# Evolutionary Feature Construction for Symbolic Regression on Black-box Datasets

Hengzhe Zhang
hengzhe.zhang@ecs.vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Qi Chen
qi.chen@ecs.vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Bing Xue
bing.xue@ecs.vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Mengjie Zhang
mengjie.zhang@ecs.vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

## ABSTRACT

Symbolic regression plays a crucial role in modeling real-world datasets. Despite its significance, automatically discovering accurate and interpretable models remains a challenging task. Inspired by recent successes in evolutionary feature construction, we propose a method for building symbolic regression models on black-box datasets. Specifically, the proposed algorithm employs genetic programming to automatically construct features for linear regression models, enabling the construction of interpretable and accurate features with limited computational resources.

## CCS CONCEPTS

• **Computing methodologies** → **Genetic programming**.

## KEYWORDS

Evolutionary feature construction, genetic programming

**ACM Reference Format:**
Hengzhe Zhang, Qi Chen, Bing Xue, and Mengjie Zhang. 2023. Evolutionary Feature Construction for Symbolic Regression on Black-box Datasets. In *GECCO '23: The Genetic and Evolutionary Computation Conference, July 15-19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 THE PROPOSED ALGORITHM

### 1.1 Model Representation

The GP-based feature construction method employed in this paper relies on multi-tree GP. Specifically, each GP individual comprises $m$ GP trees and a linear regression model to represent $m$ constructed features and a linear regression model trained on those features. An example of the model representation is presented in Figure 1. In this example, three GP-constructed features $\phi_1 = x_1 + x_2$, $\phi_2 = x_1$, and $\phi_3 = x_1 * x2$ are used as input to train a linear model and make predictions for unseen data.

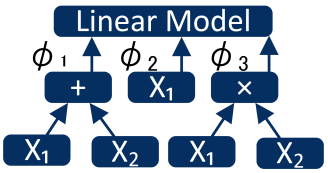


**Figure 1: Genetic Programming for Feature Construction**

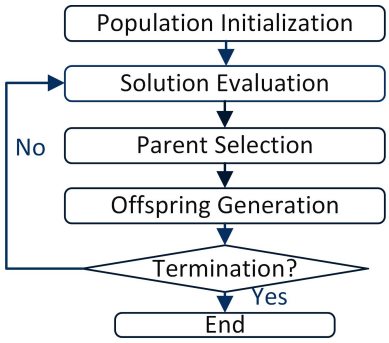


**Figure 2: Workflow of Genetic Programming for Feature Construction**

### 1.2 Algorithm Framework

In this paper, we employ the evolutionary feature construction algorithm, which follows the conventional procedure of the evolutionary algorithm. As shown in Figure 2, the steps include:

- Population Initialization: At the initialization stage, $n$ individuals are randomly initialized. For each GP individual, $m$ GP trees are randomly initialized using the ramped-half-and-half method to represent $m$ high-order features.
- Solution Evaluation: The evaluation stage evaluates the constructed features using a linear regression model. Since the objective of this task is to optimize a set of features to improve the predictive performance of the linear model on


*GECCO '23, July 15-19, 2023, Lisbon, Portugal*


ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

unseen data, leave-one-out cross-validation is used. The fitness function is defined as the squared error on each training instance. For $n$ training instances, the cross-validation losses can be represented by a vector with $n$ instances, i.e., $\mathcal{L}_1, \ldots, \mathcal{L}_n$.

- Solution Selection: Considering that a loss vector is used in the previous step , the lexicase selection iteratively constructs filters based on $min_{p \in P} \mathcal{L}_k(P) + \epsilon_k$, where $k \in [1, K]$ represents a randomly selected task. Individuals with errors greater than this threshold are filtered out. This process continues until only one individual remains, which is then selected as the parent individual. As the filter varies in each round, this selection operator favors selecting elites for different tasks as the parent, thereby promoting the discovery of specialists rather than generalists, ultimately leading to the discovery of better solutions.
- Solution Generation: New sets of features are generated based on selected individuals by applying random subtree crossover and guided subtree mutation [1] to the trees in the selected individuals. For multi-tree GP, $m$ rounds are executed to encourage exploration. In each round, a GP tree is randomly chosen from each individual for crossover and mutation.

## REFERENCES

[1] Hengzhe Zhang, Aimin Zhou, Qi Chen, Bing Xue, and Mengjie Zhang. 2023. SR-Forest: A Genetic Programming based Heterogeneous Ensemble Learning Method. *IEEE Transactions on Evolutionary Computation* (2023).