



Genetic Programming, the Reflection of Chaos, and the Bootstrap: Towards a Useful Test for Chaos

E. Howard N. Oakley

EHN & DIJ Oakley
Brooklands Lodge, Park View Close
Wroxall, Ventnor
Isle of Wight PO38 3EQ, UK
howard@quercus.demon.co.uk

ABSTRACT

This study assessed the use of genetic programming (GP) to diagnose chaos. Fifty GP runs were performed on chaotic data, generated from the Mackey-Glass delay differential equation, on one surrogate with the same Fourier power spectrum and statistics but without chaotic dynamics, and on a random walk series. Single runs were performed on 50 different surrogates of the chaotic series. Fitness was measured across 5 separate forecast periods of 65 points each, each based upon 10 prior input data points.

Fittest program fragments for the chaotic series evolved later and were more complicated than those for the surrogates. Relative to fitnesses achieved by constant linear predictions, fitnesses from the chaotic series were also better. Random walk data resulted in an impoverished GP process, with quick evolution of simple program fragments but no later evolutionary improvement. This comparative test merits assessment on other datasets, and its implications with respect to the statistical bootstrap and GP estimation are discussed.

1. Introduction

Genetic programming (GP) is a technique which has been developed for the purpose of solving problems. To date, it has almost exclusively been used to generate program fragments which are potential solutions to a problem to which it is being directly applied. Thus, interest has either been focussed on the effectiveness of the fittest program fragments in solving particular problems, or on ways of improving the technique of GP to increase that effectiveness or to reduce the effort required. A classic illustration of this is the work of Koza (1994) on Boolean parity problems.

In the course of previous work which used GP to predict chaotic time series (Oakley (1994b); Oakley (1994a)), it was claimed that the course of GP differed between chaotic and

non-chaotic input data series, thus that the internal process occurring reflected the nature of the data (Oakley (1995)). If that is the case, it has important consequences for both the theory and application of GP. It suggests that more attention should be devoted to understanding the dynamics of the internal process and how it may reflect characteristics of the data.

Chaos is notoriously easy to postulate (Glass and Kaplan (1994)) but hard to diagnose and investigate (Ruelle (1990)). If long noise-free data series are available, there is a range of tests which can be used to determine if the underlying system is chaotic or stochastic, and to estimate the dimension of the chaos (Casdagli *et al.* (1992)). Even given suitable data series, though, existing tests tend to be complicated, computationally intensive, and usually involve subjective interpretation. The original motivation behind this series of studies was to investigate the suitability of GP for system identification and prediction of short noisy time series derived from biological systems; such datasets are not amenable to existing diagnostic tests for chaos, leading to speculation but little evidence that many biological systems are chaotic (Glass and Kaplan (1994)). Any test which was capable of use on short noisy datasets would therefore be of immediate practical value.

Considerable effort has been devoted to improving the efficiency of GP using automatically defined functions (ADFs) (Koza (1994)). Although this has lacked a thorough theoretical foundation, the value of ADFs in reducing the number of individuals which have to be evaluated before 'perfect' solutions are found, indeed in some problems in allowing solution at all, has been demonstrated. However, as with most variations of the basic technique, there is little knowledge as to when it should be employed. A better understanding of the processes at work within populations of program fragments during genetic programming should enable the development of a more sound theoretical basis for the technique, and thus for the selection of variants. This is particularly true for variations such as ADFs which have direct effects on the program fragments being evolved. If it is true that the internal process during GP reflects the nature of the data being fitted, and that this is an important feature of GP, any intervention which alters

the internal process could disrupt this reflection.

Another area of both theoretical and practical importance is the relationship between genetic programming and other methods for solving problems; there has been interest in the hybridisation of GP with other machine learning (ML) methods such as neural networks. Given the overwhelming influence of chance within GP, it is perhaps surprising that there has been little combination of statistical methods with GP.

One relatively modern statistical technique which shares some features with, but which also contrasts with, GP is the bootstrap, devised and developed by Efron and Tibshirani (1993). Although derived from the same phrase as the term used in computing¹, its statistical meaning is quite different, applying to a resampling scheme based on Monte-Carlo methods. Under the bootstrap method, a large number of random samples are drawn with replacement from the available data, and a parameter (or antecedent ‘plug-in’ estimator) estimated on each bootstrap sample. The distribution of results from the bootstrap samples can then give a best estimate and confidence intervals of a statistic for the population as a whole. This contrasts with conventional statistical methodology in using probability directly in the derivation of estimators, and is particularly suited to implementation on fast computers.

For example, a bootstrap estimate of the standard error of an arbitrary statistic $s(\mathbf{x})$ could be made thus. A large number B of bootstrap samples are generated by random sampling (with replacement) from the original data set. The statistic $s(\mathbf{x})$ is then calculated for each of the bootstrap samples, and the standard deviation of those B values is then the bootstrap estimate of the standard error of $s(\mathbf{x})$.

The bootstrap has already been the source of inspiration for the use of surrogate data series in the investigation of chaos (Theiler *et al.* (1992)), and the basis of the claim for the value of GP in testing for chaos (Oakley (1995)). However, only one surrogate series was studied previously, conclusions being based upon multiple GP runs on that and truly chaotic datasets. This did not yield any information on the likelihood of those results occurring by chance, nor their dependence on that particular surrogate series.

When many GP runs are performed on a single problem, they can be viewed as applying the bootstrap using the fitness parameter. The initial population is a bootstrap sample of the vast space of program fragments which is then used to evolve a succession of non-random populations with increasing fitness, eventually yielding the fittest individual arising from that bootstrap sample. Choosing the fittest individual found from many GP runs can then be seen as a bootstrap best estimator.

This study therefore set out to compare the outcome of performing many repeated GP runs on: a known chaotic dataset, the Mackey-Glass delay differential equation (Mackey and

¹to pull oneself up by one’s bootstrap, attributed to the legendary liar Baron von Munchausen, was the technique which he purported to use to regain the surface when he had fallen to the bottom of a deep lake.

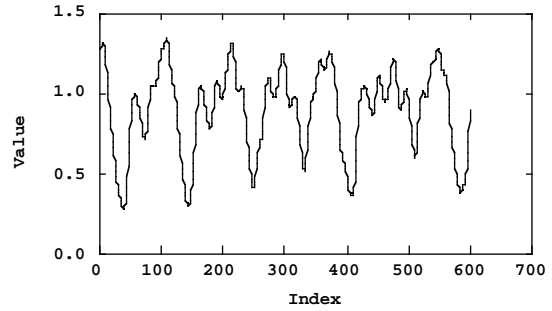


Figure 1: The Mackey-Glass delay differential series.

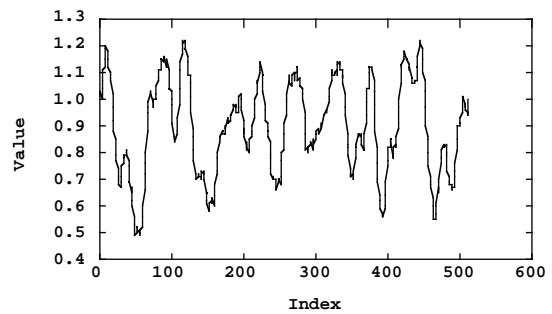


Figure 2: The first surrogate to the Mackey-Glass series.

Glass (1977))

$$\frac{dx_t}{dt} = \frac{bx_{t-\Delta}}{1 + x_{t-\Delta}^c} - ax_t$$

(where a is 0.1, b is 0.2, c is 10.0, and Δ is 30.0); surrogates to it which lack its chaotic qualities; and on random walk data which is purely stochastically determined. Rather than concentrating on the predictive performance of the fittest program fragments evolved, it adopts the bootstrap approach to understanding how the GP process might be influenced by the nature of the data series. Whilst its particular objective was to assess more thoroughly the practicality of using GP combined with surrogate data as a test for chaos, it was also intended to allow excursive insights into these other areas.

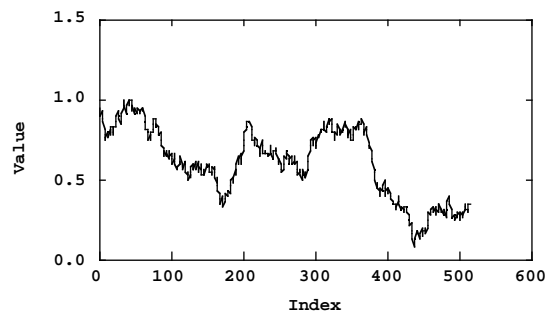


Figure 3: The random walk series.

Table 1: Tableau for fitting data series

Objective	Predict next 65 points at 5 places in series
Terminal set	Embedded data at $t = 1, 2, 3, 4, 5, 6, 11, 16, 21, 31$; \mathcal{N}
Function set	$+, -, \%, *$
Fitness cases	Actual members of the data series
Raw fitness	Sum over the 325 fitness cases of squared error between predicted and actual points
Standardised fitness	Same as raw fitness
Hits	Predicted and actual points are within 0.001 of each other
Wrapper	None
Parameters	$M = 500. G = 51$
Success predicate	None
Max. depth of new individuals	6
Max. depth of new subtrees for mutants	4
Max. depth of individuals after crossover	17
Fitness-proportionate reproduction fraction	0.1
Crossover at any point fraction	0.2
Crossover at function points fraction	0.7
Selection method	Fitness-proportionate (by normalised fitness)
Generation method	Ramped half-and-half

NOTE: % is the protected divide operation (Koza (1992)).

2. Methods

2.1. Data

Two primary data series were generated for these experiments. A 600-long exemplar of the Mackey-Glass delay differential equation was approximated using established Runge-Kutta techniques (Oakley (1995)), as shown in Figure 1. A random walk series was generated by randomly allocating a sign to each member of a suitably scaled series of randomised floating point numbers which were in turn used as successive increments to a seed, as shown in Figure 3. In both cases, the first portion of the original generated series was discarded to ensure that the data used were free of influence from startup processes.

Surrogate data series were generated using the first method of Theiler *et al.* (1992), which ensures that they possess the same Fourier spectra and statistics as the original data. In essence, this requires the application of the discrete Fourier transform to the data, randomization of the resulting phases, symmetrizing the phases (to ensure that the eventual output is real), and then inverting the discrete Fourier transform to generate the surrogate series. Fifty surrogate series were generated for the exemplar of the Mackey-Glass delay differential equation, of which the first is shown in Figure 2.

2.2. Genetic Programming

GP runs were performed using Koza's Simple Lisp code (Koza (1992)), incorporating implementation-specific optimisations to accelerate the evaluation of S-expressions. Code was compiled and run under Macintosh Common Lisp 2.0.1 (Apple Computer, Inc.) and Macintosh Common Lisp 3.0p2 (Digi-

tool, Inc.), respectively on a Macintosh IIfx and a Power Macintosh 6100 computer.

No modifications were employed to incorporate ADFs, and the settings for all GP runs are shown in Table 1. Fifty complete runs were performed using the original Mackey-Glass data, 50 using the random walk data, 50 using the first surrogate of the Mackey-Glass data, and one run each using the 50 different surrogates of the Mackey-Glass data. The same 50 different random number seeds were used for the first three sets of runs, whilst the same random number seed was used for each of the 50 runs on different surrogates.

It is emphasised that fitnesses were not derived from the assessment of single forecast series, but from the sum of five complete forecast series, each of which used input data immediately prior to the forecast period, and each forecast extended 65 steps into the future. Intervals between forecasts were chosen to ensure that there was no overlap between input and forecast data, so that input data and forecast series varied over the 5 applications to each input dataset, and so that the maximum possible number of forecast points were evaluated over each dataset.

In contrast to previous work (Oakley (1995)), the function set was restricted to the four arithmetical operators.

2.3. Analysis

For each GP run, the fittest S-expression evaluated over the 51 generations was recorded, together with the generation at which it first appeared, its fitness, and the number of left parentheses contained within that S-expression. These data were pooled for each set of 50 runs and analysed using Systat and Statistica statistical analysis software. Some of the fittest S-

expressions resulted in constant linear fits i.e., predicted constant values for all 325 fitness cases; one example of the fitness of such an S-expression was recorded for each set of runs, as an indication of the fitness of such a minimal forecast.

3. Results

Characteristics of the different input datasets are given in Table 2. The other 49 surrogates were statistically indistinguishable from surrogate series 1.

Fittest S-expressions evolved during runs using the original Mackey-Glass series displayed the relative richness seen previously, in terms of the range of generations at which they appeared, the complexity of the expressions themselves, and the range of different expressions and fitnesses. Particularly impressive was the apparent bi-modal frequency distribution (Figure 4) of the generation at which the fittest was found; thus it is clear that continuing the GP beyond the 51 generations employed would have still stood a worthwhile chance of finding fitter individuals.

In contrast, the fittest S-expressions generated from the surrogate data were achieved in relatively few generations (Figure 5), were simpler, and occurred more commonly than those of the chaotic data. This was still more true of S-expressions from the random walk data. These results are compared in Table 3. It is also important to note that the overall fittest S-expressions found for surrogate series were considerably fitter than those for the original series, and that the fittest S-expression for the random walk series was fitter still.

In spite of these observations of the fittest individuals found in each run, a few populations evolved progressively more complicated – but less fit – S-expressions when trying to fit surrogate and random walk data. But more commonly, the fittest of even late generations were simple and less fit variations of the fittest of earlier generations. Thus the GP process was impoverished and dominated by simple S-expressions of second-rank fitness, and there was little advantage in running more generations.

4. Discussion

4.1. Chaotic versus Stochastic Systems

The random walk series used in this study was not matched to the original Mackey-Glass series for Fourier power spectra or statistics, and the effects of this are seen in the wide differences in fitnesses achieved, although the means and variances of the series (Table 2) did not differ by much. Examination of the data series suggests an explanation, in that the Mackey-Glass series exhibited nearly 6 ‘pseudo-cycles’ over the 500 individuals used in GP, whilst the random walk data had less than 3 (compare Figures 1 and 3). Thus, constant linear predictions were considerably fitter when applied to the random walk data (see the entries for typical linear fitness in Table 3). This calls into question the practice of using normalised error measures to compare forecasting techniques applied to different datasets

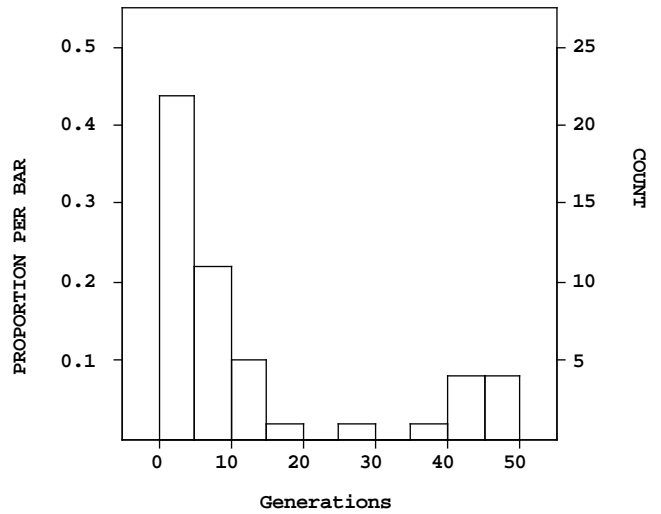


Figure 4: Frequency distribution of generations at which fittest S-expression was found, Mackey-Glass series.

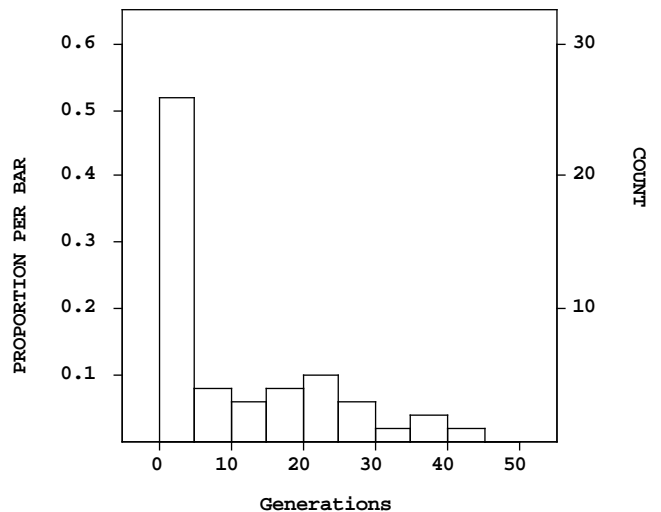


Figure 5: Frequency distribution of generations at which fittest S-expression was found, surrogate series 1.

Table 2: Characteristics of three input data series

Series	Mean	Number	Variance	Std. dev.
Mackey-Glass	0.8987	600	0.0696	0.2639
Surrogate 1	0.8975	512	0.0315	0.1774
Random walk	0.6059	512	0.0482	0.2196

Table 3: Summary of fittest S-expressions resulting

Series	Mackey-Glass	Surrogate 1	Surrogates	Random walk
Number	50	50	50	50
Mean generations to fittest	13.38	11.18	8.660	7.200
Std. dev. of generations	16.46	12.95	11.97	9.010
Median generations	5	4	3	3
No. of generations ≥ 25	10	8	7	4
No. of generations ≥ 40	9	2	2	1
Mean best fitness	10.22	4.305	4.230	0.4916
Std. dev. of best fitness	3.371	1.073	1.543	0.0571
Median best fitness	10.51	4.079	3.997	0.5105
No. of duplicate fitnesses	3	6	0	27
Overall best fitness	3.851	2.833	1.799	0.3507
Typical linear fitness	11.44	5.110	5.334	0.5246
Mean left parentheses	9.120	4.600	6.880	4.480
Std. dev. of left parens.	17.64	7.420	10.05	4.917
Median left parens.	3	2	3	2
No. left parens ≥ 10	8	4	8	7
No. left parens ≥ 20	6	2	4	1

(Gershenfeld and Weigend (1994)); instead, it is proposed that any comparisons of fitness of prediction achieved between datasets should be standardised against the fitness resulting from a constant linear fit. When expressed against the typical linear fitness value for its dataset, the overall best fitness for the Mackey-Glass data is considerably better than that for the random walk series, at 0.3366 and 0.6685 respectively, with a smaller difference between mean best fitnesses at 0.8934 and 0.9371. Because the use of fitness measures during GP runs is not normally susceptible to these effects, this issue should not affect the choice of fitness measure for GP, only comparisons between best fitnesses achieved over different datasets.

Otherwise, the original Mackey-Glass series and the random walk data were easily distinguishable in all respects, with the latter resulting in early convergence to simple fittest S-expressions. The remarkably high number of duplicate fitnesses among the runs on random walk data reflects the general poverty of the populations and the GP process. This agrees with previous claims that the ‘richness’ of the GP process mirrors the variety of the fitness landscape (Oakley (1995)). There is no evidence that running GP for more generations on the random walk data would result in any improvements in best fitness. This reflects the lack of any ‘deeper structure’ (attractors or similar) within the data, in contrast to the richer structures within the chaotic data.

4.2. Chaotic versus Surrogate Systems

When fitnesses are expressed relative to the typical constant linear fitness value for that series of runs, the mean best fitness values across the three series derived from Mackey-Glass data are very similar: 0.8934, 0.8425 and 0.7930 for the original chaotic series, the multiple runs on the first surrogate, and single runs on multiple surrogates. However, a different picture emerges when the overall best fitnesses are so compared: 0.3366, 0.5544 and 0.3373 respectively. Again, use of the typical linear fitness value to make comparisons between fitnesses derived from different datasets appears valid, although because of the similarity in Fourier power spectra, normalisation against the standard deviation of each series could also be appropriate in this instance.

The contrasts between chaotic and surrogate series are considerably less than those between chaotic and stochastic series. However, there are still some differences which are substantial enough to merit discussion. Among the Surrogate 1 runs, only 4% evolved their fittest S-expressions after 40 generations had been evaluated, whilst 18% of the original data runs did so. The visual contrast between Figures 4 and 5 is impressive, even if their statistical distinction is more difficult.

Chaotic data also resulted in the fittest S-expressions being more complicated, as shown by the frequencies of S-expressions with 20 or more left parentheses: 4% for Surrogate 1, against 12% for the chaotic series. In general, the

more complex S-expressions evolved during runs on surrogate data did not achieve the best fitnesses, again suggesting that continuing runs for more than 51 generations would not be likely to result in much improvement in fitness. However there is good evidence (Figure 4) that additional generations could have led to further improvements in fitness for the prediction of chaotic data. The GP process is not as impoverished with surrogate data as when used on purely stochastic data, but it is not as rich as when used on truly chaotic data, with its ‘deeper structure’ (attractors).

4.3. The Bootstrap

Two different variations on the bootstrap were used in this study. The first used a single surrogate series, on which multiple GP runs were performed, whilst the other performed single GP runs on each of multiple surrogates. In the first, resampling was only performed from the vast space of possible S-expressions, and they constituted the bootstrap samples. In the latter, the bootstrap samples were drawn from the vast set of possible surrogate data series, on which resampling of S-expressions was not performed. Examination of the measures shown in Table 3 shows how remarkably similar the outcomes were. The only differences which appear to be of significance are a tendency towards rather more complex fittest S-expressions, as shown by measures based on the count of left parentheses, although in general these do not reach the values shown by true chaotic data.

This is an important result with respect to the value of performing multiple GP runs on a single representative dataset. Although in this case there is little difference in programming or computational effort between that and running GP on multiple datasets, in other problems it is much easier to obtain a single dataset. This does not mean that the actual S-expressions are necessarily equivalent – applying the fittest S-expression derived from one surrogate to another surrogate series would of course be expected to result in a reduction in observed fitness – but that the estimators of the GP process would be equivalent. This result also suggests that more formal exploration of the bootstrap approach to GP would be worthwhile. One important factor which is likely to have blurred any initial distinction between the results from the two variations of the bootstrap is the process of evolution. Assuming that this is driven by selection pressure towards a common goal, the eventual outcome should be qualitatively if not quantitatively similar.

4.4. Two Views on Estimators

Conventional ML practice uses training datasets to generate estimators, perhaps to classify data, and then applies the estimators to unseen data in order to assess their efficacy. If the accuracies of training and assessment are dependent on the quantity of data (for instance the length of time series) to which they are applied, then striking the optimal balance between the quantities devoted to each can be difficult. If only

very short datasets are available, then this approach can fail altogether.

By contrast, statistical estimators are normally gained from the largest samples possible, unless a complete census is available. The bootstrap is a variation on this, although the technique is still applied to the whole sample available, no data being withheld for the purposes of assessing the accuracy of the estimate. Indeed, the bootstrap approach to measurement of confidence intervals can be seen as opposite in intention to the two phases of ML.

In many real-world situations, notably in the study of short time series, keeping half the data unseen will seriously weaken the prediction estimator generated from the other half. Consideration of the graphs shown in Figures 1 to 3 demonstrates that halving the length of each dataset would permit only 2 prediction series in the random walk and surrogate data, as opposed to the 5 used here, in turn restricting the general predictive power of the fittest S-expressions resulting from GP. Furthermore, there would be a wide disparity between fitnesses measured during ML training and assessment phases if the first half of each dataset was used for training and the second for assessment. Real data series are often non-stationary too, further frustrating efforts towards the two-phase ML approach.

There is a need for more work in this area to clarify the costs and benefits of the ML two-phase approach, against that of the statistical best estimator, before an informed choice can be made in the prediction of time series.

4.5. Proposed Test

Given this more extensive study, the previous proposal (Oakley (1995)) can now be refined into a formal test which should be applied to a wider range of chaotic and non-chaotic datasets. The data series to be studied should be subjected to a large number (50 or more) of predictive runs using a GP technique similar to that employed here, with the fittest program fragments from each run being recorded with respect to the generation in which they first occurred, their fitness, and size or complexity. If one of these program fragments does not produce a constant linear prediction, then the fitness of a typical constant linear prediction should be measured separately, to enable standardisation of fitnesses between series.

One or more surrogate datasets should be generated according to the first method of Theiler *et al.* (1992), and the same GP should be performed on these surrogates. The user has the option of running the same large number of GP runs on a single representative surrogate, or of running GP once on many individual surrogates, as has been performed in this study. If there is any doubt as to how representative a single surrogate might be, then the latter would be preferable, and should in general be encouraged until GP is better understood in terms of the bootstrap. Unfortunately, rigorous statistical tests of significance are not immediately available, and further work needs to be done to determine the most appropriate form of comparison.

It is not possible to say at this stage whether the GP method

used should exclude ADFs or other variations of the basic technique. Intuitively, it might appear that the effects of ADFs in manipulating the content of program fragments, and in encouraging early convergence, would be disadvantageous. Indeed, the results of this study could be used to argue against the whole principle of ADFs. However, formal assessment of the effects of ADFs is required before any conclusions should be drawn.

5. Conclusions

Standard GP, without the use of ADFs, has been shown to be a useful tool in trying to distinguish one chaotic series from its surrogates and purely stochastic data. The process occurring within the GP, as shown by the fittest program fragments across many runs, appears to relate to the degree of complexity in the data being predicted. The less stochastic, and the greater the influence of chaotic attractors, the richer will be the GP process. This is marked by convergence in later rather than earlier generations, fitnesses (relative to 'simple linear prediction' fitness) which tend to be better, and more complex program fragments. In contrast when trying to predict stochastic data the GP process is impoverished. These differences can be used as a diagnostic test for chaos, on the short data series commonly encountered in biological sciences, for example.

This study has also raised interesting and potentially important issues with respect to the relationship between GP and the bootstrap, even viewing the application of multiple GP runs as a form of bootstrapping. A statistical approach may also be warranted with respect to the most efficient use of limited amounts of data for best estimate prediction – whether a bootstrap view can compete with ML-based two-phase estimation. Finally, although ADFs might appear to be disadvantageous in this work, they merit formal assessment.

Acknowledgements

This work has been influenced by numerous discussions with participants at the Genetic Programming section of the AAAI Fall Symposium series, November 1995, at MIT, Cambridge MA.

Bibliography

[Casdagli *et al.*, 1992] M. Casdagli, D. des Jardins, S. Eubank, J. D. Farmer, J. Gibson, and J. Theiler. 1992. Non-linear modeling of chaotic time series: theory and applications. In J. H. Kim and J. Stringer (editors). *Applied Chaos*. New York, NY: Wiley-Interscience. Pages 335–380.

[Efron and Tibshirani, 1993] B. Efron and R. J. Tibshirani. 1993. *An Introduction to the Bootstrap, Monographs on Statistics and Applied Probability*. Volume 57. New York, NY: Chapman and Hall.

[Gershenfeld and Weigend, 1994] N. A. Gershenfeld and A. S. Weigend. 1994. The future of time series: learning and understanding. In A. S. Weigend and N. A. Gershenfeld (editors). *Time Series Prediction: Forecasting the future and understanding the past*. Reading, MA: Addison-Wesley. Pages 1–70.

[Glass and Kaplan, 1994] L. Glass and D. Kaplan. 1994. Complex dynamics in physiology and medicine. In A. S. Weigend and N. A. Gershenfeld (editors). *Time Series Prediction: Forecasting the future and understanding the past*. Reading, MA: Addison-Wesley. Pages 513–528.

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

[Koza, 1994] J. R. Koza. 1992. *Genetic Programming II: Automatic Discovery of Reusable Programs*. Cambridge, MA: The MIT Press.

[Mackey and Glass, 1977] M. C. Mackey and L. Glass. 1977. Oscillation and chaos in physiological control systems. *Science*. Volume 197. Pages 287–289.

[Oakley, 1994a] E. H. N. Oakley. 1994. The application of genetic programming to the investigation of short, noisy, chaotic data series. In T. C. Fogarty (editor). *Evolutionary Programming, Lecture Notes in Computer Science*. Number 865. Berlin, Germany: Springer-Verlag. Pages 320–332.

[Oakley, 1994b] E. H. N. Oakley. 1994. Two scientific applications of genetic programming: stack filters and non-linear equation fitting to chaotic data. In K. E. Kinnear Jr. (editor). *Advances in Genetic Programming*. Cambridge, MA: The MIT Press. Pages 369–389.

[Oakley, 1995] E. H. N. Oakley. 1995. Genetic programming as a means of assessing and reflecting chaos. In *Genetic Programming, AAAI-95 Fall Symposium Series*. American Association for Artificial Intelligence. November 1995. Pages 68–72.

[Ruelle, 1990] D. Ruelle. 1990. Deterministic chaos: the science and the fiction. *Proceedings of the Royal Society of London A*. Volume 427. Pages 241–248.

[Theiler *et al.*, 1992] J. Theiler, B. Galdrikian, A. Longtin, S. Eubank, and J. D. Farmer. 1992. Using surrogate data to detect nonlinearity in time series. In M. Casdagli and S. Eubank (editors). *Nonlinear Modeling and Forecasting*. Redwood City, CA: Addison-Wesley. Pages 163–188.