Towards a mathematical foundation for machine learning

Tyrus Berry

George Mason University

Sep. 27, 2021

Supported by NSF-DMS 1854204 and 2006808

WHY A MATHEMATICAL FOUNDATION?

Learning $f \in \mathcal{C}^s(\mathbb{R}^n, \mathbb{R})$ from N data points

- Fixed data set $\Rightarrow$ engineering problem
- Growing data set $\Rightarrow$ Evolving model $\Rightarrow$ Convergence
- Need to know that our algorithm has a limiting behavior
- Consider the infinite data limit to insure stability
- Ask if the limiting model is the truth
- Mathematical structures provide prior models for truth

VOLUME GROWS LIKE radius$^{\text{dimension}}$

Learning $f \in \mathcal{C}^s(\mathbb{R}^n, \mathbb{R})$ from N data points $\Rightarrow$ Error $\propto N^{-s/n}$

Many instances:

- Vapnik-Chervonenkis (VC) dimension [1]
- Rademacher complexity [2]
- Kolmogorov width [3]
- Interpolation error in approximation theory [3, 4, 5]
- Bias-variance tradeoff (density estimation/regression) [6, 1]
- Neural networks [7, 8] and sparse grids [9]

AVOIDING THE CURSE

Learning $f \in \mathcal{C}^s(\mathbb{R}^n, \mathbb{R})$ from N data points $\Rightarrow$ Error $\propto N^{-s/n}$

Coping mechanisms:

- **Smooth it away:** Assume f is very smooth, ie. $s \propto n$
- **Independence:** Assume $Y = f(X)$ is conditionally independent of X given $Z = g(X) \in \mathbb{R}^m$ with $m \ll n$.
- **Redundancy:** Assume $h(X) = 0$ for some $h \in \mathcal{C}^{m+1}(\mathbb{R}^n, \mathbb{R}^{n-m})$.

SLOW CHANGE REQUIRES FEW NEIGHBORS

All machine learning methods interpolate from neighbors:

- **kNN and Local Linear Regression** (x_{kNN} is k-th nearest neighbor of x):

$$F(x) \approx \frac{1}{k} \sum_{||x-x_j|| \leq ||x-x_{\text{kNN}}||} F(x_j) + a^\top (x - x_j)$$

- **Kernel Regression** (h is bump function, eg. $h(s) = \exp(-s^2)$):

$$F(x) \approx \sum_j c_j h((x - x_j)^\top A_j (x - x_j))$$

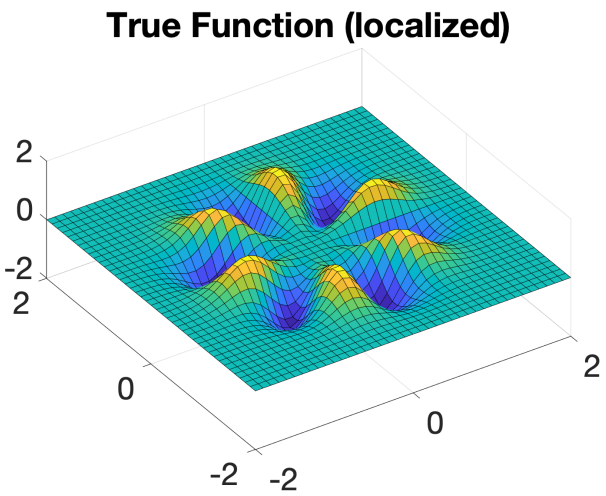
- **Neural Network** (h is typically a sigmoid, but can also be a bump):

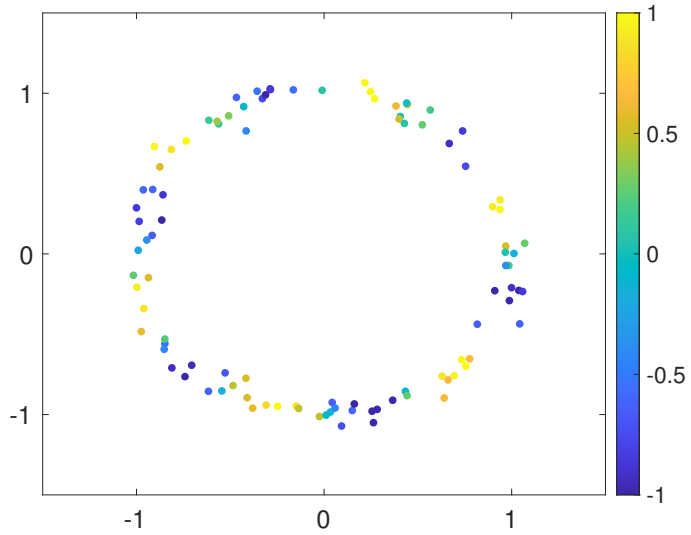
$$F(x) \approx \sum_j c_j h(a_j^\top x + b_j) = \sum_j c_j h(a_j^\top (x - \tilde{x}_j))$$

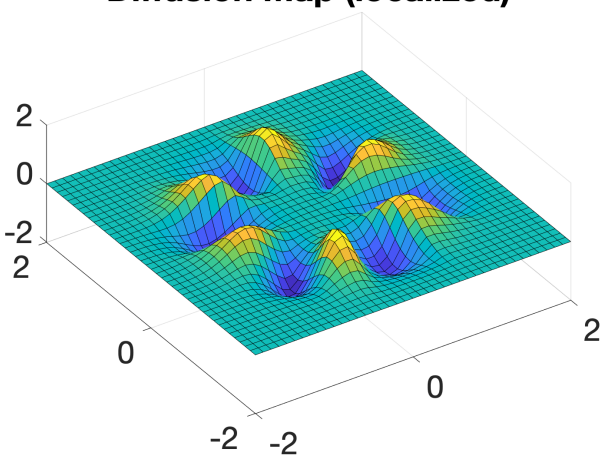
(where we write $b_j = -a_j^\top \tilde{x}_j$)

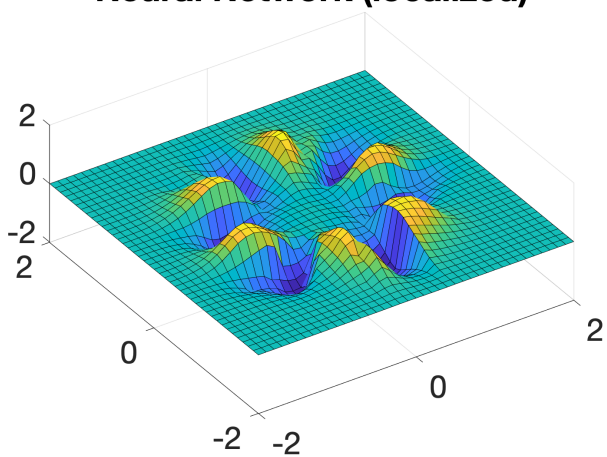
- **Reservoir Computer**: Fix a_j, b_j, regression to find c_j

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

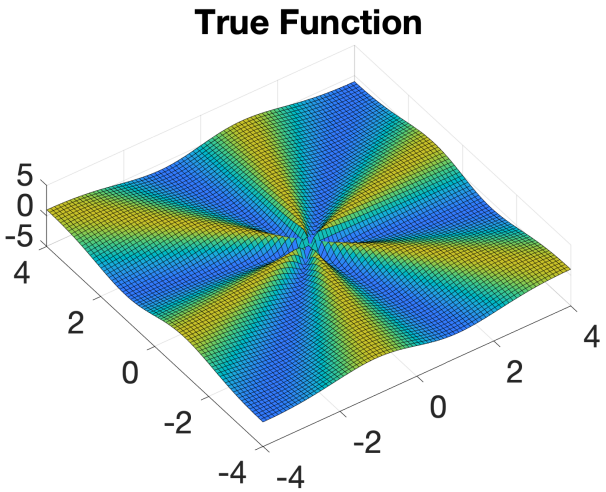


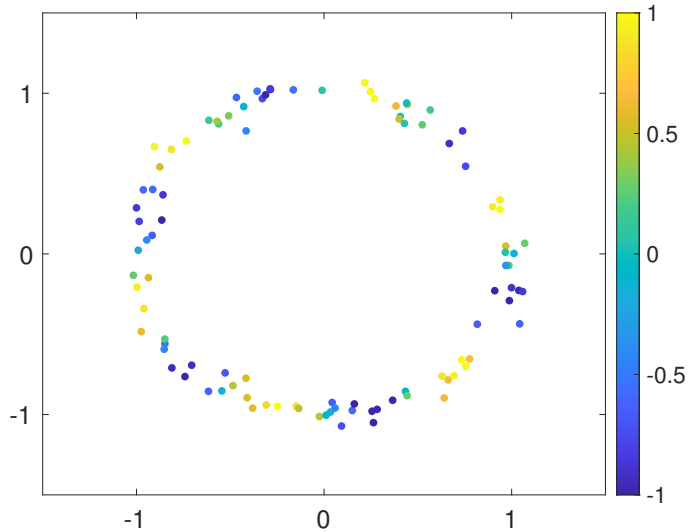


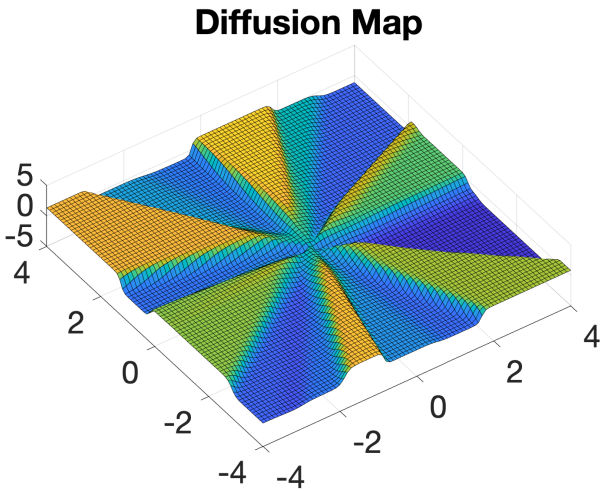


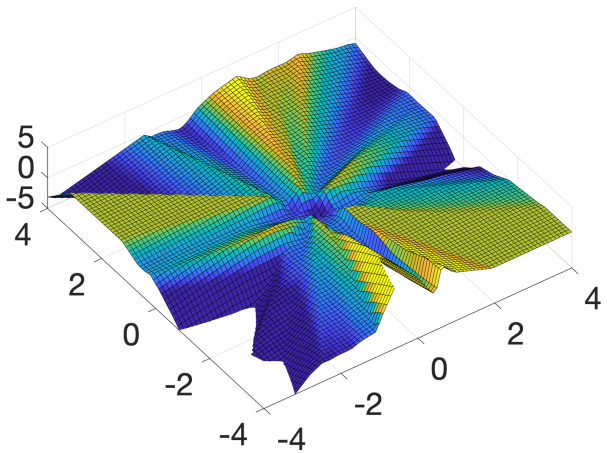


NYSTRÖM VS. DEEP NET, $(r, \theta) \mapsto \sin(6\theta)$

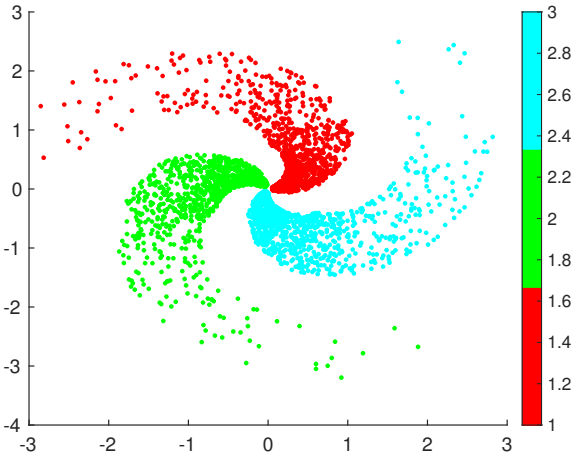


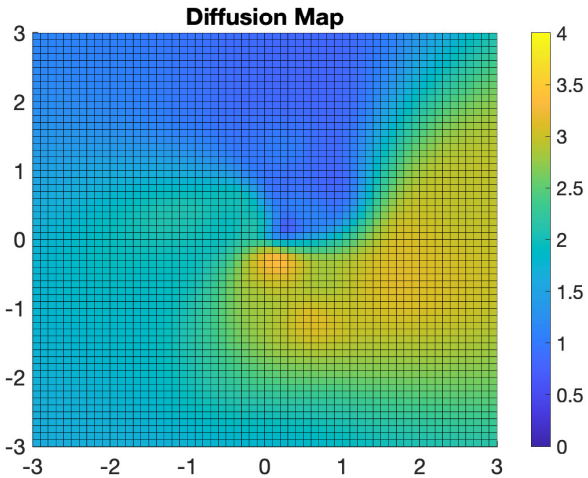


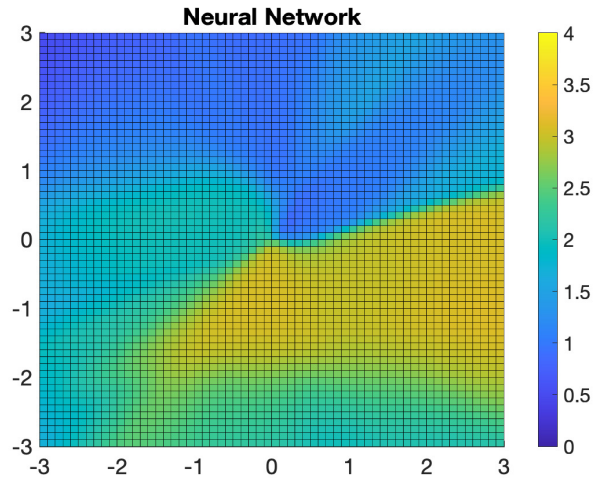




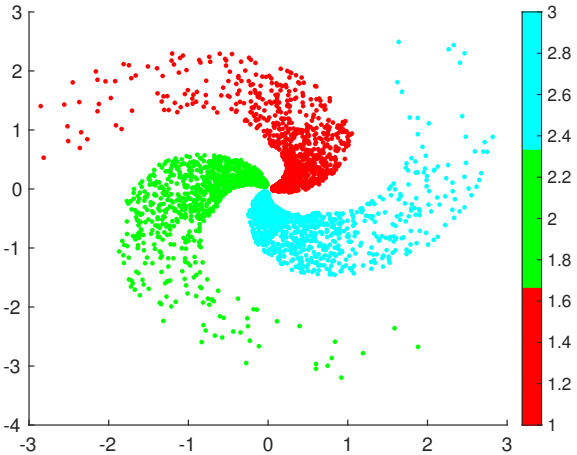
NYSTRÖM VS. DEEP NET, EXTRAPOLATION

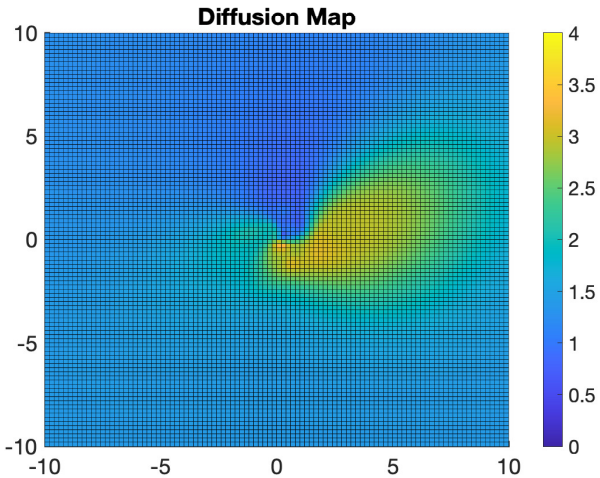


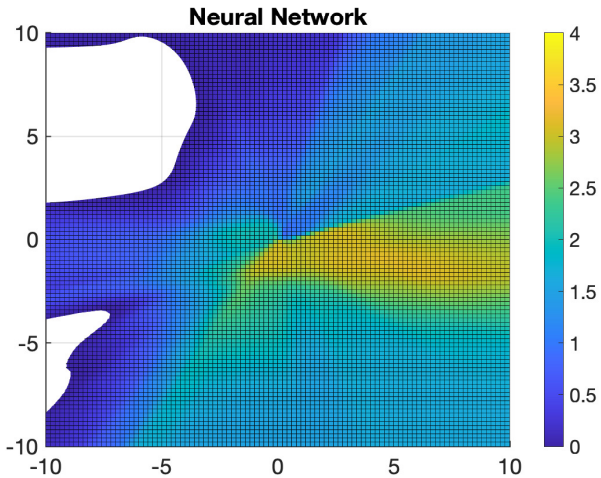




NYSTRÖM VS. DEEP NET, EXTRAPOLATION







INDEPENDENCE

Learning $f \in \mathcal{C}^s(\mathbb{R}^n, \mathbb{R})$ from N data points $\Rightarrow$ Error $\propto N^{-s/n}$

- Want to learn $Y = f(X)$ where $f : \mathbb{R}^n \to \mathbb{R}$
- Assume there is a projection $\beta \in \mathbb{R}^{n \times m}$ such that

$$Y \perp\!\!\!\perp X \,|\, \beta^\top X$$

- Find β using Sliced Inverse Regression (SIR) [10, 11]
- Learn $Y = \tilde{f}(\beta^\top X)$ since $\tilde{f} : \mathbb{R}^m \to \mathbb{R}$, $m \ll n$

INDEPENDENCE

Detect person in crosswalk

Lots of variability, most is irrelevant

INDEPENDENCE

More generally:

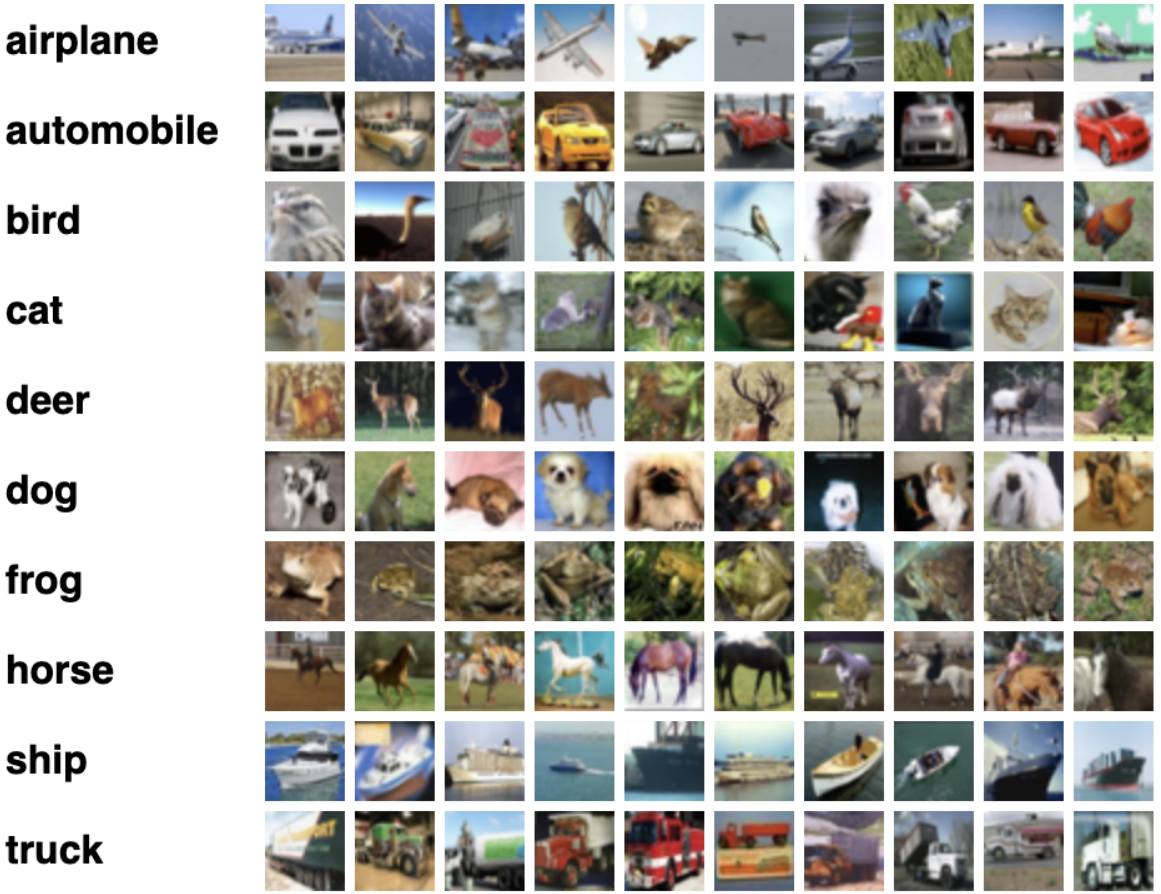
- Want to learn $P(Y \mid X)$
- Assume there is a map $\beta : \mathbb{R}^n \to \mathbb{R}^m$ such that

$$Y \perp\!\!\!\perp X \mid \beta(X)$$

- If we can find β...
- Learning $P(Y \mid \beta(X))$ may be feasible

INDEPENDENCE

CIFAR has many irrelevant modes



But they are combined nonlinearly with features

REDUNDANCY

Unlike smoothness and independence, f is not involved

- Redundancy assumes that most of $X \in \mathbb{R}^n$ is repeats
- E.g. $x_n = a_1 x_1 + \cdots + a_{n-1} x_{n-1}$ is a linear redundancy
- More generally if $AX = 0$ for some $A \in \mathbb{R}^{(n-m)\times n}$
- X appears n-dim'l (extrinsic) but is really m-dim'l (intrinsic)
- PCA finds $A^\perp X \in \mathbb{R}^m$ where $[A\ A^\perp]$ is a basis
- The reduction helps learn any f

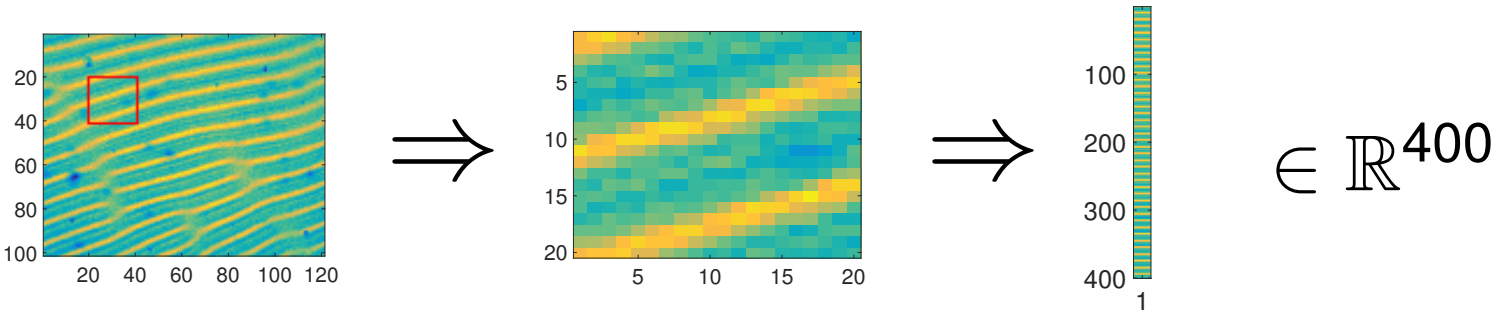
REDUNDANCY

- More generally assume $h(X) = 0$ for some $h : \mathbb{R}^n \to \mathbb{R}^{n-m}$
- **Sard's lemma:** If $h \in \mathcal{C}^{m+1}(\mathbb{R}^n, \mathbb{R}^{n-m})$ then regular values are dense in $\mathbb{R}^{n-m}$, so either 0 is regular or ϵ is regular
- **Regular Value Theorem:** The pre-image of a regular value under a smooth map is a manifold of dimension

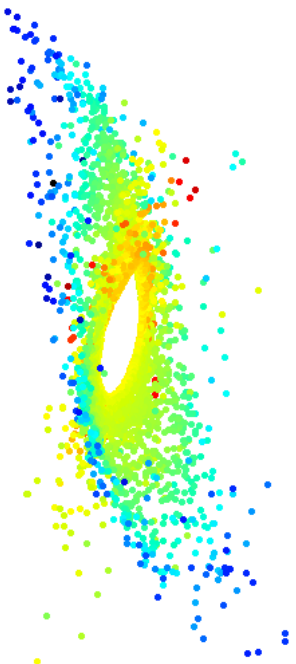
$$\dim(\text{domain}) - \dim(\text{range})$$

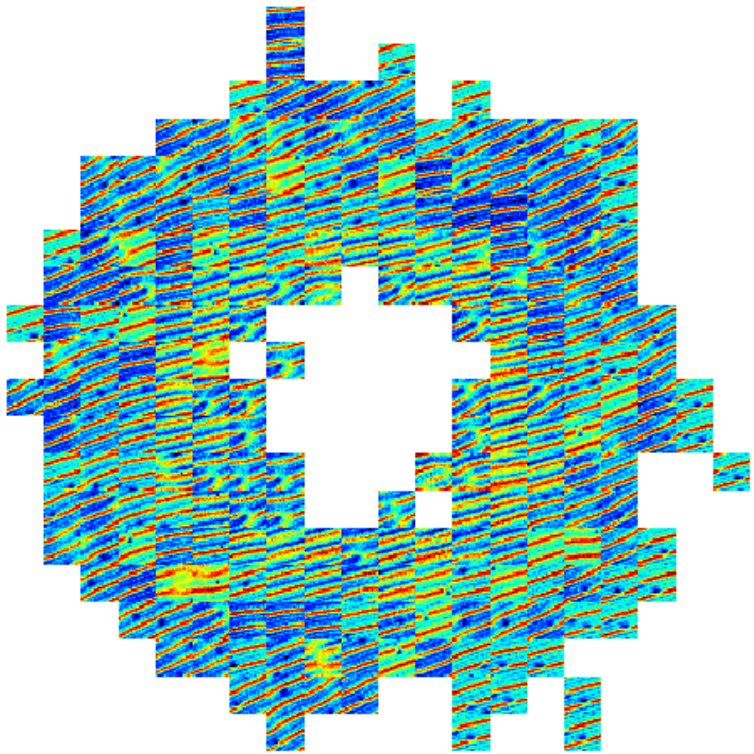
- **Upshot:** If $h(X) = 0 \in \mathbb{R}^{n-m}$ are smooth redundancies then $X = h^{-1}(0)$ is a manifold of dimension m
- Manifold learning leverages this nonlinear structure

Finding Hidden Structure in Data



The sub-image geometry:





What is Manifold Learning?

- Geometric prior: Data on Riemannian manifold $\mathcal{M} \subset \mathbb{R}^m$
- **Goal**: Represent all the information about a manifold
- A smooth embedded manifold $\mathcal{M} \subset \mathbb{R}^m$ inherits:
  - A metric tensor $g_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ (inner product)
  - g completely determines the geometry of $\mathcal{M}$
  - A volume form $dV(x) = \sqrt{\det(g_x)}\, dx^1 \wedge \cdots \wedge dx^d$
- Laplace-Beltrami operator, Δ, is equivalent to g
  - $\Delta f = \text{div} \circ \nabla = \frac{1}{\sqrt{|g|}} \partial_i g^{ij} \sqrt{|g|} \partial_j f$
  - $g(\nabla f, \nabla h) = \frac{1}{2}(f\Delta h + h\Delta f - \Delta(fh))$

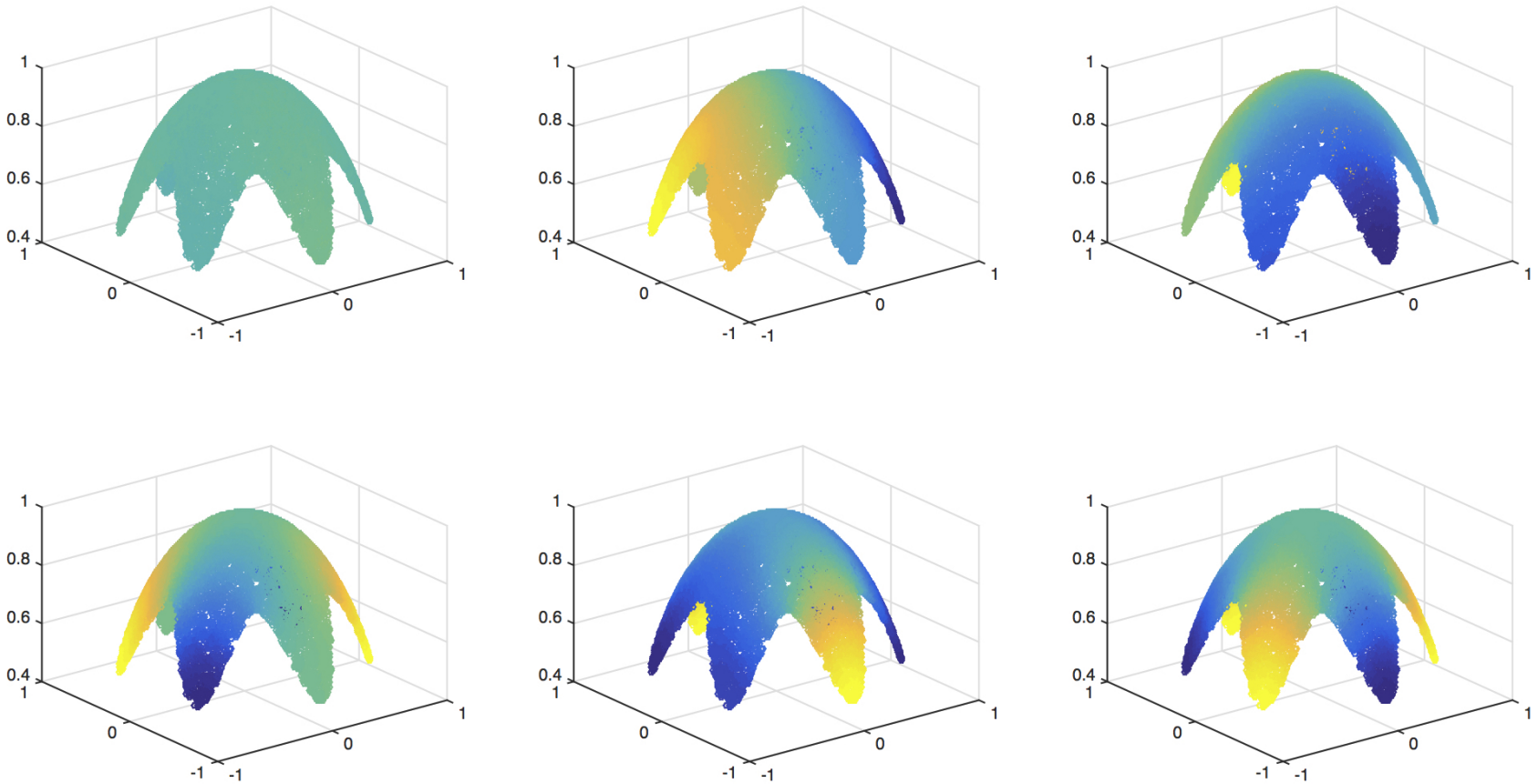
What is Manifold Learning?

- **Manifold learning $\Leftrightarrow$ Estimating Laplace-Beltrami**
- **Hodge theorem:**
  Eigenfunctions $\Delta\varphi_i = \lambda_i\varphi_i$ orthonormal basis for $L^2(\mathcal{M}, g)$
- Smoothest functions: φ_i minimizes the functional

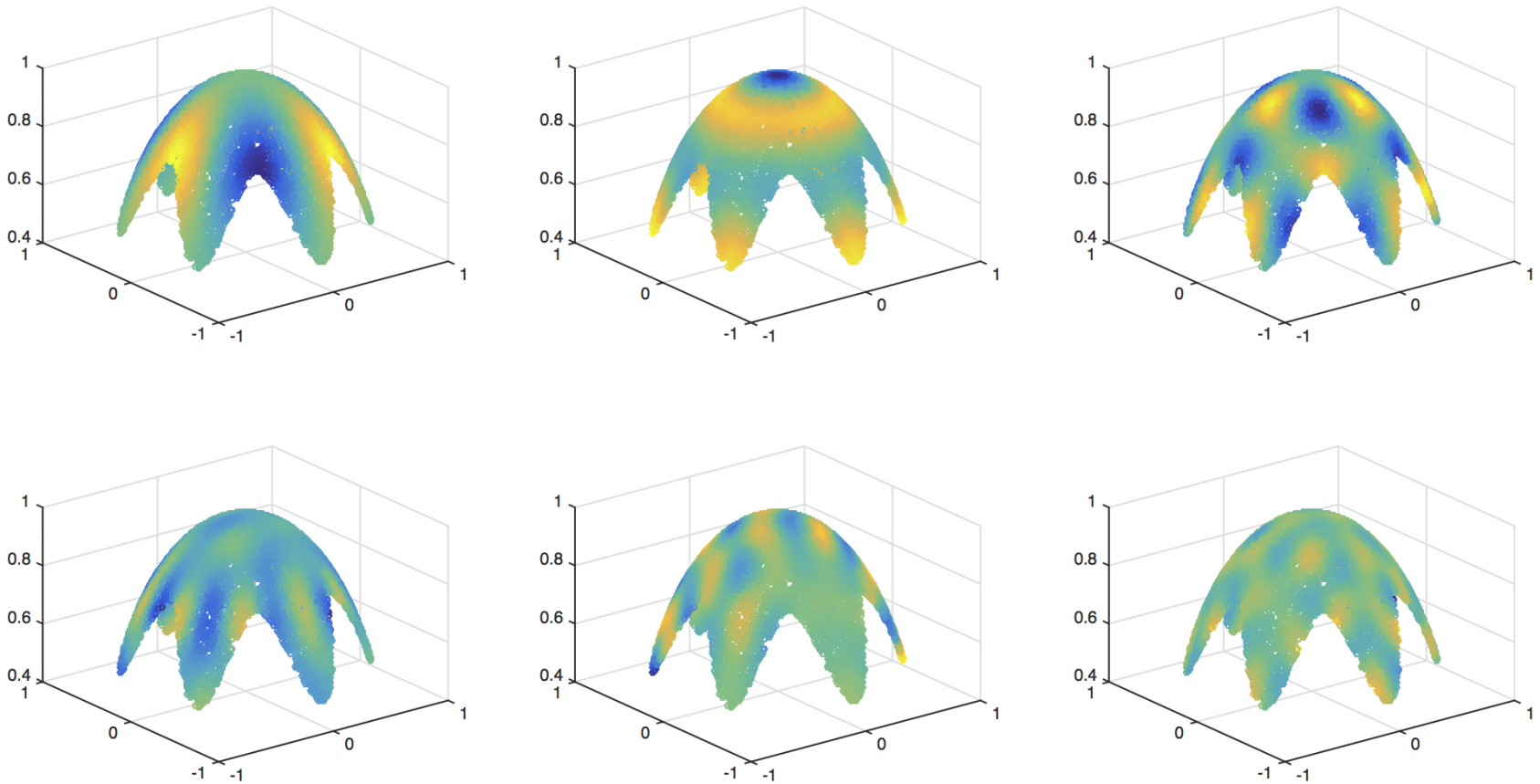
$$\lambda_i = \min_{\substack{f \perp \varphi_k \\ k=1,\ldots,i-1}} \left\{ \frac{\int_{\mathcal{M}} ||\nabla f||^2 \, dV}{\int_{\mathcal{M}} |f|^2 \, dV} \right\}$$

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

HARMONIC ANALYSIS ON MANIFOLDS

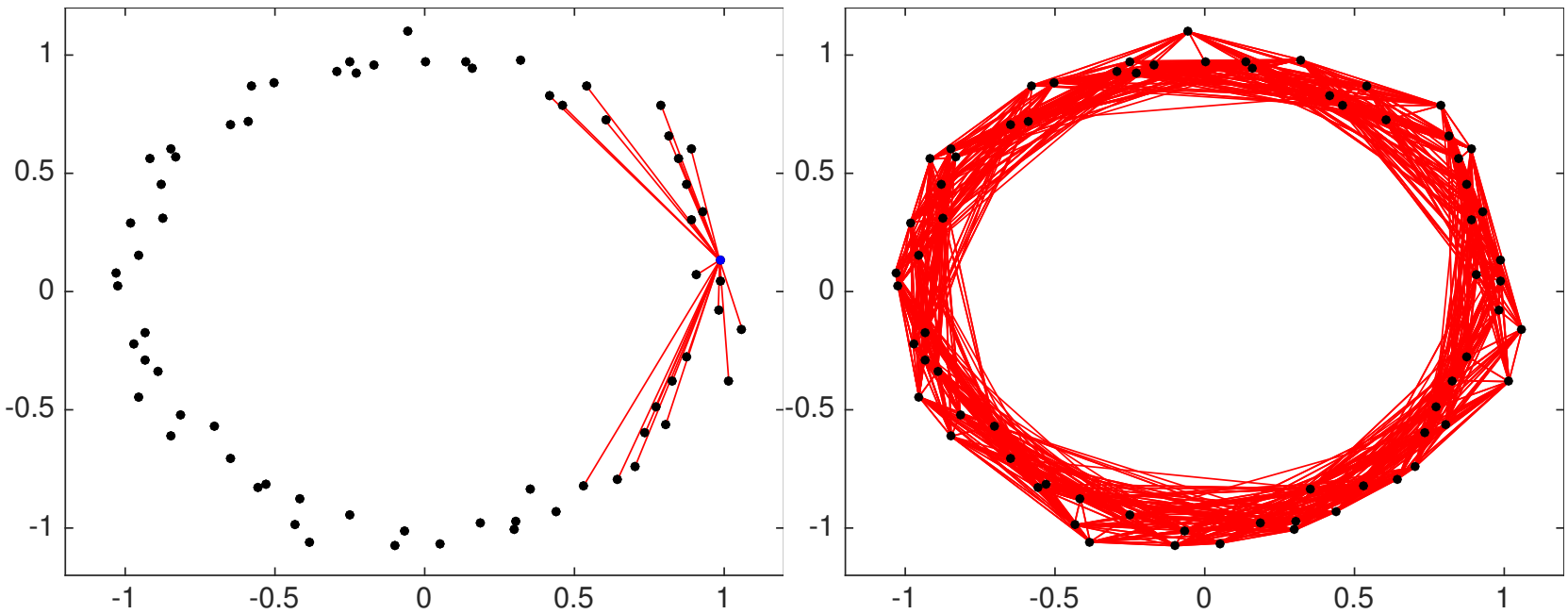


HARMONIC ANALYSIS ON MANIFOLDS



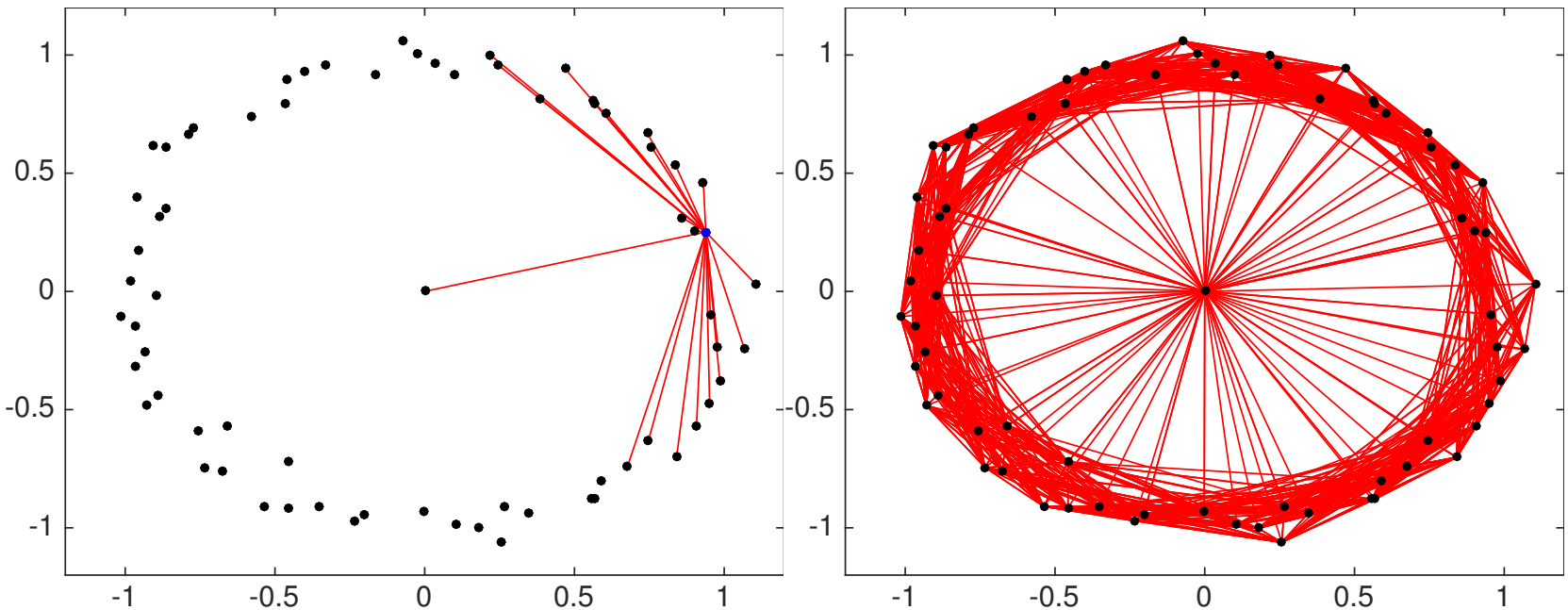
So how do we find the Laplacian from data?

- Assume data lies on (or at least near) a manifold
- Approximate manifold with graph $\Rightarrow$ Connect nearby points



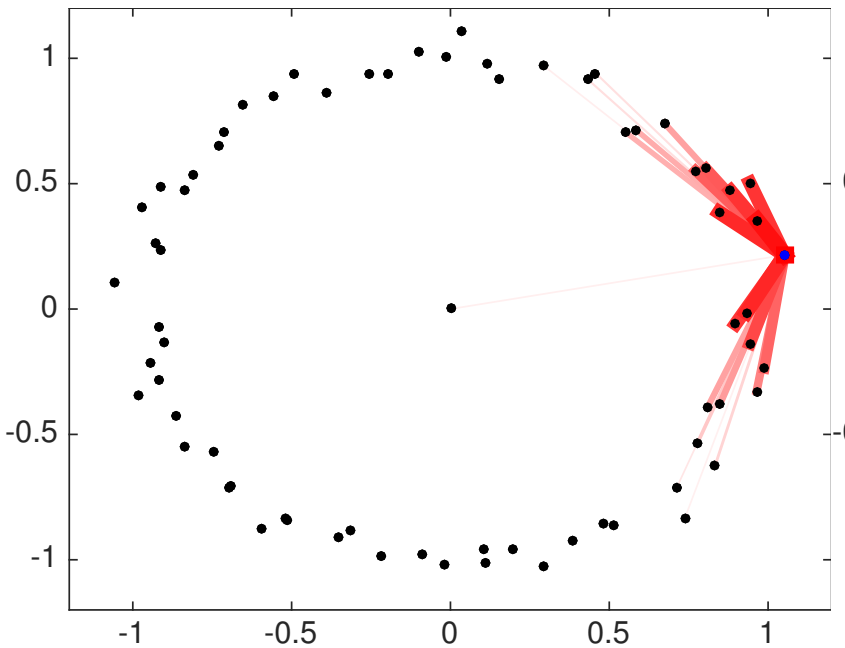
So how do we find the Laplacian from data?

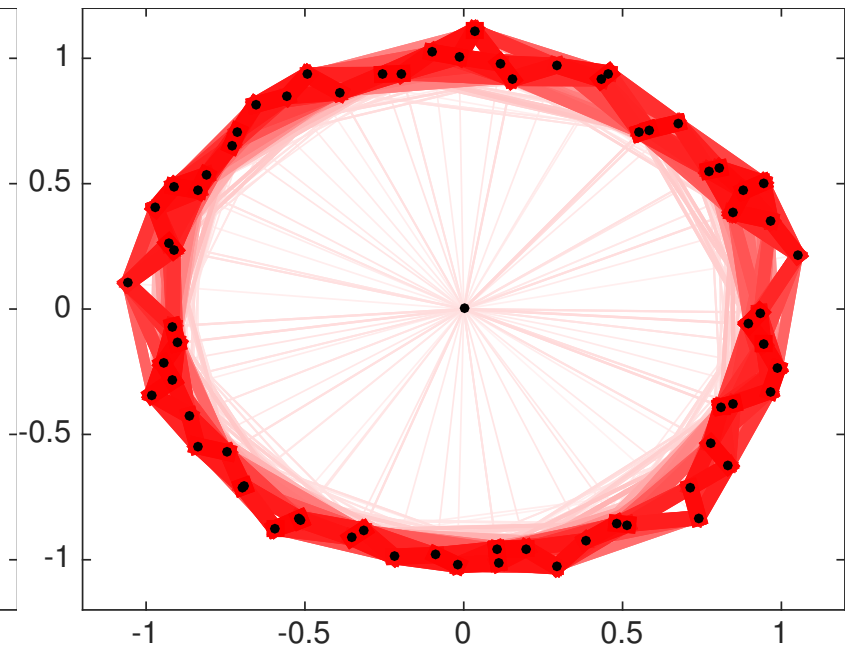
- **Problem:** Noise and outliers can lead to *bridging*



So how do we find the Laplacian from data?

- To prevent bridging we weight the edges
- Edges are given weights $K_\delta(x, y) = e^{-\frac{||x-y||^2}{4\delta^2}}$





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

Discrete analogs of continuous objects

Continuous	Discrete
$L^2(\mathcal{M}, q)$	$\mathbb{R}^N$
Functions, $f : \mathcal{M} \to \mathbb{R}$	Vectors, $\vec{f}_i = f(x_i)$
'Basis', δ_x	Basis, $\vec{e}_i = \delta_{x_i}$
Laplace-Beltrami, Δ	Normalized Graph Laplacian, $\mathbf{L}$
Eigenfunctions, $\Delta\varphi_j = \lambda_j\varphi_j$	Eigenvectors, $\mathbf{L}\vec{\varphi}_j = \lambda_j\vec{\varphi}_j$
Inner product, $\langle f, h\rangle_{L^2}$	Dot Product, $\frac{1}{N}\vec{f}\cdot\vec{h}$

$$\frac{1}{N}\vec{f}\cdot\vec{h} = \frac{1}{N}\sum_{i=1}^{N} f(x_i)h(x_i) \to_{N\to\infty} \int_{\mathcal{M}} f(x)h(x)\,dV(x)$$

RESTRICTIONS THAT HAVE BEEN OVERCOME TO DEAL WITH REAL DATA:

- Arbitrary sampling (Coifman & Lafon, 'Diffusion maps', 2006)
- Other kernel functions (Berry & Sauer, 2015)
- Non-compact manifolds (Berry & Harlim, 2015)
- Boundary (R. Vaughn Thesis 2020)

$$\vec{h}^{\top} L\vec{f} \to \int \nabla h \cdot \nabla f \, dV$$

- Spectral convergence (von Luxburg et al. 2008, Trillos et al. 2020, Berry & Sauer 2019)

Conformally Invariant Diffusion Maps (CIDM)

- Data samples $\{x_i\}_{i=1}^N \subset \mathcal{M} \subset \mathbb{R}^n$ of volume $p_{\text{eq}}\, dV$
- 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:** $L\vec{f} = \Delta_{\hat{g}} f + \mathcal{O}\left(\delta^2, N^{-1/2}\delta^{-1-d/2}\right)$, $\hat{g} = p_{\text{eq}}^{2/d} g$
- **Solve:** $(I - D^{-1/2} K D^{-1/2})\vec{v} = \lambda \vec{v}$, set $\vec{\varphi} = D^{-1/2}\vec{v}$

Beyond Manifold Learning

- Data never really lies on a manifold (due to noise)
- A manifold is a measure zero set
- Data is never sampled from a measure zero set
- **Solution 1:** Spectral robustness for bounded noise (Coifman and Lafon), but lose convergence
- **Solution 2:** Manifold + Noise, requires semi-geodesic coordinates, need new algorithms to regain convergence
- **Solution 3:** Generalize beyond manifolds
  - Metric measure spaces
  - Gromov-Hausdorff limits of manifolds

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

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 (Stochastic Koopman)

- 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

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

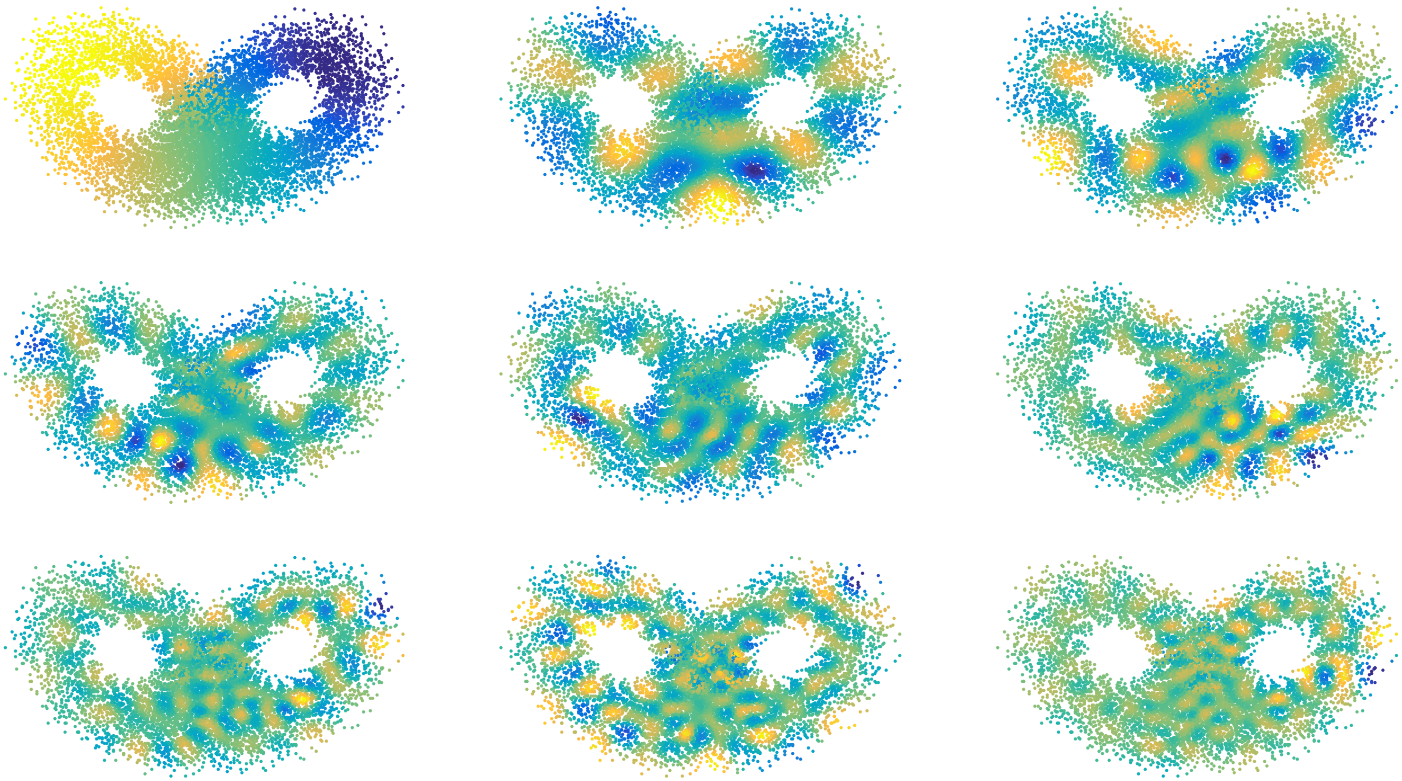
- Estimate A_{lj} with $\hat{A}_{lj} = \frac{1}{N} \sum_{i=1}^{N} \varphi_j(x_i)\varphi_l(x_{i+1})$
- $\mathbb{E}[\hat{A}_{lj}] = A_{lj}$ with error $\mathcal{O}(||\nabla \varphi_l||_{p_{\text{eq}}} \sqrt{\tau/N})$

CHOOSING A BASIS

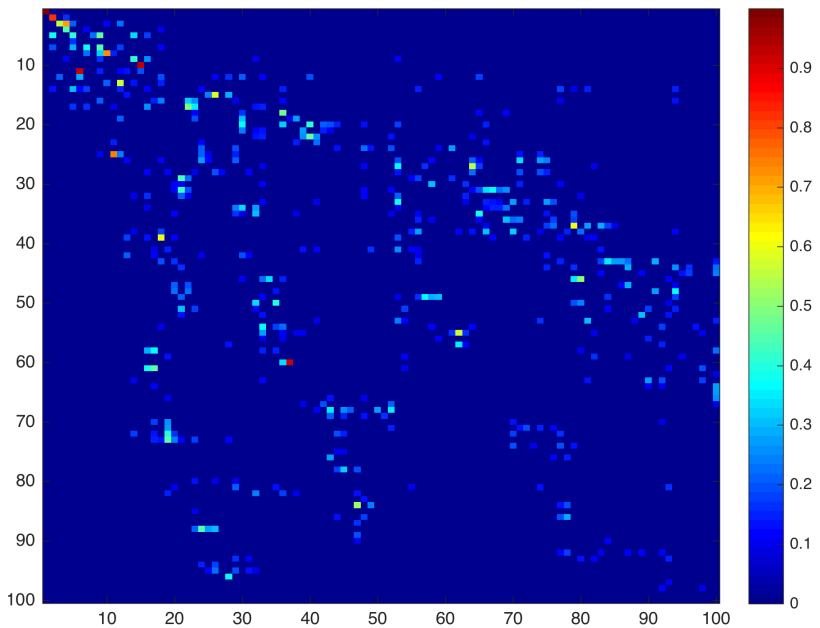
- Need to minimize the error term $\mathcal{O}(||\nabla\varphi_l||_{p_{\text{eq}}}\sqrt{\tau/N})$
- The eigenfunctions $\Delta_{\hat{g}}\varphi_j = \lambda_j\varphi_j$ minimize $||\nabla\varphi_j||_{p_{\text{eq}}} = \lambda_j$
- Find φ_j with Manifold Learning: CIDM

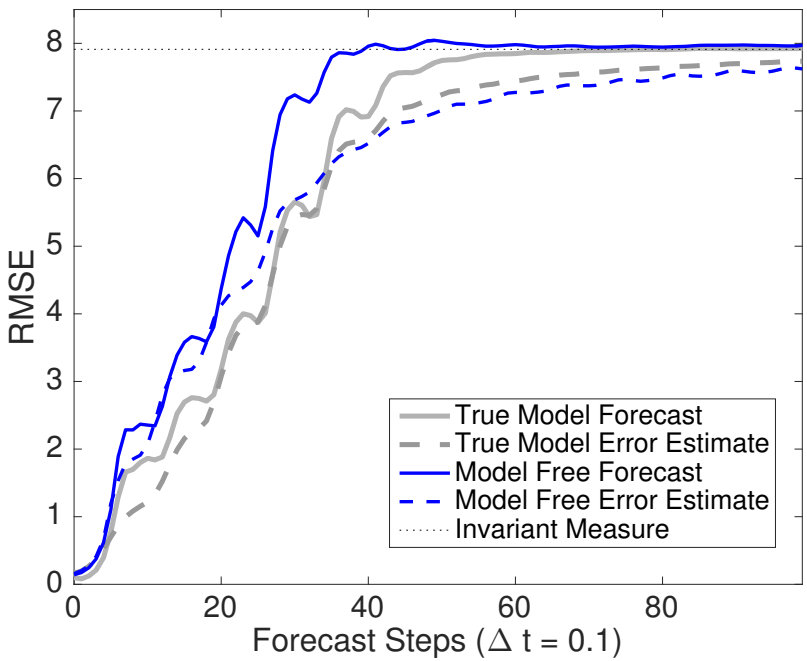
MANIFOLD LEARNING ⇒ CUSTOM 'FOURIER' BASIS

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



SHIFT MAP ⇒ MARKOV MATRIX



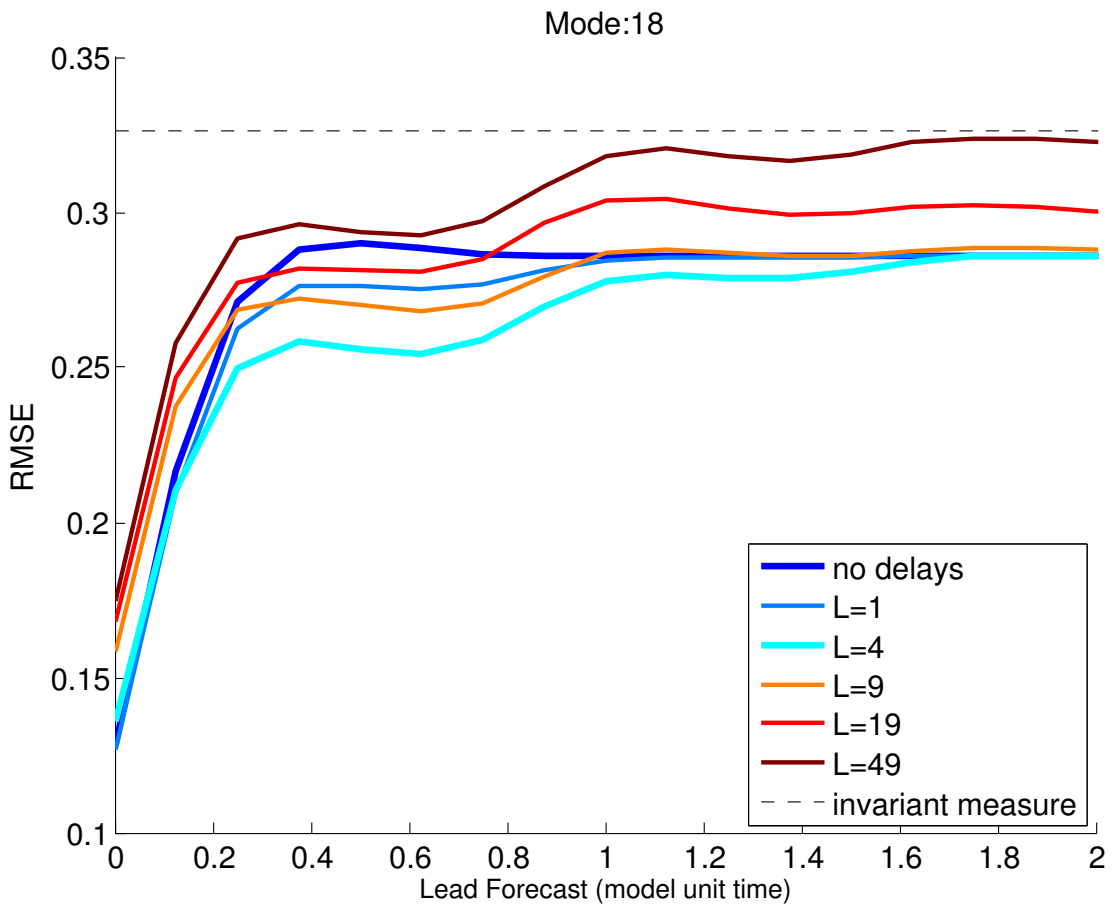


Diffusion Forecast Example

(Loading Video...)

Attractor Reconstruction

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



Code and papers available at:

http://math.gmu.edu/~berry/

Manifold Learning Papers Discussed

- B. and Giannakis, *Spectral Exterior Calclulus.*
- R. Vaughn *Diffusion Maps for Manifolds with Boundary.*
- 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.*

References

[1] V. Vapnik, The nature of statistical learning theory. Springer (2000).

[2] M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of machine learning. MIT press (2018).

[3] A. Pinkus, N-widths in Approximation Theory. Vol. 7. Springer Science & Business Media, (2012).

[4] R. A. DeVore and G. G. Lorentz. Constructive approximation. Vol. 303. Springer Science & Business Media, (1993).

[5] R. A. DeVore, Nonlinear approximation. Acta numerica 7, 51-150 (1998).

[6] D. W. Scott and S. R. Sain. Multidimensional density estimation. Handbook of statistics 24, 229-261 (2005).

[7] A. R. Barron, Universal approximation bounds for superpositions of a sigmoidal function. IEEE Transactions on Information theory 39, no. 3, 930-945 (1993).

[8] A. R. Barron, Approximation and estimation bounds for artificial neural networks. Machine learning 14, no. 1, 115-133 (1994).

[9] H. J. Bungartz and M. Griebel, Sparse grids. Acta numerica, 13, 147-269 (2004).

[10] K. Li, Sliced Inverse Regression for Dimension Reduction. Journal of the American Statistical Association, 86(414), (1991).

[11] Y. Li, L. Zhu, Asymptotics for Sliced Average Variance Estimation. The Annals of Statistics, 35(1), 41-69 (2007).