

Architectures for Evolving Fuzzy Rule-based Classifiers

Plamen Angelov
Dept of Communication
Systems, InfoLab21
Lancaster University
Lancaster LA1 4WA, UK

Xiaowei Zhou
Dept of Communication
Systems, InfoLab21
Lancaster University
Lancaster LA1 4WA, UK

Dimitar Filev
Ford Research and
Advanced Engineering,
Ford Motor Co
Detroit, MI 48239, USA

Edwin Lughofer
Dept of Knowledge-
based Mathematical Syst
J. Kepler University
A-4040, Linz, AUSTRIA

Abstract— In this paper the recently introduced evolving fuzzy classifier method called *eClass* is studied in respect to its architecture and evolution of the fuzzy rule-base. The proposed classifier has an open/evolving structure and can start ‘from scratch’, learning and adapting to the new data samples. Alternatively, if an initial fuzzy rule-based classifier, generated beforehand in off-line mode or provided by the operator, exists then *eClass* can evolve this initial classifier in on-line mode. In other words, the fuzzy rule base will evolve incorporating new rules, modifying and/or, possibly, removing some of the previously existing ones. Additionally, the parameters of both, the antecedent and the consequent parts are adapted. Note that *eClass* can start with an empty rule-base, which is a unique feature of this approach. The proposed approach is free from user-specified parameters and the mechanism of forming new rules is very robust. In this paper, four different modelling architectures are described and compared. The architectures are based on i) unsupervised cluster partitions, *eClassC*; ii) Sugeno fuzzy models with singleton consequents, *eClassA*; iii) Takagi-Sugeno fuzzy models with linear consequent functions, *eClassB*; and iv) a multi-model classification architecture, where separate TS regression models are combined to form an overall classification output of the system, *eClassM*. A thorough comparison of the results when applying each of these architectures and the results using previously existing classifiers has been made using an online interactive self-adaptive image classification framework.

Keywords: evolving fuzzy rule-based classifiers, incremental learning from scratch, Mountain and subtractive clustering, weighted recursive least squares.

I. INTRODUCTION

CLASSIFICATION is a problem that is well studied and understood. One of the widely referenced texts is [1] where classification is treated in the context of pattern recognition. In [2] classification is considered in the light of statistical learning techniques. In [3] a study on fuzzy rule-based classifiers is given. There are numerous practical problems in industrial systems, robotics, defence, bio-medical and other application domains which call for classification, such as fault detection and isolation, early cancer diagnosis, product quality monitoring and control, machine health monitoring and prognostics, automatic target recognition, etc. [4,5].

The *classifier* can be considered as a mapping from the feature domain onto the class labels domain. A number of different types of classifiers exists which use different approaches to perform this mapping such as: i) linear

discriminant analysis [1]; ii) fuzzy rule-based classifiers [6], [7],[39]; iii) decision trees (e.g. C4.5 [31], CART [32]); iv) neural networks-based [8]; v) support vector machines [9], etc. Practically all of them assume training in *batch* mode (when all the training data are known including their class labels). In many practical applications, however, we deal with data *streams* coming from sensory readings or Internet etc. [3]. Additionally, even if the data is available *off-line* as in market basket analysis, genome data etc. the volume of this data is huge and prohibits the direct use of well established learning methods [4]. Very often storing the complete data is practically impossible. This requires addressing the problem of classification of *streaming* data in *real-time* [34].

It is well known that fuzzy rule-based systems are universal function approximators [35]; they are suitable for extracting interpretable knowledge. Therefore, they are viewed as a promising framework for designing effective and powerful classifiers. The type of classifiers that can be built using the recently introduced evolving fuzzy rule-based systems [10],[11] can be called evolving [12] which differs from ‘evolutionary’. Evolving fuzzy rule-based classifiers develop and adapt in on-line mode the non-linear classification surface. Evolutionary/genetic algorithms have recently been used for design of fuzzy rule-based systems in general [13] and classifiers in particular [6],[39]. They are based on the off-line optimization of one or more criteria in designing the fuzzy rule-base (classifier) using paradigms that stem from Nature such as mutation, crossover, and reproduction. Evolving in the sense that we use it in our paper and related works includes self-organising, self-developing in terms of the classifier (rule-base) structure. In this sense this paradigm can be considered as a higher level of adaptation (adaptation is usually related to parameters not to the structure of the systems [15]). Note, that similar principles were used by the authors in developing evolving classifiers also in [14] and [23]. The concept is taken further in this paper comparing to [14] by analysing different possible architectures of *eClass*. Comparing to [23] the backbone of the approach is different – we use here and in [14] the evolving fuzzy Takagi-Sugeno, eTS approach while in [23] we extended FLEXFIS [27] and its modification FLEXFIS-Mod [36] to the classification case (called FLEXFIS-Class), both families originally designed for fuzzy regression modelling tasks. The eTS family of evolving TS models (eTS, MIMO-eTS, exTS) has been recently

applied successfully to a number of identification [11,18], time-series prediction [18], fault detection [17], and control [19] problems.

In this paper four different architectures are presented, studied, and compared in Section II: i) *eClassC* –classification based on *eClustering* combined with nearest neighbour type of classification (so called ‘winner takes all’); ii) *eClassA* – evolving fuzzy rule-based classifier with zero order Sugeno consequents (singletons that represent the class labels); iii) *eClassB* – evolving fuzzy rule-based classifier with first order TS (locally linear) consequents; iv) *eClassM* – multiple model classifier. In Section III the methodology and algorithms for evolving these structures with sample-wise loaded data are presented.

The experimental results demonstrated in Section IV include a well known benchmark problem and a real-life data of a self-adaptive online image classification framework. The results demonstrate that the proposed evolving fuzzy rule-based classifier, *eClass* in all of its modifications has a very good classification performance; it is computationally very efficient, and is, thus, suitable for real-time applications such as classification streaming data, robotic applications, e.g. target and landmark recognition, real-time machine health monitoring and prognostics, fault detection and diagnostics etc. This approach is transparent, linguistically interpretable, and applicable to both fully unsupervised and partially supervised learning. While the low order architectures (*eClassC* and *eClassA*) are computationally superior, with high transparency (low number of fuzzy rules and parameters) and simple structure, the performance of *eClassB* and *eClassM* are superior to all the other structures and to the previously existing classifiers. The multiple model structure, *eClass M* has similar performance to *eClassA* but a more complicated structure

II. THE PROPOSED ARCHITECTURES OF ECLASS

The proposed classifier, *eClass* uses fuzzy rule-bases as a framework. The antecedent part concerns the features and the consequent part differs for different types of *eClass*. The following general form of fuzzy rules is used:

$$R^i : IF(x_1 \text{ is } x_1^{i*}) \text{ AND...AND } (x_n \text{ is } x_n^{i*}) \quad (1) \\ THEN(Conseq^i)$$

where $x=[x_1, x_2, \dots, x_n]^T$ is the vector of features; R^i denotes the i^{th} fuzzy rule; $i=[1, N]$; N is the number of fuzzy rules; $(x_j \text{ is } x_j^{i*})$ denotes the j^{th} fuzzy set of the i^{th} fuzzy rule; $j=[1, n]$; x^{i*} is the prototype (focal point) of the i^{th} rule antecedent.

Note that the type of the classifier depends on the type of the consequent and can be:

- a) Associated with the nearest cluster (*eClassC*)

$$Conseq^i = Cluster^i, i=[1, N] \quad (1a)$$

- b) Zero order MIMO Takagi-Sugeno (TS) type [30] when the consequents are the class labels (*eClassA*):

$$Conseq^i = [L_{01}^i \quad L_{02}^i \quad \dots \quad L_{0M}^i]^T \quad (1b)$$

where L denotes a binary (0/1) class label;

- c) First order MIMO TS type [30] when the consequents are linear classifiers (*eClassB*):

$$Conseq^i = [1, x^T] \begin{bmatrix} \alpha_{01}^i & \alpha_{02}^i & \dots & \alpha_{0M}^i \\ \alpha_{11}^i & \alpha_{12}^i & \dots & \alpha_{1M}^i \\ \dots & \dots & \dots & \dots \\ \alpha_{n1}^i & \alpha_{n2}^i & \dots & \alpha_{nM}^i \end{bmatrix}^T \quad (1c)$$

- d) M first order TS type classifiers (*eClassM*) when each classifier predicts the degree of membership to that class:

$$Conseq_j^i = [1, x^T] [\alpha_{01}^i \quad \alpha_{02}^i \quad \dots \quad \alpha_{0n}^i]^T \quad (1c)$$

where $i=[1, N]$; $j=[1, M]$

Note that the number of fuzzy rules, N is not necessarily the same as the number of classes, M . For *eClassC* the number of clusters formed, N can be more, less or equal to the number of classes, M . Since there are no labels provided (the learning is unsupervised) there is no direct link between the two. For *eClass A, B, and M* there is a requirement to have at least one fuzzy rule per class and, therefore, the following relation holds for these types of classifiers:

$$N \geq M; i=[1, N]; j=[1, M] \quad (2)$$

The firing degree for each fuzzy rule is determined as a t-norm (a form to represent the logical AND) [21]:

$$\tau^i = \prod_{j=1}^n \mu_j^i(x_j); i=[1, N] \quad (3)$$

where μ_j^i is the membership value that describes the degree of association with the i^{th} prototype. We assume it to be of Gaussian form due to its generalization capabilities [1]:

$$\mu_j^i = e^{-\frac{1}{2} \left(\frac{d_{ij}}{\sigma_j^i} \right)^2} \quad i=[1, N]; j=[1, n] \quad (4)$$

where d_{ij} is the Euclidean distance between a sample and the prototype (focal point) of the i^{th} fuzzy rule; σ_j^i is the spread of the membership function, which also represents the radius of the zone of influence of the fuzzy rule.

The spread of the membership σ can be determined using the data scatter [18] per cluster:

$$\sigma_k^i = \sqrt{\frac{1}{S_k^i} \sum_{j=1}^{S_k^i} d(x_j^{i*}, x_j)^2}; \sigma_0^i = 1 \quad (5)$$

where $i=[1, n]$ is the number of clusters; $d(x_j^{i*}, x_j)$ denotes

the distance from cluster centre to new sample assigned into this cluster.

The scatter can be updated recursively by [14,18]:

$$(\sigma_k^l)^2 = (\sigma_{k-1}^l)^2 + \frac{1}{S_k} (d(x_k^{l*}, x_k)^2 - (\sigma_{k-1}^l)^2) \quad (6)$$

The nearest neighbour (also known as ‘winner-takes-all’) classifier is used to determine the label of the winning class [13]:

- a. For *eClassC* the sample is associated with the nearest cluster (thus associating with its label if any, note we do not need to know this label):

$$j^* = \arg \max_{i=1}^N (\tau^i); i = [1, N] \quad (7a)$$

where $j^* \in [1, N]$ denotes the index of the winning cluster.

- b. For *eClassA* the label of the winning prototype is taken:

$$L^* = \max_{i=1}^M (L_i^{j^*}); j^* = \arg \max_{j=1}^N \lambda^j \quad (7b)$$

where $\lambda^i = \tau^i / \sum_{j=1}^N \tau_j; i = [1, N]$ denotes the normalised firing level of the i^{th} fuzzy rule.

- c. For *eClassB* the label of the winning prototype is taken:

$$L^* = \max_{i=1}^M (y_i) \quad (7c)$$

where $y_l = \sum_{i=1}^N \lambda^i \left(a_{0M} + \sum_{j=1}^n a_{jl} x_j \right); l = [1, M]$ is the output of the multi-input-multi-output (MIMO) TS model used [30].

- d. For *eClassM* the label of the winning prototype is also determined by (7c) but instead of one MIMO type TS classifier [14] we build M separate multi-input-single-output (MISO) TS classifiers *per class*. This procedure is also called fuzzy regression of an indicator matrix [23], as it acts on a specific transferred form of the original input data.

III. METHODOLOGY AND PROCEDURE FOR EVOLVING THE CLASSIFIERS

A. Methodology

Evolving the fuzzy classifiers of *eClass* type can start ‘from scratch’ or from an initial classifier. *eClassC* stands out as a fully unsupervised classifier. As such it is a clustering algorithm, which is on-line and evolving (the number of clusters is not pre-specified). The algorithm is influenced by the Mountain [28] and subtractive clustering approaches [29]. It is described in more details elsewhere [11],[18]. The basic

idea is to measure the potential of a prototype to become a new cluster centre and to compare this with the potentials of the existing cluster centres so far. The potential is a measure of data density (in the feature space for the case of a classifier). That is a prototype is formed around a *representative* sample with feature vector similar to which there are many other samples. Prototypes are also formed around samples that will ensure coverage of the feature space.

The three other types of classifiers (*eClassA*, *eClassB*, and *eClassM*) are based on the evolving TS type fuzzy rule-based systems of MIMO [30] or MISO [11] type with singletons (zero order) or linear (first order) outputs of the so called eTS family [11],[18],[30]. The fuzzy rule-base build in this way is used to approximate the classification surface by a non-linear regression over the features. In essence, this approach combines the evolving clustering as described above with a fuzzily weighted mixture of recursive least square (wRLS) estimators [11]. Details of this algorithm are also given elsewhere [11],[18],[30].

A special contemplation regarding the application of the algorithms for learning eTS model family [11], [18], and [30] follows from the definitions (7b) and (7c) of the *eClassB* and *eClassM* classifiers. Since *max* aggregation of the normalized firing levels λ^j is used in the procedures for calculation of the winning prototype (7b) and (7c), it is reasonable to consider also *max* guided aggregation of the TS sub-system contributions during the learning phase. This can be accomplished by replacing the conventional normalization of

the firing levels of the rules: $\lambda^i = \tau^i / \sum_{j=1}^N \tau_j; i = [1, N]$

in the inferred TS model output:

$$y = \psi^T \theta$$

where $\theta = [\pi_1^T, \pi_2^T, \dots, \pi_R^T]^T$ is a vector formed by the sub-model parameters and $\psi = [\lambda_1 x_e^T, \lambda_2 x_e^T, \dots, \lambda_R x_e^T]^T$ is a vector of the inputs that are weighted by the normalized firing levels of the rules, $\lambda_i, i = [1, R]$, by a *max*-like normalization

$$\lambda^i = \tau^{\alpha(i)} / \sum_{j=1}^N \tau^{\alpha(j)}; i = [1, N],$$

where parameter $\alpha > 1$ determines an increasing level of *max*-like aggregation of the firing levels. For higher values of α the higher firing levels τ^i 's are reinforced providing higher weights λ_i to the respective subsystems in the TS model.

Therefore, the eTS learning algorithm is adapted to the classification objective. For $\alpha = 1$ the *mean* type aggregation of the subsystems that is characteristic for approximation type of applications of the eTS models occurs.

Note that *eClassC* coincides with *eClassA*, if the class labels are known, because according to (7a) and (7b):

$$\arg \max_{i=1}^N (\tau^i) = \arg \max_{i=1}^N \left(\tau^i / \sum_{j=1}^N \tau_j \right); i = [1, N]$$

In case when the labels are not known (a fully unsupervised classification based on the nearest neighbour principle) the learning is based on **all** data samples. The learning when the labels are known (*eClassA*) is **per class**.

B. Procedures

In this paper we describe the procedures for each of the proposed evolving fuzzy rule-based classifiers. All algorithms have one characteristic in common: the rule evolution and update of the antecedents takes place in the feature space, i.e. clusters evolve if the new incoming data is highly descriptive or if it expands the coverage of the fuzzy rule-base. The major difference lies in the update scheme for the consequent parts. In *eClassC* it is virtually not existent as such; in *eClassA* it is a label, singleton that has an integer (binary) value; in *eClassB* and *eClassM* it is (locally) linear.

Algorithm 1 *eClassC*

- 1) Initialise the classifier by either of:
 - a. The first sample, x_1 (assign potential $P_1 := 1$)
 - b. An initial rule-base
- 2) Start a loop while there are new data samples do:
 - a. Read the feature vector, x_k
 - b. Associate the sample with the nearest cluster using (7a)
 - c. Calculate its potential, P_k by [11,21]:

$$P_k(x_k) = \frac{k-1}{\alpha_k(k-1) + \beta_k - 2\gamma_k + (k-1)} \quad (8)$$

where $\alpha_k = \sum_{j=1}^n (x_k^j)^2$; $\beta_k = \beta_{k-1} + \alpha_{k-1}$; $\beta_1 = 0$

$$\gamma_k = \sum_{j=1}^n x_k^j \chi_k^j; \chi_k^j = \chi_{k-1}^j + x_{k-1}^j; \chi_1^j = 0$$

- d. Update the potential of the existing cluster prototypes by [11,21]:

$$P_k(x_k^{i*}) = \frac{(k-1)P_{k-1}(x_k^{i*})}{k-2 + P_{k-1}(x_k^{i*}) + P_{k-1}(x_k^{i*}) \sum_{j=1}^n \|x_k^{i*} - x_k\|_j^2} \quad (9)$$

- e. Compare the two and

- i. **Add** a new cluster

$$IF(P_k(x_k) > P_k(x_k^{i*})) OR (P_k(x_k) < P_k(x_k^{i*})); \forall i = [1, n] \quad (10)$$

- ii. **Replace** a cluster

$$IF(\exists i, i = [1, N]; \mu_i^j(x_k) > e^{-1}; \forall j, j = [1, n]) \quad (11)$$

- iii. **ELSE do not change** the cluster structure

Algorithm 2 *eClassA*

- 1) Initialise the classifier by either of:
 - a. the first data sample **per class** with the pair (feature vector and the class label), $z_1 = [x_1, L_1]$.
 - b. An initial rule-base.
- 2) Start a loop while there are new data samples do:
 - a. Read the feature vector, x_k
 - b. Determine the winning class using (7b)
 - c. Calculate the potential of the pair, $z_k = [x_k, L_k]$ using (8) in respect to z_k
 - d. Update the potential of the existing prototypes by (9) in respect to z_k
 - e. Compare the two and
 - i. Add a new fuzzy rule IF (10)
 - ii. Replace a rule IF (11) holds
 - iii. ELSE do not change the rule-base
 - f. **Increment the time, $k \leftarrow k+1$**

Algorithm 3 *eClassB* differs from **Algorithm 2** in step 2)b, only which is according to (7c) not (7b), whereas the linear parameters a_{ij} are updated by weighted recursive least squares (wRLS), exploiting local learning [18],[30], triggering more flexibility during learning and transparency of the consequent functions.

Algorithm 4 *eClassMM* is based on **Algorithm 3** applied to M separate multi-input-single-output (MISO) TS classifiers **per class** [23]. For classification that class is taken, whose corresponding output value is maximal (Figure 1).

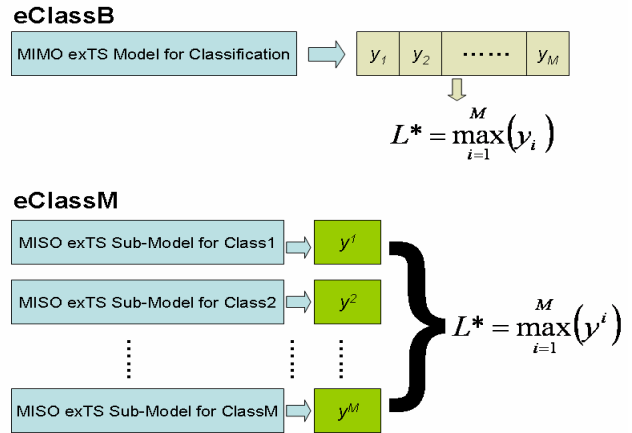


Figure 1 A graphical interpretation of *eClassB* and *eClassM*

IV. EXPERIMENTAL RESULTS

The proposed classifier, *eClass* was tested on a well known benchmark problem and on a real-life problem. It should be noted that despite the clearly off-line nature of these benchmark problems they were used to test the proposed classifier in order to have some comparison.

A. Wine Reproduction Data

The commonly used benchmark problem Wine Reproduction data set [25] is tested in order to illustrate the level of performance of the proposed evolving classifiers in comparison with other popular classification algorithms. The data set contains the chemical analysis of wines derived from three different cultivars. There are three classes, 178 samples with 13 continuous numerical features available in the data set. Original order of the data set from [25] is used, and no prior feature selection is performed.

The results of testing *eClass* with Wine data set are tabulated in Table I, along with results from some comparable

TABLE I
RESULTS FOR WINE DATA

	Class. Rate	# rules	Total time, s
iPCA [24]	80.3%	7*	-
Smooth Boost [26]	86.1%	-	-
<i>eClassC</i>	63.48%	9	0.92**
<i>eClassA</i>	90.45%	12	1.18**
<i>eClassB</i>	94.38%	9	3.52**
<i>eClassM</i>	97.19%	28	7.18**

* number of eigen-values (note that they are not transparent as the fuzzy rules and do not represent the features directly)

** The test is carried out on a laptop computer with a CPU 1.6GHz; the time is for processing all the data samples

classifiers.

The result shows that *eClassC* and *eClassA* are more computational efficient and their rule bases have better interpretability; while the first order Takagi-Sugeno rule based, *eClassB* and *eClassM* have better accuracy of classification. Please note that the rule numbers listed for *eClassM* is the summation of rule numbers of each MISO sub-models.

B. On-line Image Classification

In this section an application example is given, which includes an automatically self-reconfigurable and adaptive fault detection framework for images which classifies each image as good or bad, and evolves the classifier upon operator's feedback and the data. The images are taken from an online production process with a high frequency with the aim to supervise the system, as they may show errors in a production process. This framework including pre-processing, segmentation and classification is shown in Figure 2.

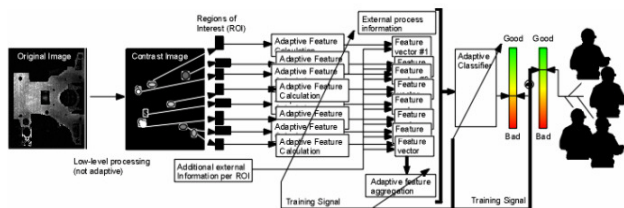


Fig 2. Dynavis framework for on-line images classification.

In principle, each type of image may be processed through

the classification framework as shown in Fig. 1. The only assumption is that a master image is available: the purpose is to generate deviation images by subtracting newly recorded images from the master one in order to be able to classify the image into good or bad (depending on the structure and characteristics of the deviation pixels).

For the evaluation of our approaches we applied image data from a CD-imprint production process, where faults due to weak colours, wrong palettes etc. should be detected within a process frequency of about 1 Hz. The data stream comprises 1164 images that were recorded one by one. *eClass* was evolved on-line starting 'from scratch', so it was able to classify from the second sample onwards. In order to compare, however, with the other approaches which require batch learning over certain data set we have evolved *eClass* with the first 776 images and stopped evolution (fixed the rule-base). With the remaining 388 samples we made classifications only (no learning and evolution). Seventeen aggregated features were extracted, describing the distribution, density, shape etc. of the pixel fragments in the deviation images.

The miss-classification rates on this test data set are demonstrated in Table II. The superiority in terms of low computational costs (time), high precision, and low complexity (low number of fuzzy rules) of *eClass* family is clearly visible.

TABLE II
RESULTS FOR ON-LINE IMAGE CLASSIFICATION

	Class. Rate	# rules	Total time, s
CART [32]	91.24%	-	Off-line
PNN [33]	90.46%	-	Off-line
FLEXFISClass SM [23]	83.33%	39	1.58
FLEXFISClass MM [23]	90.98%	62	3.56
FLEXFISClass MM* [23]	91.24%	62	3.56
<i>eClassC</i>	76.28%	12	0.05 *
<i>eClassA</i>	79.70%	12	0.05 *
<i>eClassB</i>	91.24%	7	0.28 *
<i>eClassM</i>	91.24%	14	0.53 *

* Running time for the last 388 samples with fixed *eClass* model; the test is carried out on a laptop computer with a CPU 3.0GHz.

V. CONCLUSIONS

In this paper, four different fuzzy model architectures were presented, whose inner structures and parameters are evolved and incrementally updated. In this way, four different schemes based on *eClass* procedure were presented, namely: *eClassC*, *eClassA*, *eClassB* and *eClassM*.

Both tests on Wine data and on the adaptive image data shows that the proposed techniques for training fuzzy rule-based classifiers have the advantage of that evolving its structure from scratch without losing much precision (classification rate). In the online evolving mode, the performance slightly deteriorates, but is comparable to the results of the well-known and renowned batch modelling approaches CART [32] and probabilistic NN [33]. The first order classifiers *eClassB* and *eClassM* have results of the same level of precision as the best of these classifiers. The results

demonstrate that the proposed evolving fuzzy rule-based classifier, *eClass* in any of its modifications has a very good classification performance; it is computationally very efficient, and is, thus, suitable for real-time applications such as classification streaming data, robotic applications, e.g. target and landmark recognition, machine health monitoring and prognostics, fault detection and diagnostics etc. This approach is transparent, linguistically interpretable, and applicable to both fully unsupervised and partially supervised learning. While the low order architectures (*eClassC* and *eClassA*) are computationally superior, with high transparency (low number of fuzzy rules and parameters) and simple structure, the performance of *eClassB* using multi-input- multi-output, MIMO TS is superior to all the other structures and to the previously existing classifiers. The multiple model structure, *eClassM* has similar performance to *eClassA* with MIMO TS but a more complicated structure.

ACKNOWLEDGEMENTS

This work was supported by the European Commission (project Contract No. STRP016429, acronym DynaVis). This publication reflects only the author's view.

REFERENCES

- [1] R. O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification - Second Edition*. Wiley-Interscience, Chichester, West Sussex, England, 2000.
- [2] G. Fayyad, Piatetsky-Shapiro, P. Smyth. *From Data Mining to Knowledge Discovery: An Overview*, Advances in Knowledge Discovery and Data Mining, MIT Press, 1996.
- [3] P. Domingos and G. Hulten. *Catching Up with the Data: Research Issues in Mining Data Streams*, Workshop on Research Issues in Data Mining and Knowledge Discovery, Santa Barbara, CA, 2001.
- [4] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Heidelberg, Germany: Springer Verlag, 2001.
- [5] V. N. Vapnik, *The Statistical Learning Theory*, Springer, 1998.
- [6] M. Butz, *Rule-based Evolutionary Online Learning Systems: A Principal Approach to LCS Analysis and Design*, Physica Verlag, v.191, Berlin, Heidelberg, Germany, 2006, ISBN 3-540-25379-3.
- [7] J. A. Roubos, M. Setnes and J. Abonyi, *Learning Fuzzy Classification Rules from Data, Information Sciences—Informatics and Computer Science: An International Journal*, vol. 150, pp. 77-93, 2003
- [8] C. M., Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.
- [9] V.Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995
- [10] P. Angelov, *Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems*. Berlin, Germany: Springer Verlag, 2002.
- [11] P. Angelov, D. Filev, "An approach to on-line identification of evolving Takagi-Sugeno models", *IEEE Trans. on Systems, Man and Cybernetics, part B*, vol.34, No1, pp. 484-498, 2004.
- [12] P. Angelov, R. Buswell, Identification of Evolving Rule-based Models (2002) *IEEE Transactions on Fuzzy Systems*, vol. 10, No5, pp.667-677.
- [13] Setnes M., J. A. Roubos, Transparent Fuzzy Modelling using Fuzzy Clustering and GA's. In: *Proc. 18th Annual Conf. of the North American Fuzzy Information Processing Society, NAFIPS*, 10-12 June 1999, New York, USA, pp.198-202.
- [14] P. Angelov, X. Zhou, F. Klawonn, Evolving Fuzzy Rule-based Classifiers, *First 2007 IEEE International Conference on Computational Intelligence Applications for Signal and Image Processing*, April 1-5, 2007, Honolulu, Hawaii, USA, to appear.
- [15] G. C. Goodwin, K. S. Sin, *Adaptive Filtering, Prediction, and Control*, Prentice Hall, ISBN-10: 013004069X, 1984.
- [16] F. Klawonn, P. Angelov, Evolving Extended Naive Bayes Classifier, In: S. Tsumoto, C.W. Clifton, N. Zhong, X. Wu, J. Liu, B.W. Wah, Y.-M. Cheung: *Proc. Sixth IEEE International Conference on Data Mining*. IEEE, Los Alamitos (2006), pp. 643-647 (ISBN 0769527027).
- [17] P. Angelov, V. Giglio, C. Guardiola, E. Lughofer, An Approach to Model-based Fault Detection in Industrial Measurement Systems with Application to Engine Test Benches, *Measurement Science & Technology*, v.17 (7), 2006, 1809-1818.
- [18] P. Angelov, X. Zhou, Evolving Fuzzy Systems from Data Streams in Real-Time, *Proc. 2006 International Symposium on Evolving Fuzzy Systems*, UK, IEEE Press, pp.29-35, ISBN 0-7803-9719-3.
- [19] L.-X. Wang "Fuzzy Systems are Universal Approximators", *Proc. FUZZ-IEEE*, San Diego, CA, USA, pp.1163-1170, 1992.
- [20] P. Angelov, A Fuzzy Controller with Evolving Structure, *Information Sciences*, ISSN 0020-0255, vol.161, 2004, pp.21-35.
- [21] R. R. Yager, D.P. Filev, "Learning of Fuzzy Rules by Mountain Clustering," *Proc. of SPIE Conf. on Application of Fuzzy Logic Technology*, Boston, MA, USA, pp.246-254,1993.
- [22] X. Zhou, P. Angelov, Autonomous Self-localization in Completely Unknown Environment using Evolving Fuzzy Rule-based Classifier, *1st IEEE Symp. on Comp. Intelligence for Security and Defense Applic. (CISDA) 2007*, April 1-5, 2007 Honolulu, HI, USA, to appear.
- [23] E. Lughofer, P. Angelov, X. Zhou, Evolving Single- and Multi-Model Fuzzy Classifiers with FLEXFIS-Class, *IEEE Intern. Conf. on Fuzzy Syst., FUZZ-IEEE2007* 23-26 July, 2007, London, England, to appear.
- [24] S. Pang, S. Ozawa, N. Kasabov, "Incremental Linear Discriminant Analysis for Classification of Data Streams", *IEEE Trans. on Systems, Man and Cybernetics, part B*, vol.35, No5, pp. 905-914, 2005.
- [25] UCI Machine Learning Repository, <http://www.ics.uci.edu/~mllearn/MLRepository.html>, accessed on 27 Jan 2007.
- [26] R. Jin, J. Zhang, A Smoothed Boosting Algorithm Using Probabilistic Output Codes, *Proc. 22nd International Conference on Machine Learning*, Bonn, Germany, pp.361-368, 2005.
- [27] E. Lughofer and E. P. Klement, "FLEXFIS: A variant for incremental learning of Takagi-Sugeno fuzzy systems, in Proc. FUZZ-IEEE 2005, Reno, Nevada, USA, 2005, pp.915-920.
- [28] R. Yager and D. Filev, Approximate Clustering via the Mountain Method, *IEEE Trans. on Systems and Cybernetics*, vol. 24 (8), 1994
- [29] S. Chiu, Fuzzy Model Identification based on Cluster Estimation, *Journal of Intelligent and Fuzzy Systems*, vol.2 (3), pp. 267-278, 1994
- [30] P. Angelov, C. Xydeas, D. Filev, On-line Identification of MIMO Evolving Takagi-Sugeno Fuzzy Models, Intern. Joint Conf. on Neural Networks and Intern. Conf. on Fuzzy Systems, IJCNN-FUZZ-IEEE, Budapest, Hungary, 25-29 July, 2004, 55-60, ISBN 0-7803-8354-0.
- [31] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc, U.S.A., 1993.
- [32] L. Breiman, J. Friedman, C.J. Stone and R.A. Olshen, *Classification and Regression Trees*, Chapman and Hall, Boca Raton, 1993.
- [33] T. Hastie and R. Tibshirani, 1996. Discriminant Adaptive Nearest Neighbor Classification. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*. Vol. 18 (6), pp. 607-616, 1996.
- [34] M. Garofalakis, J.Gehrke and R. Rastogi, Querying and Mining Data Streams: you only get one look, *Proc. of the 2002 ACM SIGMOD Intern. Conf. on Management of Data*, pp. 635-641, ACM Press, 2002.
- [35] L.X. Wang, Fuzzy Systems are Universal Approximators, *Proc. 1st IEEE Conf. Fuzzy Systems*, pp. 1163-1169, San Diego, 1992.
- [36] E. Lughofer and U. Bodenhofer, Incremental Learning of Fuzzy Basis Function Networks with a Modified Version of Vector Quantization, *Proc. of IPMU 2006*, Paris, France, vol. 1, pp 56-63, 2006.
- [37] P. Angelov and E.Lughofer, Data-Driven Evolving Fuzzy Systems using eTS and FLEXFIS: Comparative Analysis, Intern. Journal of General Systems, to appear
- [38] T. Takagi and M. Sugeno, Fuzzy Identification of Systems and its Applications to Modeling and Control, *IEEE Trans. on Systems, Man and Cybernetics*, vol. 15 (1), pp. 116-132, 1985.
- [39] O. Cordon, F. Herrera, M. J. del Jesus, P. Villar, A Multi-objective genetic algorithm for feature selection and granularity learning in fuzzy-rule based classification systems, *Proc. Joint 9th IFSA World Congress and 20th NAFIPS Intern. Conf.*, 25-28 July 2001, Vol. 3, pp. 1253-1258, Vancouver, BC, Canada, ISBN: 0-7803-7078-3.