Transient Terminal Set GP: A New Approach to Improving Interpretability in Symbolic Regression Models

Asher Stout

February 16, 2021

Abstract

NOTE: This is a temporary abstract. A full abstract will be written upon document completion.

I Introduction

In recent years Artificial Intelligence has experienced a rapid rise in interest in data-critical settings, most notably in the finance and healthcare sectors. As the prevalence of AI in the working world continues to expand, so does the need for its models to be understandable to an audience with limited or no exposure to AI and Machine Learning concepts. Regression problems present an additional challenge in that ideal solutions may rely on hundreds or thousands of feature variables and may represent advanced non-linear relationships. Such models are incomprehensible to humans yet are still required in their unaltered forms for domain-specific tasks. Interpretability is fast becoming as indispensable as accuracy in today's AI applications.

Several methods to improve the interpretability of symbolic regression (SR) models have been proposed in recent years, with numerous success stories for both post-hoc explanation techniques and in-model optimizations. Filho et al propose Explanation by Local Approximation (ELA), a post-hoc technique which seeks to explain a single prediction made by a model by computing the importance of each input feature. By aggregating the input feature importances at each data point using ELA, a global feature ranking is produced which shows a strong approximation with the original SR model yet is significantly more interpretable.

Multi-objective Genetic Programming (GP) has been another area of recent scholarship which has produced exciting results for interpretable SR. Kommenda et al define an adapted NSGA-II algorithm which utilizes a Pareto-optimal front to breed with the population and alter the domination strategy itself to reduce the prevalence of single-node solutions. This approach has been proven during experimentation to improve the interpretability of SR models, yet is susceptible to noisy data and can fail to converge to the accuracy of single-objective GP models.

A new multi-objective GP approach is proposed in this report which intends to further improve the interpretability of SR models, dubbed the Transient Terminal Set. The Transient Terminal Set aims to improve SR model interpretability by identifying points of greatest change in the search for global optimums via evolution-dependent terminals used during a third genetic operation called Transient Mutation.

II Previous Research in Interpretable AI

NOTE: Not many papers. Talk to Qi about what to include here.

III Transient Terminal Set GP

The Transient Terminal Set seeks to improve the interpretability of Symbolic Regression models by tracking evolutionary improvements in the population across generations. When a candidate solution undergoes either regular crossover or mutation, the trans-generational fitness change is calculated. Should this difference be a Pareto improvement and result in a significant improvement in at least one fitness value (*solution accuracy* or *solution complexity*) the altered subtree of the candidate solution is added to the Transient Terminal Set.

These subtrees are then distributed among the population via a third genetic operator, Transient Mutation. When a member of the population undergoes Transient Mutation, a randomly-selected subtree is replaced with a member of the Transient Terminal Set. Thus, the Transient Terminal Set distributes proven subtrees that result in Pareto improvements to fitness throughout the population. It is theorized that this genetic operation can result in candidate solutions with both optimal accuracy and complexity measures, and is an improvement over the entirely randomized mutation of standard multi-objective GP.

Transient Terminal Set GP (TTSGP) is described using pseudocode in Algorithm 1, however the parameters it introduces to GP require further discussion.

Not dissimilar to crossover and mutation, the rate at which Transient Mutation is applied to the population is controlled by the *transient mutation probability* (tmutpb). During evolution it is still expected that $CXPB + MUTPB + TMUTPB = 1.0$.

The *lifespan* controls the number of generations valid subtrees remain in the Transient Terminal Set for. Lower values indicate an expectation for numerous, rapid improvements over a short duration, while higher values imply gradual improvements made over an extended period. During experimentation this parameter was left at **5**.

Finally, the *subtree threshold* quantifies the change in solution fitness which is considered significant. This takes the form of a set percentile which is calculated using the fitness changes of the current population. In practice the change in fitness for a candidate solution is calculated as a percent; should either the percentage improvement in accuracy or complexity be above the subtree threshold then the corresponding subtree is added to the Transient Terminal Set.

IV Experiment Methodology

IV.I Experiment Design

Several experiments were performed to accurately determine whether TTSGP exhibits any improvement in the interpretability of SR models over other modern methods. The experiments consisted of measuring the accuracy and complexity of Symbolic Regression trees for TTSGP and several benchmark GP methods at each generation in their evolutions across several regression datasets. For the purposes of the experiments, the accuracy and complexity of a Symbolic Regression tree were defined as the RMSE over the test set and the tree size, respectively.

The GP methods used during experiments consisted of Single-objective GP (SOGP), Single-objective TTSGP (SOTTGP), Multi-objective GP and TTSGP with 50 generations and population size of 500, Multi-objective GP and TTSGP with 250 generations and population size of 200, and TTSGP with optimal parameter settings (see *IV.III*). All methods were written using Python's DEAP library and the Multi-objective methods and TTSGP utilized the NSGA-II algorithm for optimization. The regression datasets were split into 70% training instances and 30% testing instances prior to evolution. Results for the GP methods were averaged across 50 seeds on each dataset. The evolutionary parameters for the experiments are summarized in Table M.

Table 1: Evaluation Experiments' Settings

Generations	50 or 250
Population Size	200 or 500
CXPB	0.8
MUTPB	0.1
TMUTPB	0.1
Subtree Threshold	90

IV.II Datasets

Four common regression datasets were selected for the experiments, and are presented and summarized in Table N. They provide a range of distributions and non-linear relationships for identification in the GP methods.

Table 2: Dataset Information

Dataset	Size	Features
Red Wine Quality	1599	11
White Wine Quality	4898	11
Boston House Price	506	13
Concrete Compressive Strength	1030	8

None of the selected datasets were preprocessed except to remove non-numeric or constant features from the data, and are readily available at the UCI Machine Learning Repository[1]. The preprocessed versions remain accessible via the aforementioned GitHub repository[2].

IV.III TTSGP Parameter Experiments

In addition to the aforementioned experiments, two additional experiments were performed using TTSGP to determine the optimal TMUTPB and subtree threshold. The optimal values of the TMUTPB and subtree threshold were defined as the values within the range [5, 100] with step size = 5 which resulted in equal or lower RMSE than other parameter values while providing the most significant reduction in tree size at the final generation. To ensure the probabilities of the genetic operators did not sum to a value $>$ 1.0 during the TMUTPB experiment, the CXPB was inversely correlated with the TMUTPB value according to the formula $0.9 - TMUTPB$.

The Red Wine Quality dataset was utilized to evaluate the performance of TTSGP at each parameter value. As with the previous experiments, the evolutionary results for each parameter value were averaged across 50 seeds. The identified optimal parameter values were then used during the evaluation of the performance of TTSGP in the original

[1] `https://archive.ics.uci.edu/ml/index.php`

[2] `https://github.com/VeryEager/transient-terminal-gp`

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

```
Input: population size $\rho$, crossover probability $p_c$, mutation probability $p_m$, transient mutation probability $p_d$, terminal set $T$, function set $F$, lifespan $\alpha$
Define: generation $G_n$, individual fitness $f_i$, transient terminal set $M_{G_n}$, subtree 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 $s \in i\}$
```

experiments (see *IV.I*). Table O summarizes the evolutionary parameters for the parameter experiments.

Table 3: Parameter Experiments' Settings

Generations	50
Population Size	500
CXPB (threshold experiment)	0.8
MUTPB	0.1
TMUTPB	[0.05, 0.9]
Subtree Threshold	[5, 100]

V Results & Discussion

V.I Parameter Experiment Results

The results presented in Table Y and Table Z display the highest-accuracy SR model at the 50th generation of evolution averaged across 50 seeds. As discussed in IV.III, the purpose of these experiments was to determine the optimal transient mutation probability and subtree threshold values for use in TTSGP. An optimal value was defined as a value which resulted in equal or lower RMSE than other parameter values while providing the most significant reduction in tree size.

Using these constraints during analysis, the results do not indicate there is an optimal parameter value for both TMUTPB and the subtree threshold. However, there were values for both parameters which resulted in significantly lower tree size while maintaining approximately similar accuracy to nearby values. These identified points were a subtree threshold = **85th percentile** and a transient mutation probability = **0.15**, and as shown in Table Y and Table Z these points approximate the accuracy of nearby solutions while achieving the minimum tree size in their respective experiments. As such, a TMUTPB = 0.15 and subtree threshold = 85 were used as the 'optimal' parameter values during the evaluation experiments.

Table 4: Threshold Experiment Results (RMSE, tree size)

Threshold	Fitness
0.75	(0.784, 10.52)
0.80	(0.777, 10.92)
0.85	(0.786, 7.96)
0.90	(0.780, 8.4)
0.95	(0.778, 8.56)

Table 5: TMUTPB Experiment Results (RMSE, tree size)

TMUTPB	Fitness
0.05	(0.783, 9.84)
0.10	(0.780, 8.4)
0.15	(0.786, 6.92)
0.20	(0.761, 9.36)
0.25	(0.769, 9.52)

NOTE: This section will be further updated once figures have been generated.

V.II Evaluation Experiment Results

The evaluation of TTSGP and the benchmark methods considered both the best and most balanced average individual in the population. The best and most balanced individuals were respectively defined as the individual with the greatest accuracy (RMSE) and the individual which minimized the distance to the best conceivable solution (0, 0). Balanced individuals were only calculated for the multi-objective techniques. The results for these experiments are presented in Tables A-C.

The results of the experiments on the best individual do not indicate substantial improvement in the interpretability of SR models when using TTSGP over MOGP. In terms of the mean best solutions, TTSGP with g=250, p=200, th=85 and tm=0.15 achieved significantly lower tree size across three of the four datasets, however it was unable to compete with the MOGP methods and TTSGP with alternate parameter values in terms of the RMSE. MOGP with g=50, p=500 consistently ranked the highest of all multi-objective techniques in RMSE, and maintained competitive tree size with the other multi-objective techniques. Additionally, TTSGP with g=50 and p=500 was able to maintain accuracy with MOGP, however did not guarantee a reduction in tree size.

Results for experiments on the most balanced individual were similar, and did not indicate any improvement when using TTSGP over MOGP. Despite all TTSGP techniques achieving consistently lower tree sizes, they were unable to maintain or improve the RMSE of the MOGP techniques on any dataset. Concerning the runtime of the experiments, TTSGP was consistently more time-exhaustive than MOGP across all datasets, and in some cases took more time to complete than SOGP.

NOTE: This section will be further updated once figures have been generated. This is required to determine whether the search process was improved when using TTSGP.

VI Conclusions

Transient Terminal Set GP does not show significant improvement in the interpretability of SR models over existing multi-objective GP techniques. While some improvements to interpretability are made in both the best & balanced individual when using TTSGP, this improvement is not consistent between parameter values and datasets, resulting in a mixed perception of improvement. Therefore, I cannot assert with certainty that TTSGP necessarily improves SR model interpretability, or necessarily improves the search process for optimal solutions.

However, further research and refinement can be undertaken to determine whether potential exists in the TTSGP method for improving SR model interpretability. Potential future work includes identifying the optimal *lifespan* of the Transient Terminal Set, or whether a dynamic lifespan linked with the number of entries within the set would enable some improvement in performance. Additionally, the selection process for transient mutation is still entirely randomized. Using a heuristic based on a candidate solution's point of mutation and the original topography of where the mutated subtree was gathered would assist in selecting the member of the Transient Terminal Set with the greatest potential for improvement in fitness.

Table 6: Mean Best Solution at 50th generation (RMSE, tree size)

	Red Wine	White Wine	Concrete Strength	Boston House Price
SOGP	(0.725, 60.12)	(0.809, 60.36)	(10.480, 71.28)	(5.741, 67.64)
SOTTGP	(0.696, 57.84)	(0.795, 61.96)	(9.636, 74.4)	(6.121, 74.32)
MOGP (g50, p500)	(0.78, 9.08)	(0.86, 7.72)	(15.030, 10.64)	(7.330, 9.04)
MOGP (g250, p200)	(0.826, 8.32)	(0.893, 8.84)	(16.784, 11.08)	(8.132, 7.96)
TTGP (g50, p500)	(0.78, 8.4)	(0.866, 9.68)	(16.059, 8.96)	(7.801, 9.4)
TTGP (g250, p200)	(0.826, 8.68)	(0.9, 8.2)	(17.275, 11.4)	(8.131, 7.92)
TTGP (g50, p500, th85, tm0.15)	(0.78, 8.4)	(0.866, 9.68)	(16.059, 8.96)	(7.801, 9.4)
TTGP (g250, p200, th85, tm0.15)	(0.831, 7.72)	(0.916, 8.4)	(17.423, 9.8)	(8.319, 7.68)

Table 7: Mean Balanced Solution at 50th generation (RMSE, tree size)

	Red Wine	White Wine	Concrete Strength	Boston House Price
MOGP (g50, p500)	(0.868, 4.04)	(0.926, 3.56)	(16.540, 4.08)	(8.419, 3.52)
MOGP (g250, p200)	(0.961, 4.08)	(0.981, 3.72)	(19.151, 4.36)	(8.926, 3.64)
TTGP (g50, p500)	(0.995, 3.2)	(1.084, 2.84)	(22.737, 2.4)	(9.078, 2.72)
TTGP (g250, p200)	(1.20, 3.04)	(1.082, 2.8)	(22.898, 3.16)	(10.099, 2.36)
TTGP (g50, p500, th85, tm0.15)	(0.995, 3.2)	(1.084, 2.84)	(22.737, 2.4)	(9.078, 2.72)
TTGP (g250, p200, th85, tm0.15)	(1.097, 3.24)	(1.065, 3.0)	(23.825, 3.2)	(9.671, 2.56)

Table 8: Mean Evolution Time (seconds)

	Red Wine	White Wine	Concrete Strength	Boston House Price
SOGP	147.061	452.09	111.566	56.680
SOTTGP	185.135	537.598	146.086	88.924
MOGP (g50, p500)	63.762	166.265	41.024	26.782
MOGP (g250, p200)	126.596	333.973	78.214	54.165
TTGP (g50, p500)	84.279	202.852	61.3	46.111
TTGP (g250, p200)	311.537	511.121	268.446	240.753
TTGP (g50, p500, th85, tm0.15)	84.279	202.852	61.3	46.111
TTGP (g250, p200, th85, tm0.15)	321.328	522.578	279.706	259.92