Computational Forecasting of Two Exchange Rates

Mak Kaboudan[1]

School of Business, University of Redlands

1200 East Colton Avenue, Redlands, CA 92373, USA

Abstract

In this paper, genetic programming and artificial neural networks are employed to forecast two different exchange rates, US dollar/Japanese Yen and US dollar/Taiwan dollar. Extended forecasts (that go beyond one-step-ahead) obtained using the computational techniques were compared with naïve random walk predictions of the two exchange rates. Sixteen-step-ahead forecasts obtained using genetic programming outperformed the one- and sixteen-step-ahead random walk US dollar/Taiwan dollar exchange rate predictions. Further, sixteen-step-ahead forecasts of the wavelet-transformed US dollar/Japanese Yen exchange rate also using genetic programming outperformed the sixteen-step-ahead random walk predictions of the exchange rate.

Keywords: Genetic programming; Artificial neural networks; Wavelet transformations.

1. Introduction

Forecasting exchange rates is important but remains a challenge. Until 1996, work in [1-3] found no evidence to support the hypothesis that nonlinear generating processes outperform random walk behaviour. Recently, applications of computational algorithms document that artificial neural network [4] and nearest neighbour [5] approaches may deliver more accurate than no-change (random walk) forecasts of exchange rates. Although computational techniques delivered forecasts that are better than random walk predictions, these forecasts are only one-step-ahead. This means that the exchange rate today is a necessary input to forecast that of tomorrow. The objective of this paper is to obtain extended forecasts that are hopefully more useful for investment and profitable decision making. Models obtained and forecasting techniques used to predict the exchange rates in this paper therefore deliver forecasts that are 32 days ahead. The first 16 (of the 32) are used to validate forecast efficacy. If forecasting known outcomes is success, confidence in forecasting unknown outcomes using the same model or technique for decision making increases. This means that a model that succeeds in reproducing historical data (i.e., data used in fitting or training) best should not be used to forecast future outcomes unless it also happens to be the most successful one in producing validation out-of-sample forecasts.

This paper introduces use of wavelet-transformed exchange rates input data when applying genetic programming and neural networks. What is presented here is of exploratory nature. While the observed exchange rate data is used for demonstration below, use of wavelet-transformed data is examined. Results reported below seem promising. This demonstration starts in the next Section where the data used is introduced. Section 3 contains a brief review of genetic programming and artificial neural networks. It also contains an explanation of how these two techniques are used to fit and train the wavelet-transformed sets. The results are in Section 4. Section 5 has conclusions.

2. The Data

Data on the US dollar/Japanese Yen (DJY) and US dollar/Taiwan dollar (DTD) are available at Oanda.com [6]. All employed data is for exactly the same period of time that starts December 12, 2003 and ends September 8, 2004. Of the 272 observations, 256 were used in fitting and training models and 16 observations (August 24 – September 8, 2004) were used for validation. Out-of-sample forecasts were then for 16 more days (September 9 – September 24, 2004). Before fitting any models, predictability of each of the observed exchange rates to be used in training was estimated using a GP predictability measure [7]. The measure statistic is η. It approximates the probability that an observed data set is predictable. The test statistic is:

$$\eta = \max\left\{0, \frac{1}{n}\sum_{i=1}^{n}\left(1-\frac{MSE_Y}{MSE_S}\right)_i\right\} \qquad (1)$$

where MSE = mean of squared errors (typically used as fitness function), i is the number of MSE values to compare and average with i = 1, …, n runs such that n ≥ 30, Y = observed series, and S = randomly shuffled sequence of the observed Y. The original sequence of series Y is randomly shuffled using Efron's bootstrap method [8]. If the series Y contains a predictable pattern, randomly shuffling the sequence of its observed values will dismember such pattern. If the series Y contains no predictable pattern, randomly shuffling the order of its values will not have any impact. Therefore, if Y were predictable, $\eta \to 0$. If it were totally random $\eta \to 1$. The statistic provides suggestions about relative predictability given that $0 \le \eta \le 1$. When applied to the two exchange rate series, the results were as follows:

$$\eta_{DJY} = 20.4\%$$

$$\eta_{DTD} = 60.3\%.$$

These results suggest that GP will probably be successful in predicting DTD but not DJY.

Six forecasts per exchange rate are obtained below. GP fits observed data and its wavelet-transformed series to provide two forecasts: GP and W-GP. Similarly, ANN trains the observed and wavelet-transformed series to provide two other forecasts: ANN and W-ANN. These four forecasts are then compared with two benchmark random walk forecasts RW1 (one-step-ahead) and RW16 (16-step-ahead).

In wavelet analysis, data can be converted into forms to which it may be easier to fit models. Applications using wavelets in estimation were reviewed in [9]. In and by itself, wavelet analysis is not a forecasting technique. A *wavelet transformation* is a form of data conversion. (See [10] for other forms.) A wavelet transform is a function used to transform a signal into father and mother wavelets. Father wavelets are representations of a signal's smooth or low-frequency component. Mother wavelets are representations of the details or high-frequency component in a signal. The Haar wavelet is the simplest to use. It is a decimated process where at each level of scaling half the number of observations disappears. When used, first it transforms a series Y_t to obtain mid-point averages (s_1) and mid-point differences (d_1) of consecutive pairs of observations (and not pairs of consecutive points). Averages preserve the main signal while differences capture the series' detailed fluctuations. In turn, mid-point averages (s_1) are transformed to obtain their mid-point averages (s_2) and their mid-point differences (d_2), and so on. Obtaining these values is known as a *discrete wavelet transform* process (DWT). Alternatively, DWT maps a vector of Y_t values to a vector of wavelet coefficients w, or

$$w = \begin{pmatrix} s_J \\ d_J \\ d_{J-1} \\ \dots \\ d_1 \end{pmatrix} \qquad (2)$$

where J is the number of scales or multiresolution components, and

$$\begin{aligned} s_J &= (s_{J,1}, s_{J,2}, \dots, s_{J,T/2^J})' \\ d_J &= (d_{J,1}, d_{J,2}, \dots, d_{J,T/2^J})' \\ d_{J-1} &= (d_{J,1,1}, d_{J-1,2}, \dots, d_{J-1,T/2^J})' \quad (3) \\ \dots &= \dots. \\ d_1 &= (d_{1,1}, d_{1,2}, \dots, d_{1,T/2^J})'. \end{aligned}$$

To transform a series to a maximum J components, that series must be of length $T = 2^J$. For a Haar wavelet:

averages: $s_J = (s_{J-1,2t} + s_{J-1,2t-1})/2$ (4)

differences: $d_{J,t} = (s_{J-1,2t} - s_{J-1,2t-1})/2$ (5)

The Haar DWT wavelet has a desirable property that may help in forecasting. Given a series' wavelet transformed coefficients s_J and d_J, …, d_1 in (3) above, original values of that series can be reconstructed from the transformed data. If a series has T = 256, DWT with J = 3 (for example) delivers four series: s_3 with 32 coefficients, as well as d_3, d_2, and d_1 with 32, 64, and 128 coefficients in each, respectively. For each of the four data sets (s_3, d_3, d_2, and d_1) different models can then be obtained. Basically, the idea of using these four series amounts to adopting a "divide and conquer strategy". Figures 1 and 2 show the DWT (s_3, d_3, d_2, and d_1) and IDWT (the reconstructed sequence) of DJY and DTD with scaling level J = 3.

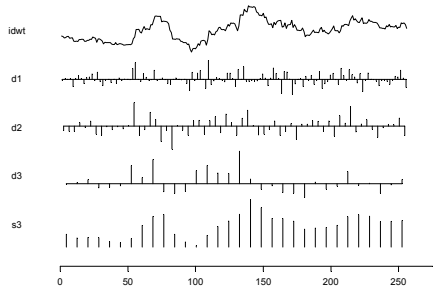

Fig. 1. Plots of DWT (s_3, d_3, d_2, & d_1) and inverse (idwt) of DJY - the reconstruction of the originally observed data.

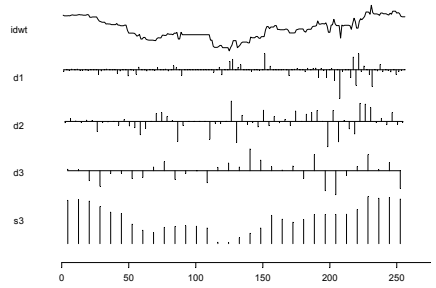

Fig. 2. Plots of DWT (s_3, d_3, d_2, & d_1) and inverse (idwt) of DTD - the reconstruction of the originally observed data.

Each component of the transformed sequences has a different level of complexity. Series s_3 basically represents the clearest signal in the observed values of the variable, and therefore is usually the least complex or most predictable. Series d_3 captures what may be viewed as systematic changes in the signal, and therefore may be fairly predictable as well. Series d_2 should be more complex because it captures changes in the averages. Noise is captured mostly in series d_1. For DJY, η = 0.64, 0.14, 0.00, and 0.00 for series s_3, d_3, d_2, and d_1, respectively. For DTD, η = 0.83, 0.04, 0.47, and 0.19 for series s_3, d_3, d_2, and d_1, respectively.

3. Methodology

The basic exchange rate equation is specified as a delayed autoregressive model or

$$Y_t = f(Y_{t-L}, Y_{t-L-1}, \dots, Y_{t-2L}). \qquad (6)$$

where L = number of lags and t = 1,…,T. Observations f = T+1, …, T+F then identify the *ex post* predictions to validate the efficacy of the forecast. In addition to training the observed data, GP fits four additional models and ANN trains four additional networks, one for each of the four DWT series. Given that each of the transformations has a different level of complexity, each should have a model with a unique specification. The model specification assumed for s_3 is:

$$s_{3,t} = f(s_{3,t-2}, s_{3,t-3}, s_{3,t-4}, s_{3,t-5}, s_{3,t-6}). \qquad (7)$$

According to (7), s_3 is assumed to be a function of five distant (starting with the second lag) but consecutive lagged values. Similarly,

$$d_{3,t} = f(d_{3,t-2}, \ldots, d_{3,t-6}); \quad (8)$$

$$d_{2,t} = f(d_{2,t-4}, \ldots, d_{2,t-10}); \quad (9)$$

$$d_{1,t} = f(d_{1,t-8}, \ldots, d_{1,t-16}). \quad (10)$$

3.1. Genetic Programming

Foundations of GP are in [11] and description of how GP is used in forecasting and its statistical properties are in [12]. The GP software used in this study is TSGP [13]. It takes two types of input files: data files and a configuration file. Data input files contain values of the dependent and each of the independent variables. 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 are in the configuration file. TSGP is designed to randomly assemble an initial population of 1000 specifications, computes their fitness, and then breeds 1000 equations as members of a new generation. New equations are bred using mutation, crossover, and self reproduction. In mutation, TSGP randomly assemble a sub-tree that replaces a randomly selected existing part of a tree. In crossover, randomly selected parts of two existing trees are swapped. In self reproduction, a top percentage of the fittest individuals in an existing population are passed on to the next generation. After completing a specified number of generations, the program terminates and saves to an output file the specification that captures the dynamics of a series best. The best equation is then used to forecast that series' future values. Finally, actual and fitted values as well as residuals and standard evaluation statistics (R^2, MSE, and *ex post* prediction MSE) are written to a different output file.

GP-resulting specifications may be viewed as coincidental equations that may capture the dynamics of a process. As such, GP is useful only as a planning tool and is not helpful in policy-making or conducting 'what if' type of analysis. Coefficients in the equations are not computed. They are random numbers (between -128 and 127). This gives GP an advantage over conventional statistical regressions. Given that there are no coefficients to compute, GP is robust against problems of multicollinearity, autocorrelation, and heteroscedasticity. There are also no degrees of freedom lost to compute coefficients.

3.2. Artificial Neural Networks

ANN is a well-established information-processing paradigm based on the way the densely interconnected parallel structure of the human brain processes information. The technique can be used to detect structure in time-series. A network is a collection of mathematical models that emulates the nervous systems and draws on the analogies of adaptive learning. Input data is presented to the network that learns to predict future outcomes. Complete description of types of networks to choose from and an explanation of how ANN can be used in forecasting are abundant. (See [14] for more details.) In time series forecasting, and similar to GP, a standard autoregressive model structure where future values of a series are determined from its own past observations is also assumed.

Basically, two network configurations – multilayer perceptions (MLP) and generalized feedforward (GFF) – were used in training models to forecast exchange rates. MLP is a layered feedforward network where the first layer contains the inputs consisting of the lagged dependent variables. The last layer delivers output or problem solution. The middle layers (which may be one or more) are hidden and contain weights and node biases estimated during a network training process. MLP usually produces good approximations. It is typically trained with static backpropagation. GFF is a generalization of MLP with connections that can jump over one or more layers. Generally, artificial neural networks are well known for their ability to replicate complex dynamics fairly well.

4. Results

Although out-of-sample forecasts are the focus of this section, in-sample results must be discussed first. GP's run parameters were as follows: population = 1000, mutation rate = 0.6, crossover rate = 0.3, selection method = roulette, maximum number of generations = 120, fitness measure = MSE, and operators = +, -, * %, sin, & cos where % is a standard protected division.

Three statistics are used to compare all outcomes. They are: The means absolute percent error (MAPE), MSE, and Theil's U-statistic. Their definitions follow:

$$\text{MAPE} = \frac{100}{F}\sum_{f=1}^{F}\left|\left(X_f - \hat{X}_f\right)/X_f\right| \quad (11)$$

where X_f = exchange rate observed values to forecast *ex post*, $\hat{X}_f$ = their forecasted values, and f = 1, …, F periods of *ex post* forecast.

$$\text{MSE} = \frac{1}{F}\sum_{f=1}^{F}\left(X_f - \hat{X}_f\right)^2, \quad (12)$$

$$U_1 = \frac{\text{RMSPE}_M}{\text{RMSPE}_{RW1}} \quad (13)$$

where M identifies the technique used and RW1 = one-day-ahead random walk prediction, and

$$U_{16} = \frac{\text{RMSPE}_M}{\text{RMSPE}_{RW16}} \quad (14)$$

where RW16 = sixteen-day-ahead random walk prediction. A $U_1 < 1$ implies that M's forecast outperforms that of RW1. A $U_{16} < 1$ implies that M's forecast outperforms that of RW16.

Table 1 contains a summary of the results using GP. *Ex post* and *ex ante* forecast statistics are much better than GP's fitting statistics, regardless of whether wavelets were used or not. (It is possible to report the *ex ante* forecast here because actual values that were not known when this experiment started became known by the time this paper was written.) Forecasts of DJY were better than forecasts of DTD when GP was applied to the wavelet-transformed data. These forecasts were

better than either one- or sixteen-step ahead random walk predictions.

Table 1. GP and W-GP fitting and forecast statistics.

Table 1. GP and W-GP fitting and forecast statistics.

	GP		W-GP	
	DJY	DTD	DJY	DTD
Fitting:				
MAPE	1.45	0.55	0.9	0.71
MSE	4.1	0.06	1.79	0.08
Ex post:				
MAPE	0.87	0.18	0.34	0.52
MSE	1.06	0.01	0.23	0.04
Theil's U_1	10.04	0.55	4.72	1.14
Theil's U_{16}	1.65	0.52	0.77	1.07
Ex ante:				
MAPE	0.49	0.24	0.29	0.59
MSE	0.46	0.01	0.15	0.05
Theil's U_1	1.76	1	1.02	2.26
Theil's U_{16}	1.21	0.54	0.7	1.23

Table 2 contains a summary of the results using ANN. *Ex post* and *ex ante* forecast statistics are much worse than ANN's training statistics, regardless of whether wavelets were used or not. Forecasts of DTD were better than forecasts of DJY when ANN was applied to the observed data. These forecasts were better than either one- or sixteen-step ahead random walk predictions.

Table 2. ANN and W-ANN fitting and forecast statistics.

	ANN		W-ANN	
	DJY	DTD	DJY	DTD
Fitting:				
MAPE	1.18	0.25	0.16	0.65
MSE	2.75	0.01	0.05	0.07
Ex post:				
MAPE	0.56	0.63	0.94	0.76
MSE	0.77	0.06	1.54	0.09
Theil's U_1	8.57	1.51	12.13	1.84
Theil's U_{16}	1.41	1.43	1.99	1.74
Ex ante:				
MAPE	0.34	0.66	0.54	0.75
MSE	0.32	0.06	0.49	0.08
Theil's U_1	1.49	2.45	1.81	3.01
Theil's U_{16}	1.02	1.33	1.24	1.64

5. Conclusion

Obtaining *extended forecasts* that go beyond one-step-ahead was shown to be feasible in this paper. Sixteen-step-ahead forecasts were obtained for two exchange rates (US dollar/Japanese yen and US dollar/Taiwan dollar) using different techniques. The benefit from that is obvious. *Extended forecasts* are more useful in decision making than one-step-ahead forecasts. Second, the notion of fitting and training different temporal resolutions of an observed set of data was introduced. Different temporal resolutions were obtained using the Haar wavelet transform. It decomposes observed values into low frequency signals and high frequency variations. The Haar wavelet was selected because it is a discrete decimation process that through inversion delivers perfect reconstruction of the signal.

Six forecasts of each exchange rate were obtained using genetic programming, artificial neural networks, one-step-ahead random walk modelling, and finally using sixteen-step-ahead random walk modelling. Genetic programming forecasted the US dollar/Taiwan dollar best. The use of wavelet-transformed data helped only in forecasting US dollar/Japanese yen. Genetic programming forecasts using the transformed data outperformed the sixteen-step-ahead random walk predictions.

6. References

[1] R., Baillie, and T. and Bollerslev, "The Message in Daily Exchange Rates: A Conditional Variance Tale," *Journal of Business and Economic Statistics* 7, pp. 297-306, 1989.

[2] D. Hsieh, "Modelling Heteroskedasticity in Daily Foreign Exchange Rates," *Journal of Business and Economic Statistics* 7, 307-317, 1989.

[3] A. Cecen, and C. Erkal, "Distinguishing between stochastic and deterministic behaviour in high frequency foreign exchange returns: can non-linear dynamic help forecasting?" *International Journal of Forecasting* 12, pp. 465-473, 1996.

[4] G. Zang, and V. Berardi, "Time Series Forecasting with Neural Network Ensembles: An Application for Exchange Rate Prediction," *Journal of Operations Research Society* 52, pp. 652-664, 2001.

[5] F. Fernandez-Rodriguez, S. Sosvilla-Rivero, and J. Andrada-Felix, "Nearest-Neighbour Predictions in Foreign Exchange Markets," in *Computational Intelligence in Economics and Finance*, eds. Shu-Heng Chen and Paul Wang, Springer-Verlag, Berlin, pp. 297-325, 2004.

[6] http://www.oanda.com.

[7] S-H. Chen, and T-W. Kuo, "Discovering Hidden Patterns with Genetic Programming," in *Computational Intelligence in Economics and Finance*, eds. Shu-Heng Chen and Paul Wang, Springer-Verlag, Berlin, pp. 329-347, 2004.

[8] B. Efron, *The Jackknife, the Bootstrap, and Other Resampling Plans*. Society for Industrial and Applied Mathematics, Philadelphia, 1982.

[9] G. Lee, "Wavelets and Wavelet Estimation: A Review," *Journal of Economic Theory and Econometrics* 4, pp. 123-157, 1998.

[10] J. Walker, *A Primer on Wavelets and Their Scientific Applications*, Chapman & Hall/CRC, Boca Raton, 1999.

[11] J. Koza, *Genetic programming*. The MIT Press, Cambridge, MA, 1992.

[12] M. Kaboudan, "Statistical Properties of Fitted Residuals from Genetically Evolved Models," *Journal of Economic Dynamics and Control* 25, pp. 1719-1749, 2001.

[13] http:// bulldog2.Redlands.edu /fac/mak_kaboudan/tsgp

[14] J. Principe, N. Euliano, and C. Lefebvre, *Neural and Adaptive Systems: Fundamentals through Simulations* John Wiley & Sons, Inc, New York, 2000.