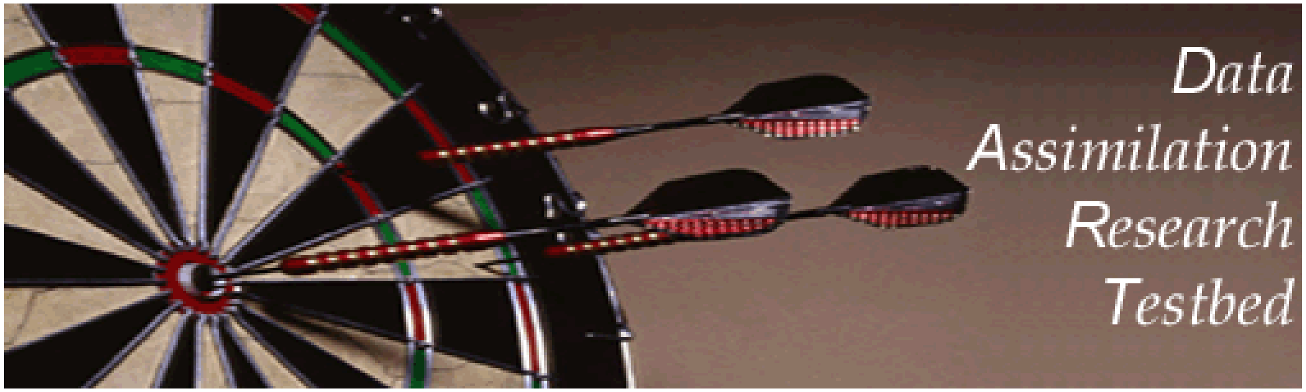
Data Assimilation Research Testbed Tutorial



Section 4: How should observations of a state variable impact an unobserved state variable? Multivariate assimilation.

Version 1.0: June, 2005

Single observed variable, single unobserved variable

So far, have known observation likelihood for single variable.

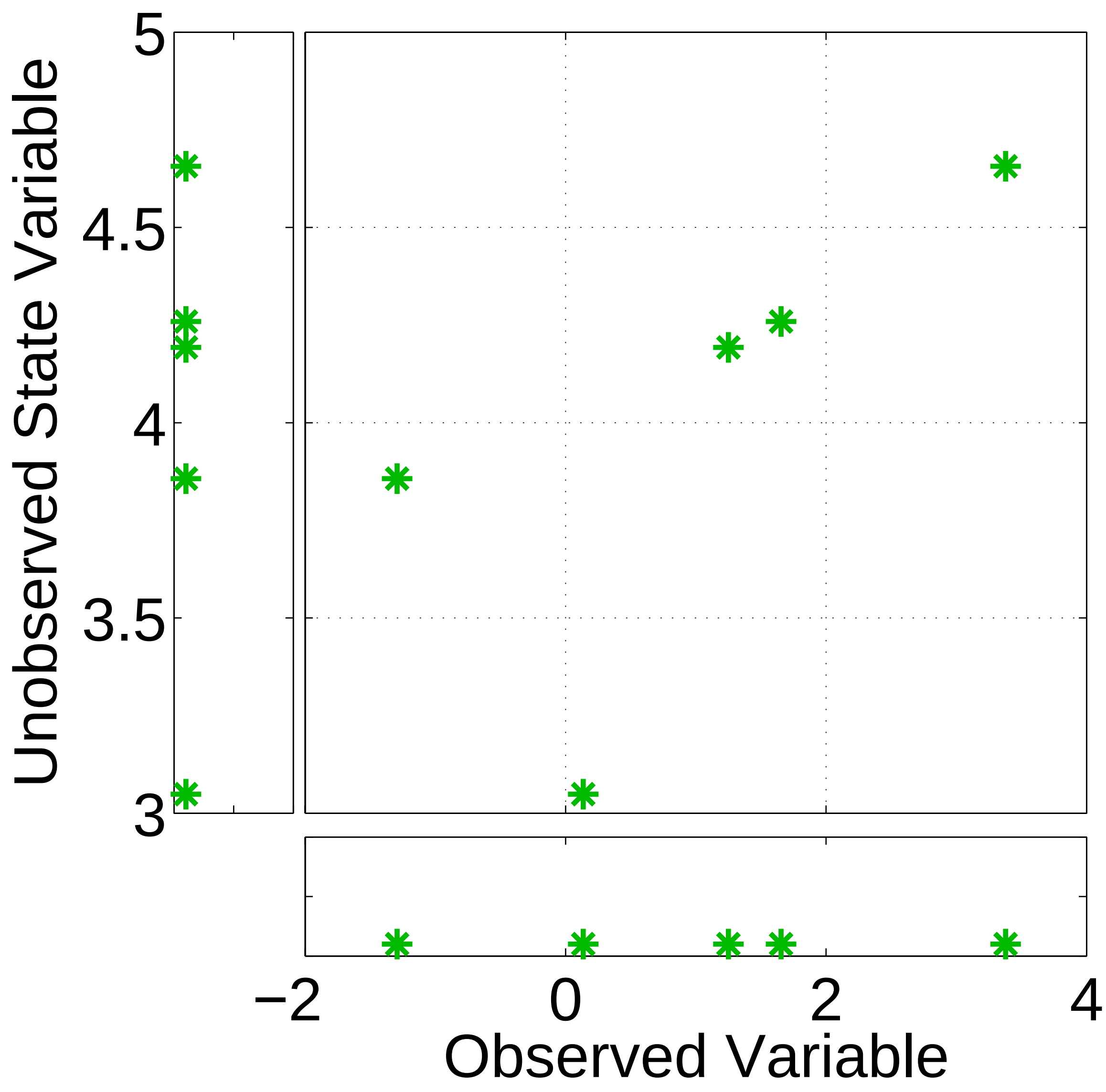
Now, suppose prior has an additional variable.

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

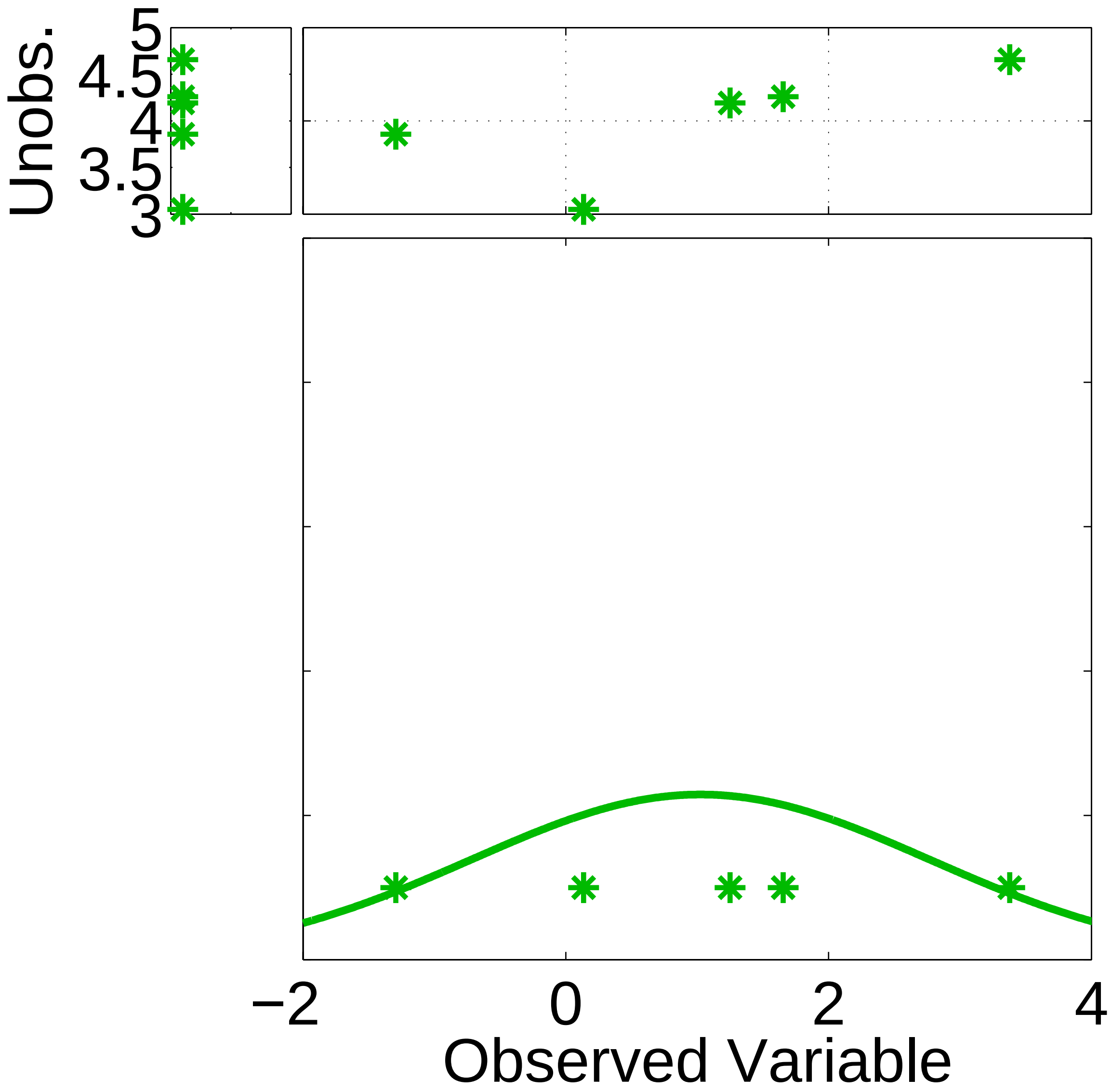


Assume that all we know is prior joint distribution.

One variable is observed.

What should happen to unobserved variable?

Ensemble filters: Updating additional prior state variables

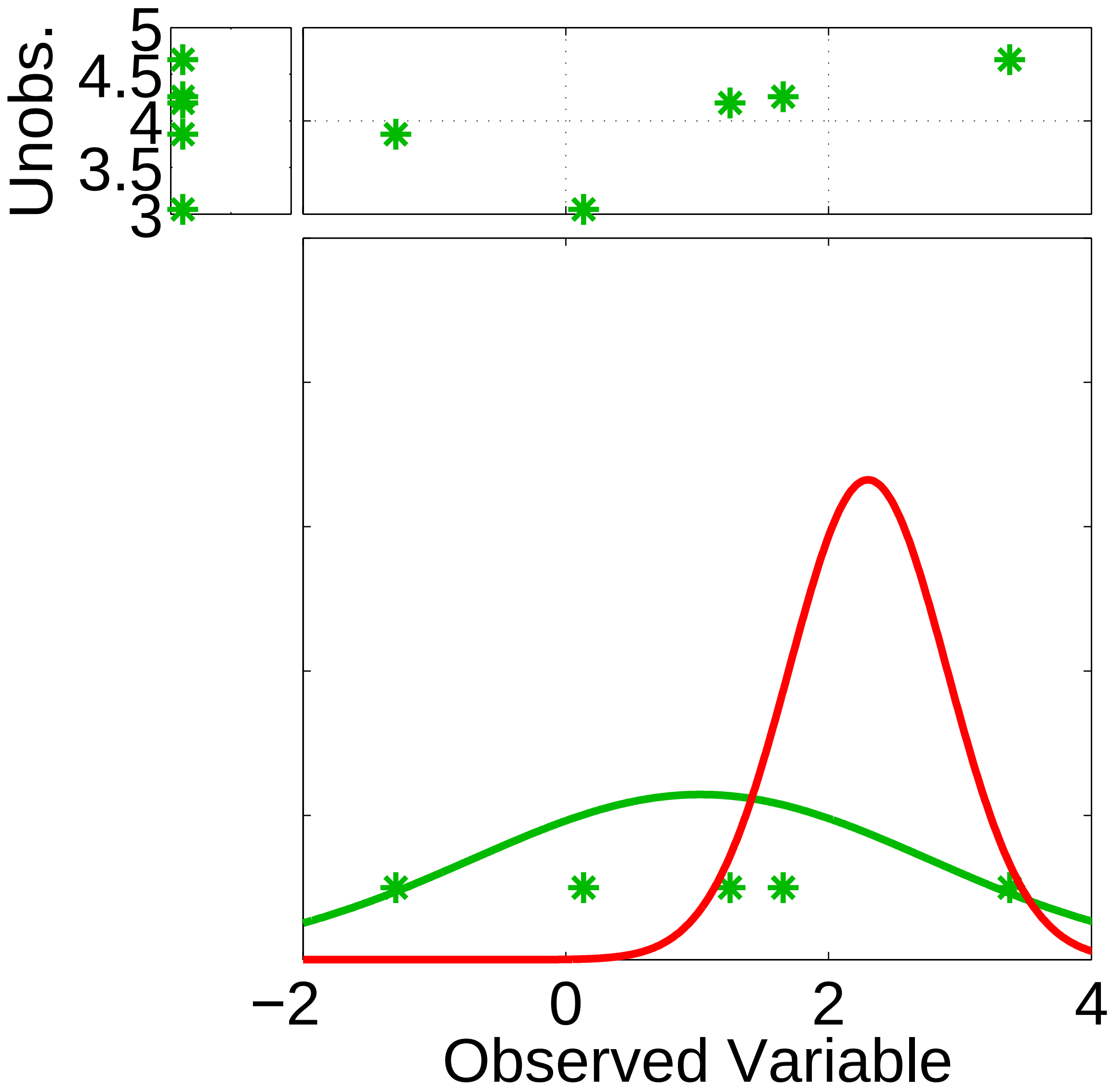


Assume that all we know is prior joint distribution.

One variable is observed.

Update observed variable with one of previous methods.

Ensemble filters: Updating additional prior state variables

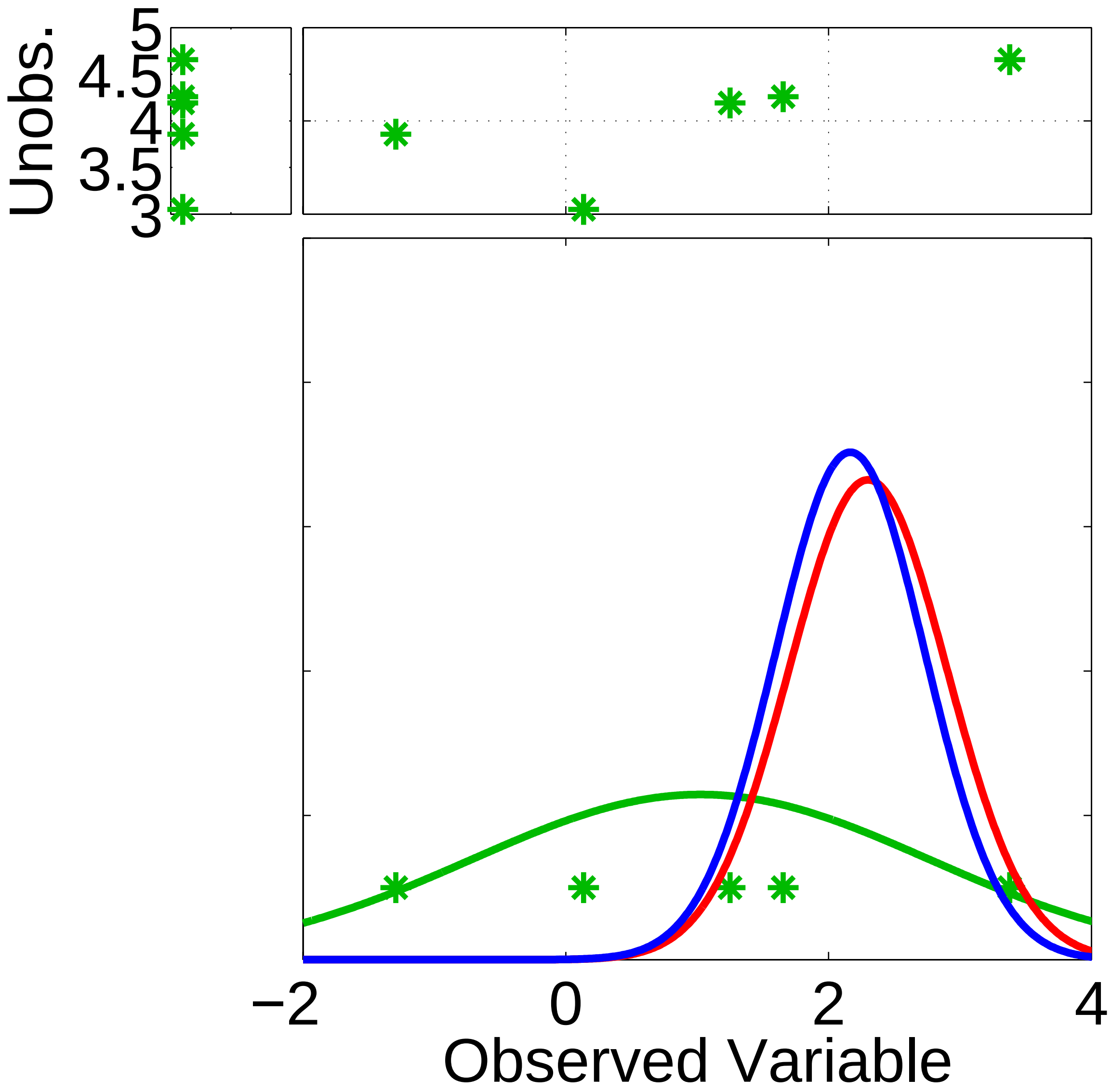


Assume that all we know is prior joint distribution.

One variable is observed.

Update observed variable with one of previous methods.

Ensemble filters: Updating additional prior state variables

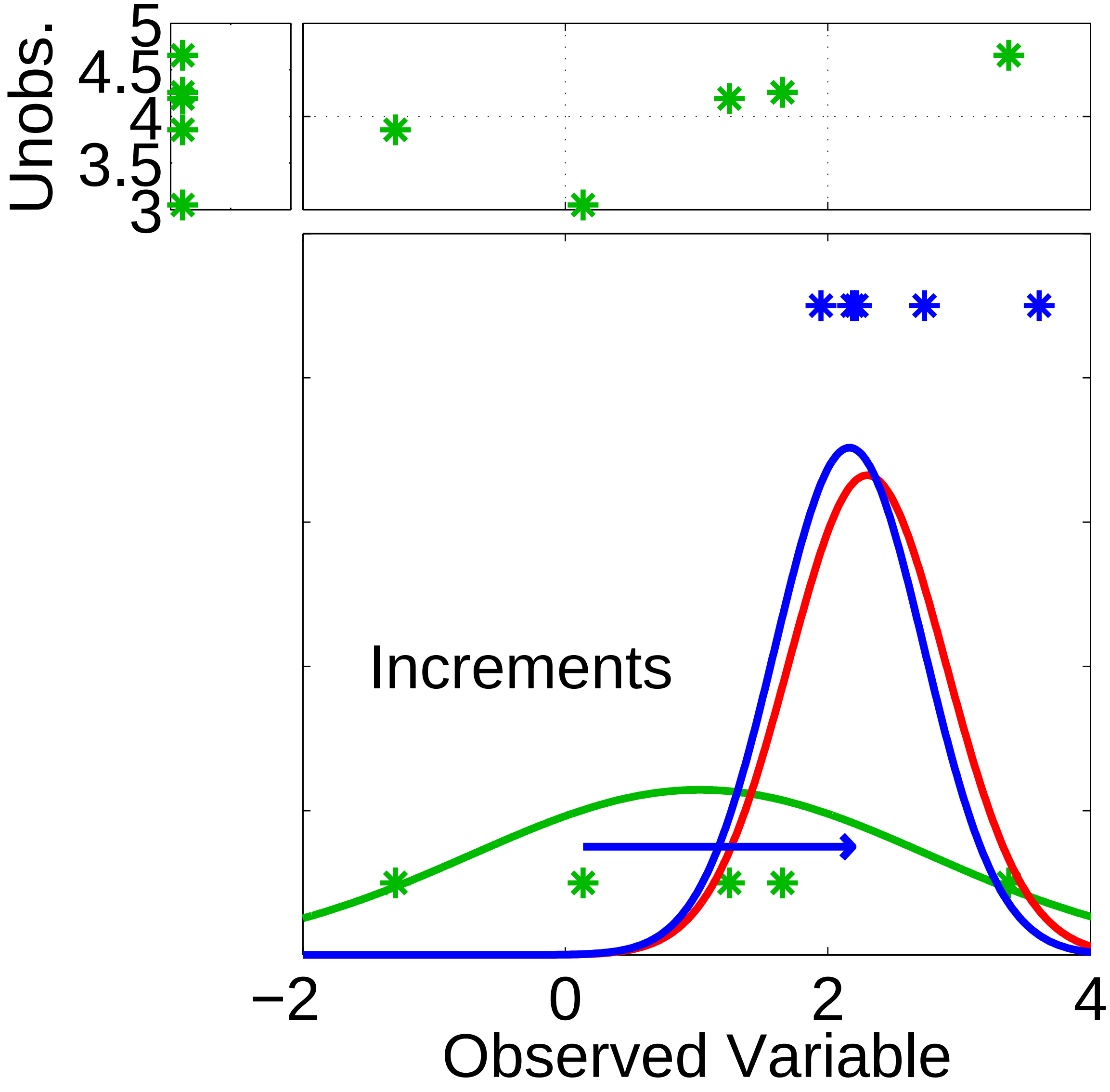


Assume that all we know is prior joint distribution.

One variable is observed.

Update observed variable with one of previous methods.

Ensemble filters: Updating additional prior state variables

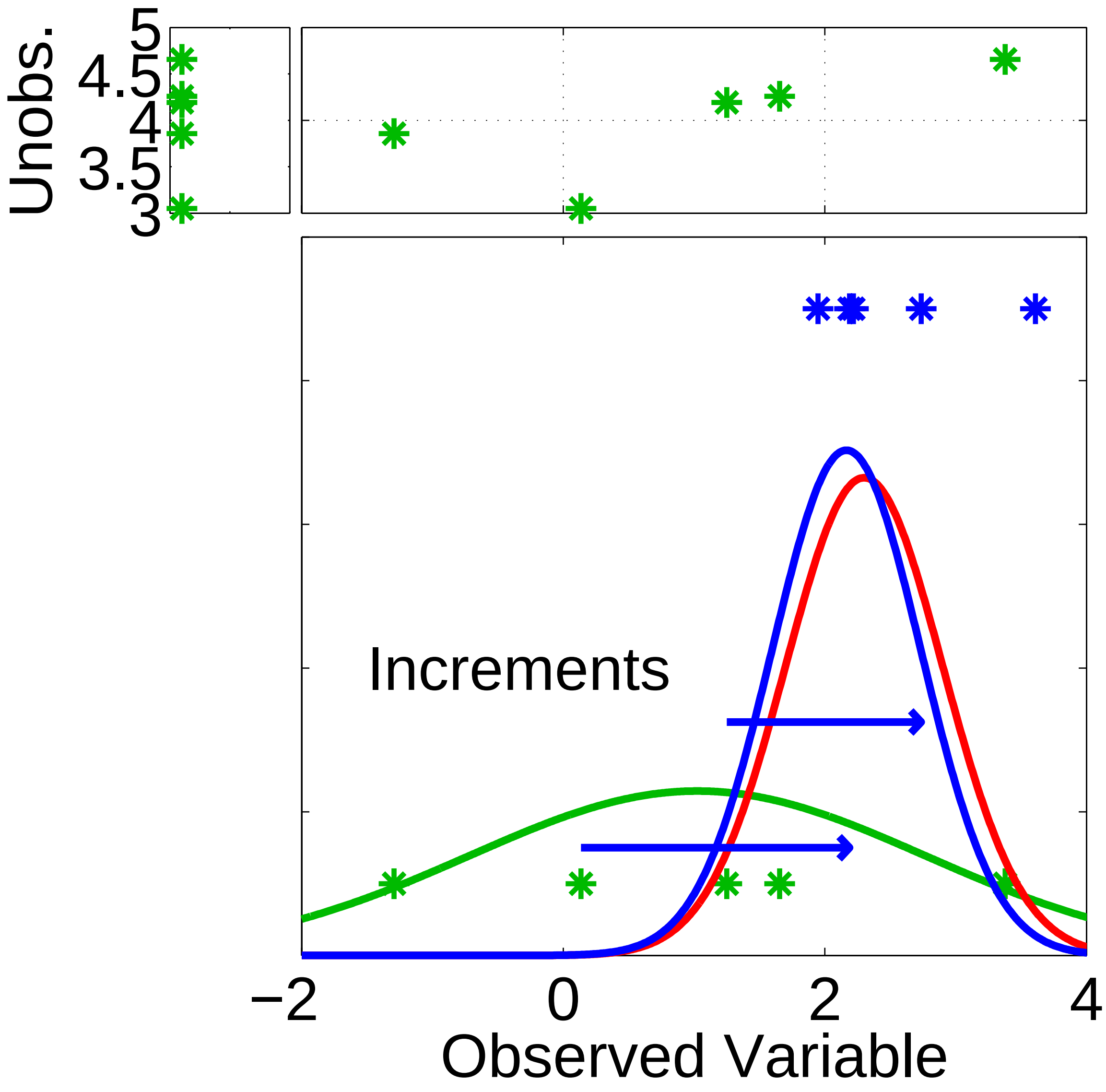


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

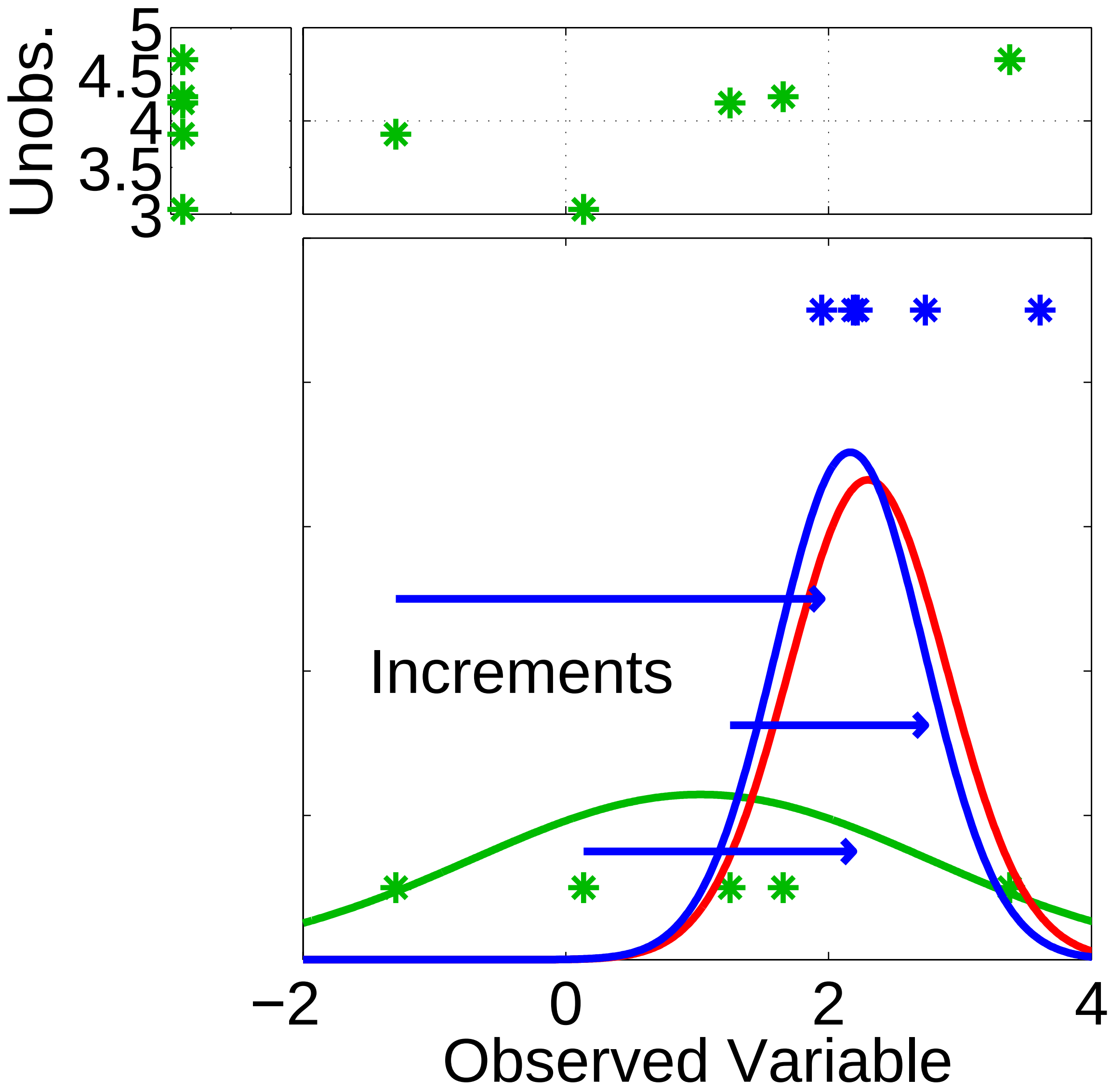


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

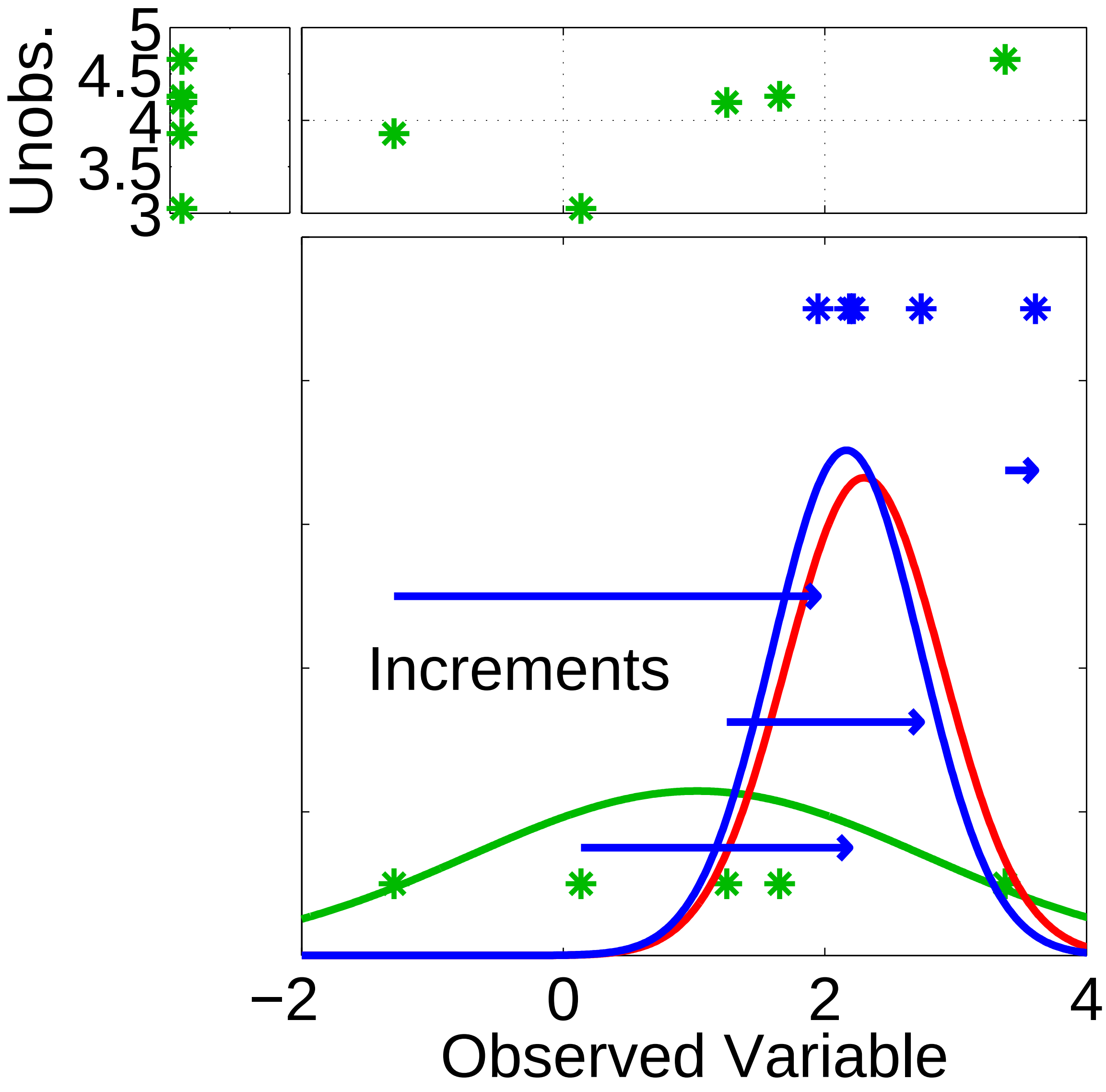


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

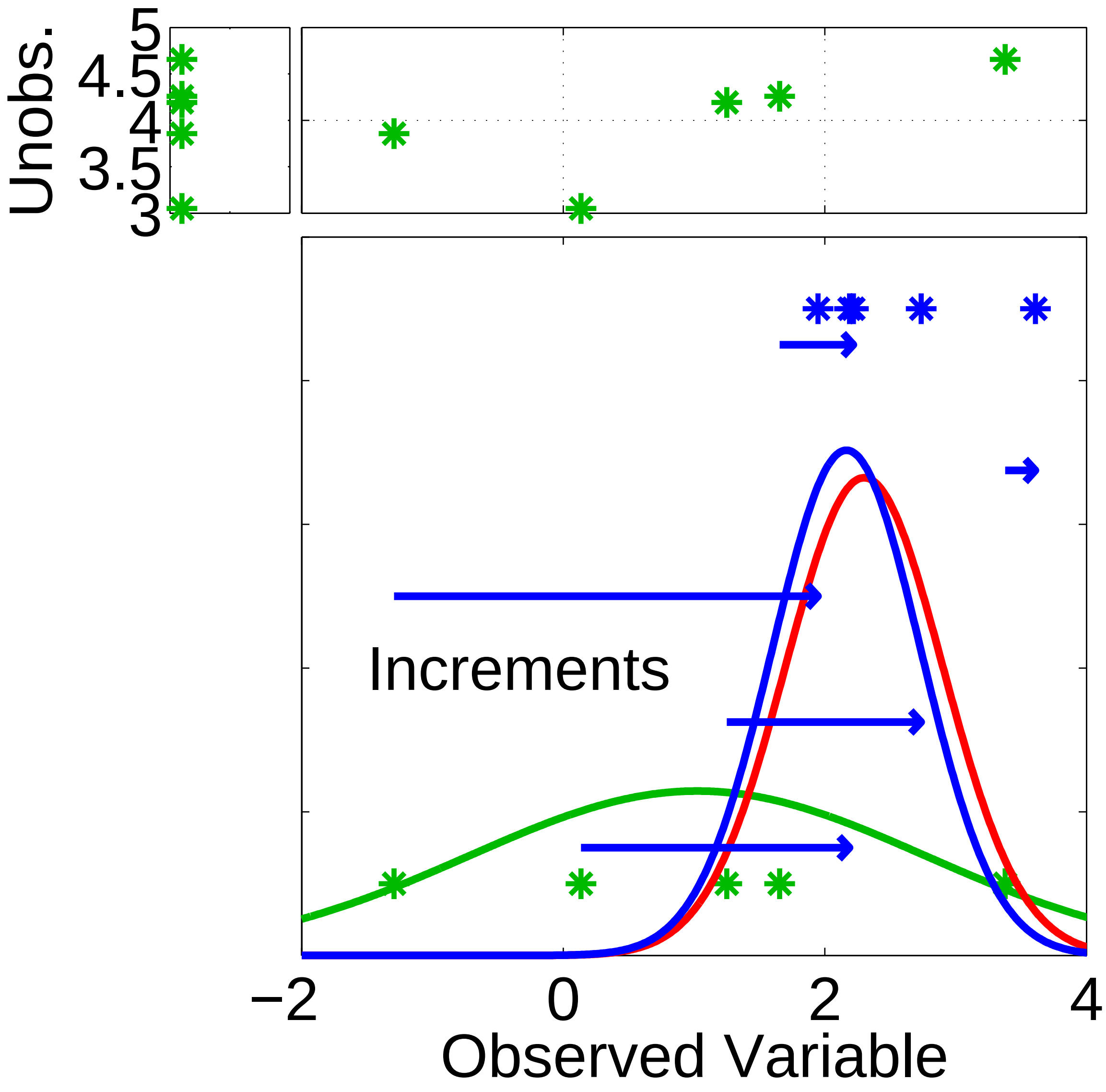


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

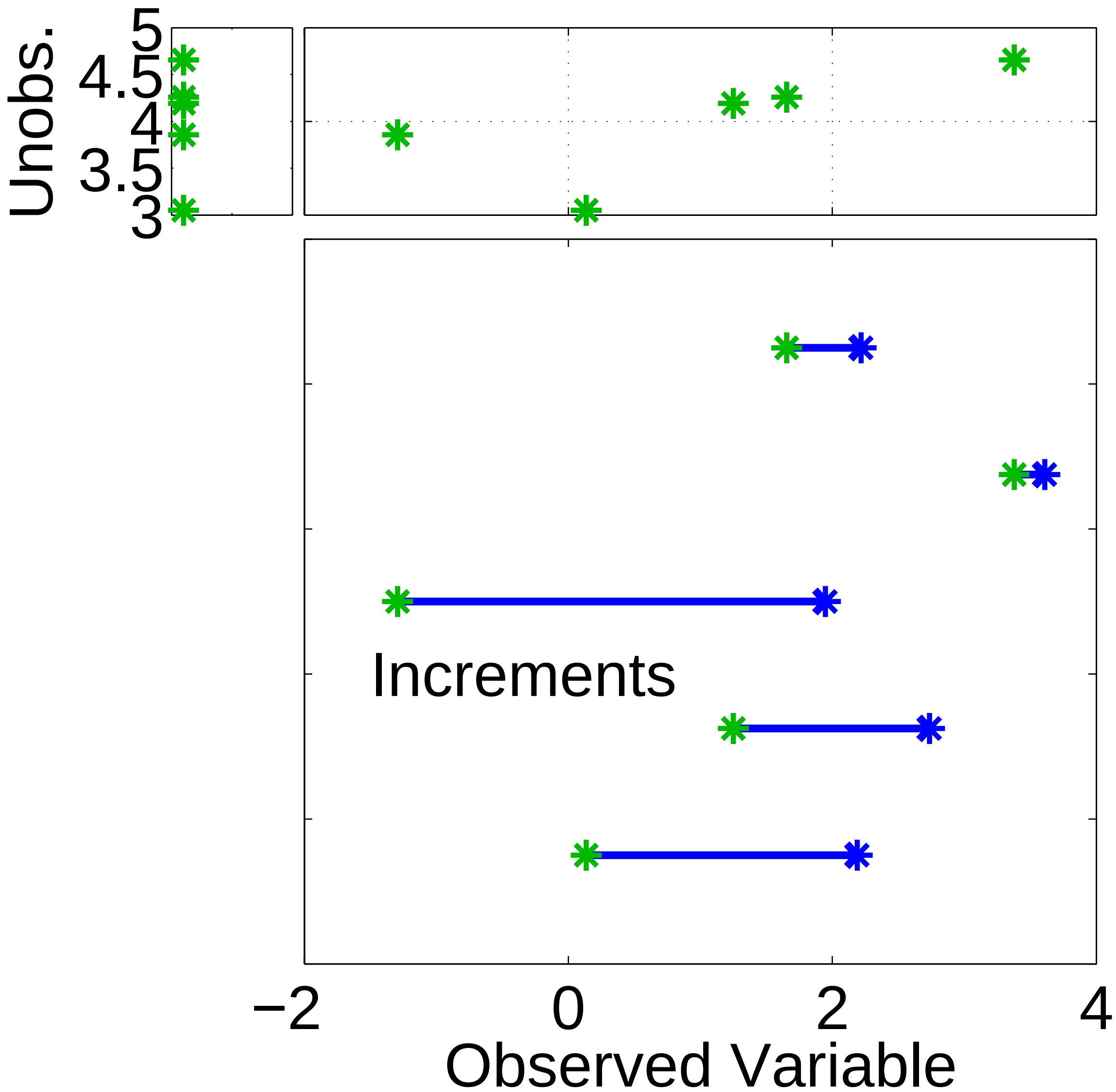


Assume that all we know is prior joint distribution.

One variable is observed.

Compute increments for prior ensemble members of observed variable.

Ensemble filters: Updating additional prior state variables

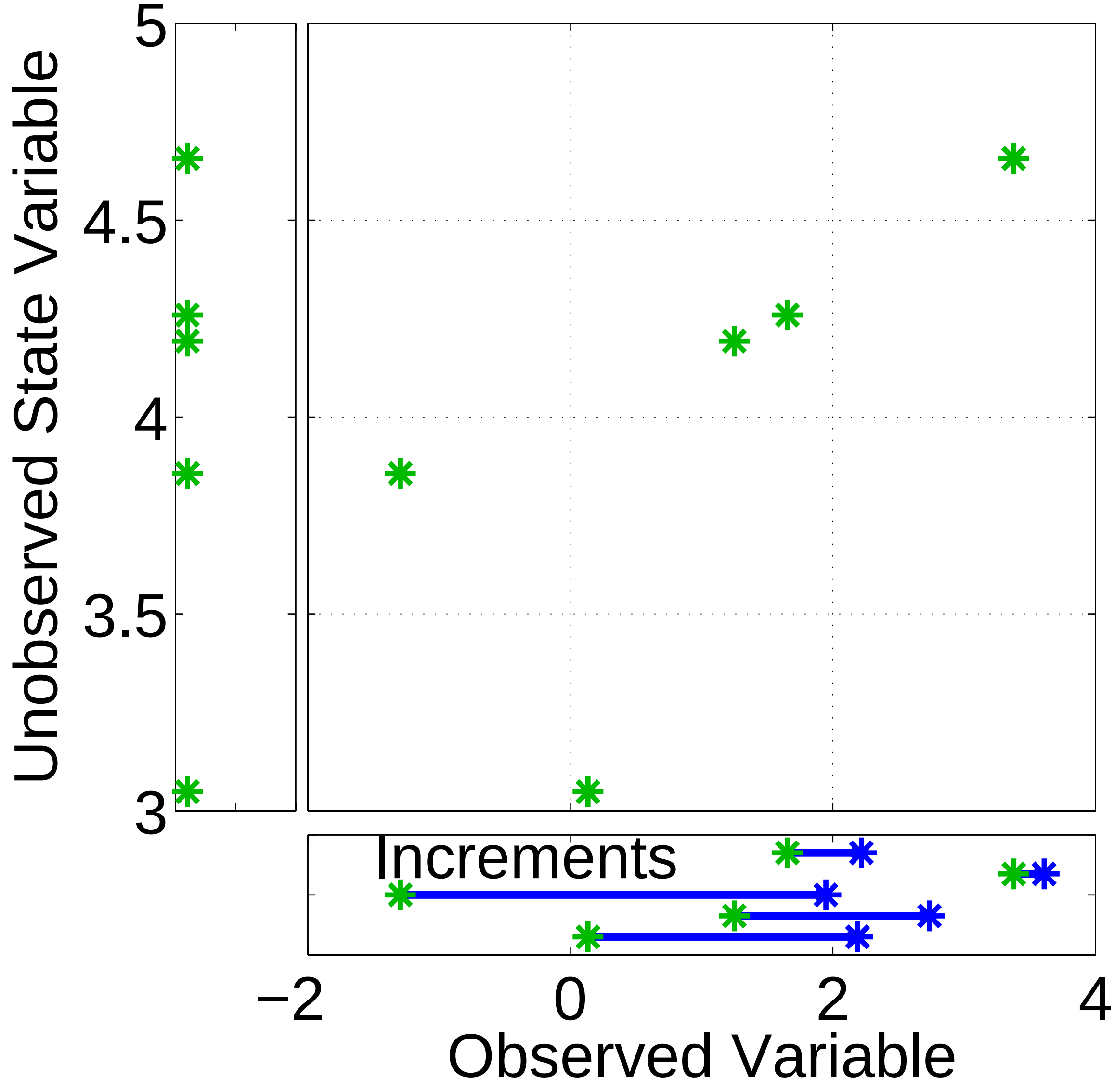


Assume that all we know is prior joint distribution.

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.

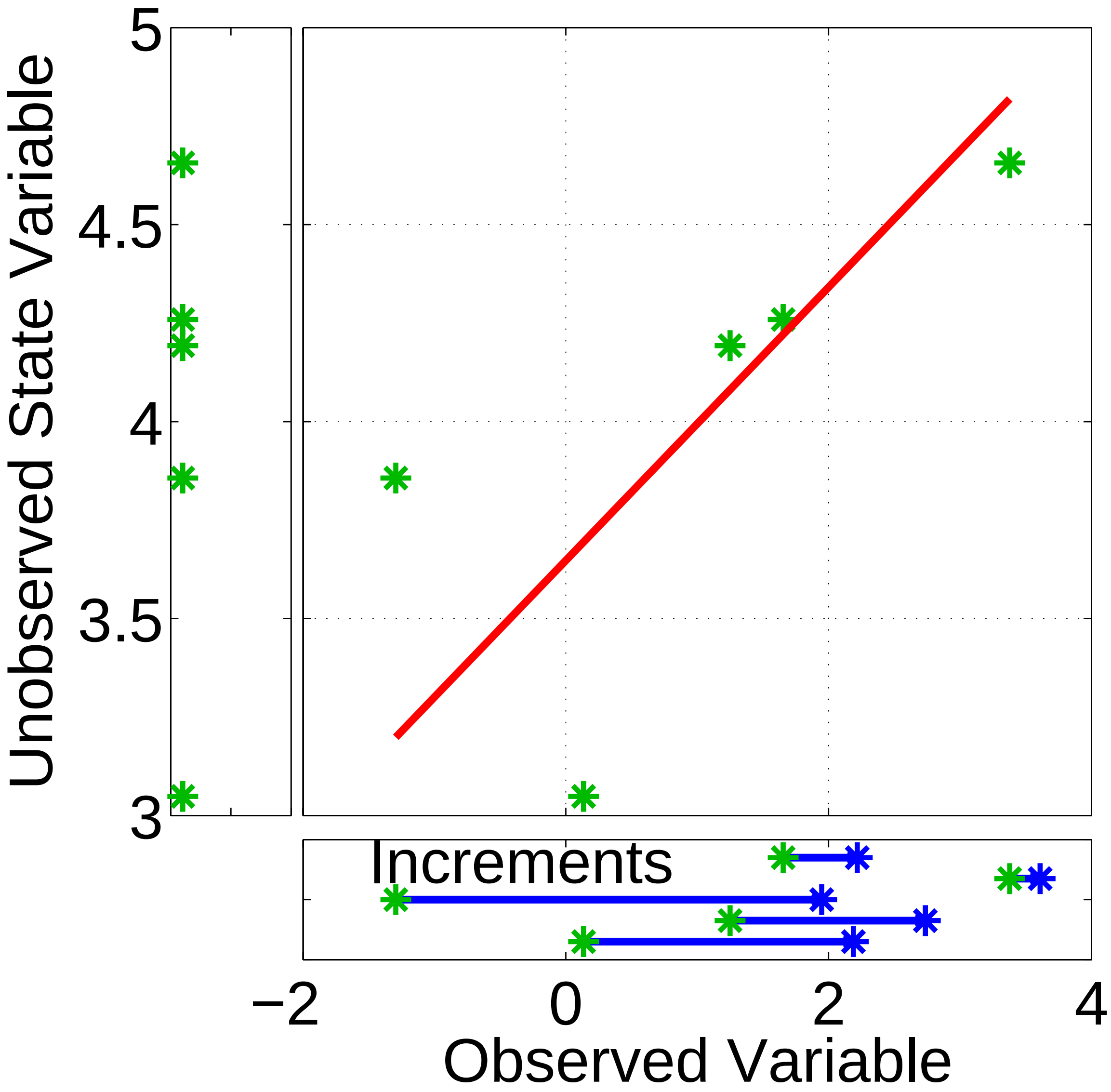
How should the unobserved variable be impacted?

First choice: least squares

Equivalent to linear regression.

Same as assuming binormal prior.

Ensemble filters: Updating additional prior state variables



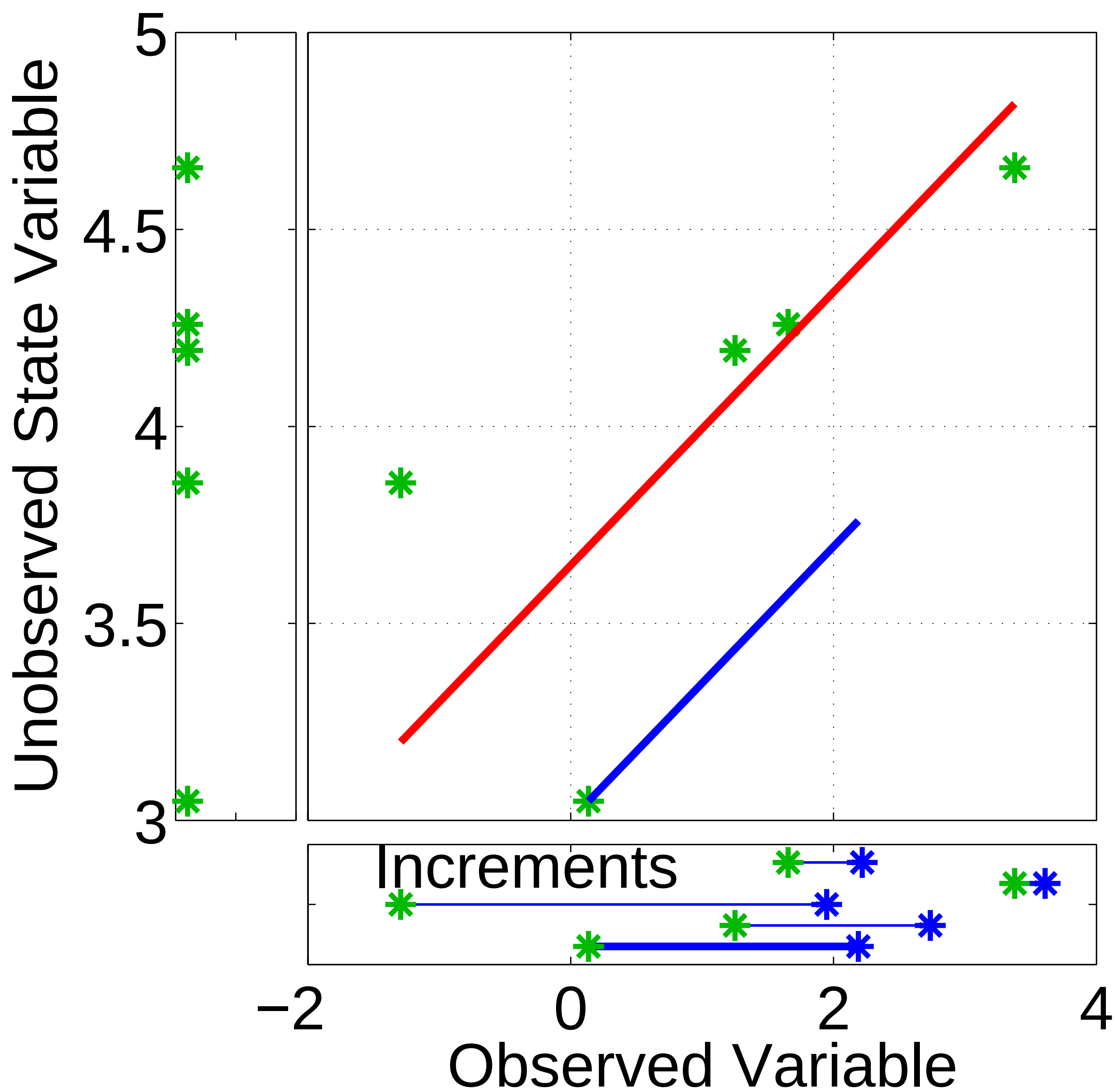
Have joint prior distribution of two variables.

How should the unobserved variable be impacted?

First choice: least squares

Begin by finding least squares fit.

Ensemble filters: Updating additional prior state variables

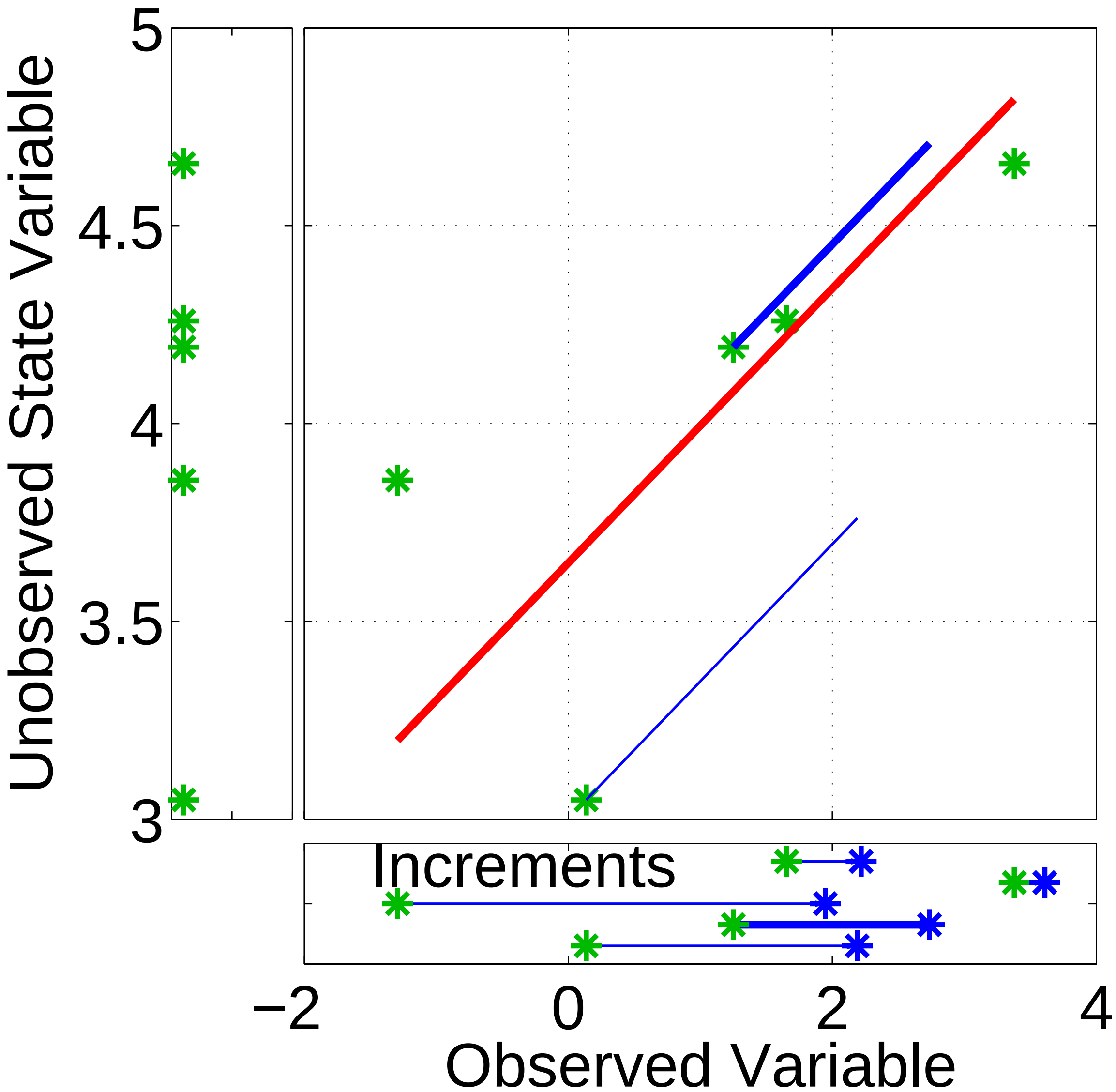


Have joint prior distribution of two variables.

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

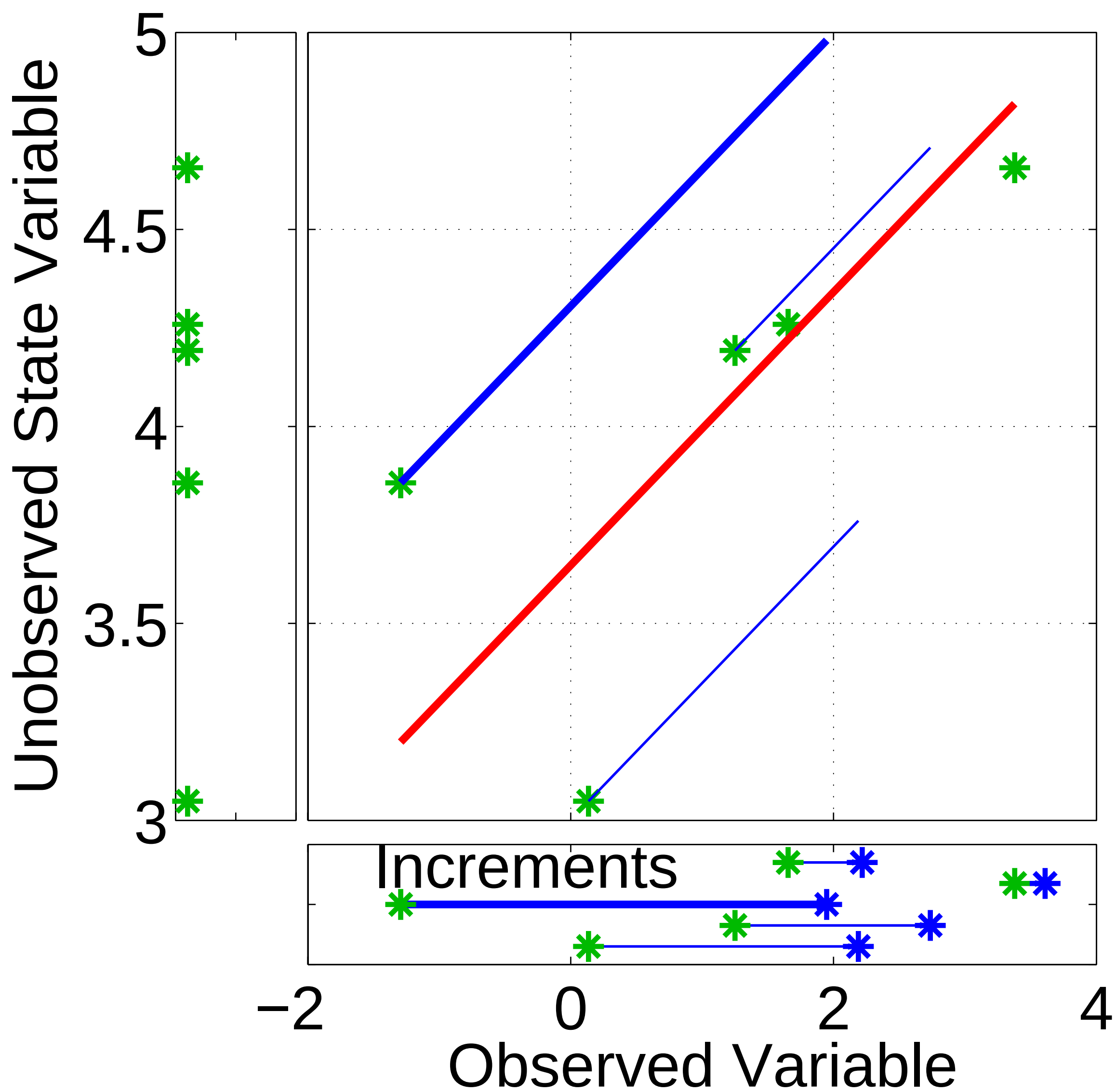


Have joint prior distribution of two variables.

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

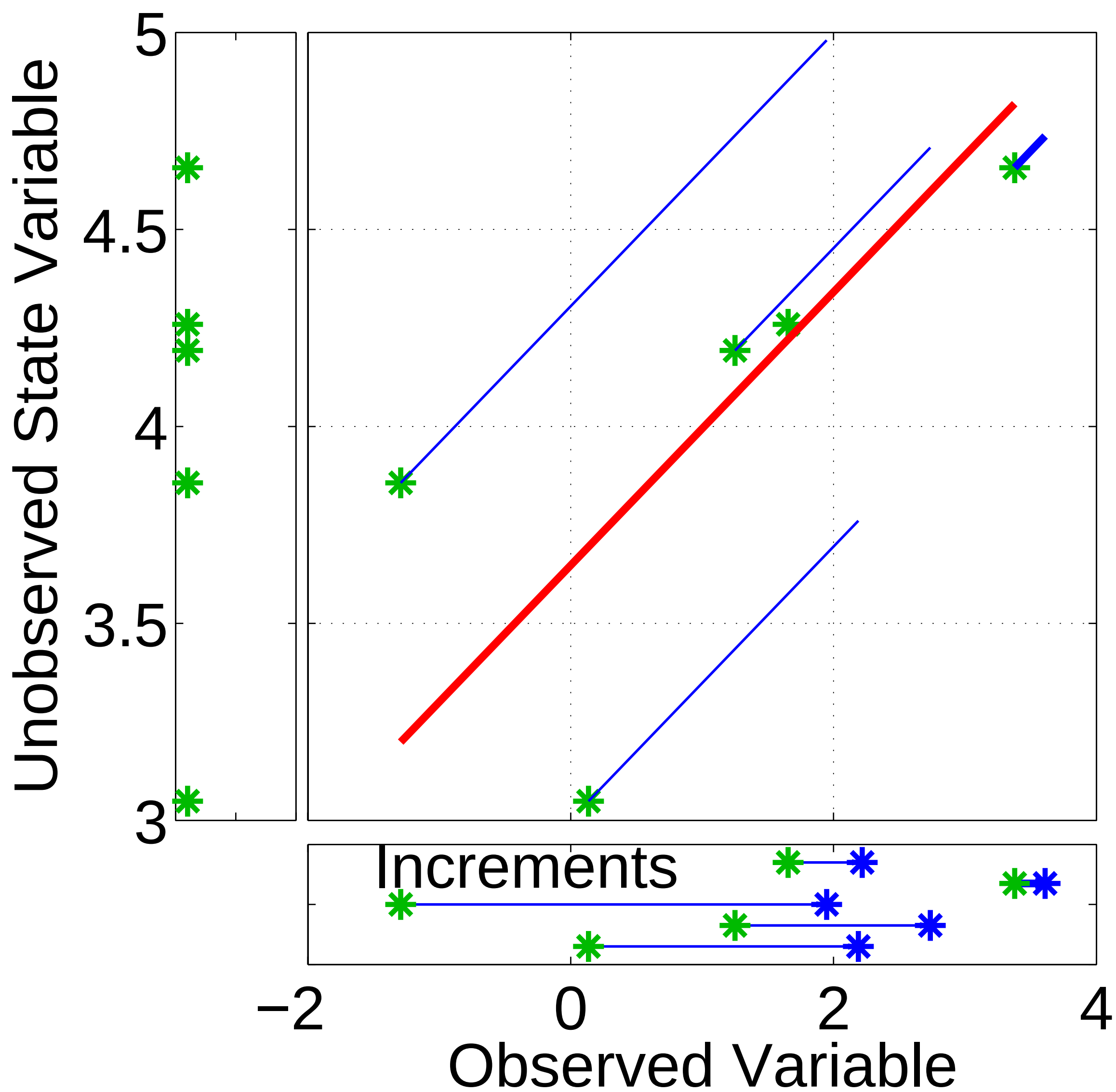


Have joint prior distribution of two variables.

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

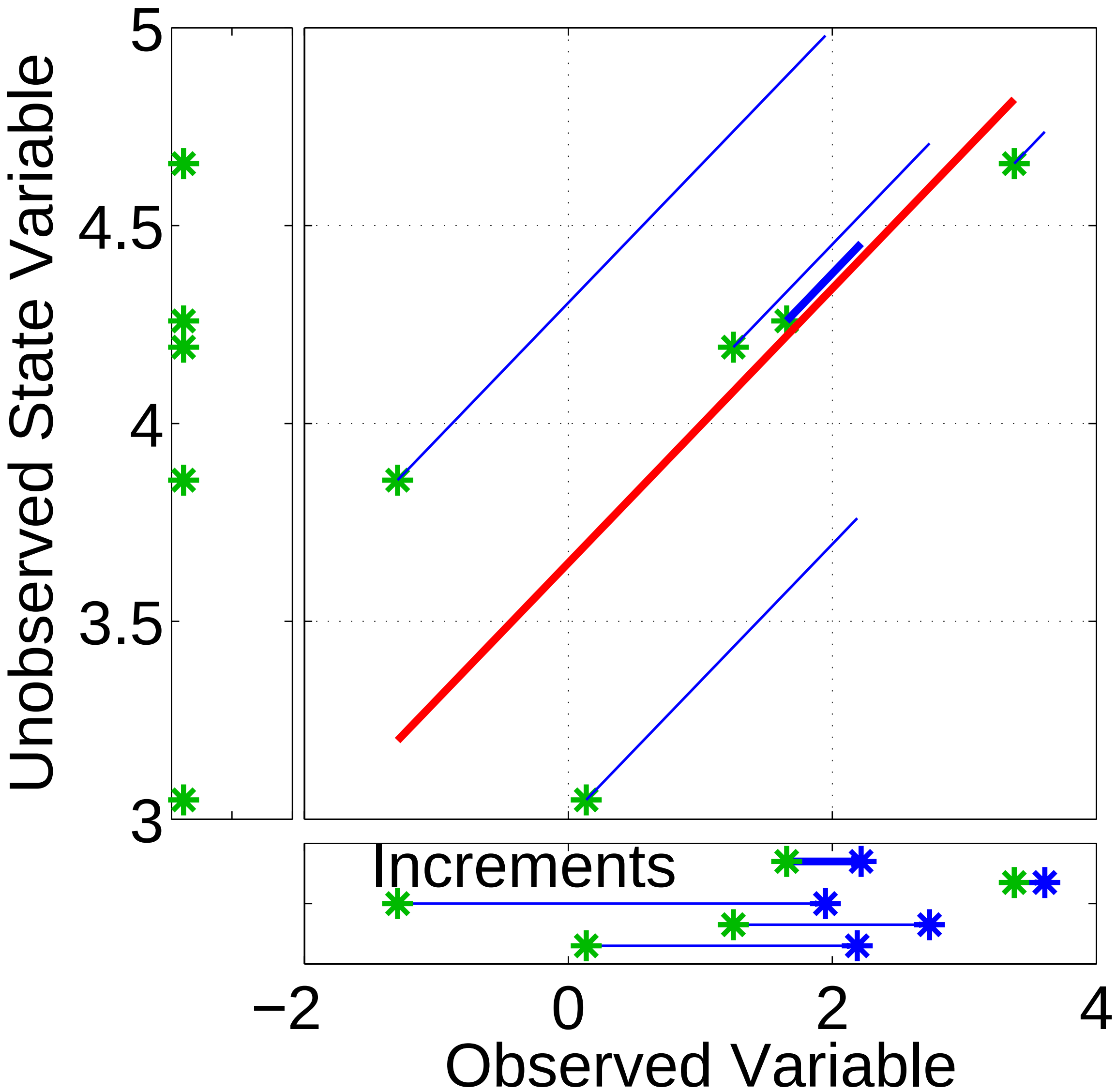


Have joint prior distribution of two variables.

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

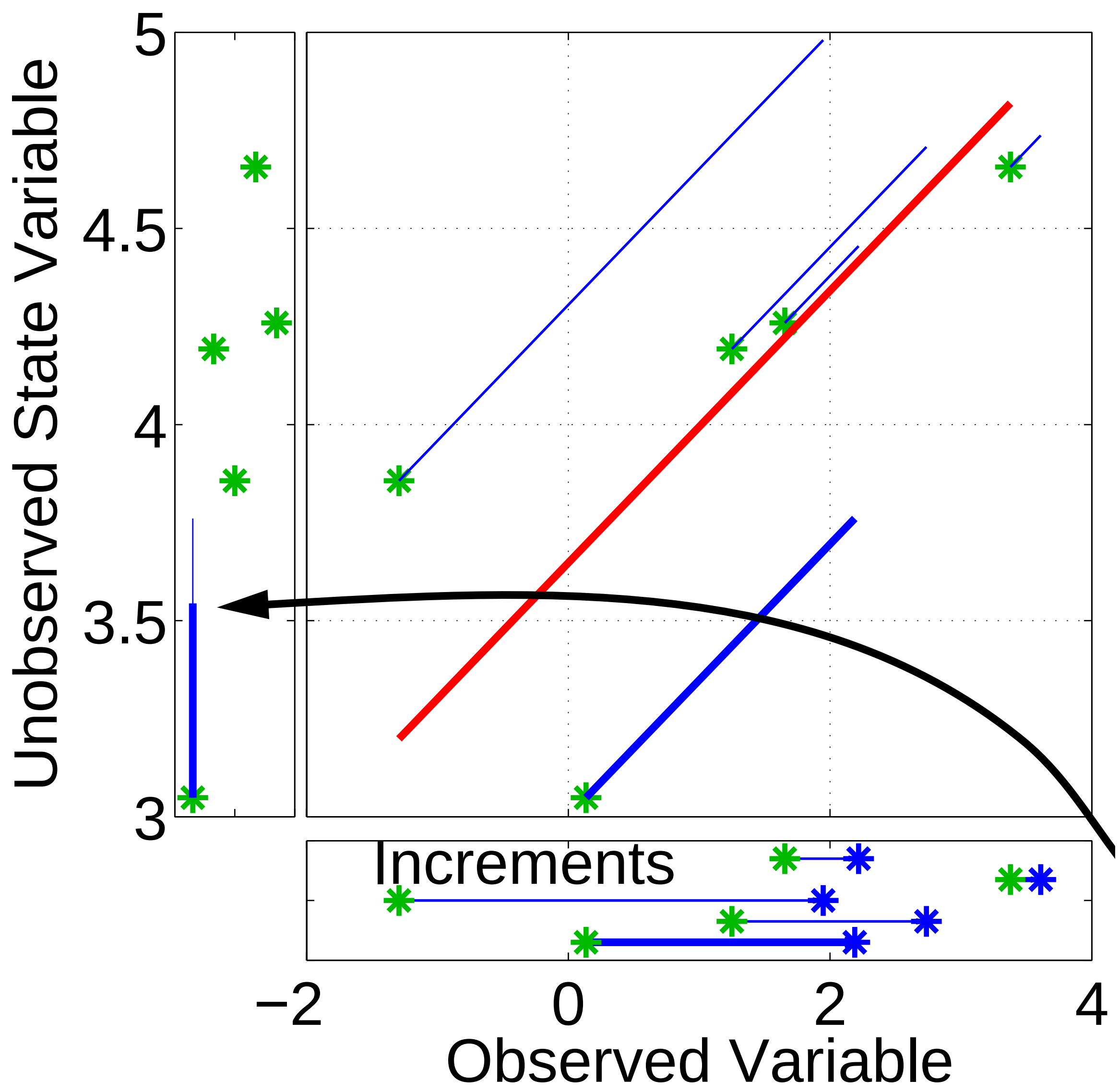


Have joint prior distribution of two variables.

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



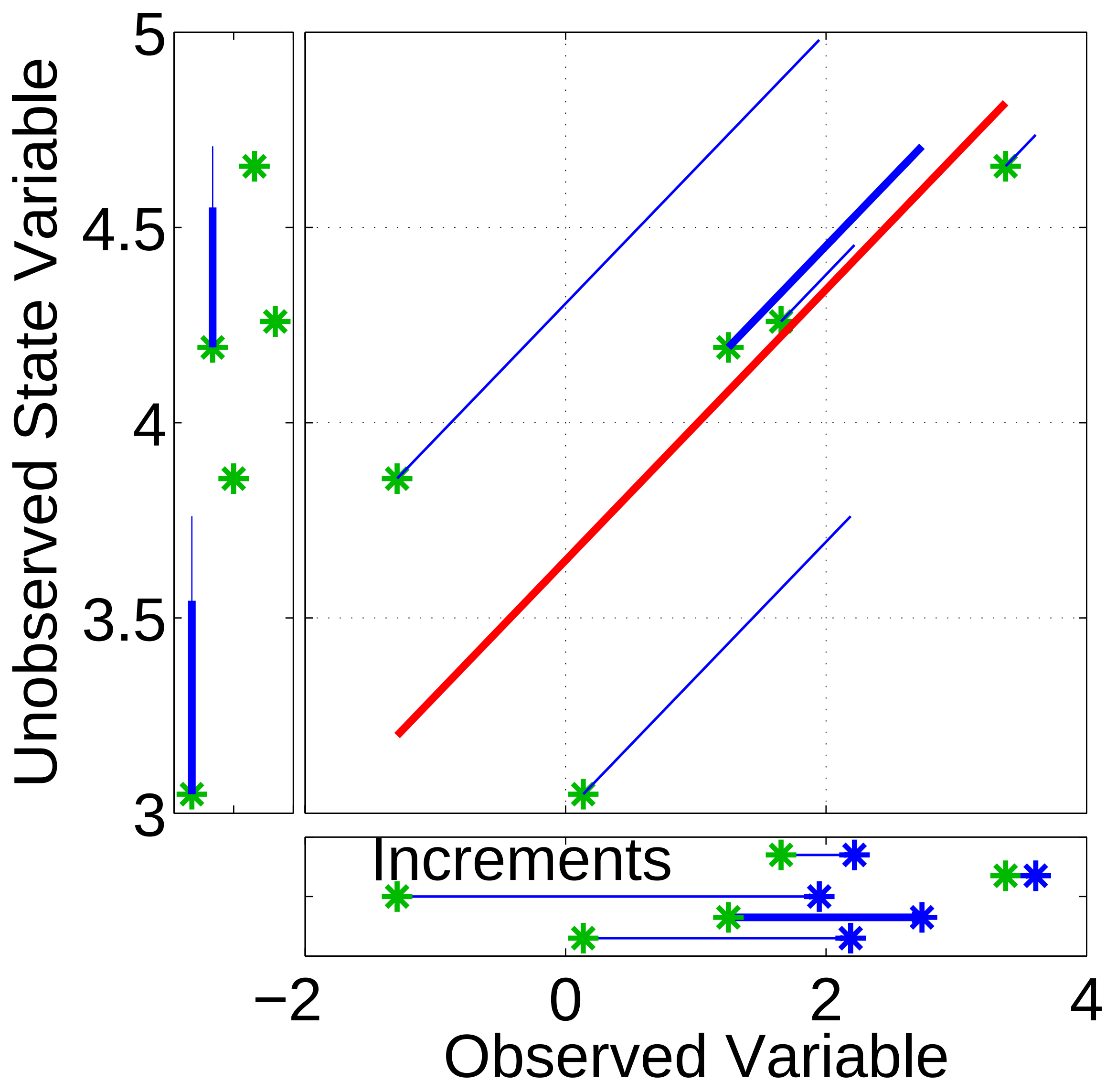
Have joint prior distribution of two variables.

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



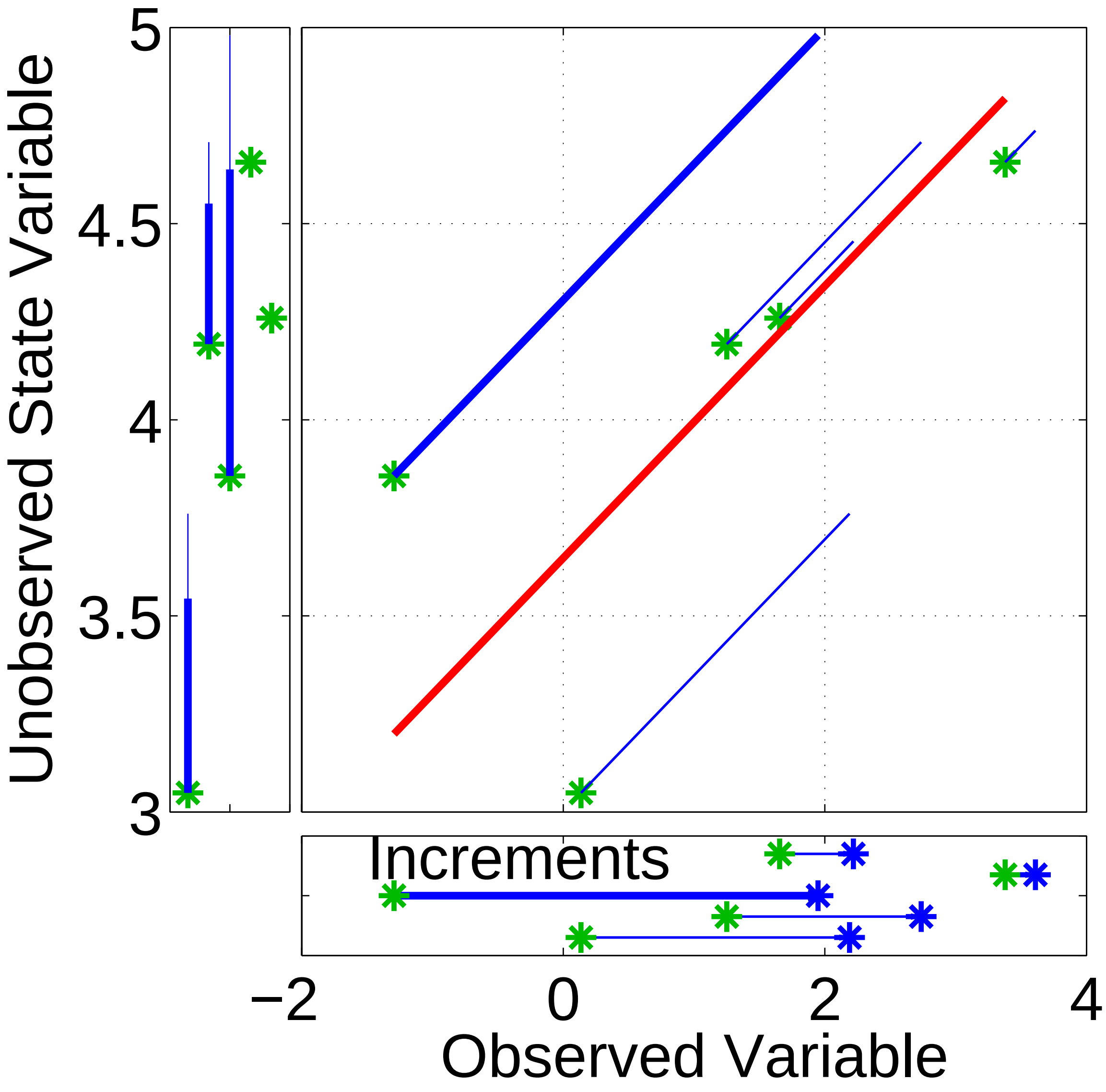
Have joint prior distribution of two variables.

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



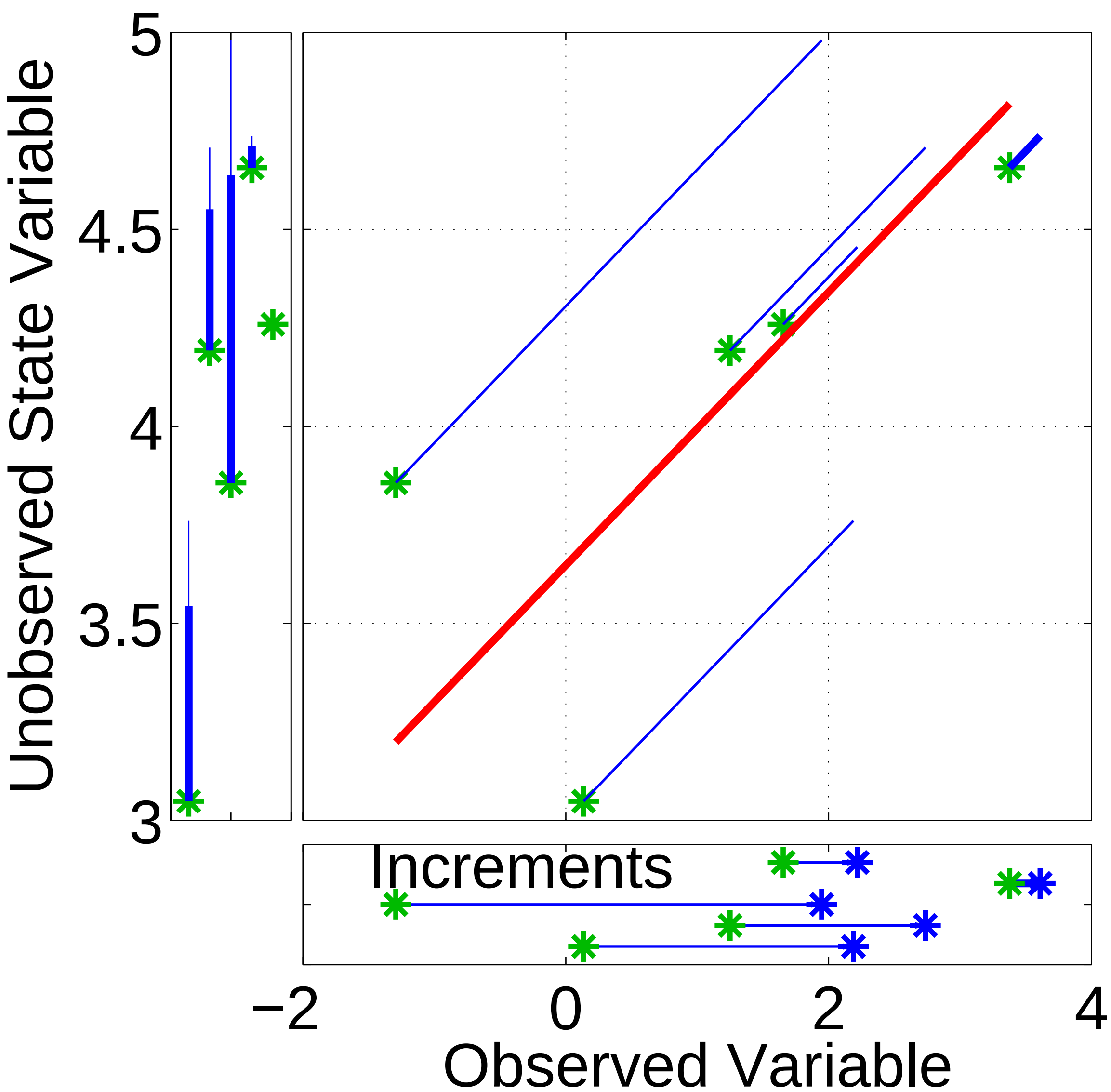
Have joint prior distribution of two variables.

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



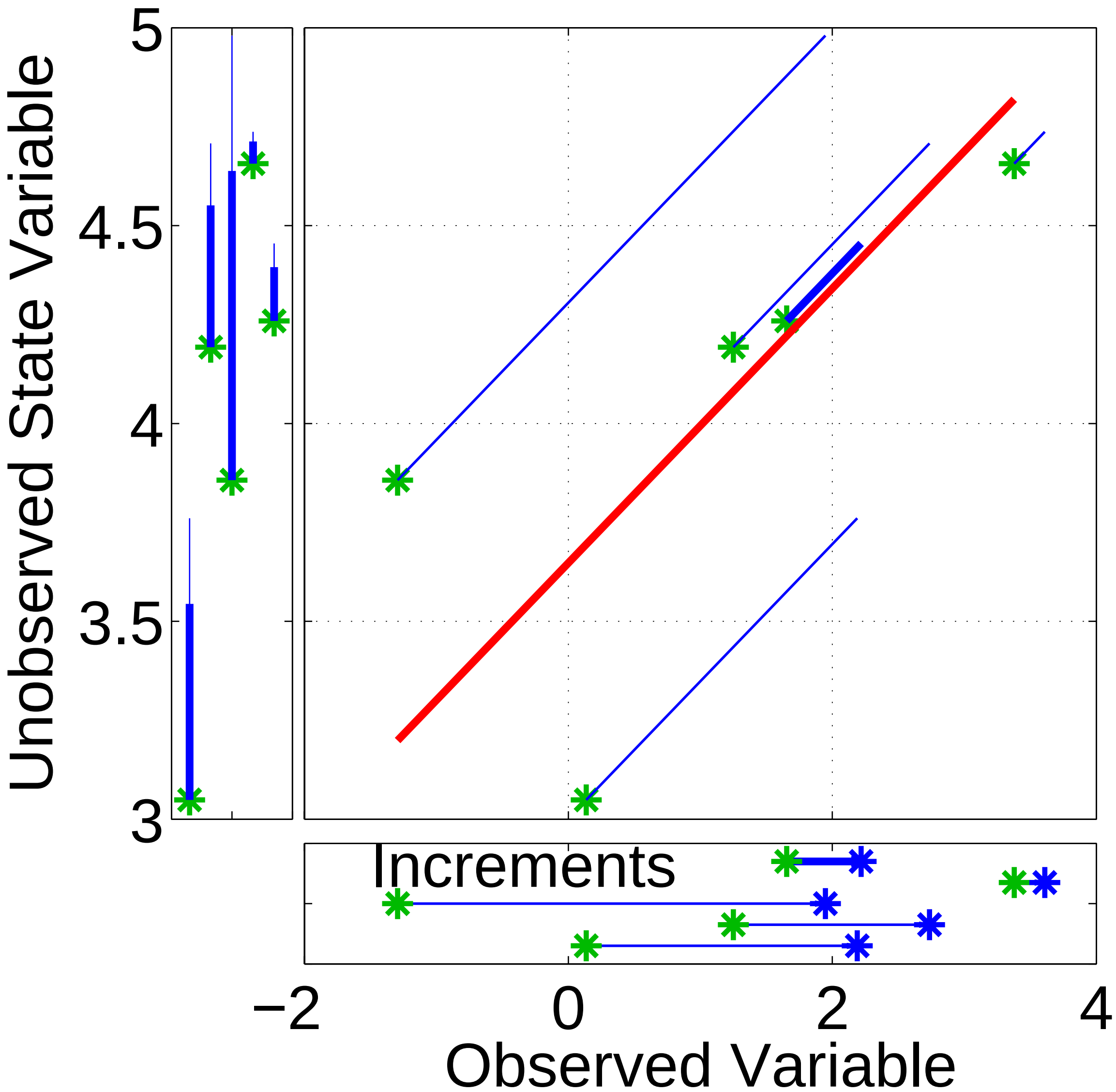
Have joint prior distribution of two variables.

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



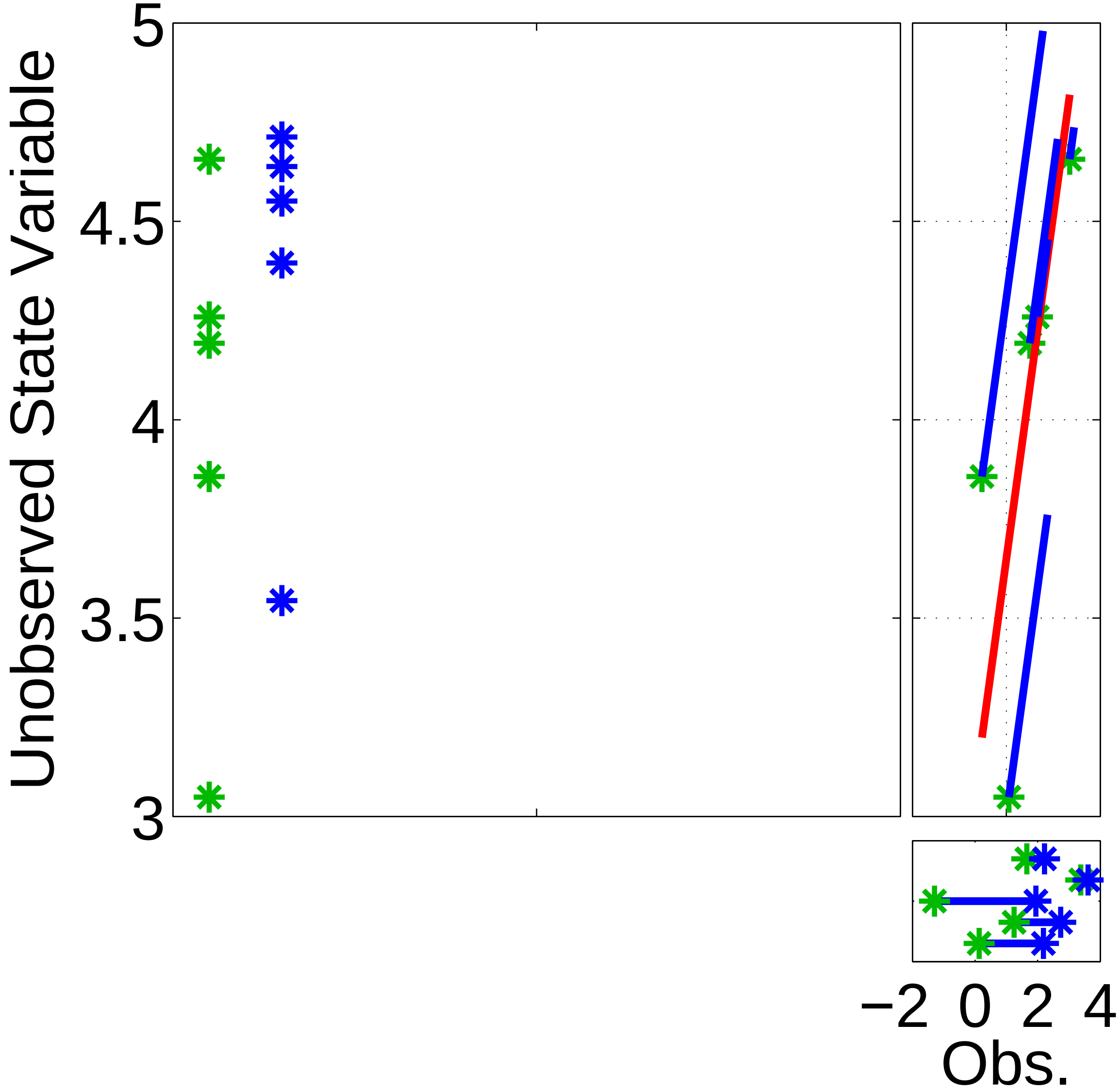
Have joint prior distribution of two variables.

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

Then projecting from joint space onto unobserved priors.

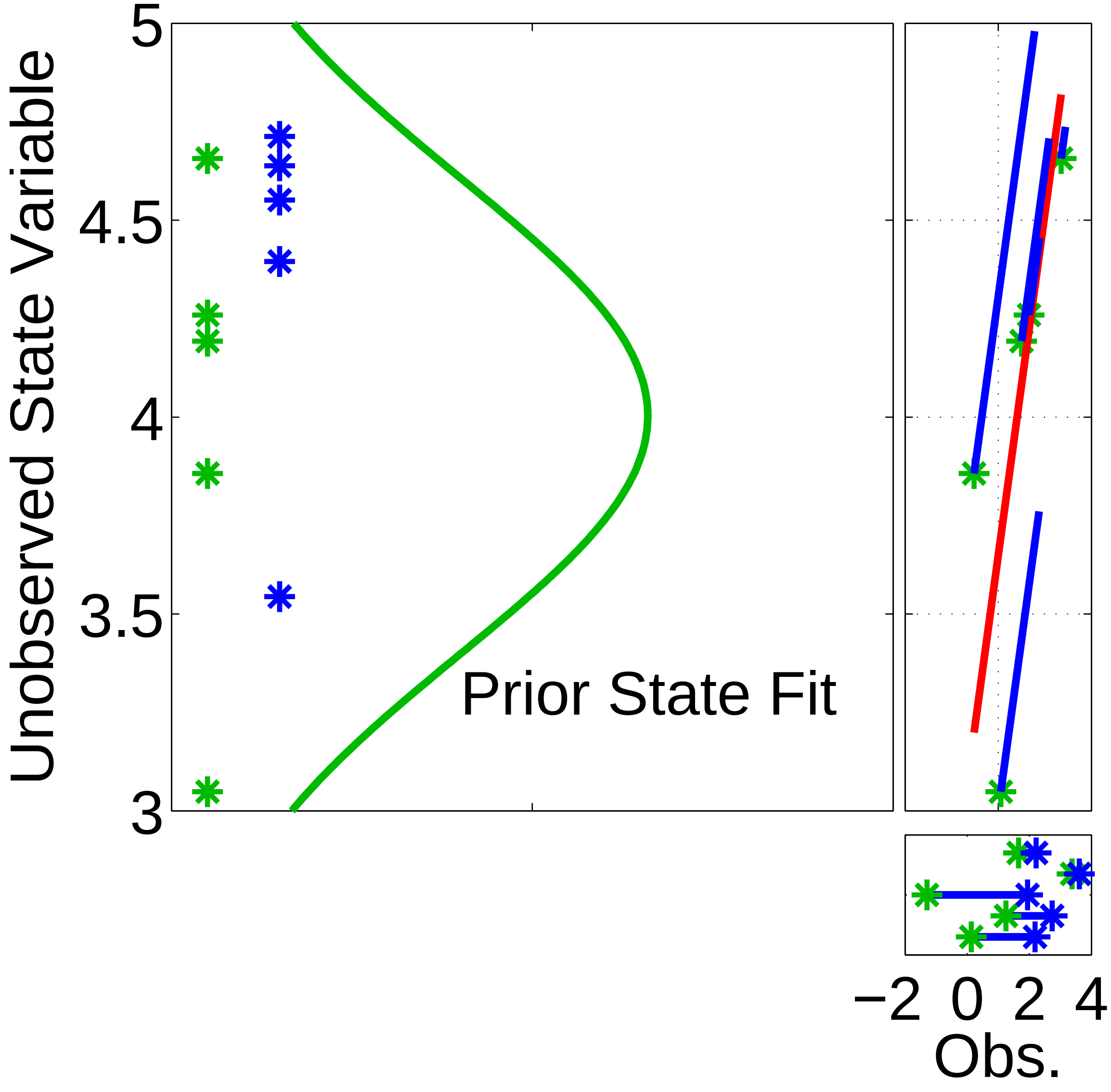
Finally, multiply by prior sample correlation.

Ensemble filters: Updating additional prior state variables



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

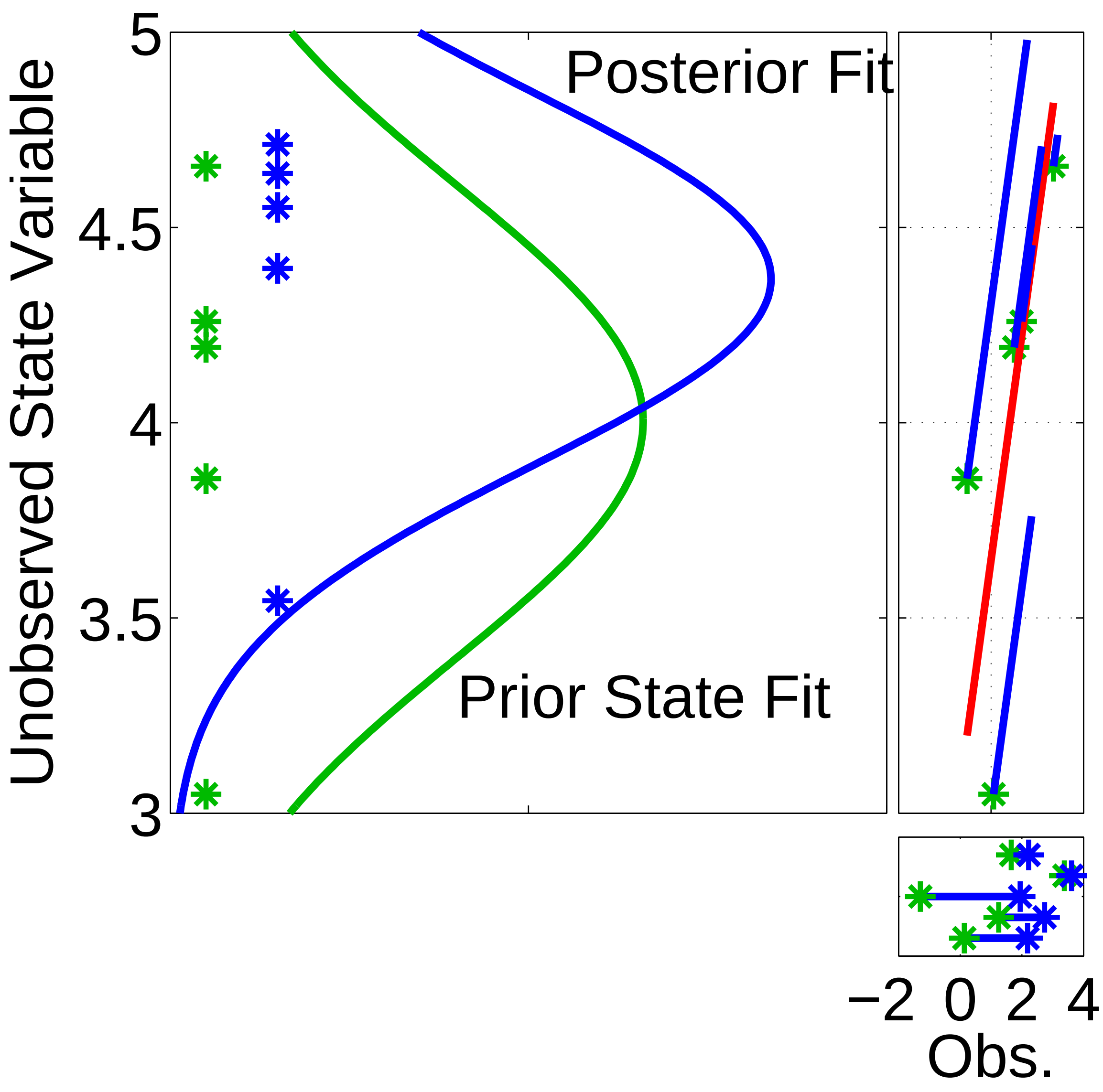
Ensemble filters: Updating additional prior state variables



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

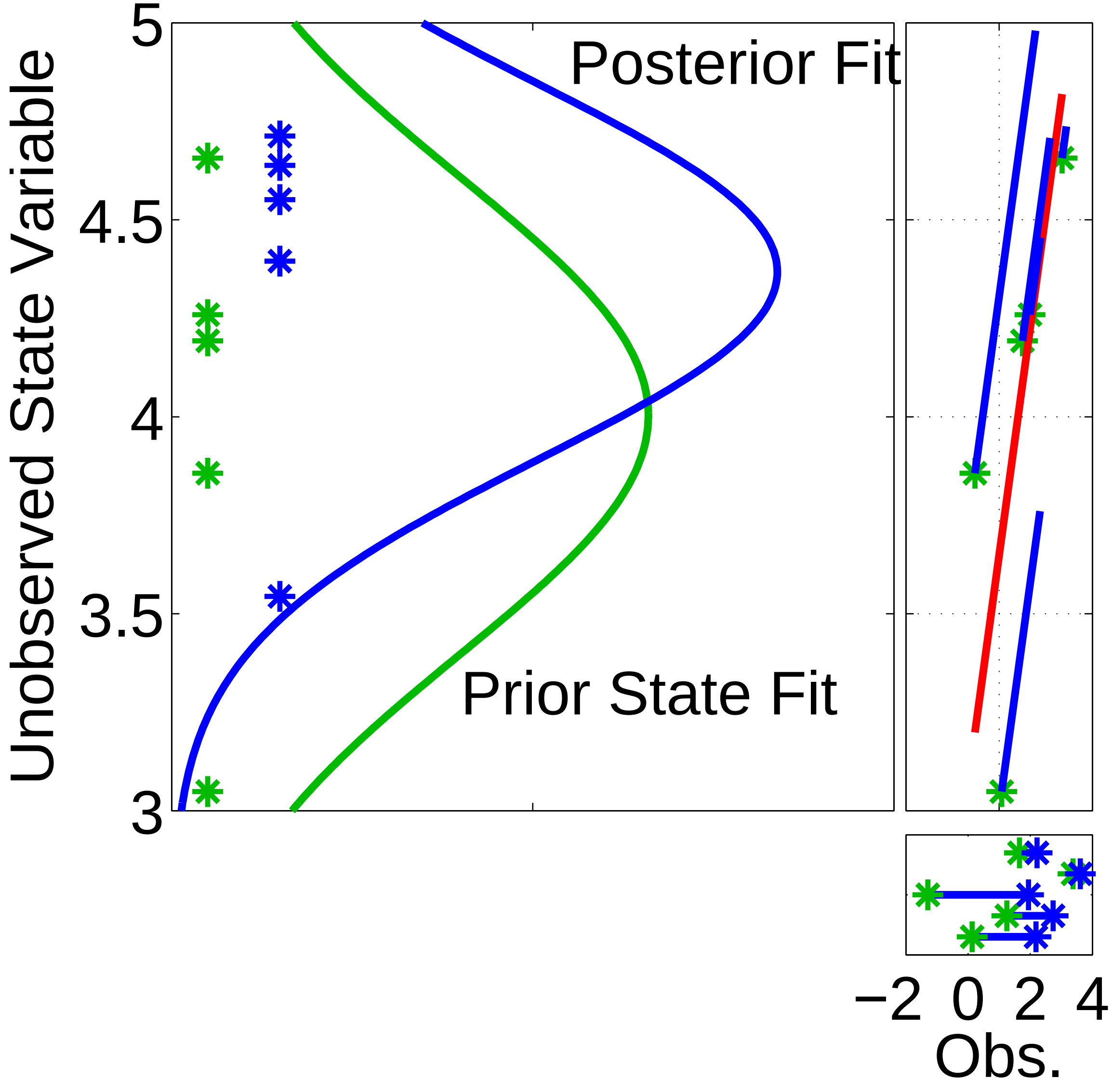


Now have an updated (posterior) ensemble for the unobserved variable.

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.

Multivariate assimilation with DART:

The regression code is trivial:

See *assim_tools/assim_tools_mod.f90*

First 10 executable lines of *subroutine update_from_obs_inc*.

To generate output from a multivariate Lorenz_63 experiment:

Run *./filter* in *models/lorenz_63/work*

Now do matlab diagnostics.

Does multivariate do better?

Be sure to record the error values for comparison.

Can you identify any obvious performance differences?

Multivariate assimilation in Lorenz_63:

What happens if not all state variables are observed?

1. Try observing only x and y (ignore z observations from above).

In *models/lorenz_63/work*
Edit *input.nml*
Change *obs_sequence_in_name* in *filter_nml* to *obs_seq.out.xy.*

Execute *./filter* to produce new assimilation.

Look at the error statistics and time series with matlab.

Record the error and spread values and compare to univariate case.

Multivariate assimilation in Lorenz_63:

2. Try observing only x (ignore y and z observations from above).

In models/lorenz_63/work
Edit input.nml
Change *obs_sequence_in_name* in *filter.nml* to
obs_seq.out.x

Execute *./filter* program to produce a new assimilation.

Look at the error statistics and time series with matlab.

Record the error and spread values and compare to univariate case.

What would happened if we made this into a univariate assimilation?
(Change the *cov_cutoff* back to small value for test).

Multivariate assimilation in Lorenz_63:

3. Try observing only z (ignore x and y observations from above).
(Change back to large value of *cov_cutoff* first).
In *models/lorenz_63/work*
Edit *input.nml*
Change *obs_sequence_in_name* in *filter.nml* to
obs_seq.out.z.

Execute the filter program to produce a new assimilation.

Look at the error statistics and time series with matlab.

Record the error and spread values and compare to univariate case.
Dynamics for x and y are symmetric; z can NOT distinguish them.
How do we want filter to handle this?
Does it do what we want in this case?