SPATIOTEMPORAL FORECASTING OF HOME PRICES: A GIS APPLICATION

M.A. Kaboudan

School of Business, University of Redlands,
1200 E. Colton Avenue, Redlands, CA 92373, USA

Abstract: Computational techniques may be useful in modelling and forecasting spatiotemporal data. Statistical challenges that emanate from specification error, aggregation error, measurement error, and perhaps model complexity among other problems encourage employing computational techniques. Genetic programming and neural networks are two such techniques that are robust with respect to autocorrelation, multicollinearity, and stationarity problems statistical and econometric methods encounter. These two computational techniques are employed to demonstrate their potential in producing dynamic forecasts of spatial data. Such forecasts can then help produce sequences of maps of the same geographic region depicting future temporal changes.

Keywords: Economic systems, Genetic algorithms, Nonlinear systems.

1. INTRODUCTION

Techniques to analyse, model, and forecast spatiotemporal series are far from being established. Over the past few decades, *spatial statistics* advanced far more than *spatial econometrics* and *spatial forecasting*. *Spatial statistics* offer measures of global spatial autocorrelation like the Moran I and Geary's c and of local spatial autocorrelation like G and G* (Getis and Ord, 1992). Geographic Information Systems (GIS) provides a mean for collecting, storing, and analysing data associated with geographic regions. However, temporal forecasting of spatial data remains a problem unresolved. Traditional econometric techniques (such as regression or maximum likelihood methods) are of little help because analysis of spatial data quickly faces a problem of spatial autocorrelation. Model misspecification and spatial heterogeneity are other problems that hinder progress. These are aggravated by other statistical complications such as measurement error, non-stationary data, and aggregation problems. Anselin *et al.* (1996) discuss spatial autocorrelation and other statistical complications encountered when analysing or modelling spatial data. When spatial data is taken over time, perhaps the presence of low dimensional nonlinear or chaotic dynamics complicates the matter further. This issue was addressed by and by Maros-Nikolaus and Martin-González (2002). Nonlinear dynamics are more difficult to model, and forecast errors or residuals tend to increase rapidly over time due to sensitivity to initial conditions.

Given that statistical problems hinder modelling and forecasting efforts when dealing with spatiotemporal data, using modelling techniques that circumvent statistical estimation of model parameters may be helpful. Two computational techniques – genetic programming and artificial neural networks – emerge as reasonable alternatives. Genetic programming (GP) produces model specifications that may be capable of forecasting spatiotemporal series. GP is a stochastic search optimisation technique based on the Darwinian survival of the fittest notion. It was popularised by John Koza (1992). One of its useful applications is its ability to deliver regression-type models. Traditional statistical calculations to estimate model coefficients and restrictions formal statistical models impose are totally absent. When properly coded, GP can

assemble large numbers of equations in search for the fittest one. Each equation is assembled by randomly combing variables, random numbers, and operators. The computer algorithm then identifies that fittest equation. Models GP produces are typically nonlinear and univariate, very difficult to interpret, but forecast rather well. A brief review of the technique is in the next Section of this paper. Neural networks (NN) are a computerized classification technique that also delivers forecasts but without delivering a model. NN architecture is based on the human neural system. It is programmed to go into a training iterative process designed to learn the dynamics of a system. NN is a more acceptable and established technique with superior power in fitting complex dynamic data and has gained attention. Gopal and Fischer (1996) and Cubiles-de-la-Vega1 et al. (2003) used NN in spatial forecasting. Both GP and NN are robust with respect to many statistical problems standard econometric or statistical modelling methods have. More specifically they are robust against problems of multicollinearity, autocorrelation, non-stationarity, and specification errors. Use of these two techniques is warranted since they forecast well. They are applied below to discrete time series of multiple geographical regions collected over a number of years.

To produce forecasts of a variable Y using explanatory or independent variables, both dependent and independent variables' values must be defined and obtained. Variables' associated with space or geographic regions will be identified by i, where i = 1, …, n. Values collected at equally spaced time intervals will be identified by t, where t = 1, …, T. The objective here is to find that model for the spatial univariate time series Y_{it}, where Y_{it} is a *Kx1* vector with *K* = n*T. GP is expected to deliver a single equation model that captures variations across geographic regions over time. The following is a general hypothetical specification of such model:

$$Y_{it} = f(S_{t,i}, X_{i,t}, Z_{it}) \quad (1)$$

where $S_{t,i}$ is a set of spatial variables that vary across regions but not over time, $X_{i,t}$ is a set of time series variables that vary over time but remain constant across regions, and $z_{i,t}$ are variables that vary over both. NN is expected to accurately reproduce spatial values of Y over the training period then forecast their values for a few periods into the future.

Delivering spatiotemporal forecasts is important. It helps produce dynamic geographic maps with future changes captured in successive images for decision making purposes. The manner in which GP and NN is used here was never attempted before. GP and NN modelling using the exact same data set and forecasting the same number of periods ahead provide limited comparison to reach preliminary conclusions about the appropriateness of these techniques and their abilities. However, it is important to demonstrate their potential their comparison is thus reasonable and should help evaluate their relative performances.

This paper introduces a different way to forecast spatiotemporal phenomena. What is presented here is of exploratory nature. While real data is used for demonstration below, model specification variables included are far short of pertinent ones. Using a complete set of possible variables was not feasible at the time this research was conducted. Only data available freely via the web was employed. In spite of such limitation, results reported below seem promising. This demonstration starts in the next Section with a brief explanation of GP and a review of NN how they can be used in forecasting. Hypothetical specification of the univariate model and the data used to obtain a GP best fit equation and NN structures used are described in Section 3. Forecast results using GP and NN are compared in Section 4. The final Section contains the conclusion.

2. GP and NN

2.1 Genetic Programming:

Avoid leaving a heading at the bottom of a column, with the subsequent text starting at the top of the next page/column. Use extra spacings (between earlier figures or sections) to push the heading up to the top of the same column as its text. In view of the tight page constraints, however, do please make the fullest possible use of the text area.

GP is utilized here to evolve model specifications useful in forecasting. A description of how GP is used in forecasting and its statistical properties are in Kaboudan (2001). TSGP (for Time Series Genetic Programming, Kaboudan, 2003) is the software used to evolve models with. TSGP is written for Windows environment in C++. It uses two types of input: data input files and a configuration file. Data values of the dependent and each of the independent variables must be supplied in separate files. The configuration file contains execution information such as: name of the dependent variable, number of observations to fit, number of observations to forecast, number of equation specifications to evolve, and other GP-specific parameters. To obtain a best-fit equation, the GP computer program starts by randomly assembling an initial population of equations. The user determines the size of such population. The user provides data input files of the variables and selects from a set of mathematical operators such as +, -, *, protected /, protected $\sqrt{\ }$, sin, cos, as well as others the program uses to assemble equations with. Protected division and square root are necessary to prevent division by zero and taking the square root of a negative number. These protections follow standards most GP researchers agree upon and are designed to avoid computational-problems. More specifically, these protections are programmed as follows:

1. If in (x÷y), y = 0, then (x÷y) = 1.
2. If in $y^{1/2}$, y < 0, then $y^{1/2} = -|y|^{1/2}$.
3. If in ln(y), y ≤ 0, then ln(y) = 1, where ln is the natural logarithm.
4. If in exp(y), y > 10, then exp(y) = exp(10).

Using a random number generator, the program randomly selects variables and operators to assemble equations members of a population. Once assembled, their respective fitness (typically MSE) is computed, where

$$MSE = \sum(Actual_i - Fitted_i)^2 / n \quad (2)$$

and where n is the sample used to obtain fitted values. That equation in a population with the lowest MSE is considered fittest. If a population contains an equation that replicates the values of the dependent variable accurately is found the program terminates. Level of accuracy is user-controlled threshold minimum MSE (e.g. Min (MSE) = 0.0001). If GP does not deliver an equation with the Min (MSE), which is most of the time, the program proceeds to breed a new population. Succeeding populations are the outcome of a programmed breeding mechanism. Self-reproduction, crossover, and mutation are used to breed new members. In self-reproduction, the best equations in an existing population are simply copied into the new one. In crossover, randomly selected sections from two (usually fitter) equations from an existing population are exchanged to breed two offspring. In mutation, a randomly selected section from a randomly selected equation from an existing population is replaced by newly assembled part(s) to breed an individual member of the new population. Thusly, GP continues to breed new generations until an equation that satisfies Min (MSE) set is found or a preset maximum number of generations is reached. The best equation in the last population bred is then used to produce fitted as well as forecast values. Parameters to evolve GP models are typically set to the following: Population size = 1000, number of generations = 100, self-reproduction rate = 20%, crossover rate = 20%, mutation rate = 60%, and number of best-fit equations to evolve in a single run = 100. Evolving 100 equations is necessary because executing the program only once is not sufficient. Assembling equations in GP is random and the fittest equation is one that has global minimum MSE. Unfortunately, GP software typically gets easily trapped at a local minimum rather than global MSE.

TSGP produces two types of output files. One has a final model specification and the other contains actual and that model's fitted values as well as performance statistics such as R^2, MSE, and mean absolute percent error or MAPE, where

$$MAPE = \frac{1}{n}\left(\sum 100 * |(Actual_i - Fitted_i)/Actual_i|\right). \quad (3)$$

GP delivers equations that may reproduce history fairly well. However, even if it succeeds, it does not necessarily forecast well. This problem is not unique to GP. NN suffers from the same type of problem. A best-fit model fails to forecast well when the algorithm used delivers outcomes that are too fit. This phenomenon is known as overfitting. (For more on overfitting, see Lo and MacKinlay, 1999.) To obtain a best forecasting equation, it seems only logical then that an *ex post* forecast by each equation be evaluated first. An *ex post* forecast is one whose dependent variable's outcome is already known but the information was not used in obtaining the model. If the dependent variable's outcome is unknown, the forecast is *ex ante*. It is natural to have more confidence in the *ex ante* forecast if the model producing it also produced an acceptable *ex post* forecast. However, if that model failed to reproduce history, most probably it will not successfully deliver a reliable forecast either. The best forecasting model is therefore identified in two steps. First, fittest equations are sorted according to lowest historical MSE. Those equations with the lowest 10 MSE (where 10 is arbitrarily set) are then sorted according to *ex post* forecast MAPE. That equation among the selected 10 with the lowest forecast MAPE is selected as best to use for *ex ante* forecasting.

2.2 Neural Networks

Neural networks architecture (NN) is used to produce forecasts that will be compared with those obtained using GP. Input data are presented to the network that learns to predict future outcomes. Principe et al. (2000) among many others provide a complete description on how NN can be used in forecasting. There are several network structures to select from when constructing a neural network to use in forecasting. Multilayer perceptrons (MLP) are layered feedforward networks. They are typically trained with static backpropagation. Although they train slowly and require large samples to train with, they are easy to use and approximate well. Generalized feedforward networks (GFF) are a generalization of MLP with connections that jump over one or more layers. They are also trained with static backpropagation. GFF are more efficient in solving problems. MLP and GFF were used to obtain that comparative forecast of the same spatiotemporal data GP produces a model for.

3. MODELING HOME PRICES

3.1 Input Data:

Applying GP or NN to forecast spatiotemporal phenomena demands data accessible mostly using GIS. Obtaining such data was not possible and a suitable set of data was obtained using Internet search instead. Annual median housing prices by neighbourhood published by the City of Cambridge, MA, Community Development Department Community Planning Division (2003) were obtained. The data set is of annual median price of single family homes for the period 1993-2002 of twelve neighbourhoods. Table 1 has a list of the explanatory variables used to obtain models and forecast P_{it} = Real median price of homes sold in neighbourhood i at time period t. Dependent and independent variables used in GP and NN are defined as follows:

$DV_{t,i}$ = Twelve dummy variable that take the value of 1 for neighbourhood i and zero otherwise. These are constant for all years.

PCY_{it-2} = Real per capita income in neighbourhood i at time period t-2. PCY varies by neighbourhood as well as over time.

P_{it-2} = Real median price lagged two periods.

$Y_{i,t-2}$ = Real household median income for the city of Cambridge, MA. This variable is held constant across neighbourhoods but varies with time.

$MR_{i,t-2}$ = Mortgage rate lagged two years. The variable is also held constant across neighbourhoods and varies with time. Mortgage rates were obtained from FRB St. Louis.

$Year_{i,t}$ = 1993, 1994, …, 2002. This variable is held constant across neighbourhoods but varies with time.

Average $P_{i,t-2}$ = Real average median price of homes in the city of Cambridge, MA, lagged two years. This variable is held constant across neighbourhoods but varies with time.

Table 1. Explanatory variables used in modelling.

Spatial Variables	Temporal Variables (Lag = 2)	Spatiotemporal Variables (Lag = 2)
DV1, DV2, DV3, DV4, DV5, DV6, DV7, DV8, DV9, DV10, DV11, DV12	Y, MR, Year, AP	PCY, P

In Table 1, the twelve dummy variables are the only strictly spatial variables used. Including all twelve basically help discriminate between neighbourhoods. Given that multicollinearity is not a problem when using GP or NN, all twelve are used. Their use demonstrates robustness of GP and NN against multicollinearity. Per capita income was not available by neighbourhood but by census tract.

Spatial autocorrelation was tested using the following OLS regression model:

$$P_{it} = \alpha + \rho\ P_{(i-1)t} \qquad (4)$$

where ρ measures the degree of autocorrelation. This equation measures autocorrelation between pairs of of contiguous neighbors over time. Autocorrelation is present if the estimated ρ is not significantly different from zero. The resulting equation using data 1995-2002 is:

$$P_{it} = 138.9 + 0.452\ P_{(i-1)t} \qquad (5)$$

where the *p-value* = 0.000 for both the intercept and estimate of ρ. Equation (5) confirms the absence of spatial autocorrelation between pairs of contiguous neighbourhoods averaged over time. All temporal data were lagged two years which were reserved to use as input to forecast unknown outcomes of 2003 and 2004. The reason for using this lag is to obtain predictions of P_{it} without having to forecast any independent variable. The number of data points available for obtaining a model using GP and for training using NN was 69 after losing three observations. They belong to twelve neighbourhoods and represent six years over the period 1995-2000.

3.2 Best-Fit Models:

TSGP was executed to find 100 best-fit equations in 100 searches. The best-fit among fittest equations is:

$$P_{it} = P_{it-2} + 3\ \cos\ (P_{it-2}) + 8\ \cos\ (MR_{t-2,i} + P_{it-2}) + DV_{5,t} - MR_{t-2,i} + Y_{i,t-2} - (Y_{t-2,i} / DV_{11,t}) \qquad (6)$$

This nonlinear equation shows that prices are determined here according to prior prices, mortgage rate, and real household median income. Only neighbourhoods 5 and 11 seem to have an effect on differences in prices.

The two network structures (MLP and GFF) were tested with different configurations. For each network structure hidden layers are varied. Hidden layers tested were set to one, two, and three. Two transfer functions were tested under each scenario, tanhAxon and sigmoidAxon. Given these options, the total number of networks to test thus far is twelve. Each was trained using learning rules with momentum set at 0.7 once then set at 0.9 another. The 24 configurations were tested with a maximum of 1000 epochs. After the best NN structure was identified, the number of epochs was then varied. Maximum epochs were tested at 5000 and 10000 in addition. Variation and control of training epochs help identify networks that succeed in depicting dynamics of training data as well as forecast well. The final model selected had the following structure and run parameters: GFF with two hidden layers; the transfer function was sigmoidAxon with learning momentum set = 0.90; the maximum number of epochs was at 2000. Number of epochs = 2000 was identified as best after determining the best configuration then comparing its results at epochs = 500, 1000, 2000, 3000, 4000, and 5000.

Table 2. Statistics on GP and NN fitted values

	GP	NN
R^2	0.84	0.93
MSE	1949.966	882.30
MAPE	20%	12.3%
Residuals:		
Mean	-0.73	-1.26
Standard Error of	5.35	3.60
Median	-4.03	-3.15
Kurtosis	-0.44	0.02
Skewness	0.28	0.35
Minimum	-89.72	-57.04
Maximum	109.28	83.97

The best NN configuration produced a better fit of P_{it} training values than the selected GP model. Table 2 contains comparative statistics on fitted values GP and NN delivered. As the table suggests, the GP equation explained 84% of the variation in prices (R^2 = 0.84) while NN explained 93% of the variation (R^2 = 0.93). Neither series of residuals is normally distributed. (A data set is approximately normally distributed if its mean is equal to its median and if the coefficients of skewness and kurtosis are approximately equal to zero.) One would tend to believe at this point that NN will produce the better

out-of-sample (2001 and 2002) forecasts. This is not the case as demonstrated next.

4. FORECASTING

Although NN delivered a better fit in reproducing data used in training (1995-2000), forecasts by the GP model were better than those by NN. Table 3 contains a comparison of the forecast statistics.

Table 3. Forecast comparison

	GP	NN
Theil's U	0.08	0.097
MAPE	9.80	19.940
MSE	2623.27	4272.921
NMSE	0.14	0.224

The Theil's U-statistic reported in the table is a measure of forecast performance. It is known as Theil's inequality coefficient and is defined as:

$$U = \frac{\sqrt{MSE}}{\sqrt{k^{-1}\sum_{j=1}^{k} P_{ik}^2} + \sqrt{k^{-1}\sum_{j=1}^{k} \dot{P}_{ik}^2}} \qquad (7)$$

where j = 1, 2, …, k (with k = 23 observations representing 2001 and 2002 forecasted), and $\dot{P}_{ik}$ are forecast values of P_{ik}. This statistic will always fall between zero and one where zero indicates a perfect fit (Pindyck and Rubinfeld, 1998, p. 387). Figure1 and provide a comparison suggesting the better performance of GP's forecast. The shorter series are actual values while the longer ones are the forecasts.

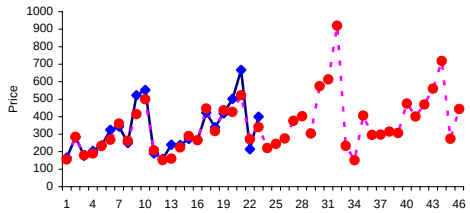

Figure 1. Actaul and GP foreast prices.

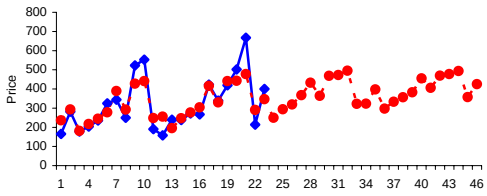

Figure 2. Actual and NN forecast of prices

5. CONCLUSION

This paper contained an experimental exercise suggesting that combining spatial and temporal data to obtain forecast using computational techniques is feasible. Data on single family home prices were used to test whether employing genetic programming and neural networks would help deliver forecasts of the spatiotemporal phenomena. Explanatory variables used contained perfect collinearity, spatial autocorrelation, and measurement error. Because both GP and NN have robustness against many statistical problems, it was possible to obtain forecasts of prices across geographical neighbourhoods and over time. NN fitted price values were better than GP's. The better out-of-sample *ex post* forecast was delivered by GP. GP's better forecast suggests that it's *ex ante* forecast may be more reliable as well.

REFERENCES

Anselin, L., A. Bera, R. Florax, and M.Yoon (1996). Simple diagnostic tests for spatial dependence,. *Regional Science and Urban Economics*, 26, 77-104.

Cambridge Community Development Department Community Planning Division (2003) Cambridge demographic, socioeconomic & real estate Market information, http://www.ci.cambridge.ma.us/~CDD/data/index.html.

Cubiles-de-la-Vega1, M., R. Pino-Mejías, A. Muñoz-Reyes and A. Pascual-Acosta (2003). Locally weighting spatial data for neural network forecasting. http://isi-eh.usc.es/resumenes/42_46_abstract.pdf.

FRB St. Louis (2003). http://research.stlouisfed.org/fred2/series/MORTG/downloaddata.

Getis, A. and K. Ord (1992). The analysis of spatial autocorrelation by use of distance statistics. *Geographical Analysis*, 24, 189-206.

Gopal, S. and M. Fischer (1996). Learning in single hidden layer feedforward neural network models: Backpropagation in a spatial Interaction modelling context. *Geographical Analysis*, 28, 38-55.

Kaboudan, M. (2001). Statistical properties of fitted residuals from genetically evolved models. *Journal of Economic Dynamics and Control*, 25, 1719-1749.

Kaboudan, M. (2003). TSGP: A time series genetic programming software. http://newton.uor.edu/facultyfolder/mahmoud_kaboudan/tsgp.

Koza, J. (1992). *Genetic Programming*, Cambridge, MA: The MIT Press.

Lo, A. and C. MacKinlay (1999). *A Non-Random Walk Down Wall Street*, Princeton, NJ: Princeton University Press.

Maros-Nikolaus, P. and J. Martin-González, (2002). Spatial forecasting: detecting determinism from single snapshots. *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, 12, 369-376.

Pindyck, R. & D. Rubinfeld (1998). *Econometric Models and Economic Forecasting*. Boston: Irwin McGraw-Hill.

Principe, J., N. Euliano and C. Lefebvre (2000). *Neural and Adaptive Systems: Fundamentals Through Simulations*, New York: John Wiley & Sons, Inc.