Deterministic regression model and visual basic code for optimal forecasting of financial time series

Alejandro Balbás [a], Beatriz Balbás [a], Inna Galperin [b], Efim Galperin [c,*]

[a] University Carlos III of Madrid, Department of Business Economics, CL. Madrid, 126. 28903 Getafe, Madrid, Spain
[b] Rotman School of Management, University of Toronto, 105 St. George Street, Toronto, ON M5S 3E6, Canada
[c] Université du Québec à Montréal, Département des mathématiques, C.P.8888, Succ. Centre Ville, Montréal, Québec, H3C 3P8, Canada

A R T I C L E I N F O

Article history:
Received 22 May 2008
Accepted 10 July 2008

Keywords:
Sliding deterministic regression models
Optimal forecasting in finance

A B S T R A C T

A new, non-statistical method is presented for analysis of the past history and current evolution of economic and financial processes. The method is based on the sliding model approach using linear differential or difference equations applied to discrete information in the form of known chronological data (time series) about the process. An algorithm is proposed that allows one to project the current evolution of the process onto some period of its future development. Computer code in visual basic is developed that has been validated in application to American stock index *S&P* 500, with predicted values within 5% of real data over long periods of the recent past history. The algorithm and the code can be applied to practical problems in finance and economy in time of its normal evolution without catastrophic events.

1. Introduction

Forecasting of different time series is ubiquitous in finance and economy. Usually it is done by statistical methods complemented by construction of a model [1]. Models are normally built on the basis of the entire sequence of available data, then some specific properties are investigated, and the sequence is projected by extrapolation onto some period in future, with subsequent improvement of the model using new data appearing in due time if those data essentially deviate from the predicted values [2].

The novelty of the method presented in this paper consists in a sliding split-level approach using linear differential and/or difference equations that incorporate polynomial, exponential or logarithmic growth or fall with intermittent oscillations reflecting disturbances of the market. A recent segment of the available sequence, say 30%, is reserved solely for verification, comparison and innovation purposes. It is not used in the construction of a model, and on the basis of preceding 70% of raw data the optimal sliding model is constructed using the least-squares solution provided by the pseudo-inverse matrices [3]. This solution is further optimized with respect to the order of the difference equation and checked with respect to a given risk factor to assure the quality of the predictor. If the quality is acceptable, the base subsequence is shifted forward with a *new* difference equation identified in the same way. Experiments with the American index *S&P* 500 and some other time series demonstrate that in the course of normal evolution of the economy without catastrophic events (such as war, massive fraud, recession or crisis) the method yields good forecasts within the range of 5% deviation in predicted values compared with the known values actually appeared.

* Corresponding author.
E-mail addresses: alejandro.balbas@uc3m.es (A. Balbás), beatriz.balbas@uc3m.es (B. Balbás), inna.galperin@rotman.utoronto.ca (I. Galperin), galperin.efim@uqam.ca (E. Galperin).

The paper is organized as follows. Section 2 presents a summary of the general forecasting method applied to predict the evolution of financial time series. This method significantly draws on the notion of the Moore–Penrose pseudo-inverse matrix [4–7] and some properties of dynamical systems. Further details may be found in [8–10] though the synopsis presented in this section is sufficient to understand the general approach. Section 3 describes numerical procedure and its application to forecasting in the presence of financial regulation. Section 4 describes the features of the Visual Basic Code that we have developed in order to apply the forecasting method proposed in [10]. The code is given in Appendix at the end of the paper. Section 5 reports the results of empirical implementation dealing with the American stock index *S&P* 500, probably the most important stock index all around the world, providing further discussion about it as well as some closely related empirical studies. Concluding remarks are presented in Section 6 followed by references immediately relative to the topics considered.

2. Sliding uniformly optimal predictor of minimum order

Given a finite sequence of evenly spaced observations $y_i = y(t_i)$, $t_i = t_0 + i\Delta t$, $\Delta t = \text{const}$, $i = 0, 1, \ldots, n$, one usually tries to fit these data by some curve (linear, exponential, or other deterministic regression) and then extrapolate this curve to forecast one or several close future values in a process given in observations. If such forecasts appear to be poor, then certain probabilities are assigned to the past observations and extrapolation onto the future is made *in probability*.

A more interesting approach is to model the process given in observations not with a curve (deterministic or stochastic) but with a vector system of differential or difference equations

$$x_{i+1} = F(p)x_i, \qquad y_i = h(p)x_i, \quad x_i \in R^m, y_i \in R^1, i = 0, 1, \ldots, n, \tag{1}$$

where p is vector-parameter to be found for a model (1) in order that y_i provided a good approximation to the known actually realized past data. System (1) is the exact sample-data representation of the corresponding continuous system of linear differential equations, $\mathrm{d}x/\mathrm{d}t = A(p)x$, $y(t) = h(p)x$, for which $F = \exp(A\Delta t) = \sum A^k \Delta t^k / k!$ Such systems incorporate all possible linear, polynomial, exponential, logarithmic, sine or cosine curves, separately or combined.

It is known [11], see also [8], that a linear stationary model, continuous or discrete (1), exists if and only if the observations $y_0, y_1, \ldots$ satisfy the linear stationary difference equations

$$y_{i+r} = a_1 y_i + a_2 y_{i+1} + \cdots + a_r\, y_{i+r-1}\,, \quad r \le m,\ a_j = \text{const}(j = 1, \ldots, r), i = 0, 1, \ldots, n - r. \tag{2}$$

Here r is the order and $a = (a_1, a_2\,, \ldots, a_r) = \text{const}$ is dynamics of the free motion system. If the observed data $\{y_i\}$ fit to an equation of the type (2), then parameters r, a can be identified and Eq. (2) so obtained can serve as predictor for the process represented by the known past data of a time series. If an exact equation (2) holds, it is a *dynamic model* corresponding to a system in the form (1) whereby r, a are functions of F, h, p. If (2) does not hold for any r, a, it means that there is no linear stationary dynamic model representing the process given in observations. In this case, instead of (2), one can write:

$$y^*_{i+r} = a_1 y_i + a_2 y_{i+1} + \cdots + a_r\, y_{i+r-1}, \quad i = 0, 1, \ldots, n - r, \tag{3}$$

$$\eta_i\,(r, a) = y_{i+r} - y^*_{i+r}, \quad a = (a_1, a_2\,, \ldots, a_r). \tag{4}$$

If $|\eta_i(r, a)|$, $i = 0, 1, \ldots$, for certain r, a are all sufficiently small, then relations (3), which we call *regression model* in contrast with dynamic model (2), yields a predictor accurate up to $|\eta_i(r, a)|$ for the process given in observations. Clearly, regression models (3) and (4) yield broader class of predictors than dynamic models. Here we consider modeling of processes with completely unknown structure. If the observations fit into Eq. (3) with acceptable degree of accuracy $|\eta_i(r, a)| \le \eta$ in (4), then the process possesses the property of approximate linearity in the regression sense stated above. Otherwise, the process is essentially nonlinear. No noise is explicitly considered in Eqs. (3) and (4), although their imprecision (4) allows for a bounded colored noise of unknown characteristics. In contrast with dynamic models, deterministic regression models do not possess the semi-group property and so represent a far more powerful instrument in mathematical modeling than dynamic models.

2.1. Least-squares solution

Repeatedly writing Eq. (2) for $i = 0, 1, \ldots, n - r$, one obtains the system

$$Y = PX, \qquad Y = \begin{bmatrix} y_r \\ y_{r+1} \\ \cdots \\ y_n \end{bmatrix}, \qquad P = \begin{bmatrix} y_0 & y_1 & \cdots & y_{r-1} \\ y_1 & y_2 & \cdots & y_r \\ \cdots & \cdots & \cdots & \cdots \\ y_{n-r} & y_{n-r+1} & \cdots & y_{n-1} \end{bmatrix}, \qquad X = \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_r \end{bmatrix}. \tag{5}$$

Vector equation (5) represents dynamic system (2) of $n-r+1$ equations starting from $i = 0$. In sliding mode, the system (2) is considered on adjacent segments $i = j, j+1, \ldots, j+n-r \le N-(n-r+1), j = 0, 1, \ldots$, where N accounts for n given observations (raw data) plus $N - n$ predicted values. Thus, Eq. (5) should be properly written as $Y_j = P_j X$, $j = 0, 1, \ldots, N^*$, where N^* corresponds to the last required predicted value. If instead of dynamic system (2) we consider regression system

(3), then (5) should be written as $Y_j^* = P_jX$, or as $Y_j^* = P_j^*X$ if some predicted values are returned to Excel sheet (see Code) and used in a base segment for further predictions. Note that dynamics $X = \{a_i\}$ in (2)–(5) is always the same for any j. For simplicity of presentation, the subscript j and asterisk (*) are dropped in the following description of the algorithm and the numerical procedure. However, they are accounted in the Visual Basic Code (see Appendix), and the reader should keep it in mind when using the Code and the Algorithm.

If for some r we have $\operatorname{rank} P = \operatorname{rank}[P, Y] = q \le r$, then X is determined (uniquely, if $q = r$, or not, otherwise). The result is a dynamic model of order r. If for all reasonable orders r we have $\operatorname{rank} P < \operatorname{rank}[P, Y]$, then systems (5) are all inconsistent, and one can use the Moore–Penrose pseudo-inverse $P^\#$, [4–7], to obtain the minimum-norm least-squares solution [3–7] for a fixed r:

$$X^0 = [a_1, a_2, \dots, a_r]^{\mathrm{T}} = P^\# Y \quad (^{\mathrm{T}}\text{-transpose}). \tag{6}$$

Resulting coefficients yield a regression model. As distinct from dynamic models with square matrix P, $\operatorname{rank} P = \operatorname{rank}[P, Y] = r$, when $P^\# = P^{-1}$, a regression model does not start from one single initial segment but rather at each step it starts from different successive segments given in observations. The predicted values $\{y^*_{i+r}\}$ defined by the base segments of a regression model (3) do not coincide with the actual observations as in the case of a dynamic model (the semi-group property), but present some other values in the neighborhoods of those observations, and such that each value is based on r immediately preceding actual (and predicted, if any) values, and not on the r very first ones as in the case of a dynamic model.

There are many pseudo-inverse matrices [3] of which the Moore–Penrose pseudo-inverse in (6) yields the *minimum-norm* solution: $\|X^0\| = \|a\| = \sum a_i^2 = \min$, with respect to other X rendered by other pseudo-inverses if used in (6), see [3, pp. 114–115]. This property assures the smallest coefficients a_i in (3), thus not allowing inaccuracies of raw data to be amplified in computation.

2.2. Further optimization and financial interpretations

The optimal solution (6) yields the residual sliding discrepancy vectors

$$Z_j(r) = P_jX^0 - Y_j, \quad j = 0, 1, \dots, N^* - r, \tag{7}$$

which are *minimal* in the Euclidean norm

$$\|Z_j(r)\| = \sum_{i=1}^{n-r} z_{ij}^2 = \|P_jX^0 - Y_j\| \le \|P_jX - Y_j\| \quad \text{for all } X, \text{ any } j. \tag{8}$$

This property assures the best precision of computed values in (3) for any order r fixed in advance. For dynamic models with $P^\# = P^{-1}$, we have $Z_j(r) = 0$ for any r, any j. For a regression model, to obtain further improvement of the model (3), the optimization should be carried out with respect to r in order to find

$$r_0 = \arg\min_r \sum_{j=0}^{N^*-r} \|Z_j(r)\|, \quad Z_j(r_0) = P_jX^0 - Y_j|_{r=r_0}. \tag{9}$$

If for all segments $\{y_i\}$ in (3), (4) serving as bases for prediction we have

$$|\eta_i(r_0, a)| \le \eta, \quad i = 1, \dots, n - r, \tag{10}$$

where η is precision of observations or an acceptable degree of accuracy in forecasting, then a regression model is found, otherwise, no acceptable least-squares regression model exists for a process given in observations. In finance, this reflects the volatility of financial data, and η can be interpreted as the acceptable risk margin.

There is an obvious upper bound for "reasonable" orders of a regression model. From system (5), it can be seen that for high orders, $r > n - r + 1$, the systems (5) are solvable, so that linear stationary dynamic models of high order always exist even for essentially nonlinear processes, and such models can be fit to any data [10, pp. 163–164]. However, validation of such models by the past history becomes impossible and the whole construction loses ground. Appearance of high orders in an attempt to find a good fit is an indication that the underlying process is essentially nonlinear and does not admit a linear stationary model. For a given number n of observations, the second minimization (9) should be carried up to the orders $r < n/4$, to assure reasonable quality of the predictor and a sufficient segment of recent observations reserved for the validation of the predictor model [10].

3. Numerical procedure and financial regulation

The matrix P in (5) that has r columns and $n - r$ rows, $r < n - r$, is usually denoted as $r \times (n - r)$ matrix P^r_{n-r}. Such a matrix of rank $q \le r$ admits full-rank factorizations $P = FG$ where F of the full column rank q can be chosen as any maximal linearly independent set of columns of P. For example, if the first q columns of P are linearly independent, then $P = [F, B]$, where sub-matrices F, B of P (vertical blocks) are known. Now, we can set $G = [I, K]^{\mathrm{T}}$ where I is the $q \times q$ identity matrix and K_q^{r-q} is the matrix to determine. We have $P = FG = [F, FK]$, hence, $F_{n-r}^{q} K_q^{r-q} = B_{n-r}^{r-q}$. Since matrix F is of the rank q, it has exactly q linearly independent rows. Suppose that those rows are the first (or the last). Then, making simultaneous elementary row operations on the two adjacent blocks F, B, we can transform the upper (or lower) square block of F into the $q \times q$ identity matrix, getting in the same place of B the required matrix K_q^{r-q}. If the above mentioned columns or blocks are not the first nor the last, they can be permuted to become such, then, after the transformation, the reverse permutation can be made to obtain the required full-rank factorization $P = FG$.

An alternative method to find a full column rank sub-matrix F of P is to apply the Gram–Schmidt orthogonalization process, see, e.g., [3, pp. 286–287].

Lemma (*MacDaffee, 1959, See [3, p. 23]*). *If matrix P has the full-rank factorization $P = FG$, then its Moore–Penrose pseudo-inverse is given by*

$$P^{\#} = G^{\mathrm{T}}(F^{\mathrm{T}}PG^{\mathrm{T}})^{-1}F^{\mathrm{T}} \quad ({}^{\mathrm{T}}\text{-transpose}), \tag{11}$$

where the matrix in parentheses is nonsingular. □

Note that formula (11) applies also to complex matrices in which case $({}^{\mathrm{T}})$ means conjugate transpose. The important special case occurs when matrix P has full column rank, thus, $P = F$, $G = I$, identity matrix, and from (11) we obtain

$$P^{\#} = (P^{\mathrm{T}}P)^{-1}P^{\mathrm{T}}. \tag{12}$$

An alternative method to obtain minimum-norm X^0 consists in direct minimization $\min \|PX - Y\| = \|Pa - Y\|$ with respect to X, and then finding the minimum-norm vector $X^0 = a$ in the set of $Z_j(r) = P_jX - Y_j$, cf. (7). The solution X^0 of the first minimization problem may be obtained by solving a linear system of equations. Indeed, it is implied by the fact that $P_jX - Y_j$ must be orthogonal to the columns of the matrix P_j, and it is characterized by this property, which clearly generates a linear system. Once P_jX has been computed, the second minimization problem is presented by the second linear system of equations. Indeed, it holds since the minimum-norm $X^0 = a$ in the set of $Z_j(r) = P_jX - Y_j$ must be orthogonal to the kernel of the linear function associated with the matrix P_j.

3.1. Free market situation

This is the case when matrix P has full column rank, thus, $P^{\#} = (P^{\mathrm{T}}P)^{-1}P^{\mathrm{T}}$ due to (12), and we have in (6):

$$X^0 = [a_1, a_2, \ldots, a_r]^{\mathrm{T}} = (P^{\mathrm{T}}P)^{-1}P^{\mathrm{T}}Y. \tag{13}$$

With free fluctuations of prices influenced by competition and changing pattern of demand, this is normal situation in finance with respect to a randomly chosen time series of financial data, cf. mutual funds. If by chance the matrix P in (5) for some $i \geq 0$ does not have full column rank, then determinant $\det P^{\mathrm{T}}P$ will be close to zero which will manifest itself in appearing of large numbers in (12). In this unlikely case, the general formula (11) or direct optimization method above should be used, or some observations y_i, $0 \leq i \leq n - r$, should be slightly perturbed along a diagonal of P to obtain linearly independent columns, thus, P of full column rank. This may involve some experimentation to assure robust final forecasts validated by the known past observations.

3.2. Totally regulated market (price and wage controls)

This may be the case when P does not have full column rank. Theoretically, it may happen that $\operatorname{rank} P = 1$ which presents dynamic model. If $\operatorname{rank} P > 1$, the general formula (11) can be used. Totally regulated market with respect to certain time series may exist. Totally regulated time series may be detected by $\det P^{\mathrm{T}}P \approx 0$ or by appearing large numbers in computation of (12) when experimenting with different time intervals Δt for sequences $y_i \in P$ in (5).

3.3. Mixed case. Known and/or unknown partial regulation

This is the general case when P most likely has full column rank. In this case, reasonable entries appear in (12), (13), and good fit is obtained for predicted values compared with known raw data over a period of past history.

3.4. Implicit sliding models and uniformity of confidence

The sliding model approach does not require a study of structural properties of a system. Those properties are implicitly contained in the known past history given in observations. If a system possesses a kind of inertia for some time in its forward evolution, which is the case in economy, finance, and some other areas of social sciences and technology, then implicit structure contained in observations is preserved for some time in future and can be used to project the evolution for this time period. This is the basis of the sliding model approach in finance and economy which can be expressed in the following statement [10] modified for economy and finance:

Uniform volatility principle. *If a sliding model forecasts are within the volatility margin over a period of past evolution and there is no new factor which may essentially affect the behavior of economy, then the predictor can be used to forecast the performance over some period in the immediate future.*

4. Features of the code

The Visual Basic code provided in Appendix is used to forecast the future evolution of a time series on the basis of known past segments (past history) of this time series. Detailed comments are presented in appropriate places within the code. The code only requires the introduction of past data of a segment of the sequence under consideration. Once the data are in an Excel sheet of the code, the code can run, given the number r of regression coefficients to be estimated, the number $N' = n - r + 1$ of equations, and the number of future terms in the sequence, usually 1–2 that one would like to predict. Then the code applies the least-squares algorithm of [10] related to the notion of the pseudo-inverse matrix, in order to compute the optimal regression coefficients. The theoretical considerations justifying the procedure can be found in [10] and are reproduced, in part, in Section 2.

The code presents the regression coefficients as well as the estimated future values of the sequence. The code also gives the absolute error (real value minus predicted value) and the relative error (absolute error over real value). The absolute and relative errors are transformed in appropriate graphics and diagrams making use of the standard options of Excel, see Section 5.

The code presents adequate accuracy and time consumption. For matrices such that $r < 50$ and $N' = n - r + 1 < 50$, the final results are obtained in one or two seconds, whereas this time period only increases up to one minute if r and the number of equations N' are close to 800. We did not address any optimization procedure in order to estimate optimal values for r and N', since our empirical tests did not require this kind of matters. This can be readily done by simple enumeration or linear programming procedures. Regarding the accuracy of computations, it is worth noting that sensitivity of the numerical results with respect to the inputs is very low. It may be easily verified by running the code several times with similar data, so as to be able to compare the output. Moreover, as pointed out in [12], those methods based on the Moore–Penrose pseudo-inverse are theoretically stable.

5. Computational experiments and graphics

The code has been used to estimate the Optimal Predictor in a sequence involving financial data. Despite that several types of data had been used, the results do not reveal significant differences, so we will only report a summary related to the daily and weekly evolution of the stock index *S&P* 500[1] which is known to be probably the most important index all around the world.

Prediction of future prices from recent past prices is not so easy according to the weak version of the Efficient Market Hypothesis [13], which assumes the independence between two random variables reflecting the returns provided by an arbitrary share in two different periods of time with void intersection. However, there is empirical evidence pointing out that the level of efficiency (i.e., the degree of independence between the two random variables above) may not be so high in the real world, which has led to the so-called "*price discovery*" assumption (the possibility to discover some characteristics of the price behavior). For instance, Jegadeesh and Titman [14] find inefficiencies (dependencies) in several stock markets, Hudson et al. [15] obtain profits when applying some technical trading rules in British markets, Kamara and Miller [16] focus on some empirical anomalies when checking the put-call parity of European options, Chen and Knez [17] detect some kind of disintegration and cross market arbitrage between the New York Stock Exchange (NYSE) and the National Association of Securities Dealer Automated Quotation (NASDAQ), Kempf and Korn [18] report some violations of the spot-future parity relationship for some trading systems, Balbás et al. [19] and [20] provide some concrete arbitrage strategies arisen in the Chicago Board of Trade (CBOT) and the Spanish Derivative Market respectively, Cheng et al. [21] predict the financial distress, and Balbás and López [22] report some pricing errors in fixed income markets. Further discussions about this question may be found in Pardo et al. [23].

Our empirical results are consistent with the previous literature, which points out that prediction is easier and performances much better for short periods of time (Jegadeesh and Titman [14] and Hudson et al. [15]). Despite that perfect prediction is obviously infeasible (see Pastor and Stambaugh [2] for an interesting discussion about this idea), our results for daily and weekly data seem to be adequate, in the sense that the committed absolute and relative errors are often negligible. Besides, higher values for r and $N' = n - r + 1$ (the number of regression coefficients and equations respectively) did not improve the quality of prediction, so we recommend to try first the values within the intervals [5,15] for r and [7,20] for $N' = n - r + 1$, where n is the number of observations used for one cycle of forecasting in (5).

Our *daily* data correspond to the period January 6, 2006–March 27, 2008, though we will only report the results for some particular periods in the whole sample. There is nothing special within the reported periods, and other periods are quite similar.

[1] The authors sincerely thank *Welzia Management SGIIC SA*, for several databases.

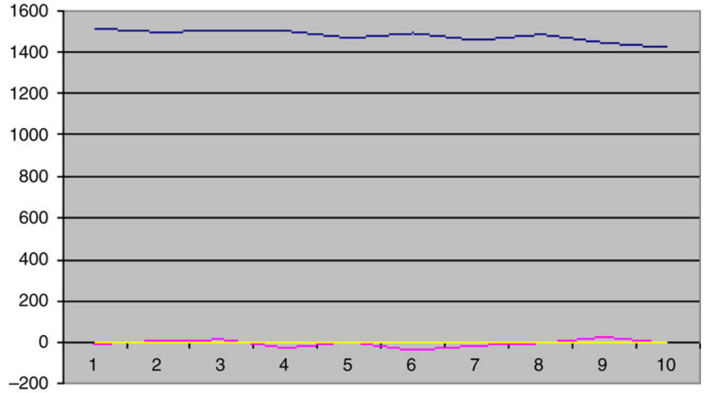

Fig. 1. Daily index real value estimate (top line), absolute error (second line) and relative error (bottom line) during the period November 7–November 20, 2007. On the horizontal axis, the value 1 is associated with November 7, whereas the value 10 corresponds to November 20. Saturdays and Sundays were not considered since the market was closed. Similar comments apply for the remaining figures illustrating the index evolution and the committed errors.

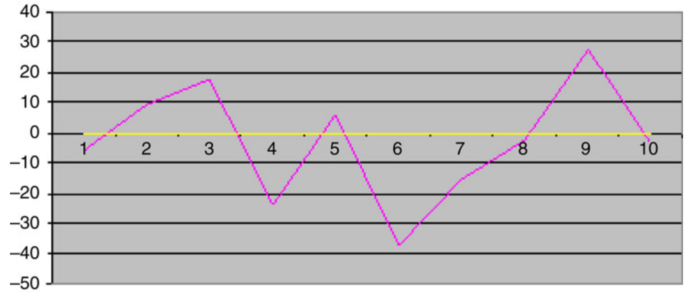

Fig. 2. Magnified 40 times daily absolute and relative errors during the period November 7–November 20, 2007.

Figs. 1–3 present the daily results for the period November 7–November 20, 2007. We have taken $r = 5$ and $N' = 7$ in order to run the code and compute the predicted values (estimates), the errors, and the regression coefficients (model dynamics). This means that the base for prediction comprised the 5 raw data *before* November 7 (the first predicted value in Fig. 1) with sliding by one day until November 12; the last estimate was computed for November 20 with the last base starting on November 14, — according to Eqs. (3)–(5) with $n = N' + r - 1 = 11$ (the length of the cycle). Each base comprised only raw data, and estimates (predictions) were compared with raw data too for model validation. Fig. 1 provides the index real value estimate (top line), the absolute error (real value minus predicted value, second line) and the relative error (absolute error over real index value, bottom line). In order to see more clearly the difference between the absolute and the relative error, we magnified the vertical scale, so that the index real value estimate is out of Fig. 2, and the scale in the vertical axis becomes large enough to see the oscillations that are always present in financial data.

Comment. Figs. 1 and 2 show that errors are within the interval (−2.3%, +1.8%). Fig. 3 shows that a_i of (3) are within (0.1, 0.33), with recent values exerting more influence.

Figs. 4 and 5 below represent the results for the period February 6–February 19, 2008, obtained with $r = 10$ and $N' = 15$. As can be seen from Figs. 4 and 5, the errors are negligible, around 0.6% with respect to the real values of the index.

For the second period February 6–February 19, 2008, we present the regression coefficients $X = (a_1, a_2, \ldots, a_r)^{\mathrm{T}}$ in Fig. 6.

Comment. Fig. 6 shows that only the last regression coefficient $a_{10} \cong 0.9$ is of importance which means that the order r, and possibly also N', could have been taken much smaller.

Daily estimates in Figs. 1 and 4 (top lines) can be used for long term decisions, whereas magnified oscillations are helpful in daily trading since they reveal that fallen prices may soon rebound and indicate the frequency and the shape of oscillations.

Regarding weekly data, the performance of the method may depend on the choice of samples for the bases. To this effect, we present below, Figs. 7–9, the empirical results for the period June 30–September 1, 2006, with parameters $r = 10$, $N' = 20$, and bases formed by the values of the index on Fridays only. The reader can see a different behavior:

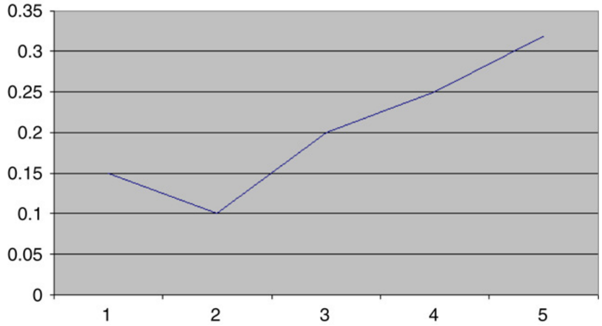

Fig. 3. Regression coefficients for daily prediction during the period November 7–November 20, 2007. On the horizontal axis we have the value of the subscript $i = 1, 2, \ldots, r$, while the vertical axis provides the value of a_i, associated to coordinates of the vector X in (5).

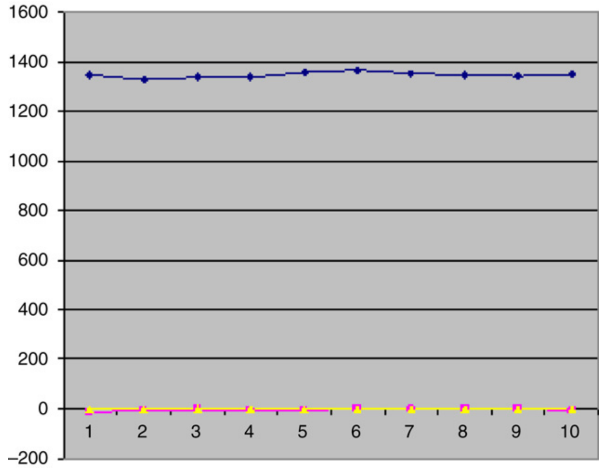

Fig. 4. Daily index real value estimate (top line), absolute error (second line) and relative error (bottom line) during the period February 6–February 19, 2008.

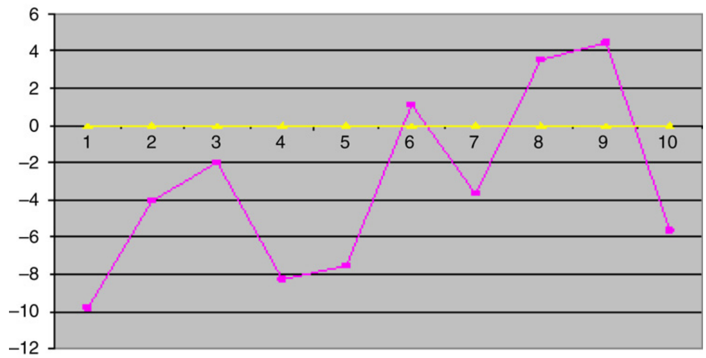

Fig. 5. Magnified 160 times absolute and relative errors during the period February 6–February 19, 2008.

errors within 8%–30% (Figs. 7 and 8), and regression coefficients a_i jumping from -4 to $+4$ (Fig. 9), asserting to instability of such one day samples.

However the errors become again negligible, within -3% to $+1.2\%$, if we take weekly average values rather than index values on a concrete day of the week for the same period as above, Figs. 10 and 11, with a_i from -0.6 to $+1.2$, on Fig. 12 showing the influence of each week on the final estimate. Our database represents the period of January 20, 1994–December 22, 2006, and this finding is consistent with those of previous papers empirically testing the fulfillment of the Efficient Market Hypothesis, see [13–15].

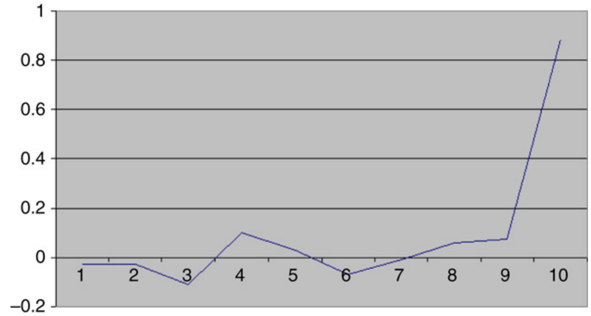

Fig. 6. Regression coefficients for daily prediction during the period February 6–February 19, 2008. On the horizontal axis we have the value of the subscript $i = 1, 2, \ldots, r$, while the vertical axis provides the value of a_i, associated to coordinates of the vector X in (5).

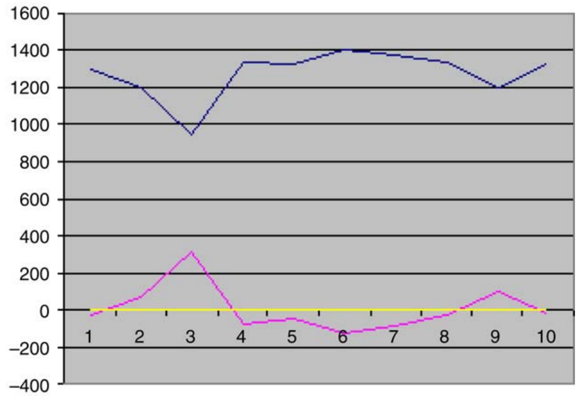

Fig. 7. Weekly data. Index real value estimate (top line), absolute error (second line) and relative error (bottom line) during the period June 30–September 1, 2006. On the horizontal axis the value 1 is associated with June 30, whereas the value 10 corresponds to September 1.

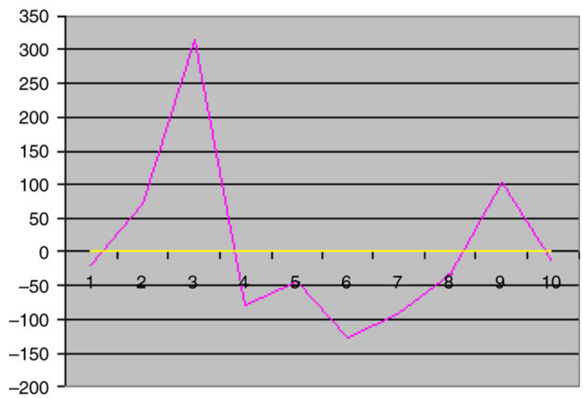

Fig. 8. Weekly data. Magnified 5 times absolute and relative errors during the period June 30–September 1, 2006.

6. Conclusions

The novelty of the method presented in this paper consists in the sliding model approach based on the known past history given in observations, without preconceived statistical and/or econometric designs related to the time series of financial data. The approach exploits linear differential and/or difference equations which provide much broader pattern of behavior than standard statistical and econometric methods.

A recent segment of the available sequence, say 30%, is reserved solely for verification and comparison purposes. It is not used in construction of a model, and on the basis of preceding 70% of raw data the optimal sliding model is devised using the min-norm least-squares solution provided by the Moore–Penrose pseudo-inverse matrix.

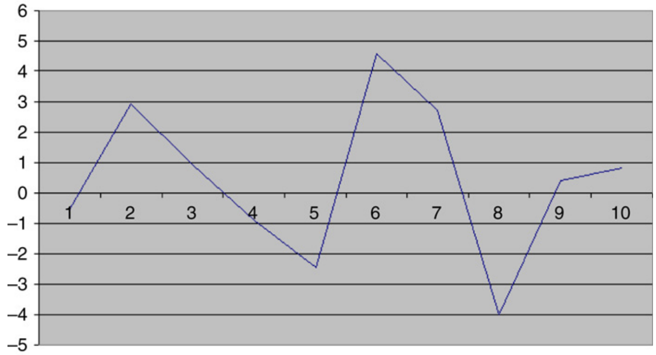

Fig. 9. Regression coefficients for the weekly prediction during the period June 30–September 1, 2006. On the horizontal axis we have the value of the subscript $i = 1, 2, \ldots, r$, while the vertical axis provides the value of a_i, associated to coordinates of the vector X in (5).

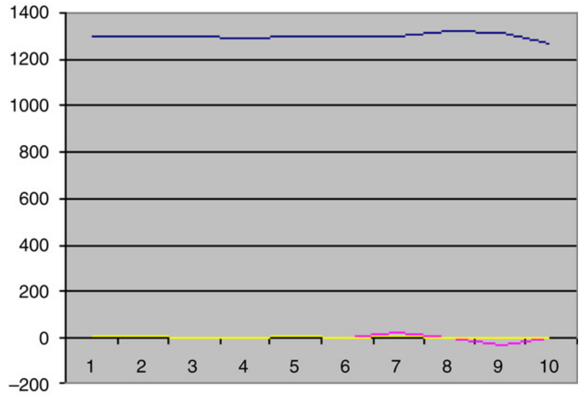

Fig. 10. Average weekly data. Index average value estimate (top line), absolute error (second line) and relative error (bottom line) during the period June 30–September 7, 2006. On the horizontal axis the value 1 is associated with the week June 30–July 6, whereas the value 10 corresponds to week September 1–September 7.

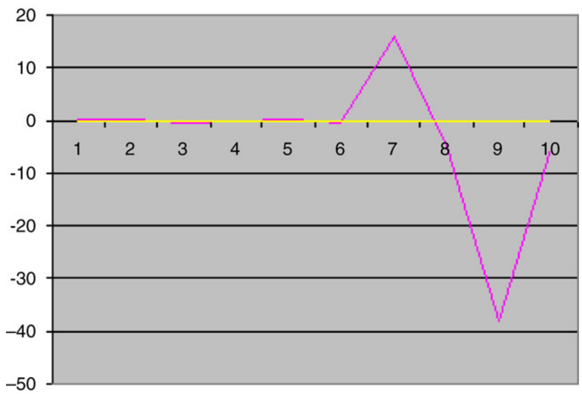

Fig. 11. Average weekly data. Magnified 37 times absolute and relative errors during the period June 30–September 7, 2006.

In the paper, the adequate software is developed and tested for several real time series. Results related to the American stock index *S&P* 500 are reported, and other empirical findings are quite similar. The empirical results confirm the validity and efficiency of the non-statistical sliding model approach to financial markets and reveal adequate level of accuracy and low sensitivity with respect to possible errors in the input data.

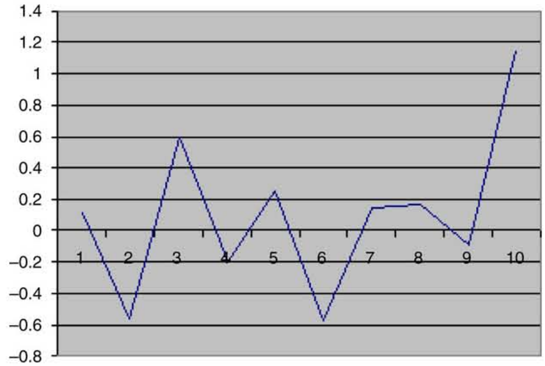

Fig. 12. Regression coefficients for the weekly average prediction during the period June 30–September 7, 2006. On the horizontal axis we have the value of the subscript $i = 1, 2, \ldots, r$, while the vertical axis provides the value of a_i, associated to coordinates of the vector X in (5).

Acknowledgements

The research of first author was partially developed during the visit of Alejandro Balbás and Beatriz Balbás to Concordia University (Montréal, Québec, Canada). These authors would like to thank the Department of Mathematics and Statistics' great hospitality, in particular José Garrido and Yogendra Chaubey.

Alejandro Balbás also thanks the partial support provided by *Welzia Management SGIIC SA*, *RD_Sistemas SA*, *Comunidad Autónoma de Madrid* (Spain), Grant s-0505/tic/000230, and *MEyC* (Spain), Grant SEJ2006-15401-C04-03.

Appendix. Visual basic code in forecasting *(comments in italics)*

Initial sentence
  Sub Deterministic_Regression()

Auxiliary variables
  Dim i As Long, j As Long, k As Long, h As Long, ll As Long
  Dim Cont As Double, Cont2 As Double, Cont3 As Double, Cont4 As Double, Cont5 As Double

$r(1)$ is the number of independent observations in the regression (the base length)
s is the number of simultaneous equations in system (5), Section 2.1
$X = \{y_i, y_{i+1}, \ldots, y_{i+r-1}\}$, $i = j, j+1, \ldots$, $\dim X = r$, *is the row vector of sliding observations in* (2), (3)
at right, - not the same notation as in (5)
A is the vector of coefficients a_i in the regression (3), (5)
  Dim $r()$ As Integer: ReDim $r(2)$: Dim s As Integer
  $r(1)$ = InputBox("*How many independent variables?:* ")
  s = InputBox("*How many equations?:* ")
  Dim $X()$ As Double, $A()$ As Double
  ReDim X(10000): ReDim A(r(1))

Download form Excel $X(1), \ldots X(r+s), \ldots$
  For $i = 1$ To 10000
    $X(i)$ = Cells$(10 + i, 2)$
    Next i

Construct the matrix XU, such that $(XU)(A)t = (X(r+1), \ldots X(r+s))t$. t = "transpose"
  Dim $XU()$ As Double: ReDim $XU(s, r(1))$
  For $i = 1$ To s
    For $j = 1$ To $r(1)$
      $XU(i, j) = X(i + j - 1)$
    Next j
  Next i

```
Construct the matrix XD = (X(r(1) + 1), ... X(r(1) + s))t, t = "transpose"
  Dim XD() As Double: ReDim XD(s)
  For i = 1 To s: XD(i) = X(r(1) + i): Next i
Square (r(1), r(1)) system leading to the orthogonal projection of XD
Z(A)t = Z(−, 0) is called System_1 and indicates that (XU)(A)t is the orthogonal projection of XD
  Dim Z() As Double: ReDim Z(r(1), r(1))
  For i = 1 To r(1)
    For j = 1 To s
      Z(i, 0) = Z(i, 0) + XD(j) * XU(j, i)
    Next j
    For j = 1 To r(1)
      For k = 1 To s
        Z(i, j) = Z(i, j) + XU(k, i) * XU(k, j)
      Next k
    Next j
  Next i

System_2 computes the kernel of XU, (XU)(A)t = 0
Making system_2 diagonal
  For i = 1 To r(1) : XU(0, i) = i: Next i
  Cont = 0
  ll = s
  If ll > r(1) Then ll = r(1)
  For i = 1 To ll
    k = i
    Do While k ⇐ r(1) And Cont = 0
      For j = i To s
        If XU(j, k) <> 0 Then Cont = Cont + 1
      Next j

      If Cont <> 0 Then
        For h = 0 To s
          Cont2 = XU(h, i) : XU(h, i) = XU(h, k) : XU(h, k) = Cont2
        Next h
      End If

      k = k + 1
    Loop
    j = i
    Do While j < s And XU(j, i) = 0
      j = j + 1
    Loop

    For k = 1 To r(1)
      Cont2 = XU(i, k) : XU(i, k) = XU(j, k) : XU(j, k) = Cont2
    Next k

    Cont3 = XU(i, i)
    If Cont3 <> 0 Then
      For k = 1 To r(1)
        XU(i, k) = XU(i, k)/Cont3
      Next k
      j = i + 1
      Do While j ⇐ s
        Cont2 = XU(j, i)
        For k = 1 To r(1)
          XU(j, k) = XU(j, k) − Cont2 * XU(i, k)
        Next k
        j = j + 1
      Loop
    End If
```

```
    Cont = 0
  Next i
```

Dimension of the Kernel. s-ll will denote the required dimension

```
  Cont = 0: Cont2 = 0
  i = s
  Do While Cont = 0 And i >= 1
    For j = 1 To r(1)
        If XU(i, j) <> 0 Then Cont = Cont + 1
    Next j
    If Cont = 0 Then Cont2 = Cont2 + 1
    i = i − 1
  Loop
  ll = Cont2
```

Making the diagonal of the new (XU) equal one

```
  For i = 1 To s − ll
    Cont4 = XU(i, i)
    For j = 1 To r(1)
        XU(i, j) = XU(i, j)/Cont4
    Next j
  Next i
```

Making terms over the diagonal of the new (XU) vanish

```
  For i = s − ll To 1 Step-1
    k = i − 1
    Do While k > 0
        Cont = XU(k, i)
        For j = 1 To r(1)
          XU(k, j) = XU(k, j) − XU(i, j) ∗ Cont
        Next j
        k = k − 1
    Loop
  Next i
```

Basis of the kernel of (XU)

```
  Dim Basis_K() As Double: ReDim Basis_K(r(1) − s + ll, r(1))
  For j = 1 To r(1)
    Basis_K(0, j) = XU(0, j)
  Next j
  For j = s − ll + 1 To r(1)
    For i = 1 To r(1) − (s − ll)
        If i = j − (s − ll) Then Basis_K(i, j) = 1
    Next i
  Next j

  For i = 1 To r(1) − (s − ll)
    For j = 1 To s − ll
        Basis_K(i, j) = −XU(j, s − ll + i)
    Next j
  Next i
```

Reorganizing Basis_K() so as to have the natural order A1, . . . , Ar

```
  For j = 1 To r(1)
    i = j
    Do While Basis_K(0, i) <> j
        i = i + 1
    Loop

    For k = 0 To r(1) − (s − ll)
        Cont = Basis_K(k, j): Basis_K(k, j) = Basis_K(k, i): Basis_K(k, i) = Cont
    Next k
  Next j
```

System_3 is obtained by adjoining system_1 and system involving Basis_K(). System_3 simultaneously imposes (XU)(A)t to be the orthogonal projection of (XD) and (A) to be orthogonal to the kernel

```
Dim ZZ() As Double: ReDim ZZ(2 * r(1) − (s − ll), r(1))
For j = 1 To r(1): ZZ(0, j) = j: Next j

For i = 1 To r(1)
   For j = 0 To r(1)
      ZZ(i, j) = Z(i, j)
   Next j
Next i

For i = r(1) + 1 To 2 * r(1) − (s − ll)
   For j = 1 To r(1)
      ZZ(i, j) = Basis_K(i − r(1), j)
   Next j
Next i
```

Making system_3 diagonal. The system has a unique solution, so not needed rows are deleted

```
For i = 1 To r(1): ZZ(0, i) = i: Next i
Cont = 0
i = 1
Do While i <= r(1)
   k = i
   Do While k <= r(1) And Cont = 0
      For j = i To 2 * r(1) − (s − ll)
         If ZZ(j, k) <> 0 Then Cont = Cont + 1
      Next j

      If Cont <> 0 Then
         For h = 0 To 2 * r(1) − (s − ll)
            Cont2 = ZZ(h, i) : ZZ(h, i) = ZZ(h, k) : ZZ(h, k) = Cont2
         Next h
      End If

      k = k + 1
   Loop

   j = i
   Do While j < 2 * r(1) − (s − ll) And ZZ(j, i) = 0
      j = j + 1
   Loop
   For k = 0 To r(1)
      Cont2 = ZZ(i, k) : ZZ(i, k) = ZZ(j, k) : ZZ(j, k) = Cont2
   Next k

   Cont2 = ZZ(i, i)
   For k = 0 To r(1)
      ZZ(i, k) = ZZ(i, k)/Cont2
   Next k

   j = i + 1
   Do While j <= 2 * r(1) − (s − ll)
      Cont2 = ZZ(j, i)
      For k = 0 To r(1)
         ZZ(j, k) = ZZ(j, k) − Cont2 * ZZ(i, k)
      Next k
      j = j + 1
   Loop

   Cont = 0
   i = i + 1
Loop
```

Solving System_3

```
For i = r(1) To 1 Step -1
  A(i) = ZZ(i, 0)
  j = r(1)
  Do While j > i
    A(i) = A(i) - ZZ(i, j) * A(j)
    j = j - 1
  Loop
Next i
```

Organizing system_3 to retrieve the natural order A(1), A(2), ...

```
For j = 1 To r(1)
  i = j
  Do While ZZ(0, i) <> j
    i = i + 1
  Loop

  For k = 0 To r(1)
    Cont = ZZ(k, j) : ZZ(k, j) = ZZ(k, i) : ZZ(k, i) = Cont
  Next k
Next j
```

Vector A (regression coefficients)

```
For i = 1 To r(1)
  Cells(10 + i, 5) = A(i)
Next i
```

Y, state variable estimate, and the deviations X − Y and (X − Y)/X
Variable rr is the horizon of forecasting

```
Dim Y() As Double: ReDim Y(10000)
Dim rr As Long
rr = InputBox("rr, Horizon of forecasting?: ")
For i = 1 To rr
  For j = 1 To r(1)
    Y(r(1) + i) = Y(r(1) + i) + A(j) * X(j + i - 1)
  Next j

  Cells(r(1) + i + 10, 8) = Y(r(1) + i)
  Cells(r(1) + i + 10, 9) = X(r(1) + i) - Y(r(1) + i)
  If X(r(1) + i) <> 0 Then Cells(r(1) + i + 10, 10) = (X(r(1) + i) - Y(r(1) + i))/X(r(1) + i)
Next i
```

Excel allows us to compose figures with the regression coefficients, predictions, absolute and relative errors, etc. To do this, make a Table, then click the graphic menu in Excel

End of the code

```
End Sub
```

References

[1] R.T. O'Connell, B.L. Bowerman, A.B. Koehler, Forecasting, Time Series, and Regression: An Applied Approach, Duxbury Press, 2005.
[2] L. Pastor, R. Stambaugh, Predictive systems: Living with imperfect predictors, Working Paper, Chicago GBS, 2006.
[3] A. Ben Israel, T.N.E. Greville, Generalized Inverses: Theory and Applications, John Wiley & Sons, New York, 1974.
[4] E.H. Moore, On the reciprocal of the general algebraic matrix (Abstract), Bull. Amer. Math. Soc. 26 (1920) 394–395.
[5] E.H. Moore, General analysis, Mem. Amer. Philos. Soc. 1 (1935) 147–209.
[6] R. Penrose, A generalized inverse for matrices, Proc. Cambridge Phylos. Soc. 51 (1955) 406–413.
[7] R. Penrose, On best approximate solution of linear matrix equations, Proc. Cambridge Phylos. Soc. 52 (1956) 17–19.
[8] E.A. Galperin, A. Zinger, Stationarity conditions for linear models, Int. J. Math. Modelling 4 (1983) 501–514.
[9] E.A. Galperin, R. Labonté, Problème d'identification de modèles de régression et sa solution par la programmation linéaire, Ann. Sci. Math. Québec 8 (1984) 29–53.
[10] E. Galperin, Deterministic regression models for prediction and control, Int. J. Math. Modelling 4 (1985) 157–171.
[11] R.C.K. Lee, Optimal estimation, identification and control, in: Research Monograph, vol. 28, MIT Press, 1964.
[12] L. Lin, T.T. Lu, Y. Wei, On level-2 condition number for the weighted Moore–Penrose inverse, Comput. Math. Appl. 55 (2008) 788–800.
[13] E. Fama, Market efficiency, long-term returns and behavioral finance, J. Fin. Econ. 49 (3) (1998) 283–306.
[14] N. Jegadeesh, S. Titman, Returns to buying winners and selling losers: Implications for stock market efficiency, J. Financ. 48 (1993) 65–91.

[15] R. Hudson, M. Dempsey, K. Keasey, A note on the weak efficiency of capital markets: The application of simple technical trading rules to UK stock prices 1935 to 1994, J. Bank. Financ. 20 (1996) 1121–1132.
[16] A. Kamara, T.W. Miller, Daily and intradaily tests of European put-call parity, J. Financ. Quant. Anal. 30 (1995) 519–541.
[17] Z. Chen, P.J. Knez, Measurement of market integration and arbitrage, The Rev. Financ. Stud. 8 (1995) 563–579.
[18] A. Kempf, O. Korn, Trading system and market integration, J. Financ. Intermed. 7 (1998) 220–239.
[19] A. Balbas, I.R. Longarela, J. Lucia, How financial theory applies to catastrophe-linked derivatives: An empirical test with several pricing models, J. Risk and Insur. 66 (1999) 551–582.
[20] A. Balbás, I.R. Longarela, A. Pardo, Integration and arbitrage in the Spanish financial markets: An empirical approach, J. Futures Mark. 20 (2000) 321–344.
[21] C.B. Cheng, C.L. Chen, C.J. Fu, Financial distress prediction by a radial basis function network with logit analysis learning, Comput. Math. Appl. 51 (2006) 579–588.
[22] A. Balbás, S. López, Sequential arbitrage measurement and interest rate envelopes, J. Optim. Theory Appl. 138 (3) (2008) 361–374.
[23] A. Pardo, A. Balbás, V. Meneu, The effectiveness of several market integration measures when facing market turmoil, Deriv. Use, Trading & Regul. 8 (2003) 345–368.