Machine Learning Approach for Identifying Nonlinear Partial Differential Equations

Project report submitted for

IVth Semester Minor Project-I

in

Department of Mathematics

By,

Aakash Agarwal (191010201)

Amulya Varshney (191000204)

Sakshi Kumari (191000043)

Department of Mathematics

Dr. Shyama Prasad Mukherjee

International Institute of Information Technology, Naya Raipur

(A Joint Initiative of Govt. of Chhattisgarh and NTPC)

Email: iiitnr@iiitnr.ac.in, Tel: (0771) 2474040, Web: www.iiitnr.ac.in

CERTIFICATE

This is to certify that the project titled “Machine Learning of nonlinear Partial Differential Equations” by “Aakash Agarwal, Amulya Varshney, Sakshi Kumari” has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree/diploma.

(Signature of Guide)

_____

Dr. Kuldip Singh Patel

Assistant Professor

Department of Mathematics

Dr. SPM IIIT-NR

March, 2021

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature of Authors)

_____

Aakash Agarwal

(191010201)

_____

Amulya Varshney

(191000204)

_____

Sakshi Kumari

(191000043)

Date : _____

Approval Sheet

This project report entitled “Machine Learning of nonlinear Partial Differential Equations” by “Aakash Agarwal, Amulya Varshney, Sakshi Kumari” is approved for IVth Semester Minor Project-I.

(Signature of Examiner - I)

_____

Dr. Kuldip Singh Patel

(Signature of Examiner - II)

_____

Dr. Ramakrishna Bandi

(Signature of Chair)

_____

Dr. Mithilesh Chaube

Date: _____ Place: _____

ABSTRACT

Gaussian processes are a powerful tool for probabilistic inference over functions, that enables us to strike a balance between model complexity and data fitting using Machine Learning Techniques.

In this Paper, We introduce the concept of Numerical Gaussian Processes, which we define as Gaussian Processes with covariance functions resulting from temporal discretization of time-dependent partial differential equations. Numerical Gaussian processes, by construction, are designed to deal with cases where we observe noisy data on black-box initial conditions, and are interested in quantifying the uncertainty associated with such noisy data in our solutions to time-dependent partial differential equations. This is an attempt to construct structured and data-efficient learning machines, which are explicitly informed by the underlying physics that possibly generated the observed data. The effectiveness of the proposed approach is demonstrated through several benchmark problems involving linear and nonlinear time-dependent operators. In all examples, we are able to recover accurate approximations of the latent solutions, and consistently propagate uncertainty in computation, even in cases involving very long time integration.

Keywords: ***Gaussian Process · Underlying Physics · Partial Differential Equations (PDEs) · Latent Solution***

Table of Contents

Title **Page No.**

ABSTRACT……. i

TABLE OF CONTENTS……..... ii

LIST OF TABLES…….. iii

LIST OF FIGURES……..... iv

CHAPTER 1 INTRODUCTION 1

1.1 Linear Multistep Methods……..... 1

CHAPTER 2 PROBLEM STATEMENT 2

CHAPTER 3 PRIOR 3

CHAPTER 4 WORKFLOW AND COST 4

CHAPTER 5 TRAINING 5

Section 5.1……. 5

CHAPTER 6 POSTERIOR AND PROPAGATING UNCERTAINTY 5

CHAPTER 7 BURGER'S EQUATION 6

REFERENCES 7

1. Introduction

Data-driven methods are taking center stage across many disciplines of science, and machine learning techniques have achieved groundbreaking results across a diverse spectrum of pattern recognition tasks. Modeling the physical world through the lens of mathematics typically translates into deriving conservation laws from first principles, which often take the form of systems of partial differential equations. In many practical settings, the solution of such systems is only accessible by means of numerical algorithms that provide sensible approximations to given quantities of interest. In this work, we aim to capitalize on the long-standing developments of classical methods in numerical analysis and revisit partial differential equations from a statistical inference viewpoint. The merits of this approach are twofold. First, it enables the construction of data-efficient learning machines that can encode physical conservation laws as structured prior information. Second, it allows the design of novel numerical algorithms that can seamlessly blend equations and noisy data, infer latent quantities of interest (e.g., the solution to a partial differential equation), and naturally quantify uncertainty in computations.

Let us consider a Linear Partial differential equation of the form.

$$u_t = \mathcal{L}_x u, \quad x \in \Omega, \ t \in [0, T],$$

here $\boldsymbol{L_x}$ is a linear operator and u(t, x) denotes the latent solution.
In general, we envision that the proposed method could be most useful in cases where one would like to extrapolate from an initial condition obtained from noisy experimental data, and where a governing equation is known.

For starters, let us try to convey the main ideas of this work using the Euler time-stepping scheme

$$u^n = u^{n-1} + \Delta t \ \mathcal{L}_x u^{n-1}.$$

We now place a Gaussian process prior on $\boldsymbol{u^{n-1}}$, i.e,

$$u^{n-1}(x) \sim \mathcal{GP}(0, k_{u,u}^{n-1,n-1}(x, x', \theta)).$$

Here θ denotes the hyperparameters of the covariance function .

1.1 Linear Multistep Methods

Let us start with the most general form of the linear multistep methods, i.e.,

$$u^n = \sum_{i=1}^{m} \alpha_i u^{n-1} + \Delta t \sum_{i=0}^{m} \beta_i L_x u^{n-i}$$

We can have different values for the parameters $\boldsymbol{\alpha_i}$ and $\boldsymbol{\beta_i}$ result in specific equations. Linear multistep methods can be equivalently written as

$$P_x u^n = \sum_{i=1}^{m} Q_x^i u^{n-i}$$

where $\mathbf{P_x u := u - \Delta t \beta_0 L_x u}$ and $\mathbf{Q_x^i u := \alpha_i u + \Delta t \beta_i L_x u}$.

For every $\mathbf{j = 0, 1, \ldots, m}$ and some $\boldsymbol{\tau \in [0, 1]}$ which depends on the specific choices for the values of the parameters $\boldsymbol{\alpha_i}$ and $\boldsymbol{\beta_i}$, we define $\mathbf{u^{n-j+\tau}}$ to be given by:

$$P_x u^{n-j+1} = u^{n-j+\tau} := \sum_{i=1}^{m} Q_x^i u^{n-i+j+1}$$

To give an example, for the trapezoidal rule we obtain $\mathbf{P_x u^{n+1/2} = u^n = Q_x u^{n-1/2}}$ and $\mathbf{P_x u^{n-1/2} = u^{n-1} = Q_x u^{n-3/2}}$. We have

$$u^n = \sum_{i=1}^{m} \mathcal{Q}_x^i u^{n-i+1-\tau} \quad \text{when } j = 0,$$

$$u^{n-j} = \mathcal{P}_x u^{n-j+1-\tau} \quad \text{when } j = 1, \ldots, m.$$

Forward Euler	$u^n = \mathcal{Q}_x u^{n-1}$ $\mathcal{Q}_x u^{n-1} = u^{n-1} + \Delta t \mathcal{L}_x u^{n-1}$
Backward Euler	$\mathcal{P}_x u^n = u^{n-1}$ $\mathcal{P}_x u^n = u^n - \Delta t \mathcal{L}_x u^n$
Trapezoidal rule	$\mathcal{P}_x u^n = \mathcal{Q}_x u^{n-1}$ $\mathcal{P}_x u^n = u^n - \frac{1}{2}\Delta t \mathcal{L}_x u^n$ $\mathcal{Q}_x u^{n-1} = u^{n-1} + \frac{1}{2}\Delta t \mathcal{L}_x u^{n-1}$

Some specific members of the family of Linear Time Stepping Scheme

2. Problem Statement

Let us consider a Linear Partial differential equation of the form.

$$u_t = \mathcal{L}_x u, \quad x \in \Omega, \ t \in [0, T],$$

here L_x is a linear operator and $u(t, x)$ denotes the latent solution.

Our goal is to predict the solution $u(t, x)$ at later times t > 0 and propagate the uncertainty due to noise in the initial data. Boundary conditions are known to us.

3. Prior

Let us apply the backward Euler scheme to the Burgers' equation. This can be written as

$$u^n + \Delta t u^n \frac{d}{dx} u^n - \nu \Delta t \frac{d^2}{dx^2} u^n = u^{n-1}$$

Similar to the ideas presented here and here, we would like to place a Gaussian process prior on $\mathbf{u}^n$. However, the nonlinear term $u^n du^n/dx$ is causing problems simply because the product of two Gaussian processes is no longer Gaussian. Hence, we will approximate the nonlinear term with $\mu^{n-1} d(u^n)/dx$, where μ^{n-1} is the posterior mean of the previous time step. Therefore, the backward Euler scheme can be approximated by

$$u^n + \Delta t \mu^{n-1} \frac{d}{dx} u^n - \nu \Delta t \frac{d^2}{dx^2} u^n = u^{n-1}$$

Let us make the prior assumption that

$\mathbf{u}^n(x) \sim GP(0, k(x, x'; \theta))$,

is a Gaussian process with $\theta=(\sigma^2_0,\sigma^2)$ denoting the hyper-parameters of the kernel $\mathbf{k}$. This enables us to obtain the following Numerical Gaussian Process

$$\begin{bmatrix} u^n \\ u^{n-1} \end{bmatrix} \sim \mathcal{GP}\left(0, \begin{bmatrix} k_{u,u}^{n,n} & k_{u,u}^{n,n-1} \\ k_{u,u}^{n-1,n} & k_{u,u}^{n-1,n-1} \end{bmatrix}\right)$$

Assuming that

$$u^{n-j+1-\tau}(x) \sim \mathcal{GP}(0, k^{j,j}(x, x'; \theta_j)), \quad j = 1, \ldots, m,$$

are m independent processes, we obtain the following numerical Gaussian process:

$$\begin{bmatrix} u^n \\ \vdots \\ u^{n-m} \end{bmatrix} \sim \mathcal{GP}\left(0, \begin{bmatrix} k_{u,u}^{n,n} & \cdots & k_{u,u}^{n,n-m} \\ & \ddots & \vdots \\ & & k_{u,u}^{n-m,n-m} \end{bmatrix}\right)$$

The information on the boundary of the domain can often be summarized by noisy observations $\{x_n^b, u_n^b\}$ of a linear transformation B_x of u^n,

$$u_b^n := \mathcal{B}_x u^n.$$

4. WORKFLOW AND COMPUTATIONAL COST

1. Starting from the initial data $\{x^0, u^0\}$ and the boundary data $\{x^1{}_b, u^1{}_b\}$, we train the kernel hyper-parameters.
2. Having identified the optimal set of kernel hyper-parameters, we utilize the conditional posterior distribution to predict the solution at the next time-step and generate the artificial data $\{x^1, u^1\}$.
3. Given the artificial data $\{x^1, u^1\}$ and boundary data $\{x^2{}_b, u^2{}_b\}$, we proceed with training the kernel hyper-parameters for the second time-step.
4. Having identified the optimal set of kernel hyper-parameters, we utilize the conditional posterior distribution to predict the solution at the next time-step and generate the artificial data $\{x^1, u^1\}$.
5. Steps 3 and 4 are repeated until the final integration time is reached.

To accelerate training, one can use the optimal set of hyper-parameters from the previous time-step as an initial guess for the current one.

5. Training

The hyperparameters can be trained by negative log marginal likelihood resulting from

$$\begin{bmatrix} \boldsymbol{u}_b^n \\ \boldsymbol{u}^{n-1} \end{bmatrix} \sim \mathcal{N}(0, K),$$

where $\{\mathbf{x}^{n}_{b}, \mathbf{u}^{n}_{b}\}$ are the (noisy) data on the boundary, $\{\mathbf{x}^{n-1}, \mathbf{u}^{n-1}\}$ are artificially generated data and,

$$K := \begin{bmatrix} k_{b,b}^{n,n}(\boldsymbol{x}_b^n, \boldsymbol{x}_b^n) + \sigma_n^2 I & k_{b,u}^{n,n-1}(\boldsymbol{x}_b^n, \boldsymbol{x}^{n-1}) \\ & k_{u,u}^{n-1,n-1}(\boldsymbol{x}^{n-1}, \boldsymbol{x}^{n-1}) + \sigma_{n-1}^2 I \end{bmatrix}.$$

Marginal likelihood provides a natural regularization mechanism.The cost of training is primarily computational, and it is associated with computing the inverse and the determinant of the covariance matrices appearing in the computation of the negative log marginal likelihood. These matrices are symmetric, positive-definite, and typically dense. The computation of their inverse and determinant is commonly done using the Cholesky decomposition, which has an unfavorable cubic scaling with the total number of training points.

5.1 Cholesky decomposition:

The Cholesky decomposition of a positive-definite matrix A, is a decomposition of the form

$$\mathbf{A = LL^*},$$

where L is lower triangular matrix and L^* is the conjugate transpose of L.

6. Posterior and propagating uncertainty

In order to predict $u^n(x^n)$ at the new test point x^n, we use the following conditional distribution

$$u^n(x_*^n) \left\| \begin{bmatrix} \boldsymbol{u}_b^n \\ \boldsymbol{u}^{n-1} \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{q}^T \boldsymbol{K}^{-1} \begin{bmatrix} \boldsymbol{u}_b^n \\ \boldsymbol{u}^{n-1} \end{bmatrix}, k_{u,u}^{n,n}(x_*^n, x_*^n) - \boldsymbol{q}^T \boldsymbol{K}^{-1} \boldsymbol{q} \right),$$

where

$$\boldsymbol{q}^T := \begin{bmatrix} k_{u,b}^{n,n}(x_*^n, \boldsymbol{x}_b^n) & k_{u,u}^{n,n-1}(x_*^n, \boldsymbol{x}^{n-1}) \end{bmatrix}.$$

The data $\{\mathbf{x}^{n-1}, \mathbf{u}^{n-1}\}$ are artificially generated data we have to marginalize them by,

$$\mathbf{u}^{n-1} \sim \mathbf{N}(\boldsymbol{\mu}^{n-1}, \boldsymbol{\Sigma}^{n-1,n-1})$$

To obtain,

$$\mathbf{u}^n(\mathbf{x}^n_*) \mid \mathbf{u}^n_b \sim \mathbf{N}(\boldsymbol{\mu}^n(\mathbf{x}^n_*), \boldsymbol{\Sigma}^{n,n}(\mathbf{x}^n_*, \mathbf{x}^n_*)),$$

where

$$\mu^n(x_*^n) = q^T K^{-1} \begin{bmatrix} \boldsymbol{u}_b^n \\ \boldsymbol{\mu}^{n-1} \end{bmatrix}$$

and

$$\Sigma^{n,n}(x_*^n, x_*^n) = k_{u,u}^{n,n}(x_*^n, x_*^n) - q^T K^{-1} q + q^T K^{-1} \begin{bmatrix} 0 & 0 \\ 0 & \boldsymbol{\Sigma}^{n-1,n-1} \end{bmatrix} K^{-1} q.$$

Now we can use the resulting posterior distribution to obtain the artificially generated data $\{\mathbf{x}^n, \mathbf{u}^n\}$ for the next time step with ,

$$\mathbf{u}^n \sim \mathbf{N}(\boldsymbol{\mu}^n, \boldsymbol{\Sigma}^{n,n})$$

Here, $\boldsymbol{\mu}^n = \boldsymbol{\mu}^n(\mathbf{x}^n)$ and $\boldsymbol{\Sigma}^{n,n} = \boldsymbol{\Sigma}^{n,n}(\mathbf{x}^n, \mathbf{x}^n)$

Throughout this work the location of the artificially generated data is chosen using a uniform sampling strategy. This simple choice is mainly motivated by the fact that it allows us to visit as many locations as possible in our spatial domain while we march in time.

7. Burgers' equation

- We can write 1-dimensional Burgers' equation as:

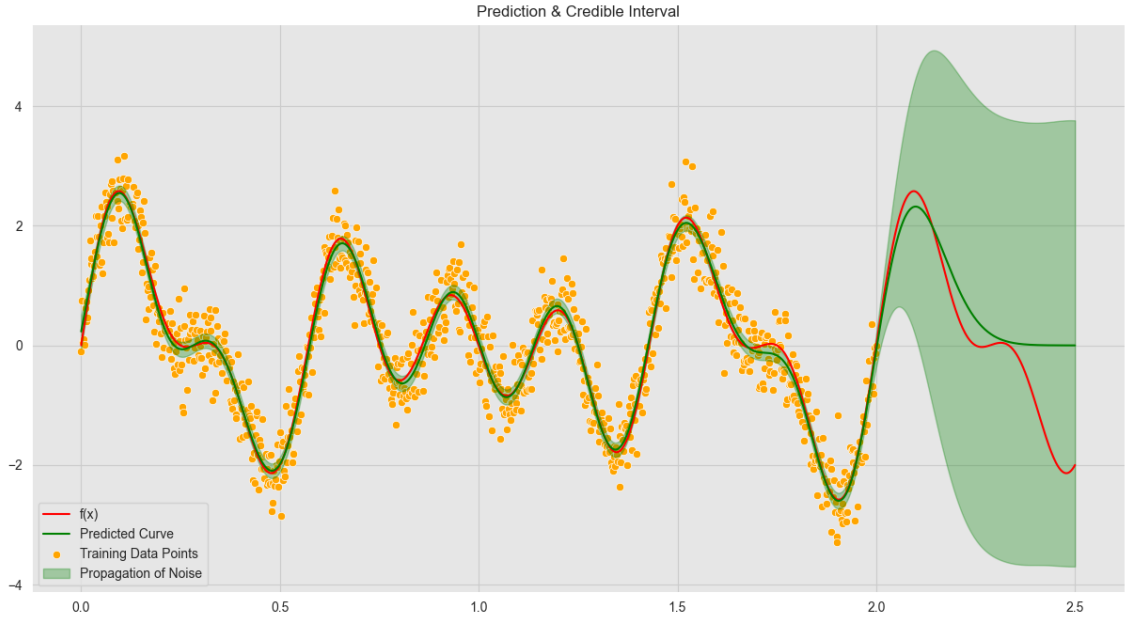
$$u_t + uu_x = \nu u_{xx},$$

- where the boundary conditions are u(t, -1) = u(t, 1) = 0 and u is the unknown solution and v is the known viscosity parameter.
- Applying backward Euler scheme to the above equation we get:

$$u^n = u^{n-1} - \Delta t u^n \frac{d}{dx} u^n + \nu \Delta t \frac{d^2}{dx^2} u^n.$$

- We would like to place a Gaussian process prior on u^n. but we can see there is a non linear term

$$\Delta t u^n \frac{d}{dx} u^n$$



REFERENCES

Journal

1. M Raissi, P Perdikaris, GE Karniadakis, “Numerical Gaussian processes for time-dependent and nonlinear partial differential equations” SIAM Journal on Scientific Computing, vol. 40, no. 1, pp. A172-A198, 2018.
2. M Raissi, GE Karniadakis, “Hidden physics models: Machine learning of nonlinear partial differential equations,” Journal of Computational Physics, vol. 357, pp. 125-141, Mar. 2018.
3. M. Raissi and G. E. Karniadakis, Machine Learning of Linear Differential Equations Using Gaussian Processes, preprint, arXiv:1701.02440, 2017.
4. M. Raissi, P. Perdikaris, and G. E. Karniadakis, Inferring solutions of differential equations using noisy multi-fidelity data, J. Comput. Phys., 335 (2017), pp. 736–746.
5. N. Aronszajn, Theory of reproducing kernels, Trans. Amer. Math. Soc., 68 (1950), pp. 337– 404.
6. J. C. Butcher, Numerical Methods for Ordinary Differential Equations, John Wiley & Sons, 2016.
7. P. Hennig, M. A. Osborne, and M. Girolami, Probabilistic numerics and uncertainty in computations, Proc. Roy. Soc. London A, 471 (2015), 20150142, https://doi.org/10.1098/ rspa.2015.0142.
8. C. E. Rasmussen, Gaussian Processes for Machine Learning, MIT Press, 2006.

URL

1. https://icerm.brown.edu/video_archive/?play=1306
2. https://youtu.be/GXZq2_WYRjo
3. Probabilistic Numerics, discussion forum, http://probabilistic-numerics.org/index.html (2017).