

Predicting Housing Value: Attribute Selection and Dependence Modelling

Utilising the Gamma Test

I. D. Wilson^{1,4}, Antonia J. Jones², D. H. Jenkins³, and J. A. Ware¹

¹ School of Computing, University of Glamorgan, Pontypridd CF37 1DL, UK

² Department of Computer Science, Cardiff University, Cardiff, CF24 3XF

³ School of Technology, University of Glamorgan, Pontypridd CF37 1DL, UK

Abstract

In this paper we show, by means of an example of its application to the problem of house price forecasting, an approach to attribute selection and dependence modelling utilising the Gamma Test (GT), a non-linear analysis algorithm that is described. The GT is employed in a two-stage process: first the GT drives a Genetic Algorithm (GA) to select a useful subset of features from a large dataset that we develop from eight economic statistical series of historical measures that may impact upon house price movement. Next we generate a predictive model utilising an Artificial Neural Network (ANN) trained to the Mean Squared Error (MSE) estimated by the GT, which accurately forecasts changes in the House Price Index (HPI). We present a background to the problem domain and demonstrate, based on results of this methodology, that the GT was of great utility in facilitating a GA based approach to extracting a sound predictive model from a large number of inputs in a data-point sparse real-world application.

Keywords: Gamma Test, Genetic Algorithm, Attribute Selection, ANN, Prediction, House Price

⁴ Corresponding Author: Dr. Ian D. Wilson, School of Computing, University of Glamorgan, Pontypridd, CF37 1DL, UK. Phone: +44 (0)1443 482268 Fax: +44 (0)1443 482715 e-mail: idwilson@glam.ac.uk

1 Introduction

Development of data-derived models (for example, Artificial Neural Networks) of smooth systems, where the objective is to construct a model directly from a set of measurements of the system's behaviour is often problematic. This is especially true of systems where the relative utility of each metric is unclear, where a large number of potentially useful inputs exist and where data is either sparse or contains a high level of noise. These problems are often addressed iteratively, with data-driven models being constructed, analysed and adjusted before repeating the cycle until either a

useful model is constructed or it becomes apparent that the system being modelled is not smooth. Clearly, this iterative procedure is labour intensive requiring considerable experience and skill on the part of the practitioner.

The Gamma Test [1] (GT) was originally developed to facilitate the exercise of constructing data-derived models of smooth systems, i.e. a model where the transformation from input to output is continuous and has bounded first partial derivatives over the input space [2]. The GT procedure provides an estimate for the noise level present in a data set computed directly from the data without assuming any a priori knowledge about the system. The GT provides a measure of the quality of the data that, in cases of high noise, indicates when a smooth model does not exist within the data. ANN derived models generalise from useful data first and reach a point where continuing the process tends to ‘over train’ the network, a situation where noise is incorporated into the model. Therefore, providing a measure of the noise inherent within a data set before the modelling exercise begins supplies a point at which the training exercise should terminate. However, the utility of a data-derived measure of noise within a non-linear system extends further. That is, supplying an estimate of the level of noise within a data set provides a means for extracting useful features before the iterative procedure of deriving a data-driven model begins.

In this paper, the authors demonstrate, by means of a real-world example that is familiar to many, the practical utility of the GT to the area of attribute selection and dependence modelling. Specifically, the authors show how a GA driven by a GT derived objective function can heuristically generate a sample of different attribute selections that, when analysed, strongly indicates which attributes are salient. In addition, the authors show how these salient attributes were utilised to generate an ANN model of the underlying relationship that resulted in good actual forecasts and trends that closely followed the actual.

First, the reader is presented with an overview of the GT, GA and ANN procedures and the problem domain. Next, an explanation is given of how these procedures were applied to the domain problem to extract predictively useful features and associated forecasts. Finally, conclusions are drawn and plans for future work explained.

2 The Gamma Test (GT): Overview

The Gamma Test is a non-linear data analysis algorithm that estimates that part of the variance of the output of an input/output data set

$$\{(\mathbf{x}_i, y_i) \mid 1 \leq i \leq M\} \quad (1)$$

that cannot be accounted for by the existence of any smooth model based on the inputs, even though the model is unknown. Here $\mathbf{x}_i = (x_1(i), \dots, x_m(i))$ represents the i th data input vector and y_i represents the associated output.

We imagine that the data is derived from an underlying smooth function $f: \mathbb{R}^m \rightarrow \mathbb{R}$ which is *unknown* and that the measured output values y are given by

$$y = f(x) + r \quad (2)$$

where r is a random variable with mean zero and bounded variance $\text{Var}(r)$.

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

$$\delta_m(k) = \frac{1}{M} \sum_{i=1}^M |x_{N[i,k]} - x_i|^2 \quad (1 \leq k \leq p) \quad (3)$$

where $|\cdot|$ denotes Euclidean distance, and

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

Note here that $y_{N[i,k]}$ is not necessarily the k th nearest neighbour of y_i in output space. Finally the regression line $\gamma = \Gamma + A\delta$ 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 $\text{Var}(r)$. The slope parameter A is also return as this normally contains useful information regarding the complexity of the unknown surface $y = f(x)$.

A formal proof that $\Gamma \rightarrow \text{Var}(r)$ in probability as $M \rightarrow \infty$ under a wide range of circumstances can be found in [3, 4]. The idea is based on the remarkable observation that the relationship between $\gamma_M(k)$ and $\delta_M(k)$ is *approximately linear* in probability as M becomes large, i.e.

$$\gamma_M(k) \approx \text{Var}(r) + A\delta_M(k) + o(\delta_M(k)) \quad (5)$$

with probability one as $M \rightarrow \infty$.

If linear regression is characterised 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.

Ideally 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 estimate for $\text{Var}(r)$. 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.

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.

However, the Gamma test has other implications that are of relevance to the present study: 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 $f: \mathbb{R}^m \rightarrow \mathbb{R}$ 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, 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. Moreover, for the purpose of comparing one selection of input variables with another, even if M is smaller than we would prefer, it may not be critical that individual Gamma test results are rather less accurate than one would like *provided* they are all computed using the same data. This is because we are primarily interested in ranking selections of inputs in order of their Gamma statistics rather than the Gamma statistic per se.

3 Genetic Algorithm overview

This section provides an introduction to the, general, GA search procedure highlighting the design methodology adopted. Genetic Algorithms are adaptive search methods that can be used to solve optimisation problems. They are based on the genetic process of evolution within biological organisms. Which is to say that, over many generations, populations have evolved according to the principles of natural selection. By adopting this process, a GA is able to ‘evolve’ solutions to real world problems [5].

Solutions are evolved utilising a genome (or structure of the problem, where a single instance of which represents a solution to the problem) and a genetic algorithm (the procedure utilised to control how evolution takes place). The GA makes use of genome operators (associated with the genome) and selection/replacement strategies (associated with the

GA) to generate new individuals. The GA uses an objective function to determine how fit each of these individual genomes is for survival. So, given a choice of GA, three things are required to solve a problem:

- A representation for the genome (determined from the definition of the problem);
- Suitable genetic operators;
- An objective function that measures the relative quality of a solution.

In summary, when using a GA to solve an optimisation problem, a series of variables are combined to form a single solution to the problem within a single genome. The GA creates a population of solutions based on the genome. The GA then operates on this population to evolve an optimum, or near optimum, solution to the problem utilising the objective function. Given this overview, the following sections expand upon each of these components.

3.1 The Genome

This section outlines the decision making process that determines how an individual solution (the genome) should be modelled and physically represented. When defining a representation appropriate to the problem at hand, a data structure that is minimal but also completely expressive should be selected. Although it may appear beneficial to include extra genetic material beyond that which is required to express a solution fully, this tends to increase the size of the search space and hinder the performance of the algorithm. Finally, each genome will have a 'fitness' score associated with it that determines its prospects for selection. This representation is an independent component of the general GA procedure, allowing separate decision making processes to be made. For example, a different GA procedure might be adopted without any need to change the structure of the genome. This is possible because the operators necessary to evolve new solutions, described in the next section, are associated with the genome and not the GA itself.

3.2 The Genome Operators

Given a general GA procedure and genome (described in Section 3.1), it is also necessary to determine how operators specific to the genome should behave. This section describes how these operators act upon the genome within a general GA procedure. Three operators can be applied to the genome, these being initialisation, mutation and crossover. These operators allow a population to be given a particular bias, and allow for mutations or crossovers specific to the problem representation. The initialisation operator determines how each genome is initialised. Here, the genome is 'filled' with the genetic material from which all new solutions will evolve. Next, the mutation operator defines the procedure for mutating the genome. Mutation is only rarely applied and randomly alters a gene in a selected 'child'. It provides a small amount of random search that facilitates convergence at the global optimum. Finally, the crossover operator defines the

procedure for generating a child from two parent genomes. The crossover operator produces new individuals as ‘offspring’, which share some features taken from each ‘parent’.

These operators are independent functions in themselves, specific to the structure of the genome, which may be altered in isolation to the other components described in these sections. For example, the crossover operator might be changed from a single point (adopted by the authors and explained in Section 8.3) to a two-point implementation without any need to adjust the other components.

3.3 Objective Functions and Fitness Scaling

This section describes how an objective-function and fitness-scaling fits into a general GA procedure. Genetic algorithms are often more attractive than gradient search methods because they do not require complicated differential equations or a smooth search space. The genetic algorithm needs only a single measure of how good an individual is compared with the other individuals. The objective function provides this, needing only a genome, or solution, and genome specific instructions for assigning and returning a measure of the solution's quality. The objective score is the raw value returned by the objective function. The fitness score is the possibly transformed objective score used by the genetic algorithm to determine the fitness of individuals for mating. Typically, the fitness score is obtained by a linear scaling of the raw objective scores. Given this, the objective function can be altered in isolation from the GA procedure and genome operators, and, once a representation for the problem has been decided upon, without any need to change the structure of the genome.

3.4 The Genetic Algorithm

Here, we present an overview of a general GA procedure, explaining how each phase fits into the evolutionary process. The GA procedure determines when the population is initialised, which individuals should survive, which should reproduce, and which should die. At each generation certain, highly fit, individuals (determined by the scaled, objective function) are allowed to reproduce (through selection) by ‘breeding’ (using the crossover operator described in Section 8.3) with other individuals within the population. Offspring may then undergo mutation, which is to say that a small part of their genetic material is altered.

Offspring are then inserted into the population, in general replacing the worst members of the existing population although other strategies exist (*e.g.* random). Typically, evolution stops after a given number of generations, but fitness of best solution, population convergence, or any other problem specific criterion can be used. Given this overview of the general design methodology adopted, the following sections describe the domain problem and how the problem was modelled.

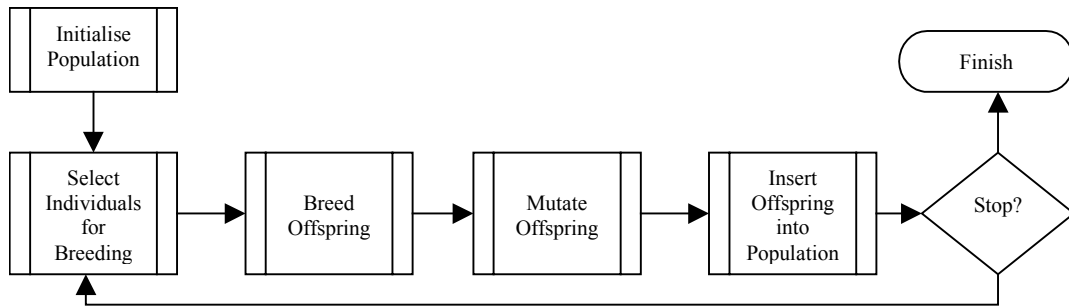
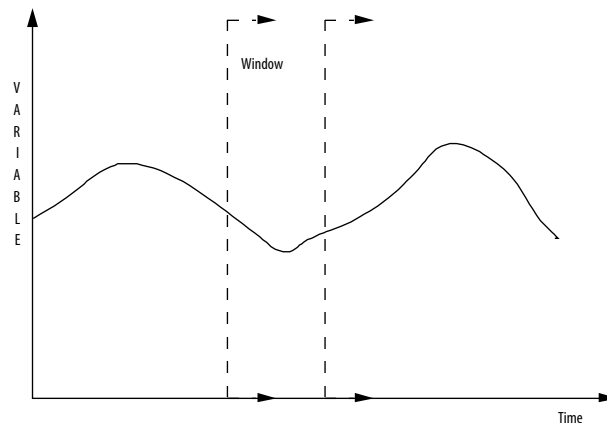


Figure 1 Genetic Algorithm Evolution Procedure

4 Forecasting with ANNs

The forward projection of a value that varies with time can be approximated from a model of the time series using a variety of traditional techniques. Most such schemes rely on the assumption of linearity, or on some kind of transformation of the data such as a logarithmic or a cosine. The necessary function has to be postulated however by the user for each individual time series.



Record	Inputs			Output or targets	
1	y_{t+2}	y_{t+3}	y_{t+4}	y_{t+5}	y_{t+6}
2	y_{t+1}	y_{t+2}	y_{t+3}	y_{t+4}	y_{t+5}
3	y_t	y_{t+1}	y_{t+2}	y_{t+3}	y_{t+4}
4	y_{t-1}	y_t	y_{t+1}	y_{t+2}	y_{t+3}
5			<i>etc.</i>		

Table 1—The time series is organised as a data table, each record being a small window of the complete time series. The full table comprising of the time series back to a convenient point in time is used to train and validate the neural network. In this example, y_{t+6} is the most recent value of the time series.

Neural networks are pattern recognition devices that can be used effectively to fit any smooth non-linear function in data. They have the advantage that there is no need to assume any transformation or functional form in the data. Windowing is a technique often used to enable ANN's to model a time series. The network has multiple inputs, $x_1 \dots$

x_m and an output y . The data is arranged in to $m+1$ columns, each column being a repeat of the one on its right, but displaced down by one time unit. If the successive values of the time series are represented by $y_t, y_{t+1}, y_{t-2} \dots$, then Table 1 shows the arrangement of the data, assuming for example that $m=4$. The number of columns is decided by the length of the data set, and the width of window considered necessary to include all the major features of the time series, a figure that is often found by trial and error.

Once a model has been trained and validated, it can be used to project the series forward in time. The most recent n values of the time series (up to y_{t+6} in this case) are applied to the neural network and the output gives the first projected value of the time series, y_{t+7} . Next, the last $n-1$ values of the series, plus the value y_{t+7} are applied to the neural network, which now gives y_{t+8} .

Often it is desirable to predict a time-series that is thought to be dependent on other time series. The input to the network is formed by concatenating the data from appropriate 'windows' of the independent time series while the output is the predicted value for the dependent time series. However, mere concatenation often leads to lengthy (which is not a good feature when training neural networks) input vectors that may contain non-salient information. The aim of feature selection is to reduce the length of the training vector by removing non-salient columns from the concatenated vector.

5 The Problem Domain: House Price Formation

Periodic overheating of residential property markets is a feature of developed economies. It is a feature that has the potential to inflict substantial socio-economic damage. Avoiding the worst effects has become a policy objective for governments' central banks [6]. Given that approximately sixty-five percent of UK residential property is mortgaged, the activity of lenders provides a potential brake to curb the worst excesses of the market. In particular it would be helpful if lending did not fuel speculation. At the moment, lenders rely on 'open market value', a bid price which by definition reflects market froth, as the key metric when determining a ceiling figure for lending. A move away from current bid price to a metric that reflected a more sustainable value would realise the policy objective without heavy regulation. Such an alternative metric – that restrains the speculative component – would need to be forward looking. Whereas the 'open market value' is determined by reference to comparable concurrent transactions, any measure of sustainable value would need to be predictive. This is not a trifling pursuit; the complexity is perhaps on a par with setting central bank lending rates. Incidentally, we might anticipate that the analysis identifies one or more heuristics similar to the Taylor rule that assists in setting the 'appropriate' interest rate at the American Treasury Department. We call this more prudent metric 'sustainable market value' and given that, historically, the highest risk in residential

lending occurs during the first three years of a mortgage, we define 'sustainable' using a three year horizon [7]. The development of models that provide a sustainable valuation for properties would be of great usefulness to the lender and consumer. The measure would provide the lender with a more sober reflection of risk. The consumer would be able to make informed decisions based upon pricing models that give a clearer indication of the real, sustainable, value of property. 'Sustainable market value' would forestall negative equity and facilitate movement between jobs in today's mobile labour market.

Two major problems exist when model building in this domain. The first problem is theoretical, the second computational. Economic theory is hardly a finished epistemological category. Writing of the 'Death of Economics' in 1994 then forecaster at Henley Management College, Professor Paul Ormerod, suggested of current economic theory that it "should be abandoned or at least suspended until it can find a sounder economic base." And of economic forecasting in particular, "By ignoring non-linearity, forecasters constantly get things wrong--missing, for example, the contagion of fear that infected Asia and the world after the fall of the Thai baht in 1997." [8]. What appears true of economic theory in general is also likely to be true of theoretical market models of residential markets in particular [9].

The extent of the computational problem can be gauged from the fact that outside the subjective, comparison heuristic used by professional valuers, no coherent model exists even for the calculation of current bid price; the countless factors that influence value have nowhere been systematised [10]. Furthermore, unless it can be proven that Takens' Theorem applies in this domain, the prediction of sustainable values is likely to be more rather than less complex than the calculation of open market values. This suggests that the kind of modelling strategy pursued here is relevant and may be significant.

6 Description of the data

Competing theories are used to explain the behaviour of markets at national, regional and urban aggregates. However, UK national and regional level models have developed primarily from the modelling of the market at the macroeconomic level and these models have not been integrated with modelling at the urban level, where professional valuers operate. In order to make some headway, we decided that the initial focus should be on the prediction of values at a national level where data has been systematically recorded for 30 years and sometimes longer.

In this paper we have not chosen data on the basis of a particular theory, rather data sets were chosen primarily because of their consistent use across the various models described in the literature, which we classify as follows:

- General, related, models [6], [9], [11], [12];
- Hedonic (a measure of general, overall opinion) regression analysis models [13]-[15];

- Artificial intelligence [16], [17], including ANNs [18]-[21].

Such models indicate that the main variables expected to influence house prices at both the national and regional levels include incomes; interest rates (real or nominal); the general level of prices; household wealth; demographic variables; the tax structure; financial liberalisation and the housing stock [9], [22] developed a highly condensed forecasting model using just three variables widely thought to play a significant causal role together with a house price index:

- Real House Price (log of house price index divided by Retail Price Index - RPI);
- Real incomes (log of real disposable incomes at constant prices);
- Retail prices (log of RPI);
- Mortgage interest rate (tax-adjusted interest rate).

In this pilot we have included these last variables in the form of quarterly percentage changes in the Bank Rate (BR)¹, the Retail Price Index (RPI)², and the Average Earnings Index (AEI)³.

However, given the data-driven nature of this model and given too that variables will be ranked in terms of their significance to model building, we were not constrained by the exigencies of economic theory. In addition to data that are thought to have a strong causal connection to house price changes, we were able to select additional time series that may have a weak (or no) connection with house price formation, or which may simply be associated with changing levels of activity/pricing in housing markets. In fact we restricted ourselves to quarterly percentage changes in the following additional variables, which *a priori* were assumed to be associated with if not causes of house price changes: claimant count (CC)⁴; consumption of durable goods (DG); GDP household consumption (GDPHC); household savings rate (HHSR) and rates of mortgage equity withdrawal (MEW).

¹ Description: TABLE 20.1: Bank Of England Money Market Intervention Rates: Changes In Bank Rate, Minimum Lending Rate, Minimum Band 1 Dealing Rate And Repo Rate Source: Bank of England

² Description: CZBH: Percentage change over 12 months (headline rate), Annual 1949 to 1999, Monthly 1948 06 to 2000 (updated approximately monthly), Quarterly 1948 Q3 to 2000 (updated approximately quarterly) Source: Office for National Statistics

³ Description: LNMU: Average earnings, Percentage change over 12 months, seasonally adjusted Monthly 1964 01 to 2000 (updated approximately monthly) Source: Office for National Statistics

⁴ Description: BCJE: Claimant count, 1950 – 2000. Estimates of claimant count (the number of people claiming unemployment related benefits) in the UK; the level (thousands) and as a percentage rate. Source: Office for National Statistics

We also injected a degree of anonymity into the modelling by not disclosing to the non-economist model builder *a priori* expectations of the likely strength of these variables, nor the lagged observation(s) in which they might be expected to feature. The house price index data in this paper relates to quarterly changes in the All UK: Average House Price (£) (House Price Index - HPI) Nationwide Building Society (see Appendix A further information).

7 Problem composition and modelling

In this section, considerations relating to the attribute generalisation procedure are described, and an overview of the model's underlying representation and physical implementation is provided, along with a detailed discussion about the objective function and its mathematical formulation.

7.1 Problem size and complexity

The eight attributes contained in the economic indicator database (quarterly measurements from 1975) that are converted into windows of length six and concatenated together provide a vector of 48 inputs and one output value. This configuration provides a search space of approximately 2.8×10^{20} combinations.

7.2 Data pre-processing

Data is often manipulated using pre-processing routines, such as data scaling, sectioning, smoothing, and removal of outliers, before being used to construct an underlying model. However, the authors made the deliberate decision to allow the data-mining procedure described in this paper to drive the decision making process without any *a priori* assumptions about the data being made. Therefore, the data was simply transformed from a series of actual values into one containing the, quarterly, annual percentage movements of each series (see Appendix B for more information). This is important, given that the object was to reduce the windowed input vector for each independent time-series where possible to a single lagged observation. This decision was based upon an assumption that a combination of single, differently lagged, observations of the direction and magnitude of the input time-series could, in combination, be used to determine the change in the output time series. In other words, was it possible to replace a window of discrete observations with a single measure of the magnitude and direction of the time-series (i.e. the annual percentage movement)? To this end, each economic time-series was independently 'windowised' (described in Section 4) and then concatenated together to provide a set of input vectors. Then, for each input vector was concatenated with a single value of the annual movement in the HPI, four quarters in advance of the end of the period covered by each input vector.

7.3 Genome representation

Our state representation, the genome (introduced in 3.1), is physically stored as a string of Boolean values (a 'mask') and its corresponding fitness value. The mask is effectively a series of on/off switches that correspond with the columns within our database (described in Section 7.2) of input vectors (illustrated in Figure 2).

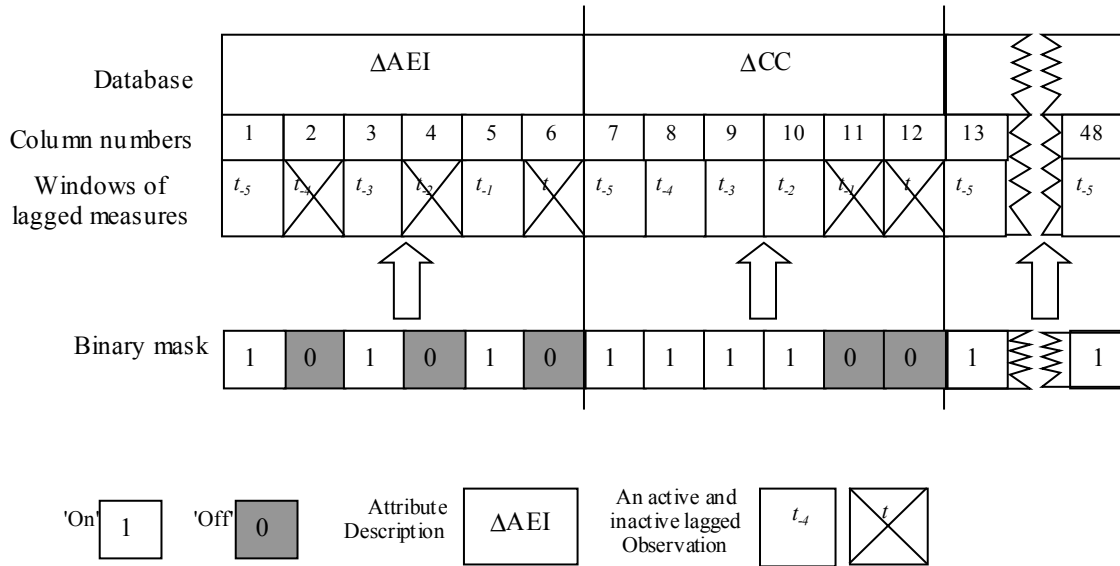


Figure 2 Mapping within the State Representation

The fitness value for any given genome is calculated using only the active columns determined by its mask (e.g. the mask provided by the partial genome shown in Figure 2 indicates that only columns $\{1, 3, 5, 7, 8, 9, 10, 13, \dots, 48\}$ were utilised when determining its fitness score). This physical implementation is easily manipulated using the GA procedures introduced in Section 3 and explained further in Section 8.

7.4 Mask evaluation

The success of any discrete optimisation problem rests upon its objective function, the purpose of which is to provide a measure for any given solution that represents its relative quality. In this section, we present the attribute selection strategies and their formulation within an objective function.

The objective function used here works by calculating the Gamma statistic for the data for a given attribute mask within our state representation and then summing metrics associated with the quality of the mask. Hence, the objective score associated with a given configuration is an abstraction of the penalties analogous with the relationships between a set of attributes determined by the configuration. In full, we consider three measures of relative quality [23], namely:

- the amount of noise within the database (should be reduced to a minimum);
- the underlying complexity of any underlying relationship between input and output data (should be minimised);

- the complexity of any ANN architecture utilised (should be optimally-minimal in terms of the number of inputs into the architecture).

7.4.1 Underlying definitions and constraints.

The objective function utilised to evaluate solutions requires a number of definitions, namely:

- $P(t)$: $\{m_1, \dots, m_m\}$ is the population of masks at generation t ;
- M : $\{m_1, \dots, m_s\}$ is the set, of length s , of all possible masks;
- t is the generation;
- n is the number of members in the population;
- s is the number of individual masks (2^a) that exist within the state-space;
- a is the total number of attribute measurements within a given mask;
- m_{ti} is a given mask i of the population at generation t ;
- w_1 is the weight given to the intercept value;
- w_2 is the weight given to the gradient value;
- w_3 is the weight given to the number of active attributes;
- $(w_1 + w_2 + w_3) > 0$;
- $f_1(mask)$ is a function that returns the Gamma statistic for a given mask $mask$ such that $0 \leq f_1(mask) \leq 1$;
- $f_2(mask)$ is a function that returns an estimate of the model's complexity for a given mask $mask$ such that $f_2(mask) \geq 0$;
- $f_3(mask)$ is a function that returns the number of active attributes within a given mask $mask$ such that $f_3(mask) \geq 0$;
- $f(mask)$ is a function that returns the weighted objective score for a given mask $mask$ such that $f_2(mask) \geq 0$;
- $V_{ratio}(mask)$ is a function that returns Gamma/Var(output), providing a standardised measure;
- $Active(mask)$ is a function that counts the number of active attributes within a given mask m ;
- $Length(mask)$ is a function that returns the number of attributes within a given mask m ;
- $Gradient(mask)$ is a function that returns the slope of the regression line used to calculate the Gamma statistic.

7.4.2 The object relationship fitness function

The objective function used to evaluate masks examines the weighted relationship between relative measures of its quality [*ibid.*]. The general expression of the objective function is:

$$f(mask) = 1 - (w_1 f_1(mask) + w_2 f_2(mask) + w_3 f_3(mask)) \quad (6)$$

Where $f_i(mask)$ and w_i represent, respectively, the number of conflicting objects and the weight of that particular measure, with a high value of $f(mask)$ indicating a good solution. These values are then scaled to a positive range.

The first term of the objective function, $f_1(mask)$, returns a measure of the quality of the intercept based upon the Vratio, which is a standardised measure of the Gamma statistic that enables a judgement to be made, independently of the output range, as to how well the output can be modelled by a smooth function. Minimising the intercept by examining different mask combinations provides a means for eliminating noisy attributes, facilitating the extraction of a smooth model from the input data, and is expressed as:

$$f_1(mask) = \left\{ \begin{array}{ll} 1 - \frac{1}{1 - 10V_{ratio}(mask)} & \text{if } V_{ratio}(mask) < 0 \\ 2 - \frac{2}{1 + V_{ratio}(mask)} & \text{otherwise} \end{array} \right\} \quad (7)$$

The second term of the objective function, f_2 , returns a measure of the complexity of any underlying smooth model based upon the gradient determined by the GT. Minimising this complexity is desirable, especially in cases where data points are relatively sparse, and is expressed as where $|\cdot|$ denotes the absolute value:

$$f_2(mask) = 1 - \frac{1}{1 + \left| \frac{gradient(mask)}{outputRange} \right|} \quad (8)$$

Lastly, the third term of the objective function, f_3 , sums the number of active elements within the input vector and returns this as a percentage of the total number elements within the input vector. Minimising the number of active genes within our genome encourages the search procedure to find solutions that are less complex, resulting in input minimal input vectors.

$$f_3 = \frac{Active(mask)}{Length(mask)} \quad (9)$$

The next section deals with our GA implementation, with special consideration being given to the sub-ordinate heuristics used to direct the procedure through the search space.

8 Attribute Selection and Dependence Modelling

Of the variations of GA available, the work presented in this paper utilised an approach similar to the Simple GA (SGA) introduced by Goldberg [5]. Our SGA uses overlapping populations but with a pre-specified amount of overlap

(expressed here as a percentage), these being the initial and next generation populations. The SGA first creates a population of individuals by cloning the initial genome. Then, at each generation during evolution, the SGA creates a temporary population of individuals, adds these to the previous population and then removes the worst individuals in order that the current population is returned to its original size. This strategy means that the newly generated offspring may or may not remain within the new population, dependant upon how they measure up against the existing members of the population. The following sections examine each component of our SGA implementation.

8.1 Configuration, search space and cost function

Given a set of lagged observations (w) for a number of economic metrics (n), a configuration s corresponds to an individual mask of Boolean values associated with each attribute. The search space S is therefore composed of all such configurations. According to equation (6), for each solution $s \in S$, $f(s)$ corresponds to a combination of minimal intercept, gradient and active attribute values.

8.2 Population size and maximum generations

The population size was set to 100 for all experiments, however, larger population sizes would facilitate the attribute selection procedure at the expense of longer run times. The maximum number of generations varied between experiments with the search procedure terminating upon population convergence (defined as when the average fitness score stabilised).

8.3 Crossover, replacement and mutation

The probability of crossover (selection for sexual reproduction) determines the proportion of parents within the population that will be selected for crossover at each generation. The single-point crossover strategy (illustrated in Figure 3) was adopted for all experiments.

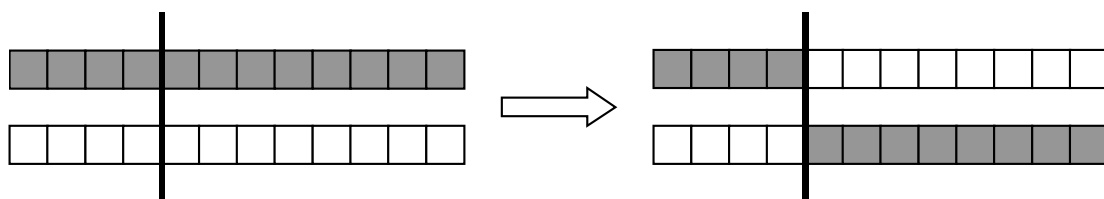


Figure 3 Single point crossover

Each time crossover occurs, two offspring were created using material from each of its parents. The results for all experiments presented in this paper were generated using a crossover percentage of 50%, which is to say that at each generation 50% of the new population were generated by splicing two parts of each genomes' parents together to generate two new genomes. The position of the join is determined randomly for each pair of parents.

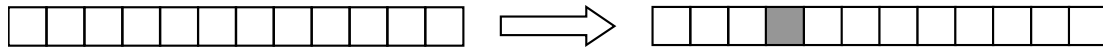


Figure 4 Single bit mutation

Mutation is introduced to facilitate movement away from local optimality to a place closer to global optimality. However, mutation should not occur too often, as this would be detrimental to the search exercise. Consequently, the results presented here were generated using a 5% mutation probability, which was determined experimentally, utilising a single bit flip mutation operator (illustrated in Figure 4).

9 Attribute selection computational experiments and results

The experimental work that formed the basis for this paper utilised nine economic metrics dating from 1974 to 2001, which were pre-processed to provide measures of annual percentage movement for each of the series. The eight input metrics were converted into a set of vectors, of length six, which were concatenated together, along with a single corresponding output metric to provide 97 vectors. Experiments with the SGA produced the consolidated results presented below, where, for every member of the population, the number of times an attribute is active is totalled to provide an indication of its significance.

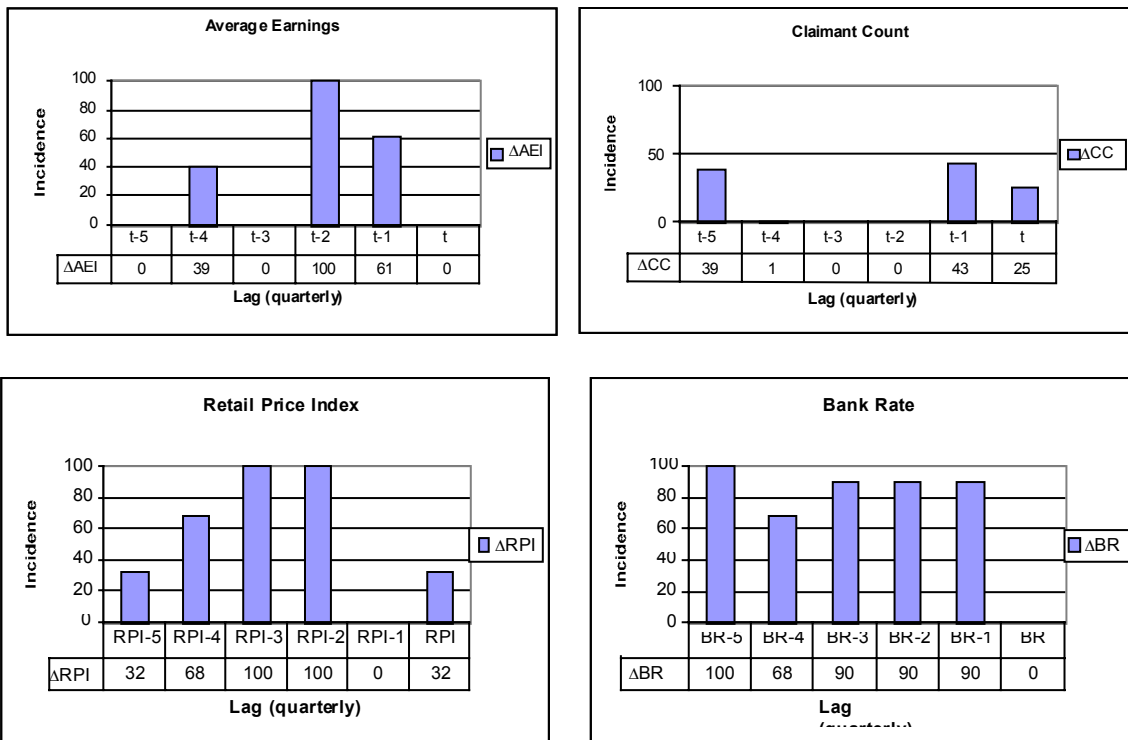




Figure 5 Graphs showing the sum of how often each lagged observation occurs in the population.

Totalling how often each lagged observation occurs within the population (presented graphically in Figure 5) provides a useful means for pruning the input vector. In addition to this, totalling the number of times each lagged observation occurs within a each attribute provides a useful heuristic for determining the weight each attribute has on the outcome (illustrated in graphically in Figure 6).

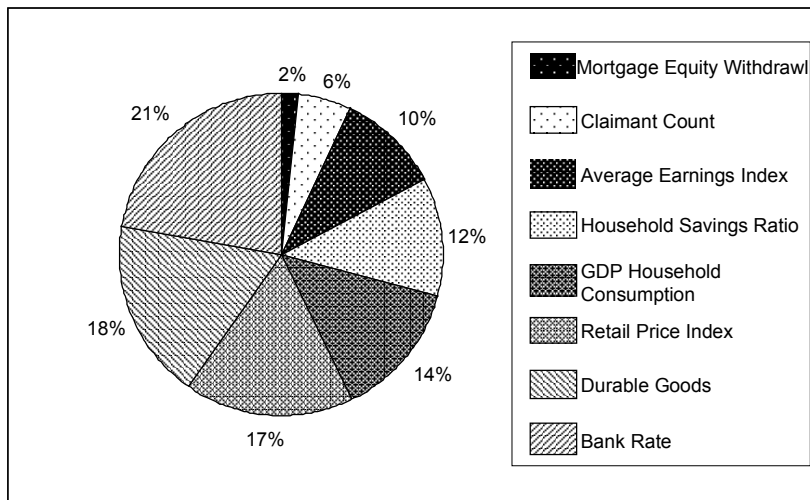


Figure 6 Analysis of relative importance of each metric.

The algorithms and features referenced in this paper were implemented in VC++ 6.0 running under Windows NT on a Viglen P3 (800MHz Pentium) with 128 Megabytes of RAM. Experimental data shows how certain lagged observations appear more often than others do. These results confirm *a priori* expectations to a significant degree. For example, it is noticeable that the recent behaviour of Average Earnings (see Figure 5) is more important than in earlier periods as would be expected. The results suggest that the Bank Rate and Retail Price Index are consistently significant in house price formation. If this model provides a ‘useful’ forecast of the house price index, it would confirm that their widespread inclusion in models as causal variables is rational.

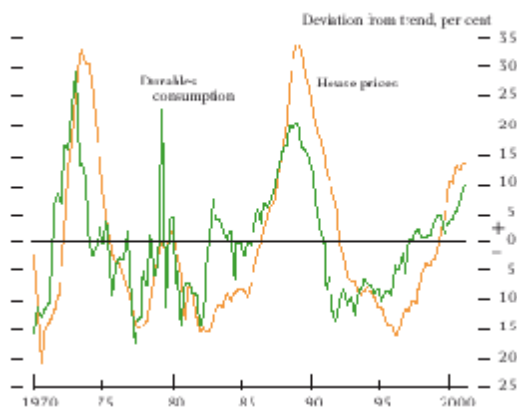


Figure 7 Movement in house prices and durable goods consumption [6].

Figure 6 ranks the variables by frequency of occurrence. The Bank Rate and Retail Price Index head the list as expected. Interestingly the Consumption of Durable Goods, not notable as a cause of house price changes per se, is also conspicuous. However, this is also expected. It is a feature of recent economic performance that the Consumption of Durable Goods and Average House Price are strongly correlated as Figure 7 demonstrates. At the other end of the scale it is observable that Mortgage Equity Withdrawal is a relatively weak input. While there are moments when this variable plays a role, across the whole time series it is expected that this would be of lesser significance.

Selecting those lagged observations that occurred in each member of the population (i.e. 100 times) significantly pruned the length of the original input vector (from 48 to 8). However, it was decided to heuristically reduce this to a single observation for each economic metric by again utilising the GT procedure. This, further, heuristic pruning of the suggested mask involved systematically eliminating each of the two ΔRPI and ΔDG metrics and measuring its affect on the Gamma Statistic and Gradient. Choosing the lagged value for the ΔRPI and ΔDG that least affected the Gamma and Gradient measures enabled a single most significant measure for each of these indices to be selected.

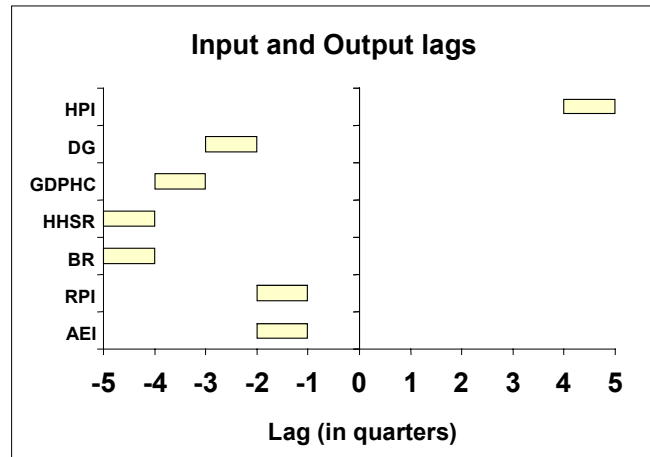


Figure 8 Lagged Observations

This procedure provided an input vector made up of varying lagged movements taken from each of the six predictively useful economic metrics (shown in Figure 8) ready for the next, or predictive, step in the modelling exercise.

10 Forecasting using Artificial Neural Networks

Despite the many satisfactory characteristics of an ANN, building a neural network for a particular forecasting problem is a nontrivial task. Modelling issues that affect the performance of an ANN must be considered carefully. First, an appropriate architecture, that is, the number of layers, the number of nodes in each layer, and the number of arcs that interconnect with the nodes must be determined. Other network design decisions include the choice of activation function for the processing nodes, the training algorithm, data normalisation methods, training data, and performance measures [24]. In this section, an overview of the Back-Propagation ANN utilised to forecast a change in the HPI is provided.

10.1 The network architecture

Our ANN is composed of an input layer, which corresponds to the length of the input vector, an output layer, which provides the forecast values, and two layers of hidden nodes. It has been shown that a single hidden layer is sufficient for an ANN to approximate any complex non-linear function with any desired accuracy [25]. However, recent findings have shown that two hidden layers can result in a more compact architecture that achieves a higher efficiency than single hidden layer networks [26]-[28].

10.1.1 The number of nodes in the hidden layers

It is important that the network has generalised across the time series and not simply fitted the inputs to their corresponding outputs. Therefore, the number of hidden nodes in each layer was determined by trial and error, with

large numbers of nodes in the hidden layers being incrementally pruned to a minimum (of four nodes in each of the two hidden layers) whilst still producing relatively good forecasting capabilities.

10.1.2 The number of nodes in the input layer

The number of nodes in the input layer corresponds to the length of the *mask* generated during the attribute selection procedure described earlier. This is the most critical decision variable for a forecasting problem, since the vector contains important information about complex (linear and/or non-linear) structure in the data. Given that there is no widely accepted systematic way to determine the optimum length (or content) for the input vector [24] the heuristically derived mask provides a significant step forward in this area of modelling.

10.1.3 The number of nodes in the output layer

For the time series forecasting problem described in this paper, the single output node corresponds to the forecasting horizon. Here, a four-step-ahead (i.e. one-year into the future) was adopted.

10.2 Performance measure

Although there can be many performance indicators associated with the construction of an ANN the decisive measure of performance is the prediction accuracy it can achieve beyond the training data. No one universally accepted measure of accuracy is available, with a number of different measures being frequently presented in literature [29]. The performance measure adopted by the authors, is the Root Mean Squared Error (RMSE) function. The RMSE provides an averaged measure of the difference between the actual (desired) and predicted value.

10.3 Updating the weights

The new values for the network weights are calculated by multiplying the negative gradient with the learning rate parameter (set at 0.25) and adding the resultant vector to the vector of network weights attached to the current layer. In order to accelerate convergence a weighted momentum term (of 0.1) is added to the weight update.

10.4 Terminating the training procedure

As over-fitting is a widely accepted problem associated with modelling utilising ANNs, the GT's ability to accurately measure the 'noise' within a data-set and, consequently, the point at which training should stop provides a significant utility for practitioners. Over-fitting occurs because the ANN will attempt to fit all data encountered, including any noise present. Given that an ANN will tend to fit useful data before any noise, providing a measure of any noise present in the data-set is of considerable utility as it allows training to end at a near optimal point. Therefore, the GT procedure (see Section 2) was applied to the input/output mask suggested by the heuristic procedure (see Section 8) to provide a MSE value at which training was stopped.

10.5 Partitioning of the vector set

Typically, training and test data sets are used during the ANN creation process. In this paper, the training set was used to construct the ANN's underlying model of the time series and the test set was used to measure the accuracy of this model. Next, the set of vectors was partitioned into a training set and a test set. The M-competition convention of retaining the last eight quarterly points for testing, mapped to input/output vectors, was adopted [30]. The remaining vectors were formed the training set. As the Gamma test statistic was used to stop the training procedure, there was no need to further partition the training set to provide a Validation Set [31].

11 Forecasting utilising an ANN experimental work

In this section, the authors present the predicted and actual annual movements in the HPI in addition to the actual and predicted values for the HPI taking into account the predicted percentage change produced by the ANN model.

11.1 Annual percentage movement results

Experimental work utilising the training set of ninety, varying lagged, six indicator input-vectors to produced an ANN model that, when tested using the last eight quarters of data, produced the results presented in Figure 9. The model resulted in an average error percentage of 9.6% and a trend line that closely followed the actual in all but one of the test vectors, i.e. the direction of the change was accurately predicted in seven of the eight test quarters.

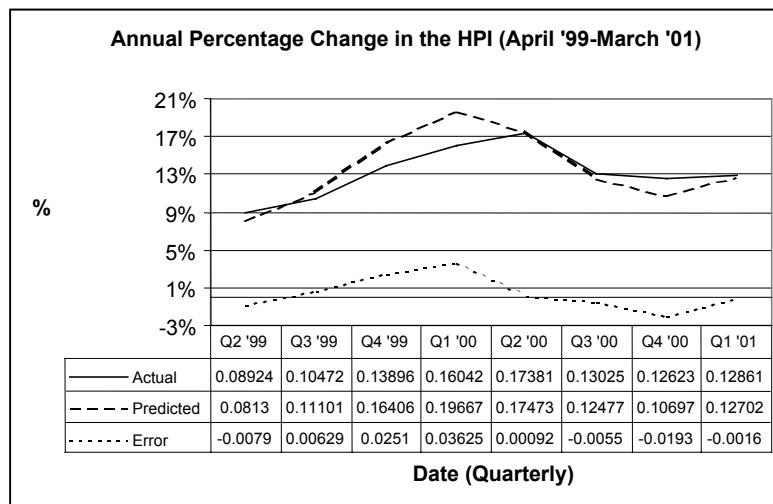


Figure 9 Annual percentage movement forecast results.

11.2 Actual movements in the House Price Index

Factoring the predicted percentage movement into the actual HPI resulted in forecasts with a range of 3% and an average of 1.1% (shown in Figure 10). The model built using the training data resulted in a standard deviation of the predictive error (0.066901), which is approximately 7% of the range.

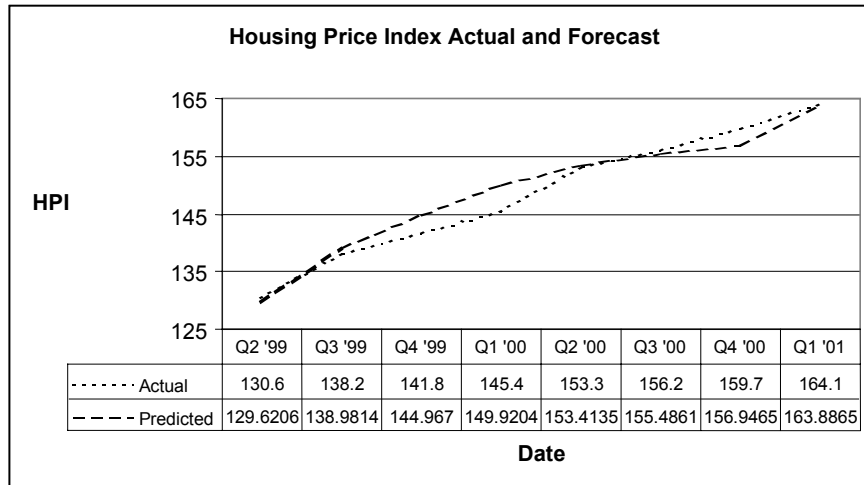


Figure 10 Housing Price Index forecast

12 Conclusion

This work has shown that promising forecasting models can be produced using an ANN trained to a MSE suggested by the GT. In addition, it has been shown how predictively useful indicators can be heuristically selected from a database of economic metrics utilising a GT/GA procedure.

13 Future Work

While care was taken in the selection of inputs for this pilot study, there may well be other useful indicators which could improve the performance of the present model. We have already identified a number of other time series that may further improve the usefulness of the forecasts beyond the national aggregate market to specific sub-markets and further work on these issues is planned.

Despite epistemological caveats, in general analysis such as this can be useful in refuting or confirming conventional wisdom regarding the relative importance of useful predictive variables. For example, there are competing theories regarding house-price formation that it might also be possible to test. One school of thought suggests that *land prices* are an important determinant of house price (at least in certain sub-markets). It will be relatively straightforward

to run models with and without the variables suggested by these commentators. We also intend to test the models at regional and then urban aggregates where data is available.

References

1. A. Stefánsson, N. Koncar, and A. J. Jones, "A Note on the Gamma Test," *Neural Computing and Applications*, vol. 5, pp. 131-133, 1997.
2. A. J. Jones, D. Evans, S. Margetts, and P. J. Durrant, *Heuristic and Optimisation for Knowledge Discovery*. Idea Publishing, Hershey, PA, 2002, ch. 9.
3. D. Evans and A. J. Jones, "A proof of the Gamma test," *Proc. Roy. Soc., Series A.*, 2002, to be published.
4. D. Evans and A. J. Jones, "Asymptotic moments of near neighbour distance distributions," *Proc. Roy. Soc. Series A*, 2002, to be published.
5. D. A. Golberg, *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Wesley, 1989.
6. K. Aoki, J. Proudman, and G. Vlieghe, "Why house prices matter," *Bank of England Working Paper*, 2001.
7. D. H. Jenkins, *Residential Valuation Theory and Practice*. *Estates Gazette*, 2002.
8. P. Ormerod, *Butterfly Economics: A New General Theory of Social and Economic Behaviour*. Pantheon Books; ASIN: 0375407650, 1998.
9. G. Meen and M. Andrew, "Modelling Regional House Prices: A Review of the Literature," *University of Reading for the Department of the Environment, Transport and the Regions*, ISBN 0 7049 1305 4, 1998.
10. D. H. Jenkins, O. M. Lewis, N.I. Almond, S. A. Gronow, and J. A. Ware, "Towards an Intelligent Residential Appraisal Model," *Journal of Property Research*, vol. 16, no. 1, pp. 67-90, 1998.
11. M. Ball and M. Grilli, "U.K. Commercial Property Investment: Time-Series Characteristics and Modelling Strategies," *Journal of Property Research*, vol. 14, no. 4, pp. 279-296, 1997.
12. R. J. Barkham and D. M. Geltner, "Price Discovery and Efficiency in the UK Housing Market," *Journal of Housing Economics*, vol. 5, no. 1, pp.41-63, 1996.
13. A. Antwi. (1995) *Multiple Regression in Property Analysis*. *Estates Gazette Interactive*. Available: <http://www.egi.co.uk/>
14. A. S. Adair, J. N. Berry, and W. S. McGreal, "Hedonic Modelling, Housing Sub-markets and Residential Valuation," *Journal of Property Research*, vol. 13, pp. 67-83, 1996.
15. ET-K Lam, "Modern Regression Models and Neural Networks for Residential Property Valuation," presented at the *Royal Institute of Chartered Surveyors, The Cutting Edge*, 1996.

16. W. McCluskey and S. Anand, "The Application of Intelligent Hybrid Techniques for the Mass Appraisal of Residential Properties," *Journal of Property Investment and Finance*, vol. 17, no. 3, pp. 218-238, 1999.
17. S. Wang, "An Adaptive Approach to Market Development Forecasting," *Neural Computing and Applications*, vol. 8, no. 1, pp. 3-8, 1999.
18. S. McGreal, A. Adair, D. McBurney, and D. Patterson, "Neural Networks: The Predication of Residential Values," *Journal of Property Valuation and Investment*, vol. 16, no. 1, pp. 57-70, 1998.
19. V. R. Vemuri and R. D. Rogers, "Artificial Neural Networks Forecasting Time Series," IEEE Computer Society Press, California, 1994.
20. E. Worzala, M. Lenk, and A. Silva, "An Exploration of Neural Networks and its Application to Real Estate Valuation," *The Journal of Real Estate Research*, vol. 10, no. 2, pp. 185-201, 1995.
21. Owen Lewis, "The Use of Artificial Intelligence Techniques to Assist in the Valuation of Residential Properties," Ph.D. dissertation, Dept. Math. and Comp., University of Glamorgan, Pontypridd, Wales, UK, 1999.
22. D. Miles and S. Andrew, "Merrill Lynch Model of the UK housing market," Financials Research, Merrill Lynch, London, 1997.
23. P. J. Durrant, "winGammaTM: a non-linear data analysis and modelling tool for the investigation of non-linear and chaotic systems with applied techniques for a flood prediction system," Ph.D. dissertation, Dept. Comp. Science, Cardiff University, Wales, UK, 2001.
24. G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with Artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35-62, 1998.
25. K. Hornik, "Approximation capabilities of multi-layer feed-forward networks," *Neural Networks*, vol. 4, pp. 251-257, 1991.
26. D. Srinivasan, A. C. Liew, and C. S. Chang, "A neural network short-term load forecaster," *Electric Power Systems Research*, vol. 28, pp. 227-234, 1994.
27. X. Zhang, "Time series analysis and prediction by neural networks," *Optimization Methods and Software*, vol. 4, pp. 151-170, 1994.
28. D. L. Chester, "Why two hidden layers are better than one?" in *Proc. International Joint Conference on Neural Networks, IJCNN-90-WASH-DC, 1990*, vol. 1, pp. 265-268.
29. S. Makridakis, S. C. Wheelwright, and V. E. McGee, *Forecasting: Methods and Applications*, 2nd ed. John Wiley, New York, 1983.

30. W. R. Foster, F. Collopy, and L. H. Ungar, "Neural network forecasting of short, noisy time series," *Computers and Chemical Engineering*, vol. 16, no. 4, pp. 293-297, 1992.
31. I. D. Wilson, S. D. Paris, J. A. Ware, and D. H. Jenkins, "Residential property price time series forecasting with neural networks," *Knowledge-Based Systems*, vol. 15, no. 5-6, pp. 335-341, 2002.

Appendix A Graphics of the source data

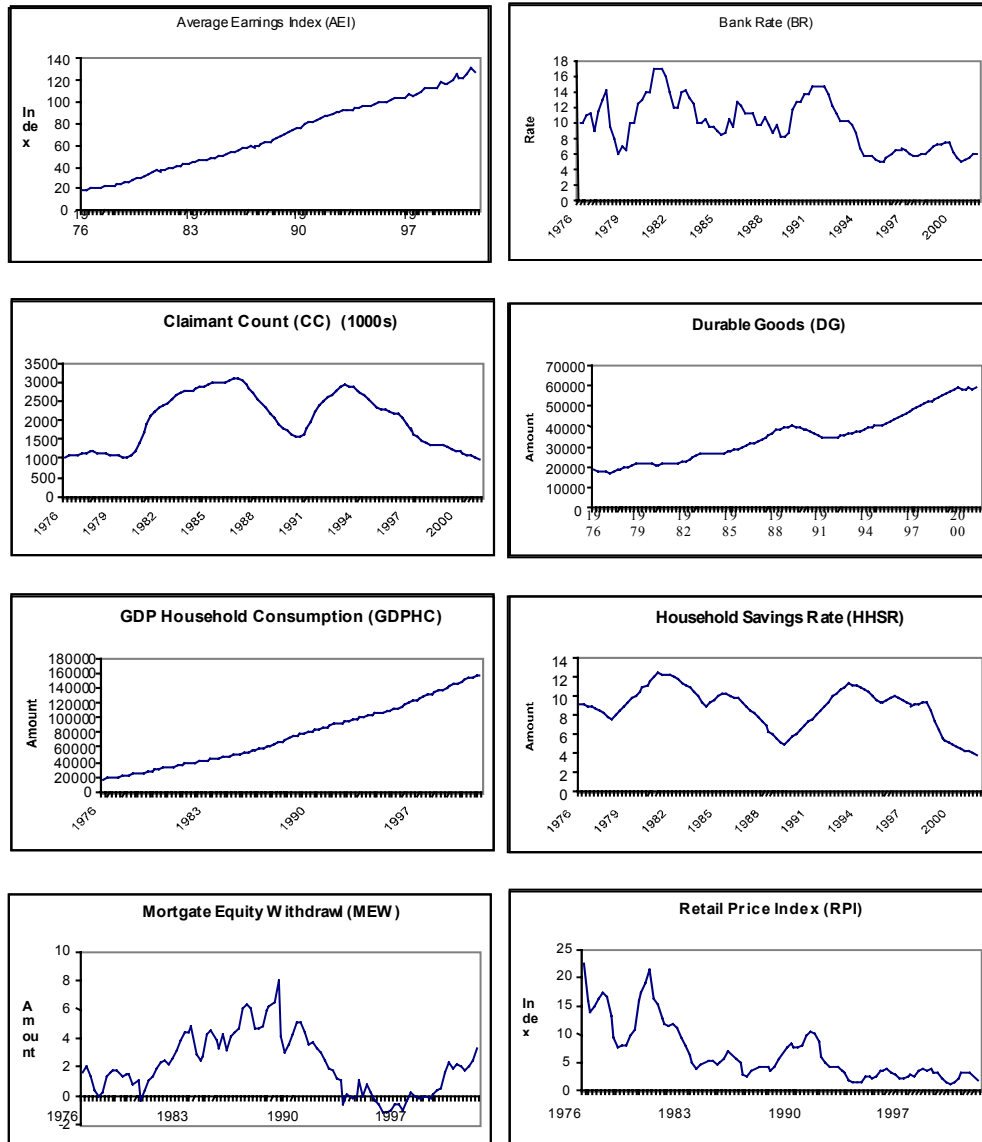


Figure 11 Graphs of the real movement within each economic input series.

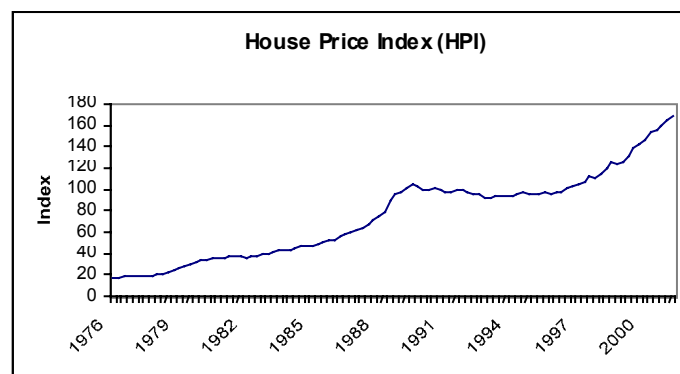


Figure 12 Graph showing the percentage movement in the House Price Index

Appendix B Graphics of the source data's annual percentage movement

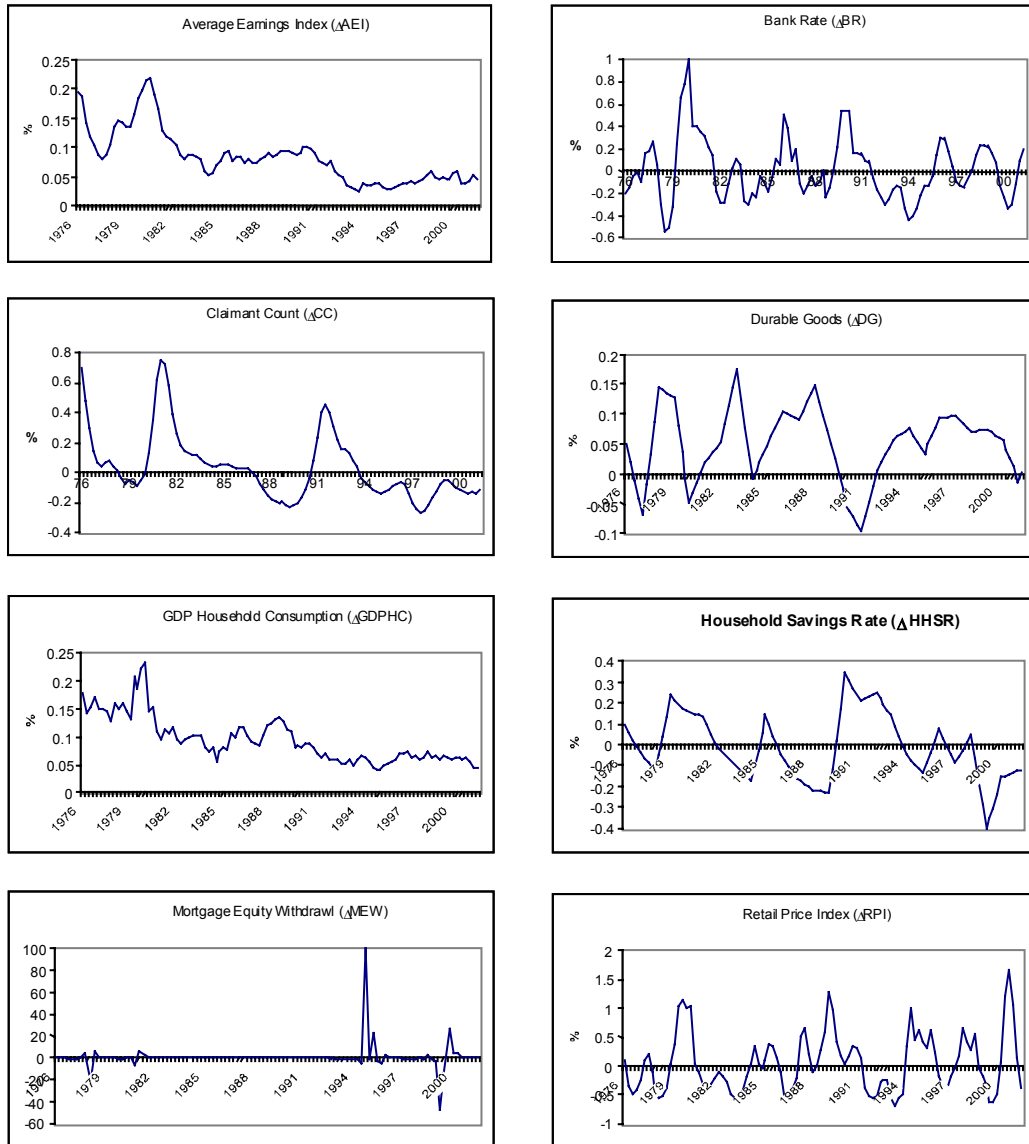


Figure 13 Graphs of the annual percentage movement within each economic input series.

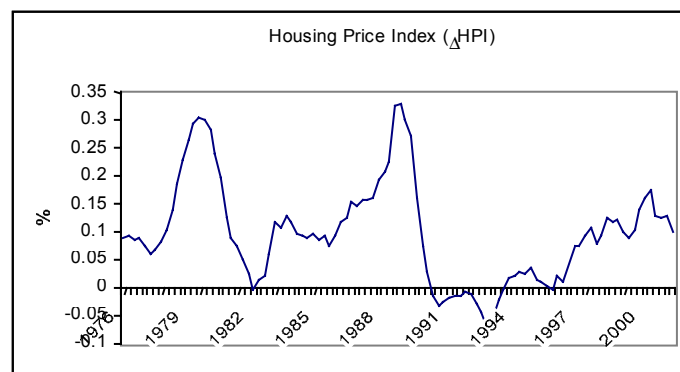


Figure 14 Graph showing the percentage movement in the House Price Index