Nonlinear, Non-Gaussian Ensemble Filters for Data Assimilation

Jeffrey Anderson
NCAR Data Assimilation Research Section (DAReS)

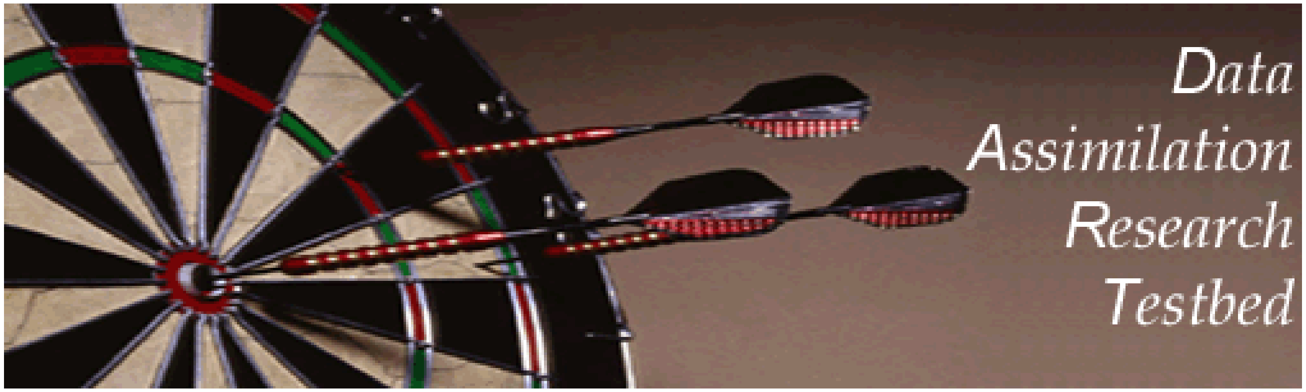
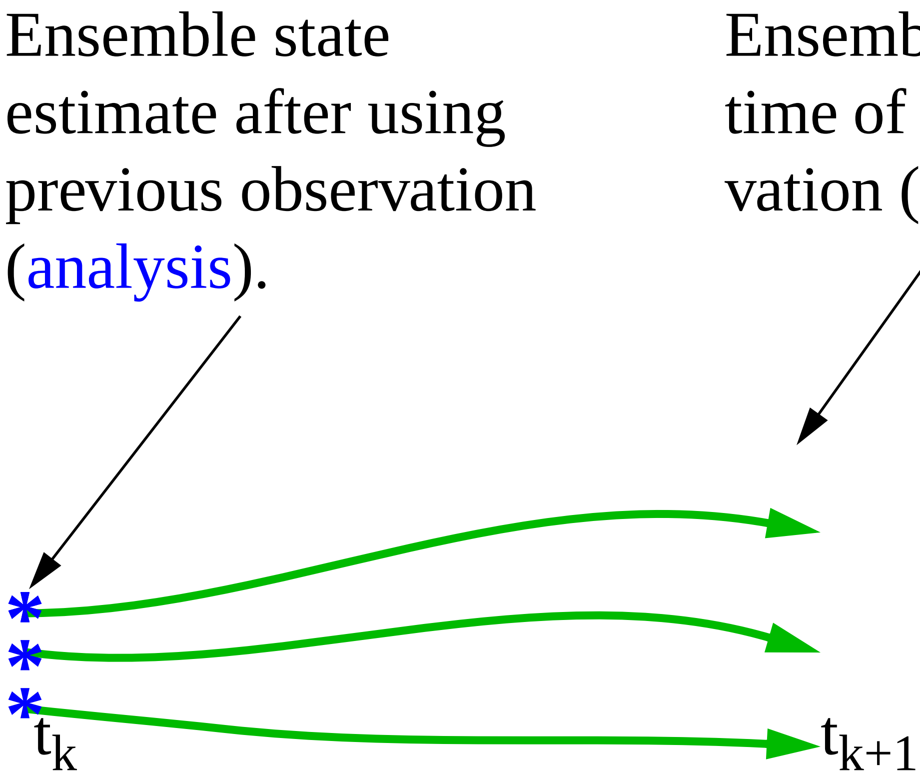


IMAGe Seminar
21 November 2008

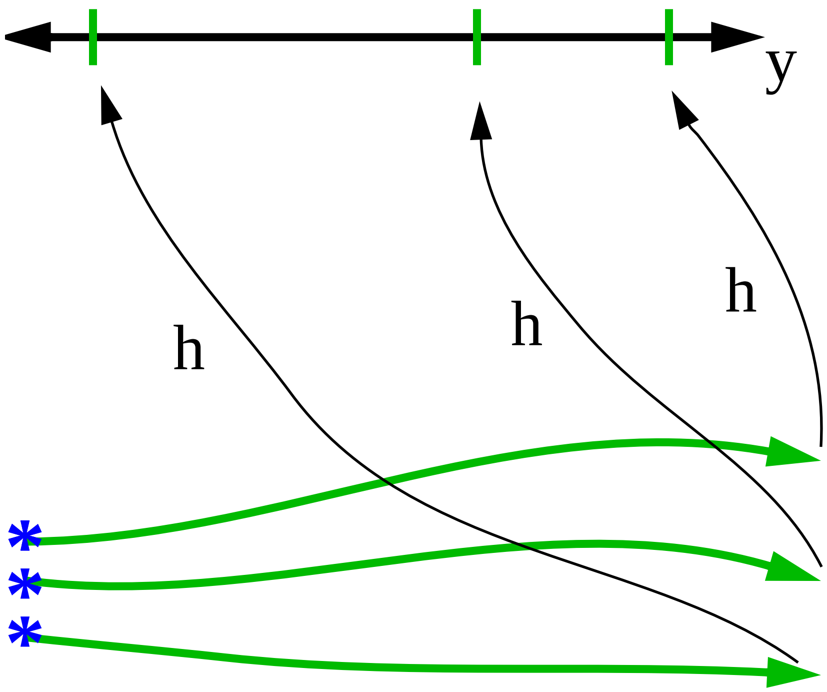
Ensemble Filter Overview.

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



Ensemble Filter Overview.

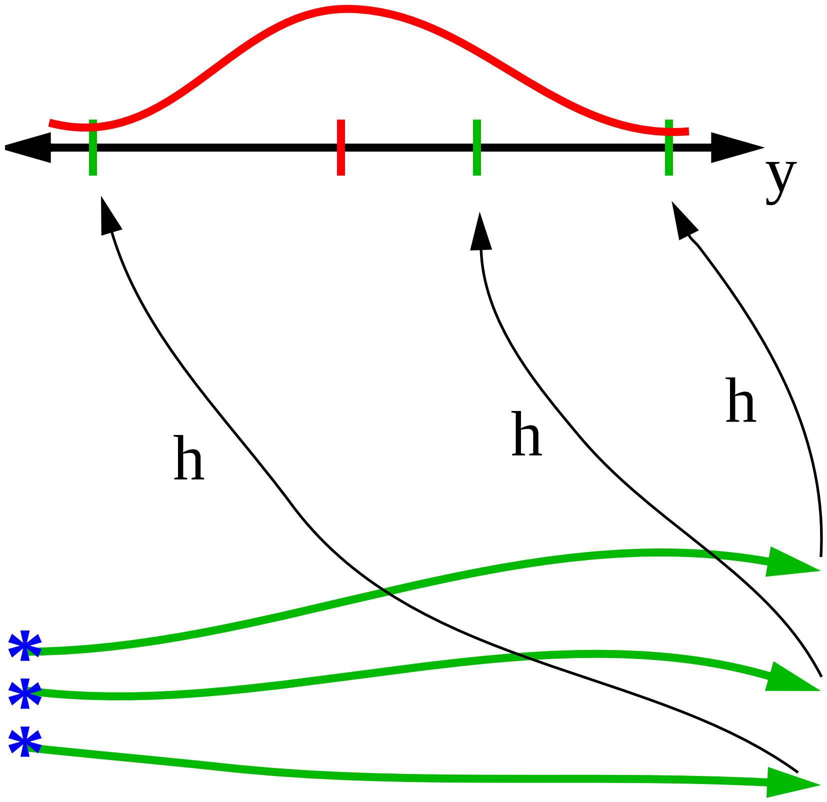
2. Get prior ensemble sample of observation, y=h(x), by applying forward operator h to each ensemble member.



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

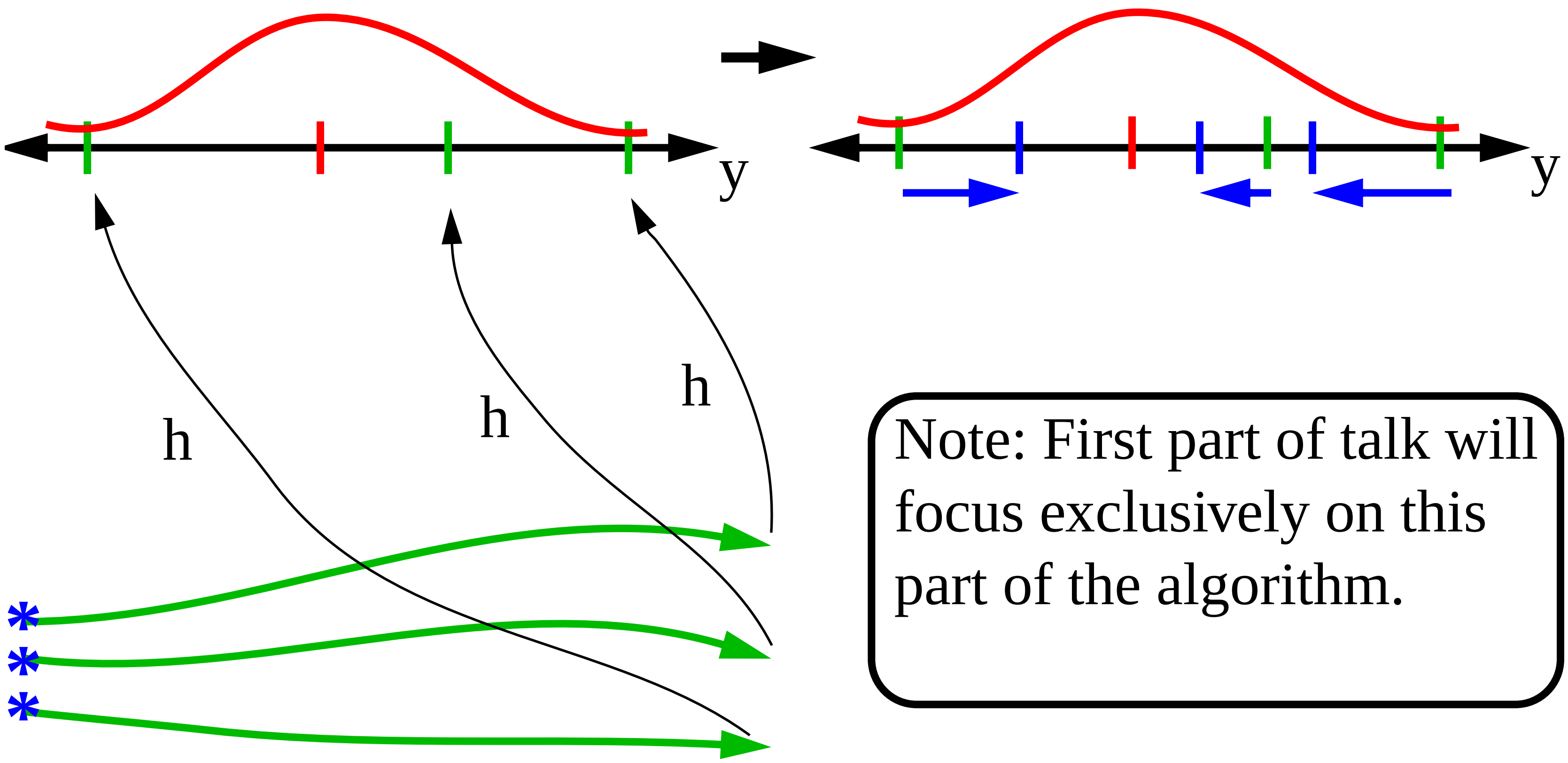
Ensemble Filter Overview.

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



Ensemble Filter Overview.

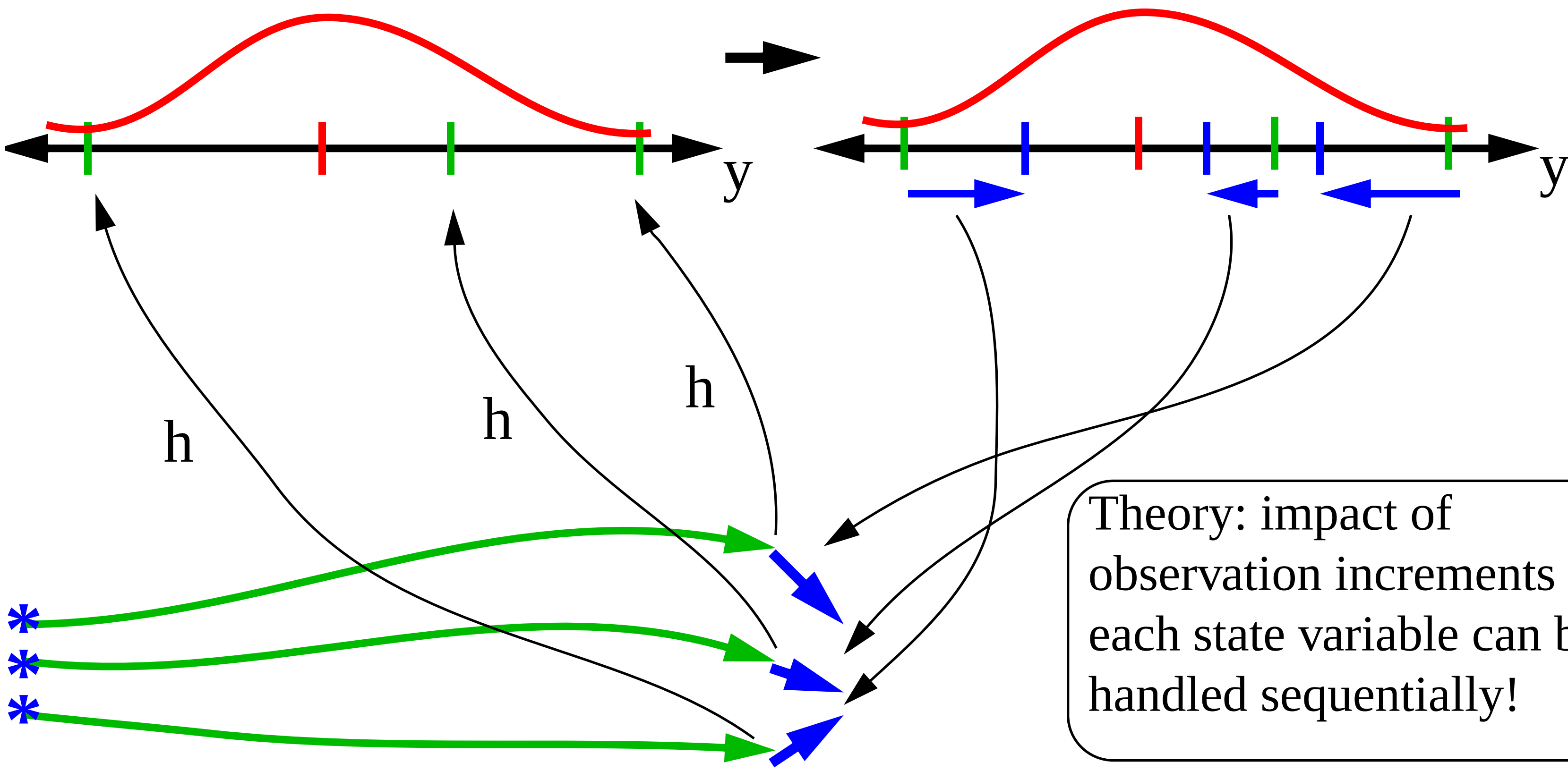
4. Find increment for each prior observation ensemble
(this is a scalar problem for uncorrelated observation errors).



Note: First part of talk will focus exclusively on this part of the algorithm.

Ensemble Filter Overview.

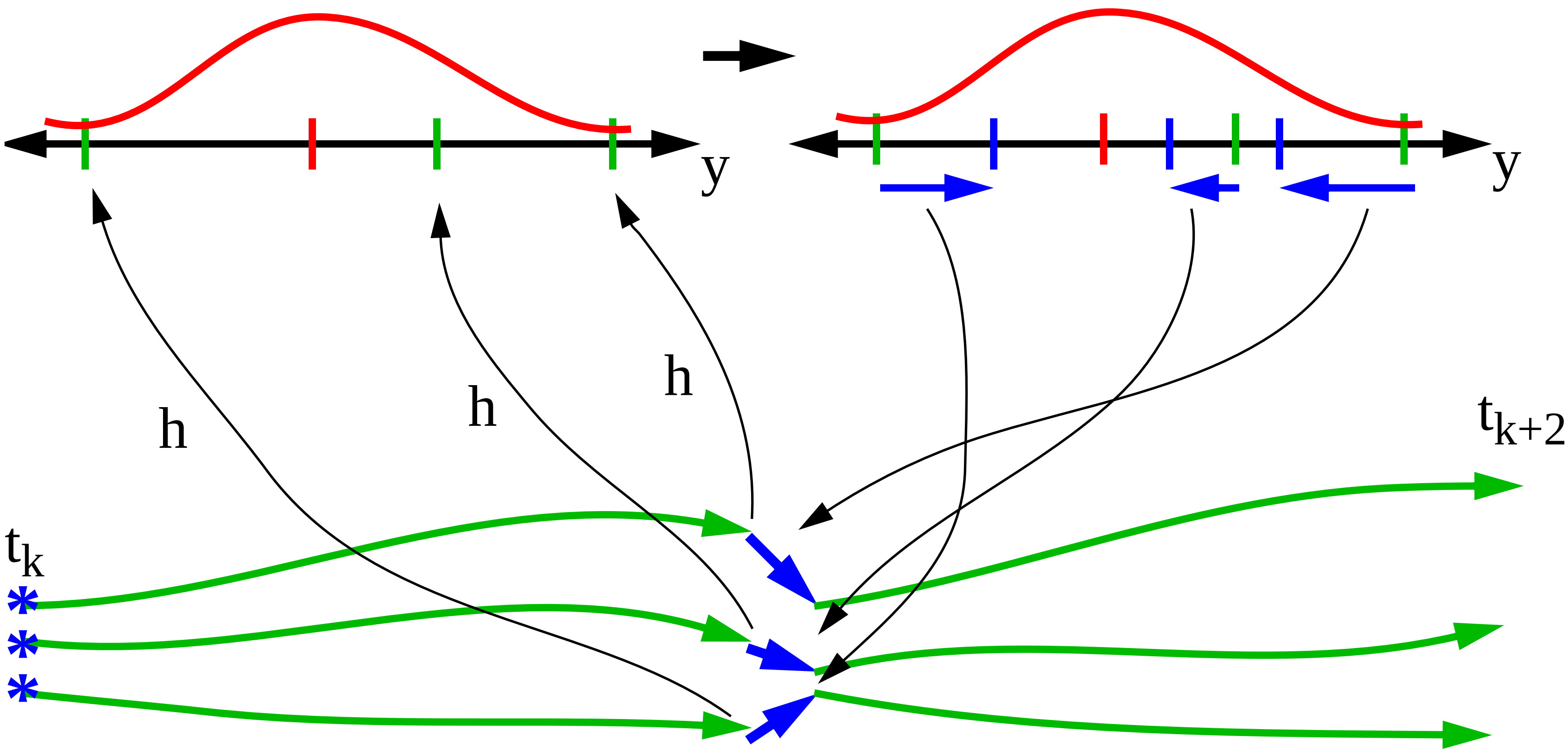
5. Use ensemble samples of y and each state variable to linearly regress observation increments onto state variable increments.



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

Ensemble Filter Overview.

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



Note: Consistent Color Scheme Throughout

Green = Prior

Red = Observation

Blue = Posterior

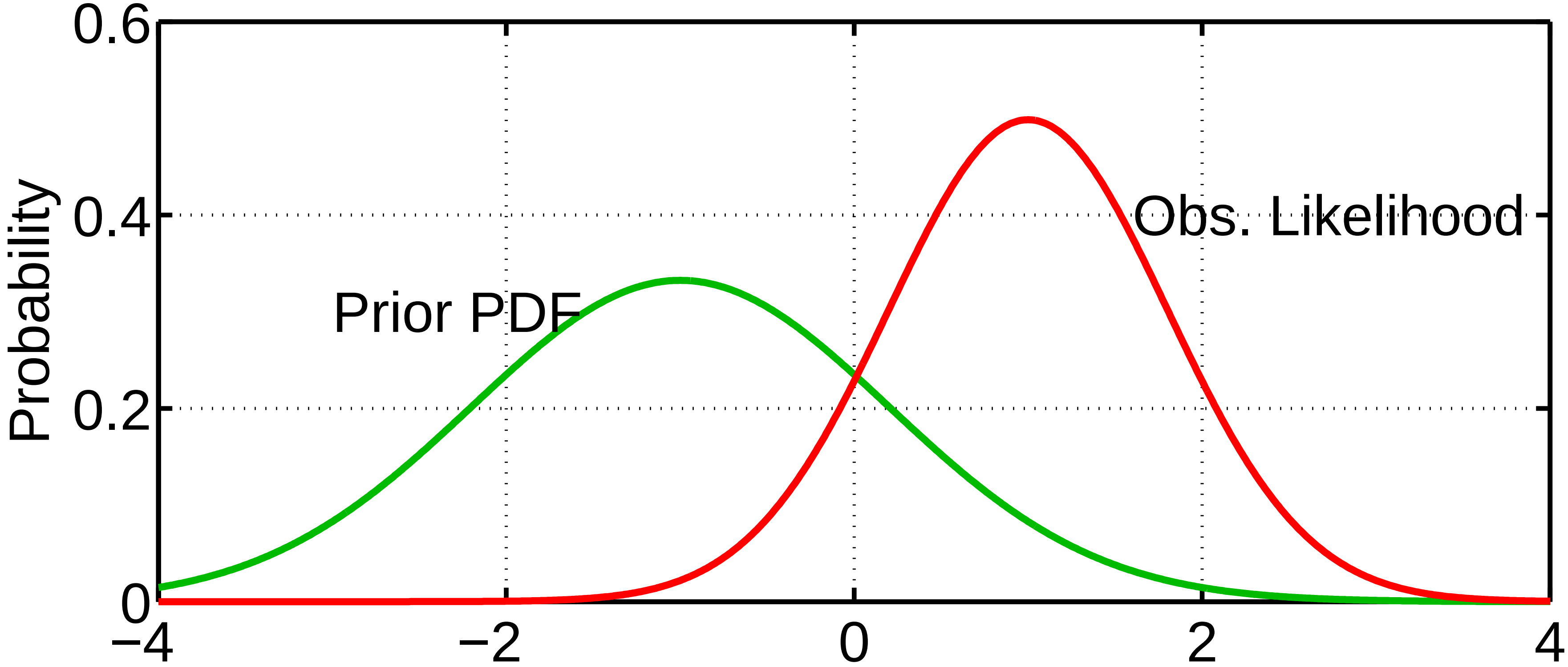
Two most common observation space update algorithms:

1. EAKF: Ensemble Adjustment KF (deterministic square root);

2. EnKF: Ensemble KF (Monte Carlo approximation).

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

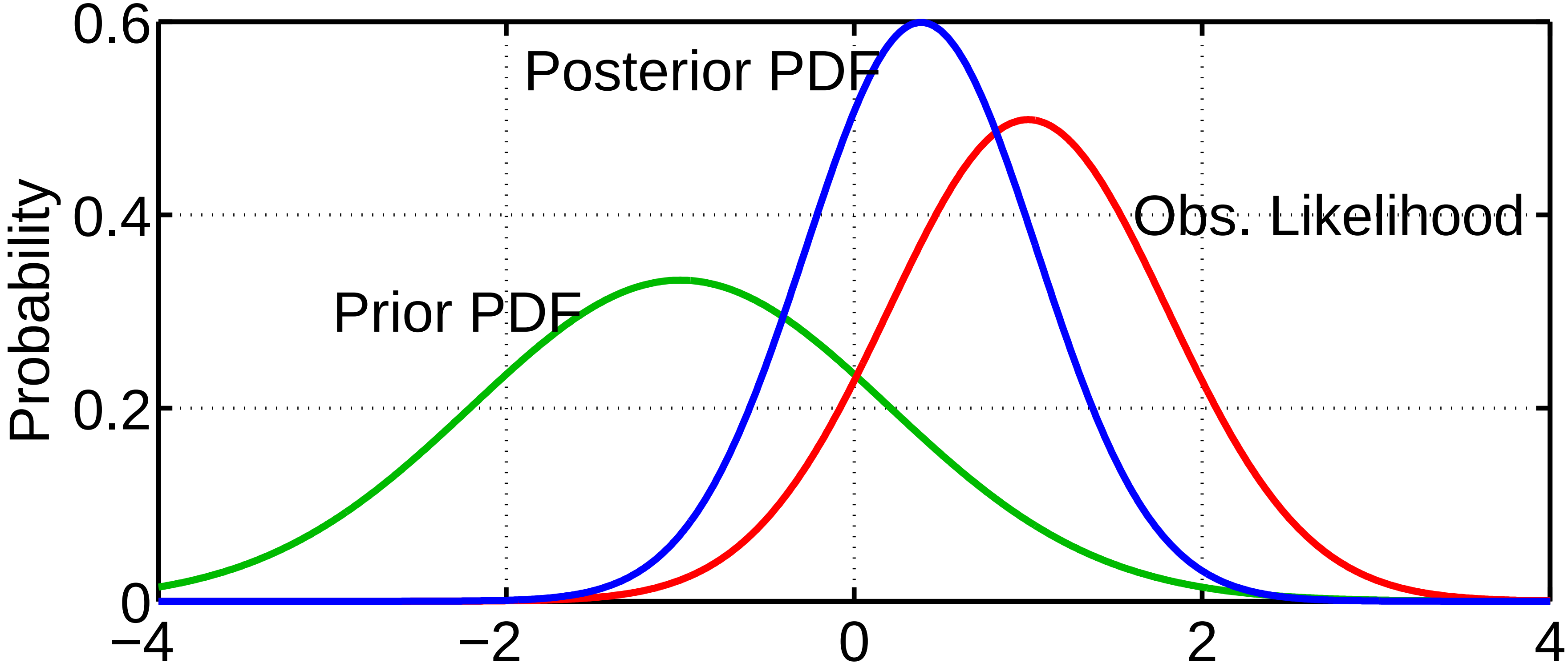
This product is closed for Gaussian distributions.



Posterior = Likelihood * Prior / Normalization

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.

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

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

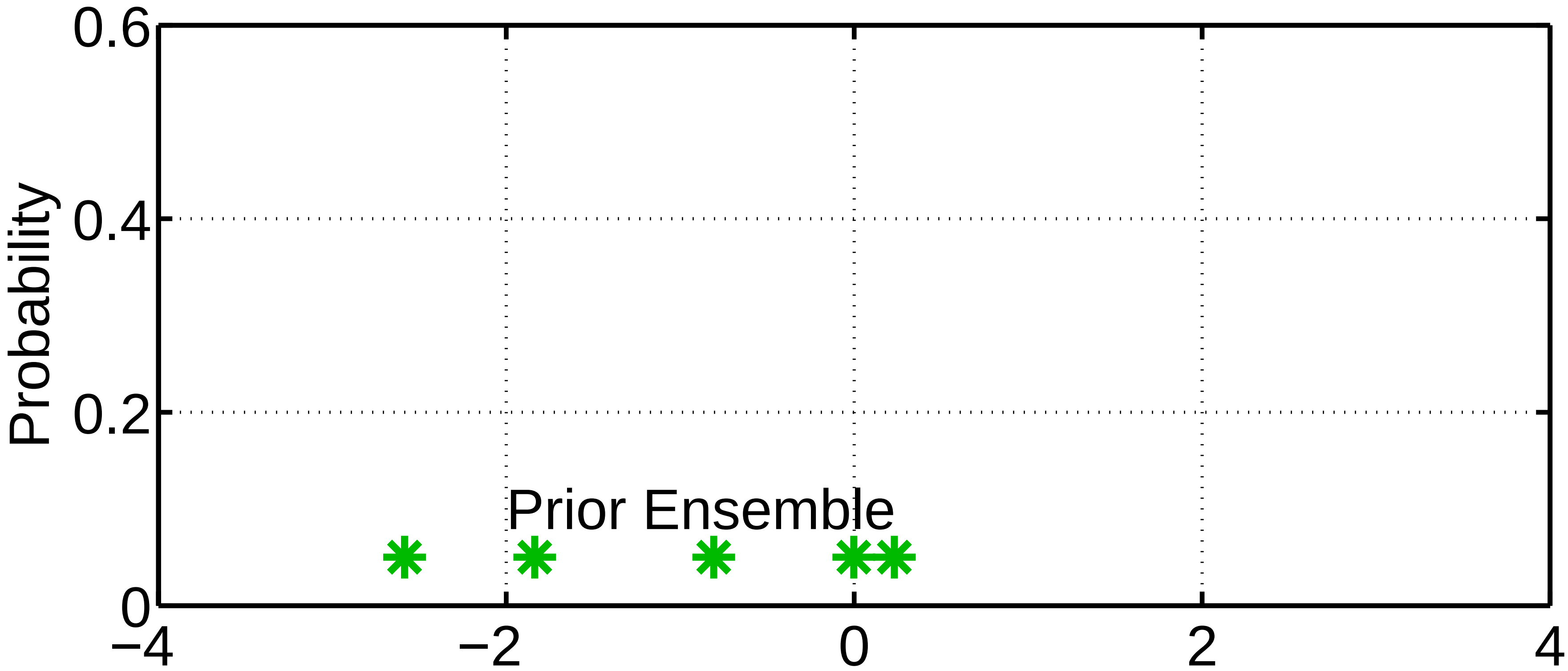
Weight: $c = \dfrac{1}{(2\Pi)^{d/2}|\Sigma_1 + \Sigma_2|^{1/2}} \exp\left\{-\frac{1}{2}[(\mu_2 - \mu_1)^T(\Sigma_1 + \Sigma_2)^{-1}(\mu_2 - \mu_1)]\right\}$

Ignore the weight for now; normalize products to be PDFs.

But it is used in the new algorithm...

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.

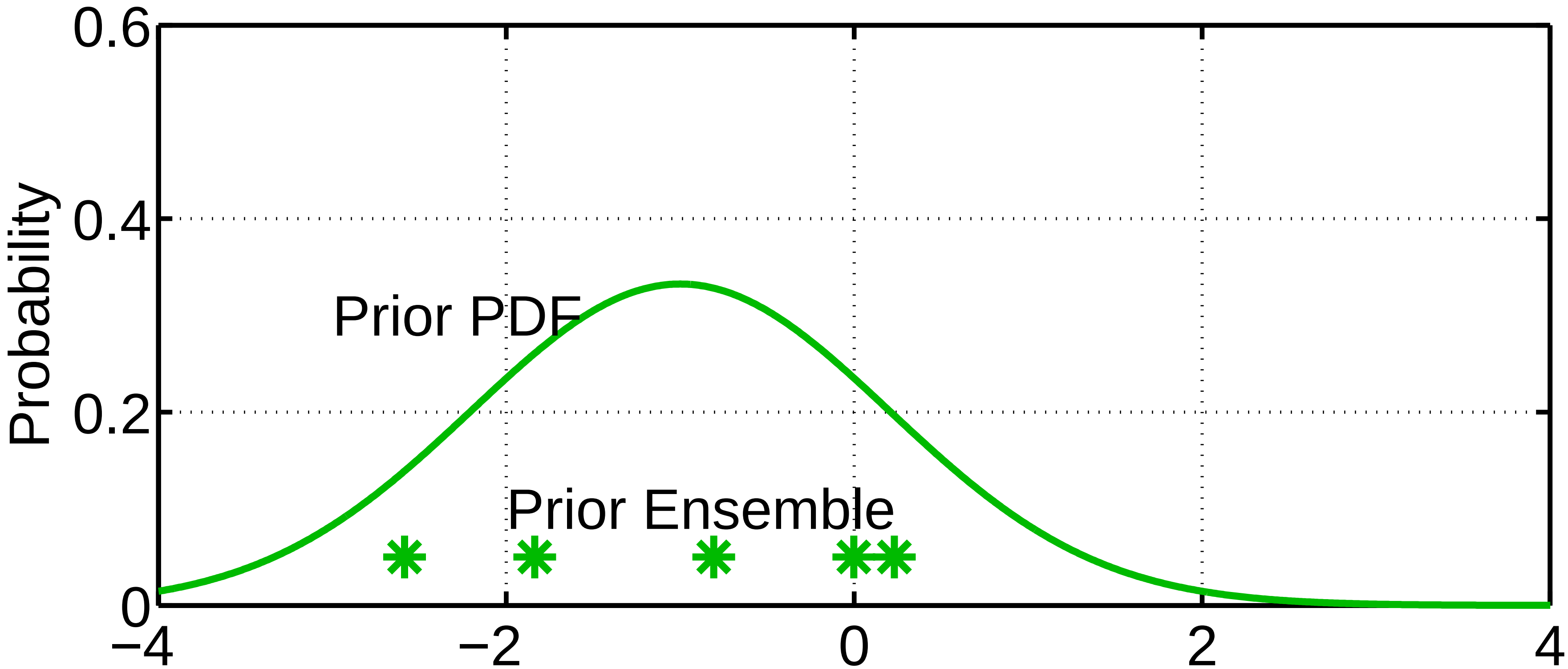


Don't know much about properties of this sample.

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

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

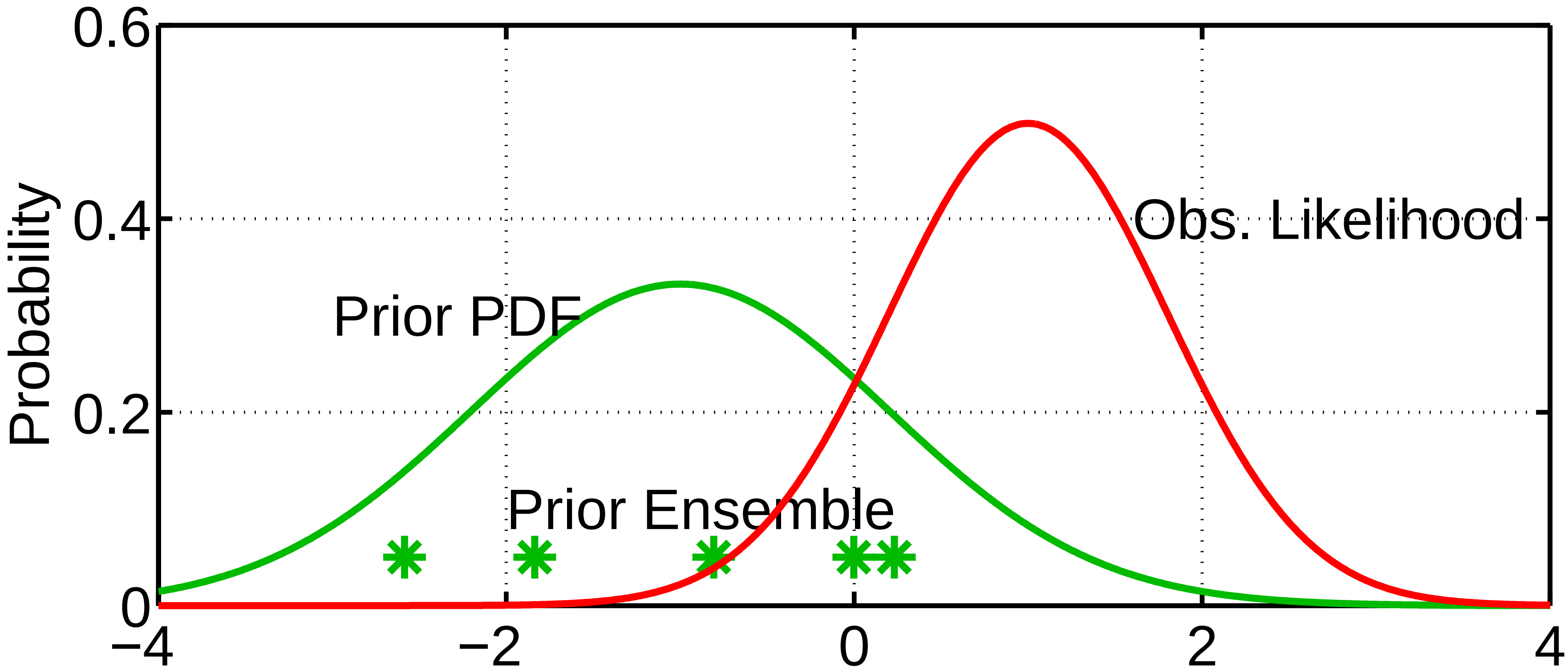
How can we take product of sample with continuous likelihood?



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

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

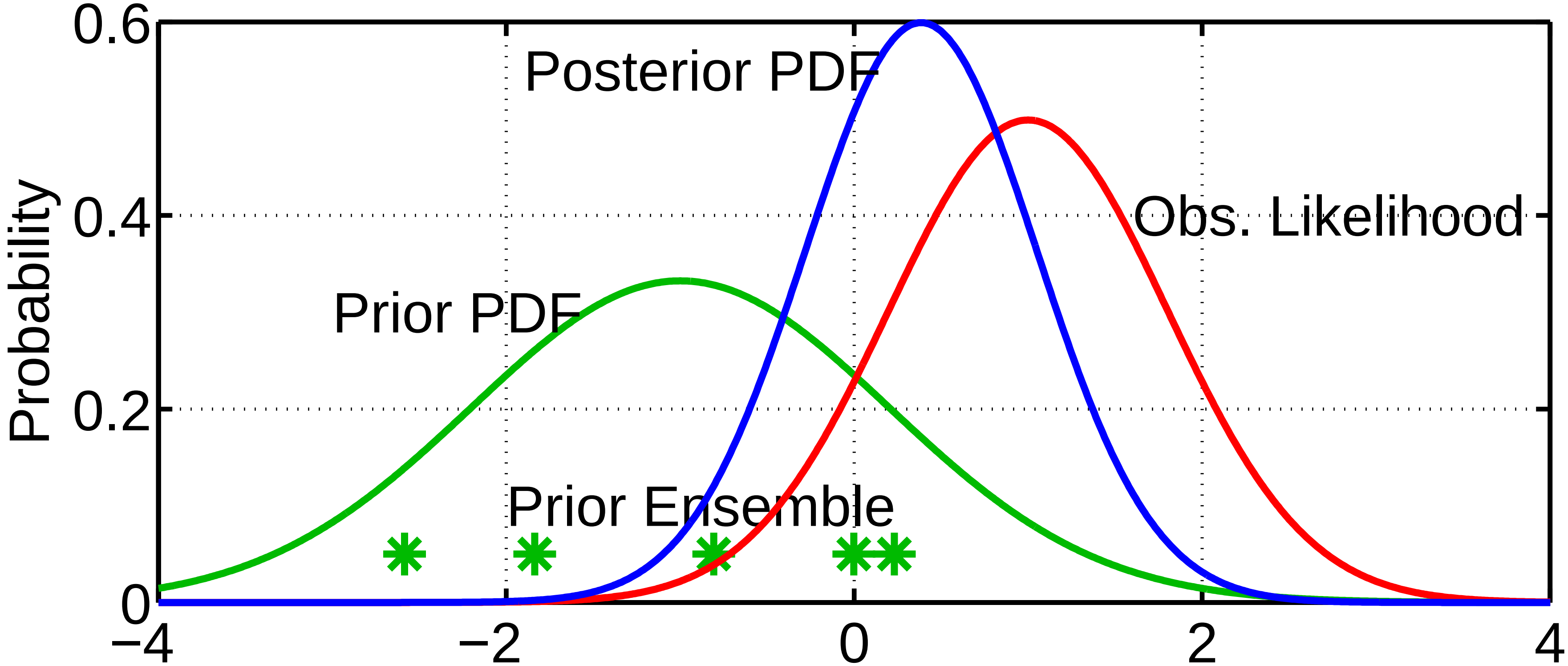
Observation likelihood usually continuous (often Gaussian).



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.

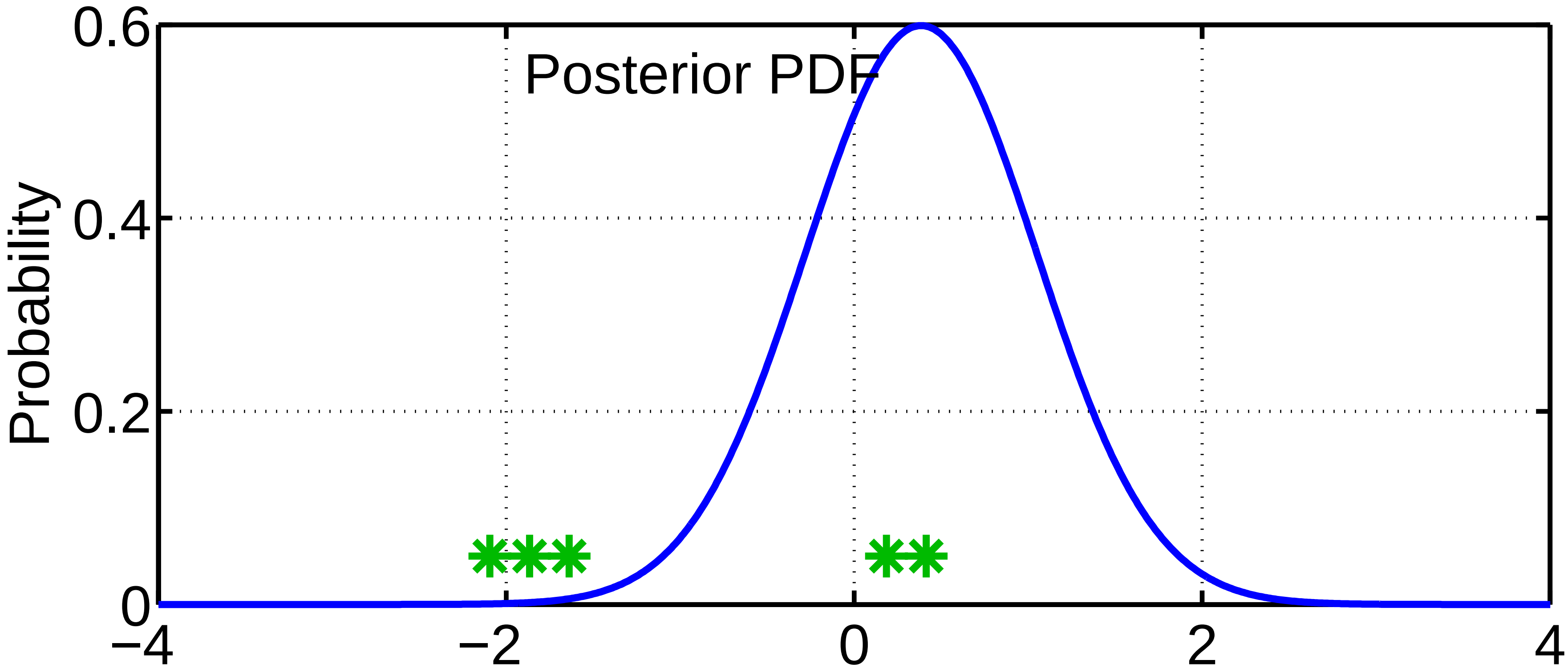


Computing continuous posterior is simple.

BUT, need to have a SAMPLE of this PDF.

Ensemble Filter Algorithms:

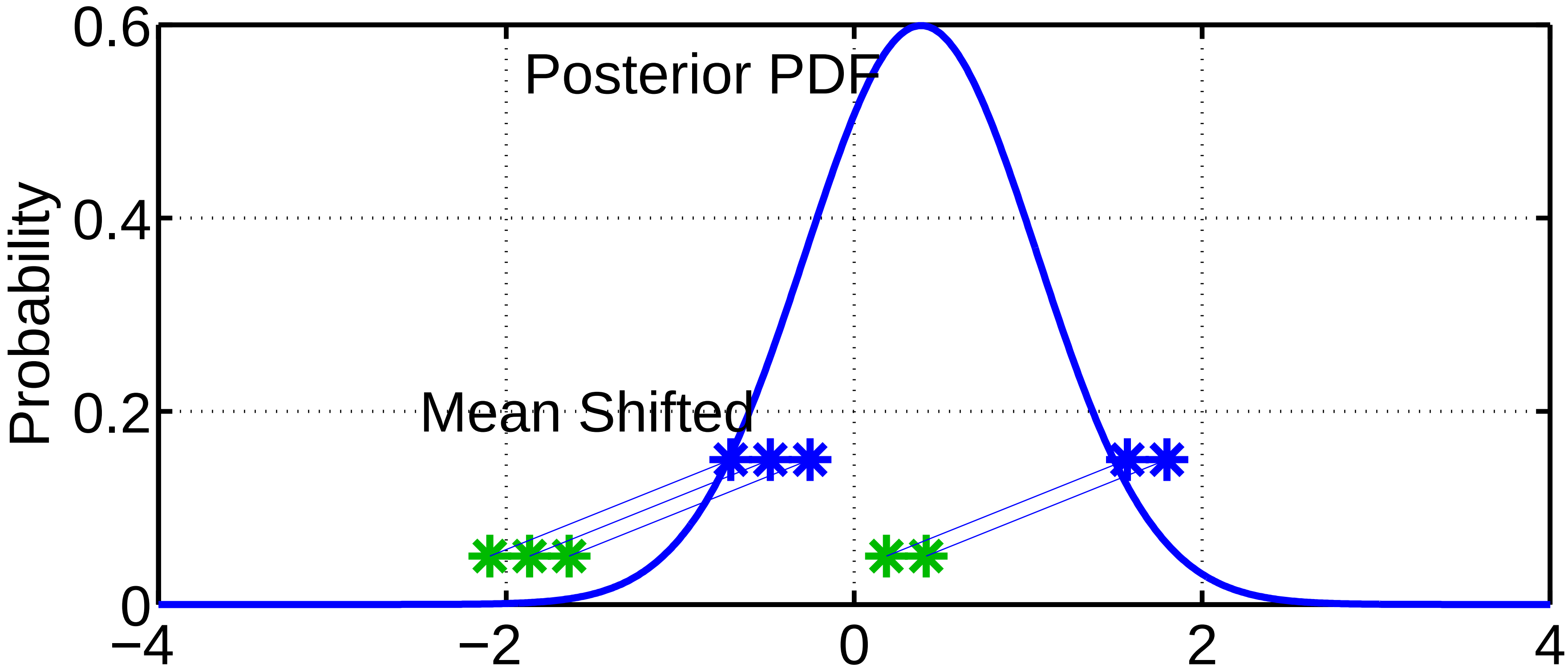
Ensemble Adjustment Filter (a deterministic square root filter).



Use deterministic algorithm to ‘adjust’ ensemble.

Ensemble Filter Algorithms:

Ensemble Adjustment Filter (a deterministic square root filter).

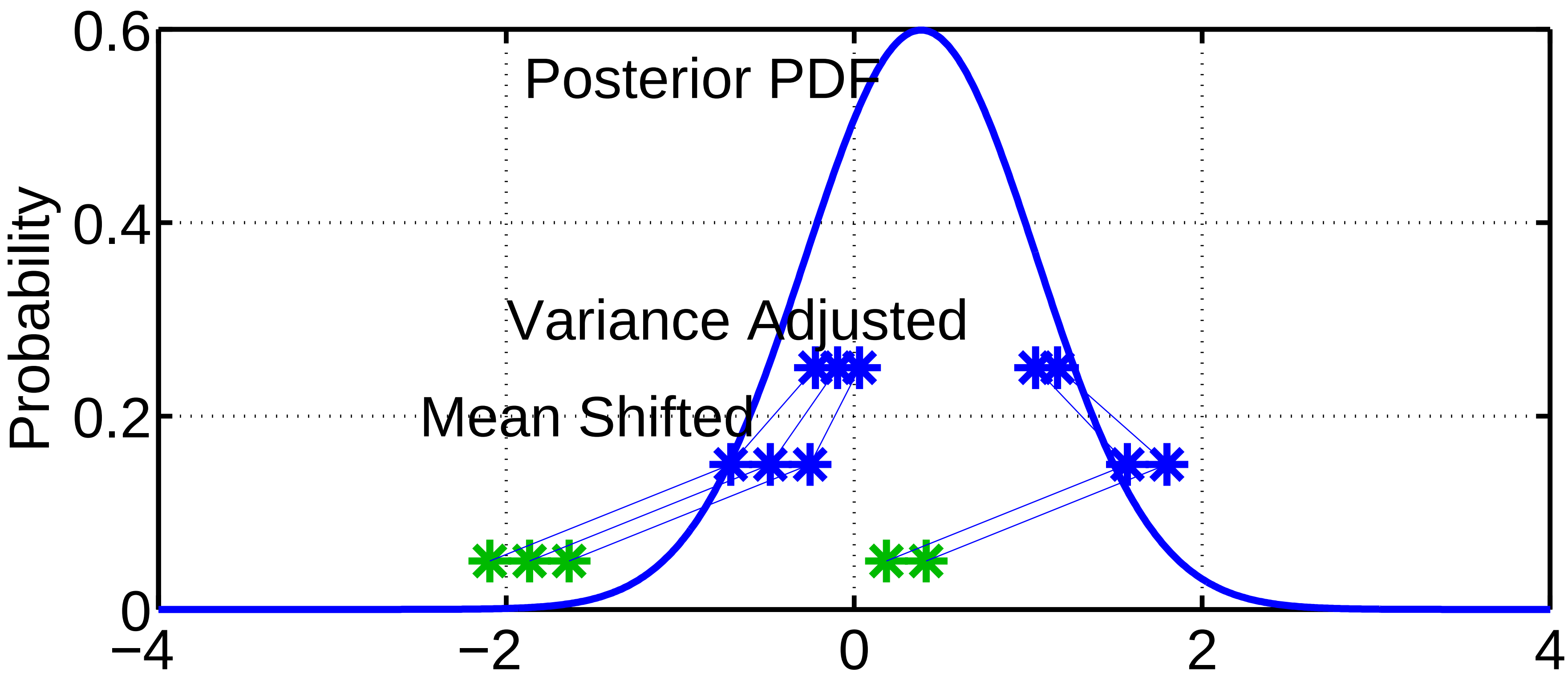


Use deterministic algorithm to 'adjust' ensemble.

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

Ensemble Filter Algorithms:

Ensemble Adjustment Filter (a deterministic square root filter).



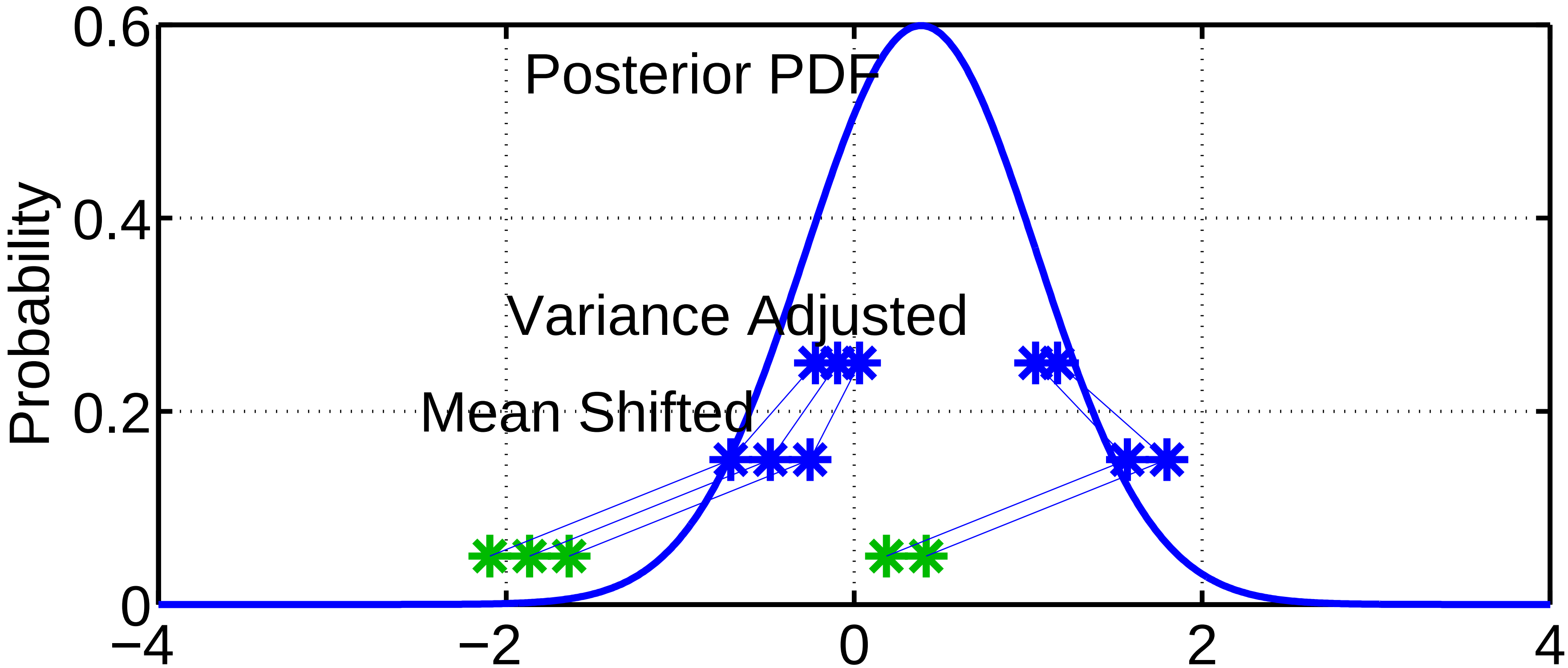
Use deterministic algorithm to ‘adjust’ ensemble.

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

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

Ensemble Filter Algorithms:

Ensemble Adjustment Filter (a deterministic square root filter).

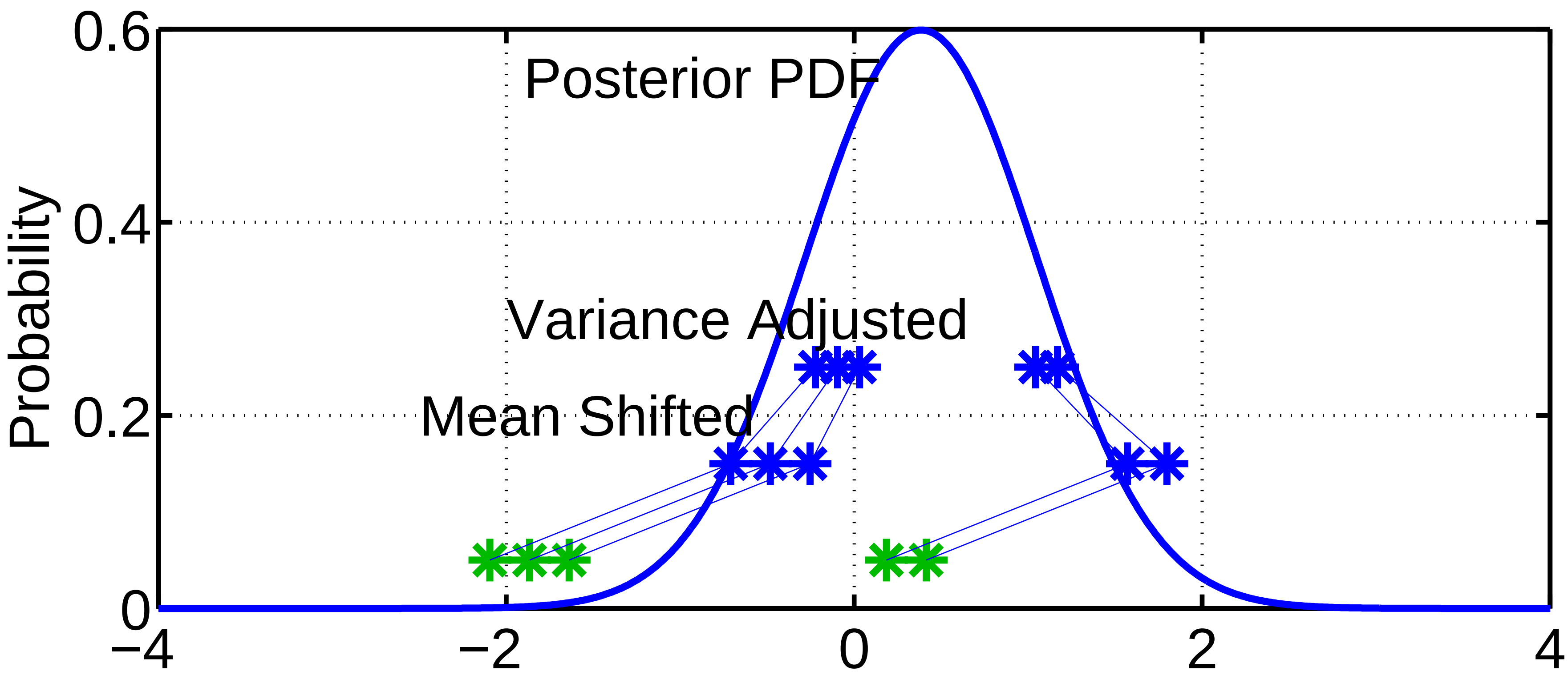


$$x_i^u = (x_i^p - \overline{x^p}) \cdot (\sigma^u / \sigma^p) + \overline{x^u} \quad \text{i = 1,..., ensemble size.}$$

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

Ensemble Filter Algorithms:

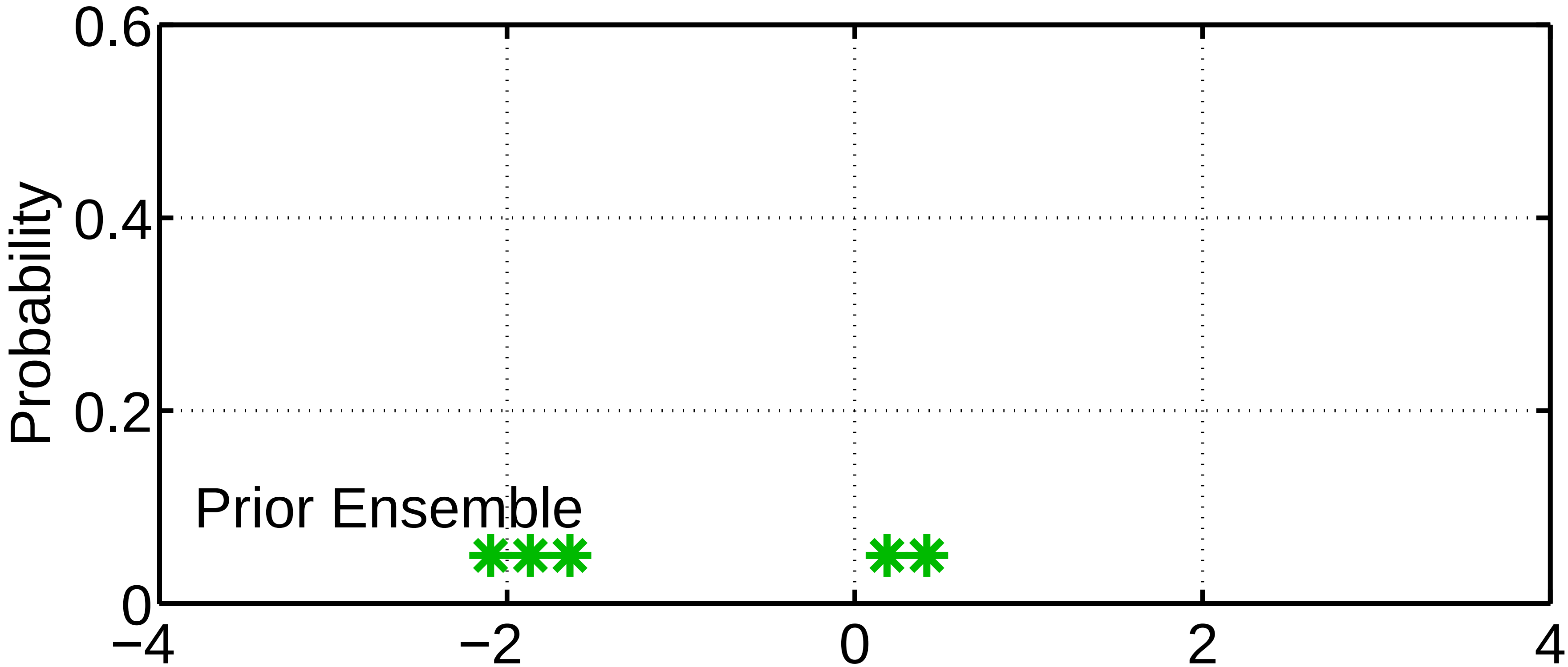
Ensemble Adjustment Filter (a deterministic square root filter).



For linear, gaussian, large enough ensemble, this is EXACTLY KF.

Ensemble Filter Algorithms:

Ensemble Kalman Filter (EnKF).

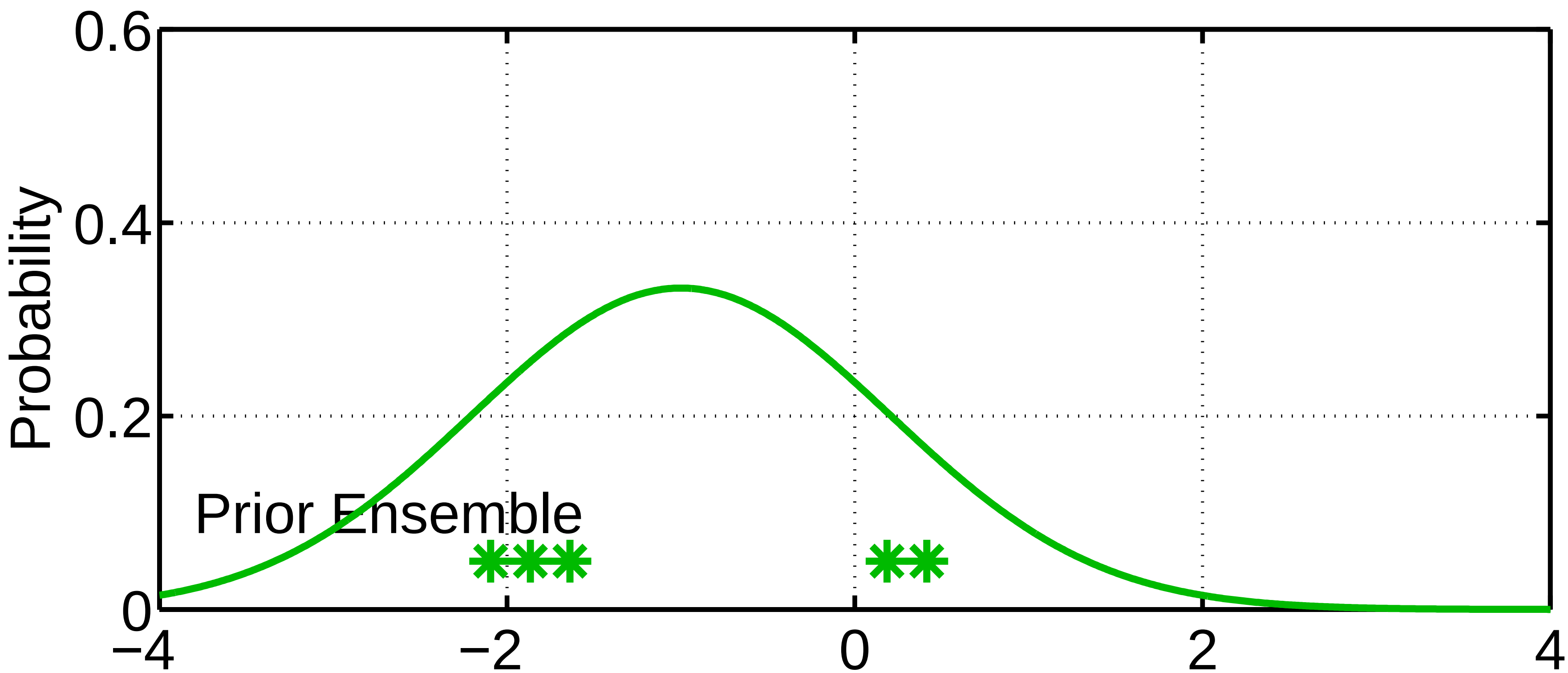


‘Classical’ Monte Carlo Algorithm for Data Assimilation.

Note: earliest refs have incorrect algorithm.

Ensemble Filter Algorithms:

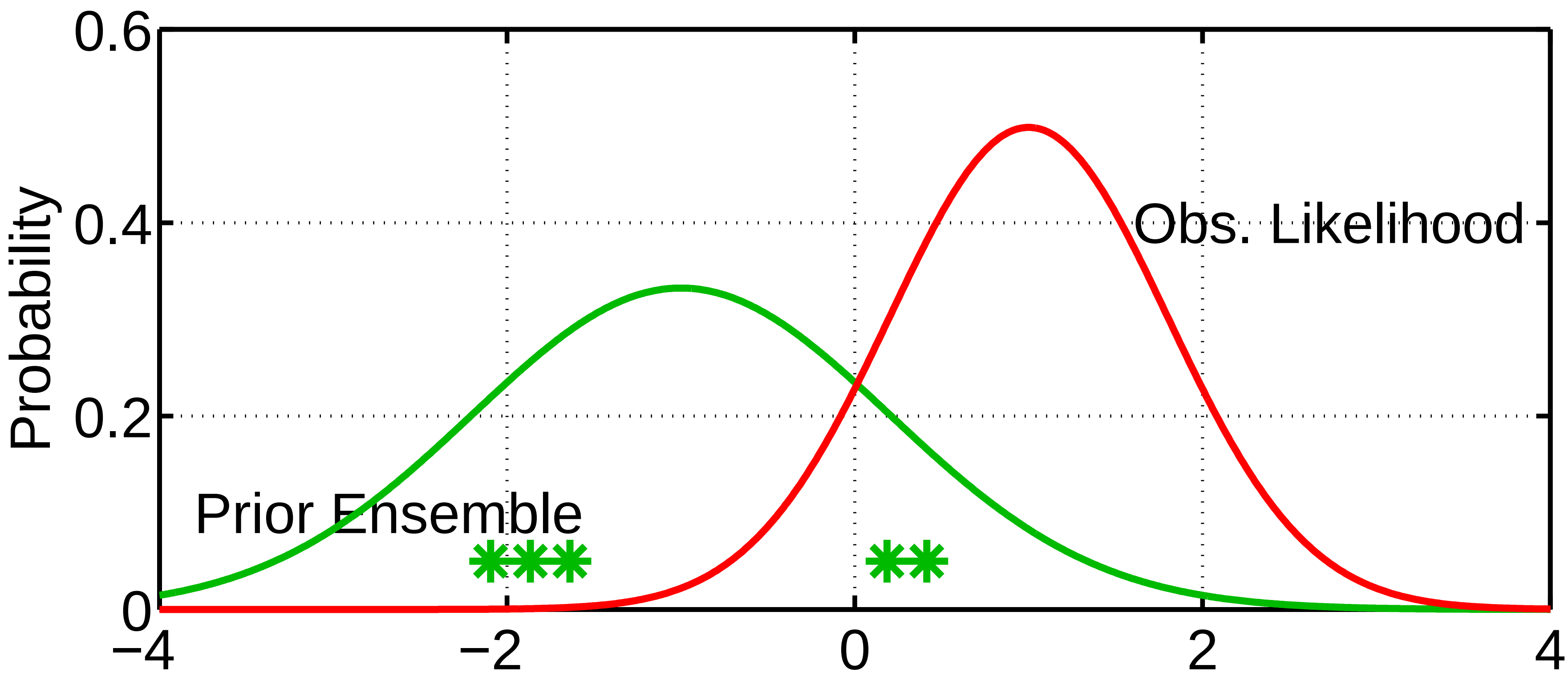
Ensemble Kalman Filter (EnKF).



Again, fit a Gaussian to sample.

Ensemble Filter Algorithms:

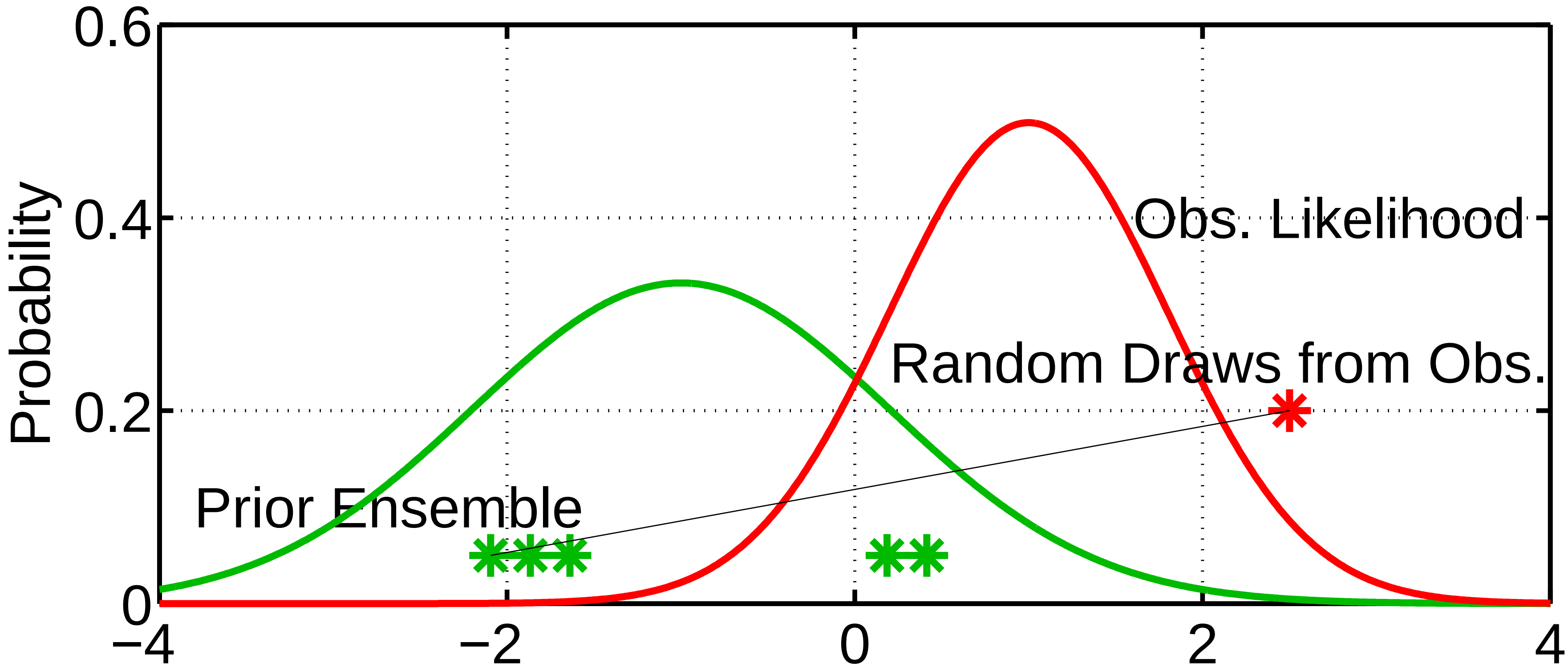
Ensemble Kalman Filter (EnKF).



Again, fit a Gaussian to sample.

Ensemble Filter Algorithms:

Ensemble Kalman Filter (EnKF).

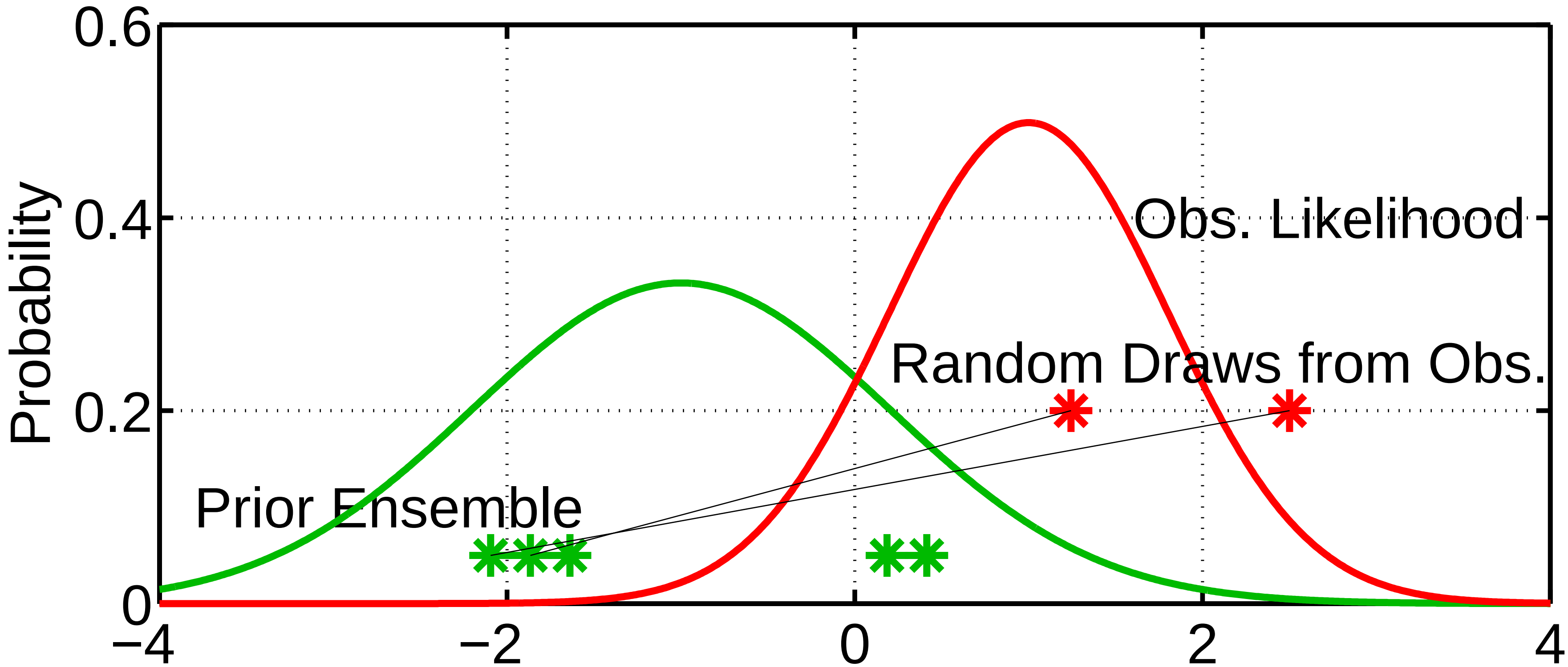


Generate a random draw from the obs. likelihood.

Associate it with the first sample of prior ensemble.

Ensemble Filter Algorithms:

Ensemble Kalman Filter (EnKF).

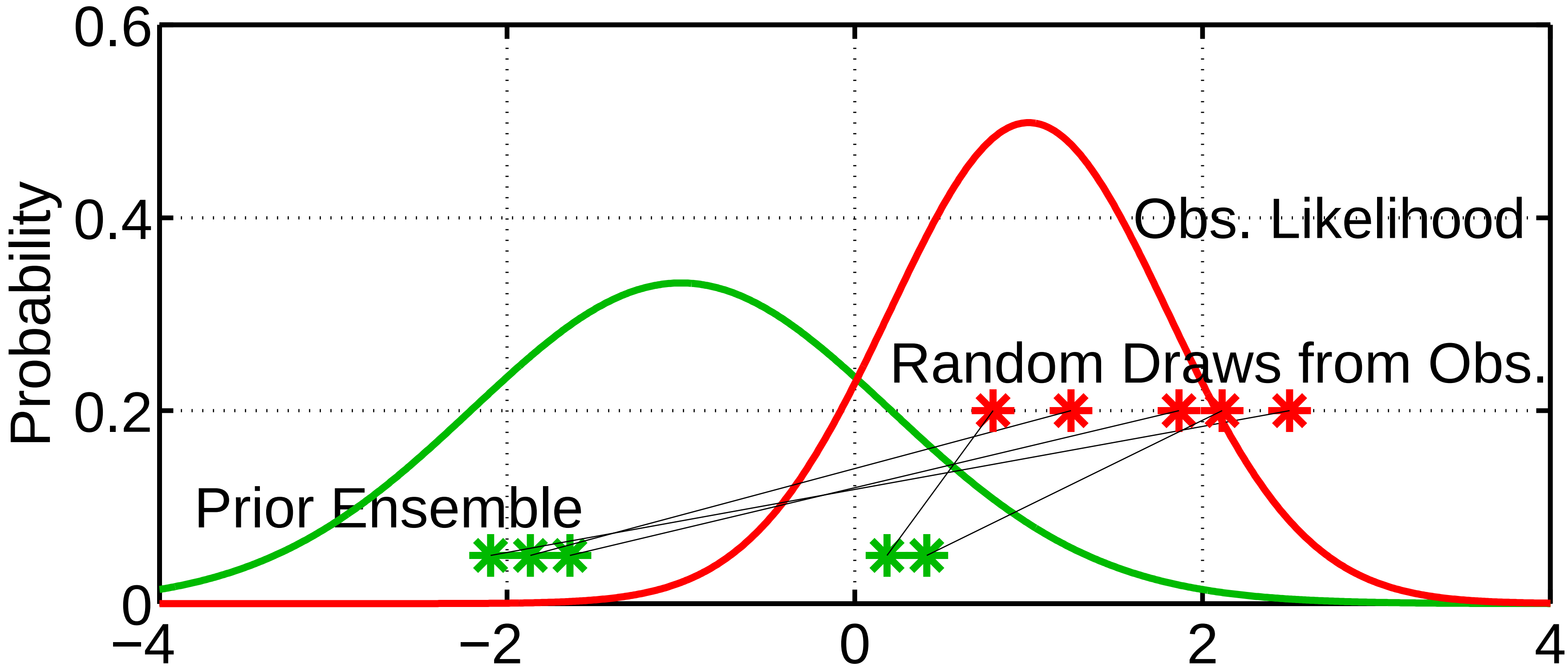


Proceed to associate a random draw from obs. with each prior sample.

Algorithm sometimes called ‘perturbed obs.’ ensemble Kalman filter.

Ensemble Filter Algorithms:

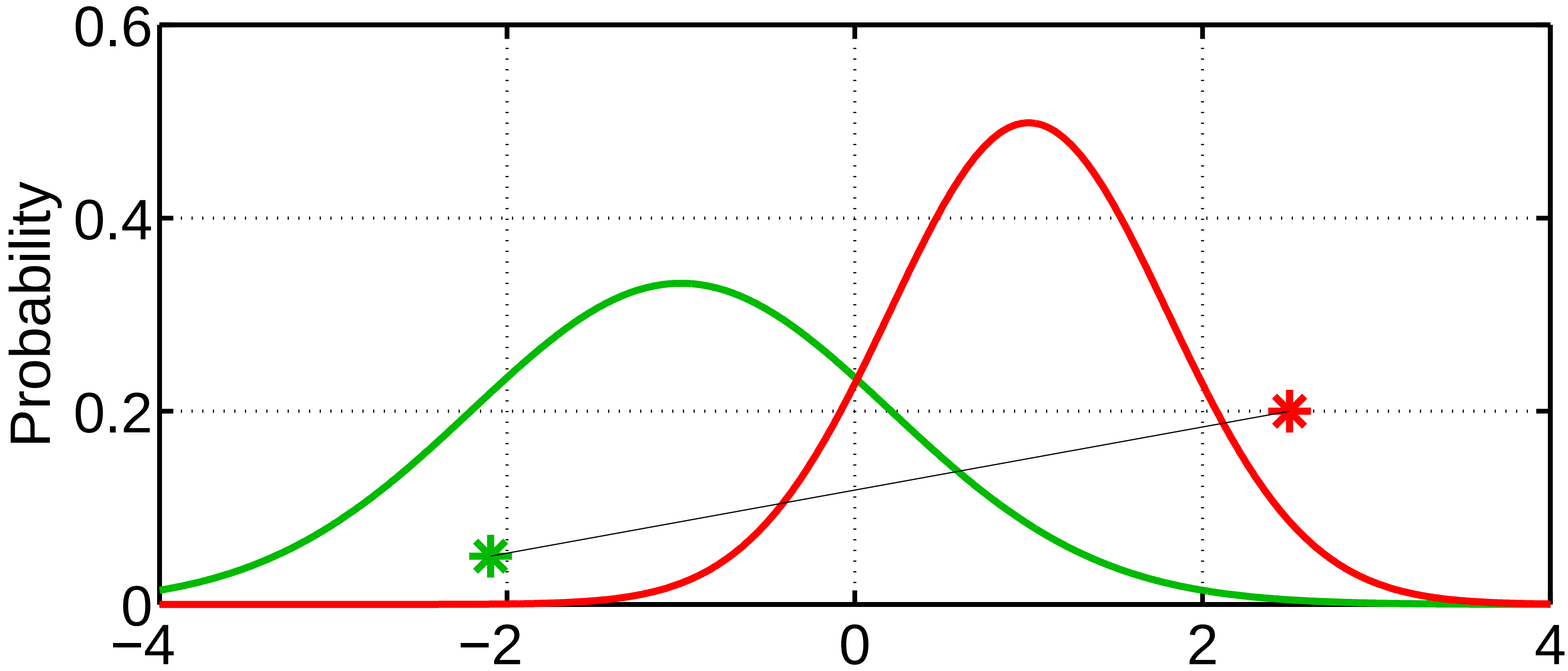
Ensemble Kalman Filter (EnKF).



Adjusting the mean of obs. sample to be exact improves performance.
Adjusting the variance may further improve performance.

Ensemble Filter Algorithms:

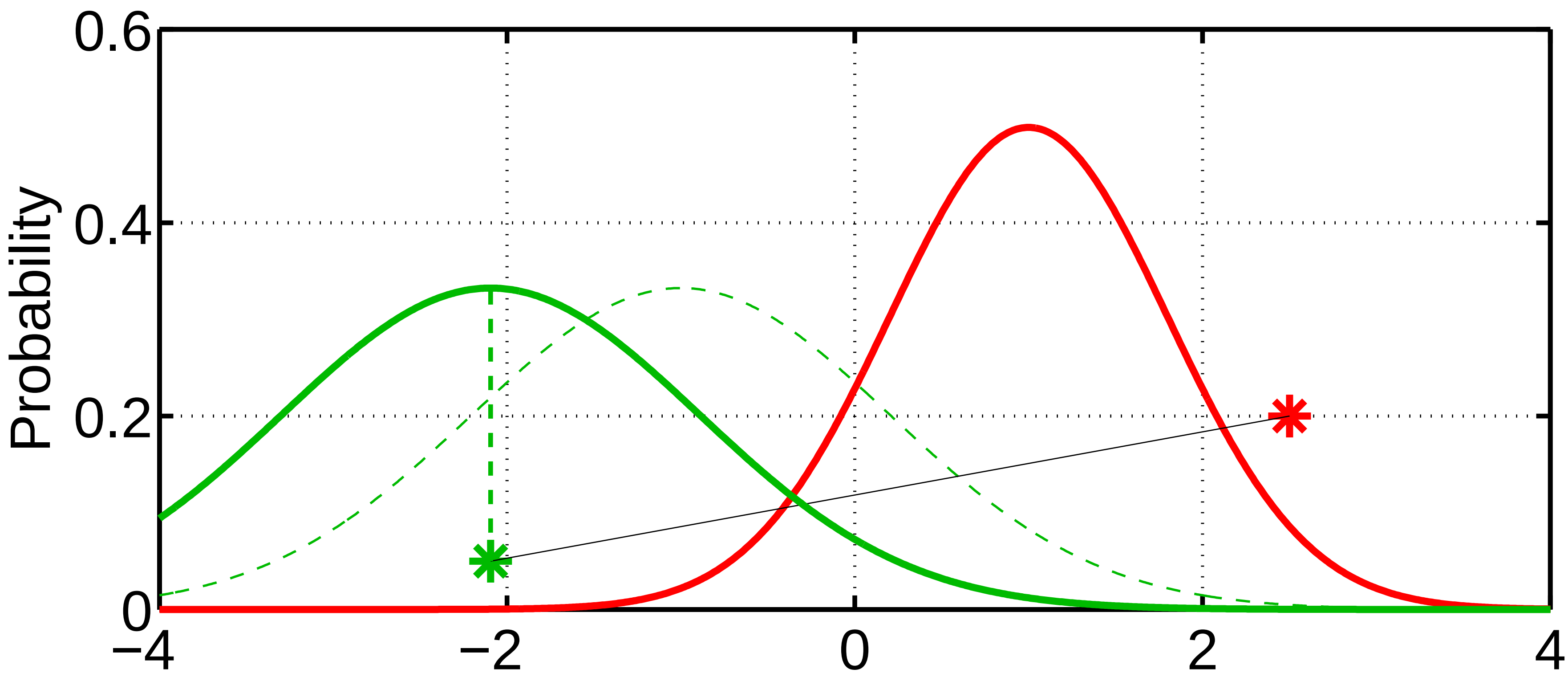
Ensemble Kalman Filter (EnKF).



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

Ensemble Filter Algorithms:

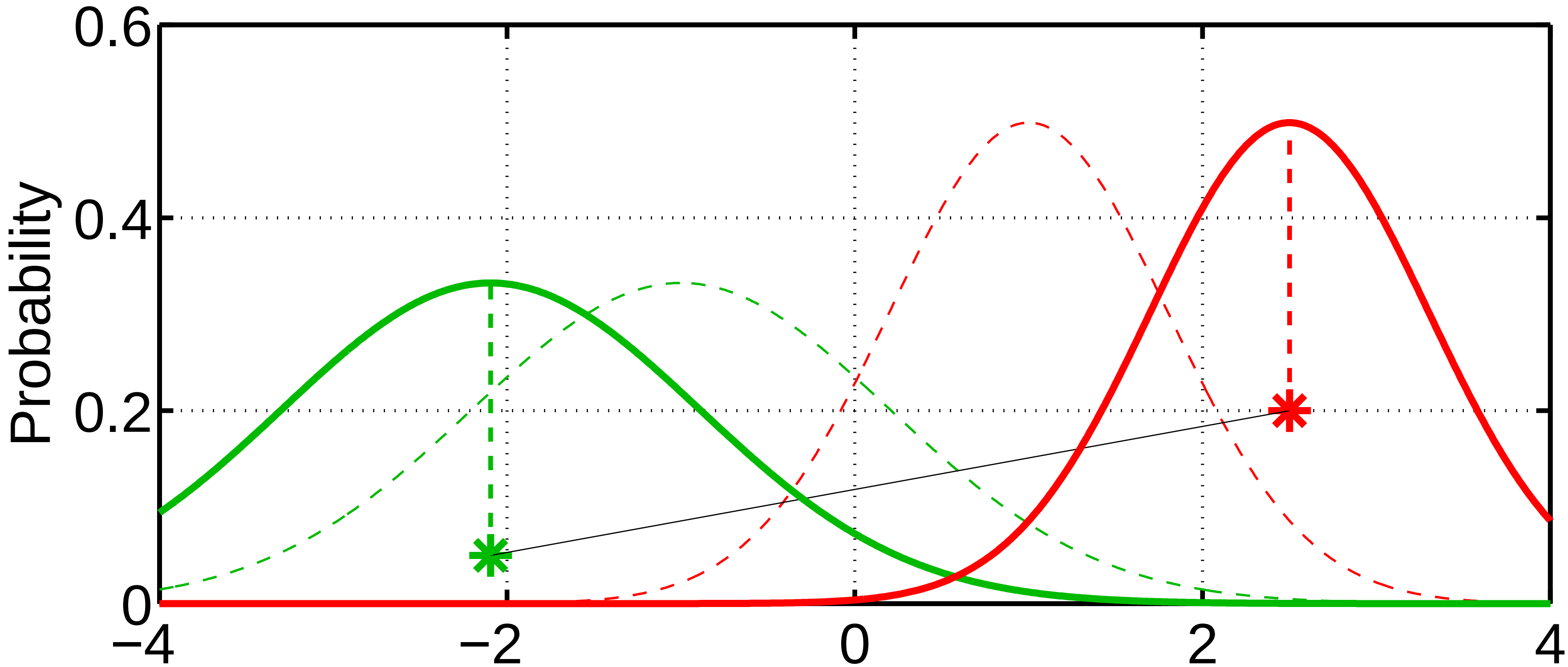
Ensemble Kalman Filter (EnKF).



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

Ensemble Filter Algorithms:

Ensemble Kalman Filter (EnKF).

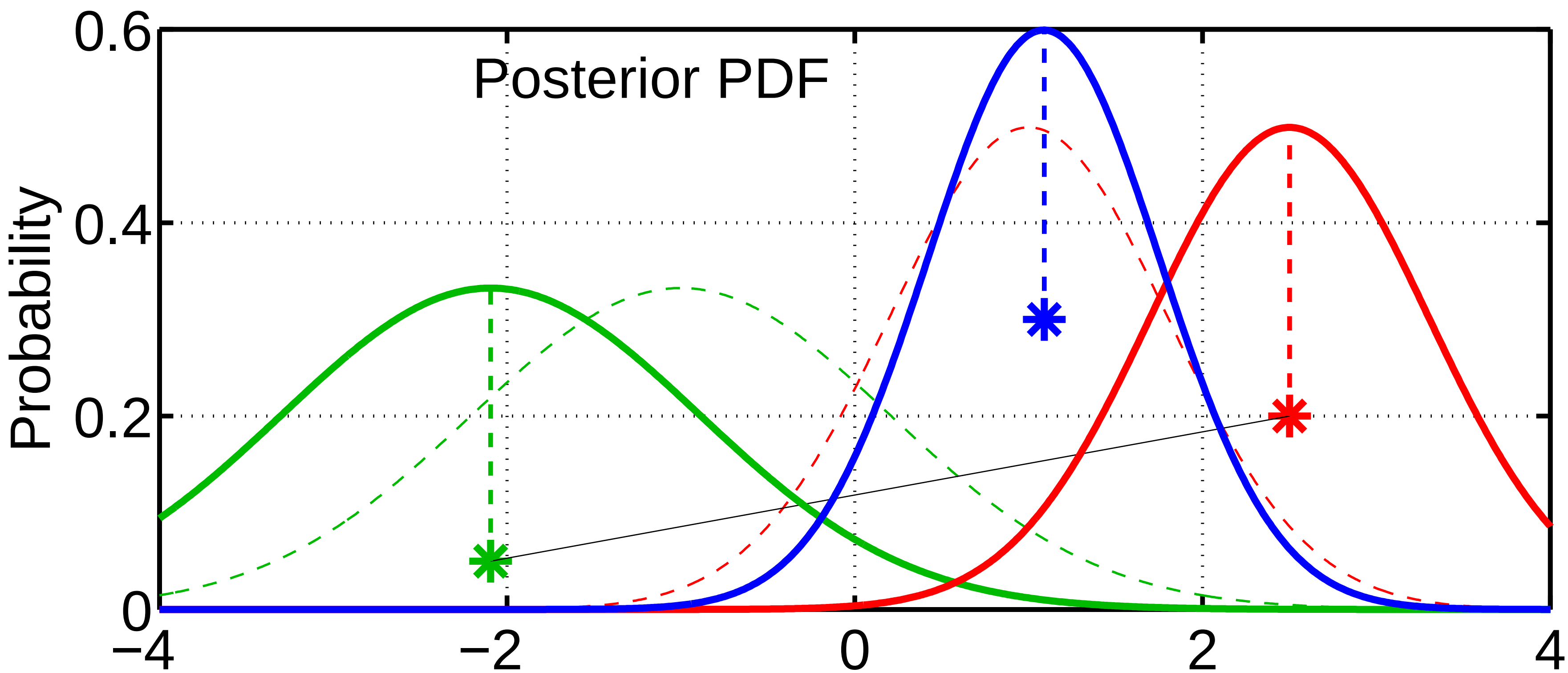


Prior sample standard deviation still measures uncertainty of prior mean estimate.

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

Ensemble Filter Algorithms:

Ensemble Kalman Filter (EnKF).

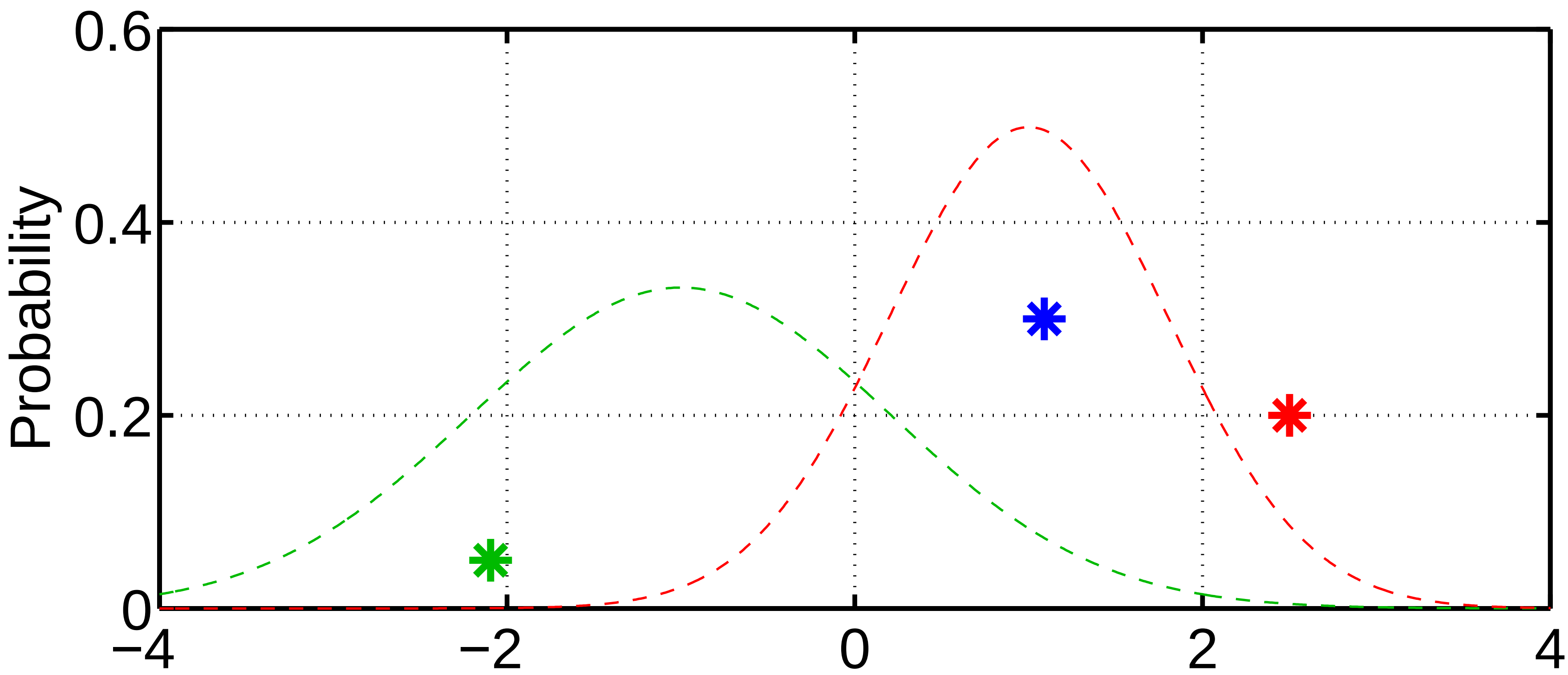


Take product of the prior/obs distributions for first sample.

This is standard Gaussian product.

Ensemble Filter Algorithms:

Ensemble Kalman Filter (EnKF).

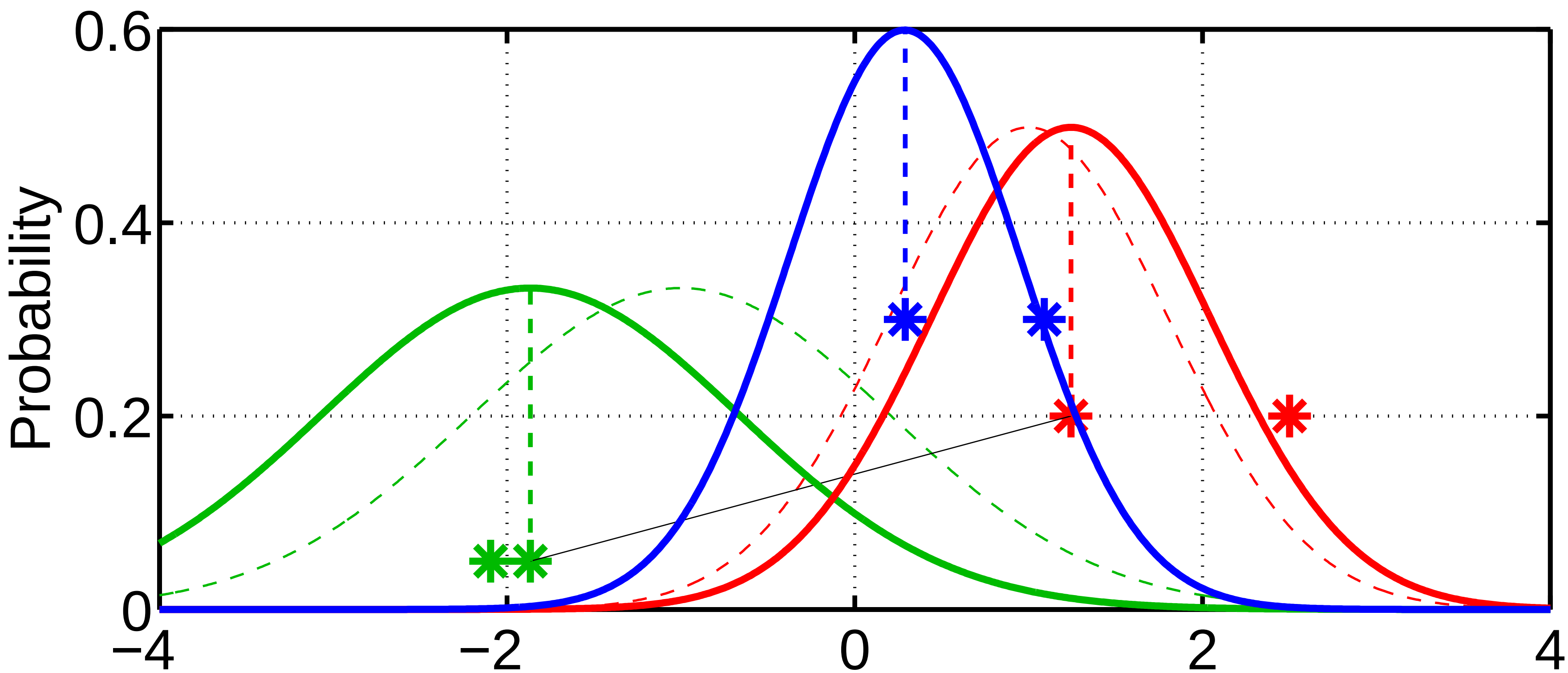


Mean of product is random sample of posterior.

Product of random samples is random sample of product.

Ensemble Filter Algorithms:

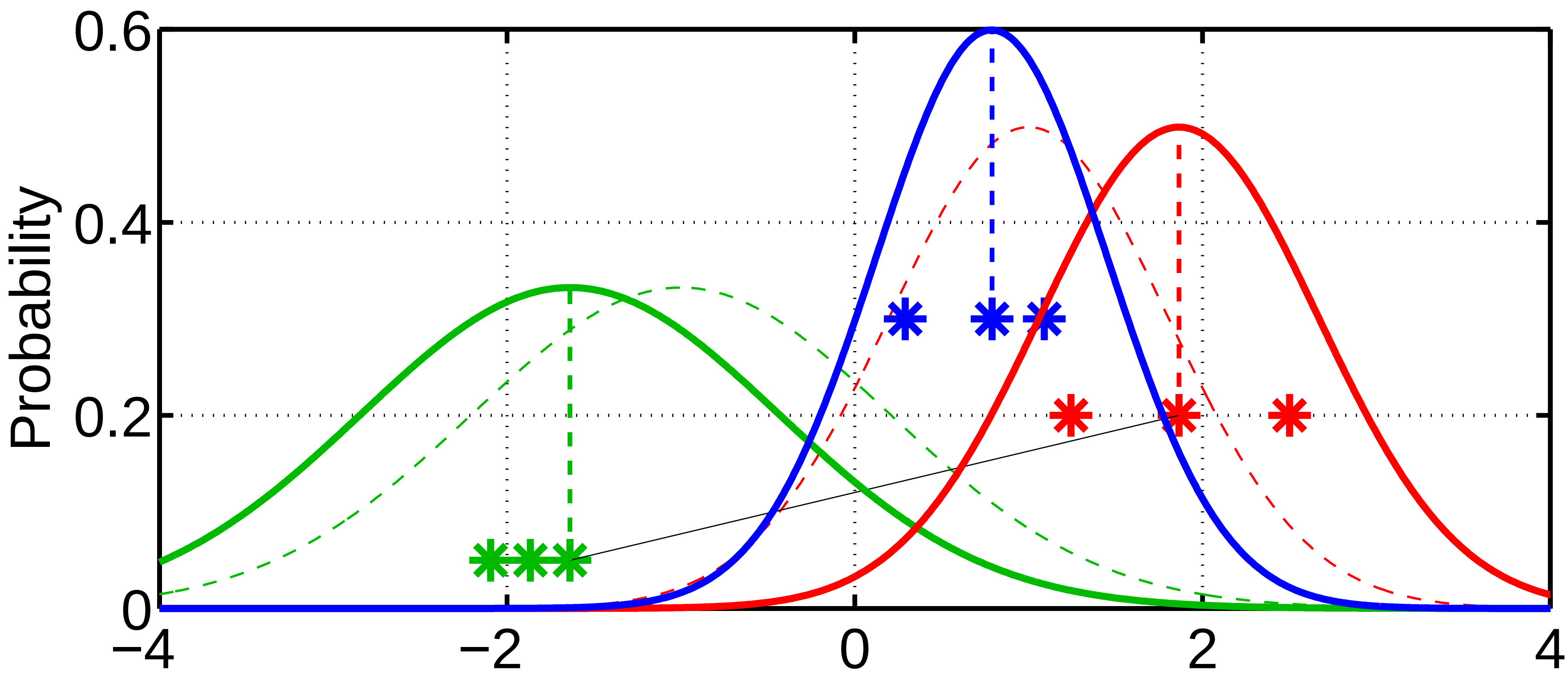
Ensemble Kalman Filter (EnKF).



Repeat this operation for each joint prior pair.

Ensemble Filter Algorithms:

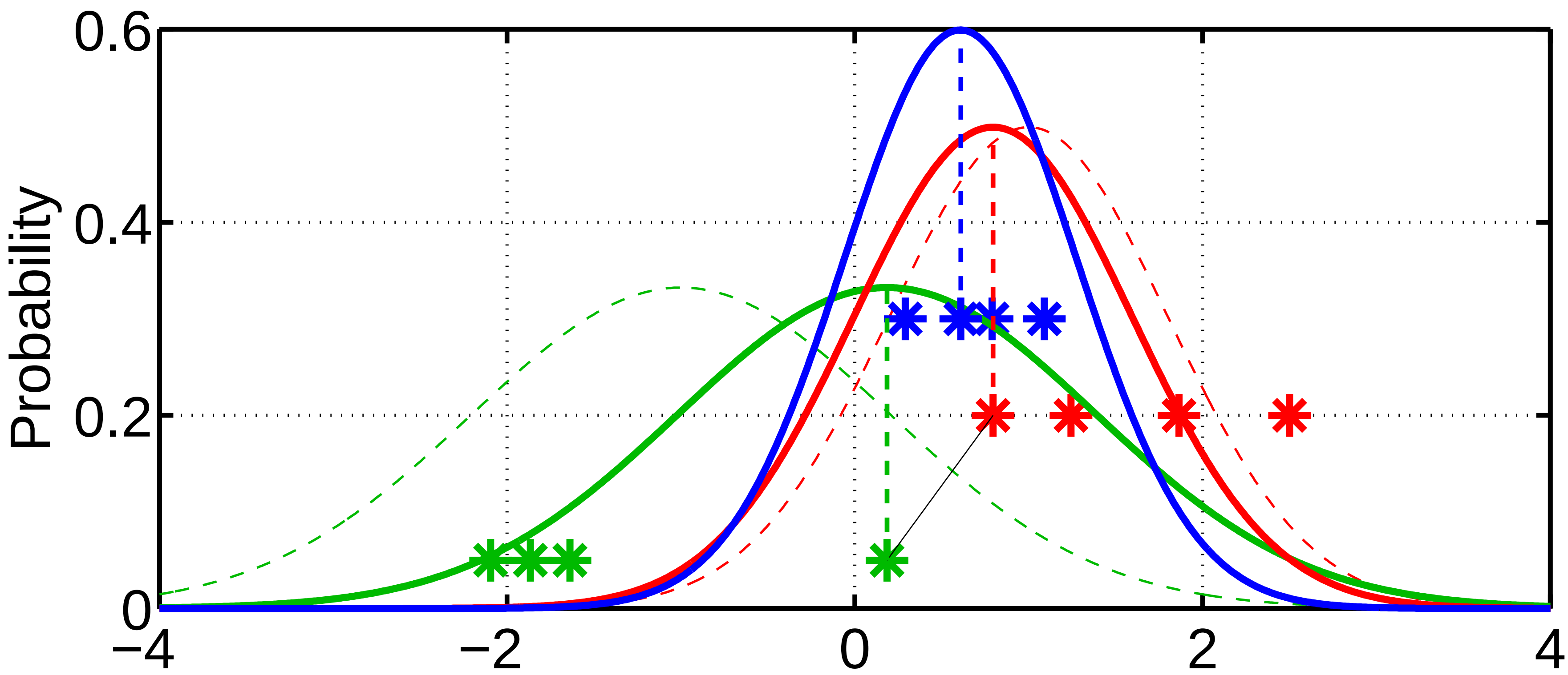
Ensemble Kalman Filter (EnKF).



Repeat this operation for each joint prior pair.

Ensemble Filter Algorithms:

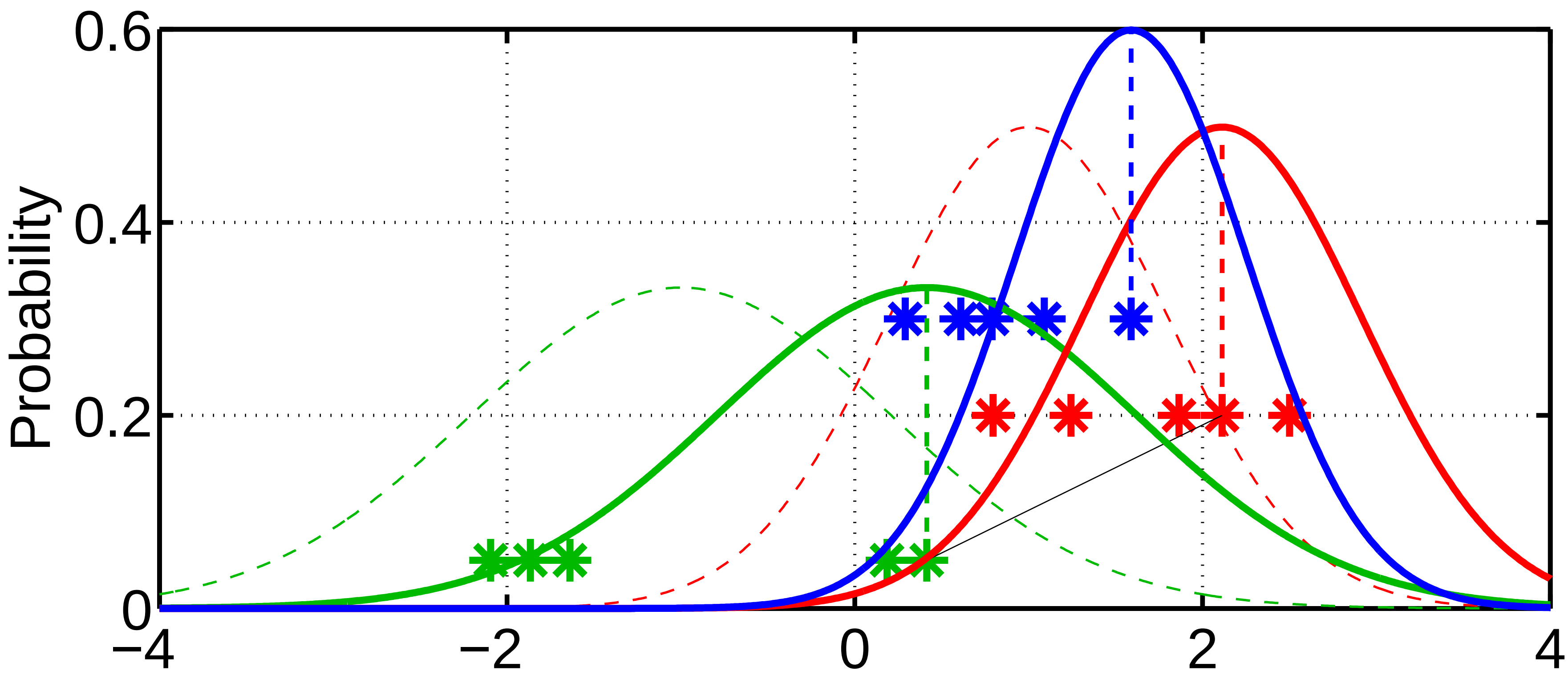
Ensemble Kalman Filter (EnKF)



Repeat this operation for each joint prior pair.

Ensemble Filter Algorithms:

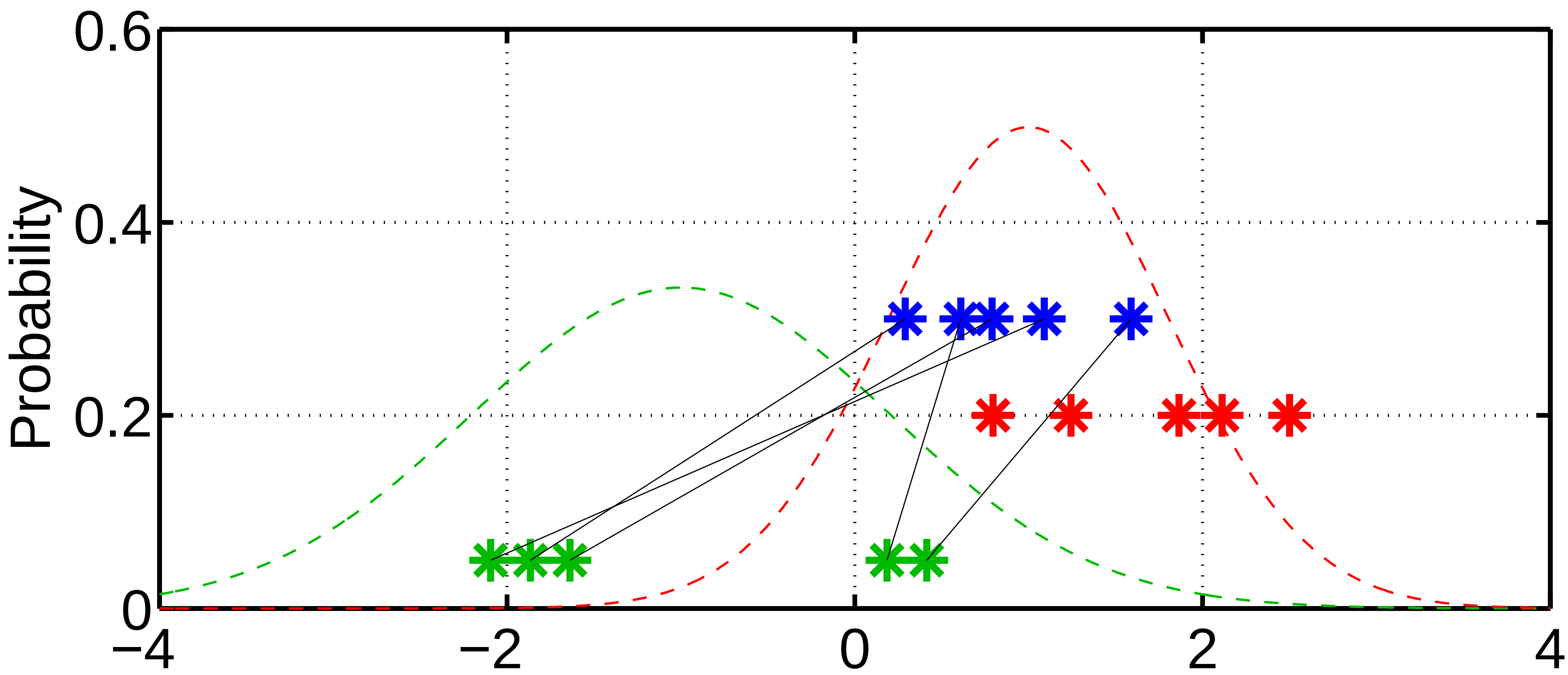
Ensemble Kalman Filter (EnKF).



Repeat this operation for each joint prior pair.

Ensemble Filter Algorithms:

Ensemble Kalman Filter (EnKF).



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

Sample is mixed by some introduced noise.

A One-Variable Test Model

$$x_{t+1} = x_t + \Delta t(x_t + \alpha|x_t|x_t), \qquad \Delta t = 0.05$$

Assume ‘true’ trajectory is just x=0.
(Same as linearizing around an arbitrary trajectory).

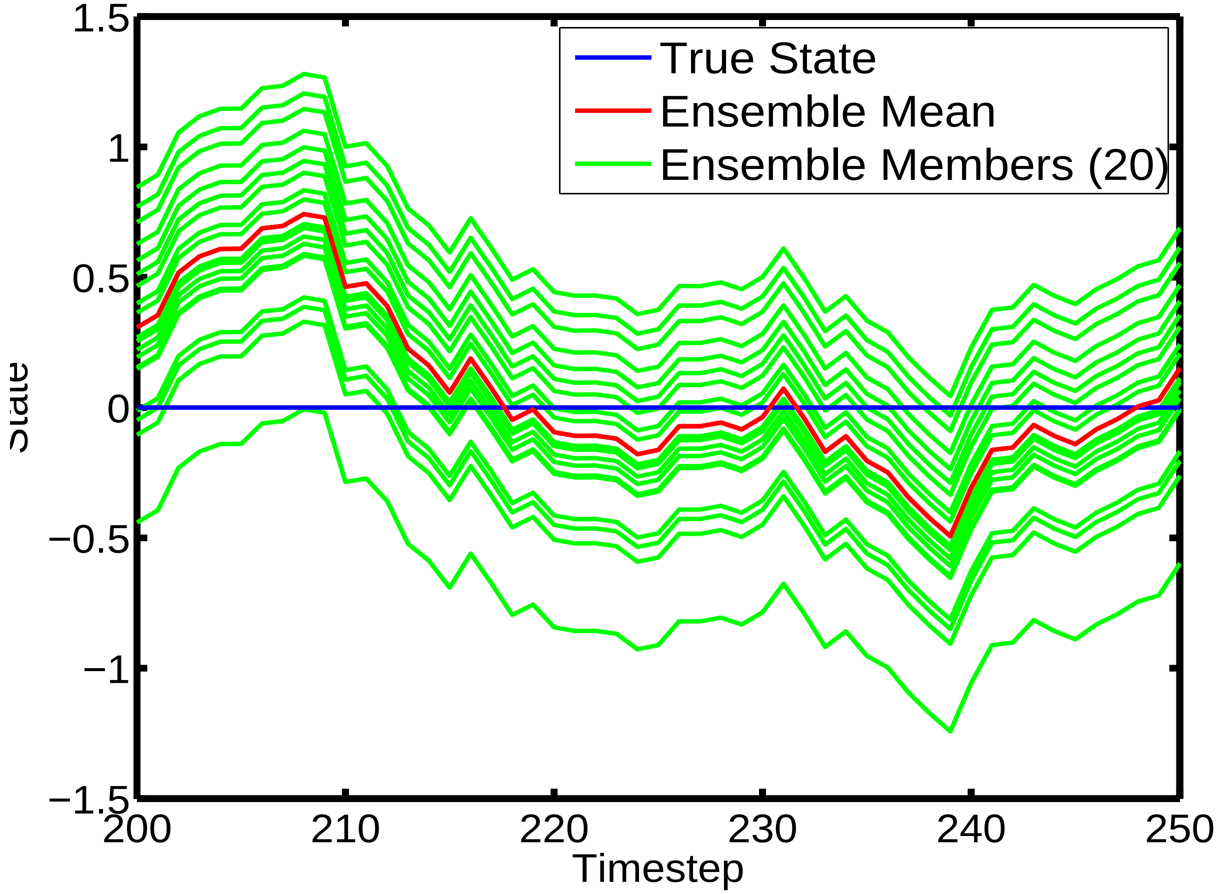
$\alpha = 0$: linear model (exponential growth).

$\alpha > 0$: have additional expansion.

Observe ‘truth’ with observational error variance 1.0 every step.
Observations are just draws from N(0, 1.0).

Linear Model Results ($\alpha = 0$): EAKF

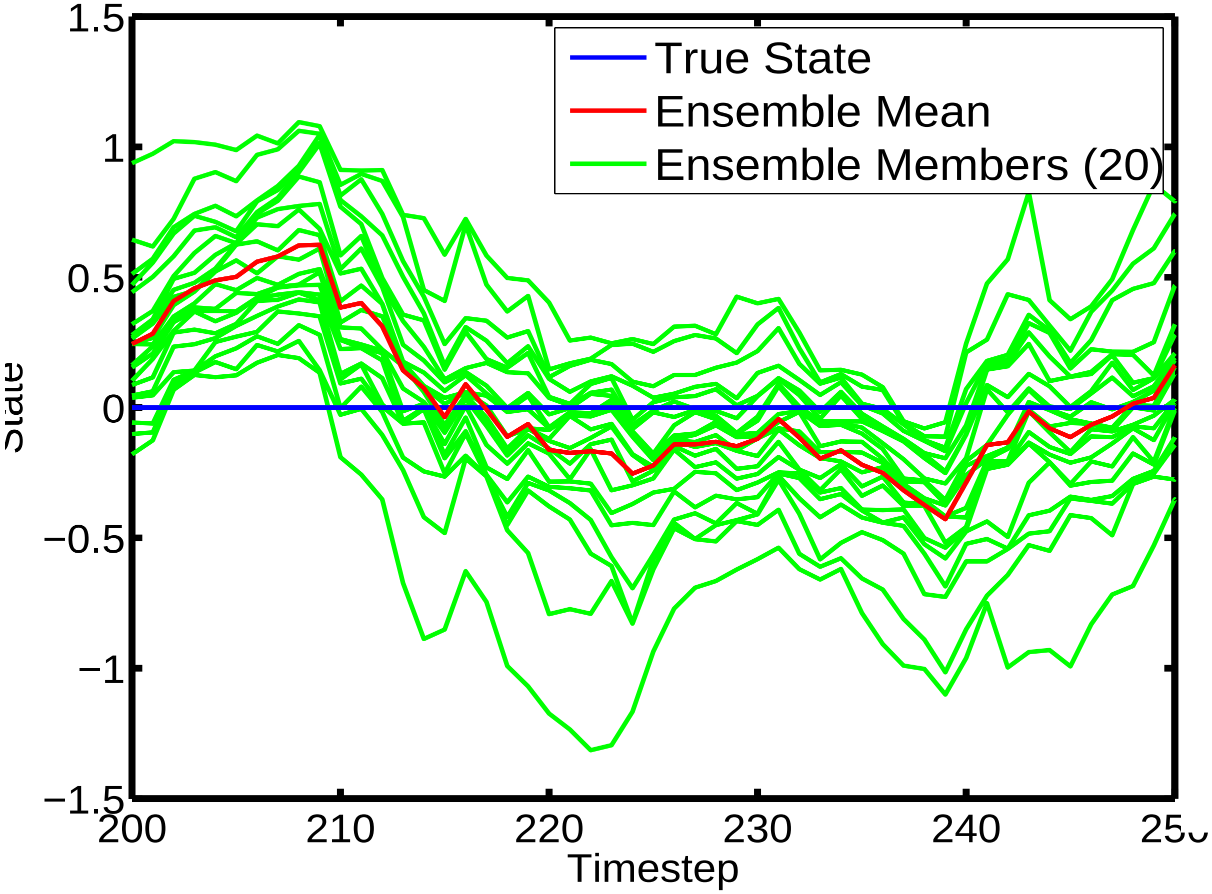
(All results throughout are for prior estimates)

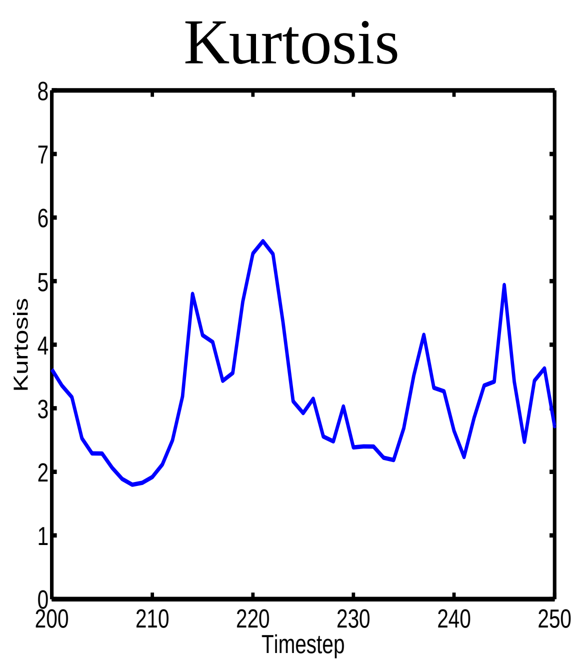


EAKF is just an algorithm for computing Kalman Filter.

Ensemble members don't cross, identical spacing when converged.

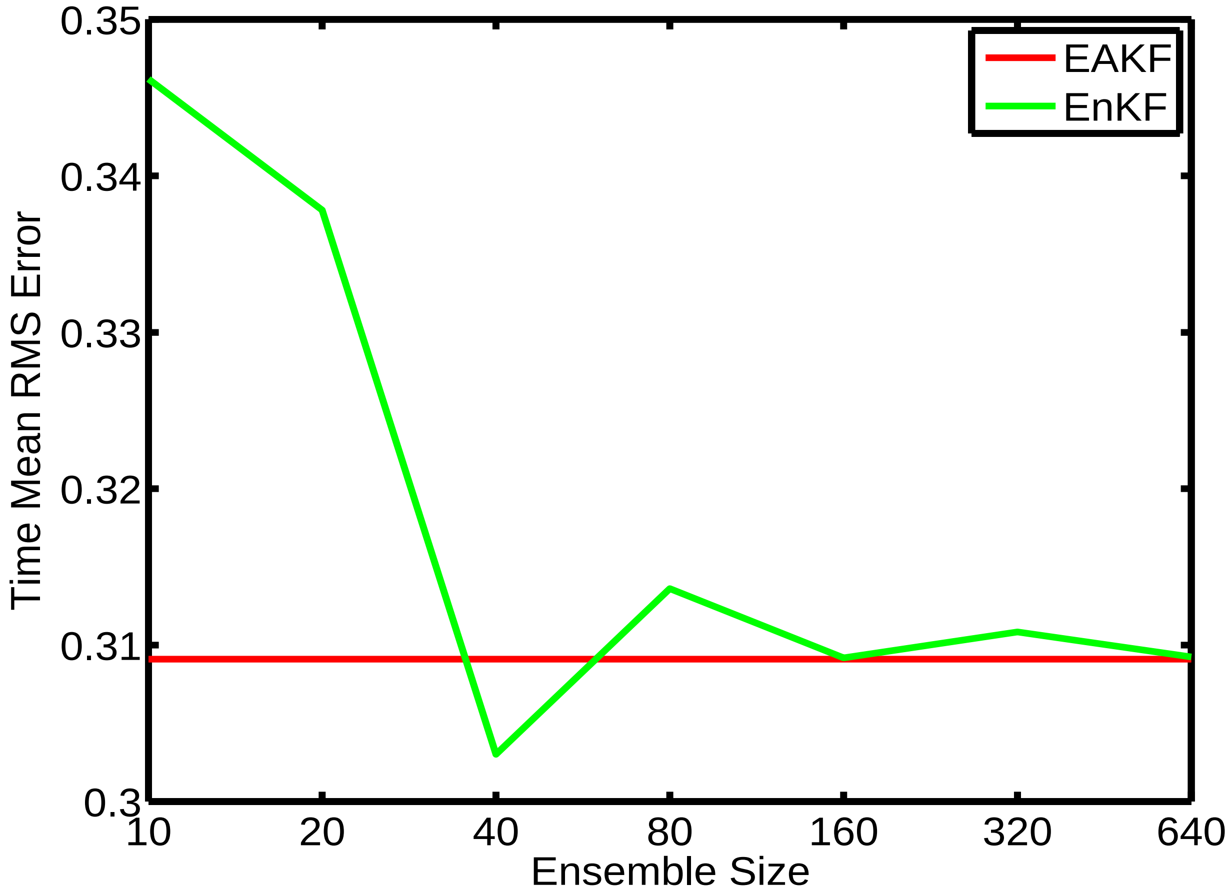
Linear Model Results (α = 0): EnKF





EnKF is a Monte Carlo algorithm approximating Kalman Filter.
Ensemble members cross, moments (like kurtosis) vary with time.
Sampling error due to small ensembles is an issue.

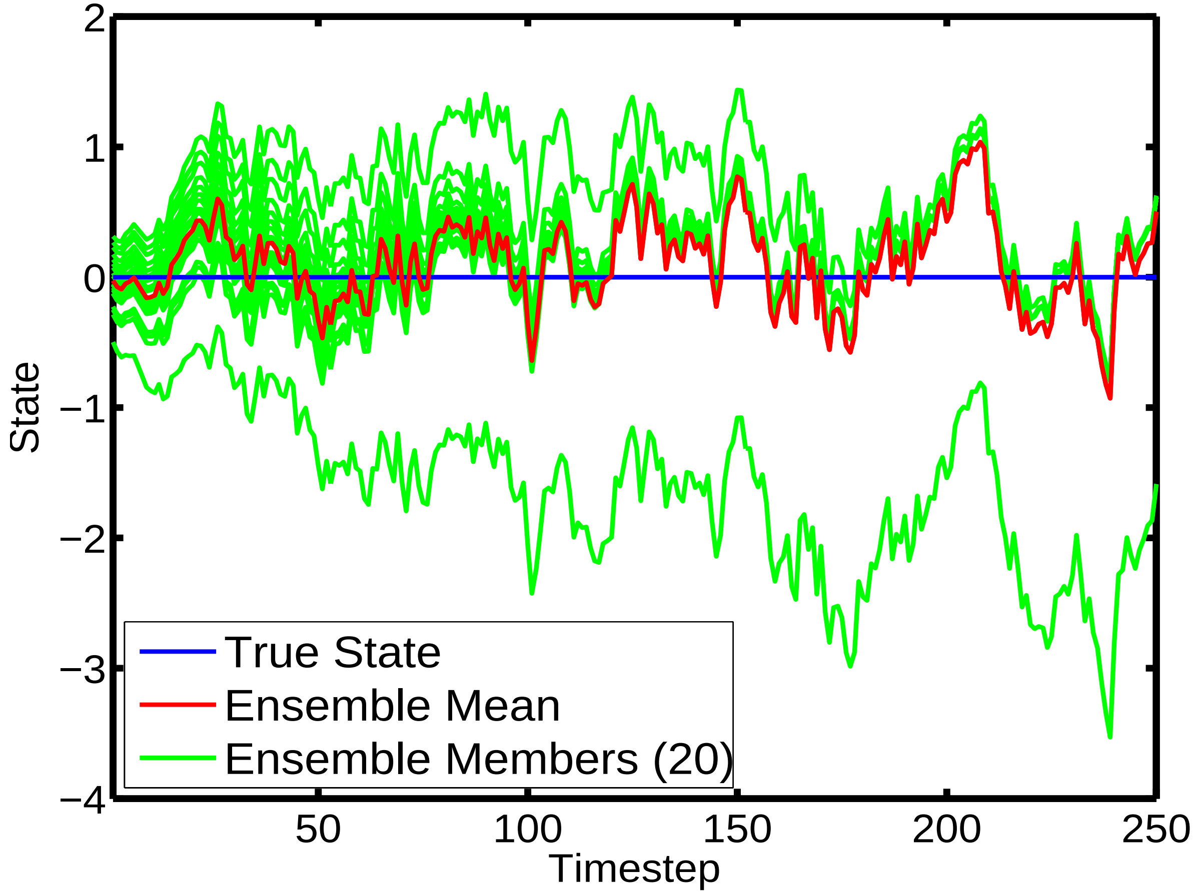
Linear Model Time-Mean RMS Error as Function of Ensemble Size



EAKF exact for any ensemble size (>1).

EnKF has sampling error (smaller value at 40 is an ‘accident’).

Nonlinear Model Results (α = 0.8): EAKF

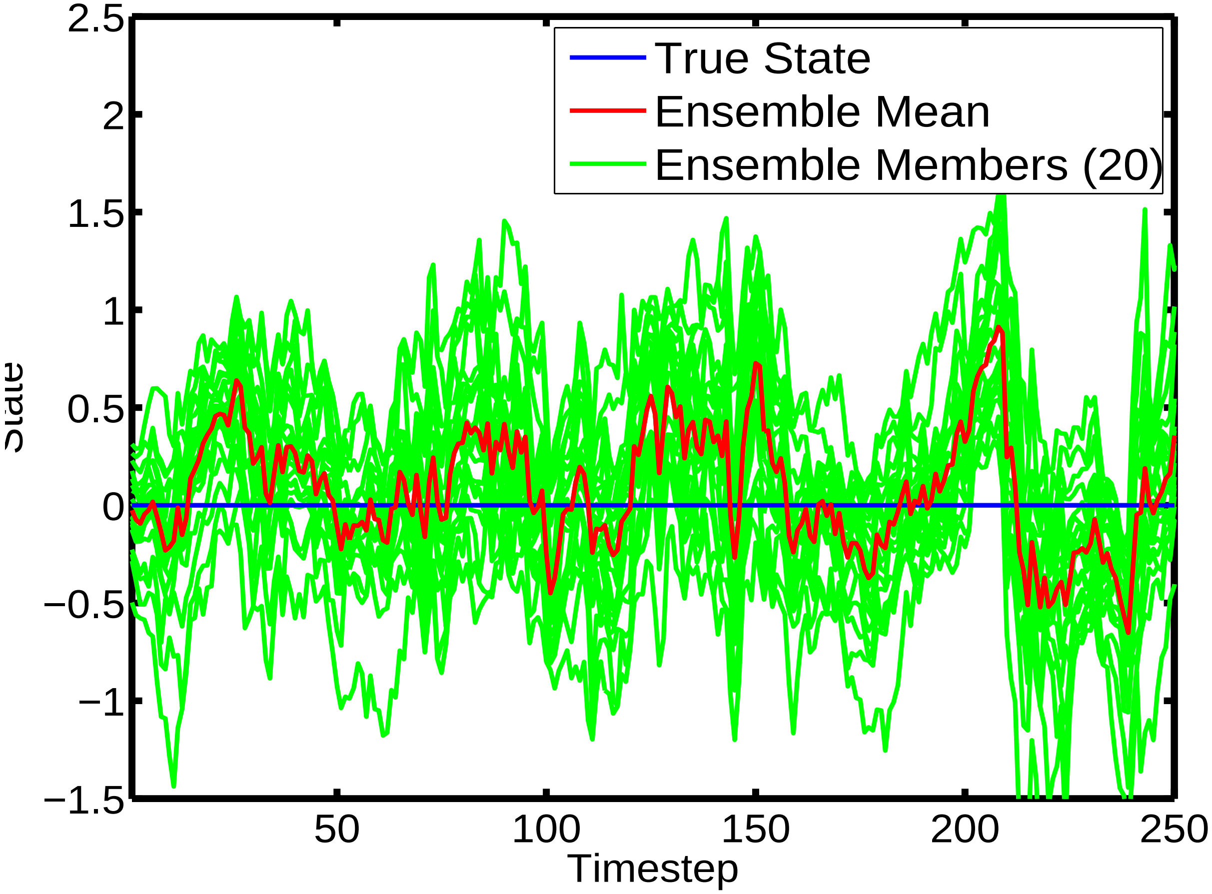


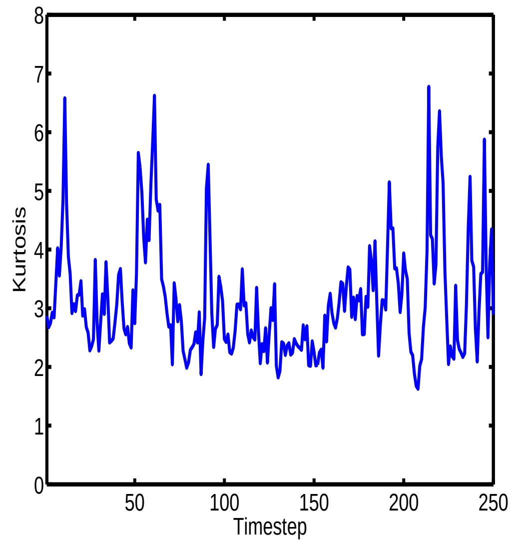
Model advance: furthest outlier pushed out fastest.

All members pulled in linearly by assimilation.

All members but outlier clump together; get huge kurtosis.

Nonlinear Model Results ($\alpha = 0.8$): EnKF



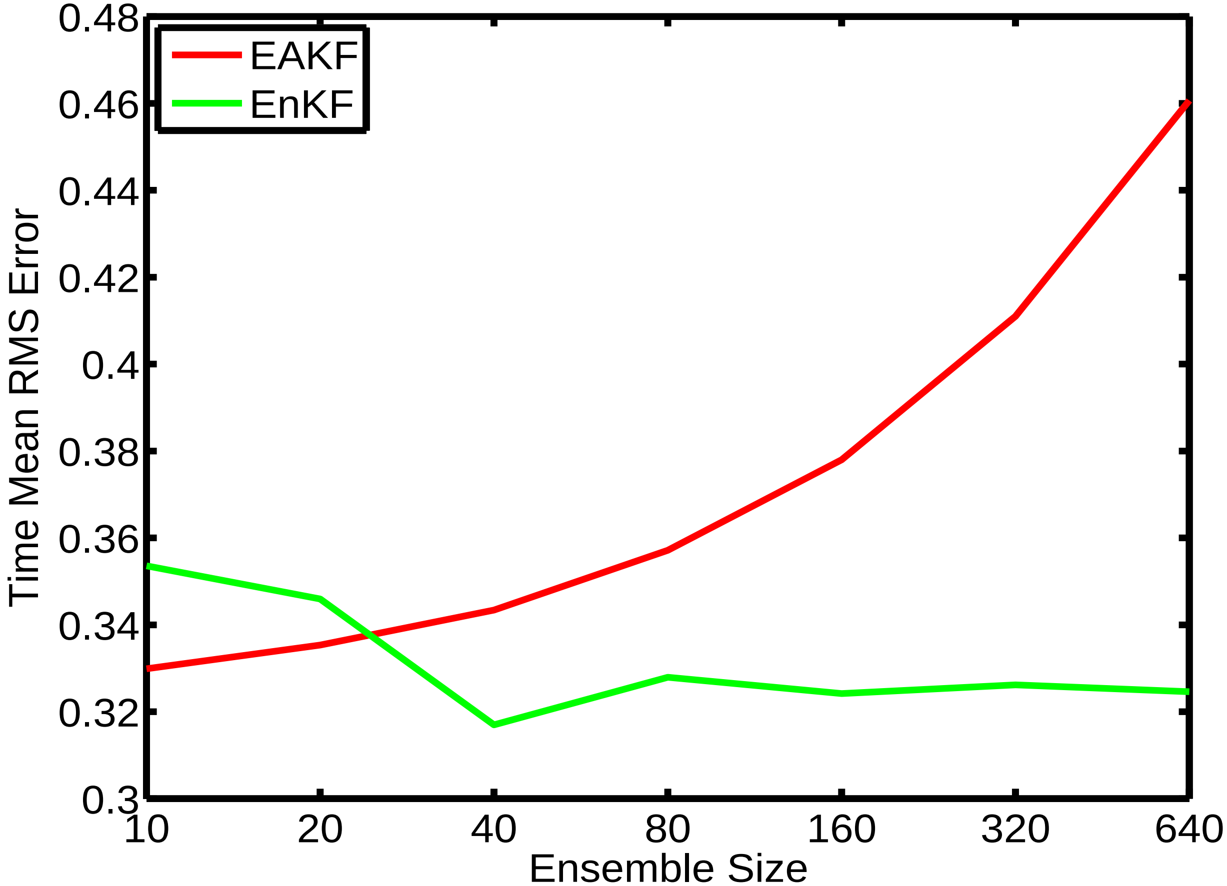


Model advance: furthest outlier pushed out fastest.

Assimilation mixes members some.

Still get high kurtosis sometimes.

EAKF Problem Gets Worse as Ensemble Size Increases (α = 0.2).



RMS error as function of ensemble size.

A Deterministic Non-Gaussian Observation Space Update.

1. Most ensemble filters assume prior and likelihood are ~gaussian.

2. Particle filters do full non-gaussian, but may not scale.

3. Assuming non-gaussian in observation space is possible.

4. Gaussian kernel filters have been proposed but are costly.

Requirements for an observation space update:

1. Low information content obs. can’t lead to large increments;

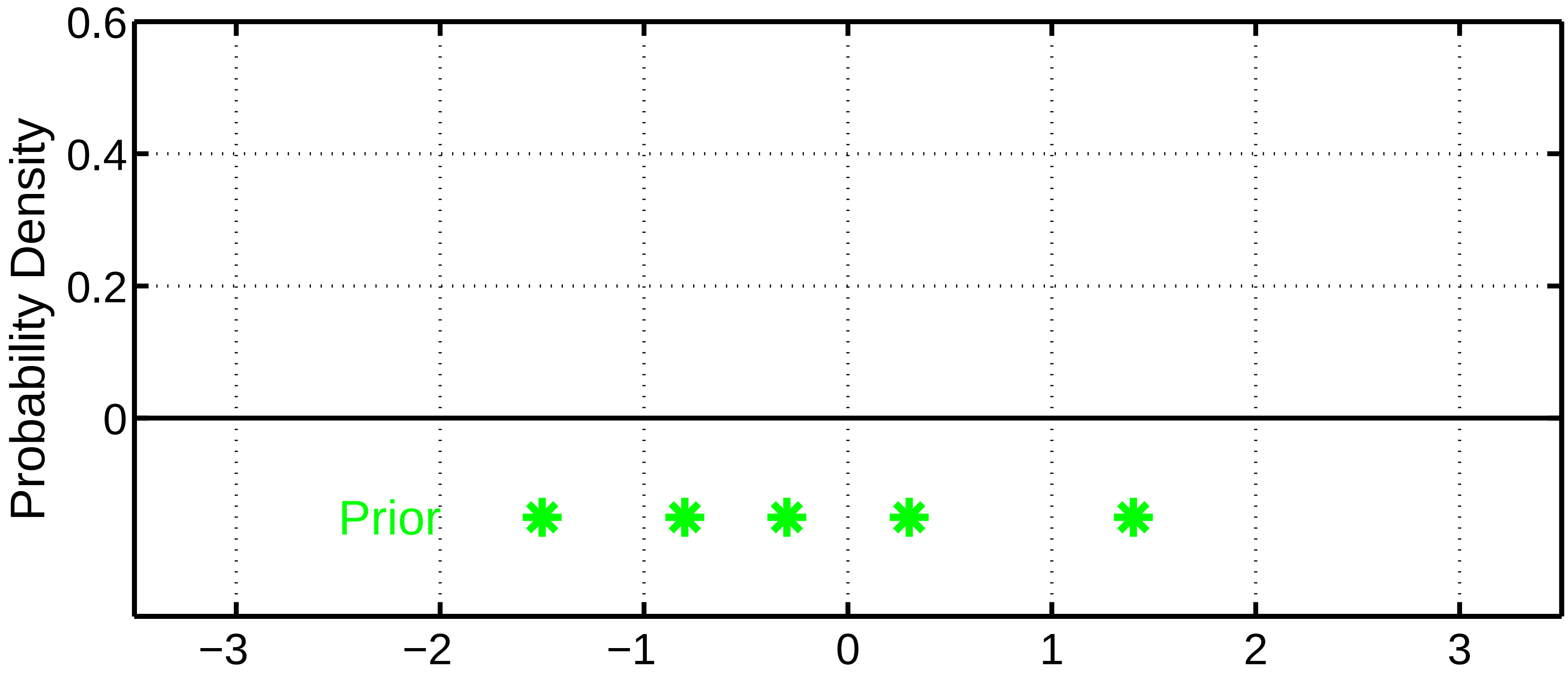
2. Want smallest possible increments for all cases;

3. Comparable to gaussian filters for ~gaussian cases;

4. Better than gaussian in non-gaussian cases;

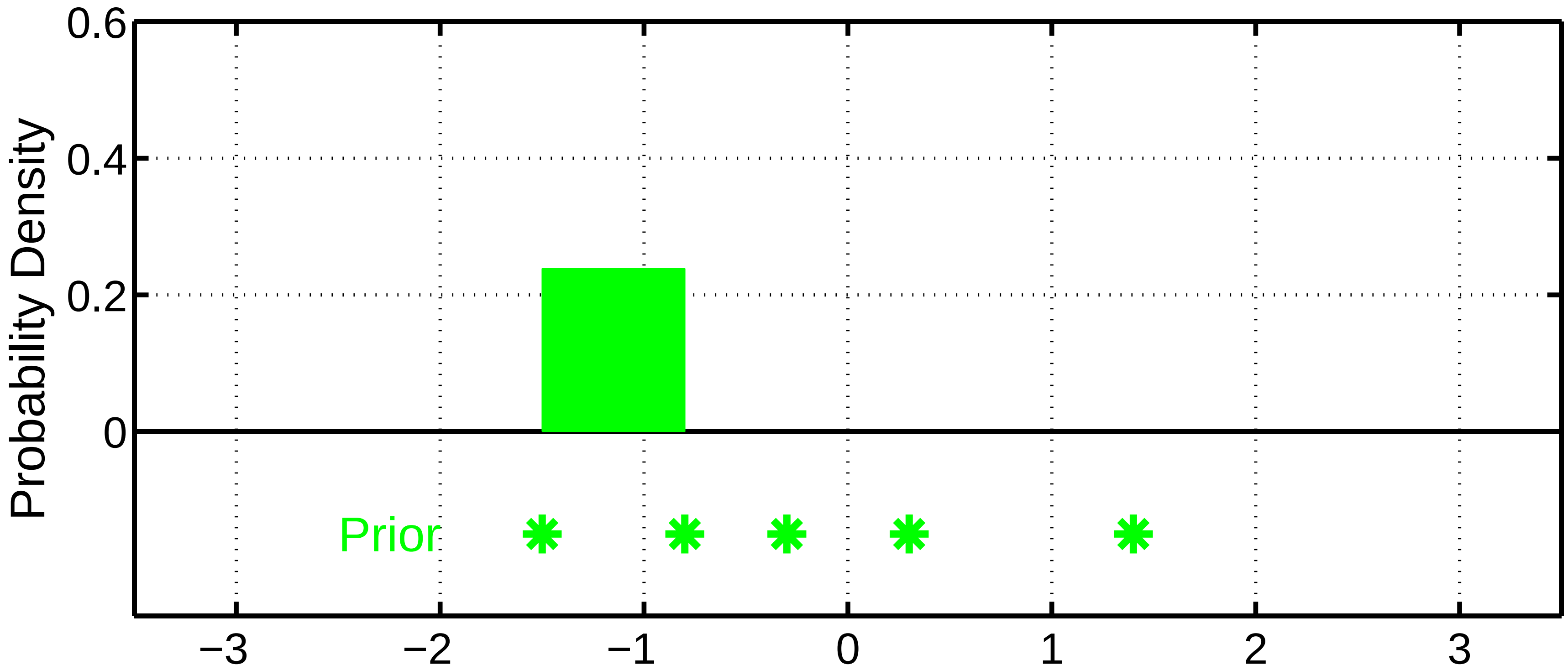
5. Computationally cheap.

Observation Space Rank Histogram Filter



Apply forward operator to each ensemble member.
Get prior ensemble in observation space.

Observation Space Rank Histogram Filter

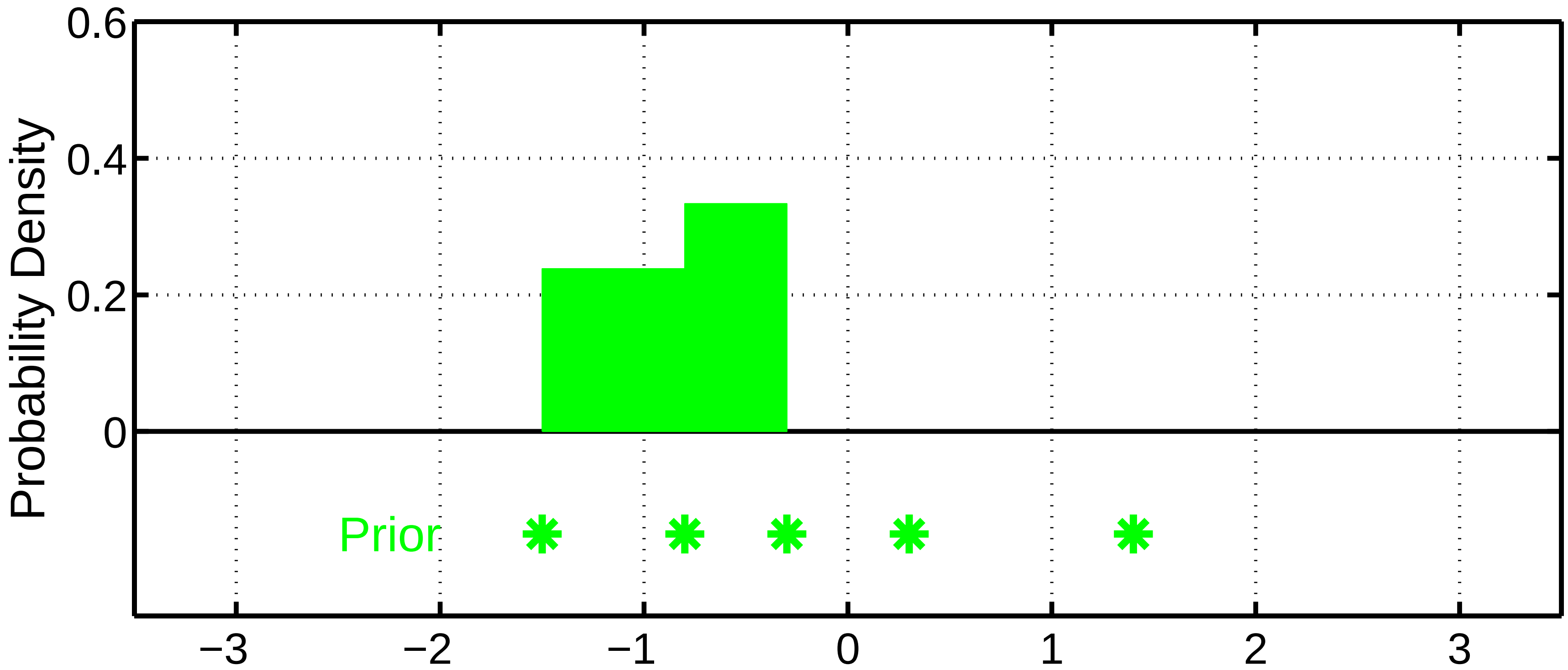


Step 1: Get continuous prior distribution density.

Place (ens_size + 1)$^{-1}$ mass between adjacent ensemble members.

Reminiscent of rank histogram evaluation method.

Observation Space Rank Histogram Filter

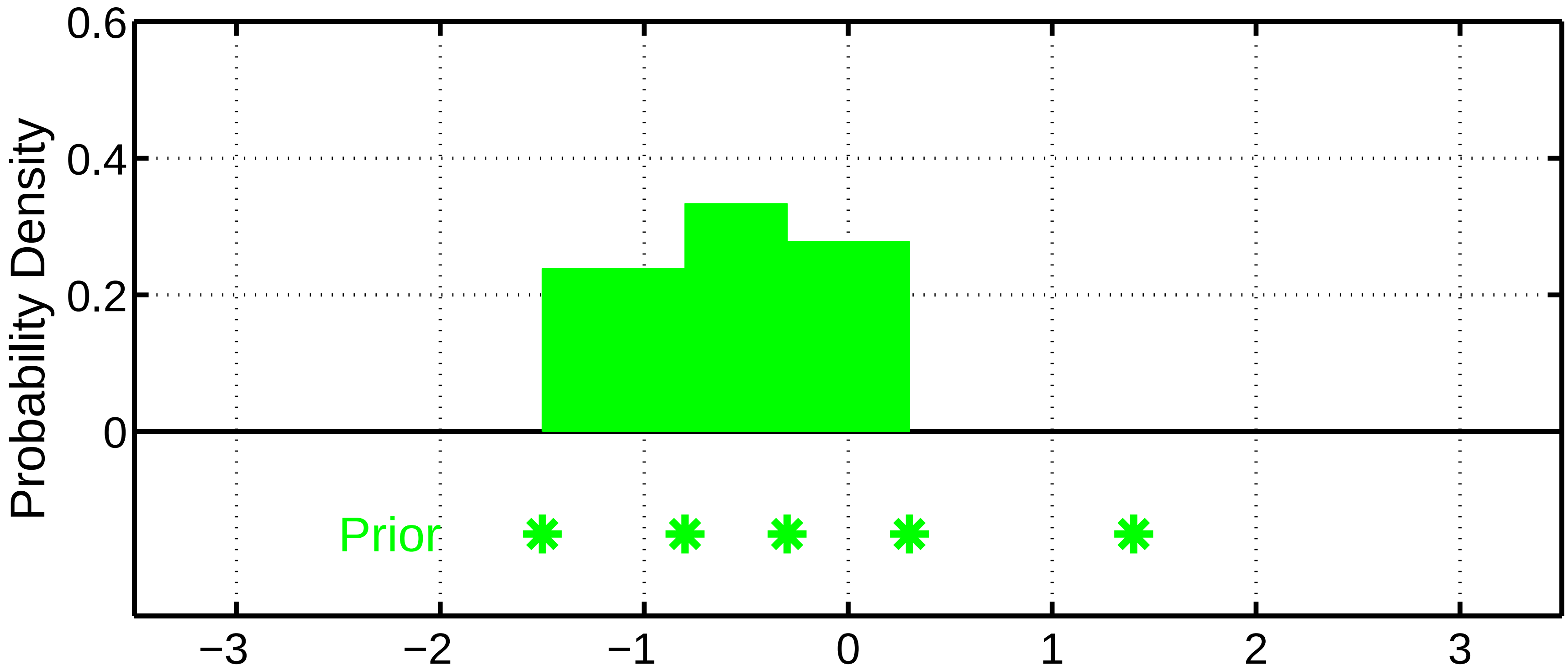


Step 1: Get continuous prior distribution density.

Place (ens_size + 1)$^{-1}$ mass between adjacent ensemble members.

Reminiscent of rank histogram evaluation method.

Observation Space Rank Histogram Filter

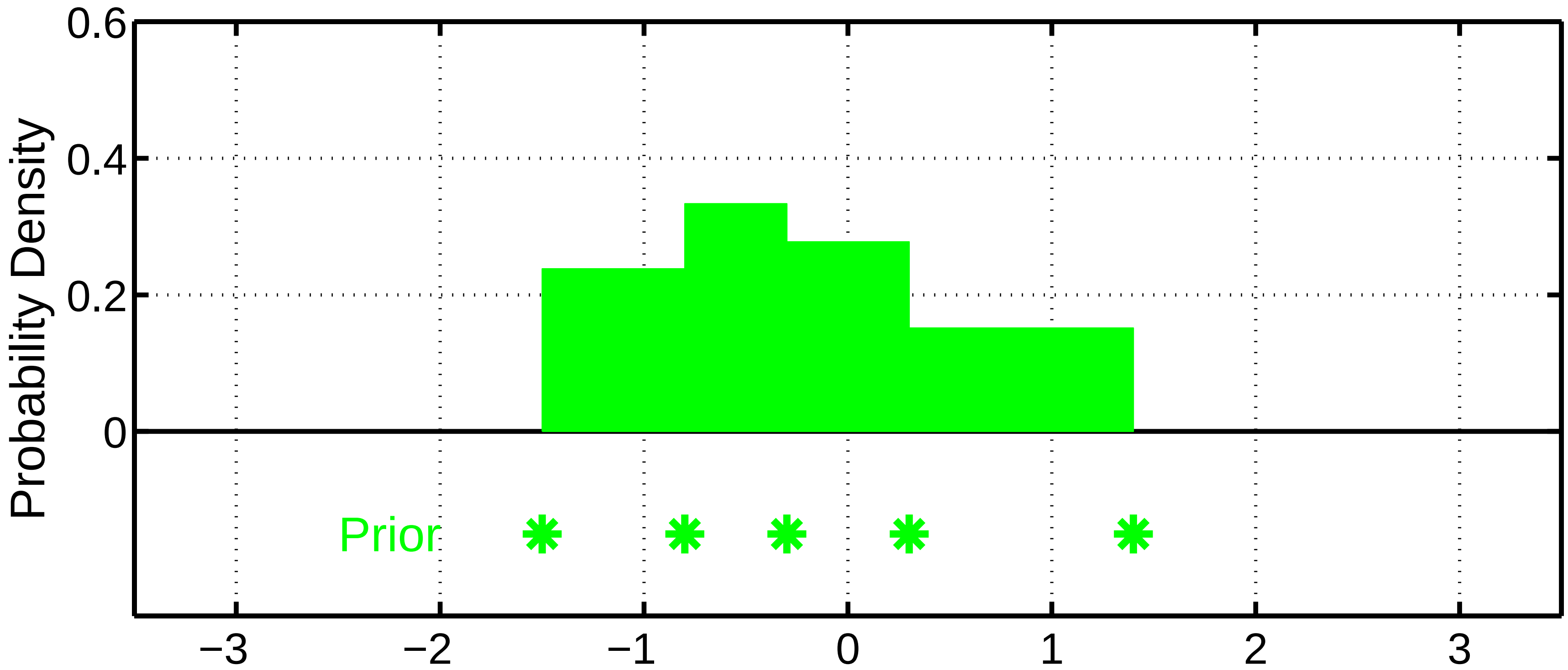


Step 1: Get continuous prior distribution density.

Place $(\text{ens\_size} + 1)^{-1}$ mass between adjacent ensemble members.

Reminiscent of rank histogram evaluation method.

Observation Space Rank Histogram Filter

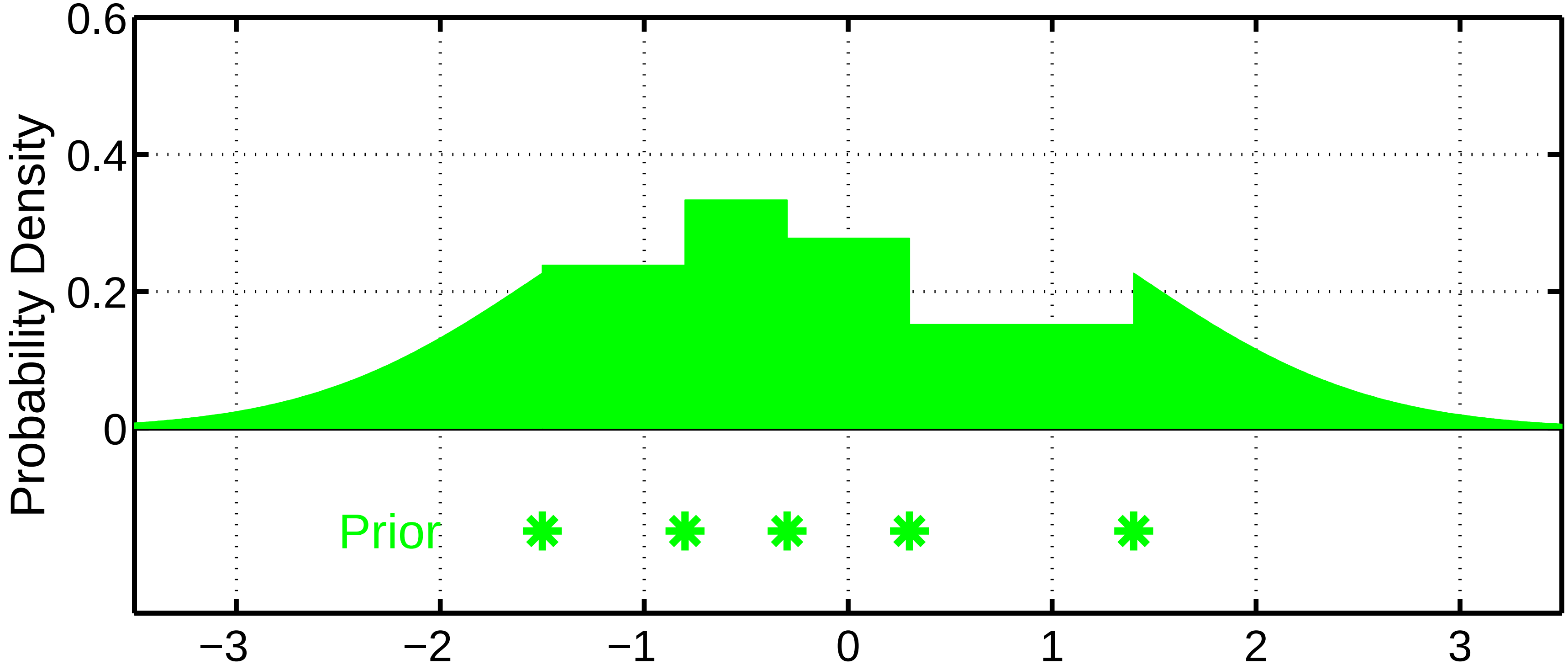


Step 1: Get continuous prior distribution density.

Place $(\text{ens\_size} + 1)^{-1}$ mass between adjacent ensemble members.

Reminiscent of rank histogram evaluation method.

Observation Space Rank Histogram Filter



Step 1: Get continuous prior distribution density.

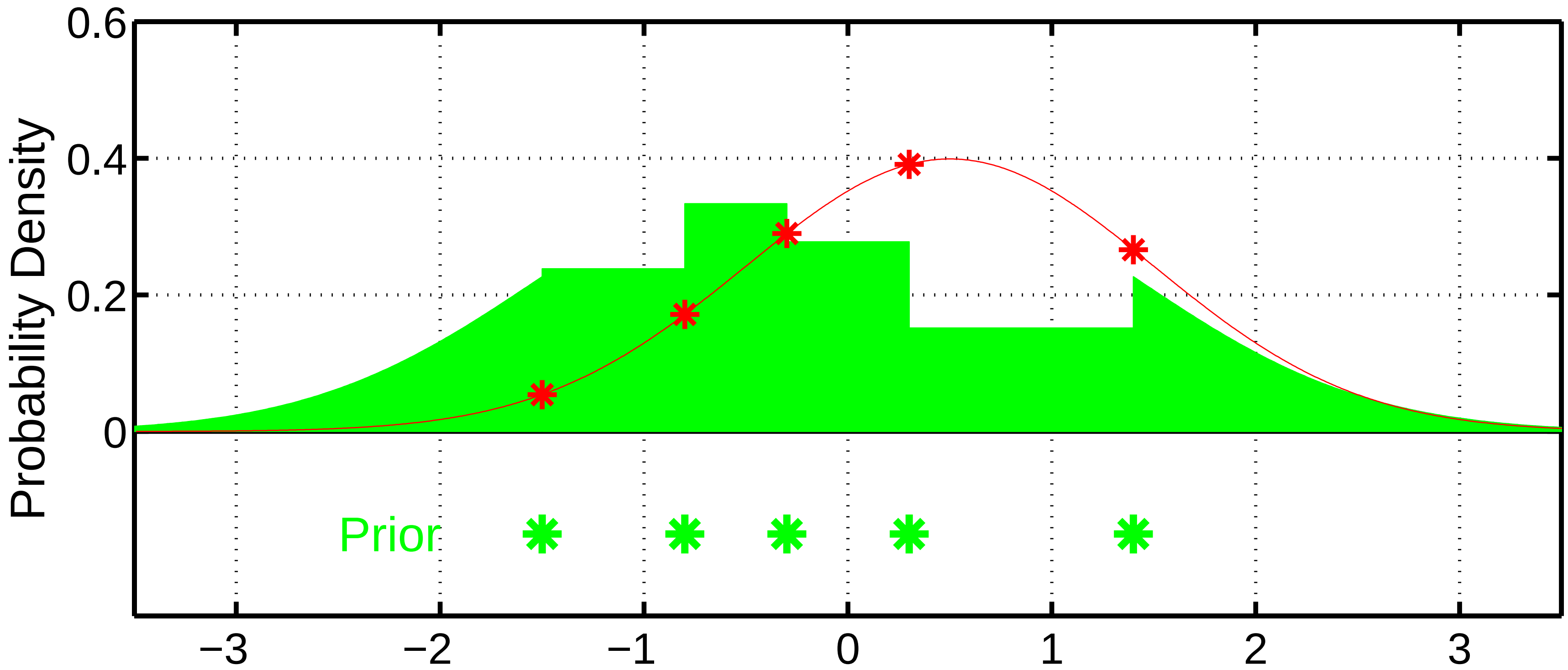
Place (ens_size + 1)$^{-1}$ mass between adjacent ensemble members.

Partial gaussian kernels on tails, N(*tail_mean*, σ_{ens}).

tail_mean selected so that (ens_size + 1)$^{-1}$ mass is in tail.

Performance is sensitive to the tail structure.

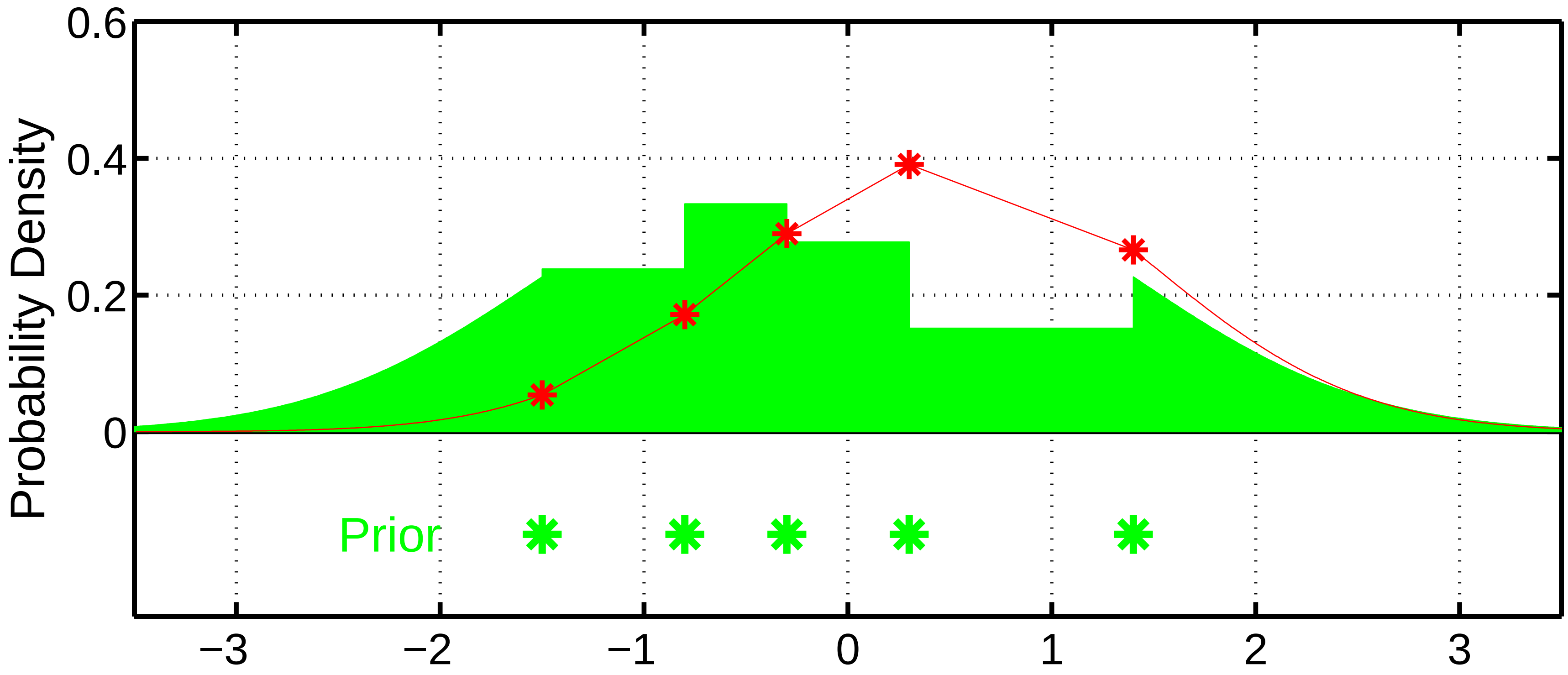
Observation Space Rank Histogram Filter



Step 2: Get an approximation of likelihood.

Could use full gaussian but this is expensive.

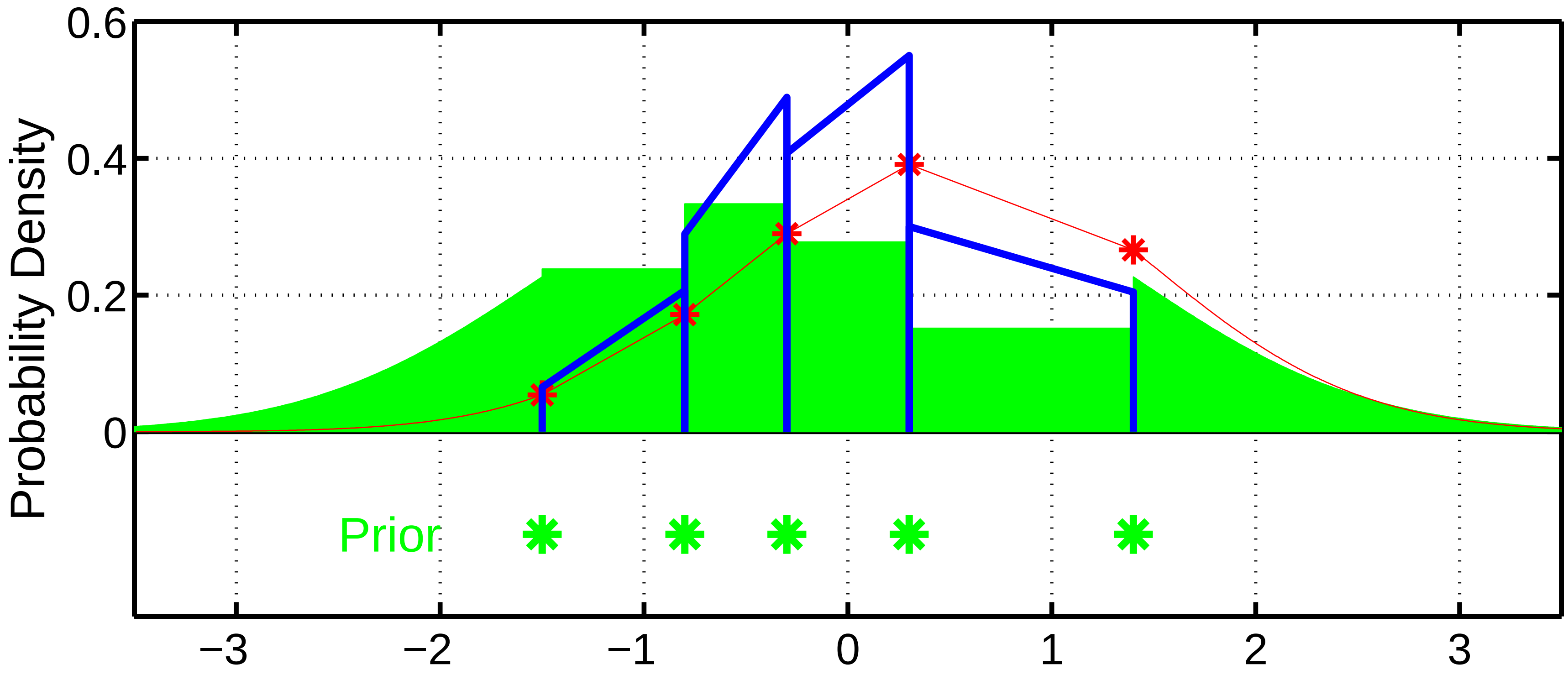
Observation Space Rank Histogram Filter



Step 2: Get an approximation of likelihood.

Example: linear fit between value at ensemble members.
Keep exact gaussian form on tails.

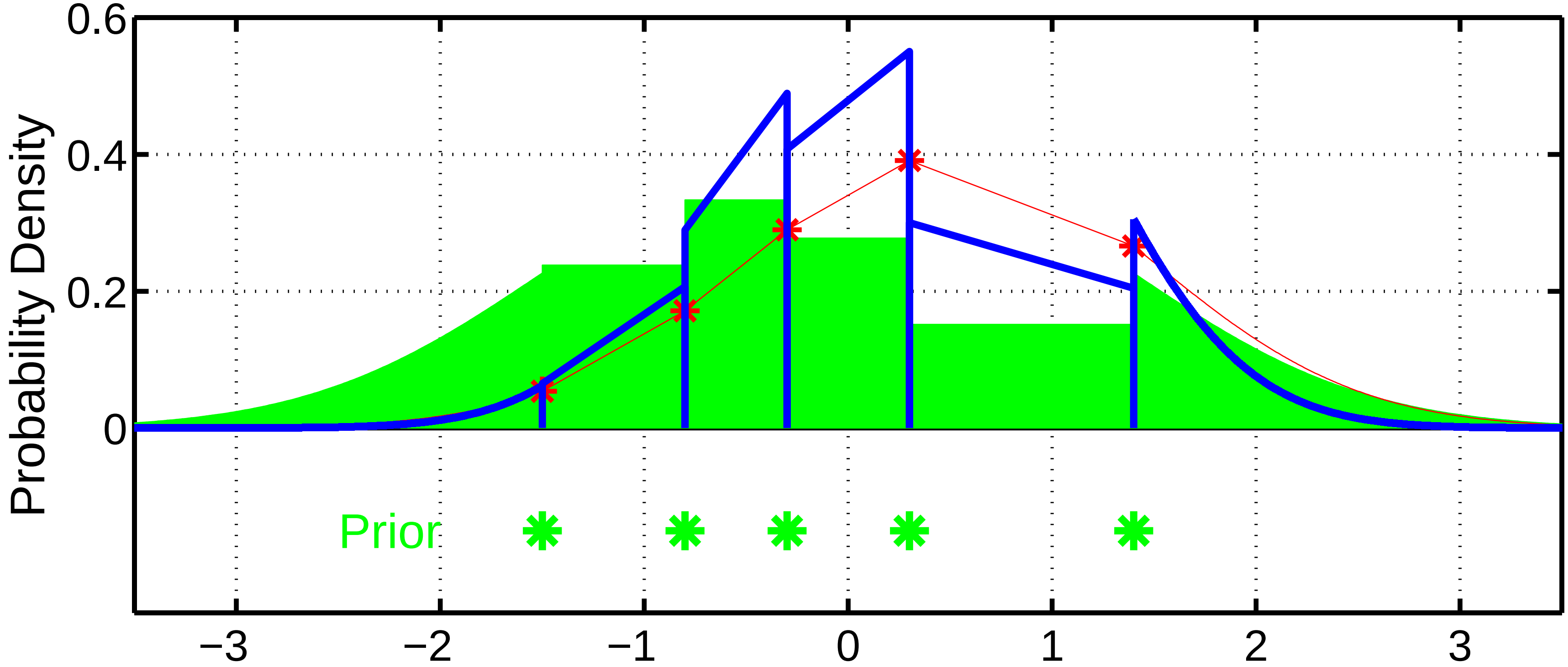
Observation Space Rank Histogram Filter



Step 3: Compute prior*likelihood to get posterior.

Simple and cheap in interior.
(Displayed product normalized to make posterior a PDF).

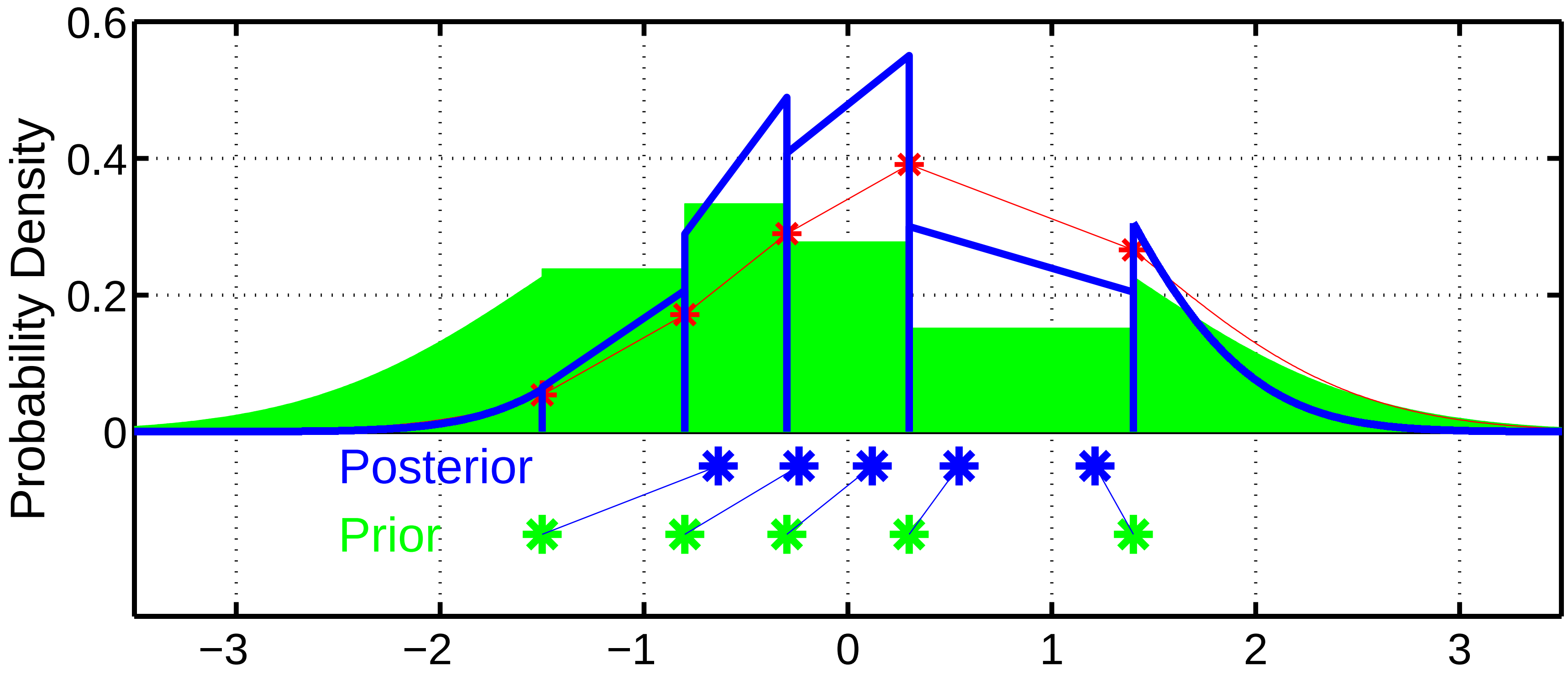
Observation Space Rank Histogram Filter



Step 3: Compute prior*likelihood to get posterior.

Product of gaussians on tails (we have to use the weight here!).

Observation Space Rank Histogram Filter



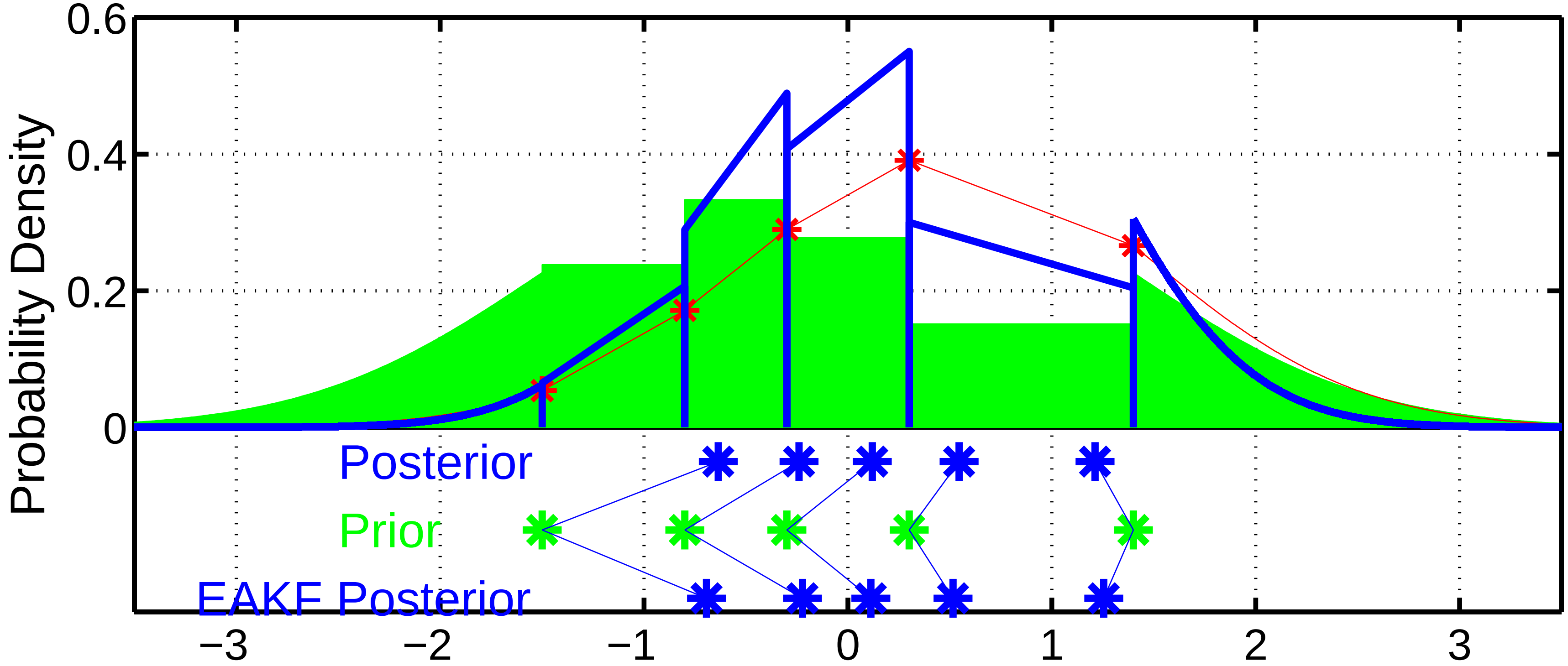
Step 4: Compute updated ensemble members:

(ens_size + 1)$^{-1}$ of posterior mass between each ensemble pair.

(ens_size + 1)$^{-1}$ in each tail.

Uninformative observation has no impact.

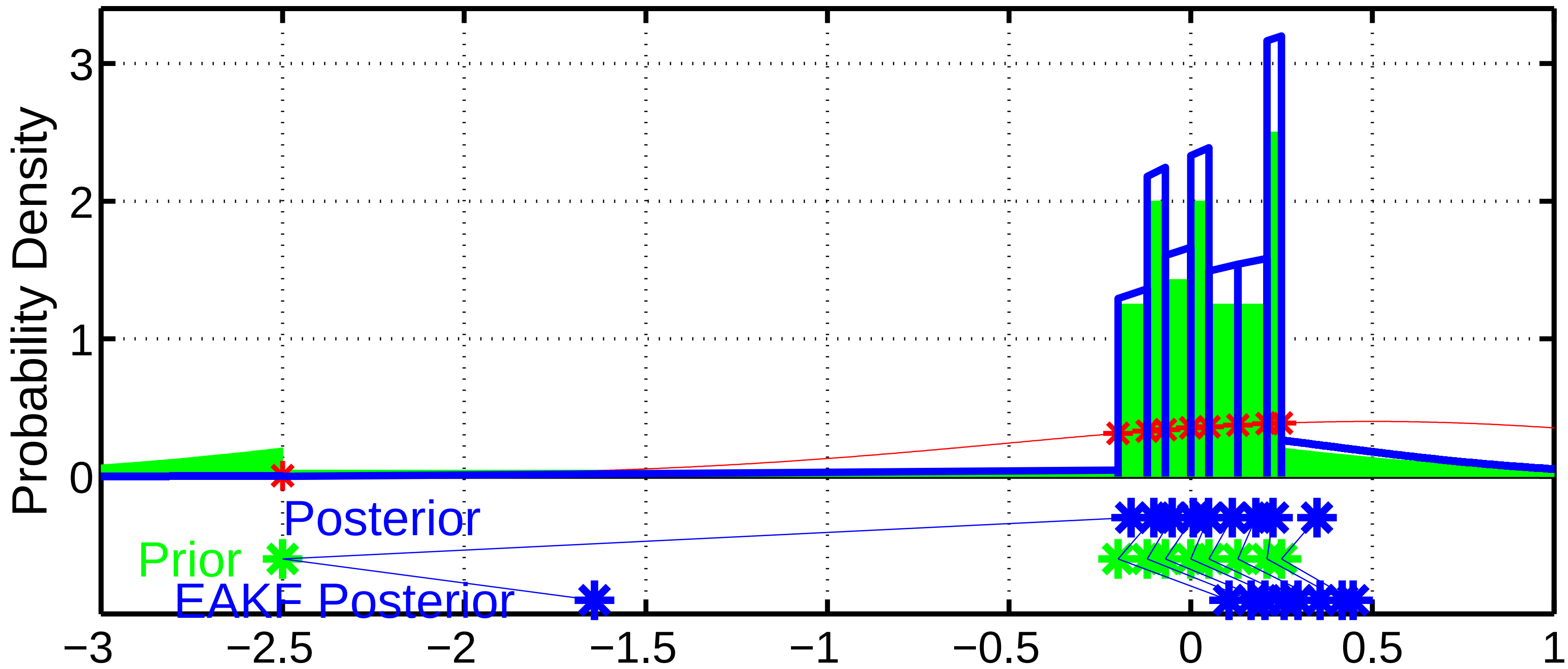
Observation Space Rank Histogram Filter



Compare to standard Ensemble Adjustment Filter (EAKF).

Nearly gaussian case, differences are small.

Outliers are a Challenge for Gaussian Filters

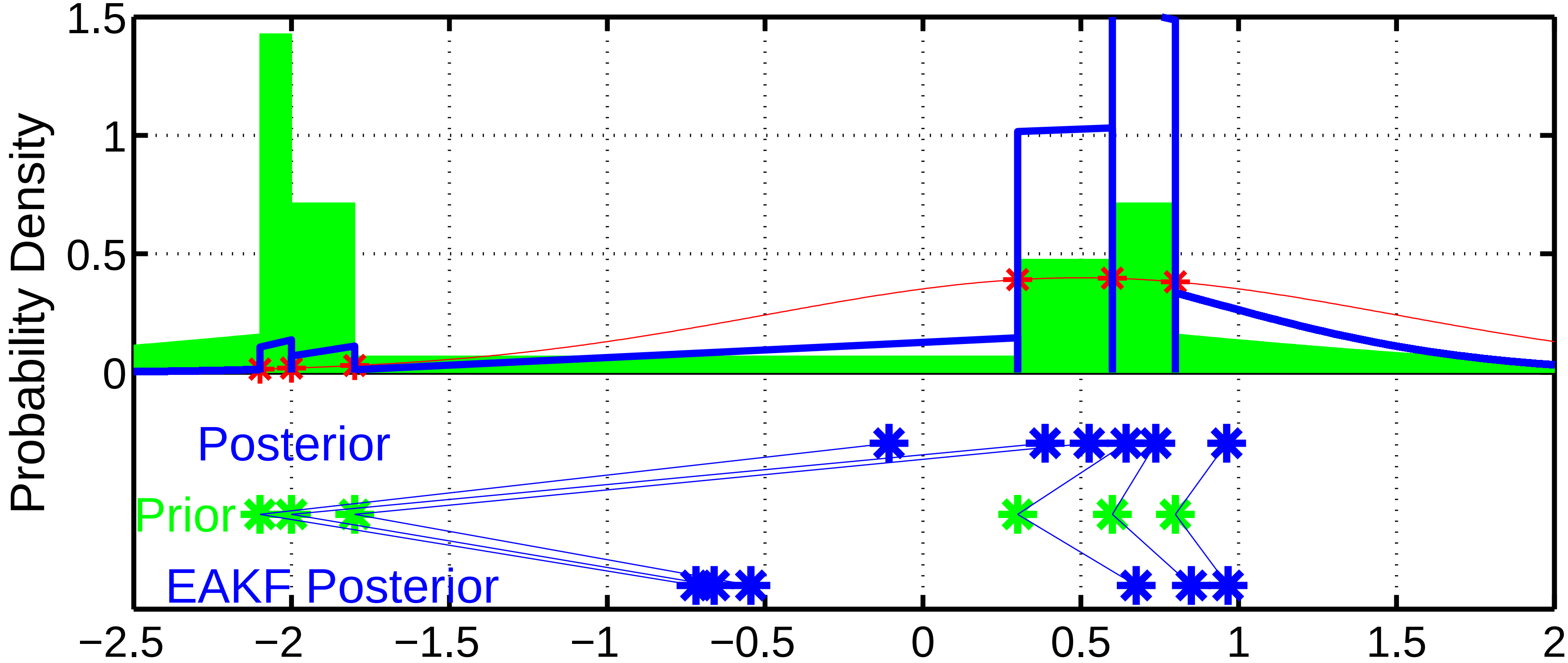


Rank Histogram gets rid of outlier that is clearly inconsistent with obs.

EAKF can't get rid of outlier.

Large prior variance from outlier causes EAKF to shift all members too much towards observation.

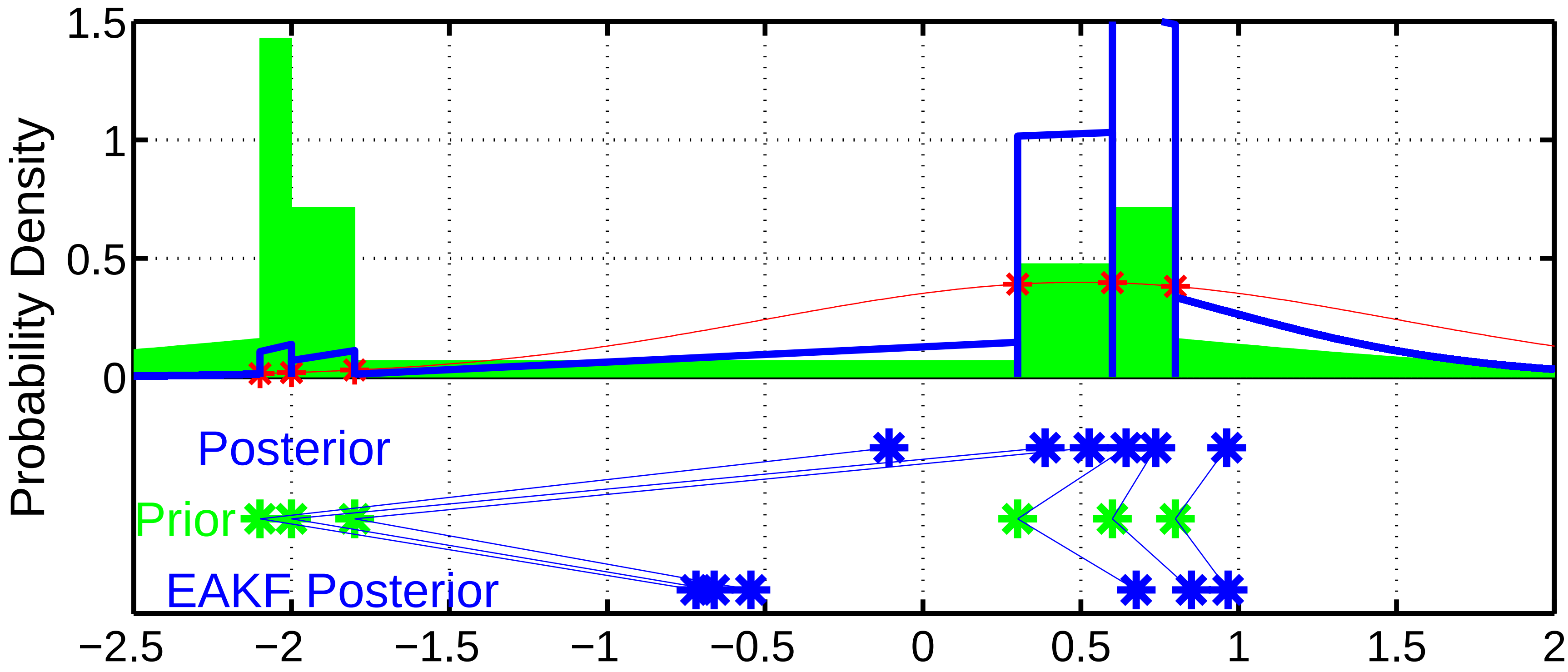
Multimodal Prior Distributions



Rank Histogram handles multimodal prior and compelling observation.
EAKF still bimodal; left mode is inconsistent with everything.

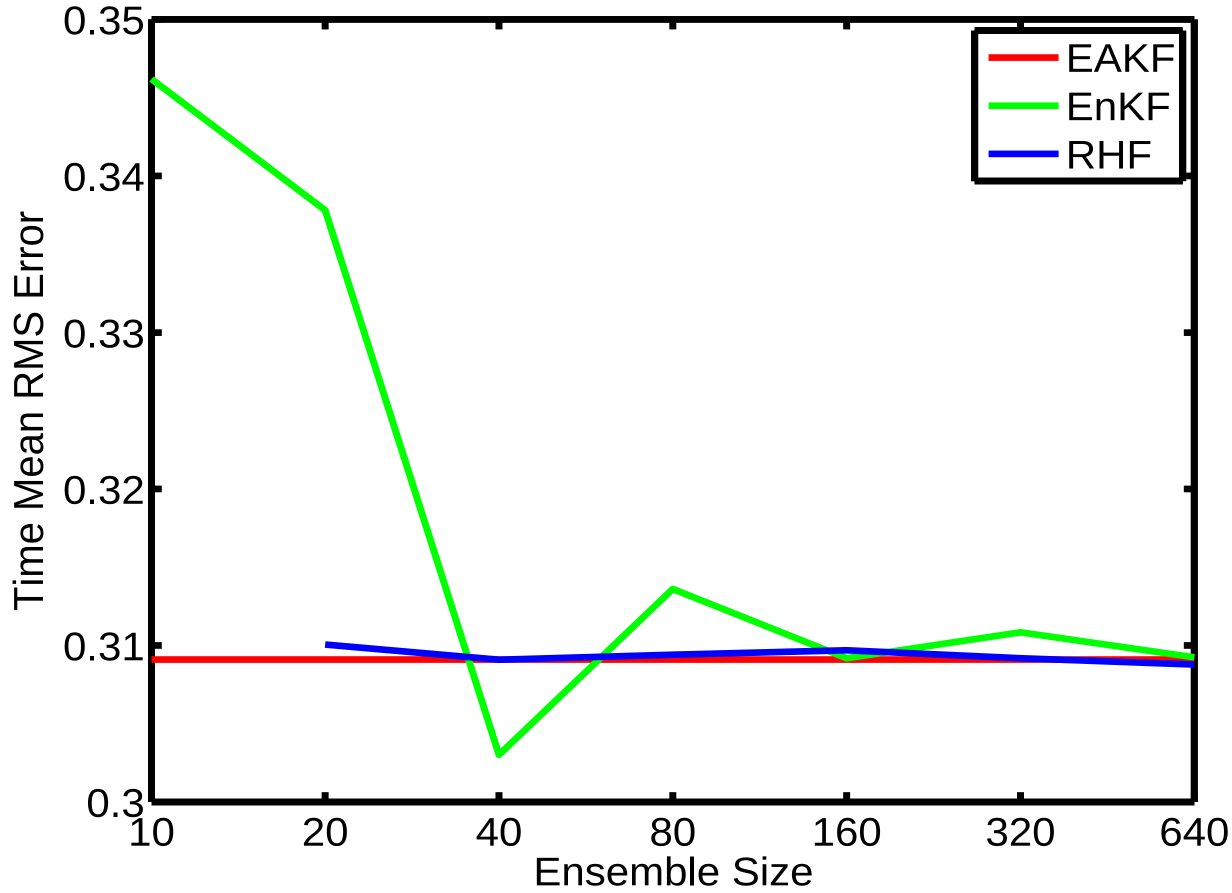
Lorenz_63 can have priors like this.

Multimodal Prior Distributions



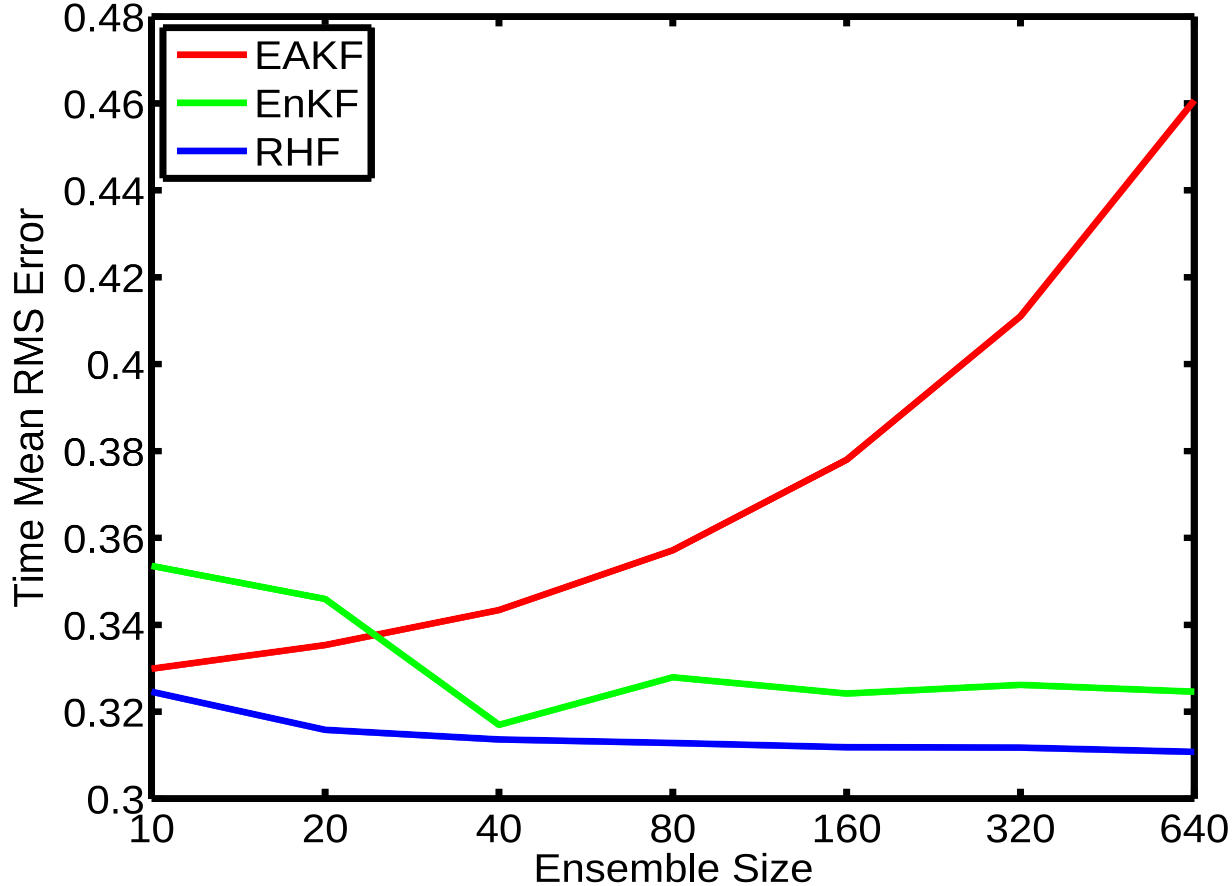
Convective scale models have analogous behavior.
Convection may fire at 'random' locations.
Subset of ensembles will be in right place, rest in wrong place.
Want to aggressively eliminate convection in wrong place.

Results: Linear Model (α = 0), Time-Mean RMS



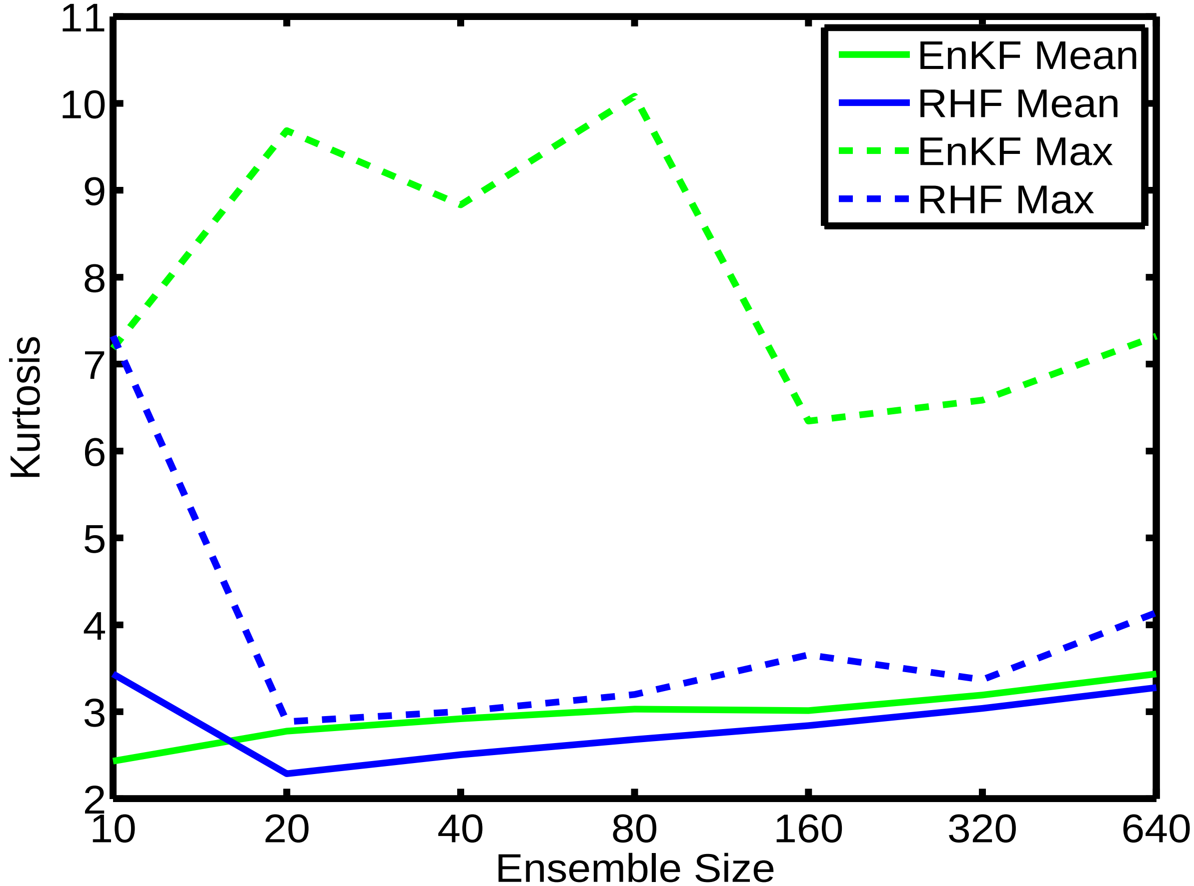
Rank Histogram Filter (RHF) fails for 10 members.
Competitive for >10 members.

Results: Nonlinear Model RMS (α = 0.2), Time-Mean RMS



RHF best for all ensemble sizes.

Results: Nonlinear Model Kurtosis ($\alpha = 0.2$)



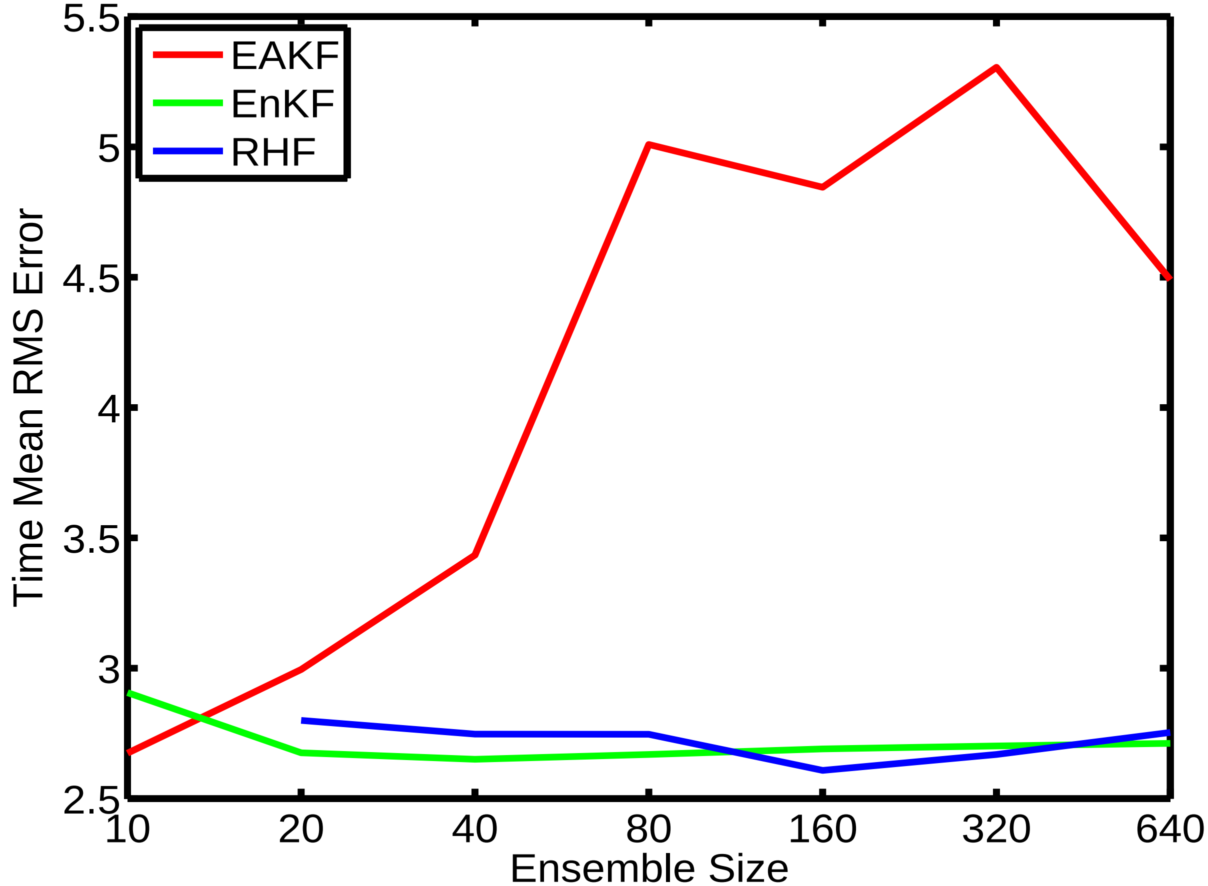
RHF smallest for ensemble sizes > 10.

Doesn't have outlier excursions (max is small).

EAKF has HUGE kurtosis (off the plot).

Results: Lorenz63 RMS

All 3 state variables observed, error variance 1.0

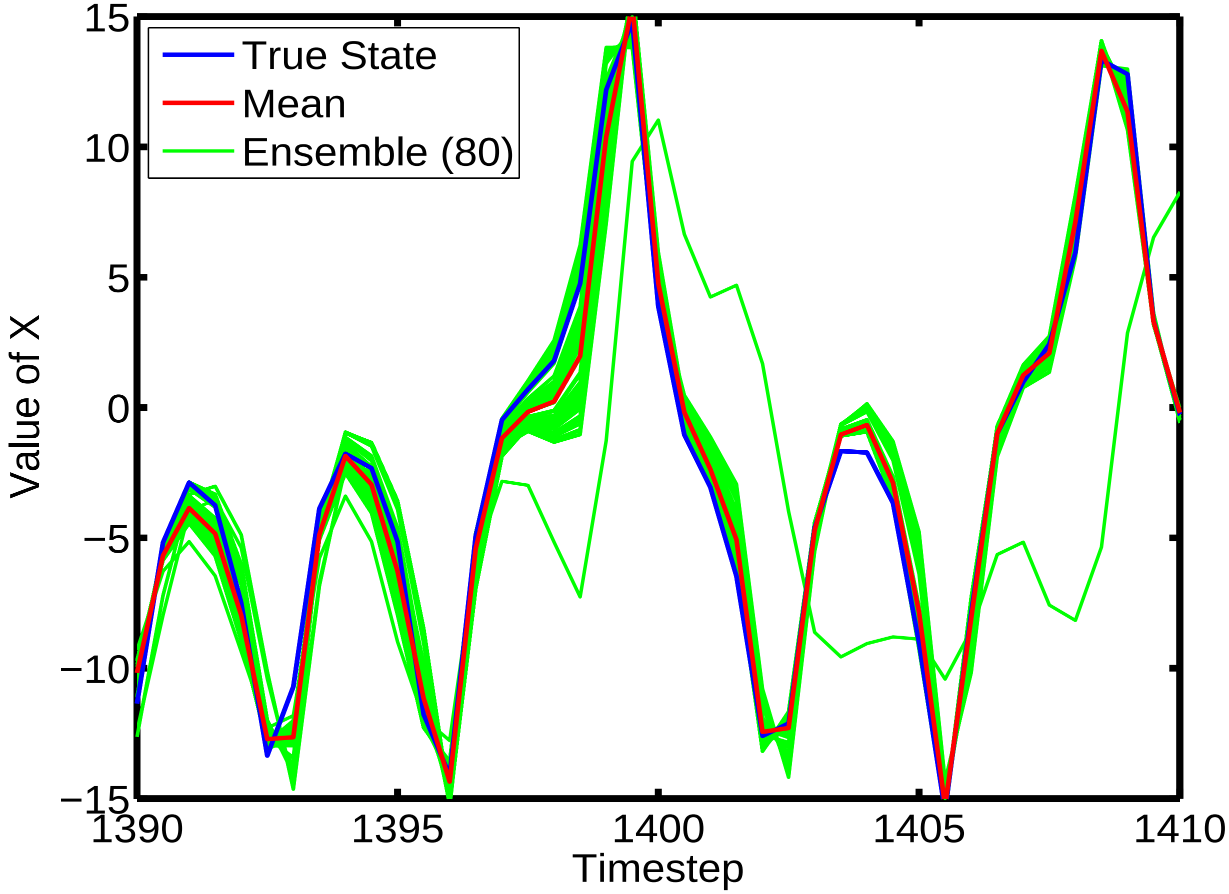


RHF and EnKF comparable.

EAKF gets progressively worse (but pretty good for 10 members).

Results: Lorenz63 EAKF

All 3 state variables observed, error variance 1.0



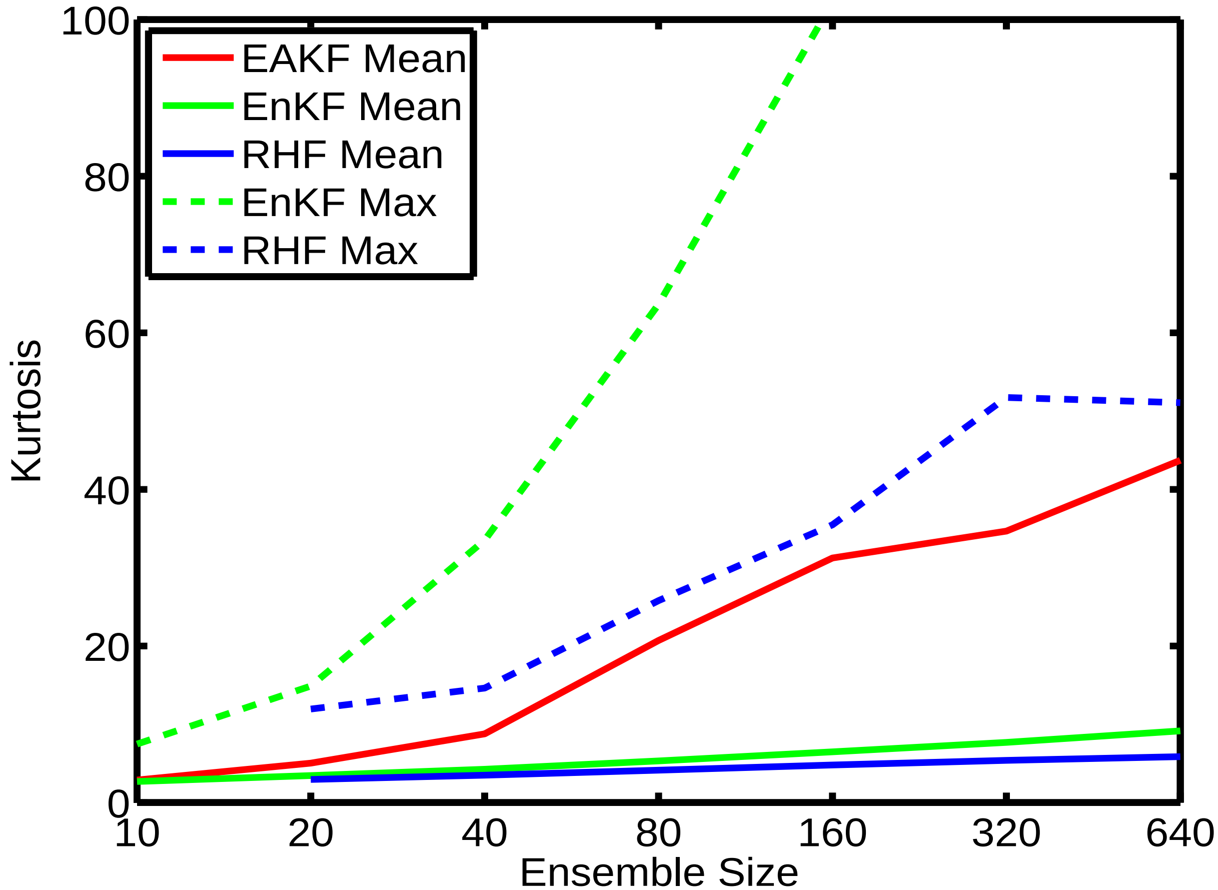
Wandering ensemble member can detach, head into wrong lobe.

Happens less frequently and severely in EnKF.

Can reattach due to mixing from other variables.

Results: Lorenz63 Kurtosis

All 3 state variables observed, error variance 1.0



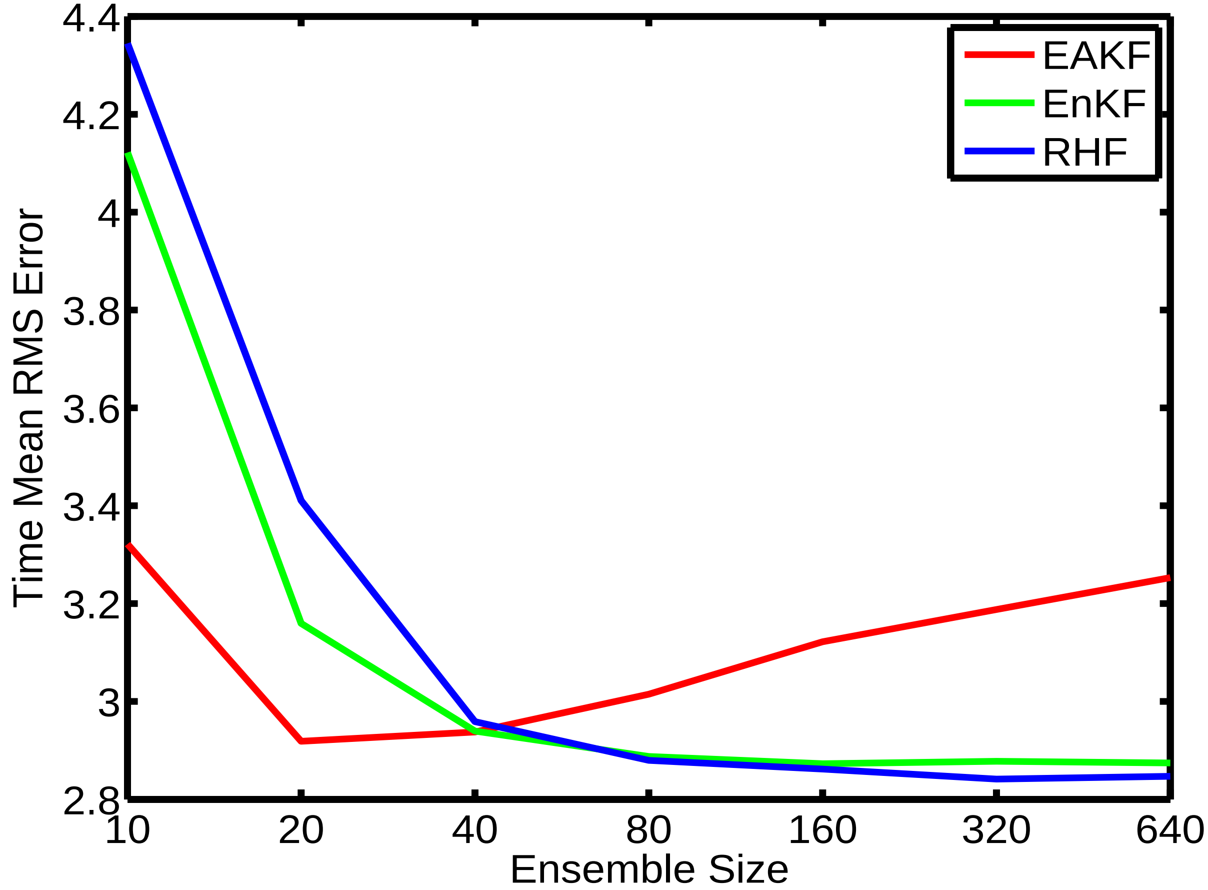
Kurtosis is big in EAKF due to detached outliers.

Happens less frequently and severely in EnKF.

Outliers quickly eliminated by RHF, kurtosis stays smaller.

Results: Lorenz96 RMS

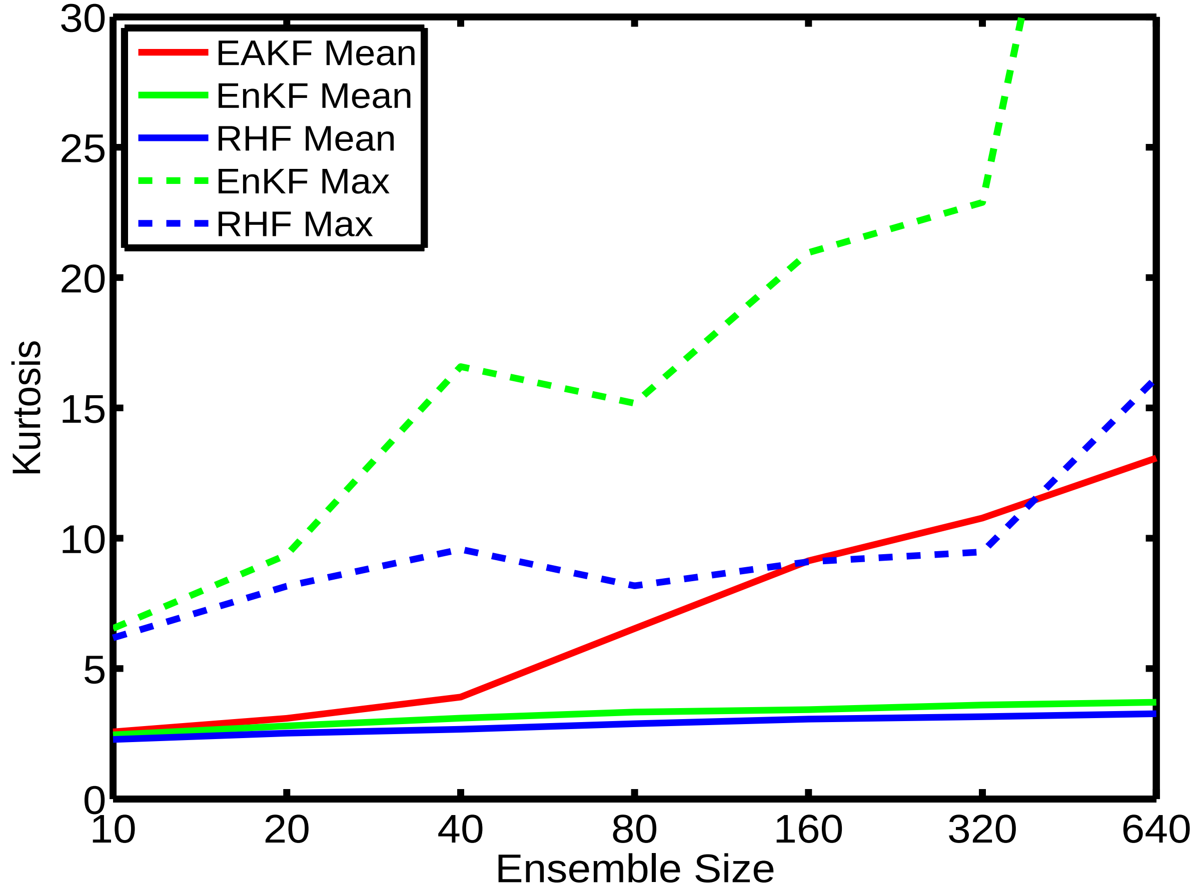
40 Observations, average of adjacent state variables, Error var = 4.
Localization halfwidth 0.3 of domain, adaptive inflation.



EAKF RMS increases moderately with ensemble size.
EnKF and RHF comparable for larger ensembles.

Results: Lorenz96 Kurtosis

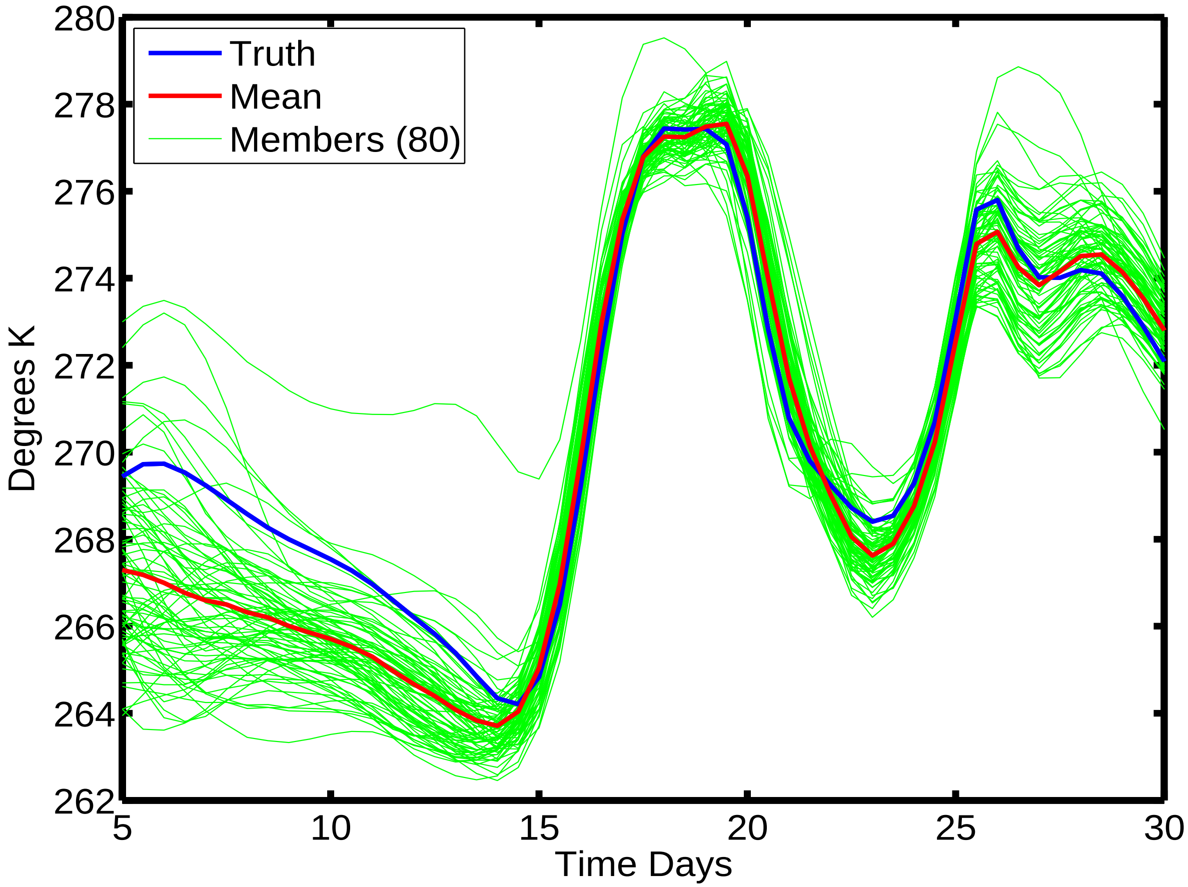
40 Observations, average of adjacent state variables, Error var = 4.
Localization halfwidth 0.3 of domain, adaptive inflation.



EnKF has sporadic large kurtosis, increases with ensemble size.
EAKF max kurtosis very large (off plot).

Results: Dynamical Core of GFDL AM2 GCM

1. EAKF: sporadic spatially localized outlier behavior.



Lowest level T 30W, 50N.

300 radiosonde profiles every 12 hours.

No inflation.

0.2 radian localization.

2. EnKF: similar behavior less frequently.

3. RHF: no evidence this occurs.

Results: Global NWP in Finite Volume CAM

1. Limited evidence of outlier events in any filter.

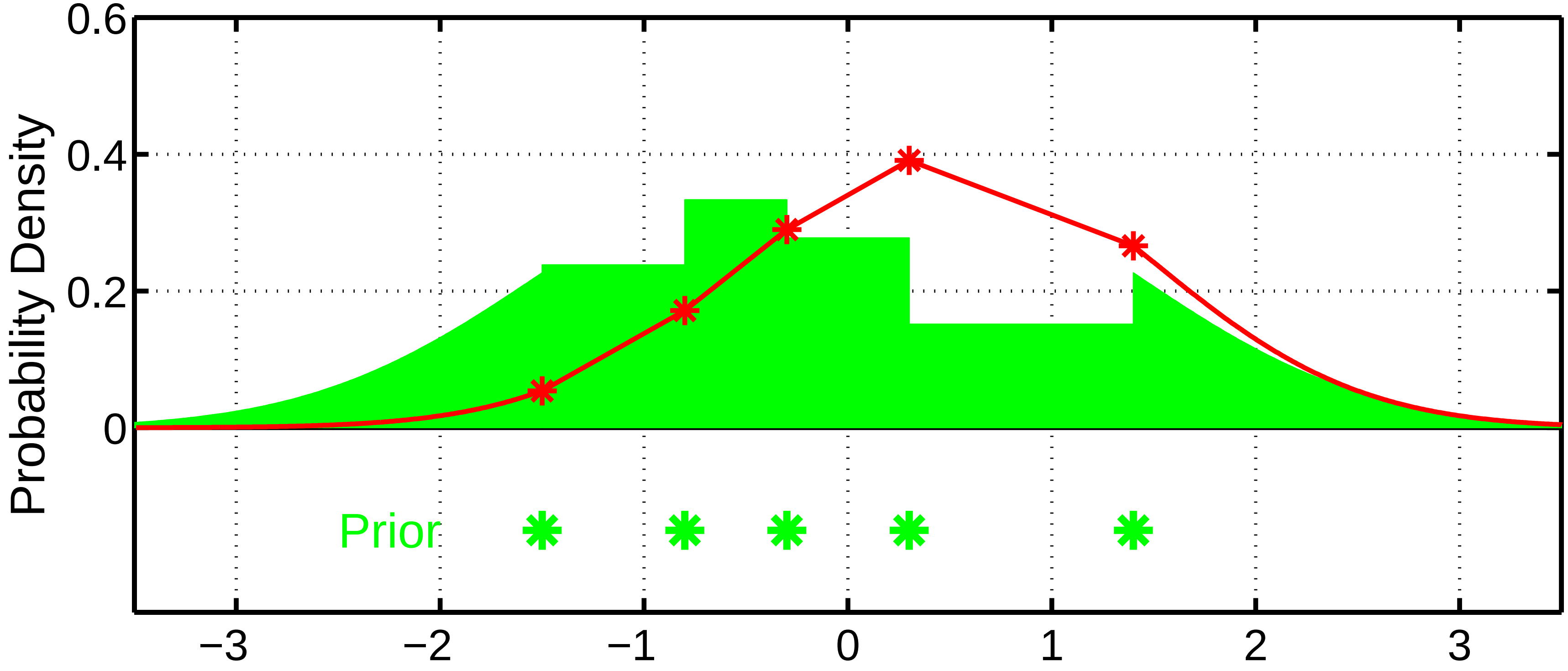
2. Prior fit to observations:

   80-member EAKF and RHF virtually indistinguishable.

   (Comparable to NCEP operational, better in tropics, near sfc.).

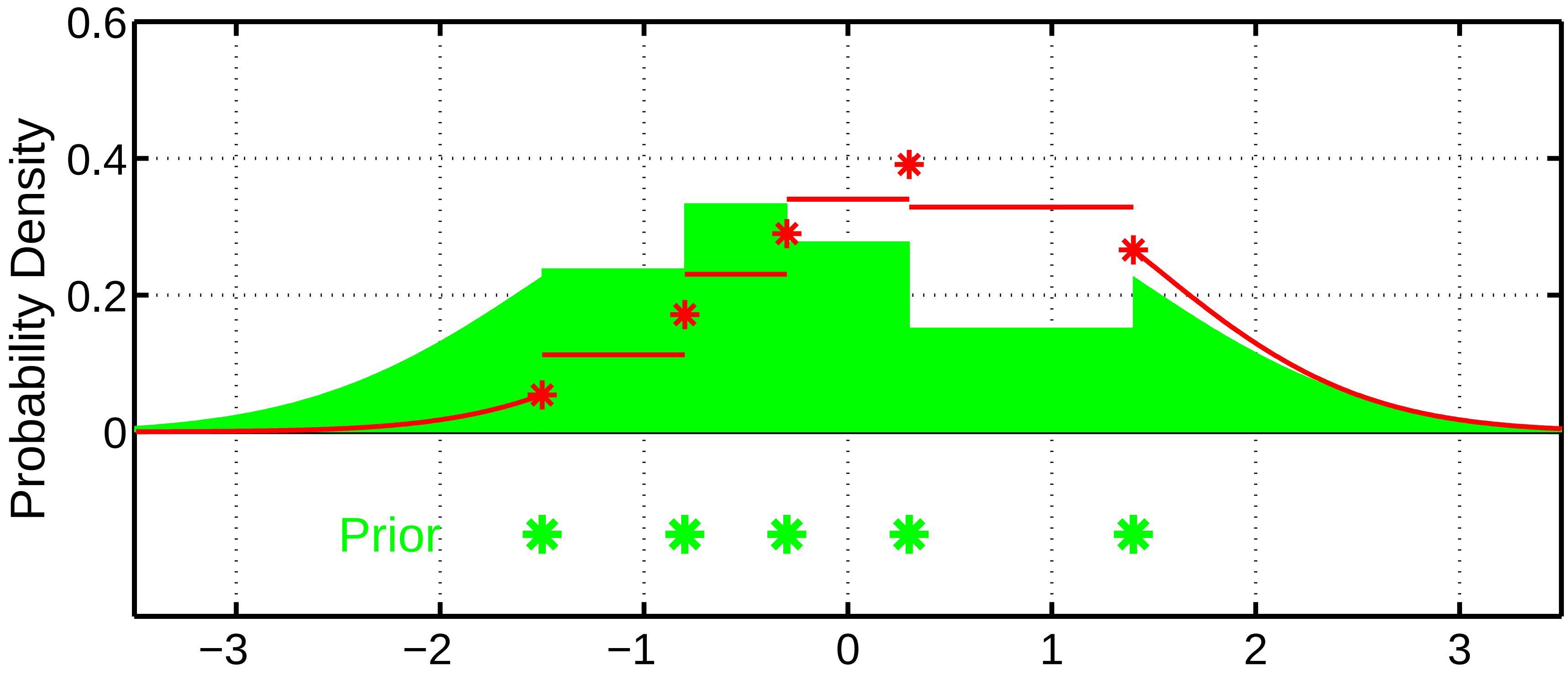
   80-member EnKF significantly worse.

Other ways to Approximate the Likelihood



Results shown for linear interpolation in interior, gaussian tails.

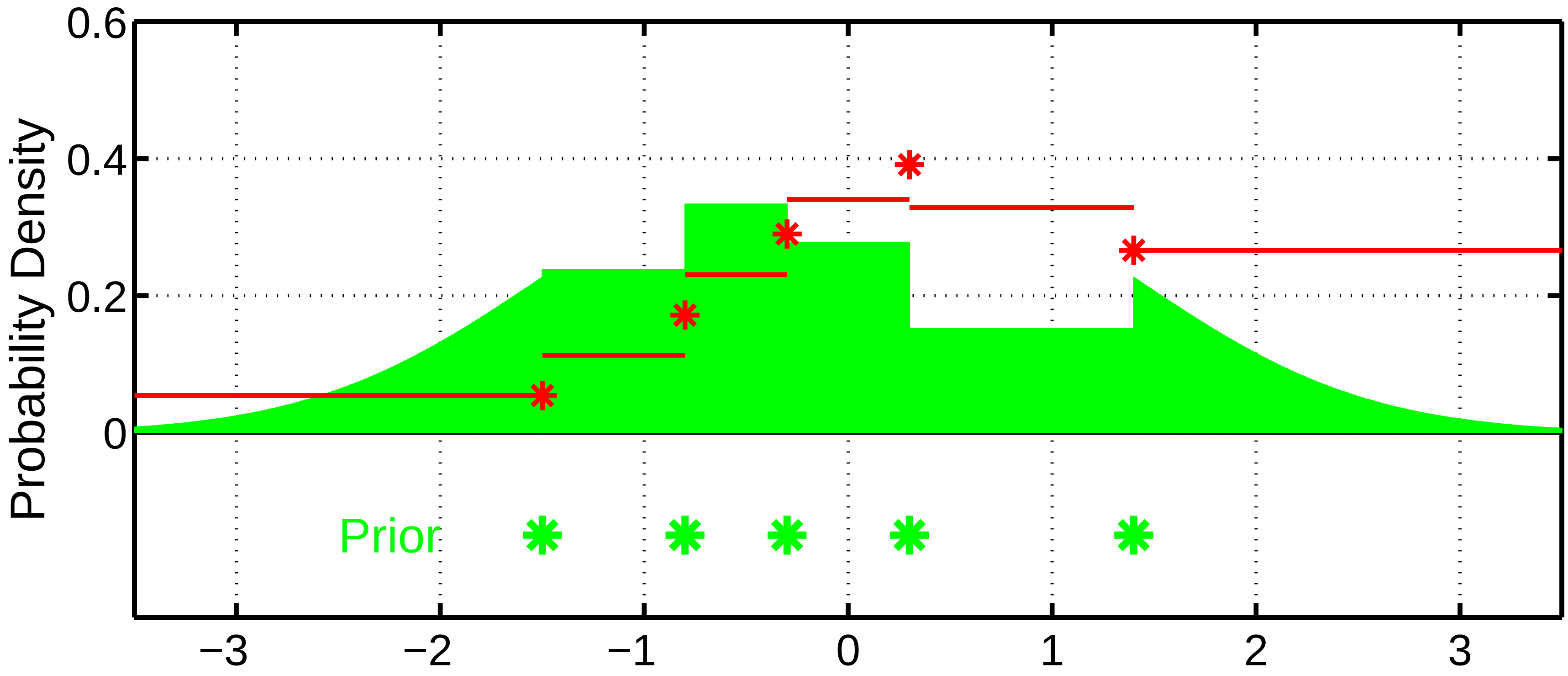
Other ways to Approximate the Likelihood



Average likelihood at bounding ensemble members for interior.

- Computing posterior ensemble becomes very cheap.
- Expected spread of posterior increased. This can be good.
- RMS errors mostly indistinguishable from base case.

Other ways to Approximate the Likelihood



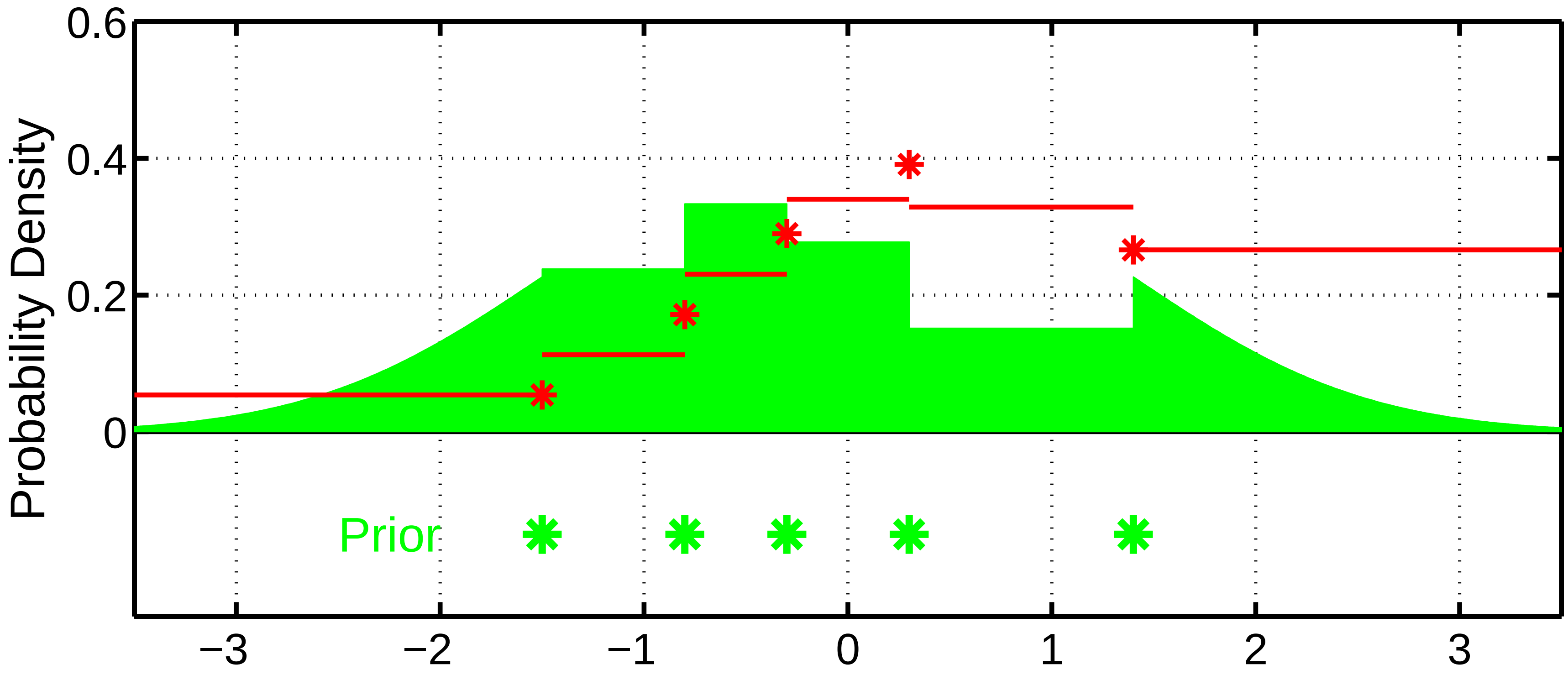
Likelihood on tails is constant value at outermost ensemble.

Also tends to increase spread.

Likelihood on tails generally unimportant.

RMS still nearly the same.

Call this a Non-parametric Rank Histogram Filter (NPRHF)



Only need value of likelihood computed for each ensemble member.
Can handle arbitrary non-gaussian likelihoods.
Can help to keep bounded quantities bounded.

A nonlinear, non-gaussian filter

Have state vector, x.

Sets of observations at discrete times, t_i.

Complete history of observations is: $Y_\tau = \{y_i;\ t_i \le \tau\}$

Go back to Bayes, assimilating observations at time t_k:

$$p\left(x, t_k \middle| Y_{t_k}\right) = \frac{p(y_k | x)\, p(x, t_k | Y_{t_{k-1}})}{norm}$$

A nonlinear, non-gaussian filter

Update marginal for jth component of x independently:

$$p_j\left(x_j, t_k \middle| Y_{t_k}\right) = \frac{p(y_k|x)\,p_j(x_j, t_k|Y_{t_{k-1}})}{norm}$$

Just use the NPRHF.

Under what conditions is the marginal update a good approximation?

This works well in Lorenz-63.

Almost always better than EAKF and EnKF.

Significantly better for infrequent observations.

A nonlinear, non-gaussian filter

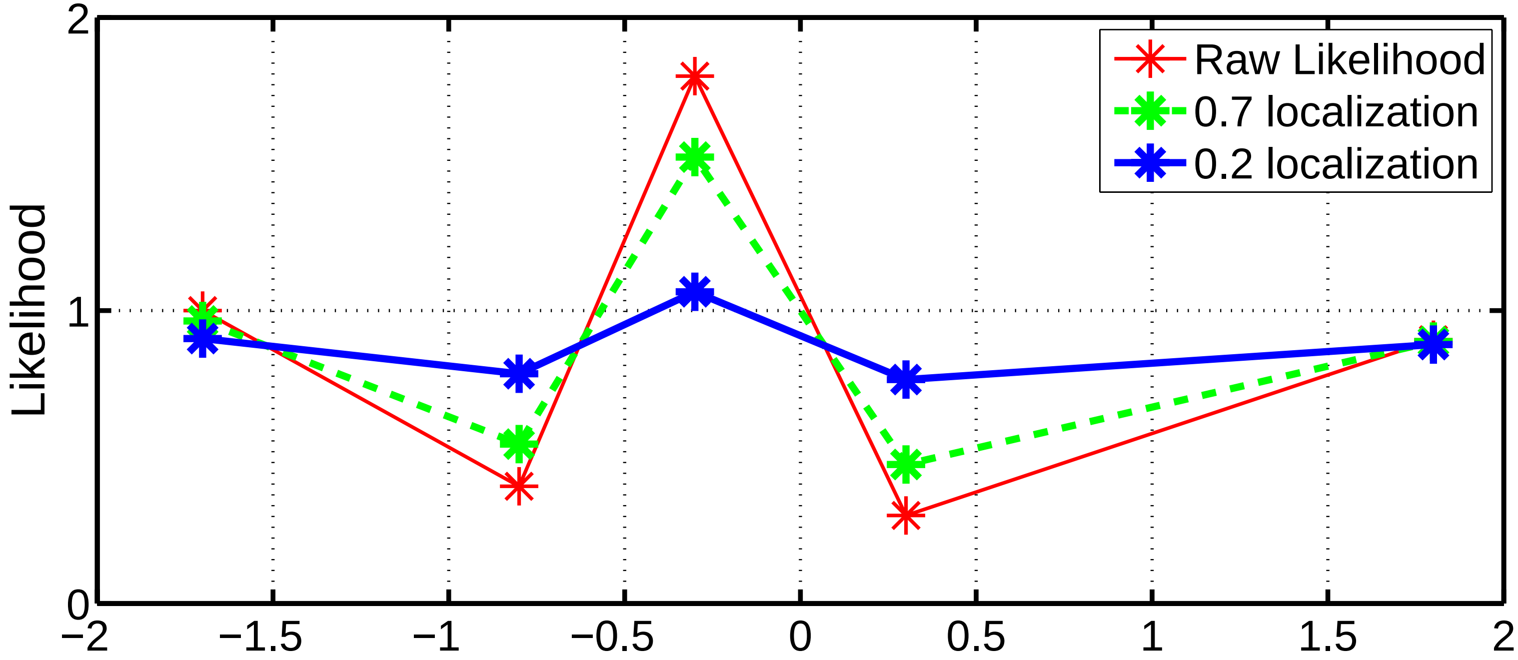
Challenge for large models:

Likelihood from remote ‘unrelated’ obs. is just noise for state.

Solution: Compute a ‘localized’ likelihood for each state variable:

Damp likelihood to ensemble mean, $\bar{L}$, as function of localization:

$$\hat{L}_n = \bar{L} + \alpha(L_n - \bar{L}), \quad n=1,...,ens\_size$$



A nonlinear, non-gaussian filter

Algorithmic summary:

1. Compute forward operators for set of observations.
2. Compute ensemble likelihoods for each observation.
3. Compute cumulative likelihood for each state variable:
    a. For each ensemble member, product of localized likelihoods.
    b. Assumes independent observational errors.
4. For each state variable, use NPRHP to update ensemble.

A nonlinear, non-gaussian filter

Very efficient:

1. No computation of covariances between priors.
2. Only one application of RHF update per state variable.

Assimilation is faster for standard atmospheric problems.

(Same scaling but constant factor smaller)

Scalable algorithm:

1. Forward operators/likelihoods all independent (parallel).
2. Products of likelihoods independent for all states/ensembles.
3. Can select subset of ‘best’ observations for each state.
4. Parallel requires NO extra computations.

But, does it work? Yes (and no).

For Lorenz-96, 80 member RMS about 20% greater than EAKF.
For GFDL dynamical core, PS observations, about 10% greater.

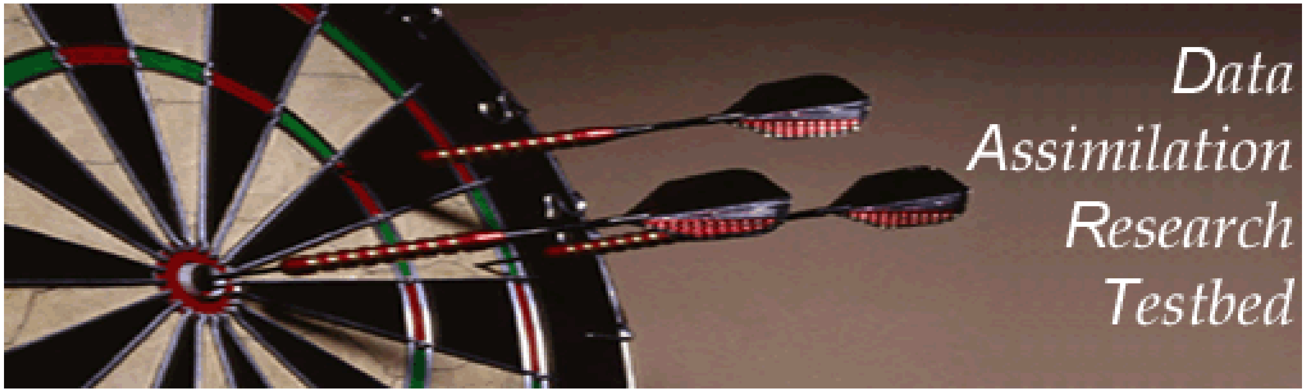
Why is it worse?

1. Doesn't know anything about likelihood on tails.
   This can hurt when everything is close to gaussian.
2. Noise from unrelated observations.
   Unrelated observations cause all ensembles to move.
   Smoothing likelihoods in clever ways should address this.

Want to try it out?

The Rank Histogram Filter and 7 other ensemble update variants are in DART.

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



The nonlinear non-gaussian filter is not yet in public releases.