All-Pairs Evolving Fuzzy Classifiers for On-line Multi-Class Classification Problems *

Edwin Lughofer[1]

[1]Department of Knowledge-Based Mathematical Systems, Johannes Kepler University, A-4040 Linz, Austria

Abstract

In this paper, we propose a novel design of evolving fuzzy classifiers in case of multi-class classification problems. Therefore, we exploit the concept of all-pairs aka all-versus-all classification using binary classifiers for each pair of classes, which has some advantages over direct multi-class as well as one-versus-rest classification variants. Regression-based as well as singleton class label fuzzy classifiers are used as architectures for the binary classifiers, which are evolved and incrementally trained based on the concepts included in the FLEXFIS family (a connection of eVQ and recursive fuzzily weighted least squares). The classification phase considers the preference levels of each pair of classes stored in a preference relation matrix and uses a weighted voting scheme of preference levels, including reliability aspects. The advantage of the new evolving fuzzy classifier concept over single model (using direct multi-class classification concept) and multi model (using one-versus-rest classification concept) architectures will be underlined by empirical evaluations and comparisons at the end of the paper based on high-dimensional real-world multi-class classification problems.

Keywords: multi-class problems, all-pairs classification, evolving fuzzy classifiers, preference relation matrix, reliability

1. Introduction

1.1. Motivation and State-of-the-Art

Evolving classifiers are serving as powerful data-driven design tool in today's real-world decision support and classification systems in order to cope with on-the-fly modeling scenarios in on-line environments and building models from huge data bases which cannot be loaded at once into the memory. Single-pass incremental learning capability, where models are built up in a step-wise fashion by using single data samples or blocks of data [13] [1], plays an important role in evolving classifiers in order to keep the update time and the virtual memory usage at a low level [22]. Evolving classifiers are not only able to adapt their parameters, but also to extend their structures and expand their memory on-the-fly in order to account for varying, expanding system states/behaviors. A specific type of evolving classifiers are *evolving fuzzy classifiers*[22], which exploit the concepts of fuzzy sets in order to account for a possibilistic evolving modeling approach for any uncertainty in the (classification) data. Furthermore, as using a non-linear fuzzy classifier structure in rule-based form, they are able to represent classifiers with high accuracy [15], to express reliability in a natural way [11] and to allow some sort of interpretability [5].

In [4], evolving fuzzy classifiers are demonstrated which exploit single and multi model architectures (SM and MM) and whose learning engines for the rule evolution and antecedent learning part rely on the concepts of *FLEXFIS* [20] and *eTS* [3]. The SM variant is based on the classical fuzzy classification models using singleton consequent class labels, the MM variant uses K Takagi-Sugeno fuzzy models for the K classes based on indicator entries in the feature vectors. *FLEXFIS-Class MM* is successfully applied as evolving image classifiers for surface inspection purposes in [21] and extended in [23] to account for dynamic soft dimensionality reduction. The approach in [2] uses an extended fuzzy classifier structure where each rule consequent part captures the representativeness degree of the corresponding rule to each class (in this sense, each rule maps to all classes, at least to some extent). In [16], and evolving fuzzy classifier approach is proposed which is trained based on the concepts of *ePL = evolving Participatory Learning* [17], deduced from R. Yager's participatory learning concept [28], and uses a classical classifier architecture with consequent labels, where the classes represent different operation modes which may arise at fault diagnosis systems. In case of multi-class (polychotomous) classification scenarios (i.e. the number of classes into where each instance may fall is greater than 2), the above variants perform either a direct mapping of new samples to any of the K classes (through one or several rules) or they are using a one-versus-rest classification approach (in case of multi-model variant). The later tries to find (binary) classifiers for discriminating each class from all other classes in one sweep (achieving K classifiers for K classes). Usually, the maximal supported classes from all classifiers is taken as overall classification response.

*This work was funded by the Austrian fund for promoting scientific research (FWF, contract number I328-N23, acronym IREFS). This publication reflects only the authors' views.

1.2. Our Approach

In this paper, we are going beyond the state-the-art by using the concept of all-pairs (aka one-versus-one) classification technique within the scope of evolving fuzzy classifiers. All-pairs classification is characterized by setting up classifiers for each pair of classes, achieving $m(m-1)/2$ classifiers or $m(m-1)$ classifiers in case of ordered pairs of classes. The output of each binary classifier, e.g. for class k and l, represents the preference degree of class k over l, spanning up a (upper right) preference relation matrix, see Section 2.1 for a detailed description. One reason for the choice of this classifier architecture is that in [7] or [24] it could be shown that it is able to out-perform one-versus-rest as well as direct multi-class classification in terms of classification accuracy. Another reason is the more efficiency regarding computation times in the (incremental) training phase (see Section 2.2). Furthermore, we will introduce a novel concept how to integrate the reliability of the binary classifiers into the classification stage of the *EFC-AP* approach. This is done by exploiting the concepts of conflict and ignorance [10] [12] and adopting their meaning to the all-pairs classifier structure (Section 4) by calculating ignorance and conflict degrees according to the structure used in the binary classifiers and integrating these degrees when producing the final classification response based on the preference relation matrix (note that a variant of such a combination is presented in [26] with the help of belief functions for batch off-line multi-class classification tasks). Based on several high-dimensional multi-class data sets (5 from the UCI repository, 2 real-world from industrial processes in own projects), it will be empirically verified that 1.) *EFC-AP = Evolving Fuzzy Classifiers using All Pairs technique* can in fact out-perform *EFC-MM = EFC one-versus-rest* and *EFC-SM = EFC direct mapping* in terms of accuracy and computation time and 2.) improve accuracy further when taking into account the reliability in the final classification decision (Section 5).

2. Classifier Structure

2.1. Definition

The classifier structure of all-pairs learning in multi-class classification scenarios is based on a decomposition of the whole problem into several binary sub-problems. Formally, this can be expressed by a classifier $\mathbb{C}_{k,l}$ which is induced by a training procedure $\mathbb{T}_{i,j}$ when using (only) the class samples falling into classes k and l:

$$\mathbb{C}_{k,l} \longleftarrow \mathbb{T}_{k,l}(X_{k,l}) \quad X_{k,l} = \{\vec{x} | L(\vec{x}) = k \vee L(\vec{x}) = l\} \tag{1}$$

with $L(\vec{x})$ the class label associated with feature vector $\vec{x}$. This means that $\mathbb{C}_{k,l}$ is a classifier for separating samples belonging to class k from those belonging to class l. Note that we do not handle any single-class classifiers (e.g. for representing convex hulls etc.), therefore we only obtain $\mathbb{C}_{k,l}$ for $k \neq l$. This means that in sum we have to train $K(K-1)$ binary classifiers with K the number of classes in order to obtain a full representation of class relations. When classifying a new sample $\vec{x}$, each classifier outputs a confidence level $conf_{k,l}$ which denotes the degree of preference of class k over class l for this sample. This degree lies in $[0,1]$ where 0 means no preference, i.e. a crisp vote for class l and 1 means a full preference, i.e. a crisp vote for class k. This is conducted for each pair of classes and stored into a preference relation matrix R (assuming to have K classes in the classification problem):

$$R = \begin{bmatrix} 1 & conf_{1,2} & conf_{1,3} & ... & conf_{1,K} \\ conf_{2,1} & 1 & conf_{2,3} & ... & conf_{2,K} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ conf_{K,1} & conf_{K,2} & conf_{K,3} & ... & 1 \end{bmatrix}$$

If we assume reciprocal preferences, i.e. $conf_{k,l} = 1 - conf_{l,k}$, then we can omit the training of half of the classifiers, hence finally obtain $\frac{K(K-1)}{2}$ binary classifiers.

The choice of the model architecture for each binary classifier depends strongly on the concept to be learned. We are dealing with fuzzy classifiers, as they can resolve any degree of non-linearity in the decision boundary between classes and also provide a natural and meaningful way how to represent conflict and ignorance in the preference relation (this will be handled in more detail in Section 4). In this paper, we are concentrating on two fuzzy classification architectures, singleton class labels and regression-based classifiers based on Takagi-Sugeno type models (note the former is used as direct multi-class classification architectures in *FLEXFIS-Class SM* and *eClassA* approach [4], the later in the one-versus-rest classification scheme as exploited by multi-model evolving fuzzy classifiers *FLEXFIS-Class MM* and *eClassM* [4], allowing a direct comparison of novel all-pairs classifier structure in EFC with former used structures possible — see Section 5).

The ith rule of a binary fuzzy classifier for class pair (k,l) with singleton class labels is defined in the following way:

$$\text{Rule}_i: \quad \text{IF } x_1 \text{ IS } \mu_{i1} \text{ AND...AND } x_p \text{ IS } \mu_{ip} \quad \text{THEN } l_i = L_i \tag{2}$$

with μ_{ij} the jth membership function (fuzzy set) in the jth antecedent part of the ith rule, p the dimensionality of the feature space and AND a conjunction operator, usually represented by a t-norm [14]. L_i is the crisp output class label from the set of two classes (k and l). In case of regression-based classifiers, regression is conducted on $\{0,1\}$ for each class pair (k,l), where 0 belongs to class k and 1 to

class l using the Takagi-Sugeno model architecture, where the ith rule is defined by:

$$\text{Rule}_i: \quad \text{IF } x_1 \text{ IS } \mu_{i1} \text{ AND...AND } x_p \text{ IS } \mu_{ip} \\ \text{THEN } l_i = w_{i0} + w_{i1}x_1 + ... + w_{ip}x_p \quad (3)$$

This means that the TS model type represents a mixture of completely linguistic antecedent part and functional consequent parts (in form of hyperplanes), achieving a tradeoff between accuracy and linguistic interpretability. The rules are combined by a weighted inference scheme to produce a regression output value $[0, 1]$, upon which a binary classification statement can be obtained, see Section 4.

2.2. Comparison with other Structures

The likelihood of balanced learning problems is higher in case of all-pairs fuzzy classifiers, compared to the one-versus-rest classification scheme based on indicator entries, where K TS models for K classes using the whole data set are trained. This is because when training the kth model for the kth class, all samples not belonging to this class are assigned the value of 0, whereas the samples falling into class k are assigned to 1. This means that for the usual classes (classes with not an extraordinary high number of representatives) much more regression target values are 0 than 1. On the other hand, imbalanced learning problems often cause significant bias towards under-represented classes [9] (no matter if used in batch or incremental training mode). In [7] [6] it was shown that pairwise classification is not only more accurate than one-versus-rest technique, but also more efficient regarding computation times. The reason for this is basically that binary classification problems contain significantly lower number of samples, as each sub-problem uses only a small subset of samples (belonging to class k and l for the class pair (k, l)). Often, the training of a fuzzy classifier requires a polynomial, at least a quadratic complexity with the number of samples $O(N^2)$: in case of n selected samples $n << N$ this leads to a complexity of $O(\frac{K(K-1)}{2}n^2)$, whereas the one-versus-rest classification approach requires $O(KN^2)$ complexity, usually $K << N$. For the on-line learning mode, the efficiency is even more clear, as for each new incoming sample $K-1$ classifiers need to be updated (e.g. for class #1 samples the classifiers for pairs $(1, 2)$, $(1, 3)$, ..., $(1, K)$ are updated), whereas for one-versus-rest classification the update of K classifiers is necessary (one with indicator entry 1, the other $K - 1$ with indicator entry 0).

Compared to the direct multi-class classification structure, the training time for all-pairs classifier structures can be expected higher, as in the multi-class case only one single model is trained. Although $\frac{K(K-1)}{2}$ classifiers are trained, this does not imply an increase of quadratic complexity in training time (due to same the consideration made above: the single sub-problems contain significantly less number of samples); usually, the increase is linearly, as the total number of samples used in the training (over all class pairs) increases linearly (i.e. by a factor of K) as shown in [7]. On the other hand, the decision boundaries between the classes are easier to learn for the all-pairs classifier than for the direct multi-class classification structure. This is especially the case with increasing number of classes.

3. Training Phase

For incremental training and evolving the single binary classifiers $\mathbb{C}_{k,l}$ for all $k = 1, ..., K$ and $l = 1, ..., K$, in principle each evolving fuzzy classifier technique can be applied (see [22] for an overview). In this paper, we use *FLEXFIS-Class SM* method [4] in case of classifier architecture with singleton consequents (rules defined in (2)) and original *FLEXFIS* method [20] in case of regression-based classifiers (rules defined in (3)), denoted as *FLEXFIS-Class AP-SC* and *FLEXFIS-Class AP-TS*, resp. *EFC AP-SC* and *EFC AP-TS* in general. In both cases, clusters are associated with rules as both are representing local regions in the high-dimensional feature space and the rules are evolved by an *evolving version of vector quantization (eVQ)* [19], which, upon new incoming samples, decides whether they fit into the current cluster structure or not (based on a vigilance parameter and distance measure). If they do not fit, new rules are evolved by setting their centers to the current sample and the ranges of influence to a small value (not to 0 in order to avoid numerical instabilities); if they do, the nearest cluster to the current sample is moved towards this sample and its range of influence is updated with a modified version of recursive variance formula (including rank-one modification, see [25]).

The incremental learning of the consequent parts is different, as in case of *EFC AP-SC* singleton class labels of the rules need to be updated, while in case of *EFC AP-TS* hyper-planes including linear weights for each dimension need to be adapted. Thus, in case of *EFC AP-SC*, we introduce a hit matrix H whose entry h_{ij} is giving rise how much samples from class $j \in \{0, 1\}$ are falling into cluster (rule) k. Ideally, the hit matrix contains one positive entry in each row, while all other entries are 0. Tmhis would be the case when extracting 'clean' clusters from the feature space, i.e. each cluster contains only samples from one class. However, in real-world data streams usually the classes show some overlap in different regions of the feature space, leading to 'unpurified' clusters. In fact, such clusters can be omitted in large parts in case of low-dimensional problems when including the label entry as additional column during clustering (tearing apart nearby lying samples with different classes), however for high-dimensional problem this effect is weakened due to the curse of dimensionality effect

[8]: adding one single column to the learn space will only slightly change the distance between two vectors. The hit matrix can be simply evolved by incrementing the count h_{ij} if a new sample falls into cluster i and class j. If a new cluster/rule is evolved, a new line (the Cth) in the hit matrix is appended and $h_{Cl} = 1$ with l the class the current sample belongs to and $h_{Ck} = 0$ for the other class k. The singleton class label $L_i \in \{0, 1\}$ for the ith rule is obtained by taking the column index of the maximal entry of the ith row in H:

$$L_i = argmax_{j=0,1} h_{ij} \tag{4}$$

In case of *FLEXFIS-Class AP-TS*, the binary models are regression-based classifiers and the consequent parts are defined through hyper-planes (here for the ith):

$$l_i = w_{i0} + w_{i1}x_1 + w_{i2}x_2 + ... + w_{ip}x_p \tag{5}$$

Therefore, linear parameters need to be updated in a regression setting. This is achieved by using local learning instead of global one (due to several advantages with respect to numerical stability, computation time and flexibility when joining new rules on demand, see [4]) and exploiting the fuzzily weighted recursive least squares estimator [3] for each rule separately (here for the ith updating from time instance N to $N + 1$):

$$\hat{\vec{w}}_i(N+1) = \hat{\vec{w}}_i(N) + \gamma(N)(y(N+1) - \vec{r}^T(N+1)\hat{\vec{w}}_i(N)) \tag{6}$$

$$\gamma(N) = \frac{P_i(N)\vec{r}(N+1)}{\frac{1}{\Psi_i(\vec{x}(N+1))} + \vec{r}^T(N+1)P_i(N)\vec{r}(N+1)} \tag{7}$$

$$P_i(N+1) = (I - \gamma(N)\vec{r}^T(N+1))P_i(N) \tag{8}$$

with $\Psi_i(\vec{x}(N+1))$ the normalized membership function value for the $(N + 1)$th data sample, $P_i(N)$ the weighted inverse Hessian matrix, $\vec{r}(N + 1) = [1 \;\; x_1(N + 1) \;\; x_2(N + 1) \;\; \ldots \;\; x_p(N + 1)]^T$ the regressor values of the $(N + 1)$th data sample and $y(N + 1)$ the target value from $\{0, 1\}$ (0 when the sample belongs to the first class k and 1 when it belongs to the second one l).

4. Classification Phase

The classification phase is divided into two stages:

- The first stage produces the output confidence levels (preferences) for each class pair and stores it in the preference relation matrix as defined in (2).
- The second stage uses the whole information of the preference matrix and produces a final class response.

4.1. Classification Outputs

For each binary singleton class label classifier, the confidences in preferring class k over l is calculated by a normalized weighted sum:

$$conf_{k,l} = \frac{\mu_1 h *_{1,k} + \mu_2 h *_{2,k}}{\mu_1 + \mu_2} \tag{9}$$

with

$$h*_{1,k} = \frac{h_{1,k}}{h_{1,k} + h_{1,l}} \quad h*_{2,k} = \frac{h_{2,k}}{h_{2,k} + h_{2,l}} \tag{10}$$

the relative frequency (weight) of class k in the two nearest rules supporting the two classes ($h*_1$ belongs to the nearest rule supporting class k, $h*_2$ belongs to the nearest rule supporting class l), μ_1 the membership degree of the current sample to the nearest rule supporting class k, i.e. in which the kth class is most frequent one and μ_2 the membership degree of the current sample to the nearest rule supporting class l, i.e. in which the lth class is the most frequent one. If one of the two classes has no majority in any rule, that rule is taken which supports the minor class most. The reason for this strategy resp. the advantage over conventional winner-takes-it-all classification (as used in the direct multi-class EFC approaches, see [4]) will be highlighted below in the reliability aspects when dealing with conflict situations, see Section 4.2. In case when using binary regression-based classifiers, the confidences in preferring class k over l is simply calculated by using the defuzzified output of the Takagi-Sugeno model $y(k, l)$. As the target values were all defined to be either 0 or 1 (belonging to one of the two classes), the output usually lies in [0,1]. If it is lying outside this interval (extrapolation cases), we round it towards the nearest integer in $\{0, 1\}$. Then, we obtain

$$conf_{k,l} = 1 - y(k, l) \tag{11}$$

In the second stage, the information obtained from the preference relation matrix as defined in (2) is exploited in a meaningful way in order to achieve high classification performance. An obvious way for doing so is to apply a weighted voting procedure [12], where the score for each class k is given by:

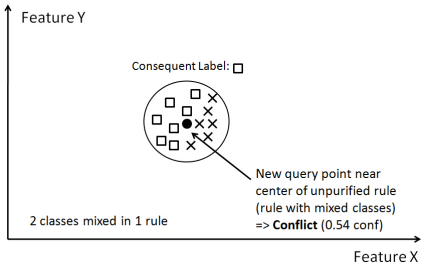
$$score_k = \sum_{K \geq i \geq 1} conf_{k,i} \tag{12}$$

i.e. the score for class k is simply the sum of the single confidences for preferring class k over all the other classes, and outputting the class with highest score as final decision:

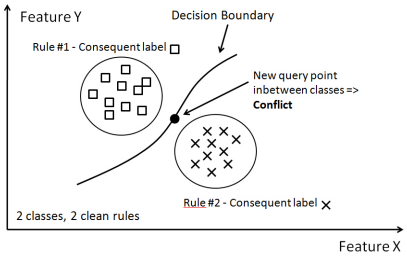
$$L_{est} = argmax_{k=1,\ldots,K} score_k \tag{13}$$

4.2. Reliability Aspects in Classification

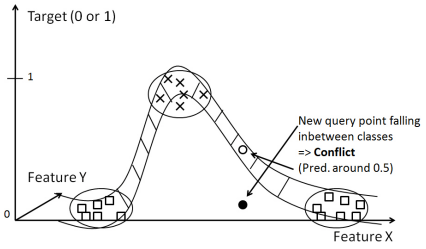
Another central aspect in classification is the *reliability* of classification outputs, i.e. how reliable the classifier itself sees its preference degree of one class over the other for new query points to be predicted. There are two basic concepts for expressing classifier's reliability, namely *conflict* and *ignorance* [11]. Conflict corresponds to that part of the



(a)



(b)



(c)

Figure 1: (a): one rule, but mixed classes → conflict, also when query point is near the center (0.54 preference to rectangular class); (b) two 'clean' rules, but query point appears very close to the decision boundary → conflict; (c) conflict in the case of *EFC AP-TS*: new query point lies between two rules containing different classes, hence is assigned a value around 0.5

classifier's uncertainty which is due to a close distance of a new query point to the decision boundary. Figure 1 presents two conflict examples: in (a) the classes are strongly overlapping inside one rule (denoted as cluster), so each query point falling into this rule causes a conflict situation, also when it is lying nearby the rule center; in (b) the classes are falling exactly into two rules (clusters), and only query points near the decision boundaries represent conflict cases. In both cases, the conflict can be expressed by a preference degree $conf_{k,l}$ of 0.5 in the preference relation matrix, which means, due to the reciprocal scores (i.e. $conf(l,k)$ also lying near 0.5), no clear preference of one class over the other can be made. When using the standard winner-takes-it-all classification scheme in *EFC AP-SC* (i.e. taking the confidence of the output class in the most active rule), the conflict of the first case can be indicated, but not the conflict in the second case (the confidence in the output class of the closest rule is 1 as both rules are 'clean'). This is one of the main reasons why we used a different weighted classification scheme which is defined in (9) and which is able to indicate conflict in both cases. Regression-based classifiers can indicate conflict in a quite natural way as assigning samples from one class a target value of 0 and samples from the other class a target value of 1 and performing regression inbetween; an example is shown in (c), where the z-axis denotes the regression target values (0 or 1) and the regression surface indicated with stripes. A query point falling inbetween two rules will be assigned a regression value of about 0.5, indicating a conflict.

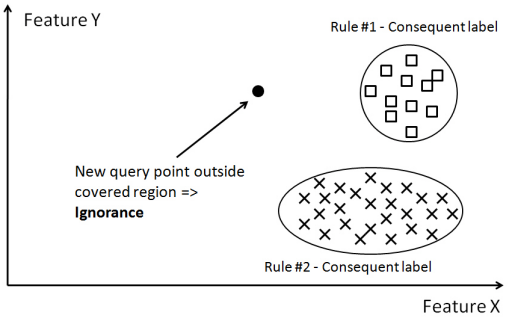


Figure 2: New query point to be predicted falls in unexplored feature space → ignorance as it is quite uncertain to assign one of the two classes as output label

Ignorance belongs to that part of classifier's uncertainty which is due to a query point falling into the extrapolation region of the feature space as indicated in Figure 2. In this case, the feature space is simply not sufficiently crowded with samples such that for a new query point falling into an unexplored region it is hard to guess whether it belongs to one of the two classes. This means that both, $conf_{k,l}$ as well as $conf_{l,k}$ should be close to 0, depending on the degree of ignorance. The nice thing is that fuzzy classifiers can represent ignorance in a quite natural way, see [11]. This is because in extrapolation cases as shown in Figure 2, no rule in the fuzzy classifier is really active with a significant firing degree ($\mu > \epsilon$ with a small positive number ϵ). Therefore, we weight the classifier preference degrees of class k over l by the maximal firing degree over all rules in the corresponding classifier and obtain the new confidences which we score through (12):

$$conf_{k,l} = conf_{k,l} * \max_{i=1,\ldots C} \mu_i(\vec{x})(k,l) \qquad (14)$$

with

$$\mu_i(\vec{x})(k,l) = \top_{j=1}^{p} \mu_{ij}(x_j)(k,l) \qquad (15)$$

with $\top$ denoting a t-norm in general. Assuming that the preference degrees are reciprocal and the ignorance degree is the same for classifier on k over

l as for classifier on l over k, we obtain:

$$conf_{l,k} = (1 - conf_{k,l}) * \max_{i=1,...C} \mu_i(\vec{x})(k,l) \quad (16)$$

i.e. the confidence of the preference of l over k is decreased by the same extent as the confidence of the preference of k over l.

5. Evaluation

This section is dedicated to the empirical evaluation of the novel evolving fuzzy classifier schemes *EFC-AP-TS* and *EFC-AP-SC*.

5.1. Experimental Setup

Hereby, we concentrated on the following data sets (taken from real-world applications in own projects and from UCI repository[1]):

- Inspection of CD Imprints: the classification problem is to detect system failures (such as color drift during offset print, a pinhole caused by a dirty sieve, occurrence of colors on dirt, palettes running out of ink) when printing the upper-side of compact discs; the inspection is done visually with high-resolution cameras installed directly along the production line. The recorded images of CD imprints are compared with a fault-free ideal reference image, also called master image. This comparison results in a grey-level contrast image, where pixels denote deviations to the master and potential fault candidates. Once, regions of interest (= pixels forming one deviation region are grouped together) are found with a specific variant of hierarchical clustering, object features are extracted for characterizing the outlook, shape, tendency etc. of the objects. (see also [27] for details and a list of features). These features are used as basis for a classifier training process (input) together with the labels of the objects they characterize.
- Pitch circle plates: there the problem is to detect various occurrences on the surface of these plates, such as dirt, scratches, porosities or splinters of glass; opposed to the CD imprint data, here no master images are calculated, but the original grey-level images are used; first, the objects are found by algorithms being able to put circumscribing ellipsoids over the regions of interest (usually identified by light and intensity indicators); second, object features (52 in sum) are extracted from these regions of interest (e.g. statistical measures from the grey level histograms, shape descriptors, size of circumscribing ellipsis, intensity features).
- Iris data set: contains samples from the species of three different flowers (setosa, versicolor and virginica) based on some length and width criteria of their blossoms; the goal is to distinguish between these three flower classes.
- Four data sets from UCI repository: iris, steel, vehicle, ecoli — due to space restriction we neglect the detailed description and refer to the corresponding data folders in the repository.

[1] http://archive.ics.uci.edu/ml/

Table 1: Characteristics of the applied data sets

	# Samples	# Feat.	# of Cl.
CD Imprint	1534	57	12
Plates	7411	52	8
Iris	300	4	3
Steel	1940	27	7
Vehicle	846	18	4
Ecoli	332	7	6

A summary of the characteristics of the data sets (# of inputs, classes, samples etc. they contain) is given in Table 1. For comparison purposes with state-of-the-art methods, we, apart from our novel algorithms, apply the following methods:

- *EFC-SM* [4]: evolving fuzzy classifier in single model architecture with singleton class labels and a winner-takes-it-all classification, which directly maps a new feature vector onto one of several classes.
- *EFC-MM* [21]: evolving fuzzy classifier exploiting multi-model architecture which uses a TS fuzzy regression model for each class based on indicator entries and performs a one-versus-rest classification scheme.
- *eVQ-Class* [18] [21]: an evolving clustering-based classifier which directly acts on the high-dimensional feature space including two variants: winner-takes-it-all classification (variant A) and an enhanced weighted classification strategy according to the distance of new samples to the decision boundary (variant B). This also maps a feature vector directly onto one of several classes.

The comparison with the first two variants is in conformity to check the advantage of the novel architecture as outlined in Section 2 over direct multi-class classification (as in *EFC SM*) and one-versus-rest approach (as in *EFC MM*), as in all cases we apply *FLEXFIS-Class* as training engine. Furthermore, in order to underline the impact of taking into account reliability aspects of the classifiers as described in Section 4.2, we include the degree of ignorance into the classifications as defined in (16) and ignore conflicting samples when calculating the classification accuracies, both indicating whether the inclusion of classifier reliability in its prediction may finally increase classification accuracy.

5.2. Results

In order to verify the performance of the classifiers on an independent test set, we used 10 shuffles of the data and for each shuffle split the whole data set into three thirds, where the first two thirds are used for incremental learning/evolving the classifiers, and the third third used for calculating the accuracies (test set as indicated in Table 1). We take the mean accuracies over all shuffles as well as the standard deviations over the shuffles in order to obtain the sensitivity of the methods with respect to the order of the data. Larger values correspond to higher sensitivity and should be taken with caution.

Table 2 visualizes the results obtained on the various data sets using the different data sets and variants of the all-pairs evolving fuzzy classifiers (the method/variant with highest accuracy for each data set shown in bold face); the rows including 'with ignorance' in the method description correspond to the down-weighting of preference degrees as outlined in (16), the rows including 'no res. on ign.' in the method description discards all samples having maximal rule membership degree smaller than 0.1 ($\rightarrow$ high ignorance) in the classification response (therefore also ignoring them when computing the classification accuracy). In all cases, *EFC-AP-TS* can mostly out-perform one-versus-rest counterpart *EFC MM* (or at least perform equally), both using Takagi-Sugeno fuzzy model architecture for the binary classifiers: in case of plates, steel, CD-Imprint and vehicle the performance boost is quite distinct (3% to 12%) as also lying significantly over the sensitivity on the different data shuffles; for the other three sets the increase is about 1% to 1.5%. This is quite remarkable when taking into account that the computational complexity is lower for *EFC-AP-TS* than for *EFC MM*, especially during the incremental update phase (as outlined in Section 2.2: small $K-1$ classifiers need to be updated instead of large K classifiers). The same is the case for *EFC AP-SC*, which can clearly outperform *EFC SM* (both using singleton consequent labels) for all data sets. Furthermore, the integration of the ignorance levels into the confidence levels after (16) (Rows #3 and #7) and ignoring ignorance (Rows #4 and #8) and conflict samples (Rows #5 and #9) (both serving as parts of the whole reliability concept) increase the accuracies over conventional all-pairs outputs further, for both, *EFC AP-SC* and *EFC-AP-TS*. The (partially significant) increase of accuracy when ignoring conflict and ignorance samples is an interesting point, as when abstaining randomly some samples from the test set to be included in the calculation of the classification accuracy (so not only those for which the classifiers were uncertain), it can be expected that the classification rates stay approximately the same. This means that the reliability concepts (including conflict and ignorance) discussed in Section 4.2 are in fact modeling the uncertainty of all-pairs classifiers on predicting new samples in an appropriate way. Finally, *EFC-AP-TS* performs equally or better than *EFS AP-SC* (except for CD-imprint data set), no matter whether the reliability concept is included or not. The standard deviations of the best performing methods (for each data set) are not showing any really severe fluctuations over the different shuffles.

6. Conclusion

In this paper, we extended state-of-the-art evolving fuzzy classifiers (EFC) with the concept of all-pairs classification in case of multi-class classification scenarios. The all-pairs model architecture reduces complexity for model updates compared to the one-versus-rest concept and and causes decision boundaries which are more easily to learn compared to direct multi-class response architecture. Furthermore, it allows some additional interpretation of the binary classifier outputs as these are stored in a preference relation matrix showing the preference degrees between all class pairs. The evaluation section shows that all-pairs evolving fuzzy classifiers can significantly out-perform the conventional state-of-the-art EFC techniques with respect to classification rates. In fact, it is remarkable that the regression-based variant *EFC-AP-TS* is the top-ranked method for five out of six high-dimensional, partly noisy real-world classification data sets. Furthermore, it is also remarkable that by including the classifiers' reliability concepts in the preference degrees (matrix), the accuracies of the *EFC-AP* classifiers can be significantly increased. This also shows the plausibility of the conflict and ignorance concept models described in Section 4.2. Future work includes the integration of the degree of non-linearity of the binary fuzzy classifiers into the ignorance levels and enhanced strategies for producing overall classification responses from the preference relation matrix including a more funded analysis from fuzzy relational point of view.

References

[1] P. Angelov, D. Filev, and N. Kasabov. *Evolving Intelligent Systems — Methodology and Applications.* John Wiley & Sons, New York, 2010.

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

[3] P.P. Angelov and D. Filev. An approach to online identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Systems, Man and Cybernetics, part B: Cybernetics*, 34(1):484–498, 2004.

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

[5] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena. *Interpretability Issues in Fuzzy Modeling.* Springer Verlag, Berlin Heidelberg, 2003.

[6] J. Fürnkranz. Round robin rule learning. In *Proceedings of the International Conference on Machine Learning (ICML 2011)*, pages 146–153, Williamstown, MA, 2001.

Table 2: Performance comparison of the various incremental learning methods on different shuffles of data of six multi-class classification applications, the first number denotes the mean accuracies over the different shuffles, the second number the standard deviation over the shuffles; first part includes results on the new methods, second on state-of-the-art methods for evolving/incremental (fuzzy) classifiers

Method	Plates	CD-Imp.	Iris	Steel	Vehicle	Ecoli
EFC AP-SC w/o reliability	61.7 ± 2.0	70.5 ± 1.4	95.2 ± 2.1	68.6 ± 1.9	61.7 ± 3.0	82.7 ± 4.0
EFC AP-SC with ign. in conf.	73.4 ± 0.9	74.9 ± 1.3	96.6 ± 2.0	67.9 ± 1.4	66.6 ± 2.4	84.7 ± 2.0
EFC AP-SC no resp. on ign.	73.9 ± 0.5	76.0 ± 1.2	96.6 ± 1.7	71.4 ± 2.8	67.2 ± 2.8	83.3 ± 2.9
EFC AP-SC no resp. on conf.	74.7±0.9	**82.6±1.5**	97.0 ± 1.6	70.7±2.3	73.3±2.3	86.3±3.5
EFC AP-TS w/o reliability	66.4 ± 1.6	71.9 ± 1.2	97.3±1.6	62.8 ± 2.0	82.0 ± 2.5	86.1 ± 1.4
EFC AP-TS with ign. in conf.	71.5 ± 1.4	74.7 ± 0.7	95.8 ± 1.2	65.8 ± 1.3	82.0 ± 1.9	85.5 ± 1.2
EFC AP-TS no resp. on ign.	74.4±1.2	76.1±1.2	96.6 ± 1.2	**76.3±2.4**	**83.1±2.0**	86.2±2.9
EFC AP-TS no resp. on conf.	**74.9±0.9**	79.7±1.2	**97.6±1.8**	68.2±2.1	82.3±2.5	**87.2±2.9**
EFC SM (direct map.)	36.4 ± 4.0	62.0 ± 2.5	94.8 ± 2.7	64.1 ± 2.3	53.2 ± 2.5	80.4 ± 3.6
EFC MM (one-vs-rest)	62.7 ± 1.8	73.1 ± 1.1	96.3 ± 1.3	65.5 ± 1.6	79.3 ± 1.5	86.0 ± 2.3
eVQ-Class var. A	57.9 ± 0.8	64.1 ± 2.4	93.7 ± 5.5	67.2 ± 1.8	54.3 ± 5.0	84.2 ± 3.6
eVQ-Class var. B	60.3 ± 1.1	74.9 ± 1.6	95.6 ± 1.1	39.3 ± 1.5	62.0 ± 1.9	84.6 ± 3.3

[7] J. Fürnkranz. Round robin classification. *Journal of Machine Learning Research*, 2:721–747, 2002.

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction - Second Edition*. Springer, New York Berlin Heidelberg, 2009.

[9] H. He and E.A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[10] E. Huellermeier. Uncertainty in clustering and classification. In A. Deshpande and A. Hunter, editors, *Springer Lecture Notes in Computer Science*, volume 6379 of *LNAI*, pages 16–19. Springer, 2010.

[11] J. Hühn and E. Hüllermeier. FR3: A fuzzy rule learner for inducing reliable classifiers. *IEEE Transactions on Fuzzy Systems*, 17(1):138–149, 2009.

[12] E. Hüllermeier and Klaus Brinker. Learning valued preference structures for solving classification problems. *Fuzzy Sets and Systems*, 159(18):2337–2352, 2008.

[13] N. Kasabov. *Evolving Connectionist Systems: The Knowledge Engineering Approach - Second Edition*. Springer Verlag, London, 2007.

[14] E.P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*. Kluwer Academic Publishers, Dordrecht Norwell New York London, 2000.

[15] L. Kuncheva. *Fuzzy Classifier Design*. Physica-Verlag, Heidelberg, 2000.

[16] A. Lemos, W. Caminhas, and F. Gomide. Fuzzy multivariate gaussian evolving approach for fault detection and diagnosis. In E. Hüllermeier, R. Kruse, and F. Hoffmann, editors, *Proc. of the 13th International Conference on Information Processing and Management of Uncertainty, Part II (Applications)*, volume 81 of *CCIS*, pages 360–369. Springer, Dortmund, Germany, 2010.

[17] E. Lima, M. Hell, R. Ballini, and F. Gomide. Evolving fuzzy modeling using participatory learning. In P. Angelov, D. Filev, and N. Kasabov, editors, *Evolving Intelligent Systems: Methodology and Applications*, pages 67–86. John Wiley & Sons, New York, 2010.

[18] E. Lughofer. Evolving vector quantization for classification of on-line data streams. In *Proc. of the Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA 2008)*, pages 780–786, Vienna, Austria, 2008.

[19] E. Lughofer. Extensions of vector quantization for incremental clustering. *Pattern Recognition*, 41(3):995–1011, 2008.

[20] E. Lughofer. FLEXFIS: A robust incremental learning approach for evolving TS fuzzy models. *IEEE Transactions on Fuzzy Systems*, 16(6):1393–1410, 2008.

[21] E. Lughofer. On-line evolving image classifiers and their application to surface inspection. *Image and Vision Computing*, 28(7):1065–1079, 2010.

[22] E. Lughofer. *Evolving Fuzzy Systems — Methodologies, Advanced Concepts and Applications*. Springer, Berlin Heidelberg, 2011. ISBN: 978-3-642-18086-6.

[23] E. Lughofer. On-line incremental feature weighing in evolving fuzzy classifiers. *Fuzzy Sets and Systems*, 163(1):1–23, 2011.

[24] V. Malathi, N. S. Marimuthu, and S. Baskar. A comprehensive evaluation of multicategory classification methods for fault classification in series compensated transmission line. *Neural Computing & Applications*, 19:595–600, 2010.

[25] S.J. Qin, W. Li, and H.H. Yue. Recursive PCA for adaptive process monitoring. *Journal of Process Control*, 10(5):471–486, 2000.

[26] B. Quost, T. Denoeux, and M.-H. Masson. Pairwise classifier combination using belief functions. *Pattern Recognition Letters*, 28(5):644–653, 2007.

[27] S. Raiser, E. Lughofer, C. Eitzinger, and J.E. Smith. Impact of object extraction methods on classification performance in surface inspection systems. *Machine Vision and Applications*, 21(5):627–641, 2010.

[28] R. R. Yager. A model of participatory learning. *IEEE Transactions on Systems, Man and Cybernetics*, 20(5):1229–1234, 1990.