Currency Exchange Rate Forecasting using Associative Models

Omar Shatagua Jurado-Sánchez[1], Cornelio Yáñez-Márquez[1], Oscar Camacho-Nieto[2], and Itzamá López-Yáñez[2].

[1] Instituto Politécnico Nacional – Centro de Investigación en Computación,
Mexico City, Mexico
shatagua@live.com.mx, cyanez@cic.ipn.mx
[2] Instituto Politécnico Nacional – Centro de Inovación y Desarrollo Tecnológico en Cómputo,
Mexico City, Mexico
{ocamacho, ilopezy}@ipn.mx

Abstract. Associative Models were created and used for pattern recognition tasks, but recently such models have shown good forecasting capabilities; by a pre-processing of a time series and some fit of the Model. In this paper, the Gamma Classifier is used as a novel alternative for currency exchange rate forecasting, where experimental results indicate that the proposed method can be effective in the Exchange Rate Time Series Prediction, compared to classical Machine Learning Models (ANN, SVM, MLP) and well known for the Financial and Economy Fields Box-Jenkins Models (AR, ARMA, ARIMA).

Keywords: Associative models, Gamma classifier, forecasting, exchange rate, time series, ARIMA, ARMA.

1 Introduction

The exchange rate forecasting is one of the most difficult and important tasks in modern study of time series prediction. Several factors influence in the value of a currency, such as oil prices, the rise or fall of imports of goods and services, inflation, consumer price index, interest rate, among others [1]. There exists a debate about what are the real factors that are affecting the currency value [2], this kind of problem with many variables, most of the times unknown, is usually treated as a stochastic and univariable problem [3, 4].

The Exchange rate as a critical issue of Financial Time Series Study has led into several forecasting models development for accurate and timely decision making, by the Economy, Financial and Statistical fields. For the analysis of financial time series, from economy perspective, the regression techniques are widely used and popular for their statistical properties [6]. Some of these models are AR, Ma, ARMA, and the more general ARIMA [4 -6].

pp. 67-76; rec. 2014-05-16; acc. 2014-07-07

On the other hand, Computer Science has various techniques for time series analysis and forecasting, some of these techniques are:

- Artificial Neural Networks [7].
- Support Vector Machines [8].
- Support Vector Regression [9].
- Fuzzy Logic [10].
- Associative Models [11, 18].

Another approach that has been rising for the financial time series forecasting is the combination of different models; this concept is based on the idea of improving the weaknesses of certain models with the strengths of others. These are called hybrid models and are the most prolific approach for financial time series forecasting in the actual literature. In 2010 Huang, Chuang, Wu and Lai create a Chaos-based support vector regressions model for Exchange rate forecasting [12], Khashei, Bijari and Ardali work on financial time series forecasting using an ARIMA with probabilistic neural networks [13] model, in 2013 Kazem, Sharifi, Khadeer, Morteza, and Khadeer used a Support vector regression with bio-inspired algorithm [14], in 2014 Wei, Cheng, and Wu, present a hybrid ANFIS (Adaptive Network-Based Fuzzy Inference System) based on n-period moving average model [15], also in 2014 Gharleghi, Shaariy and Shafighi did exchange rates forecasting using a Cointegration Based Neuro-Fuzzy System [16] and Minakhi, Majhi Babita, Majhi Ritanjali and Panda created a forecasting model for currency exchange rates using an adaptive ARMA model with differential evolution based training [17].

These sophisticated hybrid models are inherently difficult to implement, in addition to being computationally expensive. In this paper we study a simple and computationally efficient way to forecast currency exchange rates using an Associative Model, given the competitive results that the Associative Model Gamma Classifier has shown on previous works of Time Series Forecasting [18].

2 Gamma Classifier

As mentioned before, we use the Associative Model Gamma Classifier (GC). This work is strongly based on [18], however, the steps of the Gamma Classifier for exchange rate forecasting are a modification of the Algorithm shown in [18], described in section 3. For the forecasting of several points we used the separation method taken directly from [18].

The Gamma Classifier Algorithm for the forecasting task uses 3 important definitions, mentioned below; also it must do a pre-processing of the time series data, a codification of the time series, so the GC can treat the forecasting problem as a classification problem. This codification is named Modified Johnson-Möbius (MJM) see [18] for specific codification algorithm.

Definition 1 (Alpha and Beta operators). *Given the sets A = {0, 1} and B = {0, 1, 2}, the alpha (α) and beta (β) operators are defined in a tabular form as shown in table 1. The corresponding vector versions of both operators for inputs* $x \in A^n$*,* $y \in A^n$*, and* $z \in B^n$ *give an n-dimensional vector as output, whose i-th component is computed as follows.*

$$\alpha(x,y)_i = \alpha(x_i,y_i) \text{ and } \beta(z,y)_i = \beta(z_i,y_i)$$

Table 1. Alfa and Beta Operators definition

$\alpha : A \times A \rightarrow B$		
x	*y*	*α(x,y)*
0	0	1
0	1	0
1	0	2
1	1	1

$\beta : B \times A \rightarrow A$		
x	*y*	*β(x,y)*
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

Definition 2 (u_β operator). *Considering the binary pattern* $x \in A^n$ *as input, this unary operator gives the following integer as output.*

$$u_\beta(x) = \sum_{i=1}^{n} \beta(x_i, x_i) \qquad (1)$$

Definition 3 (Gamma operator). *The similarity Gamma operator takes two binary patterns —*$x \in A^n$ *and* $y \in A^m$*;* $n,m \in Z^+$ $n \le m$ *— and a non-negative integer θ as input, and outputs a binary number, according to the following rule.*

$$\gamma_g(x,y,\Theta) = \begin{cases} 1 & if\ m - u_\beta[\alpha(x,y)\ mod\ 2] \le \Theta \\ & 0\ otherwise \end{cases} \qquad (2)$$

The Gamma Classifier Algorithm for Time Series Forecasting is described below:

1. Convert the patterns in the fundamental set into binary vectors using the MJM code.
2. Code the test pattern with the MJM code, using the same parameters used for the fundamental set.

3. Compute the stop parameter $\boldsymbol{\rho} = \bigwedge_{j=1}^{n} \bigvee_{i=1}^{p} x_j^i$

4. Transform the index of all fundamental patterns into two indices, one for their class and another for their position in the class (e.g. x^{μ} in class i becomes $x^{i\omega}$).

5. Initialize θ = 0

6. Do $\gamma(x_j^{i\omega}, y_j, \Theta)$ for each component of the fundamental patterns.

7. Compute a weighted sum c_i for each class, according to this equation:

$$c_i = \frac{\sum_{\omega=1}^{k_i} \sum_{j=1}^{n} \gamma(x_j^{i\omega}, y_j, \theta)}{k_i}, \tag{3}$$

where k_i is the cardinality in the fundamental set of class *i*.

8. If there is more than one maximum among the different ci, increment θ by 1 and repeat steps 6 and 7 until there is a unique maximum, or the stop condition $\theta \geq \rho$ is fulfilled.

9. If there is a unique maximum among the c_i, assign ỹ to the class corresponding to such maximum.

10. Otherwise, assign ỹ to the class of the first maxima found.

Definition 4 (Separation). *Given a TS D with samples $d_1d_2d_3$ . . ., the separation s between a segment $d_i\ d_{i+1}$ . . . d_{n-1} (of length n) and sample d_j is given by the distance between the closest extreme of the segment and the sample.*

Based on this definition and the GC, the proposed TS forecasting method follows the algorithm presented below, considering a TS D of length l with a prediction (test) segment of length t, and a length for the patterns of size *n*.

1. Starting from the TS D, calculate the differences between successive samples in order to work with values relative to that of the previous sample. The new time series D' has length l − 1.

$$D_{lx1} \rightarrow D'_{(l-1)x1} \tag{4}$$

2. Build a set of associations between TS difference segments of length n and its corresponding difference with separation *s*, for positive values of s = 1, 2, . . . , t − 1, t. Thus, there will be a sets of associations of the form $\{a^{\mu}, b^{\mu}\}$ where $a \in R^n$ and $b \in R$, and the i-th association of the set is made up by $a^{\mu} = d_i d_{i+1} \ldots d_{i+n-1}$ and $b\mu = di+n+s-1$.

3. Train a different GC from each association; there will be t different classifiers, each with a distinct fundamental set $\{x^{\mu}, y^{\mu}\}$ for its corresponding value of s.

4. Operate each GC with all the input segments a^{μ}.

5. When multiple classifiers give different output values ỹ for the same data point in the differences TS D, there are two prominent alternatives to integrate them into one value ỹ.

   (a) Average the values given by the two classifiers with the same absolute separation |s| = {−s, s}; this is denoted as the combined method.

   (b) Average the values given by all available classifiers; this is known as the combined average method.

6. Convert back to absolute values by adding the forecast relative value (ỹ) to the original value of the previous sample, taken from D.

$$\tilde{y} = D_{i-1} + \tilde{y}_i^{'} \quad (5)$$

3 Proposed Model

The Gamma Classifier Algorithm for Exchange Rate Forecasting is described below:
1. Convert the patterns in the fundamental set into binary vectors using the MJM code.
2. Code the test pattern with the MJM code, using the same parameters used for the fundamental set.
3. Compute the stop parameter $\boldsymbol{p} = \wedge_{\boldsymbol{j=1}}^{\boldsymbol{n}} \vee_{\boldsymbol{i=1}}^{\boldsymbol{p}} \boldsymbol{x_j^i}$
4. Transform the index of all fundamental patterns into two indices, one for their class and another for their position in the class (e.g. x^{μ} in class i becomes $x^{i\omega}$).
5. Initialize θ,

$$\theta = (0.005) * \frac{1}{p}\sum_{i=1}^{p} x^i \quad (6)$$

6. Do $\gamma(x_j^{i\omega}, y_j, \Theta)$ for each component of the fundamental patterns.
7. Compute a weighted sum c_i for each class, according to equation (3).
8. If there is more than one maximum among the different c_i, calculate the average of all C_i classes that has a c_i maximum and assign the average divided by 10^d (to get the original scale of the time series S) to the unknown pattern ỹ.

$$average = \frac{1}{amount\ of\ maxima}\sum \mathrm{C_j}\ where\ j = i\ \forall\ i\ \ such\ that\ c_j = \bigvee\nolimits_{i=1}^{k} c_i \quad (7)$$

$$C_{\tilde{y}} = \frac{average}{10^z} \quad (8)$$

9. If there is a unique maximum among the c_i, increment θ by 1 and repeat steps 6 and 7 until there is more than one maximum, or the stop condition $\theta \geq \rho$ is fulfilled.
10. Otherwise, assign ỹ to the class of the first maxima found.

Opposed to the Gamma Classifier that seeks for a unique maximum for the class assignation for the unknown pattern ỹ, the proposed model seeks for several alike classes. The idea is to find similar classes (time series patterns) that have occurred in the past, and assign an average of those alike classes to the unknown pattern **ỹ.**

To do this the conventional initialization of Θ in 0 is not appropriate, because if we do this it takes more iterations to find similar patterns; also, on a low value of Θ, the lower amount of similar classes we will have (could be only two). Experimental results have shown that the best results are given when we have more than 2 similar classes, but we should be careful of not giving a high value of Θ because this could lead the classifier to determine that classes are similar, and they are not, because, while ϴ increases, the similarity between the features of the unknown pattern and the known patterns decreases.

For ϴ initialization purpose we propose the equation 3.Experimental results also have shown that low size of n, (size of the pattern codification) between 4 and 6, have better performance that high values of n (10 or more).

4 Experiments Design

4.1 Data Sets

The time series used in these experiments were taken from 2 web sites, daily values of MXN/USD (Mexican Peso/American Dollar) and JPN/USD (Japanese Yen/USD) were taken from www.forecasts.org, and USD/GBP (USD/British Pound) monthly values, were taken from http://www.ny.frb.org/markets/fxrates/historical/home.cfm (Federal Reserve Bank of New York site), for MXN/USD we used the interval from 2010-12-01 - 2013-12-20, for JPN/USD we used from 2010-01-04 - 2012-12-31 and finally for USD/GBP we used from 1971-01-01 to 2013-11-01.

We create a set of experiments where we compare our proposed model against the classical forecasting models, under the very same circumstances; we take a whole time series and hold out the last 23 values (approximately the working days in a month) for the daily values, and hold out the last 12 values for the monthly values, then we use the time series (without the last 23 or 12 values) as an input of the models and forecast the next 23 and 12 values for of the series for the daily and monthly time series respectively, then we compare the results using the last 23 and 12 values held out before and calculating the errors.

4.2 Error Metrics

The error metrics used for the models comparison are Mean Square Error (MSE), Root Mean Square Error (RMSE) and Mean Absolute Percent Error (MAPE), also we calculate the Pearson's Correlation Coefficient (PCC) to see the behavior of the forecasted result. For MSE, RMSE and MAPE lower values are better, meanwhile for PCC closest values to 1 are better. The metrics equations are given as follows:

$$MSE = \frac{\sum e_t^2}{n}, \tag{9}$$

$$RMSE = \sqrt{MSE}, \tag{10}$$

$$MAPE = \frac{\sum \frac{e_t}{a_t}}{n}(100), \tag{11}$$

$$PCC = \frac{\sigma_{xy}}{\sigma_x \sigma_y}, \tag{12}$$

where:

e_t– mean error on time t, where: $e_t = a_t - f_t$.
a_t – is the actual value.
f_t – is the forecasted value, and
n – The amount of samples.
σ_{xy} – is the covariance of (x,y)
σ_x – is the standard deviation of x
σ_y – is the standard deviation of y

4.3 Models Construction

We used 3 tools for the models construction, for the regression models (AR, ARIMA, ARMA) we used Matlab 2013 which includes an Econometric toolbox that allows to create an ARIMA(p,D,q) model, and an Estimate function that uses maximum likelihood to estimate the parameters of the ARIMA(p,D,q) model given the observed univariate time series. For the Machine Learning Algorithms we used Weka 3.7.11 that now includes a forecast utility. The model configuration used for Multi-Layer Perceptron (MLP), Support Vector Regression, MLP Regressor are the default suggested by Weka Optimization class. Finally we implemented in Java Language the Gamma Classifier.

5 Experimental Results

5.1 Models Comparison

The models are ordered from lower to higher RMSE. The tables 2-4 show the performance of the models implemented for this work. The Gamma Classifier with Combined (C) method and Combined Average (CA) method are highlighted.

For GBP-USD predictions (Table 2), the Gamma Classifier with CA method has the second best performance and also it have the best PCC.

Table 2. GBP-USD results for 12 months ahead forecasted

Model	**Correlation Coefficient**	**MSE**	**RMSE**	**MAPE**
ARMA(10,3)	0.549237	0.000202	0.014223	0.515375
Gamma Classifier CA	**0.703234**	**0.000211**	**0.014516**	**0.037408**
SVRegression	0.562688	0.000301	0.017362	0.585531
AR 10	0.573760	0.000305	0.017465	0.636467
Least Median Squared Linear Regression	-0.414414	0.000567	0.023810	0.835851
Gamma Classifier C	**0.296301**	**0.000701**	**0.026479**	**0.993409**
MLP Regressor	-0.092631	0.000818	0.028593	1.042937
ARIMA(10,1,10)	-0.028813	0.000997	0.031568	1.174118
Gaussian Processes	0.562810	0.083368	0.288736	11.376421
Multi Layer Perceptron	0.607254	0.096973	0.311405	9.630076

For JNP-USD predictions (Table 3), the Gamma Classifier with C method has the third best performance; meanwhile the Gamma Classifier with CA method has the fourth best PCC.

Table 3. JNP-USD results for 23 days ahead forecasted

Model	**Correlation Coefficient**	**MSE**	**RMSE**	**MAPE**
Multi-Layer Perceptron	0.979428	0.468221	0.684267	0.701716
SVRegression	0.966534	1.904561	1.380058	1.089557
Gamma Classifier C	**0.580982**	**3.273381**	**1.809249**	**1.554930**
ARMA(10,3)	-0.719281	3.958358	1.989562	1.629965
Least Median Squared Regression	-0.732313	4.588599	2.142101	1.809652
MLP Regressor	-0.954679	4.707212	2.169611	1.768684
ARIMA(10,1,10)	-0.850739	4.802789	2.191527	1.849221
AR 10	-0.750468	4.861758	2.204939	1.887653
Gamma Classifier CA	**0.903806**	**5.372789**	**2.317928**	**2.446975**
Linear Regression	0.983064	18.519081	4.303380	3.947253

Finally, for MXN-USD predictions (Table 4), the Gamma Classifier with CA method has the third best performance and the third best PCC.

Table 4. MXN-USD results for 23 days ahead forecasted

Model	Correlation Coefficient	MSE	RMSE	MAPE
Least Median Squared Regression	-0.126938	0.011632	0.107851	0.657486
SVRegression	-0.291778	0.012311	0.110954	0.625846
Gamma Classifier CA	**0.180517**	**0.012805**	**0.113159**	**0.690101**
Linear Regression	-0.377727	0.012867	0.113432	0.661473
MLP Regressor	-0.347532	0.013522	0.116283	0.729020
ARIMA(10,1,10)	-0.074195	0.019713	0.140405	0.808322
AR 10	0.125089	0.024416	0.156257	0.944695
ARMA(10,3)	0.381415	0.025612	0.160037	0.996850
Multi-Layer Perceptron	0.271096	0.035570	0.188601	1.183675
Gamma Classifier C	**-0.466466**	**0.049240**	**0.221902**	**1.372825**

6 Conclusions and Future Work

The Performance of the Gamma Classifier is a promising model for the forecast exchange rate problem, it is simple for understanding, implementing and is computationally efficient. However, one open problem is find a way to decide which method for the result of the Gamma Classifier should be used, combined or combined average, because where Combined method does not have the best of the performances the Combined Average improve the results and conversely.

Empirical results have shown that ranges between 4 and 6 for the size of the pattern *n* have better performance. Other open problem is, find a way to determinate the optimal initial values of *n* and *Θ* for this kind of problem. This algorithm can be extended to other forecasting financial problems, because other financial problems present .similar features that exchange rate problem also does. Another approach for this forecasting task. is that the Gamma Classifier could be considered to be part of a hybrid model for the forecasting problem, given that, sometimes GC has better performance than classical financial or machine learning approaches.

References

1. Girón, Alicia: Conceptos básicos de economía. Instituto de Investigaciones Económicas. Editorial UNAM. UNAM. México (1994)
2. Huang, Shian-Chang, Chuang, Pei-Ju, Wu, Cheng-Feng, Lai Hiuen-Jiun: Chaos-based support vector regressions for exchange rate forecasting. Expert Systems with Applications. (2010) 8590–8598.
3. Yu, Lean, Wang, Shouyang, Lai, Kin Keung: A neural-network-based nonlinear metamodeling approach to financial time series forecasting. Applied Soft Computing. (2008) 563–574.
4. Box, P. Jenkins, G.M., Reinsel, G.C.: Time Series Analysis:Forecastingand Control. 4thed. Englewood Cliffs. NJ: Prentice-Hall (2008)
5. Rout, Minakhi, Majhi, Babita, Majhi, Ritanjali, Panda, Ganapati: Forecasting of currency exchange rates using an adaptive ARMA model with differential evolution based training. In: Journal of King Saud University. (2014) 7 -18
6. Khashei, Mehdi, Bijari, Mehdi: A new class of hybrid models for time series forecasting. Expert Systems with Applications yyg. (2012) 4344–4357
7. Adetunji Philip, Adewole, AdioTaofiki, Akinwale, Ayo Bidem , Akintomide: Artificial Neural Network Model for Forecasting Foreign Exchange Rate. In: World of Computer Science and Information Technology Journal. (2011) 110-118.
8. Kim, Kyoung-jae: Financial time series forecasting using support vector machines. Neurocomputing. (2003) 307 – 319
9. Li, Zhang, Wei-Da, Zhou, Pei-Chann, Chang, Ji-Wen, Yang, Fan-Zhang, Li: Iterated time series prediction with multiple support vector regression models. Neurocomputing, (2011) 411–422
10.Aznarte, José Luis, Alcalá-Fdez, Jesús, Arauzo-Azofra, Antonio, Benítez, José Manuel: Financial time series forecasting with a bio-inspired fuzzy model. Expert Systems with Applications. (2012) 12302–12309
11. López-Yañez, I., Arguelles-Cruz, A. J., Camacho-Nieto, O.,Yañez-Marquez, C.: Polluants Time-Series Prediction Using the Gamma Classifier. International Journal of Computational Intelligence Systems. (2011) 680 - 711
12. Huang, Shian-Chang, Chuang, Pei-Ju, Wu, Cheng-Feng, Lai Hiuen-Jiun: Chaos-based support vector regressions for exchange rate forecasting. Expert Systems with Applications. (2010) 8590–8598
13. Khashei, Mehdi, Bijari, Mehdi, Ardali, Gholam: Hybridization of autoregressive integrated moving average (ARIMA) with probabilistic neural networks (PNNs). Computers & Industrial Engineering. (2012) 37–45
14 Kazem, Ahmad, Sharifi, Ebrahim, Khadeer, Farookh, Morteza, Saberi, Khadeer, Omar: Support vector regression with chaos-based firefly algorithm for stock market price forecasting. Applied Soft Computing. (2013) 947–958
15. Wei, Liang-Ying, Cheng, Ching-Hsue, Wu, Hsin-Hung: A hybrid ANFIS based on n-period moving average model to forecast TAIEX stock. Applied Soft Computing. (2014) 86–92.
16. Gharleghi, Behrooz, Shaari, Abu Hassan, Shafighi, Najla. Predicting exchange rates using a novel "cointegration based neuro-fuzzy system", International Economics. (2014) 88–103
17. Yáñez-Márquez, C., López-Yáñez, I., Sáenz-Morales, G. L.: Analysis and prediction of air quality data with the gamma classifier, Progress in Pattern Recognition. Image Analysis and Applications (2008)
18. Sheremetov, Leonid, González-Sánchez, Arturo,López-Yáñez,Itzamá, Ponomarev, A novel associative model for time series data mining. Pattern Recognition Lettersl. (2013) 23-33