N-Dimensional Surface Mapping Using Genetic Programming

David Corney
Computer Science Dept., University College London
Gower Street, London WC1E 6BT, UK
D.Corney@cs.ucl.ac.uk

Ian Parmee
PEDC, University of Plymouth
Drakes Circus, Plymouth PL4 8AA, UK
I.Parmee@plymouth.ac.uk

Abstract

This work introduces an extension to Genetic Programming (GP), known as "GP-UDF" which uses multiple User-Defined Functions (UDFs) to solve surface mapping problems. UDFs are high level primitives, such as polynomials and Gaussian hills, which simplify mapping and aid human interpretation of GP results. Preliminary results show that although UDFs do not improve GP accuracy, they may aid in "landscape classification."

1. INTRODUCTION

A researcher wants to investigate how certain chemicals interact: a series of experiments are performed, mixing various quantities of the chemicals and measuring some feature of the solution, such as acidity. The aim is to discover an equation that relates the measured feature of the solution to the concentration of the chemicals. The overall objective is twofold: firstly, such an equation will allow the results of unperformed experiments to be predicted, and secondly, the equation may provide the researcher with some insight into the behaviour of the chemical system.

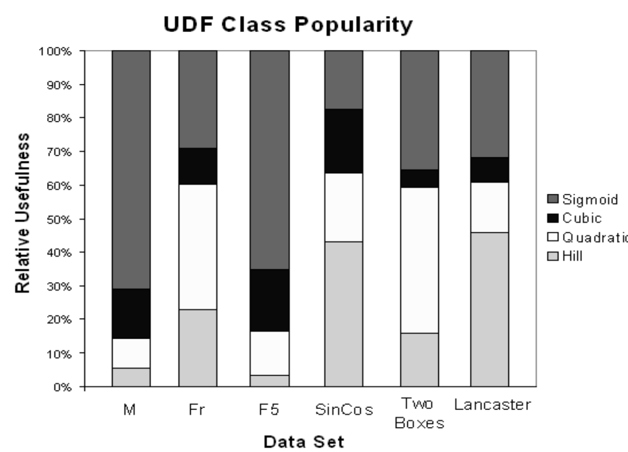
GP has been proposed by John Koza (1992) as an extension to Genetic Algorithms (GA). Although not universally accepted, the Building Block Hypothesis (Goldberg, 1989) describes how GAs work by a form of *problem decomposition*, whereby a large problem is split into a series of sub-problems, each of which is solved separately, with the results then being combined to solve the problem. GP tends to suffer more than GA from epistasis, so building blocks are less likely to form. Attempts to overcome this include ADFs (Koza 1994), Adaptive Representation GP (Rosca and Ballard, 1994), and user-defined functions (Corney and Parmee, 1999).

2. USER-DEFINED FUNCTIONS (UDFs)

Four classes of UDFs were defined: sigmoid, cubic polynomial, quadratic polynomial and Gaussian hill. Each class could have an arity between one a four (i.e. each could take up to four arguments). An instance of a UDF is defined by the class, the number of arguments and the coefficients. Thus the UDF instance "$4x^2$ –3x +2" is of class "quadratic", arity one and has coefficients [4,-3,2]. The coefficients can be found by a number of methods, including a GA using the GP population as an objective function. The instances are then added to the GP function set. Coefficient evolution can continue during the GP run.

3. LANDSCAPE CLASSIFICATION

At the end of the GP run, the number of UDFs of each class can be counted. The graph below shows this relative "popularity" for each of six problems, including three empirical data sets. This analysis could allow problems to be classified according to the basis function best able to model it. For example, problems where GP-UDF favoured sigmoid UDFs (e.g. "M", below) might be best modelled with a multi-layer perceptron, because this uses sigmoid basis functions. This hypothesis needs further verification.



Acknowledgements: We are indebted to Dr Shail Patel of Unilever Research plc, for his illuminating observations as well as for providing several data sets used here.

References

Corney, DPA. and Parmee, IC. (1999) "*n*-Dimensional Surface Mapping using Genetic Programming" Internal Report PEDC-04-99, Plymouth Engineering Design Centre, University of Plymouth, UK. March 1999

Goldberg, DE. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*

Koza, JR. (1992) *Genetic Programming*

Koza, JR. (1994) *Genetic Programming II: Automatic Discovery of Reusable Programs.*

Rosca, JP. and Ballard, DH. (1994) "Hierarchical self-organization in Genetic Programming" in *Proc. of the 11th Int. Conf. on Machine Learning*