Assessing Classification Confidence Measures in Dynamic Classifier Systems

Štefka, David
2009

Assessing Classification Confidence Measures in Dynamic Classifier Systems

Post-Graduate Student:
ING. DAVID ŠTEFKA
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
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 8, CZ
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 combining is a popular technique for improving the classification quality. Common methods for classifier combining can be further improved by using dynamic classification confidence measures. In this paper, we provide a general framework of dynamic classifier systems, which use dynamic confidence measures to adapt the aggregation to a particular pattern. We also introduce methods for assessing classification confidence measures, and we experimentally show that there is a correlation between the feasibility of a confidence measure for a given dataset and a given classifier type, and the improvement of classification quality in dynamic classifier systems.

1. Introduction

Classification is a process of dividing objects (called *patterns*) into disjoint sets called *classes* [1]. A comonly used technique for improving classification quality is *classifier combining* [2] – instead of using just one classifier, a team of classifiers is created and trained; each classifier in the team predicts independently, and the classifier outputs are aggregated into a final prediction. It can be shown that such a team of classifiers can perform better than any of the individual classifiers.

A common drawback of classifier aggregation methods is that they are static, i.e., they are not adapted to the particular pattern to classify. However, if we use the concept of dynamic classification confidence (i.e., the extent to which we can "trust" the output of a particular classifier for the currently classified pattern), the aggregation algorithms can take into account the fact that "this classifier is/is not good *for this particular pattern*".

There has already been some research done in the field of dynamic classifier aggregation. Classifier selection methods [3, 4, 5] try to find out which classifier in the team is locally better than the other classifiers, and this classifier only is used for the prediction. The weakness of these methods is that much of the information is discarded, which can lead to unstability. In classifier aggregation [6, 7], where all the classifiers are used for the prediction, most of the commonly used methods are static. However, for example Robnik-Šikonja [8] and Tsymbal et al. [9] study aggregation of Random Forests with classification confidences, and Avnimelech and Intrator use dynamic aggregation of neural networks [10].

In the wider fields of classification, pattern recognition, and case-based reasoning, the classification confidence has also been studied, e.g. in [11, 12, 13]. The goal of such approaches is usually to refuse to classify a given "hard" pattern and to leave the decision to a human expert. However, in classifier combining, where we have a battery of different classifiers if one classifier refuses to classify a pattern, the classification confidence can be used more exhaustively.

It is although common that the concept of dynamic classification confidence is tightly bound with the aggregation method, or with the particular classifier type used. In this case, it is not clear whether the reported improvements are obtained due to a particular aggregation scheme, or because a dynamic classification confidence was involved in the aggregation process. Moreover, the way a classifier classifies a pattern, the way we measure confidence of a classifier, and the way we aggregate a team of classifiers, are independent on each other, so they should be studied separately.

In this paper, we provide a general framework of dynamic classifier systems, based on three independent aspects – the classifiers in the team, the confidence measures of the individual classifiers, and the aggregation strategy. This allows us to study possible benefits of using classification confidence in classifier combining, regardless of a particular classifier type, or a particular confidence measure. The confidence measures and the aggregation strategy give us three important classes of classifier systems – confidence-free (i.e., systems that do not utilize classification confidence at all), static (i.e., systems that use only "global" confidence of a classifier), and dynamic (i.e., systems that adapt to the particular pattern submitted for classification).

Apart from that, we introduce methods for assessing confidence measures, which can be used for predicting whether a dynamic classifier system will perform better than a confidence-free or static classifier system. We define two heuristics for assessing confidence measures, and we experimentally show that there is a correlation between the feasibility of a confidence measure and the improvement in the classification quality when used in a dynamic classifier system.

The paper is structured as follows. In Section 2, we present the formalism of classification itself and classification confidence, and we introduce the framework of dynamic classifier systems. In Section 3, we deal with methods how the feasibility of classification confidence measures can be measured, and we introduce two heuristics how the assessment can be done. Section 4 experimentally studies the correlation between the feasibility of a confidence measure, and the improvement in classification when used in a dynamic classifier system. Section 5 summarizes the paper and uncovers our plans for the future research.

2. Formalism of Dynamic Classifier Systems

Throughout the rest of the paper, we use the following notation. Let $\mathcal{X} \subseteq \mathbf{R}^n$ be a n-dimensional *feature space*, let $C_1, \ldots, C_N \subseteq \mathcal{X}$, $N \geq 2$ be sets called *classes*. A *pattern* is a tuple $(\mathbf{x}, c_\mathbf{x})$, where $\mathbf{x} \in \mathcal{X}$ are *features* of the pattern, and $c_\mathbf{x} \in \{1, ..., N\}$ is the index of the class the pattern belongs to. The goal of classification is to determine to which class a given pattern belongs, i.e., to predict $c_\mathbf{x}$ for unclassified patterns. We assume that for every $\mathbf{x} \in \mathcal{X}$, there is a unique classification $c_\mathbf{x}$ (e.g., provided by some expert), but when we are classifying a pattern, we do not know it – due to this fact, we will sometimes refer to a pattern only as $\mathbf{x} \in \mathcal{X}$.

Definition 1 *Let $[0, 1]$ denote the unit interval. We call a* classifier *every mapping $\phi : \mathcal{X} \to [0, 1]^N$, where for $\mathbf{x} \in \mathcal{X}$, $\phi(\mathbf{x}) = (\mu_1(\mathbf{x}), \ldots, \mu_N(\mathbf{x}))$ 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 $\mu_i(\mathbf{x}) > \mu_j(\mathbf{x})$, it means that the pattern $\mathbf{x}$ 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].*

The prediction of $c_\mathbf{x}$ for an unknown pattern $\mathbf{x}$ is done by converting the continuous d.o.c. of the classifier into a crisp output.

Definition 2 *Let ϕ be a classifier, $\mathbf{x} \in \mathcal{X}$, $\phi(\mathbf{x}) = (\mu_1(\mathbf{x}), \ldots, \mu_N(\mathbf{x}))$.* Crisp output *of ϕ on $\mathbf{x}$ is defined as $\phi^{(cr)}(\mathbf{x}) = \arg\max_{i=1,\ldots,N} \mu_i(\mathbf{x})$ if there are no ties (i.e., $|\arg\max_{i=1,\ldots,N} \mu_i(\mathbf{x})| = 1$), defined arbitrarily as $\phi^{(cr)}(\mathbf{x}) \in \arg\max_{i=1,\ldots,N} \mu_i(\mathbf{x})$ in the case of ties.*

2.1. Classification Confidence

In addition to the classifier output (the d.o.c.s), which predicts to which class a pattern belongs to, we will work with *confidence* of the prediction, i.e., the extent to which we can "trust" the output of the classifier.

Definition 3 *Let ϕ be a classifier. We call a* confidence measure *of classifier ϕ every mapping $\kappa_\phi : \mathcal{X} \to [0, 1]$. Let $\mathbf{x} \in \mathcal{X}$. $\kappa_\phi(\mathbf{x})$ is called* classification confidence *of ϕ on $\mathbf{x}$.*

Classification confidence expresses the degree of trust we can give to a classifier ϕ when classifying a pattern $\mathbf{x}$. $\kappa_\phi(\mathbf{x}) = 0$ means that the classification does not need to be correct, while $\kappa_\phi(\mathbf{x}) = 1$ means the classification is probably correct.

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

Remark 2 *Since static confidence measures are constant, independent on the currently classified pattern, we will omit the pattern* $\mathbf{x}$ *in the notation, i.e., we will denote their values just as* κ_ϕ.

In the rest of the paper, we will use the indicator operator I, defined as $I(\text{true}) = 1$, $I(\text{false}) = 0$.

2.1.1 Static Confidence Measures: After the classifier has been trained, we can use a validation set (i.e., a set of patterns the classifier has not been trained on; we could also use training patterns, but in that case, the results would be biased) to assess its predictive power as a whole (from a global view). These methods include accuracy, precision, sensitivity, resemblance, etc. [1, 14], 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 validation set:

$$\kappa_\phi^{(GA)} = \frac{\sum_{(\mathbf{y},c_\mathbf{y})\in\mathcal{M}} I(\phi^{(cr)}(\mathbf{y}) \stackrel{?}{=} c_\mathbf{y})}{|\mathcal{M}|}, \quad (1)$$

where $\mathcal{M} \subseteq \mathcal{X} \times \{1,\ldots,N\}$ is the validation set and $\phi^{(cr)}(\mathbf{y})$ is the crisp output of ϕ on $\mathbf{y}$.

2.1.2 Dynamic Confidence Measures: An easy way how a dynamic confidence measure can be defined is to compute some property on patterns neighboring $\mathbf{x}$. Let $N(\mathbf{x})$ denote a set of neighboring patterns from the validation set. In this paper, we define $N(\mathbf{x})$ as the set of k patterns nearest to $\mathbf{x}$ under Euclidean metric. Now we will define two dynamic confidence measures which use $N(\mathbf{x})$:

Euclidean Local Accuracy (ELA), used in [5], measures the local accuracy of ϕ in $N(\mathbf{x})$:

$$\kappa_\phi^{(ELA)}(\mathbf{x}) = \frac{\sum_{(\mathbf{y},c_\mathbf{y})\in N(\mathbf{x})} I(\phi^{(cr)}(\mathbf{y}) \stackrel{?}{=} c_\mathbf{y})}{|N(\mathbf{x})|}, \quad (2)$$

where $\phi^{(cr)}(\mathbf{y})$ is the crisp output of ϕ on $\mathbf{y}$.

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

$$\kappa_\phi^{(ELM)}(\mathbf{x}) = \frac{\sum_{(\mathbf{y},c_\mathbf{y})\in N(\mathbf{x})} I(\phi^{(cr)}(\mathbf{x}) \stackrel{?}{=} c_\mathbf{y})}{|N(\mathbf{x})|}, \quad (3)$$

where $\phi^{(cr)}(\mathbf{x})$ is the crisp output of ϕ on $\mathbf{x}$. The difference between (2) and (3) is that in the latter case, there is $\phi^{(cr)}(\mathbf{x})$ instead of $\phi^{(cr)}(\mathbf{y})$ in the indicator.

The dynamic confidence measures defined in this section have one drawback – they need to compute neighboring patterns of $\mathbf{x}$, 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 the degrees of classification [10, 11], 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 [15].

2.1.3 The Oracle Confidence Measure: For reference purposes, we also define a so-called *Oracle confidence measure*, which represents the "best-we-can-do" approach.

Oracle (OR) confidence is equal to 1 iff the pattern is classified correctly, 0 otherwise:

$$\kappa_\phi^{(OR)}(\mathbf{x}) = I(\phi^{(cr)}(\mathbf{x}) \stackrel{?}{=} c_\mathbf{x}) \quad (4)$$

Of course, in practical applications, we cannot use the Oracle confidence measure, because we do not know the actual class the pattern belong to ($c_\mathbf{x}$). However, the Oracle confidence measure can give us upper bound for performance of a classifier system using classification confidence, and it can also be used to assess the feasibility of a given confidence measure.

2.2. 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 own confidence measure defined.

Definition 5 *Let* $r \in \mathbf{N}$, $r \geq 2$. Classifier team *is a tuple* $(\mathcal{T},\mathcal{K})$, *where* $\mathcal{T} = (\phi_1,\ldots,\phi_r)$ *is a set of classifiers, and* $\mathcal{K} = (\kappa_{\phi_1},\ldots,\kappa_{\phi_r})$ *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*. 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* [16], *boosting* [17], *random forests* [18], *error correction codes* [2], or *multiple feature subset* methods [19].

Remark 3 *The goal of these methods is to create an ensemble of classifiers which are both* accurate *and* diverse *[20]. Here we cite only some of the basic papers about ensemble methods – in the literature, modified and improved versions of the methods can be found. In our framework, any method for creating a team (or ensemble) can be used – i.e., ensemble methods are not competitive to our approach, but they are more or less supplementary. After the classifier team has been created, the aggregation rule is totally independent of the method by which the team has been created.*

If a pattern $\mathbf{x}$ is submitted for classification, the team of classifiers gives us information of two kinds – outputs of the individual classifiers (a *decision profile*), and classification confidences of the classifiers on $\mathbf{x}$ (a *confidence vector*).

Definition 6 *Let $(\mathcal{T},\mathcal{K})$, where $\mathcal{T}=(\phi_1,\ldots,\phi_r),\mathcal{K}=(\kappa_{\phi_1},\ldots,\kappa_{\phi_r})$, be a classifier team, and let $\mathbf{x}\in\mathcal{X}$. Then we define* decision profile *$\mathcal{T}(\mathbf{x})\in[0,1]^{r\times N}$*

$$\mathcal{T}(\mathbf{x})=\begin{pmatrix}\phi_1(\mathbf{x})\\ \phi_2(\mathbf{x})\\ \vdots\\ \phi_r(\mathbf{x})\end{pmatrix}=\begin{pmatrix}\mu_{1,1}(\mathbf{x}) & \mu_{1,2}(\mathbf{x}) & \ldots & \mu_{1,N}(\mathbf{x})\\ \mu_{2,1}(\mathbf{x}) & \mu_{2,2}(\mathbf{x}) & \ldots & \mu_{2,N}(\mathbf{x})\\ & & \ddots & \\ \mu_{r,1}(\mathbf{x}) & \mu_{r,2}(\mathbf{x}) & \ldots & \mu_{r,N}(\mathbf{x})\end{pmatrix}, \tag{5}$$

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

$$\mathcal{K}(\mathbf{x})=\begin{pmatrix}\kappa_{\phi_1}(\mathbf{x})\\ \kappa_{\phi_2}(\mathbf{x})\\ \vdots\\ \kappa_{\phi_r}(\mathbf{x})\end{pmatrix} \tag{6}$$

Remark 4 *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 always be followed by $(\mathbf{x})$.*

2.3. Classifier Systems

After the pattern $\mathbf{x}$ has been classified by all the classifiers in the team, and the confidences have been 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 7 *Let $r,N\in\mathbf{N}$, $r,N\geq 2$. A* team aggregator *of dimension (r,N) is any mapping $\mathcal{A}:[0,1]^{r\times N}\times[0,1]^r\to[0,1]^N$.*

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

Definition 8 *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. The triple $\mathcal{S}=(\mathcal{T},\mathcal{K},\mathcal{A})$ is called a* classifier system. *We define an* induced classifier *of $\mathcal{S}$ as a classifier Φ, defined as*

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

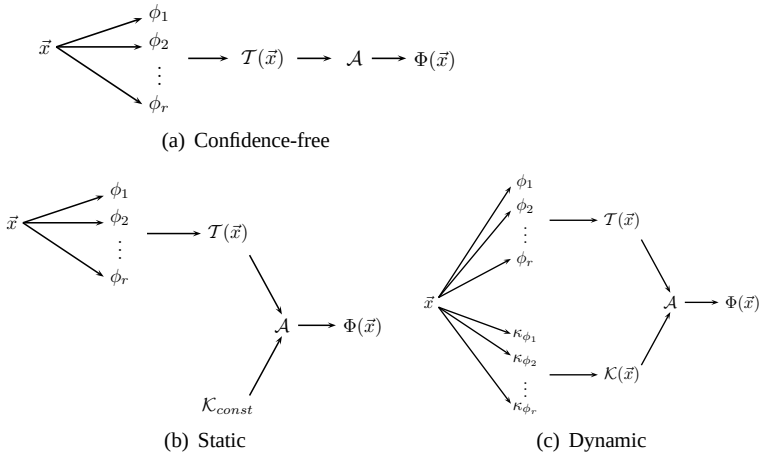
Depending on the way how a classifier system utilizes the classification confidence, we can distinguish several types of classifier systems.

Definition 9 *Let $(\mathcal{T},\mathcal{K})$ be a classifier team. $(\mathcal{T},\mathcal{K})$ is called* static, *iff $\forall\kappa\in\mathcal{K}:\kappa$ is a static confidence measure. $(\mathcal{T},\mathcal{K})$ is called* dynamic, *iff $\exists\kappa\in\mathcal{K}:\kappa$ is a dynamic confidence measure.*

Definition 10 *Let $\mathcal{A}$ be a team aggregator of dimension (r,N). We call $\mathcal{A}$* confidence-free, *iff it is constant in the second argument.*

Definition 11 *Let $\mathcal{S}=(\mathcal{T},\mathcal{K},\mathcal{A})$ 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 classifier systems do not utilize the classification confidence at all. Static classifier systems utilize classification confidence, but only as a global property (constant for all patterns). Dynamic classifier systems utilize classification confidence in a dynamic way, i.e. the aggregation is adapted to the particular pattern submitted for classification. The different approaches are shown in Fig. 1.



(a) Confidence-free

(b) Static

(c) Dynamic

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

2.3.1 Classifier Selection: Classifier selection methods [3, 4, 5] use some criterion to determine which classifier is most suitable for the current pattern $\mathbf{x}$, and the output of this classifier is taken as the final result – outputs of the other classifiers are entirely discarded.

These methods are a special case of dynamic classifier systems – the selection criterion can be viewed as a dynamic confidence measure evaluated on all the classifiers in the team, and the team aggregator $\mathcal{A}$ corresponding to the classifier selection method is defined as $\mathcal{A}(\mathcal{T}(\mathbf{x}), \mathcal{K}(\mathbf{x})) = \Phi(\mathbf{x}) = \phi_i(\mathbf{x})$, where $i \in \arg\max_{i=1,\dots,r} \kappa_{\phi_i}(\mathbf{x})$.

The weakness of classifier selection methods is that they discard much potentially useful information, which can lead to unstable results in the induced classifier's predictions [21]. In the rest of the paper, we do not deal with classifier selection.

2.3.2 Classifier Aggregation: Many methods for aggregating a team of classifiers into one final classifier have been proposed in the literature [2, 6, 7]. The simplest methods use only some simple arithmetic operation to aggregate the team's output (e.g., voting, sum, maximum, minimum, mean, weighted mean, weighted voting, product, etc.). More advanced methods use for example probability theory (e.g., behavior knowledge space [22], product rule [6], Dempster-Shafer fusion [6]), fuzzy logic (e.g., fuzzy integral [23, 24], decision templates [6, 23]), or second-level classifiers [6].

To emphasize the difference between confidence-free, static, and dynamic classifier systems, we will not consider complex aggregation algorithms, and we will define three simple aggregation algorithms, based on mean value, each representing confidence-free, static, or dynamic classifier system. This will allow us to compare the different classifier systems without bias.

We will use the notation from Def. 6 and Def. 8. Let $\Phi(\mathbf{x}) = \mathcal{A}(\mathcal{T}(\mathbf{x}), \mathcal{K}(\mathbf{x})) = (\mu_1(\mathbf{x}), \dots, \mu_N(\mathbf{x}))$, and let $j = 1, \dots, N$.

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

$$\mu_j(\mathbf{x}) = \frac{\sum_{i=1,\dots,r} \mu_{i,j}(\mathbf{x})}{r}. \tag{7}$$

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:

$$\mu_j(\mathbf{x}) = \frac{\sum_{i=1,\dots,r} \kappa_{\phi_i} \mu_{i,j}(\mathbf{x})}{\sum_{i=1,\dots,r} \kappa_{\phi_i}}. \tag{8}$$

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

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

Remark 5 *If we aggregate a team of classifiers with the Oracle confidence measure using the DWM aggregator, we obtain an* Oracle classifier *– a common reference*

classifier system, which gives us correct prediction if and only if any of its classifiers gives correct prediction. The Oracle classifier serves as the "best how classifier combining can be done" approach.

3. Assessing Confidence Measures

In [15, 25], we have experimentally shown that dynamic classifier systems of Random Forests [18] and Quadratic Discriminant Classifiers [1] using the ELA and ELM confidence measures can significantly improve the quality of classification, compared to confidence-free, or static classifier systems.

However, in these experiments, the performance of the dynamic classifier systems varied from dataset to dataset. For some datasets, the ELM confidence measure obtained better results, for others the ELA was more successful, and for some datasets, neither of them improved the classification. In other words, the performance of a dynamic classifier system is heavily influenced by the particular confidence measure used.

Given a particular dataset to classify, and given a set of classifiers which form a classifier team, there are several questions which come into one's mind:

- Will a dynamic classifier system yield improvement in the classification quality compared to confidence-free or static classifier system?
- Which confidence measure will perform the best for the given classifiers and the given dataset?
- Are the benefits of a dynamic classifier system worth the higher computational complexity?

To answer these questions, we could of course build the classifier systems and compare their performance using crossvalidation or other standard machine learning technique. However, it would be more convenient if we had some criterion of feasibility of a given confidence measure, which could answer these questions *prior* to building and crossvalidating the models. In this paper, we introduce two such criteria. Before that, we summarize the properties which should hold for a "good" confidence measure. Intuitively, if $\kappa_\phi(\mathbf{x})$ estimates the degree of trust we can give to the classifier ϕ when classifying a pattern $\mathbf{x}$, the following should be satisfied:

- If the classification confidence $\kappa_\phi(\mathbf{x})$ is high (close to 1), the classifier's prediction $\phi^{(cr)}(\mathbf{x})$ should be correct.
- If the classifier's prediction $\phi^{(cr)}(\mathbf{x})$ is not correct, the classification confidence $\kappa_\phi(\mathbf{x})$ should be low (close to 0).

For example, if $\kappa_\phi(\mathbf{x})$ is an estimate of the probability of correct classification of $\mathbf{x}$ by ϕ (for example the ELA confidence measure), both these implications are satisfied, if the estimate is good enough. According to these two properties, the ideal confidence measure is the Oracle confidence measure.

In this paper, we propose an approach in which the feasibility of a confidence measure is measured empirically, on a set of validation patterns. Let ϕ be a classifier, κ_ϕ a confidence measure, and $\mathcal{M} \subseteq \mathcal{X} \times \{1, \ldots, N\}$ the validation set. The feasibility of κ_ϕ for classifier ϕ, measured empirically on data $(\mathbf{x}, c_\mathbf{x}) \in \mathcal{M}$ will be denoted to as $\mathcal{F}(\phi, \kappa_\phi, \mathcal{M}) \in [0, 1]$. The particular methods how $\mathcal{F}(\phi, \kappa_\phi, \mathcal{M})$ can be defined will be shown in Sec. 3.2 and 3.3.

However, in classifier combining, we do not have a single classifier and its corresponding confidence measure – we have a set of classifiers $\mathcal{T}$, and a set of corresponding confidence measures $\mathcal{K}$. Therefore, we define $\mathcal{F}(\mathcal{T}, \mathcal{K}, \mathcal{M}) \in [0, 1]$ as the average feasibility of $\kappa_\phi \in \mathcal{K}$ for the corresponding classifier $\phi \in \mathcal{T}$, measured on $\mathcal{M}$:

$$\mathcal{F}(\mathcal{T}, \mathcal{K}, \mathcal{M}) = \frac{\sum_{\phi \in \mathcal{T}} \mathcal{F}(\phi, \kappa_\phi, \mathcal{M})}{|\mathcal{T}|}. \qquad (10)$$

3.1. Restricting the Validation Set

There is one more important aspect in which assessing the feasibility of a confidence measure differs in the context of classifier systems. If we measure $\mathcal{F}(\phi, \kappa_\phi, \mathcal{M})$ on the whole validation set $\mathcal{M}$, we have an estimate how κ_ϕ predicts the classification confidence *for a single classifier*. However, if we want to assess a confidence measure's performance in the context of dynamic classifier systems, we need to know something different: can this particular confidence measure improve the prediction of the classifier system?

What is the difference between these two information? A typical situation in classifier aggregation is as follows: for most patterns, the crisp outputs of the individual classifiers in a classifier system show consensus on a certain class (i.e., a vast majority of the classifiers predicts one particular class), and the team aggregator is not able to break this consensus, even when incorporating the classification confidences. Therefore, the behavior of the confidence measures on such patterns

is totally irrelevant. On the other hand, for patterns where there is no such consensus, the behavior of the confidence measure is *much* more important. Therefore, we need to identify such patterns, and restrict $\mathcal{M}$ to a such subset.

Let $0 \leq s \leq r$, where $r = |\mathcal{T}|$. Let $U(s) \subseteq \mathcal{M}$ be the set of patterns $(\mathbf{x}, c_{\mathbf{x}})$, for which for all classes C_j, $j = 1, \ldots, N$, we have

$$|\{i; i = 1, \ldots, r,\ \phi_i^{(cr)}(\mathbf{x}) = j\}| \leq s. \qquad (11)$$

$U(s)$ denotes set of patterns, for which at most s classifiers vote for any particular class. For lower s, this means that there is no consensus on a particular class, and so the team aggregator can easily use the classification confidence to improve the prediction – this suggests that restricted validation set for lower s are more important for the analysis. However, the smaller s, the smaller $|U(s)|$, which leads us to the fact that we need s big enough so the feasibility is measured on enough data. To solve the dilemma, we use the following heuristic: choose smallest s, for which $U(s)$ covers a given portion (5-10%) of the validation data, i.e., $|U(s)| \geq \alpha|\mathcal{M}|$, where $\alpha \in (0, 1)$.

3.2. Similarity to OR

The first approach how $\mathcal{F}(\phi, \kappa_\phi, \mathcal{M})$ can be measured is to compute the similarity of values $\kappa_\phi(\mathbf{x})$ to the values of the Oracle confidence $\kappa_\phi^{(OR)}(\mathbf{x})$ for patterns $(\mathbf{x}, c_{\mathbf{x}}) \in \mathcal{M}$, where $\mathcal{M}$ is the (restricted) validation set. This can be done by taking the average absolute value of the differences of the confidences:

$$\mathcal{F}^{(SOR)}(\phi, \kappa_\phi, \mathcal{M}) = 1 - \frac{\sum_{(\mathbf{x}, c_{\mathbf{x}}) \in \mathcal{M}} |\kappa_\phi(\mathbf{x}) - \kappa_\phi^{(OR)}(\mathbf{x})|}{|\mathcal{M}|}. \qquad (12)$$

3.3. AUC for OK/NOK Histogram

The second approach how $\mathcal{F}(\phi, \kappa_\phi, \mathcal{M})$ can be measured is to analyze histograms of $\kappa_\phi(\mathbf{x})$ for patterns classified correctly by ϕ (*OK patterns*) and for patterns classified incorrectly by ϕ (*NOK patterns*). Values of $\kappa_\phi(\mathbf{x})$ for the OK patterns should be concentrated near 0, while for the NOK patterns, $\kappa_\phi(\mathbf{x})$ should concentrate near 1. Moreover, these two distributions should not overlap.

Let $\mathcal{M}$ be the (restricted) validation set, and let $\mathcal{M}_i \subseteq \mathcal{M}$ for $i = 1, \ldots, N$ denote the sets of validation patterns from class C_i. For two arbitrary classes C_k, C_j, we define the multiset

$$H_{kj} = \{\kappa_\phi(\mathbf{x}) | (\mathbf{x}, c_{\mathbf{x}}) \in \mathcal{M}_k,\ \phi^{(cr)}(\mathbf{x}) = j\}, \qquad (13)$$

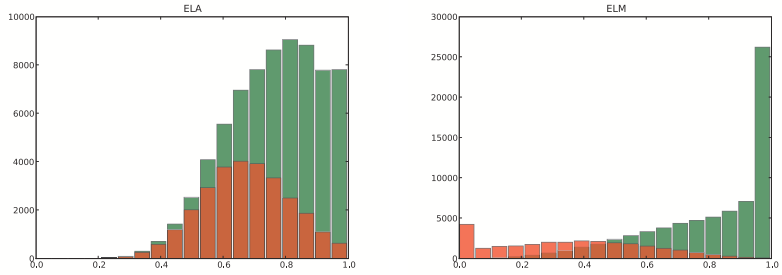
as a multiset of classification confidence values for all validation patterns from class C_k, which have been classified to class C_j by ϕ. Using this notation, we can define the *OK histogram* as the histogram computed from $\bigcup_k H_{kk},\ k = 1, \ldots, N$ and the *NOK histogram* as the histogram computed from $\bigcup_{k \neq j} H_{kj},\ k, j = 1, \ldots, N$.

The OK and NOK histograms of the ELA and ELM confidence measures for a Random Forest ensemble for the Waveform dataset (non-restricted) are shown in Fig. 2. Fig. 3 shows the evolution of the histograms for the restricted validation set. Observe that for lower s, the histograms are very different from the histograms for higher values of s.

Altough the OK/NOK (restricted) histograms give us visual information, we need to evaluate the degree of overlapping using a single number. This is possible, if we represent the OK/NOK confidence values by a ROC curve, and then we compute the area under the ROC curve.

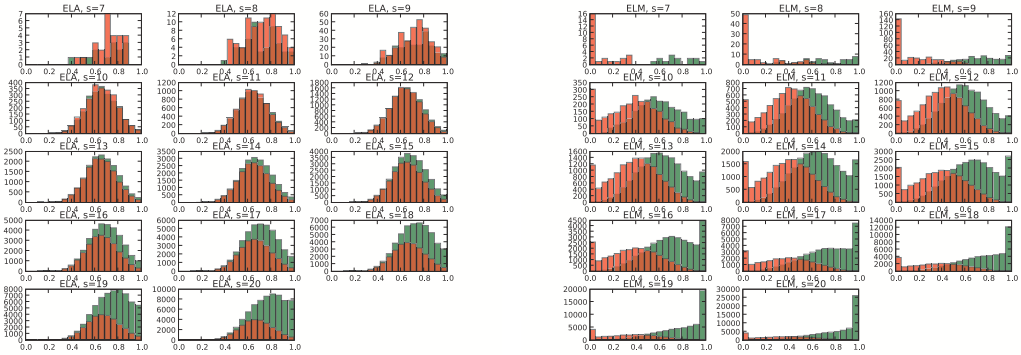
Remark 6 *Receiver operating characteristic (ROC) curves [26] are a standard tool in data mining and machine learning. ROC is basically a plot of the fraction of true positives vs. the fraction of false positives of a binary classifier, as some parameter is being varied (e.g., the discrimination threshold of the classifier). If a classifier assigns patterns to classes entirely at random, its ROC curve is the diagonal. On the other hand, for an ideal classifier, the ROC curve consist only of one point* $(0, 1)$. *The closer we are to the ROC of the ideal classifier (i.e., the farther the ROC curve is from the diagonal (above the diagonal)), the better discrimination of the classifier. The strong point of the ROC curve aprroach is that we can summarize the ROC curve into a single number – area under ROC curve (AUC) – which can be used as a criterion of quality of a binary classifier. For a random classifier, AUC=0.5, for an ideal classifier, AUC=1. The higher the AUC, the better discrimination of the classifier. Classifiers with AUC below 0.5 are actually* worse *than a random classifier.*

In the context of classification confidence, we will study the AUC of a so-called *OK/NOK classifier*, which assigns a pattern to the class "correctly classified" if the classification confidence is higher than some threshold T, and to the class "incorrectly classified" instead. By varying T between 0 and 1, we obtain the ROC curve. The AUC of the OK/NOK classifier measured on a validation set $\mathcal{M}$ (or, on a restricted set $U(s)$) can be used as an empirical property expressing the degree of



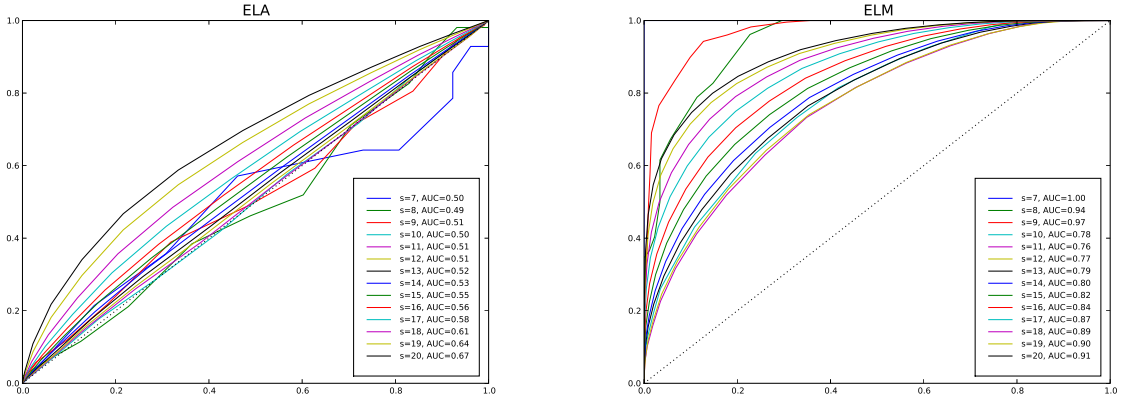
(a) ELA – bad separation (b) ELM – relatively good separation

Figure 2: The OK (green) and NOK (red) histograms of κ_ϕ of a Random Forest ensemble for the Waveform dataset.



(a) ELA (b) ELM

Figure 3: The restricted OK (green) and NOK (red) histograms of κ_ϕ of a Random Forest ensemble for the Waveform dataset for $s = 7, \ldots, 20$.



(a) ELA (b) ELM

Figure 4: The ROC curves and the AUCs of the OK/NOK classifiers for the Waveform dataset, measured on $U(s)$, $s = 7, \ldots, 20$, for a Random Forest ensemble.

overlapping of the OK and NOK distributions. Now we can define $\mathcal{F}^{(AUC)}(\phi, \kappa_\phi, \mathcal{M})$ as the AUC of the OK/NOK classifier for the confidence κ_ϕ, measured on $\mathcal{M}$. Fig. 4 shows an example of the ROCs for the ELA and ELM confidence measures for a Random Forest ensemble for the Waveform dataset.

Remark 7 *Receiver operating characteristic (ROC) curves [26] are a standard tool in data mining and machine learning. ROC is basically a plot of the fraction of true positives vs. the fraction of false positives of a binary classifier, as some parameter is being varied (e.g., the discrimination threshold of the classifier). If a classifier assigns patterns to classes entirely at random, its ROC curve is the diagonal. On the other hand, for an ideal classifier, the ROC curve consist only of one point* $(0, 1)$. *The closer we are to the ROC of the ideal classifier (i.e., the farther the ROC curve is from the diagonal (above the diagonal)), the better discrimination of the classifier. The strong point of the ROC curve aprroach is that we can summarize the ROC curve into a single number – area under ROC curve (AUC) – which can be used as a criterion of quality of a binary classifier. For a random classifier, AUC=0.5, for an ideal classifier, AUC=1. The higher the AUC, the better discrimination of the classifier. Classifiers with AUC below 0.5 are actually* worse *than a random classifier.*

4. Experiments

To find out whether the methods for assessing confidence measures described in the previous sections can really predict the improvement in the classification quality of a dynamic classifier system, we designed the following experiment. Suppose we have a classifier team $(\mathcal{T}, \mathcal{K})$. Given a dataset, we put apart 20% of the data (this was done only for the datasets which contained more than 500 patterns; for smaller datasets, we used the whole dataset) to measure $\mathcal{F}(\mathcal{T}, \mathcal{K}, \mathcal{M})$ using 5-fold crossvalidation. After that, we use the remaining data to measure the relative improvement of the error rate of a dynamic classifier system (aggregated using DWM) compared to the error rate of a confidence-free classifier system (aggregated using MV), using 10-fold crossvalidation:

$$\mathcal{I}(S_1, S_2) = \frac{Err(S_1) - Err(S_2)}{Err(S_1)}, \qquad (14)$$

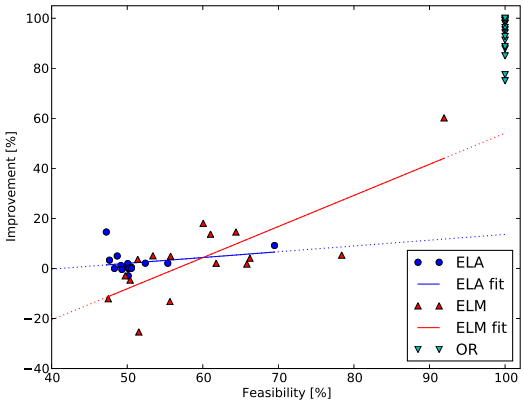
where $Err(S_1)$ denotes the error rate of the reference classifier system (using MV aggregator), and $Err(S_2)$ denotes the error rate of the dynamic classifier system (using DWM aggregator).

Our goal in this experiment was to study the correlation between $\mathcal{F}$ and $\mathcal{I}$. We performed the experiment on 5 artificial and 11 real-world datasets from the Elena database [27] and from the UCI repository [28]. The classifier teams were created using the Random Forest method [18], and as the classification confidences we used both ELA and ELM. For reference purposes, we also used the Oracle confidence measure (for which $\mathcal{F} = 1$ by definition). For assessing the confidence measures, we used methods descibed in the previous section, i.e., similarity to the Oracle confidence (SOR) and the area under ROC curve of the OK/NOK classifier (AUC), measured on the restricted validation set $U(s)$, for s such that $U(s)$ covers 5% of the data.

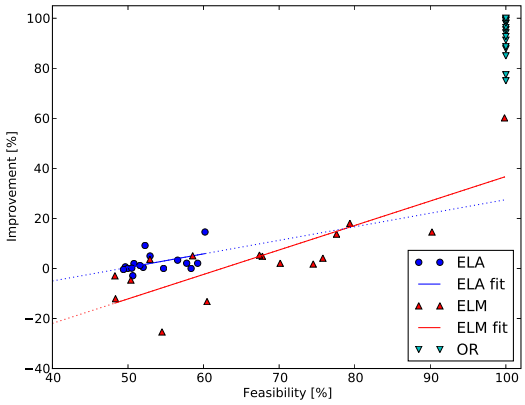
For each feasibility measure, we obtained a scatterplot of $(\mathcal{F}, \mathcal{I})$ values, which is shown in Fig. 5. We also computed a least-squares linear approximation of the scatterplot. To test the statistical significance of the results, we used the Spearman's rank correlation test [29], implemented in the Scipy Python package [30]. The Spearman's rank correlation test computes the Spearman's rank correlation coefficient $\rho \in [-1, 1]$, which expresses the degree of correlation of two variables X, Y based on their order in X and Y domains. $\rho = 0$ means there is no correlation between X and Y, $\rho = 1$ means there is a total correlation, and $\rho = -1$ indicates anticorrelation. The value of ρ is then compared to a critical value for a chosen significance level α, under the null hypothesis that there is no correlation between the variables.

For $\mathcal{F}^{(SOR)}$, the scatterplot shows a statistically significant correlation between $\mathcal{F}$ and $\mathcal{I}$ for the ELM confidence measure (at 1% significance level). For the ELA confidence measure, the correlation is not clear, and is not statistically significant. The linear least-squares fit shows that there is an increasing tendency for both confidence measures (however, much smaller for ELA). Regrettably, values of $\mathcal{F}$ for ELA are clustered mainly in the area between 50% and approx. 60%, and thus we cannot study the improvement for higher AUC values.

For $\mathcal{F}^{(AUC)}$, the scatterplot shows a statistically significant correlation between $\mathcal{F}$ and $\mathcal{I}$ for both the ELA (at 5% significance level) and ELM (at 1% significance level) confidence measures. The linear lease-squares fit shows clear increasing tendency for both confidence measures. Again, values of $\mathcal{F}$ for ELA span only the area between 50% and approx. 60%, and thus we cannot study the improvement for higher AUC values.



(a) *SOR*, ELA: $\rho = -0.07, p = 80\%$, ELM: $\rho = 0.64, p = 0.8\%$



(b) *AUC*, ELA: $\rho = 0.53, p = 3.4\%$, ELM: $\rho = 0.76, p = 0.1\%$

Figure 5: Scatterplot of $\mathcal{I}$ versus $\mathcal{F}$ for restricted validation set $U(s)$, covering 5% of the validation data for 16 datasets for the ELA, ELM, and OR confidence measures. The solid/dotted lines represent least-squares linear intrapolations/extrapolations of the data. ρ denotes the Spearman's rank correlation coefficient and p denotes the statistical significance level of the Spearman's test.

These results suggest that the methods for assessing confidence measures could be used for predicting the performance of a dynamic classifier system using classification confidence. As ELM obtains better feasibility values than ELA, the correlation between its feasibility and the improvement is more visible than for ELA. In this experiment, the AUC approach for assessing confidences showed better results than the SOR approach.

5. Summary & Future Work

In this paper, we have introduced a general framework of dynamic classifier systems, built on three main elements – the individual classifiers, their confidence measures, and the aggregator of the system. We have shown examples of one static (Global Accuracy), two dynamic (Euclidean Local Accuracy, Euclidean Local Match), and one reference (Oracle) classification confidence measures, which can be used in the framework.

We have introduced two different heuristics (the similarity to the Oracle confidence measure, and the area under ROC curve of a OK/NOK histogram) how the feasibility of a confidence measure can be assessed for a particular classifier and data. We have also shown that it is useful to compute the feasibility of a confidence measure on a set of patterns for which there is no consensus in the classifier system.

In the experiments, we have shown a correlation between the feasibility of a confidence measure and the improvement of the classification quality of a dynamic classifier system, compared to a confidence-free classifier system (at least for the OK/NOK histogram-based approach).

In our future research, we would like to study methods for assessing classification confidence measures in more detail. We would like to study deeper the way how dynamic classifier systems work and why (and when) the dynamic classification confidence can improve the classification quality.

We would also like to perform experiments with dynamic classifier systems for other classifier types than Quadratic Discriminant Classifiers and Random Forests, mainly Support Vector Machines and k-Nearest Neighbor classifiers. Apart from that, we would like to incorporate dynamic classification confidence into more advanced classifier aggregation methods, for example fuzzy t-conorm integral.

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. FÃL'rnkranz, T. Scheffer, and M. Spiliopoulou, eds.), vol. 4212 of *Lecture Notes in Computer Science*, pp. 801–808, Springer, 2006.

[10] R. Avnimelech and N. Intrator, "Boosted mixture of experts: An ensemble learning scheme," *Neural Computation*, vol. 11, no. 2, pp. 483–497, 1999.

[11] D.R. Wilson and T.R. Martinez, "Combining cross-validation and confidence to measure fitness," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN'99), paper 163*, 1999.

[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 Int. Conf., ICCBR 2005, Chicago, USA* (H. Muñoz-Avila and F. Ricci, eds.), vol. 3620 of *LNCS*, pp. 177–190, Springer, 2005.

[13] W. Cheetham, "Case-based reasoning with confidence," in *EWCBR '00: Proceedings of the 5th European Workshop on Advances in Case-Based Reasoning*, (London, UK), pp. 15–25, Springer-Verlag, 2000.

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

[15] D. Štefka and M. Holeňa, "Classifier aggregation using local classification confidence," in *Proceedings of the ICAART 2009 First International Conference on Agents and Artificial Intelligence, Porto, Portugal*, pp. 173–178, INSTICC Press, 2009.

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

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

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

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

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

[21] D. Štefka, "Confidence of classification and its application to classifier aggregation," in *Doktorandské dny KM FJFI ČVUT 2007, Prague, Czech Republic, 16. and 23. 11. 2007* (Z. Ambrož, P. Masáková, ed.), pp. 201–210, Česká technika ČVUT, 2007.

[22] Y.S. Huang and C.Y. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 1, pp. 90–94, 1995.

[23] L.I. Kuncheva, "Fuzzy versus nonfuzzy in combining classifiers designed by boosting," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 6, pp. 729–741, 2003.

[24] D. Štefka and M. Holeňa, "The use of fuzzy t-conorm integral for combining classifiers," in *Proceedings of the ECSQARU 2007 Ninth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Hammamet, Tunisia, 31.10.-02.11. 2007* (K. Mellouli, ed.), vol. 4724 of *Lecture Notes in Computer Science*, pp. 755–766, Springer, 2007.

[25] D. Štefka and M. Holeňa, "Dynamic classifier systems and their applications to random forest ensembles," in *Proceedings of the ICANNGA 2009 Ninth International Conference on Adaptive and Natural Computing Algorithms, Kuopio, Finland*,

vol. 5495 of *Lecture Notes in Computer Science*, p. 458-468, Springer, 2009.

[26] T. Fawcett, "An introduction to ROC analysis," *Pattern Recogn. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.

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

[28] 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.

[29] C. Spearman, "The proof and measurement of association between two things. By C. Spearman, 1904.," *The American journal of psychology*, vol. 100, no. 3-4, pp. 441–471, 1987.

[30] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001.