New Tools in Non-Linear Modelling and Prediction

by

Antonia J. Jones
Department of Computer Science
Cardiff University
University of Wales, PO Box 916, Cardiff CF24 3XF,
UK
Submitted to: Computational Management Science

First draft: 10 September 2002

Contents

Abstract

In this paper we give an account of a new change of perspective in non-linear modelling and prediction as applied to smooth systems. The core element of these developments is the *Gamma test* a non-linear modelling and analysis tool which allows us to examine the nature of a hypothetical input/output relationship in a numerical data-set. In essence, the Gamma test allows us to efficiently calculate that part of the variance of the output which cannot be accounted for by the existence of any smooth model based on the inputs, even though this model be unknown. A key aspect of this tool is its speed: the Gamma test has time complexity $\mathrm{O}(M \log M)$, where M is the number of data-points. For data-sets consisting of a few thousand points and a reasonable number of attributes, a single run of the Gamma test typically takes a few seconds.

Around this essentially simple procedure a new set of analytical tools has evolved which allow us to model smooth non-linear systems directly from the data with a precision and confidence that hitherto was inaccessible. In this paper we briefly describe the Gamma test, its benefits in model identification and model building, and then in more detail explain and motivate the procedures which facilitate a Gamma analysis.

We briefly report on a case study applying these ideas to the practical problem of predicting level and flow rates in the Thames valley river basin.

Finally we speculate on the future development and enhancement of these techniques into areas such as datamining and the production of complex non-linear models directly from data via graphical representations of process charts and automated Gamma analysis of each input-output node.

1 Introduction

Over the last seven years a quiet revolution has been taking place in the subject of data driven, non-parametric, non-linear modelling and prediction.

Suppose we are given a set of input-output data

$$\{x_1(i), ..., x_m(i), y_i\} = \{(\boldsymbol{x}_i, y_i) \,|\, 1 \leq i \leq M\}\,, \tag{1}$$

where we think of the vector $\boldsymbol{x} = (x_1, \ldots, x_m) \in \mathbb{R}^m$ as the *input*, confined to some closed bounded set $C \subset \mathbb{R}^m$, and (without loss of generality) the corresponding scalar $y \in \mathbb{R}$ as the *output*, and asked: to what extent is the output determined by the input?

If we have sufficient *a priori* knowledge of the system under investigation we can use the method of *parametric statistics* for model construction, where we make some specific assumptions about the form of the relationship between $\boldsymbol{x}$ and y and attempt to find the 'best fit' for the parameters in the hypothesized relationship. In many cases however we do not have sufficient information with which to construct a parametric model and traditionally we are forced to study quantities such as correlations, auto–regressions or co–variances, all of which are likely to be rather crude estimators of the 'average' causal relations between the input variables and the output we seek to predict.

Data–derived modelling techniques seek to construct models of a system directly from a set of measurements of the system's behaviour, without assuming any *a priori* knowledge of the underlying logical rules or equations that determine this behaviour. Without further assumptions the class of potential models is enormous, ranging from logic functions through rule based systems and probabilistic models to parameterized functions.

Historically the construction of non-linear models from sampled data has been very much a subjective process. This in part stems from the enormous diversity of possible modelling techniques and the difficulty of assessing the quality of the data. For example, for data describing discrete input attributes with continuous or discrete outputs one might consider a rule based system of modelling such as a decision-tree approach [Quinlan, 1986]. At the other extreme, input and output variables are continuous and, if the unknown process being described by the data is suspected to be non-linear, one might consider a modelling technique based on neural networks (see [Bishop, 1996] for an excellent up-to-date account), but this itself historically has been a somewhat hit and miss procedure. Moreover, the validation of the chosen modelling technique is frequently purely empirical – the best possible non-linear model is built using the selected technique. If these attempts are successful then the original choice is deemed to be vindicated, otherwise an alternative technique is tried or the failure simply ascribed to 'bad data'.

A dispassionate observer might be forgiven for concluding that this state of affairs is somewhat unsatisfactory, perhaps lacking in good scientific methodology.

No single approach can address all of these issues – the extraction of good non-parametric models from data of diverse types and diverse quality is a very broad problem. However, some aspects of these issues can be addressed in a more systematic fashion.

In this paper the focus is on *smooth* models of *continuous* variables. We do not consider the case of discrete input or output variables, although in some circumstances some of the techniques described here might conceivably be applicable. One consequence of this decision is that we largely sidestep the question of 'what type of model should be constructed'.

This work started in 1995 (first reported in [Končar, 1997], [Aðalbjörn Stefánsson et al., 1997]), with the conjecture that a very simple algorithm, the Gamma test, could be used to estimate directly from a given input/output set of data the extent to which the data derived from an underlying smooth model, even though this model was unknown. This is more remarkable than might first be apparent.

- If linear regression is characterized as the ability to provide an estimate of 'goodness of fit' against the class of linear models, then the Gamma test is *non-linear regression*, because it provides an estimate of 'goodness of fit' against the class of non-linear smooth models which have bounded partial derivatives.

The potential advantages of having available a data-derived estimate of noise

were apparently largely overlooked, in all probability, first because efforts were mostly focussed on the approximate reconstruction of the unknown functional mapping and second most people seemed to have assumed that derivation of a noise estimate was impossible *because* the underlying functional mapping was unknown.

In 2001 we were eventually able to supply a proof of the original conjecture in a wide class of situations [Evans, 2002], [Evans and Jones, 2002], [Evans et al., 2002] and the tool was thus promoted from a useful heuristic algorithm to a validated tool for statistical analysis. Far from concluding the theoretical analysis, these two papers have merely opened the door to what promises to be a new rich area of investigation.

Although our recent efforts have mostly focussed on developing the underlying mathematical theory of the Gamma test and its extensions, work on applications has been continuous over the period 1995-2002 and these practical experimental investigations have resulted in the development of various techniques, or protocols, in the application of the Gamma test.

To apply, or experiment with, the Gamma test a software implementation is required. If one is dealing with relatively small data sets, where the number of data points M is at most a few hundred, and the requirement is for a few simple Gamma tests, then the time complexity of the algorithm is not critical and it can be implemented very easily with time complexity $\mathrm{O}(M^2)$ in half an hour of programming effort. However, if one wants to process large, high dimensional, data sets, and perform many Gamma tests, then the time complexity of the implementation becomes critical and more programming effort is required to produce a fast running $\mathrm{O}(M \log M)$implementation. We packaged our own implementation as a software product[1] described in [Durrant, 2001] called *winGamma*^{TM} and the illustrative results generated for this paper were produced using this software package.

We refer the reader to the introductory section of [Evans and Jones, 2002] for a more detailed description of the Gamma test and some of the generic applications. Here, for completeness, in the next section we develop some notation and then simply state the algorithm. We then proceed to describe some useful analytical tools which facilitate the whole process of model identification and model building, briefly describe the highlights of a case study on Flood prediction, and finally speculate on future developments.

1.1 The Gamma test

The Gamma test was first briefly reported in [Končar, 1997] and [Aðalbjörn Stefánsson et al., 1997], and later discussed and used in [Chuzhanova et al., 1998], [de Oliveira, 1999], [Tsui, 1999], [Durrant, 2001], [Jones et al., 2002], and [Tsui et al., 2002].

Suppose we are given an input-output data set of the form (1). If y is

[1] Available under under licence from the Department of Computer Science, Cardiff University.

a vector we can treat each component separately and, with very little extra computational overhead, return a Gamma statistic for each component of y.

As the terms 'input' and 'output' might suggest we assume that in some sense or other y is 'determined' by $\boldsymbol{x}$, or by some subset of the components of $\boldsymbol{x}$. For our purposes a *model* is an algorithm constructed from the initial data set $\{(\boldsymbol{x}_i, y_i), 1 \leq i \leq M\}$ which, when given a previously unseen query vector $\boldsymbol{x}$, can be used to predict the associated output y. It is an implicit requirement that the process of model construction and query should be computationally efficient. In practice this means that at worst model construction should scale as $\mathrm{O}(M \log M)$ and querying the model for a single input vector should at worst scale as $\mathrm{O}(\log M)$ as $M \to \infty$.

Suppose also that the underlying relationship between the input vector $\boldsymbol{x}$ and the output y is highly *non-linear* and we have no *a priori* knowledge which we can use to construct a parametric model.

In [Aðalbjörn Stefánsson et al., 1997] a radical new approach, termed the Gamma test, to this general problem is outlined. The idea is quite distinct from earlier attempts at non-linear analysis. Here, rather than pre-suppose some particular parametric form for the underlying non-linear model, we suppose that *it belongs to some general class of functions*. In particular we suppose that the underlying relationship is of the form

$$y = f(x_1 \dots x_m) + r \tag{2}$$

where f is a suitably smooth function and r is a random variable which represents noise[2]. Without loss of generality we can assume that the mean of the distribution of r is zero (since any constant bias can just as well be subsumed into the unknown function f) and we may wish to make some further reasonable restrictions, for example that the variance $\mathrm{Var}(r)$ of r is bounded. Thus the domain of possible models now becomes restricted to (say) the class of functions which have bounded first and second partial derivatives.

Despite the fact that f is unknown, under these circumstances, and making some other (fairly reasonable) assumptions, the Gamma test provides an estimate for $\mathrm{Var}(r)$. Moreover, this estimate can be derived in $\mathrm{O}(M \log M)$ time, where the implied constant depends on the dimension m of the input space. The estimate of that part of the variance of the output that cannot be accounted for by a smooth data model is called the *Gamma statistic*, denoted by Γ. The Gamma test is non-parametric and the results apply regardless of the particular methods used to subsequently build a model.

The Gamma test estimates $\mathrm{Var}(r)$ in $\mathrm{O}(M \log M)$ time by first constructing a kd-tree using the input vectors $\boldsymbol{x_i}(1 \leq i \leq M)$ and then using the kd-tree to construct lists of the kth $(1 \leq k \leq p)$ nearest neighbours $\boldsymbol{x}_{N[i,k]}(1 \leq i \leq M)$ of $\boldsymbol{x}_i$. Here p is fixed and bounded, typically $p \approx 10$. The algorithm next computes

$$\delta_M(k) = \frac{1}{M} \sum_{i=1}^{M} |\boldsymbol{x}_{N[i,k]} - \boldsymbol{x}_i|^2 \tag{3}$$

[2]Actually, the notion of 'noise' we use here can be quite subtle and we shall discuss this later in more detail.

where $|.|$ denotes Euclidean distance[3], and

$$\gamma_M(k) = \frac{1}{2M}\sum_{i=1}^{M}(y_{N[i,k]} - y_i)^2 \tag{4}$$

where $y_{N[i,k]}$ is the output value associated with $\boldsymbol{x}_{N[i,k]}$ (note that $y_{N[i,k]}$ is not necessarily the kth nearest neighbour of y_i in output space). Finally the regression line $\gamma_M(k) = \Gamma + A\delta_M(k)$ of the points $(\delta_M(k), \gamma_M(k))(1 \leq k \leq p)$ is computed and the vertical intercept Γ returned as the estimate for $\mathrm{Var}(r)$. The slope parameter A is also returned as this normally contains useful information regarding the complexity of the unknown surface $y = f(\boldsymbol{x})$.

The main result of [Evans and Jones, 2002] is that if C is a compact convex body in $\mathbb{R}^m$ and data samples $\boldsymbol{x} \in C$ are selected with a smooth positive sampling density ϕ, then the number Γ returned by the algorithm converges in probability to Var(r) as $M \to \infty$. [4]

How the computation of the kth nearest neighbour lists is implemented is a significant factor in both the speed of this algorithm and the time-complexity scaling properties as M increases. Using the kd-tree data structure first discussed by [Bentley, 1975], (see also [Friedman et al., 1979]), the near neighbour lists can be constructed in $\mathrm{O}(M \log M)$ time, although there are now numerous fast methods for constructing kth nearest neighbour lists.

We note, in passing, that in principle it is possible to parallelize the Gamma test algorithm by partitioning the input space appropriately and combining the resulting tree structures as a final step. In practice this has never proved necessary for the data sets we have considered, although there other aspects of a full Gamma analysis which it may be worthwhile to consider parallelizing (see section 3.3.3).

1.1.1 The slope constant A

The Gamma test algorithm also returns the slope A of the regression line. Following equation (7.6) of [Evans and Jones, 2002] if we assume that the near neighbour vectors $\boldsymbol{x}_{N[i,k]} - \boldsymbol{x}_i$ are independent of the function f then

$$A = A(M,k) = \frac{1}{2}\mathcal{E}_\phi(|\nabla f(\boldsymbol{x}_i)|^2 \cos^2\theta_i) \tag{5}$$

where θ_i denote the angle between the vectors $(\boldsymbol{x}_{N[i,k]} - \boldsymbol{x}_i)$ and $\nabla f(\boldsymbol{x}_i)$.

The phrase 'independent of the function f' deserves some explanation. In the theory of [Evans and Jones, 2002] the sampling distribution Φ, with density function ϕ, in input space is considered to be unrelated to the function f we

[3] Other metrics can be used, and which is most appropriate may be application dependent. However, since any two metrics in Euclidean space are equivalent to within a positive constant, the asymptotics of interest remain invariant with respect to a change of metric. Nevertheless, the *rate* of convergence may be optimized by an improved choice of metric.

[4] It seems likely that this result might be strengthened so that the convergence is 'almost surely'.

seek to model. However, in an iterative recursive model of a chaotic dynamical system, as described later in section 3.4, the ergodic sampling process in input space is driven by the very function we are trying to model (as in equation (9)), so it is in principle possible that the near neighbour vectors $\boldsymbol{x}_{N[i,k]} - \boldsymbol{x}_i$ are not statistically independent of the gradient vectors $\nabla f(\boldsymbol{x}_i)$, in which case the step from equation (7.5) of [Evans and Jones, 2002] to equation (7.6) would not be justified.

Under the conditions of a smooth positive sampling density ϕ it seems clear that the limiting value of $A(M,k)$ is actually independent of k (although for ergodic sampling over a chaotic attractor this is not always the case - see Figure 11 [Evans and Jones, 2002]).

If we further assume that $|\nabla f(\boldsymbol{x}_i)|^2$ is independent of $\cos^2\theta_i$ then

$$A = \frac{1}{2}\mathcal{E}_\phi(|\nabla f(\boldsymbol{x}_i)|^2)\mathcal{E}_\phi(\cos^2\theta_i) \tag{6}$$

If $m = 1$ then θ_i takes only the values 0 and π. If we assume it takes these values with equal probability then $\mathcal{E}_\phi(\cos^2\theta_i) = 1$. If $m > 1$ then one might assume instead that the θ_i are uniformly distributed over $[-\pi, \pi]$ and then $\mathcal{E}_\phi(\cos^2\theta_i) = 1/2$. Asymptotically we then have

$$A = \begin{cases} \frac{1}{2}\mathcal{E}_\phi(|\nabla f(\boldsymbol{x}_i)|^2) & \text{if} \quad m = 1, \\ \frac{1}{4}\mathcal{E}_\phi(|\nabla f(\boldsymbol{x}_i)|^2) & \text{if} \quad m \geq 2. \end{cases} \tag{7}$$

Whilst these diverse assumptions are not always true, for example, see section 8(c) of [Evans and Jones, 2002], equation (7) is remarkably robust in practical data analysis. Thus the slope A returned by the Gamma test gives a crude estimate of the complexity of the unknown surface f we seek to model.

1.1.2 Local versus global

We have assumed in our discussion that the noise distribution is homogeneous (i.e. constant or fixed) across the input space, so *a fortiori* the noise variance is constant. Suppose we had sufficient data to evaluate a Gamma statistic in a small localised region of the input space. Then this estimate for the local noise variance will give essentially (allowing for sampling variation) the same result as the Gamma statistic evaluated globally. This reflects the 'noise homogeneity' assumption made in the proof of the Gamma test presented in [Evans and Jones, 2002].

However, if the noise variance is non-homogeneous across input space, then localised Gamma statistics will vary, but (subject to reasonable conditions) the global Gamma statistic will represent an *average* (with respect to the sampling density ϕ) of the local noise variances across the whole space (an example is given in section 3.2).

Thus the Gamma statistic can be viewed as *either* local *or* global, depending on the region of the input space from which the data used to compute it is drawn. Mostly we use a global estimate, because we rarely have sufficient data to afford the luxury of computing many local Gamma statistics.

1.2 Implications of the Gamma test

In practice when building a non-linear model from data we want to know the answers to questions such as:

- Do the inputs determine the output by a smooth model?
- Given an input vector $\boldsymbol{x}$ how accurately can we predict the output y?
- How many data points does one 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?

Given sufficient data all of these questions are often easily answered by the Gamma test. For example, if the Gamma statistic is large (compared with the variance of y) then it is not likely that the outputs are determined from the inputs by a smooth model, whereas if the Gamma statistic is small or close to zero then this becomes more likely. Thus $V_{ratio} = \Gamma/\mathrm{Var}(y)$ provides a scale invariant measure, normally in the range [0, 1], of the 'goodness of fit' of the data with respect to the class of smooth functions with bounded derivatives. Indeed, $1 - |V_{ratio}| = 1 - |\Gamma|/\mathrm{Var}(y)$ is closely analogous to the conventional r^2 statistic which estimates the extent to which the data fits a linear model, except here V_{ratio} estimates the extent to which the data fits a smooth *non-linear* model. We say V_{ratio} is 'normally' in the range [0, 1] because if $\mathrm{Var}(r)$ is equal to (or close to) zero (or M is too small), it can happen that Γ derived from the algorithm is negative (in which case we might replace Γ by $|\Gamma|$ for our estimate of $\mathrm{Var}(r)$), similarly it is possible that $\Gamma > Var(y)$.

- If V_{ratio} is close to zero the data is highly likely to derive from a smooth function f. If V_{ratio} is close to one then we should abandon any attempt to fit the data to a smooth model.

Of course, this assumes that we have some way to measure our confidence in the Gamma statistic as an estimate for $\mathrm{Var}(r)$, and this is an issue we examine in section 2.2.2.

Unlike the conventional 'least-squares-fit' of linear regression, which provides not just the goodness of fit, in terms of the standard error, but also an estimate for the actual underlying linear function, the Gamma test provides only an estimate for the 'non-linear goodness of fit'. Although the Gamma test gives very little information about the best fitting function from the allowed class, it nevertheless *facilitates* the construction of such a model. To actually build the model we use information from the Gamma test combined with other non-parametric techniques, such as local linear regression or neural networks which we discuss briefly in section 4.

1.2.1 How much data does one need?

The bane of the non-linear analyst's existence is the issue of how much data is available. One is frequently presented with data sets consisting of twenty inputs and a sample of less than one hundred points! Even with the restriction to models having bounded derivative, when one considers the enormous range of possible models and the fact that initially one has no idea of the intrinsic noise level present in the data, the issue plainly becomes more a question of 'Can I say *anything at all* at some reasonable confidence level?'. If the data reflects an intrinsically *linear* process, or one has some insight into the parametric form one might expect the relationship to take, then one may be able to perform an analysis that is meaningful. Otherwise, from the viewpoint of the non-linear analyst the situation is rather dire, and one may be reduced to suggesting the application of Bayesian statistics, or other forms of statistical analysis. [5]

Clients, government agencies, and sometimes scientists who should know better, persistently underestimate the importance of accurate, copious and timely data collection if they want to have confidence in the resulting analysis and predictions of non-linear phenomena. Whilst the Gamma test has introduced new possibilities into non-linear analysis, it is not a 'magic wand' that can make the very real problems inherent in the subject simply vanish. The golden rule is:

- You *cannot* predict or control a non-linear system that you do not measure.

To be effective in accurately estimating $\mathrm{Var}(r)$ the Gamma test requires the number of data points M to be relatively large; even for one dimensional input vectors and moderately noisy data we may require over a hundred data points before we can have confidence in our conclusions. Similarly to build a model which performs with the predicted $MSError$ we may need a comparably large number of points. With high dimensional input data we may, of necessity, require orders of magnitude more data. However, this should not surprise us, it is intrinsic to the nature of the undertaking. A linear model is determined by very few parameters and naturally requires less data to fit, whereas here we seek to quantify the goodness of fit against a huge class of potential models, each of which may be determined by an infinite set of parameters.

- What is surprising is that this can be done at all.

However, all this does not necessarily mean that the situation is entirely hopeless when faced with small data sets. We may still be able to exploit the fact that the Gamma test can be used to identify important input variables, because in such an analysis we are comparing numerous[6] Gamma statistics computed from the same data, and what is important is their mutual *relative magnitude*, not their precise values. In this way we may be able to simplify the

[5] This is not intended to suggest that "Bayesian statistics is the last refuge of the scoundrel". On the contrary, there are circumstances where Bayesian methods make a lot of sense.

[6] Often an enormous number - see section 3.

model by eliminating input variables. Each such variable eliminated makes the model simpler and enhances the value of the data set when it comes to model building. For example, we have obtained very interesting predictive models for UK Property Prices on relatively small data sets, see [Wilson et al., 2003].

Still, the fact remains that initially the principal application areas of the Gamma test are likely to be those where instrumentation is used to automate the gathering of relatively large quantities of data, such as physics, engineering, and signal processing. In practice, given current desktop computing power, the computations become time consuming at around $M \approx 100,000$.

1.3 Over-fitting and the hypothesis of bounded derivative

One simple measure of the performance of a model on unseen data for which the measured outputs are known is the mean-squared error over the test data. If $\{y_i : i \in U\}$ is a previously unseen set of measured values of an output and $\{\hat{y}_i : i \in U\}$ is a set of predictions for y_i then the mean-squared error *MSError* of the predictions is given by

$$MSError = \frac{1}{|U|} \sum_{i \in U} (\hat{y}_i - y_i)^2 \tag{8}$$

The *MSError* is not an ideal measure in all respects. For example, in financial applications we may have a time series model with a small *MSError*, but which shows no particular propensity to accurately anticipate turning points (a factor of particular interest to users in this context). This may be because of some limitation in the model or it may be, for example, because the underlying process has some characteristics of a random walk. In any event, small *MSError* may not be sufficient for the intended purpose in some situations.

When constructing a non-parametric non-linear model, the natural tendency is to try to minimize the mean squared error of the model against the training data. However, if there is significant intrinsic noise in the data then training down to a *MSError* close to zero (an option not available to us with a linear model) will be counter productive. If we attempt to do this then the only way that the model can accommodate the noise in the data is to evolve into a surface $y = h(\boldsymbol{x})$ with more and more folds. Since the region C in which the input vectors lie is supposed closed and bounded, this means that as progressively more data points are used in training, i.e. as M becomes large, the average of $|\nabla h|^2$ must increase, as the surface 'crinkles' in an attempt to accommodate the noise. Thus in data with significant noise, as we increase the number of training points, and in each case attempt to train to a *MSError* of *zero*, we should expect to see the mean square of the gradient of the model surface tend to infinity. [7]

- But the fundamental assumption we work on, is that the true surface

[7] Indeed, this is exactly what happens to the slope parameter A if we run the Gamma test on data in which the output variable y is randomly generated (of course, Γ approaches $\mathrm{Var}(y)$ in this case).

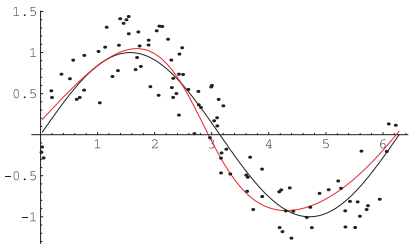

Figure 1: Model (red) trained ($M = 100$) to a mean squared error of 0.0786 (*Courtesy of* [Evans, 2002]).

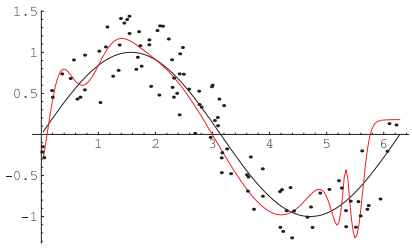

Figure 2: Model (red) trained ($M = 100$) to a mean squared error of 0.056 (*Courtesy of* [Evans, 2002]).

$y = f(\boldsymbol{x})$, which we seek to approximate with the model $y = h(\boldsymbol{x})$, is *fixed* with bounded derivative and so cannot have arbitrarily large $|\nabla f|^2$.

Thus there must come a point where the model is effectively starting to memorize the noise, and it will then perform poorly when tested against data not used in the training process. Indeed, it is generally well understood that driving the *MSError* of the model h as close as possible to zero, regardless of the noise level in the data, is counter-productive, and this is referred to as *over-fitting* the model.

Example: The noisy sine data. Suppose we define $f(x) = \sin(x)$, generate $M = 500$ uniformly distributed points $x \in [0, 2\pi]$ and construct the corresponding output values y by adding a uniformly distributed noise component with a variance of 0.075 to each of the $f(x)$ values.

Figure 1 shows the model obtained by training a $1 \to 5 \to 5 \to 1$ neural network[8] to a mean squared error of 0.0786, which is close to the Gamma statistic of $\Gamma = 0.0795$ derived with $p = 10$, on $M = 100$ points of noisy sine data, while Figure 2 shows the model obtained by training a $1 \to 5 \to 5 \to 1$ neural network to a mean squared error of 0.056, significantly below the Gamma statistic. In both cases the model is plotted in red while the function $f(x) = \sin(x)$ is plotted in black. Figure 2 vividly illustrates the effects of over-fitting. What is surprising in practice is how rapidly the model degenerates once training proceeds beyond the point where $MSError \approx \Gamma$.

From what has been said it should be plain that there is no point in training or building a smooth non-linear model beyond the point where the *MSError* over the training set falls much below $\mathrm{Var}(r)$. All things being equal, stopping training at this point this should result in a smooth model which has near minimal *MSError* when predicting the output using input data not seen in the model construction process[9].

[8]The notation indicates a feedforward network with 1 input node, 2 hidden layers of 5 nodes and 1 output node.

[9]One might go further than this: a *perfect model* in this context is one for which the *error distribution* on unseen data is identical to the noise distribution. In particular not just the variances should be the same but all the higher moments (see [Durrant, 2001]).

Thus we can now dispense with the necessity of spitting the data into three sets, and during training periodically testing a neural network against the second set, using a rise in *MSError* as a stopping criterion for training. Of course, it still remains prudent to retain the third unseen test set to validate the model. We should expect the performance of the model on the unseen test set to produce a *MSError* which is also close to the Γ value. Moreover, this is going to be close to the best performance obtainable by a smooth model when tested against the (noisy) data, although the model itself may be near perfect.

- Thus one problem of model construction solved by the Gamma test is at what point to cease training.

It is also useful to know just how much training data is required to produce a good model that will perform with this *MSError*. In practice the situation appears to be that if the M-test (see section 2) graph is stabilizing and has reached a particular $\Gamma = \Gamma_0$ at $M = M_0$, then by using these M_0 points we can train down to a *MSError* that closely approximates Γ_0 and obtain a model which performs at this level when tested against unseen data drawn from the same process that generated the training data.

- Thus another problem of model construction which the Gamma test helps with in practice, is that it indicates just how much data we need to produce a model that performs with the appropriate $MSError$.

If (as M becomes very large) the Gamma statistic indicates that $\mathrm{Var}(r)$ is essentially zero, then in principle by taking progressively more training data we can reduce the *MSError* of our model as close to zero as we wish without risk of over-fitting. However, this is a situation more likely to arise in artificially constructed demonstrations rather than in the analysis of real data sets.

2 Analytic tools

Whilst a single Gamma test provides a useful first insight into the nature of the data, if we wish to provide a comprehensive data analysis we require more supporting tools, which may involve a large number of Gamma tests. In this section we describe some of the basic data analysis tools that have developed around a fast implementation of the Gamma test.

2.1 Nearest neighbour analysis: the regression line and scatter plot

The Gamma test computes Γ by means of a linear regression on the points $(\delta_M(k),\ \gamma_M(k))$ $(1 \le i \le M)$, where typically $p \approx 10$, and examination of this regression line can be revealing. It is even more revealing if it is combined with a scatter plot of the points from which it is computed, i.e. $(|\boldsymbol{x}_{N[i,k]} - \boldsymbol{x}_i|^2, \frac{1}{2}(y_{N[i,k]} - y_i)^2)$, $(1 \le k \le p, 1 \le i \le M)$. If M is very large

a random selection of these points suffices for the scatter plot. For simplicity we refer to these points as (δ, γ) pairs. In low noise data we expect that as $\delta \to 0$ then $\gamma \to 0$ and we should see a characteristic wedge shaped blank in the distribution in the lower left hand corner of the scatter plot. The main features of the **Gamma regression line and scatter plot** are illustrated in Figure 3.

If the data is low precision then vertical (or horizontal) bands will be apparent in the scatter plot, since there will be many pairs $(\boldsymbol{x}_{N[i,k]}, y_i), (\boldsymbol{x}_i, y_i)$ for which $\delta = |\boldsymbol{x}_{N[i,k]} - \boldsymbol{x}_i|^2$ (or $\gamma = \frac{1}{2}(y_{N[i,k]} - y_i)^2$) take the same value.

High density of plot points in the shaded region corresponds to high noise on y. Points with high δ and relatively high γ may correspond to outliers. It is sometimes helpful to isolate and examine such points.

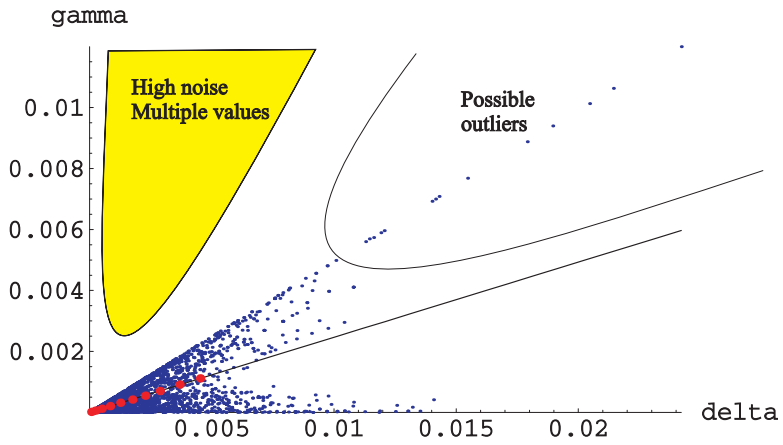

Figure 3: Main features of the scatter plot and associated regression line with plot points (red).

Example: Comparison of scatter plots. To illustrate the interpretation of scatter plots we return to the noisy sine data. Initially, if we add no noise, we obtain Figure 4 - note the wedge shaped absence of points in the lower left corner. If we now add noise with a variance of 0.075 we obtain Figure 5. Here the blank wedge has degraded, since noise is present, even when δ is small we may still obtain a relatively large γ.

The Gamma regression line and scatter plot can sometimes reveal other unexpected characteristics of the data. If the data is generated by a composite dynamical system, e.g. a non-stationary system with more than one dynamical regime, this can be revealed by regression line *plot points* which might lie, for example, on two quite distinct lines rather than a single line. Each such line would essentially stem from a distinct f applicable to a particular sub-region of the input-space, so these lines could exhibit quite different A values as well as possibly different Gamma statistics. This is a different type of local/global variation

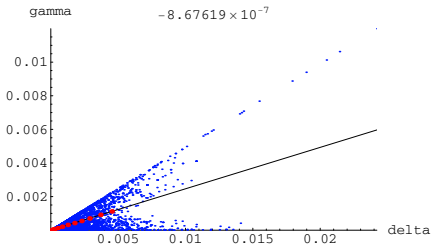

Figure 4: Sine data (no noise). Scatter plot ($M = 500$, $p = 10$) and regression line.

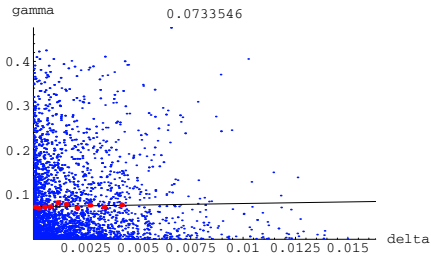

Figure 5: Sine data, added-noise variance 0.075. Scatter plot ($M = 500$, $p = 10$) and regression line.

from that of inhomogeneous noise discussed in section 1.1.2. Observing this, one might then attempt to separate the input space into two distinct regions, each region corresponding to a different dynamical regime. Building separate models for each of these regions can then lead to much improved analysis and prediction.[10]

2.1.1 The optimal value of p for estimating Γ

What is the optimal value of p in any particular situation? One can examine the *standard error* (SE) of the regression line fit as in Figure 6. Here we have used the noisy sine data with $M = 500$ illustrated in Figure 1. For this experiment we see that the Γ estimate is quite robust with respect to p over the range $8 \leq p \leq 65$. The SE is minimised when $p = 17$ for which $\Gamma = 0.07417$ (the actual noise variance is 0.075). Thus in this case selecting the value of $p \geq 8$ that minimises the SE does indeed produce a better estimate for $\mathrm{Var}(r)$ than the default $p = 10$. When one is data poor such considerations may become important. Of course, when we have plenty of data to hand we can afford to take p proportionately larger, but it is a remarkable fact that over many thousands of experiments we have found that taking $p = 10$ usually gives quite good results, and that time spent seeking to optimise p is often not worth the normally marginal gains.

2.1.2 Large p scatter plots can sometimes be useful

Although selecting larger p is of dubious value (indeed it may be counter productive) when computing the Gamma statistic, it can nevertheless produce interesting results in terms of the scatter plot.

For example, if we take a low frequency sine curve sampled over one period and then add a small amplitude modulation with several times the frequency

[10] This situation was first observed by Mr. J. Terry, a control engineer who encountered it in practice.

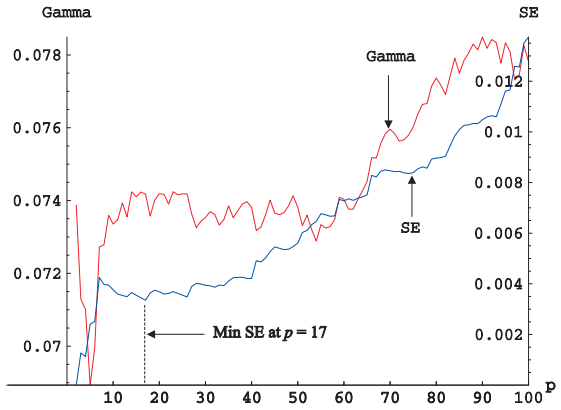

Figure 6: Increasing the number of near neighbours p for the noisy sine data ($M = 500$): Γ (red) is indicated on the left scale and SE (blue) on the right scale.

then the scatter plot for large p will quite clearly reveal the underlying periodicities. One can also understand that with inadequate data set size (i.e. M too small) the high frequency component would give the appearance of noise when the curve is sampled at low density.

2.2 How reliable is the Gamma statistic as an estimate of Var(r): the M-test

The conventional goodness of fit, the standard error (SE), of the Gamma plot regression line can be useful as an indicator of the reliability of the Gamma statistic as an estimate for Var(r). However, theoretical analysis of SE as a measure of confidence is complicated by the fact that the convergence of Γ to Var(r) as $M \to \infty$ is convergence in probability.

If one has adequate data one simple way to approach this issue is to plot Γ for increasing M. If the graph stabilizes then we can have some confidence that our estimate is reasonably accurate. We call this an **M-test**.

Example: Comparison of M-tests. To illustrate the interpretation of M-test graphs we again return to the noisy sine data. Initially, if we add no noise, we obtain Figure 7 which illustrates asymptotic convergence of Γ to zero. If we now add noise with a variance of 0.075 we see asymptotic convergence to Var(r) = 0.075 in Figure 8.

2.2.1 What factors affect the rate of convergence?

It is natural to ask: what factors affect the rate of convergence of Γ to Var(r) as M increases, and the amount of data that one needs to perform an analysis and subsequently build a model that performs with the expected $MSError$?

There are two principal factors:

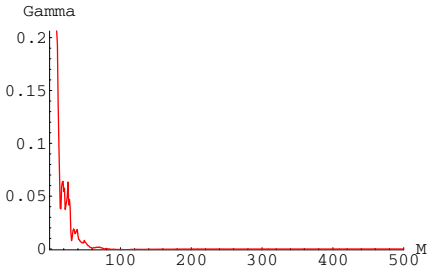

Figure 7: M-test ($p = 10$) on sine curve data with no added noise.

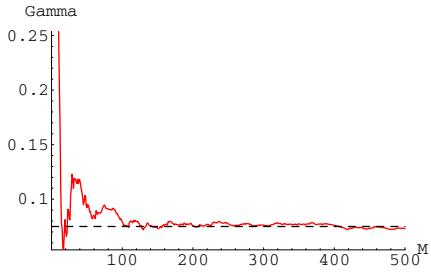

Figure 8: M-test ($p = 10$) on sine curve data with added-noise variance 0.075 (indicated by the dashed line).

- The complexity of the unknown surface $y = f(\boldsymbol{x})$.

We cannot hope to accurately estimate noise if the sampling density is low in comparison with the complexity of the underlying function.

For a complex surface an inadequate sampling density may produce a misleadingly high Gamma statistic, the M-test graph will not have stabilized. In general, the more complex the unknown surface $y = f(\boldsymbol{x})$ the more data points will be required to stabilize the M-test graph.

- The noise level Var(r) and more generally the distribution of r.

Plainly if Var(r)is high then more data will be needed to stabilize the M-test graph (and to train a model). It is also the case that the particular form of the noise distribution will affect the rate of convergence. For example, the convergence will be faster for a uniform distribution with a given variance than for a normal distribution with the same variance. This is because rare occurrences of a large noise component in the normal distribution have the effect of slowing the overall convergence.

Another factor would seem to be the rate at which the expectation $\mathcal{E}_\phi(\delta_M(k))$ tends to zero as $M \to \infty$. In [Evans et al., 2002] we showed that for a smooth positive sampling density ϕ over a compact convex support $C \subset \mathbb{R}^m$ we have $\mathcal{E}_\phi(\delta_M(k)) \approx c(m,k,\phi)/M^{2/m}$ as $M \to \infty$, where $c > 0$ is a suitable constant. For fractional dimensional sets C of dimension $d < m$ the asymptotic convergence of $\mathcal{E}_\phi(\delta_M(k))$ would appear to be faster (see, for example Table 1 of [Evans and Jones, 2002]). Moreover, experimental evidence suggests that in such cases the M-test convergence is faster than would be the case if the sampling were over a set C having full dimension m. The application of the Gamma test to chaotic dynamical systems has been extensively illustrated elsewhere, see [Tsui et al., 2002] and [Jones et al., 2002].

2.3 Phenomenological considerations: Dynamical stability

We also need to consider whether the phenomenon we seek to model has temporal stability over the time scale of our analysis and prediction, i.e. are the underlying processes or dynamics fixed, or themselves subject to change. The analysis on which the Gamma test and associated techniques are based assumes stability both in the noise distribution and in the underlying dynamics or functional mappings.[11]

In many situations the extent to which the theoretical preconditions are satisfied is unknown, and the quickest way to determine the utility of the Gamma test may be simply to try it. With the Flood Forecasting application discussed in section 5 we can be reasonably confident that, unless the physical characteristics of the system change, the laws of physics will ensure a degree of stability in the water transport processes we are modelling. However, with financial models such as the Property Price study [Wilson et al., 2003] or the Crime Prediction study [Corcoran et al., 2003], the 'laws' which govern the process are likely to be more ephemeral and, although we may get quite accurate short term forecasts for a while, circumstances could readily change for a variety of reasons that are easy to imagine. Thus some care and considerable domain knowledge is required in interpreting such analyses or models.

On the other hand computing Gamma statistics progressively over suitable width time windows can at least give the analyst some insight into how stable the process being modelled actually is; just as the comparison of a prediction with the actual data value could form the basis of an alerting system, calling attention to the fact that something unusual is going on.

3 Selection of a suitable subset of the inputs

We have seen that the combination of Gamma regression line, scatter plot, and M-test can provide us with an estimate of $\mathrm{Var}(r)$, a qualitative degree of confidence in this estimate, and an indication of how much data we require to build or train a model which performs at the appropriate *MSError* level.

However, the Gamma test has other implications: it can be used for *model identification*. In this context we might say that the goal of model identification for a particular output is to choose a selection of input variables that best models the output y. Although mathematically the inclusion of an irrelevant variable in the list of inputs makes no difference to the fact that $f : C \to \mathbb{R}$ is a *function*, nevertheless in practice it is very important to eliminate counter-productive inputs. This reduces training time for neural networks and can substantially improve the resulting model.

Some input variables may be irrelevant, or subject to high measurement error, so their inclusion as inputs into the model may be counter-productive,

[11] Although, as we have indicated, useful results can still be obtained if these conditions are weakened somewhat.

leading to a higher effective noise level on the desired output. Since a single Gamma test is a relatively fast procedure it is possible (provided m is not too large) to find that selection of inputs which minimises the asymptotic value of the Gamma statistic and thereby make the 'best selection' of inputs. Thus

- Another issue addressed by the Gamma test is that it provides a new tool for the selection of the most useful input variables for predicting a particular output variable.

Remark. 'Predictively useful' should not be confused with 'causal'. For example, in economics leading indicators are frequently predictively useful but not necessarily causal.

The notion of 'effective noise' and the technique of using repeated Gamma tests as a tool for selecting the most useful predictive input variables raises some interesting questions.

3.1 What is noise in this context?

Noise on the output y may arise for a variety of reasons:

- Inaccuracy of measurements for both the inputs and the output.

Errors in measurement of the inputs can result in *effective noise* on the output, even when measurement of the output is itself subject to very low measurement error.

- Not all predictively useful variables that influence the output are included in the input.

This is another kind of effective noise which we illustrate in section 3.3.2.

It may also happen that

- The underlying relationship between input and output is not smooth, or there is no relationship.

For example, a linear congruence pseudo-random number generator produces a sequence which is completely deterministic and in which the next value depends in a quite simple manner on the previous values, but this relationship is not smooth and so we will obtain $V_{ratio} \approx 1$ if we run the Gamma test on a time series embedding of such data.

We originally introduced the noise variable r as a random variable with some distribution function which has mean zero and variance $\mathrm{Var}(r)$. This was more for the purposes of formulating a convincing theorem in [Evans and Jones, 2002] than a reflection of data analysis in practice. In fact it is often expedient to assign the following rather loose interpretation to r in equation (2):

- r represents that part of the data which cannot be accounted for by a smooth model.

There is rather subtle point to be made here: namely that the Gamma test is a powerful tool in the determination of useful input variables even in the *complete absence* of noise in the sense in which it was originally introduced. For consider the following example.

3.2 A zero noise example

Our hypothetical surface is part of the 3-dimensional cone illustrated in Figure 9. This example derives from [Durrant, 2001]. $M = 5000$ data points were sampled uniformly in input space to produce a 3-dimensional data structure $z = f(x, y)$ of two inputs (x, y) and one output z across the surface. Plainly, the height of the cone z is dependent on both the (x, y) co-ordinates.[12]

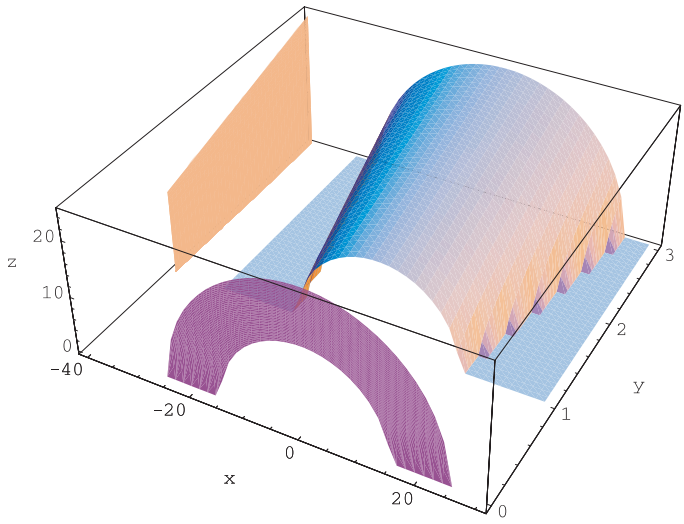

Figure 9: A conical function with 'flats'. The darkly shaded projection on the $x - z$ plane shows the effective noise from sampling in the x-dimension only. The lighter shaded projection on the $y - z$ plane shows the effective noise from sampling in the y-dimension only.

Now consider the effects of randomly sampling the input space and ignoring one of the coordinates. Figure 9 illustrates the effective noise distributions which arise from ignoring one or other coordinate. Insofar as these are statistical distributions at all, they arise from the sampling distribution Φ in input space,

[12]The function has discontinuities of the gradient on a set of measure zero. Although not strictly covered by the theoretical treatment of [Evans and Jones, 2002] this example is deliberately selected to suggest that such minor violations do not significantly affect the analysis.

Table 1: Gamma test results ($p = 10$) using uniformly sampled (x, y) data for the cone section illustrated in Figure 9.

	Γ	A	SE	V_{ratio}	x	y
$M = 500$	0.65958	10.69136	0.33860	0.011616	1	1
$M = 5000$	0.05517	15.41930	0.02814	0.000887		
$M = 500$	16.3829	−1.472708	0.59423	0.288518	1	0
$M = 5000$	15.9693	−114.6871	0.16735	0.256798		
$M = 500$	48.88932	902.94109	2.25273	0.860988	0	1
$M = 5000$	49.98136	−103961.7	0.81371	0.803740		

and not from an r-component in the data, for at this point no noise has been introduced.

If we run Gamma tests on the raw data first using both inputs and then ignoring one or other input we obtain Table 1. Here in the last column a 1 (or 0) indicates the inclusion (or exclusion) of a variable. These results conform to what we might expect from examining Figure 9. Focussing first on the $M = 500$ results, we see the effective noise variance from treating z purely as a function of y is very high, it is less when treating z purely as a function of x, and it is minimised when we treat z as a function of both x and y.

Thus the Gamma test results give a form of sensitivity analysis, in that they suggest that ignoring x has worse consequences when predicting z than ignoring y.

Although this is a case of 'effective noise' rather than 'real noise' we can make an interesting observation regarding inhomogeneous noise using this example. If we regard the function as a function of x only then the effective noise distribution illustrated in Figure 9 is not homogeneous across the x-input space. If we average the local noise variances across the x-input space we obtain the value 14.0126 and indeed the Gamma statistics corresponding to the embedding 10 in Table 1 do approximate this value.

The effect of increasing M to 5000 is to reduce the Gamma statistic in the first row of Table 1, where both variables are included, but not to substantially reduce the Gamma statistic in the other two rows. Note also that the gradient estimate A for these rows fluctuates wildly as M becomes large [13], whilst the Gamma estimates stabilize.

This idea for variable selection continues to be effective in the presence of 'real noise', but we must expect to use more data as the noise variance increases.

3.3 The gamma histogram and frequency analysis of bins

In general if there are m input variables then there are $2^m - 1$ possible selections of input variables which one might consider (selecting none of the inputs is

[13] One can ascribe this to the discontinuities of the gradient mentioned earlier.

pointless). Given currently available computing power, running a Gamma test on every possible combination of inputs becomes infeasible at around $m = 20$. Moreover, ideally one also prefers M to be quite large to justify the effort for high dimensional data. Hence there is a case for parallelizing this aspect of a Gamma analysis. We call this process of exhaustive examination a **full embedding search**.

With $m = 20$ there are around 1.048×10^6 Gamma tests to be run. Each result is specified by the embedding used to generate it and this can be indicated by a binary string of length m, in which the presence, or absence, of a variable is indicated by 1, or 0, respectively. We call such a binary string a **mask**.

- If a full embedding search is practical then the simplest rule of thumb is to select the mask that gives the Gamma statistic closest to zero. If there are several mask's with approximately equivalent low Gammas then choose the mask that has the smallest A value, on the grounds that this choice is likely to give the simplest model.

A useful way to represent the results of a full embedding search is in the form of a **Gamma histogram**. Here we divide the range of the resulting Gamma statistics on the horizontal axis into bins and plot the frequency of the Gammas per-bin vertically. The form of this histogram is often very revealing.

Example: The sunspot data [Tsui, 1999]. 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 et al., 1990]. The data consists of 280 points representing sunspot activity over the period $1700 - 1979$ and was used in [Weigend et al., 1991]. The range of the data has been scaled to [0, 1] and we found the variance to be 0.0410558. Figure 10 shows the variation of sunspot activity over the full range of the data. It is known that the primary sunspot cycle is approximately

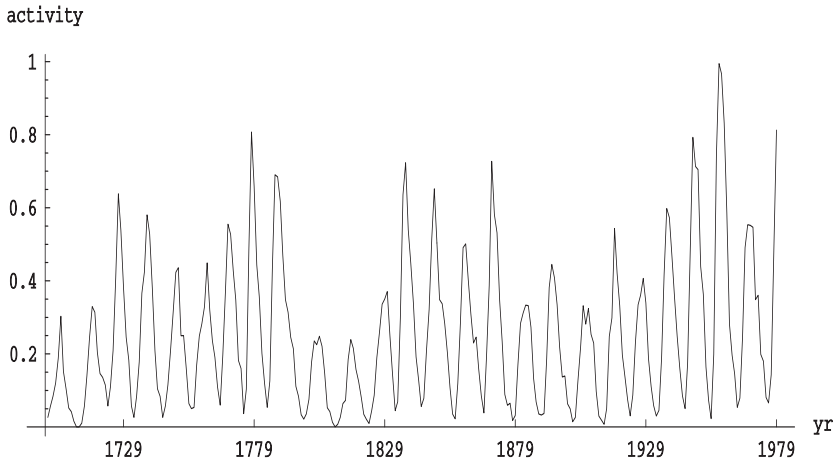

Figure 10: Time series of sunspot activity over the period $1700 - 1979$.

periodic over 11 years. It is probably not a co-incidence that this is roughly the period of Jupiter, the largest planet in the solar system. 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 then 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.

We examined all possible 15-dimensional embeddings. 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 $x(t-1)$, $x(t-2)$, $x(t-3)$, $x(t-5)$, $x(t-10)$ and $x(t-13)$, an irregular embedding of dimension six.

If we run the Gamma test on the six inputs/one output I/O data file constructed using this mask we get $\Gamma = 0.0015$ and $V_{ratio} = 0.0368$ (SE $= 0.000936$). Note the M-test of Figure 12, which indicates that we really do not have enough data (the graph has not stabilized). Therefore if we construct a model and test

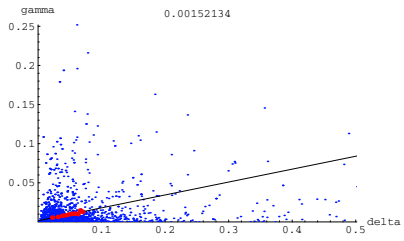

Figure 11: Scatter plot, regression line and plot points for the best embedding found for the sunspot data.

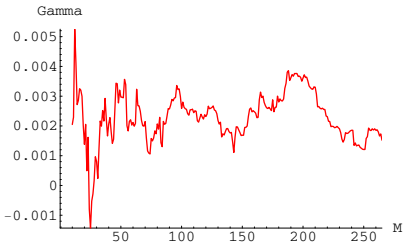

Figure 12: M-test for the best embedding found for the sunspot data.

on unseen data we might expect to get a higher *MSError* than the estimated Gamma value. We shall construct such a model in section 4.4.1.

3.3.1 Bin-frequency analysis

It is interesting to note the bimodal distribution of the full embedding 100 bin histogram of Figure 13. The bimodal distribution is partly explained by the observation that only 2.38% of the embeddings with a Gamma statistic > 0.008 include $x(t-1)$, as compared with 99.8% of those having a Gamma statistic < 0.008. Put plainly this says that the most important predictive factor for the sunspot activity this year is the value for last year.

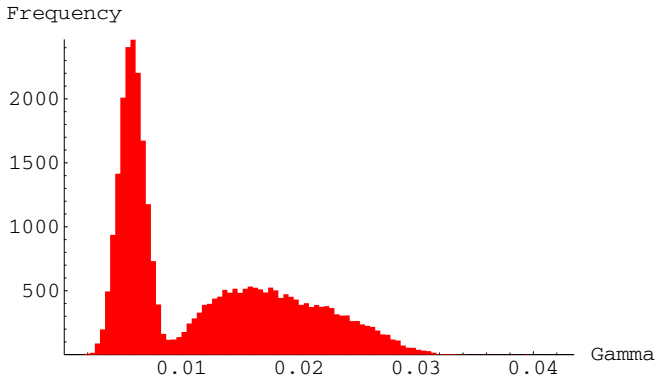

Figure 13: Histogram of gamma values from a full embedding search of the sunspot data.

In general a Gamma histogram such as this, having distinct peak structure, reveals different predictive value for different subsets of the inputs.

If we examine which variables appear in the best few embeddings for the sunspot data these indicate that using the values for 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. We found this a rather convincing demonstration of the ability of the software to extract salient features, since *a priori* it has no way of knowing about sunspot cycles.

More generally if we analyse the frequency with which different input variables appear in embeddings corresponding to Gamma values in the peak regions then the relevant subsets of variables become apparent.

We particularly look for variables appearing with *high* frequency in embeddings corresponding to low-Gamma regions, and with *low* frequency in embeddings corresponding to high-Gamma regions. Armed with this information we can then examine which *combinations* of variables are important.

- Practical experience suggests that a *low* frequency of a variable in a high-Gamma region is often a more telling indicator of its importance than a *high* frequency in a low-Gamma region.

This aspect of a Gamma analysis is described and illustrated in [Evans and Jones, 2003a].

If a full embedding search is impractical then we can run a fixed number of Gamma tests using randomly generated binary masks. The above techniques for extracting the significance of input variables by studying the Gamma histogram bin-frequencies and relative bin-frequencies can then still be applied.

In the search for good irregular embeddings in a high dimensional input space, an alternative to the frequency analysis of a Gamma histogram is to use a genetic algorithm in which a mask's fitness is inversely proportional to its Gamma value. For multiple time series a *hierarchical* GA over the mask space has proved particularly effective (this will be reported elsewhere).

Other heuristics include:

- 'Leave one out': in which a Gamma statistic is computed for the full mask and then each mask bit is flipped in turn: if leaving a variable out results in a higher Gamma value then the variable was relevant and the bit is reset, otherwise the variable is omitted from the final mask.
- 'Hill climbing': in which we work along the mask (for time series we start with with the most recent inputs in time) flipping each bit. If the resulting masks yields an improved Gamma statistic the bit setting is retained. After working along the whole mask in this way we return to the start and repeat the process. This procedure is repeated until no further bits are altered.

3.4 Analysis of time series

In the analysis of a time series such as the sunspot data, in which we hypothesize that the variable of interest is just one of a number of variables of a complex dynamic system determined by a system of differential equations, following the work of [Takens, 1981] we seek to predict the next value (the *output*) based on a number m of previous values (the *input* vector). In this context the input vector is called an *embedding vector* - which is consistent with our earlier use. If we consider all potential inputs up to m, the embedding is called a *regular* embedding (of dimension m), otherwise it is an *irregular* embedding.

Thus for a sampled time series (x_t) Taken's theorem and its subsequent extensions, assures us that, under a broad range of circumstances, there does exist a *smooth* model f such that

$$x_t = f(x_{t-1}, x_{t-2}, \ldots x_{t-m}) \tag{9}$$

which, provided m is sufficiently large to unfold the dynamics, can be used as the basis for a recursive one-step prediction. Thus, in general, the Gamma test can be applied in such situations.[14]

This predictive technique is very intuitive and has an illustrious history in forecasting, Lorenz called it the 'method of analogies'. The idea is that we make a prediction based on historical evidence by asking 'what happened in the past when we saw a *similar* sequence of events?'.

We can implement this idea efficiently if we simply recognise that finding sequences of historically similar events exactly corresponds to finding near neighbours in the embedding space i.e. to the construction of a *kd*-tree, which we have already done in computing the Gamma test result.

[14] Although if the dynamics is *chaotic* we have to take the extension of the existing theory on trust at present.

A useful algorithm for estimating an appropriate embedding dimension is the *False Nearest Neighbour* technique (FNN) [Kennel et al., 1992]. A Gamma test analysis provides an alternative approach: we can estimate the embedding dimension by progressively increasing the length of the mask for a regular embedding, working backwards from the most recent samples, and examining for which m the Γ value is closest to zero.[15] We should not take this too literally because it may be, as with the sunspot data, that by taking a larger value of m and choosing a suitable irregular embedding we might do better. Both approaches scale like $\mathrm{O}(M \log M)$ and we have found the two approaches to be remarkably consistent. On balance FNN seems to be slightly faster in practice and somewhat more tolerant to high noise levels.

Remark. We are talking here of the global dynamical embedding dimension not a geometrical dimension, such as a fractal or Hausdorff dimension (which may vary locally), although the two are often related. When iterating a chaotic dynamical system we can expect in time to cover the whole attractor and so it is the global dynamical embedding dimension which is important in constructing a model.

Once the embedding dimension has been determined we can use the techniques described in section 3.3 to determine a suitable irregular embedding.

More generally, when dealing with real time series data, in which any underlying dynamics may be somewhat conjectural, we may instead look for other time series data[16] which has predictive value for the target time series. The same techniques can also be used in such a situation - we can use domain knowledge to determine which time series may contain relevant predictive information, and then test such hypotheses as illustrated below.

4 Model construction

We shall give two examples of how these Gamma analysis techniques can be used with some standard tools in non-linear model building. Here we shall construct a local linear regression model. Later in section 5 we construct a neural network model for predicting water level at a river site.

4.1 Local linear regression

One of the simplest non-parametric non-linear modelling techniques is *local linear regression* (LLR). Here using the input training data we first build a *kd*-tree, a process with time-complexity $\mathrm{O}(M \log M)$. To construct a prediction for a previously unseen input $\boldsymbol{x}$ we use the *kd*-tree to locate the set $\{\boldsymbol{x}_i | i \in S[p]\}$ of the pth nearest neighbours of $\boldsymbol{x}$, which can be accomplished in $\mathrm{O}(\log M)$ time [17]. We next construct a linear regression model H using the pairs $\{(\boldsymbol{x}_i, y_i) | i \in S[p]\}$.

[15] We call this an **Increasing Embedding** search.

[16] Termed a *leading indicator* in economic analysis.

[17] Since we need to construct near neighbour lists for the Gamma test analysis the LLR model comes virtually free.

Finally the output for the query vector is $y = H(\boldsymbol{x})$. A further elaboration, particularly for time series, is a principal components threshold filter on the eigenvectors of the local linear model (usually involving a user settable parameter). We are trying to predict along the tangent plane of the local flow, and eigenvectors corresponding to relatively small eigenvalues probably represent noise and lie outside the tangent plane. LLR models are quite fast to construct and fast to execute a query.

LLR models can also be easily updated as new training data becomes available, which is not the case with neural networks (where a prolonged extra period of training, or starting training all over again, may be required to modify the model on the basis of new data). It may seem odd that, although the topic under discussion is the construction of smooth models, the global function produced by patching together many LLR predictions in general is not even continuous! However, as the number of well distributed data points increases, the global function produced by LLR will converge in probability (usually quite rapidly) to the unknown function generating the data, provided this is itself a smooth function.

LLR can produce very accurate predictions in regions of high data density in input space, but it is liable to produce unreliable results for non-linear functions in regions of low data density. In other words LLR does not generalise well but is a very good interpolative tool if we have large amounts of data.

The optimal number of near neighbours p to use in LLR is principally dependent on the noise level (high noise levels will require a larger p), the sampling density (low sampling density in the vicinity of the query requires smaller p), and the local curvature of the unknown non-linear function f in the vicinity of the query point.

Example. The Sunspot data. In section 3.3.3 we found the irregular embedding mask 001001000010111 for the sunspot data with $\Gamma = 0.0015$. The scatter plot and M-test for this embedding were given in Figures 11 and 12 respectively.

We now predict the last 59 year's data, using one-step-ahead predictive local linear regression with $p = 60$ near neighbours and a local flow threshold of 0.0001. On the basis of all the previous years we obtain Figure 14 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 statistic. Even so, the results are reasonably satisfactory, the model predicts the exceptional peak of 1956. A peak of this magnitude did not previously occur in the data.

4.2 Neural networks

We use the BFGS neural network training algorithm [Fletcher, 1987], which provides progressive adjustment of the neural network weights by gradient descent. This is a quasi-Newton method performed iteratively using successively improved approximations to the inverse Hessian, instead of the true inverse.

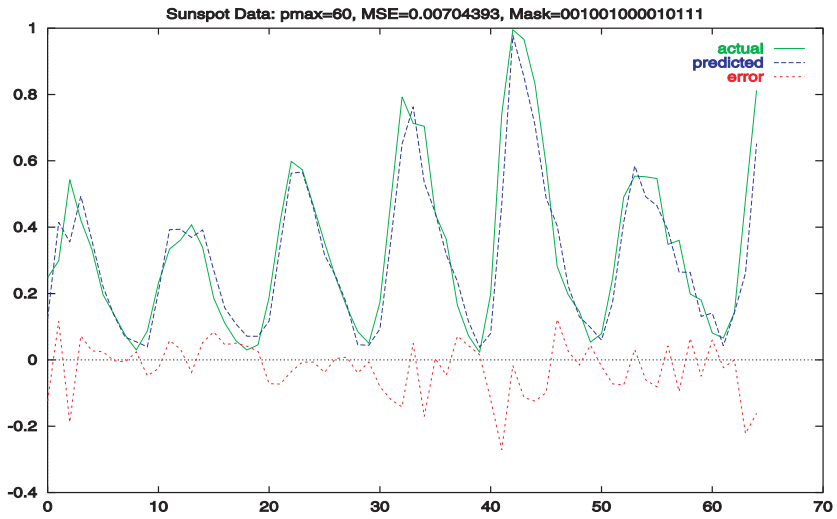

Figure 14: Test of the LLR sunspot model predicting one step ahead on 59 unseen data points. The data is shown in green, the one step prediction in blue, and the error in red.

The improved approximations are obtained from information generated during the gradient descent process. We use this algorithm in section 5.3.

We know that feedforward networks with as few as one hidden layer can act as universal approximators for continuous functions over a compact set [Cybenko, 1989], [Hornik et al., 1989]. In practice it is usually most effective to use two hidden layers. Plainly, the precise architecture required for a particular model will depend on the complexity of the surface to be modelled.[18] However, this dependence is quite subtle. For example, because a neural network approximates a surface by the superposition of 'humps', it is actually quite difficult for a network to accurately model an essentially *linear* surface, and a disproportionate number of nodes in the hidden layers might be required in such a case.

Assuming there is adequate data, once one has a suitable architecture, training down to the Gamma statistic should be quite straightforward. If training fails to reach the Gamma statistic, or takes excessive time, this is invariably because the network architecture is unsuited to the function being modelled, and the number of units in the hidden layers should be increased accordingly.

Detailed examples of such modelling exercises for chaotic systems can be found in [Jones et al., 2002], [Tsui et al., 2002], and [Evans and Jones, 2003a].

[18] So we might expect some loose relationship with the slope parameter A returned by the Gamma test.

5 A case study: Thames River Valley

We briefly describe how these techniques were used to good effect in a case study directed towards short term prediction of river levels at selected sites on rivers in the Thames valley region. This work, drawn from [Durrant, 2001], will be reported in more detail elsewhere [Durrant and Jones, 2003], and was a feasibility study for the *MAPFLOWS* (Modular Automated Prediction and Flood Warning System) project.

Once precipitation has occurred the process of runoff, although highly complex in any particular catchment area, is completely determined by physical and hydraulic processes, geomorpholocal processes, boundary and initial conditions, and any system parameter such gating openings.

Thus in many respects downstream water flow/level prediction is an important application ideally suited to the algorithms incorporated into $winGamma^{TM}$. The reason being that once precipitation has occurred (as measured by suitable sensors) the entire water transport process to the sites for which prediction is required is essentially determined by a smooth (albeit complex) process with lags.

5.1 The Thames river valley region

The river system data used in this analysis was measured in the Thames river basin above Windsor, see Figure 15. The data was provided by the UK Envi-

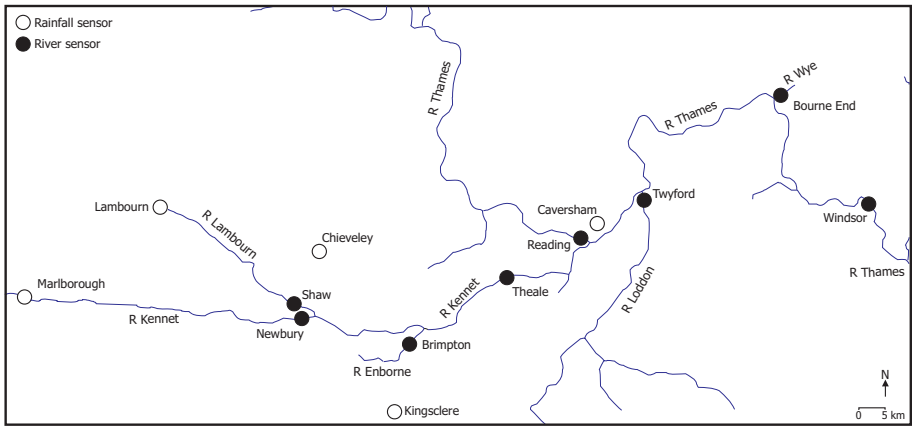

Figure 15: The Thames study area.

ronment Agency. It consists of *flow rate* (cubic metres per second) and *level* readings (metres) for the rivers at Newbury, Shaw, Brimpton, Theale, Reading, Twyford, Bourne End and Windsor. The rainfall measured in mm/hour at five sites in the region was also collected. All of the measurements were collected hourly over one year. The river and rainfall sensor positions are marked on the map. The general direction of flow is from west to east.

The data was first scanned for sensor malfunctions and a simple thresholding algorithm, designed to operate in real-time when future sensor values would not be known, was used to correct obviously faulty readings by replacing them with their last known reliable value. An illustrative graph of of the cleaned level data is given in Figure 16. Similar graphs were obtained for the cleaned flow data.

Since different data types such as flow, level and rainfall were in different units and over significantly different range scales all data was normalized prior to analysis by mapping the mean to zero and the standard deviation to $\frac{1}{2}$.

5.2 Model identification

Given this data one could build predictive models for both level and flow, but we report here on a level model. Examination of the regional map in Figure 15 shows that two level models can be sensibly constructed from the data measured at the marked sensor sites:

- Theale: (Rainfall, Newbury, Shaw, Brimpton) → Theale
- Windsor: (Rainfall, Theale, Reading, Twyford, Bourne End) → Windsor

Here 'Rainfall' indicates some combination of lagged and aggregated rainfall measurements, and the site name indicates level and flow measurements from the relevant site.

These models are determined by the location of the level/flow sensor sites. The first model covers the rivers flowing into Theale, primarily the River Kennet. The second models covers the rivers flowing into Windsor, primarily the River Thames, but also the flow from the River Kennet through Theale. This second model allows us to use either the real data measured at Theale, or the predicted river levels from the first model. This enabled us to investigate the modular design of a predictive system.

Having normalized these data series our first task is to determine lags where possible. The most obvious way to determine the correct transfer times between successive measurement points is by direct on-site measurement, preferably under a variety of flow rate conditions. This would be the recommended approach in a real system. It is relatively straightforward to accomplish and, once performed, leaves no room for doubt; although one should be aware that transfer times will decrease under flood conditions.[19] Additionally such physical measurements act to validate algorithmic approaches, such as described below, to determining lags.

We found the following approach, which uses a combination of two algorithms for lag determination, worked well in practice.

The first algorithm we called *Delta Correlation*. Here the delta correlation $\Delta_c(d)$ of the target time series $y(t)$ with an input time series $x(t)$ $(1 \leq t \leq M)$

[19] In practice this might mean one should use multiple inputs bracketing the plausible transfer times.

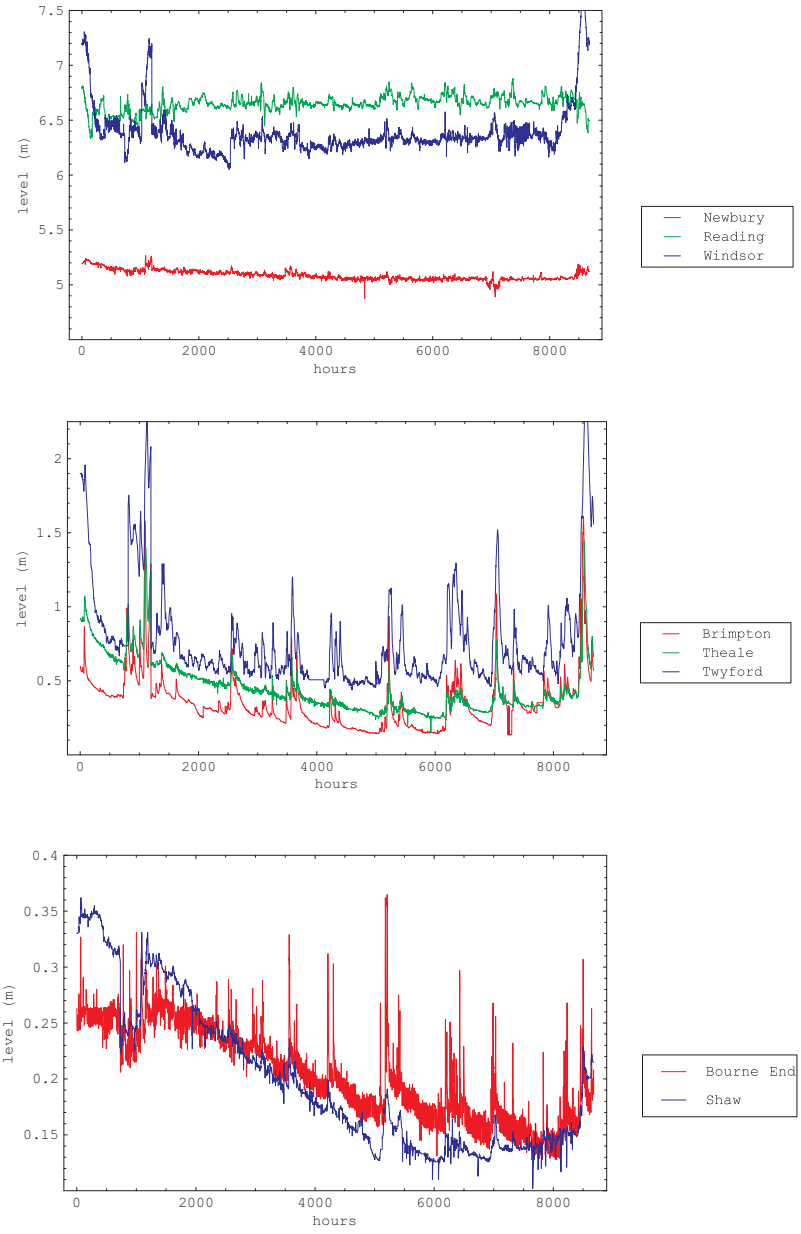

Figure 16: Cleaned river level data at the various sites.

at lag d is defined by

$$\Delta_c(d) = \frac{1}{AB} \sum_{i=1}^{M-d-1} (x(i+1) - x(i)) (y(i+1+d) - y(i+d)) \qquad (10)$$

where

$$A^2 = \sum_{i=1}^{M-d-1} (x(i+1) - x(i))^2$$

$$B^2 = \sum_{i=1}^{M-d-1} (y(i+1+d) - y(i+d))^2$$

The idea here is to correlate *changes* in the input time series with later *changes* in the output time series at some lagged time d. The time lag with the highest positive correlation[20] should indicate the flow time between sensor points.

An important aspect of Delta Correlation is that it is very fast - so one can obtain an initial overview of what lags are likely to be important very quickly. We show typical graphs for the delta correlation (vertical axis) plotted against the lags in hours in Figures 17 and 18.

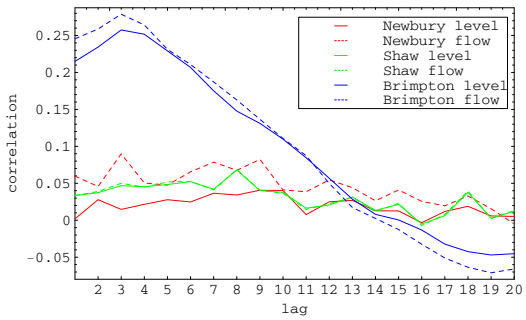

Figure 17: The Delta flow and level correlations from Newbury, Shaw and Brimpton measured against the river level at Theale.

Usually identification of lags for flow and level time series was not too difficult: we simply picked the lag time with the largest delta correlation, provided this was consistent with out understanding of the relative distances involved. Rainfall and aggregated rainfall lags were often harder to decide.

After determining the lags by selecting the maximum correlation we arrive at the Delta correlation results shown in Table 2. It interesting to compare the results of Delta correlation in Table 2 with those arrived at by performing successive Gamma tests in which a single input time series with different lags is used - we call this a *Time-lag* Gamma test. The Time-lag Gamma test compares the target time series $y(t)$ with an input time series $x(t)$ $(1 \leq t \leq M)$

[20] In the context of river flows, the correlations will be positive since the expectation is that as an upstream river rises (or falls) then downstream the river will correspondingly rise (or fall) at a later point in time. In general, for other types of problem, strong negative correlations may be as significant as strong positive ones.

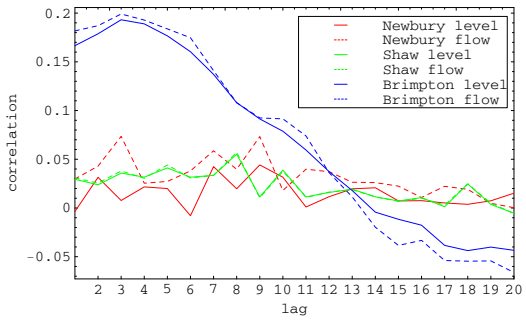

Figure 18: The Delta flow and level correlations from Newbury, Shaw and Brimpton measured against the river flow at Theale.

by computing Gamma statistics for data sets $(x(t-d), y(t))$ for $d = 1, 2, 3, \ldots$ and then choosing the lag d which produces a Γ closest to zero. This approach worked very well on simulated river data and we were interested to see how it fared on real data. In the context of Gamma analysis, it is a rather crude technique, as it takes none of the other inputs into account. As one can see by comparing the columns headed 'Delta' and 'Gamma' in Table 2 in practice on real data the 'Time-lag' Gamma test did not compare well with the simpler and faster Delta correlation.

Table 2: Estimated lags for the Theale area measurements. The lags chosen for the analysis were derived from the Delta correlation analysis. The lag for Kingsclere rainfall was manually selected as 8 hours.

	Delta		Gamma		used
measurement	level	flow	level	flow	lag
Newbury level	9	9	6	7	9
Newbury flow	3	3	1	1	9
Shaw level	8	8	1	3	8
Shaw flow	8	8	1	3	8
Brimpton level	3	3	6	6	3
Brimpton flow	3	3	1	1	3
Regional rainfall 1-hour average	13	13	1	16	13
Regional rainfall 1-day average	4	4	16	16	4
Regional rainfall 7-day average	8	8	1	1	8
Regional rainfall 28-day average	7	4	1	2	7
Marlborough rainfall	13	9	-	-	13
Lambourn rainfall	13	9	-	-	9
Chieveley rainfall	13	13	-	-	13
Kingsclere rainfall	20	20	-	-	8

The Delta correlation analysis unambiguously identifies the lags from Shaw and Brimpton to Theale to be 8 hours and 3 hours respectively. The lag between Newbury and Theale is less clear cut. The analysis produces a 3 hour lag using the flow data and a 9 hour lag using the level data. The distance between Newbury and Shaw would suggest that the lag to Theale should indeed be around 9 hours. A closer examination of the data used to produce Figure 17 shows that the 3 hour lag had a correlation of 0.0735 and the 9 hour lag a correlation of 0.0732. We can concluded that the likely lag is indeed 9 hours given all of the available evidence.

For the regional rainfall aggregated over 28 days we obtain a Theale level correlation of 0.111 corresponding to a lag of 7 hours, whereas for the flow we obtain a correlation of 0.102 corresponding to a lag of 4 hours. In this case the meaning of a lag against a 28 day aggregated rainfall is less clear cut, but examining the graphs we decide that a 7 hour lag may be more appropriate here. The Delta correlation between individual rainfall sensor sites and Theale were also analysed as they could introduce additional local information that the aggregated regional rainfall cannot describe.

The lags calculated in Table 2 were used to construct a data set for the Theale area model. The choice of inputs was validated using the full-embedding routine. This analysis determined that the rainfall at Lambourn and the flow at Newbury were irrelevant ($|\Gamma| = 0.00077$ with Lambourn rainfall and Newbury flow and $|\Gamma| = 2.1 \times 10^{-6}$ excluding Lambourn rainfall and Newbury flow). The results of the analysis are shown in Table 3.

Table 3: The Gamma test analysis results on the Theale area data set. The two results compare the effect of including or excluding the Lambourn rainfall and the Newbury flow (indicated by a `1` or `0` in the mask respectively).

	Including Lambourn rainfall and Newbury flow	Excluding Lambourn rainfall and Newbury flow
$\|\Gamma\|$	0.00077	2.0638×10^{-6}
Gradient A	0.01865	0.022833
Standard error	0.00066	0.00037164
|V-ratio|	0.00306	8.255×10^{-6}
Near neighbours	10	10
M	8076	8076
Zero nearest neighbours	175	414
Lower 95% confidence	-0.00123	-0.0011044
Upper95% confidence	0.001921	0.0017852
Mask	`11111111111111`	`11011111101111`

5.3 Model construction and testing

The consequence of the analysis is to use the inputs and lags in the Theale model that correspond to those shown in Table 2 without the rainfall measurements at Lambourn and the flow measurements at Newbury.

Now that the 'optimal' inputs have been selected using the combination of Delta correlation and the Gamma test, we can perform the usual analysis.

First, the Gamma scatter plot of Figure 19 for this data set shows a moderate level of noise. The *quantity* of data was analysed, using the M-test to determine

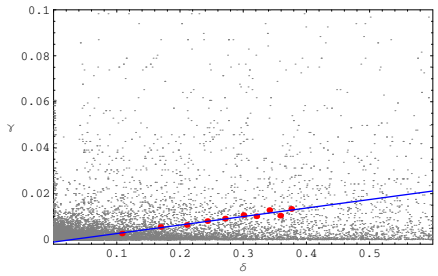

Figure 19: Theale data scatter plot ($M = 8076$, $p = 10$) and regression line.

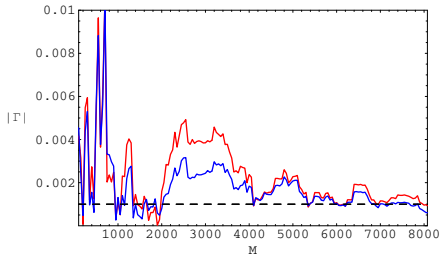

Figure 20: Absolute Gamma statistic $|\Gamma|$ for the randomized data ($M = 8076$, $p = 10$). Red - flow, Blue - level.

whether there was sufficient data to provide an asymptotic Gamma estimate and subsequently a reliable model. The results of this analysis are shown in Figure 20. To average the seasonal effects implicit in the data, an M-test was performed on order randomised data and the results plotted.[21] As the M-test proceeded, the Gamma test algorithm was exposed to points randomly sampled throughout the year. This produced an asymptotic convergence of the Gamma statistic, $\Gamma = 0.0007$ (there was very little difference between level and flow in this respect), and indicated that there was sufficient data at around $M = 6000$ data points.

The data order was randomized for model training. The target $MSError$ for the models was 0.000841. This was calculated for the training set created from 6500 randomly selected data points and using the mask 11011111101111 from Table 3. Since the minimum lag used is the 3 hour lag from Brimpton to Theale, these models give a three hour ahead prediction.

Two types of model were constructed and tested. The first was a LLR model (with 30 near neighbours) and the second was a $12 \rightarrow 10 \rightarrow 10 \rightarrow 1$ BFGS neural network. The results are summarized in Table 4

We give graphs for the neural network model in Figure 21 which shows the response of the model over the entire data set in which 20% of the points are unseen, and Figure 22 which shows a close up view of a model test on a subset of the unseen randomized data.

[21] By this we mean, of course, that the entire input/output data set we have constructed was row randomised.

Table 4: A comparison of the *MSError* values of the two Theale level models showing the scaled and unscaled data performance.

	Local-linear regression		Neural network	
	Scaled	Unscaled	Scaled	Unscaled
Training data	0.000776	6.679×10^{-5}	0.000863	8.002×10^{-5}
Test data	0.00202	0.000187	0.00124	0.000115

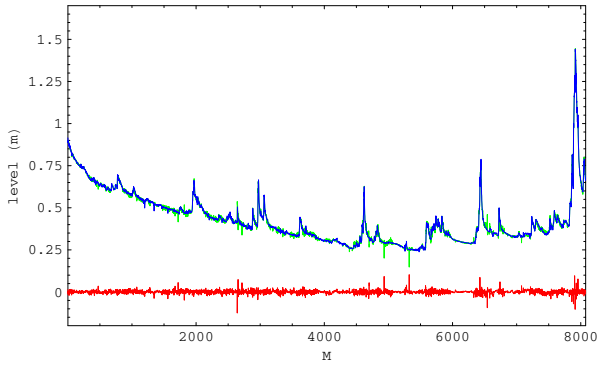

Figure 21: The BFGS model response in chronological order (20% of points are unseen). Blue - model prediction, Green - actual data, Red - error.

Overall the results are rather encouraging. In Table 4 the unscaled MSE of 0.000115 on the unseen test data translates to an error standard deviation for the level prediction [22] of 0.0107m, for the 3 hour look-ahead neural network model, i.e. around 1cm.

Although we have tackled the task in terms of *level* prediction it is interesting to examine the effectiveness of this model in terms of the accuracy of the three hour ahead *changes* in level it predicts. The average absolute change in level over three hours for the whole data set was approximately 7.35mm, i.e. around the *noise level*. So the accuracy measured relative to *level change* over three hours is inevitably[23] not particularly impressive. However, the model is performing well in the absolute sense of progressive level prediction, and one can illustrate this by examining its propensity to anticipate turning points, or more generally the correct *direction* of change. On the unseen test set the model predicts the correct direction of level change 75.25% of the time, but, as one might expect, most of the errors occur when the real three hour level change is small. If we examine only those cases where the absolute change of level over three hours is greater than half the average, i.e. greater than 3.67mm then on the unseen

[22] Calculated by taking the square root of the MSE.
[23] For the reasons explained in section 1.3.

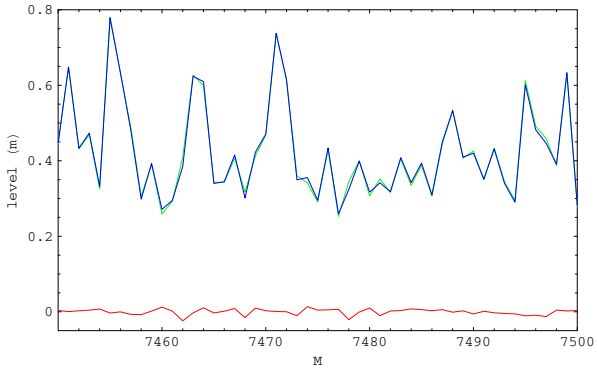

Figure 22: A closer inspection of the BFGS model performance on the (randomized) unseen data shows an acceptable error level. Blue - model prediction, Green - actual data, Red - error.

test set the model predicts the correct direction of level change 81.36% of the time. For absolute changes of three hour level one standard deviation above the average absolute three hour level change, this rises to around 92%.

In a corresponding exercise for the sensor site at Windsor, deviations from the model prediction show periodic daily fluctuations. We attribute these fluctuations to periodic extractions and replacements, such as are created by factory or agricultural use of the river water. If this is indeed the case, then our observations could be used to develop an automatic monitoring program for detecting the unlicensed use of river water.

6 Conclusions

We have illustrated how the basic Gamma test algorithm can be enhanced with various supporting procedures to answer a number of practically quite critical questions for those engaged in the construction of non-parametric smooth models. These tools include: the **scatter plot** and associated regression line and plot points, the M**-test**, the **full embedding** search, the **Gamma histogram and bin frequency analysis**, and **increasing embedding** as an alternative method of determining the embedding dimension for a time series.

The application of the Gamma test to the selection of relevant variables in the construction of non-linear models is a useful addition to the standard library of such techniques. In application areas, such as financial modelling, where the underlying processes are conjectural, there are epistemological caveats, but such analysis can still be useful in refuting or confirming conventional wisdom regarding the relative importance of useful predictive variables.

Many users of $winGamma^{TM}$ are explicitly interested in time series prediction of economic data. Here the most promising approach seems to be to bring to bear user domain knowledge to determine which other available time series

data can act as leading indicators for the target time series. We propose in the first instance to provide a set of time-series editing tools that facilitate the alignment in time of attribute values from different time series and the selection of subsets of lagged data.

In other types of situation it may well prove useful to be able to estimate not just the second moment of the noise but higher moments as well. For example, in macro-economic modelling the skewness and kurtosis are sometimes of especial interest. In another paper [Evans and Jones, 2003b] we show how the mathematical techniques used in [Evans and Jones, 2002] may be extended to estimate as many higher moments of the noise as the amount of data might justify. It emerges from that work that the Gamma test is just one of a whole class of similar such algorithms. In particular, apart from the Gamma test itself, there are other similar but different ways to estimate the second and higher moments.

There are a range of more certain potential applications of these ideas in many fields of science and engineering, although one might single out signal processing for immediate attention.

6.1 Guidelines for applicability

We re-iterate some cautionary notes regarding the range of applicability of the Gamma test. As the existing theory stands it is applicable in circumstances where:

- Input and output measurements constitute *real numbers*, not categorical or discrete values.
- The number of data points is large in relation to:
  - The dimensionality of the input vector.
  - The complexity of the underlying smooth functional model.
  - The noise level and the nature of the noise distribution.
- Sampling of inputs is well distributed (i.e. smooth positive sampling density) across the input space.
- Measurements are to a precision commensurate with the number of data points being processed (because the algorithm computes *differences* of point coordinates, with very large data sets measurements should be high precision, as should the corresponding calculations).

We can have confidence in the results if the above conditions are satisfied and the M-test graph has stabilised. In general the scatter plot and associated regression line and plot points are very useful indicative tools which can often highlight problems with the data.

Although there is a large body of experimental evidence supporting the utility of the Gamma test in the analysis of chaotic dynamical systems, the

existing theory does not cover this case.[24] It seems like that the theory could be extended, but this would require answering some quite tricky questions in Hausdorff measures.

In any event, such questions may be of mainly academic interest. For often in practical applications we have no way to determine if theoretical pre-conditions are (or are not) satisfied. Thus the most direct approach in determining the utility of the Gamma Test may be to simply try it! In this case it behoves the analyst to treat the conclusions, and test the resulting models, with more than a normal level of scepticism.

6.2 Future application developments

The Gamma test would appear to be an interesting new tool for model discovery. $winGamma^{TM}$ was constructed as a non-linear analyst's workbench, and as with any such tool there is a learning curve which must be ascended to acquire the necessary skills to apply the tool effectively. However, as we have gained more experience in the use of $winGamma^{TM}$, and began to develop an analysis protocol, it has become apparent that the analysis process could be *automated* with little loss of effectiveness.

6.2.1 Datamining

Because the Gamma test runs extremely quickly one can therefore envisage a more sophisticated program (*GammaMiner*) which automatically scans large databases looking for relationships between numerical fields which can be used for smooth modelling and prediction. The user could define which attributes were of particular interest (the targets or outputs required to be predicted) and which other attributes the targets might reasonably depend on (these would form the set of potential inputs to the model). Designing such a program is not without pitfalls. For example, attribute values may not be time-stamped and one could easily find the program 'predicting' values that predate the attribute values used as inputs. Thus there are some problems regarding database semantics that need to be addressed. Because not all data falls into the category of numerical fields which might be modelled by a smooth function and because other types of tools (e.g. decision trees) may be more appropriate for constructing predictive models on discrete inputs or categorical outputs, one could also envisage engineering a sub-set of *GammaMiner* as a re-usable component designed to be integrated into existing or future data mining tools.

Nevertheless, it is possible to imagine such a program running continually in the background and notifying its owner only when it found something interesting. E.g. "By the way I have a predictive model for X for one month ahead which gives an expected error of 0.5% are you interested?" While such program behaviour is not intelligent in any real sense, there is no doubt that such a tool would be useful.

[24] For one reason, the sampling of input space is not well distributed in the sense intended here.

One very interesting and potentially extremely beneficial application of these ideas, and other techniques for dealing with discrete input and output variables, is to large medical databases. However, a not inconsiderable problem in this respect is that, certainly under present UK law, even anonymised data is protected to an extent that hampers access for such blanket research approaches. Moreover, much useful data is propriety to drugs companies and often inaccessible to the academic researcher.

6.2.2 General purpose non-linear modelling tools

As we have illustrated, we are developing the application of these ideas to a Modular Automated Prediction and Flood Warning System (*MAPFLOWS*) for the non-linear modelling of river systems.

More generally it would be of great utility to create a macro-language to facilitate the generation of code for data driven, non-parametric, non-linear modelling of complex dynamic systems. Thus the automated Gamma test would become a single component in such a system. Building on the earlier ideas of *Simula* we could generate a system rather similar to *STELLA*, in that flow process charts could be used via a graphical user interface to specify inputs and outputs for specific nodes and link the nodes together into a model of a complex dynamic system. The difference would be that nodes would be non-linear input-output models automatically generated from the data. Each node would have an associated Gamma statistic, which would quantify the expected error relative to the actual data.[25] Using this information the propagation of errors through the system could be studied and quantified. In this way we could identify critical nodes which limit overall accuracy, and assign measures of confidence to the outcomes of simulations.

Acknowledgements. The work developing the analysis techniques described in this paper was carried out over some seven years, and would not have been possible without the enthusiastic efforts of many Ph.D. students, several of whom generously allowed me to use their examples in this paper. I regard myself as fortunate indeed in having had the opportunity to work with them.

For copies of their theses, pre-prints of papers not yet published, and other relevant references the reader might wish to consult the web-site:

http://www.cs.cf.ac.uk/user/Antonia.J.Jones/GammaArchive

[25] We again emphasize that the model might actually be near perfect and the data incomplete or noisy - so that the model predictions could in principle be more accurate than the measured outcome!

References

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

[Bentley, 1975] Bentley, J. (1975). Multidimensional binary search trees used for associative search. *Comm. ACM*, 18:309–517.

[Bishop, 1996] Bishop, C. M. (1996). *Neural Networks for Pattern Recognition*. Oxford University Press. ISBN 0-19-853864-2.

[Chuzhanova et al., 1998] Chuzhanova, N. A., Jones, A. J., and Margetts, S. (1998). Feature selection for genetic sequence classification. *Bioinformatics*, 14(2):139–143.

[Corcoran et al., 2003] Corcoran, J., Wilson, I., and Ware, J. (2003). Predicting the geo-temporal variation of crime and disorder. *To appear in International Journal of Forecasting, Special Issue on Crime Forecasting*.

[Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signal and Systems*, 2:303–314.

[de Oliveira, 1999] de Oliveira, A. G. (1999). *Synchronisation of chaos and applications to secure communications*. PhD thesis, Department of Computing, Imperial College of Science, Technology and Medicine, University of London.

[Durrant, 2001] Durrant, P. J. (2001). winGamma^{TM}*: A non-linear data analysis and modelling tool with applications to Flood Prediction*. PhD thesis, Department of Computer Science, Cardiff University, Wales, UK.

[Durrant and Jones, 2003] Durrant, P. J. and Jones, A. J. (2003). Non-linear models of river levels using the gamma test. *Paper submitted to Journal of Hydrology*.

[Evans, 2002] Evans, D. (2002). *Data Derived Estimates of Noise using Near Neighbour Asymptotics*. PhD thesis, Department of Computer Science, Cardiff University, Wales, UK.

[Evans and Jones, 2002] Evans, D. and Jones, A. J. (2002). A proof of the gamma test. *Proc. Roy. Soc. Series A*, 458(2027):2759–2799. ISSN 1364-5021.

[Evans and Jones, 2003a] Evans, D. and Jones, A. J. (2003a). Gamma test protocols. *In preparation*.

[Evans and Jones, 2003b] Evans, D. and Jones, A. J. (2003b). On the reconstruction of noise. *In preparation*.

[Evans et al., 2002] Evans, D., Jones, A. J., and Schmidt, W. M. (2002). Asymptotic moments of near neighbour distance distributions. *Proc. Royal Soc. Series A*, 458(2028):2839–2829. ISSN 1364-5021.

[Fletcher, 1987] Fletcher, R. (1987). *Practical Methods of Optimization*. John Wiley & Sons, 2nd edition.

[Friedman et al., 1979] Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1979). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226.

[Hornik et al., 1989] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:356–366.

[Jones et al., 2002] Jones, A. J., Tsui, A. P. M., and de Oliveira, A. G. (2002). Neural models of arbitrary chaotic systems: construction and the role of time delayed feedback in control and synchronization, paper id: tsui01, url: http://www.csu.edu.au/ci/vol09/tsui01/. *Complexity International*, 9.

[Kennel et al., 1992] Kennel, M. B., Brown, R., and Abarbanel, H. D. I. (1992). Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, 45(6):3403–3411.

[Končar, 1997] Končar, N. (1997). *Optimisation methodologies for direct inverse neurocontrol*. PhD thesis, Department of Computing, Imperial College of Science, Technology and Medicine, University of London.

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1:81–106.

[Takens, 1981] Takens, F. (1981). Detecting strange attractors in turbulence. In Rand, D. and Young, L., editors, *Dynamical Systems and Turbulence*, volume 898 of *Lecture Notes in Mathematics*, pages 366–381. Springer-Verlag.

[Tsui, 1999] Tsui, A. (1999). *Smooth Data Modelling and Stimulus-Response via Stabilisation of Neural Chaos*. PhD thesis, Department of Computing, Imperial College of Science, Technology and Medicine, University of London.

[Tsui et al., 2002] Tsui, A. P. M., Jones, A. J., and de Oliveira, A. G. (2002). The construction of smooth models using irregular embeddings determined by a gamma test analysis. *Neural Computing & Applications*, 10(4):318–329.

[Weigend et al., 1990] Weigend, A. S., Huberman, B. A., and Rumelhart, D. E. (1990). Predicting the future: a connectionist approach. *International Journal of Neural Systems*, 1:193–209.

[Weigend et al., 1991] Weigend, A. S., Rumelhart, D. E., and Huberman, B. A. (1991). Back-propagation, weight-elimination and time series prediction. In et al, D. S. T., editor, *Connectionist Models, Proceedings of the 1990 Summer School*. Morgan-Kaufmann, San Mateo, CA 1991.

[Wilson et al., 2003] Wilson, D., Jones, A. J., Jenkins, D. H., and Ware, J. A. (2003). Predicting housing value: Attribute selection and dependence modelling utilising the gamma test. *Paper submitted to Neural Computing & Applications*.