Data-driven Correction of Model and Representation Error in Data Assimilation

Tyrus Berry
Dept. of Mathematical Sciences, GMU

SIAM UQ
April 18, 2018

Joint work with John Harlim and Dimitris Giannakis

Roadmap: Correcting model error

- What is manifold learning? $\Rightarrow$ Custom Fourier Basis
- Nonparametric methods (no model)
  - Diffusion Forecast
- Semiparametric methods (model error)
- Correcting observation model error

Manifold Learning

- Geometric prior: Data lie on smooth manifold $\mathcal{M} \subset \mathbb{R}^m$
- **Manifold learning $\Leftrightarrow$ Estimating Laplace-Beltrami**
- Eigenfunctions $\Delta \varphi_i = \lambda_i \varphi_i$ orthonormal basis for $L^2(\mathcal{M})$
- Smoothest functions: φ_i minimizes the functional

$$\lambda_i = \min_{\substack{f \perp \varphi_k \\ k=1,\dots,i-1}} \left\{ \frac{\int_{\mathcal{M}} ||\nabla f||^2 \, dV}{\int_{\mathcal{M}} |f|^2 \, dV} \right\}$$

- Eigenfunctions of Δ are custom Fourier basis
  - Smoothest orthonormal basis for $L^2(\mathcal{M})$
  - Can be used to define wavelets
  - Define the Hilbert/Sobolev spaces on $\mathcal{M}$

SO HOW DO WE FIND THE LAPLACIAN FROM DATA?

- Data set $\Rightarrow$ *weighted graph*
- Edge Weights defined by a kernel function

$$K_\delta(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{4\delta^2}}$$

- Bandwidth δ determines localization
- 'Adjacency' matrix: $\mathbf{K}_{ij} = K(x_i, x_j)$
- 'Degree' matrix: $\mathbf{D}_{ii} = \sum_j \mathbf{K}_{ij}$
- Normalized graph Laplacian: $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{K}$

POINTWISE CONVERGENCE

Theorem: (Belkin & Niyogi, 2005, Singer, 2006)
For $\{x_i\}_{i=1}^N \subset \mathcal{M} \subset \mathbb{R}^m$ uniformly sampled on a compact manifold and for $\vec{f}_i = f(x_i)$ where $f \in C^3(\mathcal{M})$

$$\frac{1}{\delta^2}\left(\mathbf{L}\vec{f}\right)_i = \Delta f(x_i) + \mathcal{O}\left(\delta^2, \frac{1}{N^{1/2}\delta^{1+d/2}}\right)$$

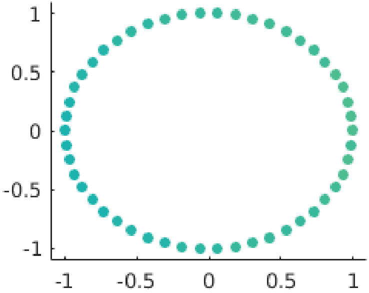
$\delta =$ bandwidth
$N =$ number of points

RESTRICTIONS THAT HAVE BEEN OVERCOME TO DEAL WITH REAL DATA:

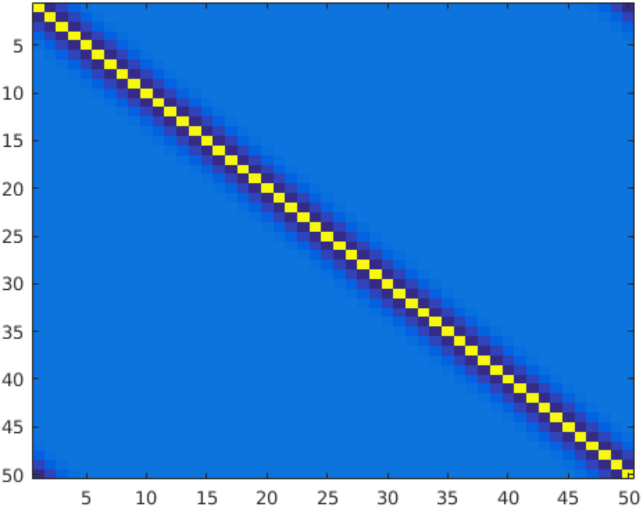
- Arbitrary sampling (Coifman & Lafon, 'Diffusion maps', ACHA 2006)
- Non-compact manifolds (Berry & Harlim, ACHA 2015)
- Other kernel functions (Thesis 2013; Berry & Sauer, ACHA 2015)
- Boundary (Coifman & Lafon, ACHA 2006; Berry & Sauer, J. Comp. Stat. 2016)
- Spectral convergence (Luxburg et al., Ann. Stat. 2008, Berry & Sauer, submitted)

EXAMPLE: 50 DATA POINTS ON S^1

- True Laplacian: $\Delta = \frac{d^2}{d\theta^2}$
- True Eigenfunctions: $\{\sin(k\theta), \cos(k\theta)\}$

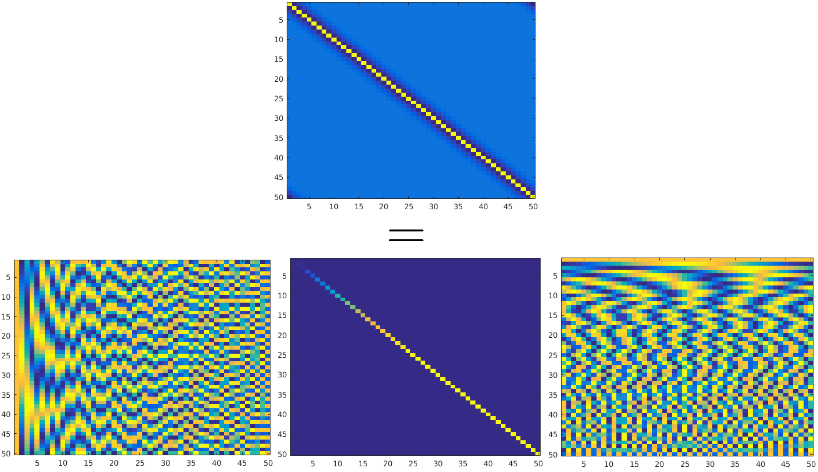


Example: L matrix for S^1



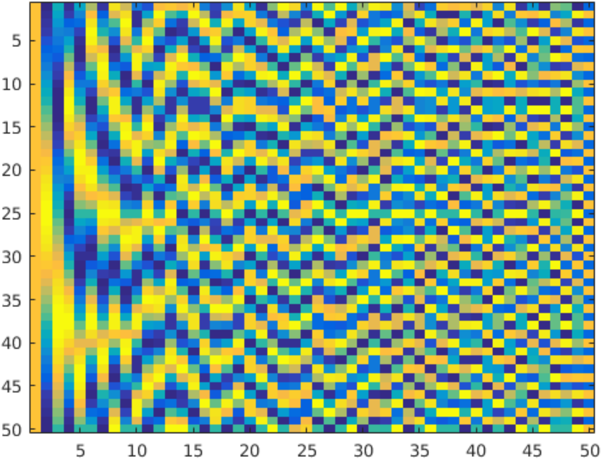
Example S^1: Eigenvector Decomposition

$$L = U \Lambda U^\top$$

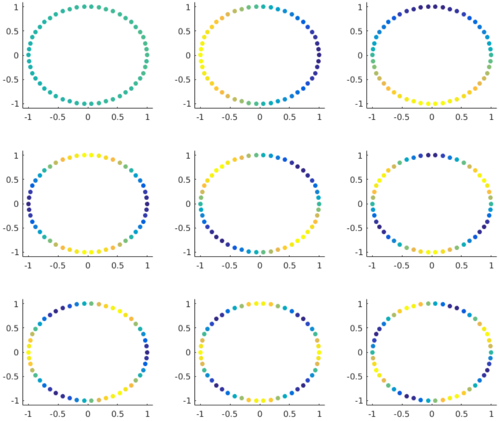


U Λ $U^\top$

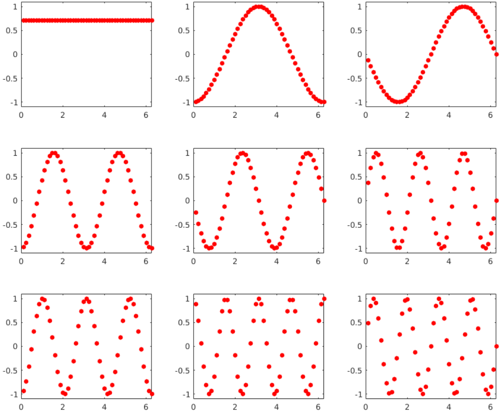
Example S^1: Matrix of Eigenvectors, U



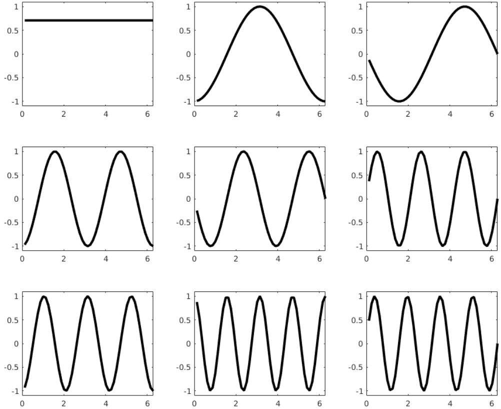
EXAMPLE S^1: EIGENVECTORS ON DATA



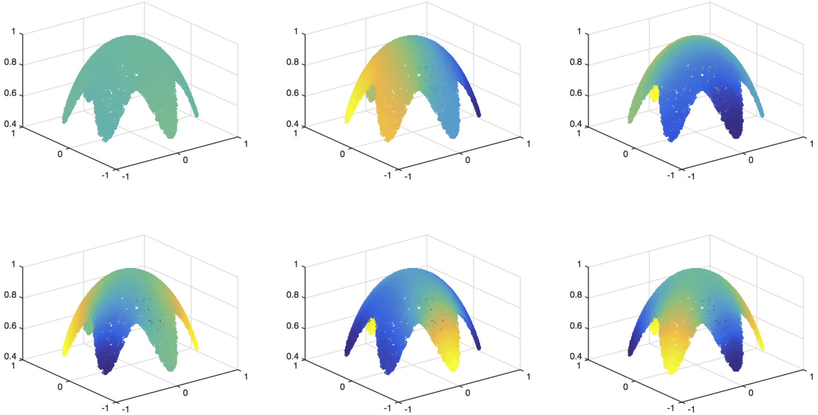
Example S^1: Eigenvectors vs. θ



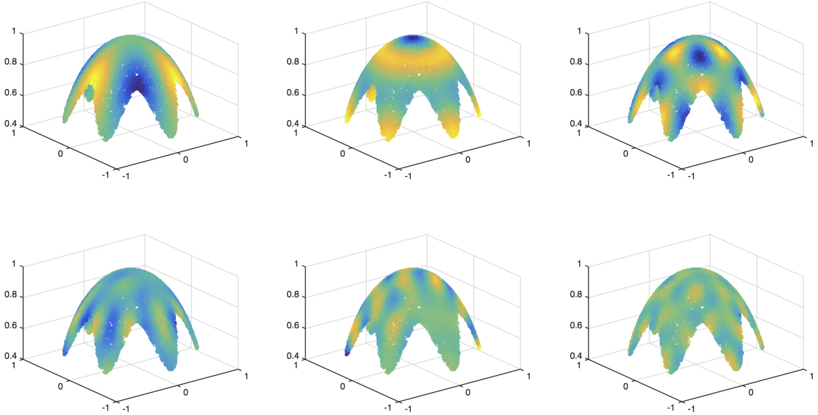
EXAMPLE S^1: CONNECTING THE DOTS



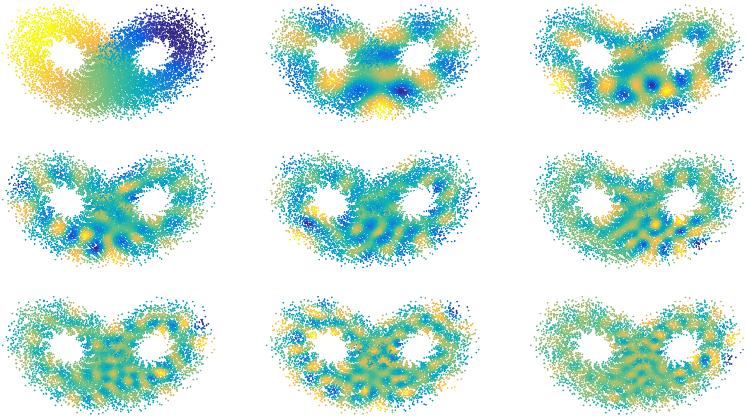
Harmonic Analysis on Manifolds/Data Sets



Harmonic Analysis on Manifolds/Data Sets



HARMONIC ANALYSIS ON MANIFOLDS/DATA SETS



DIFFUSION FORECAST

- Autonomous SDE: $dx = a(x)\,dt + b(x)\,dW_t$
- Density solves Fokker-Planck PDE: $\frac{\partial}{\partial t}p = \mathcal{L}^* p$
- Project onto the custom Fourier basis $\{\varphi_j\}$
- Forecast operator is linear $\Rightarrow$ Matrix $A_{lj} = \langle \varphi_j, e^{t\mathcal{L}} \varphi_l \rangle$

$$\begin{tikzcd}
p(x,t) \arrow[r, dashed, "\text{Diffusion Forecast}"] \arrow[d, "\langle p, \varphi_j \rangle"] & p(x,t+\tau) = e^{\tau\mathcal{L}^*} p(x,t) \\
\vec{c}(t) \arrow[r, "A_{lj} \equiv \mathbb{E}[\langle \varphi_j, S\varphi_l \rangle_{p_{\mathrm{eq}}}]"] & \vec{c}(t+\tau) = A\vec{c}(t). \arrow[u, "\sum_j c_j \varphi_j q"']
\end{tikzcd}$$

Diffusion Forecast Lorenz-63 Example

(Loading Video...)

NONPARAMETRIC FORECAST ON A TORUS

- Stochastic dynamics on a torus $(\theta, \phi) \in [0, 2\pi)^2$

$$d(\theta,\phi)^\top = a(\theta,\phi)\,dt + b(\theta,\phi)\,dW_t$$

- Drift and diffusion coefficients,

$$a(\theta,\phi) = \begin{pmatrix} \frac{1}{2} + \frac{1}{8}\cos(\theta)\cos(2\phi) + \frac{1}{2}\cos(\theta+\pi/2) \\ 10 + \frac{1}{2}\cos(\theta+\phi/2) + \cos(\theta+\pi/2)) \end{pmatrix},$$

$$b(\theta,\phi) = \begin{pmatrix} \frac{1}{4} + \frac{1}{4}\sin(\theta) & \frac{1}{4}\cos(\theta+\phi) \\ \frac{1}{4}\cos(\theta+\phi) & \frac{1}{40} + \frac{1}{40}\sin(\phi)\cos(\theta) \end{pmatrix}.$$

DIFFUSION FORECAST TORUS EXAMPLE

(Loading Video...)

Problem: Curse of dimensionality

- Learning the basis → Data exponential in manifold dim
- Monte-Carlo type estimates $\mathcal{O}(N^{-1/2})$:
  - Coefficients:

$$c_l(t) = \langle p(x,t), \varphi_l \rangle \approx \frac{1}{N}\sum_{i=1}^{N} \varphi_l(x_i) p(x_i,t)/p_{\mathrm{eq}}(x_i)$$

  - Markov Matrix:

$$A_{lj} = \left\langle \varphi_j, e^{\tau\mathcal{L}}\varphi_l \right\rangle_{p_{\mathrm{eq}}} \approx \frac{1}{N}\sum_{i=1}^{N} \varphi_j(x_i)\varphi_l(x_{i+1})$$

- Maybe we shouldn't throw out the model...
- Use diffusion forecast to fix model error!

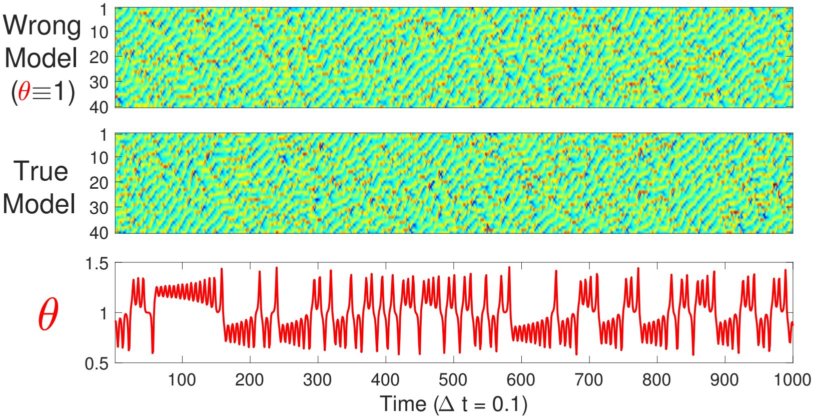
Semiparametric Forecast Model

- Partially known model $\dot{x} = f(x, \theta)$
- Unknown: $d\theta = a(\theta)\, dt + b(\theta)\, dW_t$
- Apply the Diffusion Forecast to $p(\theta, t)$
- Sample $\theta^k(t) \sim p(\theta, t)$ and pair with ensemble $x^k(t)$

$$\begin{tikzcd}
(x^k(t), \theta^k(t)) \arrow[r, "\dot{x}=f(x,\theta)"] & (x^k(t+\tau), \theta^k(t+\tau)) \\
p(\theta, t) \arrow[u, "\theta^k(t)"'] \arrow[r, dashed, "\text{Diffusion Forecast}"] & p(\theta, t+\tau) \arrow[u, "\theta^k(t+\tau)"']
\end{tikzcd}$$

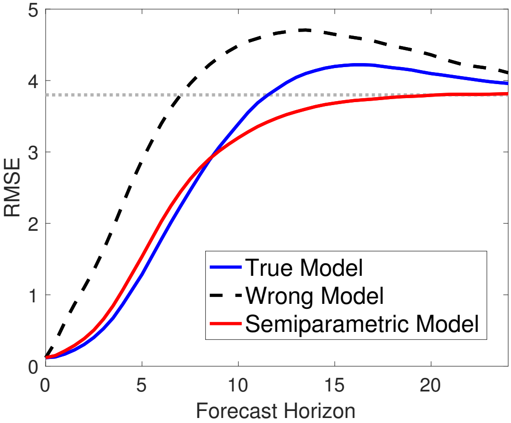
Example: 40-dimensional Lorenz-96 system

$$\dot{x}_i = \theta x_{i-1}x_{i+1} - x_{i-1}x_{i-2} - x_i + 8$$



Example: 40-dimensional Lorenz-96 system

$$\dot{x}_i = {\color{red}\theta} x_{i-1}x_{i+1} - x_{i-1}x_{i-2} - x_i + 8$$



SEMIPARAMETRIC FILTER: PUT IT ALL TOGETHER...

$$\begin{tikzcd}
\begin{pmatrix} x^{k,a}(t-\tau) \\ \theta^{k,a}(t-\tau) \end{pmatrix} \arrow[r, "\dot{x}=f(x,\theta)"] \arrow[d, "\theta^a"] & \begin{pmatrix} x^{k,f}(t) \\ \theta^{k,f}(t) \end{pmatrix} \arrow[r, "\text{EnKF } y^o(t)"] & \begin{pmatrix} x^{k,a}(t) \\ \theta^{k,a}(t) \end{pmatrix} \arrow[d, "p(\theta^a(t)\,|\,\theta(t))"'] \\
p^a(\theta,t-\tau) \arrow[r, dashed, "\text{Diffusion Forecast}"] \arrow[d, "\langle p^a,\varphi_j\rangle"] & p^f(\theta,t) \arrow[u, "\theta^{k,f}(t)"'] \arrow[r, "p^f(\theta)p(y\,|\,\theta)"] & p^a(\theta,t) \arrow[d, "\langle p^a,\varphi_j\rangle"'] \\
\vec{c}^{\,a}(t-\tau) \arrow[r, "A_{lj}c^{\,a}(t-\tau)"] & \vec{c}^{\,f}(t) \arrow[u, "\sum_j c_j^f\varphi_j p_{\mathrm{eq}}"'] \arrow[r, dashed, "\text{Bayesian Update}"] & \vec{c}^{\,a}(t)
\end{tikzcd}$$

MODEL ERROR OVERVIEW

- Consider the standard filtering problem,

$$\begin{aligned} x_i &= f(x_{i-1}, \theta) + \omega_{i-1} \\ y_i &= h(x_i) + \eta_i \end{aligned}$$

- So far we have focused on the dynamics, *f*
  - Diffusion Forecast ⇒ Learn *f* from data
  - Diffusion Forecast for θ to correct model error
- We have assumed that *h* is fully known...

BIAS IN OBSERVATION MODELS

- Consider the standard filtering problem,

$$x_i = f(x_{i-1}) + \omega_{i-1}$$
$$y_i = h(x_i) + \eta_i$$

- We assume the true observation function $h(x)$ is unknown
- An approximate model is available $\tilde{h}(x)$ so that

$$y_i = h(x_i) + \eta_i = \tilde{h}(x_i) + b_i + \eta_i$$

- Where $b_i \equiv h(x_i) - \tilde{h}(x_i)$ is called the bias

EXAMPLE 1: LORENZ-96

- Consider the standard 40-dimensional Lorenz-96,

$$\dot{x}_j = x_{j-1}(x_{j+1} - x_{j-2}) - x_j + 8$$

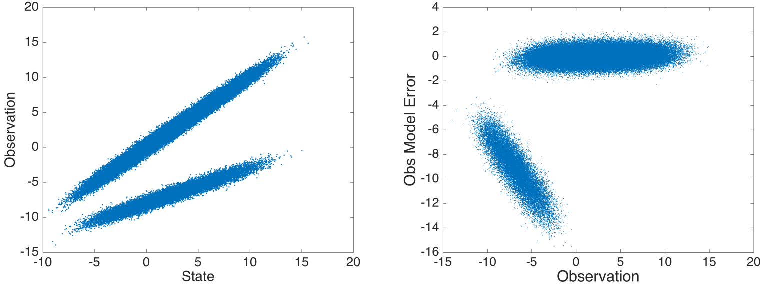
- We observe 20 of the 40 variables
- We draw $\xi_i \sim \mathcal{U}(0,1)$ and let the observations be,

$$h(x_k) = \begin{cases} x_k & \xi_i > 0.8 \\ \beta_k x_k - 8 & \text{else} \end{cases}$$
$$\beta_k \sim \mathcal{N}(0.5, 1/50).$$

- *h* is applied to 7 randomly chosen variables
- Remaining 13 are directly observed

Example 1: Lorenz-96

- The result is a bimodal distribution, “cloudy/clear”
- Obs Model Error = True Obs - $\tilde{h}$(True State)



CORRECTING THE BIAS

- Our goal is to find $p(b_i \,|\, y_i)$
- We can then correct our observation function

$$\hat{h}(x_i^f) \equiv \tilde{h}(x_i^f) + \hat{b}_i$$

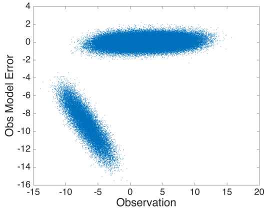
- Where $\hat{b}_i = \mathbb{E}_{p(b_i \,|\, y_i)}[b_i]$
- Since $\hat{b}_i$ random:
  - Inflate the obs noise covariance
  - Use $\hat{R}_{b_i} = \mathbb{E}_{p(b_i \,|\, y_i)}[(b_i - \hat{b}_i)(b_i - \hat{b}_i)^\top]$

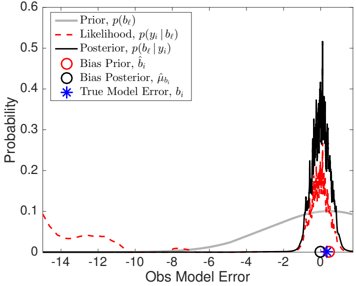
Correcting the bias

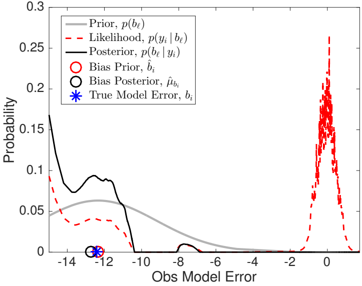
- If we can estimate $p(b_i \mid y_i)$ we can fix the obs
- From the forecast step we have a prior $p(b_i)$
- Can use Bayes' $p(b_i \mid y_i) = p(b_i)p(y_i \mid b_i)$
- Need the likelihood $p(y_i \mid b_i)$
- Use kernel estimation of conditional distributions

Correcting the bias

- Below plots have $y_i \approx -4$
- Left is clear, right is cloudy
- Notice bimodal likelihood







Learning the conditional distribution

- Given training data (y_i, b_i) our goal is to learn $p(y_i \mid b_i)$
- For a kernel $K(\alpha, \beta) = e^{-\frac{||\alpha-\beta||^2}{\delta^2}}$ we define Hilbert spaces

$$\mathcal{H}_y = \left\{ \sum_{i=1}^{N} a_i K(y_i, \cdot) \,:\, \vec{a} \in \mathbb{R}^N \right\}, \mathcal{H}_b = \left\{ \sum_{i=1}^{N} a_i K(b_i, \cdot) \,:\, \vec{a} \in \mathbb{R}^N \right\}$$

- For example the kernel density estimate (KDE) $\hat{q}$ is in $\mathcal{H}_y$

$$\hat{q}(y) = \frac{1}{m_0 N} \sum_{i=1}^{N} K(y_i, y)$$

- Eigenvectors ϕ_ℓ of $K_{ij} = K(y_i, y_j)$ form an orthonormal basis for $\mathcal{H}_y$. Similarly φ_k are a basis for $\mathcal{H}_b$.

Learning the conditional distribution

- We assume that $p(y \mid b)$ can be approximated in $\mathcal{H}_y \otimes \mathcal{H}_b$
- Let $C^{yb}_{ij} = \langle \phi_i, \varphi_j \rangle$ and $C^{bb}_{ij} = \langle \varphi_i, \varphi_j \rangle$ then define

$$C^{y|b} = C^{yb} \left(C^{bb} + \lambda I \right)^{-1}$$

- We can then define a consistent estimator of $p(y \mid b)$ by

$$\hat{p}(y \mid b) = \sum_{i,j=1}^{N} C^{y|b}_{i,j} \phi_i(y) \varphi_j(b) \hat{q}(y)$$

- We define eigenfunctions with Nystöm extension

$$\varphi_j(b) = \lambda_j^{-1} \sum_{i=1}^{N} \varphi_j(b_i) K(b_i, b)$$

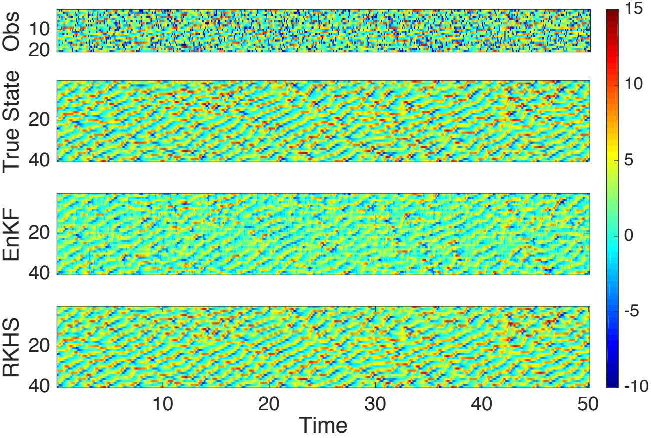
OVERVIEW

- **Learning Phase:** Given training data set (x_i, y_i)
  - Compute the biases $b_i = y_i - \tilde{h}(x_i)$
  - Learn the conditional distribution $p(y \,|\, b)$
- **Filtering:** Forecast $x_i^f \Rightarrow$ innovation $\epsilon_i = y_i - \tilde{h}(x_i^f)$
- Use prior $p(b) = \mathcal{N}(\epsilon_i, P_i^y)$
- Combine with conditional to find $p(b \,|\, y_i) = p(b)p(y_i \,|\, b)$
- Estimate conditional mean $\hat{b}_i$ and covariance $\hat{R}_{b_i}$
- Adjust innovation $\hat{\epsilon}_i = \epsilon_i + \hat{b}_i$ and $R_i = R^o + \hat{R}_{b_i}$
- Apply Kalman update, continue to the next filter step

OVERVIEW

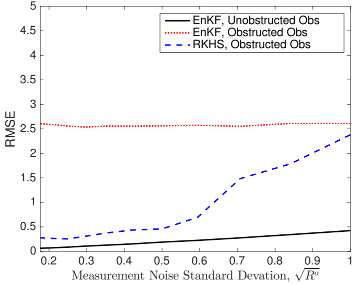
$$\begin{tikzcd}
\begin{array}{c}\text{Prior}\\ p(x_i)\end{array} \arrow[r, "\text{Primary Filter}"] \arrow[d] & \begin{array}{c}\text{Posterior}\\ p(x_i \mid y_i)\end{array} \\
\begin{array}{c}\text{Error Prior}\\ p(b)\end{array} \arrow[r, "\text{Secondary Filter}"] & \begin{array}{c}\text{Error Posterior}\\ p(b \mid y_i)\end{array} \arrow[u] \\
\begin{array}{c}\text{Observation}\\ y_i\end{array} \arrow[u] \arrow[r, "\text{RKHS+Training Data}"] & \begin{array}{c}\text{Likelihood}\\ p(y_i \mid b)\end{array} \arrow[u]
\end{tikzcd}$$

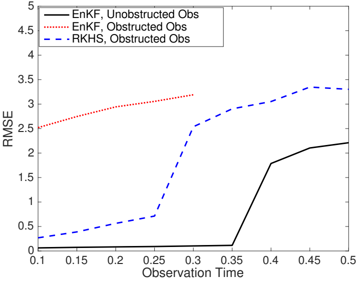
LORENZ-96 RESULTS



Lorenz-96 Results

- Works well with small measurement noise
- Observations need to be precise, but not accurate


EnKF, Unobstructed Obs
EnKF, Obstructed Obs
RKHS, Obstructed Obs
RMSE
Measurement Noise Standard Devation, $\sqrt{R^o}$


EnKF, Unobstructed Obs
EnKF, Obstructed Obs
RKHS, Obstructed Obs
RMSE
Observation Time