Prelude: The Data Assimilation Problem:

Given: 1. A physical system (atmosphere, ocean...)

---

2. Observations of the physical system

Usually sparse and irregular in time and space.
Instruments have error of which we have a (poor) estimate.
Observations may be of ‘non-state’ quantities.
Many observations may have very low information content.

---

3. A model of the physical system

Usually thought of as approximating time evolution.
Could also be just a model of balance (attractor) relations.
Truncated representation of ‘continuous’ physical system.
Often quasi-regular discretization in space and/or time.
Generally characterized by ‘large’ systematic errors.
May be ergodic with some sort of ‘attractor’.

We want to increase our information about all three pieces:

1. Get an improved estimate of state of physical system

Includes time evolution and 'balances'.
Initial conditions for forecasts.
High quality analyses (re-analyses).

2. Get better estimates of observing system error characteristics

Estimate value of existing observations.
Design observing systems that provide increased information.

3. Improve model of physical system

Evaluate model systematic errors.
Select appropriate values for model parameters.
Evaluate relative characteristics of different models.

Towards a General, Flexible Assimilation Facility

Goals:

1. Assimilation that works with variety of models and obs. types.
2. Coding for system must be easy to implement (weeks max.).

    (This appears to rule out variational methods at present).

3. Must allow complicated forward operators.
4. GOOD assimilation results for novice users.
   EXCELLENT results with added expertise/development.
5. GOOD performance on variety of platforms with little effort.
   EXCELLENT performance with added expertise/development.

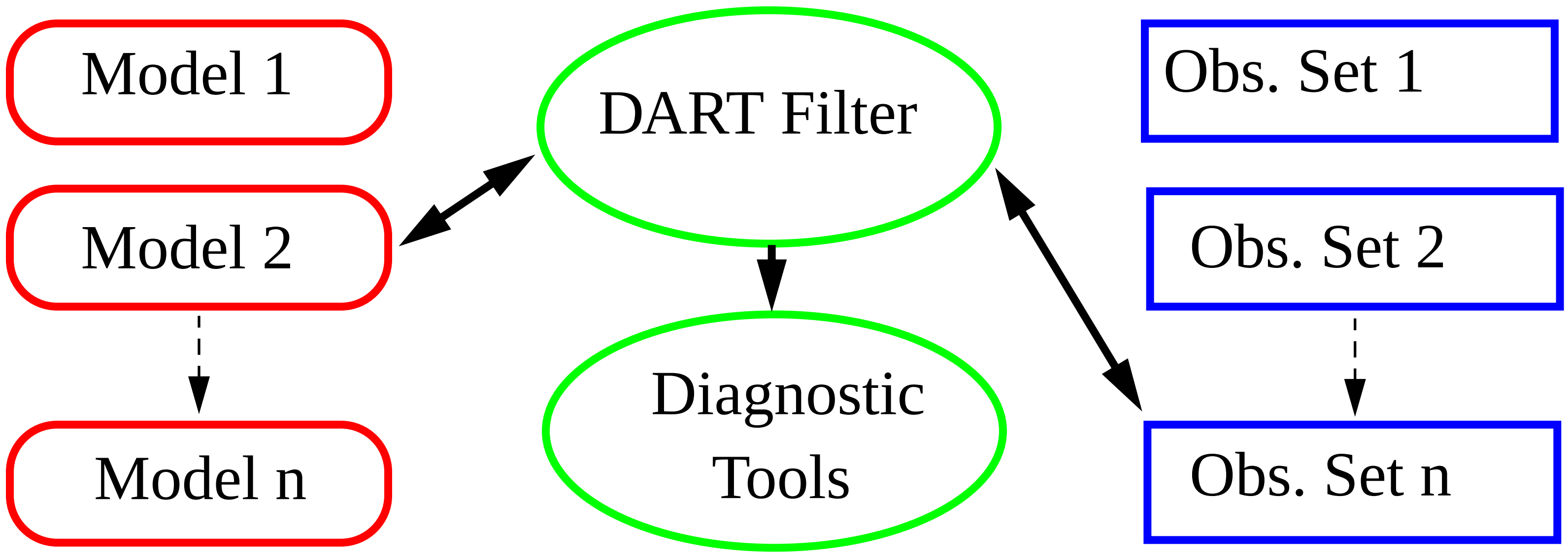
The Data Assimilation Research Testbed (DART)

A. Combine assimilation algorithms, models, and observation sets.

B. Diagnostic tools for assimilation experimentation.

C. Compliant models and observation sets (real and synthetic).

D. A high-quality, generic ensemble filtering algorithm.NCAR Data



Assimilation Initiative

Ensemble Filters for Geophysical Data Assimilation: A Tutorial

Jeffrey Anderson
NCAR Data Assimilation Initiative

Objective: Provide a simple but clear introduction to ensemble filters.

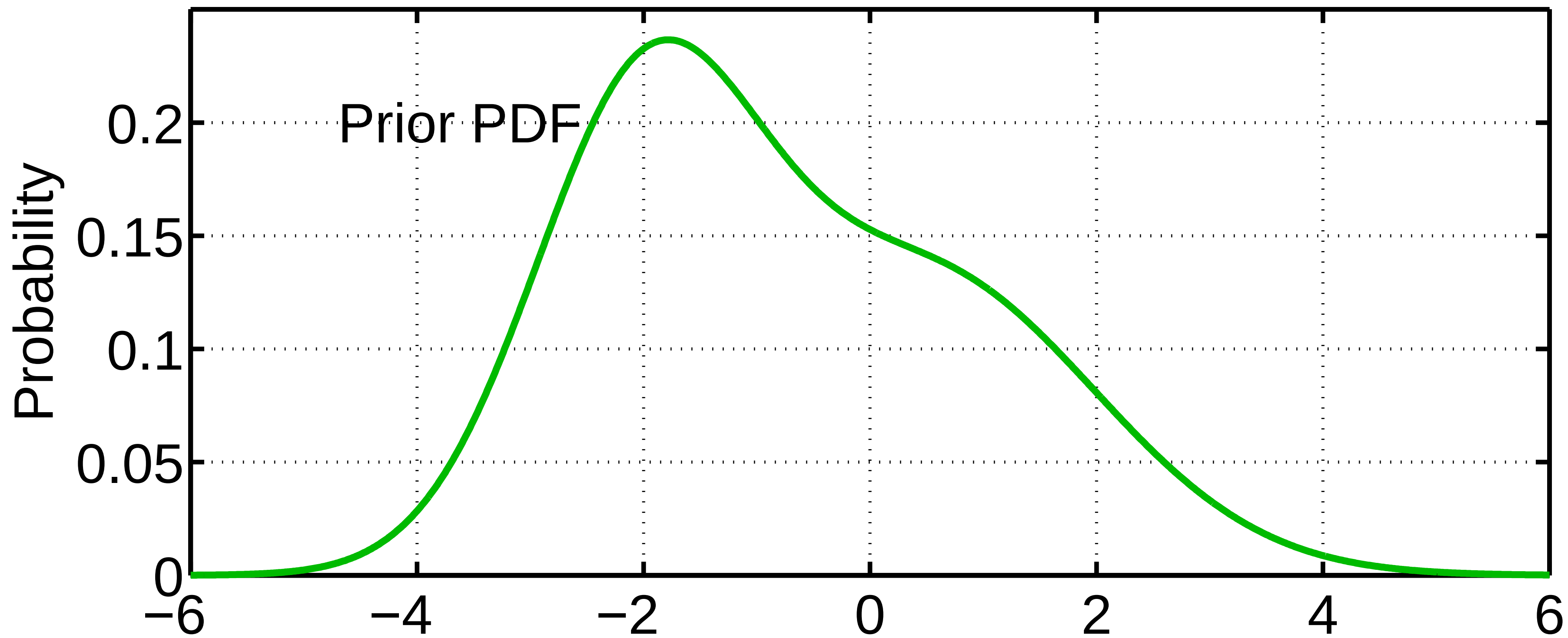
Phase 1: Single variable and observation of that variable.

Phase 2: Single observed variable, single unobserved variable.

Phase 3: Generalize to geophysical models and observations.

Phase 4: Quick look at a real atmospheric application.

Bayes rule: $p(A|BC) = \dfrac{p(B|AC)p(A|C)}{p(B|C)} = \dfrac{p(B|AC)p(A|C)}{\int p(B|x)p(x|C)dx}$

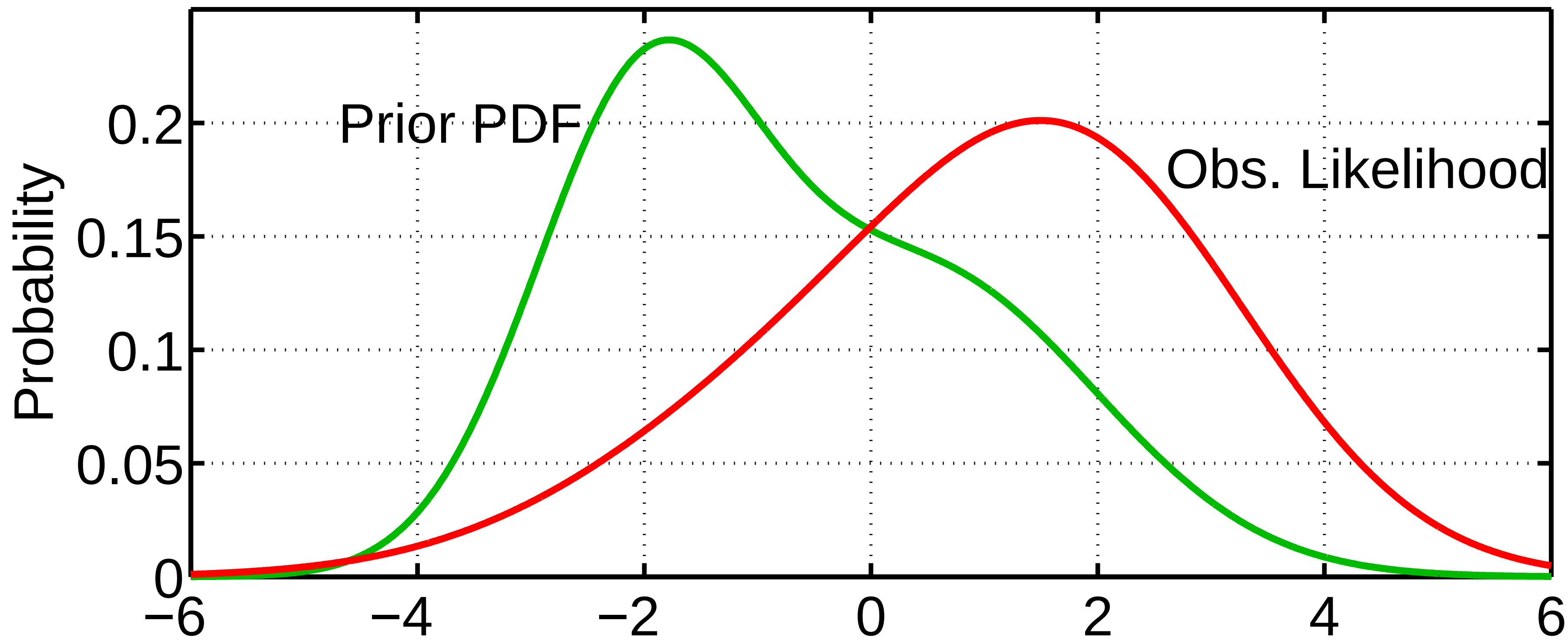


A: Prior estimate based on all previous information, C.

B: An additional observation.

$p(A|BC)$: Posterior (updated estimate) based on C and B.

Bayes rule: $p(A|BC) = \frac{p(B|AC)p(A|C)}{p(B|C)} = \frac{p(B|AC)p(A|C)}{\int p(B|x)p(x|C)dx}$

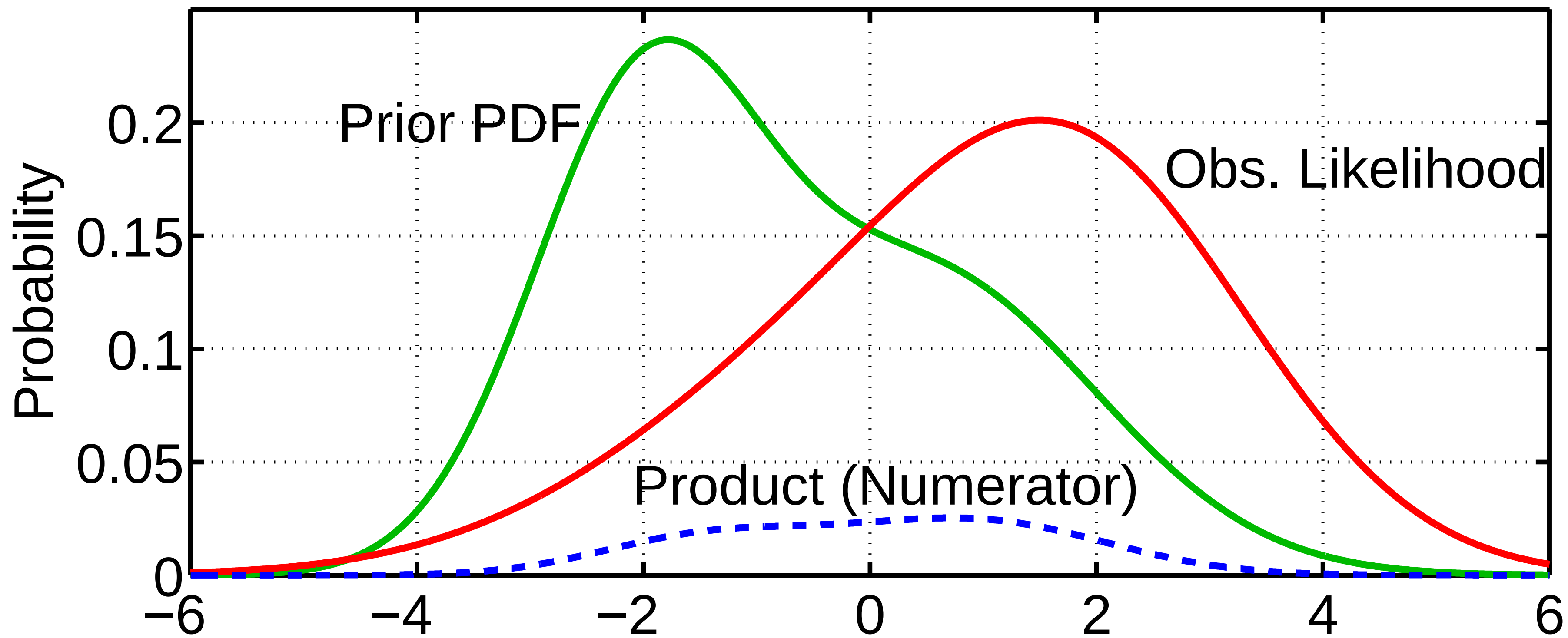


A: Prior estimate based on all previous information, C.

B: An additional observation.

$p(A|BC)$: Posterior (updated estimate) based on C and B.

Bayes rule: $p(A|BC) = \frac{p(B|AC)p(A|C)}{p(B|C)} = \frac{p(B|AC)p(A|C)}{\int p(B|x)p(x|C)dx}$

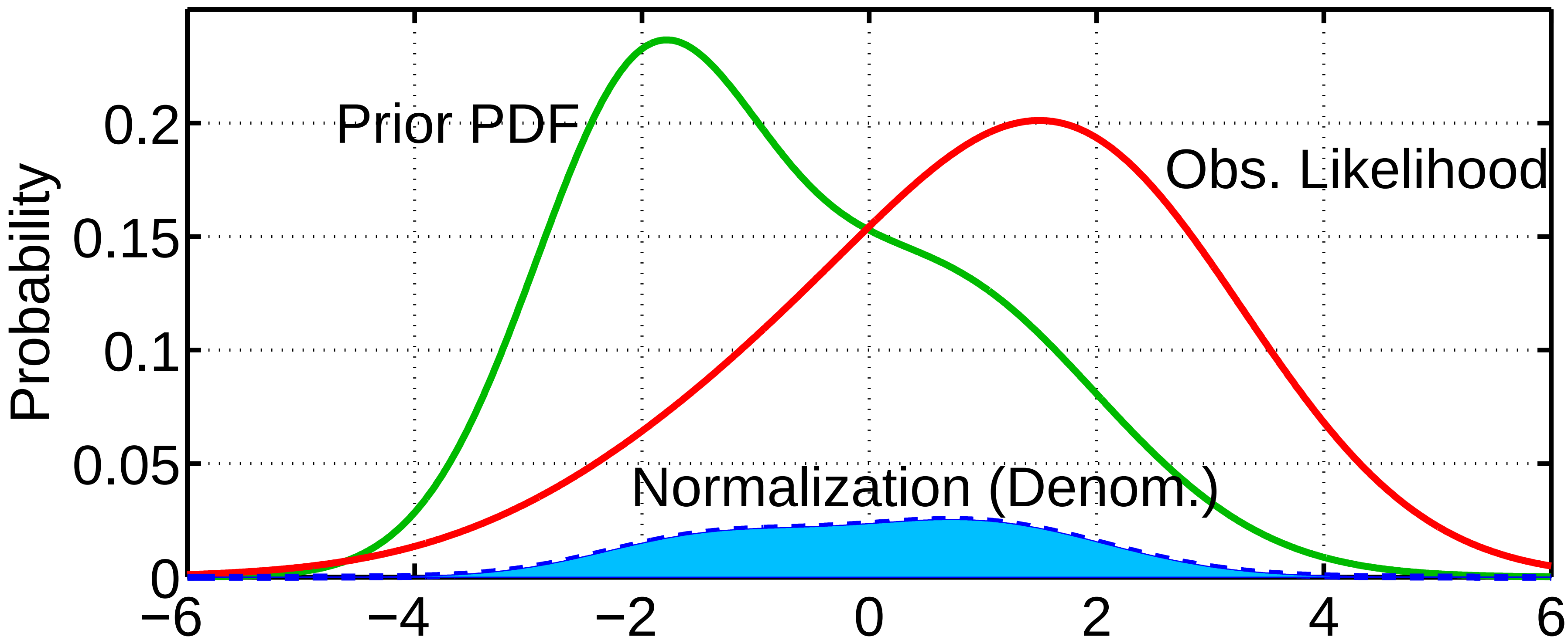


A: Prior estimate based on all previous information, C.

B: An additional observation.

$p(A|BC)$: Posterior (updated estimate) based on C and B.

Bayes rule: $p(A|BC) = \dfrac{p(B|AC)p(A|C)}{p(B|C)} = \dfrac{p(B|AC)p(A|C)}{\int p(B|x)p(x|C)dx}$

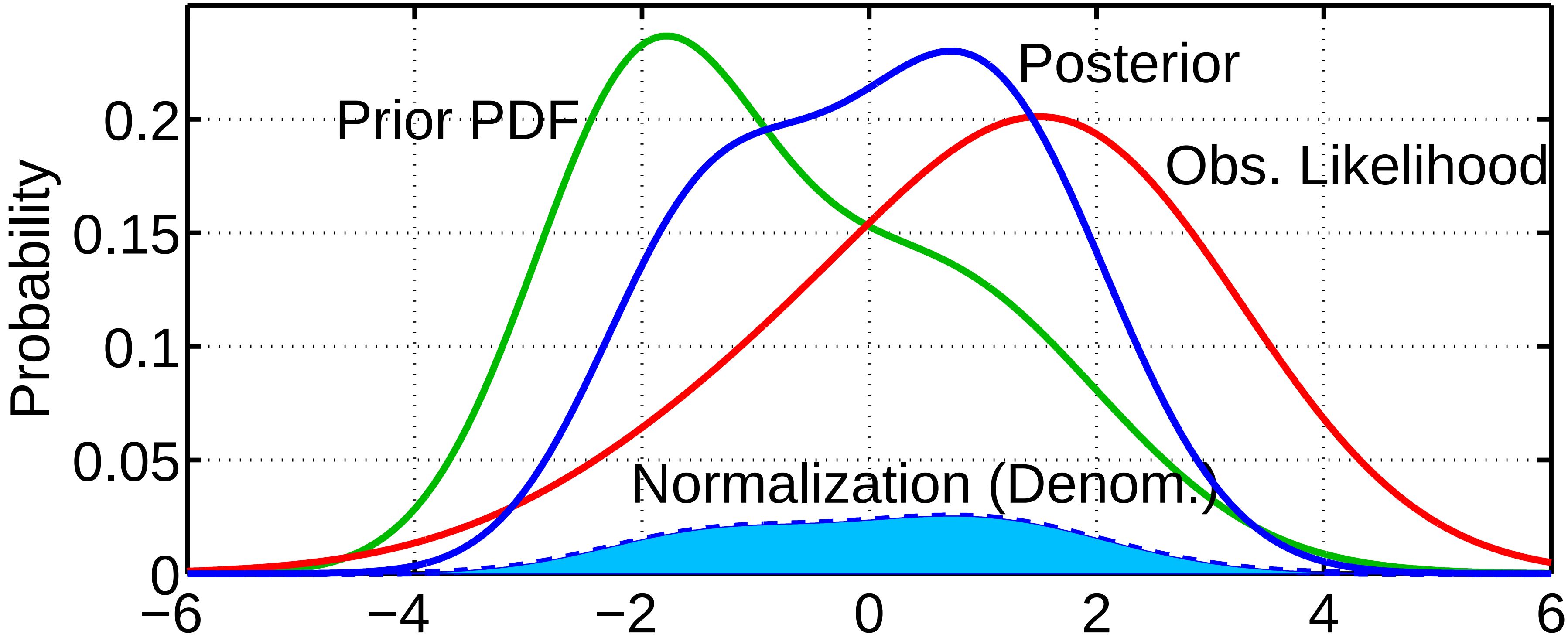


A: Prior estimate based on all previous information, C.

B: An additional observation.

$p(A|BC)$: Posterior (updated estimate) based on C and B.

Bayes rule: $p(A|BC) = \dfrac{p(B|AC)p(A|C)}{p(B|C)} = \dfrac{p(B|AC)p(A|C)}{\int p(B|x)p(x|C)dx}$



A: Prior estimate based on all previous information, C.

B: An additional observation.

$p(A|BC)$: Posterior (updated estimate) based on C and B.

Consistent Color Scheme Throughout Tutorial

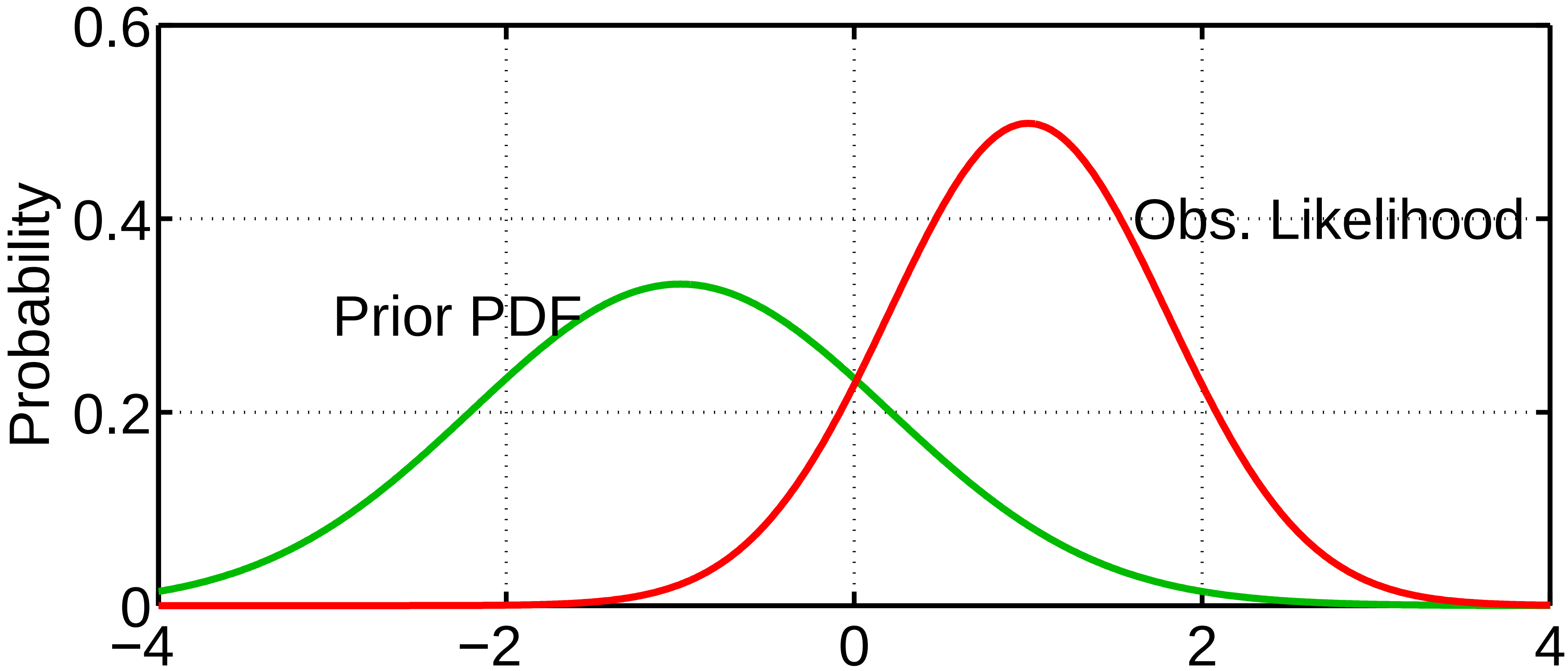
Green = Prior

Red = Observation

Blue = Posterior

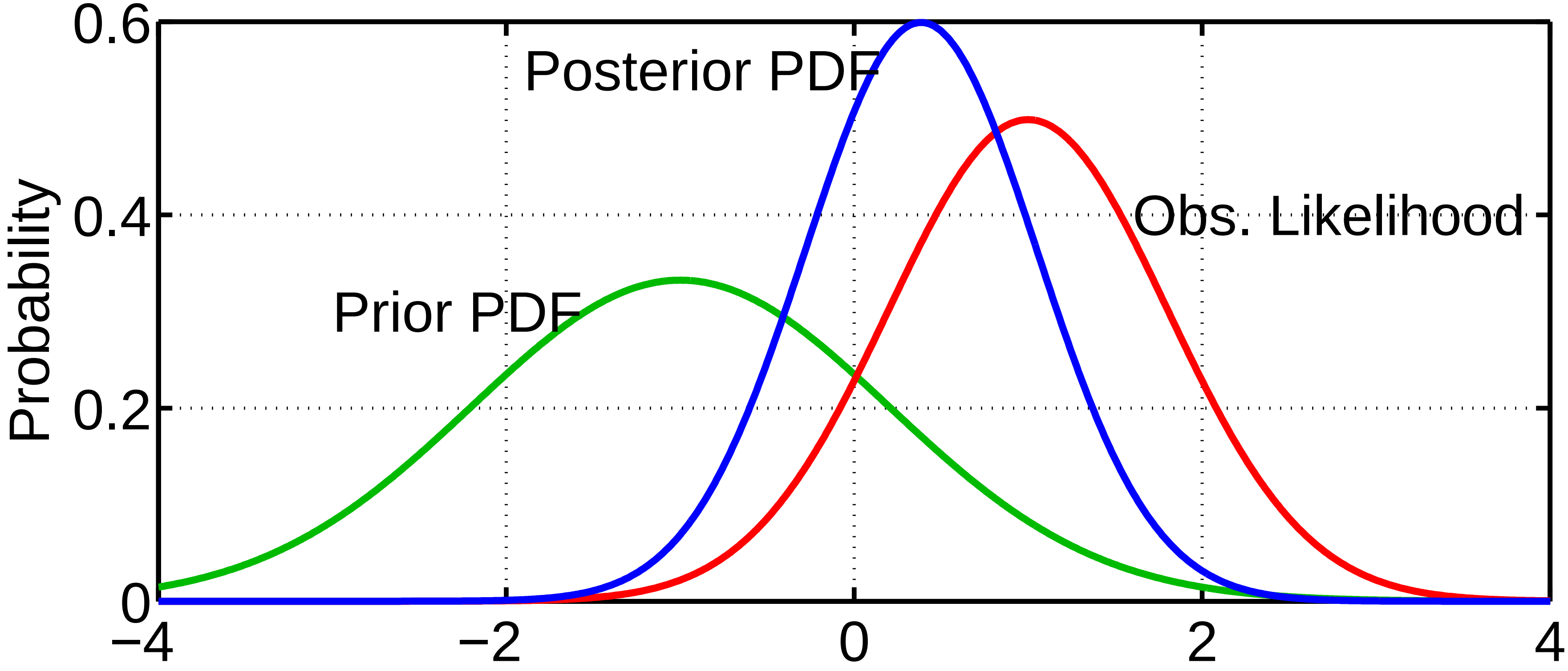
Bayes rule: $p(A|BC) = \dfrac{p(B|AC)p(A|C)}{p(B|C)} = \dfrac{p(B|AC)p(A|C)}{\int p(B|x)p(x|C)dx}$

This product is closed for Gaussian distributions.



Bayes rule: $p(A|BC) = \frac{p(B|AC)p(A|C)}{p(B|C)} = \frac{p(B|AC)p(A|C)}{\int p(B|x)p(x|C)dx}$

This product is closed for Gaussian distributions.



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.

$$\mathbf{N}(\mu_1, \Sigma_1)\boldsymbol{N}(\mu_2, \Sigma_2) = c\mathbf{N}(\mu, \Sigma)$$

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.

$$\mathbf{N}(\mu_1, \Sigma_1)\boldsymbol{N}(\mu_2, \Sigma_2) = c\mathbf{N}(\mu, \Sigma)$$

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

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

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.

$$\mathbf{N}(\mu_1, \Sigma_1)N(\mu_2, \Sigma_2) = c\mathbf{N}(\mu, \Sigma)$$

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

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

Weight: $c = \dfrac{1}{(2\Pi)^{d/2}\left|\Sigma_1 + \Sigma_2\right|^{1/2}} \exp\left\{-\dfrac{1}{2}[(\mu_2 - \mu_1)^T(\Sigma_1 + \Sigma_2)^{-1}(\mu_2 - \mu_1)]\right\}$

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

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.

$$\mathbf{N}(\mu_1, \Sigma_1)N(\mu_2, \Sigma_2) = c\mathbf{N}(\mu, \Sigma)$$

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

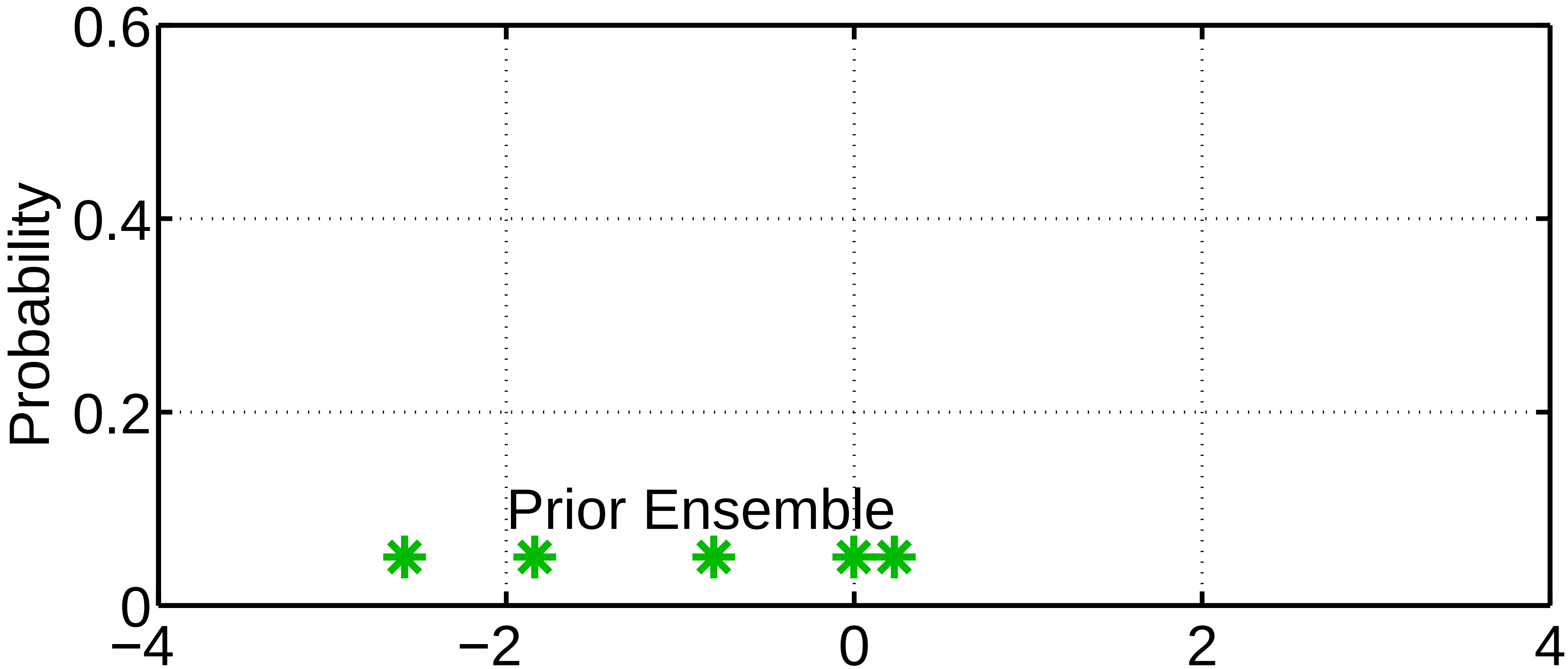
Mean: $\mu = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}(\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2)$

Weight: $c = \dfrac{1}{(2\Pi)^{d/2}\left|\Sigma_1 + \Sigma_2\right|^{1/2}} \exp\left\{-\dfrac{1}{2}[(\mu_2 - \mu_1)^T(\Sigma_1 + \Sigma_2)^{-1}(\mu_2 - \mu_1)]\right\}$

Easy to derive for 1-D Gaussians; just do products of exponentials.

Bayes rule: $p(A|BC) = \dfrac{p(B|AC)p(A|C)}{p(B|C)} = \dfrac{p(B|AC)p(A|C)}{\int p(B|x)p(x|C)dx}$

Ensemble filters: Prior is available as finite sample.

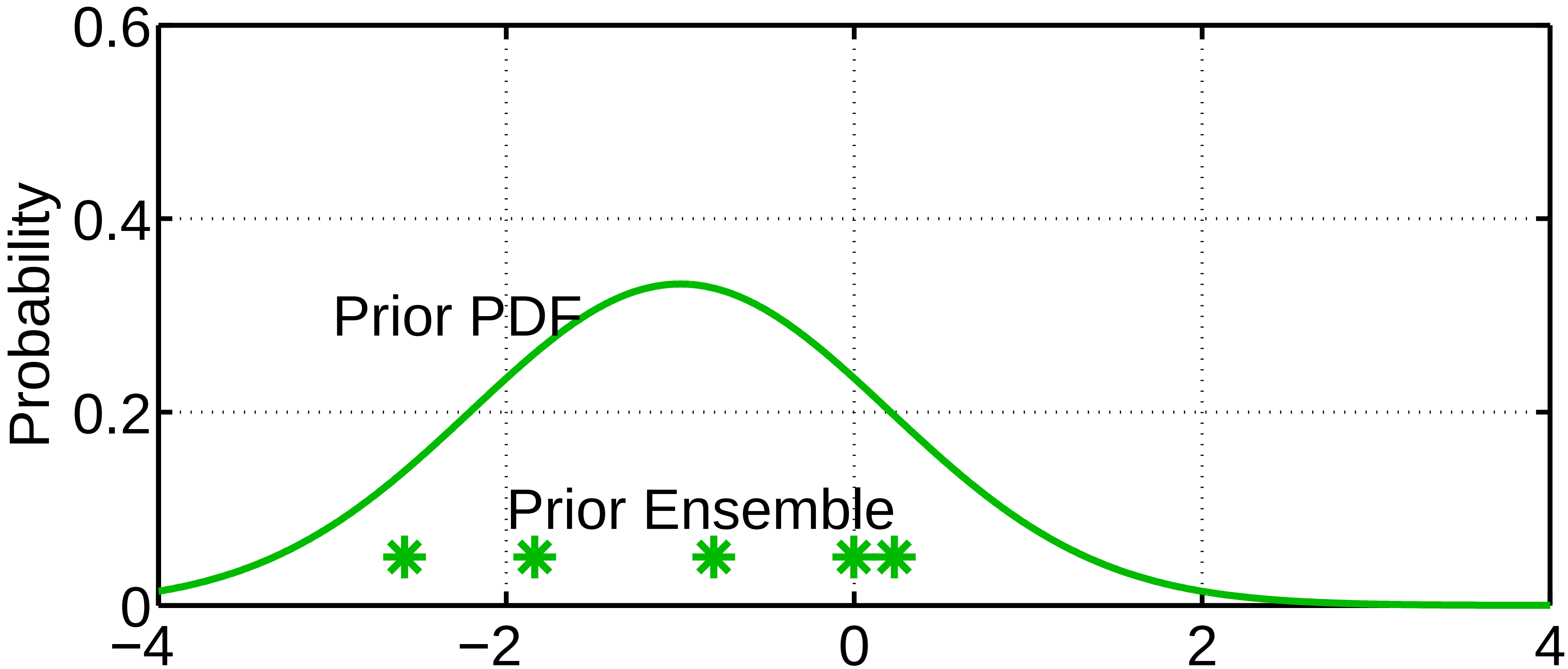


Don't know much about properties of this sample.

May naively assume it is random draw from 'truth'.

Bayes rule: $p(A|BC) = \dfrac{p(B|AC)p(A|C)}{p(B|C)} = \dfrac{p(B|AC)p(A|C)}{\int p(B|x)p(x|C)dx}$

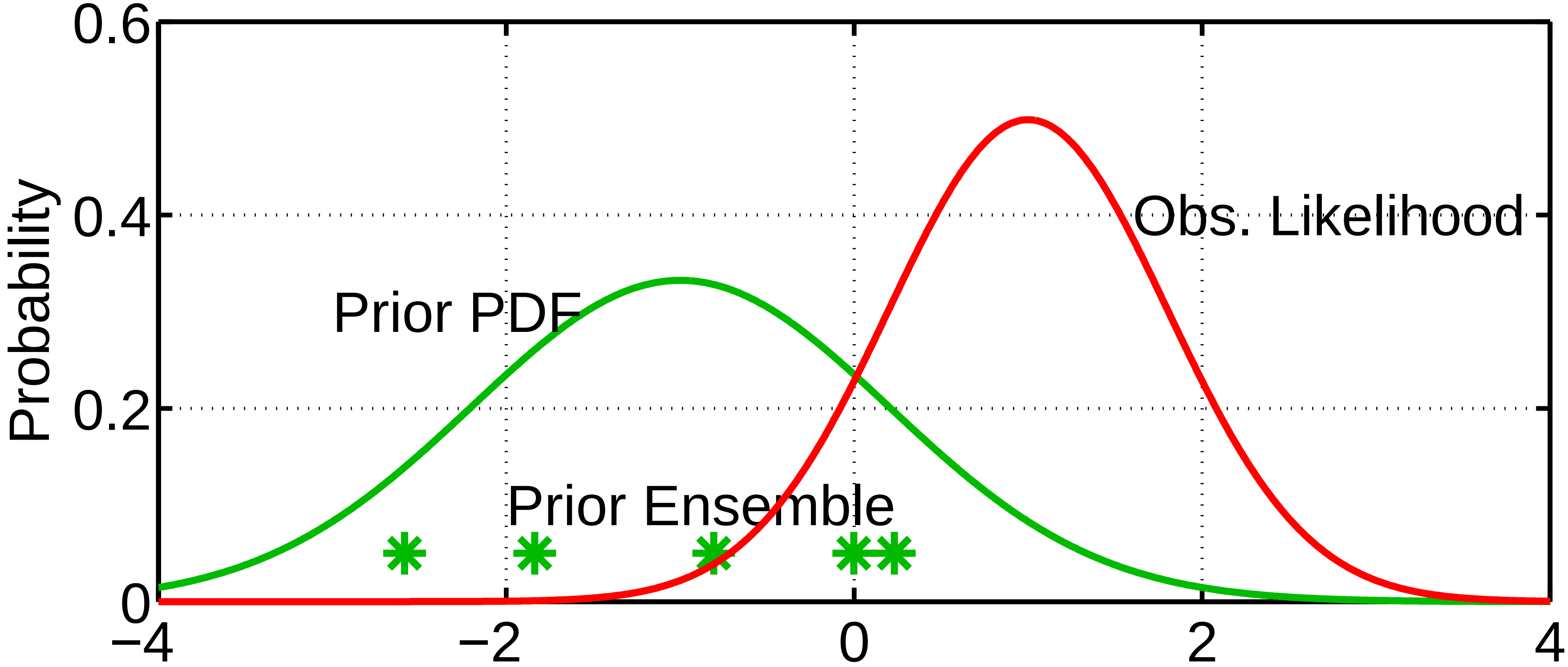
How can we take product of sample with continuous likelihood?



Fit a continuous (Gaussian for now) distribution to sample.

Bayes rule: $p(A|BC) = \frac{p(B|AC)p(A|C)}{p(B|C)} = \frac{p(B|AC)p(A|C)}{\int p(B|x)p(x|C)dx}$

Observation likelihood usually continuous (nearly always Gaussian).

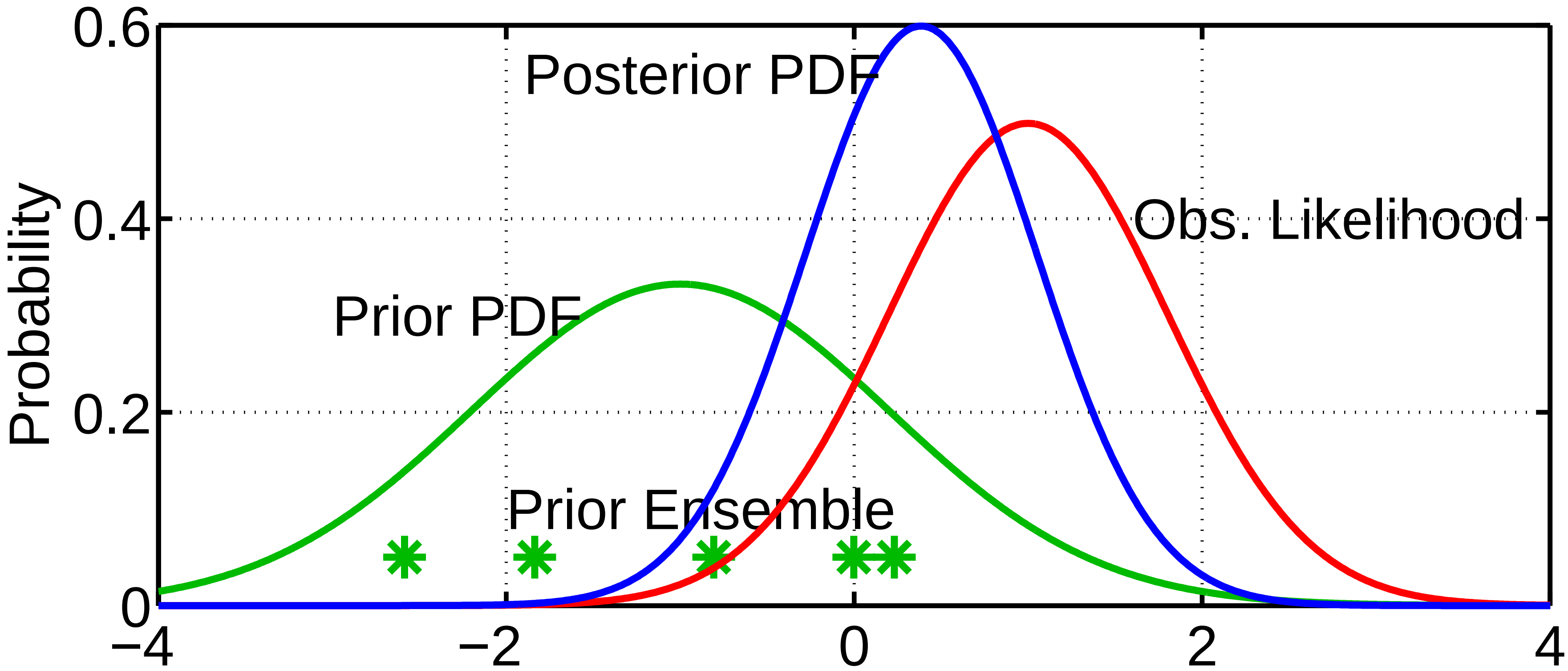


If Obs. Likelihood isn't Gaussian, can generalize methods below.

For instance, can fit set of Gaussian kernels to obs. likelihood.

Bayes rule: $p(A|BC) = \dfrac{p(B|AC)p(A|C)}{p(B|C)} = \dfrac{p(B|AC)p(A|C)}{\int p(B|x)p(x|C)dx}$

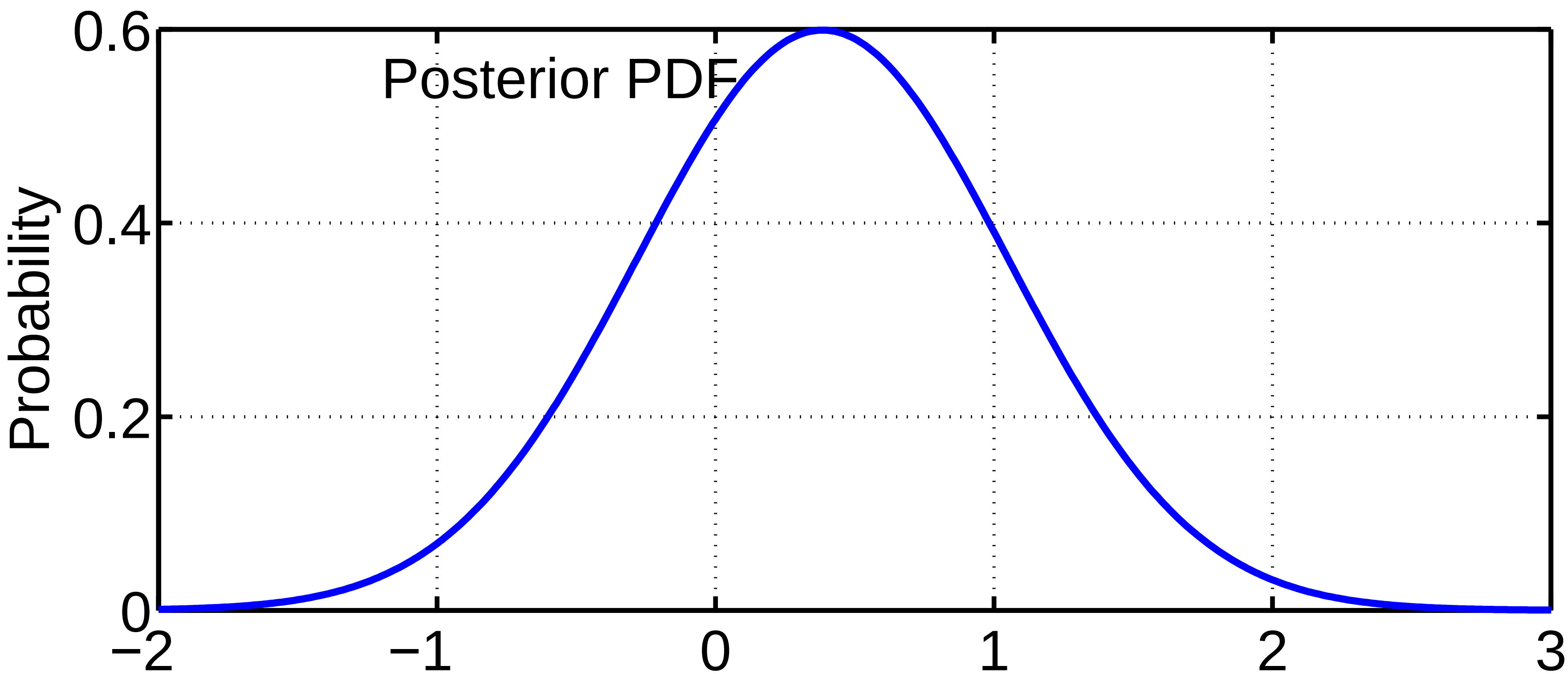
Product of prior Gaussian fit and Obs. likelihood is Gaussian.



Computing continuous posterior is simple.
BUT, need to have a SAMPLE of this PDF.

Sampling Posterior PDF:

There are many ways to do this.



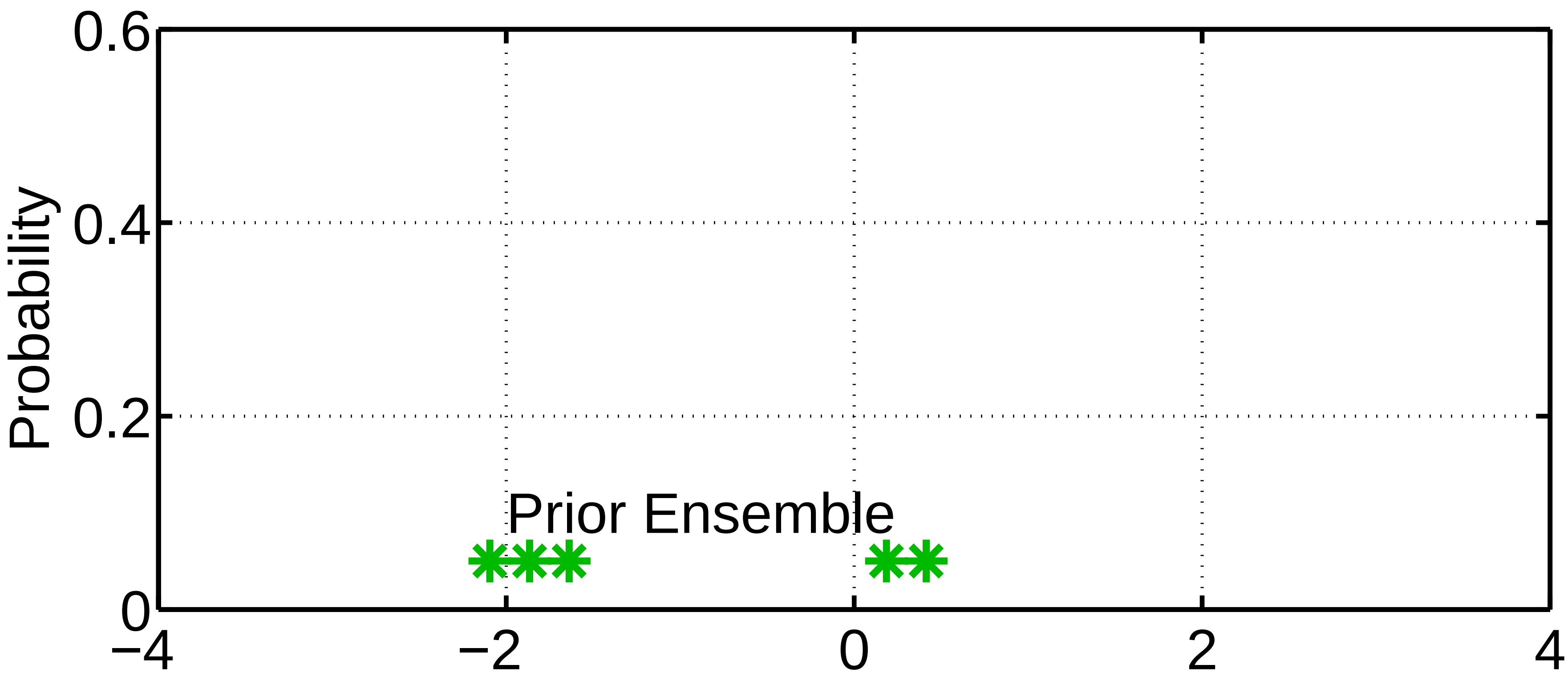
Exact properties of different methods may be unclear.

Trial and error still best way to see how they perform.

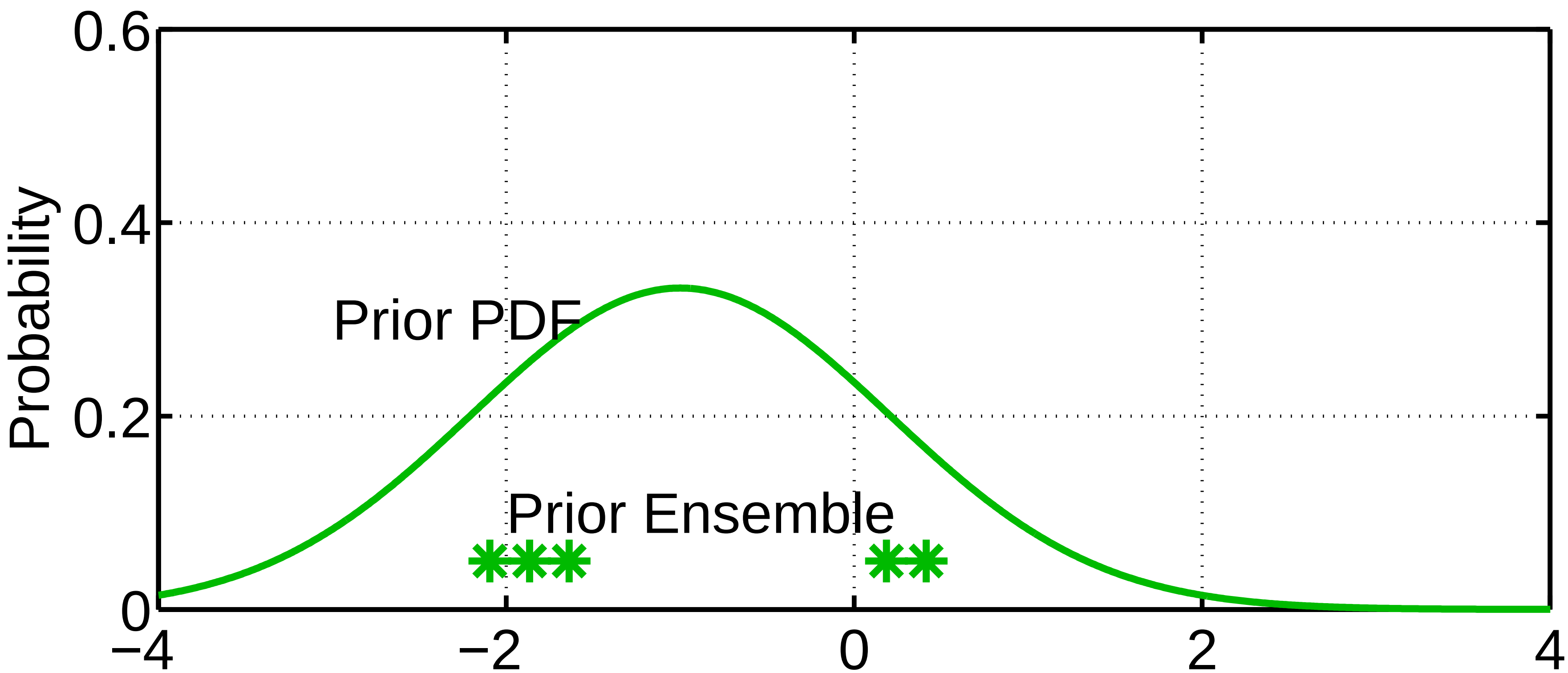
Will interact with properties of prediction models, etc.

Ensemble Filter Algorithms:

3. Ensemble Adjustment (Kalman) Filter.



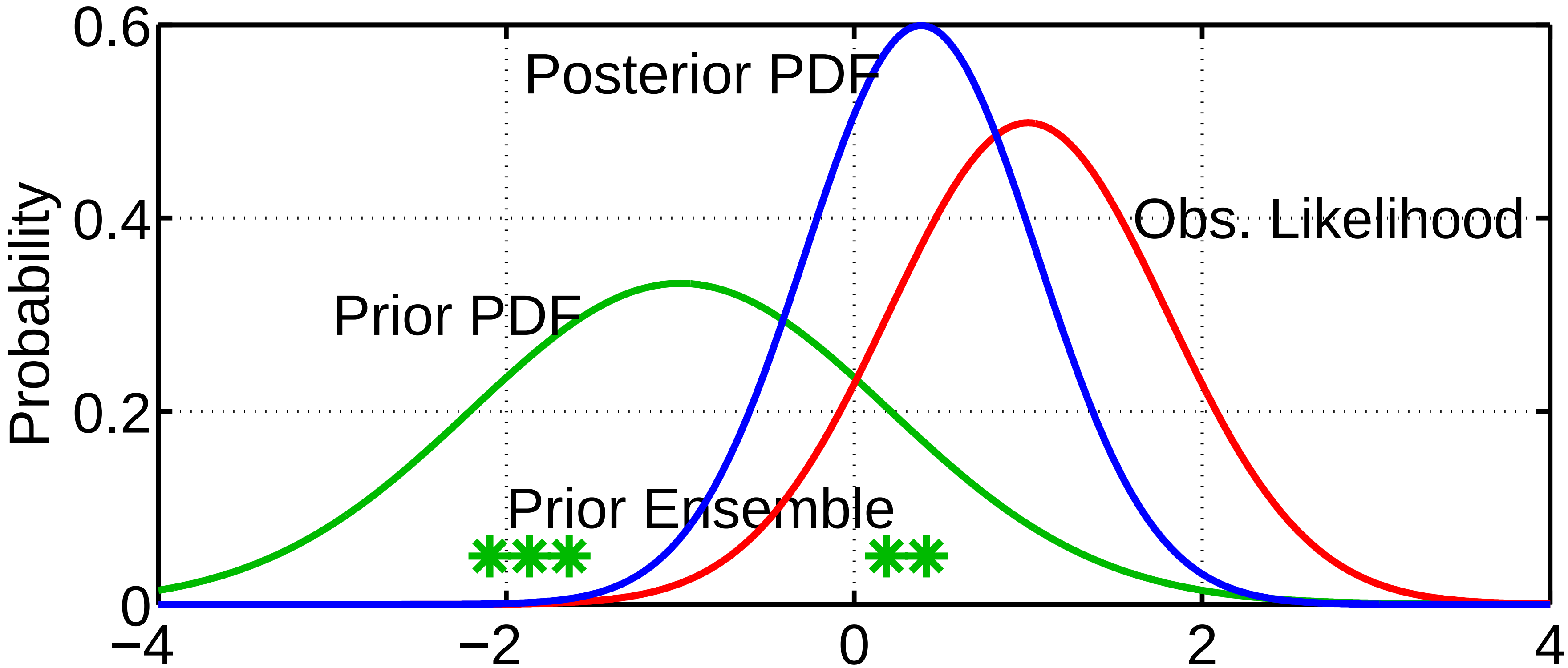
3. Ensemble Adjustment (Kalman) Filter.



Again, fit a Gaussian to sample.

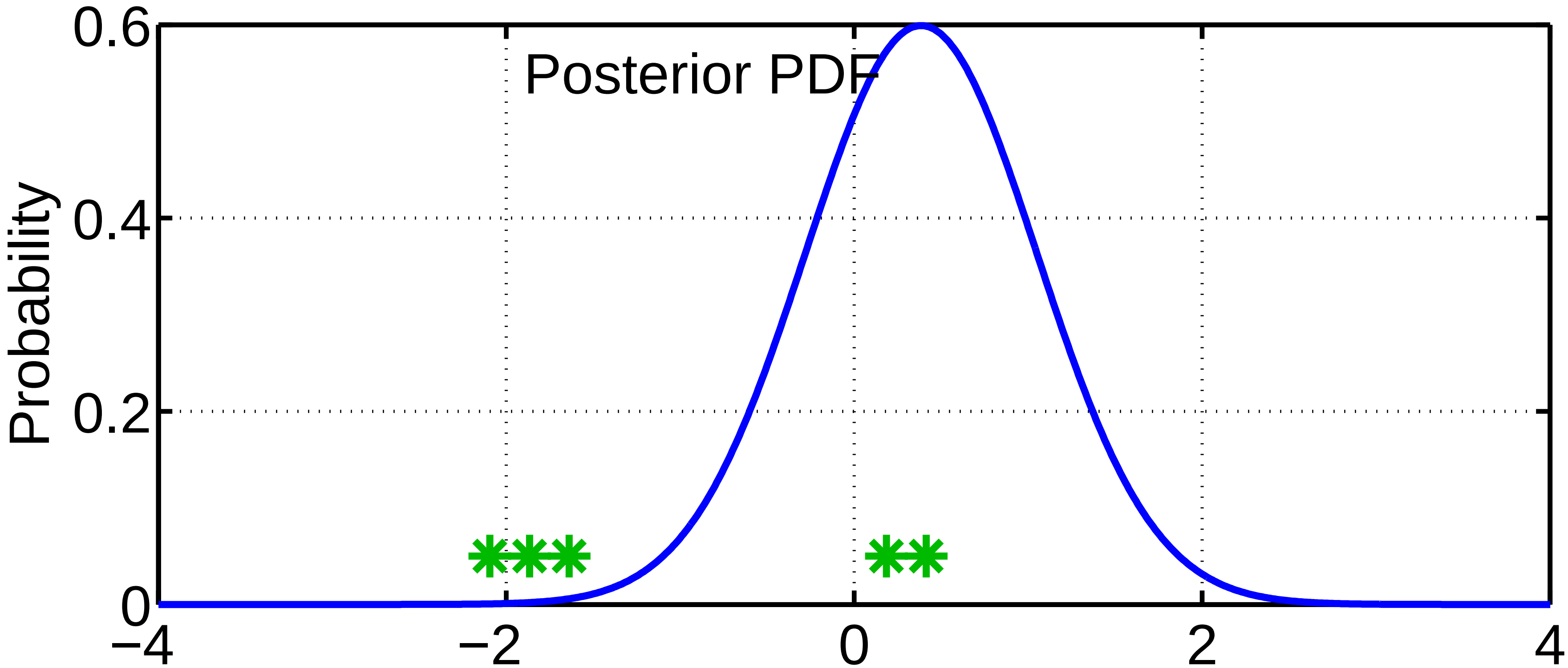
Ensemble Filter Algorithms:

3. Ensemble Adjustment (Kalman) Filter.



Compute posterior PDF (same as previous algorithms).

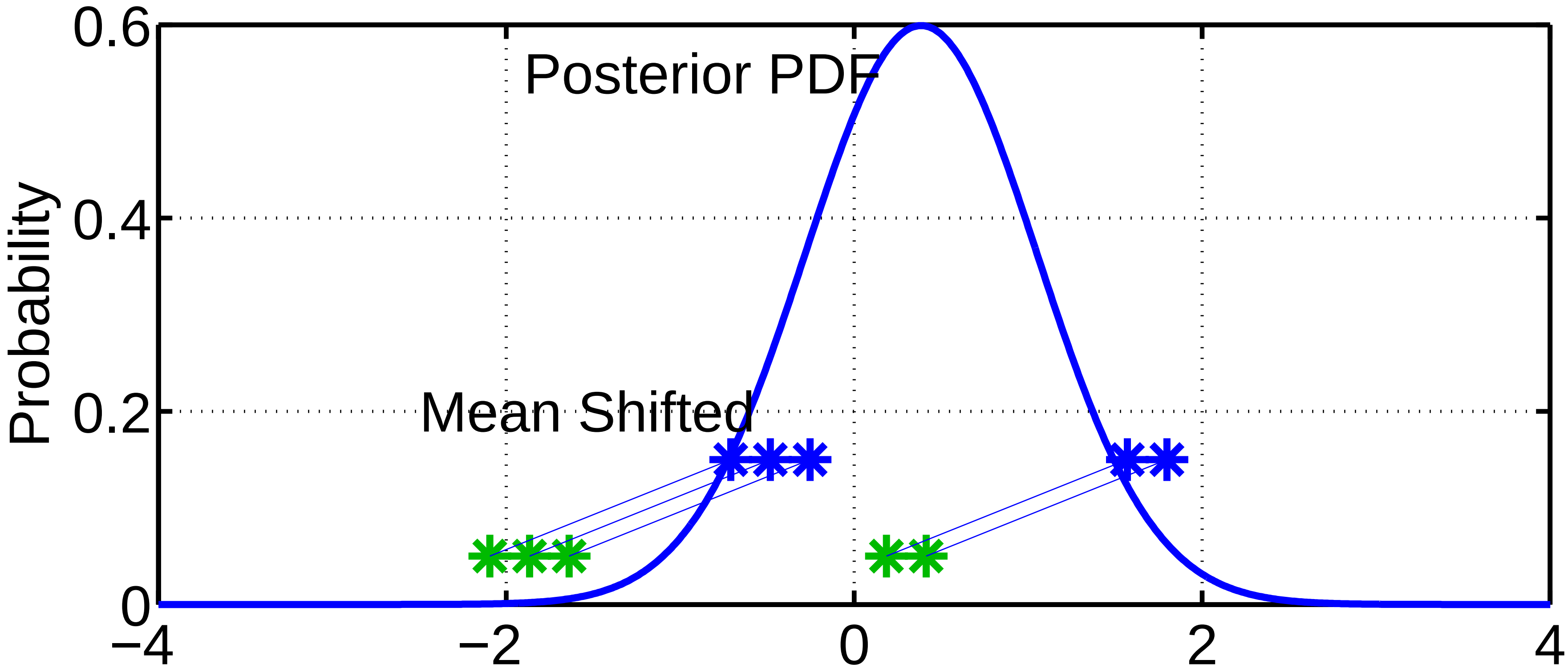
3. Ensemble Adjustment (Kalman) Filter.



Use deterministic algorithm to ‘adjust’ ensemble.

Ensemble Filter Algorithms:

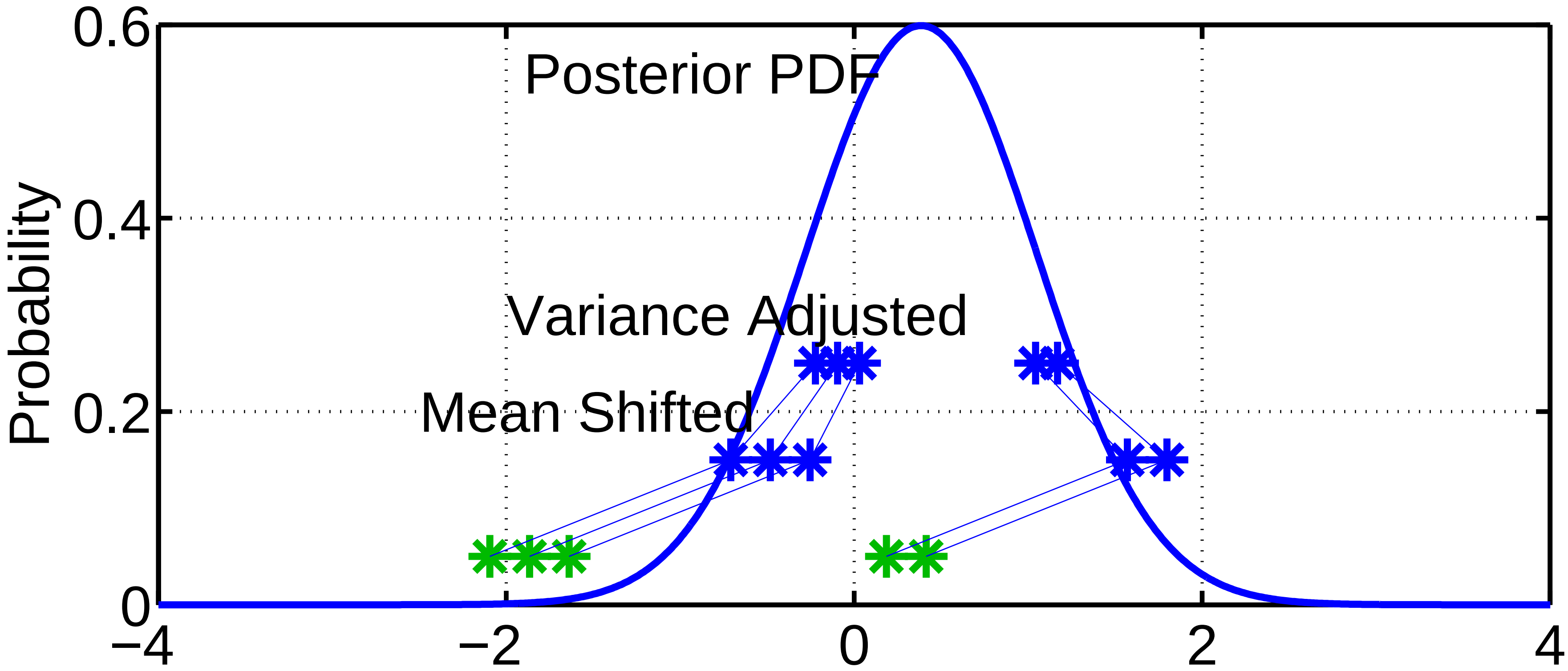
3. Ensemble Adjustment (Kalman) Filter.



Use deterministic algorithm to 'adjust' ensemble.

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

3. Ensemble Adjustment (Kalman) Filter.



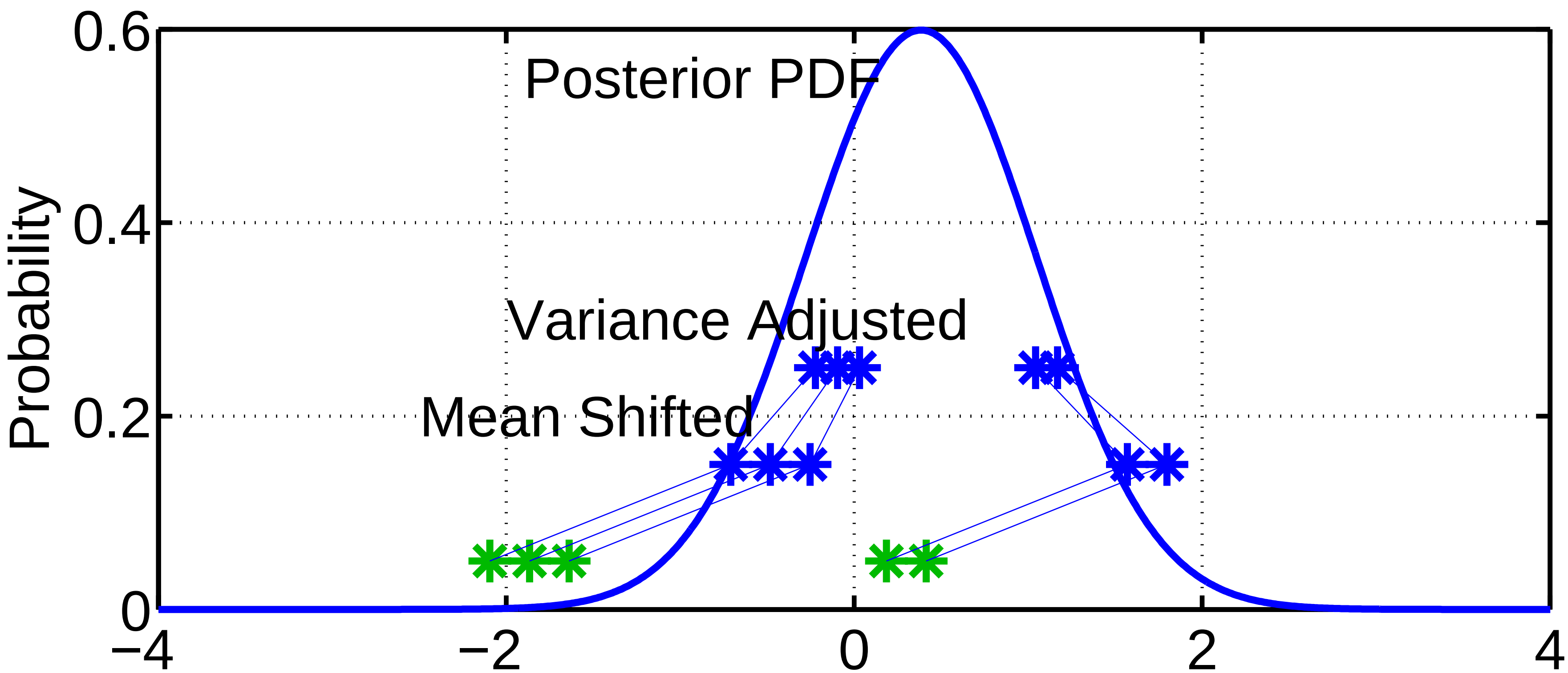
Use deterministic algorithm to ‘adjust’ ensemble.

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

Second, use linear contraction to have exact variance of posterior.

Ensemble Filter Algorithms:

3. Ensemble Adjustment (Kalman) Filter.



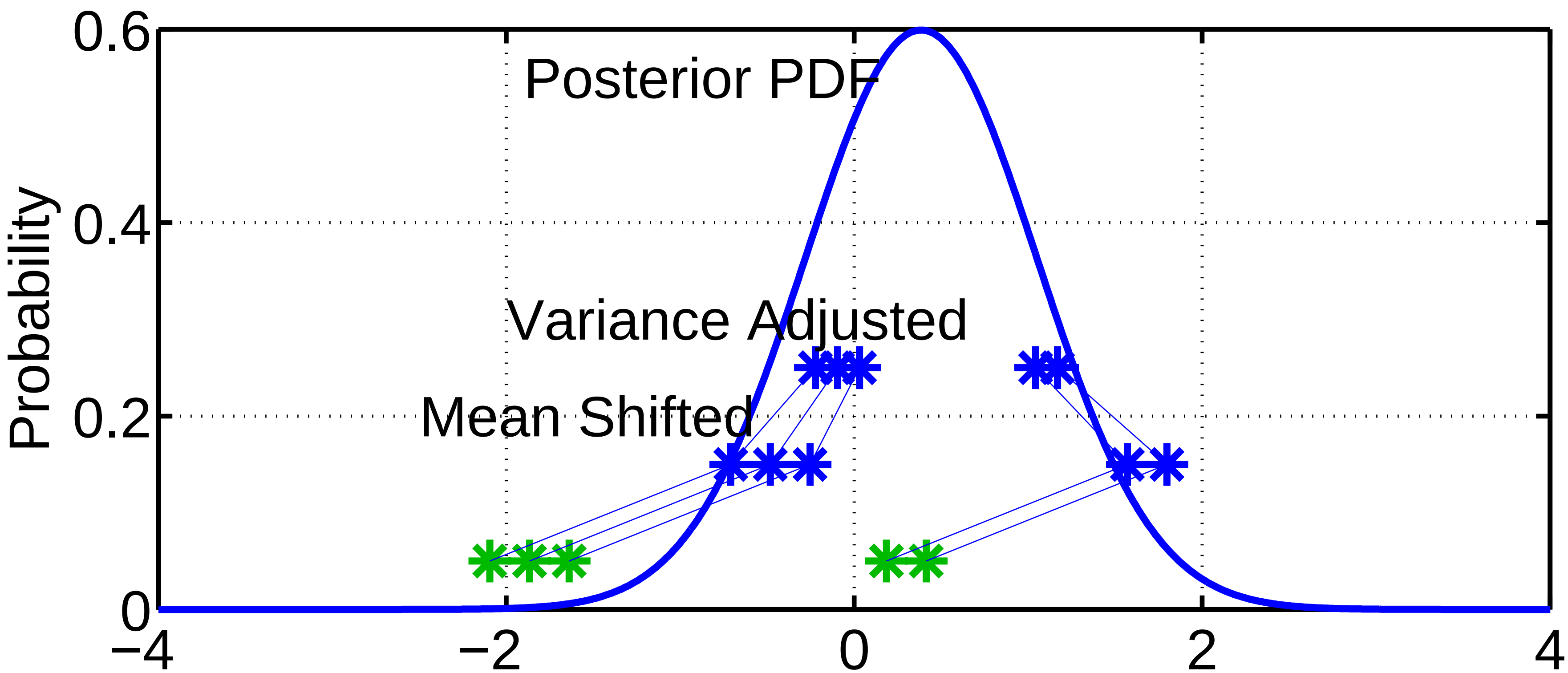
$$x_i^u = (x_i^p - \overline{x^p}) \cdot (\sigma^u / \sigma^p) + \overline{x^u} \quad \text{i = 1,..., ensemble size.}$$

p is prior, u is update (posterior), overbar is ensemble mean,

σ is standard deviation.

Ensemble Filter Algorithms:

3. Ensemble Adjustment (Kalman) Filter.



Bimodality maintained, but not appropriately positioned or weighted.
No problem with random outliers.

Phase 2: Single observed variable, single unobserved variable

So far, have known observation likelihood for single variable.

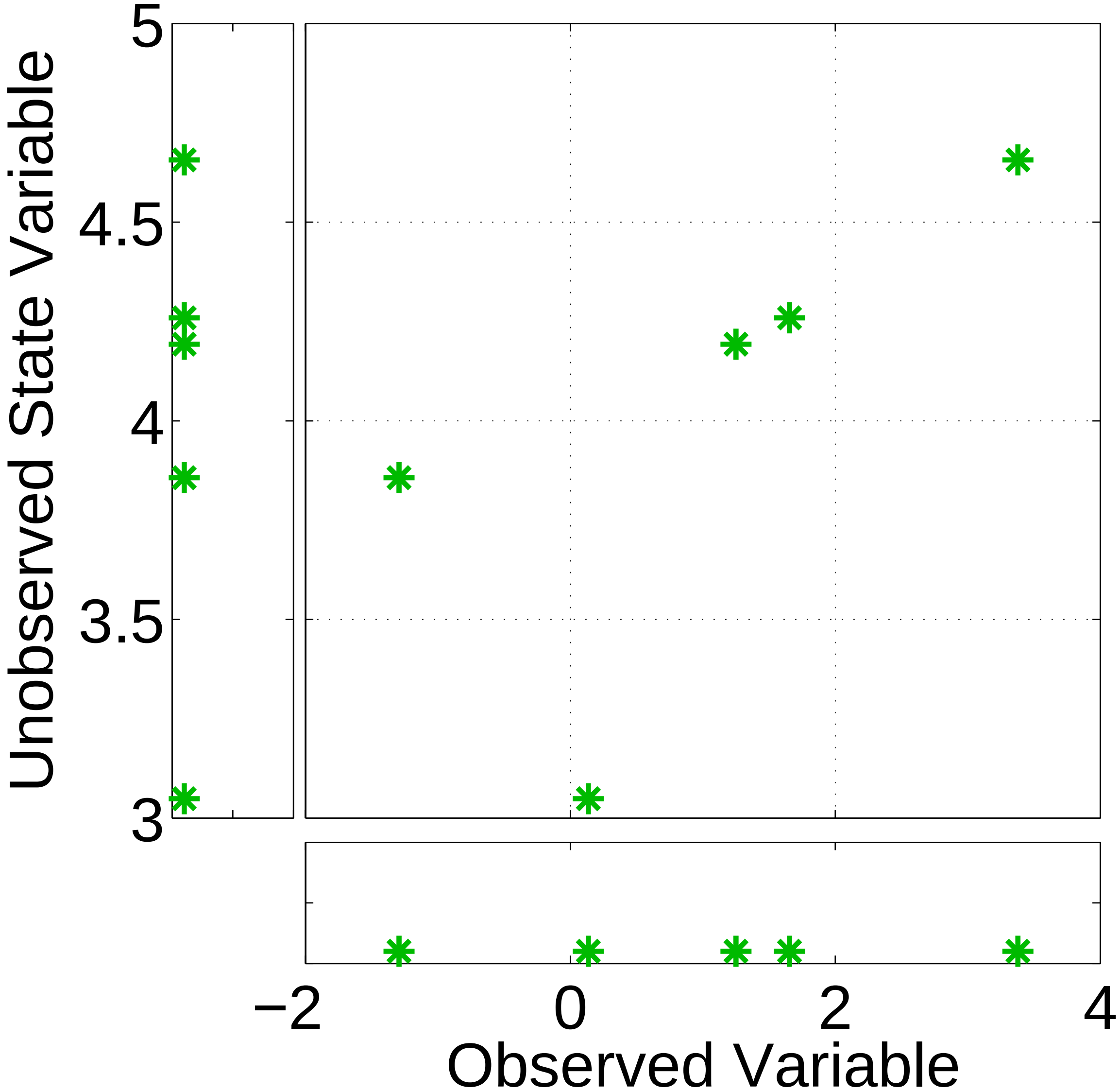
Now, suppose prior has an additional variable.

Will examine how ensemble methods update additional variable.

Basic method generalizes to any number of additional variables.

Methods related to Kalman filter in some sense, but not done here.

Ensemble filters: Updating additional prior state variables

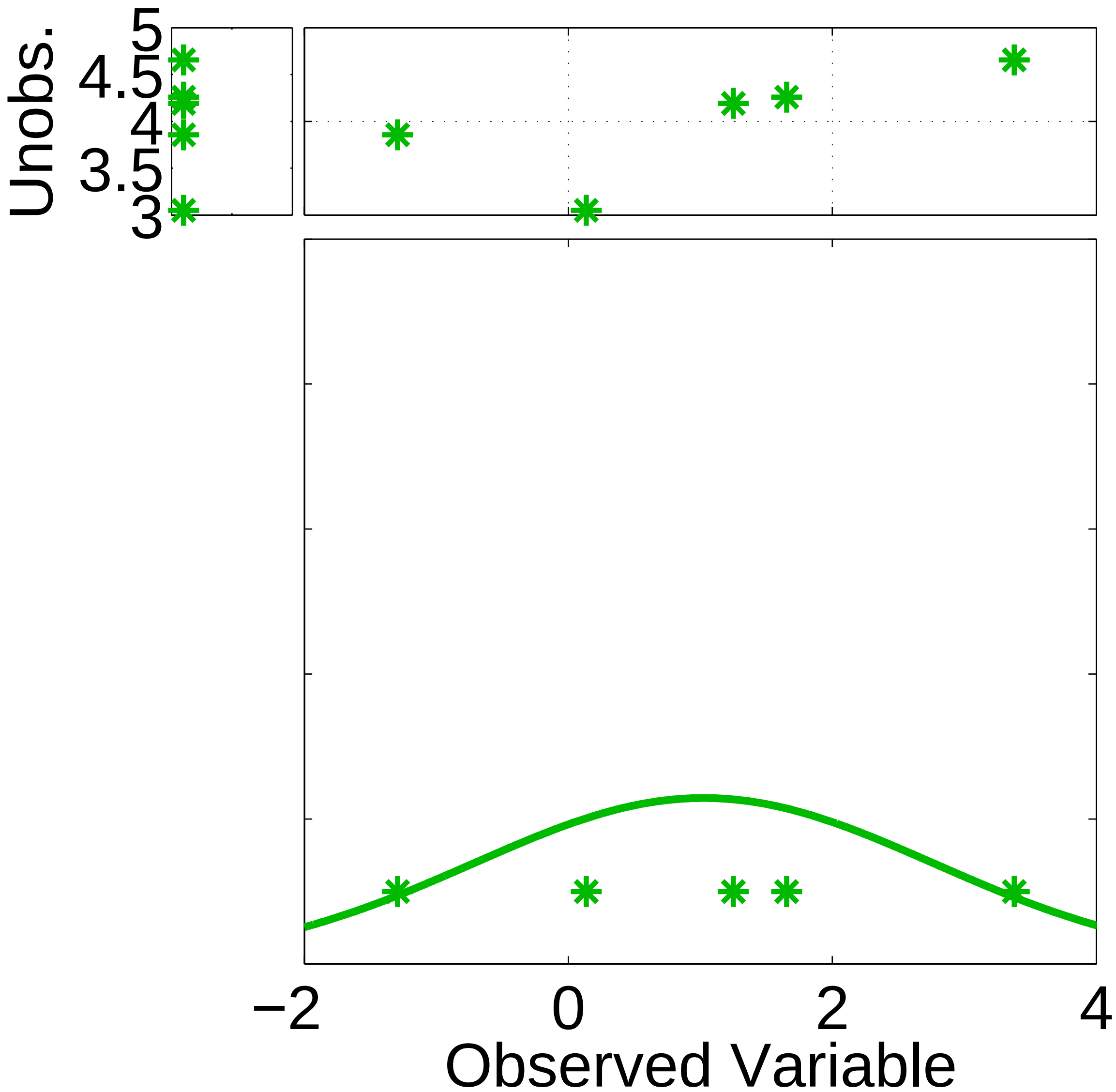


Assume that all we know is prior joint distribution.

One variable is observed.

What should happen to unobserved variable?

Ensemble filters: Updating additional prior state variables

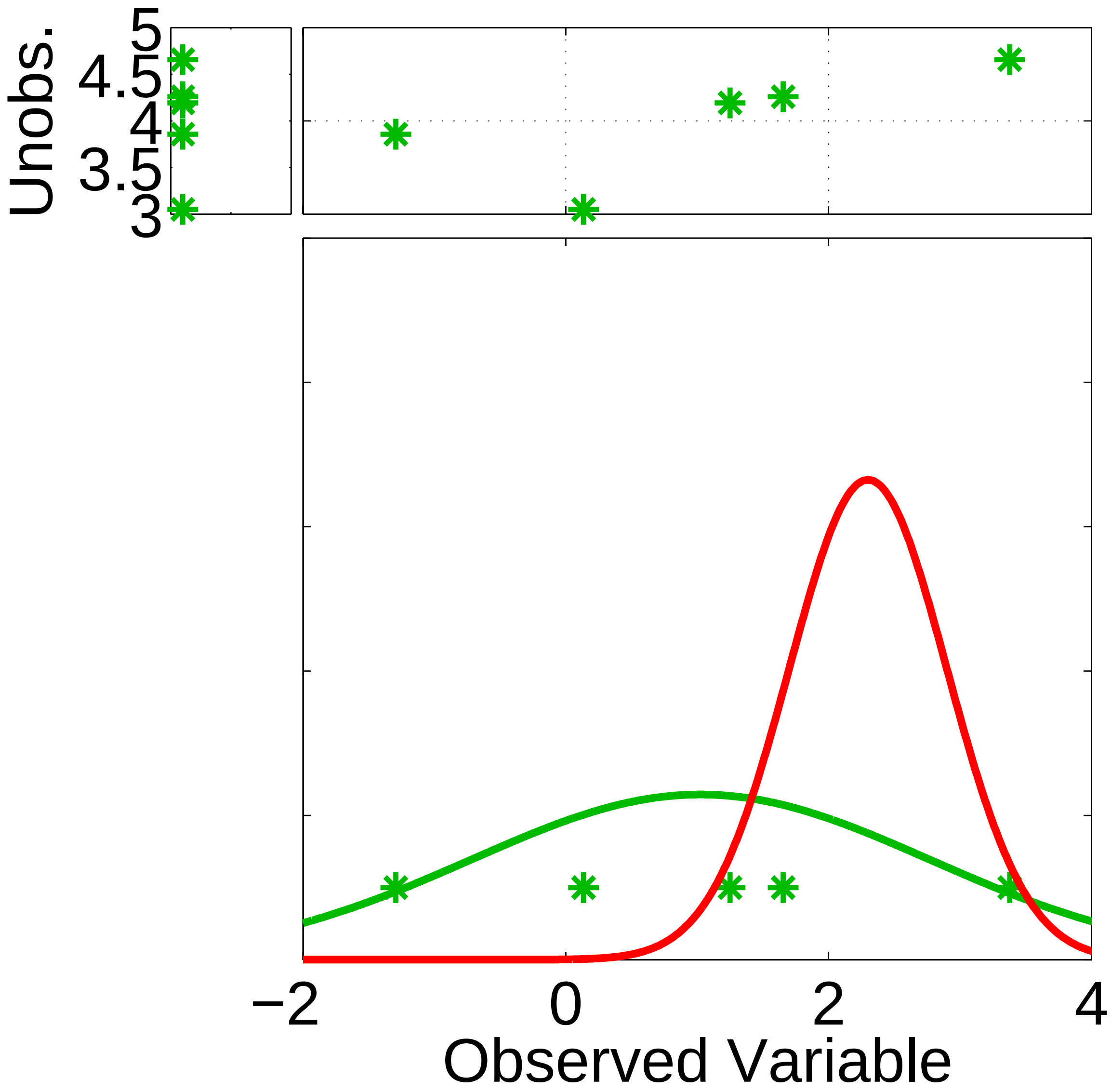


Assume that all we know is prior joint distribution.

One variable is observed.

Update observed variable with one of previous methods.

Ensemble filters: Updating additional prior state variables

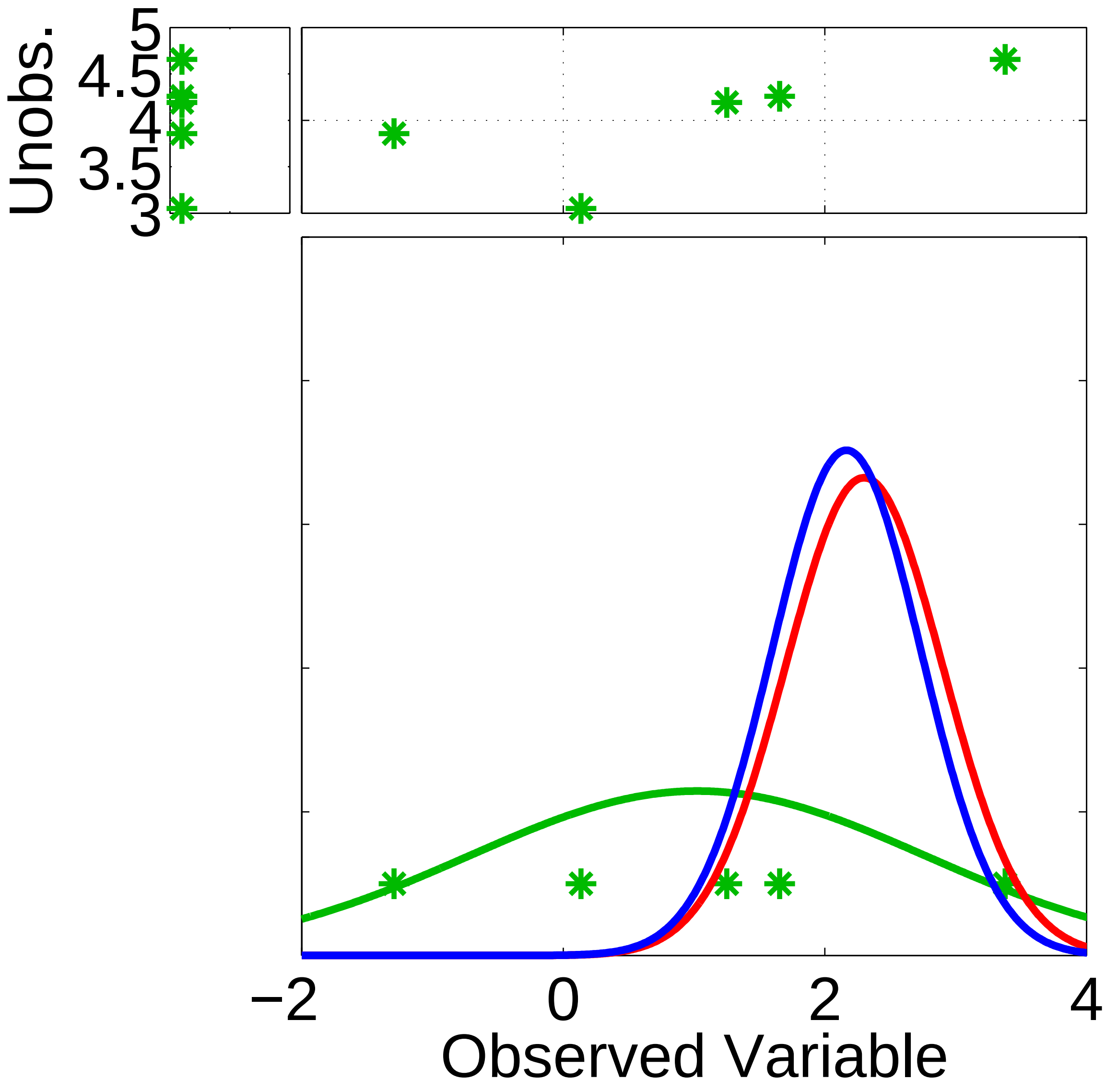


Assume that all we know is prior joint distribution.

One variable is observed.

Update observed variable with one of previous methods.

Ensemble filters: Updating additional prior state variables

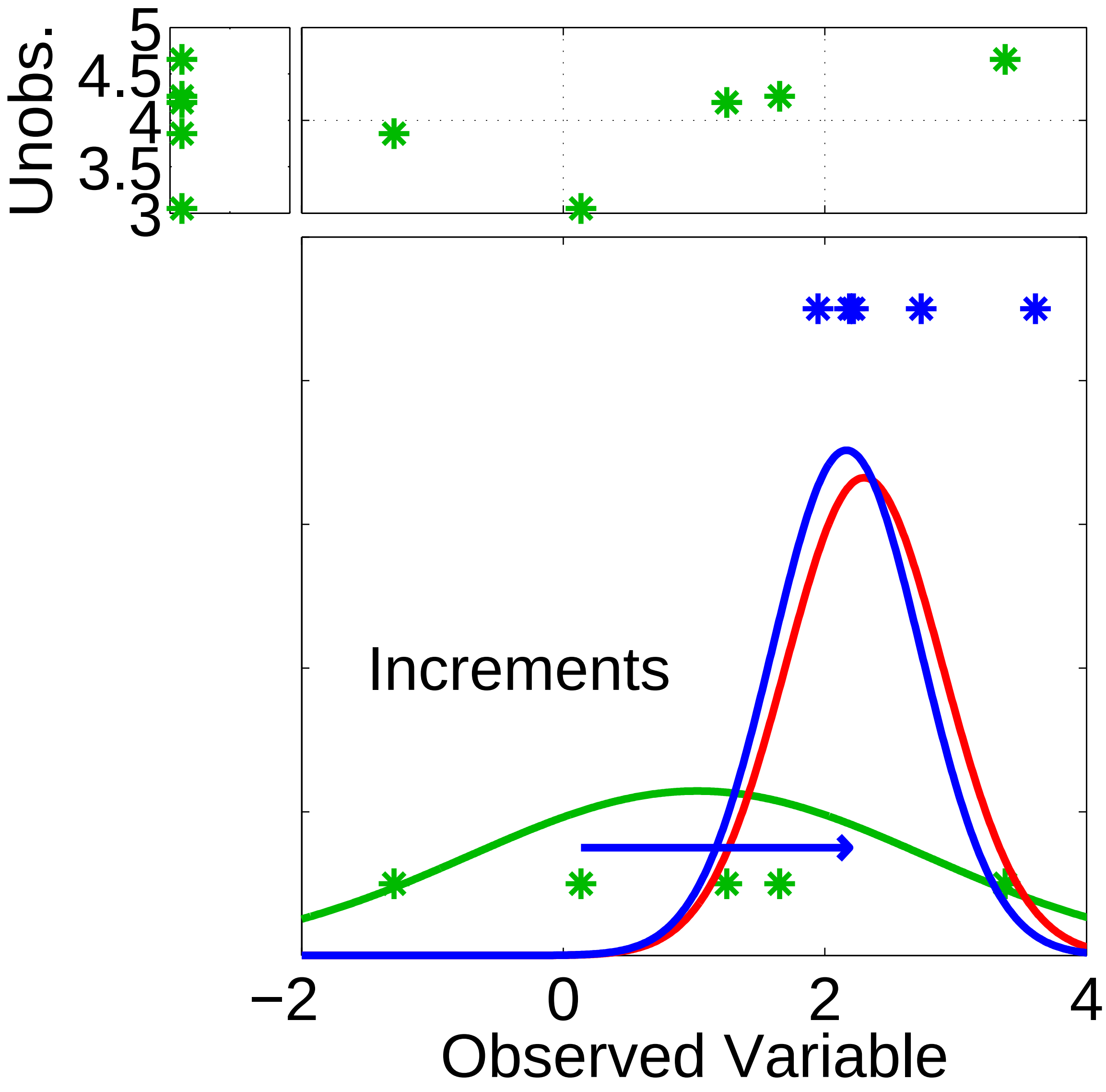


Assume that all we know is prior joint distribution.

One variable is observed.

Update observed variable with one of previous methods.

Ensemble filters: Updating additional prior state variables

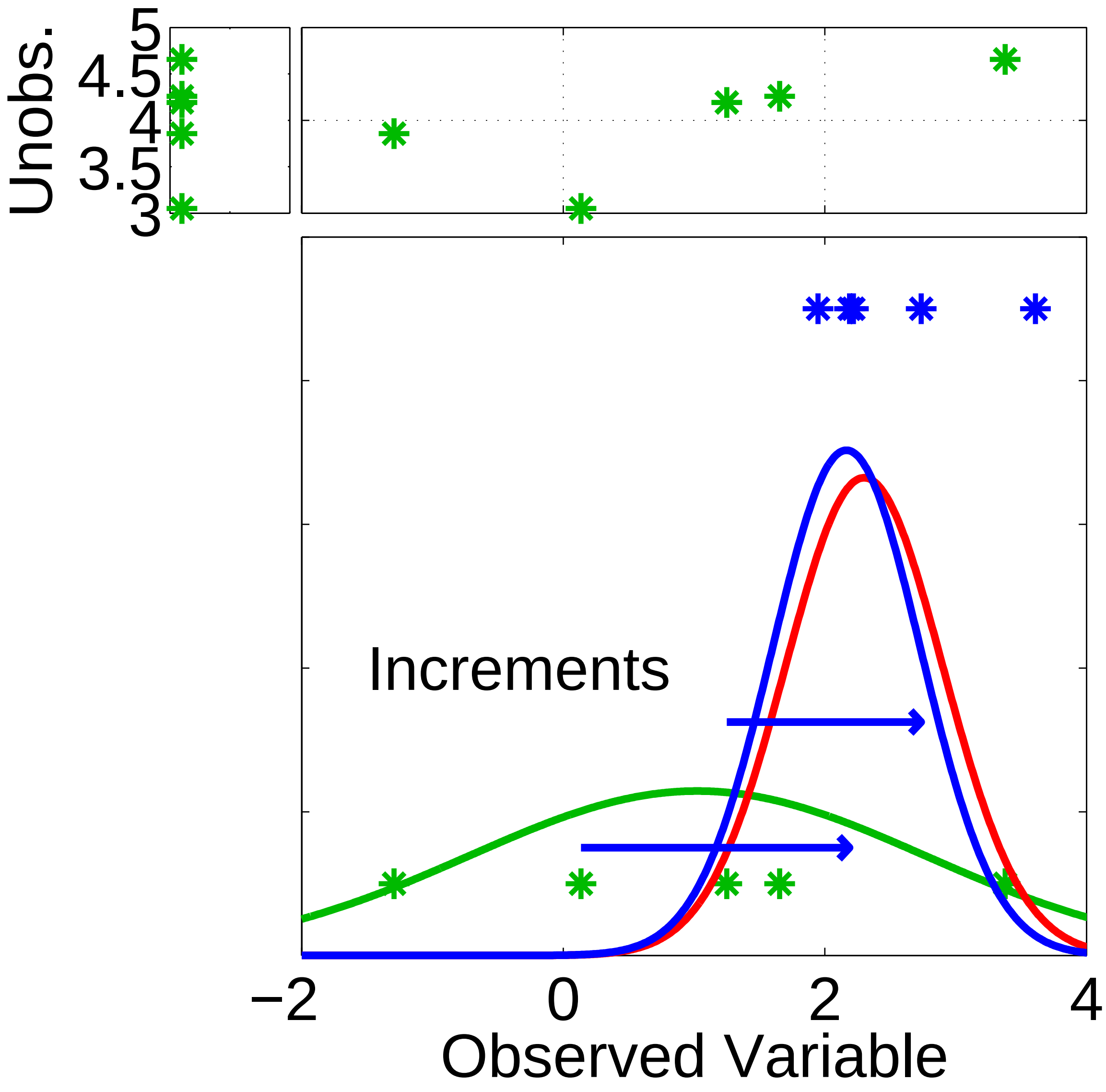


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

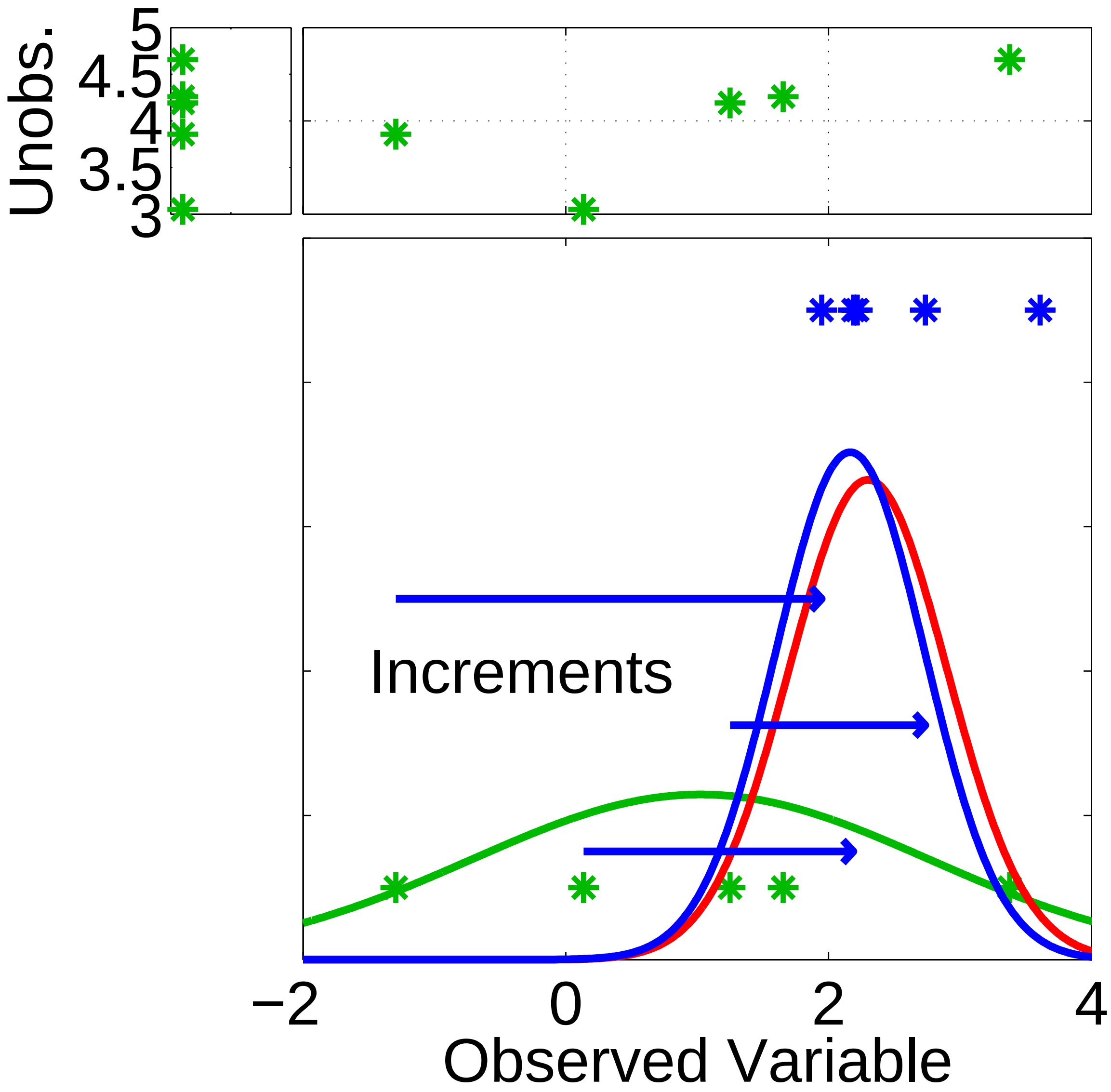


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

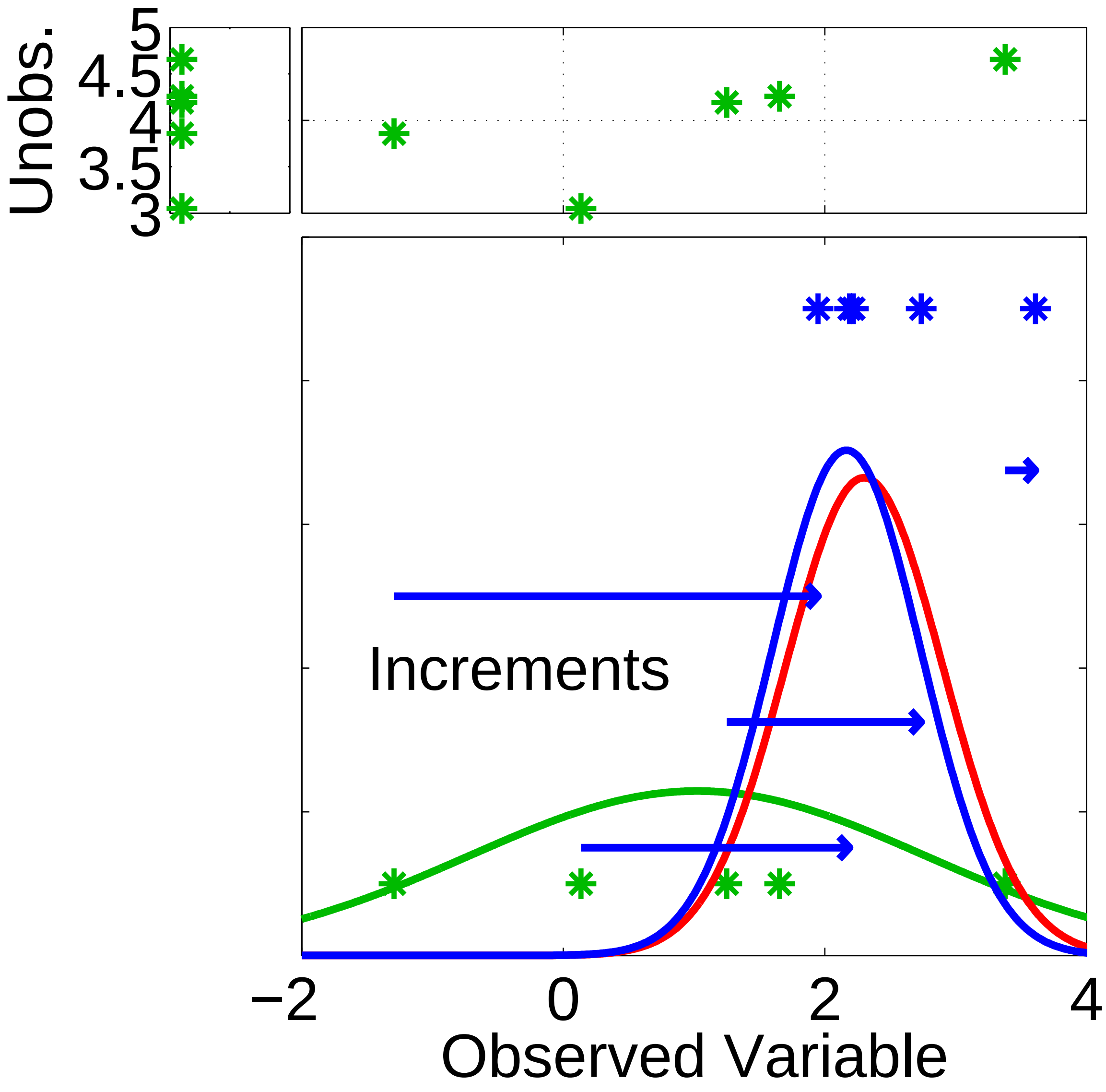


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

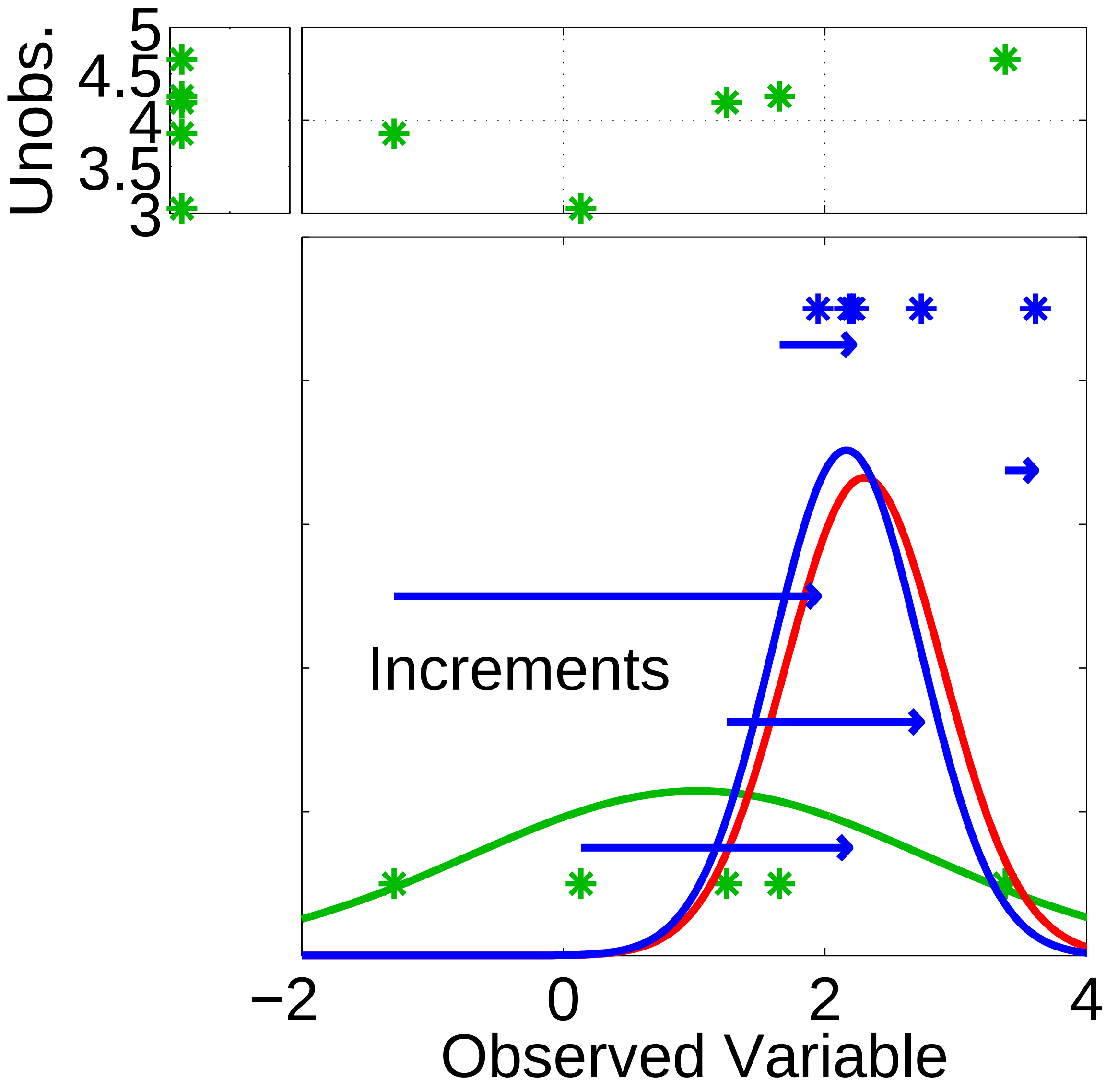


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

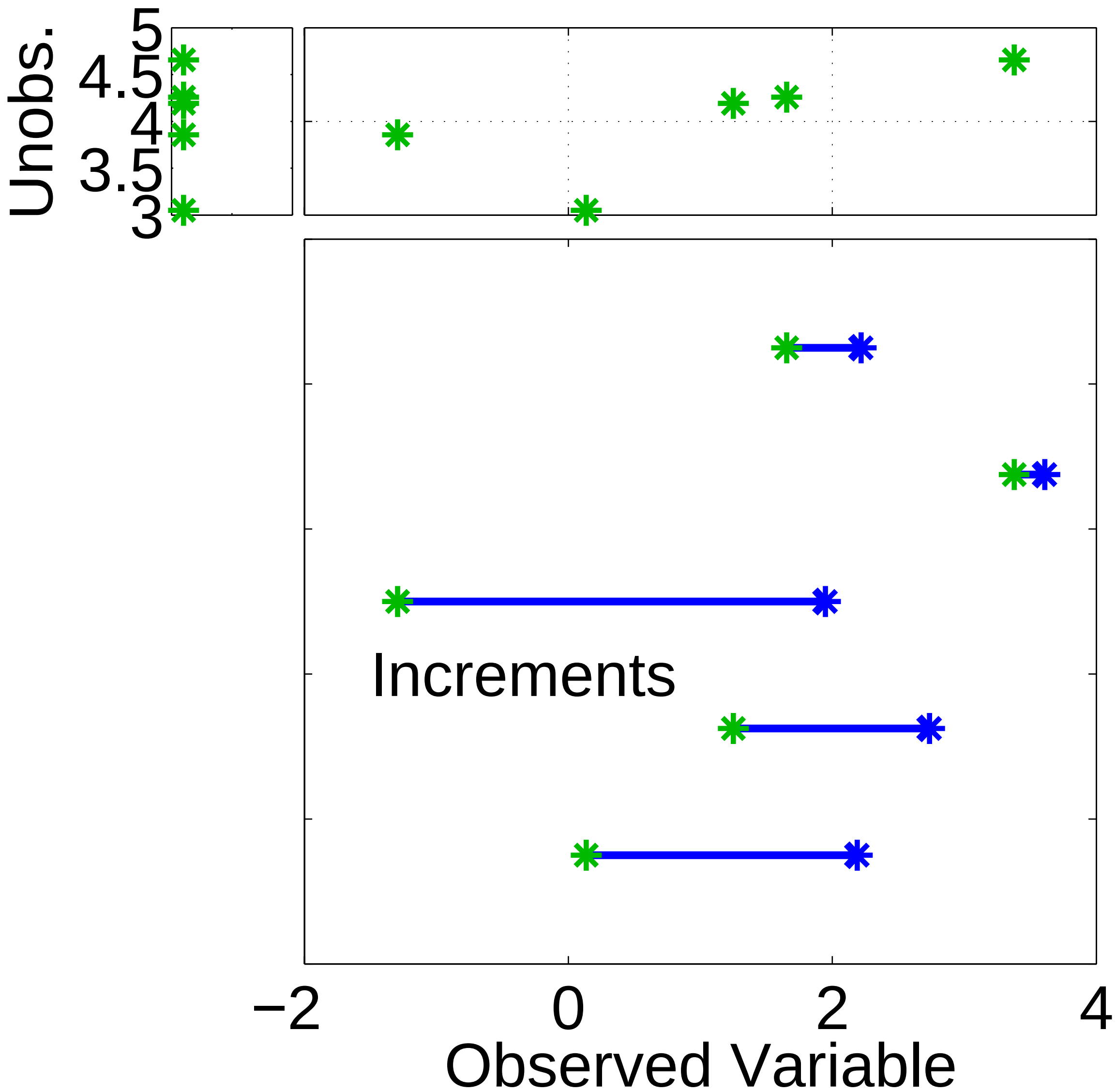


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

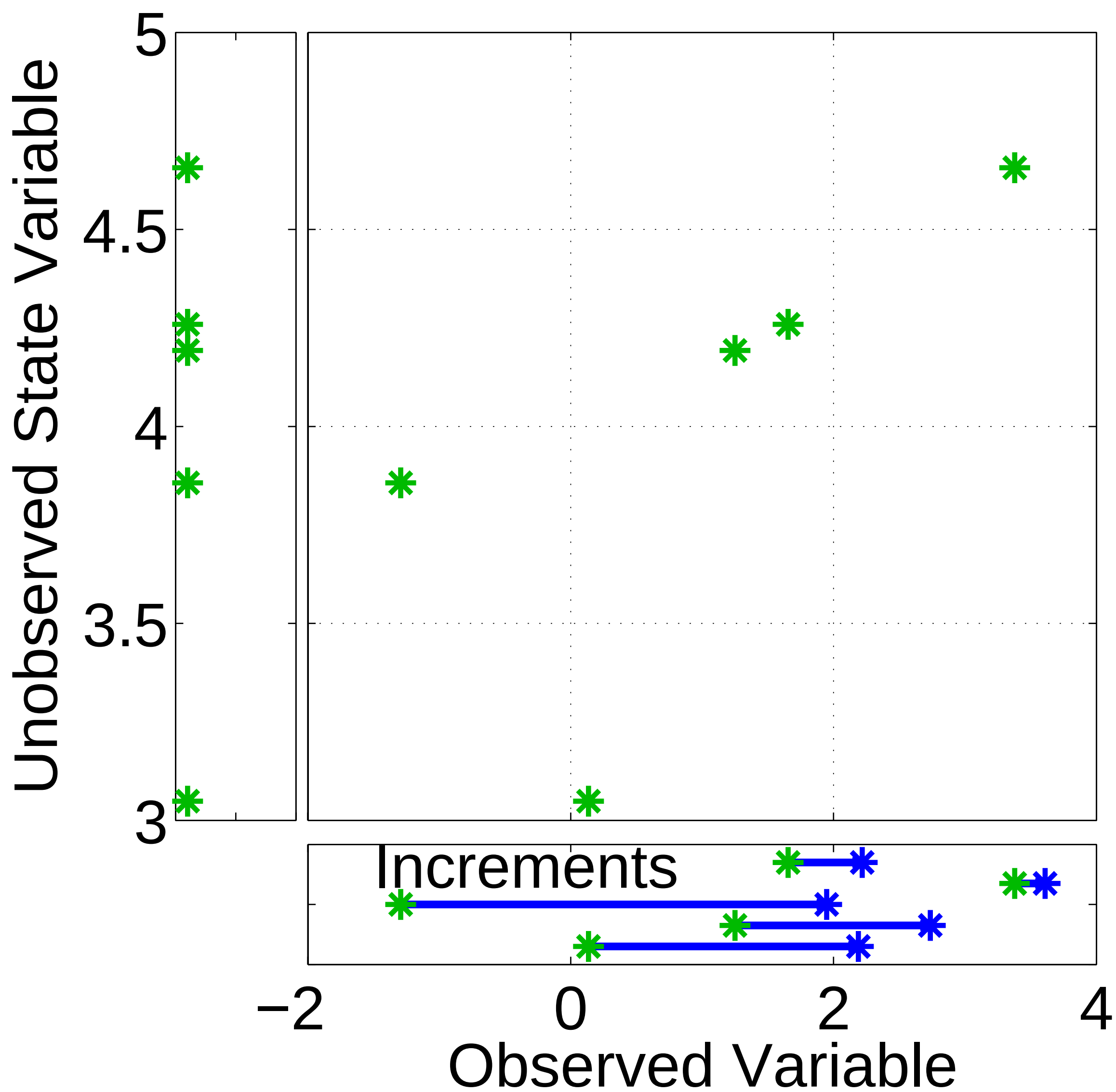


Assume that all we know is prior joint distribution.

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.

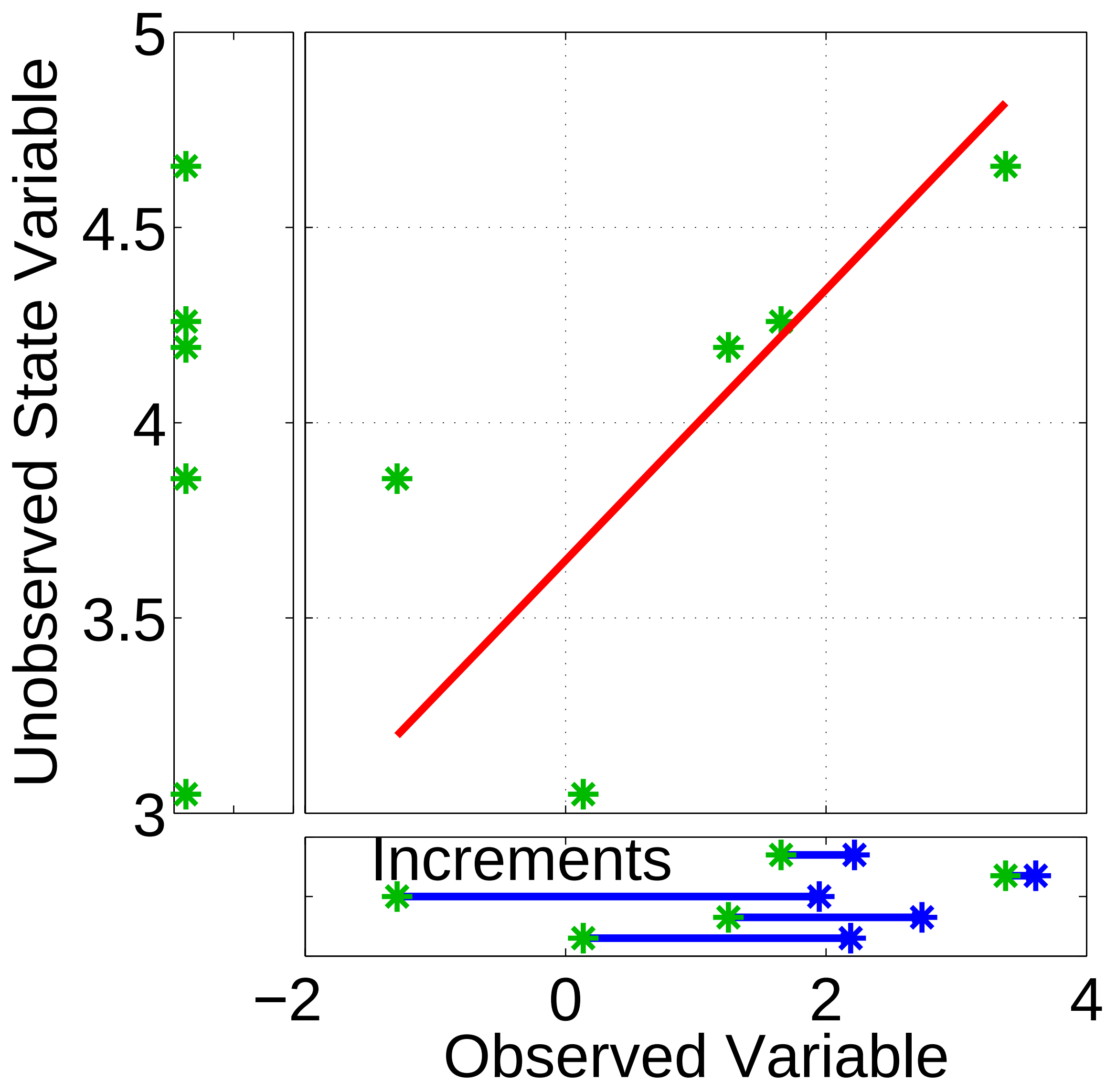
How should the unobserved variable be impacted?

First choice: least squares

Equivalent to linear regression.

Same as assuming binormal prior.

Ensemble filters: Updating additional prior state variables



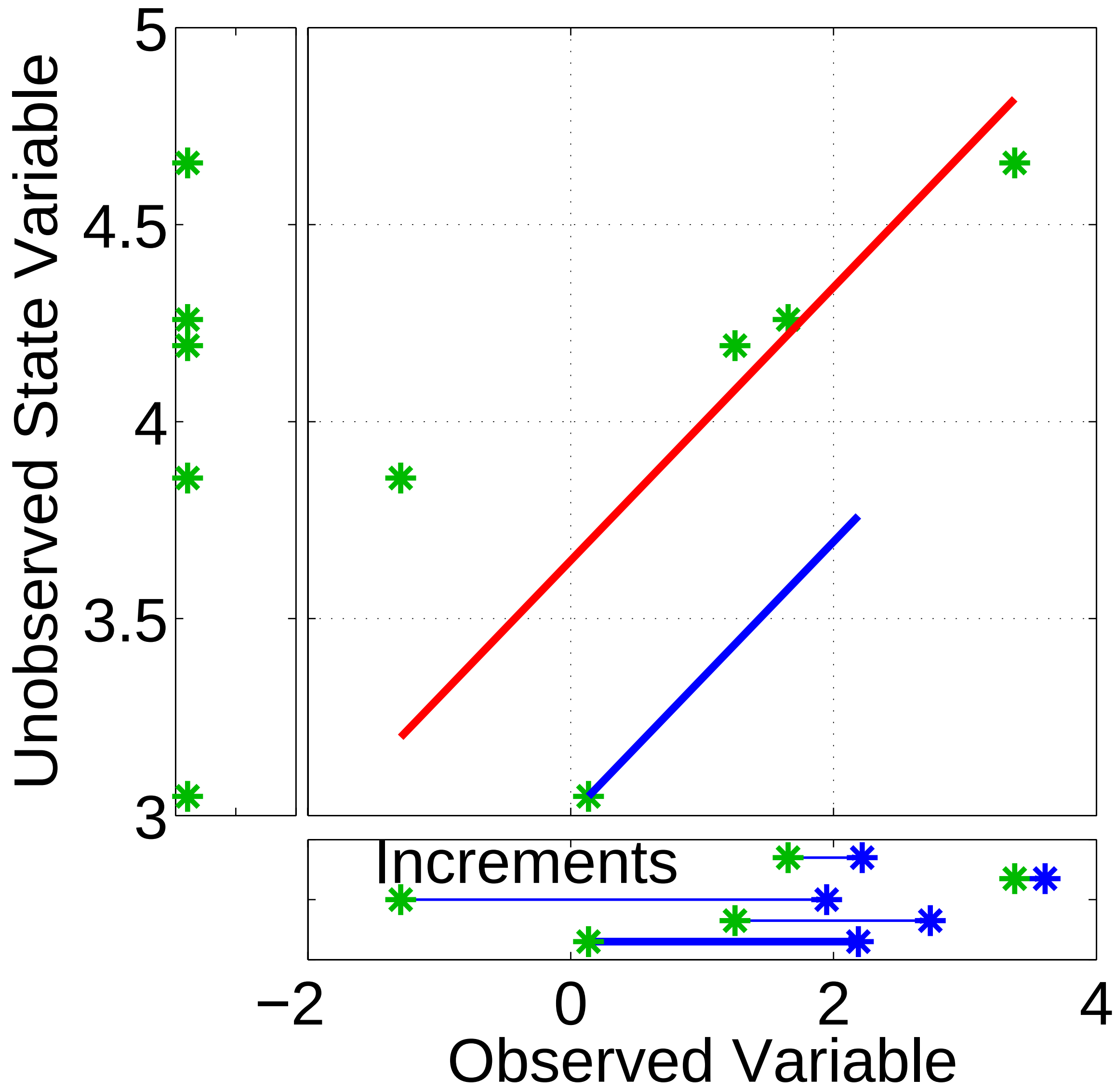
Have joint prior distribution of two variables.

How should the unobserved variable be impacted?

First choice: least squares

Begin by finding least squares fit.

Ensemble filters: Updating additional prior state variables

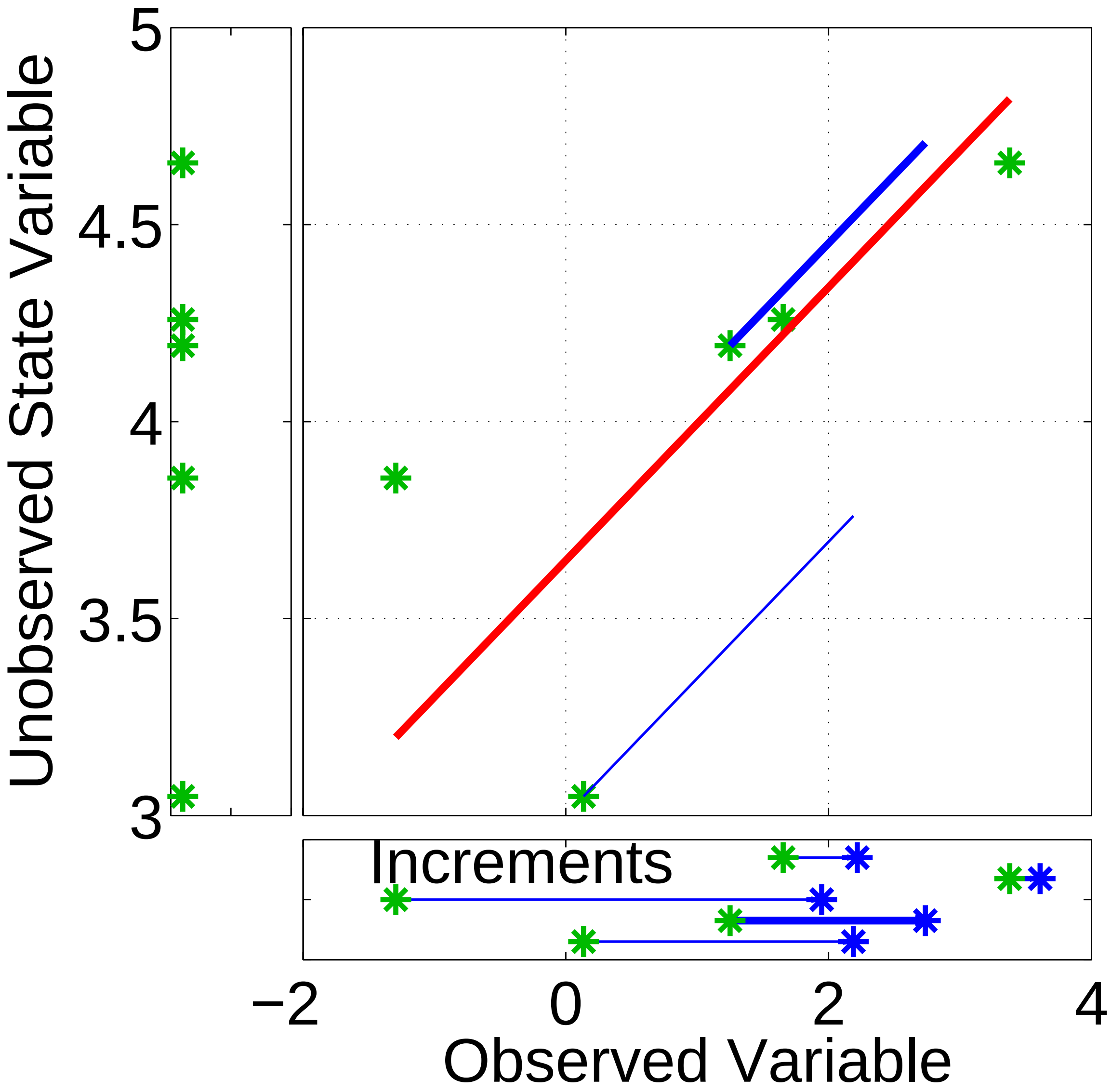


Have joint prior distribution of two variables.

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

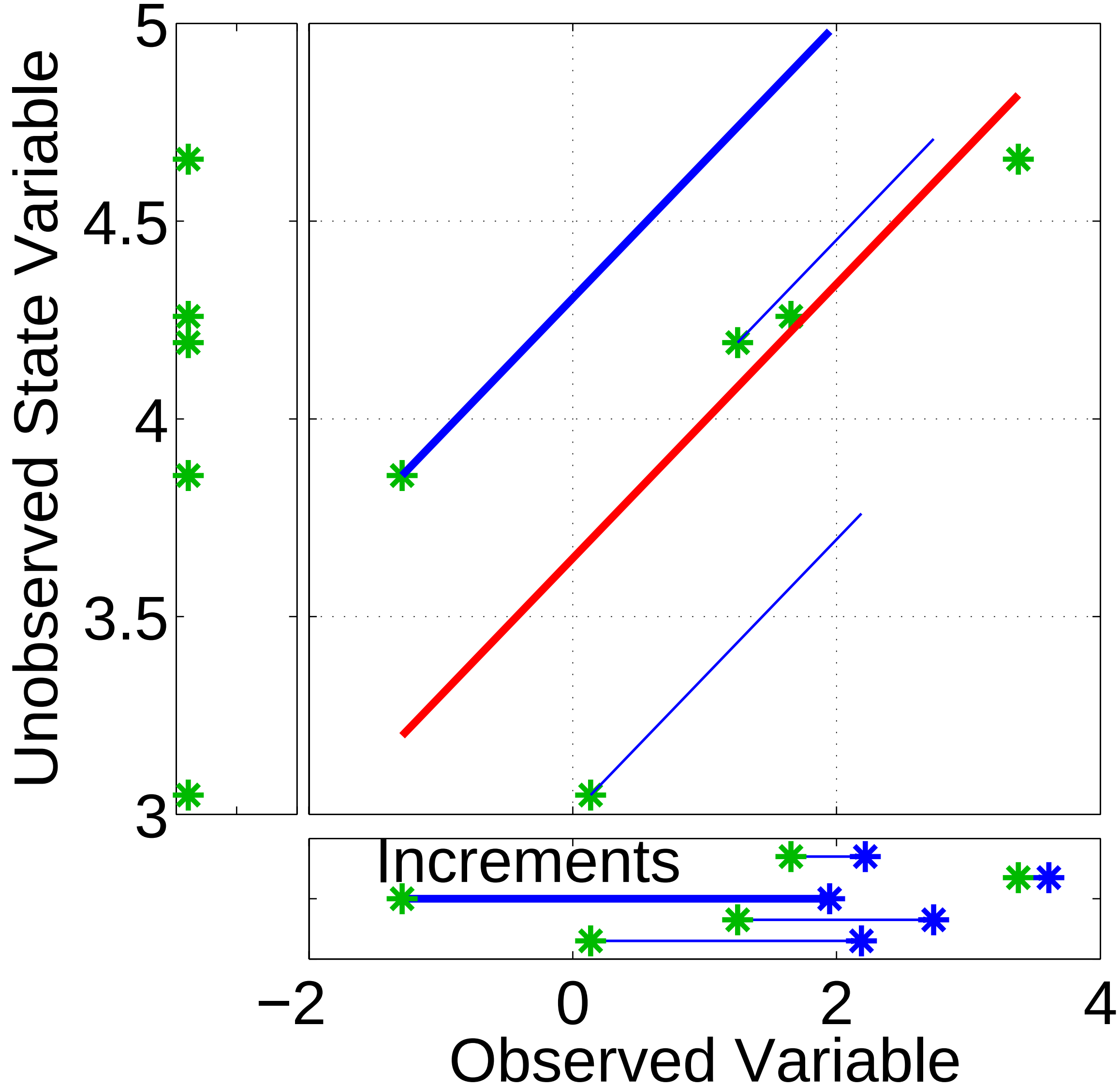


Have joint prior distribution of two variables.

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

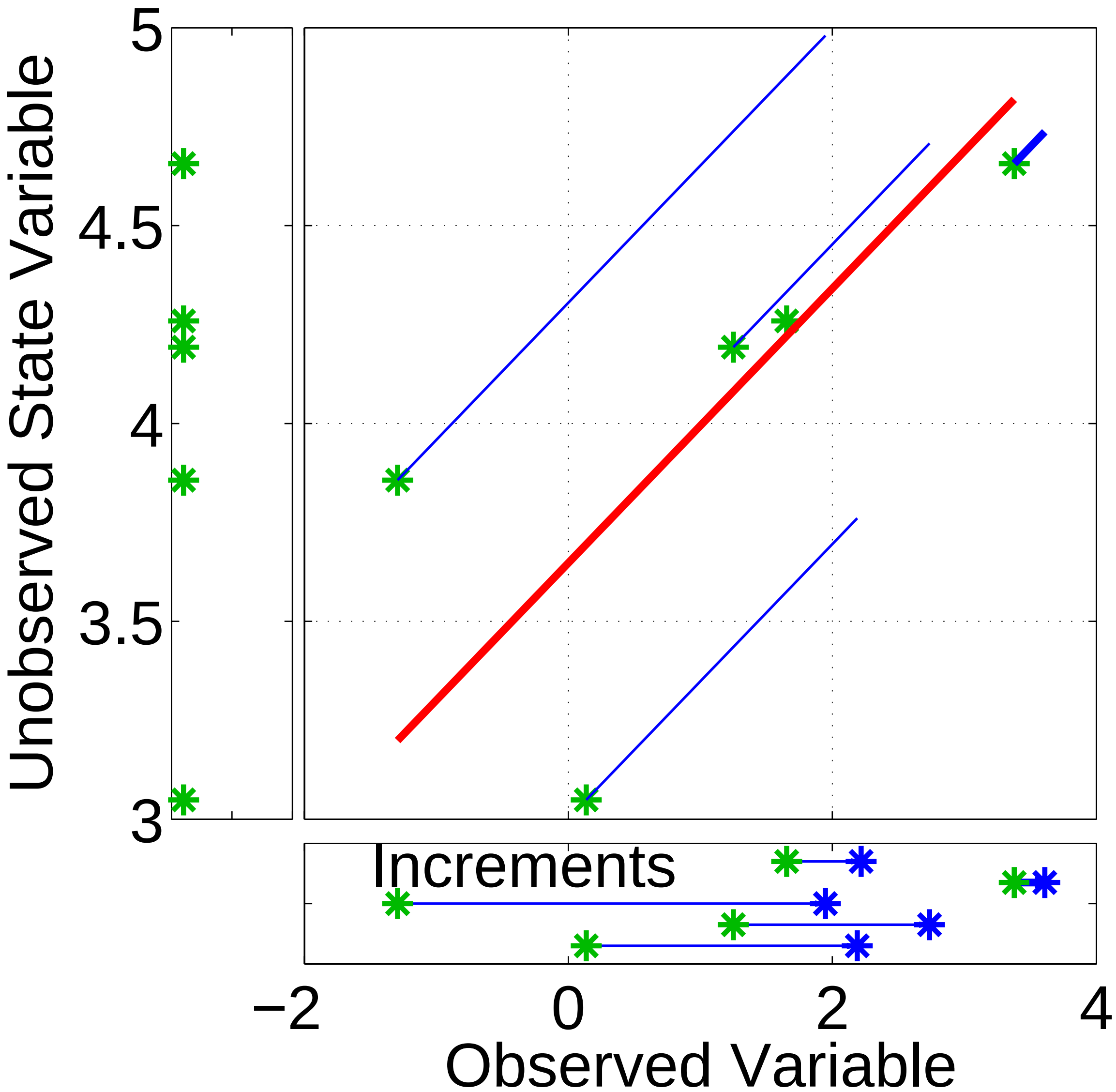


Have joint prior distribution of two variables.

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

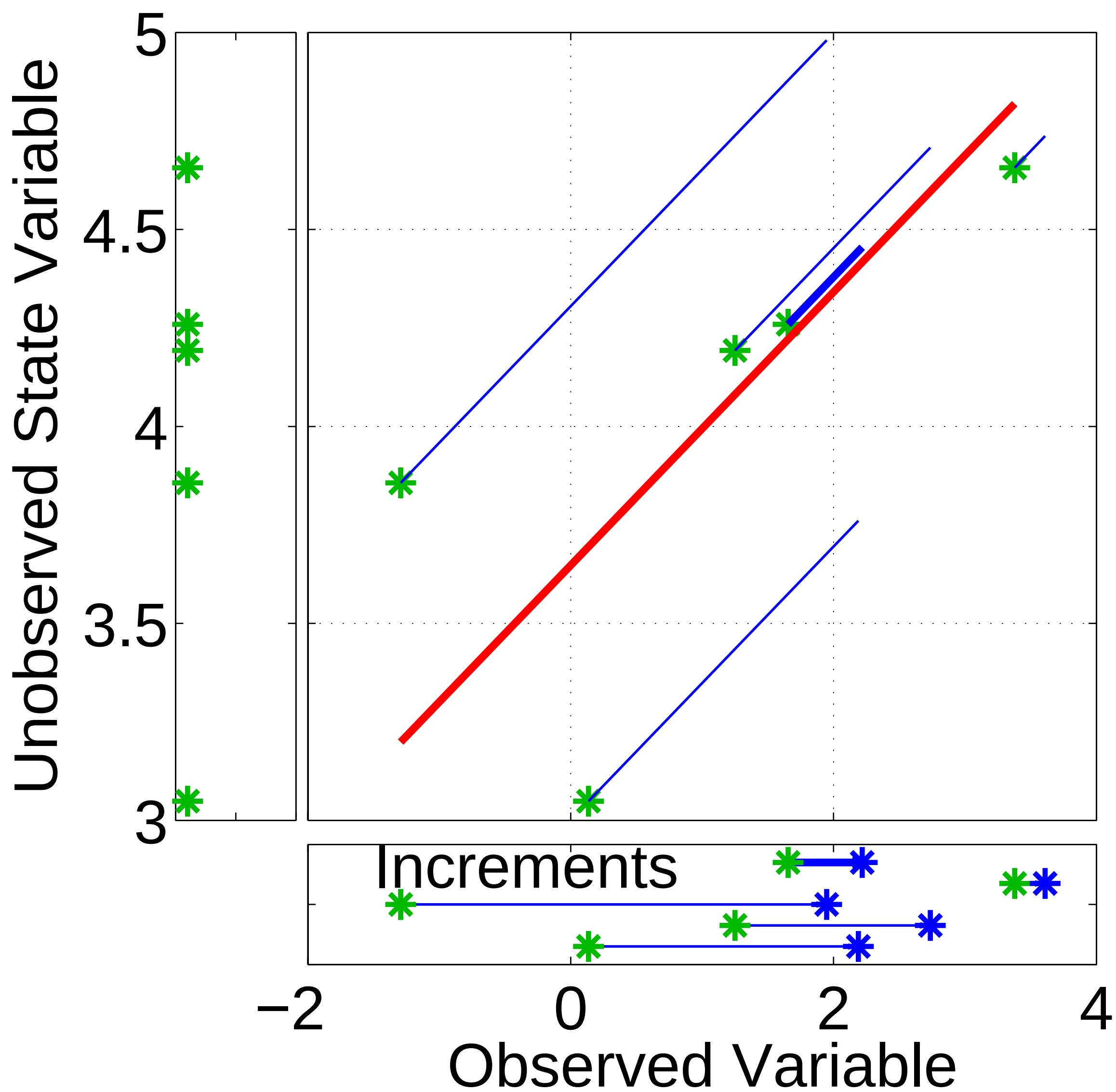


Have joint prior distribution of two variables.

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

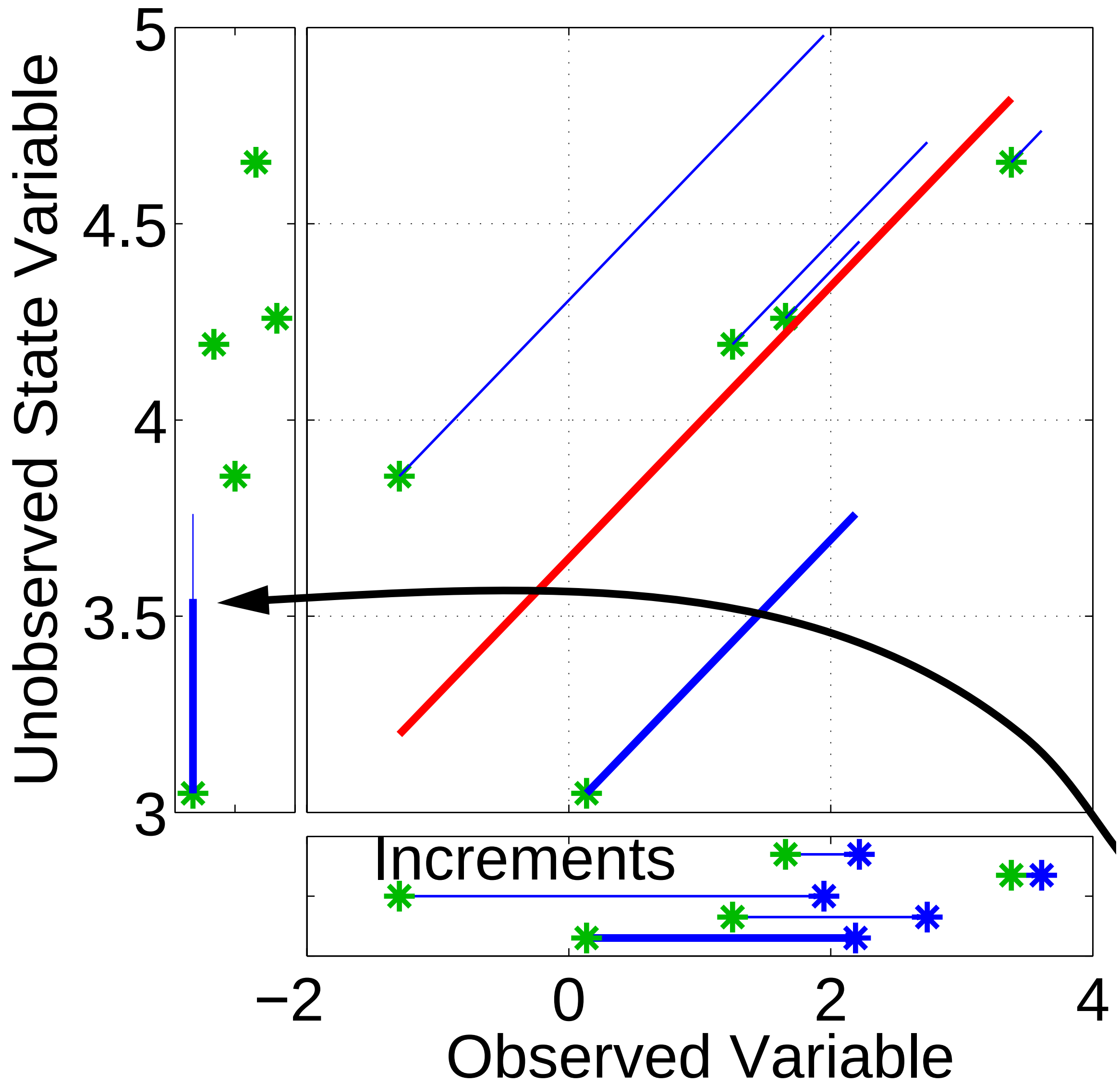


Have joint prior distribution of two variables.

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



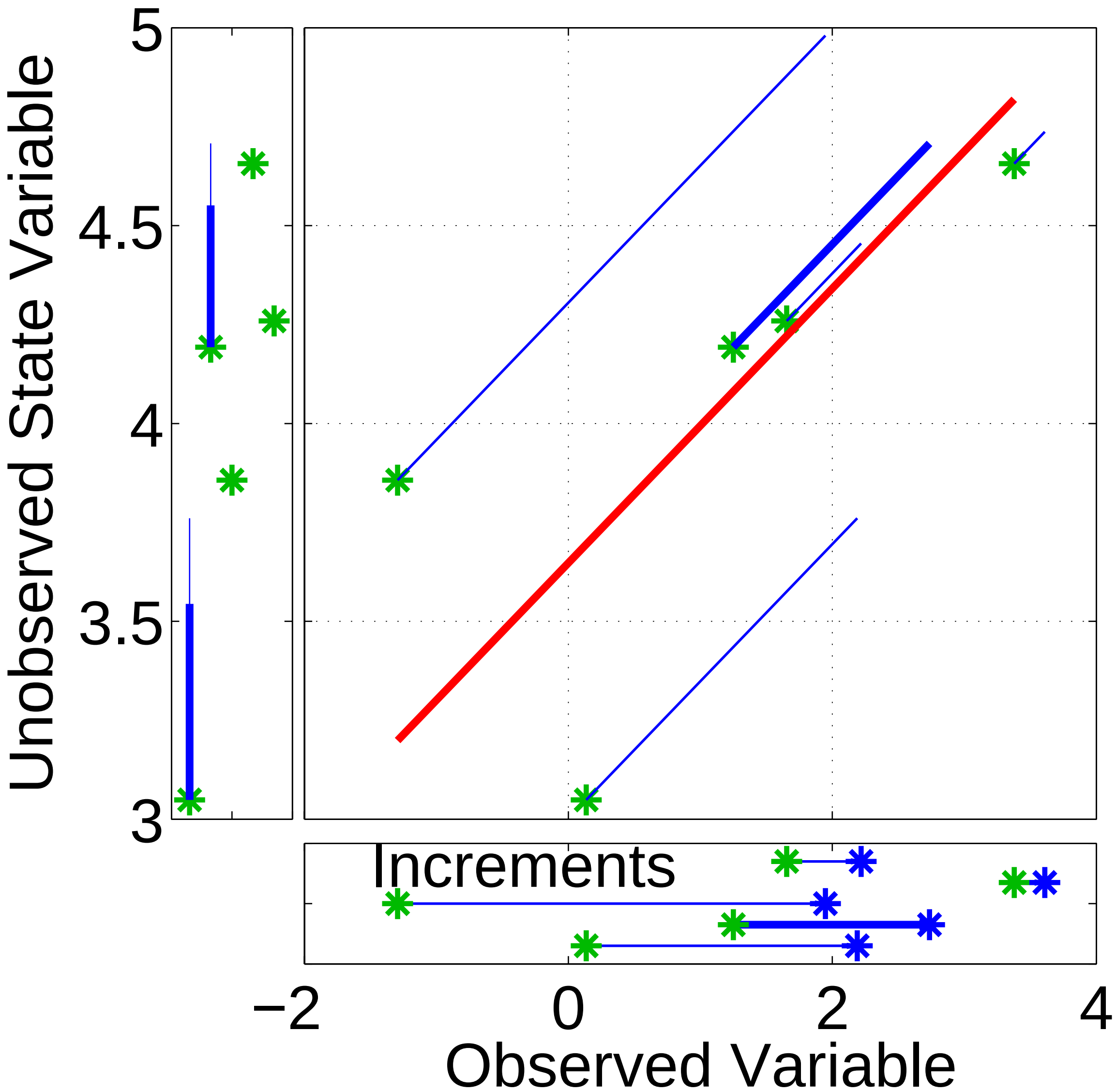
Have joint prior distribution of two variables.

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

Then projecting from joint space onto unobserved priors.

Finally, multiply by prior sample correlation.

Ensemble filters: Updating additional prior state variables



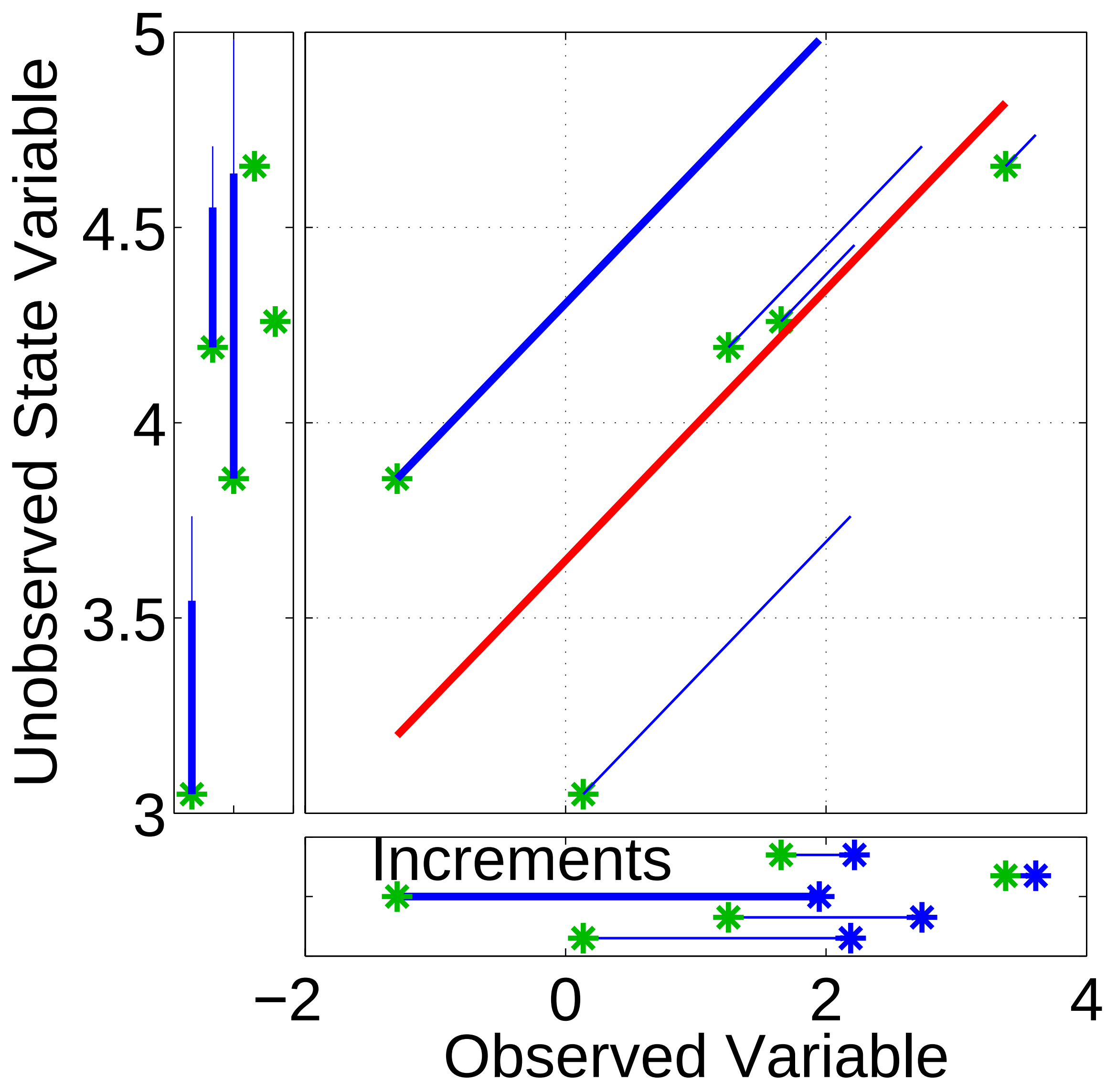
Have joint prior distribution of two variables.

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

Then projecting from joint space onto unobserved priors.

Finally, multiply by prior sample correlation.

Ensemble filters: Updating additional prior state variables



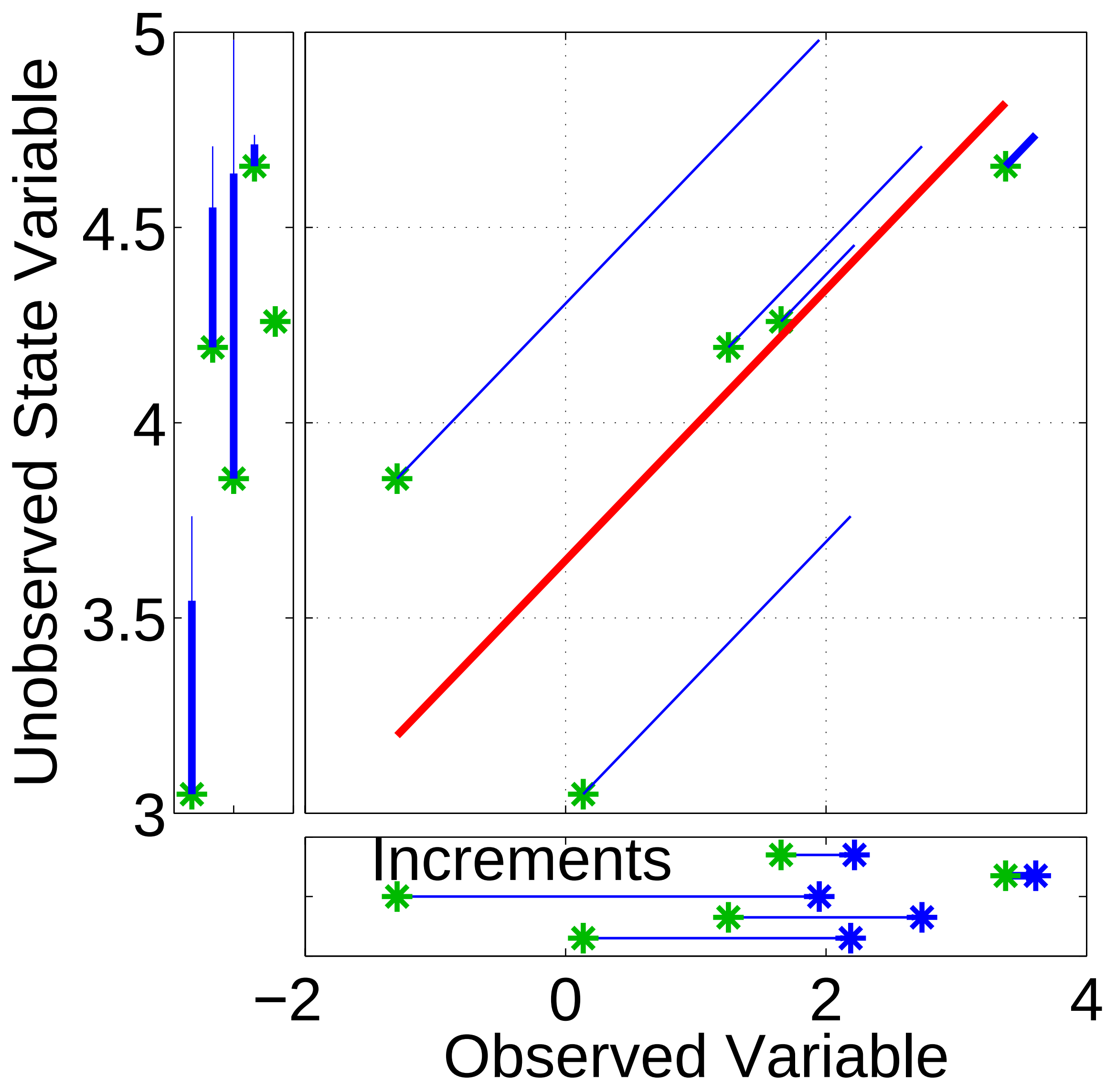
Have joint prior distribution of two variables.

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

Then projecting from joint space onto unobserved priors.

Finally, multiply by prior sample correlation.

Ensemble filters: Updating additional prior state variables



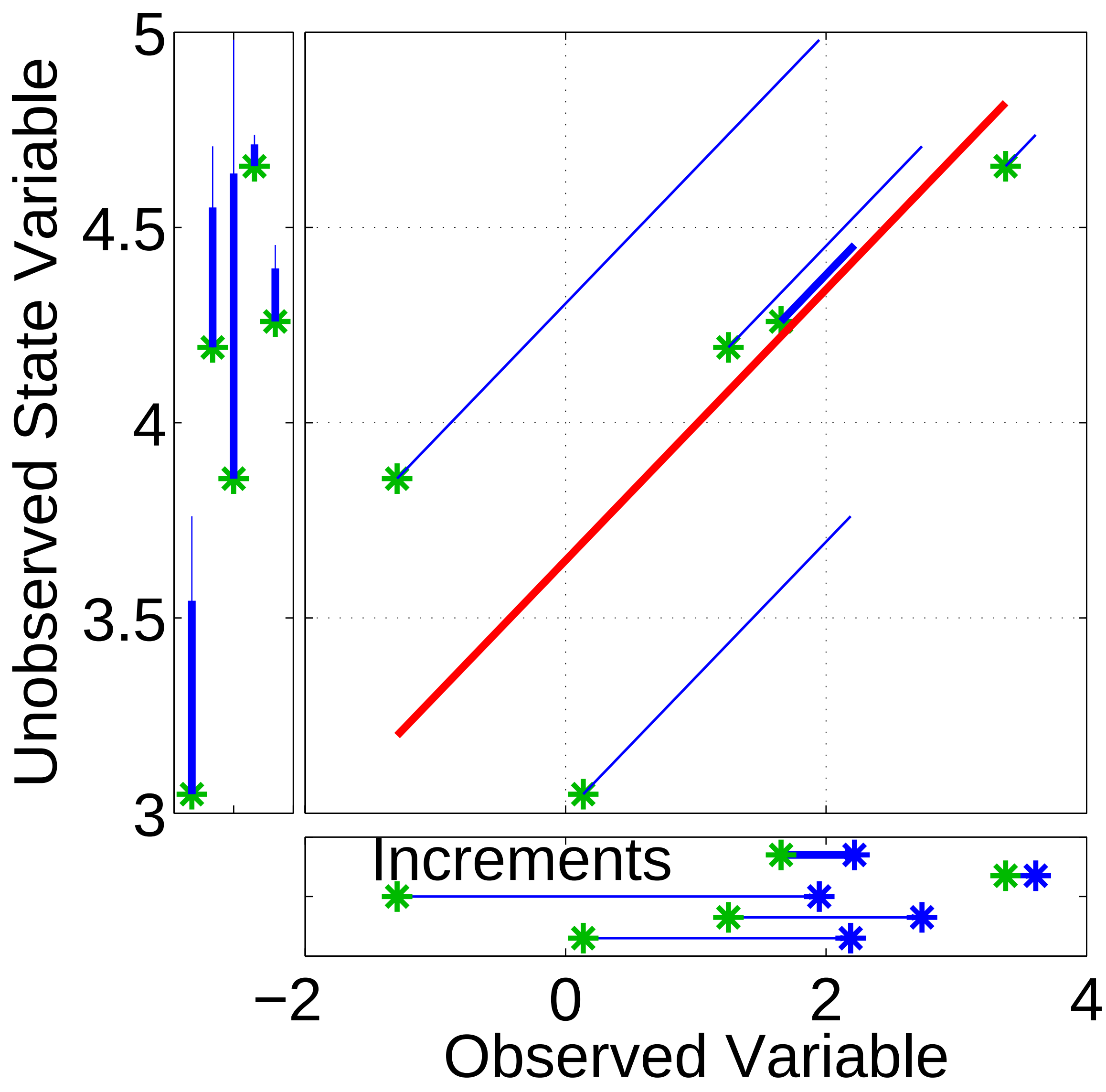
Have joint prior distribution of two variables.

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

Then projecting from joint space onto unobserved priors.

Finally, multiply by prior sample correlation.

Ensemble filters: Updating additional prior state variables



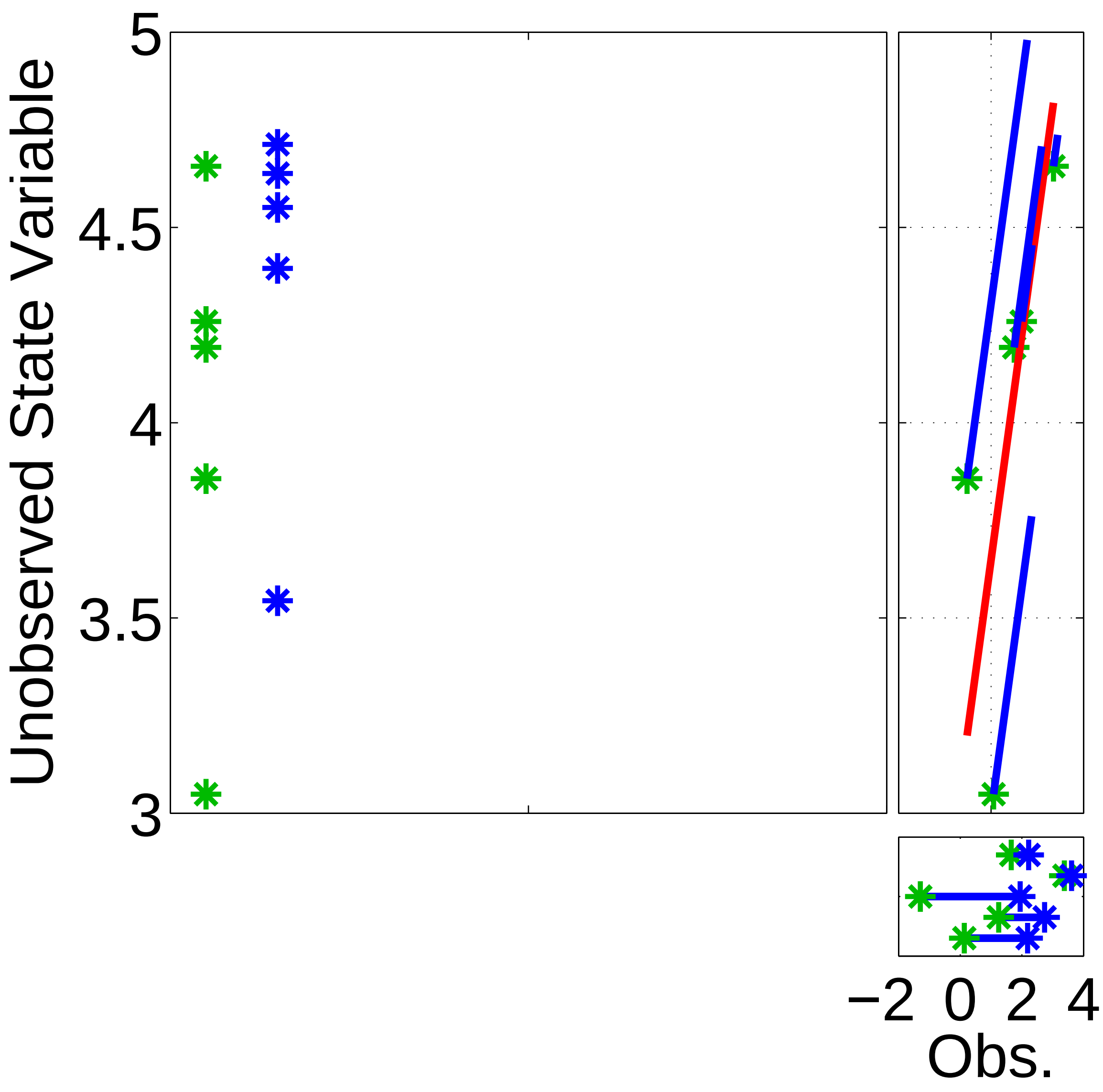
Have joint prior distribution of two variables.

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

Then projecting from joint space onto unobserved priors.

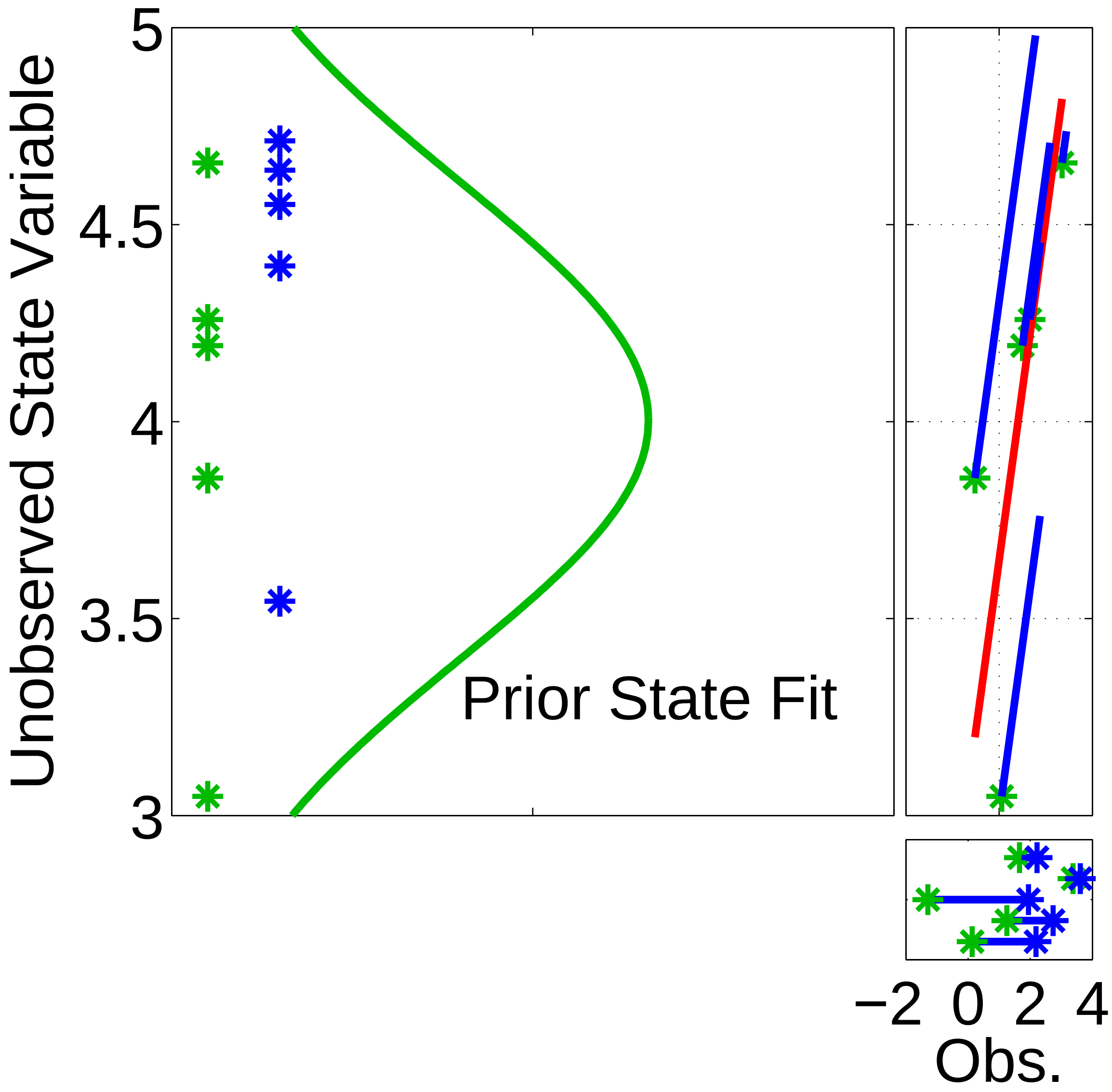
Finally, multiply by prior sample correlation.

Ensemble filters: Updating additional prior state variables



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

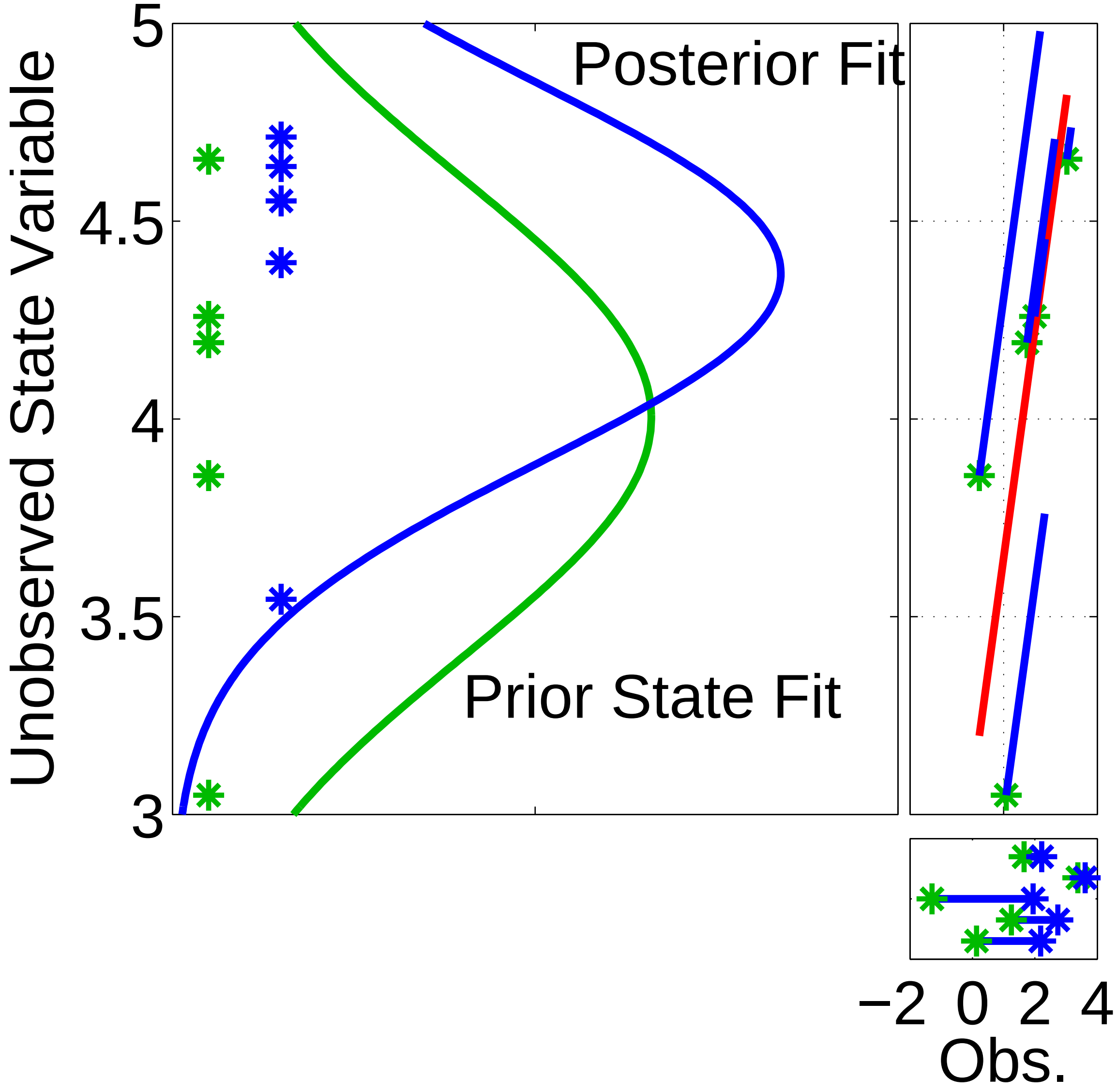
Ensemble filters: Updating additional prior state variables



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

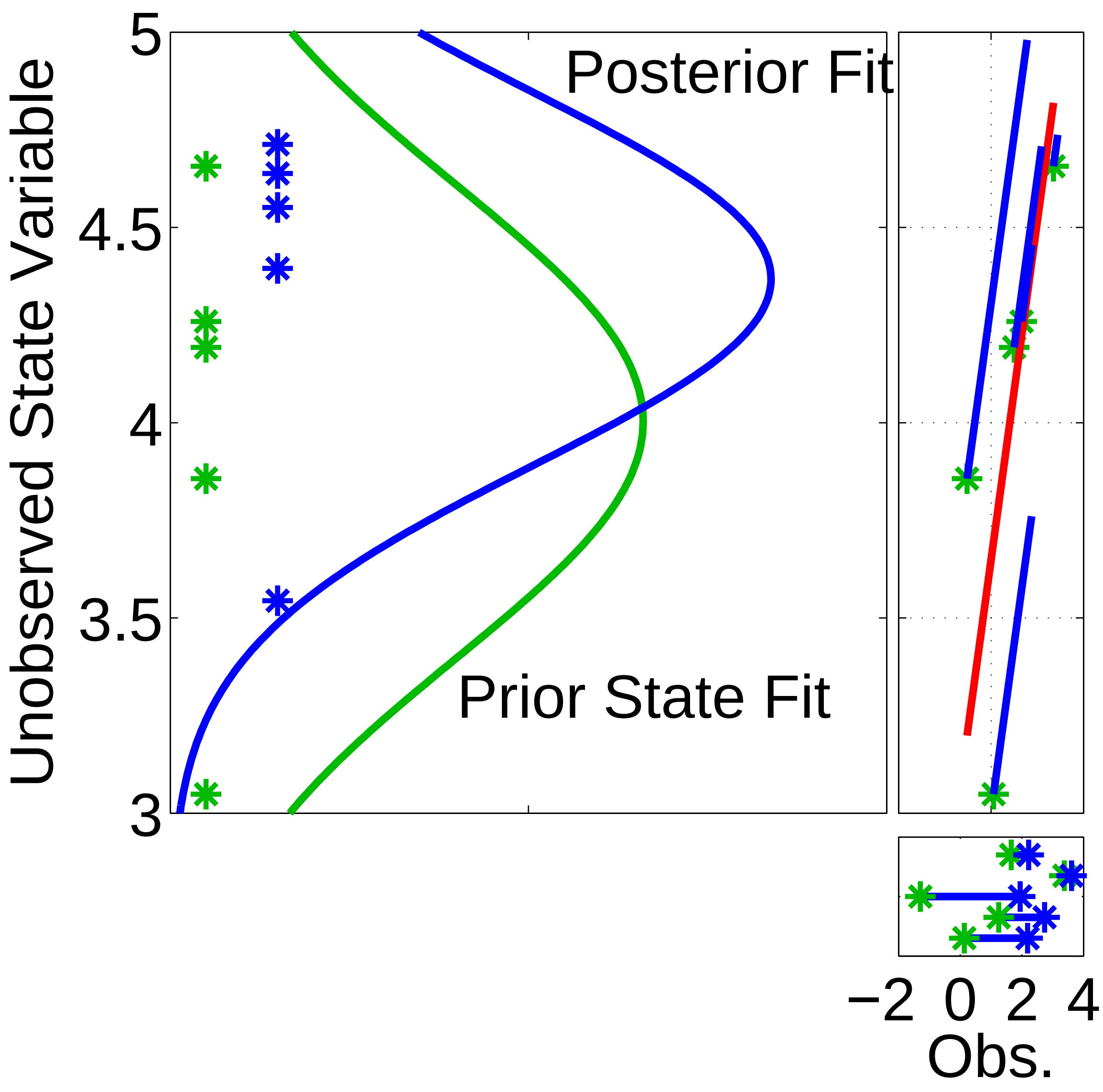


Now have an updated (posterior) ensemble for the unobserved variable.

Fitting Gaussians shows that mean and variance have changed.

Other features of the prior distribution may also have changed.

Ensemble filters: Updating additional prior state variables



CRITICAL POINT:

Since impact on unobserved variable is simply a linear regression, can do this INDEPENDENTLY for any number of unobserved variables!

Could also do many at once using matrix algebra as in traditional Kalman Filter.

Ensemble filters: Updating additional prior state variables

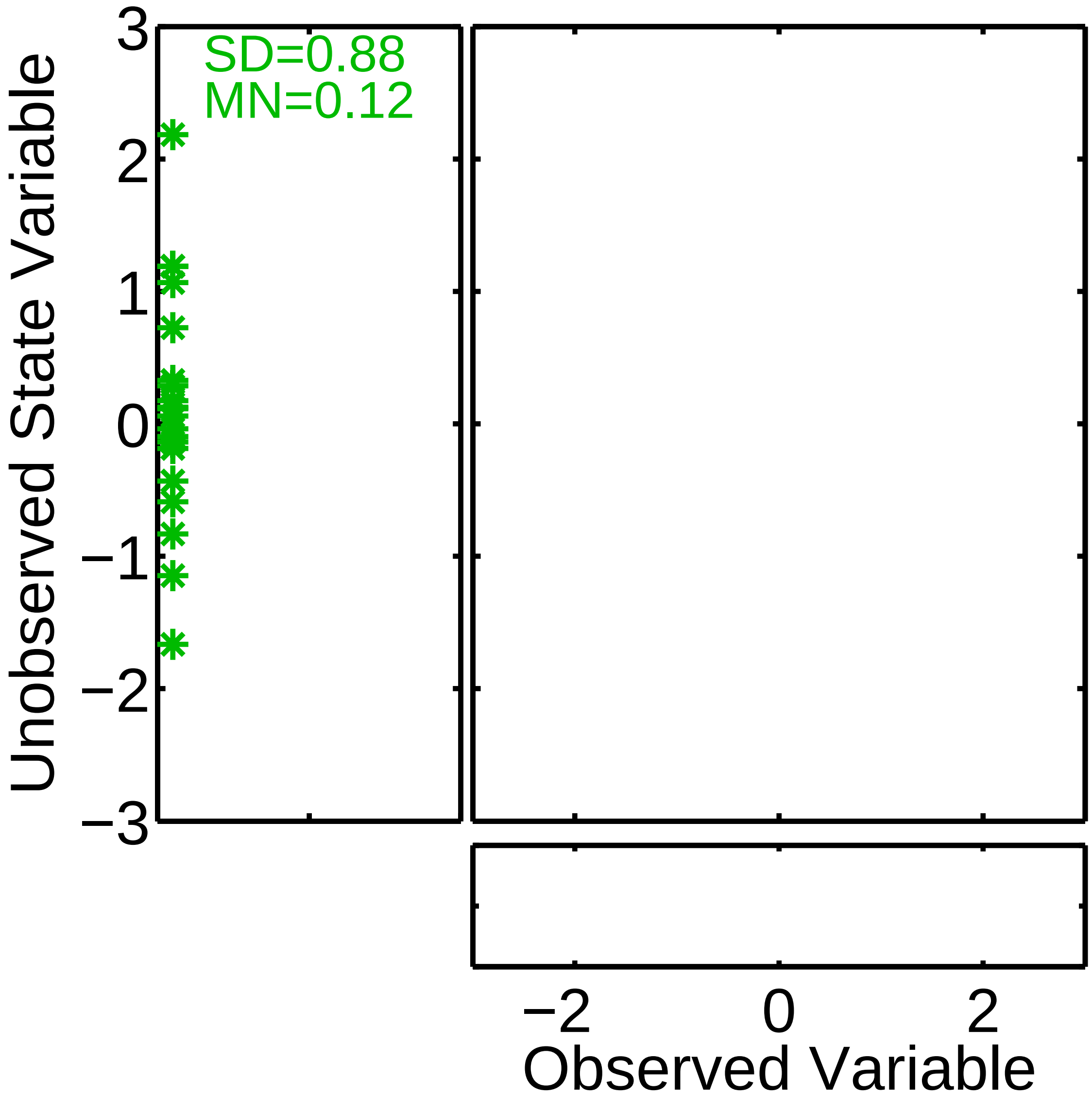
Two primary error sources:

1. Linear approximation is invalid.
   Substantial nonlinearity in ‘true’ relation over range of prior.

2. Sampling error due to noise.
   Even if linear relation, sample regression coefficient imprecise.

May need to address both issues for good performance.

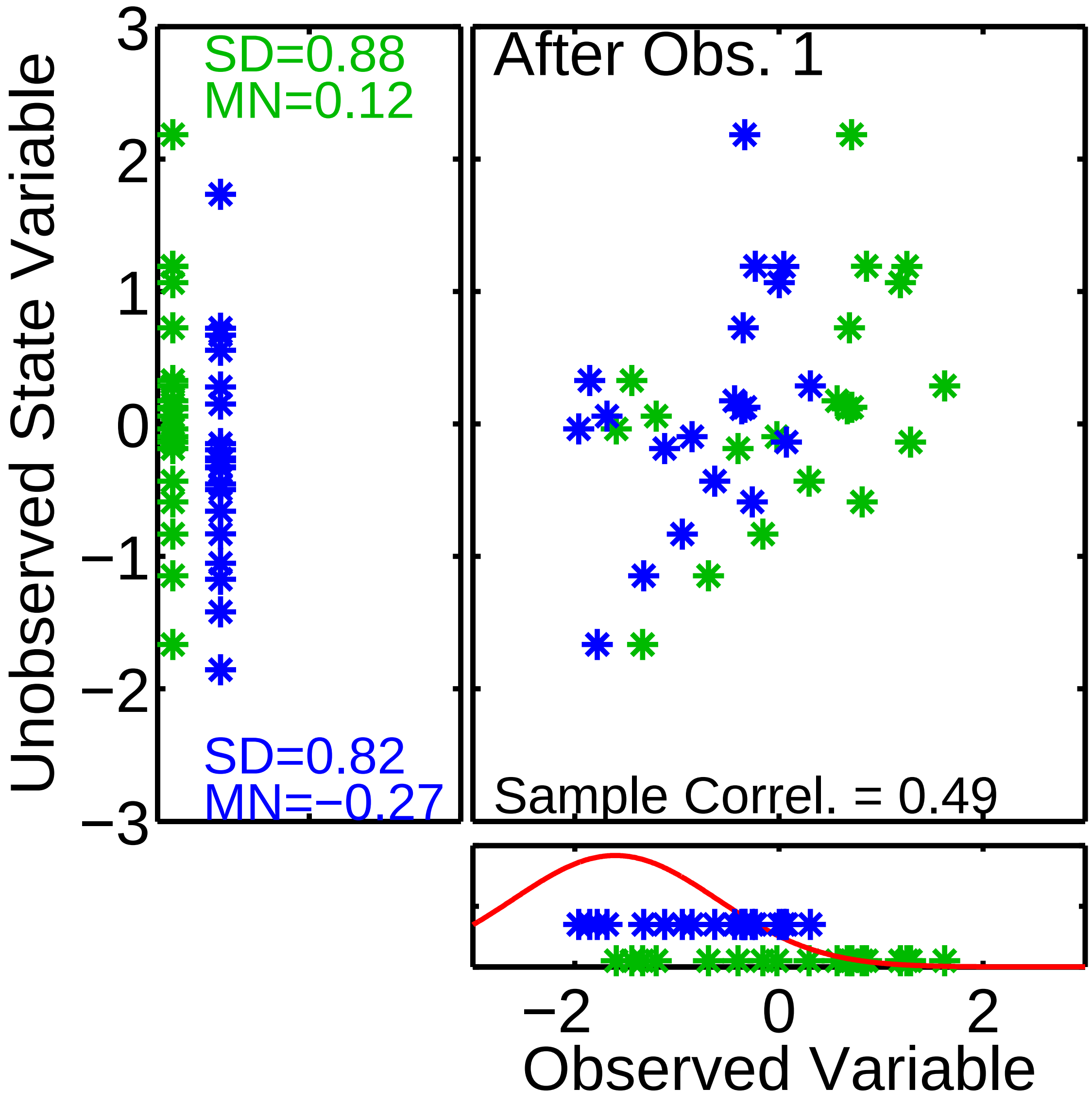
Regression sampling error and filter divergence



Suppose unobserved state variable is known to be unrelated to set of observed variables.

Unobserved variable should remain unchanged.

Regression sampling error and filter divergence

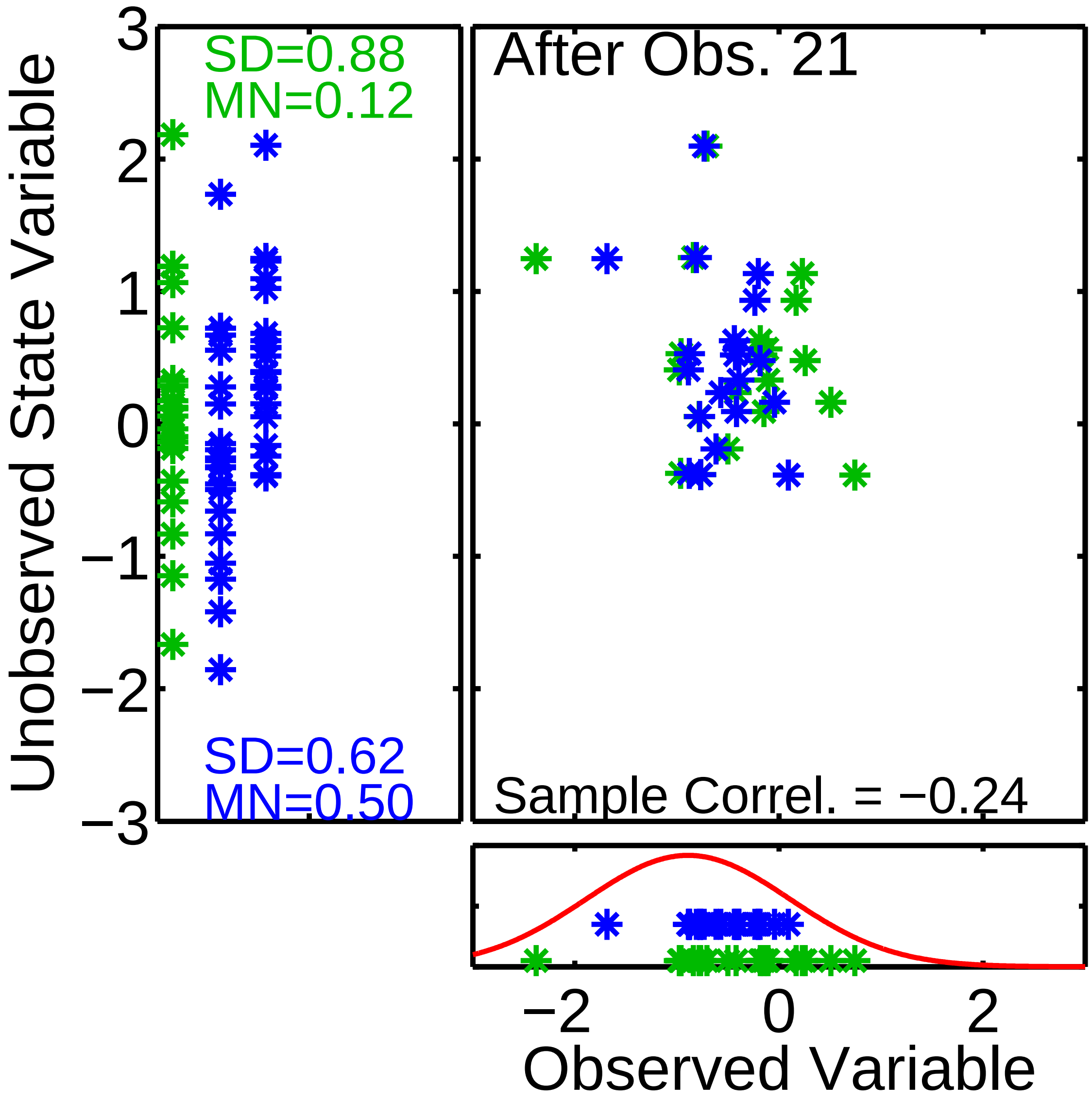


Suppose unobserved state variable is known to be unrelated to set of observed variables.

Finite samples from joint distribution will have non-zero correlation (expected |corr| = 0.19 for 20 samples).

After one observation, unobs. variable mean and S.D. change.

Regression sampling error and filter divergence

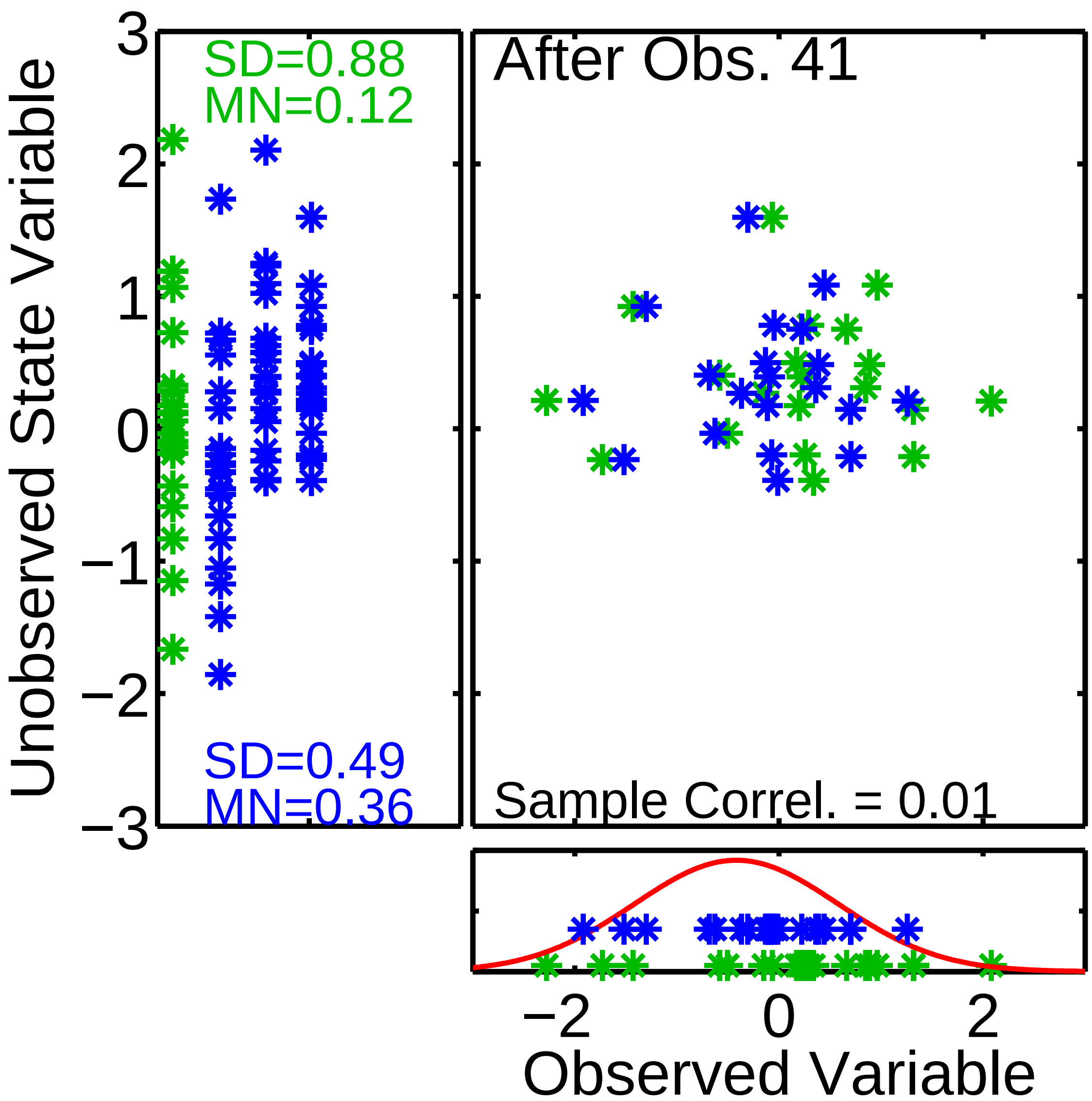


Suppose unobserved state variable is known to be unrelated to set of observed variables.

Unobserved variable should remain unchanged

Unobserved mean follows a random walk as more obs. are used.

Regression sampling error and filter divergence



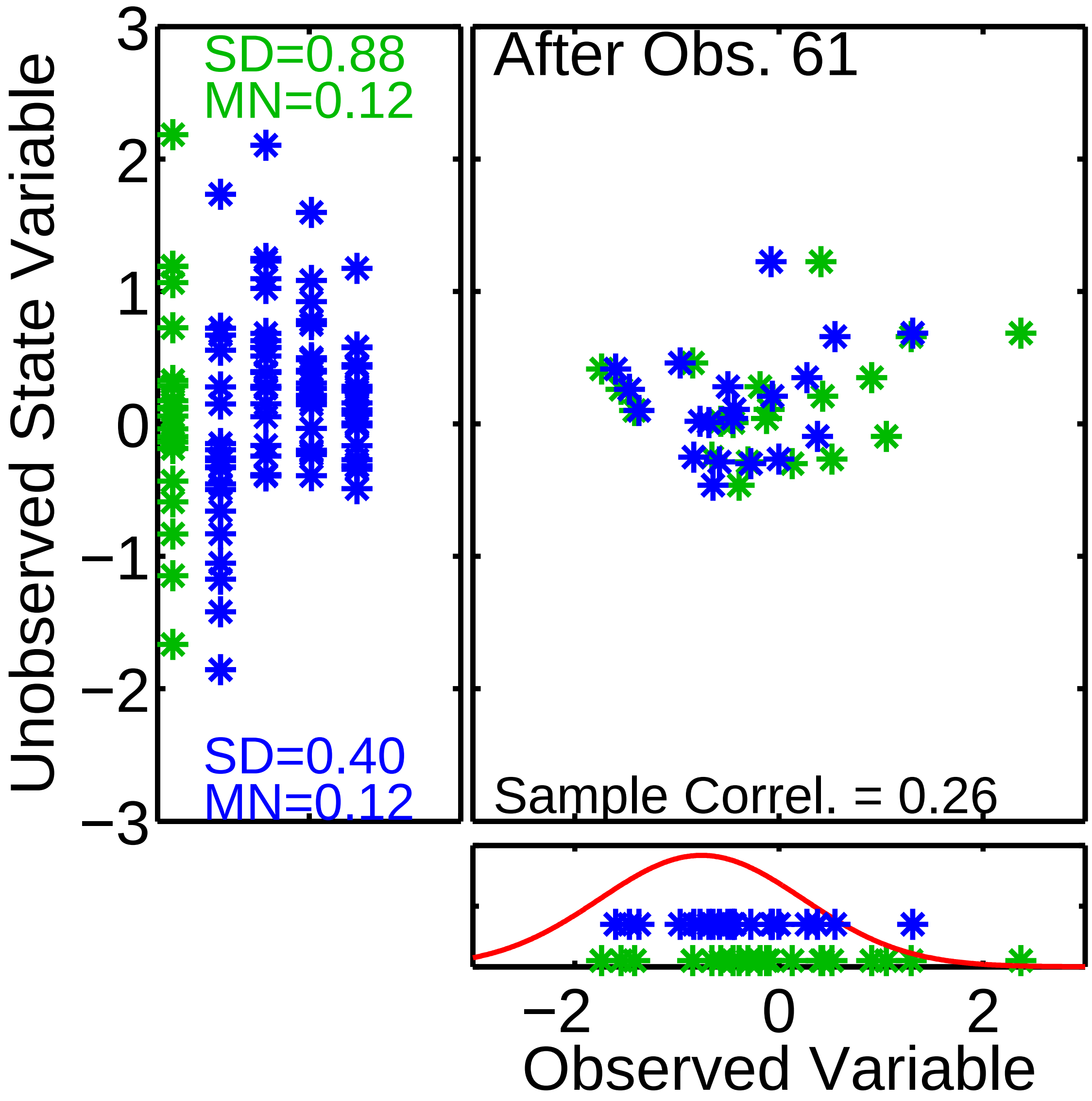
Suppose unobserved state variable is known to be unrelated to set of observed variables.

Unobserved variable should remain unchanged

Unobserved standard deviation is persistently decreased.

Expected change in |SD| is negative for any non-zero sample correlation!

Regression sampling error and filter divergence



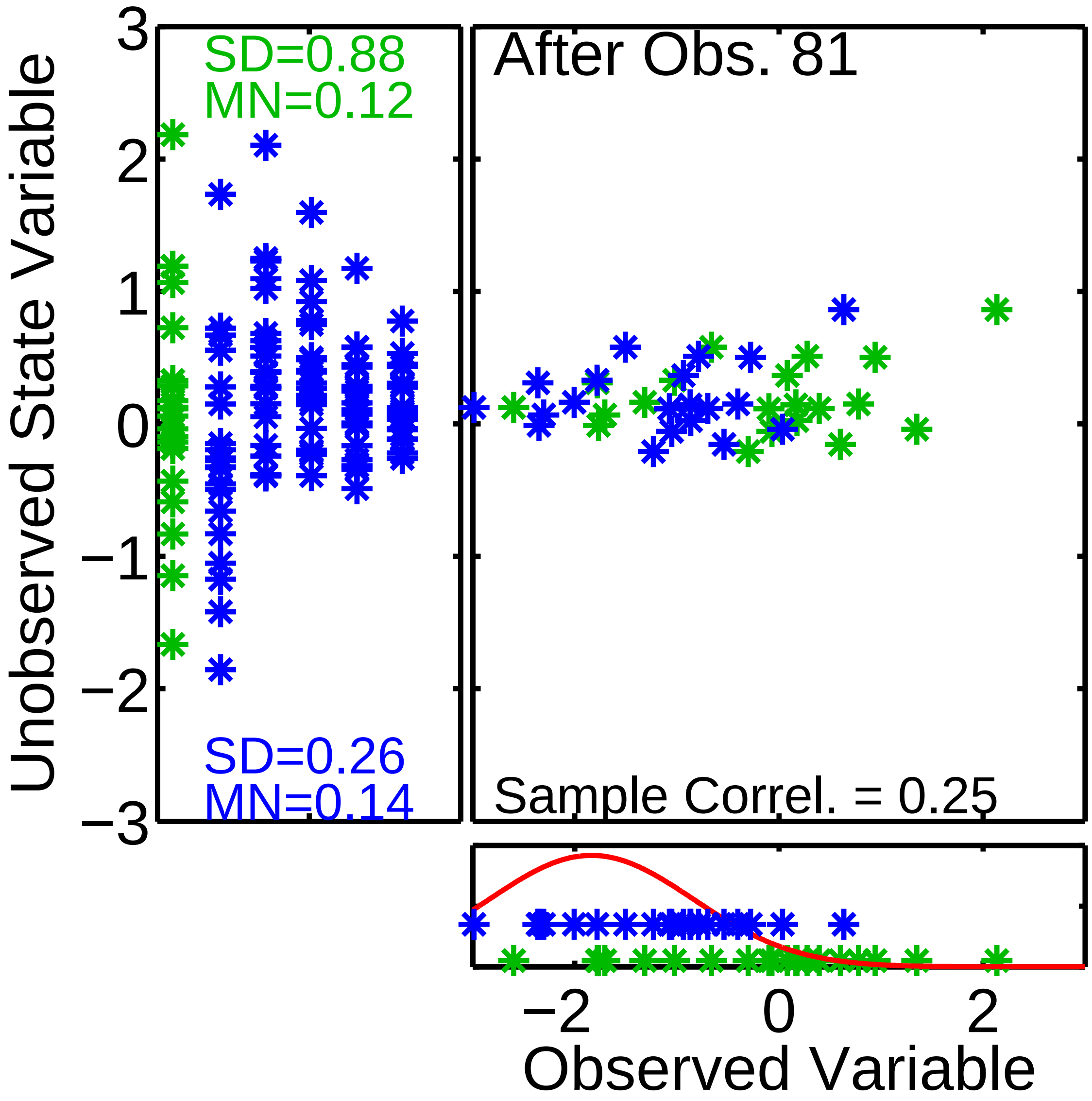
Suppose unobserved state variable is known to be unrelated to set of observed variables.

Unobserved variable should remain unchanged

Unobserved standard deviation is persistently decreased.

Expected change in |SD| is negative for any non-zero sample correlation!

Regression sampling error and filter divergence



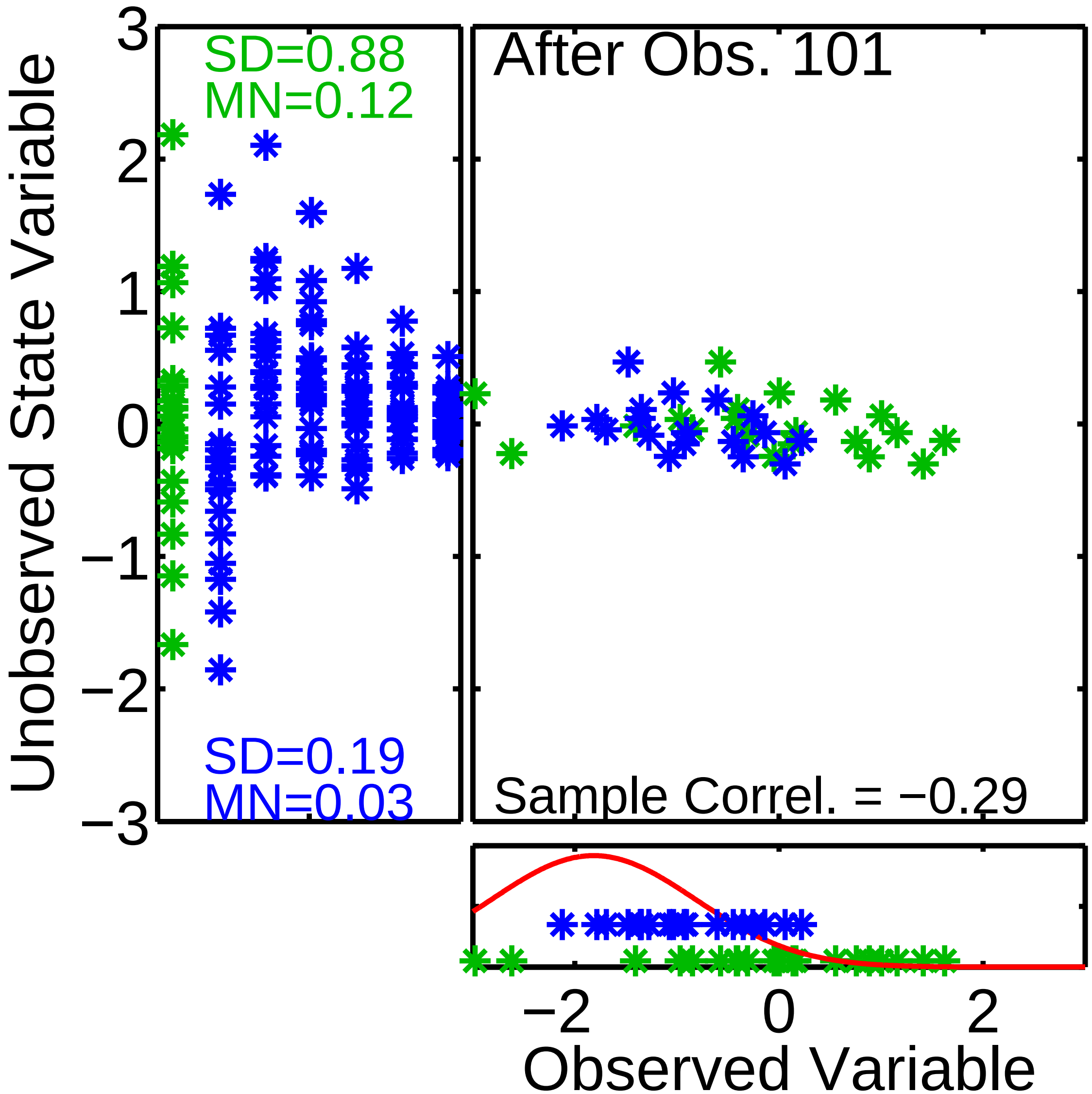
Suppose unobserved state variable is known to be unrelated to set of observed variables.

Unobserved variable should remain unchanged

Unobserved standard deviation is persistently decreased.

Expected change in |SD| is negative for any non-zero sample correlation!

Regression sampling error and filter divergence



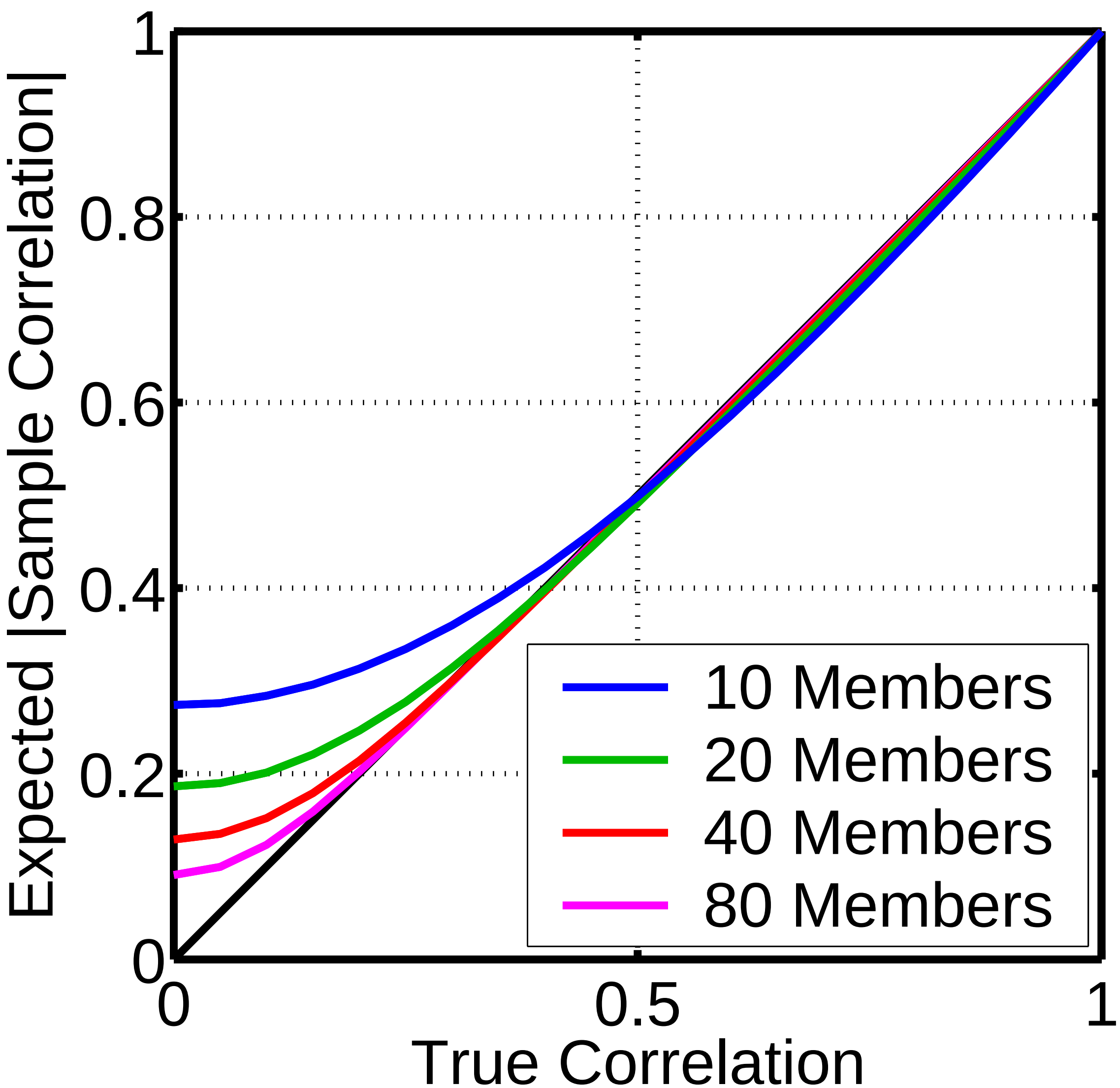
Suppose unobserved state variable is known to be unrelated to set of observed variables.

Estimates of unobs. become too confident

Give progressively less weight to any meaningful observations.

End result can be that meaningful obs. are essentially ignored.

Regression sampling error and filter divergence



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

Errors decrease with sample size and for large |real correlations|.

Ways to deal with regression sampling error:

1. Ignore it: if number of unrelated observations is small and there is some way of maintaining variance in priors.

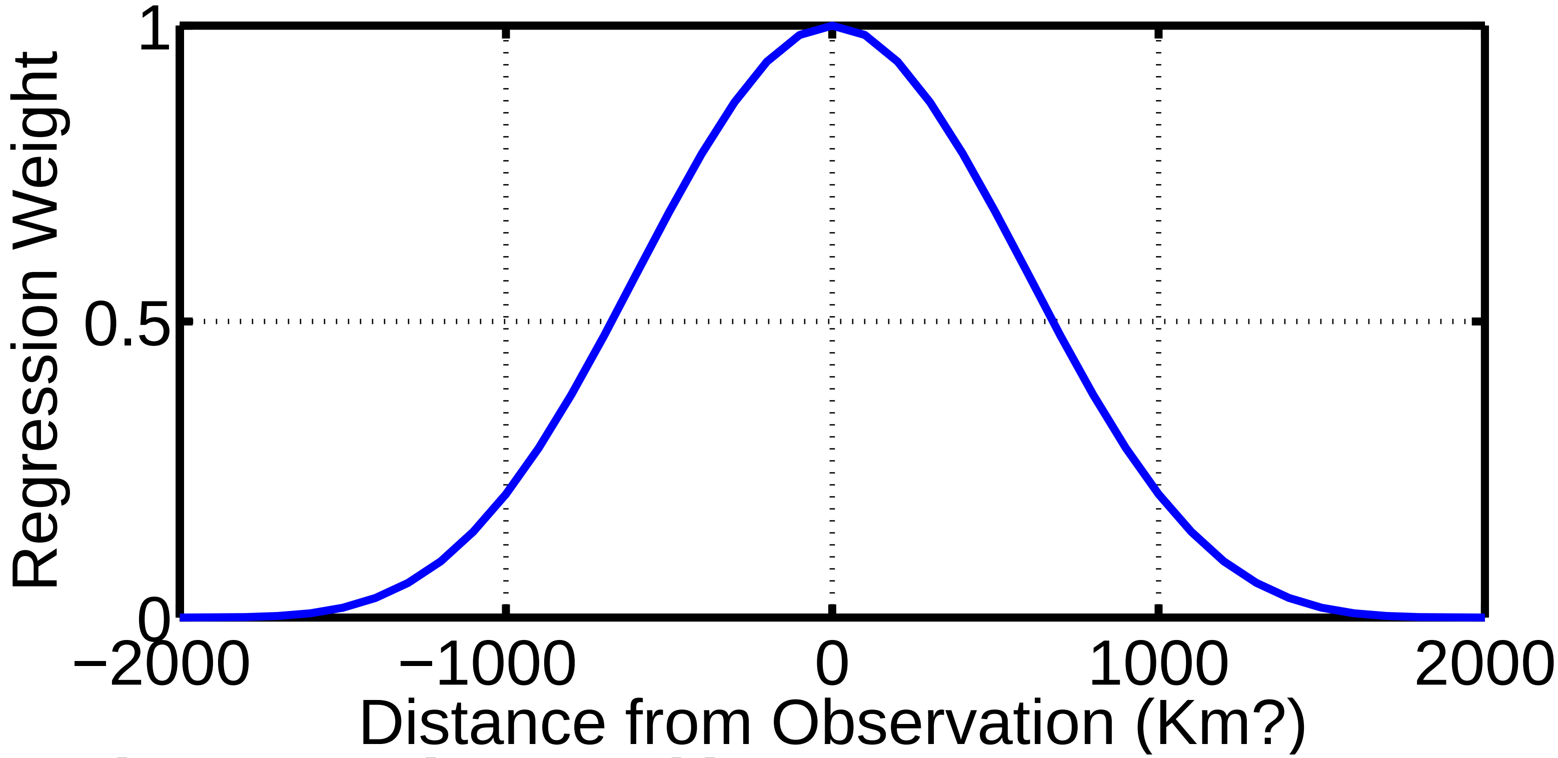
2. Use larger ensembles to limit sampling error.

3. Use additional a priori information about relation between observations and state variables.

4. Try to determine the amount of sampling error and correct for it.

Ways to deal with regression sampling error:

3. Use additional a priori information about relation between observations and state variables.



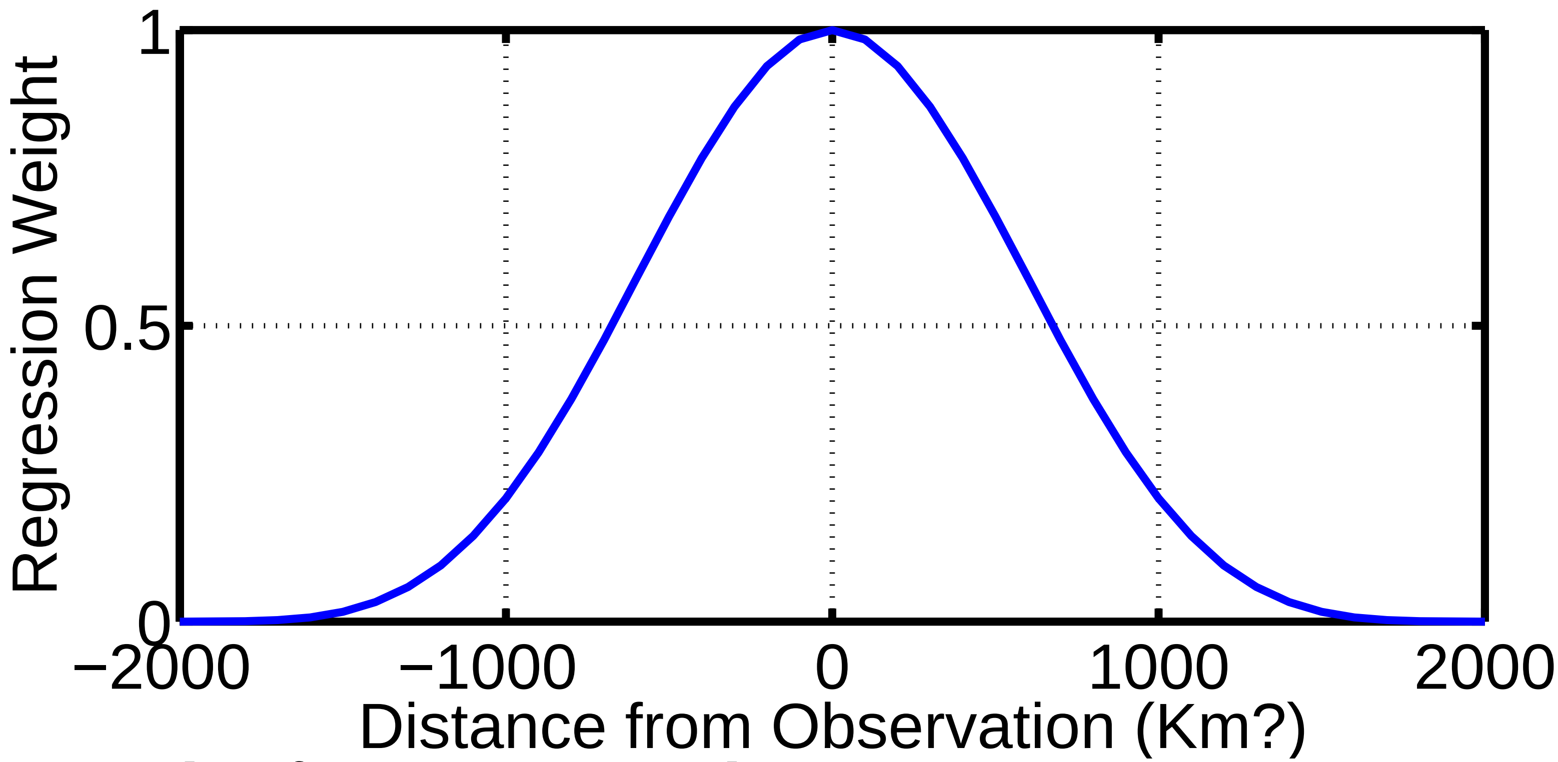
Atmospheric assimilation problems.

Weight regression as function of horizontal *distance* from observation.

Gaspari-Cohn: 5th order compactly supported polynomial.

Ways to deal with regression sampling error:

3. Use additional a priori information about relation between observations and state variables.



Can use other functions to weight regression.

Unclear what *distance* means for some obs./state variable pairs.

Referred to as LOCALIZATION.

Ways to deal with regression sampling error:

4. Try to determine the amount of sampling error and correct for it:

   A. Could weight regressions based on sample correlation.
      Limited success in tests.
      For small true correlations, can still get large sample correl.

   B. Do bootstrap with sample correlation to measure sampling error.
      Limited success.
      Repeatedly compute sample correlation with a sample removed.

   C. Use hierarchical Monte Carlo.
      Have a ‘sample’ of samples.
      Compute expected error in regression coefficients and weight.

Ways to deal with regression sampling error:

4C. Use hierarchical Monte Carlo: ensemble of ensembles.

Split ensemble into M independent groups.
For instance, 80 ensemble members becomes 4 groups of 20.

With M groups get M estimates of regression coefficient, β_i.

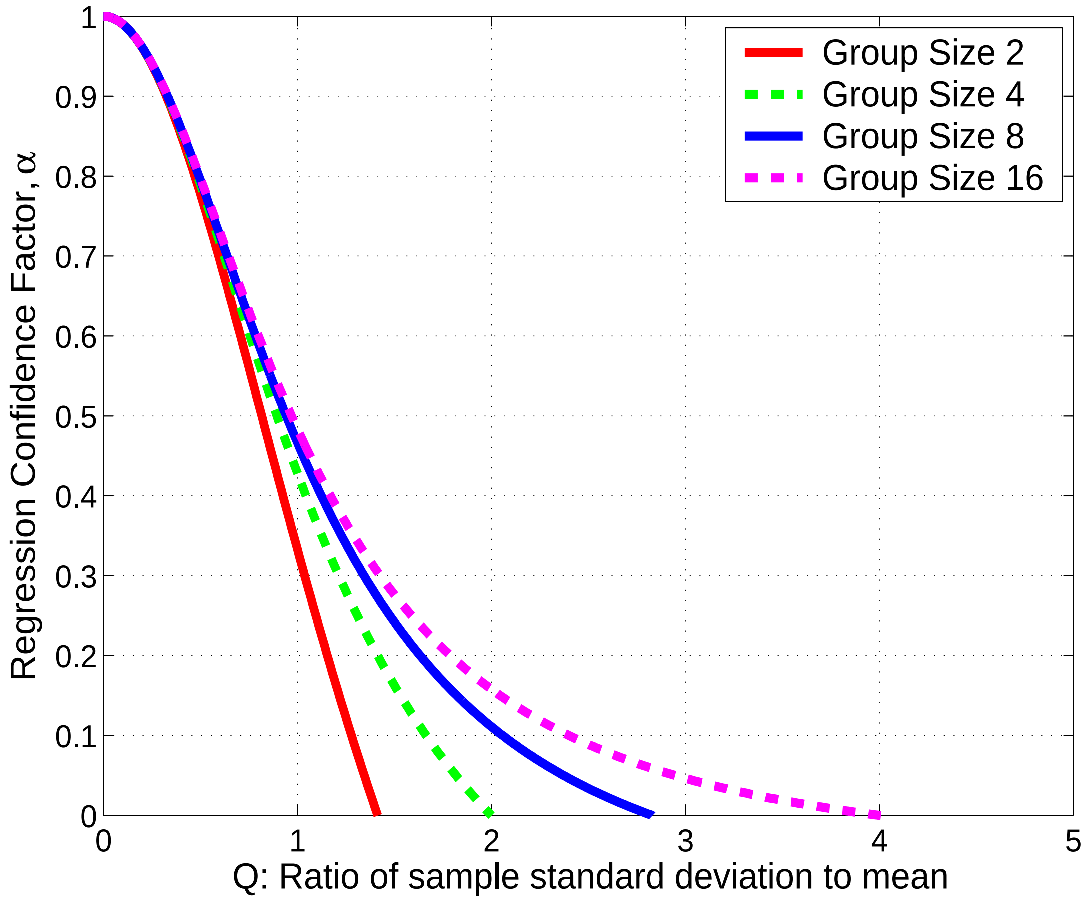
Find regression confidence factor α (weight) that minimizes:

$$\sqrt{\sum_{j=1}^{M} \sum_{i=1, i \neq j}^{M} [\alpha\beta_i - \beta_j]^2}$$

Minimizes RMS error in the regression (and state increments).

Ways to deal with regression sampling error:

4C. Use hierarchical Monte Carlo: ensemble of ensembles.



Weight regression by α.

If one has repeated observations, can generate sample mean or median statistics for α.

Mean α can be used in subsequent assimilations as a localization.

α is function of M and $Q = \Sigma_\beta / \bar{\beta}$ (sample SD /sample mean regression)

Phase 3: Generalize to geophysical models and observations

Dynamical system governed by (stochastic) Difference Equation:

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

Observations at discrete times:

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

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

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

Complete history of observations is:

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

Goal: Find probability distribution for state at time t:

$$p(x, t | Y_t) \tag{5}$$

Phase 3: Generalize to geophysical models and observations

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

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

Apply Bayes rule:

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

Noise is white in time (3) so:

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

Integrate numerator to get normalizing denominator:

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

Phase 3: Generalize to geophysical models and observations

Probability after new observation:

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

Exactly analogous to earlier derivation except that x and y are vectors.

EXCEPT, no guarantee we have prior sample for each observation.

SO, let’s make sure we have priors by ‘extending’ state vector.

Phase 3: Generalize to geophysical models and observations

Extending the state vector to joint state-observation vector.

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

Applying h to x at a given time gives expected values of observations.

Get prior sample of obs. by applying h to each sample of state vector x.

Let z = [x, y] be the combined vector of state and observations.

Phase 3: Generalize to geophysical models and observations

NOW, we have a prior for each observation:

$$p\left(z, t_k \middle| Y_{t_k}\right) = \frac{p(y_k|z)p(z, t_k|Y_{t_{k-1}})}{\int p(y_k|\xi)p(\xi, t_k|Y_{t_{k-1}})d\xi} \qquad (10.\text{ext})$$

Phase 3: Generalize to geophysical models and observations

One more issue: how to deal with many observations in set y_k?

Let y_k be composed of s subsets of observations: $y_k = \{y_k^1, y_k^2, \ldots, y_k^s\}$

Observational errors for obs. in set i independent of those in set j.

Then: $p(y_k|z) = \prod_{i=1}^{s} p(y_k^i|z)$

Can rewrite (10.ext) as series of products and normalizations.

Phase 3: Generalize to geophysical models and observations

One more issue: how to deal with many observations in set y_k?

Implication: can assimilate observation subsets sequentially.

If subsets are scalar (individual obs. have mutually independent error distributions), can assimilate each observation sequentially.

If not, have two options:

1. Repeat everything above with matrix algebra.

2. Do singular value decomposition; diagonalize obs. error covariance.
   Assimilate observations sequentially in rotated space.
   Rotate result back to original space.

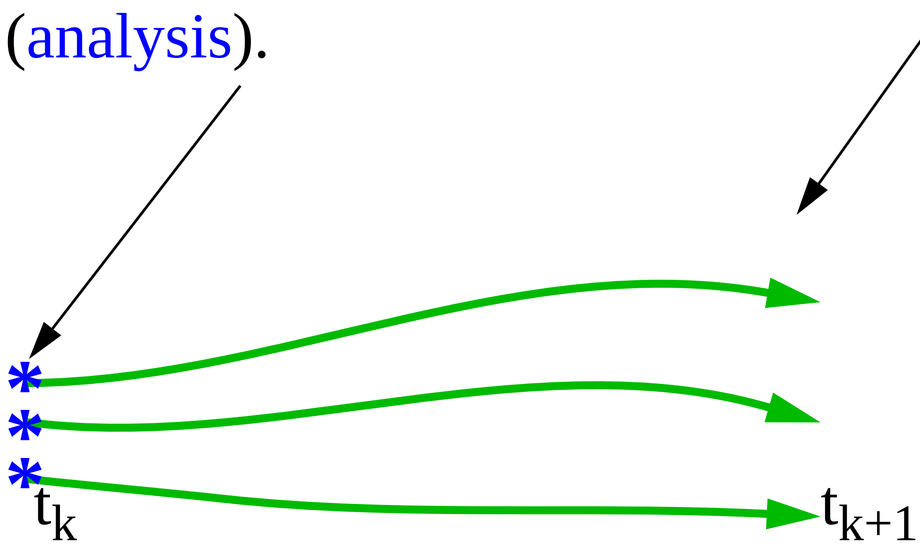
Good news: Most geophysical obs. have independent errors!

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.

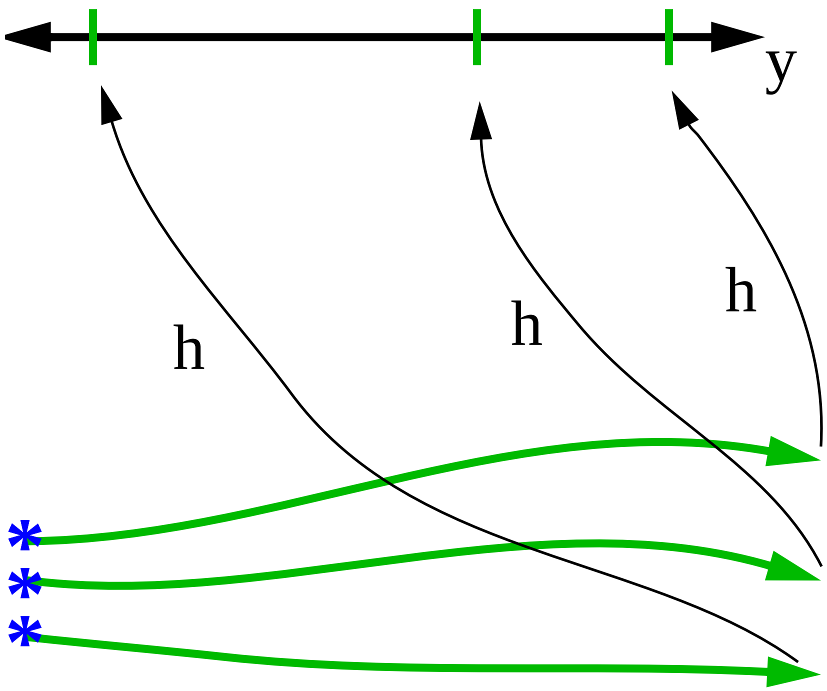
Ensemble state estimate after using previous observation (analysis).

Ensemble state at time of next observation (prior).



How an Ensemble Filter Works for Geophysical Data Assimilation

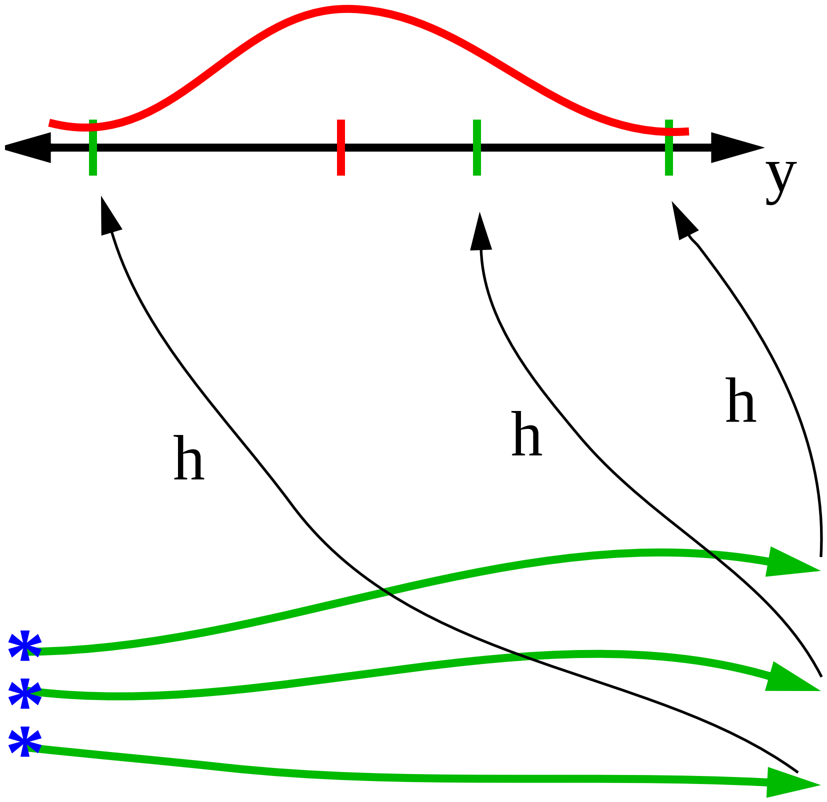
2. Get prior ensemble sample of observation, y=h(x), by applying forward operator h to each ensemble member.



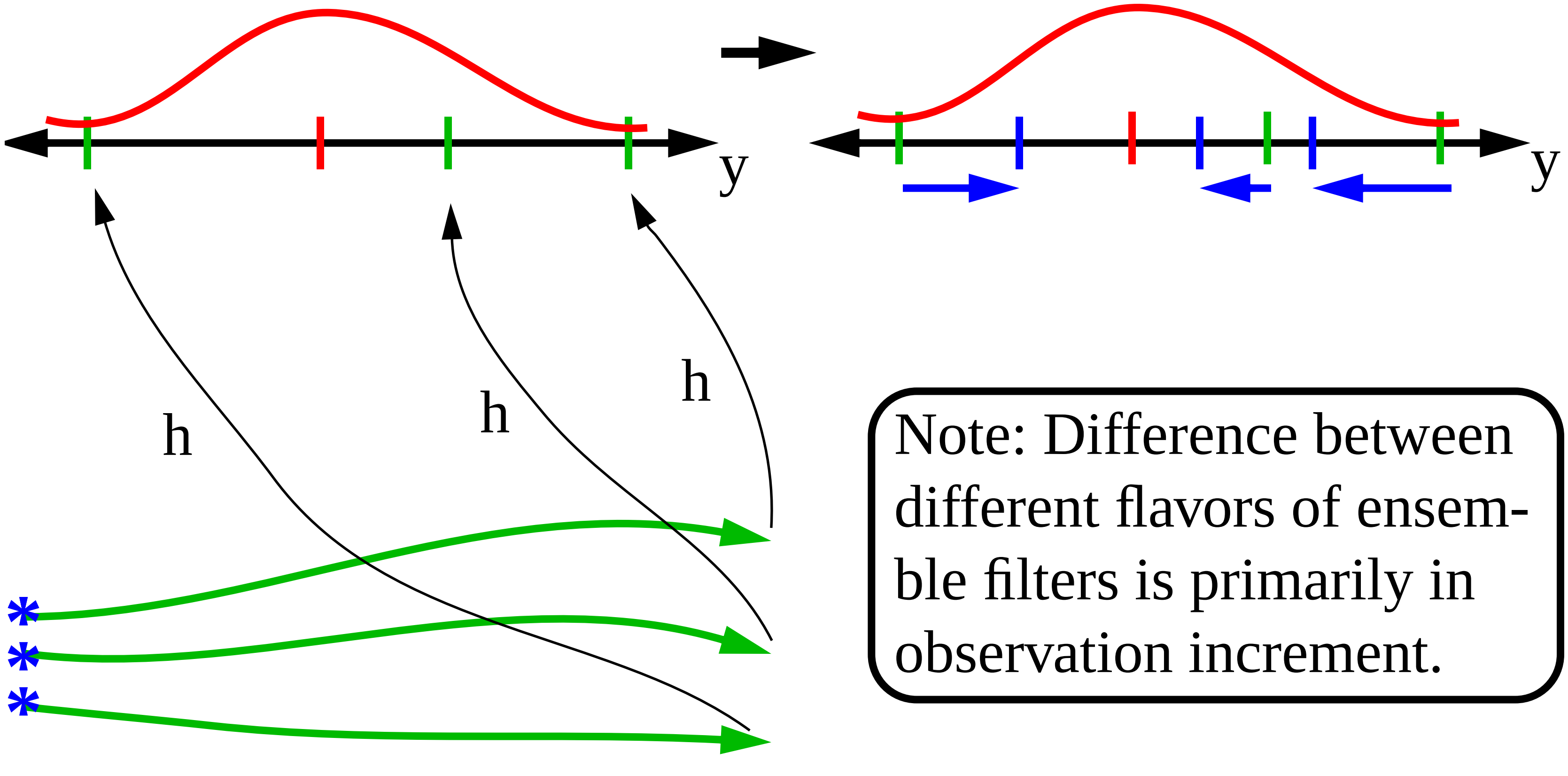
Theory: observations from instruments with uncorrelated errors can be done sequentially.

How an Ensemble Filter Works for Geophysical Data Assimilation

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



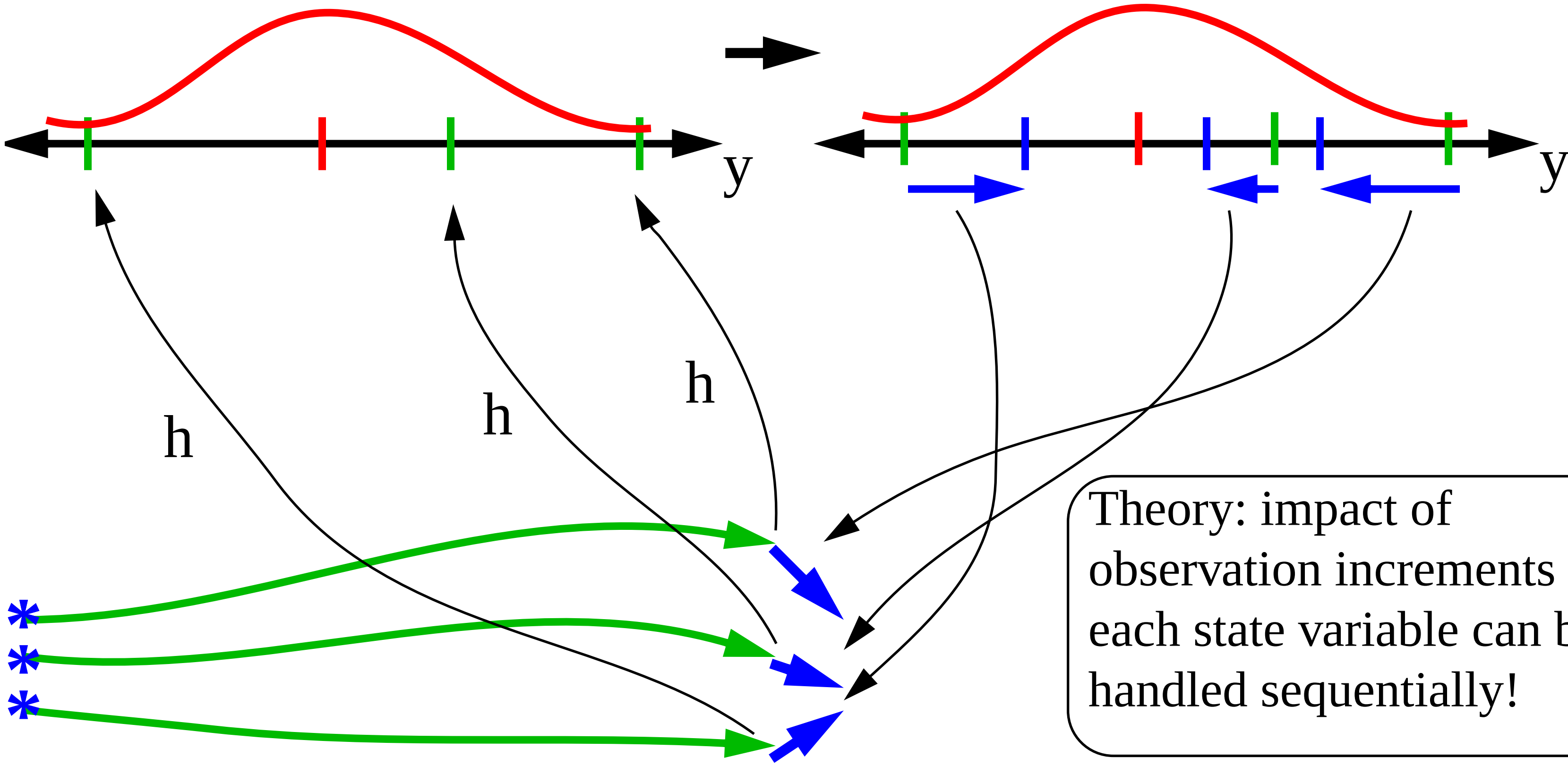
4. Find increment for each prior observation ensemble
(this is a scalar problem for uncorrelated observation errors).



Note: Difference between different flavors of ensemble filters is primarily in observation increment.

How an Ensemble Filter Works for Geophysical Data Assimilation

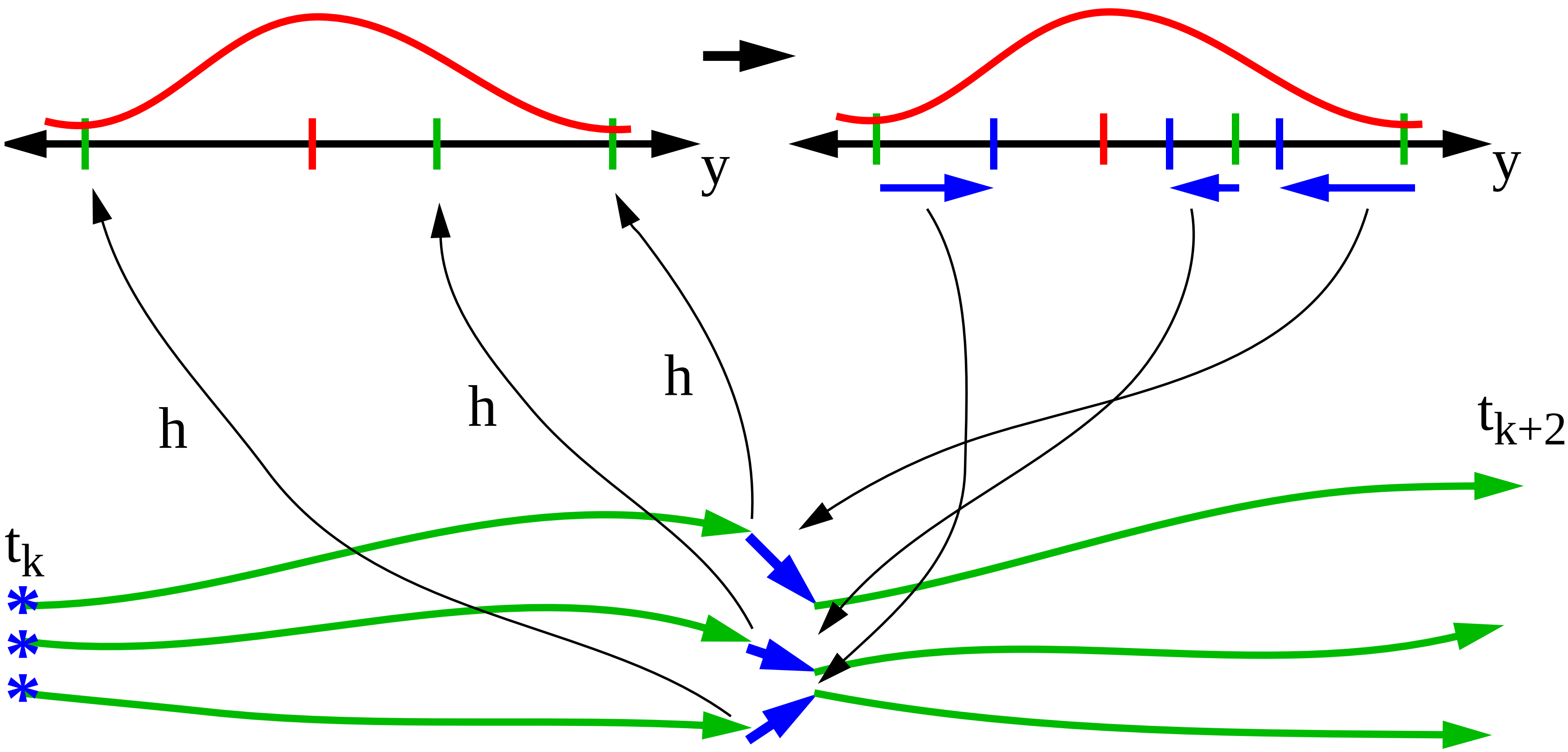
5. Use ensemble samples of y and each state variable to linearly regress observation increments onto state variable increments.



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

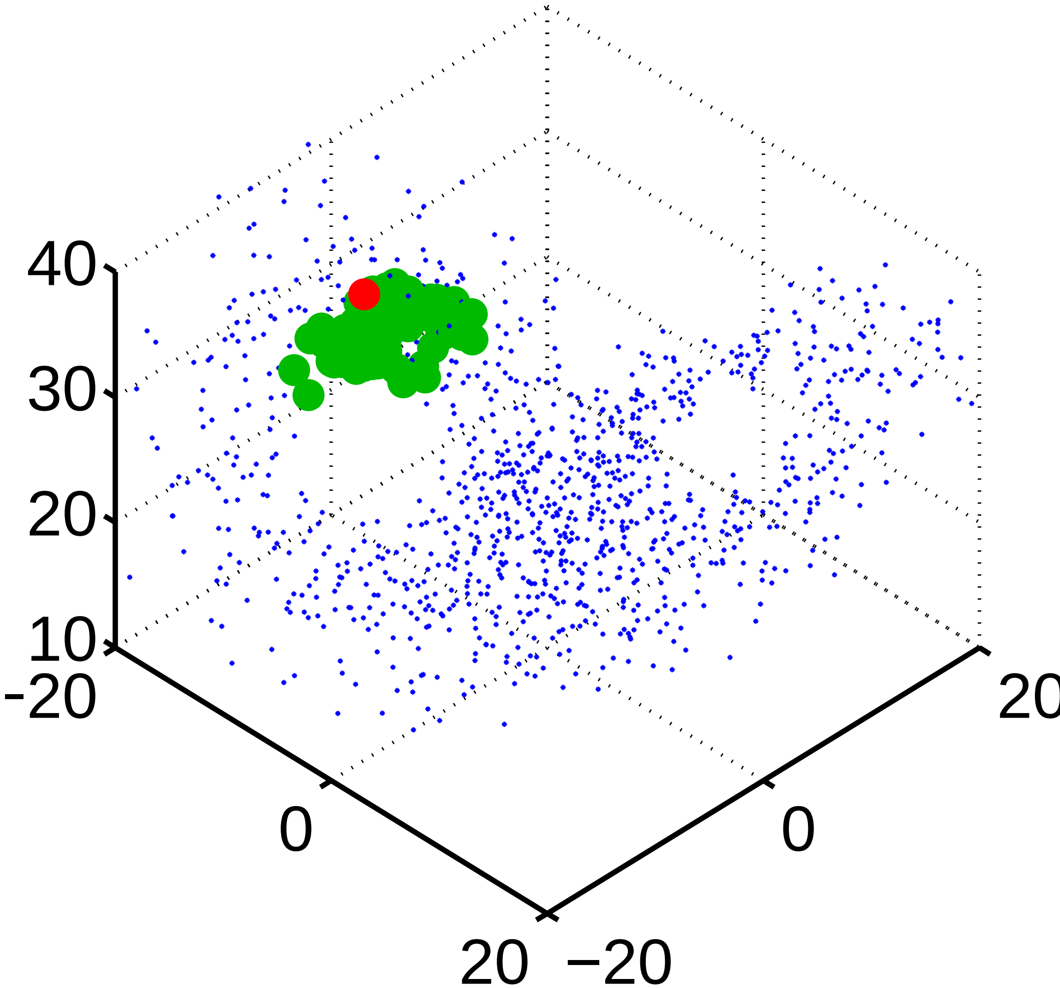
How an Ensemble Filter Works for Geophysical Data Assimilation

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



Phase 3: Generalize to geophysical models and observations

Simple example: Lorenz-63 3-variable chaotic model.



Observation in red.

Prior ensemble in green.

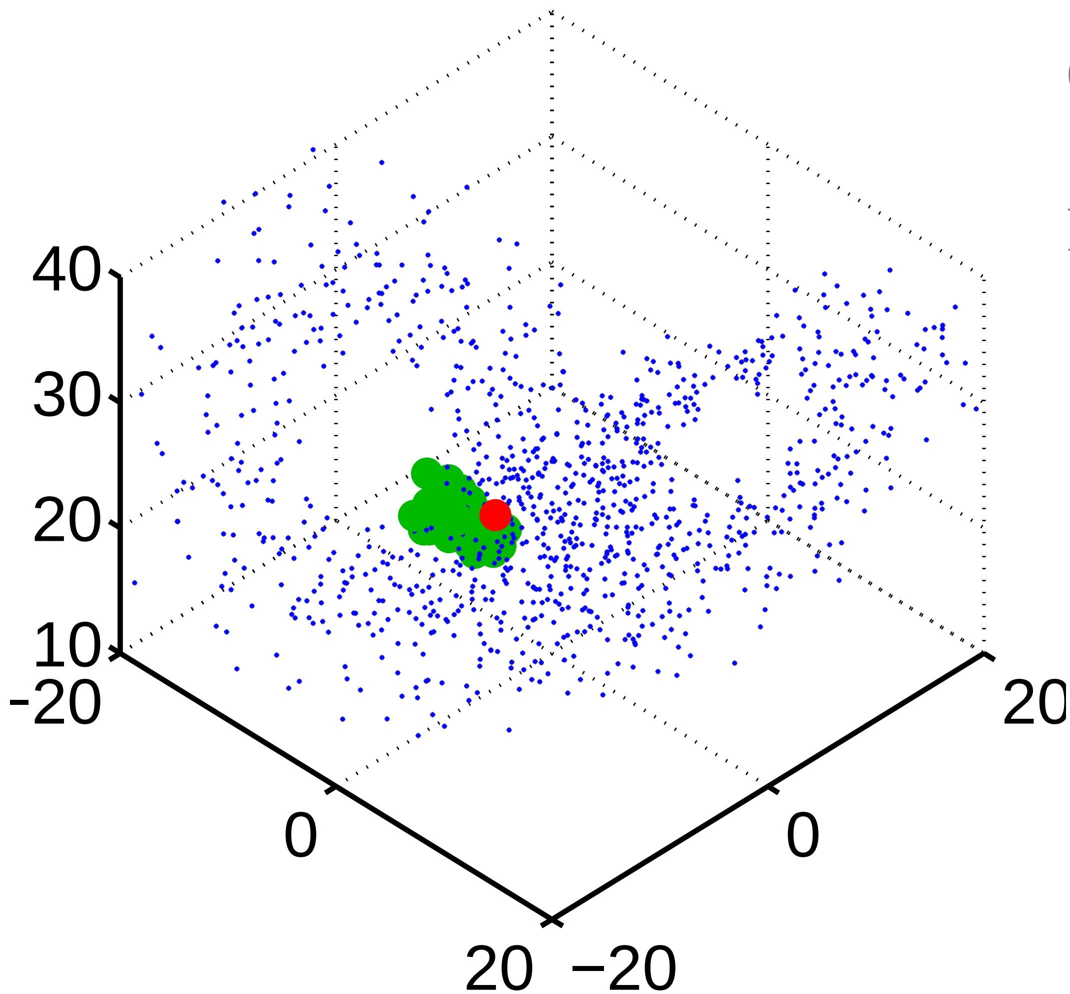
Observing all three state variables.

Obs. error variance = 4.0.

4 20-member ensembles.

Phase 3: Generalize to geophysical models and observations

Simple example: Lorenz-63 3-variable chaotic model.

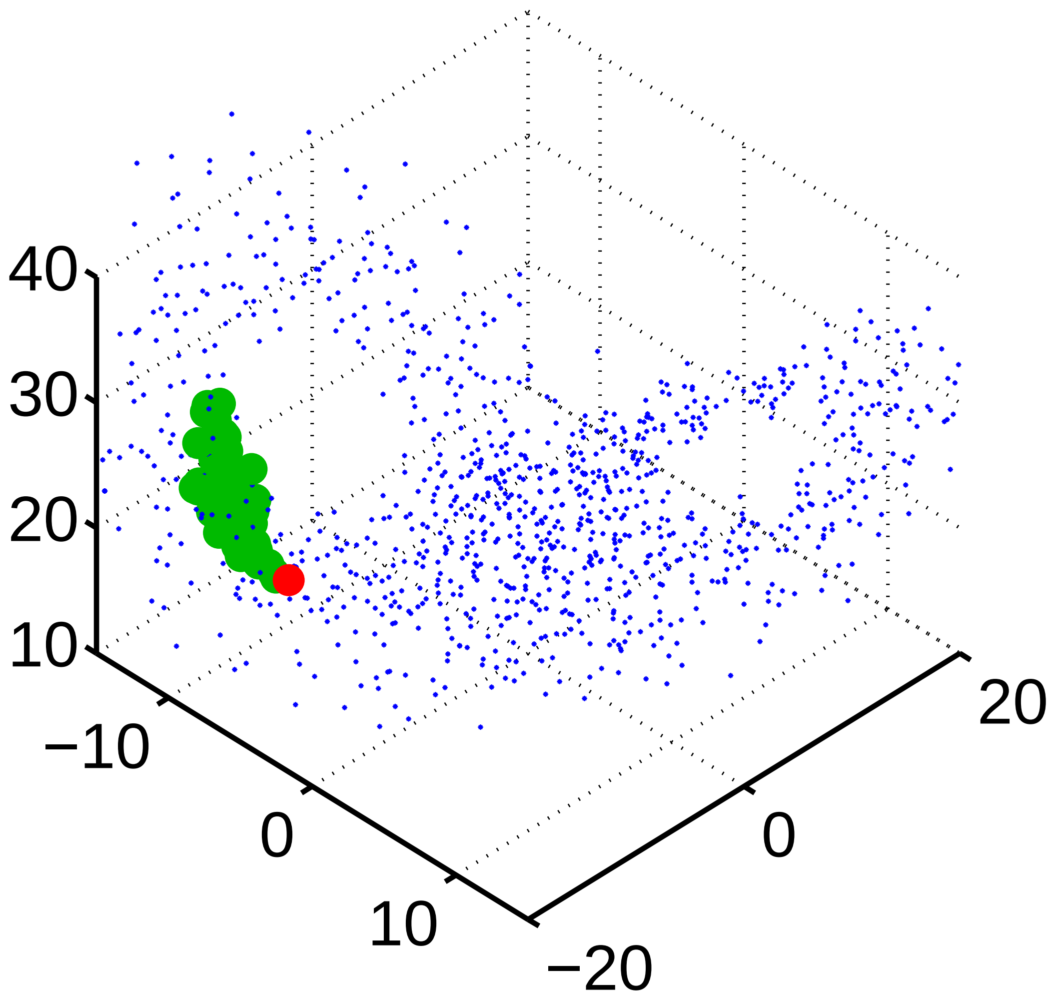


Observation in red.

Prior ensemble in green.

Phase 3: Generalize to geophysical models and observations

Simple example: Lorenz-63 3-variable chaotic model.

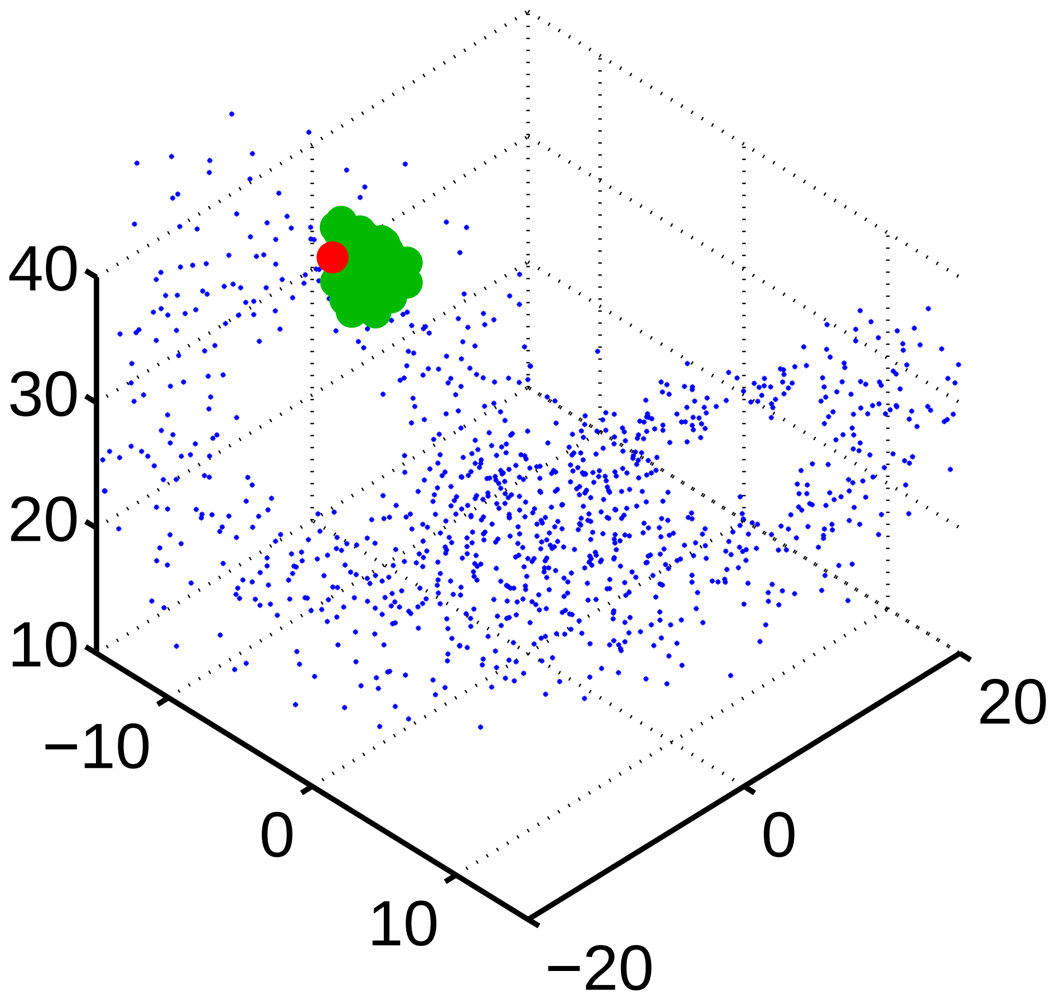


Observation in red.

Prior ensemble in green.

Phase 3: Generalize to geophysical models and observations

Simple example: Lorenz-63 3-variable chaotic model.

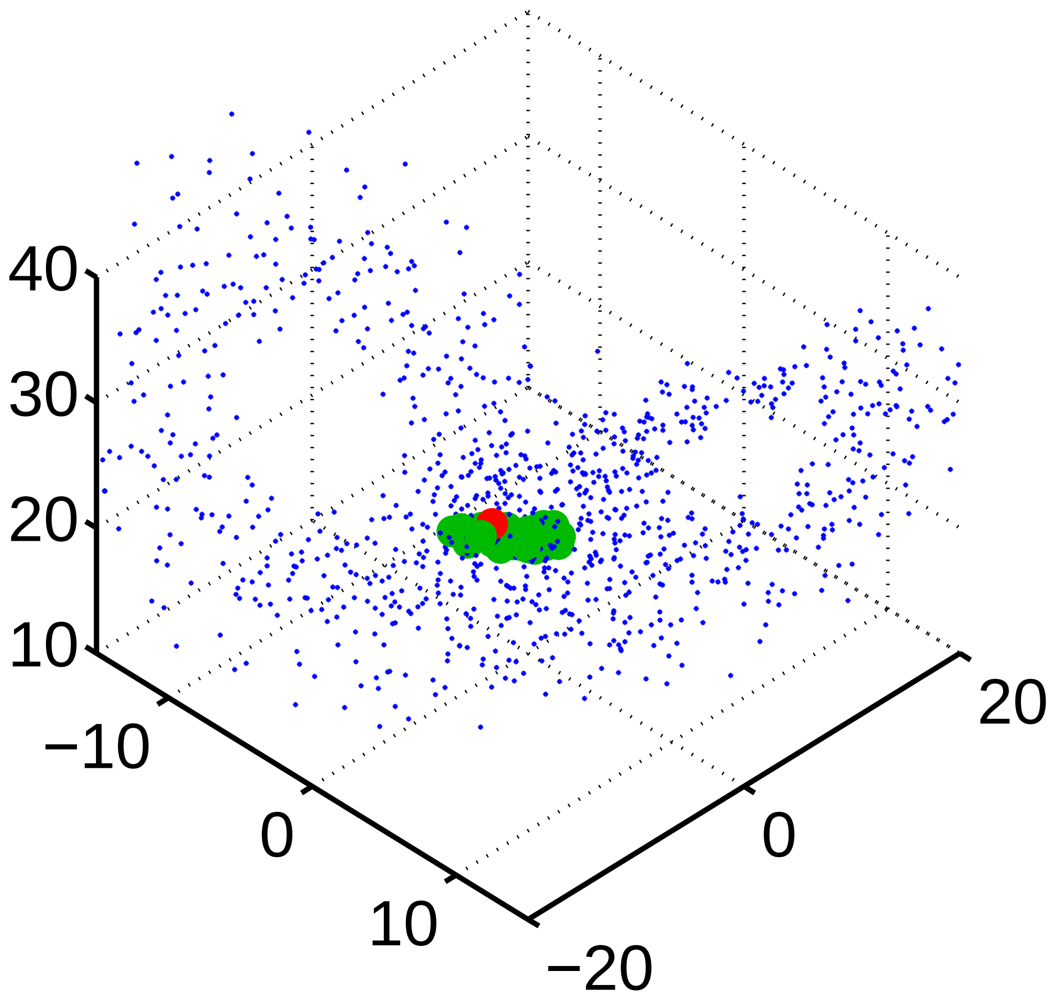


Observation in red.

Prior ensemble in green.

Phase 3: Generalize to geophysical models and observations

Simple example: Lorenz-63 3-variable chaotic model.



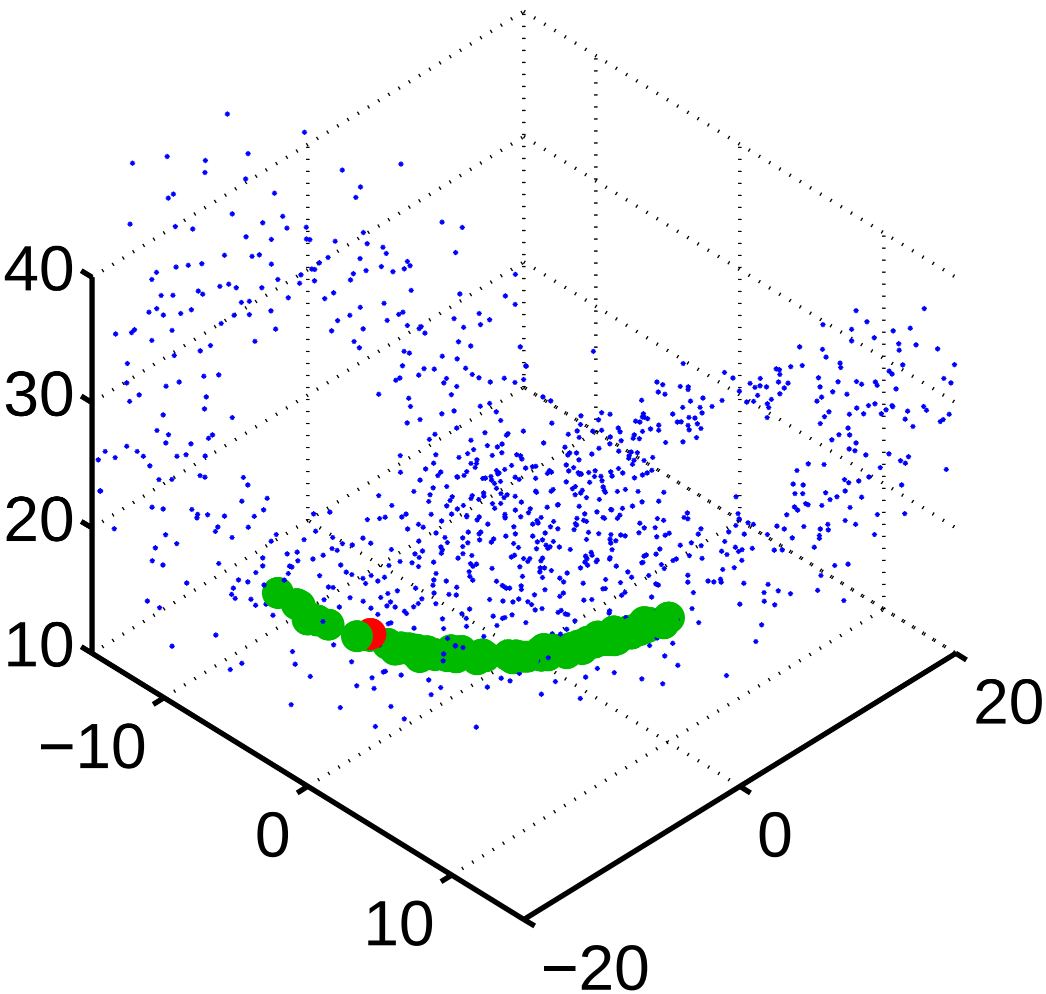
Observation in red.

Prior ensemble in green.

Ensemble is passing through unpredictable region.

Phase 3: Generalize to geophysical models and observations

Simple example: Lorenz-63 3-variable chaotic model.



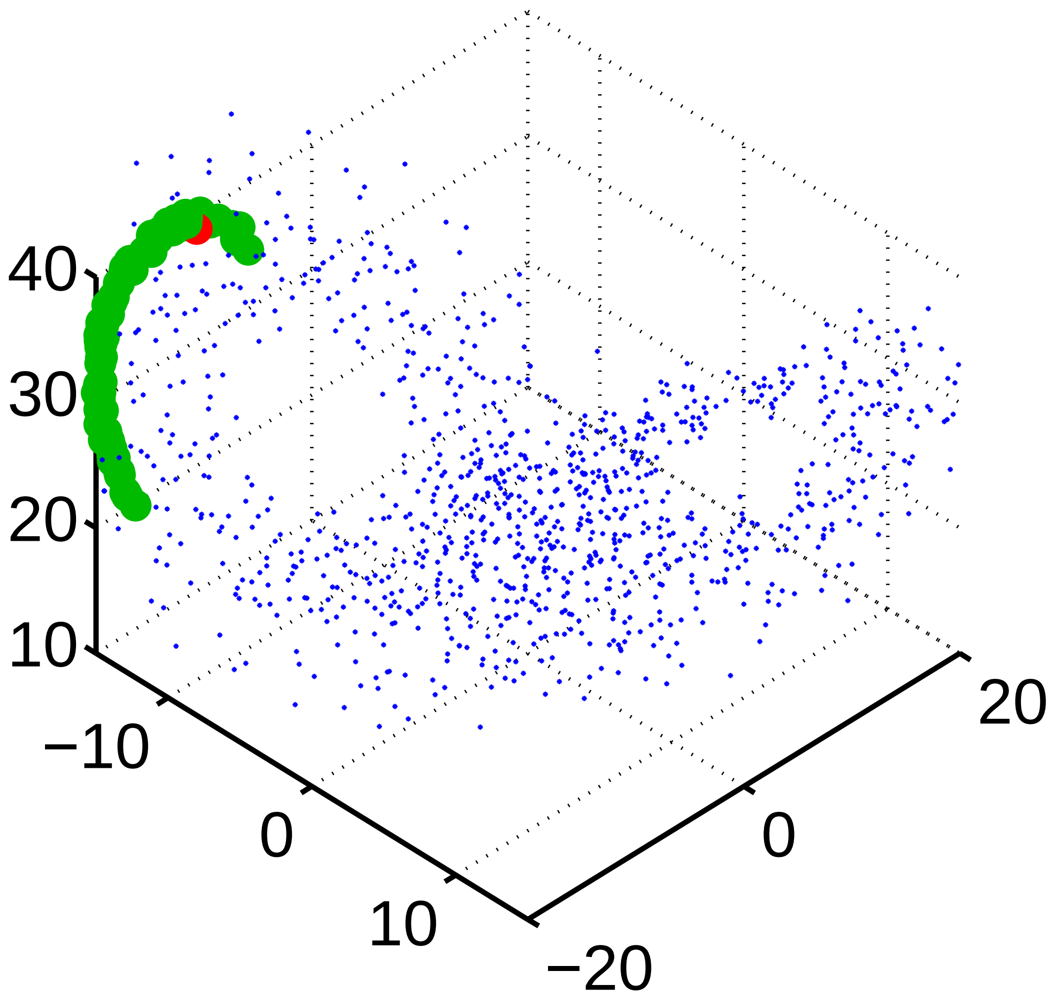
Observation in red.

Prior ensemble in green.

Part of ensemble heads for one lobe, the rest for the other.

Phase 3: Generalize to geophysical models and observations

Simple example: Lorenz-63 3-variable chaotic model.



Observation in red.

Prior ensemble in green.

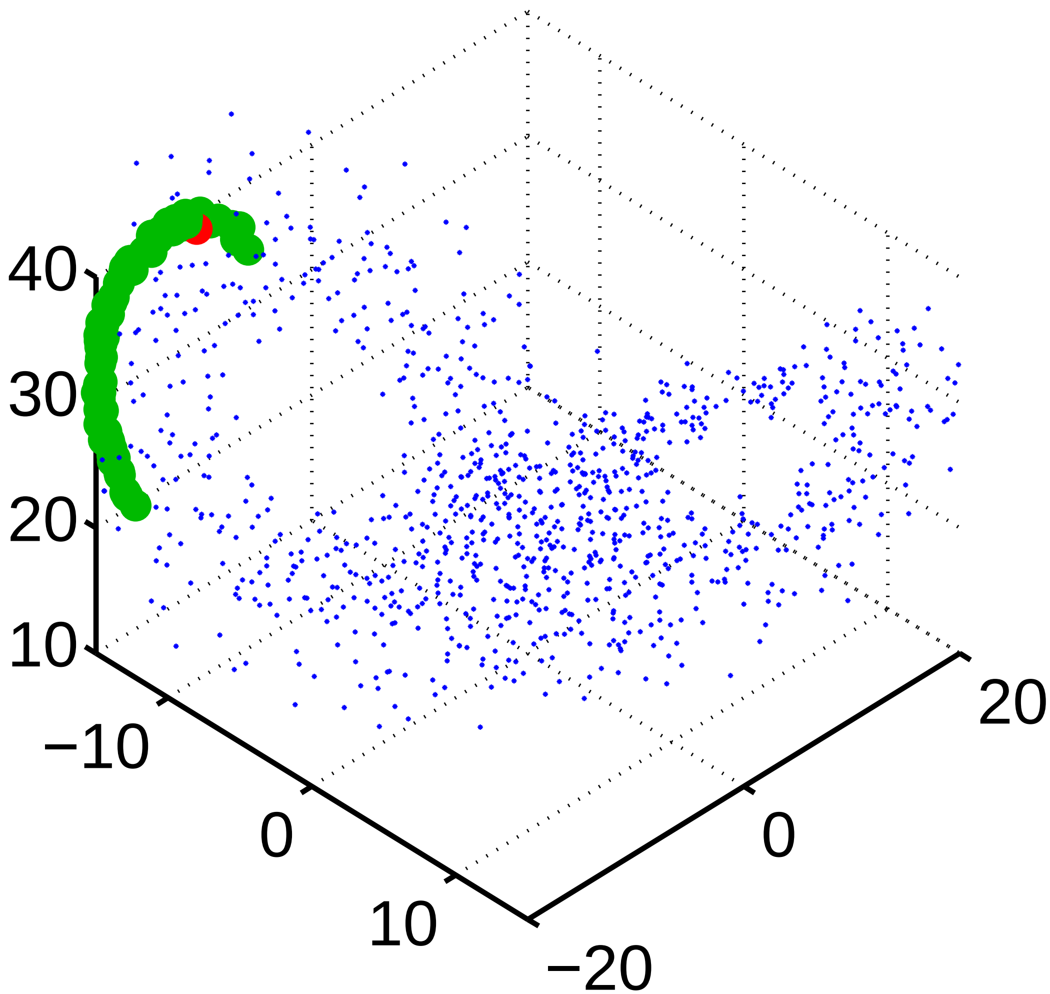
The prior is not linear here.

Standard regression might be pretty bad.

Covariance inflation might also be bad, pushing ensemble off the attractor.

Phase 3: Generalize to geophysical models and observations

Simple example: Lorenz-63 3-variable chaotic model.



Observation in red.

Prior ensemble in green.

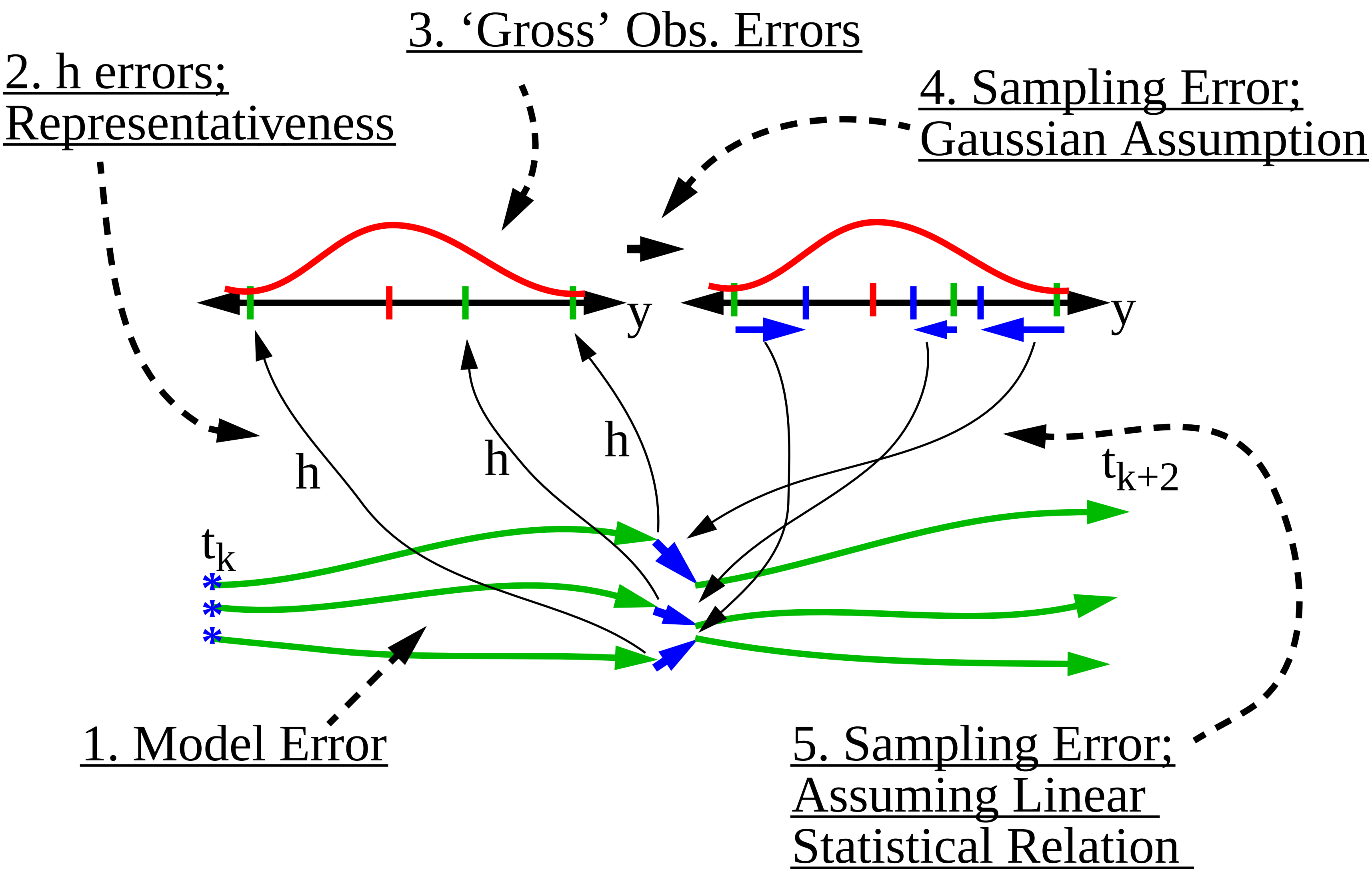
The prior is not linear here.

On the other hand...

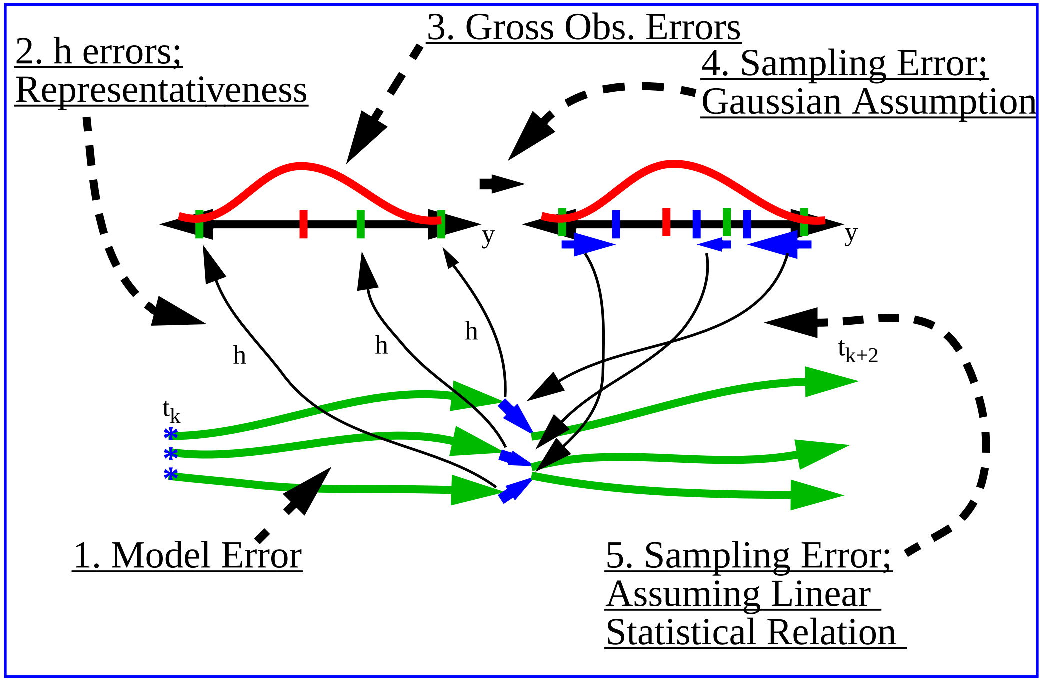
Hard to contrive examples this bad.

Behavior like this not apparent in real assimilations.

Some Error Sources in Ensemble Filters



Dealing With Ensemble Filter Errors



Fix 1, 2, 3 independently HARD but ongoing.

Often, ensemble filters...

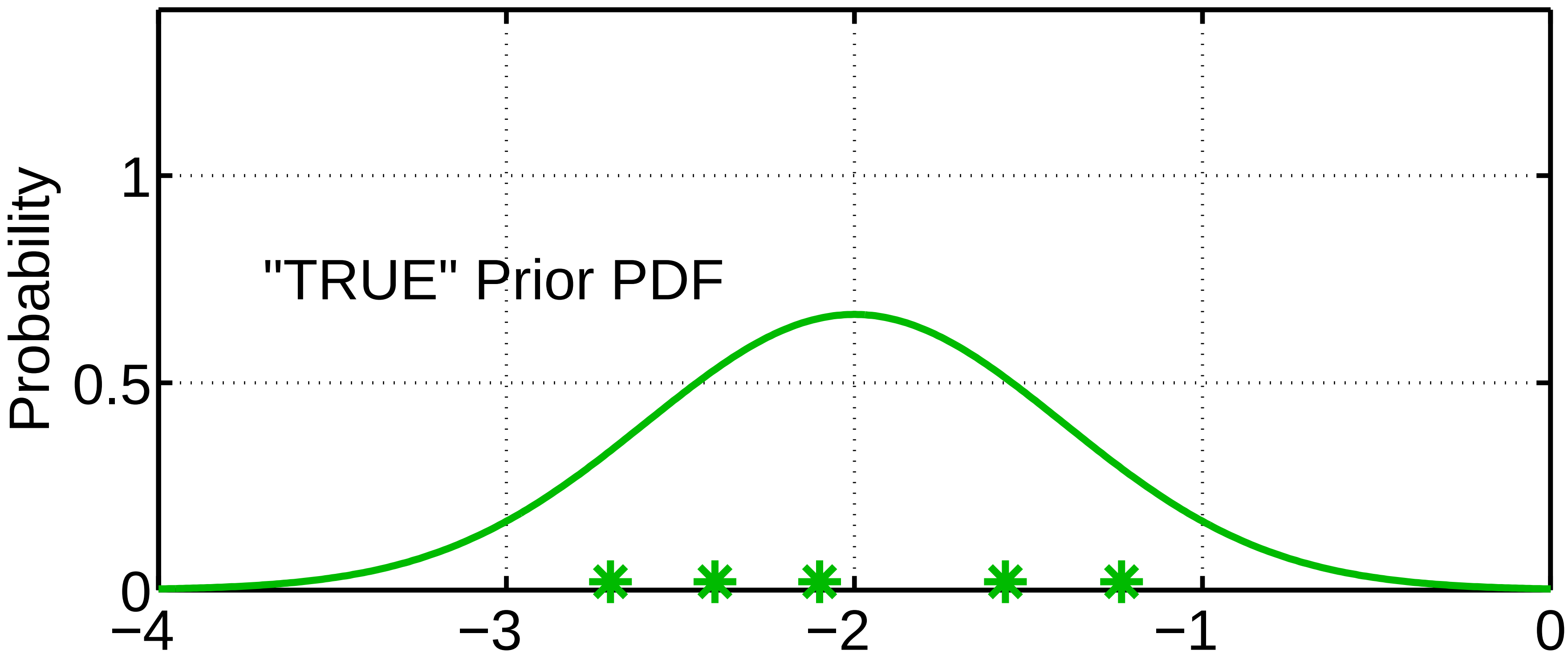
1-4: Covariance inflation, Increase prior uncertainty to give obs more impact.

5. ‘Localization’: only let obs. impact a set of ‘nearby’ state variables.

Often smoothly decrease impact to 0 as function of distance.

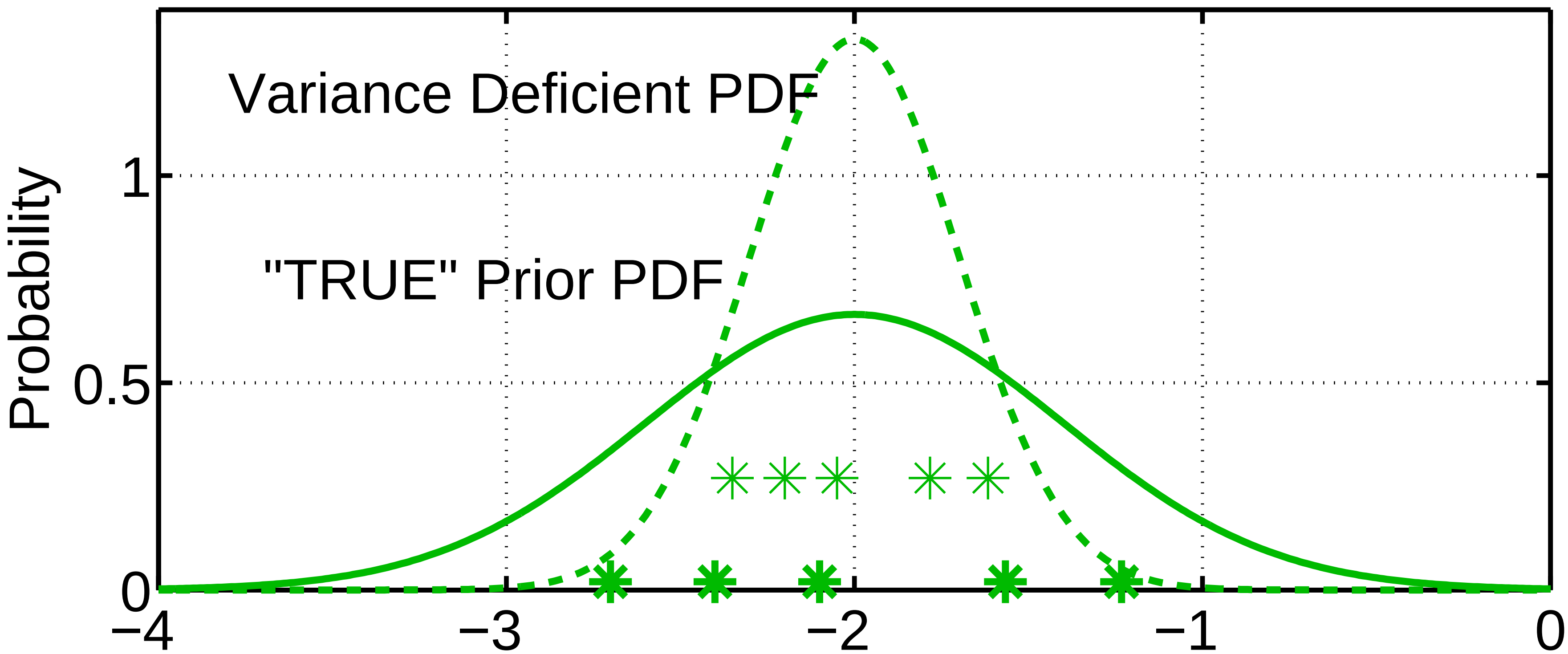
Model/Filter Error; Filter Divergence and Variance Inflation

1. History of observations and physical system => ‘true’ distribution.



Model/Filter Error; Filter Divergence and Variance Inflation

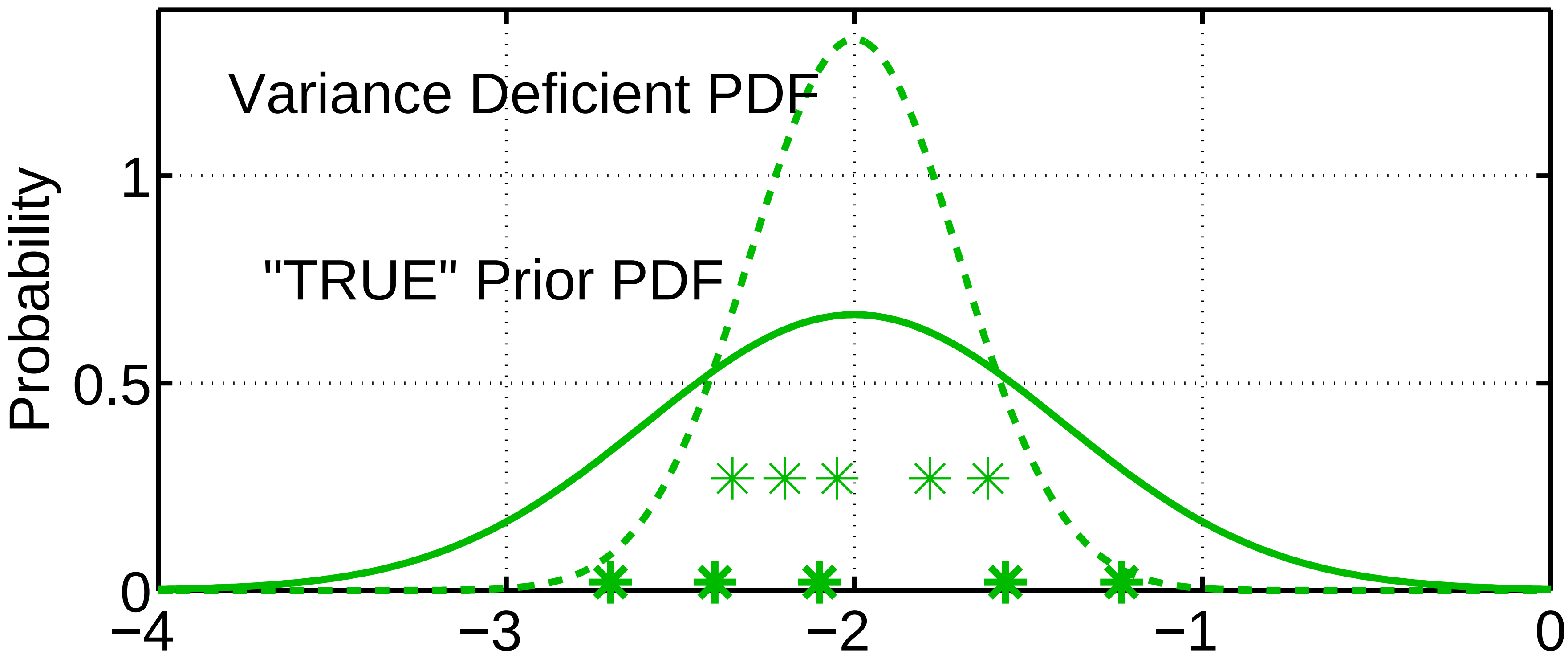
1. History of observations and physical system => ‘true’ distribution.
2. Sampling error, some model errors lead to insufficient prior variance.



3. Can lead to ‘filter divergence’: prior is too confident, obs. ignored

Model/Filter Error; Filter Divergence and Variance Inflation

1. History of observations and physical system => ‘true’ distribution.
2. Sampling error, some model errors lead to insufficient prior variance.

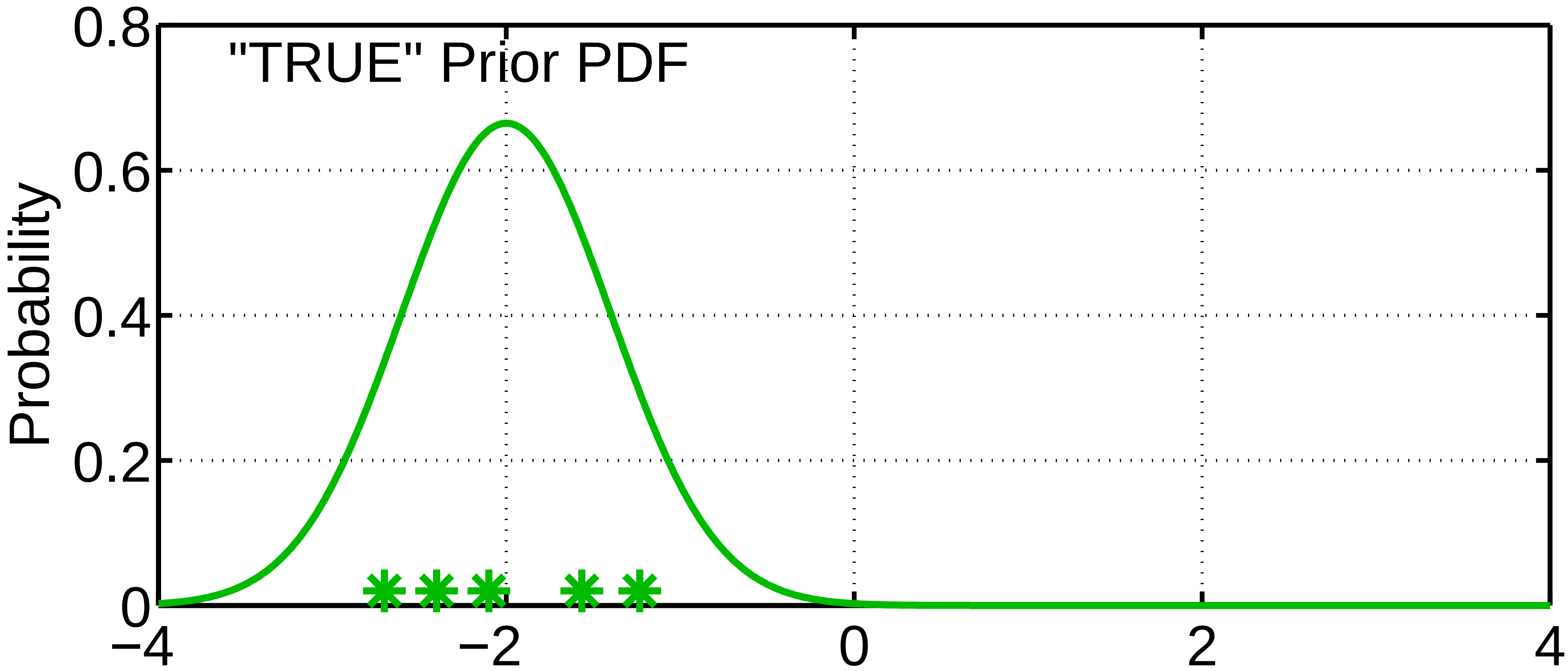


3. Naive solution is Variance inflation: just increase spread of prior
4. For ensemble member i, $inflate(x_i) = \sqrt{\lambda}(x_i - \bar{x}) + \bar{x}$.

Model/Filter Error; Filter Divergence and Variance Inflation

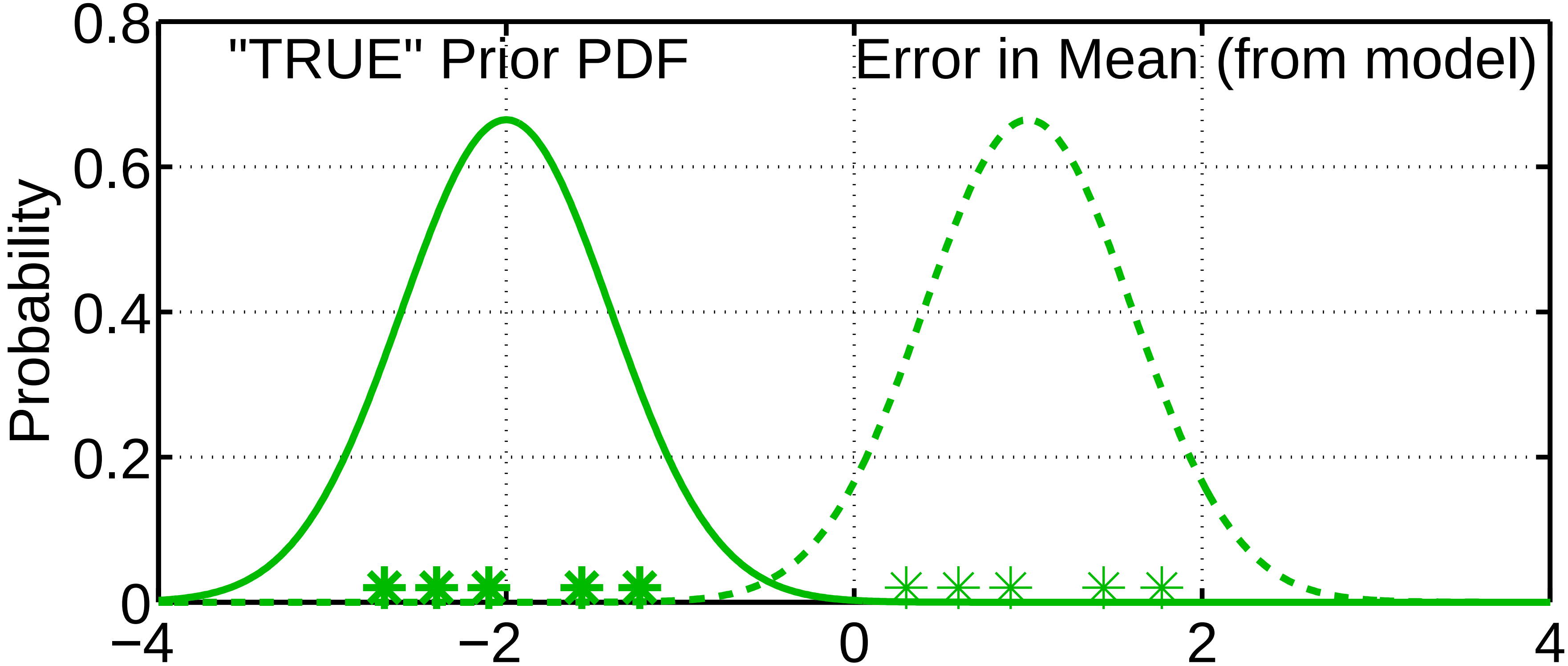
1. History of observations and physical system => ‘true’ distribution.

.



Model/Filter Error; Filter Divergence and Variance Inflation

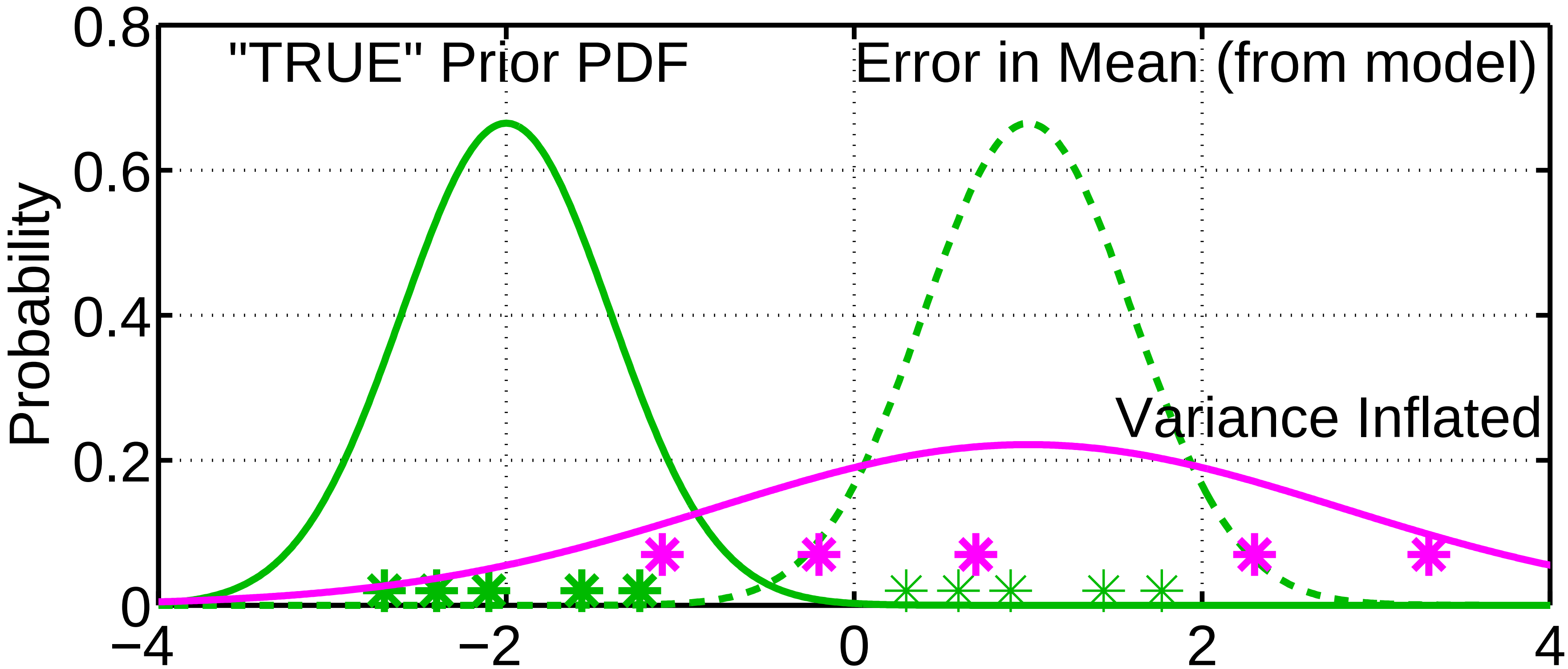
1. History of observations and physical system => ‘true’ distribution.
2. Most model errors also lead to erroneous shift in entire distribution.



3. Again, prior can be viewed as being TOO CERTAIN

Model/Filter Error; Filter Divergence and Variance Inflation

1. History of observations and physical system => ‘true’ distribution.
2. Most model errors also lead to erroneous shift in entire distribution.



3. Again, prior can be viewed as being TOO CERTAIN
4. Inflating can ameliorate this
5. Obviously, if we knew E(error), we’d correct for it directly

Physical Space Variance Inflation

Inflate all state variables by same amount before assimilation

Capabilities:

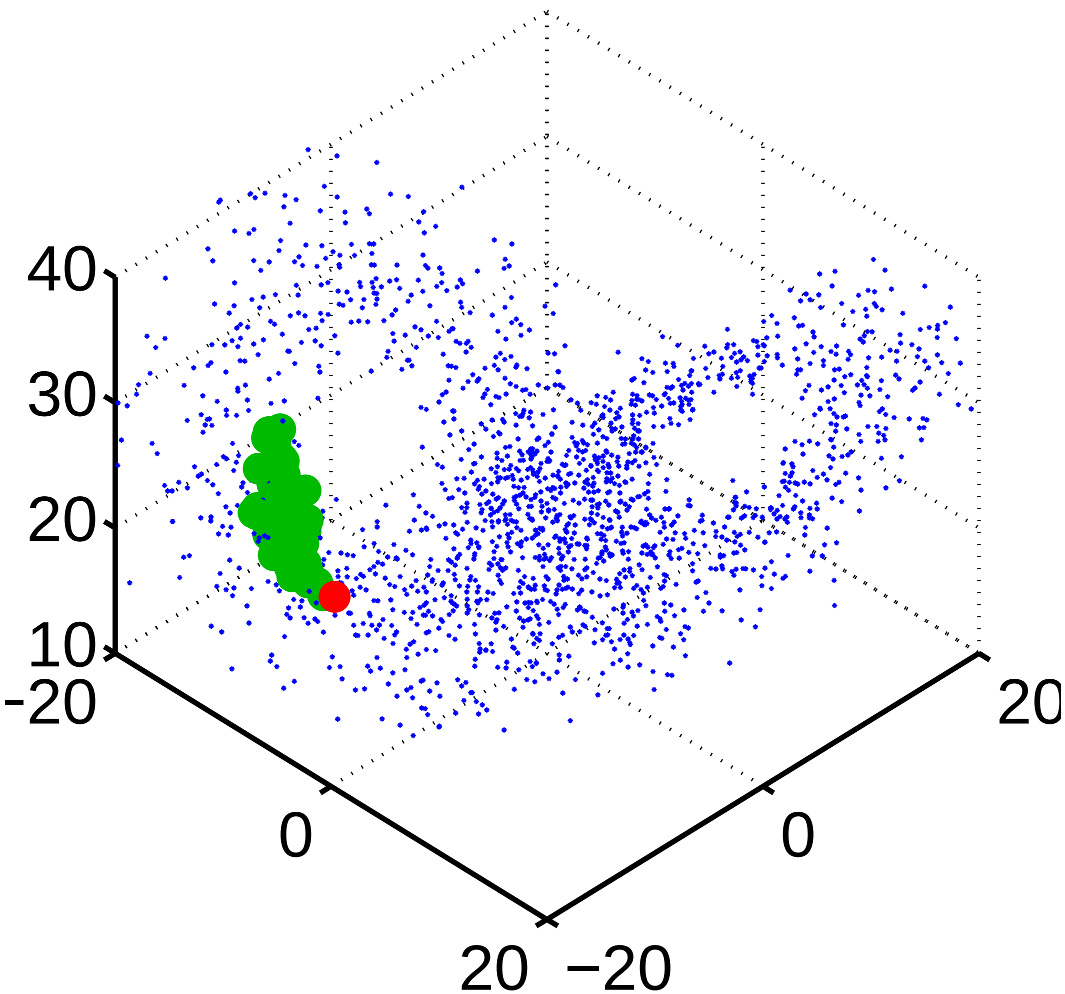
1. Can be very effective for a variety of models.
2. Can maintain linear balances.
3. Stays on local flat manifolds.
4. Simple and inexpensive.

Liabilities:

1. State variables not constrained by observations can 'blow up'.
   For instance unobserved regions near the top of AGCMs.
2. Magnitude of λ normally selected by trial and error.

Physical space covariance inflation in Lorenz-63

Observation outside prior: danger of filter divergence

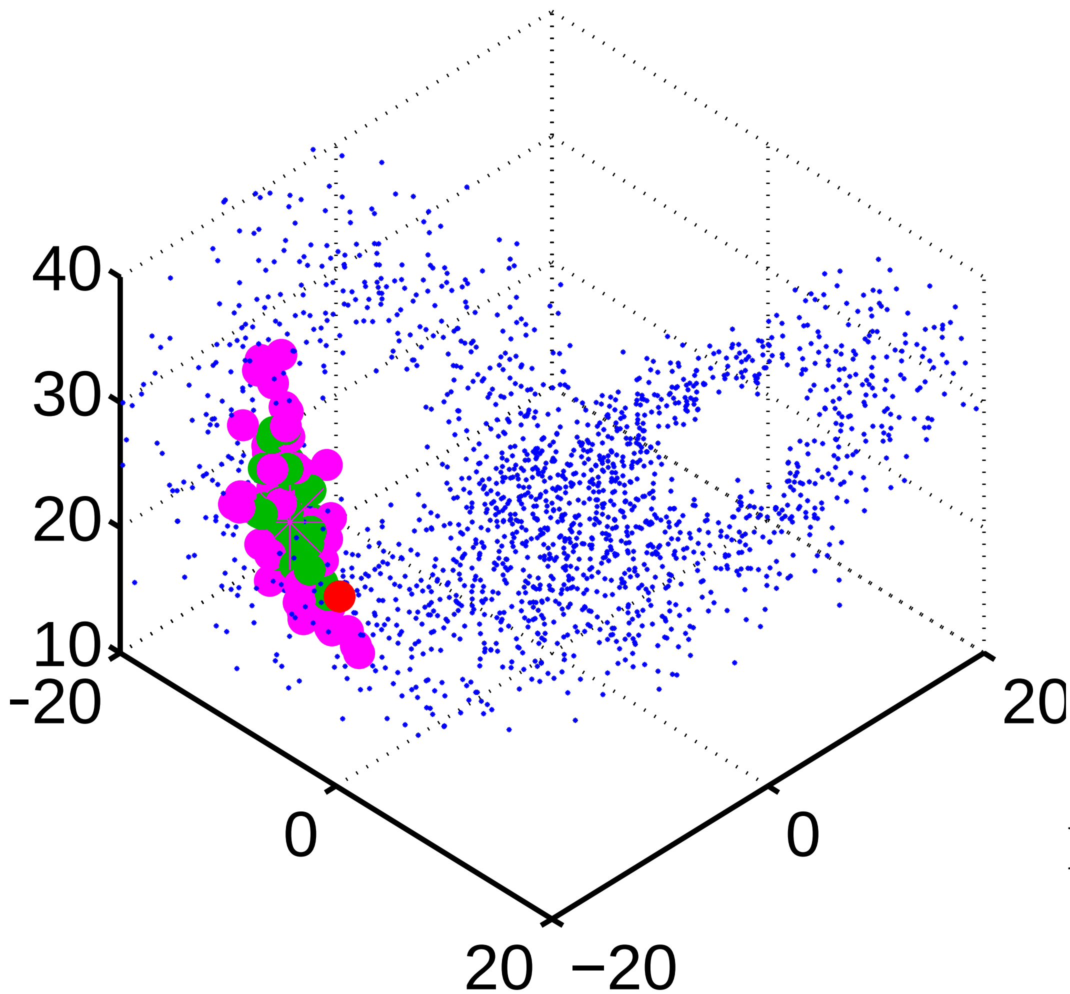


Observation in red.

Prior ensemble in green.

Physical space covariance inflation in Lorenz-63

After inflating, observation is in prior cloud: filter divergence avoided



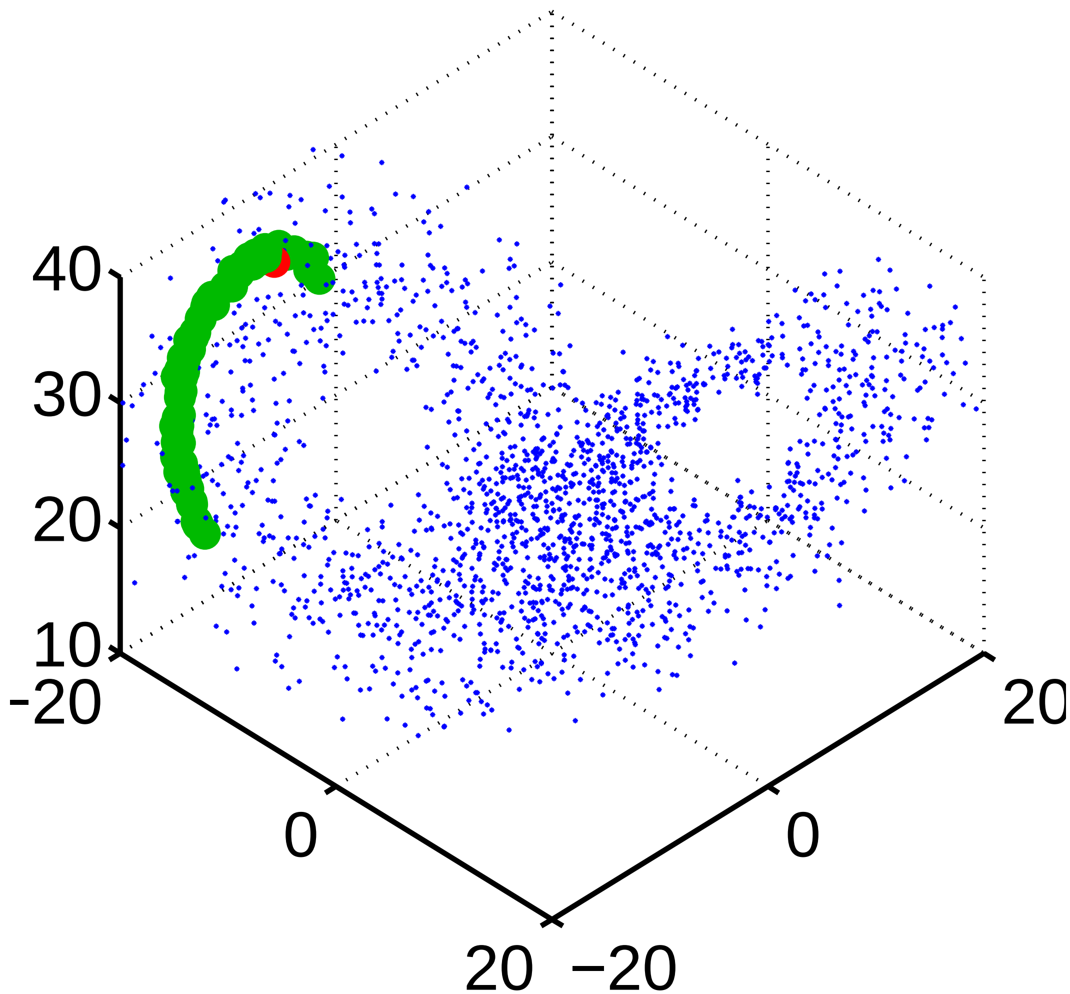
Observation in red.

Prior ensemble in green.

Inflated ensemble in magenta.

Physical space covariance inflation in Lorenz-63

Prior distribution is significantly ‘curved’

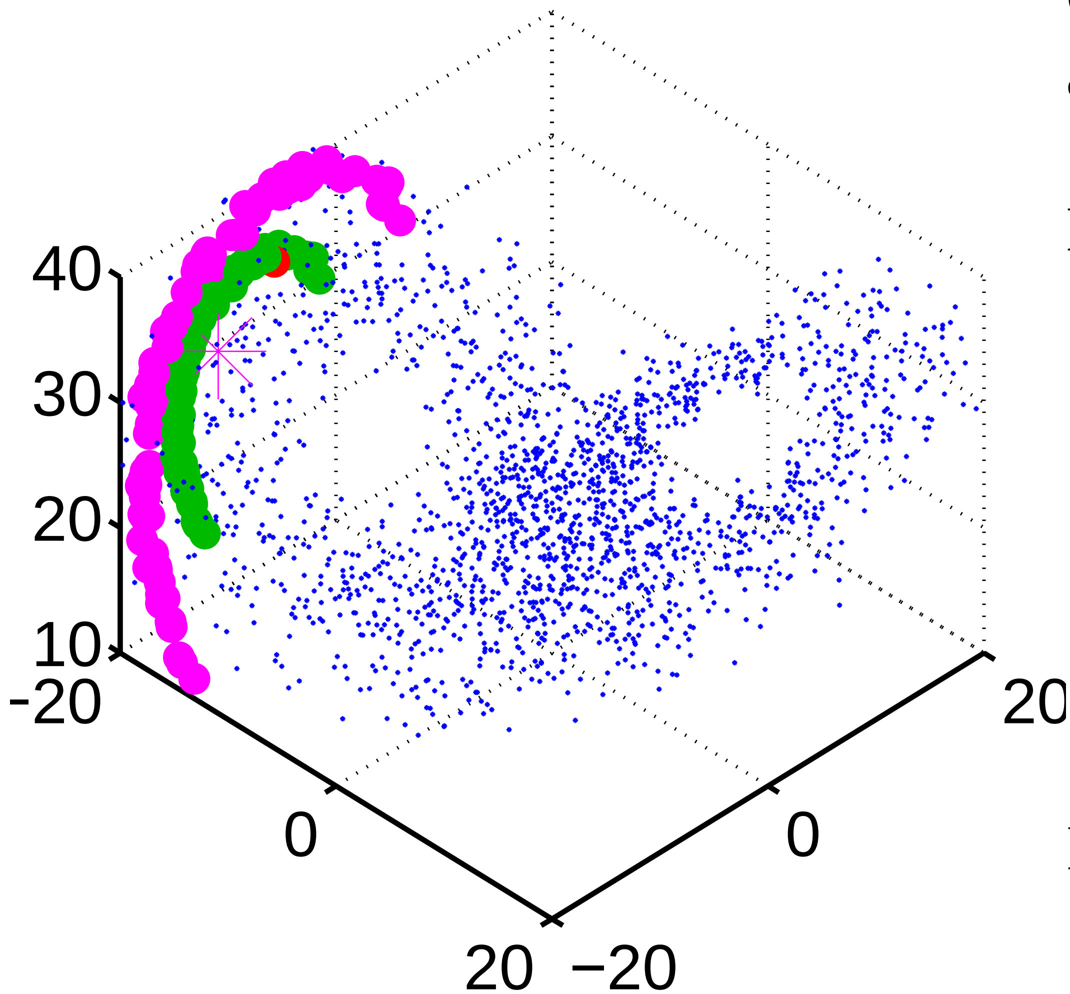


Observation in red.

Prior ensemble in green.

Physical space covariance inflation in Lorenz-63

Inflated prior outside attractor. Posterior will also be off attractor.



Can lead to transient off-attractor behavior or...

Model ‘blow-up’.

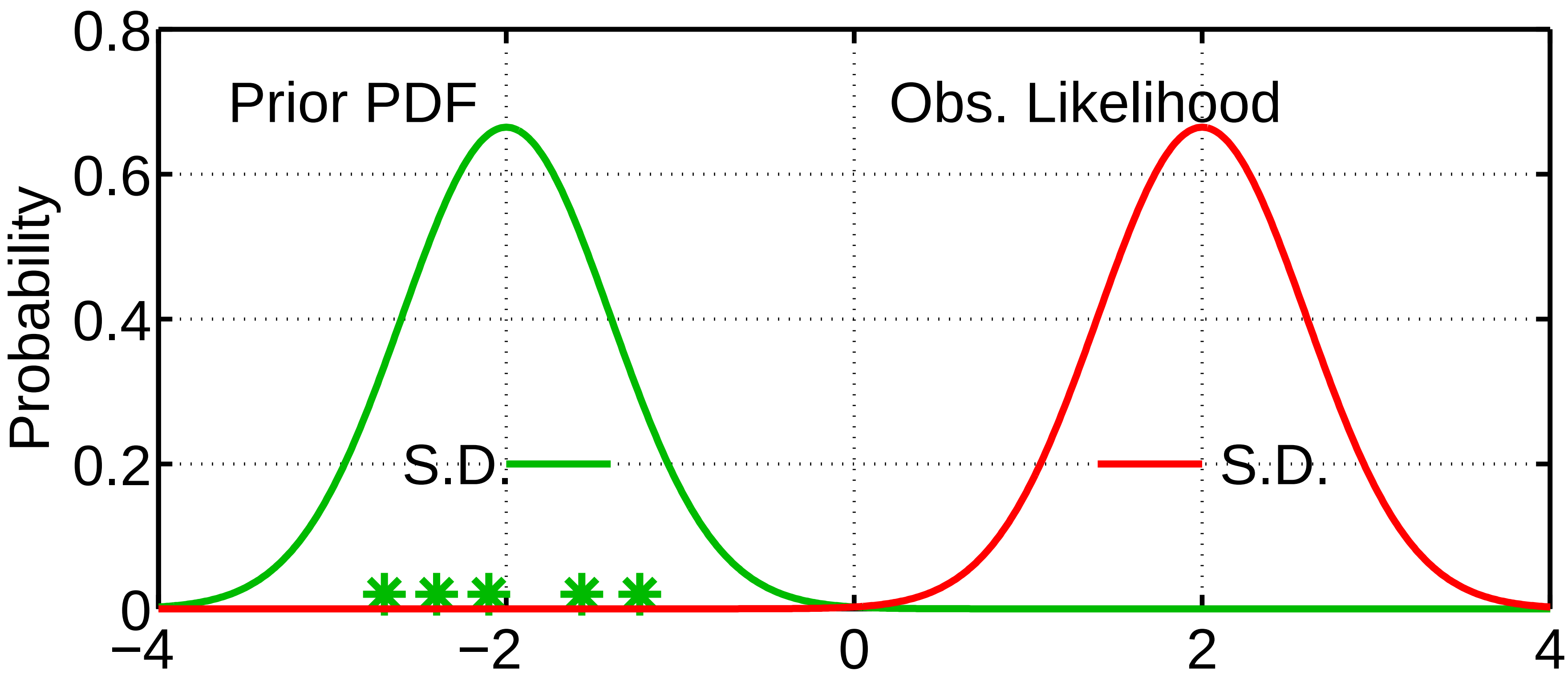
Observation in red.

Prior ensemble in green.

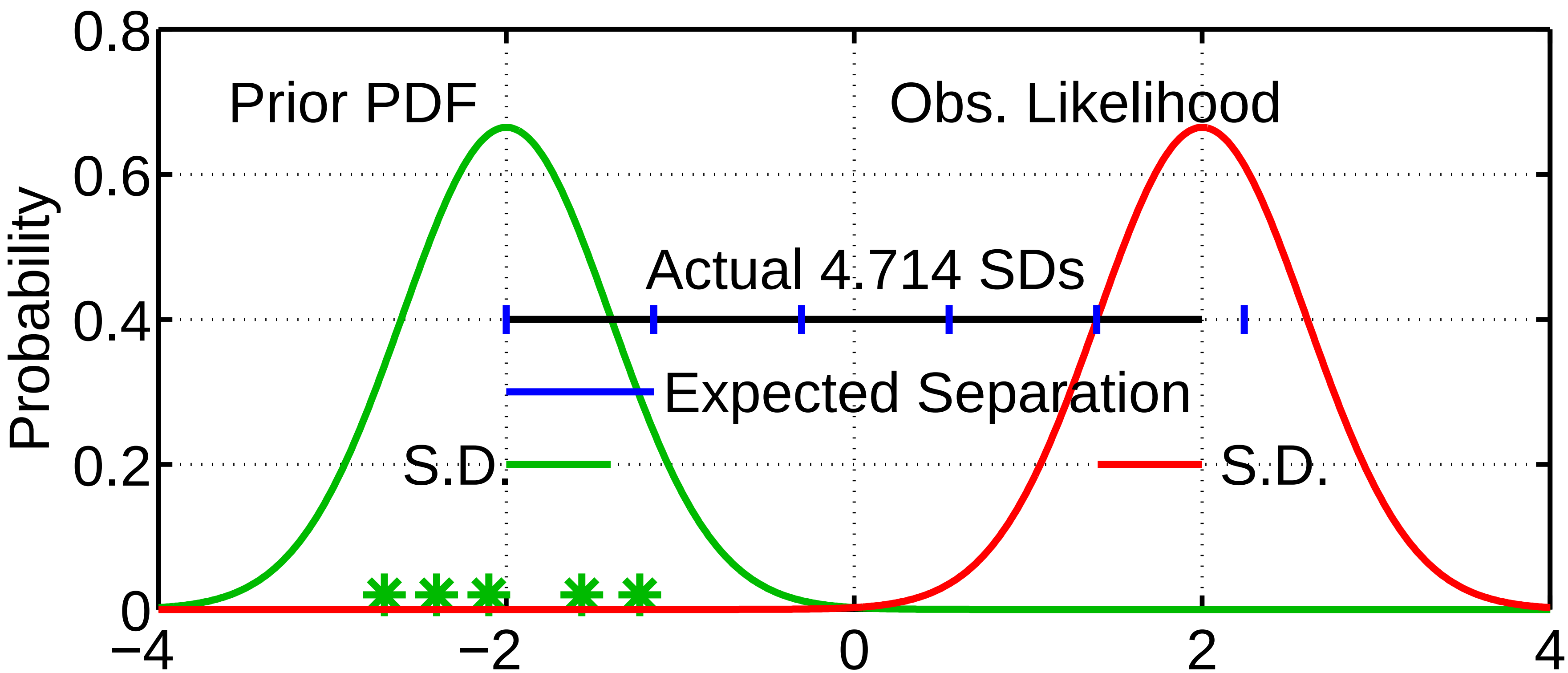
Inflated ensemble in magenta.

Variance inflation for Observations: An Adaptive Error Tolerant Filter

1. For observed variable, have estimate of prior-observed inconsistency



1. For observed variable, have estimate of prior-observed inconsistency



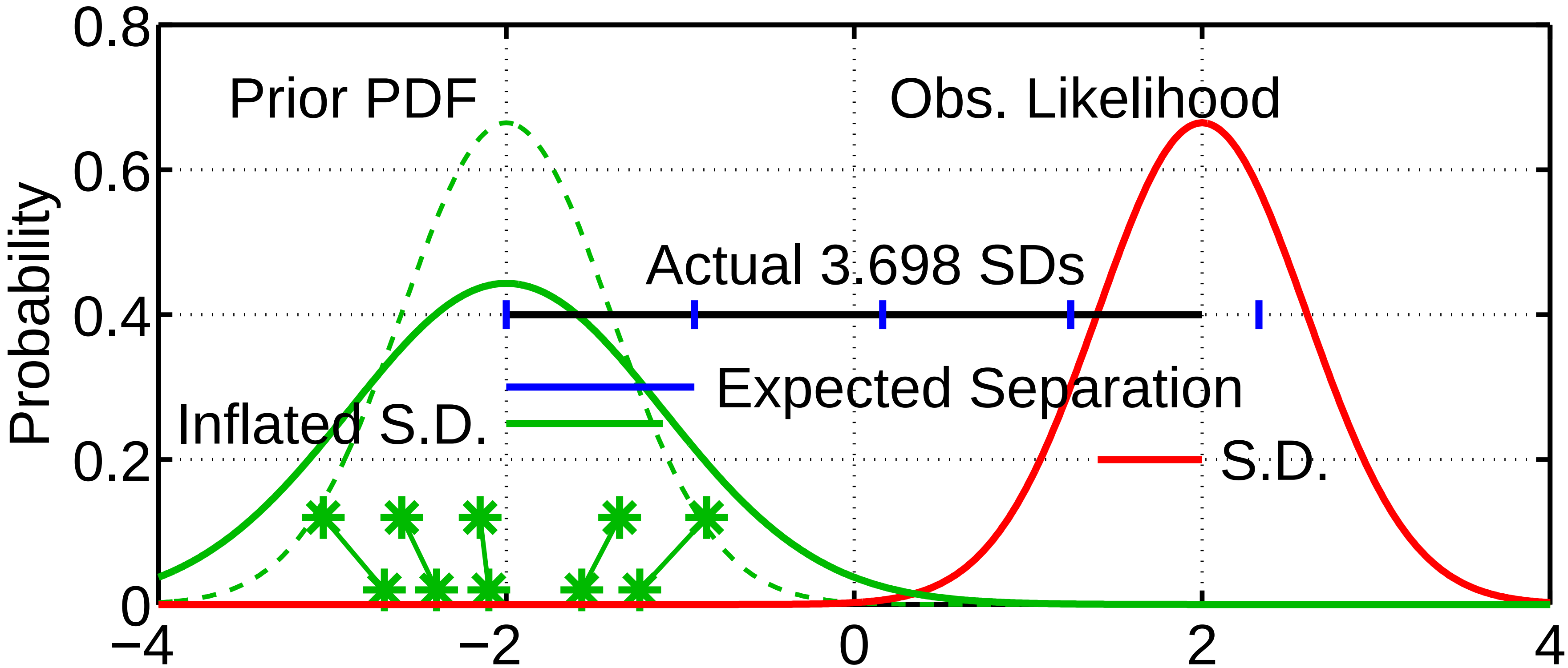
2. Expected(prior mean - observation) = $\sqrt{\sigma^2_{prior} + \sigma^2_{obs}}$.

Assumes that prior and observation are supposed to be unbiased.

Is it model error or random chance?

Variance inflation for Observations: An Adaptive Error Tolerant Filter

1. For observed variable, have estimate of prior-observed inconsistency



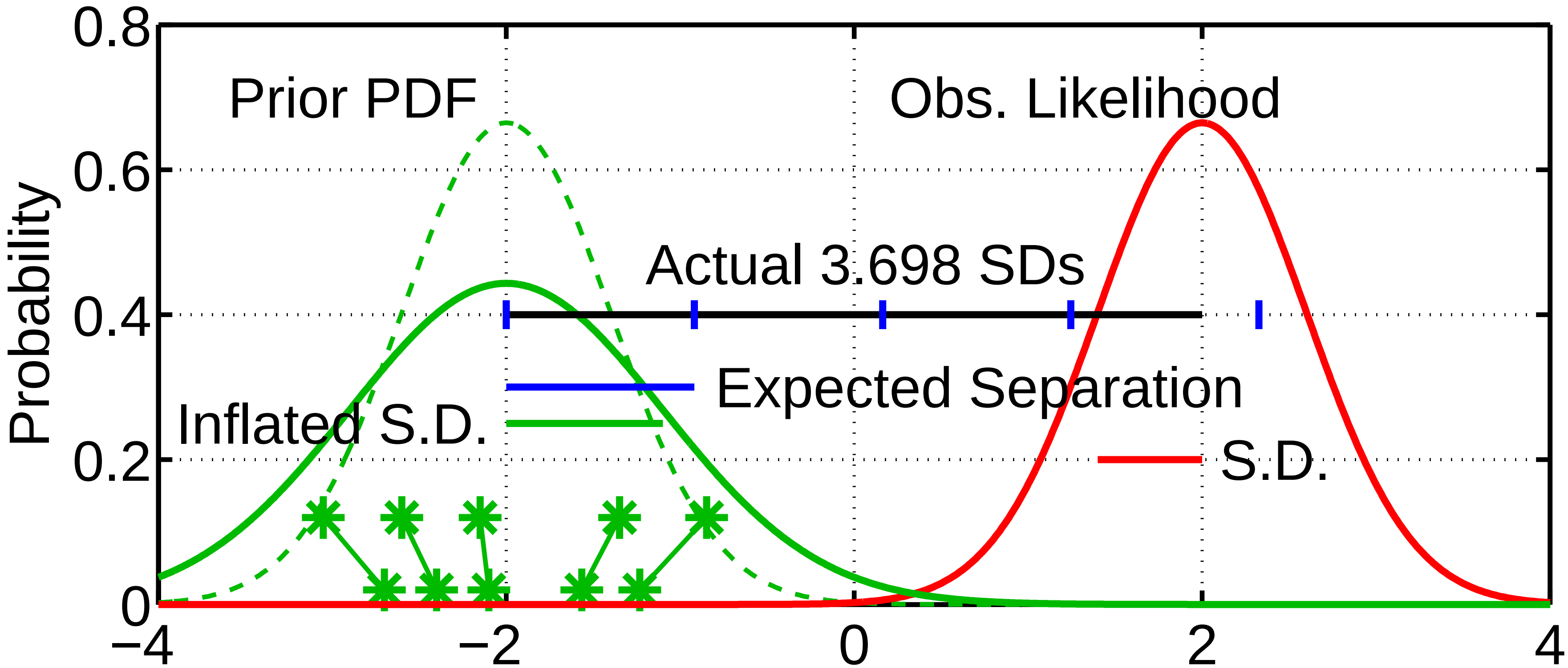
2. Expected(prior mean - observation) = $\sqrt{\sigma_{prior}^2 + \sigma_{obs}^2}$.

3. Inflating increases expected separation.

Increases 'apparent' consistency between prior and observation.

Variance inflation for Observations: An Adaptive Error Tolerant Filter

1. For observed variable, have estimate of prior-observed inconsistency

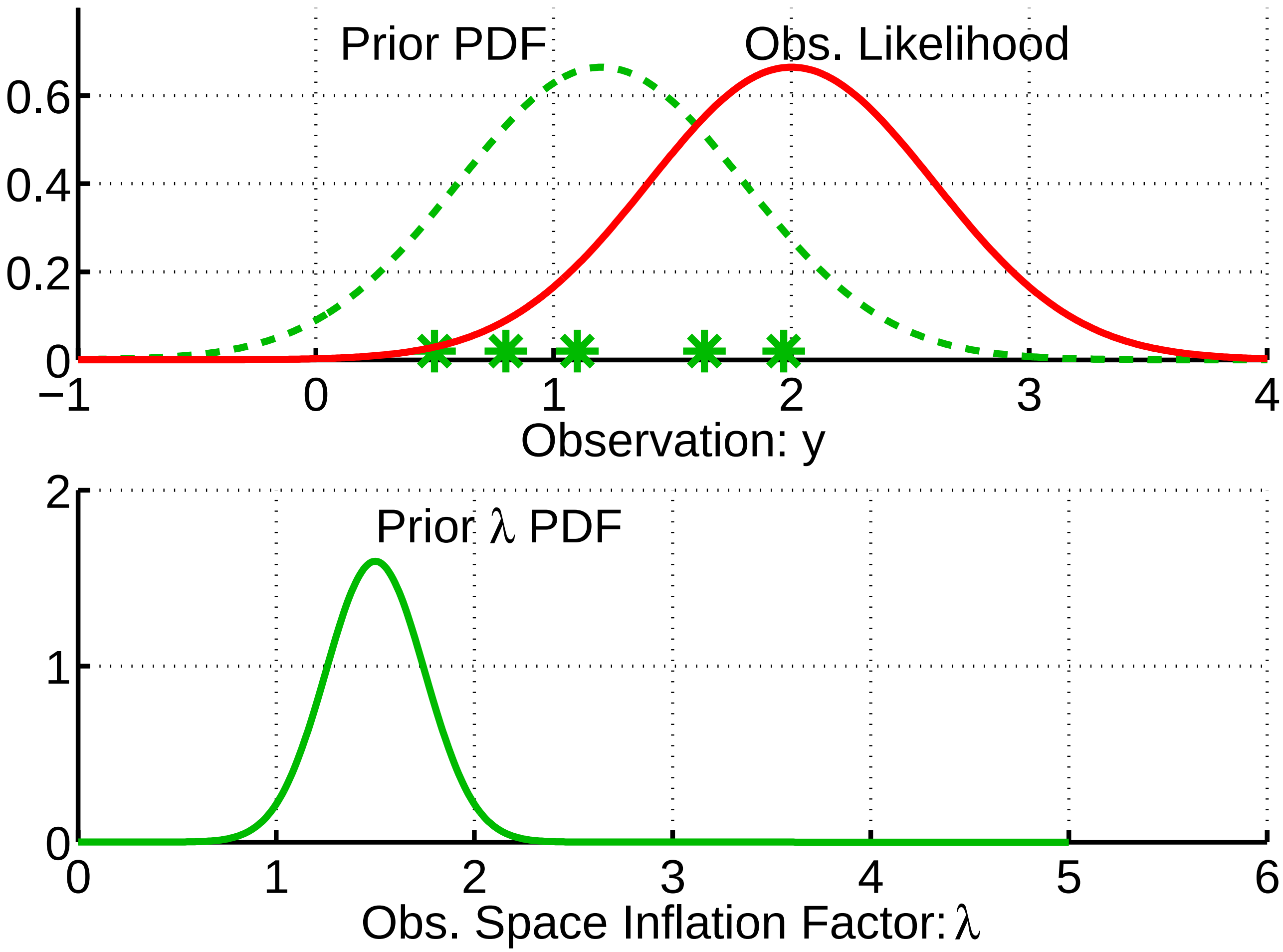


Distance, D, from prior mean y to obs. is $N\left(0, \sqrt{\lambda\sigma_{prior}^2 + \sigma_{obs}^2}\right) = N(0, \theta)$

Prob. y_o is observed given λ: $p(y_o|\lambda) = (2\Pi\theta)^{-1/2}\exp(-D^2/2\theta^2)$

Variance inflation for Observations: An Adaptive Error Tolerant Filter

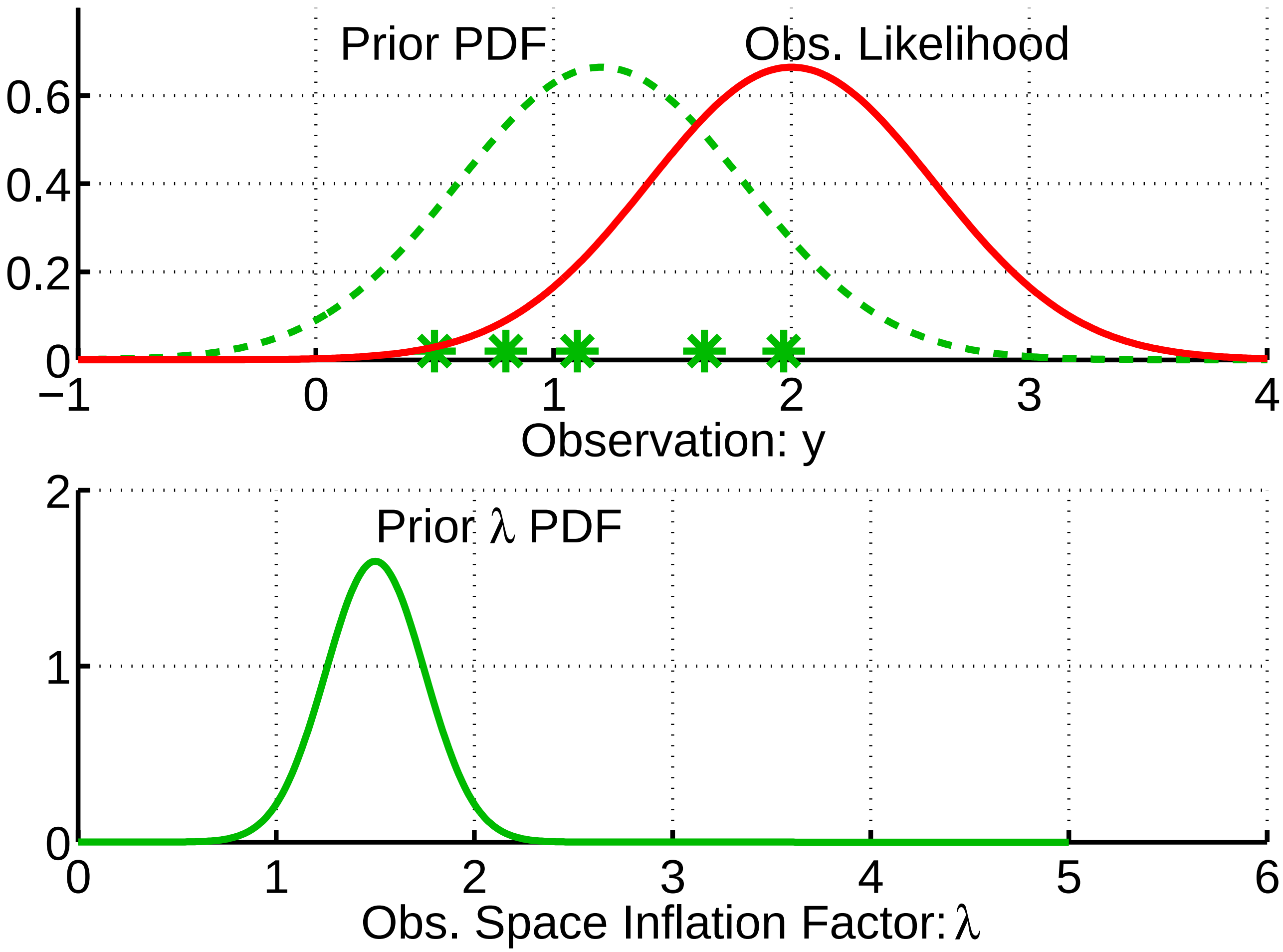
Use Bayesian statistics to get estimate of inflation factor, λ.



Assume some form for prior distribution for λ (Gaussian, gamma).
(Could assume other type of distribution or even use ensemble).

Variance inflation for Observations: An Adaptive Error Tolerant Filter

Use Bayesian statistics to get estimate of inflation factor, λ.



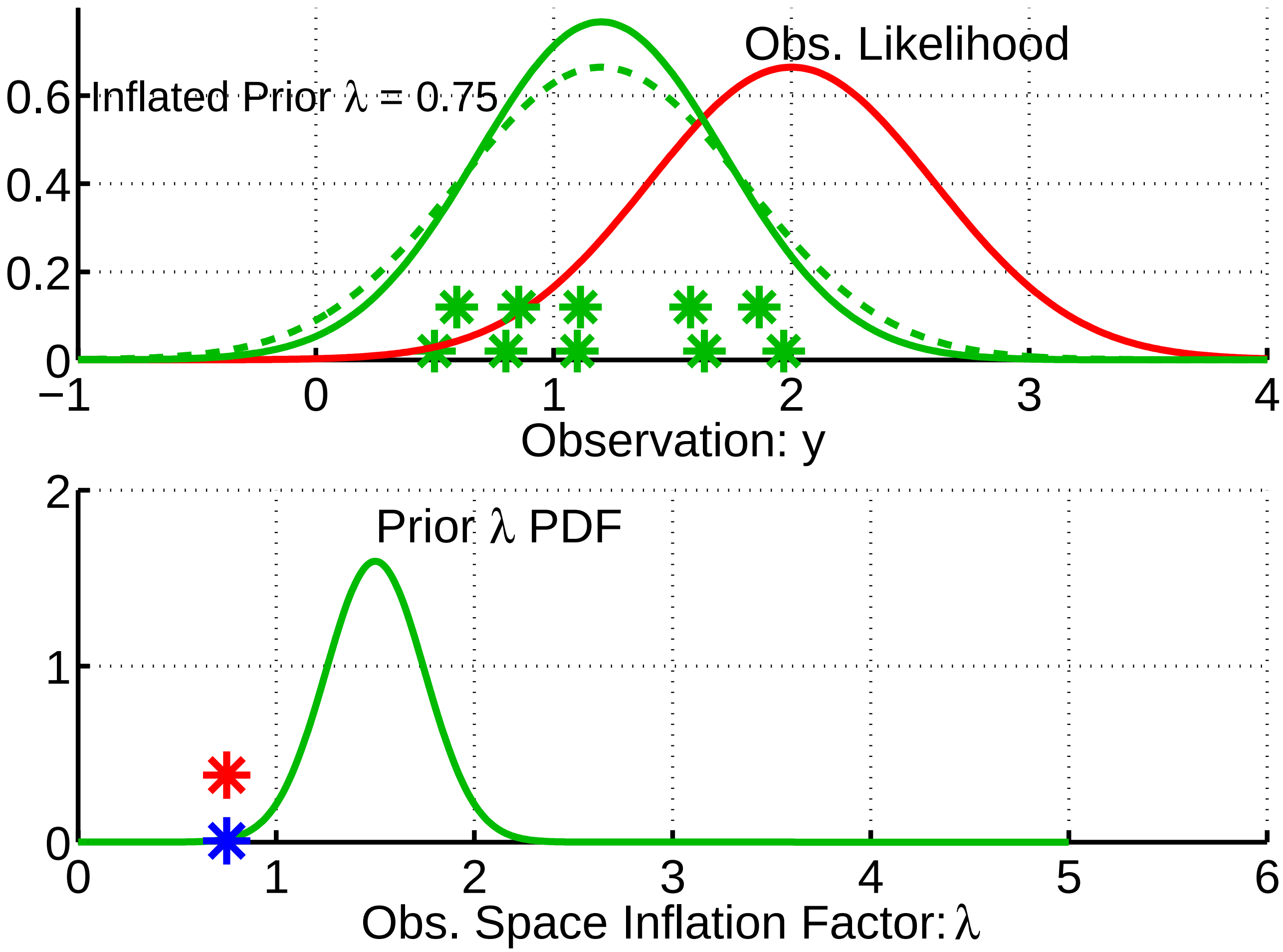
We've assumed a form for prior PDF $p(\lambda, t_k | Y_{t_{k-1}})$.

Recall that $p(y_k | \lambda)$ can be evaluated from normal PDF.

$$p(\lambda, t_k | Y_{t_k}) = p(y_k | \lambda) p(\lambda, t_k | Y_{t_{k-1}}) / normalization.$$

Variance inflation for Observations: An Adaptive Error Tolerant Filter

Use Bayesian statistics to get estimate of inflation factor, λ.



Get $p(y_k|\lambda = 0.75)$

from normal PDF.

Multiply by

$p(\lambda = 0.75, t_k|Y_{t_{k-1}})$

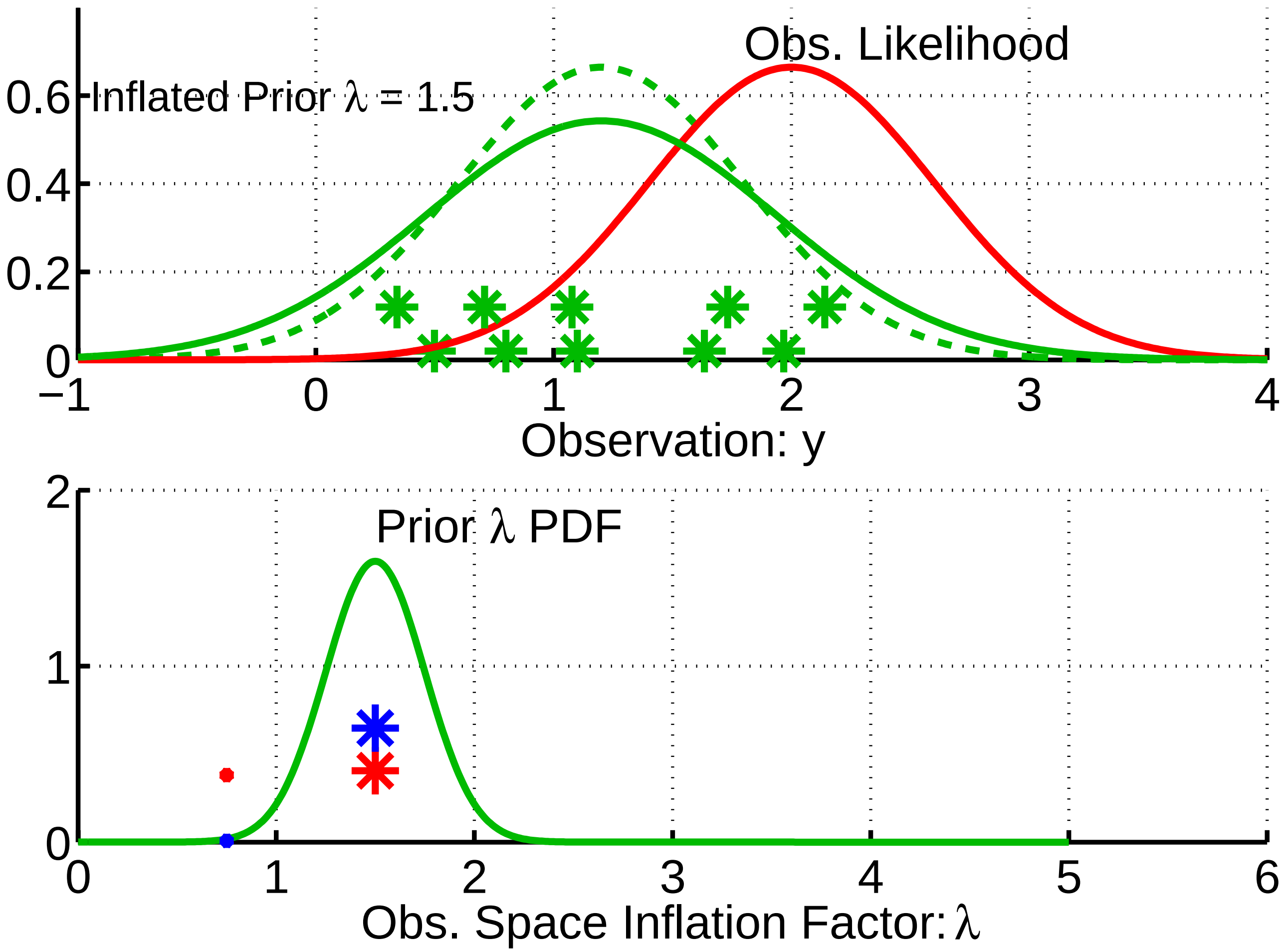
to get

$p(\lambda = 0.75, t_k|Y_{t_k})$

$$p(\lambda, t_k|Y_{t_k}) = p(y_k|\lambda)p(\lambda, t_k|Y_{t_{k-1}})/normalization.$$

Variance inflation for Observations: An Adaptive Error Tolerant Filter

Use Bayesian statistics to get estimate of inflation factor, λ.



Get $p(y_k|\lambda = 1.50)$

from normal PDF.

Multiply by

$p(\lambda = 1.50, t_k|Y_{t_{k-1}})$

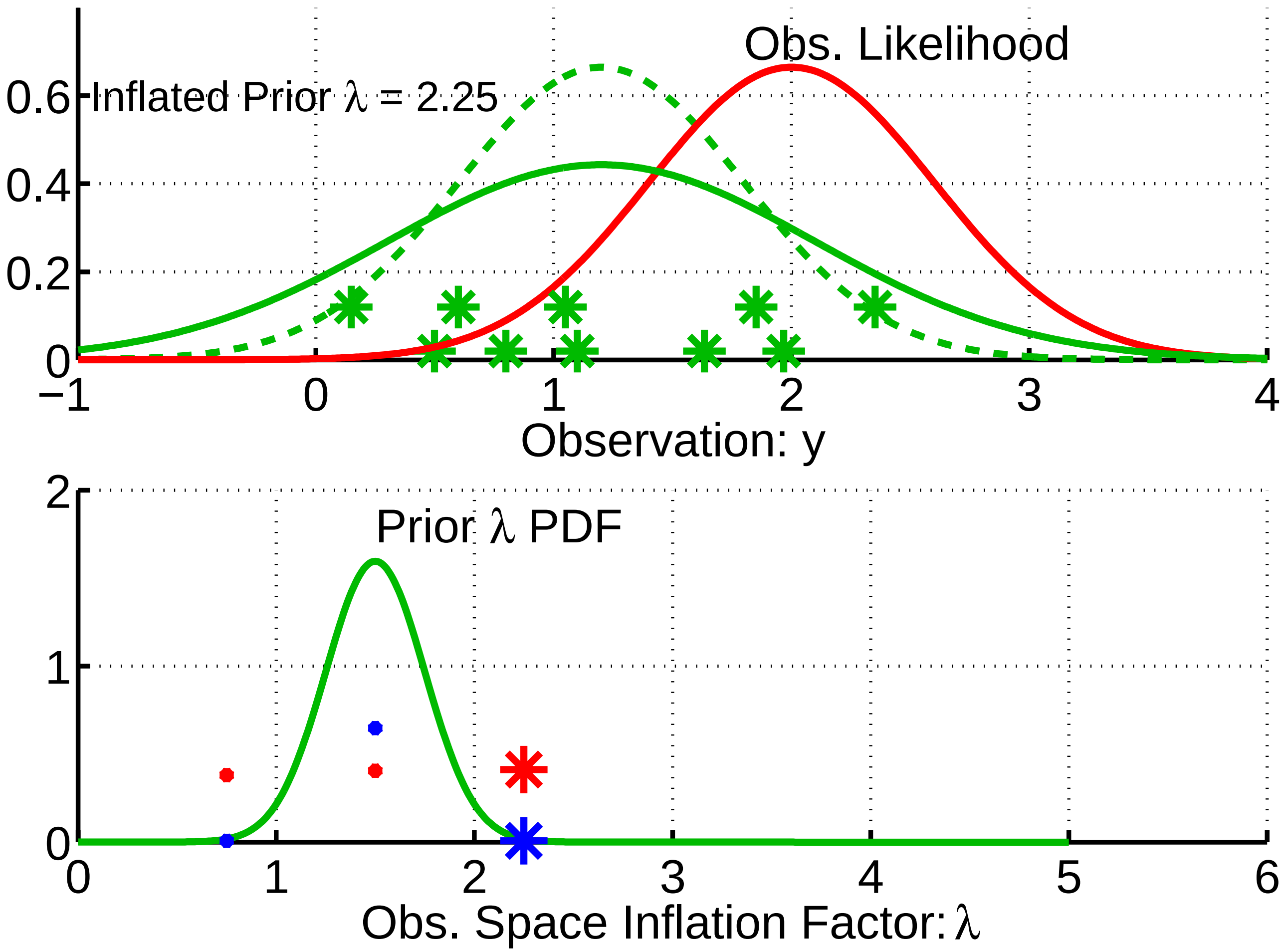
to get

$p(\lambda = 1.50, t_k|Y_{t_k})$

$$p(\lambda, t_k|Y_{t_k}) = p(y_k|\lambda)p(\lambda, t_k|Y_{t_{k-1}})/normalization.$$

Variance inflation for Observations: An Adaptive Error Tolerant Filter

Use Bayesian statistics to get estimate of inflation factor, λ.



Get $p(y_k|\lambda = 2.25)$

from normal PDF.

Multiply by

$p(\lambda = 2.25, t_k|Y_{t_{k-1}})$

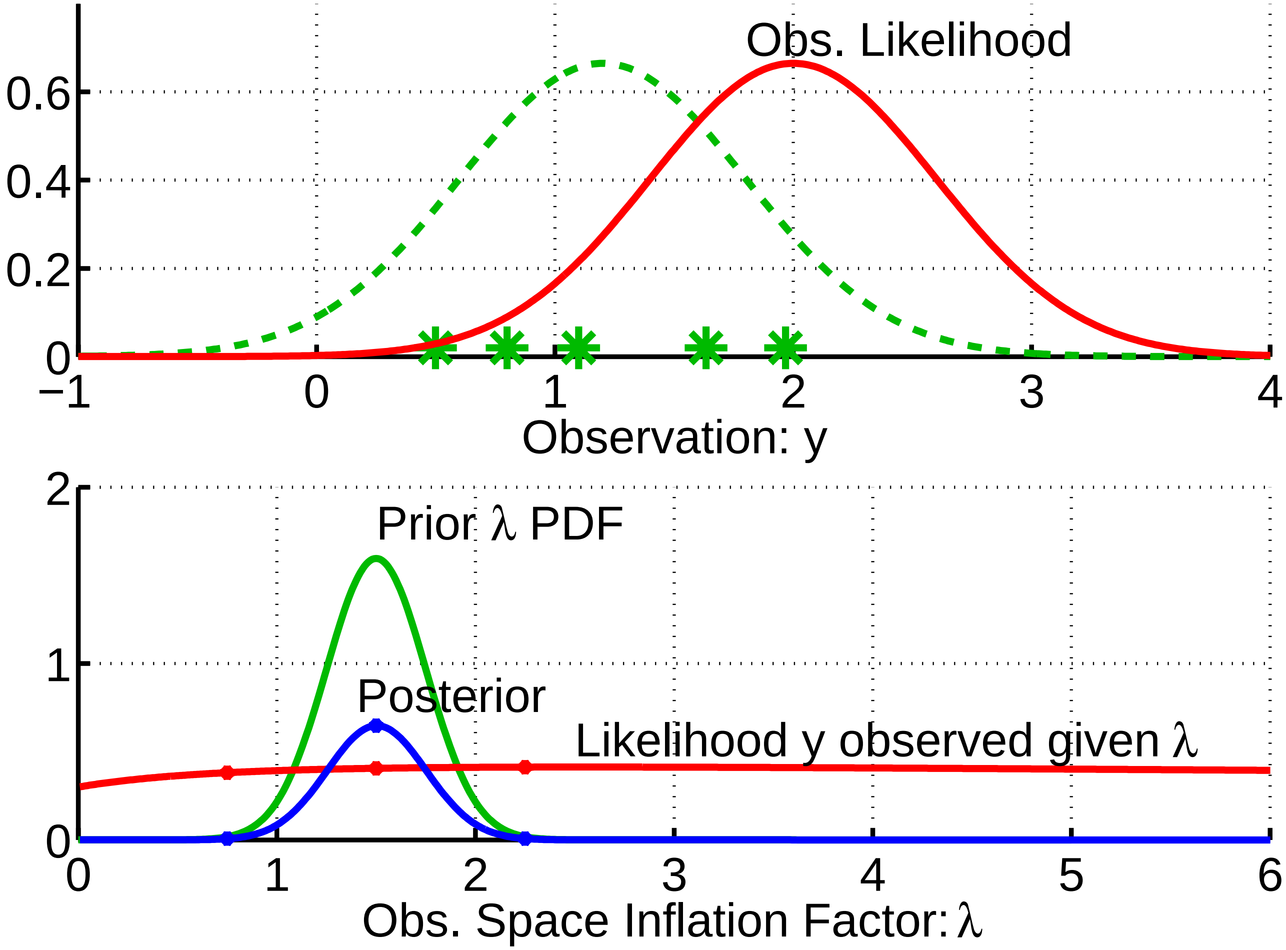
to get

$p(\lambda = 2.25, t_k|Y_{t_k})$

$$p(\lambda, t_k|Y_{t_k}) = p(y_k|\lambda)p(\lambda, t_k|Y_{t_{k-1}})/normalization.$$

Variance inflation for Observations: An Adaptive Error Tolerant Filter

Use Bayesian statistics to get estimate of inflation factor, λ.



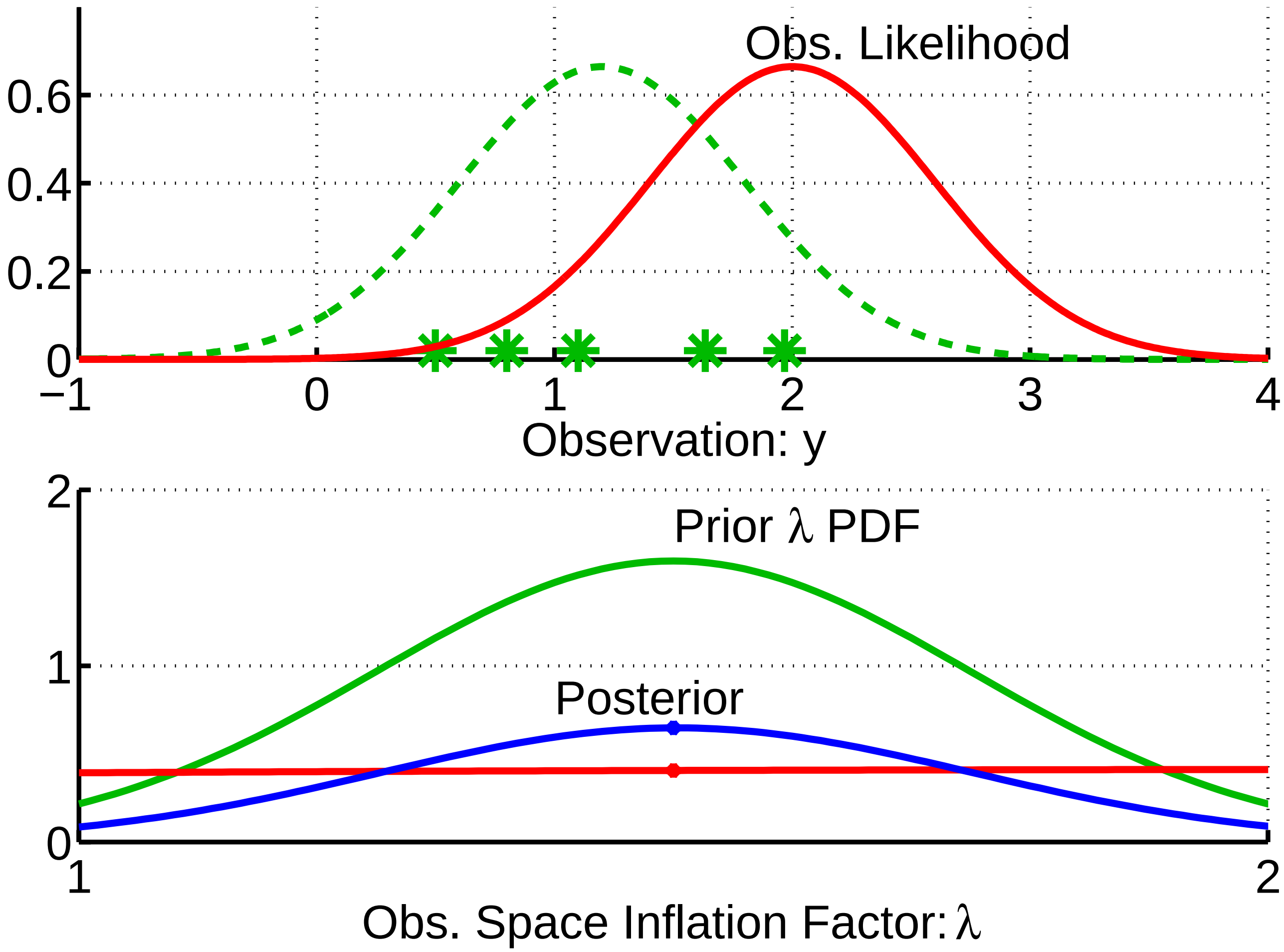
Repeat for a range of values of λ.

Now must get posterior in same form as prior.

$$p(\lambda, t_k | Y_{t_k}) = p(y_k | \lambda) p(\lambda, t_k | Y_{t_{k-1}}) / normalization.$$

Variance inflation for Observations: An Adaptive Error Tolerant Filter

Use Bayesian statistics to get estimate of inflation factor, λ.



Very little information about λ in a single observation.

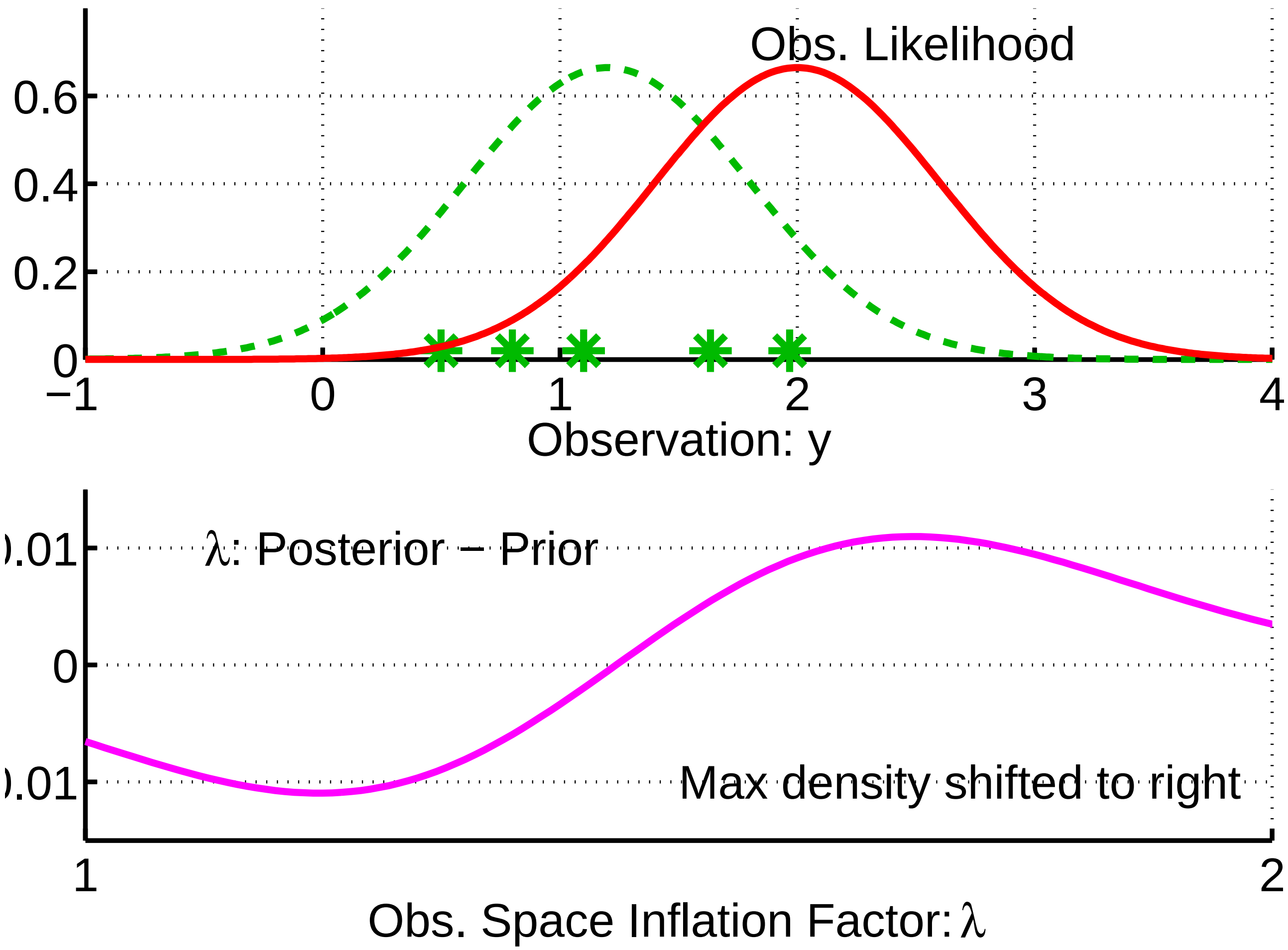
Posterior and prior are very similar.

Normalized posterior indistinguishable from prior.

$$p(\lambda, t_k | Y_{t_k}) = p(y_k | \lambda) p(\lambda, t_k | Y_{t_{k-1}}) / normalization.$$

Variance inflation for Observations: An Adaptive Error Tolerant Filter

Use Bayesian statistics to get estimate of inflation factor, λ.



Very little information about λ in a single observation.

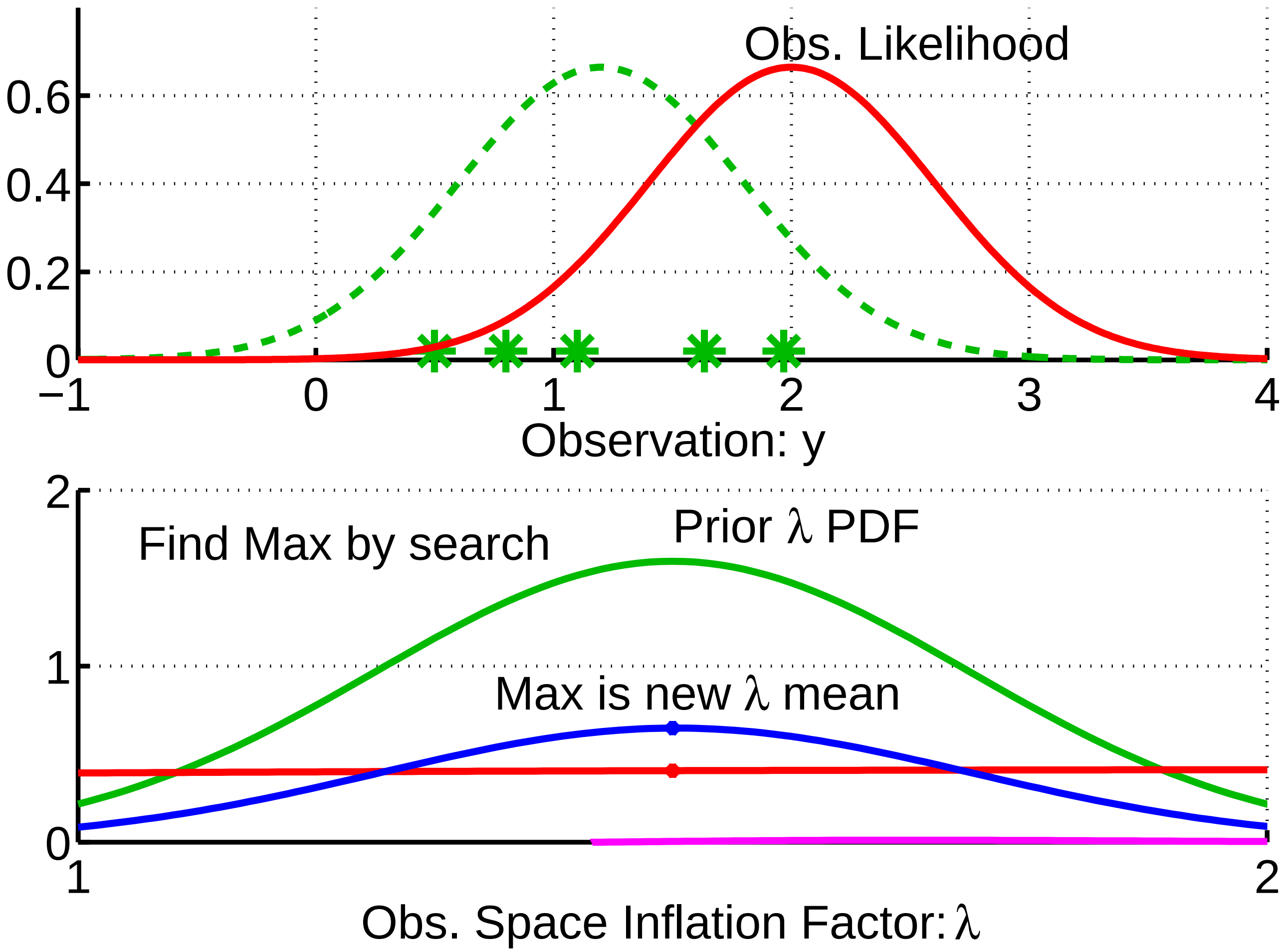
Posterior and prior are very similar.

Difference shows slight shift to larger values of λ.

$$p(\lambda, t_k | Y_{t_k}) = p(y_k | \lambda) p(\lambda, t_k | Y_{t_{k-1}}) / normalization.$$

Variance inflation for Observations: An Adaptive Error Tolerant Filter

Use Bayesian statistics to get estimate of inflation factor, λ.



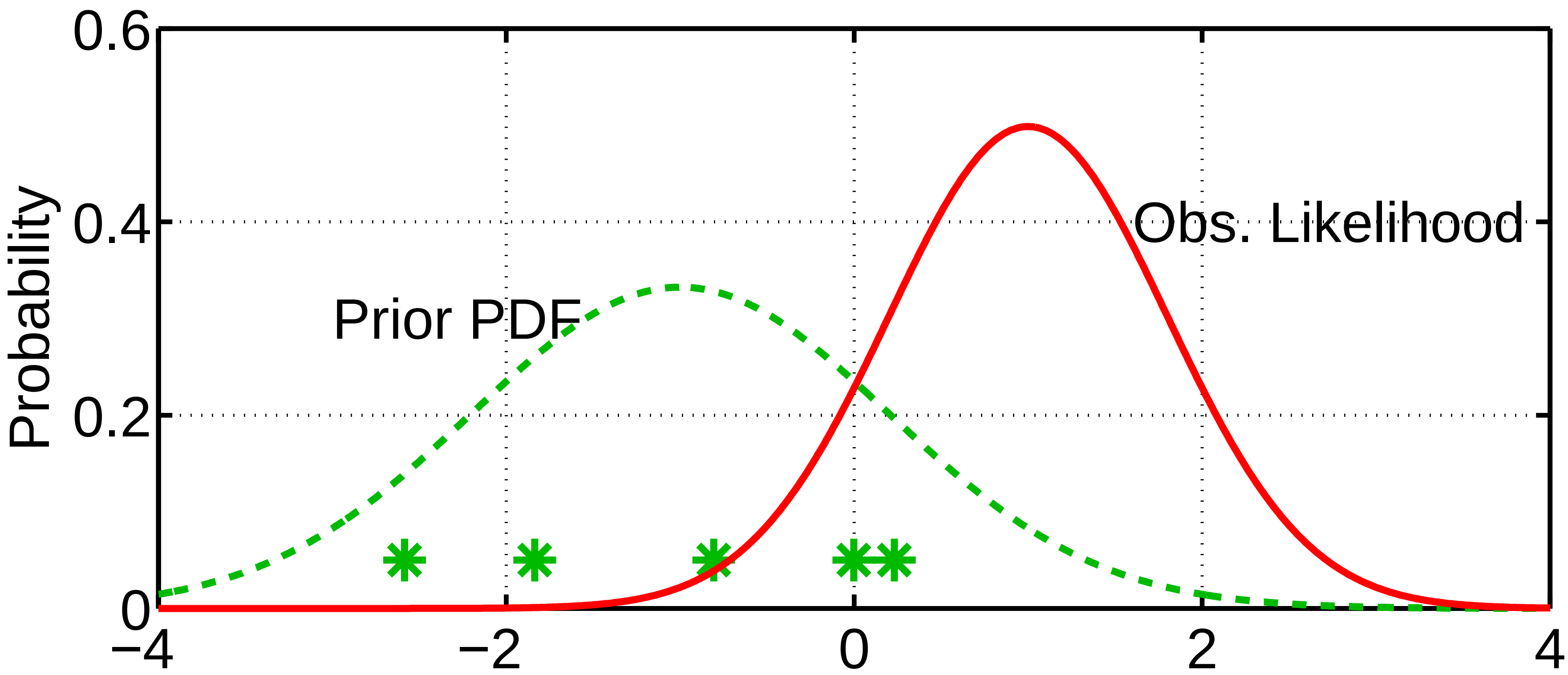
One option is to use Gaussian prior for λ.

Select max of posterior as mean of updated Gaussian.

Do a fit for updated standard deviation.

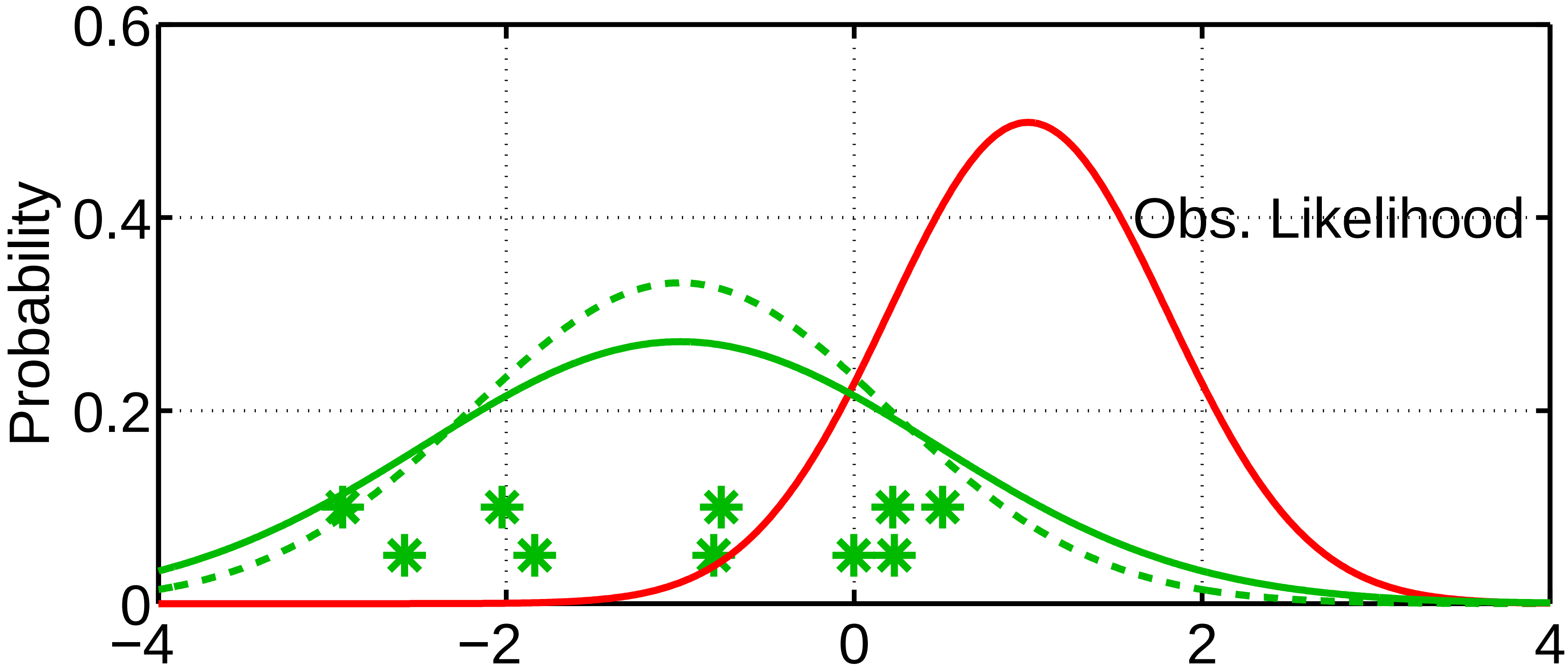
$$p(\lambda, t_k | Y_{t_k}) = p(y_k | \lambda) p(\lambda, t_k | Y_{t_{k-1}}) / normalization.$$

Observation Space Computations with Adaptive Error Correction



1. Compute updated inflation distribution, $p(\lambda, t_k | Y_{t_k})$.

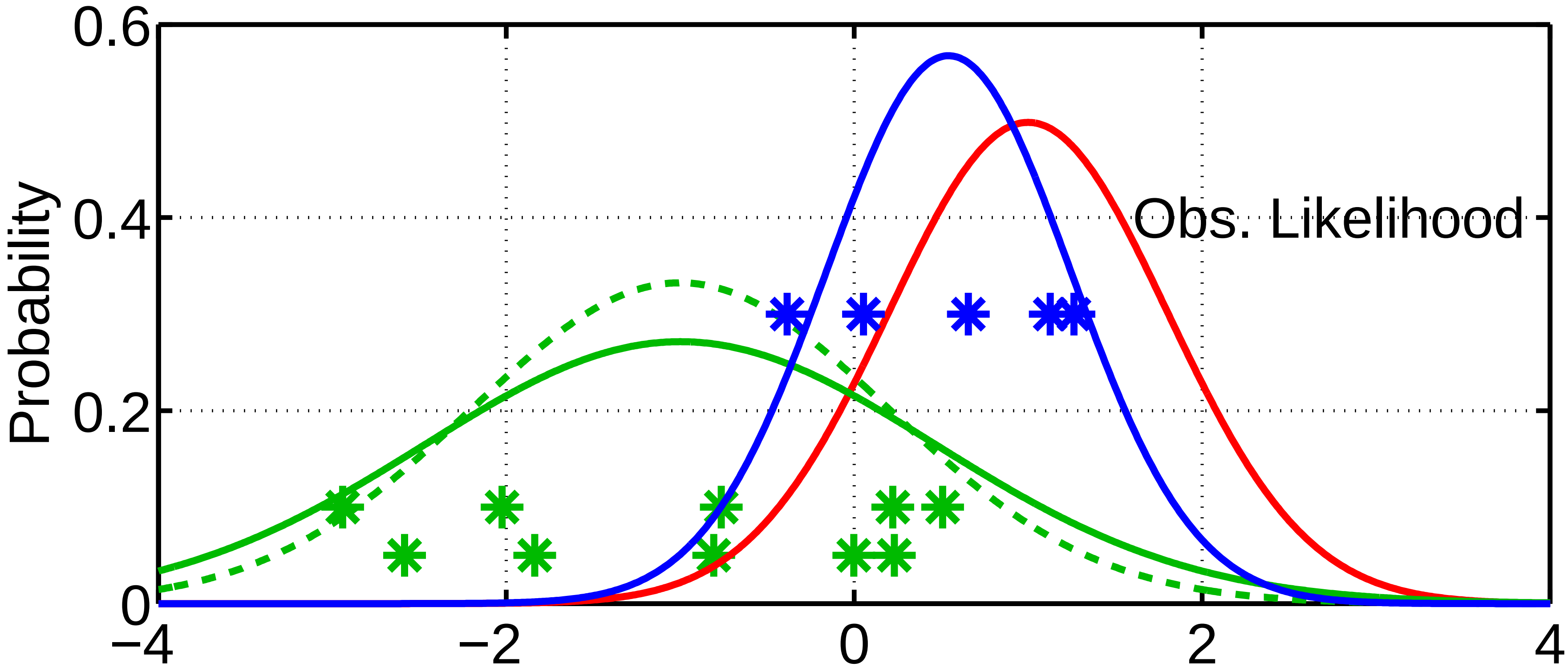
Observation Space Computations with Adaptive Error Correction



1. Compute updated inflation distribution, $p(\lambda, t_k | Y_{t_k})$.

2. Inflate ensemble using mean of updated λ distribution.

Observation Space Computations with Adaptive Error Correction

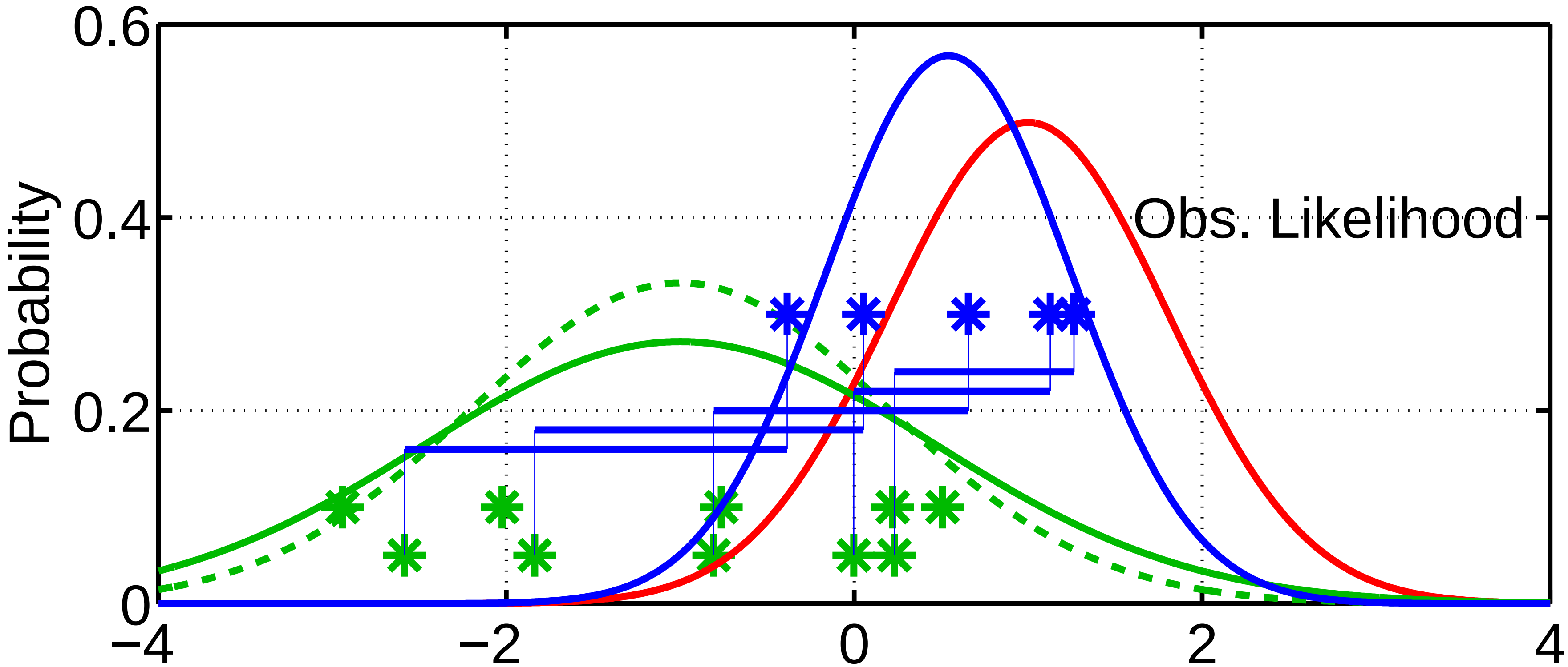


1. Compute updated inflation distribution, $p(\lambda, t_k | Y_{t_k})$.

2. Inflate ensemble using mean of updated λ distribution.

3. Compute posterior for y using inflated prior.

Observation Space Computations with Adaptive Error Correction



1. Compute updated inflation distribution, $p(\lambda, t_k | Y_{t_k})$.

2. Inflate ensemble using mean of updated λ distribution.

3. Compute posterior for y using inflated prior.

4. Compute increments from ORIGINAL prior ensemble.

Phase 4: Quick look at a real atmospheric application

Results from CAM Assimilation: January, 2003

Model:

CAM 3.0 T42L26

U,V, T, Q and PS state variables impacted by observations.

Land model (CLM 2.0) not impacted by observations.

Climatological SSTs.

Assimilation / Prediction Experiments:

80 member ensemble divided into 4 equal groups.

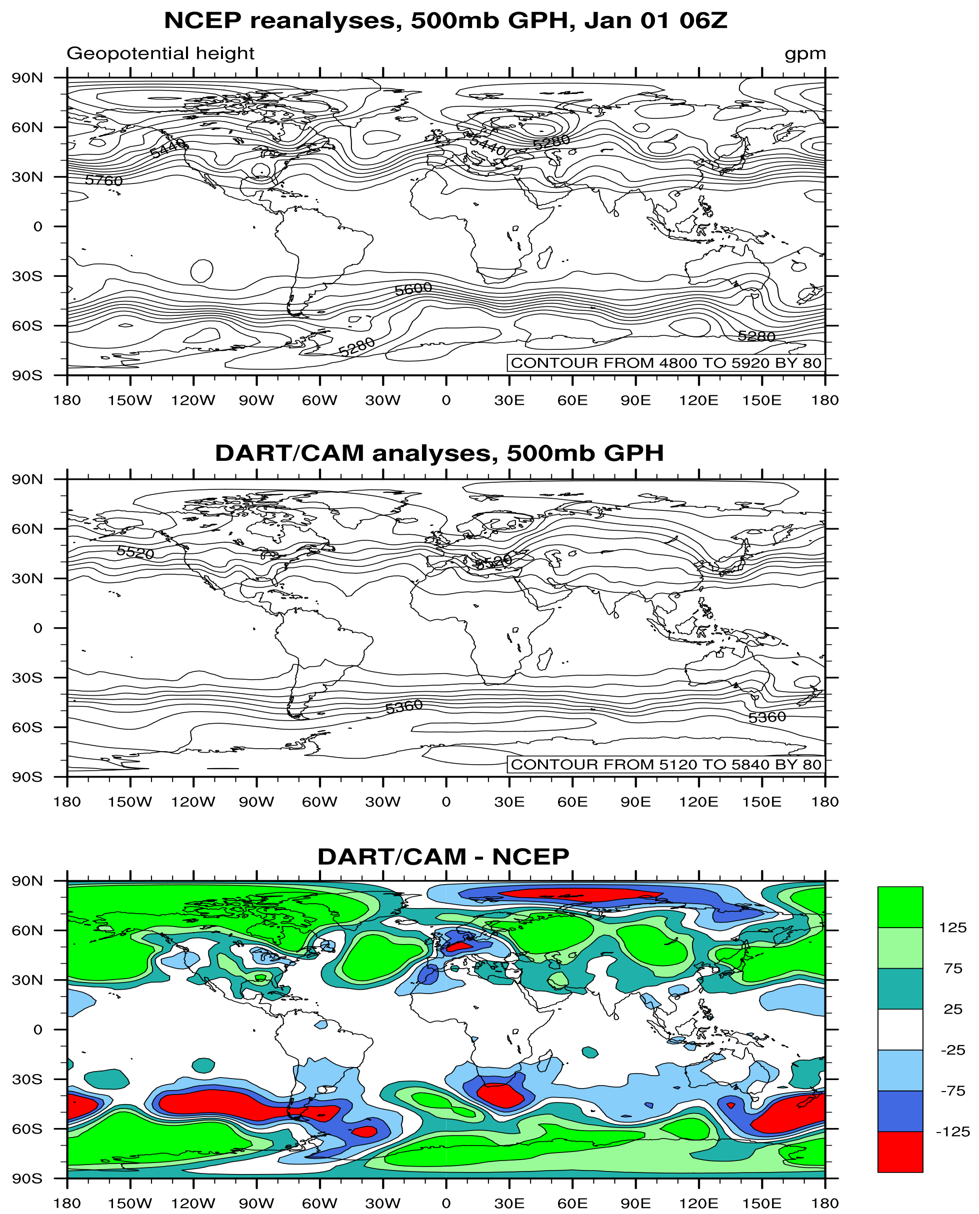
Initialized from a climatological distribution (huge spread).

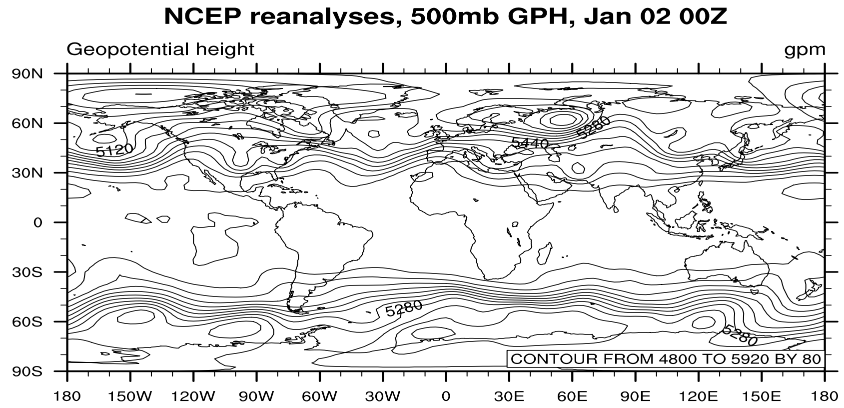
Initial tests for January, 2003.

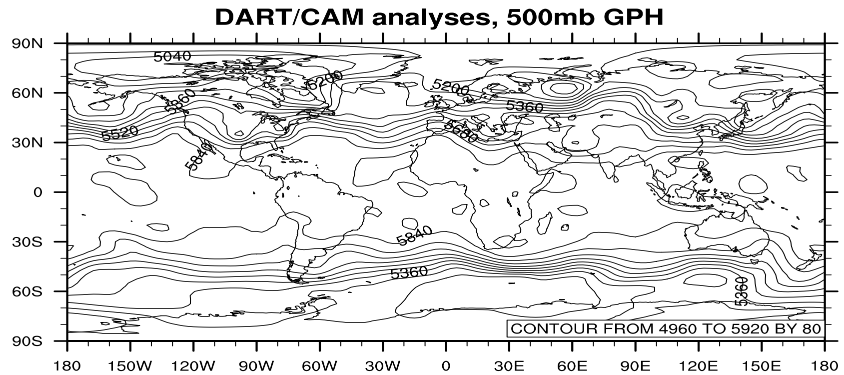
Uses most observations used in reanalysis
(Radiosondes, ACARS, Satellite Winds..., no surface obs.).

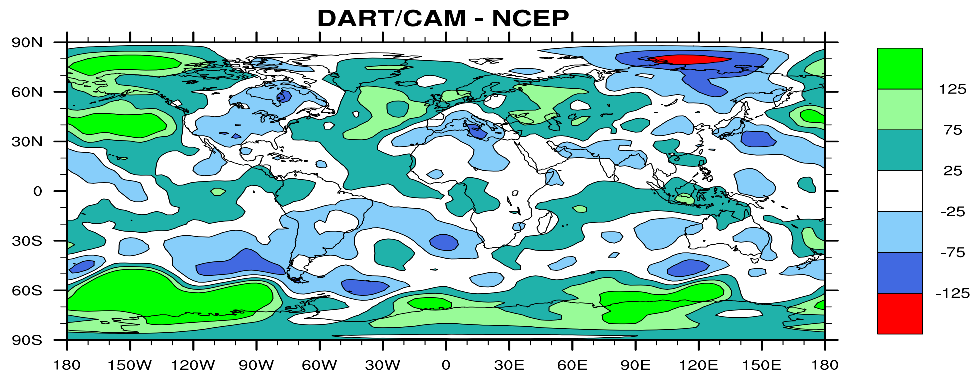
Assimilated every 6 hours; +/- 1.5 hour window for obs.

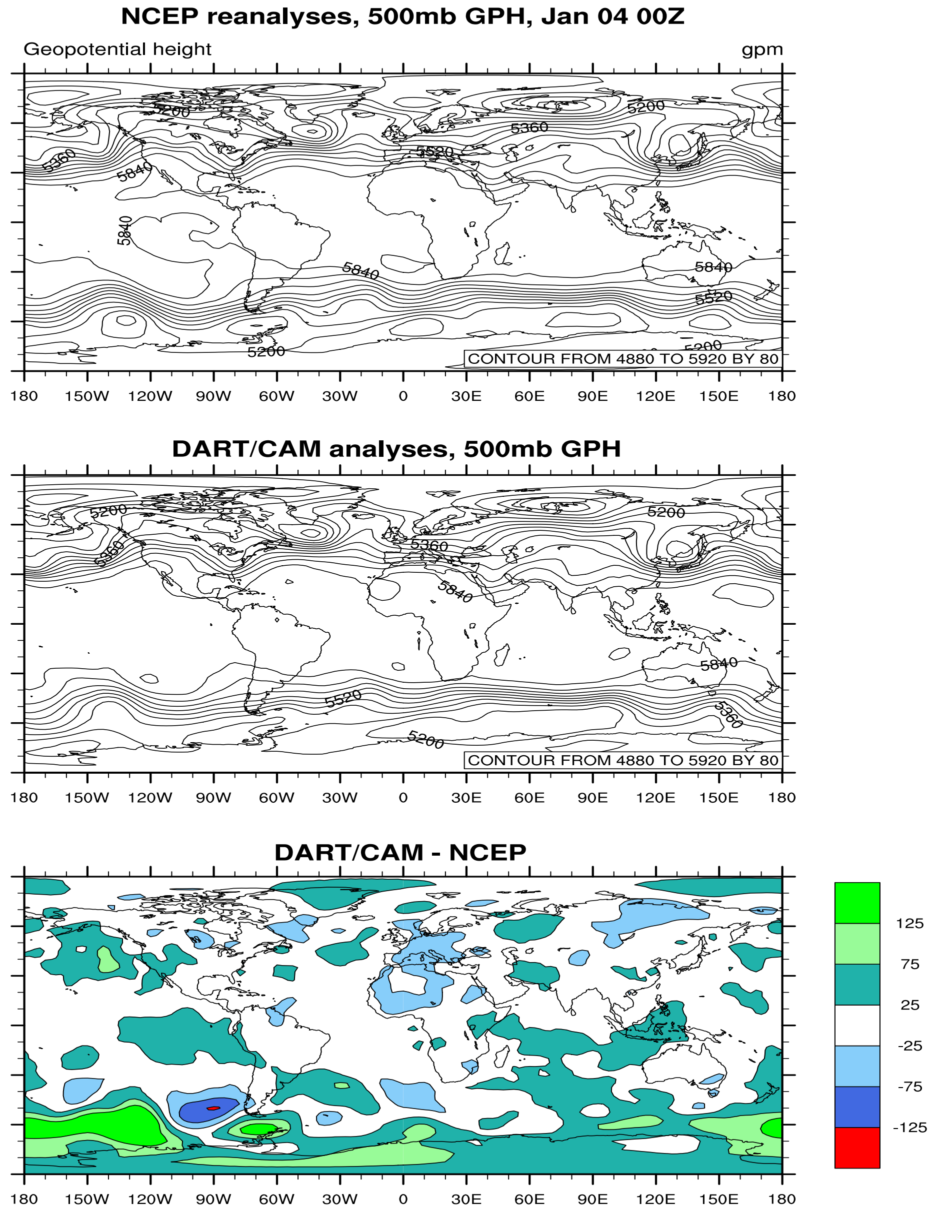
Adaptive error correction algorithm with fixed variance

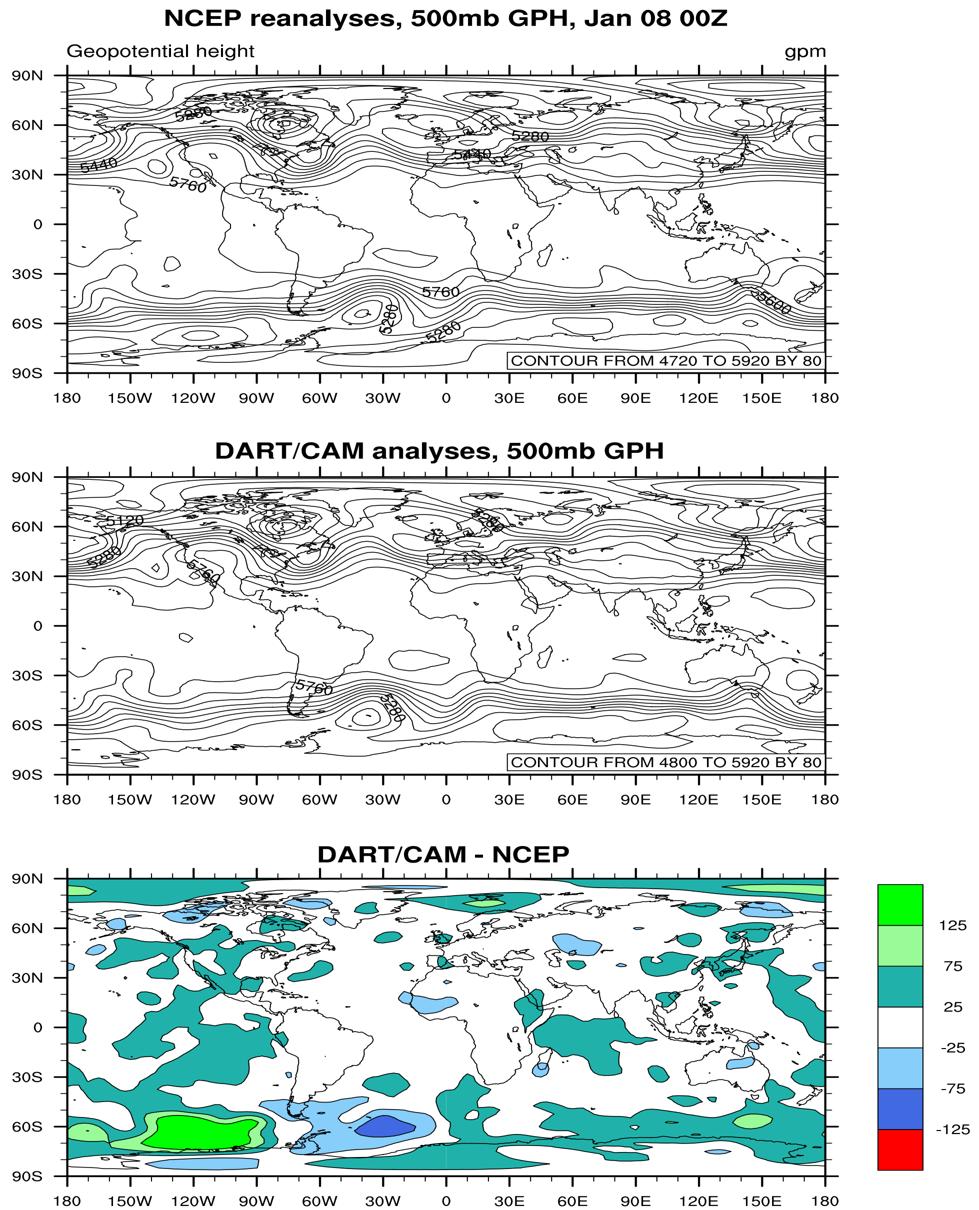

NCEP reanalyses, 500mb GPH, Jan 01 06Z
Geopotential height
gpm
CONTOUR FROM 4800 TO 5920 BY 80
DART/CAM analyses, 500mb GPH
CONTOUR FROM 5120 TO 5840 BY 80
DART/CAM - NCEP
125
75
25
-25
-75
-125


NCEP reanalyses, 500mb GPH, Jan 02 00Z
Geopotential height
gpm
CONTOUR FROM 4800 TO 5920 BY 80

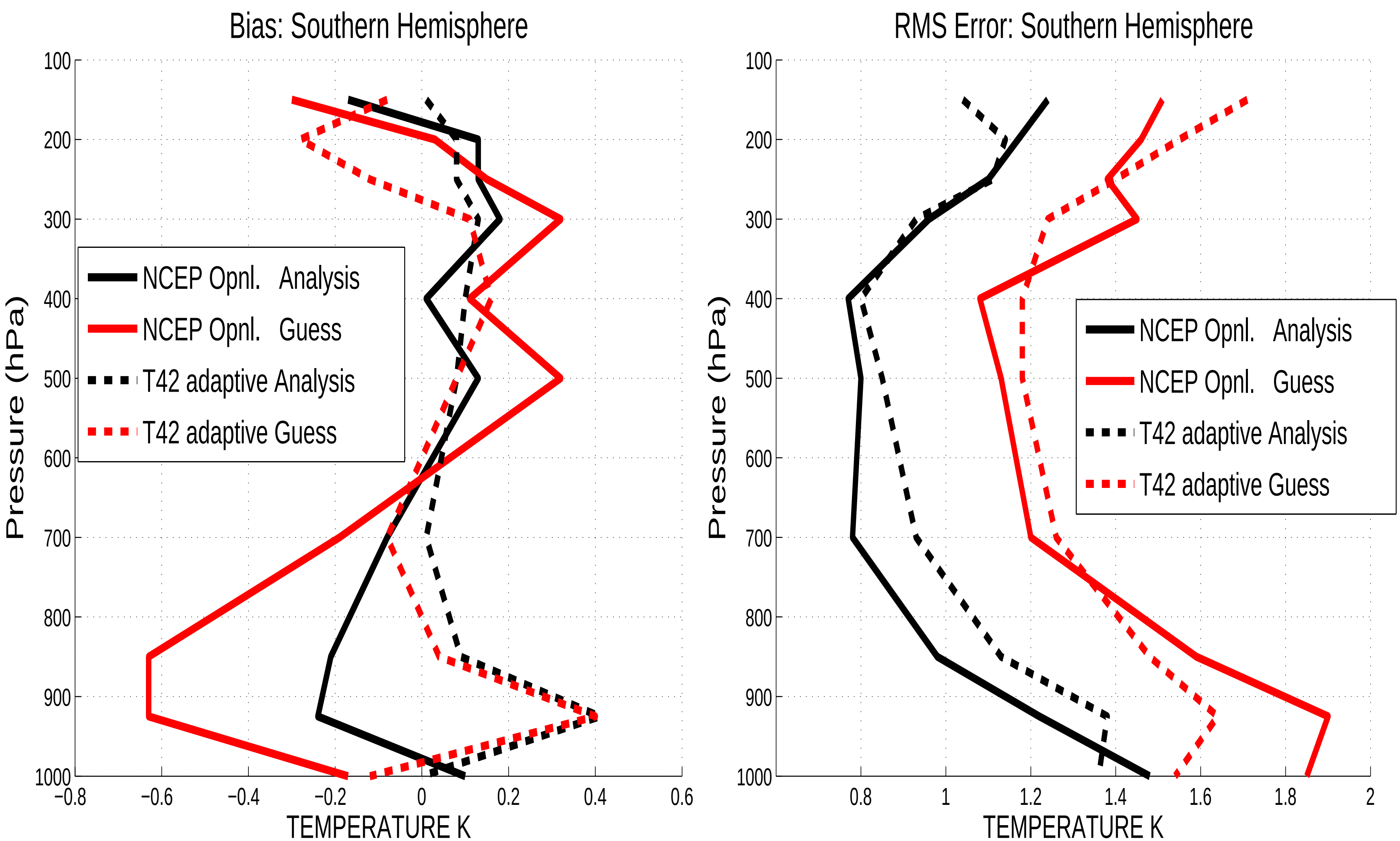

DART/CAM analyses, 500mb GPH
CONTOUR FROM 4960 TO 5920 BY 80


DART/CAM - NCEP
125
75
25
-25
-75
-125


NCEP reanalyses, 500mb GPH, Jan 04 00Z
Geopotential height
gpm
CONTOUR FROM 4880 TO 5920 BY 80
DART/CAM analyses, 500mb GPH
CONTOUR FROM 4880 TO 5920 BY 80
DART/CAM - NCEP
125
75
25
-25
-75
-125


NCEP reanalyses, 500mb GPH, Jan 08 00Z
Geopotential height
gpm
CONTOUR FROM 4720 TO 5920 BY 80
DART/CAM analyses, 500mb GPH
CONTOUR FROM 4800 TO 5920 BY 80
DART/CAM - NCEP
125
75
25
-25
-75
-125

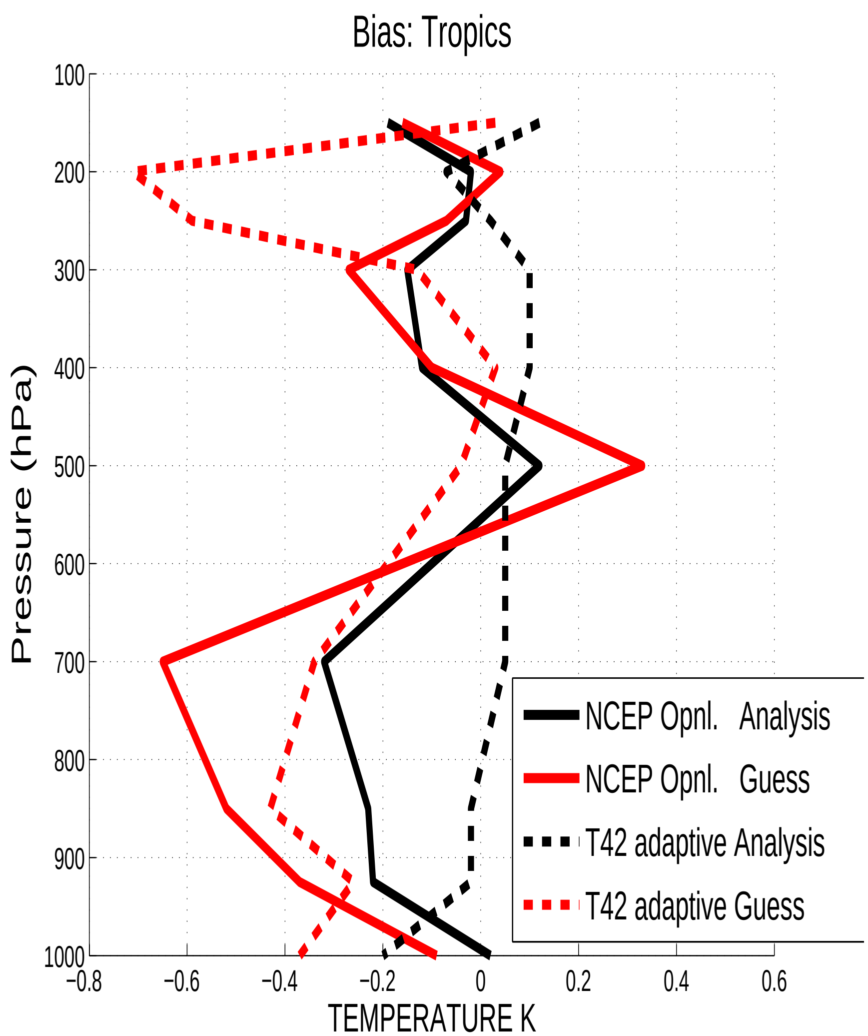
Southern Hemisphere Temperature: Bias and RMSE



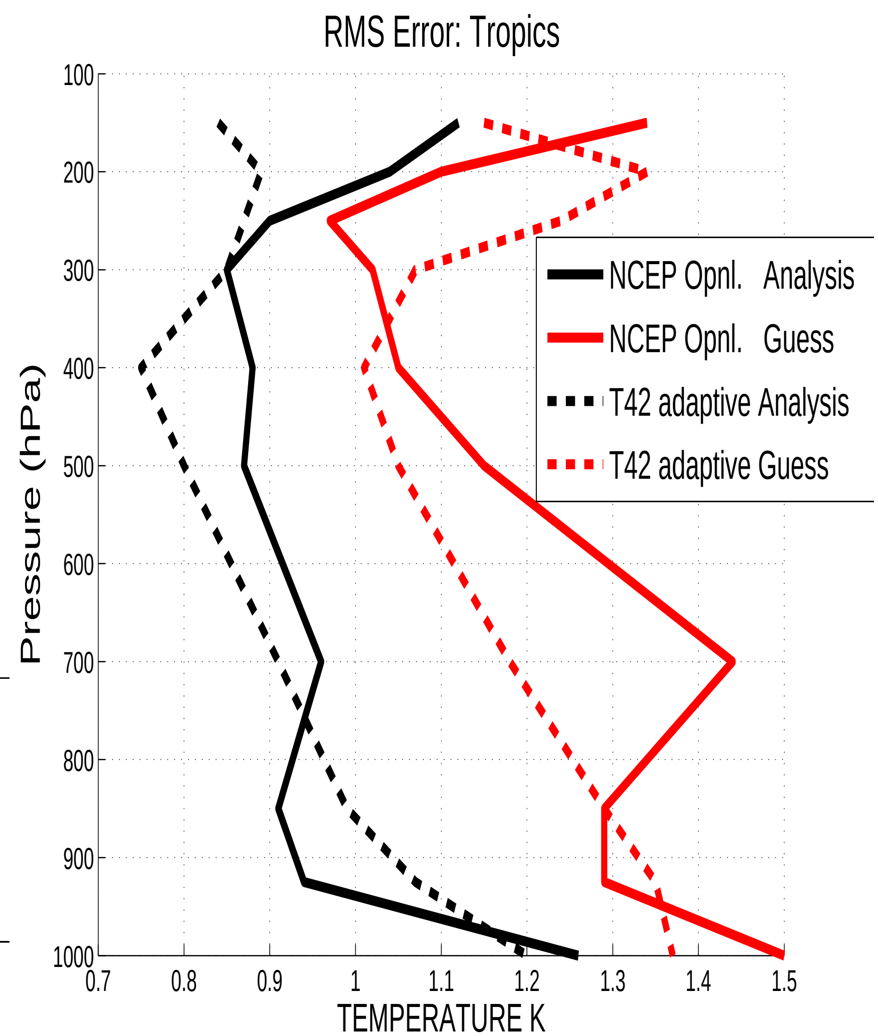
Bias

RMS

Tropics Temperature: Bias and RMSE

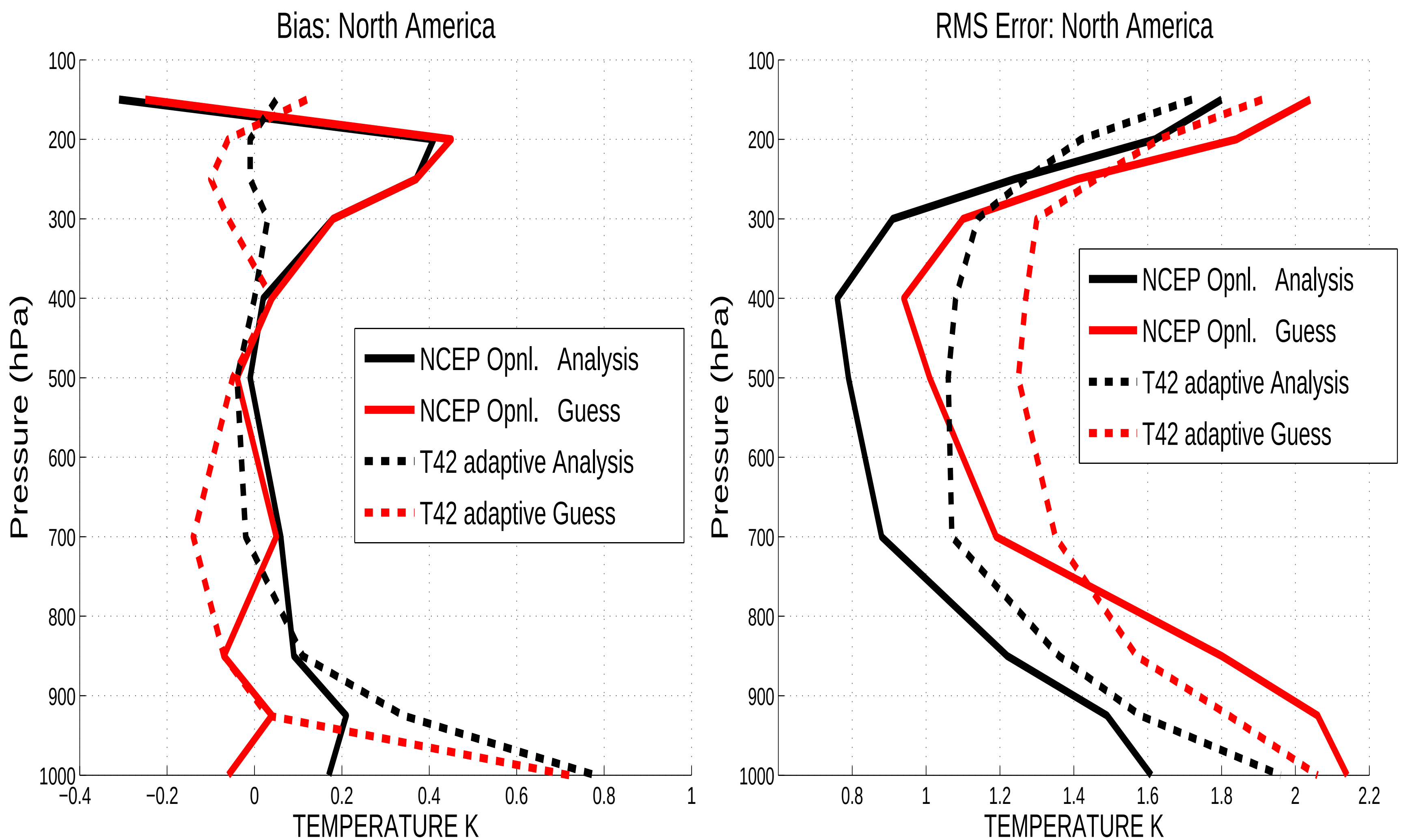


Bias



RMS

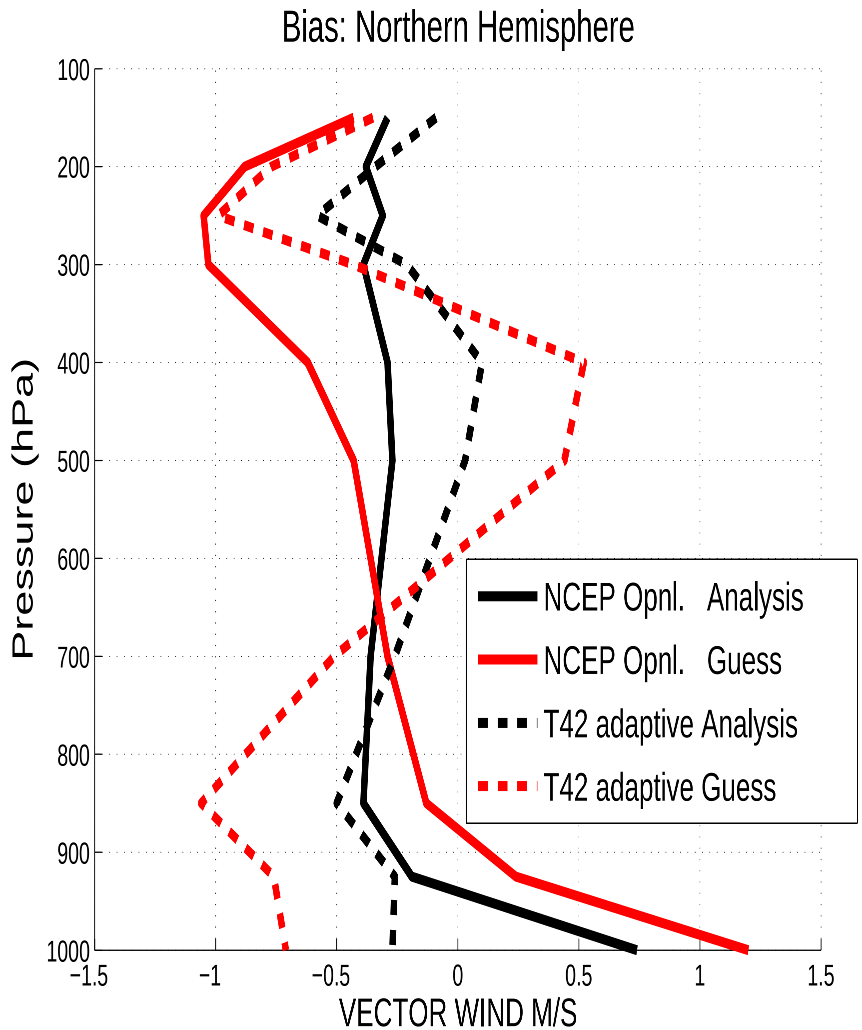
North America Temperature: Bias and RMSE

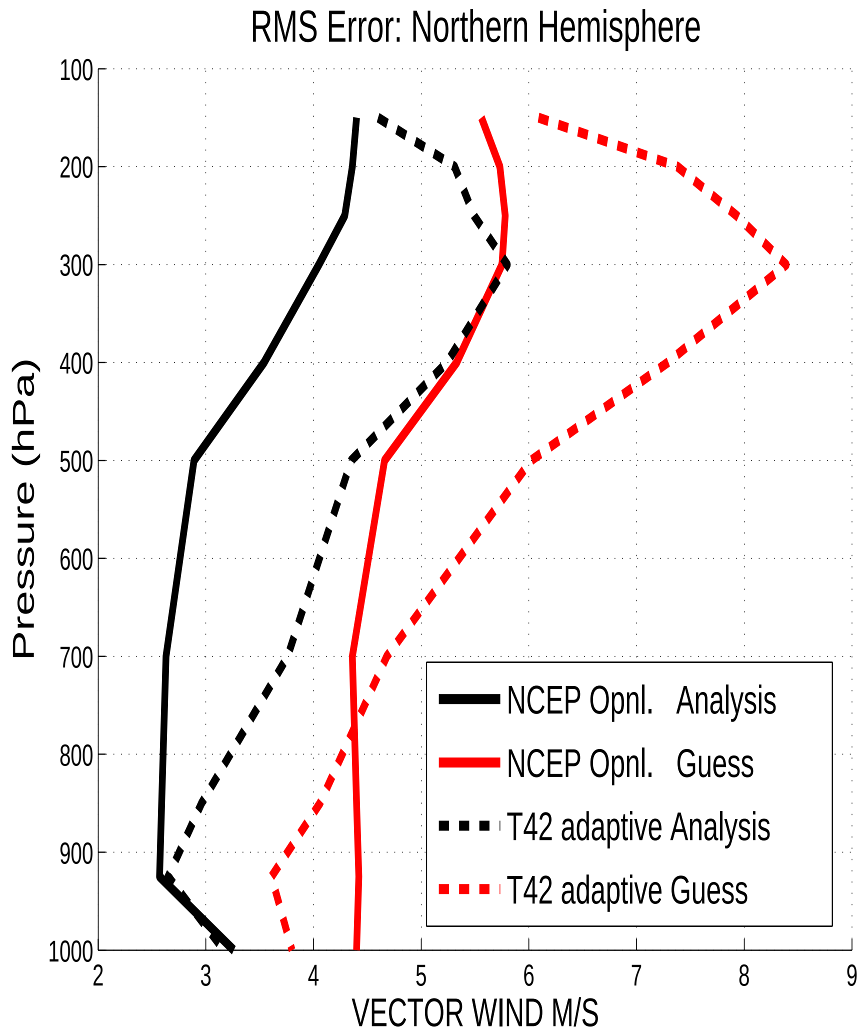


Bias

RMS

Northern Hemisphere Wind: Bias and RMSE

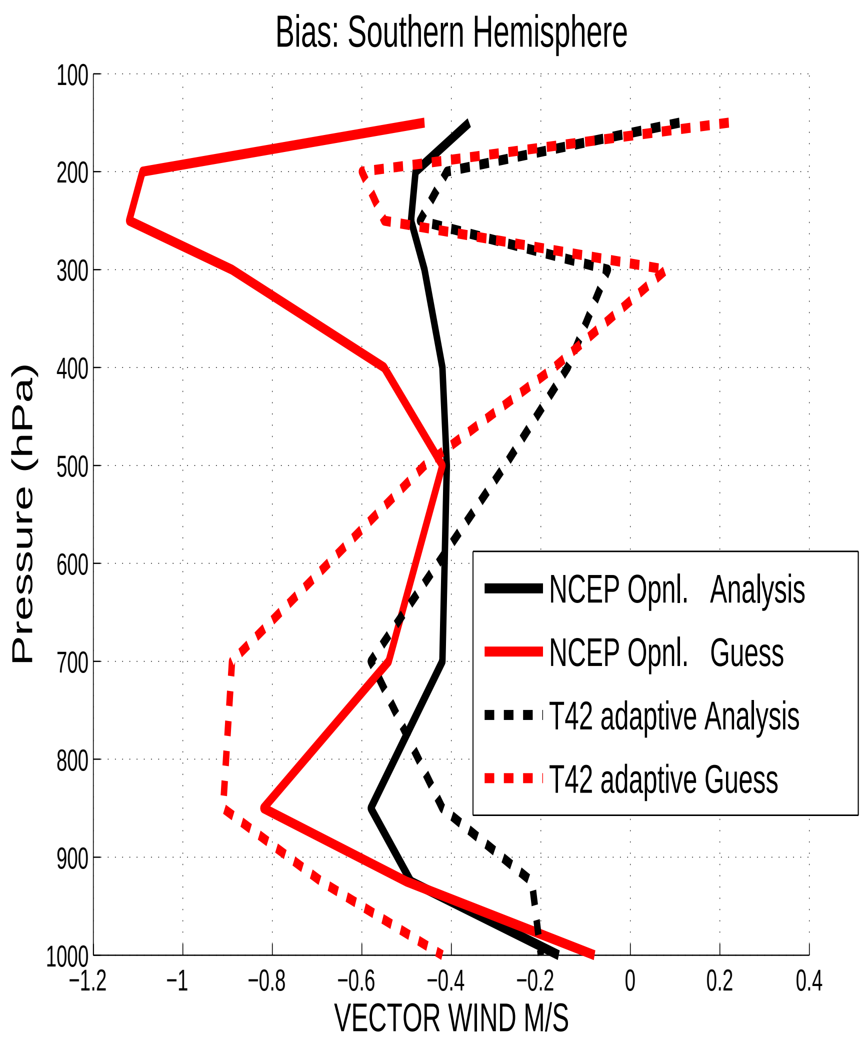




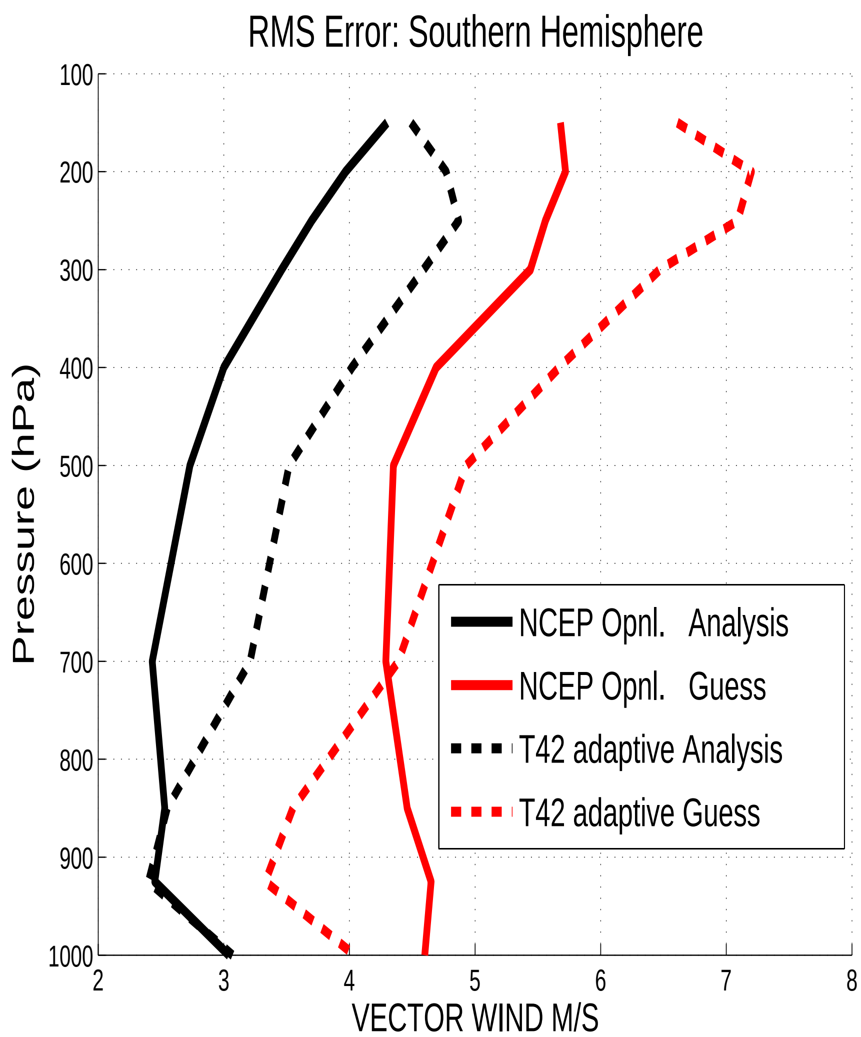
<u>Bias</u>

<u>**RMS**</u>

Southern Hemisphere Wind: Bias and RMSE

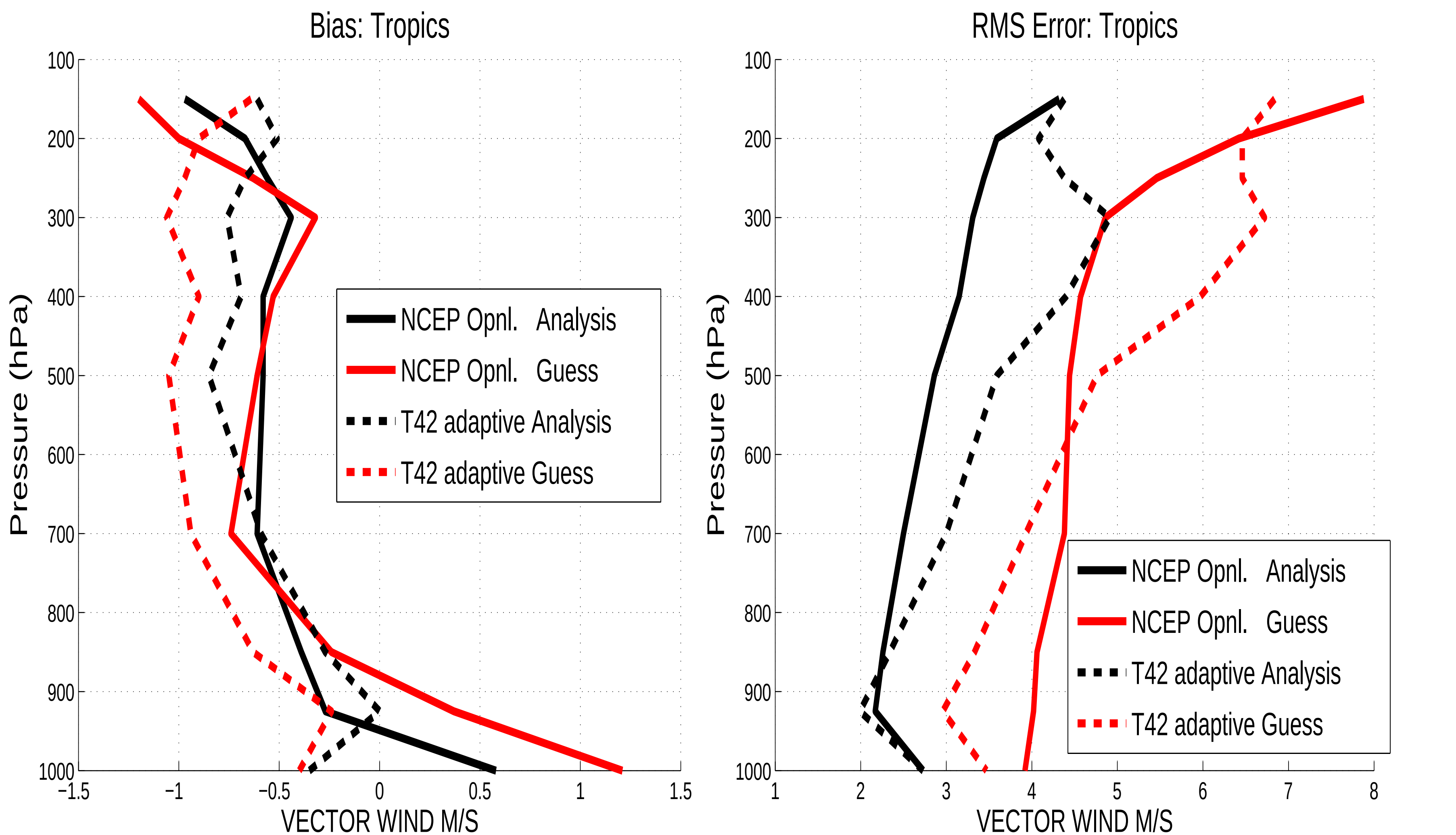


Bias



RMS

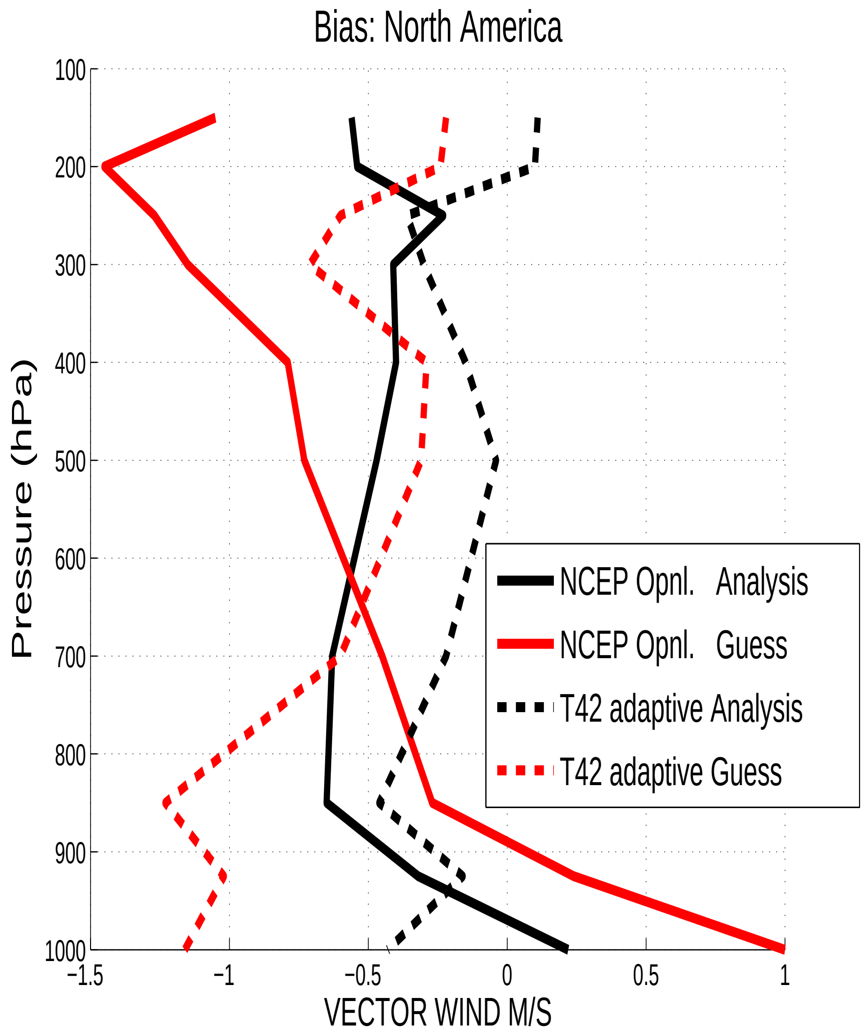
Tropical Wind: Bias and RMSE

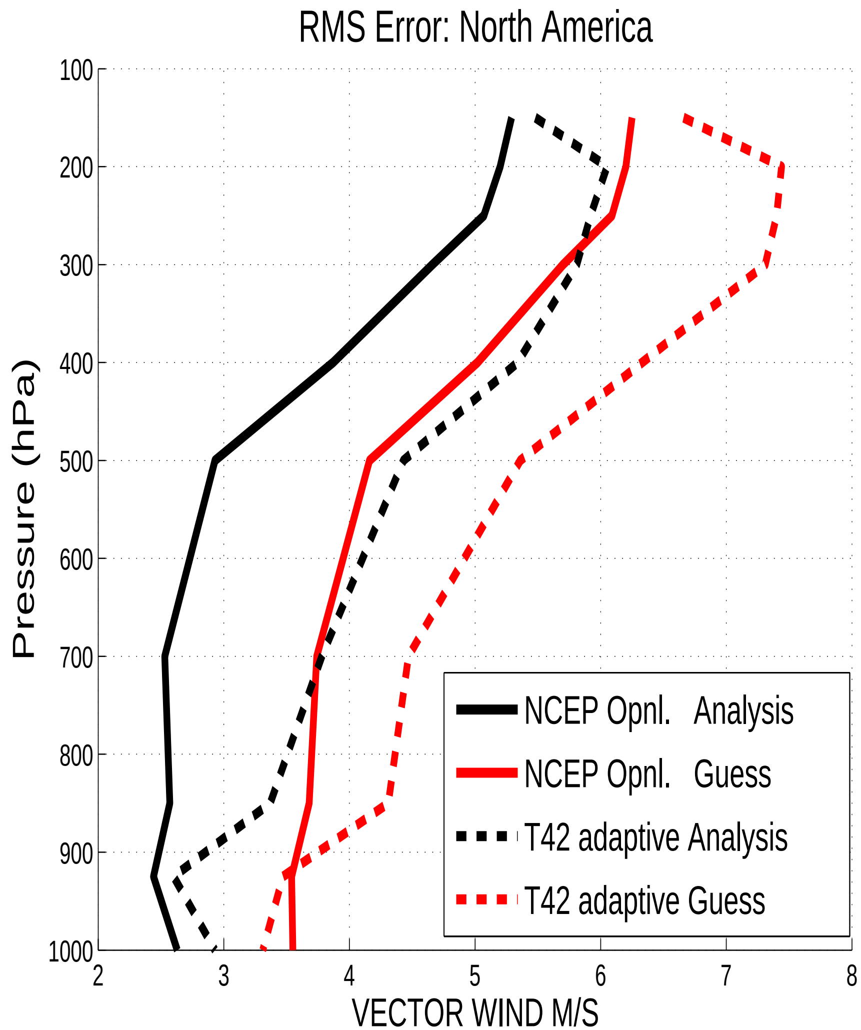


Bias

RMS

Northern America Wind: Bias and RMSE





Bias

RMS

Conclusions

1. It is easy to incorporate a model in an ensemble filter.

2. Naive use of ensemble filters cannot compete with variational.

3. Algorithms to deal with variance loss are available.

4. Algorithms to deal with systematic error under development.

5. Hard part remaining is dealing with nasty, dirty observational data.

6. Ensemble algorithms are now nearly competitive with variational.

7. Don't believe any comparative results that aren't from same system.
    QC, plotting details, who knows what else come into play.

Data Assimilation Research Testbed (DART)

Software to do everything here (and more) is in DART.

Requires F90 compiler, Matlab.

Available from www.image.ucar.edu/DAI/DART/.

DART compliant models (largest set ever with assim system?)

1. Many low-order models (Lorenz63, L84, L96, L2004,...)
2. Global 2-level PE model (from NOAA/CDC)
3. CGD’s CAM 2.0 & 3.0 (global spectral model)
4. GFDL FMS B-grid GCM (global grid point model)
5. MIT GCM (from Jim Hansen; configured for annulus)
6. WRF model
7. NCEP GFS (assisted by NOAA/CDC)
8. GFDL MOM3/4 ocean model
9. ACD’s ROSE model (upper atmosphere with chemistry)

This allows for a hierarchical approach to filter development.

DART compliant Forward Operators and Datasets

Many linear and non-linear forward operators for low-order models.

U, V, T, Ps, Q, for realistic models.

Radar reflectivity, GPS refractivity for realistic models.

Can ingest observations from reanalysis or operational BUFR files.

Can create synthetic (perfect model) observations for any of these.