Ensemble Filters: general descriptions and some thoughts on types and interfaces

Anna Shlyaeva

15 June 2017

Ensemble filters: “types” generally used

Inputs:

x	state ensemble	`type(state_ensemble)`
$\bar{x}$	mean state	`type(state / state_vector?)`
$X = \frac{1}{\sqrt{k-1}}(x - \bar{x})$	ensemble of normalized perturbations	`type(state_ensemble)`

Compute:

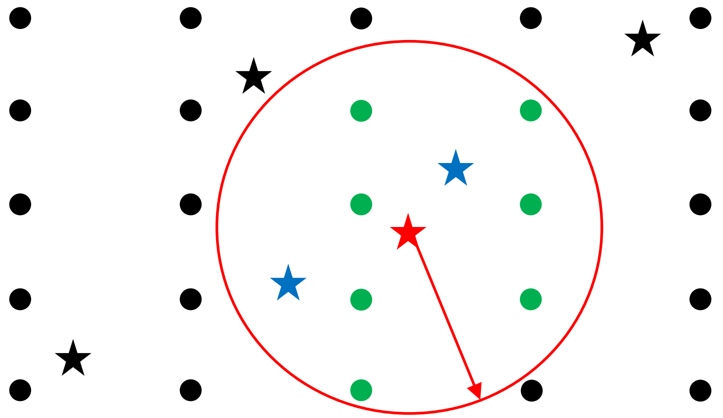
$\bar{y} = H(\bar{x})$	mean observation prior	`type(obs_vector)`
$Y = \frac{1}{\sqrt{k-1}}\left(H(x) - \overline{H(x)}\right)$	normalized obs prior perturbations	`type(obs_ensemble)`

(or $Y = \frac{1}{\sqrt{k-1}}\mathbf{H}(x - \bar{x})$ where $\mathbf{H}$ is observation operator Jacobian)

Ensemble filters algorithms: very generally

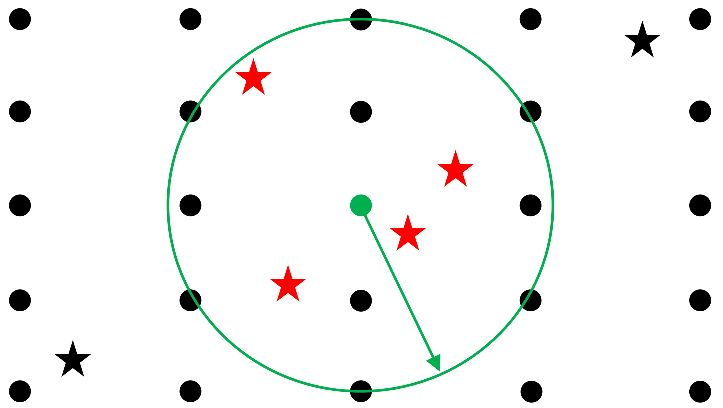
Serial filters (EnSRF run at NCEP, DART, Canadian pert-obs EnKF):

```
for each obs batch
          [opt] calc observation priors for the batch
          for each state var to update
                    update ensemble state var
          [opt] for each obs prior to update
                    [opt] update ensemble obs prior
```



LETKF (option in NOAA EnKF; UMD LETKF):

```
for each state var to update
          find observations that are used for the update
          update ensemble state var
```



Ensemble filters algorithms: very generally

For all algorithms we need to have methods to select subsets of observation vectors/ensembles and state vector/ensembles (or observation space and state space) to:

- Iterate:
  - over observation [batches] in serial filters: randomly, sequentially, ‘smartly’ (e.g., first use the observations with the greatest impact, or group observations that are nearby in the batch)
  - over gridpoints in LETKF
- Select obs/state within some distance from some location.

We would need locations (separate type) for both observation and state classes.

Q: is ’within some distance from some location’ always a simple concept (say, just great circle distance)?

More detailed algorithms: serial EnSRF

```
[opt] calc observation priors for all obs    // obs_operator%h_oper for both state_vector and state_ensemble

for each obs batch    // method to iterate over observation batches (select from obs_space?)
    [opt] calc observation priors for the batch
    calc $HBH^T + R = YY^T + R$ for the batch    // method (constructor?) to create obs_error_covariance from obs_ensemble (for $HBH^T = YY^T$). Note: localization happens here for batches of >1 obs; method to localize obs_error_covariance: need info about locations of obs.

    for each state var to update    // method to select state_vector and state_ensemble within localization distance from obs batch locations.

        calc $BH^T = XY^T$    // class for state_obs_error_covariance from state_ensemble and obs_ensemble? Note: localization happens here; method for localizing state_obs_error_covariance: need info about locations of obs and state

        calc $\mathrm{K} = BH^T(HBH^T + R)^{-1}$    // method for multiplying state_obs_error_covariance by inverse obs_error_covariance

        update $\bar{x} = \bar{x} + K(y - H(\bar{x}))$    // method for multiplying state_obs_error_covariance by obs_vector
        update $X = X - \alpha KY$    // method for multiplying state_obs_error_covariance by obs_ensemble

    [opt] for each obs prior to update
        [opt] update ensemble obs priors    // methods for multiplying obs_error_covariance by obs_vector and obs_ensemble
```

More detailed algorithms: LETKF

```
calc observation priors for all obs                 // obs_operator%h_oper for both state_vector and
                                                       state_ensemble

for each state var to update                        // method to iterate over gridpoints (select from state_space?)
        find ‘local’ observations                   // method to select obs_vector and obs_ensemble within localization
                                                       distance from gridpoint location.
        calc $Y^T R^{-1}$                           // this is the transpose of obs_ensemble (another class?). Needs a method
                                                       of multiplying itself by inverse obs_error_covariance. Note:
                                                       localization happens here: using $C \circ R^{-1}$ instead of $R^{-1}$. Where should it be?
        calc $\widetilde{P^a} = (Y^T R^{-1} Y + (k-1)\mathbf{I})^{-1}$   // this is kxk matrix. Class ensemble_error_covariance?
        calc $W^a = \left((k-1)\widetilde{P^a}\right)^{1/2}$            // method for square root or eig, eiv for ensemble_error_covariance
        calc $\overline{w^a} = \widetilde{P^a} Y^T R^{-1}\left(y - H(\bar{x})\right)$   // this is k-size vector. ensemble_vector class?
        calc $w^a$ as $W^a$ + $\overline{w^a}$
        calc $x = x + Xw^a$
```