Evolving Systems (2012) 3:5–18
DOI 10.1007/s12530-011-9042-1

ORIGINAL PAPER

Evolving fuzzy systems for pricing fixed income options

Leandro Maciel · Andre Lemos · Fernando Gomide · Rosangela Ballini

Received: 3 April 2011 / Accepted: 25 October 2011 / Published online: 1 December 2011

Abstract During the recent decades, option pricing became an important topic in computational finance. The main issue is to obtain a model of option prices that reflects price movements observed in the real world. In this paper we address option pricing using an evolving fuzzy system model and Brazilian interest rate options data. Evolving models are particularly appropriate because they gradually develops the model structure and parameters from a stream of data. Therefore, evolving fuzzy models provide a higher level of system adaptation and learns the system dynamics continuously, an essential attribute in pricing options estimation. In particular, we emphasize the use of the evolving participatory learning methods. The participatory evolving models considered in this paper are compared against the traditional Black's closed-form formula, artificial neural networks structures, and alternative evolving fuzzy system approaches reported in the literature. Actual daily data used in the experiments cover the period from January 2003 to June 2008. We measure forecast performance of all models and report the statistical tests done for the competing forecast models. The results show that the participatory evolving fuzzy system modeling approach is effective to estimate prices of fixed income options.

Keywords Evolving fuzzy systems · Option pricing · Neural networks · Interest rate · Derivatives

1 Introduction

One of the most important field in financial engineering is option pricing, particularly in derivatives market. Option pricing is an essential information for players because it provides mechanisms for hedge operations against higher market fluctuations. The term structure of interest rates or yield curve is a basis for most investments, the reason why pricing interest rate derivatives currently attract attention of researches and practitioners.

In the Brazilian derivative markets, one important fixed income instrument are One-Day Interbank Deposit Contract Index (IDI) options. IDI are contracts that reflect the behavior of interest rates between the trade date and option maturity. In USA and European derivative markets, standard interest rate options has as underlying asset a fixed income equity with maturity greater than the option considered. This peculiarity means that IDI becomes option prices, and the factors that affect them, differently from those that follow standard models, demand a particular approach.

Recently Junior et al. (2003), Barbedo et al. (2009), Almeida and Vicente (2006) employed static models based on Black–Scholes (BS) formula (Black and Scholes 1973) and its derivations to price IDI options. Their results have shown that the theoretical prices differ significantly from

L. Maciel · F. Gomide
School of Electrical and Computer Engineering,
University of Campinas, Campinas 13083-852, Brazil
e-mail: maciel@dca.fee.unicamp.br

F. Gomide
e-mail: gomide@dca.fee.unicamp.br

A. Lemos
Department of Electronic Engineering,
Federal University of Minas Gerais,
Belo Horizonte 31270-010, Brazil
e-mail: andrepl@cpdee.ufmg.br

R. Ballini (✉)
Institute of Economics, University of Campinas,
Campinas 13083-857, Brazil
e-mail: ballini@eco.unicamp.br

actual market prices, despite considering the interest rate options peculiarities.

The BS formula can be put in closed-form and is based on the following explicit assumptions: (a) it is possible to borrow and lend cash at a known constant risk-free interest rate; (b) the stock price follows a geometric Brownian motion with constant drift and volatility; (c) there are no transaction costs, taxes or bid-ask spread; (d) the underlying security does not pay a dividend; (e) there is no arbitrage opportunity; (f) it is possible to buy any fraction of a share, and (g) there are no restrictions on short selling.[1] Several of these original assumptions have been removed in subsequent extensions of the model, but they cause misspecification when the prices estimated are compared against actual prices. Moreover, one must realize that option pricing occurs daily, and BS formula does not take past information into account. Input variables are time series and models based on static formulas are not able to capture the temporal dependence of the series involved. Models that describe the dynamical properties of the series are more suitable for forecasting. Therefore, static models like BS and its variations, with more restrictive assumptions, do not consider temporal information associated with the variables that determine the price of an option. These features suggest the need to develop more accurate option pricing models to overcome temporal restrictions and to consider information contained in data.

As option pricing theory typically gives non-linear relations between option prices and the variables that determine them, a highly flexible model is required to capture the empirical pricing mechanism. Computational intelligence (CI) methods based on Artificial Neural Networks (ANN) and Fuzzy Systems are particularly useful for this purpose due to their ability to handle complex systems. During the last decades, several researchers have attempted ANN for option pricing (Gençay and Qi 2001 to Maciel and Ballini 2010). In general, their results show that neural networks models outperform traditional option pricing models.

The use of Fuzzy Set Theory in option pricing became a new research field in financial engineering. For instance, Wu (2005) uses fuzzy sets and the BS formula considering the risk-free interest rate, volatility, and asset price to price European options. This allows financial analysts to choose the European price with an acceptance degree.

A fuzzy pricing model of currency options was addressed in Liu (2009). Here the option price is a fuzzy number. The author shows that the model helps financial investors to pick any currency option price with an acceptable degree for later use. The approach is useful to handle the imprecise nature of financial environments.

Considering the main classes of stochastic volatility models (the endogenous and exogenous sources of risk), Figa-Talamanca and Guerra (2009) generalized the Black–Scholes option valuation model. In this case, the option price is also a fuzzy number which, according to the authors, helps to reduce price misspecifications associated with uncertainty and vagueness. More recently, Leu (2010) introduced a fuzzy time series-based neural network (FTSNN), a hybrid approach composed of a fuzzy time series model and a neural network model to price options. The results show that FTSNN outperforms many existing methods in terms of distinct error measures.

The purpose of this paper is to develop and test evolving fuzzy rule-based models to price IDI call options traded on the Securities, Commodities and Future Exchange (BM&FBOVESPA).

The concept of evolving fuzzy systems introduces the idea of gradual self-organization and parameter learning in fuzzy rule-based models (Angelov 2002). Evolving fuzzy systems use data streams to continuously adapt the structure and functionality of fuzzy rule-based models. The evolving mechanism ensures greater generality of the structural changes because rules are able to describe a number of data samples. Evolving fuzzy rule-based models include mechanisms for rule modification to replace a less informative rule by a more informative one (Angelov 2002). Overall, the evolving mechanism guarantees gradual change of the rule base structure inheriting structural information. The idea of parameter adaptation of rules antecedent and consequent is similar in the framework of evolving connectionists systems (Kasabov and Song 2002), evolving Takagi-Sugeno (eTS) and extended Takagi-Sugeno (xTS) models, and their variations (Angelov 2002; Angelov and Filev 2004; Angelov and Filev 2005; Angelov and Zhou 2006a). In particular the eTS model is a functional fuzzy model in the Takagi-Sugeno (TS) form whose rule base and parameters continually evolve by adding new rules with higher summarization power and modifying existing rules and parameters to match current knowledge.

The evolving fuzzy participatory learning (ePL) modeling was suggested in (Lima et al. 2010b). The approach joins the concept of participatory learning (PL) (Yager 1990) with the evolving fuzzy modeling idea (Angelov 2002; Angelov and Filev 2004). In evolving systems the PL concept is viewed as an unsupervised clustering algorithm (Silva et al. 2005) and is a natural candidate to find rule base structures in dynamic environments. Here we focus on functional fuzzy (TS) models. Similarly as in eTS,

[1] The Black model (Black 1976) considered in this work follows the Black–Scholes model exactly, except for the assumption that the spot price is a log-normal process which is replaced by the assumption that the forward price at maturity of the option is log-normally distributed. The derivation is identical and the final formula is the same except that the spot price is replaced by the forward price. The forward price represents the undiscounted expected future value.

structure identification and self-organization in ePL means estimation of the focal points of the rules, the antecedents parameters, except that ePL uses participatory learning fuzzy clustering instead of scattering, density or information potential. With the antecedent parameters fixed, the remaining TS model parameters can be found using least squares methods (Chiu 1994; Ljung 1999b). The evolving fuzzy participatory learning captures the rule base structure at each step using convex combinations of new data samples and the closest cluster center, the focal point of a rule. After, the rule base structure is updated and, similarly as in eTS, the parameters of the rule consequents are computed using the recursive least squares algorithm.

Recently, a new class ePL model has been developed exploring fuzzy rule-based systems with multivariable Gaussian membership functions, namely, the evolving Multivariable Gaussian (eMG) (Lemos et al. 2011). This model consider the possibility that input variables may interact with each other, avoids the *curse of dimensionality*, and introduces a more sound and systematic approach for learning. The result is an more robust algorithm with less parameters.

Evolving fuzzy models are inherently adaptive and particularly appropriate for pricing options because dynamic markets require rapid information processing and accurate results, as reflected in investment decisions, form portfolios and hedging strategies. Moreover, evolving models require only the latest data, which results in more efficient processing and the capability to capture option prices movements in the derivatives market.

After this introduction, this paper proceeds as follows. Section 2 briefly reviews the idea of evolving fuzzy rule-based modeling and the basic eTS method. Next, Sect. 3 introduces the concept of participatory learning, the features of the evolving fuzzy participatory learning (ePL) method and its computational details. Section 4 describes the eMG, an extension ePL model. Section 5 explains IDI contracts, the model of Black, and alternative neural network models. Section 6 compares the evolving fuzzy models addressed in this paper against Black model, neural network and alternative evolving fuzzy models using actual Brazilian financial market data from 2003 to 2008. Finally, Sect. 7 concludes the paper summarizing its contributions and suggesting issues for further investigation.

2 Evolving fuzzy systems

When learning models online, data are collected and processed continuously. New data may confirm and reinforce the current model if data is compatible with existing knowledge. Otherwise, new data may suggest changes and a need to review the model. For instance, this is the case when modeling systems whose operating conditions modify, fault occurs, or parameters of the processes change.

In evolving systems, a key question is how modify the current model structure using the newest data sample. Evolving systems use incoming data to continuously develop their structure and functionality through online self-organization.

Fuzzy rule-based models whose rules are endowed with local models forming their consequents are commonly referred to as fuzzy functional models. The Takagi-Sugeno (TS) is a typical example of a fuzzy functional model. A particularly important case is when rule consequents are linear functions. The evolving Takagi-Sugeno (eTS) model and its variations (Angelov and Filev 2004) assume rule-based models whose fuzzy rules are as follows

$$\mathcal{R}_i : \mathrm{IF}\, x\, \mathrm{is}\, \Gamma_i \quad \mathrm{THEN} \quad y_i = \gamma_{i0} + \sum_{j=1}^{m} \gamma_{ij} x_j \quad i = 1, \ldots, c^k \tag{1}$$

where

$\mathcal{R}_i$ ith fuzzy rule
c^k number of fuzzy rules at $k; k = 0, 1, \ldots$
$x \in [0,1]^m$ input data
y_i output of the ith rule
Γ_i vector of antecedents fuzzy sets
γ_{ij} parameters of the consequent

The collection of the rules assembles a model as a combination of local linear models. The contribution of a local linear model to the overall output is proportional to the degree of firing of each rule. eTS uses antecedent fuzzy sets with Gaussian membership functions:

$$\mu_i = e^{-r||x^k - v_i||^2} \tag{2}$$

where r is a positive constant which defines the zone of influence of ith local model and v_i is the respective cluster center, the focal point, $i = 1, \ldots, c^k$.

Online learning with eTS needs online clustering to find cluster centers, assumes gradual changes of the rule base, and uses a recursive least squares to compute the consequent parameters. Each cluster defines a rule.

The TS model output at k is found as the weighted average of the individual rule contributions as follows:

$$y^k = \frac{\sum_{i=1}^{c^k} \mu_i y_i}{\sum_{i=1}^{c^k} \mu_i} \tag{3}$$

where c^k is the number of rules after k observations and y_i the output of ith rule at k.

Clustering starts with the first data point as the center of the first cluster. The procedure is a form of subtractive clustering, a variation of the Filev and Yager mountain clustering approach (Yager and Filev 1994). The capability of a point to become a cluster center is evaluated through its potential. Data potentials are calculated recursively using Cauchy function to measure the potential. If the potential of a new data is higher than the potential of the current cluster centers, then the new data becomes a new cluster center and a new rule is created. If the potential of a new data is higher than the potential of the current centers, but it is close to an existing center, then the new data replaces the existing center. See Angelov (2002) and Angelov and Filev (2004) for more details. Current implementations of eTS adopt Cauchy functions to define the notion of data density evaluated around the last data of a data stream, monitors the clustering step, and contains several mechanisms to improve model efficiency such as online structure simplification.

The eXtended Takagi-Sugeno (xTS) fuzzy system, developed by Angelov and Zhou (2006a), introduces the idea of an daptive recursively updated radius of the clusters (zone of the influence of the fuzzy rules) that learns the data distribution/variance/scatter in each cluster and a new condition to replace clusters that excludes contradictory rules as the main differences from eTS model.

The clustering method of xTS model is based on the recursive calculation of the potential of the new data point, as in eTS model. However, xTS does not consider a constant radius/spread, r. In eTS model the spread is a different, fixed value for each input variable. During clusters updating process, if a new data point has higher potential than the existing cluster centers, then a new cluster is created centered at the new point. Otherwise, the cluster structure does not change. This mechanism is similar in xTS model, but if the condition mentioned above holds and the new point is well represented by existing cluster centers, then the closest cluster center is replaced by this new point. Details about xTS models is found in Angelov and Zhou (2006a).

3 Evolving fuzzy participatory learning

Evolving fuzzy participatory learning (ePL) modeling adopts the same philosophy as eTS. After the initialization phase, data processing is performed at each step to verify if a new cluster must be created, if an old cluster should be modified to account for the new data, or if redundant clusters must be eliminated. Cluster centers are the focal point of the rules. Each rule corresponds to a cluster. Parameters of the consequent functions are computed using the local recursive least squares algorithm. In this paper we assume, without loss of generality, linear consequent functions.

The main difference between ePL and eTS concerns the procedure to update the rule base structure. Differently from eTS, ePL uses a compatibility measure to determine the proximity between new data and the existing rule base structure. The rule base structure is isomorphic to the cluster structure because each rule is associated with a cluster. Participatory learning assumes that learning depends on what the system already knows about the model. Therefore, in ePL, the current model is part of the evolving process itself and influences the way in which new observations are used for self-organization. An essential property of participatory learning is that the impact of new data in causing self-organization or model revision depends on its compatibility with the current rule base structure, or equivalently, on its compatibility with the current cluster structure.

3.1 Participatory learning

Let $v_i^k \in [0,1]^m$ be a variable that encodes the ith $(i = 1,\ldots,c^k)$ cluster center at the kth step. The aim of the participatory mechanism is to learn the value of v_i^k, using a stream of data $x^k \in [0,1]^m$. In other words, each $x^k, k = 1,\ldots,$ is used as a vehicle to learn about v_i^k. We say that the learning process is participatory if the contribution of each data x^k to the learning process depends upon its acceptance by the current estimate of v_i^k as being valid. Implicit in this idea is that, to be useful and to contribute to the learning of v_i^k, observations x^k must somehow be compatible with current estimates of v_i^k.

In ePL, the object of learning is a cluster structure. Cluster structure is defined by a set of cluster centers (focal points, prototypes). More formally, given an initial cluster structure, a set of vectors $v_i^k \in [0,1]^m$, $i = 1,\ldots,c^k$, is updated using a compatibility measure, $\rho_i^k \in [0,1]$ and an arousal index, $a_i^k \in [0,1]$. While ρ_i^k measures how much a data point is compatible with the current cluster structure, the arousal index a_i^k acts as a critic to remind when current cluster structure should be revised in front of new information contained in data. Figure 1 summarizes the main

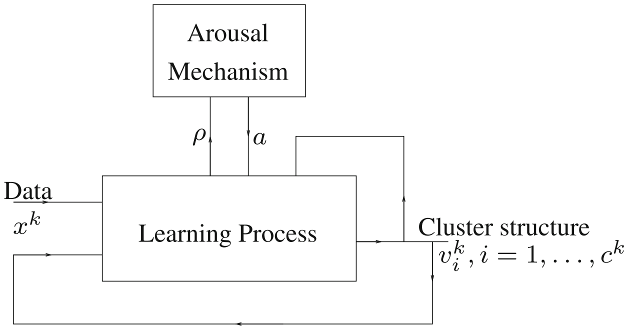

Fig. 1 Participatory learning

constituents and functioning of the participatory learning approach.

Due to its unsupervised, self-organizing nature, the PL clustering procedure may create a new cluster or modify the existing ones at each step. If the arousal index is greater than a threshold value $\tau \in [0, 1]$, then a new cluster is created. Otherwise, the ith cluster center, the one most compatible with x^k, is adjusted as follows:

$$v_i^{k+1} = v_i^k + G_i^k (x^k - v_i^k) \tag{4}$$

where

$$G_i^k = \alpha \rho_i^k \tag{5}$$

$\alpha \in [0, 1]$ is the primary learning rate, and

$$\rho_i^k = 1 - \frac{||x^k - v_i^k||}{m} \tag{6}$$

with $|| \cdot ||$ a norm, m the dimension of input space, and

$$i = \arg\max_j \{\rho_j^k\} \tag{7}$$

Notice that the ith cluster center is a convex combination of the new data sample x^k and the closest cluster center.

Similarly as (4), the arousal index a_i^k is updated as follows:

$$a_i^{k+1} = a_i^k + \beta(1 - \rho_i^{k+1} - a_i^k) \tag{8}$$

The value of $\beta \in [0, 1]$ controls the rate of change of arousal: the closer β is to one, the faster the system is to sense compatibility variations.

The way in which ePL considers the arousal mechanism is to incorporate the arousal index (8) into (5). Here we assume

$$G_i^k = \alpha(\rho_i^k)^{1-a_i^k} \tag{9}$$

When $a_i^k = 0$, we have $G_i^k = \alpha\rho_i^k$ which is the PL procedure with no arousal. Notice that if the arousal index increases, the similarity measure has a reduced effect. The arousal index can be interpreted as the complement of the confidence we have in the truth of the current belief, the rule base structure. The arousal mechanism monitors the performance of the system by observing the compatibility of the current model with the observations. Therefore learning is dynamic in the sense that (4) can be viewed as a belief revision strategy whose effective learning rate (9) depends on the compatibility between new data, the current cluster structure, and on model confidence as well.

Notice also that the learning rate is modulated by compatibility. In conventional learning models, there are no participatory considerations and the learning rate is usually set small to avoid undesirable oscillations due to spurious values of data that are far from cluster centers. Small values of learning rate while protecting against the influence of noisy data, slow down learning. Participatory learning allows the use of higher values of the learning rate and the compatibility index acts to lower the effective learning rate when large deviations occur. On the contrary, when the compatibility is large, it increases the effective rate, which means speeding up the learning process.

Clearly, whenever a cluster center is updated or a new cluster added, the PL fuzzy clustering procedure should verify if redundant clusters are created. This is because updating a cluster center using (4) may push a given center closer to another one and a redundant cluster may be formed. Therefore a mechanism to exclude redundancy is needed. An alternative is to verify if similar outputs due to distinct rules are produced. In PL clustering, a cluster center is declared redundant whenever its compatibility with another center is greater than or equal to a threshold value θ. If this is the case, then we can either maintain the original cluster center or replace it by the average of the cluster centers. Similarly as in (6), the compatibility index among cluster centers is computed as follows:

$$\rho_{ij}^k = 1 - \sum_{j=1}^{p} |v_i^k - v_j^k| \tag{10}$$

Therefore, if

$$\rho_{ij}^k \geq \theta \tag{11}$$

then the cluster i is declared redundant.

Participatory clustering requires choosing parameters $\alpha, \beta,\ \theta$ and τ. The choice can be pursued as follows. If a given data lead to an increase in the arousal index greater than the threshold $\tau \in [0, 1]$, then this data will be the cluster center of a new cluster (Lima 2008). Consequently, if the cluster with the highest compatibility, s, is updated, then it is the case that $a_s^{k+1} < \tau$. Thus from (8) we get:

$$\begin{aligned} a_s^k + \beta(1 - \rho_s^{k+1} - a_s^k) = (1-\beta)a_s^k + \beta d_s^k < \tau \\ \Rightarrow d_s^k < \frac{\tau - (1-\beta)a_s^k}{\beta} < \frac{\tau}{\beta} \end{aligned} \tag{12}$$

where $d_s^k = d(v_s^k, x^k)$ is the distance between the cluster center s and the data x^k at k.

On the other hand, considering the compatibility measure ρ_{ij}^k, for any two distinct cluster $i,j = 1, \ldots, c^k$, we have:

$$\rho_{ij}^k = \rho_{ij}^k(v_i^k, v_j^k) = 1 - d_{ij}^k < \theta \Rightarrow d_{ij}^k > 1 - \lambda \tag{13}$$

If $d_{ij}^k < 1 - \lambda$, then clusters i and j are considered redundant and redefined as a single cluster. Here $d_{ij}^k = d(v_i^k, v_j^k)$ is the distance between the centers v_i^k and v_j^k.

From expressions (12) and (13), to ensure that a new, nonredundant cluster is added, we should choose values of $\beta,\ \theta$ and τ such that:

$$0 < \frac{\tau}{\beta} \leq 1 - \theta \leq 1$$

where

$$\tau \leq \beta(1 - \theta)$$

Analysis of the dynamic behavior of the participatory learning considering the compatibility and arousal mechanisms simultaneously with the learning rate is discussed in Lima et al. (2010a). The primary learning rate α is a small value, typically $\alpha \in [10^{-1}, 10^{-5}]$.

3.2 Parameter estimation

After clustering, the fuzzy rule based is constructed using a similar procedure as eTS described in Sect. 2. The cluster centers define the modal values of the Gaussian membership functions while dispersions are chosen to achieve appropriate levels of rule overlapping. Moreover, for each cluster found, the corresponding rule has linear consequent function (see (1)) and its parameters adjusted using the recursive least squares algorithm. Only the rule with highest compatibility index has its consequent updated. The computational details are as follows.

Let $x^k = [x_1^k, x_2^k, \ldots, x_m^k] \in [0,1]^m$ be the vector of observations and $y_i^k \in [0,1]$ the output of the ith rule, $i = \arg\max_j\{\rho_j^k\}$, at $k = 1, 2, \ldots$. Notice that the i-th rule is the one with highest compatibility index. The consequent parameters of the ith rule is estimated using (index i is omitted below for sake of notation simplicity):

$$Y^k = X^k \gamma^k \tag{14}$$

where $\gamma^k \in \mathbb{R}^{m+1}$ is the vector of unknown parameters

$$(\gamma^k)^T = [\gamma_0^k \, \gamma_1^k \, \gamma_2^k \ldots \gamma_m^k]$$

and $X^k = [1 \; x^k] \in [0,1]^{1 \times (m+1)}$ is composed by the k-th input vector x^k and a constant term. $Y^k = [y_i^k]$ is the output vector.

Model (14) gives a local description of the system, but the vector of parameters γ is unknown. One way to estimate the values for γ is to use the data available. Assume that

$$Y^k = X^k \gamma^k + e^k \tag{15}$$

where γ^k represents the parameters to be recursively computed and e^k is the modeling error at k. The least squares algorithm chooses γ^k to minimize a sum of squares errors

$$J_k = J(\gamma^k) = (e^k)^T e^k \tag{16}$$

Define X^{k+1} and Y^{k+1} as follows:

$$X^{k+1} = \begin{pmatrix} X^k \\ 1 \quad x^{k+1} \end{pmatrix}, \qquad Y^{k+1} = \begin{pmatrix} Y^k \\ y_i^{k+1} \end{pmatrix} \tag{17}$$

where x^{k+1} is the current input data and y_i^{k+1} is the corresponding model output. The vector of parameters that minimizes the functional J_k at k is (Young 1984):

$$\gamma^k = P^k b^k \tag{18}$$

where $P^k = [(X^k)^t X^k]^{-1}$ and $b^k = (X^k)^T Y^k$. Using the matrix inversion lemma (Young 1984):

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

and making

$$A = (P^k)^{-1}, \; C = I, \; B = X^{k+1}, \; D = (X^{k+1})^T$$

we get

$$P^{k+1} = P^k \left[I - \frac{X^{k+1}(X^{k+1})^T P^k}{1 + (X^{k+1})^T P^k X^{k+1}} \right] \tag{19}$$

where I is the identity matrix. After simple mathematical transformations, the vector of parameters is computed recursively as follows:

$$\gamma^{k+1} = \gamma^k + P^{k+1} X^{k+1} \left(Y^{k+1} - (X^{k+1})^T \gamma^k \right) \tag{20}$$

Detailed derivation can be also found in Astrom and Wittenmark (1994). For convergence proofs see Johnson (1988) for example. Expression (20) is used to update the rule consequent parameters at each k.

The use of the recursive least squares algorithm depends of the initial values of the parameters $\widehat{\gamma}^0$, and of the initial values of the entries of matrix P^0. These initial values are chosen based on:

1. Existence of previous knowledge about the system, exploring a database to find an initial rule base and, consequently, $\widehat{\gamma}^0$ and P^0.
2. A useful technique when no previous information is available is to choose large values for the entries of matrix P^0 (Wellstead and Zarrop 1995). If the initial values of consequent parameters are similar to exact values, then it is enough to choose small values for the entries of P^0. A standard choice of P^0 is

$$P^0 = sI_m$$

where I_m is the identity matrix of the order m, and m is the number of consequent parameters. The value of s usually is chosen such that $s \in [100, 10000]$ if large values are required, while for the small values $s \in [1, 10]$. More details can be found in Wellstead and Zarrop (1995).

In this paper, we use the first option, that is, we use a database to choose the initial rule base and its parameters.

4 Evolving multivariable Gaussian

In Lemos et al. (2011) a new type for ePL modeling is developed. The evolving Multivariable Gaussian (eMG) uses an evolving Gaussian clustering algorithm also rooted in the concept of participatory learning to define the rule base at each step. However, the clustering procedure considers the possibility that input variables may interact with each other. Clusters are estimated using a normalized distance measure (similar to the *Mahalanobis* distance) and trigger ellipsoidal clusters whose axes are not necessarily parallel to the input variables axes, as it would be the case if the Euclidean distance were used (Kasabov and Song 2002; Lughofer 2008; Angelov and Filev 2004). The idea is to preserve information about interactions between input variables. The fuzzy sets of the rules antecedents are multivariable Gaussian membership functions characterized by a center vector, and a dispersion matrix representing the dispersion of each variable and their interactions. Similarly as in other evolving system modeling approaches (Lughofer 2008; Angelov and Filev 2004), the parameters of the fuzzy rules consequents are updated using weighted recursive least squares.

The eMG model uses membership functions of the form:

$$H(x) = \exp\left[-\frac{1}{2}(x-v)\Sigma^{-1}(x-v)^T\right] \tag{21}$$

where x is an $1 \times m$ input vector, v is the $1 \times m$ center vector and Σ is a $m \times m$ symmetric, positive definite matrix. The center vector v is the modal value and represents the typical element of $H(x)$. The matrix Σ denotes the dispersion and represents the spread of $H(x)$ (Pedrycz and Gomide 2007). Both, v and Σ, are parameters of the membership function to be associated with cluster center and cluster spread, respectively.

Most of evolving fuzzy systems perform clustering in the input or input-output data space, and rules are created using one-dimensional, single variable fuzzy sets which are projections of the clusters on each input variable space. During fuzzy inference, the fuzzy relation induced by the antecedent of each fuzzy rule is computed using an aggregation operator (e.g. a t-norm) and the input fuzzy sets. This approach is commonly used, but it may cause information loss if input variables interact (Kim et al. 1998; Abonyi et al. 2002). For instance, system identification and time series forecasting usually use lagged values of the input and/or output as inputs, and these lagged values tend to be highly related.

To avoid information loss, the algorithm introduced herein uses multivariable, instead of single variable Gaussian membership functions to represent each cluster found by the recursive clustering algorithm. The parameters of the membership functions are extracted directly from the corresponding clusters. These multivariable membership functions use the information about the dispersion matrix of each cluster (estimated by the clustering procedure) and thus provide information about input variables interactions.

4.1 Gaussian participatory evolving clustering

The evolving clustering algorithm used by the eMG model to construct the rule base at each step assumes that the object of learning is, similarly as in the previous section, the cluster structure, i.e, the cluster centers $v_i^k, i = 1, \ldots, c^k$, where c^k is the number of clusters at step k. The shape of the clusters are encoded by a dispersion matrix Σ^k. At each step, the learning process may create a new cluster, modify the parameters of an existing one, or merge two similar clusters.

The cluster structure is updated using a compatibility measure $\rho_i^k \in [0, 1]$ and an arousal index, $a_i^k \in [0, 1]$, similar to ePL. Thresholds are defined for the compatibility measure (T_ρ) and the arousal index (T_a). If at each step the compatibility measure of the current observation is less than the threshold for all clusters, i.e., $\rho_i^k < T_\rho \, \forall \, i = 1, \ldots, c_k$, and the arousal index of the cluster with the greatest compatibility is greater than the threshold, i.e, $a_i^k > T_a$ for $i = \arg\max_j\{\rho_j^k\}$, then a new cluster is created. Otherwise the cluster center with the highest compatibility is adjusted using (4).

The compatibility measure ρ_i^k suggested here uses the squared value of the normalized distance between the new observation and cluster centers (*M-Distance*):

$$M(x^k, v_i^k) = (x^k - v_i^k)(\Sigma_i^k)^{-1}(x^k - v_i^k)^T \tag{22}$$

To compute the *M-Distance*, the dispersion matrix of each cluster Σ_i^k must be estimated at each step. The recursive estimation of the dispersion matrix proceeds as follows:

$$\Sigma_i^{k+1} = (1 - G_i^k)(\Sigma_i^k - G_i^k(x^k - v_i^k)(x^k - v_i^k)^T) \tag{23}$$

The compatibility measure at each step k is given by:

$$\rho_i^k = \exp\left[-\frac{1}{2}M(x^k, v_i^k)\right] \tag{24}$$

To find a threshold value for the compatibility measure, we assume that the values $M(x^k, v_i^k)$ can be modeled by a Chi-Square distribution. Thus, given a significance level λ, the threshold can be computed as follows:

$$T_\rho = \exp\left[-\frac{1}{2}\chi^2_{m,\lambda}\right] \tag{25}$$

where $\chi^2_{m,\lambda}$ is the λ upper unilateral confidence interval of a Chi-Square distribution with m degrees of freedom, where m is the number of inputs.

The compatibility measure is based on a normalized distance measure (22). The corresponding threshold (25) must be adjusted considering the input space dimension to avoid the *curse of dimensionality*. This is because, as the input space dimension increases, the distance between two adjacent points also increases (Hastie et al. 2001). If a fixed threshold value is used and it does not depend of the input space dimension, then the number of threshold violations will increase, which may lead to an excessive generation of clusters (Lughofer 2008). Looking at expression (25), one can note that the compatibility measure threshold includes information about the data space dimensionality because $\chi^2_{m,\lambda}$ is a function of the number m of inputs. Therefore no manual adjust is needed and the *curse of dimensionality* is automatically avoided. In other words, the clustering method has an automatic mechanism to adjust the compatibility measure threshold according to input space dimension. As the data dimension increases, the distance between two adjacent points also increases, and the respective compatibility measure decreases. However, the compatibility measure threshold also decreases avoiding excessive threshold violations.

The arousal mechanism adopted by the eMG model uses a sliding window assembled by the last w observations. More specifically, we define the arousal index as the probability of observing less then nv violations of the compatibility threshold on a sequence of w observations. Low values of the arousal index are associated with no or few violations of the compatibility threshold, implying a high confidence about the system knowledge. High values of the arousal index are associated with several threshold violations, meaning that the current cluster structure must be revised.

To compute the arousal index for each observation, a related *occurrence* value o_i^k is found using the following expression

$$o_i^k = \begin{cases} 0, & \text{for } M(x^k, v_i^k) < \chi^2_{m,\lambda} \\ 1, & \text{otherwise} \end{cases} \tag{26}$$

Notice that the occurrence value $o_i^k = 1$ indicates threshold violation.

Occurrence value o_i^k can also be viewed as the output of a statistical test to evaluate if the values of $M(x^k,\ v_i^k)$ are the expected ones. The null hypothesis of the corresponding test is that $M(x^k,\ v_i^k)$ can be modeled by a Chi-Square distribution with m degrees of freedom. Under null hypothesis, the probability of observing $o_i^k = 1$ is λ because λ defines $\chi^2_{m,\lambda}$ and it is the probability of observing a false positive, i.e., $M(x^k,\ v_i^k) > \chi^2_{m,\lambda}$.

Since the nature of o_i^k is binary and the probability of observing $o_i^k = 1$ is known, the random variable associated with o_i^k can be described by a *Bernoulli* distribution with probability of success λ.

Given a sequence assembled by the last w observations, the number of threshold violations nv_i^k is:

$$nv_i^k = \begin{cases} \sum_{j=0}^{w-1} o_i^{k-j}, & k > w \\ 0, & \text{otherwise} \end{cases} \tag{27}$$

Notice that nv_i^k is computed during the first w steps. This means that the algorithm has an initial latency of w steps. However this causes no problem because usually w is much smaller than the number of steps in which learning occurs. For instance, in real-time applications learning can happen continuously.

The discrete probability distribution of observing nv threshold violations on a window of size w is $P(NV_i^k = nv)$, with NV_i^k assuming the values $nv = 0, 1, \ldots, w$. Thus, because NV_i^k is the sum of a sequence of i.i.d. random variables drawn from a *Bernoulli* distribution with the same probability of success λ, $P(NV_i^k = nv)$ can be characterized by the *Binomial* distribution:

$$P(NV_i^k = nv) = \begin{cases} \begin{pmatrix} w \\ nv \end{pmatrix} & \lambda^{nv}(1-\lambda)^{w-nv}, nv = 0, \ldots, w \\ 0, & \text{otherwise} \end{cases} \tag{28}$$

The binomial distribution gives the probability of observing nv threshold violations in a sequence of w observations. High probability values enforce the assumption that observations fit the current cluster structure while low probability values suggests that the observations should be described by a new cluster.

The arousal index is defined as the value of the cumulative probability of NV_i^k, i.e.

$$a_i^k = P(NV_i^k < nv) \tag{29}$$

The threshold value of the arousal index T_a is $1 - \lambda$, where λ is the same as the one that defines the threshold for the compatibility measure. The minimum number of compatibility threshold violations on a window of size w necessary to exceed T_a can be computed numerically looking for the first value of nv for which the discrete cumulative distribution is equal to or greater than $1 - \lambda$. More formally

$$nv^* = \arg\min_{nv} \left| \sum_{k=1}^{nv} \begin{pmatrix} w \\ nv \end{pmatrix} \lambda^k (1-\lambda)^{w-k} - (1-\lambda) \right| \tag{30}$$

The clustering algorithm of eMG continually revises the current cluster structure and eventually merges similar clusters. The compatibility between the updated or created cluster and all remaining cluster centers is computed at each step. If, for a given pair, the compatibility exceeds the threshold T_ρ, then the two clusters are merged, i.e., if $\rho_i^k(v_j^k,\ v_i^k) > T_\rho$ or $\rho_j^k(v_i^k,\ v_j^k) > T_\rho$, then clusters j and i are merged.

The compatibility between two clusters i and j is computed as follows:

$$\rho_i^k(v_j^k, v_i^k) = \exp\left[-\frac{1}{2}M(v_j^k, v_i^k)\right] \tag{31}$$

where $M(v_j^k, v_i^k)$ is the *M-distance* between cluster centers i and j, that is:

$$M(v_j^k, v_i^k) = (v_j^k - v_i^k)(\Sigma_i^k)^{-1}(v_j^k - v_i^k)^T \tag{32}$$

To check if two clusters are similar, we need to compute $\rho_i^k(v_j^k, v_i^k)$ and $\rho_j^k(v_i^k, v_j^k)$ because usually $\Sigma_i^k \neq \Sigma_j^k$.

Notice that the clustering algorithm has only three parameters:

- the primary learning rate α used to compute v_i^k and Σ_i^k;
- the window size w used by the arousal mechanism;
- the confidence level λ to compute thresholds T_ρ and T_a.

The primary learning rate is usually set to a small value, i.e., typically $\alpha \in [10^{-1}, 10^{-5}]$.

The window size w is a problem specific parameter because it defines how many consecutive observations must be considered to compute the arousal index. In other words, considering the current system knowledge, w defines the length of the anomaly pattern needed to classify data either as a new cluster or as a noise or outlier.

The value of the significance level λ depends on w. It must be set such that the arousal threshold T_a corresponds to more than one compatibility threshold violation, i.e., $nv > 1$ when $a_i^k > T_a$. Suggested ranges for values of λ, given w, are:

$$\lambda \geq \begin{cases} 0.01, & \text{if } w \geq 100 \\ 0.05, & \text{if } 20 \leq w < 100 \\ 0.1, & \text{if } 10 \leq w < 20 \end{cases} \tag{33}$$

The clustering process may start with either a single observation or an initial data set. If initial data set is available, then an off-line clustering algorithm can be used to estimate the initial number of clusters and their respective parameters. The off-line algorithm should be capable to provide both, cluster centers and respective dispersion matrices. If the clustering process starts with a single observation, then an initial dispersion matrix Σ_{init} must be chosen, eventually using a priori information about the problem.

Whenever a new cluster is created during the clustering process, the new cluster center is set as the current observation, and the new dispersion matrix is the initial value Σ_{init}.

If two clusters are merged, then the center of the resulting cluster is the average of the corresponding clusters centers and the dispersion matrix is Σ_{init}.

4.2 Evolving multivariable fuzzy model

The eMG model uses the evolving clustering algorithm described above to construct the rule base. The number of eMG rules is the same as the number of clusters found by the clustering algorithm at each step when a new cluster can be created, an existing cluster removed, or existing clusters updated. Summing up, rules can be created, merged, or adapted at each step of the algorithm. Rules antecedents are of the form:

$$x^k \text{ is } H_i \tag{34}$$

where x^k is a $1 \times m$ input vector and H_i is a fuzzy set with multivariable Gaussian membership function (21) and parameters extracted from the corresponding cluster center and dispersion.

The model is formed by a set of functional fuzzy rules:

$$R_i : \text{IF } x^k \text{ is } H_i \quad \text{THEN} \quad y_i^k = \gamma_{io}^k + \sum_{j=1}^{m} \gamma_{ij}^k x_i^k \tag{35}$$

where R_i is the ith fuzzy rule, for $i = 1, \ldots, c^k$, c^k is the number of rules, and γ_{io}^k and γ_{ij}^k are the parameters of the consequent at step k.

The model output is the weighted average of the outputs of the each rule, that is:

$$\hat{y}^k = \sum_{i=1}^{c^k} \Psi_i(x^k) y_i^k \tag{36}$$

with normalized membership functions:

$$\Psi_i(x^k) = \frac{\exp\left[(x^k - v_i^k)\Sigma_i^{-1}(x^k - v_i^k)^T\right]}{\sum_{i=1}^{c^k} \exp\left[(x^k - v_i^k)(\Sigma_i^k)^{-1}(x^k - v_i^k)^T\right]} \tag{37}$$

where v_i^k and Σ_i^k are the center and dispersion matrix of the ith cluster membership function at step k.

Contrary to ePL, the parameters of the consequent are updated using the weighted recursive least squares (Ljung 1999; Astrom and Wittenmark 1994) algorithm, similarly as other TS evolving fuzzy models (Angelov and Filev 2004; Lughofer 2008). Thus, the consequent parameters and matrix P_i of the update formulas for rule i at each iteration k become:

$$\begin{aligned} \gamma_i^{k+1} &= \gamma_i^k + P_i^{k+1} x^k \Psi_i(x^k)\left[y_i^k - ((x^k)^T \gamma_i^k)\right] \\ P_i^{k+1} &= P_i^k - \frac{\Psi_i(x^k) P_i^k x^k (x^k)^T P_i^k}{1 + (x^k)^T P_i^k x^k} \end{aligned} \tag{38}$$

The eMG algorithm can be initialized either with an existing data set, or with a single observation.

If the eMG starts with an existing data set, then an offline clustering algorithm can be used to estimate the number and parameters of the initial set of rules. Clustering can be done in the input space and a rule created for each cluster. The antecedent parameters of each rule is extracted from the clusters, and the consequent parameters estimated by the weighted least squares algorithm.

If the eMG starts with a single observation, then one rule is created with the antecedent membership function centered at the observation, and the respective dispersion matrix set at the pre-defined initial value. The consequent parameters are initialized as $\gamma^0 = [y^0 0 \cdots 0]$ and $P^k = \omega I_{m+1}$, where I_{m+1} is an $m + 1$ identity matrix and ω is a large real value, for example, $\omega \in [10^2, 10^4]$ (Astrom and Wittenmark 1994).

As new data is input, the eMG algorithm may create, update or merge clusters. Thus, the set of rules, the rule-base, must be updated as well. This is done as follows.

If a new cluster is created, then a corresponding rule is also created with antecedent parameters extracted from the cluster and consequent parameters computed as the weighted average of the parameters of the existing clusters:

$$\gamma^k_{\text{new}} = \frac{\sum_{i=1}^{c^k} \gamma^k_i \rho^k_i}{\sum_{i=1}^{c^k} \rho^k_i} \tag{39}$$

The matrix P is set as $P^k_{\text{new}} = \omega I_{m+1}$.

If an existing cluster is updated, then the antecedent parameters of the corresponding rule are updated accordingly.

Finally, if two clusters i and j are merged, then the consequent parameters of the resulting rule are computed as follows:

$$\gamma^k_{\text{new}} = \frac{\gamma^k_i \rho^k_i + \gamma^k_j \rho^k_j}{\rho^k_i + \rho^k_j} \tag{40}$$

The matrix P is set as $P^k_{\text{new}} = \omega I_{m+1}$.

5 Fixed income options and pricing model

5.1 IDI contracts

The most important Brazilian interest rate is the One-Day Interbank Deposit Contract rate, or “CDI rate”. It is computed as the average rate of all interbank overnight transactions in Brazil, published daily by ANBIMA (Brazilian National Association of Investment Banks).[2]

The underlying asset of IDI options is the One-Day Interbank Deposit Contract Index, or IDI index. This index is calculated by BM&FBOVESPA as the result of the accrual daily CDI rate. The IDI index was set to 100,000 on January 2nd, 2003. The IDI index at t is

$$\text{IDI}_t = 100,000 \cdot \prod_{u=1}^{t} (1 + \text{CDI}_u) \tag{41}$$

where $u = 0$ refers to the date when the IDI was set to 100,000 and CDI_u is the CDI rate of day u.

IDI options are European-styled cash-settled options that entitles the owner to receive the maximum between zero and the difference of the index and the strike price, according to the option style, i.e., call or put.

5.2 Black formula

The model of Black (Black formula) is commonly used by players of BM&FBOVESPA. The model is based on Black (1976) and gives the price of an IDI call option as:

$$c_t = \text{IDI}_t \cdot N(d_1) - X \cdot P(t,T) \cdot N(d_2) \tag{42}$$

where:

$$d_1 = \frac{\ln\left(\frac{\text{IDI}_t}{X \cdot P(t,T)}\right) + \frac{\sigma^2 \cdot (T-t)^3}{6}}{\sigma \cdot \sqrt{\frac{(T-t)^3}{3}}} \quad \text{and} \quad d_2 = d_1 - \sqrt{\frac{(T-t)^3}{3}}$$

where IDI_t is the value of the IDI index at time t, $P(t, T)$ is the price at time t of a discounted 1$ equity with maturity T, σ is the short-term interest rate volatility, X is the strike price, and $N(\cdot)$ the normal cumulative distribution function.

In this model, short-term interest rate volatility is estimated using a Generalized Autoregressive Conditional Heteroskedasticity, GARCH (1,1), process based on CDI rate returns, parametrized according to Bayesian Information Criterion (BIC) (Schwarz 1978). Volatility is the unobservable parameter in the model of Black. The strike price, IDI index, and maturity form the database.

5.3 Neural networks models

A number of empirical studies that include Zurada et al. (1999) show that ANN are better than time series models in finance and economics applications. However, Yang et al. (1999) pointed that Black and neural networks models have similar performance in some cases.

For comparison purposes, we adopt Elman and Jordan recurrent neural networks structures, ERNN and JRNN, respectively, for IDI calls option pricing.

The main issue in neural network models is how to find the optimal architecture, that is, the optimal number of hidden layers and neurons. For inputs we used a model with all parameters that can influence IDI call option price: IDI index, short-term interest rate volatility, maturity and strike price.

[2] Brazilian banks usually express their costs of funding in percentage of the published CDI terms, and it can be said that CDI is the relevant cost of opportunity for Brazilian banks.

The database was partitioned into three sets: training set, testing set and validation set, with 65, 20 and 15% of total data, respectively. The gradient algorithm was used to train the neural network.

6 Results and discussion

Data consists on time series of IDI options for different strikes and maturities. The data covers the period from January 2nd, 2003 to June 5th, 2008.[3] We selected the most liquid IDI calls within each day with negotiated contracts greater or equal to 1000. Nevertheless, the CDI daily index was selected for all business days of the period.

The models were adjusted considering the same inputs that determines IDI call options price according to the Black model: IDI index at day t, short-term interest rate volatility, estimated by a GARCH(1,1) process considering CDI spot rate returns, strike price and maturity.

Table 1 summarizes the neural networks structures adjusted to IDI call options. The hidden neurons numbers were selected according to a Bayesian Information Criterion procedure (BIC) (Schwarz 1978) based on root mean squared error (RMSE). We can see that the obtained structures are composed by three hidden layers.

The ePL model adopted the following values: $\beta = \tau = \alpha = 0.01$ and $\theta = 0.11$. The ePL found 4 rules during the 3,512 days (trainning) period. During testing, ePL uses daily data and run as in online mode.

The eTS and xTS models were developed considering the first 3,512 days before the testing period. The value of the cluster radii for the eTS model was $r = 0.4$ and the initial value for covariance matrix $\Omega = 450$. The eTS model found 8 rules. For the xTS model is necessary to define only the initial value for covariance matrix $\Omega = 750$ to initialize the recursive least squares. xTS found 6 rules. The eTS and xTS implementations used here are reported in Angelov and Zhou (2006b).

The eMG started clustering with the first observation and the parameters were $\lambda = 0.01$, $w = 50$, $\alpha = 0.01$ and Σ_{init} defined as a diagonal matrix containing the variance of each input variable in the diagonal, estimated using the first 250 input samples. The eMG model derived 6 rules.

The superiority in terms of low computational costs (time) and low complexity (low number of fuzzy rules) of evolving family is clearly visible in Table 2. The test was carried out on a laptop computer Intel(R) Core(TM)2 Duo with a CPU 2.00 GHz. Total CPU time means processing all available data samples. The programming language of the eTS and xTS models is Java. ePL and eMG were written in Matlab.

Table 1 Recurrent neural network structures

Models	Number of neurons		
	1st layer	2nd layer	3rd layer
ERNN	4	5	3
JRNN	5	6	4

Table 2 Structure for each evolving systems

Models	Parameters	Rules	Total CPU time (s)
eTS	2	8	8.78
xTS	1	6	9.03
ePL	4	4	9.02
eMG	3	6	11.95

6.1 Performance measures

We consider the mean absolute percentage error (MAPE), maximum percentage error (MPE), root mean squared error (RMSE), mean absolute error (MAE) and Theil's inequality coefficient (TIC)[4] as performance measures:

$$\mathrm{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\frac{|c_t - \hat{c}_t|}{c_t} \tag{43}$$

$$\mathrm{MPE} = \max_{t=1,\ldots,N}\left\{100\frac{|c_t - \hat{c}_t|}{c_t}\right\} \tag{44}$$

$$\mathrm{RMSE} = \sqrt{\left(\frac{1}{N}\right)\sum_{t=1}^{N}(c_t - \hat{c}_t)^2} \tag{45}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N}|c_t - \hat{c}_t| \tag{46}$$

$$\mathrm{TIC} = \frac{\sqrt{\frac{\sum_{t=1}^{N}(c_t - \hat{c}_t)^2}{N}}}{\sqrt{\frac{\sum_{t=1}^{N}(c_t)^2}{N}} + \sqrt{\frac{\sum_{t=1}^{N}(\hat{c}_t)^2}{N}}} \tag{47}$$

where c_t is the true value of the tth point of the series of length $N, \hat{c}_t$ is the predicted value.

Moreover, we computed the determination coefficient R^2 which is also an indicator of performance. This indicator is obtained by a simple regression model of the market prices estimated by models against the theoretical prices.

The results were evaluated according to the degree of moneyness (M), defined by the relation between the strike price present value and the underlying asset spot price:

[3] The data was obtained with BM&FBOVESPA.

[4] The best model, in terms of accuracy, is the one that shown TIC index close to zero.

Table 3 Forecast evaluation and error measures for IDI call options

Models	MAPE	MPE	RMSE	MAE	TIC	R^2
Black	47.34	549.10	0.27	0.23	0.21	0.01
RNRE	15.42	500.53	0.10	0.07	0.09	0.18
RNRJ	15.04	504.35	0.09	0.07	0.09	0.20
eTS	12.73	501.12	0.08	0.06	0.07	0.39
xTS	11.60	491.57	0.07	0.05	0.07	0.45
ePL	12.14	494.57	0.08	0.06	0.07	0.42
eMG	9.81	453.94	0.07	0.05	0.06	0.44

$$M = \frac{\mathrm{IDI_t}}{X \cdot P(t,T)} \tag{48}$$

In this case, IDI options were considered according to the degree of moneyness: out-of-the-money ($M \leq 1 - p$), at-the-money ($1 - p < M \leq 1 + p$) and in-the-money ($M > 1 + p$), considering $p = 0.05$.

Although all performance measures of forecasting accuracy that have been extensively employed in practice, they do not reveal whether the forecast of a model is statistically superior to another one. Therefore, it is imperative to use additional tests to help comparison among two or more competing models in terms of forecasting accuracy.

The parametric test of equal forecast accuracy adopted here is the Morgan–Granger–Newbold (MGN) test, proposed by Diebold and Mariano (1995). This test is employed when the assumption of contemporaneous correlation in the prediction errors is relaxed. Let $x_t = (e_{it} + e_{jt})$ and $z_t = (e_{it} - e_{jt})$, where e_{it} and e_{jt} are the residuals of two models, i and j, adjusted. The null hypothesis of equal forecast accuracy is equivalent to zero correlation between x and z (that is, $\rho_{xz} = 0$) and the test statistic

$$\mathrm{MGN} = \frac{\hat{\rho}_{xz}}{\sqrt{\frac{1-\hat{\rho}^2_{xz}}{N-1}}} \tag{49}$$

is distributed as Student's t with $N - 1$ degrees of freedom. $\hat{\rho}_{xz}$ is the estimated correlation coefficient between x and z.

The significance (SIGN) test, due to Lehmann (1988), is a nonparametric test that does not require errors to be normally distributed or serially uncorrelated. The null hypothesis is $H_0 : \Theta_N = \frac{N}{2}$, which indicates that the models will be equally accurate. The null hypothesis is rejected when Θ_N is sufficiently large. The test statistic

$$\mathrm{SIGN} = \frac{(\Theta_N - \frac{N}{2})}{\frac{1}{2}\sqrt{N}} \tag{50}$$

tends to the standard normal distribution. Θ_N denotes the number of periods that the forecasting errors of a model is expected to surpass the errors of another model.

Table 4 Forecast evaluation for IDI call options, according to the degree of moneyness

Models	MAPE	MPE	RMSE	MAE	TIC	R^2
Out-of-the-Money						
Black	68.12	625.91	0.38	0.32	0.24	0.01
ERNR	14.76	502.87	0.13	0.10	0.08	0.18
JRNR	15.32	478.76	0.12	0.09	0.09	0.20
eTS	11.93	453.09	0.07	0.05	0.06	0.39
xTS	10.86	462.12	0.06	0.06	0.07	0.43
ePL	11.04	449.87	0.09	0.08	0.07	0.44
eMG	10.82	444.82	0.07	0.05	0.06	0.42
At-the-Money						
Black	44.31	501.87	0.30	0.22	0.19	0.02
ERNR	14.51	507.65	0.10	0.08	0.09	0.21
JRNR	15.23	487.01	0.09	0.07	0.09	0.21
eTS	12.21	444.39	0.07	0.05	0.07	0.40
xTS	10.92	450.92	0.06	0.04	0.06	0.44
ePL	10.43	449.44	0.09	0.06	0.07	0.44
eMG	9.67	458.23	0.06	0.05	0.06	0.43
In-the-Money						
Black	45.76	564.98	0.29	0.22	0.19	0.02
ERNR	16.23	501.23	0.11	0.09	0.09	0.19
JRNR	15.99	489.00	0.10	0.10	0.08	0.19
eTS	11.76	451.92	0.07	0.06	0.07	0.37
xTS	11.89	448.76	0.07	0.05	0.07	0.44
ePL	11.06	451.22	0.06	0.04	0.08	0.43
eMG	9.55	439.82	0.06	0.04	0.06	0.41

6.2 Comparison and analysis

Here we evaluate the models using the performance measures and accuracy statistical tests. The test set is composed by the data from August 1st, 2007 to June 5th, 2008, a sample with 884 points.

Table 3 shows the results for each performance measures. According to these measures, we can see that Back model performed worst. The neural networks have shown no significant performance difference. The evolving fuzzy models provided forecasts with the lowest performance measures values. This shows the capability of evolving models to capture temporal data dependence. The evolving fuzzy models give IDI call prices closer to real prices than the remaining models.

Table 4 shows the performance measures according to the degree of moneyness. It confirms the superiority of the evolving models. They also capture IDI option pricing movements. The Black model, in general, has low capability to price out-of-the-money IDI call options because the higher the degree of moneyness, the higher the error.

The results of the MGN test are summarized in Table 5. It gives pair-wise comparisons between forecasts of two

Table 5 MGN and SIGN tests evaluation for IDI call options

Models	Test	RNRE	RNRJ	eTS	xTS	ePL	eMG
Black	MGN	21.393 (0.0000)	21.922 (0.0000)	26.7089 (0.0000)	29.6673 (0.0000)	27.161 (0.0000)	32.8118 (0.0000)
	SIGN	−5.264 (0.0000)	−5.213 (0.0000)	−5.112 (0.0000)	−5.348 (0.0000)	−5.314 (0.0000)	−5.550 (0.0000)
ERNR	MGN	–	1.602 (0.1094)	4.488 (0.0000)	8.025 (0.0000)	5.383 (0.0000)	10.381 (0.0000)
	SIGN	–	0.631 (0.3267)	−0.632 (0.3268)	−1.137 (0.2090)	−1.035 (0.2332)	−2.063 (0.0508)
JRNR	MGN	–	–	4.062 (0.0001)	6.992 (0.0000)	4.416 (0.0000)	9.505 (0.0000)
	SIGN	–	–	−0.396 (0.3689)	−1.204 (0.1932)	−1.002 (0.2414)	−2.029 (0.0508)
		–					
eTS	MGN	–	–	–	−12.245 (0.0000)	−0.4023 (0.6875)	−8.4094 (0.0000)
	SIGN	–	–	–	3.866 (0.0002)	3.5628 (0.0007)	4.5398 (0.0001)
xTS	MGN	–	–	–	–	5.2204 (0.0000)	−5.0660 (0.0000)
	SIGN	–	–	–	–	−1.7940 (0.0798)	3.8828 (0.0002)
ePL	MGN	–	–	–	–	–	7.437 (0.0000)
	SIGN	–	–	–	–	–	−4.051 (0.0001)

competing models. It reveals that statistically all the models outperform the Black model when pricing IDI call options.[5] Table 5 also reveals that, from the MGN point view, eMG is statistically more accurate than ePL.

The nonparametric test, SIGN, also compares pair-wise forecasts of competing models. Table 5 shows that, agreeing with the results pointed by MGN, that neural networks and evolving fuzzy models are better to price IDI options. SIGN also confirms the capability of eMG to model the Brazilian fixed income option pricing.

7 Conclusion

This paper has suggested evolving fuzzy systems as approaches to develop evolving fuzzy rule-based models for options pricing. Option pricing is an important field of financial engineering and particularly relevant for derivatives market. Computational experiments with prices the Brazilian fixed income options data have shown that evolving fuzzy modeling outperforms conventional Black model and neural network models. Further work shall address human perception as part of modeling and models, and experiment the models in actual decision-making instances.

Acknowledgments This work was supported by the Brazilian National Research Council, CNPq, and CAPES. The authors thank the anonymous referees for the comments which helped us to improve the paper.

References

Abonyi J, Babuska R, Szeifert F (2002) Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models. IEEE Trans Syst Man Cybern Part B 32(5):612–621

Almeida CIR, Vicente JV (2006) Pricing and hedging brazilian fixed income options. In: Proceedings of the 6th Brazilian financial meeting, Vitória, Brazil

Angelov P (2002) Evolving rule-based models: a tool for design of flexible adaptive systems. Springer, Heidelberg, ISBN:3-7908-1457-1

Angelov P, Filev D (2004) An approach to on-line identification of evolving Takagi-Sugeno fuzzy models. IEEE Trans Syst Man Cybern Part B 34(1):484–498

Angelov P, Filev D (2005) Simpl_eTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models. In: Proceedings of the 2005 international conference on fuzzy systems, FUZZ-IEEE, Reno, USA, pp 1068–1073

Angelov P, Zhou X (2006a) Evolving fuzzy systems from data streams in real-time. In: Proceedings of the international symposium on evolving fuzzy systems, pp 29–35

Angelov P, Zhou X (2006b) Online identification of Takagi-Sugeno fuzzy models DEMO. http://194803445/eTS/

Astrom KJ, Wittenmark B (1994) Adaptive control, 2nd edn. Addison-Wesley Longman Publishing Co., Inc., Boston

Barbedo CHS, Vicente JVM, Lion OMB (2009) Pricing asian interest rate options with a Three-Factor HJM model. Working Paper Series 188. http://www.bcb.gov.br/pec/wps/ingl/wps188.pdf

Black F (1976) The pricing of commodity contracts. J Financ Econ 3:167–179

Black F, Scholes M (1973) The pricing of options and corporate liabilities. J Political Econ 81(3):637–654

Chiu SL (1994) Fuzzy model identification based on cluster estimation. J Intell Fuzzy Syst 2:267–278

Diebold FX, Mariano RS (1995) Comparing predictive accuracy. J Bus Econ Stat 13(3):253–263

Figa-Talamanca G, Guerra ML (2009) Fuzzy option value with stochastic volatility models. In: Proceedings of the 8th international conference on intelligent systems design and applications, Pisa, Italy

Gençay R, Qi M (2001) Pricing and hedging derivative securities with neural networks:Bayesian regularization, early stopping and bagging. IEEE Trans Neural Netw 12(4):726–734

Hastie T, Tibshirani R, Friedman JH (2001) The element of statistical learning: data mining, inference, and prediction, 2nd edn. Springer, New York

Johnson CR (1988) Lectures & adaptive parameter estimation, 1st edn. Prentice-Hall, Upper Saddle River

Junior AF, Greco F, Lauro C, Francisco G, Rosenfeld R, Oliveira R (2003) Application of Hull-White model to Brazilian IDI

[5] The p values are shown in parenthesis.

options. In: Proceedings of the 3th Brazilian financial meeting, Florianópolis, Brazil

Kasabov N, Song Q (2002) DENFIS: dynamic evolving neural-fuzzy inference system and its application for time series prediction. IEEE Trans Fuzzy Syst 10(2):144–154

Kim E, Park M, Kim S, Park M (1998) A transformed input-domain approach to fuzzy modeling. IEEE Trans Fuzzy Syst 6(4): 596–604

Lehmann EL (1988) Nonparametric statistical methods based on ranks. Prentice Hall, Englewood Cliffs

Lemos A, Gomide F, Caminhas W (2011) Multivariable gaussian evolving fuzzy modeling system. IEEE Trans Fuzzy Syst 19(1): 91–104

Leu Y (2010) A fuzzy time series-based neural network approach to option price forecasting. Lect Notes Comput Sci 5990:360–369. doi:10.1007/978-3-642-12145-6_37

Lima E (2008) Modelo Takagi-Sugeno evolutivo participativo. Master's thesis, School of Electrical and Computer Engineering, University of Campinas

Lima E, Ballini R, Gomide F (2010a) Dynamic analysis of the participatory learning algorithm. In: Proceedings of the 36th annual convention of the Society for the Study of Artificial Intelligence and Simulation of Behaviour (AISB), Leicester, UK

Lima E, Hell M, Ballini R, Gomide F (2010b) Evolving fuzzy modeling using participatory learning. In: Angelov P, Filev D, Kasabov N (eds) Evolving intelligent systems, chap 4. Wiley-IEEE-Press, NJ, pp 67–86

Liu F (2009) Pricing currency options based on fuzzy techniques. Eur J Oper Res 193(2):530–5402

Ljung L (1999) System identification—theory for the user, 2nd edn. Prentice-Hall, Englewood Cliffs

Ljung L (1999b) System identification: theory for the user. Prentice-Hall, Heidelberg, ISBN:0-13-656695-2

Lughofer ED (2008) FLEXFIS: a robust incremental learning approach for evolving Takagi-Sugeno fuzzy models. IEEE Trans Fuzzy Syst 16(6):1393–1410

Maciel LS, Ballini R (2010) Option pricing using fuzzy and neuro-fuzzy inference systems. In: Proceedings of the 4th CSDA international conference on computational and financial economics, London

Pedrycz W, Gomide F (2007) Fuzzy systems engineering: toward human-centric computing. Wiley Interscience, NJ

Schwarz G (1978) Estimating the dimension of a model. Ann Stat 6(2): 461–468

Silva L, Gomide F, Yager RR (2005) Participatory learning in fuzzy clustering. In: The 2005 IEEE international conference on fuzzy systems FUZZ-IEEE, pp 857–861

Wellstead PE, Zarrop MB (1995) Self-tuning systems: control and signal processing, 1st edn. Wiley, New York

Wu H (2005) European option pricing under fuzzy environments. Int J Intell Syst 20(1):89–102

Yager RR (1990) A model of participatory learning. IEEE Trans Syst Man Cybern 20:1229–1234

Yager R, Filev D (1994) Approximate clustering via the mountain method. IEEE Trans Syst Man Cybern 24(8):1279–1284

Yang ZR, Platt MB, Platt HD (1999) Probabilistic neural network in bankruptcy predictions. J Bus Res 44(2):67–74

Young PC (1984) Recursive estimation and time-series analysis. Springer, Berlin

Zurada JM, Foster BP, Ward TJ, Barker PM (1999) Neural networks versus logit regression models for financial distress response variables. J Appl Bus Res 15:21–29