Using Feature-based Fitness Evaluation in Symbolic Regression with Added Noise

Janine H. Imada
Brock University
Department of Computer Science
500 Glenridge Ave.
St. Catharines, ON, Canada
ji02fd@brocku.ca

Brian J. Ross
Brock University
Department of Computer Science
500 Glenridge Ave.
St. Catharines, ON, Canada
bross@brocku.ca

ABSTRACT

Symbolic regression is a popular genetic programming (GP) application. Typically, the fitness function for this task is based on a sum-of-errors, involving the values of the dependent variable directly calculated from the candidate expression. While this approach is extremely successful in many instances, its performance can deteriorate in the presence of noise. In this paper, a feature-based fitness function is considered, in which the fitness scores are determined by comparing the statistical features of the sequence of values, rather than the actual values themselves. The set of features used in the fitness evaluation are customized according to the target, and are drawn from a wide set of features capable of characterizing a variety of behaviours. Experiments examining the performance of the feature-based and standard fitness functions are carried out for non-oscillating and oscillating targets in a GP system which introduces noise during the evaluation of candidate expressions. Results show strength in the feature-based fitness function, especially for the oscillating target.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms

Keywords

genetic programming, noisy signals, symbolic regression

1. INTRODUCTION

Symbolic regression is an established genetic programming application which evolves mathematical expressions with the goal to produce a pre-specified target behaviour. In this algorithm, a fitness function is employed to evaluate how well each candidate expression matches the target. For this particular task, fitness functions are commonly based on the difference between the value of the dependent variable produced by the candidate expression and the corresponding target value. However, the effectiveness of this fitness evaluation approach can be degraded in the presence of noise.

This paper considers the use of fitness functions based on statistical features of the sequence of values, rather than the actual values themselves. Features lower the dimensionality of the data and can mitigate the disruptive effects of noise. Furthermore, a carefully selected set of features can target specific behaviours and eliminate irrelevant or redundant facets of the data.

In this study, two target expressions, one non-oscillating and the other oscillating, are tested amidst varying levels of added noise. Features considered for use in the fitness function are drawn from a rich set of features including those that characterize oscillating behaviour. Results are compared to similar runs which use a fitness function based on the standard GP approach. This work serves as a preparatory step in ongoing research that employs GP to infer expressions for a stochastic language.

Section 2 outlines related work and provides a brief description of genetic programming. Section 3 describes the feature-based fitness function in detail. Section 4 outlines the experiments performed, with the results presented in Section 5. Conclusions and suggestions for subsequent work are found in Section 6.

2. BACKGROUND

2.1 Related Work

Much work has been carried out in the areas of feature-based search spaces and symbolic regression with noisy data. Only a cursory review has been performed, and representative papers are cited.

Features have been used to define search spaces in machine learning tasks such as data mining, signal and image processing, and classification, particularly when noisy signals or considerable amounts of data are involved. The full feature set considered in this study was derived from feature-based approaches to clustering and classification of time series data [7, 10].

Borrelli et al explored the use of multi-objective GP for symbolic regression of noisy time series data in which 2 of 3 objectives were based on combinations of statistical features (mean, standard deviation, skew, kurtosis) [2]. The multi-objective approach improved performance over the standard GP fitness function, permitting a level of noise to be accounted for in symbolic regression.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-131-6/08/07.

Other (non-feature-based) GP approaches to symbolic regression on noisy data have also been investigated. De Falco et al incorporated a machine learning concept, Solomonoff complexity, as a heuristic within their evaluation [3]. An advantage of this approach is that the Solomonoff complexity reduces expression size, and hence bloat that naturally arises in GP experiments. Bautu et al included a variety of random number generators in their function set to account for the noise in the data [1].

GP has also been used for feature discovery, to generate new features which are linear and non-linear combinations of a basic set of features [6, 8, 9]. It is common for these constructed features to be subsequently used for classification purposes.

2.2 Genetic Programming

Genetic Programming is an evolutionary computational algorithm which offers a framework to effectively synthesize programs aimed to produce a targeted behaviour [5]. Programs are constructed as trees with internal nodes selected from a set of basic functions and leaf nodes selected from a set of terminals. Through a series of generations, genetic operators such as crossover and mutation are applied to selected individuals from the population of trees. Since selection favours those which score better fitness values, the population progressively evolves to more closely behave like the target.

3. FEATURE-BASED FITNESS FUNCTION

The feature-based fitness function used in this study first evaluates the expression through evenly-spaced points over a set interval. Features are then determined from the resulting course of values. A fitness score for the evaluation is then calculated:

$$fitness\ score = \sqrt{\sum_{i=1}^{n} \left(\frac{F_{i,target} - F_i}{F_{i,target}} \right)^2}$$

where F is the value of the feature, and n is the number of features.

The error is normalized so that each feature obtains equal weight in the overall score. If the target value is close to zero, then one is added to both the target and calculated feature in order to avoid division by zero errors.

To reduce the effects of noise, the expression is evaluated several times, and the resulting fitness scores are averaged to obtain the overall fitness for the expression.

The feature-based fitness function draws from a set of 17 statistical features (Table 1), based on the work of Wang et al [10] and Nanopoulos et al [7] which dealt with classifying and clustering time series. With consideration towards the specific target at hand, a subset of these features is selected for direct use in the above fitness function. For this study, selection of the subset was based on preliminary experimentation which considered the variation of the features over several evaluations of the target expression with added noise. Favourable features had low coefficients of variation (ratio of standard deviation over mean) and overall fitness scores which had minimal overlap with those of near-target expressions. A formal means of feature subset selection would be more effective and is in development. At the same time, the number of expression evaluations performed per fitness score was also decided upon. The goal was to keep this number low, for run-time considerations.

Table 1: Full Set of 17 Features (tsa: trend and seasonally adjusted)

1.	mean	10.	mean (tsa)
2.	standard deviation	11.	standard deviation (tsa)
3.	skew	12.	skew (tsa)
4.	kurtosis	13.	kurtosis (tsa)
5.	serial correlation	14.	serial correlation (tsa)
6.	non-linearity	15.	non-linearity (tsa)
7.	chaos	16.	trend
8.	self-similarity	17.	seasonality
9.	periodicity		

4. EXPERIMENTS

The experiments involved comparing the performance of the feature-based and standard GP fitness functions for two target expressions. Various levels of noise were added during evaluation of the candidate expressions.

4.1 Target Expressions

The feature-based fitness function was tested on two target expressions, inspired by early work on symbolic regression [5]:

1. non-oscillating: $x^4 + x^3 + x^2 + x$, in the interval $[-1, 1]$
2. oscillating: $1 + \sin 3x$, in the interval $[0, 2\pi]$

The leftmost graphs in Figures 1 and 2 plot the target expressions through their corresponding intervals.

Values for the target features were obtained by evaluating the expression without noise at 201 evenly-spaced points within the interval.

4.2 Added Noise

At each point that the candidate expressions were evaluated, Gaussian noise, $g(0, s)$, was added, where g is a Gaussian random number with zero mean and standard deviation, s. Noise levels considered in the experiments corresponded to standard deviation levels of roughly 2.5% and 5% of the range of target values within the interval considered. To help visualize these quantities, Figures 1 and 2 show sample evaluations of the target expressions with the levels of noise added for the non-oscillating and oscillating targets, respectively.

To demonstrate the effectiveness of the feature-based fitness function in eliminating the effects of oscillation lag which can be introduced by stochastic processes, zero mean Gaussian noise, $g(0, \pi)$ was added to the oscillating function evaluations. This lag was applied in such a manner that the entire oscillating curve was varied horizontally, shifting the entire curve by the same amount.

4.3 Fitness Functions

Each candidate expression was evaluated at 201 evenly-spaced points over the interval. For runs with added noise, 4 evaluations per expression were carried out and the resulting fitnesses were averaged to obtain the final score. The goal of the GP was to minimize fitness, with the lowest attainable score of zero.

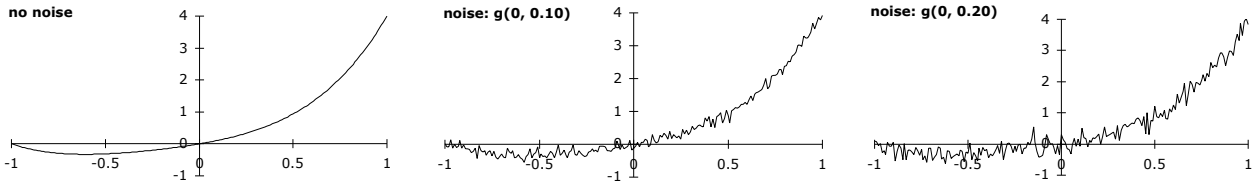

Figure 1: Non-oscillating Target

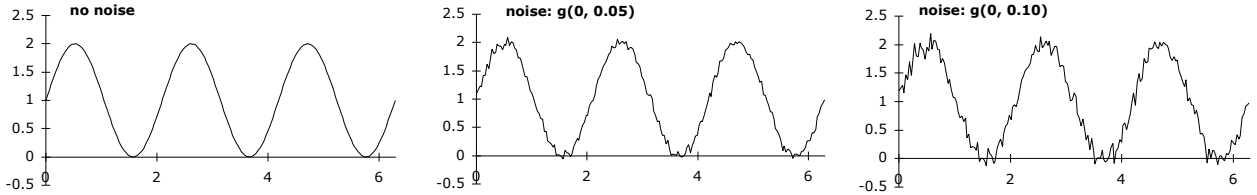

Figure 2: Oscillating Target

Table 2: GP Parameters

population	500
maximum no. of generations	20
probability of crossover	0.9
probability of mutation	0.1
probability of reproduction	0.0
elitism	none
selection	tournament (size 3)
initial population	ramped half and half
min. initial tree depth	2
max. initial tree depth	6
maximum tree depth	17
prob. crossover point is branch	0.9
max. regenerative depth for mutation	5

4.3.1 Feature-based Fitness Function

The feature-based fitness function was described in Section 3. The following feature subsets were used for the targets:

1. non-oscillating: mean, standard deviation, skew
2. oscillating: mean, standard deviation, skew, kurtosis, periodicity, seasonality

4.3.2 Standard GP Fitness Function

A sum-of-absolute-errors approach was used for the standard GP fitness function:

$$fitness\ score = \sum_{i=1}^{201} |f_{target}(x_i) - f(x_i)|$$

where $f_{target}(x)$ is the target expression and $f(x)$ is the expression being evaluated.

Table 3: GP Function and Terminal Sets

	Non-oscillating Target	Oscillating Target
function set	$+, -, *, \%,$ $\sin, \cos, \exp, \ln$	$+, -, *, \%, \sin$
terminal set	x	$x, 1$

Table 4: Non-oscillating Target Results

	Fitness Function	Added Noise	No. Runs Target Found (of 20)	Average Generation Target Found
1	feature	none	5	10.8
2	feature	$g(0, 0.1)$	6	10.3
3	feature	$g(0, 0.2)$	6	10.7
4	standard	none	9	13.0
5	standard	$g(0, 0.1)$	6	12.7
6	standard	$g(0, 0.2)$	5	13.4
baseline	feature	none	0	—

4.4 GP Parameters and Settings

Table 2 lists GP parameters common to all runs, while Table 3 lists the function and terminal sets for each target. % and ln were protected functions to ensure closure.

5. RESULTS

GP runs were performed on open BEAGLE software [4]. For both targets, baseline runs with tournament size 1 were also carried out using the feature-based fitness function. Results are included in the tables.

5.1 Non-oscillating Target

Twenty runs per configuration were executed and the results are shown in Table 4. The best-of-generation fitnesses averaged over all runs are illustrated in Figure 3 for the feature-based fitness function and Figure 4 for the standard

Table 5: Oscillating Target Results

	Fitness Function	Added Noise	Added Lag	Number of Runs Target Found (of 10)	Average Generation Target Found
1	feature	none	$g(0, \pi)$	10	8.8
2	feature	$g(0, 0.05)$	$g(0, \pi)$	10	9.9
3	feature	$g(0, 0.10)$	$g(0, \pi)$	9	8.2
4	standard	none	none	2	5.5
5	standard	$g(0, 0.05)$	none	0	—
6	standard	$g(0, 0.10)$	none	0	—
baseline	feature	none	none	1	6

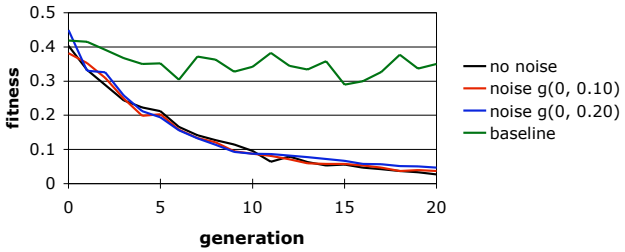

Figure 3: Average Best-of-Population Fitness by Generation for the Non-oscillating Target with the Feature-based Fitness Function

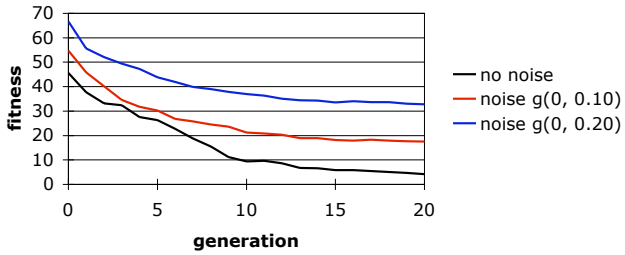

Figure 4: Average Best-of-Population Fitness by Generation for the Non-oscillating Target with the Standard GP Fitness Function

GP fitness function. As the noise levels increased, the GP converged to higher fitness values. This was due to the noise. Interestingly, a few of the feature-based GP runs yielded the mirror image expression, $x^4 - x^3 + x^2 - x$, which received near-zero scores since it exhibited the same characteristics as the target.

Without added noise, the standard GP fitness function was the superior performer for symbolic regression of the non-oscillating target. As noise was added to the candidate expressions, both fitness functions appeared to be performing similarly.

5.2 Oscillating Target

Ten runs per configuration were executed for the oscillating target. It was considered that the target was found if any function in the form $1 + \sin(c \pm 3x)$ (where c is any constant) was constructed. Expressions of this form exhibit the same characteristics. Results are listed in Table 5 and the best-of-generation fitness averaged over all runs is plotted in Figure 5 for the feature-based fitness function and Figure 6 for the standard GP fitness function.

The feature-based fitness function was found to be extremely successful for all noise levels, in contrast with the standard GP fitness function's poor performance, regardless of the varying amounts of significant lag introduced at each evaluation.

6. CONCLUSIONS

The standard sum-of-errors approach to symbolic regression is suitable for noiseless data [5], and in fact may be more efficient than the use of features. However, as shown in our study, when considering noise, performance of the standard evaluation is compromised. Similar results have been shown in [2, 3].

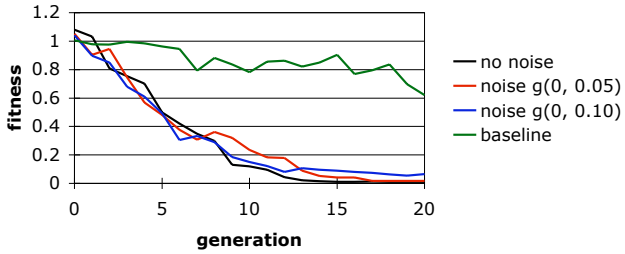

Figure 5: Average Best-of-Population Fitness by Generation for the Oscillating Target with the Feature-based Fitness Function

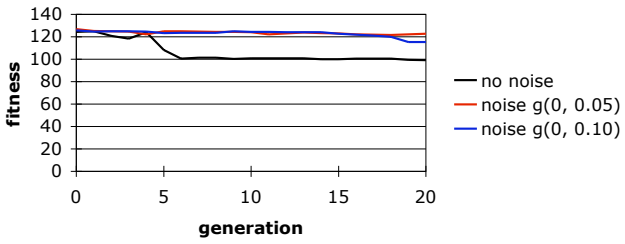

Figure 6: Average Best-of-Population Fitness by Generation for the Oscillating Target with the Standard GP Fitness Function

Results of the experiments demonstrate the ability of the feature-based fitness function to perform symbolic regression in the presence of noise. Particular strength in this fitness function was observed for the oscillating target.

Similar to our study, Borrelli et al applied a small set of statistical feature tests to the symbolic regression of noisy data [2]. We draw from a larger set of statistical features and select some different characteristics (periodicity, seasonality) for our oscillating experiment. The investigation of stochastic processes will require more sophisticated statistical tests, and such behaviours are a subject of ongoing research.

Increased performance may be achieved through application of a rigorous feature subset selection technique to identify a more effective subset for the fitness function. As well, Borrelli et al's use of multi-objective evaluation [2] could possibly lead to improvements over the use of weighted sums, and we are also considering this for the future.

7. ACKNOWLEDGMENTS

Funding support is from an NSERC PGS and NSERC Operating Grant 138467.

8. REFERENCES

[1] E. Bautu, A. Bautu, and H. Luchian. Symbolic regression on noisy data with genetic and gene expression programming. In *SYNASC '05: Proceedings of the Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, page 321, Washington, DC, USA, 2005. IEEE Computer Society.

[2] A. Borrelli, I. De Falco, A. Della Cioppa, M. Nicodemi, and G. Trautteura. Performance of genetic programming to extract the trend in noisy data series. *Physica A: Statistical and Theoretical Physics*, 370(1):104–108, 1 Oct. 2006.

[3] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, and E. Tarantino. Parsimony doesn't mean simplicity: Genetic programming for inductive inference on noisy data. In M. Ebner, M. O'Neill, A. Ekárt, L. Vanneschi, and A. I. Esparcia-Alcázar, editors, *Proceedings of the 10th European Conference on Genetic Programming*, volume 4445 of *Lecture Notes in Computer Science* pages 351–360, Valencia, Spain, 11 - 13 Apr. 2007. Springer.

[4] C. Gagné and M. Parizeau. Genericity in evolutionary computation software tools: Principles and case-study. *International Journal on Artificial Intelligence Tools*, 15(2):173–194, 2006.

[5] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[6] H. S. Lopes. Genetic programming for epileptic pattern recognition in electroencephalographic signals. *Appl. Soft Comput.*, 7(1):343–352, 2007.

[7] A. Nanopoulos, R. Alcock, and Y. Manolopoulos. Feature-based classification of time-series data. In *Information processing and technology*, pages 49–61. Nova Science Publishers, Inc., Commack, NY, USA, 2001.

[8] S. Silva and Y.-T. Tseng. Classification of seafloor habitats using genetic programming. In F. Rothlauf, editor, *Late breaking paper at Genetic and Evolutionary Computation Conference (GECCO'2005)*, Washington, D.C., USA, 25-29 June 2005.

[9] R. Sun, F. Tsung, and L. Qu. Combining bootstrap and genetic programming for feature discovery in diesel engine diagnosis. *International Journal of Industrial Engineering*, 11(3):273–281, 2004.

[10] X. Wang, K. Smith, and R. Hyndman. Characteristic-based clustering for time series data. *Data Min. Knowl. Discov.*, 13(3):335–364, 2006.