On a New Competence Measure Applied to the Dynamic Selection of Classifiers Ensemble

Marek Kurzynski(✉) and Pawel Trajdos

Department of Systems and Computer Networks,
Wroclaw University of Science and Technology,
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
marek.kurzynski@pwr.edu.pl

Abstract. In this paper a new method for calculating the classifier competence in the dynamic mode is developed. In the method, first decision profile of the classified object is calculated using K nearest objects from the validation set. Next, the decision profile is compared with the support vector produced by the classifier. The competence measure reflects the outcome of this comparison and rates the classifier with respect to the similarity of its support vector and decision profile of the test object in a continuous manner. Three different procedures for calculating decision profile and three different measures for comparing decision profile and support vector are proposed, which leads to nine methods of competence calculation. Two multiclassifier systems (MC) with homogeneous and heterogeneous pool of base classifiers and with dynamic ensemble selection scheme (DES) were constructed using the methods developed. The performance of constructed MC systems was compared against seven state-of-the-art MC systems using 15 benchmark data sets taken from the UCI Machine Learning Repository. The experimental investigations clearly show the effectiveness of the combined multiclassifier system in dynamic fashion with the use of the proposed measures of competence regardless of the ensemble type used.

Keywords: Multiclassifier system · Dynamic ensemble selection · Measure of competence

1 Introduction

In the last two decades, multiclassifier (MC) systems which combine responses of set of classifiers have been intensively developed. The reason is that different classifiers offer complementary information about the object to be classified and therefore MC system can achieve better classification accuracy than any single classifier in the ensemble.

MC system has three general phases [2]: (1) generation in which the training set is used to generate a pool of classifiers; (2) selection in which a single classifier (or an ensemble of classifiers) is selected to perform the classification; (3)

A. Yamamoto et al. (Eds.): DS 2017, LNAI 10558, pp. 93–107, 2017.
DOI: 10.1007/978-3-319-67786-6_7

combination (or integration) in which the final decision is made based on the predictions of the classifiers. It must be noted that selection and integration phases may be facultative, since for the classifier combination two main approaches used are classifier fusion and classifier selection [13]. In the first method, all classifiers in the ensemble contribute to the decision of the MC system, e.g. through sum or majority voting [11]. In the second approach, a single classifier is selected from the ensemble and its decision is treated as the decision of the MC system. The selection of classifiers can be either static or dynamic. In the static selection scheme, classifier is selected for all test objects, whereas dynamic classifier selection (DCS) approach explores the use of different classifiers for different test objects [6].

Recently, dynamic ensemble selection (DES) methods have been developed which first dynamically select an ensemble of classifiers from the entire set (pool) and then combine the selected classifiers by majority voting [3,4,12,18]. In this way a DES based system takes advantage of both selection and fusion approaches. In most methods, the base classifiers are selected from the pool on the basis of their individual accuracy measure called competence in a local region of the feature space. These methods differ in algorithms for determining classifier competence and ways of defining the local regions.

In [23] two methods were proposed where the local accuracy (competence) of classifier is calculated as a simple percentage of correct classified samples from the validation set. In the first method called OLA (overall local accuracy), local accuracy is calculated in the region containing K-nearest validation objects of a test object. Whereas in the LCA (local class accuracy) method, classifier competence is determined considering only these validation objects from the K-nearest neighbors set which belong to the same class into which an unknown object is assigned. In [20–22] two methods using probabilistic model were developed. The idea of the first method is based on relating the response of the classifier with the response obtained by random guessing. The measure of competence reflects this relation and rates the classifier with respect to random guessing in a continuous manner. In this way, it is possible to evaluate a group of classifiers against a common reference point. Competent (incompetent) classifiers gain with such approach meaningful interpretation, i.e. they are more (less) accurate than the random classifier. In the second method, first a randomized reference classifier (RRC) is constructed which, on average, acts like the classifier evaluated. Next the competence of the classifier evaluated is calculated as the probability of correct classification of the respective RRC. Two interesting methods called *A priori* and *A posteriori* selection scheme was presented in [9]. In the *A priori* method, a classifier is selected based on its accuracy within the local region, without considering the class assigned to the unknown pattern. Similarly, in the *A posteriori* method, local accuracies are estimated using the class posterior probabilities and the distances of the samples in the defined local region. In [17] an interesting ranking-based approach to determine competence measure was proposed. In the method the ranking of base classifiers is done by estimating parameters related to the correctness of the classifiers in the pool. An interesting method called

MCB (Multiple Classifier Behavior) was proposed in [10]. In this method the competence is defined as the classification accuracy calculated for a subset of a validation set which is generated as follows. First, the MCB is calculated for a test object and its K-nearest validation objects as a vector whose elements are class labels assigned by all classifiers in the ensemble. Next, similarity between the MCB's are calculated using the averaged Hamming distance. Finally, the objects in the validation set that are the most similar to the test object are used to generate the subset. The original KNORA-Eliminate (KE) method belonging to the category of oracle-based methods was proposed in [12]. The oracles are represented by the K-nearest neighbors of the unknown pattern in the validation set and the KE method selects only those classifiers which are able to recognize the entire K-neighborhood of the test pattern.

In this paper a new method for calculating the classifier competence in the feature space is presented. In the proposed method, first the so-called decision profile of the classified object is determined using K-nearest validation objects. The decision profile provides the chance that the recognized object belongs to the specified class. In the probabilistic model the natural concept of decision profile is based on *a posteriori* probabilities of classes at the point x. Next, the decision profile is compared with the response produced by the classifier (support vector or values of discriminant functions) [7] and the competence is calculated according to the similarity rule: the closer the response to the profile is, the more competent the classifier is [14,15]. Three different procedures for calculating a decision profile and three different measures for comparing the decision profile and the support vector are proposed in this study.

In a nutshell, originality of the proposed approach consists in a different use of the validation set. In the state-of-the-art-methods described above, the validation set is directly used for calculating local accuracy of a classifier (i.e. its local competence) via ranking-based, accuracy-based, probabilistic-based, behavior-based and oracle-based measures. However, in the proposed method, validation set is used for evaluating the classification profile of the test point and competence of the classifier is determined by similarity of its response to this evaluation.

The paper is divided into four sections and organized as follows. In Sect. 2 the measures of classifier competence are presented and two multiclassifier systems using proposed measures of competence in a dynamic fashion are developed. The performance of proposed MCS's were compared with seven multiple classifier systems using 15 datasets taken from the UCI Machine Learning Repository. The results of computer experiments are described in Sects. 3, and 4 concludes the paper.

2 Multiclassifier System

2.1 Preliminaries

In the multiclassifier (MC) system we assume that a set of trained classifiers $\Psi = \{\psi_1, \psi_2, \ldots, \psi_L\}$ called base classifiers is given.

A classifier ψ_l is a function $\psi_l : \mathcal{X} \to \mathcal{M}$ from a metric feature space $\mathcal{X} \subseteq \mathbb{R}^{dim}$ to a set of class labels $\mathcal{M} = \{1, 2, \ldots, M\}$. Classification is made according to the maximum rule

$$\psi_l(x) = i \Leftrightarrow d_{li}(x) = \max_{j \in \mathcal{M}} d_{lj}(x), \tag{1}$$

where $[d_{l1}(x), d_{l2}(x), \ldots, d_{lM}(x)]$ is a vector of class supports (classifying function) produced by ψ_l. Without loss of generality we assume that $d_{lj}(x) \geq 0$ and $\sum_j d_{lj}(x) = 1$.

In this paper, we propose MC systems which use a dynamic ensemble selection scheme and trainable combining methods based on a competence measure $c(\psi_l|x)$ of each base classifier $(l = 1, 2, ..., L)$ evaluating the competence of classifier ψ_l at a point $x \in \mathcal{X}$. Competence measure is normalized, i.e. $0 \leq c(\psi_l|x) \leq 1$. $c(\psi_l|x) = 0(1)$ denotes the most incompetent (competent) classifier ψ_l.

For the training methods of combining the base classifiers, it is assumed that a validation set

$$\mathcal{V} = \{(x_1, j_1), (x_2, j_2), \ldots, (x_N, j_N)\}; \;\; x_k \in \mathcal{X}, \; j_k \in \mathcal{M} \tag{2}$$

containing pairs of feature vectors and their corresponding class labels is available.

2.2 Measure of Competence

***K*-neighborhood.** Let first introduce the concept of K-neighborhood of object $x \in \mathcal{X}$ which is defined as the set of K nearest neighbors of the point x from validation set $\mathcal{V}$, viz.

$$\mathcal{S}_K(x) = \{x_{n_1}, x_{n_2}, \ldots x_{n_K} \in \mathcal{V} : \\ \max_{k=1,2,\ldots,K} ||x_{n_k} - x|| \leq \min_{x_l \notin \mathcal{S}_K(x)} ||x_l - x||\}, \tag{3}$$

where $|| \cdot ||$ denotes the distance in the feature space $\mathcal{X}$. The neighborhood size K is a parameter of the method – its value can be selected experimentally.

Decision Profile. Decision profile of object $x \in \mathcal{X}$

$$\delta(x) = [\delta_1(x), \delta_2(x), \ldots, \delta_M(x)], \; \delta_j(x) \geq 0, \; \sum_j \delta_j(x) = 1 \tag{4}$$

denotes the vector of normalized values where the jth value $\delta_j(x)$ is interpreted as a measure of chance that object x belongs to the jth class $(j \in \mathcal{M})$. In the probabilistic model the natural value of $\delta_j(x)$ is *a posteriori* probability of jth class at the point x.

We propose the following methods for calculating decision profile at the point x using its K-neighborhood.

The Fraction-based Method (FM)

In this approach, $\delta_j(x)$ is calculated as the fraction of objects from the jth class in the set $\mathcal{S}_K(x)$. Let $M_j^{(K)}(x)$ be the number of validation objects from $\mathcal{S}_K(x)$ belonging to the j-th class. Then

$$\delta_j(x) = \frac{M_j^{(K)}(x)}{K}, \quad j \in \mathcal{M}. \tag{5}$$

The Ranking-based Method (RM)

In the RM method $\delta_j(x)$ is equal to the normalized sum of ranks of objects belonging to the jth class in the set $\mathcal{S}_K(x)$. Let $r(x_k)$ be the rank of validation object $x_k \in \mathcal{S}_K(x)$. The nearest neighbour has the rank equal to K, the rank of the furthest neighbor is equal to 1. Then

$$r_j(x) = \sum_{x_k \in \mathcal{S}_K(x) : j_k = j} r(x_k) \tag{6}$$

is the sum of ranks of validation objects from the K neighborhood of x belonging to the jth class. And next

$$\delta_j(x) = \frac{r_j(x)}{\sum_{k=1}^{K} k}. \tag{7}$$

The Potential Function Method (PM)

Let $H(x, x_k)$ be a non-negative potential function [16] decreasing with the increasing distance between x and x_k. In this study, a Gaussian potential function with the Euclidean distance is used:

$$H(x, x_k) = \exp(-||x - x_k||^2). \tag{8}$$

Then, we can calculate $\delta_j(x)$ as a normalized sum of potential functions (8) for objects belonging to the jth class from the set $\mathcal{S}_K(x)$, namely:

$$\delta_j(x) = \frac{\sum_{x_k \in \mathcal{S}_K(x) : j_k = j} \exp(-||x - x_k||^2)}{\sum_{j \in \mathcal{M}} \sum_{x_k \in \mathcal{S}_K(x) : j_k = j} \exp(-||x - x_k||^2)}. \tag{9}$$

Distance Between Decision Profile and Vector of Supports. In order to evaluate ψ_l at x and determine its competence $c(\psi_l|x)$, we must compare decision profile $\delta(x)$ and vector of supports $d_l(x)$ and calculate distance $dist[\delta(x), d_l(x)]$. Competence measure is a normalized function of this distance decreasing with the increasing distance between $\delta(x)$ and $d_l(x)$. In particular, $c(\psi_l|x)$ is equal to 1 (0) if distance is equal to 0 (is the greatest one).

We propose three different methods for calculating distance $dist[\delta(x), d_l(x)]$ and the resulting measures of competence.

Euclidean Distance (ED)

We adopt the Euclidean distance

$$dist[(\delta(x), d_l(x)] = ||\delta(x) - d_l(x)|| \tag{10}$$

and hence we get

$$c(\psi_l|x) = \frac{\sqrt{2} - ||\delta(x) - d_l(x)||}{\sqrt{2}}. \tag{11}$$

Max-Max Distance (MD)

Let j be the class number for which classifier ψ_l produced the greatest support value at the point x (i.e. $d_{lj}(x) = \max_{k \in \mathcal{M}}(d_{lk}(x))$). Similarly, let i be the class number with the greatest value in the decision profile $\delta(x)$ at x. Then, the max–max distance is defined as:

$$dist[(\delta(x), d_l(x)] = |d_{lj}(x) - \delta_j(x)| + |d_{li}(x) - \delta_i(x)|. \tag{12}$$

Hence we have the following formula for competence measure:

$$c(\psi_l|x) = \frac{2 - |d_{lj}(x) - \delta_j(x)| + |d_{li}(x) - \delta_i(x)|}{2}. \tag{13}$$

Hamming Distance (HD)

Let $h(\psi_l(x)) = [j_1^{'}, j_2^{'}, \ldots, j_M^{'}]$ and $h(x) = [j_1^{''}, j_2^{''}, \ldots, j_M^{''}]$ be the vectors of class numbers ordered according to the decreasing values of supports produced by ψ_l at x and decision profile of x, respectively. Distance between $d_l(x)$ and $\delta_j(x)$ is defined as the Hamming distance between vectors $h(\psi_l(x))$ and $h(x)$, namely

$$dist[(\delta(x), d_l(x)] = D_H[h(\psi_l(x)), h(x)]. \tag{14}$$

Hence we get the following form of competence measure

$$c(\psi_l|x) = \frac{M - D_H[h(\psi_l(x)), h(x)]}{M}. \tag{15}$$

Example. Consider a classification problem with three classes ($M = 3$). Figure 1 presents 6-neighborhood of an object x in the two-dimensional feature space. Additional unit grid will help to determine distances between objects. Suppose that classifier ψ produced supports $d_1(x) = 0.3$, $d_2(x) = 0.6$ and $d_3(x) = 0.1$. Our purpose is to determine the competence $c(\psi|x)$ of the classifier ψ at the point x using presented methods.

From Fig. 1 we simply get the Euclidean distances between x and validation objects:

$$||x - x_1|| = 2,\ ||x - x_2|| = 5,\ ||x - x_3|| = 2.83,$$

$$||x - x_4|| = 2.23,\ ||x - x_5|| = 3,\ ||x - x_6|| = 3, 6.$$

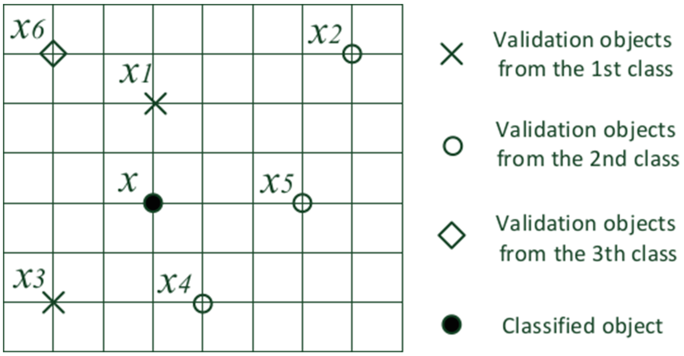
x6
x2
x1
x
x5
x3
x4
Validation objects from the 1st class
Validation objects from the 2nd class
Validation objects from the 3th class
Classified object

Fig. 1. Illustration of Example: 6-neighborhood of an object x.

First, we calculate decision profiles for the proposed methods.
FM method:

$$M_1^{(5)} = 2, M_2^{(5)} = 3, M_3^{(5)} = 1 \text{ and hence } \delta_1(x) = 1/3, \delta_2(x) = 1/2, \delta_3(x) = 1/6. \tag{16}$$

RM method:

$$r_1(x) = 10,\ r_2(x) = 9,\ r_3(x) = 2$$

and hence

$$\delta_1(x) = 10/21,\ \delta_2(x) = 9/21,\ \delta_3(x) = 2/21. \tag{17}$$

PM method:

$$H(x, x_1) + H(x, x_3) = 0.135 + 0.059 = 0.194,\ H(x, x_6) = 0.027$$

$$H(x, x_2) + H(x, x_3) + H(x, x_5) = 0.006 + 0.108 + 0.049 = 0.163$$

and hence

$$\delta_1(x) = \frac{0.194}{0.384} = 0.505,\ \delta_2(x) = \frac{0.163}{0.384} = 0.424,\ \delta_3(x) = \frac{0.027}{0.384} = 0.071, \tag{18}$$

Now, using formulas (10)–(15) and calculated decision profiles (16), (17) and (18), we can calculate competence $c(\psi|x)$ of classifier ψ at the point x. Results for all combination of calculating decision profile methods and concept of distance between decision profile and vector of supports are presented in Table 1.

2.3 DES Systems

The proposed measure of competence can be applied in any multiclassifier system in selection/fusion algorithm provided that the feature space $\mathcal{X}$ is a metric space. In this subsection we describe two multiclassifier systems based on the DES strategy.

Table 1. Results of example.

		Distance $dist[(\delta(x), d_l(x)]$	Competence $c(\psi\|x)$
FM	ED	0.197	0.86
RM		0.245	0.826
PM		0.272	0.808
FM	MD	0.2	0.9
RM		0.347	0.827
PM		0.381	0.809
FM	HD	0	1
RM		2	0.333
PM		2	0.333

Multiclassifier System with Fusion at the Decision Level (MC1). In this system, first a subset $\Psi^*(x)$ of base classifiers with the competences greater than the random guess is selected for a given x:

$$\Psi^*(x) = \{\psi_{l1}, \psi_{l2}, \dots, \psi_{lT}\}, \text{ where } c(\psi_{lt}|x) > 1/M. \tag{19}$$

The selected classifiers are combined using the weighted majority voting rule where the weights are equal to the competences. This fusion method leads to the following class supports ($j = 1, 2, \dots M$):

$$d_j^{(MC1)}(x) = \sum_{t=1}^{T} c(\psi_{lt}|x) \lfloor \psi_{lt}(x) = j \rfloor, \tag{20}$$

where $\lfloor \cdot \rfloor$ denotes the Iverson bracket.

The MC1 system $\psi^{(MC1)}$ classifies x using the maximum rule:

$$\psi^{(MC1)}(x) = i \Leftrightarrow d_i^{(MC1)}(x) = \max_{j \in \mathcal{M}} d_j^{(MC1)}(x). \tag{21}$$

Multiclassifier System with Fusion at the Continuous-Value Level (MC2). The MC2 system is identical to the MC1 system except that selected classifiers (19) are combined at the continuous-value level ($j = 1, 2, \dots M$):

$$d_j^{(MC2)}(x) = \sum_{t=1}^{T} c(\psi_{lt}|x) d_{lt,j}(x). \tag{22}$$

Final decision – as previously – is made according to the maximum rule:

$$\psi^{(MC2)}(x) = i \Leftrightarrow d_i^{(MC2)}(x) = \max_{j \in \mathcal{M}} d_j^{(MC2)}(x). \tag{23}$$

The MC2 system with competence measures developed will be applied in the experimental investigations.

3 Experiments

3.1 Experimental Setup

The performance of the developed MC systems was evaluated in experiments using 15 benchmark data sets. In the first experiment, the MC2 system was evaluated using different methods for calculating decision profile (FM, RM and PM) and different distances between decision profile and support vector (ED, MD and HD). The methods that showed the best performance were identified. In the second experiment, the methods identified were compared with other competence–based MC systems. The experiments were conducted in MATLAB using PRTools 4.1 [8]. In both experiments, the value of $K = 5 \times M$ (M denotes the number of classes) was used as the neighborhood size.

The 15 benchmark data sets were taken from the UCI Machine Learning Repository [1]. We selected the same data sets which were used in experimental investigations presented in [21]. A brief description of the data sets used is given in Table 2.

Table 2. The data sets used in the experiments.

Data set	#Objects	#Features	#Classes
Blood transfusion	748	4	2
Breast cancer Wisconsin	699	9	2
Clouds	5000	2	2
Dermatology	366	34	6
EColi	336	7	8
Glass	214	9	6
Ionosphere	351	34	2
OptDigits	3823	64	10
Page blocks	5473	10	5
Pima Indians	768	8	2
Segmentation	2310	19	7
Spam	4601	57	2
Vowel	990	10	11
Wine	178	13	3
Yeast	1484	8	10

For each data set, feature vectors were normalized to zero mean and unit standard deviation. Two-fold cross-validation was used to extract training and test sets from each data set. For the calculation of the competences, a two-fold stacked generalization method was used [24]. In the method, the training set is split into two sets A and B of roughly equal sizes. Set A is first used for the training of the classifiers in the ensemble while set B is used for the calculation of the competences. Then, set B is used for the training while the competences are calculated using set A. Finally, the competences calculated for both sets are stacked together and the classifiers in the ensemble are trained using the union of sets A and B (i.e. the original training set). In this way, the competences of the classifiers are calculated for all the feature vectors in the original training set, but the data used for the calculation is unseen during the classifier training.

The experiments were conducted using two ensemble types: homogeneous and heterogeneous. The homogeneous ensemble consisted of 50 feed-forward backpropagation neural network classifiers with one hidden layer and the maximum number of learning epochs set to 80. Each neural network classifier was trained using randomly selected 70% of the objects from the training data set. The heterogeneous ensemble consisted of the following 11 base classifiers [7]:

- (1) linear classifier based on normal distribution with the same covariance matrix for each class;
- (2) quadratic classifier based on normal distribution with different covariance matrix for each class;
- (3) nearest mean classifier;
- (4–6) k-nearest neighbors classifiers with k = 1, 5, 10;
- (7, 8) Parzen density based classifier with the Gaussian kernel and the optimal smoothing parameter h_{opt} (and the smoothing parameter $h_{opt}/2$);
- (9) pruned decision tree classifier with the Gini splitting criterion;
- (10–11) feed-forward backpropagation neural network classifier containing one hidden layer with 10 neurons (two hidden layers with 5 neurons each) and the maximum number of learning epochs set to 80;

The performance of the systems constructed was compared with the following seven MC systems:

1. **Overall local accuracy method (OLA1)** [23]**.** In this method the competence at a test point x is calculated as the percentage of the correct recognition of the K-nearest validation samples of x;
2. **Local class accurracy method (LCA)** [23]**.** In this method the competence is estimated for each base classifier as the percentage of correct classifications within the local region (the K neighborhood), but considering only examples from the class as classifier gives for the unknown pattern;
3. **Overall local accuracy method (OLA2)** [19]**.** In this method the competence is calculated as in OLA1 approach but validation objects from the K-neighborhood are additionally weighted by their Euclidean distances to the unknown object x;
4. **Multiple classifier behavior method (MCB)** [10]**.** In this method the competence is calculated using a similarity function to measure the degree of similarity of the output profiles of all base classifiers;
5. **Oracle KNORRA-eliminate method (ORE)** [12]**.** In this method all classifiers are selected that correctly classify all samples in the local region (the K neighborhood). If no classifiers are selected, the local region is reduced until at least one classifier is selected;

6. **Randomized reference classifier method (RRC)** [21]**.** In this method the competence of base classifier is calculated as the probability of correct classification of randomized reference classifier (RRC) which - on average - acts as a modeled base classifier;
7. **Random guessing based method (RGM)** [22]**.** In this method the competence is calculated in relation to the random guessing method – the classifier is considered as competent (incompetent) if it is more (less) accurate than the random classifier.

3.2 Results and Discussion

Classification accuracies were averaged over 5 repetitions of two-fold cross-validation. Statistical differences in rank between the systems were obtained using the Friedman test with Iman and Davenport correction combined with the post hoc Holm's stepdown procedure [5]. The average ranks of the systems and a critical rank difference calculated using the Bonferroni-Dunn test are visualised. The level of $p < 0.05$ is considered as statistically significant.

The average ranks obtained from the first experiment for the nine methods proposed and for the homogeneous and heterogeneous ensembles are presented in Figs. 2A and B, respectively. The use of the potential function method for calculating decision profile and max-max distance between decision profile and support vector (PM-MD) resulted in the best average rank regardless of the ensemble type used. The average rank of PM-MD method is significantly better than average ranks for FM-MD, RM-MD, PM-HD, FM-HD and RM-HD methods. Methods with the Hamming distance achieved the worst average ranks regardless of the method for calculating decision profile and the ensemble type used. Thus, for the second experiment the PM-MD method was selected.

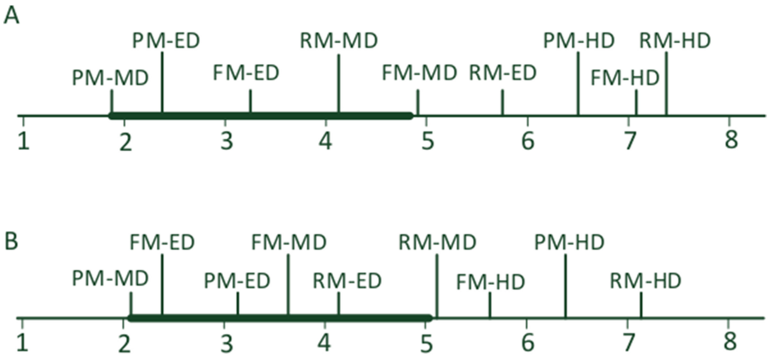

Fig. 2. Average ranks of the MC2 systems for different methods of calculating decision profile and different distances between decision profile and support vector for homogeneous (A) and heterogeneous (B) ensemble of base classifiers. The interval (thick line) is the critical rank difference (2.991) calculated using the Bonferroni-Dunn test ($p < 0.05$).

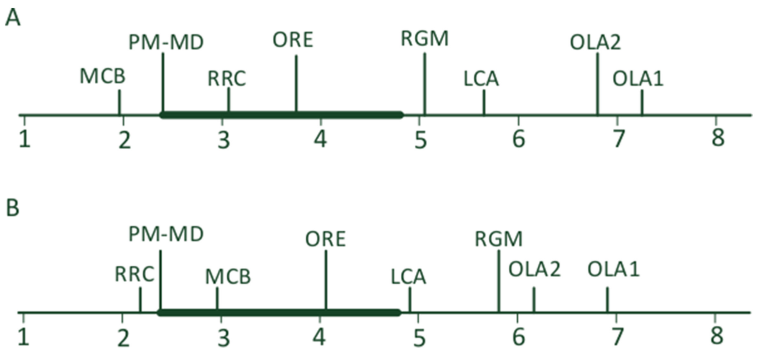

Fig. 3. Average ranks of MC systems compared for homogeneous (A) and heterogeneous (B) ensemble of base classifiers. The interval (thick line) is the critical rank difference (2.394) calculated using the Bonferroni-Dunn test ($p < 0.05$).

Table 3. Classification accuracies (in percent) and average ranks of the PM-MD system and the eight MCSs for the homogeneous ensemble. The best result for each data set is in bold.

Dataset	OLA1	LCA	OLA2	MCB	ORE	RRC	RGM	PM-MD
Blood	76.18	75.29	76.44	**77.15**	76.28	76.44	76.32	77.02
Breast	94.21	93.88	94.92	97.25	95.29	**97.88**	95.94	97.12
Clouds	62.92	63.15	64.22	**65.80**	64.00	63.64	63.82	65.27
Dermat	68.88	70.44	69.35	**75.28**	72.61	73.91	71.45	74.82
EColi	67.62	69.48	70.35	76.44	71.43	**78.01**	76.22	77.25
Glass	50.72	54.22	52.83	**69.95**	57.86	67.22	60.28	67.38
Iono	83.95	84.25	84.15	**88.57**	85.47	87.62	86.15	86.47
OptDig	81.28	86.55	87.12	88.32	86.48	88.21	87.35	**89.42**
Page	94.92	95.12	95.23	96.21	95.80	**96.35**	95.05	95.92
Pima	65.23	65.55	64.92	67.48	65.49	66.30	65.78	**68.52**
Segment	87.88	86.54	88.75	**96.23**	91.32	95.72	91.44	94.96
Spam	81.84	83.57	84.21	**89.12**	85.29	88.85	87.17	88.24
Vowel	49.92	53.22	50.5	**60.25**	55.86	59.45	57.73	**61.15**
Wine	91.28	93.15	91.22	94.47	89.52	95.84	92.78	**96.03**
Yeast	49.85	53.22	50.81	56.28	52.28	56.36	55.42	**57.27**
Av.Rank	7.26	5.68	6.83	1.98	3.79	3.06	5.05	2.35

The results obtained in the second experiment for the PM-MD method and seven MC systems and for the homogeneous and heterogeneous ensembles are presented in Tables 3 and 4 and in Figs. 3A and B, respectively. The system constructed achieved the second best average ranks for both types of classifier ensemble. The average rank of PM-MD method is significantly better than average ranks for LCA, RGM, OLA1 and OLA2 methods regardless of the ensemble type used.

Table 4. Classification accuracies (in percent) and average ranks of the PM-MD system and the eight MCSs for the heterogeneous ensemble. The best result for each data set is in bold.

Data base	OLA1	LCA	OLA2	MCB	ORE	RRC	RGM	PM-MD
Blood	75.70	77.23	75.93	77.43	76.48	**78.26**	77.58	78.12
Breast	95.12	95.84	95.48	95.17	**96.42**	96.28	94.87	95.54
Clouds	74.50	75.13	74.21	79.27	77.12	79.07	75.12	**80.02**
Dermat	93.18	93.28	93.42	**96.31**	94.82	96.27	93.88	95.92
EColi	82.13	84.55	82.86	84.18	86.12	**86.24**	83.15	84.88
Glass	64.18	65.28	64.05	67.40	67.15	67.35	64.88	**67.48**
Iono	82.98	83.17	82.75	86.12	85.94	**86.95**	83.15	86.92
OptDig	87.92	91.15	88.24	95.31	95.15	97.43	90.65	**97.48**
Page	89.24	92.38	90.82	95.84	95.21	**96.24**	91.13	96.18
Pima	67.21	68.89	67.15	69.12	68.73	**69.45**	67.45	69.32
Segment	84.02	87.55	86.58	96.41	95.11	95.32	89.55	**97.12**
Spam	88.21	89.45	88.85	**92.17**	90.32	91.91	90.05	91.72
Vowel	82.24	85.92	84.72	88.32	83.51	**90.18**	85.77	89.71
Wine	95.42	96.41	96.15	97.05	96.84	97.17	96.32	**98.03**
Yeast	55.66	57.05	55.51	56.94	56.83	**57.79**	57.12	57.11
Av.Rank	6.96	4.98	6.24	2.98	4.16	2.30	5.87	2.48

4 Conclusion

Nowadays, many researches have been focused on MC systems and consequently, many new solutions have been dedicated to each of the two main approaches: classifiers fusion and classifiers selection. In the proposed solutions the fundamental role plays the assessment of competence of base classifiers which is crucial in the DES scheme and in the combining of base classifiers. In the paper a new method for calculating the competence of a classifier in the feature space was presented. In the proposed method, first the K-neighborhood is used to determine the so-called decision profile of a test object. The decision profile is an evaluation of the chance that the recognized object belongs to particular classes. Next, the decision profile is compared with the response produced by the classifier and the competence is calculated according to the similarity rule. The MC systems with DES scheme using the proposed competence measure were developed and experimentally evaluated using 15 benchmark datasets. Experimental results showed that the idea of calculating the competence of a classifier by comparing its response with the decision profile of the classified object is a correct method and leads to the accurate and efficient multiclassifier systems.

Acknowledgment. This work was supported by the statutory funds of the Department of Systems and Computer Networks, Wroclaw University of Technology.

References

1. Lichman, M.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine (2013). http://archive.ics.uci.edu/ml
2. Britto, A., Sabourin, R., de Oliveira, L.: Dynamic selection of classifiers - a comprehensive review. Pattern Recogn. **47**(11), 3665–3680 (2014)
3. Cavalin, P., Sabourin, R., Suen, C.: Dynamic selection approaches for multiple classifier systems. Neural Comput. Appl. **22**(3–4), 673–688 (2013)
4. Cruz, R., Sabourin, R., et al.: META-DES: a dynamic ensemble selection framework using meta-learning. Pattern Recogn. **48**, 1925–1935 (2015)
5. Demsar, J.: Statistical comparison of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
6. Didaci, L., Giacinto, G., Roli, F., Marcialis, G.: A study of the performance of dynamic classifier selection based on local accuracy estimation. Pattern Recogn. **38**, 2188–2191 (2005)
7. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley-Interscience, New York (2001)
8. Duin, R., Juszczak, P., et al.: PR-Tools 4.1, A Matlab Toolbox for Pattern Recognition. Delft University of Technology (2007). http://prtools.org
9. Giacinto, G., Roli, F.: Methods for dynamic classifier selection. In: Proceedings of the 10th International Conference on Image Analysis and Processing, pp. 659–664 (1999)
10. Giacinto, G., Roli, F.: Dynamic classifier selection based on multiple classifier behaviour. Pattern Recogn. **34**, 1879–1881 (2001)
11. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifier. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI–20**, 226–239 (1998)
12. Ko, A., Sabourin, R., Britto, A.: From dynamic classifier selection to dynamic ensemble selection. Pattern Recogn. **41**(5), 1718–1731 (2008)
13. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience, Hoboken (2004)
14. Kurzynski, M.: On a new competence measure applied to the combining multiclassifier system. Int. J. Sig. Process. Syst. **4**(3), 185–191 (2016)
15. Kurzynski, M., Krysmann, M., Trajdos, P., Wolczowski, A.: Multiclassifier system with hybrid learning applied to control of bioprosthetic hand. Comput. Biol. Med. **69**, 286–297 (2016)
16. Meisel, W.: Potential functions in mathematical pattern recognition. IEEE Trans. Comput. **C–18**, 911–918 (1969)
17. Sabourin, M., Mitiche, A., Thomas, D., Nagy, G.: Classifier combination for handprinted digit recognition. In: Proceedings of the 2nd International Conference on Document Analysis and Recognition, pp. 163–166 (1993)
18. dos Santos, E., Sabourin, R., Maupin, P.: A dynamic over produce-and-choose strategy for the selection of classifier ensembles. Pattern Recogn. **41**(10), 2993–3009 (2008)
19. Smits, P.: Multiple classifier systems for supervised remote sensing image classification based on dynamics classifier selection. IEEE Trans. Geosci. Remote Sens. **40**, 801–813 (2002)
20. Woloszynski, T., Kurzynski, M.: On a new measure of classifier competence applied to the design of multiclassifier systems. In: Foggia, P., Sansone, C., Vento, M. (eds.) ICIAP 2009. LNCS, vol. 5716, pp. 995–1004. Springer, Heidelberg (2009). doi:10.1007/978-3-642-04146-4_106

21. Woloszynski, T., Kurzynski, M.: A probabilistic model of classifier competence for dynamic ensemble selection. Pattern Recogn. **44**, 2656–2668 (2011)
22. Woloszynski, T., Kurzynski, M., et al.: A measure of competence based on random classification for dynamic ensemble selection. Inf. Fusion **13**, 207–213 (2012)
23. Woods, K., Kegelmeyer, W., Bowyer, K.: Combination of multiple classifiers using local accuracy estimates. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI–19**, 405–410 (1997)
24. Wolpert, D.: Stacked generalization. Neural Netw. **5**, 214–259 (1992)