MIMO Evolving Participatory Learning Fuzzy Modeling

Leandro Maciel and Fernando Gomide
Department of Computer Engineering and Automation
School of Electrical and Computer Engineering
University of Campinas
Campinas, São Paulo, Brazil 13083–852
Email: {maciel,gomide}@dca.fee.unicamp.br

Rosangela Ballini
Department of Economic Theory
Institute of Economics
University of Campinas
Campinas, São Paulo, Brazil 13083–857
Email: ballini@eco.unicamp.br

Abstract—Evolving participatory learning fuzzy modeling is a flexible and effective method to handle real world complex systems. It is capable to process and learn from streams of data online, and is a natural candidate to find fuzzy rule-based model structures in dynamic environments. This paper extends the evolving participatory learning fuzzy approach for multi-input multi-output - MIMO - processes modeling and suggests the use of subtractive clustering (SC) algorithm to obtain an initial rule base when a priori knowledge is available. SC improves autonomy because it adds learning flexibility. Modeling uses the participatory learning fuzzy clustering algorithm to find rule antecedents, and the recursive MIMO least squares algorithm to estimate the parameters of the linear rules consequents. A novel application concerning modeling the term structure of interest rates and forecasting is also included. Computational results based on the US fixed income market data show that the MIMO evolving participatory learning fuzzy model describes the interest rate behavior accurately, revealing a high potential to forecast complex nonlinear dynamics in uncertain environments.

I. INTRODUCTION

In the recent years, the need to process the constantly growing amount of information efficiently, and to extract meaningful knowledge plays a key role in the modeling of complex systems. Often the data come in enormous quantities, are dynamic, and arrive in the form of streams of data [1]. To extract knowledge from streams of data, online data-driven rule/knowledge methods have been developed [12]. Adaptive fuzzy systems models which are self-developed from a stream of data are a form of evolving fuzzy systems (eFS) that can be approached via fuzzy and neuro-fuzzy techniques. Evolving systems allow simultaneous learning of both, model structure and functionality from flows of data. They can, for instance, be viewed as a fuzzy mixtures of adaptive Gaussians to approximate the real data density distribution [2]. Essentially, eFS possesses the ability to expand or shrink the structure of the model, and to estimate its parameters. This can be done incrementally, online, and in real time, if this is the case [1].

Some of the eFS currently reported in the literature consider evolvable functional rule-based systems in which the structure (number of rules and antecedents/consequent parameters) evolves continuously using the clusters created/excluded by recursive clustering algorithms [14]. For instance, a pioneering approach to the online learning of an evolving Takagi-Sugeno (eTS) model was introduced in [4]. The rule-base and parameters of the TS model continually evolve by adding new rules with more summarization power and modifying existing rules and parameters. The rule base structure is constructed using an incremental version of the subtractive clustering algorithm [10]. The recursive evaluation of the information potential of new data samples is used to create new clusters or revise the existing ones. The rule-consequent parameters are updated with the recursive least-squares algorithm [19].

To reduce the complexity of information potential calculations in eTS, the Simpl_eTS model was developed later in [3]. Simpl_eTS is a simplified version of eTS which replaces the notion of information potential by the concept of scatter to provide a similar, but computationally more effective, algorithm. An eXtended eTS (xTS) was also developed in [2], bringing in the idea to adapt the cluster radius (zone of influence) of eTS models. Cluster radius is an important parameter that affects the results because it is part of the membership function and thus of the activation level of the fuzzy sets. Extensions to the original eTS and xTS models comprise the eTS+ approach [1]. In eTS+, the antecedent parameters and rule-base structure are updated using criteria such as age, utility, local density and zone of influence.

A distinct, but conceptually similar approach for adaptive TS modeling is the dynamic evolving neural-fuzzy model [13]. The approach uses a distance-based recursive evolving clustering method to adapt the rule base structure, and a weighted recursive least squares with forgetting factor algorithm to update rules consequent parameters. A recursive clustering algorithm derived from a modification of the vector quantization technique, called evolving vector quantization, is another significant methodology to construct the flexible fuzzy inference system (FLEXFIS) [20]. More recent examples of evolving eFS in the realm of neuro-fuzzy type-2 are the self-organizing fuzzy modeling and modified least-squares network (SOFMLS) [25]. SOFMLS employs an evolving nearest neighborhood clustering algorithm and the sequential adaptive fuzzy inference system (SAFIS) [24], using a distance criterion in conjunction with an influence measure of the new rules created. The fuzzy self-organizing neural network [15] is an alternative evolvable system which adopts an error criterion to
consider the generalization performance of the network.

As reported in [14], a weakness in the recursive clustering algorithms adopted by most of the eFS approaches is the lack of robustness due to noise or outliers. In these situations, the algorithms may create new clusters instead of rejecting or smoothing the noisy data or outliers. A method to improve robustness was suggested in [16] namely, the evolving participatory learning (ePL) fuzzy modeling. The approach joins the concept of participatory learning (PL) [28] with the evolving fuzzy modeling idea [6], [4]. In evolving systems the PL concept is implemented as an unsupervised clustering algorithm [26] and is a natural candidate to find rule base structures in dynamic environments. Similarly as in eTS, structure identification and self-organization in ePL means estimation of the focal points of the rules, 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 [10]. ePL learns the rule base structure at each step through convex combinations of new data samples and the closest cluster center, the focal point of a rule. Next, the rule base structure is updated and, similarly as in eTS, the parameters of the rule consequents computed using the recursive least squares algorithm. Based on the concept of ePL algorithm, [14] developed the multivariable Gaussian evolving fuzzy modeling approach (eMG). The eMG uses an evolving Gaussian clustering algorithm rooted in the concept of PL because each cluster is represented by a multivariable Gaussian membership function. The cluster structure (number, center, and shape of the clusters) is recursively updated at each step of the algorithm, and threshold parameters set automatically.

This paper extends the original multi-input-single-output ePL model for the case of multi-input-multi-output (MIMO) system model. Both parts of the ePL algorithm algorithm, the unsupervised fuzzy rule base antecedents and the supervised linear consequent parameters learning, are detailed for MIMO system modeling. Contrary to the original version of ePL [16], in which the initial number of clusters (and respective cluster centers) is estimated using the fuzzy c-Means (FCM) clustering algorithm [9] when a priori knowledge about the problem is available, in this paper we suggest to employ the subtractive clustering [10] algorithm (SC). The use of SC to set an initial rule-base structure is interesting because, differently from FCM, the user does not need to estimate the initial number of clusters. Thus, SC enhances autonomy when data is available beforehand. In addition, a novel application concerning modeling the term structure of interest rate forecasting for the US fixed income market, is also tackled. Term structure of interest rate is a difficult forecast problem, but essential for many tasks, including pricing financial assets and their derivatives, allocating portfolios, managing financial risk, conducting monetary policy, structuring fiscal debt, and valuing capital goods.

After this introduction, the rest of the paper proceeds as follows. The problem of online identification of MIMO ePL models is presented in section II. Section III addresses the yield curve forecasting of the term structure of interest rate. Computational results reported in Section IV show that the evolving approaches outperform current methodology. Concluding remarks and issues for further investigation are summarized in Section V.

II. ONLINE IDENTIFICATION OF MIMO EPL MODELS

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 current model. This is the case in recurrent models and dynamic systems whose operating conditions modify, fault occurs, or parameters of the processes change. Evolving systems use incoming information 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) [27] is a typical example of a fuzzy functional model. A particularly important case is when the rule consequents are linear functions which can, in general, be of the multi-input-multi-output (MIMO) type as follows [4]:

\[ R^i : \text{IF } x_1 \text{ is } \Gamma_{1}^i \text{ AND } x_2 \text{ is } \Gamma_{2}^i \text{ AND } \ldots \]
\[ \text{AND } x_n \text{ is } \Gamma_{n}^i \text{ THEN } y^i = \gamma_0^i + \sum_{j=1}^{n} \gamma_j^i x_j \quad (1) \]

where \( R^i \) is the \( i^{th} \) fuzzy rule \((i = 1, 2, \ldots, R)\), \( x_j \in \mathbb{R} \) is the input data \((j = 1, 2, \ldots, n)\), \( \Gamma_j^i \) is the fuzzy set associated with the \( j^{th} \) input variable of the \( i^{th} \) fuzzy rule, \( y^i \in \mathbb{R}^m \) is the linear output of the \( i^{th} \) rule, and \( \gamma_j^i \) are the parameters of the consequent.

In a TS model, the fuzzy regions are parameterized and each region is associated with a linear sub-system. Hence, the nonlinear system forms a collection of loosely (fuzzily) coupled (blended) multiple linear models [5]. The contribution of a local linear model to the overall output is proportional to the degree of firing of each rule. Antecedent fuzzy sets are described with Gaussian membership functions, which ensures greatest possible generalization of the description [4], [5]:

\[ \mu^i(x_j) = e^{-\frac{(x_j - v_j)^2}{2r^2}} \quad (2) \]

where \( \mu^i(x_j) \) is the membership degree of the current input \( x_j \) in \( \Gamma_j^i \), \( v_j^i \) is the respective cluster center or focal point, and \( r \) is a positive constant, which defines the radius of the antecedent, the zone of influence of the \( i^{th} \) model\(^1\).

The activation degree of particular fuzzy rule is, choosing the algebraic product t-norm:

\[ \tau^i(x_k) = \prod_{j=1}^{n} \mu_j^i(x_k) = \mu_1^i(x_k) \times \ldots \times \mu_n^i(x_k) \quad (3) \]

\(^1\)Reference [2] suggests a mechanism to adapt the cluster radius \( r \). Here we assume, without loss of generality as in [1], that it is constant.
The TS model output at \( k \) is the weighted average of the individual rule contributions:

\[
y = \sum_{i=1}^{R} \lambda_i y_i, \quad \lambda_i = \frac{\tau_i}{\sum_{i=1}^{R} \tau_i}
\]

where \( \lambda_i \) is the normalized firing level of the \( i^{th} \) rule, and \( R \) is the number of fuzzy rules.

As stated in [4], identification of a TS model requires two sub-tasks: i) learning the antecedent part of the model using a fuzzy clustering algorithm, and ii) learning the parameters of the linear consequents. To the problem of this paper we focus on the evolving fuzzy participatory learning algorithm for antecedent learning, and on the recursive least squares algorithm to estimate the consequent parameters.

A. Evolving Fuzzy Participatory Learning

Evolving fuzzy participatory learning 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. The main difference between ePL and eTS concerns the procedure to update the rule base structure. Differently from eTS, ePL uses a fuzzy similarity measure to determine the proximity between new data and the existing rule base structure. The rule base structure is isomorphic to the cluster structure, and on model confidence as well.

Participatory learning assumes that model 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 [16].

In online mode, the training data are collected continuously, rather than being a fixed set [2]. Let \( v_k^i \) be a variable that encodes the \( i^{th} \) \((i = 1, \ldots, R_k)\) cluster center at the \( k^{th} \) step. The aim of the participatory mechanism is to learn the value of \( v_k^i \), using a stream of data \( x_k \). In other words, each \( x_k \), \( k = 1, 2, \ldots \), is used as a vehicle to learn about \( v_k^i \). 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 \( v_k^i \) as being a valid. Implicit in this idea is that, to be useful and to contribute to the learning of \( v_k^i \), observations \( x_k \) must somehow be compatible with current estimates of \( v_k^i \).

In ePL, the object of learning are cluster structures. Cluster structures are defined by cluster centers (or prototypes). More formally, given an initial cluster structure, a set of vectors \( v_k^i, i = 1, \ldots, R_k \), is updated using a compatibility measure, \( \rho_k^i \in [0, 1] \) and an arousal index, \( a_k^i \in [0, 1] \). While \( \rho_k^i \) measures how much a data point is compatible with the current cluster structure, the arousal index \( a_k^i \) acts as a critic to remind when current cluster structure should be revised in front of new information contained in data.

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 \( i^{th} \) cluster center, the one most compatible with \( x_k \), is adjusted as follows:

\[
v_{k+1}^i = v_k^i + G_k^i (x_k - v_k^i)
\]

where

\[
G_k^i = \alpha \rho_k^i
\]

\( \alpha \in [0, 1] \) is the learning rate, and

\[
\rho_k^i = 1 - \frac{||x_k - v_k^i||}{n}
\]

with \( || \cdot || \) a norm, \( n \) the dimension of input space, and

\[i = \arg \max_j \{\rho_j^k\}\]

Notice that the \( i^{th} \) cluster center is a convex combination of the new data sample \( x_k \) and the closest cluster center.

Similarly as (5), the arousal index \( a_k^i \) is updated as follows:

\[a_{k+1}^i = a_k^i + \beta (1 - \rho_k^{i+1} - a_k^i)\]

The value of \( \beta \in [0, 1] \) controls the rate of change of arousal: the closer \( \beta \) 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 (9) into (6), that is, we assume

\[G_k^i = \alpha (\rho_k^i)^{1-a_k^i}\]

When \( a_k^i = 0 \), we have \( G_k^i = \alpha \rho_k^i \) 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 (5) can be viewed as a belief revision strategy whose effective learning rate (10) 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 (5) may push a given center closer to another one and a redundant cluster may be formed. Therefore a mechanism to exclude redundancy is needed. One mechanism is to verify if similar outputs due to distinct rules are produced. In PL clustering, a cluster center is declared redundant whenever its similarity with another center is greater than or equal to a threshold value. If this is the case, then we can either maintain the original cluster center or replace it by the average between the new data and the current cluster center. Similarly as in (7), the compatibility index among cluster centers is computed as follows:

\[
\rho_{c_i,k} = 1 - \frac{1}{n} \sum_{j=1}^{n} |v_{k}^j - v_{i}^j|
\]  

(11)

Therefore, if

\[
\rho_{c_i,k} \geq \lambda
\]  

(12)

then the cluster \(i\) is declared redundant.

Detailed guidelines to choose appropriate values of \(\alpha, \beta, \lambda\) and \(\tau\) are given in [21], where it is shown that they should be chosen such that:

\[
0 < \frac{\tau}{\beta} \leq 1 - \lambda \leq 1
\]

where

\[
\tau \leq \beta \text{ and } \lambda \leq 1 - \tau
\]

Rule base initialization can be done differently if data is available beforehand. In this case, we suggest the use of subtractive clustering (SC) algorithm [10] to obtain an initial rule base, instead of the fuzzy c-means as in [17]. This is because SC do not require the number of clusters from the user. This adds flexibility and increases PL autonomy once the number of clusters and the clusters themselves are derived from data. For completeness, we summarize the procedure of the SC next. Further details about SC algorithm is given in [10].

i) For all training data, select the one with the highest potential to be the first cluster center. The potential of a point is measured as the spatial proximity between all other data points, computed using a Euclidean distance operator;

ii) Reduce the potential of all other points by an amount proportional to the distance to this center.

iii) Define two boundary (lower and upper) conditions as a function of the maximal potential. If a potential of certain data point is higher than the upper threshold, it defines a new cluster center.

iv) If the potential of a point lies between the two boundaries and if this point is very close to some cluster center, then replace this cluster center.

When data is not available beforehand, the PL algorithm assumes the first data point as a cluster center.

**B. Parameter Identification by RLS Algorithm**

Estimation of the parameters of the consequent linear models can be formulated as a least squared problem [4]. Equation 4 can be transformed into a vector form as follows:

\[
y = \Lambda^T \Phi\]

(13)

where \(y = [y_1, y_2, \ldots, y_m]^T\) is the m-dimensional output of the MIMO ePL, \(\Lambda = [\lambda_1 x_e^T, \lambda_2 x_e^T, \ldots, \lambda_n x_e^T]^T\) denotes the fuzzily weighted extended inputs vector, \(x_e = [1 x^T]^T\) is the expanded data vector, \(\Phi = [\Psi_1^T, \Psi_2^T, \ldots, \Psi_R^T]^T\) represents the vector of parameters of the rule base, and , and \(\Psi_i = \begin{bmatrix} \gamma_{i0} & \ldots & \gamma_{i0m} \\ \vdots & \ddots & \vdots \\ \gamma_{in1} & \ldots & \gamma_{innm} \end{bmatrix}^T\) is the matrix of consequent part parameters (parameters of the \(i^{th}\) linear local subsystem) assuming \(m\) outputs.

Since the actual target output is provided at each step, the parameters of the consequents can be updated using recursive least squares algorithm RLS [10] considering locally or globally optimization. In this paper we use the locally optimal error criterion:

\[
\min E_L^i = \min \sum_{l=1}^{k} \lambda_i(x_l) (y_l - x_{el}^T \Psi_l)^2
\]

(14)

There are not only fuzzily coupled linear subsystems and streaming data, but also structure evolution, therefore the optimal update of the parameters of the \(i^{th}\) local subsystems is given by [4], [2], [1]:

\[
\Psi_{k+1}^i = \Psi_k^i + \sum_{k} x_e \lambda_k (y_k - (x_e^T \Psi_k^i)) \quad \Psi_1^i = 0
\]

(15)

\[
\Sigma_{k+1}^i = \Sigma_k - \frac{\lambda_k x_e^T (x_e^T \Sigma_k x_e)}{1 + \lambda_k x_e^T x_e} \quad \Sigma_1 = \Omega I_{(n+1) \times (n+1)}
\]

(16)

where \(I\) is a \((n+1) \times (n+1)\) identity matrix, \(\Omega\) denotes a large number, usually \(\Omega = 1000\), and \(\Sigma\) a dispersion matrix.

When a new fuzzy rule is added, a new dispersion matrix is initiated \(\Sigma_{k+1}^R = I \Omega\). Parameters of the new rules are approximated from the parameters of the existing \(R\) fuzzy rules as [1]:

\[
\Psi_{k+1}^R = \sum_{i=1}^{R} \lambda_i \Psi_{k+1}^i
\]

(17)

Otherwise, parameters of all other rules are inherited from the previous time step, while the dispersion matrices are updated independently. Finally, with the consequent parameters, the prediction of the output is obtained by Equation (4).

The use of the recursive least squares algorithm depends of the initial values of the parameters \(\Psi_0\), and of the initial values of the entries of the dispersion matrix \(\Sigma_0\). These initial values are chosen based on: i) existence of previous knowledge about the system, exploring a database to find an initial rule base.
and, consequently, $\Psi_0$ and $\Sigma_0$; ii) a useful technique when no previous information is to choose large values for the entries of matrix as described above. In this paper, we use the first option, that is, we use a database to choose the initial rule base and its parameters. However, different from [17] which propose the use of the fuzzy c-means algorithm, in this paper we suggest the use of subtractive clustering algorithm [10] to determine the initial rule base structure.

C. MIMO ePL Algorithm

The detailed steps of the MIMO ePL system are as follows:

1. Initialization: if data is available beforehand, then use SC algorithm, otherwise assume the first data point as the cluster center and create the first rule.
2. Read new data.
3. Compute the compatibility index.
4. Compute the arousal index.
5. If the actual model is coherent, then update the cluster with the highest compatibility, else create a new cluster/rule.
7. Compute rule consequent parameters.
8. Compute the rule base outputs.

All the steps of the algorithm are non-iterative. The model can develop/evolve an existing model when the data pattern changes, and by being recursive it means that is computationally efficient. Moreover, besides the use of SC algorithm to determine the initial rule base structure, it is a step applied only when a prior knowledge about the system is available, otherwise the learning process can start from a single data point and improve the performance of the model predictions online [5].

III. YIELD CURVE FORECASTING

The MIMO evolving participatory learning fuzzy approach has been tested using a financial application testbed: forecasting of the term structure of interest rates. The interest rate market is a complex, uncertain, and nonlinear dynamical system. Financial and macroeconomic variables as interest rates are inherently noisy, nonstationary, and deterministically chaotic. The unavailability of complete information from the past behavior of interest rates to fully capture the dependency between future and past rates depict the noisy characteristic. The information that is not included in the model is considered as noise. The nonstationary characteristic implies that the distribution of interest rate is changing over time. By deterministically chaotic, one means is that stock price is short-term random but long-term deterministic [18]. These characteristics enforce that high adaptive systems are required to capture the interest rate dynamics. Hence, we address the use of the MIMO ePL fuzzy system to forecast the yield curve in the US fixed income market.

In the finance and economics literature we find recent applications of evolving fuzzy rule-based models. For instance, [7] uses different evolving fuzzy structures to estimate the Value-at-Risk (VaR) and compare its accuracy against GARCH models using the Sâ£o Paulo Stock Exchange index data. They showed that evolving fuzzy modeling outperforms traditional benchmarks for VaR estimation according to the number of failures. Similar results were found for sovereign bonds modeling [8]. Financial time series forecasting was addressed by [22] using an eTS model with memory for modeling and prediction of GBP/EUR closing price data and US gross domestic product data. The authors conclude that the predictive power of eTS with memory is higher, and its benefits can be appropriately exploited. Recently, [21] suggested an evolving fuzzy systems modeling approach for fixed income option pricing. Results, based on error measures and statistical tests, reveal that evolving fuzzy models outperform traditional methods based on Black-Scholes closed-form formula and some alternative neural network approaches.

Therefore, to forecast the term structure of interest rates with MIMO ePL, the behavior of the interest rate was described by the Nelson and Siegel [23] (NS) factor model, in which the spot interest rate is function of the maturity and some latent factors. For forecasting, the two-stage procedure proposed by Diebold and Li [11] (DL) was adopted. The methodology is as follows.

A. Interest Rates Factor Model

Motivated by the fact that the yield curve is essentially humped, Nelson and Siegel [23] proposed the following exponential expansion function form to describe the instantaneous forward rate curve at time $t$ as function of maturity $\tau$:

$$f_t(\tau) = \beta_{1t} + \beta_{2t} e^{-\lambda_t \tau} + \beta_{3t} \lambda_t e^{-\lambda_t \tau}$$  \hspace{1cm} (18)

where $\beta_{1t}$, $\beta_{2t}$, $\beta_{3t}$ and $\lambda_t$ are the parameters.

The spot rate on a zero-coupon bond with $\tau$ periods to maturity, denoted by $y_t(\tau)$, is obtained by integrating the forward rate function in expression (18):

$$y_t(\tau) = \beta_{1t} + (\beta_{2t} + \beta_{3t}) \left( \frac{1 - e^{-\lambda_t \tau}}{\lambda_t \tau} \right) + \beta_{3t} \left( -e^{-\lambda_t \tau} \right)$$  \hspace{1cm} (19)

One of the key feature of the NS function is that the yield curve converges to $\beta_{1t}$ as maturity increases, while it converges to $\beta_{1t} + \beta_{2t}$ as maturity decreases to 0, which correspond the interpretation of both very long and very short-term spot interest rates, respectively. The parameter $\lambda_t$ governs the exponential decay rate, which small (large) values of $\lambda_t$ produce slow (fast) decay and can better fit the curve at long (short) maturities. Nevertheless, $\lambda_t$ also governs where the loading on $\beta_{3t}$ achieves its maximum.$^2$

B. Two-stage Procedure Forecasting Method

The main idea of the DL procedure is to model and forecast the NS factors as univariate autoregressive $AR(1)$ process. Considering the parameters’ time series, its is supposed that each one follow an $AR(1)$ process, thus since one has the

$^2$Literature interpret $\beta_{1t}$, $\beta_{2t}$ and $\beta_{3t}$ as three latent dynamic factors, defined, respectively, as level, slope and curvature factors [11].
factors forecasts consequently the whole curve is obtained. Therefore, forecasts are obtained from:

\[ \hat{\theta}_{i,t} = \hat{\alpha}_0 + \hat{\alpha}_1 \theta_{i,t-1} \]  (20)

where \( \theta = \{\beta_1, \beta_2, \beta_3, \lambda\} \) is set of the latent factors, and \( \alpha_0 \) and \( \alpha_1 \) are the autoregressive parameters.

DL proposed a simpler procedure. First, one estimates the parameters set \( \theta \), for all \( t \), by some nonlinear optimization algorithm considering the following objective function:

\[ \min_{\theta} f(\theta) = \min_{\beta, \lambda} \sum_{j=1}^{N} (y_j - \hat{y}_j)^2 \quad i = 1, 2, 3 \]  (21)

subject to:

\[ \beta_{1t} > 0 \]
\[ \beta_{1t} + \beta_{2t} > 0 \]  (22)

where \( y_j \) and \( \hat{y}_j \) is the actual and fitted spot interest rates, respectively, and \( N \) is the sample size.

The first step constructs a time series of each latent factor. In the second step all factors are modeled according to equation (20) to produce forecasts and obtain the predicted yield curve as a whole. Notice that the data used in these steps (to perform a time series of all parameters and calibrate each autoregressive processes) correspond to a training set, and forecasts are obtained using out-of-sample data.

IV. Computational Results and Analysis

A. Data

Data of US zero coupon bond yields with maturities of 1, 3, 6, 12, 24, 36, 60, 84 and 120 months were collected from January 1987 to March 2009\(^3\). The database was split into two parts. The first one is composed of 197 monthly observation from January 1987 to June 2003 forming the in-sample period in which the estimations of the models are made. The second, from July 2003 to March 2009, is the out-of-sample period in which the forecasting power of the models is evaluated\(^4\).

B. Estimation

To evaluate the potential of MIMO ePL modeling approach for the term structure of interest rate forecasting we used the two-stage procedure as detailed above. In the second stage all latent factors were modeled with a MIMO ePL model (see Figure 1) instead of estimating each one by an autoregressive process as in [11]. Therefore, time series of the parameters is the one produced by the nonlinear least squares algorithm. The second stage was performed using the MIMO ePL. The results are compared against the DL approach and the eTS+ methodology [1].

\(^3\)These data are available at: http://www.federalreserv.gov/econoresdata/releases/statisticsdata.htm.

\(^4\)It should be noted that this procedure is not necessary for evolving models. However, for comparisons purposes, the out-of-sample set was considered the same for all models.
the high fit obtained by MIMO ePL, the results must also be evaluated in terms of out-of-sample forecasts.

Table I presents the in-sample mean squared error (MSE) for the yield curve factors. The table shows that MIMO ePL has the lowest MSE compared to the other methods, indicating superior performance.

Table II shows the out-of-sample mean squared error (MSFE) for 1-, 3-, and 6-month ahead forecasting for maturities of 1, 3, 6, and 12 months. The results indicate that MIMO ePL consistently outperforms the other methods, with the smallest MSFE values.

In the estimated curves, slight differences may result in significant differences when pricing bonds, which require high accuracy of interest rate forecasts.

V. Conclusion

This paper has considered the extension of the concept of evolving participatory learning (ePL) for MIMO systems. The use of subtractive clustering algorithm to build an initial rule base when data is available in advance was also suggested. Participatory learning appears in the evolving fuzzy participatory procedure as an unsupervised clustering mechanism. Clustering is a key step in evolving fuzzy modeling because it provides a mechanism to extract knowledge in data in the form of a group structure. Participatory clustering is dynamic and provides an efficient mechanism to implement self-organization. This paper has also addressed the application of MIMO ePL to the yield curve forecasting using US market data. Results showed the high capability of MIMO ePL to capture interest rate dynamics, outperforming conventional benchmarks. Future work shall examine enhanced adaptive mechanisms such as input variable parameters selection to...
increase autonomy of the modeling approach and to alleviate users from parameters choice.

ACKNOWLEDGMENT

The authors are grateful to the Brazilian Ministry of Education (CAPES), the Brazilian National Research Council (CNPq) grant 304596/2009-4, and the Foundation for Research Support of the State of São Paulo (FAPESP) for their support.

REFERENCES


