A non parametric data mining & predictive modelling tool based on genetic programming

Stuart Webb

'The challenge is to pirouette beyond informal pathway pathway diagrams to formal models that represent biological processes in a precise mathematical or computational form'

**Nat Goodman – cofounder Whitehead-MIT Centre for Genome Research**

The ideal analytical methodology

- Specific or highly selective
- Precise, Accurate, Reproducible
- Rapid, Sensitive, Non-destructive
- Low cost, Reagentless / Probes biologically inert
- Robust equipment
- Easy to set up and calibrate
- Capable of axenic operation
- Signals linear with determinant concentration
- Global in scope (for 'omics methods); no prejudgement of 'the answer'
- User-intelligible output ⟵

Modern analytical methods

Require both advanced instrumentation capable of <u>delivering</u> high-dimensional data in a robust manner, and…

…the advanced and <u>intelligent</u> computer-based methods necessary for turning the data into knowledge

2-D
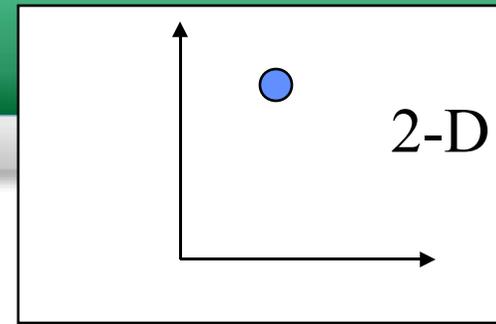
Multivariate data

Multivariate data result from measurements, or observations (variables), of many different characteristics of each of a number of individuals (objects). Incorporate CHAOS (remember Benoit Mandelbrot) !

If there are *n* variables, each object represents a unique position in multi-dimensional hyperspace.

Modern <u>high-resolution</u> techniques (such as expression profiling "omics" methods) purposively produce data with 100s of dimensions, and very large data sets.
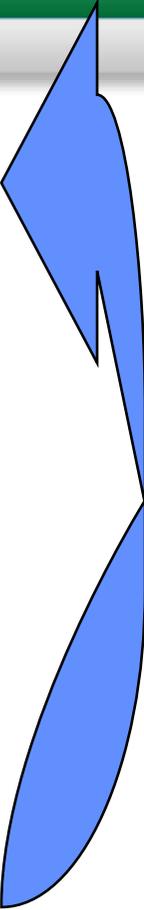
The combinatorial optimization problem

- Making a predictive model using <u>n</u> x-variables to predict just 1 y-variable gives 2^n models in which each one is used or not, <u>before we even parametrise it</u>, which is OK…but….
- ...if n = 100, $2^n = 2^{100} \sim 10^{30}$; the lifetime of the Universe in seconds $\sim 10^{17}$…..
- And then each variable can take just 10 values this is 10^n, etc…
- Machine learning methods are designed to search these huge spaces effectively

The combinatorial optimization problem

... the number of combinations if we only allow it to use 1,2,3,4 or 5 variables is just 100, 4950, 1.6×10^5, 3.9×10^6 and 7.5×10^7. These are much more tractable numbers, and are also likely to provide comprehensible explanations

Some chemometric and related methods

Unsupervised
Just work on x-data

- Principal Components analysis
- Clustering methods
- Kohonen neural networks

- Canonical variates analysis
- Genetic Algorithms
- **Genetic programming**
- Classification & Regression trees

Supervised
use y-data too

Back-prop neural networks
Partial least squares regression

Discrimiant Function Analysis
Inductive Logic Programming

Tools based on a branch of machine learning

Definition of Machine Learning

Coined by Samuel in 1959, the term "machine learning" was the name given to the field of study that gives computers the ability to learn without being explicitly programmed.

Another definition reflecting today's large databases is: ML uses the computer to find unexpected relationships among the many variables in a database without being explicitly programmed.

Genomic computing – evolves solutions in the form of rules by Darwinian methods of natural selection

Genetic programming
Genomic Computing

<u>Evolves</u> by Darwinian rules of natural selection models which provide simple and comprehensible explanations of complex phenomena from huge data sources
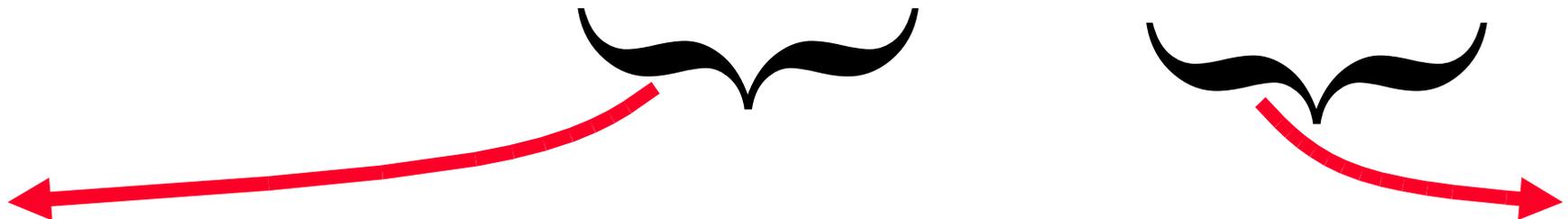
The effectiveness of each model is its 'fitness', and fitter rules are favoured over less fit ones.

GP <u>simultaneously</u> combines the best of other approaches such as rule-based, statistical and neural computing methods
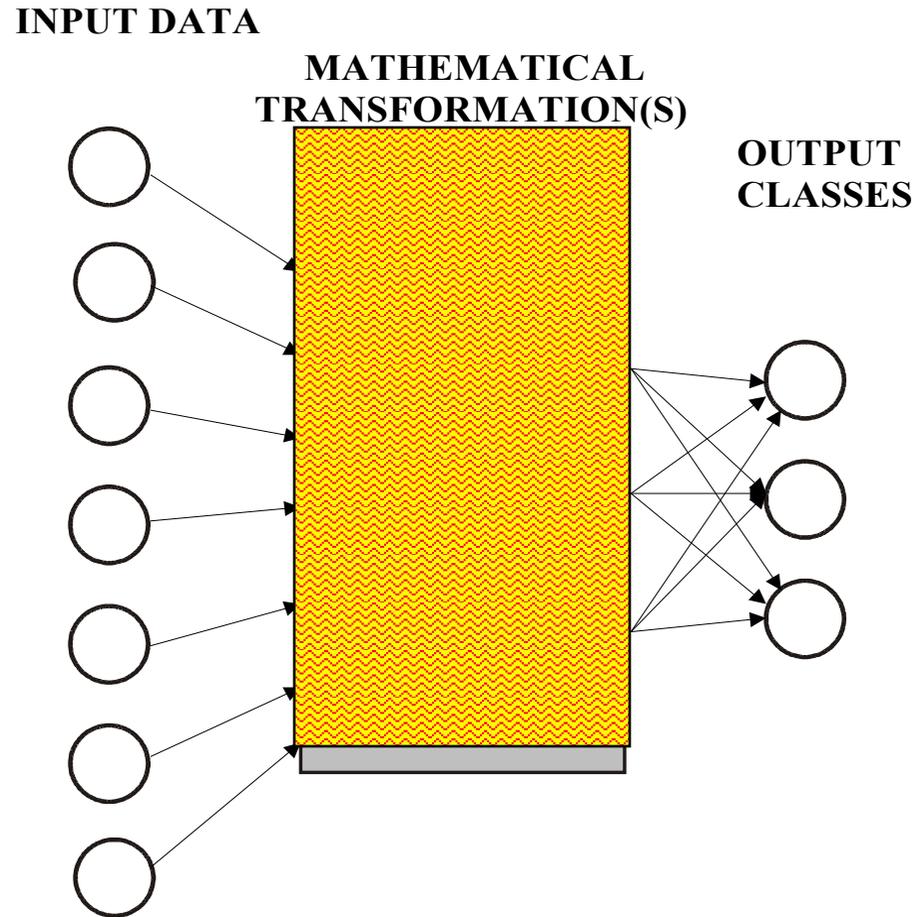
Accepted structure of propositional systems

Variables going across in different columns

Objects going down in different rows		X-var 1	X-var 2	X-var 3	Y-var 1	Y-var 2
Sample 1						
Sample 2…						

The machine learning paradigm

INPUT DATA

MATHEMATICAL
TRANSFORMATION(S)

OUTPUT
CLASSES

The machine learning paradigm
The Genotype-Phenotype (non-linear) mapping problem

Evolutionary computing: The way forward ! (subsets include Genetic Algorithms & Genetic Programming)
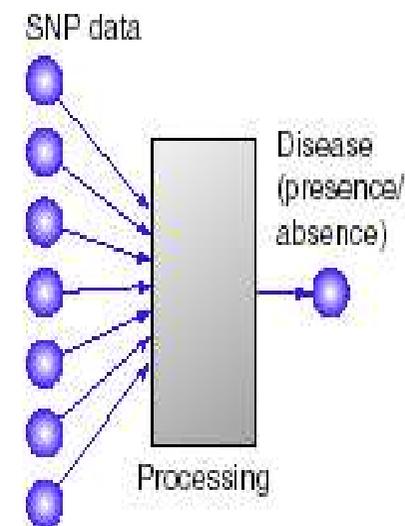
A population of individuals, each encoding a particular solution to a problem

A 'fitness function', by which we can evaluate how good that solution is

A selection strategy for determining who contributes to the next generation

Introduction of genetic diversity by e.g. mutation and recombination

A stopping criterion.

ALGORITHM CYCLES THROUGH STEPS 1 TO 4 UNTIL 5 IS SATISFIED
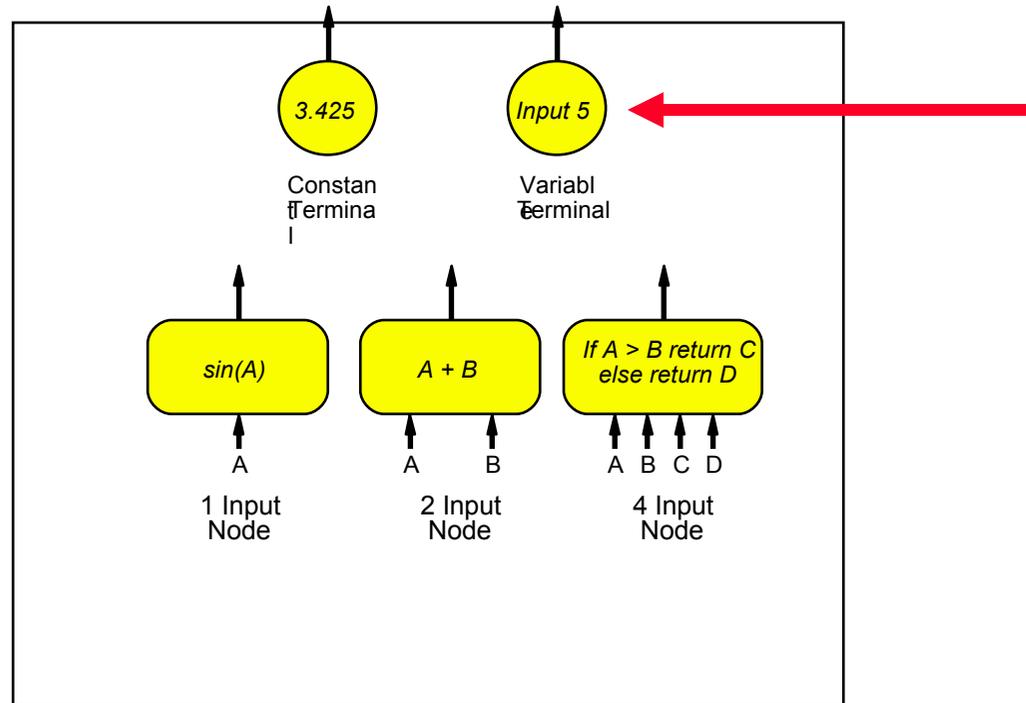
What does genetic programming do ?

TheGmax performs 'supervised learning'.

It learns relationships from examples you provide.

You can use those relationships to interpret new data of similar characteristics.

Alternatively, the relationships may already contain the knowledge you were seeking (such as which are the important variables).

GP/GC BUILDING BLOCKS

A GP (e.g. Koza 1992) has two types of 'gene'

Terminals: numerical constants or input (x-) variables.

Nodes: mathematical operators or program functions.

GP/ GC FUNCTION (PARSE) TREE

Output

divide

add   z

x   cos

y

$$Output = \frac{x + cos(y)}{z}$$

The genes are organized into a chromosome with a tree structure.

The number of nodes is variable.

Nodes can be of any type.
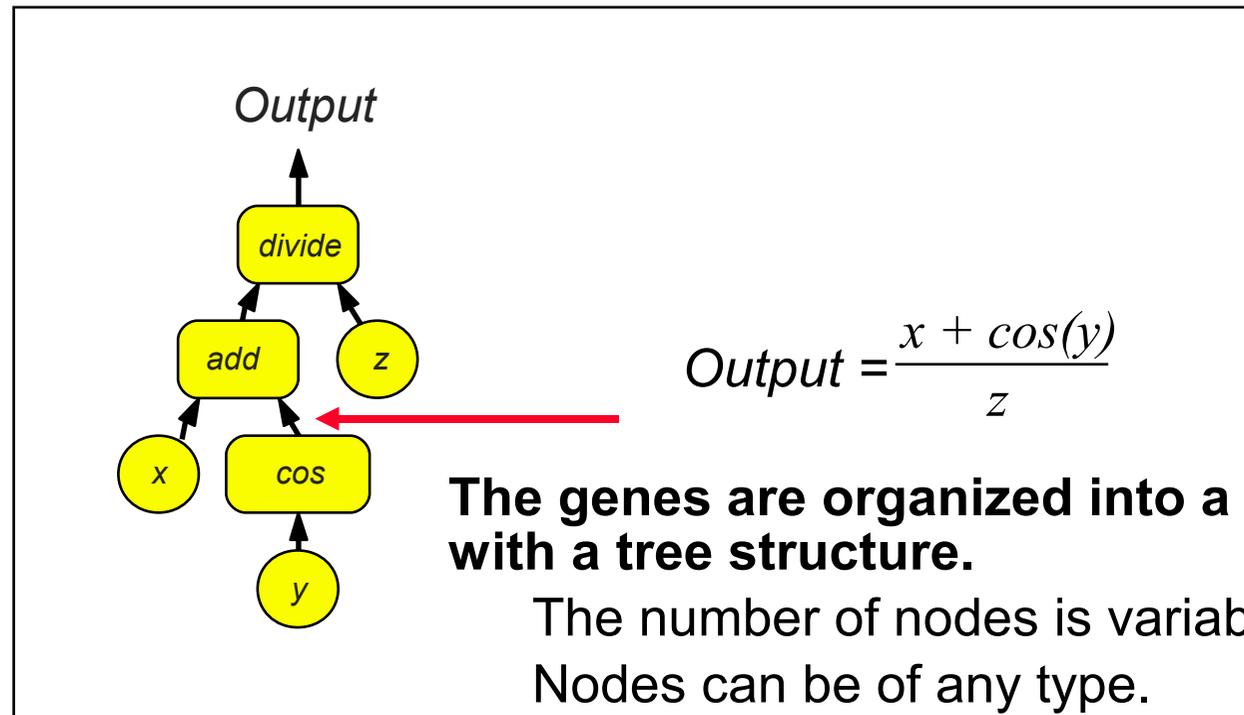
To evaluate the tree, each node evaluates its argument nodes, processes the returned values, and returns its own value.

GP MUTATION
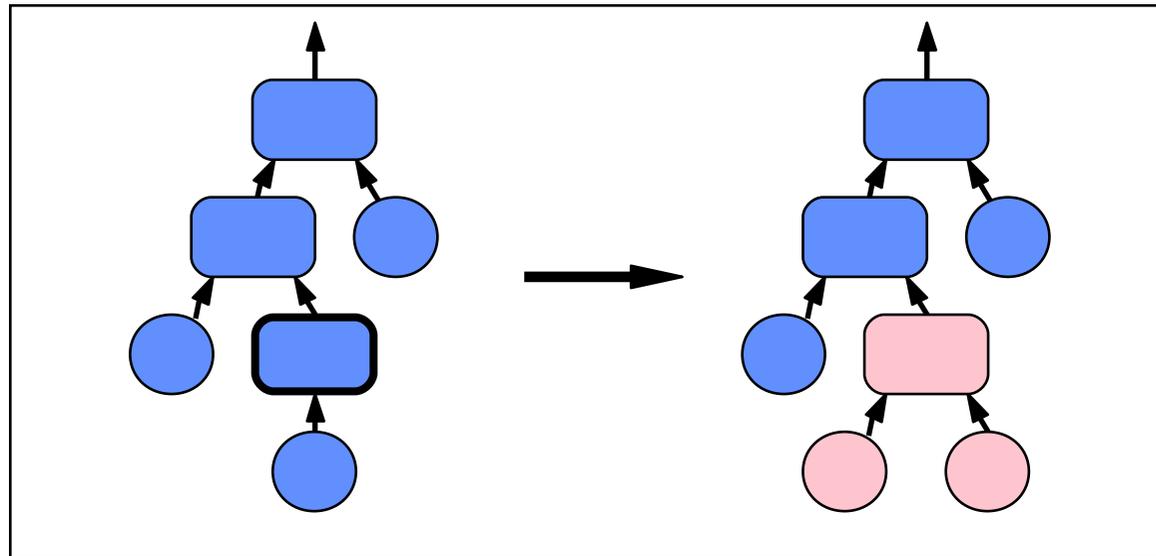
Each node accepts and returns values of the same type.

Trees are modular, allowing logically consistent changes to be introduced.
A node is randomly chosen and modified.

It may be given a different operator with the same number of arguments.

It may be replaced by a new random sub-tree.
Terminals are mutated by slightly perturbing their numerical values, or randomly choosing a new input variable.

GP CROSSOVER
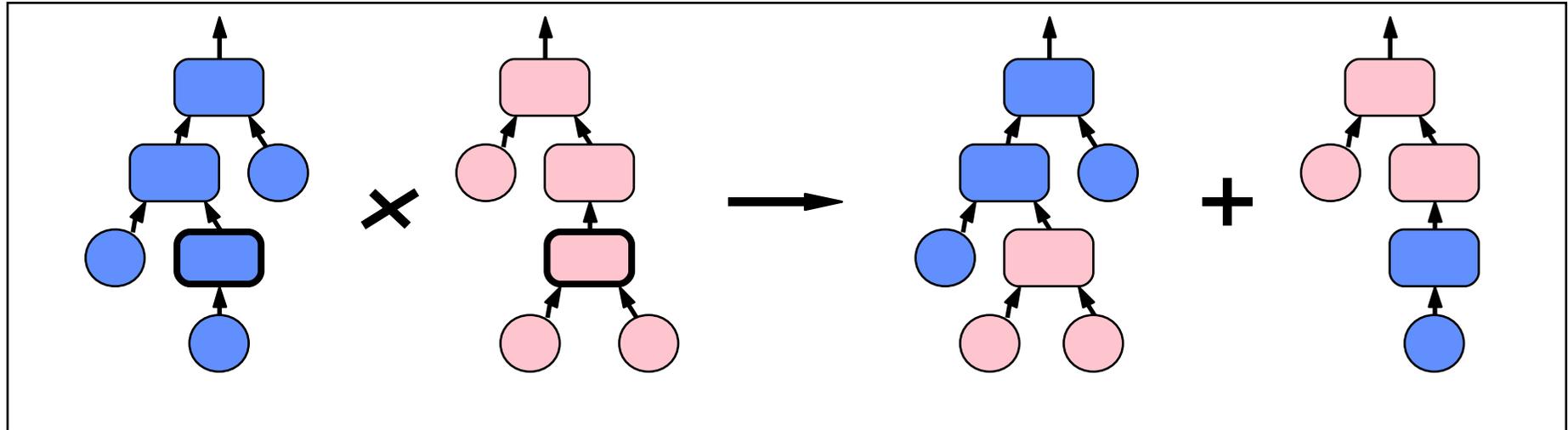
Two parents are chosen with a probability proportional to their fitness.

A node is randomly chosen on each parent tree.

The selected sub-trees are swapped.

The new trees are still syntactically correct.

The new individuals replace less fit members of the population.

Genes as Computer Programs
- a new 'omics paradigm

Genotype-phenotype mapping:genes as computer programs

By Professor Douglas Kell

Trends in genetics Vol 18, No 11 November 2002

The affects of genes on phenotype are mediated by processes that are typically unknown but whose determination is desirable. The conversion from gene to phenotype is not a simple function of individual genes;it is what is known as a non linear mapping problem. A computational method called genetic programming allows the representation of candidate nonlinear mappings in several possible trees.To find the best model,the trees are 'evolved' by processes akin to mutation and recombination, and the trees that more closely represent the actual data are preferentially selected. The result is an improved tree of rules that represent the nonlinear mapping directly. In this way, the encoding of cellular and higher-order activities by genes is seen as directly analogous to computer programs.

This analogy is of utility in biological genetics and in problems of genotype-phenotype mapping.

Specific advantages of Genetic Programming

- Not all variables are used – this at once both (a) cuts the search space hugely and (b) makes the rules intelligible

- Totally assumption free – no chance of operator or statisitcal bias

- Evolutionary computing methods build on partially successesful rules and are <u>highly</u> efficient at negotiating complex search spaces

- Ranking of objects exploits the full range of information available (conventional methods throw it away), and ranking of variables forces explanation to be as simple as useful – which avoids overfitting and greatly improves generalisation

- Preprocessing and normalization are unnecessary – the system learns what it are the best parameters to normalize .

Case Study: Direct Mail Auto Insurance

Goal: Increase subscription rate of auto insurance policies within budget constraint, limiting the mailing to 20% of total customer base (5 million households). Data mining on customer base were performed in four different ways:

	Response Rate	Total Responders	Total $ Spend (average of $50 each)	Net Revenues*	Time to Produce
GMAX – Analyst User:	17.4%	867,500	43,375,000	38,375,000	30 mins
SAS workbench – PhD Stat User:	15.1%	755,000	37,750,000	32,750,000	6 weeks
Leading Data Mining Product – Stat User:	7.2%	360,000	18,000,000	13,000,000	2 weeks
Random (using no model):	5.9%	295,000	14,750,000	9,750,000	n/a

Case Study: Direct Mail Auto Insurance

	Response Rate	Total Responders	Total $ Spend (average of $50 each)	Net Revenues*	Time to Produce
GMAX – Analyst User:	17.4%	867,500	43,375,000	38,375,000	30 mins
SAS workbench – PhD Stat User:	15.1%	755,000	37,750,000	32,750,000	6 weeks
Leading Data Mining Product – Stat User:	7.2%	360,000	18,000,000	13,000,000	2 weeks
Random (using no model):	5.9%	295,000	14,750,000	9,750,000	n/a

By targeting with GMAX:

- **Business analyst captured nearly 3 times more net revenue than the leading competitors in a fraction of the time.**

- **More than 250% improvement over everyday results with competitive software**

- **Beat the results of dedicated PhD team in industry competition**
    - **in 30 minutes vs. 6 weeks**

Conclusions

- Gmax is a data mining and analysis tool that uses genetic programming methods to encourage the spontaneous emergence of *natural* models.

- Natural models capture the unsuspected, latent structure and relationships between variables that conventional methods are not designed to reveal.

- Natural models are robust and have unparalleled predictive power.