Evolving Intelligent Systems

Selected Topics of Evolving Intelligent Systems
José de Jesús Rubio, Manuel Jimenez, Humberto Perez, Maricela Figueroa

Abstract
In this paper, some interesting characteristics of evolving intelligent systems are described. They are known as: the definition of the evolving intelligent systems, the applications, evolving pattern classification, evolving control, evolving identification, evolving prediction, and evolving optimization. For the details of the described methods, please see the references.

1. Introduction
There is some interesting research about evolving systems as is [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], and [12].

The paper of [2] introduces a hybrid evolving architecture for dealing with incremental learning, consisting of two sequential and incremental learning modules: growing Gaussian mixture model (GGMM) and resource allocating neural network, (RAN), the rationale of the architecture rest on two issues: incrementally and the possibility of partially labeled data processing in the context of classification. The paper of [3] presents an online self-evolving fuzzy controller with global learning capabilities, starting from very simple, even empty configurations, the controller learns from it own actions while controlling the plant. The paper of [4] proposes the application of evolving fuzzy modeling to fault tolerant in two steps: fault detection and fault accommodation, fault accommodation uses evolving Takagi-Sugeno fuzzy models, and fault detection uses a model-based approach also based on fuzzy models. In [5], an online Takagi Sugeno fuzzy model is presented, the presented method combines the recursive Gsatfson-Kessel clustering algorithm and the fuzzy recursive least square method. The article of [6] proposes knowledge-based short time prediction methods for multivariate streaming time series that rely on the early recognition of such local patterns a parametric fuzzy model for patterns is presented, along with an online, classification based recognition procedure, which will introduce the notion of evolving classification results. In [7], they propose a new incremental linear discriminant analysis (ILDA) for multitask pattern recognition (MTPR) problems in which a chunk of training data for a particular task are given sequentially is switched to another related task one after another. In [8], they present a general approach to the classification of streaming data which represent a specific agent behavior based on evolving systems, their approach can efficiently model and recognize different behaviors. In [9], they propose a new online predictor model for complex nonlinear processes, while the developed model can be as complex as a TS fuzzy model with exible precedents, it habitually tends to shrink to an adaptive linear model. The paper of [10] introduces a new approach for evolving fuzzy modeling using tree structures, the model is a fuzzy linear regression tree whose topology can be continuously updated through a statistical model selection test. In [11], they revisit the underlaying concept and identify an essential optimization problems arising therein. In [12], a stable backpropagation algorithm is used to train an evolving radial basis function neural network, is generates groups with an online clustering, the centers are updated to achieve they are near the incoming data each iteration, so it does not need to prune the neurons.

From the afore mentioned papers, [2], [6], [7], and [8] work with evolving pattern classification, [3] and [4] work with evolving control, [5], [10], and [12], work with evolving identification, [9] works with evolving prediction, and [11] works with evolving optimization.

In this paper, some interesting characteristics of evolving intelligent systems are described. They are known as: the definition of the evolving intelligent systems, the applications, evolving pattern classification, evolving control, evolving identification, evolving prediction, and evolving optimization. For the details of the described methods, please see the references.
2. EVOLVING INTELLIGENT SYSTEMS

The evolving intelligent systems, are characterized by habilities to adjust their structure as well as parameters to the varying characteristics of the environment (with the term of environment embracing processes/phenomena with wich the system has to interact and or deal with the users using the system) [11].

The neural networks born in the eighties, the philosopy of the neural networks is to simulate the behavior of the brain, now the evolving intelligent systems simulate better the brain than the neural networks because, the neural networks only train their parameters while the evolving intelligent systems train their structure and their parameters. Please see Figure 1.

![Fig. 1. Evolving Intelligent Systems vs Neural Networks](image)

The emerging area of the evolving intelligent systems was conceived around the areas of neural networks, fuzy-rule based systems, and neuro-fuzzy hybrids [1].

It is currently being expanded also to the areas of general systems, control hardware, implementarions, etc. Numerous interesting applications of such systems to robotic, au-tonomous unmanned systems, vehicle systems, process monitoring & control, bio-medical data processing, etc., have been reported [1]. Please see Figure 2.

![Fig. 2. Applications of the evolving intelligent systems](image)

3. EVOLVING PATTERN RECOGNITION

In [2] he presents the growing Gaussian mixture model(GGMM). The first module of the proposed architecture is the growing mixture models used to a self-learn the labels of the unlabeled data. The GGMM can be seen as a clustering method that is model oriented. In essence, the model-based clustering method perceives the data as a population with K different components, where each component is generated by an underlaying probability distribution. The density of an m-dimensional data point \( x_i \) from the jth component is \( f(x_i; \theta_j) \), where \( \theta_j \) represents a vector of some unknown parameters associated with the component j. The density function associated is expressed as:

\[
f(x_i | \theta) = \sum_{j=1}^{K} \tau_j f_j(x_i; \theta_j)
\]

where \( \tau_j \) is the weight of the jth component such that:
A typical case is when $f_j(x_i; \theta_j)$ is multivariate normal (Gaussian) density $\phi_j$ with $\theta_j$ representing the characteristics of each component $j$ and standing for the mean $\mu_j$ and the covariance matrix, $\sum_j$. The density $\phi_j$ is described as follows:

$$
\phi_j(x_i; \theta_j) = \exp \left\{ -\frac{1}{2} (x_i - \mu_j)^T \sum_j^{-1} (x_i - \mu_j) \right\} \frac{1}{\sqrt{\text{det}(2\pi\sum_j)}}
$$

Gaussian mixtures are usually trained using the iterative Expectation Maximization (EM) algorithm. The existing approaches can be split into two categories, he names them as: re nement based methods and learning methods.

In [2] he presents the minimal resource allocating network (MRAN). The second module of the proposed architecture consists of a RAN which is mainly responsible for the class- si cation. Originally proposed to approximate functions, it is able according to several studies to learn non-stationary data. RAN consists of three layers: input layer, hidden layer, and output layer. Each output node will be computed as follows:

$$
y_i = w_0 + \sum_{j=1}^{H} w_{ij}\phi_j(x)
$$

where $w_0$ is a bias and $\phi_j(x)$ is a Gaussian function associated with the $j$th hidden unit which is defined by the mean $\mu_j$ and the standard deviation $\sigma_j$. $\phi_j(x)$ is defined as:

$$
\phi_j(x) = \exp \left( -\frac{1}{\sigma_j^2} \left\| x - \mu_j \right\|^2 \right), j = 1, \ldots, H
$$

where $\left\| \cdot \right\|$ denoted the Euclidean norm.

In [6], they formulate a multivariate case extension to a fuzzy set $\mu$ defined in a $N$-dimensional space of features $x_i; i = 1; \ldots; N$, an intersection of $N$ normalized fuzzy sets of this type is being performed and assigned one joint maximum truth value $a$, which will again occur at the modal point $x = r$ of this set:

$$
\mu(x) = a(\mu_1(x_1) \cap \ldots \cap \mu_N(x_N))
$$

with

$$
x = (x_1, \ldots, x_N)^T \in \mathbb{R}^N
$$

For high-dimensional fuzzy sets, i.e. larger number of $N$, $\mu(x)$ would too quickly decrease to zero when moving away from the modal point if T-norm intersection operators would be used. The $N$-fold intersection of $N$ truth values $\mu_i, i = 1; \ldots; N$, is defined as:

$$
\bigcap_{i=1}^{N} \text{Hamacher} \mu_i = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\mu_i}
$$

Applying this Hamacher intersection operator (6) to (5) yields:

$$
\mu(x) = aN \left( \sum_{i=1}^{N} \frac{1}{\mu_i(x_i)} \right)^{-1}
$$

which, if we insert the normalized univariate fuzzy sets for each $\mu_i(x_i)$ in rewritten form as given:

$$
\mu_i(x_i) = \left\{ 1 + \frac{1}{2} \left[ 1 - \text{sgn}(x_i - r_i) \right] \left( \frac{1}{b_i} - 1 \right) \left| \frac{x_i - r_i}{c_i} \right| \right\}
$$

$$
+ \frac{1}{2} \left[ 1 + \text{sgn}(x_i - r_i) \left( \frac{1}{b_i} - 1 \right) \left| \frac{x_i - r_i}{c_i} \right| \right]
$$

In [7], they uses the incremental linear discriminant analysis (ILDA). They assume that there are $N$ training data which belong, to either of $C$ classes, the set of class $c$ data is denoted as:


03/12/2011
where $x_{cj}$ is the $j$th data of class $c$ and $n_c$ is the number of class $c$ data.

The whole set of training data is denoted as:

$$X = \{X_c\}_{c=1}^{C}$$

For $X$, they can define the between-class scatter matrix $S_B$ and the within class scatter matrix $S_W$:

$$S_B = \sum_{c=1}^{C} n_c (\bar{x}_c - \bar{x})(\bar{x}_c - \bar{x})^T$$

(9)

$$S_W = \sum_{c=1}^{C} V_C$$

(10)

where

$$V_C = \sum_{j=1}^{nc} (\bar{x}_{cj} - \bar{x}_c)(\bar{x}_{cj} - \bar{x}_c)^T$$

(11)

Here, $\bar{x}_c$ and $\bar{x}$ are the mean vector of class $c$ data and that of all data, respectively.

In LDA, a discriminant vector $w$ is obtained by maximizing the following class separability:

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

(12)

The following objective function has to be maximized:

$$J(w) = tr(W^T S_W^{-1} S_B W)$$

(13)

It is well known that $W$ is computed by solving the following generalized eigenvalue problem:

$$S_B W = S_W W \Lambda$$

(14)

where $\Lambda$ is a diagonal matrix whose diagonal element $\lambda_i$ is the eigenvalue of $w$ in their paper, a discriminant space model is represented by the sextuplet:

$$\Omega = \{S_B, S_W, \Lambda, W, \{\bar{x}_c\}_{c=1}^{C}, \{n_c\}_{c=1}^{C}\}$$

(15)

In [8], they present an evolving classifier of behavior models module (EvCBM). The procedure of this classifier includes the following stage/steps: 1.- Classify a new sample (agent behavior) in a group represented by a prototype, 2.- Calculate the potential of the new data sample to be a prototype, 3.- Update all the prototypes considering the new data sample, 4.- Insert the new data sample as a new prototype if needed, 5.- Remove existing prototypes if needed.

### 4. EVOLVING CONTROL

In [3], an online self-evolving fuzzy controller (OSEFC) is presented. For instance, consider a single-input single-output plant, whose dynamics are given by:

$$y_{k+1} = f(x_k, u_k)$$

(16)

where $x_k = (y_k, y_{k-1}, \ldots, y_{k-p}; u_k, u_{k-1}, \ldots, u_{k-q})$ is the state of the plant, $u_k$ is the control signal exerted by the controller, $f$ is an unknown, continuous, and differentiable function, $p$ and $q$ are constants that determine the plant order.

The controller can be expressed as a function $G$ such that:

$$u_k = G(\tilde{x}_k; \Phi)$$

(17)

where $\tilde{x}_k = (y_k, y_{k-1}, \ldots, y_{k-p}; u_k, \ldots, u_{k-q})$ and

$\Phi$ represents the set of parameters that define the controller.

The selected fuzzy system uses the product as the Tnorm for the conjunction and the weighted average as the defuzzification strategy. Thus the output of the fuzzy system at instant $k$ is given by:
where $\Phi$ is the set of parameters defining the fuzzy controller at time $k$, $Q_i$ is a scalar value representing the rule consequent, and $\tilde{\alpha}_i(x_k)$ is the firing strength of the $i$th rule, which is calculated by:

$$\alpha_i(x_k) = \alpha_i(x_1, x_2, ..., x_N) = \prod_{j=1}^{N} \mu_j^i(x_j)$$

(19)

where $\mu_j^i(x_j)$ is the activation degree of the $j$th MF in the antecedent part of the rule $i$.

In [4], they propose control which uses a fuzzy model. The proposed fuzzy modeling uses the scatter approach and the potential approach.

Online learning of the ETS fuzzy models using the scatter approach entails the following stages: a) initialization of rule base structure (antecedent part of the rule), b) reading the next data sample at the next time step, c) recursive calculation of the scatter of the new data, d) recursively update the scatter at the focal point (rule center) of the existing rule, e) possible modification of the rule base upgrade based on the scatter of the new data in comparison to scatter of the existing rule, f) recursive calculation of the consequent parameters, g) prediction of the model output for the next time step.

In the potential approach, for the data that establish the focal point of the first cluster, and later it is updated recursively.

For the model predictive control using evolving fuzzy models the control signals change only inside the control horizon:

$$u(k + j) = u(k + Hc - 1)$$

(20)

for $j = Hc; ..., Hp - 1$. The sequence of future control signals is obtained by optimizing a cost function which describes the control goal, and is usually of the following form:

$$J = \sum_{i=1}^{HF} \alpha_i(\tilde{e}(k + i))^2 + \beta_i(\Delta \tilde{g}(k + i))^2$$

(21)

where $\tilde{e}(k + i)$ denotes the predicted errors given by the difference between the reference $r$ and the output of the system $y$:

$$\tilde{e}(k + i) = r(k + i) - \tilde{g}(k + i)$$

(22)

5.EVOLVING IDENTIFICATION

In [5] they present the recursive Gustafson-Kessel clustering algorithm. The algorithm is given as follows:

- Step 0: Initialization: a) define the number of clusters ($c$), the fuzziness and the overlapping, ($\eta, \eta_m, \eta_l$), the forgetting factors ($\lambda_r, \gamma_v, \gamma_c$), number of measured variables ($p$) and the cluster volumes ($\rho_i$), b) determine the initial value of $P_i$ for $i = 1; \ldots; c$, c) determine the initial $F_i$ for $i = 1; \ldots; c$, d) determine the initial centers: $v_i = x(k)$, for $i = k = 1; \ldots; c$ and membership degrees $\mu_i(x(k)) = 1$ for $i = k_i$, $\mu_i(x(k)) = 0$ for $i = k_i$, e) calculate the initial $s_i$ for $i = 1; \ldots; c$, f) calculate the inverse of the fuzzy covariance matrix.

- Step 1: calculate the matrix $A_i$
- Step 2: calculate the distance $D_i$
- Step 3: calculate the current membership degree
- Step 4: calculate the $s_i(r + 1)$
- Step 5: calculate the change of the cluster centers $\Delta v_i$
- Step 6: calculate the new centers
- Step 7: calculate the new fuzzy covariance matrix $F_i(r + 1)$ and the inverse fuzzy covariance matrix
- Step 8: calculate the membership functions $\mu_i$
- Step 9: calculate the membership degrees $\beta_i$
- Step 10: apply the recursive least squares

In [10], it is presented an algorithm to evolve fuzzy linear regression models which is as follows:

- 1. Compute the output and the membership value of all leaves
- 2. Update the linear models
3.- Select the leaf with the highest membership value
4.- for all inputs (m) do
5.- for all candidate splits (k) do
6.- Estimate the output replacing the selected leaf with the candidate split
7.- Compute the p-value of the model selection test for the candidate split
8.- end for
9.- end for
10.- Select the candidate split associated with the minimum p-value
11.- if p-value < α/(k×m) then
12.- Replace the selected leaf by the candidate split
13.- end if

In [12], they present an evolving radial basis neural net-work algorithm which is described as follows:

0) Select the following parameters: 0 < \( r < 1 \), 0 < \( \eta_0 < 1 \)
1) For the first data \( k = 1 \), \( M = 1 \), \( v_i^{(1)} = y^{(1)} \), \( i = 1; \ldots; N \), \( c_i^{(1)} = x_i^{(1)} \), \( \sigma_i^{(1)} = 1/rand \)
2) For other data \( k \geq 2 \), evaluate \( p(k-1) \), update \( c_{ij}^{(k)} \), \( \sigma_{ij}^{(k)} \) and \( v_j^{(k)} \)
3) If \( p(k-1) \geq r \), then a rule or neuron is not generated and in the case of \( z_j^{(k-1)} = p(k-1) \) we have the winner neuron \( c_{ij}^{(k)} \) which is updated, go to 2
4) if \( p(k-1) < r \), then a new neuron or rule is generated (\( M = M + 1 \)), assign the initial values to \( c_{iM+1}^{(k)} \), \( \sigma_{iM+1}^{(k)} \) and \( v_{M+1}^{(k)} \)

### 6. EVOLVING PREDICTION

In [9], they propose an adaptive habitually linear and tran-siently nonlinear model (AHLTNM). The algorithm is given as follows:

1.- Define the parameters of the first rule at \( t = t_0 \): \( w_{1t_0} = z_{(n+1)×1} \), \( P_{1t_0} = \lambda I_{(n+1)×(n+1)} \) where \( z \) and \( I \) denote the zero vector and an unit matrix, respectively and \( \lambda \) is a large positive constant which is used as the resetting factor
2.- for initial \( N_0 = 20n \) intake data points, \( t = 1 \) up to \( t = N_0 \) perform following tasks
   2.1.- estimate the output of linear model and absolute errors
   2.2.- update the parameters the RLS technique
   2.3.- compute \( e_0 \) as \( 1/20 \) for the mean value of absolute errors
3.- define the centers and lengths of the first hyper rectangle using the considered initial \( N_0 \) data points, also initialize adaptive thresholds as \( E_t = e_0 \)
4.- let \( t = t+1 \) and compute the output of the model
5.- update the parameters of the (lth one) with highest validation function at \( x_t \)
6.- update the adaptive error threshold, \( E_t \)
7.- if \( |e_t| \geq E_t \) and the number of rules is smaller than \( M_{max} \) the run split operation for lth rule
8.- if \( |e_t| < E_t \) and there are at least two rules, run merge operation for lth rule, and also make an intra merge for blocks of the new rule if it is possible
9.- Go to step 4

### 7. EVOLVING OPTIMIZATION

In [11], he presents the dynamic formation of fuzzy clusters splitting and merging mechanisms. He describes that the objective function minimized by the FCM for the 1st data snapshot exploits the standard sum of distances:

\[
Q[1] = \frac{1}{N} \sum_{i=1}^{c[1]} \sum_{k \in X_i^{[1]}} u_{ik}^{[1]} \| x_k - v_i^{[1]} \|^2
\]

Here \( Q[1] \) concerns the first data snapshot, let us also note that the number of clusters, say \( c[1] \) may be selected on a basis of the assumed threshold level, \( V_{max} \).

The formulation of the clustering splitting is given in the form:
min \sum_{i=1}^{2} \sum_{k \in X_i} f_{ik} ||x_k - z_i||^2 \quad (24)

subject to the constant:
\sum_{j=1}^{2} f_{jk} = u_{lok} \quad (25)

z_i are the prototypes and F = [f_{ik}] denotes the partition matrix that satisfies constraint (25). The detailed calculations of the partition matrix F and the two prototypes z_1 and z_2 are carried out iteratively according to the two expressions:

\hat{f}_{ik} = \frac{u_{lok}}{\sum_{j=1}^{2} \left( \frac{||x_k - z_j||}{||x_k - z_i||} \right)^{(m-1)}} \quad (26)

and

z_i = \frac{\sum_{k \in X_i} f^m_{ik} x_k}{\sum_{k \in X_i} f^m_{ik}}, i = 1, 2; \quad x_k \in X_i \quad (27)

For the cluster merging, the new prototype \( \hat{v} \) results form the minimization of the following performance index:

\[ Q = \sum_{i=1}^{N} (u_{lok} + \hat{u}_{lok})^m ||x_k - \hat{v}||^2 \rightarrow \text{Min} \ Q \text{ with respect to } \hat{v} \quad (28) \]

8. CONCLUSION

In this paper, some interesting characteristics of evolving intelligent systems were described. They are known as: the definition of the evolving intelligent systems, the applications, evolving pattern classification, evolving control, evolving identification, evolving prediction, and evolving optimization. For the details of the described methods, please see the references.

9. ACKNOWLEDGEMENTS

The authors are grateful with the Dr. Plamen Angelov for inviting us to write this paper. The authors thank the Secretaria de Investigacion y Posgrado del IPN, the Comision de Operacion y Fomento de Actividades Academicas del IPN, and the Consejo Nacional de Ciencia y Tecnologia for their help in this research.

10. REFERENCES


About the Authors

José de Jesús Rubio (M’08) was born in MÉxico City in 1979. He received the B.S. degree from the Instituto Politécnico Nacional in MÉxico in 2001. He received the M.S. and neural networks.in automatic control form the CINVESTAV IPN in MÉxico in 2004, and the Ph.D. in automatic control from the CINVES- TAV IPN in MÉxico in 2007. He was a full time professor of the Autonomous Metropolitan University - Mexico City from 2006 to 2008. Since 2008, he is a full time professor of the SecciÛn de Estudios de Posgrado e InvestigaciÛn - Instituto Politécnico Nacional Ñ ESIME Azcapotzalco. He has published 28 papers in International Journals, 8 chapters in International Books, and he has presented 27 papers in International Conferences with more than 100 citations. He is a member of the IEEE AFS Adaptive Fuzzy Systems. He is part of the editorial board of the journal Evolving Systems. He has been the tutor of 14 M.S. students and 10 B.S. students. His research interests are primarily focused on evolving intelligent systems, intelligent control, nonlin- ear control, adaptive control, sliding mode control, optimal control, neural-fuzzy systems, Kalman filter, least square, bounded ellipsoid, delayed systems, collisions detector, trajec- tory generator, pattern recognition, identi- cation, prediction, image processing, robotic, mechatronic, medic, automotive, alternative energy, signal processing, greenhouse, petroleum, incubators, warehouse, chemical reactor, mixing.

Manuel Jimenez-Lizarraga, received the BS degree in elec- trical engineering from Instituto Tecnologico de Culiacan, Mexico, and the MS and PhD degrees in Automatic Control from CINVESTAV-IPN Mexico in 1996, 2000 and 2006 respectively. He was postdoc fellow at the ECE Department of the Ohio State University, USA from 2008-2009. He is currently with the Faculty of Physical and Mathematical Sci- ences of Autonomous University of Nuevo Leon, Mexico. His research interests include differential games, robust, optimal and sliding mode control and applications.

J. H. PÈrez-Cruz received the diploma in electronic en- gineering from Oaxaca Institute of Technology, Mexico in 1999, the M.S degree from Toluca Institute of Technology, Mexico in 2004, and the Ph.D. degree from CINVESTAV, Mexico in 2008. He is currently Professor at Oaxaca Institute of Technology. His fields of interest are system identi- cation, control and neural networks.