On a New Competence Measure Applied to the Combining Multiclassifier System

Marek Kurzynski
Department of Systems and Computer Network, Wroclaw University of Technology, Wroclaw, Poland
Email: marek.kurzynski@pwr.edu.pl

***Abstract*—In this paper a new method for calculating competence of a classifier in the feature space is presented. The idea of method is based on relating the response of the classifier with the decision profile of a test object which is evaluated using K nearest objects from the validation set. The measure of competence reflects this relation and rates the classifier with respect to the similarity of its response to the decision profile of a test object in a continuous manner. Two multiclassifier systems (MCS's) representing the dynamic classifier selection (DCS) and dynamic ensemble selection (DES) strategies are developed using proposed measure of competence. The performance of proposed MCS's was compared against seven multiple classification systems using six benchmark datasets taken from the UCI Machine Learning Repository and Ludmila Kuncheva Collection. The experimental investigations clearly show the effectiveness of the combining multiclassifier system in dynamic fashion using proposed measure of competence regardless of the ensemble type used (homogeneous or heterogeneous).**

***Index Terms*—multiclassifier system, dynamic ensemble selection, measure of competence**

I. Introduction

In the last two decades multiclassifier systems (MCS's) which combine responses of set of classifiers are intensively developed. The reason is that different classifiers offer complementary information about the object to be classified and therefore MCS can achieved better classification accuracy than any single classifier in the ensemble [1], [2].

For the classifier combination two main approaches used are classifiers fusion and classifiers selection. In the first method, all classifiers in the ensemble contribute to the decision of the MCS, e.g. through sum or majority voting [3]. In the second approach, a single classifier is selected from the ensemble and its decision is treated as the decision of the MCS. 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 [4].

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 [5]. In this way a DES based system takes advantage of both selection and fusion approaches.

In the most methods, the base classifiers are selected from the pool on the base 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 [6] 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 unknown object is assigned.

In [7], [8] 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 a random guessing. The measure of competence reflects this relation and rates the classifier with respect to the 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.

Interesting method called MCB (Multiple Classifier Behavior) was proposed in [9]. In this method the competence is defined as the classification accuracy calculated for a subset of validation set which is generated as follows. First the MCB's 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 test object are used to generate the subset.

Manuscript received February 18, 2015; revised May 22, 2015.

doi: 10.18178/ijsps.4.3.185-191

In this paper a new method for calculating competence of a classifier in the feature space is developed. In the proposed method, first the so-called decision profile of classified object is determined using K-nearest validation objects. The decision profile indicates the class with the greatest chance of being true class together with the value of this chance. Next, the decision profile is compared with the response produced by the classifier and the competence is calculated according to the similarity rule: response closer to the profile – classifier more competent.

In a nutshell, originality of proposed approach consists in other use of the validation set. In the methods described above, the validation set is directly used for calculation of local accuracy of classifier, i.e. its local competence. However, in the proposed method, validation set is used for estimate the classification profile of a test point and competence of a classifier is determining by relating its response to this estimation.

The paper is divided into four sections and organized as follows. In Section 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 against seven multiple classifier systems using six databases taken from the UCI Machine Learning Repository and Ludmila Kuncheva Collection. The results of computer experiments are described in Section 3 and Section 4 concludes the paper.

II. Multiclassifier System

A. Preliminaries

In the multiclassifier 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 $(l = 1,2,\ldots,L)$ is a function $\psi_l: X \rightarrow \mathcal{M}$ from a feature space to a set of class labels $\mathcal{M} = \{1,2,\ldots,M\}$. Classification is made according to the maximum rule:

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

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

The ensemble Ψ is used for classification through a combination function which, for example, can select a single classifier or a subset of classifiers from the ensemble, it can be independent or dependent on the feature vector x (in the latter case the function is said to be dynamic), and it can be non-trainable or trainable [1], [3]. The proposed multiclassifier systems use both dynamic classifier selection (DCS) and dynamic ensemble selection (DES) strategies with trainable selection/fusion algorithms. The basis for dynamic selection of classifiers from the pool is a competence measure $c(\psi_l, x)$ of each base classifier (l=1, 2, …, L), which evaluates the competence of classifier ψ_l i.e. its capability to correct activity (correct classification) at a point $x \in X$.

In this paper trainable competence function is proposed what leads to the assumption that a validation set containing pairs of feature vectors and their corresponding class labels is available, viz:

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

The next subsection describes the procedure of determining competence measure $c(\psi_l, x)$ of classifier ψ_l using validation set (2) in detail.

B. Measure of Classifier Competence

For the calculation of the classifier competence $c(\psi_l, x)$ at a point x, the so-called K-neighborhood of x, i.e. K nearest neighbors of x from validation set V is used. However, in contrast to other methods [5]-[7], [9], [10] the K-neighborhood is not used directly to evaluate the local accuracy of classifier, which is the basis for calculation of competence at a point x.

In the proposed method, first the K-neighborhood is used to determine the so-called decision profile of an object x. The decision profile determines the class number with the greatest chance of being true class together with the normalized (from the interval [0,1]) value of this chance. For the probabilistic model of classification task the decision profile can be interpreted as the greatest *a posteriori* probability of a class at a point x.

Next, the decision profile is compared with the support produced by classifier ψ_l at a point x for the same class. Finally, competence is calculated according to the following rule: the competence is maximum and equal to 1 if the decision profile and the classifier support are identical and the competence decreases with increasing difference between the decision profile and classifier support.

In order to determine decision profile of x let first define decision value of a validation object x_k $(k = 1,2,\ldots,N)$ as follows:

$$D_j(x_k) = \begin{cases} 1 & \text{for } j = j_k \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Decision values (3) of validation objects from the jth class $(j \in \mathcal{M})$ belonging to the K-neighborhood $V_K(x)$ of a point $x \in X$ create for a point x the class-dependent decision profile $D_j(x)$. The class-dependent decision profile $D_j(x)$ is a result of the cumulative influence of validation objects from $V_K(x)$ and from the jth class where the influence of each validation object $x_k \in V_K(x)$ decreases as the distance between x and x_k increases. This interpretation allows for using the potential function model [8] to determine the class-dependent decision profile of x as follows:

$$D_j(x) = \sum_{x_k \in V_K(x):\, j_k = j} D_j(x_k)\, G(x, x_k), \quad j \in \mathcal{M} \quad (4)$$

where $G(x, x_k)$ is a non-negative potential function decreasing with the increasing distance between x and x_k. Although any given metric can be used in the definition of the distance $dist(x, x_k)$ and potential function $G(x, x_k)$ can has any form, in this study we propose an Euclidean distance:

$$dist(x, x_k) = \sqrt{(x - x_k)^T (x - x_k)} \quad (5)$$

And a Gaussian potential function:

$$G(x, x_k) = \exp(-dist(x, x_k)\ \ ) \tag{6}$$

This function is substituted into (4) which is then normalized in order for the $D_j(x)$ to take values in the interval $[0, 1]$. This resulted in the following formula:

$$D_j(x) = \frac{\sum_{x_k \in V_K(x):\, j_k = j} D_j(x_k) \exp(-dist(x, x_k)\ \ )}{\sum_{j \in \mathcal{M}} D_j(x)} \tag{7}$$

Hence we get the decision profile of x as a greatest value of class-dependent decision profile, namely:

$$dv_i(x) = D_i(x), \text{ where } D_i(x) = \max_{j \in \mathcal{M}} D_j\,(x) \tag{8}$$

Finally, normalized competence $c(\psi_l, x) \in [0,1]$ of base classifier ψ_l at a point x is defined as follows:

$$c(\psi_l, x) = 1 - |dv_i(x) - d_{li}(x)| \tag{9}$$

C. Example

Consider a classification problem with two classes (M=2). Fig. 1 presents the 5-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 $d_1(x) = 0.4$ and $d_2(x) = 0.6$ at a point x.

Our purpose is to determine the competence $c(\psi, x)$ of the classifier ψ for an object x.

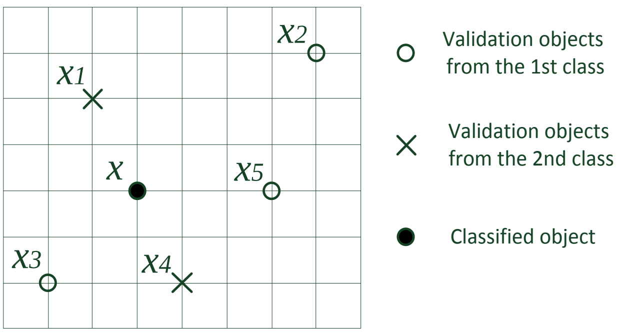

Figure 1. Illustration of example: The 5-neighborhood of an object x.

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

$$dist(x, x_1) = 2.24, \quad dist(x, x_2) = 5.00,$$
$$dist(x, x_3) = 2.83, \quad dist(x, x_4) = 2.24,$$
$$dist(x, x_5) = 3.00.$$

Now, from (6) we can calculate the class-dependent decision values of x:

$$D_1(x) = \frac{\exp(-5)+\exp(-2.83)+\exp(-3)}{D_1(x)+D_2(x)} = \frac{0.1156}{0.3285} = 0.3519,$$

$$D_2(x) = \frac{\exp(-2.24)+\exp(-2.24)}{D_1(x)+D_2(x)} = \frac{0.2129}{0.3285} = 0.6481$$

And decision profile of x as the greatest class-dependent decision value:

$$dv_2(x) = D_2(x) = 0.6481$$

Finally, from (8) we get competence of ψ for an object x:

$$c(\psi, x) = 1 - |dv_2(x) - d_2(x)| = 0.9519$$

D. Multiclassifier Systems

The proposed measure of competence can be incorporated in virtually any multiclassifier system in selection/fusion algorithm provided that feature space X is a metric space.

In this subsection we describe two multiclassifier systems based on proposed measure of competence employing both DCS and DES strategies.

1) DCS-Most competent system (DCS-MC)

In this system, first the competence $c(\psi_l, x)$ is calculated for each base classifier (l=1,2,…,L). Then the DCS-MC system ψ_{MC} selects the most competent classifier from the ensemble and uses it for the classification of x:

$$\psi_{MC}(x, \Psi, V) = i \iff d_{ki}(x) = \max_{j \in \mathcal{M}} d_{kj}(x) \tag{10}$$

And:

$$c(\psi_k, x) = \max_{l=1,2,\dots,L} c(\psi_l, x) \tag{11}$$

The DCS-MC system uses a selection strategy, i.e. for each object described by a feature vector x it selects a single classifier to be used for classification.

2) DES-Competence based system (DES-CS)

This system is based on continuous-values outputs and weighted majority voting procedure. First, a subset $\Psi_x^*(\propto)$ of base classifiers with the competences greater than the adopted threshold value α is selected for a given x:

$$\Psi_x^*(\propto) = \{\psi_{l1}, \psi_{l2}, \dots, \psi_{lT}\}, \text{ where } c(\psi_{lt}, x) > \alpha \tag{12}$$

This step eliminates inaccurate classifiers and keeps the ensemble relatively diverse. The selected classifiers are combined using the weighted majority voting rule where the weights are equal to the competences. This results in the following vector of class supports:

$$d_j^{CS}(x) = \sum_{t=1}^{T} c(\psi_{lt}, x)\, d_{lt,j}(x) \tag{13}$$

The DES-CS system ψ_{CS} classifies x using the maximum rule:

$$\psi_{CS}(x, \Psi, V) = i \iff d_i^{CS}(x) = \max_{j \in \mathcal{M}} d_j^{CS}\,(x) \tag{14}$$

The DES-CS system represents a fusion approach where the final classification is based on responses given by all competent base classifiers.

Algorithms of DCS-MC and DES-CS systems are presented in Table I and Table II in details.

TABLE I. PSEUDOCODE OF THE DCS-MC SYSTEM

Input data:
- Ψ – the pool of trained base classifiers
- V – validation set
- x – testing object
- K – the size of neighborhood

Output:
- $\psi_{MC}(x, \Psi, V) = i$ - classification result

1. Find $V_K(x)$ – the set of K nearest objects from V to x
2. **For** each class number j ∈ $\mathcal{M}$ **do**
   - $D_j(x) = 0$
   - **For** each validation point $x_k \in V_K(x)$ **do**
     - **If** $j_k = j$ **do**
       - $D_j(x) = D_j(x) + \exp(-dist(x, x_k)$
     - **End if**

End for
End for
3. **For** each class number j ∈ $\mathcal{M}$ **do**
$D_j(x) = D_j(x)/\Sigma\, D_j(x)$
End for
4. Calculate $dv_i(x) = D_i(x) = \max_{j\in\mathcal{M}} D_j(x)$
5. **For** each base classifier $\psi_l \in \Psi$ **do**
$c(\psi_l, x) = 1 - |dv_i(x) - d_{li}(x)|$
End for
6. **Find classifier** ψ_k for which
$c(\psi_k, x) = \max_{l=1,2,\ldots,L} c(\psi_l, x)$.
7. Determine decision $\psi_{MC}(x, \Psi, V) = i$ for which
$d_{ki}(x) = \max_{j\in\mathcal{M}} d_{kj}(x)$

TABLE II. PSEUDOCODE OF THE DES-CS SYSTEM

Input data:
Ψ – the pool of trained base classifiers
V – validation set
x – testing object
K – the size of neighborhood
α – competence threshold
Output:
$\psi_{CS}(x, \Psi, V) = i$ - classification result
Steps 1 - 4 as in the previous algorithm
5. $\Psi_x^*(\propto) = \emptyset,\ d_j^{CS}(x) = 0$
6. For each class number j ∈ $\mathcal{M}$ **do**
$d_j^{CS}(x) = 0$
End for
7. **For** each base classifier $\psi_l \in \Psi$ **do**
$c(\psi_l, x) = 1 - |dv_i(x) - d_{li}(x)|$
For each class number j ∈ $\mathcal{M}$ **do**
If $c(\psi_l, x) > \alpha$ then do
$d_j^{CS}(x) = d_j^{CS}(x) + d_{lj}(x)$
End if
End for
End for
8. Determine decision $\psi_{CS}(x, \Psi, V) = i$ for which
$d_i^{CS}(x) = \max_{j\in\mathcal{M}} d_j^{CS}(x)$

III. EXPERIMENTS

A. Experimental Setup

The benchmark databases used in the experiments were obtained from the UCI Machine Learning Repository [11] (Breast, Cancer Wisconsin, Glass, Image Segmentation, Vowel) and Ludmila Kuncheva Collection [12] (Laryngeal3, Thyroid). Selected databases represent classification problems with object described by continuous feature vector. For each database, feature vectors were normalized for zero mean and unit standard deviation. The training and testing datasets were extracted from each database using two-fold cross-validation. A half of objects from the training dataset formed the validation dataset V and the other half of objects formed the actual training dataset. A brief description of each database is given in Table III. The experiments were conducted using MATLAB with PRTools package [13].

TABLE III. A BRIEF DESCRIPTION OF THE DATABASES USED

Database	Source	#Objects	#Features	#Classes
Breast C.W.	UCI	699	9	2
Glass	UCI	214	9	6
Laryngeal3	LKC	353	16	3
Segmentation	UCI	2310	19	7
Thyroid	LKC	215	5	3
Vowel	UCI	990	10	11

The DCS-MC and DES-CS systems were compared against seven muticlassifier systems:

1) SB system [1]: This system selects the single best classifier in the ensemble.

2) MV system [1]: This system is based on majority voting of all classifiers in the ensemble.

3) DCS-LA system [6]: In this system competence at a testing point *x* is calculated as the percentage of the correct recognition of the *k*-nearest validation samples of *x*. *k* = 10 was chosen since for this value the DCS-LA system had the best overall performance in previous studies.

4) DCS-MLA system [10]: This system is similar to the DCS-LA system, except the local classification accuracy is estimated using weighted *k* nearest neighbours of the test object *x* that are taken from *V*.

5) DES-KE system [5]: This system dynamically selects a subset of classifiers with the perfect classification accuracy of *k* nearest neighbours of the test object *x*. The *k* nearest neighbours are taken from the validation dataset *V*. If there is no classifier with the perfect classification accuracy of all *k* nearest neighbours, the value of *k* is decreased until at least one such classifier is found. *k* = 8 was chosen since for this value the DES-KE system had the best performance.

6) DCS-RRC system [8], [14]: In this system first the competence of base classifiers is calculated using the concept of randomized reference classifier (RRC), and next the most competent classifier is selected for the classification of *x*.

RRC is a classifier whose class supports are realization of the random variables with beta probability distributions. The parameters of the distributions are chosen in such a way that, for each feature vector in a validation set, the expected values of the class supports produced by the RRC and the class supports produced by a modeled classifier are equal. This allows for using the probability of correct classification of the RRC as the competence of the modeled classifier. The competences calculated for a validation set are then generalized to an entire feature space by constructing a competence measure based on a potential function model.

7) DES-RRC system [8], [15]: This system is the same as the DCS-RRC except that the set of classifiers with the competence greater than the probability of random classification is selected for an object *x*. Decision is made using weighted majority voting rule.

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 layers and the maximum number of learning epochs set to 80. Each neural network classifier was trained using randomly selected 70% of objects from the training dataset.

The heterogeneous consisted of the following 11 classifiers [16]:

- (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 Gini splitting criterion;
- (10) Feed-Forward backpropagation neural network classifier containing two hidden layers with 5 neurons each and the maximum number of learning epochs set to 80;
- (11) Feed-Forward backpropagation neural network classifier containing one hidden layer with 10 neurons and the maximum number of learning epochs set to 80

Performance of the systems constructed was evaluated in two experiments. In the first experiment, DCS-MC and DES-CS systems were evaluated using different values of parameter K defining the K-neighborhood concept and for DES-CS system different threshold values α in (12). Experiments were conducted for K=3M, 5M, 10M (M is the number of classes) and for α=0.95, 0.9, 0.8, 0.7.

In the second experiment, the systems that showed the best performance were compared against other MCS's.

B. Results and Discussion

The average ranks of DCS-MC and DES-CS systems for different values of K and α and a critical rank difference calculated using a Bonferroni-Dunn test [17] are visualized in Fig. 2. The DCS-MC system achieved the best results for K=5M (homogeneous ensemble) and for K=3M (heterogeneous ensemble). The DES-CS system achieved the best results for K=5M and α=0.8 (homogeneous ensemble) and for K=5M and α=0.8 (heterogeneous ensemble).

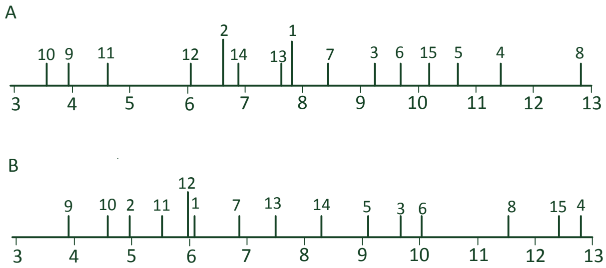

Figure 2. Average ranks of the systems constructed for different values of K and α: A) the homogeneous ensemble, B) the heterogeneous ensemble (DCS-MC for K=3M (1), 5M (2) 10M (3), DES-CS for K=3M and α=0.95 (4), 3M, 0.9 (5), 3M, 0.8 (6), 3M, 0.7 (7), 5M, 0.95 (8), 5M, 0.9 (9), 5M, 0.8 (10), 5M, 0.7 (11), 10M, 0.95 (12), 10M, 0.9 (13), 10M, 0.8 (14), 10M, 0.7 (15)).

TABLE IV. CLASSIFICATION ACCURACIES OF THE MCSS USING HOMOGENEOUS ENSEMBLES. THE BEST RESULT FOR EACH DATASET IS BOLDED.

Database	SB (1)	MV (2)	DCS-LA (3)	DCS-MLA (4)	DES-KE (5)	DCS-RRC (6)	DES-RRC (7)	DCS-MC (8)	DES-CS (9)
Breast C.W.	95.01	**96.30**	95.01	95.07	95.85	94.39	95.88	94.15	94.88
Glass	51.70 *#	55.94 *#	58.68 *#	59.34 *#	61.89 #	65.23	**67.22**	61.35	63.17
Laryngeal3	67.40 *#	70.11	69.15 #	67.51 *#	68.02 *#	70.89	**72.69**	69.55	70.05
Segmentation	84.24 *#	94.51	94.39	94.60	95.57	94.13	**95.72**	94.42	94.47
Thyroid	90.56 *#	91.68 *#	92.99	92.99	**94.21**	92.51	93.21	92.81	93.15
Vowel	48.91 *#	55.26 *#	65.12 *#	75.51 *#	78.46	78.15	**79.45**	77.52	78.35
Average rank	8.25	5.33	6.16	5.25	3.17	5.33	1.33	6.00	4.00
Average	72.97	77.30	79.22	80.84	82.33	82.55	**84.03**	81.63	82.34

TABLE V. CLASSIFICATION ACCURACIES OF THE MCSS USING HETEROGENEOUS ENSEMBLES. THE BEST RESULT FOR EACH DATASET IS BOLDED.

Database	SB (1)	MV (2)	DCS-LA (3)	DCS-MLA (4)	DES-KE (5)	DCS-RRC (6)	DES-RRC (7)	DCS-MC (8)	DES-CS (9)
Breast C.W.	**96.31**	96.29	96.14	96.14	95.25 #	95.93	96.28	96.11	96.21
Glass	66.39	64.96	64.03 *#	62.72 *#	64.20 *#	64.40 #	**67.35**	65.41	65.83
Laryngeal3	71.96	**73.92**	67.95 *#	68.81 *#	70.12 #	70.27 #	73.90	71.22	71.49
Segmentation	93.66 #	94.78	94.09	94.28	94.47	94.51	**95.32**	94.63	95.01
Thyroid	**93.95**	92.11	92.86	93.18	93.32	92.81	93.62	92.75	93.09
Vowel	86.99 *#	87.14 *#	84.03 *#	82.84 *#	84.55 *#	86.38 *#	**90.18**	88.73	89.54
Average rank	3.50	4.00	7.42	7.08	6.50	6.50	1.66	5.00	3.33
Average	84.87	84.86	83.18	82.99	83.65	84.05	**86.11**	84.81	85.19

The results obtained in the second experiments for homogeneous and heterogeneous ensembles are shown in Table IV and Table V, respectively. These results are the classification accuracies (i.e. the percentage of correctly classified objects) averaged over 10 runs (5 replications of 2-fold cross validation).

Statistical differences between the performances of the DCS-MC and DES-CS systems and the seven MCS's were evaluated using Dietterich's 5x2cv test [18]. The level of $p < 0.05$ was considered statistically significant. In Tables, statistically significant differences are marked by asterisks and hash signs with respect to the DCS-MC and DES-CS methods, respectively.

The DES systems achieved better average ranks than DCS systems, regardless of the competence method and the ensemble type used. This indicated that DES is superior to DCS and that neither of the competence methods used affects relative ranks of the systems constructed. This could be attributed to the fact that the former uses the supports of classifiers ensemble, while the latter uses supports of a single classifier.

For all MCS's classification accuracies for heterogeneous ensemble are better than for homogeneous ensemble. One possible reason for this is that learning procedure of multilayer perceptrons was restricted to the 80 epochs.

The DCS-MC (DES-CS) systems for homogeneous ensemble outperformed the SB, MV, DCS-LA and DCS-MLA systems by 28.61% (29.44%), 22.26% (23.09%), 12.4% (13.23%) and 2.01% (2.84%) on average, respectively.

The DCS-MC system for heterogeneous ensemble outperformed the DCS-LA, DCS-MLA, DES-KE and DCS-RRC systems by 1.63%, 1.82%, 1.16% and 0.76% on average, respectively.

The DES-CS system for heterogeneous ensemble that was the second-best scoring system outperformed the SB, MV, DCS-LA, DCS-MLA, DES-KE and DCS-RRC systems by 0.32%, 0.33%, 2.01%, 2.2%, 1.54 and 1.14% on average, respectively.

The systems developed produced statistically significant higher accuracies than the other MCS's in 57 out of 168 cases (6 datasets $\times$7 MCS's $\times$2 systems developed $\times$2 ensemble types).

IV. Conclusions

Nowadays, many researchers have been focused on Multiclassifier 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 dynamic ensemble selection scheme and in the weighted mechanism of combining classifiers.

In this study 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 determines the class number with the greatest chance of being true class together with the normalized value of this chance. Next, the decision profile is compared with the support produced by the classifier at a test point for the same class. Finally, we calculate the measure of competence which rates the classifier with respect to the similarity of its response to the decision profile of a test object in a continuous manner.

Two multiclassifier systems based on DCS and DES schemes using in the selection process proposed competence measure were developed and experimentally evaluated using 6 benchmark datasets.

Experimental results showed that the idea of calculating the competence of a classifier by relating its response to the decision profile of a testing object is correct and leads to the accurate and efficient multiclassifier systems.

Acknowledgment

This work was financed from the National Science Center resources in 2012-2014 years as a research project No ST6/06168 and supported by the statutory funds of the Department of Systems and Computer Networks, Wroclaw University of Technology.

References

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

[2] L. Rokah, *Pattern Classification Using Ensemble Methods*, New Jersey: World Scientific, 2010.

[3] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifier," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226-239, 1998.

[4] L. Didaci, G. Giacinto, F. Roli, and G. Marcialis, "A study on the performance of dynamic classifier selection based on local accuracy estimation," *Pattern Recognition*, vol. 38, no. 11, pp. 2188-2191, 2005.

[5] H. Ko, R. Sabourin, and A. Britto, "From dynamic classifier selection to dynamic ensemble selection," *Pattern Recognition*, vol. 41, no. 5, pp. 1718-1731, 2008.

[6] K. Woods, W. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 405-410, 1997.

[7] M. Kurzynski, T. Woloszynski, and R. Lysiak, "On two measures of classifier competence for dynamic ensemble selection – Experimental comparative analysis," in *Proc. Int. Symposium on Communication and Information Technologies*, Tokyo, 2010, pp. 1108-1113.

[8] T. Woloszynski and M. Kurzynski, "A probabilistic model of classifier competence for dynamic ensemble selection," *Pattern Recognition*, vol. 44, no. 10, pp. 2656-2668, 2011.

[9] G. Giacinto and F. Roli, "Dynamic classifier selection based on multiple classifier behavior," *Pattern Recognition*, vol. 34, no. 9, pp. 1879-1881, 2001.

[10] P. C. Smits, "Multiple classifier systems for supervised remote sensing image classification based on dynamics classifier selection," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 40, no. 4, pp. 801-813, 2002.

[11] A. Asuncion and D. Newman, *UCI Machine Learning Repository*, Irvine, CA: University of California, Department of Information and Computer Science, 2007.

[12] L. Kuncheva. (2004). *Ludmila Kuncheva collection*. [Online]. Available: http://www.bangor.ac.uk/ mas00a/activities/patrec1.html

[13] R. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. Ridder, and D. Tax, "PR-Tools 4.1, a matlab toolbox for pattern recognition," Delft University of Technology, 2007.

[14] T. Woloszynski and M. Kurzynski, "A measure of competence based on randomized reference classifier for dynamic ensemble selection," in *Proc. 20th Int. Conf. on Pattern Recognition*, Istanbul, 2010, pp. 4194-4197.

[15] T. Woloszynski and M. Kurzynski, "A measure of competence based on random classification for dynamic ensemble selection," *Information Fusion*, vol. 12, pp. 378-392, 2012.

[16] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, New York: John Wiley and Sons, 2000.

[17] J. Demsar, "Statistical comparison of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1-30, 2006.

[18] T. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, pp. 1895-1923, 1998.

Marek Kurzynski received Ph.D. in automatic control from the Wroclaw University of Technology (Poland) in 1974 and 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, Wroclaw University of Technology, Poland. Prof. Kurzynski has been the recipient, investigator and co-investigator of numerous research grants from EU, Polish Ministry of Science and other industrially founded research projects. He has published more than 180 refereed journal and conference papers and four books. He is an Associated Editor of the Journal of Applied Computer Science, the Int. Journal of Applied Mathematics and Computer Science and the Int. Journal of Machine Graphics and Vision. He was also one of the editors of the book series on Computer Recognition Systems (2005, 2007, 2009, 2011, 2013) published by Springer Verlag. His research interests include pattern recognition, artificial intelligence methods, biological signal processing and medical informatics.