Soft Comput (2017) 21:2357–2366
DOI 10.1007/s00500-015-1946-4

METHODOLOGIES AND APPLICATION

MSAFIS: an evolving fuzzy inference system

José de Jesús Rubio[1] · Abdelhamid Bouchachia[2]

Published online: 19 November 2015

Abstract In this paper, the problem of learning in big data is considered. To solve this problem, a new algorithm is proposed as the combination of two important evolving and stable intelligent algorithms: the sequential adaptive fuzzy inference system (SAFIS), and stable gradient descent algorithm (SGD). The modified sequential adaptive fuzzy inference system (MSAFIS) is the SAFIS with the difference that the SGD is used instead of the Kalman filter for the updating of parameters. The SGD improves the Kalman filter, because it first obtains a better learning in big data. The effectiveness of the introduced method is verified by two experiments.

Keywords Intelligent systems · Gradient descent · Learning · Big data

1 Introduction

The recent years have witnessed the emergence of an important topic related to process learning which is learning from big data (LBD). LBD is concerned with the development and application of learning algorithms for very large, possibly complex, datasets that cannot be accommodated in the main memory. To cope with this requirement, different techniques and technologies have been proposed:

1. Parallel and distributed computing (e.g., Hadoop): data are split into portions and sent to parallel machines to be processed and learned from.
2. Online learning, known also as sequential learning, one-pass learning, real-time learning, evolving systems, etc.: the learning algorithms learn sequentially, either batch-based or point-based, potentially using one single machine.

Although these techniques are not new from a pure scientific point of view, the deluge of data available everywhere has given a refreshing and renewable interest to them. In this paper, we will focus on online learning.

Online learning faces the challenge of accurately estimating models using incoming data whose statistical characteristics are not known a priori. In non-stationary environments, the challenge becomes even more important, since the model's behavior may need to change drastically over time (Gama et al. 2014). Online learning aims at ensuring continuous adaptation of the model being fitted to the data. When learning, ideally only the model should be stored in memory. For instance in rule-based systems (RBS), only rules should be memorized. The model is then adjusted in future learning steps. In the case of RBS, as new data arrive, new rules may be created and existing ones may be modified or removed allowing the overall model to evolve over time (Bouchachia and Vanaret 2014; Rubio et al. 2011). In Precup et al. (2015), online fuzzy models are discussed. In general evolving systems are online learning algorithms whose structure and parameters are very flexible in order to adapt to ever-changing

Communicated by V. Loia.

✉ José de Jesús Rubio
jrubioa@ipn.mx; rubio.josedejesus@gmail.com

Abdelhamid Bouchachia
abouchachia@bournemouth.ac.uk

1 Sección de Estudios de Posgrado e Investigación, ESIME Azcapotzalco, Instituto Politécnico Nacional, Av. de las Granjas no. 682, Col. Santa Catarina, 02250 Mexico D.F., Mexico

2 Department of Computing and Informatics, Faculty of Science and Technology, Bournemouth University, Dorset, UK

environments (Angelov et al. 2010; Bouchachia 2008; Gama 2010; Kasabov 2007; Lughofer 2011; Sayed-Mouchaweh and Lughofer 2012). Online processing of data with a particular focus on the design issues of online evolving systems is considered in Bouchachia (2014). In Bouchachia and Vanaret (2014), online self-learning fuzzy classifier, called GT2FC standing for "Growing Type-2 Fuzzy Classifier" is presented. The proposed approach shows how type-2 fuzzy rules can be learned online in an evolving way from data streams. GT2FC was applied in the context of smart homes. In Bouchachia et al. (2014), the authors explore the application of interactive and online learning of user profiles in the context of information filtering using evolutionary algorithms. In Iglesias et al. (2014), an evolving algorithm for learning computer user behavior is introduced.

Evolving systems have been very popular, for instance in Bordignon and Gomide (2014), a learning approach to train uninorm-based hybrid neural networks is mentioned. The use of evolving classifiers for activity recognition is described in Garcia-Cuesta and Iglesias (2012) and Ordoñez et al. (2013). In Gomide and Lughofer (2014), Iglesias and Skrjanc (2014), and Lughofer and Sayed-Mouchaweh (2015), novel efficient techniques of evolving intelligent systems are discussed. A dynamic pattern recognition method is introduced in Hartert and Sayed-Mouchaweh (2014). In Iglesias et al. (2015), an approach for classifying huge amounts of different news articles is designed. An evolving method that is able to keep track of computer users is proposed in Iglesias et al. (2014). In Klancar and Skrjanc (2015), a new approach called evolving principal component clustering is addressed. A new clustering method is suggested in Lughofer and Sayed-Mouchaweh (2015). In Lughofer et al. (2015) and Pratama et al. (2015), novel evolving fuzzy rule-based classifiers are addressed. An evolving neural fuzzy modeling approach is constructed in Marques Silva et al. (2014). In Sayed-Mouchaweh and Lughofer (2015), a novel approach in fault diagnosis is studied. Stable systems are characterized by the boundedness criterion, i.e., if bounded algorithms inputs are employed, then the outputs and parameters exponentially decay to a small and bounded zone. In Ahn (2014), the author uses an induced $L\infty$ approach to create a new filter with a finite impulse response structure for state-space models with external disturbances. The model predictive stabilization problem for Takagi–Sugeno fuzzy multilayer neural networks with general terminal weighting matrix are investigated in Ahn and Lim (2013). In Ahn (2012), an error passivation approach is used to derive a new passive and exponential filter for switched Hopfield neural networks with time-delay and noise disturbance. Two robust intelligent controllers for nonlinear systems with dead-zone are addressed in Perez-Cruz et al. (2014) and Perez-Cruz et al. (2014). In Torres et al. (2014) and Zdesar et al. (2014), two stable controllers are introduced. However, most of these algorithm operate offline and are not designed to handle big data. The present paper presents the combination of two algorithms: the sequential adaptive fuzzy inference system (SAFIS) (Rong et al. 2006) which is an evolving algorithm and the stable gradient descent algorithm (SGD) (Rubio et al. 2011) which is a stable algorithm. Such combination, called the Modified Sequential Adaptive Fuzzy Inference System (MSAFIS), aims to devise an efficient evolving algorithm that can cope with data streams as a case of dig data. MSAFIS exploits the SGD algorithm to update of parameters, while in SAFIS relies on the Kalman filter. SGD has the advantage that it outperforms Kalman filter (Rubio et al. 2011).

The paper is organized as follows. In Sect. 2, the SAFIS, SGD, and MSAFIS algorithms are detailed. In Sect. 3, the brain encephalography (EEG) and the eye electro-oculogram (EOG) signals are described. Using an EEG and EOG dataset, SAFIS, SGD and MSAFIS are evaluated and compared in Sect. 4. Section 5 concludes the paper and suggests future research directions.

2 Presentation of the algorithms

In this section, the three algorithms SAFIS, SGD, and MSAFIS are described. Furthermore, the differences of the three algorithms are explained.

2.1 SAFIS algorithm

The sequential adaptive fuzzy inference system (SAFIS) is developed based on the functional equivalence between a radial basis function network and a fuzzy inference system (FIS) resulting in a neuro-fuzzy system. In SAFIS, the concept of "Influence" of a fuzzy rule is introduced and using this the fuzzy rules are added or removed based on the input data received so far. If the input data do not warrant adding of fuzzy rules, then only the parameters of the "closest" (in a Euclidean sense) rule are updated using an extended Kalman filter (EKF) scheme.

The SAFIS algorithm is summarized as below (Rong et al. 2006):

For each observation $(z(k), y(k))$ where $z(k) \in \Re^N$, $y(k) \in \Re$ and $k = 1, 2, \ldots,$ do

1. Compute the overall system output:

$$\widehat{y}(k) = \frac{\sum_{j=1}^{M} o_j(k) R_j(z_i(k))}{\sum_{j=1}^{M} R_j(z_i(k))} \tag{1}$$

where

$$R_j(z_i(k)) = \exp\left(-\frac{1}{\delta_j^2(k)}\left\|z_i(k) - m_j(k)\right\|^2\right)$$

and M is the number of fuzzy rules, $R_j(z_i(k))$ is the firing strength of the jth rule, $o_j(k)$ is the weight of the normalized rule. Note that each rule is represented as a radial basis function described by its center $m_j(k)$ and its spread $\delta_j(k)$.

2. Calculate the parameters required in the growth criterion:

$$\epsilon(k) = \max\left\{\epsilon_{\max}\tau^k, \epsilon_{\min}\right\}, \quad 0 < \tau < 1, \tag{2}$$

where $\epsilon_{\max}$ and $\epsilon_{\min}$ are the threshold largest and smallest distances admitted between the inputs and corresponding nearest center of rules. The parameter $\tau (0 < \tau < 1)$ indicates the decay constant. The error of the kth input is given as follows:

$$\widetilde{y}(k) = y(k) - \widehat{y}(k), \tag{3}$$

where $y(k)$ and $\widehat{y}(k)$ are the output and the estimated output, respectively.

3. Apply the criterion for adding rules, if the following two conditions are satisfied: if

$$\left\|z_i(k) - m_j(k)\right\| > \epsilon(k) \tag{4}$$

and

$$Y_{\inf}(M+1) = |\widetilde{y}(k)| \frac{\left(1.8K\left\|z_i(k) - m_j(k)\right\|\right)^N}{\sum_{j=1}^{M+1}(1.8\delta_j(k))^N} > y_g \tag{5}$$

where y_g is the growing threshold. A new rule $M+1$ is added if y_g is exceeded. The new rule $M+1$ is given as follows:

$$\begin{aligned} o_{M+1}(k) &= \widetilde{y}(k) \\ m_{M+1}(k) &= z_i(k) \\ \delta_{M+1}(k) &= K\left\|z_i(k) - m_{M+1}(k)\right\| \end{aligned} \tag{6}$$

If no rule is added, the nearest rule jm is obtained as follows:

$$\min_j R_j(z(k)) \Longrightarrow jm = j \tag{7}$$

and adjust the system parameters $o_j(k)$, $m_j(k)$, $\delta_j(k)$ for the nearest rule only by using the extended Kalman filter (EKF) method:

$$\begin{aligned} \varphi(k) &= \varphi(k-1) + P_{k-1}b(k-1) \\ &\quad \times\left[a + b^T(k-1)P_{k-1}b(k-1)\right]^{-1}\widetilde{y}(k) \\ P_k &= P_{k-1} - P_{k-1}b(k-1) \\ &\quad \times\left[p + b^T(k-1)P_{k-1}b(k-1)\right]^{-1} \\ &\quad \times b^T(k-1)P_{k-1} + qI \end{aligned} \tag{8}$$

where $\varphi(k) = [\varphi_1(k)\ldots\varphi_3(k)]^T = [m_{jm}(k), o_{jm}(k), \delta_{jm}(k)]^T$, $P_1 = qI$, q and p are parameters selected by the designer, $0 < q < 1$, $0 < p < 1$, $b(k) = [b_1(k), b_2(k), b_3(k)]^T$, $b_1(k) = \frac{2[o_{jm}(k)-\widehat{y}(k)]R_{jm}(z_i(k))[z_i(k)-m_{jm}(k)]}{\left[\sum_{j=1}^{M}R_j(z_i(k))\right]\delta_{jm}^2(k)}$, $b_2(k) = \frac{2[o_{jm}(k)-\widehat{y}(k)]R_{jm}(z_i(k))\|z_i(k)-m_{jm}(k)\|^2}{\left[\sum_{j=1}^{M}R_j(z_i(k))\right]\delta_{jm}^3(k)}$, $b_3(k) = \frac{R_{jm}(z_i(k))}{\left[\sum_{j=1}^{M}R_j(z_i(k))\right]}$, I is the identity matrix.

4. If the following criterion is satisfied:

$$Y_{\inf}(jm) = \left|o_{jm}(k)\right| \frac{\left(1.8\delta_{jm}(k)\right)^N}{\sum_{j=1}^{M}(1.8\delta_j(k))^N} < y_p \tag{9}$$

then remove the jm rule and reduce the dimensionality of EKF. Note that y_p is the pruning threshold.

Remark 1 The significance of a rule proposed in growing and pruning radial basis function (GAP-RBF) neural network is defined based on the average contribution of an individual rule to the output of the RBF network. Under this definition, one may need to estimate the input distribution range $S(z) = \frac{|o_{jm}(k)|}{\sum_{j=1}^{M}(1.8\delta_j(k))^N}$. However, the influence of a rule introduced in this paper is different from the significance of a rule proposed in GAP-RBF. In fact, the influence of a rule is defined as the relevant significance of the rule compared to summation of significance of all the existing RBF rules. As seen from Eq. (7), with the introduction of influence one need not estimate the input distribution range and the implementation has been simplified.

Remark 2 In parameter modification, SAFIS utilizes a winner rule strategy similar to the work done by Huang et al. (2004). The key idea of the winner rule strategy is that only the parameters related to the selected winner rule are updated by the EKF algorithm in every step. The 'winner rule' is defined as the rule that is closest (in the Euclidean distance sense) to the current input data. As a result, SAFIS is computationally efficient.

Remark 3 In SAFIS, some parameters need to be decided in advance according to the problems considered. They include the distance thresholds ($\epsilon_{\max}, \epsilon_{\min}, \tau$), the overlap factor K for determining the width of the newly added rule, the growing threshold (y_g) for a new rule and the pruning threshold (y_p) for removing an insignificant rule. A general selection procedure for the predefined parameters is given as follows: max is set to around the upper bound of input variables; $\epsilon_{\min}$ is set to around 10 % of $\epsilon_{\max}$; τ is set to around 0.99. y_p is set to around 10 % of y_g. $\epsilon_{\max}$ is observed in the range [1.0, 10.0]. The overlap factor K is utilized to initialize the width of the newly added rule and chosen according to different problems, it is observed in the range [1.0, 2.0]. The growing threshold y_g is chosen according to the system performance, it is observed in the range [0.001, 0.05] . The smaller the y_g, the better the system performance, but the resulting system structure is more complex.

2.2 SGD algorithm

The stable gradient descent (SGD) algorithm is developed with a new time-varying rate to guarantee its uniformly stability for online identification and its identification error converges to a small zone bounded by the uncertainty. The weights error is bounded by the initial weights error, i.e., hence the overfitting is avoided. The SGD algorithm is as follows (Rubio et al. 2011):

1. Compute the output of the nonlinear system $y(k)$ with Eq. (10). Note, that the nonlinear system may have the structure represented by Eq. (10) and the parameter N is selected according to this nonlinear system.

$$y(k) = f\,[z(k)]\,, \tag{10}$$

   where $z(k) = [z_1(k)\ldots, z_i(k), \ldots, z_N(k)]^T = [y(k-1), \ldots, y(k-n), u(k-1), \ldots, u(k-m)]^T \in \Re^{N\times 1}$ ($N = n+m$) is the input vector, $u(k-1) \in \Re$ is the input of the plant, $y(k) \in \Re$ is the output of the plant, and f is an unknown nonlinear function, $f \in C^\infty$.
2. Select the following parameters; $o(1)$ and $w(1)$ as random numbers between 0 and 1; M as an integer number, and α_0 as a positive value smaller or equal to 1; obtain the output $\widehat{y}(1)$ using Eq. (11).

$$\widehat{y}(k) = \sum_{j=1}^{M} o_j(k)\beta_j(k)$$
$$\beta_j(k) = \tanh\left(\sum_{i=1}^{N} w_{ij}(k)z_i(k)\right) \tag{11}$$

3. For each iteration k, obtain the output $\widehat{y}(k)$ with Eq. (11), obtain the identification error $\widetilde{y}(k)$ with Eq. (12):

$$\widetilde{y}(k) = \widehat{y}(k) - y(k) \tag{12}$$

   and update the parameters $o_j(k)$ and $w_{ij}(k)$ using Eq. (13):

$$\begin{aligned} o_j(k) &= o_j(k-1) - \alpha(k-1)\beta_j(k-1)\widetilde{y}(k-1) \\ w_{ij}(k) &= w_{ij}(k-1) - \alpha(k-1)\gamma_{ij}(k-1)\widetilde{y}(k-1) \end{aligned} \tag{13}$$

   where the new time-varying rate $\alpha(k)$ is:

$$\alpha(k-1) = \frac{\alpha_0}{2\left(\frac{1}{2} + \sum_{j=1}^{M}\beta_j^2(k-1) + \sum_{j=1}^{M}\sum_{i=1}^{N}\gamma_{ij}^2(k-1)\right)}$$

   where $i = 1, \ldots, N$, $j = 1, \ldots, M$, $\gamma_{ij}(k-1) = o_j(k)\text{sech}^2\left(\sum_{i=1}^{N} w_{ij}(k-1)z_i(k-1)\right)z_i(k-1) \in \Re$.

Remark 4 There are two conditions for applying this algorithm for nonlinear systems: the first one is that the nonlinear system may have the form described by (10), and the second one is that the uncertainty $\mu(k) = y(k) - \sum_{j=1}^{M} o_j^*\beta_j^*$ may be bounded, $\beta_j^* = \tanh\left(\sum_{i=1}^{N} w_{ij}^* z_i(k)\right)$, o_j^* and w_{ij}^* are unknown weights such that the uncertainty $\mu(k)$ is minimized.

Remark 5 The value of the parameter used for the stability of the algorithm $\overline{\mu}$ is unimportant, because this parameter is not used in the algorithm. The bound of $\mu(k)$ is needed to guarantee the stability of the algorithm, but it is not used in the SGD algorithm (11), (12), (13).

Remark 6 The proposed SGD has one hidden layer. It was reported in the literature that a feedforward neural network with one hidden layer is enough to approximate any nonlinear system.

Remark 7 Note that the behavior of the algorithm could be improved or deteriorated by changing the values of M or α_0.

2.3 MSAFIS

The modified sequential adaptive fuzzy inference system (MSAFIS) is the SAFIS algorithm with the modification of Eqs. (3) and (8) by Eqs. (12), (13), and using the parameters of the SAFIS algorithm $m_j(k)$, $\delta_j(k)$, $o_j(k)$ instead of the parameters of the SGD algorithm $w_{ij}(k)$, $o_j(k)$. The MSAFIS algorithm is summarized as follows.

For each observation $(z(k), y(k))$ where $z(k) \in \Re^N$, $y(k) \in \Re$ and $k = 1, 2, \ldots,$ do

1. Compute the overall system output:

$$\widehat{y}(k) = \frac{\sum_{j=1}^{M} o_j(k) R_j(z_i(k))}{\sum_{j=1}^{M} R_j(z_i(k))} \tag{14}$$

where

$$R_j(z_i(k)) = \exp\left(-\frac{1}{\delta_j^2(k)} \left\| z_i(k) - m_j(k) \right\|^2\right)$$

and M is the number of fuzzy rules, $R_j(z_i(k))$ is the firing strength of the jth rule, $o_j(k)$ is the weight of the normalized rule. Note that each rule is represented as a radial basis function described by its center $m_j(k)$ and its spread $\delta_j(k)$.

2. Calculate the parameters required in the growth criterion:

$$\epsilon(k) = \max\left\{\epsilon_{\max}\tau^k, \epsilon_{\min}\right\}, \quad 0 < \tau < 1, \tag{15}$$

where $\epsilon_{\max}$ and $\epsilon_{\min}$ are the threshold largest and smallest distances admitted between the inputs and corresponding nearest center of rules. The parameter $\tau (0 < \tau < 1)$ indicates the decay constant. The error of the kth input is given as follows:

$$\widetilde{y}(k) = \widehat{y}(k) - y(k). \tag{16}$$

3. Apply the criterion for adding rules, if the following two conditions are satisfied: if

$$\left\| z_i(k) - m_j(k) \right\| > \epsilon(k) \tag{17}$$

and

$$Y_{\inf}(M+1) = |\widetilde{y}(k)| \frac{\left(1.8K \left\| z_i(k) - m_j(k) \right\|\right)^N}{\sum_{j=1}^{M+1} (1.8\delta_j(k))^N} > y_g \tag{18}$$

where y_g is the growing threshold. A new rule $M+1$ is added if y_g is exceeded.
The new rule $M+1$ is given as follows:

$$o_{M+1}(k) = \widetilde{y}(k)$$
$$m_{M+1}(k) = z_i(k)$$
$$\delta_{M+1}(k) = K \left\| z_i(k) - m_{M+1}(k) \right\| \tag{19}$$

If no rule is added, the nearest rule jm is obtained as follows:

$$\min_j R_j(z(k)) \Longrightarrow jm = j \tag{20}$$

and adjust the system parameters $o_j(k), m_j(k), \delta_j(k)$ for the nearest rule only by using the stable gradient descent algorithm:

$$\varphi(k) = \varphi(k-1) - \alpha(k-1) b(k-1) \widetilde{y}(k-1) \tag{21}$$

where $\varphi(k)=[\varphi_1(k),\ \varphi_2(k),\ \varphi_3(k)]^T=[m_{jm}(k),\ o_{jm}(k),\ \delta_{jm}(k)]^T$, $b(k)=[b_1(k),\ b_2(k),\ b_3(k)]^T$, $b_1(k)=\frac{2[o_{jm}(k)-\widehat{y}(k)]R_{jm}(z_i(k))[z_i(k)-m_{jm}(k)]}{\left[\sum_{j=1}^{M} R_j(z_i(k))\right]\delta_{jm}^2(k)}$, $b_2(k)=\frac{2[o_{jm}(k)-\widehat{y}(k)]R_{jm}(z_i(k))\|z_i(k)-m_{jm}(k)\|^2}{\left[\sum_{j=1}^{M} R_j(z_i(k))\right]\delta_{jm}^3(k)}$, $b_3(k)=\frac{R_{jm}(z_i(k))}{\left[\sum_{j=1}^{M} R_j(z_i(k))\right]}$, the new time-varying rate $\alpha(k-1)$ is:

$$\alpha(k-1) = \frac{\alpha_0}{2\left(\frac{1}{2} + \sum_{l=1}^{3} b_l^2(k-1)\right)}$$

where α_0 is a parameter selected by the designer, $0 < \alpha_0 < 1$.

4. If the following criterion is satisfied: If

$$Y_{\inf}(jm) = \left|o_{jm}(k)\right| \frac{\left(1.8\delta_{jm}(k)\right)^N}{\sum_{j=1}^{M} (1.8\delta_j(k))^N} < y_p \tag{22}$$

then remove the jm rule and reduce the dimensionality of SGD. Note that y_p is the pruning threshold.

Remark 8 In MSAFIS, some parameters need to be decided in advance according to the problems considered. They include the distance thresholds ($\epsilon_{\max}$, $\epsilon_{\min}$, τ), the overlap factor K for determining the width of the newly added rule, the growing threshold (y_g) for a new rule and the pruning threshold (y_p) for removing an insignificant rule. A general selection procedure for the predefined parameters is given as follows: max is set to around the upper bound of input variables; $\epsilon_{\min}$ is set to around 10 % of $\epsilon_{\max}$; τ is set to around 0.99. y_p is set to around 10 % of y_g. $\epsilon_{\max}$ is observed in the range [1.0, 10.0]. The overlap factor K is utilized to initialize the width of the newly added rule and chosen according to different problems, it is observed in the range [1.0, 2.0].

Table 1 Characteristics of the three algorithms

SAFIS	SGD	MSAFIS
If it is applied to systems which have important changes through the time, an acceptable result can be assured	If it is applied to systems which have important changes through the time, an acceptable result cannot be assured	If it is applied to systems which have important changes through the time, an acceptable result can be assured
If it is applied to unstable systems, an acceptable result cannot be assured	If it is applied to unstable systems, an acceptable result can be assured	If it is applied to unstable systems, an acceptable result can be assured
It can be applied many systems as are the biology, mechatronic, mechanic, thermal, robotic, economic, etc	It can be applied many systems as are the biology, mechatronic, mechanic, thermal, robotic, economic, etc	It can be applied many systems as are the biology, mechatronic, mechanic, thermal, robotic, economic, etc

Table 2 Differences between the SAFIS and MSAFIS

SAFIS	MSAFIS
Equation (3): the error is obtained by subtracting the estimated output to the output	Equation (16): the error is obtained by subtracting the output to the estimated output
Equation (8): the parameters are adjusted using the extended Kalman filter algorithm	Equation (21): the parameters are adjusted using the stable gradient descent algorithm

The growing threshold y_g is chosen according to the system performance, it is observed in the range [0.001, 0.05] . The smaller the y_g, the better the system performance, but the resulting system structure is more complex.

2.4 Comparison of the three algorithms

In this subsection, the comparison between the three algorithms is described.

Table 1 shows several aspects about the three algorithms.

Table 2 shows an overview of the modifications made to the SAFIS to evolve the new method, called MSAFIS.

Note that the SGD is not included in Table 2, because it is more different than the other two algorithms.

3 The brain and eye signals

This section describes the characteristics of the brain and eye signals.

3.1 The EEG signals

The difference of the potential in one membrane is obtained by the exchange between the ions (Na+,Cl-,K+) being in the same. The rules have a potential difference's between the inside and outside which is called rest potential, this potential represents constant changes because of the impulses given by the neighbor rules. This potential difference's can be measured in the brain cortex using electrodes which convert the ion flow in electric flow. The characteristic of the encephalography signal (EEG) is of 5–300 μV in amplitude and of 0–150 Hz in frequency (Rubio et al. 2013).

The EEG signals are waves similar to periodic but the waves can change from one time to other, and they have some characteristics which allow the learning, as are the amplitude, the frequency, the morphology, the band, the rhythm, and the duration (Rubio et al. 2013).

The following paragraphs show the characteristics which are considered for an adult in vigilance (Rubio et al. 2013).

Alpha signal Is the normal rhythm of the bottom, is the most stable and typical in the human. It is found in the frequencies of 8–12 Hz ± 1 Hz. The amplitude is between 20 and 60 μV. It can be seen generally in posterior regions with more amplitude in the occipital lobes (see Fig. 1). It is more evident when the patient is awake with closed eyes and in physical and mental rest, it is stopped when the eyes are opened or with the mental activity.

Beta signal It is found in the frequencies >13 Hz, in general between 14 and 35 Hz. The amplitude is usually low, from 5 to 10 μV and is symmetric (see Fig. 1).

Theta signal It has a frequency of 4–8 Hz; is of half of low voltage, and is found in the temporal regions (see Fig. 1).

Delta signal It is found in the second and the third stages of the dream. It has a frequency of 0.5–3.5 Hz and the amplitude is generally higher than 75 μV (see Fig. 1).

3.2 The EOG signals

The electro-oculograms (EOG) are the signals obtained as a result of the eye movements of a patient and these EOG are detected using three electrodes, one electrode on the temple, one above and other underneath of the eye. Usually, the

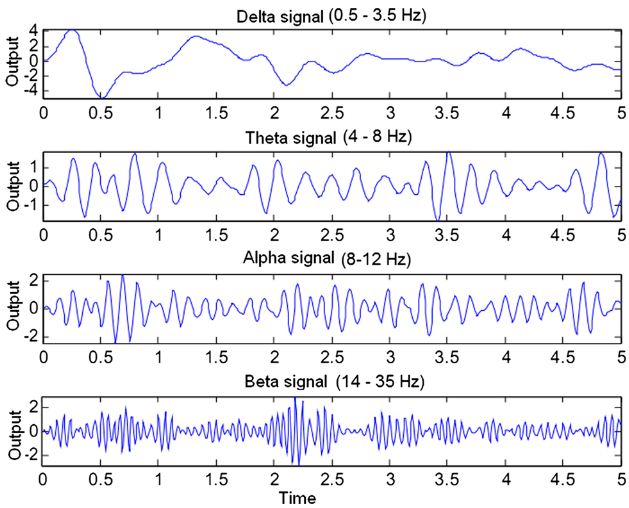


Fig. 1 EEG signals

detected signals are by direct current (DC) coupling to specify the direction of the gaze. In the experiments of this paper, three electrodes are placed on the dominant side of the patient eye according to the optimum positions suggested by Rubio et al. (2013).

Figure 2 shows the relationship between real eye movements (input) and the EOG signals (output) of the system. Denoted the upper and lower thresholds of the vertical channel Ch.V as V1 and V2, respectively, and denote the upper and lower thresholds of the horizontal channel Ch.H as H1 and H2, respectively. When the EOG potential exceeds one of these thresholds, the output assumes ON, and when the EOG potential does not exceed one of these thresholds, the output assumes OFF . The process of transforming the EOG signals from the intention of the patient is as follows (Rubio et al. 2013):

1. Output Up is when it is obtained an Up behavior, first, Threshold V1 of the vertical channel becomes ON while Threshold V2 is OFF, second, Threshold V2 of the vertical channel becomes ON while Threshold V1 becomes OFF. H1 and H2 of the horizontal channel remain OFF all the time.
2. Output Down is when it is obtained a Down behavior, first, Threshold V2 of the vertical channel becomes ON while Threshold V1 is OFF, second, Threshold V1 of the vertical channel becomes ON while Threshold V2 becomes OFF. H1 and H2 of the horizontal channel remain OFF all the time.

4 Results

In this section, the three above detailed algorithms are applied for the learning of brain and eye signals with big data. The aforementioned signals could be applied for patient who cannot move their bodies; consequently, they could use their brains or their eyes to say what they want or need. The SAFIS of Rong et al. (2006), SGD of Rubio et al. (2011), and MSAFIS are compared for the learning sequentially

- Brain signals: experiment 1,
- Eye signals: experiment 2.

The training of the learning phase, the parameters of the algorithms are incrementally learned as data are presented, while in the testing phase such parameters do not change and hence the algorithms can be compared in terms of performance.

The root mean square error (RMSE) of Rubio et al. (2011, 2013) is used to measure the performance and is expressed as:

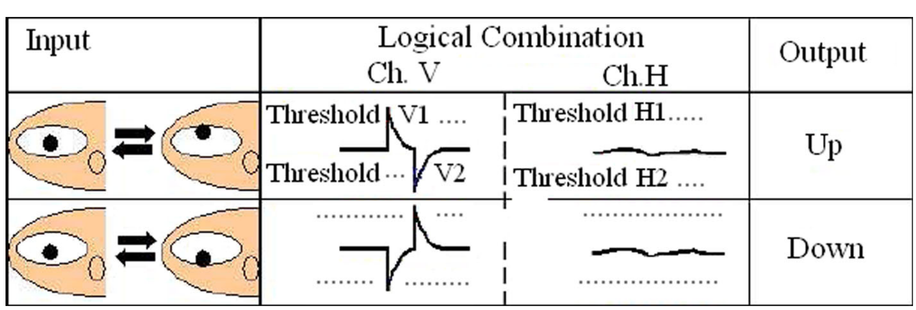
$$RMSE = \left(\frac{1}{N} \sum_{k=1}^{N} \widetilde{y}^2(k) \right)^{\frac{1}{2}} \tag{23}$$

where $\widetilde{y}(k)$ is the learning error expressed by Eqs. (3), (12), and (16).

4.1 Experiment 1

Here a real dataset of brain signals consisting of 20,000 pairs $(u(k), y(k))$ of 20 s are used to train the training, 2000 pairs $(u(k), y(k))$ for 2 s are used to test the learning. The alpha

Fig. 2 EOG signals



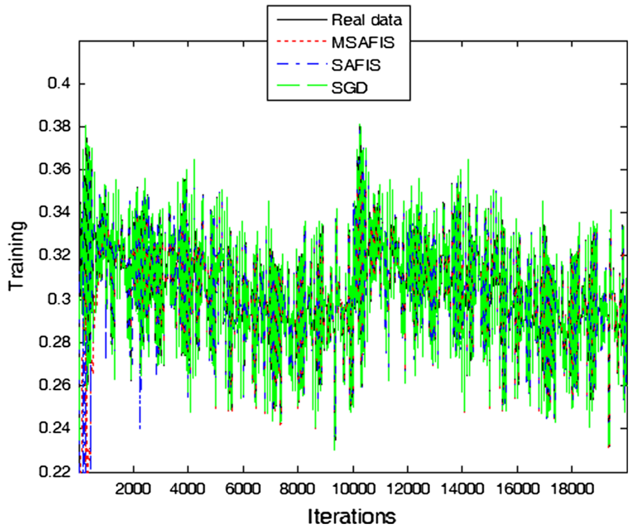


Fig. 3 Training for experiment 1

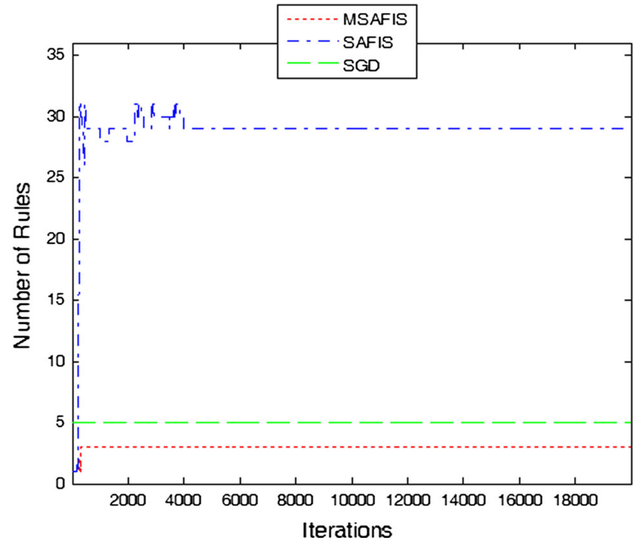


Fig. 4 Rule evolution for experiment 1

signal is obtained in this study because it has more probabilities to be found. The acquisition system is applied with a 28-year-old healthy man when his eyes are closed. The inputs of all the intelligent systems are $y(k)$, $y(k+1)$, $y(k+2)$, $y(k+3)$, and the output of the intelligent systems is $y(k+4)$. Considering Remark 3, the parameters for the SAFIS algorithm (Rong et al. 2006) are $N = 4$, $\tau = 0.99$, $K = 2$, $\epsilon_{\max} = 1$, $\epsilon_{\min} = 0.1$, $y_g = 0.01$, $y_p = 0.001$, $q = 0.1$, $p = 0.1$. Considering Remark 7, the parameters of the SGD algorithm of Rubio et al. (2011) are $N = 4, M = 5, \alpha_0 = 0.5$. Considering Remark 8, the parameters of the MSAFIS are $N = 4$, $\tau = 0.99$, $K = 2$, $\epsilon_{\max} = 2$, $\epsilon_{\min} = 0.2$, $y_g = 0.05$, $y_p = 0.005$, $\alpha_0 = 1$.

Figure 3 shows the comparison results for the training of learning in the three algorithms. Figure 4 introduces the illustration of the rule evolution for the three algorithms during training. Figure 5 presents the comparison results for the testing of learning in the three algorithms. Table 3 shows the RMSE comparison results for the algorithms using (23).

From Figs. 3, 4, 5, and Table 3, it can be seen that the SGD presents the smallest training RMSE, the MSAFIS presents the smallest testing RMSE, and the MSAFIS obtains the smallest number of rules.

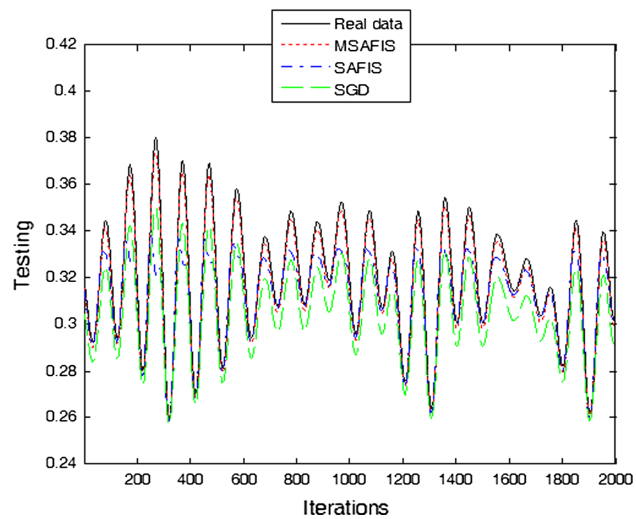


Fig. 5 Testing for experiment 1

Table 3 Results for experiment 1

Methods	Rules	Training RMSE	Testing RMSE
SGD	5	0.0043	0.0217
SAFIS	29	0.0145	0.0177
MSAFIS	3	0.0331	0.0045

4.2 Experiment 2

Here a dataset of eye signals of the down behavior is considered where 3572 pairs $(u(k), y(k))$ of 3.572 s are used to train the learning, 1192 pairs $(u(k), y(k))$ for 1.192 s are used to test the learning. The acquisition system is applied with a 25-year-old healthy man when his eyes are moving and two electrodes are used to find the signals as described in the aforementioned section. The inputs of all the intelligent systems are $y(k)$, $y(k+1)$, $y(k+2)$, $y(k+3)$, and the output of the intelligent systems is $y(k+4)$.

Considering Remark 3, the parameters for the SAFIS (Rong et al. 2006) are $N = 4$, $\tau = 0.986$, $K = 2$, $\epsilon_{\max} = 2$, $\epsilon_{\min} = 0.2$, $y_g = 0.01$, $y_p = 0.001$, $q = 0.1$, $p = 0.1$. Considering Remark 7, the parameters of the SGD (Rubio et al. 2011) are $N = 4$, $M = 9$, $\alpha_0 = 0.5$. Considering Remark 8, the parameters of the MSAFIS are $N = 4$, $\tau = 0.986$,

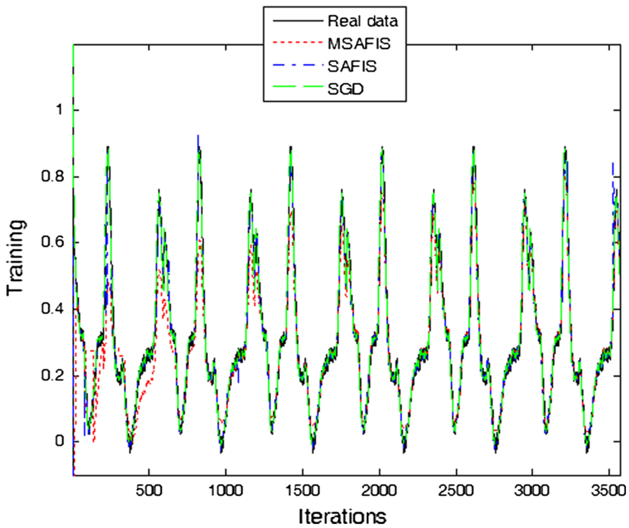


Fig. 6 Training for experiment 2

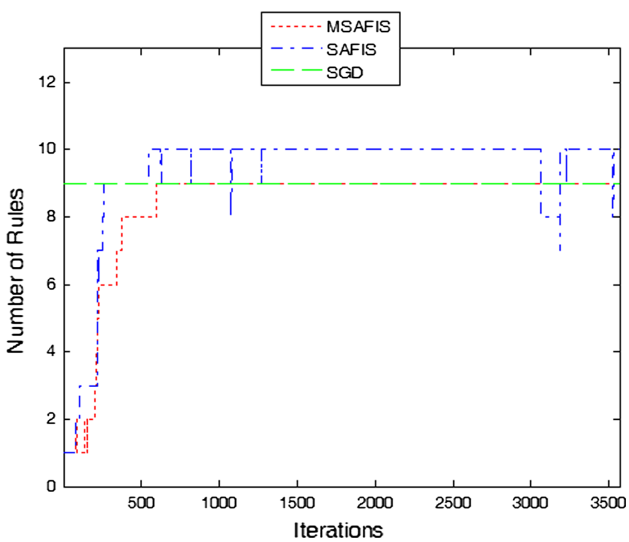


Fig. 7 Rule evolution for experiment 2

$K = 2$, $\epsilon_{\max} = 2$, $\epsilon_{\min} = 0.2$, $y_g = 0.01$, $y_p = 0.001$, $\alpha_0 = 1$.

Figure 6 shows the comparison results for the training of learning in the three algorithms. Figure 7 introduces the illustration of the rule evolution for the three algorithms during training. Figure 8 presents the comparison results for the testing of learning in the three algorithms. Table 4 shows the RMSE comparison results for the algorithms using (23).

From Figs. 6, 7, 8, and Table 4, it can be seen that the SGD presents the smallest training RMSE, the MSAFIS presents the smallest testing RMSE, and the MSAFIS and SGD obtain the smallest number of rules.

Remark 9 The SAFIS algorithm is applied in two synthetic examples and in the Mackey–Glass time series prediction problem (Rong et al. 2006). The SGD algorithm is applied in a synthetic example and in the prediction of the loads distribution in a warehouse (Rubio et al. 2011). This study is novel, because it shows that the three algorithms can be used for the learning of other different kinds of systems which are the real brain and eye signals with big data.

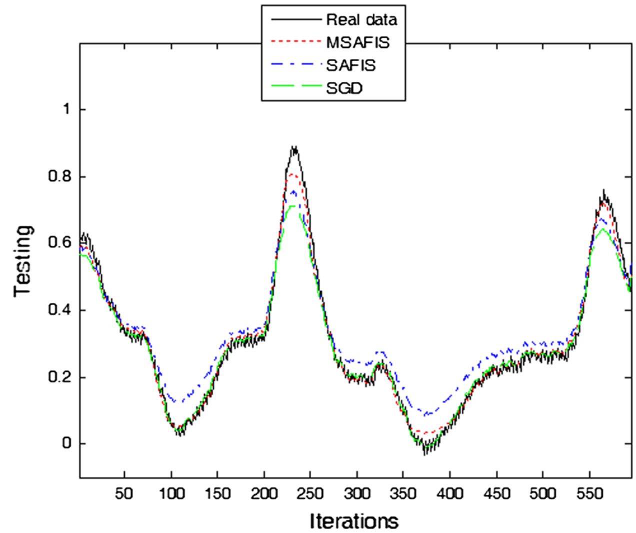


Fig. 8 Testing for experiment 2

Table 4 Results for experiment 2

Methods	Rules	Training RMSE	Testing RMSE
SGD	9	0.0252	0.0290
SAFIS	10	0.0263	0.0404
MSAFIS	9	0.0706	0.0172

5 Conclusion

This study proposed a combination of two algorithms SAFIS and SGD resulting in MSAFIS. Considering the different experiments, this new algorithm provides better compactness and higher accuracy compared to the original ones. It is worthwhile to mention, because as MSAFIS as well as SAFIS and SGD are based on online learning, they can handle big datasets of any size. They can also be applied to control, prediction, classification, and diagnosis. Here they were successfully used to learn from a challenging dataset of brain and eye signals. As a future work, the stability of the MSAFIS will be analyzed.

Acknowledgements The authors are grateful to the editors and the reviewers for their valuable comments. The first author thanks the Secretaría de Investigación y Posgrado, Comisión de Operación y Fomento de Actividades Académicas, and Consejo Nacional de Ciencia y Tecnología for their help in this research.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Ahn CK (2012) An error passivation approach to filtering for switched neural networks with noise disturbance. Neural Comput Appl 21(5):853–861

Ahn CK (2014) A new solution to the induced l∞ finite impulse response filtering problem based on two matrix inequalities. Int J Control 87(2):404–409

Ahn CK, Lim MT (2013) Model predictive stabilizer for T-S fuzzy recurrent multilayer neural network models with general terminal weighting matrix. Neural Comput Appl 23(Suppl 1):S271–S277

Angelov P, Filev D, Kasabov N (2010) Evolving intelligent systems—methodology and applications. Wiley, New York

Bordignon F, Gomide F (2014) Uninorm based evolving neural networks and approximation capabilities. Neurocomputing 127:13–20

Bouchachia A (2008) Incremental Learning. Encyclopedia of Data Warehousing and Mining, pp 1006–1012

Bouchachia A (2014) Online dataprocessing. Neurocomputing 126:116–117

Bouchachia A, Lena A, Vanaret C (2014) Online and interactive self-adaptive learning of user profile using incremental evolutionary algorithms. Evol Syst 5:143–157

Bouchachia A, Vanaret C (2014) GT2FC: an online growing interval type-2 self-learning fuzzy classifier. IEEE Trans Fuzzy Syst 22(4):999–1018

Gama J (2010) Knowledge discovery from data streams. Chapman & Hall/CRC, Boca Raton

Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM Comput Surv 46(4):44

Garcia-Cuesta E, Iglesias JA (2012) User modeling: Through statistical analysis and subspace learning. Expert Syst Appl 39:5243–5250

Gomide F, Lughofer E (2014) Recent advances on evolving intelligent systems and applications. Evolv Syst 5:217–218

Hartert L, Sayed-Mouchaweh M (2014) Dynamic supervised classification method for online monitoring in non-stationary environments. Neurocomputing 126:118–131

Huang G-B, Saratchandran P, Sundararajan N (2004) An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks. IEEE Trans Syst Man Cybern Part B Cybern 34(6):2284–2292

Iglesias JA, Tiemblo A, Ledezma A, Sanchis A (2015) Web news mining in an evolving framework. Inf Fusion. doi:10.1016/j.inffus.2015.07.004

Iglesias JA, Ledezma A, Sanchis A (2014) Evolving classification of UNIX users' behaviors. Evolv Syst 5:231–238

Iglesias JA, Skrjanc I (2014) Applications, results and future direction. Evolv Syst 5:1–2

Kasabov N (2007) Evolving connectionist systems: the knowledge engineering approach, 2nd edn. Springer Verlag, London

Klancar G, Skrjanc I (2015) Evolving principal component clustering with a low run-timecomplexity for LRF data mapping. Appl Soft Comput 35:349–358

Lughofer E (2011) Evolving fuzzy systems—methodologies, advanced concepts and applications. Springer, Berlin, Heidelberg

Lughofer E, Cernuda C, Kindermann S, Pratama M (2015) Generalized smart evolving fuzzy systems. Evolv Syst. doi:10.1007/s12530-015-9132-6

Lughofer E, Sayed-Mouchaweh M (2015) Autonomous data stream clustering implementing split-and-merge concepts—towards a plug-and-play approach. Inf Sci 304:54–79

Lughofer E, Sayed-Mouchaweh M (2015) Adaptive and on-line learning in non-stationary environments. Evol Syst 6:75–77

Marques Silva A, Caminhas W, Lemos A, Gomide F (2014) A fast learning algorithm for evolving neo-fuzzy neuron. Appl Soft Comput 14:194–209

Ordoñez FJ, Iglesias JA, de Toledo P, Ledezma A, Sanchis A (2013) Online activity recognition using evolving classifiers. Expert Syst Appl 40:1248–1255

Perez-Cruz JH, Rubio JJ, Pacheco J, Soriano E (2014) State estimation in MIMO nonlinear systems subject to unknown deadzones using recurrent neural networks. Neural Comput Appl 25(3–4):693–701

Perez-Cruz JH, Rubio JJ, Encinas R, Balcazar R (2014) Singularity-free neural control for the exponential trajectory tracking in multiple-input uncertain systems with unknown deadzone nonlinearities. The Scientific World Journal 2014:1–10

Pratama M, Anavatti SG, Er MJ, Lughofer ED (2015) pClass: an effective classifier for streaming examples. IEEE Trans Fuzzy Syst 23(2):369–386

Precup R-E, Sabau M-C, Petriu EM (2015) Nature-inspired optimal tuning of input membership functions ofTakagi-Sugeno-Kang fuzzy models for Anti-lock Braking Systems. Appl Soft Comput 27:575–589

Rong HJ, Sundararajan N, Huang GB, Saratchandran P (2006) Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction. Fuzzy Sets Syst 157(9):1260–1275

Rubio JJ, Angelov P, Pacheco J (2011) An uniformly stable backpropagation algorithm to train a feedforward neural network. IEEE Trans Neural Netw 22(3):356–366

Rubio JJ, Ortiz F, Mariaca CR, Tovar JC (2013) A method for online pattern recognition for abnormal eye movements. Neural Comput Appl 22(3–4):597–605

Rubio JJ, Vazquez DM, Mujica-Vargas D (2013) Acquisition system and approximation of brain signals. IET Sci Meas Technol 7(4):232–239

Sayed-Mouchaweh M, Lughofer E (2015) Decentralized fault diagnosis approach without a global model for fault diagnosis of discrete event systems. Int J Control. doi:10.1080/00207179.2015.1039594

Sayed-Mouchaweh M, Lughofer E (2012) Learning in non-stationary environments: methods and applications. Springer, New York

Torres C, Rubio JJ, Aguilar-Ibañez C, Perez-Cruz JH (2014) Stable optimal control applied to a cylindrical robotic arm. Neural Comput Appl 24(3–4):937–944

Zdesar A, Dovzan D, Skrjanc I (2014) Self-tuning of 2 DOF control based on evolving fuzzy model. Appl Soft Comput 19:403–418