Endless forms (of regression models)

Darwinian approaches to free-form numerical modelling

James McDermott

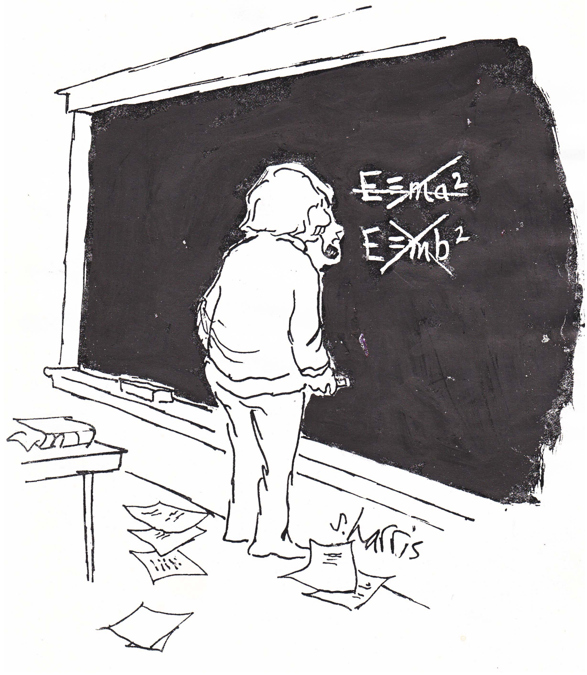
UCD Complex and Adaptive Systems Lab
UCD Lochlann Quinn School of Business
http://jmmcd.net
jmmcd@jmmcd.net

Abstract

Typical regression methods are well-understood, and fitting is fast and reliable. But they require the form of the relationship between variables to be specified in advance. In the field of genetic programming, the symbolic regression task is to both find a model expressing the right relationship, and simultaneously to fit it. It does this using evolutionary search in a large space of arithmetic expressions. In this talk I will introduce the ideas of genetic programming and symbolic regression; describe recent research into hybridising symbolic regression with more typical regression methods; and talk about some applications.

Section 1

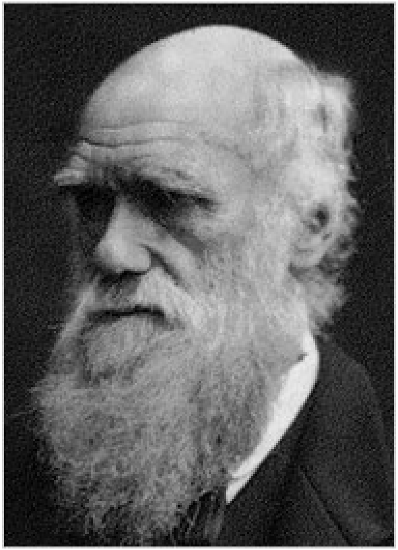
Introduction



E=ma²
E=mb²
S. Harris

Endless forms (of regression models)

"...from so simple a beginning endless forms most beautiful and most wonderful have been, and are being, evolved."



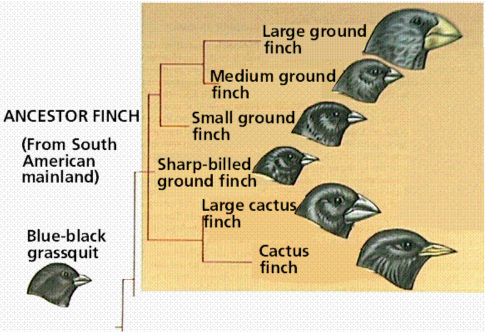
Outline

Section 2

Evolutionary algorithms

Evolutionary algorithms

- Metaheuristic methods of search and optimisation
- Inspired by Darwinian evolution by natural selection

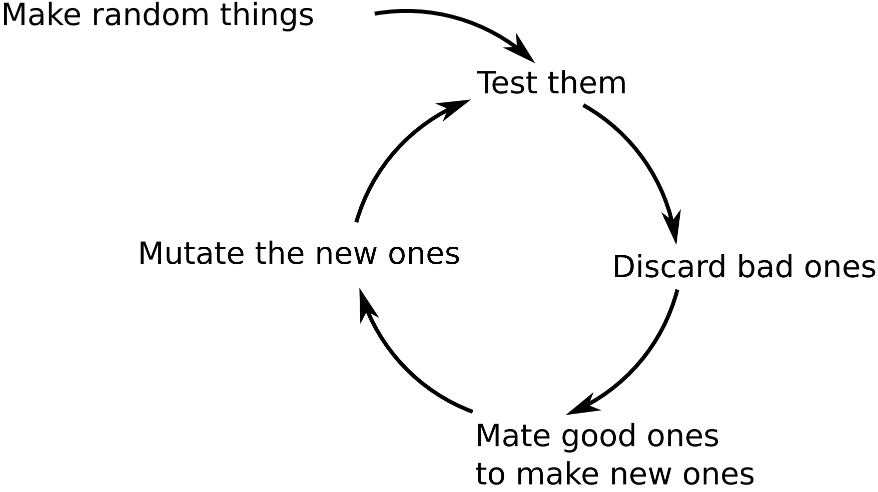


Source: `http://chsweb.lr.k12.nj.us/mstanley`

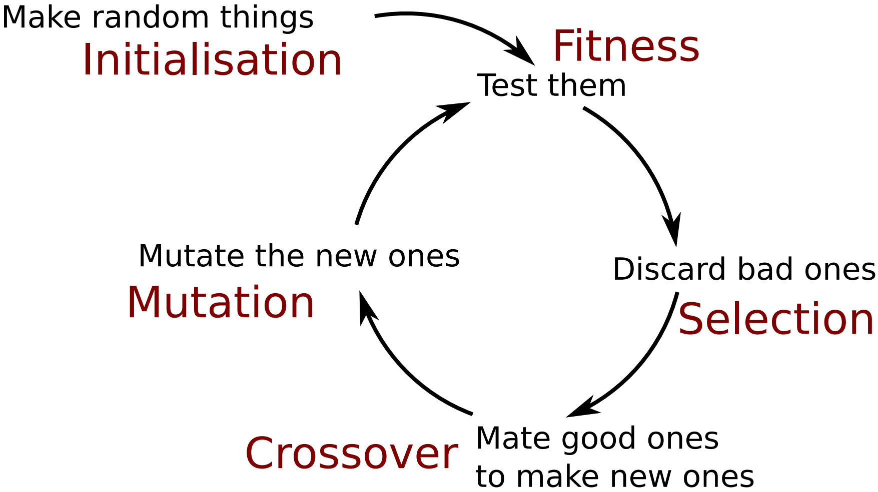
Evolutionary algorithms

- Where a hill-climbing algorithm has only a single solution, EAs have a *population*
- At each iteration, evaluate the whole population using fitness function
- Discard the bad ones
- Mate (recombine, crossover) the good ones
- Mutate the new ones
- Repeat

Evolutionary algorithms



Evolutionary algorithms

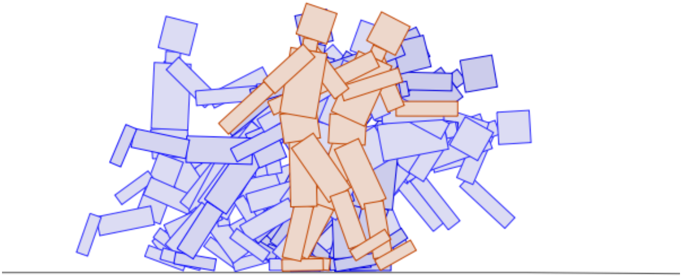


EAs are suitable for black-box search

- Given a set X, find the best $x \in X$
- “Best” is judged by a function $f(x) \rightarrow \mathbb{R}$

EAs are suitable for black-box search

- E.g. x is a robot in a simulated physics engine and f is the distance it walks
- `http://rednuht.org/genetic_walkers/`



EAs are suitable for black-box search

- Searching for coefficients in (eg) linear regression: smooth, easy, NOT black box
- Searching for regression models: black box, both discrete and continuous dimensions, variable number of dimensions, discontinuities in fitness...

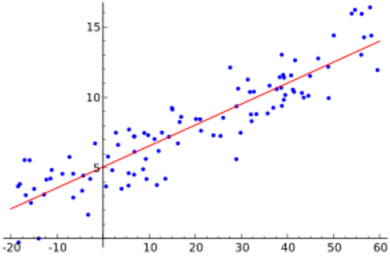
Section 3

Genetic programming and symbolic regression

Genetic programming

- Genetic programming is very ambitious:
- *Automatic programming*
- You say what you want the program to do, the GP system figures out how to do it
- Dates back to 1992 (Koza) or 1950s (Turing)
- Automatic programming is hard :(
- But we've seen some success

Regression

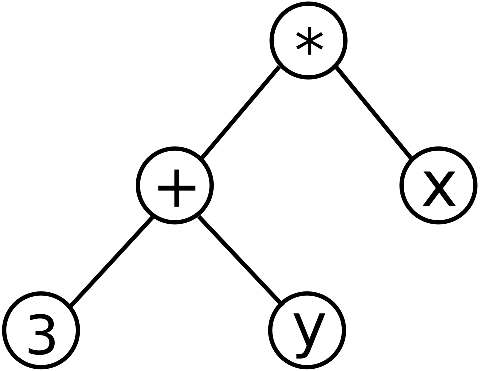


Source: Wikipedia

- The regression problem: given numerical data, find a function that fits the data
- Problem: have to specify model in advance, e.g. linear regression: find a straight line

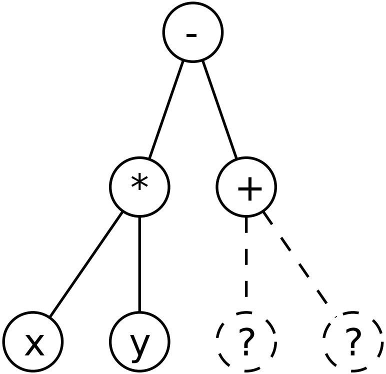
Regression

- *Free-form* regression using genetic programming
- Programs = Functions = Numerical formulae
- E.g. $(3 + y) * x$



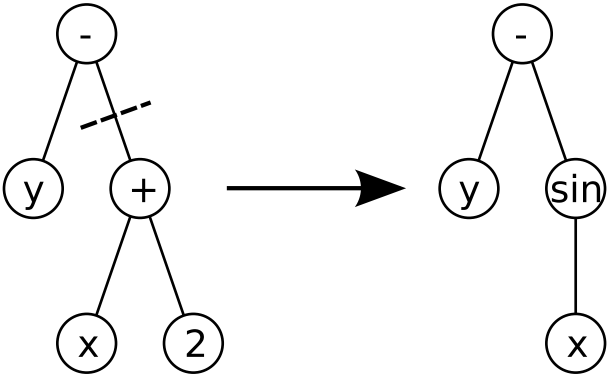
Initialisation

- Make a random program from scratch



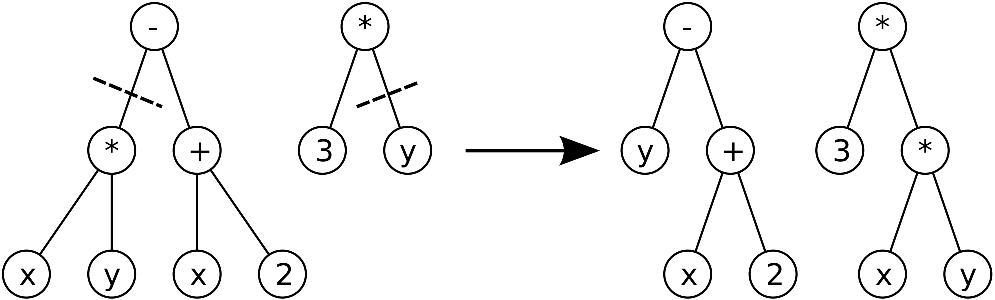
Mutation

- Make a new program by changing an existing one



Crossover

- Make two new programs from two existing ones



Fitness

- Root mean square error RMSE against the data

GP advantages

- Robust fitting
- Multiple solutions in final population, can be used in ensembles
- Readable models, cf. neural networks

GP: black magic that doesn't always work

GP disadvantages

- A lot of hyperparameters to think about
- No guarantee of success
- Many runs needed for confidence
- Bloat: results are often huge, unreadable programs
- Over-fitting

Section 4

Modern approaches

Some (partial) solutions

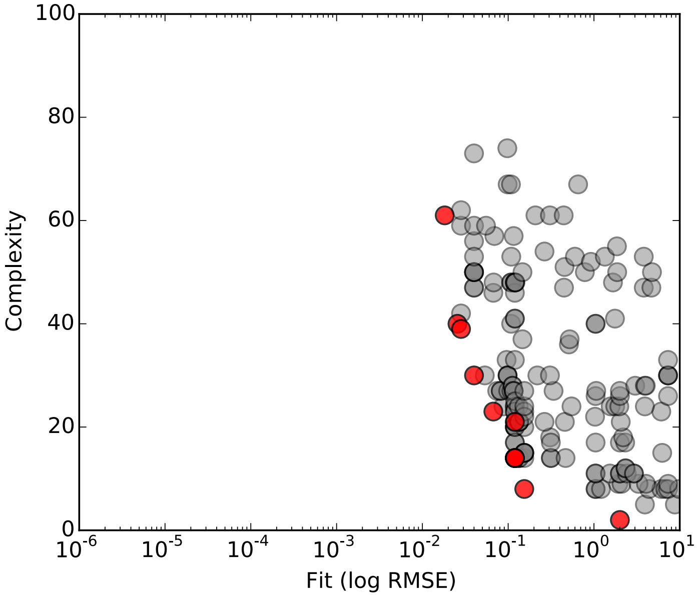
Ingredient 1: Multi-objective optimisation

- Instead of just $f(x)$, we have $f_1(x)$, $f_2(x)$, ...
- In GP, we might have $f_1 =$ RMSE and $f_2 =$ function complexity
- For both, lower is better (simple models are more readable and generalise better)
- These objectives are sometimes conflicting

Pareto dominance

- We say x Pareto-dominates y if:
  - $\forall i : f_i(x) \leq f_i(y)$ and
  - $\exists i : f_i(x) < f_i(y)$
- That is, x is strictly better on at least one objective, and at least as good on all others

Pareto front



NSGA-II Algorithm

- NSGA-II is a multi-objective evolutionary algorithm
- Individuals in Pareto front get Rank $= 1$
- Then Rank 1 are removed and new Pareto front is formed, get Rank $= 2$
- Repeat
- Lower ranks are preferred for selection
- "Crowding" is avoided

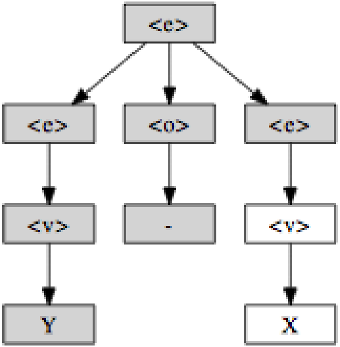
Ingredient 2: optimisation of constants

- Suppose the true model is $3.4x_0^{x_1 - 17.9}$
- Don't try to evolve these constants!
- Evolve the form, then use standard tools to optimise c in $c_0 x_0^{x_1 - c_1}$
- Levenberg-Marquardt (fast, more local)
- Covariance matrix adaptation evolutionary strategy CMA-ES [Hansen et al., 2003] (slower, more global)

Ingredient 3: context-free grammars

```
<expr> ::= (<expr> <bop> <expr>) | <uop>(<expr>) | <var> | <const>
<bop> ::= + | - | * | / | **
<uop> ::= sin | log
<var> ::= X_<varidx>
<varidx> ::= GE_RANGE:n_vars
<const> ::= CC
```

Crossover and mutation on derivation trees



Ingredient 4: simplification of expressions

- Canonicalise and simplify expressions
- Avoid testing the same expression in multiple forms
- Eg $2x$, $x + x$
- Use a symbolic maths system (Maple, Mathematica, Sympy)

Ingredient 5: determinism

- Forget GP, it's too random!
- Just try a limited set of possible expression forms
- Or use a deterministic method to fill a queue prioritised by fitness

Recent approaches

- FFX [McConaghy, 2011]: non-GP, optimisation of constants, ensembles
- PGE [Worm and Chiu, 2013]: non-GP, Pareto, grammars, optimisation of constants, simplification
- Pareto GP [Vladislavleva et al., 2009]: GP, Pareto, optimisation of constants, ensembles
- My version: GP with grammars, Pareto, optimisation of constants, simplification

Section 5

Research in progress

My ingredients

- Grammar specifies search space
- Crossover and mutation on the derivation trees
- Simplification/canonicalisation of expressions
- Optimisation of constants
- Multi-objective (model fit and complexity)

Results

- Nguyen-7 test problem (extra slides)
- LMA beats CMA-ES for optimisation of constants, also 15x faster
- More testing needed

Optimisation of constants

- True model:

$$\log(x_0 + 1) + \log(x_0^2 + 1)$$

- Can we optimise c_0 and c_1?

$$\log(x_0 + c_0) + \log(x_0^2 + c_1)$$

- No! Both Levenberg-Marquardt and CMA-ES fail.

Section 6

Applications

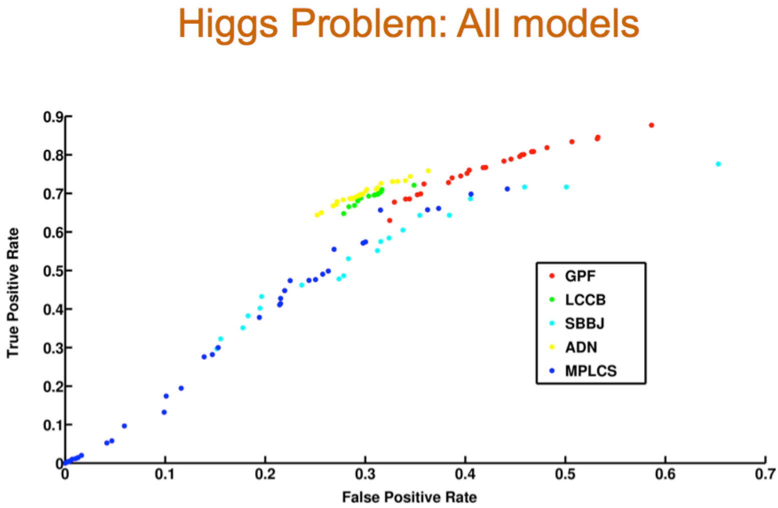
Applications

- Ocean wave modelling Nicolau, [Donne et al., 2014]
- Finance (me, O'Neill, Brabazon, et al)

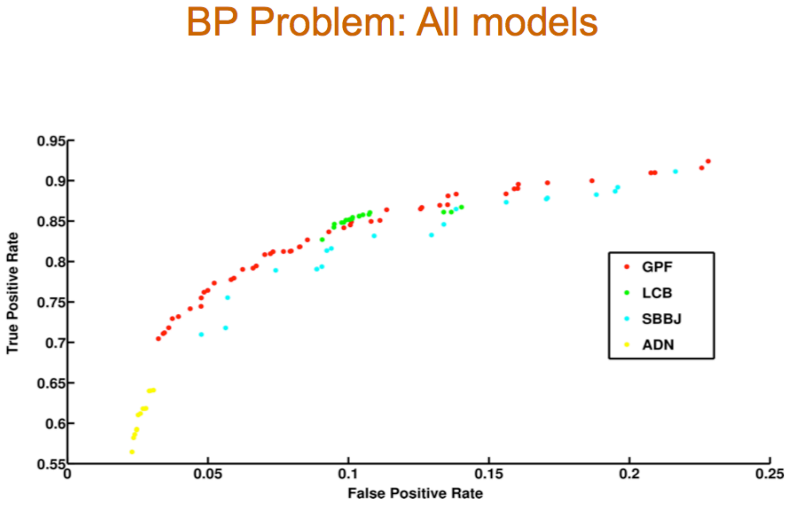
Surprising applications

- A lot of problems can be cast as symbolic regression:
- Various classification datasets: Higgs Boson, Blood Pressure, ...
- Pole-balancing (AI test problem) [Nicolau et al., 2010]
- Generating structured graph structures [D'Ambrosio and Stanley, 2008]
- Graphical art [Sims, 1991, Hart, 2007]
- Music (me)

Higgs Boson classification data



Blood pressure classification data



ANYSCALE LEARNING FOR ALL
© U.M.

Graphical art (Hart)

GP Benchmarks project

- Ongoing effort to improve experimental practices in GP community
- Choice of test problems
- Statistical treatment
- gpbenchmarks.org
- McDermott et al. [2012]

Section 7

Next steps

Next steps

- Use semantics (output) of candidate functions in Pareto selection
- Smarter grammars

Section 8

References

Software

- Thanks: Python, Numpy, Scipy, Scikit-learn, Sympy, CMA

References

David B D'Ambrosio and Kenneth O Stanley. Generative encoding for multiagent learning. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 819–826. ACM, 2008.

Sarah Donne, Miguel Nicolau, Christopher Bean, and Michael O'Neill. Wave height quantification using land based seismic data with grammatical evolution. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 2909–2916. IEEE, 2014.

Nikolaus Hansen, Sibylle Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary Computation*, 11(1):1–18, 2003.

David A. Hart. Toward greater artistic control for interactive evolution of images and animation. In Mario Giacobini, editor, *Applications of Evolutionary Computing*, volume 4448 of *LNCS*, pages 527–536. Springer, 2007. ISBN 978-3-540-71804-8.

Trent McConaghy. FFX: Fast, scalable, deterministic symbolic regression technology. In *Genetic Programming Theory and Practice IX*, pages 235–260. Springer, 2011.

James McDermott, David R. White, Sean Luke, Luca Manzoni, Mauro Castelli, Leonardo Vanneschi, Wojciech Jaśkowski, Krzysztof Krawiec, Robin Harper, Kenneth De Jong, and Una-May O'Reilly. Genetic programming needs better benchmarks. In *Proceedings of GECCO 2012*, Philadelphia, 2012. ACM.

Miguel Nicolau, Marc Schoenauer, and Wolfgang Banzhaf. Evolving genes to balance a pole. In *Genetic Programming*, pages 196–207. Springer, 2010.

Karl Sims. Artificial evolution for computer graphics. In *SIGGRAPH '91: Proceedings of the 18th annual conference on computer graphics and interactive techniques*, pages 319–328, New York, NY, USA, 1991. ACM. ISBN 0-89791-436-8. doi: http://doi.acm.org/10.1145/122718.122752.

E.J. Vladislavleva, G.F. Smits, and D. Den Hertog. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. *Transactions on Evolutionary Computation*, 13(2):333–349, 2009.

Tony Worm and Kenneth Chiu. Prioritized grammar enumeration: Symbolic regression by dynamic programming. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 1021–1028. ACM, 2013.

About me

- University of Limerick PhD 2003-2008
- CASL post-doc (Natural Computing Research and Applications group) 2008-2010
- EvoDesignOpt (now ALFA) group, CSAIL, MIT 2010-2012
- Lochlann Quinn School and CASL, 2012-present
- Research interests: evolutionary algorithms, representations, evolutionary music, analytics
- `http://jmmcd.net`