A Langevinized Ensemble Kalman Filter for Large-Scale Dynamic Learning

Purdue University

November 4, 2024

Problem Setup

We are interested in conducting Bayesian on-line learning for the dynamic system:

$$\begin{aligned} x_t &= g(x_{t-1}) + u_t, \quad u_t \sim N(0, U_t), \\ y_t &= H_t x_t + \eta_t, \quad \eta_t \sim N(0, \Gamma_t), \end{aligned} \tag{1}$$

for stages $t = 1, 2, \ldots, T$, where $x_t \in \mathbb{R}^p$ and $y_t \in \mathbb{R}^{N_t}$ denote, respectively, the state and observations at stage t; and the dimension p, the total number of stages T, and the sample sizes N_t't are all assumed to be very large.

Notations: $f(y_t|x_t)$ denotes the likelihood function of y_t, $\pi(x_t|y_{1:t})$ denotes the filtering distribution at stage t given $y_{1:t} = \{y_1, y_2, \ldots, y_t\}$, and $\pi(x_t|y_{1:t-1}) = \int \pi(x_t|x_{t-1})\pi(x_{t-1}|y_{1:t-1})dx_{t-1}$ denotes the predictive distribution of x_t given $y_{1:t-1}$.

Problem Setup

The dynamic system (1) is of central importance: It models the data assimilation problems with linear measurement equations, and many other problems such as inverse problems and data assimilation with nonlinear measurement equations can be converted to it via appropriate transformations.

Ensemble Kalman Filter (EnKF)

- Initialization: Initialize an ensemble $\{x_0^{a,1}, x_0^{a,2}, \ldots, x_0^{a,m}\}$ of size m.
  For $t = 1, 2, \ldots, T$, do the following steps:
- Forecast: For $i = 1, 2, \ldots, m$, draw $u_t^i \sim N(0, U_t)$ and set $x_t^{f,i} = g(x_{t-1}^{a,i}) + u_t^i$; calculate the sample covariance matrix of $x_t^{f,1}, \ldots, x_t^{f,m}$ and denote it by C_t.
- Analysis: For $i = 1, 2, \ldots, m$, draw $\eta_t^i \sim N(0, \Gamma_t)$ and set $x_t^{a,i} = x_t^{f,i} + \hat{K}_t(y_t - H_t x_t^{f,i} - \eta_t^i) \stackrel{\Delta}{=} x_t^{f,i} + \hat{K}_t(y_t - y_t^{f,i})$, where $\hat{K}_t = C_t H_t^T (H_t C_t H_t^T + \Gamma_t)^{-1}$ forms an estimator for the Kalman gain matrix $K_t = S_t H_t^T (H_t S_t H_t^T + \Gamma_t)^{-1}$ and S_t denotes the covariance matrix of x_t^f.

Features of EnKF

The EnKF has two attractive features which make it extremely successful in dealing with high-dimensional data assimilation problems:

- It approximates each filtering distribution $\pi(x_t|y_{1:t})$ using an ensemble of particles. Since the ensemble size m is typically much smaller than p, it leads to dimension reduction and computational feasibility compared to the Kalman filter.
- In generating particles from each filtering distribution, it avoids covariance matrix decomposition compared to conventional particle filters. It is known that an LU-decomposition of the covariance matrix has a computational complexity of $O(p^3)$. Instead, EnKF employs a forecast-analysis procedure to generate particles, which has a computational complexity of $O(\max\{p^2 N_t, N_t^3\} + mpN_t)$ for m particles at stage t.

Features of EnKF

Despite its great successes, the performance of EnKF is sub-optimal: It converges only to a mean-field filter, which provides the optimal linear estimator of the conditional mean but not the filtering distribution except for linear systems in the large sample limit.

Linear Inverse Problem

Consider a Bayesian inverse problem for the regression

$$y = Hx + \eta, \tag{2}$$

where H is a known matrix, $\eta \sim N(0, \Gamma)$ for some covariance matrix Γ, $y \in \mathbb{R}^N$, and $x \in \mathbb{R}^p$ is an unknown continuous parameter vector.

Linear Inverse Problem: Formulation

Let $\pi(x)$ denote the prior density function of x, which is assumed to be differentiable with respect to x. Let $\pi(x|y)$ denote the posterior distribution.
We reformulate the model (2) as a state-space model through subsampling and Langevin diffusion:

$$\begin{aligned} x_t &= x_{t-1} + \epsilon_t \frac{n}{2N} \nabla \log \pi(x_{t-1}) + w_t, \\ y_t &= H_t x_t + v_t, \end{aligned} \tag{3}$$

where $w_t \sim N(0, \frac{n}{N}\epsilon_t I_p) = N(0, \frac{n}{N} Q_t)$, i.e., $Q_t = \epsilon_t I_p$, y_t denotes a block randomly drawn from $\{\tilde{y}_1, \ldots, \tilde{y}_B\}$, $v_t \sim N(0, V)$, and H_t is a submatrix of H extracted with the corresponding y_t.

Linear Inverse Problem: Algorithm

0. (Initialization) Start with an initial ensemble $x_0^{a,1}, x_0^{a,2}, \ldots, x_0^{a,m}$, where m denotes the ensemble size. For each stage $t = 1, 2, \ldots, T$ (first loop), do the following:
1. (Subsampling) Draw without replacement a mini-batch data, denoted by (y_t, H_t), of size n from the full dataset of size N.

- Set $Q_t = \epsilon_t I_p$, $R_t = 2V_t$, and the Kalman gain matrix $K_t = Q_t H_t^T (H_t Q_t H_t^T + R_t)^{-1}$. For each chain $i = 1, 2, \ldots, m$ (second loop), do steps 2-3:

2. (Forecast) Draw $w_t^i \sim N_p(0, \frac{n}{N} Q_t)$ and calculate

$$x_t^{f,i} = x_{t-1}^{a,i} + \epsilon_t \frac{n}{2N} \nabla \log \pi(x_{t-1}^{a,i}) + w_t^i. \tag{4}$$

3. (Analysis) Draw $v_t^i \sim N_n(0, \frac{n}{N} R_t)$ and calculate

$$x_t^{a,i} = x_t^{f,i} + K_t(y_t - H_t x_t^{f,i} - v_t^i) \stackrel{\Delta}{=} x_t^{f,i} + K_t(y_t - y_t^{f,i}). \tag{5}$$

Linear Inverse Problem: Convergence

Theorem 1. For Algorithm 1, if V is positive and definite, then the algorithm is reduced to a parallel pre-conditioned SGLD algorithm; i.e., for each chain $i \in \{1, 2, \ldots, m\}$,

$$x_t^{a,i} = x_{t-1}^{a,i} + \frac{\epsilon_t}{2} \Sigma_t \widehat{\nabla} \log \pi(x_{t-1}^{a,i}|y) + e_t, \tag{6}$$

where $\Sigma_t = \frac{n}{N}(I - K_t H_t)$ is a constant matrix of x, e_t is a zero mean Gaussian random error with covariance $\mathrm{Var}(e_t) = \epsilon_t \Sigma_t$, and $\widehat{\nabla} \log \pi(x_{t-1}^{a,i}|y) = \frac{N}{n} H_t^T V_t^{-1}(y_t - H_t x_{t-1}^{a,i}) + \nabla \log \pi(x_{t-1}^{a,i})$ represents an unbiased estimate of $\nabla \log \pi(x_{t-1}^{a,i}|y)$.

Data Assimilation: Bayesian Formulation

To motivate the development of the algorithm, we consider the Bayesian formula

$$\pi(x_t|y_{1:t}) = \frac{f(y_t|x_t)\pi(x_t|y_{1:t-1})}{\int f(y_t|x_t)\pi(x_t|y_{1:t-1})dx_t}, \tag{7}$$

which suggests that in order to get the filtering distribution $\pi(x_t|y_{1:t})$, the predictive distribution $\pi(x_t|y_{1:t-1})$ should be used as the prior at stage t.

Data Assimilation: Bayesian Formulation

The gradient $\nabla_{x_t} \log \pi(x_t|y_{1:t-1})$ can be estimated as follows:

$$\begin{aligned} &\nabla_{x_t} \log \pi(x_t|y_{1:t-1}) \\ &= \int \nabla_{x_t} \log \pi(x_t|x_{t-1}) \omega(x_{t-1}|x_t) \pi(x_{t-1}|y_{1:t-1}) dx_{t-1}, \end{aligned} \tag{8}$$

where $\omega(x_{t-1}|x_t) = \pi(x_{t-1}|x_t, y_{1:t-1})/\pi(x_{t-1}|y_{1:t-1}) = \pi(x_t|x_{t-1})/\pi(x_t|y_{1:t-1}) \propto \pi(x_t|x_{t-1})$, as $\pi(x_t|y_{1:t-1})$ is a constant with respect to x_{t-1}, given particle x_t and the data $y_{1:t-1}$. Given a set of samples $\mathcal{X}_{t-1} = \{x_{t-1,1}, x_{t-1,2}, \ldots, x_{t-1,m'}\}$ drawn from the filtering distribution $\pi(x_{t-1}|y_{1:t-1})$, an importance resampling procedure can be employed to draw a sample from $\pi(x_{t-1}|x_t, y_{1:t-1})$.

Data Assimilation: Bayesian Formulation

With the above formulas and Langevin dynamics, we can construct a dynamic system similar to (3) at stage t as

$$\begin{aligned} x_{t,k} &= x_{t,k-1} - \epsilon_t \frac{n_t}{2N_t} U_t^{-1}(x_{t,k-1} - g(\tilde{x}_{t-1,k-1})) + w_{t,k}, \\ y_{t,k} &= H_{t,k} x_{t,k} + v_{t,k}, \end{aligned} \tag{9}$$

for $k = 1, 2, \ldots$, where $x_{t,0} = g(x_{t-1}) + u_t$; $\tilde{x}_{t-1,k-1}$ denotes a sample drawn from $\pi(x_{t-1}|x_{t,k-1}, y_{1:t-1})$ at iteration k of stage t through the importance resampling procedure;
$w_{t,k} \sim N(0, \frac{n_t}{N_t}\epsilon_{t,k} I_p)$, $Q_{t,k} = \epsilon_{t,k} I_p$, and p is the dimension of x_t.

Data Assimilation: Algorithm

0. (Initialization) Start with an initial ensemble $x_{1,0}^{a,1}, x_{1,0}^{a,2}, \ldots, x_{1,0}^{a,m}$. Set $\mathcal{X}_t = \emptyset$ for $t = 1, 2, \ldots, T$. Set k_0 as the common burnin period for each stage t. For each stage $t = 1, 2, \ldots, T$ (first loop), and for each iteration $k = 1, 2, \ldots, \mathcal{K}$ (second loop), do the following:

1. (Subsampling) Draw without replacement a mini-batch sample, denoted by $(y_{t,k}, H_{t,k})$, of size n_t from the full dataset of size N_t.

- Set $Q_{t,k} = \epsilon_{t,k} I_p$, $R_t = 2V_t$, and the Kalman gain matrix $K_{t,k} = Q_{t,k} H_{t,k}^T (H_{t,k} Q_{t,k} H_{t,k}^T + R_t)^{-1}$.

  For each chain $i = 1, 2, \ldots, m$ (third loop), do steps 2-5:

2. (Importance resampling) If $t > 1$, calculate importance weights $\omega_{t,k-1,j}^i = \pi(x_{t,k-1}^{a,i} | x_{t-1,j}) = \phi(x_{t,k-1}^{a,i} : g(x_{t-1,j}), U_t)$ for $j = 1, 2, \ldots, |\mathcal{X}_{t-1}|$, where $\phi(\cdot)$ denotes a Gaussian density, and $x_{t-1,j} \in \mathcal{X}_{t-1}$ denotes the jth sample in $\mathcal{X}_{t-1}$; if $k = 1$, set $x_{t,0}^{a,i} = g(x_{t-1,\mathcal{K}}^{a,i}) + u_t^{a,i}$ and $u_t^{a,i} \sim N(0, U_t)$. Resample $s \in \{1, 2, \ldots, |\mathcal{X}_{t-1}|\}$ with a probability $\propto \omega_{t,k-1,s}^i$, i.e., $P(S_{t,k,i} = s) = \omega_{t,k-1,s}^i / \sum_{j=1}^{|\mathcal{X}_{t-1}|} \omega_{t,k-1,j}^i$, and denote the sample drawn from $\mathcal{X}_{t-1}$ by $\tilde{x}_{t-1,k-1}^i$.

Data Assimilation: Algorithm (continuation)

3. (Forecast) Draw $w_{t,k}^i \sim N_p(0, \frac{n}{N} Q_{t,k})$. If $t = 1$, set

$$x_{t,k}^{f,i} = x_{t,k-1}^{a,i} - \epsilon_{t,k} \frac{n_t}{2N_t} \nabla \log \pi(x_{t,k-1}^{a,i}) + w_{t,k}^i, \tag{10}$$

where $\pi(\cdot)$ denotes the prior distribution of x_1. If $t > 1$, set

$$x_{t,k}^{f,i} = x_{t,k-1}^{a,i} - \epsilon_{t,k} \frac{n_t}{2N_t} U_t^{-1}(x_{t,k-1}^{a,i} - g(\tilde{x}_{t-1,k-1}^i)) + w_{t,k}^i. \tag{11}$$

4. (Analysis) Draw $v_{t,k}^i \sim N_n(0, \frac{n}{N} R_t)$ and set

$$x_{t,k}^{a,i} = x_{t,k}^{f,i} + K_{t,k}(y_{t,k} - H_{t,k} x_{t,k}^{f,i} - v_{t,k}^i) \triangleq x_{t,k}^{f,i} + K_{t,k}(y_{t,k} - y_{t,k}^{f,i}). \tag{12}$$

5. (Sample collection) If $k > k_0$, add the sample $x_{t,k}^{a,i}$ into the set $\mathcal{X}_t$.

Data Assimilation: Theory

Theorem 2 We consider a dynamic system with $t = 1, 2, \ldots, T$ stages. Let $\pi_t = \pi(x_t|y_{1:t})$ denote the filtering distribution at stage t. Suppose that appropriate assumptions (see Zhang et al., 2024) hold, and N_t's are larger than certain threshold. If $\epsilon_{t,k} \propto \frac{1}{n_t^2 \log \mathcal{K}} k^{-\varpi}$ for some $\varpi \in (0, 1)$ and any $k \in \{1, 2, \ldots, \mathcal{K}\}$, then uniformly with dominating probability, for any $t \in \{1, 2, \ldots, T\}$, $x_{t,\mathcal{K}}^{a,i}$ follows a probability law $\tilde{\pi}_t$ and $\lim_{\mathcal{K}\to\infty} W_2(\tilde{\pi}_t, \pi_t) = 0$, where $W_2(\cdot, \cdot)$ denotes 2-Wasserstein distance between two distributions.

Data Assimilation: On Sample Degeneracy

It is known that sample degeneracy is an inherent default of sequential importance sampling (SIS), especially when the dimension of the system is high.
Fortunately, LEnKF is essentially immune to this issue. In LEnKF, the importance resampling procedure is to draw from $\mathcal{X}_{t-1}$ a particle which matches a given particle x_t in state propagation such that the gradient $\nabla_{x_t} \log \pi(x_t|y_{1:t-1})$ can be reasonably well estimated. By (7), $\pi(x_t|y_{1:t-1})$ works as the prior distribution of x_t for the filtering distribution $\pi(x_t|y_{1:t})$. Therefore, the effect of the importance resampling procedure on the performance of the algorithm is limited if the sample size N_t is reasonably large at each stage t.

Data Assimilation: Uncertainty Quantification

LEnKF is run in ensemble which provides a convenient way for uncertainty quantification. At each stage and for each chain, the state can be estimated by averaging over iterations. The state estimates can then be further averaged over the chains. It is easy to see that the central limit theorem holds for this chain-averaged estimator approximately given the weak dependence between different chains, and uncertainty quantification can be made accordingly.

Lorenz-96 Model

The Lorenz-96 model was developed by Edward Lorenz in 1996 to study difficult questions regarding predictability in weather forecasting. The model is given by

$$\frac{dx^i}{dt} = (x^{i+1} - x^{i-2})x^{i-1} - x^i + F, \quad i = 1, 2, \cdots, p,$$

where $F = 8$, $p = 40$, and it is assumed that $x^{-1} = x^{p-1}$, $x^0 = x^p$, and $x^{p+1} = x^1$. Here F is known as a forcing constant, and $F = 8$ is a common value known to cause chaotic behavior.

Lorenz-96 Model

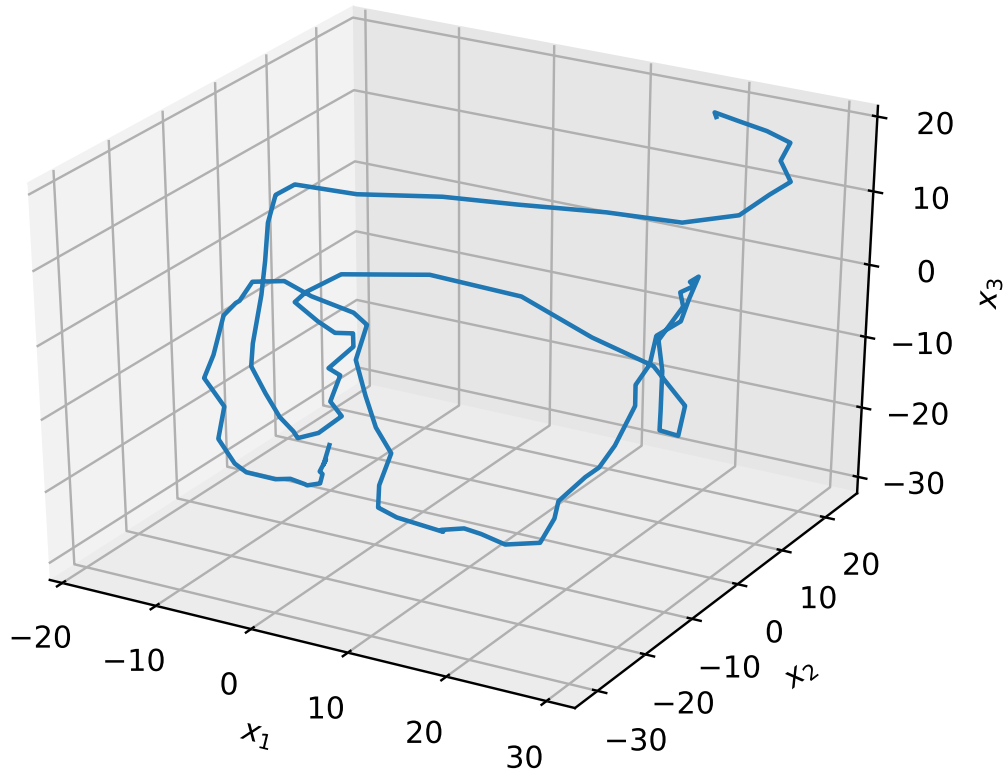

Figure: Chaotic path of the partial state variables (X_t^1, X_t^2, X_t^3) for $t = 1, 2, \ldots, 100$, simulated from the Lorenz-96 Model.

Lorenz-96 Model: State estimation

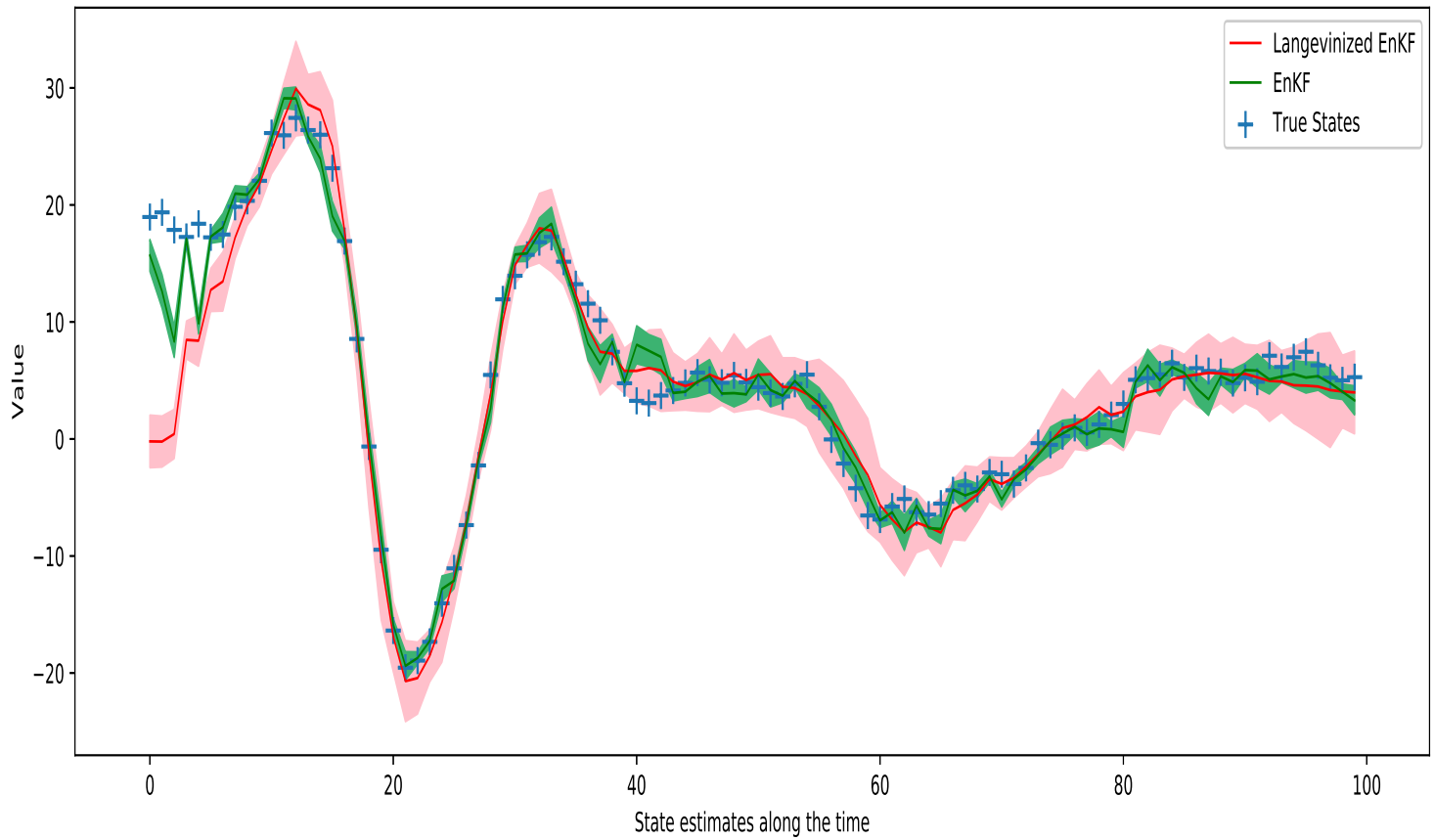

Figure: State estimates produced by LEnKF (red) and EnKF (green) for the Lorenz-96 model along with $t = 1, 2, \ldots, 100$.

Lorenz-96 Model: State estimation

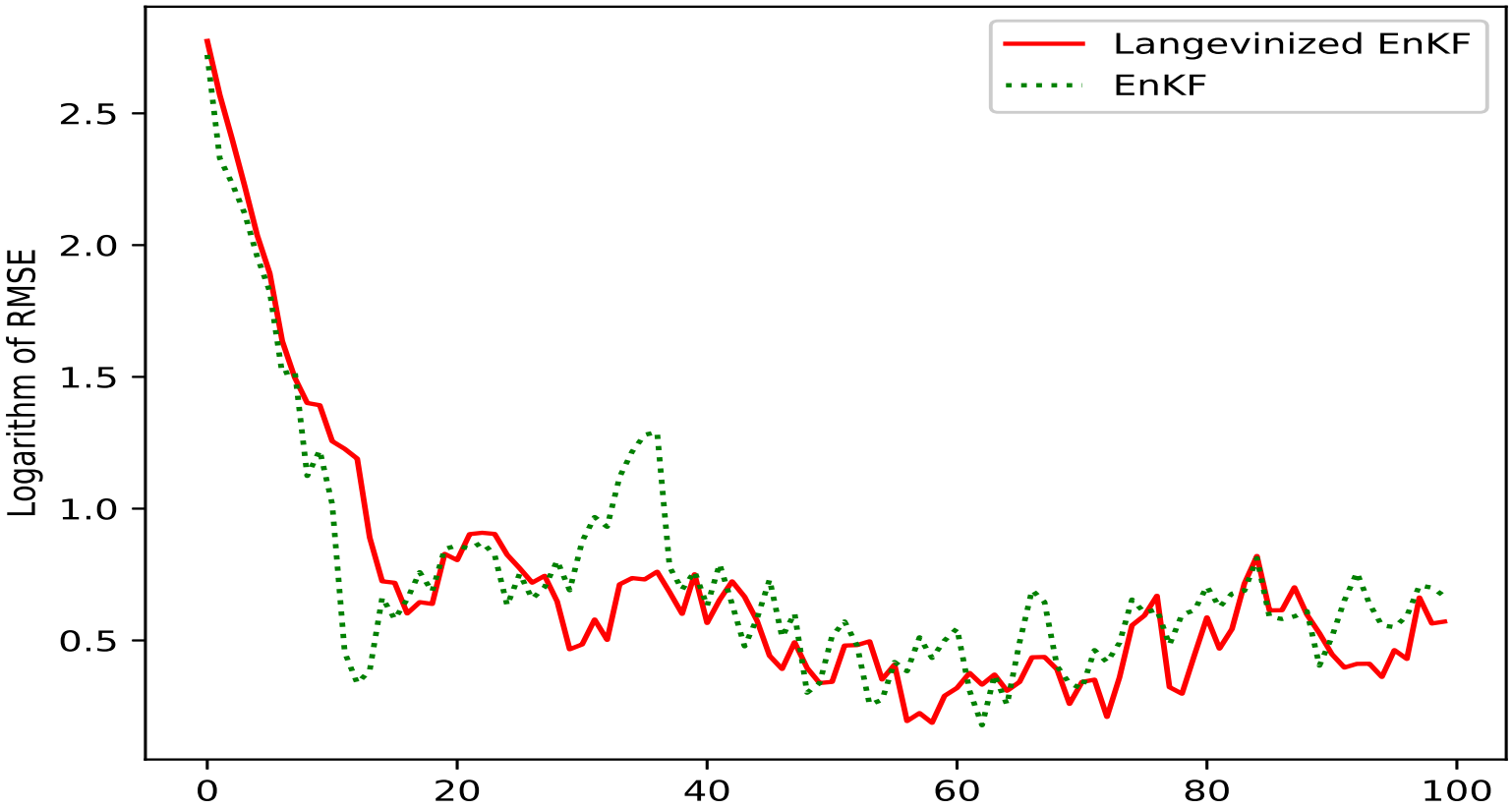

Figure: State estimates produced by LEnKF (red) and EnKF (green) for the Lorenz-96 model: $\log(\text{RMSE}_t)$ along with stage t.

Lorenz-96 Model: Coverage Prob.

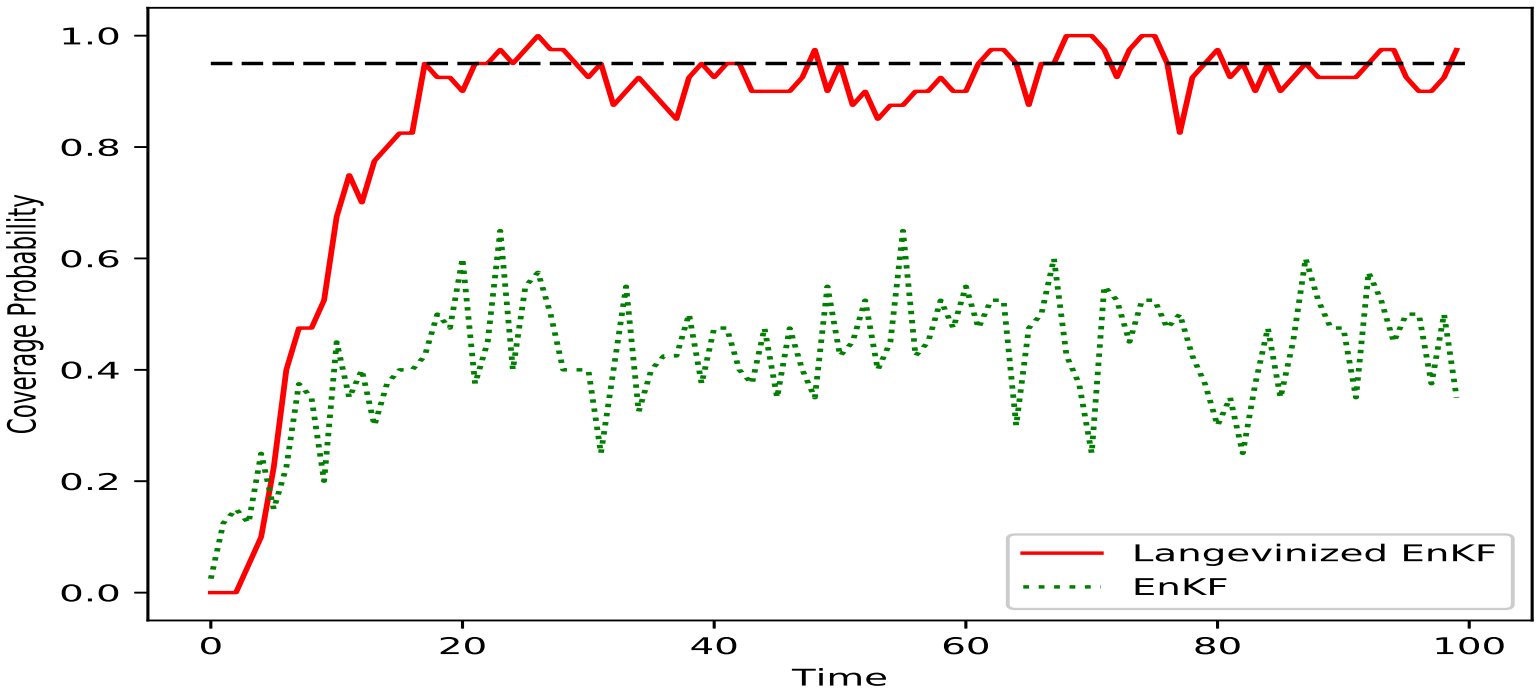

Figure: Coverage probabilities of the 95% confidence intervals produced by LEnKF (red) and EnKF (green) for Lorenz-96 Model for along with stage $t = 1, 2, \ldots, 100$: results for one dataset.

Lorenz-96 Model: Coverage Prob.

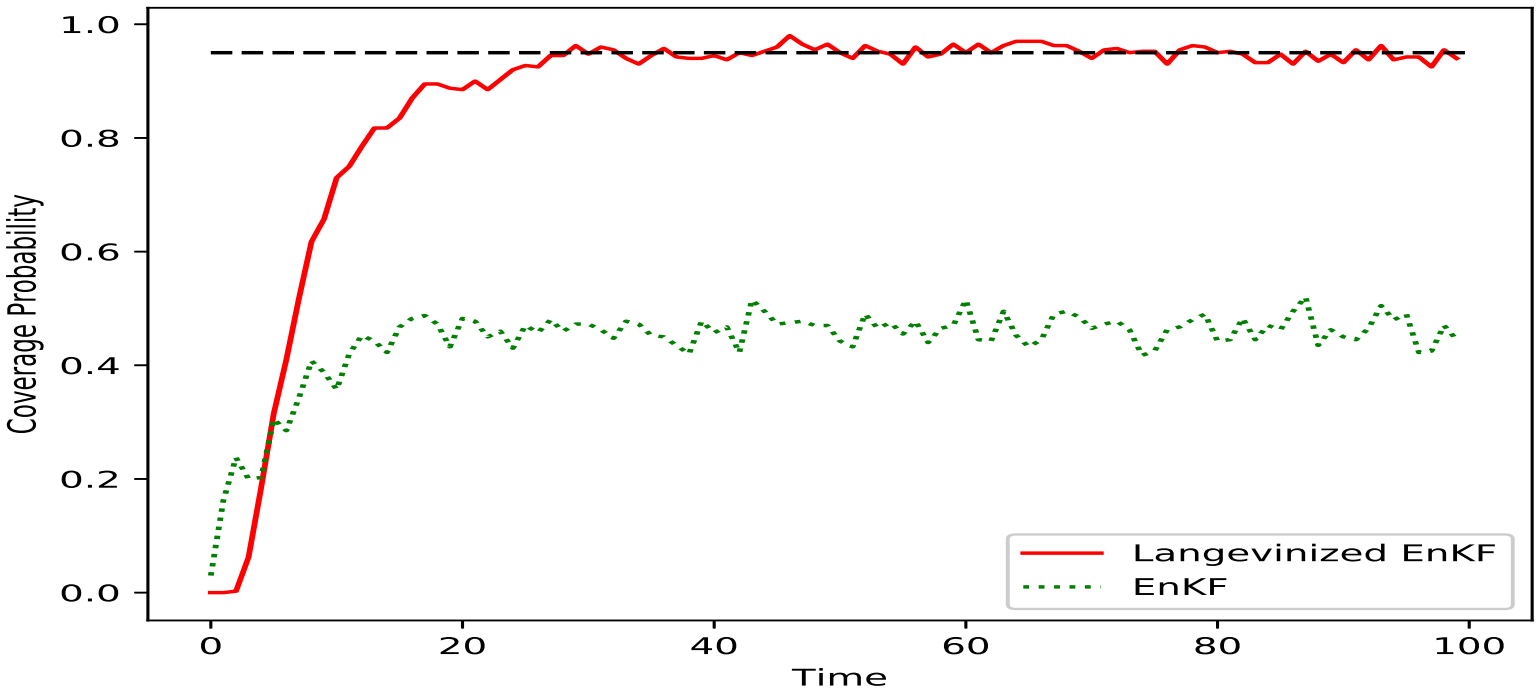

Figure: Coverage probabilities of the 95% confidence intervals produced by LEnKF (red) and EnKF (green) for Lorenz-96 Model for along with stage $t = 1, 2, \ldots, 100$: results averaged over 10 datasets.

Lorenz-96 Model: Coverage Prob.

Table: Comparison of the EnKF and LEnKF, where the averages over 10 independent datasets are reported with the standard deviation given in the parentheses. The CPU time was recorded for a single run of the method.

	LEnKF		EnKF
	$k_0 = \mathcal{K}/2$	$k_0 = \mathcal{K} - 1$	
Am-RMSE	1.702(0.0343)	1.714(0.0360)	1.722(0.0230)
Am-CP	0.948(0.0028)	0.947(0.0034)	0.460(0.0029)
CPU(s)	6.37(0.3942)	3.350(0.0807)	0.817(0.0426)

LEnKF with Unknown Parameters

(i) (Initialization) Initialize an ensemble $\{x_{1,0}^{a,1}, x_{1,0}^{a,2}, \ldots, x_{1,0}^{a,m}\}$ of size m by drawing from the prior $\pi(x_1)$, set $\mathcal{X}_t = \emptyset$ for $t = 1, 2, \ldots, T$, and set k_0 as the common burn-in period of each stage.

For $t = 1, 2, \ldots, T$**, do:**

For $k = 1, 2, \ldots, \mathcal{K}$**, do:**

(ii) (Subsampling) Draw a mini-batch data (i.e., a block), denoted by $(y_{t,k}, H_{t,k})$, of size n_t from the full dataset of size N_t. Set $Q_{t,k} = \epsilon_{t,k} I_p$, $R_t = 2V_t$, and the Kalman gain matrix $K_{t,k} = Q_{t,k} H_{t,k}^T (H_{t,k} Q_{t,k} H_{t,k}^T + R_t)^{-1}$.

For $i = 1, 2, \ldots, m$**, do:**

(iii) (Importance resampling) If $t > 1$, calculate importance weights $\omega_{t,k-1,j}^i = \pi(x_{t,k-1}^{a,i} | x_{t-1,j}) = \phi(x_{t,k-1}^{a,i} : g(x_{t-1,j}), U_t)$ for each sample $x_{t-1,j} \in \mathcal{X}_{t-1}$, where j ranges from 1 to $|\mathcal{X}_{t-1}|$, $x_{t,k-1}^{a,i}$ is the sample generated in the analysis step by chain i at iteration $k-1$ of stage t, and $\phi(\cdot)$ denotes a Gaussian density; if $k = 1$, set $x_{t,0}^{a,i} = g(x_{t-1,\mathcal{K}}^{a,i}) + u_t^{a,i}$ and $u_t^{a,i} \sim N(0, U_t)$. Resample $s \in \{1, 2, \ldots, |\mathcal{X}_{t-1}|\}$ with a probability $\propto \omega_{t,k-1,s}^i$, i.e., $P(S_{t,k,i} = s) = \omega_{t,k-1,s}^i / \sum_{j=1}^{|\mathcal{X}_{t-1}|} \omega_{t,k-1,j}^i$, and denote the sample drawn from $\mathcal{X}_{t-1}$ by $\tilde{x}_{t-1,k-1}^i$.

LEnKF with Unknown Parameters

(iv) (Forecast) Draw $w_{t,k}^i \sim N_p(0, \frac{n_t}{N_t} Q_{t,k})$. If $t = 1$, set

$$x_{t,k}^{f,i} = x_{t,k-1}^{a,i} - \epsilon_{t,k} \frac{n_t}{2N_t} \nabla \log \pi(x_{t,k-1}^{a,i}) + w_{t,k}^i, \tag{13}$$

where $\pi(\cdot)$ denotes the prior distribution of x_1. If $t > 1$, set

$$x_{t,k}^{f,i} = x_{t,k-1}^{a,i} - \epsilon_{t,k} \frac{n_t}{2N_t} U_t^{-1}(x_{t,k-1}^{a,i} - g(\tilde{x}_{t-1,k-1}^i)) + w_{t,k}^i. \tag{14}$$

(v) (Analysis) Draw $v_{t,k}^i \sim N_n(0, \frac{n_t}{N_t} R_t)$ and set

$$x_{t,k}^{a,i} = x_{t,k}^{f,i} + K_{t,k}(y_{t,k} - H_{t,k} x_{t,k}^{f,i} - v_{t,k}^i) \stackrel{\Delta}{=} x_{t,k}^{f,i} + K_{t,k}(y_{t,k} - y_{t,k}^{f,i}). \tag{15}$$

(vi) (Sample collection) If $k > k_0$, add the sample $x_{t,k}^{a,i}$ into the set $\mathcal{X}_t$.
End For

(vii) (Parameter updating) If $t > 1$, calculate

$$\boldsymbol{\theta}_{t,k} = \boldsymbol{\theta}_{t,k-1} + \gamma_{t,k} \Psi(y_{t,k}, \tilde{\boldsymbol{x}}_{t-1,k-1}, \boldsymbol{\theta}_{t,k-1}), \tag{16}$$

where $\boldsymbol{\theta}_{t,k}$ denotes the estimate of $\boldsymbol{\theta}$ obtained at iteration k of stage t,

$$\Psi(y_{t,k}, \tilde{\boldsymbol{x}}_{t-1,k-1}, \boldsymbol{\theta}_{t,k-1}) = \frac{N_t}{m n_t} \sum_{i=1}^{m} \nabla_{\boldsymbol{\theta}} \log \pi(y_{t,k} | \tilde{x}_{t-1,k-1}^i, \boldsymbol{\theta}_{t,k-1}), \tag{17}$$

End For
End For

Discussion

- We propose LEnKF as a scalable particle filter by reformulating EnKF under the framework of Langevin dynamics. LEnKF turns out to be a sequential preconditioned SGLD algorithm but with an accelerated ensemble implementation as for EnKF, and it converges to the correct filtering distribution under the big data scenario where the dynamic system consists of a large number of stages and there are a large number of observations at each stage.
- LEnKF can be applied to state estimation for both inverse and data assimilation problems, and uncertainty quantification of the state estimates.
- LEnKF is not only scalable with respect to the state dimension and sample size, but also tends to be immune to the sample degeneracy issue that is often suffered by conventional particle filters.
- LEnKF can be easily extended to dynamic systems with nonlinear measurement equations.