Researh Proposal
Bayesian Symbolic Regression

David Coba

UvA

Project description

Symbolic regression is a machine learning technique that attempts to find a mathematical expression that describes relationships between features of the data. The mathematical expressions can be arbitrarily complex and are constructed from a prespecified set of features and operators (e.g. addition, multiplication, trigonometric functions, etc.). The main advantage of symbolic regression over other machine learning techniques is that the resulting models are explicit and interpretable. This lack of interpretability is one of the biggest barriers to the use machine learning methods for basic scientific research, but symbolic regression completely bypasses this issue. The goal of this research is to implement an easy to use Bayesian Symbolic Regression framework and to compare its performance with other methods.

The space of possible mathematical expressions is arbitrarily large, and therefore it is not viable to explore it exhaustively. The most common approach to perform symbolic regression is to do a targeted search using evolutionary algorithms, which work by mimicking the evolution of a population of candidate expressions. An example of this approach is the widely used software *Eureqa*[1] (Schmidt & Lipson, 2009). More recently, some researchers have explored the use of neural networks to assist in the search of expressions. Udrescu and Tegmark (2020) use a neural network to discover hidden structure

[1] https://www.creativemachineslab.com/eureqa.html

that is common in physical formulas (e.g. coherent units, symmetry), and they outperform by large margins the previous best efforts. On the other hand, Cranmer et al. (2020) fit neural networks that induce bias about the structure of the target system, and then they recover a mathematical expression with evolutionary algorithms from the networks instead of directly from the target system. This approach has better generalizability and predictive performance than just the neural networks or evolutionary algorithms on their own. The key aspect of these methods is that the neural networks encode prior information about the structure of the target mathematical expression, and therefore encourage a low-dimensional representation. These methods are the current state of the art when it comes to recovering or discovering new mathematical expressions.

A different approach to perform symbolic regression is the use of Bayesian methods. The main feature of these models are the Markov chain Monte Carlo samplers that explore the space of possible mathematical expressions. One of the disadvantages of having to sample from a multidimensional space is that Bayesian methods are usually more computationally demanding than alternative techniques, but in this case they are not necessarily more demanding than their evolutionary or machine learning counterparts. Under these methods we can encode prior information about the components and the structure of mathematical expressions. Guimerà et al. (2020) use the possibilities of Bayesian modelling to incorporate prior information about the frequency of mathematical operators, while Jin et al. (2019) specify a model that is constrained to a linear combination of mathematical expressions. However, as long as we assume a distribution over the residuals, we can specify models with other structures. In this way, we could encode prior information about the structure of the target system that matches our knowledge about how the system behaves, similarly to how Cranmer et al. (2020) do with neural networks. And that is the key of this research. We aim to develop a general way to specify Bayesian symbolic regression models.

[And ideally a package with a nice API for general use if it works nicely.]

Word count: 542/1200

Procedure

[We expect this section to change significantly on the final report. For now it describes the steps that we plan to follow in the following months.]

The first step will be to implement Jin et al. (2019) Bayesian symbolic regression algorithm. Their algorithm is constrained to a linear combination of mathematical expressions represented as symbolic trees.

$$y = \beta_0 + \beta_1 \Psi_1(x) + \cdots + \beta_K \Psi_K(x)$$

Where Ψ_i is the ith symbolic tree that represents a function of the features x. We want to test two possible modifications to their algorithm. First, since a single symbolic tree can capture the default linear combination of tress we want to explore the differences between using K trees versus only using 1 tree. And second, their algorithm generates possible movements for the MCMC sampler from the prior distribution of the parameters. We want to test if there is any benefit from generating proposals from a distribution centered around the current values of the parameters.

We will compare the original algorithm's performance against the performance of the modifications and the performance of a standard evolutionary symbolic regression algorithm[2]. To assess the predictive performance of the different models we will use the variance of the residuals. We will also look at the complexity of the models measured as the number of nodes, the speed to convergence and the stability of the solutions. We will perform the comparisons with data generated from a standard set of functions (Expression 1) that have been used to benchmark other symbolic regression algorithms. Additionally, we plan on using a publicly available data set[3] that we could use to evaluate the interpretability of the expressions produced by the model.

[2]There are two main evolutionary algorithms we could choose. `ExprOptimization.jl`, which is based on the same tooling that we have chosen to use, and `SymbolicRegression.jl`, which is the one developed by the authors of Cranmer et al. (2020).

[3]Possibly a psychological data set instead of one of the examples used in the referenced literature.

$$
\begin{aligned}
f_1(x_0, x_1) &= 2.5x_0^4 - 1.3x_0^3 + 0.5x_1^2 - 1.7x_1 \\
f_2(x_0, x_1) &= 8x_0^2 + 8x_1^3 - 15 \\
f_3(x_0, x_1) &= 0.2x_0^3 + 0.5x_1^3 - 1.2x_1 - 0.5x_0 \\
f_4(x_0, x_1) &= 1.5\exp(x_0) + 5\cos(x_1) \\
f_5(x_0, x_1) &= 6.0\sin(x_0)\cos(x_1) \\
f_6(x_0, x_1) &= 1.35x_0x_1 + 5.5\sin\left[(x_0 - 1)(x_1 - 1)\right]
\end{aligned} \tag{1}
$$

The last step will be to implement a Bayesian symbolic regression algorithm that incorporates information about the structure of the Newtonian Dynamics system described in Cranmer et al. (2020). They use graphical neural networks to encode prior information about the shape of the target mathematical expression. For example, in a model of particles attached to each other by springs, the acceleration that a particle experiences is proportional to the sum of forces between that particle and all other particles in the system. Moreover, the mathematical expression that describes every force should be the same for any pair of particles. In a Bayesian symbolic regression approach we can encode this structure in the following model

$$\vec{a}_i \propto \sum_{\forall j \neq i} F_{ij} \approx \sum_{\forall j \neq i} \Psi_F(x_i, x_j) \ ,$$

where Ψ_F denotes the mathematical expression that the algorithm needs to estimate and x_i is the set of features corresponding to the ith particle. To compare the performance of the Bayesian algorithm versus their approach we plan on using the same system that they use and measure the predictive performance and speed of both methods.

Word count: 480/1000

Intended results

Symbolic regression algorithms perform best when the relationships they are trying to capture can be represented by a sparse mathematical expression. Including prior knowledge about the properties of a system in the model is a way of encouraging such sparsity,

reducing the amount of relationships that the algorithm needs to capture on its own. The main advantages we see of using a Bayesian approach is that it is a general framework that could be used in a multitude in contexts. If the Bayesian models end up performing better than the alternatives, they will be a straightforward alternative that avoids formulating neural networks to incorporating prior knowledge.

Word count: 106/250

Work plan

Time schedule

This internship project consists of 18EC which corresponds to 504 hours of work. Over a period of 21 weeks it averages to 24 hours of work per week.

- February/March: During these months we have reviewed the relevant literature, selected and learned the tooling we are going to use, and designed the structure of the project.

- April: We plan to implement a simple Bayesian symbolic regression algorithm and evaluate its performance.

- May: We plan to implement and evaluate a Bayesian symbolic regression model equivalent to Cranmer et al. (2020) Newtonian Dynamics case.

- June: I will write my internship report. I intend to present the final draft of my internship report on the 18th of June.

If we encounter delays in our planning we could cut down on the number of models to which we compare the performance of the Bayesian symbolic regression algorithm at any step. If it were necessary we could cut the whole comparison with the Newtonian Dynamics case too.

Data storage

We plan on only using either synthetic or publicly available datasets. We are keeping and will keep all project files under version control, with physical and remote daily backups.

Word count: 191/500

References

Cranmer, M., Sanchez-Gonzalez, A., Battaglia, P., Xu, R., Cranmer, K., Spergel, D., & Ho, S. (2020). Discovering symbolic models from deep learning with inductive biases. *CoRR*. http://arxiv.org/abs/2006.11287v2

Guimerà, R., Reichardt, I., Aguilar-Mogas, A., Massucci, F. A., Miranda, M., Pallarès, J., & Sales-Pardo, M. (2020). A bayesian machine scientist to aid in the solution of challenging scientific problems. *Science Advances*, *6*(5), eaav6971. https://doi.org/10.1126/sciadv.aav6971

Jin, Y., Fu, W., Kang, J., Guo, J., & Guo, J. (2019). Bayesian symbolic regression. *CoRR*. http://arxiv.org/abs/1910.08892v3

Schmidt, M., & Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science*, *324*(5923), 81–85.

Udrescu, S.-M., & Tegmark, M. (2020). Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, *6*(16). https://doi.org/10.1126/sciadv.aay2631