Proceedings of IWANN'99 – International Work-conference on Artificial and Natural Neural Networks, Alicante (Spain), June 2-4, 1999, Springer, Lecture Notes in Computer Science 1607, J. Mira, Juan V. Sanchez-Andres eds., ISBN 3-540-66068-2, pp. II-596 – II-605.

Forecasting financial time series through intrinsic dimension estimation and non-linear data projection

M. Verleysen[1], E. de Bodt[2], A. Lendasse[3]

[1] Université catholique de Louvain, CERTI,
3 pl. du Levant, 1348 Louvain-la-Neuve, Belgium
verleysen@dice.ucl.ac.be
[2] Université Catholique de Louvain, IAG-FIN,
1 pl. des Doyens, 1348 Louvain-la-Neuve, Belgium
debodt@fin.ucl.ac.be
[3] Université catholique de Louvain, CESAME-AUTO,
4 av. G. Lemaître, 1348 Louvain-la-Neuve, Belgium
lendasse@auto.ucl.ac.be

Abstract. A crucial problem in non-linear time series forecasting is to determine its auto-regressive order, in particular when the prediction method is non-linear. We show in this paper that this problem is related to the fractal dimension of the time series, and suggest using the Curvilinear Component Analysis (CCA) to project the data in a non-linear way on a space of adequately chosen dimension, before the prediction itself. The performances of this method are illustrated on the SBF 250 index.

1. Introduction

Time series forecasting is a problem encountered in many industrial (electrical load, river flow…) and economic (exchange rates, stock exchange…) tasks. Often, prediction must be done without indication about the (unknown) underlying process; input values to the prediction method must thus be chosen by trial and error. In some situations, a priori information can be fed into the prediction method, but this remains an exception: as an example, weekly and monthly past values are obviously good candidates to predict the electrical load.

In most situations however information about the underlying process is hardly available. Forecasting with non-linear methods is then usually achieved through one of the two following methods:

- linear prediction models (for example ARX) are built; the best auto-regressive order of the linear model is used for the non-linear prediction method too.
- non-linear prediction models only are used: many possible auto-regressive orders are investigated and the best one is chosen by trial and error.

These methods are often the only possible ones but they have large defects. The first over-estimates the autoregressive order necessary (because it does not take account of non-linear dependencies between the data) and lead to overfitting. The second is very heavy to implement and often not very reliable; indeed, the various trainings can be sullied with errors which are caused by the method of prediction itself, such as for example the presence of local minima in the optimization of Multilayer Perceptrons.

The method suggested in this paper try to overcome these disadvantages. It will be exposed in the second part and then applied to a simple artificial example. In the fourth part, we will try to predict the successive fluctuations of the SBF 250 Stock Market Index.

2. Forecasting method

2.1. Autoregressive order and vector

The autoregressive non-linear order can be defined as the optimal number of past values to use in a time series for a good prediction. The autoregressive vector includes these past values. Using a non-linear method to evaluate the autoregressive order must make it possible to take into account the non-linear relations between past values of the series; a traditional linear method to estimate the autoregressive order only takes into account the correlation (linear dependence) between past values.

One can still choose the autoregressive vector in two ways. The first one consists in estimating the optimal autoregressive order *n*, and to look for the best *n* past values in the series to use for the prediction [5, 9]. Another possibility is to look for a *n*-dimensional vector built with non-linear mixings of the past values of the series, instead of the raw values themselves.

In the following we will use the second possibility. We will first look for a way to estimate the non-linear autoregressive order, and secondly we will build the auto-regressive vector with a projection method.

2.2. Intrinsic dimension

In order to determine the non-linear autoregressive order, we will use the notion of “intrinsic” dimension of a set of points. Without going into mathematical details, the intrinsic dimension of a data set can be defined as the minimum number of coordinates that would be necessary to describe the data without loss of information, is these coordinates were measured on curved axes. For example the intrinsic dimension of a set of points forming a string in dimension 2 (or higher) is 1, and the intrinsic dimension of a set of points forming a non-planar surface in dimension 3 (like the well-known horseshoe distribution) is 2.

First we build an autoregressive vector of size *m* from the last past values of the raw time series. This vector will have to be sufficiently large to contain all information necessary to a good prediction. One possible solution is to take the optimal autoregressive vector for an ARX model [5]; indeed this one is built in a way that it contains “sufficient” information when used with a linear prediction method, and will thus obviously contain enough information too when used with a non-linear prediction method. Larger vectors can be taken for more security, but they would make more difficult the continuation of work. An autoregressive vector is built at each time step; they are laid out as rows in a matrix called autoregressive matrix.

Since it is supposed that there is an excess of information in the autoregressive vectors, we will try to reduce their dimension. This goes through a first step which consists in estimating an optimal reduced dimension, which will be identified to the fractal dimension of the set of points (the autoregressive vectors) in a *m*-dimensional space. This value will be further referred as the fractal dimension of the autoregressive matrix. It can be interpreted as the number of columns "non-linearly independent" of this matrix: there is a non-linear transformation which makes it possible to entirely rebuild the matrix from *d* columns.

To estimate the fractal dimension of the autoregressive matrix, we use the Grassberger and Procaccia method [4]; many other methods can however be used to estimate a fractal dimension [1, 6, 7]. It must be mentioned that the concept itself of non-linear dependency is difficult to define. Therefore the fractal dimension found by these methods can vary; in difficult situations, it may be worthwhile to use several methods in order to asses their results. The intrinsic dimension can also be a non-integer value; in the following, we will use the integer value nearest to the intrinsic dimension as an approximation of the *non-linear autoregressive vector* size defined below.

2.3 Non-linear autoregressive vector

The following step consists in building a *non-linear autoregressive vector* of size *d* from each of the *m*-dimensional autoreressive vectors.

The set of points defined by the rows of the autoregressive matrix form a *d*-surface in a *m*-dimensional space. If we could unfold this *d*-surface by projecting the *m*-dimensional space onto a *d*-dimensional one, keeping the topology of the initial set, we would obtain a *d*-dimensional non-linear autoregressive matrix that could be used for further prediction.

Many non-linear “projection” methods exist. Kohonen’s self-organizing map is probably the most widely known example. Yet in our experiments we will use another method, the Curvilign Component Analysis (CCA) [3]; unlike the Kohonen maps, this method doe not make any assumption on the shape of the projection space, and was found to give better results in our application.

2.3 Non-linear forecasting

After this projection, we obtain the required non-linear autoregressive matrix. Its rows will be used as input vectors to any non-linear forecasting method. We used in our experiments the standard multi-layer perceptron (MLP) and radial-basis functions (RBF) as prediction core.

Obviously, the prediction method could also use the initial *m*-dimensional autoregressive vectors extracted from the raw series. Nevertheless, it must be reminded that even if neural networks are known to be good candidates (compared to other non-linear interpolators) when dealing with the curse of dimensionality, it remains that, for a fixed number of training vectors, their performance decrease with the dimension of their input vectors. The interest of our method is precisely here: we expect that the little information lost in the non-linear projection will be largely compensated by the gain of performance in the forecasting itself. This will be illustrated in the examples below.

3. Artificial time series example

In order to test the above method, we built a chaotic artificial time series from the following non-linear equation:

$$x_{t+1} = \mathrm{a}\, x_t^2 + b\, x_{t-2} + \varepsilon \qquad (1)$$

Obviously, the non-linear autoregressive order of this time series is 2 (it is generated from 2 past values). Let us note the lack of a x_{t-1} term, as well as the presence of a noise ε (about 10% of the maximum value of the series).

This series is represented on Figure 1.

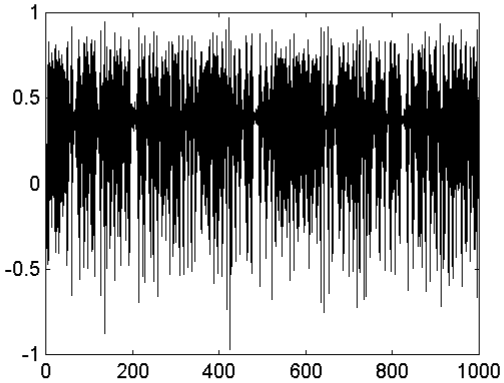

Fig. 1. Artificial time series generated according to equation (1).

The first step of our method consists in the search for the optimal autoregressive matrix for a linear ARX prediction model.

Figure 2 shows the sum (on 1000 test points) of the quadratic errors obtained if one uses a standard ARX model of increasing size; the *x*-coordinate of the figure is the autoregressive order.

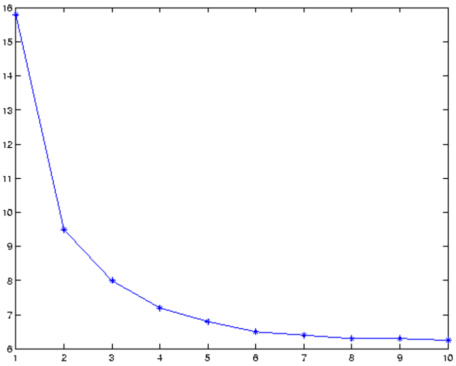

Fig. 2. Sum of quadratic errors (on 1000 test points) obtained with an ARX model for different values of the autoregressive order.

To ensure to collect the whole dynamics of the series, we will build an initial autoregressive matrix of order 6. The estimation of the fractal dimension of this matrix gives 2.12, which is very close to reality.

The following step of the method is the projection of the set of the points (rows of the autoregressive matrix) from R^6 to R^2. Note that in the simulations we added the x_t term to the two coordinated found by this projection, in order to improve the results. The final autoregressive vector dimension is thus equal to 3.

In a next step we used this 3-dimensional autoregressive vector as inputs to a non-linear prediction model. We used a Multi-Layer Perceptron with one hidden layer. The sum of quadratic errors obtained with this MLP is around 5 (on 1000 points), which significantly lower than the errors illustrated in Figure 2 (linear model).

We also compared this result to the error obtained with a similar Multi-Layer Perceptron, where the input vector is the set of *p* last values from the raw series. Figure 3 shows this error for different values of *p*. The horizontal line corresponds to the error obtained with our method; we conclude that we obtain (for this example) an error similar to a result obtained by trial and error on several non-linear models, which was the goal of our investigation. This easiness of implementation will be valuable when dealing with a "real-size" dataset for which the non-linear autoregressive order is unknown.

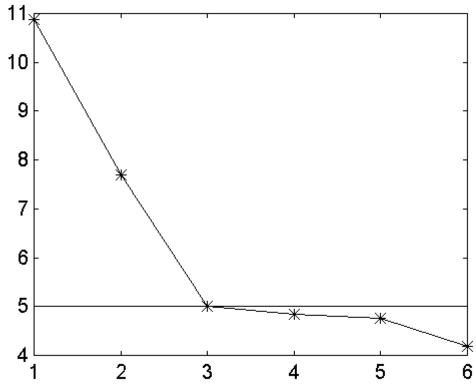

Fig. 3. Sum of quadratic errors (on 1000 points) obtained with a MLP network for different values of the autoregressive order. The horizontal line corresponds to the result of the proposed method.

4. Application to the SBF250 Stock Market Index

An interesting example of time series in the field of finance is the SBF 250[1] index. The application of time series forecasting produces to financial market data is a real

[1] The SBF 250 is one of the reference index of the French stock market. As suggested by its name, it is based on a representative sample of 250 individual stocks.

challenge. The efficient market hypothesis (EMH) remains up to now the most generally admitted one in the academic community, while essentially challenged by the practitioners. Under EMH, one of the classical econometric tool used to model the behavior of stock market prices is the geometric Brownian motion[2]. If it does represent the true generating process of stock returns, the best prediction that we can obtain of the future value is the actual one. Results presented in this section must therefore be analyzed with a lot of caution.

To succeed in determining the variations of the SBF250, other variables being able to influence its fluctuations are included as inputs (extrinsic variables). We selected three international indexes of security prices (S&P500, Topix and FTSE100, respectively American, Japanese and English), two rates of exchange (Dollar/Mark and Dollar/Yen), and two American interest rates (T-Bills 3 months and US Treasury Constant Maturity 10 years). We used daily data over 5 years (from 01/06/92 to 01/12/97), to have a significant data set.

The problem considered here is the forecasting of the SBF250 index at time *t*+1, from available data at time *t*.

To capture the relations existing between the French (non-stationary) index and the other variables chosen, a co-integration is necessary. The result of this co-integration is the (stationary) residues of the SBF250 index, defined by the difference between the true value SBF_{t+1} and the approximation $\hat{\mathrm{S}}\mathrm{BF}_{t+1}$ given by the model:

$$R_t = \mathrm{SBF}_{t+1} - \hat{\mathrm{S}}\mathrm{BF}_{t+1} = \mathrm{SBF}_{t+1} - (\alpha_{\mathrm{t}} + \sum_{\mathrm{i}=1}^{7} \beta_{\mathrm{t,i}} \cdot \mathrm{I}_{\mathrm{t,i}}) \tag{2}$$

where $I_{t,i}$ $(1 \le i \le 7)$ are the 7 selected variables at time *t*..

In the following, we will focus on the forecast of these residues, or more exactly on the forecast of daily return of these residues. Indeed, it is more useful for somebody eager to play on the market, to forecast its fluctuations rather than its level. To predict that the level of the SBF index tomorrow is close to the level today is trivial. On the contrary, to determine if the market will raise or fall is much more complex and interesting.

The daily return ρ_t of the R_t residue at time *t* is defined by:

$$\rho_{\mathrm{t}} = \frac{R_t - R_{t-1}}{R_{t-1}} \tag{3}$$

[2] Stock prices would follow the following diffusion process : $\frac{dS}{S} = \mu dt + \sigma dz,$ where $dz = \varepsilon\sqrt{dt}$ and $\varepsilon \approx N(0,1)$. *S* is the stock price, μ is the drift rate by unit of time and σ is the instantaneous volatility.

According to Refenes et al. [12], we will use technical indicators directly resulting from the outputs of the residues:

• ρ_t , ρ_{t-10}, $\rho_{t-20,}$ ρ_{t-40} :	returns ;
• ρ_t - ρ_{t-5}, ρ_{t-5} - ρ_{t-10}, ρ_{t-10} - ρ_{t-15}, ρ_{t-15} - ρ_{t-20} :	differences of returns ;
• K(20), K(40) :	oscillators ;
• MM(10), MM(50) :	moving averages ;
• MME(10), MME(50) :	exponential moving averages. ;
• ρ-MME(10), ρ-MME(50)	return and moving average differences ;
• MME(10)-MME(50).	moving average differences

.

If we carry out Principal Component Analysis (PCA) on these 17 indicators, we note that 99,72% of the original variance is kept with the first eleven principal components: 6 technical indicators can be removed without loss of information.

The target variable, whose sign has to be predicted, is a forecast variable over 5 days:

$$C_{t+6} = \sqrt[5]{\frac{R_{t+6}}{R_{t+1}}} \tag{4}$$

The time series of this variable is illustrated in Figure 4.

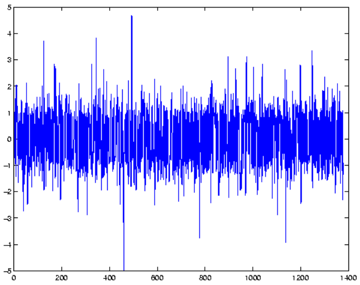

Fig. 4. Time series of the target variable according to equation (4).

This variable has to be predicted using the 11 indicators selected after PCA. The interpolator we used is a Radial-Basis Function (RBF) network with the learning algorithm presented in [13]. The network is trained with 1000 points and tested on 100 other points. Our interest goes to the sign of the prediction only, which will be compared to the real sign of the prediction variable.

The best results we obtained are 60,2% correct approximations of the sign of the series on the training set, and 48 % on the test set. This result is obviously bad: it is worst than a pure random guess on the test set!

On the other hand, is we use the proposed method and estimate the fractal dimension of the data set, we obtain an approximate value of 5. We then use the CCA method to project the 11-dimensional data (after PCA) on a 5-dimensional space. Thereafter, we use another RBF network to approximate the variable to predict. We obtain 61% of correct sign prediction on the training set and 57 on the test set. This result seems to be significantly better than the result that we could get by using a purely naïve approach (for example, by predicting always a + sign). A lot of simulation work remains however to be done to validate it (by, for example, constructing a bootstrap estimator).

Still better results were obtained using a MLP instead of a network (more than 62% correct sign predictions on the validation set). Unfortunately, the results obtained with a MLP are difficult to repeat for various initial conditions, convergence parameters,… We prefer to restrict our performances to those obtained with a RBF network, because they are much less parameter-dependent.

5. Conclusion

The proposed method for the determination of the best autoregressive vector gives satisfactory results on a financial series. Indeed, the quality of the prediction obtained is either comparable to the quality obtained with other methods (slightly higher on a real-world financial time series, and equivalent on an artificial data set). The advantage of our method mainly comes from the systematization of the procedure: there is no need for many trials and errors for the determination of the variables to use at the input of the predictor and of its parameters. Moreover, the determination of the autoregressive vector is completely independent from the prediction method. Ameliorations of the proposed method could be searched in alternative ways to estimate the fractal dimension of the series or to project the data in a non-linear way.

The question of the predictability of a series such as the SBF250 index remains. The results presented in this paper are promising, but could certainly be improved. We must also remind that predicting a complex, mostly stochastic time series as the SBF250 must be achieved with several prediction methods, in order to cross-validate their results. It must also be noted that, the simple fact of being able to forecast, at a certain level of confidence, a financial time series is not in itself sufficient to invalidate the EMH. The problem is to see if it is possible to exploit the prediction algorithm to obtain abnormal returns, that is to say returns that take into account the level of the risk generated by the trading strategy as well as the associated transaction costs.

References

1. Alligood K. T., Sauer T. D., Yorke J. A.: Chaos: An Introduction to Dynamical Systems. Springer-Verlag, New York (1997), pp. 537-556
2. Box G.E.P., Jenkins G.: Time Series analysis: Forecasting and Control. Cambridge University Press (1976)
3. Demartines P., Hérault J.: Curvilinear Component Analysis: A self-organizing neural network for nonlinear mapping of data sets. IEEE Trans. on Neural Networks **8**(1) (1997) 148-154
4. Grassberger P., Procaccia I.: Measuring the Strangeness of Strange Attractors. Physica **D56** (1983) 189-208
5. Ljung L.: System Identification - Theory for User. Prentice-Hall (1987)
6. Tackens F.: On the numerical Determination of the dimension of an attractor. In: Lecture Notes in Mathematics Vol. 1125, Springer-Verlag (1985) 99-106
7. Theiler J.: Statistical Precision of Dimension Estimators. Phys. Rev. **A41** (1990) 3038-3051
8. Weigend A. S., Gershenfeld N.A.: Times Series Prediction: Forcasting the future and Understanding the Past. Addison-Wesley Publishing Company (1994)
9. Xiangdong He, Haruhiko Asada: A New Method for Identifying Orders of Input-Output Models for Nonlinear Dynamic Systems. In: Proc. of the American Control Conf., San Francisco (CA) (1993) 2520-2523
10. Burgess A.N.: Non-linear Model Identification and Statistical Significance Tests and their Application to Financial Modelling. In: Artificial Neural Networks, Inst. Elect. Eng. Conf., June (1995)
11. Fama E.: Efficient Capital Markets: A Review of Theory and Empirical Work. Journal of Finance **XXV** No 2 (1970) 383-417
12. Refenes A. N., Burgess A.N. and Bentz Y.: Neural Networks in Financial Engineering: A Study in Methodology. IEEE Transactions on Neural Networks **8**(6) (1997) 1222-1267
13. Verleysen M., Hlavačkova K..: An Optimized RBF Network for Approximation of Functions. In: Proc of European Symposium on Artificial Neural Networks, Brussels (Belgium), April 1994, D facto publications (Brussels).