Model-free forecasting with applications to multi-sensor arrays

Tyrus Berry

George Mason University

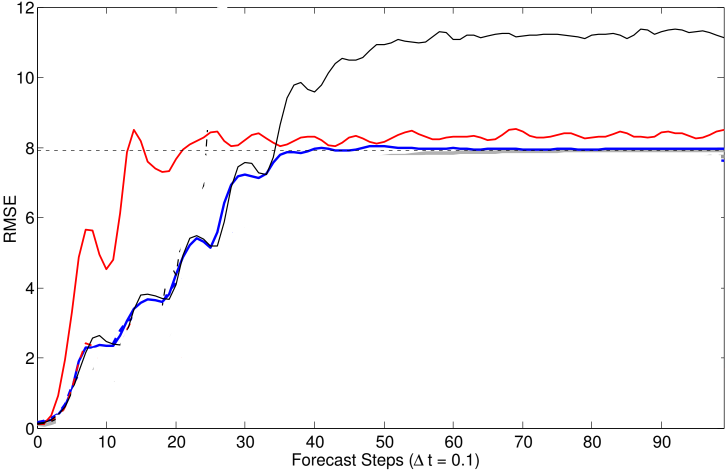
March 16, 2021

Joint work with John Harlim, PSU and Dimitris Giannakis, NYU
Supported by NSF-DMS 1854204 and 1723175

Types of Forecasting: Deterministic

- Deterministic Forecasting, $x_{k+1} = F(x_k)$
- Regression problem: Learn F from data
- Iterative Methods: $x_{k+n} = \tilde{F}^n(x_k)$ where $\tilde{F} \approx F$
- Direct Methods: $x_{k+n} = \tilde{F}_n(x_k)$ where $\tilde{F}_n \approx F^n$

DIRECT VS. **Iterative** VS PROBABILISTIC



Regression Comparison

- Local Linear Regression (x_j near x):

$$F(x) \approx F(x_j) + DF(x_j)(x - x_j)$$

- Kernel Regression (h is bump function):

$$F(x) \approx \sum_j c_j h(||x - x_j||_{A_j})$$

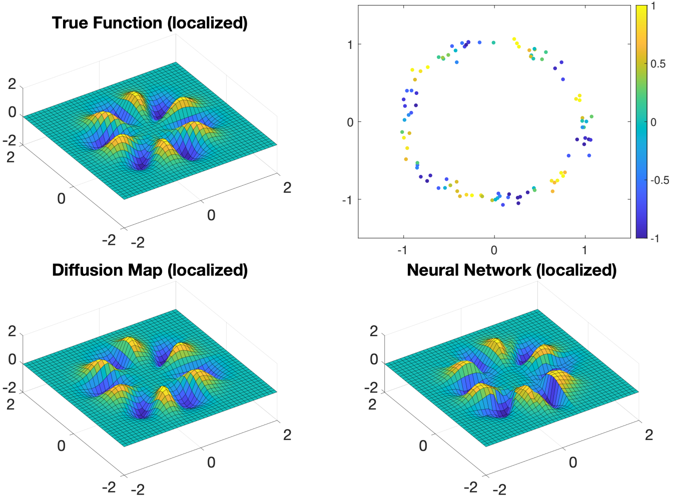
- Neural Network (h is sigmoid):

$$F(x) \approx \sum_j c_j h(a_j^\top (x - \tilde{x}_j))$$

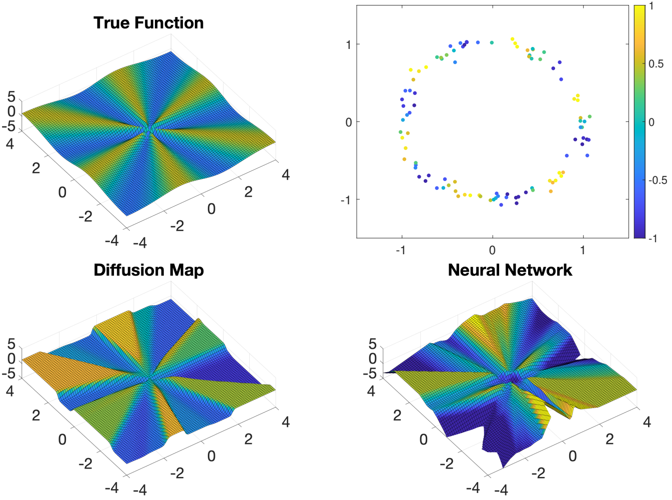
(where we write $b_j = a_j^\top \tilde{x}_j$)

- Deep Network: Composition of Neural Networks
- Reservoir Computer: Fix a_j, b_j, linear regression for c_j

NYSTRÖM VS. DEEP NET, $(r,\theta) \mapsto sin(6\theta)$



NYSTRÖM VS. DEEP NET, $(r, \theta) \mapsto \sin(6\theta)$



TYPES OF FORECASTING: UQ

- Deterministic Forecasting, $x_{k+1} = F(x_k), x_0 \sim p_0$
- Uncertainty Quantification, $p_{k+1} = \mathcal{F}(p_k) = p_k \circ F$
- *Can be* considered a regression problem
- *Option 1:* Learn F, then apply UQ (MC, PC, etc.)
- *Option 2:* Learn $\mathcal{F}$ directly in a basis

$$A_{ij} = \langle \phi_i, \mathcal{F}\phi_j \rangle = \langle \phi_i, \phi_j \circ F \rangle \approx \frac{1}{N}\sum_{k=1}^{N} \phi_i(x_k)\phi_j(x_{k+1})$$

Types of Forecasting: Stochastic

- Stochastic Forecasting, $x_{k+1} = F(x_k, \omega_k)$
- *Not* a regression problem
- Don't just want $\bar{F}(\cdot) = \mathbb{E}_\omega[F(\cdot,\omega)]$
- We want the forward operator

$$p_{k+1} = \mathcal{F}(p_k) = \int p_k \circ F(\cdot,\omega)\, d\pi(\omega)$$

- Note: $\int p_k \circ F(\cdot,\omega)\, d\pi(\omega) \neq p_k \circ \int F(\cdot,\omega)\, d\pi(\omega)$

STOCHASTIC FORECASTING = OPERATOR ESTIMATION

- Represent $\mathcal{F}$ in a basis

$$A_{ij} = \langle \phi_i, \mathcal{F}\phi_j \rangle = \langle \phi_i, \phi_j \circ F \rangle \approx \frac{1}{N}\sum_{k=1}^{N} \phi_i(x_k)\phi_j(x_{k+1})$$

- Error Sources: Bias, variance, and truncation
- Which basis?
  - Respect the measure $\Rightarrow$ Eliminate bias
  - Leverage smoothness $\Rightarrow$ Minimize variance
  - Capture global structure $\Rightarrow$ Minimize truncation

What is Manifold Learning?

- **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

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

- 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 $\mathcal{M}$

CONFORMALLY INVARIANT DIFFUSION MAPS (CIDM)

- Data samples $\{x_i\}_{i=1}^N \subset \mathcal{M} \subset \mathbb{R}^n$ of volume $p_{\mathrm{eq}}\, dV$
- Continuous k-Nearest Neighbors (CkNN) dissimilarity:

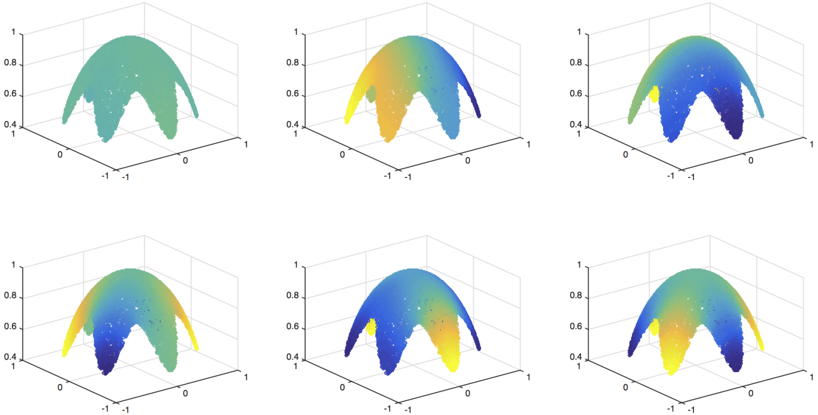
$$d(x_i, x_j) \equiv \frac{||x_i - x_j||}{\sqrt{||x_i - x_{kNN(i)}||\,||x_j - x_{kNN(j)}||}}$$

- Variable bandwidth kernel, $K_{ij} = \exp\left(\frac{-d(x_i,x_j)^2}{\delta^2}\right)$
- Degree matrix $D_{ii} = \sum_j K_{ij}$ (diagonal)
- Graph Laplacian, $L = \frac{D-K}{\delta^{d+2}}$
- **Theorem:** $L\vec{f} = \Delta_{\hat{g}} f + \mathcal{O}\left(\delta^2, N^{-1/2}\delta^{-1-d/2}\right)$, $\hat{g} = p_{\mathrm{eq}}^{2/d} g$
- **Solve:** $(I - D^{-1/2}KD^{-1/2})\vec{v} = \lambda\vec{v}$, set $\vec{\varphi} = D^{-1/2}\vec{v}$

Harmonic Analysis on Manifolds/Data Sets

- Manifolds with boundary, (R. Vaughn)

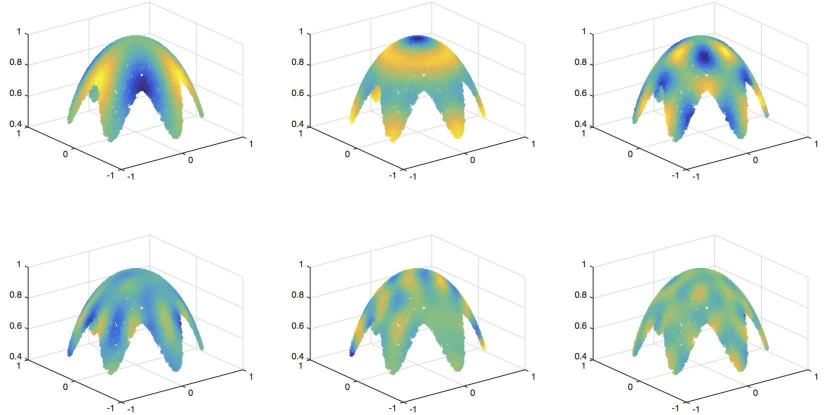
$$\vec{h}^\top L\vec{f} \to \int (\nabla h \cdot \nabla f)\, p_{\mathrm{eq}}\, dV$$



Harmonic Analysis on Manifolds/Data Sets

- Manifolds with boundary, (R. Vaughn)

$$\vec{h}^\top L\vec{f} \to \langle\langle \nabla_{\hat{g}} h, \nabla_{\hat{g}} f \rangle\rangle_{\hat{g}} = \int \hat{g}(\nabla_{\hat{g}} h, \nabla_{\hat{g}} f)\, dV_{\hat{g}}$$



FORECASTING THE FOKKER-PLANK PDE

- Dynamical system: $dx = a(x)\,dt + b(x)\,dW_t$
- Uncertain initial state $x(0)$ with density $p(x,0)$
- Density solves Fokker-Planck PDE, $p_t = \mathcal{L}^* p$ where

$$\mathcal{L}^* p = -\nabla \circ (pa) + \frac{1}{2}\sum_{i,j} \frac{\partial^2}{\partial x_i \partial x_j}\left(p \sum_k b_{ik} b_{jk}\right)$$

- Semigroup solution, $p(x,t) = e^{t\mathcal{L}^*} p(x,0)$

THE SHIFT MAP

- Given data samples $x_i = x(t_i)$ with $\tau = t_{i+1} - t_i$
- Define the *shift map* of a function by $Sf(x_i) = f(x_{i+1})$
- Using the Itô lemma we can show:

$$Sf(x_i) = f(x_{i+1}) = e^{\tau\mathcal{L}} f(x_i) + \int_{t_i}^{t_{i+1}} \nabla f^\top b \, dW_s + \int_{t_i}^{t_{i+1}} Bf \, ds$$

- Notice: $\mathbb{E}[S(f)] = e^{\tau\mathcal{L}} f$
- Need to minimize the stochastic integrand $\nabla f^\top b$

Forecasting with the Shift Map

$$
\begin{array}{ccc}
p(x,t) & \xrightarrow{\text{Diffusion Forecast}} & p(x,t+\tau) \\
\Big\downarrow \langle p, \varphi_j \rangle & & \Big\uparrow \sum_j c_j \varphi_j p_{\text{eq}} \\
\vec{c}(t) & \xrightarrow{A_{lj} \equiv \mathbb{E}[\langle \varphi_j, S\varphi_l \rangle_{p_{\text{eq}}}]} & \vec{c}(t+\tau) = A\vec{c}(t).
\end{array}
$$

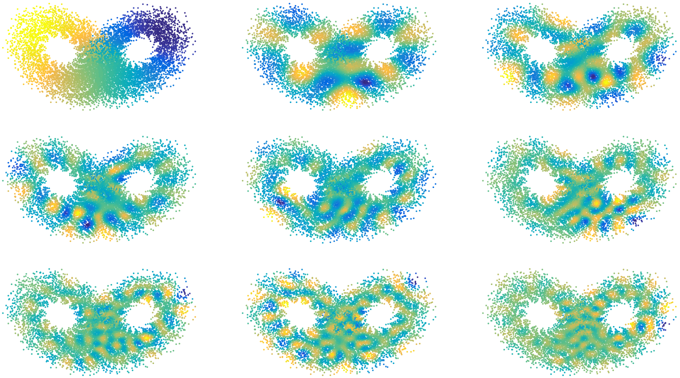
- Estimate A_{lj} with $\hat{A}_{lj} = \frac{1}{N} \sum_{i=1}^{N} \varphi_j(x_i)\varphi_l(x_{i+1})$
- $\mathbb{E}[\hat{A}_{lj}] = A_{lj}$ with error $\mathcal{O}(||\nabla \varphi_l||_{p_{\text{eq}}} \sqrt{\tau/N})$

Choosing a basis

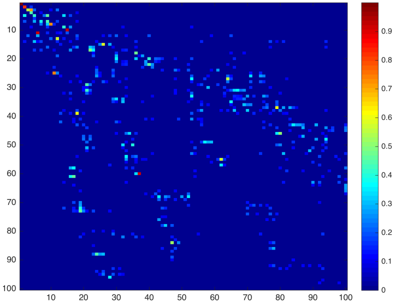
- Need to minimize the error term $\mathcal{O}(||\nabla \varphi_l||_{p_{\rm eq}} \sqrt{\tau/N})$
- The eigenfunctions $\Delta_{\hat{g}} \varphi_j = \lambda_j \varphi_j$ minimize $||\nabla \varphi_j||_{p_{\rm eq}} = \lambda_j$
- Find φ_j with Manifold Learning: CIDM

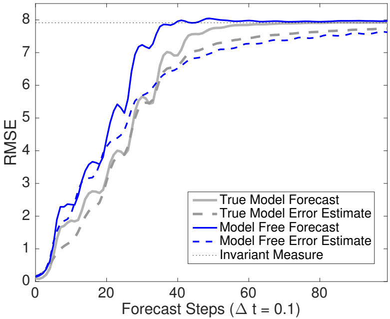
MANIFOLD LEARNING ⇒ CUSTOM 'FOURIER' BASIS

- **Optimal basis:** Minimum variance $A_{lj} \equiv \mathbb{E}[\langle \varphi_j, S\varphi_l \rangle_{p_{\mathrm{eq}}}]$



SHIFT MAP ⇒ MARKOV MATRIX





DIFFUSION FORECAST EXAMPLE

(Loading Video...)

RELATIONSHIP TO CLASSICAL METHODS

- For partial observations, use Takens' reconstruction
- Local linear representations
  - Nearest neighbor interpolation
  - Diffusion forecast extends the map to distributions
- Partition state space $\Rightarrow$ Markov matrix
  - Also uses the shift map, just a different basis
  - Diffusion forecast is optimal basis for estimation

RELATIONSHIP TO RESERVOIR COMPUTERS

- Create a random (recurrent) network $v_k \in \mathbb{R}^N$

$$v_{k+1} = f(Av_k + Bx_k)$$

- Continuously feed in the time series x_k

$$\begin{aligned} v_{k+1} &= f(Af(Av_{k-1} + Bx_{k-1}) + Bx_k) = \cdots \\ &= f(Af(A \cdots f(Av_{k-\tau} + Bx_{k-\tau}) + \cdots) + Bx_k) \\ &= g(x_k, x_{k-1}, ..., x_{k-\tau}) \end{aligned}$$

- Predict: $x_{k+1} = Wv_k = Wg(x_k, ..., x_{k-\tau})$
- Since $\lambda_{\max}(A) < 1$ network forgets distant past
- Effectively a random diffeomorphism of a delay embedding
- Effectively uses a linear combination W of random basis!

PROBLEM: CURSE OF DIMENSIONALITY

- Nonparametric methods $\rightarrow$ Data required grows like a^{dim}

PROJECTIONS OF HIGH DIMENSIONAL DYNAMICS

- Consider the 40-dimensional Lorenz-96 system:

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

- Assume we only observe a projection of this system

$$y = h(x_1, ..., x_{40})$$

- Example: Spatial Fourier mode $y = \hat{x}_\omega = \sum_{k=1}^{40} x_i e^{-k\omega}$
- Evolution of y is not closed, sometimes modeled by SDEs

Attractor Reconstruction

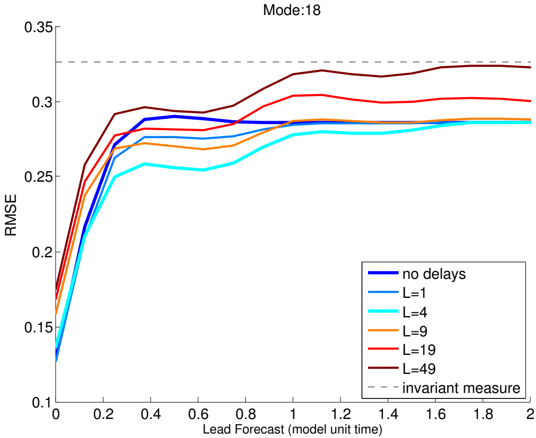
- Evolution of $y = h(x)$ is not closed (missing information)
- Idea: Use delay-embedding to recover the missing info
- Problem 1: Delay embeddings are biased towards stable directions

$$\tilde{y}_t \equiv (y_t, y_{t-\tau}, ..., y_{t-L\tau}) = (h(x_t), h(F_{-\tau}(x_t), ..., h(F_{-L\tau}(x_t))$$

- Problem 2: Curse-of-dimensionality prevents learning the full attractor
- Adding some delays helps, but adding too many hurts

Attractor Reconstruction

- Evolution of $y = h(x)$ is not closed
- Adding some delays helps, but adding too many hurts



NEXT STEPS: MORI-ZWANZIG FORMALISM

- Evolution of $y = h(x)$ is not closed
- Delay-embedding, $\tilde{y}_t$ only yeilds partial reconstruction
- Projections of dynamical systems can be closed as

  Mori-Zwanzig formalism: $$\frac{d}{dt}\tilde{y} = V + K + R$$

- Diffusion Forecast includes: V (Markovian), R (stochastic)
- Missing the memory term: $K = \int_{-\infty}^{t} K(s, \tilde{y}_t, \tilde{y}_s)\tilde{y}_s \, ds$

SYNTHETIC SENSOR

Consider a synthetic sensor given by a generic sigmoid,

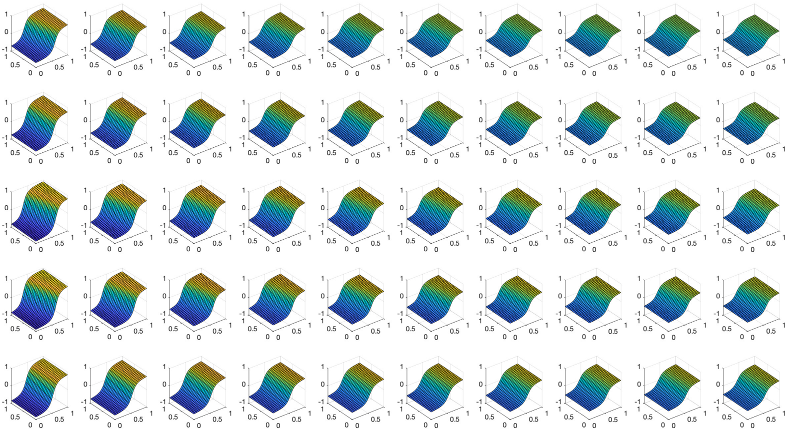
$$h_s(\vec{x},\vec{z}) = \arctan\left((\vec{\alpha}(\vec{z})^\top \vec{x}) + (\vec{\beta}(\vec{z})^\top \vec{x}) + (\vec{x})_s\right)$$

with input $\vec{x}$, sensor parameters $\vec{z}$, and crosstalk:

$$\alpha(\vec{z}) = \vec{\alpha}_0 + (\vec{z})_1\vec{\alpha}_1 + (\vec{z})_2\vec{\alpha}_2 \qquad \vec{\beta}(\vec{z}) = \vec{\beta}_0 + (\vec{z})_1\vec{\beta}_1 + (\vec{z})_2\vec{\beta}_2$$

SYNTHETIC SENSOR

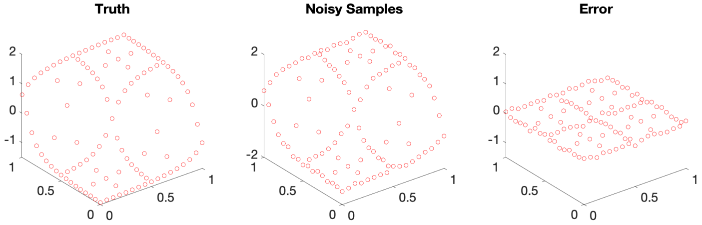
Each plot is a response curve for a synthetic sensor:



Response Curve Representation

A response curve could be represented by:

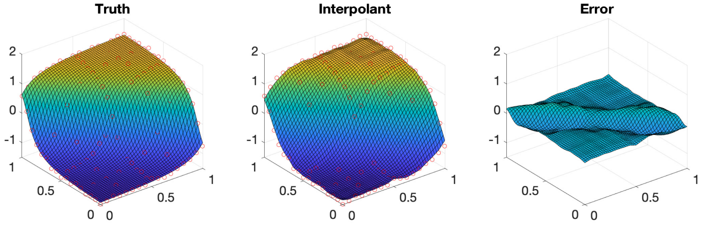
- A list of responses at each input grid point (vectorize)
- Coefficients in a basis (sparse grids used here)
- The true parameters $\vec{z}$ (unknown)
- A data-driven 'sensor space' (copy of $\vec{z}$)



Response Curve Representation

A response curve could be represented by:

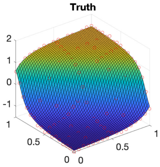
- A list of responses at each input grid point (vectorize)
- Coefficients in a basis (sparse grids used here)
- The true parameters $\vec{z}$ (unknown)
- A data-driven 'sensor space' (copy of $\vec{z}$)



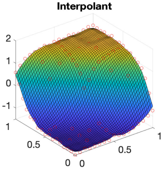
SENSOR SPACE

True parameters, $\vec{z}$
$\Rightarrow$ Response curve, $h(\vec{x},\vec{z})$
$\Rightarrow$ Sparse grid coefficients, $h(\vec{x},\vec{z}) = \sum_{i,j} c_{i,j}(\vec{z})\phi_{i,j}(\vec{x})$
$\Rightarrow$ Sensor space, $\mathcal{P}(\vec{c}(\vec{z}))$

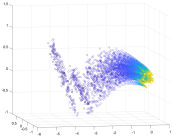
$\vec{z} \Rightarrow$



$\Rightarrow$

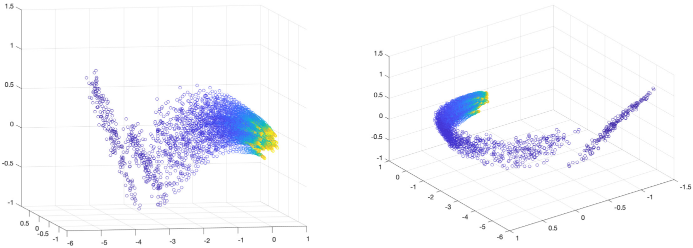


$\Rightarrow$



SENSOR SPACE

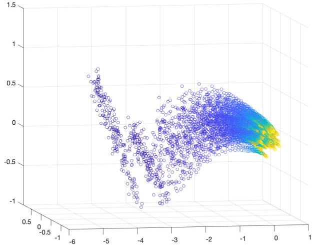
Projecting 200 response curves, each dot represents a sensor

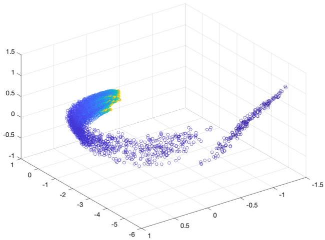


Response curves evolve over time, color represents time

Advantage of Sensor Space

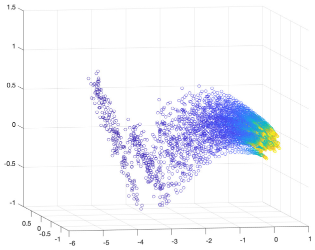
- Learn sensor space from a training set of sensors, sampled over time if possible
- Quick calibration: A few tests can determine location in sensor space
- Active calibration: Update sensor space location, eg. using Kalman filter

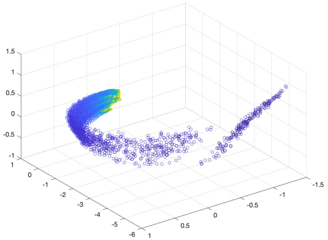




DIFFUSION MAP/FORECAST APPLICATION

- Represent sensor as a distribution on sensor space
- Diffusion map provides the basis functions
- Diffusion forecast can predict sensor drift
- No model required, purely data driven





FILTER MODEL

- Observation Function (sparse grid interpolation):

$$\vec{y}_k = h(\vec{x}_k, \vec{z}_k) + \vec{\nu}_k \qquad \vec{\nu}_k \sim \mathcal{N}(0, R).$$

- Sensor evolution (data-driven forecast):

$$\vec{z}_{k+1} = g(\vec{z}_k) + \vec{\eta}_k \qquad \vec{\eta}_k \sim \mathcal{N}(0, T).$$

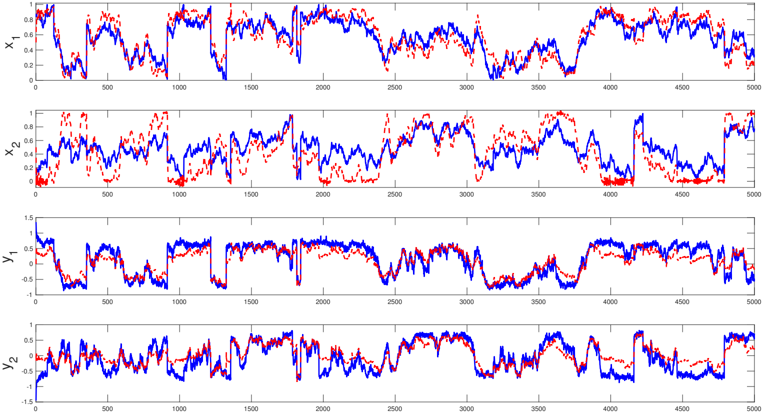
- State evolution (minimal continuity assumption):

$$\vec{x}_{k+1} = \vec{x}_k + \vec{\omega}_k \qquad \vec{\omega}_k \sim \mathcal{N}(0, Q).$$

- Kalman Filter:

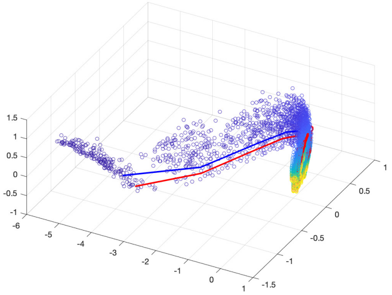
$$\begin{pmatrix} \hat{x}_{k+1} \\ \hat{z}_{k+1} \end{pmatrix} = \begin{pmatrix} \alpha \hat{x}_k \\ g(\hat{z}_k) \end{pmatrix} + K_k(\vec{y}_{k+1} - h(\alpha \hat{x}_k, g(\hat{z}_k)))$$

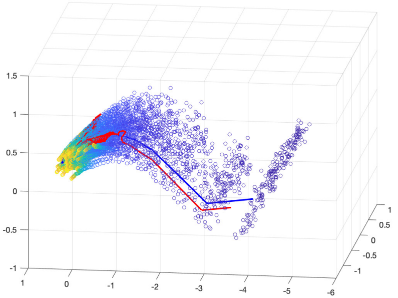
Filter Example



Filter Example

Recovering the location in sensor space:





FUTURE DIRECTIONS: SENSOR-ARRAY CALIBRATION

- Collection of sensor $\{y_i(t) = h_i(t)(x(t), z_i(t))\}_{i=1}^N$
- x is very high-dimensional, sensors only partially observe
- Build cross-sensor forecast models:

$$\hat{y}_i(t) = f_{ij}(\vec{y}_{I(i,j)}(t)) + \eta_{ij}(t)$$

- Choose predictors, $I(i,j)$ via cross-validation
- Use $\hat{y}_i - y_i$ to auto-calibrate and detect sensor failure

Code and papers available at:

http://math.gmu.edu/˜berry/

Building the basis

- B. and Sauer, *Consistent Manifold Representation for Topological Data Analysis.*
- Coifman and Lafon, *Diffusion maps.*
- B. and Harlim, *Variable Bandwidth Diffusion Kernels.*
- B. and Sauer, *Local Kernels and Geometric Structure of Data.*

Diffusion forecast

- B., Giannakis, and Harlim, *Nonparametric forecasting of low-dimensional dynamical systems.*
- B. and Harlim, *Forecasting Turbulent Modes with Nonparametric Diffusion Models.*