Improving Interpretability in Symbolic Regression Models Using Multi-objective GP and the Transient Terminal Set

Asher Stout, 300432820

December 15, 2020

1 Overview of Work & Research

Work began on the project with research into previous approaches for improving the interpretability of Symbolic Regression models, and ultimately Multi-objective GP (MOGP) was selected as an area for further study. Several relevant papers defined the interpretability of a solution as the depth or size of its tree representation, which may not accurately correlate to its computational complexity. Thus, it was decided that a new measure of complexity be developed for use in MOGP. In addition, the entirely random nature of mutation operators in MOGP was identified as a place for potential innovation in improving Symbolic Regression model interpretability. This resulted in a new approach being developed called the **Transient Terminal Set**, which is described in further detail in **Section 2**.

Aside from research, extensive time was invested into learning the tools required for the completion of the project objectives - particularly Python, DEAP, and LaTeX. This involved practical applications in the form of tutorials and implementing a standard MOGP and culminated in the formal definition of the Transient Terminal Set algorithm (see **Section 2**) and its implementation using DEAP (see **Section 3**).

The total hours spent on the project, as of 15 December 2020, were **135**.

2 Transient Terminal Set

Algorithm 1 Multi-objective GP using the Transient Terminal Set (TTSGP)

Input: population size ρ, crossover probability p_c, mutation probability p_m, transient mutation probability p_d, terminal set T, function set F, death age α

Define: generation G_n, individual fitness f_i, transient terminal set M_{G_n}, fitness threshold f_{t,G_n}

```
Initialize starting population $P_{G_0}$, $M_{G_0} \leftarrow \emptyset$, $f_{t,G_0} \leftarrow 0$
while no improvement in $\max f_i \in P_{G_n}$ since $P_{G_{n-5}}$ do          ▷ Evolve generation $G_{n+1}$
    $P_{G_{n+1}} \leftarrow \emptyset$, $M_{G_{n+1}} \leftarrow M_{G_n}$
    while len$P_{G_{n+1}} \neq \rho$ do                                          ▷ Update population $P_{G_{n+1}}$
        Perform crossover $\forall i \in P_{G_n}$ with $p_c$
        Perform mutation $\forall i \in P_{G_n}$ with $p_m$, $T$, $F$
        Perform transient mutation $\forall i \in P_{G_n}$ with $p_d$, $M_{G_n}$
        $P_{G_{n+1}} \leftarrow P_{G_{n+1}} \cup \{i | i_{offspring}\}$

    for all subtree $s \in M_{G_{n+1}}$ do                              ▷ Update transient terminal set $M_{G_{n+1}}$
        if $age(s) > \alpha$ then
            Prune $s$ from $M_{G_{n+1}}$
    Compute $f_{t,G_n}$ from $\forall f_i \in P_{G_{n+1}}$
    for $i \in P_{G_{n+1}}$ do
        $f_c \leftarrow \Delta f_i$ from $G_n$ to $G_{n+1}$
        if $f_c > f_{t,G_n}$ then
            $M_{G_{n+1}} \leftarrow M_{G_{n+1}} \cup \{$subtree $s \in i\}$
```

Note: The transient terminal set is utilized during a genetic operation called *transient mutation*, in which a candidate solution is mutated with a member of the set. The transient terminal set is composed of subtrees generated in the population (either through crossover or normal mutation) which have resulted in substantial increases in the fitness of candidate solutions.

The Transient Terminal Set seeks to improve the interpretability of Symbolic Regression models via the use of multi-objective GP by improving the search process itself. By utilizing a complexity measure in addition to an error measure as the Pareto-efficient objectives for the algorithm, and pairing this with the proposed transient terminal set, it is theorized that candidate solutions will become less complex when compared with standard multi-objective GP. As the selection process for the transient terminal set follows this multi-objective framework, improvements in either objective will result in a candidate solution's altered subtree being added to the set. Thus, the transient terminal set distributes proven subtrees that result in lower errors and/or complexities throughout the population. This process potentially results in candidate solutions with both minimized error and complexity measures, and is an improvement over the entirely randomized mutation of standard multi-objective GP.

3 Experimentation Results

Note: The code for this experiment can be accessed at `https://github.com/VeryEager/transient-terminal-gp`, and the data utilized is freely accessible at `https://archive.ics.uci.edu/ml/datasets/wine+quality`.

Figure 1: Evolution using Standard GP

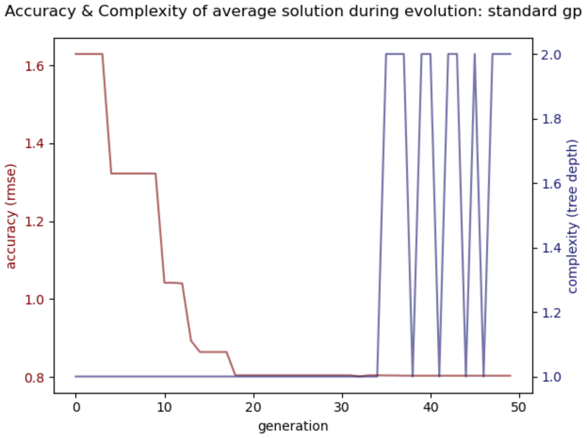


Figure 2: Evolution using Transient Terminal Set GP

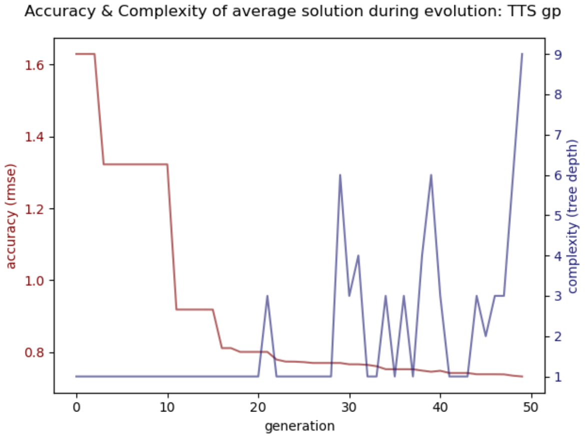


Figure 3: Best individual in Standard GP Evolution

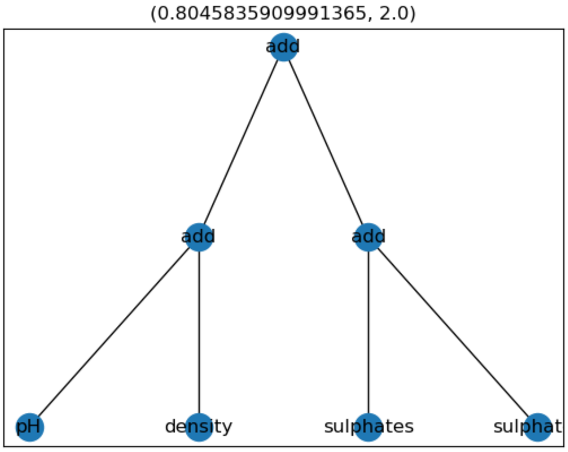
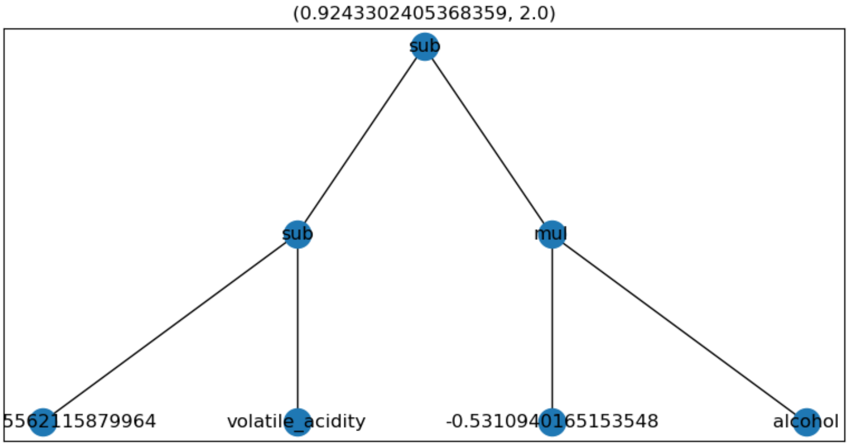


Figure 4: Best individual in TTSGP Evolution



The parameters for the experiment were *generations = 50*, *population = 100*, *crossover probability = 0.5*, *mutation probability = 0.1*, and *transient mutation probability = 0.05*. The results of this experiment indicate that TTSGP is competitive with standard MOGP in terms of accuracy (see **Figures 3 4**), however does not reduce the tree depth of candidate solutions. As shown in **Figure 1** and **Figure 2**, TTSGP achieves a better accuracy in terms of rmse, however the best candidate solutions are approximately five times as complex (on average). Upon investigation I believe this behavior is caused by two factors: the selection process for members of the Transient Terminal Set, which is extremely strict and causes very few subtrees to be identified as improvements, as well as the absence of dynamic mutation probabilities for nodes in a candidate solution, which results in large subtrees being mutated deep into preexisting subtrees.

4 Intended Future Work

While the majority of the Transient Terminal Set algorithm has been implemented in DEAP, there are several noteworthy aspects which will be further developed or implemented to improve the results of TTSGP.

- First, pruning of the set will be fully implemented to ensure only relevant solutions are distributed during transient mutation.
- Second, the criteria for a subtree to be added to the set will be adjusted, as the mean approach is susceptible to outliers. Experimentation will be done to investigate which alternative (median, percentile, etc) is best suited for this problem.
- Third, as mentioned in **Section 1**, a more appropriate measure of model complexity will be created to ensure candidate solutions in TTSGP are actually less complex than those generated in standard GP.
- Fourth, further research and experimentation into the *probability of node insertion* and *probability of a terminal* during mutation will be performed, as these concepts are closely linked with the Transient Terminal Set.
- Fifth, various bugfixes will be performed to ensure the implementation functions as intended.