2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

A Novel Meta-Cognitive-based Scaffolding Classifier to Sequential Non-stationary Classification Problems

Mahardhika Pratama [1], Meng Joo Er [2,3], Sreenatha.G.Anavatti [1], Edwin Lughofer [4] Ning Wang [3], Imam Arifin [5]

[1] School of Engineering and Information Technology, The University of New South Wales, email: pratama@ieee.org, s.anavatti@adfa.edu.au

[2] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, email: emjer@ntu.edu.sg

[3] Marine Engineering College, Dalian Maritime University, China, email: n.wang.dmu.cn@gmail.com

[4] Department of Knowledge-Based Mathematical Systems, Johannes Kepler University, Austria, email: edwin.lughofer@jku.at

[5] Department of electrical engineering, Institut Teknologi Sepuluh Nopember, Indonesia, email :arifin-i@ee.its.ac.id

***Abstract*—a novel meta-cognitive-based scaffolding classifier, namely Generic-Classifier (gClass), is proposed in this paper to handle non-stationary classification problems in the single-pass learning mode. Meta-cognitive learning is a breakthrough in the machine learning where the learning process is not only directed to craft learning strategies to exacerbate the classification rates , i.e., *how-to-learn* aspect, but also is focused to accommodate the emotional reasoning and commonsense of human being in terms of *what-to-learn* and *when-to-learn* facets. The crux of gClass is to synergize the scaffolding learning concept, which constitutes a well-known tutoring theory in the psychological literatures, in the how-to-learn context of meta-cognitive learning, in order to boost the learner's performance in dealing with complex data. A comprehensive empirical studies in time-varying datasets is carried out, where gClass numerical results are benchmarked with other state-of-the-art classifiers. gClass is, generally speaking, capable of delivering the most encouraging numerical results where a trade-off between predictive accuracy and classifier's complexity can be achieved.**

Keywords—gClass, Evolving Fuzzy Classifier, Fuzzy System, Neural Network.

I. INTRODUCTION

The realm of evolving system of [16], [34]-[36] has been enriched by Suresh et al of [1]-[5] where they in essence unveil a new avenue to amalgamate the emotional perspective and commonsense of human being into the evolving system scope, which is thus far cognitive in nature. Suresh et al in [1]-[5] contends that meta-cognitive aspects can avail to mimic the ability of human being to appraise the observed knowledge with respect to environment and their knowledge. Specifically, it hedges the learning machine to learn full streaming data without being able to favor particular training samples and, in parallel to that, underpins the learners to figure out the compatible time instants to not regularly learn streaming data.

The meta-cognitive learning with respect to the meta-cognition and meta-memory theories in [6]-[8] comprises three cornerstones which are *what-to-learn*, *how-to-learn* and *when-to-learn* components. In retrospect to [1]-[5], these three learning phases can be emulated in the scope of machine learning with the sample deletion strategy, schema theory and sample reserve strategy.

Scaffolding theory constitutes a firm theoretical framework of tutoring theory [9], aiding the learner in sorting out the complex task. The centric notion is to confer the learning supervision, capable of deciphering the complicated learning problem into a simpler one. The passive supervision endows the learner with the experience-feedback rule, where the corrective actions, supplied by this mechanism, conceive a reciprocal relationship with the predictive output of the machine learning. The active supervision consist of three phases, which are problematizing (drift detection) [10], complexity reduction (feature selection) [11], fading (rule pruning).

This paper serves a novel meta-cognitive classifier of the so-called Generic Classifier (gClass), where the *how-to-learn* component is specifically devised in accordance with the Scaffolding and Schema theories of [12]. gClass complies a holistic concept of evolving system, where it can embark the training process from scratch with an empty knowledge. The fuzzy rules can be autonomously extracted from the high potential and summarization power streaming data, whereas the superfluous and obsolete fuzzy rules can be pruned to relieve the computational and structural complexities. More interestingly, the fuzzy rules, which have been trimmed in the earlier training episodes, can be recalled, whenever their contributions are entailed to apprehend the up-to-date data distribution. This component plays a precarious role in the case of cyclic drift, where the outdated data distribution is revisited. Apart from that, the online feature weighting mechanism and local drift handling strategy are amalgamated into the how-to-learn component. In the realm of the what-to-learn learning module, the active learning procedure is incorporated, where the merit, in comparison with Suresh et al work in [1]-[5], is not only capable of truncating the inconsequential training samples, but also capable of mitigating the annotation efforts by operator, which is desired in practice. Conversely, the standard sample reverse technique is plugged in the when-to-learn learning module. Importantly, all learning modules are committed in the training process in the single-pass learning mode and exclude pre or post-processing approach.

Another novelty of gClass leans on its cognitive component, where the generalized fuzzy rule is put forward. The premise part of the generalized fuzzy rule triggers a hyper-ellipsoidal cluster arbitrarily rotated in any positions, which reaps some merits against a hyper-ellipsoidal cluster in main position or a hyper-spherical cluster. Unlike traditional Takagi Sugeno Kang (TSK) fuzzy system, the consequent part of gClass is carved on the local non-linear Chebyshev function [13]. The construct of this approach is to increase the degree of freedom of the rule consequent, thus rectifying a local output mapping ability in comparison with the local liner hyper-plane

This work is supported by the National Natural Science Foundation of P. R. China (under Grants 51009017 and 51379002), Applied Basic Research Funds from Ministry of Transport of P. R. China (under Grant 2012-329-225-060), China Postdoctoral Science Foundation (under Grant 2012M520629), Program for Liaoning Excellent Talents in University (under Grant LJQ2013055), and Fundamental Research Funds for the Central Universities of P. R. China (under Grants 2009QN025, 2011JC002 and 3132013025). The third author acknowledges financial support of the Austrian fund for promoting scientific research (FWF, contract number I328-N23, acronym IREFS), and the research programme at the Austrian Center of Competence in Mechatronics (ACCM), which is a part of the COMET K2 program of the Austrian government.

function. Comprehensive numerical studies have been undertaken making use of datasets emphasizing various concept drifts, engaging benchmarks with gClass counterparts. In what follows, gClass can outperform other consolidated algorithms in terms of the predictive quality, rule-base simplicity and runtime, while consuming and labeling less streaming data than those of its counterparts.

The remainder of this paper is organized as follows: Section 2 discloses the cognitive component of gClass, Section 3 elaborates algorithmic development of gClass, Section 4 outlines the empirical study on the numerous artificial and real-life datasets and last section deduces this paper.

II. COGNITIVE COMPONENT OF GCLASS

The classification accuracy of TSK fuzzy system can be substantiated with the use of the non-linear function as the consequent parameters in lieu of the standard linear hyper-plane. The linear hyper-plane consequent, arguably, does not exploit a full merit of the local approximation [14],[15]. The use of the functional link-based rule consequent has been pioneered with Y-Y.Lin et al in [14], stemming from the functional link neural network in [15]. Nonetheless, the work of [14] is reliant on the trigonometric-based rule consequent, which is adjusted by the global adaptation scheme. gClass benefits from the Chebishev function affirming a simpler mathematical from than that of the trigonometric function. Furthermore, the rule consequent is locally adjusted, thus being able to be interpreted as the non-linear function, snuggling along the real trend of the classification surface. Accordingly, the local learning approach can grant a sort of rule transparency in the context of operating region of the system being modeled. The generalized fuzzy rule is expressed as follow:

$$R_i : \textbf{IF}\ X \textit{ is close to } R_i \textbf{ Then } y_i^{\,o} = x_e \Omega_i \tag{1}$$

where R_i stands for a multi-dimensional kernel, constructed from a concatenation of fuzzy sets, representing a rule (membership function) and all its antecedent parts, whereas $y_i^{\,o}$ denotes the regression output to *o-th* class in the *i-th* rule. Conversely, x_e constitutes an expanded input vector yielded by a non-linear mapping based on the Chebyshev function. The mathematical expression of Chebyshev polynomial is as follows:

$$T_{n+1}(x) = 2x_j T_n(x_j) - T_{n-1}(x_j) \tag{2}$$

For instance: suppose X is 2-D input pattern $[x_1, x_2]$. Hence, the expanded input vector turns out to be $x_e = [1, x_1, T_2(x_1), x_2, T_2(x_2)]$. Obviously, the Chebyshev function-based extended input vector $x_e \in \Re^{1\times(2u+1)}$ can mitigate the consequent parameters in comparison with the use of the trigonometric function-based extended input vector $x_e \in \Re^{1\times(3u+1)}$, where u denotes input dimension. . Note that Ω_i is a weight vector, which can be formulated as *Multi-Input-Single-Output(MISO)* $y_i \in \Re^{(2u+1)\times 1}$ or as *Multi-Input-Multi-Output(MIMO)* $y_i \in \Re^{(2u+1)\times m}$ structure, where m labels the number of output dimension or classes. The output of the local subsystem is inferred as follow:

$$\hat{y}_o = \frac{\sum_{i=1}^{p} R_i y_i^{\,o}}{\sum_{i=1}^{p} R_i} = \frac{\sum_{i=1}^{p} \exp(-(X-C_i)\Sigma_i^{-1}(X-C_i)^T) y_i^{\,o}}{\sum_{i=1}^{p} \exp(-(X-C_i)\Sigma_i^{-1}(X-C_i)^T)} \tag{3}$$

where C_i designates a Center or template vector of *i*-th Gaussian function $C_i \in \Re^{1\times u}$ and $\sum_i^{-1}$ points out an inverse non-diagonal covariance matrix $\sum_i^{-1} \in \Re^{u\times u}$, whose elements stands for the spreads of the Gaussian function in every direction or dimension σ_{hi} whereas p signifies the number of fuzzy rules. If the MIMO architecture of [16] is benefited to craft the decision boundary, the output of the classifier can be produced taking maximum output of the local sub-system as follow:

$$y = \max_{o=1,..,m}(\hat{y}_o) \tag{4}$$

It is worth-noting that the MIMO architecture is put forward to deliver the classification decision as it is more reliable to surmount the class overlapping problem [17]-[20]. Moreover, the non axis-parallel cluster can be spurred by the non diagonal covariance matrix. This property can shed some virtues over the traditional axis-parallel or hyper-spherical cluster. It is endued by the scale-invariant characteristic and allows a more exact coverage of data distributions or classes, notably when the data are not scattered in the main axis [21]. Apart from that, it can evade a severe information loss of input variable interactions, which can be catastrophically omitted by the diagonal covariance matrix or the same spread per input variable. One can envisage that this rule promise can arouse a demerit in the sense of the rule transparency as the input variables cannot directly associated with a linguistic label. Even so, two avenues to extract the fuzzy set from the generalized rule premise have been proposed in our past works [17],[18], however, we do not detail these techniques in this paper for the sake of conciseness.

III. META-COGNITIVE LEARNING SCHEME

Meta-cognitive learning enjoys a mechanism, monitoring the knowledge being injected to it and its existing knowledge. A datum, which is deemed relevant with the learning context, can be captured and labeled with a true class label by the what-to-learn learning module for a subject of learning purpose. Otherwise, it is ruled out without being learned and annotated with the target class. The how-to-learn module is devised by virtues of Schema and Scaffolding theories, encompassing an automatic knowledge building process, rule pruning scenario, drift handling technique, rule adaptation mechanism and online feature weighting algorithm. Another landmark of meta-cognitive learning is capable of pinpointing suitable time instants to consume the training samples, where the sample reserve strategy is utilized. The training sample, which does not satisfy the criteria of the rule growing and adaptation, is consumed after that of the last training sample. This mechanism is fruitful to unveil possible unexplored states of already seen training stimuli.

A. What-to-learn

The what-to-learn facet of gClass is distinguishable with those of Suresh et al works in [1]-[5], where sample pruning strategy is employed. What-to-learn-based active learning devolves an ability to suppress operator's efforts to annotate the training samples, as it does not solicit the true class labels to be at hand, when delving the relevant training samples. Accordingly, gClass can be subsumed as a semi-supervised learning machine, whereas Suresh et al in [1]-[5] is categorized as a fully supervised classifier.

Table 1. consolidated results of benchmarked system in three datasets

algorithm		sin	circle	boolean	Electricity pricing
gClass	classification rate	**0.92±0.3**	**0.91±0.06**	**0.92±0.2**	**0.79±0.08**
	Rule	**3.3±0.9**	**2.4±1.6**	**2.3±0.5**	**2.7±0.5**
	Time	**0.16±0.02**	0.15±0.02	**0.01±0.03**	**2.3±0.5**
	Rule base	**39.6**	**28.8**	**46**	**243**
	Samples	**86.4±24.5**	**156.2±53.7**	**52.2±20.12**	**887.95±15**
pClass	classification rate	0.82±0.2	0.72±0.13	0.83±0.2	0.78±0.05
	Rule	3.3±1.2	2.8±1.1	2.6±0.8	3.2±1.2
	Time	0.17±0.04	0.17±0.008	0.08±0.002	4.23±0.8
	Rule base	39.6	33.6	52	288
	Samples	200	200	100	3172
GENEFIS-class	classification rate	0.81±0.2	0.7±0.03	0.82±0.2	0.75±0.0
	Rule	5.4±2.2	3.2±1.03	2.6±1.1	3.5±1.5
	Time	0.32±0.3	0.25±0.01	0.09±0.05	4.49±0.4
	Rule base	58.8	38.4	52	315
	Samples	200	200	100	3172
eClass	classification rate	0.8±0.2	0.66±0.14	0.8±0.17	0.57±0.09
	Rule	50	50	50	50
	Time	0.25±0.02	**0.08±0.02**	0.24±0.02	2.43±0.2
	Rule base	500	500	250	550
	Samples	200	200	100	3172

Ubiquitous works in the active learning [22] are built upon the pool-based approach, which is computationally prohibitive and does not underpin an online fast labeling process. Lughofer et al in [23] has enriched a landscape of the active learning, where a conflict and ignorance method, compromising with an online learning scenario, was proposed. In this paper, the extended conflict and ignorance (ECI) method is put forward, where the crux ameliorates the measure of ignorance of the original work. The original work enumerates the compatibility measure by means of the rule firing strength, which is deemed conservative as it just takes into account the spatial proximity of the rule focal points and the datum. The original work in [23] excludes the impacts of all training data, which can indeed influence the ignorance, thus being vulnerable with outliers. In what follows, we adopt the Extended Recursive Density Estimation (ERDE) method, empowered as the rule growing module, to appraise the significances of the training data as follow:

$$\min_{i=1,..,P}(ERDE_i) \le ERDE_{p+1} \le \max_{i=1,..,P}(ERDE_i) \quad (5)$$

In this circumstance, the datum clearly does not either incur a novelty to the system or is posited under a coverage of the existing fuzzy rules. In other words, this datum, noticeably, poses a redundant training sample. Another paramount occasion, delineating the ignorance case, is when the conflict situation does not ensue. No conflict case can be observed, when the classifier can yield a confident prediction as follow:

$$conf_{final} = \frac{y_1}{y_1 + y_2} \le (0.5 + \delta) \quad (6)$$

where y_1 and y_2 label the outputs of the two most dominant class whereas δ denotes the tolerance constant, which is set as $\delta = 0.05$ in all of our empirical studies. Undoubtedly, if these two conditions are met, the datum can be repealed without a detrimental effect of classification quality.

B. When-to-learn

In the training process, there is a likelihood for the training sample, which does not fall into the criteria of *what-to-learn* and *when-to-learn*. In principle, such samples do not attract the model updates in the current time, nevertheless, they might be precarious in the future, in order to explore uncovered states of the learned training samples. Such samples are pushed into the rear stack and depleted subsequent to spend the last training signal. In general, the training process is capped off when no further samples exist in the stack. Nevertheless, this may be impossible as the reserve samples are still not the subject of the model updates. Hence, we terminate the training process when number of reserve samples remains the same.

C. How-to-learn

1) Autonomous fuzzy rule recruitment : gClass makes use of three cursors, termed datum significance (DS) method, extended recursive density estimation (ERDE) method and generalized adaptive resonance+ (GART+) theory, to fathom the quality of the datum. The DS method is blueprinted to pry the datum statistical contribution, in turn supplies a contribution of a hypothetical fuzzy rule in the future. The ERDE method is used to figure out the position of the focalpoint in the input space, whether or not it lies on a strategic position in the input space with respect to the all training samples. Meanwhile, GART+ deters the so-called *cluster delamination* phenomenon. That is, one cluster contains two or more data clouds, inevitably worsening the logic of the inpur space parition and rule semantic. In a nutshell, three rule growing criteria are expressed as follows:

$$V_{P+1} \ge \max_{i=1,..,P}(V_i) \quad (7)$$

$$V_{win} \ge \rho_1 \sum_{i=1}^{P} V_i \quad (8)$$

$$ERDE_{P+1} \ge \max_{i=1,..,P}(ERDE_i) or ERDE_{P+1} \le \min_{i=1,..,P}(ERDE_i) \quad (9)$$

where V_{P+1} indicates the volume of a hypothetical new rule (the R+1[st]) and V_i denotes the volume of the *i-th* rule, whereas $ERDE_{P+1}$ stands for the ERDE of the newest datum. ρ_1 labels a predefined constant, whose value is stipulated in the range of [0.1,0.5]. More specifically, the density of the datum can be recursively elicited as follow:

$$ERDE_N = \sqrt{\frac{U_N}{U_N(1+a_N)-2b_N+c_N}} \tag{10}$$

$U_N{}^i = U_{N-1}{}^i + ERDE_{N-1}$, $a_N = C_i \sum_i{}^{-1} C_i{}^T$,

$b_N = ERDE_{N-1} C_i \alpha_N$, $\alpha_N = \alpha_{N-1} + \Sigma_i{}^{-1} X_{N-1}{}^T$

$c_N = c_{N-1} + ERDE_{N-1} X_{N-1} \sum_i{}^{-1} X_{N-1}$.

One can perceive that ERDE method, mounted in gClass, is tantamount with the one in eClass of [24]. Nevertheless, we dissent with this argument as three differences are at hand. We apply the different fuzzy rule exemplar with eClass and utilize the inverse multi-quadratic function in lieu of the Cauchy function. Apart from that, we reinforce the ERDE method with a weighting factor, to hamper a large pair-wise distance problem due to outliers [25]. The volume of the non axis-parallel ellipsoidal cluster can be concocted by the determinant operator. Nevertheless, it is a heuristic approach, so that it is less accurate. A volume of hyper-ellipsoidal cluster arbitrarily rotated in any positions can be quantified more accurately as follow:

$$V_i = \frac{2 * \prod_{j=1}^{u} (r_i / \lambda_{ij}) * \pi^{u/2}}{\Gamma(u/2)} \tag{11}$$

$$\Gamma(p) = \int_0^{\infty} x^{p-1} e^{-x} dx \tag{11b}$$

where r_i the Mahalanobis distance radius of the *i-th* fuzzy rule, which defines its (inner) contour (with default setting of 1) , λ_{ij} is the *j*-th eigenvalue of the *i-th* fuzzy rule and Γ is the gamma function. To expedite the execution of (11b), a look up table can be a priori generated and used during the training process. Conversely, the Bayesian concept is explored to accord the winning category, which is effective to grasp the winning rule owing to the prior probability, when two or more rules dwell an input zone, which is in the similar proximity to the datum. The posterior, prior probabilities as well as the likelihood function are mathematically illustrated respectively as follows:

$$\hat{P}(R_i|X) = \frac{\hat{p}(X|R_i)\hat{P}(R_i)}{\sum_{i=1}^{P} \hat{p}(X|R_i)\hat{P}(R_i)} \tag{12}$$

$$\hat{P}(R_i) = \frac{\log(N_{i,o}+1)}{\sum_{o=1}^{m} \log(N_{i,o}+1)} \tag{13}$$

$$\hat{P}(X|R_i) = \frac{1}{(2\pi)^{1/2} V_i^{1/2}} \exp(-(X-C_i)\Sigma_i^{-1}(X-C_i)^T) \tag{14}$$

where $N_{i,o}$ stands for the number of samples covered by *i-th* cluster falling in the *o-th* class. Note that, equation (13) is a refurbished version of prior probability formula of [26], in order to allow the newly born cluster to win the competition and to evolve its shape as such clusters are usually populated with a lower number of samples than the older clusters.

2) Rule base simplification : gClass relies on two rule pruning scenarios to appraise the fuzzy rule significances. On the one hand, The first method, namely the extended rule significance (ERS) theory is workable to infer the statistical contribution of the fuzzy rule and the fuzzy rule contribution with respect to the classifier's output. The fuzzy rule, holding a tiny influence zone and a small output parameters, can be eroded with a marginal leverage to the resultant classifier's output. On the other hand, the second method, namely potential+ (P+) theory, is a plausible approach, in order to seize the obsolete or outdated fuzzy rules. The P+ method scrutinizes the density of the cluster in accord with the data distribution, hence, the obsolete cluster can be written off due to a concept drift, when the potential is low. The ERS and P+ methods are respectively executed by the following equations.

$$\beta_i = \sum_{o=1}^{m} \sum_{j=1}^{2u+1} y_{ij}{}^o \frac{V_i{}^u}{\sum_{i=1}^{P} V_i{}^u} \tag{15}$$

$$\chi_i = \sqrt{\frac{(N-1)\chi_{n-1,i}{}^2}{(N-1)\chi_{n-1,i}{}^2 + (N-2)(1-\chi_{n-1,i}{}^2) + \chi_{n-1,i}{}^2 d_i{}^n}} \tag{16}$$

where β_i denotes the rule significance of ERS method and χ_i exhibits the rule sensitivity of P+ method, $d_i{}^n$ stands for the Mahalanobis distance between the current training sample and the focal point of interest. If the training observation complies either $\chi_i < \hat{\chi} - \chi_\sigma$ or $\beta_i < \hat{\beta} - \beta_\sigma$, the fuzzy rules can be pruned, where $\hat{\chi}, \chi_\sigma$ and $\hat{\beta}, \beta_\sigma$ stand for the mean and standard deviation of the rule significance and rule potential of existing rules. Nonetheless, if the fuzzy rules are pruned due to P+ method, they are just temporarily impounded and can be a subject of the rule recall mechanism. That is, such fuzzy rules can be noteworthy to the system in the future, as the old data distribution can be worked up as a causal relationship of the recurrent drift. It will be counterproductive to append a completely new fuzzy rule to spotlight this phenomenon, as the adaptation history granted to the pruned fuzzy rules in the earlier training episodes will be catastrophically dissolved. Accordingly, the rule recall mechanism is triggered, when the potentials of the pruned fuzzy rules are amplified by the up-to-date data distribution, leading to the following condition.

$$\max_{i*=1,..,P*} (\chi_{i*}) > \max_{i=1,..,P+1} (ERDE_i) \tag{17}$$

where P^* signifies the number of rules which are dispossessed by the P+ method. It is worth-stressing that this condition is synergized by the criteria of the fuzzy rule generation. In other words, the recall mechanism is sparked when an extraneous fuzzy rule is demanded. Note that, the fuzzy rules pruned by the P+ method are merely used to cultivate (16), where they are involved neither to stimulate other learning strategies nor to underpin the inference process. Therefore, the computational cost can still dwindle and the training process can be expedited. If the recall mechanism is concurred, the fuzzy rule is assigned as follow:

$$C_{P+1} = C_{i*}, \sum_{P+1}^{-1} = \Sigma_{i^*}^{-1}, \Psi_{P+1} = \Psi_{i^*}, \Omega_{P+1} = \Omega_{i^*} \quad (17)$$

In the viewpoint of the functionality, the P+ method is distinguishable with eClass [24], where the P+ method is revamped to work out the rule pruning purpose as with [27]. Nevertheless, we contend that our approach is different with the one in [27], where the P+ method is in line to facilitate the generalized fuzzy rule.

3) Initialization of new fuzzy rule parameters: This step is inherent with the so-called class overlapping stumbling block. Generally speaking, the new fuzzy rule should be posited in the feature space in such a way to be hedged with the fuzzy rules of a different class. As the new fuzzy rule can verge the fuzzy rule of a different class during its evolution, the new fuzzy rule should not be too imminent with another class fuzzy rule. Suresh et al in [1]-[5] offers a specific concept of the inter and intra class distance to circumvent the class overlapping problem. The main drawback of this method is yet an irrelevant assumption of the same class cluster. Obviously, the cluster can be occupied by the training samples of various classes in the real-life problem, undermining its efficacy to devastate this fact.

A new strategy is offerred by this paper, where, first of all, we vet the compatibility degree of the new rule with the winning cluster. If we encounter $R_{win} \geq \rho_2$, the new fuzzy rule is susceptible to be entrapped with the fuzzy rule redundancy. Instead of augmenting it as an extranrous fuzzy rule, it replaces the winning fuzzy rule, in order to eradicate the cluster redundancy.

$$i^* = \max_{i=1,..,P}(R_i), C_{i^*} = X_N \quad (18)$$

where ρ_2 is a predefined constant, that can be statistically represented by the critical value of a χ^2 distribution with p degrees of freedom and a significance level of α, termed a $\chi_p{}^2(\alpha)$. A typical value of α is 5%, and the degree of freedom is represented by the dimensionality of the learning problem, thus $p = k$. Therefore, we set $\rho_a = \exp(-\chi_p{}^2(\alpha))$. In contrast, if we confront $\max_{i=1,..,P}(R_i) < \rho_2$, we turn on the potential class cluster as follow:

$$ERDE_N{}^o = \sqrt{\frac{(N-1)}{(N-1)(a_n+1)+c_n-2b_n}} \quad (19)$$

$$a_n = \sum_{j=1}^{u+m}(x_j{}^N)^2 \ , \ c_n = c_{n-1} + \sum_{j=1}^{u+m}(x_j{}^{N-1})^2 \ , \ b_n = \sum_{j=1}^{u+m} x_j{}^N d_j{}^n$$

$$d_n = d_{n-1} + x_{N-1}$$

where N_o denotes the number of samples falling in the *o-th* class, whereas $x_j{}^N$ respectively stand for the latest received datum of the respective class. The crux of (19) is alike (10), however, we amend it to the per-class circumstance. An issue can come up, if the new fuzzy rule inclines to evolve to the different class cluster, where we land on the following condition.

$$\max_{o=1,..,m}(ERDE_N{}^o) \neq true\_class\_label \quad (20)$$

This condition implies that the latest datum is more adjacent to the different class cluster than the same class cluster. Notwithstanding warranted by $\max_{i=1,..,P}(R_i) < \rho_2$ where the new cluster is not redundant, the fuzzy region of the cluster is lessened, so as to minimize the class overlapping. That is, the fuzzy region may grow and may later on overlap with the different class cluster, thus decreasing the zone of influence as the coherent option to dodge the class overlapping. By extension, we apply (17), where we inhibit the size of the fuzzy region to culminate uncontrollably. As such, the cluster overlapping contingency can be axed notably as the new rule is allocated with a small initial volume as follow:

$$C_{p+1} = X_N \quad (21)$$

$$dist_j = k_1 \min_{i=1,..,p}(x_j - c_{i,j}), \sum_{p+1} = dist^T dist \quad (22)$$

where $k_1 \in [0.5, 0.9]$ is an overlapping factor. In this regard, we resort to boil down the radius of the new rule, therefore, the possibility of class overlapping can be minimized as aforementioned. In contrast, if the datum dwells the area nearby the data points in the same class, the overlapping factor k_1 is fixed as $k_1 = 1.1$ choosing a slightly bigger fuzzy region of the new rule than the actual distance of the neighboring rule.

Conversely, the winning rule is adjusted, when the datum possesses the minor difference with the existing fuzzy rule or the winning rule can accommodate the datum given an adaptation of the winning rule to modify its coverage span. Another worth-mentioning fact in refining the winning rule is whether this rule is allowable to extend its coverage with respect to (7), guiding to $V_{win} < \rho_1 \sum_{i=1}^{P} V_i$. Thereafter, the winning rule adaptation is conferred by the following equations.

$$C_{winner}{}^N = \frac{N_{win}{}^{N-1}}{N_{win}{}^{N-1}+1} C_{win}{}^{N-1} + \frac{(X_N - C_{win}{}^{N-1})}{N_{win}{}^{N-1}+1} \quad (23)$$

$$\Sigma_{win}(N)^{-1} = \frac{\Sigma_{win}(N-1)^{-1}}{1-\alpha} + \frac{\alpha}{1-\alpha} \frac{(\Sigma_{win}(N-1)^{-1}(X_N - C_{win}{}^{N-1}))(\Sigma_{win}(N-1)^{-1}(X_N - C_{win}{}^{N-1}))^T}{1+\alpha(X_N - C_{win}{}^{N-1})\Sigma_{win}(old)^{-1}(X_N - C_{win}{}^{N-1})^T} \quad (24)$$

$$N_{win}{}^N = N_{win}{}^{N-1} + 1 \quad (25)$$

where $\alpha = 1/(N_{win}{}^{N-1}+1)$ .Note that equation (24) is a desirable one to be cultivated on the fly as no re-inversion process is sought. In parallel to that, the output parameters of the new rule are set up as follow:

$$\Omega_{R+1} = \Omega_{winner} \quad (26)$$

$$\Psi_{R+1} = \omega I \quad (27)$$

where ω is a large positive constant. The setting of the output covariance matrix is a good one that can emulate the real solution yielded by the batch learning process when a

constant ω is selected as a large positive constant. Meanwhile, the weight vector of the new rule is enacted as the weight vector of the winning rule, as the winning rule is presumed to portray the pertinent data trend of the new rule. Obviously, the fundamental working principle of gClass adopts a local learning adaptation scheme, wherein the growing and pruning of the fuzzy rules impose a marginal impact of the stability and convergence of other local sub-systems.

4) Fuzzily Weighted Generalized Recursive Least Square (FWGRLS) Procedure: gClass is equipped by the FWGRLS method to polish up the rule consequent. This method is a local learning version of Generalized Recursive Least Square (GRLS) method of [28]. The prominent aspect of this method is capable of restraining a weight decay effect, thus boosting the generalization. The crux is to drive the weight vector to hower arround a small bounded range, which is also efficacious to evoke a compact rule base. As the inactive fuzzy rule possess a very small weight vector, it can be easily detected by the ERS method in equation (15). The FWGRLS method is formulated as follows:

$$\psi(n) = \Psi_i(n-1)F(n)(\frac{\lambda_i \Delta(n)}{\Lambda_i(n)} + F(n)\Psi_i(n-1)F^T(n))^{-1} \quad (28)$$

$$\Psi_i(n) = \Psi_i(n-1) - \psi(n)F(n)\Psi_i(n-1) \quad (29)$$

$$\Omega_i(n) = \Omega_i(n-1) - \varpi\Psi_i(n)\nabla\xi(\Omega_i(n-1)) + \Psi(n)(t(n) - y(n)) \quad (30)$$

$$y(n) = x_{en}\Omega_i(n)\,,\; F(n) = \frac{\partial y(n)}{\partial \Omega(n)} = x_{en} \quad (31)$$

where $\Lambda_i(n) \in \Re^{(P+1)\times(P\times 1)}$ indicates a diagonal matrix, whose diagonal element consists of the firing strength of fuzzy rule R_i and the covariance matrix of the modeling error is shown by $\Delta(n)$ which is managed as an identity matrix for the sake of simplicity and λ_i is a local forgetting factor, intended to ditch a detrimental effect of the concept drift, elaborated in the next sub-section of this paper. Meanwhile, ϖ is a predefined constant specified as $\varpi \approx 10^{-15}$ and $\nabla\xi(\Omega_i(n-1))$ stands for the gradient of the weight decay function. The weight decay function can be any non-linear function to which the exact solution of the gradient is unavailable. In consequence, it is expanded to the *n-1* time step, whenever the gradient information is inconvenient to be elicited. For our case, we make use of the quadratic weight decay function $\xi(y_i(n-1)) = \frac{1}{2}(\Omega_i(n-1))^2$ as it is capable of shrinking the weight vector proportionally to its current values. Note that the consequent adaptation is performed, when the criteria of the rule growing and the rule premise adjustment are satisfied.

5) Local drift handling strategy

A drift is more troublesome to be surmounted than a shift as the data distribution changes from one local region to another in a smooth way. As reciprocal leverage, the fuzzy rules are enforced to move more strongly in accordance with a drift rate, in order to trace the change of the data distribution. Otherwise, the fuzzy rules cannot chase the data distribution change, thus being alleged as a downtrend of the classification rate. This is mainly the case, if the conventional model update without a particular forgetting scheme adjusts the fuzzy rules. Some researchers resorts to deal with the drift in [29],[30], benefiting from the local or global drift handling technique. The former can be presumed as a more plausible one, as the drift can be managed locally. The drift can occur differently in each local region, so that equating the same forgetting for each local region can inflict a detrimental effect of stability and convergence.

We propose a novel local drift handling strategy of the so-called locally recursive density estimation (LRDE) method. The crux of this approach is akin to ERDE method, but we merely take into account the density of the local region (cluster). In a nutshell, LRDE method is formulated as follow:

$$LRDE_N{}^i = \sqrt{\frac{1}{1 + \sum_{n=1}^{N_i-1}(X_n - C_i)\Sigma_i^{-1}(X_n - C_i)^T}} = \sqrt{\frac{1}{1 + a_N - 2b_N + c_N}} \quad (32)$$

$a_N = a_{N-1} + X_n \sum_i^{-1} X_n{}^T$, $d_N = d_N + X_n \sum_i^{-1}$, $b_N = d_N C_i$ and $c_N = C_i \sum_i^{-1} C_i^T$

where N_i epitomizes the number of sample belonging to the *i-th* cluster. It is straightforward to discern that equation (32) from the time step *N* to *N-1* is equivalent with a change of the data distribution, envisaged as the drift intensity. Therefore, the rate of change or first order derivative of equation (32) can devolve a cue of the drift, whereby a forgetting mechanism should be governed accordingly. According to [30], a strong forgetting mechanism is relinquished by setting $\lambda_i = 0.9$, whereas no forgetting mechanism is signified with $\lambda_i = 1$. To assure the forgetting factor in the range of $\lambda_i \in [0.9,1]$, the following equations are exploited:

$$\lambda_i = 1 - 0.1\Delta LRDE_N{}^i \quad (33)$$

$$N_i = N_i - N_i \min(\lambda_{trans}, 0.99) \quad (34)$$

$$\lambda_{trans} = -9.9\lambda_i + 9.9 \quad (35)$$

One can conceive that an unique forgetting level of each rule is laid out by (32)-(34), so that a forgetting mechanism of a particular rule does not agitate other rules. Furthermore, equation (33) affects a more vigorous tuning of the rule consequent, alluded by equation (33). Meanwhile, the forgetting mechanism of the rule premise can be induced by (35), lessening the cluster population, overwhelming (23)-(25). Nonetheless, the fuzzy rules can gain a forgetting mechanism, if they have an adequate support, i.e., holding at least 30 loaded samples, unless, they can compel unlearning effect.

6) Input feature weighting technique

In this paper, the online feature weighting-based the separability criterion optimization in the empirical feature space is devised, which is extended from [31] in the offline case. It is worth-stressing that the optimization of the separability criterion in the feature weighting context can rectify the classification rate, as it essentially maximizes the between class distance, while minimizing within class distance. Our point of departure is the mathematical

expression of the FSC in the empirical feature space, between class scatter matrix and within class scatter matrix respectively as follows:

$$J = trace(S_w^{-1} S_b) = \frac{\Sigma W - \frac{\Sigma K}{N}}{tr(K) - \Sigma W} \tag{36}$$

$$tr(S_b) = \Sigma W - \frac{\Sigma K}{N} \tag{37}$$

$$tr(S_w) = tr(K) - \sum W \tag{38}$$

where ΣW signifies the sum of every dimension of matrix $\Sigma_{i,j} W$, K denotes a kernel-Gram-matrix to which ΣW and K are stipulated as follows:

$$W = \frac{1}{N} diag(K_{ii} / N_i), i = 1,...,m \tag{38}$$

$$K = \begin{bmatrix} K_{11}, K_{12},.., K_{1k},.., K_{1K} \\ K_{21}, K_{22},.., K_{2k},.., K_{2K} \\ ..... \\ K_{K1}, K_{K2},.., K_{Kk},.., K_{KK} \end{bmatrix} \tag{39}$$

Note that $K_{11} \in \Re^{N_1 \times N_1}$ demonstrates a kernel-Gram-sub-matrix emanating from data in class 1, whereas $K_{12} \in \Re^{N_1 \times N_2}$ constitutes a kernel-Gram-sub-matrix originating from data in classes 1 and 2 and so on. N indicates the total number of samples seen so far. The key idea lies on the recursive construction of the kernel-gram matrix by means of the Cauchy kernel as follows:

$$K_{o\hat{o}}{}^{N} = \frac{(N-1)}{(N-1)(\vartheta_N + 1) + \theta_N - 2\varsigma_N} \tag{40}$$

$$\vartheta_N = \sum_{j=1}^{u+m} (x_j(N)^o)^2 \quad , \quad \theta_N = \theta_{N-1} + \sum_{j=1}^{u+m} (x_j(N-1)^{\hat{o}})^2$$

$$\varsigma_N = \sum_{j=1}^{j+m} x_j(N)^o \nu_N \, , \nu_N = \nu_{N-1} + x_{N-1}{}^{\hat{o}}$$

where $x_j(N)^o$ is the *j-th* element of the *N-th* training sample falling into class o. θ_0 and ν_0 can be initialized as zero before the process ensues. Given that the kernel-gram matrix has been built upon the training observation, it allows to undertake the gradient ascent optimization scheme. To this end, the alignment matrix ought to be computed as follow:

$$A(K, K^*) = \frac{< K, K^* >_F}{\|K\|_F \|K^*\|_F} = \frac{tr(S_b)}{\|K\|_F} \tag{41}$$

where $\|K\|_F$ stands for the Frobenious norm of the kernel-Gram matrix K , taking the gradient of alignment matrix can produce the following expression, which is, in turn, utilized in the subsequent gradient ascent optimization procedure:

$$\partial_\theta (A(K, K^*)) = \frac{\partial_\theta tr(S_b)}{\|\partial_\theta K\|_F} = \frac{\Sigma(\partial_\theta W) - \frac{\Sigma \partial_\theta (K)}{N}}{\|\partial_\theta K\|_F} \tag{42}$$

$$\theta_N = \theta_{N-1} - \eta \partial_\theta (A(K, K^*)) \tag{43}$$

where θ_N stands for the weight vector which is initialized as 1. η exhibits the learning rate which decays overtime in the training process and is established according to [31] as follow:

$$\eta_N = \eta_0 (1 - {}^{n}\!/_{training\_samples}) \tag{44}$$

where η_0 is the initial learning rate with the default setting of 0.01 in accordance with [31] as the choice of initial learning rate, when the Gaussian kernel is employed. This choice of learning rate is coherent owing to the identical nature of Cauchy kernel with the Gaussian kernel. Meanwhile, the term *training_samples* means the total number of training samples to be learned in the training process. After identifying the input weight vector of the current training episode θ_N , it is integrated in the training and evolution engine of gClass: in all distance calculations, in all calculations involving solely the covariance or inverse covariance matrix as explained in [31]. It is worth-mentioning that gClass does not engage the Leave-One-Feature-Out (LOFO), which is always to be committed, in order to dispatch the input weights in [20], [31]. The LOFO approach is time-consuming to be employed, thus constraining the low computational cost of the classifier.

IV. EXPERIMENTATION

gClass is numerically validated with 3 artificial data streams, namely sin, circle and Boolean, acquired from the so-called diversity for dealing with drift (DDD) database of [32] and a electricity pricing dataset [1)] . All of 4 datasets explored herein characterizes the various concept drifts, which are the main challenge in learning in the dynamic and evolving environments. gClass is benchmarked with state-of-the-art classifiers, encompassing pClass [20], eClass [24], GENEFIS-class [19],. pClass, eClass and GENEFIS-class can be categorized as the evolving classifier, which constitutes a predecessor of the meta-cognitive classifier.

The empirical studies in 4 datasets are availed by the periodic hold-out experimental procedure as it can simulate the training and testing process in the real-time. The classification decision is drawn by the MIMO architecture for all consolidated algorithms, in order to underpin a fair comparison. The classifier performance is appraised in terms of the classification rate of a testing data block, the run time, the number of fuzzy rules flourished in the training process and the number of rule base parameters garnered in the memory. The predefined parameters of other algorithms are assigned according to the rules of thumb of the parameter selection presented in their original publications so as to goad the valid and fair comparisons with rClass. The numerical studies are sorted out by the intel (R) core (TM) i7-2600 CPU @3.4 GHz processor and 8 GB memory. Table 1 encapsulates the consolidated results of the benchmarked algorithms.

Referring to Table 1, gClass can deliver the most encouraging performances in all 4 criteria. gClass endures the most accurate classification rates in 4 study cases, showcasing 10-20% refinements in average with another classifier yielding the second best predictive accuracy. gClass proliferates more parsimonious rule bases than those of other benchmarked

[1)] http://moa.cms.waikato.ac.nz/datasets/

classifiers, evolving the smallest numbers of fuzzy rules in all 4 study cases. More interestingly, gClass can endure the lightest memory burdens, albeit coalescing with the generalized fuzzy rules, alleged to bear more expensive memory demands. That is, gClass deploys a more frugal number of fuzzy rules, thus alleviating the rule base parameters stored in the memory. Apart from that, gClass experiences swifter training processes than those of other algorithms in 3 out of 4 training processes, notwithstanding committing numerous learning maneuvers. This is affected by which the classifier is equipped by the what-to-learn-based active learning module, thus relieving the annotation efforts and diminishing the number of training samples.

V. CONCLUSION

This paper introduces a novel meta-cognitive learning machine termed Generic Classifier (gClass). The novelty of gClass lies on the design of the how-to-learn learning module, which is in line with the Schema and Scaffolding theories in the psychological literatures. A series of study cases via 4 data streams, omnipresent in the literatures to posses various concept drifts, has numerically validated the efficacy of gClass, where it can outperform its counterparts in poising the accuracy and simplicity. The future works will be devoted to render a more flexible rule base management and to endow the recurrent property of the gClass cognitive component.

REFERENCES

[1] K.Subramanian, S.Suresh, N.Sundararajan, " A Meta-Cognitive Neuro-Fuzzy Inference System (McFIS) for sequential classification systems", *IEEE Transactions on Fuzzy Systems,* on-line and in-press, (2013)

[2] G.S.Babu, S.Suresh, "Sequential Projection-Based Metacognitive Learning in a Radial Basis Function Network for Classification Problems", *IEEE Transactions on Neural Network and Learning Systems,* Vol.24(2), pp.194-206, (2013)

[3] G. Sateesh Babu and S. Suresh, "Meta-cognitive neural network for classification problems in a sequential learning framework," *Neurocomputing*, vol. 81, no. 1, pp. 86 – 96, (2012)

[4] S. Suresh, K. Dong, and H. Kim, "A sequential learning algorithm for self-adaptive resource allocation network classifier," *Neurocomputing*, vol. 73, no. 16-18, pp. 3012–3019, (2010)

[5] R. Savitha, S. Suresh, and N. Sundararajan, "Metacognitive learning in a fully complex-valued radial basis function neural network," *Neural Computation*, vol. 24, no. 5:1, pp. 297 – 328, (2012)

[6] R. Isaacson and F. Fujita, "Metacognitive knowledge monitoring and self-regulated learning: Academic success and reflection on learning," *Journal of the Scholarship of Teaching and Learning*, vol. 6, no. 1, pp.39–55, 2006.

[7] T.-O. Nelson and L. Narens, "Metamemory: A theoretical framework and new findings," Psychology of Learning and Motivation, vol. 26,no. C, pp. 125 – 173, (1990)

[8] J. Flavell, "Meta-cognition and cognitive monitoring: A new area of cognitive-developmental inquiry," *American Psychologist*, vol. 34(10), pp. 906 – 911, (1979)

[9] L.S.Vygotsky, *Mind and Society: The Development of Higher Psychological Processes,* Cambridge, U.K: Harvard University Press, (1978)

[10] B.J.Reiser, " Scaffolding complex learning: The mechanisms of structuring and problematizing student work", *Journal of Learning Sciences*, Vol.13(3), pp.273-304, (2004)

[11] D.Wood, " Scaffolding contingent tutoring and computer-based learning", *International Journal of Artificial Intelligence in Education,* vol.12(3), pp.280-292, (2001)

[12] J.HFlavell, "Piagiet's legacy", *Psychological Science,* vol.7.(4), pp.200-203, (1996)

[13] J.C.Patra, A.C.Kot, " Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks", *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics,* Vol.32(4), pp.505-511, (2002)

[14] Y-Y.Lin, J-Y.Chang, C-T.Lin, " Identification and prediction of dynamic systems using an interactively recurrent self-evolving fuzzy neural network", *IEEE Transactions on Neural Networks and Learning Systems,* Vol.24(2), pp.310-321, (2013)

[15] Y.H.Pao, " *Adaptive Pattern Recognition and Neural Networks*", Reading, MA: Addison-Wesley, 1989

[16] P. Angelov and X. Zhou, "Evolving fuzzy-rule-based classifiers from data streams," *IEEE Transactions on Fuzzy Systems*, vol.16(6), pp. 1462–1475, (2008)

[17] M.Pratama, M-J.Er, X.Li, R.J.Oentaryo, E.Lughofer, I.Arifin, "Data Driven Modelling Based on Dynamic Parsimonious Fuzzy Neural Network", *Neurocomputing*, Vol.110, pp.18-28, 2013, (2013)

[18] M.Pratama, S.Anavatti, P.Angelov, E.Lughofer, "PANFIS: A Novel Incremental Learning Machine", *IEEE Transactions on Neural Networks and Learning Systems,* online and in-press, http://dx.doi.org/10.1109/TNNLS.2013.2271933

[19] M.Pratama, S.Anavatti, E.Lughofer, " GENEFIS:Towards An Effective Localist Network", *IEEE Transactions on Fuzzy Systems, online and in press,* http://dx.doi.org/10.1109/TFUZZ.2013.2264938

[20] M.Pratama, S.Anavatti, E.Lughofer, pClass: An Effective Classifier to Streaming Examples, *Submitted to IEEE Transactions on Fuzzy Systems,* Under Review, 7th of June 2013

[21] A. Lemos, W. Caminhas and F. Gomide, Adaptive fault detection and diagnosis using an evolving fuzzy classifier, *Information Sciences*, vol. 220, pp. 64--85, (2013)

[22] S.L. Wang, K. Shafi, C. Lokan, and H.A. Abbass," Adversarial learning: the impact of statistical sample selection techniques on neural ensembles", *Evolving Systems*, vol.1(3),pp.181-197, (2010)

[23] E. Lughofer, Single-Pass Active Learning with Conflict and Ignorance, Evolving Systems, vol. 3 (4), pp. 251--271, 2012

[24] P. Angelov, E. Lughofer, and X. Zhou, "Evolving fuzzy classifiers using different model architectures," *Fuzzy Sets and Systems*, vol. 159(23), pp. 3160–3182, (2008)

[25] L.Wang, H-B.Ji, Y.Jin, " Fuzzy Passive-Aggressive Classification: A Robust and Efficient Algorithm for Online Classification Problems", *Information Sciences*, Vol.220,pp.46-63, (2013)

[26] B. Vigdor and B. Lerner, "The Bayesian ARTMAP," *IEEE Transactions on Neural Networks*, vol. 18(6), pp. 1628–1644, (2007)

[27] J-C.de Barros, A.L.Dexter, " On-line Identification of Computationally Undemanding Evolving Fuzzy Models", *Fuzzy Sets and Systems,* Vol.158, pp.1997-2012, (2007)

[28] Y.Xu, K.W.Wong, C.S.Leung, " Generalized Recursive Least Square to The Training of Neural Network", *IEEE Transactions on Neural Networks*, Vol.17 (1), (2006)

[29] A. Shaker and E. Lughofer," Resolving Global and Local Drifts in Data Stream Regression using Evolving Rule-Based Models", *in proceeding of IEEE Symposium Series on Computational intelligence,* Singapore, pp.9-16, (2013)

[30] E.Lughofer, P.Angelov, " Handling drifts and shifts in online data streams with evolving fuzzy systems", *Applied Soft Computing,* Vol.11(2), pp.2057-2068, (2011)

[31] H.Xiong, M.N.S.Swamy, M.O.Ahmad," Optimizing The Kernel in The Empirical Feature Space", *IEEE Transactions on Neural Networks*, Vol.16(2), pp.460-474, (2005)

[32] L.LMinku, X.Yao, " DDD: A New Ensemble Approach for Dealing with Drifts", *IEEE Transactions on Knowledge and Data Engineering,* Vol.24(4), 2012

[33] Nan-Ying Liang, Guang-Bin Huang, P.Saratchandran, N.Sundararajan, " A fast and accurate online sequential learning algorithm for feedforward networks", *IEEE Transactions on Neural Networks and Learning Systems,* Vol.17(6), pp.1411-1423, (2006)

[34] P. Angelov and E. Lughofer and X. Zhou, Evolving Fuzzy Classifiers using Different Model Architectures, *Fuzzy Sets and Systems*, vol. 159 (23), pp.3160--3182, 2008

[35] E. Lughofer, Evolving Fuzzy Systems --- Methodologies, Advanced Concepts and Applications, Springer, Berlin Heidelberg, 2011

[36] E. Lughofer and O. Buchtala, Reliable All-Pairs Evolving Fuzzy Classifiers, *IEEE Transactions on Fuzzy Systems*, vol. 21(4), pp. 625--641, 2013