Research Internship Report

Bayesian Symbolic Regression

David Coba

St. no. 12439665

Research Master's Psychology

University of Amsterdam

Psychological Methods

Supervised by:

Don van den Bergh

Eric-Jan Wagenmakers

18 June 2021

Abstract

[To be edited] Symbolic regression is a machine learning method that generates explicit mathematical expressions by composing basic functions. Since the models are just mathematical expressions they are fully interpretable, unlike most other machine learning techniques. The goal of this project is to develop and test a general Bayesian symbolic regression framework. The current state of the art in symbolic regression are methods that are able to include information about the structure of the target system they are trying to model. However, they use an approach with neural networks that is convoluted and hard to generalize. We believe that Bayesian methods could be a straightforward alternative to incorporate prior knowledge.

Research Internship Report

Bayesian Symbolic Regression

Introduction

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, interpretable and more generalizable. The goal of this internship was to implement an easy to use Bayesian symbolic regression program and to compare its performance against other methods.

- This lack of interpretability is one of the biggest barriers to the use machine learning methods for basic scientific research, but symbolic regression avoids this issue. -> Rewrite as paragraph introduction
- Use of symbolic regression to assist scientific and complex/dynamical systems modeling
  - Discovering expressions that fill in gaps

Output
  * Explainable models
  * Require less data than traditional deep learning
  * Utilize the prior knowledge about how the system works
  * Mathematical expressions generalize way better than deep neural networks

  - Example from `SciML.jl` Covid spread model or pharmacology one
- Link with the relevance of modeling for psychological / social sciences research

The space of possible mathematical expressions is infinitely large, and therefore it is not viable to explore it exhaustively.

- Explain the symbolic trees representation of an expression

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).

- `SymbolicRegression.jl` (with `PySR` interface) as an open source alternative
- Describe the estimation methods in a bit more detail
  - Islands/archipelago model

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.

- Mention that this is a novel topic and mention that we've only found two previous examples of work done in this line
  - Re-check the literature… again
- Explain conceptually how a sampler can jump between the space of possible trees

- More recent use of neural networks in combination with symbolic regression.

Udrescu and Tegmark (2020) use a neural network to discover hidden structure 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.

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

- Explain in more detail how the graphical network encodes the structure of the 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.

Bayesian methods allow us to encode prior information about the components and the structure of mathematical expressions.

- Explain that we can encode this prior information in a *formal* way
  - e.g. on the complexity of the prior distributions of trees, weights on different operators
  - Guimerà et al. (2020) use the possibilities of Bayesian modelling to incorporate prior information about the frequency of mathematical operators.
- Jin et al. (2019) specify a model that is constrained to a linear combination of mathematical expressions.
- As long as we assume a distribution over the residuals, we can specify models with any structure.
- In this way, we could encode prior information about the structure of the target system that matches our knowledge about how the system behaves

For example, in the model of particles attached to each other by springs described on Cranmer et al. (2020), 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 *i*th particle. This is a less convoluted approach than the use of graphical neural networks to encode this same structure.

- And definitely more flexible.
- Make the point that however this is a form of "restructuring the data" to encode that information, rather than part of the algorithm per se.
- We can use either the evolutionary algorithms or the Bayesian.

Bayesian symbolic regression algorithms are equivalent in scope with the evolutionary algorithms. We could use either of them to recover mathematical expressions from the neural networks used in the approach by Cranmer et al. (2020) or encode that information in the data directly.

- The main advantage of including a neural network step is to increase the number data points that the symbolic regression algorithms use as inputs.
- But the recovered expressions generalize way better than just the neural networks.
- Small mention to another approach is the use of Sparse Identification algorithms.
  - Matrices with values and numerical derivatives of all variables at different time points
  - Represent the matrices in a space of bases that are the active functions
  - Lasso on the coefficients of the bases
  - They require a lot of data
  - They work faster than the evolutionary or Bayesians
  - Ideal to be used on top of the approximations of deep neural networks

Goals of the internship

- Implement Jin's algorithm in a user friendly and computationally fast package

- Modifications to the algorithm
- `BayesianSR.jl`
- Tests how it performs (accuracy + computationally + interpretability)

Bayesian symbolic regression algorithm

- Explain Jin's algorithm in it's current state Jin et al. (2019) Bayesian symbolic regression algorithm is constrained to a linear combination of mathematical expressions represented as symbolic trees:

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

  where Ψ_i is the ith symbolic tree that represents a function of the features x, y is the outcome variable and β are the linear coefficients.

- Describe full model with residuals, OLS, etc
- Describe special case of the linear operators
- Describe tree movements briefly
- Describe tree prior
- Explain the modifications that we are going to test
  - Centered proposals + better ratio calculation
  - Multiple chains
  - Symbolic simplification step
- Full model as an appendix

Procedure

We will compare the original algorithm's performance against the performance of the modifications and the performance of a standard evolutionary symbolic regression algorithm. 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 that we could use to evaluate the interpretability of the expressions produced by the model.

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

- Describe in a bit more detail the comparisons

Results

- Measure of computational speed
  - Jin julia vs Jin python
- Show RMSE progression over time for all versions of the algorithms
- Show distribution of complexity of the expressions over time
- Test run of evolutionary vs Bayesian on the dataset
  - Interpretability

Discussion

- Limitations of the comparisons

    - Unclear things
        * Effects of hyperparameters
        * Comparisons with sparse regression method

- Bayesian symbolic regression as an alternative to evolutionary algorithms
- Is it faster?
- Does it offer more control?
- Are the expressions more generalizable?
- Are the expressions more interpretable?
- The adoption of symbolic regression techniques in general in modeling / prediction use cases.

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