Non-parametric smooth non-linear model identification and construction

by

W. Haythorn, S. Margetts, P. Durrant and Antonia J. Jones

Abstract. We discuss the application of new techniques for the identification of smooth models of many variables. Based on the Gamma test, these non-parametric methods enable us to quickly evaluate, *prior to model construction*, an estimate for the best mean-squared error that can be achieved by a smooth model on unseen data for a given selection of inputs. By examining this estimate for alternative selections of inputs, we show how the best choice of inputs for modelling a particular target output can be selected. We also discuss how the distribution of the data in input space and the complexity of the unknown function that we seek to model influence the size of the required data set. The techniques have been implemented in a Windows application called *winGamma*™ licensed by the University of Wales, Cardiff. We demonstrate some typical *winGamma* analyses for time series prediction, including chaotic time series with lags.

Keywords: Gamma test, model identification, chaos, time series, prediction, smooth model.

Wayne Haythorn

Granite Software
Mosier
Oregon 97040
USA

email: haythorn@gorgenet

S. Margetts, Peter Durant and Antonia J. Jones

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF WALES, Cardiff
PO BOX 916
CF2 3XF, Wales, UK

Telephone: 0-122-287-4812
Telefax: 0-122-287-4598

Date/version: 7 June 1999
International Seminar on Forecasting 99. Washington 1999.

Non-parametric smooth non-linear model identification and construction

CONTENTS

FIGURES

Non-parametric smooth non-linear model identification and construction

Wayne Haythorn
Grantite Software, Moiser Orgeon

S. Margetts, P. Durrant and Antonia J. Jones
Department of Computer Science, University of Wales,
Cardiff, PO Box 916, CF2 3XF, Wales UK

Abstract. We discuss the application of new techniques for the identification of smooth models of many variables. Based on the Gamma test, these non-parametric methods enable us to quickly evaluate, *prior to model construction*, an estimate for the best mean-squared error that can be achieved by a smooth model on unseen data for a given selection of inputs. By examining this estimate for alternative selections of inputs we show how the best choice of inputs for modelling a particular target output can be selected. We also discuss how the distribution of the data in input space and the complexity of the unknown function that we seek to model influence the size of the required data set. The techniques have been implemented in a Windows application called *winGamma*™ licensed by the University of Wales, Cardiff. We demonstrate some typical *winGamma* analyses for time series prediction, including chaotic time series with lags.

Introduction.

Suppose we are given an input/output data set of the form $(\mathbf{x}(i), \mathbf{y}(i))$ $(1 \le i \le M)$, where $\mathbf{x}(i) \in \mathbb{R}^m$ and, without loss of generality, $y(\mathrm{i}) \in \mathbb{R}$. [1] Assume the data is described by an underlying model of the form

$$y = f(x_1, \ldots, x_m) + r = f(\mathbf{x}) + r \qquad \textbf{(1)}$$

where f is a smooth function with bounded partial derivatives, and r is a stochastic variable with mean zero and bounded variance.

The Gamma test [Stefánsson 1997], [Končar 1997] is designed to estimate the variance of r, Var[r], i.e. that part of the variance of the output which cannot be accounted for by the existence of some underlying smooth model f (even though f is unknown). A single run of the Gamma Test has a time complexity of O(Mlog$M$) and depending on the input dimension m, executes in a few seconds for data sets of a reasonable size (i.e. $M \le 1000$). This algorithm requires certain pre-conditions:

- **Assumptions**. We assume that training and testing data are different sample sets in which:

  (a) Input and output variables take continuous values;
  (b) The training set inputs are non-sparse in input-space;
  (c) Each output is determined from the inputs by a deterministic process which is the same for

[1] If y is a vector we treat each component separately and at very little extra computational cost obtain an individual estimate of the Gamma Test result for each output.

both training and test sets;
(d) each output is subjected to statistical noise whose distribution may be different for different outputs but which is the same in both training and test sets for corresponding outputs.

Here we shall not enter into the details of the algorithm or a theoretical discussion of its range of applicability. For sampling distributions in input space whose support have positive measure theoretical proofs of the algorithm have been given. However, the test seems quite robust with respect to the probability density function φ of the sampling distribution in input space and to the precise nature of its support. This is fortunate because many of the more interesting applications involve chaotic (possibly noisy) time series, and with typical chaotic time series the support of φ (in the embedding space) has measure zero and a Hausdorff dimension which is often substantially less that *m*. For such cases the test works well and the number of data points *M* required to obtain an accurate estimate of Var[*r*] is substantially less than might otherwise be the case (e.g. if the sampling distribution were uniform over input space).

Why is the test useful?

Suppose we have some experimental data with $m = 20$ inputs and one output. We want to know the answers to questions such as:

- Do the inputs determine the output by a smooth model?
- Given an input vector **x** how accurately can we predict the output *y*?
- How many data points do I need to be able to make a prediction with the best possible accuracy?
- Which inputs are relevant in making the prediction and which are irrelevant?

Example. ***Noisy sine***.

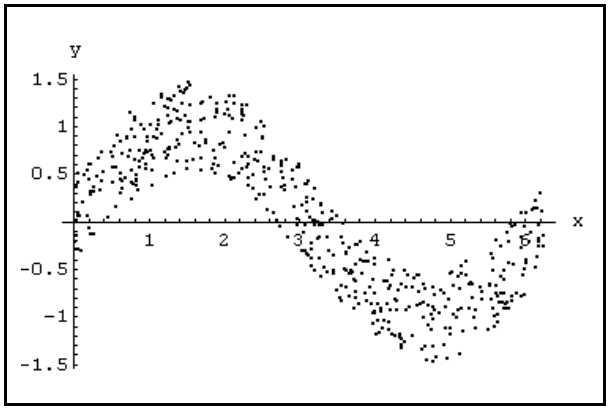

Figure 1 The noisy sine data (500 points).

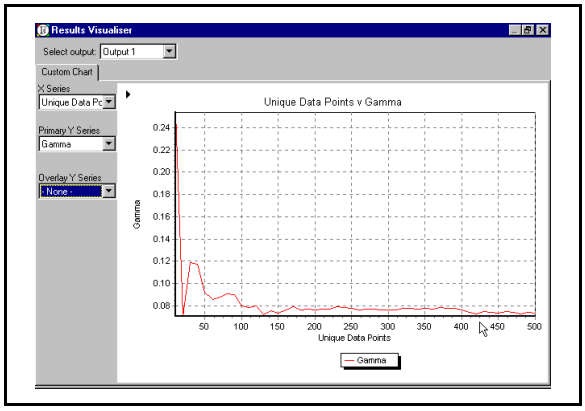

Figure 2 An M-test on the noisy sine data.

First get your input/output data and produce a file in the correct format. The number of input/output pairs should be as large as possible. Typically $M < 200$ is going to yield poor results but one needs to take account of a number of factors which we shall discuss shortly. Second run the software, load the data file, and run the Gamma Test. What is the resulting gamma value (i.e. Γ)? You can see the relevance in the following table.

Table 1 Gamma against expected error.

gamma (or Γ)	Average Absolute Error of Prediction
0.01	0.100
0.0025	0.050
0.0001	0.010
0.000001	0.001

- The variance of the noise on the output determines the best Mean Squared Error that can be obtained for a

prediction on the basis of the data.

Thus if the resulting Γ value is very low we can conclude that there is a smooth model which can make accurate predictions.

Strictly, speaking we should compare the Γ value with the variance of the actual output. It is the ratio of these two which is important. If $V_{ratio} = \Gamma/\mathrm{Var}[y] \approx \mathrm{Var}[r]/\mathrm{Var}[y]$ is low then the output is highly predictable. This statistic is closely related to what the statisticians call R^2, in fact $R^2 = 1 - V_{ratio}$

The data illustrated in **Figure 1** was obtained by adding uniformly distributed noise with a variance of 0.075 to the y values of y = sin(x) for 500 randomly selected values of x. The result of running the Gamma test on this data is a Gamma value of 0.07355 which is quite close to the theoretical noise variance. The *Vratio* of 0.12762 suggests that we will not be able to predict the value of an output very accurately, which in view of the data plot in **Figure 1** is not too surprising. The *SE* is 0.0037651 which indicates a fair degree of reliability in this assessment.

How stable is the Gamma statistic as the number of data points varies? We can answer this question by evaluateing the Gamma statistic for increasing *M*. The result is shown in **Figure 2** and we can see that after around 425 points the graph is fairly stable. Here the standard error is 0.003765 and we can see that the estimate of $\Gamma \approx 0.073354$ is accurate to 2.19%.

Probably the most useful aspect of the Gamma test is that it can be used to determine the relative importance of independent variables. Of course the test itself is independent of the particular non-linear modelling technique selected but it has also proved very useful in reducing the number of feedforward neural networks trained by a trial and error approach, since it allows us to estimate when a network has reached optimum performance.

Another, potentially more important application of this method, is that it makes it possible to avoid wasting time on developing forecasting systems that, because of lack of data or lack of dependencies in the data, will never produce sufficiently accurate predictions whatever the choice of modelling paradigm.

How much data is required?

Assuming all the preconditions are satisfied (later versions of the software will automatically check for gross violations) then the principal factors which will determine how accurately the returned value of Γ estimates Var[*r*] are:

- The number of data points *M*, which affects the local density of sampling in input space.
- The complexity of the unknown surface *f*. Complex surfaces with high curvature will require many more data points.

The M-test.

One of the key questions we need to answer in a practical situation is how much data do I need to get an accurate estimate of gamma and to subsequently build a model which can predict with this Mean Squared Error.

This question can be answered by running an ***M*-test** (as in **Figure 2**). All this means is that we run the Gamma Test using increasing *M* and then plot a graph of *gamma* against *M*. Typically what will happen is that for small *M* the graph will show large variability but as *M* increases the graph will stabilize to an asymptote which reflects the true value of the noise variance. When the graph has stabilized there is nothing much to be gained by using a larger *M*.

Selecting a good embedding for time series prediction

The standard approach for time series modelling is to construct a predictive model based on some number of

previous values (this number is called the *embedding dimension*).

- We can estimate the required embedding dimension by running the Gamma test with an increasing number of past values *m* as the inputs and the output being the current value.

Typically we find that the Gamma value first decreases as more past values are included and the increases. The value of *m* which minimise the Gamma value gives a good initial estimate of the embedding dimension.

If we were following Takens theorem exactly we should include all past values up to the embedding dimension as inputs for our predictive model. However, we have found that often it is better to omit some previous values an include others - a so called *irregular embedding* [Oliveira 1999], [Tsui 1999].

The Gamma test is sufficiently fast that we can probably search over all 2^m - 1 embeddings provided $m \leq 20$. For larger *m* we can search using a genetic algorithm.

Example: *The sunspot data*.

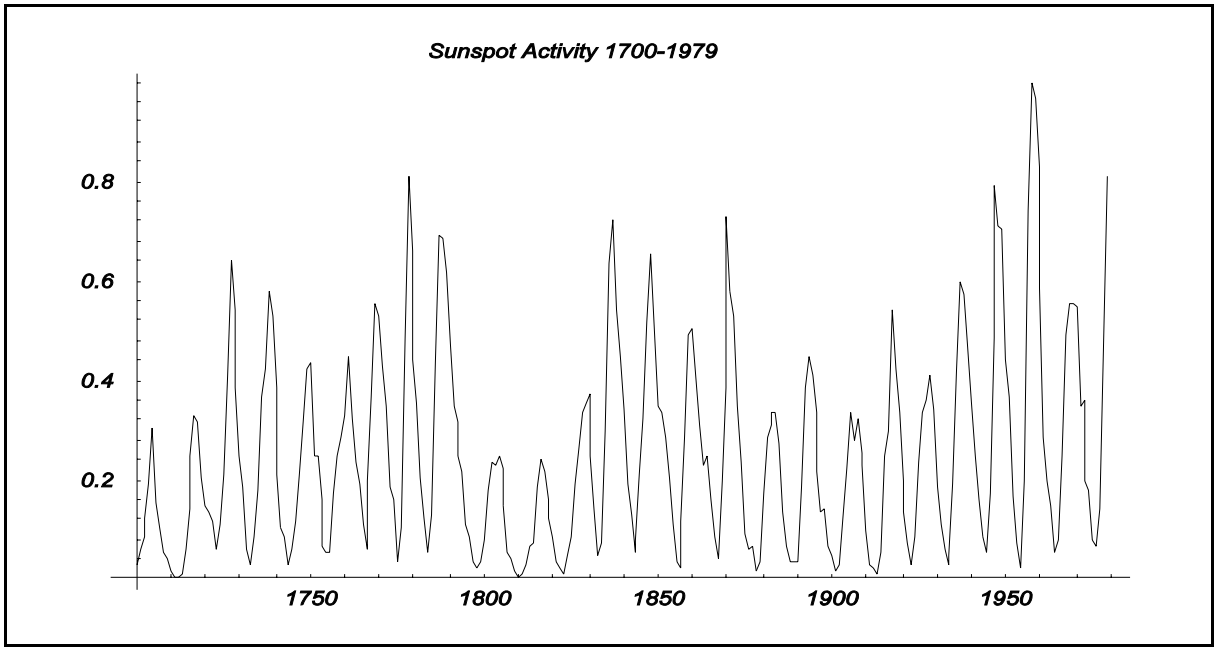

Figure 3 Variation of sunspot activity.

The data used in this experiment was FTP-ed from ftp address: *ftp.santafe.edu*, directory: *pub/Time-Series/data*. Its origin, normalization and training/test regions are described in [Weigend 1990]. The data consists of 280 points representing sunspot activity over the period 1700 - 1979 and was used in [Weigend 1990]. The range of the data has been scaled to [0, 1] and we found the variance to be 0.0410558. **Figure 3** shows the variation of sunspot activity over the full range of the data.

It is known that the primary sunspot cycle is approximately periodic over 11 years. Other shorter and longer cycles are also known. For radio propagation the short period cycle of 28 days is particularly significant. The data used here is collected from telescopic observations projected onto a white paper card. The sunspots are counted and classified by size and a correction factor applied depending on the magnification of the telescope. The virtue of this data is that it has been regularly collected since 1700. Of course, if one were really interested in predicting sunspot activity much more accurate data is available. The data provided is often used as a test of prediction techniques and can give a reasonable model of gross sunspot activity.

Selecting a best embedding. If we are prepared for a several day run we can use the Full-Embedding option of the software to search for a good embedding. In this example we searched over the previous 15 years.

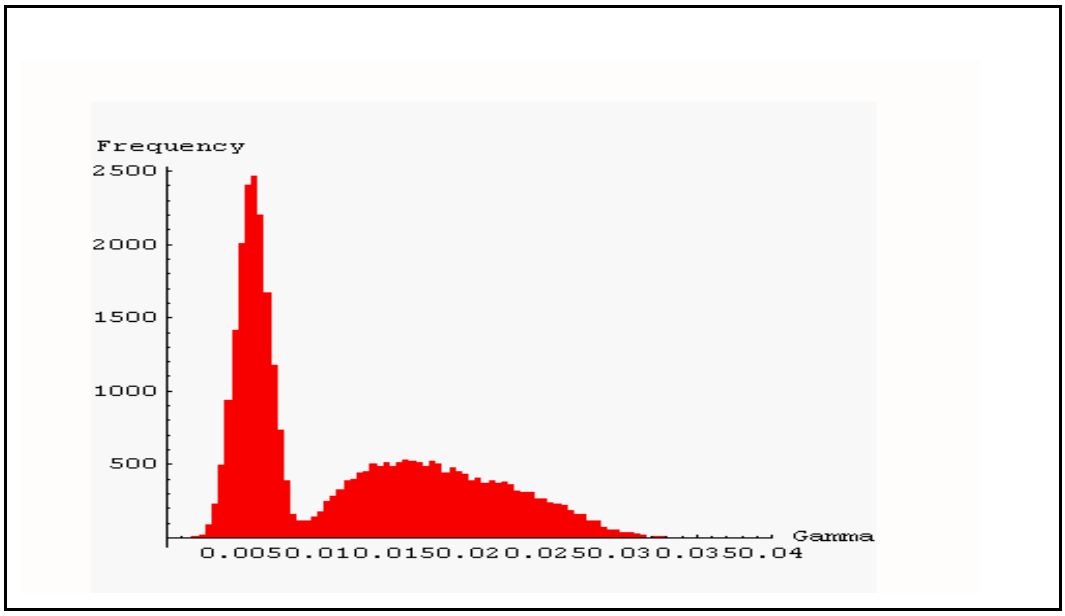

Figure 4 Histogram of gamma values for all embeddings up to length 15 for the sunspot data.

The best embedding found was 001001000010111. Here the most recent data comes last. So this embedding says that to predict this year's sunspot activity $x(t)$ we should use the data $x(t\text{-}1)$, $x(t\text{-}2)$, $x(t\text{-}3)$, $x(t\text{-}5)$, $x(t\text{-}10)$ and $x(t\text{-}13)$, an embedding of dimension six. It is interesting to note the bimodal distribution of **Figure 4**. The bimodal distribution is partly explained by the observation that only 2.38% of the embeddings with $\Gamma > 0.008$ include $x(t\text{-}1)$ as compared with 99.8% of those having $\Gamma < 0.008$. Put plainly this says that the most important predictive factor for the sunspot activity this year is the value for last year. It is also interesting to see which variables appear in the best few embeddings. These indicate that the last few years, plus the value approximately one 11 year cycle back, plus a value about half way through the previous cycle, give the best results. This is rather impressive since the software has no way of knowing about sunspot cycles.

If we run the Gamma test on the six inputs/one output I/O data file constructed using this mask we get $\Gamma \approx 0.0015$ and $V_{ratio} = 0.036$ (SE ≈ 0.00093) with the summary of results in **Table 2**. Note the *M*-test of **Figure 5** indicates that there is not really enough data (the graph has not stabilized). Therefore if we construct a model and test on unseen data we might expect to get a higher MSError than the estimated gamma value. If we now predict the last 59 years data, using local linear regression with $p_{max} = 60$, on the basis of all the previous years we obtain **Figure 6**, which gives a MSError around 0.007. In cases such as this, where there is insufficient data, it is not uncommon to see a MSError on unseen data around an order of magnitude greater than the gamma value.

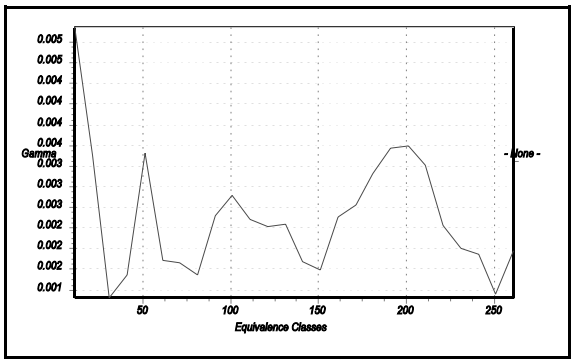

Figure 5 M-test for *sunpairs.asc*.

Table 2 Basic results for *Sunpairs* (267 points).

SunPairs.asc (p_{max} = 10)	
True noise	Unknown
Γ	0.0015140995
A	0.164317091
SE	0.00093484
V_{ratio}	0.0365864

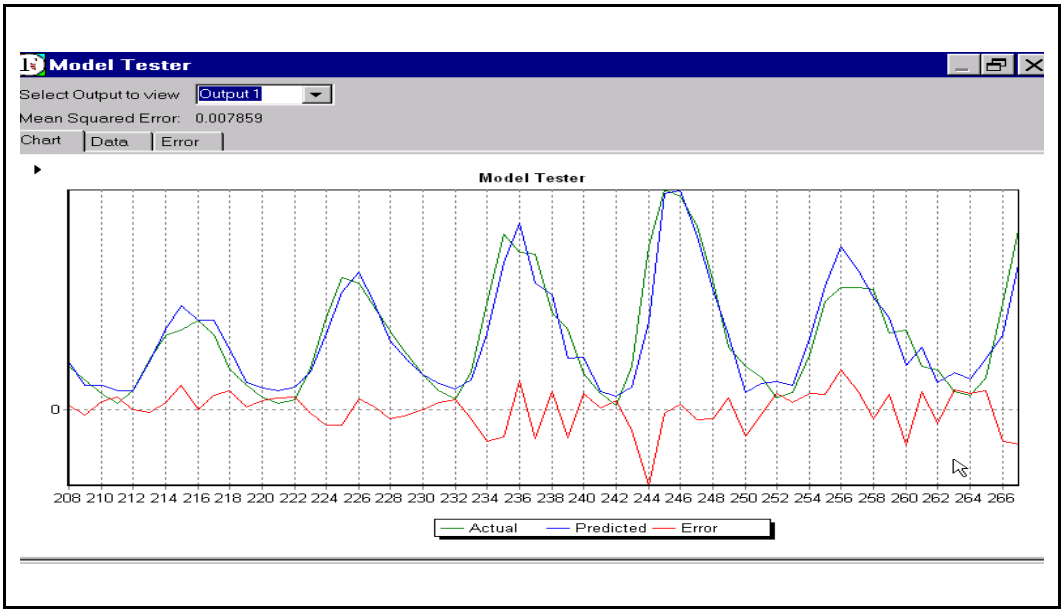

Figure 6 Predicting sunspots on unseen data.

Figure 6 shows predictions using dynamic local linear regression on unseen data, using all the other data for model building. As we can see this is really rather good, especially considering that the test data contains a totally unprecedented sequence of increasing sunspot maxima.

Example: *The Mackey-Glass equation*.

The Mackey-Glass equation is a time delayed differential equation which produces a chaotically evolving continuous dynamic system. The version used to generate the data in *MGls500.asc* is given by

$$\frac{dx}{dt} + 0.1x(t) = \frac{0.2x(t-\tau)}{1 + x(t-\tau)^{10}} \qquad \textbf{(2)}$$

where $\tau = 30$ (N.B. $\tau > 17$). We integrated the equation over $t \in [0, 5000]$ with the initial condition $x(t) = 2$. No noise was added. The graph of the function over $t \in [0, 1000]$ is given in **Figure 7**.

The file *MGls500* was created by writing out the values of $x(t)$ at $t = 10, 20, 30, ..., 5000$ ($\Delta t = 10$) giving 500 data points of a chaotic time series. If smaller time steps are taken then using several previous values to predict $x(t)$ we find that the resulting Γ is extremely small, indicating that predicting this function *small* steps ahead is very easy.

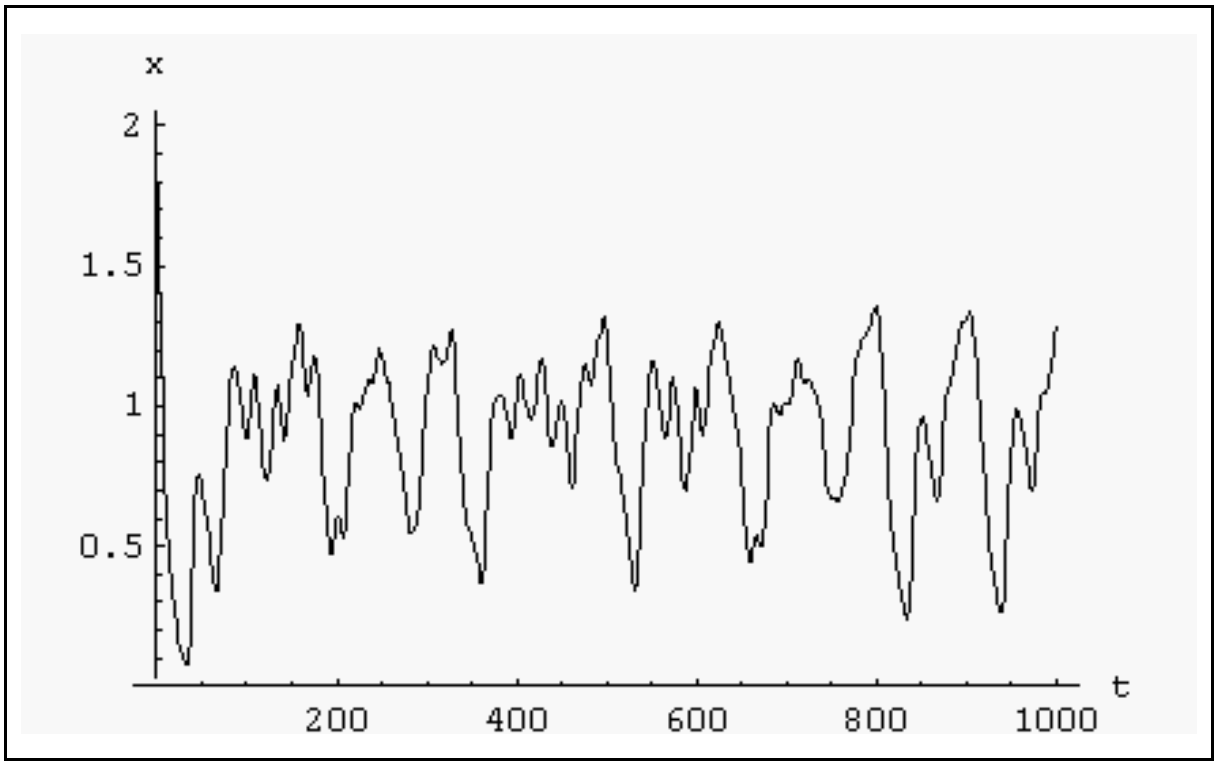

Figure 7 The Mackey-Glass time series.

Suppose we examine the prospect of trying to predict $x(t)$ using the last 5 values. Since $2^5 - 1 = 31$ it is no problem to do a full embedding search. We find that the best embedding (i.e. the embedding with smallest Γ) is 11110, which means that we predict $x(t)$ using $x(t\text{-}2.\Delta t)$, $x(t\text{-}3.\Delta t)$, $x(t\text{-}4.\Delta t)$ and $x(t\text{-}5.\Delta t)$.

On this basis we generate the results in **Table 3**. It is interesting to note that the full embedding search obtained the best model by omitting $x(t\text{-}1.\Delta t)$. Why is this? In the original time delay equation the value $x(t)$ depends on the value $x(t\text{-}30)$ and on the derivative. $x(t\text{-}20)$ is probably needed to estimate the derivative at $x(t\text{-}30)$ but $x(t\text{-}10)$ is not needed at all, as the software discovered.

Given a reasonable amount of data, predicting a chaotic time series a small time ahead is usually not too difficult. The problem is to predict a long way ahead. Here $\Delta t = 10$ is a modest time ahead.

The embedding 11110 provides a four input/one output set of I/O pairs. The low noise level $\Gamma \approx 0.001$, combined with the rapid fall off of the M-test graph, and $V_{ratio} \approx 0.016$ indicates the existence of a reasonably accurate smooth model. Taken together these are clear indicators that it should be quite straightforward to construct a predictive model using around 500 data points with an expected MSE around 0.001.

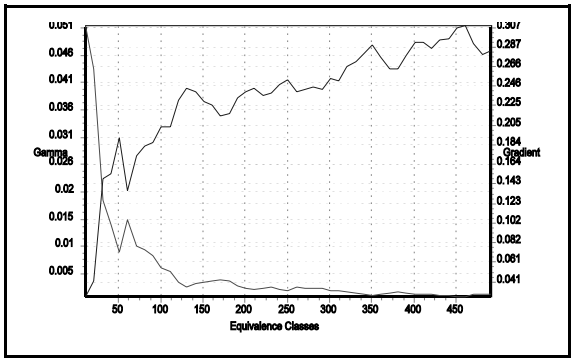

Figure 8 M-test for *MGls500* using the embedding 11110. The upper trace is the slope *A*.

Table 3 Basic results for *MGls500* using 11110.

MGls500 with embedding 11110 (p_{max} = 10)	
True noise	0
Γ	0.001304373
A	0.2795143
SE	0.0004096
V_{ratio}	0.016027830

It is now straightforward to generate a locally linear regression model using the given embedding. A locally linear regression model was trained on the first 400 points from the 495 points generated using the embedding 11110.

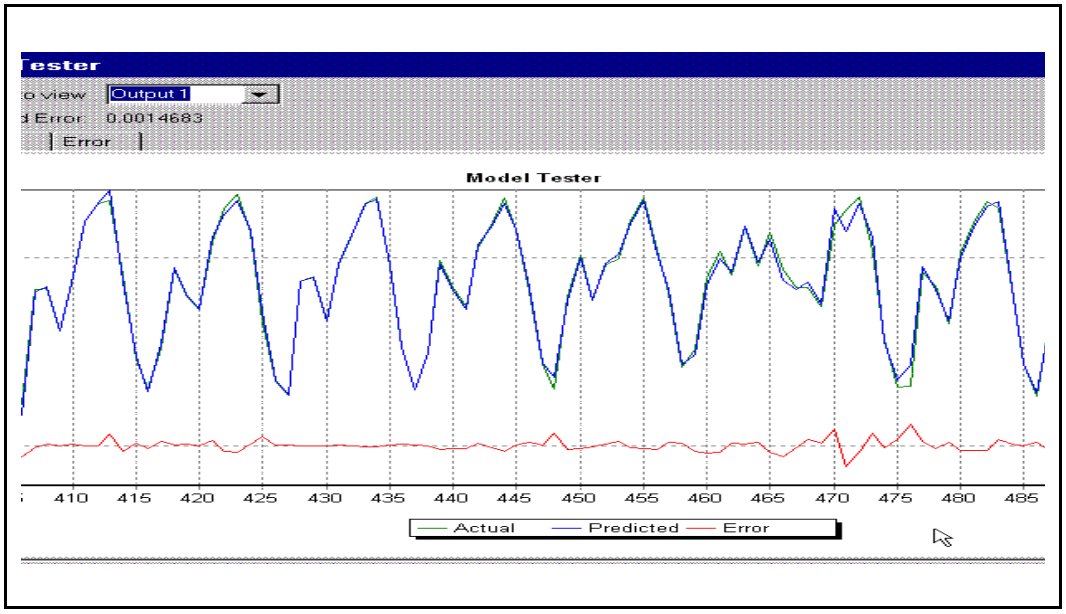

Figure 9 Predicting the Mackey-Glass time series on unseen data.

Figure 9 shows the results of testing the model on the remaining 95 previously unseen points. The resulting MSError is 0.001468 which is remarkably close to the value predicted by the Gamma test.

Using the Gamma test software we could also investigate how the error of prediction is liable to vary as Δt increases, where we search for the best embedding for each Δ*t*.

Conclusions

We have briefly illustrated the utility of the Gamma test in constructing non-linear models. All the results presented here were generated using the *winGamma*™ software. What is less apparent in such a presentation is the extreme ease of with which models can be identified and constructed using this software.

References.

[Stefánsson 1997] Aðalbjörn Stefánsson, N. Končar and Antonia J. Jones. *A note on the Gamma test*. Neural Computing & Applications **5**(3):131-133, 1997. ISSN 0-941-0643.

[Končar 1997] N. Končar. *Optimisation methodologies for direct inverse neurocontrol*. Ph.D. Thesis. Department of Computing, Imperial College, London 1997.

[Oliveira 1999] Ana Oliveira. *Synchronization of chaos and applications to secure communications*. Ph.D. Thesis. Department of Computing, Imperial College, London 1999.

[Tsui 1999] Alban P. M. Tsui. *Smooth data modelling and stimulus-response via stabilisation of neural chaos*. Ph.D. Thesis. Department of Computing, Imperial College, London 1999.

[Weigend 1990] Andreas S. Weigend, Bernardo A. Huberman and David E. Rumelhart. *Predicting the Future: a Connectionist Approach*. International Journal of Neural Systems, **1**:193-209, 1990.