A Hierarchical Bayes Ensemble Kalman Filter (HBEF):

Description of the HBEF package in the R language

by

Michael Tsyrulnikov[a,*], Alexander Rakitko[a]

[a] *HydroMetCenter of Russia*

Abstract

Here, we describe the software package (in the R language) developed to conduct the numerical experiments presented in the paper A Hierarchical Bayes Ensemble Kalman Filter submitted to Physica D. With this package, all the results of the numerical experiments reported in the paper can be reproduced.

1. Introduction

The paper **A Hierarchical Bayes Ensemble Kalman Filter** (submitted to Physica D) presents a new filter, the Hierarchical Bayes Ensemble Kalman Filter (HBEF), designed to extend the Ensemble Kalman Filter (EnKF) for problems with substantially uncertain ensemble covariances. The HBEF accommodates the conditions under which a high-dimensional EnKF actually works:

1. The ensemble size is small, so that the predictability-error covariances matrix P is substantially uncertain.
2. The model-error covariance matrix Q is variable in time and explicitly unknown.

The HBEF accounts for the uncertainty in P and Q and updates them along with the state x. In this update, ensemble members are used as generalized

[*] Corresponding author

Email address: mik.tsyrulnikov@gmail.com (Michael Tsyrulnikov) observations and ordinary observations are allowed to influence the covariances.

With the intention to study the performance of the HBEF in detail, it is tested in this study with a one-dimensional (scalar) model of “truth”. Thereby, the HBEF is compared with the following filters:

1. The reference Kalman Filter (KF) that “knows” the “true” model-error variance Q_k and is allowed to precisely compute the predictability-error variance P_k.
2. A variational filter (Var), where the background-error covariance matrix B_k is postulated to be constant $B_k = \bar{B}$.
3. A stochastic EnKF.
4. A predecessor of our filter, the Hierarchical Ensemble Kalman Filter (HEnKF) by Myrseth and Omre (2010).

2. Outlook

In the rest of this Description, we outline technical details needed to run the code (in the R language), briefly describe the program code, and show how it can be used to reproduce the results presented in the paper.

3. Technical details on how to interpret the code and run the program

Having installed a basic R interpreter (e.g. RStudio), you need to install the following standard packages:

```
library(mixAK)
library(MCMCpack)
```

E.g., in RStudio, type `install.packages('MCMCpack')`.

Then, you need to include the R source file that we have developed in this study:

```
source('functions.R')
```

Note that the file `functions.R` as well as the scripts described below are to be placed in the working directory.

4. General description of the main functions and how to invoke them

Here, we outline the R functions (placed in the file 'functions.R') that comprise our package "HBEF" and describe their input and output arguments.

4.1. Set up parameters

Function `create_parameters_universe_world` has no input arguments.

In the function's code, one may specify/change the following basic setup parameters:

the x's mean time scale (length scale) $\bar{\tau}_x$: `tau_x`,

the F's time scale τ_F: `tau_F`,

the Σ's time scale τ_Σ: `tau_Sigma`,

the Σ's standard deviation $\texttt{s.d.}\,\Sigma$: `std_Sigma`,

observation-error standard deviation $\texttt{s.d.}\,\eta \equiv \sqrt{R}$: `std_eta`,

as well as

the number of time moments in the experiment n_{time}: `time`,

ensemble size N,

number of independent runs (worlds) L,

coefficient of Q distortion $q_{distort}$: `distort_Q`,

and four seeds for pseudo-random number generation:

seed for initiation of the $\{F_k\}$ time series `seed_for_universe1`,

seed for initiation of the $\{\Sigma_k\}$ time series `seed_for_universe2`,

seed for initiation of the pseudo-random "truth" x_k and observations y_k `seed_for_filters`, and

seed for initiation of other pseudo-random sources `seed_for_filters`.

The function `create_parameters_universe_world` then calculates several internal parameters, which, together with the external ones, are encapsulated in the combined output argument `list` written, on return from the function, to the variable `parameters`.

4.2. Generate the sequences $\{F_k\}$, $\{\Sigma_k\}$, and $\{Q_k\}$

Function `generate_universe` has `parameters` as the input argument. It generates pseudo-random sequences $\{F_k\}$ and $\{\Sigma_k\}$, computes $\{Q_k\}$, and writes all of them to the output variable `universe`.

4.3. Generate a realization of the "truth" $\{x_k\}$ and observations $\{y_k\}$

Function `generate_world` has `parameters` and `universe` as the input arguments. It generates pseudo-random sequences $\{x_k\}$ and $\{y_k\}$ and writes them to the output variable `world`.

4.4. Run the KF

Function `filter_kf` has `world, universe, parameters, parameters_kf` as the input arguments. The KF-specific input variable `parameters_kf` contains `B_f_0` used as B_0 to start the filter.

Function `filter_kf` produces the output variable `output_kf`, which contains the time series (sequences) of:
deterministic background forecasts x_k^f,
deterministic analyses x_k^a,
prior background-error variances B_k^f,
posterior background-error variances $B_k^a = B_k^f$,
and
posterior analysis-error variances A_k.

4.5. Run the Var

Function `filter_var` has `world, universe, parameters, parameters_var` as the input arguments. The Var-specific input variable `parameters_var` contains `mean_B` used as the constant background-error variance $\bar{B}$.

Function `filter_var` produces the output variable `output_var`, which contains the time series (sequences) of:
deterministic background forecasts x_k^f,
deterministic analyses x_k^a,
prior background-error variances $B_k^f = \bar{B}$,
and
posterior background-error variances $B_k^a = \bar{B}$.

4.6. Run the EnKF

Function `filter_enkf` has `world, universe, parameters, parameters_enkf` as the input arguments. The EnKF-specific input variable `parameters_enkf` contains the variance inflation parameter `kappa` used to multiply the background-ensemble perturbations.

Function `filter_enkf` produces the output variable `output_enkf`, which contains the time series (sequences) of:

deterministic background forecasts x_k^f,
deterministic analyses x_k^a,
prior background-error variances B_k^f,
and
posterior background-error variances $B_k^a = B_k^f$.

4.7. Run the HEnKF

Function `filter_henkf` has `world, universe, parameters, parameters_henkf` as the input arguments. The HEnKF-specific input variable `parameters_henkf` contains `mean_B` ($\bar{B}$) used to start the filter and the Inverse Gamma dispersion parameter `theta` (θ) used to define the prior for B_k.

Function `filter_henkf` produces the output variable `output_henkf`, which contains the time series (sequences) of:
deterministic background forecasts x_k^f,
deterministic analyses x_k^a,
prior background-error variances B_k^f,
and
posterior background-error variances B_k^a.

4.8. Run the HBEF

Function `filter_hbef` has `world, universe, parameters, parameters_hbef` as the input arguments. The HBEF-specific input variable `parameters_hbef` contains:
the dispersion parameter for the Inverse Gamma distribution $Q|Q^f$: χ,
the dispersion parameter for the Inverse Gamma distribution $\Pi|\Pi^f$: ϕ,
the dispersion parameter for the Inverse Gamma distribution $P|\Pi$: θ,
size of the Monte-Carlo sample used to estimate the posterior mean $\bar{m}_{MC}^a$: `size_for_MC`,
the analysis-error variance A,
starting value for the analysis-error variance A: `mean_A`,
starting value for the model-error variance Q: `mean_Q`, and
the logical switch controlling whether the approximated posterior in to be used (=TRUTE if yes): `approximation`.

Function `filter_hbef` produces the output variable `output_hbef`, which contains the time series (sequences) of:
deterministic background forecasts x_k^f,
deterministic analyses x_k^a,

prior model-error variances Q_k^f,
posterior model-error variances Q_k^a,
prior predictability-error variances P_k^f,
posterior predictability-error variances P_k^a,
prior background-error variances B_k^f,
and
posterior background-error variances B_k^a.

5. Numerical experiments: "technology"

In the following sections, we outline the R code that enables the reproduction of all numerical experiments presented in the paper. Then, for each experiment, we give the experimental results that are to be reproduced if the user runs the program in the default setting. If the output we give coincides with that obtained by the user, then everything is OK and the user can change the setup parameters and run other experiments.

5.1. Computing and storing data needed to estimate the "true" background-error variances B_k (for all filters), the variances of the "truth" V_k, and the filters' outputs

Run the script `Calculate_data_for_B_evaluation.R`. Before running the script, you may change the number of time steps `parameters$time` and the number L of independent realizations (assimilation runs) `parameters$L`. As a result of an execution of the script, you may see the appearance, in the working directory, of 10 new data files like `X_true`, `B_a_hbef`, etc.

We recommend to run this script first because a number of other scripts (as indicated in each particular case below and in the header comments of the respective script).

5.2. Calculate RMSEs of the analyses for all filters

Just run the script `RMSE.R`.

The output should be as follows:

```
                    RMSE
KF              2.467379
Var             2.679794
EnKF            2.649394
```

```
HEnKF           2.581825
HBEF_simplest  2.520073
```

5.3. Plot a segment of the time series of F, σ, V, B

Plot a segment of the time series of: the model's operator F_k, the system-noise standard deviation σ_k, the variance of the "truth" V_k, and the estimated "true" background-error variance for the HBEF B_k.

Run the script `Timeseries.R`. Before running the script, you may change the number of time moments in the time series, `parameters$time` and select the segment to be plotted, `t1,t2` (within the range from 1 to `parameters$time`).

The output should be as in Fig.1 in the paper.

5.4. Quantile-quantile (q-q) plot for the unconditional prior distribution of the state

Compute the *q-q* (quantile-quantile) plot, which reflects the degree of Gaussianity, and the Gaussian (normal) approximation for the for the unconditional background-error distribution $p(x - m^f)$.

Run the script `qqplot_1.R`. Note that before running the script, you need to execute the script `Calculate_data_for_B_evaluation.R`.

Before running the script `qqplot_1.R`, you may change the the sample size parameters: `parameters$time` and `parameters$L`.

The output should be as in Fig.2(right) in the paper.

5.5. Quantile-quantile (q-q) plots for the conditional prior distribution of the state

Compute the *q-q* (quantile-quantile) plots, and the Gaussian (normal) approximations for the conditional background-error distribution $p(x - m^f|B)$.

Run the script `qqplot_2.R`. Note that before running the script, you need to execute the script `Calculate_data_for_B_evaluation.R`.

Before running the script `qqplot_2.R`, you may change the the sample size parameters: `parameters$time` and `parameters$L`.

The output should be as in Fig.2(left) in the paper.

5.6. Compute biases in the sample variances of the forecast-error ensemble

Compute and plot the biases in the sample variances S_k (along with their bootstrap confidence intervals) of the forecast-error ensemble for the EnKF and the HBEF. The errors are computed w.r.t.the estimated true B_k.

Run the script `ScndFltErrStats.R`. Before running the script, you may change the number of time steps *parameters$time* and the number of independent assimilation runs *parameters$L* (the number of "worlds" in the current "universe").

The output should be as in Fig.3 in the paper.

5.7. Compute RMSEs for the state x as functions of N

Compute and plot the RMSEs for the state x for the KF, Var, EnKF, HEnKF, and HBEF – as functions of the ensemble size N.

Run the script `RMSE_N.R`. Before running the script, you may change the range of N for which the computations are to be performed, `range`.

The output should be as in Fig.4(top, left) in the paper.

5.8. Compute and plot RMSEs for the state x as functions of the $\sqrt{R}$

Compute and plot the RMSEs for the state x for the KF, Var, EnKF, HEnKF, and HBEF – as functions of the observation-error standard deviation $\sqrt{R}$.

Run the script `RMSE_R.R`. Before running the script, you may change the values of $\sqrt{R}$ for which the computations are to be performed, `range`.

The output should be as in Fig.4(top, right) in the paper.

5.9. Compute and plot RMSEs for the state x as functions of π

Compute and plot the RMSEs for the state x for the KF, Var, EnKF, HEnKF, and HBEF – as functions of the degree of instability of the system measured by the probability π of the event $|F_k| > 1$: $\pi = P(|F_k| > 1)$.

Run the script `RMSE_pi.R`. Before running the script, you may change the values of π for which the computations are to be performed, `range`.

The output should be as in Fig.4(bottom, left) in the paper.

5.10. Compute and plot RMSEs for the state x as functions of s.d. Σ

Compute and plot the RMSEs for the state x for the KF, Var, EnKF, HEnKF, and HBEF – as functions of the degree of variability of the system-noise (model error) measured by the st.dev of Σ.

Run the script `RMSE_sdSigma.R`. Before running the script, you may change the values of s.d. Σ for which the computations are to be performed, `range`.

The output should be as in Fig.4(bottom, right) in the paper.

5.11. Estimation of the variances and their error statistics

The script `Evaluate_B.R` estimates the "true" background-error variances B_k for each filter separately, the variances of the "truth" V_k and computes the error statistics for B_k: bias and RMS of the predicted by the filters B w.r.t. the "true" one.

Execution: run the script `Evaluate_B.R`. Note that before running the script, you need to execute the script `Calculate_data_for_B_evaluation.R`.

Before running the script `Evaluate_B.R`, you may change the number of time steps `parameters$time` and the number L of independent realizations (assimilation runs) `parameters$L`. As a result of an execution of the script, you should obtain the following statistics:

```
      Mean(B_flt-B_true) RMSE(B_flt-B_true) MAE(B_flt-B_true) Mean(B_true)
KF           -0.01297727          0.5283938         0.3373792     6.705943
Var          -0.88641993          6.5327760         4.2170089     7.579386
EnKF         -1.35511119          6.1634808         3.9352956     7.530828
HEnKF        -1.79375823          4.3743991         2.9528241     7.238290
HBEF         -0.45360835          3.1726181         2.1345574     6.965745
```

These statistics appear in Table 1 in the paper.

5.12. Compute and plot the RMSEs of the EnKF's and HBEF's estimates of B_k

Compute and plot the of $\mathsf{rms}\,(B_k^a - B_k)$ for the EnKF and the HBEF.

Use output from the previous run of the script `ScndFltErrStats.R`.

In the paper, this is Fig.5.

5.13. Compute and plot RMSEs for the state x for misspecified model-error variance Q

Compute and plot the RMSEs for the state x as functions of the coefficient of distortion of Q—for KF, Var, EnKF, HEnKF, and three flavors of HBEF, specifically,

1. The simplest HBEF.
2. The HBEF with the Monte Carlo based analysis (the Monte-Carlo size is $M = 100$) with the optimally tuned θ.

3. The HBEF with the Monte Carlo based analysis (the Monte-Carlo size is $M = 100$) with $\theta = \infty$.

Run the script `RMSE_Q_distort.R`. Before running the script, you may change the values of n_{time} (`parameters$time` in the script, but not exceeding `time` set up in `functions.R`), the ensemble size N (`parameters$N` in the script), and the observation-error standard deviation $\sqrt{R}$ (`parameters$std_eta` in the script), for which the computations are to be performed.

In the paper, this is Fig.6.