SURE

Sheffield Undergraduate Research Experience

University of Sheffield

Faculty of Engineering,
Department of computer science

Genetic programming for causal testing by Luca Devlin and Michael Foster

Background

Computational models are used widely around the world to simulate complex phenomena such as water flow and the spread of disease.

These systems are used to decide real-world policies so it is key that they are accurate. In this experiment, we will be trying to maximise the similarity between a test equation and the equation produced by our program.

Some existing approaches exist for creating these models such as **genetic programming** and **linear regression**.

To help explain these concepts we will approach equations as trees, this allows us to easily produce new random equations, merge certain parts of equations and mutate existing equations. We will also introduce **fitness**, the distance of our estimated equation to our given test one, which is how we measure performance.

Genetic programming is when we produce a population of equations, select the best-performing ones (lowest fitness), mutate and breed them. We continue this for a certain amount of generations where a final selection of equations is given.

During linear regression, we create a population of equations and then give, each part of the tree separated by a +/-, a coefficient to maximise its fitness.

Research questions

RQ1: To what extent does the use of GP for linear regression create a more accurate computational model than pure GP or linear regression?

RQ2: How does changing the size of the data set used affect the accuracy of our computational model?

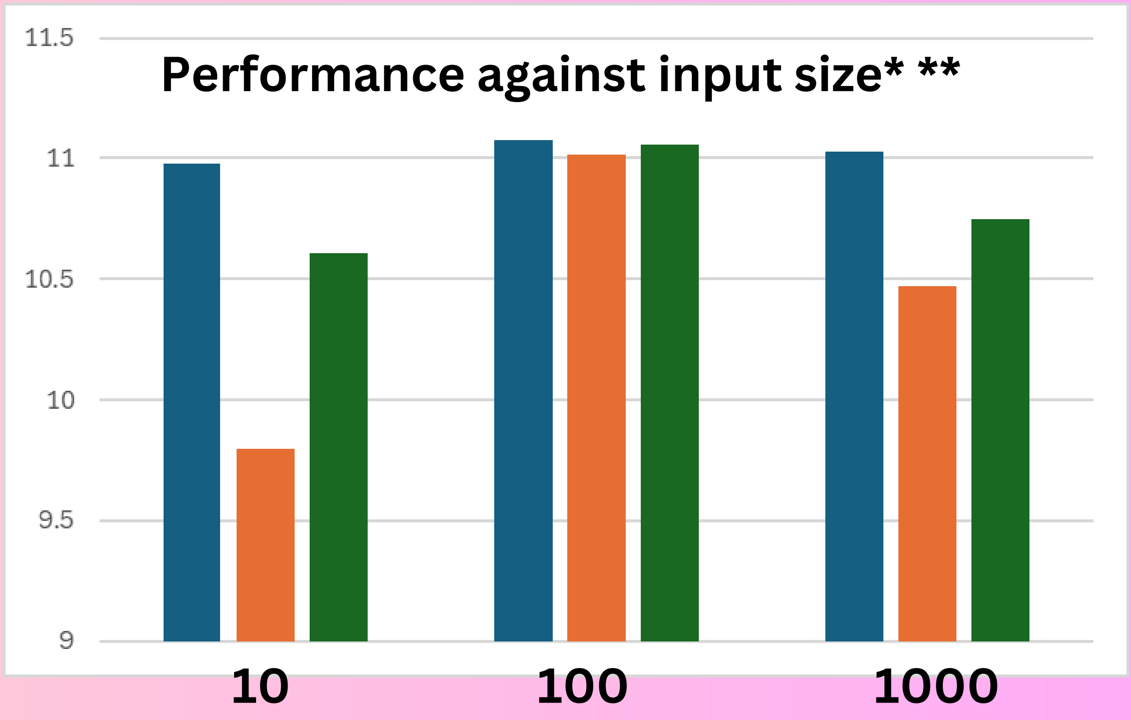
RQ3: How does the use of a noisy data set affect the accuracy of our computational model?

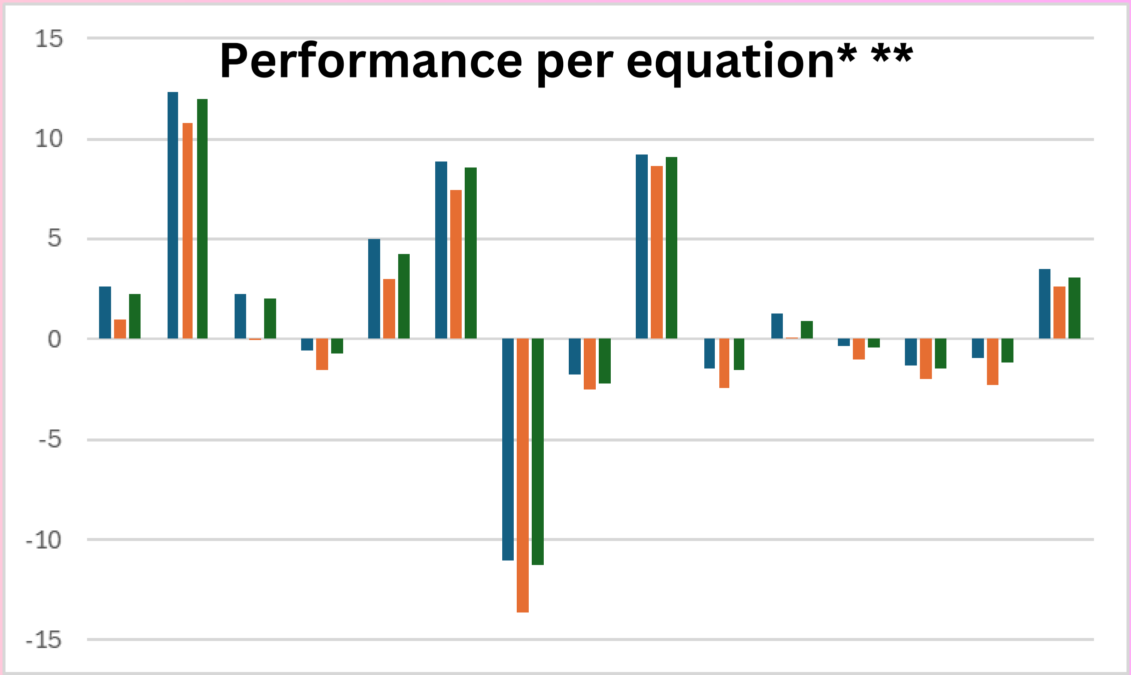
Figures

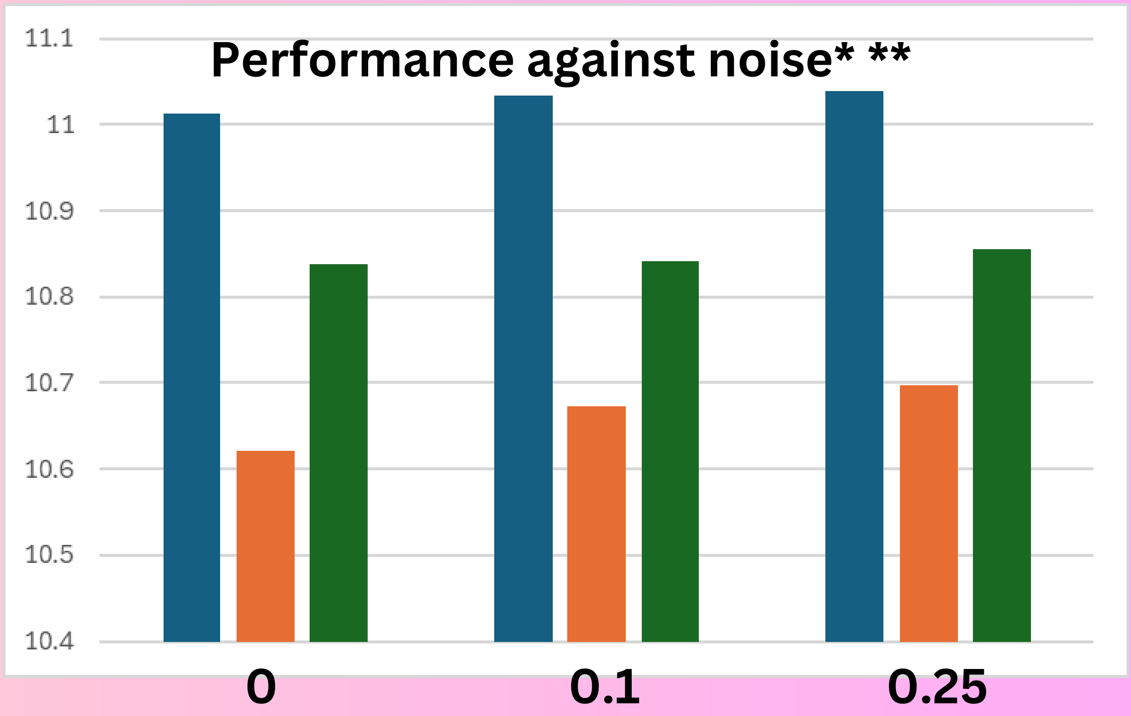
● Linear Regression

● Genetic Programming + Linear Regression

● Genetic Programming


Performance against input size* **


Performance per equation* **


Performance against noise* **

**Lower = better *Log graph

Conclusion

On average, when using genetic programming in combination with linear regression it performed **65** times better than when just using linear regression and **33** times better than when using only genetic programming.

The combination of both methods performed best when there were lower levels of noise and data points but the other two performances didn't change in terms of noise. Interestingly the performance of just using linear regression didn't change with the data size at all whereas the combination of both changed dramatically. The reason is for this is unknown and could be an idea for **future research**.