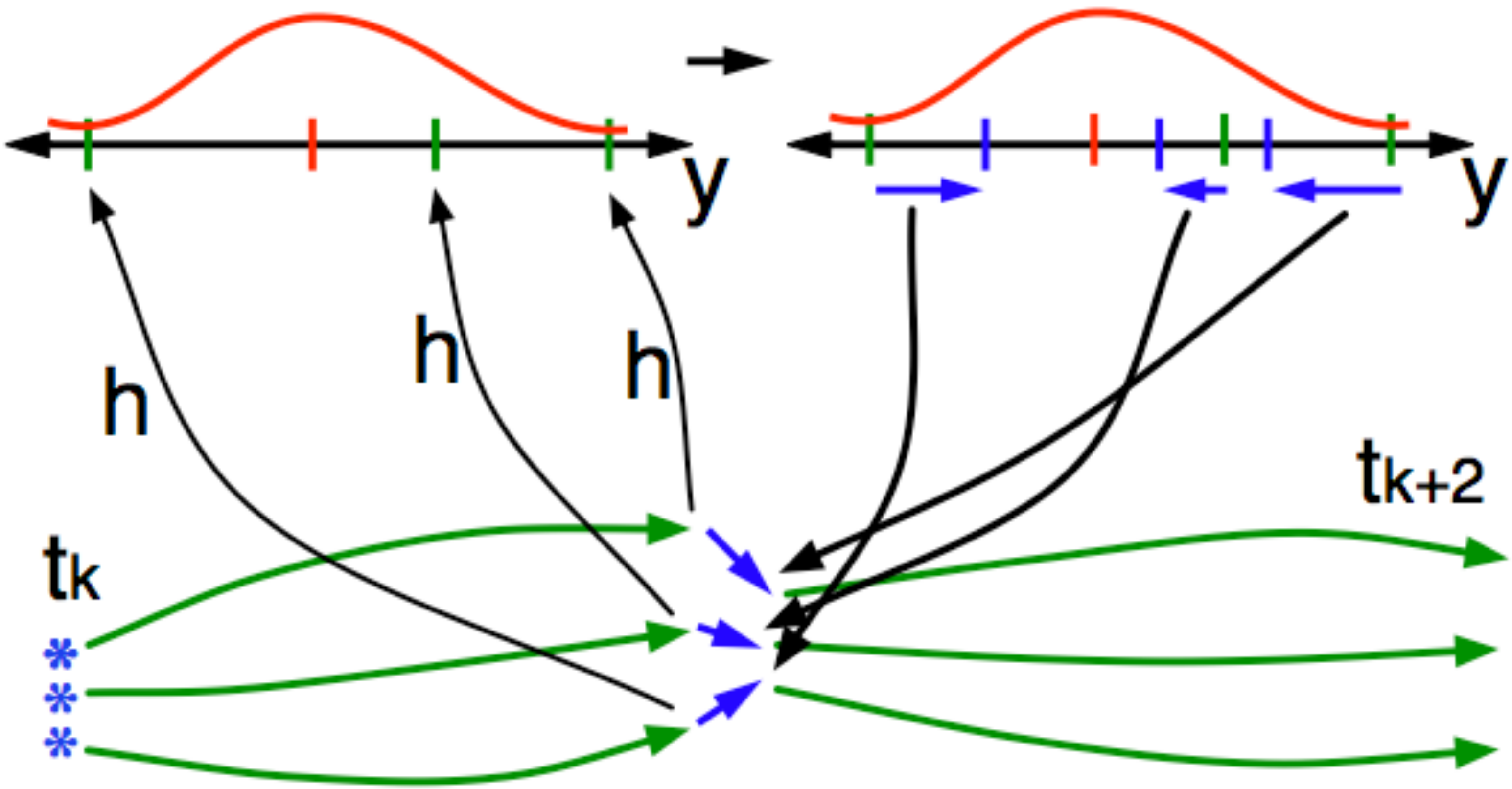


How an Ensemble Filter Works for Geophysical Data Assimilation

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



6. When all ensemble members for each state variable are updated, integrate to time of next observation …



Removing the Kalman from the Ensemble Kalman Filter

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

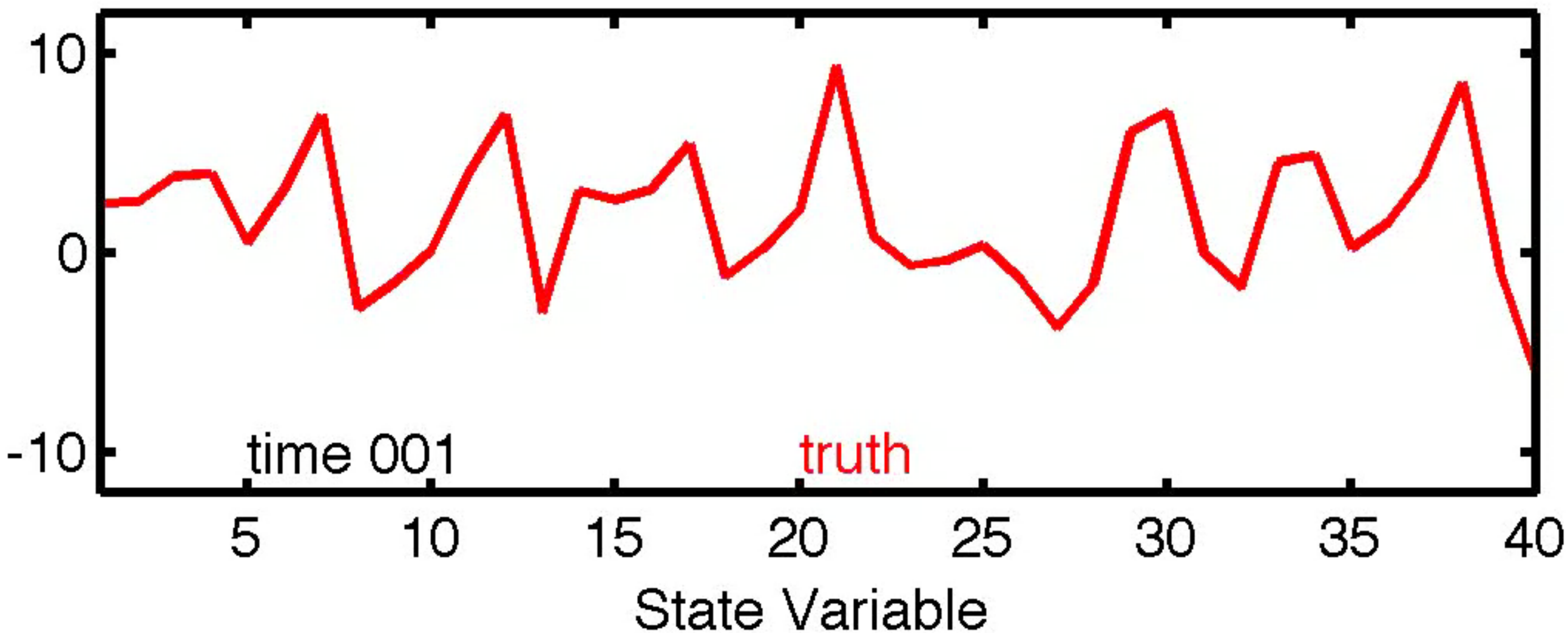
2. No need for linear forward operator.

Ensemble Filter for Lorenz-96 40-Variable Model

40 state variables: X_1, X_2,..., X_{40}.

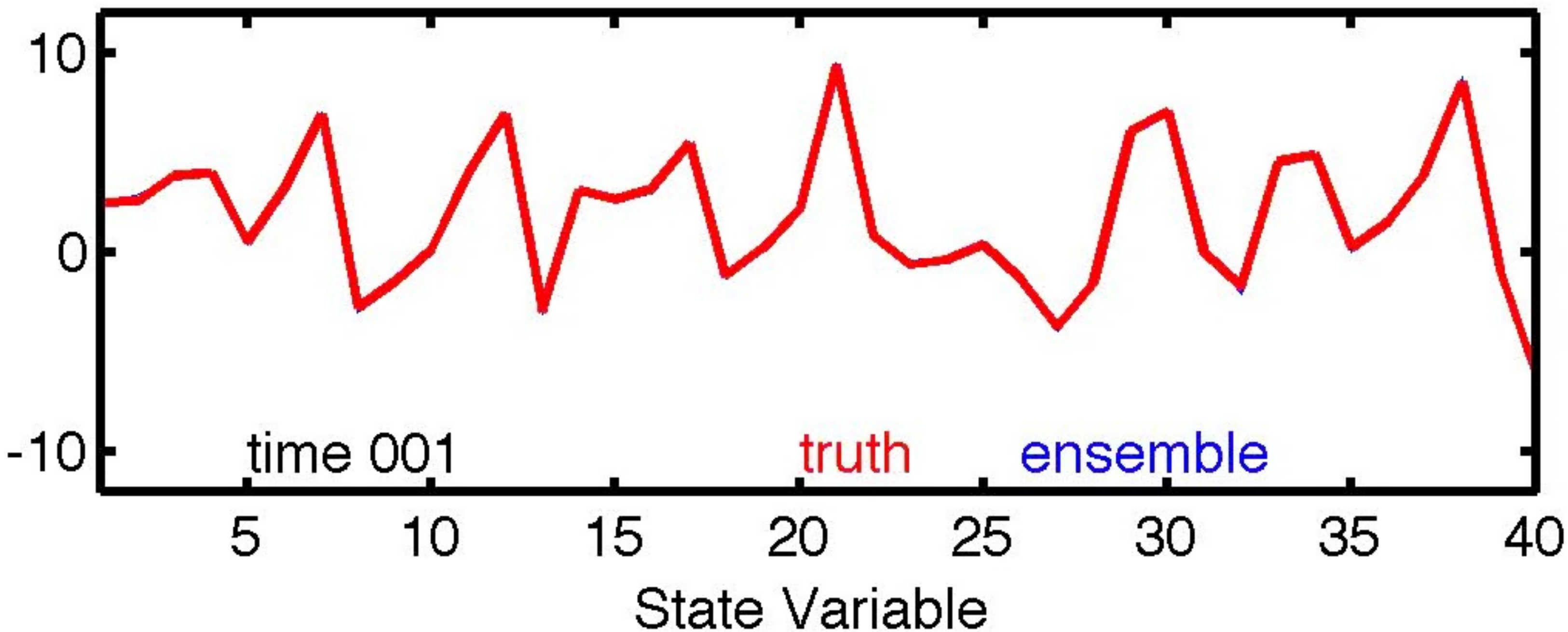
$dX_i / dt = (X_{i+1} - X_{i-2})X_{i-1} - X_i + F$.

Acts ‘something’ like weather around a latitude band.



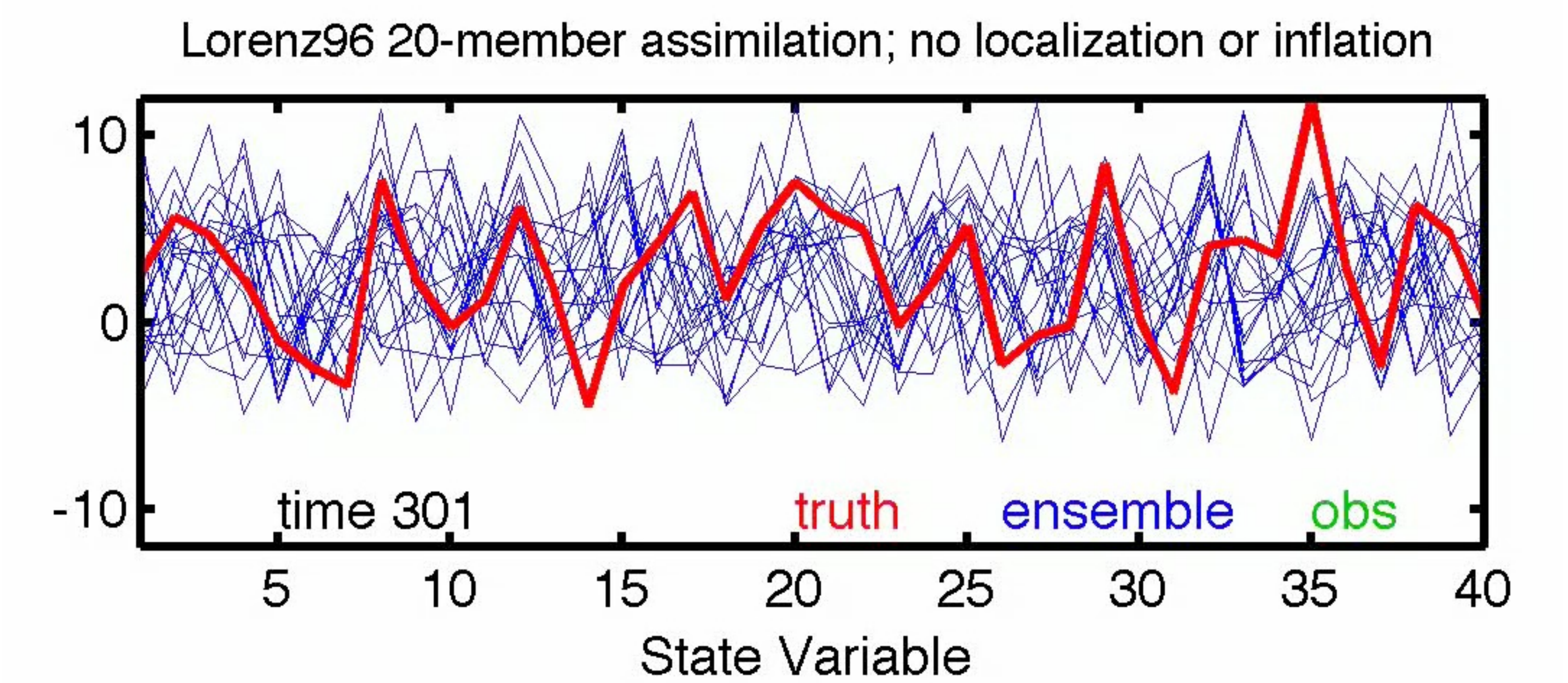
Lorenz-96 is sensitive to small perturbations

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

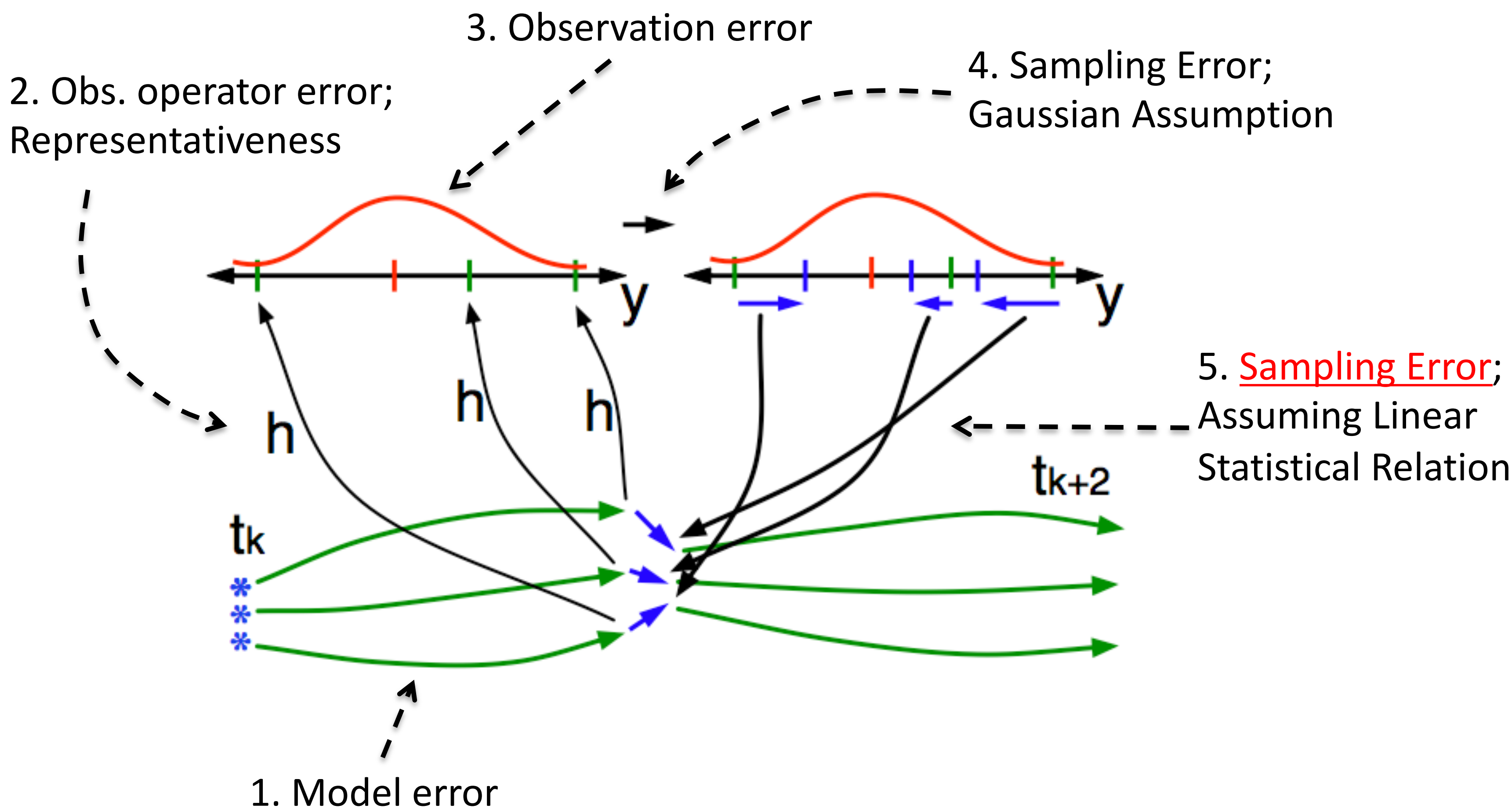


Assimilate 'observations' from 40 random locations.

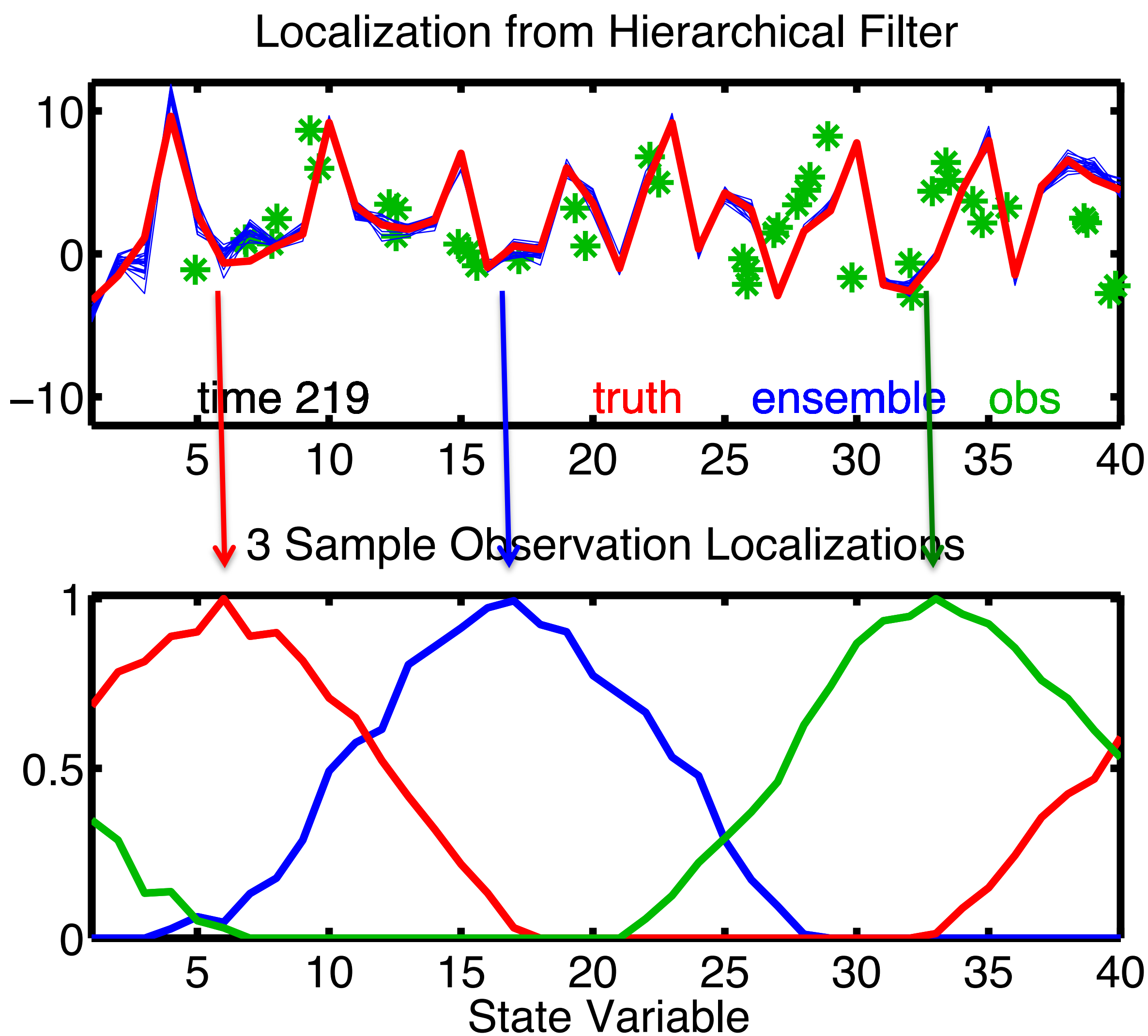
Interpolate truth to station location.
Simulate observational error:
Add random draw from N(0, 16) to each.
Start from 'climatological' 20-member ensemble.



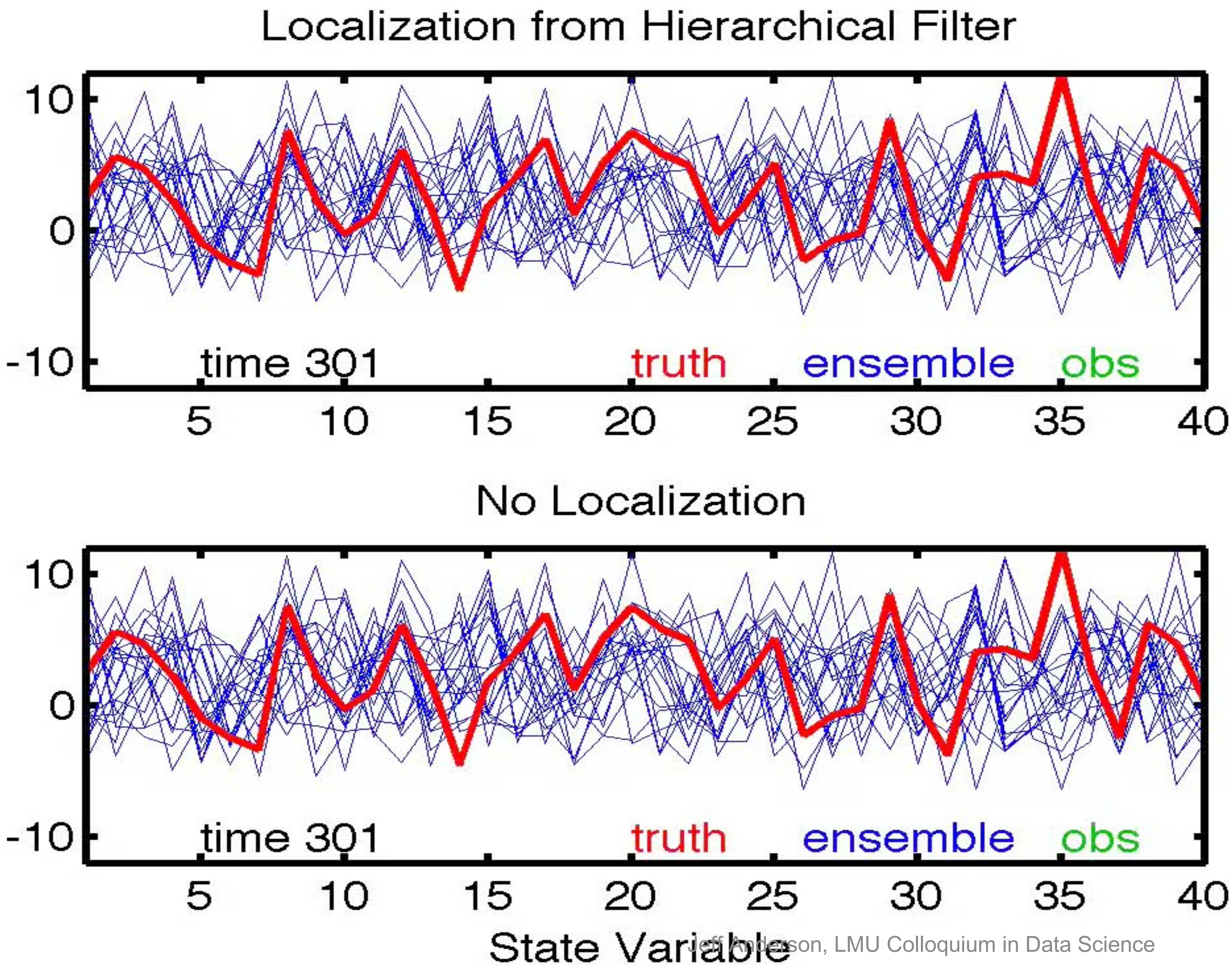
Some Error Sources in Ensemble Filters



Lorenz-96 Assimilation with localization of observation impact



Lorenz-96 Assimilation with localization of observation impact



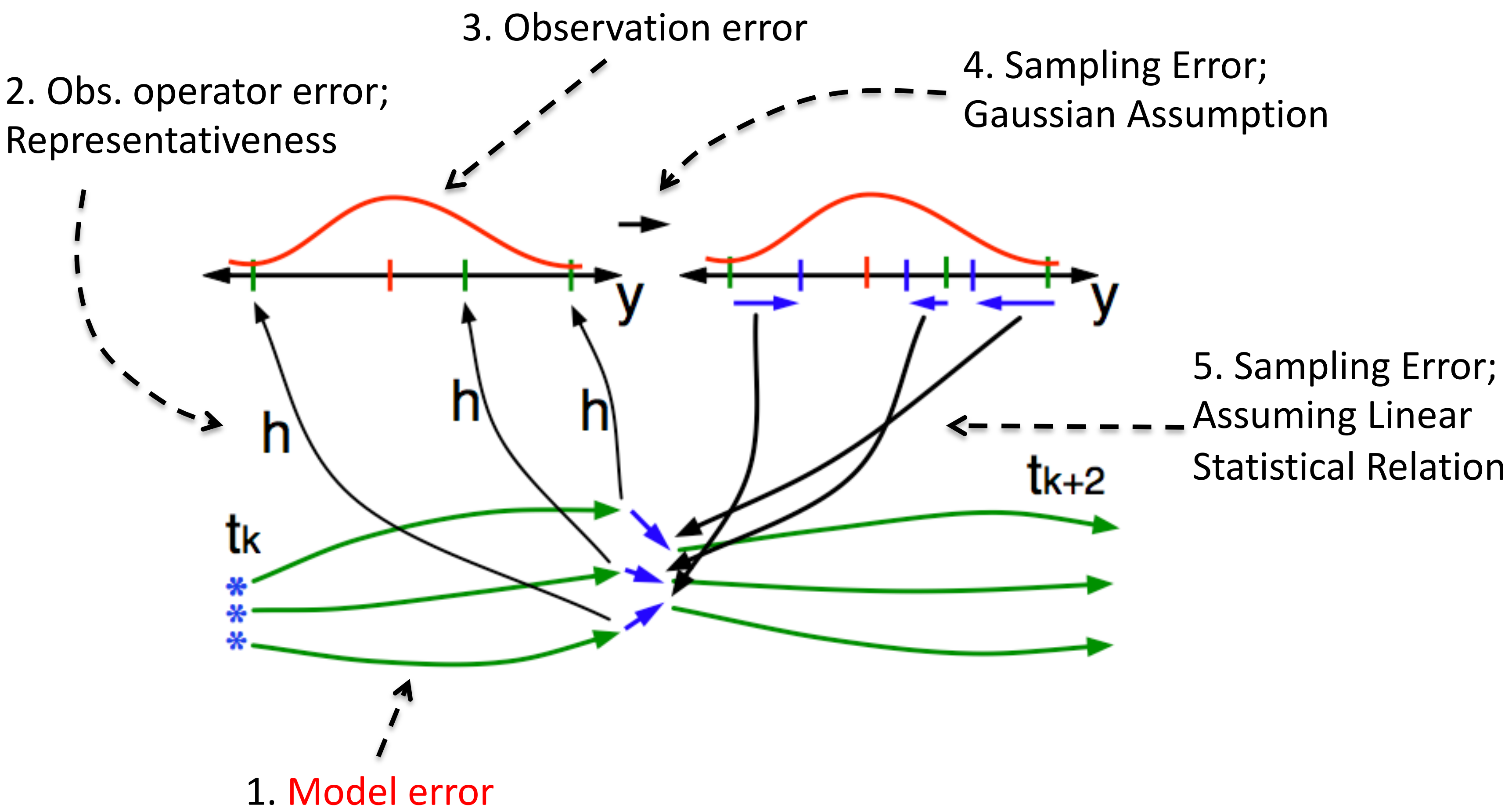
Removing the Kalman from the Ensemble Kalman Filter

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

2. No need for linear forward operator.

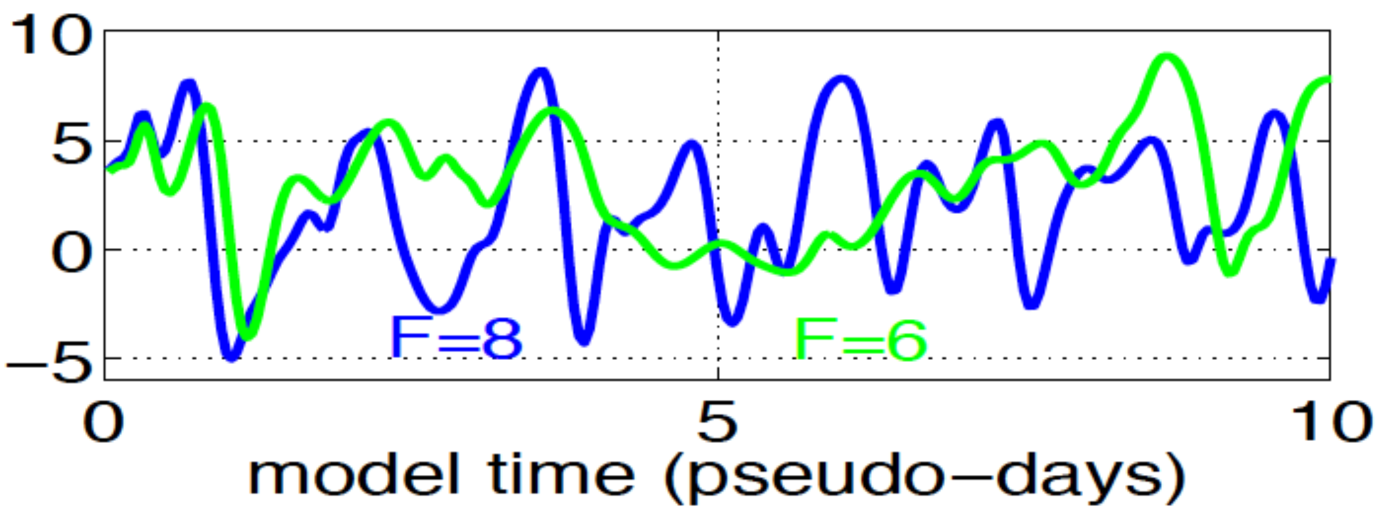
3. No need for unbiased estimate of covariance.

Some Error Sources in Ensemble Filters



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.



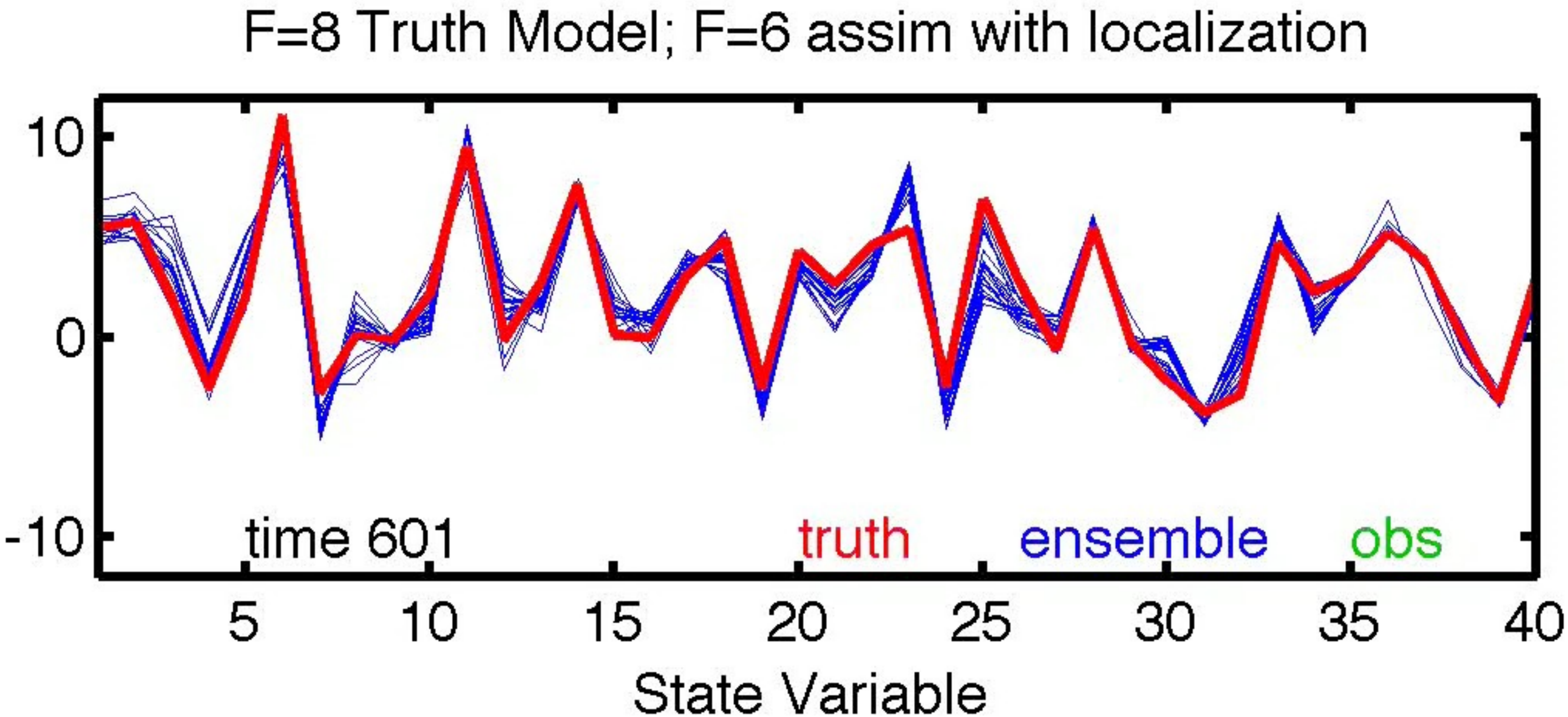
Time evolution for first state variable shown.
Assimilating model quickly diverges from ‘true’ model.

Assimilating in the presence of simulated model error

dXi / dt = (Xi+1 - Xi-2)Xi-1 - Xi + F.

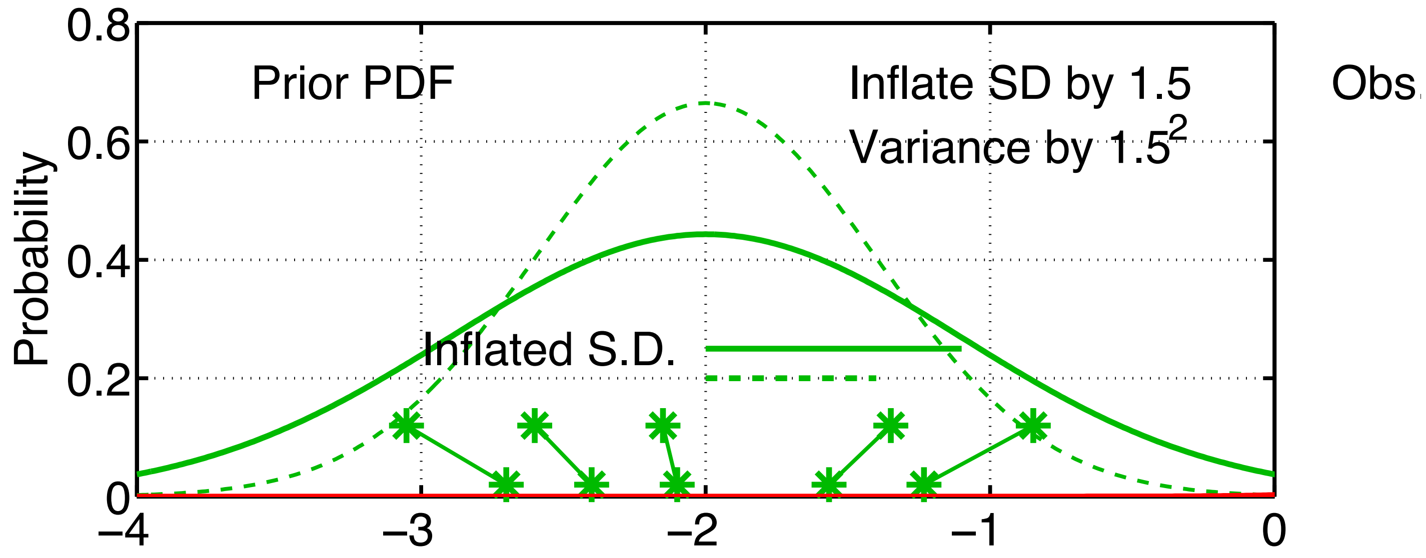
For truth, use F = 8.

In assimilating model, use F = 6.



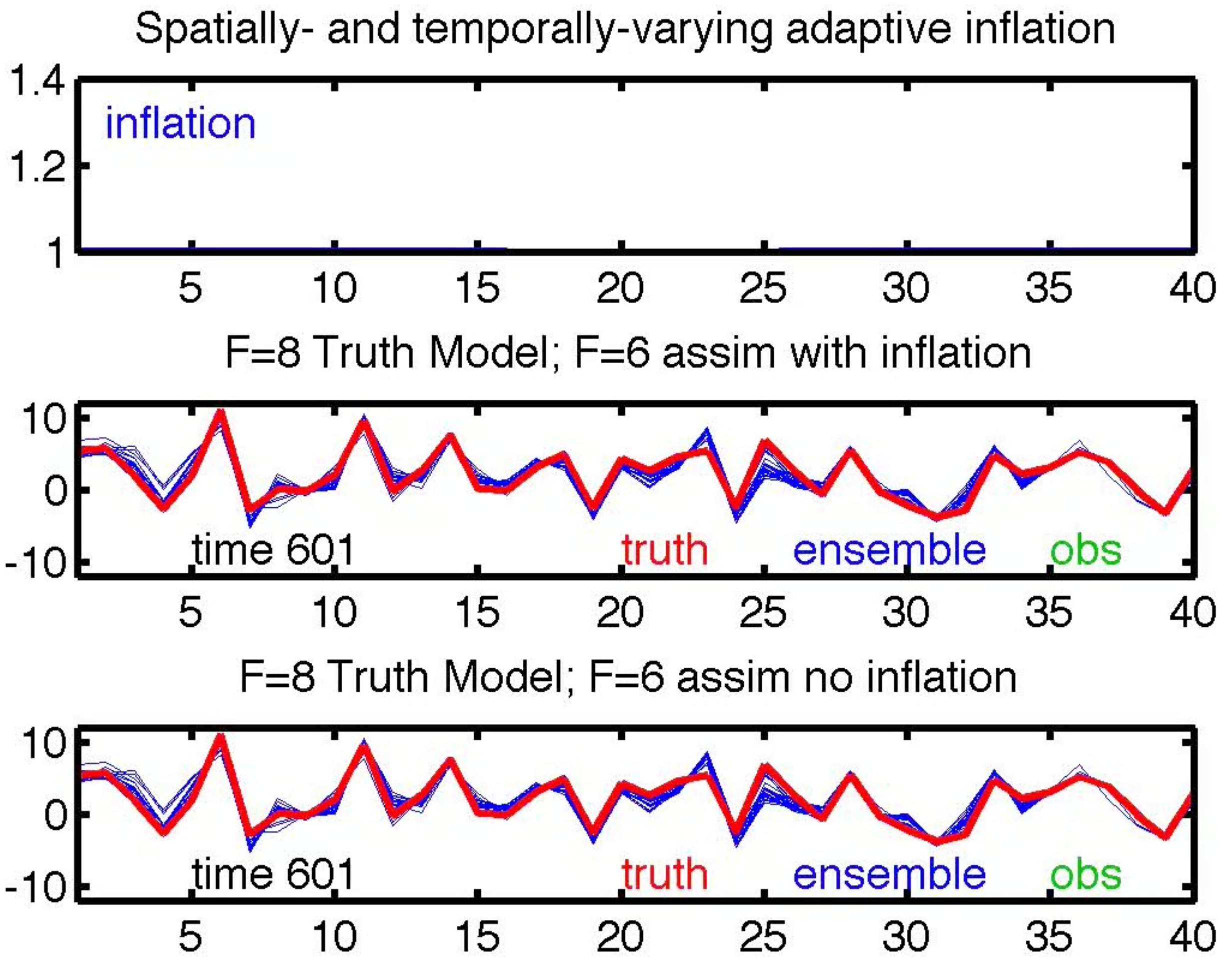
Reduce confidence in prior to deal with model error

Use inflation.
Simply increase prior ensemble variance for each state variable.
Adaptive algorithms use observations to guide this.



Assimilating with Inflation in Presence of Model Error

Inflation is a function of state variable and time.
Automatically selected by adaptive inflation algorithm.



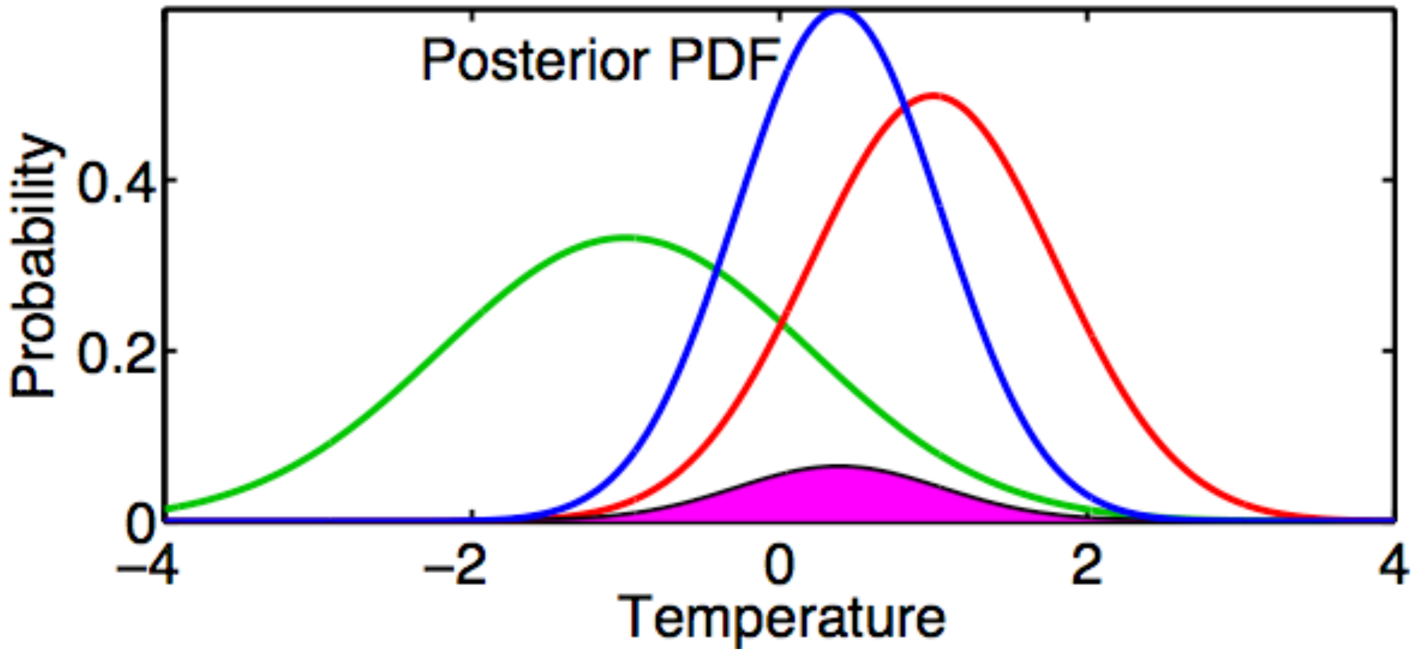
Removing the Kalman from the Ensemble Kalman Filter

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.

Bayes Rule (1D example in 'observation space')

Kalman assimilation algorithms assume Gaussians.
May be okay for quantity like temperature.

$$P\left(\mathbf{x}_{t_k}|\mathbf{Y}_{t_k}\right) = \frac{P(\mathbf{y}_k|\mathbf{x})P\left(\mathbf{x}_{t_k}|\mathbf{Y}_{t_{k-1}}\right)}{Normalization}$$

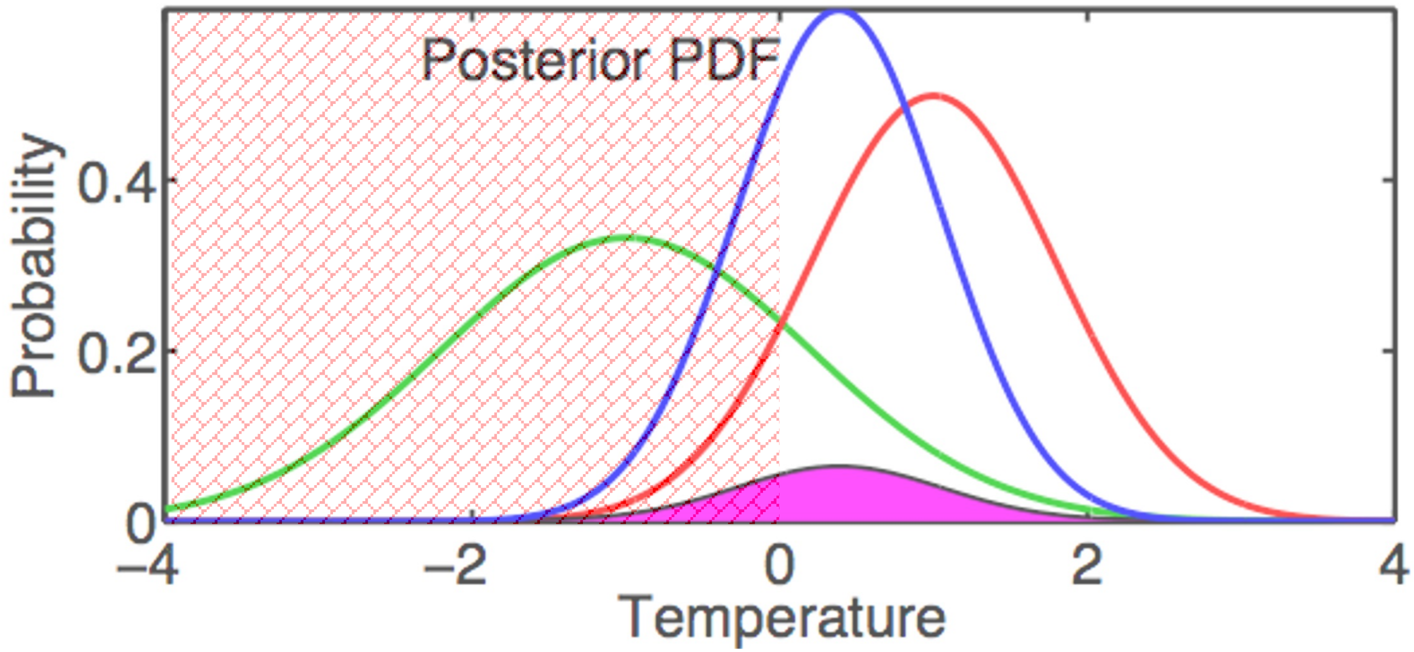


Bayes Rule (1D example in 'observation space')

Kalman assimilation algorithms assume Gaussians.

Tracer concentration is bounded. Gaussian a poor choice.

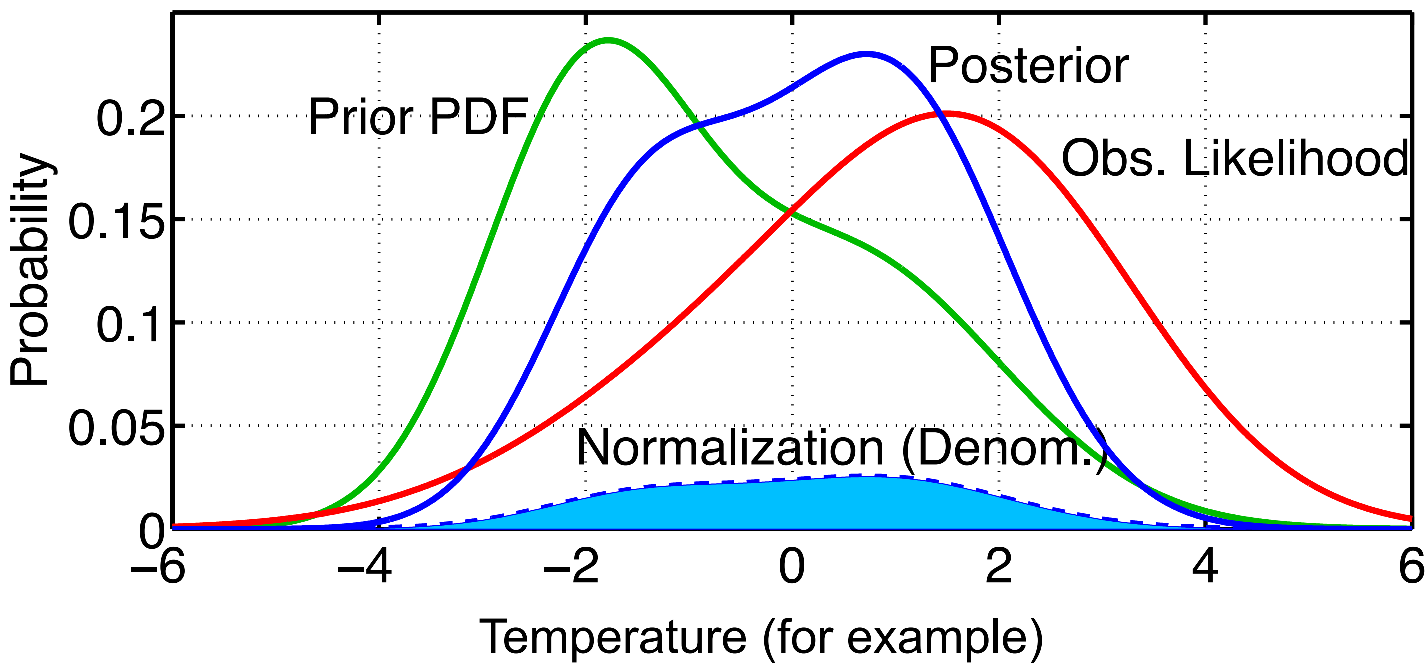
$$P\left(\mathbf{x}_{t_k}|\mathbf{Y}_{t_k}\right) = \frac{P(\mathbf{y}_k|\mathbf{x})P\left(\mathbf{x}_{t_k}|\mathbf{Y}_{t_{k-1}}\right)}{Normalization}$$



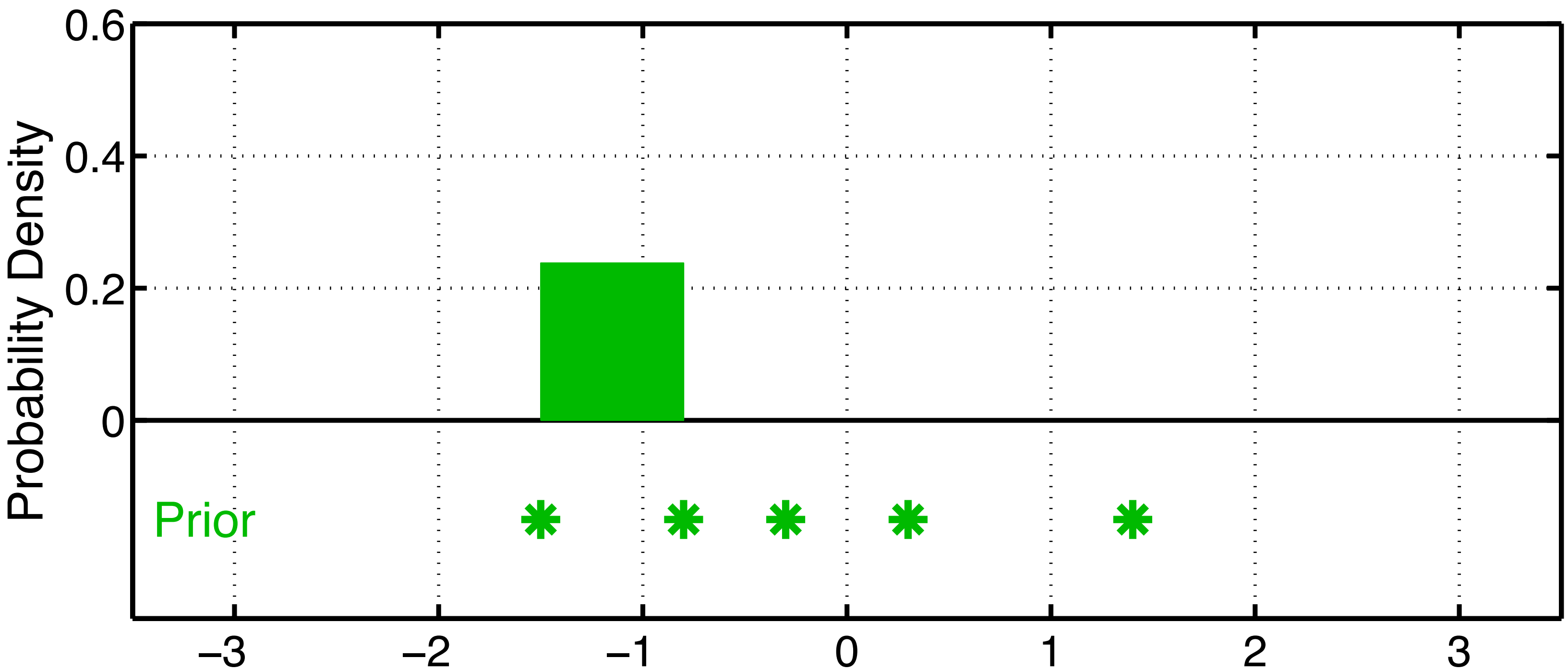
Bayes Rule (1D example in 'observation space')

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

$$P\left(\mathbf{x}_{t_k}|\mathbf{Y}_{t_k}\right) = \frac{P(\mathbf{y}_k|\mathbf{x})P\left(\mathbf{x}_{t_k}|\mathbf{Y}_{t_{k-1}}\right)}{Normalization}$$



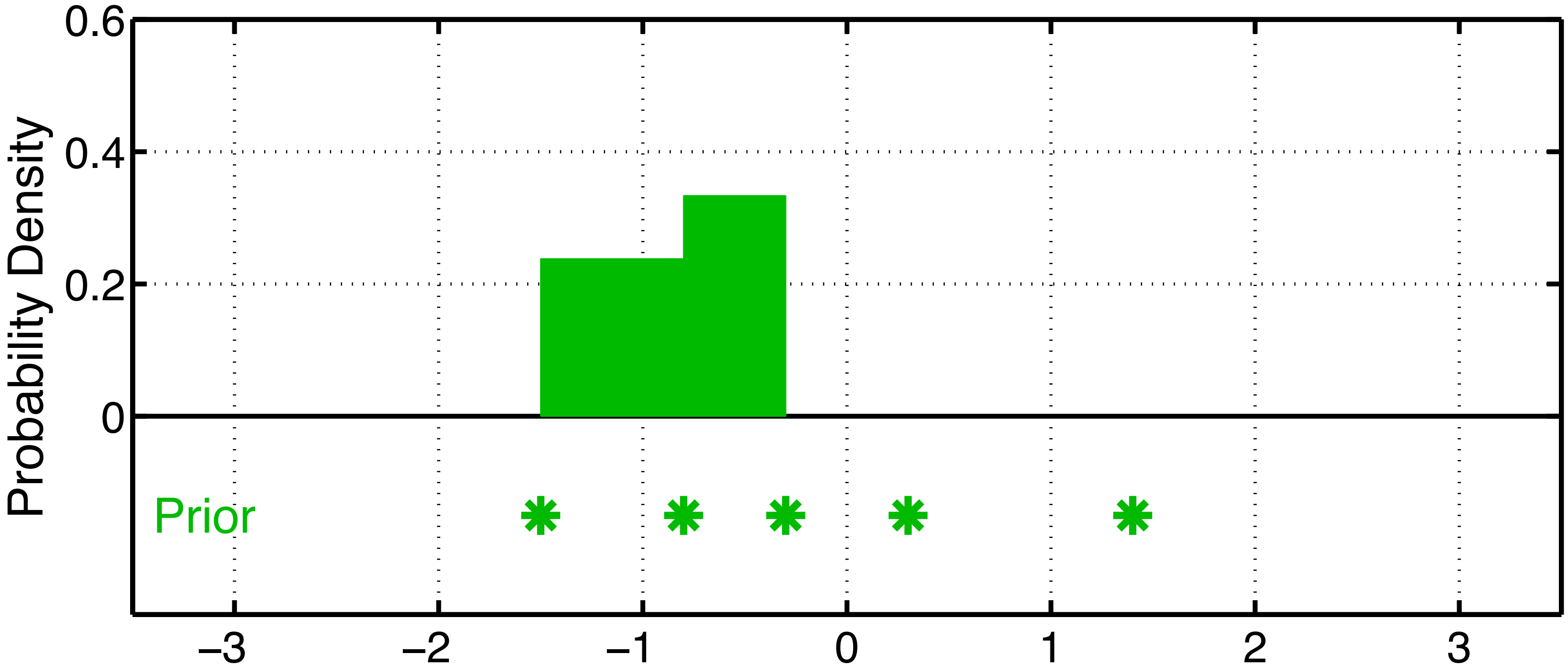
Observation-Space Rank Histogram Filter



Step 1: Get continuous prior distribution density.

- Place (ens_size + 1)$^{-1}$ mass between adjacent ensemble members.
- Reminiscent of rank histogram evaluation method.

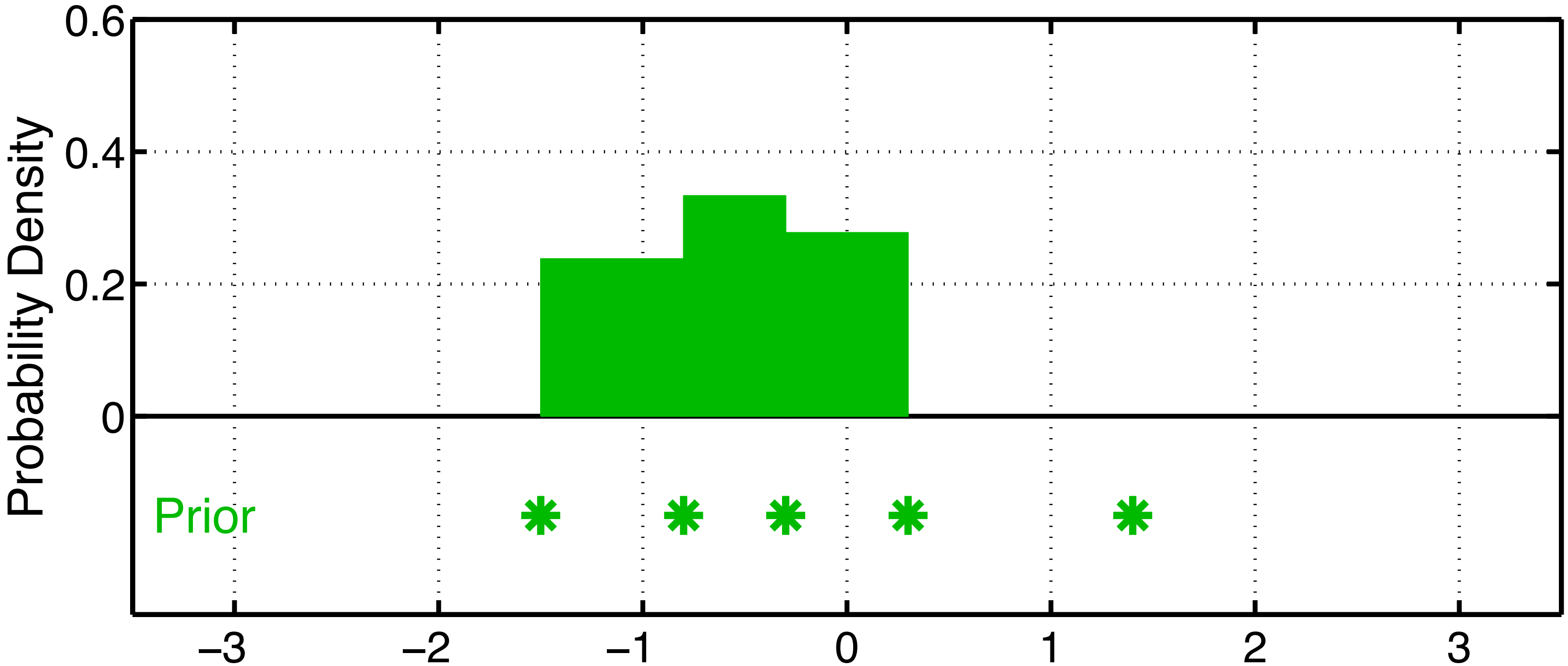
Observation-Space Rank Histogram Filter



Step 1: Get continuous prior distribution density.

- Place (ens_size + 1)$^{-1}$ mass between adjacent ensemble members.
- Reminiscent of rank histogram evaluation method.

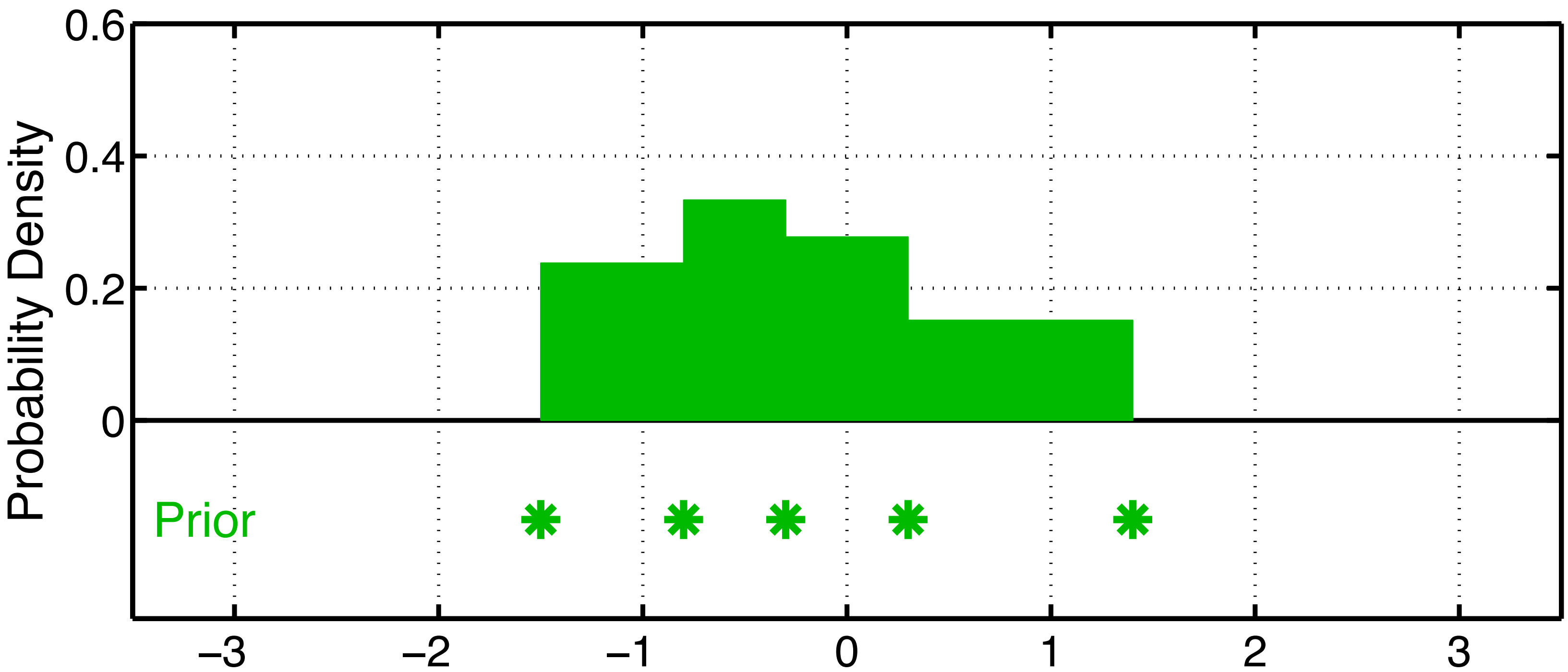
Observation-Space Rank Histogram Filter



Step 1: Get continuous prior distribution density.

- Place (ens_size + 1)$^{-1}$ mass between adjacent ensemble members.
- Reminiscent of rank histogram evaluation method.

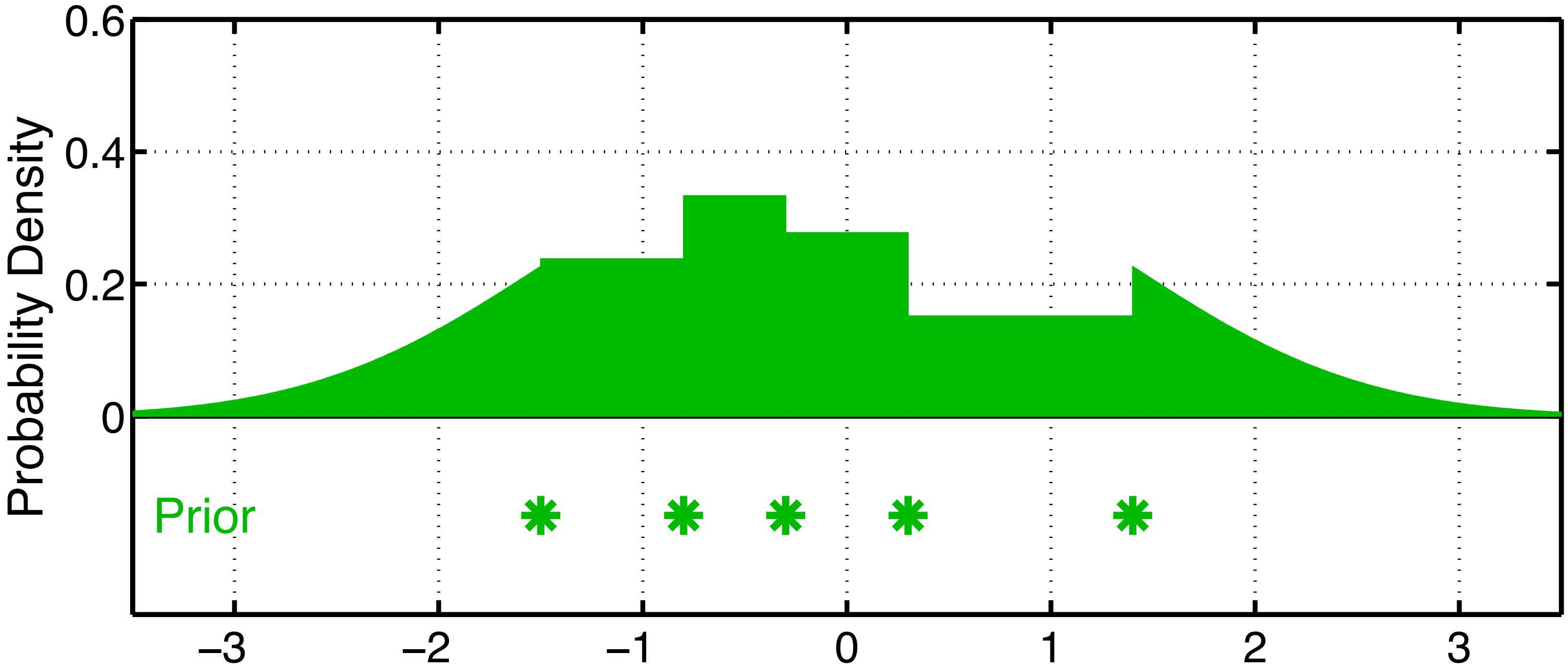
Observation-Space Rank Histogram Filter



Step 1: Get continuous prior distribution density.

- Place $(\text{ens\_size} + 1)^{-1}$ mass between adjacent ensemble members.
- Reminiscent of rank histogram evaluation method.

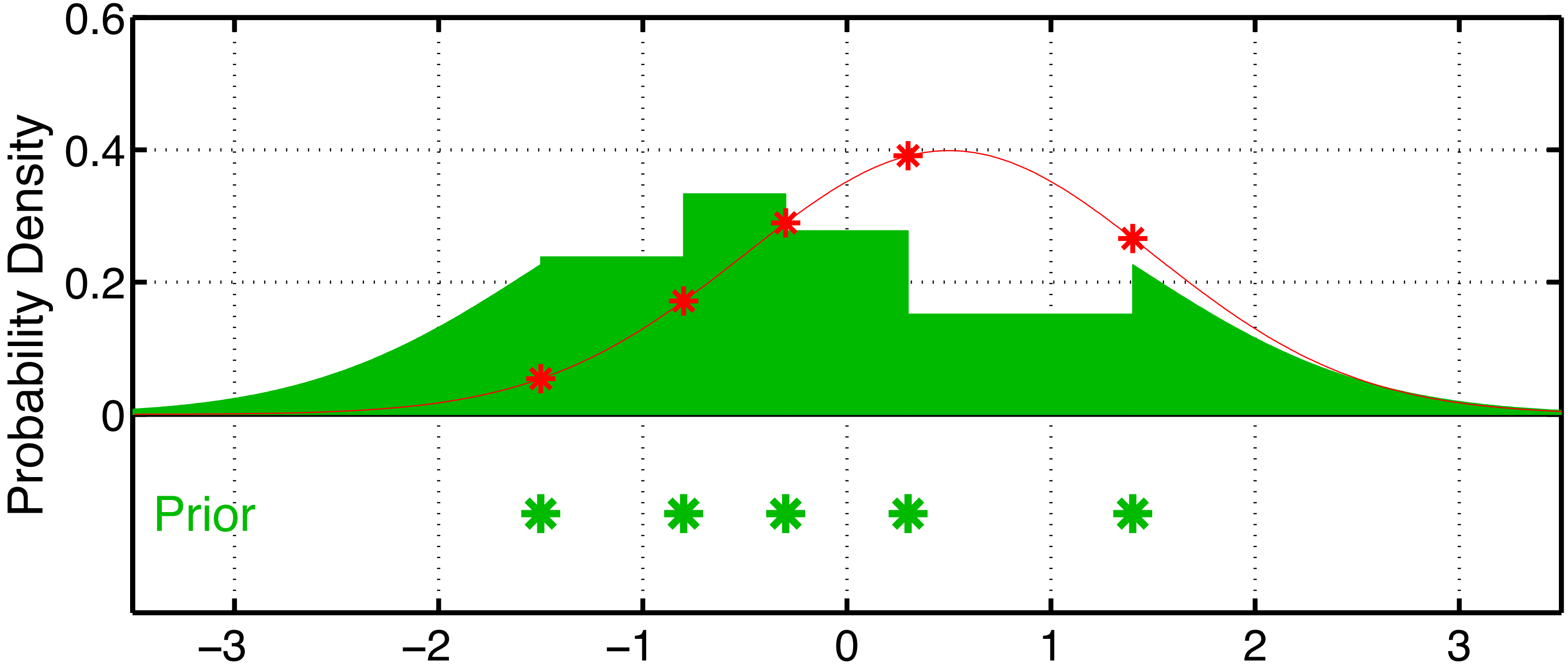
Observation-Space Rank Histogram Filter



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.*

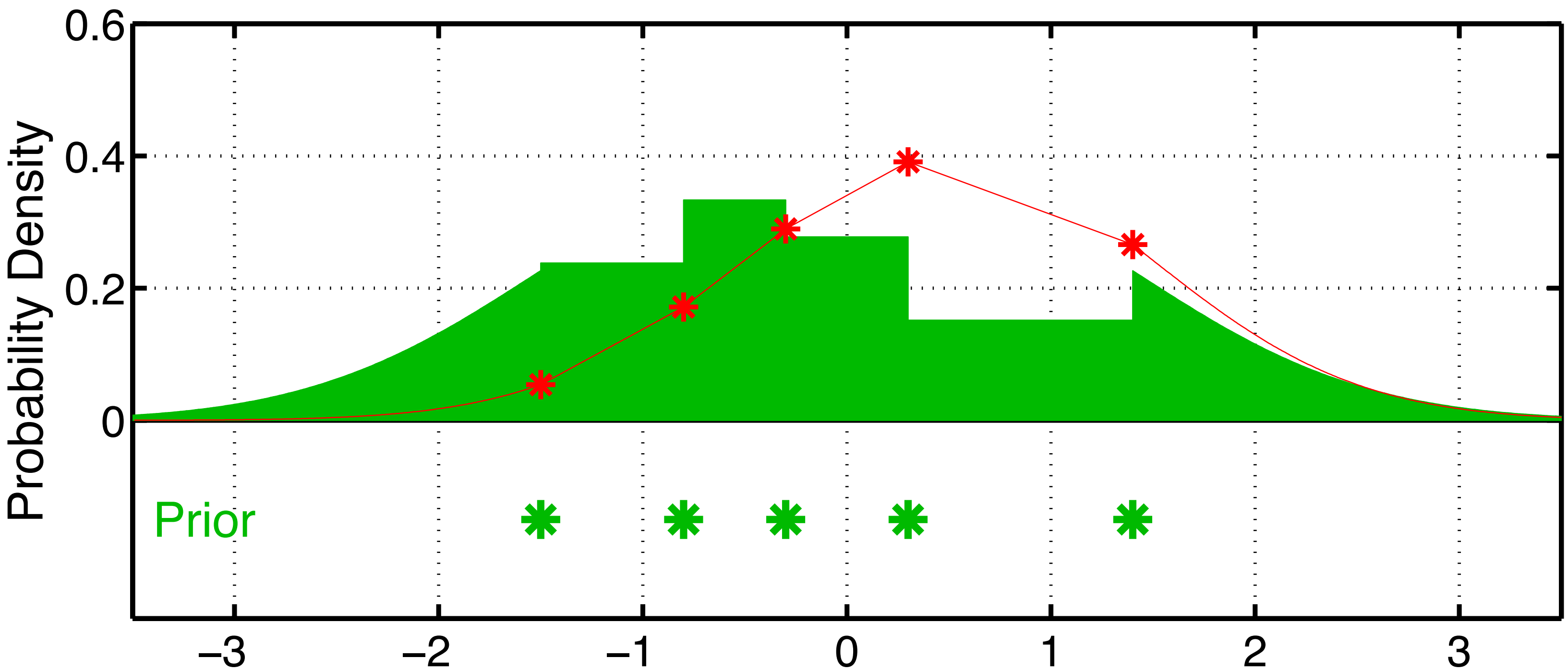
Observation-Space Rank Histogram Filter



Step 2: Use likelihood to compute weight for each ensemble member.

- Analogous to classical particle filter.
- Can be extended to non-gaussian obs. likelihoods.

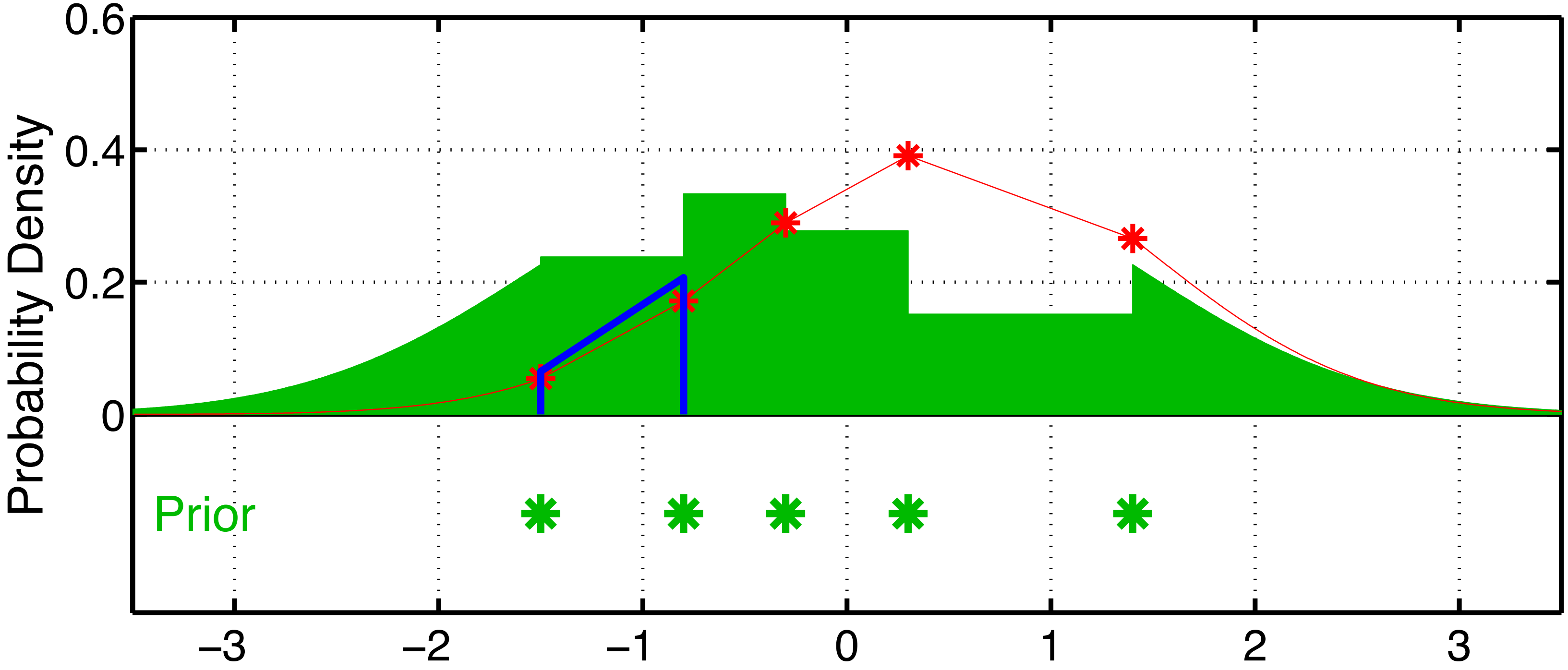
Observation-Space Rank Histogram Filter



Step 2: Use likelihood to compute weight for each ensemble member.

- Can approximate interior likelihood with linear fit; for efficiency.

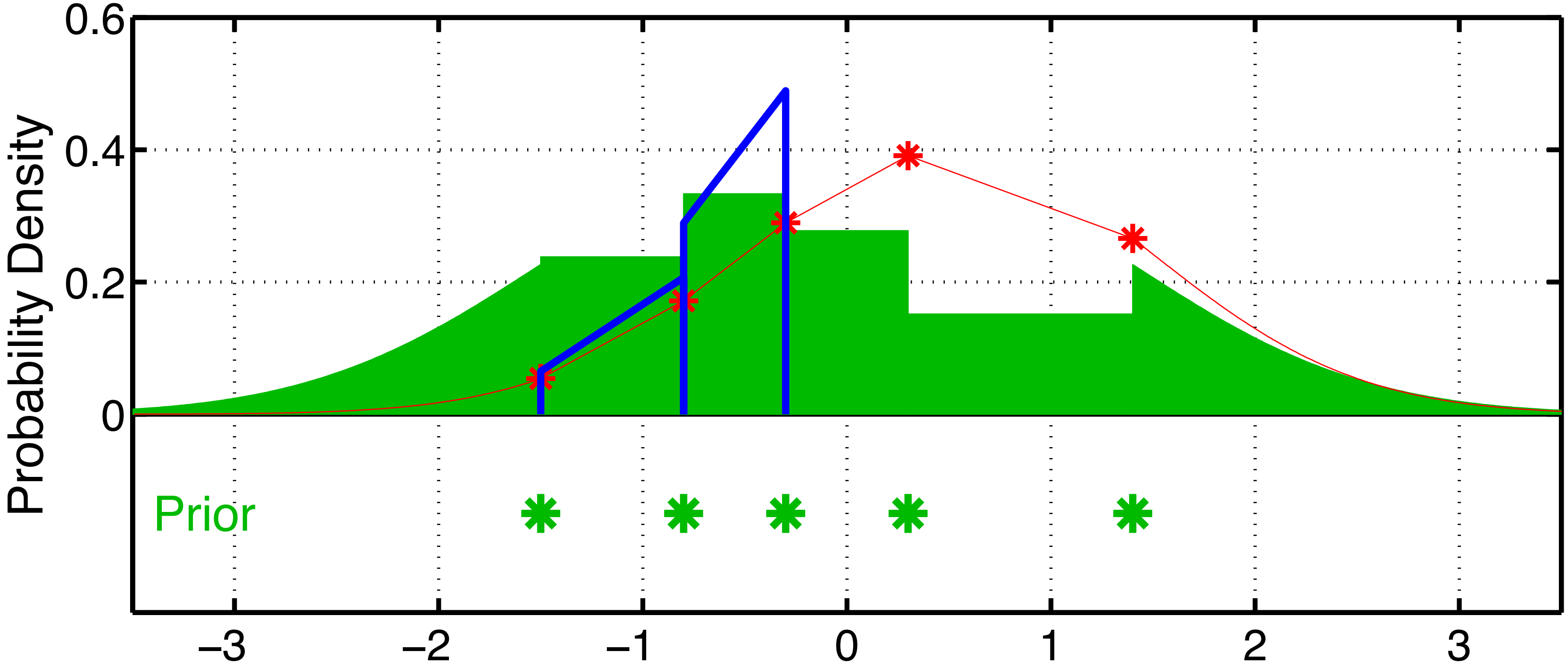
Observation-Space Rank Histogram Filter



Step 3: Compute continuous posterior distribution.

- Approximate likelihood with trapezoidal quadrature, take product.
  (Displayed product normalized to make posterior a PDF).

Observation-Space Rank Histogram Filter

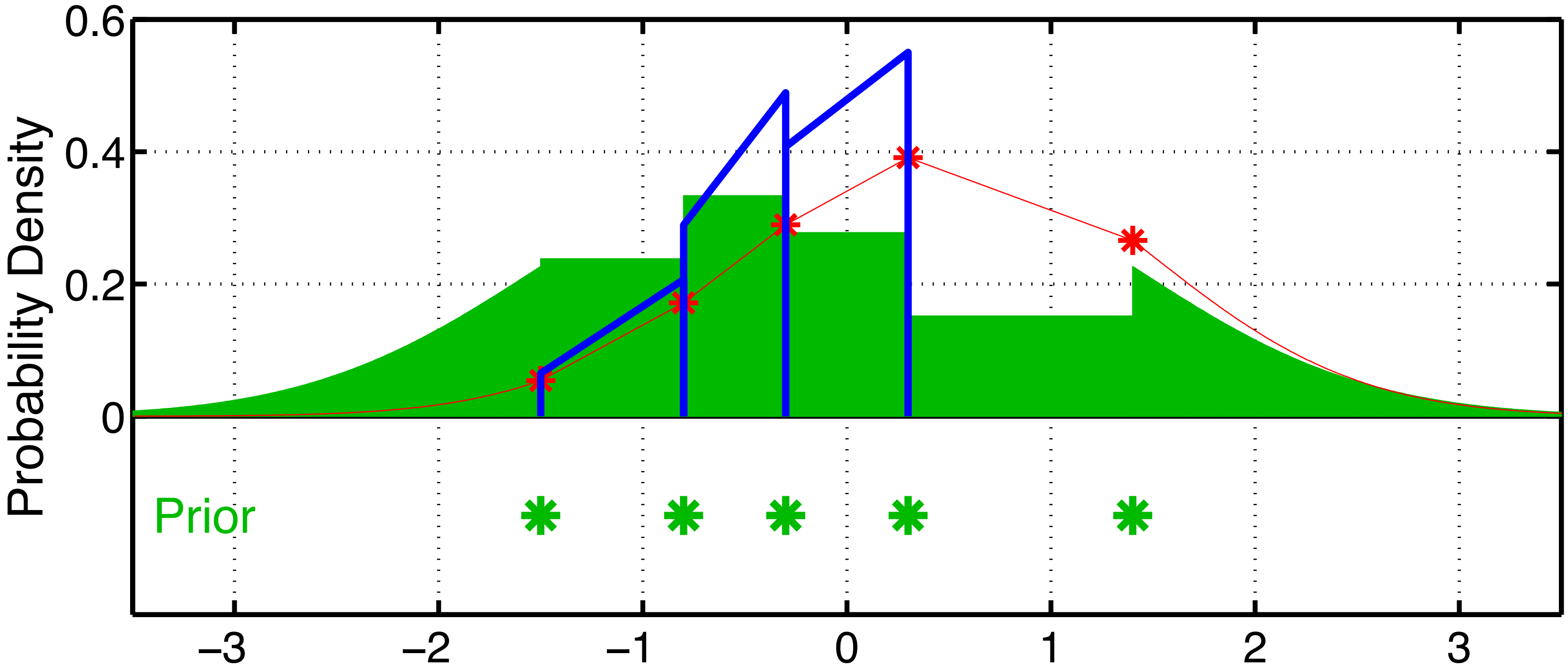


Step 3: Compute continuous posterior distribution.

- Approximate likelihood with trapezoidal quadrature, take product.

  (Displayed product normalized to make posterior a PDF).

Observation-Space Rank Histogram Filter

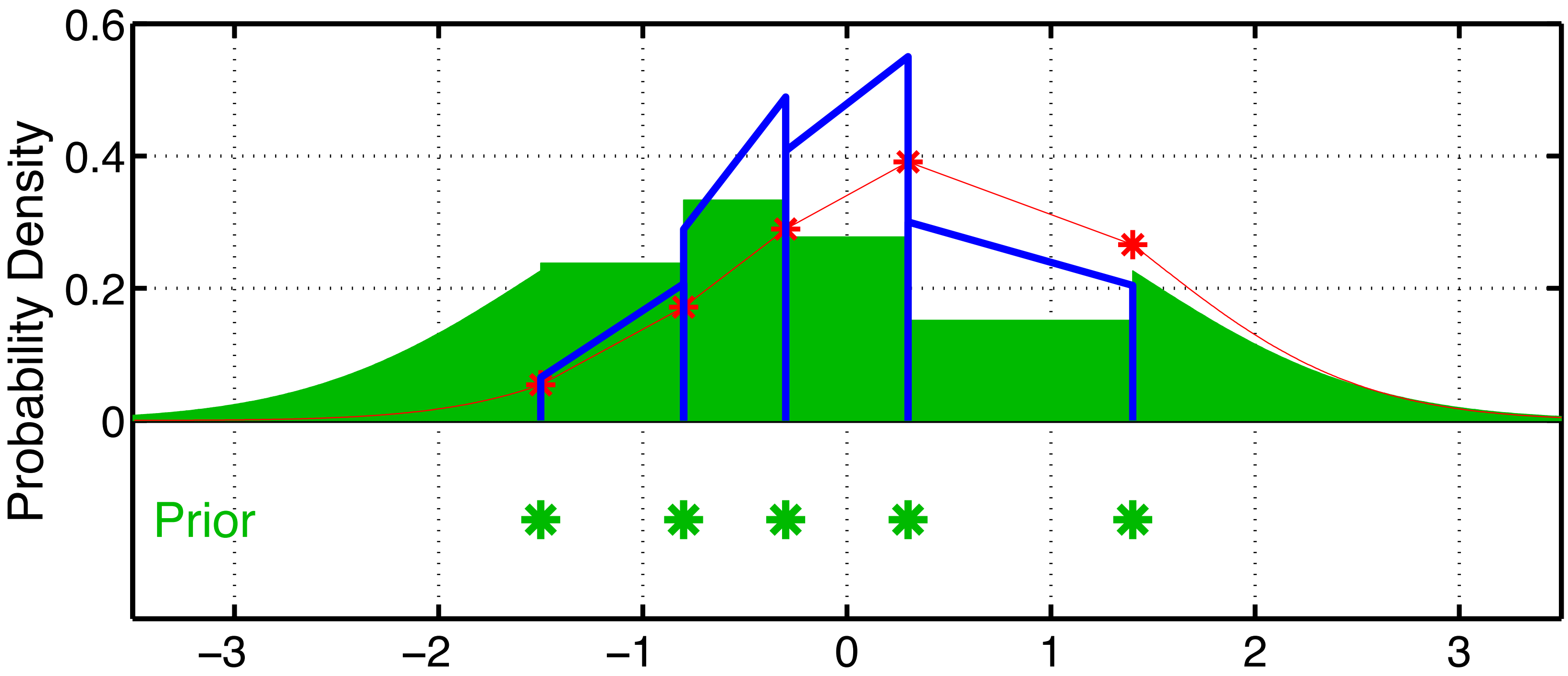


Step 3: Compute continuous posterior distribution.

- Approximate likelihood with trapezoidal quadrature, take product.

  (Displayed product normalized to make posterior a PDF).

Observation-Space Rank Histogram Filter

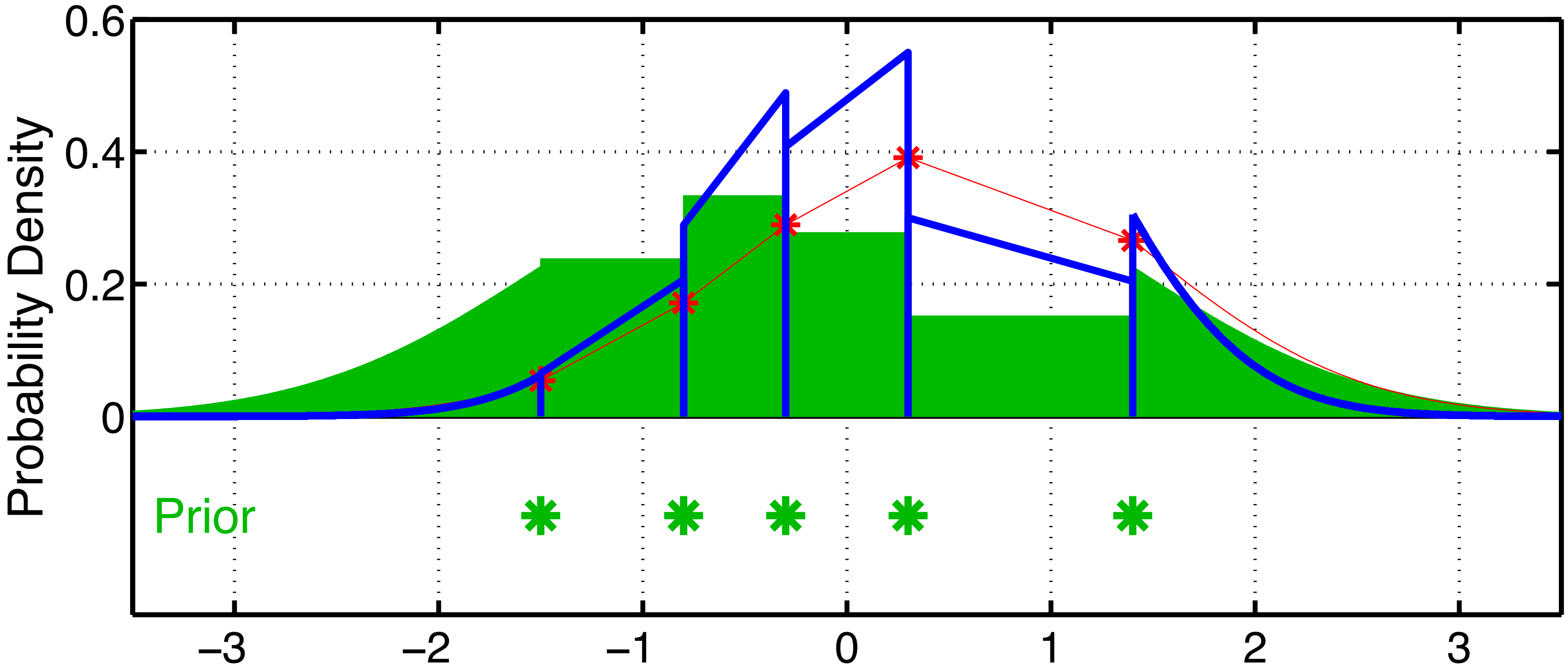


Step 3: Compute continuous posterior distribution.

- Approximate likelihood with trapezoidal quadrature, take product.

  (Displayed product normalized to make posterior a PDF).

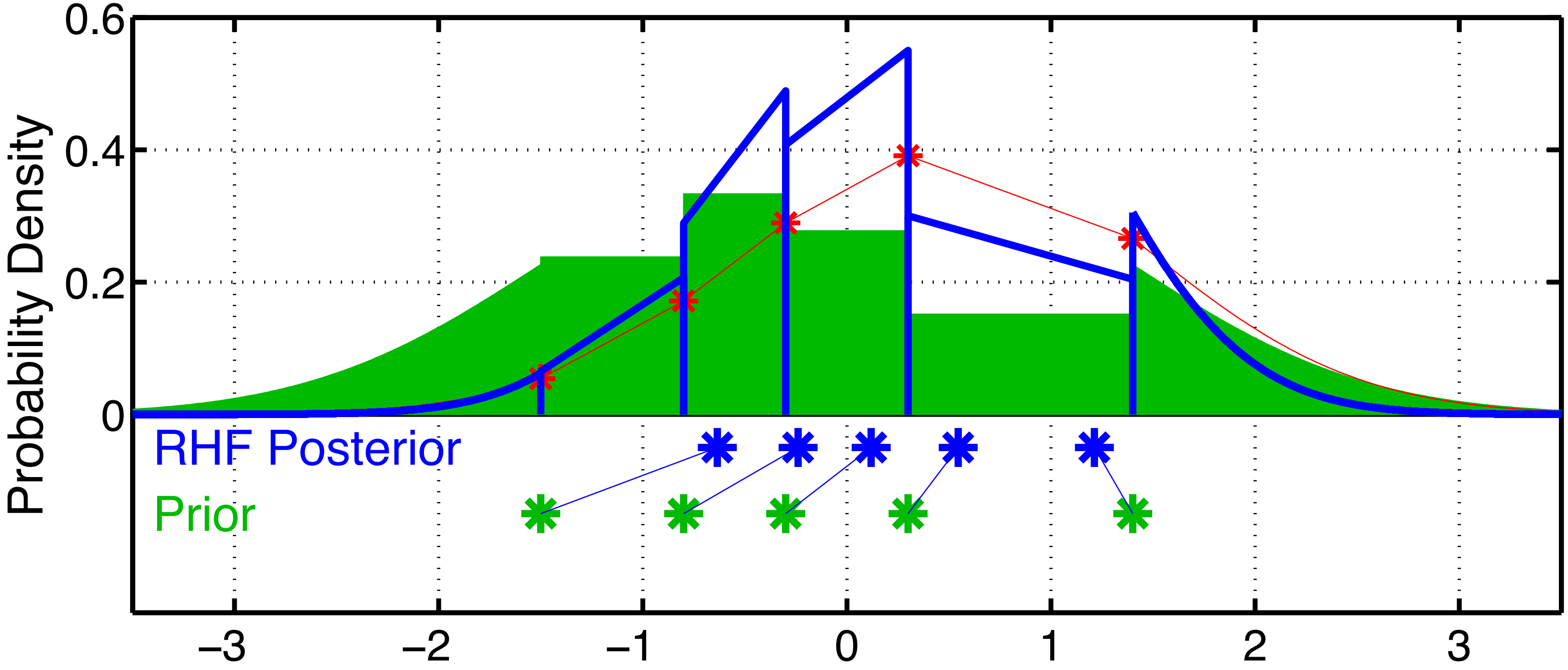
Observation-Space Rank Histogram Filter



Step 3: Compute continuous posterior distribution.

- Product of prior gaussian kernel with likelihood for tails.
- Easy for gaussian likelihood.

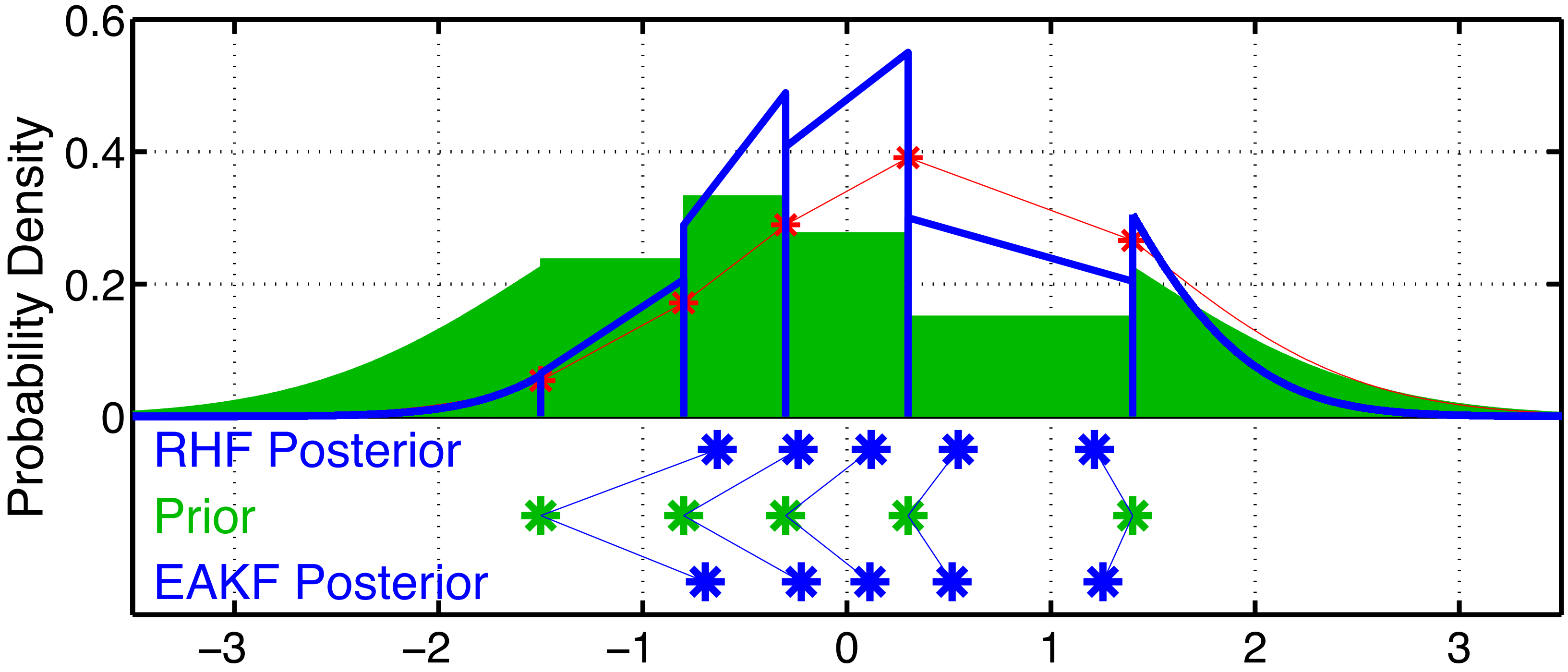
Observation-Space Rank Histogram Filter



Step 4: Compute posterior ensemble members:

- $(\text{ens\_size} + 1)^{-1}$ of posterior mass between each ensemble pair.
- $(\text{ens\_size} + 1)^{-1}$ in each tail.

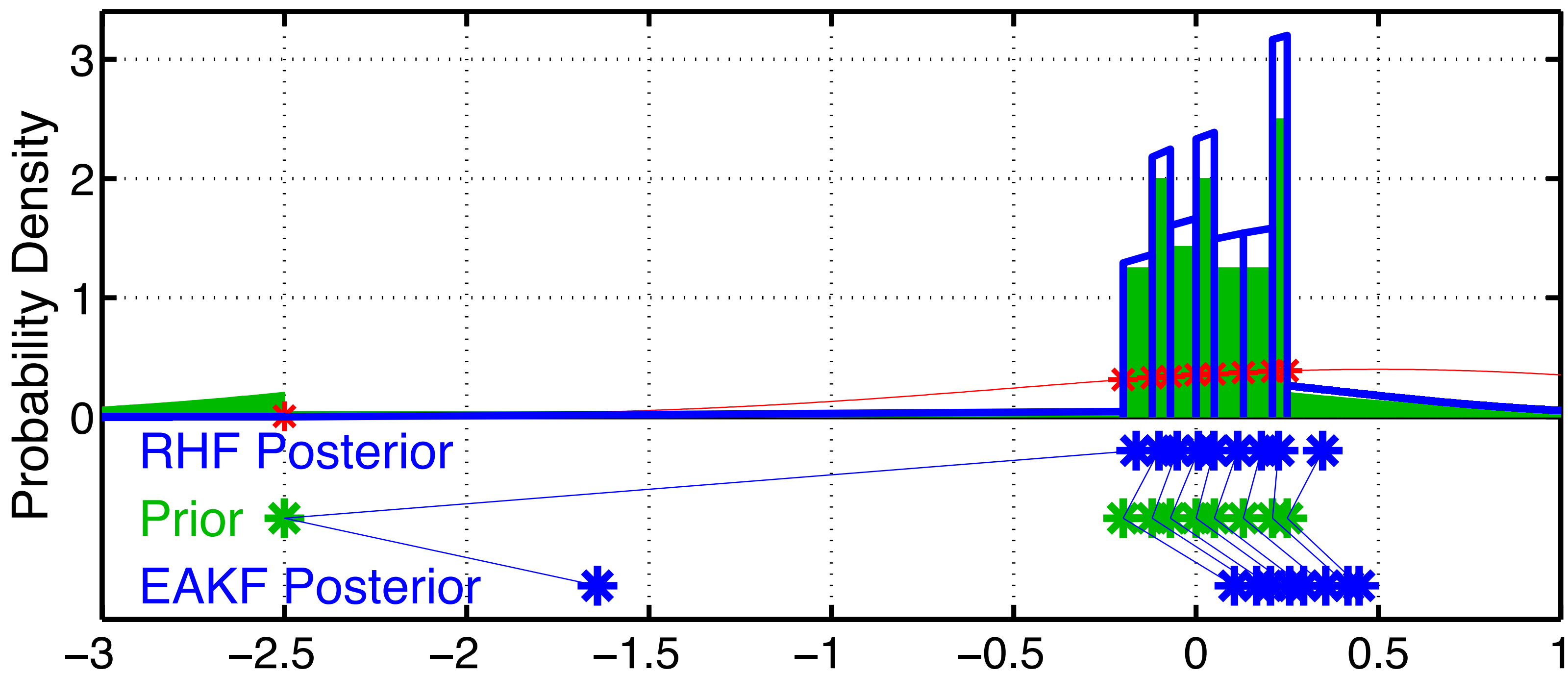
Observation-Space Rank Histogram Filter



Compare to standard Ensemble Adjustment Filter (EAKF).

Nearly gaussian case, differences are small.

Observation-Space Rank Histogram Filter



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.

Removing the Kalman from the Ensemble Kalman Filter

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

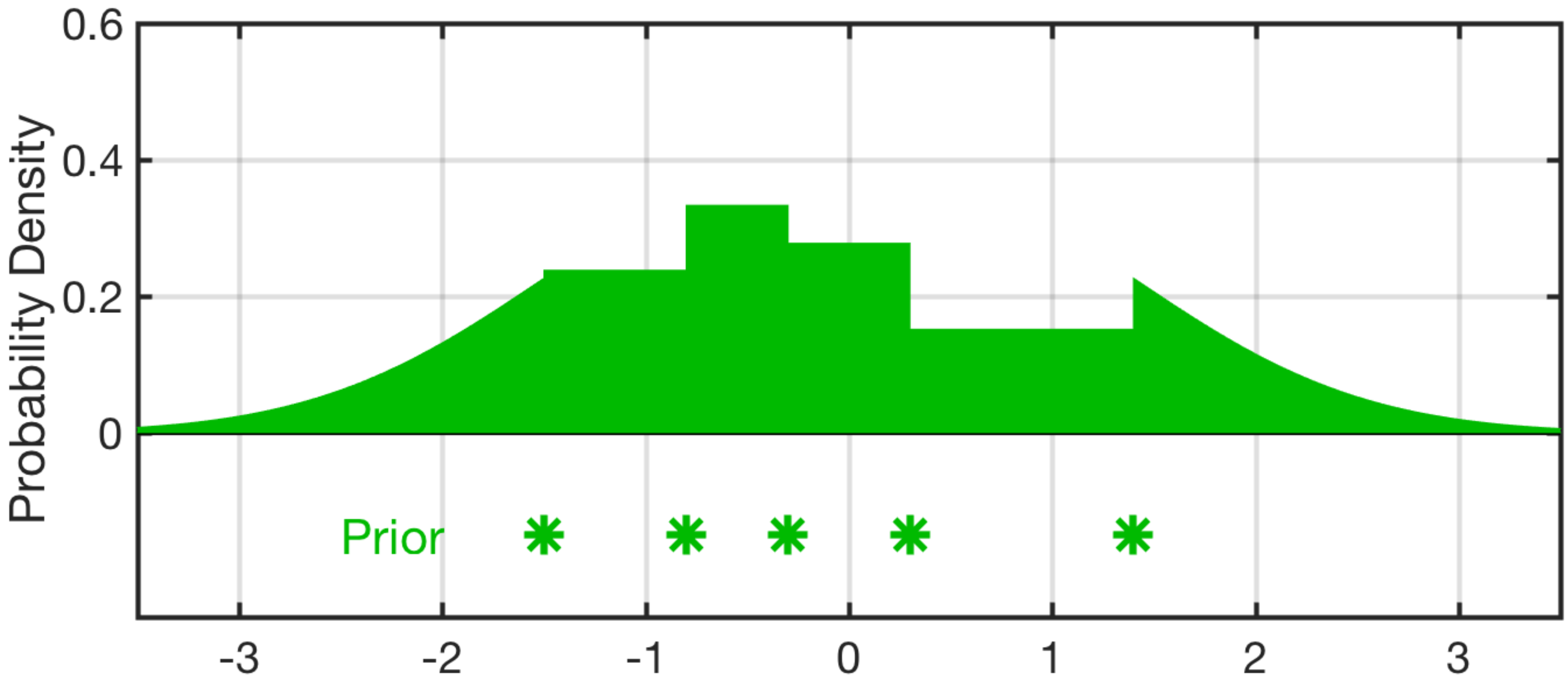
2. No need for linear forward operator.

3. No need for unbiased estimate of covariance.

4. No need for unbiased model prior.

5. (Almost) no Gaussian assumed for prior.

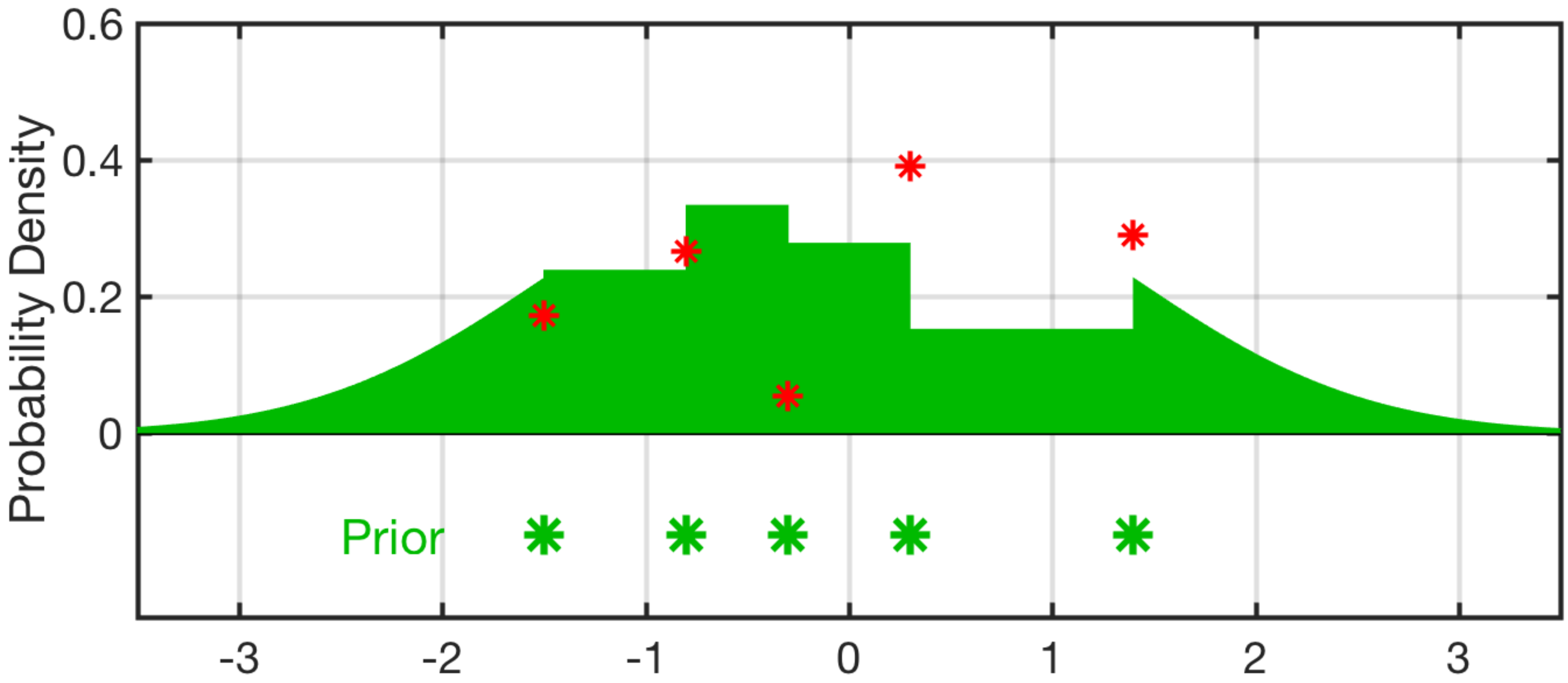
Rank Histogram Filter for State Marginals



Step 1: Get continuous prior distribution density (same).

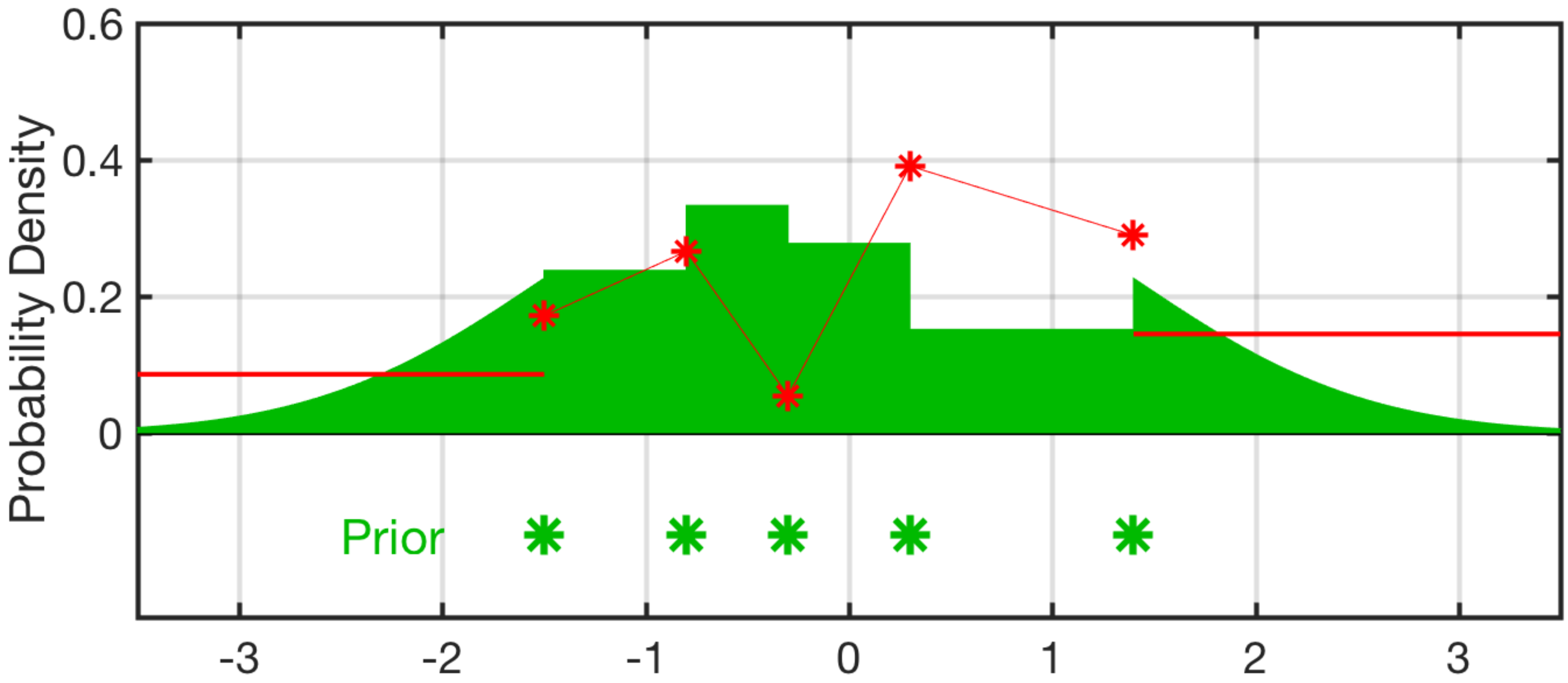
- Partial gaussian kernels on tails, N(*tail_mean*, *ens_sd*).
- *tail_mean selected so that (ens_size + 1)$^{-1}$ mass is in tail.*

Rank Histogram Filter for State Marginals



Step 2: Use likelihood to compute weight for each ensemble member (same).

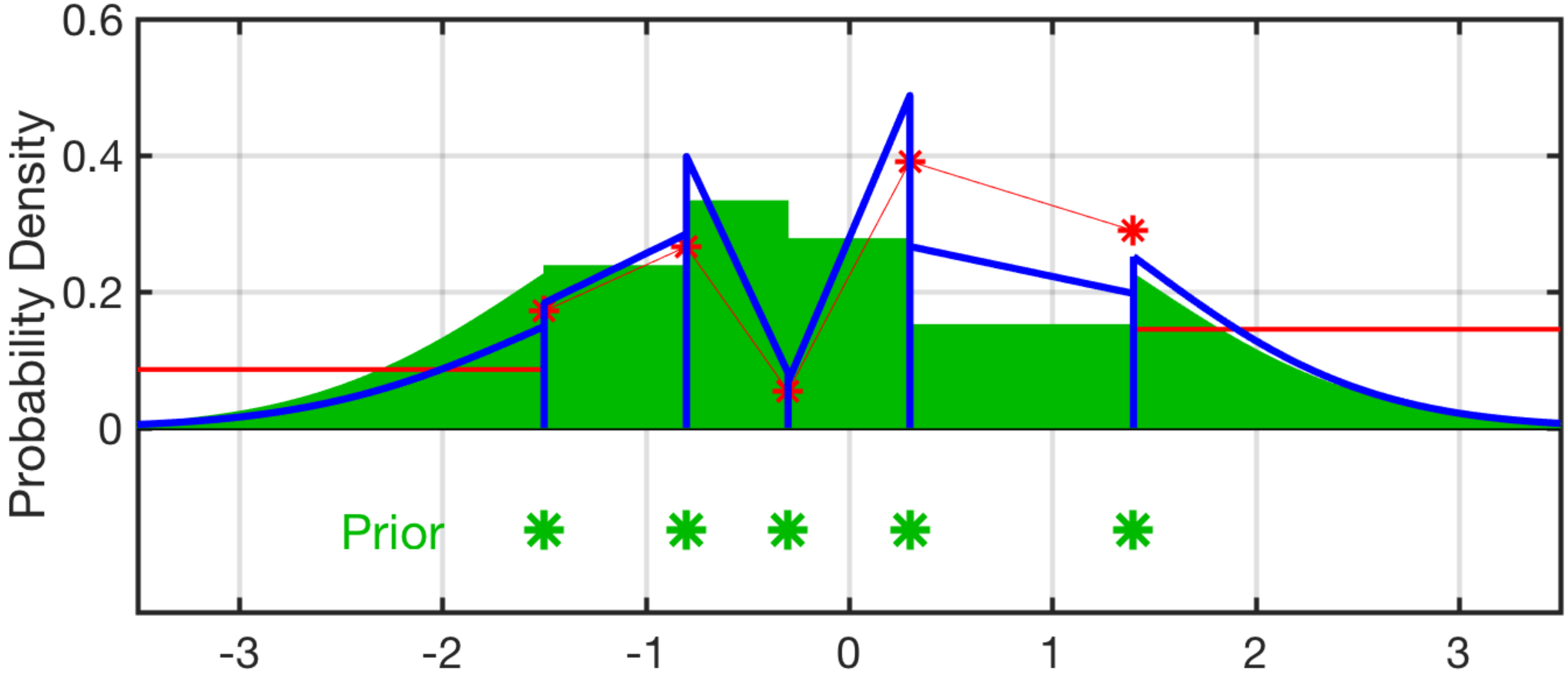
Rank Histogram Filter for State Marginals



Step 3: Compute continuous posterior distribution.

- Approximate likelihood with trapezoidal quadrature.
- Uniform likelihood tails! (Different). No Gaussian assumption left.

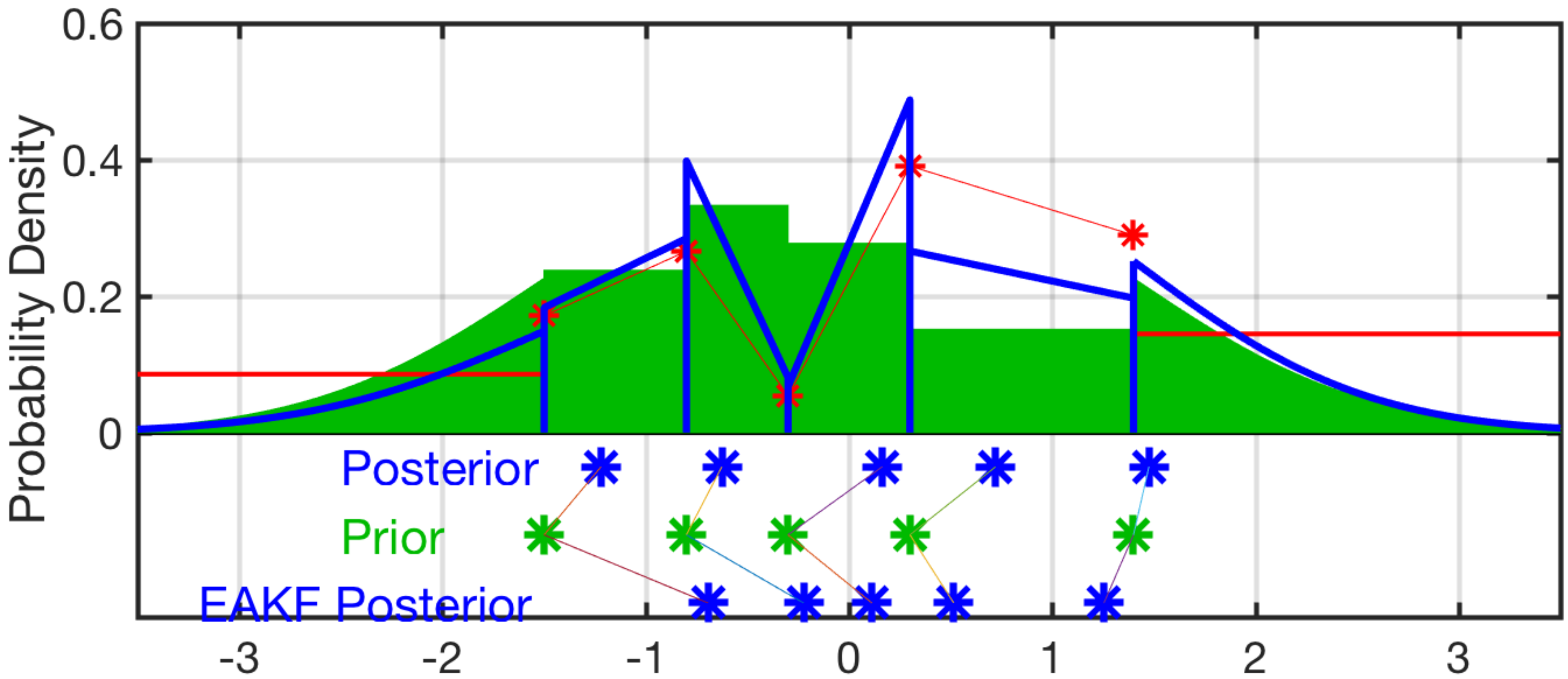
Rank Histogram Filter for State Marginals



Step 3: Compute continuous posterior distribution (same).

- Really simple with uniform likelihood tails.

Rank Histogram Filter for State Marginals



Step 4: Compute updated ensemble members (same):

- $(\text{ens\_size} + 1)^{-1}$ of posterior mass between each ensemble pair.
- $(\text{ens\_size} + 1)^{-1}$ in each tail.

Removing the Kalman from the Ensemble Kalman Filter

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.

Removing the Kalman from the Ensemble Kalman Filter

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

Anderson, J., Hoar, T., Raeder, K., Liu, H., Collins, N., Torn, R., Arellano, A., 2009: *The Data Assimilation Research Testbed: A community facility.* BAMS, **90**, 1283—1296, doi: 10.1175/2009BAMS2618.1