Project Ideas

AMATH 563, Spring 2023

This document outlines project ideas and directions for AMATH 563 in Spring of 2023. Each project idea is accompanied by a few references that demonstrate the main ideas/problems of interest in that project.

- **Consider reading these references and assemble a team of 1-3 people to do the project.**
- **You have until April 21 to choose your project idea and form your team. Convey this information to Katherine Johnston who will also help with the organization of the projects.**
- **You are also welcome to propose your own project. To do so please talk to me about it and write a brief description of your idea and goals, no more than one page, and send it to me and Katherine including references. Your proposal should outline the following information:**
  - **What do you want to do?**
  - **Why do you want to do it?**
  - **How do you plan to achieve your goals?**

Project Ideas

Generative modeling with operator valued kernels

Context: Generative modeling is the problem of generating approximate samples from a target measure ν, such as a Bayesian posterior measure. In the statistical inference literature the most widely used approach for this task is Markov chain Monte Carlo [18]. In recent years, transport based generative models have become very popular with the rise of models such as Generative Adversarial Nets (GANs)[5] and Normalizing flows (NFs) [11]. The basic idea is to train a map T such that $T_\sharp \eta \approx \nu$ where η is some reference measure, such as a standard Gaussian, and $T_\sharp \eta$ is simply the law of $T(x)$ for $x \sim \eta$. This is often achieved by solving optimization problems of the form

$$T = \mathrm{argmin}_S D(S_\sharp \eta^N, \nu^N),$$

where D is an appropriate statistical divergence such as MMD and η^N, ν^N are empirical approximations to η and ν obtained from i.i.d. samples.

Goal: You will design and implement a transport based generative model where D is taken to be a divergence of your choosing such as MMD and approximate T by parameterizing it within an RKHS defined by a matrix/operator valued kernel [9]. You can derive a representer theorem for this problem that enables the efficient learning of T and benchmark your algorithm on a few example data sets of your choosing.

Kernel PDE solvers

Context: Numerical solution of PDEs is a fundamental task in applied mathematics, dominated by classic approaches such as finite differences, finite elements, and finite volume methods. Recently, a lot of interest has been generated around the idea of solving PDEs with ML techniques such as neural nets (NNs) [17, 7]. The popular PINNs family can be thought of as a collocation method that parameterizes the solution u using the NN. One can design an analogous algorithm by looking for a solution in an RKHS which leads to optimization problems of the form

$$\mathrm{minimize}_{u \in \mathrm{RKHS}} \|u\| \qquad \text{subject to} \qquad \mathcal{P}(u) = f \qquad \text{at collocation points.}$$

where $\mathcal{P}$ is the differential operator of the PDE and f is the forcing/source term of the PDE. This approach was developed in [16] for linear elliptic PDEs and extended to generic nonlinear PDEs in [4].

Goal: You will implement and develop a kernel PDE solver for up to three nonlinear PDEs of your choosing. You can investigate the convergence properties of the algorithm in relation to the choice of the kernel, the distribution of collocation points, and other parameters in the algorithm. You may also investigate approaches for speeding up performance, such as sparse GPs or random feature formulations.

Functional PDE regression

Context: One of the modern problems of scientific computing, due to the rise of ML and its applications in science and engineering, is that of discovering differential equations that govern physical phenomenon. The fundamental question is: given a set of pairs $\{u_i, f_i\}_{i=1}^N$ satisfying a PDE $\mathcal{P}(u_i) = f_i$, learn the functional form of $\mathcal{P}$, i.e., the relationship between the partial derivatives of u that the describe the left hand side of the PDE. For example,

$$\mathcal{P}(u) = -\Delta u + u^2 \equiv P(\Delta u, u)$$

where $P(x, y) = -x + y^2$. The goal of functional PDE regression or PDE discovery is to learn P. The most widely known example of such an algorithm is Sparse Identification of Nonlinear Dynamics (SINDy) [10] and its PDE extension PDE-Find [19].

Goal: Recently kernel methods have been proposed as an alternative approach for functional PDE regression [14]. Your goal in this project is to implement and investigate the performance of the kernel approach for denoising of input data and for learning the functional form of the PDE. You will benchmark this method on two to three PDEs or ODEs of your choosing.

Operator learning with operator valued kernels

Context: Operator learning is the task of approximating a mapping between two, possibly infinite dimensional, Banach spaces

$$\mathcal{G} : \mathcal{X} \to \mathcal{Y},$$

given a training data set $\{x_i, y_i\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$ [12, 2, 6]. In scientific computing and engineering this mapping is often the solution map of a PDE or the parameter-to-state map of a complex physical process; see the examples in [6]. Recently this field has attracted a lot of attention due to the rise of NN based methods such as Fourier Neural Operator [13] and the Deep Operator Nets [15].

Goal: Your goal is to investigate the competitiveness of kernel methods, in particular operator valued kernels, for the task of operator learning. You may design and implement an operator learning framework using kernel regression and validate and benchmark it on some of the test data sets in the literature. The paper [6] has a lot of nice examples and available data sets and code.

Graphical semi-supervised learning

Context: Semi-supervised learning (SSL) is the problem of labeling a set of data points from a small labeled subset [21]. More precisely, suppose we are given a set of inputs $\{x_i\}_{i=1}^N$ among which only the labels of the first M points are known, i.e., we have $\{x_i, y_i = \text{Label}(x_i)\}_{i=1}^M$. Then the goal of SSL is to find/estimate $\{\text{Label}(x_i)\}_{i=M+1}^N$. Typically the assumption of SSL is that $M \ll N$ so that the labeled data is very sparse. A particularly useful approach to SSL is the so called family of graphical algorithms [1, 8], where a graph G is built upon the x_i's and then a regularized regression problem is formulated on this graph to find a (latent) function $u : G \mapsto$ Label space that predicts the label of the remaining points. The regularization often uses a Graph Laplacian matrix which can be thought of as a discretization of the usual Laplacian differential operator.

Goal: Your goal is to formulate such Laplacian based SSL algorithms within the framework of RKHS methods via the connection between PDS kernels such as the Matérn family, and the Green's function of elliptic differential operators; see for example [3, 20]. You will further implement and benchmark graphical SSL algorithms such as the probit method within your RKHS framework and investigate the choice of your kernel on the performance of the method.

References

[1] Mikhail Belkin and Partha Niyogi. Semi-supervised learning on riemannian manifolds. *Machine learning*, 56:209–239, 2004.

[2] Kaushik Bhattacharya, Bamdad Hosseini, Nikola B Kovachki, and Andrew M Stuart. Model reduction and neural networks for parametric pdes. *The SMAI journal of computational mathematics*, 7:121–157, 2021.

[3] Viacheslav Borovitskiy, Iskander Azangulov, Alexander Terenin, Peter Mostowsky, Marc Deisenroth, and Nicolas Durrande. Matérn gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics*, pages 2593–2601. PMLR, 2021.

[4] Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M Stuart. Solving and learning nonlinear pdes with gaussian processes. *Journal of Computational Physics*, 447:110668, 2021.

[5] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

[6] Maarten De Hoop, Daniel Zhengyu Huang, Elizabeth Qian, and Andrew M Stuart. The cost-accuracy trade-off in operator learning with neural networks. 2022.

[7] Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

[8] Franca Hoffmann, Bamdad Hosseini, Zhi Ren, and Andrew M Stuart. Consistency of semi-supervised learning algorithms on graphs: Probit and one-hot methods. *The Journal of Machine Learning Research*, 21(1):7549–7603, 2020.

[9] Hachem Kadri, Emmanuel Duflos, Philippe Preux, Stéphane Canu, Alain Rakotomamonjy, and Julien Audiffren. Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17, 2016.

[10] Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A*, 474(2219):20180335, 2018.

[11] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020.

[12] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[13] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, et al. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*.

[14] Da Long, Nicole Mrvaljevic, Shandian Zhe, and Bamdad Hosseini. A kernel approach for pde discovery and operator learning. 2022.

[15] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

[16] Houman Owhadi. Bayesian numerical homogenization. *Multiscale Modeling & Simulation*, 13(3):812–828, 2015.

[17] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[18] Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*. Springer, 1999.

[19] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science advances*, 3(4):e1602614, 2017.

[20] Daniel Sanz-Alonso and Ruiyi Yang. The spde approach to matérn fields: Graph representations. *Statistical Science*, 37(4):519–540, 2022.

[21] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. 2005.