On learning kernels for numerical approximation and learning

Houman Owhadi

JHU Applied Math & Statistics department seminar, March 25, 2021
Interpolation problem

Recover \( f^\dagger : D \subset \mathbb{R}^d \to \mathbb{R} \)

Given \( f^\dagger(X_1), \ldots, f^\dagger(X_N) \)

Family of kernels

\( K_\theta : D \times D \to \mathbb{R} \)

\( \theta \): Hierarchical parameter

Kernel/GP interpolant

\[
f(\cdot, \theta, X) = K_\theta(\cdot, X)K_\theta(X, X)^{-1}f^\dagger(X)
\]

\( f^\dagger(X) := (f^\dagger(X_1), \ldots, f^\dagger(X_N)) \in \mathbb{R}^N \)

\( K_\theta(X, X) \): \( N \times N \) matrix with entries \( K_\theta(X_i, X_j) \)

\( K_\theta(x, X) \): \( 1 \times N \) vector with entries \( K_\theta(x, X_i) \)

Question

Which \( \theta \) do we pick?
Main objectives of this talk

Show why this question is important
Cover the following answers

- Bayesian (MLE, MAP)
- Cross validation
- Deep Learning (Bayesian, MAP)

Kernel Flows: from learning kernels from data into the abyss.
Journal of Computational Physics, 2019

Empirical Bayes answer

Place a prior on $\theta$

Assume that $f^{\dagger}|\theta \sim \mathcal{N}(0, K_\theta)$

Select the $\theta$ maximizing the marginal probability of $\theta$ subject to conditioning on $f^{\dagger}(X)$

Uninformative prior on $\theta$

Maximum Likelihood Estimate

$$\theta^{EB} = \arg\min_{\theta} L^{EB}(\theta, X, f^{\dagger})$$

$$L^{EB}(\theta, X, f^{\dagger}) = f^{\dagger}(X)^T K_\theta(X, X)^{-1} f^{\dagger}(X) + \log \det K_\theta(X, X)$$
Kernel Flow answer (Variant of cross-validation, O., Yoo, 2019)

Pick a $\theta$ such that subsampling the data does not influence the interpolant much

$$\theta^{KF} = \underset{\theta}{\text{argmin}} \ L^{KF}(\theta, X, \pi X, f^\dagger)$$

$$L^{KF}(\theta, X, \pi X, f^\dagger) = \frac{\left\| f(\cdot, \theta, X) - f(\cdot, \theta, \pi X) \right\|_K^2}{\left\| f(\cdot, \theta, X) \right\|_K^2}$$

$$f(\cdot, \theta, X) = K_\theta(\cdot, X)K_\theta(X, X)^{-1}f^\dagger(X)$$

$\pi$: subsampling operator, $\pi X$ is a subvector of $X$

$$\| \cdot \|_K: \text{RKHS norm determined by } K_\theta$$

A kernel is good if subsampling the data does not influence the interpolant much
How do $\theta^{EB}$ and $\theta^{KF}$ behave as $\#$ of data $\to \infty$?

**Model**

- Domain $D = \mathbb{T}^d = [0, 1]_\text{per}^d$
- Lattice data $X_q = \{j \cdot 2^{-q}, j \in J_q\}$
  where $J_q = \{0, 1, \ldots, 2^q - 1\}^d$, $\#$ of data $2^{qd}$
- Kernel $K_\theta = (-\Delta)^{-\theta}$
- Subsampling in KF: $\pi X_q = X_{q-1}$

**Theorem** *(Chen, O., Stuart, 2020)*

If $f^{\dagger} \sim \mathcal{N}(0, (-\Delta)^{-s})$ for some $s > d/2$, then as $q \to \infty$

$\theta^{EB} \to s$ and $\theta^{KF} \to \frac{s - \frac{d}{2}}{2}$ in probability
How are the limits $s$ and $\frac{s-d}{2}$ special?

$d = 1$, $s = 2.5$, # of data $N = 2^9$

$L^2$ error vs $\theta$  
$L^2$ error vs $\log(\# \text{ data points})$

- $s = 2.5$ is the $\theta$ that minimizes the mean squared error
- $\frac{s-d}{2} = 1$ is the smallest $\theta$ that suffices to achieve fastest rate in $L^2$
Takeaway message

- EB selects the $\theta$ that minimizes the mean squared error.
- KF selects the smallest $\theta$ that suffices for the fastest rate of convergence in mean squared error.

More comparisons

- EB may be brittle (not robust) to model misspecification
- KF has some degree of robustness to model misspecification


Extrapolation problem

Given time series $z_1, \ldots, z_N$

predict $z_{N+1}, z_{N+2}, z_{N+3}, \ldots$

Assumption

$$z_{k+1} = f^\dagger (z_k, \ldots, z_{k-\tau^\dagger + 1})$$

$f^\dagger$, $\tau^\dagger$ unknown

Fundamental problem

[Box, Jenkins, 1976]: Time Series Analysis
Mezíc, Klus, Budišíć, R. Mohr,...: Koopman operator
[Alexander, Giannakis, 2020]: Operator theoretic framework
[Bittracher et al, 2019]: kernel embeddings of transition manifolds
[Brunton, Proctor, Kutz, 2016]: SINDy
Brian, Hunt, Ott, Pathak, Lu, Hunt, Girvan, Ott,...: Reservoir computing
Ralaivola, Chattopadhyay,...: LSTM
Dietrich, Mahdi Kooshkbaghi, Bollt, Kevrekidis: Manifold learning
Simplest solution

Approximate $f^\dagger$ with Kernel interpolant $f$

$$f(z_k, \ldots, z_{k-\tau^\dagger+1}) = z_{k+1}\quad k = \tau^\dagger, \tau^\dagger + 1, \ldots, N - 1$$

$$f(x) = K(x, X)K(X, X)^{-1}Y$$

$$X_k = (z_k, \ldots, z_{k-\tau^\dagger+1})$$

$$Y_k = z_{k+1} = f^\dagger(X_k)$$

Predict future values of the time series by simulating the dynamical system

$$s_{k+1} = f(s_k, \ldots, s_{k-\tau^\dagger+1})$$

Example: Bernoulli map

\[ z_{k+1} = 2z_k \mod 1 \]

\[ K(x, x') = e^{-\|x-x'\|^2} \]
Example: Bernoulli map

\[ z_{k+1} = 2z_k \mod 1 \]

\[
K(x, x') = \alpha_0 \max\{0, 1 - \frac{\|x-x'\|^2}{\sigma_0}\} + \alpha_1 e^{-\frac{\|x-x'\|^2}{\sigma_1^2}}
\]

\[ \alpha_0, \sigma_0, \alpha_1, \sigma_1^2: \]
Learned parameters
(using Kernel Flows)

True dynamic

Predicted dynamic
Example: Hénon map

\[
x(k+1) = 1 - ax(k)^2 + y(k)
\]

\[
y(k+1) = bx(k)
\]

\[
K(x, x') = \begin{pmatrix} k_1(x, x') & 0 \\ 0 & k_2(x, x') \end{pmatrix}
\]

\[
k_i(x, y) = \alpha_i + (\beta_i + \|x - y\|_2^\kappa)\sigma_i + \delta_i e^{-\|x - y\|_2^2/\mu_i^2}
\]
Example: Lorenz system

\[
\begin{align*}
\frac{dx}{dt} &= s(y - x) \\
\frac{dy}{dt} &= rx - y - xz \\
\frac{dz}{dt} &= xy - bz
\end{align*}
\]

\[
k_i(x, y) = \alpha + (\beta_i + ||x - y||^k_i) \sigma_i + \delta_i e^{-||x - y||^2_i / \mu_i^2}
\]

True dynamic \quad Predicted dynamic with learned kernel

| \( \mathbf{x} \) | \( \mathbf{y} \) | \( \mathbf{z} \) |
Data-driven geophysical forecasting

HYCOM: 800 core-hours per day of forecast on a Cray XC40 system

CESM: 17 million core-hours on Yellowstone, NCAR’s high-performance computing resource

Architecture optimized LSTM: 3 hours of wall time on 128 compute nodes of the Theta supercomputer.

Our method: 40 seconds to train on a single node machine (laptop) without acceleration

Data-driven geophysical forecasting: Simple, low-cost, and accurate baselines with kernel methods, Hamzi, Maulik, O.
NOAA-SST data set (low noise dataset)

(a) Prediction error

(b) CESM error

<table>
<thead>
<tr>
<th></th>
<th>Week 1</th>
<th>Week 2</th>
<th>Week 3</th>
<th>Week 4</th>
<th>Week 5</th>
<th>Week 6</th>
<th>Week 7</th>
<th>Week 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAS-LSTM</td>
<td>0.62</td>
<td>0.63</td>
<td>0.64</td>
<td>0.66</td>
<td>0.63</td>
<td>0.66</td>
<td>0.69</td>
<td>0.65</td>
</tr>
<tr>
<td>CESM</td>
<td>1.88</td>
<td>1.87</td>
<td>1.83</td>
<td>1.85</td>
<td>1.86</td>
<td>1.87</td>
<td>1.86</td>
<td>1.83</td>
</tr>
<tr>
<td>HYCOM</td>
<td>0.99</td>
<td>0.99</td>
<td>1.03</td>
<td>1.04</td>
<td>1.02</td>
<td>1.05</td>
<td>1.03</td>
<td>1.05</td>
</tr>
<tr>
<td>Predicted</td>
<td>0.76</td>
<td>0.67</td>
<td>0.66</td>
<td>0.69</td>
<td>0.69</td>
<td>0.72</td>
<td>0.77</td>
<td>0.76</td>
</tr>
</tbody>
</table>
NAM (North American Mesoscale Forecast System) dataset (high noise dataset)
Kernel methods can perform well on extrapolation problems if the kernel is also learned from data.


Kernel Mode Decomposition and programmable/interpretable regression networks, O., Scovel, Yoo, 2019
arXiv:1907.08592
Which kernel do we pick?

- Deep learning approach

\[ f(x, \theta) \]

Ridge regression with a kernel learned from data

Problem

\[ \mathcal{X} \xrightarrow{f^\dagger} \mathcal{Y} \]

\( f^\dagger \): Unknown

Given \( f^\dagger(X) = Y \) with \( (X, Y) \in \mathcal{X}^N \times \mathcal{Y}^N \) approximate \( f^\dagger \)

\( \mathcal{X}, \mathcal{Y} \): Finite-dimensional Hilbert spaces

\( X := (X_1, \ldots, X_N) \in \mathcal{X}^N \)

\( f^\dagger(X) := (f^\dagger(X_1), \ldots, f^\dagger(X_N)) \in \mathcal{Y}^N \)

\( Y := (Y_1, \ldots, Y_N) \in \mathcal{Y}^N \)
Problem

\[ X \xrightarrow{f^\dagger} Y \]

\( f^\dagger \): Unknown

Given \( f^\dagger(X) = Y \) with \((X, Y) \in X^N \times Y^N\) approximate \( f^\dagger \)

Ridge regression solution

Approximate \( f^\dagger \) with minimizer of

\[
\min_f \lambda \| f \|_K^2 + \| f(X) - Y \|_{Y^N}^2
\]

\[
f(x) = K(x, X)(K(X, X) + \lambda I)^{-1}Y
\]

\( K : X \times X \to \mathcal{L}(Y) \)

\( \mathcal{L}(Y) \): Set of bounded linear operators on \( Y \).

\( K(X, X) \): \( N \times N \) block matrix with blocks \( K(X_i, X_j) \)

\( K(x, X) \): \( 1 \times N \) block vector with blocks \( K(x, X_i) \)

[Alvarez et Al, 2012]: Vector-valued kernels

[Kadri et Al, 2016]: Operator-valued kernels
Artificial neural network solution

Approximate $f^\dagger$ with

$$f = f_D \circ \cdots \circ f_1$$

\[ x = x_1 \quad f_1 \quad x_2 \quad \ldots \quad x_k \quad f_k \quad x_{k+1} \quad \ldots \quad x_{D+1} = y \]

\[ f_k(x) = a(W_k x + b_{k+1}) \]

\textbf{a}: Activation function / Elementwise nonlinearity

\[ \mathcal{L}(x_k, x_{k+1}): \text{Set of bounded linear operators from } x_k \text{ to } x_{k+1} \]

\[ W_k \in \mathcal{L}(x_k, x_{k+1}), \quad b_{k+1} \in x_{k+1} \text{ identified as minimizers of} \]

\[ \min_{W_k, b_k} \| f(X) - Y \|_\mathcal{Y}^2 \]

\[ \| Y \|_{\mathcal{Y}^N}^2 := \sum_{i=1}^N \| Y_i \|_{\mathcal{Y}}^2 \]
Residual neural network solution

Approximate $f^\dagger$ with

$$f = F_D \circ \cdots \circ F_1$$

[He et Al, 2016]

$$\begin{align*}
\chi &= \chi_1 \\
F_1 &\quad \chi_2 \\
&\quad \cdots \\
F_k &\quad \chi_k \\
&\quad \cdots \\
\chi_{D+1} &= \chi
\end{align*}$$

$$\begin{align*}
\chi_k + v_1 &\quad \chi_k \\
&\quad \cdots \\
\chi_k + v_{L_k} &\quad \chi_k \\
&\quad f_k \\
&\quad \chi_{k+1}
\end{align*}$$

$$F_k = f_k \circ (I + v^k_{L_k}) \circ \cdots \circ (I + v^k_1)$$

$$\begin{align*}
f_k : \chi_k &\rightarrow \chi_{k+1} \\
f_k(x) &= a(W_k x + b_{k+1})
\end{align*}$$

$$\begin{align*}
v^s_k : \chi_k &\rightarrow \chi_k \\
v^s_k(x) &= a(W^s_k x + b^s_k)
\end{align*}$$

$$\min_{W_k, b_k, W^s_k, b^s_k} \|f(X) - Y\|^2_{\mathcal{Y}^N}$$
Approximate $f^\dagger$ with

$$f^\dagger = f \circ \phi_L$$

$$\phi_L : \mathcal{X} \to \mathcal{X}$$

$$\phi_L = (I + v_L) \circ \cdots \circ (I + v_1)$$

$f : \mathcal{X} \to \mathcal{Y}$ and $v_s : \mathcal{X} \to \mathcal{X}$ identified as minimizers of

$$\min_{f,v_1,\ldots,v_L} \frac{\nu L}{2} \sum_{s=1}^L \|v_s\|_\Gamma^2 + \lambda \|f\|_K^2 + \|f \circ \phi_L(X) - Y\|^2_{\mathcal{Y}^N}$$

$$K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$$

$$\Gamma : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{X})$$

Particular case: ResNet block with L2 regularization on weights and biases!
Particular case

\[ \Gamma(x, x') = \varphi^T(x) \varphi(x') I_X \]

\[ K(x, x') = \varphi^T(x) \varphi(x') I_Y \]

\[ \varphi(x) = (a(x), 1) \quad \varphi : X \to X \oplus \mathbb{R} \]

\[ a(x): \text{Activation function} \quad a : X \to X \]

\[ f \circ \phi_L(x) = (\tilde{w} \varphi) \circ (I + w_L \varphi) \circ \cdots \circ (I + w_1 \varphi) \]

\[ \tilde{w} \in \mathcal{L}(X \oplus \mathbb{R}, Y) \text{ and } w_s \in \mathcal{L}(X \oplus \mathbb{R}, X) \text{ minimizers of } \]

\[ \min_{\tilde{w}, w_1, \ldots, w_L} \frac{\nu L}{2} \sum_{s=1}^{L} \|w_s\|^2_{\mathcal{L}(X \oplus \mathbb{R}, X)} + \lambda \|\tilde{w}\|^2_{\mathcal{L}(X \oplus \mathbb{R}, Y)} + \|f \circ \phi_L(X) - Y\|^2_{\mathcal{Y}_N} \]

This is one ResNet block with L2 regularization on weights and biases!
Approximate $f^\dagger$ with

$$f^\dagger = f \circ \phi_L$$

$\phi_L : \mathcal{X} \to \mathcal{X}$

$\phi_L = (I + v_L) \circ \cdots \circ (I + v_1)$

$f : \mathcal{X} \to \mathcal{Y}$ and $v_s : \mathcal{X} \to \mathcal{X}$ identified as minimizers of

$$\min_{f, v_1, \ldots, v_L} \frac{\nu L}{2} \sum_{s=1}^{L} \|v_s\|_{\Gamma}^2 + \lambda \|f\|_{K}^2 + \|f \circ \phi_L(X) - Y\|_{Y_N}^2$$

$K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$

$\Gamma : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{X})$
As $L \to \infty$, adherence values of $f \circ \phi_L(x)$ are

$$f \circ \phi^v(x)$$

$$\begin{aligned}
\dot{\phi}(x, t) &= v(\phi(x, t), t) \\
\phi(x, 0) &= x
\end{aligned}$$

$v : \mathcal{X} \times [0, 1] \to \mathcal{X}$ and $f : \mathcal{X} \to \mathcal{Y}$ are minimizers of

$$\min_{v,f} \frac{\nu}{2} \int_0^1 \|v(\cdot, t)\|_F^2 \, dt + \lambda \|f\|_K^2 + \|f \circ \phi^v(X, 1) - Y\|_{\mathcal{Y}_N}^2$$

**What kind of optimization problem is this?**

Looks like an image registration/computational anatomy variational problem
How to best align image $I$ and image $I'$?

[Grenander, Miller, 1998]: Computational anatomy

[Joshi, Miller, 2000], [Micheli, 2008], [Beg, Miller, Trouvé, Younes, 2005], [Dupuis, Grenander, Miller, 1998], [Vialard, Risser, Rueckert, Cotter, 2012].
\[
\min_{\nu} \lambda \int_0^1 \| \Delta v(\cdot, t) \|_{L^2([0,1]^2)}^2 \, dt + \| I(\phi^v(\cdot, 1)) - I' \|_{L^2([0,1]^2)}^2
\]

\[
\begin{cases}
\dot{\phi}(x, t) = v(\phi(x, t), t) \\
\phi(x, 0) = x
\end{cases}
\]
Image registration with landmarks

\[
\min_v \lambda \int_0^1 \| \Delta v \|_{L^2([0,1]^2)}^2 \, dt + \sum_i |\phi^v(X_i, 1) - Y_i|^2
\]

\[
\begin{align*}
\dot{\phi}(x, t) &= v(\phi(x, t), t) \\
\phi(x, 0) &= x
\end{align*}
\]

[Joshi, Miller, 2000]: Landmark matching
Image registration with landmark matching

$$\min_{\nu} \lambda \int_0^1 \| \Delta v \|_{L^2([0,1]^2)}^2 \, dt + \sum_i |\phi^v(X_i, 1) - Y_i|^2$$

\[
\begin{align*}
\dot{\phi}(x, t) &= v(\phi(x, t), t) \\
\phi(x, 0) &= x
\end{align*}
\]

Generalization

$$\min_{\nu, f} \frac{\nu}{2} \int_0^1 \| v(\cdot, t) \|_{L^2}^2 \, dt + \lambda \| f \|_K^2 + \| f \circ \phi^v(X, 1) - Y \|_{\mathcal{Y}_N}^2$$

$X_i, X_j \in \mathcal{X} = \mathbb{R}^{1024}$
### Image registration

<table>
<thead>
<tr>
<th><strong>Image registration</strong></th>
<th><strong>Idea registration</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image</strong></td>
<td></td>
</tr>
<tr>
<td>$I : [0, 1]^2 \to \mathbb{R}_+$</td>
<td></td>
</tr>
<tr>
<td>$I' : [0, 1]^2 \to \mathbb{R}_+$</td>
<td></td>
</tr>
<tr>
<td><strong>$X_i, Y_i$</strong></td>
<td></td>
</tr>
<tr>
<td>Landmark/material points</td>
<td></td>
</tr>
<tr>
<td>$X_i \in [0, 1]^2$, $Y_i \in [0, 1]^2$</td>
<td></td>
</tr>
<tr>
<td><strong>$\phi$</strong></td>
<td></td>
</tr>
<tr>
<td>Deforms $[0, 1]^2$ and $I : [0, 1]^2 \to \mathbb{R}_+$</td>
<td></td>
</tr>
</tbody>
</table>

### Idea registration

| **Idea** |
| $I : \mathcal{X} \to \mathcal{F}$ |
| $I' : \mathcal{Y} \to \mathcal{F}$ |
| **Data points** |
| $X_i \in \mathcal{X}$, $Y_i \in \mathcal{Y}$ |
| **Deforms** |
| Deforms $\mathcal{X}$ and $I : \mathcal{X} \to \mathcal{F}$ |
Idea registration is ridge regression with a warped kernel

\[(IR) \quad \min_{v,f} \frac{\nu}{2} \int_0^1 \|v(\cdot, t)\|_1^2 \, dt + \lambda \|f\|_K^2 + \|f \circ \phi^v(X, 1) - Y\|_{\mathcal{Y}_N}^2\]

\[f^{IR} = f \circ \phi^v(x)\]

\[\begin{align*}
\phi(x, t) &= v(\phi(x, t), t) \\
\phi(x, 0) &= x
\end{align*}\]

\[(RR) \quad \min_f \lambda \|f\|_{K^v}^2 + \|f(X) - Y\|_{\mathcal{Y}_N}^2\]

\[K^v(x, x') = K(\phi^v(x, 1), \phi^v(x', 1))\]

\[f^{RR} = f\]

Theorem \[f^{IR} = f^{RR}\]

See also Diffeomorphic learning: [Younes, 2019], [Rousseau, Fablet, 2018], [Zammit-Mangion et al, 2019], [O., Yoo, 2018]
Idea registration is Gaussian Process Regression with a prior learned from data

\[
\begin{align*}
\text{(IR)} \quad & \min_v, f \frac{\nu}{2} \int_0^1 \|v(\cdot, t)\|^2 dt + \lambda \|f\|_K^2 + \|f \circ \phi^v(X, 1) - Y\|_{\mathcal{Y}_N}^2 \\
\Rightarrow \quad & f^{IR} = f \circ \phi^v(x) \\
\text{(RR)} \quad & \min_f \lambda \|f\|_K^2 + \|f(X) - Y\|_{\mathcal{Y}_N}^2 \\
\Rightarrow \quad & f^{RR} = f \\
\end{align*}
\]

\[
\begin{align*}
\dot{\phi}(x, t) &= v(\phi(x, t), t) \\
\phi(x, 0) &= x
\end{align*}
\]

\[
\begin{align*}
K^v(x, x') &= K(\phi^v(x, 1), \phi^v(x', 1))
\end{align*}
\]

**Theorem**

\[
\begin{align*}
f^{IR} &= f^{RR} \\
f^{IR}(x) &= \mathbb{E}_{\xi \sim \mathcal{N}(0, K^v)} [\xi(x) \mid \xi(X) = Y + Z] \\
&\quad \text{for } Z \sim \mathcal{N}(0, \lambda I)
\end{align*}
\]
Brittleness of Bayesian inference implies the brittleness of ANNs.

\[
f^{\text{IR}}(x) = \mathbb{E}_{\xi \sim \mathcal{N}(0, K^\nu)} \left[ \xi(x) \mid \xi(X) = Y + Z \right]
\]

\[Z \sim \mathcal{N}(0, \lambda I)\]

[O., Scovel, Sullivan, Apr 2013]: Bayesian inference is brittle w.r. to perturbations of the prior

[McKerns, SyiPy, June 2013]: Bayesian brittleness can lead machine learning algorithms to be increasingly confident in incorrect solutions

https://youtu.be/o-nwSnLC6DU?t=74

Brittleness of Bayesian inference implies the brittleness of ANNs.
[Biggio et al, 2012-2018], [Moisejevs et al, 2019]: ANNs are brittle to data poisoning

[Szegedy et al, Dec 2013]: ANNs are brittle to adversarial noise

“pig”

+ 0.005 x

“airliner”

[Madry, Schmidt, 2018]
How do we fix it?

Training without regularization

$$\min_{v,f} \frac{v}{2} \int_0^1 \|v(\cdot, t)\|_V^2 \, dt + \lambda \|f\|_K^2 + \|f \circ \phi^v(X, 1) - Y\|_Y^2$$

$$\begin{cases}
\dot{\phi}(x, t) = v(\phi(x, t), t) \\
\phi(x, 0) = x
\end{cases}$$

Training with regularization

$$\Gamma \quad \leftrightarrow \quad \Gamma + rI$$

$$K \quad \leftrightarrow \quad K + \rho I$$

$$\min_{v, f, q, Y'} \frac{v}{2} \int_0^1 \|v(\cdot, t)\|_V^2 \, dt + \frac{1}{r} \int_0^1 \|\dot{q} - v(q(t))\|_X^2 \, dt$$

$$+ \lambda \|f\|_K^2 + \frac{\lambda}{\rho} \|f(q(1)) - Y'\|_Y^2 + \|Y' - Y\|_Y^2$$

$$q : [0, 1] \rightarrow X^N \quad \quad q(0) = X$$

Equivalent to metamorphosis in image registration:

[Micheli, 2008], [Niethammer et al, 2011], [Charon, Charlier, Trouvé, 2018]
Kernel methods

Kernel:

Feature map:

RKHS space:

GP:

Kernel representation

Feature map representation

Idea registration

Bayesian MAP estimation
Theorem

\( f \circ \phi^v(\cdot, 1) \) is a MAP estimator of \( \xi \circ \phi \sqrt{\frac{\lambda}{\nu}} \zeta(\cdot, 1) \) given the information

\[ \xi \circ \phi \sqrt{\frac{\lambda}{\nu}} \zeta(X, 1) + \sqrt{\lambda} Z = Y \]

\( \xi \sim \mathcal{N}(0, K) \)

\( \phi^\zeta(x, t) \): solution of

\[
\begin{align*}
\dot{z} &= \zeta(z, t) \\
z(0) &= x
\end{align*}
\]

\( \zeta \) centered GP defined by norm \( \int_0^1 \| v(\cdot, t) \|_1^2 \, dt \) (independent from \( \xi \))

\( Z = (Z_1, \ldots, Z_N) \): centered random Gaussian vector, independent from \( \zeta \) and \( \xi \), with i.i.d. \( \mathcal{N}(0, I_\gamma) \) entries

See also: Deep Gaussian processes [Damianou, Lawrence, 2013] and Brownian flow of diffeomorphisms [Kunita, 1997], [Baxendale., 1984], [Dupuis, Grenander, Miller, 1998].
Idea registration

\[
\min_{\nu,f} \frac{\nu}{2} \int_0^1 \|v(\cdot, t)\|_1^2 \, dt + \lambda \|f\|_K^2 + \|f \circ \phi^\nu(X, 1) - Y\|_{\mathcal{V}_N}^2
\]

\[
\begin{align*}
\dot{\phi}(x, t) &= u(\phi(x, t), t) \\
\phi(x, 0) &= x
\end{align*}
\]

Theorem

\[
v(x, t) = \Gamma(x, q) \Gamma(q, q)^{-1} \dot{q}
\]

\(q\) position variable in \(\mathcal{X}^N\) started from \(q(0) = X\), minimizing the least action principle

\[
\min_{f,q} \frac{\nu}{2} \int_0^1 \dot{q}^T \Gamma(q, q)^{-1} \dot{q} + \lambda \|f\|_K^2 + \|f(q(1)) - Y\|_{\mathcal{V}_N}^2
\]
Idea registration

$$\min_{v, f} \frac{\nu}{2} \int_0^1 \|v(\cdot, t)\|_I^2 \, dt + \lambda \|f\|_K^2 + \|f \circ \phi^v(X, 1) - Y\|_N^2$$

$$\begin{cases}
\dot{\phi}(x, t) = v(\phi(x, t), t) \\
\phi(x, 0) = x
\end{cases}$$

Corollary

$$v(x, t) = \Gamma(x, q)p$$

$$p = \Gamma(q, q)^{-1} \dot{q}$$

(q, p) position and momentum variables in $X^N$ started from $q(0) = X$

$$\begin{cases}
\dot{q}_i = \partial_{p_i} \mathcal{H}(q, p) \\
\dot{p}_i = -\partial_{q_i} \mathcal{H}(q, p)
\end{cases}$$

$$\mathcal{H}(q, p) = \frac{1}{2} p^T \Gamma(q, q)p$$

$v, f$ uniquely determined by $p(0)$

$$\|v(\cdot, t)\|_I^2$$ constant over $t \in [0, 1]$

See also ODE interpretations of ResNets: [E, 2017], [Haber, Ruthotto, 2017], [Chen, Rubanova, Bettencourt, Duvenaud, 2018], [Chang et al 2018]
Idea registration/Resnet learn warping kernels of the form

\[ K^v(x, x') = K(\phi^v(x, 1), \phi^v(x', 1)) \]

\( K \): Base kernel

\( \phi^v \): Warping of the input space

\[ \begin{align*}
\dot{\phi}(x, t) &= v(\phi(x, t), t) \\
\phi(x, 0) &= x
\end{align*} \]

\[ v(x, t) = \Gamma(x, q)\Gamma(q, q)^{-1} \dot{q} \]

\( q \) position variable in \( \mathcal{X}^N \) started from \( q(0) = X \), minimizing the least action principle

\[
\min_{f, q} \frac{\nu}{2} \int_0^1 \dot{q}^T \Gamma(q, q)^{-1} \dot{q} + \lambda \| f \|^2_K + \| f(q(1)) - Y \|^2_{\mathcal{Y}^N}
\]
Replace MAP estimation (idea registration) with cross validation to learn the warping (kernel flows, no need for backpropagation)

\[ K^v(x, x') = K(\phi^v(x, 1), \phi^v(x', 1)) \]  

\( K \): Base kernel

\( \phi^v \): Warping of the input space

\[
\begin{align*}
\dot{\phi}(x, t) &= v(\phi(x, t), t) \\
\phi(x, 0) &= x
\end{align*}
\]

\( v(x, t) = \Gamma(x, q)\Gamma(q, q)^{-1}\dot{q} \)

\( q \): position variables in \( \mathcal{X}^N \) started from \( q(0) = X \)

\[
\dot{q} = -\nabla \rho(q)
\]

\( \rho \): Kernel flow loss
The effective dynamical system

\[ q_i(t) := \phi(X_i, t) \]

\( Y_i \): Label of \( X_i \)

\( u: \) interpolate \( ((q_i, Y_i))_{1 \leq i \leq N} \) with \( K \)
The effective dynamical system

\[ \phi(X_i, t) \]

\( \pi(1), \ldots, \pi(N/2) \): random selection of \( N/2 \) points out of \( N \) colored yellow

\( w \): interpolate \( (q_{\pi(i)}, Y_{\pi(i)})_{1 \leq i \leq \frac{N}{2}} \) with \( K \)

\[
\rho(q) = \mathbb{E}_\pi \left[ \frac{||u-w||^2}{||u||^2_K} \right]
\]
Application: Swiss Roll Cheesecake

\[ N = 100 \text{ data points } x_i \in \mathbb{R}^2 \]
\[ Y_i = +1 \text{ if point at } X_i \text{ is red} \]
\[ Y_i = -1 \text{ if point at } X_i \text{ is blue} \]

Objective:
Visualize \( t \rightarrow q(t) \)
Application to Fashion-MNIST
Average distance vs $t$

Fashion- MNIST
Composed idea registration

$X_i, X_j \in \mathbb{R}^{1024}$

Composed idea registration blocks

- time discretization
- ANNs and ResNets
- Projected equivariant kernels for $K$ and $\Gamma$
- CNNs and their generalization to arbitrary groups of symmetries

$\mathcal{X} = \mathbb{R}^{1024}$

$\mathcal{V} = \mathbb{R}^{100}$
Related work

• Deep kernel learning. [Wilson et al, 2016], [Bohn, Rieger, Griebel. 2019]

• Computational anatomy and image registration. [Joshi, Miller, 2000],
  [Micheli, 2008], [Beg, Miller, Trouvé, Younes, 2005], [Dupuis, Grenander,
  Miller, 1998], [Vialard, Risser, Rueckert, Cotter, 2012].

• Statistical numerical approximation. [O. 2015, 2017], [O., Scovel, 2019],
  [O., Scovel, Schäfer, 2019], [Raissi, Perdikaris, Karniadakis, 2019], [Cock-
 ayne, Oates, Sullivan, Girolami, 2019], [Hennig, Osborne, Girolami, 2015]

• ODE interpretations of ResNets. [E, 2017], [Haber, Ruthotto, 2017],
  [Chen, Rubanova, Bettencourt, Duvenaud, 2018], [Chang, Meng, Haber,
  Ruthotto, Begert, Holtham, 2018]

• Warping kernels [O., Zhang, 2005], [Sampson, Guttorp, 1992], [Perrin,
  Monestiez, 1999], [Schmidt, O’Hagan, 2003]

• Kernel Flows [O., Yoo, 2019], [Chen, O., Stuart, 2020], [Hamzi, O., 2020],
  [Yoo, O., 2020]

• Deep Gaussian processes. [Damianou, Lawrence, 2013]

• Brownian flow of diffeomorphisms [Kunita, 1997], [Baxendale., 1984]

• Equivariant kernels [Reisert, Burkhardt, 2007]

• Operator valued kernels [Kadri et al, 2016]

• Diffeomorphic learning: [Younes, 2019], [Rousseau, Fablet, 2018], [Zammit-
  Mangion et al, 2019]

This work

• Do ideas have shape? Plato’s theory of forms as the continuous limit of
Why is our main question (which kernel do we pick?) relevant to numerical approximation?

**ANNs**


**Kernel methods**

“Statistical Numerical Approximation”, O., Scovel, Schäfer, Notices of the AMS, 2019

Solving and learning nonlinear PDEs with Gaussian Processes. Chen, Hosseini, O., Stuart, 2021

- Provably convergent.
- Inherits the state of the art complexity vs accuracy guarantees of linear solvers for dense kernel matrices.
- Interpretable and amenable to numerical analysis.
Most numerical approximation methods are kernel interpolation methods.


Book: Cambridge University Press, O., Scovel, 2019

**Cardinal splines**  [Schoenberg, 1973]

Cardinal spline interpolants are optimal recovery (kernel interpolants) splines
Polyharmonic splines

\[
\begin{cases}
-\Delta f^\dagger = g, & x \in \Omega, \quad g \in L^2(\Omega) \\
 f^\dagger = 0, & x \in \partial\Omega,
\end{cases}
\]

\[\Omega \subset \mathbb{R}^d\]
\[d \leq 3\]

Problem: Given \(f^\dagger(X)\) recover \(f^\dagger\)

\[
\begin{cases}
\text{Minimize} & \int_\Omega |\Delta f|^2 \\
\text{subject to} & f(X) = Y
\end{cases}
\quad \|f^\dagger - f\|_{L^2(\Omega)} \lesssim N^{-\frac{2}{d}} \|g\|_{L^2}
\]
The convergence can be arbitrarily bad if the kernel is not adapted

\[
\begin{align*}
& - \text{div}(a \nabla f^\dagger) = g, \quad x \in \Omega, \\
& f^\dagger = 0, \quad x \in \partial \Omega,
\end{align*}
\]

\[g \in L^2(\Omega)\]

\[\Omega \subset \mathbb{R}^d\]

\[d \leq 3\]

Minimize

\[
\int_{\Omega} |\Delta f|^2
\]

subject to

\[f(X) = Y\]

\[\|f^\dagger - f\|_{L^2(\Omega)} \geq \chi(N)\]

The convergence of \(\chi(N)\) towards zero can be arbitrarily slow

[Babuška, Osborn, 2000]: Can a finite element method perform arbitrarily badly?
PDE adapted kernel

\[ \begin{aligned}
- \text{div}(a \nabla f^\dagger) &= g, \quad x \in \Omega, \\
\quad f^\dagger &= 0, \quad x \in \partial \Omega,
\end{aligned} \]

\[ g \in L^2(\Omega) \]

\[ \Omega \subset \mathbb{R}^d \quad d \leq 3 \]

Minimize \[ \int_{\Omega} \left| \text{div}(a \nabla f) \right|^2 \]
subject to \[ f(X) = Y \]

\[ \| f^\dagger - f \|_{L^2(\Omega)} \lesssim N^{-\frac{2}{d}} \| g \|_{L^2} \]

[O., Berlyand, Zhang, 2014]: Rough polyharmonic splines
[O., 2014]: Bayesian Numerical Homogenization
[O., 2015], [O., Zhang, 2016], [O., Scovel, 2019], [Schäfer, Sullivan, O., 2017]: Gamblets
[Schäfer, Katzfuss and O., 2020]
Generalization to non-linear PDE

\[
\begin{cases}
-\Delta f^\dagger + \tau(f^\dagger) = g, & x \in \Omega, \\
f^\dagger = b, & x \in \partial\Omega,
\end{cases}
\]

Minimize \( \|f\|^2_K \)
subject to \(-\Delta f(X_i) + \tau(f(X_i)) = g(X_i), \ X_i \in \Omega, \)
and \( f(X_i) = b(X_i), \ X_i \in \partial\Omega, \)

Theorem \( f \rightarrow f^\dagger, \) as fill distance (in \( \bar{\Omega} \)) of collocation points goes to 0, provided that
\[
f^\dagger \in \mathcal{H} \subseteq H^s(\Omega) \quad \text{with} \ s > 2 + d/2
\]
Minimize \( \| f \|_K^2 \)

subject to \( -\Delta f(X_i) + \tau(f(X_i)) = g(X_i), \ X_i \in \Omega, \)

and \( f(X_i) = b(X_i), \ X_i \in \partial\Omega, \)

\[
\min_{z^{(1)}, z^{(2)}} \left\{ \begin{array}{l}
\min f \| f \|_K^2 \\
\text{s.t. } f(X_i) = z_i^{(1)} \text{ and } -\Delta f(X_i) = z_i^{(2)} \\
z_i^{(2)} + \tau(z_i^{(1)}) = g(X_i) \text{ for } X_i \in \Omega \\
z_i^{(1)} = b(X_i) \text{ for } X_i \in \partial\Omega
\end{array} \right\}
\]

\[
z = (z^{(1)}, z^{(2)})
\]

\[
\phi = (\phi^{(1)}, \phi^{(2)})
\]

\[
\phi_i^{(1)} = \delta X_i
\]

\[
\phi_i^{(2)} = \delta X_i \circ \Delta
\]

\[
f(x) = K(x, \phi) K(\phi, \phi)^{-1} z
\]

\[
\text{Near linear complexity with [Schäfer, Katzfuss and O., 2020]}
\]
\[ \partial_t u + u \partial_s u - \nu \partial_s^2 u = 0, \quad \forall (s, t) \in [-1, 1] \times [0, \infty), \]
\[ u(s, 0) = -\sin(\pi x), \]
\[ u(-1, t) = u(1, t) = 0. \]

\[ K((x, t), (x', t')) = \exp\left(-\alpha |x - x'|^2 - \beta |t - t'|^2\right) \]

<table>
<thead>
<tr>
<th>N</th>
<th>64</th>
<th>256</th>
<th>1024</th>
<th>4096</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L^2$ error</td>
<td>2.7797e+00</td>
<td>2.1015e-02</td>
<td>5.6348e-04</td>
<td>8.5275e-05</td>
</tr>
</tbody>
</table>

At time $t = 0.2$:

At time $t = 0.5$:

At time $t = 0.8$:
\[
\left\{ \begin{aligned}
\| \nabla u(x) \|^2 &= f(x)^2 + \epsilon \Delta u(x), \quad \forall x \in \Omega, \\
u(x) &= 0, \quad \forall x \in \partial \Omega,
\end{aligned} \right.
\]

\[K(x, x') = \exp \left( -\alpha |x - x'|^2 \right)\]

<table>
<thead>
<tr>
<th>(N)</th>
<th>1200</th>
<th>1800</th>
<th>2400</th>
<th>3000</th>
</tr>
</thead>
<tbody>
<tr>
<td>(L^2) error</td>
<td>3.7942e-04</td>
<td>1.3721e-04</td>
<td>1.2606e-04</td>
<td>1.1025e-04</td>
</tr>
<tr>
<td>(L^\infty) error</td>
<td>5.5768e-03</td>
<td>1.4820e-03</td>
<td>1.3982e-03</td>
<td>9.5978e-04</td>
</tr>
</tbody>
</table>

---

**Collocation points**

**Loss function history**

**Contour of errors**
Inverse Problem

\[ \begin{align*}
- \text{div} \left( \exp(a) \nabla u \right)(x) &= f(x), \quad x \in \Omega, \\
\quad u(x) &= 0, \quad x \in \partial\Omega.
\end{align*} \]

\( a, u \): Unknown. \( u \) observed at pink points.
Problem: Recover \( a \) and \( u \).

Minimize \( \| u \|_K^2 + \| a \|_I^2 \)

subject to \( - \text{div} \left( \exp(a) \nabla u \right) (X_i) = f(X_i), \quad X_i \in \Omega \),

and \( u(X_i) = Y_i, \quad (X_i, Y_i) \) is data point,

and \( u(X_i) = 0, \quad X_i \in \partial\Omega \),
Inverse Problem

\[
\begin{cases}
- \text{div} \left( \exp(a) \nabla u \right)(x) = f(x), \quad x \in \Omega, \\
u(x) = 0, \quad x \in \partial \Omega.
\end{cases}
\]

\(a, u\): Unknown. \(u\) observed at pink points.
Problem: Recover \(a\) and \(u\).
Main messages

It is all about learning kernels.

ANNs are essentially discretized solvers for a generalization of image registration/computational anatomy variational problems.

Image registration

Generalization

Idea registration

Thank you