Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007)

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

Frank Klawonn
Dept of Computer Science
University of Applied Sciences
BS/WF, Salzdahlumer Str. 46/48
D-38302 Wolfenbuettel, Germany

***Abstract*— A novel approach to on-line classification based on fuzzy rules with an open/evolving structure is introduced in this paper. This classifier can start 'from scratch', learning and adapting to the new data samples or from an initial rule-based classifier that can be updated based on the new information contained in the new samples. It is 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. Each prototype is a data sample that represents the focal point of a fuzzy rule per class and is selected based on the data density by an incremental and evolving procedure. This approach is transparent, linguistically interpretable, and applicable to both fully unsupervised and partially supervised learning. It has been validated by two well known benchmark problems and by real-life data in a parallel paper. The contributions of this paper are: i) introduction of the concept of evolving (open structure) classification (*eClass*) of streaming data; ii) experiments with well known benchmark classification problems (Iris and wine reproduction data sets).**

I. INTRODUCTION

CLASSIFICATION problems appear quite often in industrial systems, robotics, defence and are the basic tool used for pattern recognition tasks in signal and image processing, decision making, data mining, fault detection, automatic object recognition etc [1,2]. The basic problem in classification is to induce a classification function, or *classifier*, from a set of data samples. In many practical problems nowadays the data are produced in large quantity and very fast [3]. Such high-volume, non-stationary data streams bring new challenges to the well established learning methods [4]. In particular storing the complete data is often practically impossible and as a result the data streams cannot be analyzed in a batch mode. At the same time, most conventional learning methods, such as support vector machines [5,6], which aims at designing a classifier with guaranteed boundaries of the error, discriminant analysis [1], decision trees [7] or neural network classifiers [8], design the *classifier* in *batch* mode, that is, by using the complete data and labels that has been observed. Thus, they allow for extracting knowledge from a snapshot of the *data stream* at a certain point of time. Facing the challenge to cope with real-time classification of streaming data there is a need to develop classifiers that extract tractable *knowledge* from the data or digital signals in *real-time*.

More recently, learning classifiers [9] have been developed in the framework of evolutionary/genetic algorithms (EA/GA) and fuzzy rule-based systems that adapt their rule-base with new samples arriving on-line. They are, however, driven by a 'directed' random search according to the EA/GA concepts. Incremental Bayesian classifiers has also been reported which allow data samples to arrive one at a time, but the classifier structure is assumed to be fixed [10]. An evolving Bayesian classifier is under development and the preliminary results are reported in [11].

On the other hand, fuzzy rule-based systems that are evolving in the sense that their structure is not fixed, but can grow and shrink has been recently developed [12,29] and applied successfully to a number of identification [13], time-series prediction [14], fault detection [15], and control problems [16]. These systems are transparent and interpretable. We use the term '*evolving*' in a different context to the context when used in EA/GA. According to the Oxford Dictionary '*evolve*' means '*unfold; develop; be developed, naturally and gradually*' [17, p.294]. One can contrast this to the more general '*evolutionary*' [17, p.294] 'development of more complicated forms of life (plants, animals) from earlier and simpler forms', which is naturally related to the '*genetic*' [17, p.358] 'branch of biology dealing with the heredity, the ways in which characteristics are passed on from parents to offspring.' We use further the term '*evolving*' fuzzy classifier (*eClass*) in the sense of 'gradual development' of the classifier structure (fuzzy rule-base). This new paradigm introduced for neural networks in [18,19], for decision trees in [20], and for fuzzy rule-based systems in [21] can be regarded as a *higher level adaptation*. This emerging new paradigm mimics the evolution of **individuals** in nature during their life-cycle, specifically the autonomous mental development typical of humans: learning from experience, inheritance, *gradual* change, knowledge generation from routine operations, and rules extraction from the data. A trivial analogy is the way people learn during their life – starting with an empty rule-base they learn new rules during their life from experience and based on the data streams that their preceptors generate to the brain. The development of the rule-base is gradual, but the rules are not fixed or pre-defined. We generate new rules when new facts (data samples) that can not be described by the existing rules and when they are descriptive enough, not to be 'one-off'' outliers [22]. It is well known that fuzzy rule-based systems are universal function approximators [23]; they are suitable for extracting interpretable knowledge, therefore, they are

viewed as a promising framework for designing effective and powerful classifiers.

II. The Structure of the Proposed Classifier

The rule base that describes the non-linear evolving classifier *eClass* can be described as a set of fuzzy rules of the following form:

$$R_l^i : IF(x_1 \ is \ x_1^{i*}) \ AND...AND \ (x_n \ is \ x_n^{i*}) \ THEN\left(y_l^i = f^i\right) \quad (1)$$

where $x=[x_1,x_2,...,x_n]^T$ is the vector of features; R_l^i denotes the i^{th} fuzzy rule; *i=[1, Nl]; l=[1,L]; Nl* is the number of fuzzy rules *per cluster*; L is the number of classes (note that $L \le N$; $N=\sum_{l=1}^{L} N_l$, N is the overall number of fuzzy rules; that is there is at least one fuzzy rule per class); $\left(x_j \ is \ x_j^{i*}\right)$ 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; $y^i=[y_1^i,y_2^i,...,y_l^i]$ is the L-dimensional binary output of the MIMO exTS [22] fuzzy system.

Note that the type of the fuzzy rule depends on the type of the consequent [22]:

a) first order Takagi-Sugeno, TS (the consequents are linear classifiers):

$$f^i=\left[1,x^T\right]\begin{bmatrix}\alpha_{01}^i & \alpha_{02}^i & \dots & \alpha_{0l}^i\\ \alpha_{11}^i & \alpha_{12}^i & \dots & \alpha_{1l}^i\\ \dots & \dots & \dots & \dots\\ \alpha_{n1}^i & \alpha_{n2}^i & \dots & \alpha_{nl}^i\end{bmatrix}^T \quad (1a)$$

b) zero order TS (the consequents are the class labels):

$$f^i=\begin{bmatrix}\alpha_{01}^j & \alpha_{02}^j & \alpha_{0l}^j\end{bmatrix}^T \quad (1b)$$

The output of the exTS, *y* is formed as [22]:

$$y_l=\sum_{i=1}^{N}\frac{\prod_{j=1}^{n}\mu_{jl}^i(x_j)}{\sum_{j=1}^{N}\prod_{j=1}^{n}\mu_{jl}^i(x_j)}y_l^i \quad (1c)$$

where μ_{jl}^i is the membership value of the j^{th} feature, *j=[1,n]; l=[1,L].*

We use the so called 'winner-takes-all' de-fuzzification to determine the correct class, which is the usual choice in classification problems:

$$Class=\arg\max_{l=1}^{L}(y_l) \quad (2)$$

The membership functions that describe the closeness to the prototype can be of any form, but usually the Gaussian bell function is preferred due to its generalization capabilities:

$$\mu_{jl}^i = e^{-\frac{1}{2}\left(\frac{d_{jl}^i}{\sigma_{jl}^i}\right)^2} \quad i=[1,N_l]; \ j=[1,n]; \ l=[1,L] \quad (3)$$

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

Note that in order to simplify notations further the index *l* will be omitted and a remark will be made that calculations are made *per class*.

The spread of the membership σ is determined based on the scatter [26] of the data per cluster. In Figure 1 both, so called 'one sigma' and 'two sigma' zones, are given around each prototype (each fuzzy rule) for the IRIS data set using Euclidean distance. In this particular example there are two prototypes (respectively, two fuzzy rules) for one of the classes and one rule for the other two classes.

The scatter resembles standard deviation and is given by [22]:

$$\sigma_l(k)=\sqrt{\frac{1}{S_l(k)}\sum_{j=1}^{S_l(k)}d_{\cos}(x_{,}^{i*}x_j)^2}; \ \sigma_l(0)=1 \quad (4)$$

where *l=[1,L]* is the number of clusters; $d_{\cos}(x_{,}^{i*}x_j)$ denotes the distance from cluster centre to new sample assigned into this cluster.

The scatter can be updated recursively by:

$$\left[\sigma_l(k)\right]^2=\left[\sigma_l(k-1)\right]^2+\frac{1}{S_l(k)}\left[d^2\left(x_{,}^{i*},x(k)\right)-\left[\sigma_l(k-1)\right]^2\right] \quad (5)$$

When a new cluster/rule is formed, $N_l \leftarrow N_l+1$, its initial local scatter [26] is approximated by the average of the local scatters for the existing fuzzy rules *for that class* [22]:

$$\sigma^{N_l+1}(k)=\frac{1}{N_l}\sum_{i=1}^{N_l}\sigma_k^i \quad (6)$$

The structure of the proposed classifier is thus formed by sets of fuzzy rules of type (1) in such a way that there is at least one fuzzy rule per class. The prototypes around which the fuzzy rules are formed are samples selected from the available data by unsupervised learning (*eClustering*). The learning of the exTS fuzzy model is described in detail in [22].

III. Evolving Classifiers from Data Streams

A. Potential of the prototypes and its recursive calculation

The *eClass* is initialized with the first data samples. A fuzzy rule is formed around each one of these samples *per class*. Potential of the prototype is set to $P_1^j(x^{1*})=1$. Consequently, each data sample is first being classified to one of the classes defined in the classifier and then its suitability to become a prototype (to form another fuzzy rule) has been checked. The decision whether a data sample is used to form a prototype or to replace an existing prototype is based on the

data spatial density measure, called *potential* [13,24].

The potential calculated for a data sample is a function of the accumulated distance between this sample and all other samples in the data space *per class*. Thus, it represents the *density* of the data that surrounds a certain data sample. Originally [21,24] using Euclidean distance:

$$P_{kl}(x(k))=\frac{1}{1+\left(\sum_{j=1}^{n}\sum_{i=1}^{S_l(k)-1}\left\|x_i-x(k)\right\|_j^2\right)/\left(S_l(k)-1\right)};k=2,3,.. \quad (7)$$

where Pk(xk) denotes the potential of the kth data sample, xk.

Note that formula (7) requires accumulating the information from history of all the data which obviously contradicts the requirement for real-time and on-line application. The derivation of the recursive version of the potential expression (7) is given in [13]. Starting from substituting (4) into (7) and performing certain manipulations that are given in the Appendix we arrive at:

$$P_k(x(k))=\frac{S_l(k)-1}{\alpha(k)\left(S_l(k)-1\right)+\beta(k)-2\gamma(k)+(k-1)} \quad (8)$$

where $\alpha(k)=\sum_{j=1}^{n}\left(x^j(k)\right)^2$ (9a)

$$\beta_k=\sum_{i=1}^{S_l(k)-1}\sum_{j=1}^{n}\left(x^j(i)\right)^2 \quad (9b)$$

$$\gamma(k)=\sum_{j=1}^{n}x^j(k)\Gamma^j(k) \quad (9c)$$

$$\Gamma^j(k)=\sum_{i=1}^{S_l(k)-1}x^j(i) \quad (9d)$$

In this equation, values $\alpha(k)$ and $\gamma(k)$ can easily be calculated based on the availability of the current data point, *x(k)* only. The values $\beta(k)$ and $\Gamma^j(k)$ that require accumulation of past information can be easily stored in two variables with small size (the scalar, $\beta(k-1)$ and the *n*-dimensional vector-column $\Gamma(k)=\left(\Gamma^1(k),\Gamma^2(k),..,\Gamma^n(k)\right)^T$). Then one can calculate recursively $\beta(k)$ and $\Gamma^j(k)$ by [13]:

$$\beta(k)=\beta(k-1)+\alpha(k-1)$$
$$\beta(1)=0 \quad (10a)$$

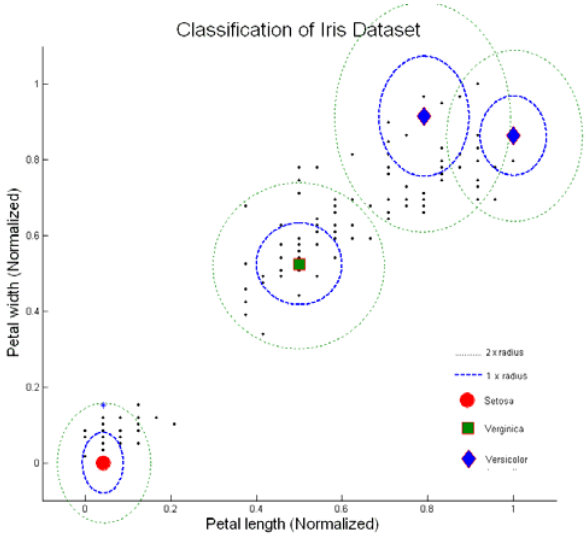

Fig. 1 Classification for Iris problem: a snapshot illustrating four prototypes (two for the class Versicolor and one for each of the remaining classes); σ are different for different clusters on different dimensions.

$$\Gamma^j(k)=\Gamma^j(k-1)+z^j(k-1)$$
$$\Gamma^j(1)=0 \quad (10b)$$

Each time a new data sample is read it affects the data density of the data space of the *respective class*, therefore the potentials of the previous centre needs to be updated. This update can also be done in a recursive way as detailed in [13], and no extra variable needs to be memorized, apart from the current potential of the existing prototypes (focal points):

$$P_k(x^*(k))=\frac{\left(S_l(k)-1\right)P_{k-1}\left(x^*(k)\right)}{S_l(k)-2+P_{k-1}\left(x^*(k)\right)+P_{k-1}\left(x^*(k)\right)\sum_{j=1}^{n}\left\|x^*(k)-x(k)\right\|_j^2} \quad (11)$$

B. *eClass Procedure*

The proposed evolving fuzzy rule-based classifier can start either '*from scratch*' (with an empty rule-base) or with some pre-specified set of fuzzy rules in form of (1). Each new data sample that has been read can be used to upgrade or modify the rule base if the label is also provided. We call this mode *D* (for Design) and we apply it in on-line mode (possibly in real-time). If the label is not provided, the existing fuzzy rule base will generate the predicted class, that is, it will work in mode *C* (for Classification). Note that *eClass* can work in any combination of these two modes. For example, one can perform *D* for certain number of samples and afterwards perform *C* for another set of samples. This is close to the off-line classifiers design concept. One can also perform *D* and *C* for the same sample (performing C first and using the rule-base that existed before this sample was read and only after the classification to perform *D* if the class label is provided). This is close to the adaptive modelling and control concept when the prediction and learning are combined in the same time step. An important specific of *eClass* is that not only the number of fuzzy rules but the number of classes, *L* may also be evolving and does not need to be pre-fixed. This specific is not used in this paper, because the two benchmark

problems that have been considered have a pre-specified and fixed number of features. This characteristic of *eClass* has been, however, used in a real-life example [27].

When *eClass* is in *D* mode from the second data sample onwards its *potential*, $P_k(x(k))$ is updated recursively by (8)-(9). Then the potential of each of the previously existing prototypes, $P_k\left(x^*(k)\right)$ is also updated using (11). Comparing the potential of the new data sample with the potential of each of the existing prototypes the following three outcomes are possible:

a) $P_k\left(x(k)\right) > \max_{i=1}^{R_i} P_k\left(x^{i*}(k)\right)$ (12)

If condition (12) occurs that means that we have a data sample that is strongly representative [21,22] and thus we add a new prototype to the rule base. For each newly added prototype we form a new fuzzy rule based at that prototype as a focal point. Additionally, we check whether any of the already existing prototypes are described *well* by the newly added fuzzy rule. By *well* we mean [22] that the value of the membership function satisfies:

$$\exists i, i=[1,N];\quad \mu_{il}^{j}\left(x(k)\right) > e^{-1} \quad \forall j, j=[1,n]; l=[1,L] \quad (13)$$

For the previously existing prototypes for which this is the case, we remove them.

Alternatively (if (12) does not hold), we do not change the overall structure of the classifier.

The procedure for the evolving fuzzy rule based classifier *eClass* when it applies a joint classification and classifier design *(C+D)* can be summarised in the following pseudo-code:

```
BEGIN eClass (C + D)
  Initialize (get first data sample or
  start from a pre-trained classifier);
  DO for the pair xk, k=2,3,… WHILE data
  stream ends
    Classify xk (assign it to a class, C^l);
    Calculate Pk(xk) using (8)-(9);
    Update P(x*) using (11)
    Calculate the potential difference
    IF (12) THEN add a new fuzzy rule
    around xk;
    IF (11) THEN remove the prototype(s),
    x^i* for which it is true;
    Get the real class label, C^l
  END DO
```

The flow chart (Figure 2) describes the structure of eClass as a server thread in real-time application implementations.

IV. Experimental Results

The proposed classifier, *eClass* was tested on two well known benchmark problems. 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. In a parallel paper [27], a real-life application for image data on-line classification is presented. Results on these two datasets are also compared with the results from the approach 'Incremental Principle Component Analysis' [28].

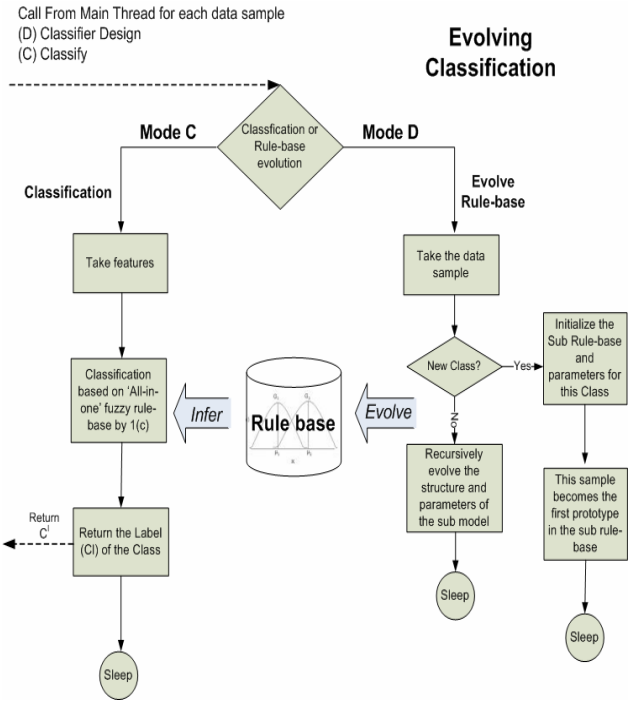

Fig. 2 Flow chart of Evolving Classification for real-time applications.

A. Iris Data set

The Iris data set is a widely used benchmark for classification studies. The data set has three classes that represent three types of the Iris plants, namely Iris Setosa, Iris Verginica, and Iris Versicolor. There are four features of the plants that are available for all the samples in the data set. They represent the sepal and petal lengths and widths in cm. The data set consists of 150 samples that is, 50 for each species. In [25] 19 different classifiers were reviewed, and all of them give between three and 24 misclassifications. The results when using *eClass* are tabulated in Table I. When comparing the result given by *eClass* with these result, one should take into account the fact that *eClass* starts in this experiment '*from scratch*' with an ***empty*** rule base and no pre-training and evolves its structure from the data stream.

In our experiment, tests have been carried out in three different ways. In the first test we classify each sample first without knowing its label. Afterwards, we get its correct class label and use the pair (sample and label) to evolve the rule-base. The joint classification and learning process *(C + D)* is continued during the whole process of 150 samples. The samples are in the original order as presented in the UCI repository. In the second test, 100 randomly ordered samples are used to evolve the classifier (*D* mode), and the rest 50 are used to test (*C* mode). In the last test, so called *'leave one out'* strategy is used that is *eClass* runs 150 times, each time

having 149 samples for classifier design (*D*) and the remaining single sample is used for classification.

B. Wine Reproduction Data

Wine Reproduction data set is another common benchmark problem. The data set comes from the chemical analysis of wines grown in Italy derived from three different cultivars. There are three classes, 178 samples with thirteen continuous numerical features available in the data set.

In a similar way the three experiments have been carried out with this data set. In the third experiment, 80 randomly organized samples are used as training data to evolve the rule base, and the remaining 98 data samples are used to test the fixed classifier. All the results of testing *eClass* with Wine data set are tabulated in Table II.

TABLE I
RESULTS FOR IRIS PROBLEM

	Design	Classify	Mode	Rules	Rate
I	150	150	(C&D)x150	4	98.0%
II	100	50	100D+50C	4	96.0%
III	149 (x 150)	1 (x 150)	149D+1C (x150)	4	99.3%
iPCA [28]	178	178	-	2 eigen vectors	93.3%

Both tests on Iris data and Wine data shows that the proposed classifier *eClass* has 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 other ***off-line*** approaches reported in [25]. In its offline mode, *eClass* has results of the same level of precision as as the best of the classifiers reported in [25].

V. CONCLUSION

A novel approach, *eClass*, to on-line classification is introduced in this paper. It stems from subtractive and Mountain clustering [24]. It works in online mode and can work in real-time. It is non-iterative, and recursive calculations are used which avoids memorizing the whole history of the data without loss of information about the data density. This enables low memory requirements and much less computation. The learning process can start 'from scratch' learning and adapting to the new data samples. It can also start from some initial rule-based classifier that can be updated based on the new information contained in the new samples. Thus it is suitable for on-line and real-time applications such as classification of signals and streaming data, robotic applications, e.g. target and landmark recognition, real-time machine health monitoring and prognostics, fault detection and diagnostics etc. It can work for classification or evolvable classifier design only or for joint classification and classifier design. A novel formula for recursive calculation of the cosine distance is introduced in the paper that is suitable for on-line and real-time applications. Experiments with well known benchmark classification problems (Iris and wine data sets) have been performed and the results are very promising. The results indicate that *eClass* can achieve high classification rate (comparable or better than the off-line pre-fixed classifiers) as reported in the literature for two benchmark problems. At the same time it has a transparent form and can accommodate new data samples integrating the new data and existing knowledge.

TABLE II
RESULTS FOR WINE PROBLEM

	Design	Classify	Mode	Rules	Rate
I	178	178	(C&D)x178	7	96.1%
II	80	98	80D+98C	7	95.9%
III	177 (x 150)	1 (x 150)	177D+1C (x150)	7	98.7%
iPCA [28]	178	178	-	7 eigen vectors	87.6%

REFERENCES

[1] R. O. Duda, P. E. Hart, and D.G. Stork. "*Pattern Classification" - Second Edition*. Wiley-Interscience, Chichester, West Sussex, England, 2000.

[2] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From Data Mining to Knowledge Discovery: An Overview, Advances in Knowledge Discovery and Data Mining, MIT Press, Cambridge, Massachusetts, USA, 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, USA, 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 Verlag, Berlin, Germany, 1998.

[6] V. Kecman. *Learning and Soft Computing*. MIT Press, Cambridge, Massachusetts, London, England, 2001.

[7] J. R. Quinlan, "Simplifying decision trees", *International Journal of Man-Machine Studies*, vol. 27, No3, pp.221–234, 1987.

[8] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.

[9] 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.

[10] K. M. A. Chai, H. T. Ng, and H. L. Chieu, "Bayesian Online Classifiers for Text Classification and Filtering", *Proc.of the SIGIR'02*, August 11-15, 2002, pp. 97-104, Tampere, Finland.

[11] F. Klawonn and P. Angelov, Evolving Extended Naive Bayesian Classifier, *2006 IEEE Intern. Conf. on Data Mining*, 18-22 December, 2006, Hong Kong, to appear

[12] P. Angelov, *Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems*. Heidelberg, Germany: Springer Verlag, 2002.

[13] P. Angelov and D. Filev, "An approach to on-line identification of evolving Takagi-Sugeno models*", IEEE Transactions on Systems, Man and Cybernetics, part B, Cybernetics,* vol.34, No1, pp. 484-498, 2004.

[14] P. Angelov and R. Buswell, "Identification of Evolving Rule-based Models", *IEEE Transactions on Fuzzy Systems*, vol.10, No5, pp.667-677, 2002.

[15] P. Angelov, V. Giglio, C. Guardiola, E. Lughofer, and J. M. Lujan, "An Approach to Model-based Fault Detection in In- dustrial Measurement Systems with Application to Engine Test Benches", *Measurement Science and Technology*, vol.17, No7, 2006, pp.1809-1818.

[16] P. Angelov**, "**A Fuzzy Controller with Evolving Structure", *Information Sciences,* ISSN 0020-0255, vol.161, 2004, pp.21-35.

[17] A. S. Hornby, *Oxford Advance Learner's Dictionary*, Oxford University Press, UK, 1974.

[18] B. Fritzke, "Growing cell structures – a self-organizing network for unsupervised and supervised learning", *Neural Networks*, vol. 7, No 9, pp.1441-1460, 1994.

[19] N. Kasabov, "Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge-based learning," *IEEE Transactions on Systems, Man, and cybernetics, part B – Cybernetics,* vol. 31, pp. 902-918, 2001.

[20] R. Jin and G. Agrawal, "Efficient Decision Tree Construction on Streaming Data", *Proc. of ACM SIGKDD*, 2003.

[21] P. Angelov and R. Buswell, Evolving Rule-based Models: A Tool for Intelligent Adaptation, *Proc. of the 9th IFSA World Congress*, Vancouver, BC, Canada, 25-28 July 2001, pp.1062-1067.

[22] P. Angelov and 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, September 2006, ISBN 0-7803-9719-3.

[23] L.-X. Wang, "Fuzzy Systems are Universal Approximators", *Proc. 1st IEEE International Conference on Fuzzy Systems, FUZZ-IEEE,* San Diego, CA, USA, pp.1163-1170, 1992.

[24] R. R. Yager and D. P. Filev, "Learning of Fuzzy Rules by Mountain Clustering," *Proc. of SPIE Conference on Application of Fuzzy Logic Technology,* Boston, MA, USA, pp.246-254,1993.

[25] H. Ishibuchi and T. Nakashima, "Voting in fuzzy rule-based systems for pattern classification problems", *Fuzzy Sets and Systems*, vol. 103, pp.223-238, 1999.

[26] P. Angelov and D. Filev, "Simpl_eTS: A Simplified Method for Learning Evolving Takagi-Sugeno Fuzzy Models", *The 2005 IEEE International Conference on Fuzzy Systems FUZZ-IEEE*, Reno, Las Vegas, USA, 22-25 May 2005, pp.1068-1073, ISSN 0-7803-9158-6/05.

[27] X. Zhou and P. Angelov, "Autonomous Self-localization in Completely Unknown Environment using Evolving Fuzzy Rule-based Classifier", *1st IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2007*, April 1-5, 2007 Honolulu, HI, USA, to appear.

[28] S.Pang, S.Ozawa, and N.Kasabov, "Incremental Linear Discriminant Analysis for Classification of Data Streams", IEEE Transactions on Systems, Man, And Cybernetics –Part B: Cybernetics, Vol35, No.5, October 2005.

[29] E. Lughofer and E. Klement, "FLEXFIS: A variant for incremental learning of Takagi-Sugeno fuzzy systems," in *Proceedings of FUZZ-IEEE 2005, Reno, Nevada,* U.S.A., 2005.