Electronics and Computer Science
Faculty of Engineering and Physical Sciences
University of Southampton

Ashwinkrishna Azhagesh

25/03/2025

An AI Approach to Chaotic Physical Systems:

Project supervisor: **Adam Peugeot**

Second examiner: **David Millard**

Progress report submitted for the award of
Bachelors of Science

Abstract

Empirical laws are mathematical generalisations found through observing the physical world. It has taken us centuries of gathering data, keen research along with repeated experiments, and no doubt plenty of talented scientists to discover these laws. Leading us to understand everything from the mysteries that govern the collision of two objects to the shape of the path planets thread upon.

Recent advances in neural networks including increases in computational power permit us to train models, that replicate, fasten and automate our discovery of empirical laws. This extends to even noisy chaotic systems such as the double pendulum. Combined with white box models, symbolic regression and explanable A.I., we can peer into the ”mind,” of how such models, process data and conclude their observations. Human congition is inherently finite in its capacity for thought and observational ability, has been historically overcome through the development of new tools such as the microscope. Similarly, congitive biases can be mitigated, by utillising artificial intelligence, which is a rapidly emerging technology capable of expanding our perception and analysis.

Statement of Originality

- I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.

- I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.

- I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

You must change the statements in the boxes if you do not agree with them.

We expect you to acknowledge all sources of information (e.g. ideas, algorithms, data) using citations. You must also put quotation marks around any sections of text that you have copied without paraphrasing. If any figures or tables have been taken or modified from another source, you must explain this in the caption and cite the original source.

I have acknowledged all sources, and identified any content taken from elsewhere.

If you have used any code (e.g. open-source code), reference designs, or similar resources that have been produced by anyone else, you must list them in the box below. In the report, you must explain what was used and how it relates to the work you have done.

I have not used any resources produced by anyone else.

You can consult with module teaching staff/demonstrators, but you should not show anyone else your work (this includes uploading your work to publicly-accessible repositories e.g. Github, unless expressly permitted by the module leader), or help them to do theirs. For individual assignments, we expect you to work on your own. For group assignments, we expect that you work only with your allocated group. You must get permission in writing from the module teaching staff before you seek outside assistance, e.g. a proofreading service, and declare it here.

I did all the work myself, or with my allocated group, and have not helped anyone else.

We expect that you have not fabricated, modified or distorted any data, evidence, references, experimental results, or other material used or presented in the report. You must clearly describe your experiments and how the results were obtained, and include all data, source code and/or designs (either in the report, or submitted as a separate file) so that your results could be reproduced.

The material in the report is genuine, and I have included all my data/code/designs.

We expect that you have not previously submitted any part of this work for another assessment. You must get permission in writing from the module teaching staff before re-using any of your previously submitted work for

this assessment.

I have not submitted any part of this work for another assessment.

If your work involved research/studies (including surveys) on human participants, their cells or data, or on animals, you must have been granted ethical approval before the work was carried out, and any experiments must have followed these requirements. You must give details of this in the report, and list the ethical approval reference number(s) in the box below.

My work did not involve human participants, their cells or data, or animals.

ECS Statement of Originality Template, updated August 2018, Alex Weddell aiofficer@ecs.soton.ac.uk

Abstract

I would like to thank my supervisors, Professor Adam Peugeot and Professor David Millard, for all the help and advise I received throughout this project.

Contents

1 Introduction:

1.1 Motivation:

As there is more data being generated than ever before and new experiments, we need a systematic and automatic way to deduce various mathematical patterns and laws in these data. Through the use of symbolic regression we can utilise these data, and in an explainable manner deduce various new physical laws. In this research I have also extended this beyond physics and have applied this to biological data sets which is a novel application of this method. Perhaps extend this beyond or add a sectionsaying this can also be applied to nlp and that it can learn the rules in language and writing etc.

Talk a little about the way this is used outside of this niche use case, and in research, so of course I need to look and research into this.

2 Previous Work:

2.1 Literature Review:

3 Noise:

In this section, I aimed to explore how noise affects the model, and potential ways to mitigate it. Continuing onwards from the previous model, in the data generation step, noise was artificially added, and the results were observed.

So in order to model the noise, I used the python random library, and generated random numbers between 0 and an ever increasing amount of randomness, in oder to guage the accuracy as noise increased for the model. I was also part

3.1 How noise affects the model:

So in order to add noise to the generated data set, I imported in random, and used the randn.int function. In order to vary the inputs, another function was created that incrememntally passes in higher numbers as parameters to the random function, allowing each set of generated data to incrememntally become more and more noisy. Then the symbolic regression model is run on these new data sets, and the resulting equations levels of noise are then plotted in a graph. Furthermore using the Time library to measure the amount of time it takes to run the model as the amount of random error increases.

3.2 How to mitigate noise in data:

Ways to mitigate the noise and it's affects on the model were explored. Functions such as "denoise," in the symbolic regression library helped to some extent. However after a certain point, such methods do not seem to offer much assistance.

I also made my own denoise algorithm. I implemented various different denoise algorithms to see what effects they had. Firstly I implemented a simple moving avergae as a way to mitigate the noise in the dataset. reword this -¿ " Simple and fast, smooths data well by averaging neighbors. However, it blurs sharp changes and is sensitive to extreme outlier values, pulling the average significantly and distorting the signal."

These were my results, this is the pseudo code, explain the alogrithm.

The second denoise algoirthm I implemented is a median filter, and this is what effects it has, and this is how i implemented it. Insert Pseudo code. reword: "Excellent at removing spikes and preserving edges better than averaging. Less affected by outliers. Can sometimes slightly distort the overall shape of the signal, especially with large window sizes."

Finally this is the third algorithm that I had implemented for denoising. Wavelet Denoising, this is the effects, and this is the pesudo code. Reword this -¿ "Transforms data to isolate noise, preserving both smooth and sharp signal features effectively. More complex to understand and requires careful selection of wavelet type and parameters for optimal results, which can be tricky."

3.3 Modelling the noise:

4 Writing my own symbolic regressor from scratch:

4.1 The core;

The core and essential part of any symbolic regression model, lies in the way it at the simplest level, generates and traverses the search tree of possible equations and expressions that may fit the data presented to it.

In order to save time, and to test if my expression generation was working as intended, i have started with simple 2 variable equaitons, and also pass in the speific operations used in the equation. Furthermore this is also extended to handle constants and more later on.

enter in the pseudo code here.

Then I further improved this, by designing a resursive way to generate these expressions, to allow to generate more robust equations from the given variables. Also this is dynamic, so it can

enter in pseudo code

4.2 exploiting physical properties

The next step is to then start to truncate these generated expressions as much as possible to prune the search tree. One of the ways you can do this is through exploiting the symmertrical property of physical equations and how they are mathematically equivalent. Such as removing duplicate expressions.

This is how I achieved that.
Give pseudo code here.

4.3 Dealing with constants:

Another way i further pruned the amount of expression, is through filtering all the expressions generated through the newer recursive generator, by removing all the expressions that did not contain all the specified variables. This is in order to save further time later on during the evaluation seciton.

Insert in pseudo code:

Then later on i designed it next to work with nested expressions and i wanted it to work with more than one constant.

insert in some code that has changed.

I had then filtered out any of the expressions generated that did not have both of the constants.

4.4 Loading data:

Next I needed a way to load the data I has typed up. At this point I was focused on testing as quick as possible and in order to proceed in a prompt manner I made up some dummy data values. Afterwards I made the decision to keep the data as a numpy array, because this will be faster then a text file, there are some various reasons for this, such as numpy arrays being stored in memory, the efficiency of the nderlying data format it is stored in (binary), and finally numpy uses c, and so it vectorises operations, making it far faster.

Insert in pseudo code.

As you can see I check if the number of variables entered matches the shape of the arrary in X, which here is the input data, and y being the target data, as in the final result. Ie x contains the mass and acceleration values, and y is the array of the result of the equation f = ma, so it only contains the value of f in it. This is a basic check to make sure the number of columns all have a corresponding variable.

4.5 Evaluating expressions:

Next I evaluate the expressions that I had generated, and i assign the variables to a column of the data, in increasing order. Then this is substituted into the equation, and the expressions are run, and there is an array of outputs of the expression. This essentially evaluates every generated expressions that has been pruned, and returns a np arrya of the results of those expressions based on the input data.

insert in pseudo code here.

Then like the paper suggested, insert in paper here, I used a medium error description length loss function, and have implemented it in the same way as in the paper. Using error squared, making all the errors positive, and added 1 as a constant to ensure that all the errors are greater than the value, when taking the log.

Insert in pseudo code.

Then furthermore I also implemented 2 other loss algorithms, specifically root mean squared loss as well as mean absolute error.

insert in pseudo code.

This was to help bridge and improve upon the loss algorithm used in the paper, as these two have their own advantages, and a combined hybrid approach seemed smarter.
Explain why later.