Dynamic Classifier Systems for Classifier Aggregation

Štefka, David
2008

Dynamic Classifier Systems for Classifier Aggregation

Post-Graduate Student:
ING. DAVID ŠTEFKA
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague, Czech Republic
stefka@cs.cas.cz

Supervisor:
ING. RNDR. MARTIN HOLEŇA, CSC.
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague, Czech Republic
martin@cs.cas.cz

Field of Study:
Mathematical Engineering

The research reported in this paper was partially supported by the Program “Information Society” under project 1ET100300517 and by the grant ME949 of the Ministry of Education, Youth and Sports of the Czech Republic.

Abstract

Classifier aggregation is a method for improving quality of classification – instead of using just one classifier, a team of classifiers is created, and the outputs of the individual classifiers are aggregated into the final prediction. Common methods for classifier aggregation, such as mean value aggregation or weighted mean aggregation are *static*, i.e., they do not adapt to the currently classified pattern. In this paper, we introduce a formalism of *dynamic* classifier systems, which use the concept of dynamic classification confidence in the aggregation process, and therefore they dynamically adapt to the currently classified pattern. The results of the experiments with quadratic discriminant classifiers on four artificial and four real-world benchmark datasets show that dynamic classifier systems can significantly outperform both confidence-free and static classifier systems.

1. Introduction

Classification is a process of dividing objects (called *patterns*) into disjoint sets called *classes* [1]. Many machine learning algorithms for classification have been developed – for example naive Bayes classifiers, linear and quadratic discriminant classifiers, k-nearest neighbor classifiers, support vector machines, neural networks, or decision trees. If the quality of classification (i.e., the classifier’s predictive power) is low, there are several methods we can use to improve it.

One comonly used technique for improving classification quality is called *classifier combining* [2] – instead of using just one classifier, we create and train a team of classifiers, let each of them predict independently, and then combine (aggregate) their results. It can be shown that a team of classifiers can perform better in the classification task than any of the individual classifiers.

There are two main approaches to classifier combining: *classifier selection* [3, 4, 5] and *classifier aggregation* [6, 7]. If a pattern is submitted for classification, the former technique uses some rule to select one particular classifier, and only this classifier is used to obtain the final prediction. The latter technique uses some aggregation rule to aggregate the results of all the classifiers in a team to get the final prediction.

A common drawback of classifier aggregation methods is that they are static, i.e., they are not adapted to the particular patterns that are currently classified. In other words, the aggregation is specified during a training phase, prior to classifying a test pattern. However, if we use the concept of dynamic classification confidence (i.e., the extent to which we can “trust” the output of the particular classifier for the currently classified pattern), the aggregation algorithms can take into account the fact that “this classifier is not good *for this particular pattern*”.

Surprisingly, such dynamic classifier systems are not used very often in classifier combining. However, there has already been some work done in the field of dynamic classifier systems – Robnik-Šikonja and Tsymbal et al. [8, 9] study dynamic aggregation of random forests [10], i.e., dynamic classifier systems of decision trees. The authors report significant improvements in classification quality when using dynamic voting compared to simple voting. However, they study dynamic classifier systems only in the context of random forests, and they use only confidence measures based on the so-called margin.

In this paper, we provide a general formalism of dynamic classification confidence measures and dynamic classifier systems, and we experimentally study the performance of confidence-free classifier systems (i.e., systems that do not utilize classification confidence at all), static classifier systems (i.e., systems that use only "global" confidence of a classifier), and dynamic classifier systems (i.e., systems that adapt to the particular pattern submitted for classification).

The paper is structured as follows. In Section 2, we introduce the formalism of classifier combining, namely in Section 2.1, we define basic concepts of classification, in Section 2.2 we introduce the concept of classification confidence, and we introduce three dynamic confidence measures, in Section 2.3 we deal with classifier teams and ensembles, and in Section 2.4, we finally define classifier systems and show several examples of dynamic classifier systems. In Section 3, we experimentally investigate the suitability of the proposed dynamic confidence measures, and the performance of the proposed dynamic classifier systems. Section 4 then concludes the paper.

2. Formalism of Classifier Combining with Classification Confidence

2.1. Classification

Throughout the rest of the paper, we use the following notation. Let $\mathcal{X} \subseteq \mathbf{R}^n$ be a n-dimensional *feature space*, an element $\vec{x} \in \mathcal{X}$ of this space is called a *pattern*, and let $C_1, \ldots, C_N \subseteq \mathcal{X}$, $N \geq 2$, be disjoint sets called *classes*. The index of the class a pattern $\vec{x}$ belongs to will be denoted as $c(\vec{x})$ (i.e., $c(\vec{x}) = i$ iff $\vec{x} \in C_i$). The goal of classification is to determine to which class a given pattern belongs, i.e., to predict $c(\vec{x})$ for unknown patterns.

Definition 1 *We call a* classifier *every mapping* $\phi : \mathcal{X} \rightarrow [0,1]^N$*, where* $[0,1]$ *is the unit interval, and* $\phi(\vec{x}) = (\mu_1(\vec{x}), \ldots, \mu_N(\vec{x}))$ *are* degrees of classification (d.o.c.) *to each class.*

The d.o.c. to class C_j expresses the extent to which the pattern belongs to class C_j (if $\mu_i(\vec{x}) > \mu_j(\vec{x})$, it means that the pattern $(\vec{x})$ belongs to class C_i rather than to C_j). Depending on the classifier type, it can be modelled by probability, fuzzy membership, etc.

Remark 1 *This definition is of course not the only way how a classifier can be defined, but in the theory of classifier combining, this one is used most often [2].*

Definition 2 *Classifier ϕ is called* crisp*, iff* $\forall \vec{x} \in \mathcal{X}\ \exists i$*, such that:*

$$\mu_i(\vec{x}) = 1, \text{ and } \forall j \neq i\ \mu_j(\vec{x}) = 0.$$

Classifier ϕ is called normalized*, iff*

$$\forall \vec{x} \in \mathcal{X} : \sum_{i=1}^{N} \mu_i(\vec{x}) = 1,$$

where $\phi(\vec{x}) = (\mu_1(\vec{x}), \ldots, \mu_N(\vec{x}))$.

Remark 2 *Normalized classifiers are sometimes called* probabilistic *[6]. However, they do not need to be based on probability theory, so we will call them just normalized.*

Definition 3 *Let ϕ be a classifier,* $\vec{x} \in \mathcal{X}$, $\phi(\vec{x}) = (\mu_1(\vec{x}), \ldots, \mu_N(\vec{x}))$. Crisp output *of ϕ on $\vec{x}$ is defined as* $\phi_{cr}(\vec{x}) = \arg\max_{i=1,\ldots,N} \mu_i(\vec{x})$.

2.2. Classification Confidence

Classification confidence expresses the degree of trust we can give to a classifier ϕ when classifying a pattern $\vec{x}$. It is modelled by a mapping κ_ϕ.

Definition 4 *Let ϕ be a classifier. We call a* confidence measure *of classifier ϕ every mapping* $\kappa_\phi : \mathcal{X} \rightarrow [0,1]$.

The higher the confidence, the higher the probability of correct classification. $\kappa_\phi(\vec{x}) = 0$ means that the classification may not be correct, while $\kappa_\phi(\vec{x}) = 1$ means the classification is probably correct. However, κ_ϕ does not need to be modelled by a probability measure.

A confidence measure can be either *static*, i.e., it is a constant of the classifier, or *dynamic*, i.e., it adjusts itself to the currently classified pattern.

Definition 5 *Let ϕ be a classifier and κ_ϕ its confidence measure. We call κ_ϕ* static*, iff it is constant in $\vec{x}$, we call κ_ϕ* dynamic *otherwise.*

Remark 3 *Since static confidence measures are constant, independent on the currently classified pattern, we will omit the pattern $(\vec{x})$ in the notation, i.e., we will denote them just κ_ϕ.*

Remark 4 *In the rest of the paper, we will use the indicator operator I, defined as $I($true$) = 1$, $I($false$) = 0$.*

2.2.1 Static confidence measures:

After the classifier has been trained, we can use a testing set (i.e., a set of patterns on which the classifier has not been trained) to assess its predictive power as a whole (from global view). These methods include accuracy, precision, sensitivity, resemblance, etc. [1, 11], and we can use these measures as static confidence measures. In this paper, we will use the Global Accuracy measure.

Global Accuracy (GA) of a classifier ϕ is defined as the proportion of correctly classified patterns from the testing set:

$$\kappa_\phi^{(GA)} = \frac{\sum_{\vec{y}\in\mathcal{M}} I(\phi(\vec{y}) \overset{?}{=} c(\vec{y}))}{|\mathcal{M}|}, \qquad (1)$$

where $\mathcal{M}$ is the testing set of ϕ.

2.2.2 Dynamic confidence measures:

An easy way how a dynamic confidence measure can be defined is to compute some property on patterns neighboring with $\vec{x}$. Let $N(\vec{x})$ denote a set of neighboring training or validating patterns (we can use both training and validating set for computing $N(\vec{x})$, but it is usually better to use validating set, because if we use training patterns, the results will be biased). In this paper, we define $N(\vec{x})$ as the set of k patterns nearest to $\vec{x}$ under Euclidean metric. Now we will define three dynamic confidence measures which use $N(\vec{x})$:

Euclidean Local Accuracy (ELA) measures the local accuracy of ϕ in $N(\vec{x})$:

$$\kappa_\phi^{(ELA)}(\vec{x}) = \frac{\sum_{\vec{y}\in N(\vec{x})} I(\phi_{cr}(\vec{y}) \overset{?}{=} c(\vec{y}))}{|N(\vec{x})|}, \quad (2)$$

where $\phi_{cr}(\vec{y})$ is the crisp output of ϕ on $\vec{y}$.

Euclidean Local Match (ELM) is based on the ideas from [12], and measures the proportion of patterns in $N(\vec{x})$ from the same class as ϕ is predicting for $\vec{x}$:

$$\kappa_\phi^{(ELM)}(\vec{x}) = \frac{\sum_{\vec{y}\in N(\vec{x})} I(\phi_{cr}(\vec{x}) \overset{?}{=} c(\vec{y}))}{|N(\vec{x})|}, \quad (3)$$

where $\phi_{cr}(\vec{x})$ is the crisp output of ϕ on $\vec{x}$.

Euclidean Average Margin (EAM) is defined as mean value of the margin [10, 8, 9] in $N(\vec{x})$:

$$\kappa_\phi^{(EAM)}(\vec{x}) = \frac{\sum_{\vec{y}\in N(\vec{x})} mg(\phi(\vec{y}))}{|N(\vec{x})|}, \qquad (4)$$

where the margin is defined as $mg(\phi(\vec{y})) =$

$$\begin{cases} \mu_{c(\vec{y})}(\vec{y}) - \max\limits_{\substack{i=1,\dots,N \\ i\neq c(\vec{y})}} \mu_i(\vec{y}) & \text{if } \phi_{cr}(\vec{y}) = c(\vec{y}), \\ 0 & \text{otherwise.} \end{cases}, \qquad (5)$$

where $\phi(\vec{y}) = (\mu_1(\vec{y}), \dots, \mu_N(\vec{y}))$, and $\phi_{cr}(\vec{y})$ is the crisp output of ϕ on $\vec{y}$.

The dynamic confidence measures defined in this section have one drawback – they need to compute $N(\vec{x})$, which can be time-consuming, and sensitive to the similarity measure used. There are also dynamic confidence measures, which compute the classification confidence directly from $\phi(\vec{x})$, e.g., the ratio of the highest degree of classification to the sum of all degrees of classification. However, our preliminary experiments with such measures with quadratic discriminant classifiers and random forests show that such confidence measures give very poor results.

Remark 5 *All the previous confidence measures are* model-indifferent*, i.e., they could be used for any classifier. However, measures which take into account specific aspects of the classification method could be designed – for example, Robnik-Šikonja and Tsymbal et al. [8, 9] use dynamic confidence of a decision tree in a random forest [10] as average margin computed on instances similar to the currently classified pattern, where the similarity is based on specific aspects of random forests. Such* model-specific *measures could use the information from the classification process better than model-indifferent measures. However, due to space constraints we do not deal with model-specific measures in this paper.*

2.3. Classifier Teams

In classifier combining, instead of using just one classifier, a team of classifiers is created, and the team is then aggregated into one final classifier. If we want to utilize classification confidence in the aggregation process, each classifier must have its confidence measure defined.

Definition 6 Classifier team *is a tuple $(\mathcal{T}, \mathcal{K})$, where $\mathcal{T} = (\phi_1, \dots, \phi_r)$ is a set of classifiers, and $\mathcal{K} = (\kappa_{\phi_1}, \dots, \kappa_{\phi_r})$ is a set of corresponding confidence measures.*

If a classifier team consists only of classifiers of the same type, which differ only in their parameters, dimensionality, or training sets, the team is usually called an *ensemble of classifiers*. For this reason the methods which create a team of classifiers are sometimes called *ensemble methods*. The restriction to classifiers of the same type is not essential, but it ensures that the outputs of the classifiers are consistent.

Well-known methods for ensemble creation are *bagging* [13], *boosting* [14], *error correction codes* [2], or *multiple feature subset* methods [15]. These methods try to create an ensemble of classifiers which are both *accurate* and *diverse* [16].

Since the main focus of this paper lies in studying classification confidence, we will not study these methods here, and we will just assume in the rest of the paper that we have constructed a classifier team $(\mathcal{T}, \mathcal{K})$ of r classifiers using some of these methods.

If a pattern is submitted for classification, the team of classifiers gives us two different informations – outputs of the individual classifiers (a *decision profile*), and values of classification confidences of the classifiers (a *confidence vector*).

Definition 7 *Let* $(\mathcal{T}, \mathcal{K})$ *be a classifier team,* $\mathcal{T} = (\phi_1, \ldots, \phi_r)$*,* $\mathcal{K} = (\kappa_{\phi_1}, \ldots, \kappa_{\phi_r})$*, and let* $\vec{x} \in \mathcal{X}$*. Then we define* decision profile $\mathcal{T}(\vec{x}) \in [0,1]^{r,N}$ *as*

$$\mathcal{T}(\vec{x}) = \begin{pmatrix} \phi_1(\vec{x}) \\ \phi_2(\vec{x}) \\ \vdots \\ \phi_r(\vec{x}) \end{pmatrix} = \begin{pmatrix} \mu_{1,1} & \mu_{1,2} & \cdots & \mu_{1,N} \\ \mu_{2,1} & \mu_{2,2} & \cdots & \mu_{2,N} \\ & & \ddots & \\ \mu_{r,1} & \mu_{r,2} & \cdots & \mu_{r,N} \end{pmatrix}, \tag{6}$$

and confidence vector $\mathcal{K}(\vec{x}) \in [0,1]^r$ *as*

$$\mathcal{K}(\vec{x}) = \begin{pmatrix} \kappa_{\phi_1}(\vec{x}) \\ \kappa_{\phi_2}(\vec{x}) \\ \vdots \\ \kappa_{\phi_r}(\vec{x}) \end{pmatrix} \tag{7}$$

Remark 6 *Here we use the notation* $\mathcal{T}$ *for both the set of classifiers, and for the decision profile, and similarly for* $\mathcal{K}$*. To avoid any confusion, the decision profile and confidence vector will be always followed by* $(\vec{x})$*.*

2.4. Classifier Systems

After the pattern $\vec{x}$ has been classified by all the classifiers in the team, and the confidences were computed, these outputs have to be aggregated using a *team aggregator*, which takes the decision profile as its first argument, the confidence vector as its second argument, and returns the aggregated degrees of classification to all the classes.

Definition 8 *Let* $r, N \in \mathbf{N}$*,* $r, N \geq 2$*. A* team aggregator *of dimension* (r, N) *is any mapping* $\mathcal{A} : [0,1]^{r,N} \times [0,1]^r \to [0,1]^N$*.*

A classifier team with an aggregator will be called a *classifier system*. Such system can be also viewed as a single classifier.

Definition 9 *Let* $(\mathcal{T}, \mathcal{K})$ *be a classifier team, and let* $\mathcal{A}$ *be a team aggregator of dimension* (r, N)*, where r is the number of classifiers in the team, and* N *is the number of classes. We define an* induced classifier *of* $(\mathcal{T}, \mathcal{K}, \mathcal{A})$ *as a classifier* Φ*, defined as*

$$\Phi(\vec{x}) = \mathcal{A}(\mathcal{T}(\vec{x}), \mathcal{K}(\vec{x})).$$

The 4-tuple $\mathcal{S} = (\mathcal{T}, \mathcal{K}, \mathcal{A}, \Phi)$ *is called a* classifier system.

Depending on the way how a classifier system utilizes the classification confidence, we can distinguish several kinds of classifier systems.

Definition 10 *Let* $(\mathcal{T}, \mathcal{K})$ *be a classifier team.* $(\mathcal{T}, \mathcal{K})$ *is called* static*, iff*

$$\forall \kappa \in \mathcal{K} : \kappa \textit{ is a static confidence measure}.$$

$(\mathcal{T}, \mathcal{K})$ *is called* dynamic*, iff*

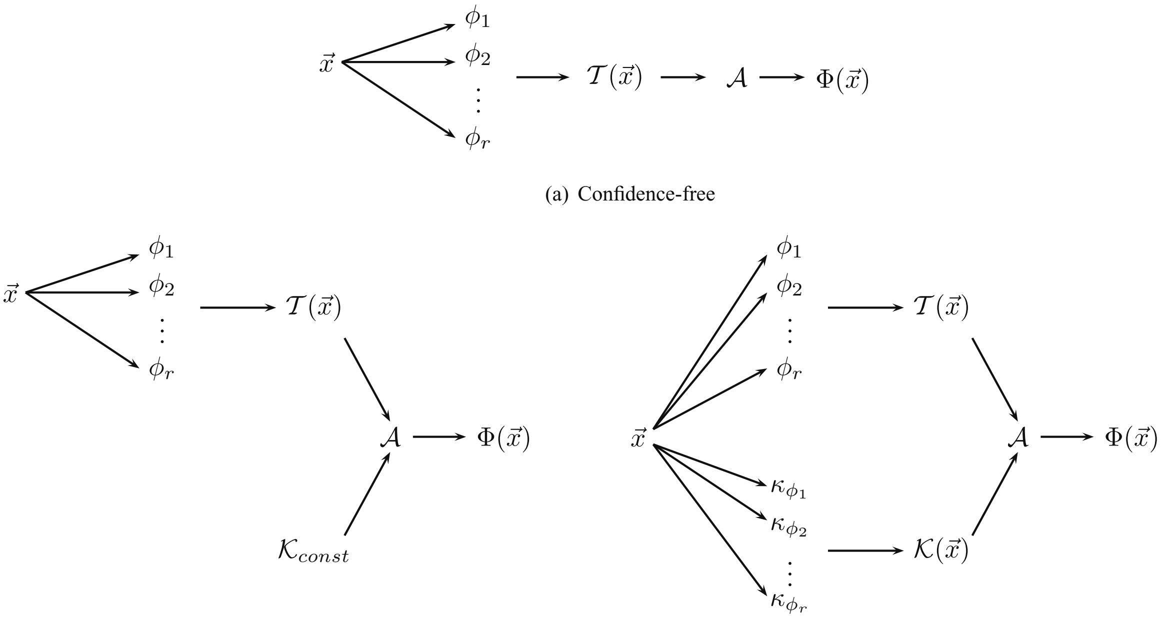
$$\forall \kappa \in \mathcal{K} : \kappa \textit{ is a dynamic confidence measure}.$$

Definition 11 *Let* $\mathcal{A}$ *be a team aggregator of dimension* (r, N)*. We call* $\mathcal{A}$ confidence-free*, iff* $\forall\, \mathbf{T} \in [0,1]^{r,N}$:

$$(\forall \vec{k}_1, \vec{k}_2 \in [0,1]^r : \mathcal{A}(\mathbf{T}, k_1) = \mathcal{A}(\mathbf{T}, k_2)).$$

Definition 12 *Let* $\mathcal{S} = (\mathcal{T}, \mathcal{K}, \mathcal{A}, \Phi)$ *be a classifier system. We call* $\mathcal{S}$ confidence-free*, iff* $\mathcal{A}$ *is confidence-free. We call* $\mathcal{S}$ static*, iff* $(\mathcal{T}, \mathcal{K})$ *is static, and* $\mathcal{A}$ *is not confidence-free. We call* $\mathcal{S}$ dynamic*, iff* $(\mathcal{T}, \mathcal{K})$ *is dynamic, and* $\mathcal{A}$ *is not confidence-free.*

Confidence-free systems do not utilize the classification confidence at all (for example a team of classifiers aggregated by simple voting). Static systems utilize classification confidence, but only as a global property (for example a team of classifiers aggregated by weighted voting with constant classifier weights). Dynamic systems utilize classification confidence in a dynamic way, i.e. the aggregation is adapted to the particular pattern submitted for classification (for example a team of classifiers aggregated by weighted voting with classifier weights computed for every pattern). The different approaches are schematically shown in Fig. 1.



(a) Confidence-free

(b) Static

(c) Dynamic

Figure 1: Schematic comparison of confidence-free, static, and dynamic classifier systems.

Remark 7 *Since confidence-free classifier systems do not utilize the classification confidence, we will denote them $\mathcal{S} = (\mathcal{T}, \mathcal{A}, \Phi)$, and their team aggregators will be defined as a mapping $\mathcal{A} : [0,1]^{r,N} \to [0,1]^N$.*

Many methods for aggregating the team of classifiers into one final classifier have been proposed in the literature. A good overview of commonly used aggregation methods can be found in [6]. These methods comprise simple arithmetic rules (voting, sum, product, maximum, minimum, average, weighted average, etc.), fuzzy integral, Dempster-Shafer fusion, second-level classifiers, decision templates, and many others.

In the following text, we define several team aggregators. We will use the notation from Def. 7 and Def. 9. Let $\Phi(\vec{x}) = \mathcal{A}(\mathcal{T}(\vec{x}), \mathcal{K}(\vec{x})) = (\mu_1(\vec{x}), \ldots, \mu_N(\vec{x}))$.

Mean value aggregation (MV) is the most common (confidence-free) aggregation technique. Its aggregator is defined as

$$\mu_j(\vec{x}) = \frac{\sum_{i=1,\ldots,r} \mu_{i,j}(\vec{x})}{r}. \tag{8}$$

If the classifiers in the team are crisp, MV coincides with voting.

Static weighted mean aggregation (SWM) computes aggregated d.o.c. as weighted mean of d.o.c. given by the individual classifiers, where the weights are static classification confidences:

$$\mu_j(\vec{x}) = \frac{\sum_{i=1,\ldots,r} \kappa_{\phi_i} \mu_{i,j}(\vec{x})}{\sum_{i=1,\ldots,r} \kappa_{\phi_i}}. \tag{9}$$

Dynamic weighted mean aggregation (DWM) has the same aggregator as SWM, but the weights are dynamic classification confidences:

$$\mu_j(\vec{x}) = \frac{\sum_{i=1,\ldots,r} \kappa_{\phi_i}(\vec{x}) \mu_{i,j}(\vec{x})}{\sum_{i=1,\ldots,r} \kappa_{\phi_i}(\vec{x})}. \tag{10}$$

Filtered mean aggregation (FM) has the same aggregator as MV, but prior to computing the aggregated values, the classifiers which have (dynamic) classification confidence lower than $T \in [0,1]$ are discarded:

$$\mu_j(\vec{x}) = \frac{\sum\limits_{\substack{i=1,\ldots,r \\ \kappa_{\phi_i}(\vec{x}) > T}} \mu_{i,j}(\vec{x})}{|\{\phi \in \mathcal{T} | \kappa_{\phi_i}(\vec{x}) > T\}|}. \tag{11}$$

3. Experiments

3.1. Experiment 1 – Choosing the Right Confidence Measure

To gain a general idea to which extent the proposed dynamic confidence measures (ELA, ELM, and EAM) really express the probability that the classification of the currently classified pattern is right, we examined the distributions of the confidence values for correctly classified and for misclassified patterns.

The confidence measures were tested on quadratic discriminant classifiers [1]. The classifiers were implemented in Java programming language and 10-fold crossvalidation was performed to obtain the results. We measured histograms of the local classification confidence values for correctly classified and for misclassified patterns from four artificial (Clouds, Concentric, Gauss_3D, Waveform) and four real-world (Breast, Phoneme, Pima, Satimage) datasets from the Elena database [17] and from the UCI repository [18]. As $N(\vec{x})$, we used the set of 20 nearest neighbors of $\vec{x}$ under Euclidean metric.

The histograms of the dynamic confidence values for the particular datasets are shown in Fig. 2. Before discussing the results, we should say a few words about how the results *should* ideally look like. We will denote the distribution of local classification confidence values for correctly classified patterns as “OK distribution”, and for misclassified patterns as “NOK distribution”. The OK distribution should be concentrated near one, while the NOK distribution should be concentrated near zero, and ideally, the distributions should be clearly separated. If the distributions overlap, or if the NOK distribution has high values near one, it means that the measure does not really express the probability that the classification of the currently classified pattern is right.

The results show that for some datasets, all the dynamic confidence measures provide good separation of the OK and NOK patterns, which suggests the measures are suitable for using in dynamic classifier systems. The most representative example of such behavior is the Phoneme dataset, where the OK and NOK distributions for all three dynamic confidence measures are clearly separated.

For some datasets, there are notable differences in the dynamic confidence measures – e.g., in the case of the Satimage dataset, the EAM confidence measure provides much better separation of the OK and NOK patterns than the other two measures. In the case of the Concentric dataset, the ELM confidence measure is an obvious winner. This means that the performance of a confidence measure is dependent on the particular dataset, and that the choice of a confidence measure should be always done with respect to the particular data.

For several datasets, all three dynamic confidence measures provided very poor separation of the OK and NOK patterns, which raises doubts about the suitability of the measures in dynamic classifier systems. This is the case of the Gauss_3D or the Pima dataset.

However, we cannot make direct conclusions about suitability of the measures just from the separation properties of the OK and NOK patterns. To give one example: even if the separation is good enough, the high values of dynamic classification confidence may be obtained on the “easy” patterns, and the low values on the “hard” patterns. Moreover, if the classifiers in the classifier system are “similar”, all of them will have similar confidence on a particular pattern. Therefore, dynamic aggregation of the system will bring no improvement in the classification quality, since all the classifiers appear the same for the system’s aggregator. This may be the explanation of the result of Exp. 2 for the Phoneme dataset, where the FM aggregation has gives very different performance for ELM and EAM confidence measures, even if the OK and NOK separation of the measures is nearly the same (see Fig. 2).

3.2. Experiment 2 – Confidence-free vs. Static vs. Dynamic Classifier Systems

In the second experiment, we compared the performance of the classifier aggregation algorithms described in Section 2.4. The main emphasis was given to comparing confidence-free vs. static vs. dynamic classifier systems. We used the same datasets as in Exp. 1.

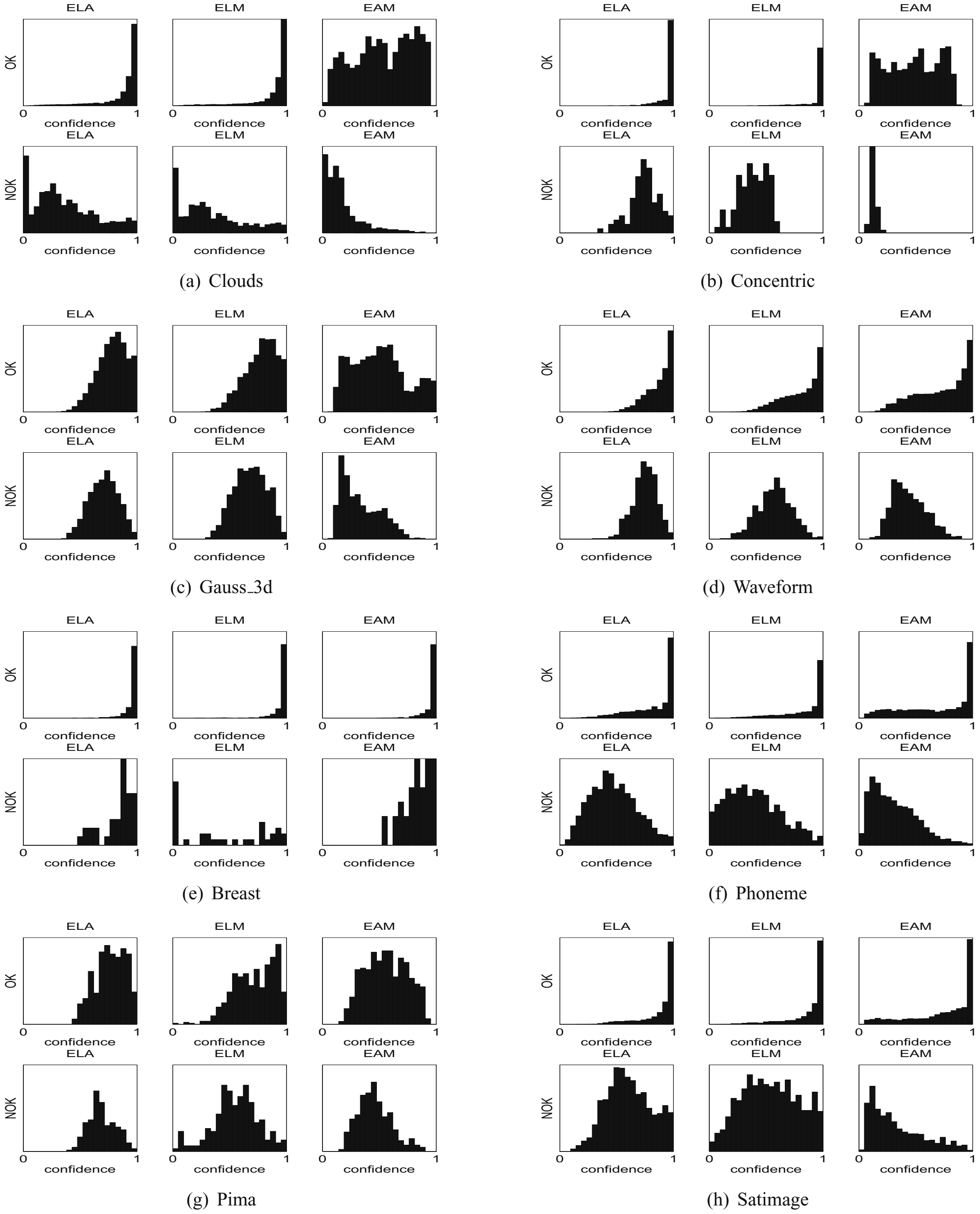
For all the classifier systems we used, the classifier team $\mathcal{T}$ was an ensemble of quadratic discriminant classifiers, created either by the bagging algorithm [13] (which creates classifiers trained on random samples drawn from the original training set with replacement), or by the multiple feature subset method [15] (which creates classifiers using different combinations of features), depending on which method was more suitable for the particular dataset.

For the comparison, we designed the following classifier systems (refer to Section 2.2 and Section 2.4 for the description of the algorithms):

MV confidence-free system aggregated by mean value aggregation

SWM cl. system aggregated by static weighted mean aggregation; as a confidence measure, we used GA

DWM cl. system aggregated by dynamic weighted mean; as a confidence measure, we used ELA, ELM, and EAM



(a) Clouds

(b) Concentric

(c) Gauss_3d

(d) Waveform

(e) Breast

(f) Phoneme

(g) Pima

(h) Satimage

Figure 2: Histograms of dynamic confidence values of a quadratic discriminant classifier (ELA - Euclidean Local Accuracy, ELM - Euclidean Local Match, EAM - Euclidean Average Margin) for correctly classified (OK) and misclassified (NOK) patterns.

Table 1: Comparison of the aggregation methods – non-combined classifier (NC), mean value (MV), static weighted mean (SWM) using GA confidence measure, dynamic weighted mean (DWM) using three confidence measures (ELA, ELM, EAM), and filtered mean (FM) using three confidence measures (ELA, ELM, EAM). Mean error rate (in %) ± standard deviation of error rate from a 10-fold crossvalidation was measured. The best result is displayed in boldface, statistically significant improvements to NC, MV, and SWM are marked by footnote signs. The (B/M) after dataset name means whether the ensemble was created by Bagging or Multiple feature subset algorithm.

	Non-Combined	Conf.-free	Static		Dynamic		
Dataset	NC	MV	κ	SWM	κ	DWM	FM
Clouds (M)	25.0 ± 1.7	25.0 ± 2.1	GA	24.7 ± 1.6	ELA	23.4 ± 1.5	22.3 ± 1.5 *†‡
					ELM	23.2 ± 1.2	$\mathbf{22.0 \pm 2.1}$ *†‡
					EAM	23.5 ± 1.5	23.3 ± 1.4
Concentric (B)	3.5 ± 1.0	3.8 ± 0.6	GA	4.0 ± 0.8	ELA	3.2 ± 1.1	2.1 ± 1.3 †‡
					ELM	2.9 ± 1.6	$\mathbf{1.8 \pm 0.8}$ *†‡
					EAM	3.8 ± 1.3	4.3 ± 1.5
Gauss_3D (B)	21.4 ± 1.7	21.6 ± 1.1	GA	21.5 ± 2.1	ELA	21.5 ± 1.4	21.7 ± 1.3
					ELM	$\mathbf{21.3 \pm 2.0}$	22.0 ± 1.3
					EAM	21.5 ± 2.0	21.7 ± 1.3
Waveform (B)	14.9 ± 2.5	15.0 ± 1.4	GA	14.8 ± 0.9	ELA	14.7 ± 1.9	15.0 ± 1.2
					ELM	14.8 ± 2.5	$\mathbf{14.5 \pm 1.2}$
					EAM	14.6 ± 2.0	15.5 ± 1.0
Breast (M)	4.8 ± 2.9	4.7 ± 2.5	GA	4.2 ± 2.4	ELA	3.0 ± 2.1	2.9 ± 1.8
					ELM	3.0 ± 1.9	3.1 ± 2.1
					EAM	3.2 ± 2.0	$\mathbf{2.9 \pm 1.7}$
Phoneme (M)	24.7 ± 1.1	23.5 ± 1.6	GA	24.0 ± 1.4	ELA	21.5 ± 1.9 *‡	17.2 ± 1.4 *†‡
					ELM	21.2 ± 1.8 *‡	$\mathbf{16.9 \pm 2.0}$ *†‡
					EAM	21.9 ± 0.9 *	20.7 ± 1.7 *†‡
Pima (M)	27.1 ± 4.4	25.4 ± 3.6	GA	25.0 ± 5.6	ELA	25.8 ± 6.5	24.0 ± 2.7
					ELM	24.0 ± 4.1	25.0 ± 7.4
					EAM	24.8 ± 6.3	$\mathbf{23.5 \pm 5.4}$
Satimage (B)	15.6 ± 1.7	15.5 ± 1.2	GA	15.5 ± 1.7	ELA	15.3 ± 1.6	15.2 ± 2.4
					ELM	15.3 ± 1.3	$\mathbf{14.4 \pm 1.0}$
					EAM	15.5 ± 1.2	15.0 ± 1.5

*Significant improvement to NC
†Significant improvement to MV
‡Significant improvement to SWM

FM cl. system aggregated by filtered mean; as a confidence measure, we used ELA, ELM, and EAM

We also compared the systems' performance with the so-called *non-combined classifier* (NC), i.e., a common quadratic discriminant classifier (the NC classifier represents an approach which we had to use if we could use only one classifier).

All the methods were implemented in Java programming language, and a 10-fold crossvalidation was performed to obtain the results. For the dynamic confidence measures, we used the same definition of $N(\vec{x})$ as in Exp. 1, and the threshold T for FM aggregators was set to $T = 0.8$ or $T = 0.9$, depending on the particular dataset (based on some preliminary testing; no fine-tuning or optimization was done).

The results of the testing are shown in Table 1. Mean error rate and standard deviation of the error rate of the induced classifiers from a 10-fold crossvalidation was measured. We also measured statistical significance of the results – at 5% confidence level by the analysis of variance using the Tukey-Kramer method (by the 'multcomp' function from the Matlab statistics toolbox).

The results show that for most datasets, the dynamic classifier systems outperform both confidence-free and static classifier systems. For three datasets, these results were statistically significant. FM usually gives better results than DWM, and if we compare the three dynamic confidence measures, we can say that ELM gives usually the best results, ELA and ELM being slightly worse. However, as we already discussed in Exp. 1, the performance of the individual confidence measures is dependent on the particular dataset. Generally speaking, the FM-ELM was the most successfull algorithm in this experiment.

It should be noted that the experimental results from this paper are relevant only to quadratic discriminant classifiers, because for any other classifier types (k-NN, SVM, decision trees, etc.), the dynamic confidence measures could give quite different results.

4. Summary

In this paper, we have studied dynamic classifier aggregation. We have introduced the formalism of classifier systems which can be used with (dynamic) classification confidence, and we have defined confidence-free, static, and dynamic classifier systems. We have introduced three dynamic classification confidence measures (ELA, ELM, EAM), and we have shown a way how these measures can be used in dynamic classifier systems – we have introduced two algorithms for dynamic classifier aggregation.

In our first experiment, we have studied the distributions of values of the proposed dynamic classification confidence measures for correctly classified and misclassified patterns, which can give us a hint about suitability of the measures in dynamic classifier systems. The results show that the performance of the particular confidence measure is dependent of the particular dataset.

In the second experiment, we have compared the performance of confidence-free, static, and dynamic classifier systems of quadratic discriminant classifiers. The results show that dynamic classifier systems can significantly outperform both confidence-free and static classifier systems.

The main contribution of this paper is the verification that the concept of dynamic classification confidence can significantly improve the classification quality, and that it is a general concept, which can be incorporated into the theory of classifier aggregation in a systematic way.

In our future work, we plan to study dynamic classification confidence measures for other classifiers than quadratic discriminant classifier, mainly decision trees and support vector machines, and to study model-specific confidence measures for these classifier types. We will also incorporate local classification confidence into more sophisticated classifier aggregation methods, for example fuzzy t-conorm integral [19].

References

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[2] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[3] X. Zhu, X. Wu, and Y. Yang, "Dynamic classifier selection for effective mining from noisy data streams," in *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, (Washington, DC, USA), pp. 305–312, IEEE Computer Society, 2004.

[4] M. Aksela, "Comparison of classifier selection methods for improving committee performance.," in *Multiple Classifier Systems*, pp. 84–93, 2003.

[5] K. Woods, J. W. Philip Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 405–410, 1997.

[6] L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin, "Decision templates for multiple classifier fusion: an experimental comparison.," *Pattern Recognition*, vol. 34, no. 2, pp. 299–314, 2001.

[7] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, 1998.

[8] M. Robnik-Šikonja, "Improving random forests," in *ECML* (J. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, eds.), vol. 3201 of *Lecture Notes in Computer Science*, pp. 359–370, Springer, 2004.

[9] A. Tsymbal, M. Pechenizkiy, and P. Cunningham, "Dynamic integration with random forests.," in *ECML* (J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, eds.), vol. 4212 of *Lecture Notes in Computer Science*, pp. 801–808, Springer, 2006.

[10] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[11] D. J. Hand, *Construction and Assessment of Classification Rules*. Wiley, 1997.

[12] S. J. Delany, P. Cunningham, D. Doyle, and A. Zamolotskikh, "Generating estimates of classification confidence for a case-based spam filter," in *Case-Based Reasoning, Research and Development, 6th International Conference, on Case-Based Reasoning, ICCBR 2005, Chicago, USA, Proceedings* (H. Muñoz-Avila and F. Ricci,

eds.), vol. 3620 of *Lecture Notes in Computer Science*, pp. 177–190, Springer, 2005.

[13] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[14] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *International Conference on Machine Learning*, pp. 148–156, 1996.

[15] S. D. Bay, "Nearest neighbor classification from multiple feature subsets," *Intelligent Data Analysis*, vol. 3, no. 3, pp. 191–209, 1999.

[16] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles," *Machine Learning*, vol. 51, pp. 181–207, 2003.

[17] UCL MLG, "Elena database," 1995. http://www.dice.ucl.ac.be/mlg/?page=Elena.

[18] C. B. D.J. Newman, S. Hettich and C. Merz, "UCI repository of machine learning databases," 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[19] D. Štefka and M. Holeňa, "The use of fuzzy t-conorm integral for combining classifiers.," in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 9th European Conference, ECSQARU 2007, Hammamet, Tunisia* (K. Mellouli, ed.), vol. 4724 of *Lecture Notes in Computer Science*, pp. 755–766, Springer, 2007.