Information Sciences 504 (2019) 95–112

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Ensemble of evolving optimal granular experts, OWA aggregation, and time series prediction

Daniel Leite[a,*], Igor Škrjanc[b]

[a] Department of Engineering, Federal University of Lavras, Brazil
[b] Faculty of Electrical Engineering, University of Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 19 April 2019
Revised 12 July 2019
Accepted 14 July 2019
Available online 15 July 2019

Keywords:
Evolving fuzzy systems
Ensemble learning
Aggregation functions
Granular computing
Weather time series prediction

ABSTRACT

This paper presents an online-learning ensemble framework for nonstationary time series prediction. Optimal granular fuzzy rule-based models with different objective functions and constraints are evolved from data streams. Evolving optimal granular systems (eOGS) consider multiobjective optimization, the specificity of information, model compactness, and variability and coverage of the data within the process of modeling data streams. Forecasts of individual base eOGS models are combined using averaging aggregation functions: ordered weighted averaging (OWA), weighted arithmetic mean, median, and linear non-inclusive centered OWA. Some aggregation functions use specific weights for the relevance of the base models and exclude extreme values and outliers. The weights of other aggregation functions are adapted over time based on a quadratic programming problem and the data within a sliding window. This paper investigates whether an online-learning ensemble can outperform individual eOGS models, and which aggregation function provides the most accurate forecasts. Real multivariate weather time series, particularly time series of daily mean temperature, air humidity, and wind speed from different weather stations, such as Paris–Orly, Frankfurt–Main, Reykjavik, and Oslo–Blindern, are used for evaluation. The results show that ensemble schemes outperform individual models. The proposed linear non-inclusive centered OWA function provided the most accurate numerical predictions.

1. Introduction

1.1. Contextualization

In recent years, nonstationary data stream processing and real-time model adaptation have become pivotal research topics. Data analysis has changed from offline batch processing of data and the use of deterministic learning and optimization methods that give multiple passes over datasets to the incremental and dynamic handling of online data streams [7,26,34].

Granular computing methods have been proposed to extract meaningful knowledge from large volumes of data. These methods examine the information flow in dynamic environments and produce and keep updated a granular model (classifier, regressor, predictor, controller) that can be linguistically understood [6,17,18,24,40]. The concepts of granules and granular mapping are used in the processes of learning and adaptation from data streams. Granules are objects or elements of a

* Corresponding author.
E-mail addresses: daniel.leite@deg.ufla.br (D. Leite), igor.skrjanc@fe.uni-lj.si (I. Škrjanc).

https://doi.org/10.1016/j.ins.2019.07.053
0020-0255/

domain drawn together by similarity or functionality [48]. A granular map is defined over granules. A set of granules in an input space are mapped into a set of granules expressed in an output space. Granular maps are found in rule-based systems [42,49].

The concept of granule is related to the concept of specificity [45]. An increase in the specificity of information tends to increase its usefulness to assist decisions and actions. However, when being very specific we risk being incorrect since experimental evidence may not be supported by the information. Moreover, we return to the original large dataset problem. In contrast, being very little specific, we can assure that the true values are included. However, we may end up with an unhelpful model. In information theory, this is called the specificity-correctness tradeoff [45]. The granularity [27] of granular models should reflect the available data as much as possible in the sense that granules should cover the data space properly [21], while being specific for a more meaningful and purposeful description.

Granular models built from data streams can be supported by many computational frameworks, such as interval analysis, statistics, fuzzy sets, rough sets, cluster analysis, neighborhood systems. Generalized constraints [49] are used to delimit and represent granules within the different frameworks. Computing with granules allows choices of representative objects and data handling tools. Regardless of the framework, online granulation aims to retain the essence of stream data as granular objects [29]. While direct application of machine learning methods to data streams is very often not feasible since it is difficult to maintain all the data in memory, computing with adaptive granules in an online environment aims to gradually develop more abstract, human-centered representations of time-varying data.

Evolving systems is a mainstream area of research in online data modeling that has been shown to be particularly effective in addressing life-long learning in non-stationary environments [2,12,23,33,37]. *Evolving* systems should be differentiated from *evolutionary* and *adaptive* systems. Evolutionary refers to genetic algorithms and genetic programming and deals with populations of individuals using recombination, mutation, and selection operators during multiple generations, typically in a static way [16]. Adaptive systems in control and systems theory stands for models with adaptive parameters only. Evolving systems refer to models with a higher level of flexibility and autonomy as not only their parameters but also their structures can be changed over time according to novelties, such as new behaviors, anomalies, drifts and switches [35]. With real-time parametric and structural adaptation, the issue of redesigning models from scratch is avoided in evolving methods. A variety of heuristics have been proposed to guide the development and incremental adaptation of rule-based models from numerical [2,23] and granular [18,19] data streams. Persuasive practical solutions to immediate goals have been achieved. However, assurances that certain conditions will be optimally fulfilled remain scarce in the area.

An optimal framework to evolve rule-based granular models from numerical data streams, called evolving optimal granular system (eOGS), was proposed in [15]. eOGS uses piecewise affine and inclusion functions connected to Gaussian and hyper-rectangular forms of granules to produce granular and numerical estimates of time-varying functions. By functions, we mean time-series, decision boundaries between classes, and control and regression functions in general. eOGS adapts its granular structure and parameters incrementally if new behaviors emerge or a change occurs in the data stream, while minimizing the multiobjective function. The eOGS framework [15] is briefly recalled and utilized for online ensemble learning in this paper, with focus on time series prediction.

In online ensemble learning for prediction, new data are processed on a per sample basis. The base evolving prediction models and ensemble aggregation weights are updated dynamically according to new instances and the accumulated estimation error within a sliding window to track nonstationarities. Since the estimation errors are averaged, an ensemble may outperform and be more robust than individual predictors [30]. Aggregation functions play the key role of performing information fusion within ensembles. Ordered weighted averaging (OWA) functions is a parameterized class of means [47]. By choosing specific weights for an OWA function, we are able to obtain various aggregation functions. This flexibility makes the OWA operator quite suitable for data-driven learning. Specifically, we investigate the conventional OWA, median, weighted arithmetic mean, and a linear non-inclusive centered OWA. These give preference to arguments lying in the middle.

1.2. Objective, research issues, and contributions

In this paper, an ensemble of eOGS prediction models for nonstationary time series prediction is constructed and evaluated. Ensemble averaging consists of combining the outputs of multiple base models to produce a potentially more accurate and robust estimation. The same data stream (i.e., past values of time series) is fed into a set of eOGS models. The models are essentially different from each other due to a different set of meta-parameters (constraints on the eOGS objectives) and, therefore, a different number of rules, granules, and different local parameters. Individual estimations are combined using an aggregation function. We evaluate the usual OWA function as well as particular functions obtained from specific OWA weights, e.g., weighted arithmetic mean, median, and a central OWA function.

We seek to answer the following questions: (i) Does an ensemble of evolving prediction models (in particular of the eOGS type) perform better than an individual evolving prediction model in a nonstationary environment? (ii) Is there a better aggregation function to be used for information fusion of multiple models within ensembles? We consider weather time series, in particular time series of daily mean temperature, air humidity and mean wind speed from different stations, viz. Paris–Orly, Frankfurt–Main, Reykjavik and Oslo–Blindern for empirical evaluation.

The main contributions are: (i) While existing online ensembles are very often devoted to classification tasks, this paper proposes an ensemble of prediction models. Moreover, the base prediction models are evolved (parametrically and structurally) autonomously in real-time according to a multiobjective function in a formal and systematic fashion;

(ii) A structured analysis of averaging aggregation functions, such as the median and a linear non-inclusive centered OWA function, is provided for the exclusion of the extremes and outliers. In addition, the conventional OWA function and the weighted arithmetic mean, whose weights are adapted from a quadratic programming problem over data within a sliding window, are investigated; and (iii) A thorough analysis of experimental results using real multivariate weather time series is given.

1.3. Brief review of the literature

Ensemble-based classifiers are mostly constructed using batch classifiers and the concepts of bagging or boosting, or from the accumulation of samples in a sliding window for multiple retraining steps [50]. Online data streams are time varying and generated continuously, often at a fast sampling rate, which makes multiple training steps considering multiple base models infeasible. Few studies about evolving base models within ensemble structures can be found in the literature. However, such evolving base models and the ensemble itself are classifiers. This study deals with regressor learning, aggregation functions, and optimal evolving granular fuzzy prediction models.

An ensemble of evolving classifiers based on the concept of stacking is described in [9]. The base classifiers are self-developing fuzzy-rule-based models of the eClass family [2]. Experimental results on two benchmark datasets suggest that eStacking improves, in general, the performance of individual eClass models as long as they are sufficiently diverse. The constraint of single-scanning through the data, inherent to online data-stream environments, imposed significant challenges to learning and classification; however, the ensemble outperformed the individual evolving classifier.

A parallel implementation of a typicality and eccentricity data analytics classification framework, known as TEDAClass, was proposed in [11]. The idea is to divide the samples of a data stream into chunks and distribute the chunks to multiple computing nodes. A parallel processing scheme using five processors can be viewed as an ensemble of classifiers. The ensemble was shown to spend 12% of the computing time compared to the time spent by a single processor at the price of a 1.1% worse classification accuracy due to the parallelization. The parallel approach is convenient for dealing with thousands or millions of data in high performance big-data processing.

A fast deep learning network for handwriting recognition is proposed in [1]. The approach provides interpretable models and is entirely data-driven. The network comprises an ensemble of zero-order evolving fuzzy rule-based models, which are built in parallel through an Autonomous Learning Multiple Model (ALMMo) method. The decision on the class label is made by a committee on the basis of the fuzzy mixture of the trained zero-order fuzzy models and an overall confidence score for each class.

Parsimonious ensemble (pENsemble) is proposed in [31] as an evolving variation of the dynamic weighted majority ensemble method by [14]. pENsemble employs base evolving classifiers from data streams in its structure and is equipped with a pruning mechanism. Base classifiers are pruned depending on localized generalization errors. Feature selection is also performed during online learning. It has been shown that the pENsemble can handle local drift in data streams to maintain a level of classification accuracy.

Ensembles of intelligent prediction models are addressed in [25,32]. In this case, the neural and neuro-fuzzy base predictors are neither evolving nor adaptive. Often only the weights that give different levels of importance to the base models are adaptive over time [8].

An ensemble of adaptive neuro-fuzzy inference systems (ANFIS) for chaotic time series prediction is described in [25]. Simple and weighted averages were used to aggregate the predictions of the base ANFIS models. The diversity of the base models is given by different types of membership functions and error goals. The performance of the ensemble architecture overcomes that of several standard statistical methods and individual neural network models. Time series such as Mackey–Glass, Dow Jones, and the Mexican stock exchange were considered in the evaluation. It should be noted that although ANFIS is an acronym that suggests online parameter adaptation, ANFIS models are static.

An adaptive ensemble based on Extreme Learning Machines (ELMs) for one-step prediction is presented in [8]. The ensemble consists of a number of randomly initialized ELMs, from 60 to 70 models. A base ELM may have different numbers of features, hidden neurons, biases, and input layer weights. The nonstationary time series *Quebec Births* was used to show that the adaptive ensemble outperforms a static Least Squares Support Vector Machine approach. Although the weight of each base ELM is adaptive according to the square error of the predictions, the base ELMs themselves are pre-trained and static.

All-pairs evolving fuzzy models to handle online multiclass classification problems are proposed in [22]. Binary classifiers are developed for each pair of classes, which produces less complex decision boundaries, and faster data processing and model updating. Type 1 Takagi–Sugeno fuzzy models, i.e., regressors, were also considered for each pair. The approach can be viewed as an ensemble strategy that employs weighted voting based on a preference relation matrix to decide about the class of a sample. The concepts of the reliability of classifiers, the degree of ignorance, and conflicts are discussed. Empirical evaluation based on high-dimensional multi-class problems show that the all-pair evolving approach can boost classifier accuracy.

Two studies on ensembles of evolving predictors can be found in the literature [4,36]. In [4], an ensemble of Fuzzy-set-Based evolving Models (FBeM) [19] with different structures and parameters for multivariate weather time series prediction is outlined. Each member of the ensemble is able to model the weather dynamics from data streams of wet bulb temperature, atmospheric pressure, maximum temperature, and relative humidity of the air. Rainfall levels were

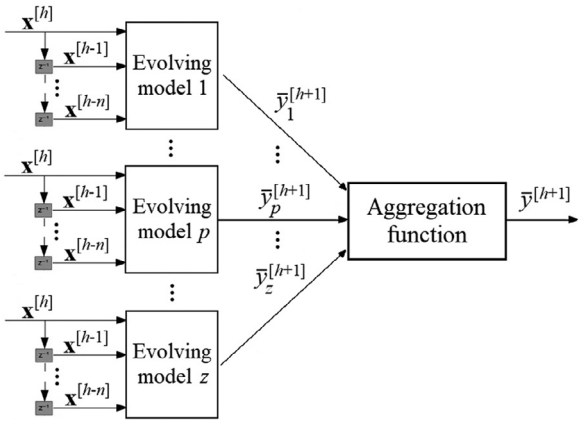

Fig. 1. Ensemble of evolving models.

predicted five days in advance. Empirical results show that the FBeM ensemble outperformed other ensembles, such as ensembles of ANFIS, evolving Takagi–Sugeno (eTS+) and Dynamic Evolving Neuro–Fuzzy Inference System (DENFIS) models, in terms of accurate numerical predictions. Arithmetic mean wasbasically used to combine the contribution of base FBeM predictors.

In [36] a variation of data cloud-based intelligent method known as TEDA [11] is given. In addition, an ensemble of cloud and evolving fuzzy models were combined through the weighted arithmetic mean to give numerical predictions of mean monthly temperature. Past temperature values, as well as previous values of exogenous variables, such as cloudiness, rainfall, and humidity, are considered. A non-parametric Spearman correlation-based method is proposed to rank and select the most relevant features and time delays for a more accurate prediction. A statistical hypothesis test showed that the average performance of TEDA and the ensemble of models is essentially the same for a p-value of 0.05, and that TEDA and the ensemble of models are statistically superior to the eTS and evolving extended Takagi–Sugeno (xTS) approaches individually.

2. Ensemble of predictors and aggregation functions

2.1. Evolving ensemble

In standard ensembles, all base models attempt to predict the same function. Each model is constructed independently and then outputs are aggregated without preference to a model. Online ensembles have become attractive because they display robustness to new data and patterns [30].

When a concept (a pattern of the time series) is abruptly replaced with a new concept, or when a drift (a gradual change of the mean and/or variance of the data) occurs, any model tends to have its prediction accuracy instantaneously or permanently reduced. Evolving models, however, track such changes because they utilize online incremental algorithms to adapt their parameters as well as their structures. After a relatively quick transient adaptation period, an evolving model usually returns to a stable accuracy rate.

The steps to generate an evolving ensemble are: generate and evolve separately z eOGS experts (each operating with different constraints and parameters) and, concomitantly, combine their estimations to provide the ensemble estimation. Different constraints for the objectives of the base eOGS models introduce diversity to the ensemble. Fig. 1 shows the proposed evolving ensemble scheme. The eOGS framework, its basic concepts and incremental learning algorithm will be briefly reviewed in Section 3.

Different types of aggregation functions can be used to perform information fusion within the evolving ensemble. In general, there are no specific guidelines to choose a particular aggregation function. Weighted averaging aggregation is a broad class of aggregation functions that are particularly more robust to errors in the data and to outliers, such as individual estimation values provided during transient adaptation periods as a response to concept change. Different weights, as components of the aggregation function, are assigned to the base models. The weights reflect the current relevance and contribution of each base model to the ensemble estimation.

In the following, weighted averaging functions to be used in the ensemble scheme are investigated. The classical OWA, the weighted arithmetic mean, the median, and the linear non-inclusive centered OWA are studied. While the two first functions have their weights updated according to a quadratic programming problem and previous estimation errors, the two latter functions have a fixed and specific set of weights, but the special property of excluding extreme values.

2.2. Aggregation functions

Aggregation functions $\mathbb{C} : [0, 1]^n \to [0, 1]$, $n > 1$ combine real values in the unit hypercube $[0, 1]^n$ into a single real value in [0, 1]. They must satisfy two fundamental properties: (*i*) monotonicity, i.e., given $\mathbf{x}^1 = (x_1^1, \ldots, x_n^1)$ and $\mathbf{x}^2 = (x_1^2, \ldots, x_n^2)$, if $x_j^1 \leq x_j^2 \ \forall j$ then $\mathbb{C}(\mathbf{x}^1) \leq \mathbb{C}(\mathbf{x}^2)$; (*ii*) boundary conditions: $\mathbb{C}(0, 0, \ldots, 0) = 0$ and $\mathbb{C}(1, 1, \ldots, 1) = 1$ [3,47].

2.2.1. T-norm and S-norm

T-norms (T) are commutative, associative, and monotone operators whose boundary conditions are $T(\alpha, \alpha, \ldots, 0) = 0$ and $T(\alpha, 1, \ldots, 1) = \alpha$, $\alpha \in [0, 1]$. The neutral element of T-norms is $e = 1$. An example is the minimum operator, $T_{\min}(\mathbf{x}) = \min_{j=1,\ldots,n} x_j$, which is the strongest T-norm because $T(\mathbf{x}) \leq T_{\min}(\mathbf{x})$ for any $\mathbf{x} \in [0, 1]^n$. The minimum is also idempotent, symmetric, and Lipschitz-continuous. Further examples of T-norms include the product and the Lukasiewicz T-norm.

S-norms (S) are commutative, associative, and monotone operators. $S(\alpha, \alpha, \ldots, 1) = 1$ and $S(\alpha, 0, \ldots, 0) = \alpha$ are the boundary conditions of S-norms. It follows that $e = 0$ is the neutral element of S-norms, which are stronger than T-norms. The maximum operator, $S_{\max}(\mathbf{x}) = \max_{j=1,\ldots,n} x_j$, is the weakest S-norm, that is, $S(\mathbf{x}) \geq S_{\max}(\mathbf{x}) \geq T(\mathbf{x})$, for any $\mathbf{x} \in [0, 1]^n$. Other examples include the probabilistic sum and the Lukasiewicz S-norm.

2.2.2. Averaging aggregation

An aggregation operator $\mathbb{C}$ is averaging if for every $\mathbf{x} \in [0, 1]^n$ it is bounded by $T_{\min}(\mathbf{x}) \leq \mathbb{C}(\mathbf{x}) \leq S_{\max}(\mathbf{x})$. The basic rule is that the output value cannot be lower or higher than any input value. An example of an averaging operator is the arithmetic mean,

$$M(\mathbf{x}) = \frac{1}{n} \sum_{j=1}^{n} x_j. \tag{1}$$

Averaging operators are idempotent, strictly increasing, symmetric, homogeneous, and Lipschitz continuous.

Definition 1 (Weight Vector)**.** A vector $\mathbf{w} = (w_1, \ldots, w_j, \ldots, w_n)$ is called a weighting vector if $w_j \in [0, 1] \, \forall j$ and $\sum_{j=1}^{n} w_j = 1$.

Definition 2 (Weighted Arithmetic Mean)**.** Given weights $\mathbf{w}$, the weighted arithmetic mean is the function

$$M_w(\mathbf{x}) = \frac{1}{n} \sum_{j=1}^{n} w_j x_j = \ \frac{1}{n} < \mathbf{w}, \mathbf{x} > . \tag{2}$$

M_w is strictly increasing for $w_j > 0$, and Lipschitz continuous. It is an asymmetric idempotent function, unless $w_j = 1/n \ \forall j$; additive, i.e., $M_w(\mathbf{x}_1 + \mathbf{x}_2) = M_w(\mathbf{x}_1) + M_w(\mathbf{x}_2) \ \ \forall \mathbf{x}_1, \mathbf{x}_2 \in [0, 1]^n$ and $\mathbf{x}_1 + \mathbf{x}_2 \in [0, 1]^n$; and homogeneous, i.e., $M_w(\lambda \mathbf{x}) = \lambda M_w(\mathbf{x}) \ \forall \mathbf{x} \in [0, 1]^n$ and $\lambda \in \Re$. It is a special case of Choquet integrals and Kernel function [3]. Determining the weights of M_w is an issue. Section 2.3 describes a quadratic programming problem in online data-stream context to obtain the weights $\mathbf{w}$ of M_w in an optimal way.

2.2.3. Ordered weighted averaging

Ordered weighted averaging (OWA) functions provide a parameterized class of mean [47]. These functions give weights not to a particular input but rather to its current value [47]. Let $\mathbf{x}\downarrow$ be obtained by sorting the elements of $\mathbf{x}$ in a nonincreasing order, $x_1 \geq \ldots \geq x_j \geq \ldots \geq x_n$.

Definition 3 (OWA)**.** Given a weighting vector $\mathbf{w}$, the OWA function is

$$\mathrm{OWA}_{\mathbf{w}}(\mathbf{x}) = \sum_{j=1}^{n} w_j x_j = < \mathbf{w}, \mathbf{x} \downarrow > . \tag{3}$$

If all weights are equal, OWA becomes the arithmetic mean. The weights $\mathbf{w} = (1, 0, ..., 0)$ and $\mathbf{w} = (0, ..., 0, 1)$ produces the maximum and minimum. For an odd or even n, $\mathbf{w} = (0, \ldots, 0, 1, 0, \ldots, 0)$ or $\mathbf{w} = (0, \ldots, 0, 1/2, 1/2, 0, \ldots, 0)$, respectively, result in $\mathrm{OWA}_{\mathbf{w}}(\mathbf{x}) = Med(\mathbf{x})$, i.e., the median. OWA is strictly increasing, idempotent, continuous, symmetric, homogeneous and shift-invariant. It is a special case of Choquet integrals. If we take $\mathbf{w}$ with some non-zero values in the middle, for example $\mathbf{w} = (0, \ldots, 0, 1/6, 1/3, 1/3, 1/6, 0, \ldots, 0)$, we obtain the Central OWA function proposed in [46]. The central OWA takes into account central inputs. Nonetheless, we can identify the weights of $\mathrm{OWA}_{\mathbf{w}}$ using quadratic programming and estimation errors, see Section 2.3. In this case, it is unlikely that $\mathrm{OWA}_{\mathbf{w}}$ assumes the form of other specific operators.

Definition 4 (Centered OWA)**.** An OWA operator of dimension n is centered if its associated weighting vector $\mathbf{w}$ is: (i) symmetric, i.e., $w_j = w_{n-j+1}, \ j = 1, \ldots, n$; (ii) strongly decaying, i.e., if $j_1 < j_2 \leq (n+1)/2$, then $w_{j_1} < w_{j_2}$, and if $j_1 > j_2 \geq (n+1)/2$, then $w_{j_1} < w_{j_2}$; and (iii) inclusive: $w_j > 0 \, \forall j$.

In this case, we say $\mathbf{w}$ is a centered weighting vector. These types of aggregation functions have the property of giving the highest weight to the central scores and less weighting to extreme values. If we remove condition (iii), and let $w_j = 0$

in the extremes, then $\mathbf{w}$ is a non-inclusive centered aggregator. In this case, the strong decay requirement is changed when we hit $w_j = 0$, as zeros may repeat (soft decaying [46]).

The median is a prototypical example of centered OWA. It is an average that is more representative of a typical value than the mean is. It essentially discards very high and very low values (potential outliers).

Definition 5 (Median)**.** Given an ordered n-dimensional vector $\mathbf{x}\downarrow$. The median is the function

$$Med(\mathbf{x}) = \begin{cases} \frac{1}{2}(x_{n/2} + x_{(n/2)+1}), & \text{if } n \text{ is even} \\ x_{(n+1)/2}, & \text{otherwise.} \end{cases} \tag{4}$$

A specific weighting vector $\mathbf{w}$ can be used to express the median as an OWA function. For an odd n, let $w_{(n+1)/2} = 1$ and $w_j = 0\ \forall j, j \neq (n+1)/2$. For an even n, let $w_{n/2} = w_{(n/2)+1} = 1/2$ and $w_j = 0$ for all other j. Then $Med(\mathbf{x}) = \text{OWA}_{\mathbf{w}}(\mathbf{x})$.

Otherwise, a linear non-inclusive centered OWA function that discards only the extreme left and right elements of $\mathbf{x}\downarrow$ is given as follows.

Definition 6 (Linear Non-inclusive Centered OWA)**.** Linear Non-inclusive Centered OWA is a centered OWA function whose weights are

$$\mathbf{w} = \begin{cases} \text{If } j \leq \frac{n}{2} \text{ then } w_j = \frac{j-1}{\frac{n+1}{2}-1} \\ \text{If } j > \frac{n}{2} \text{ then } w_j = \frac{-2}{n-1}(j-1) + 2. \end{cases} \tag{5}$$

The resulting weights should be rescaled, $w_j = \frac{w_j}{\sum_{j=1}^{n} w_j}$, so that $\sum_{j=1}^{n} w_j = 1$.

2.3. Online learning of ensemble weights

While the median (Definition 5) and linear non-inclusive centered OWA (Definition 6) have specific weights $\mathbf{w}$, the weights of the weighted arithmetic mean (Definition 2) and OWA (Definition 3) should be chosen consistently according to the newest data $(x, y)^{[h-\nu]}$, ..., $(x, y)^{[h]}$; ν is the length of the time window. Identification of classical OWA weights was studied in [43,44]. The issue is similar to that of the identification of the weights of arithmetic means. We use square estimation errors and solve a standard quadratic programming problem recursively as follows.

We minimize the sum of squares of the differences between estimated, $\bar{y}_p$, and actual, y, values, within a time window as follows:

$$\begin{aligned} Q &= min \sum_{h_a=h-\nu}^{h} \left(\sum_{p=1}^{z} w_p \bar{y}_p^{[h_a]} - y^{[h_a]} \right)^2 \\ s.t. &\quad \sum_{p=1}^{z} w_p = 1, \quad w_p \geq 0\ \forall p, \end{aligned} \tag{6}$$

where $\bar{y}_p$ is the estimation of the pth base model (refer to Fig. 1), ν is the size of the sliding time window. Parameter ν means the lifetime of information within the short-term memory of eOGS models. This will be discussed in the next sections. Therefore, the choice of ν depends on the purpose of the model. Models consider the last ν samples only to keep evolution active. Eq. (6) is a quadratic programming problem with convex objective function, which can be solved by any standard deterministic method, such as gradient descent. Quadratic programming has been shown to be numerically efficient and stable with respect to rank deficiency (if the data are linearly dependent) [41].

Initially, $w_p^{[0]} = 1/z\ \forall p$. Then,

$$w_p^{[h+1]} = w_p^{[h]} - \eta \frac{\partial Q}{\partial w_p}, \quad p = 1, \ldots, z, \tag{7}$$

where $\eta = 0.05$ is constant by default. As evolving experts may outperform each other in different periods, convergence of the weights is not expected, i.e., weights should change dynamically in an online environment. A small constant η is sufficient to allow that such behavior happens depending on a successful sequence of accurate estimations given by an expert. Relation (7) penalizes more severely the weight associated with a given expert if its estimations have been worse than those of the remaining experts. After the adaptation of w_p, the weights are rescaled, $w_p^{[h+1]} = \frac{w_p^{[h+1]}}{\sum_{p=1}^{z} w_p^{[h+1]}}\ \forall p$, so that $\sum_{p=1}^{z} w_p^{[h+1]} = 1$.

3. Optimal evolving granular framework

3.1. Preliminaries

The realization of granules in datasets has been discussed [20,27,28,38]. The principle of justifiable granularity [10,42] is a broad concept to guide the formation of meaningful granules based on experimental evidence. The more data are included within the bounds of a granule, the better. Granules should enclose all, or at least most, of the data. However, granules should be as specific as possible to come with an appropriate semantic and be more supportive of decisions and actions. If many data are included in a granule, the granule may become too wide. In contrast, if the granule is too small, few data are covered. The requirements of experimental evidence and specificity are conflicting. eOGS relies on multi-objective optimization, α-level sets, and Pareto fronts to achieve a trade-off between these requirements.

Constructing eOGS models from data streams requires incremental learning to keep an updated summary of the data. We want to capture spatial and temporal information from numerical data $x^{[h]} = (x_1, \ldots, x_j, \ldots, x_n)^{[h]},\ h = 1, \ldots,$ using a set of granules $\gamma = \{\gamma^1, \ldots, \gamma^c\}$. A local γ^i is chosen to fit a sample $x^{[h]}$, that is, to include the information conveyed by $x^{[h]}$. Otherwise, if $x^{[h]}$ is not sufficiently related to any γ^i, then a new granule γ^{c+1} can be created – expanding the current collection γ. Therefore, γ is a granular model that describes an unbounded data stream $x^{[h]}$, $h = 1, \ldots$.

Granules of an interval nature essentially depend on the lower l_j^i and upper L_j^i endpoints of axis-aligned hyper-rectangles. Endpoints are determined by spanning α-level subsets of individual features. Assume a Gaussian membership function $\Pi_j^i = \mathcal{G}(\mu_j^i, \sigma_j^i)$, which conveys the representation of the jth feature of the ith granule. We are interested in the range of values that the data rest on most occasions. Parameter α cuts Π_j^i at each tail and gives an interval $[l_j^i, L_j^i]$, whose extension over the Cartesian product space assembles a granule. Interval granules cover the data, and fuzzy Gaussians keep the essence of the information. Depending on α and the tightness of Π_j^i, granules and rule-based models may achieve any specificity. For example, if a learning algorithm is used to recursively adapt μ_j^i and σ_j^i, then Π_j^i may enlarge, shrink, and drift to track nonstationarities in data streams. At the same time, we can use α to control the size of intervals $[l_j^i, L_j^i]$, i.e., to manage the specificity of granules.

3.2. Specificity measure

Specificity refers to the amount of information conveyed by a granule [45]. Specificity measures approach their maximum values as an object closes in a single element. Values of specificity range within the interval [0,1], where 0 means that all points of a domain are equally possible, and 1 implies singularity. Let Π be a subset of X. Yager [45] defines a general class of specificity over the continuous domain as

$$\mathrm{Sp}(\Pi) = \int_0^{\alpha_{max}} F(\lambda(\Pi_\alpha))\mathrm{d}\alpha, \tag{8}$$

where $\Pi_\alpha = \{x : \Pi(x) \geq \alpha\}$ is the α-level of Π; $\alpha_{\max}$ is the height of Π; λ is a monotonic measure; and F: $[0, 1] \rightarrow [0, 1]$ is a function such that $F(0) = 1$; $F(1) = 0$; and $0 \leq F(x_1) \leq F(x_2)$, for $x_1 > x_2$.

Let Π be a Gaussian membership function in X, with modal value μ and dispersion σ,

$$\Pi = e^{-(x-\mu)^2/2\sigma^2}. \tag{9}$$

Gaussians are normal (height equal to 1), and have infinite support. Level sets of Gaussian membership functions are intervals Π_α, whose centers and radii are of the form $\mu \pm \sqrt{-2\sigma^2 \ln(\alpha)}$.

Using the Yager definition of specificity (8), the α-level set, Π_α, the Lebesgue–Stieltjes measure (8) for λ in a totally bounded domain, $X = [c, d]$, and $F(z) = 1 - z$, similar to [15], we obtain the specificity of Π as

$$\mathrm{Sp}(\Pi) = 1 - \frac{2\sqrt{2}\sigma\sqrt{-ln(\alpha)}}{d - c}, \tag{10}$$

where $[c, d] = [0, 1]$ if the data $x \in [0, 1]$ – see the development steps in [15]. Let $\Pi^i = \{\Pi_1^i, \ldots, \Pi_j^i, \ldots, \Pi_n^i\}$ be a set of Gaussian functions such that Π_j^i has domain X_j, and let $X^n = X_1 \times \ldots \times X_j \times \ldots \times X_n$. Through α-cuts we may assemble an interval granule γ^i in X^n. We define the specificity of a n-dimensional granule γ^i as the average of the specificity measure of each of its n components, namely,

$$\mathrm{Spa}(\gamma^i) = 1 - \frac{2\sqrt{2}}{n}\sum_{j=1}^{n}\frac{\sigma_j^i\sqrt{-ln(\alpha)}}{d_j - c_j}. \tag{11}$$

3.3. Multiple objectives

Online learning methods to build granular rule-based models should ideally update key decision parameters to provide the requested outcomes in an optimal sense and satisfy constraints. A decision maker may either interactively set eOGS

meta-parameters, namely $\mathbb{P}$, to force a specific model structure and behavior, or use automatic procedures for the purpose of optimizing an objective function within the feasible region formed by constraints. Parameters $\mathbb{P}$ will be presented formally in the next section.

Possible criteria to guide the adaptation of eOGS meta-parameters $\mathbb{P}$ include the numerical estimation error (E_n), granular estimation error (E_g), specificity of the granular map (Spa(γ)), and total number of rules (c). Generally, we want to reduce the numerical and granular errors, increase the specificity/meaningfulness of the granules, and maintain a compact rule-base structure. Clear tradeoffs emerge, e.g., a smaller number of rules tends to produce less specific granular maps, and may generate better or worse estimations depending on the data stream.

A multiobjective function to be minimized is

$$\begin{aligned} F(\mathbb{P}) = min \;\; & [f_1(\mathbb{P}), f_2(\mathbb{P}), f_3(\mathbb{P}), f_4(\mathbb{P})] \\ s.t. \;\; & \mathbb{P} \in \Omega \end{aligned} \tag{12}$$

where Ω is the parameter space, and

$$\begin{aligned} E_n &\triangleq f_1(\mathbb{P}), \\ E_g &\triangleq f_2(\mathbb{P}), \\ -\text{Spa}(\gamma) = -\sum_{i=1}^{c} \text{Spa}(\gamma^i) &\triangleq f_3(\mathbb{P}), \\ c &\triangleq f_4(\mathbb{P}). \end{aligned}$$

The objectives of (12) have no analytic expression in terms of $\mathbb{P}$, and they compete so that the solution is not unique. Improvement in one objective may degrade the others but maintain Pareto optimality. Experts can express preferences for a solution along a tradeoff surface. An interactive approach to choose $\mathbb{P}$ and produce a set of solutions, which is suggested to the decision maker, is given in [15]. In the next section, we describe a fully-autonomous approach to choose $\mathbb{P}$ in a balanced way.

In the ϵ-constraint method [5], a priority objective is chosen to be optimized while the other objectives are converted into constraints by setting an upper bound to each of them. Problem (12) takes the form

$$\begin{aligned} F(\mathbb{P}) = min \;\; & f_s(\mathbb{P}) \\ s.t. \;\; & f_t(\mathbb{P}) \leq \epsilon_t, \;\; \forall t, \;\; t \neq s \\ & \mathbb{P} \in \Omega \end{aligned} \tag{13}$$

whose solution is Pareto optimal if the objective functions are convex [5,15]. Otherwise, as convexity is never guaranteed in nonstationary context, the method provides local non-inferior solutions [5,15].

A systematic and fully autonomous heuristic procedure to change the decision parameters $\mathbb{P}$ of eOGS models is given in Section 4.2. In this case, the heuristic procedure is the decision maker, and no human is involved in the process of finding a solution to (12) from (13). In practice, users can also systematically select values for $\mathbb{P}$ manually, and observe if the price that should be paid by the secondary objectives for an improvement of the priority objective, f_s, is acceptable. In this case, the user is the decision maker.

To summarize, once a priority objective, and the bounds for the remaining objectives are chosen, eOGS aims to generate local non-inferior solutions in terms of numerical and granular estimation errors, structural compactness, and granular specificity. eOGS is a heuristic procedure that continuously attempts to solve (13) at each time step. In other words, constraints are not violated during the autonomous operation of the algorithm since meta-parameters $\mathbb{P}$ are changed to guarantee their feasibility. Additionally, the primary objective is constantly driven to a minimum by means of a granulation mechanism, granule updating procedure, and Recursive Least Squares method, which are performed within the eOGS framework. Approaches for the eOGS heuristics may either involve user preferences in an interactive mode, or be fully autonomous. In both cases, the solutions developed during the processing steps are only approximations of the Pareto optimal solutions. This is because the systems we are concerned with are nonstationary, and eventually subject to functional changes. Details of the eOGS are given in the following sub-sections. Section 4 summarizes the interactive design, and the fully autonomous operation approaches of the eOGS.

3.4. eOGS Rule Base

An eOGS granule, γ^i, is defined by spanning α-level sets of membership functions $\Pi_j^i = \mathcal{G}(\mu_j^i, \sigma_j^i),\ j = 1, \ldots, n$, in the space $X_1 \times \ldots \times X_n$. Granule γ^i can be explicitly described by the rule

R^i: IF $(l_1^i \leq x_1 \leq L_1^i)$ AND ... AND $(l_n^i \leq x_n \leq L_n^i)$

THEN $(u_1^i \leq y_1 \leq U_1^i)$ AND $\bar{y}_1^i = p_1^i(x_1, \ldots, x_n)$ AND

...

$(u_m^i \leq y_m \leq U_m^i)$ AND $\bar{y}_m^i = p_m^i(x_1, \ldots, x_n)$,

where l_j^i and $L_j^i,\ j = 1, \ldots, n$; $i = 1, \ldots, c$, are the jth lower and upper bounds of the attribute x_j according to the ith rule;

u_k^i and U_k^i, $k = 1, \ldots, m$, are the kth lower and upper bounds of the output y_k; and p_k^i are affine functions,

$$\tilde{y}_k^i = p_k^i(x_1, \ldots, x_n) = a_{0k}^i + \sum_{j=1}^{n} a_{jk}^i x_j. \tag{14}$$

In general, each p_k^i can be of a different type and does not need to be linear. The hyper-rectangle γ^i conveys Gaussian membership functions $\Pi_j^i = \mathcal{G}(\mu_j^i, \sigma_j^i)$, $j = 1, \ldots, n$, as internal representation – additional information that summarizes past data belonging to γ^i. Modal values, μ_j^i, and dispersions, σ_j^i, are captured recursively from the data stream. Moreover, the Recursive Least Squares (RLS) algorithm [17] is used to determine the coefficients a_{jk}^i, $j = 0, \ldots, n$, $k = 1, \ldots, m$, of the functions p_k^i whenever the ith rule is active for an input sample $x = (x_1, \ldots, x_n)$.

3.5. Stiegler initialization procedure

Granule and rule are created either if $x^{[h]}$ is not in $[l_j^i, L_j^i]$ $\forall i$ and some j; or $y^{[h]}$ is not in $[u_k^i, U_k^i]$ $\forall i$ and some k. Notice that the bounds of the underlying intervals depend on the value of α that cuts Π_j^i and Π_k^i. If $\alpha \to 0^+$, then $[l_j^i, L_j^i]\forall j$ and $[u_k^i, U_k^i]\forall k$ cover the whole input and output spaces, i.e., they form unit n- and m-dimensional hyperboxes. Contrariwise, if $\alpha \to 1^-$, then γ^i degenerates into a point.

One approach to initialize the parameters of a new granule is to consider Stigler's standard Gaussian function [39]. The new granule γ^{c+1} has modal value

$$\mu_{j,k}^{c+1} = (x_j, y_k)^{[h]} \ \forall j, k, \tag{15}$$

and dispersion

$$(\sigma_{j,k}^{c+1})^2 = 1/2\pi \ \forall j, k. \tag{16}$$

A hyper-rectangle centralized in $(x, y)^{[h]}$ is obtained as the Cartesian product of unidimensional α-cuts. Granule bounds are given as

$$[l_j^{c+1}, L_j^{c+1}] = x_j^{[h]} \pm \sqrt{-2(\sigma_j^{c+1})^2 \ln(\alpha)} \tag{17}$$

and

$$[u_k^{c+1}, U_k^{c+1}] = y_k^{[h]} \pm \sqrt{-2(\sigma_k^{c+1})^2 \ln(\alpha)}, \tag{18}$$

$0 < \alpha < 1$. Polynomial coefficients are set as

$$a_{0k}^{c+1} = y_k^{[h]}, \ \forall k; \ \text{and} \ a_{jk}^{c+1} = 0, \ \forall k, \ j = 1, \ldots, n. \tag{19}$$

Initial granule specificity is obtained from (11).

The Stiegler initialization approach may require some steps for the new granule to shrink and be sized similarly to the other granules. New samples within the bounds of the new granule are used to adapt its modal value, dispersion, and consequent coefficients. In [15], a *Minimal* initialization procedure that is useful for univariate time-series prediction is also given.

3.6. Parameter adaptation

Real-world data streams change over time. Adapting eOGS rules consists in enlarging or contracting granules, and simultaneously changing the coefficients of local functions.

Consider data within a time window, $(x, y)^{[h-\nu]}$, $(x, y)^{[h-\nu+1]}$, $\ldots$, $(x, y)^{[h]}$; ν is the length of the window, h is the time step. If a new $x^{[h]}$ fits $[l_j^i, L_j^i]$ $\forall j$, and $y^{[h]}$ fits $[u_k^i, U_k^i]$ $\forall k$ and some i, then the parameters of the local Gaussians Π^i are updated recursively from

$$\mu_{j,k}^i(\text{new}) = \frac{\nu^i \mu_{j,k}^i(\text{old}) + (x_j, y_k)^{[h]}}{\nu^i + 1}, \forall j, k \tag{20}$$

and

$$\left(\sigma_{j,k}^i(\text{new})\right)^2 = \frac{\nu^i(\mu_{j,k}^i - (l_j, u_k)) + \beta(|\mu_{j,k}^i - (x_j, y_k)^{[h]}|)}{(\nu^i + 1)(\mu_{j,k}^i - (l_j, u_k))} \left(\sigma_{j,k}^i(\text{old})\right)^2, \forall j, k, \tag{21}$$

where ν^i is the number of samples belonging to γ^i out of the past ν samples; $\beta = 2$ is a default value. Larger or smaller values for the parameter β force the expansion or contraction of Gaussians over the iterations, and more or less specific granular maps.

The α-level set of the updated granule, γ^i, is found as

$$[(l_j^i, u_k^i), (L_j^i, U_k^i)] = \mu_{j,k}^i(\text{new}) \pm \sqrt{-2\ln(\alpha)\left(\sigma_{j,k}^i(\text{new})\right)^2} \tag{22}$$

$\forall j, k$. Moreover, the specificity of the underlying granule, $\text{Spa}(\gamma^i)$, is calculated as in (11).

Given a priority objective f_s (numerical or granular estimation error, specificity of the granular map, or total number of rules), the ϵ-constraint method (13) finds α and β to satisfy the constraints f_t, and minimize f_s. The values of α and β are used in (21) and (22) to update the dispersions Π_j^i and Π_k^i and determine the bounds $[l_j^i, L_j^i]$ and $[u_k^i, U_k^i]$. Users manifest their preferences by choosing the main objective and setting admissible values for the constraints $\epsilon_t \forall t$.

Notice that the adaptation procedures described are recursive, i.e., old values are replaced by new values. Therefore, a sample $(x, y)^{[h]}$ can be discarded after being processed. The RLS algorithm [17] adapts the coefficients $a_{jk}^i \ \forall j, k$, for samples that activate γ^i. Notice also that the size of the time window, ν, means the lifetime of information within the short-term memory of eOGS models. Models consider the last ν samples only to keep evolution active. Additionally, notice that only one granule, and its associated local functions, needs to be updated per learning step. If two or more granules are active for a sample, then the most active granule after the application of the minimum (Gödel) T-norm over fuzzy membership degrees is used to determine the winner granule, i.e., the granule to be updated.

3.7. Merging and deleting granules

Merging granules, say γ^{i_1} and γ^{i_2}, is helpful in reducing the number of rules and eliminating partially overlapping granules representing similar information. We take the 2-norm of the difference between midpoints of all pairs of granules, i.e.,

$$\arg \min_{i_1, i_2 = 1, \ldots, c;\ i_1 \neq i_2} \frac{||\mu^{i_1} - \mu^{i_2}||}{n}, \tag{23}$$

where n is the number of features, or dimensions of μ. Given the closest granules, γ^{i_1} and γ^{i_2}, if

$$\frac{||\mu^{i_1} - \mu^{i_2}||}{n} \leq \omega, \tag{24}$$

being ω a constant or time-varying threshold, then the granules are merged. Manual and automatic adaptation approaches for ω according to an objective and constraints will be described in the Methodology section.

The resulting granule, say γ^{c+1}, is formed by $n+m$ Gaussian membership functions (n inputs and m outputs) whose modal values depend on the amount of samples each of the merged granules used to represent. Formally,

$$\mu_{j,k}^{c+1} = \frac{\nu^{i_1}\mu_{j,k}^{i_1} + \nu^{i_2}\mu_{j,k}^{i_2}}{\nu^{i_1} + \nu^{i_2}}, \ \forall j, k. \tag{25}$$

The maximum dispersion approach takes

$$\sigma_{j,k}^{c+1} = \max(\sigma_{j,k}^{i_1}, \sigma_{j,k}^{i_2}), \ \forall j, k, \tag{26}$$

as the dispersions of γ^{c+1}. The idea is to preserve information of a variety of samples in the merging region. Given (25) and (26), granule bounds and specificity can be obtained from (22) and (11) by analogy. Coefficients are set as

$$a_{jk}^{c+1} = \frac{a_{jk}^{i_1} + a_{jk}^{i_2}}{2}, \ \forall j, k. \tag{27}$$

A number of methods for merging can be developed considering dispersions, specificity, α-cuts, slope of local functions, and time windows, which may provide different results. An extensive analysis of this issue is beyond the scope of this paper.

Inactive rules for a number of iterations can be deleted. This may mean that the underlying system changed and removing granules is practical to keep the rule base size as compact as possible. Remember ν^i, the amount of samples that activated γ^i out of the newest ν samples, $x^{[h-\nu]}, \ldots, x^{[h]}$. For inactive granules, $\nu^i = 0$, and the respective granule and rule can be removed.

3.8. Interval function for granular prediction

The image of γ^i through a real function p_k^i is

$$p_k^i([l_1^i, L_1^i], \ldots, [l_n^i, L_n^i]) = \{p_k^i(x_1, \ldots, x_n) : x_j \in [l_j^i, L_j^i], \ j = 1, \ldots, n\}.$$

Generally, the image of γ^i through p_k^i is not a hyperrectangle, and it may be difficult to obtain in closed form. In practice, p_k^i can be approximated by an inclusion function P_k^i, which is a hyperrectangle in the range of p_k^i, namely $[u_k^i, U_k^i]$. An interval function P_k^i is called an inclusion function if $p_k^i \subseteq P_k^i \forall i$. Inclusion functions are not unique; they depend on how we choose P.

An inclusion function P_k^i is optimal if it is the hull of p_k^i. In other words, the optimal inclusion function for p_k^i is the smallest hyperrectangle P_k^{i*} that contains p_k^i. P_k^{i*} is unique. Its specificity is the highest possible value that guarantees inclusion.

Let p_k^i be monotonically increasing in $[l_j^i, L_j^i]$, $j = 1, \ldots, n$. Then we can obtain p_k^i from

$$p_k^i(x) = [p_k^i(l_1^i, \ldots, l_n^i), p_k^i(L_1^i, \ldots, L_n^i)].$$

Consequently, for any $x \in \gamma^i$, $p_k^i(x) \subseteq [p_k(l^i), p_k(L^i)]$. For monotonic decreasing functions we have

$$p_k^i(x) = [p_k^i(L_1^i, \ldots, L_n^i), p_k^i(l_1^i, \ldots, l_n^i)].$$

We adopt

$$p_k^i(x \in \gamma^i) = a_{0k}^i + \sum_{j=1}^{n} a_{jk}^i [l_j^i, L_j^i], \tag{28}$$

where $(a_{0k}^i, \ldots, a_{nk}^i)$ are degenerated intervals. If the slope $a_{jk}^i < 0$, then the bounds should be inverted, i.e. $[L_j^i, l_j^i]$.

3.9. Incremental learning algorithm

The eOGS learning algorithm for time series prediction is summarized as follows:

BEGIN
Set parameters $\mathbb{P} = \{\alpha, \beta, \nu, \omega\}$; (default values can be chosen, see
... Section 4.2. They are: $\alpha = 0.1$, $\beta = 2$, $\nu = 500$ and $\omega = 0.01$);
Choose a priority objective f_s;
Choose admissible values for the remaining objectives $\epsilon_t \forall t$;
for $h = 1, \ldots$
Read input data $x^{[h]}$;
if $h = 1$
Create granule γ^1 and rule R^1 (Eqs. (15)–(19) and (11));
Provide singular $\bar{y}^{[h]}$ and granular $[u^1, U^1]^{[h]}$ predictions;
else
Provide singular $\bar{y}^{[h]}$ and granular $[u^i, U^i]^{[h]}$ predictions;
// The actual output $y^{[h]}$ becomes available;
if $x_j^{[h]} \notin [l_j^i, L_j^i]$ $\forall i$ and some j
Create granule γ^{c+1} and rule R^{c+1} (Eqs. (15)–(19) and (11));
else
Adapt the most active granule γ^i (Eqs. (20)–(22) and (11));
Use the RLS algorithm to update the coefficients a_{jk}^i;
end
end
Delete inactive granules and rules;
Merge granules and rules (Eqs. (23)–(27), (11), (22));
Adapt parameters $\mathbb{P} = \{\alpha, \beta, \nu, \omega\}$ to minimize f_s respecting $\epsilon_t \forall t$;
end
END

---

4. Methodology

We describe a numerical and a granular error measure to evaluate model accuracy. A fully-autonomous procedure to adapt eOGS meta-parameters is addressed. The characteristics of real multivariate weather datasets are summarized.

4.1. Error indices

The root mean square error of numerical predictions,

$$RMSE = \frac{1}{H} \sum_{h=1}^{H} \sqrt{\frac{1}{m} \sum_{k=1}^{m} \left(y_k^{[h]} - \bar{y}_k^{[h]} \right)^2}, \tag{29}$$

where H is the current time step, and m is the number of outputs, is a measure of accuracy to compare different models for a particular dataset.

An error measure for granular predictions that takes into account data inclusion and the narrowness of the enclosure, known as mean granular error [15], is

$$MGE = \frac{1}{kH} \sum_{h=1}^{H} \sum_{k=1}^{m} 1 - \mathfrak{z}_k^{[h]} \left(1 - \left(U_k^{[h]} - u_k^{[h]} \right) \right), \tag{30}$$

where

$$\mathfrak{z}_k^{[h]} = \begin{cases} 1 & \text{if } y_k^{[h]} \in [u_k, U_k]^{[h]} \\ 0 & \text{otherwise.} \end{cases} \tag{31}$$

If the actual output, $y_k^{[h]}$, is out of the prediction bounds, $[u_k, U_k]^{[h]}$, then $\mathfrak{z}_k^{[h]} = 0$, giving the maximum *MGE*. The *MGE* index is less than 1 only if the granular prediction encloses $y_k^{[h]}$. The narrower the bounds $[u_k, U_k]^{[h]}$ that encloses $y_k^{[h]}$ are, the smaller the *MGE*.

4.2. Fully autonomous operation

A typical form of the optimization problem (13) takes the numerical estimation error as primary objective, i.e.,

$$\begin{aligned} F(\mathbb{P}) = min \quad & RMSE \\ s.t. \quad MGE &\leq \epsilon_1 \\ c &\leq \epsilon_2 \\ \text{Spa}(\gamma) &\geq \epsilon_3 \end{aligned} \tag{32}$$

The *RMSE* can be interchanged with any constraint. If a constraint is violated, the algorithm should respond as soon as possible by means of the parameters $\mathbb{P} = \{\alpha, \beta, \nu, \omega\}$. The default values for $\mathbb{P}$ are $\alpha = 0.1$, $\beta = 2$, $\nu = 500$ and $\omega = 0.01$, see [15]. If a prior set of samples is available, a data stream can be simulated in a matter of seconds and, perhaps, more appropriate parameters can be found. Convenient boundary values are $0.01 \leq \alpha \leq 1$; $1.5 \leq \beta \leq 2.5$; $\nu \geq n$, i.e., ν should be at greater or equal than the number of features; and $0.01 \leq \omega \leq 0.05$ [15]. As this paper focuses on numerical prediction accuracy, (32) is an appropriate formulation.

During the online operation of the eOGS algorithm, if a constraint is violated, parameters are changed as follows:

- If $MGE > \epsilon_1$, then α is stepped up by 0.01, and σ^2 is stepped down by 0.01 for new granules, but kept within a minimal and the Stiegler range of values, i.e. $0.01 \leq \sigma^2 \leq 1/2\pi$.
- If $c > \epsilon_2$, then α is stepped down by 0.01, and ω is stepped up by 0.001. After some iterations, when c assumes a feasible value, ω is reset to default.
- If $\text{Spa}(\gamma) < \epsilon_3$, then α is stepped up by 0.01, and β is stepped down by 0.001. After some iterations, when $\text{Spa}(\gamma)$ assumes a feasible value, β is reset to default.

Additionally,

- If $RMSE > \epsilon_4$, then α is stepped down by 0.01, and σ^2 is stepped up by 0.01 for new granules, but kept within a minimal and the Stiegler range of values, i.e. $0.01 \leq \sigma^2 \leq 1/2\pi$.

If the values ϵ_t are set in an unrealistic way, parameters $\mathbb{P}$ reach a limit and the best possible solution is presented.

While the eOGS algorithm (Section 3.9) attempts to minimize the *RMSE* and *MGE* indices, and the number of rules, c; and to maximize the specificity, $\text{Spa}(\gamma)$, the procedure described in this section monitors the constraints. If a constraint is violated, then parameters $\mathbb{P}$ are updated to reestablish the feasibility of the optimization problem.

4.3. Weather time series

Weather prediction, in particular daily mean temperature prediction, is useful to plan activities, protect property, and assist decision making in several economic sectors, such as energy, transportation, aviation, agriculture, inventory planning. Any system that is sensitive to the state of the atmosphere may benefit from such predictions.

Computational experiments assume data from different weather stations, namely,

- PARIS-ORLY, FR, Lat: +48:43:00, Lon: +002:22:59, elevation: 90 m;
- FRANKFURT-MAIN (FELDBERGSTR.), DE, Lat: +50:07:21, Lon: +008:39:39, elevation: 109 m;
- REYKJAVIK, IS, Lat: +64:07:37, Lon: −021:54:09, elevation: 52 m;
- OSLO-BLINDERN, NO, Lat: +59:56:34, Lon: +010:43:14, elevation: 94 m.

The datasets are summarized in Table 1, considering the daily mean temperature (x_1) in [°C], daily humidity (x_2) in [%], and daily mean wind speed (x_3) in [m/s]; SD is the standard deviation. The mean and standard deviation of weather time series change over weeks, months, and years, characterizing nonstationarities in many time granularities. The Olso-Blindern dataset contains 5 missing humidity values (06/07/2011, 11/22/2012, 12/03/2012, 01/12/2013, 12/11/2013). We imputed the mean value. The datasets are available at eca.knmi.nl, see also [13]. The data were subsequently normalized in the range [0, 1].

The task of the prediction models is to give one-step forecasts of the daily mean temperature $y^{[h+1]}$ using the last 3 observations of the daily mean temperature (x_1), air humidity (x_2) and mean wind speed (x_3). Therefore, the input vector contains 9 attributes, i.e., $x^{[h]} = (x_1^{[h-2]}, x_1^{[h-1]}, x_1^{[h]}, x_2^{[h-2]}, x_2^{[h-1]}, x_2^{[h]}, x_3^{[h-2]}, x_3^{[h-1]}, x_3^{[h]})$.

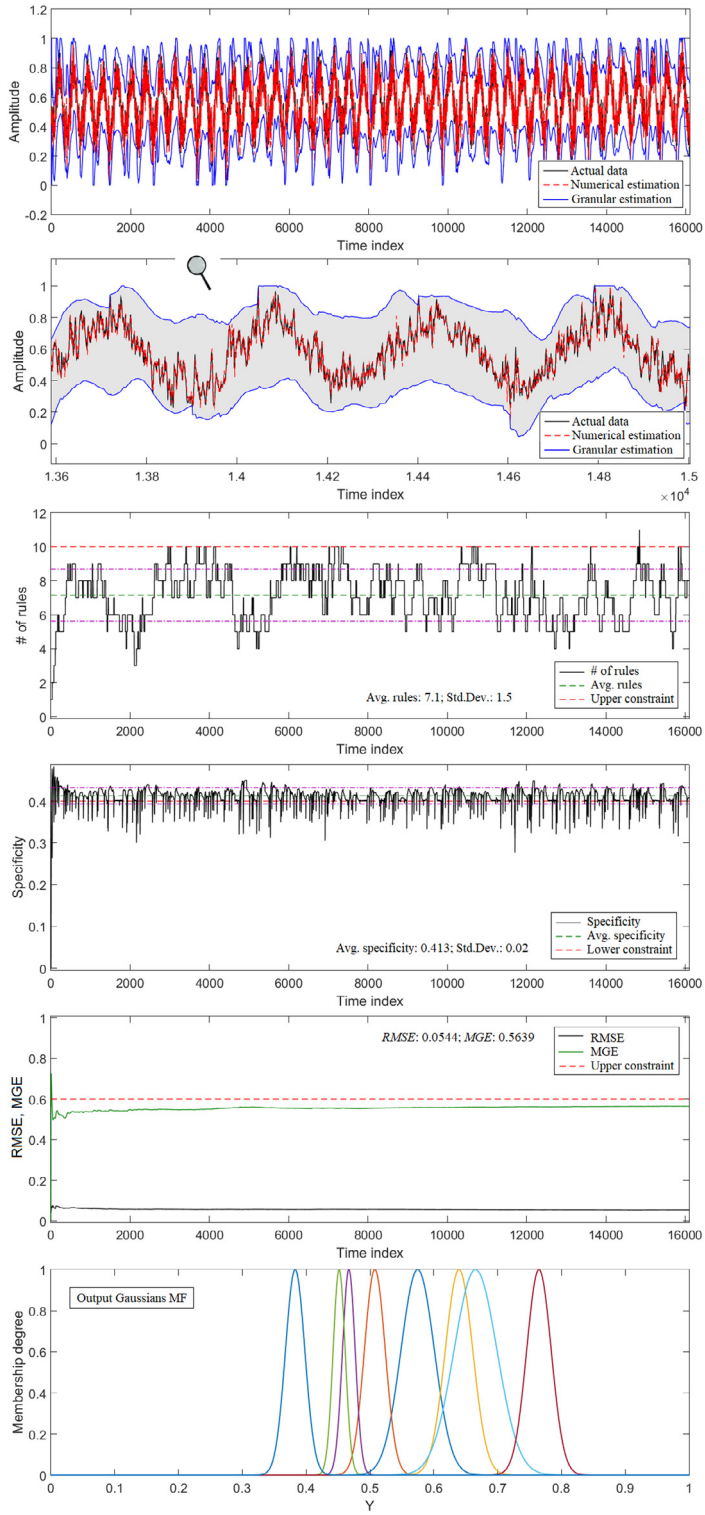

Fig. 2. eOGS prediction results: numerical and granular estimates of the Frankfurt daily mean temperature in the next day; evolution of the number of rules over time; average specificity of the granular map; evolution of the numerical and granular error indices over time; and final output Gaussian membership functions.

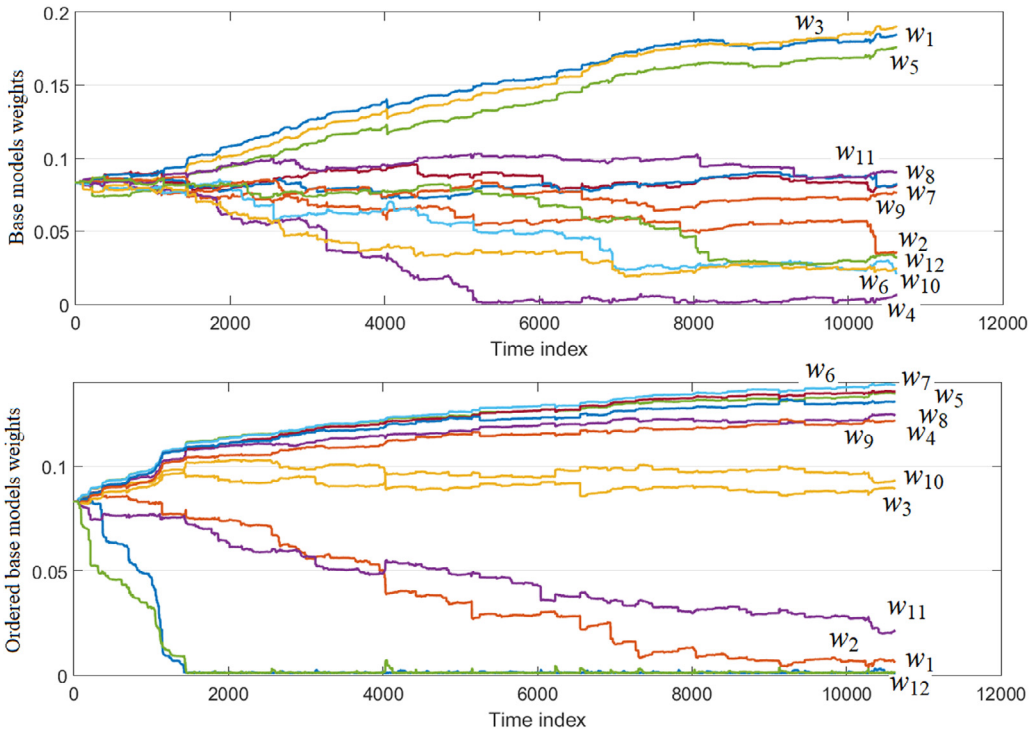

Fig. 3. Evolution of the ensemble base models weights for the Paris-Orly weather station using weighted arithmetic mean (top) and OWA (bottom) aggregation functions.

Table 1
Daily mean time series from different weather stations.

Station	Samples	From	To	Mean ± SD (x_1)	Mean ± SD (x_2)	Mean ± SD (x_3)
PARIS–ORLY	10,623	01/01/1990	01/31/2019	12.04 ± 6.62	73.70 ± 18.78	3.86 ± 1.73
FRANKFURT–MAIN	16,102	01/01/1975	01/31/2019	10.63 ± 7.52	74.38 ± 13.08	3.26 ± 1.60
REYKJAVIK	18,262	01/01/1967	12/31/2016	4.70 ± 5.15	77.90 ± 9.76	5.24 ± 2.76
OSLO–BLINDERN	21,581	01/01/1960	01/31/2019	6.27 ± 8.28	73.43 ± 15.87	2.73 ± 1.58

x_1: daily mean temperature [°C]; x_2: daily humidity [%]; x_3: daily mean wind speed [m/s]

Online learning methods employ the sample-per-sample testing-before-training approach, as follows. First, an estimation $\bar{y}^{[h+1]}$ is obtained for a given input $x^{[h]}$. One time step later, the actual value $y^{[h+1]}$ becomes available, and model adaptation is performed if necessary. eOGS models are evaluated individually for each weather station, and as part of an ensemble of 12 eOGS models. In the ensemble cases, different averaging aggregation functions (weighted arithmetic mean, median, OWA, and linear non-inclusive centered OWA) are studied with a focus on the accuracy of the numerical predictions.

5. Results and discussions

eOGS models were designed with focus on the numerical prediction accuracy, as in (32). In other words, the primary objective of the eOGS models is the minimization of the root mean square error given by the square of the difference between actual and estimated daily mean temperatures. Additionally, ϵ_1 and ϵ_2 are upper boundaries for the mean granular error and the number of rules in the rule base, respectively; and ϵ_3 is the lower boundary for the specificity of the granular map. Naturally, ϵ_i, $i = 1, 2, 3,$ is a preference. Different users may choose different values and, therefore, produce different models. Table 2 shows the results for 12 eOGS models constructed from combinations of ϵ_i. The Paris–Orly, Frankfurt–Main, Reykjavik, and Oslo–Blindern weather stations were taken into consideration. The table also compares the results of the 12 eOGS models, as base models of ensembles, individually. Four ensemble structures were evaluated using different aggregation functions: weighted arithmetic mean (WAM), ordered weighted averaging (OWA), median, and linear non-inclusive central OWA.

From Table 2, we observe that all ensembles outperformed all individual models for the weather stations under consideration. The central OWA aggregation function provides slightly better results compared to the OWA and Median functions. In common, and different from the WAM function, these functions order the input data before weighting and processing the data. In general, a reason for the relatively better estimates for Oslo and worse estimates for Reykjavik can be attributed to,

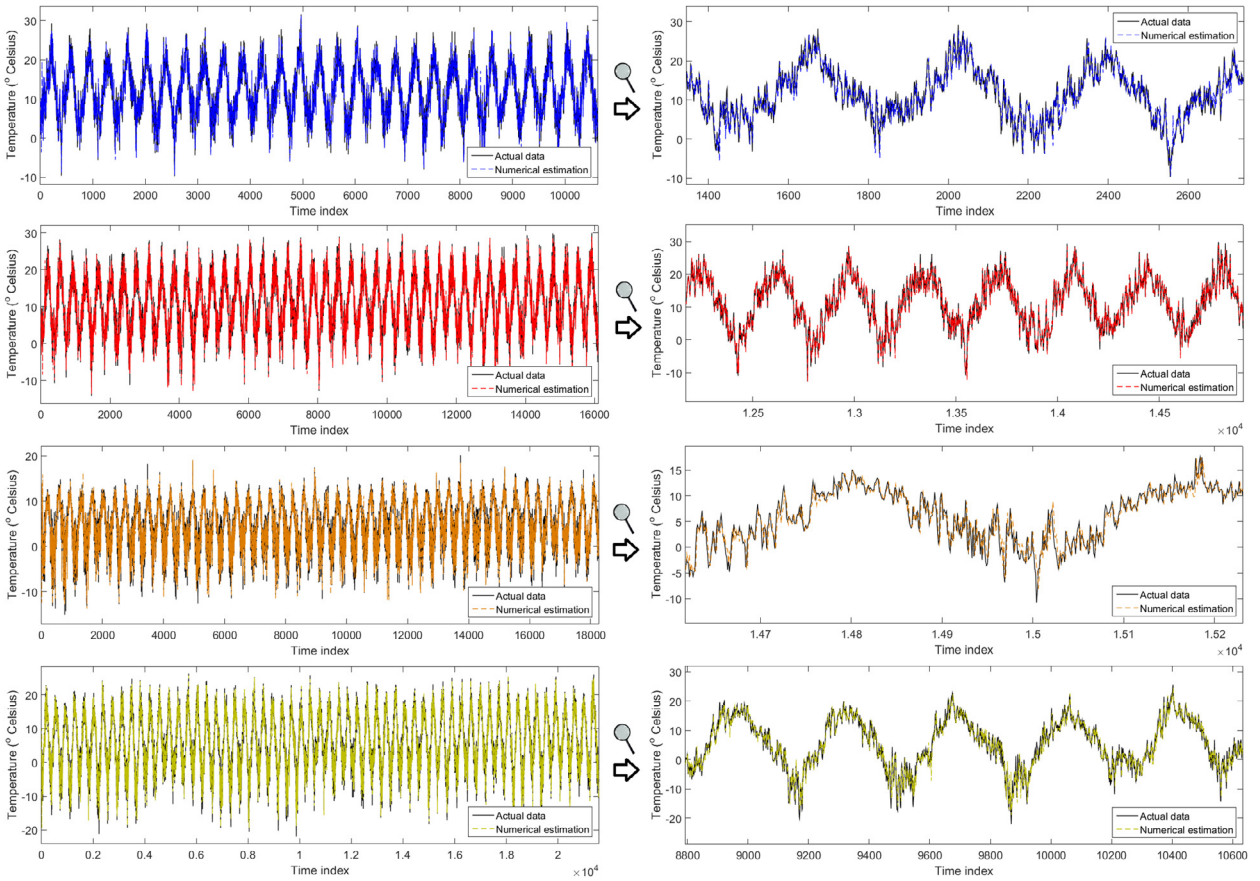

Fig. 4. Best numerical estimates for: (top) Paris–Orly using OWA Ensemble; (top middle) Frankfurt–Main using Central OWA Ensemble; (bottom middle) Reykjavik using Median Ensemble; (bottom) Oslo-Blindern using Central OWA Ensemble.

respectively, a lower and higher mean wind speed and variance in these cities, as shown in Table 1, which affect the daily mean temperature.

To illustrate the results generated by an individual eOGS model, we consider eOGS-1 and the Frankfurt station. In this case, the overall performance of the model is $RMSE = 0.0544$, $MGE = 0.5639$, $c = 8$, and $\text{Spa}(\gamma) = 0.4130$. Notice that the MGE is less than $\epsilon_1 = 0.6$; c is less than $\epsilon_2 = 10$, and $\text{Spa}(\gamma)$ is greater than $\epsilon_3 = 0.4$, as expected; see Table 2. The time spent to process 16,099 data samples was 18.06 s in a i7-8550U CPU dual-core 1.88–1.99 GHz processor with 8GB of RAM, which gives an average of 1.1 ms per sample. Such data processing and model adaptation speed makes the proposed method suitable for several meteorological applications and for other applications involving very-large and high-frequency data streams. By means of parallelism of the base models, the ensemble can also perform online adaptation and prediction in sampling frequencies similar to that of individual eOGS models. The key issue is to keep the fuzzy rule base of each base model compact. This is achieved by means of the merging procedure, the mechanism to delete rules that are no longer used, and the explicit constraint on the number of rules. Fig. 2 shows the results for the Frankfurt–Main (Feldbergstr.) station using eOGS-1.

From Fig. 2, we observe that eOGS provides an envelope around the actual data along with the numerical prediction. The upper and lower bounds of the enclosure may have different meanings and be useful to support decisions and actions in particular applications. The enclosure can be made narrower if we demand a smaller MGE and allow the model to evolve a larger number of rules (more specific granules) regarding the formulation of the original optimization problem. The intermediate plots show the evolution of the rule base, specificity of the granular map, and error indices. The average number of rules was 7.1, with a standard deviation of 1.5 rules over time. The number of rules becomes larger than the allowed value only once (see iteration 15,000, approximately). As soon as the constraint ϵ_2 is violated, the parameters $\mathbb{P}$ are automatically changed, and the most similar granules and rules are merged. The final output Gaussian membership functions are shown in the bottom plot. In the last iterations, the Gaussians were concentrated approximately in the range of amplitudes between 0.33 and 0.82 (see x axis). The Gaussians usually drift, to the left and to the right, along the time steps, according to the oscillations of amplitude of the flowing data. New Gaussians may be included; highly overlapped Gaussians can be merged; and inactive Gaussians can be deleted by the eOGS online learning algorithm.

Table 2
Numerical prediction results for the individual eOGS models and the ensembles using different aggregation functions.

–	Constraints			*RMSE* (Place)			
Base model	ϵ_1	ϵ_2	ϵ_3	Paris	Frankfurt	Reykjavik	Oslo
eOGS-1	0.6	10	0.4	0.0565 (6th)	0.0544 (5th)	0.0689 (5th)	0.0574 (7th)
eOGS-2	0.5	10	0.4	0.0626 (12th)	0.0676 (15th)	0.0841 (15th)	0.0687 (15th)
eOGS-3	0.6	7	0.4	0.0563 (5th)	0.0547 (6th)	0.0696 (6th)	0.0573 (6th)
eOGS-4	0.5	7	0.4	0.0654 (16th)	0.0690 (16th)	0.0847 (16th)	0.0673 (11th)
eOGS-5	0.6	5	0.4	0.0569 (7th)	0.0567 (7th)	0.0701 (7th)	0.0567 (5th)
eOGS-6	0.5	5	0.4	0.0631 (15th)	0.0665 (13th)	0.0838 (14th)	0.0667 (10th)
eOGS-7	0.6	10	0.5	0.0608 (10th)	0.0623 (9th)	0.0783 (9th)	0.0697 (16th)
eOGS-8	0.5	10	0.5	0.0607 (9th)	0.0662 (12th)	0.0809 (11th)	0.0682 (14th)
eOGS-9	0.6	7	0.5	0.0610 (11th)	0.0630 (10th)	0.0783 (9th)	0.0664 (9th)
eOGS-10	0.5	7	0.5	0.0630 (14th)	0.0666 (14th)	0.0811 (12th)	0.0680 (13th)
eOGS-11	0.6	5	0.5	0.0604 (8th)	0.0612 (8th)	0.0776 (8th)	0.0622 (8th)
eOGS-12	0.5	5	0.5	0.0627 (13th)	0.0659 (11th)	0.0815 (13th)	0.0678 (12th)
Ensemble (WAM)				0.0542 (4th)	0.0518 (4th)	0.0643 (4th)	0.0503 (4th)
Ensemble (OWA)				**0.0537** (1st)	0.0511 (2nd)	0.0625 (3rd)	0.0490 (3rd)
Ensemble (Median)				0.0540 (3rd)	0.0512 (3rd)	**0.0624** (1st)	**0.0488** (1st)
Ensemble (Central OWA)				**0.0537** (1st)	**0.0509** (1st)	**0.0624** (1st)	**0.0488** (1st)

The ensemble that uses the central OWA aggregation function produced the overall best numerical estimates, as shown in Table 2. The central OWA function operates with constant weights, which prioritize the middle elements after the input vector, $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_{12})$, is sorted in descending order. For 12 base models, the central OWA weights are

$$\mathbf{w}_{(CentralOWA)} = (0, 0.0333, 0.0667, 0.1000, 0.1333, 0.1667, 0.1667, 0.1333, 0.1000, 0.0667, 0.0333, 0),$$

i.e., the extremes are excluded, and the central elements, w_6 and w_7, are more relevant to the estimates during the whole data processing cycle. The median function uses

$$\mathbf{w}_{(Median)} = (0,\ 0,\ 0,\ 0,\ 0,\ 0.5,\ 0.5,\ 0,\ 0,\ 0,\ 0,\ 0)$$

to achieve the second-best result. The weights are also constant, and applied over input vectors sorted in descending order. Emphasis is entirely on w_6 and w_7, and x_6 and x_7.

Weights of the weighted arithmetic mean and OWA aggregation functions are updated according to the quadratic programming problem given in (6) and (7). In the weighted-arithmetic-mean case, a weight is associated to the prediction of a specific base model. For example, for the Paris weather station, the final weight vector (iteration 10620) is

$$\mathbf{w}_{WAM} = (0.1845, 0.0354, 0.1900, 0.0063, 0.1759, 0.0211, 0.0813, 0.0821, 0.0765, 0.0244, 0.0904, 0.0321).$$

Notice that the highest weight is w_3, followed by w_1 and w_5, and that the best base model for Paris, according to Table 2, is eOGS-3, followed by eOGS-1 and eOGS-5. In other words, the most accurate eOGS base models are underlined along the iterations to contribute more to the ensemble estimates. Additionally, for the Paris station, the final OWA weights are given as

$$\mathbf{w}_{OWA} = (0.0010, 0.0064, 0.0894, 0.1247, 0.1349, 0.1391, 0.1359, 0.1310, 0.1219, 0.0931, 0.0212, 0.0013).$$

Interestingly, in spite of the asymmetry, the OWA weights approach the weights of a central OWA function along the iterations. Fig. 3 shows the evolution of the weighted-arithmetic-mean and OWA-aggregation-function weights over time. In the former case, the weights related to the best base models prevail, see w_3, w_1, and w_5. In the latter case, the middle elements, w_6 and w_7, prevail, similarly to the weights of other centered aggregation functions. The same results, in essence, were obtained for the other time series.

Finally, the best numerical estimates for Paris, Frankfurt, Reykjavik, and Oslo, given respectively by the OWA, Central-OWA, Median, and Central-OWA functions, are shown in Fig. 4. The results are quite accurate and, in general, may come with additional information from the fuzzy granular framework, such as granule bounds, enclosures, Gaussian membership functions, and linguistic description by means of rules.

6. Conclusion

We introduced an ensemble approach for nonstationary time series prediction in which the base models evolve their granular rule-based structures and parameters over time. The base models are guided by constrained optimization problems whose primary objective is to minimize the root mean square error between estimates and actual time-series values. The specificity-compactness tradeoff and the variability and coverage of the data along the process of data stream modeling are taken into consideration. The contributions of the base models to the ensemble estimation are merged using different averaging aggregation functions, such as ordered weighted averaging, weighted arithmetic mean, median, and linear non-inclusive centered OWA. These functions were comparatively studied.

Real multivariate time series from the Paris–Orly, Frankfurt–Main, Reykjavik, and Oslo–Blindern weather stations were used for model development and performance evaluation. We aimed to estimate the daily mean temperature from the last 3 available temperature values, and from exogenous attributes such as air humidity and mean wind speed. We observed that all ensembles performed better than all base evolving models individually. Moreover, in general, the predictions for Oslo and Reykjavik were the most and least accurate, respectively, due to a lower and higher mean wind speed and data variance in these cities, which affect the daily mean temperature.

The proposed linear non-inclusive central OWA aggregation function was superior to the remaining functions by a small margin, especially compared to other functions that also emphasize the elements in the middle, i.e., OWA and median. Averaging the estimates and excluding the extremes have shown to be important characteristics because evolving models are constantly changing and, therefore, are subject to transient periods. For example, the creation of a new information granule and rule requires some steps for an appropriate adjustment of the local parameters. The estimations of such a base model during a transient may come with a level of uncertainty. The ensemble plays a key role in this case, providing robustness and more consistent estimations. In addition, the proposed evolving-optimal-granular-modeling approach provides numerical predictions as well as granular predictions of time series. Lower and upper prediction bounds encapsulate the actual data and may be important information to encourage decision making in a variety of applications.

Averaging aggregation to fuse information within ensembles should be evaluated in other data stream applications in the future. Other classes of problems, such as granular data-stream processing, and nonuniform sampling, will be investigated.

Conflicts of interest

The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Acknowledgment

This work was supported by the Serrapilheira Institute (grant number Serra-1812-26777). Igor Škrjanc is grateful to the Slovenian Research Agency - Program P2-0219: Modeling, Simulation and Control.

References

[1] P. Angelov, X. Gu, MICE: multi-layer multi-model images classifier ensemble, in: Proc. IEEE Int. Conf. Cybern., 2017, p. 8.
[2] P. Angelov, Autonomous Learning Systems: From Data Streams to Knowledge in Real-time, Wiley, 2013.
[3] G. Beliakov, H. Sola, T. Sanchez, A Practical Guide to Averaging Functions, 329, Springer: Cham, Studies in Fuzziness and Soft Computing, 2016.
[4] L. Bueno, P. Costa, I. Mendes, E. Cruz, D. Leite, Evolving ensemble of fuzzy models for multivariate time series prediction, in: IEEE Int Conf on Fuzzy Systems (FUZZ-IEEE), 2015, p. 6p.
[5] V. Chankong, Y. Haimes, Multiobjective Decision Making Theory and Methodology, North-Holland, NY, 1983.
[6] D. Dovžan, V. Logar, I. Škrjanc, Implementation of an evolving fuzzy model (EFUMO) in a monitoring system for a waste-water treatment process, IEEE Trans. Fuzzy Syst. 23 (5) (2015) 1761–1776.
[7] J. Gama, Knowledge Discovery from Data Streams, Chapman & Hall/CRC: Boca Raton, Florida, 2010.
[8] M. Heeswijk, Y. Miche, T. Lindh-Knuutila, P. Hilbers, T. Honkela, E. Oja, A. Lendasse, Adaptive Ensemble Models of Extreme Learning Machines for Time Series Prediction, in: C. Alippi, M. Polycarpou, C. Panayiotou, G. Ellinas (Eds.), Artificial Neural Networks - ICANN Lecture Notes in Computer Science, volume 5769, Springer, Berlin, Heidelberg, 2009.
[9] J.A. Iglesias, A. Ledezma, A. Sanchis, An ensemble method based on evolving classifiers: eStacking, IEEE Symp. Evol. Auton. Learn. Syst. (EALS) (2014) 8p.
[10] H. Ju, W. Pedrycz, H. Li, W. Ding, X. Yang, X. Zhou, Sequential three-way classifier with justifiable granularity, Knowl.-Based Syst. 163 (2019) 103–119.
[11] D. Kangin, P. Angelov, J.A. Iglesias, A. Sanchis, Evolving classifier TEDAClass for big data, Procedia Comput. Sci. 53 (2015) 9–18.
[12] N. Kasabov, Evolving Connectionist Systems: The Knowledge Engineering Approach, second ed., Springer, 2007.
[13] T. Klein, et al., Daily dataset of 20th-century surface air temperature and precipitation series for the european climate assessment, Int. J. Climatol. 22 (12) (2002) 1441–1453.
[14] J. Kolter, M. Maloof, Dynamic weighted majority: an ensemble method for drifting concepts, J. Mach. Learn. Res. 8 (2007) 2755–2790.
[15] D. Leite, G. Andonovski, I. Škrjanc, F. Gomide, Optimal rule-based granular systems from data streams, IEEE Trans. Fuzzy Syst. (2019) 14, doi:10.1109/TFUZZ.2019.2911493.
[16] D. Leite, Comparison of Genetic and Incremental Learning Methods for Neural Network-based Electrical Machine Fault Detection, in: E. Lughofer, M. Sayed-Mouchaweh (Eds.), Predictive Maintenance in Dynamic Systems, Springer, 2019, pp. 231–268. Cham
[17] D. Leite, R. Palhares, V. Campos, F. Gomide, Evolving granular fuzzy model-based control of nonlinear dynamic systems, IEEE Trans. Fuzzy Syst. 23 (4) (2015) 923–938.
[18] D. Leite, P. Costa, F. Gomide, Evolving granular neural networks from fuzzy data streams, Neural Netw. 38 (2013) 1–16.
[19] D. Leite, R. Ballini, P. Costa, F. Gomide, Evolving fuzzy granular modeling from nonstationary fuzzy data streams, Evol. Syst. 3 (2) (2012) 65–79.
[20] S. Liu, W. Pedrycz, A. Gacek, Y. Dai, Development of information granules of higher type and their applications to granular models of time series, Eng. Appl. Artif. Intell. 71 (2018) 60–72.
[21] E. Lughofer, S. Kindermann, M. Pratama, J. Rubio, Top-down sparse fuzzy regression modeling from data with improved coverage, Int. J. Fuzzy Syst. 19 (5) (2017) 1645–1658.
[22] E. Lughofer, O. Buchtala, Reliable all-pairs evolving fuzzy classifiers, IEEE Trans. Fuzzy Syst. 21 (4) (2013) 625–641.
[23] E. Lughofer, Evolving Fuzzy Systems: Methodologies, Advanced Concepts and Applications, Springer, Verlag Berlin Heidelberg, 2011.
[24] L. Maciel, R. Ballini, F. Gomide, Evolving granular analytics for interval time series forecasting, Granul. Comput. 1 (4) (2016) 213–224.
[25] P. Melin, J. Soto, O. Castillo, J. Soria, A new approach for time series prediction using ensembles of ANFIS models, Expert Syst. Appl. 39 (2012) 3494–3506.

[26] H.L. Nguyen, Y.K. Woon, W.K. Ng, A survey on data stream clustering and classification, Knowl. Inf. Syst. 45 (3) (2015) 535–569.
[27] W. Pedrycz, W. Homenda, Building the fundamentals of granular computing: a principle of justifiable granularity, Appl. Soft Comput. 13 (10) (2013) 4209–4218.
[28] W. Pedrycz, R. Al-Hmouz, A. Morfeq, A. Balamash, The design of free structure granular mappings: the use of the principle of justifiable granularity, IEEE Trans. Cybern. 43 (6) (2013) 2105–2113.
[29] W. Pedrycz, J. Berezowski, I. Jamal, A granular description of data: a study in evolvable systems, in: M. Sayed-Mouchaweh, E. Lughofer (Eds.), Learning in Non-Stationary Environments: Methods and Applications, Springer, New York, 2012, pp. 57–76.
[30] R. Polikar, Ensemble based systems in decision making, IEEE Circ. Syst. Mag. 6 (3) (2006) 21–45.
[31] M. Pratama, W. Pedrycz, E. Lughofer, Evolving ensemble fuzzy classifier, IEEE Trans. Fuzzy Syst. 26 (5) (2018) 2552–2567.
[32] M. Raza, N. Mithulananthan, A. Summerfield, Solar output power forecast using an ensemble framework with neural predictors and Bayesian adaptive combination, Sol. Energy 166 (2018) 226–241.
[33] J.J. Rubio, E. Lughofer, J.A. Meda-Campana, L.A. Paramo, J.F. Novoa, J. Pacheco, Neural network updating via argument Kalman filter for modeling of Takagi-Sugeno fuzzy models, J. Intell. Fuzzy Syst. 35 (2) (2018) 2585–2596.
[34] M. Sayed-Mouchaweh, E. Lughofer, Learning in Non-Stationary Environments: Methods and Applications, Springer, New York, 2012.
[35] S. Silva, P. Costa, M. Gouvea, A. Lacerda, F. Alves, D. Leite, High impedance fault detection in power distribution systems using wavelet transform and evolving neural network, Electr. Power Syst Res 154 (2018) 474–483.
[36] E. Soares, P. Costa, B. Costa, D. Leite, Ensemble of evolving data clouds and fuzzy models for weather time series prediction, Appl. Soft Comput. 64 (2018) 445–453.
[37] I. Škrjanc, J. Iglesias, A. Sanchis, D. Leite, E. Lughofer, F. Gomide, Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: a survey, Inf. Sci. 490 (2019) 344–368.
[38] I. Škrjanc, Fuzzy confidence interval for ph titration curve, Appl. Math. Model. 35 (8) (2011) 4083–4090.
[39] S.M. Stiegler, A modest proposal: a new standard for the normal, Am. Stat. (1982). 36-2, JSTOR.
[40] S. Tomažič, D. Dovžan, I. Škrjanc, Confidence-interval fuzzy model-based indoor localization, IEEE Trans. Ind. Electron. 66 (3) (2019) 2015–2024.
[41] V. Torra, OWA operators in data modeling and reidentification, IEEE Trans. Fuzzy Syst. 12 (2004) 652–660.
[42] X. Wang, W. Pedrycz, A. Gacek, X. Liu, From numeric data to information granules: a design through clustering and the principle of justifiable granularity, Knowl.-Based Syst. 101 (2016) 100–113.
[43] Z.S. Xu, An overview of methods for determining OWA weights, Int. J. Intell. Syst. 20 (2005) 843–865.
[44] R. Yager, N. Alajlan, On characterizing features of OWA aggregation operators, Fuzzy Optim. Decis. Making 13 (2014) 1–32.
[45] R. Yager, Measures of specificity over continuous spaces under similarity relations, Fuzzy Sets Syst. 159 (17) (2008) 2193–2210.
[46] R. Yager, Centered OWA operators, Soft. Comput. 11 (2007) 631–639.
[47] R. Yager, On ordered weighted averaging aggregation operators in multicriteria decision making, IEEE Trans. Syst. Man Cybern. 18 (1988) 183–190.
[48] J. Yao, A. Vasilakos, W. Pedrycz, Granular computing: perspectives and challenges, IEEE Trans. Cybern. 43 (6) (2013) 1977–1989.
[49] L.A. Zadeh, Generalized theory of uncertainty (GTU) - principal concepts and ideas, Comput. Stat. Data Anal. 51 (1) (2006) 15–46.
[50] Z.H. Zhou, Ensemble Methods: Foundations and Algorithms, CRC Press, Boca Raton, FL, 2012.