Ensemble Filters for Geophysical Data Assimilation: A Tutorial

Jeffrey Anderson
NCAR Data Assimilation Initiative

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

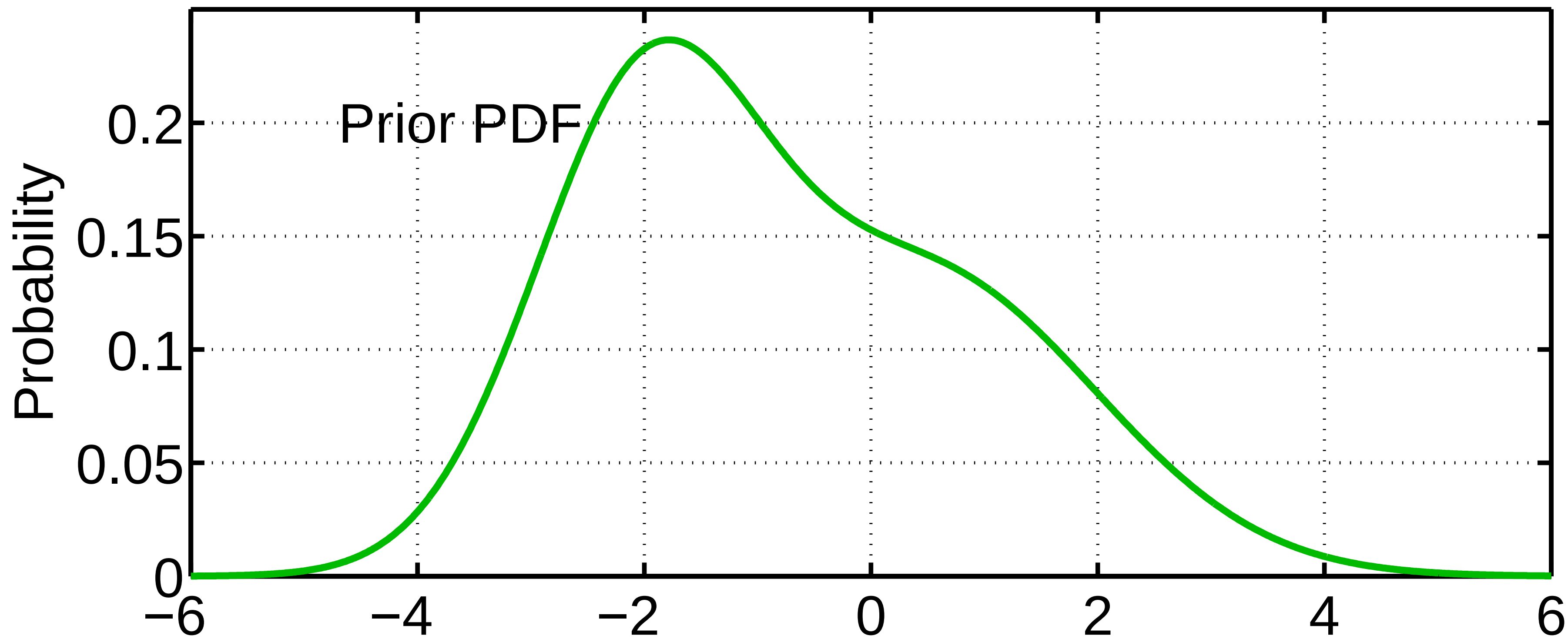
Phase 1: Single variable and observation of that variable.

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

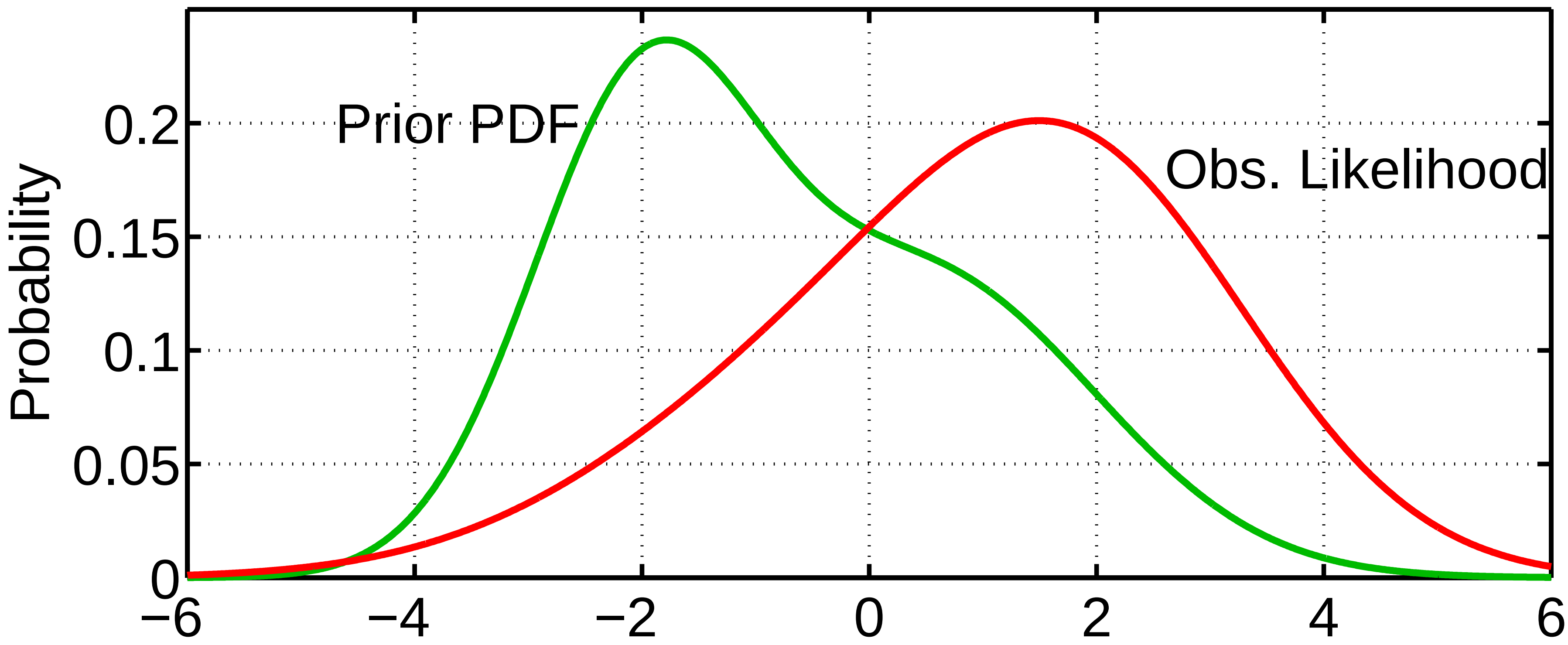


A: Prior estimate based on all previous information, C.

B: An additional observation.

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

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

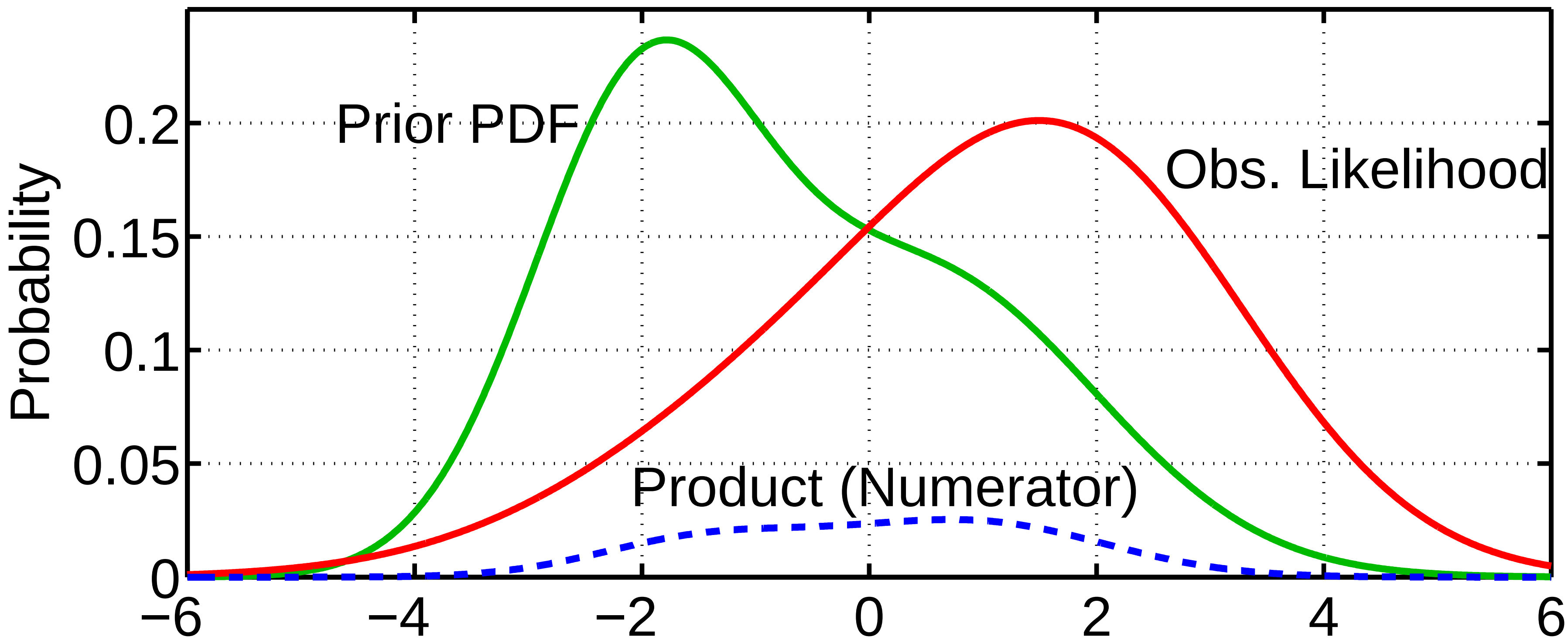


A: Prior estimate based on all previous information, C.

B: An additional observation.

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

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

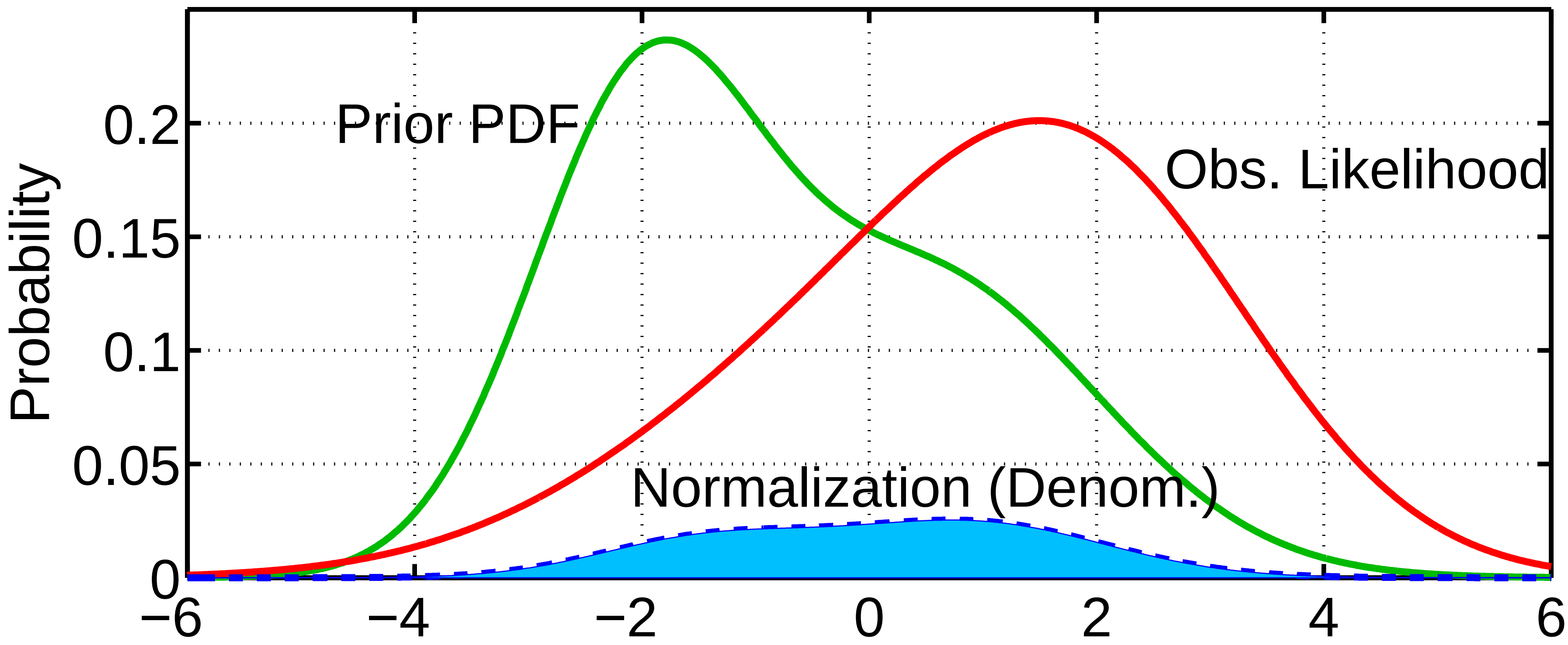


A: Prior estimate based on all previous information, C.

B: An additional observation.

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

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

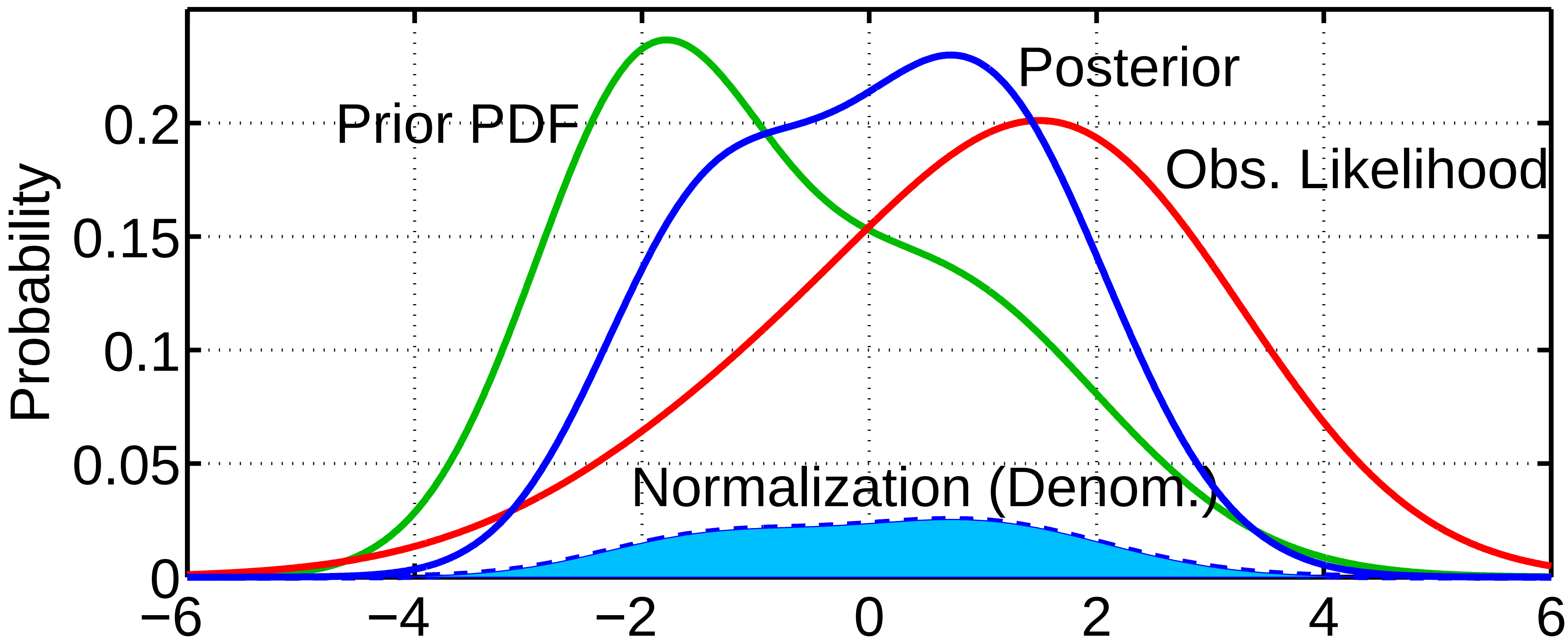


A: Prior estimate based on all previous information, C.

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

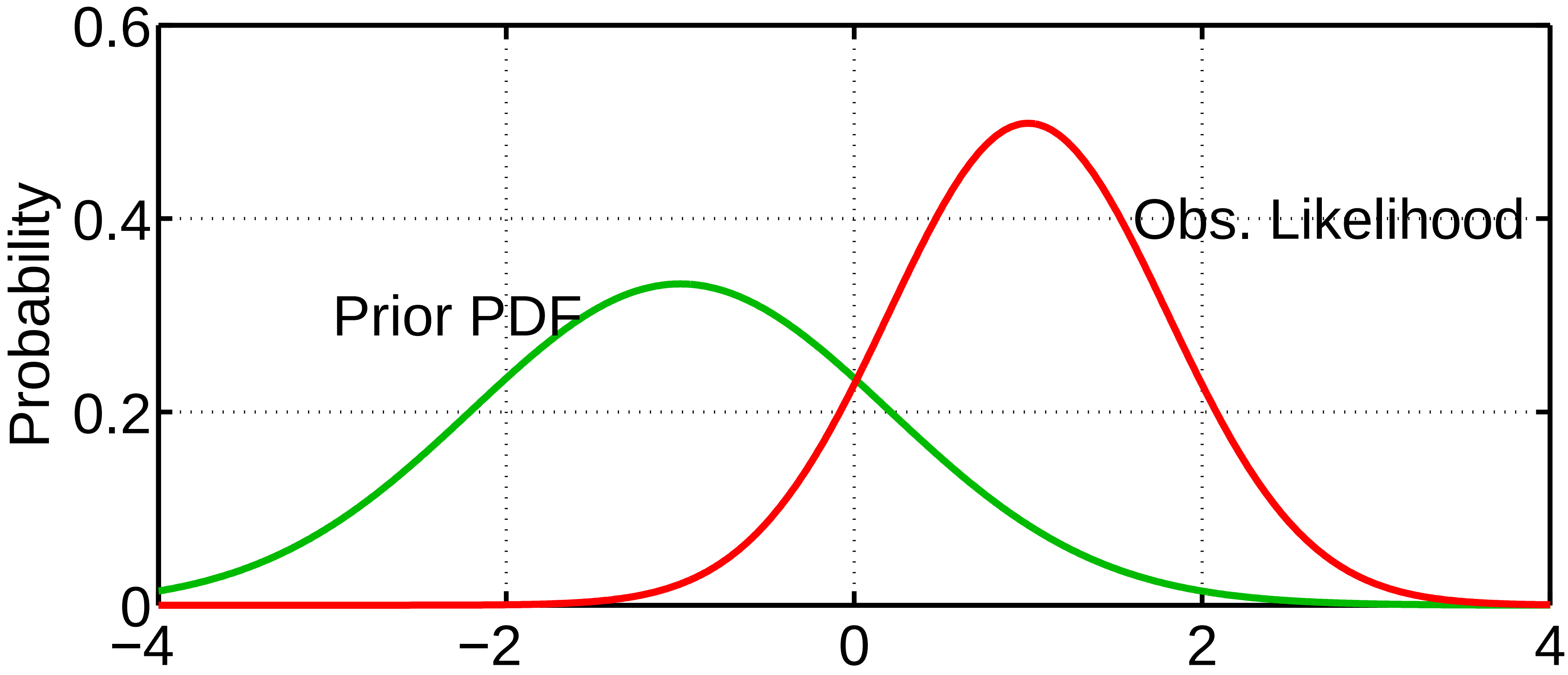
Green = Prior

Red = Observation

Blue = Posterior

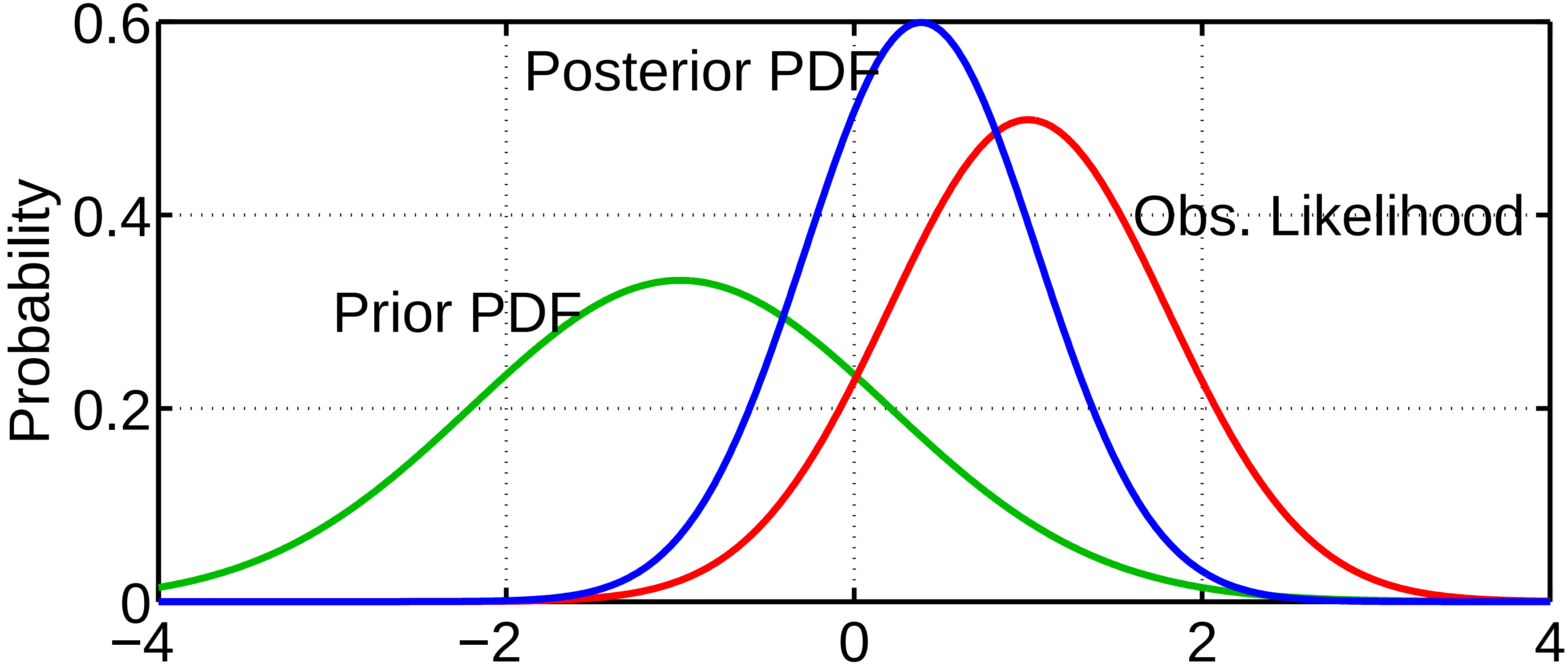
Bayes rule: $p(A|BC) = \dfrac{p(B|AC)p(A|C)}{p(B|C)} = \dfrac{p(B|AC)p(A|C)}{\int p(B|x)p(x|C)dx}$

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

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

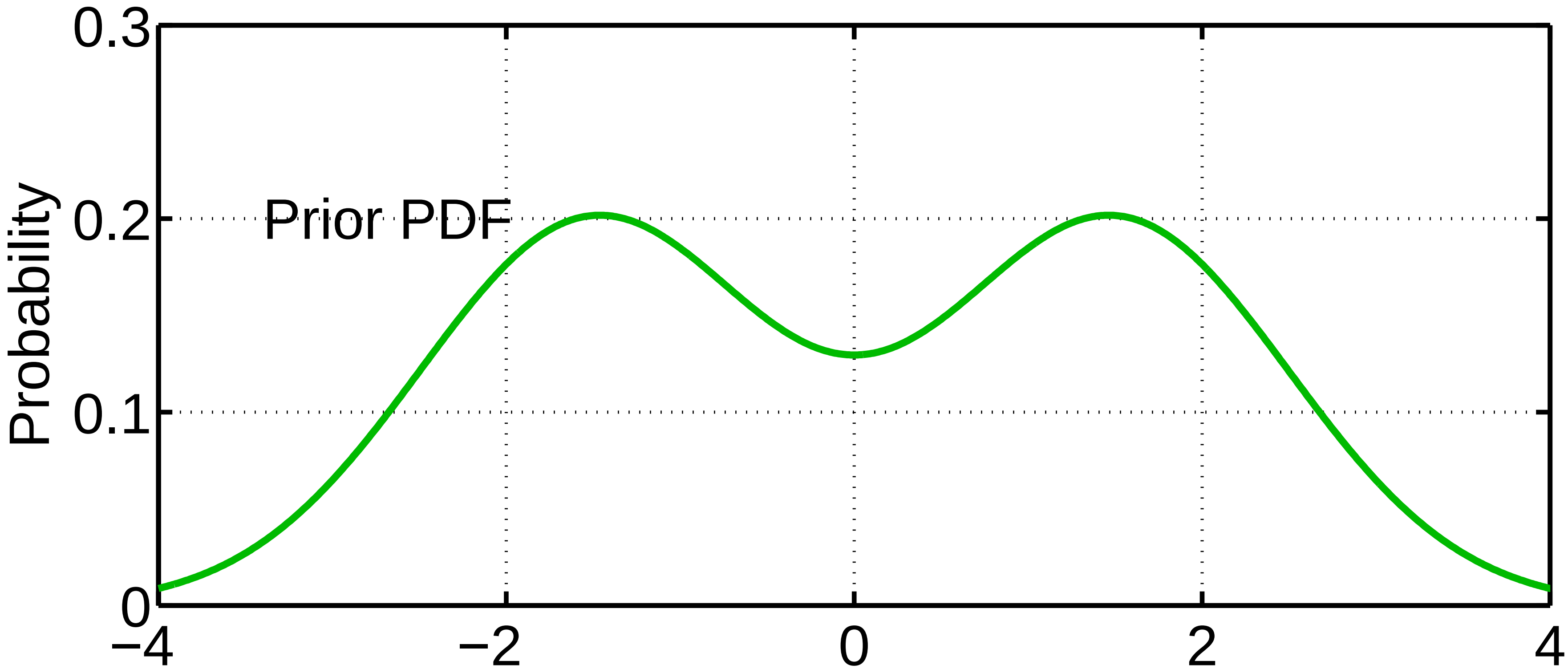
Mean: $\mu = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}(\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2)$

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

This product is closed for Gaussian distributions.

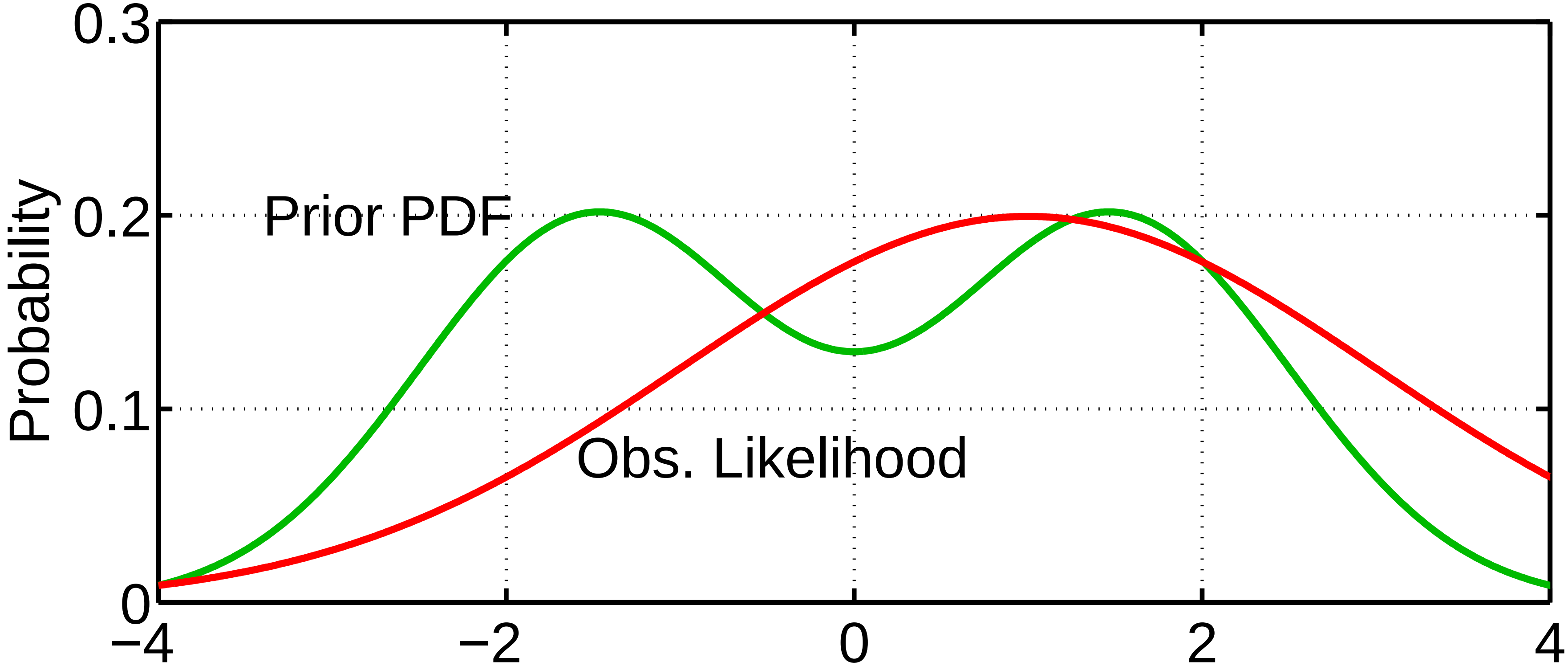


There are other families of functions for which it is closed...

But, for general distributions, there's no analytical product.

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.

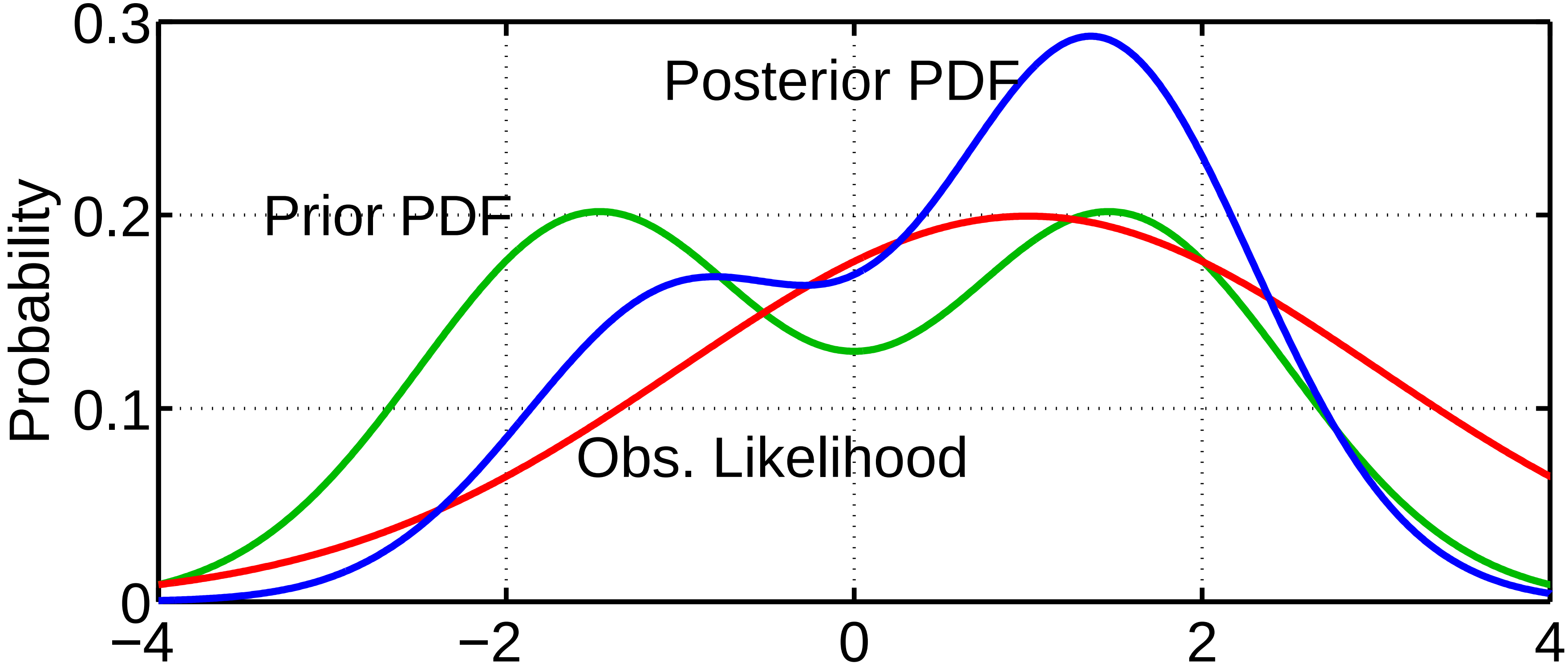


There are other families of functions for which it is closed...

But, for general distributions, there's no analytical product.

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.

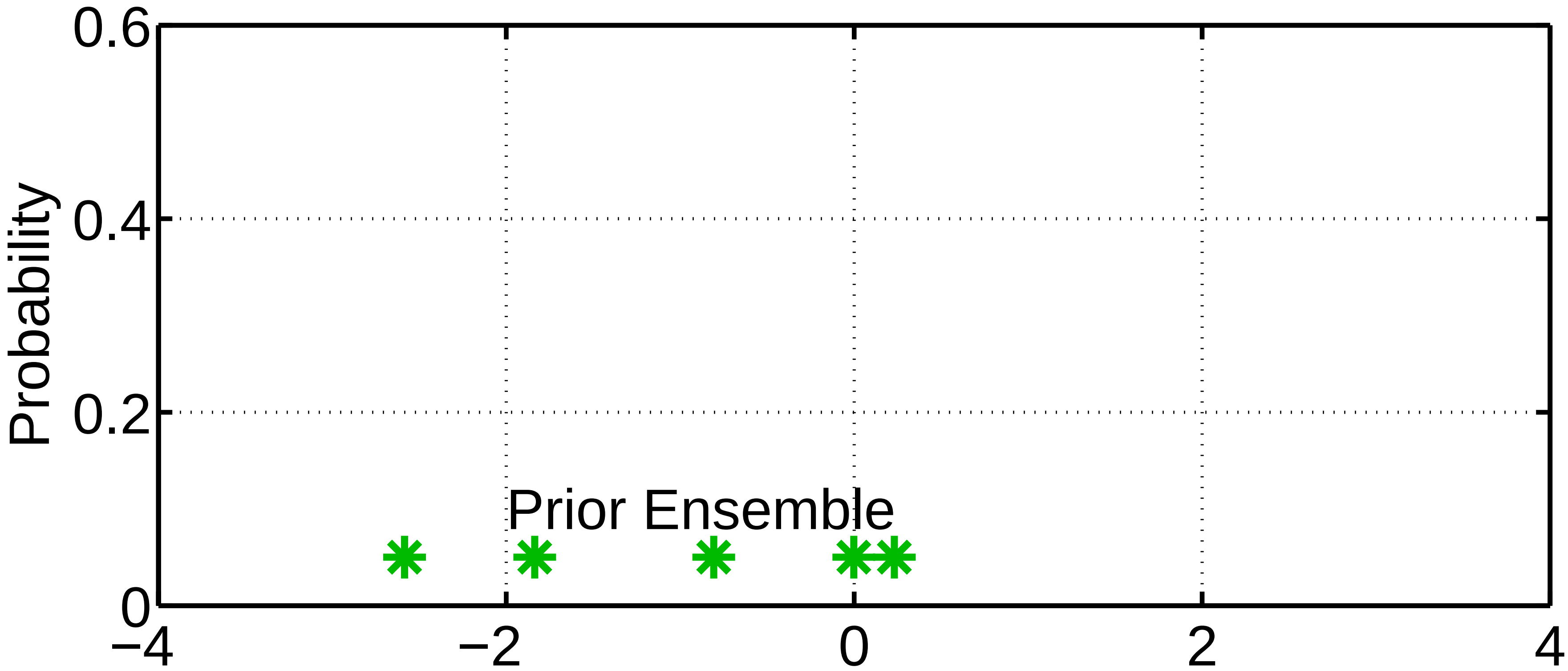


There are other families of functions for which it is closed...

But, for general distributions, there’s no analytical product.

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

Ensemble filters: Prior is available as finite sample.

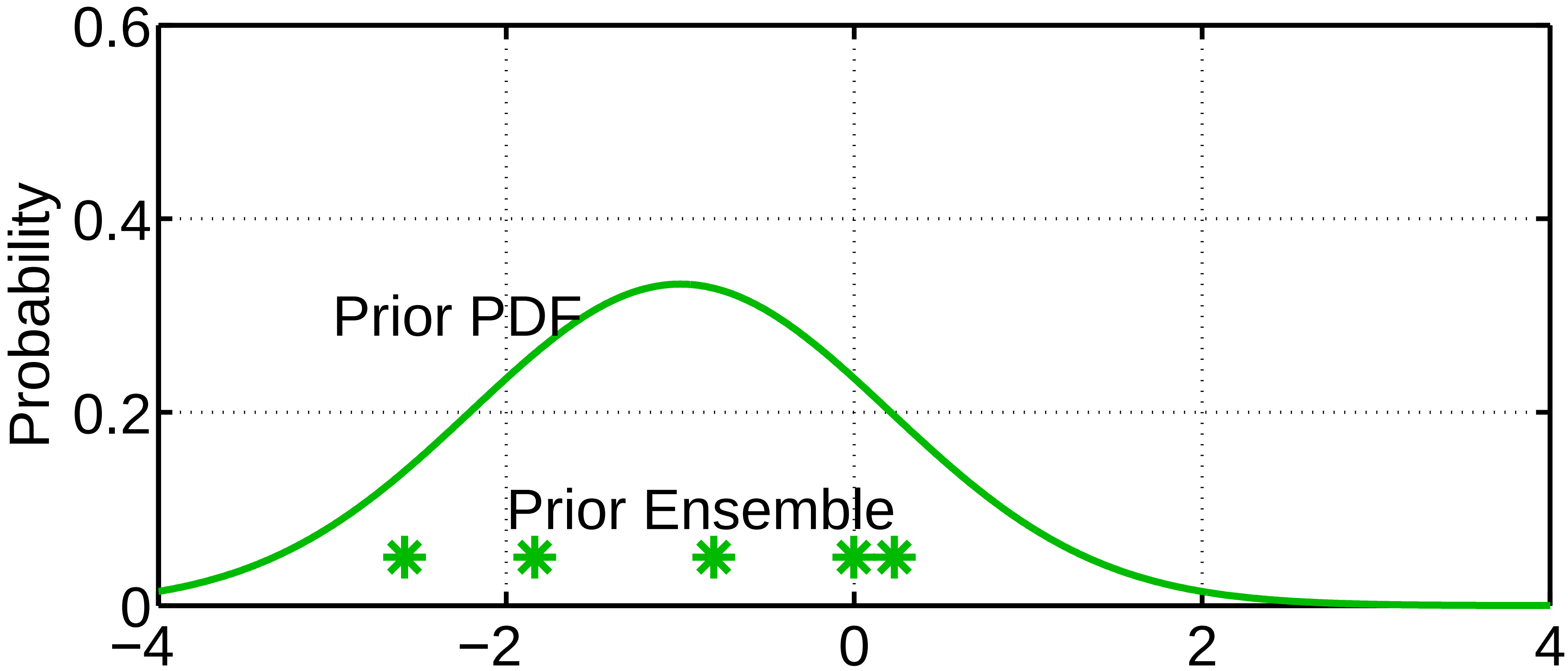


Don't know much about properties of this sample.

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

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

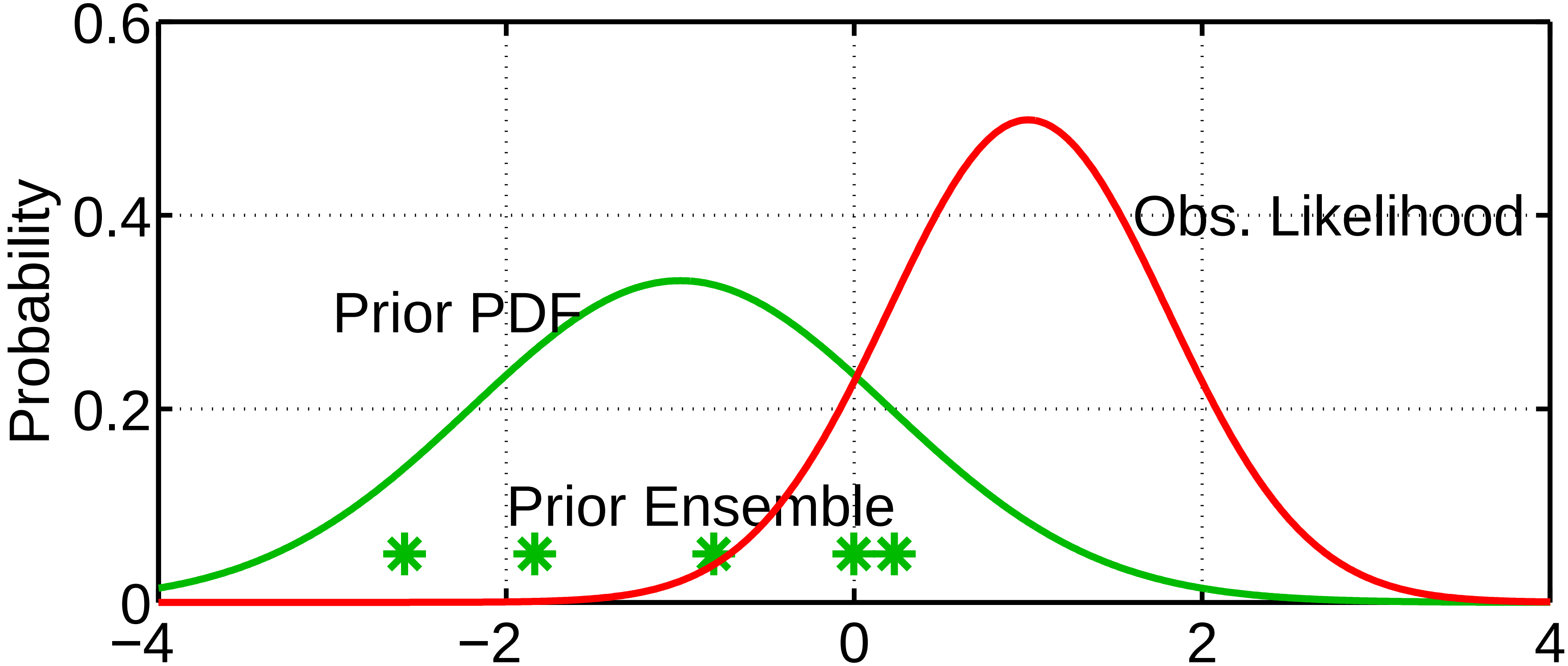
How can we take product of sample with continuous likelihood?



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

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

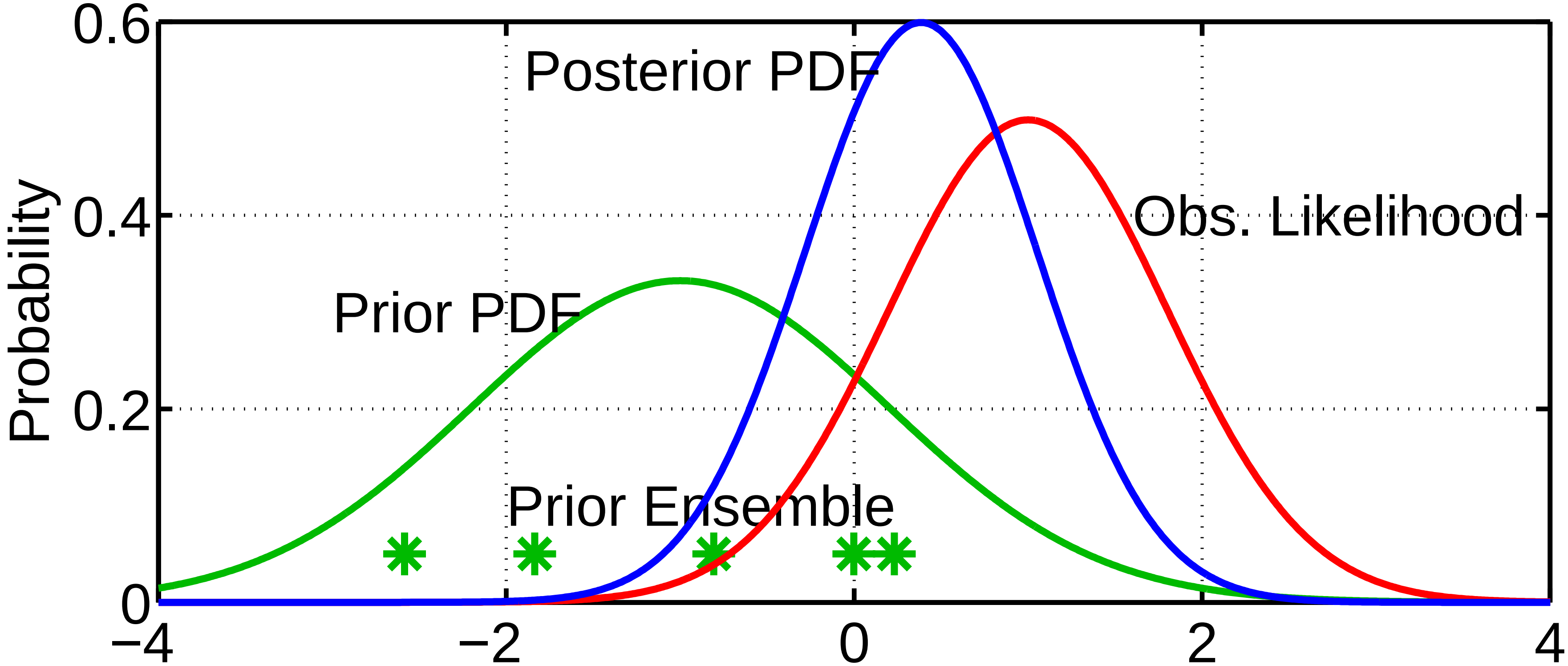
Observation likelihood usually continuous (nearly always Gaussian).



If Obs. Likelihood isn’t Gaussian, can generalize methods below.
For instance, can fit set of Gaussian kernels to obs. likelihood.

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

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

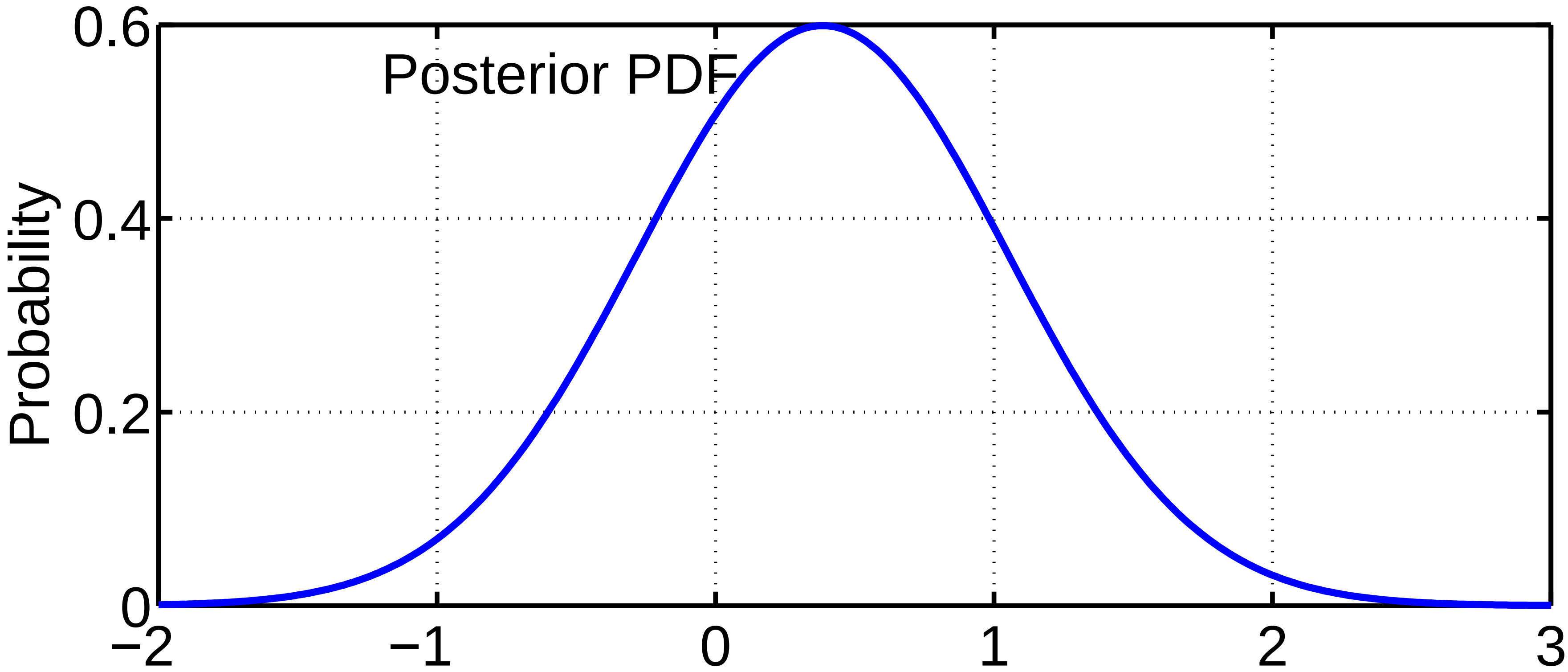


Computing continuous posterior is simple.

BUT, need to have a SAMPLE of this PDF.

Sampling Posterior PDF:

There are many ways to do this.



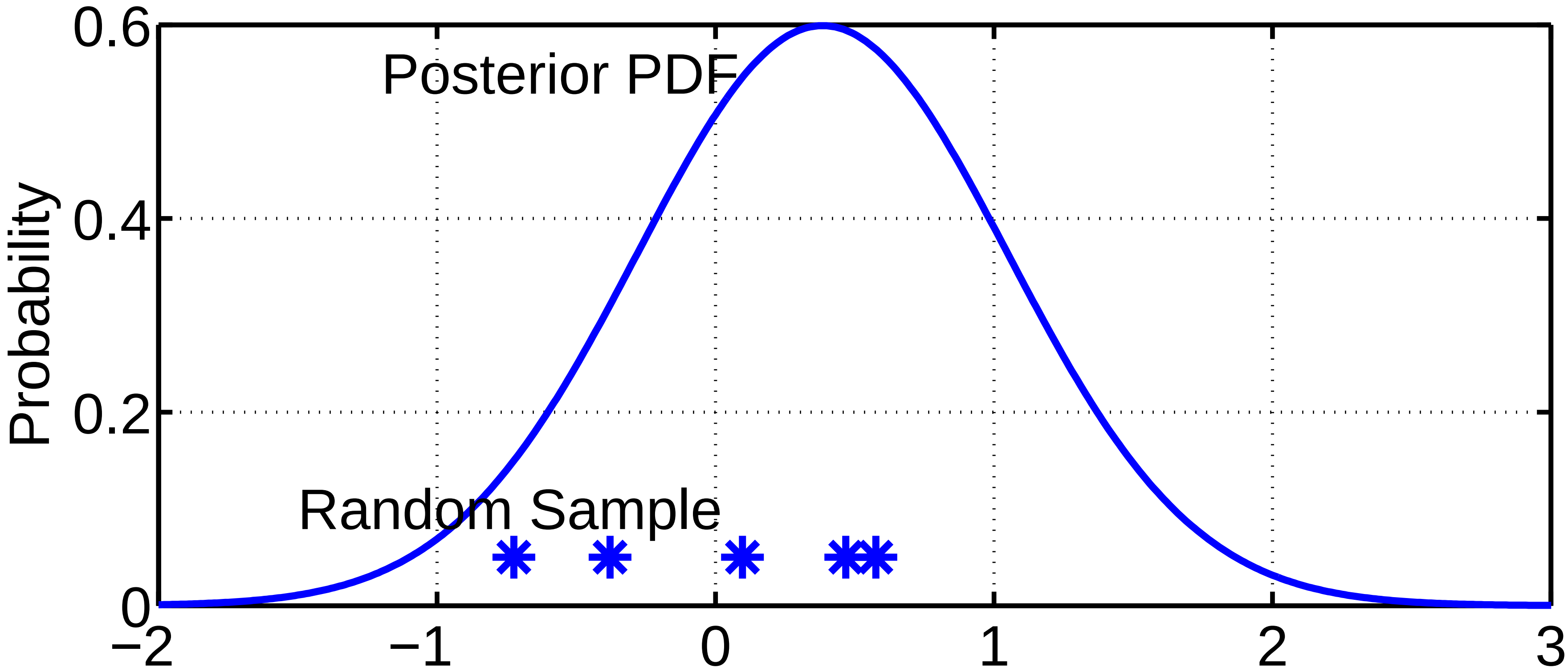
Exact properties of different methods may be unclear.

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

Will interact with properties of prediction models, etc.

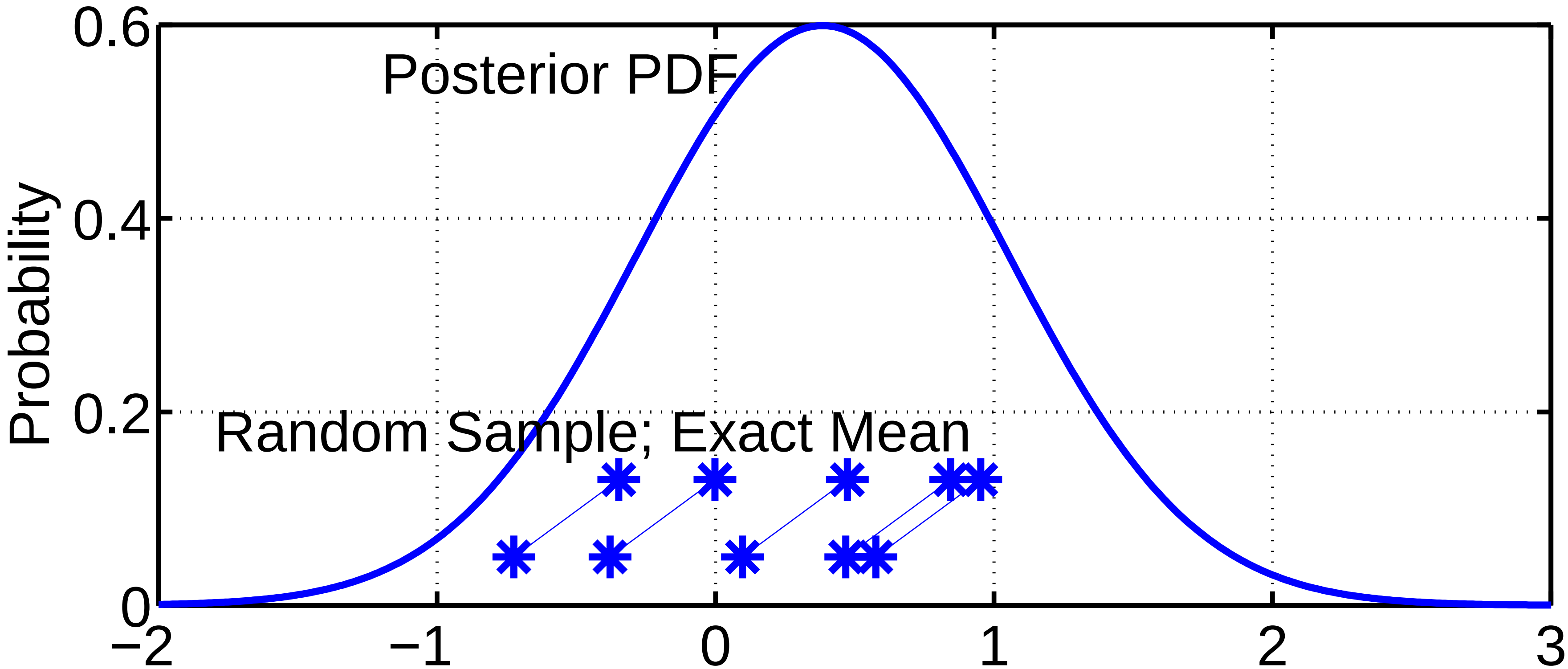
Sampling Posterior PDF:

1. Just draw a random sample.



Sampling Posterior PDF:

1. Just draw a random sample.

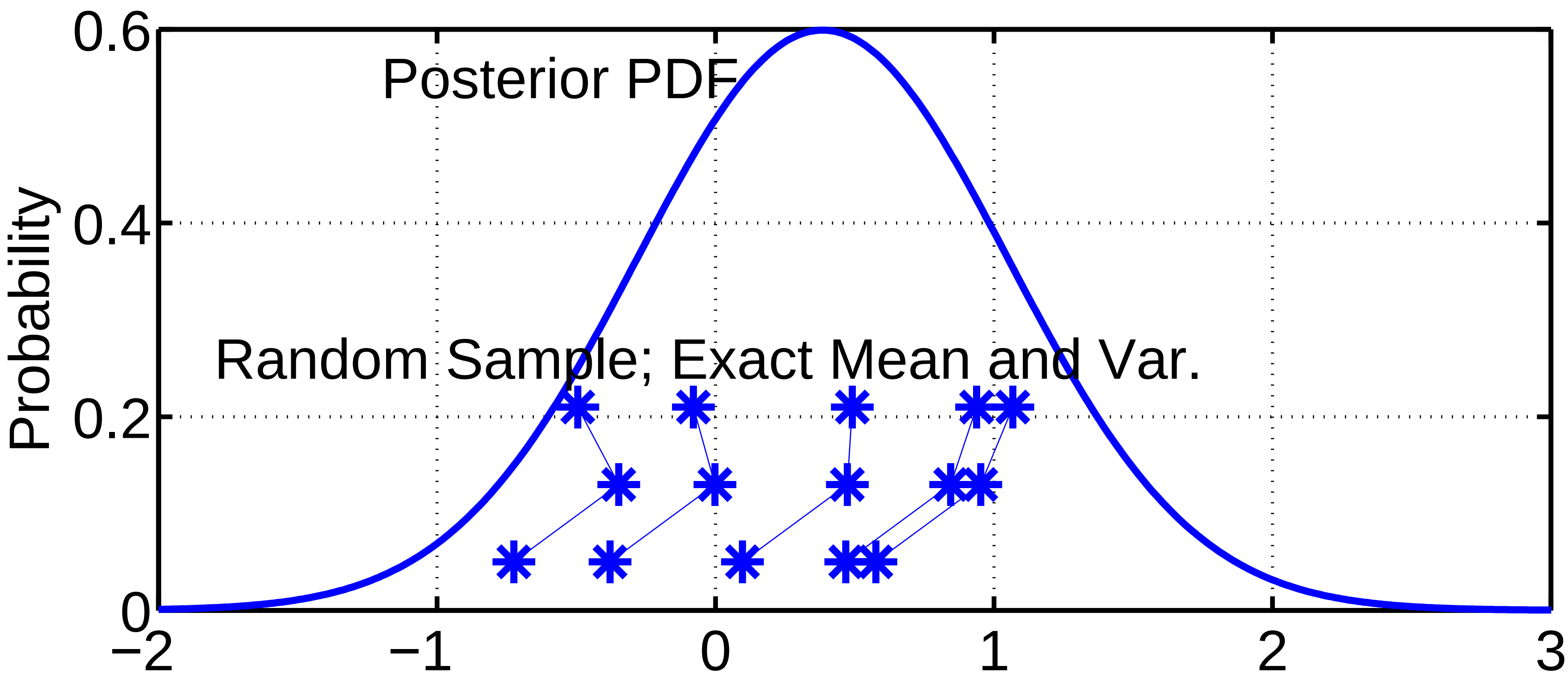


Can ‘play games’ with this sample to improve (modify) its properties.

Example: Adjust the mean of sample to be exact.

Sampling Posterior PDF:

1. Just draw a random sample.

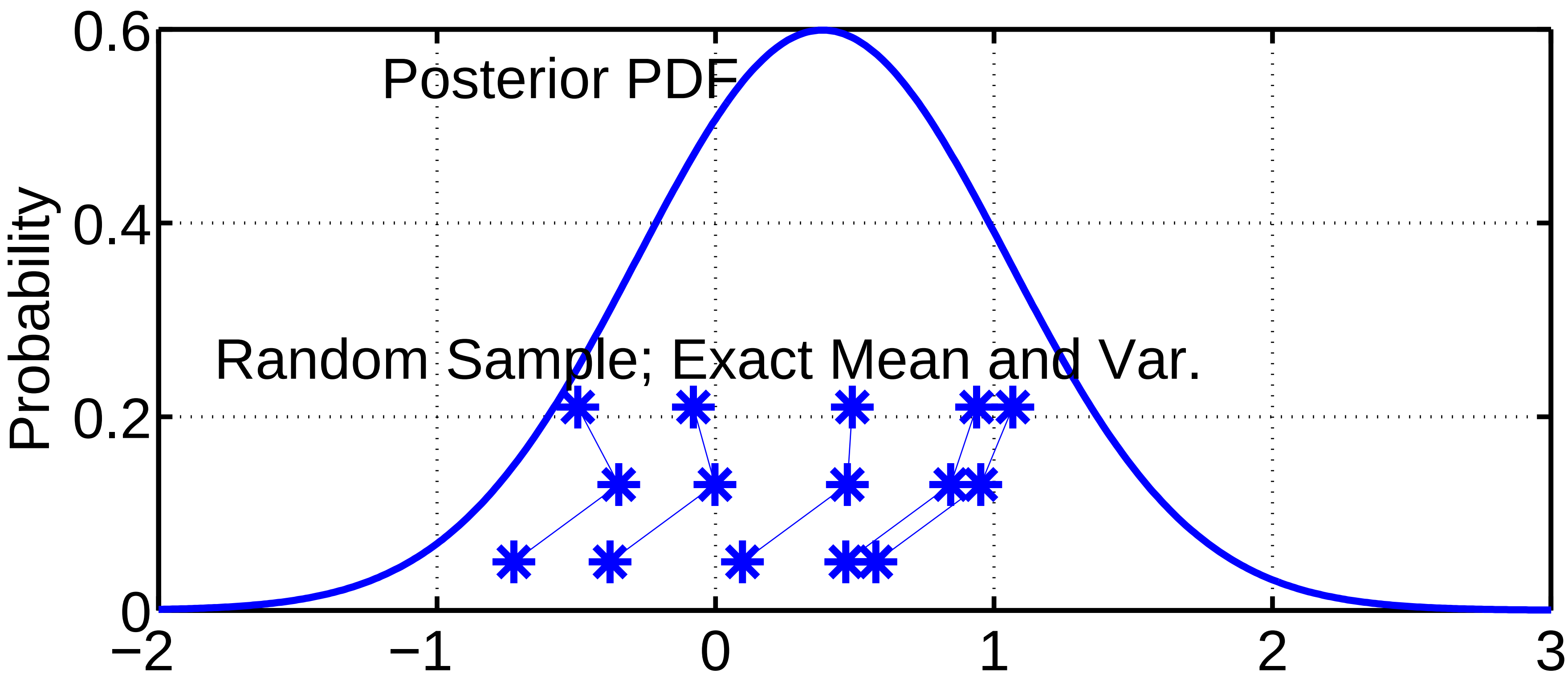


Can ‘play games’ with this sample to improve (modify) its properties.

Example: Adjust the mean of sample to be exact.
Can also adjust the variance to be exact.

Sampling Posterior PDF:

1. Just draw a random sample.

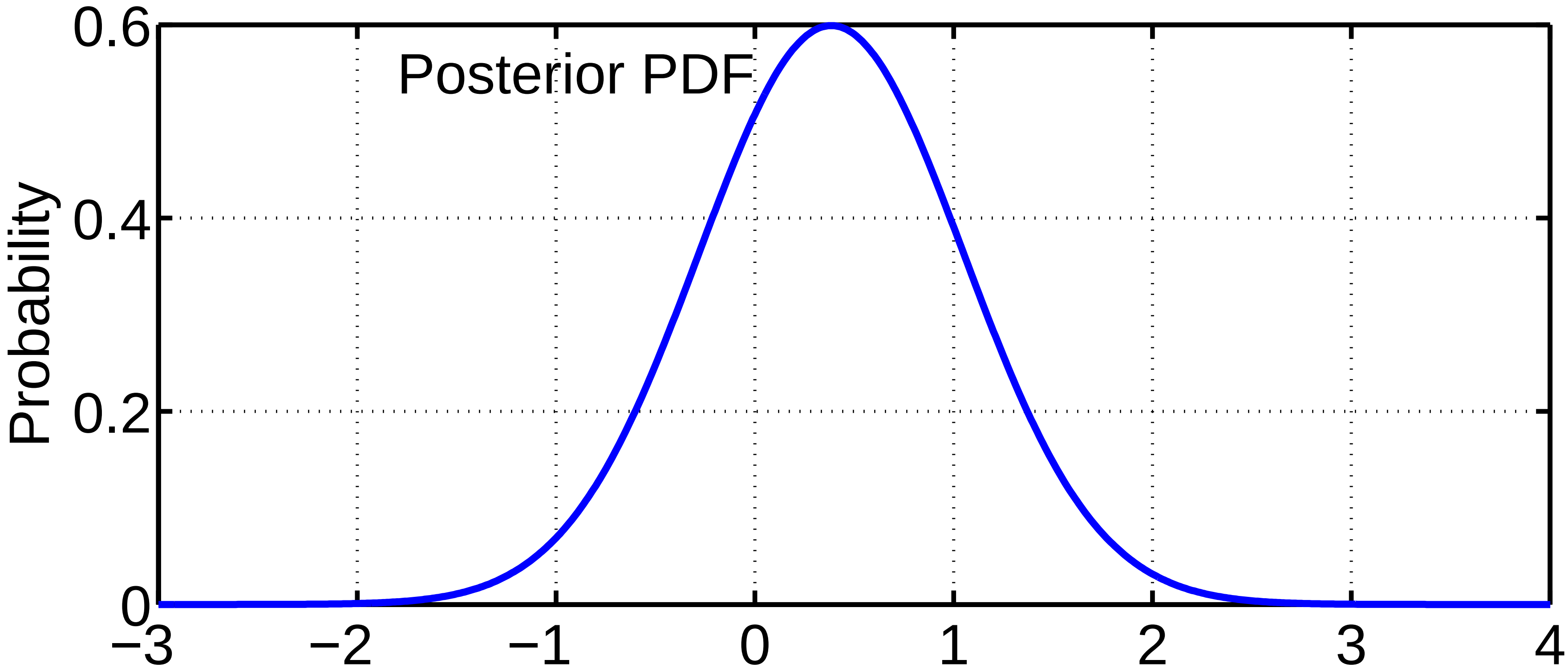


Might also want to eliminate rare extreme outliers.

NOTE: Properties of these adjusted samples can be quite different.
How these properties interact with rest of assimilation is open question.

Sampling Posterior PDF:

2. Construct a 'deterministic' sample with certain features.

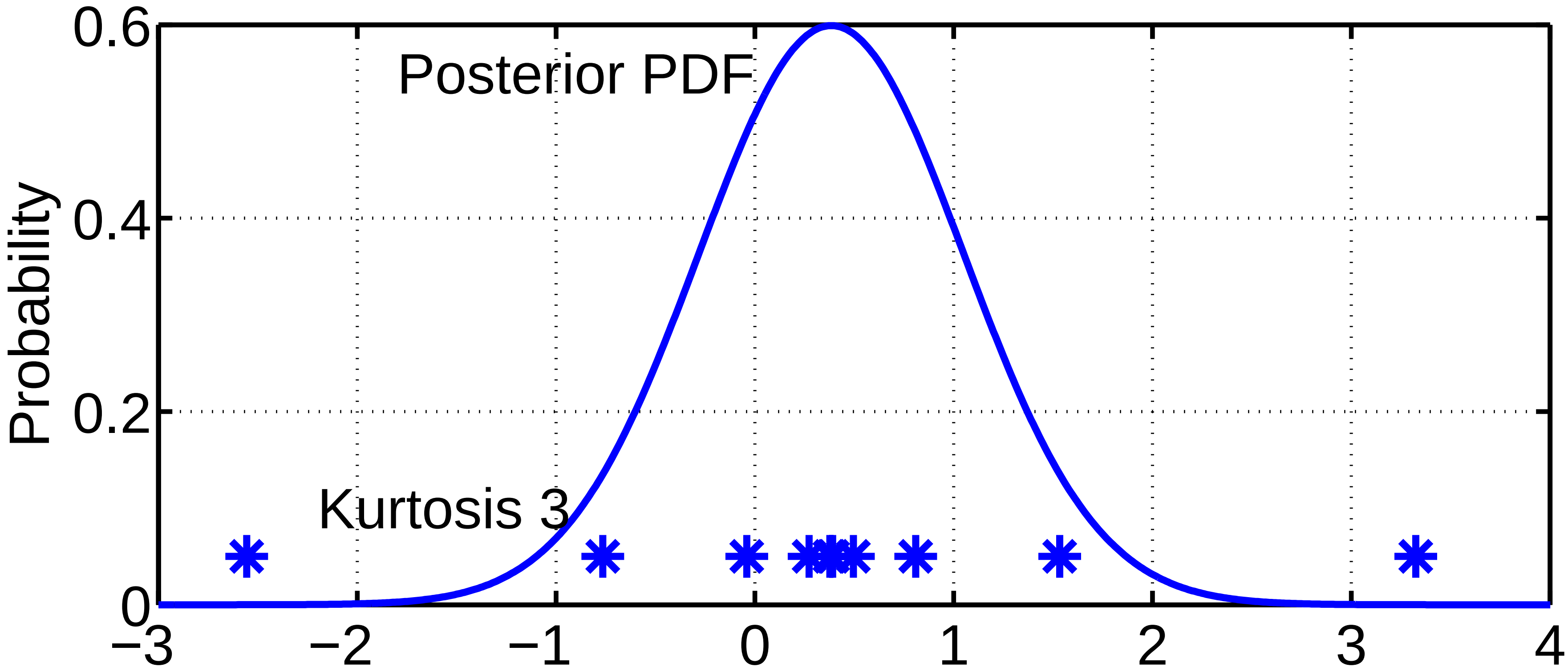


For instance: Sample could have exact mean and variance.

This is insufficient to constrain ensemble, need other constraints.

Sampling Posterior PDF:

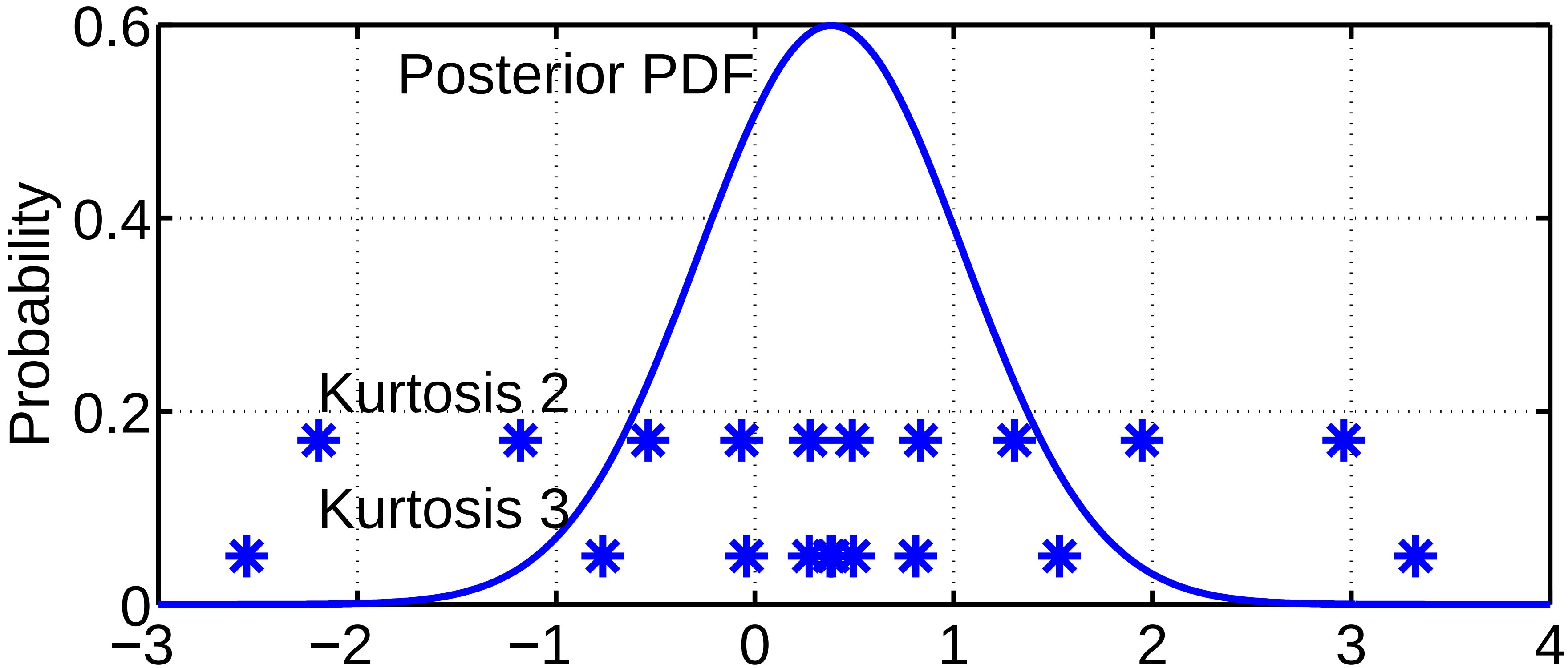
2. Construct a 'deterministic' sample with certain features.



Example: Exact sample mean and variance.
Sample kurtosis is 3 (expected value for Gaussian in large sample limit)
(Constructed by starting uniformly spaced and adjusting quadratically).

Sampling Posterior PDF:

2. Construct a ‘deterministic’ sample with certain feature.



Example: Exact sample mean and variance.

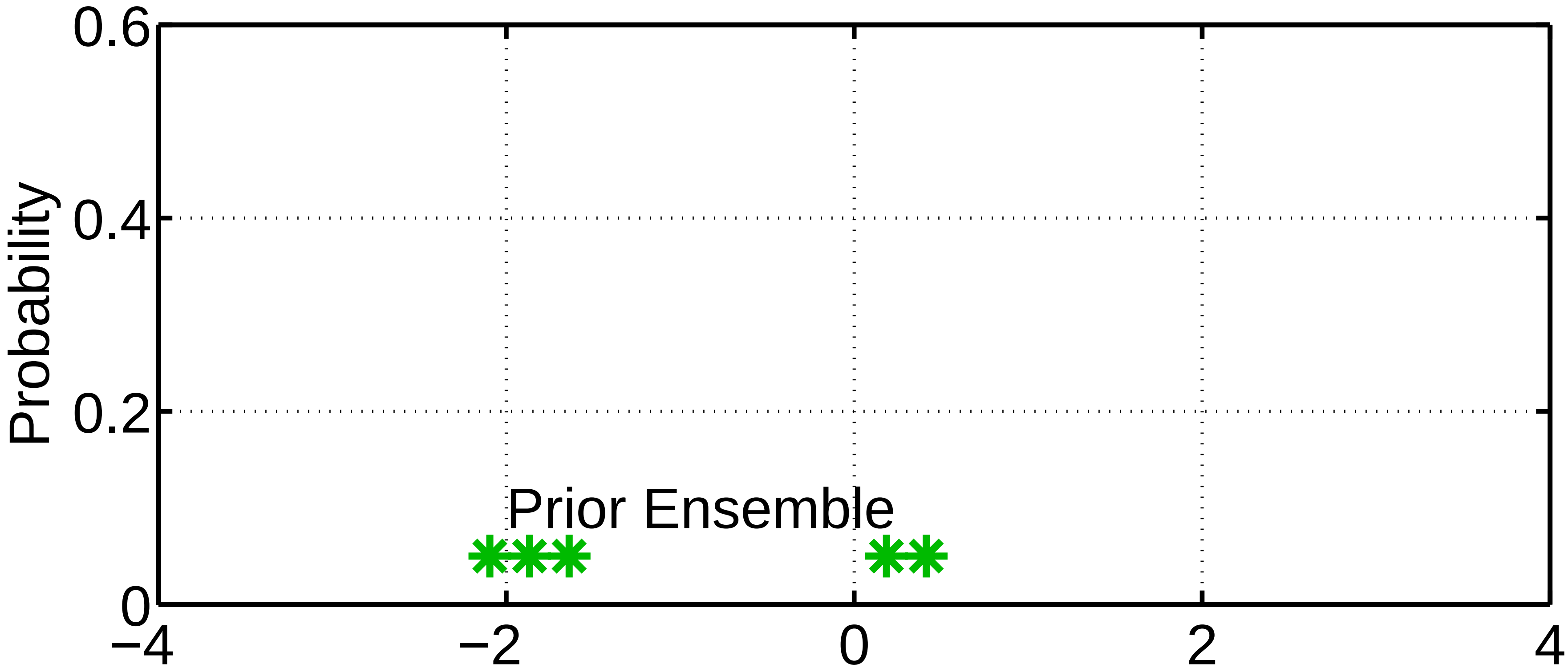
Sample kurtosis 2: less extreme outliers, less dense near mean.

Avoiding outliers might be nice in certain applications.

Sampling heavily near mean might be nice.

Sampling Posterior PDF:

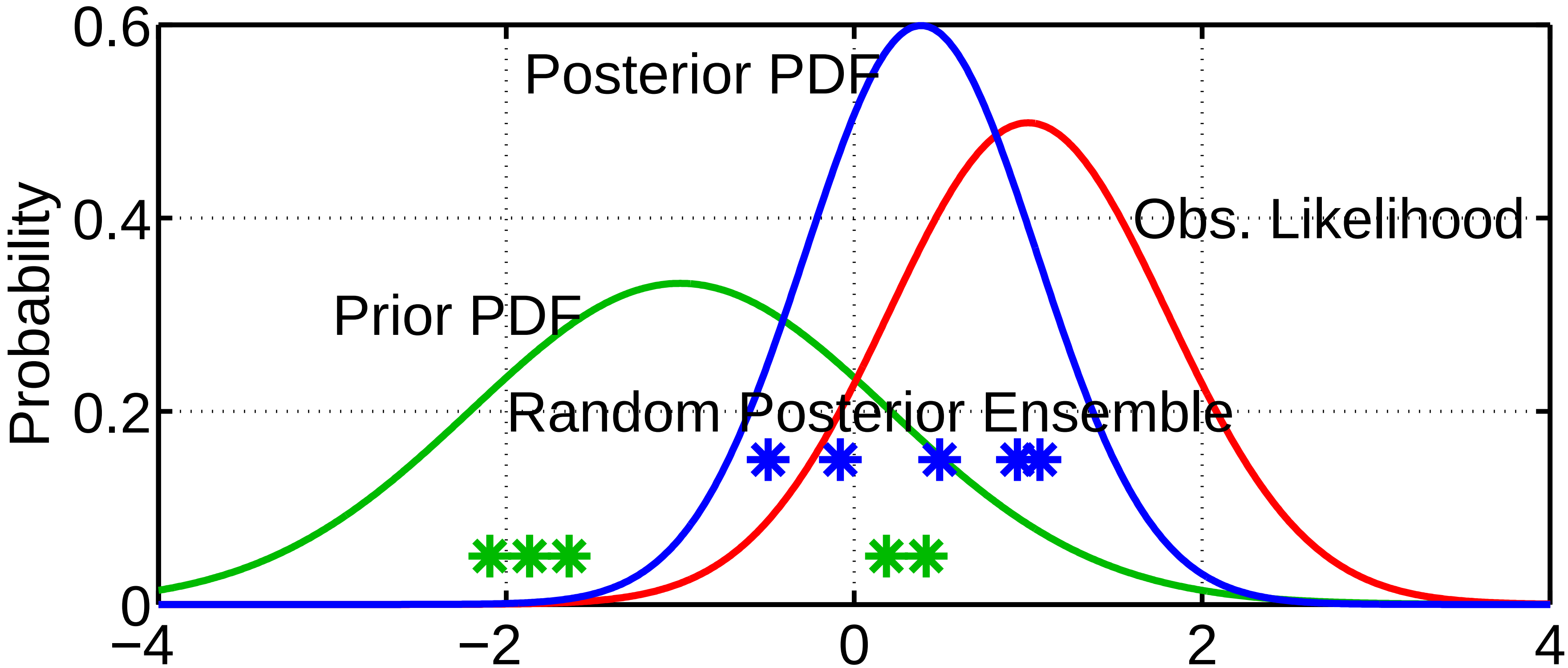
First two methods depend only on mean and variance of prior sample.



Example: Suppose prior sample is (significantly) bimodal?

Sampling Posterior PDF:

First two methods depend only on mean and variance of prior sample.

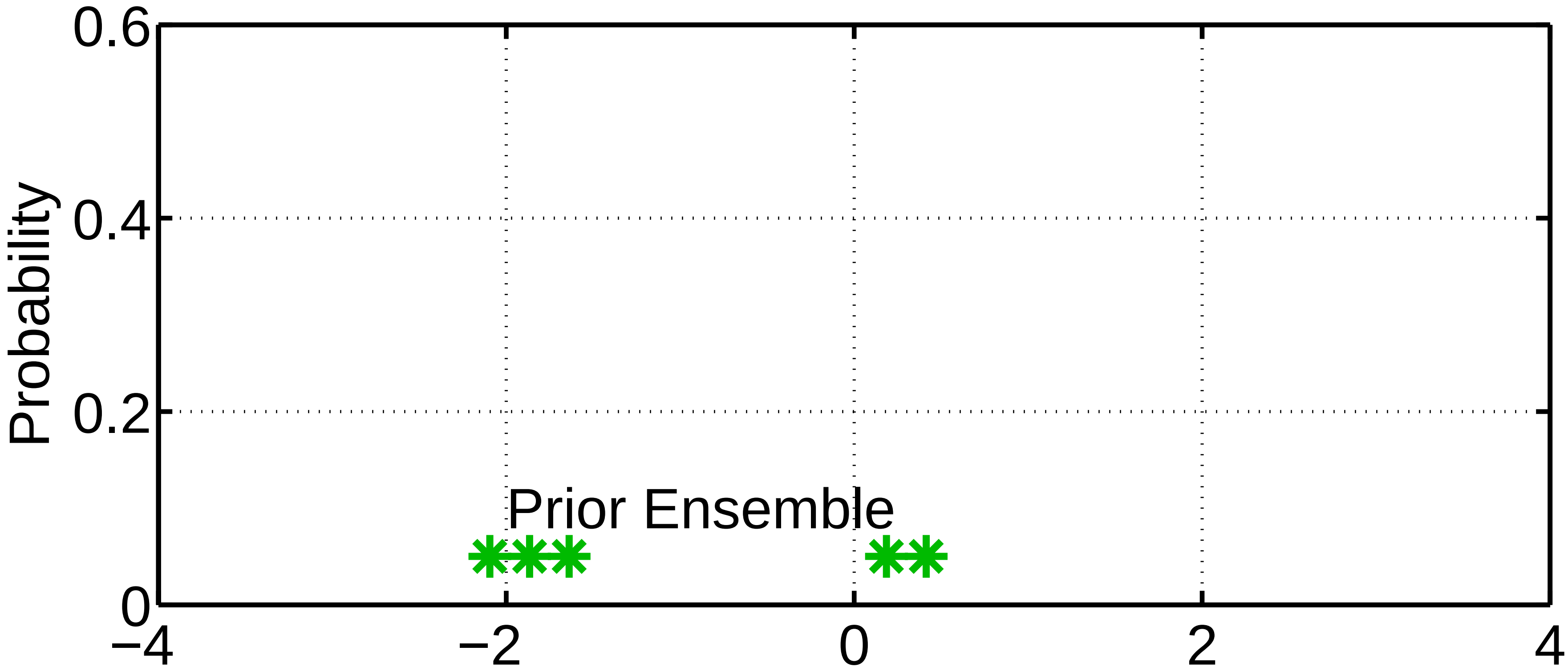


Example: Suppose prior sample is (significantly) bimodal?

Might want to retain additional information from prior.

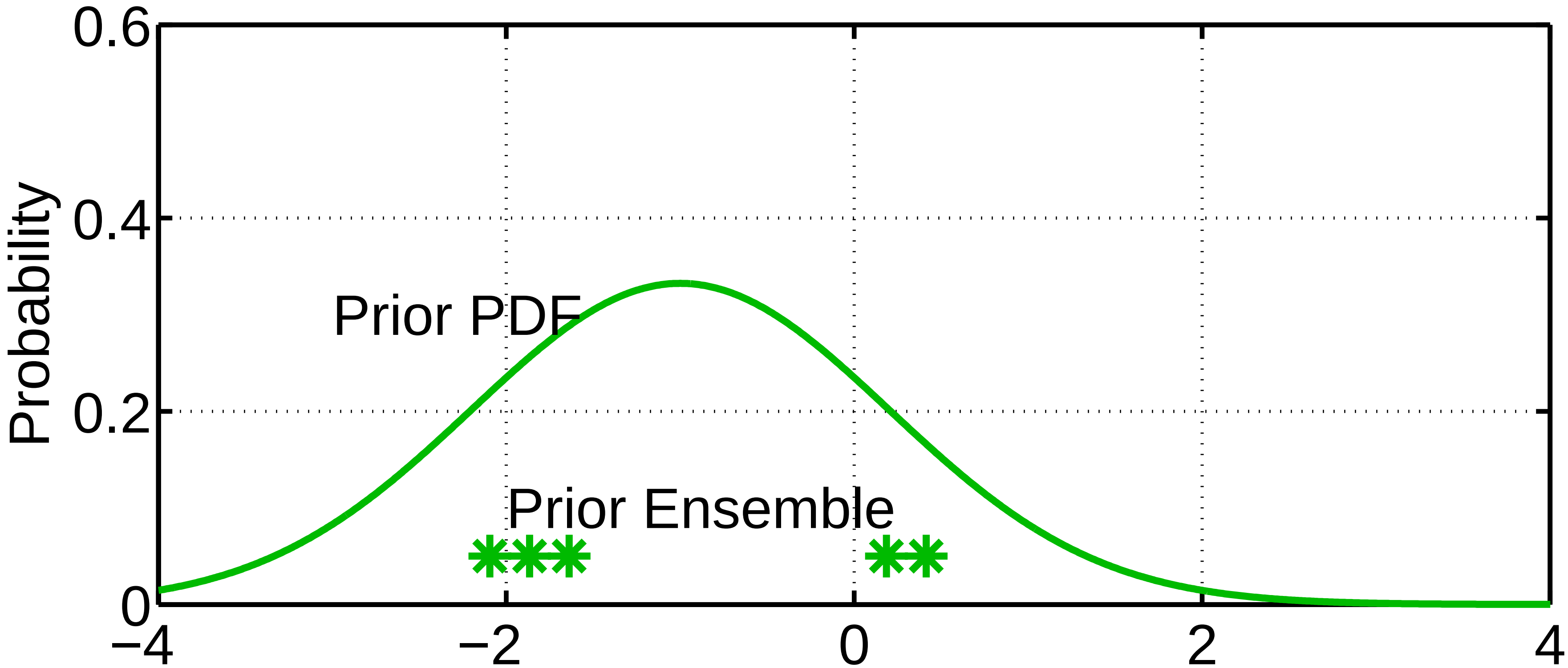
Ensemble Filter Algorithms:

3. Ensemble Adjustment (Kalman) Filter.



Ensemble Filter Algorithms:

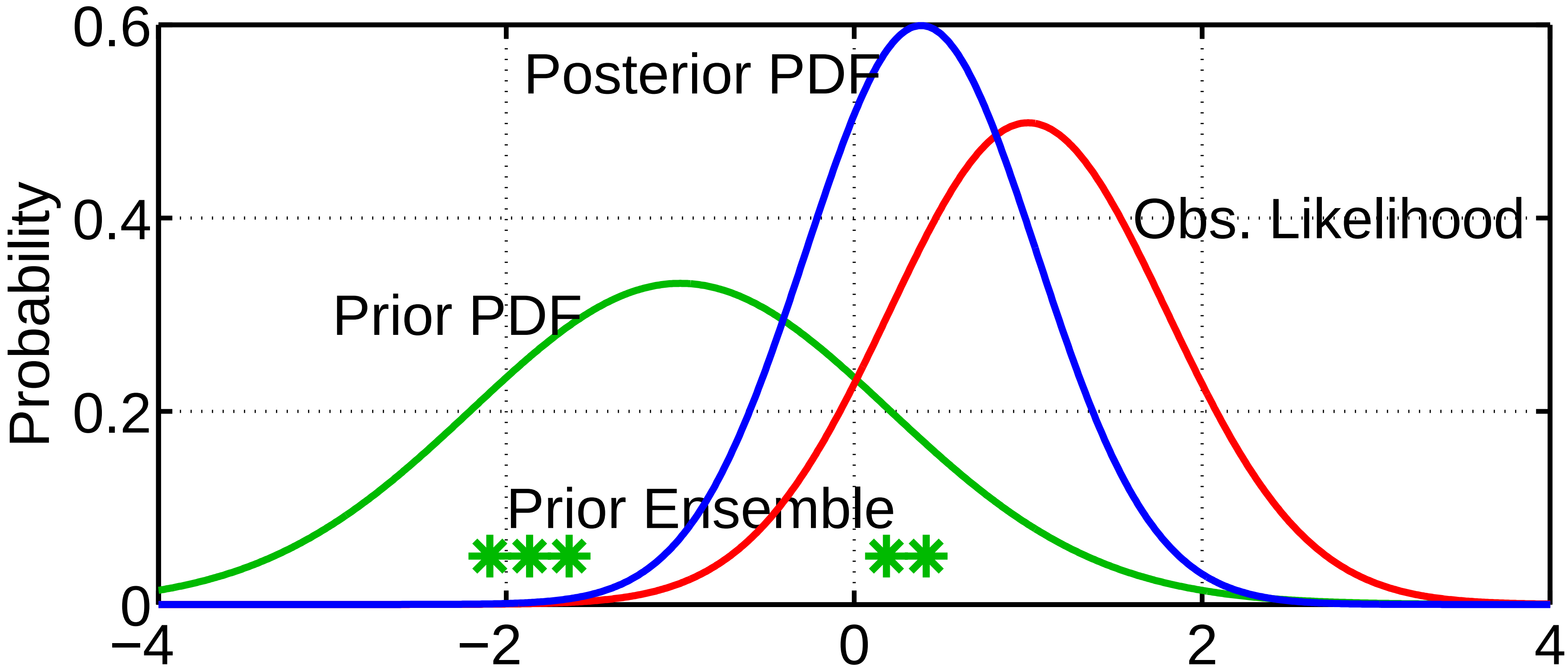
3. Ensemble Adjustment (Kalman) Filter.



Again, fit a Gaussian to sample.

Ensemble Filter Algorithms:

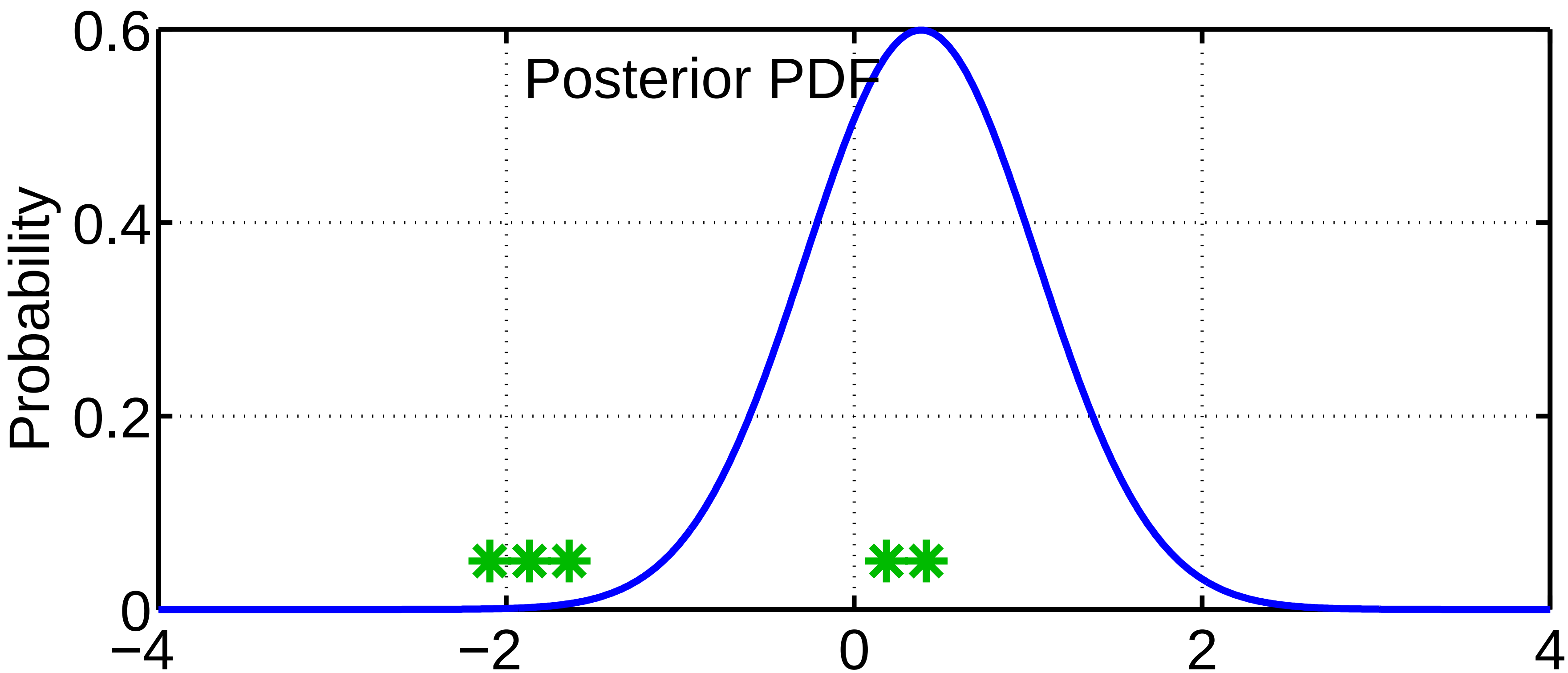
3. Ensemble Adjustment (Kalman) Filter.



Compute posterior PDF (same as previous algorithms).

Ensemble Filter Algorithms:

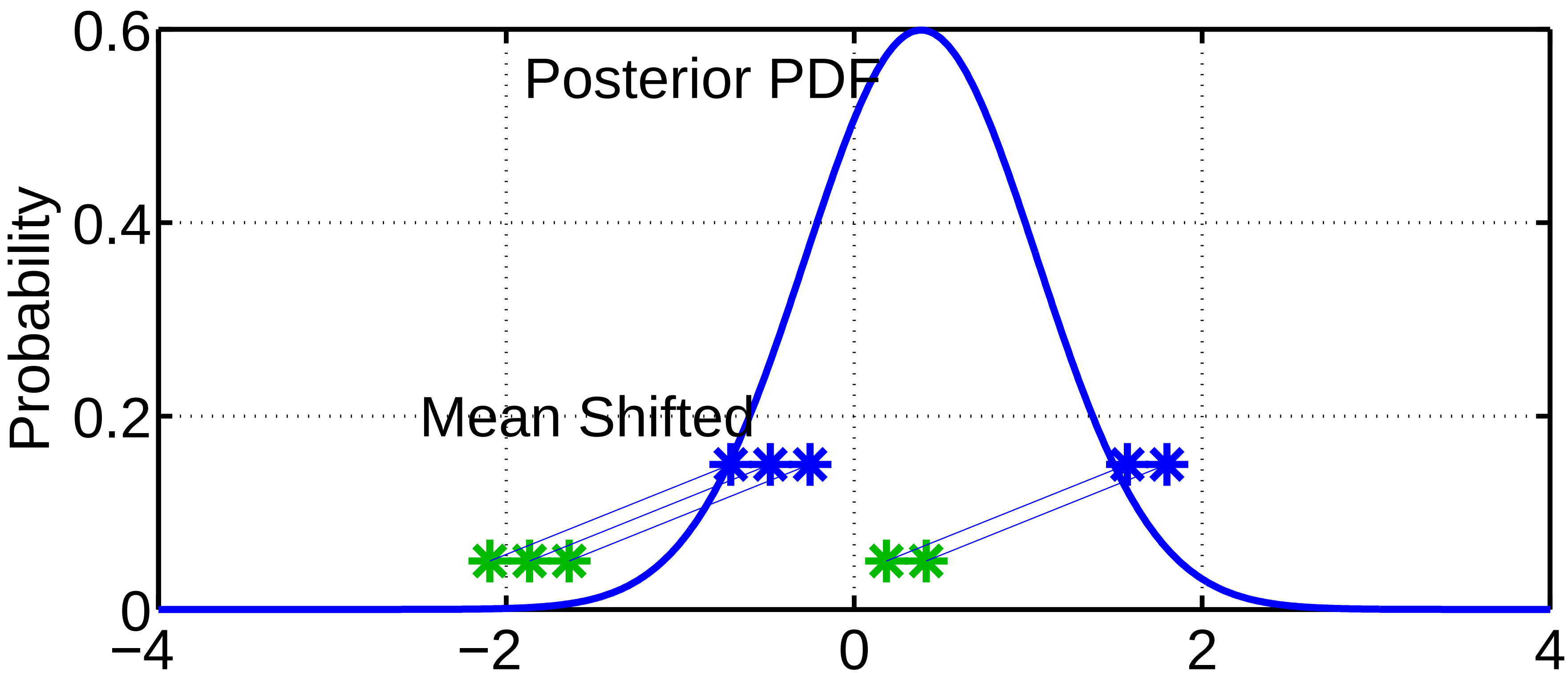
3. Ensemble Adjustment (Kalman) Filter.



Use deterministic algorithm to ‘adjust’ ensemble.

Ensemble Filter Algorithms:

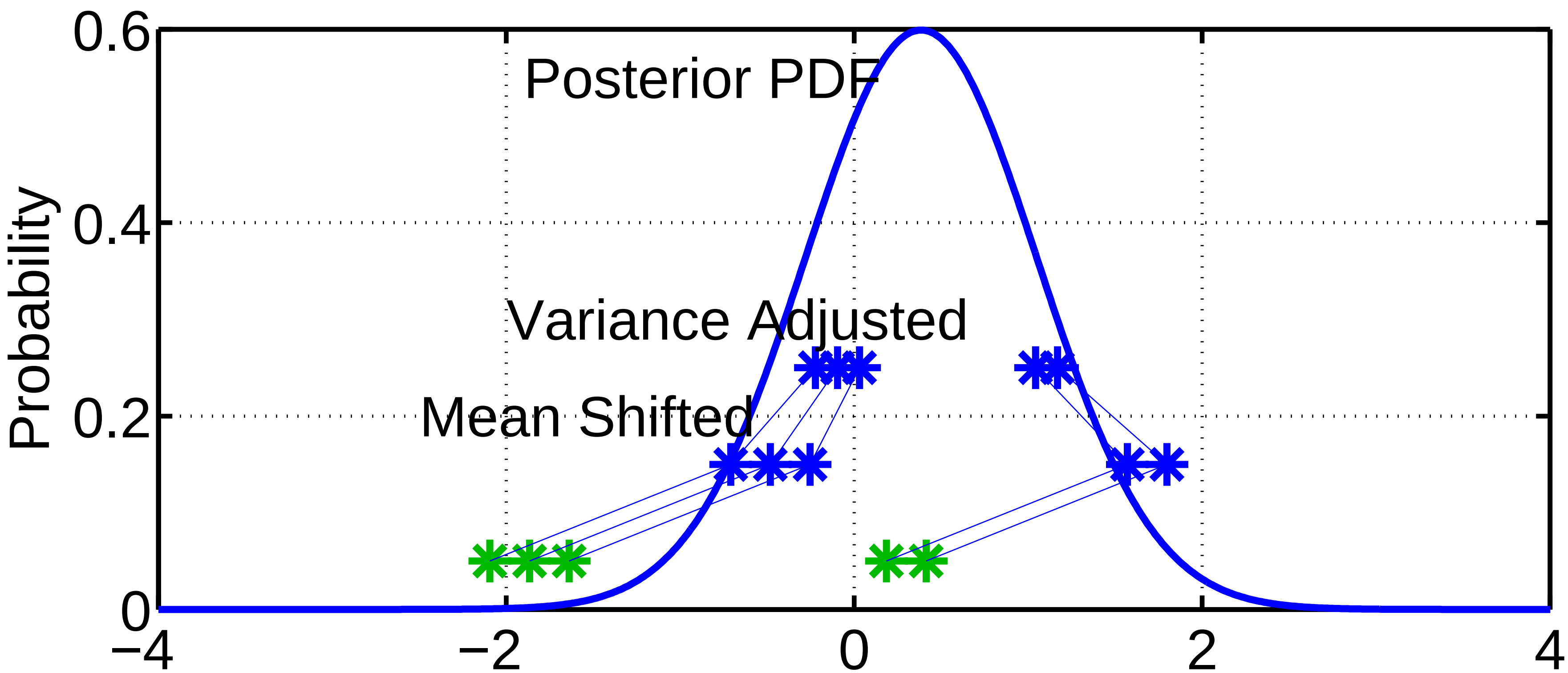
3. Ensemble Adjustment (Kalman) Filter.



Use deterministic algorithm to ‘adjust’ ensemble.

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

3. Ensemble Adjustment (Kalman) Filter.



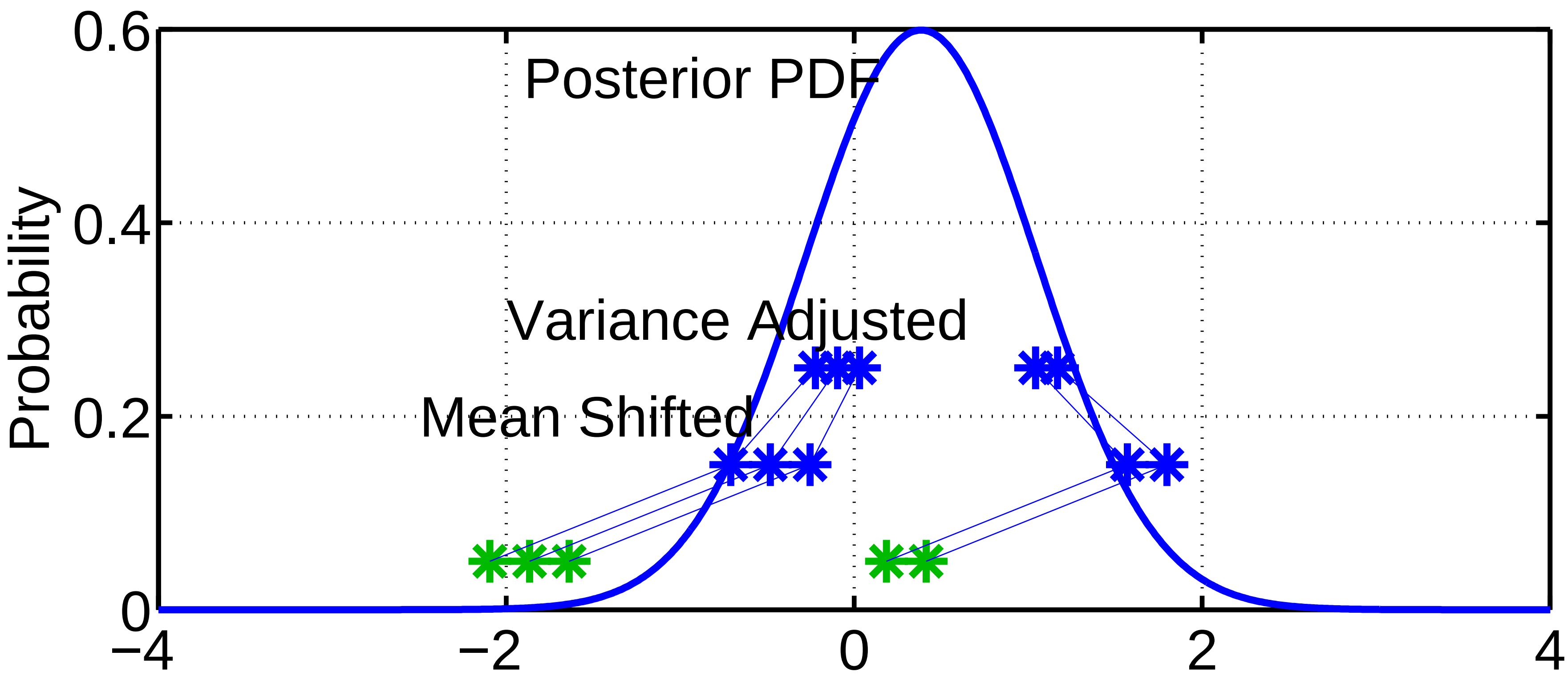
Use deterministic algorithm to ‘adjust’ ensemble.

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.

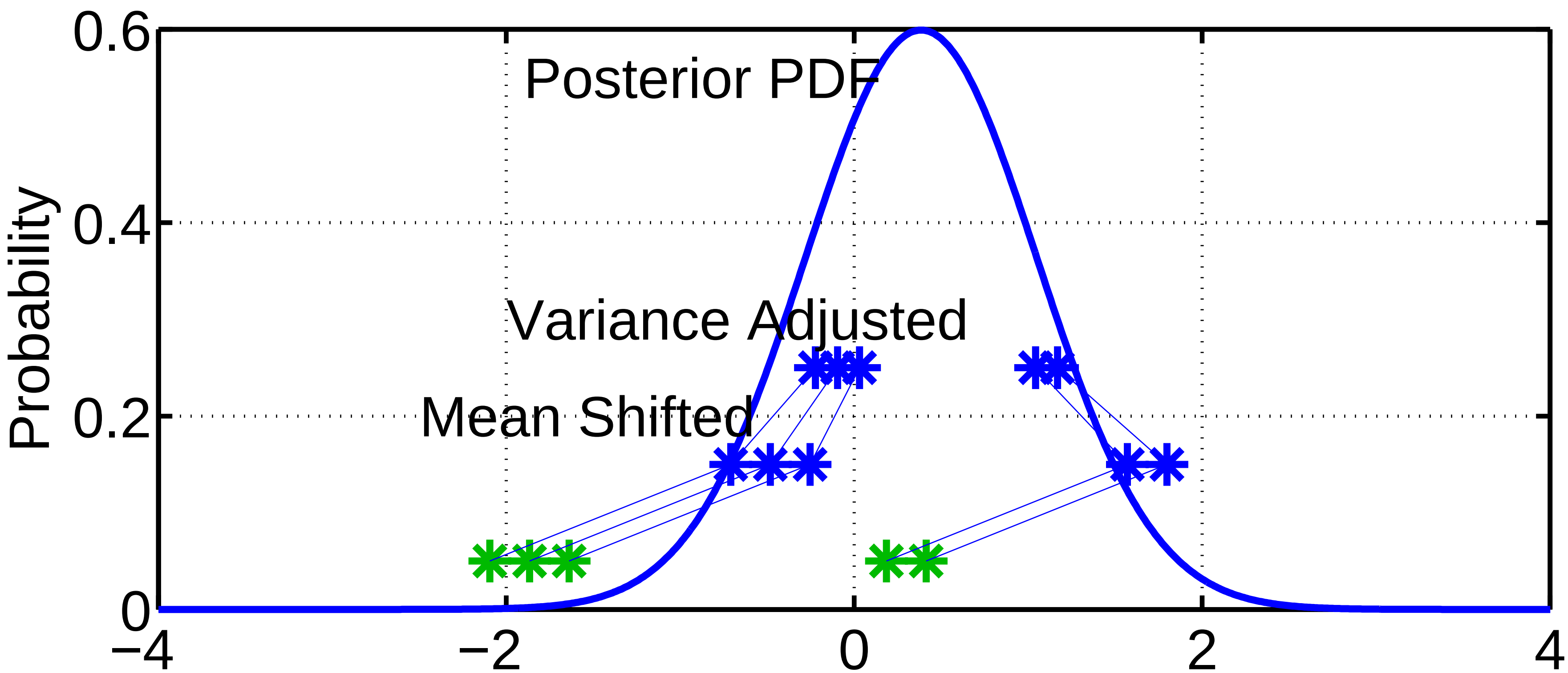


$$x_i^u = (x_i^p - \overline{x^p}) \cdot (\sigma^u / \sigma^p) + \overline{x^u} \quad \text{i = 1,..., ensemble size.}$$

p is prior, u is update (posterior), overbar is ensemble mean, σ is standard deviation.

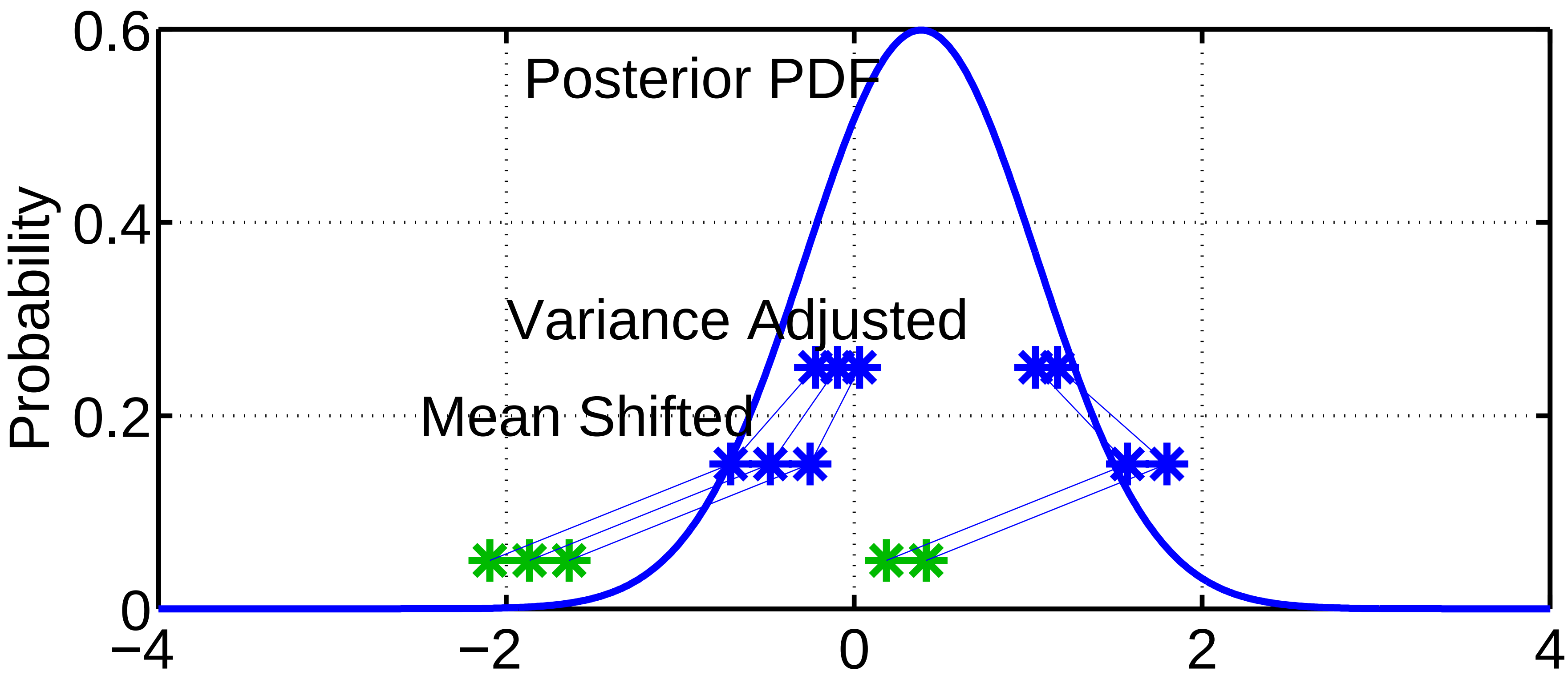
Ensemble Filter Algorithms:

3. Ensemble Adjustment (Kalman) Filter.



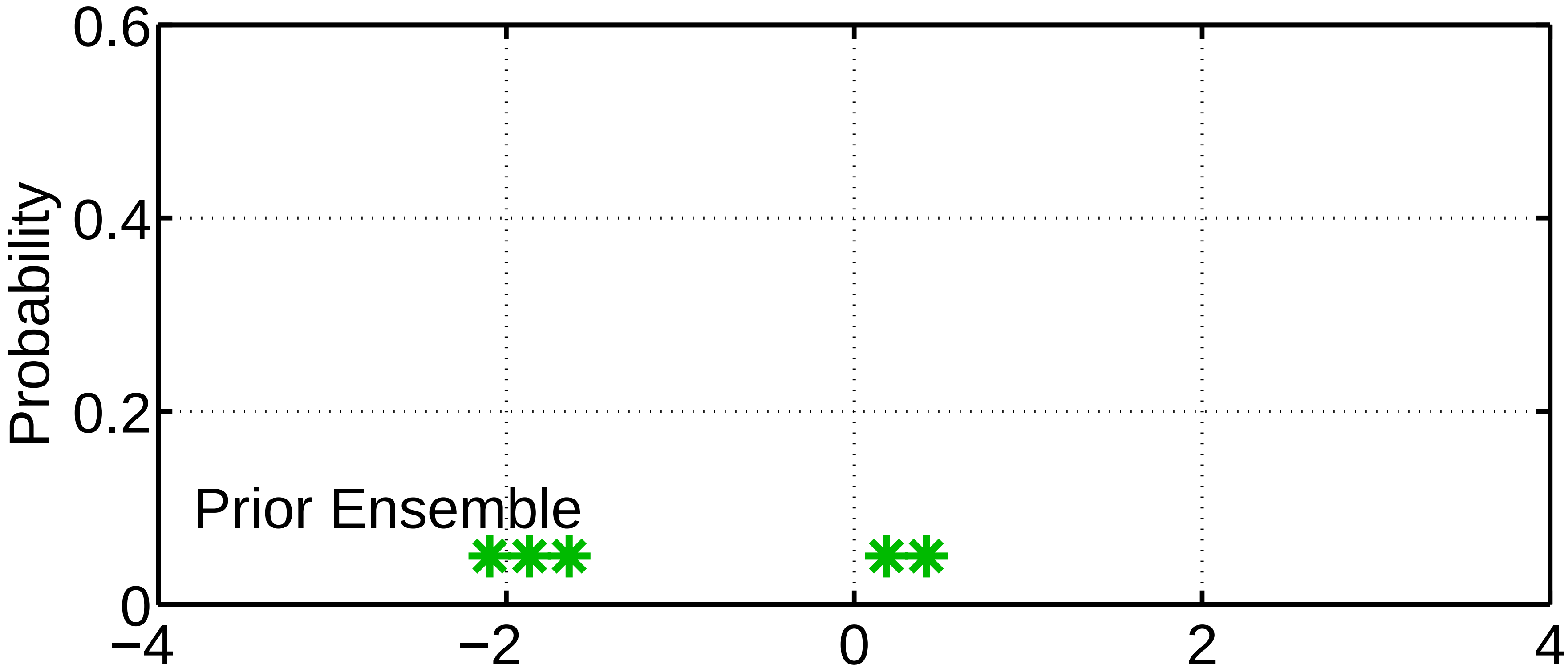
Bimodality maintained, but not appropriately positioned or weighted.
No problem with random outliers.

3. Ensemble Adjustment (Kalman) Filter.



There are a variety of other ways to deterministically adjust ensemble.
Class of algorithms sometimes called deterministic square root filters.

4. Ensemble Kalman Filter (EnKF).

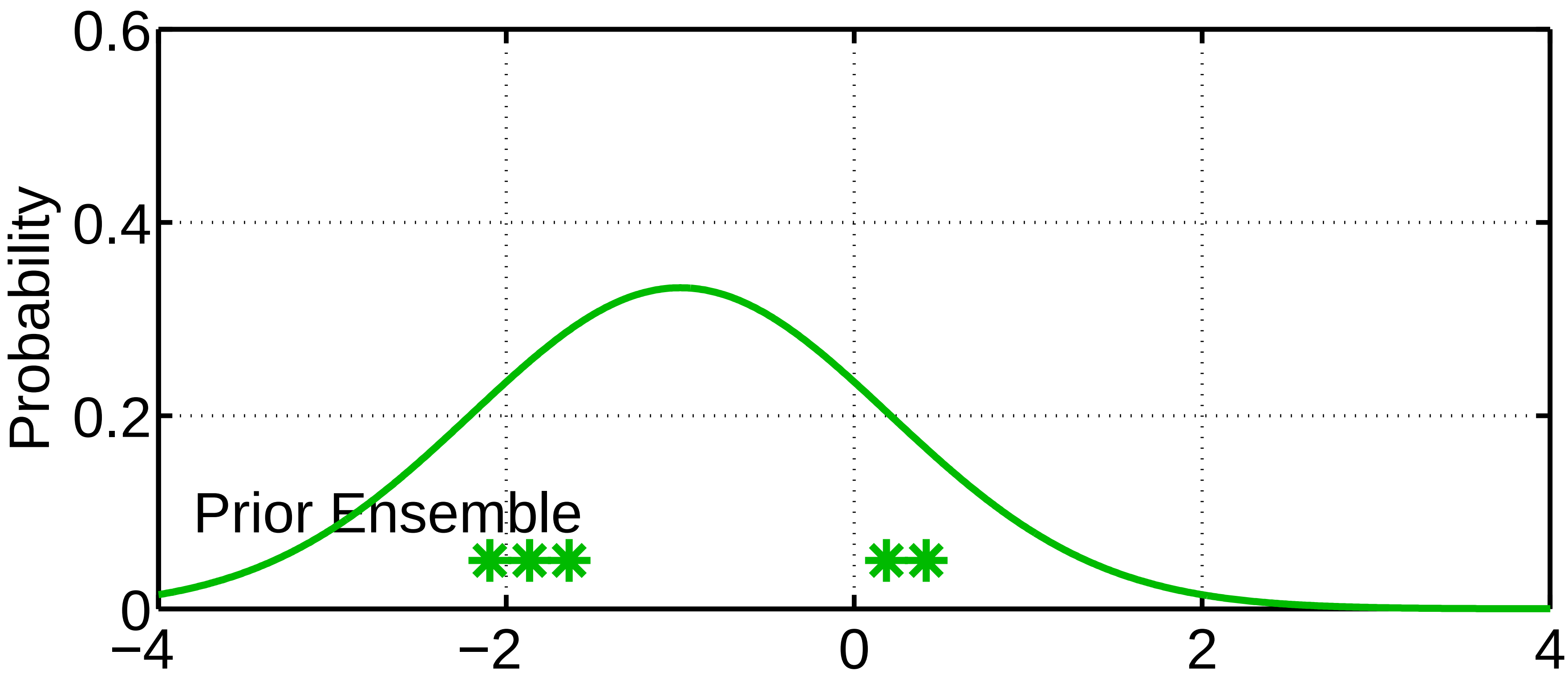


'Classical' Monte Carlo Algorithm for Data Assimilation.

Warning: earliest refs have incorrect algorithm (more in a minute).

Ensemble Filter Algorithms:

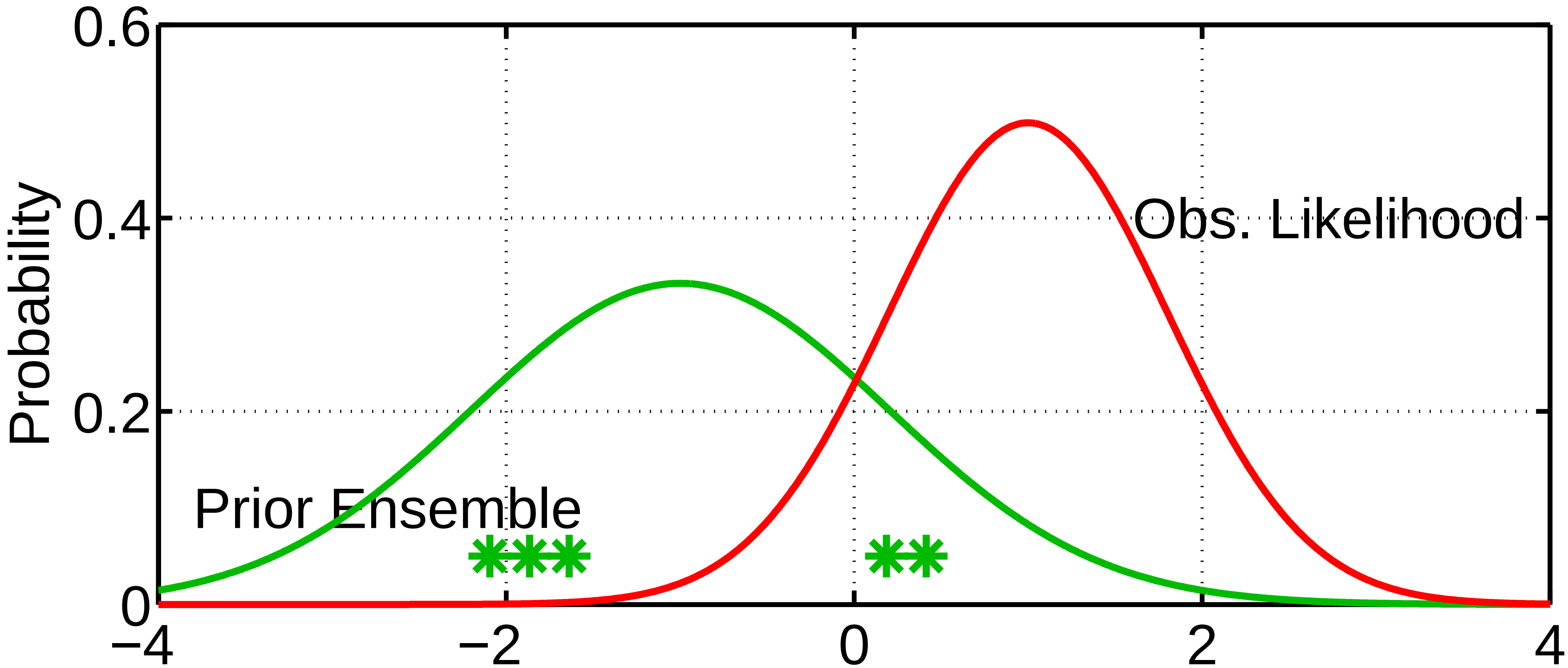
4. Ensemble Kalman Filter (EnKF).



Again, fit a Gaussian to sample.

Are there ways to do this without computing prior sample stats?

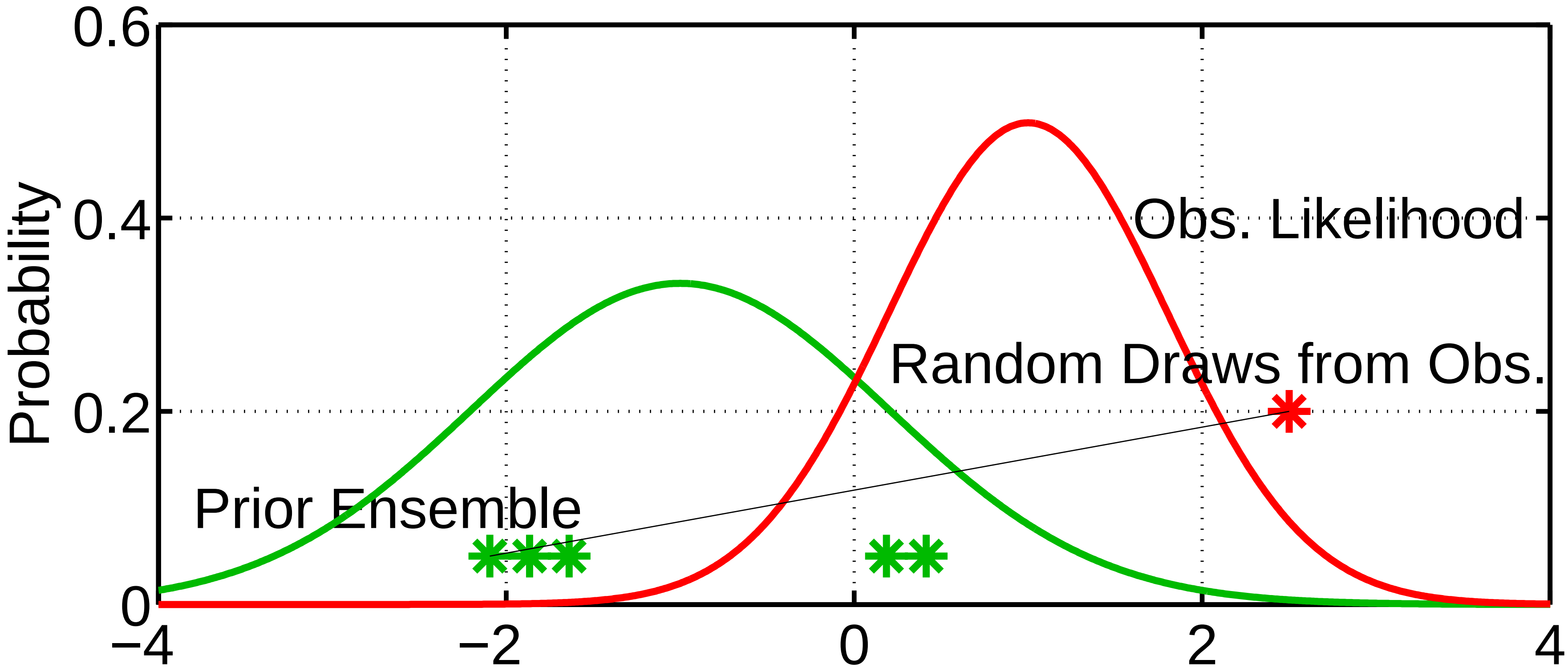
4. Ensemble Kalman Filter (EnKF).



Again, fit a Gaussian to sample.

Ensemble Filter Algorithms:

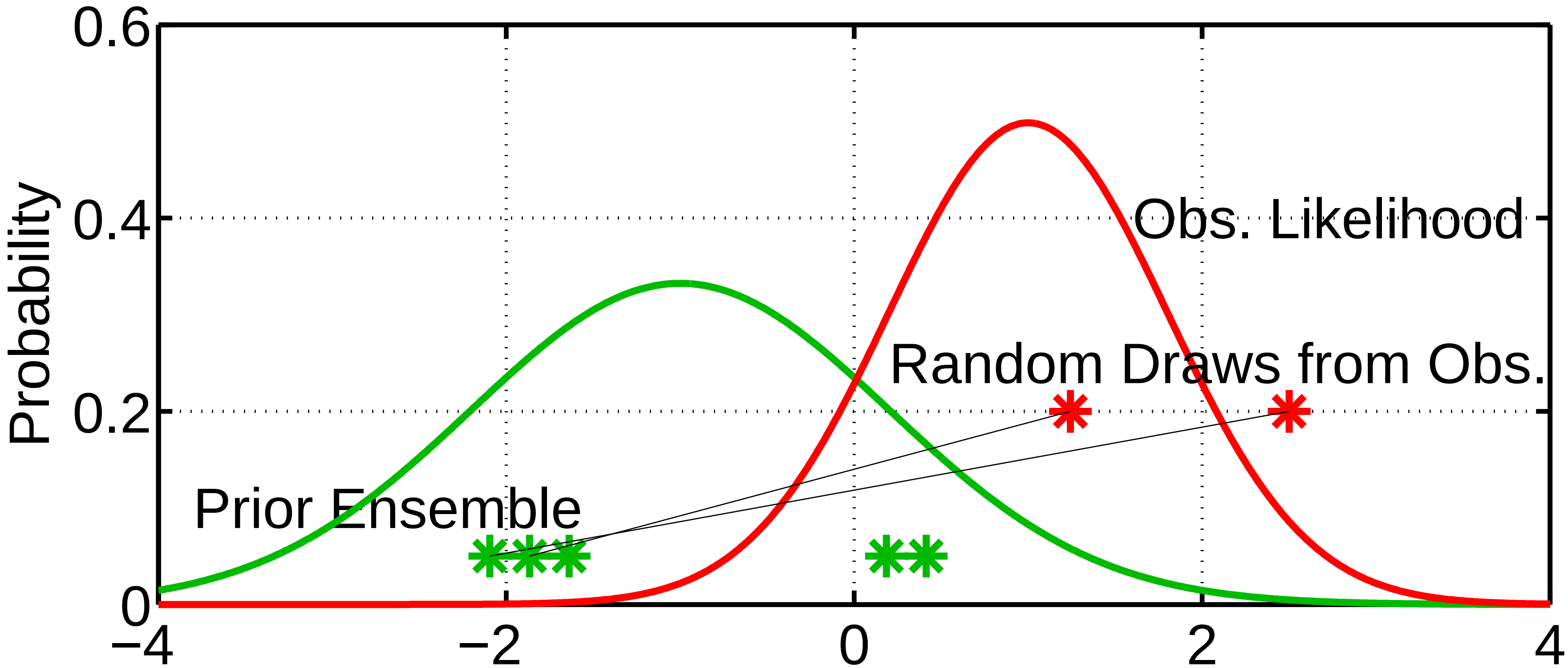
4. Ensemble Kalman Filter (EnKF).



Generate a random draw from the obs. likelihood.

Associate it with the first sample of prior ensemble.

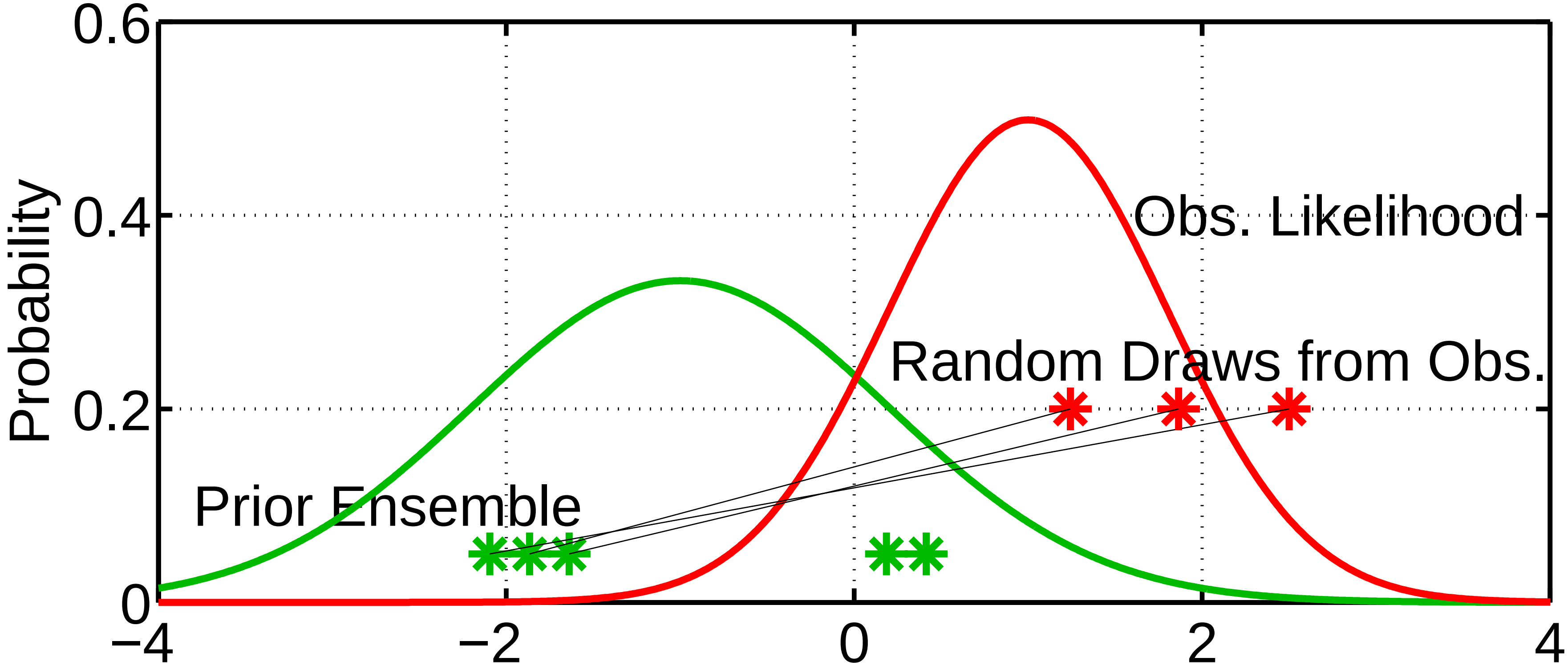
4. Ensemble Kalman Filter (EnKF).



Proceed to associate a random draw from obs. with each prior sample.
This has been called ‘perturbed’ observations.
Algorithm sometimes called ‘perturbed obs.’ ensemble Kalman filter.

Ensemble Filter Algorithms:

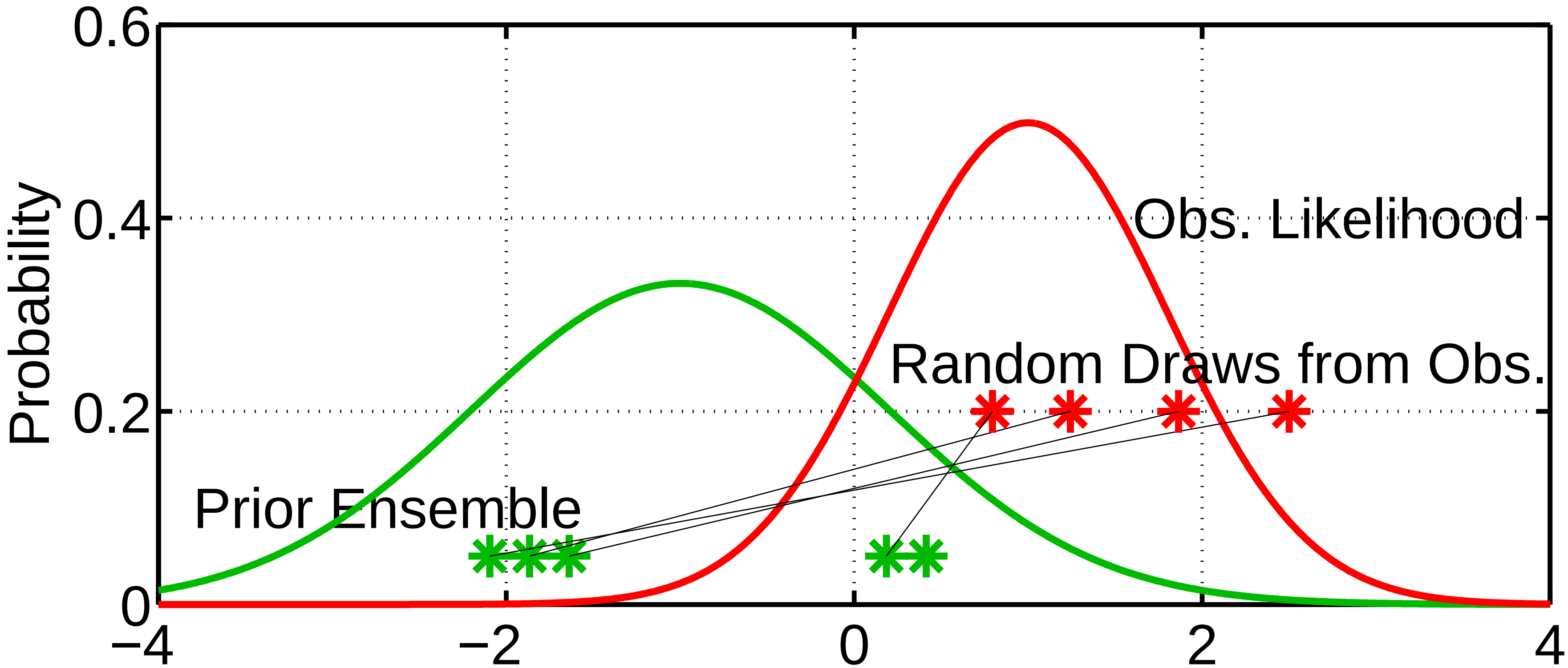
4. Ensemble Kalman Filter (EnKF).



Proceed to associate a random draw from obs. with each prior sample.
This has been called ‘perturbed’ observations.
Algorithm sometimes called ‘perturbed obs.’ ensemble Kalman filter.

Ensemble Filter Algorithms:

4. Ensemble Kalman Filter (EnKF).



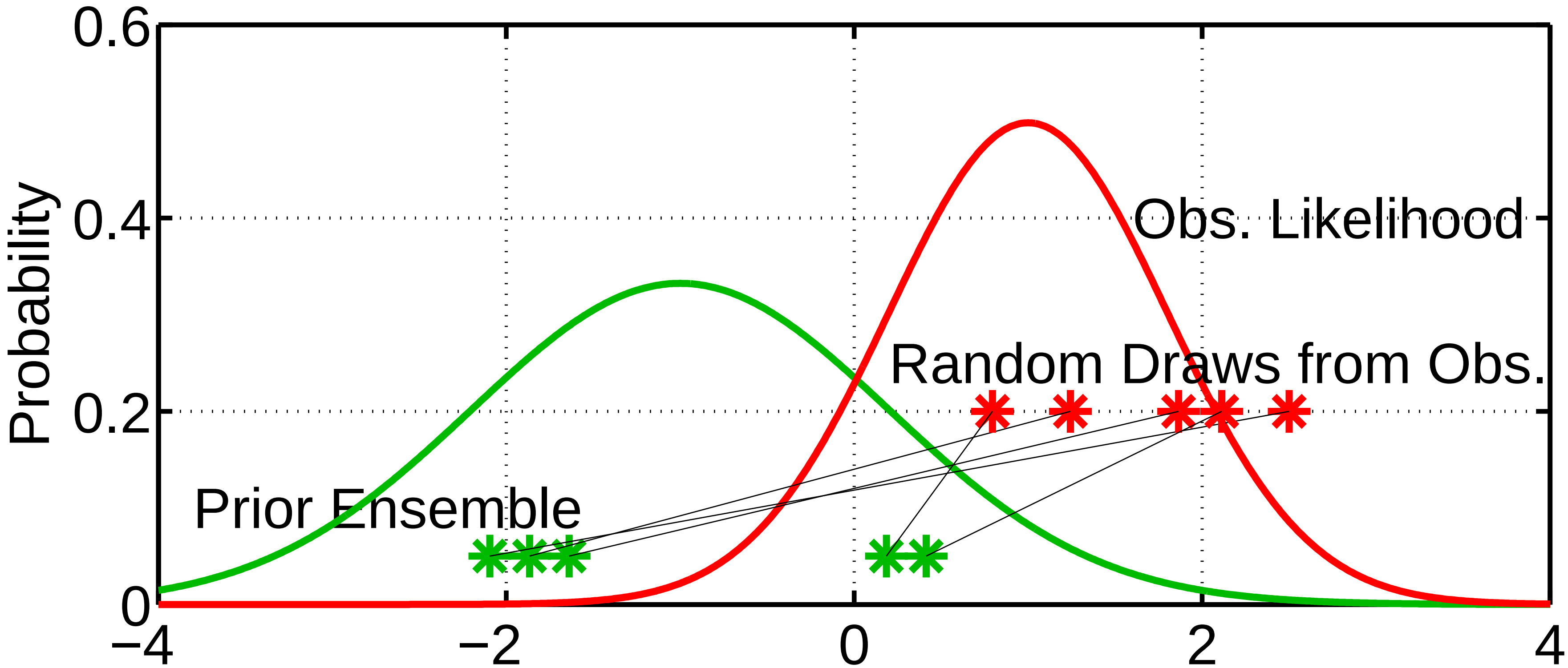
Proceed to associate a random draw from obs. with each prior sample.

Earliest publications associated mean of obs. likelihood with each prior

This resulted in insufficient variance in posterior.

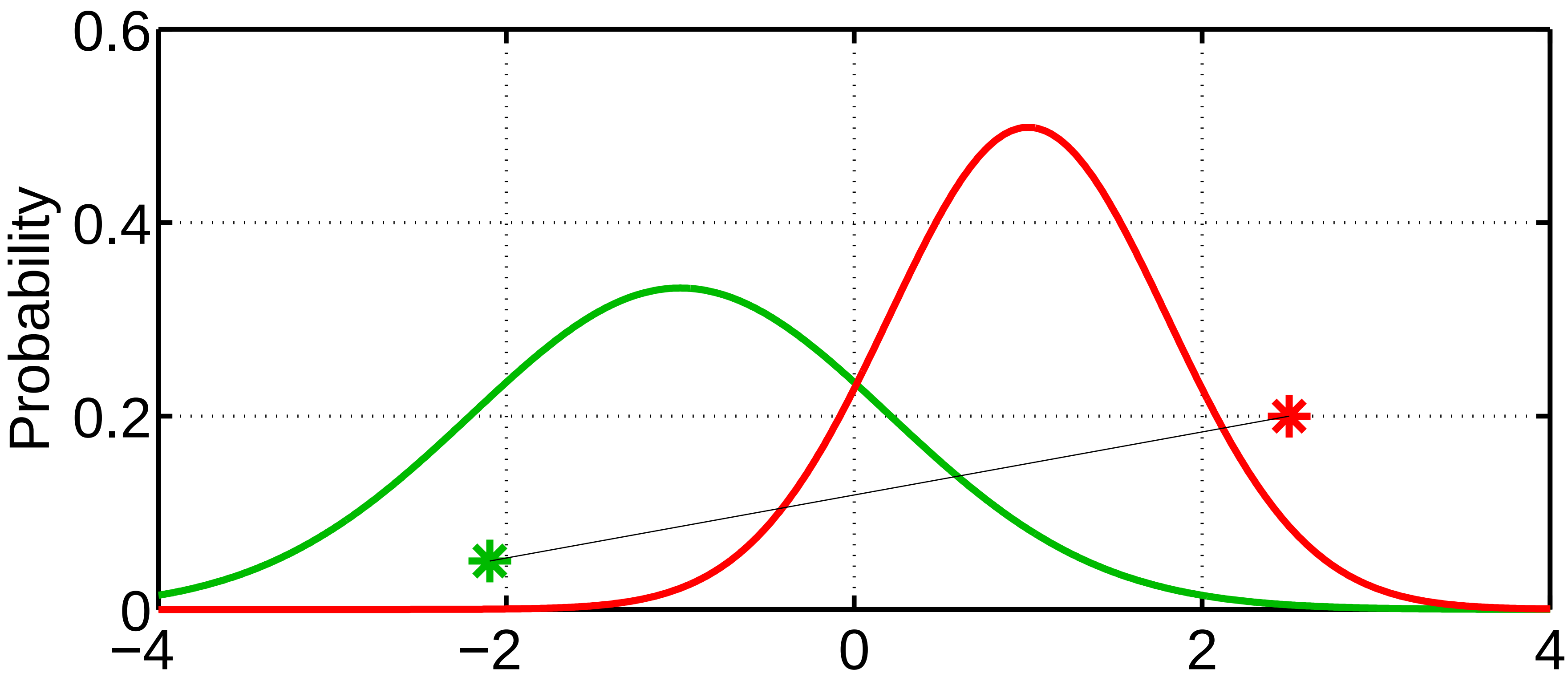
Ensemble Filter Algorithms:

4. Ensemble Kalman Filter (EnKF).



Have sample of joint prior distribution for observation and prior MEAN
Adjusting the mean of obs. sample to be exact improves performance.
Adjusting the variance may further improve performance.
Outliers are potential problem, but can be removed.

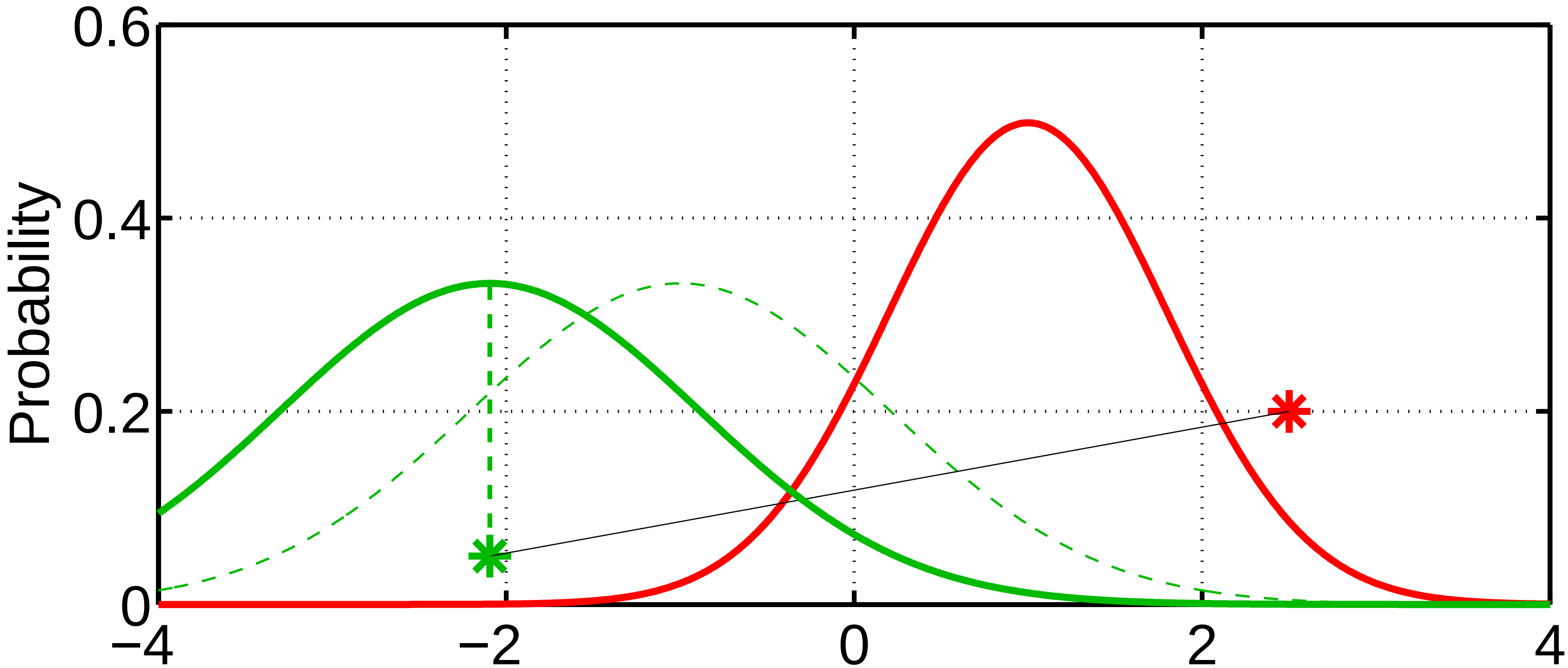
4. Ensemble Kalman Filter (EnKF).



For each prior mean/obs. pair, find mean of posterior PDF.

Ensemble Filter Algorithms:

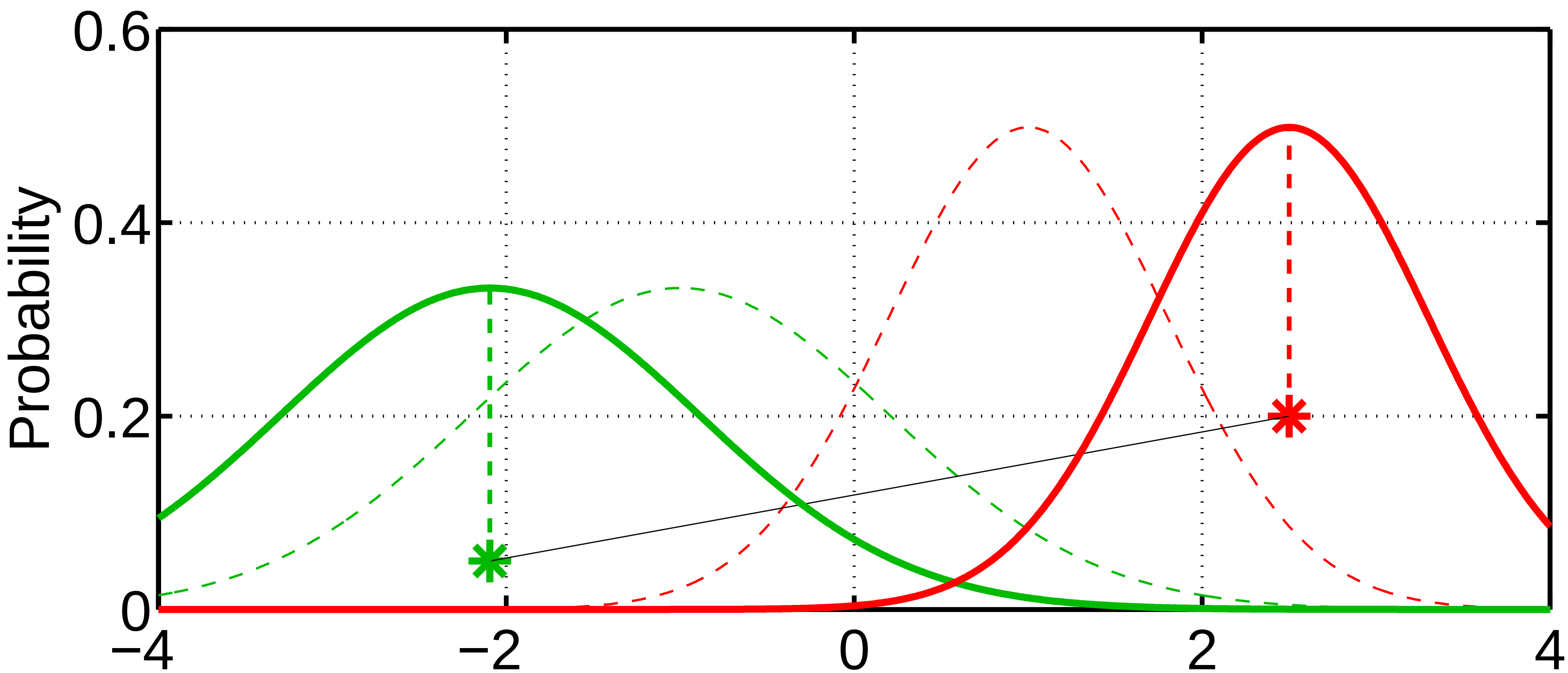
4. Ensemble Kalman Filter (EnKF).



Prior sample standard deviation still measures uncertainty of prior mean estimate.

Ensemble Filter Algorithms:

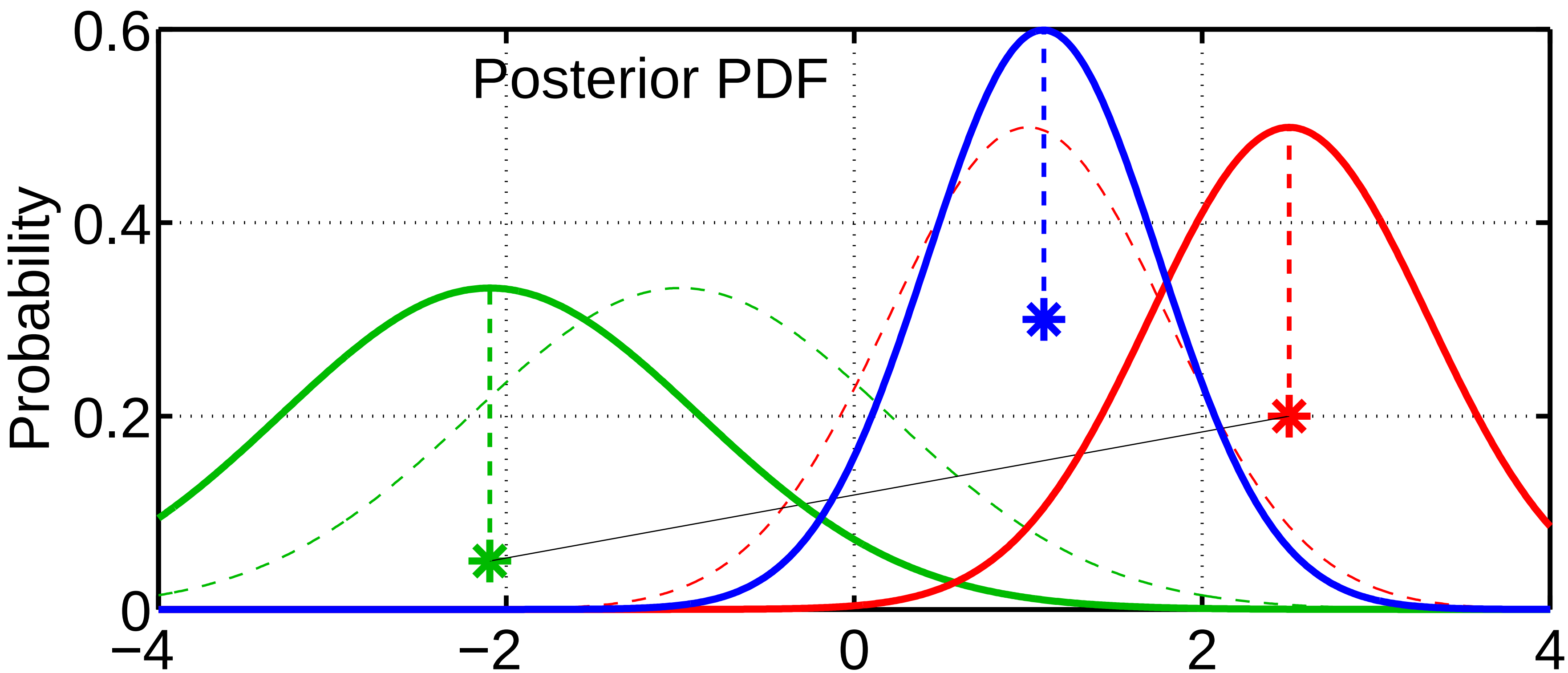
4. Ensemble Kalman Filter (EnKF).



Prior sample standard deviation still measures uncertainty of prior mean estimate.

Obs. likelihood standard deviation measures uncertainty of obs. estimate.

4. Ensemble Kalman Filter (EnKF).

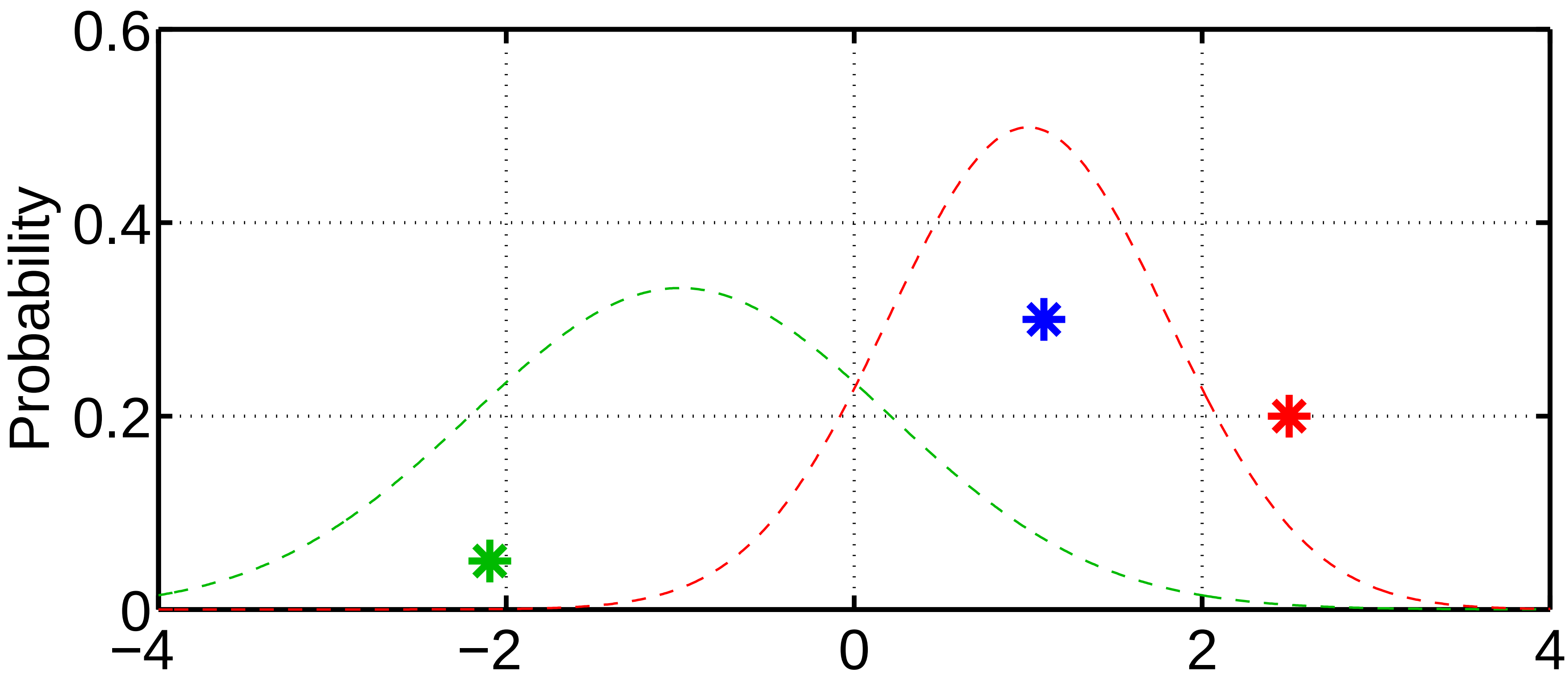


Take product of the prior/obs distributions for first sample.

This is standard Gaussian product.

Ensemble Filter Algorithms:

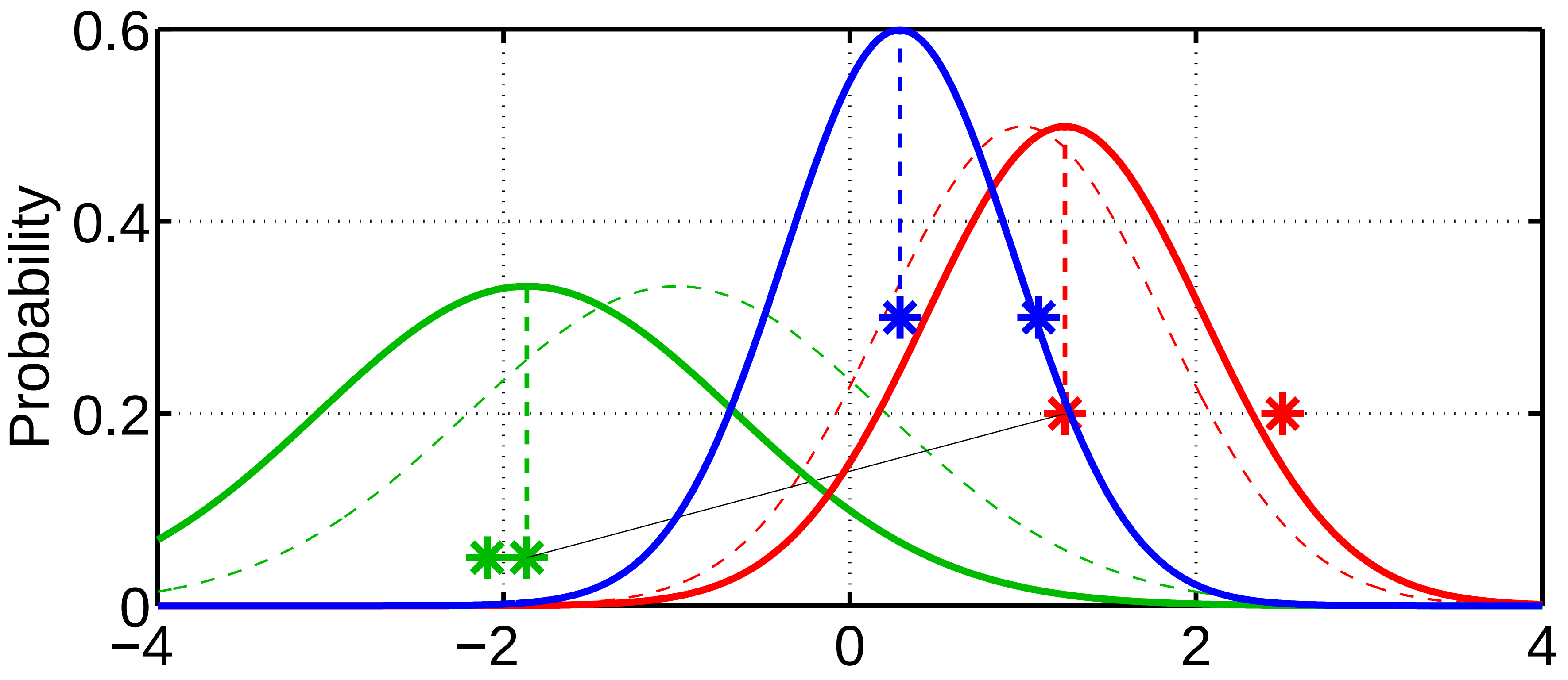
4. Ensemble Kalman Filter (EnKF).



Mean of product is random sample of posterior.

Product of random samples is random sample of product.

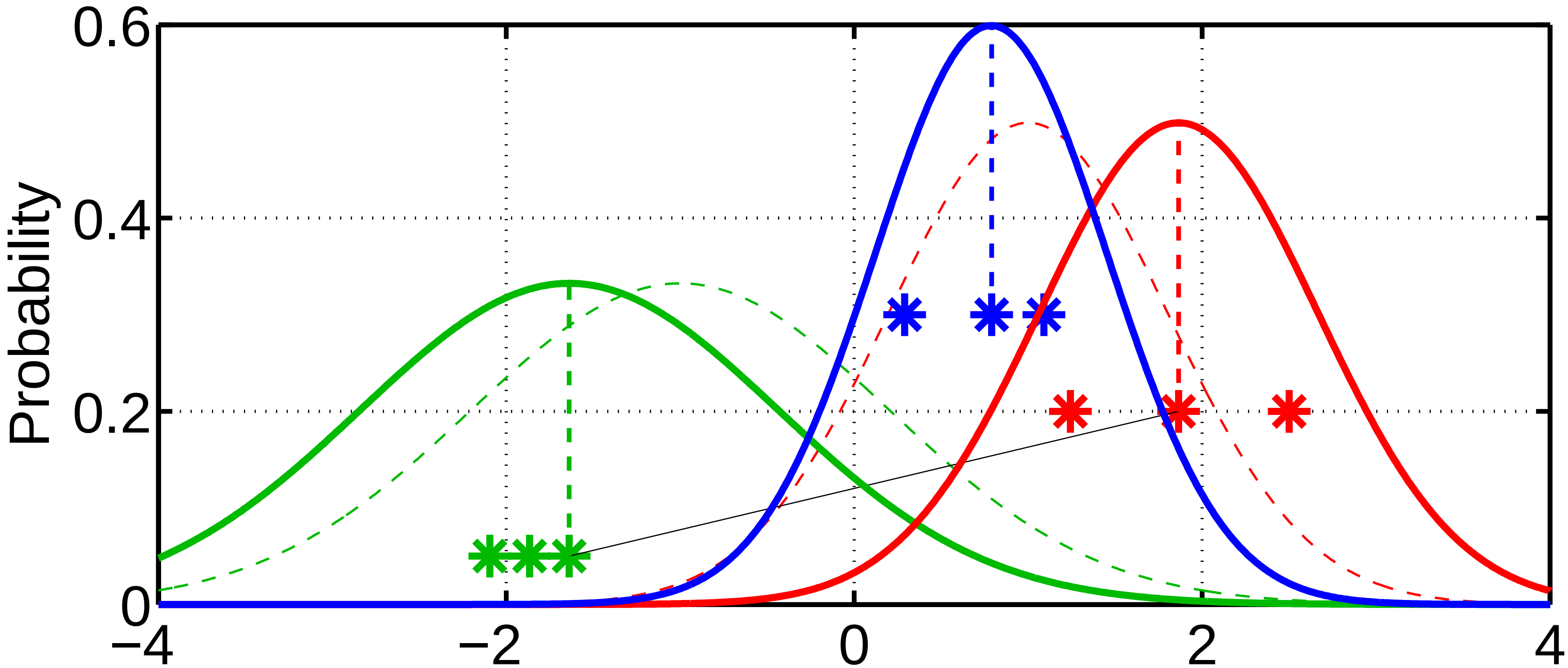
4. Ensemble Kalman Filter (EnKF).



Repeat this operation for each joint prior pair.

Ensemble Filter Algorithms:

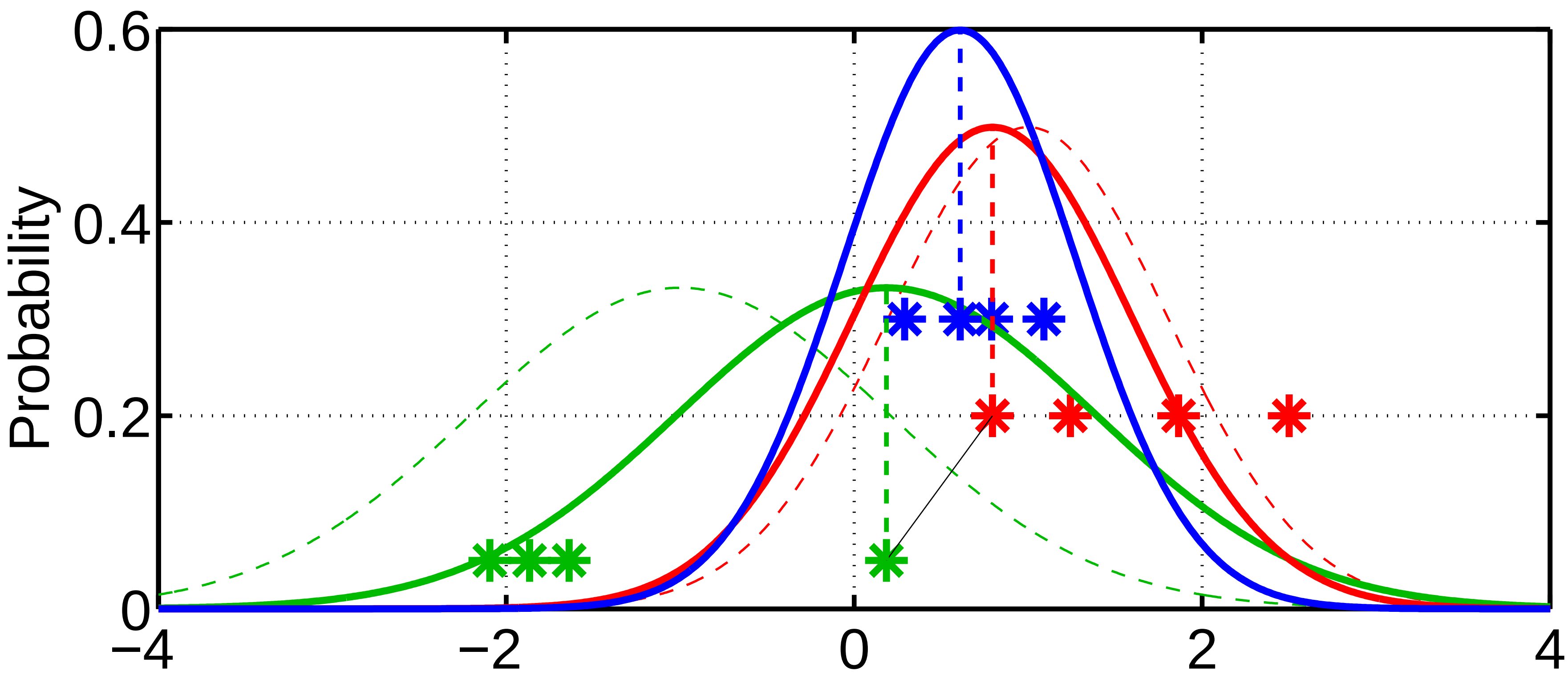
4. Ensemble Kalman Filter (EnKF).



Repeat this operation for each joint prior pair.

Ensemble Filter Algorithms:

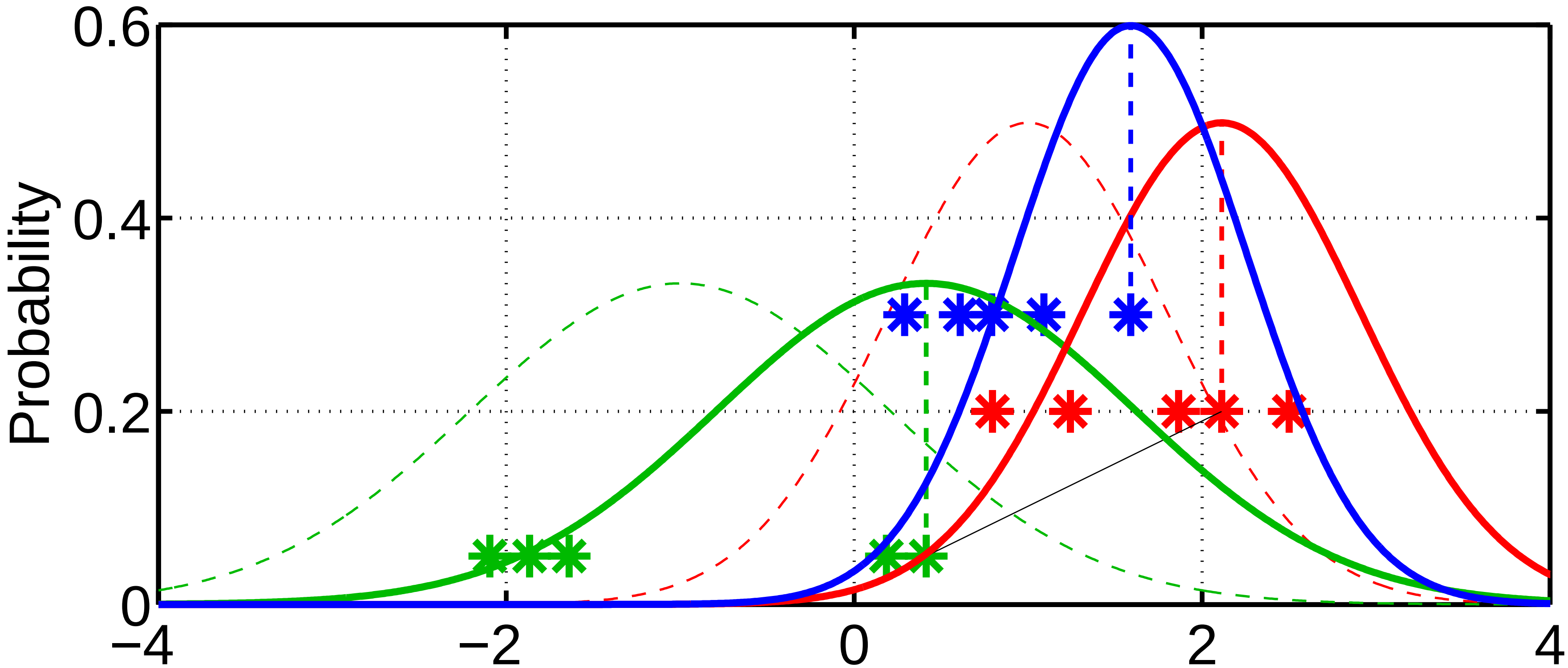
4. Ensemble Kalman Filter (EnKF).



Repeat this operation for each joint prior pair.

Ensemble Filter Algorithms:

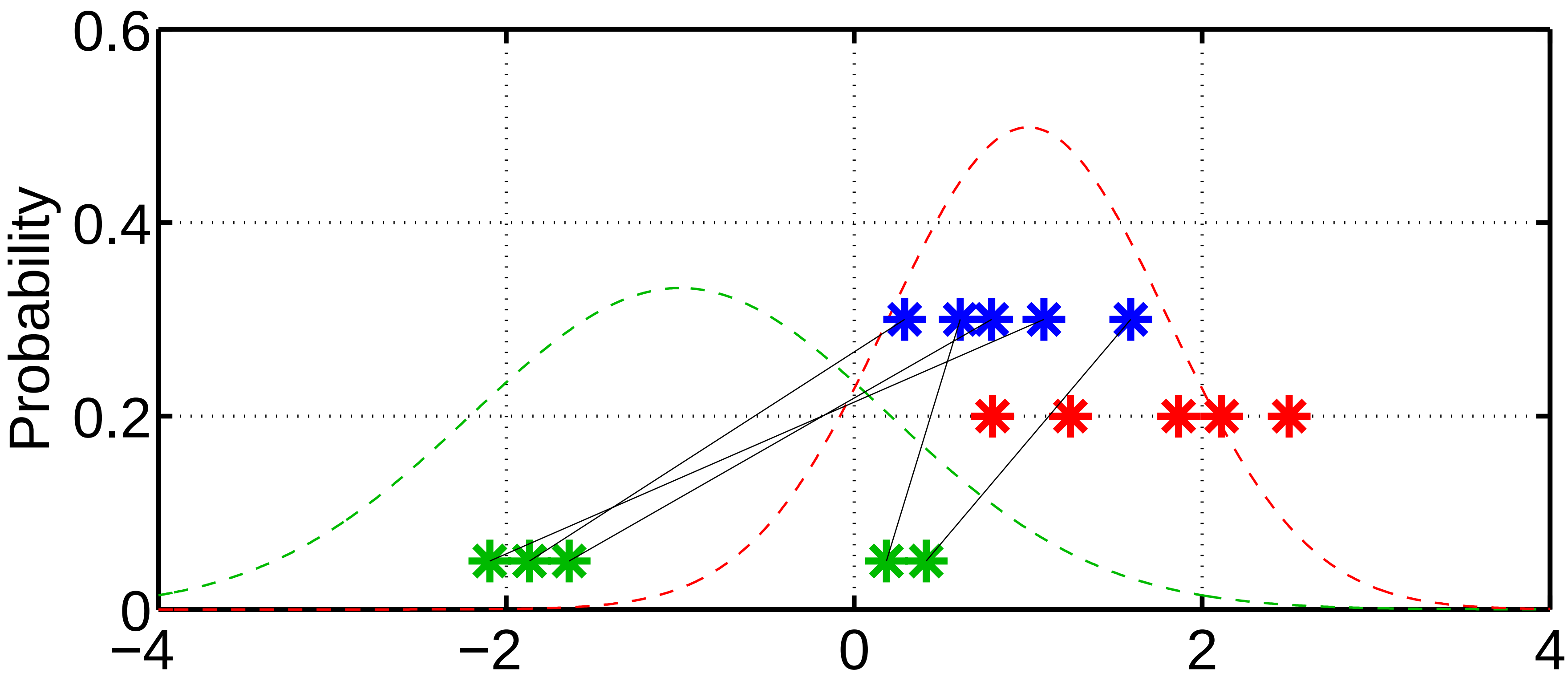
4. Ensemble Kalman Filter (EnKF).



Repeat this operation for each joint prior pair.

Ensemble Filter Algorithms:

4. Ensemble Kalman Filter (EnKF).



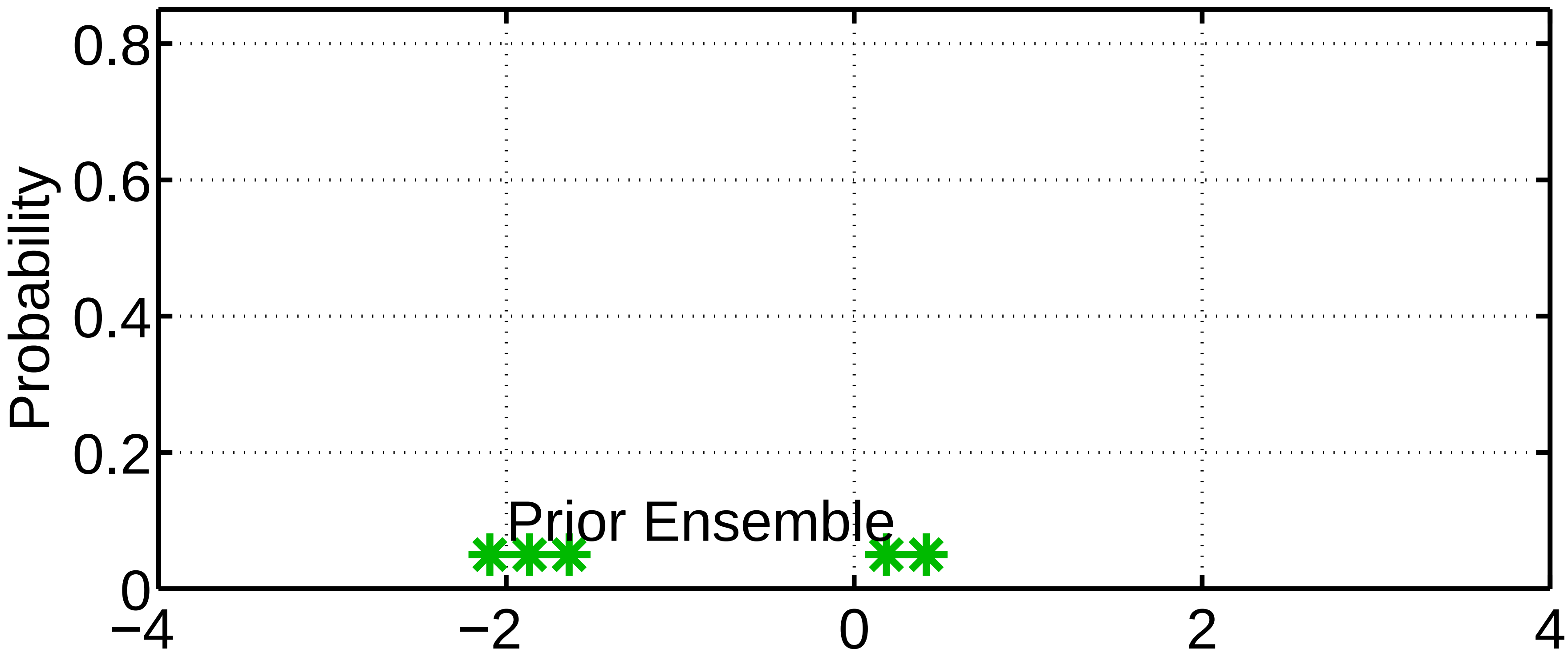
Posterior sample maintains much of prior sample structure.

(This is more apparent for larger ensemble sizes).

Posterior sample mean and variance converge to ‘exact’ for large samples.

Ensemble Filter Algorithms:

5. Ensemble Kernel filter.

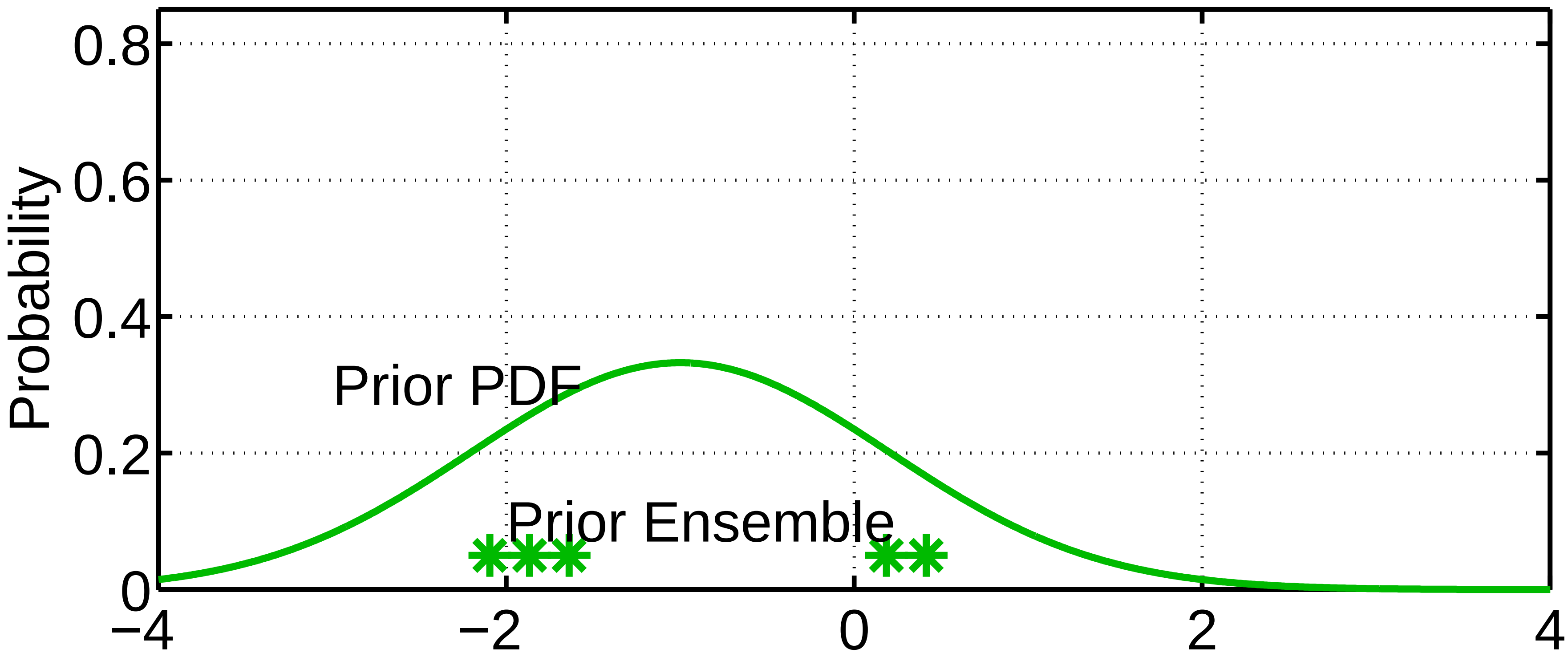


Can retain more correct information about non-Gaussian priors.

Can also be used for obs. likelihood term in product (not shown here).

Ensemble Filter Algorithms:

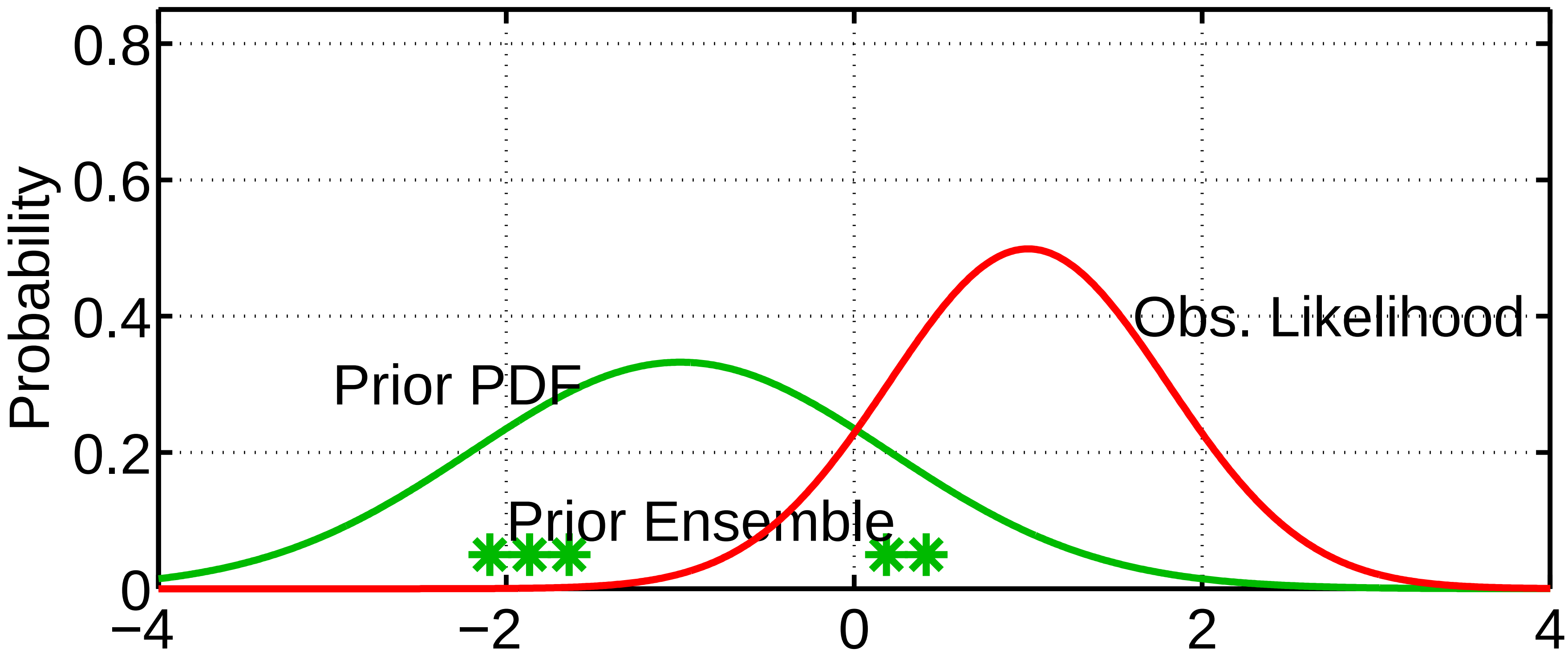
5. Ensemble Kernel filter.



Usually, kernel widths are a function of the sample variance.

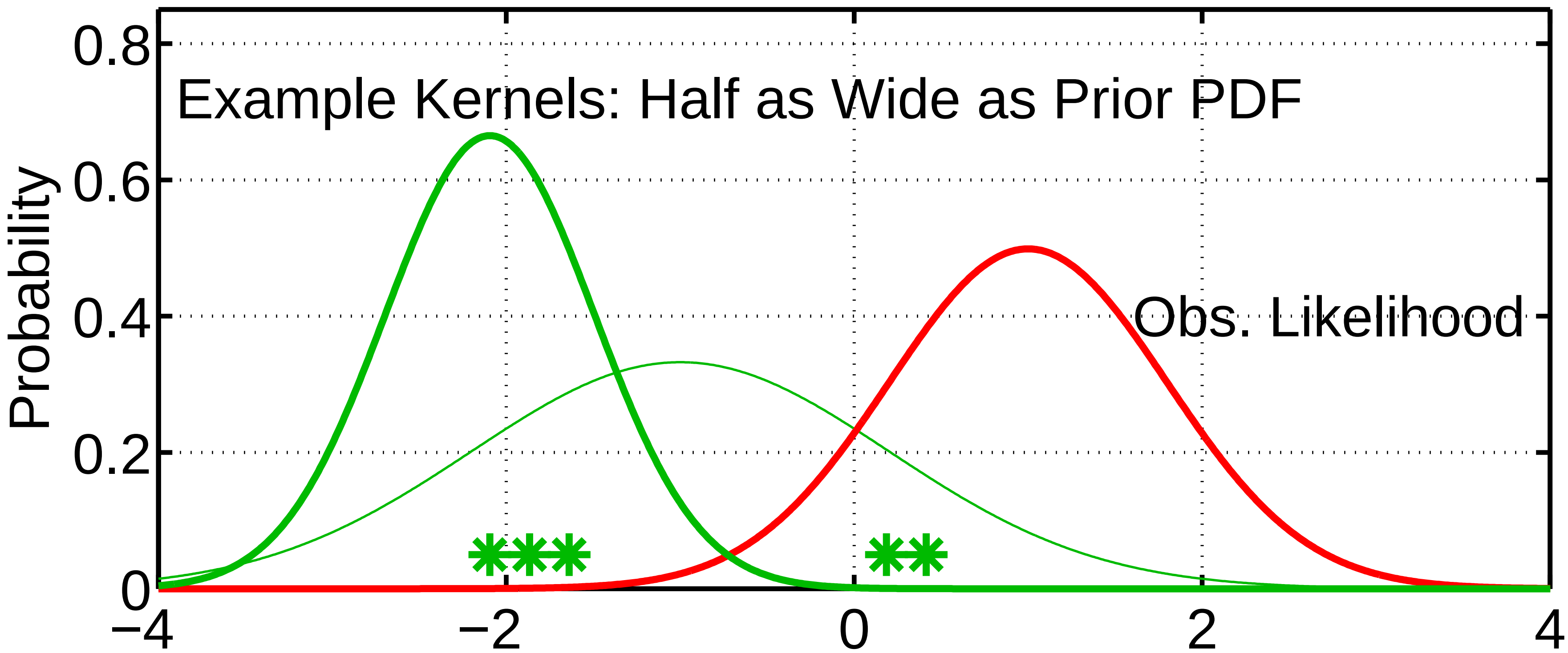
Almost avoids using prior sample variance.

5. Ensemble Kernel filter.



Usually, kernel widths are a function of the sample variance.
Almost avoids using prior sample variance.

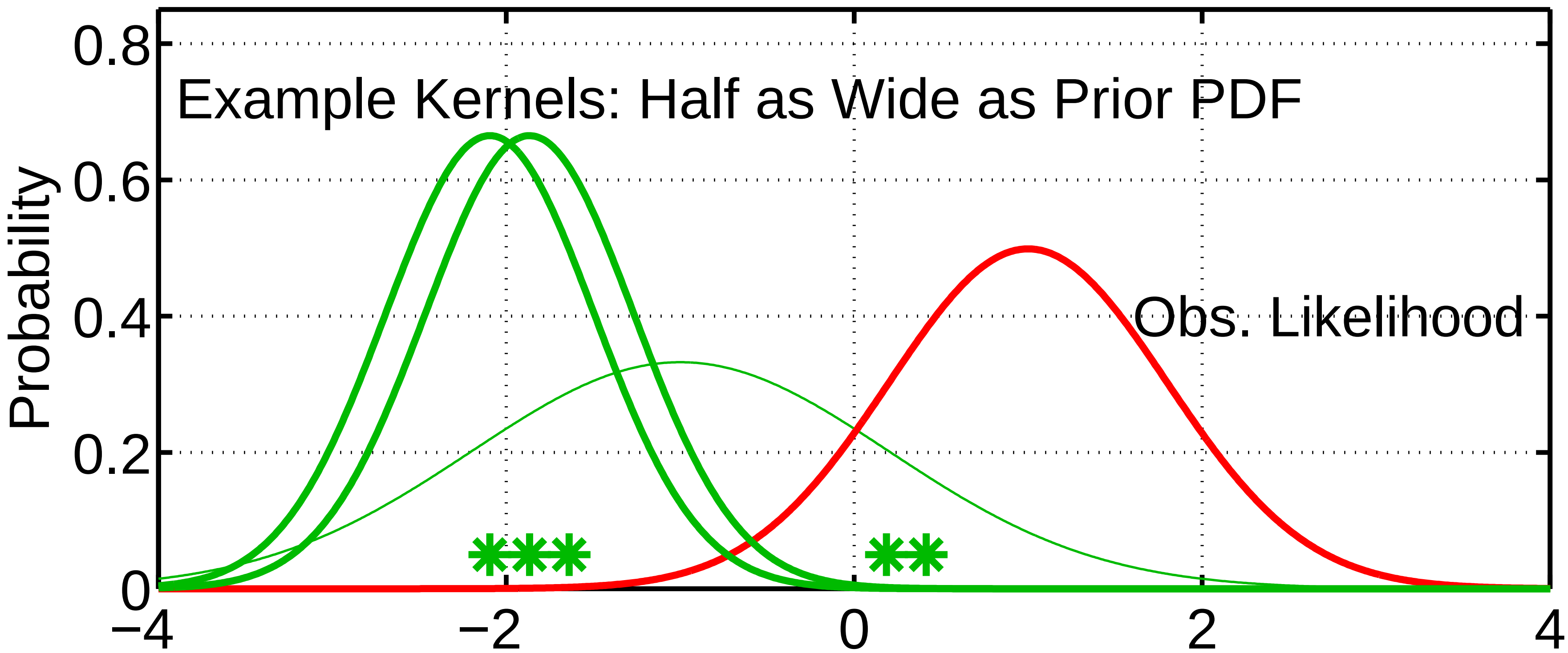
5. Ensemble Kernel filter.



Approximate prior as sum of Gaussians centered on each sample.

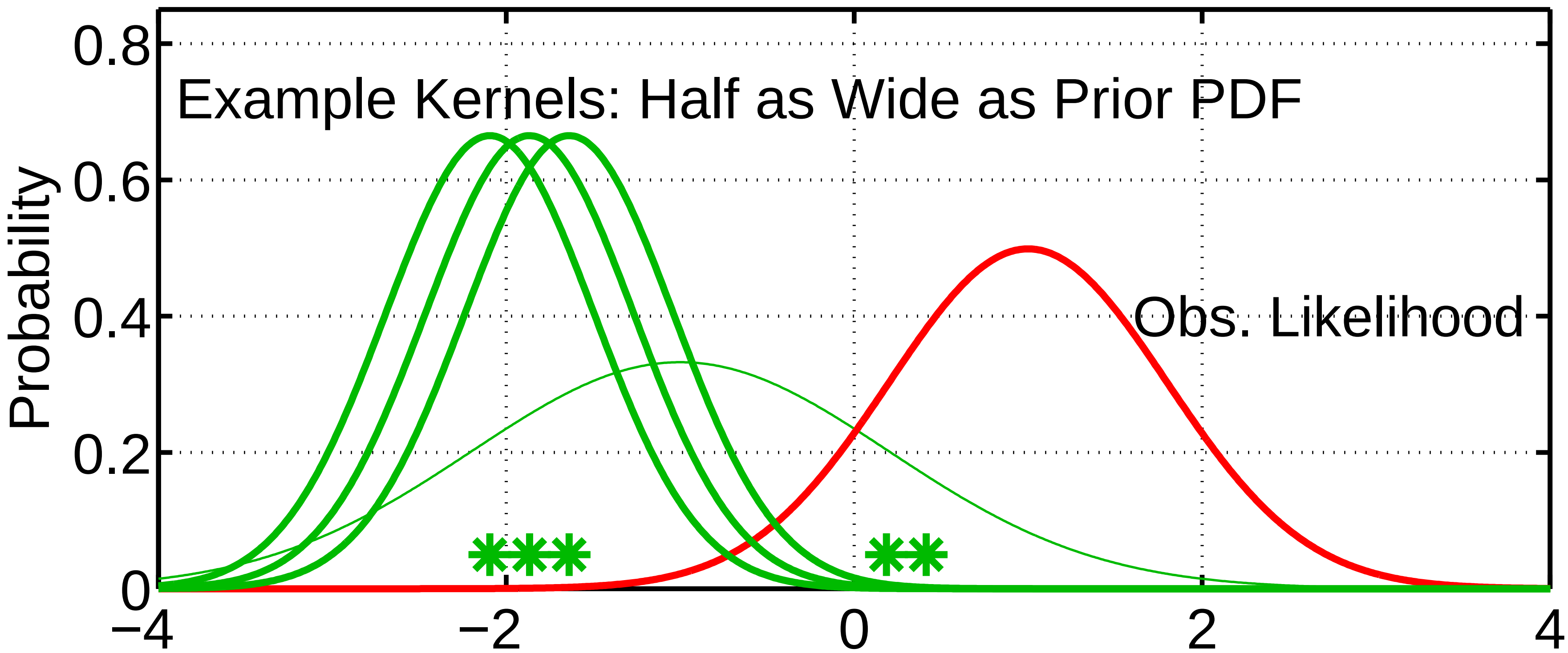
Ensemble Filter Algorithms:

5. Ensemble Kernel filter.



Approximate prior as sum of Gaussians centered on each sample.

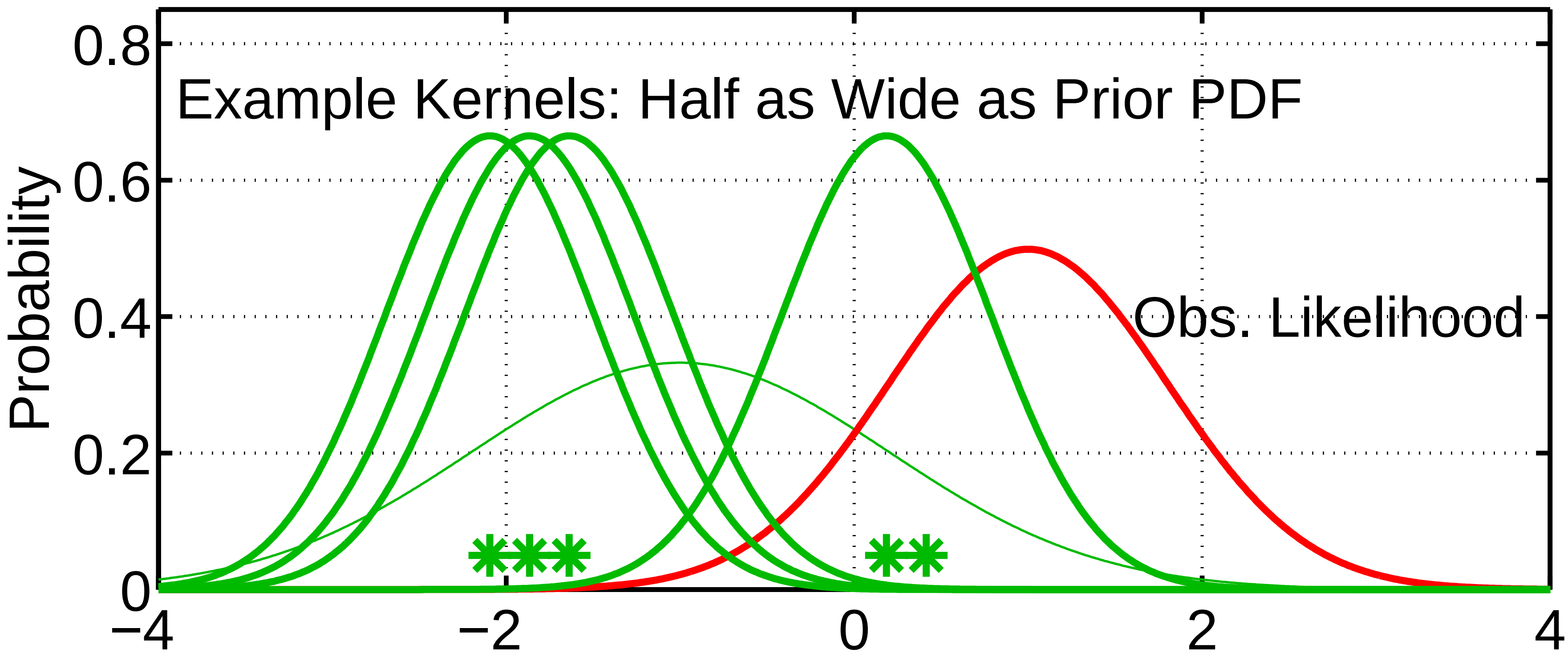
5. Ensemble Kernel filter.



Approximate prior as sum of Gaussians centered on each sample.

Ensemble Filter Algorithms:

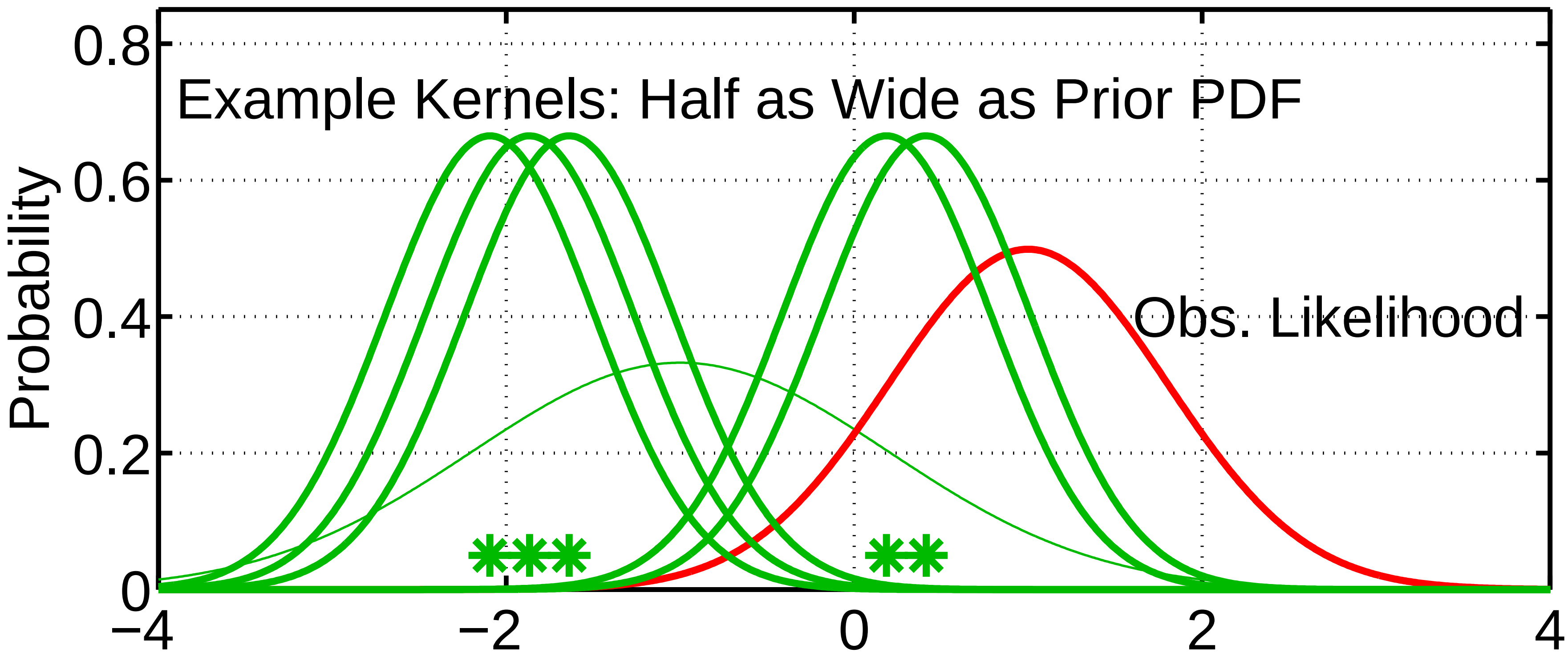
5. Ensemble Kernel filter.



Approximate prior as sum of Gaussians centered on each sample.

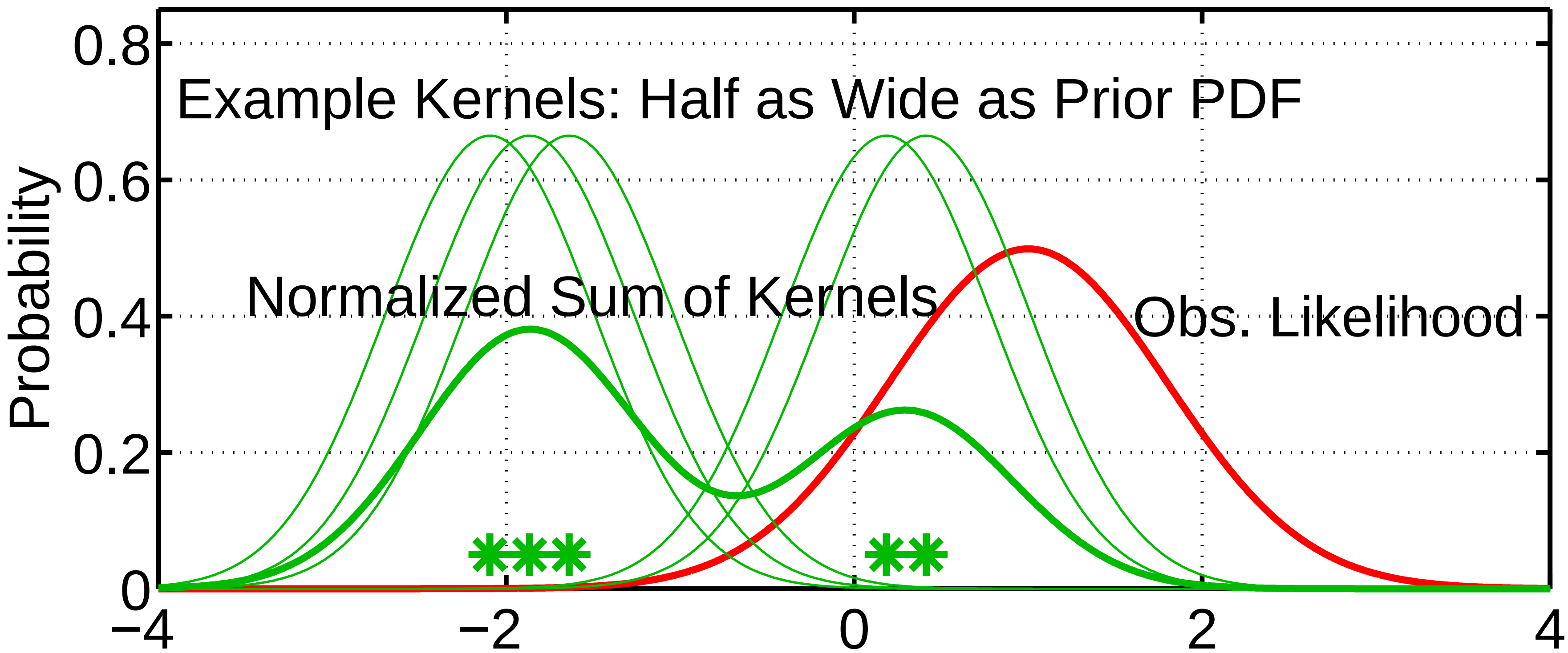
Ensemble Filter Algorithms:

5. Ensemble Kernel filter.



Approximate prior as sum of Gaussians centered on each sample.

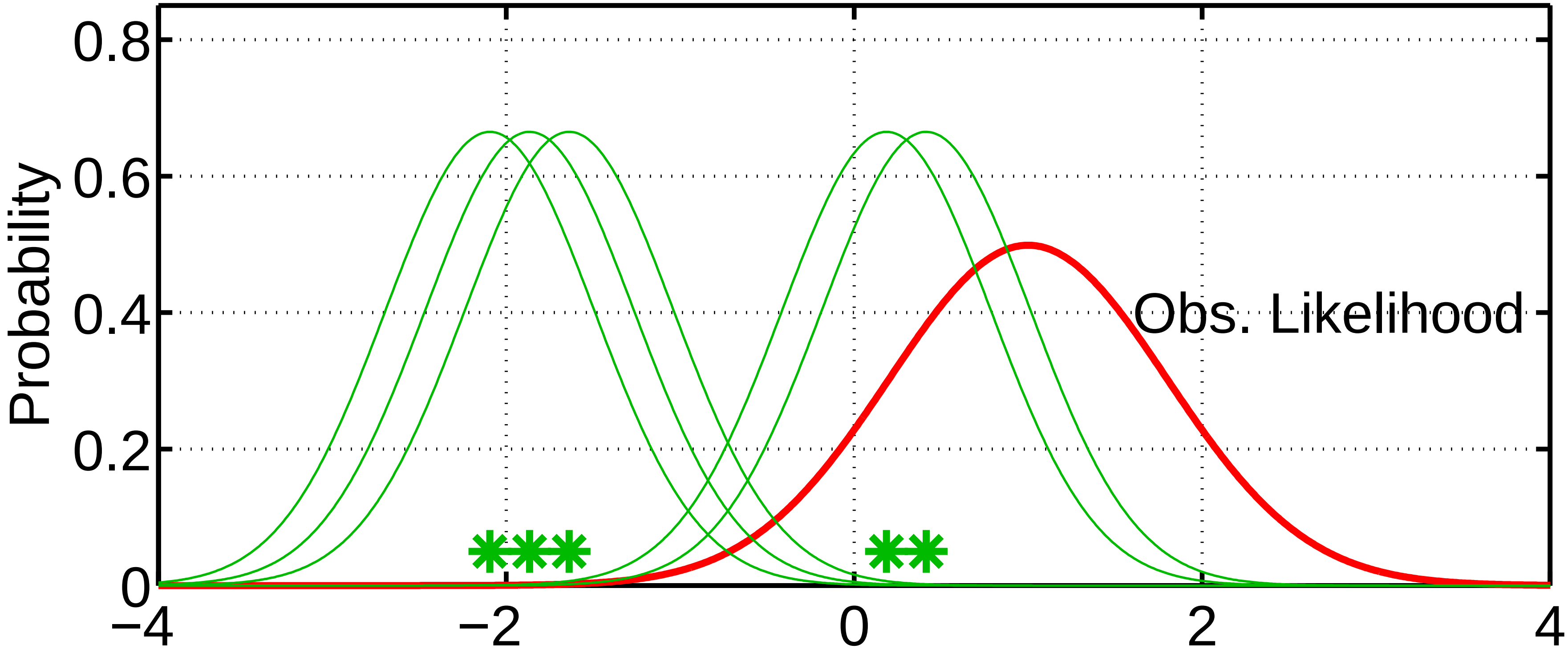
5. Ensemble Kernel filter.



Estimate of prior is normalized sum of all kernels.

Ensemble Filter Algorithms:

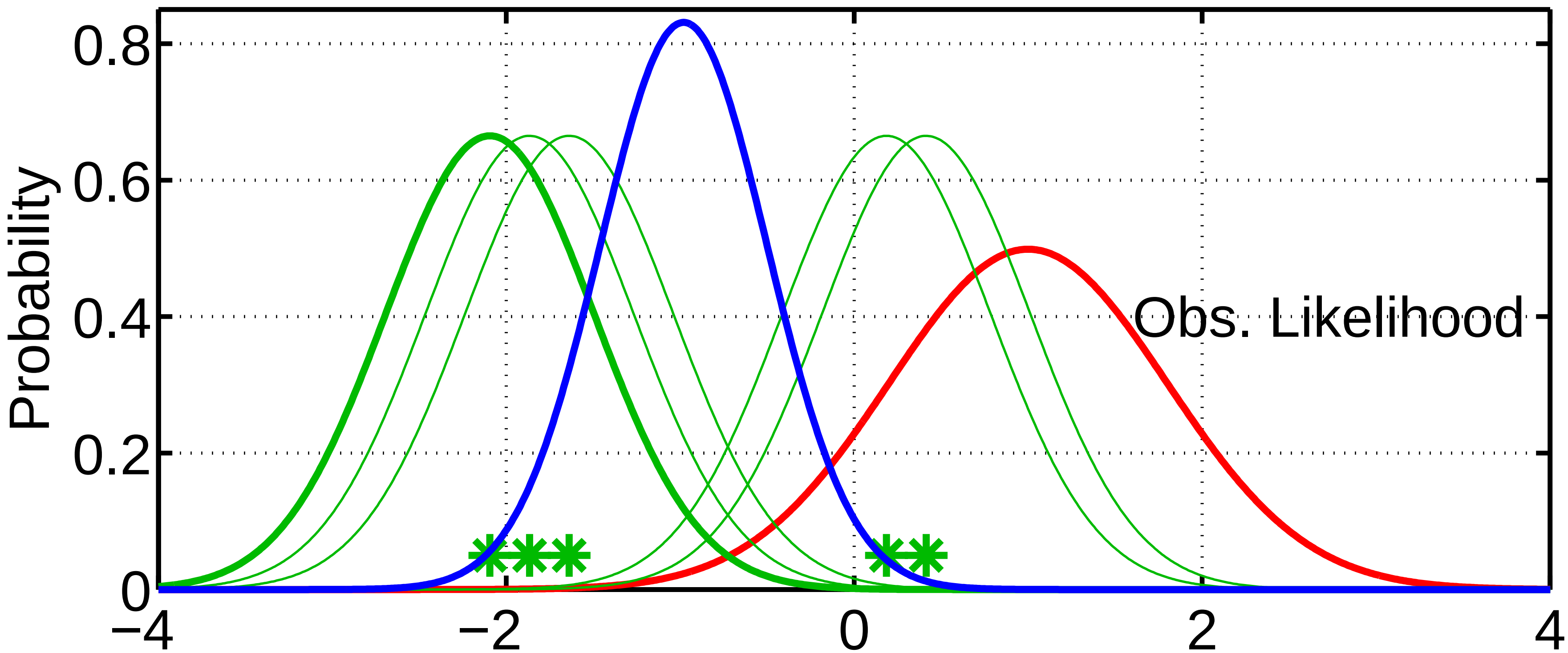
5. Ensemble Kernel filter.



Apply distributive law to take product.

Product of sum is sum of products.

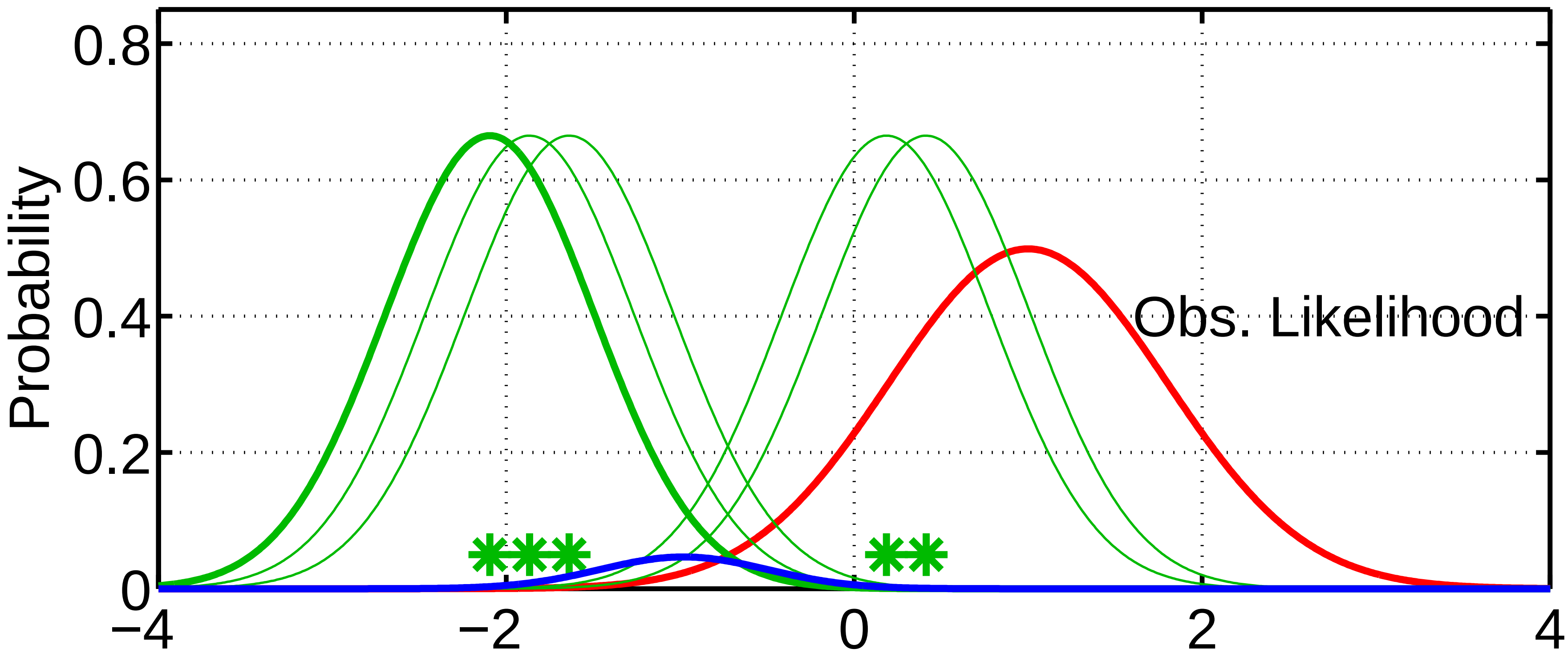
5. Ensemble Kernel filter.



Compute product of first kernel with Obs. Likelihood.

Ensemble Filter Algorithms:

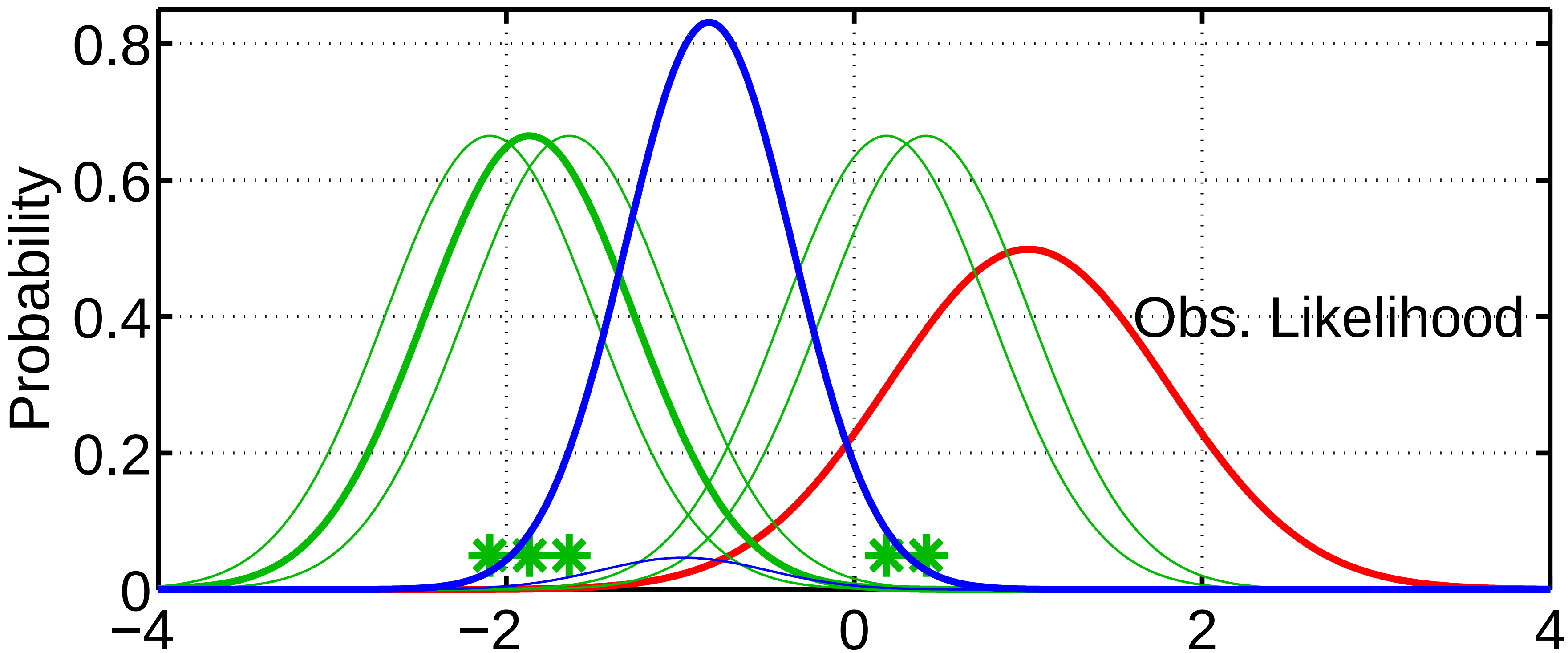
5. Ensemble Kernel filter.



But, can no longer ignore the weight term for product of Gaussians.
Kernels with mean further from observation get less weight.

Ensemble Filter Algorithms:

5. Ensemble Kernel filter.

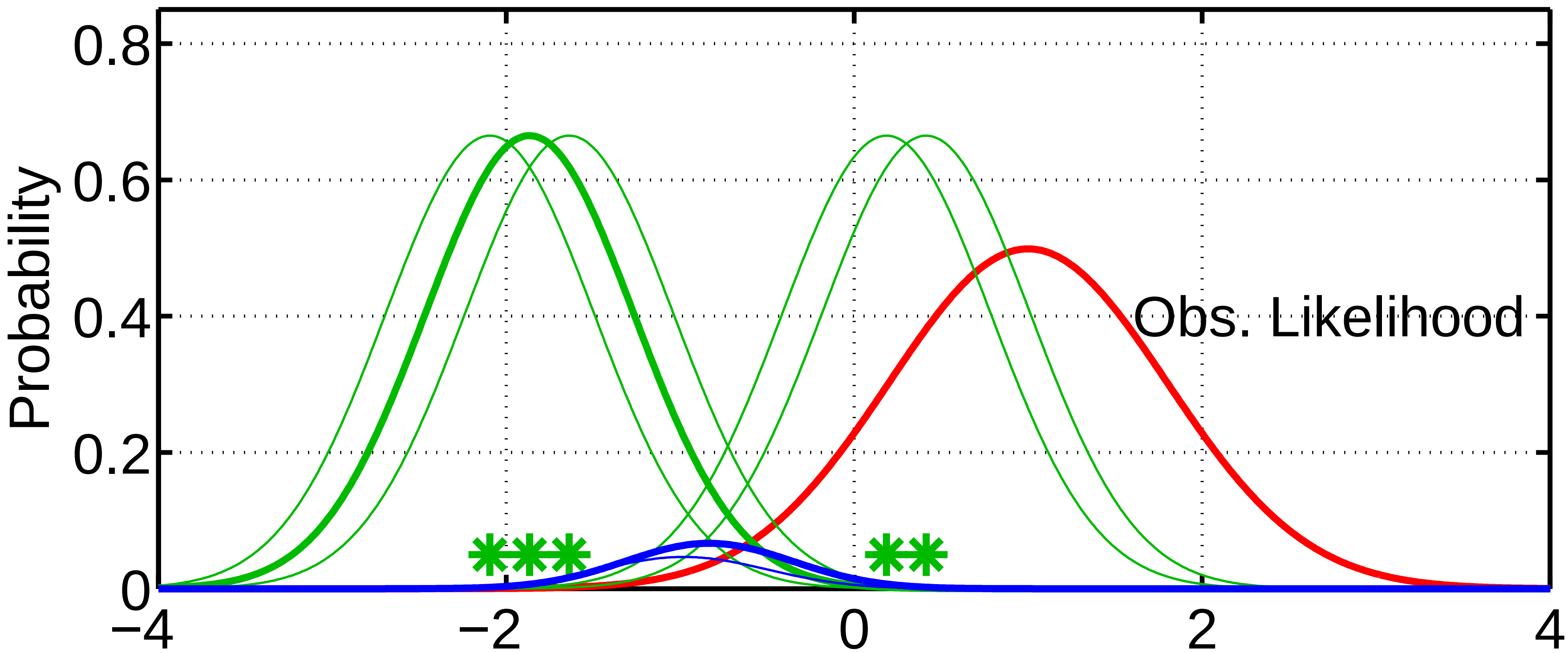


Continue to take products for each kernel in turn.

More distant kernels have small impact on posterior.

Ensemble Filter Algorithms:

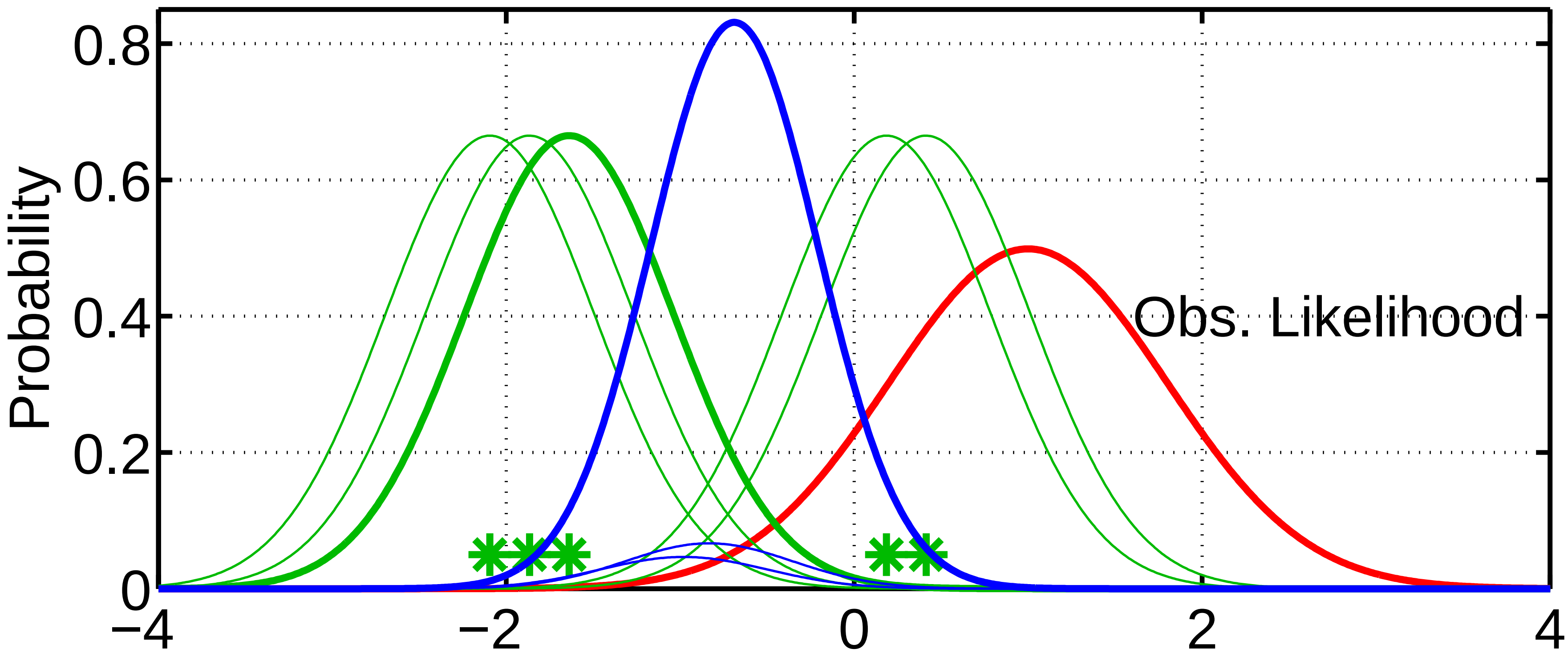
5. Ensemble Kernel filter.



Continue to take products for each kernel in turn.
More distant kernels have small impact on posterior.

Ensemble Filter Algorithms:

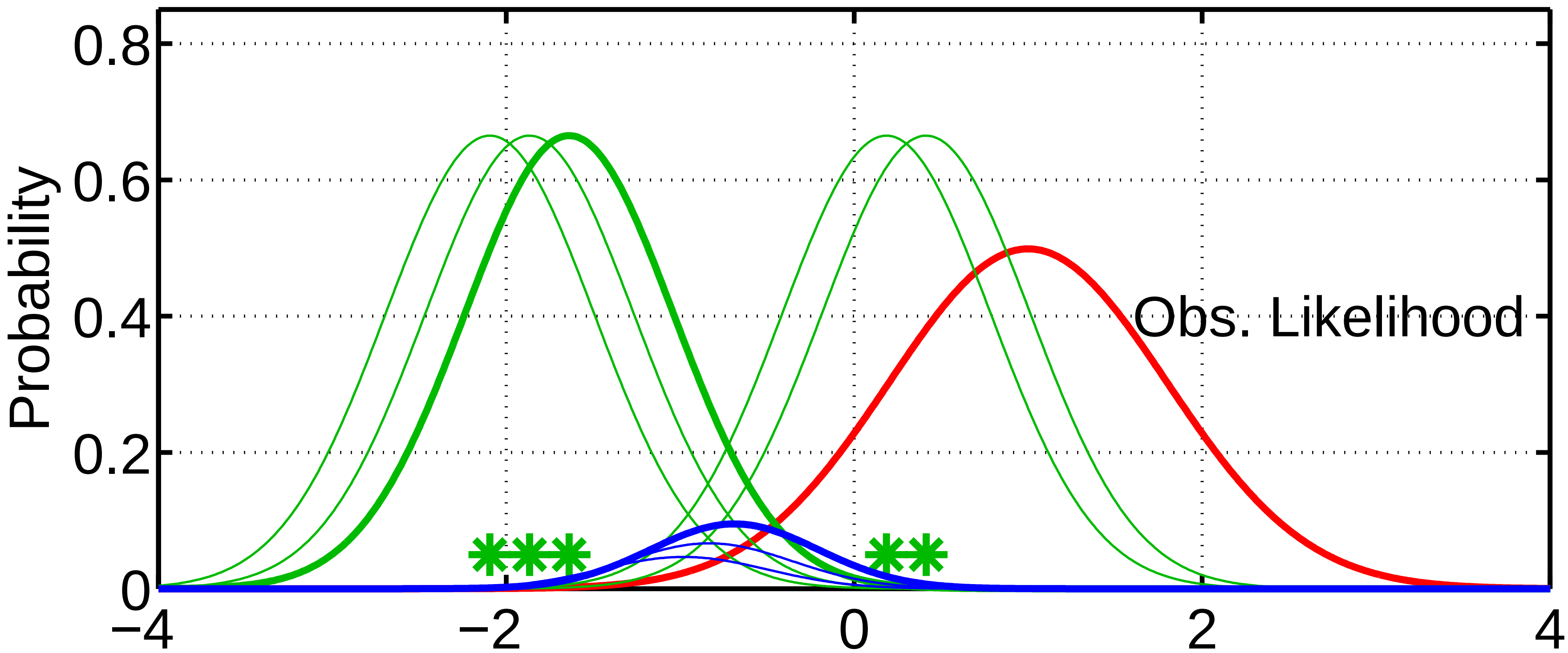
5. Ensemble Kernel filter.



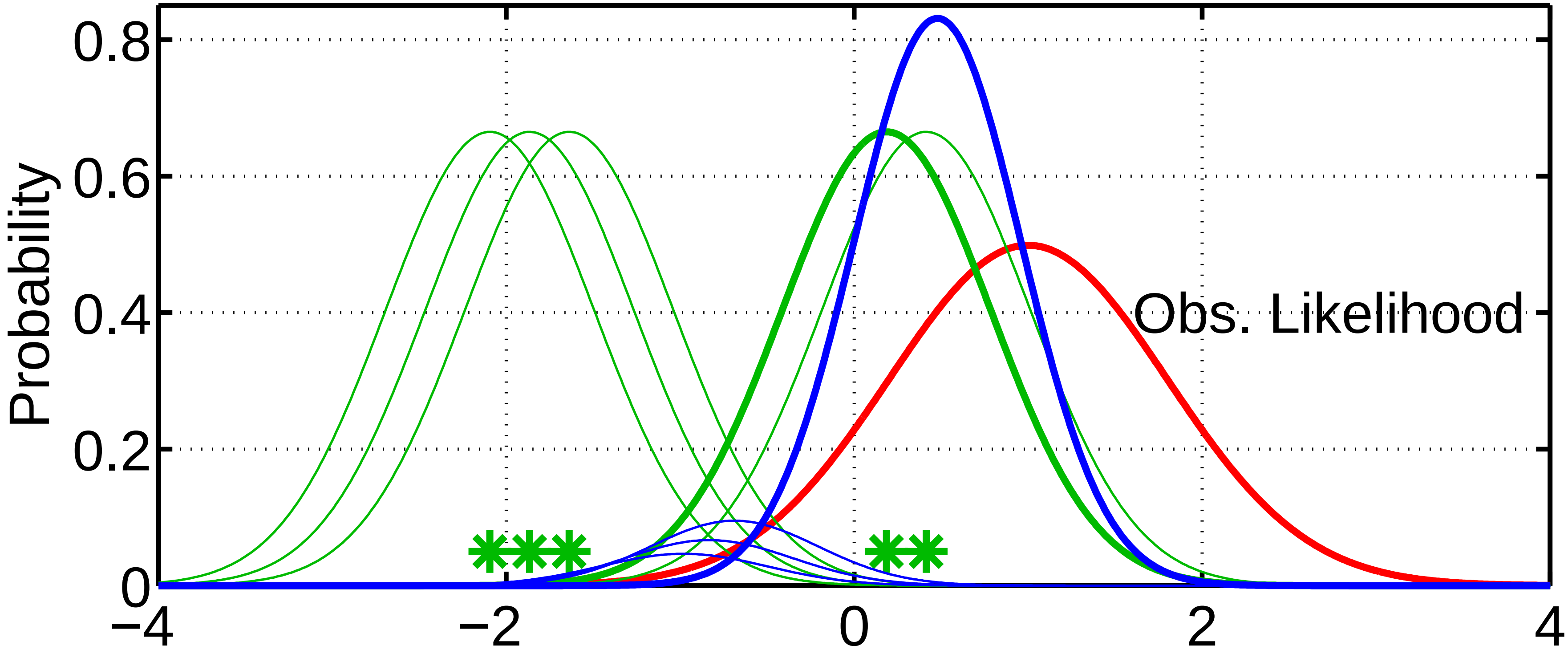
Continue to take products for each kernel in turn.
More distant kernels have small impact on posterior.

Ensemble Filter Algorithms:

5. Ensemble Kernel filter.



5. Ensemble Kernel filter.

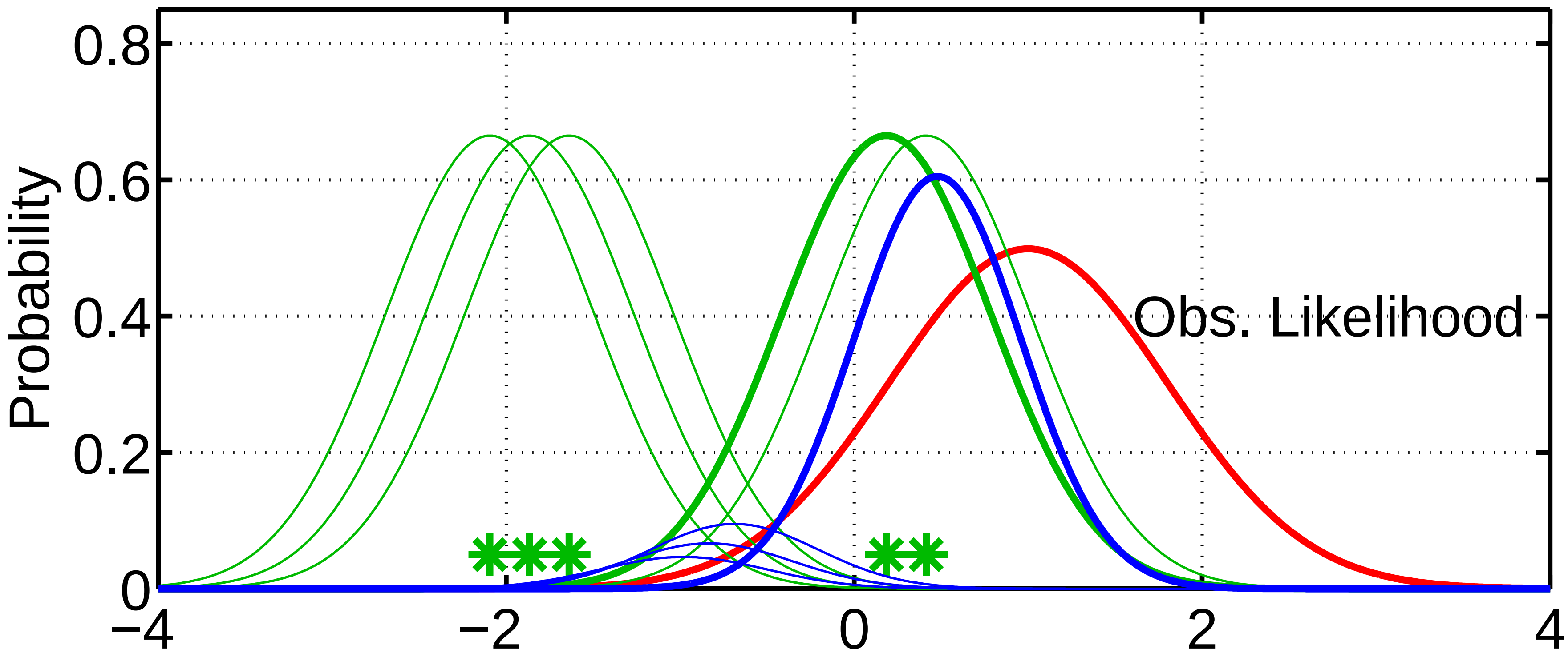


Continue to take products for each kernel in turn.

Closer kernels dominate posterior.

Ensemble Filter Algorithms:

5. Ensemble Kernel filter.

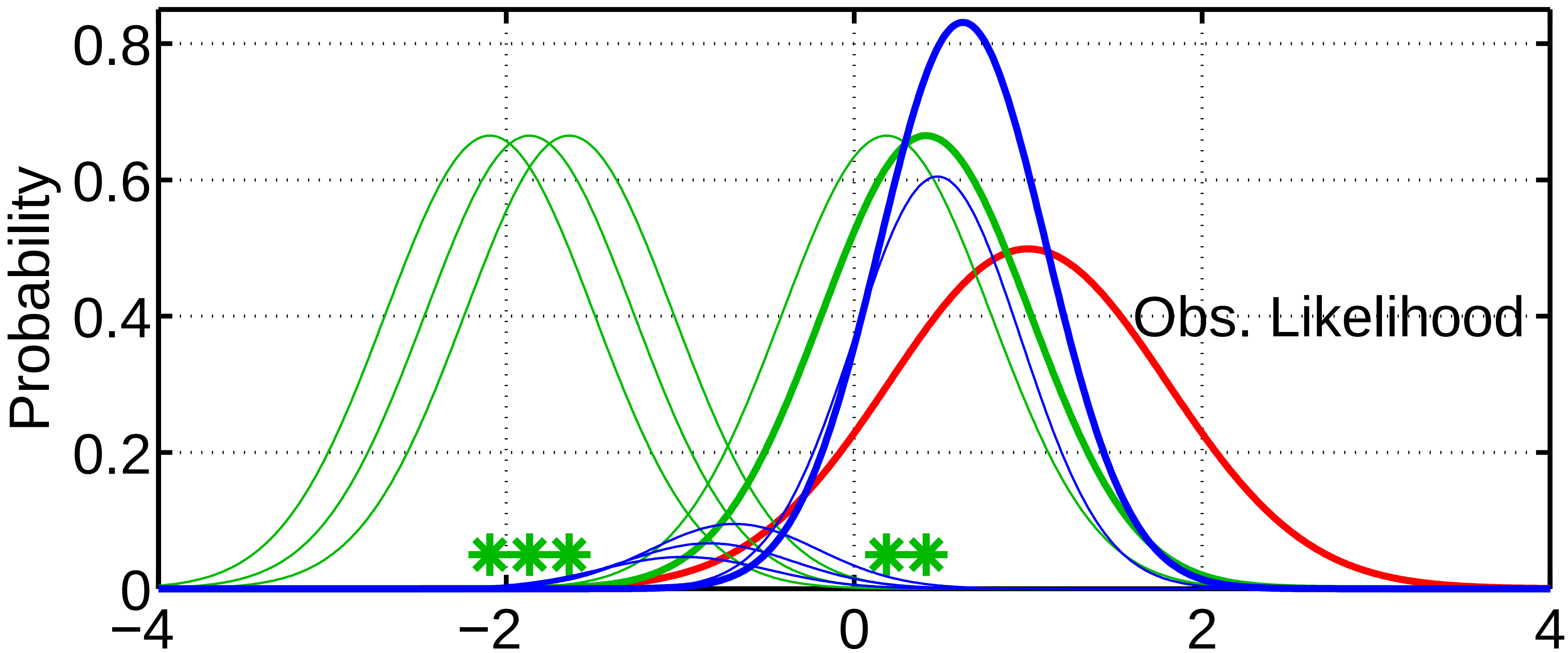


Continue to take products for each kernel in turn.

Closer kernels dominate posterior.

Ensemble Filter Algorithms:

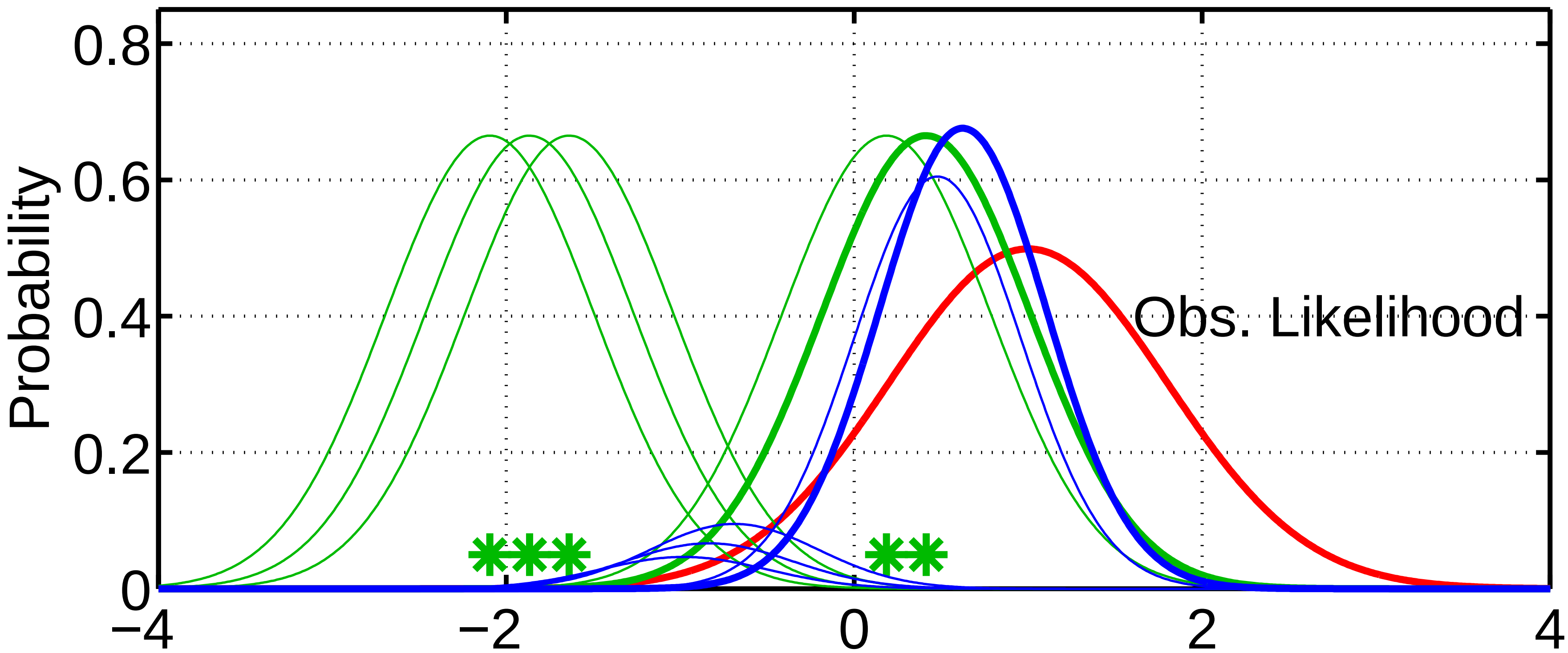
5. Ensemble Kernel filter.



Continue to take products for each kernel in turn.

Closer kernels dominate posterior.

5. Ensemble Kernel filter.

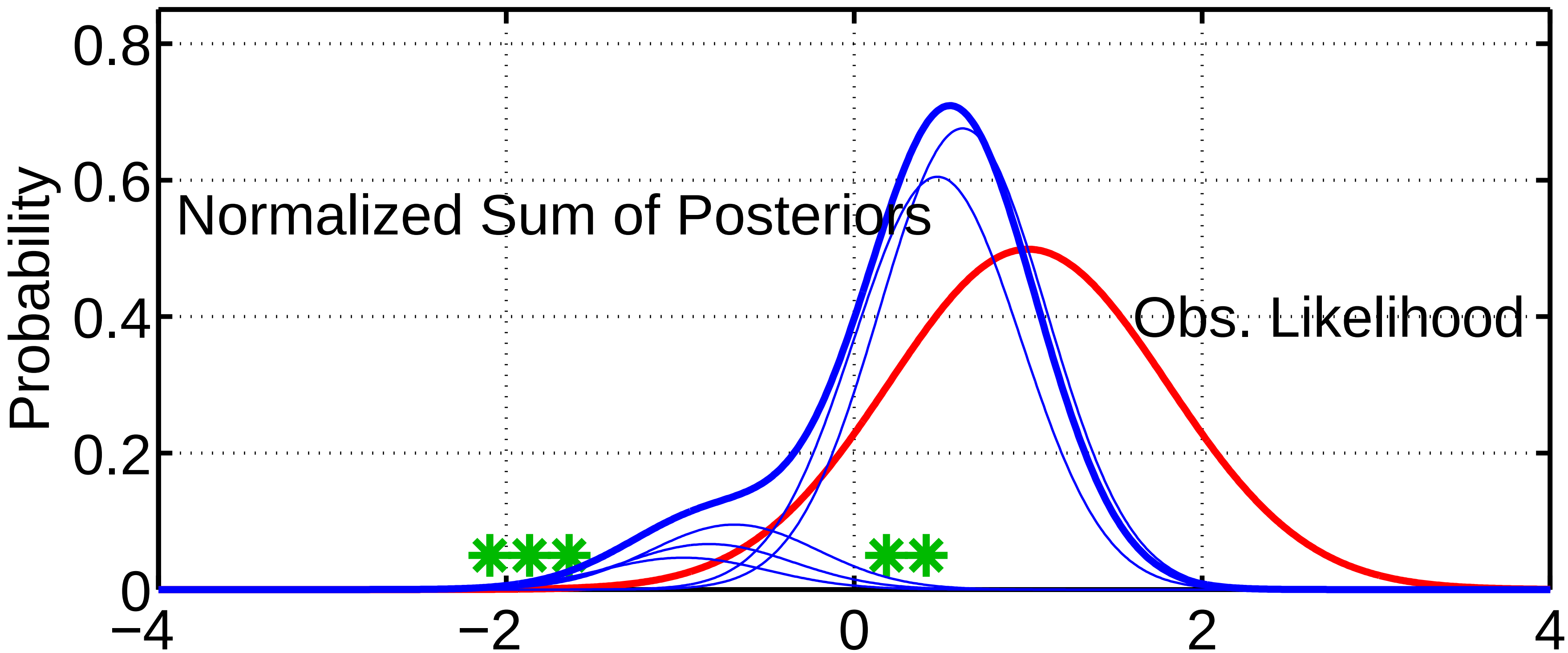


Continue to take products for each kernel in turn.

Closer kernels dominate posterior.

Ensemble Filter Algorithms:

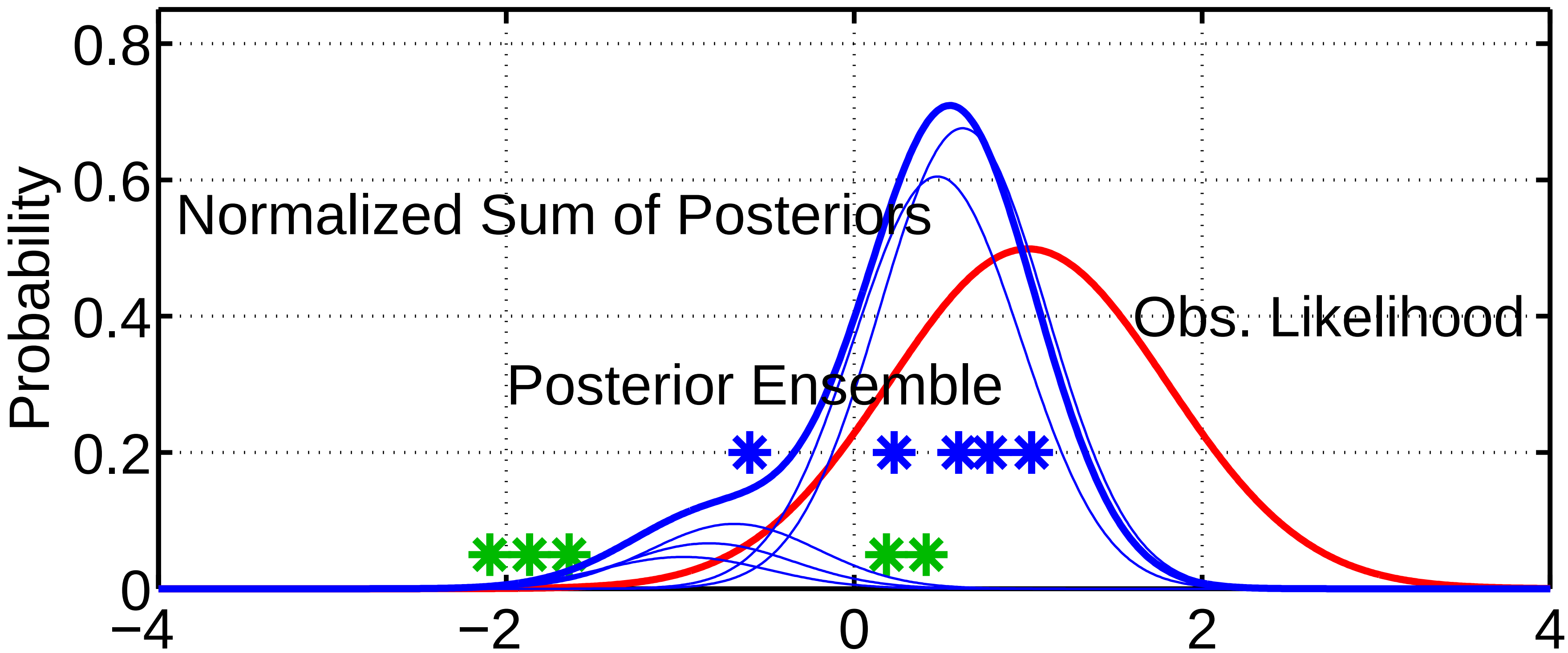
5. Ensemble Kernel filter.



Final posterior is weight-normalized sum of kernel products.

Posterior is somewhat different than for ensemble adjustment or ensemble Kalman filter (much less density in left lobe).

5. Ensemble Kernel filter.



Forming sample of the posterior can be problematic.

Random sample is simple.

Deterministic sampling is much more tricky here (few results available)

Ensemble Filter Algorithms:

6. Particle filter methods:

These are ‘classical’ ensemble methods from statistical literature.

Size of ensembles required scales hyper-exponentially with model size.

Ensembles > 1000 required for models with > 4 degrees of freedom.

This rules out naive application to any meaningful atmospheric model.

At present, nobody knows ways to attack this so no details here.

Phase 2: Single observed variable, single unobserved variable

So far, have known observation likelihood for single variable.

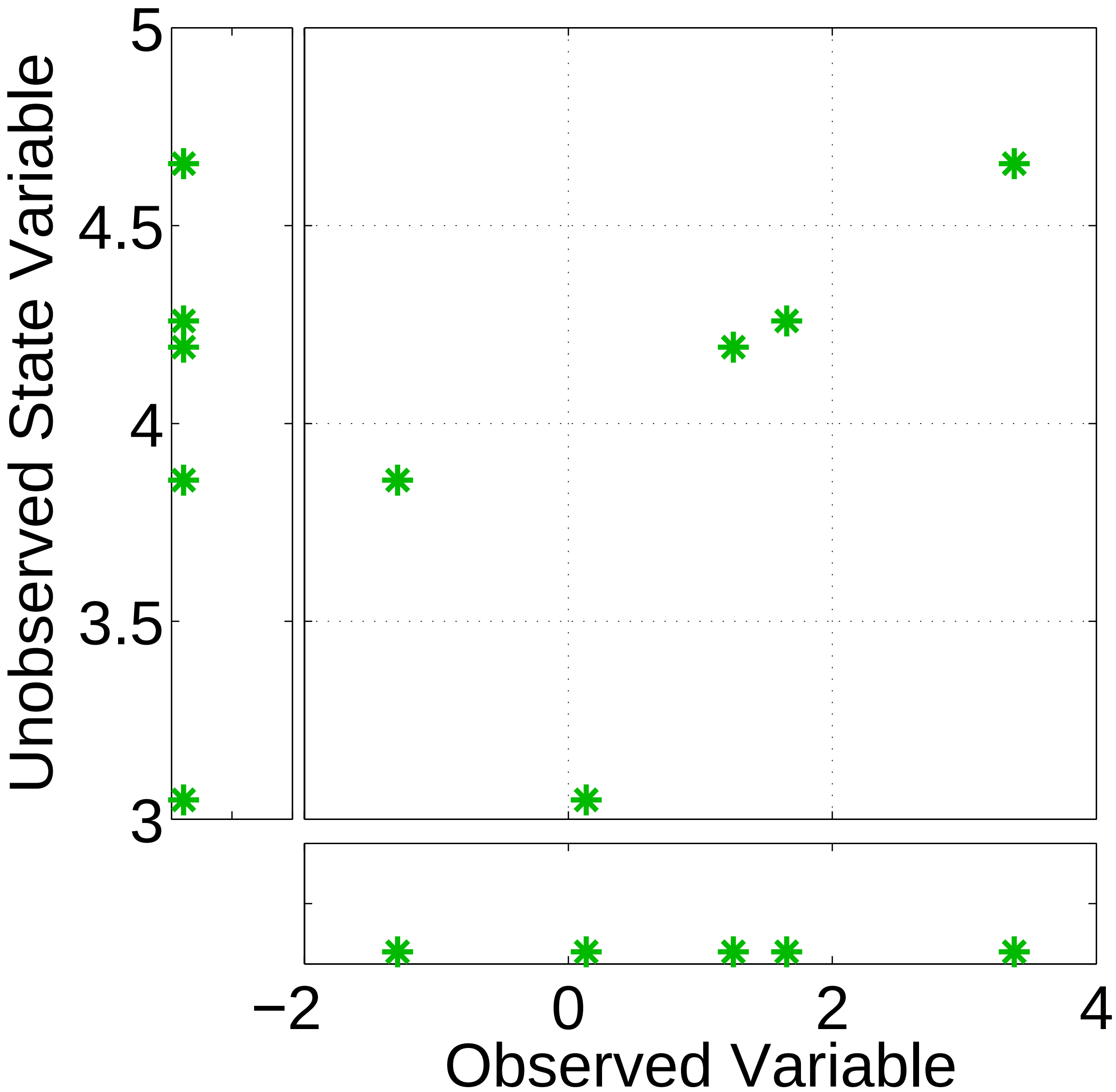
Now, suppose prior has an additional variable.

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

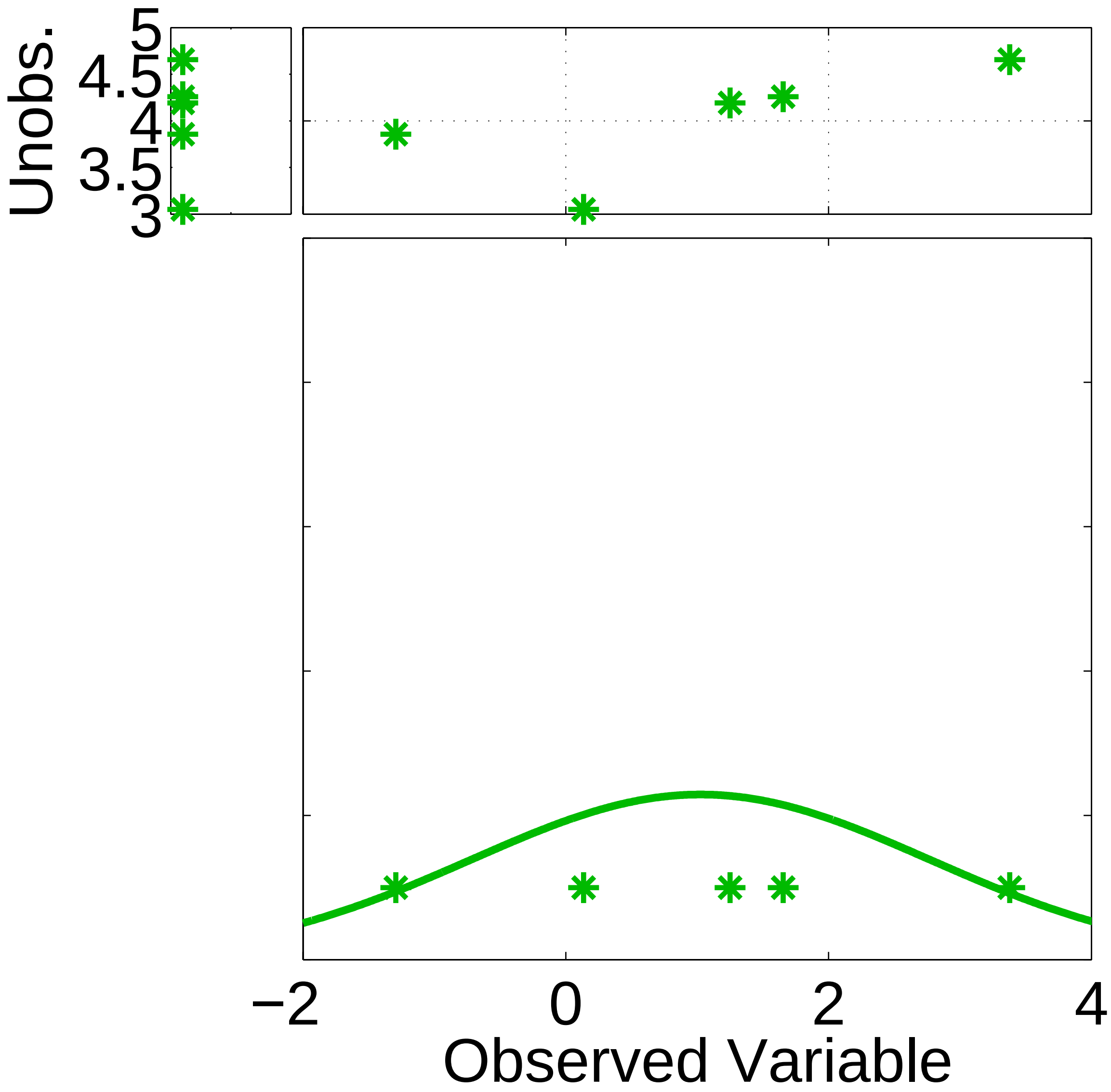


Assume that all we know is prior joint distribution.

One variable is observed.

What should happen to unobserved variable?

Ensemble filters: Updating additional prior state variables

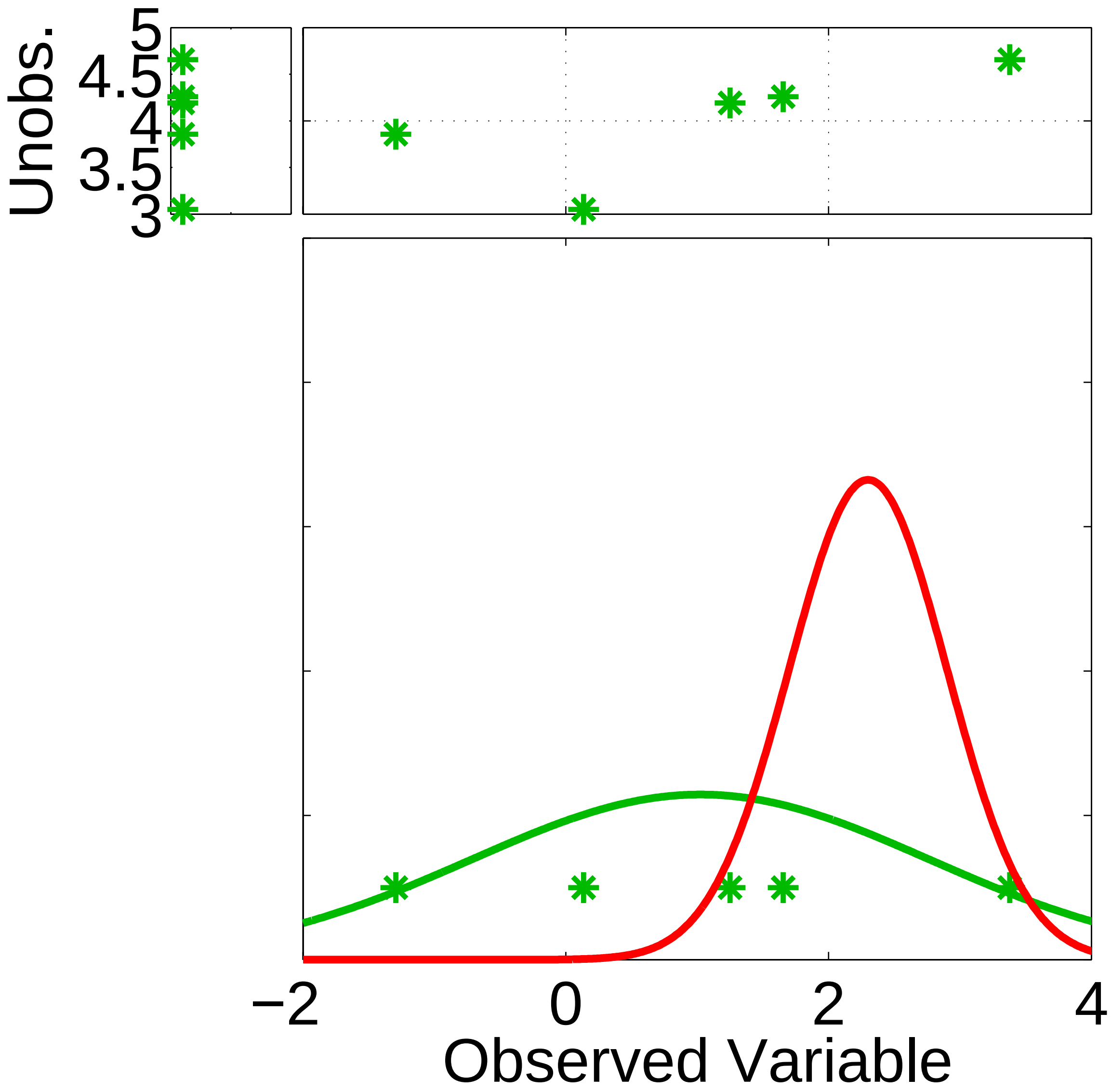


Assume that all we know is prior joint distribution.

One variable is observed.

Update observed variable with one of previous methods.

Ensemble filters: Updating additional prior state variables

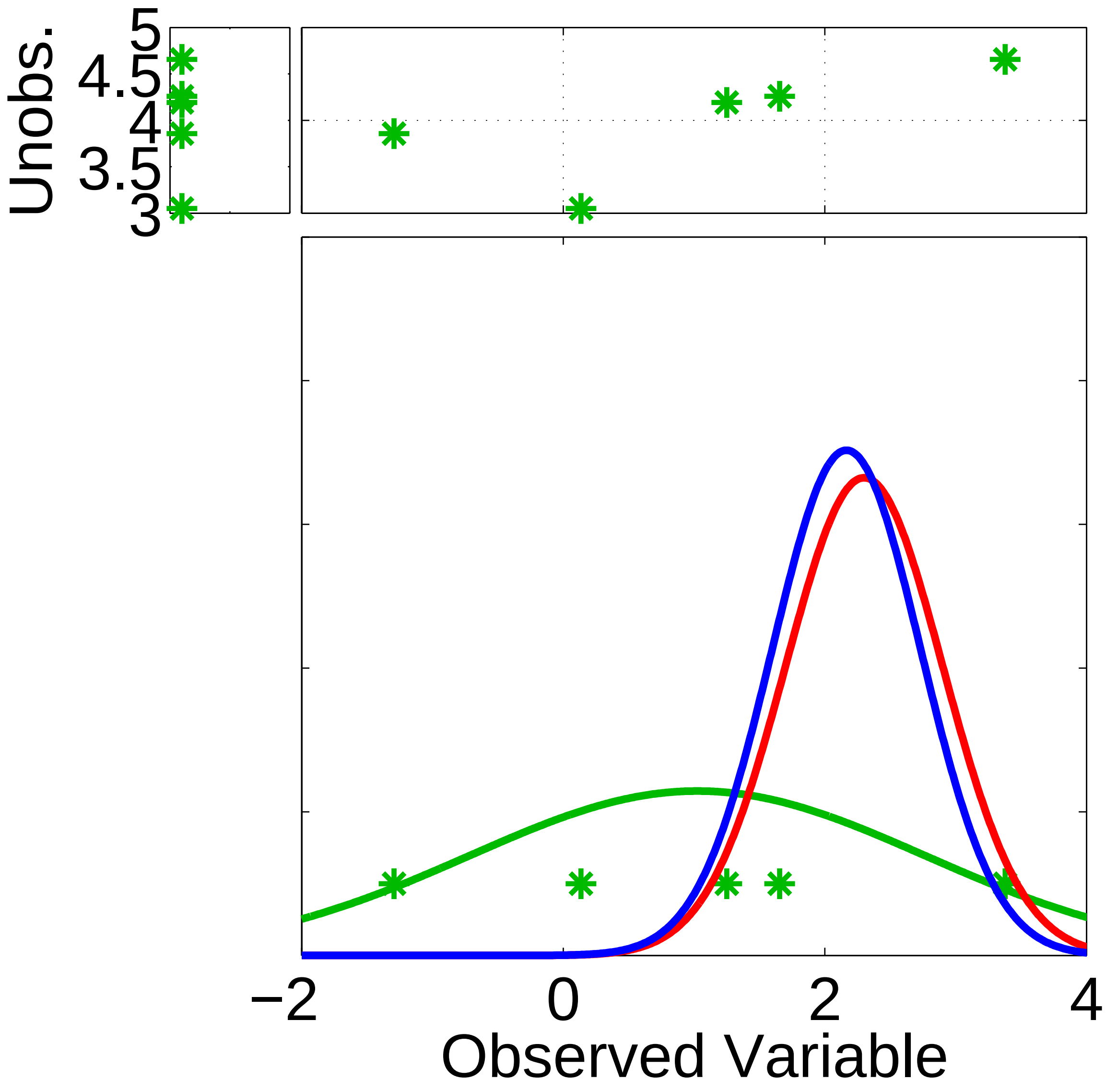


Assume that all we know is prior joint distribution.

One variable is observed.

Update observed variable with one of previous methods.

Ensemble filters: Updating additional prior state variables

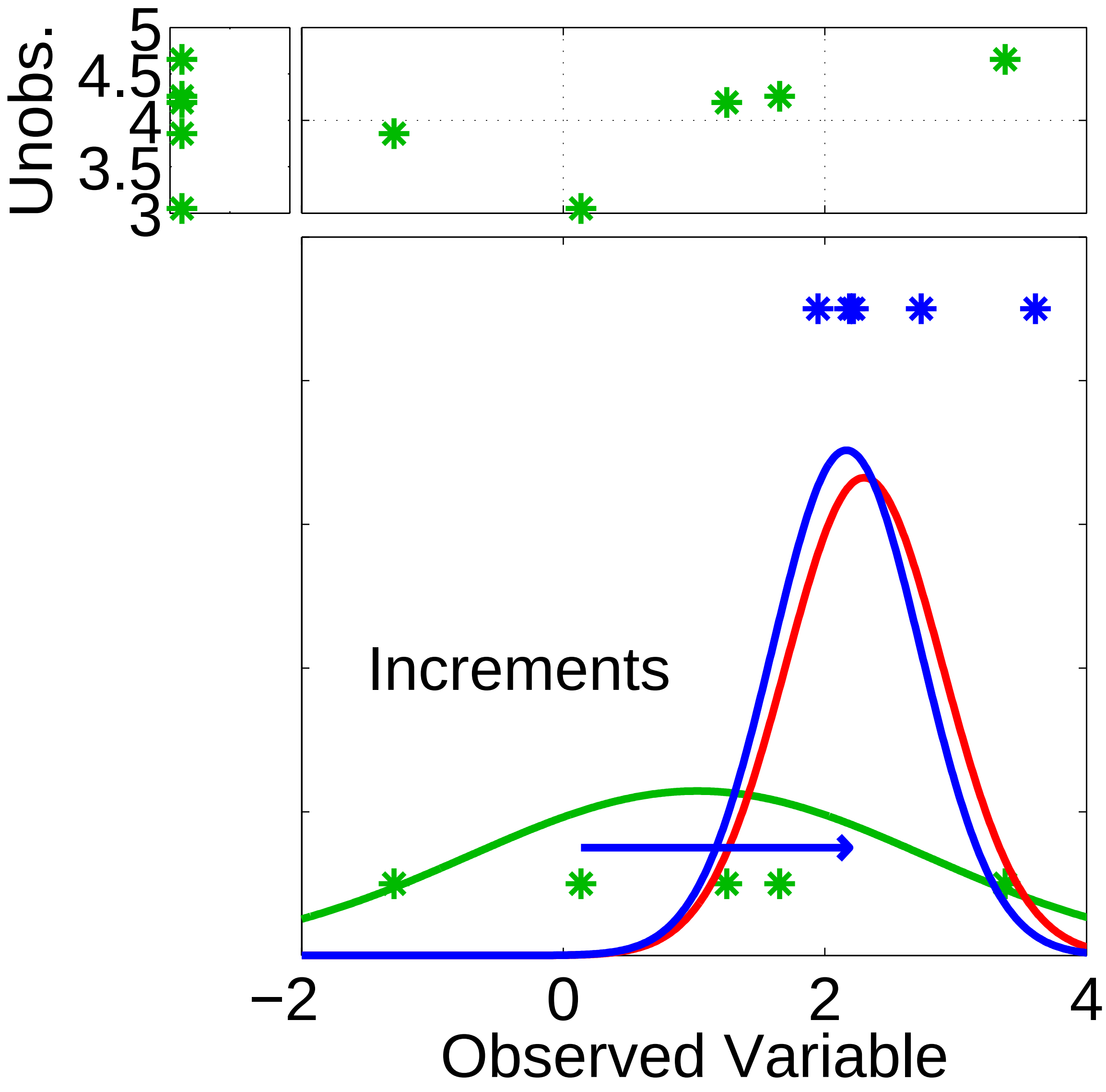


Assume that all we know is prior joint distribution.

One variable is observed.

Update observed variable with one of previous methods.

Ensemble filters: Updating additional prior state variables

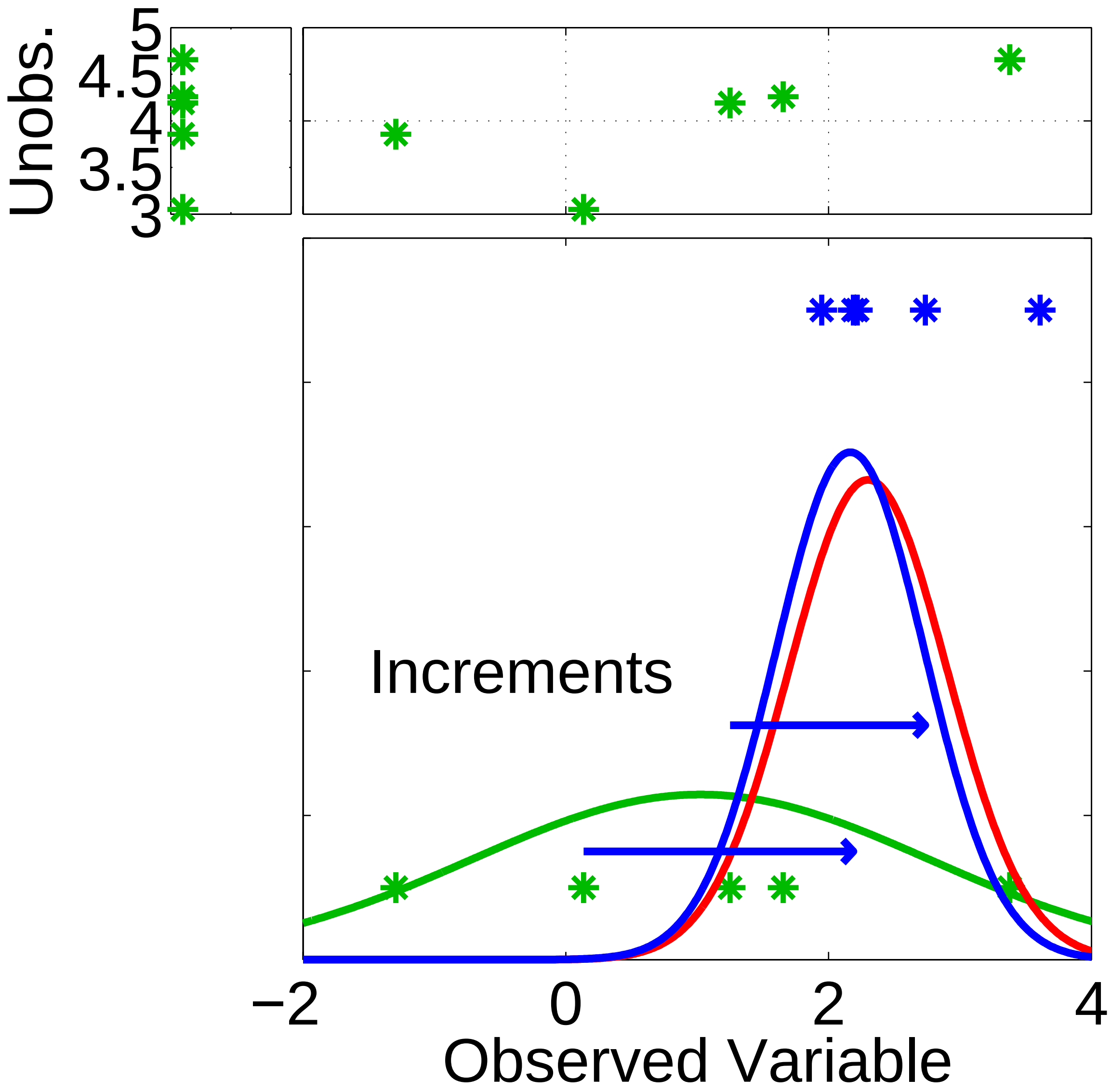


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

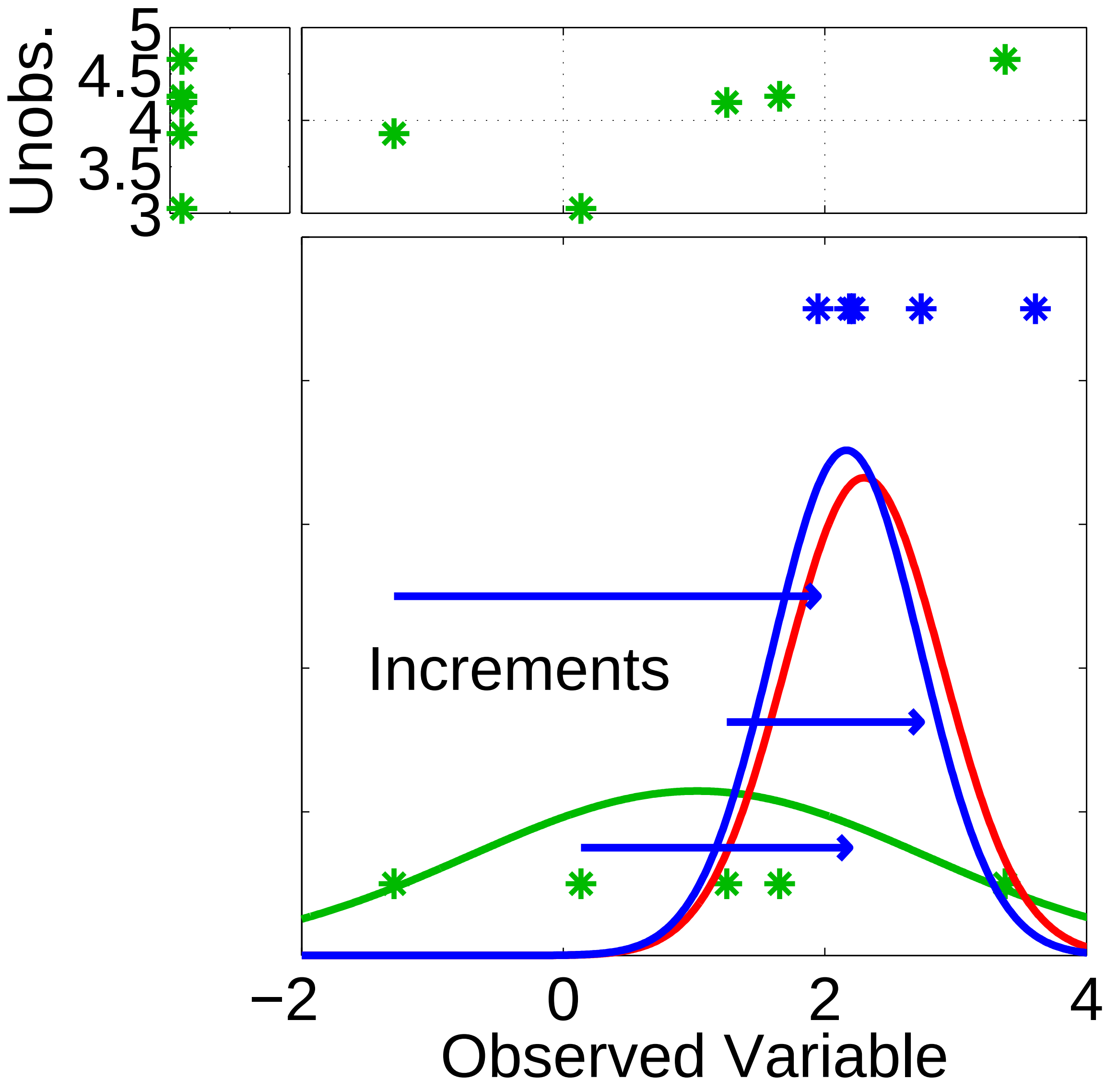


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

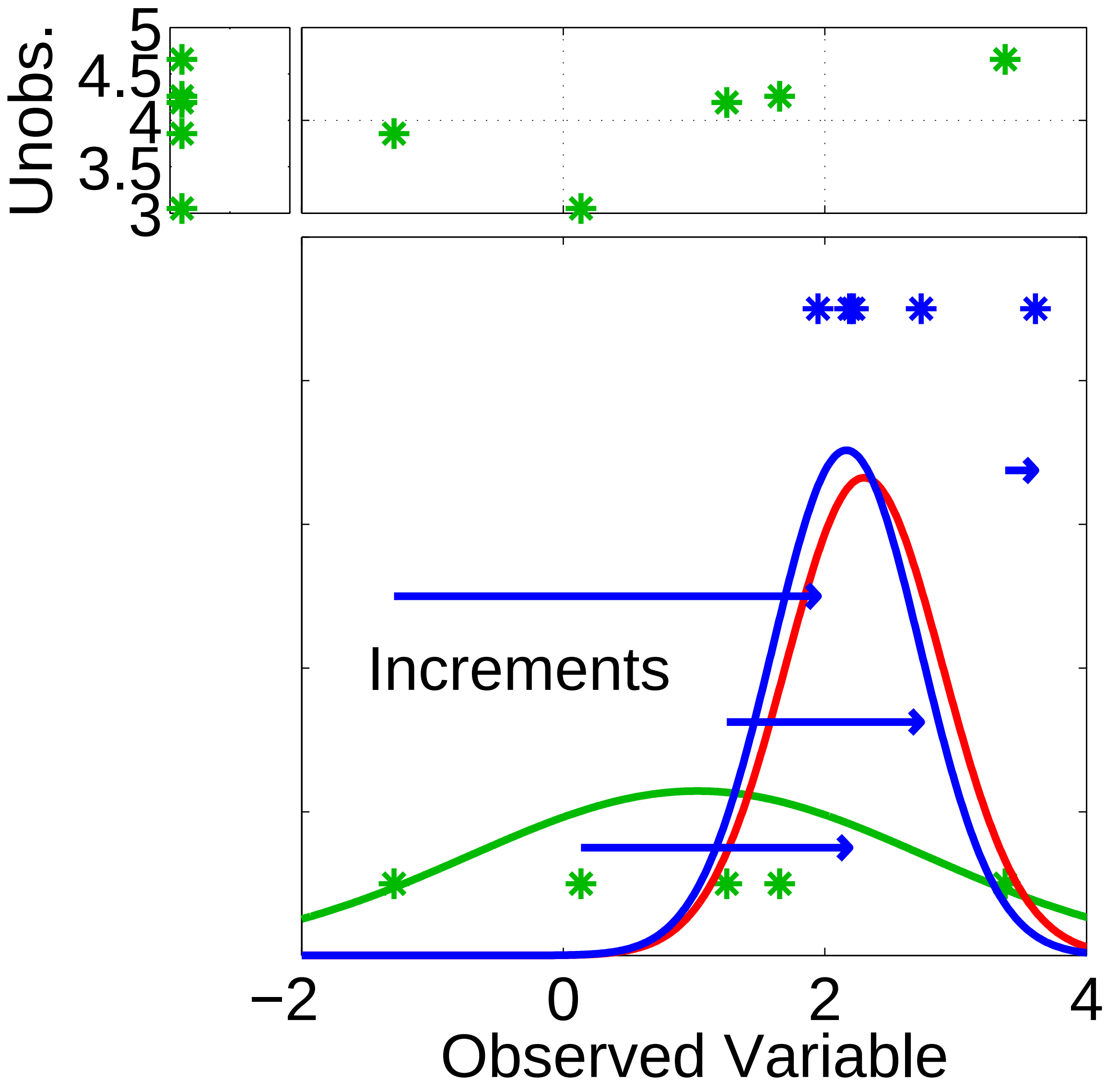


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

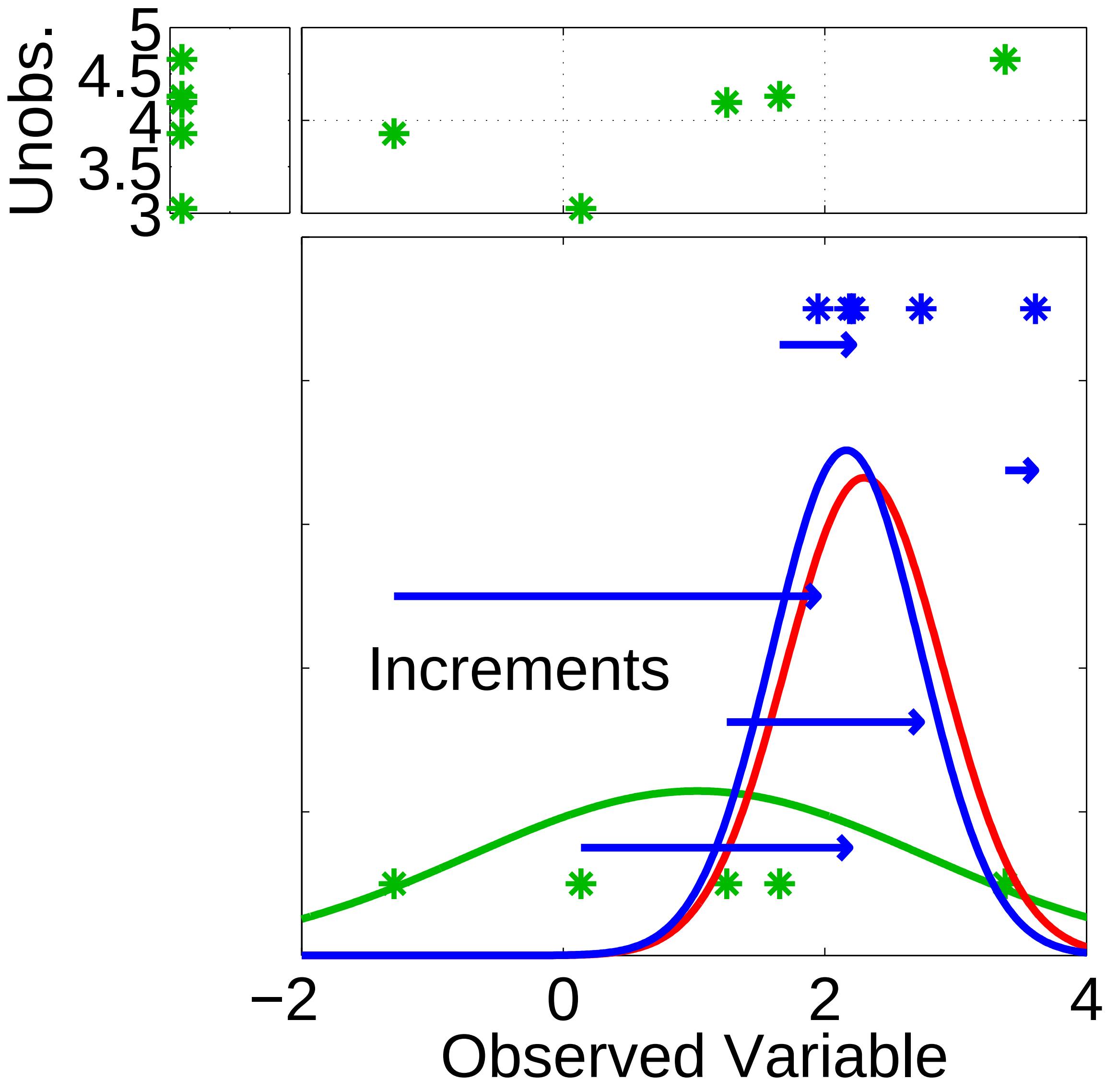


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

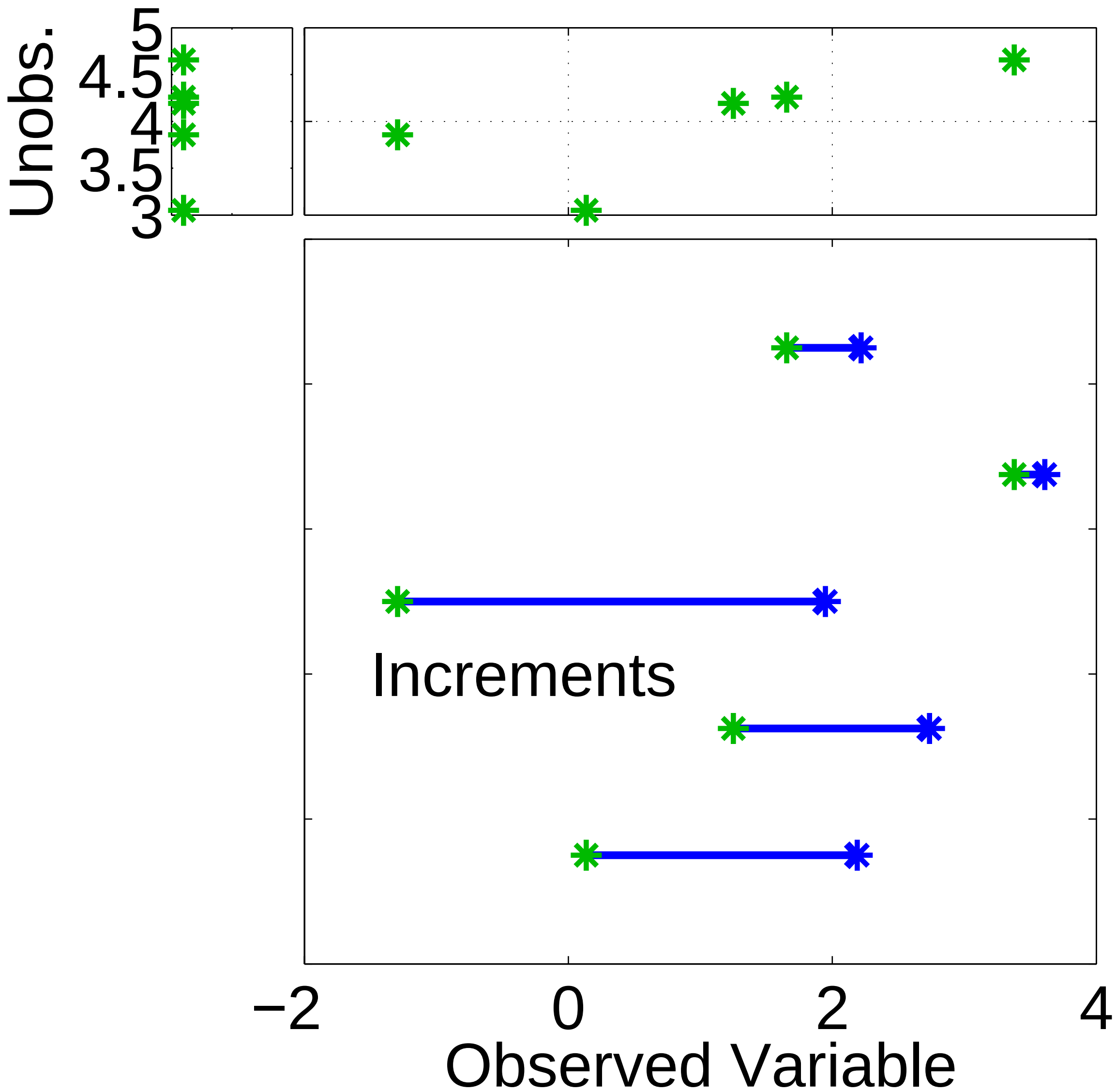


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

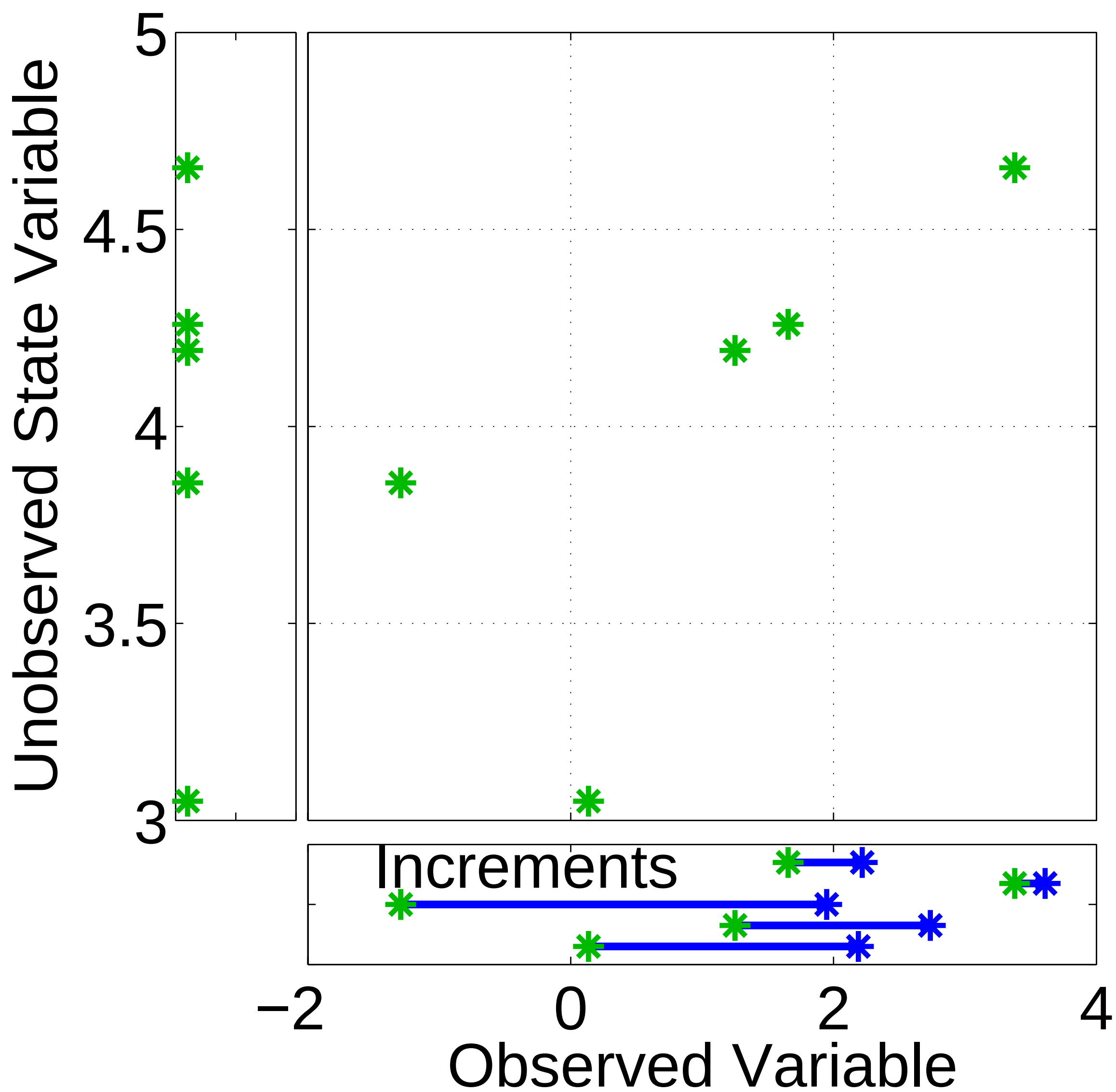


Assume that all we know is prior joint distribution.

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.

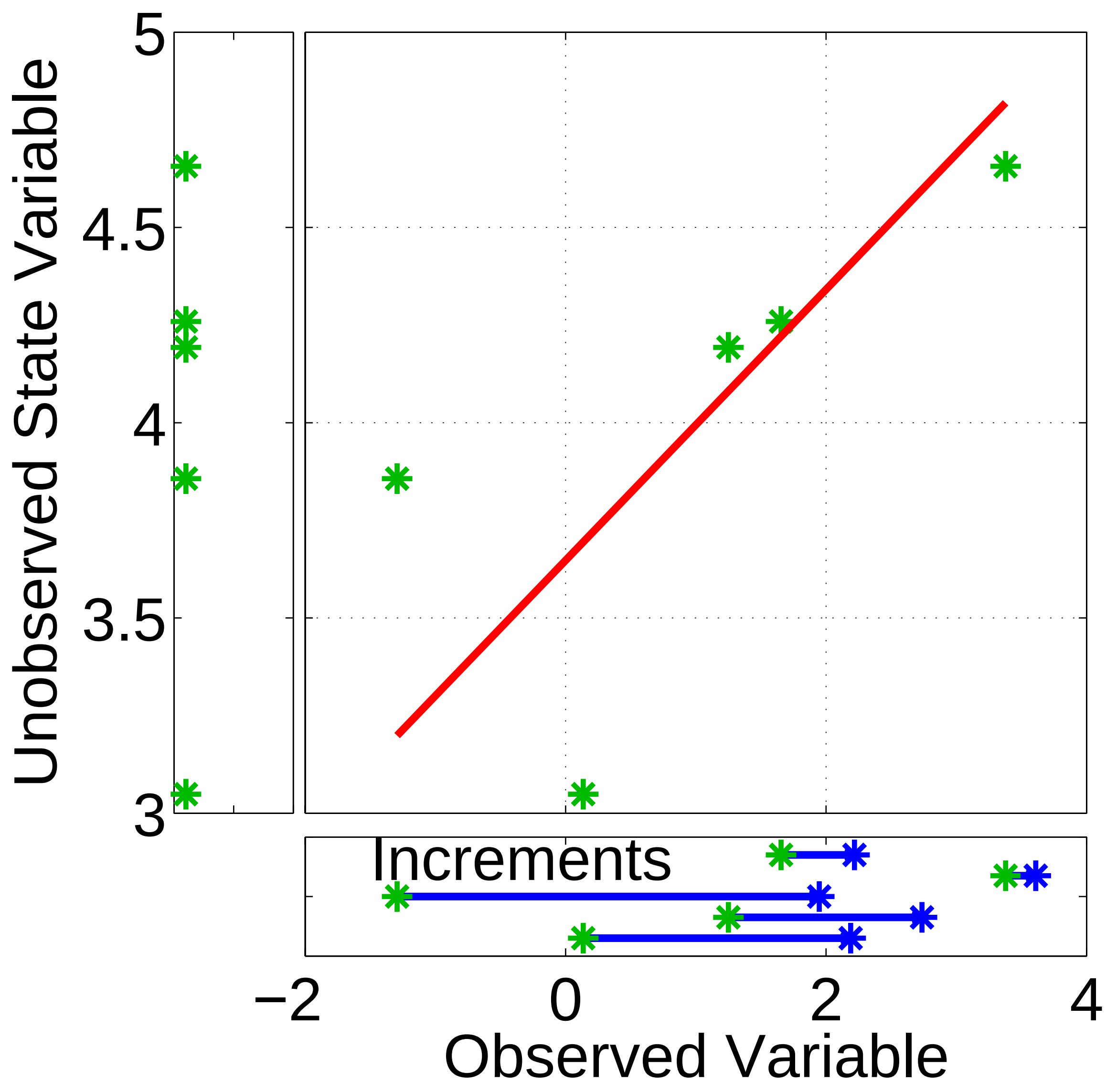
How should the unobserved variable be impacted?

First choice: least squares

Equivalent to linear regression.

Same as assuming binormal prior.

Ensemble filters: Updating additional prior state variables



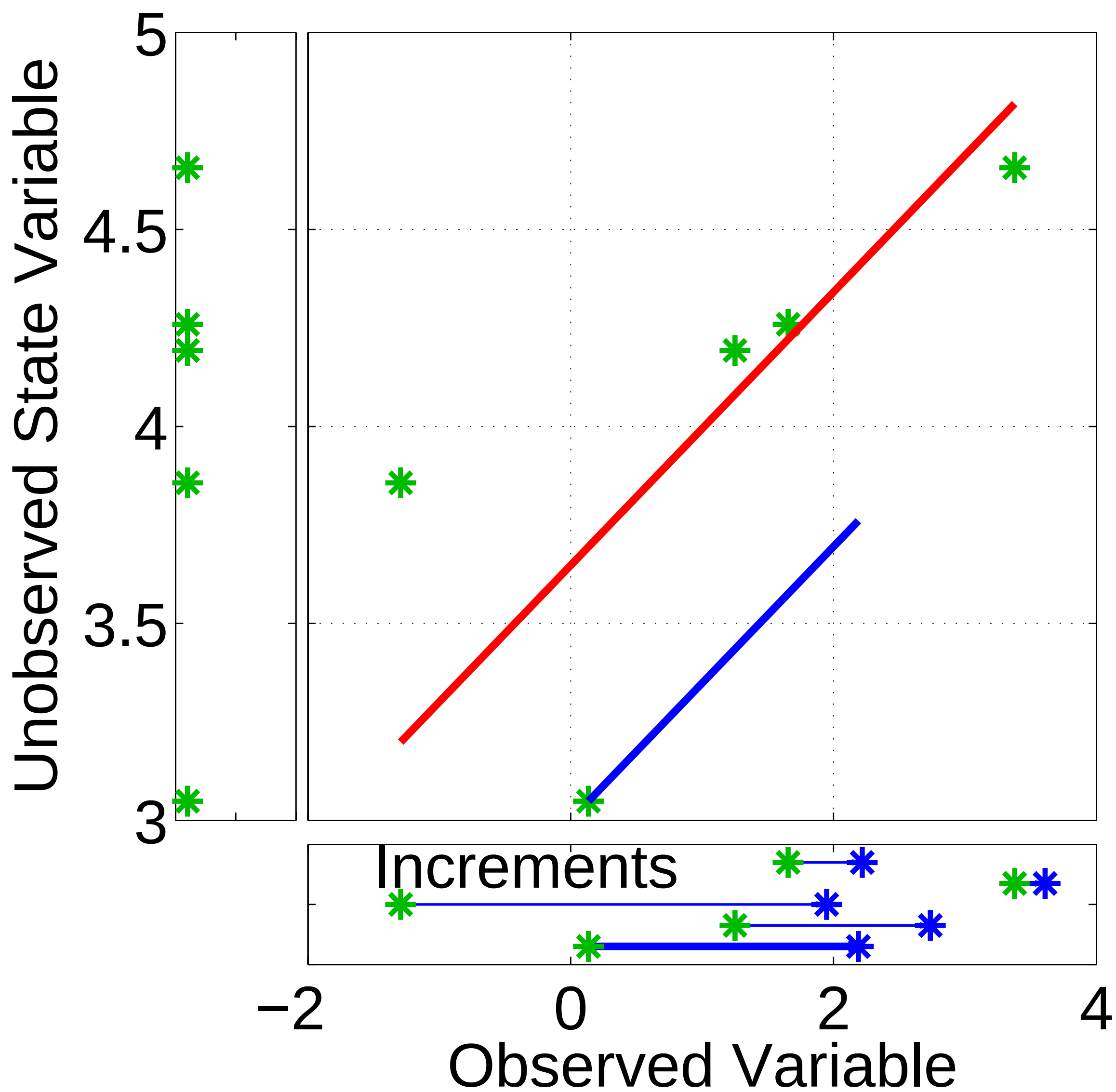
Have joint prior distribution of two variables.

How should the unobserved variable be impacted?

First choice: least squares

Begin by finding least squares fit.

Ensemble filters: Updating additional prior state variables

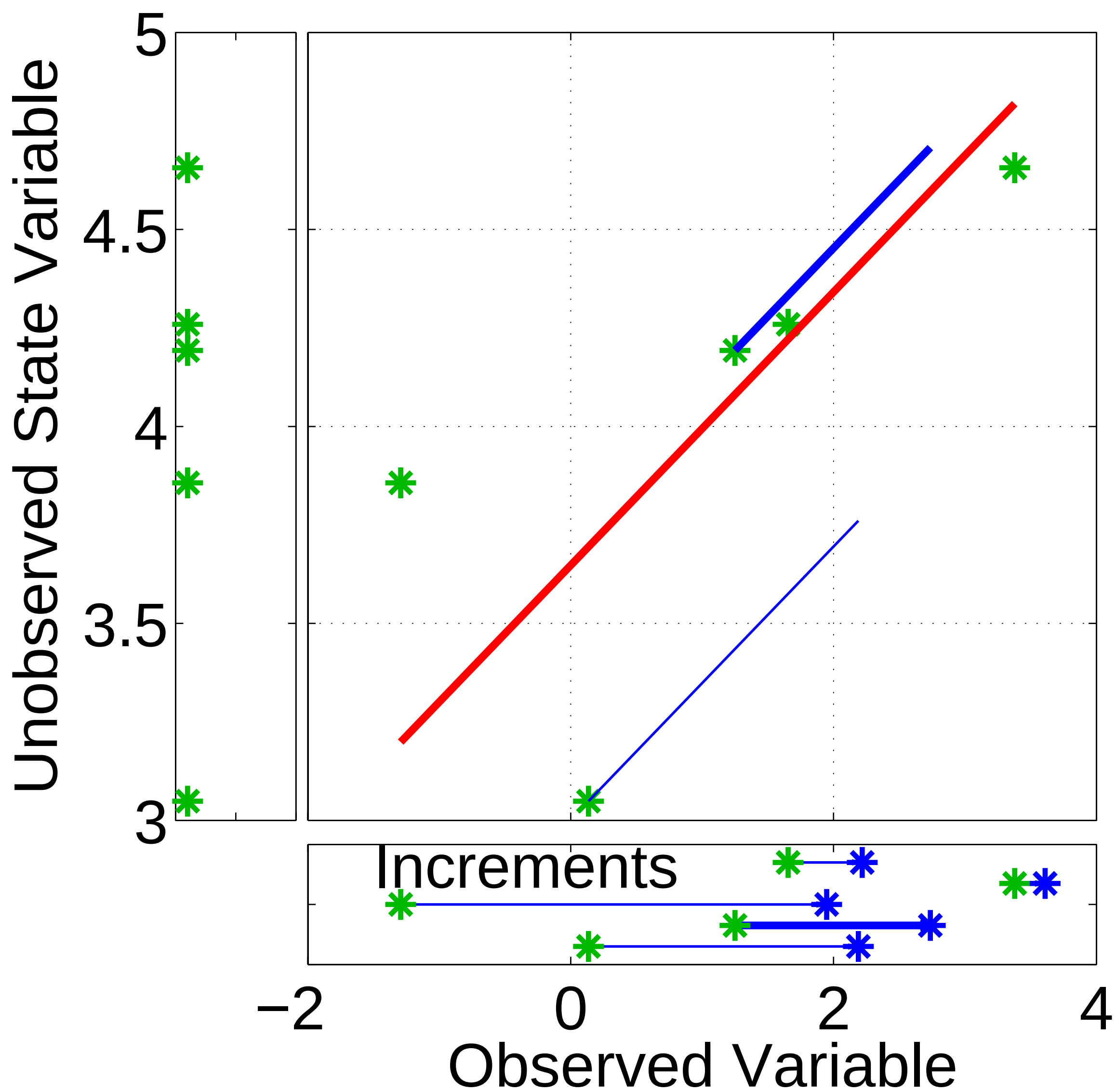


Have joint prior distribution of two variables.

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

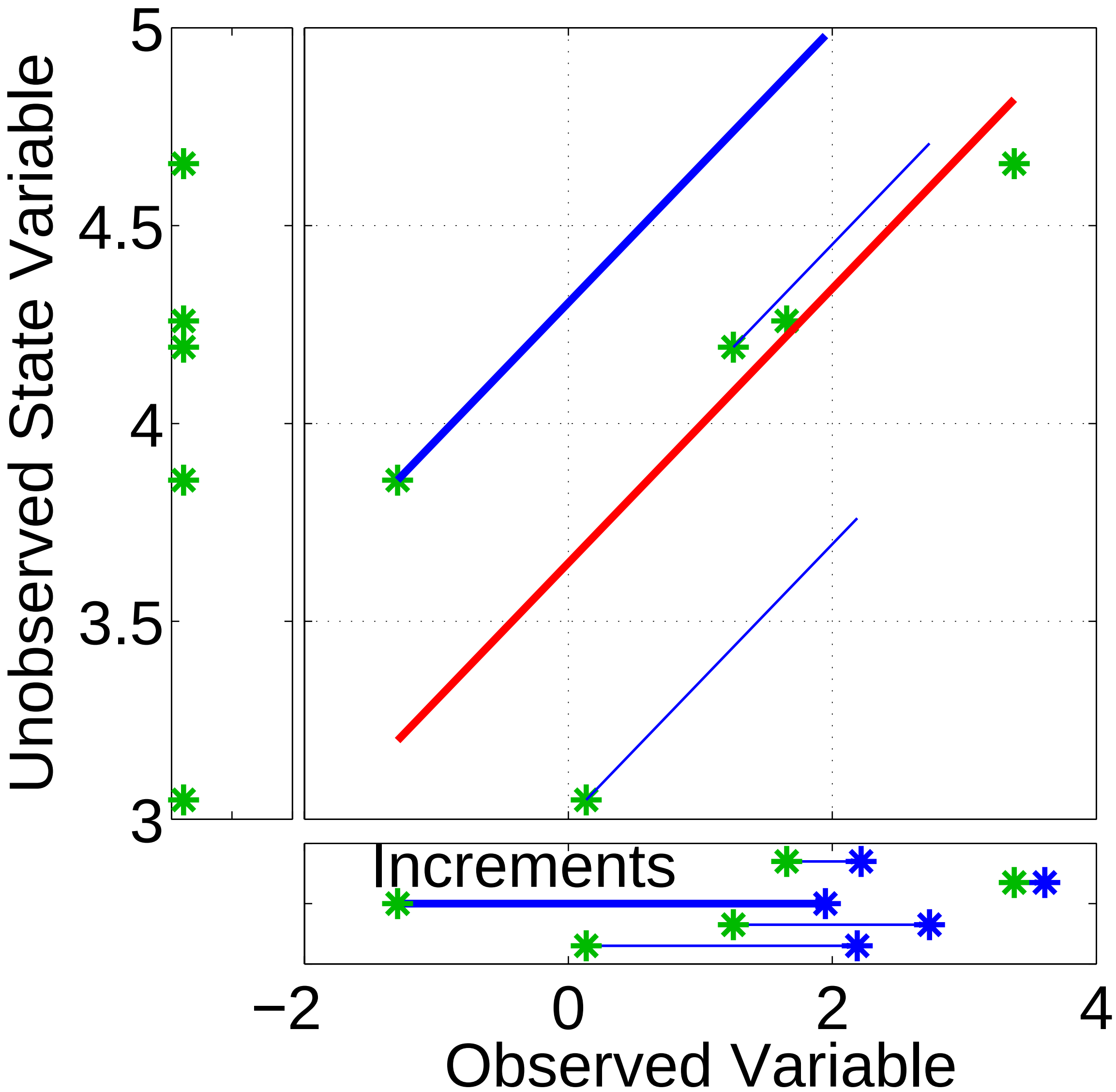


Have joint prior distribution of two variables.

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

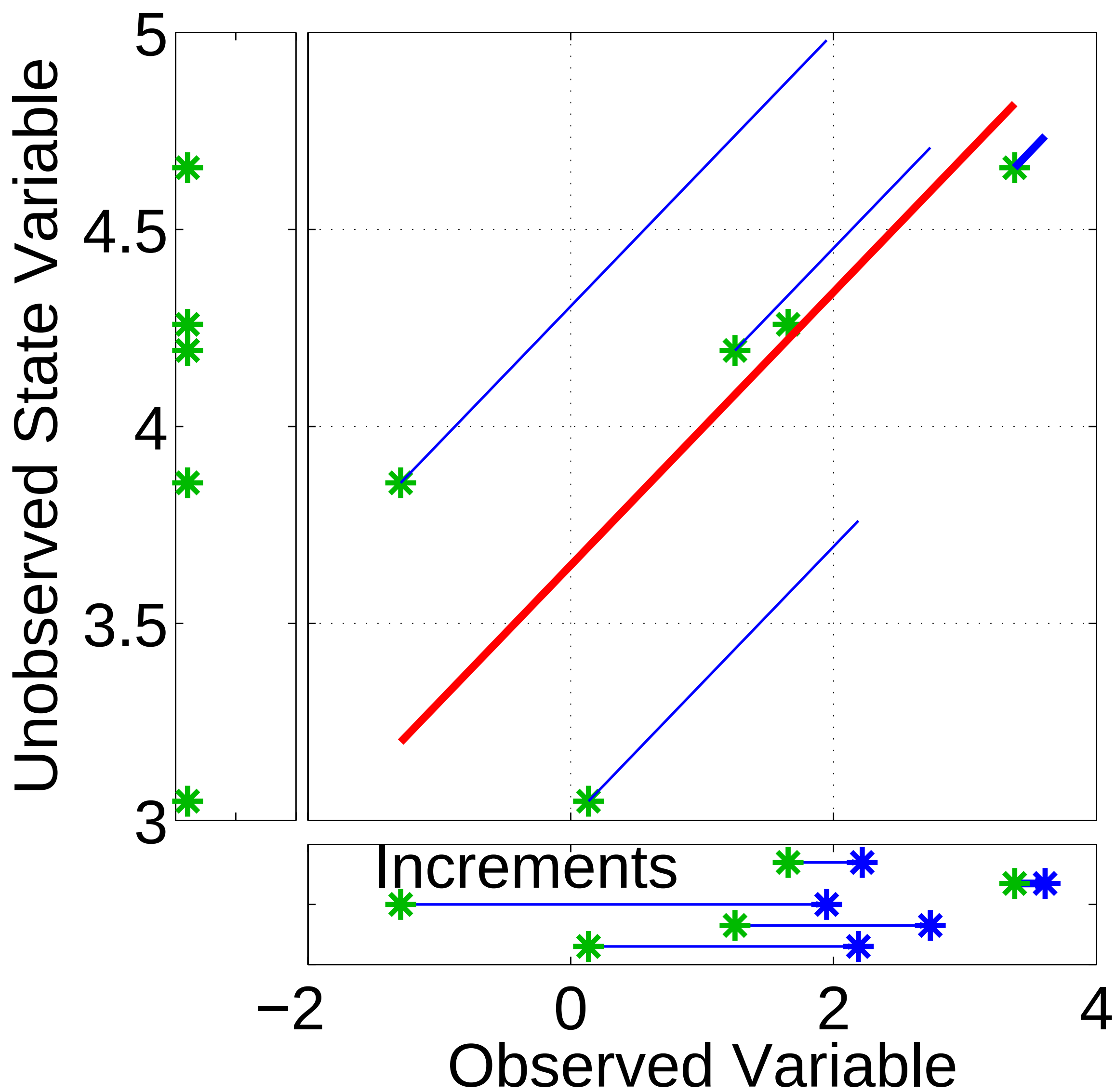


Have joint prior distribution of two variables.

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

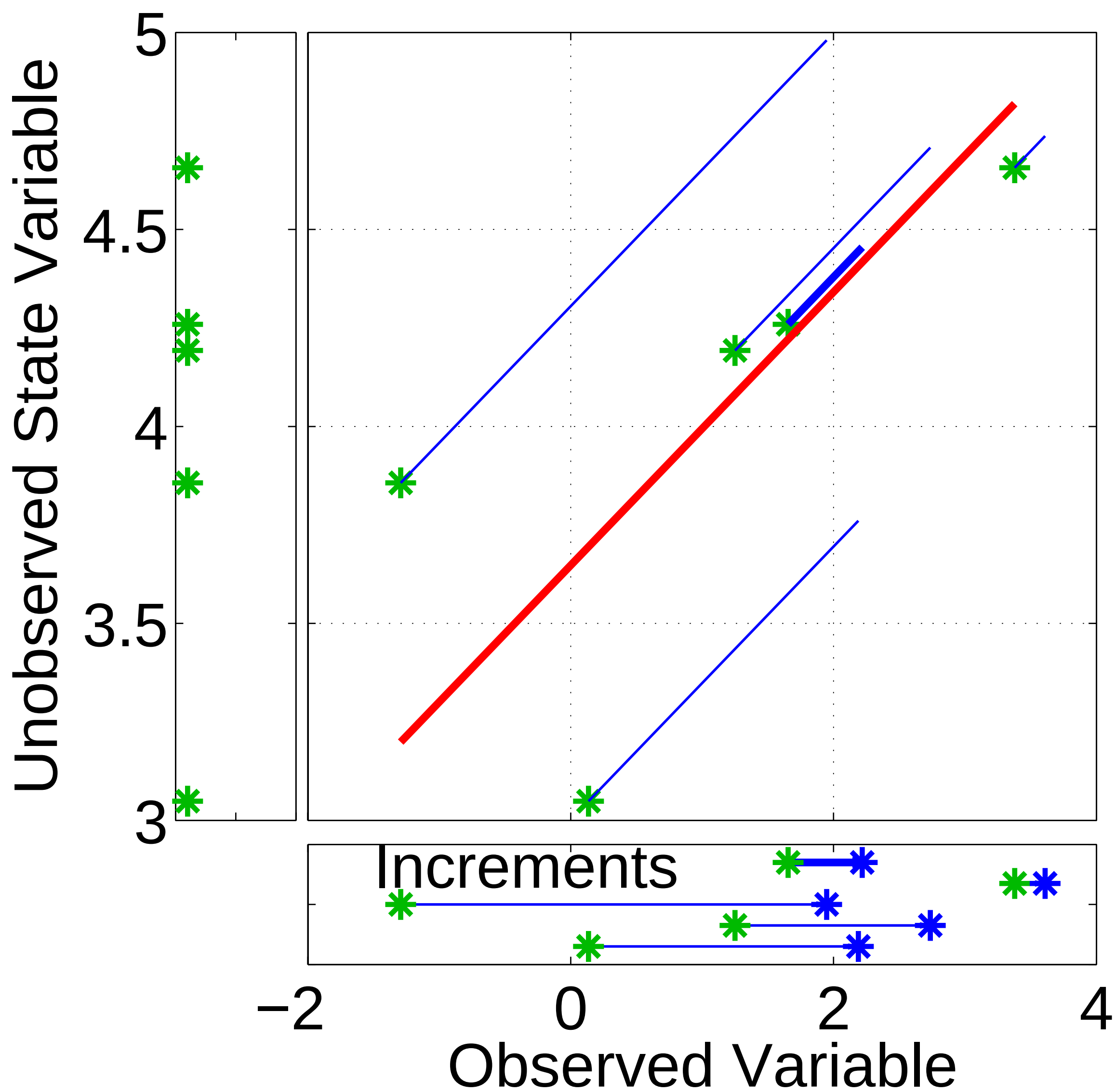


Have joint prior distribution of two variables.

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

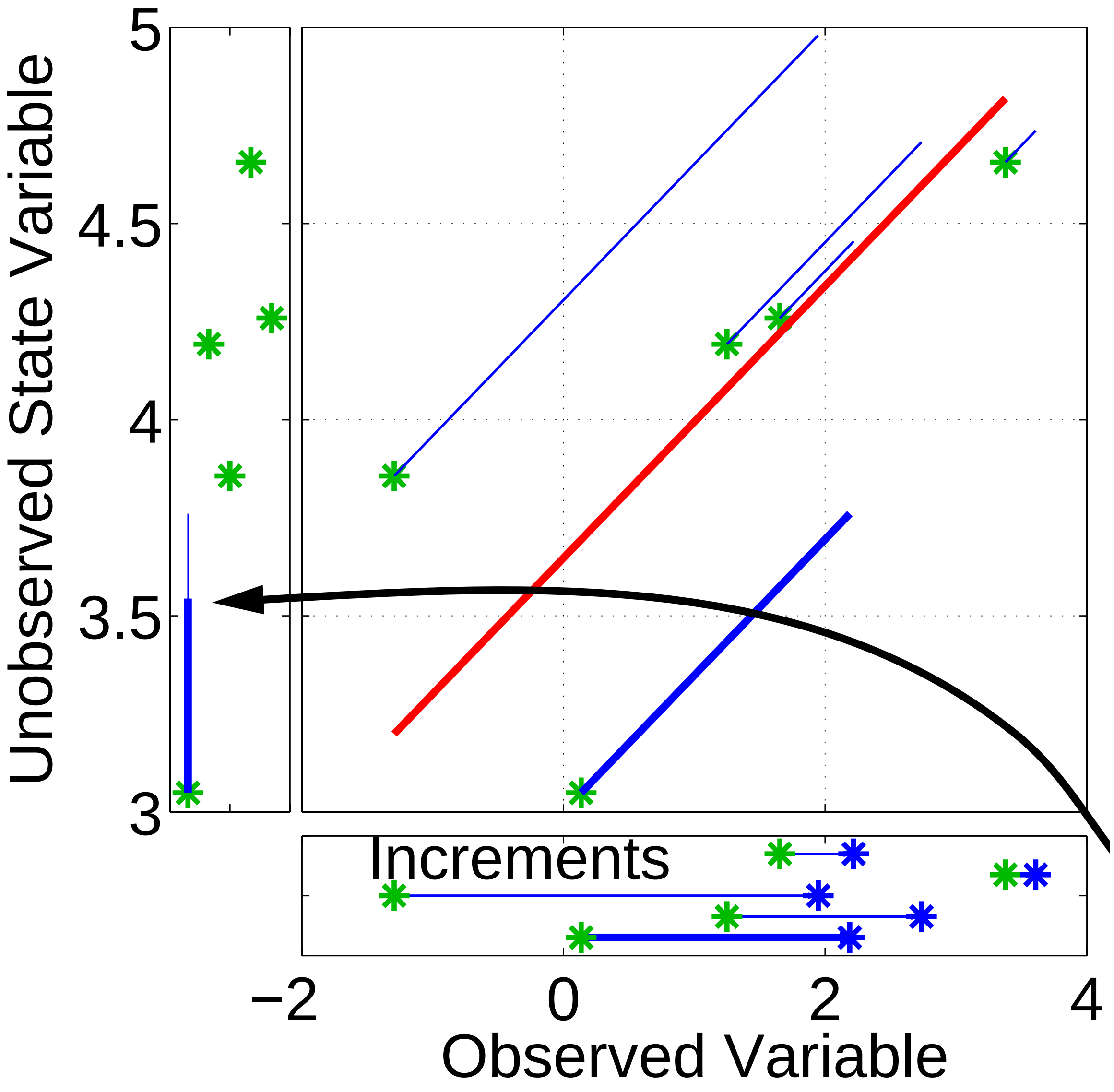


Have joint prior distribution of two variables.

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



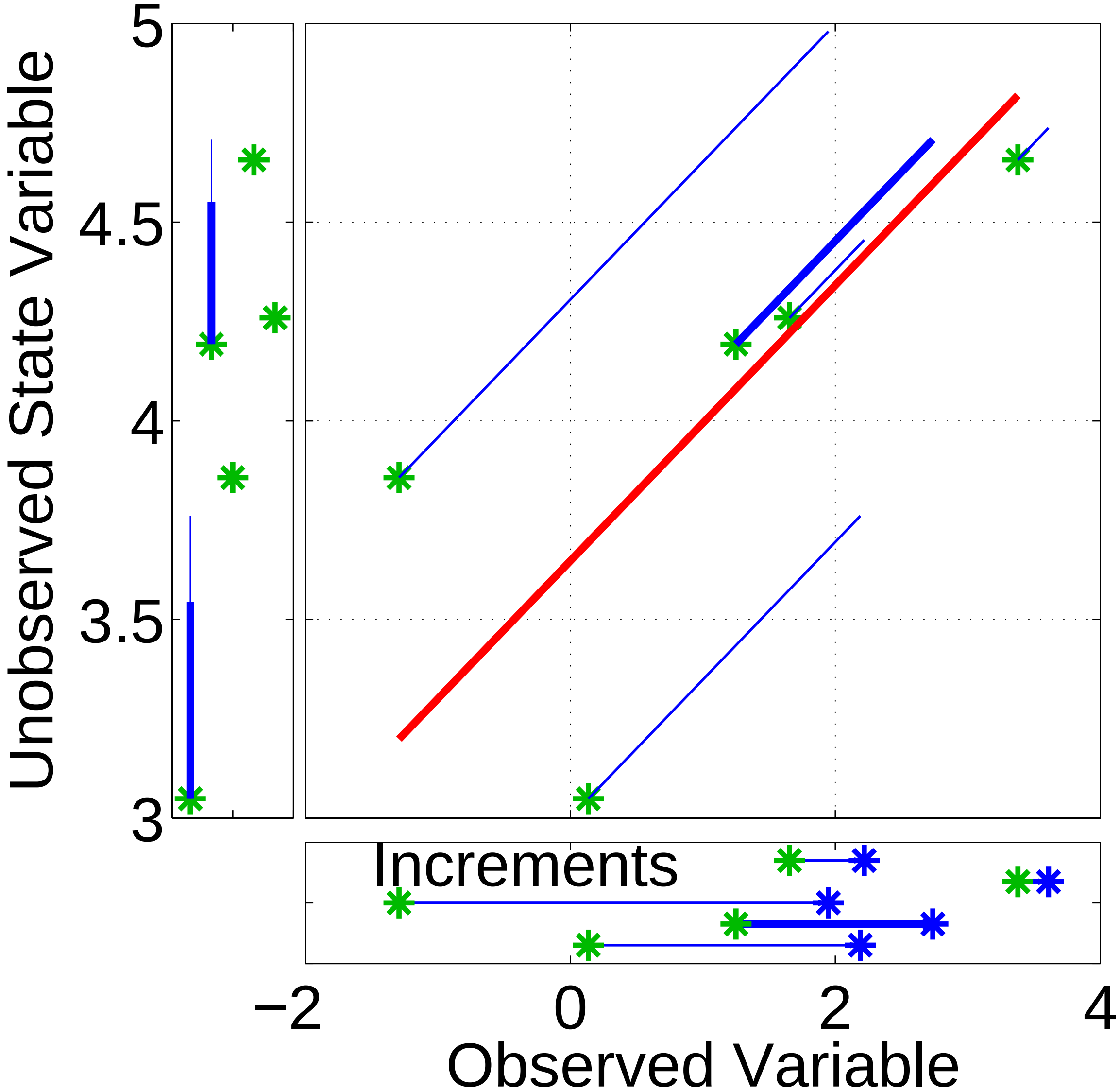
Have joint prior distribution of two variables.

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



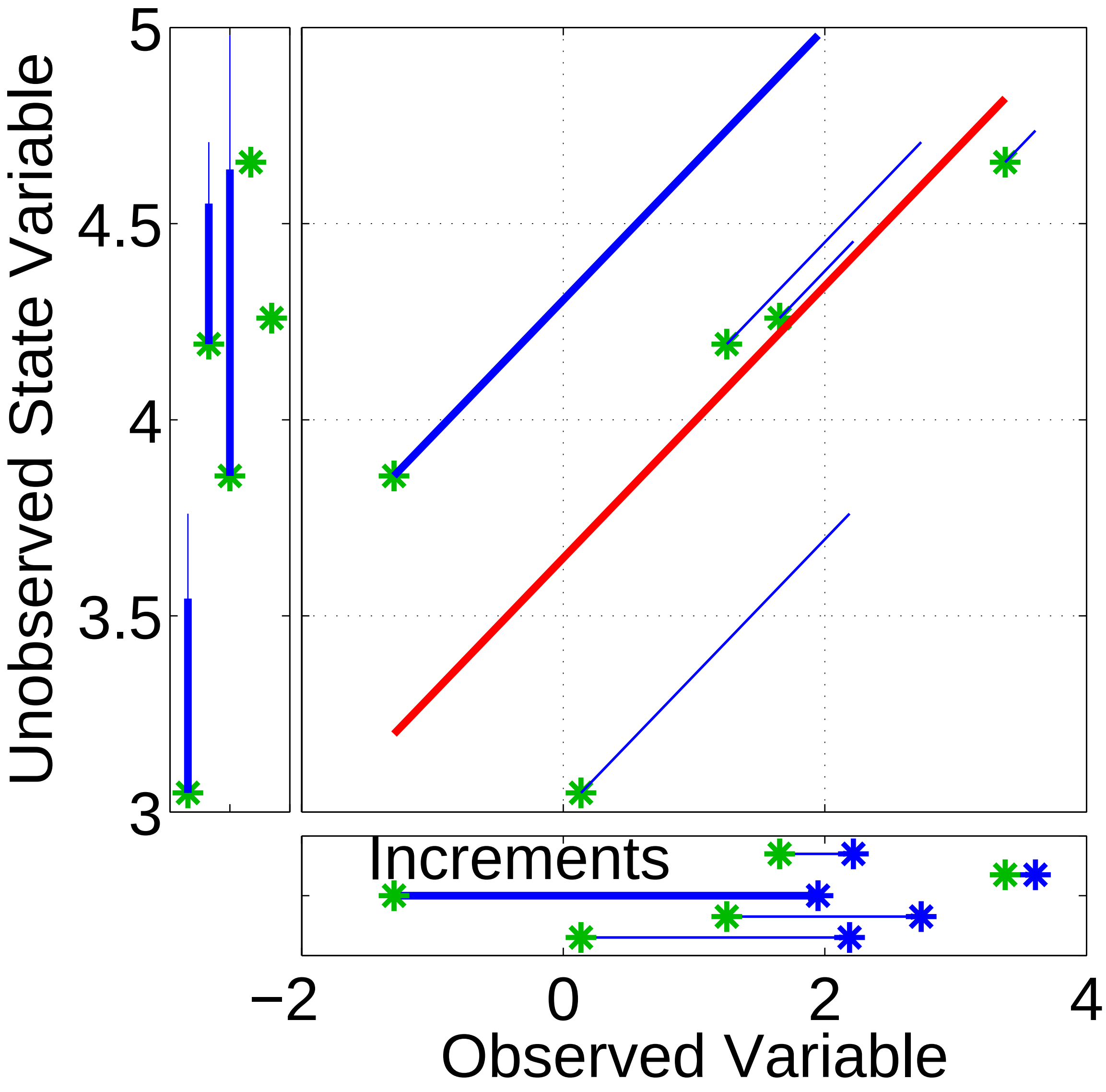
Have joint prior distribution of two variables.

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



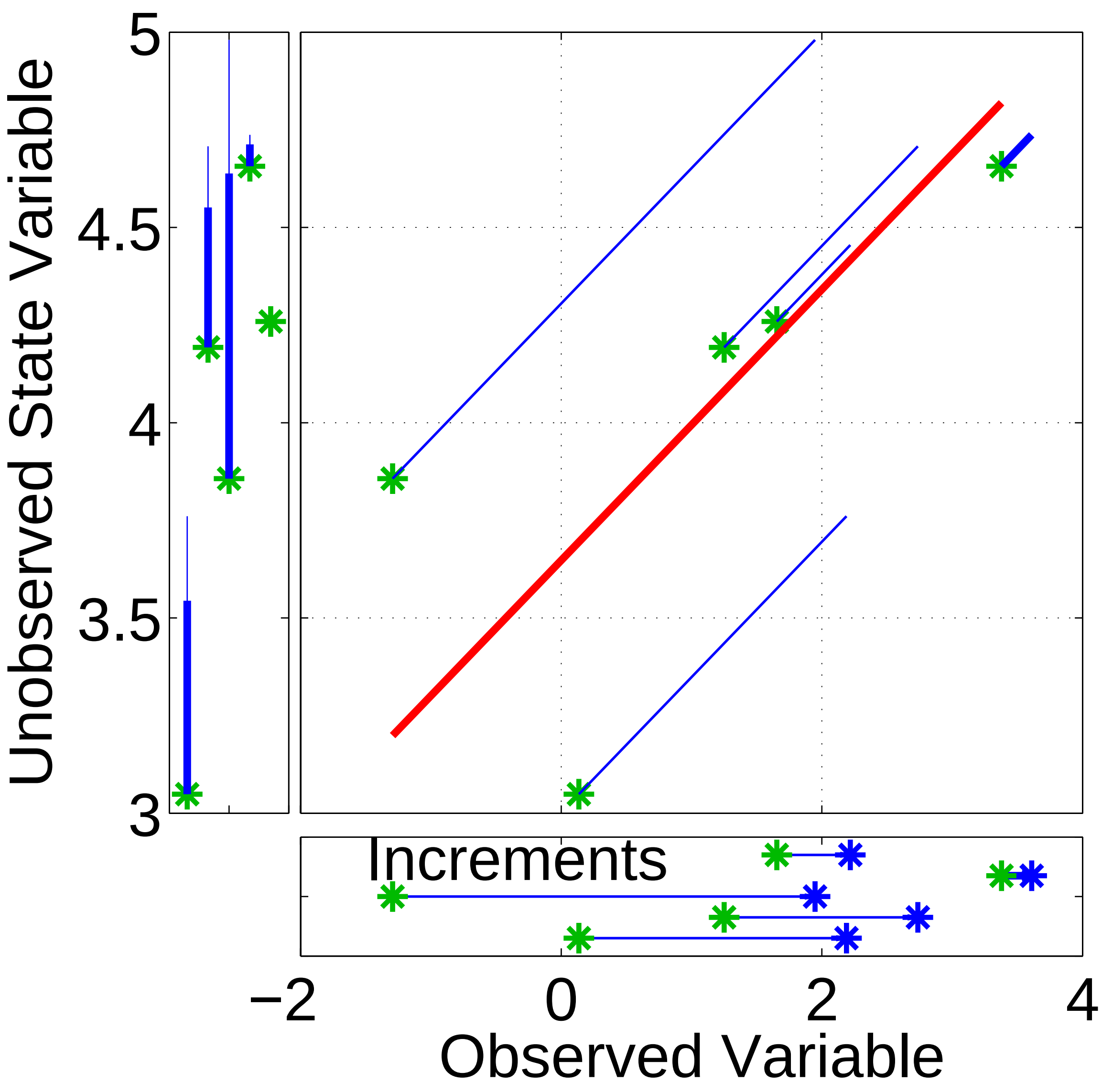
Have joint prior distribution of two variables.

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



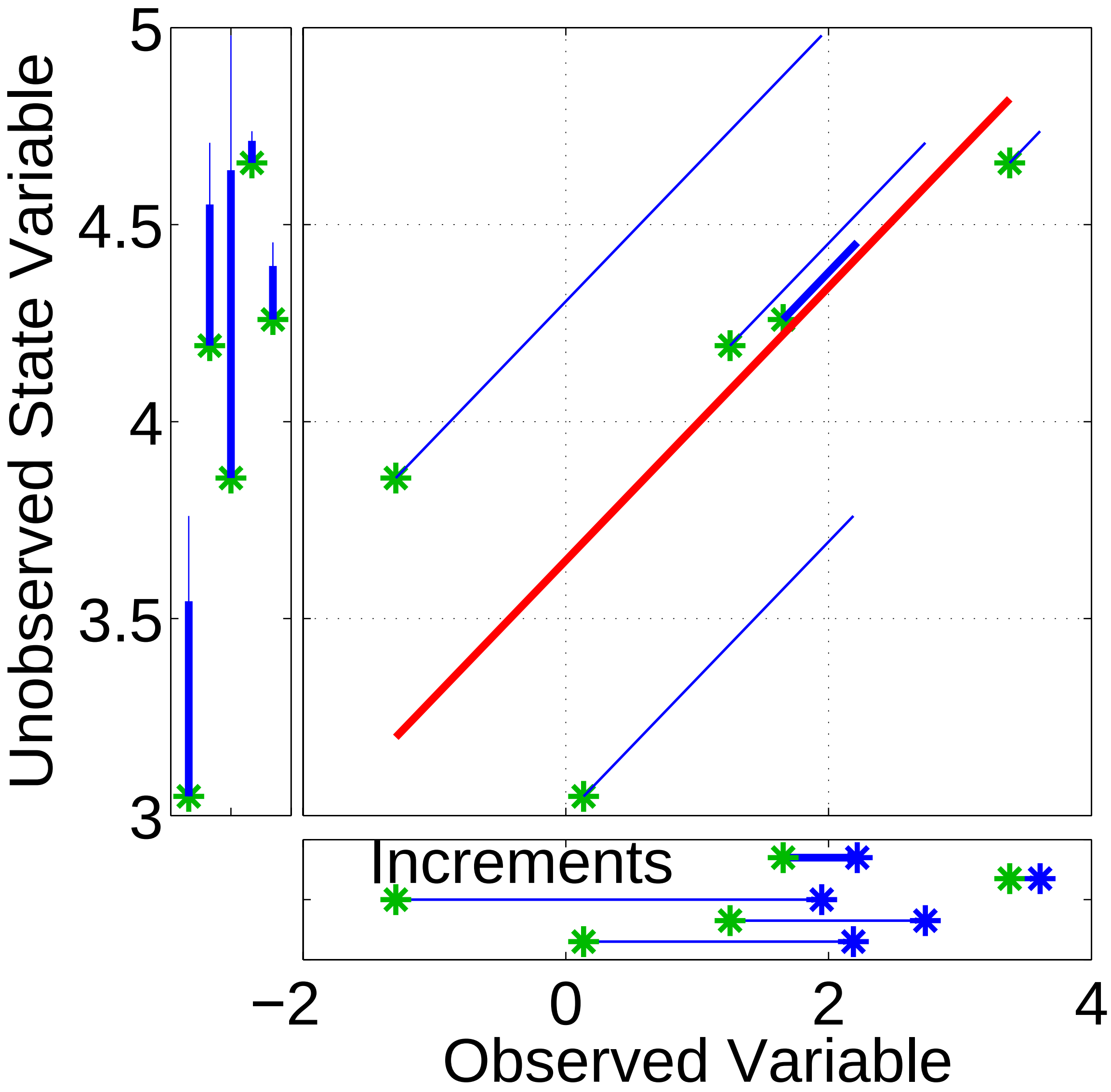
Have joint prior distribution of two variables.

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



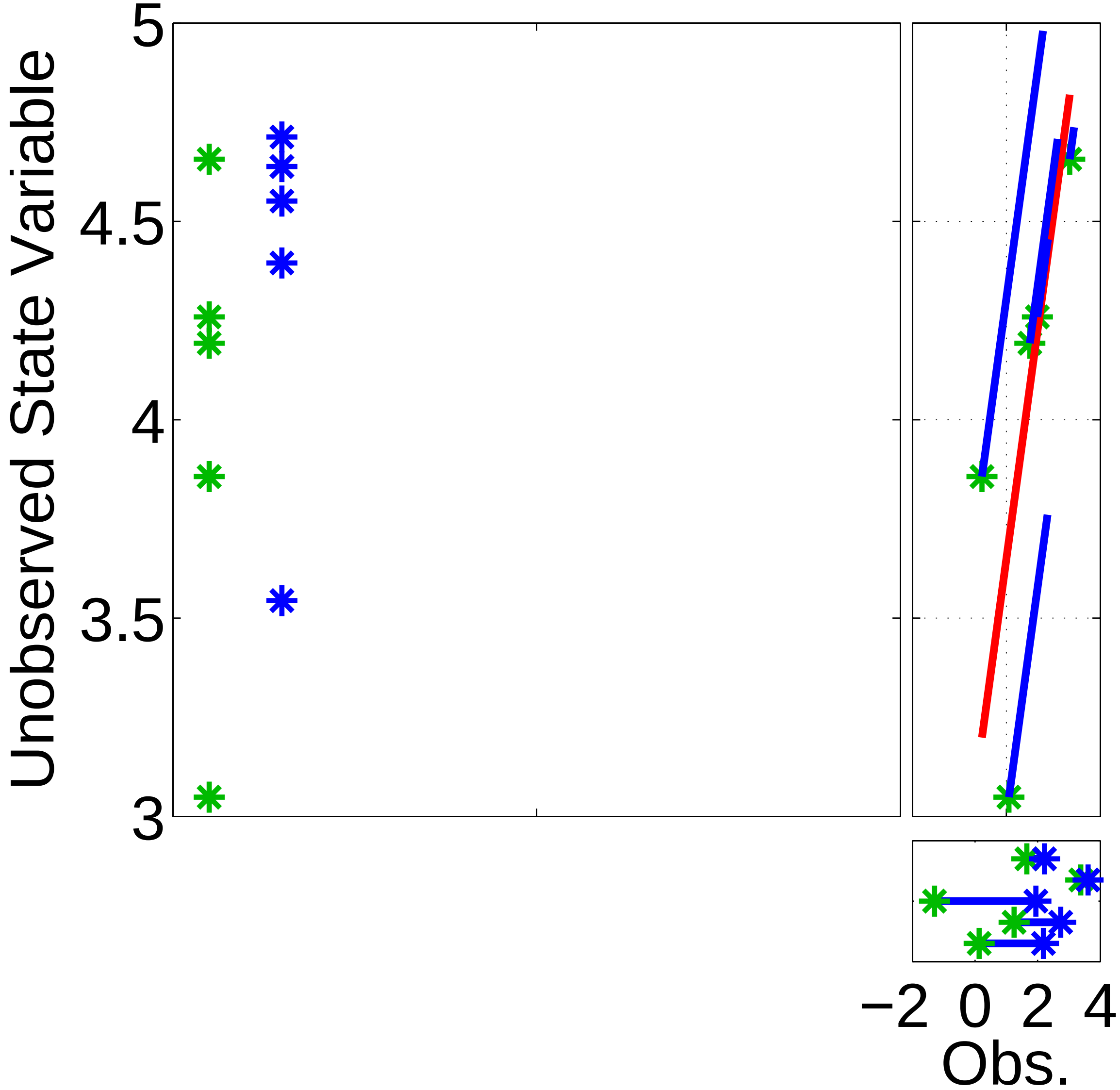
Have joint prior distribution of two variables.

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

Then projecting from joint space onto unobserved priors.

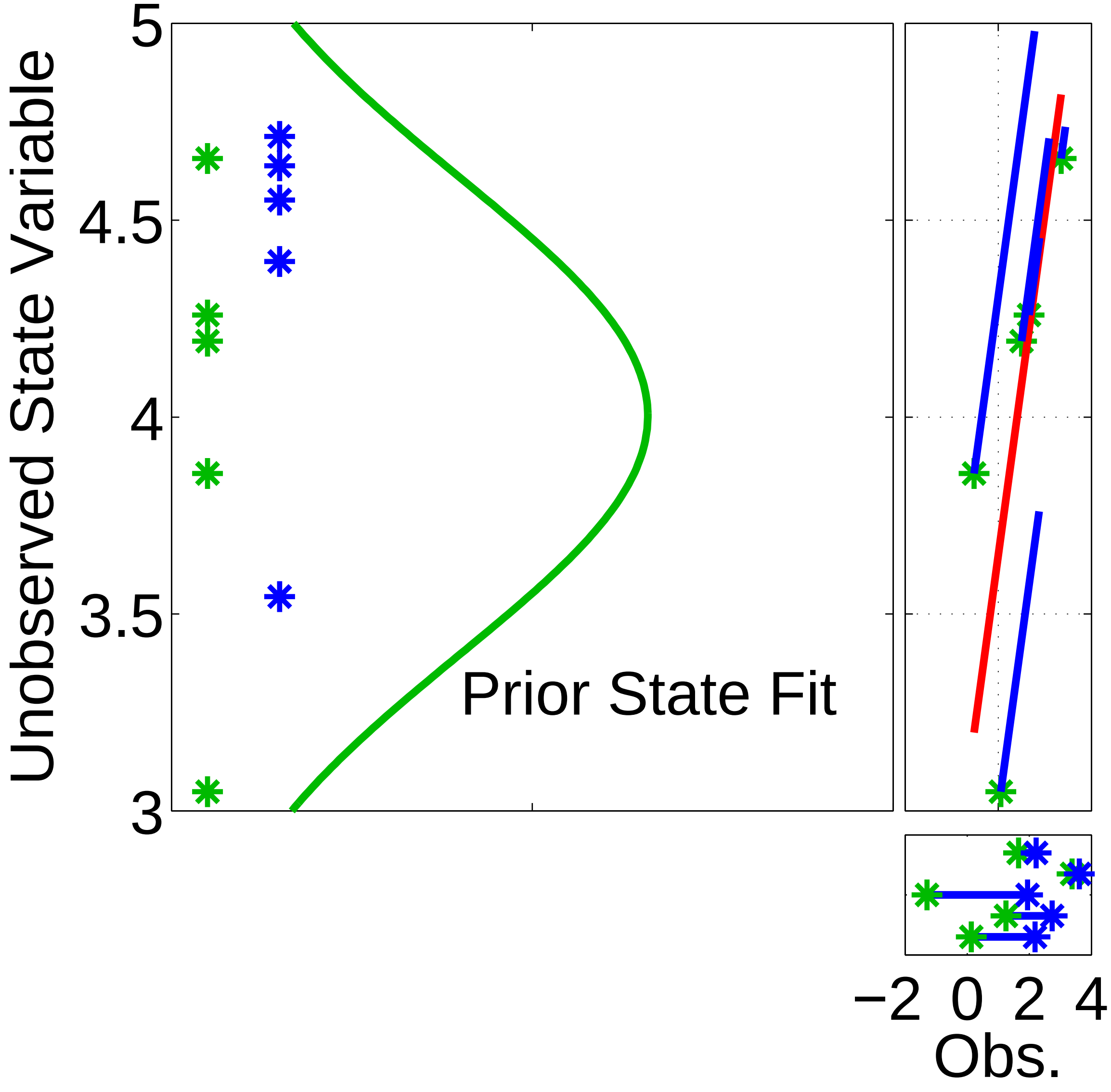
Finally, multiply by prior sample correlation.

Ensemble filters: Updating additional prior state variables



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

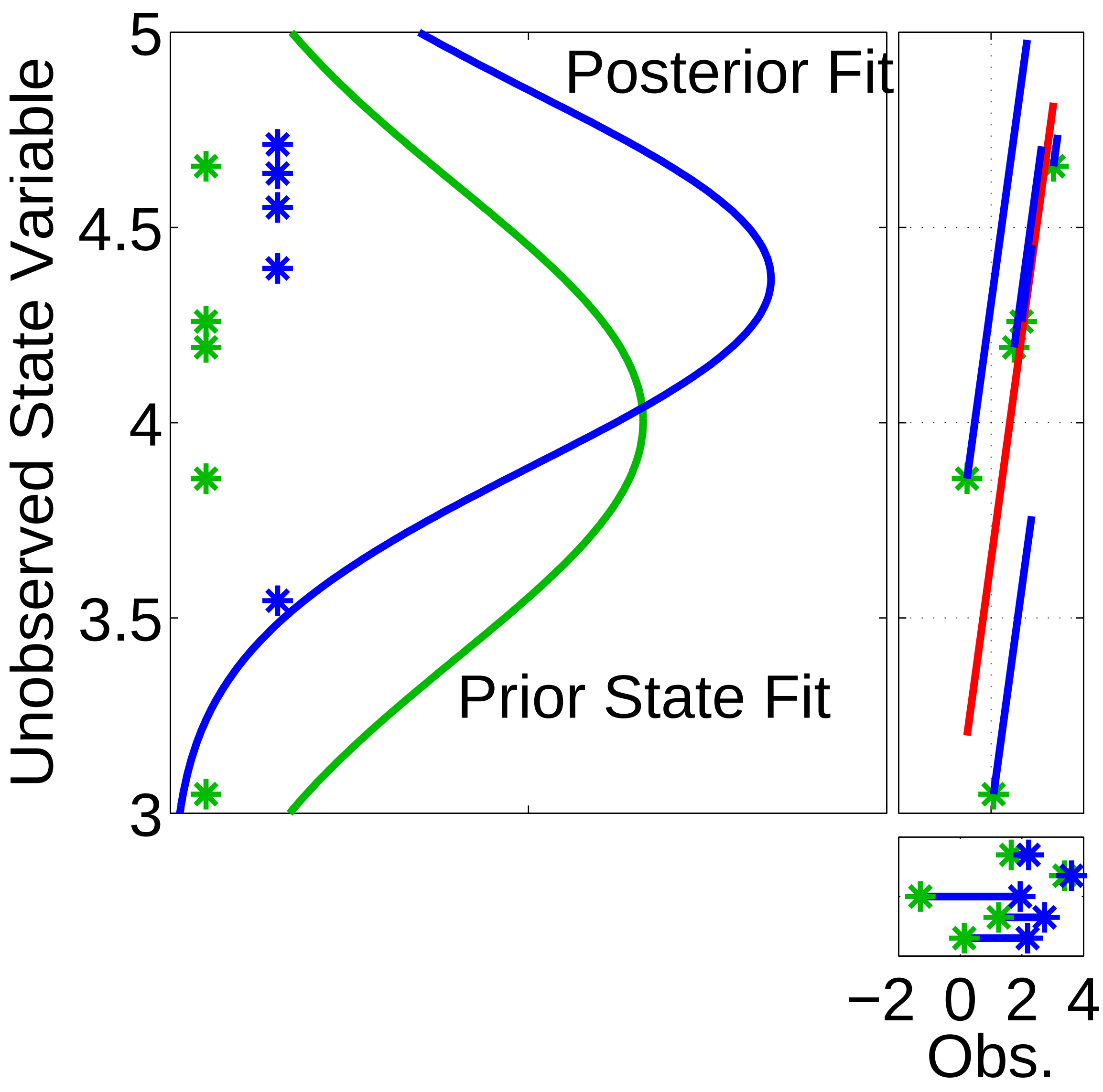
Ensemble filters: Updating additional prior state variables



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

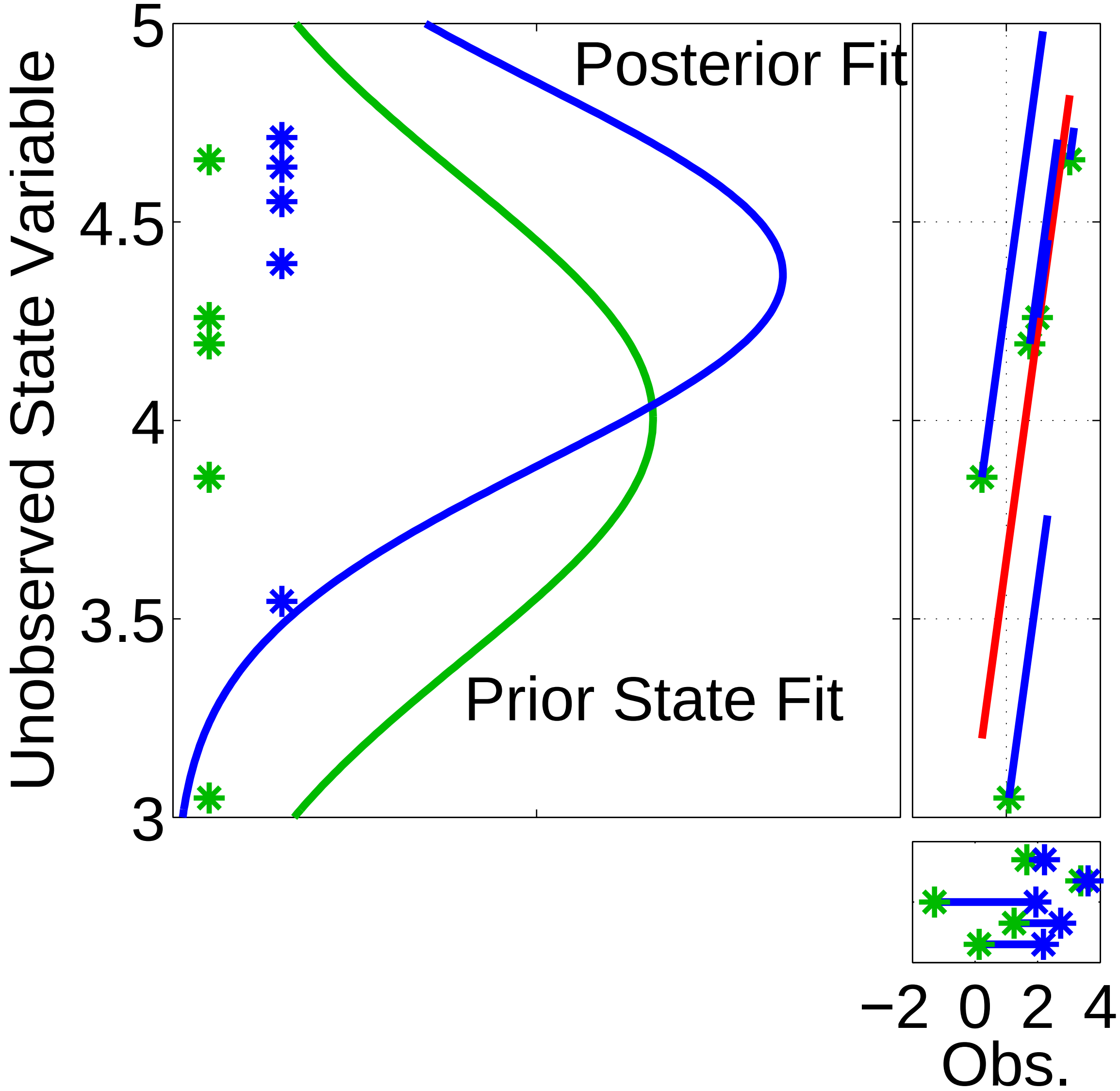


Now have an updated (posterior) ensemble for the unobserved variable.

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

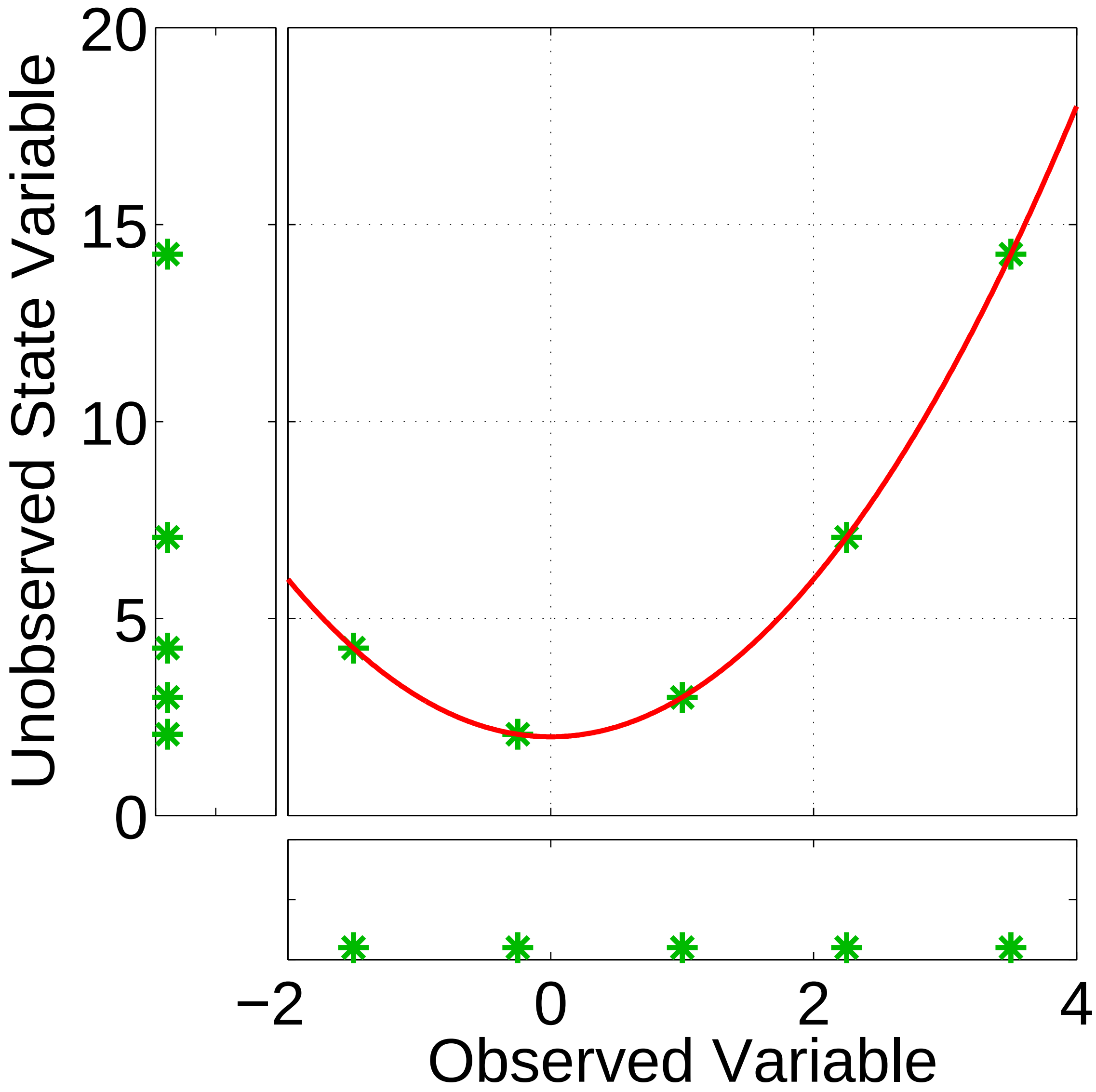
Two primary error sources:

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.

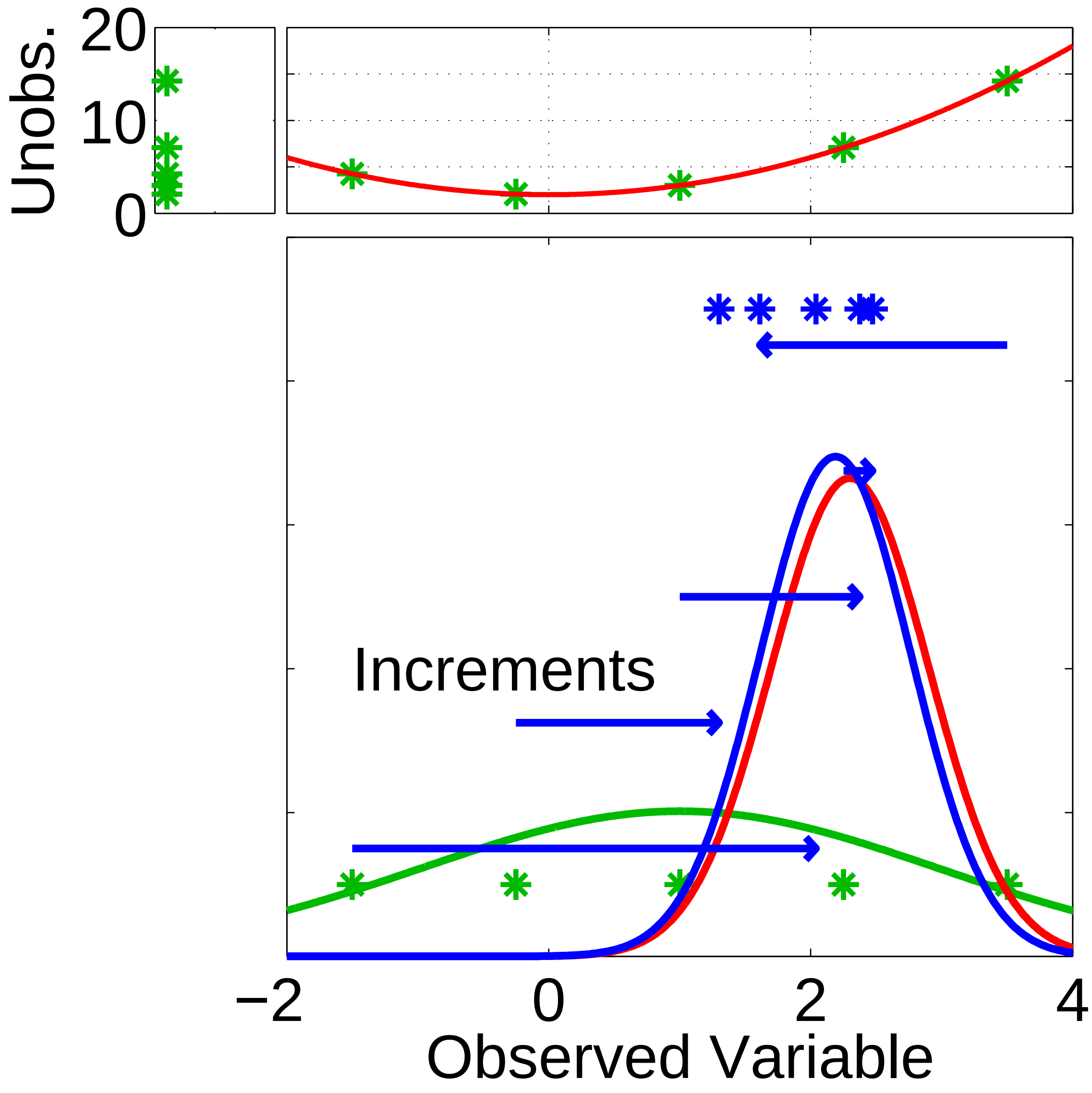
Nonlinear relations between variables; sorting increments



Suppose prior sample has NO noise.

But, relation between un/observed variables is non-linear.

Nonlinear relations between variables; sorting increments

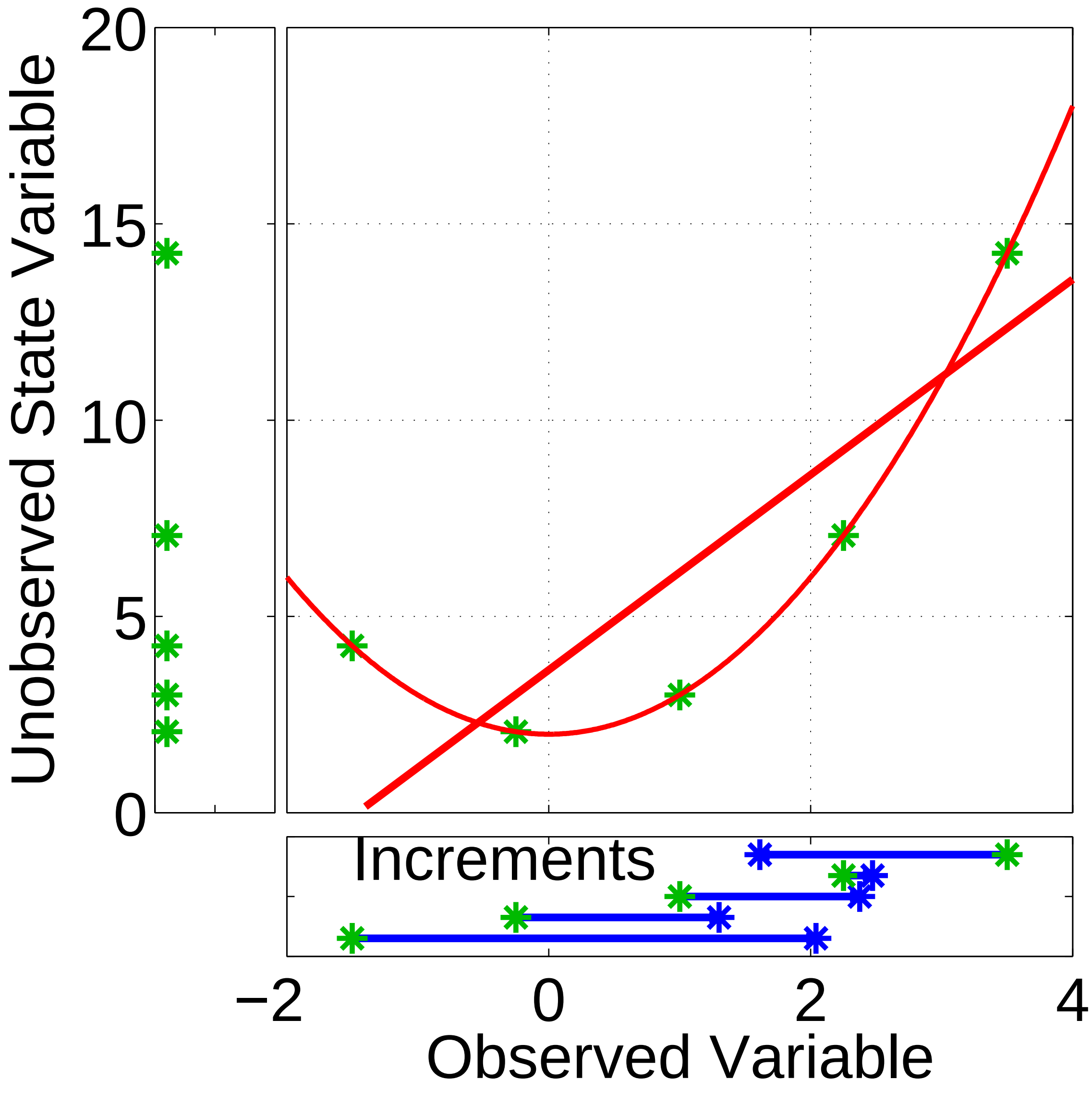


Suppose prior sample has NO noise.

But, relation between un/observed variables is non-linear.

Update observed sample and compute increments.

Nonlinear relations between variables; sorting increments

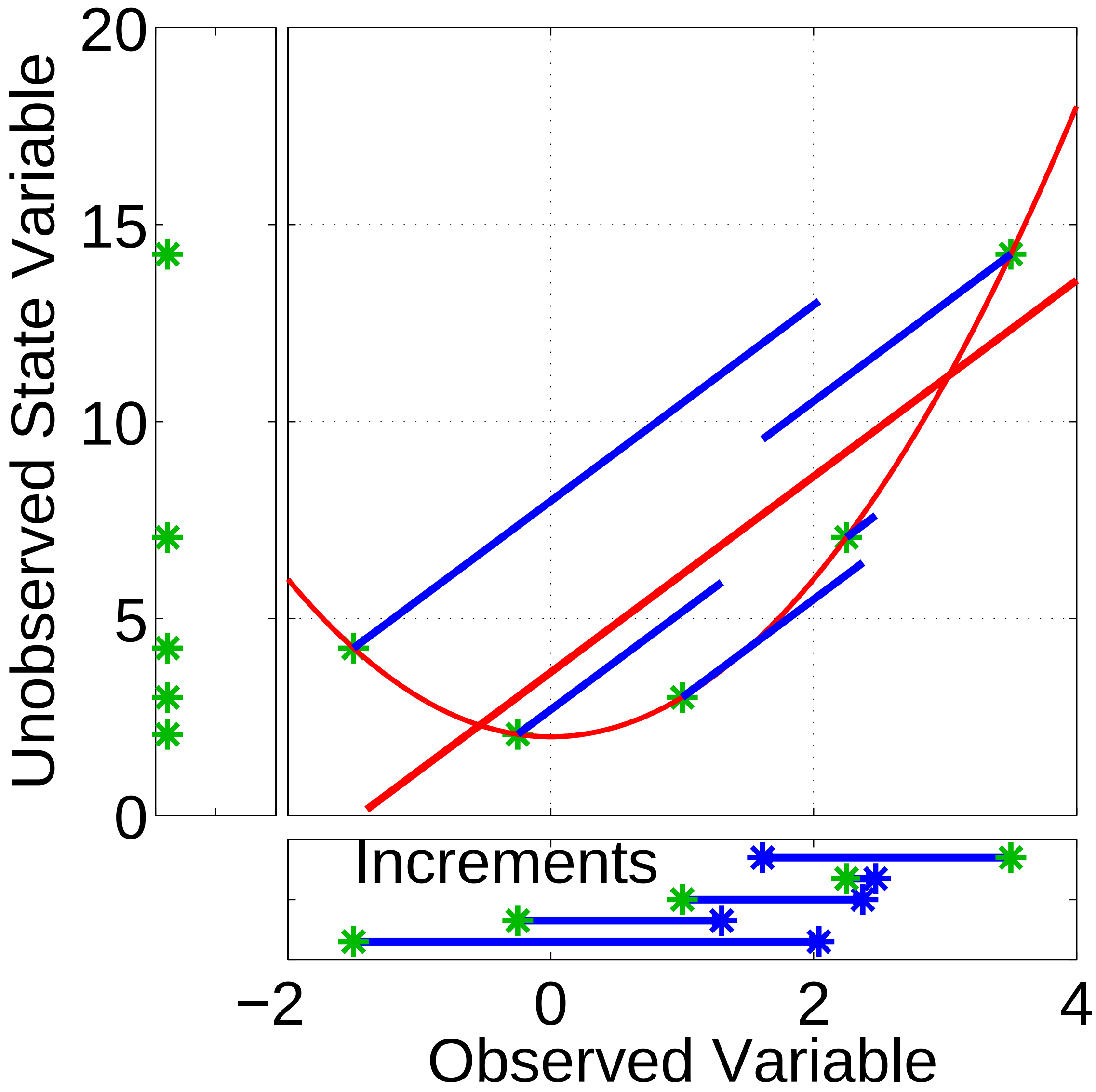


Suppose prior sample has NO noise.

But, relation between un/observed variables is non-linear.

Regression error varies with value of observed variable.

Nonlinear relations between variables; sorting increments



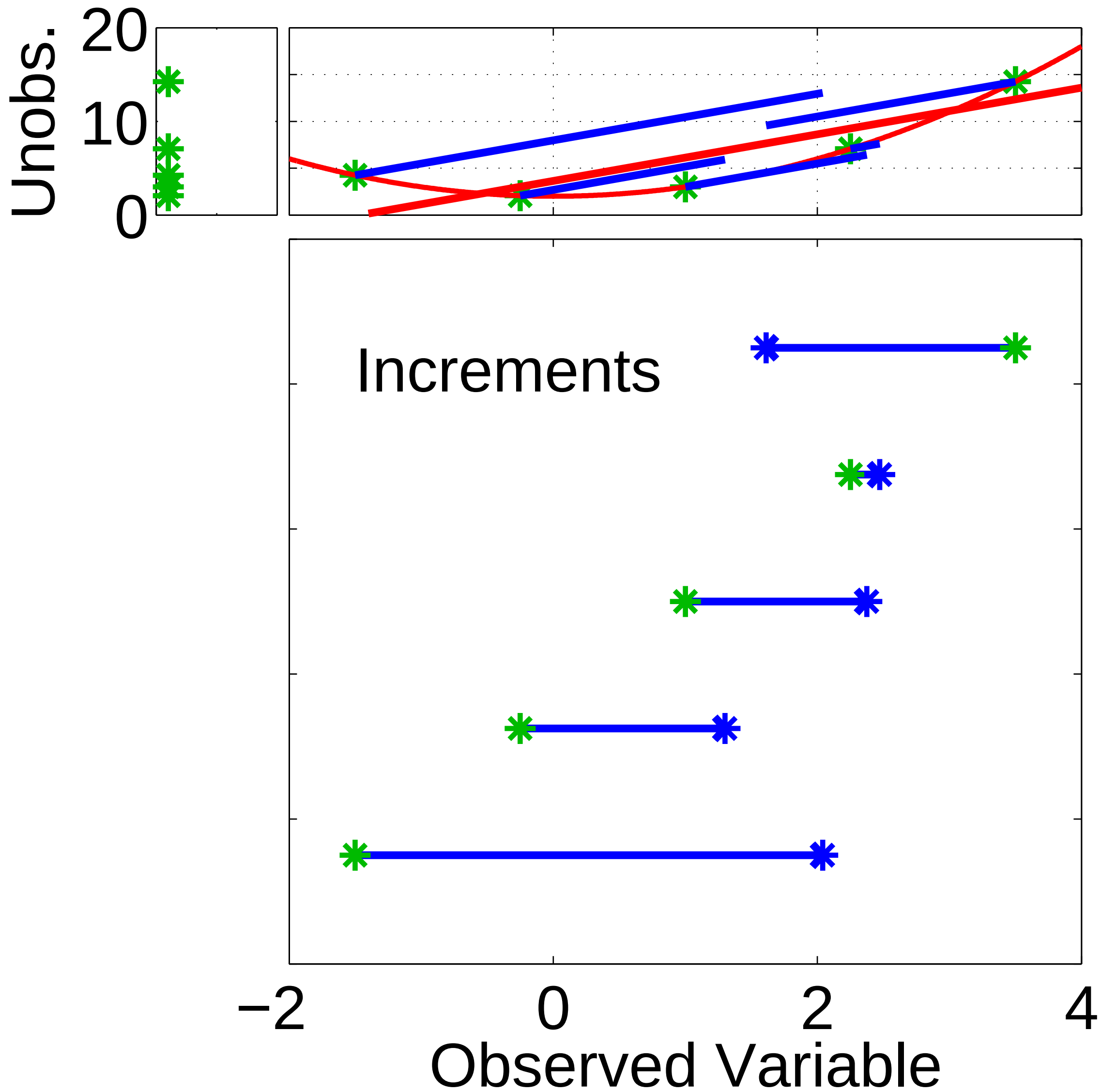
Suppose prior sample has NO noise.

But, relation between un/observed variables is non-linear.

Regression error varies with value of observed variable.

Smaller increments have smaller expected errors.

Nonlinear relations between variables; sorting increments



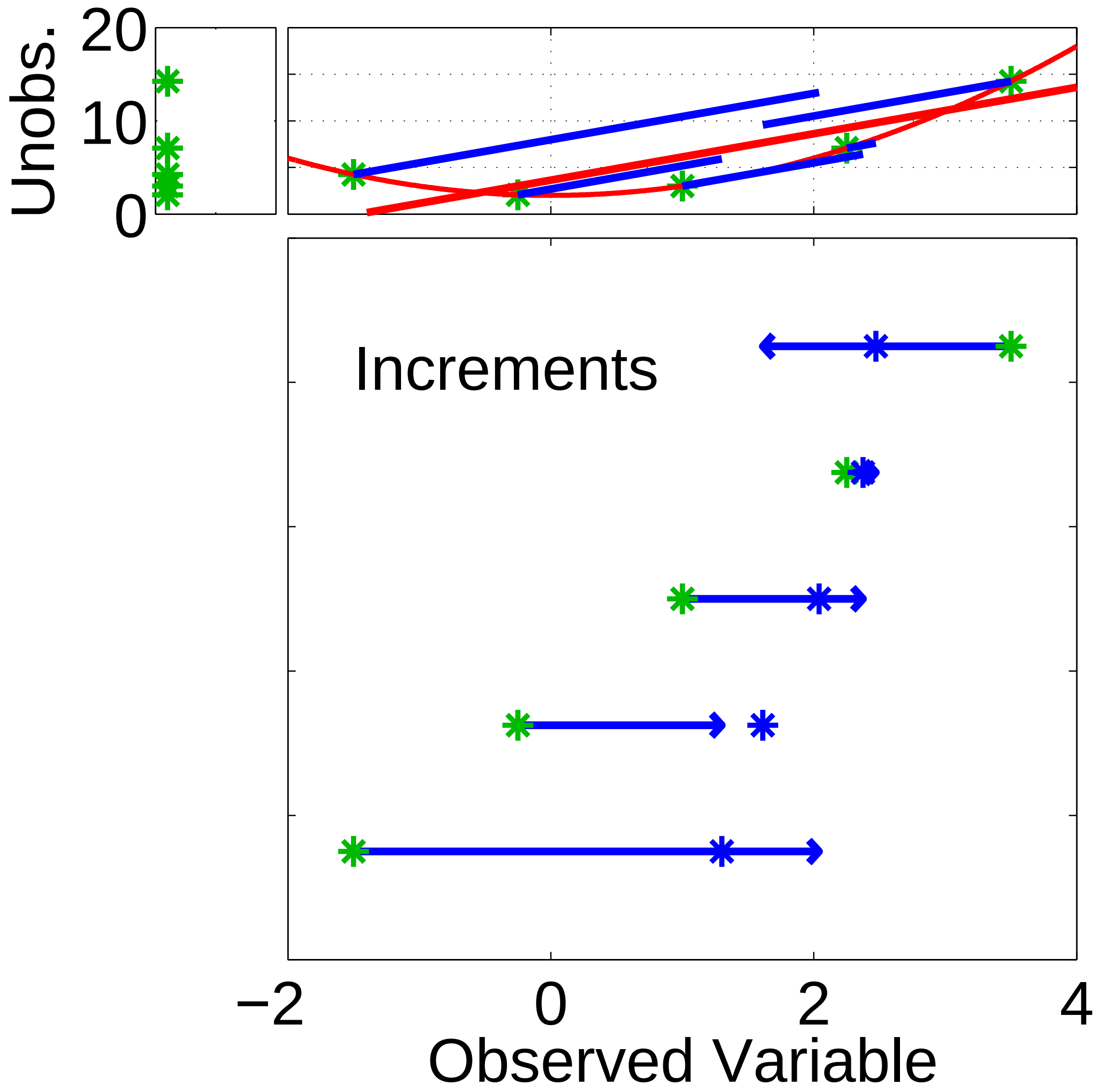
Suppose prior sample has NO noise.

But, relation between un/observed variables is non-linear.

Pairing between prior and posterior sample of observed variable can be viewed as arbitrary.

Posterior is same sample no matter how it is paired.

Nonlinear relations between variables; sorting increments



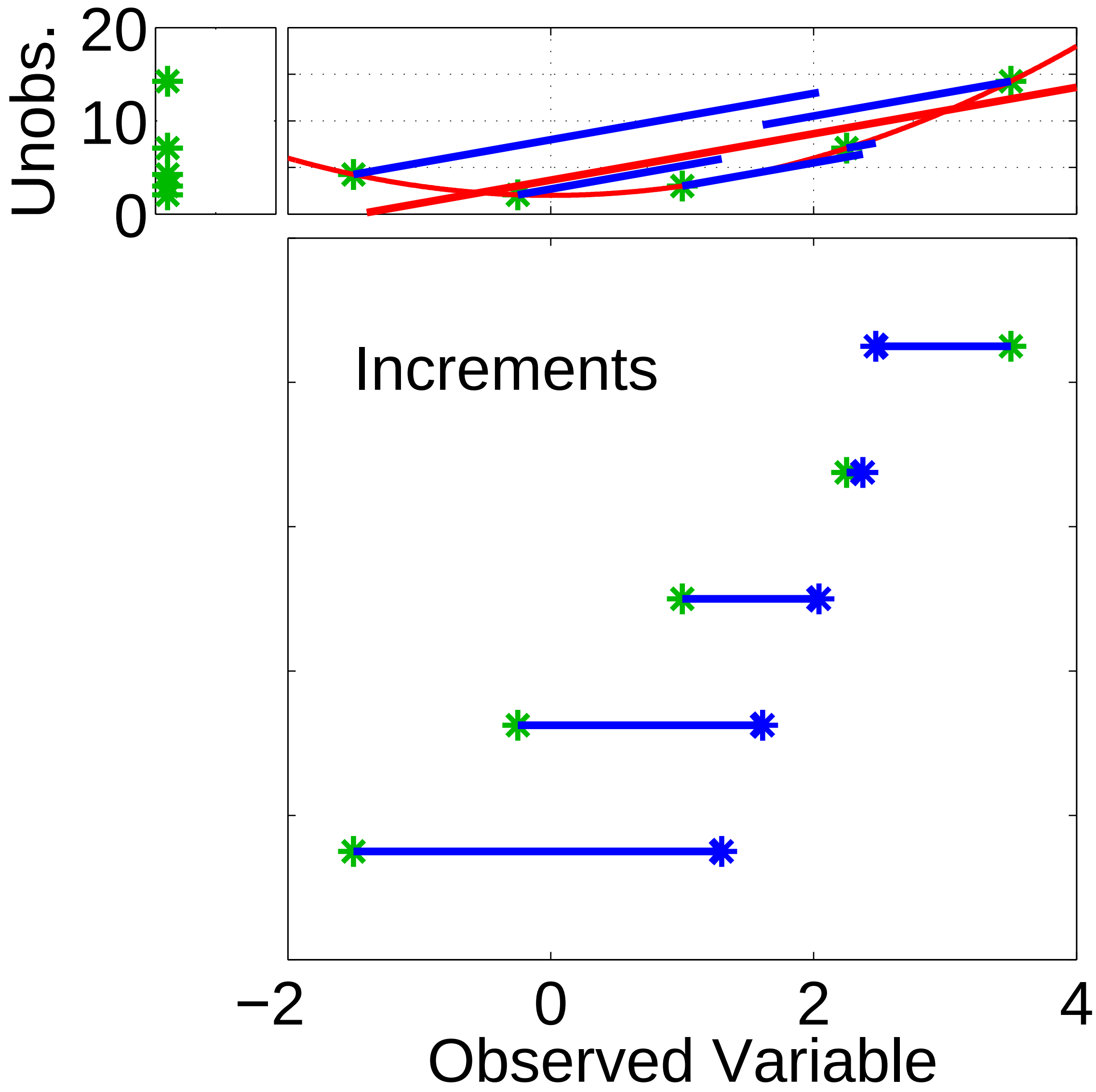
Suppose prior sample has NO noise.

But, relation between un/observed variables is non-linear.

Can minimize increments by changing pairing.

Sorting prior and posterior and pairing samples minimizes one norm of increment size (could do other methods)

Nonlinear relations between variables; sorting increments



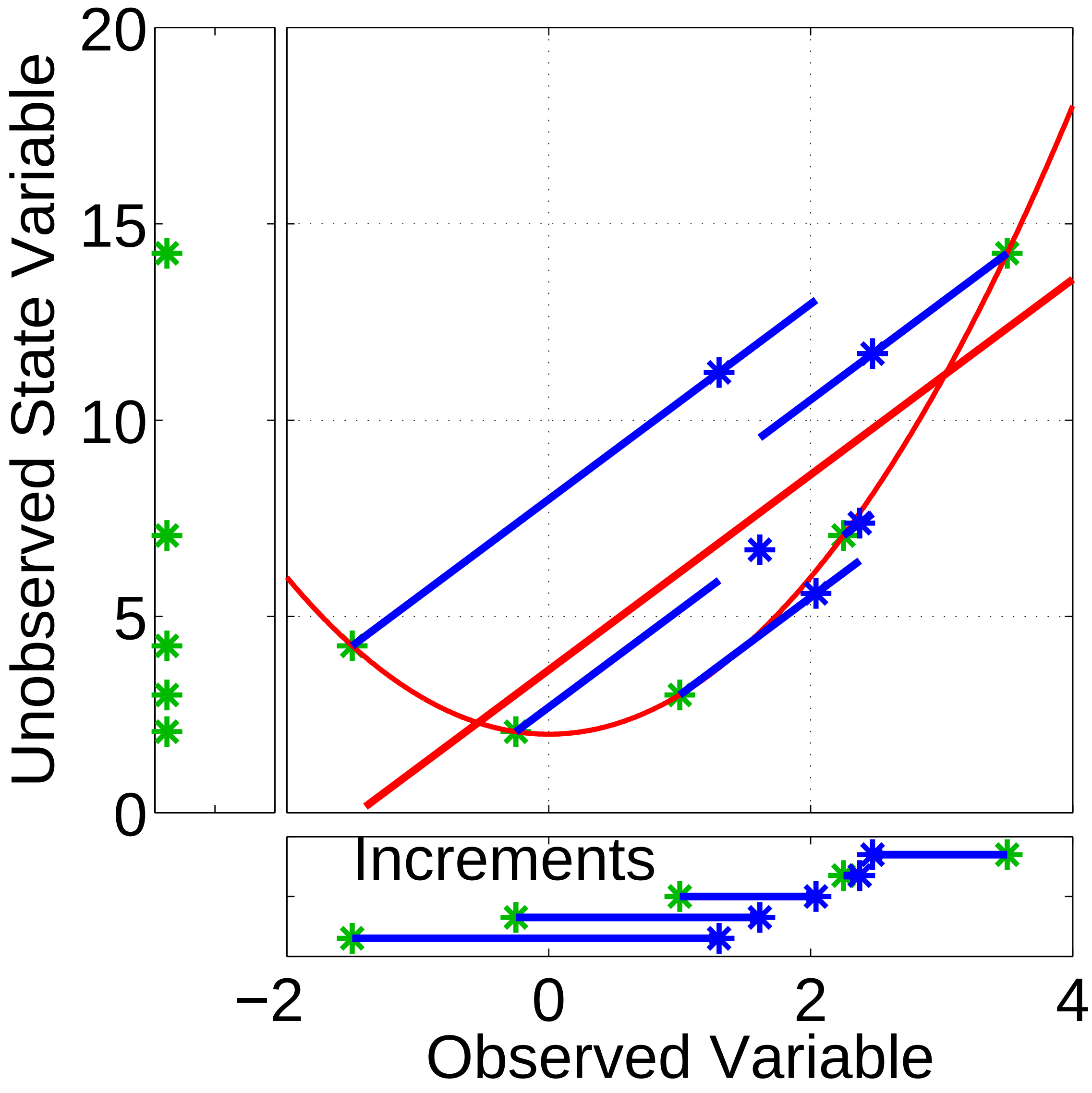
Suppose prior sample has NO noise.

But, relation between un/observed variables is non-linear.

Can minimize increments by changing pairing.

Sorting prior and posterior and pairing samples minimizes one norm of increment size (could do other methods)

Nonlinear relations between variables; sorting increments



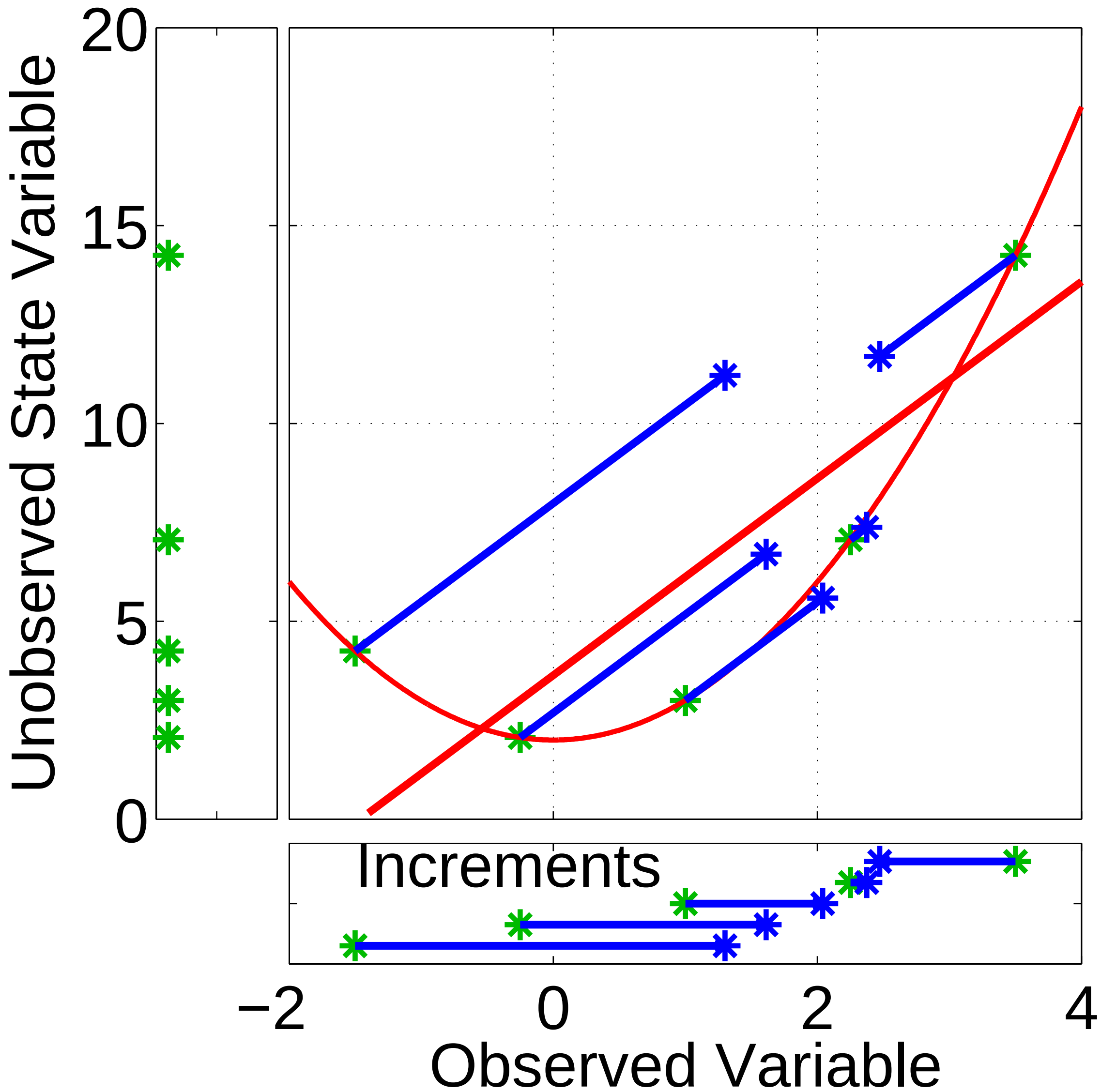
Suppose prior sample has NO noise.

Sorting prior and posterior and pairing samples minimizes one norm of increment size.

Resulting regression error is minimized.

Impact of sorting can be very large when posterior selected by ‘random’ algorithms.

Nonlinear relations between variables; sorting increments



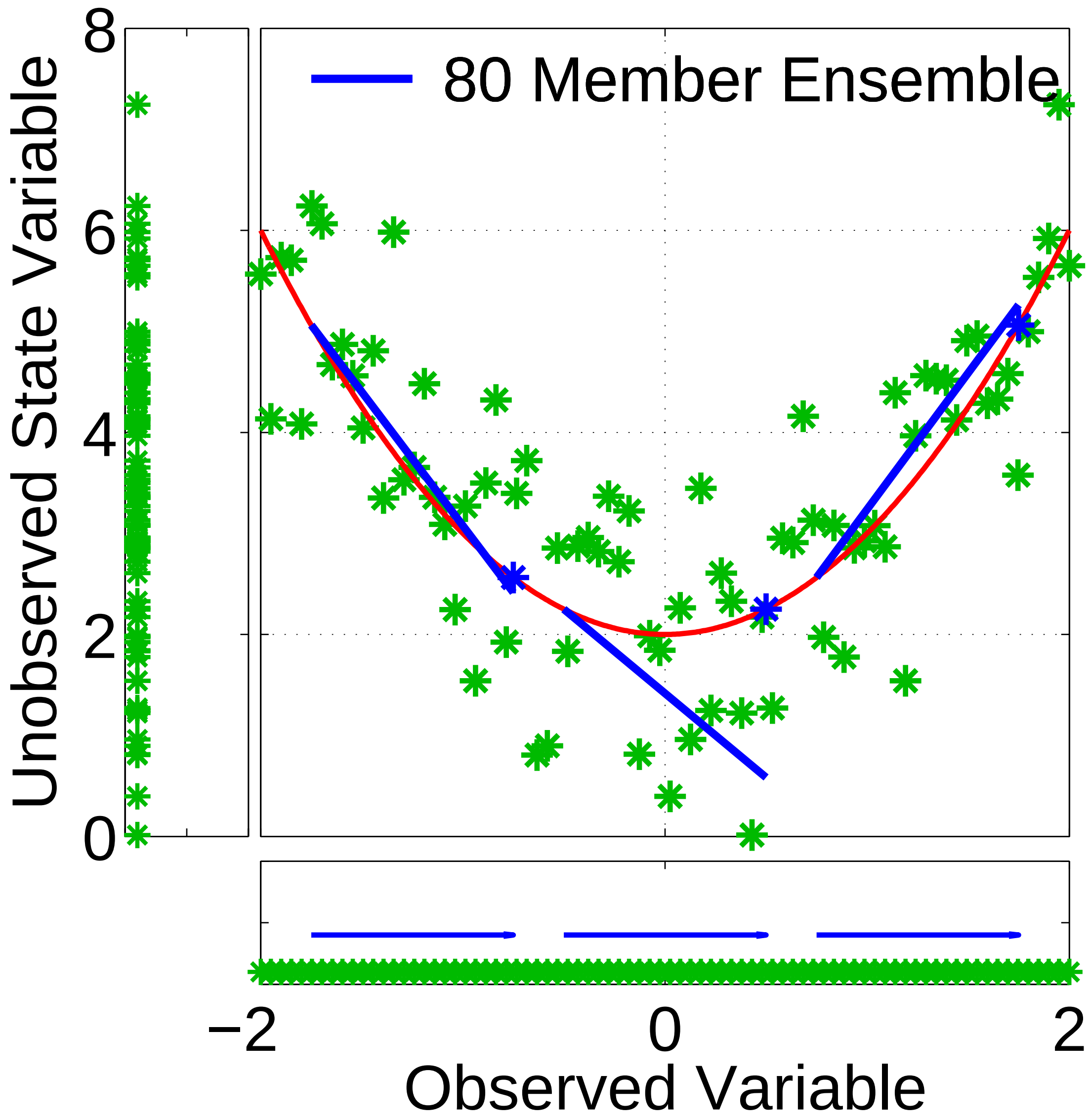
Suppose prior sample has NO noise.

Sorting prior and posterior and pairing samples minimizes one norm of increment size.

Resulting regression error is minimized.

Impact of sorting can be very large when posterior selected by ‘random’ algorithms.

Nonlinear relations between variables: Local regression



Prior sample is noisy.

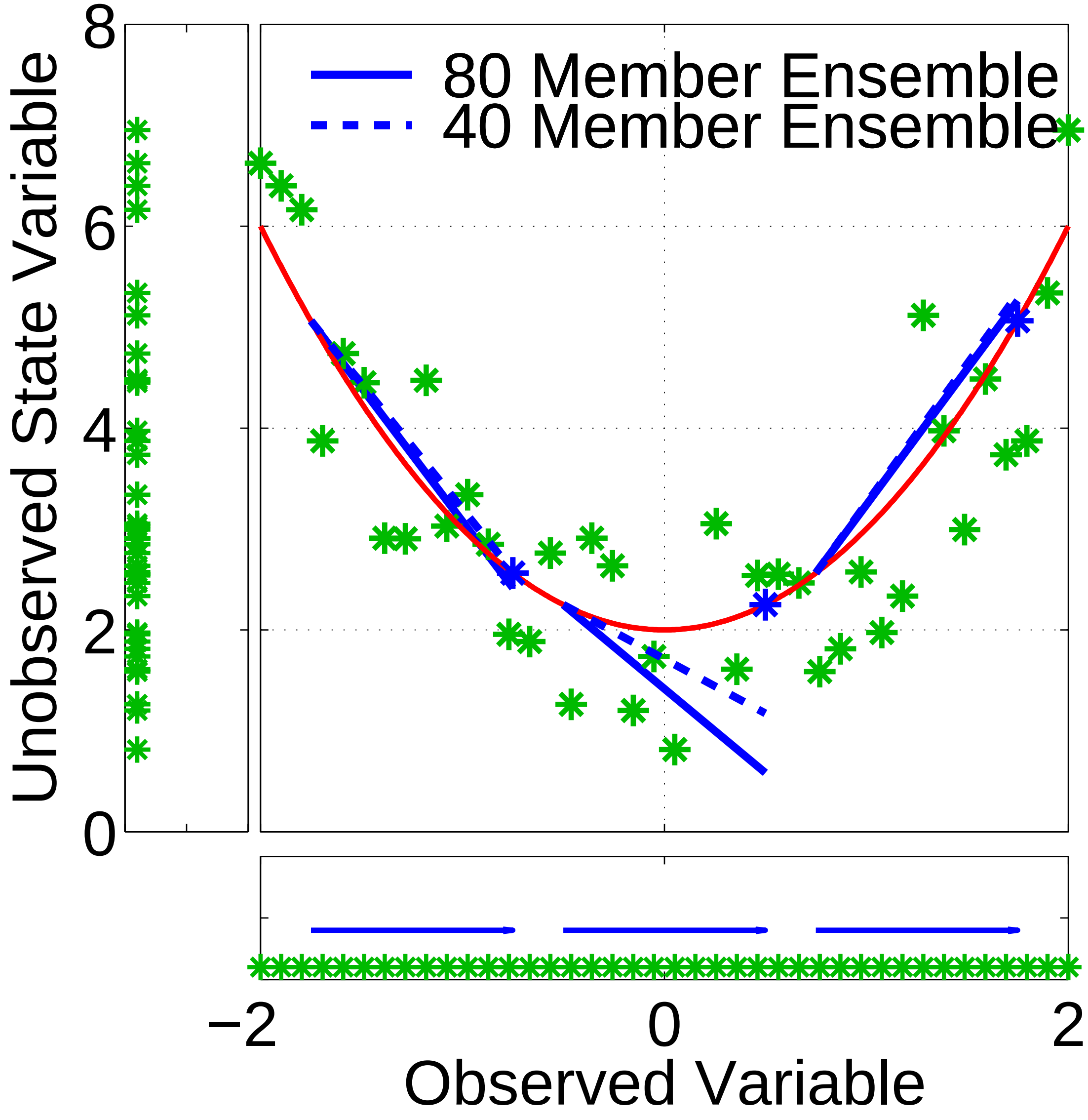
Un/observed relation is non-linear.

Doing global regression would be BAD here.

Can do regression only for points that lie in range of update increment.

Could also pick local sets in other ways.

Nonlinear relations between variables; Local regression



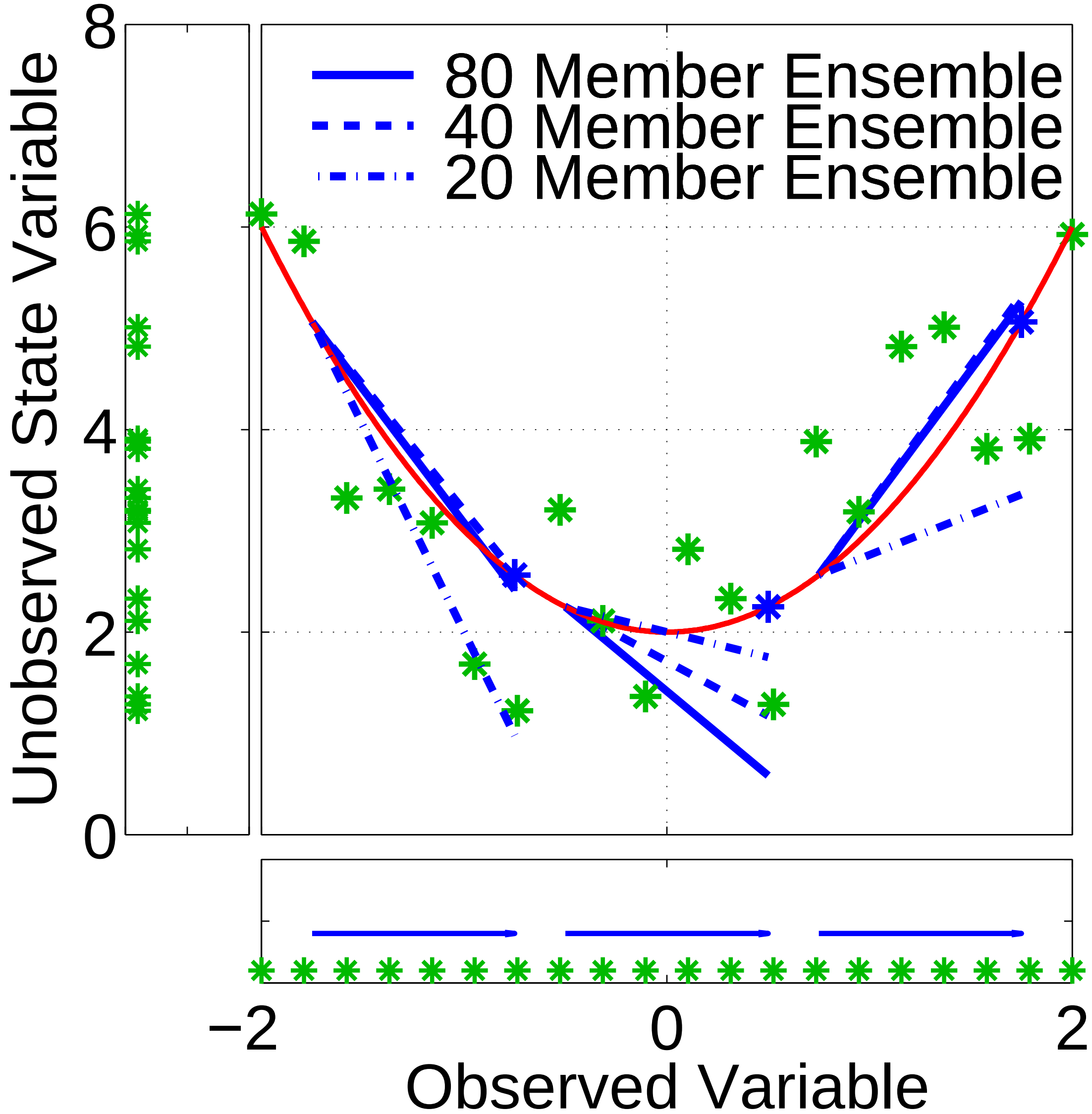
Prior sample is noisy.

Un/observed relation is non-linear.

For larger ensembles, local regressions can work well.

Error is largest where signal is weakest (near bottom of parabola here).

Nonlinear relations between variables; Local regression



Prior sample is noisy.

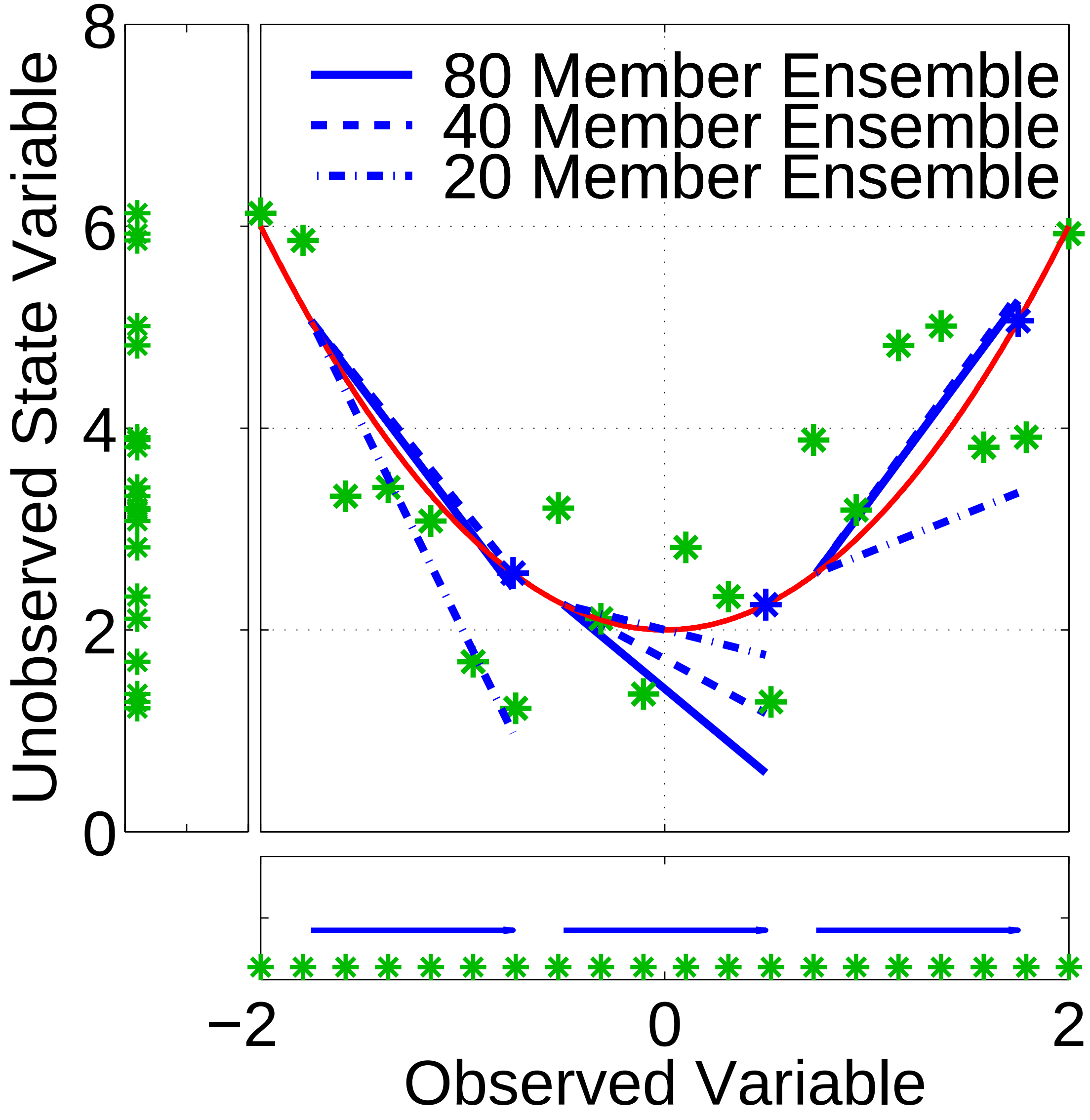
Un/observed relation is non-linear.

As sample size decreases, error grows.

(Except where it was rotten to start).

Applications where local regression is useful are unknown to me.

Nonlinear relations between variables; Local regression

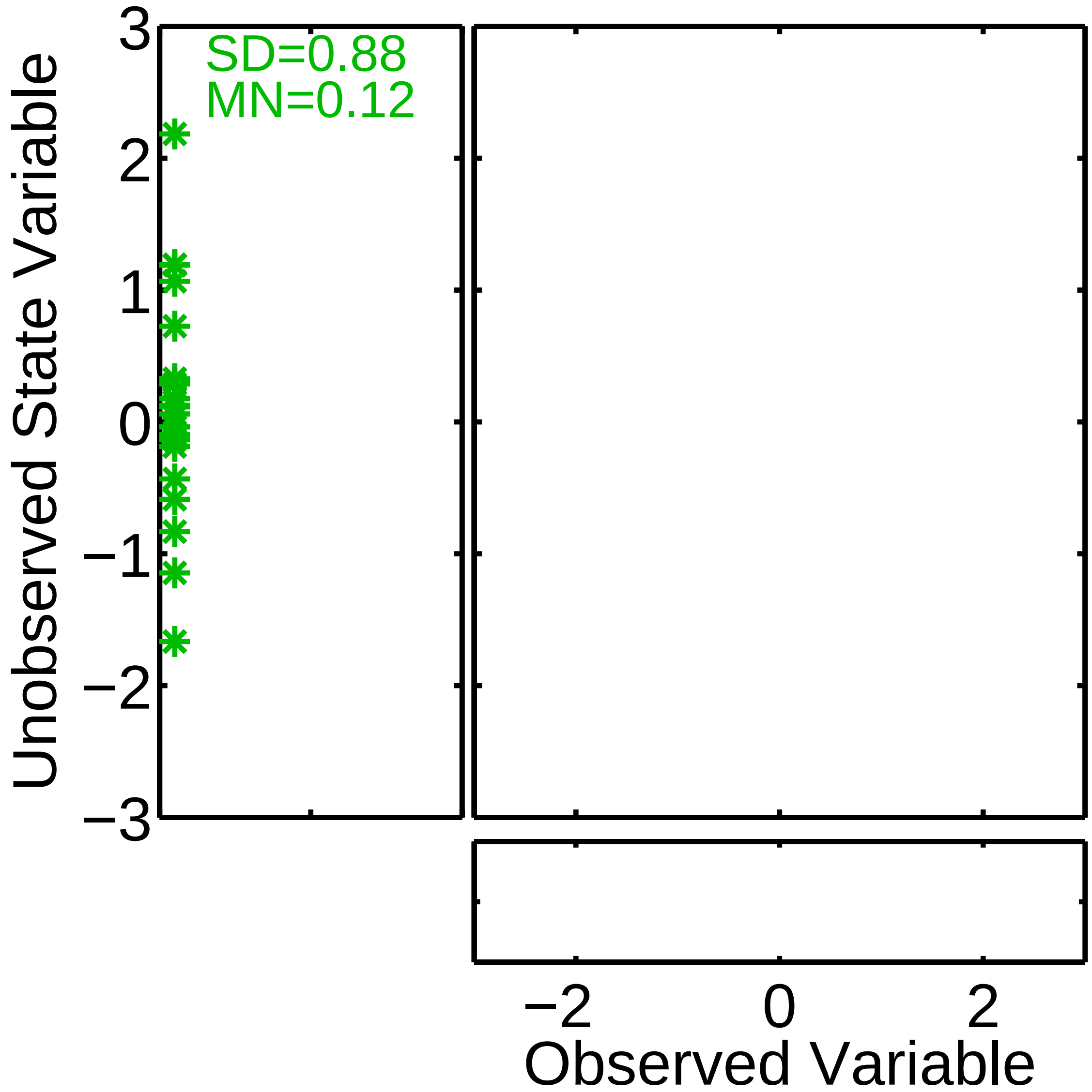


Prior sample is noisy.

Un/observed relation is non-linear.

Serious issues may exist if local regression is used with multiple unobserved state variables.

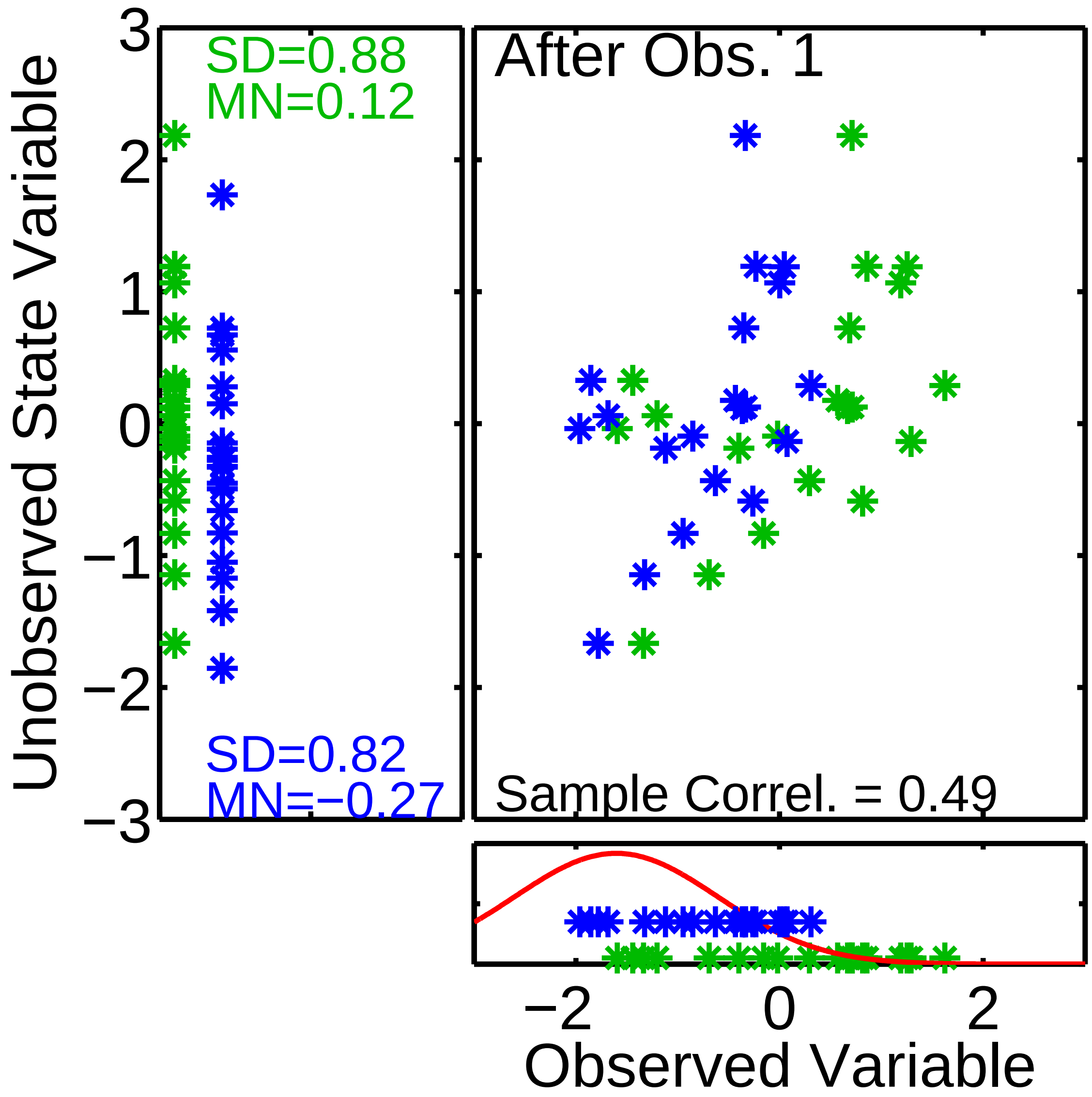
Regression sampling error and filter divergence



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

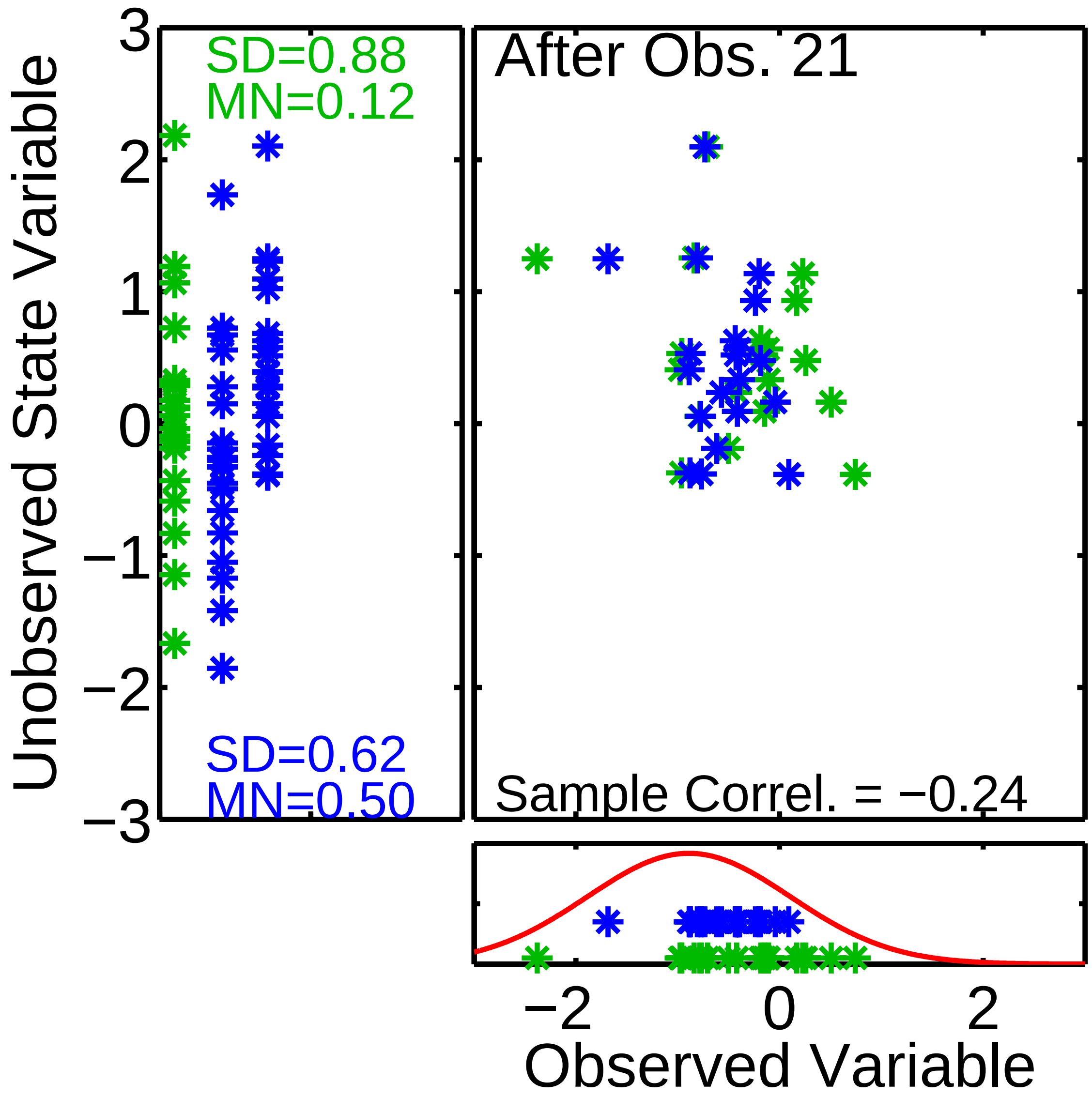


Suppose unobserved state variable is known to be unrelated to set of observed variables.

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

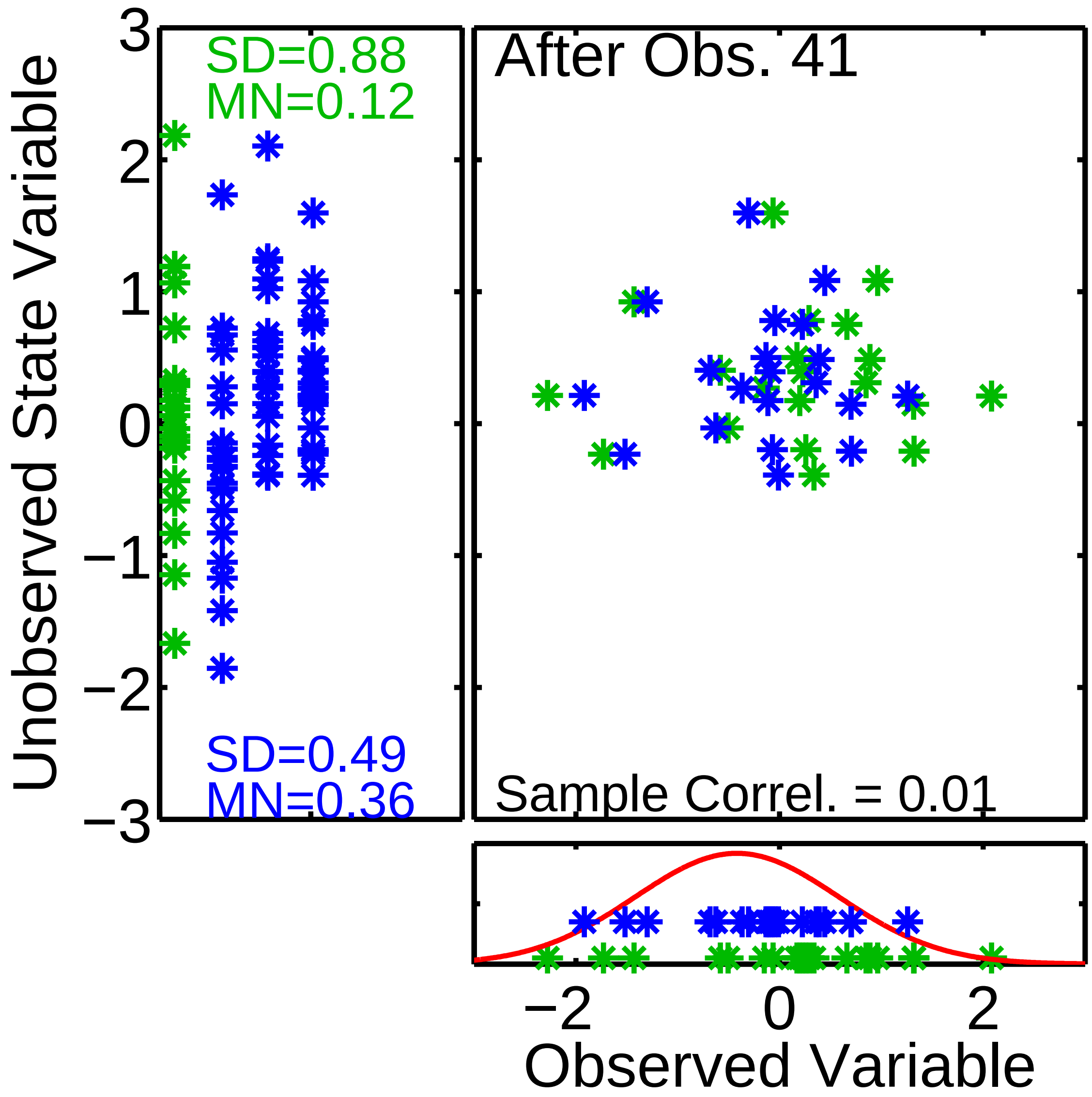


Suppose unobserved state variable is known to be unrelated to set of observed variables.

Unobserved variable should remain unchanged

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

Regression sampling error and filter divergence



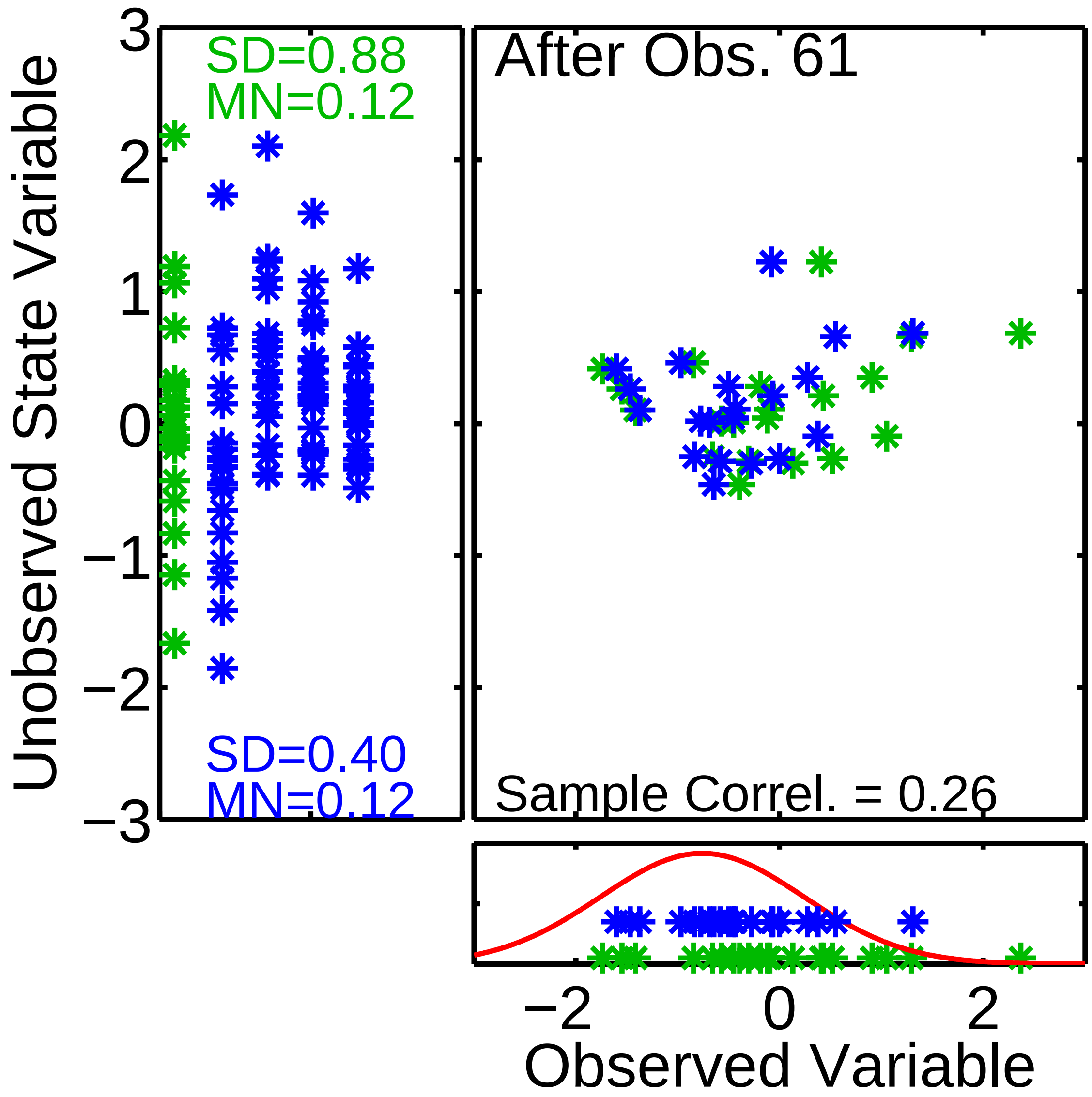
Suppose unobserved state variable is known to be unrelated to set of observed variables.

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



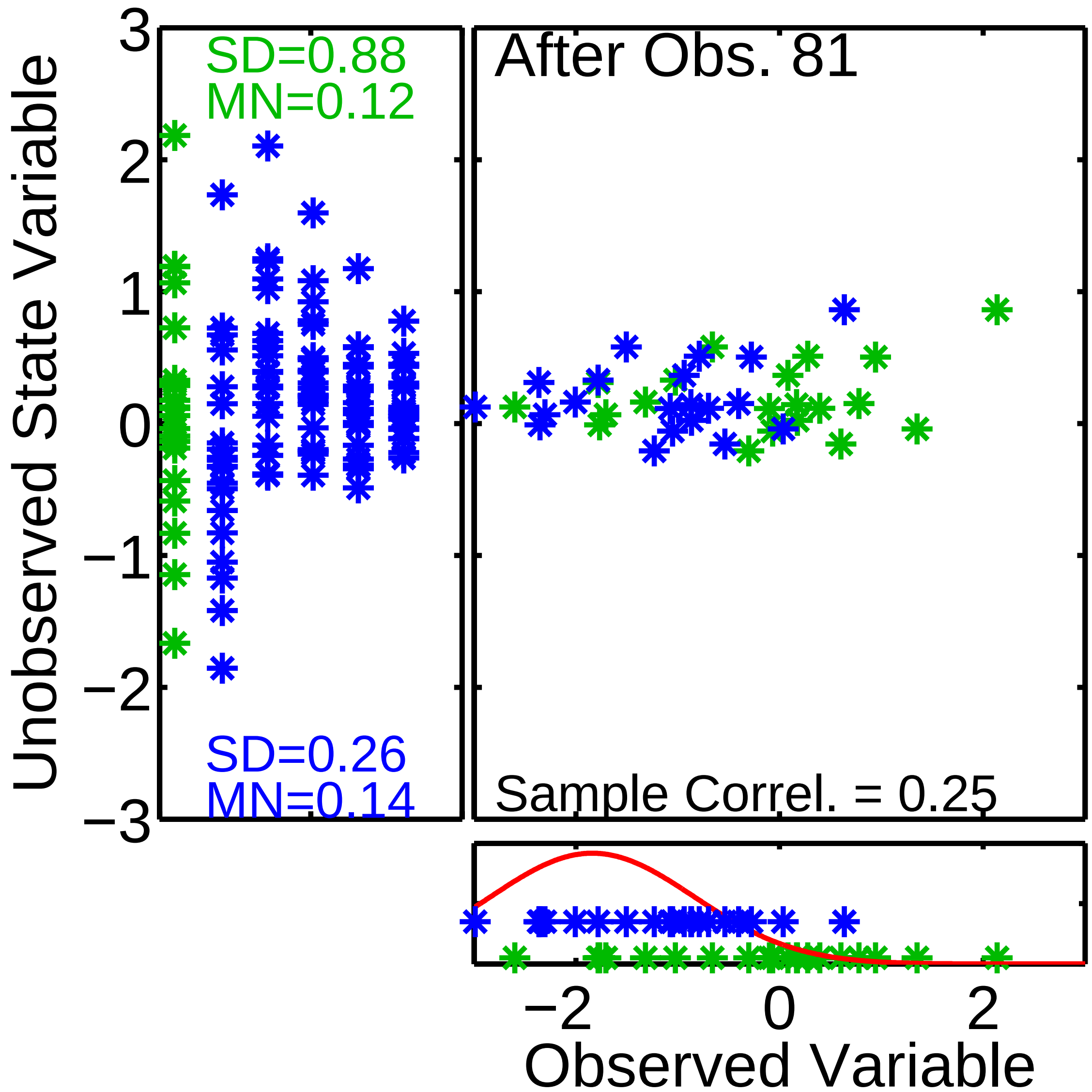
Suppose unobserved state variable is known to be unrelated to set of observed variables.

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



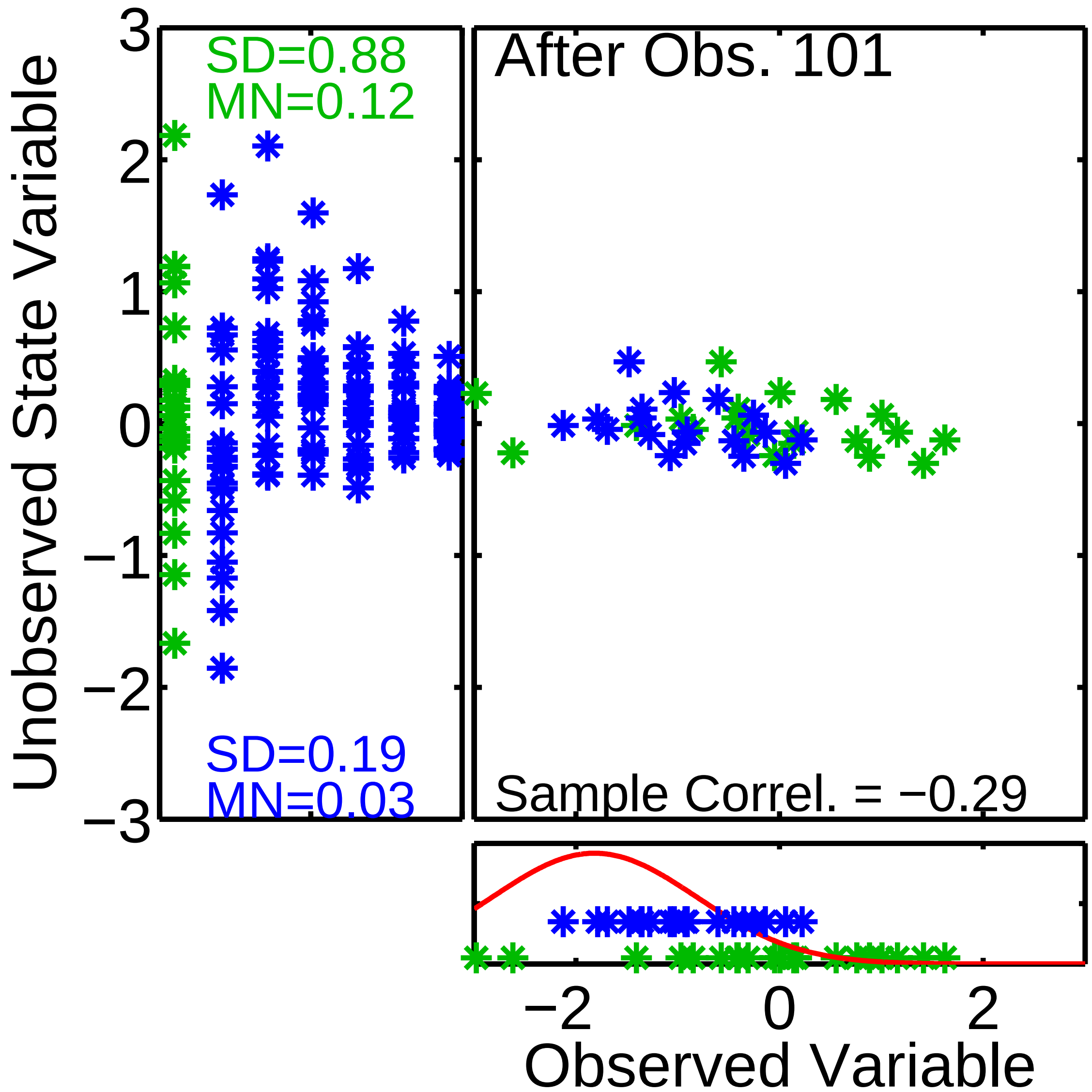
Suppose unobserved state variable is known to be unrelated to set of observed variables.

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



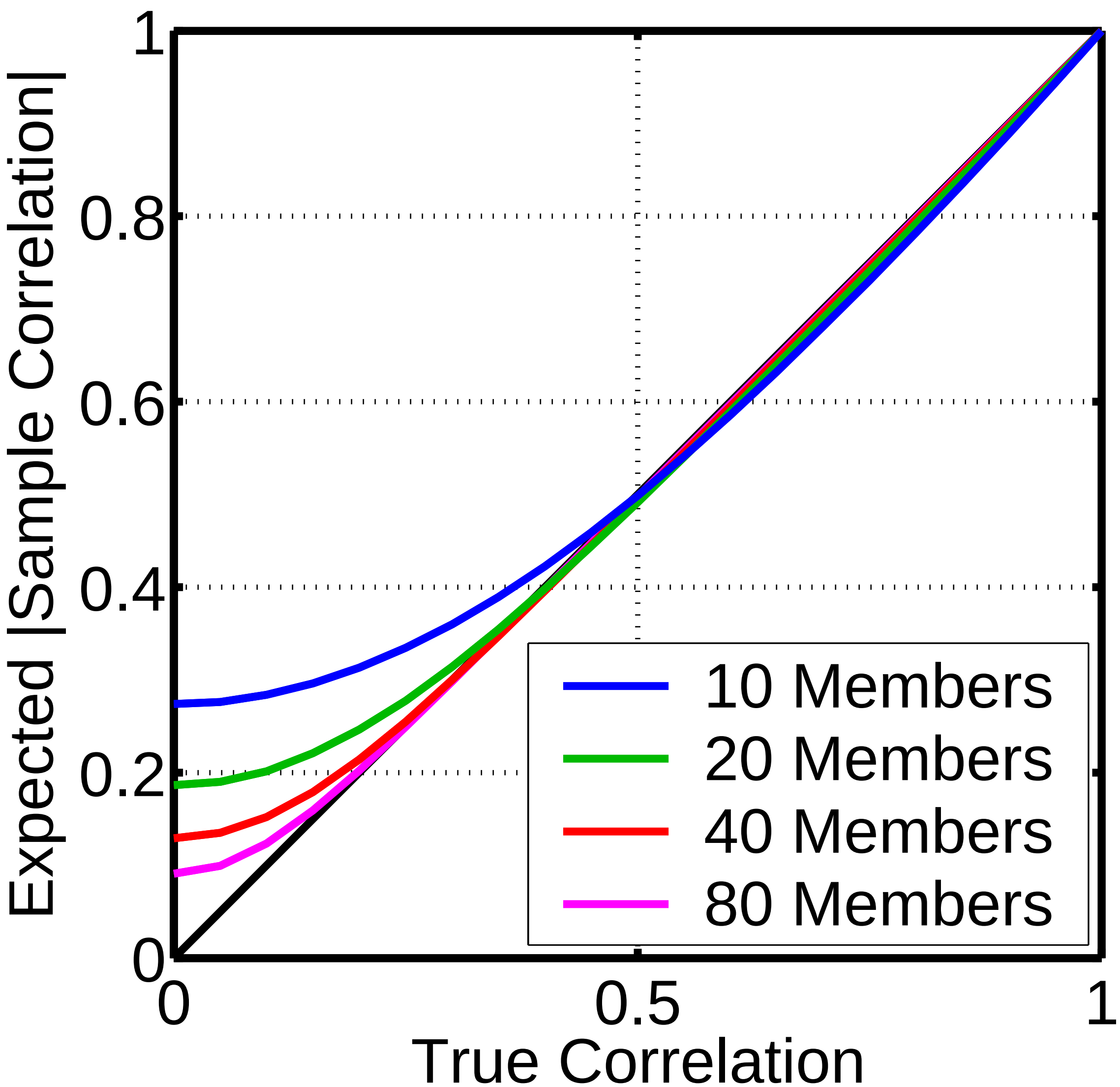
Suppose unobserved state variable is known to be unrelated to set of observed variables.

Estimates of unobs. become too confident

Give progressively less weight to any meaningful observations.

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

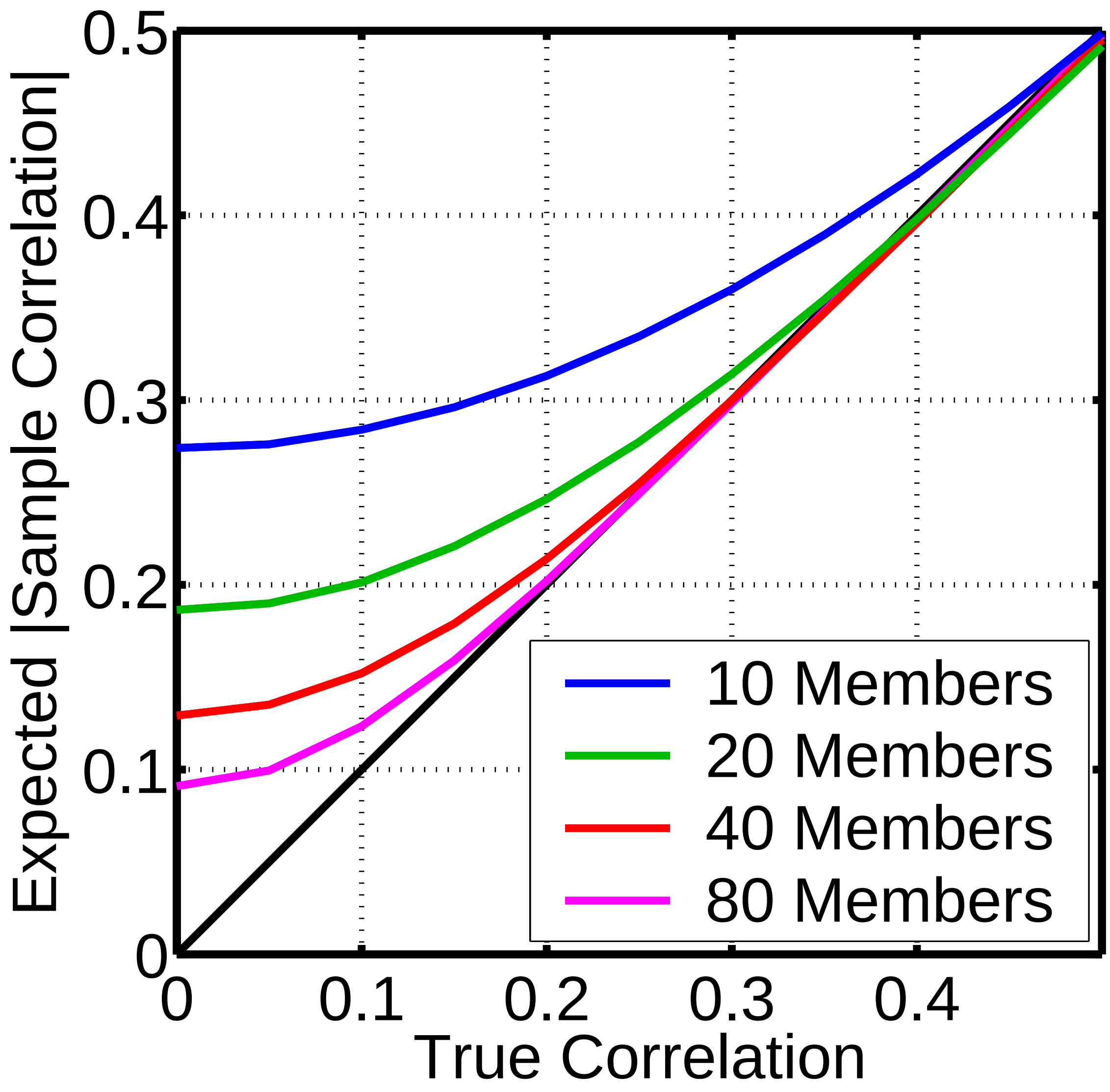
Regression sampling error and filter divergence



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

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

Regression sampling error and filter divergence



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

For negligible true correlations, errors are still significant even for 80 member ensembles.

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.

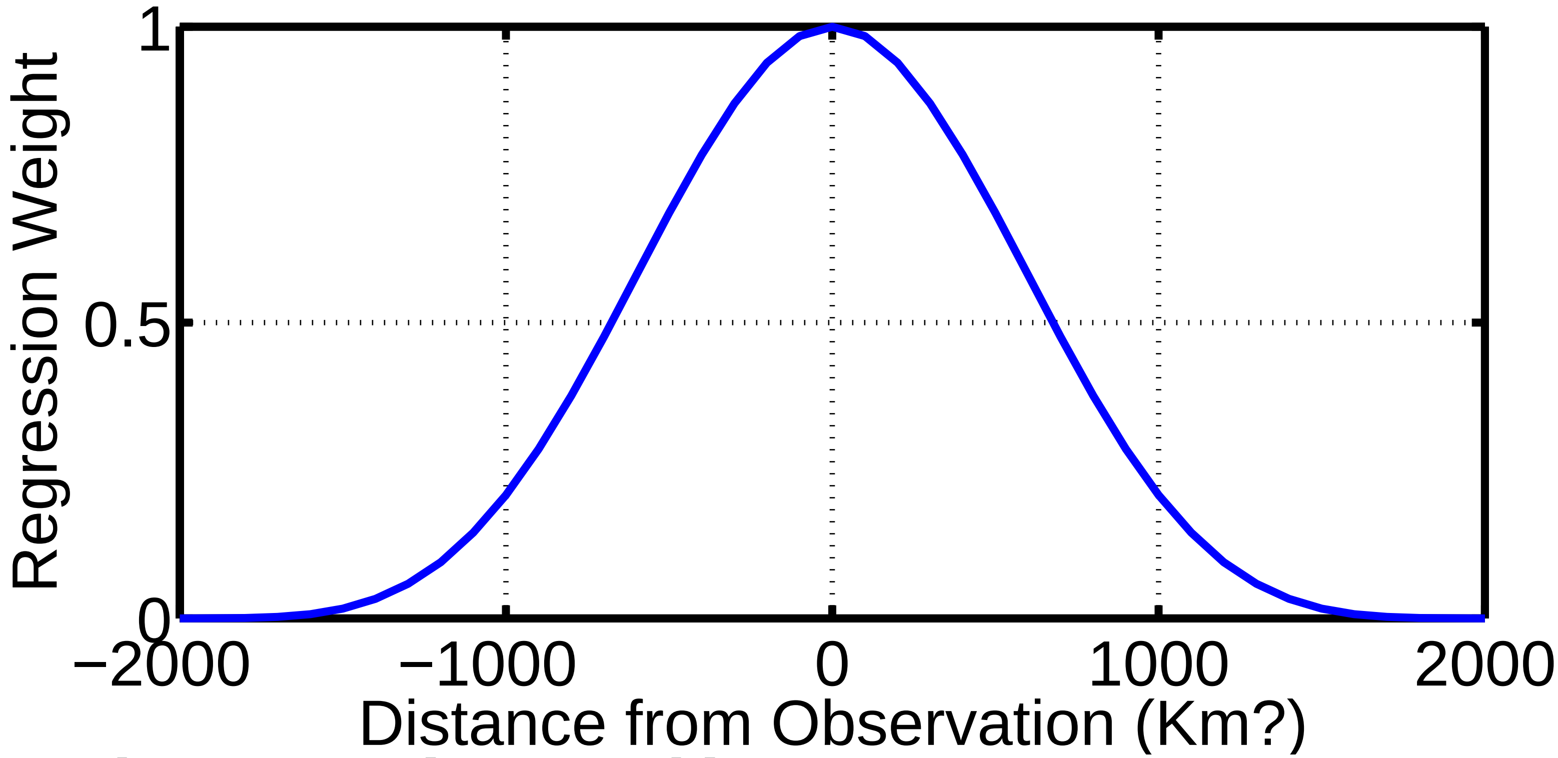
2. Use larger ensembles to limit sampling error.

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.



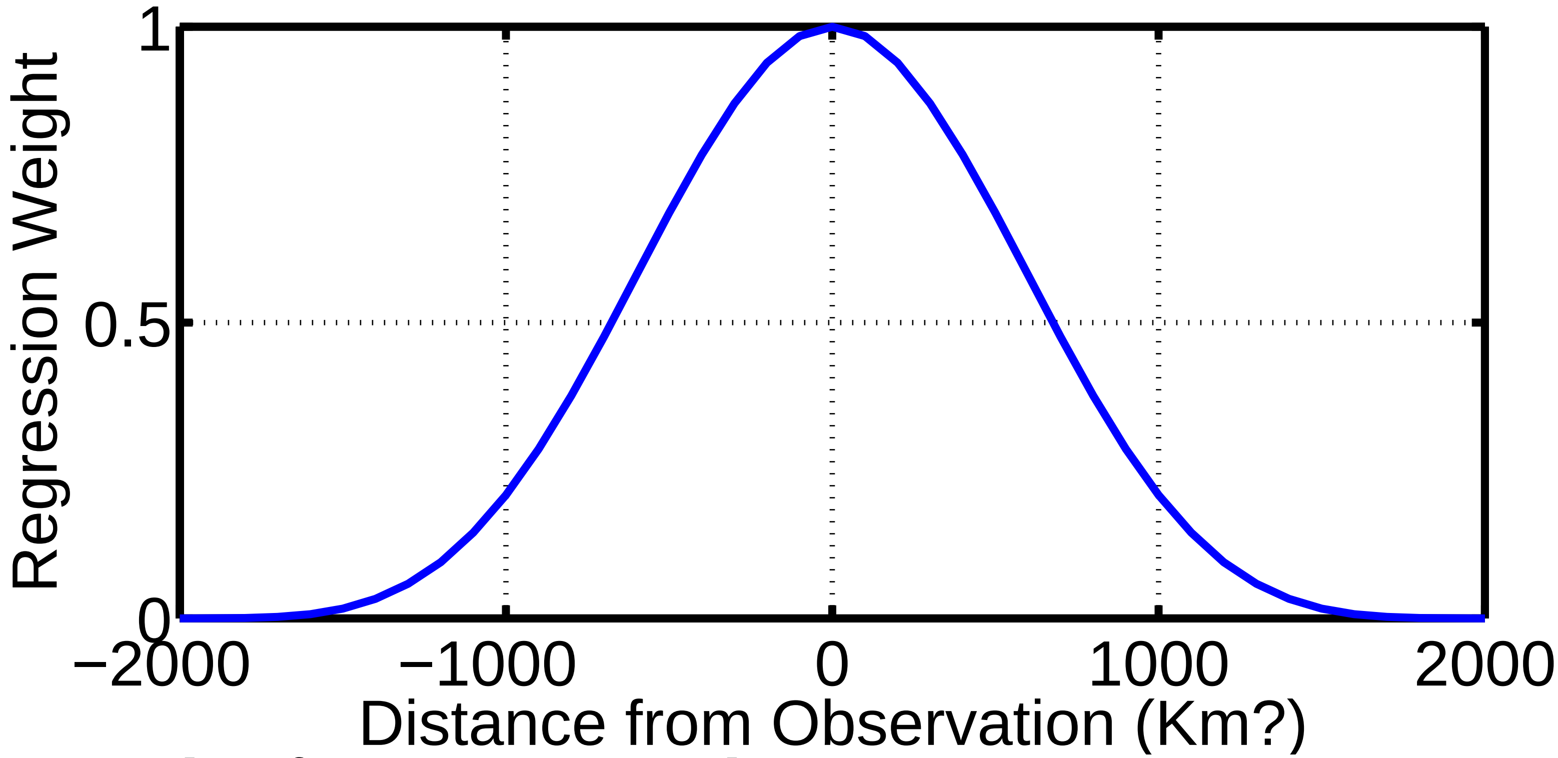
Atmospheric assimilation problems.

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.

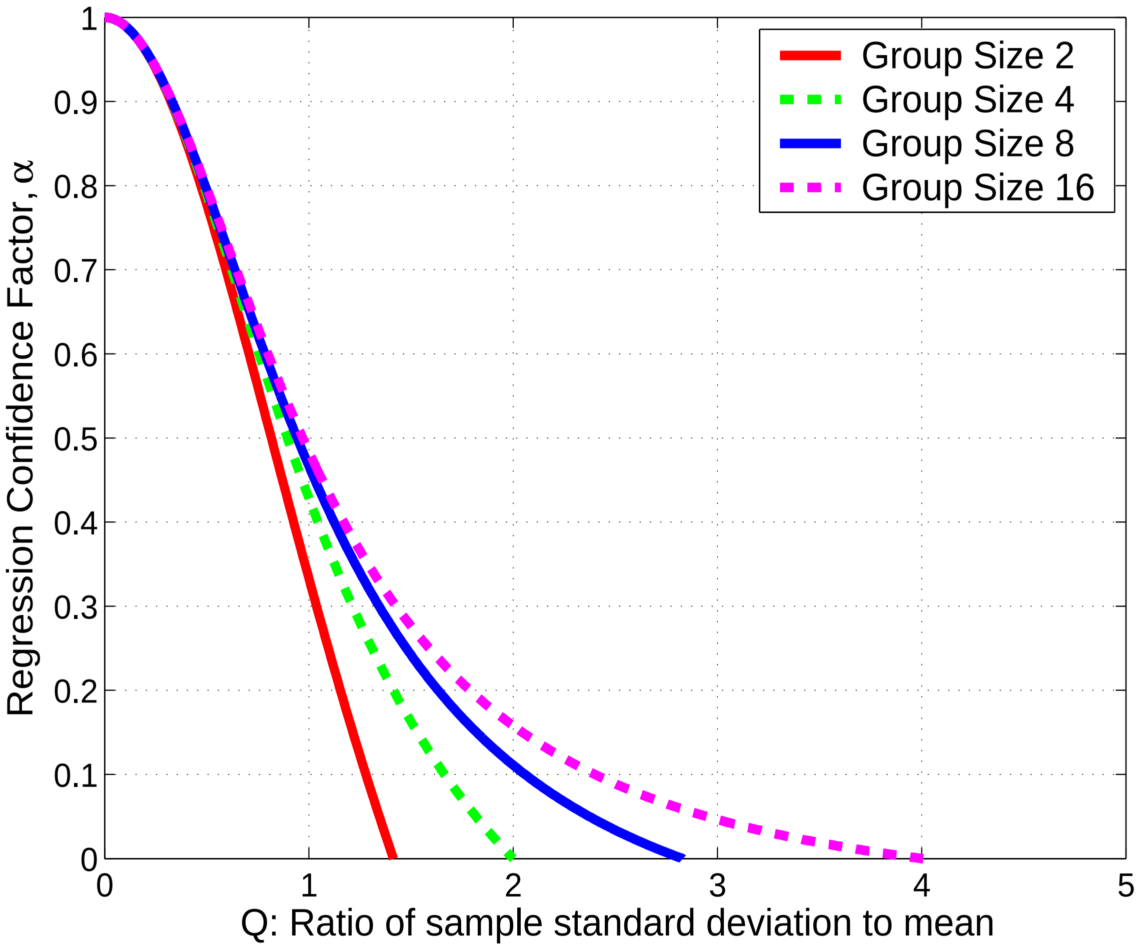
Find regression confidence factor α (weight) that minimizes:

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

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

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

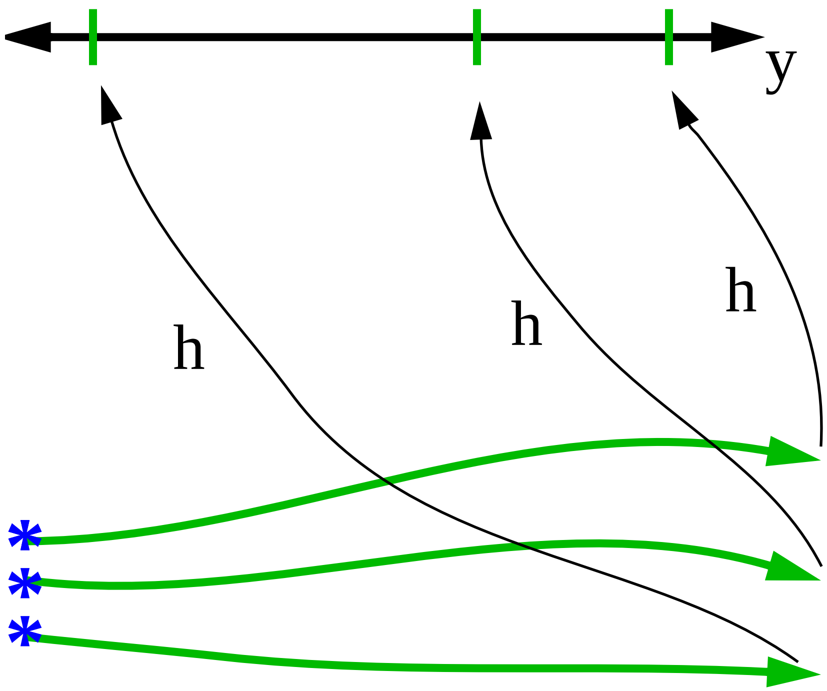
1. Use model to advance ensemble (3 members here) to time at which next observation becomes available.

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

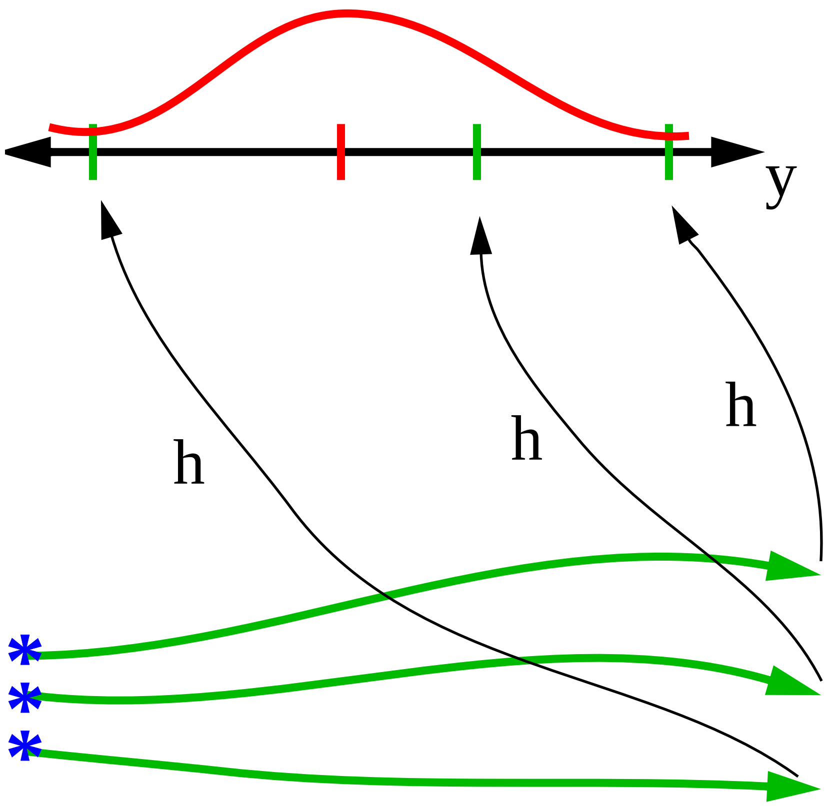
2. Get prior ensemble sample of observation, y=h(x), by applying forward operator h to each ensemble member.



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

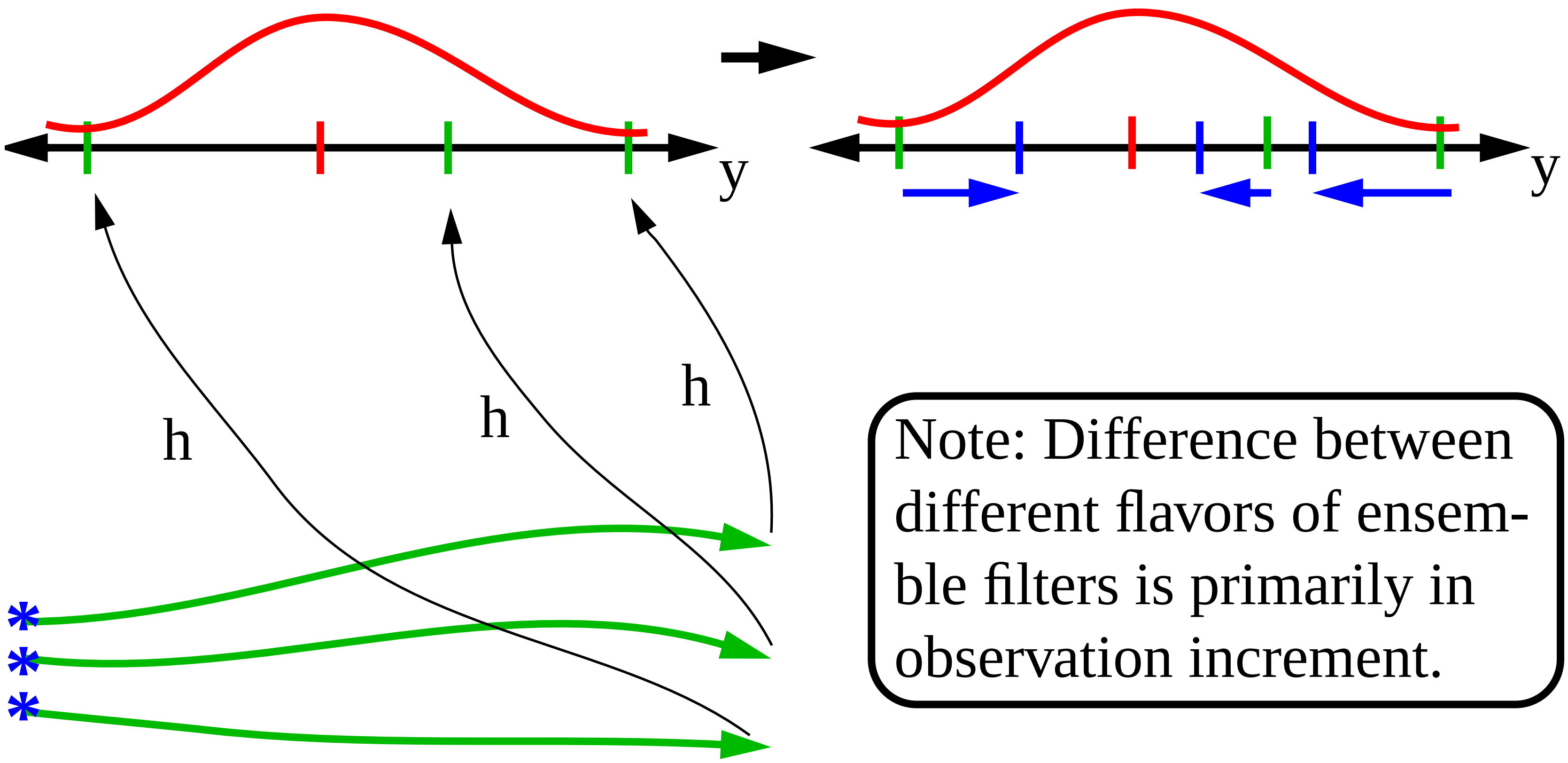
How an Ensemble Filter Works for Geophysical Data Assimilation

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



How an Ensemble Filter Works for Geophysical Data Assimilation

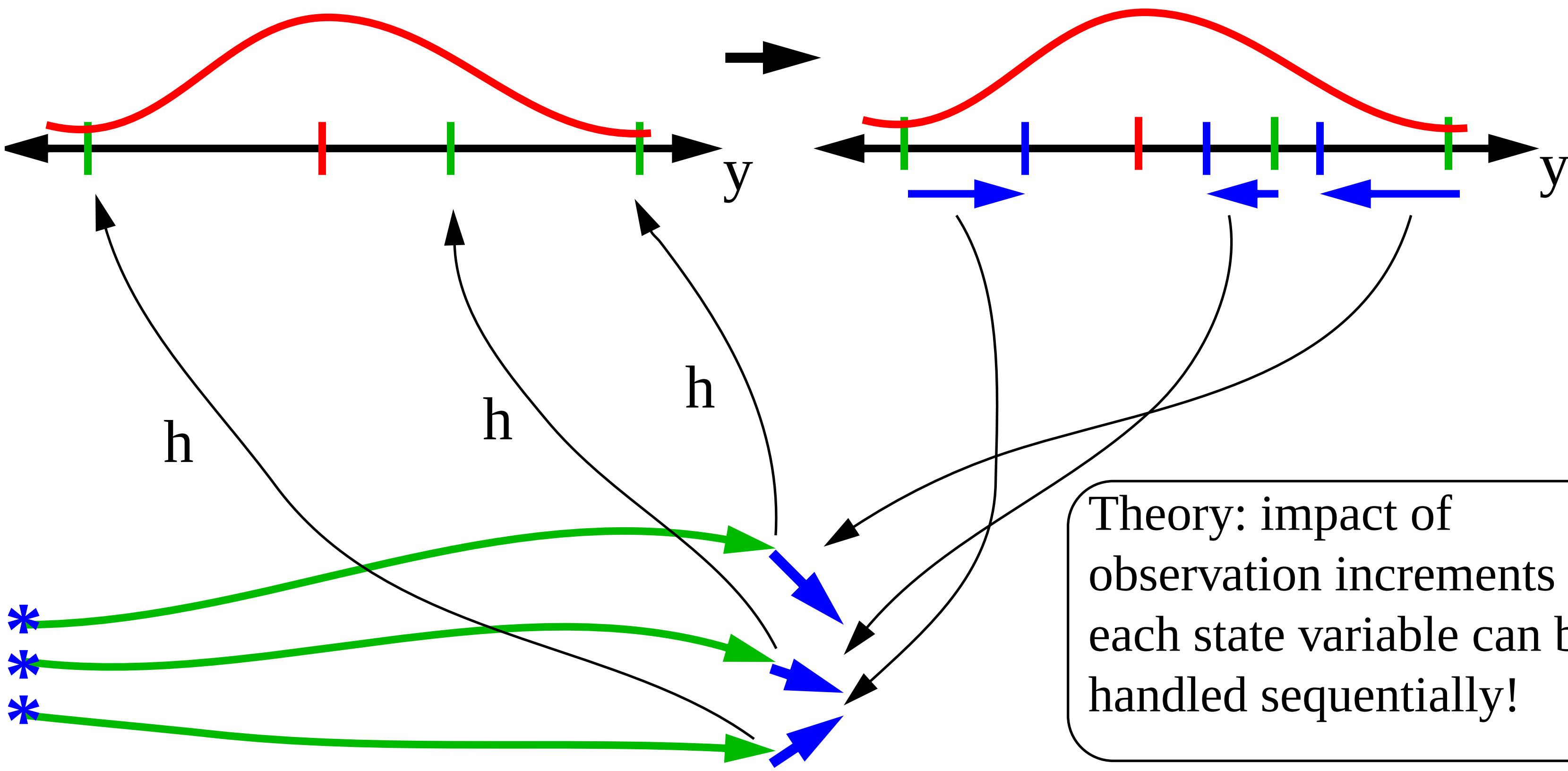
4. Find increment for each prior observation ensemble
(this is a scalar problem for uncorrelated observation errors).



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

How an Ensemble Filter Works for Geophysical Data Assimilation

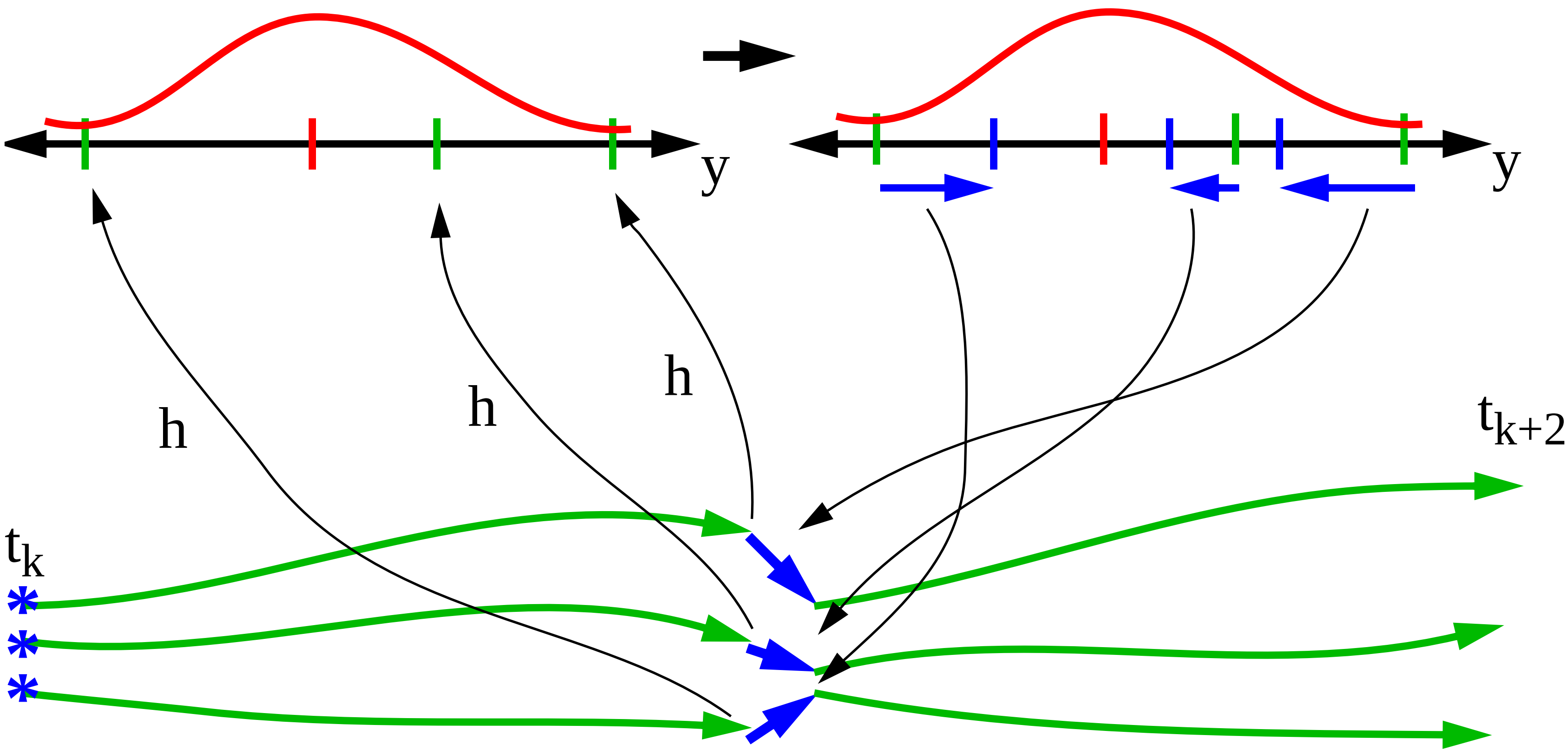
5. Use ensemble samples of y and each state variable to linearly regress observation increments onto state variable increments.



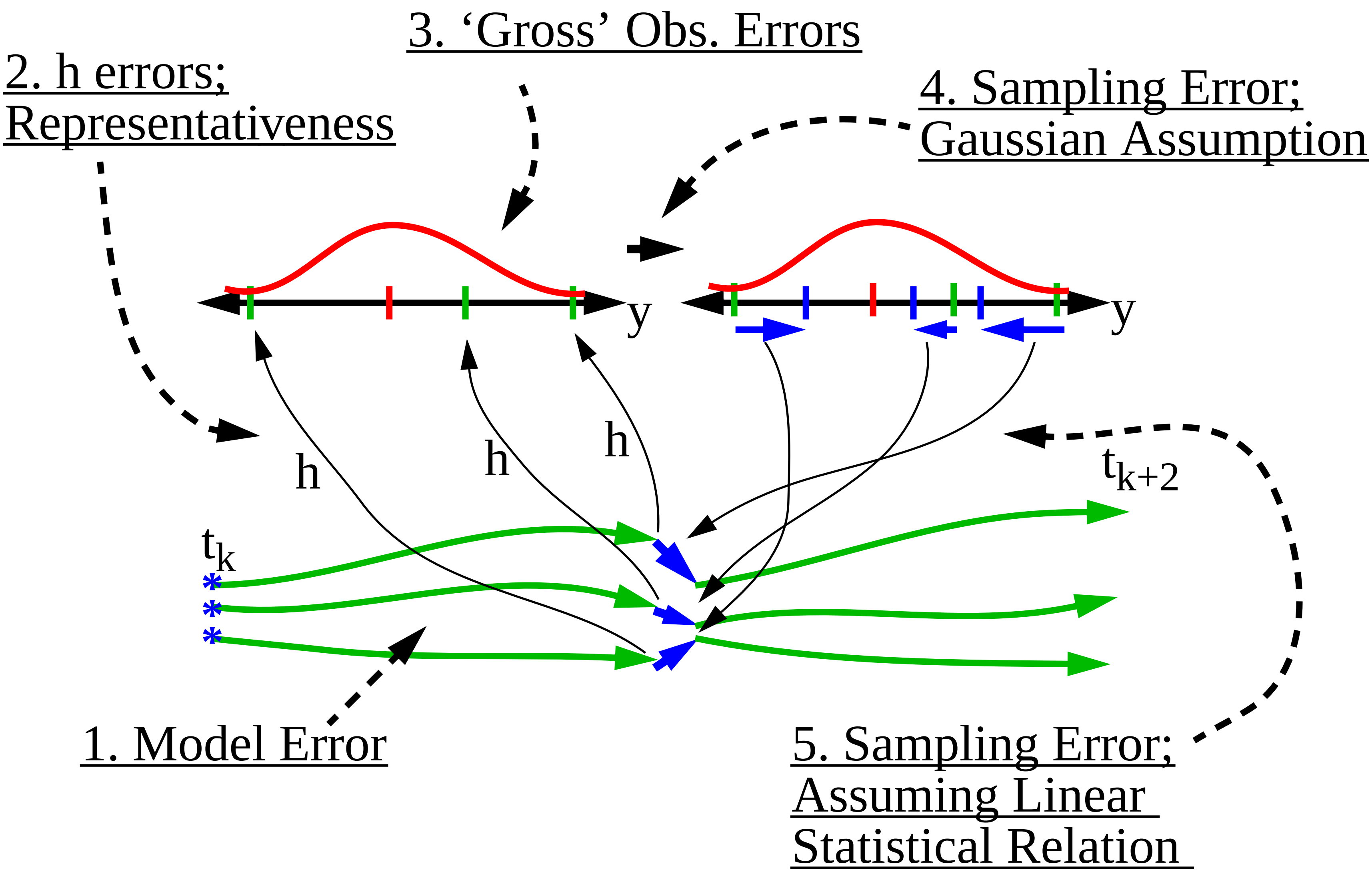
Theory: impact of observation increments on each state variable can be handled sequentially!

How an Ensemble Filter Works for Geophysical Data Assimilation

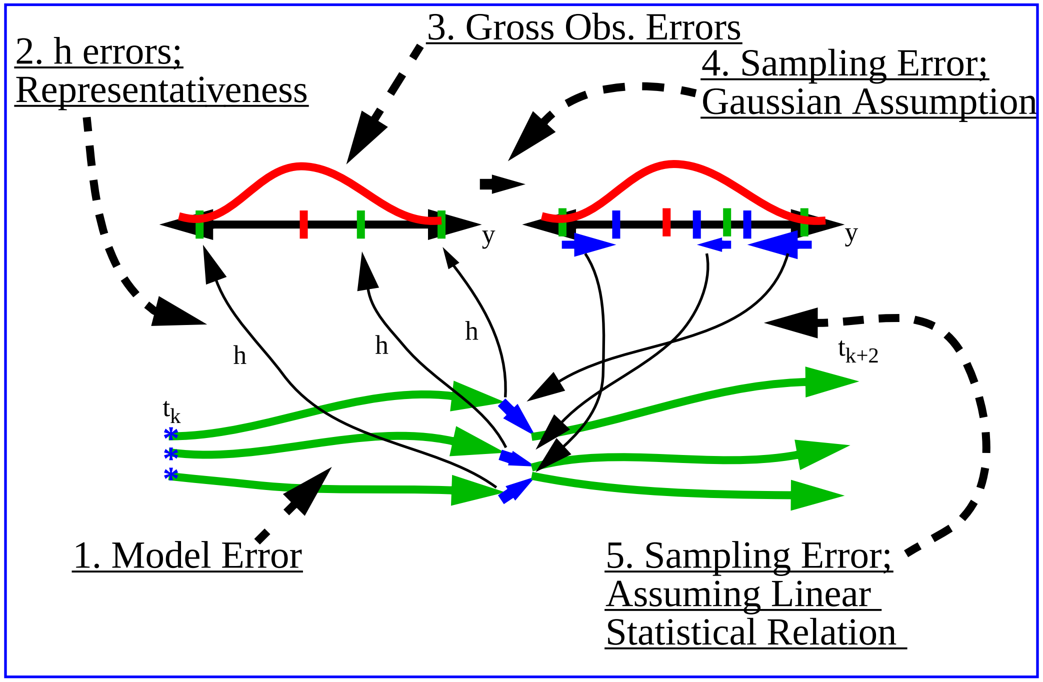
6. When all ensemble members for each state variable are updated, have a new analysis. Integrate to time of next observation...



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.

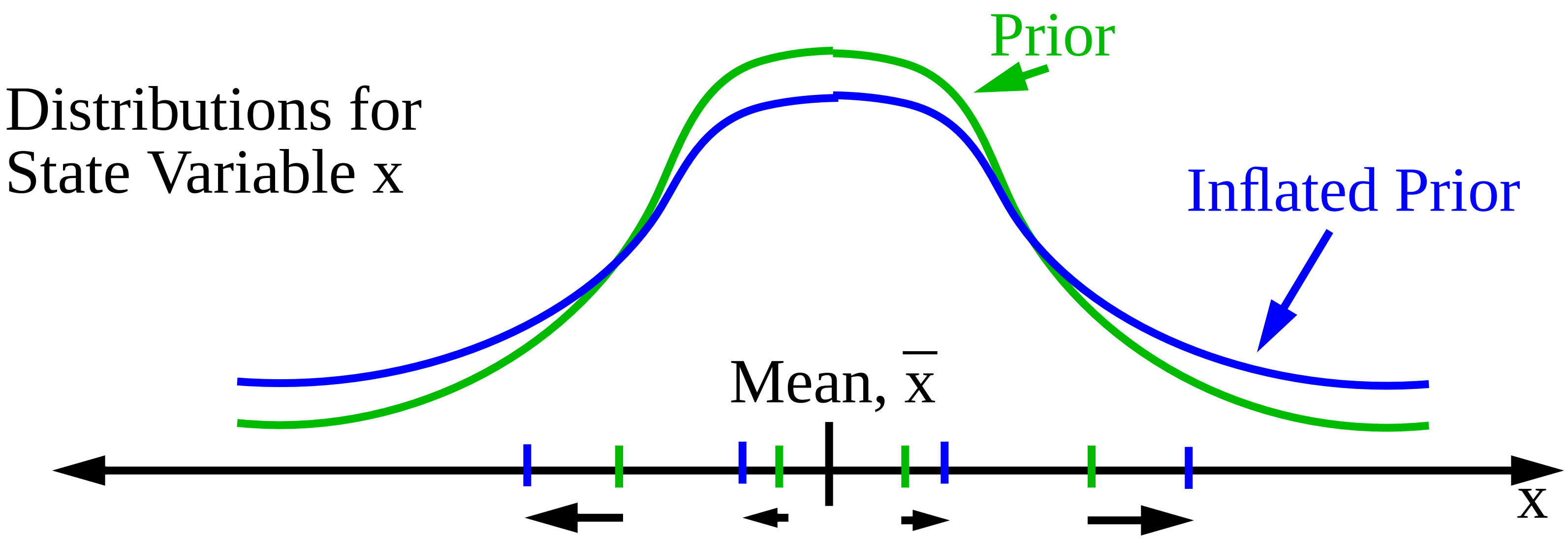
Often smoothly decrease impact to 0 as function of distance.

Model/Filter Error; Filter Divergence and Covariance Inflation

1. Model imperfections lead to erroneous prior distributions.

2. Filter sampling errors lead to too little variance in priors.

3. Covariance inflation one way to attack this.



4. Inflated variance is λ times raw variance.

5. For ensemble member i, $inflate(x_i) = \sqrt{\lambda}(x_i - \bar{x}) + \bar{x}$.

Physical Space Covariance Inflation

Capabilities:

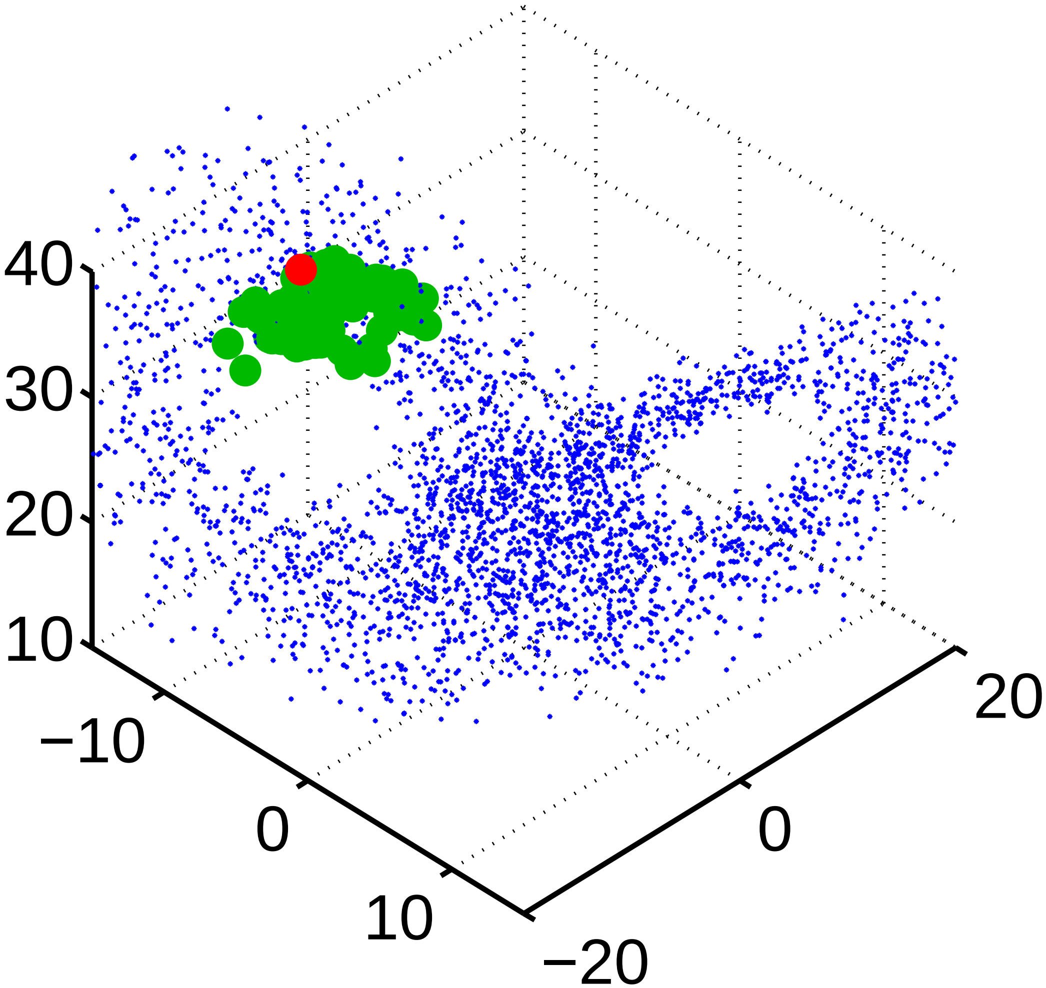
1. Can be very effective for a variety of models.
2. Can maintain linear balances.
3. Stays on local flat manifolds.
4. Simple and inexpensive.

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.

Phase 3: Generalize to geophysical models and observations

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



Observation in red.

Prior ensemble in green.

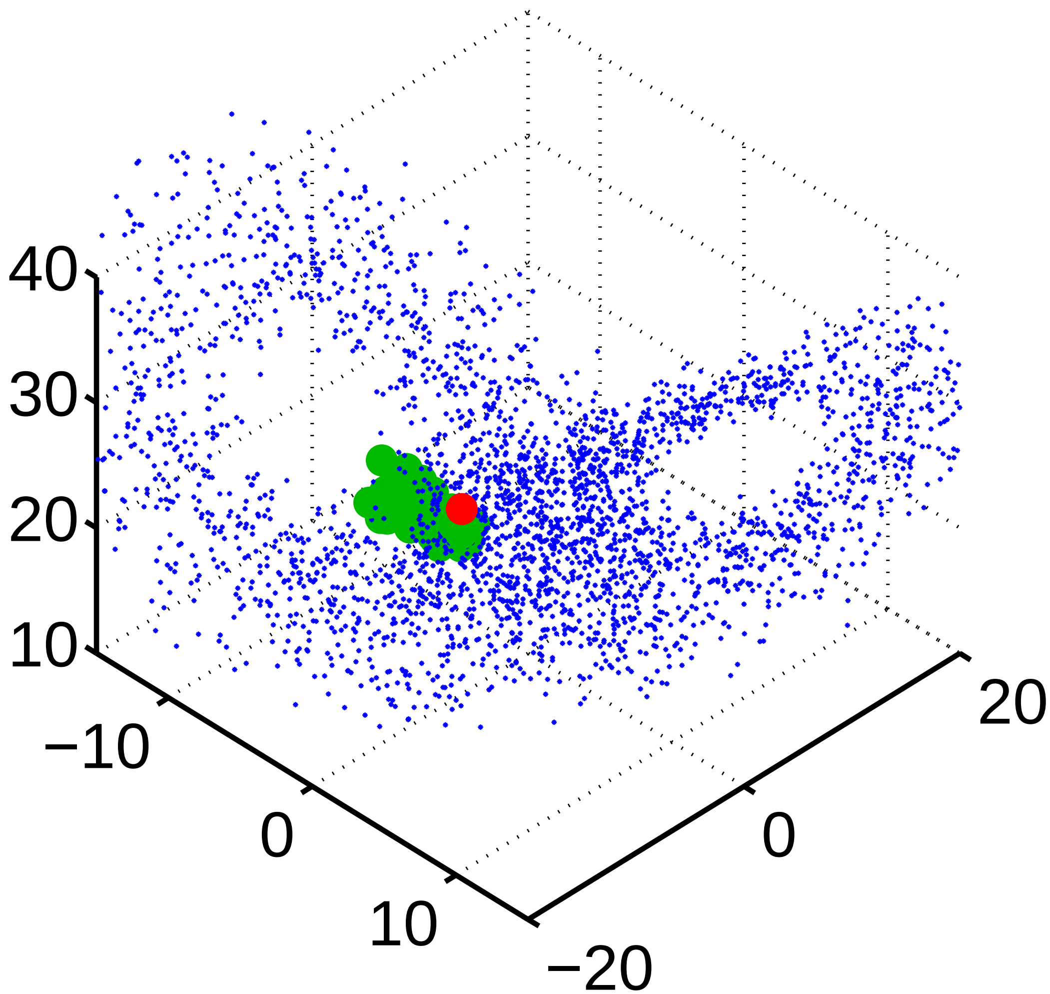
Observing all three state variables.

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.

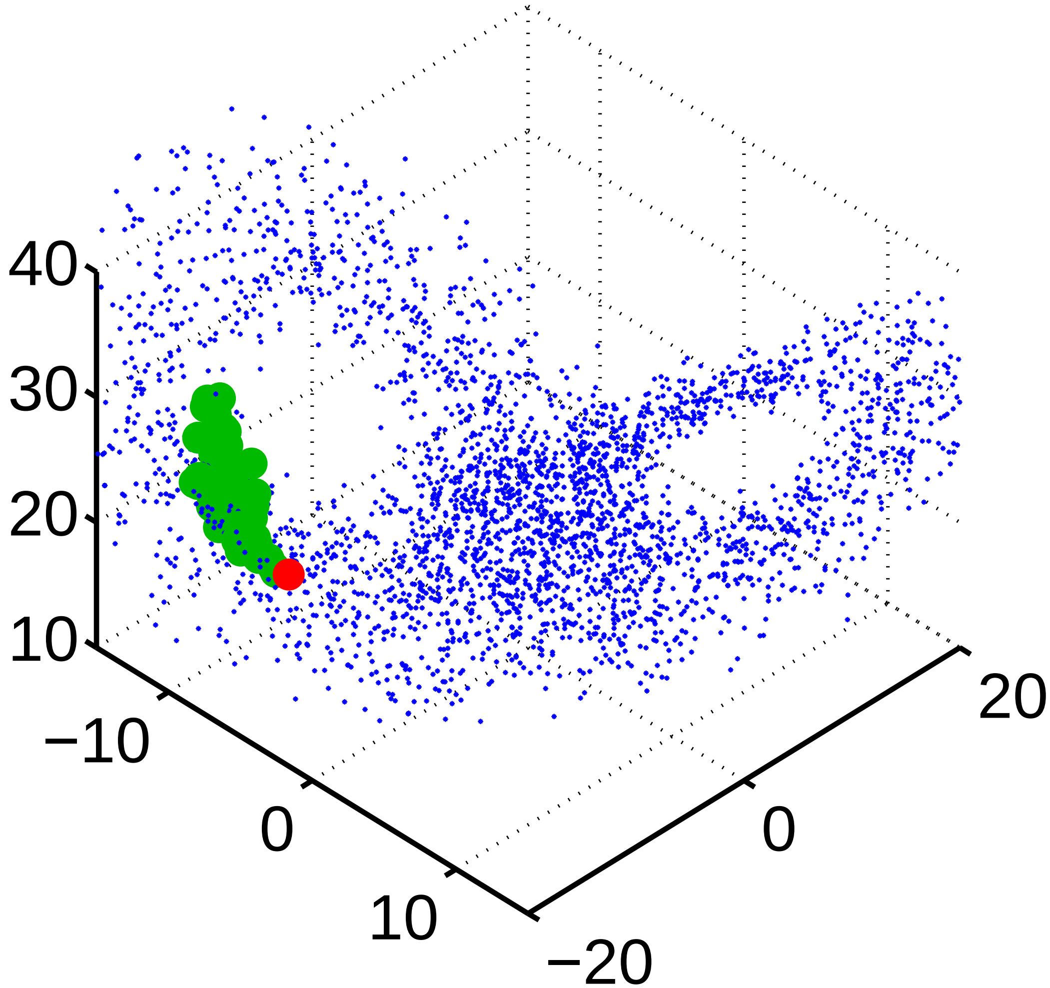


Observation in red.

Prior ensemble in green.

Phase 3: Generalize to geophysical models and observations

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

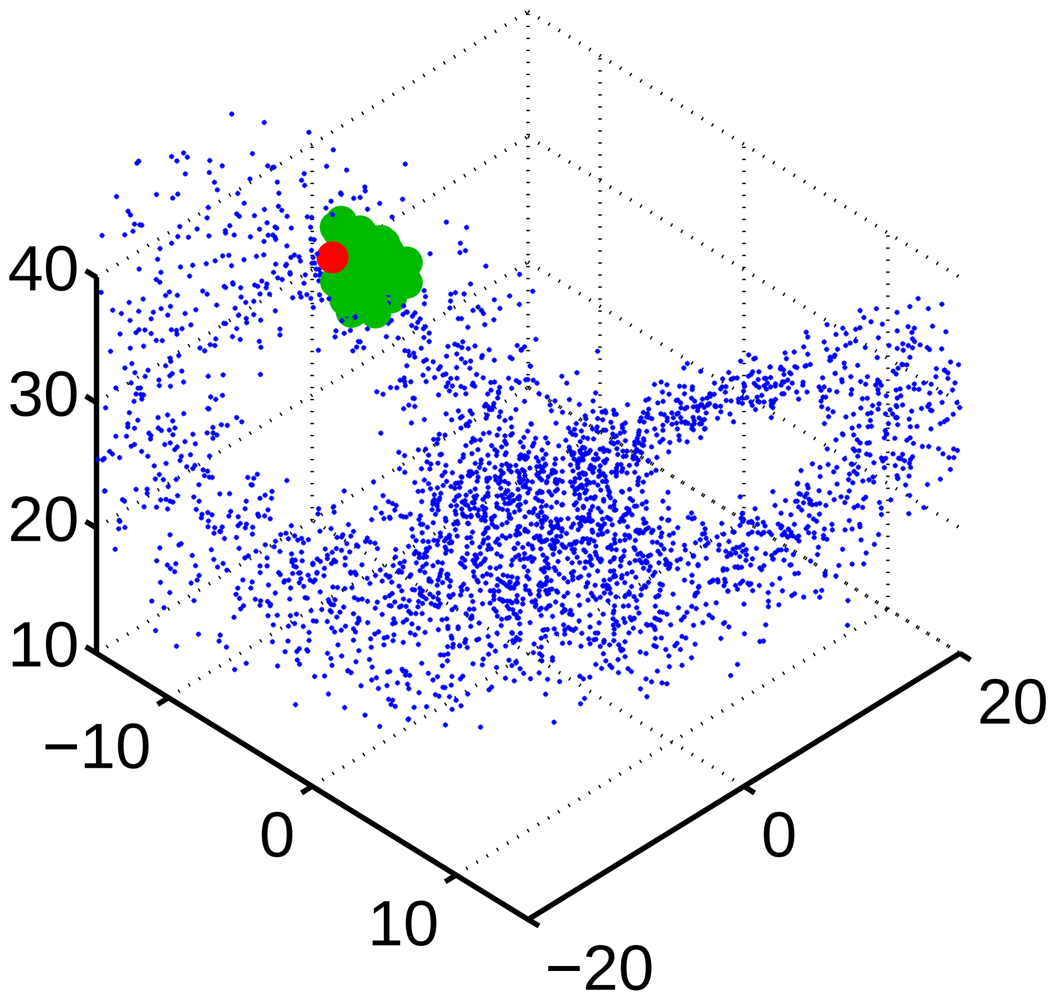


Observation in red.

Prior ensemble in green.

Phase 3: Generalize to geophysical models and observations

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

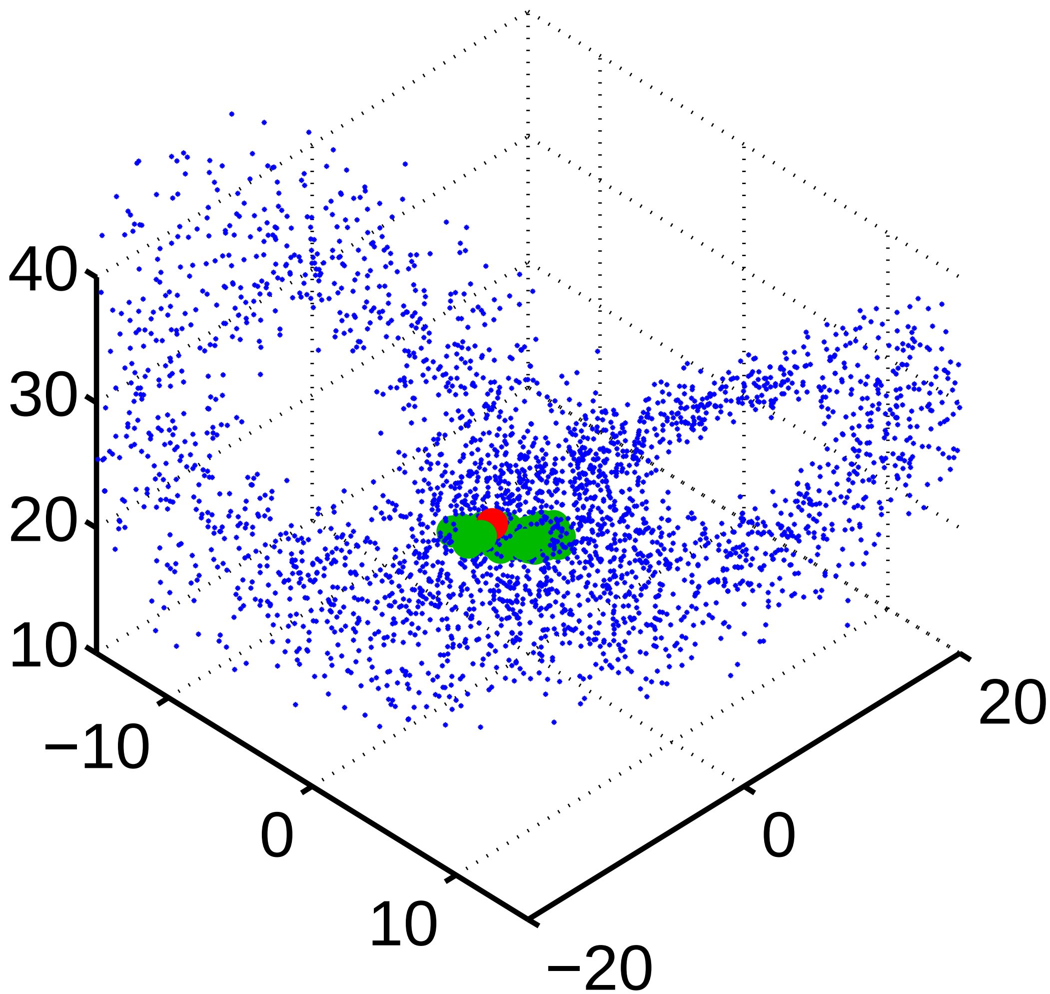


Observation in red.

Prior ensemble in green.

Phase 3: Generalize to geophysical models and observations

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



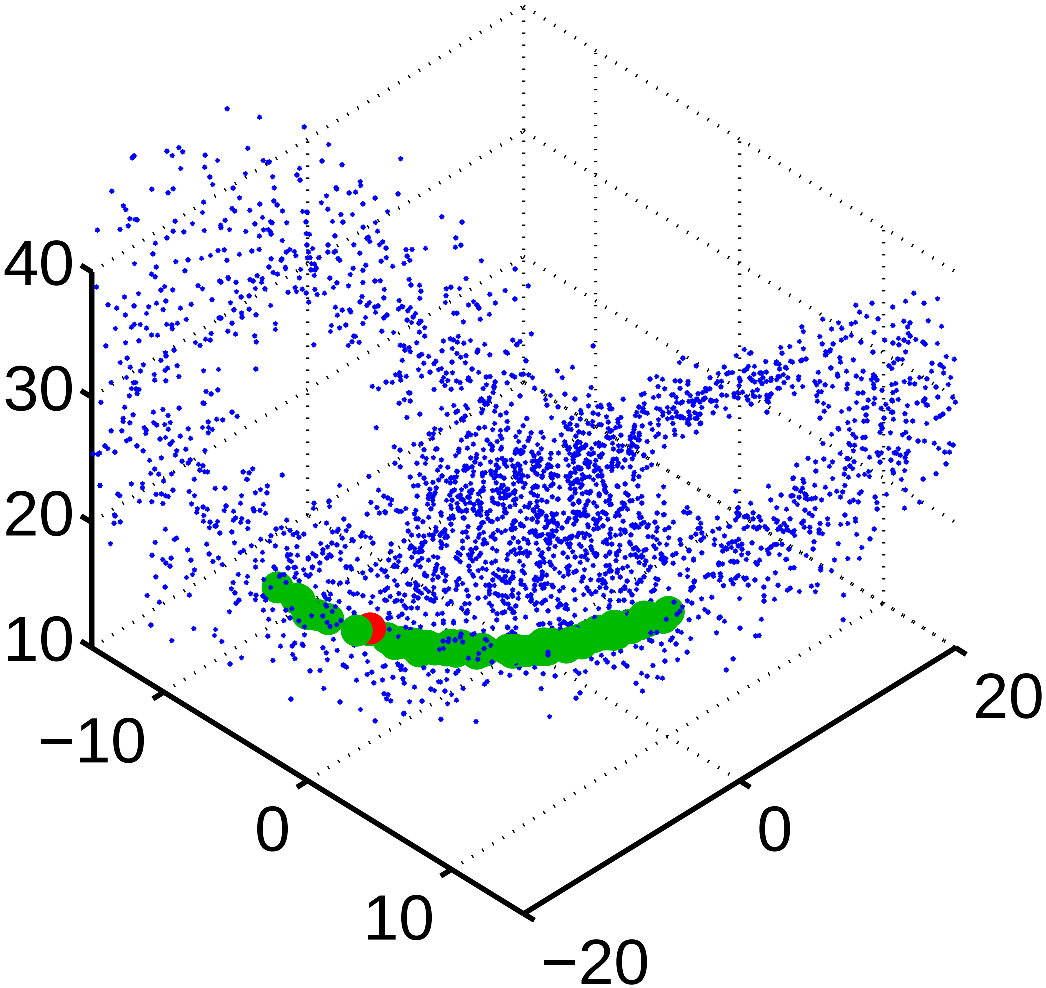
Observation in red.

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.



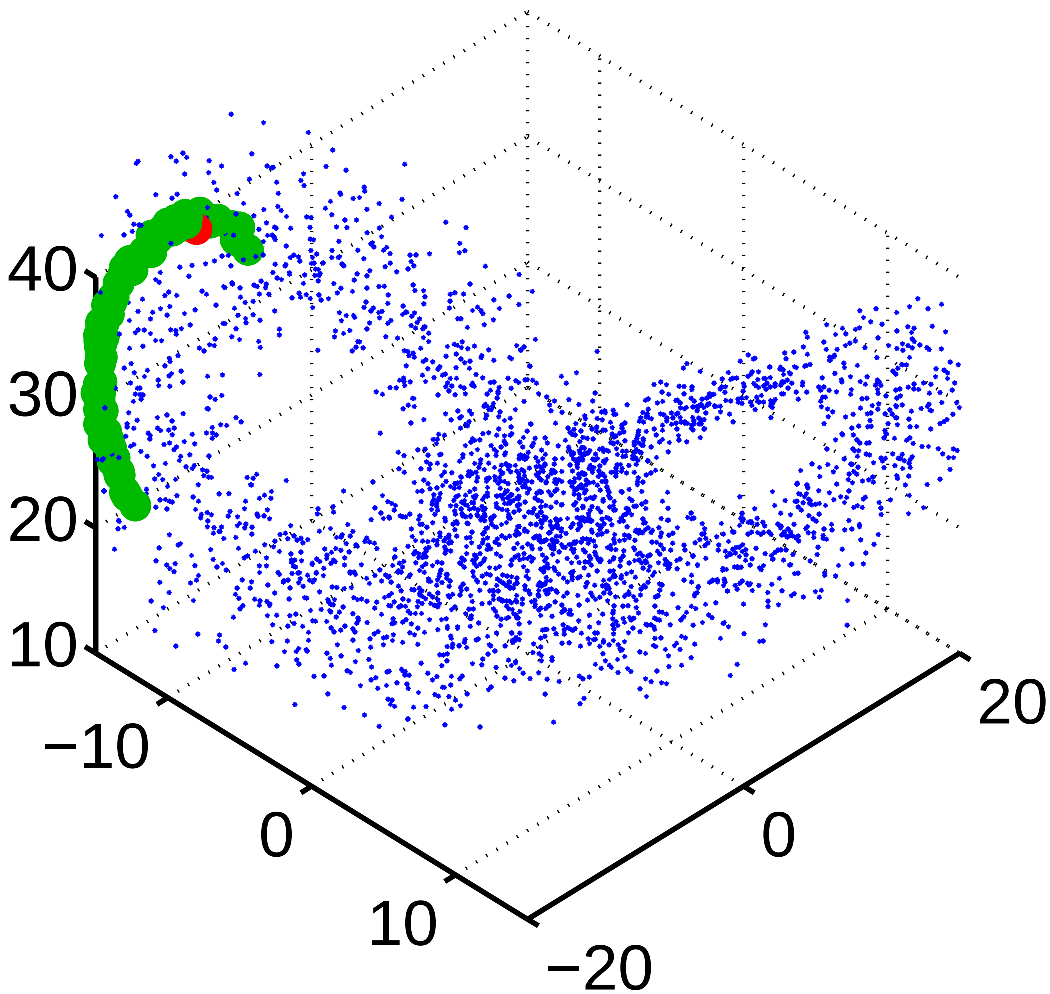
Observation in red.

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.

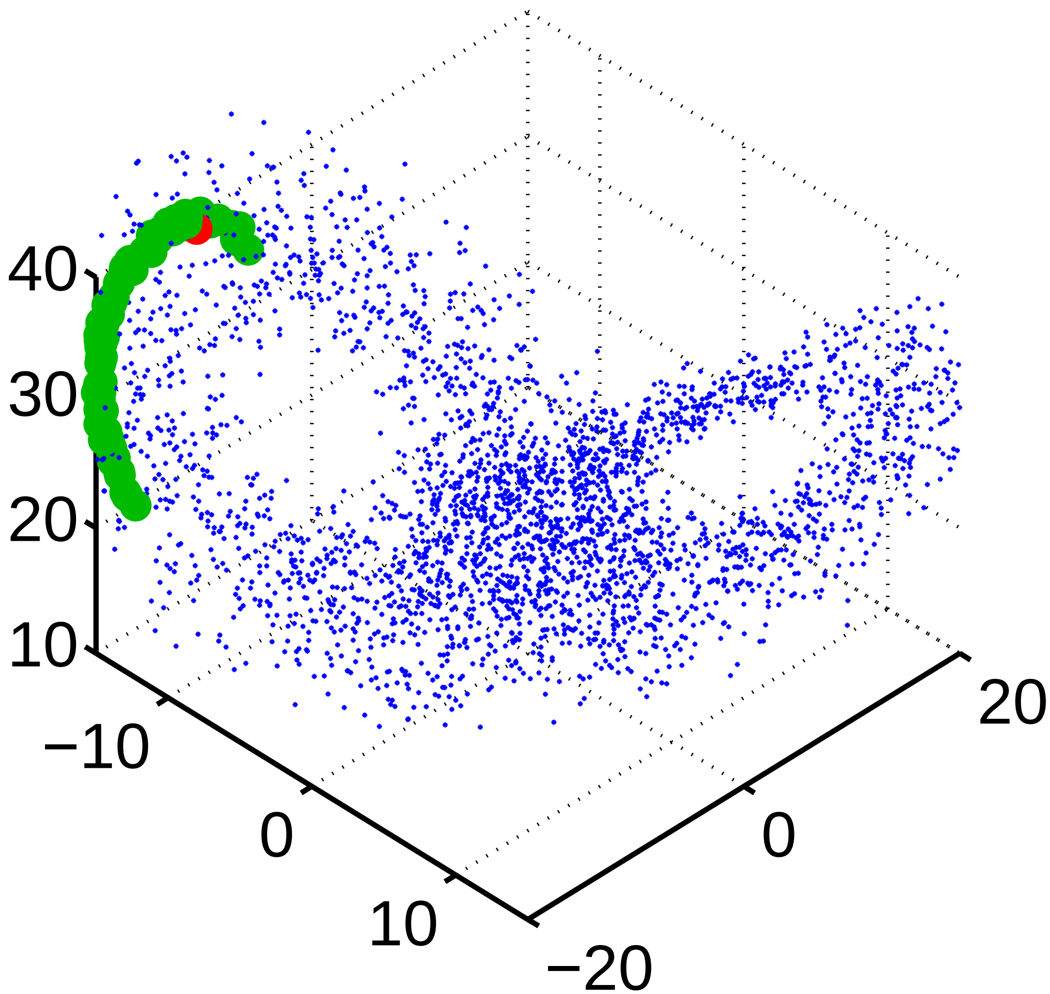
The prior is not linear here.

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.

Phase 4: Quick look at a real atmospheric application

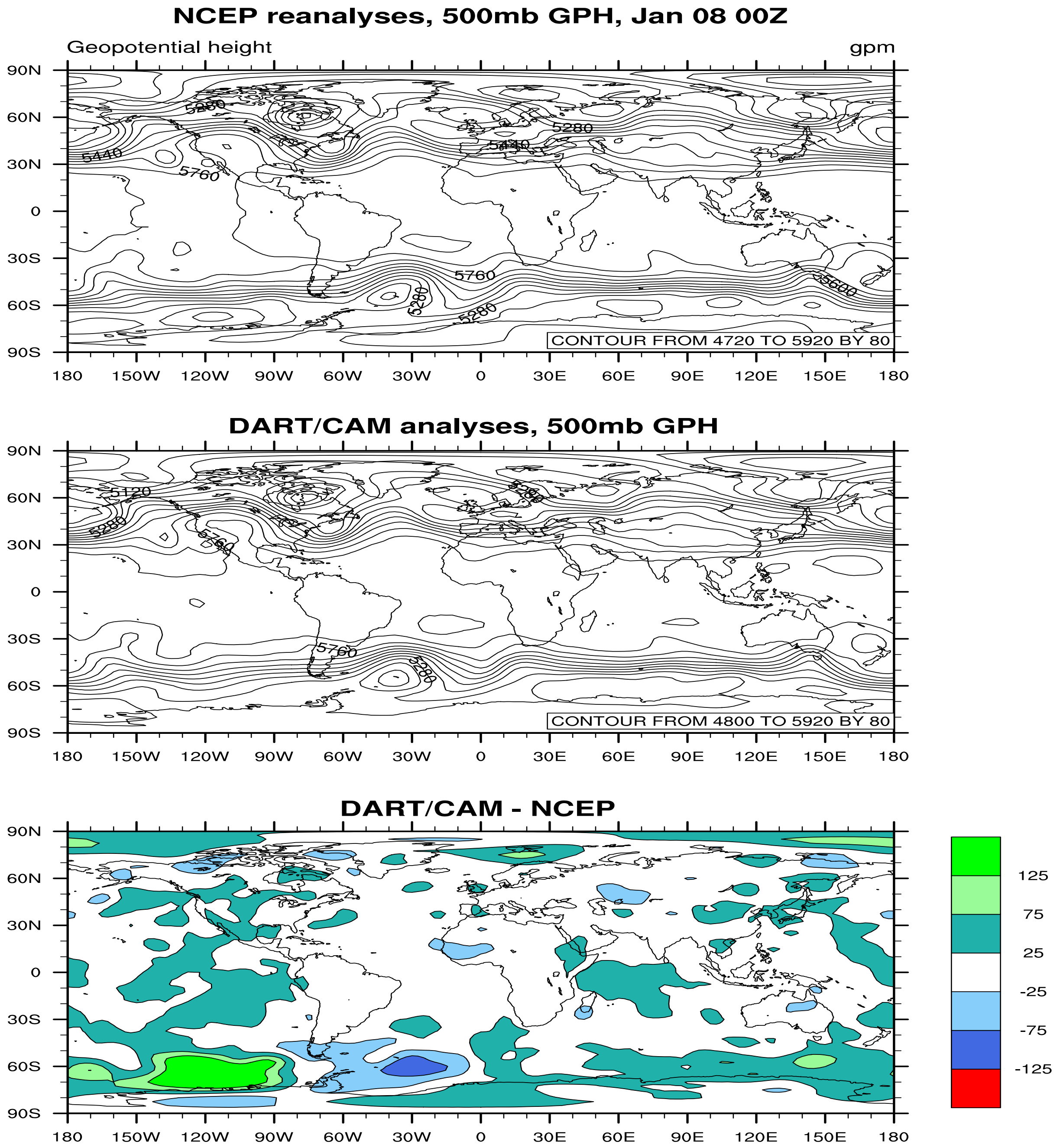
Results from CAM Assimilation: January, 2003

Model:

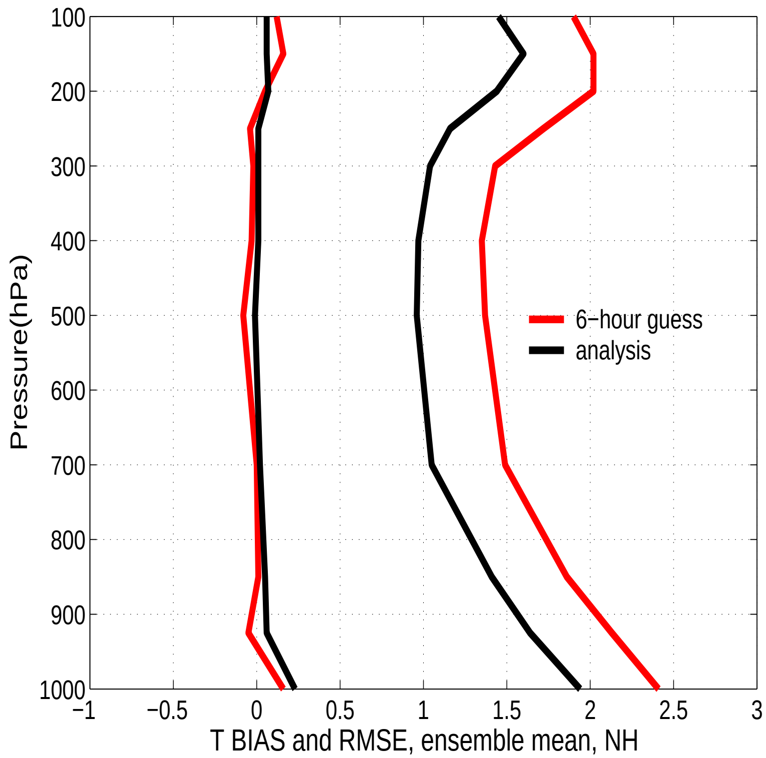
CAM 2.0 T42L26.
U,V, T, Q and PS state variables impacted by observations.
Land model (CLM 2.0) not impacted by observations.
Observed SSTs.

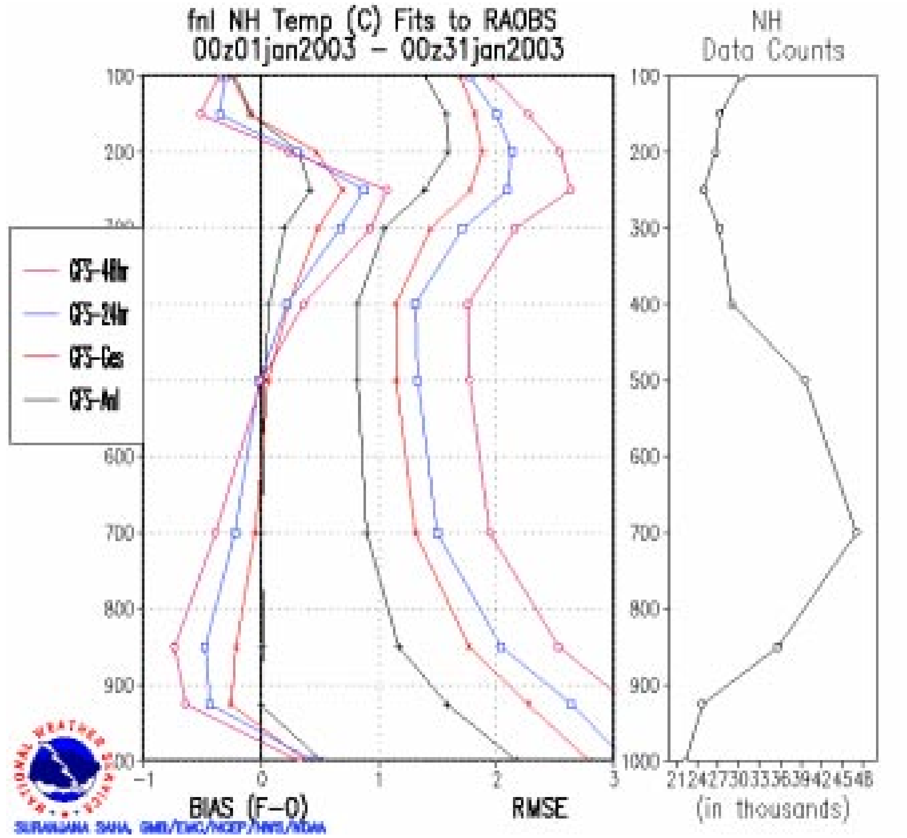
Assimilation / Prediction Experiments:

Uses observations used in reanalysis
(Radiosondes, ACARS, Satellite Winds..., no surface obs.).
Initial tests for January, 2003.
Assimilated every 6 hours; +/- 1.5 hour window for obs.
Run on CGD linux cluster Anchorage.


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
90N
60N
30N
0
30S
60S
90S
180
150W
120W
90W
60W
30W
0
30E
60E
90E
120E
150E
180

Northern Hemisphere Temperature: Bias and RMSE

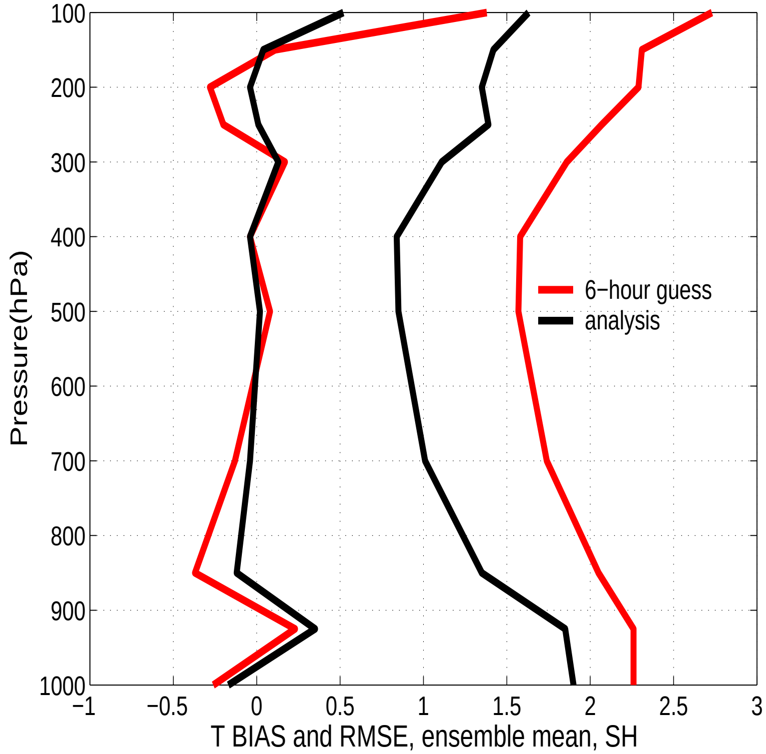


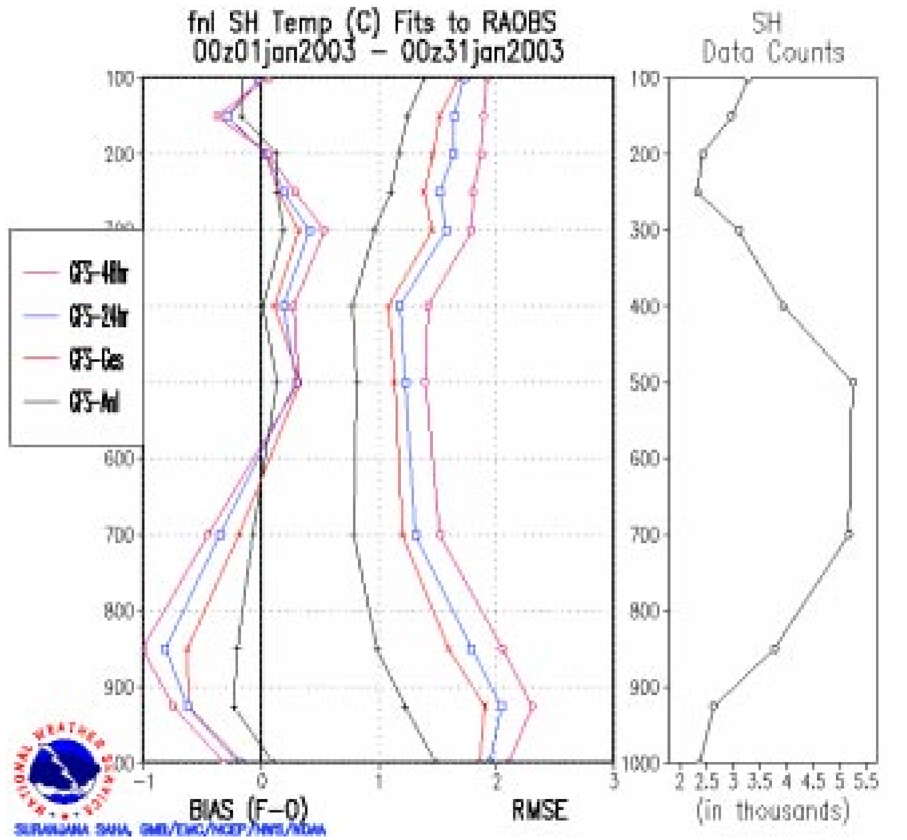


CAM

NCEP Reanalysis

Southern Hemisphere Temperature

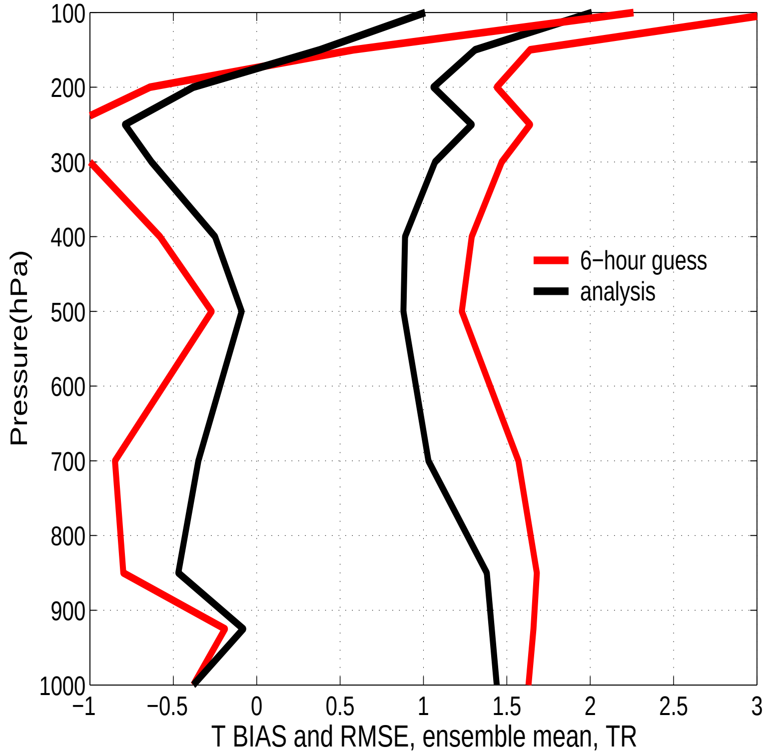


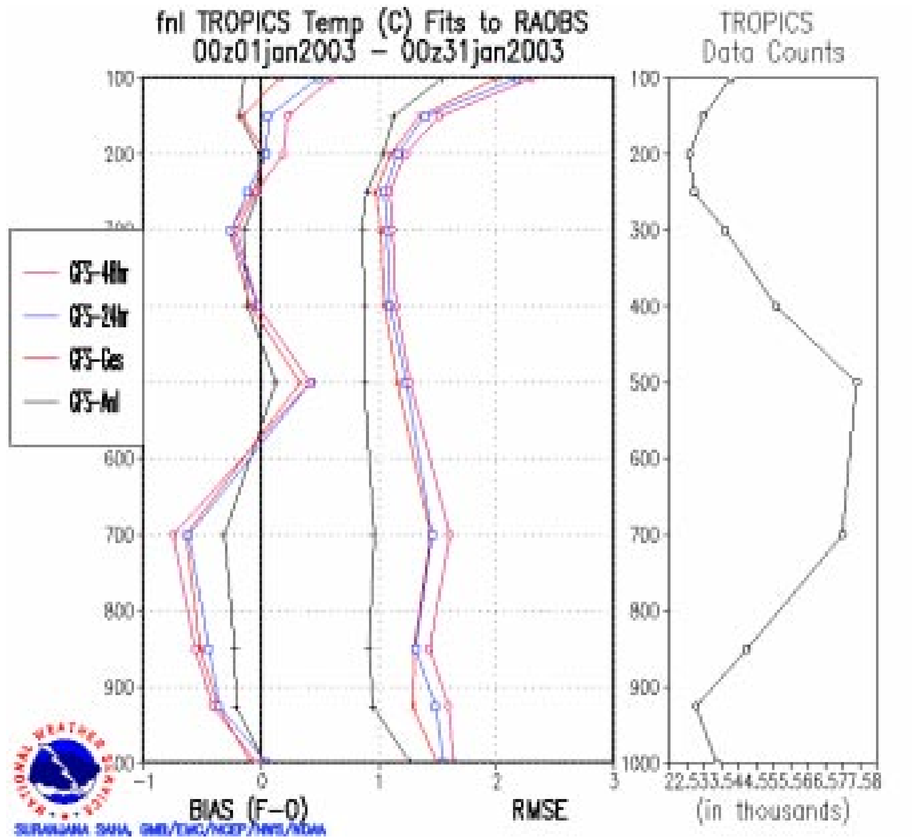


CAM

NCEP Reanalysis

Tropical Temperatures

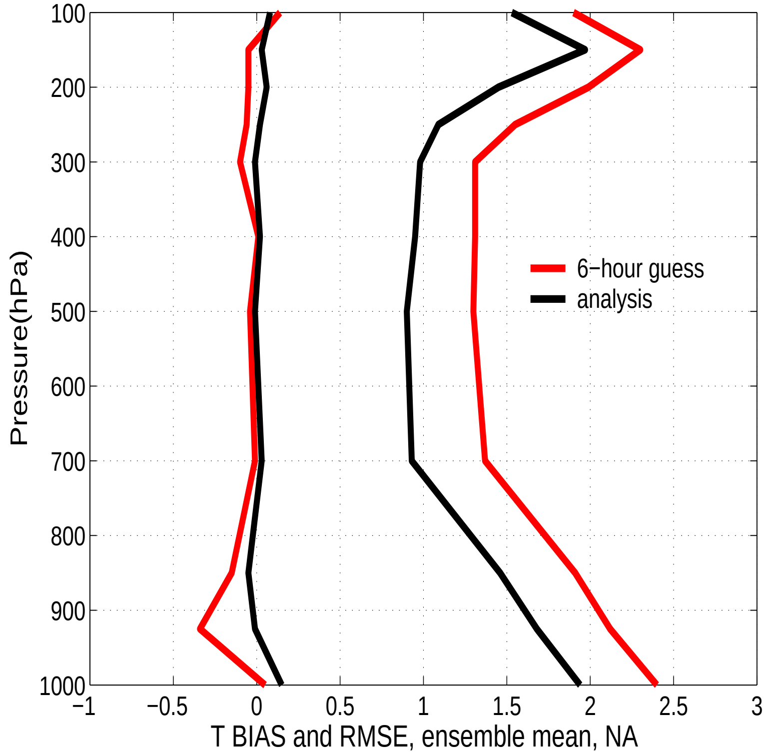


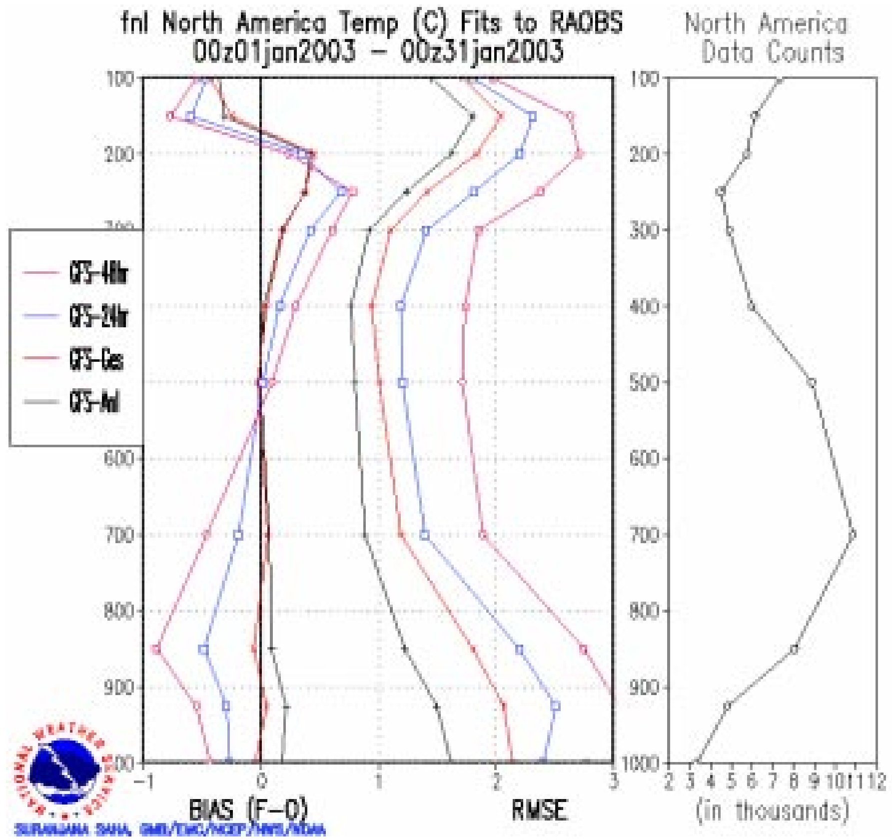


CAM

NCEP Reanalysis

North America Temperatures

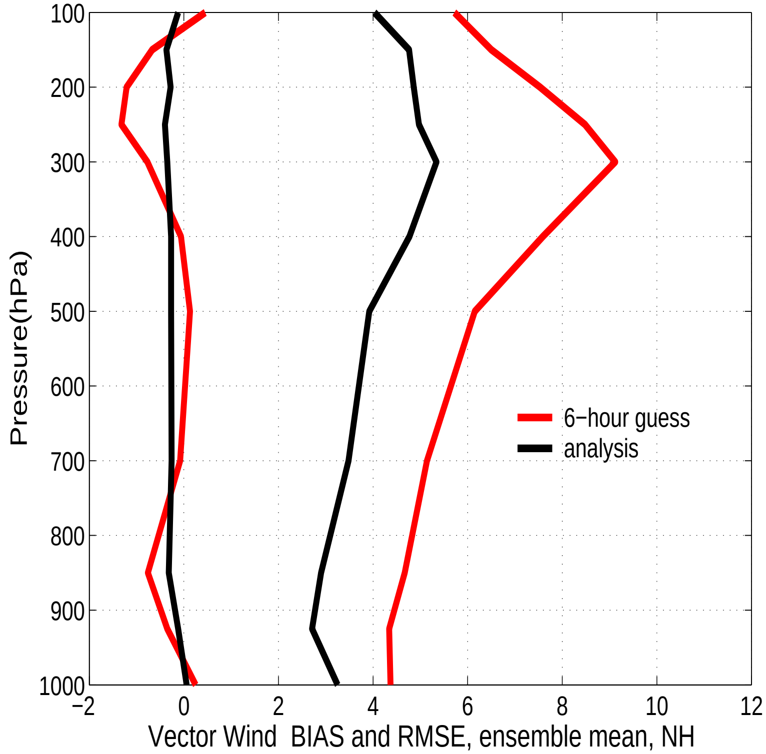


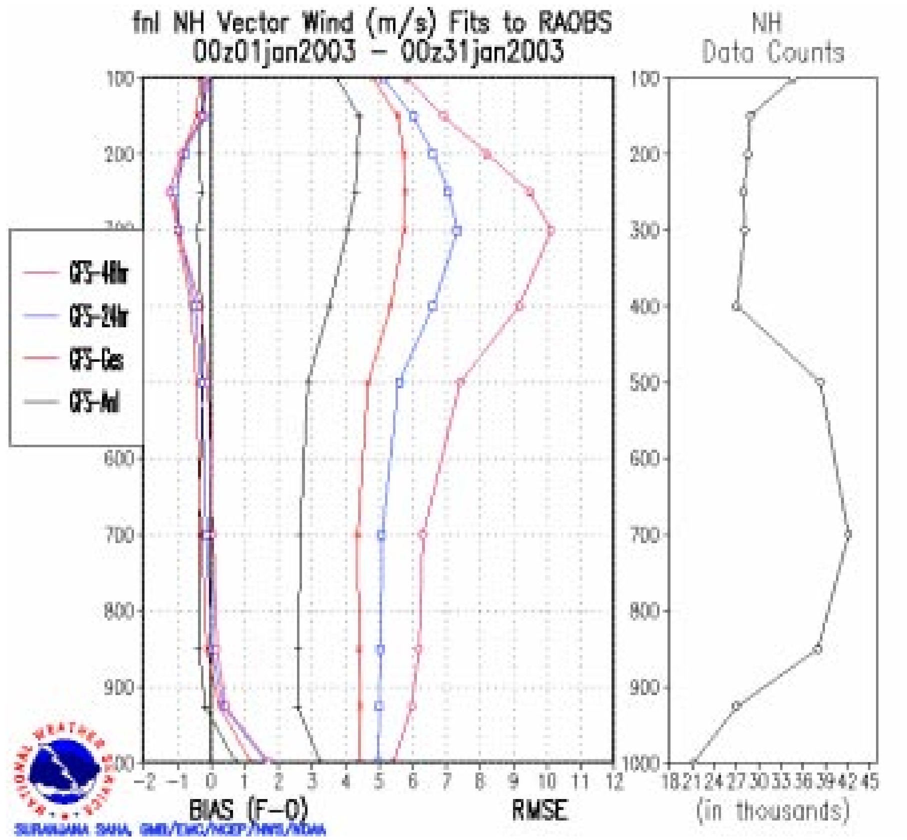


CAM

NCEP Reanalysis

Northern Hemisphere Winds

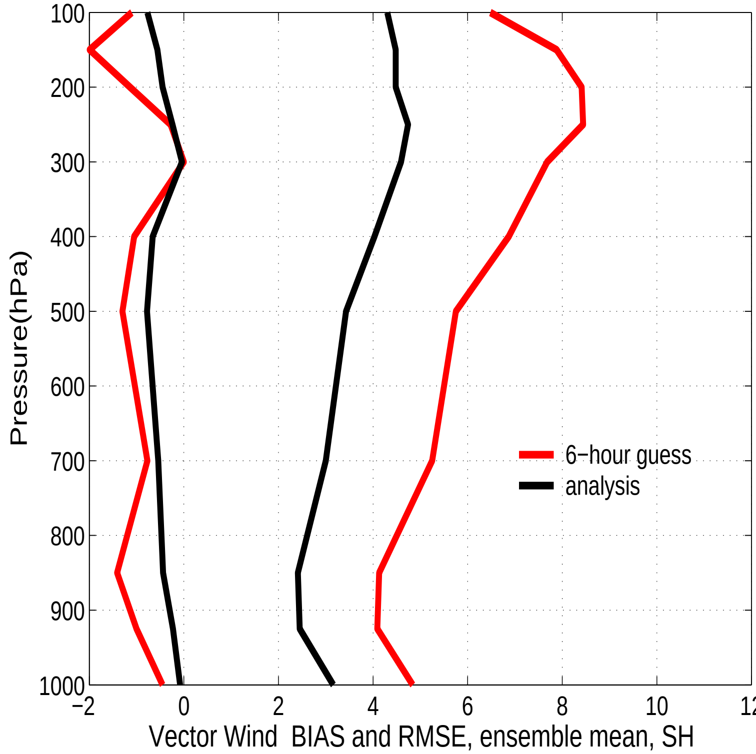


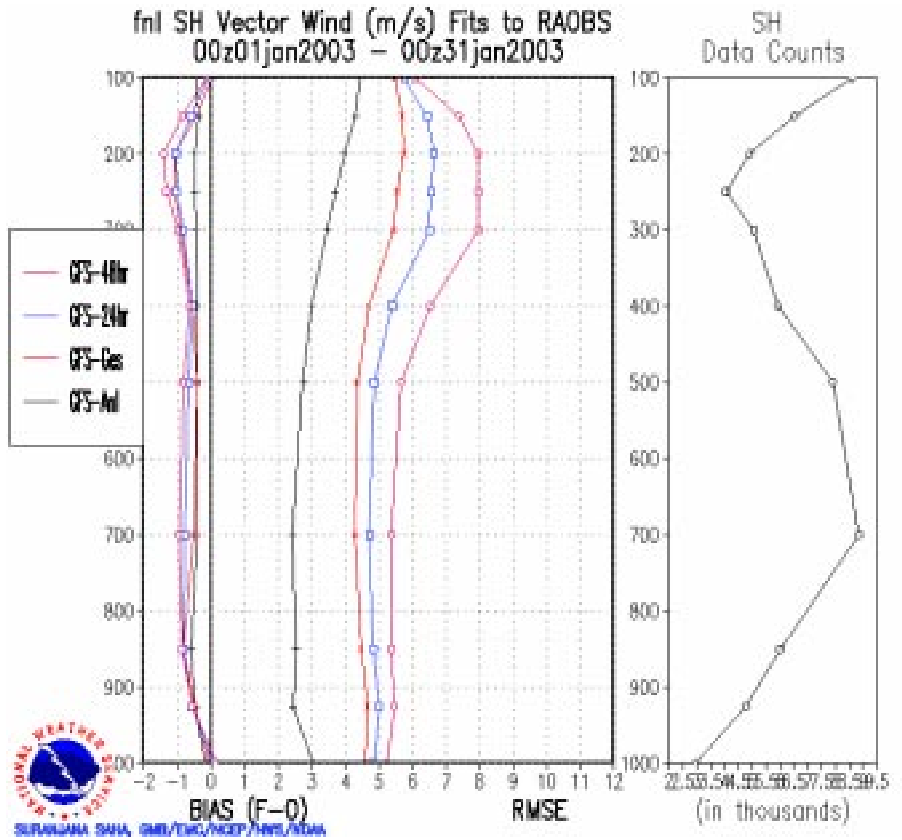


CAM

NCEP Reanalysis

Southern Hemisphere Winds

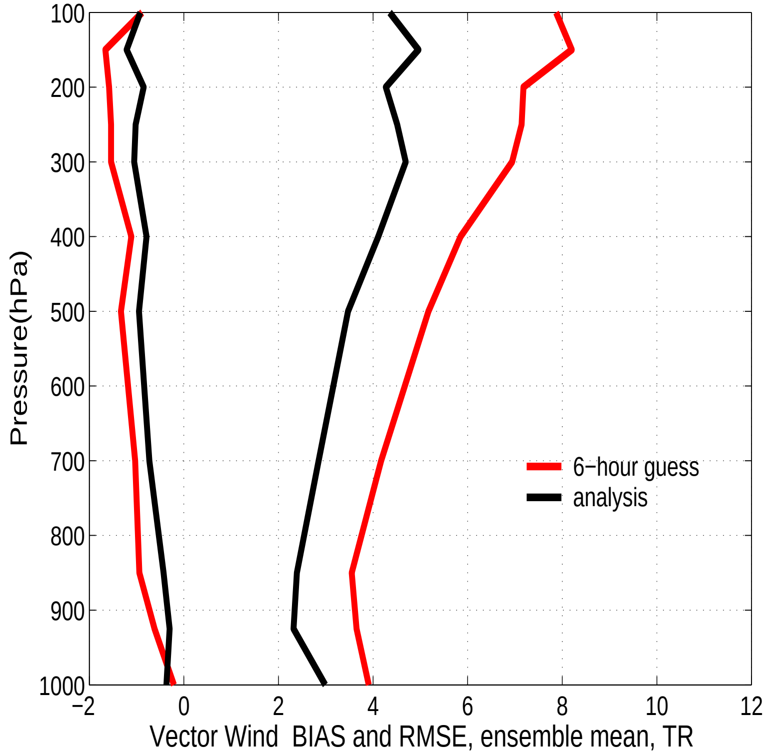


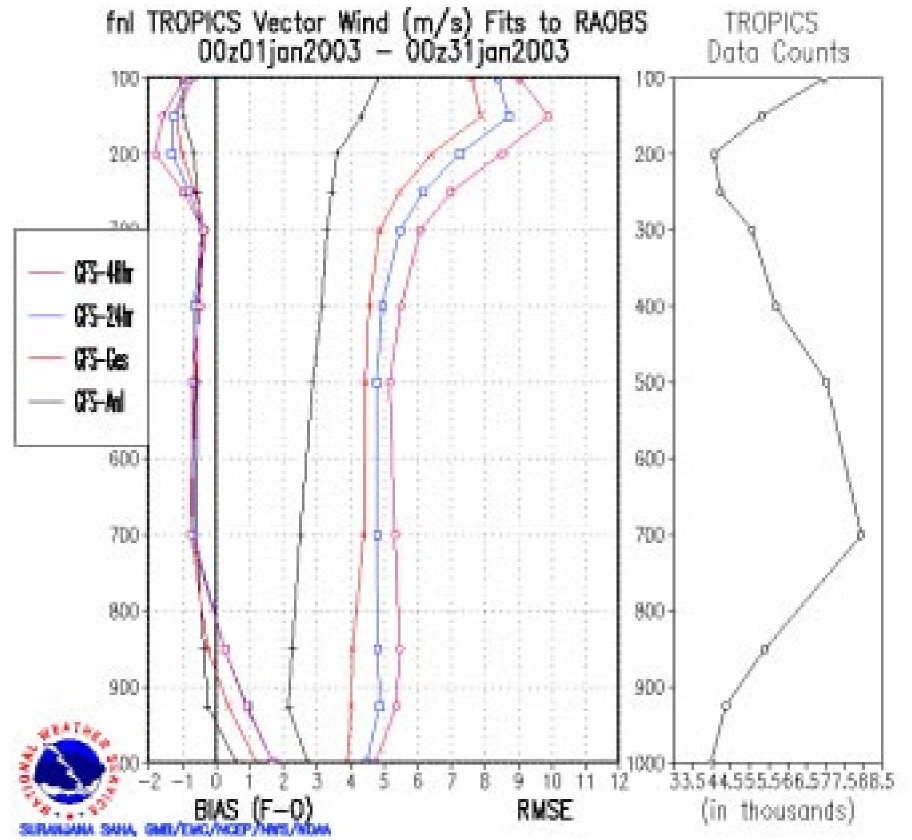


CAM

NCEP Reanalysis

Tropical Winds

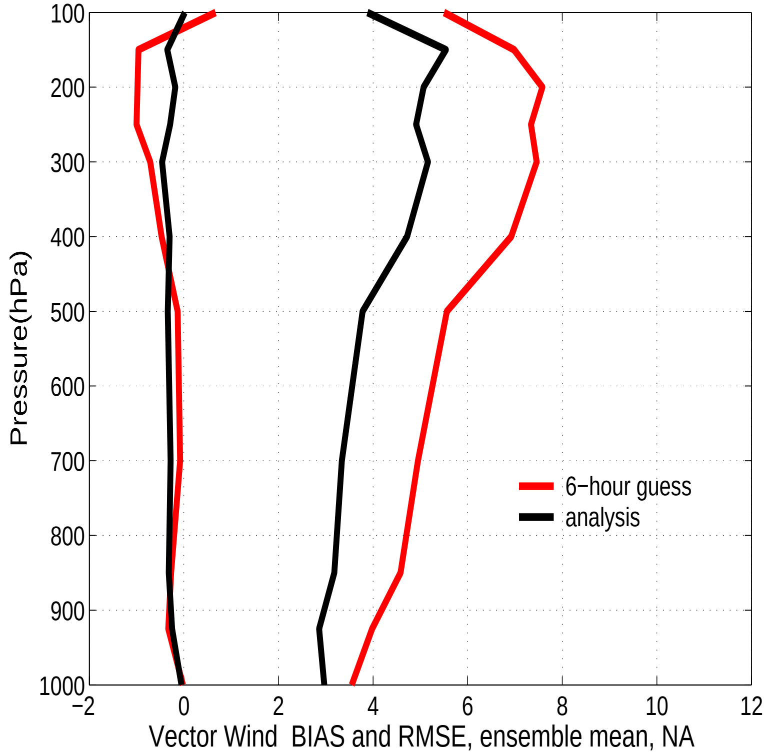


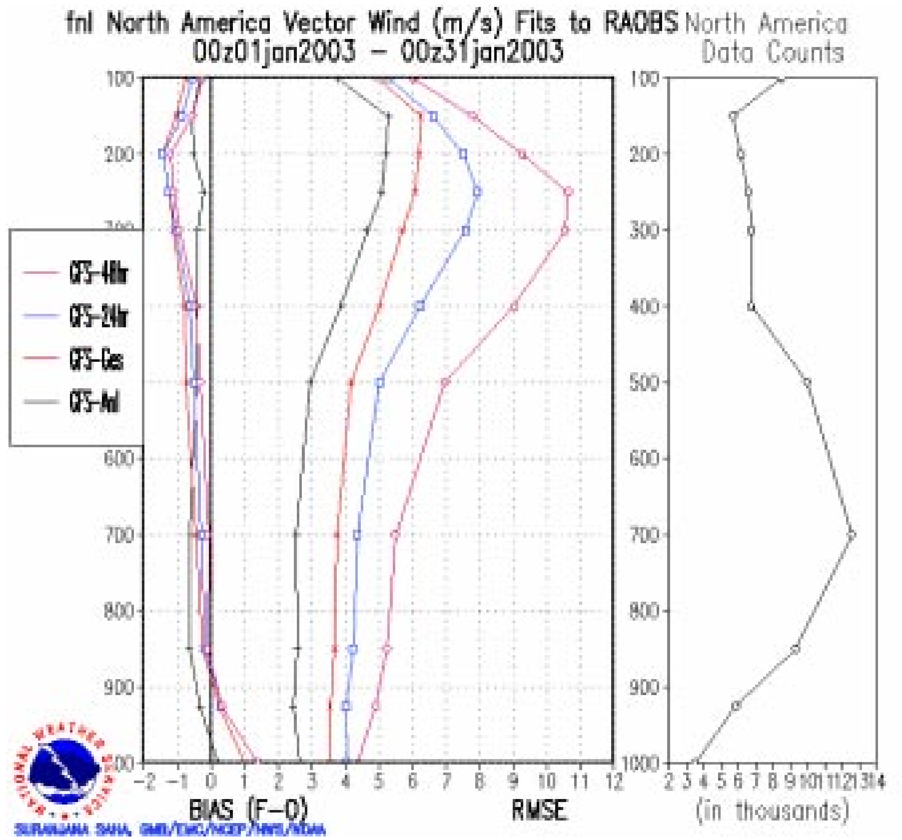


CAM

NCEP Reanalysis

North America Winds

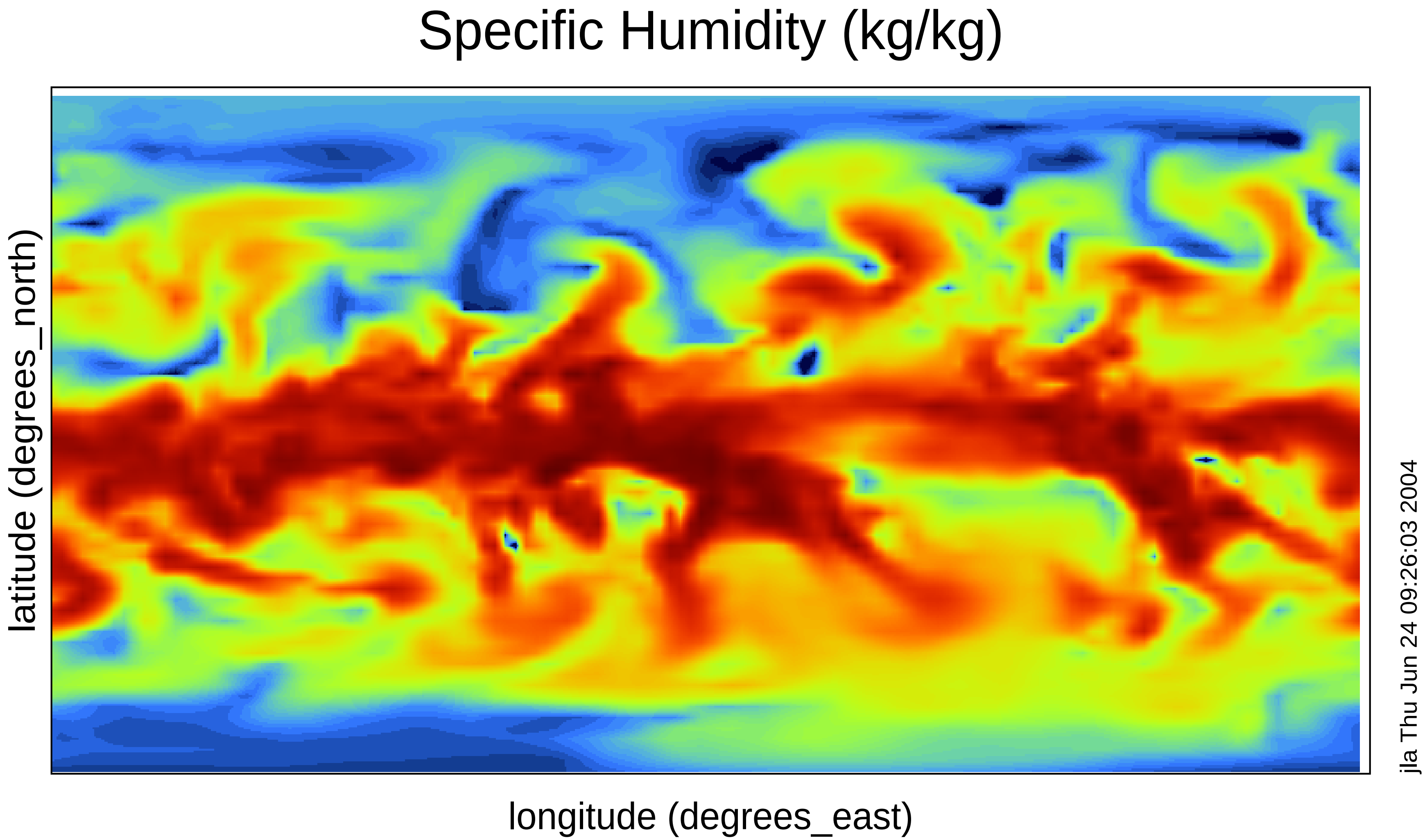




CAM

NCEP Reanalysis

12 GMT 4 January, 2003, CAM Analysis



Ensemble filters: What's next?

1. Adaptive error correction.
2. Parameter estimation for models.
3. Better understanding of error characteristics.
4. Understanding ensemble size requirements for given problem.
5. Dealing with complicated forward observation operators.
6. Ensemble smoothers (using data from the future).
7. Many, many, many exotic applications.

Data Assimilation Research Testbed (DART)

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

Requires F90 compiler, Matlab.

Available from www.cgd.ucar.edu/DAI/.