Time Series Prediction by Genetic Programming with Relaxed Assumptions in *Mathematica*[1]

Stuart W. Card

PhD candidate, Syracuse University
cards@ntcnet.com
http://users.ntcnet.com/~cards

Abstract. Time series produced by black box systems with both stochastic and nonlinear dynamical components have proven resistant to prediction. Also, prediction alone is unsatisfying: insight into the hidden dynamics is desired. Automatic induction of a system model would be ideal. A genetic programming (GP) / neural network (NN) / wavelet approach is motivated. An initial test problem selection is justified. Data preprocessing is described. The GP is shown to rely on weaker assumptions than those implicit in orthodox methods. An implementation in *Mathematica* is illustrated. GP discovery of equations, NN optimization of their parameters, and joint time-frequency representations, should provide highly parsimonious descriptions, capturing local and global characteristics of stochastic attractors, amenable to meaningful interpretation.

1 Introduction

1.1 Motivation: Genetic Programming / Neural Networks Hybrid

A classical system dynamics description is a single, higher order, ordinary differential [difference] equation (ODE). Another is a system of several, first order, ODEs. A GP can evolve a tree representing a single ODE, or a forest representing a system of ODEs, providing insight into hidden system dynamics. NNs effectively optimize numeric parameters, within a basin of attraction of a global optimum, but may not locate such a basin. Natural biology inspires a hybrid approach: exploit synergy of phylogenetic (population) learning of structures by GP, and ontogenetic (individual) learning of parameters with NNs. Individuals in the GP population are sparse high order NNs. Adaptive individuals may become trapped in local minima of the error function, but the evolving population should sample the fitness landscape sufficiently densely to locate a basin in which may be found a global optimum, providing point prediction accuracy.

1.2 Motivation: Wavelet Representation

Pure time [frequency] domain approaches may miss features readily evident in the frequency [time] domain; thus, the proposed approach uses wavelets to facilitate detection of features irrespective of their 'native' domain. Here 'wavelet' refers to various joint time-frequency (or time-scale) representations:

- some are wavelets in a strict sense;
- some are 'wavelet packets';
- others are more general transforms of the lag vector of observed values.

In all cases, the transforms are motivated by several conflicting objectives:

- to ameliorate the effects of measurement noise in the data;
- to facilitate the discovery of hidden structure in the series;
- to represent discovered relationships in a form amenable to interpretation.

1.3 Suitable Problems

This approach is being developed specifically to predict, and understand the possible origins of, time series that have resisted prediction by other methods, but which are potentially predictable to some extent. The most obvious sources of such data sets are processes that combine chaotic and stochastic dynamics. Chaotic dynamics are deterministic, but difficult to predict due to rapid divergence of nearby trajectories (sensitive dependence on initial conditions). Stochastic dynamics are by definition non-deterministic, but still 'predictable' in the sense of [conditional] probability density functions or distributions. Systems that combine both, confound techniques developed for each. Randomness that enters into system dynamics, versus mere noise corruption of the observable, defeats nonlinear dynamical systems time series embedding techniques. Chaos allows for short-term predictions unachievable by statistical techniques, and introduces apparent nonstationarity that defeats them.

Non-invertible maps with one or more degrees of freedom, invertible maps with two or more degrees of freedom, and continuous time flows with three or more degrees of freedom, can exhibit chaos. Many well-known chaotic systems, including some that model important real-world processes, are of these minimal orders. Higher order systems are also of interest, but present greater modeling and prediction difficulties. The number of degrees of freedom of the system is an upper bound on the attractor dimension $\boldsymbol{d}$, and the bounds on minimum length $\boldsymbol{N}$ of a lag vector used to embed the series are given by Takens' embedding theorem [1] as

$$\boldsymbol{d \le N \le 2\,d + 1}\ . \qquad (\mathbf{1})$$

Above $\boldsymbol{d=5}$, catastrophe theory results related to the Thom classification theorem [2] seem to impose limits on generic modeling even in the absence of noise: there is no universal unfolding for codimensions over 5.

1.4 Initial Test Problem

The ideal initial test problem is the simplest that exhibits all the above characteristics. A process described by a single higher order ODE is preferred to one described by a system of multiple first order ODEs. Sprott has investigated and catalogued many simple chaotic systems, especially jerk equations [3]; simplest is

$$x''' = - a\, x'' + (x')^2 - x \,. \qquad (2)$$

As a is reduced from 2.082, the system follows the common "period doubling route to chaos". From 2.0577 to 2.0168 there is the usual structure of bands of chaos interrupted by periodic cycles. To produce a test problem, the deterministic dynamics can be made stochastic by 'randomly' varying a over the period doubling and chaotic interval, or by adding a 'random' term to the governing equation.

1.5 Data Preprocessing

The time series produced by the initial test process, and a class of similar processes, can be embedded in a lag vector of length 4. Multiplication of a lag vector, by a 4x4 matrix of positive and negative ones, and scalar division by two, is a computationally inexpensive Walsh transform. It yields an orthogonal representation that estimates the observable and its first three derivatives. The hybrid GP system can approximate the third derivative as a function of the rest of the transformed vector, yielding the desired description: a single third order ODE. If the dynamics were strictly deterministic, the ODE would contain only the observable, its derivatives, constant factors and a constant term; stochastic dynamics will introduce random variables that the system also must characterize.

2 Relaxed Assumptions

The hybrid GP system is described. Where its design choices differ from those of orthodox GP methods, the motivation for deviating is identified. In most cases, the purpose is relaxation of the assumptions implicit in standard GP methodology.

The GP incrementally evolves the population at each time step, with no concept of 'generation'. This goes beyond "steady state GP" [4] to something akin to an a-life environment. Time passes and asynchronous events occur. This relaxes many of the assumptions implicit in GP: that all fitnesses are evaluated simultaneously; that all survival challenges occur simultaneously; that all reproduction opportunities occur simultaneously; that fitness is reset to zero for all individuals at the beginning of each generation; etc. At each time step, the GP randomly selects one of 4 event types. In order of decreasing probability, they are: train, test, survive, reproduce. Each is an opportunity or challenge for one or more randomly selected individuals.

2.1 Reproduce

When a reproduction event occurs, a single individual is selected at random from the population, with uniform probability, and presented with a reproduction opportunity. That opportunity is either exploited by the individual, or not, with probability based on scaled fitness. If the opportunity is exploited, one of 3 reproduction methods is selected per configurable probabilities. Recombination is standard GP subtree crossover, trivially implemented in *Mathematica*.

2.1.1 Asexual

The one parent remains in the population. One child, derived strictly by mutation from that one parent, is added to the population. The child inherits half the fitness previously accumulated by its parent. The parent retains the other half. Individuals with high fitness may become immortal in a standard steady state GP; expenditure of fitness (in a-life, *energy*) for the privilege of reproduction mitigates this.

2.1.2 'Autosexual'

The one parent remains in the population; 2 complementary children are derived from the parent by crossover with itself, followed by mutation. Each child inherits a fourth the fitness accumulated by its parent; the parent retains the other half. This is a non-essential extension of the hybrid approach, akin to incorporating Goldberg's genetic algorithm [5] inversion into Fogel's evolutionary programming [6].

2.1.3 Sexual

Candidate partners to the selected individual are successively randomly selected from the population and presented with the reproduction opportunity; the partner is the first candidate that with probability based on scaled fitness exploits it. Both parents stay in the population; 2 complementary children are derived by crossover, followed by mutation. Each child inherits half the average of the fitnesses previously accumulated by its parents. Each parent retains half its original fitness.

2.2 Survive

An individual is randomly selected from the population with uniform probability and presented a survival challenge it meets with probability based on scaled fitness. If it survives, its fitness is reduced by a random amount demanded by the challenge. Survival probability is higher than in a generational GP (as individuals face many challenges), biased by population size $\boldsymbol{S}$ varying logistically with fecundity $\boldsymbol{r = 4}$:

$$\boldsymbol{S[n+1] / S_{max} = r \, (S[n] / S_{max}) \, (1 - S[n] / S_{max})} \; . \qquad (3)$$

If the current population is larger than the size predicted by the logistic equation, survival probability is reduced; if the current population is smaller than predicted,

survival probability is increased. This relaxes the usual assumption that population size remains fixed or varies during the run according to some arbitrary schedule.

2.3 Test

An individual is randomly selected from the population with uniform probability and presented with a test, usually a computationally inexpensive single short term point prediction. It accumulates fitness inversely related to the error in that prediction.[2] Sometimes the test is an expensive iterated prediction. A long-term point prediction can be evaluated as to time, frequency or joint domain error. After many iterations, an estimate of the attractor is reconstructed, the characteristics of which (box counting dimension, Lyapunov exponents, etc.) can be compared with those of the data set. This is very expensive, but necessary for insight into dynamics. The conventional GP assumption relaxed here is that all fitness evaluations are alike; the motivation is to match the methodology more closely to the needs of the application while limiting labor and computational costs. Another view is that short term tests apply to point prediction accuracy of the optimized parameters, whereas long term tests apply to validity of the structural model.

Not only error, but also complexity and resource cost, are penalized. Rather than counting nodes and assigning arbitrary complexity costs to operators and operand types, a fair metric is applied: maximum memory required by function evaluation, multiplied by CPU time required by evaluation. While this will vary by implementation: from an engineering perspective it is a fair and relevant metric of resource cost; and from a theoretical perspective it is as good a metric as any for algorithmic complexity, as that is non-computable anyway. Cost is assessed against accumulated fitness: a mediocre prediction achieved at low cost may accumulate more fitness than an excellent prediction achieved at great cost; accumulated fitness can even decrease rather than increase due to a test.[3]

2.4 Train

An individual is randomly selected from the population with uniform probability and presented with a training opportunity. It expends fitness proportional to the cost of its exploitation of that opportunity. Many of the points above describing testing also apply to training, but fitness is not accumulated in training. Merging training and testing is worth consideration, but that could complicate holding back testing data from the training set, which has generally been found necessary to avoid overfitting. The amount of testing data to be withheld depends upon relative

---

[2] The author thanks the anonymous reviewers for raising the issue of error fluctuation and instability, on which he hopes to receive some advice in the Graduate Student Workshop.

[3] A disadvantage of this method of penalizing resource cost is error in the measurement of memory consumption and evaluation time, due to uncontrolled variables such as interrupts. Tests will be run to determine whether the *Mathematica* resource usage metrics are reliable.

frequency of testing versus training events, and length of GP runs. Related is the distinction between testing the parameter values to which an individual has trained, and testing the structure of the individual: should different test sets be used for short term point prediction and long term iterated tests?

Specific training techniques will be selected, compromising between the desire for general sparse high order neural network structures with learning, and the need to minimize computational cost. *Mathematica's* numerical optimization routines will be exploited. Backpropagation of errors was designed for sum of products neurons with sigmoid activation functions; though it has been generalized to other architectures, there is no general rule applicable to arbitrary GP function trees.

3 Ongoing *Mathematica* Implementation and Testing

Wolfram Research, Inc. has "a fully integrated environment for technical computing", with substantial capabilities for nonlinear stochastic systems analysis and a powerful programming language. It has been considered for GP work by many, but used by few, as it reduces expressions, interfering with their explicit maintenance and manipulation by GP programmers.

Individual models will make point predictions. Population histograms will predict the non-stationary probability density function up to the prediction threshold (where it should converge to an estimate of a measure on the attractor, independent of initial conditions). If the mean squared error (across various prediction lengths, up to the maximum usable prediction length of the better system) of the hybrid system is less than that of the best known general purpose predictors (not equipped with a processs model) on the initial test problem, given comparable human effort and computer resources, the hypothesis will be proved: that such a system can predict time series that have resisted prediction by other automated techniques.

References

1. Takens, F.: Detecting Strange Attractors in Turbulence. Lecture Notes in Mathematics. Springer, Berlin (1981) 366
2. Casti, J.: Reality Rules, Vol. 1. Wiley-Interscience, New York (1992)
3. Sprott, J., Linz, S.: Algebraically Simple Chaotic Flows. Int. J. of Chaos Theory and Applications, Vol. 5 No. 2 (2000)
4. DeJong, K., Sarma, J.: Generation Gaps Revisited. Foundations of Genetic Algorithms 2. Morgan Kauffman, San Mateo CA (1993) 19-28
5. Goldberg, D.: Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley, Reading (1989) 166 ff
6. Fogel, D.: System Identification through Simulated Evolution… Ginn Press (1991)