Multiscale Computation and Parameter Learning for Kernels from PDEs

— Two Provable Examples —

Yifan Chen, Caltech

CMX Student Seminar

November 20, 2020

- **The Mathematical Model**:

$$\mathcal{L}_\theta u = f \qquad \text{(Partial Differential Equation)}$$

  - $\mathcal{L}_\theta$: *linear* operators, encoding physics / interaction laws
  - θ: parameters (a family of parameterized laws)
  - u, f: functions (to be specified / observed / computed / predicted)

  e.g. $\mathcal{L}_\theta u = -\nabla \cdot (a \nabla u) = f$ in $\Omega = [0,1]^d$ and $u \in H_0^1(\Omega)$; here, $\theta = a(\cdot)$
  ($\mathcal{L}_\theta^{-1}$: Kernel operator / Green's function)

- Problems of interests:

Computation: given $\mathcal{L}_\theta$ and f, compute u

– Solve the PDE numerically

Learning: predict $u(x), x \in \Omega$ from some $u(x_i)$ for $x_i \subset \mathcal{X} \subset \Omega$

– Even more: learn the physics underlying the data, i.e., learn θ

Sketch of contributions I

- For **Computation** Problems:
  - Model: $\mathcal{L}_\theta u = -\nabla \cdot (a \nabla u) - k^2 V u = f$ for $0 \leq a, V \in L^\infty(\Omega)$, $k \in \mathbb{R}_+$ (Helmholtz's equation + D/N/R boundary conditions)

Our result: on a mesh of lengthscale $H = O(1/k)$, u can be computed by

$$u = \underbrace{\sum_{i \in I_1} c_i \psi_i^{(1)}}_{\text{(I)}} + \underbrace{\sum_{i \in I_2} d_i \psi_i^{(2)}}_{\text{(II)}} + \underbrace{\sum_{i \in I_3} \psi_i^{(3)}}_{\text{(III)}} + O\left(\exp\left(-m^{\frac{1}{d+1}-\epsilon}\right)\right) \quad \text{(Energy norm)}$$

where: $\psi_i^{(1)}, \psi_i^{(2)}, \psi_i^{(3)}$ all have *local* support of size H

- $\psi_i^{(1)}$ obtained by *local* SVD of $\mathcal{L}_\theta$ $\qquad \#I_1 = O(m/H^d)$
- $\psi_i^{(2)}, \psi_i^{(3)}$ obtained by solving *local* $\mathcal{L}_\theta u = f$ $\qquad \#I_2, \#I_3 = O(1/H^d)$
- $c_i, d_i \in \mathbb{R}$ obtained by Galerkin's methods with basis functions $\psi_i^{(1)}, \psi_i^{(2)}$
- (II),(III)$= O(H)$ (Energy norm)

A data-adaptive coarse-fine scale decomposition

- Y. Chen, T. Y. Hou, and Y. Wang, Exponential convergence for multiscale linear elliptic pdes via adaptive edge basis functions, *arXiv:2007.07418*, 2020.

Sketch of contributions II

- For **Learning** Problems:
  - General approach: Gaussian Process Regression + Kernel selection
  - Selection algorithms: *probabilistic* Empirical Bayes (EB) / *approximation-theoretic* Kernel Flow (KF)
  - Model: $\mathcal{L}_\theta = (-\Delta)^{\frac{s}{2}}$ on a torus, and f is the white noise

Our result: u and s can be learned provably from ($\mathcal{X}$: a uniform lattice)

$$\{u(x_i) \quad \text{for} \quad x_i \subset \mathcal{X} \subset \Omega\}$$

Moreover,

- EB and KF have different consistency result in the large data limit, yielding different selection bias
- EB and KF behave differently regarding model misspecification

- Y. Chen, H. Owhadi, and A. M. Stuart, Consistency of empirical bayes and kernel flow for hierarchical parameter estimation, *arXiv:2005.11375*, 2020.

Roadmap

1. Multiscale Computation with Exponential Convergence
   - Solve PDE as an approximation problem
   - Coarse-fine scale decomposition of solution space
   - Exponential efficiency of the coarse part
   - Numerical experiments
   - Related works
   - Take-aways

2. Parameter Learning with Provable Guarantees
   - Learning a kernel
   - Bayes' approach
   - Kernel Flow approach
   - Consistency as # of data $\to \infty$, and beyond
   - Take-aways

Roadmap

1. Multiscale Computation with Exponential Convergence
   - Solve PDE as an approximation problem
   - Coarse-fine scale decomposition of solution space
   - Exponential efficiency of the coarse part
   - Numerical experiments
   - Related works
   - Take-aways

2. Parameter Learning with Provable Guarantees
   - Learning a kernel
   - Bayes' approach
   - Kernel Flow approach
   - Consistency as # of data $\to \infty$, and beyond
   - Take-aways

Focus on Elliptic PDEs

- **Formulation**: $\mathcal{L}_\theta = -\nabla \cdot (a\nabla \cdot)$; thus $\theta = a(\cdot) \in L^\infty(\Omega)$

$$\begin{cases} -\nabla \cdot (a\nabla u) = f, & \text{in } \Omega \\ u = 0, & \text{on } \partial\Omega\,. \end{cases}$$

  $\Omega = [0,1]^2$ and $u \in H_0^1(\Omega), f \in L^2(\Omega)$

- **Galerkin methods**: choose a finite-dim space $V_H \subset H_0^1(\Omega)$

$$\text{Find } u_H \in V_H \quad \text{such that} \quad \int_\Omega a\nabla u_H \cdot \nabla v = \int_\Omega fv \quad \text{for any } v \in V_H\,.$$

  Optimality: (notation $\|u\|_{H_a^1(\Omega)} := \int_\Omega a|\nabla u|^2$)

$$\|u - u_H\|_{H_a^1(\Omega)} = \inf_{v \in V_H} \|u - v\|_{H_a^1(\Omega)}\,.$$

V_H needs to approximate the solution space well in the $H_a^1(\Omega)$ norm

Roadmap

1. Multiscale Computation with Exponential Convergence
   - Solve PDE as an approximation problem
   - Coarse-fine scale decomposition of solution space
   - Exponential efficiency of the coarse part
   - Numerical experiments
   - Related works
   - Take-aways

2. Parameter Learning with Provable Guarantees
   - Learning a kernel
   - Bayes' approach
   - Kernel Flow approach
   - Consistency as # of data $\to \infty$, and beyond
   - Take-aways

Explore the solution space

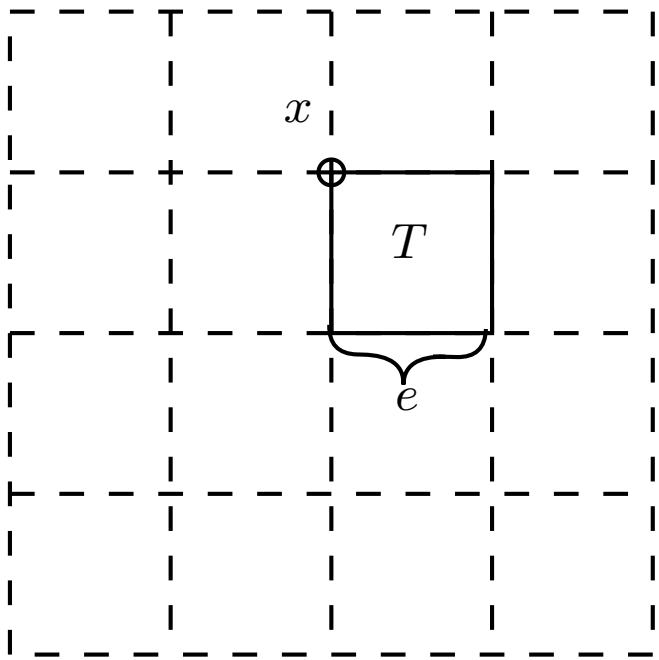
- **Mesh structure**:
  nodes, edges and elements

- **Split** the solution locally:
  in each T, $u = u_T^{\mathsf{h}} + u_T^{\mathsf{b}}$

$$\begin{cases} -\nabla \cdot (a \nabla u_T^{\mathsf{h}}) = 0, & \text{in } T \\ u_T^{\mathsf{h}} = u, & \text{on } \partial T\,, \end{cases}$$

$$\begin{cases} -\nabla \cdot (a \nabla u_T^{\mathsf{b}}) = f, & \text{in } T \\ u_T^{\mathsf{b}} = 0, & \text{on } \partial T\,. \end{cases}$$

- **Merge**: $u^{\mathsf{h}}(x) = u_T^{\mathsf{h}}(x)$ and $u^{\mathsf{b}}(x) = u_T^{\mathsf{b}}(x)$ when $x \in T$ for each T



$x \in \mathcal{N}_H, e \in \mathcal{E}_H, T \in \mathcal{T}_H$

Coarse-fine scale decomposition

- **Decomposition**: $u = u^{\mathsf{h}} + u^{\mathsf{b}} \in V^{\mathsf{h}} \oplus_a V^{\mathsf{b}}$

  $V^{\mathsf{h}} = \{v \in H_0^1(\Omega) : -\nabla \cdot (a\nabla v) = 0 \text{ in every } T \in \mathcal{T}_H\}$ (*harmonic part*)

  $V^{\mathsf{b}} = \{v \in H_0^1(\Omega) : v = 0 \text{ on } \partial T, \text{ for every } T \in \mathcal{T}_H\}$ (*bubble part*)

$$\boxed{H_0^1(\Omega) = V^{\mathsf{h}} \oplus_a V^{\mathsf{b}}}$$

- *Bubble part* is *local* and *small*
  - *local*: $u^{\mathsf{b}} = \sum_{i \in I_3} \psi_i^{(3)}$ (term (III))

    each $\psi_i^{(3)}$ solves an elliptic equation inside each T
  - *small*: elliptic estimate

$$\|u^{\mathsf{b}}\|_{H_a^1(\Omega)} \leq CH\|f\|_{L^2(\Omega)}$$

  i.e. u^{b} oscillates at a frequency large than $O(1/H)$

Bubble part *is* the *fine scale* part

Roadmap

1. Multiscale Computation with Exponential Convergence
   - Solve PDE as an approximation problem
   - Coarse-fine scale decomposition of solution space
   - Exponential efficiency of the coarse part
   - Numerical experiments
   - Related works
   - Take-aways

2. Parameter Learning with Provable Guarantees
   - Learning a kernel
   - Bayes' approach
   - Kernel Flow approach
   - Consistency as # of data $\to \infty$, and beyond
   - Take-aways

For the *a-harmonic* part u^{h}

- **Idea**: choose $V_H \subset V^{\mathsf{h}}$ in Galerkin's method, yielding

$$\|u^{\mathsf{h}} - u_H\|_{H^1_a(\Omega)} = \inf_{v \in V_H} \|u^{\mathsf{h}} - v\|_{H^1_a(\Omega)}$$

(Recall: a-orthogonality between V^{h} and V^{b})

Galerkin's solution u_H now *only* approximates the *a-harmonic* part

- $V^{\mathsf{h}} = \{v \in H^1_0(\Omega) : -\nabla \cdot (a\nabla v) = 0 \text{ in every } T \in \mathcal{T}_H\}$
only depends on values of v on edges

Observation: V^{h} is isomorphic to an edge space

Find basis functions to approximate edge values

- **Edge function**: $u^{\mathsf{h}} : \Omega \to \mathbb{R}$ restricted to edges: $\tilde{u}^{\mathsf{h}} : E_H \to \mathbb{R}$

Task: find edge basis functions to approximate $\tilde{u}^{\mathsf{h}}$

- **Localization to each edge**: $(\tilde{u}^{\mathsf{h}} - I_H \tilde{u}^{\mathsf{h}})|_e$ vanishes at boundaries where, I_H is nodal interpolation operator, e.g., by linear tent functions

Task then: find edge basis functions to approximate $(\tilde{u}^{\mathsf{h}} - I_H \tilde{u}^{\mathsf{h}})|_e$ for each e

Which norm for approximation?

- **The $\mathcal{H}^{1/2}(e)$ norm**: (connect back to energy norms)

$$\|\tilde{\psi}\|^2_{\mathcal{H}^{1/2}(e)} := \int_\Omega a|\nabla\psi|^2$$

  where, ψ is the a-harmonic extension of $\tilde{\psi}$ to neighboring elements

- **Edge Coupling**: from local to global

If on each edge, there is $\tilde{v}_e$ such that the *local* error satisfies

$$\|\tilde{u}^{\mathsf{h}} - I_H\tilde{u}^{\mathsf{h}} - \tilde{v}_e\|_{\mathcal{H}^{1/2}(e)} \leq \epsilon_e$$

then the *global* error satisfies

$$\|u^{\mathsf{h}} - I_H u^{\mathsf{h}} - \sum_{e\in\mathcal{E}_H} v_e\|^2_{H^1_a(\Omega)} \leq \sum_{e\in\mathcal{E}_H}\epsilon_e^2$$

Task now: find basis functions for v_e

Explore $u^{\mathsf{h}} - I_H u^{\mathsf{h}}$ on each e

- **Oversampling**: $e \subset \omega_e$

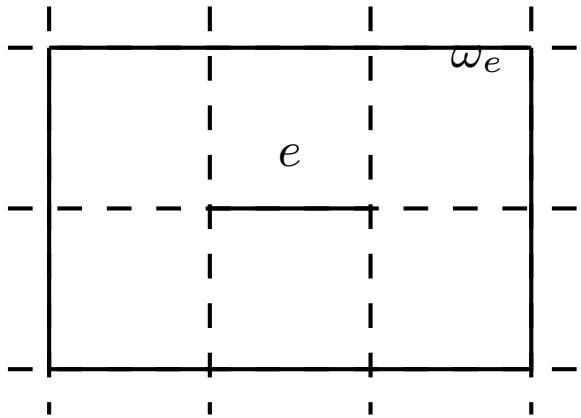
$$\text{on } e: \ u^{\mathsf{h}} - I_H u^{\mathsf{h}} = (u^{\mathsf{h}}_{\omega_e} - I_H u^{\mathsf{h}}_{\omega_e}) + (u^{\mathsf{b}}_{\omega_e} - I_H u^{\mathsf{b}}_{\omega_e})$$

$u^{\mathsf{h}}_{\omega_e}, u^{\mathsf{b}}_{\omega_e}$: oversampling harmonic / bubble part

Recall the definition:

$$\begin{cases} -\nabla \cdot (a \nabla u^{\mathsf{h}}_{\omega_e}) = 0, & \text{in } \omega_e \\ u^{\mathsf{h}}_{\omega_e} = u, & \text{on } \partial\omega_e \,, \end{cases}$$

$$\begin{cases} -\nabla \cdot (a \nabla u^{\mathsf{b}}_{\omega_e}) = f, & \text{in } \omega_e \\ u^{\mathsf{b}}_{\omega_e} = 0, & \text{on } \partial\omega_e \,. \end{cases}$$



interior edge

Why write in *this* form?

— *restrictions* of harmonic functions are of *low* complexity!

Low complexity: restrictions of a-harmonic functions

Theorem (Y. Chen, T.Y. Hou, Y. Wang, 2020)

For any a-harmonic functions v in ω_e and any $\epsilon > 0$, there exists an $N_\epsilon > 0$, such that for all $m > N_\epsilon$, we can find an $(m-1)$ dimensional space $W_e^m = \text{span}\,\{\tilde{v}_e^k\}_{k=1}^{m-1}$ so that

$$\min_{\tilde{v}_e \in W_e^m} \|v - I_H v - \tilde{v}_e\|_{\mathcal{H}^{1/2}(e)} \leq C \exp\left(-m^{(\frac{1}{d+1}-\epsilon)}\right) \|v\|_{H_a^1(\omega_e)}$$

- W_e^m obtained by the left singular vectors of the operator

$$R_e v = v - I_H v$$

from: space of a-harmonic functions (with energy norm) in ω_e
to: space $\mathcal{H}^{1/2}(e)$

- Proof technique combines [Babuska, Lipton 2011] and C^α estimates

Summary of approximations:

- $u = u^{\mathsf{h}} + \overbrace{u^{\mathsf{b}}}^{\text{(III) local-n-small}}$ (harmonic-bubble splitting)

- $u^{\mathsf{h}} = \overbrace{(u^{\mathsf{h}} - I_H u^{\mathsf{h}})}^{\text{localized to each edge}} + \overbrace{I_H u^{\mathsf{h}}}^{\text{basis functions in (I)}}$ (interpolation part)

- $(u^{\mathsf{h}} - I_H u^{\mathsf{h}})|_e = \overbrace{(u^{\mathsf{h}}_{\omega_e} - I_H u^{\mathsf{h}}_{\omega_e})|_e}^{\text{restriction of } a\text{-harmonic func}} + \overbrace{(u^{\mathsf{b}}_{\omega_e} - I_H u^{\mathsf{b}}_{\omega_e})|_e}^{\text{basis functions in (II), small}}$

- $(u^{\mathsf{h}}_{\omega_e} - I_H u^{\mathsf{h}}_{\omega_e})|_e = \overbrace{\sum_{j=1}^{m-1} c_j v_e^j}^{\text{basis functions in (I)}} + O\left(\exp\left(-m^{\frac{1}{d+1}-\epsilon}\right)\|u\|_{H^1_a(\omega_e)}\right)$

(I) basis functions not dependent on f, but may depend on $\mathcal{L}_\theta$ (*local*)
(II) basis functions adapted to $\mathcal{L}_\theta$ and f (*local* and *small*)
(III) bubble part (*local* and *small*)

Summary of approximations:

- $u = u^{\mathsf{h}} + \overbrace{u^{\mathsf{b}}}^{\text{(III) local-n-small}}$ (harmonic-bubble splitting)
- $u^{\mathsf{h}} = \overbrace{(u^{\mathsf{h}} - I_H u^{\mathsf{h}})}^{\text{localized to each edge}} + \overbrace{I_H u^{\mathsf{h}}}^{\text{basis functions in (I)}}$ (interpolation part)
- $(u^{\mathsf{h}} - I_H u^{\mathsf{h}})|_e = \overbrace{(u^{\mathsf{h}}_{\omega_e} - I_H u^{\mathsf{h}}_{\omega_e})|_e}^{\text{restriction of } a\text{-harmonic func}} + \overbrace{(u^{\mathsf{b}}_{\omega_e} - I_H u^{\mathsf{b}}_{\omega_e})|_e}^{\text{basis functions in (II), small}}$
- $(u^{\mathsf{h}}_{\omega_e} - I_H u^{\mathsf{h}}_{\omega_e})|_e = \overbrace{\sum_{j=1}^{m-1} c_j v_e^j}^{\text{basis functions in (I)}} + O\left(\exp\left(-m^{\frac{1}{d+1}-\epsilon}\right)\|u\|_{H^1_a(\omega_e)}\right)$

(I) basis functions not dependent on f, but may depend on $\mathcal{L}_\theta$ (*local*)
(II) basis functions adapted to $\mathcal{L}_\theta$ and f (*local* and *small*)
(III) bubble part (*local* and *small*)

Summary of approximations:

- $u = u^{\mathsf{h}} + \overbrace{u^{\mathsf{b}}}^{\text{(III) local-n-small}}$ (harmonic-bubble splitting)
- $u^{\mathsf{h}} = \overbrace{(u^{\mathsf{h}} - I_H u^{\mathsf{h}})}^{\text{localized to each edge}} + \overbrace{I_H u^{\mathsf{h}}}^{\text{basis functions in (I)}}$ (interpolation part)
- $(u^{\mathsf{h}} - I_H u^{\mathsf{h}})|_e = \overbrace{(u^{\mathsf{h}}_{\omega_e} - I_H u^{\mathsf{h}}_{\omega_e})|_e}^{\text{restriction of } a\text{-harmonic func}} + \overbrace{(u^{\mathsf{b}}_{\omega_e} - I_H u^{\mathsf{b}}_{\omega_e})|_e}^{\text{basis functions in (II), small}}$
- $(u^{\mathsf{h}}_{\omega_e} - I_H u^{\mathsf{h}}_{\omega_e})|_e = \overbrace{\sum_{j=1}^{m-1} c_j v_e^j}^{\text{basis functions in (I)}} + O\left(\exp\left(-m^{\frac{1}{d+1}-\epsilon}\right)\|u\|_{H^1_a(\omega_e)}\right)$

(I) basis functions not dependent on f, but may depend on $\mathcal{L}_\theta$ (*local*)
(II) basis functions adapted to $\mathcal{L}_\theta$ and f (*local* and *small*)
(III) bubble part (*local* and *small*)

Summary of approximations:

- $u = u^{\mathsf{h}} + \overbrace{u^{\mathsf{b}}}^{\text{(III) local-n-small}}$ (harmonic-bubble splitting)
- $u^{\mathsf{h}} = \overbrace{(u^{\mathsf{h}} - I_H u^{\mathsf{h}})}^{\text{localized to each edge}} + \overbrace{I_H u^{\mathsf{h}}}^{\text{basis functions in (I)}}$ (interpolation part)
- $(u^{\mathsf{h}} - I_H u^{\mathsf{h}})|_e = \overbrace{(u^{\mathsf{h}}_{\omega_e} - I_H u^{\mathsf{h}}_{\omega_e})|_e}^{\text{restriction of } a\text{-harmonic func}} + \overbrace{(u^{\mathsf{b}}_{\omega_e} - I_H u^{\mathsf{b}}_{\omega_e})|_e}^{\text{basis functions in (II), small}}$
- $(u^{\mathsf{h}}_{\omega_e} - I_H u^{\mathsf{h}}_{\omega_e})|_e = \overbrace{\sum_{j=1}^{m-1} c_j v_e^j}^{\text{basis functions in (I)}} + O\left(\exp\left(-m^{\frac{1}{d+1}-\epsilon}\right)\|u\|_{H^1_a(\omega_e)}\right)$

(I) basis functions not dependent on f, but may depend on $\mathcal{L}_\theta$ (*local*)
(II) basis functions adapted to $\mathcal{L}_\theta$ and f (*local* and *small*)
(III) bubble part (*local* and *small*)

Summary of contributions

Our result: on an $O(H)$ mesh, u can be computed by

$$u = \underbrace{\sum_{i\in I_1} c_i\psi_i^{(1)}}_{\text{(I)}} + \underbrace{\sum_{i\in I_2} d_i\psi_i^{(2)}}_{\text{(II)}} + \underbrace{\sum_{i\in I_3} \psi_i^{(3)}}_{\text{(III)}} + O\left(\exp\left(-m^{\frac{1}{d+1}-\epsilon}\right)\right)$$

(Energy norm)

where: $\psi_i^{(1)}, \psi_i^{(2)}, \psi_i^{(3)}$ all have *local* support of size H

- $\psi_i^{(1)}$ obtained by *local* SVD of $\mathcal{L}_\theta$ $\qquad \#I_1 = O(m/H^d)$
- $\psi_i^{(2)}, \psi_i^{(3)}$ obtained by solving *local* $\mathcal{L}_\theta u = f$ $\qquad \#I_2, \#I_3 = O(1/H^d)$
- $c_i, d_i \in \mathbb{R}$ obtained by Galerkin's methods with basis functions $\psi_i^{(1)}, \psi_i^{(2)}$
- (II),(III)$= O(H)$ (Energy norm)

Can be generalized to Helmholtz's equations

Roadmap

1. Multiscale Computation with Exponential Convergence
   - Solve PDE as an approximation problem
   - Coarse-fine scale decomposition of solution space
   - Exponential efficiency of the coarse part
   - Numerical experiments
   - Related works
   - Take-aways

2. Parameter Learning with Provable Guarantees
   - Learning a kernel
   - Bayes' approach
   - Kernel Flow approach
   - Consistency as # of data $\to \infty$, and beyond
   - Take-aways

Numerical experiments: Elliptic equation

- The coefficient:

$$a(x) = \frac{1}{6}\left(\frac{1.1+\sin\left(2\pi x_1/\epsilon_1\right)}{1.1+\sin\left(2\pi x_2/\epsilon_1\right)} + \frac{1.1+\sin\left(2\pi x_2/\epsilon_2\right)}{1.1+\cos\left(2\pi x_1/\epsilon_2\right)} + \frac{1.1+\cos\left(2\pi x_1/\epsilon_3\right)}{1.1+\sin\left(2\pi x_2/\epsilon_3\right)} + \frac{1.1+\sin\left(2\pi x_2/\epsilon_4\right)}{1.1+\cos\left(2\pi x_1/\epsilon_4\right)} + \frac{1.1+\cos\left(2\pi x_1/\epsilon_5\right)}{1.1+\sin\left(2\pi x_2/\epsilon_5\right)} + \sin\left(4x_1^2x_2^2\right)+1\right),$$

  where $\epsilon_1 = 1/5$, $\epsilon_2 = 1/13$, $\epsilon_3 = 1/17$, $\epsilon_4 = 1/31$, $\epsilon_5 = 1/65$.

- The right hand side $f = -1$

Only using (I)

only use terms in (I) for the approximation (no information of f)

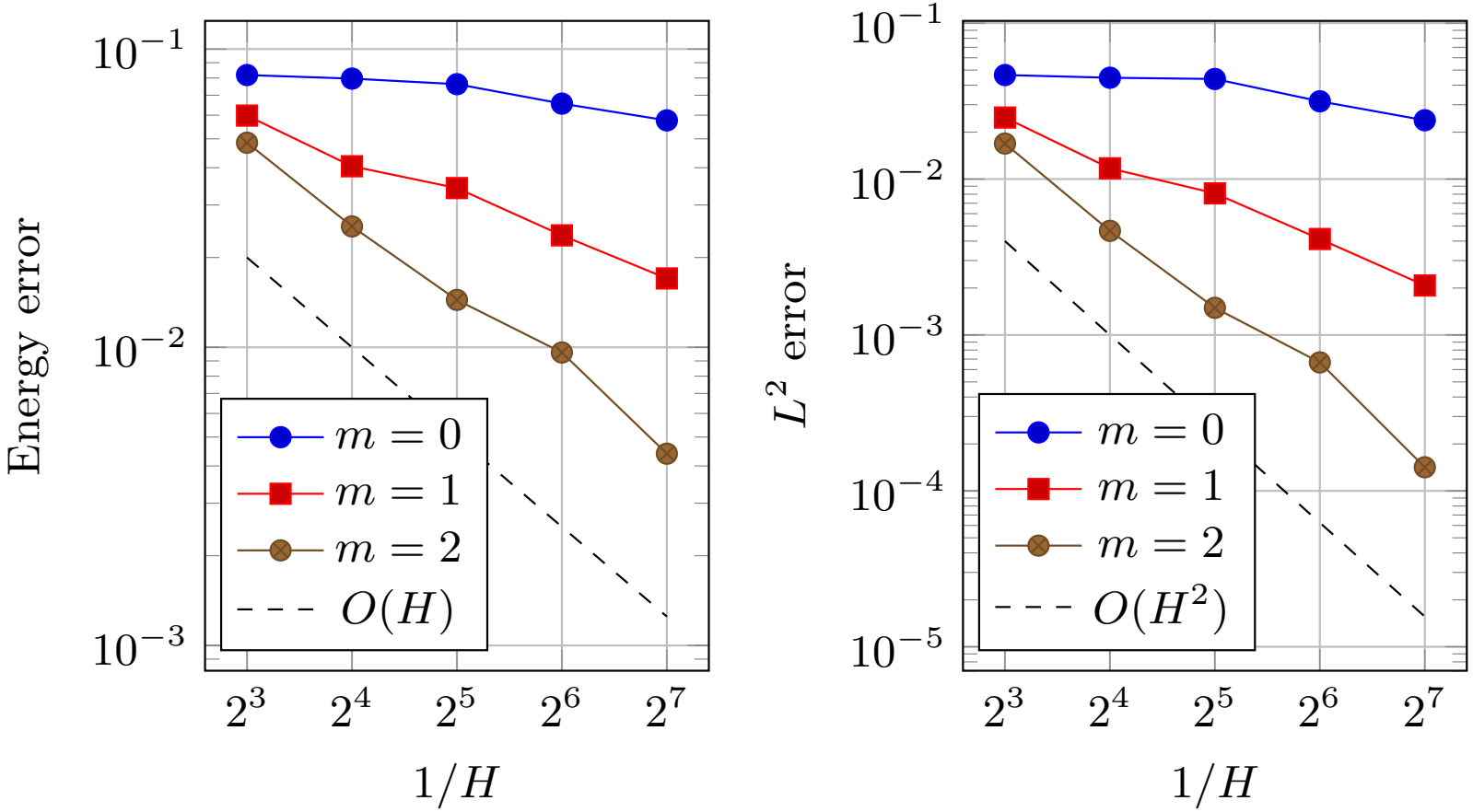


Figure: The coefficient a has multiple scales, $f = -1$

Observation: increasing $m \to O(H)$ and $O(H^2)$ accuracy

Using (I)(II)(III)

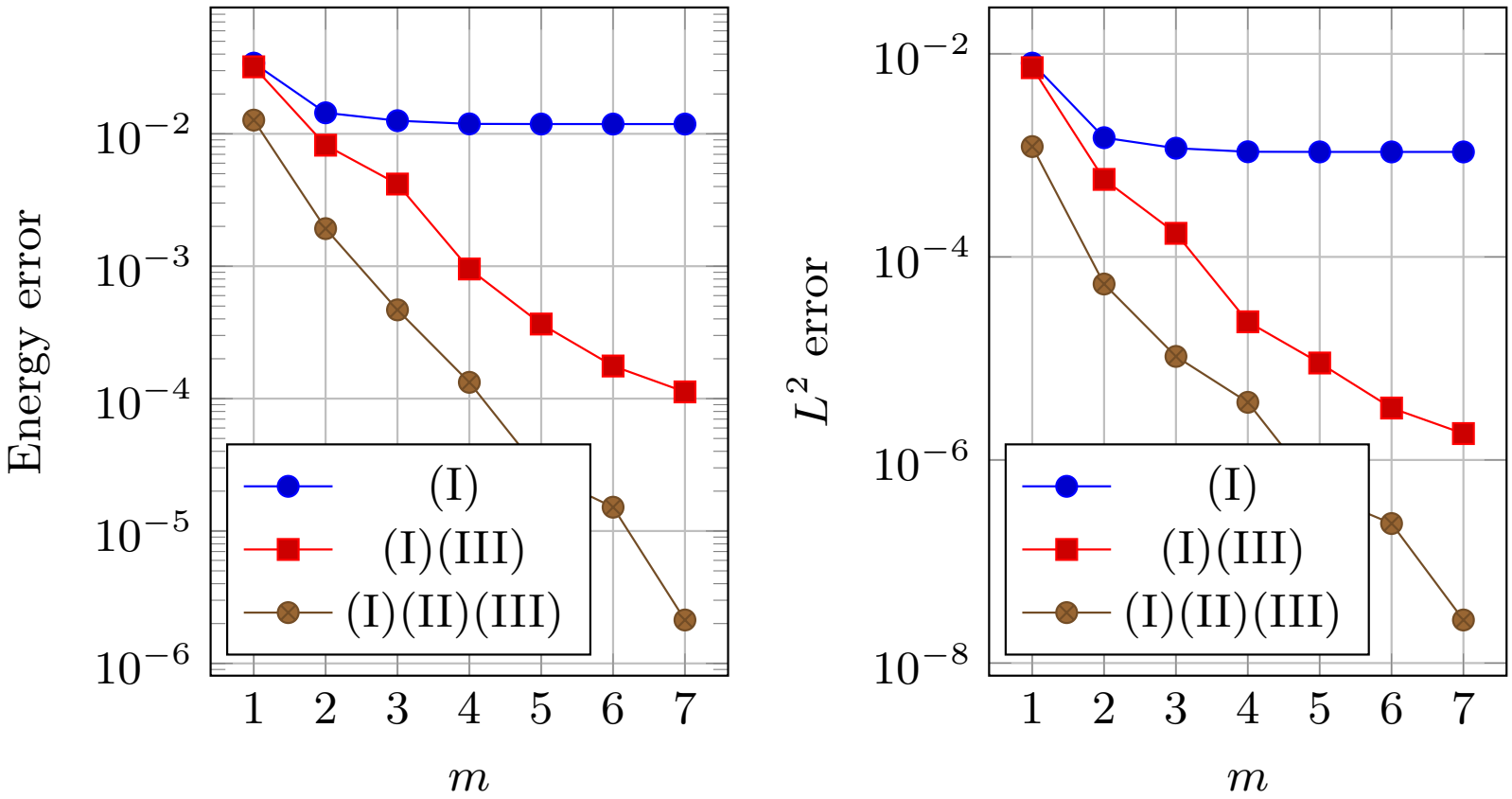


Figure: The coefficient a has multiple scales, $f = -1$, $H = 1/32$

Nearly exponential convergence for both (I)+(III) and (I)+(II)+(III)

Roadmap

1. Multiscale Computation with Exponential Convergence
   - Solve PDE as an approximation problem
   - Coarse-fine scale decomposition of solution space
   - Exponential efficiency of the coarse part
   - Numerical experiments
   - Related works
   - Take-aways

2. Parameter Learning with Provable Guarantees
   - Learning a kernel
   - Bayes' approach
   - Kernel Flow approach
   - Consistency as # of data $\to \infty$, and beyond
   - Take-aways

Related works for elliptic equations

- Generalized FEM, Multiscale FEM (MsFEM), GMsFEM, etc.
  - General strategy: local approximation + global coupling
  - Our method belongs to this family, but with a noval edge coupling seen as an extension of [Hou, Liu 2016]
  - [Babuska, Lipton 2011] first obtained nearly exponential convergence results, using partition of unity (PUM)
  - Compared to the PUM used in [Babuska, Lipton 2011], our edge coupling does not destroy the orthogonality between a-harmonic and bubble parts; (I)+(III) very efficient!
  - Our non-overlapped domain decomposition yields smaller support for basis functions, but # increases since edges are more than elements

- Variational Multiscale Methods (VMS), LOD, Gamblets, etc.
  - General strategy: coarse-fine scale decomposition of solution space + localization of coarse part
  - Our methods use an energy-orthogonal coarse-fine decomposition
  - Compared to LOD [Målqvist, Peterseim 2014] and Gamblets [Owahdi, 2017], our methods have better local adaptivity, and convenient exp accuracy
  - However, our current algorithm is limited to two levels, as in constrast with the multiple levels in Gamblets

Roadmap

1. Multiscale Computation with Exponential Convergence
   - Solve PDE as an approximation problem
   - Coarse-fine scale decomposition of solution space
   - Exponential efficiency of the coarse part
   - Numerical experiments
   - Related works
   - Take-aways

2. Parameter Learning with Provable Guarantees
   - Learning a kernel
   - Bayes' approach
   - Kernel Flow approach
   - Consistency as # of data $\to \infty$, and beyond
   - Take-aways

Take-aways

Coarse-fine scale decomposition adapted to $\mathcal{L}_\theta$ and f

- Coarse a-harmonic part: exponentially efficient approximation
  - Restrictions of a-harmonic functions are of low complexity
- Fine bubble part: local computation
  - Small magnitude; can be ignored

Nearly exponential convergence!

Can be generalized to Helmholtz's equation

Roadmap

1. Multiscale Computation with Exponential Convergence
   - Solve PDE as an approximation problem
   - Coarse-fine scale decomposition of solution space
   - Exponential efficiency of the coarse part
   - Numerical experiments
   - Related works
   - Take-aways

2. Parameter Learning with Provable Guarantees
   - Learning a kernel
   - Bayes' approach
   - Kernel Flow approach
   - Consistency as # of data $\to \infty$, and beyond
   - Take-aways

For learning problems

- **Context**: Supervised learning
- **Approach**: Gaussian process regression / kernel methods
- **Question of focus**: How to select kernels based on data
- **Algorithms in use:**
  - Bayesian: Empirical Bayes
  - Approximation theoretic: Kernel Flow
- **Contribution**:
  - Theory: Consistency for a Matérn class model (**this talk**)
  - Experiments: beyond Matérn model, and include model misspecification

Gaussian process regression (GPR)

- Supervised learning / nonparameteric regression / interpolation

Recover $u^\dagger : D \subset \mathbb{R}^d \to \mathbb{R}$ from

$$y_i = u^\dagger(x_i), 1 \leq i \leq N \qquad \text{(Noise-free data)}$$

- GPR solution / Kernel method:

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

(Depend on kernel K_θ, data set $\mathcal{X}$, and truth $u^\dagger$)

Notation: ($\theta \in \Theta$ is a *hierarchical parameter*)

$$K_\theta : D \times D \to \mathbb{R}$$

$$\mathcal{X} = \{x_1, ..., x_N\}, \text{ and } u^\dagger(\mathcal{X}) \in \mathbb{R}^N, K_\theta(\mathcal{X}, \mathcal{X}) \in \mathbb{R}^{N \times N}$$

$$K_\theta(\cdot, \mathcal{X}) : D \to \mathbb{R}^N, \text{ and } u(\cdot, \theta, \mathcal{X}) : D \to \mathbb{R}$$

Gaussian process regression (GPR)

- Supervised learning / nonparameteric regression / interpolation

Recover $u^\dagger : D \subset \mathbb{R}^d \to \mathbb{R}$ from

$$y_i = u^\dagger(x_i), 1 \le i \le N \qquad \text{(Noise-free data)}$$

- GPR solution / Kernel method:

$$u(\cdot, \theta, \mathcal{X}) = K_\theta(\cdot, \mathcal{X})[K_\theta(\mathcal{X}, \mathcal{X})]^{-1} u^\dagger(\mathcal{X})$$
(Depend on kernel K_θ, data set $\mathcal{X}$, and truth $u^\dagger$)

Notation: ($\theta \in \Theta$ is a *hierarchical parameter*)

$$K_\theta : D \times D \to \mathbb{R}$$
$$\mathcal{X} = \{x_1, ..., x_N\}, \text{ and } u^\dagger(\mathcal{X}) \in \mathbb{R}^N, K_\theta(\mathcal{X}, \mathcal{X}) \in \mathbb{R}^{N \times N}$$
$$K_\theta(\cdot, \mathcal{X}) : D \to \mathbb{R}^N, \text{ and } u(\cdot, \theta, \mathcal{X}) : D \to \mathbb{R}$$

What's the problem?

- Any $\theta \in \Theta$ yields an interpolated solution on $\mathcal{X}$:

$$u^\dagger(x_i) = u(x_i, \theta, \mathcal{X}), 1 \leq i \leq N$$

  i.e., zero training error

But, for out-of-sample / generalization errors, how to pick a good θ?

- θ can encode the "physics" underlying the data
- A model selection problem – learn the hierarchical parameter θ

Roadmap

1. Multiscale Computation with Exponential Convergence
   - Solve PDE as an approximation problem
   - Coarse-fine scale decomposition of solution space
   - Exponential efficiency of the coarse part
   - Numerical experiments
   - Related works
   - Take-aways

2. Parameter Learning with Provable Guarantees
   - Learning a kernel
   - Bayes' approach
   - Kernel Flow approach
   - Consistency as # of data $\to \infty$, and beyond
   - Take-aways

Bayes' solution

- Put a prior on θ, and $u^\dagger|\theta \sim \mathcal{N}(0, K_\theta)$ — then calculate the posterior
- Empirical Bayes (EB) with uninformative prior:

$$\theta^{\mathrm{EB}}(\mathcal{X}, u^\dagger) = \operatorname*{argmin}_{\theta\in\Theta} \mathsf{L}^{\mathrm{EB}}(\theta, \mathcal{X}, u^\dagger)$$

$$\mathsf{L}^{\mathrm{EB}}(\theta, \mathcal{X}, u^\dagger) = u^\dagger(\mathcal{X})^{\mathsf{T}}[K_\theta(\mathcal{X}, \mathcal{X})]^{-1}u^\dagger(\mathcal{X}) + \log\det K_\theta(\mathcal{X}, \mathcal{X})$$

Maximum Likelihood Estimate!

Roadmap

1. Multiscale Computation with Exponential Convergence
   - Solve PDE as an approximation problem
   - Coarse-fine scale decomposition of solution space
   - Exponential efficiency of the coarse part
   - Numerical experiments
   - Related works
   - Take-aways

2. Parameter Learning with Provable Guarantees
   - Learning a kernel
   - Bayes' approach
   - Kernel Flow approach
   - Consistency as # of data $\to \infty$, and beyond
   - Take-aways

Approximation-theoretic approach

- Why $\theta, u^\dagger$ have a prior distribution? — may be brittle to misspecification
- Go straightforward: set a target cost d, and optimize$_\theta$ $\mathsf{d}(u^\dagger, u(\cdot, \theta, \mathcal{X}))$
- Problem: $u^\dagger$ not available — solution: numeric approximation

$$\min_\theta \mathsf{d}(u(\cdot, \theta, \mathcal{X}), u(\cdot, \theta, \pi\mathcal{X})) \qquad \text{(One example)}$$

π: subsampling operator (similar to cross-validation)

Approximation-theoretic approach

- Why $\theta, u^\dagger$ have a prior distribution? — may be brittle to misspecification
- Go straightforward: set a target cost d, and optimize$_\theta$ $\mathsf{d}(u^\dagger, u(\cdot, \theta, \mathcal{X}))$
- Problem: $u^\dagger$ not available — solution: numeric approximation

$$\min_\theta \mathsf{d}(u(\cdot, \theta, \mathcal{X}), u(\cdot, \theta, \pi\mathcal{X})) \qquad \text{(One example)}$$

π: subsampling operator (similar to cross-validation)

Approximation-theoretic approach

- Why $\theta, u^\dagger$ have a prior distribution? — may be brittle to misspecification
- Go straightforward: set a target cost d, and optimize$_\theta$ $\mathsf{d}(u^\dagger, u(\cdot, \theta, \mathcal{X}))$
- Problem: $u^\dagger$ not available — solution: numeric approximation

$$\min_\theta \mathsf{d}(u(\cdot, \theta, \mathcal{X}), u(\cdot, \theta, \pi\mathcal{X})) \qquad \text{(One example)}$$

π: subsampling operator (similar to cross-validation)

Kernel Flow (KF)

KF uses a specific d: [Owhadi, Yoo 2018 & 2020], [Hamzi, Owhadi 2020]

$$\theta^{\mathrm{KF}}(\mathcal{X}, \pi\mathcal{X}, u^\dagger) = \operatorname*{argmin}_{\theta\in\Theta} \mathsf{L}^{\mathrm{KF}}(\theta, \mathcal{X}, \pi\mathcal{X}, u^\dagger)$$

$$\mathsf{L}^{\mathrm{KF}}(\theta, \mathcal{X}, \pi\mathcal{X}, u^\dagger) = \frac{\|u(\cdot,\theta,\mathcal{X}) - u(\cdot,\theta,\pi\mathcal{X})\|^2_{K_\theta}}{\|u(\cdot,\theta,\mathcal{X})\|^2_{K_\theta}}$$

where

- π: a subsampling operator, so $\pi\mathcal{X} \subset \mathcal{X}$
- $\|\cdot\|_{K_\theta}$: RKHS norm determined by K_θ

A kernel is good, if subsampling data does not influence solution much.

Roadmap

1. Multiscale Computation with Exponential Convergence
   - Solve PDE as an approximation problem
   - Coarse-fine scale decomposition of solution space
   - Exponential efficiency of the coarse part
   - Numerical experiments
   - Related works
   - Take-aways

2. Parameter Learning with Provable Guarantees
   - Learning a kernel
   - Bayes' approach
   - Kernel Flow approach
   - Consistency as # of data $\to \infty$, and beyond
   - Take-aways

Consistency?

Question: How do θ^{EB} and θ^{KF} behave, as # of data $\to \infty$?

- We answer the question for some specific model of $u^\dagger, \theta$ and $\mathcal{X}$

Theory: set-up and theorem

A specific Matérn-like regularity model:

- Domain: $D = \mathbb{T}^d = [0,1]^d_{\text{per}}$
- Lattice data $\mathcal{X}_q = \{j \cdot 2^{-q}, j \in J_q\}$
  where $J_q = \{0, 1, ..., 2^q - 1\}^d$, # of data: 2^{qd}
- Kernel $K_\theta = (-\Delta)^{-t}$, and $\theta = t$
- Subsampling operator in KF: $\pi\mathcal{X}_q = \mathcal{X}_{q-1}$

Theorem (Y. Chen, H. Owhadi, A.M. Stuart, 2020)

Informal: if $u^\dagger \sim \mathcal{N}(0, (-\Delta)^{-s})$ for some s, then as $q \to \infty$,

$$\theta^{\text{EB}} \to s \quad \text{and} \quad \theta^{\text{KF}} \to \frac{s - d/2}{2} \quad \text{in probability}$$

- Equivalently, $u^\dagger$ is the solution to $(-\Delta)^{s/2} u^\dagger = f$ for white noise f
  Thus, can learn the *fractional physical laws* underlying the data
- Analysis based on multiresolution decomposition and uniform convergence of random series

Theory: set-up and theorem

A specific Matérn-like regularity model:

- Domain: $D = \mathbb{T}^d = [0,1]^d_{\text{per}}$
- Lattice data $\mathcal{X}_q = \{j \cdot 2^{-q}, j \in J_q\}$
  where $J_q = \{0, 1, ..., 2^q - 1\}^d$, # of data: 2^{qd}
- Kernel $K_\theta = (-\Delta)^{-t}$, and $\theta = t$
- Subsampling operator in KF: $\pi\mathcal{X}_q = \mathcal{X}_{q-1}$

Theorem (Y. Chen, H. Owhadi, A.M. Stuart, 2020)

Informal: if $u^\dagger \sim \mathcal{N}(0, (-\Delta)^{-s})$ for some s, then as $q \to \infty$,

$$\theta^{\text{EB}} \to s \quad \text{and} \quad \theta^{\text{KF}} \to \frac{s - d/2}{2} \quad \text{in probability}$$

- Equivalently, $u^\dagger$ is the solution to $(-\Delta)^{s/2} u^\dagger = f$ for white noise f
  Thus, can learn the *fractional physical laws* underlying the data
- Analysis based on multiresolution decomposition and uniform convergence of random series

Experiments justifying the theory

How it works in practice?

- $d = 1, s = 2.5$, # of data $N = 2^9$, mesh size 2^{-10}

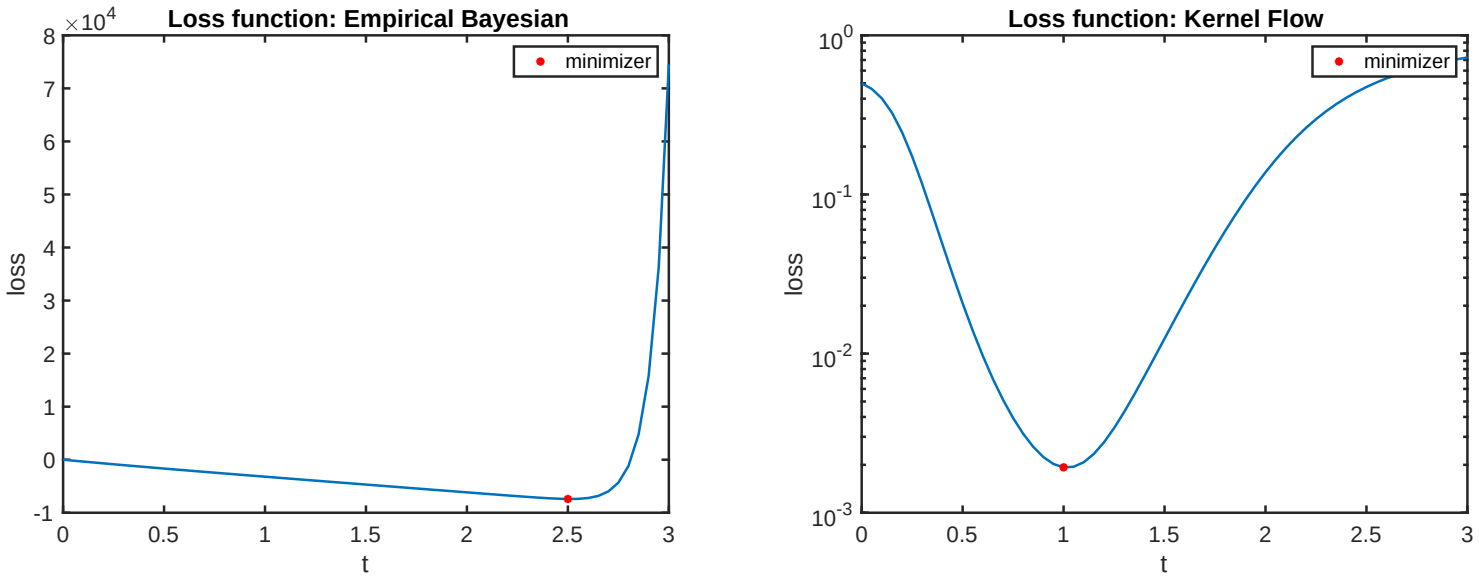


Figure: Left: EB loss; right: KF loss

- Patterns in the loss function (our theory can predict!)
  - EB: first linear, then blow up quickly
  - KF: more symmetric

Experiments justifying the theory

How it works in practice?

- $d = 1, s = 2.5$, # of data $N = 2^9$, mesh size 2^{-10}

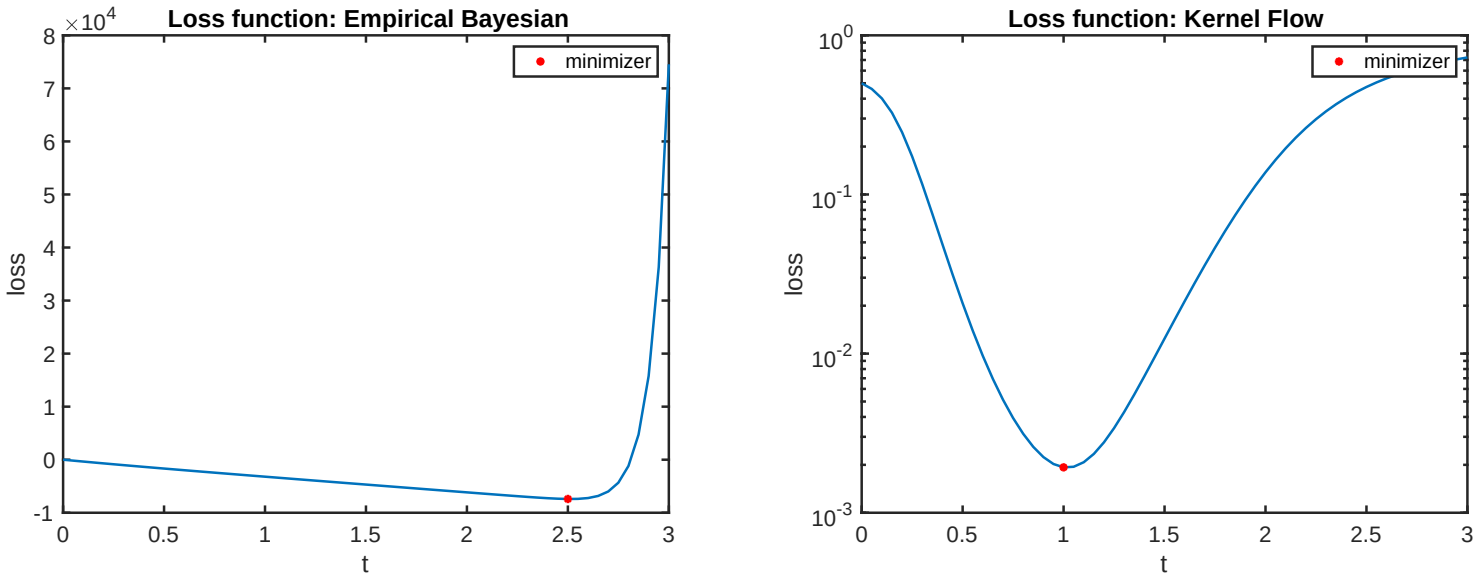


Figure: Left: EB loss; right: KF loss

- Patterns in the loss function (our theory can predict!)
  - EB: first linear, then blow up quickly
  - KF: more symmetric

Selection Bias

Next Question: How are the limits s $(= 2.5)$ and $\frac{s-d/2}{2}$ $(= 1)$ special?

- What is the *implicit bias* of EB and KF algorithms?
- Our strategy: look at their L^2 population errors

Experiment I

- # of data: 2^q; compute $\mathbb{E}_{u^\dagger} \|u^\dagger(\cdot) - u(\cdot, t, \mathcal{X}_q)\|_{L^2}^2$

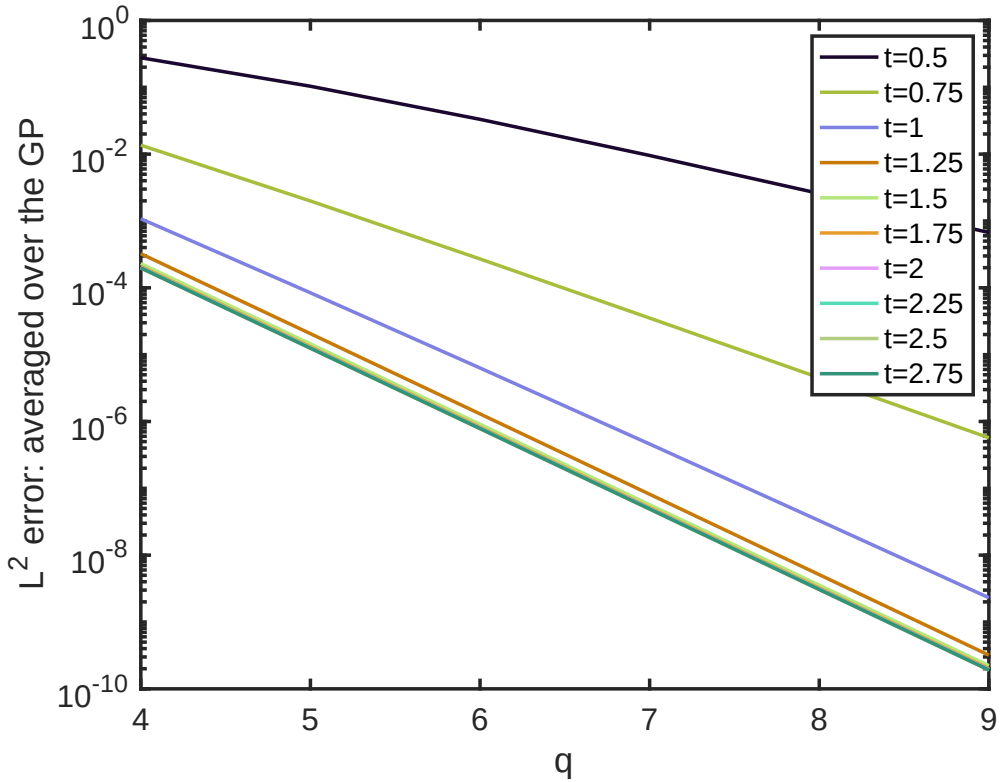


Figure: L^2 error: averaged over the GP

- $\frac{s-d/2}{2}$ $(= 1)$ is the minimal t that suffices for the fastest rate of L^2 error

Experiment II

- # of data: $2^q, q = 9$; compute $\mathbb{E}_{u^\dagger} \|u^\dagger(\cdot) - u(\cdot, t, \mathcal{X}_q)\|_{L^2}^2$

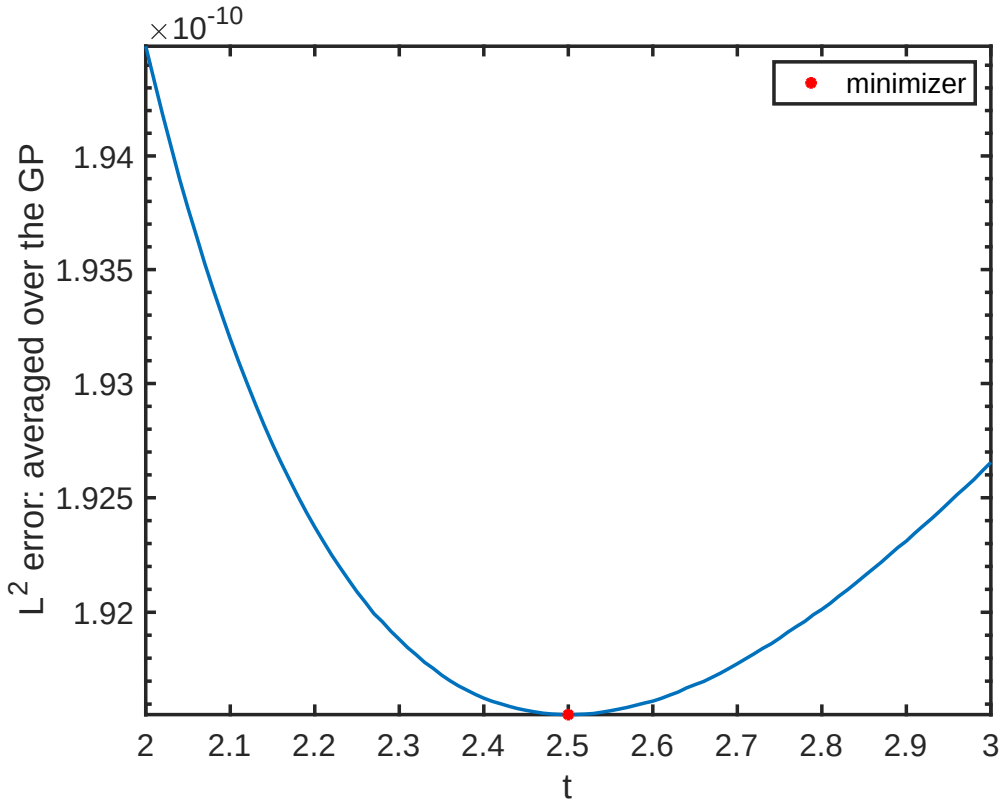


Figure: L^2 error: averaged over the GP, for $q = 9$

- s $(= 2.5)$ is the t that achieves the minimal L^2 error in expectation

Roadmap

1. Multiscale Computation with Exponential Convergence
   - Solve PDE as an approximation problem
   - Coarse-fine scale decomposition of solution space
   - Exponential efficiency of the coarse part
   - Numerical experiments
   - Related works
   - Take-aways

2. Parameter Learning with Provable Guarantees
   - Learning a kernel
   - Bayes' approach
   - Kernel Flow approach
   - Consistency as # of data $\to \infty$, and beyond
   - Take-aways

Take-aways

- For Matérn-like kernel model, EB and KF have different selection bias
  - EB selects the θ that achieves the minimal L^2 error in expectation
  - KF selects the minimal θ that suffices for the fastest rate of L^2 error
- More comparisons between EB and KF in our paper
  - Estimate amplitude and lengthscale in $\mathcal{N}(0, \sigma^2(-\Delta + \tau^2 I)^{-s})$
  - Variance of estimators
  - Robustness to model misspecification (important!)
  - Computational cost

Parameter learning: via Bayes or approximation-theoretic?

Thank you!