Int. J. Appl. Math. Comput. Sci., 2016, Vol. 26, No. 1, 175–189
DOI: 10.1515/amcs-2016-0012

A DYNAMIC MODEL OF CLASSIFIER COMPETENCE BASED ON THE LOCAL FUZZY CONFUSION MATRIX AND THE RANDOM REFERENCE CLASSIFIER

PAWEL TRAJDOS [a,*], MAREK KURZYNSKI [a]

[a] Department of Systems and Computer Networks
Wrocław University Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: pawel.trajdos@pwr.edu.pl

Nowadays, multiclassifier systems (MCSs) are being widely applied in various machine learning problems and in many different domains. Over the last two decades, a variety of ensemble systems have been developed, but there is still room for improvement. This paper focuses on developing competence and interclass cross-competence measures which can be applied as a method for classifiers combination. The cross-competence measure allows an ensemble to harness pieces of information obtained from incompetent classifiers instead of removing them from the ensemble. The cross-competence measure originally determined on the basis of a validation set (static mode) can be further easily updated using additional feedback information on correct/incorrect classification during the recognition process (dynamic mode). The analysis of computational and storage complexity of the proposed method is presented. The performance of the MCS with the proposed cross-competence function was experimentally compared against five reference MCSs and one reference MCS for static and dynamic modes, respectively. Results for the static mode show that the proposed technique is comparable with the reference methods in terms of classification accuracy. For the dynamic mode, the system developed achieves the highest classification accuracy, demonstrating the potential of the MCS for practical applications when feedback information is available.

Keywords: multiclassifier, cross-competence measure, confusion matrix, feedback information.

1. Introduction

A multiclassifier system or an ensemble classifier system is a set of individual classifiers whose decisions are combined in order to produce a final decision of the system (Wozniak *et al.*, 2014). There are a few main reasons for combining multiple classifiers into one classification system. First, the process of classifier learning can be seen as exploration of the hypothesis space in order to find the best hypothesis that fits the training data. A single classifier can search only a limited subspace of the hypothesis space, so harnessing a set of diverse classifiers can extend a subspace searched by the whole system (Dietterich, 2000). A limited set of training data may result in finding a set of hypotheses which achieve the same classification quality on the training/validation data. Combining outputs of these classifiers prevents the system from choosing the wrong classifier (Dietterich, 2000). The idea of building ensemble systems has been being widely explored over the last two decades and it still has a great potential (Jurek *et al.*, 2013; Dietterich, 2000; Wozniak *et al.*, 2014). Multiclassifier systems proved to be an efficient tool for solving classification problems across domains such as bioinformatics (Plumpton, 2014; Fraz *et al.*, 2012), economy (Hsieh and Hung, 2010) and many more (Wozniak *et al.*, 2014). Ensemble systems have been extensively adopted to machine learning problems such as multi-label learning (Tsoumakas *et al.*, 2010) and on-line learning (Plumpton, 2014).

Basically, the process of creating an ensemble classifier consists of two main phases: ensemble building and output combination (Dietterich, 2000; Wozniak *et al.*, 2014). The main goal of the ensemble building step is to provide the system with a set of accurate (the classification quality of an accurate classifier is higher than the quality of a random guessing) and diverse (roughly speaking, diverse classifiers make different errors on a

*Corresponding author

set of new objects) classifiers. The diversity of base classifiers is even more important than their high accuracy because extending an ensemble with new classifiers whose error patterns are identical provides no additional information to the classification committee (Dietterich, 2000). There are two common ways of building a diverse ensemble. One is to construct a heterogeneous ensemble which consists of classifiers based on different learning paradigms (Tahir *et al.*, 2012). The other is to build a set of homogeneous classifiers (the same learning paradigm) which are learned on different training sets. The most widely used methods of creating homogeneous ensembles are bagging (Breiman, 1996), boosting (Freund and Shapire, 1996) and random subspaces (Plumpton *et al.*, 2012).

The second step of the ensemble building process is to develop a combination method (a combiner). Basically, there are two methods of building the combiner, namely, output weighting methods and meta-learning (Rokach, 2010). The output weighting methods can be essentially divided into (Rokach, 2010)

- voting based (Kuncheva and Rodríguez, 2014) and support based (Kittler, 1998; Valdovinos and Sánchez, 2009),
- trainable (Kuncheva and Rodríguez, 2014) and untrainable (Kittler, 1998),
- static (Kuncheva and Rodríguez, 2014; Kittler, 1998) and dynamic (Valdovinos and Sánchez, 2009; Woloszynski and Kurzynski, 2011).

In the meta-learning methods there is a need to train at least two levels of classifiers. Those on the first level are trained using object description and the ones on higher levels are trained using outputs of classifiers from the lower level (Kuncheva, 2004; Wolpert, 1992). Sometimes, before the final output combination step is performed, a pruning step is applied. During the pruning phase inaccurate classifiers are removed from the ensemble (Woloszynski and Kurzynski, 2011; Dai, 2013).

In various practical tasks of classification we are faced with a situation in which, in the process of recognition, additional information is available about the correct/incorrect classification. This information (hereinafter called feedback information) may come from an expert who continuously monitors the recognition system and evaluates it (e.g., in medical diagnosis or industrial inspection), or may arise from a specific nature of the object being recognized. An example of the latter situation is recognition of patients' intention to move hand bioprosthesis while grasping objects based on analysis of biosignals (EMG, MMG, EEG). In this case, sensory feedback from the contact of prosthesis with the grasped object is able to provide information about the correctness of grasping movement classification or, if misclassification is made, this information can determine the group of classes of grasping movements into which the correct grip belongs (Kurzynski *et al.*, 2014).

The aim of this paper is to introduce and provide an evaluation of a novel method of classifier combination based on the original competence measure. For the calculation of the competence, various performance estimates are used, such as local accuracy estimation (Didaci *et al.*, 2005), the Bayes confidence measure (Huenupán *et al.*, 2008), multiple classifier behaviour (Giacinto and Roli, 2001), the oracle based measure (Ko *et al.*, 2008), methods based on relating that of the classifier with the response obtained by random guessing (Woloszynski *et al.*, 2012) or the randomized classification model (Woloszynski and Kurzynski, 2011), among others.

Regardless of the interpretation, the competence measure evaluates the classifier ability to correct an activity (correct classification) on a defined neighbourhood or a local region. The proposed competence measure evaluates both the local probability of correct classification and probabilities of class-dependent misclassification using the concept of a randomized reference classifier (Woloszynski and Kurzynski, 2011) and a local fuzzy confusion matrix. Such an idea of cross-competence measure allows the ensemble to exploit even the activity of incompetent classifiers instead of removing them from the ensemble. This measure can also be easily tuned in the course of a recognition process if feedback information is available.

The paper is organized as follows. Section 2 provides a mathematical model of the proposed cross-competence measure and presents an algorithm of dynamic updating of the measure. The experiments conducted and the results with a discussion are presented in Section 3. Section 4 concludes the paper.

2. Theoretical framework

2.1. Preliminaries. The multiclassifier system consists of a given set of trained classifiers $\Psi = \{\psi_1, \psi_2, \ldots, \psi_L\}$ called base classifiers. A base classifier is a function $\psi_l : \mathcal{X} \to \mathcal{M}$ that performs mapping from the feature space $\mathcal{X}$ ($\mathcal{X}$ is considered to be an n-dimensional space) to a set of class labels $\mathcal{M} = \{1, 2, \ldots, M\}$. We adopt the canonical model of a classifier (Kuncheva, 2004), which means that for a given $x \in \mathcal{X}$ the base classifier ψ_l produces a vector of class supports $d_l(x) = [d_{l1}(x), d_{l2}(x), \ldots, d_{lM}(x)]$. The $d_{lk}(x)$ is the support that classifier ψ_l gives to the hypothesis that the object x belongs to the class k. With no loss of generality, it can be assumed that the support

vector satisfies the following conditions:

$$d_{li}(x) \geq 0, \quad \forall\, l, i, \tag{1}$$

$$\sum_{i=1}^{M} d_{li}(x) = 1, \quad \forall\, l. \tag{2}$$

When the above conditions are not satisfied, the original support vector must be normalized using, for example, the soft-max rule (Kuncheva, 2004). The final decision is made according to the maximum rule

$$\psi_l(x) = \arg \max_{1 \leq i \leq M} d_{li}(x). \tag{3}$$

Now, our purpose is to propose a combining method using a trainable scheme for determining the MC system. In other words, it is assumed that a validation set

$$\mathcal{V} = \{(x_1, j_1), (x_2, j_2), \ldots, (x_N, j_N)\}, \tag{4}$$

$x_k \in \mathcal{X}, j_k \in \mathcal{M}$ containing feature vectors and their corresponding class labels is available for learning the combination function of base classifiers.

2.2. Combination function. The proposed combination function is based on an assessment of the probability of classifying an object $x \in \mathcal{X}$ to class $i \in \mathcal{M}$ by the base classifier ψ_l. Such an approach requires a probabilistic model which assumes that the result of classification $i \in \mathcal{M}$ of object x by base classifier ψ_l, true class number $j \in \mathcal{M}$ and feature vector $x \in \mathcal{X}$ are observed values of random variables $I_l(x), J, X$, respectively. Random j and x being the basis of a Bayesian model of the classification task mean that the *prior* probabilities of classes

$$P(J = j) = P(j), \quad j \in \mathcal{M}, \tag{5}$$

and class-conditional probability distribution of features

$$P(x|j) = P_j(x), \quad x \in \mathcal{X}, \tag{6}$$

exist.

Random $\psi_l(x) = i$ for a given x denotes that base classifier ψ_l is a randomized classifier which is defined by the conditional probabilities $P(\psi_l(x) = i) = P_l(i|x) \in [0, 1]$ (Berger and Berger, 1985). For a deterministic classifier, these probabilities are equal to 0 or 1.

The natural concept for the support of the j-th class is its *a posteriori* probability, which (under the adopted model) can be expressed as follows:

$$P_l(j|x) = \sum_{i=1}^{M} P_l(i, j|x) = \sum_{i=1}^{M} P_l(i|x) P_l(j|i, x), \tag{7}$$

where $P_l(j|i, x)$ denotes the probability that an object x belongs to the class j given that $\psi_l(x) = i$.

Unfortunately, the assumption that the base classifiers assign a class label in a stochastic manner has little or no practical use, and hence it should be avoided. For this reason, we replace analysis of probabilistic properties of base classifier ψ_l with its equivalent randomized form called a randomized reference classifier (RRC) (Woloszynski and Kurzynski, 2011). RRC $\psi_l^{(RRC)}$ for a given x produces random class supports whose expected values are equal to the supports produced by the modelled base classifier ψ_l. This means that $\psi_l^{(RRC)}$ acts, on average, as base classifier ψ_l; hence probabilities $P(\psi_l^{(RRC)}(x) = i) = P_l^{(RRC)}(i|x)$ can be used in (7) instead of probabilities $P_l(i|x)$, viz.

$$P_l^{(RRC)}(i|x) \approx P_l(i|x). \tag{8}$$

In turn, the approximation of probabilities $P_l(j|i, x)$,

$$m_{ji}^{(\psi_l)}(x) \approx P_l(j|i, x), \tag{9}$$

can be calculated using a local confusion matrix of ψ_l, i.e., the matrix of class-dependent frequencies of classification by ψ_l in the neighbourhood of x. Approximations $m_{ji}^{(\psi_l)}(x)$ for $i = j$ can be considered to be class-dependent competence and for $i \neq j$ interclass competence (cross-competence). This interpretation leads to the following conclusions:

- $m_{ji}^{(\psi_l)}(x)$ values are interrelated; therefore it is difficult to define a threshold of competence (usually equal to the probability of random guessing) in the combining mechanism above which the base classifier becomes a member of the classifier ensemble;
- a high value of cross-competence indicating a malfunction of the classifier (the classifier is incompetent) does not mean that the classifier should be removed from the ensemble. Information about the erroneous operation of the base classifier can be useful in making the final decision by the ensemble. For example, if the classifier instead of class 3 often indicates erroneously class 2 and for a given x support for class 2 is high, then in the mechanism of combining this fact should be transferred into increasing support for class 3.

Consequently, we get support for class j ($j \in \mathcal{M}$) produced by the base classifier ψ_l at a point x as approximated value of probability (7), namely,

$$d_{lj}(x) = \sum_{i=1}^{M} m_{ji}^{(\psi_l)}(x) P_l^{(RRC)}(i|x). \tag{10}$$

In the next subsections, methods of calculation of the approximations (8) and (9) used in this study will be presented in detail.

2.3. Randomized reference classifier. A base classifier ψ_l is modelled by a randomized reference classifier (RRC), which is a stochastic classifier defined using a probability distribution over the set of class labels $\mathcal{M}$. The RRC $\psi_l^{(RRC)}$ classifies object x according to the maximum rule (3) for a vector of class supports $[\delta_{l1}(x), \delta_{l2}(x), \ldots, \delta_{lM}(x)]$ which are observed values of random variables (rvs) $[\Delta_{l1}(x), \Delta_{l2}(x), \ldots, \Delta_{lM}(x)]$. The probability distribution of rvs is chosen in such a way that the following conditions are satisfied:

$$\Delta_{lj}(x) \in (0,1), \tag{11}$$

$$\sum_{j=1}^{M} \Delta_{lj}(x) = 1, \tag{12}$$

$$\mathrm{E}\left[\Delta_{lj}(x)\right] = d_{lj}(x), \quad j \in \mathcal{M}, \tag{13}$$

where E is the expected value. The conditions (11) and (12) follow from the normalisation properties of class supports while the condition (13) relates the RRC $\psi_l^{(RRC)}$ to base classifier ψ_l, ensuring their equivalence.

Since the RRC performs classification in a stochastic manner, it is possible to calculate the probability of classification an object x to the i-th class:

$$P_l^{(RRC)}(i|x) = \Pr\left[\forall_{k=1,\ldots,M,\ k\neq i}\ \Delta_{li}(x) > \Delta_{lk}(x)\right]. \tag{14}$$

The most important step in the process of building the RRC is the choice of probability distributions for rvs $\Delta_{lk}(x),\ k \in \mathcal{M}$ so that the conditions (11)–(13) are satisfied. In this study we use beta distributions with parameters $\alpha_{li}(x), \beta_{li}(x),\ i \in \mathcal{M}$. The justification of the choice of the beta distribution resulting from the theory of order statistics and a detailed description of parameter estimation of α, β can be found in the work of Woloszynski and Kurzynski (2011).

For the beta distribution, we get the following formula for the probability (14):

$$P_l^{(RRC)}(i|x) = \int_0^1 b(u, \alpha_{li}(x), \beta_{li}(x)) \times \left[\prod_{\substack{j=1 \\ æ\neq i}}^{M} B(u, \alpha_{lj}(x), \beta_{lj}(x)\right] \mathrm{d}u, \tag{15}$$

where $B(\cdot)$ is a beta cumulative distribution function. The MATLAB code for calculating the probabilities (15) was developed and is freely available for download (Woloszynski, 2013). It should be noted that to determine the probability (15), a validation set is not necessary, because it does not need to know the correct classification of the object x.

2.4. Local confusion matrix. A confusion matrix (CM) gives the complete picture of correct and incorrect classification made by classifiers ψ for separate classes (Devroye *et al.*, 1996). The rows (columns) correspond to the true classes (results of classification made by classifier ψ_l), as shown in Table 1. The values of elements in the matrix depend on the adopted model of classification and data available. For the probabilistic model with known probability distributions (5), (6) and deterministic classifier ψ_l, matrix elements can be calculated as follows ($i, j \in \mathcal{M}$):

$$\varepsilon_{j,i}^{(\psi_l)} = P(i,j) = P(j) \int_{\mathcal{X}} P(x|j) R_i^{(\psi_l)}(x)\, \mathrm{d}x, \tag{16}$$

where

$$R_i^{(\psi_l)}(x) = \begin{cases} 1 & \text{if } \psi_l(x) = i, \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

signifies the indicator of decision region $\mathcal{D}_i^{(\psi_l)} = \{x \in \mathcal{X} : \psi_l(x) = i\}$ of classifier ψ_l.

Table 1. Multiclass confusion matrix of classifier ψ_l.

		classification by ψ_l			
		1	2	...	M
true class	1	$\varepsilon_{1,1}^{(\psi_l)}$	$\varepsilon_{1,2}^{(\psi_l)}$	...	$\varepsilon_{1,M}^{(\psi_l)}$
	2	$\varepsilon_{2,1}^{(\psi_l)}$	$\varepsilon_{2,2}^{(\psi_l)}$	...	$\varepsilon_{2,M}^{(\psi_l)}$
	$\vdots$	$\vdots$	$\vdots$		$\vdots$
	M	$\varepsilon_{M,1}^{(\psi_l)}$	$\varepsilon_{M,2}^{(\psi_l)}$	...	$\varepsilon_{M,M}^{(\psi_l)}$

If, as in this study, a randomized classifier is used, then

$$R_i^{(\psi_l)}(x) = P_l^{(RRC)}(i|x), \tag{18}$$

and decision region $\mathcal{D}_i^{(\psi_l)}$ can be interpreted as a fuzzy set with its membership function equal to $P_l^{(RRC)}(i|x)$.

In the real world, the probability distributions (5) and (6) are not known, while the validation set (4) is available. In such a case, entries of the confusion matrix must be defined so as to constitute an empirical approximation of theoretical entries (16). For this purpose, let us first define the following subsets of the validation set in the common convention of fuzzy sets:

$$\mathcal{V}_j = \left\{(\mu_{\mathcal{V}_j}(x_k) = 1, x_k) : x_k \in \mathcal{V} \wedge j_k = j\right\}, \tag{19}$$

which denotes the subset of validation objects belonging to the j-th class ($j \in \mathcal{M}$), and

$$\mathcal{D}_i^{(\psi_l)} = \left\{(\mu_{\mathcal{D}}(x_k) = P_l^{(RRC)}(i|x_k), x_k) : x_k \in \mathcal{V}\right\}, \tag{20}$$

which is decision set ($i \in \mathcal{M}$) of randomized classifier ψ_l in the validation set $\mathcal{V}$.

The sets (19) and (20) can be used to approximate confusion matrix entries (16), namely,

$$\hat{\varepsilon}_{j,i}^{(\psi_l)} = \frac{\left|\mathcal{V}_j \cap \mathcal{D}_i^{(\psi_l)}\right|}{\sum_{j=1}^{M} |\mathcal{V}_j|} = \frac{\sum_{x_k \in \mathcal{V}_j} P_l^{(RRC)}(i|x_k)}{N}, \tag{21}$$

where $|A|$ is the cardinality of a fuzzy set A (Mamoni, 2013).

The confusion matrix for a point $x \in \mathcal{X}$ or a local confusion matrix has the structure shown in Table 1, while matrix entries describe the correct and incorrect classification for separate classes made by classifier ψ_l only in the neighbourhood $\mathcal{N}(x)$ of point $x \in \mathcal{X}$. This means that, in order to calculate matrix element $\varepsilon_{j,i}^{(\psi_l)}(x)$ according to the formula (16), instead of the integral over the whole feature space $\mathcal{X}$ one must take the integral over the neighbourhood $\mathcal{N}(x)$ of the given point x.

For the empirical case when validation set $\mathcal{V}$ is given, defining, as before, the neighbourhood $\mathcal{N}(x)$ as a fuzzy set in the validation set,

$$\mathcal{N}(x) = \left\{(\mu_{\mathcal{N}(x)}(x_k), x_k), x_k \in \mathcal{V}\right\}, \tag{22}$$

we get the following formula for the approximation of local confusion matrix entry $\varepsilon_{j,i}^{(\psi_l)}(x)$:

$$\hat{\varepsilon}_{j,i}^{(\psi)}(x) = \frac{\left|\mathcal{V}_j \cap \mathcal{D}_i^{(\psi_l)} \cap \mathcal{N}(x)\right|}{\sum_{j=1}^{M} |\mathcal{V}_j|}. \tag{23}$$

The key element of the approximation (23) is the definition of neighbourhood $\mathcal{N}(x)$ or its membership function in (22). In this study, the Gaussian membership function is adopted,

$$\mu_{\mathcal{N}(x)}(x_k) = \exp(-\gamma\delta(x_k, x)^2), \tag{24}$$

where $\gamma \in \mathbb{R}^+$ and $\delta(x_k, x)$ is the Euclidean distance between x_k and x. The preliminary experimental evaluation showed that the best results are obtained when γ is set to 1. Such a neighbourhood model means that $\text{supp}(\mathcal{N}(x)) = \mathcal{V}$ and $\mu_{\mathcal{N}(x)}(x_k)$ is a membership function decreasing with the increasing distance between x and x_k.

Finally, from (23) and (24) we get

$$\hat{\varepsilon}_{j,i}^{(\psi_l)}(x) = \frac{\sum_{x_k \in \mathcal{V}_j} P_l^{(RRC)}(i|x_k) \exp(-\gamma\delta(x_k, x)^2)}{N}. \tag{25}$$

and the approximation (9) of $P_l(j|i, x)$ can be calculated as a normalized value of (25), namely,

$$m_{ji}^{(\psi_l)}(x) = \frac{\hat{\varepsilon}_{j,i}^{(\psi_l)}(x)}{\sum_{j=1}^{M} \hat{\varepsilon}_{j,i}^{(\psi_l)}(x)}. \tag{26}$$

The block-diagram of the proposed method for calculating the probabilities (7) as class supports of base classifier (ψ_l) at a test point x is presented in Fig. 1. An important role in the algorithm is played by randomized reference classifier, used for calculation of approximation of probabilities $P(\psi_l = i|x)$ and for the confusion matrix. A characteristic feature of

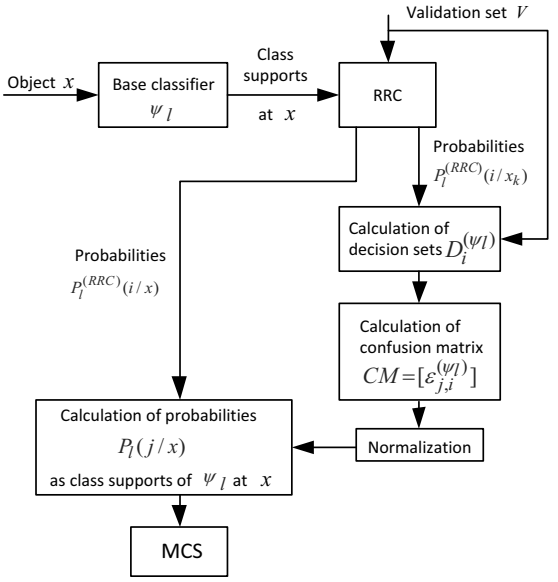


Fig. 1. Block-diagram of the proposed method for calculating class supports of the base classifier.

the proposed method is that it integrates in a single discriminant function both the paradigm of the base classifier activity and its ability of correct classification (competence) evaluated using a validation set. This means that in the combining procedure of the MCS only the majority voting scheme (on the class or the support level) can be implemented. For comparison, the flowchart of a literature method (Woloszynski and Kurzynski, 2011) for combining base classifiers in an MCS is shown in Fig. 2. The method, although uses the same concept of the RRC, is based on a completely different scheme. In this approach the competence of the base classifier is calculated and the whole procedure can be divided into the following two steps. First the set of competences at all points of the validation set is calculated; second, the competence measure (function) of the classifier is constructed. This construction is based on extending (generalizing) the competence set to the entire feature space. In other words, this step can be considered a problem of supervised learning of the competence function using the competence set. So, as a result, we separately obtain class supports produced by base classifier ψ_l and its competence at the point x. Consequently, in the MCS, both selection (according to the DCS and the DES scheme) and fusion methods can be used. From this point of view, the method developed in this study is less flexible, but one should remember that when applying this method we can use all base classifiers (even incompetent) and therefore selection procedure is not necessary.

Given (7) it is obvious that $P_l(j|x)$ can also be approximated using only the local confusion matrix.

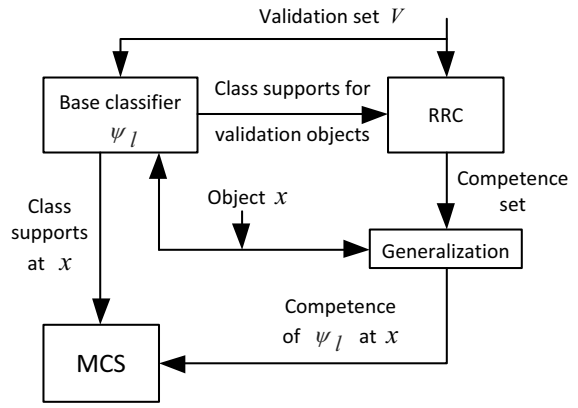


Fig. 2. Block-diagram of the RRC-based reference method for calculating the competence of the base classifier.

However, it was decided to employ the way of approximation using the point approximation (15). This decision was made due to the fact that approximation provided solely by the confusion matrix is less accurate because the approximation of $P_l(i|x)$ provided by (15) is a point approximation. On the other hand, the approximation obtained using solely the confusion matrix is utilized (see Eqn. (7)) when random variables I and J are conditionally independent given x (a classifier whose output is modelled as I is a random guessing classifier).

2.5. Competence tuning based on feedback information. A method of competence tuning for the RRC was proposed by Kurzynski and Wolczowski (2012). The approach is a particular case of the on-line learning problem (Blum, 1998). First, the set of class labels $\mathcal{M}$ is divided into disjoint subsets $\mathcal{M}_k$ by domain experts. In training and validation stages, the ensemble is learned and the competence model is built. Then, in the classification stage, it is assumed that feedback information is available, which provides us with information related to correct/incorrect classification. When the classification outcome is correct, the feedback information confirms the prediction. In the case of incorrect classification, the feedback delivers the number of the subset $\mathcal{M}_k$ to which the classified object belongs.

The method proposed in this paper can also be tuned using feedback information. The algorithm of tuning the combination function consists in adding new objects into the validation set. Object x_n is located in an appropriate fuzzy set $\mathcal{V}_j$ with membership degree $\mu_{\mathcal{V}_j}(x_n)$ depending on feedback information. If object x_n was correctly classified to class j, i.e., $i_n = j$, then $\mu_{V_j}(x_n) = 1$. On the other hand, if we know from the feedback information that the object x_n belongs to the class from the set $\mathcal{M}_k$ and its classification was incorrect, then the value of $\mu_{\mathcal{V}_j}(x_n)$ is $1/|\mathcal{M}_k|$ for all classes belonging to the set $\mathcal{M}_k$.

Algorithm 1. Pseudocode of validation set update: FCM.

Require: x: classified point,
ψ_l: base classifier,
i: result of classification of x by ψ_l,
j: true class of x,
$\mathcal{M}_k$: subset of the classes determined by feedback information

if $i = j$ **then**
  $\mu_i(x) := 1$
  $\mu_n(x) = 0,\ \forall n \in \{1, 2, \ldots, M\} \setminus i$
else if $i \in \mathcal{M}_{k(x)}$ **then**
  $\mu_i(x) := 0$
  $\mu_n(x) := \frac{1}{|\mathcal{M}_{k(x)}|-1},\ \forall n \in \mathcal{M}_{k(x)} \setminus i$
  $\mu_m(x) := 0,\ \forall m \notin \mathcal{M}_{k(x)}$
else
  $\mu_n(x) := \frac{1}{|\mathcal{M}_{k(x)}|},\ \forall n \in \mathcal{M}_{k(x)}$
  $\mu_m(x) := 0,\ \forall m \notin \mathcal{M}_{k(x)}$
end if
$\mathcal{V} := \mathcal{V} \cup (x, (\mu_n(x)), n \in \{1, 2, \ldots, M\}$
return $\mathcal{V}$ {Updated validation set}

The whole procedure of extending validation set $\mathcal{V}$ during the classification stage is presented in Algorithm 1 in detail.

2.6. Computational complexity. This section is devoted to theoretical analysis of computational and storage complexity of the proposed method (we denote this approach by the FCM, which stands for the fuzzy confusion matrix). Additionally, the complexity of the FCM system is compared against the method proposed by Woloszynski and Kurzynski (2011) (RRCS: randomized reference classifier system). During the analysis, we provide a description of the complexity of each of four main stages of the procedures, that is, training, validation, inference and parameter tuning.

Before we proceed, let us make a set of assumptions which are aimed at simplifying the analysis. First of all, we study multiclassifier systems based on homogeneous ensembles. Each base classifier of these committees is considered to be trained using a bootstrap sample taken from the training set $\mathcal{T}$, and the cardinality of the samples is equal to that of the original training set $|\mathcal{T}|$. As a consequence, the complexity of the training, inference and storage of the committee, which consists of L base classifiers, is $O\left(L \times c_t(|\mathcal{T}|, M, n)\right)$, $O\left(L \times c_i(|\mathcal{T}|, M, n)\right)$ and $O\left(L \times c_m(|\mathcal{T}|, M, n)\right)$, respectively. The quantities $c_t(|\mathcal{T}|, M, n))$, $c_i(|\mathcal{T}|, M, n)$ and $c_m(|\mathcal{T}|, M, n)$ represent respectively training, inference and storage complexities of each base classifier as functions of the training set cardinality, the number of classes and the dimensionality of the input space. The functions are specific to the base

classifier upon which the committee is built. For example, if we consider a naive (including computation of all distances and finding nearest neighbours using the quick sort algorithm) implementation of the KNN algorithm, we get $c_t(|\mathcal{T}|, M, n) = 1$, as well as $c_i(|\mathcal{T}|, M, n) = |\mathcal{T}|n + |\mathcal{T}| \times \log_2(|\mathcal{T}|)$ and $c_m(|\mathcal{T}|, M, n) = |\mathcal{T}|n$.

The next phase is a validation procedure during which a competence set is formed. In order to construct this set, we need to get outcomes of the base classifiers for each instance in the validation set $\mathcal{V}$. The computational burden of this operation follows $O\left(L \times |\mathcal{V}| \times c_i(|\mathcal{T}|, M, n)\right)$. After that, both the methods considered calculate probabilities $P_l^{(RRC)}(i|x)$ according to Eqn. (15). The RRCS method calculates only the probability of correct classification while the proposed one computes probabilities for each class. Taking this into consideration, the complexity is proportional to $O(L \times |\mathcal{V}| \times S \times M)$ and $O(L \times |\mathcal{V}| \times S \times M^2)$, respectively, where S is the length of the sequence which is used to perform numerical integration. Since the FCM method incorporates a more complex competence model, its storage complexity ($O(|\mathcal{V}| \times [d + L \times M])$) is greater than the complexity of the original RRC approach ($O(|\mathcal{V}| \times [d + L])$).

In this paragraph we examine the computational burden related to the classification of a single instance. Similarly to the above-mentioned procedure, this one begins with obtaining the outcome of base classifiers ($O\left(L \times c_i(|\mathcal{T}|, M, n)\right)$). After that, the competence set is employed to produce a final result. The RRCS uses a general measure of competence, which is calculated as a weighted mean of the competence coefficient related to the points that constitute the competence set. The weights are calculated using the Gaussian potential function (24). To compute this mean value, the number of operations proportional to $O(|\mathcal{V}|[n + L])$ is required, and then the final outcome takes $O(M \times L)$ calculations. On the other hand, the FCM system calculates class-specific measures of competence and cross competence. As a result, the complexity of this phase grows to $O(|\mathcal{V}|[n + M^2 L])$. The final support for this system is produced according to Eqn. (10), which requires the number of operations that follows $O(M^2 L)$

The complexity of the tuning procedure (for a single instance) is identical for both investigated methods, and it follows $O(L \times \left[c_i(|\mathcal{T}|, M, n) + S \times M^2\right])$.

3. Experiments

3.1. Experimental setup. The experimental study was generally divided into two main stages. The first one was aimed at comparing the static mode of the proposed approach against state-of-the-art methods of competence evaluation. The goal for the second stage was to assess the effectiveness of utilization of feedback information. Since the parameter tuning procedure is performed during the classification phase, we called this approach dynamic mode. A detailed description of the performed experimental study is provided in the following subsections.

3.1.1. Static mode. Most of the benchmark data sets used in the experimental study were obtained from the UCI Machine Learning Repository (Bache and Lichman, 2013). The original names of some sets from the repository were shortened, i.e., wine quality red (wq_red), wine quality white (wq_white), multiple features data set (mfdig_x), Hill-Valley (HillVall), banknote authentication (bank_auth), Urban Land Cover (ULC). The acute set refers to the acute abdominal pain diagnosis problem and comes from the Orthopaedic and Traumatologic Surgery Clinic of Wrocław Medical Academy, and it was described by Kurzynski (1987). During the prepossessing stage, the datasets were normalized to have zero mean and unit variance. Additionally, classes of lowest cardinalities from the Ecoli, wq_white and wq_red datasets, were removed. The training and testing sets were extracted from original datasets using a ten-fold stratified cross-validation. Table 2 shows summary information related to transformed sets.

During the experiments, homogeneous and heterogeneous classifier ensembles were evaluated. The heterogeneous ensemble consists of the following classifiers: the pruned tree classifier (Gini splitting index) (Breiman *et al.*, 1984), k-nearest neighbours classifiers (k-NN) (Cover and Hart, 1967) with $k = 5, 10, 15$, a single layer perceptron network with the number of neurons in the hidden layer set to $N_h = 5, 10$, a two-layer perceptron network with the number of neurons in both the hidden layers set to $N_{hh} = 5, 10$, respectively (Bishop, 1995), a linear SVM classifier and SVM classifiers with radial, quadratic, sigmoid and polynomial kernels (Scholkopf and Smola, 2001). Base classifiers of the heterogeneous were trained using the original training set. The homogeneous ensembles also consist of 20 classifiers.

We used the same types of classifiers as in the case of the heterogeneous ensembles. That is, the first homogeneous ensemble is formed using 20 tree classifiers, the next one consists on 20 5-NN classifiers, and so on. Each base classifier was trained using a randomly selected bagging sample from the original dataset. The proposed multiclassifier system was compared with five state-of-the-art multiclassifier systems. The first of them was a system with a non-trainable combiner, namely, a simple mean combiner (Kittler, 1998). The second method was the DES-CS (Woloszynski and Kurzynski, 2011) system based on the RRC classifier, and the remaining ones were

the Dudani, Shepard and average distance weight (ADW) combiners described by Valdovinos and Sánchez (2009).

During the experimental study the proposed method is denoted by the FCM (fuzzy confusion matrix). We also evaluated the ability of the proposed method to eliminate the impact of inaccurate classifiers. This property is particularly important for a multiclassifier system that does not exclude the outcome of incompetent classifiers, so we decided to perform an additional experiment to assess the efficiency of the proposed method of dealing with these classifiers. In order to conduct the aforesaid evaluation, we employed ensembles that consist of inaccurate random classifiers. Each of these classifiers assigns an object to a class according to the uniform probability distribution. Under such circumstances, the prediction ability of the whole system relies only on the conditional probability estimation $P_l(i|x)$ computed on the basis of the fuzzy confusion matrix. From this perspective, the FCM can be seen as a kind of lazy classifier.

3.1.2. Dynamic mode. For practical reasons, the competence tuning method was evaluated using a subset of original datasets. We eliminated binary classification sets and sets in which the number of instances per class was too low. The names of the selected sets are highlighted in boldface in Table 2. In order to perform dynamic parameter tuning, the original sets were modified by adding artificial class groups. The class groups were created using the following procedure. First, positions of class centroids were computed. Then the centroids were clustered using the hierarchical clustering algorithm (Rokach and Maimon, 2005), and Ward's criterion was used as a merging criterion (Ward, 1963). Assuming that the optimal number of clusters lies between 2 and $M-1$, we determined the number of clusters using the average silhouette index (Rousseeuw, 1987).

Anticipating the discussion related to the outcome of the static experiment, we can say that both the proposed static combiner and the RRC combiner achieved the best performance using tree-based classifiers.

As a consequence, we decided to present only results obtained using the aforementioned base classifiers. As was mentioned in the previous subsection, the experiments were conducted using a homogeneous ensemble which consists of 20 base classifiers. To assess the effectiveness of the dynamic mode of the investigated classifiers, we applied an experimental procedure based on the methodology which is characteristic of data stream classification. Namely, the test-then-update procedure that uses data chunks was employed (Gama, 2010). The evaluation was conducted as follows. First, the dataset was divided into 20 non-overlapping subsets using stratified cross-validation. Then, the first chunk was used to train multiclassifier systems. The subsequent chunks

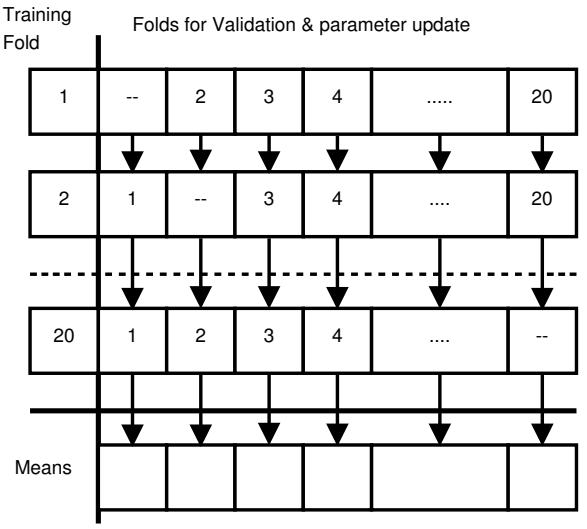


Fig. 3. Fold-wide means computation.

were used to evaluate the classification performance, and after that we utilized them to tune competence sets of the evaluated systems. The procedure was repeated 20 times: in each turn, a new fold was used as a learning set. Finally, according to Fig. 3, the mean classification error was calculated for each fold. The obtained means are compared by the Wilcoxon signed rank test (Wilcoxon, 1945; Demšar, 2006), and the significance level for statistical comparison was set to $\lambda = 0.05$. We also compared the relative difference between non-updatable classifier and its updatable version. The difference was computed using the following formula (fold and iteration indices were omitted):

$$\mathrm{rd} = \begin{cases} \dfrac{e - e_u}{e} & \text{if} \quad e > 0, \\ e - e_u & \text{if} \quad e = 0, \end{cases} \tag{27}$$

where e and e_u stand for the non-updatable and the updatable classifier error, respectively. The relative differences were computed for fold-wide means (Fig. 3). Vectors of differences were compared using the two-tailed paired Wilcoxon test. The comparison in terms of the relative difference provides information about the improvement achieved by the updatable version of the classifier. We decided to compare updatable versions of the combiners using the relative difference because the comparison in terms of mean error is biased towards the classifier whose untrainable version obtained better performance.

Statistical evaluation of the obtained results was assessed using the Wilcoxon signed rank test (p-values were corrected using Holm's procedure (Holm, 1979)), the Friedman test (Friedman, 1940) and the two-tailed Bonferroni–Dunn post-hoc test (Dunn, 1961; Demšar, 2006).

All of the experiments were conducted using the R

Table 2. Dataset summary.

Name	#dim	#object	#class	Name	#dim	#object	#class
iris	4	150	3	**semeion**	256	1593	10
wine	13	177	3	**mfdig_fou**	76	2000	10
wq_white	11	3651	3	**mfdig_fac**	216	2000	10
wq_red	11	1278	3	**mfdig_zer**	47	2000	10
acute	31	476	8	**mfdig_pix**	240	2000	10
Seeds	7	210	3	**mfdig_kar**	64	2000	10
Ecoli	7	327	5	**mfdig_mor**	6	2000	10
Faults	27	1940	7	**ULC**	146	675	9
Vertebral Column	6	310	3	bank_auth	4	1371	2
Breast Tissue	9	105	6	ionosphere	34	351	2
pima	8	767	2	spectF	44	267	2
Glass	9	213	6	fertility	9	100	2
HillVall	100	1212	2				

environment (R Core Team, 2012).

3.2. Results and discussion.

3.2.1. Static mode. Due to space limitation, we present only summary results connected to classification performance of multiclassifier systems build upon different base classifiers. The summarized outcome is shown in Table 3. The table contains mean ranks (average across the data sets) achieved by the evaluated methods. Additionally, the average ranks are also visualized in Fig. 4. The Friedman test and the post-hoc Dunn–Bonferroni tests confirmed that in the case of such base classifies as the KNN and MLP there was no significant difference between the evaluated combiners.

On the other hand, the conducted experiments demonstrated that for the tree-based and SVM-sigmoid base classifiers the proposed method is in the group of classifiers which achieved significantly better results. In contrast to the above-mentioned outcome, in the case of the heterogeneous and SVM-radial classifiers, the classification quality obtained by the proposed method is significantly lower. The paired test showed that the performance of the proposed method does not differ significantly among most of benchmark sets. Taking these results under consideration, we can conclude that the proposed algorithm obtained the highest classification quality for the tree-based ensemble and the lowest performance is achieved for the heterogeneous committee.

More precise results for these ensembles are provided in Tables 4 and 5. A detailed look at the presented tables leads to a conclusion that the proposed model of class-dependent competence and cross-competence is sensitive to the form of the decision sets produced by base classifiers. This phenomenon can be observed in Table 3, and the best examples are SVM-based classifiers. The results indicate that the change of the kernel function leads to a substantial change in overall performance. What is more, the introduced procedure is unable to take benefits of combining base classifiers built upon different learning paradigms. However, in general, the FCM approach achieved a classification quality comparable to that to state-of-the-art algorithms.

The experimental study confirmed our claim that the proposed method has the ability to correct the outcome of an inaccurate classifier (see Section 2.4 and Eqn. (7)). The mean error rates presented in Table 6 clearly showed that even in the worst case scenario (that is, the ensemble consists of random guessing classifiers) the performance of the introduced algorithm is significantly higher than that of remaining approaches. What is more, results obtained by the FCM combiner are comparable to those obtained using non-random-guessing ensembles (Tables 4 and 5). However, it must be noted that in most real-world applications ensembles do not contain inaccurate classifiers (Dietterich, 2000). The experiment also revealed that the conditional probability estimation computed using the fuzzy confusion matrix can be considered a standalone MAP (maximum a posteriori) classifier. In consequence, inaccurate classifiers were eliminated from the ensemble by substituting their predictions by outcomes of the corresponding lazy classifiers, so the quality of classification was higher in comparison with methods which simply remove the inaccurate classifier. On the other hand, it should be emphasized that in the scenario considered the diversity of the system must have been low. The reason behind this situation is the fact that all random-guessing base classifiers were evaluated on a single validation set. Consequently, all conditional probabilities were calculated using the same model.

3.2.2. Dynamic mode. The results of the experiments related to ensemble parameter tuning are shown in Tables 7 and 8. The header of each table contains,

Table 3. Classification quality: mean ranks (lower is better). The critical difference (Bonferroni correction) is $CD_{\lambda=0.05} = 1.363$. ↑/↓ means that the method is significantly better/worse than the proposed one. The lowest rank for each classification committee is highlighted in boldface.

Base classifier	class no.	FCM (1)	RRC (2)	Mean (3)	Dudani (4)	ADW (5)	Shepard (6)
Heterogeneous	1	5.240	3.600↑	3.120↑	2.340↑	3.920	2.780↑
Tree	2	**2.480**	3.220	3.280	4.220↓	4.120↓	3.680
KNN-5	3	4.040	4.500	**2.780**	3.420	3.400	2.860
KNN-10	3	3.900	4.520	3.340	**2.740**	3.420	3.080
KNN-15	5	3.220	4.480	**3.080**	3.120	3.740	3.360
MLP_5	6	3.820	4.360	3.060	3.480	3.640	**2.640**
MLP_10	7	4.040	4.020	3.380	3.120	3.340	**3.100**
MLP_5_5	8	3.820	4.400	3.360	**2.760**	3.500	3.160
MLP_10_10	9	3.660	5.020	3.400	**2.620**	2.960	3.340
SVM-linear	10	4.360	4.300	2.920↑	2.960↑	3.600	**2.860**↑
SVM-radial	11	4.880	4.620	3.280↑	**2.680**↑	2.820↑	2.720↑
SVM-sigmoid	12	**2.720**	3.320	3.620	4.140↓	3.820	3.380
SVM-square	13	4.600	3.800	3.120↑	**2.800**↑	3.360	3.320
SVM-cubic	14	4.660	4.080	2.980↑	3.120↑	3.320	**2.840**↑

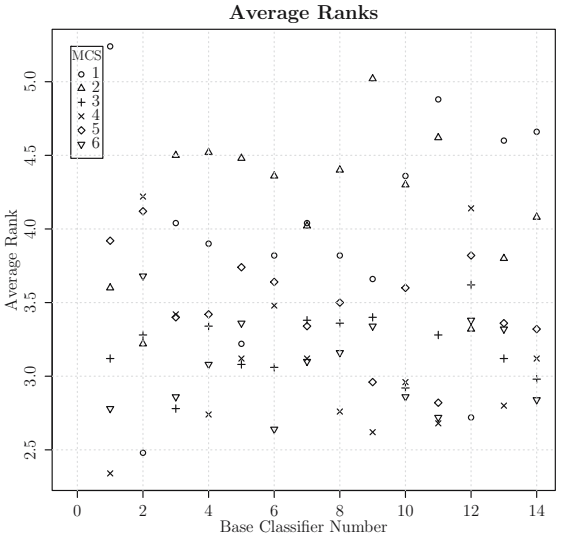


Fig. 4. Visualization of the average ranks: base classifiers and MCSs are numbered according to Table 3.

among others, the names of the compared algorithms. To distinguish static and dynamic versions of the investigated algorithms, we used a naming convention according to which updatable variants of the FCM and the RRC were denoted by the FCMU and the RRCU, respectively.

Table 7 is divided into two main sections. The first one presents the mean classification error and the standard deviation of the classification error which were obtained by the evaluated classifiers. The second one provides us with the p-values related to the Wilcoxon test which was performed to separately compare performance on each dataset. The results proved that, on most datasets, parameter tuning results in a significant improvement in terms of classification accuracy. Consequently, the overall efficiency of the dynamic systems also turned out to be significantly better. The outcome of the Wilcoxon testing procedure, which confirms the aforesaid improvement is provided in the last row of the table (denoted as *Wilcox p-Val*). The mean ranks also support this observation. The results demonstrate as well that the proposed updatable method is significantly better at $\lambda = 0.1$ in terms of the mean classification error, although there is no significant difference in the static mode. The outcome proves that when the number of training examples is relatively low, namely, 5% of the original number of instances, the proposed method achieves better results than the original updatable RRC combiner.

The results of comparison, in terms of the mean relative difference, are presented in Table 8, also divided into two sections. The first presents the mean relative difference for each benchmark set (a negative result means that on the given set the classification quality of the updatable combiner was lower in comparison with the static system). The performed statistical test revealed the differences between algorithms on a majority of sets are significantly different. Contrary to this result, the overall difference does not vary significantly between the evaluated methods of competence tuning (the p-value presented in the last row of the analysed table). However, the mean ranks may suggest that the FCMU combiner performs slightly better (in the case of the relative difference the higher rank stand, for a better improvement rate).

4. Conclusions

In this study a multiclassifier combination method was developed. The method is based on the random reference classifier and the local fuzzy confusion matrix. We

Table 4. Mean error rate ± standard deviation for the tree ensemble. ↑/↓ means that the method is significantly better/worse than the proposed one ($\lambda = 0.05$). For each set the lowest classification error is highlighted in boldface.

Set name	FCM	RRC	Mean	Dudani	ADW	Shepard
Breast Tissue	.294±.149	.310±.124	.298±.109	.308±.140	.272±.091	**.263±.135**
Ecoli	**.134±.044**	.143±.067	.165±.067	.156±.066	.149±.055	.150± .064
Faults	**.246±.024**	.260±.044	.287±.043	.291±.029	.277±.033	.276±.038
Glass	.273±.088	.295±.088	.257±.059	**.237±.073**	.269±.083	.246±.077
HillVall	**.424±.057**	.427±.049	.426±.060	.427±.034	.425±.038	.431±.054
Seeds	.081±.081	.105±.063	.090±.035	.086±.049	**.067±.051**	.086±.054
ULC	.214± .052	.167±.025	**.166±.043**↑	.173±.037	.173±.020	.167±.049
Vertebral Column	**.148±.051**	.181±.057	.155±.056	.158±.051	.177±.061	.181±.038
acute	.138±.046	.158±.045	.136±.049	.156±.059	.156±.073	**.134±.067**
bank_auth	**.018±.016**	.028±.014	.032±.019	.028±.012	.033±.020	.026±.016
fertility	.167±.106	.119± .034	.119±.034	.119±.034	.119±.034	.119±.034
ionosphere	.100±.039	**.083±.059**	.094±.045	.105±.043	.103±.041	.097±.043
iris	.053±.028	**.033±.035**	.060±.058	.067±.063	.060±.058	.060±.058
mfdig_fac	**.036±.011**	.101±.022↓	.138±.035↓	.140±.027↓	.138±.039↓	.139±.028↓
mfdig_fou	**.204±.033**	.245±.036↓	.265±.038	.266±.034↓	.271±.033↓	.271±.031↓
mfdig_kar	**.036±.013**	.181±.027↓	.213±.038 ↓	.225±.029↓	.220±.025↓	.229±.037↓
mfdig_mor	**.283±.026**	.298±.026	.291±.020	.297±.023	.302±.028	.295±.023
mfdig_pix	**.026±.009**	.107±.022↓	.131±.031↓	.137±.029↓	.141±.026↓	.136±.041↓
mfdig_zer	**.215±.013**	.299±.029↓	.311±.022↓	.319±.026↓	.317±.024↓	.321±.027↓
pima	.246±.042	.237±.038	.232±.029	**.225±.028**	.235±.034	.248±.028
semeion	**.081±.024**	.249±.034↓	.318±.035↓	.305±.044↓	.318±.039↓	.318±.032↓
spectF	.292±.077	.194±.070	**.187±.052**	.213±.048	.202±.043	**.187±.058**
wine	**.050**±.061	.084±.102	.107±.116	.074±.091	.095±.133	.073±.123
wq_red	.387±.029	.365±.045	.371±.031	.387±.033	.375±.032	.377±.028
wq_white	**.401±.023**	.418±.022	.427±.027	.425±.029	.427±.030	.424±.025↓
Avg. rank	2.480	3.220	3.280	4.220	4.120	3.680

harnessed the local confusion matrix to compute classifier competence and cross-competence. The experimental evaluation confirmed that the cross-competence measure allows us to utilize information from classifiers, which consistently misclassified some patterns. Moreover, the combiner is able to substitute the output of inaccurate classifiers by the output computed using local confusion matrix.

Our experiments showed that, even if the classification committee consists of random-guessing classifiers, the classification quality achieved by the entire system is still significantly better than random guessing. Those properties, combined with the ability to tune its parameters, suggests that the proposed model can be an effective tool in stream data classification. However, its performance in this field must be carefully assessed. The obtained results are promising, so we are willing to continue our work in order to improve the proposed algorithm.

Acknowledgment

This work was financed with the National Science Center resources for the years 2012–2014 within the research project no. DEC-2011/01/B/ST6/06168. The computational resources were provided by the PL-Grid Infrastructure.

We would like to thank the anonymous reviewers for their constructive comments and helpful suggestions.

References

Bache, K. and Lichman, M. (2013). UCI machine learning repository, `http://archive.ics.uci.edu/ml`.

Berger, J.O. and Berger, J.O. (1985). *Statistical Decision Theory and Bayesian Analysis*, Springer-Verlag, New York, NY.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*, Clarendon Press/Oxford University Press, Oxford/New York, NY.

Blum, A. (1998). On-line algorithms in machine learning, *in* A. Fiat and G.J. Woeginger (Eds.), *Developments from a June 1996 Seminar on Online Algorithms: The State of the Art*, Springer-Verlag, London, pp. 306–325.

Breiman, L. (1996). Bagging predictors, *Machine Learning* **24**(2): 123–140.

Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees*, Wadsworth and Brooks, Monterey, CA.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification, *IEEE Transactions on Information Theory* **13**(1): 21–27, DOI:10.1109/TIT.1967.1053964.

Table 5. Mean error rate ± standard deviation for the heterogeneous ensemble. ↑/↓ means that the method is significantly better/worse than the proposed one ($\lambda = 0.05$). For each set the lowest classification error is highlighted in boldface.

Set name	FCM	RRC1	Mean	Dudani	ADW	Shepard
Breast Tissue	.350±.197	.326±.151	.333±.128	**.313±.126**	.360±.117	.336±.091
Ecoli	.113±.046	.120± .049	**.111±.047**	.113±.046	.113±.046	.117± .048
Faults	.252±.031	.238±.018	.243±.029	**.236±.033**	.238±.028	.243±.030
Glass	.305±.124	**.303±.131**	.318±.118	.307±.142	.333±.142	.308±.133
HillVall	.446±.039	.367±.050↑	.337±.046↑	**.332±.052**↑	.342±.035↑	.335±.043↑
Seeds	.067±.033	.052±.035↑	**.048±.039**	**.048±.039**	**.048±.039**	.052±.042↑
ULC	.236±.064	.169±.045	.155±.047	**.154±.049**	.160±.050	.158± .044
Vertebral Column	.152±.046	**.145±.059**	.148±.053	**.145±.053**	**.145±.051**	.148±.051
acute	.137±.072	.105±.063	**.091±.048**	.101±.043	.101±.051	.097±.050
bank_auth	**.000±.000**	.001±.003	**.000±.000**	**.000±.000**	.013±.009	**.000±.000**
fertility	.151±.072	.130±.047	**.119±.034**	**.119±.034**	**.119±.058**	**.119±.034**
ionosphere	.100±.048	.092±.055	**.077±.050**	.081±.052	.092±.048	.089±.049
iris	.053±.061	.040±.064	**.047±.063**	**.047±.063**	**.047±.063**	**.047±.063**
mfdig_fac	.033±.016	**.019±.008**↑	.021±.013↑	.020±.010↑	.025±.013	.020±.010↑
mfdig_fou	.199±.021	.167±.015↑	.162±.014↑	**.159±.021**	.164±.013↑	.163±.018↑
mfdig_kar	.032±.010	.033±.012	.029±.008	.033±.010	.031±.007	**.029±.009**
mfdig_mor	.290±.030	.264±.024	.276±.028	.277±.030	.292±.022	**.269±.025**
mfdig_pix	.026±.009	.026±.008	.026±.011	**.021±.009**	.029±.010	.023±.009
mfdig_zer	.202±.018	.192±.011↑	.150±.016↑	**.146±.014**↑	.150±.018↑	**.146±.018**↑
pima	.246±.041	.237±.065	.246±.034	.246±.037	.246±.035	**.243±.040**
semeion	.076±.012	.068±.021	**.054±.018**↑	.056±.015	.058±.017	**.054±.021**↑
spectF	.286±.068	.205±.054	.199±.053	**.198±.047**	.217±.073	.210±.076
wine	.022±.039	.017±.027	.017±.027	.017±.027	**.011±.023**	**.011±.023**
wq_red	.372±.036	**.362±.052**	.374±.046	.365±.040	.371±.035	.368±.048
wq_white	.400±.017	.395±.017	.394±.011	.393±.011	.391±.020	**.390±.018**
Avg. rank	5.240	3.600	3.120	2.340	3.920	2.780

Table 6. Random guessing base classifiers: mean classification error ± standard deviation, $CD_{\lambda=0.05} = 0.633$. ↑/↓ means that the method is significantly better/worse than the proposed one ($\lambda = 0.05$). For each set the lowest classification error is highlighted in boldface.

Set name	FCM	RRC	Mean	Set name	FCM	RRC	Mean
Br. Tis.	**.374±.140**	.735±.142↓	.835±.113↓	mfdig_fac	**.034±.009**	.921±.040↓	.941±.050↓
Ecoli	**.125±.076**	.845±.126↓	.820±.183↓	mfdig_fou	**.208±.025**	.929±.051↓	.935±.044↓
Faults	**.245±.024**	.853±.123↓	.840±.176↓	mfdig_kar	**.043±.014**	.891±.065↓	.908±.070↓
Glass	**.319±.125**	.746±.146↓	.769±.151↓	mfdig_mor	**.298±.021**	.908±.086↓	.904±.036↓
HillVall	**.470±.029**	.516±.040	.494±.034	mfdig_pix	**.026±.011**	.941±.063↓	.918±.053↓
Seeds	**.067±.051**	.814±.146↓	.762±.125↓	mfdig_zer	**.217±.020**	.908±.094↓	.910±.040↓
ULC	**.241±.066**	.922±.059↓	.892±.059↓	pima	**.257±.040**	.475±.065↓	.494±.070↓
Vert. Col.	**.200±.104**	.594±.167↓	.658±.212↓	semeion	**.082±.019**	.915±.098↓	.890±.078↓
acute	**.183±.046**	.927±.063↓	.900±.082↓	spectF	**.291±.092**	.513±.074↓	.502±.094↓
bank_auth	**.020±.013**	.508±.046↓	.501±.038↓	wine	**.046±.072**	.772±.166↓	.757±.190↓
fertility	**.141±.073**	.543±.137↓	.489±.224↓	wq_red	**.391±.021**	.745±.131↓	.684±.133↓
ionosphere	**.134±.050**	.556±.061↓	.501±.060↓	wq_white	**.399±.024**	.689±.121↓	.658±.145 ↓
iris	**.053±.053**	.633±.254↓	.753±.122↓	**Wilcox p-Val**	–	2.384E-07	2.384E-07
				avg rank.	1.00	2.64	2.36

Dai, Q. (2013). A competitive ensemble pruning approach based on cross-validation technique, *Knowledge-Based Systems* **37**(9): 394–414, DOI: 10.1016/j.knosys.2012.08.024.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets, *The Journal of Machine Learning Research* **7**: 1–30.

Devroye, L., Györfi, L. and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*, Springer, New York, NY.

Didaci, L., Giacinto, G., Roli, F. and Marcialis, G.L. (2005). A study on the performances of dynamic classifier selection based on local accuracy estimation, *Pattern Recognition* **38**(11): 2188–2191.

Dietterich, T.G. (2000). Ensemble methods in machine learning, *Proceedings of the 1st International Workshop on Multiple Classifier Systems, MCS'00, Cagliari, Italy*, pp. 1–15.

Dunn, O.J. (1961). Multiple comparisons among means, *Journal*

Table 7. Mean error rate ± standard deviation for updatable ensembles.

	mean classification error				set comp p-value		
Set name	FCMU	FCM	RRCU	RRC	FCM–FCMU	RRC–RRCU	FCMU–RRCU
Ecoli	.153±.049	.203±026	.562±.019	.562±.019	.00148	1.00000	.00049
Faults	.295±.047	.392±.021	.387±.025	.419±.017	.00001	.00002	.00000
Seeds	.085±.064	.121±.056	.186±.051	.213±.043	.00107	.00038	.21012
ULC	.656±.033	.725±.011	.713±.017	.726±.010	.00002	.00474	.00000
Vert. Col.	.217±.084	.304±.051	.480±.024	.481±.017	.00126	1.00000	.00005
iris	.064±.078	.153±.055	.238±.061	.344±.039	.00064	.00001	.09551
mfdig_fac	.167±.016	.179±.015	.153±.015	.159±.015	.00474	.02215	.02293
mfdig_fou	.340±.019	.328±.020	.314±.022	.321±.021	.00688	.02667	.00064
mfdig_kar	.303±.017	.291±.022	.238±.026	.256±.022	.00413	.00043	.00017
mfdig_mor	.304±.031	.313±.022	.335±.020	.333±.020	.02665	1.00000	.00514
mfdig_pix	.203±.023	.206±.022	.160±.022	.168±.021	.72932	.02366	.24125
mfdig_zer	.237±.025	.321±.017	.347±.014	.364±.013	.00050	.00050	.00017
semeion	.521±.033	.437±.021	.337±.018	.379±.021	.00001	.00008	.00001
wine	.052±.042	.093±.036	.194±.030	.227±.030	.00690	.00384	.00001
wq_red	.409±.056	.471±.026	.424±.026	.440±.026	.00049	.00193	.21850
wq_white	.427±.035	.489±.014	.443±.021	.459±.021	.00001	.00003	.16372
avg. rank	1.812	2.812	2.219	3.156			
Wilcox p-Val	–	.00919	–	.00109			
Wilcox p-Val	–		.07391				
Wilcox p-Val		–		.27440			

of the American Statistical Association **56**(293): 52–64.

Fraz, M.M., Remagnino, P., Hoppe, A., Uyyanonvara, B., Rudnicka, A.R., Owen, C.G. and Barman, S. (2012). An ensemble classification-based approach applied to retinal blood vessel segmentation, *IEEE Transactions on Biomedical Engineering* **59**(9): 2538–2548.

Freund, Y. and Shapire, R. (1996). Experiments with a new boosting algorithm, *Machine Learning: Proceedings of the 13th International Conference, Bari, Italy*, pp. 148–156.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings, *The Annals of Mathematical Statistics* **11**(1): 86–92, DOI: 10.2307/2235971.

Gama, J. (2010). *Knowledge Discovery from Data Streams*, 1st Edn., Chapman & Hall/CRC, London.

Giacinto, G. and Roli, F. (2001). Dynamic classifier selection based on multiple classifier behaviour, *Pattern Recognition* **34**(9): 1879–1881.

Holm, S. (1979). A simple sequentially rejective multiple test procedure, *Scandinavian Journal of Statistics* **6**(2): 65–70.

Hsieh, N.-C. and Hung, L.-P. (2010). A data driven ensemble classifier for credit scoring analysis, *Expert systems with Applications* **37**(1): 534–545.

Huenupán, F., Yoma, N.B., Molina, C. and Garretón, C. (2008). Confidence based multiple classifier fusion in speaker verification, *Pattern Recognition Letters* **29**(7): 957–966.

Jurek, A., Bi, Y., Wu, S. and Nugent, C. (2013). A survey of commonly used ensemble-based classification techniques, *The Knowledge Engineering Review* **29**(5): 551–581, DOI: 10.1017/s0269888913000155.

Kittler, J. (1998). Combining classifiers: A theoretical framework, *Pattern Analysis and Applications* **1**(1): 18–27.

Ko, A.H., Sabourin, R. and Britto, Jr., A.S. (2008). From dynamic classifier selection to dynamic ensemble selection, *Pattern Recognition* **41**(5): 1718–1731.

Kuncheva, L.I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*, 1st Edn., Wiley-Interscience, New York, NY.

Kuncheva, L.I. and Rodríguez, J.J. (2014). A weighted voting framework for classifiers ensembles, *Knowledge-Based Systems* **38**(2): 259–275.

Kurzynski, M. (1987). Diagnosis of acute abdominal pain using three-stage classifier, *Computers in Biology and Medicine* **17**(1): 19–27.

Kurzynski, M., Krysmann, M., Trajdos, P. and Wolczowski, A. (2014). Two-stage multiclassifier system with correction of competence of base classifiers applied to the control of bioprosthetic hand, *IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2014, Limassol, Cyprus*.

Kurzynski, M. and Wolczowski, A. (2012). Control system of bioprosthetic hand based on advanced analysis of biosignals and feedback from the prosthesis sensors, *Proceedings of the 3rd International Conference on Information Technologies in Biomedicine, ITIB 12, Kamień Śląski, Poland*, pp. 199–208.

Mamoni, D. (2013). On cardinality of fuzzy sets, *International Journal of Intelligent Systems and Applications* **5**(6): 47–52.

Plumpton, C.O. (2014). Semi-supervised ensemble update strategies for on-line classification of FMRI data, *Pattern Recognition Letters* **37**: 172–177.

Table 8. Mean relative differences ± standard deviation.

	relative diff.		set comp p-val
Set name	FCM–FCMU	RRC–RRCU	FCM–RRC
Ecoli	.04988±.03953	.00000±.00000	.00049
Faults	.09642±.03969	.03217±.01386	.00000
Seeds	-.01729±.02503	.02656±.02091	.21012
ULC	.06925±.02637	.01339±.01492	.00000
Vertebral Column	.08727±.06674	.00058±.01585	.00005
iris	-.01645±.04768	.10547±.05139	.09551
mfdig_fac	.01237±.01115	.00584±.00796	.02293
mfdig_fou	-.01161±.01276	.00739±.01034	.00064
mfdig_kar	-.01182±.01273	.01837±.01072	.00017
mfdig_mor	.00921±.01222	-.00121±.00657	.00514
mfdig_pix	.00350±.01160	.00803±.01070	.24125
mfdig_zer	.08421±.02888	.01676±.00978	.00017
semeion	-.08356±.02516	.04148±.02367	.00000
wine	-.06377±.04559	.03304±.03429	.85957
wq_red	.06202±.04952	.01635±.01648	.00027
wq_white	.06212±.03214	.01595±.00869	.00001
avg. rank	1.56250	1.43750	
Wilcox p-Val	–	.74356	

Plumpton, C.O., Kuncheva, L.I., Oosterhof, N.N. and Johnston, S.J. (2012). Naive random subspace ensemble with linear classifiers for real-time classification of FMRI data, *Pattern Recognition* **45**(6): 2101–2108.

R Core Team (2012). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, `http://www.R-project.org/`.

Rokach, L. (2010). Ensemble-based classifiers, *Artificial Intelligence Review* **33**(1–2): 1–39.

Rokach, L. and Maimon, O. (2005). Clustering methods, *Data Mining and Knowledge Discovery Handbook*, Springer Science + Business Media, New York, NY, pp. 321–352.

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics* **20**(1): 53–65.

Scholkopf, B. and Smola, A.J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA.

Tahir, M.A., Kittler, J. and Bouridane, A. (2012). Multilabel classification using heterogeneous ensemble of multi-label classifiers, *Pattern Recognition Letters* **33**(5): 513–523.

Tsoumakas, G., Katakis, I. and Vlahavas, I. (2010). Random k-labelsets for multi-label classification, *IEEE Transactions on Knowledge and Data Engineering* **99**(1): 1079–1089.

Valdovinos, R. and Sánchez, J. (2009). Combining multiple classifiers with dynamic weighted voting, *in* E. Corchado *et al.* (Eds.), *Hybrid Artificial Intelligence Systems*, Lecture Notes in Computer Science, Vol. 5572, Springer, Berlin/Heidelberg, pp. 510–516.

Ward, J. (1963). Hierarchical grouping to optimize an objective function, *Journal of the American Statistical Association* **58**(301): 236–244.

Wilcoxon, F. (1945). Individual comparisons by ranking methods, *Biometrics Bulletin* **1**(6): 80–83.

Woloszynski, T. (2013). Classifier competence based on probabilistic modeling (ccprmod.m) at Matlab central file exchange, `http://www.mathworks.com/matlabcentral/fileexchange/28391-a-probabilistic-model-of-classifier-competence`.

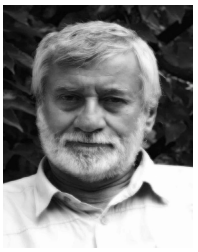
Woloszynski, T. and Kurzynski, M. (2011). A probabilistic model of classifier competence for dynamic ensemble selection, *Pattern Recognition* **44**(10–11): 2656–2668.

Woloszynski, T., Kurzynski, M., Podsiadlo, P. and Stachowiak, G.W. (2012). A measure of competence based on random classification for dynamic ensemble selection, *Information Fusion* **13**(3): 207–213.

Wolpert, D.H. (1992). Stacked generalization, *Neural Networks* **5**(2): 214–259.

Wozniak, M., Graña, M. and Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems, *Information Fusion* **16**(1): 3–17.

Pawel Trajdos received the M.Sc. degree in computer science from the Wrocław University of Technology in 2013. Since then he has been pursuing his Ph.D. at the Department of Systems and Computer Networks of the same university. His research interests cover the area of computational inteligence and machine learning with application to biology and medicine.



Marek Kurzynski received the Ph.D. in automatic control from the Wrocław University of Technology (Poland) in 1974 and the D.Sc. degree in computer science from the Silesian University (Poland) in 1987. He is currently a full professor in the Department of Systems and Computer Networks, Wrocław University of Technology. Professor Kurzynski has been a recipient, investigator and co-investigator of numerous research grants from the EU, the Polish Ministry of Science and other industrially founded research projects. He has published more than 200 refereed journal and conference papers as well as four books. His research interests include pattern recognition, artificial intelligence methods, biological signal processing and medical informatics.

Received: 10 November 2014
Revised: 4 May 2015