An Adaptive Neural System for Financial Time Series Tracking

A. C. H. Dantas[1], J. M. Seixas[1]
[1]Signal Processing Lab
COPPE/EP - Federal University of Rio de Janeiro
C.P. 68504, Rio de Janeiro 21941-972, Brazil
E-mail: {augusto, seixas}@lps.ufrj.br

Abstract

In this paper, we present a neural network based system to generate an adaptive model for financial time series tracking. This kind of data is quite relevant for data quality monitoring in large databases. The proposed system uses the past samples of the series to indicate its future trend and to generate a corridor inside which the future samples should lie. This corridor is derived from an adaptive forecasting model, which makes use of the walk-forward method to take into account the most recent observations of the series and bring up to date the values of the neural model parameters. The model can serve also to manage other time series characteristics, such as the detection of irregularities.

1 Introduction

Specialists say that this century is surely the century of data [1]. In fact, the poor quality of customer data costs to U.S. business more than US$600 billion a year [2]. It is then easy to see that data are critical assets in the information economy, and that the quality of a company's data is a good predictor of its future success.

In this context, we propose a neural network based tool to treat one of the most important types of data involved in information systems: the financial time series. The development of time series forecasting methods is gaining importance as companies and research centers give more emphasis to a data based knowledge and make their investments in a data driven way. The idea in this paper is to generate an adaptive model from the series observations, and to use it to derive a monitoring "corridor" for the future samples of the series.

Note that our aim is not simply to develop a prediction algorithm; actually, what we propose is a system that monitors the quality of time series data by drawing reliability regions (the corridors) from the predicted values. These corridors may be estimated either for short-term or long-term prediction, and must be updated periodically.

In the next section, we give a brief description of some time series data quality issues. In Section 3, we summarize the neural system developed for time series modeling. The experimental data used in this work is presented in Section 4. Implementation and obtained results are detailed in Section 5, and conclusions are addressed in Section 6.

2 Time Series Data Quality Issues

Defining Data Quality (DQ) is a very context dependent task. Common sense says that it refers to conformity to the specifications. Concretely, what one is supposed to do is to set some attributes based on indicators and parameters, in order to get quantifiable metrics capable to inform the quality of a specific database [3].

For this work, since we are dealing with specific time series data, the considered DQ dimensions are *timeliness* (how up-to-date is your database?), *completeness* (does your database have missing values?) and *correctness* (how free-of-error are your data?).

2.1 Modeling Financial Time Series

Financial time series data have been traditionally decomposed in the following terms [4]:

- *Irregularities.* Sometimes a time series is affected by brusque changes that may not be predicted by any model. The September, 11 case is an example in civil aviation time series. So, in order to make data as "clean" as possible before they start to be processed, one must compensate for irregularities. The procedure proposed in this paper works towards furnishing to the system (or to the supervisor) the probabilities of the presence of irregularities in the time series;
- *Trend.* The first regular component to be removed of a time series before it can be efficiently modeled is its trend. Generally, the trend is a linear component that indicates the increase or the decrease of the series. It is important that the trend be calculated periodically, in order to permit the model to incorporate the most recent features introduced by new observations. Rarely, the trend may be modeled by nonlinear functions, such as exponential or polynomial ones [4];

- *Seasonal and cyclical components*. Besides the trend, time series generally contains cycles of regular nature. These cycles are due to the series seasonalities and ciclicities, and may be extracted by the removal of sinusoidal components (those which have greater spectral energy when compared to the others) [4].

Therefore, a time series Y_t (after the removal of the irregularities) may be expressed as:

$$Y_t = T_t + C_t + S_t + \varepsilon_t \quad (1)$$

where the right-hand terms are, respectively, the global trend, the cycles (longer period cyclical variations), the seasonal variations (weekly, monthly, annual) and a highly uncorrelated non-linear component ε_t, which we call the residual series, that is, the part of the series that remains after preprocessing is concluded. This residue may be generally viewed as a stationary stochastic process and is what should be processed (in our case, by neural models). The predicted values are then summed to the "deterministic" parts of the series in order to generate the desired forecasts.

3 The Adaptive Neural System

Since the 1990's, neural networks have been more and more applied in finance [5]. The number of financial applications such as pattern recognition, classification and time series forecasting have dramatically increased as a consequence of the fact that financial services organizations turned to be the second largest sponsors of neural network research [6].

Neural networks have an advantage over ARMA/ARIMA methods [7] because they can construct nonlinear models when mapping the input space to the output space, since they are universal function approximators. In this work, we use feedforward networks trained with the error backpropagation algorithm [8].

The proposed neural system makes use of two mechanisms that work together for performing the time series tracking task: classification and estimation. The first one is responsible to discover whether the next samples of the series should be increasing or decreasing (or even remaining within a stability region) the overall series values, and the second one gives an estimate of the value itself of the future samples. The aggregation of these two mechanisms gives us the requirements for tracking the series.

Figure 1 shows the architecture of the used network, for both classification and estimation tasks. The N most recent observations of the series form the network input vector. The hidden layer contains H biased neurons with the hyperbolic tangent as the activation function. For the output layer, we use either a hyperbolic tangent single neuron (for the classification problem) or a non-biased linear neuron (estimation). For both classification and estimation problems, the output stands for x_{t+F}, where F, the future lag, may be greater than one.

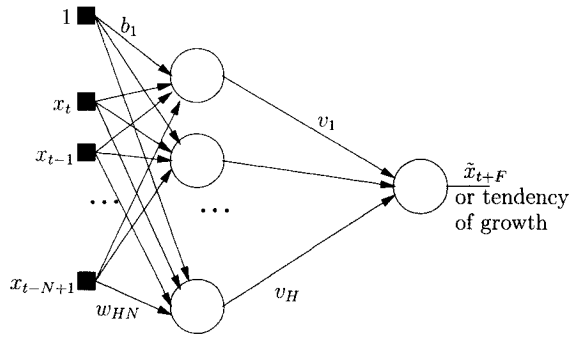

Fig. 1. Neural network architecture for time series tracking.

3.1 The Walk-forward Procedure

Time series data, due to their nature, demand efficient signal processing techniques for feature extraction and model adaption, in order to incorporate the changes of the series behaviour. For achieving this goal, the monitoring procedure will be continuously tuned to generate satisfactory predictions for the future values of the series.

The walk-forward procedure [6] is, in general, applied to an out-of-sample (offline) data set in order to simulate real-life trading and to test the robustness of the model through its frequent retraining. In our case, as it is assumed that we are continuously receiving new data, we may apply the procedure by substituting the older samples for the most recent ones, retraining the network and then predicting the future values in a up to date way. Figure 2 illustrates this process.

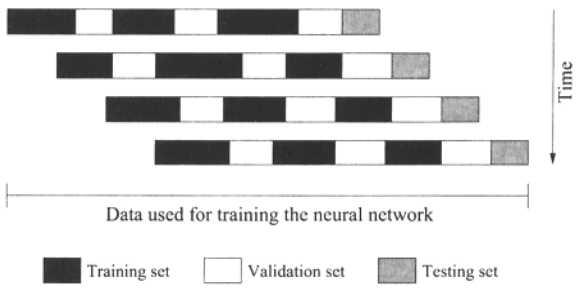

Fig. 2. Generic scheme for the walk-forward procedure.

Note that the size of the testing set is kept constant, in order to assure an independent check of the neural network; the training and the validation sets may be scaled

together since the purpose of the validation set is to determine the ability of the network to generalize[1].

4 The Experimental Data

Financial market data may be bought from specialized companies (mainly if they are *intraday*, high-frequency data) or freely downloaded from websites that offer some amount of daily stock data for the past years. The latter alternative does not guarantee "total quality" data; this means that data may be noisy, corrupted, incomplete and so on. This reinforces the interest in DQ tool developments and introduces a new challenge to the prediction problem: if we are able to properly filter the noisy data out and then correctly predict the future values of the series, we can avoid the extra cost of certified stock data.

The procedure proposed in this work is suitable to operate in this noisy data environment: as the developed model generates "corridors" inside which future data are expected to lie, we may use these "certified" regions to check for the presence of noisy samples in non-certified data (this may be used to filter irregularities out, for example).

In this work, we used five years (1998-2002) of daily stock data for companies that appear in the SP500 index. Data are under the "open-high-low-close" (OHLC) format. In order to test the methodology, three companies were selected from the computer business segment: IBM, Sun and Microsoft.

5 Implementation and Results

The first step for implementing the procedure proposed here concerns the development of an automatic preprocessing block. In our system, the neural model is updated (and so the preprocessing of the series) for every new month of data[2]. The data lag used for the network training phase is initially set to 6 months, and the tune window (the number of observations N that comprises an input vector) is set to 10. The preprocessing phase aims at achieving a series decomposition according to Equation 1, which allows to obtain the residual term. Such residual term will be used to train the network.

We considered that trends are always linear, which may be a good approximation for financial time series in non-short periods of time. After the definition of the time lag, we fit a straight line to the series (by the least squares method) and define it as the series trend.

The resulting zero-trend series is now ready to have its cycles and seasons removed. This is done by computing the series spectrum and removing frequency components that exhibit higher energies. The number NC of components to be removed is defined by an energy threshold for surviving components[3].

Figure 3 shows an example of the series trend estimation, its compensation and the spectrum of the zero-trend series before removing cycles and seasonal components. Note that the component for $f = 0$ was zero even before the removal, because the detrended series has zero mean.

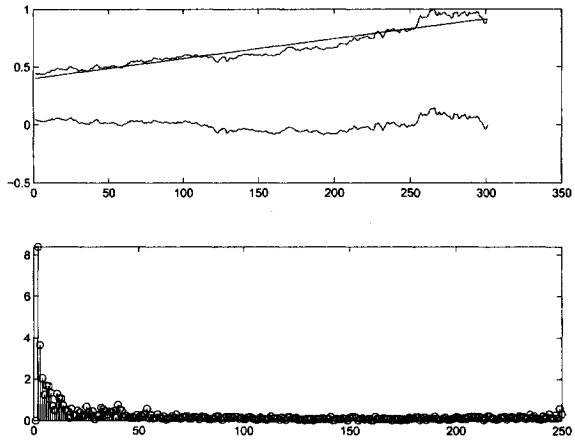

Fig. 3. The series trend compensation (top) and the spectrum of the zero-trend series (bottom).

Once the training, validation and testing sets are defined, a last normalization must be performed in order to flatten the distribution of the inputs according to the training set distribution. We calculate the mean μ and the standard deviation σ of the training set, and each input pattern x_i (for the training, validation and testing sets), becomes $x'_i = \frac{x_i - \mu}{S \times \sigma}$, where the parameter S is determined according to how wide the distribution of the training set is.

5.1 Generating the corridor

Once the networks (for classification and estimation) have converged, the validation corridor for the future values of the series may be computed. As shown in Figure 1, the network has a single output node, that is, it is optimized to predict the sample x_{t+F} having x_t to x_{t-N+1} as inputs.

For the prediction task, since there is an error in the estimation of x_{t+F} ($\tilde{x}_{t+F}$), that is, the output of the network may be seen as $x_{t+F} \pm \epsilon$, we will have access to upper and lower estimations that tend to be less accurate

[1] Usually, the training set correponds to 60-70% of the non-testing data.

[2] It may be shown that, for daily financial time series, one needs approximately one month to characterise a stable trend change. The user may, however, set another value for the retraining periodicity.

[3] We chose to remove cycles and seasons together, by looking at the spectrum of the detrended series; another option is to first eliminate seasons (when one knows their periods) and then remove cycles by spectrum analisys.

as we go further in time for the estimation process. These bounds form the estimation corridor, which is limited by the classification network output, that is, if the sample is expected to increase (decrease) the series value, the corridor may not go beyond a minimum (maximum) value, which is done by the series previous value[4]. The most accurate estimation we may produce is given by the one-day predictor ($F = 1$), in which the corridor is generated for the next series observation only. The bounds of the corridor are calculated from the last N estimations: for example, the corridor $\tilde{x}_{t+1} \pm \delta_N$ has

$$\delta_N = \frac{\sum_{i=0}^{N-1} (\tilde{x}_{t-i} - x_{t-i})}{N} \quad (2)$$

that is, the average error from the last N predictions. The corridor interval also furnishes the value for the *correctness* metric (it is 1 if $x_{t+1} = \tilde{x}_{t+1}$). In general, this one-day predictor is sufficient to provide Data Quality Management, given that we are treating daily series and we can always apply a DQ test for a new sample. *Timeliness* and *completeness* have, in our case, the maximum value (1) if data are complete and up to date, and loses 0.1 for each missing or delayed sample. Thus, the system can always returns these DQ metrics to the user.

For the available experimental series, the topology of the best trained network was 10-4-1, with constant learning rate (0.2) and momentum term (0.1). For the treated series, the neural method achieved an average of less than 2% of estimation error for the reconstructed series (residue plus "deterministic" parts of the series), while an ARMA(10,1) process obtained an error of 2.24%. For the classification problem, we achieved more than 99% of efficiency for the classification problem when determining a one-day trend detection assuming 6% of stability region, that is, we were able to establish if the next future sample of the series would increase or decrease the series values (or remain within a 6% variation region).

6 Conclusions

A neural method for Data Quality assessment of time series data was developed. The neural system provides to the supervisor a tool for monitoring the quality of the next observed samples of the series: whenever a new sample falls outside the corridor, the system warns the supervisor and either apply a correction tool (when available) or label the unreliable sample. In both cases, an increase in the overall DQ is aimed. This monitoring system can be used for detecting irregularities (with posterior filtering) and in forecasting assessment, as discussed in the text.

[4] A sample that falls very outside of this region should be considered as an irregularity.

The generation of long-term corridors through the feedback of short-term predictions may also be exploited. The main problem here is to maintain the stability of the network, which is not guaranteed due to the feedback error. We found that two kinds of difficulties may arise: the fading of the network output due to the feedback of negative errors, or the divergence of the output due to the feedback of positive errors. A new approach is being developed in which a mechanism to assure network stability is inserted in the recurrent prediction model. Preliminary results indicate that this new model will be able to furnish corridors longer than 10 days, enabling the supervisor to work with longer term forecasting.

Wavelet methods are also being tested in the preprocessing phase, in order to filter out irregularities from the training set and turn the network more sensitive to their presence in the testing set.

7 Acknowledgements

We would like to thank CNPq and FAPERJ/Brazil for funding, Insightful/USA for software support and Dr. Frank Block (Finscore/Switzerland) for his collaboration in this research.

8 References

[1] Donoho, D. (2000) High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality. Lecture for the American Math. Society "Math Challenges of the 21st Century"

[2] Eckerson, W. (2001) Data Quality and the Bottom Line. Report from "The Data Warehousing Institute"

[3] Pipino, L., Lee, Y., Wang, R. (2002). Data Quality Assessment. Communications of the ACM, vol. 45, n. 4ve, pp. 211-218

[4] Chatfield, C. (1984). Analysis of Time Series. Chapman and Hall

[5] Poddig, T., Rehkugler, H. (1996). A 'world' model of integrated financial markets using artificial neural networks. In: Neurocomputing 10, Elsevier, pp. 251-273

[6] Kaastra, I., Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. In: Neurocomputing 10, Elsevier, pp. 215-236

[7] Box, G., Jenkins, G. (1970). Time Series Analisys, Forecasting and Control. Holden Day

[8] Haykin, S. (1999). Neural Networks - a Comprehensive Foundation, 2nd ed. Prentice-Hall