Improving the Accuracy and Robustness of Genetic Programming through Expression Simplification

Dale C. Hooper
Space Dynamics Laboratory
1695 N. Research Park Way
Logan, UT 84341
dale.hooper@sdl.usu.edu

Nicholas S. Flann
Department of Computer Science
Utah State University
Logan, UT 84322-4205
flann@nick.cs.usu.edu

Introduction Genetic Programming (GP) is clearly an inductive learning approach because the program discovered must reproduce the behavior of the desired program over all the input space, not just the space represented in the training examples [Koz92e], [Alt94]. When GP uses only consistency with the training examples as a guide, very large (bloated) programs can result that "overfit" the training examples and therefore perform poorly over the complete input space. To avoid this problem, the search process can be biased by applying Occam's razor to prefer simpler, smaller programs that are more likely to reproduce the desired program. This paper introduces a general way of doing this through expression simplification.

Approach Studies have shown that the size of programs in a GP search tend to increase monotonically, due mostly to the application of crossover operations; in particular, where large sub-expressions replace deep sub-expressions [Tac94b]. Much of this complexity is redundant in that sub-expressions are unreachable (introns), or the sub-expressions can be replaced by a much simpler, smaller expression which would compute the same result. This work applies an expression simplifier to identify and eliminate these introns and to rewrite the bloated expressions into simpler, but equivalent expressions. In this way the search process is biased towards producing smaller and simpler expressions.

Expression Simplification Expression simplification is a process of repeatly applying truth preserving transformations to an expression until a fixed point, which tends to reduce its overall size. For the experiments performed here, the simplification system employed over 200 rules. To control the application of simplification in the GP search process a new parameter is introduced, termed the *simplification rate*, that defines the probability that an individual in a generation will be simplified. With this parameter set to 0, we have standard GP.

Empirical Study The primary purpose of this study was to determine whether applying an expression simplifier improved the accuracy and robustness of a GP system, when tested on unseen data. A simple symbolic regression problem was used where a function was to be identified for characterizing limited experimental data. Runs were performed with different simplification rates, from 0 to 1.0, to determine both the effectiveness of the approach and any sensitivity to particular choices of the simplification rate parameter. All experiments were repeated 10 times and the average and standard deviation of results computed.

Results and Analysis The RMS error values measured over the unseen test set for the simplified runs were found to be significantly smaller than the RMS error values for standard GP runs, confirming our hypothesis. No significant increase in run times was noted with the introduction of simplification. Additionally, simplification was found to both prevent monotonic growth in program size and introduce new useful constants into the terminal mix. Furthermore, little sensitivity to the choice of simplification rate was noted, with values between 0.1 and 0.8 working equally well.

Conclusions We have shown that one effective way to implement a simplification bias into the GP search process is to apply an expression simplifier nondeterministically during search. Results demonstrate that simplification reduces program bloating, increases robustness and produces programs that are more accurate over the unseen data.

References

[Alt94] L. Altenberg. The evolution of evolvability in genetic programming. In Kenneth E. Kinnear, Jr., Editor, *Advances in Genetic Programming.* MIT Press, 1994.

[Koz92e] J. Koza. *Genetic Programming: On the Programming of Computers by Natural Selection.* MIT Press, Cambridge, MA, USA, 1992.

[Tac94b] W. A. Tackett. Recombination, Selection, and the Genetic Construction of Computer Programs. PhD thesis, USC, Department of Electrical Engineering Systems, 1994.