Transport methods for Bayesian computation: Part 2

Youssef Marzouk
joint work with Daniele Bigoni, Matthew Parno,
Alessio Spantini, & Olivier Zahm

Department of Aeronautics and Astronautics
Center for Computational Engineering
Statistics and Data Science Center
Massachusetts Institute of Technology
http://uqgroup.mit.edu

Support from AFOSR, DARPA, DOE

25–26 September 2019
What to do when $T \# \eta \neq \pi$?

- Maybe close enough? Can evaluate variance diagnostic $\text{Var}_\eta[\log(\eta / T^{-1}_\# \bar{\pi})]$, bound $\text{Tr}(H_{T^{-1}_\# \pi})$, etc.

- Enrich $T$, e.g., add a layer or expand $T^h_\Delta$ in the given layer

- **Sample the pullback:** treat $T^{-1}_\# \pi$ with an asymptotically exact scheme, e.g., Markov chain Monte Carlo
What to do when $\mathcal{T}_\# \eta \neq \pi$?

- Maybe close enough? Can evaluate variance diagnostic
  $\mathbb{V} \text{ar}_\eta[\log(\eta/\mathcal{T}_\#^{-1} \pi)]$, bound $\text{Tr}(H_{\mathcal{T}_\#^{-1} \pi})$, etc.

- Enrich $T$, e.g., add a layer or expand $\mathcal{T}_\Delta^h$ in the given layer

- **Sample the pullback:** treat $\mathcal{T}_\#^{-1} \pi$ with an asymptotically exact scheme, e.g., Markov chain Monte Carlo

One possible construction: **transport-accelerated MCMC**

- Transport map “preconditions” MCMC target; use MCMC iterates in maps-from-samples construction

- Can be understood in the framework of *adaptive MCMC*
Effective MCMC proposal = adapted to the target

- Can we transform proposals or, equivalently, targets for better sampling?
Recall maps-from-samples construction

\[
\min_{S \in S^h_\Delta} \mathcal{D}_{KL}(S\# \pi \| \eta) = \min_{S \in S^h_\Delta} \mathcal{D}_{KL}(\pi \| S^{-1}\# \eta)
\]

- Suppose we have Monte Carlo samples \( \{x_i\}_{i=1}^M \sim \pi \)
- For standard Gaussian \( \eta \), this problem is \textbf{convex} and \textbf{separable} for any \( \pi \)
- This is \textit{density estimation via transport!} (cf. Tabak & Turner 2013)
Recall maps-from-samples construction

\[
\min_{S \in S^h_\triangle} \mathcal{D}_{KL}(S \| \eta) = \min_{S \in S^h_\triangle} \mathcal{D}_{KL}(\pi \| S^{-1} \eta)
\]

- Suppose we have Monte Carlo samples \( \{x_i\}_{i=1}^M \sim \pi \)
- For standard Gaussian \( \eta \), this problem is convex and separable for any \( \pi \)
- This is density estimation via transport! (cf. Tabak & Turner 2013)
- Equivalent to maximum likelihood estimation of \( S \)

\[
\hat{S} \in \arg \max_{S \in S^h_\triangle} \frac{1}{M} \sum_{i=1}^M \log S^{-1} \eta(x_i), \quad \eta = \mathcal{N}(0, I_n),
\]

- Each component \( \hat{S}^k \) of \( \hat{S} \) can be computed separately, via smooth convex optimization

\[
\hat{S}^k \in \arg \min_{S^k \in S^h_{\triangle,k}} \frac{1}{M} \sum_{i=1}^M \left( \frac{1}{2} S^k(x_i)^2 - \log \partial_k S^k(x_i) \right)
\]
View $\hat{S}_\# \pi$ as the “preconditioned” target

- In the MCMC setting, $\{x_i\}_{i=1}^M$ comprises dependent MCMC samples
- $\hat{S}_\# \pi$ may be far from standard Gaussian for small $M$ and/or crude $S^h_\triangle$
Map-accelerated MCMC

- **Ingredient #1: static map**
  - Perform MCMC in the reference space, on the “preconditioned” density
  - Simple proposal in reference space (e.g., random walk) corresponds to a more complex/tailored proposal on target
**Map-accelerated MCMC**

- **Ingredient #1: static map**
  - Perform MCMC in the reference space, on the “preconditioned” density
  - Simple proposal in reference space (e.g., random walk) corresponds to a more complex/tailored proposal on target

\[
\alpha = \frac{\pi(S^{-1}(r')) | \nabla S^{-1} |_{r'} q_r(r | r')}{\pi(S^{-1}(r)) | \nabla S^{-1} |_{r} q_r(r' | r)}
\]

simple proposal \( q_r \) on pushforward of target through map
Map-accelerated MCMC

- **Ingredient #1: static map**
  - Perform MCMC in the reference space, on the “preconditioned” density
  - Simple proposal in reference space (e.g., random walk) corresponds to a more complex/tailored proposal on target

\[ \alpha = \frac{\pi(S^{-1}(r')) | \nabla S^{-1} |_{r'} q_r(r | r')} {\pi(S^{-1}(r)) | \nabla S^{-1} |_{r} q_r(r' | r)} \]

more complex proposal, directly on target distribution
Map-accelerated MCMC

- **Ingredient #2: adaptive map**
  - Update the map with each MCMC iteration: *more samples, more accurate $E_{\pi}$, better $S$*
  - Adaptive MCMC [Haario 2001, Andrieu 2006], but with nonlinear transformation to capture non-Gaussian structure
Map-accelerated MCMC

- **Ingredient #2: adaptive map**
  - Update the map with each MCMC iteration: *more samples, more accurate $E_{\pi}$, better $S$*
  - Adaptive MCMC [Haario 2001, Andrieu 2006], but with nonlinear transformation to capture non-Gaussian structure
• **Ingredient #3: global proposals**
  – If the map becomes sufficiently accurate, would like to avoid random-walk behavior

reference RW proposal  
mapped RW proposal
• **Ingredient #3: global proposals**
  – If the map becomes sufficiently accurate, would like to avoid random-walk behavior
Map-accelerated MCMC

- **Ingredient #3: global proposals**
  - If the map becomes sufficiently accurate, would like to avoid random-walk behavior
  - Solution: *delayed rejection* MCMC [Mira 2001]
  - First proposal = independent sample from $\eta$ (global, more efficient); second proposal = random walk (local, more robust)

- Entire scheme is provably *ergodic* with respect to the exact posterior measure [Parno & M, *SIAM JUQ* 2018]
  - Requires enforcing some regularity conditions on maps, to preserve tail behavior of transformed target
Example: biological oxygen demand model

- Likelihood model:
  \[ d = \theta_1 (1 - \exp(-\theta_2 x)) + \epsilon \]
  \[ \epsilon \sim N(0, 2 \times 10^{-4}) \]

- 20 noisy observations at
  \[ x = \left\{ \frac{5}{5}, \frac{6}{5}, \ldots, \frac{25}{5} \right\} \]

- Degree-three polynomial map

True posterior density
Results: MCMC chain

$\theta_1$ component of MCMC chain

DRAM

NUTS

TM+MIX

TM+DRG

Marzouk et al.

ETICS Research School

7 / 49
Transformed distribution

Original posterior $\pi$

Pushforward of posterior through \textit{learned} map, $S_{\#}\pi$
Results: ESS per computational effort

ESS/(1,000 Evaluations) – $\theta_1$

<table>
<thead>
<tr>
<th>Method</th>
<th>ESS/(1,000 Evaluations)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM+DR</td>
<td>161</td>
</tr>
<tr>
<td>TM+NUTS</td>
<td>57</td>
</tr>
<tr>
<td>TM+LAN</td>
<td></td>
</tr>
<tr>
<td>NUTS</td>
<td>179</td>
</tr>
<tr>
<td>DRAM</td>
<td></td>
</tr>
</tbody>
</table>

ESS/second– $\theta_1$

<table>
<thead>
<tr>
<th>Method</th>
<th>ESS/second</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM+DRG</td>
<td>1468</td>
</tr>
<tr>
<td>TM+DRL</td>
<td>487</td>
</tr>
<tr>
<td>TM+MIX</td>
<td>1495</td>
</tr>
<tr>
<td>NUTS</td>
<td>57</td>
</tr>
<tr>
<td>DRAM</td>
<td>127</td>
</tr>
</tbody>
</table>
Example #2: predator-prey model

- Six parameter ODE population model
  \[ \frac{dP}{dt} = rP \left( 1 - \frac{P}{K} \right) - s \frac{PQ}{a + P} \]
  \[ \frac{dQ}{dt} = u \frac{PQ}{a + P} - vQ \]

- Five noisy observations of both populations
- Infer 6 parameters + 2 initial values; uniform priors
Predator-prey model: chains

![Graph showing MCMC Steps with different algorithms (DRAM, sMMALA, AMALA, TMDRG, TMDRL, TMRWM)]

- DRAM
- sMMALA
- AMALA
- TMDRG
- TMDRL
- TMRWM
Example: maple sap dynamics model

- Coupled PDE system for ice, water, and gas locations [Ceseri & Stockie 2013]
- Measure gas pressure in vessel
- Infer 10 physical model parameters
- Very challenging posterior!

Image from Ceseri and Stockie, 2013
Maple posterior distribution

\[
\begin{align*}
\theta_1 & \sim \text{Normal}(\mu_1, \sigma_1) \\
\theta_2 & \sim \text{Normal}(\mu_2, \sigma_2) \\
\theta_3 & \sim \text{Normal}(\mu_3, \sigma_3) \\
\theta_4 & \sim \text{Normal}(\mu_4, \sigma_4) \\
\theta_5 & \sim \text{Normal}(\mu_5, \sigma_5) \\
\theta_6 & \sim \text{Normal}(\mu_6, \sigma_6) \\
\theta_7 & \sim \text{Normal}(\mu_7, \sigma_7) \\
\theta_8 & \sim \text{Normal}(\mu_8, \sigma_8) \\
\theta_9 & \sim \text{Normal}(\mu_9, \sigma_9) \\
\theta_{10} & \sim \text{Normal}(\mu_{10}, \sigma_{10})
\end{align*}
\]
Results: ESS per computational effort

ESS/(10,000 Evaluations)

ESS/(1000 seconds)
Comments on MCMC with transport maps

Useful characteristics of the algorithm:

- Map construction is easily parallelizable
- Requires no gradients from posterior density

Generalizes many current MCMC techniques:

- Adaptive Metropolis: map enables non-Gaussian proposals and a natural mixing between local and global moves
- Manifold MCMC [Girolami & Calderhead 2011]: map also defines a Riemannian metric
Looking to higher dimensions: regularized estimation of $S$

For simplicity, consider map components $S^k(x) = \sum_j \beta_j \psi_j(x_{1:k-1}) + \alpha_k x_k$

$$\hat{S}^k \in \arg\min_{S^k \in S_{\Delta,k}^h} \frac{1}{N} \sum_{i=1}^N \left( \frac{1}{2} S^k(x_i)^2 - \log \partial_k S^k(x_i) \right) + \lambda_N \|\beta\|_1$$

Assume sub-Gaussian $\pi$ and basis functions $\psi_j(x)$

Theorem [BZM]

For polynomial maps of degree $m$ with sparsity $s$, with high probability

$$\mathbb{E}_\pi \left[ D_{KL}(\pi(x_k|x_{1:k-1}) \| \hat{S}_k^\# \eta) \right] \lesssim \sqrt{\frac{s^2 m \log k}{N}}$$

Takeaways

- Accurate estimation is feasible in high dimensions with $N \ll k$
- From factorization property of density, error in conditionals ensures

$$D_{KL}(\pi \| \hat{S}_k^\# \eta) \lesssim d \sqrt{\frac{s^2 m \log d}{N}}$$
Nonlinear/non-Gaussian state-space model:

- Transition density $\pi_{Z_k|Z_{k-1}}$
- Observation density (likelihood) $\pi_{Y_k|Z_k}$

Interested in recursively updating the full Bayesian solution:

$$\pi_{Z_{0:k}|y_{0:k}} \rightarrow \pi_{Z_{0:k+1}|y_{0:k+1}}$$ (smoothing)
Next topic: ensemble filtering via transport

- **Nonlinear/non-Gaussian** state-space model:
  - Transition density $\pi_{Z_k|Z_{k-1}}$
  - Observation density (likelihood) $\pi_{Y_k|Z_k}$

- Interested in **recursively** updating the **full Bayesian solution**:
  $\pi_{Z_{0:k}|y_{0:k}} \rightarrow \pi_{Z_{0:k+1}|y_{0:k+1}}$ (smoothing)

- Or focus on approximating the **filtering distribution**:
  $\pi_{Z_k|y_{0:k}} \rightarrow \pi_{Z_{k+1}|y_{0:k+1}}$ (marginals of the full Bayesian/smoothing solution)
Consider the filtering of state-space models with:

1. High-dimensional states
2. Challenging nonlinear dynamics (e.g., chaotic systems)
3. Intractable transition kernels: can only obtain forecast samples, i.e., draws from $\pi_{Z_{k+1}|z_k}$
4. Limited model evaluations, e.g., small ensemble sizes
5. Sparse and local observations in space/time

These constraints reflect typical challenges faced in numerical weather prediction, geophysical data assimilation.
State-of-the-art results (in terms of tracking) are typically obtained with the ensemble Kalman filter (EnKF).

Forecast step:

\[ \pi Z_{k-1} | Y_{0:k-1} \]

Analysis step:

\[ \pi Z_k | Y_{0:k-1} \quad \pi Z_k | Y_{0:k} \]

Bayesian inference

Move samples via an affine transformation; no weights or resampling!
Yet ultimately inconsistent: does not converge to the true posterior.
Ensemble Kalman filter

- State-of-the-art results (in terms of tracking) are typically obtained with the ensemble Kalman filter (EnKF).

\[ \pi Z_{k-1} | Y_{0:k-1} \rightarrow \] forecast step

\[ \pi Z_k | Y_{0:k-1} \rightarrow \] analysis step

Bayesian inference

- Move samples via an **affine** transformation; no weights or resampling!
- Yet ultimately **inconsistent**: does not converge to the true posterior.

Can we improve and **generalize** the EnKF, preserving scalability, via **nonlinear** transformations?
Assimilation step

At any assimilation time $k$, we have a Bayesian inference problem:

\[
\begin{align*}
\pi_X &\text{ prior} \\
\pi_X|Y = y^* &\text{ posterior}
\end{align*}
\]

- $\pi_X$ is the forecast distribution on $\mathbb{R}^n$
- $\pi_{Y|X}$ is the likelihood of the observations $Y \in \mathbb{R}^d$
- $\pi_{X|Y = y^*}$ is the filtering distribution for a realization $y^*$ of the data

Goal: sample the posterior given only $M$ prior samples $x_1, \ldots, x_M$
Inference as a transportation of measures

Seek a map $T$ that pushes forward prior to posterior

$$(x_1, \ldots, x_M) \sim \pi_X \implies (T(x_1), \ldots, T(x_M)) \sim \pi_{X|Y=y^*}$$

The map induces a coupling between prior and posterior measures

How to construct a “good” coupling from very few prior samples?
Consider the joint distribution of state and observations

\[ \pi_{Y,X} \]

Construct a map \( T \) from the joint distribution \( \pi_{Y,X} \) to the posterior

\( T \) can be computed via convex optimization given samples from \( \pi_{Y,X} \)

Sample \( \pi_{Y,X} \) using the forecast ensemble and the likelihood

\[ (y_i, x_i) \quad y_i \sim \pi_{Y|X=x_i} \]

Intuition: a generalization of the “perturbed observation” EnKFs
Couple the joint distribution with a standard normal

\[ \pi_{Y, X} \xrightarrow{T} \mathbb{R}^{d+n} \rightarrow \mathbb{R}^n \]

\[ \pi_{X|Y = y^*} \]

We can find \( T \) by computing a Knothe–Rosenblatt (KR) rearrangement \( S \) between \( \pi_{Y, X} \) and \( \mathcal{N}(0, I_{d+n}) \)

\[ \pi_{Y, X} \xrightarrow{S} \mathbb{R}^{d+n} \rightarrow \mathbb{R}^{d+n} \]

\[ \mathcal{N}(0, I_{d+n}) \]

- We will show how to derive \( T \) from \( S \)
**Definition:** for any pair of absolutely continuous densities \( \pi, \eta \) on \( \mathbb{R}^m \), there exists a unique triangular and monotone map \( S : \mathbb{R}^m \to \mathbb{R}^m \) such that

\[
S \llcorner \pi = \eta
\]

**Triangular function (nonlinear generalization of a triangular matrix):**

\[
S(x_1, \ldots, x_m) = \begin{bmatrix}
S^1(x_1) \\
S^2(x_1, x_2) \\
\vdots \\
S^m(x_1, x_2, \ldots, x_m)
\end{bmatrix}
\]

**Existence stems from general factorization properties of a density,**

\[
\pi = \pi_{x_1} \pi_{x_2|x_1} \cdots \pi_{x_m|x_1,\ldots,x_{m-1}}
\]
Triangular maps enable conditional simulation

\[ S(x_1, \ldots, x_m) = \begin{bmatrix}
    S^1(x_1) \\
    S^2(x_1, x_2) \\
    \vdots \\
    S^m(x_1, x_2, \ldots, x_m)
\end{bmatrix} \]

- Each component \( S^k \) links marginal conditionals of \( \pi \) and \( \eta \)
- For instance, if \( \eta = \mathcal{N}(0, I) \), then for all \( x_1, \ldots, x_{k-1} \in \mathbb{R}^{k-1} \)

  \[ \xi \mapsto S^k(x_1, \ldots, x_{k-1}, \xi) \] pushes \( \pi_{x_k|x_{1:k-1}}(\xi|x_{1:k-1}) \) to \( \mathcal{N}(0, 1) \)

- Simulate the conditional \( \pi_{x_k|x_{1:k-1}} \) by inverting a 1-D map
  \[ \xi \mapsto S^k(x_{1:k-1}, \xi) \] at Gaussian samples (\emph{need triangular structure})
We are interested in the KR map $S$ that pushes $\pi_{Y,X}$ to $\mathcal{N}(0, I_{d+n})$.

The KR map immediately has a block structure

$$S(y, x) = \begin{bmatrix} S^Y(y) \\ S^X(y, x) \end{bmatrix},$$

which suggests two properties:

- $S^X$ pushes $\pi_{Y,X}$ to $\mathcal{N}(0, I_n)$

- $\xi \mapsto S^X(y^*, \xi)$ pushes $\pi_{X|Y=y^*}$ to $\mathcal{N}(0, I_n)$

The analysis map that pushes $\pi_{Y,X}$ to $\pi_{X|Y=y^*}$ is then given by

$$T(y, x) = S^X(y^*, \cdot)^{-1} \circ S^X(y, x)$$
A novel filtering algorithm with maps

\[ \pi_{Y|X=x_i} \quad \pi_{Y,X} \quad T(y,x) \]

\[ \pi_{X} \quad \pi_{X|Y=y^*} \]

**Transport map ensemble filter**

1. Compute forecast ensemble \( x_1, \ldots, x_M \)
2. Generate samples \((y_i, x_i)\) from \( \pi_{Y,X} \) with \( y_i \sim \pi_{Y|X=x_i} \)
3. Build an estimator \( \hat{T} \) of \( T \)
4. Compute analysis ensemble as \( x^a_i = \hat{T}(y_i, x_i) \) for \( i = 1, \ldots, M \)
Recall the form of $S$:

$$S(y, x) = \begin{bmatrix} S_Y(y) \\ S_X(y, x) \end{bmatrix}, \quad S_{\#} \pi_{Y,X} = \mathcal{N}(0, I_d + n).$$

We propose the following estimator $\hat{T}$ of $T$:

$$\hat{T}(y, x) = \hat{S}^X(y^*, \cdot)^{-1} \circ \hat{S}^X(y, x),$$

where $\hat{S}$ is a maximum likelihood estimator of $S$. 
Given samples $\mathbf{x}_1, \ldots, \mathbf{x}_M$ from a distribution $\pi$ on $\mathbb{R}^{d+n}$, estimate the KR rearrangement $S$ that pushes forward $\pi$ to $\mathcal{N}(0, \mathbf{I}_{d+n})$.

- Constrained MLE for $S$

$$
\hat{S} \in \arg \max_{S \in S^h_{\Delta}} \frac{1}{M} \sum_{i=1}^{M} \log S_{#}^{-1} \eta(\mathbf{x}_i), \quad \eta = \mathcal{N}(0, \mathbf{I}_n),
$$

where $S^h_{\Delta}$ is an approximation space for the rearrangement.

- Each component $\hat{S}^k$ of $\hat{S}$ can be computed separately, via smooth convex optimization

$$
\hat{S}^k \in \arg \min_{S^k \in S^h_{\Delta,k}} \frac{1}{M} \sum_{i=1}^{M} \left( \frac{1}{2} S^k(\mathbf{x}_i)^2 - \log \partial_k S^k(\mathbf{x}_i) \right)
$$
Map parameterizations

\[ \hat{S}^k \in \arg \min_{S^k \in S^h_{\Delta,k}} \frac{1}{M} \sum_{i=1}^{M} \left( \frac{1}{2} S^k(x_i)^2 - \log \partial_k S^k(x_i) \right) \]

- In general, convex optimization
- Optimization is not needed for nonlinear separable parameterizations of the form \( \hat{S}^k(x_{1:k}) = \alpha x_k + g(x_{1:k-1}) \) (just linear regression)
- **Connection to EnKF:** a linear parameterization of \( \hat{S}^k \) yields a particular form of EnKF with “perturbed observations”
- Choice of approximation space allows control of the bias and variance of \( \hat{S} \)
- Richer parameterizations yield less bias, but potentially higher variance

**Strategy:** depart gradually from the linear ansatz by introducing local nonlinearities + regularization
**Example: Lorenz-63**

**Simple example:** three-dimensional Lorenz-63 system

\[
\begin{align*}
\frac{dX_1}{dt} &= \sigma(X_2 - X_1), \\
\frac{dX_2}{dt} &= X_1(\rho - X_3) - X_2 \\
\frac{dX_3}{dt} &= X_1X_2 - \beta X_3
\end{align*}
\]

- Chaotic setting: \( \rho = 28, \sigma = 10, \beta = 8/3 \)
- Fully observed, with additive Gaussian observation noise \( \mathcal{E}_j \sim \mathcal{N}(0, 2^2) \)
- Assimilation interval \( \Delta t = 0.1 \)
- Results computed over 2000 assimilation cycles, following spin-up

**Map parameterizations:** \( S^k(x_{1:k}) = \sum_{i \leq k} \psi_i(x_i), \) with \( \psi_i = \) linear + \{RBFs or sigmoids\}
Example: Lorenz-63

Mean “tracking” error vs. ensemble size and choice of map

![Graph showing mean RMSE versus ensemble size for different methods.]

- EnKF
- Linear
- Linear + 1 RBF
- Linear + 2 RBF
- Particle Filter

Ensemble size $M$
Example: Lorenz-63

How do $M \to \infty$ “plateaus” depend on assimilation interval?
What about comparison to the *true Bayesian solution*?
Example: Lorenz-63

What about comparison to the *true Bayesian solution*?
Regularize the estimator $\hat{S}$ of $S$ by imposing sparsity, e.g.,

$$
\hat{S}(x_1, \ldots, x_4) = \begin{bmatrix}
\hat{S}^1(x_1) \\
\hat{S}^2(x_1, x_2) \\
\hat{S}^3(x_2, x_3) \\
\hat{S}^4(x_3, x_4)
\end{bmatrix}
$$

The sparsity of the $k$th component of $S$ depends on the sparsity of the marginal conditional function $\pi_{X_k|X_{1:k-1}}(x_k|x_{1:k-1})$.

**Localization heuristic:** let each $\hat{S}^k$ depend on variables $(x_j)_{j<k}$ that are within a distance $\ell$ from $x_k$ in state space. Estimate optimal $\ell$ offline.

Explicit link between sparsity of $S$ and conditional independence in non-Gaussian graphical models.
Lorenz-96 in chaotic regime (40-dimensional state)

- A **hard** test-case configuration [Bengtsson et al. 2003]:

\[
\frac{dX_j}{dt} = (X_{j+1} - X_{j-2})X_{j-1} - X_j + F, \quad j = 1, \ldots, 40
\]
\[
Y_j = X_j + \mathcal{E}_j, \quad j = 1, 3, 5 \ldots, 39
\]

- \( F = 8 \) (chaotic) and \( \mathcal{E}_j \sim \mathcal{N}(0, 0.5) \) (**small noise for PF**)
- Time between observations: \( \Delta_{\text{obs}} = 0.4 \) (**large**)
- Results computed over 2000 assimilation cycles, following spin-up
Lorenz-96: “hard” case

![Graph showing the average RMSE for different ensemble sizes](image-url)

- **EnKF**
- **Linear**
- **Linear + 1 RBF**
- **Linear + 2 RBF**
- **Var(ε_t)^1/2**
Lorenz-96: “hard” case

- The nonlinear filter is \( \approx 25\% \) more accurate in RMSE than EnKF
Lorenz-96: “hard” case

![Graph showing the relationship between average coverage probability and ensemble size. The x-axis represents the ensemble size (M) ranging from 60 to 600, and the y-axis represents the average coverage probability ranging from 0.86 to 0.98. The graph compares different methods: EnKF, Linear, Linear + 1 RBF, and Linear + 2 RBF.]
Lorenz-96: non-Gaussian noise

- A heavy-tailed noise configuration:

\[
\frac{dX_j}{dt} = (X_{j+1} - X_{j-2})X_{j-1} - X_j + F, \quad j = 1, \ldots, 40
\]

\[
Y_j = X_j + E_j, \quad j = 1, 5, 9, 13, \ldots, 37
\]

- \( F = 8 \) (chaotic) and \( E_j \sim \text{Laplace}(\lambda = 1) \)
- Time between observations: \( \Delta_{\text{obs}} = 0.1 \)
- Results computed over 2000 assimilation cycles, following spin-up
Lorenz-96: non-Gaussian noise
Lorenz-96: non-Gaussian noise

![Graph showing the average coverage probability for different ensemble sizes and methods. The graph compares EnKF, Linear, Linear + 1 RBF, and Linear + 2 RBF methods over ensemble sizes ranging from 50 to 600. The average coverage probability is depicted on the y-axis, and the ensemble size on the x-axis.]
Observations were assimilated one at a time

Impose sparsity of the map with a 5-way interaction model (*above*)

Separable and nonlinear parameterization of each component

\[
\hat{S}^k(x_{j_1}, \ldots, x_{j_p}, x_k) = \psi(x_{j_1}) + \ldots + \psi(x_{j_p}) + \tilde{\psi}(x_k),
\]

where \( \psi(x) = a_0 + a_1 \cdot x + \sum_{i>1} a_i \exp\left(-\frac{(x - c_i)^2}{\sigma}\right) \).

Much more general parameterizations are of course possible
Lorenz-96: tracking performance of the filter

Simple and localized nonlinearities have significant impact!
Nonlinear generalization of the EnKF: move the ensemble members via local nonlinear transport maps, *no weights or degeneracy*

Learn non-Gaussian features via nonlinear continuous transport and *convex optimization*

Choice of map basis and *sparsity* provide regularization (e.g., *localization*)
Remarks and questions

- Nonlinear generalization of the EnKF: move the ensemble members via local nonlinear transport maps, *no weights or degeneracy*

- Learn non-Gaussian features via nonlinear continuous transport and *convex optimization*

- Choice of map basis and sparsity provide regularization (e.g., localization)

- In principle, filter is consistent as $S^h_{\triangle}$ is enriched and $M \to \infty$, but a careful *error analysis* is needed!

- What is a good or even optimal choice of $S^h_{\triangle}$ for any fixed ensemble size $M$?

- Can regularization penalties (e.g., $\ell_1$) help identify sparse structure, and/or learn sparse maps from few samples?
Regularized estimation of $S$

For simplicity, consider map components $S^k(x) = \sum_j \beta_j \psi_j(x_{1:k-1}) + \alpha_k x_k$

$$\hat{S}^k \in \arg \min_{S^k \in S^h_{\Delta,k}} \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1}{2} S^k(x_i)^2 - \log \partial_k S^k(x_i) \right) + \lambda_N \| \beta \|_1$$

Assume sub-Gaussian $\pi$ and basis functions $\psi_j(x)$

Theorem [BZM]

For polynomial maps of degree $m$ with sparsity $s$, with high probability

$$\mathbb{E}_\pi \left[ D_{KL}(\pi(x_k|x_{1:k-1}) \parallel \hat{S}^k\|_\eta) \right] \lesssim \sqrt{\frac{s^2 m \log k}{N}}$$

Takeaways

- Accurate estimation is feasible in high dimensions with $N \ll k$
- From factorization property of density, error in conditionals ensures

$$D_{KL}(\pi \parallel \hat{S}^k\|_\eta) \lesssim d\sqrt{\frac{s^2 m \log d}{N}}$$
Linear–Gaussian problem

- Prior: \( \mathbf{X} \sim \mathcal{N}(\mu, \Sigma_{pr}) \) with exponential covariance
- Likelihood: Local observations \( \mathbf{Y} = \mathbf{H}\mathbf{X} + \mathcal{E} \) with \( \mathcal{E} \sim \mathcal{N}(0, I) \)

Takeaway

- Learning sparse prior-to-posterior map matches oracle scaling
Compare two approaches for posterior sampling

\[ \pi(y, x) \xrightarrow{S_x(y^*, \cdot)^{-1} \circ S_x(y, x)} T(y, x) \xrightarrow{\pi(x|y^*)} \]

\[ S_x(y, x) \xrightarrow{\eta(z_2)} S_x(y^*, \cdot)^{-1} \]
Compare two approaches for posterior sampling

\[
\mathbf{X}|\mathbf{y}^* \sim \hat{\mathcal{S}}_{\mathbf{X}}(\mathbf{y}^*, \cdot)^{-1}\eta
\]

Propagating forecast through composed maps has lower error

This is in fact a general approach to likelihood-free inference/ABC
There is also a “square root” version of the nonlinear ensemble filter
Continuous-time formulations?
Nonlinear ensemble smoothers
Open questions about estimation and regularization of continuous transport maps:
  How to choose and adapt approximation space/basis to the forecast ensemble?
  Properties of the estimator $\hat{T}$, e.g., consistency, sample size requirements and scaling
  Other forms of low-dimensional structure and regularization
Applications to inference in “likelihood-free” settings
References

- **General python code at** [http://transportmaps.mit.edu](http://transportmaps.mit.edu)