

On a New Method for Improving Weak Classifiers Using Bayes Metaclassifier

Marcin Majak^(✉) and Marek Kurzyński

Wrocław University of Science and Technology, Wrocław, Poland
{marcin.majak,marek.kurzynski}@pwr.edu.pl

Abstract. In this paper new algorithm called Bayes metaclassifier (BMC) will be introduced as a method for improving weak classifiers performance. In general, BMC constitutes the probabilistic generalization of any base classifier and has the form of the Bayes scheme. To validate BMC classification two experiments were designed. In the first one three synthetic datasets were generated from normal distribution to calculate and check empirically upper bound for improving base classifier when BMC approach is applied. Furthermore, to validate usefulness of this algorithm extensive simulations from 22 available benchmarks were performed comparing BMC model against 8 base classifiers with different design paradigms.

Keywords: Classification · Improving weak classifiers · Randomized reference classifier

1 Introduction

Pattern recognition task was successfully applied in many areas from medical, financial and critical safety applications. Each problem requires its individual approach and there is no easy solution which classifiers or algorithm should be applied because each method has its pros and cons. Nowadays, due to the huge development of the artificial intelligence, many classification methods are at hand, there are statistical, distance based, neural methods to name only a few [4, 6, 14]. Since classification problems are very complex and multidimensional, simple base classifier usage is not sufficient so advanced method for building multiclassifier systems (MC) are developed, but still proper base classifier choice is critical because we need accurate and diverse ones [19]. In the literature one can find many method for constructing robust MC models, but the most commonly used are bagging, boosting and random subspace [5, 12, 13]. In shortly, bagging applies sampling with replacement to obtain independent training datasets for each individual classifier. Boosting modifies the input data distribution processed by each classifier in a sequence from the results of classifiers trained before, paying more attention on difficult samples. All the aforementioned algorithms focus on using weak classifiers for building powerful MC systems. In this paper

we introduce an algorithm called Bayes metaclassifier (BMC) as a method for improving weak classifier in terms of its classification performance. In general, BMC constitutes the probabilistic generalization of any base classifier independent of its design paradigm and has the form of the Bayes scheme. Since BMC provides probabilistic interpretation for base classifier correct classification and misclassification, this method can be used in sequential classification or as a fusing mechanism in MC systems [10,11]. The main goal of this study is to investigate how BMC can improve base classifier performance. For this purpose two experiments were performed, the first one with synthetic data and the second one on 22 real life benchmark datasets. The paper is divided into four sections and organized as follows. In Sect. 2 pattern recognition task is introduced and later the concept of probabilistic Bayes metaclassifier is described. The results of computer experiments are described in Sect. 3 and Sect. 4 concludes the paper together with proposition of future plans and improvements.

2 Problem Statement

2.1 Pattern Recognition Task

This paper deals with pattern recognition task in which we assume that the pattern is in class $j \in \mathcal{M}$, where \mathcal{M} is an m -element set of possible states numbered with the successive natural numbers ($j \in \mathcal{M} = \{1, 2, \dots, M\}$). Label j is unknown and does not undergo our direct observation. What we can only observe are the features by which an object manifests itself. We will denote a d -dimensional measured feature vector by $x \in \mathcal{X}$ (thus \mathcal{X} is the feature vector space). In order to classify unknown patterns, as usual in practice, we assume that we have to our disposal so called training set, which in the investigated decision task consists of N training patterns:

$$\mathcal{S} = \{(x_1, j_1), \dots, (x_N, j_N)\}, \quad (1)$$

where x_k, j_k denote d -dimensional pattern and its true classification, respectively. In general, the decision algorithm with learning should use every time as well observed data i.e. the feature vector x as the knowledge included in the training set \mathcal{S} . In consequence, the general, classification algorithm with learning is of the following form:

$$i = \psi(\mathcal{S}, x), i \in \mathcal{M}. \quad (2)$$

ψ is a canonical model of a base classifier which means that for a given $x \in \mathcal{X}$, it produces class label and a vector of class supports [8]. Single $d_j(x)$ indicates the support of ψ to the hypothesis that the object x belongs to the class j .

$$d(x) = [d_1(x), d_2(x), \dots, d_M(x)], \sum_{i=1}^M d_i(x) = 1. \quad (3)$$

2.2 BMC Algorithm Construction

In this section BMC algorithm construction over single classifier will be presented. Firstly, the probabilistic model of classification must be introduced in which the feature vector $x \in \mathcal{X}$ and class label $j \in \mathcal{M}$ are observed values of the pair of random variables \mathbf{X} and \mathbf{J} , respectively. The probability distribution of (\mathbf{X}, \mathbf{J}) is determined by the *a priori* class probabilities $p_j = P(\mathbf{J} = j)$ and class-conditional density functions $f(x|j) = f_j(x)$. The Bayes metaclassifier (BMC) ψ^{BMC} , which originally was introduced in [11], constitutes the probabilistic generalization of base classifier (2) which has the form of the Bayes scheme built over the classifier ψ . This means, that ψ^{BMC} takes the decision according to the maximum *a posteriori* probability rule [10]:

$$\psi^{BMC}(\psi(x) = k|i) \longleftrightarrow p(i|\psi = k) = \max_{i \in \mathcal{M}} p(i|\psi = k). \quad (4)$$

A posteriori probabilities $p(i|k) \equiv P(\mathbf{J} = i|\psi(x) = k)$, $i \in \mathcal{M}$ are given by Bayes rule:

$$P(\mathbf{J} = i|\psi = k) = \frac{p_i p(k|i)}{\sum_j p_j p(k|j)}, \quad (5)$$

where probability $p(k|i) \equiv P(\psi(x) = k|i)$ denotes class-dependent probability of error (if $k \neq i$) or correct (if $k = i$) classification of an object x by the base classifier ψ . Placing the base classifier ψ in a probabilistic frame defined by the BMC (ψ^{BMC}), we get a common probabilistic interpretation of responses of base classifiers, regardless of their design paradigms. The key element in the BMC scheme described by Eqs. (4) and (5), is the calculation of probabilities $P(\psi(x) = k|i)$ at point x , i.e. class-dependent probabilities of correct and misclassification for base classifier ψ . Normally, for any base deterministic classifier these probabilities are equal to 0 or 1. In this paper, the proposed method of evaluation of these probabilities is based on the original concept of a hypothetical classifier called a Randomized Reference Classifier (RRC) [16, 17]. The RRC $\psi_i^{RRC}(x)$ is a stochastic classifier that classifies object x according to the maximum rule for vector of class support $[\gamma_1(x), \gamma_2(x), \dots, \gamma_M(x)]$ which are observed values of random variables (rvs) $[\Delta_1(x), \Delta_2(x), \dots, \Delta_M(x)]$. Probability distribution of rvs is chosen in such a way that RRC acts, on average, as a modeled base classifier only when the following constraints are fulfilled:

$$\Delta_j(x) \in \langle 0, 1 \rangle, \quad \sum_{j=1}^M \Delta_j(x) = 1 \quad (6)$$

$$E[\Delta_j(x)] = d_j(x), \quad j \in \mathcal{M},$$

where E is the expected value operator. Since RRC performs classification in a stochastic manner, so it is possible to calculate class-dependent probabilities of correct classification $P_c(j|x)$ and misclassification $P_e(j|x)$ and furthermore consider them equivalent to the modeled base classifier:

$$P(\psi(x) = k|i) \approx P(\psi^{RRC}(x) = k|i). \quad (7)$$

It should be noted that (7) is a heuristic approach for estimating base classifier probabilities of correct classification (misclassification). For BMC training phase it is assumed that a validation set containing pairs of feature vectors and their corresponding class labels is available [15]. This set can be obtained by random ratio subsampling from training set or alternatively as a stack generalization [18]:

$$\mathcal{V} = \{(x_1, j_1), (x_2, j_2), \dots, (x_N, j_N)\}; \quad x_k \in \mathcal{X}, \quad j_k \in \mathcal{M}. \quad (8)$$

\mathcal{V} is used to calculate $P(\psi(x) = k|i) \approx P(\psi_i^{RRC}(x) = k|i)$ which denotes that an objects x belongs to class i given that $\psi(x) = k$. These values are only known at discrete points from \mathcal{V} so to enable dynamic calculation of any new object x from feature space we need a neighborhood function describing how probabilities at validation points affects new x . In this study, Gaussian potential function was used:

$$P_c^{RRC}(j|x) = \frac{\sum_{x_k \in \mathcal{V}, j_k=j} P_c(j|x) \cdot \exp(-\alpha \cdot \|x, x_k\|^2)}{\sum_{x_k \in \mathcal{V}, j_k=j} \exp(-\alpha \cdot \|x, x_k\|^2)}, \quad (9)$$

$$P_e^{RRC}(j|x) = \frac{\sum_{x_k \in \mathcal{V}, j_k \neq j} P_e(j|x) \cdot \exp(-\alpha \cdot \|x, x_k\|^2)}{\sum_{x_k \in \mathcal{V}, j_k \neq j} \exp(-\alpha \cdot \|x, x_k\|^2)},$$

where α value in Eq. (9) is a scaling factor and should be adjusted independently to classification problem. In this article, instead of manual selection, α is selected during preparation step of BMC using training and validation set. In this case, for each object from \mathcal{V} we calculate $P_e(j|x)$, $P_c(j|x)$ and further we randomly select 40% of objects from training set with stratification and use this new set to check for which α BMC obtains the highest classification rate. For all experiments, α was chosen from set: $\alpha = \{1, 2, 4, 6, 8\}$.

3 Experiments

The goal of this study was twofold. Firstly, we wanted to check how BMC works with synthetically generated data coming from 1D normal distribution described by (μ, σ) . This is very simplified approach, but allows us to calculate an optimal Bayes decision boundary and Bayes probability of misclassification. In a consequence, we can determine empirical probability of error of BMC in relation to this probability. In the second experiment BMC model was applied to 22 benchmark data sets taken from the UCI Machine Learning Repository [3], Ludmila Kuncheva Repository [9] and Keel [1] for 8 classifiers with different decision paradigm. A brief description of the data sets used is given in Table 1.

3.1 Experimental Setup

In the first experiment three synthetic datasets with two classes were generated from normal distribution $N(\mu, \sigma)$ using different (μ, σ) (see Fig. 1). For generated

set it was confirmed (for $p \ll 0.01$) that each sample is not statistically significantly different from normal distribution using Kolmogorow Smirnow, Lillieforsa and W Shapiro-Wilka statistical tests. For simulations, 50k and 10k objects were generated for each class for testing and validation sets, and *a priori* class probabilities were equal. Each simulation was repeated 10 times and results were averaged. Since we deal with scalar values, we use linear classifier (ψ_l) which decision boundary location is determined by point $x = p$ so training procedure is not needed. Furthermore, using $\mu_{1,2}, \sigma_{1,2}$ for each model an optimal Bayes decision boundary can be calculated and set as $p_{optimal}$.

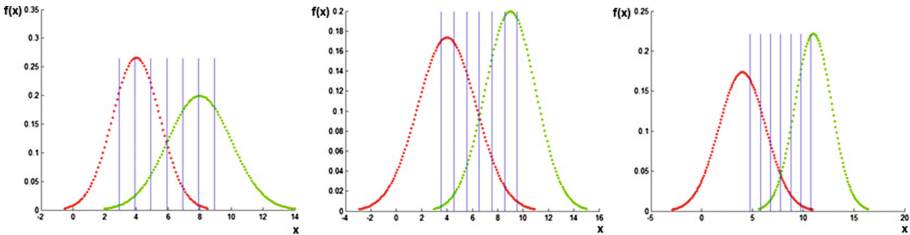


Fig. 1. Pdf functions for three synthetic datasets used in the first experiment. From the left: (1) $\mu_1 = 4, \sigma_1 = 1.5, \mu_2 = 8, \sigma_2 = 2.0$, (2) $\mu_1 = 4, \sigma_1 = 2.3, \mu_2 = 9, \sigma_2 = 2.0$, (3) $\mu_1 = 4, \sigma_1 = 2.3, \mu_2 = 11, \sigma_2 = 1.8$

For the second experiment the following base classifiers were used:

- **qdc** - quadratic classifier based on normal distributions with the same covariance matrix for each class,
- **nmc** - nearest mean classifier,
- **6-nn** - 6 nearest neighbors,
- **parzen** - Parzen classifier,
- **dtc** - Decision tree with information gain splitting criterion,
- **bpxn-1** - Feed-forward back-propagation neural network with 1 hidden layer and the number of learning epochs set to 120,
- **bpxn-2** - Feed-forward back-propagation neural network with 2 hidden layer and the number of learning epochs set to 120,
- **svm** - support vector machine with 3rd order polynomial kernel.

For each data set from the second experiment, feature vectors were normalized to zero mean and unit standard deviation. Simulations were conducted according to 5×2 cross-validation method for extracting training and testing sets from each data set [2]. For the comparison purposes the following approach was applied. As a reference, each base classifier was trained using whole training set, while for BMC preparation, from available training set 30% of objects were selected randomly to form validation set while the rest objects were used for base classifier fitting. All experiments were conducted in MATLAB using PRTools 5.0 packet.

Table 1. Datasets used in experiments

#	Database	Source	# Objects	# Features	# Classes	<i>a priori</i> probability
1	Weaning	LKC	302	17	2	.5, .5
2	Parkinson	UCI	195	22	2	.75, .25
3	Transfusion	UCI	748	4	2	.76, .24
4	Wisconsin	UCI	683	9	2	.65, .35
5	Dermatology	UCI	358	34	6	.3, .2, .2, .1, .1, .1
6	Glass	UCI	214	9	6	.3, .4, .1, .1, .0, .1
7	Haberman	UCI	306	3	2	.7, .3
8	Mamographics	Keel	830	5	2	.5, .5
9	Iris	UCI	150	4	3	$3 \times \frac{1}{3}$
10	Laryngeal3	LKC	353	16	3	.2, .6, .2
11	Voice3	LKC	238	16	3	.2, .1, .7
12	Pima	UCI	768	5	2	.3, .7
13	Thyroid	UCI	215	5	3	.7, .2, .1
14	Vowel	UCI	990	10	11	$11 \times \frac{1}{11}$
15	Segmentation	UCI	2310	19	7	$7 \times \frac{1}{7}$
16	Yeast	UCI	1484	8	10	.164, .289, .312, .030, .034, .110, .024, .020, .014, .003
17	Sonar	UCI	208	60	2	.5, .5
18	OptDigits	UCI	3823	64	10	$10 \times \frac{1}{10}$
19	Spam	UCI	4601	57	2	.4, .6
20	Statlog	UCI	270	13	2	.45, .55
21	Page blocks	UCI	5473	10	5	.90, .06, .01, .01, .02
22	Phoneme	Keel	5404	5	2	.3, .7

3.2 Experiment Results

Results for the first simulation scenario are presented in Table 2 where 7 classification error rates are shown for ψ_l with decision boundary set to: $x = p_{optimal} - 3, x = p_{optimal} - 2, x = p_{optimal} - 1, x = p_{optimal}, x = p_{optimal} + 1, x = p_{optimal} + 2, x = p_{optimal} + 3$. After analyzing results in Table 2 it can be seen that the upper bound for BMC improvement is Bayes error calculated for optimal decision boundary. Higher correction can be obtained for classifier ψ_l when decision boundary is farther from the optimal. One should keep in mind, that this experiment is very simple, but it can prove that BMC works according to Bayes scheme. Table 3 shows results for base classifiers versus BMC overlay for benchmark dataset. Classification accuracies were averaged over 5 repetitions of two-fold cross-validation. We have performed pairwise and multiple comparison statistical tests. For the first one 5×2 CV F-test was applied to indicate which

Table 2. Classification error rates for the first experiment with synthetic data. Numbers: **1, 2, 3** denote generated datasets. Notation used in the first column is as follows: p - dividing (boundary) point, C - an analytic calculation of the Bayes probability of error for decision boundary at point $x = p$, ψ_l - an average classification error rate for linear classifier with decision boundary at point $x = p$, BMC - an average classification error rate for ψ_l with applied BMC scheme.

P	$p_{optimal - 3}$	$p_{optimal - 2}$	$p_{optimal - 1}$	$p_{optimal}$	$p_{optimal + 1}$	$p_{optimal + 2}$	$p_{optimal + 3}$
1							
C	0.384	0.270	0.165	0.125	0.161	0.245	0.339
ψ_l	0.389	0.271	0.165	0.121	0.153	0.238	0.338
BMC	0.122	0.124	0.158	0.121	0.121	0.122	0.122
2							
C	0.291	0.210	0.146	0.122	0.148	0.217	0.308
ψ_l	0.284	0.208	0.141	0.119	0.145	0.211	0.303
BMC	0.119	0.120	0.120	0.119	0.125	0.120	0.120
3							
C	0.184	0.111	0.061	0.044	0.064	0.128	0.227
ψ_l	0.180	0.108	0.062	0.044	0.062	0.128	0.225
BMC	0.044	0.044	0.044	0.044	0.048	0.044	0.044

algorithm is better. In Table 3 $+/-$ means that the BMC system is statistically significantly better/no statistically significant difference/statistically significantly worse, respectively. For assessing the ranks of proposed methods over all examined benchmarks, we have used Friedman ranking test and after that the Shaffer post-hoc test to check which of the tested methods are distinctive among $n \times n$ comparisons [7]. The level of $p < 0.05$ was considered as statistically significant. From the Friedman ranking test, which checks the overall performance on all available datasets BMC proposal was 5 out of 8 cases statistically significantly better than a base classifier. Major improvement can be observed for problems: 1, 5, 8, 9, 13, 14, 15, 17, 18. This indicates that BMC works better with balanced datasets, while for imbalanced problems application of a priori probabilities in BMC decision scheme causes to prefer majority class. For imbalanced problems such as: 12, 21, 22 BMC does not improve classifier accuracy. This can be explained by the fact that random sampling from training set is used to create validation set. In this case, \mathcal{V} contains mainly examples from the majority class. Another important conclusion coming from this experiment is presented in Fig. 2 which shows a box plot for an exemplary laryngeal database for selected base classifier versus BMC algorithm. It indicates that apart from improving classification performance also BMC variance between subsequent cross-validation sets is decreased.

Table 3. Classification accuracies (in percentage) for the second experiment using different base classifiers. For each comparison, the first column presents results for base classifier trained with the whole training set and the second one contains accuracies for BMC algorithm. +/- =/- means that the BMC system is statistically significantly better/no statistically significant difference/statistically significantly worse ($p = 0.05$). F stands for Friedman ranking test checking overall performance on all databases.

	qdc	nmc	6-nn	parzen	dtc	bpxnc-1	bpxnc-2	svm								
1	89.6	90.2 =	77.9	90.5 +	68.7	73.3 +	76.4	76.7 +	75.4	75.9 =	55.2	74.3 =	53.3	74.5 =	82.6	83.0 =
2	72.1	86.8 +	74.6	86.0 +	84.1	85.1 =	88.2	88.5 +	81.2	84.3 =	74.5	83.3 =	72.5	86.2 =	85.8	86.6 =
3	75.3	76.6 =	75.8	76.6 =	77.3	76.6 =	76.6	76.6 =	75.2	76.6 =	69.3	76.5 =	76.1	76.6 =	76.0	76.4 +
4	96.0	96.5 =	95.7	96.2 =	96.3	96.2 =	96.5	96.9 +	93.8	95.1 =	80.9	95.7 =	88.7	95.2 =	95.3	96.1 =
5	18.9	94.1 +	95.3	95.1 =	94.6	95.5 +	94.3	94.7 +	93.8	95.3 =	17.2	93.9 +	18.9	94.7 +	93.5	93.9 +
6	7.5	57.0 +	52.0	55.6 =	63.4	62.6 =	59.9	60.6 =	63.5	64.3 =	13.6	50.7 +	22.9	53.7 +	60.2	60.6 =
7	74.8	73.5 =	74.8	73.5 =	72.9	73.5 =	73.4	73.5 =	72.0	73.3 =	73.0	74.0 +	71.0	73.5 =	74.1	73.5 =
8	80.9	81.7 +	81.8	82.2 +	79.7	80.4 =	80.5	80.9 +	82.8	83.2 +	68.1	80.6 =	68.8	80.8 =	82.0	81.3 =
9	94.1	94.5 +	95.6	96.0 +	95.5	95.9 +	96.3	96.7 +	93.5	93.9 +	39.3	90.8 +	31.7	93.1 +	95.6	96.0 +
10	62.3	65.4 =	71.2	68.6 =	71.9	72.6 =	72.4	71.7 =	66.3	69.5 =	43.9	67.6 =	51.7	66.6 =	72.3	72.1 =
11	62.9	71.9 =	74.7	70.9 =	77.7	72.9 =	71.2	69.5 =	70.4	73.2 =	55.5	70.3 =	67.6	70.7 =	75.8	72.2 =
12	73.8	74.8 =	74.4	72.9 =	72.2	73.6 =	73.1	73.1 =	71.0	71.4 =	61.3	67.2 =	64.5	66.8 =	75.3	69.3 =
13	96.6	97.0 +	91.2	91.6 +	89.9	90.3 +	93.0	93.4 +	90.6	91.2 =	65.6	77.7 =	43.6	77.5 +	92.0	88.7 =
14	63.6	72.0 +	53.4	69.1 +	65.4	68.6 +	88.7	88.7 =	61.4	68.5 +	7.4	61.7 +	9.7	62.1 +	56.9	77.6 +
15	62.4	82.2 =	81.3	84.7 =	80.4	84.0 =	82.1	82.8 =	81.1	83.6 =	43.7	77.6 +	45.0	77.8 +	88.5	90.2 =
16	21.6	43.1 +	50.6	53.5 =	54.9	52.3 =	55.1	54.8 =	56.2	55.8 =	26.0	37.7 =	33.6	37.5 =	36.6	42.4 =
17	68.4	84.6 +	71.2	85.5 =	69.5	84.2 +	82.9	83.8 =	68.4	71.5 +	74.3	78.5 +	77.3	81.0 +	85.1	83.9 =
18	62.5	96.8 +	91.7	96.5 +	97.3	97.9 +	98.1	98.5 +	85.2	94.6 +	25.0	97.4 +	28.1	97.5 +	98.5	98.9 +
19	80.4	86.9 +	83.9	85.1 =	87.1	87.5 =	85.0	85.0 =	89.8	89.8 =	71.1	70.4 =	75.3	75.2 =	66.2	70.4 =
20	80.6	82.7 =	86.4	85.0 =	85.7	84.8 =	85.1	84.8 =	82.9	83.3 =	85.2	85.3 =	85.3	85.6 =	81.9	84.4 =
21	83.4	91.4 =	74.5	90.5 +	94.9	90.9 =	94.1	91.5 =	96.6	93.2 =	89.8	89.9 =	89.8	89.9 =	94.6	93.5 =
22	75.9	70.7 =	73.5	70.7 =	84.4	84.4 =	85.5	85.5 =	83.5	83.5 =	74.6	71.1 =	77.0	71.5 =	78.6	70.7 =
F	+	10 =	6 =	5 +	6 +	5 +	8 +	7 =	3							

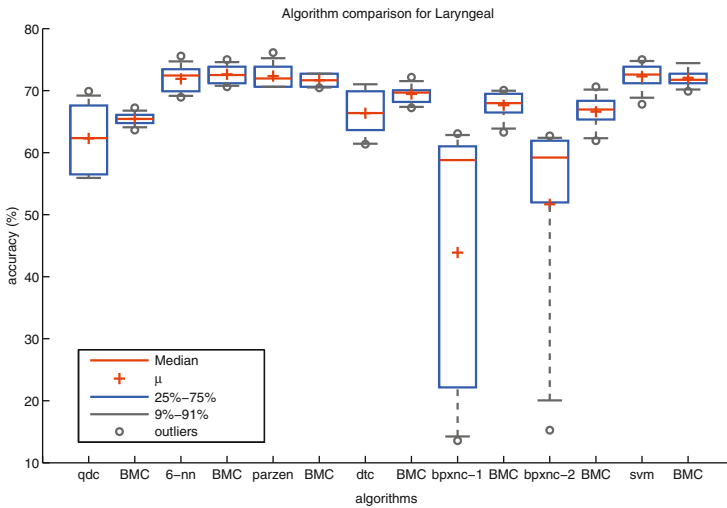


Fig. 2. Box plot for laryngeal dataset for selected base classifier versus BMC algorithm.

4 Conclusions

In this paper a detailed BMC algorithm construction was presented and validated. The first experiment confirmed that the upper bound of BMC improvement over base classifier is the Bayes error. This is not very surprising since BMC works according to Bayes scheme maximizing *a posteriori* probabilities for decision of a base classifier. From three synthetic dataset analysis it is straightforward that BMC performance directly depends on the classification quality of a base classifier ψ . Further experiment on benchmark datasets shows that BMC can significantly correct base classifier classification, especially for cases where base classifier is not optimal. Overall performance according to Friedman ranking test indicates that BMC proposal was statistically significantly better 5 out of 8 different classifiers. Additionally, when analyzing results from each cross-validation phase, it is clearly visible that BMC decreases classification variance comparing to its base counterpart results. On the other hand, the main drawback of BMC involves proper validation set generation. BMC works better with balanced datasets, while for imbalanced problems usage of *a priori* probabilities in BMC design, causes that its decision boundary is moved towards majority class. To sum up, since BMC provides probabilistic interpretation for a base classifier response for correct and incorrect classification this method can be directly used in a sequential classification problems, or in MC systems, especially during classifier fusion. As a future work, we would like to apply BMC for ensemble construction at both crisp and continuous level of classifier response.

Acknowledgments. This work was supported by the statutory funds of the Department of Systems and Computer Networks, Wrocław University of Science and Technology.

References

1. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: data set repository. *J. Multiple-Valued Logic Soft Comput.* **17**, 255–287 (2011)
2. Alpaydin, E.: Combined 5×2 cv F test for comparing supervised classification learning algorithms. *J. Neural Comput.* **11**, 1885–1892 (1999)
3. Asuncion, A., Newman, D.: UCI machine learning repository (2007). <http://www.ics.uci.edu/mllearn/MLRepository.html>
4. Duda, R., Hart, P., Stork, D.: *Pattern Classification*, 3rd edn. Wiley-Interscience, New York (2001)
5. Freund, Y.: Boosting a weak learning algorithm by majority. *Inf. Comput.* **121**(2), 256–285 (1995)
6. Fukunaga, K.: *Introduction to Statistical Pattern Recognition*, 2nd edn. Academic Press Professional Inc., San Diego (1990)
7. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Inf. Sci.* **180**, 2044–2064 (2010)

8. Kuncheva, L.: *Combining Pattern Classifiers: Methods and Algorithms*, 3rd edn. Wiley-Interscience, New York (2004)
9. Kuncheva, L.: Real medical data sets repository (2004). http://pages.bangor.ac.uk/~mas00a/activities/real_data.htm
10. Kurzynski, M., Majak, M.: Meta-Bayes classifier with Markov model applied to the control of bioprosthetic hand. In: Czarnowski, I., Caballero, A.M., Howlett, R.J., Jain, L.C. (eds.) *Intelligent Decision Technologies 2016*. SIST, vol. 57, pp. 107–117. Springer, Cham (2016). doi:[10.1007/978-3-319-39627-9_10](https://doi.org/10.1007/978-3-319-39627-9_10)
11. Kurzynski, M., Majak, M., Zolnierek, A.: Multiclassifier systems applied to the computer-aided sequential medical diagnosis. *J. Biocybern. Biomed. Eng.* **36**, 619–625 (2016)
12. Skurichina, M., Duin, R.P.W.: Bagging, boosting and the random subspace method for linear classifiers. *Pattern Anal. Appl.* **5**(2), 121–135 (2002)
13. Skurichina, M., Duin, R.P.W.: Bagging and the random subspace method for redundant feature spaces. In: Kittler, J., Roli, F. (eds.) *MCS 2001*. LNCS, vol. 2096, pp. 1–10. Springer, Heidelberg (2001). doi:[10.1007/3-540-48219-9_1](https://doi.org/10.1007/3-540-48219-9_1)
14. Strunk Jr., W., White, E.B.: *Statistical Decision Theory and Bayesian Analysis*, 3rd edn. Springer, New York (1987)
15. Woloszynski, T.: Classifier competence based on probabilistic modeling (ccprmod.m) (2013)
16. Woloszynski, T., Kurzynski, M.: A probabilistic model of classifier competence for dynamic ensemble selection. *Pattern Recogn.* **44**, 2656–2668 (2011)
17. Woloszynski, T., Kurzynski, M.: A measure of competence based on random classification for dynamic ensemble selection. *Inf. Fusion* **13**, 207–213 (2012)
18. Wolpert, D.: Stacked generalization. *Neural Netw.* **5**, 241–259 (1992)
19. Wozniak, M., Grana, M., Corchado, E.: A survey of multiple classifier systems as hybrid systems. *Inf. Fusion* **16**, 3–17 (2014)