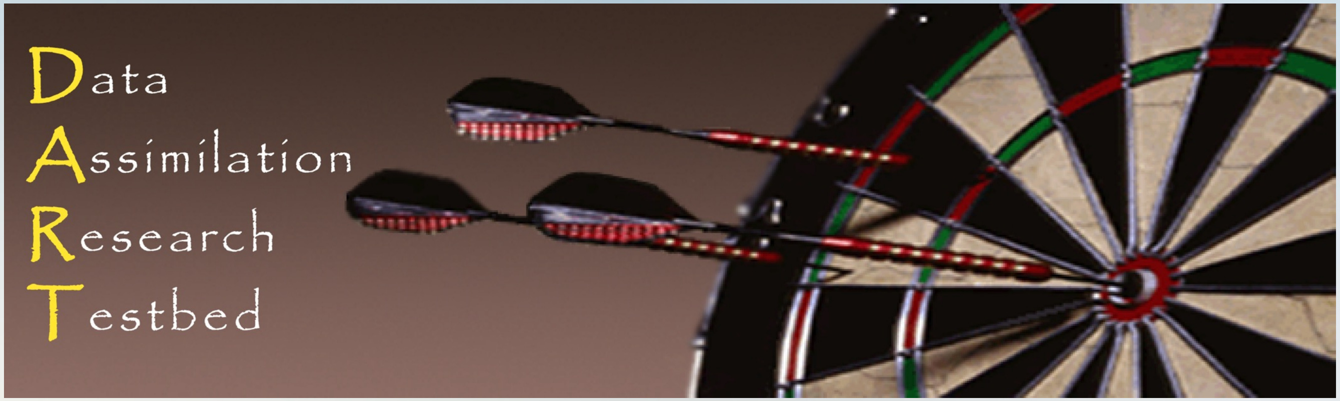
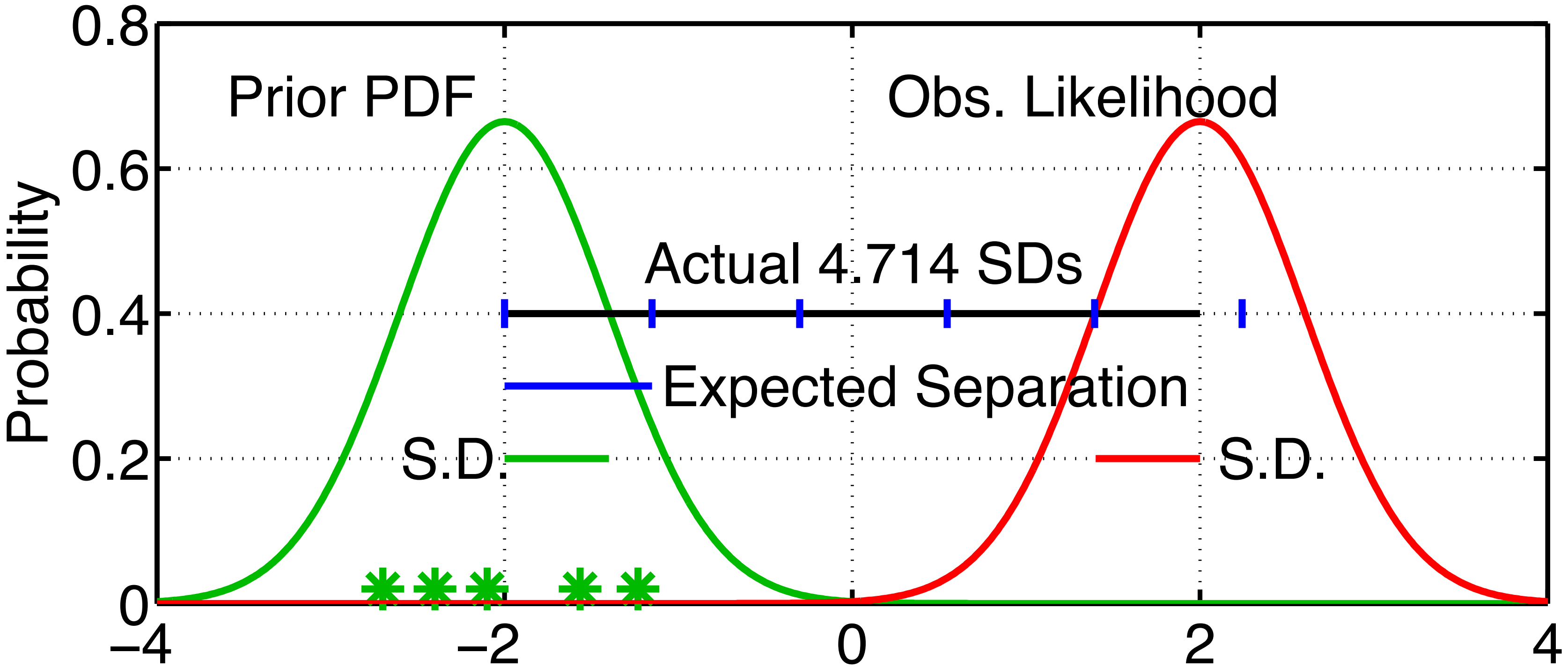


DART_LAB Tutorial Section 5: Adaptive Inflation

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

Variance inflation for observations: An adaptive error tolerant filter

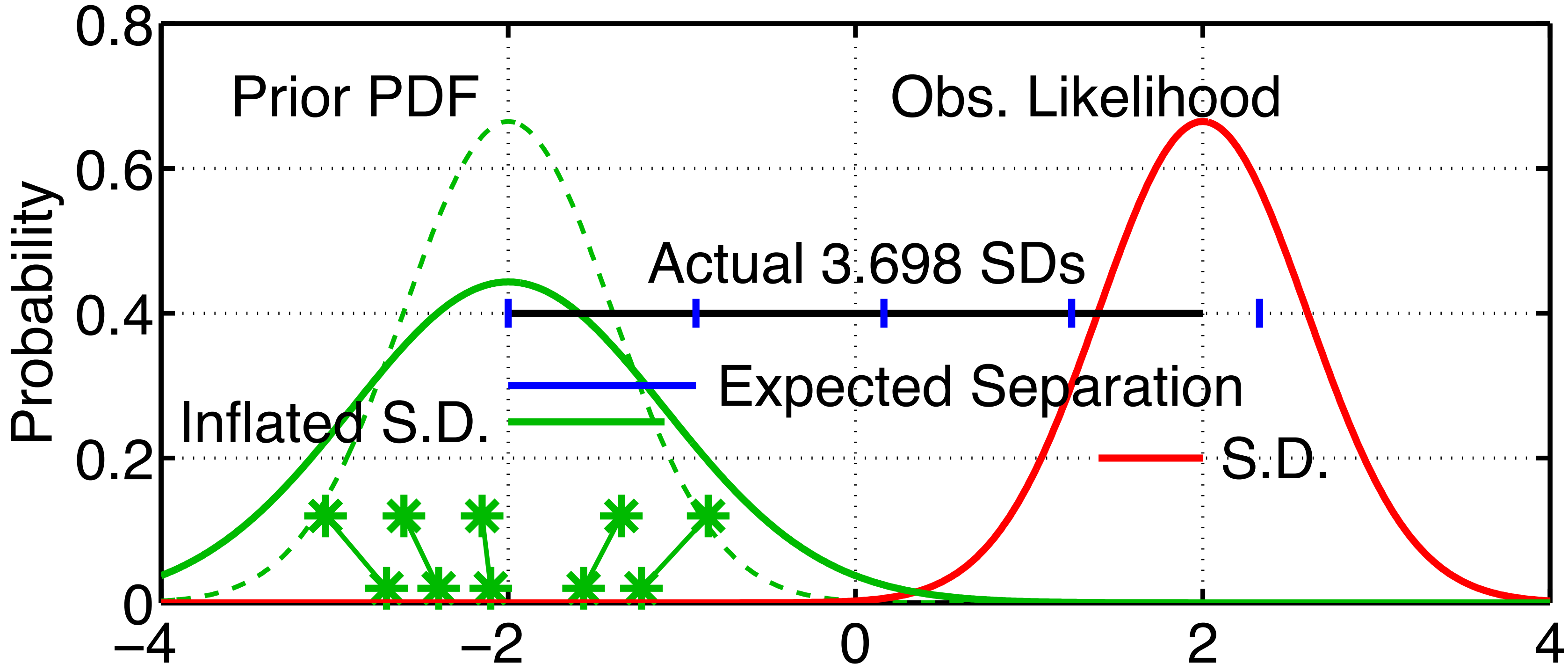


1. For observed variable, have estimate of prior-observed inconsistency.
2. Expected (prior_mean – observation) = $\sqrt{\sigma_{prior}^2 + \sigma_{obs}^2}$

   Assumes that prior and observation are supposed to be unbiased.

   Is it model error or random chance?

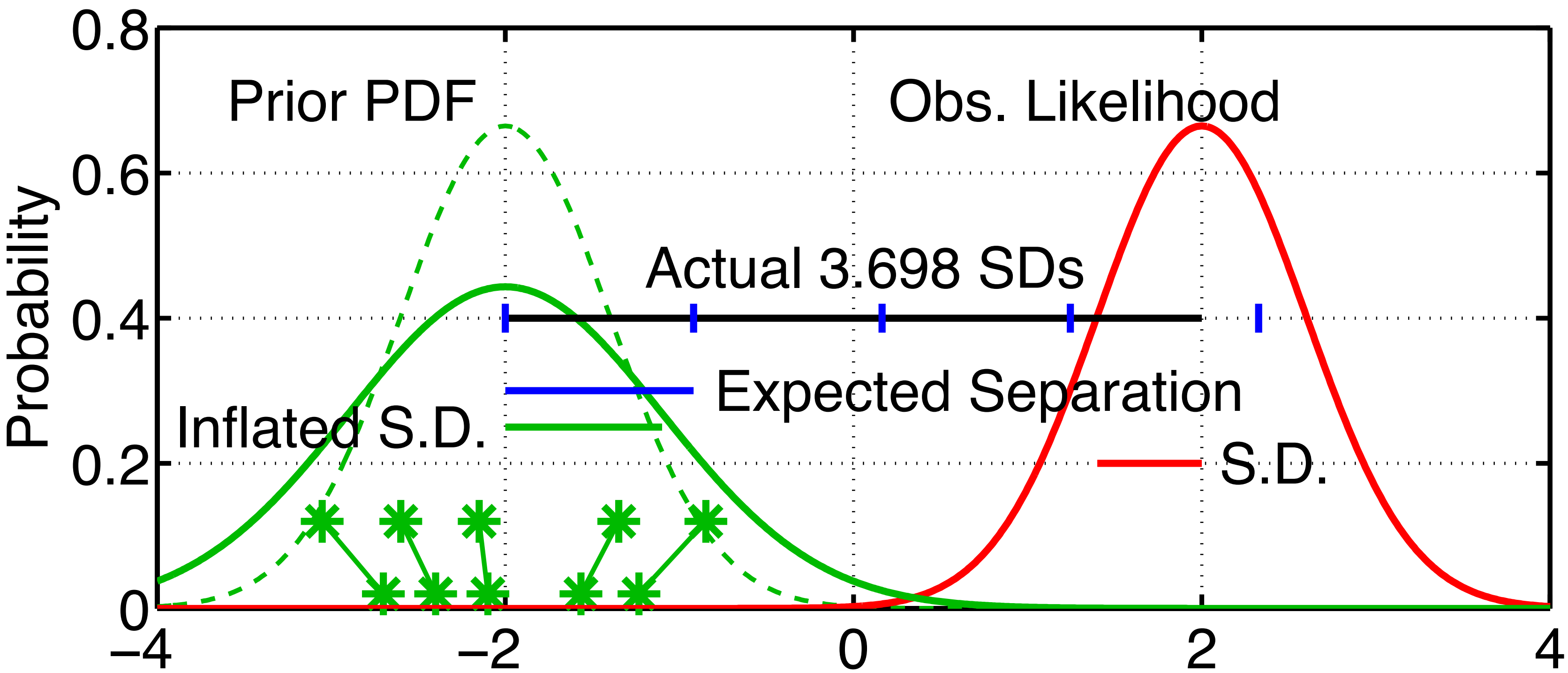
Variance inflation for observations: An adaptive error tolerant filter



1. For observed variable, have estimate of prior-observed inconsistency.
2. Expected (prior_mean – observation) = $\sqrt{\sigma^2_{prior} + \sigma^2_{obs}}$
3. Inflating increases expected separation

Increases 'apparent' consistency between prior and observation.

Variance inflation for observations: An adaptive error tolerant filter

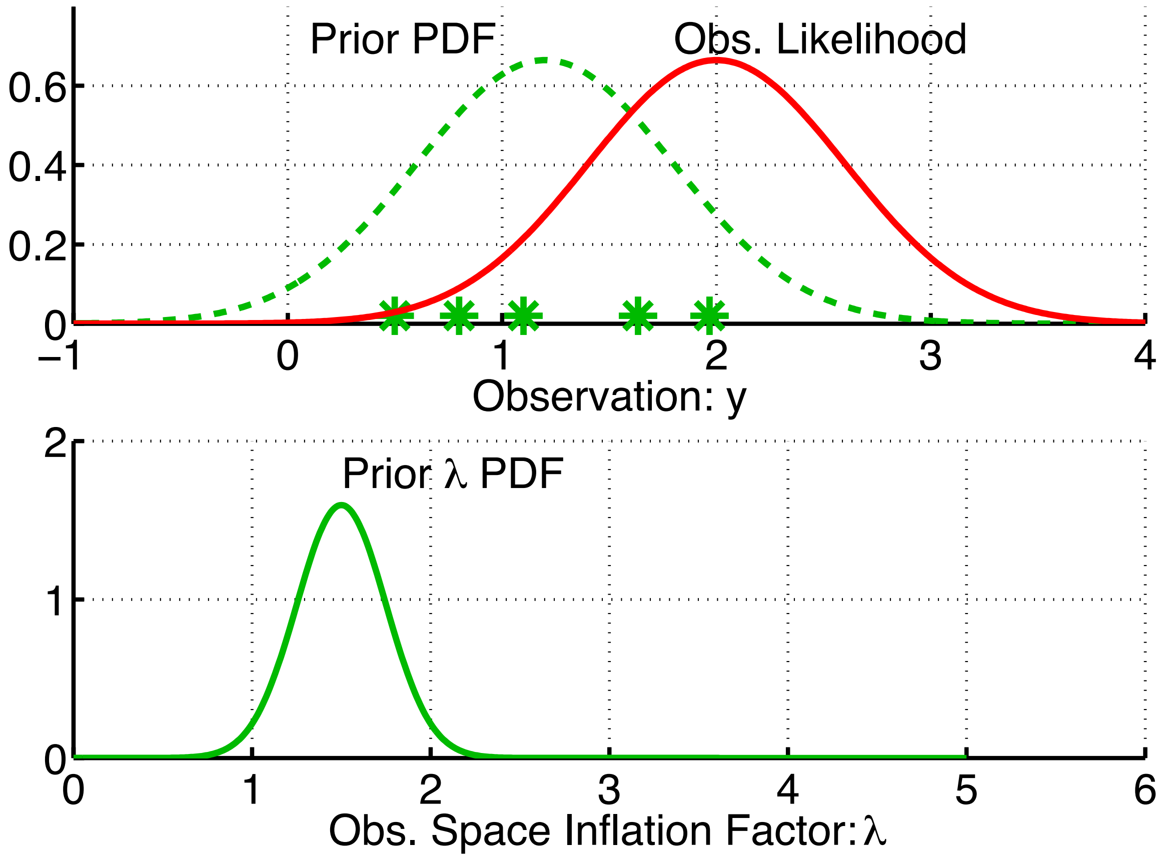


Distance D from prior mean y to obs is $N\left(0,\sqrt{\lambda\sigma_{prior}^2+\sigma_{obs}^2}\right)=N(0,\theta)$

Prob y_0 is observed given λ: $p(y_o\mid\lambda)=\left(2\pi\theta^2\right)^{-1/2}\exp\left(-D^2/2\theta^2\right)$

Variance inflation for observations: An adaptive error tolerant filter

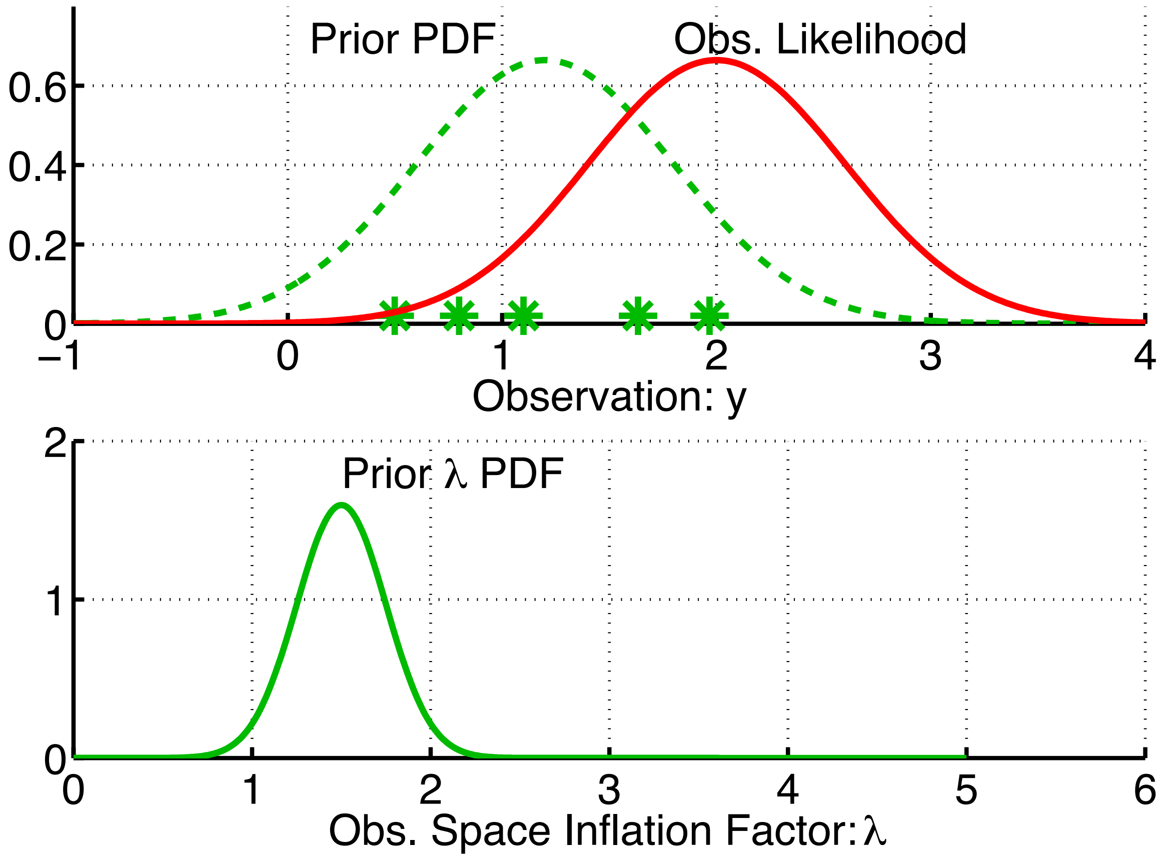
Use Bayesian statistics to get estimate of inflation factor λ.



Assume prior is gaussian: $p\left(\lambda, t_k \mid Y_{t_{k-1}}\right) = N\left(\bar{\lambda}_p, \sigma^2_{\lambda,p}\right)$

Variance inflation for observations: An adaptive error tolerant filter

Use Bayesian statistics to get estimate of inflation factor λ.



We've assumed a gaussian for prior.

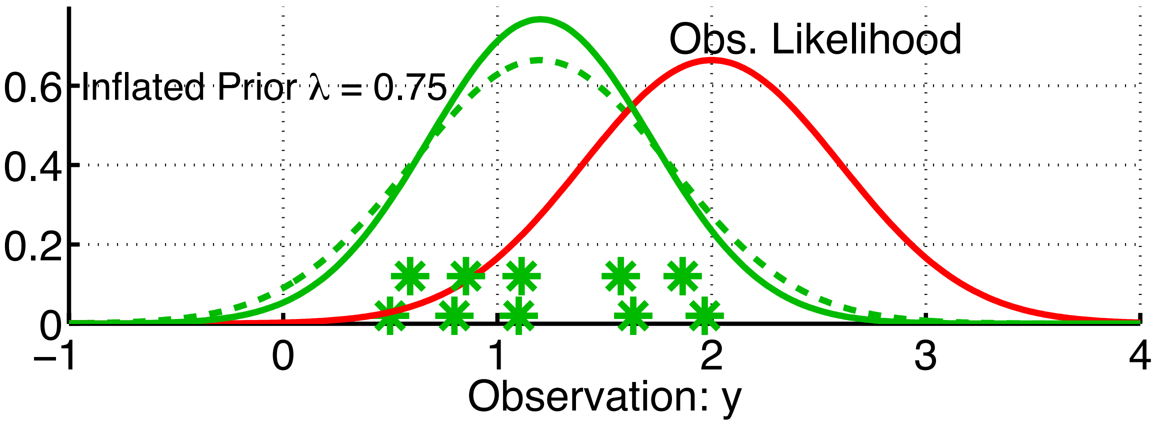
$p\left(\lambda, t_k \mid Y_{t_{k-1}}\right)$

Recall that $p\left(y_k \mid \lambda\right)$ can be evaluated From normal PDF.

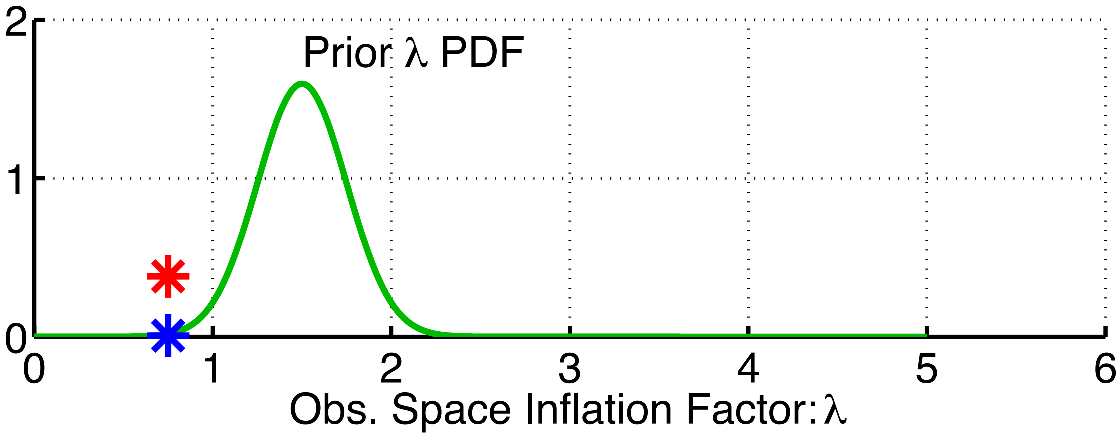
$$p\left(\lambda, t_k \mid Y_{t_k}\right) = p\left(y_k \mid \lambda\right) p\left(\lambda, t_k \mid Y_{t_{k-1}}\right) / normalization$$

Variance inflation for observations: An adaptive error tolerant filter

Use Bayesian statistics to get estimate of inflation factor λ.



Get $p(y_k \mid \lambda = 0.75)$ from normal PDF.

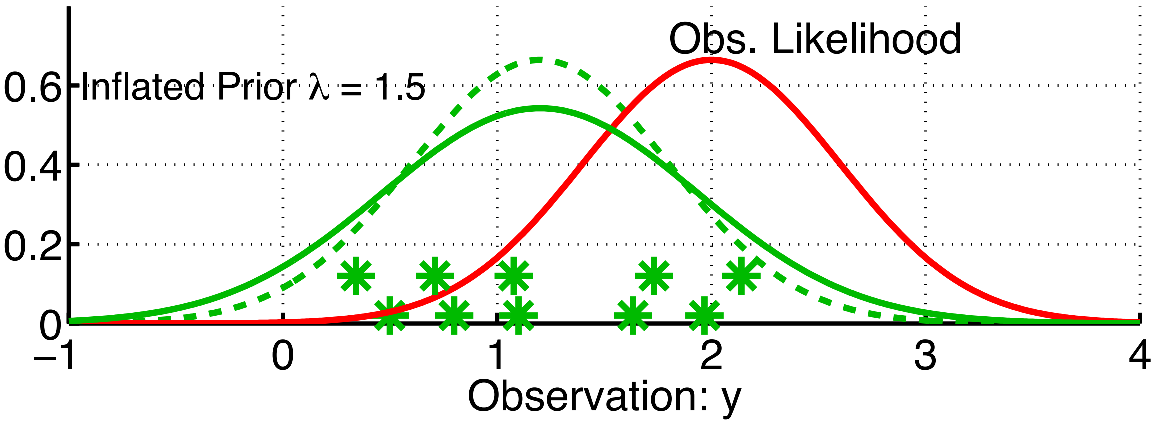


Multiply by $p(\lambda = 0.75, t_k \mid Y_{t_{k-1}})$

to get $p(\lambda = 0.75, t_k \mid Y_{t_k})$

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

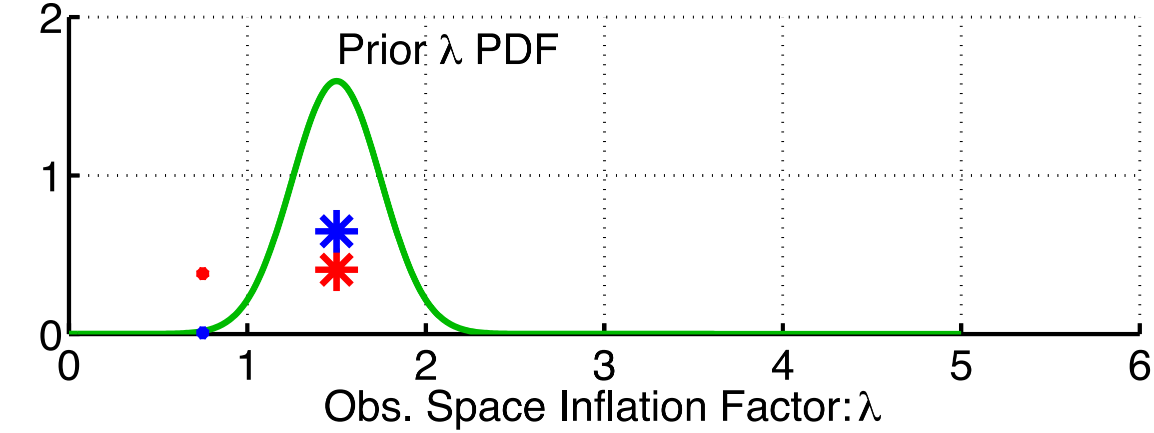
Use Bayesian statistics to get estimate of inflation factor λ.



Get $p\left(y_k \mid \lambda = 1.50\right)$ from normal PDF.

Multiply by $p\left(\lambda = 1.50, t_k \mid Y_{t_{k-1}}\right)$

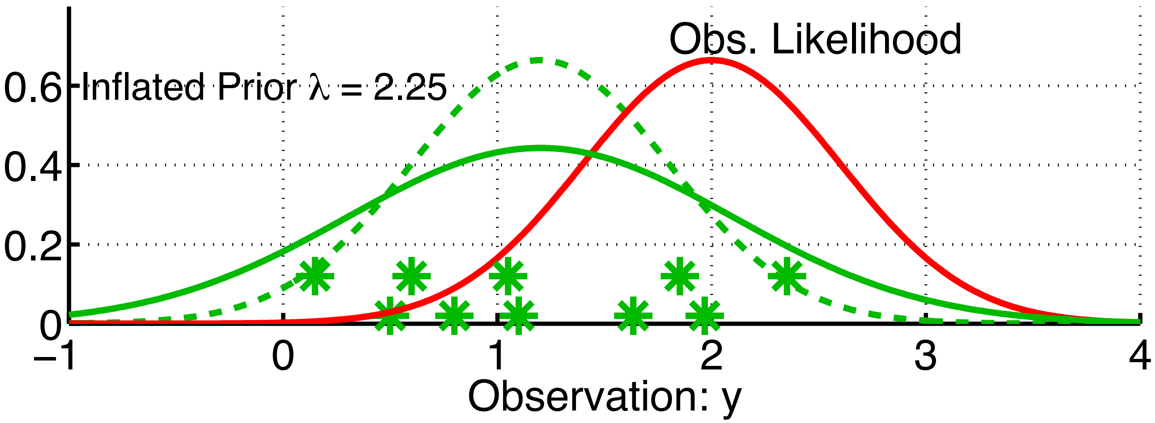
to get $p\left(\lambda = 1.50, t_k \mid Y_{t_k}\right)$



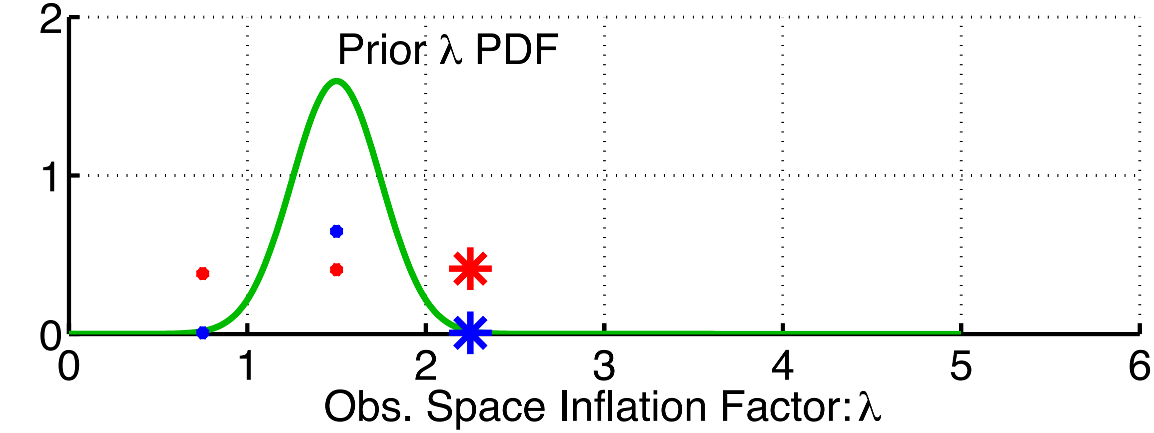
$$p\left(\lambda, t_k \mid Y_{t_k}\right) = p\left(y_k \mid \lambda\right) p\left(\lambda, t_k \mid Y_{t_{k-1}}\right) / normalization$$

Variance inflation for observations: An adaptive error tolerant filter

Use Bayesian statistics to get estimate of inflation factor λ.



Get $p(y_k \mid \lambda = 2.25)$ from normal PDF.



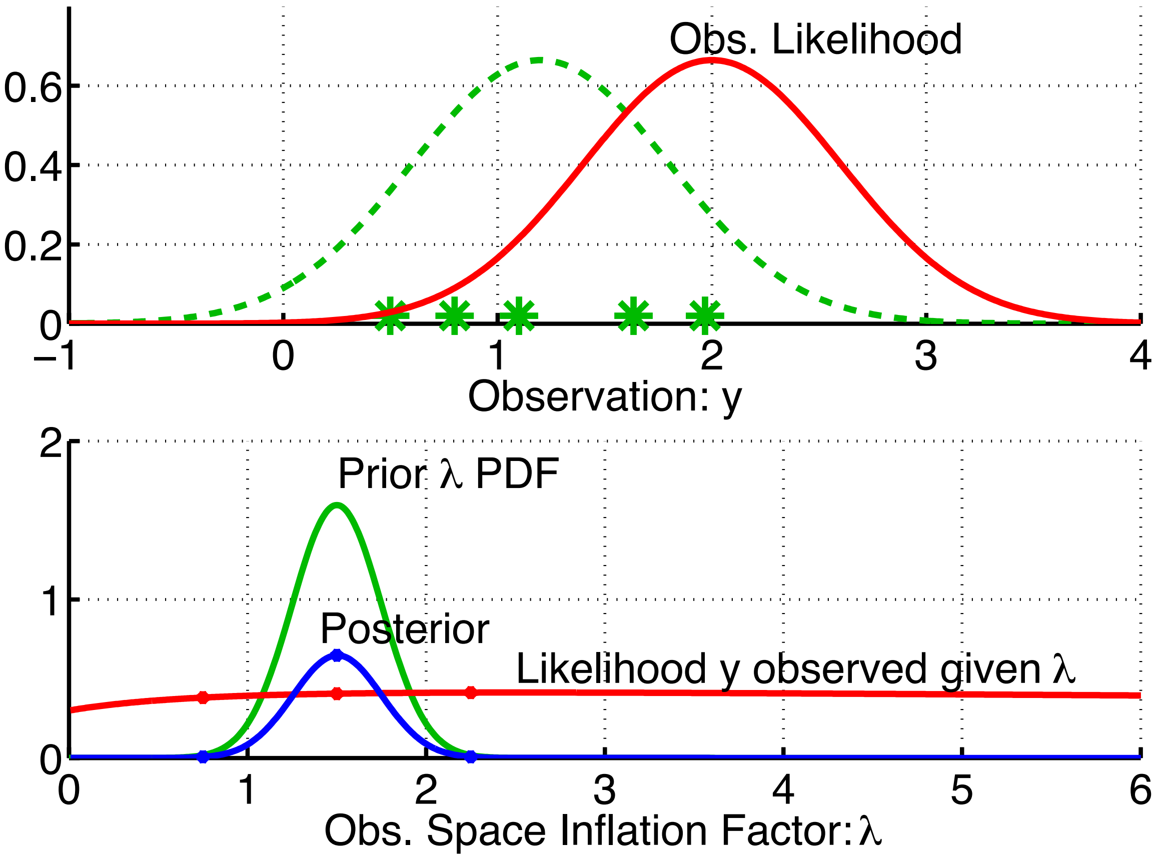
Multiply by $p(\lambda = 2.25, t_k \mid Y_{t_{k-1}})$

to get $p(\lambda = 2.25, t_k \mid Y_{t_k})$

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

Variance inflation for observations: An adaptive error tolerant filter

Use Bayesian statistics to get estimate of inflation factor λ.



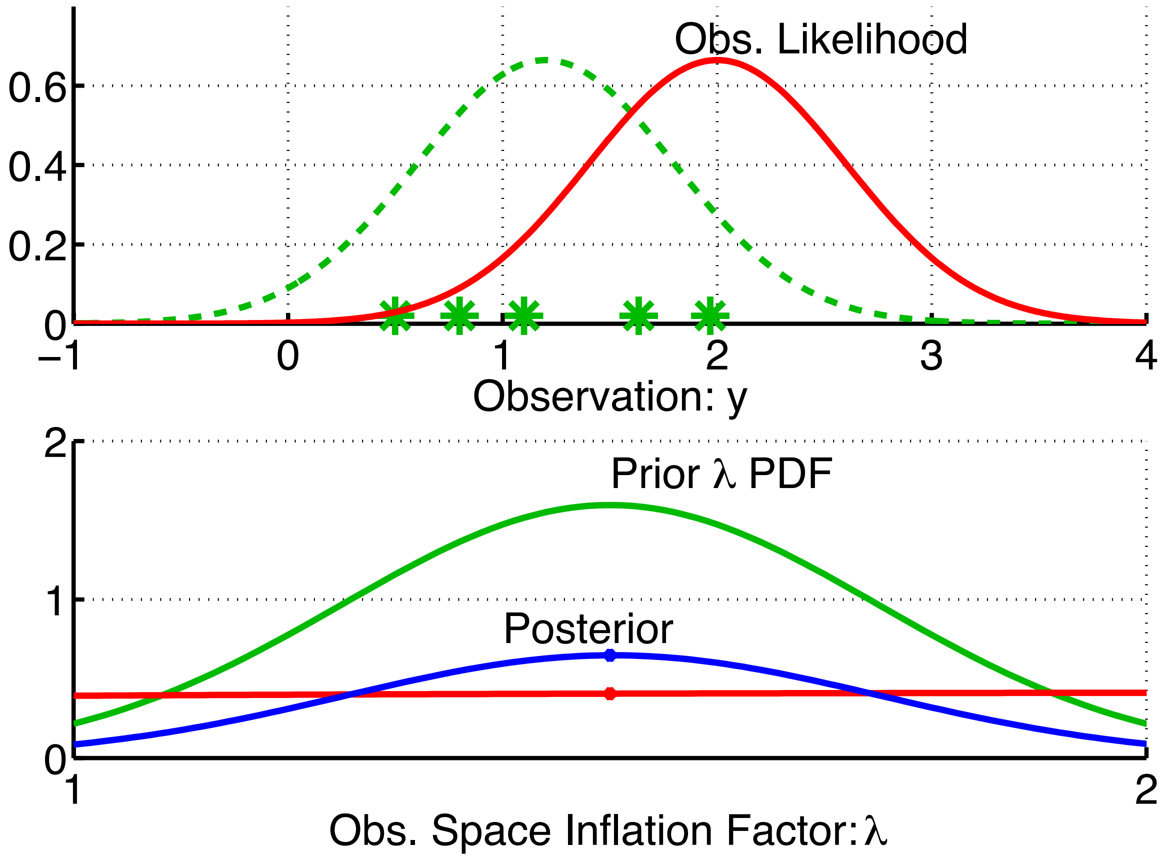
Repeat for a range of values of λ.

Now must get posterior in same form as prior (gaussian).

$$p\left(\lambda, t_k \mid Y_{t_k}\right) = p\left(y_k \mid \lambda\right) p\left(\lambda, t_k \mid Y_{t_{k-1}}\right) / normalization$$

Variance inflation for observations: An adaptive error tolerant filter

Use Bayesian statistics to get estimate of inflation factor λ.



Very little information about λ in a single observation.

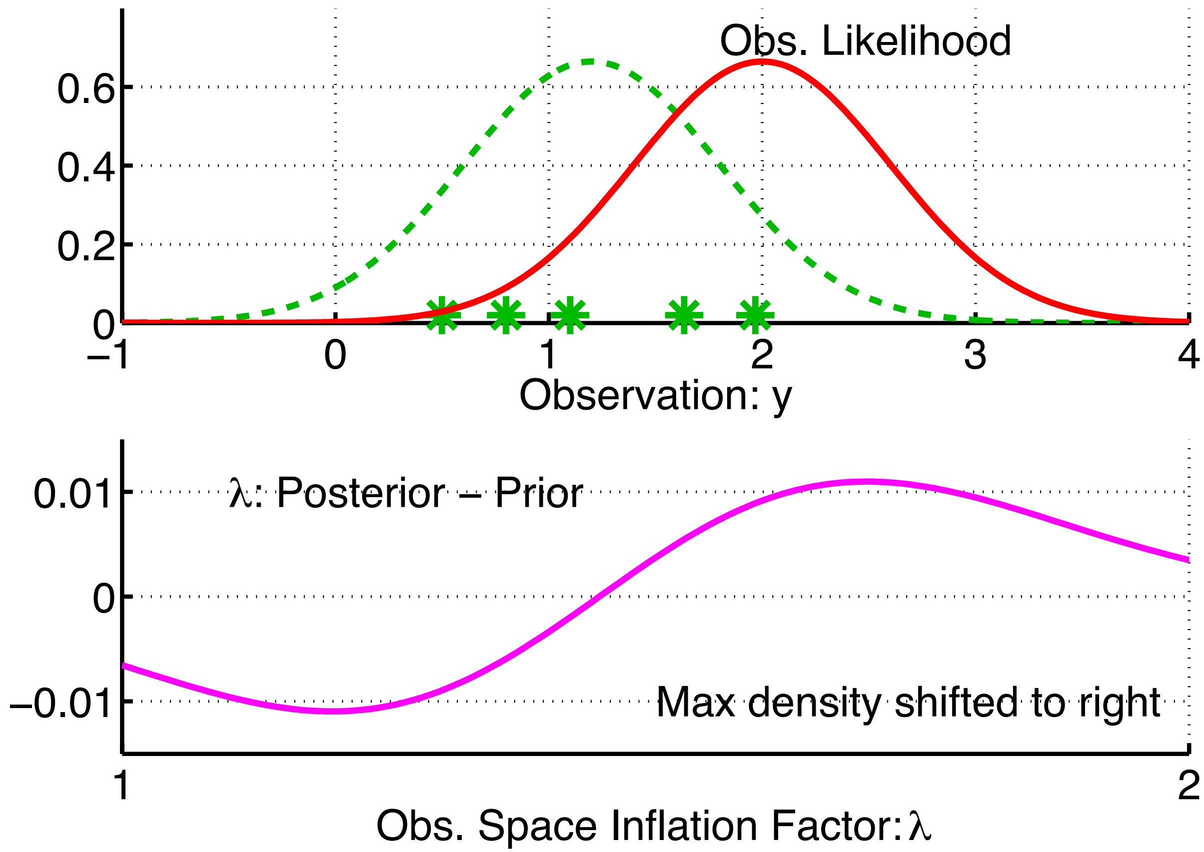
Posterior and prior are very similar.

Normalized posterior indistinguishable from prior.

$$p\left(\lambda, t_k \mid Y_{t_k}\right) = p\left(y_k \mid \lambda\right) p\left(\lambda, t_k \mid Y_{t_{k-1}}\right) / normalization$$

Variance inflation for observations: An adaptive error tolerant filter

Use Bayesian statistics to get estimate of inflation factor λ.



Very little information about λ in a single observation.

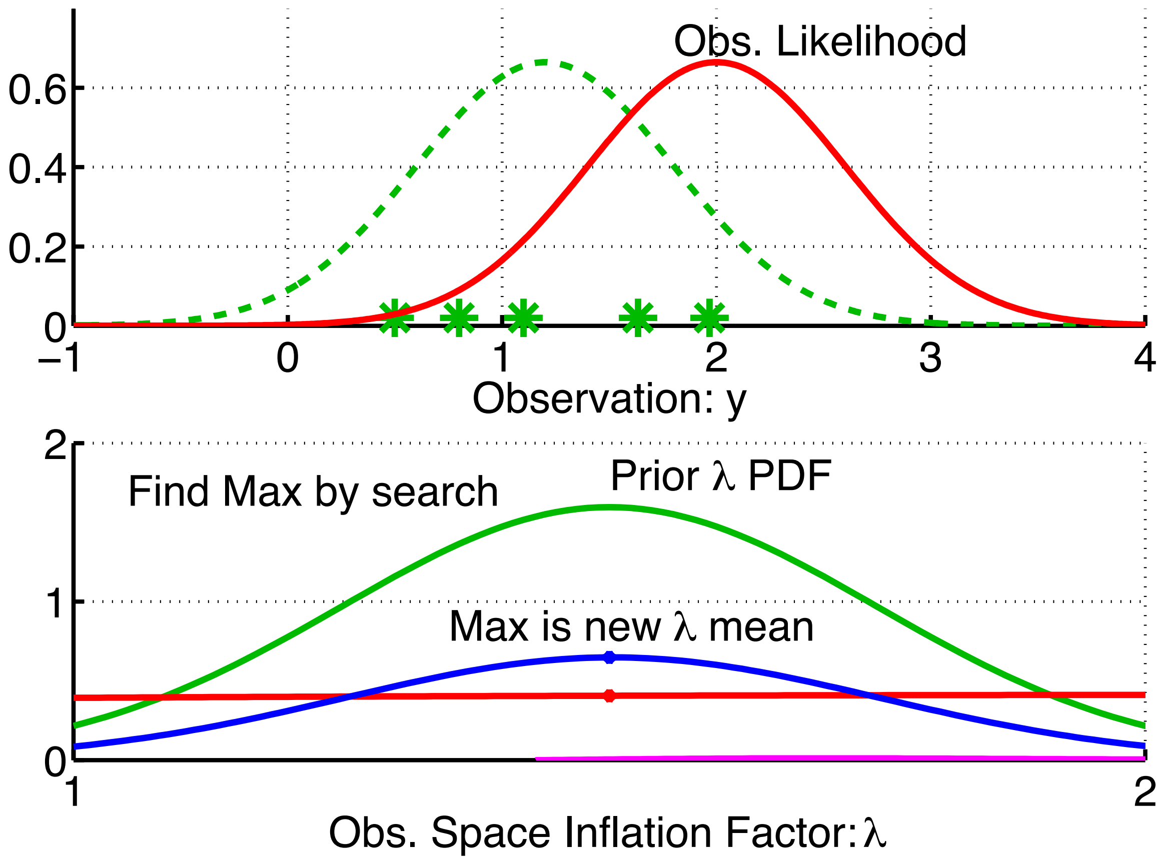
Posterior and prior are very similar.

Difference shows slight shift to larger values of λ.

$$p\left(\lambda, t_k \mid Y_{t_k}\right) = p\left(y_k \mid \lambda\right) p\left(\lambda, t_k \mid Y_{t_{k-1}}\right) / normalization$$

Variance inflation for observations: An adaptive error tolerant filter

Use Bayesian statistics to get estimate of inflation factor λ.



One option is to use Gaussian prior for λ.

Select max (mode) of posterior as mean of updated Gaussian.

Do a fit for updated standard deviation.

$$p\left(\lambda, t_k \mid Y_{t_k}\right) = p\left(y_k \mid \lambda\right) p\left(\lambda, t_k \mid Y_{t_{k-1}}\right) / normalization$$

Variance inflation for observations: An adaptive error tolerant filter

A. Computing updated inflation mean, $\bar{\lambda}_u$.

Mode of $p(y_k \mid \lambda)\, p(\lambda, t_k \mid Y_{t_{k-1}})$ can be found analytically!

Solving $\partial\left[p(y_k \mid \lambda)\, p(\lambda, t_k \mid Y_{t_{k-1}})\right] / \partial y = 0$ leads to 6th order poly in θ.

This can be reduced to a cubic equation and solved to give mode.

New $\bar{\lambda}_u$ is set to the mode.

This is relatively cheap compared to computing regressions .

Variance inflation for observations: An adaptive error tolerant filter

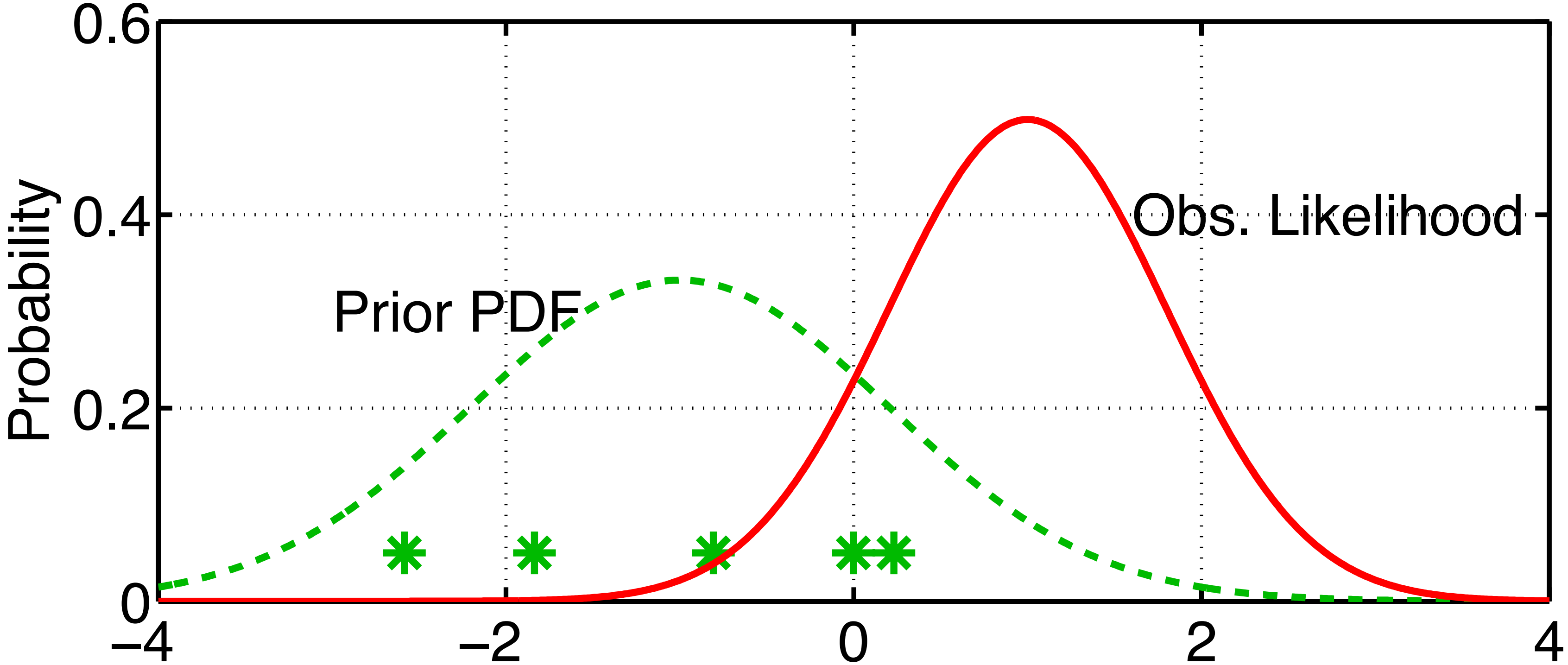
B. Computing updated inflation variance, $\sigma^2_{\lambda,u}$.

1. Evaluate numerator at mean $\bar{\lambda}_u$ and second point, e.g. $\bar{\lambda}_u + \sigma_{\lambda,p}$

2. Find $\sigma^2_{\lambda,u}$ so $N\left(\bar{\lambda}_u, \sigma^2_{\lambda,u}\right)$ goes through $p\left(\bar{\lambda}_u\right)$ and $p\left(\bar{\lambda}_u + \sigma_{\lambda,p}\right)$

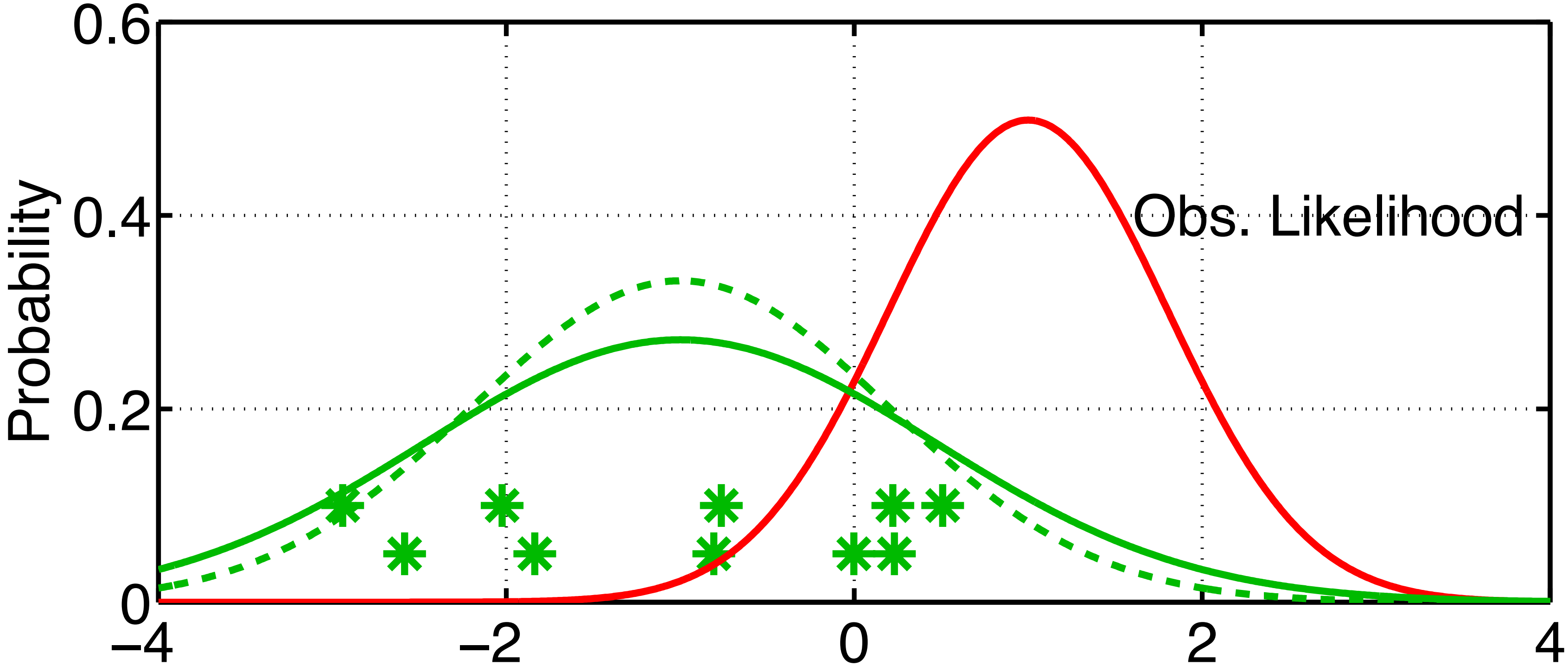
3. Compute as $\sigma^2_{\lambda,u} = -\sigma^2_{\lambda,p} / 2\ln r$ where $r = p\left(\bar{\lambda}_u + \sigma_{\lambda,p}\right) / p\left(\bar{\lambda}_u\right)$

Single Variable Computations with Adaptive Error Correction



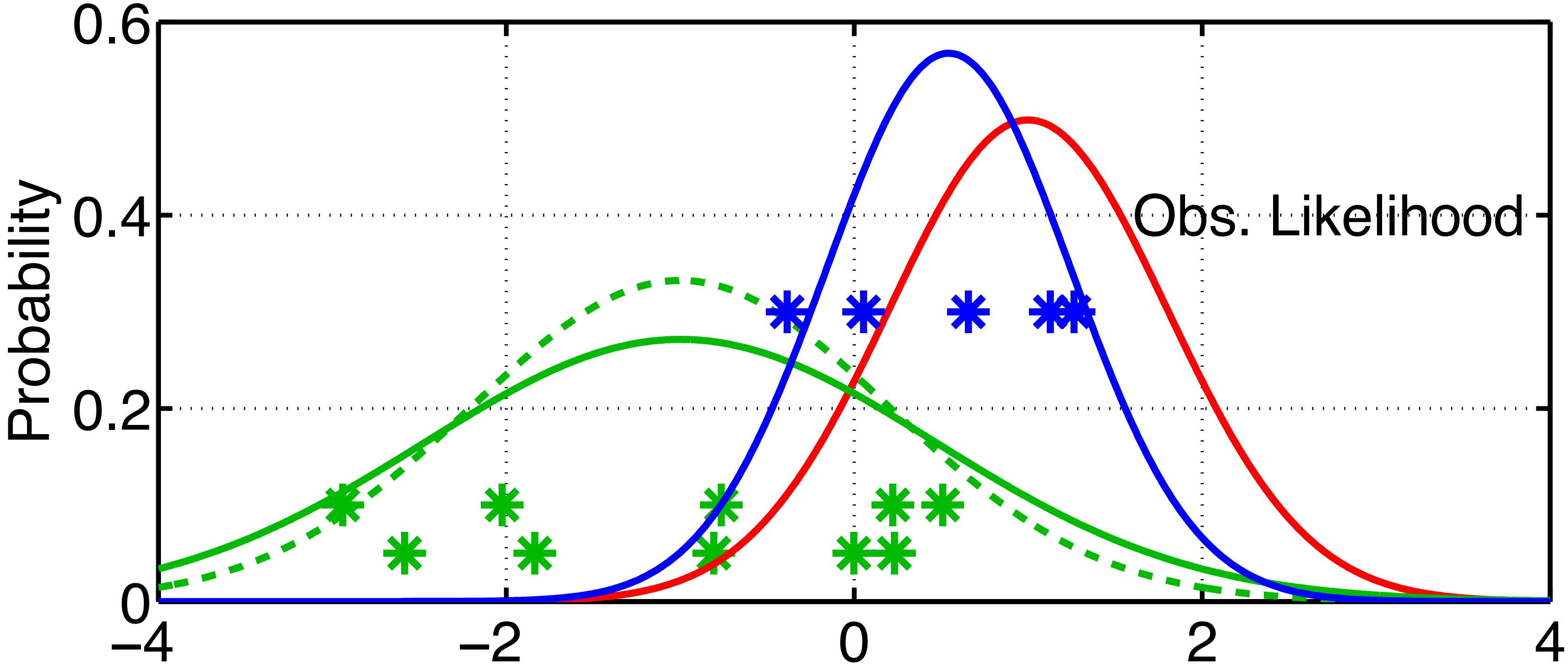
1. Compute updated inflation distribution, $p\left(\lambda, t_k \mid Y_{t_k}\right)$.

Single Variable Computations with Adaptive Error Correction



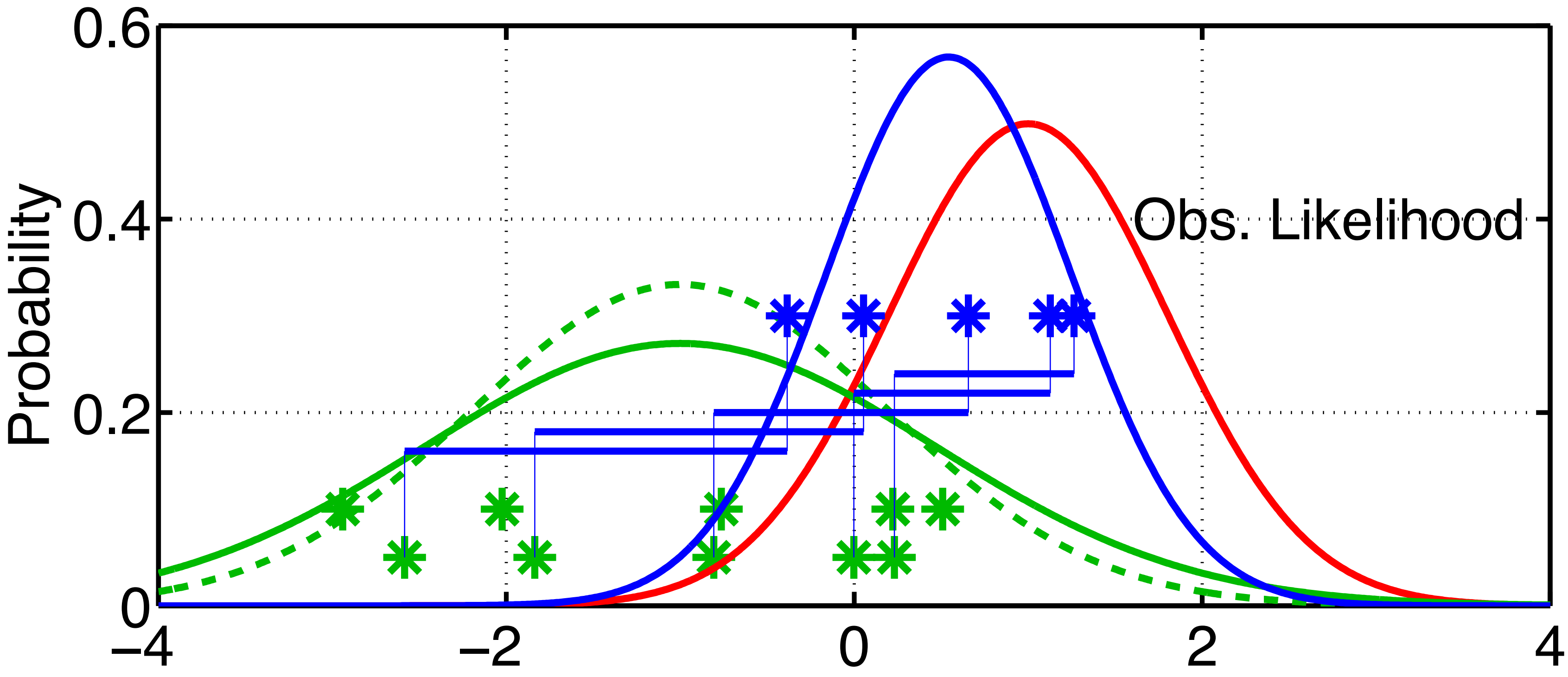
1. Compute updated inflation distribution, $p\left(\lambda, t_k \mid Y_{t_k}\right)$.
2. Inflate ensemble using mean of updated λ distribution.

Single Variable Computations with Adaptive Error Correction



1. Compute updated inflation distribution, $p\left(\lambda, t_k \mid Y_{t_k}\right)$.
2. Inflate ensemble using mean of updated λ distribution.
3. Compute posterior for y using inflated prior.

Single Variable Computations with Adaptive Error Correction



1. Compute updated inflation distribution, $p\left(\lambda, t_k \mid Y_{t_k}\right)$.
2. Inflate ensemble using mean of updated λ distribution.
3. Compute posterior for y using inflated prior.
4. Compute increments from ORIGINAL prior ensemble.

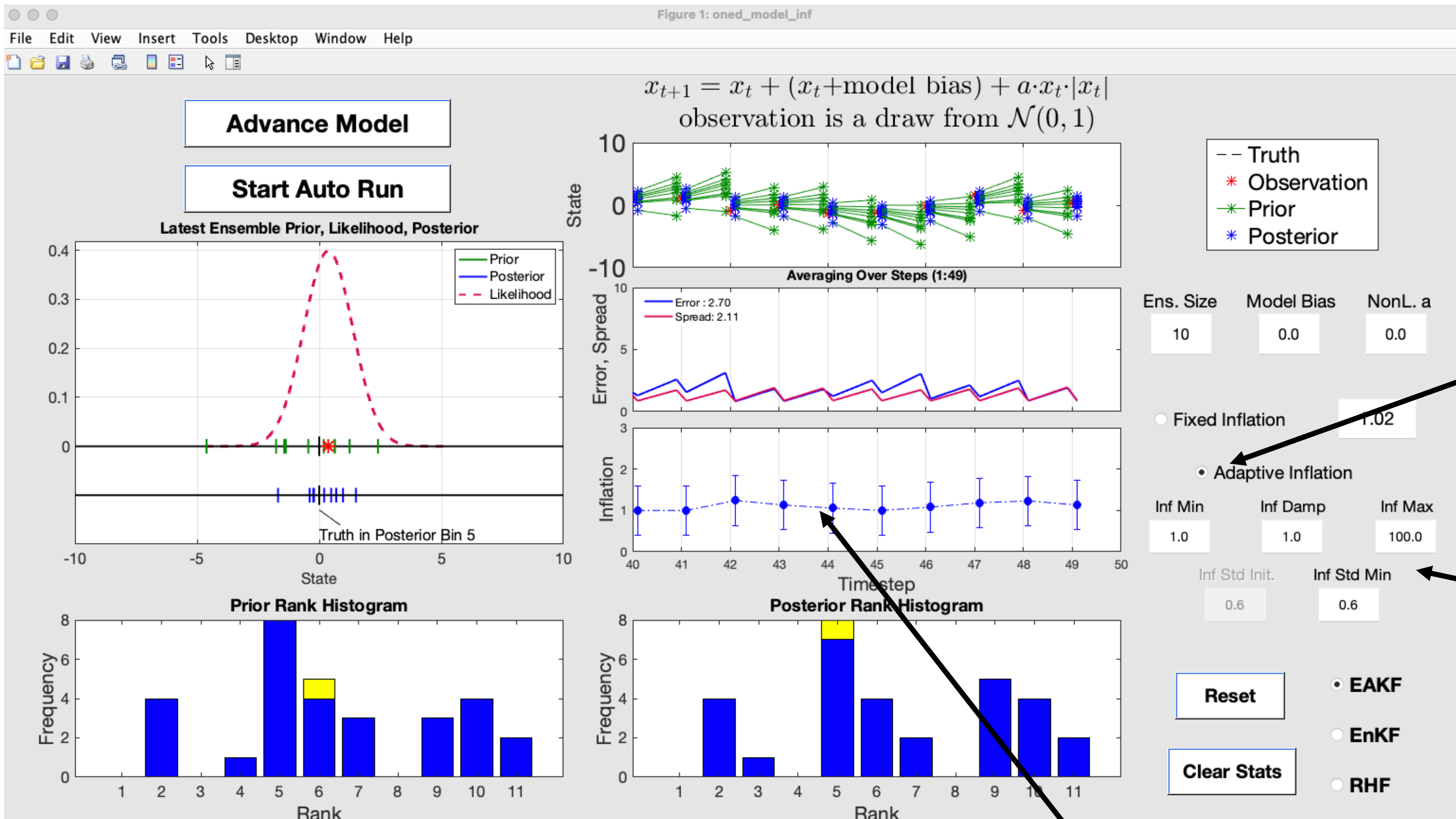
Matlab Hands-on: oned_model_inf

Adaptive inflation can be tested with Matlab script *oned_model_inf.m*

Can explore 5 different values that control adaptive inflation:

- Minimum value of inflation, often set to 1 (no deflation).
- Inflation damping, more on this later. Value of 1.0 turns it off.
- Maximum value of inflation.
- The initial value of the inflation standard deviation.
- A lower bound on inflation standard deviation (it will asymptote to zero if allowed).

Matlab Hands-on: oned_model_inf



Turn on adaptive inflation.

Inflation settings.

Inflation value and its standard deviation.

Matlab Hands-on: oned_model_inf

Oned_model_inf generates a summary file each time statistics are reset.

File *oned_model_inf.log* keeps track of parameter settings and metrics.

```
# Time step: 1 (Initial configuration)
  - Ensemble size = 4
  - Model bias = 0.00
  - Nonlinear `a` parameter = 0.00
  - Inflation value = 1.02
  - (Adaptive) Inflation lower bound = 1.00
  - (Adaptive) Inflation upper bound = 100.00
  - (Adaptive) Inflation damping factor = 1.00
  - (Adaptive) Inflation standard deviation = 0.60
  - (Adaptive) Inflation standard deviation lower bound = 0.60

# Time step: 19
  >> Statistics over period (1:19): avg. RMSE = 3.03, avg. Spread = 2.19
  $$ User input: Ensemble size has been changed from 4 to 10
```

Matlab Hands-on: oned_model_inf

Try adaptive inflation.

Pick a lower value for standard deviation.
Initial lower bound on inflation of 1.0, upper bound large (100).

Try introducing a model bias.

What happens if lower bound is less than 1?

Inflation Damping

Inflation mean damped towards 1 every assimilation time.

- *inf_damping* 0.9: 90% of the inflation difference from 1.0 is retained.

Can be useful in models with heterogeneous observations in time.
For instance, a well-observed hurricane crosses a model domain.
Adaptive inflation increases along hurricane track.
After hurricane, fewer observations, no longer need so much inflation.

For large earth system models, following values may work:

inf_sd_initial = 0.6,
inf_damping = 0.9,
inf_sd_lower_bound = 0.6.

Adaptive Multivariate Inflation Algorithm

Suppose we want a global multivariate inflation, λ_s, instead.

Make same least squares assumption that is used in ensemble filter.

Inflation of λ_s for state variables inflates obs. priors by same amount.

Get same likelihood as before: $p(y_o \mid \lambda) = \left(2\pi\theta^2\right)^{-1/2} \exp\left(-D^2 / 2\theta^2\right)$

$$\theta = \sqrt{\lambda_s \sigma_{prior}^2 + \sigma_{obs}^2}$$

Compute updated distribution for λ_s exactly as for single observed variable.

Implementation of Adaptive Multivariate Inflation Algorithm

1. Apply inflation to state variables with mean of λ_s distribution.

2. Do following for observations at given time sequentially:
    a. Compute forward operator to get prior ensemble.
    b. Compute updated estimate for λ_s mean and variance.
    c. Compute increments for prior ensemble.
    d. Regress increments onto state variables.

Spatially varying adaptive inflation algorithm

Have a distribution for λ for each state variable, $\lambda_{s,i.}$

Use prior correlation from ensemble to determine impact of $\lambda_{s,i}$ on prior variance for given observation.

If γ is correlation between state variable i and observation then

$$\theta = \sqrt{\left[1+\gamma\left(\sqrt{\lambda_{s,i}}-1\right)\right]^2 \sigma_{prior}^2 + \sigma_{obs}^2}$$

Equation for finding mode of posterior is now full 12th order:

Analytic solution appears unlikely.

Can do Taylor expansion of θ around $\lambda_{s,i}$.

Retaining linear term is normally quite accurate.

There is an analytic solution to find mode of product in this case!

Matlab Hands-on: run_Lorenz_96_inf

Spatially Varying Adaptive inflation can be tested with Matlab script *run_lorenz96_inf.m*

Can explore 3 different values that control adaptive inflation:

- Minimum value of inflation, often set to 1 (no deflation).
- Inflation damping. Value of 1.0 turns it off.
- The value of the inflation standard deviation.
    - Lower bound on standard deviation is set to same value.
    - In this case, standard deviation just stays fixed at selected value.

Matlab Hands-on: run_Lorenz_96_inf

Adaptive inflation controls.

Adaptive inflation spatial plot.

Matlab Hands-on: run_Lorenz_96_inf

Explore some of the following:

How does adaptive inflation change as localization is changed?
How does adaptive inflation change for different values of inflation standard deviation?
If the lower bound is smaller than 1, does deflation (inflation < 1) happen?

Matlab Hands-on: run_Lorenz_96_inf

run_Lorenz_96_inf generates a summary file each time statistics are reset.

File *run_lorenz_96_inf.log* keeps track of parameter settings and metrics.

```
# Time step: 63
  >> Statistics over period (35:63): avg. Prior RMSE = 1.25, avg. Prior
Spread = 1.32
  $$ User input: Assimilation Type has been changed from `EAKF` to `RHF`


     Current configuration:
     - Forcing `F` parameter = 8.00
     - Assimilation type is `RHF`
     - Observation Network is `1:40:1`
     - Ensemble size = 20
     - Localization = 0.30
     - Inflation Algorithm is `Gaussian`
     - Adaptive inflation lower bound = 1.00
     - Adaptive inflation upper bound = 5.00
     - Adaptive inflation damping factor = 0.90
     - Adaptive inflation standard deviation = 0.60
     - Adaptive inflation standard deviation lower bound = 0.60
```

Simulating Model Error in 40-Variable Lorenz-96 Model

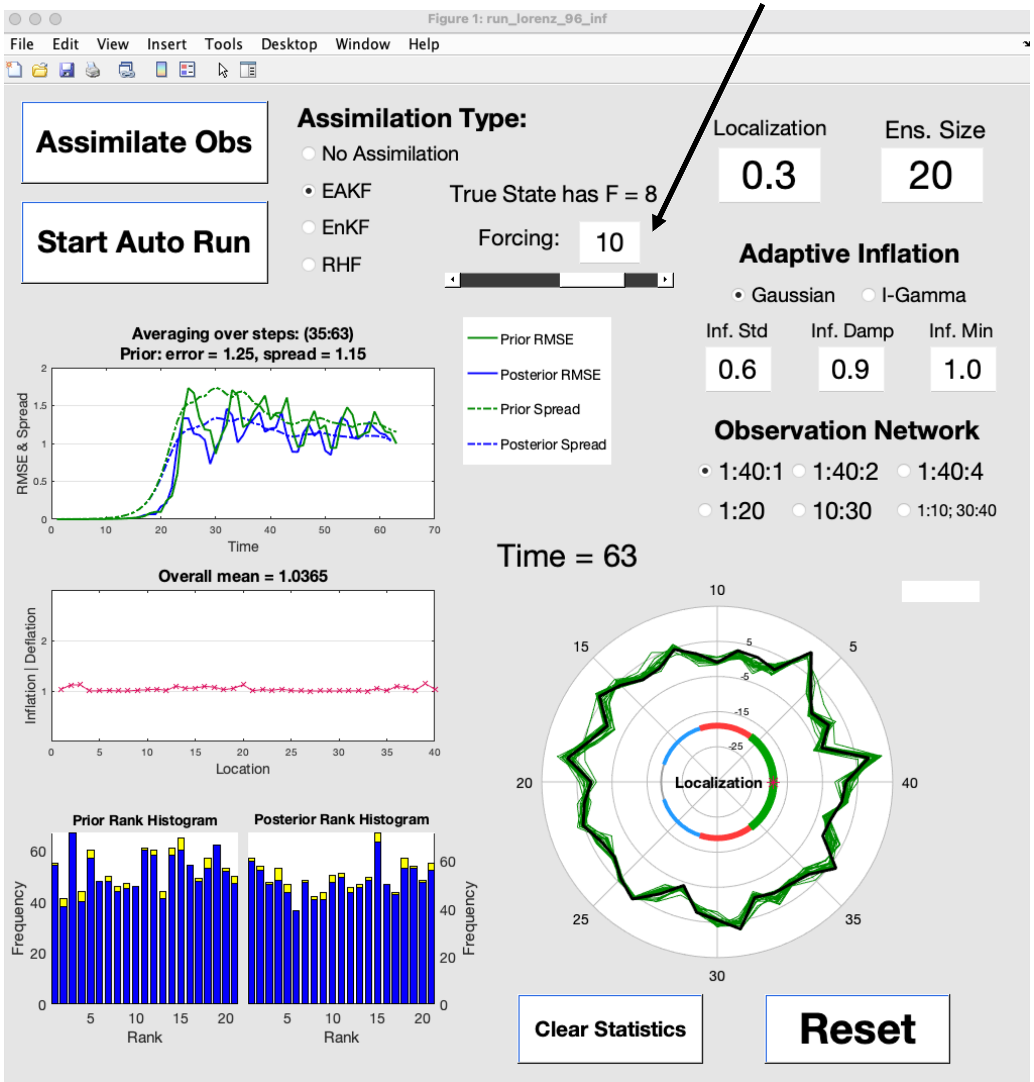
Inflation can deal with all sorts of errors, including model error.

Can simulate model error in Lorenz96 by changing forcing.
Synthetic observations are from model with forcing = 8.0.

Both *run_lorenz_96* and *run_lorenz_96_inf* allow model error.

Spatially Varying Adaptive Inflation

Model error: Change forcing for assimilating model here.



Spatially Varying Adaptive Inflation with Model Error

Explore some of the following:

Change the model forcing to a larger or smaller value (say 6 or 10).

How does adaptive inflation respond to model error?
Do good values of localization change as model error increases?

Observing Network Exploration

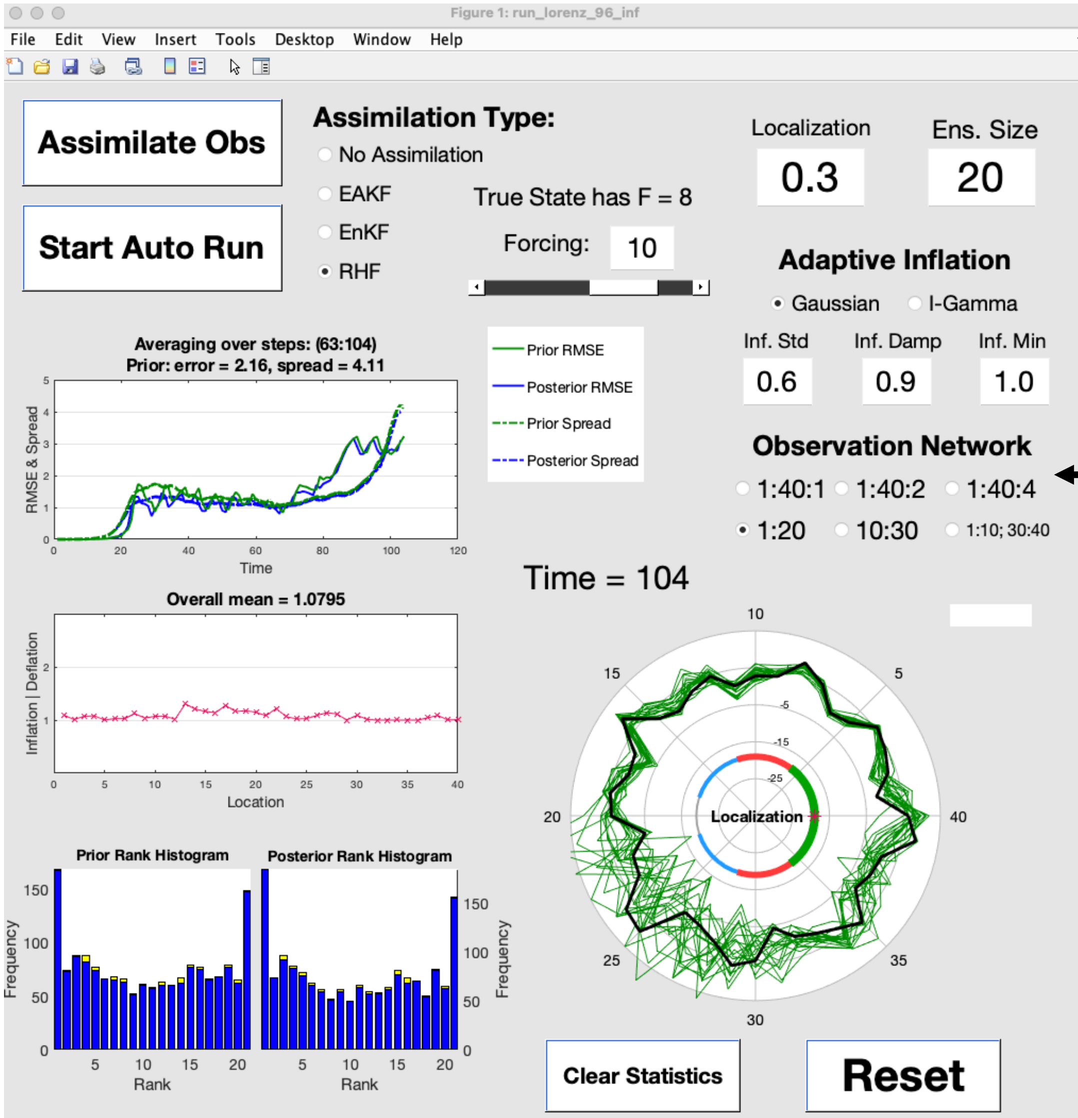
The impact of the observing system can be explored.

Lorenz-96 has 40 state variables.

Following observing systems are available:

- Observe all 40 variables (1:40:1)
- Observe every other variable (1:40:2)
- Observe every 4^{th} variable (1:40:4)
- Observe the first 20 variables only (1:20)
- Observe the 10^{th} to 30^{th} variables only (10:30)
- Observe variables 30 to 40 and 1 to 10 (1:10; 30:40)

Observing Network Exploration: run_Lorenz_96_inf



Observing network controls.

Inflation Distribution Family

The inflation for each state variable is a random variable.

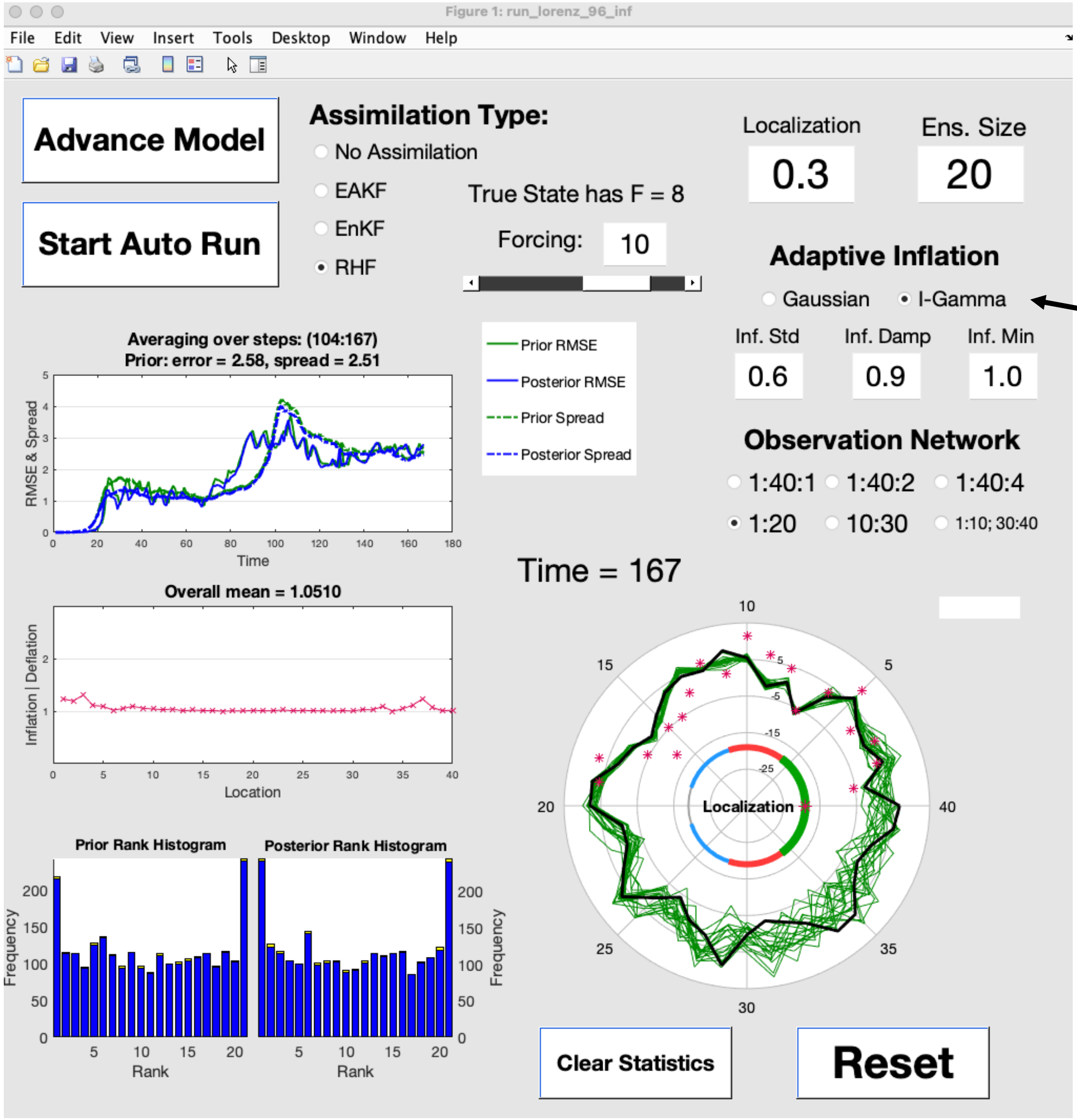
So far, it has been a normal distribution.

But, inflation/deflation is bounded below at zero.

Just like for bounded state variables, we can use a more appropriate distribution.

Inverse gamma is a good choice and is now the DART default.

Inflation Distribution Family: run_Lorenz_96_inf



Choose distribution family for inflation.