DUKASCOPY
RESEARCH PRODUCTS

22/04/2014

# Economic Research

Tuesday, April 22, 2014

# Forecasting Models for Exchange Rate

*The science of prognostics has been going through a rapid and fruitful development in the past decades, with various forecasting methods, procedures and approaches flooding the economic world. It is estimated that there are more than 100 prediction methods, and sometimes the diversity makes it difficult to choose the one that would do the trick. In our new research we try and compare a few of the most popular techniques, and see if they are in fact suitable for forecasting currency exchange rates.*

We look at five models, starting with the most naïve methods and moving up to relatively complex systems, with their own paradigms and special way of thinking. We test the models on EUR/USD currency pair in the time frame from 1/1/2014 to 3/24/2014.

## Methodology

### Moving Average

Simple moving average (*SMA*) is a naive forecasting method. It uses the most recent observations to forecast future values. The basic *SMA* model assumption is that the time series is stationary, and the model used for forecasting is $y_t = \mu + \varepsilon_t$, where $\mu$ is a constant and $\varepsilon_t$ is a random variable with mean 0 and variation $\sigma^2$.

The only parameter in the model, $\mu$, is estimated as the average value of the last *k* observations at every time moment *t*:

$$\hat{\mu}_t = \frac{(y_t + y_{t-1} + \cdots y_{t-k})}{k}$$

The forecasted value of *y* is then the same as the estimate of the $\hat{y}_{t+1} = \hat{\mu}_t$ parameter:

*Advantages and disadvantages:*

The main advantage of this model is its simplicity and the speed of implementation. But the primality of the model affects the quality of its results – it is not able to foresee the peaks and troughs of the series. It can also prove to be moderately useless if the time series is nonlinear or non-stationary.

### Exponential smoothing

Simple exponential smoothing (*EMA—exponential moving average*) is the modification of a weighted moving average. It has the same time series model, $y_t = \mu + \varepsilon_t$, but uses a different approximation. Namely, the single parameter $\mu$ is estimated as the weighted average of the last observation and the previous estimate:

$$\hat{\mu}_t = \alpha y_t + (1 - \alpha)\,\hat{\mu}_{t-1}$$

Here $\alpha$ is a smoothing constant in the interval between 0 and 1. The constant $\alpha$ controls the closeness of the interpolated (smoothed) value to the most recent observation, $y_t$. The forecast value is equal to the parameter estimate: $\hat{y}_{t+1} = \hat{\mu}_t$

*Advantages and disadvantages:*

The properties of the *SES* are similar to the ones of *SMA* - it also works only with linear and stationary time series and inevitably lags behind a trend.

There are a lot of other exponential smoothing methods and moving average modifications, but all of them share these main drawbacks.

Both methods mentioned above can only be used for stationary time series. For the data we use the *ADF* test (see Appendix) does not reject the null hypothesis of non-stationary, meaning that more sophisticated models are needed to work with real-life exchange rates.

## ARIMA

An autoregressive integrated moving average (*ARIMA*) model is the generalization of both *SES* and *SMA*. The most widely known special cases of *ARIMA* models are autoregressive model and random walk and random trend models. *ARIMA(p,d,q)* is a notation used for the general model, with parameters *p*, *d* and *q* describing autoregressive, integrated and moving average components, respectively.

ARIMA(p, d, q):

$$\alpha(L)(1 - L)^d y_t = \alpha_0 + \beta(L)\varepsilon_t$$

where $\alpha(L) = 1 - a_1 L - \ldots - a_p L^p$ is an autoregressive polynomial, and $\beta(L) = 1 + b_1 L + \ldots + b_q L^q$ is a moving average polynomial. *L* is a lag operator:

$$L x_t = x_{t-1}$$

*Advantages and disadvantages:*

*ARIMA* model serves a wide spectrum of time series and may be used for both stationary and non-stationary series. Unlike many other complex models, *ARIMA* gives the prediction directly from model estimations and does not require any additional calculations. It does, however, have its practical disadvantages, too - it is sensitive to outliers and requires a large number of observations.

## Dynamic Linear Model

Dynamic Linear Models (*DLM*), also called Linear-Gaussian state space models, are a particular group of state space models - a wide class of models for time series analysis. The main idea of *DLM* is to express a time series through a Markov process (represented by a so called state vector) and additive white noise.

The modeling starts with the assumption that at the initial time moment the state vector is normally distributed $\theta_0 \sim N(m_0, C_0)$ and all consequent time moments are described by the set of equations:

$$y_t = F_t\theta_t + v_t, \quad v_t \sim N(0, V_t)$$
$$\theta_t = G_t\theta_{t-1} + w_t, \quad w_t \sim N(0, W_t)$$

where $\theta_t$ is the unobservable value of the state vector at time *t*, and $y_t$ is the observable time series vector.

It is therefore assumed that the evolution of the unobservable states of the system is described by operator $G_t$, but the $F_t$ operator transforms the model states into observations. The whole analysis is thus aimed at estimating the $F_t$ and $G_t$ operators and covariance matrices $V_t$ and $W_t$.

*Advantages and disadvantages:*

The key advantage of the *DLM* is its flexibility. As all system parameters are recalculated at every step, the model adjusts quickly to any changes in system. Thus it allows to capture the main features of a wide variety of different time series. The advantages, however, come at the cost of relative complexity of the algorithm. Moreover, the method forces quite heavy restrictions on the underlying processes, such as the normal distribution of parameters and independence of model errors.

## Artificial neural networks

The artificial neural network (*ANN*) method is very different from other techniques of data processing. *ANN* paradigms are inspired by the biological neural system, and its main goal is to simulate the way a human brain works. The reason of the popularity of *ANN*s is its ability to approximate almost any nonlinear function. *ANN*s are used not only in time series analysis, but also in data comparison, clustering analysis, classification problems, and control theory.

The smallest unit of *ANN*, just like in a biological neural network, is a neuron.

The *ANN* algorithm is divided into two parts: training and testing. The first stage is training, when neuron weights are calculated and adjusted to desired precision. Then, at the testing stage, the goodness of weights is tested on real data.

We chose to test a two-layer *ANN* with construction 3-2-1, meaning that there are three neurons on the first layer, two on the second (hidden) layer, and one at the output.

*Advantages and disadvantages:*

The *ANN* has a more complex structure than any of the methods mentioned above. This quality makes the method harder to implement, but it is also accountable for the main benefit of the *ANN*. The network's structure is parallel, so if one element of the *ANN* fails to produce an adequate outcome, it has little effect on the work of other neurons, and the process can continue without any problems. It also enables us to use the *ANN* while working with time series of any structure. The biggest weakness of the model, similarly to the case of *ARIMA*, is that it requires a large number of observations to properly train the network.

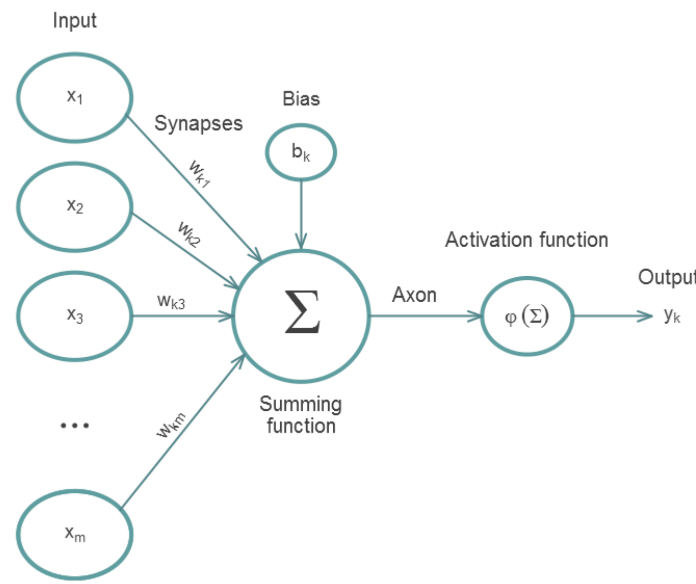


***Figure 1: Nonlinear model of a neuron***

In mathematical terms, neuron *k* can be described by the following equations:

$$u_k = \sum_{j=1}^{m} x_j w_{kj}$$

and

$$y_k = \varphi\left(u_k + b_k\right)$$

Where $x_1, x_2, \ldots, x_m$ are the input signals; $w_{k1}, w_{k2}, \ldots, w_{km}$ are the synaptic weights of neuron *k*; $b_k$ is the bias; $u_k$ — summing function, and $y_k$ is the output of the neuron. The activation function φ controls the *ANN* learning process, i.e., the adjustment of the synaptic weights.

## Results

The 5-day window *SMA* model gives the worst results in our analysis. The model captures the general direction of the series and the main peaks and troughs, but is not sensitive to small changes. We can thus say that the result is too smoothed. Which is hardly surprising since the forecast is just the average of the previous values. The exponential smoothing, on the other hand, gives much better results. The *EMA* curve not only moves in the right direction, but also shows even the small motions of the price. In this case a lot depends on the choice of the constant $\alpha$. We use $\alpha$*=0.8*, therefore putting the largest weight on the last observation. Visually, however, it is obvious that the best results come from *ARIMA(1,1,1)*, *DLM*, and *ANN(3-2-1)* - their predictions practically coincide (see Figure 4).

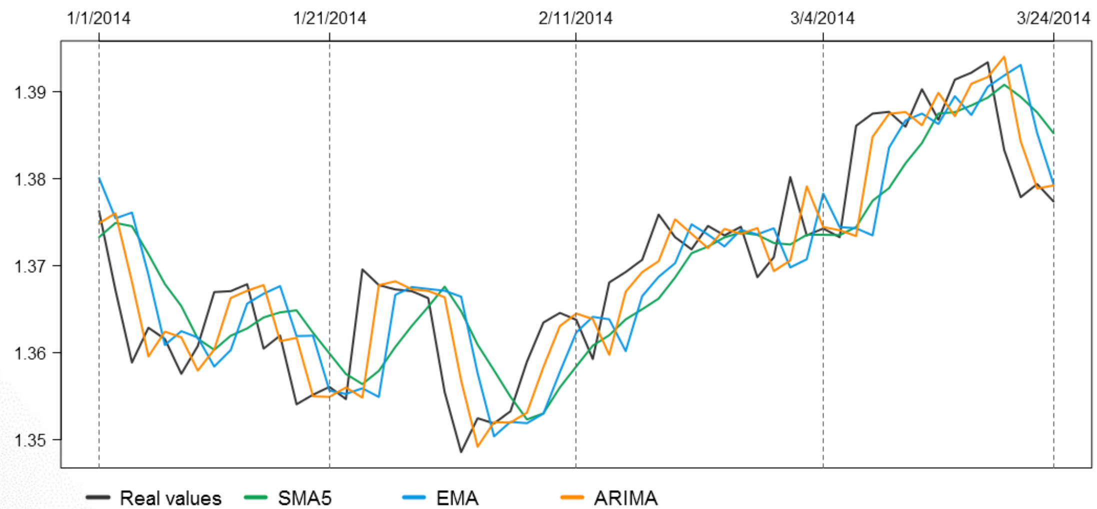


***Figure 2: The one-step-ahead forecast results by the SMA5, EMA and ARIMA(1, 1, 1).***

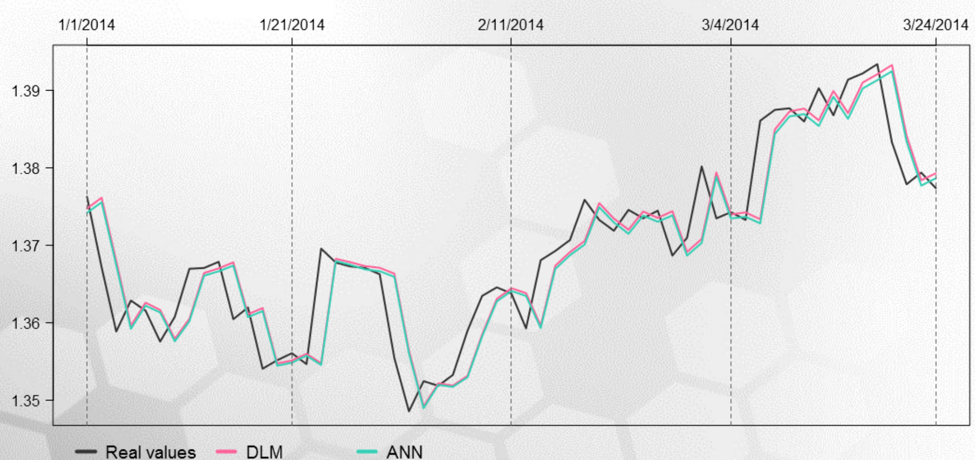


***Figure 3: The one-step-ahead forecast results by the DLM and ANN (3-2-1).***

All forecasting curves are lagged behind the real values. One of the reasons for this is the strong autocorrelation in the series, but the main cause of the forecast delay is the huge weight of the previous observation in the models. All mentioned techniques increasingly depend on the last value of the time series, with *SMA* and *ANN* being the only exceptions. *SMA5* puts equal weights on all five last observations, while *ANN* fits weights values for the three previous data points. As this forecasting bias seems to be rather considerable, it is natural to wonder whether it is altogether worth it to take trouble over the models. Thus we compare our results to the ones we would obtain by simply assuming the present value of the prediction to be equal to the observation in the previous time moment.

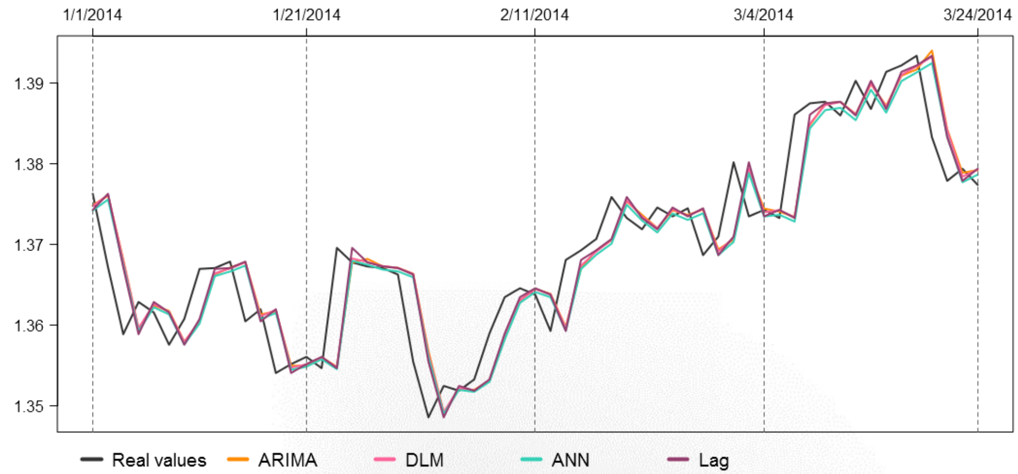


***Figure 4: The one-step-ahead forecast of the SMA5, EMA, ARIMA, DLM, ANN, and the lagged series.***

We can see that the series of the lagged values is very similar to the best forecasting models - *ARIMA(1,1,1)*, *DLM*, and *ANN(3-2-1)*. To make the comparison numerical and thus more precise we use the error measures - *MSE*, *MAE*, and *MAPE* (see Appendix).

| | SMA5 | EMA | ARIMA | DLM | ANN | Lag |
|---|---|---|---|---|---|---|
| MSE | $4.99 \times 10^{-5}$ | $4.94 \times 10^{-5}$ | $2.58 \times 10^{-5}$ | $2.55 \times 10^{-5}$ | $2.55 \times 10^{-5}$ | $2.54 \times 10^{-5}$ |
| MAE | 0.0059 | 0.0054 | 0.00365 | 0.00365 | 0.00372 | 0.0037 |
| MAPE | 0.431 | 0.394 | 0.267 | 0.267 | 0.271 | 0.27 |

***Table 1: MSE, MAE and MAPE results***

| SMA5 | EMA | ARIMA | DLM | ANN |
|---|---|---|---|---|
| 0.01 | 0.1 | 0.51 | 4.93 | 0.36 |

***Table 2: SMA5, EMA, ARIMA DLA and ANN algorithms working time (sec.)***

The *ARIMA* model has the smallest error values, with DLM, lagged series and ANN scoring almost insignificantly worse.

Another important characteristic of any algorithm is its working time, and here *SMA5* proves to be the fastest. However, it is *ARIMA(1,1,1)* model that takes the first place in the overall standings.

## Conclusion

We have studied five forecasting models with different construction techniques and levels of difficulty. In total, all received results followed the general motion of the real series, some showing slightly more precision than others. Notably, the forecast given by the classical *ARIMA* model was even better than the results of more complex *DLM* and *ANN*. As *ARIMA* is also easier and faster to employ, the superior results make it unquestionably the best forecasting method in our study. However, *SMA* and *EMA* models are also very simple in both understanding and realisation, with *SMA* giving the most smooth curves, but *EMA* being more dynamic. So if the high precision of the forecast is not the main target of analysis, both these methods can also be used.

It is noteworthy, however, that all the models give lagged, or delayed, predictions, and do not offer much superiority over a simple lagged series. Such outcome is dictated by the nature of time series – it is non-stationary and strongly autocorrelated. Therefore, in cases when pointwise and indefinably accurate predictions would suffice, it might be more sensible to save time and effort, and use the lagged series.

# Appendix

Mean squared error:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Mean absolute error:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Mean absolute percentage error:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i}$$

Augmented Dickey-Fuller (*ADF*) test models:

$$1)\Delta y_t = \gamma y_{t-1} + \sum_{i=1}^{p}\beta_i \Delta y_{t-i} + \varepsilon_t$$

$$2)\Delta y_t = a_0 + \gamma y_{t-1} + \sum_{i=1}^{p}\beta_i \Delta y_{t-i} + \varepsilon_t$$

$$2)\Delta y_t = a_0 + \gamma y_{t-1} + \sum_{i=1}^{p}\beta_i \Delta y_{t-i} + \varepsilon_t$$

$$H_0 : \gamma = 0 \; (y \text{ is a random walk})$$
$$H_{alt} : -2 < \gamma < 0 \; (y \text{ is stationary})$$












**Newest releases and archive:**

- Fundamental Analysis
- Technical Analysis
- Press Review
- Market Research
- Expert Commentary
- Dukascopy Sentiment Index
- Trade Pattern Ideas
- Global Stock Market Review
- Commodity Overview
- Economic Research
- Quarterly Report
- Aggregate Technical Indicator

**Additional information:**

- Dukascopy Group Home Page
- Market News & Research
- FXSpider
- Live Webinars
- Dukascopy TV
- Daily Pivot Point Levels
- Economic Calendar
- Daily Highs/Lows
- SWFX Sentiment Index
- Movers & Shakers FX
- Forex Calculators
- Currency Converter
- Currency Index
- CoT Charts

**Social networks:**



Dukascopy Bank SA, Route de Pre-Bois 20, International Center Cointrin, Entrance H, 1215 Geneva 15, Switzerland tel: +41 (0) 22 799 4888, fax: +41 (0) 22 799 4880 research@dukascopy.com