GENETIC PROGRAMMING IN HYPERSPECTRAL IMAGES

Yurong Chen[*]
Hunan University
chenyurong1998@outlook.com

Bingrui Zhao[†]
Hunan University
neuzhaobr@163.com

"Owing to this struggle for life, any variation, however slight and from whatever cause proceeding, if it be in any degree profitable to an individual of any species, in its infinitely complex relations to other organic beings and to external nature, will tend to the preservation of that individual, and will generally be inherited by its offspring." - Darwin

1 Introduction

Automated machine learning (AutoML) is the process of automating the tasks of applying machine learning to real-world problems. AutoML can target various stages of the machine learning process, such as feature engineering, model selection et al. [1]. Grid search, random search, Bayesian optimization, and evolutionary algorithm (EA) are four common approaches to build AutoML systems.

EAs are used to discover solutions to problems humans do not know how to solve, directly [2], in which many optimization problems fall. Due to their random nature, EAs are never guaranteed to find an optimal solution for any problem, but they will often find a good solution if one exists. Genetic Programming (GP) is a type of Evolutionary Algorithm (EA), a subset of machine learning. GP can be used to discover a functional relationship between features in data (symbolic regression), to group data into categories (classification), and et al. [3]. Symbolic regression is one of the most exciting machine learning techniques that aim to identify an underlying mathematical expression that best describes a relationship. It begins by building a population of naive random formulas to represent a relationship between known independent variables and their dependent variable targets in order to predict new data.

Genetic programming is capable of taking a series of totally random programs, untrained and unaware of any given target function you might have had in mind, and making them breed, mutate and evolve their way towards the truth. Think of genetic programming as a stochastic optimization process. Every time an initial population is conceived, and with every selection and evolution step in the process, random individuals from the current generation are selected to undergo random changes in order to enter the next [4].

1.1 Representation

GP seeks to find a mathematical formula to represent a relationship. Say we have two variables X0 and X1 that interact as follows to define a dependent variable y:

$$y = X_0^2 - 3X_1 + 0.5. \tag{1}$$

As a LISP symbolic expression representation which uses prefix-notation, and happens to be very common in GP, as:

$$y = (+(- \times X_0 X_0)(\times 3X_1))0.5). \tag{2}$$

This also can be denoted as a tree-like data as shown in Fig.1.

1.2 Fitness

Now that we can represent a formula as an executable program, we need to determine how well it performs. In a throwback to Darwin, in GP this measure is called a program's fitness. The common error metrics includes mean absolute error for regression, Pearson's product-moment correlation coefficient for indirect optimization, i.e., the output will not directly predict the target. But in real life, things are less easy to quantify.

[*]https://yurongchen1998.github.io

[†]Author contributed equally.

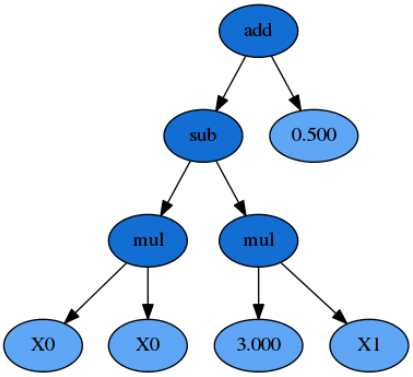

Figure 1: Represent the formula as a syntax tree, where the functions are interior nodes, shown in dark blue, and the variables and constants make up the leaves (or terminal nodes), shown in light blue

1.3 Sufficiency

Another requirement of a successful GP run is called sufficiency. Basically, can this problem be solved to an adequate degree with the functions and variables available. In practice, try to set the constant range to a value that will be helpful to get close to the target.

2 Processes of GP Algorithm

2.1 Initialization

When starting a GP run, the first generation is unaware. These naive programs are a random mix of the available functions and variables and will generally perform.

The first parameter to look at is the **initialization depth** of the programs. Though, the first generation programs may be smaller than this range specifies, so that this initialization depth range only applies to the first generation.

Next, we should consider **population size**, which controls the number of programs competing in the first generation and every generation thereafter.

Finally, we need to decide on the **initialization method**, which can be one of *grow*, *full*, or *half and half* [4].

2.2 Selection

Now that we have a population of programs, we need to decide which ones will get to evolve into the next generation. From the population, a smaller subset is selected at random to compete. The fittest individual in this subset is then selected to move on to the next generation.

2.3 Evolution

We use the fitness measure to find the fittest individual in the tournament to survive. But this individual does not just graduate unaltered to the next generation. They need to be performed genetic operations: *Crossover*, *Subtree Mutation*, *Hoist Mutation*, *Point Mutation*, *Reproduction*. A simple and effective approach to selection involves drawing k candidates from the population randomly and selecting the member from the group with the best fitness.

2.4 Termination

There are two ways that the evolution process will stop. The first is that the maximum number of generations. The second way is that at least one program in the population has a fitness that exceeds the parameter criteria.

Some simple examples are implemented with gplearn library [3].

[3] https://github.com/trevorstephens/gplearn/blob/master/doc/gp_examples.ipynb
https://featureselectionga.readthedocs.io/en/latest/

3 GP in Feature Selection

In real-world problems, not all features are essential since many of them are redundant or even irrelevant [5], which may reduce the performance of an algorithm. Different from feature construction (extraction), feature selection aims to solve this problem by selecting only a small subset of relevant features from the original large set of features and does not create novel features.

Feature selection can be generally divided into two major categories: **filter approaches** and **wrapper approaches** [11]. Filter approaches are utilized to select the subsets of features before the actual model learning algorithm is applied. The best subset of features is selected in one pass by evaluating some predefined criteria independent of the actual generalization performance of the learning machine. They are argued to be computational less expensive and more general, however, might fail to select the right subset of features if the used criterion deviates from the one used for training the learning machine.

Wrapper methods, on the other hand, utilize the learning machine as a fitness function ans search for the best best subset of features in the space of all feature subsets. They generally outperform filter methods in terms of prediction accuracy, but they are generally computationally more intensive.

4 GP in Hyperspectral Imaging

Hyperspectral images have not only spatial dimension information, but also rich spectral dimension information. This feature provides additional reference information for hyperspectral image processing, and can improve the accuracy of classification, detection and recognition. Nowadays, hyperspectral image processing has been widely used in various scenarios, including agriculture[8], military[9], medical[10] and other fields.

GP, as we discussed before, is a powerful ML evolutionary algorithm that can produce readable white-box models. It is proved well-performed in solving an array of problems in different scientific areas. However, GP is still not well known in hyperspectral imagenary [7]. Recently, some works tried to introducing GP methods to this field. [12] introduces Silhouettes score [13] as the fitness function to select ans combine the band.

In processing...

References

[1] https://en.wikipedia.org/wiki/Automated_machine_learning

[2] https://watchmaker.uncommons.org/manual/ch01s02.html

[3] http://geneticprogramming.com/

[4] https://gplearn.readthedocs.io/en/stable/intro.html

[5] B. Xue, M. Zhang, W. N. Browne and X. Yao, "A Survey on Evolutionary Computation Approaches to Feature Selection," in *IEEE Transactions on Evolutionary Computation,* vol. 20, no. 4, pp. 606-626, Aug. 2016.

[6] http://epistasislab.github.io/tpot/

[7] Batista J E, Cabral A I R, Vasconcelos M J P, et al. "Improving Land Cover Classification Using Genetic Programming for Feature Construction." in *Remote Sensing*, 2021, 13(9): 1623.

[8] Gevaert C M, Suomalainen J, Tang J, et al. "Generation of spectral–temporal response surfaces by combining multispectral satellite and hyperspectral UAV imagery for precision agriculture applications," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2015, 8(6): 3140-3146.

[9] Zhang L, Zhang L, Tao D, et al. "Hyperspectral remote sensing image subpixel target detection based on supervised metric learning," *IEEE transactions on geoscience and remote sensing*, 2013, 52(8): 4955-4965.

[10] Lu G, Fei B. "Medical hyperspectral imaging: a review," *Journal of biomedical optics*, 2014, 19(1): 010901.

[11] A. Purohit, et al., "Construction of classifier with feature selection based on genetic programming," *IEEE Congress on Evolutionary Computation*, 2010, pp. 1-5.

[12] J. F. Hernández Albarracín, J. A. Dos Santos and R. d. S. Torres, "Learning to Combine Spectral Indices with Genetic Programming," *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2016, pp. 408-415.

[13] Peter J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, 1987.