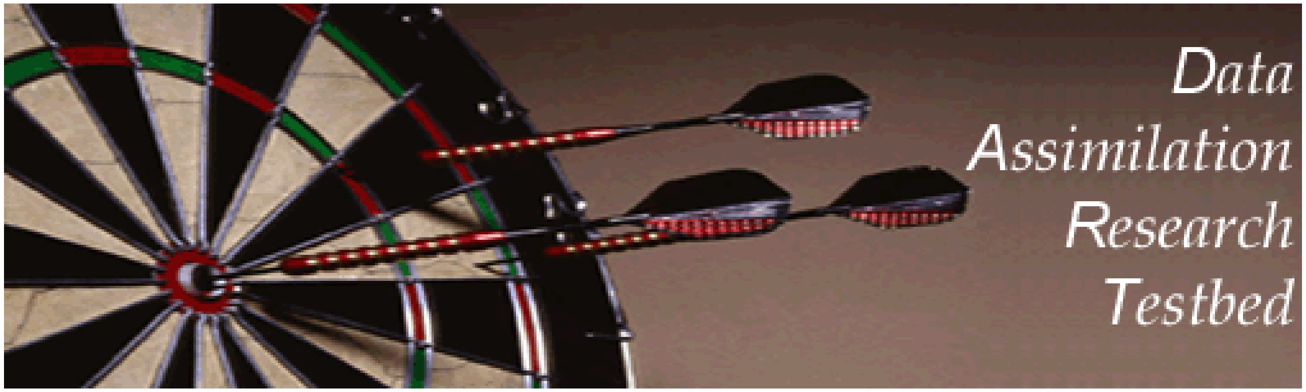
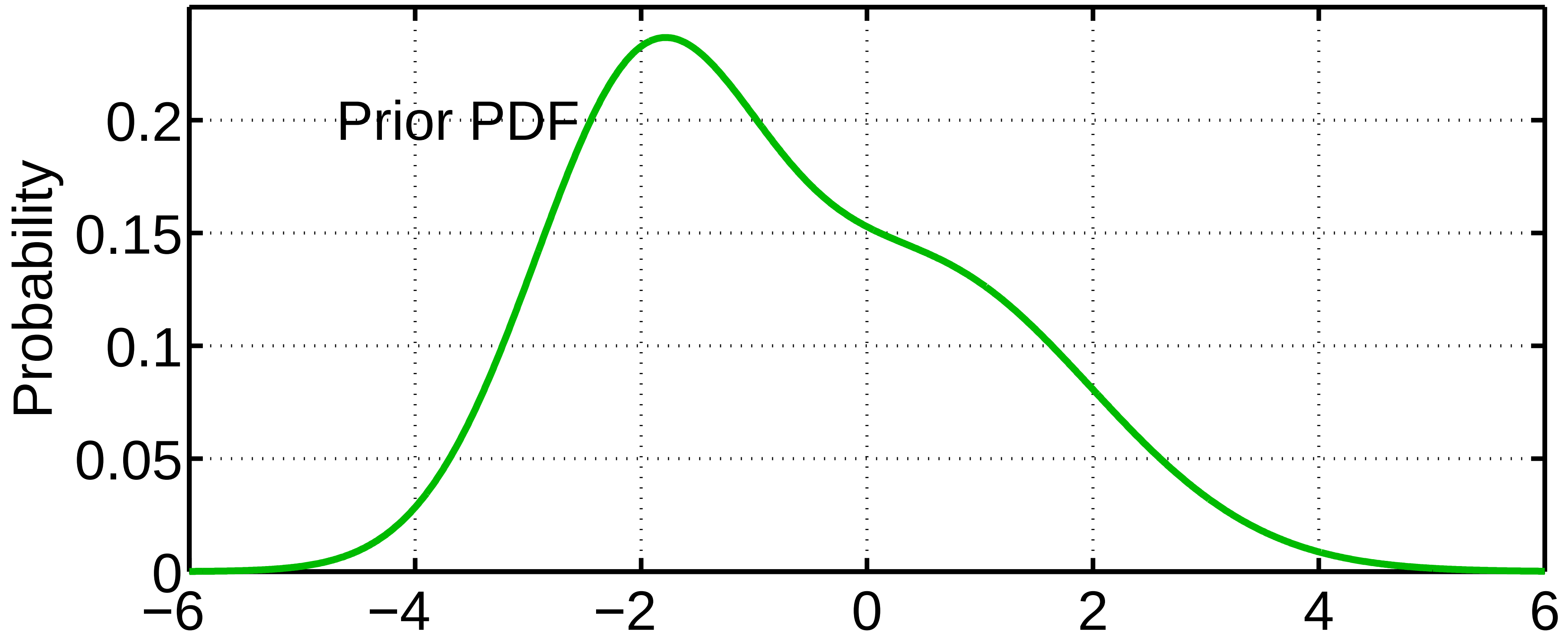
Data Assimilation Research Testbed Tutorial



Section 1: Filtering For a One Variable System

Version 1.0: June, 2005

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

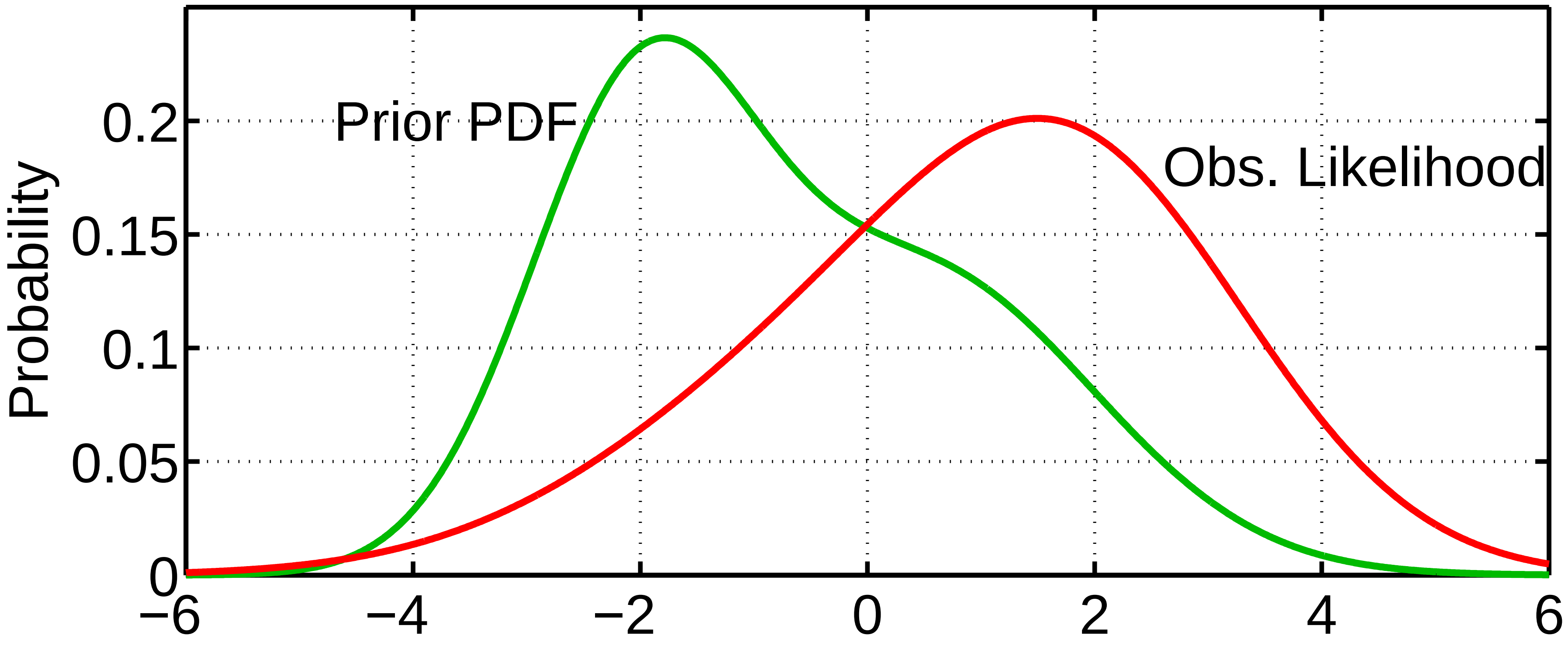


A: Prior estimate based on all previous information, C.

B: An additional observation.

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

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

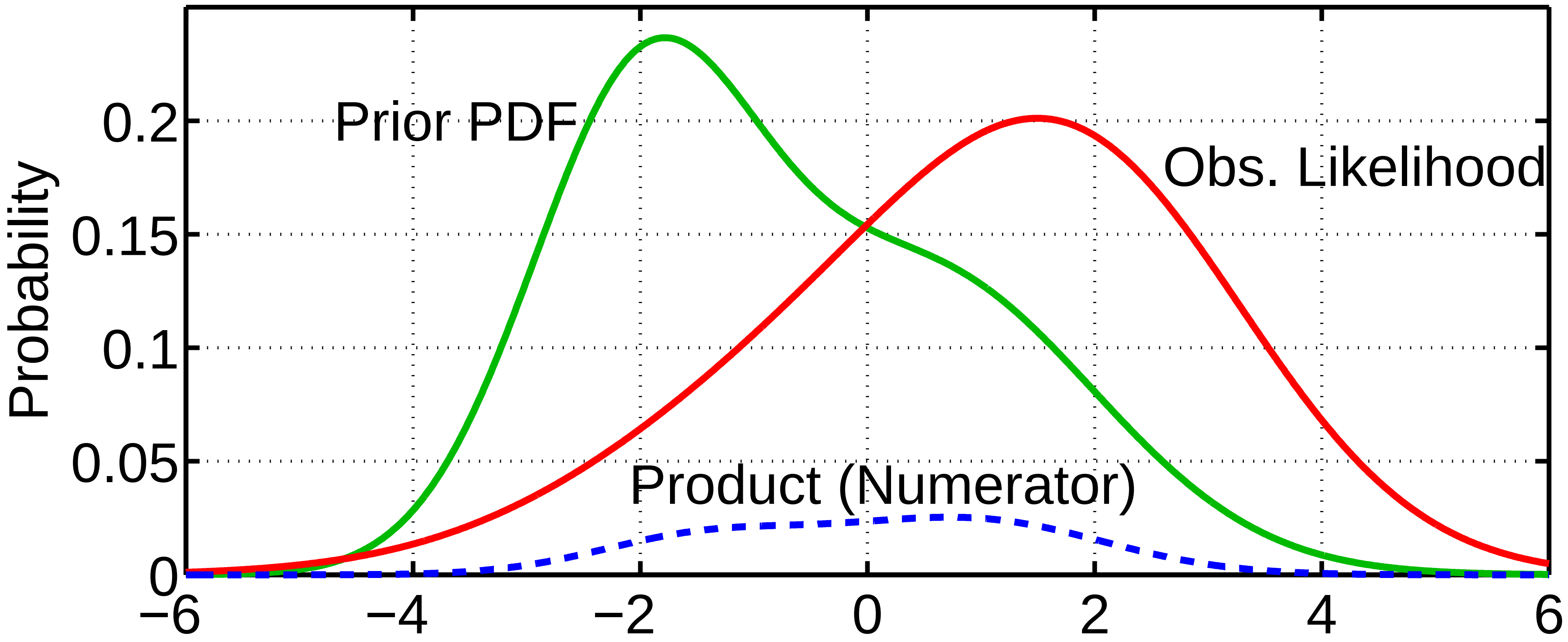


A: Prior estimate based on all previous information, C.

B: An additional observation.

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

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

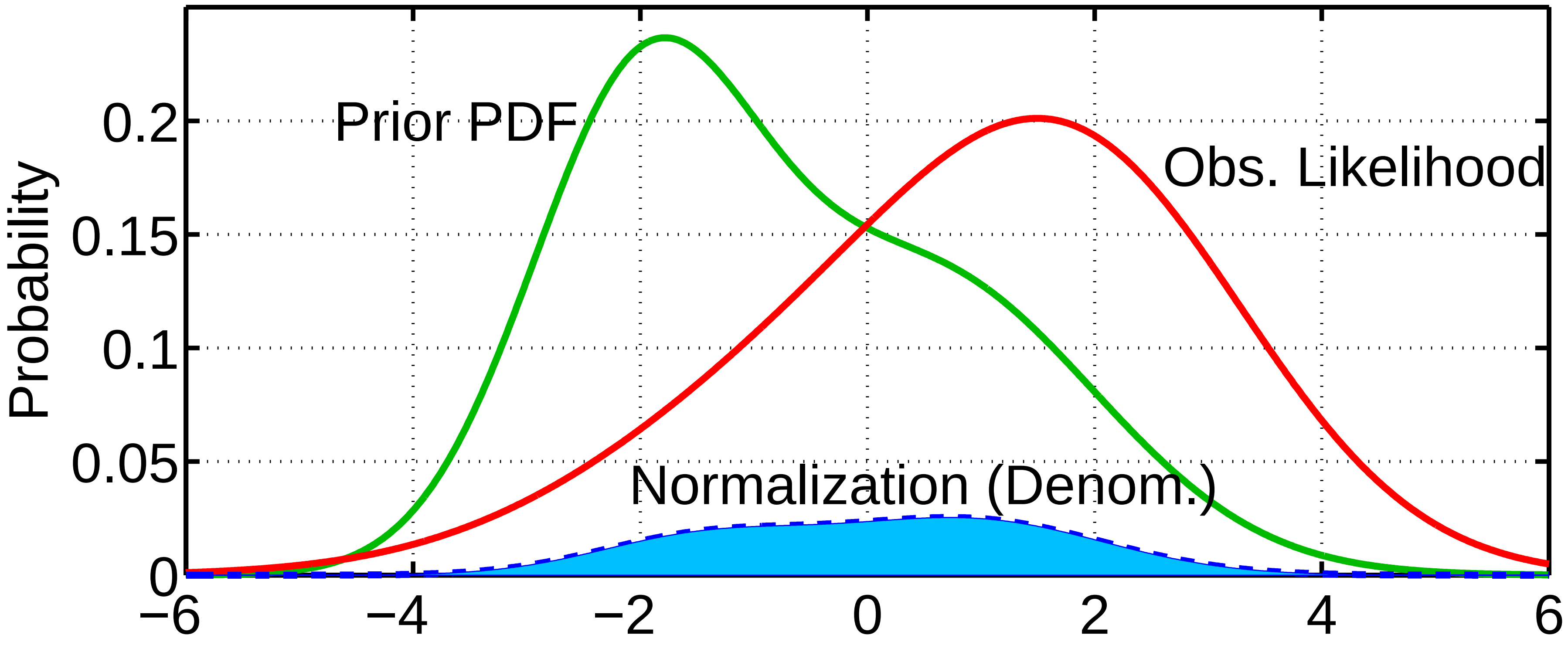


A: Prior estimate based on all previous information, C.

B: An additional observation.

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

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

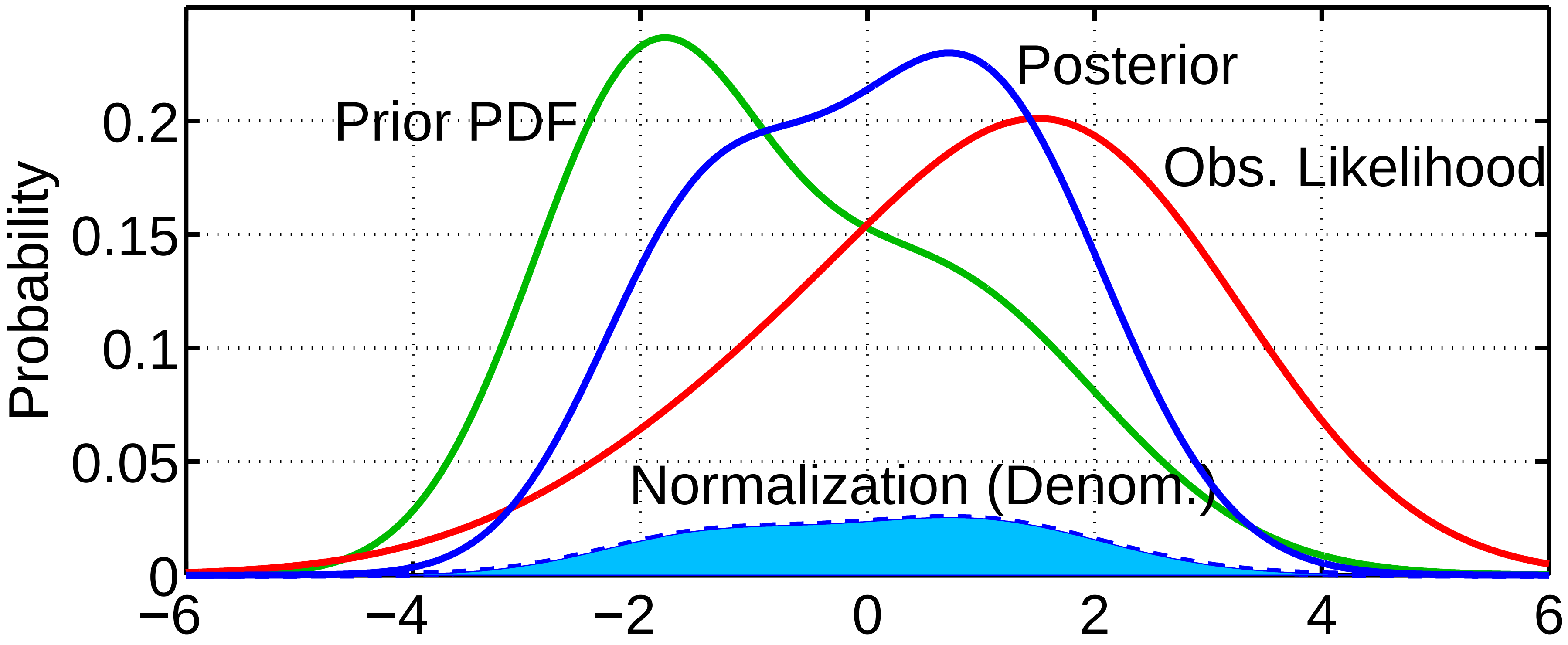


A: Prior estimate based on all previous information, C.

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.

Consistent Color Scheme Throughout Tutorial

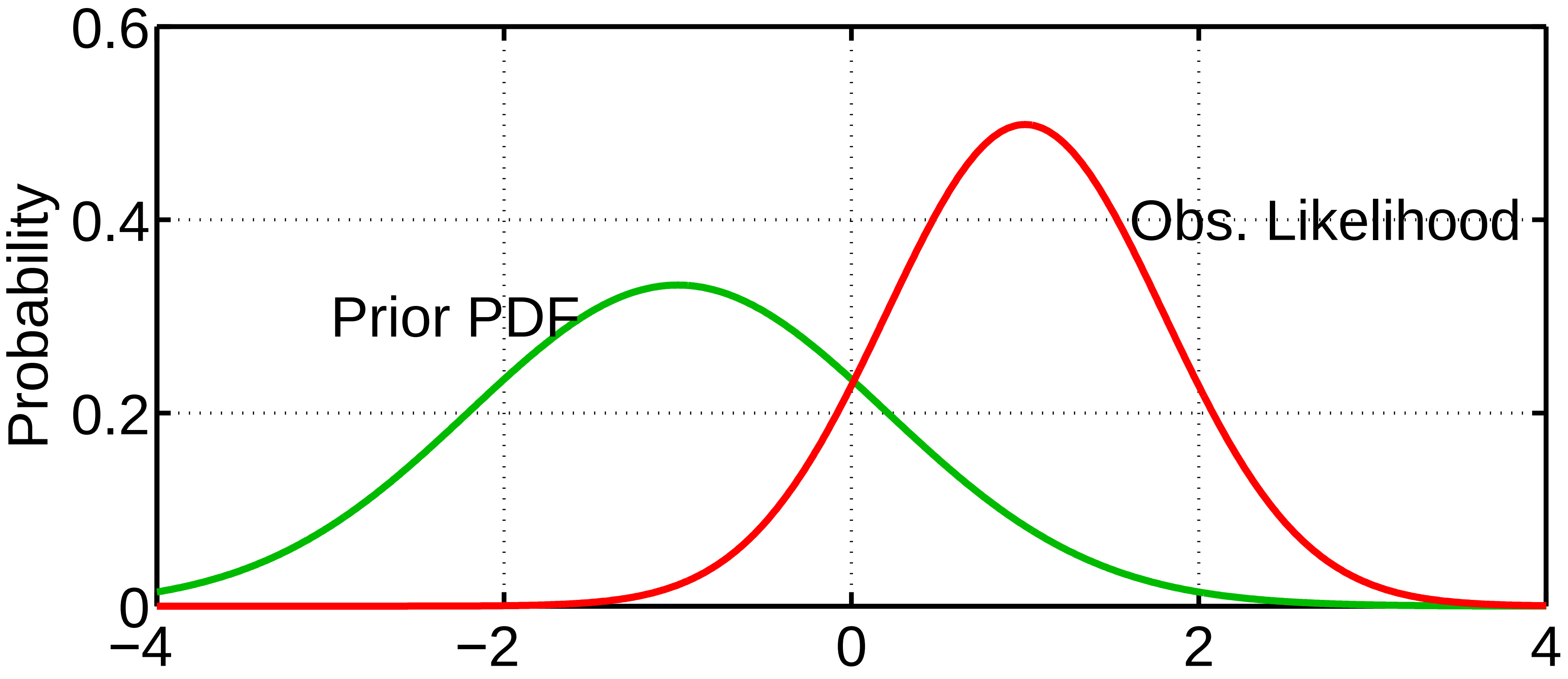
Green = Prior

Red = Observation

Blue = Posterior

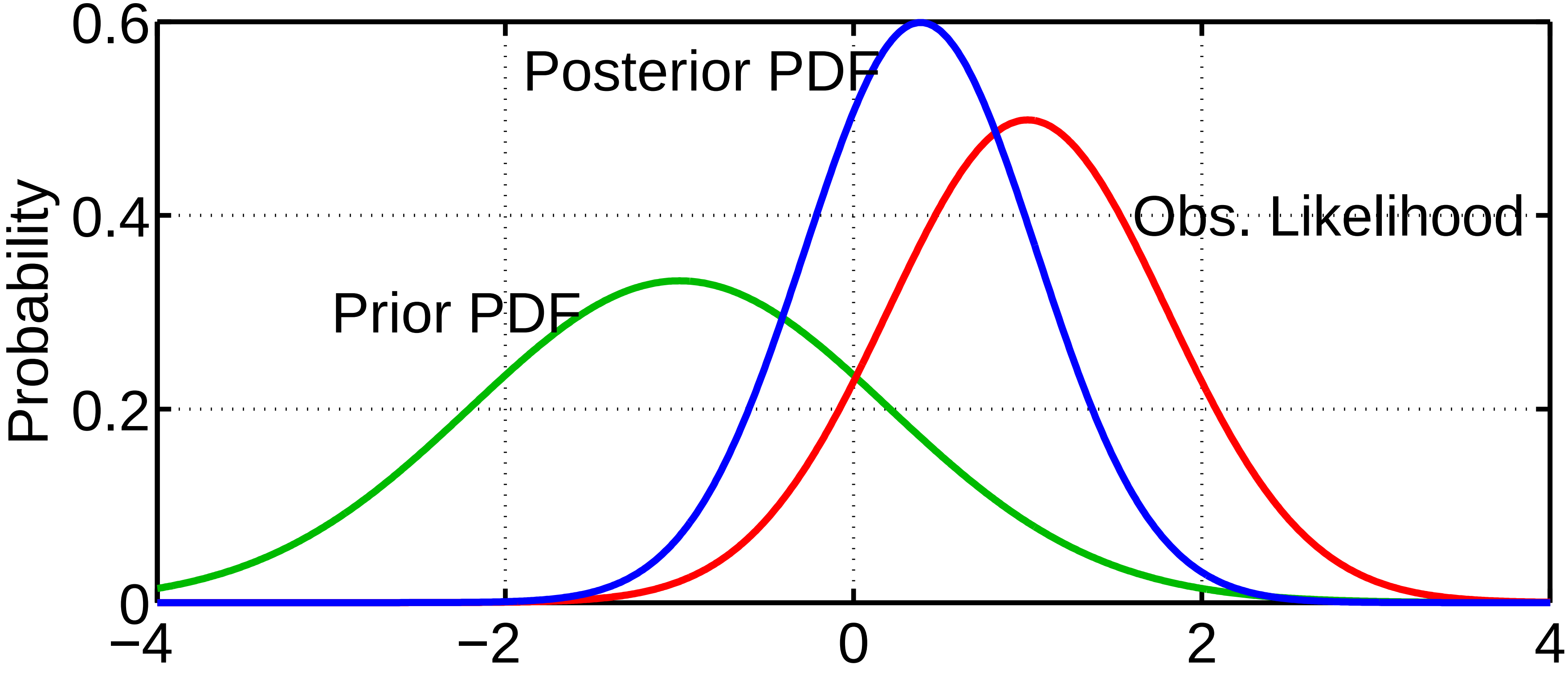
Bayes rule: $p(A|BC) = \frac{p(B|AC)p(A|C)}{p(B|C)} = \frac{p(B|AC)p(A|C)}{\int p(B|x)p(x|C)dx}$

This product is closed for Gaussian distributions.



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.

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

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.

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

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

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

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.

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

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

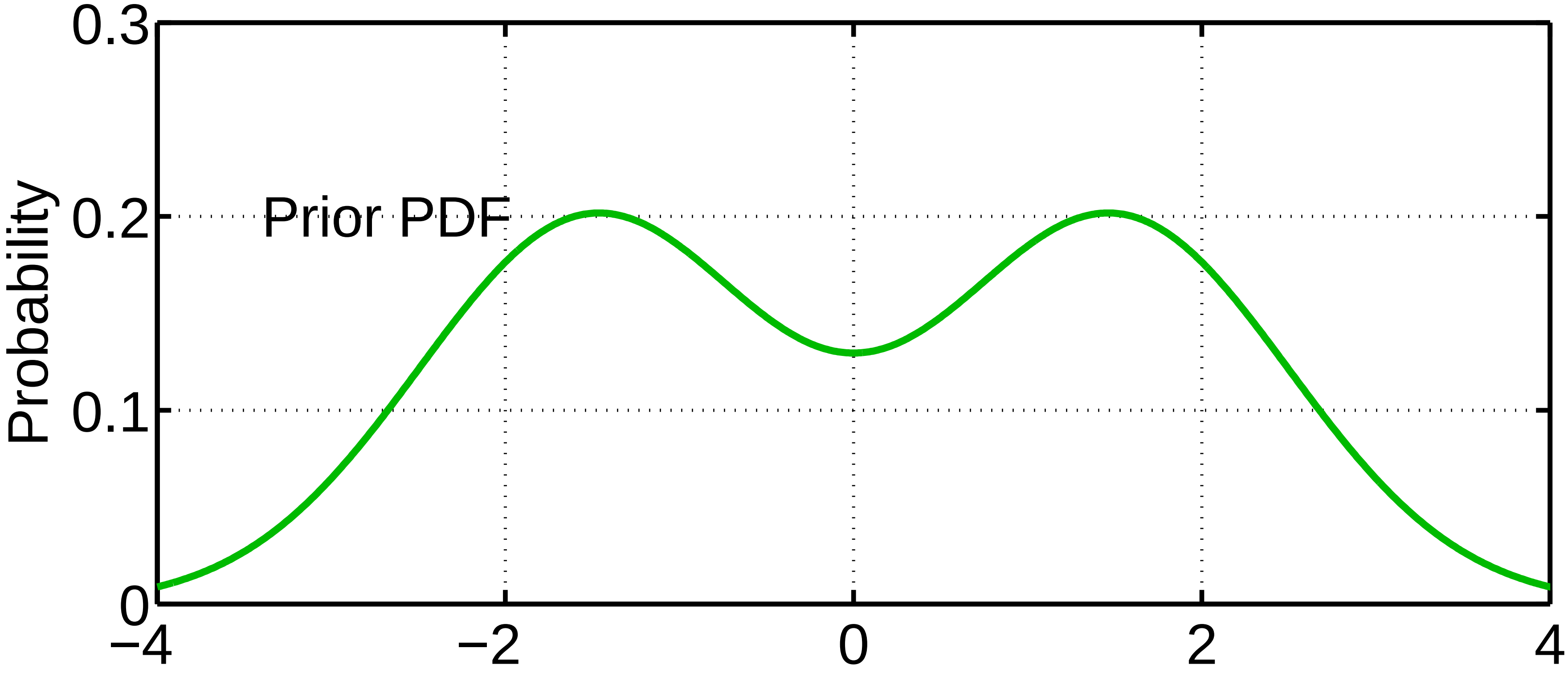
Mean: $\mu = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}(\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2)$

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

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

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

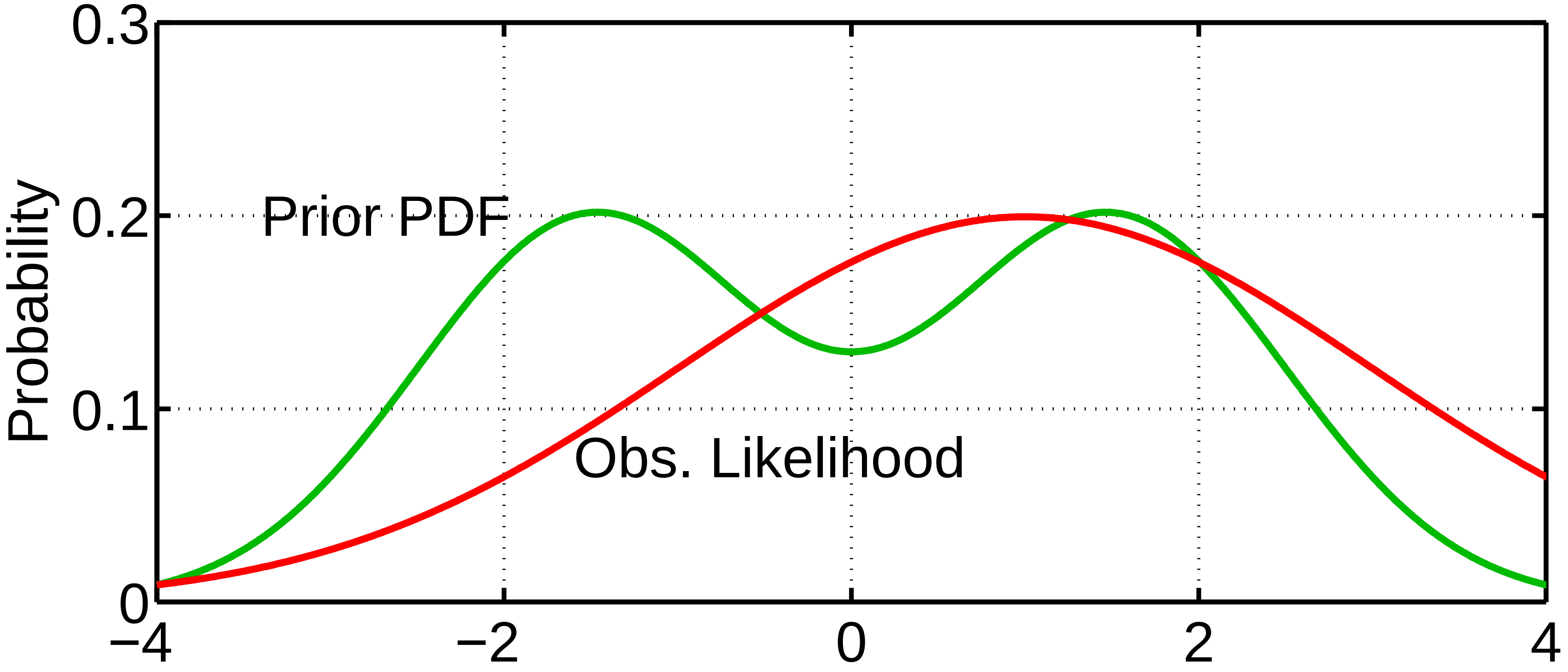
This product is closed for Gaussian distributions.



There are other families of functions for which it is closed...
But, for general distributions, there’s no analytical product.

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

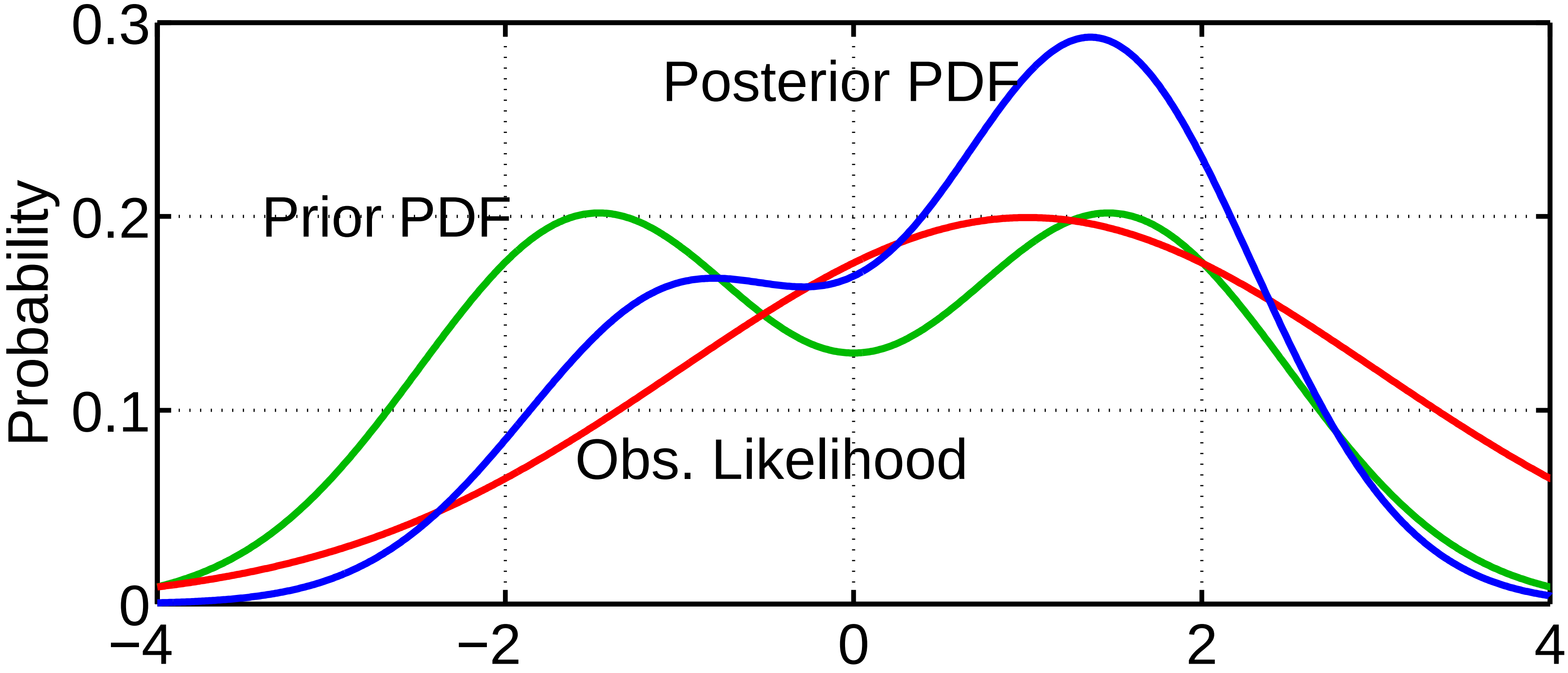
This product is closed for Gaussian distributions.



There are other families of functions for which it is closed...
But, for general distributions, there’s no analytical product.

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

This product is closed for Gaussian distributions.

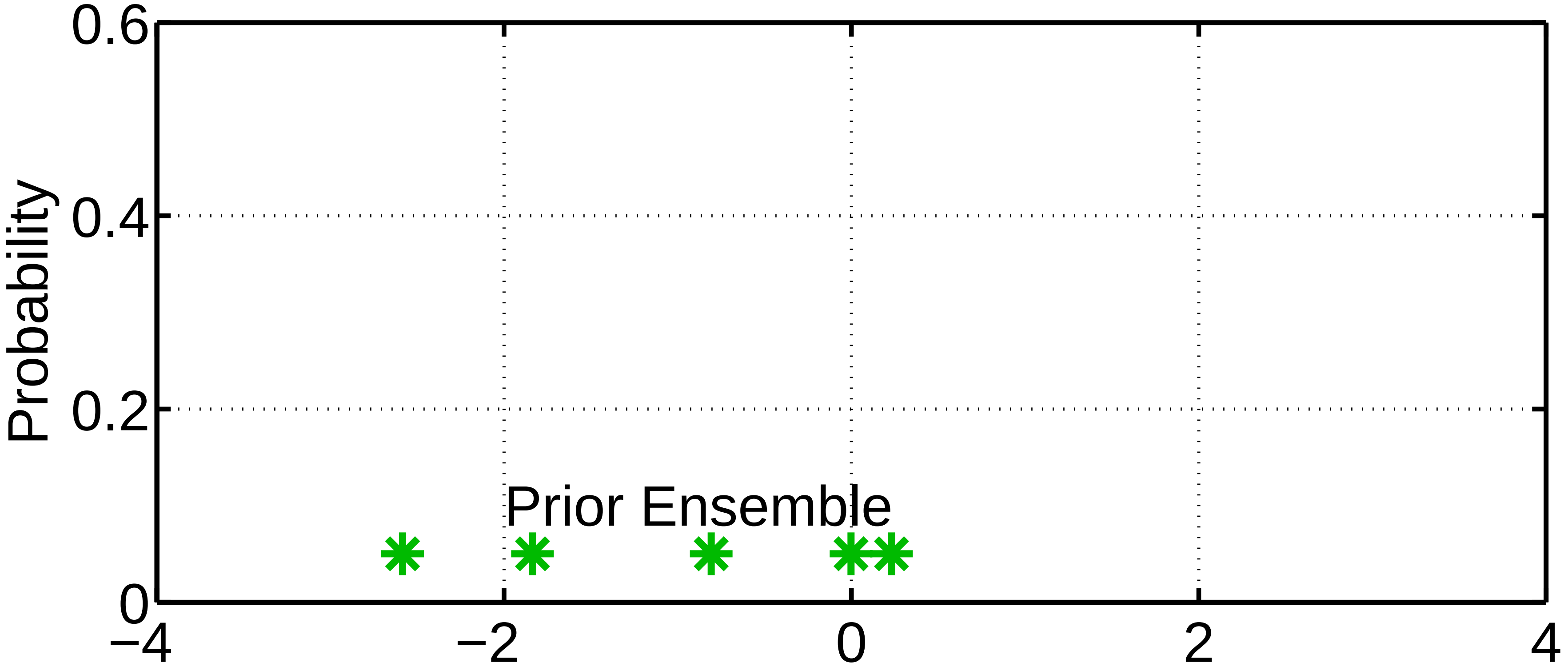


There are other families of functions for which it is closed...

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

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

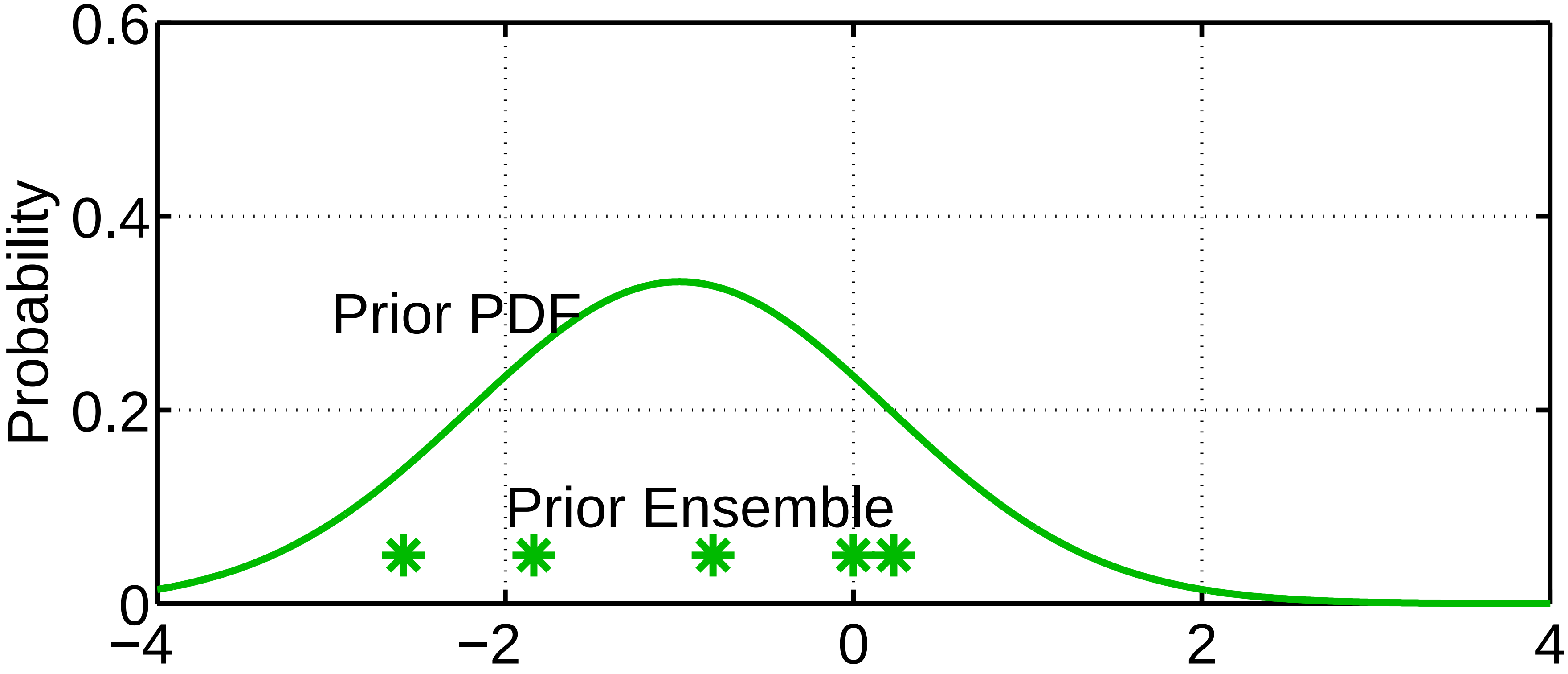
Ensemble filters: Prior is available as finite sample.



Don’t know much about properties of this sample.
May naively assume it is random draw from ‘truth’.

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

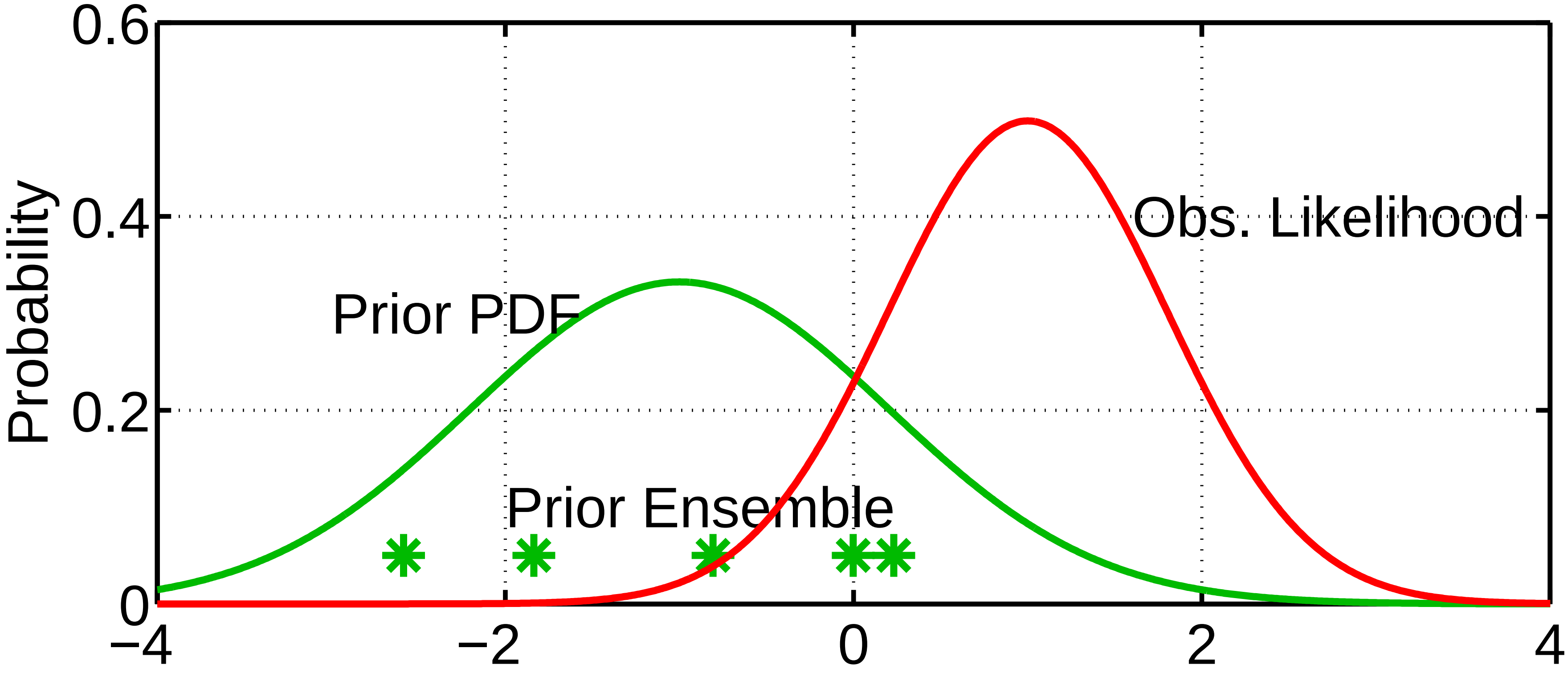
How can we take product of sample with continuous likelihood?



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

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

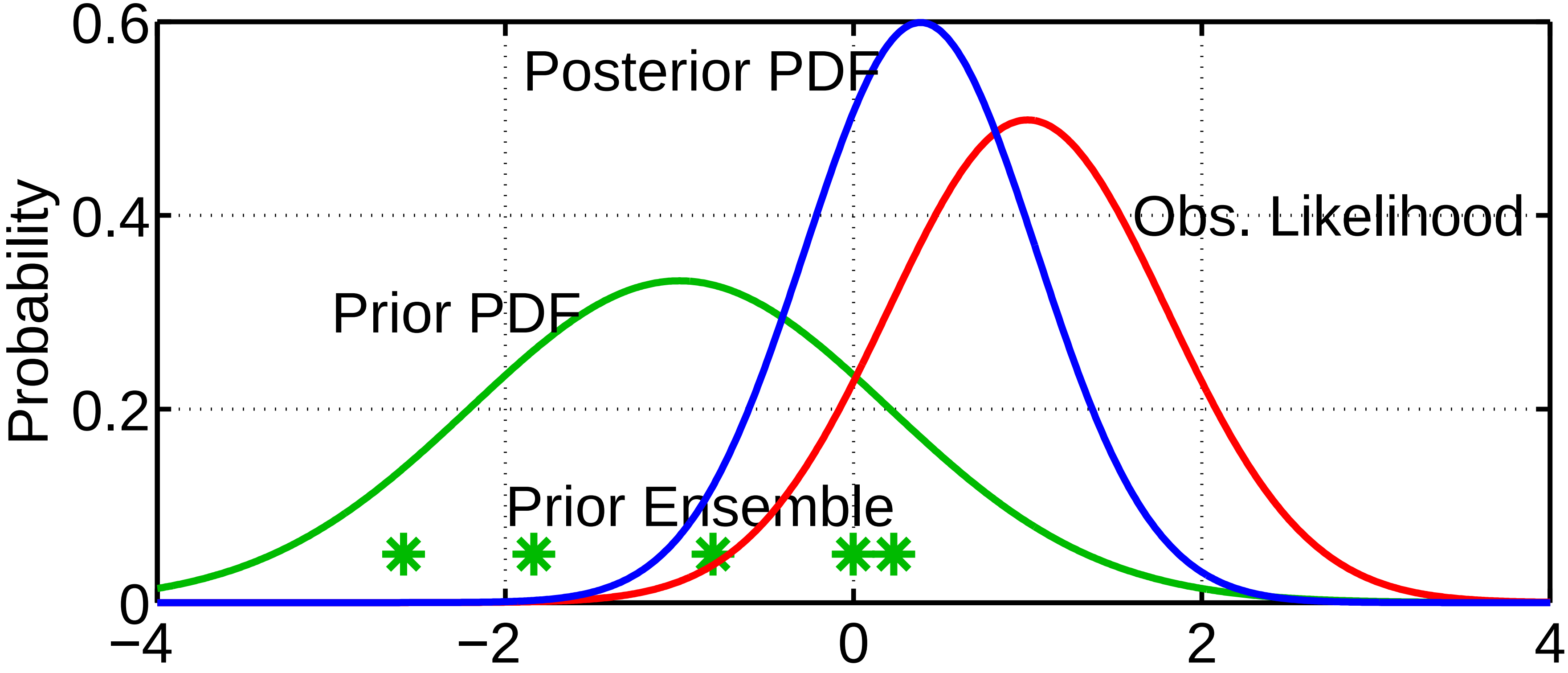
Observation likelihood usually continuous (nearly always Gaussian).



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

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

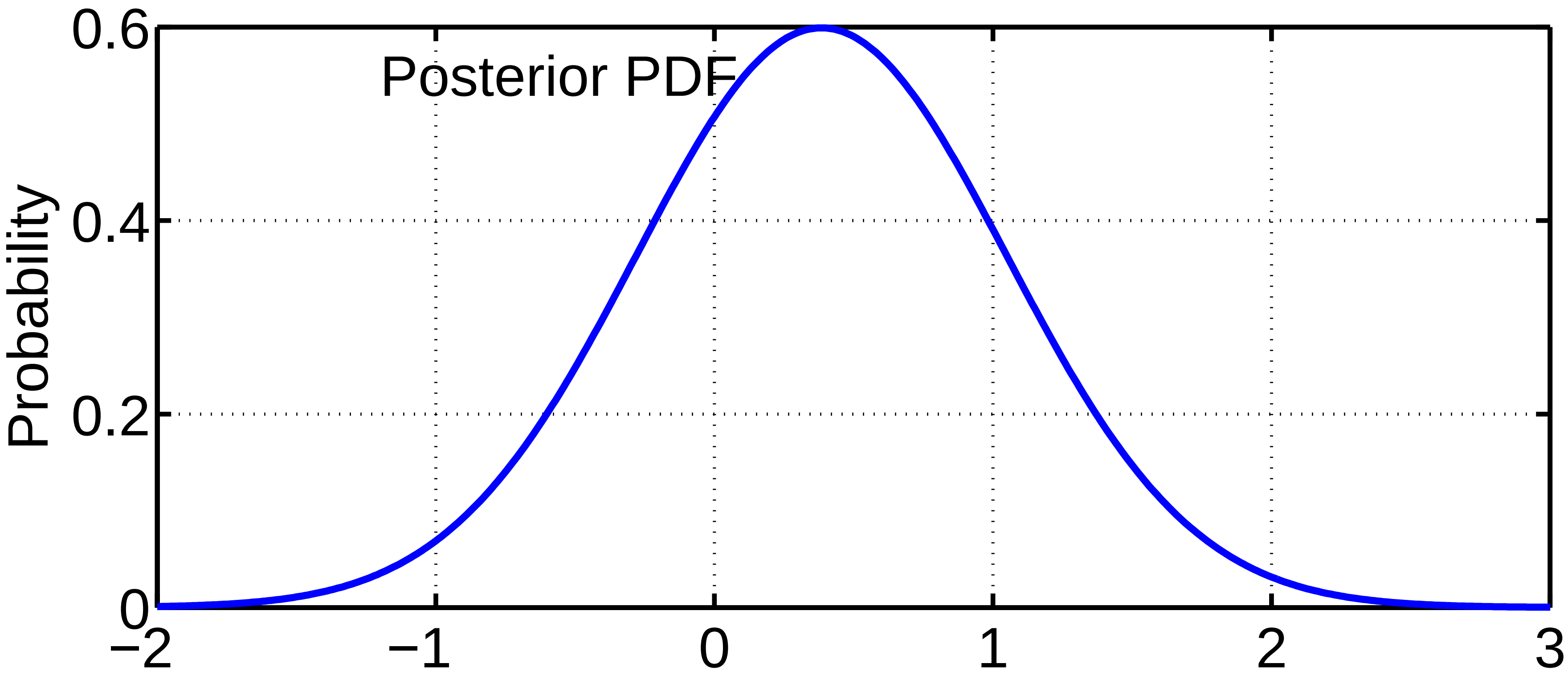
Product of prior Gaussian fit and Obs. likelihood is Gaussian.



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

Sampling Posterior PDF:

There are many ways to do this.



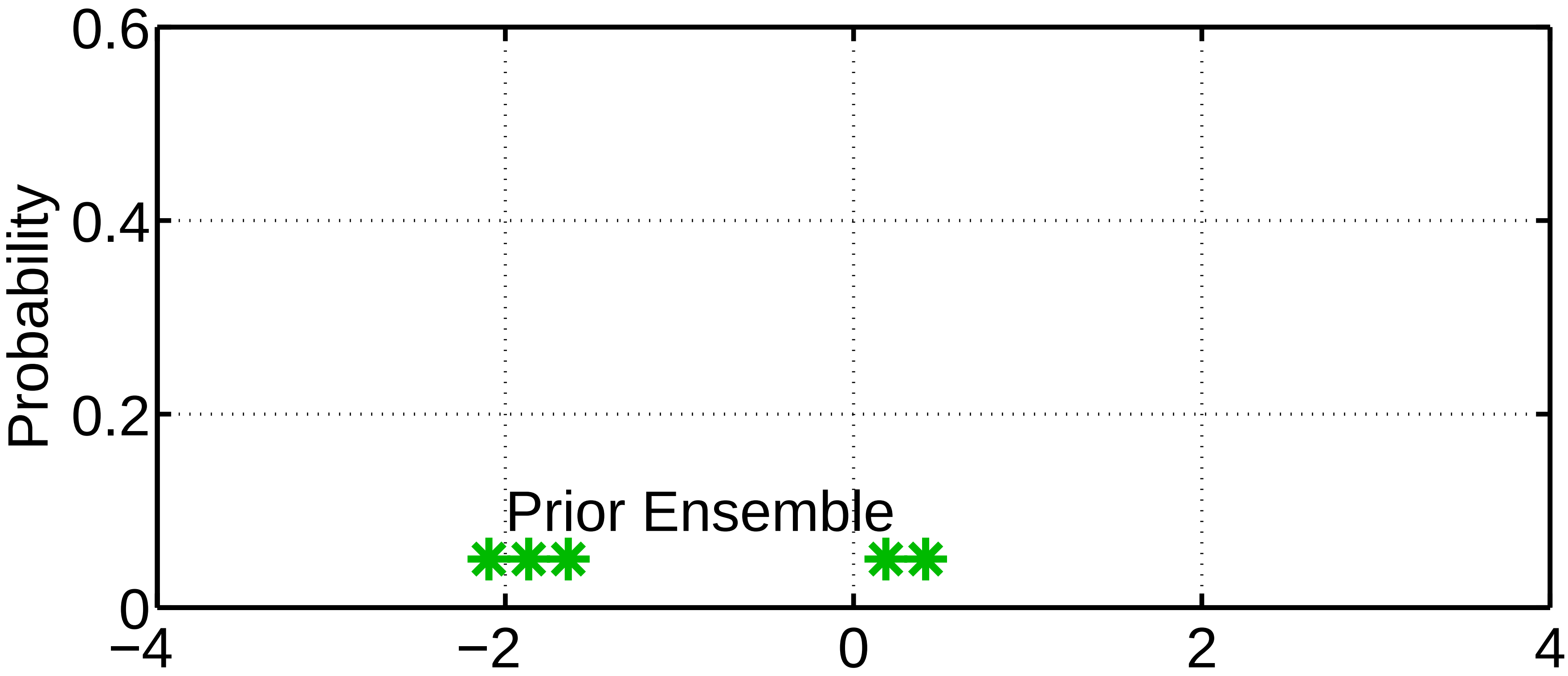
Exact properties of different methods may be unclear.

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

Will interact with properties of prediction models, etc.

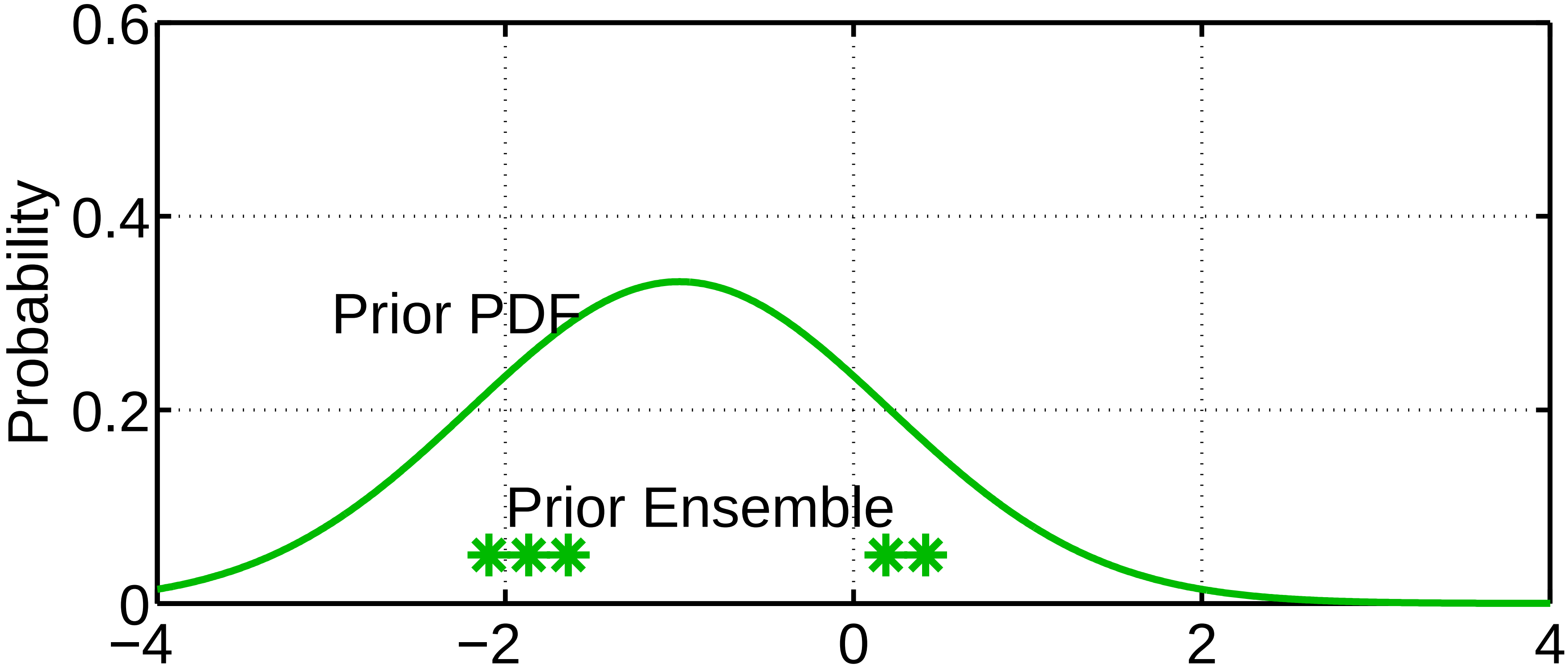
Sampling Posterior PDF:

Ensemble Adjustment (Kalman) Filter



Sampling Posterior PDF:

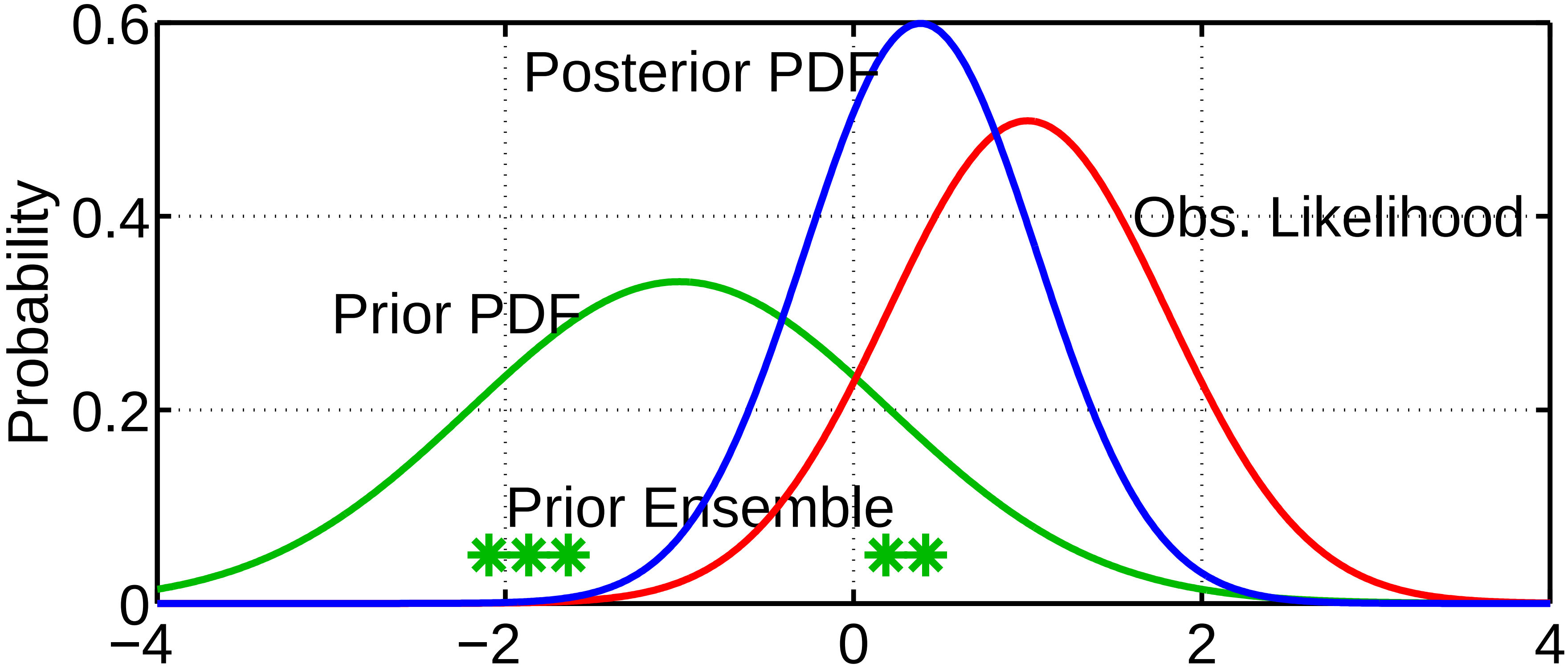
Ensemble Adjustment (Kalman) Filter.



Again, fit a Gaussian to sample.

Sampling Posterior PDF:

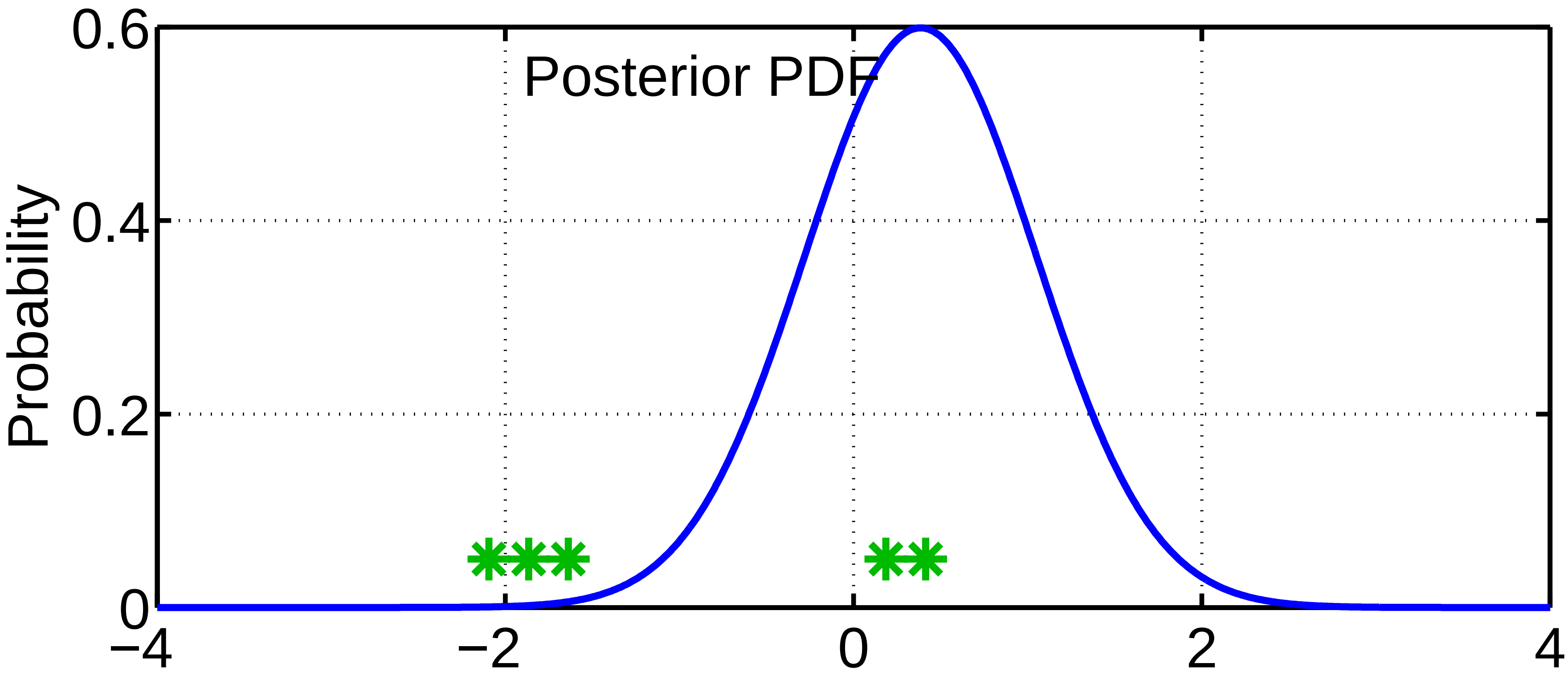
Ensemble Adjustment (Kalman) Filter



Compute posterior PDF (same as previous algorithms).

Sampling Posterior PDF:

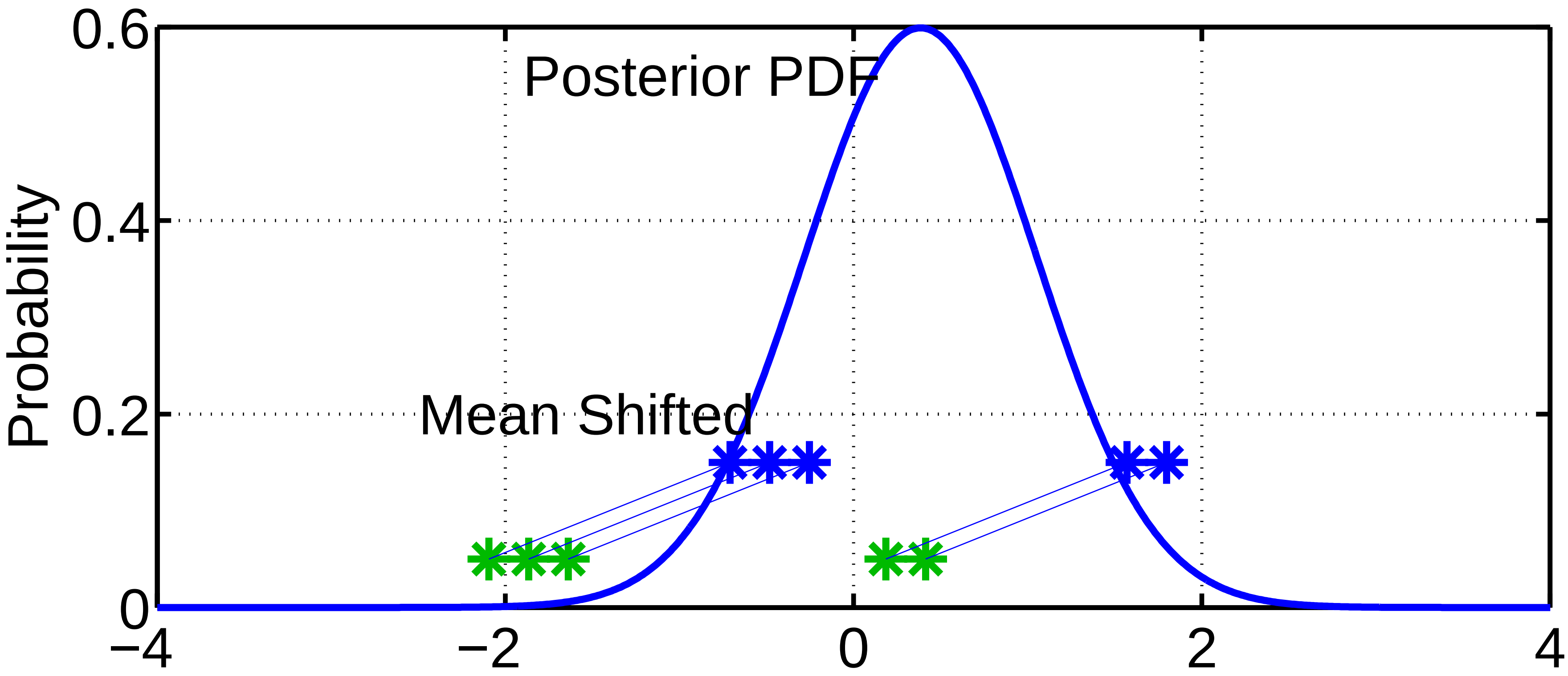
Ensemble Adjustment (Kalman) Filter



Use deterministic algorithm to ‘adjust’ ensemble.

Sampling Posterior PDF:

Ensemble Adjustment (Kalman) Filter.

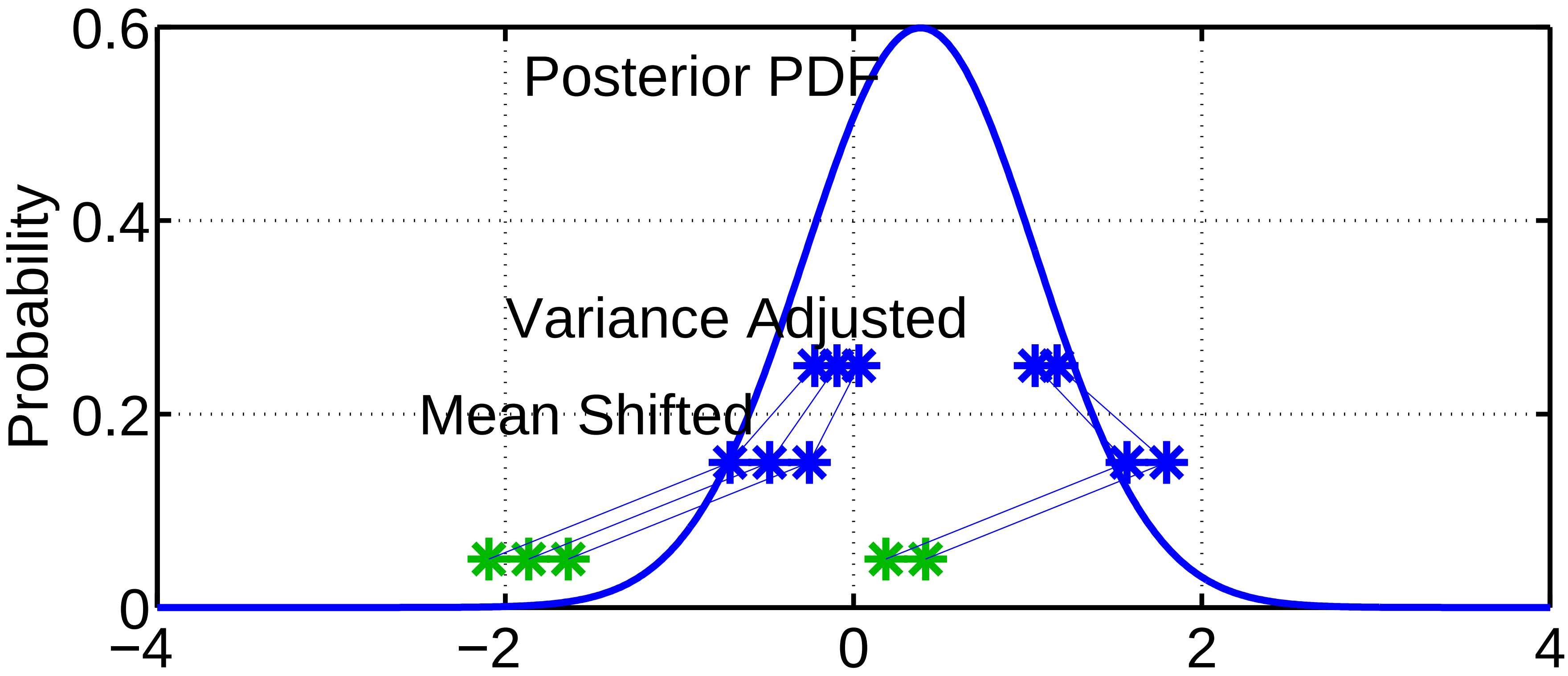


Use deterministic algorithm to ‘adjust’ ensemble.

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

Sampling Posterior PDF:

Ensemble Adjustment (Kalman) Filter.



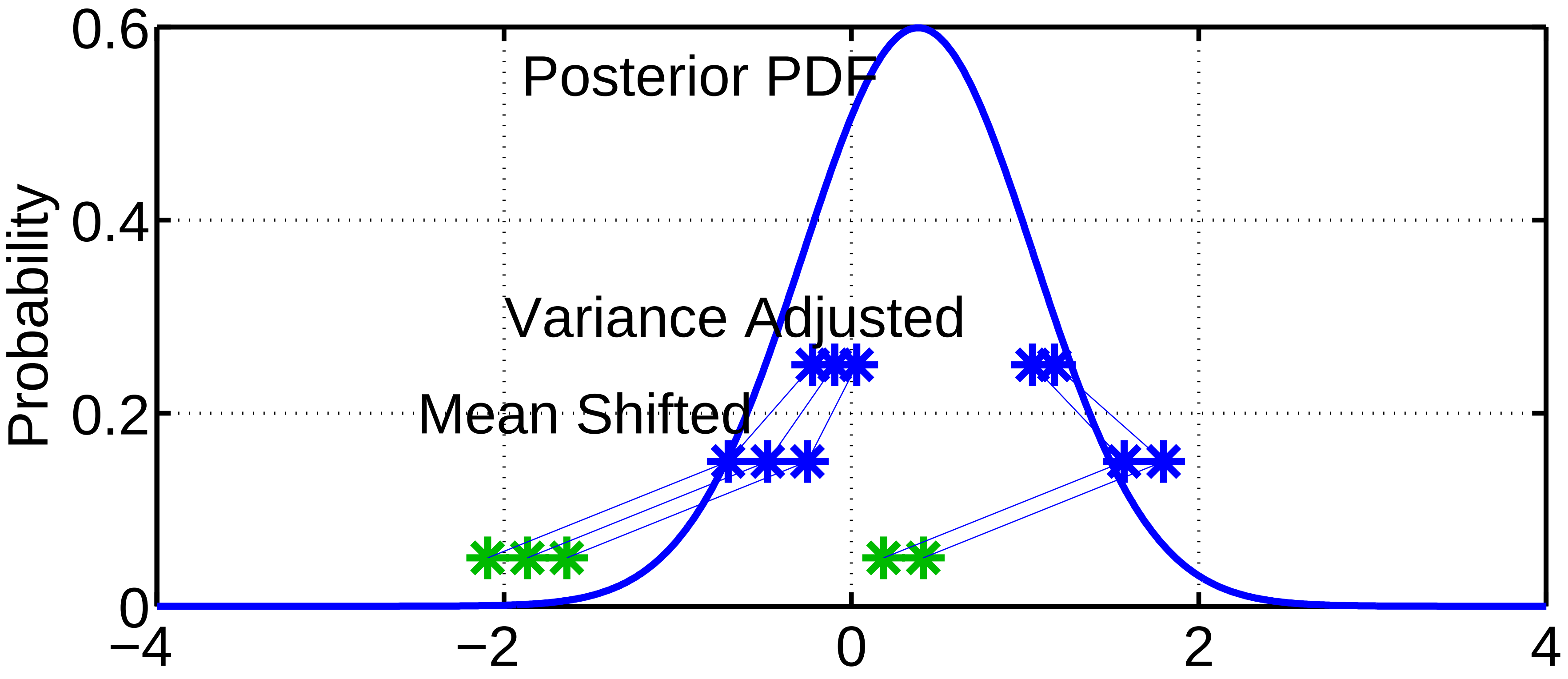
Use deterministic algorithm to ‘adjust’ ensemble.

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

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

Sampling Posterior PDF:

Ensemble Adjustment (Kalman) Filter.

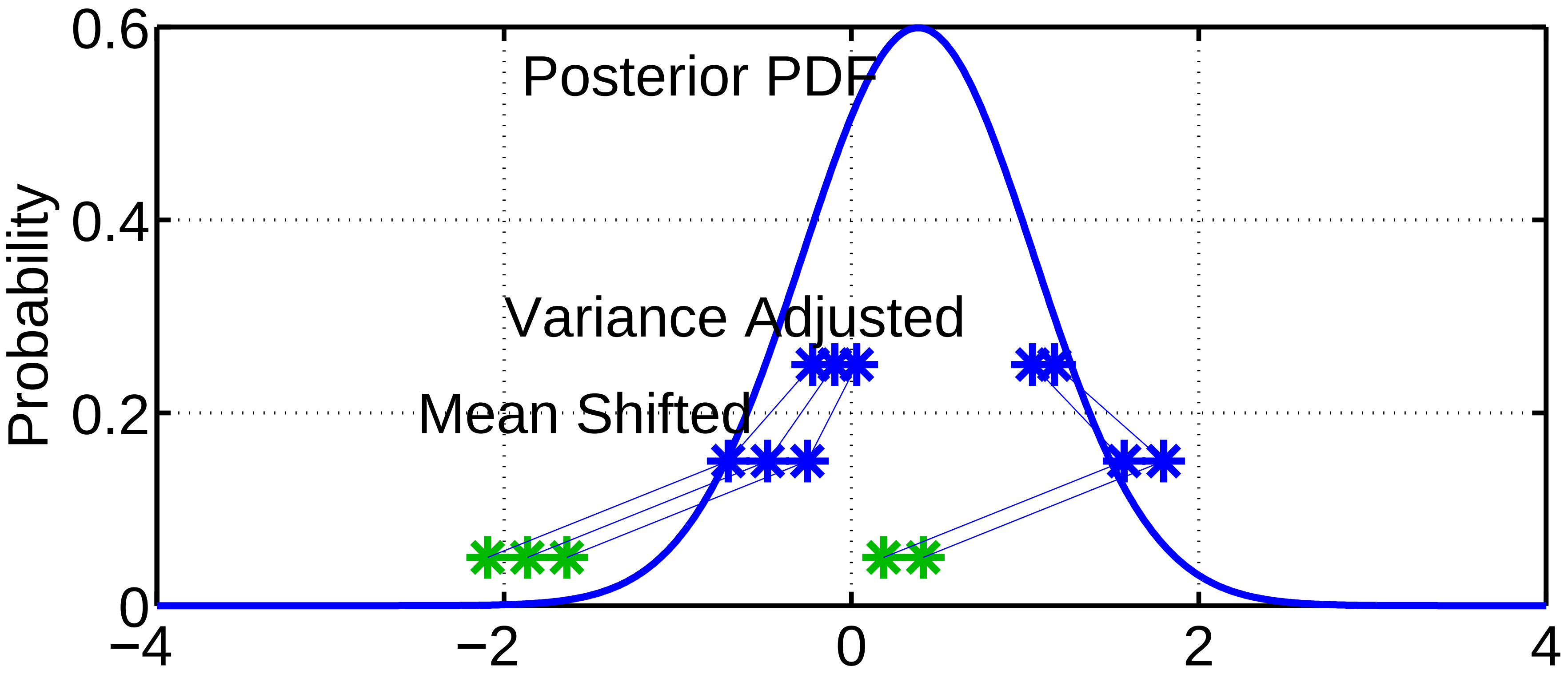


$x_i^u = (x_i^p - \overline{x^p}) \cdot (\sigma^u / \sigma^p) + \overline{x^u}$ i = 1,..., ensemble size.

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

Sampling Posterior PDF:

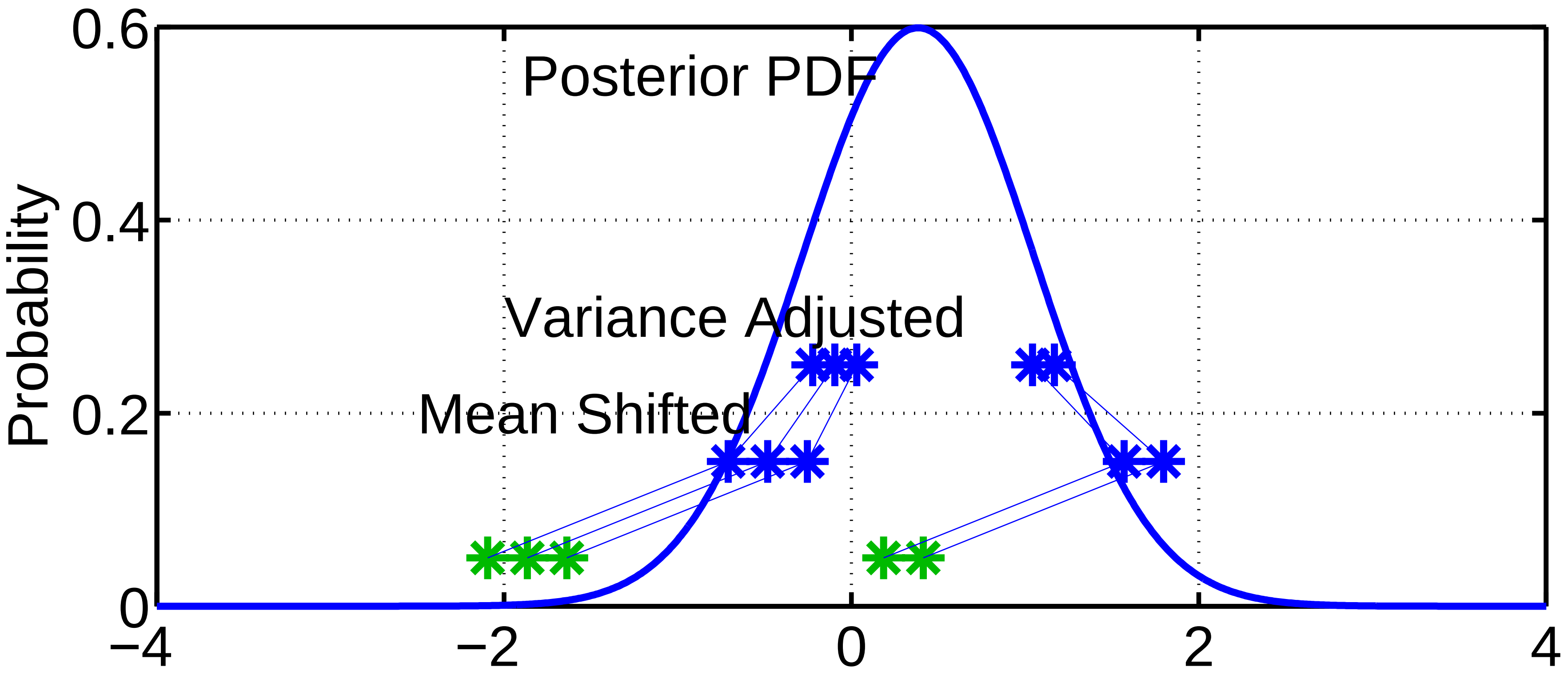
Ensemble Adjustment (Kalman) Filter.



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

Sampling Posterior PDF:

Ensemble Adjustment (Kalman) Filter.



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

First look at DART Diagnostics

cd models/lorenz_63/work in your DART sandbox.
csh workshop_setup.csh Does stuff you'll learn to do later.
matlab -nojvm

Output from a DART assimilation in 3-variable model.
20 member ensemble.
Observations of each variable once every '6 hours'; error variance 8.
Observation ONLY impacts its own state variable.
For assimilation, looks like 3 independent single variable problems.
Model advance between assimilations isn't independent.

Initial ensemble members are random selection from long model run.
Initial error should be an upper bound (random guess).

First look at DART Diagnostics

Try the following matlab commands:

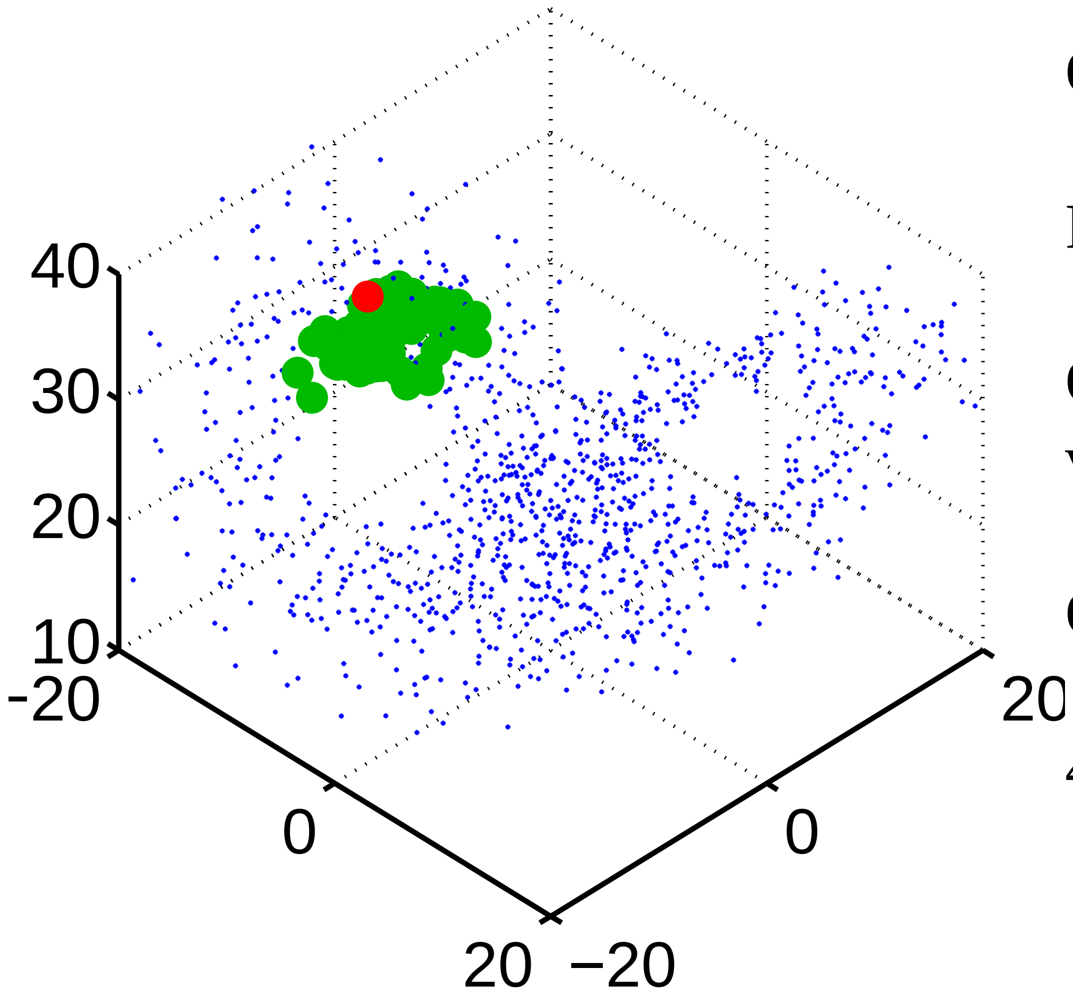
plot_total_err time series of distance between prior ensemble mean and truth in blue; spread: average prior distance between ensemble members and mean in red.

(Record total values of total error and spread for later).

plot_ens_mean_time_series time series of truth in blue; ensemble mean prior.

plot_ens_time_series Also includes prior ensemble members.

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



Observation in red.

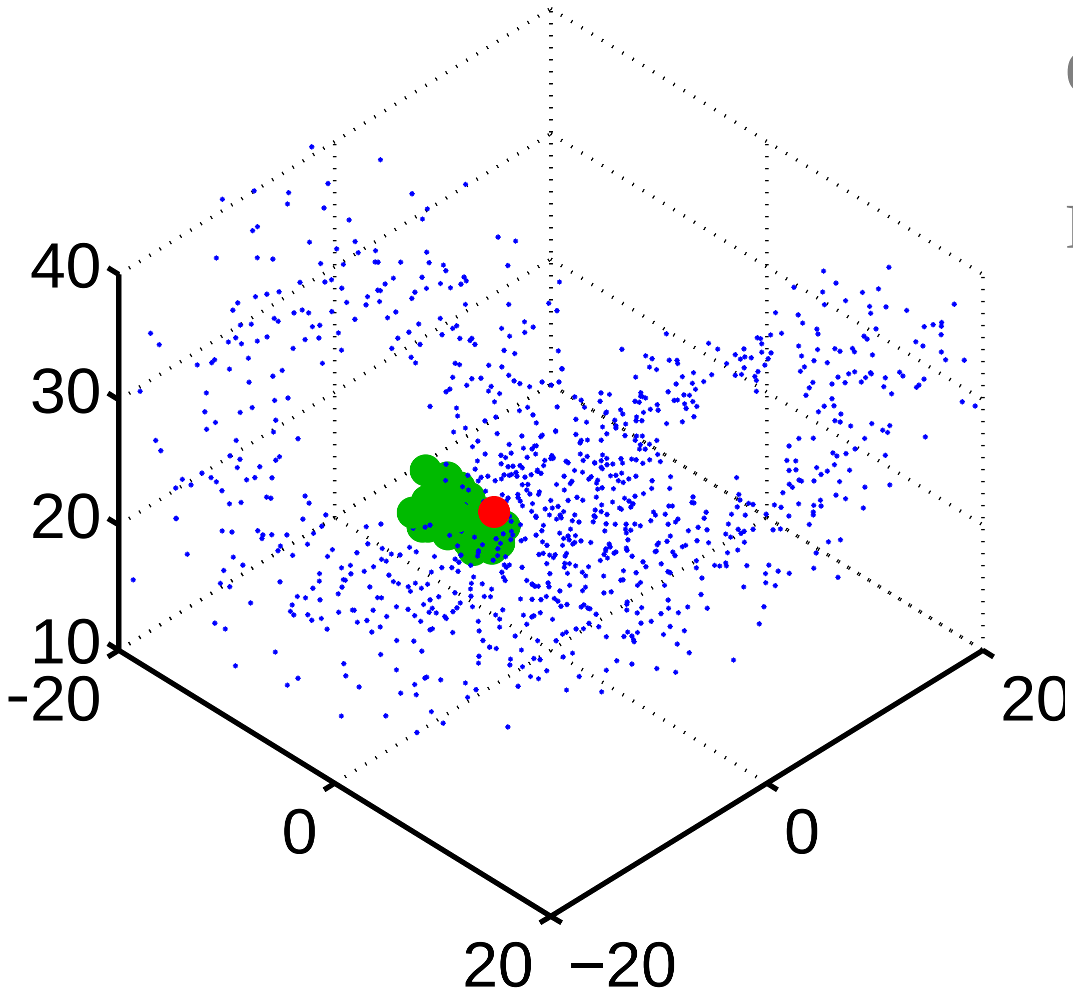
Prior ensemble in green.

Observing all three state variables.

Obs. error variance = 4.0.

4 20-member ensembles.

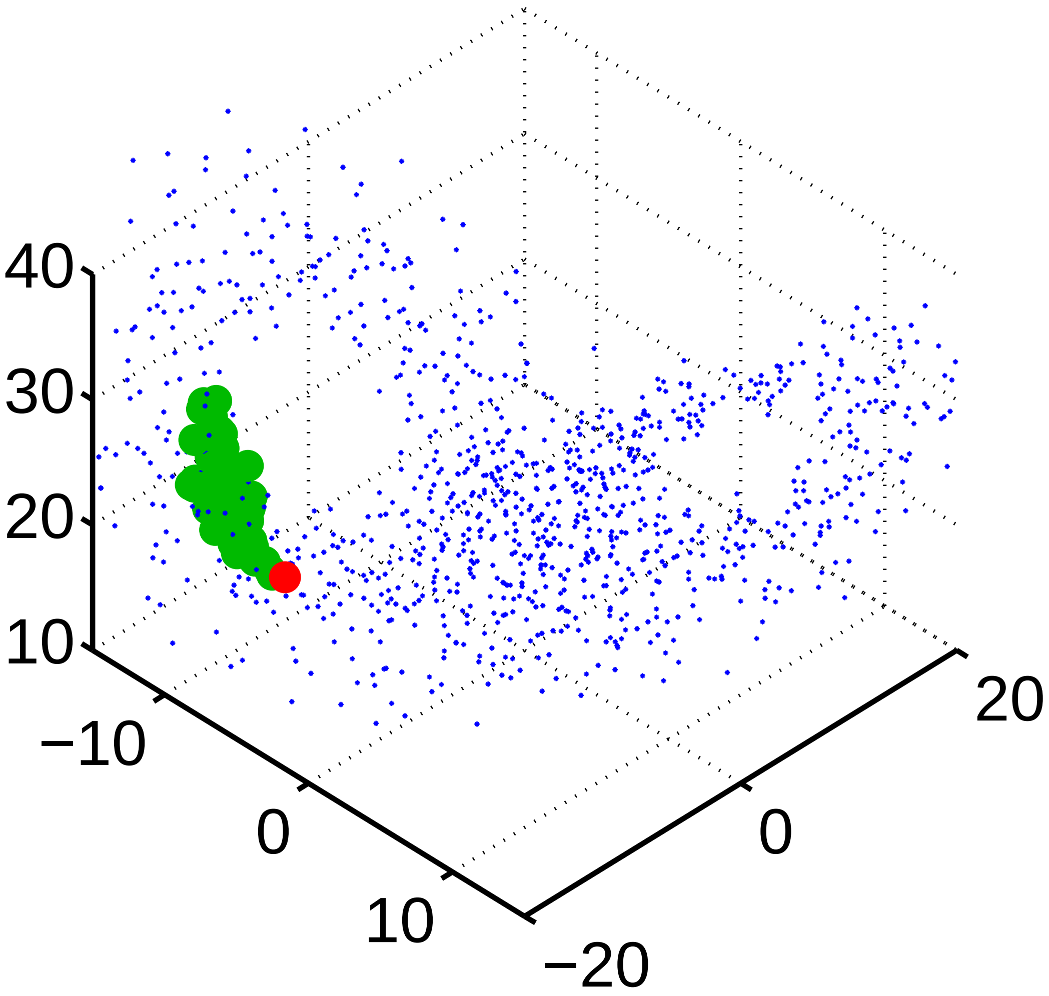
Simple example: Lorenz-63 3-variable chaotic model.



Observation in red.

Prior ensemble in green.

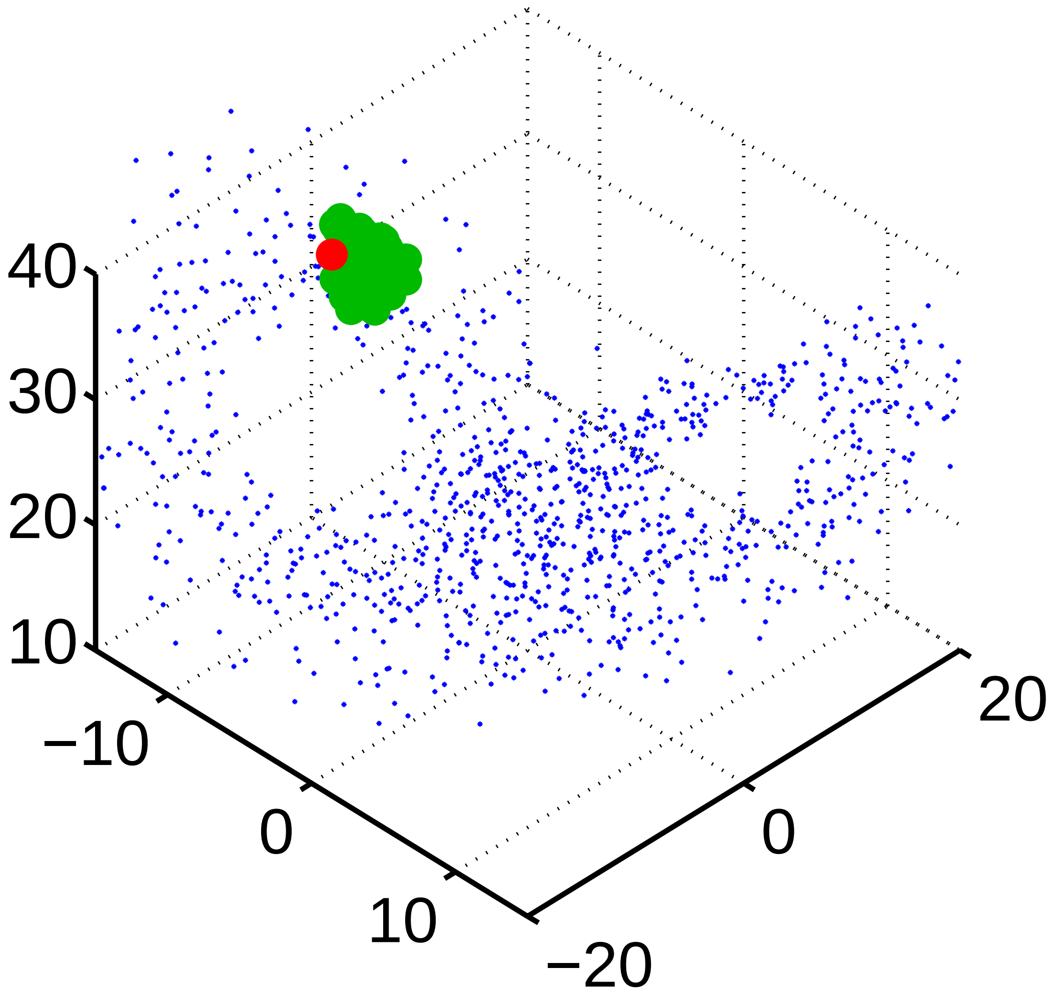
Simple example: Lorenz-63 3-variable chaotic model.



Observation in red.

Prior ensemble in green.

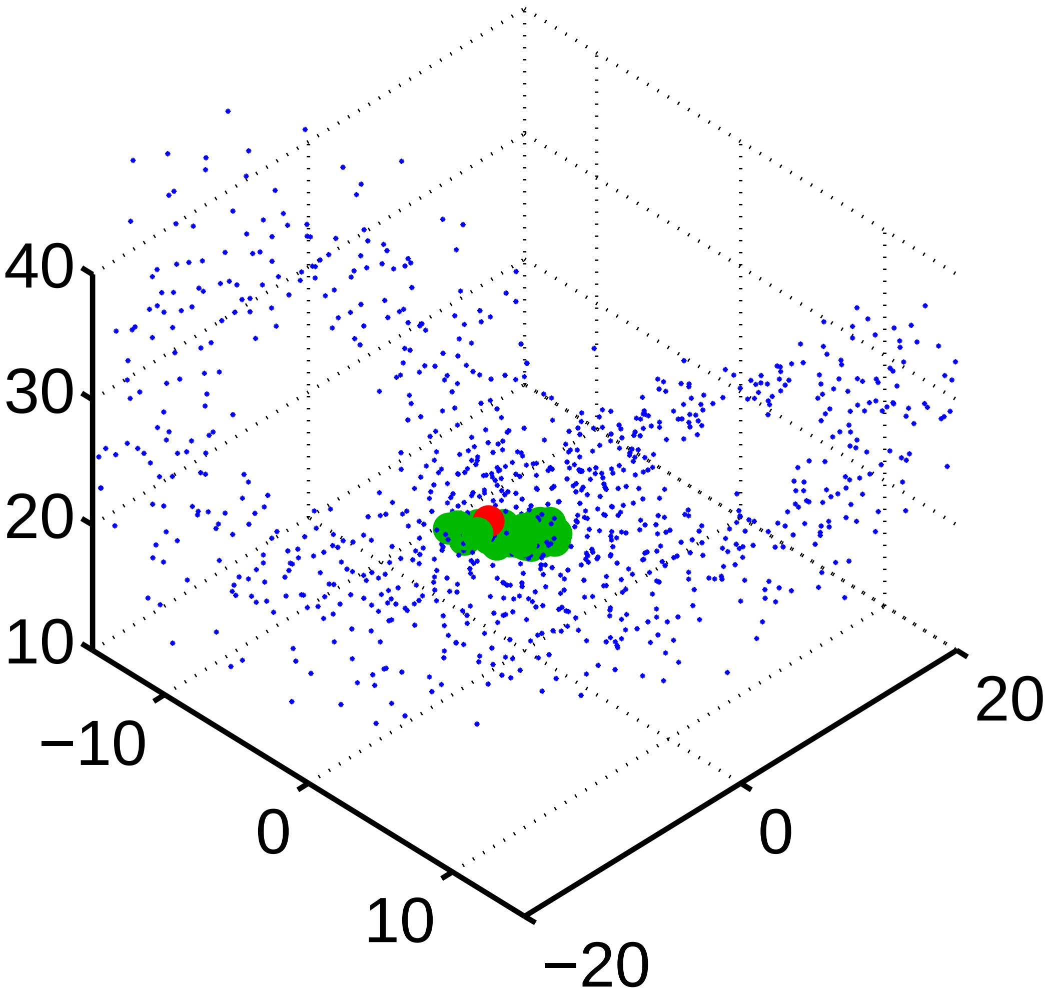
Simple example: Lorenz-63 3-variable chaotic model.



Observation in red.

Prior ensemble in green.

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

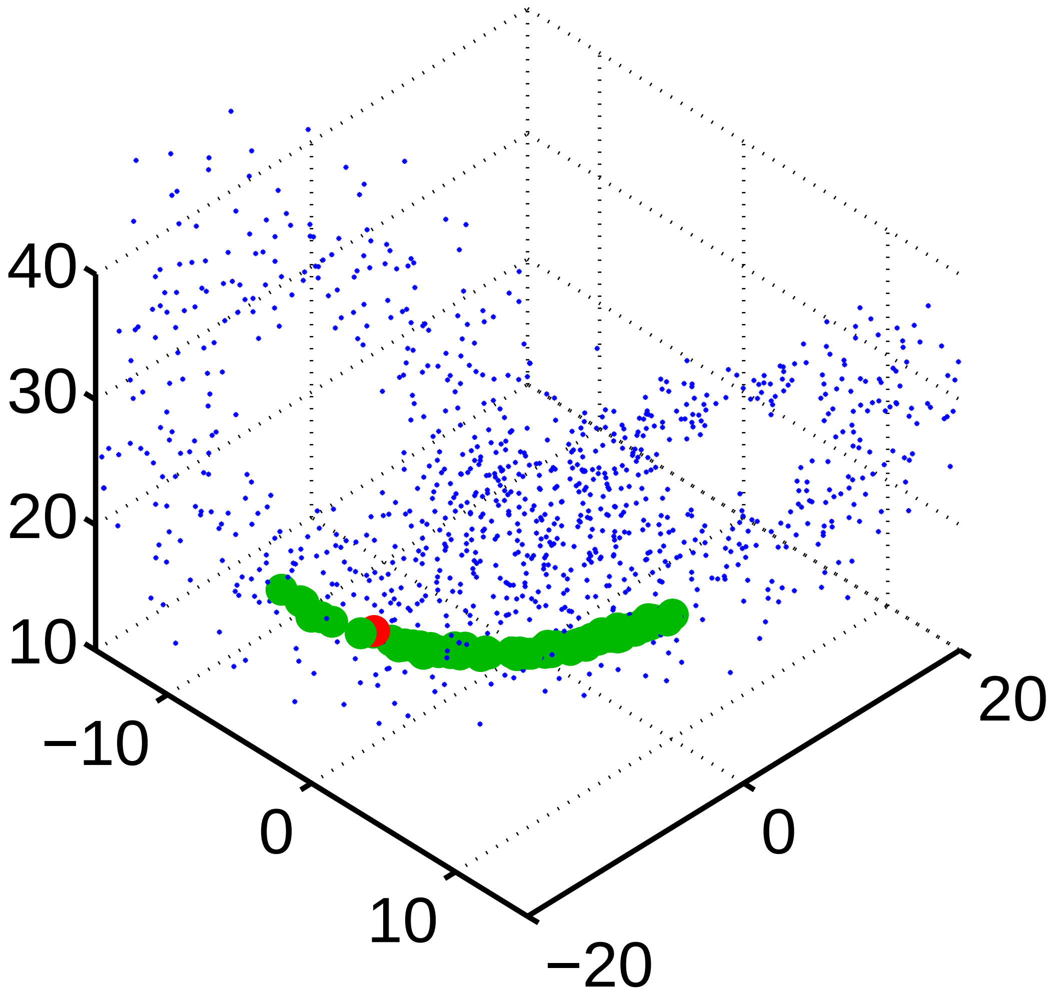


Observation in red.

Prior ensemble in green.

Ensemble is passing through unpredictable region.

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

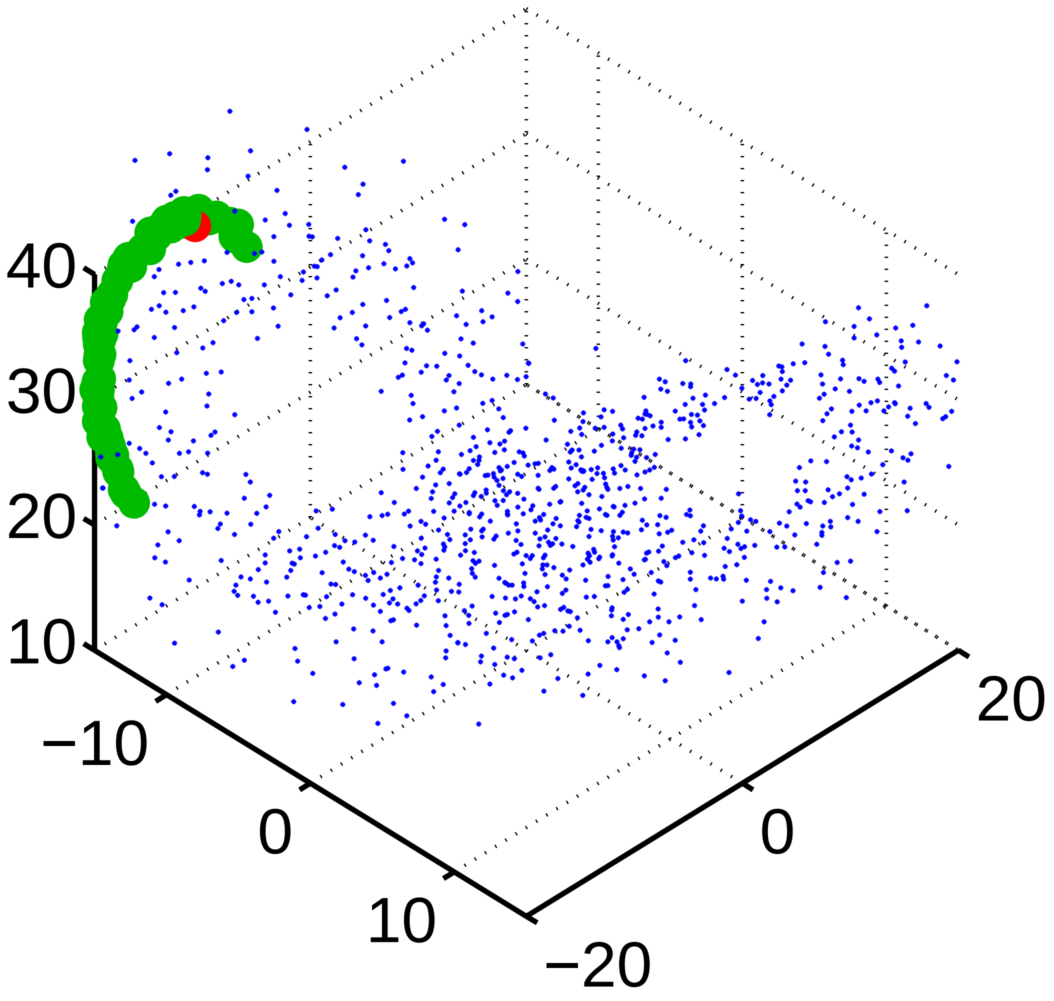


Observation in red.

Prior ensemble in green.

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

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



Observation in red.

Prior ensemble in green.

Using DART Diagnostics

Using DART diagnostics from the simple Lorenz-63 assimilation:

Can you see evidence of enhanced uncertainty?

Where does this occur?

Does the ensemble appear to be consistent with the truth?
(Is the truth normally inside the ensemble range?)