

# Evolving fuzzy granular modeling from nonstationary fuzzy data streams

Daniel Leite · Rosangela Ballini · Pyramo Costa ·  
Fernando Gomide

Received: 10 November 2011 / Accepted: 24 January 2012 / Published online: 18 February 2012  
© Springer-Verlag 2012

**Abstract** Evolving granular modeling is an approach that considers online granular data stream processing and structurally adaptive rule-based models. As uncertain data prevail in stream applications, excessive data granularity becomes unnecessary and inefficient. This paper introduces an evolving fuzzy granular framework to learn from and model time-varying fuzzy input–output data streams. The fuzzy-set based evolving modeling framework consists of a one-pass learning algorithm capable to gradually develop the structure of rule-based models. This framework is particularly suitable to handle potentially unbounded fuzzy data streams and render singular and granular approximations of nonstationary functions. The main objective of this paper is to shed light into the role of evolving fuzzy granular computing in providing high-quality approximate solutions from large volumes of real-world online data streams. An application example in weather temperature prediction using actual data is used to evaluate and illustrate the usefulness of the modeling approach. The behavior of nonstationary fuzzy data streams with gradual

and abrupt regime shifts is also verified in the realm of the weather temperature prediction.

**Keywords** Fuzzy data stream · Granular computing · Information granule · Online learning · Time series prediction

## 1 Introduction

Theories and methodologies that make use of granules to solve problems featured by supplying huge amount of data, information and knowledge label a new area of multi-disciplinary study called granular computing (Bargiela and Pedrycz 2002; Pedrycz 2007; Yao 2008; Lin 2002; Zadeh 1979; Yao et al. 2007). Granular computing as a paradigm of information processing spotlights multiple levels of data detailing to provide useful abstractions and approximate solutions to hard real-world problems (Pedrycz et al. 2008; Bargiela and Pedrycz 2005, 2008; Yao 2010). Particularly, this paper lays emphasis on online granular modeling of time-varying fuzzy data streams.

Data streams have become available in increasing amounts. The ability to analyze them holds the premise that it is possible to outline a fraction of the data which carries unprecedented information to understand the very nature of the underlying system (Angelov and Filev 2004; Beringer and Hullermeier 2007; Bouchachia 2010; Kasabov 2007; Lughofer and Angelov 2011). Evolving granular modeling (Pedrycz 2010; Angelov and Zhou 2008; Leite and Gomide 2012; Leite et al. 2010a, b, 2011, 2012a; Lemos et al. 2011; Rubio 2010;) comes not only as an approach to capture the essence of stream data but also as a framework to extrapolate spatio-temporal correlations from lower-level raw data and provide a more abstract human-like representation

---

D. Leite (✉) · F. Gomide  
School of Electrical and Computer Engineering,  
University of Campinas, Campinas, SP, Brazil  
e-mail: danfl7@dca.fee.unicamp.br

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

R. Ballini  
Institute of Economics, University of Campinas,  
Campinas, SP, Brazil  
e-mail: ballini@eco.unicamp.br

P. Costa  
Graduate Program in Electrical Engineering, Pontifical Catholic  
University of Minas Gerais, Belo Horizonte, MG, Brazil  
e-mail: pyramo@pucminas.br

of them. Research effort into granular computing toward online environment-related tasks is supported by a manifold of relevant applications such as financial, health care, video and image processing, GPS navigation, click stream analysis.

Elementary processing units in evolving granular systems are referred to as information granules. A granule defines a subset of a universal set which holds an internal representation. A granular structure is a family of granules closed by union (Bargiela and Pedrycz 2002; Yao et al. 2008). In online data modeling, arriving data are responsible for creating and expanding granules, guiding parameter adaptation, and finding appropriate model granularity. Algorithms to handle online data streams face odd challenges concerned to the value of the current knowledge, which reduces as the concept changes, and the need to neither store nor retrieve the data once read.

Granular systems have appeared under different names in related fields such as interval analysis, fuzzy and rough sets, divide and conquer, quotient space theory, information fusion, and others (see Yao et al. 2007). However, structurally-adaptive granular systems have only been formally investigated in the early 2000s. Currently, a number of evolving granular systems have succeeded in dealing with time-varying numeric data by means of recursive clustering algorithms and adaptive local models. Notwithstanding, these systems are often unable to process granular data, e.g. fuzzy data, and realize granule-stream-oriented computing. In this paper we address granular systems modeling with a fuzzy-data-stream-driven recursive algorithm in unknown nonstationary environments.

The fuzzy set based evolving modeling (FBeM) framework employs fuzzy granular models to deal with more detailed fuzzy granular data and therefore provide a more intelligible exposition of the data. For each granular model there exists an associated fuzzy rule base. The antecedent part of FBeM rules consists of fuzzy hyperboxes, which are interpretable transparent descriptors of input granular data. The consequent part of FBeM rules has a linguistic and a functional component. The linguistic component arises from fuzzy hyperboxes formed by output data granulation. It facilitates model interpretation and encloses possible model outputs. The functional component is derived from input data and real-valued local functions. This component produces more accurate approximants. The rationale behind the FBeM approach is that it looks to input–output data streams under different resolutions and decide when to adopt coarser or more detailed granularities. Our experimental goal in this paper is to provide predictions of monthly mean, minimum, and maximum temperatures in regions known by their different climatic patterns.

The remainder of this paper is organized as follows. Section 2 overviews works related with granular data

stream modeling. Section 3 introduces the FBeM framework. Section 4 addresses weather temperature predictions developed by FBeM and alternative approaches using actual temperature time series data. Section 5 concludes the paper and suggests issues for future research.

## 2 Related works

This section summarizes recent research related to incremental learning methods that are capable to handle granular data streams. We do not intend to give an exhaustive literature review. The purpose is to overview works closely related with the approach suggested in this paper.

Interval based evolving modeling (IBeM) (Leite et al. 2012a) is an interval granular approach whose focus is to enclose imprecise data streams and produce a rule-based summary. IBeM emphasizes imprecise data manifesting as tolerance intervals and recursive learning procedures grounded in fundamentals of interval mathematics. Antecedent and consequent parts of IBeM rules are interval hyperboxes, which are linked by an interval granular mapping—or inclusion function in the interval analysis terminology. The IBeM approach for function approximation makes no specific assumption about the properties of the data. Structural development is fully guided by interval data streams. Applications in actual meteorological and financial time series (Leite et al. 2012a) have shown the usefulness of the approach.

General fuzzy min–max neural network (GFMM) (Gabrys and Bargiela 2000) is a generalization of the fuzzy min–max clustering and classification neural networks (Simpson 1992, 1993). It handles labeled and unlabeled data simultaneously to develop a single neural network structure. GFMM combines supervised and unsupervised learning to perform hybrid clustering and classification. Learning can be done in one pass over data sets and data can be intervals. Basically, the GFMM algorithm places and gradually adjusts fuzzy hyperboxes in the feature space using the expansion-contraction paradigm.

Granular reflex fuzzy min–max neural networks (GrRFMN) (Nandedkar and Biswas 2009) learn from and classify interval granular data in online mode. The structure of the GrRFMN network simulates the reflex mechanism of the human brain and deals with class overlapping using compensation neurons. The GrRFMN training algorithm gives a way to calculate datum-model membership degree which potentially leads better overall network performance. Experiments with real data sets assert the effectiveness of the approach.

Uncertain micro-clustering algorithm (UMicro) (Aggarwal et al. 2008) considers that stream data arrive together with their underlying standard error instead of

assuming the entire probability distribution function of the data is known. The algorithm uses uncertainty information to improve the quality of the underlying results. UMicro incorporates a time decay method to update the statistics of micro-clusters. The decaying method is especially useful to model drifting concepts in evolving data streams. The efficiency of the UMicro approach has been demonstrated in a variety of data sets.

Evolving granular neural network (eGNN) (Leite et al. 2010a, 2012b) is an approach derived from a parallel research we have conducted on fuzzy granular data stream mining and modeling. eGNN uses fuzzy granules and fuzzy aggregation neurons for information fusion. Its learning algorithm is committed to build and incrementally adapt the network using data to approximate nonstationary functions. Application examples in pattern recognition and forecasting in material and biomedical engineering have shown eGNN can outperform alternative online approaches in terms of accuracy and compactness.

### 3 Fuzzy set based evolving modeling

FBeM was first suggested in Leite and Gomide (2012) as a general framework for function approximation and robust control. Later, its learning algorithm was modified to handle time-series prediction (Leite et al. 2011). Both cases assume numeric (singular) data streams. In this paper, we supply FBeM with a recursive incremental algorithm suited to deal with time-varying fuzzy data stream.

The commitment of FBeM is to deliver simultaneous singular and granular function approximation and linguistic description of the behavior of a system. Local FBeM models are a set of If-Then rules developed incrementally from input–output data streams. Learning can start from scratch and, as new information is brought by the data stream, granules and rules are created and their parameters adjusted. Therefore, FBeM becomes more flexible to handle data so that redesign and retraining models all along are needless. The resulting input–output granular mapping may be eventually refined or coarsened according to inter-granules relationships and error indices.

#### 3.1 Problem statement

The generic form of the problem addressed in this paper is as follows:

Given a time-varying unknown function  $f^{[h]}$ , where  $h = 1, \dots$  is the time index; and a pair of observations  $(x, y)^{[h]}$ ,  $x \in X$  and  $y \in Y$ , find a finite collection of information granules  $\gamma = \{\gamma^1, \dots, \gamma^c\}$  and a time-varying real-

valued map  $p^{[h]} : X \rightarrow Y$  such that  $\gamma^i \subseteq X \times Y$  and  $p^{[h]}$  minimizes  $(f^{[h]} - p^{[h]})^2$ .

We assume the following: (1) the output  $y^{[h]}$  does not need to be known when the input  $x^{[h]}$  is available, but must be known afterwards; (2) attributes  $x_j$  of an input vector  $x = (x_1, \dots, x_n)$  and the output  $y$  are considered trapezoidal fuzzy data. Triangular, interval and numeric types of data arise as particular arrangements of trapezoids; (3) spatio-temporal constraints: data streams are not stored (space constraint); the per-sample latency of algorithms should not be larger than the time interval between samples (time constraint).

#### 3.2 Fuzzy data stream

Empirical data may take various forms depending on how they are modeled formally, e.g., intervals, probability distributions, fuzzy numbers (Dubois and Prade 2004). Fuzzy data arise when measurements are inaccurate, variables are hard to be quantified, pre-processing steps introduce uncertainty to numeric data or when the data are derived from expert knowledge. Often, data are purely numerical, but the process which generated the data can be uncertain.

A fuzzy interval is a fuzzy set on the real line that satisfies the conditions of normality ( $G(x) = 1$  for at least one  $x \in \mathfrak{R}$ ) and convexity ( $G(\kappa x^1 + (1 - \kappa) x^2) \geq \min \{G(x^1), G(x^2)\}$ ,  $x^1, x^2 \in \mathfrak{R}$ ,  $\kappa \in [0,1]$ ).

This paper considers data streams of fuzzy intervals whose membership function are trapezoidal. A trapezoidal fuzzy interval can be represented by a quadruple  $(\underline{x}, \underline{x}, \bar{x}, \bar{x})$ . It satisfies a series of properties such as normality, unimodality, continuity, and boundedness of support (Pedrycz et al. 2007). Fuzzy granular data streams generalize singular (numeric) data streams by allowing fuzziness.

#### 3.3 Structure and processing

Rules  $R^i$  governing FBeM information granules  $\gamma^i$  are of the type:  $R^i : \text{IF } (x_1 \text{ is } A_1^i) \text{ AND} \dots \text{AND } (x_j \text{ is } A_j^i) \text{ AND} \dots \text{AND } (x_n \text{ is } A_n^i) \text{ THEN } \underbrace{(y \text{ is } B^i)}_{\text{linguistic}} \text{ AND } \underbrace{y = p^i(x_1, \dots, x_n)}_{\text{functional}}$ ,

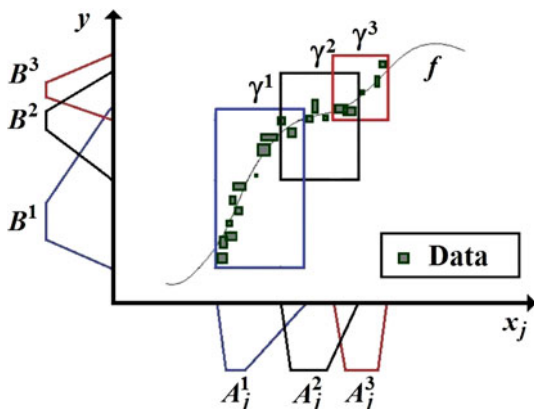
where  $A_j^i$  and  $B^i$  are membership functions built in light of input and output data being available;  $p^i$  is a local approximation function. The collection of rules  $R^i$ ,  $i = 1, \dots, c$ , casts a rule base. Rules in FBeM are created and adapted on-demand whenever the data asks for improvement in the current model. Notice that an FBeM rule combines both, linguistic and functional consequents. The linguistic part of the consequent favors interpretability once fuzzy sets may come with a label. The functional part of the consequent offers accuracy. Thus, FBeM takes

advantage of both, linguistic and functional consequents, within a single framework.

Fuzzy sets  $A_j^i$  and  $B^i$  are generated from scattered fuzzy granulation. The scattering approach clusters the data into fuzzy sets when appropriate and takes into account the coexistence of a manifold of granularities in the data stream. Sets  $A_j^i$  and  $B^i$  can be easily extended to fuzzy hyperboxes  $\gamma^i$  (granules) in a product space. Granules are positioned at locations populated by input and output data in the product space. Figure 1 illustrates the scatter granulation mechanism of fuzzy data. Note in the figure that the granularity of models is coarser than the granularity of data. This is to obtain data compression and to provide a more effective, human-intelligible representation.

Fitting data into conveniently placed and sized granules through scattering leaves substantial flexibility for incremental learning. The FBeM approach grants freedom in choosing the internal structure of granules.

Yager et al. (2007) and Yager (2009) has demonstrated that a trapezoidal fuzzy set  $A_j^i = (l_j^i, \lambda_j^i, \Lambda_j^i, L_j^i)$  allows the modeling of a wide class of granular objects. A triangular fuzzy set is a trapezoid where  $\lambda_j^i = \Lambda_j^i$ ; an interval is a trapezoid where  $l_j^i = \lambda_j^i$  and  $\Lambda_j^i = L_j^i$ ; a singleton (singular datum) is a trapezoid where  $l_j^i = \lambda_j^i = \Lambda_j^i = L_j^i$ . Additional features that make the trapezoidal representation attractive comprise: (1) ease of acquiring the necessary parameters. Only four parameters need to be captured; (2) many operations on trapezoids can be performed using the endpoints of intervals which are level sets of trapezoids. Moreover, the piecewise linearity of the trapezoidal representation allows calculation of only two level sets, corresponding to the core and support, respectively, to obtain a complete implementation; (3) trapezoids are easier to be translated to linguistic labels.



**Fig. 1** Scattering approach for fuzzy data granulation

Fuzzy sets  $B^i = (u^i, v^i, \Upsilon^i, U^i)$  are used to assemble granules in the output space. The local function  $p^i$  is adapted for samples that rest inside the granule  $\gamma^i$ . In general, functions  $p^i$  can be of different type and are not required to be linear. Here we assume affine functions:

$$p^i = a_0^i + \sum_{j=1}^n a_j^i x_j, \quad (1)$$

for simplicity. If higher order functions are used to approximate  $f$ , then the number of coefficients to be estimated increases, especially when the number of input variables  $n$  is large. The recursive least squares (RLS) algorithm is used to adjust the coefficients  $a_j^i$  of  $p^i$ .

Trapezoidal fuzzy sets and scatter granulation allow granules to overlap. Therefore, two or more granules can accommodate the same data sample. FBeM singular output is found as the weighted mean value

$$p = \frac{\sum_{i=1}^c \min(A_1^i(x_1), \dots, A_n^i(x_n)) p^i(x_1, \dots, x_n)}{\sum_{i=1}^c \min(A_1^i(x_1), \dots, A_n^i(x_n))}. \quad (2)$$

Granular output is given by the convex hull of output fuzzy sets  $B^{i^*}$ , where  $i^*$  are indices of granules that can accommodate the data sample. The convex hull of trapezoidal fuzzy sets  $B^1, \dots, B^c$  is given as follows:

$$\text{ch}(B^1, \dots, B^c) = (\min(u^1, \dots, u^c), \min(v^1, \dots, v^c), \max(\Upsilon^1, \dots, \Upsilon^c), \max(U^1, \dots, U^c)). \quad (3)$$

The granular output given by  $B^{i^*}$  enriches decision making and motivates interpretability. While being specific from  $p$  we risk to be incorrect, being unspecific from  $B^{i^*}$  increases our confidence to be correct.

### 3.4 Setting the granularity

The width of a fuzzy set  $A_j^i$  is defined as the length of its support,

$$\text{wdt}(A_j^i) = L_j^i - l_j^i. \quad (4)$$

The maximum width fuzzy sets  $A_j^i$  are allowed to expand is denoted by  $\rho$ , that is,  $\text{wdt}(A_j^i) \leq \rho$ ,  $j = 1, \dots, n$ ;  $i = 1, \dots, c$ . Values of  $\rho$  ensue different representations of the same problem in different levels of detail (granularities).

Let the expansion region of a set  $A_j^i$  be denoted by

$$E_j^i = \left[ \text{mp}(A_j^i) - \frac{\rho}{2}, \text{mp}(A_j^i) + \frac{\rho}{2} \right], \quad (5)$$

where

$$\text{mp}(A_j^i) = \frac{\lambda_j^i + \Lambda_j^i}{2} \quad (6)$$

is the midpoint of  $A_j^i$ . Expansion regions help to derive criteria for deciding whether or not data samples should be in the same granule.

For normalized data,  $\rho$  takes values in  $[0,1]$ . If  $\rho$  is equal to 0, then FBeM granules do not enlarge. Learning creates a new rule for each sample, what may cause overfitting, that is, excessive complexity and irreproducible optimistic results. If  $\rho$  equals 1, then a single granule may cover the entire data domain so that FBeM becomes unable to handle nonstationarities. Meaningful life-long adaptability is reached choosing intermediate values for  $\rho$ .

In the most general case, FBeM starts learning with an empty rule base and devoid of knowledge about the data. It is reasonable in this case to initialize  $\rho$  halfway to yield structural stability and plasticity equally. We consider  $\rho^{[0]} = 0.5$  as the default initial value.

A fast procedure to evolve  $\rho$  over time is as follows. Let  $r$  be the difference between the current number of granules and the number of granules  $h_r$  steps earlier,  $r = c^{[h]} - c^{[h-h_r]}$ . If the quantity of granules grows faster than a given rate  $\eta$ , that is,  $r > \eta$ , then  $\rho$  is increased,

$$\rho(\text{new}) = \left(1 + \frac{r}{h_r}\right)\rho(\text{old}). \tag{7}$$

The idea here is to reject large rule bases because they increase model complexity and may not help generalization. Equation (7) controls  $\rho$  and acts against outbursts of growth.

If the number of granules grows at a rate smaller than  $\eta$ , that is,  $r \leq \eta$ , then  $\rho$  is decreased as follows:

$$\rho(\text{new}) = \left(1 - \frac{(\eta - r)}{h_r}\right)\rho(\text{old}). \tag{8}$$

With this mechanism we maintain a data-dependent fluctuating granularity. Alternative heuristic approaches to evolve the value of  $\rho$  over time take into account estimation errors and their derivatives as addressed in Leite et al. (2011).

Reducing the maximum width allowed for granules requires shrinking larger granules to fit them to the new value. In this case, the support of a set  $A_j^i$  is narrowed as follows:

$$\text{If } \text{mp}(A_j^i) - \frac{\rho(\text{new})}{2} > l_j^i \text{ then } l_j^i(\text{new}) = \text{mp}(A_j^i) - \frac{\rho(\text{new})}{2}$$

$$\text{If } \text{mp}(A_j^i) + \frac{\rho(\text{new})}{2} < L_j^i \text{ then } L_j^i(\text{new}) = \text{mp}(A_j^i) + \frac{\rho(\text{new})}{2}.$$

Cores  $[l_j^i, L_j^i]$  are handled similarly. Time-varying granularity is useful to avoid guesses on how fast and how often the data stream changes.

### 3.5 Time granulation

Time granulation aims at both, reducing the sampling rate of fast data streams, and synchronizing concurrent data streams that are input at random time intervals. A time granule describes the data for a certain time period. Whenever the bounds of a time granule are aligned with significant shifts in the target function, the underlying granulation provides a good abstraction of the data. Conversely, if the alignment is poor, models may be inadequate. Manifold granularities require temporal reasoning and respective formalizations.

Broadly stated, information evoked from time granules can be bounds of intervals, probability distributions or membership functions, and features such as frequency and correlation between events, patterns, prototypes. The internal structure of a granule and its associated variables provide full description and characterization of the granule.

Consider a fuzzy data stream  $(x, y)^{[h]}$ ,  $h = 1, \dots$ . Time granulation groups a set of successive instances  $(x, y)^{[h]}$ ,  $h = h_b, h_{b+1}, \dots, h_e$ , where  $h_b$  and  $h_e$  denote the lower and upper bounds of a time interval  $[h_b, h_e]$ . The set of instances input during  $[h_b, h_e]$  produces a unique granule  $\gamma^{[H]}$  whose corresponding fuzzy sets are

$$A_j^{[H]} = \left( \min\left(\underline{x}_j^{[h_b]}, \dots, \underline{x}_j^{[h_e]}\right), \min\left(\underline{x}_j^{[h_b]}, \dots, \underline{x}_j^{[h_e]}\right), \max\left(\bar{x}_j^{[h_b]}, \dots, \bar{x}_j^{[h_e]}\right), \max\left(\bar{x}_j^{[h_b]}, \dots, \bar{x}_j^{[h_e]}\right) \right), \tag{9}$$

and  $B^{[H]}$  which is constructed similarly from the output stream. Instances falling within  $A_j^{[H]}$ ,  $j = 1, \dots, n$ , and  $B^{[H]}$  are considered indiscernible and the inequalities

$$\text{wdt}(A_j^{[H]}) \leq \rho, \quad j = 1, \dots, n, \quad \text{and} \quad \text{wdt}(B^{[H]}) \leq \rho \tag{10}$$

hold true.

Whenever input data arrive at different rates, for example,  $x_1$  arrives at each 2 s and  $x_2$  at each 10 s, or the amount of data exceeds the affordable computational cost (e.g. in high-frequency applications), we resort to granulated views of the time domain. Thereafter, rule construction is based on the resulting fuzzy granules,  $A_j^{[H]}$  and  $B^{[H]}$ , rather than on original data  $(x, y)^{[h]}$ . FBeM does not need to be exposed to all original data, which are far more numerous than time granules.

### 3.6 Creating granules

No rule does necessarily exist before learning starts. The incremental procedure to create rules runs whenever at least one entry of an input  $(x_1, \dots, x_n)$  does not belong to expansion regions  $(E_1^i, \dots, E_n^i)$ ,  $i = 1, \dots, c$ . Otherwise, the current rule base is not modified. When the output  $y \notin E^i$ , it should be enclosed by a new granule.

A new granule  $\gamma^{c+1}$  is assembled from fuzzy sets  $A_j^{c+1}$  and  $B^{c+1}$  whose parameters match the sample, that is,

$$\begin{aligned} A_j^{c+1} &= (l_j^{c+1}, \lambda_j^{c+1}, \Lambda_j^{c+1}, L_j^{c+1}) = (\underline{x}_j, \underline{x}_j, \bar{x}_j, \bar{x}_j), \\ B^{c+1} &= (u^{c+1}, v^{c+1}, \Upsilon^{c+1}, U^{c+1}) = (\underline{y}, \underline{y}, \bar{y}, \bar{y}). \end{aligned} \quad (11)$$

Coefficients of the real-valued local function  $p^{c+1}$  are set to

$$a_0^{c+1} = \text{mp}(y); \quad a_j^{c+1} = 0, \quad j \neq 0. \quad (12)$$

### 3.7 Adapting granules

Adaptation of granules either expands or contracts the support and the core of fuzzy sets  $A_j^i$  and  $B^i$  to enclose new data, and simultaneously refines the coefficients of local functions  $p^i$  to increase accuracy. A granule is chosen to be adapted whenever an instance of the data stream falls within its expansion region. In situations in which two or more granules are qualified to enclose the data, adapting only one of the granules is enough.

Data and granules are fuzzy objects of trapezoidal nature. A useful similarity measure for trapezoids is:

$$\begin{aligned} S(x, A^i) &= 1 - \frac{1}{4n} \sum_{j=1}^n (|\underline{x}_j - l_j^i| + |\underline{x}_j - \lambda_j^i| + |\bar{x}_j - \Lambda_j^i| \\ &\quad + |\bar{x}_j - L_j^i|). \end{aligned} \quad (13)$$

This measure quantifies the degree that input data match the current knowledge. It returns 1 for identical trapezoids and decreases linearly when  $x$  and  $A^i$  move away from each other. Naturally, among all granules qualified to accommodate a particular sample, the one with highest similarity should be chosen. This procedure prevents conflict and helps to keep the FBeM construction simple.

Adaptation proceeds depending on how far an input datum  $x_j$  is from fuzzy set  $A_j^i$ , namely,

$$\begin{aligned} \text{If } \underline{x}_j &\in [\text{mp}(A_j^i) - \frac{\rho}{2}, l_j^i] && \text{then } l_j^i(\text{new}) = \underline{x}_j \\ \text{If } \underline{x}_j &\in [\text{mp}(A_j^i) - \frac{\rho}{2}, \lambda_j^i] && \text{then } \lambda_j^i(\text{new}) = \underline{x}_j \\ \text{If } \underline{x}_j &\in [\lambda_j^i, \text{mp}(A_j^i)] && \text{then } \lambda_j^i(\text{new}) = \underline{x}_j \\ \text{If } \underline{x}_j &\in [\text{mp}(A_j^i), \text{mp}(A_j^i) + \frac{\rho}{2}] && \text{then } \lambda_j^i(\text{new}) = \text{mp}(A_j^i) \\ \text{If } \bar{x}_j &\in [\text{mp}(A_j^i) - \frac{\rho}{2}, \text{mp}(A_j^i)] && \text{then } \Lambda_j^i(\text{new}) = \text{mp}(A_j^i) \\ \text{If } \bar{x}_j &\in [\text{mp}(A_j^i), \Lambda_j^i] && \text{then } \Lambda_j^i(\text{new}) = \bar{x}_j \\ \text{If } \bar{x}_j &\in [\Lambda_j^i, \text{mp}(A_j^i) + \frac{\rho}{2}] && \text{then } \Lambda_j^i(\text{new}) = \bar{x}_j \\ \text{If } \bar{x}_j &\in [L_j^i, \text{mp}(A_j^i) + \frac{\rho}{2}] && \text{then } L_j^i(\text{new}) = \bar{x}_j. \end{aligned}$$

The first and the eighth rules suggest support expansion while the second and seventh recommend core expansion. The remaining cases advise core contraction.

Operations on core parameters,  $\lambda_j^i$  and  $\Lambda_j^i$ , require further adjustment of the midpoint of the respective granule:

$$\text{mp}(A_j^i)(\text{new}) = \frac{\lambda_j^i(\text{new}) + \Lambda_j^i(\text{new})}{2}. \quad (14)$$

As a result, support contraction may happen in two occasions:

$$\begin{aligned} \text{If } \text{mp}(A_j^i)(\text{new}) - \frac{\rho}{2} &> l_j^i && \text{then } l_j^i(\text{new}) = \text{mp}(A_j^i)(\text{new}) - \frac{\rho}{2} \\ \text{If } \text{mp}(A_j^i)(\text{new}) + \frac{\rho}{2} &< L_j^i && \text{then } L_j^i(\text{new}) = \text{mp}(A_j^i)(\text{new}) + \frac{\rho}{2}. \end{aligned}$$

Adaptation of consequent fuzzy sets  $B^i$  is done similarly using output data  $y$ . Coefficients  $a_j^i$  are updated using the RLS algorithm, as detailed next.

### 3.8 Recursive least squares

The RLS algorithm is used to adapt consequent function parameters  $a_j^i$  as follows.

Let  $(x, y)^{[h]}$  be the sample available for training at instant  $h$ . We adjust the coefficients  $a_j^i$  of  $p^i$  assuming that

$$y^{[h]} = a_0^i + \sum_{j=1}^n a_j^i x_j^{[h]}. \quad (15)$$

Due to the trapezoidal anatomy of  $x_j$  and  $y$ , we rely on their midpoints to adapt the coefficients  $a_j^i$  using the standard form of the RLS algorithm. In the remainder of this sub-section we assume that  $(x, y)^{[h]}$  are real numbers, the midpoints of the trapezoidal fuzzy input–output data.

In the matrix form, the Eq. (15) becomes

$$Y = X\Omega^i, \quad (16)$$

where  $Y = [y]$ ,  $X = [1 \ x_1 \ \dots \ x_n]$ ; and  $\Omega^i = [a_0^i \ \dots \ a_n^i]^T$  is the vector of unknown parameters. To estimate the coefficients  $a_j^i$  we let

$$Y = X\Omega^i + E, \quad (17)$$

where

$$E = [\epsilon] = y - p(x) \quad (18)$$

is the estimation error. While in batch estimation the rows in  $Y$ ,  $X$  and  $E$  increase with the number of available instances, in recursive mode only two rows are kept and we reformulate input and output data, and error as

$$\begin{aligned} Y &= \begin{bmatrix} y^{[h-1]} \\ y^{[h]} \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_1^{[h-1]} & \dots & x_n^{[h-1]} \\ 1 & x_1^{[h]} & \dots & x_n^{[h]} \end{bmatrix} \\ \text{and } E &= \begin{bmatrix} \epsilon^{[h-1]} \\ \epsilon^{[h]} \end{bmatrix}. \end{aligned} \quad (19)$$

Rows in (19) refer to values before and just after adaptation. The RLS algorithm finds  $\Omega^i$  to minimize the functional

$$J(A^i) = E^T E. \quad (20)$$

Derived from Young et al. (1984),  $\Omega^i$  can be estimated by

$$\Omega^i = (X^T X)^{-1} X^T Y. \tag{21}$$

Assuming  $P = (X^T X)^{-1}$  and the matrix inversion lemma (Young et al. 1984), we avoid inverting  $X^T X$  using:

$$P(\text{new}) = P(\text{old}) \left[ I - \frac{X X^T P(\text{old})}{1 + X^T P(\text{old}) X} \right], \tag{22}$$

where  $I$  is identity matrix. In practice it is usual to choose large initial values for the entries of the main diagonal of  $P$ . We use  $P^{[0]} = 10^3 I$  as default value.

After simple mathematical transformations, the vector of parameters is rearranged recursively as follows:

$$\Omega^i(\text{new}) = \Omega^i(\text{old}) + P(\text{new}) X (Y - X^T \Omega^i(\text{old})). \tag{23}$$

Detailed derivation of the RLS algorithm is found in Astrom et al. (1994). For a convergence proof see Johnson (1988).

### 3.9 Coarsening the granular model

Relationships between granules may be strong enough to justify assembling a more abstract granule that inherits the information of lower level granules. The similarity measure (13) can be used to quantify granule-granule resemblance if we restate it as

$$S(A^{i_1}, A^{i_2}) = 1 - \frac{1}{4n} \sum_{j=1}^n (|l_j^{i_1} - l_j^{i_2}| + |\lambda_j^{i_1} - \lambda_j^{i_2}| + |\Lambda_j^{i_1} - \Lambda_j^{i_2}| + |L_j^{i_1} - L_j^{i_2}|) \tag{24}$$

This measure has good discrimination capability and its calculation is fast.

FBeM combines granules in intervals of  $h_r$  steps considering the lowest entry of  $S(A^{i_1}, A^{i_2})$ ,  $i_1, i_2 = 1, \dots, c$ ,  $i_1 \neq i_2$ , and a decision criterion. The decision may be based on whether the new granule obeys the maximum width allowed  $\rho$ .

A new granule  $\gamma^i$ , coarsening of  $\gamma^{i_1}$  and  $\gamma^{i_2}$ , is formed by trapezoidal membership functions  $A_j^i$  with parameters derived from  $A_j^{i_1}$  and  $A_j^{i_2}$  as follows:

$$\begin{aligned} l_j^i &= \min(l_j^{i_1}, l_j^{i_2}) \\ \lambda_j^i &= \min(\lambda_j^{i_1}, \lambda_j^{i_2}) \\ \Lambda_j^i &= \max(\Lambda_j^{i_1}, \Lambda_j^{i_2}) \\ L_j^i &= \max(L_j^{i_1}, L_j^{i_2}). \end{aligned} \tag{25}$$

Granule  $\gamma^i$  encloses all the content of the granules  $\gamma^{i_1}$  and  $\gamma^{i_2}$ . The same coarsening procedure is used to determine the parameters of the output membership

function  $B^i$ . The coefficients of the local function of granule  $\gamma^i$  are

$$a_j^i = \frac{1}{2}(a_j^{i_1} + a_j^{i_2}), \quad j = 0, \dots, n. \tag{26}$$

Combining granules reduces the size of the rule base and eliminates redundancy. The importance of complexity reduction in evolving fuzzy systems is discussed in Lughofer et al. (2011).

### 3.10 Removing granules

A granule should be removed from the system model if it seems to be inconsistent with the current knowledge. A common approach consists in deleting the most inactive granules (Leite et al. 2011).

Let

$$\Theta^i = 2^{(-\psi(h-h_a^i))} \tag{27}$$

be the activity factor associated to the granule  $\gamma^i$ ;  $\psi$  is a decay rate,  $h$  the current time step, and  $h_a^i$  the last time step that granule  $\gamma^i$  was processed. Factor  $\Theta^i$  decreases exponentially when  $h$  increases. The half-life of a granule is the time spent to reduce the factor  $\Theta^i$  by half, that is,  $1/\psi$ .

Half-life  $1/\psi$  is a value that suggests deletion of inactive granules. As a rule,  $\psi$  is domain-dependent. Large values of  $\psi$  express lower tolerance to inactivity and higher privilege of more compact structures. Small values of  $\psi$  add robustness in the sense that they prevent catastrophic forgetting. If the application requires memorization of isolated events or seasonality is expected, then it may be the case to set  $\psi$  to 0 and let granules and rules exist forever. In general,  $\psi$  should be set in  $]0,1[$  to keep model evolution active.

### 3.11 Learning algorithm

The learning procedure to evolve FBeM can be summarized as follows:

```

BEGIN
Set parameters  $\rho, h_r, \eta, \psi, c = 0$ ;
Read  $(x, y)^{[h]}$ ,  $h = 1$ ;
Create granule  $\gamma^{c+1}$ ;
For  $h = 2, \dots$  do
  Read  $(x, y)^{[h]}$ ;
  Provide singular approximation  $p(x^{[h]})$ ;
  Provide granular approximation  $B^{i*}$ ;
  Calculate output error  $\epsilon^{[h]} = y^{[h]} - p(x^{[h]})$ ;
  If  $x^{[h]}$  or  $y^{[h]}$  is not into granules' expanded regions  $E^i \forall i$ 
    Create granule  $\gamma^{c+1}$ ;
  Else
    Adapt the most active granule  $\gamma^i$ ,  $i = \max_i(S(x, A^1), \dots, S(x, A^c))$ ;
    Adapt local function parameters  $a_j^i$  using RLS;
  If  $h = \alpha h_r$ ,  $\alpha = 1, 2, \dots$ 
    Combine granules when feasible;
    Update model granularity  $\rho$ ;
    Remove inactive granules;
END
    
```

#### 4 Application: weather temperature prediction

The example addressed in this section consider fuzzy granular data streams derived from monthly mean, minimum, and maximum temperatures of weather time series of geographic regions with different climatic patterns. The aim is to predict monthly temperatures for all regions.

##### 4.1 Weather prediction

Weather predictions at given locations are useful not only for people to plan activities or protect property, but also to assist decision making in many different sectors such as energy, transportation, aviation, agriculture, commodity markets, inventory planning. Any system that is sensitive to the state of the atmosphere may benefit from weather predictions.

Monthly temperature data carry a degree of uncertainty due to imprecision of atmospheric measurements, instrument malfunction, equivocated transcripts, and different standards in acquiring and pre-processing the collected data. Usually temperature data sets are numerical, but it is known that the processes which originate and supply the data are imprecise. Temperature estimates in finer time granularities (days, weeks) are commonly demanded. The FBeM approach provides guaranteed granular predictions of the time series in these cases. The satisfaction in relation to the granular prediction depends on its compactness. Granular predictions together with singular predictions are important because they convey a value and a range of possible temperature values.

In the experiment we translate average minimum, mean and maximum monthly temperatures into normal triangular fuzzy numbers. We use data collected by the Death Valley (Furnace Creek), Ottawa, and Lisbon weather stations to evaluate FBeM. In Death Valley, super-heated moving air masses are trapped into the valley by surrounding steep mountain ranges creating an extremely dry climate with high temperatures. Refer to Roof and Callagan (2003) for a complete list of factors that produce high air temperatures in Death Valley. Conversely, Ottawa is one of the coldest capitals in the world. During the year, a wide range of temperatures can be observed, but the winters are very cold and snowy. Lisbon experiences more usual weather patterns. Summers are warm, sometimes hot, whereas winters are mild and moist.

The Death Valley, Ottawa and Lisbon data sets consist of 1,302, 1,374, and 1,194 time indexed instances comprising average minimum, mean and maximum temperatures per month recorded from January of 1901, 1895 and 1910, respectively, up to December of 2009. In all experiments described subsequently, FBeM inputs data only

once to build model structure and adjust its parameters. This aims at simulating online data stream processing.

Testing and training are performed concomitantly on a per-sample basis. The performance of algorithms is evaluated using the root mean square error of singular predictions,

$$RMSE = \sqrt{\frac{1}{H} \sum_{h=1}^H (\text{mp}(y)^{[h]} - p^{[h]})^2}, \quad (28)$$

the number of rules in the model structure, and processing (CPU) time in seconds. We used a dual-core 2.54 GHz processor with 4 GB of RAM.

##### 4.2 Comparisons

Representative statistical and computational intelligence algorithms were chosen for performance assessment. The methods used for comparison are: moving average (MA) (Box et al. 2008), square weighted moving average (SWMA) (Box et al. 2008), multilayer perceptron neural network (MLP) (Haykin 1999), evolving Takagi–Sugeno (eTS) (Angelov and Filev 2004), extended Takagi–Sugeno (xTS) (Angelov and Zhou 2006), dynamic evolving neuro-fuzzy inference system (DENFIS) (Kasabov and Song 2002), and FBeM.

The task of the different methods is to provide one step prediction of the monthly temperature  $y^{[h+1]}$ , using the last five observations,  $x^{[h-4]}, \dots, x^{[h]}$ . Online methods employ the sample-per-sample testing-before-training approach as follows. First, an estimation  $p^{[h+1]}$  is derived for a given input  $(x^{[h-4]}, \dots, x^{[h]})$ . One time step after, the actual value  $y^{[h+1]}$  becomes available and model adaptation is performed if necessary. In general, models should be robust to the trend and seasonal components of the time series, and not to the random noise component. Because the observed data contain random noise and irregular patterns, models that do not over fit them produce better generalizations and predictions of future values. Table 1 shows the results for the Death Valley, Ottawa, and Lisbon monthly temperature data. FBeM uses  $\rho = 0.7$ ,  $h_r = 1/\psi = 48$ , and  $\eta = 2$ .

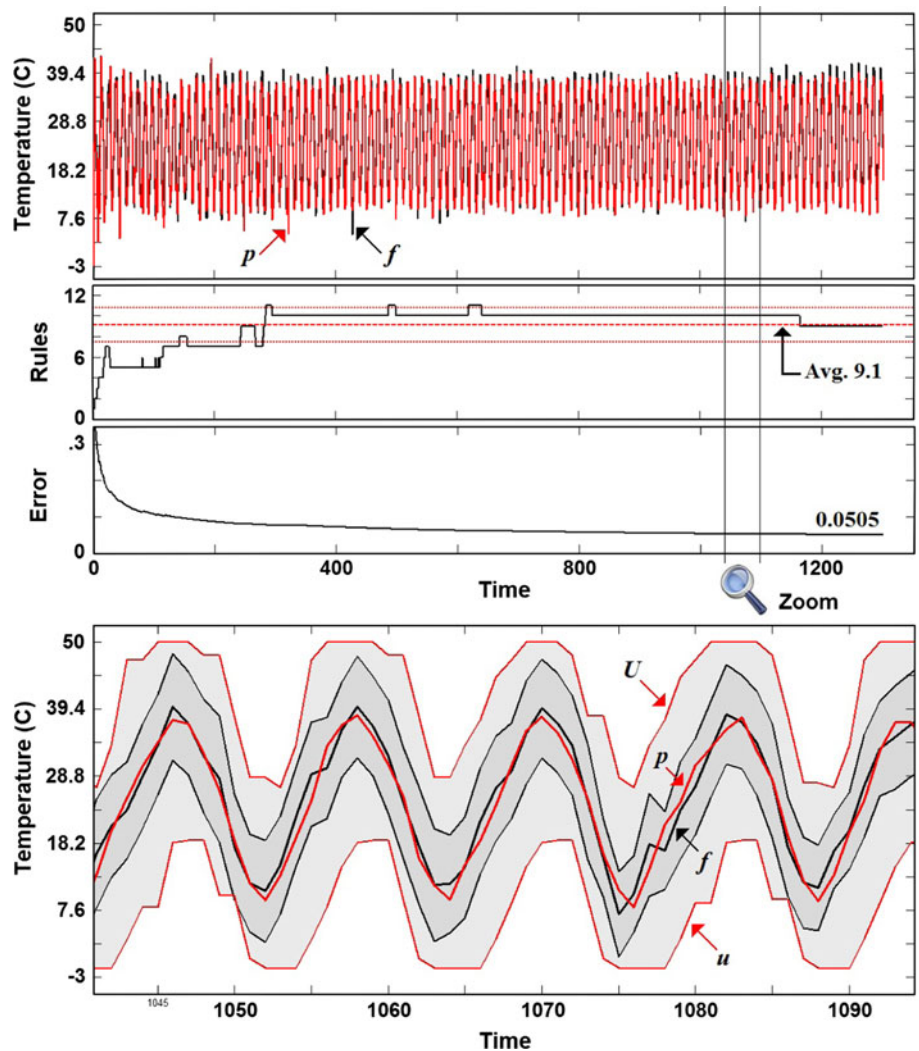
Table 1 summarizes the performance of the different algorithms in one-step prediction of the monthly mean temperature. In particular, FBeM gives more accurate predictions than the remaining methods without necessarily using larger structures. The trend component of the time series is taken into account in FBeM by procedures that gradually adapt granules and rules. The seasonal component is captured through different granules which represent different seasons and transitions between seasons. Since the content of a granule carries seasonal information, its corresponding rule tends to be activated in the specific months.



**Table 1** Temperature prediction performance

Model	Death Valley			Ottawa			Lisbon		
	Rules	RMSE	CPU	Rules	RMSE	CPU	Rules	RMSE	CPU
MA	–	0.1668	0.003	–	0.1624	0.003	–	0.1405	0.003
SWMA	–	0.0826	0.003	–	0.0810	0.003	–	0.0711	0.003
MLP	20	0.0647	13.302	20	0.0839	13.288	20	0.1078	13.432
eTS	5	0.0863	0.895	8	0.0840	1.402	7	0.0935	0.955
xTS	5	0.0856	0.474	11	0.0850	0.410	7	0.0917	0.370
DENFIS	13	0.0681	2.602	23	0.0860	3.670	27	0.0940	3.301
FBeM	9	0.0505	1.362	6	0.0602	1.339	7	0.0673	1.232

**Fig. 2** FBeM prediction of Death Valley temperatures



The evolving approaches eTS, xTS and DENFIS use singular data, the mean temperature. In contrast, FBeM takes into account the mean and neighbor data to bound predictions. The xTS has been the fastest method among the rule-based evolving methods considered here in this paper.

We can also notice in Table 1 that moving average methods are fast and that SWMA is particularly competitive. SWMA can operate in online mode, but it does not provide comprehensible models to support data description and interpretation. The MLP neural network behaved well for the Death Valley, Ottawa and Lisbon temperature time

series. Our hypothesis is that the temperature time series obtained by the weather stations have not changed very much during the time period considered. In general, offline methods, such as the MLP, cannot deal with nonstationary functions, do not support one-pass training, and require higher CPU time and memory when compared with online methods.

One-step singular and granular predictions of FBeM for the Death Valley, Ottawa, and Lisbon time series are shown in Figs. 2, 3 and 4. In these figures, the bottom plots enlarge the temperature predictions for the time intervals [1,040, 1,095], [1,108, 1,178] and [916, 992], as marked by the zoom signs. The middle plots show the evolution of the number of rules and *RMSE* index.

It is worth noting that while prediction  $p$  attempts to match the actual mean temperature value, the corresponding granular information  $[u, U]$  formed by the lower and upper bounds of consequent trapezoidal membership functions intends to envelop previous data and uncertainty from the actual, but unknown temperature function  $f$ . Therefore  $[u, U]$  is the range of values that bounds

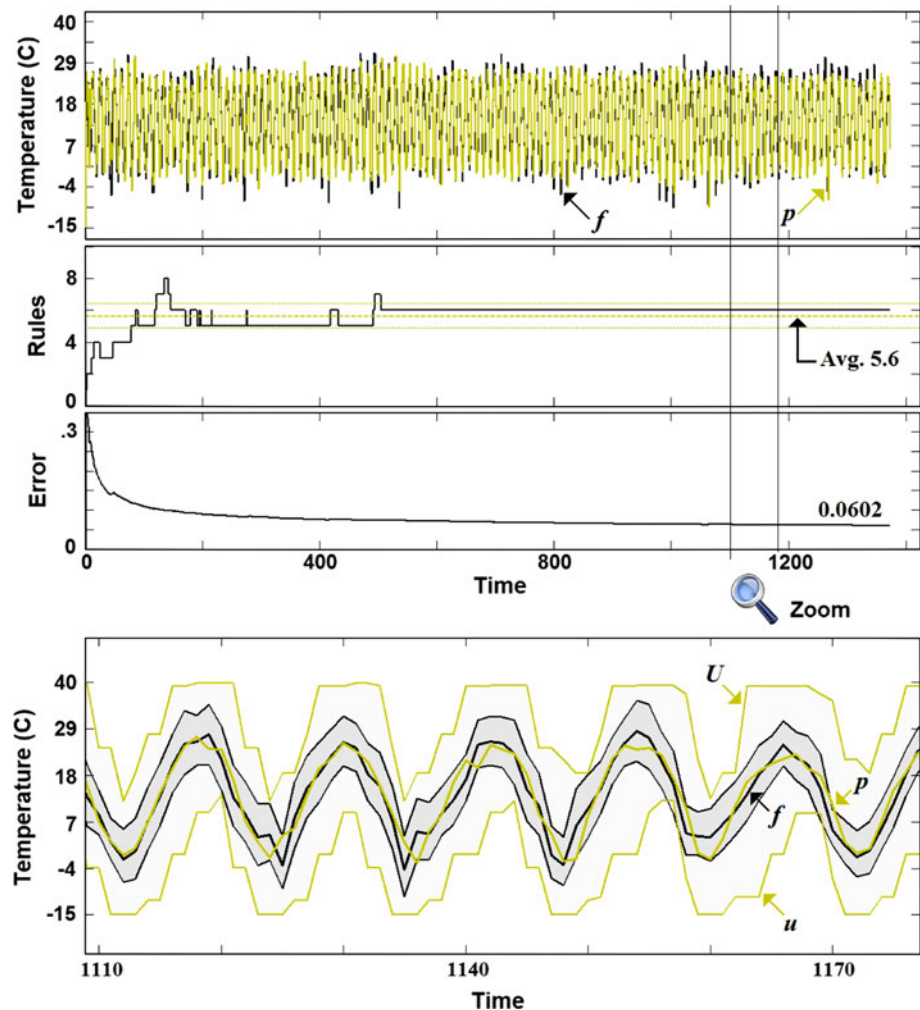
predictions based on past actual temperature values. Moreover, if required, each granular prediction may come with a label and a proper linguistic description. FBeM is an evolving approach to handle fuzzy granular data streams, and to simultaneously provide singular and granular predictions.

#### 4.3 Time complexity

In this section we examine how the performance of FBeM is affected by the number of input variables and fuzzy rules. Here performance concerns temporal scalability and *RMSE* to access processing time and prediction error, respectively.

For these purposes, we first performed several independent experiments varying the number of input variables (lagged observations of temperature values). Next, the FBeM parameters were chosen to give a rule base with about ten fuzzy rules. This means that the size of the rule base should not interfere in temporal scalability analysis of FBeM. We evaluate the processing time and prediction

**Fig. 3** FBeM prediction of Ottawa temperatures



error when the number of input variables increases. Evaluation was performed in the context of temperature prediction. Figure 5 shows the processing time and *RSME* considering different numbers of input variables from the Death Valley, Ottawa and Lisbon time series.

The bottom plot of Fig. 5 suggests that FBeM time complexity is quasi-linear with the number of inputs. This is important once many computational intelligence and statistical algorithms behave polynomially or exponentially which prohibits their use in handling massive data streams and large-scale online modeling. FBeM runs linearly with respect to the number of samplings once its learning algorithm is one-pass, of incremental nature.

It is worth noting at the top of Fig. 5 that weather time series requires a small number of input variables, while the remainders tend to confuse FBeM. The *RMSE* index for Death Valley, Ottawa, and Lisbon suggest a local optima in the range between six to twelve input variables.

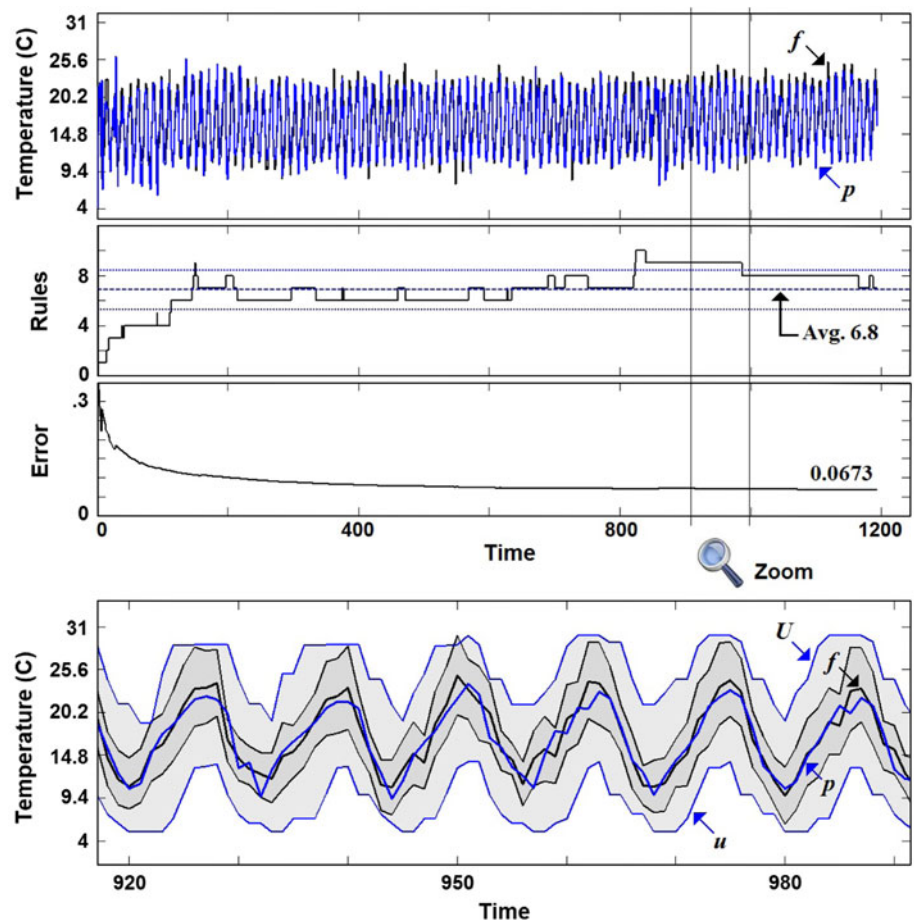
In the next experiment, we fix the number of input variables to five (5) and run the FBeM algorithm with parameters that force it to generate an increasing number of rules. The aim here is to evaluate temporal scalability and *RMSE* when the size of the rule base increases. Figure 6

shows the results obtained for the Death Valley, Ottawa, and Lisbon time series data.

The bottom plot of Fig. 6 shows that the processing time of FBeM grows exponentially with the number of rules. Although the algorithm deals linearly with the number of samples and input variables, granularity constraints within the FBeM framework is of utmost importance to keep the system operating online. Effective procedures to bound the rule base and protect FBeM from outbursts of growth are: (1) using the half-life value  $1/\psi$  as in Sect. 3.10. The set of FBeM rules,  $c$ , is guaranteed to be less than or equal to  $1/\psi$  anytime. For example, suppose  $1/\psi = 6$  and that the rule base contains seven rules. The last six samples can only activate six or less of the existing rules. Thus, at least one of the rules should be inactive for seven time steps, which contradicts that  $1/\psi = 6$ ; (2) adapting the maximum width allowed for granules,  $\rho$ , as in Sect. 3.4. This procedure develops only the necessary quantity of granules and rules [see (7), (8)]. Notice that the points at the right of the plots of Fig. 6 are obtained setting  $1/\psi$  to a large value, e.g. 10,000, and turning the granularity adaptation procedure off.

The error curves at the top plot of Fig. 6 show that quite small and large rule bases decrease model accuracy. We

**Fig. 4** FBeM prediction of Lisbon temperatures

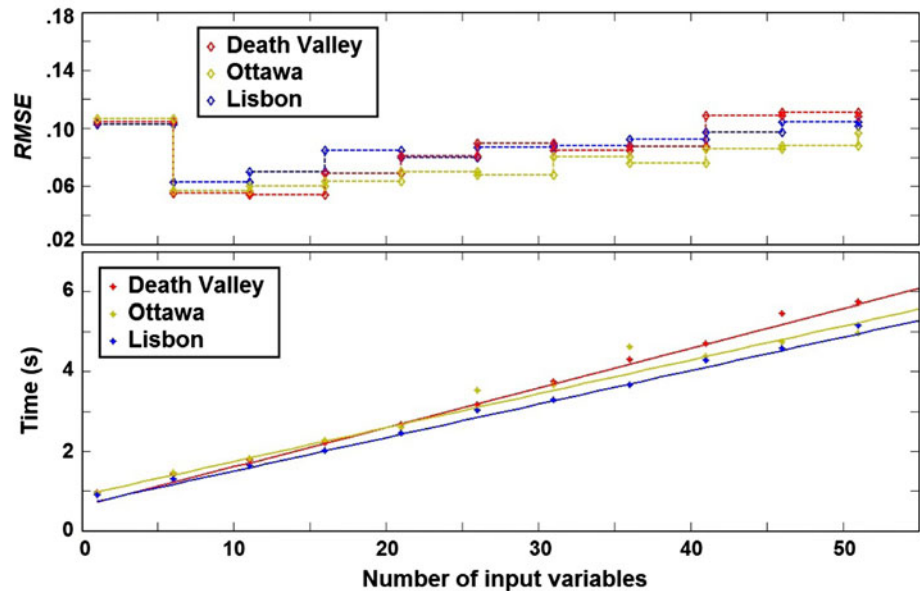


employ piecewise cubic Hermite interpolation polynomials to fit the error data. Curiously, error values suggest more appropriate models with about 6 to 12 rules. This reinforces the hypothesis that seasonal trends are better modeled by a single FBeM rule for each of them. Excessive granularity is detrimental because similar information is forcibly split into different granules and the underlying local models do not profit from the full information.

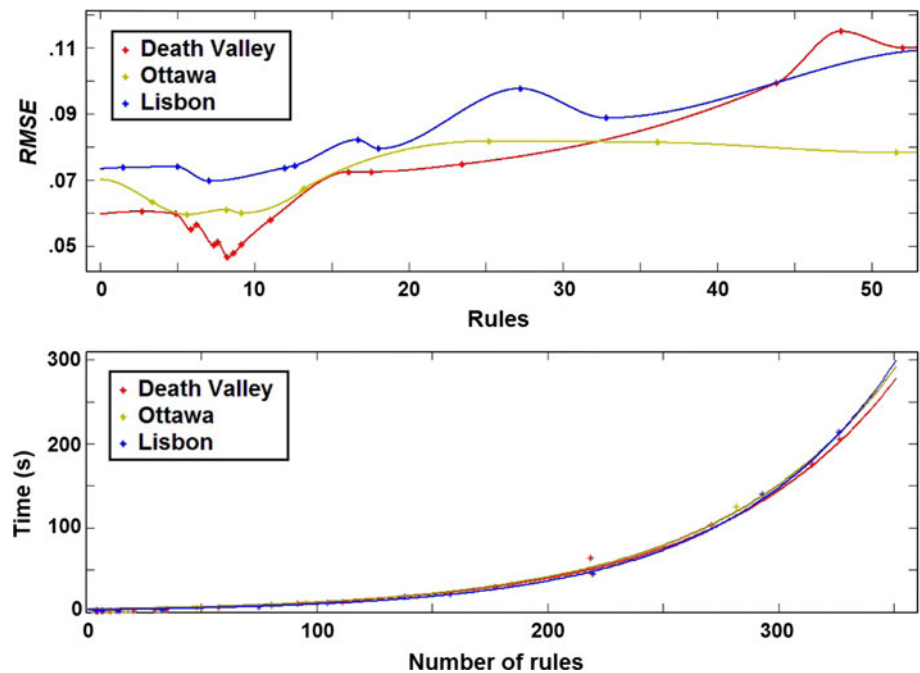
The average number of rules in FBeM depends on the choice of  $\rho$  and  $1/\psi$ . Reference Leite et al. (2011)

recommends  $\rho^{[0]} = 0.5$  to balance structural stability and plasticity whenever we lack detailed knowledge of the modeling task and data properties. Monthly mean temperature prediction experiments suggest  $\rho^{[0]}$  in the range from 0.6 to 0.8 to avoid rule overshoot after learning starts. This helps to attain smoother structural development along next time steps. Gradual adaptation of the granularity also alleviates initial guesses and guide the value of  $\rho$  according to the data stream. For monthly weather prediction, we suggest  $h_r$  values in the range between 12 and

**Fig. 5** FBeM processing time and *RMSE* for Death Valley, Ottawa, and Lisbon time series



**Fig. 6** FBeM processing time and *RMSE* for Death Valley, Ottawa, and Lisbon time series



48. The idea here is: if a trend does not appear again in the next year/four years, then remove its corresponding rule.

#### 4.4 Handling abrupt regime changes

Long term climate changes cause average monthly temperatures to gradually drift over time yet abrupt shifts are

hardly noticeable. The experiment addressed in this section show how FBeM reacts when abrupt changes occur in nonstationary time series.

For this purpose, we consider a hypothetical situation in which the time series of Death Valley, Ottawa, and Lisbon occur sequentially, forming a single time series. Two severe regime shifts are easily identified as the top plot of

**Fig. 7** FBeM prediction of the Death Valley, Ottawa, and Lisbon temperature time series combined

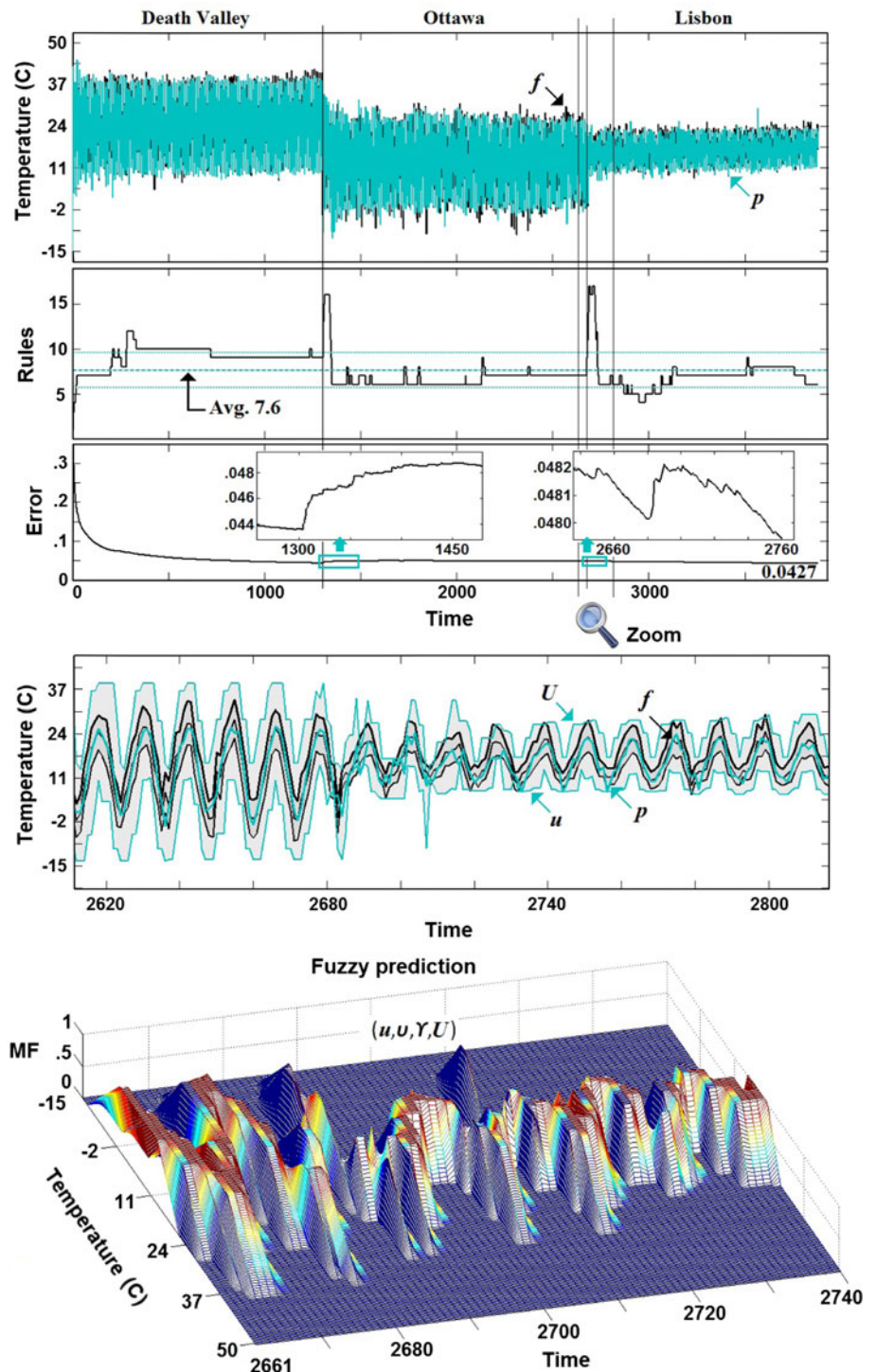


Fig. 7 illustrates. The bottom plot of Fig. 7 shows the fuzzy temperature predictions during the Ottawa–Lisbon shift (time interval between 2,661 and 2,740). In this experiment, FBeM should adapt the model to capture the new temperature profile and forget what is no longer relevant for the current environment. The initial parameters of FBeM were:  $\rho = 0.6$ ,  $h_r = 1/\psi = 48$  and  $\eta = 2$ . Figure 7 shows the *RMSE*, the number of rules and the granular and corresponding singular predictions. Notice in Fig. 7 that the number of rules of the rule base peaks after the Death Valley–Ottawa and Ottawa–Lisbon transitions and decreases afterwards. Similarly, the *RMSE* increases slightly and decreases in the next steps after time series transitions. Online adaptability improves prediction accuracy after the transitions. FBeM is stable to abrupt changes in fuzzy data streams, a challenge to a variety of machine learning algorithms.

## 5 Conclusion

This work has suggested FBeM, an evolving granular fuzzy modeling framework based on fuzzy granular data streams. FBeM carries a series of properties that makes it suitable to model online nonstationary functions using fuzzy data. FBeM gives accurate and granular information simultaneously. Granular model predictions contain a range of possible values which turns the predictions more reliable and truthful. We have addressed short-term weather temperature prediction as an application example. The data consist of triangular fuzzy numbers drawn from monthly minimum, mean and maximum average temperatures measured by meteorological stations. FBeM has been capable to handle fuzzy granular data and outperform alternative evolving methods in one step temperature prediction. Future research will explore fuzzy granular modeling of very large scale fuzzy system and optimization.

**Acknowledgments** D. Leite acknowledges CAPES, Brazilian Ministry of Education, for his fellowship. R. Ballini thanks FAPESP, the Research Foundation of the State of Sao Paulo, and CNPq, the Brazilian National Research Council, for Grants 2011/13851-3 and 302407/2008-1, respectively. P. Costa is grateful to the Energy Company of Minas Gerais-CEMIG, Brazil, for Grant P&D178. F. Gomide thanks CNPq for Grant 304596/2009-4. We are also grateful to the anonymous reviewers for their helpful comments and suggestions.

## References

- Aggarwal CC, Yu PS (2008) A framework for clustering uncertain data streams. *IEEE international conference on data engineering*, pp 150–159
- Angelov P, Filev D (2004) An approach to online identification of Takagi–Sugeno fuzzy models. *IEEE Trans Syst Man Cybern Part B* 34(1):484–498
- Angelov P, Zhou X (2006) Evolving fuzzy systems from data streams in real-time. *International symposium on evolving fuzzy systems*, pp 29–35
- Angelov P, Zhou X (2008) Evolving fuzzy-rule-based classifiers from data streams. *IEEE Trans Fuzzy Syst* 16(6):1462–1475
- Astrom KJ, Wittenmark B (1994) *Adaptive control*, 2nd edn. Addison-Wesley Longman Publishing Co., Inc., Boston
- Bargiela A, Pedrycz W (2002) *Granular computing: an introduction*, 1st edn. Kluwer Academic, Dordrecht
- Bargiela A, Pedrycz W (2005) Granular mappings. *IEEE Trans Syst Man Cybern Part A* 35(2):292–297
- Bargiela A, Pedrycz W (2008) Toward a theory of granular computing for human-centered information processing. *IEEE Trans Fuzzy Syst* 16(2):320–330
- Beringer J, Hullermeier E (2007) Efficient instance-based learning on data streams. *Intell Data Anal* 11(6):627–650
- Bouchachia A (2010) An evolving classification cascade with self-learning. *Evol Syst* 1(3):143–160
- Box GEP, Jenkins GM, Reinsel GC (2008) *Time series analysis: forecasting and control*, 4th edn. Wiley Series in Probability and Statistics, New York
- Dubois D, Prade H (2004) On the use of aggregation operations in information fusion processes. *Fuzzy Sets Syst* 142(1):143–161
- Gabrys B, Bargiela A (2000) General fuzzy min–max neural network for clustering and classification. *IEEE Trans Neural Netw* 11(3):769–783
- Haykin S (1999) *Neural networks: a comprehensive foundation*, 2nd edn. Prentice Hall, Englewood Cliffs
- Johnson CR (1988) *Lectures on adaptive parameter estimation*. Prentice-Hall, Inc., Upper Saddle River
- Kasabov N (2007) *Evolving connectionist systems: the knowledge engineering approach*, 2nd edn. Springer, Berlin
- 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
- Leite D, Gomide F (2012) Evolving linguistic fuzzy models from data streams. In: Trillas E, Bonissone P, Magdalena L, Kacprzyk J (eds) *Studies in fuzziness and soft computing: a homage to Abe Mamdani*. Springer, Berlin, pp 209–223
- Leite D, Costa P, Gomide F (2010a) Evolving granular neural network for semi-supervised data stream classification. *Int Joint Conf Neural Netw*, pp 1–8
- Leite D, Costa P, Gomide F (2010b) Granular approach for evolving system modeling. In: Hullermeier E, Kruse R, Hoffmann F (eds) *Lecture notes in artificial intelligence*, vol 6178. Springer, Berlin, pp 340–349
- Leite D, Costa P, Gomide F (2012a) Interval approach for evolving granular system modeling. In: Mouchaweh MS, Lughofer E (eds) *Learning in non-stationary environments: methods and applications*. Springer, Berlin
- Leite D, Costa P, Gomide F (2012b) Evolving granular neural networks from fuzzy data streams. *Neural Netwo* (Submitted)
- Leite D, Gomide F, Ballini R, Costa P (2011) Fuzzy granular evolving modeling for time series prediction. *IEEE international conference on fuzzy systems*, pp 2794–2801
- Lemos A, Caminhas W, Gomide F (2011) Fuzzy evolving linear regression trees. *Evol Syst* 2(1):1–14
- Lin TY (2002) *Neural networks, qualitative fuzzy logic and granular adaptive systems*. World Congress of Computational Intelligence, pp 566–571
- Lughofer E, Angelov P (2011) Handling drifts and shifts in on-line data streams with evolving fuzzy systems. *Appl Soft Comput* 11(2):2057–2068
- Lughofer E, Bouchot J-L, Shaker A (2011) On-line elimination of local redundancies in evolving fuzzy systems. *Evol Syst* 2(3):165–187

- Nandedkar AV, Biswas PK (2009) A granular reflex fuzzy min–max neural network for classification. *IEEE Trans Neural Netw* 20(7):1117–1134
- Pedrycz W (2007) Granular computing—the emerging paradigm. *J Uncertain Syst* 1:38–61
- Pedrycz W (2010) Evolvable fuzzy systems: some insights and challenges. *Evol Syst* 1(2):73–82
- Pedrycz W, Gomide F (2007) *Fuzzy systems engineering: toward human-centric computing*. Wiley-Hoboken, NJ
- Pedrycz W, Skowron A, Kreinovich V (eds) (2008) *Handbook of granular computing*. Wiley-Interscience, New York
- Roof S, Callagan C (2003) The climate of Death Valley, California. *Bull Am Meteorol Soc* 84:1725–1739
- Rubio JJ (2010) Stability analysis for an online evolving neuro-fuzzy recurrent network. In: Angelov P, Filev D, Kasabov N (eds) *Evolving intelligent systems: methodology and applications*, Wiley/IEEE Press, New York, pp 173–199
- Simpson PK (1992) Fuzzy min–max neural networks. Part I: classification. *IEEE Trans Neural Netw* 3(5):776–786
- Simpson PK (1993) Fuzzy min–max neural networks. Part II: clustering. *IEEE Trans Fuzzy Syst* 1(1):32–45
- Yager RR (2007) Learning from imprecise granular data using trapezoidal fuzzy set representations. In: Prade H, Subrahmanian VS (eds) *Lecture notes in computer science*. Springer, Berlin, vol 4772, pp 244–254
- Yager RR (2009) Participatory learning with granular observations. *IEEE Trans Fuzzy Syst* 17(1):1–13
- Yao JT (2007) A ten-year review of granular computing. *IEEE international conference on granular computing*, pp 734–739
- Yao YY (2008) Granular computing: past, present and future. *IEEE international conference on granular computing*, pp 80–85
- Yao YY (2010) Human-inspired granular computing. In: Yao JT (ed) *Novel developments in granular computing: applications for advanced human reasoning and soft computation*
- Young PC (1984) *Recursive estimation and time-series analysis: an introduction*. Springer, Berlin
- Zadeh LA (1979) Fuzzy sets and information granularity. In: Gupta MM, Ragade RK, Yager RR (eds) *Advances in fuzzy set theory and applications*, North Holland, Amsterdam, pp 3–18