Biocybernetics and Biomedical Engineering
2012, Volume 32, Number 2, pp. 17–31

Boosting, Bagging and Fixed Fusion Methods Performance for Aiding Diagnosis

MAŁGORZATA ĆWIKLIŃSKA-JURKOWSKA*

Department Theoretical Foundations of Biomedical Sciences and Medical Informatics, Group of Mathematical Modeling, Collegium Medicum UMK, Bydgoszcz, Poland

Multiple classifier fusion may generate more accurate classification than each of the constituent classifiers. The aim was to examine the ensemble performance by the comparison of boosting, bagging and fixed fusion methods for aiding diagnosis. Real-life medical data set for thyroid diseases recognition was applied. Different fixed combined classifiers (mean, average, product, minimum, maximum, and majority vote) built on parametric and nonparametric Bayesian discriminant methods have been employed. No very significant improvement of recognition rates by a fixed classifier combination was achieved on the examined data. The best performance was obtained for resampling methods with classification trees, for both the bagging and the boosting combining methods. The bagging and the boosting logistic regression methods have proven less efficient than the bagging or the boosting of neural networks. Difference between the bagging and the boosting performance for the examined data set was not obtained.

K e y w o r d s: thyroid disease diagnosis, combining classifiers performance, bagging, boosting, trees, logistic regression, neural networks

1. Introduction

If in a discriminant problem we have a small learning set relative to the number of variables, e.g. if a data set is high dimensional, it is often difficult to build a good single classifying function. Such a classifier is biased or has a large variance and, consequently, a poor performance. In order to improve the generalization properties of a weak classifier (which has a poor performance) by stabilizing its decision, the techniques of regularization or the method of noise injection have been developed [1]. The further approach is merging of the classifiers into a power decision rule.

* Correspondence to: Małgorzata Ćwiklińska-Jurkowska, Department Theoretical Foundations of Biomedical Sciences, Collegium Medicum, Nicolaus Copernicus University, ul. Jagiellońska 13, 85-067 Bydgoszcz, Poland, e-mail: mjurkowska@cm.umk.pl
Received 20 June 2011; accepted 17 February 2012

This idea comes from the following considerations. Individual learners perform well in some situations and fail under other conditions. Thus the identification of the best classifier is often not easy, especially when the performance of the learners is assessed for train sets with restricted number of cases. The results of many studies that have led to the conclusions that no single discriminant function is applicable to all problems have stimulated the technique of combining of classifiers. Instead of often no optimal selecting only one best single learner, we may advance the classification performance by combining the "rival" learners. Two types of combining of classifiers are known: classifier selection (e.g. selection of the classifier with best performance in the input subspace, where the observation belongs- i.e. selection of the classifier which is an expert in some local area) and classifier fusion (ensemble of classifiers). In the paper the latter technique is used for combining.

Joining of classifiers to achieve a higher accuracy is an important research topic and is developed nowadays in many different directions. The aim of merging of individual classifiers is also elimination of a possible loss of information. Duin et al. [2] showed that although the individual classification performances on the difficult datasets are weak, they can still provide valuable information for the combining rules. Combining of classifiers may create a relationship not available in any base classifier. The performance of the combining method is usually expected to be better than (mean) recognition rate of the constituent classifiers.

Combining the classifiers is a way of model variance reducing, though in certain situations it also reduces bias. Multiple classifier systems have been created in a variety of pattern recognition fields [3].

There are several approaches to obtain the ensemble of different classifiers, e.g. a linear combination of the estimated conditional class probabilities, averaging of the resulting classifiers' parameters or a majority voting of the predictions of the individual classifiers.

The purpose of the study is an experimental comparison of classification errors of three ensemble methods: simple fixed combining of classification rules (that are built from three different Bayesian discriminant methods) and bagging as well as boosting of classification trees, logistic regression or neural networks based on the dataset for the thyroid disease recognition.

2. Materials and Methods

The idea of combining of classifiers was applied to assess the ensemble performance with the usage of a real medical dataset. The big data set of thyroid disease records (3 groups) was used (//www.ics.uci.edu/~mlearn/MLRepository.html) [Accessed 2011, February 1]. The training and the testing sets consisted of 3772 and 3428 examples, respectively. All of 21 variables (15 binary and 6 continuous)

were considered. The data set has a big size of learning groups as compared to the dimensionality, but the difficulty is that the size of third group is much smaller than the others.

2.1. Combined Classifiers

In many applications of statistics in biomedical sciences, the challenge is to classify new observations according to multiple correlated variables and classifiers built from these variables. However, the results may diverge substantially depending on the choice of the specific classifier. One option is to attempt a different procedure of the classifiers "fusion" into a one combining rule. Combining of information is very important in modern research domains. For example, in modern statistics the statistical tests based on combined results have been developed. The idea of information ensemble is very intensively developed in multidimensional discriminant analysis. Intuitive justification of fairness of information joining concept may be presented in the following way. Averaged measurements are usually more correct than a single outcome, if the individual measurements are more often different than similar. Additionally, averaging of the measurements is the more stable method than taking into account only an individual mensuration. The corresponding fact is met in the discriminant methods: a weighted average of outputs is often more accurate and more stable than an individual discriminant function result. Multi-sensor data fusion (e.g. joining the recognition of face, handwritten text and voice or merging outcomes from different wave frequency ranges) and decision ensemble has been applied in the modern classifiers methods.

The simplest situation, in which combining is very efficient, is the one, when we have a set of two-groups two-dimensional observations arranged in a plane in the way similar to four points illustrating well known XOR task (exclusive alternative problem). It is known that Vapnik-Chervonekis dimension (cardinality of the largest set of points that the algorithm can shatter) of single linear classifier is equal to 3. However, for four points described above no linear discriminant function can manage separating the points correctly (each linear function has an error bigger or equal to 0.5), while the simplest procedure of the combining only two linear discriminant functions constructs a perfect classifier. Another clear example of pooling of classifiers can be the fusion of classifiers which are appropriate (or even feasible) for discrimination of only two groups. Dichotomous classifiers can be merged in several ways, e.g. by the fusion of the classifiers which discriminate between all the pairs of groups or by joining the classifiers that distinguish between every group and all the remaining groups (for example, one-against-one ensemble or one-against-all ensemble, respectively, can be applied to expand Support Vector Machines binary classification to the multiclass case).

The recognition rate of the combination is usually better than that of each individual classifier [3]. This is particularly met when the ideas of construction of the

constituent classifiers are distinct (e.g. combining of linear discriminant function, classification tree and radial basis neural network) or different sources of diversity between classifiers can be employed, e.g. by resampling the training set (like in bagging, boosting or random subspace procedure) or by using different subsets of variables- disjoint or not disjoint (like in random subspace methods).

Duin and Tax [2] present three groups of combining of classifiers:

1. Parallel combining of classifiers –for different feature sets (parallel combiners are often of the same type).

2. Stacked combining of different combiners on the same future space (stacked classifiers are often of different nature, e.g. the nearest neighbor, Bayesian parametric discrimination and the neural network).

3. Combining of weak classifiers (large sets of simple classifiers: bootstrapping-bagging, boosting or random subspace methods).

Simple fusion (fixed) methods as minimum, maximum, mean, median, product and majority vote can be used as the parallel (for different, maybe disjoint, feature sets) and also as the stacked classifiers [2] (on the same future space). The results of the latter method, i.e. the stacked classifiers, will be examined. Combining of the multiple models by (weighted) voting, averaging, by median, product rule (product rule is equivalent to taking geometric mean of the base classifiers results) makes available considerable improvement in performance along with tighter confidence intervals. Another ensemble method is to treat the classifier outputs simply as the input of the second-level classifier and then the classical pattern recognition techniques for the second-level design can be used (e.g. linear, quadratic, kernel, logistic or nearest neighbor in second-level step). These methods are called trained combining classifiers [2]. In the work the trained combining with Bayesian discrimination in the second discrimination step was applied. However, the performance of the trained combining have proved worse than for the fixed combining, so the results of the trained combining are not in detail discussed in the paper.

Usually, the methods from the group of resampling of the dataset are applied to decision trees and neural networks, but they also perform well with other classification rules [5].

In this paper the bagging and the boosting ensembles were applied for e.g. for trees known as unstable (i.e. small changes in the training set lead to important changes in the classifier) and weak classifiers. A weak classifier means a classifier with the accuracy only slightly better than the chance (for example for two groups discrimination the lower limit of the accuracy level is equal to 0.5). To understand the idea of combining of the weak classifiers let's imagine the following example. If there is a big group of committee members, who "rather" do not make mistake (i.e. when the probability of good answer is bigger than 0.5) and we assume that they answer independently, a correct answer after voting their answers can be obtained with big probability. If p-common error rates of the constituent classifiers are assumed, the upper boundary of the probability that the majority vote

of L independent classifiers is incorrect, can be simply calculated from binomial distribution by:

$$\sum_{m=\lfloor (L+1)/2 \rfloor}^{L} \frac{L!}{m!(L-m)!} p^{m}(1-p)^{L-m}.$$

It can be derived from the formula that, after pooling as little as $L = 7$ independent classifiers, the increase of correct classification possibility is obtained even for weak base classifiers. For bigger number of the independent classifiers $L = 13$, each with classification error equal to 0.3, the voting ensemble classifiers error is equal to 0.06, so is significantly smaller. However, if the classifiers are not better than chance (it can be imagined that are not better than the classification of the pattern into groups randomly – with the probability proportional to the sizes of the groups), then merging of the constituent classifiers may fail to improve the performance. The weak classifiers are often unstable. Base trees are unstable methods. In the presented empirical examination bagging and boosting were applied for trees but also for neural networks (also known as unstable) and for logistic regressions.

The methods from the group of combining of classifiers work as follows: In bagging – Bootstrap AGGregatING [4], subsets from the training set are sampled, generating random independent bootstrap replicates. Next the classifier on each of these bootstrap samples is constructed and finally the constituent classifiers are aggregated by a simple majority vote. On the contrary, in the boosting method, the classifiers are constructed on the weighted versions of the training set, which are dependent on the previous classification results. In the next section those methods are described in details.

2.2. Combining of Classifiers Based on Resampling

Very important area of combining information is to apply classifiers built on the learning data subsets generated randomly. Succeeding loops in the construction of classifiers are defined by random subsets based on the same training set. These subsets may depend on the results of combining classifier performance achieved in the previous loops (like in the boosting ensemble) or may be independent (like in the bagging and the random subsets ensemble, e.g. the random forests methods). Next, the results of constituent classifiers can be merged in different ways**.** Traditionally, the constituent classifiers in bagging and boosting are of the same general form. For example, exclusively neural networks or decision trees can be such homogenous constituent classifiers. In such a case only the final parameter values differ among them due to their different sets of the training patterns. Besides the training samples' and variables' selection or the extractors' selection to reduce the dimensionality, one needs to make a decision on the number and types of classifier to be used and finally how to merge them.

2.2.1. Bagging (Bootstrap AGGregatING) Breiman [4]

Bagging attracted much attention, probably due to its implemental simplicity and popularity of the bootstrap methodology. Bootstrap sample is trained B times from the learning set with the possible replacements, where n is the size of training set. The classifier on each bootstrap data set is trained. The resulting classifiers are then combined, e.g. by the average of posterior probability or the unweighted majority vote. The process of generating the classifiers is parallel, so can be executed on different computers.

Bagging algorithm
Assume that we have a training set $(\boldsymbol{x}_i, z_i)$, $i = 1,\ldots,n$, of patterns $\boldsymbol{x}_i$ and class labels z_i.

1. For $b = 1,...,B$, do the following:
(b-the number of loop)
(a) Generate a bootstrap sample of size n by sampling with replacement from the training set; some patterns will be replicated, others will be omitted.
(b) Design a classifier, $K_b(\boldsymbol{x})$.
2. Classify a test pattern $\boldsymbol{x}$ by recording the class predicted by $K_b(\boldsymbol{x})$, $b = 1,...,B$, and assigning $\boldsymbol{x}$ to the class most represented.

2.2.2. Boosting Procedure

Increasing the performance of weak classifiers, called from this reason "boosting", is originated from Freund & Schapire [6] ARCing-Adaptive Resampling and Combining. In the boosting method the weights of misclassified cases are increased. This technique focuses on the informative or difficult patterns. In boosting, firstly a classifier with accuracy greater than average on the training set is created. Next, the boosting method adds new component classifiers to form the combined classifier. The joint decision rule of the ensemble has an arbitrarily high accuracy on the learning set. In this way, performance of the joint classifier is improved. Boosting is a deterministic procedure. It sequentially generates learning sets, where weights of misclassified cases are increasing, and as the effect it also generates the classifiers constructed on them.

The most popular boosting procedure is AdaBoost (Adaptive Boosting). This procedure allows the designer to continue adding weak learners until some desired low training error has been achieved. AdaBoost procedure used for weak classifiers can reduce the training error even exponentially if the number of the components is increased [1].

AdaBoost algorithm (Adaptive Boosting)

1. Initialize weights $w_i = 1/n$, $i = 1,...,n$.
2. For $t = 1,...,T$, (t – number of loop)

(a) construct a classifier $K_t(\boldsymbol{x})$ from the training data with the weights w_i, $i = 1,...,n$;
(b) calculate e_t error as the sum of the weights w_i corresponding of misclassified patterns;
(c) if $e_t > 0.5$ or $e_t = 0$ then terminate the procedure, otherwise set $w_i = w_i(1 - e_t)/e_t$ for the misclassified patterns and renormalize the weights so that they sum to unity.

3. For a two-class classifier, in which
$K_t(\boldsymbol{x}) = 1$ implies $\boldsymbol{x}$ is from population Π_1,
$K_t(\boldsymbol{x}) = -1$ implies $\boldsymbol{x}$ is from population Π_2
create the weighted sum of the classifiers,

$$K(x) = \sum_{t=1}^{T} \ln \frac{1-e_t}{e_t} K_t(x)$$

and assign $\boldsymbol{x}$ to the population Π_1 if $K(\boldsymbol{x}) > 0$.

Boosting is connected with the game theory and learning programming. The relationship between boosting and logistic regression also exists. The outcome base classifiers can be combined by the simple majority vote or the weighted version of the majority vote. In the case of the linear weighting of classifiers- the weighting is performed two times: firstly in the construction of data sets focusing on the erroneously classified patterns, and, secondly- in combining the obtained classifiers.

A weak classifier that is only slightly better than chance is the minimum requirement for bagging and boosting. For example, for a big training set, the easy linear Nearest Mean Classifier, after boosting, performs similarly to the efficient support vector machines [7] (SVM and boosting focus on difficult patterns) .

The boosting procedure can be applied in different ways. Effectiveness of three variants of the boosting classifiers, with aggressive, conservative and inverse changing weights, were examined by Kuncheva and Whitaker [8].

Bagging can be useful for critically small data sets [5], in unstable situations, while the theory of boosting is developed for weak classifiers built on large training sample sizes. For very large sample sizes, classifiers constructed on bootstrap replicates are similar and therefore bagging (or random forests) can be not beneficial. Thus, for stable classifiers (e.g. linear for big datasets), bagging has been regarded as useless. However, in contrast to the common opinion, Skurichina & Duin [5] demonstrated in respect to the linear discriminant function that usefulness of boosting does not depend directly on instability of the classifier. It depends rather on quality of the incorrectly classified objects (usually near the boundary between the discriminated classes) and on ability of the classifier to distinguish objects correctly. From the bias-variance point of view boosting reduces both the variance and the bias of the classifier, while bagging is the most efficient in reducing the variance.

If correct Bayesian classifiers are trained on distinct sets of variables, then a weighted product rule is the optimal combination scheme (e.g. the classifiers built on data from different sensors). Product combing is more sensitive to imperfections in the individual classifier, and a sum (or median) fixed combining is more reliable in practical situations.

In the presented paper effectiveness of the combining of multiple classifiers in aiding diagnosis of the thyroid disease was compared. Firstly, for simple, fixed combining of the constituent Bayesian discriminant classifiers, such as parametric linear discriminant method and also nonparametric kernel and nearest neighbor, were used (Table 1). For these discriminant functions, the operations of simple, fixed aggregation rules, such as minimum, maximum, average, median, sum, product and majority vote, were studied.

In turn, individual classifiers coming from the group with other idea i.e. ensemble based on generating the random subsets of training set were applied for ensemble methods. Namely, for the resampling methods, all classification trees, all logistic regression methods or all neural networks-linear multilayer perceptron procedures were applied.

Performance of the combined classifiers was assessed for comparison purposes by both resubstitution (i.e. apparent errors, errors on training set) and test sample errors.

A different character of the constituent classifiers is usually a reason for diversity of the ensemble. Diversity among the individual classifiers of the team is expected to be the important factor to improve effectiveness in the classifier combination. Diversity can be obtained by different ways: different character of the classifiers (eg. parametric and nonparametric; crisp and fuzzy), different parameters of the same type classifier, different subsets of variables and different subspace of observations obtained, eg. by the resampling methods. Ensemble of the classifiers merges the diverse classifiers, especially boosting imputes diversity by design.

3. Results and Discussion

For all studied ensemble methods, the combinations of different classifiers on the same feature set were examined, so stacked combining was applied. First of all, the simple fixed fusion methods and finally bagging and boosting were performed. Fixed methods can be considered as simple combining methods, because for decision fusion we do not need to model a joint distribution. The simple fusion procedure may be based on fixed combination rules like for example product or average; however, only under strict probabilistic conditions these rules are optimal. For example, the product combiner rule needs strict conditions for optimality- it is optimal under the conditional independence in the given class [1].

Firstly, only 3 constituent Bayesian classifiers were combined:

LDF linear discriminant function (with test error= 0.06),
KDF kernel discrimination with normal kernel, radius for kernel function $r = 0.5$ (test error = 0.0614) and
NN the nearest neighbor with $k = 7$ neighbors (test error = 0.0597).

The above parameters: radius r and number of neighbors k were chosen to obtain the smallest leave-one-out error. Combining makes usage of the different advantages of diverse classifiers. Averaging or other ways of joining of classifiers' results (e.g. voting) usually reduce effects of base classifiers' overtraining, so can be treated as some kind of regularization.

The results for combining those classifiers can be seen in Table 1 presenting the test-sample errors of classification. From the table we may conclude that performance of the combination rules of only three Bayesian discriminant methods examined are not greatly better than the best individual classifiers. It can be explained by the information of a very big sample size as compared to the number of variables (21) and also by the fact that the used constituent classifiers were chosen as the best in their classes. The lack of performance improvement can be considered also as coming from the following reason: the linear, the kernel and the nearest neighbor discriminations come from the same group of Bayesian discriminations, so the high diversity among the component classifiers is not hold. However, a multiple classifier system can significantly improve the performance, when the members in the system are not only different from each other, but if also the base classifiers are not the best in their classes.

Table 1. Test-sample classification errors of the stacked fixed combined classifiers (linear discriminant function LDF, kernel discriminant function KDF and nearest neighbor classifier NN)

Fixed combined method	Test errors
Maximum	0.032
Minimum	0.034
Majority Vote	0.058
Product	0.034
Median	0.075
Average	0.053

When a high diversity between two or more different data mining techniques exists, combining them often produces better classifications. The relationship between different combining accuracy and diversity of the classifiers' ensemble was studied on data generated by Kucheva &Whitaker [8] and Shipp et al. [9]. Ćwiklińska-Jurkowska et al. [10] studied this relationship of ensemble methods' performance with diversity for the large real-life medical data of thyroid examined in this paper.

It is interesting to compare the performance of such fixed methods (Table 1) with more time-consuming resampling methods, such as bagging and boosting. Usually bagging, boosting (and random subspace method) are applied to decision trees, where they often produce an ensemble of classifiers, which is superior

to a single classification tree rule. In the present study, the bagging and boosting ensembles applied with the following discriminant methods: classification trees, neural networks multilayer perceptron with 3 hidden units and logistic regression are summarized in Table 2. Different relationships are represented by each of these applied base classifiers. For example, logistic regression assumes a linear relation between the explaining variables and the target, while neural networks suppose a nonlinear relation, which is discovered depending on the architecture and the activation functions. In turn, decision trees assess constant values within rectangular or cuboid regions of the input space.

Classification test errors obtained for the resampling methods with the unstable classification trees were smaller than for the simple fixed combining of three Bayesian classifiers (LDF, KDF and nearest neighbor). Additionally, the test errors of bagging and boosting of neural networks, which are also unstable methods, are also lower, however, only for some fixed fusion methods like median, average and majority vote in combining three Bayesian classifiers (Tables 1, 2).

From Table 2 we can also make the comparison of the bagging and boosting performance for the constituent classifiers such as logistic regression, neural networks and classification trees.

The difference between the bagging and the boosting of neural networks performance is not evident and additionally is not visible between the bagging and the boosting of trees errors (Table 2). It can be explained by the big size of the data sample in comparison to dimensionality (21 features), because the difference between the bagging and the boosting performance for the unstable classifiers is especially clear for small datasets.

Only for the stable logistic regression classifiers, the difference between bagging and boosting is observed. For the logistic regression the results of bagging are better than the boosting outcomes- the difference of performance measured by the test-sample error is on the level from 0.016 to 0.026.

Bagging and boosting of logistic regression procedures give worse performance than bagging and boosting of unstable methods: trees and neural networks. Bagging and boosting of logistic regression are also worse than for almost all simple fixed rules, i.e. for minimum, minimum, majority voting, product and average of three posterior probabilities coming from the considered Bayesian classifiers (Tables 1, 2).

Table 2. Test-sample and training classification errors of bagging and boosting classifier on thyroid data set

Resampling method	Train error	Test error
Boosting logistic regressions (100, 150 loops)	0.05-0.06	0.07-0.08
Bagging logistic regression (100, 150 loops)	0.045	0.054
Boosting neural networks (10 loops)	0.0267	0.0367
Bagging neural networks (10,20,30,40,50,60 loops)	0.028-0.034	0.035-0.04
Boosting trees (10,15,20,50, 100 loops)	0	0.01
Bagging trees (10,15,20,50, 100 loops)	0	0.01

The boosting classification trees errors (both resubstitution error and test-sample error) for loops from 1 to 20 are presented on Fig. 1, as well for the base classifiers as for the ensemble classifier. On this plot the train and test-sample errors of the succeeding loops are contrasted to the ensemble classifier errors (described in the caption). Analyzing Fig. 1, the very good performance for the combined decision trees after the process of boosting with 20 loops can be noted (the level lines denoted in the caption as "Boosting Train error" and "Boosting Test error"). Because the boosting method is focused on difficult patterns, the training error of each successive component tree classifier (the line " Succeeding loops train error") for number of loop from second to tenth is usually larger than for the previous classifiers in the loop (which represent points lying on the left side to the current loop point). At the same time the train and test ensemble error (the levels of two errors are denoted by the lines "Boosting Train error" and "Boosting Test error") is considerably smaller than the single component tree errors (and the mean errors) at the beginning first 10 loops of the procedure. For the test errors of single tree classifiers (the line "Succeeding loops test error") increasing tendency is also visible. However, the magnitude of test error is much smaller, not greater than 0.02, while the train error of succeeding loops reaches even 0.08. After first ten loops the train errors rapidly diminish, the decrease of test errors is also important.

Comparison of the "Boosting Train error" level with the averaged values of the line "Succeeding loops train error" and comparing the "Boosting Test error" level with the averaged values of the line " Succeeding loops test error" visualize the following facts. The train and test errors of the combined methods are significantly

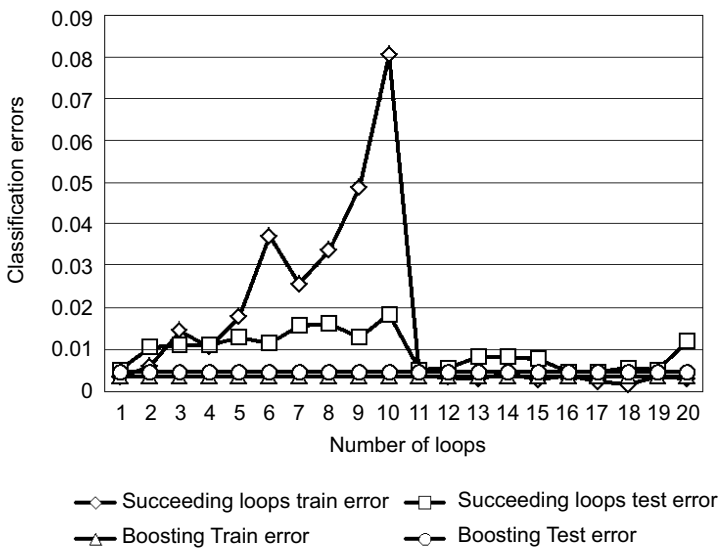

Fig. 1. Component classifiers and combined methods errors for the boosting procedure of 20 classification trees

smaller than the corresponding averaged constituent classifiers errors from the loops, for both bagging and boosting. This benefit of the combining can be explained by the fact that when the component classifier performs better than chance (met in the performed analysis of the examined dataset), the weighted decision ensures that the training error will be smaller than for the constituent classifiers [11].

Figure 2, constructed similarly to Fig. 1, contains the component classifiers and combined methods' errors for the boosting procedure, however for 25 base logistic regression discriminations. Though for 25 loops of the boosting logistic regression functions, the train and the test errors of the successive loops do not have such tendency of diminishing after some number of loops (the lines "Succeeding loops train error" and the "Succeeding loops test error" on Fig. 2), the combining method has again the ensemble error significantly smaller than the average of all loops errors. It holds both for the test and train error.

Summarizing, the averaged train and test-sample errors of the succeeding loops are bigger than the ensemble classifier errors for boosting as well classification trees and for boosting 25 logistic regressions. The very important stage in the construction of the ensemble classifier feature for the model is the selection of the number of loops which can be considered as a regularizing parameter. For this reason plots of the type similar to Figs 1 and 2, in the form of learning curves (dependency of the errors on number of loops) may be useful in the selection of ensemble.

Relationship between the train-sample and test-sample errors of the constituent classifiers in all 150 loops of boosting of logistic regressions can be assessed in

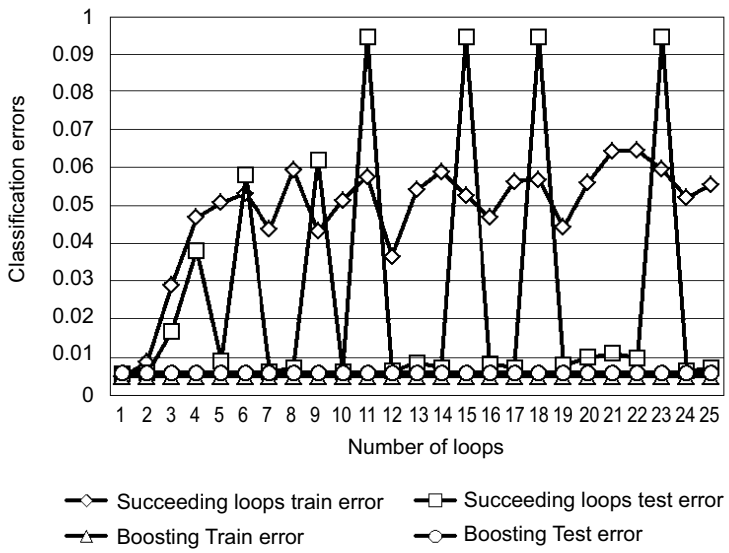

Fig. 2. Boosting of 25 logistic regressions. The train and test errors of succeeding loops compared to the ensemble classifier errors

Fig. 3, where the different loops are represented as different points. Spearman correlation between these errors is equal to -0.34. Thus, generally smaller classification errors assessments for new observations (which are independent on training set), measured by the test sample error, is not obtained for the very small training errors. Boosting is focused on difficult patterns, what can consequently cause the high train errors for sequential constituent classifiers. Classifiers with the high train errors are repesented by points mostly located in the right-bottom part of Fig.3. The number of component logistic classifiers with train errors over 0.4 is much higher than the number of classifiers with test errors over 0.4. More than half of the base classifiers test error are below 0.1, while only a few learning errors of the base classifiers are below 0.1.

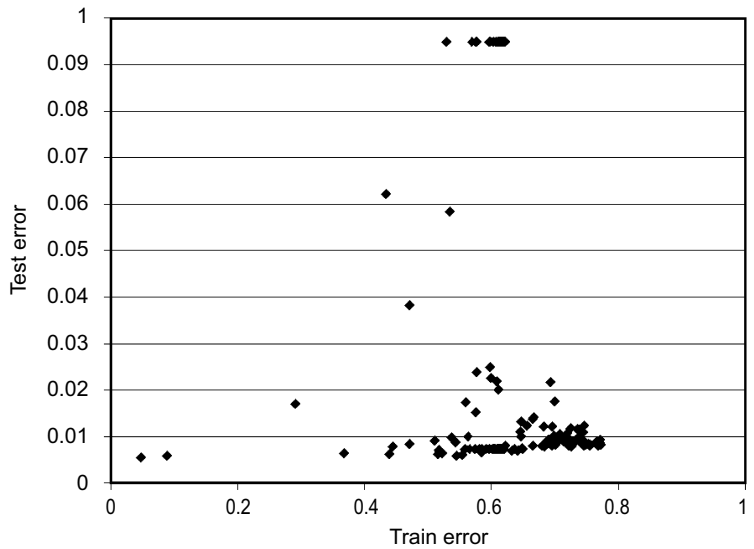

Fig. 3. Relationship between the train and test errors of the constituent classifiers in boosting of 150 logistic regressions

The best performance among the examined ensemble methods was obtained for both bagging and boosting of classification trees. For the resampling methods (for both bagging and boosting ones) better pooled classifiers are obtained when build on the trees than the ensembles constructed on the logistic regressions (Table 2). Also results of the trees combined classifiers are better than for the simple fixed combining of three Bayesian classifiers, while the resampling of the logistic regression is not considerably more efficient, than the fixed combining of three Bayesian classifiers (compare Table 1 and Table 2). For both resampling methods, bagging and boosting, the smaller train errors and test errors were obtained for the base neural networks, than for the logistic regression classifiers.

Skurichina and Duin [5] report that for linear classifiers, bagging may improve the performance on classifiers constructed on a critical training data set (when size of the learning set is comparable with the dimensionality, when even the linear classifier can be unstable); however, when the base classifier is stable (e.g. the nearest mean classifier is typically stable), the bagging procedure is useless, will lead to a little improvement. In the data set, presented in the current paper, the dimensionality is not smaller than the number of observations and the logistic regressions are not unstable. From the results obtained in the presented study it can be concluded that bagging is particularly useful for the unstable methods: neural network and classification trees and is useless for the stable logistic discrimination.

Bagging and boosting of trees and neural networks are more time-consuming methods, so they may perform better than fixed combining of three Bayesian methods. These resampling procedures can use different functions as a base classifier, but are especially useful for unstable classifiers and they make efficient usage of the data.

In the combined classifiers methods, each classifier may obtain somewhat diverse subsets of the train data or parameters. Bagging and boosting are connected with the resampling the original dataset, so in this way diversity is forced on the base classifiers and diversity may be useful for performance improving [8–10].

Despite the performance benefits coming from the bagging and boosting procedure, a drawback should also be mentioned in the context of supporting the diagnosis. The discomfort for the resampling methods comes from the fact that for both bagging and boosting: physician has no simple interpretation, like for single tree or else easy method, e.g. linear discrimination leading to score diagnosis system.

4. Concluding Remarks

Bagging and boosting are more time- consuming methods than the fixed combining methods and the resampling methods as bagging or boosting perform better than the fixed combining methods. Bagging and boosting are connected with resampling the original dataset, so the useful diversity is forced on the base classifiers. Thus the improvement of the performance is obtained. Better improvement by using these resampling methods was obtained for the unstable methods such as classification trees and neural networks, where the resampling methods make efficient usage of the data. The lack of significant difference of performances between bagging and boosting may be explained by the big sample size in comparison to its dimensionality.

The smallest test classification errors for the bagging and boosting method of the classification trees were obtained. The excellent results were obtained after only 10 loops. The number of loops is a very important feature of the resampling method and can be considered as some kind regularization parameter.

References

1. Webb A. R.: Statistical pattern recognition, John Wiley & Sons, Ltd, Chichester 2003. DOI: 10.1002/0470854774.ch1
2. Duin R.P.W., Tax D.M.J.: Experiments with Classifier Combining Rules; in: Multiple Classifier Systems. Kittler J., Roli F. (Eds.), Berlin 2000, Springer-Verlag.
3. Yu K. Jiang X. Bunke H.: Lipreading: A classifier combination approach. Pattern Recognition Letters 1997, 18, 1421–1426.
4. Breiman L.: Bagging predictions. Machine Learning 1996, 24 (2), 123–140.
5. Skurichina M., Duin R.: Boosting in Linear Discriminant Analysis; in: Multiple Classifier Systems. Lecture Notes in Computer Science 2000, 190–199, DOI: 10.1007/3-540-45014-9_18.
6. Freund Y., Schapire R.: Experiments with a new boosting algorithm. In machine learning. Proc.13th Intern. Conf., Morgan Kaufmann, San Francisco 1996, 148–156.
7. Skurichina M., Kuncheva L., Duin R.: Bagging and boosting for the nearest mean classifier: Effects of sample size on diversity and accuracy; in: Multiple Classifier Systems. Lecture Notes in Computer Science 2002, 2364, 307–311. DOI: 10.1007/3-540-45428-4_6.
8. Kuncheva L.I., Whitaker Ch.: Using Diversity with Three Variants of Boosting: Aggressive, Conservative and Inverse. In: Multiple Classifier Systems. Lecture Notes in Computer Science, 2002, 2364, 717–720.
9. Shipp C.A., Kuncheva L.I.: Relationships between combination methods and measures of diversity in combining classifiers. Information Fusion 2002, 3 (2), 135–148.
10. Ćwiklińska-Jurkowska M., Jurkowski P.: Effectiveness in Ensemble of Classifiers and their Diversity on Big Medical Data Set. Computational Statistics. COMPSTAT 2004, 855–862.
11. Duda R., Hart P., Stork D.: Pattern classification. Wiley, New York 2001.