

Bagging-based instance selection for instance-based classification

Dmytro Kavrin¹[0000-0002-8952-4067], Sergey Subbotin²[0000-0001-5814-8268],

^{1,2}National University "Zaporizhzhia polytechnic", Zhukovsky str., 64,
Zaporizhzhia, 69063, Ukraine

¹kavrin@zntu.edu.ua, ²subbotin@zntu.edu.ua

Abstract. The task of reducing marked large size samples for building diagnostic and recognizing models by precedents is considered. The method allowing to reduce essentially the size of a training sample increasing at the same time its efficiency, through removal of irrelevant and redundant instances is proposed. The given method provides an opportunity to estimate each instance of a training sample by synthesis of an ensemble of weak classifiers using a bagging model, and to create a reduced sample of the most significant instances by estimations. Software is developed to implement the proposed method. This software has been experimentally investigated in solving the task of reducing synthetic and real world data. The results of the conducted experiments allow recommending the use of the developed method and its software realization for solving the task in the sphere of technical diagnostics.

Keywords: base classifier, class, classification, ensemble of classifiers, instance, meta-estimator, metric, sampling, training sample.

1 Introduction

The constantly increasing volume of available information requires significant computing resources for successful data processing. So the task of reducing dimension data for further building models based on them is urgent [1-4]. It is especially necessary when solving practical tasks of industrial diagnostics, when the diagnostic system should immediately respond to any deviations in the operation of equipment. In the conditions of continuous production, diagnostic systems of nondestructive testing, which allow making diagnostics in real time, are especially important. The main component of nondestructive testing diagnostic systems is the model of pattern classification by precedents (classifier) [5-8]. The classifier is a set of rules, which allow determining whether new observations (instances) belong to one of the existing classes. To generate a model of recognition by precedents it is necessary to have a set of instances (precedents) with known values of classes (training sample), and a classification method, which will form the rules of recognition (training) using the training sample [9-10].

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

There are many classification methods with different principles and approaches [11-12]. The family of metric classification methods based on precedents is quite effectively used in building diagnostic models. Metric methods of training refer to the geometrical paradigm of machine learning, which assumes that instances have a geometric structure, where each of them is described by numerical features and is considered as a point in a multidimensional feature space. These methods are based on the assumption of local compactness of classes, from which it follows that the similarity of two instances on N independent features also assumes their similarity on one dependent feature $N+1$. Thus, only instances of the same class can exist in the neighborhood of one class instances. And the closer a control instance is to the neighborhood of a class, the more likely it will belong to this class [13-14]. Metric methods have such advantages as simplicity of implementation, clear logic of methods' work, geometrical nature, simplicity of model results interpretation, developed theoretical base, adaptation to the necessary task by metric selection. The disadvantage of the metric methods recognition is the necessity to store the whole training sample in the computer memory. Thus, the use of large size training samples may require significant computing resources and classification time. In technical diagnostics systems, where the speed of decision making is a priority, models based on large training samples may be ineffective. Reducing the size of training samples will help reduce both the classification time and the computational complexity of diagnostic models.

The most widely used approach to reducing the data dimension is the selection of informative features [15-17], which implies the selection from the initial set of features a smaller subset of features, sufficient to solve the problem with the required accuracy or meeting some criterion. However, if the size of the feature space is small, or features individually are low informative, but together they contain enough information for building a model, selection of informative features does not allow to reduce the data dimension effectively.

Another approach to reducing data dimensional is the selection of instances [1, 18-22]. To date, the instance selection has been considered a necessary preliminary data processing procedure [1, 23]. Successful use of instance selection methods allows for the selection of a small sample size, independently of the model in which it will be used in a future and without performance loss [2]. In the process of selecting instances, irrelevant and redundant instances are removed from the sample, so in some cases the performance of models that are built on processed samples may be higher than on the initial data.

Many instance selection methods have been proposed in the past few decades, each with weaknesses and advantages [23-26]. However, there is no universal method which can achieve equally high results with various data samples. In general, the task of instance selection is to select the most relevant instances of a training sample. Virtually, for the selection of instances must be solved the problem of binary classification where each instance of the training sample can be classified as selected or unselected. Therefore, it can be assumed that the approaches and methods used to solve the classification problem can also be applied to the instance selection task.

One of the successful directions for increase of models productivity in the tasks of classification is the use ensembles of classifiers [27-29].

In this paper we study the possibility of using ensembles of metric classifiers to select the most relevant instances in solving the problem of reducing training samples and increasing their representativeness.

2 Formal problem statement

The task of instance selection for classification model building based on precedents is to generate such a training subsample of minimum size from the most relevant instances of the initial training sample, which will allow to classify new unmarked data with an accuracy at least as accurate as using the initial training sample [2, 30]. Formally, the task can be written in the following way.

Let the initial training sample be presented as a set of S precedents of dependence $y(x)$ and be defined by the expression $X = \langle x, y \rangle$ is a sign matrix of input features x and output features y . The set of input attributes x is defined by the standard object – feature matrix:

$$x = \left\| x_{ij} \right\|_{S \times N}, \quad (1)$$

where S is a number of instances, N is a number of input features, x_{ij} is a value of the j -th feature on the i -th instance. The set of output features is defined by the vector:

$$y = (y_1, \dots, y_S)^T, y_i \in \{1, 2, \dots, K\}, \quad (2)$$

where K is a number of classes in the sample ($K > 1$). Each i -th instance is represented as $X_i = \langle x_i, y_i \rangle$. Then the task of selecting instances is to select from the initial sample $X = \langle x, y \rangle$ such a subsample $X' = \langle x', y' \rangle$ that the following conditions are executed:

$$x' \in x, y' = \{y_i \mid x_i \in x'\}, S' = |y'|, S' < S, f(\langle x', y' \rangle, \langle x, y \rangle) \rightarrow opt, \quad (3)$$

where S' is a number of instances of the resulting subsample, x' is a set of input features of the resulting subsample, y' is a set of output features of the resulting subsample.

3 Review of the literature

At solving the complex practical problems of pattern recognition by precedents, there may be occasions when no classifier provides the necessary accuracy of pattern recognition (classes). Increase in accuracy allows creating a model composed of a set of classifiers (ensemble). The strategy of ensembles is that a set of independent models

of classification (base classifiers) is created, and the results of their work are combined. Thus, the productivity of base classifiers increases due to compensation of errors of some classifiers by correct work of other ones [27-29]. Generally, the ensemble of classifiers can be described as follows:

$$y(x) = \Omega(b_1(x), \dots, b_r(x)), \quad (4)$$

where b_i is a base classifier, Ω is a meta-estimator that creates a decisive rule that the recognized instance belongs to a certain class $y(x)$.

The basic properties of classifiers are the ability of each base classifier to independently solve the initial classification task and the possibility to use existing classification training standard methods.

The one of the most important conditions for the efficiency of the classifier ensembles in pattern recognition problems is the requirement for a sufficient variety of base classifiers [27-28]. Thus there is a compensation of classification errors of some base classifiers by work of other ones. Therefore, it is necessary to combine the results of the base classifiers, so as to increase the influence of true decisions and minimize the influence of wrong decisions on the response of the ensemble. The basic strategies for building ensembles are the synthesis of independent classifiers and decision-making on a bagging basis [31-33], special coding of target values and reduction of the task solution to solving several tasks (error-correcting output code) [34], building of meta-signatures on the basis of responses of base classifiers on subsets of samples and training meta-functions on them (stacking) [28, 35], sequential summarization of several classifiers, with each next classifier being trained taking into account the errors of the previous ones (boosting) [27-29, 32-33, 36-37], heuristic methods of combining the answers of base classifiers by training in special subspaces and visualizations (mixture-of-experts) [28, 38-39], recursive synthesis of homogeneous ensembles (neural networks) [40].

Bagging-based ensembles are the most common when it comes to solving real world tasks due to their simplicity of implementation and high generalization ability. The main advantages of bagging are the ability to perform parallel computations at high classification accuracy.

During ensemble forming by the bagging method, each base classifier is trained on a random subset of the training sample. With this approach, the variety of methods is achieved even with the use of one classification method for all base classifiers [27, 33, 38]. In the traditional bagging model, the bootstrap technique [27, 29, 31, 33, 41] is used to select random subsets of a training sample, which implies the formation of subsets by random selection with return. The basic bagging method works well on small samples, when exclusion of a small number of instances leads to a significant distribution transformation in the sample. For larger training samples it is possible to use other sampling methods. In this case, the task of selecting the optimal size of the selected subsamples S .

There are many different ways to extract subsets (resampling) of a training sample for the synthesis of bagging ensembles [11, 31, 41-46]. The most known methods are

bootstrap [31, 41-46], pasting [47], random subspaces [28-29], random patches [48] and cross-validation [41].

The base classifiers forming an ensemble using a bagging strategy do not need a separate test subsample to evaluate the accuracy of the generated model. It is possible because each base classifier is trained on a subsample containing only part of the training sample. Thus, it allows to estimate the accuracy of each base classifier with instances not included in the selected subsample. Provided that the number of base classifiers is quite numerous, the evaluation will be performed for almost every instance of the training sample. Moreover, this evaluation will be independent, because the accuracy of each base classifier will be evaluated by a subsample of instances with the dependent variable values unknown to it. Further training and test samples of base classifiers will be called local for certainty.

By constructing ensembles of classifiers, it is important to take into account that there are classification methods stable with respect to the selection of random subsets (for example, the SVM (support-vector machine) method, or the kNN (k-nearest neighbors) method at $k > 3$) [27, 49]. Application of such methods in ensembles based on bagging is ineffective because diversity of methods is not achieved.

The result of ensemble work depends on the choice of meta-estimator. In the most common case, the meta-estimator is a majority vote function for the classification task:

$$y(x) = \Omega(b_1(x), \dots, b_T(x)) = \arg \max_{k=1}^K \sum_{t=1}^T b_t^k. \quad (5)$$

In more complex cases, weighted voting can be used, where each base classifier has a weighting characteristic:

$$y(x) = \Omega(b_1(x), \dots, b_T(x)) = \arg \max_{k=1}^K \sum_{t=1}^T w_t b_t^k, \quad (6)$$

where w_t is a weight characteristic of the base classifier. The weighting characteristic primarily depends on the accuracy parameter of new instances recognition by the classifier. Such characteristic can be a relative number of correctly recognized instances of the test sample (relative accuracy):

$$E = \frac{1}{S^b} \sum_{i=1}^{S^b} [1 | b(x_i) = y_i], \quad (7)$$

where E is a relative accuracy of the classifier, S^b is a number of instances in the local test sample of the current classifier, b is an approximating function of the classifier.

Another important parameter depends on the number of instances in the selected training sample. According to the size minimization task, in order to calculate the weighting characteristics of each base classifier it is possible to combine the charac-

teristics of classification accuracy within the local test sample and the share of instances reduced from the initial training sample:

$$w = \gamma E + (1 - \gamma)r, \quad (8)$$

where $\gamma \in [0..1]$ is a coefficient indicating the degree to which factors affect the value of the overall score, r is a share of reduced instances $r = S^b/S$, since the local test sample is formed from instances not included in the local training sample.

The relative accuracy of classification (7) gives an objective assessment of the classifier provided it is sufficiently balanced by classes the test sample. If the test sample has an imbalance of classes, for example, a minority class is 1% of the sample, it is possible that a classifier that incorrectly classified all minority instances and correctly classified the majority class will have abnormally high relative accuracy ($E = 99\%$). Selection of instances, when constructing a bagging-ensemble, is carried out randomly, so it is impossible to guarantee the balancing of training and test local samples. Stratified instances selection for the local training sample can be one solution to the problem, but if the initial sample is imbalanced by classes, the local test sample will also have class imbalances. Therefore, for imbalanced samples, an assessment based on a confusion matrix may be more appropriate [50]. The confusion matrix is a way of grouping the instances depending on the combination of the true answer and the classifier's answer and allows to get a set of different metrics. In case of binary classification, instances can be divided into four categories (Table 1).

Table 1. – Confusion matrix

	$y = 1$	$y = 0$
$b(x) = 1$	True Positive (TP)	False Positive (FP)
$b(x) = 0$	False negative (FN)	True Negative (TN)

The instances of the class of greater interest are called positive instances and another class is called negative. When dealing with imbalanced data, the minority class is usually presented as positive. Using the confusion matrix, it is possible to obtain precision and recall metrics. The precision:

$$P = TP/(TP + FP), \quad (9)$$

where TP is the correctly classified positive instances, FP is a incorrectly classified positive instances, shows the share of correctly predicted positive instances. The recall:

$$R = TP/(TP + FN), \quad (10)$$

where FN is the incorrectly classified negative instances, shows the share of correctly predicted positive instances of all predicted instances as positive instances. Obviously, the higher the values of these metrics, then the classifier is better. However, it is impossible in practice to reach the maximum values of precision and recall simulta-

neously, so it is necessary to choose which characteristic is more important for a particular task, or to search for balance between these values. The harmonic mean of the precision and recall (F-measure) allows combining these parameters [51]:

$$F = 2PR/(P + R). \quad (11)$$

4 Materials and methods

To select the most relevant instances of a training sample using an ensemble of classifiers based on a bagging model, you need to solve the problem of binary classification for each instance of the training sample. In addition, each instance will be assigned to the selected class of instances, or to the class of instances that are not meeting the selection condition.

The classification of instances from the initial sample is based on the voting results of base classifiers. The base classifier is a model trained on the marked sample, in which the value of the output feature $y(x)$ for each instance is known. To synthesize the base classifier of a bagging ensemble, a random subset of instances is selected from the initial sample by the bootstrap method. The resulting local sample is used to train the base classifier. The nearest neighbor kNN method with one nearest neighbor and Manhattan distance metric was chosen as training method [52]. This choice is conditioned by the necessity of obtaining less stable classifiers and increasing the rate of synthesis of a large number of base classifiers. This model uses a passive learning strategy, in which there is no learning phase of the classifier, instead, the learning sample is stored in memory, which is used to classify new data. The main advantage of this model is the ability to use new data without retraining, simply adding new significant instances to the sample. However, in such a model, large size training samples will require significant memory resources for storage. Using a basic bootstrap method for building an ensemble of classifiers implies retrieving subsamples of the same size as the initial sample. Thus, each ensemble classifier must store almost the entire training sample in memory. Since this research is aimed at reducing the size of large training samples, extraction of random subsamples was performed by random selection with return, but in contrast to bootstrap, the length of the subsamples was randomly determined in a predefined range. When creating each classifier, the unselected instances were used as a local test sample for the particular classifier estimate. The weighting parameter w of each base classifier was obtained using the following equation (8).

The primary aim of the study was to select the most representative data from the training sample, so the task of the selection method is to investigate each instance of the sample and assess its relevance. Random selection methods do not guarantee the examination of each instance, so at the preliminary stage of creating an ensemble it is proposed to divide the initial sample into some number approximately equal in size subsamples and then to classify each subsample using ensembles built on the remaining training subset. With this approach, it is possible to ensure that every instance of the initial training sample is examined. The number of subsamples can take the value

$M > 1$, increasing the number of subsamples will lead to more stable and accurate results, but on the other hand will increase the computational complexity of the model and the processing time.

Formally, the proposed instance selection method can be presented as follows:

1. Set the initial training sample $X = \langle x, y \rangle$, initialize the resulting sample $X' = \langle x', y' \rangle = \emptyset$. Set the number of subsamples $M > 1$. Set the number of base classifiers T . Set the value of the coefficient $\gamma \in [0...1]$. Set the threshold $\delta \in [0...1]$, instance selection in a new training sample.
2. Split the initial sample X into M subsamples of approximately equal size:

$$X = \{X^m = \langle x^m, y^m \rangle, m = 1...M\}. \quad (12)$$

3. Set the number of subsamples $m = 1$.
4. Set the number of base classifiers $t = 1$.
5. Using a method of simple random selection with return, take a local training sample X^\wedge from the subsample $X \setminus X^m$. To define as a local test sample X^\top the set of instances not selected in the sample X^\wedge .
6. Train the base classifier using the local training sample $b = fit(X^\wedge)$.
7. Calculate the harmonic mean value F (11) for the current base classifier. If $F < 0.5$, go to step 5.
8. Calculate the weight characteristic of the current base classifier w (8).
9. Classify the subsample X^m and calculate the weight of each instance, taking into account the weight characteristic of the base classifier :

$$\chi_t = \{w[b(x_i^m) = y_i^m]\}_{i=1}^{s^m}. \quad (13)$$

10. Set $t = t + 1$. If $t \leq T$, go to step 5.
11. For each instance of a subsample, calculate the value of the meta-estimator:

$$\Omega_i^m = \left\{ \sum_{t=1}^T \chi_t \right\}_{i=1}^{s^m}. \quad (14)$$

12. Set $m = m + 1$. If $m \leq M$, go to step 4.
13. Merge M vectors of meta-estimators and normalize the values to a unit segment:

$$\left\{ \Omega_i = \frac{\Omega_i}{\arg \max_{\{\Omega_j\}_{j=1}^S} \Omega_j} \right\}_{i=1}^S. \quad (15)$$

14. Form a new training sample:

$$X' = \langle \langle x_i, y_i \rangle | \Omega_i > \delta \rangle_{i=1}^S. \quad (16)$$

5 Experiments and Results

To obtain a summary evaluation of the method, the experiments were conducted on two different samples, which differed in the number of instances, features and classes (Table 2). To evaluate the obtained training samples, at the first stage of the experiment, the initial data set X^0 was divided by the stratification method [53] into training X and validation X^v samples in a ratio of 75/25. The training sample obtained by stratification method was later considered as the initial sample. Classifiers were built on the basis of the initial and the resulting samples and tested with the validation sample. Then values of relative accuracy of classification and number of instances of samples were compared. The nearest neighbor method with the Euclidean distance metric was used as a method of classifying the recognition model. The resulting sample was formed using an ensemble of base classifiers. All base classifiers used the nearest neighbor method with one neighbor and Manhattan distance metric. Using the Manhattan distance metric reduced the stability of the base classifiers and computational complexity. A variety of base classifiers was achieved through the use of simple random selection with return when forming a local training sample of each base classifier.

Table 2. - Experimental data sets

Dataset	S^0	S	S^v	N	K
Pulsars Recognition (Pulsar) [54]	17 898	13 424	4 474	8	2
Sample Classification (Banana) [55]	5 300	3 975	1 325	2	2

To guarantee the evaluation of each instance, the training sample was randomly divided into two subsamples with approximately equal numbers of instances. An ensemble of base classifiers was built upon each such subsample. Figure 1 shows the scheme of work of the instance selection method based on an ensemble of classifiers.

The number of basic classifiers varied in the interval $T \in [1...200]$. The instance selection for the local training sample of each base classifier was performed by simple random selection with return. The local sample size was determined randomly in the range of $[1\%...100\%]$ of the initial subsample size. For the base classifier the F-measure value F was calculated using the local test sample, consisting of instances not included in the local training sample. Moreover, if the F-measure value F of the base classifier was less than 50%, the synthesis procedure for this classifier was repeated. Then the weight of each base classifier was calculated according to the equation (8) with parameter $\gamma = 0,75$. The second subsample was classified by each base classifier according to its weight. Thus, a vector of weights corresponding to all the base classifiers was formed for each instance of the initial sample. The resulting training sample X' was formed from the instances having the maximum total weight.

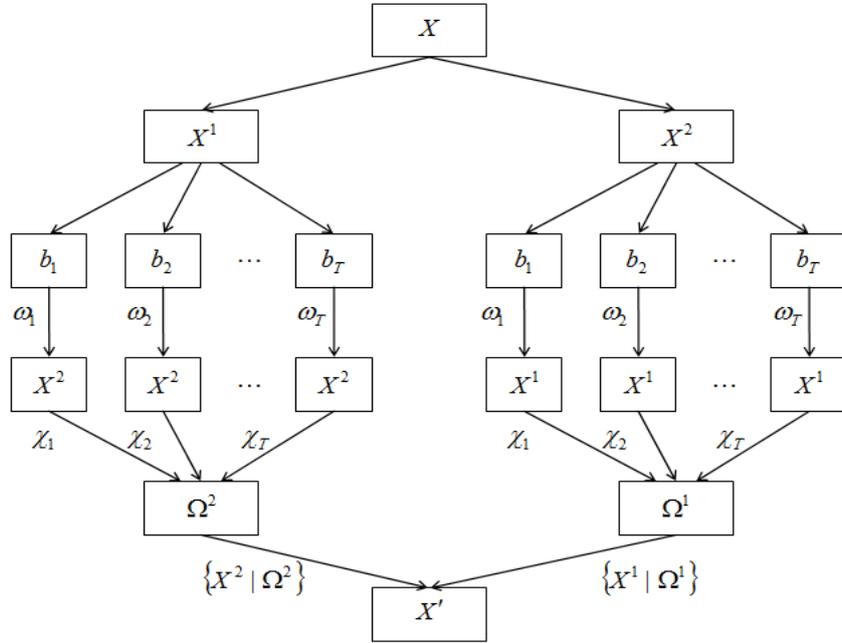


Fig. 1. – Scheme of work of the instance selection method based on an ensemble of classifiers

At the next stage of the study, classification of the validation sample X^v by the method of the nearest neighbor with one nearest neighbor and the Euclidean distance metric was performed. The initial sample X and the resulting sample X' were used as training samples. The relative accuracy of the models was calculated according to the equation (7). Using the obtained data, the dependencies of relative accuracy and sample length on the number of base classifiers were plotted (Fig. 2-5).

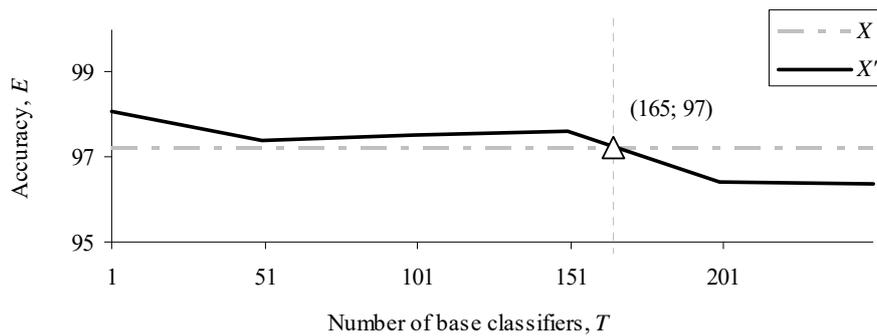


Fig. 2. – Dependence of model accuracy on the number of base classifiers for the Pulsar dataset

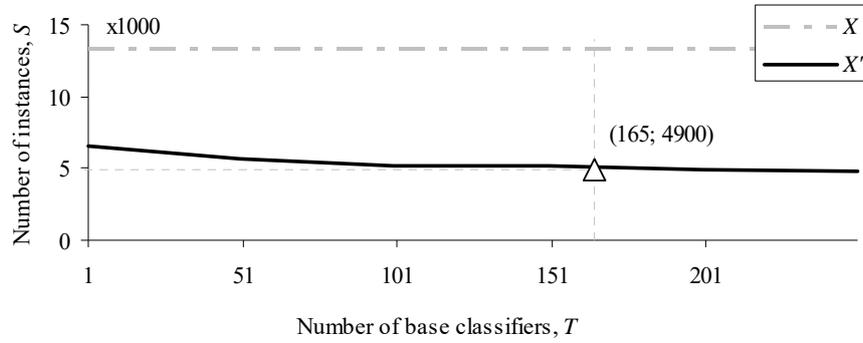


Fig. 3. – Dependence the number of instances on the number of base classifiers for the Pulsar dataset

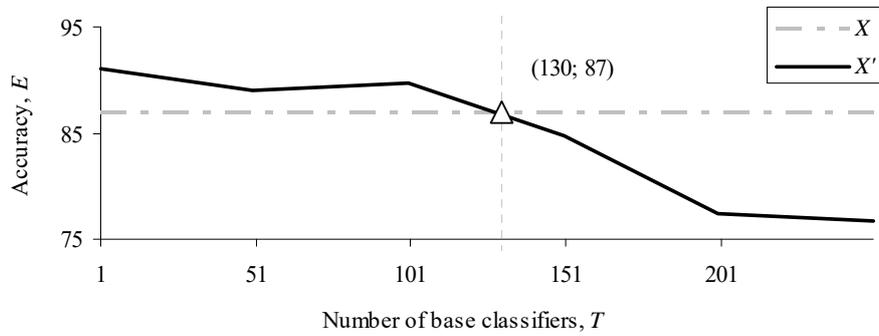


Fig. 4. – Dependence of model accuracy on the number of base classifiers for the Banana dataset

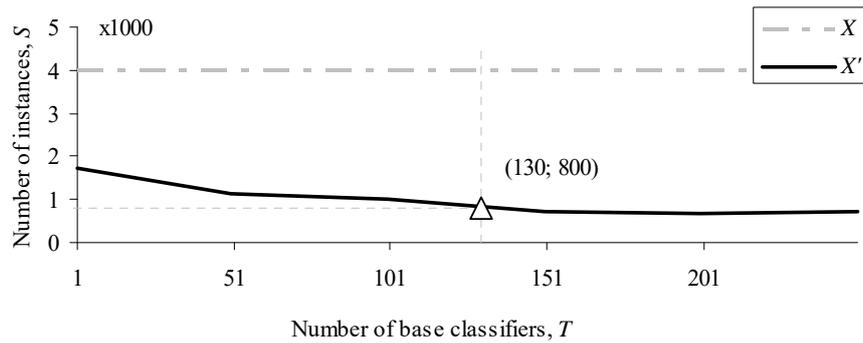


Fig. 5. – Dependence the number of instances on the number of base classifiers for the Banana dataset

For clarity, the figures showed the local areas within the critical values of model accuracy, at which the accuracy of classification of the resulting sample became less than the accuracy of the initial training sample. Thus, it was possible to estimate the critical number of instances of the resulting sample, below which the relative accuracy of the model based on the resulting sample became lower than the relative accuracy of the initial sample model.

6 Discussion

The proposed method showed high efficiency on all investigated datasets. All models on the basis of obtained training samples had relative accuracy higher than models with initial samples. At the same time, the size of the obtained samples was less than two times than the initial samples, even with a minimum number of classifiers. The increase in the number of base classifiers led to a decrease in the size of the resulting sample and a decrease in the relative accuracy of the model. Such results are due to the fact that with the increase in the number of base classifiers decreased the number of instances, the total weight of which reached the value of a given threshold, and probably removed significant instances, which reduced the effectiveness of the model. Despite the disadvantages of the proposed method, in practical application there is a range of the number of classifiers, in which the relative accuracy of the method will be higher, the classifier based on the initial sample. Also, the proposed method of instance selection requires a quite large number of initial parameters. Therefore, there is a need to create a method capable of independently estimating the initial parameters of the model. For this purpose, it is necessary to develop mechanisms of model evaluation and determination of the initial parameters of the method.

7 Conclusions

The task of reducing marked data samples of large size for building diagnostic and recognizing models by precedents is considered. The results of the experiments have shown the efficiency of the proposed method on all investigated samples.

The scientific novelty of the obtained results is that a new method has been created, reducing the size of the marked samples, saving the most significant instances and removing the less informative ones. Thus, the proposed method allows to solve the data reduction problem, increasing the efficiency of the training sample, by removing irrelevant and redundant instances.

The practical significance of the results obtained is that the software implementing the proposed method has been developed. The given software has been experimentally investigated at the decision of problems reduction of synthetic and real world data. The conducted experiments have confirmed working capacity of the developed software. The results of the performed experiments allow recommending the use of the developed method and its software for solving the problems of technical diagnostics.

Further research in the field of reducing training samples by building ensembles of classifiers can be conducted in the following directions:

- development of adaptive ensembles of classifiers with a minimum number of initial parameters;
- using ensembles of classifiers different types;
- the use of different approaches to the formation of local samples for base classifiers, to create balanced data sets;
- search for optimal classification methods for the synthesis a base classifiers;
- implementation of the proposed method for multiprocessor systems operating in parallel modes.

8 References

1. García, S., Luengo, J., Herrera, F.: *Data Preprocessing in Data Mining*. Springer, Switzerland (2015)
2. García, S., Ramírez-Gallego, S., Luengo, J. et al.: Big data preprocessing: methods and prospects. In: *Big Data Anal*, vol. 1(9). Springer (2016). doi: 10.1186/s41044-016-0014-0
3. Chen, M., Mao, S., & Liu, Y.: Big Data: A Survey. In: *Mobile Networks and Applications*, vol. 19(2), pp. 171–209. Springer (2014). doi:10.1007/s11036-013-0489-0
4. Han, J., Kamber, M., Pei, J.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann, USA (2011)
5. Subbotin, S.: The sample properties evaluation for pattern recognition and intelligent diagnosis. In: *The 10th International Conference on Digital Technologies 2014, Zilina, 9–11 July 2014*, pp. 332–343. IEEE, Los Alamitos (2014)
6. Subbotin, S., Oliinyk, A., Levashenko, V., Zaitseva, E.: Diagnostic rule mining based on artificial immune system for a case of uneven distribution of classes in sample. In: *Communications - Scientific Letters of the University of Zilina*, vol. 18(3), pp. 3-11 (2016)
7. Boguslayev, A. V., Oleynik, Al. A., Oleynik, An. A. et al.: *Progressivnyye tekhnologii modelirovaniya, optimizatsii i intellektual'noy avtomatizatsii etapov zhiznennogo tsikla aviatsionnykh dvigateley: monografiya* [Progressive technologies for modeling, optimization, and intelligent automation of aircraft engine life cycle stages: monograph]. OAO «Motor Sich», Zaporozh'ye (2009)
8. Harley, J. B., Sparkman, D.: Machine learning and NDE: Past, present, and future. In: *AIP Conference Proceedings*, vol. 2102(1) (2019). doi:10.1063/1.5099819
9. Murphy, K.: *Machine Learning: A Probabilistics Perspective*. The MIT Press, Cambridge, Massachusetts (2012)
10. Bishop, C. M.: *Pattern recognition and machine learning*. Springer, New York (2014)
11. Lavrakas, P.J.: *Encyclopedia of survey research methods*. Sage Publications, Thousand Oaks (2008)
12. Fernández-Delgado, M., Cernadas, E., Barro, S. et al.: Do we need hundreds of classifiers to solve real world classification problems? In: *Journal of Machine Learning Research*, vol. 15(1), pp. 3133-3181 (2014)
13. Samarev, R., Vasnetsov, A., Smelkova, E.: Generalization of metric classification algorithms for sequences classification and labelling. In: *ArXiv: 1610.04718* (2016)
14. Abu Alfeilat, H. A., Hassanat, A. B. A., Lasassmeh, O. et al.: Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. In: *Big Data*, vol. 7(4), pp. 221-248. Mary Ann Liebert, Inc. (2019). doi:10.1089/big.2018.0175

15. Subbotin, S.: The neuro-fuzzy network synthesis and simplification on precedents in problems of diagnosis and pattern recognition. In: *Optical Memory and Neural Networks (Information Optics)*, vol. 22(2), pp. 97-103. Springer (2013)
16. Subbotin, S.: Quasi-relief method of informative features selection for classification. In: *2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT 2018)*, Lviv, 11-14 September 2018, pp. 318-321. Vezha i Ko, Lviv (2018)
17. Subbotin, S.: Methods of data sample metrics evaluation based on fractal dimension for computational intelligence model building. In *4th International Scientific-Practical Conference. Problems of Infocommunications. Science and Technology (PICS&T)*, Kharkov, 10-13 Oct. 2017, pp. 1-6. IEEE, Los Alamitos (2017)
18. Haro-García, A., Cerruela-García, G., García-Pedrajas, N.: Instance selection based on boosting for instance-based learners. In: *Pattern Recognition*, vol. 96. Elsevier (2019). doi: 10.1016/j.patcog.2019.07.004
19. Hamidzadeh, J., Monsefi, R., Sadoghi Yazdi, H.: IRAHC: Instance Reduction Algorithm using Hyperrectangle Clustering. In: *Pattern Recognition*, vol. 48(5), pp. 1878–1889. Elsevier (2015). doi: 10.1016/j.patcog.2014.11.005
20. Subbotin, S., Oliinyk, A.: The sample and instance selection for data dimensionality reduction. In: Szewczyk, R., Kaliczyńska, M. (eds.) *Advances in Intelligent Systems and Computing*, vol. 543, pp. 97-103. Springer, Cham (2017)
21. Subbotin, S.: The instance and feature selection for neural network based diagnosis of chronic obstructive bronchitis. In: Bris, R., Majernik, J., Pancerz, K., Zaitseva, E. (eds.) *Studies in Computational Intelligence*, vol 606, pp. 215–228. Springer, Cham (2015)
22. Subbotin, S.: Methods of sampling based on exhaustive and evolutionary search. In: *Automatic Control and Computer Sciences*, vol. 47(3), pp. 113–121. Springer (2013)
23. Liu, H., Motoda, H.: On issues of instance selection. In: *Data Mining and Knowledge Discovery*, vol. 6, pp. 115–130. Springer (2002)
24. Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F. et. al.: A review of instance selection methods. In: *Artificial Intelligence Review*, vol. 34(2), pp. 133–143. Springer (2010). doi:10.1007/s10462-010-9165-y
25. Garcia, S., Derrac, J., Cano, J. R. et. al.: Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34(3), pp. 417–435. IEEE (2012). doi: 10.1109/tpami.2011.142
26. Kavrin, D., Subbotin, S.: The sampling method preserving interclass boundaries. In: *CEUR Workshop Proceedings of the Second International Workshop on Computer Modeling and Intelligent Systems (CMIS-2019)*, vol. 2353, pp. 664-673. CEUR-WS (2019)
27. Kuncheva, L. I.: *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, Inc., Hoboken, New Jersey (2014)
28. Polikar, R.: Ensemble based systems in decision making. In: *IEEE Circuits and Systems Magazine*, 6(3), pp. 21–45. IEEE (2006). doi:10.1109/mcas.2006.1688199
29. Valentini, G., Re., M.: Ensemble methods: a review. In: *Advances in Machine Learning and Data Mining for Astronomy*, pp. 563-594. Chapman & Hall (2012)
30. Sammut, C., Webb, G.: *Encyclopedia of Machine Learning*. Springer (2017)
31. Breiman, L.: Bagging predictors. In: *Machine Learning*, vol. 24, pp. 123–140. Springer (1996)
32. Kotsiantis, S., B.: Bagging and boosting variants for handling classifications problems: a survey. In: *The Knowledge Engineering Review*, 29(1), pp. 78–100. Cambridge University Press (2013). doi:10.1017/s0269888913000313
33. Zhou, Z., H.: *Ensemble Methods Foundations and Algorithms*. Chapman & Hall (2012)

34. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. In: *Journal of Artificial Intelligence Research*, vol. 2, pp. 263–286. AI Access Foundation, USA (1995)
35. Wolpert, D., H.: Stacked generalization. In: *Neural Networks*, vol. 5(2), pp. 241–259. Elsevier (1992). doi: 10.1016/S0893-6080(05)80023-1
36. Yoav, F., Schapire, R., Abe, N.: A Short Introduction to Boosting. In: *Journal of Japanese Society For Artificial Intelligence*, vol. 14(5), pp. 771-780. (1999)
37. Blachnik, M.: Instance Selection for Classifier Performance Estimation in Meta Learning. In: *Entropy*, vol. 19(11) (2017). doi:10.3390/e19110583
38. Rokach, L.: Ensemble-based classifiers. In *Artificial Intelligence Review*, vol. 33, pp. 1–39. Springer (2010). doi:10.1007/s10462-009-9124-7
39. Jordan, M., I., Jacobs, R., A.: Hierarchical mixtures of experts and the EM algorithm. In: *Neural computation*, vol. 6(2), pp. 181-214. IEEE (1994). doi: 10.1162/neco.1994.6.2.181
40. Haykin, S., O.: *Neural Networks and Learning Machines*. Pearson (2008)
41. Kuhn, M., Johnson, K.: *Applied Predictive Modeling*. Springer (2018)
42. Thompson, S.K.: *Sampling*. John Wiley & Sons, Hoboken (2012)
43. Cochran, W.G.: *Sampling Techniques*. John Wiley & Sons, New York (1977)
44. Chaudhuri, A., Stenger, H.: *Survey sampling theory and method*. Chapman & Hall, New York (2005)
45. Good, P., I.: *Resampling methods: a practical guide to data analysis*. Birkhäuser (2005)
46. Scheaffer, L., Mendenhall, W., Lyman Ott, R. at. al.: *Elementary Survey Sampling*. Cengage Learning (2011)
47. Breiman, L.: Pasting Small Votes for Classification in Large Databases and On-Line. In: *Machine Learning*, vol. 36, pp. 85–103. Springer (1999)
48. Louppe, G., Geurts, P.: Ensembles on Random Patches. In: *Lecture Notes in Computer Science*, pp. 346–361. Springer (2012). doi:10.1007/978-3-642-33460-3_28
49. Wu, X., Kumar, V., Quinlan, J. at. al.: Top 10 algorithms in data mining. In: *Knowledge and Information Systems*, vol. 14, pp. 1–37. Springer (2008). doi: 10.1007/s10115-007-0114-2
50. Elkan, C.: The foundations of cost-sensitive learning. In: *17th international joint conference on Artificial intelligence 2001*, vol. 2, pp. 973-978. Morgan Kaufmann Publishers Inc. (2001)
51. Fawcett T.: An Introduction to ROC Analysis. In: *Pattern Recognition Letters*, vol. 27(8), pp. 861-874. Elsevier (2005). doi: 10.1016/j.patrec.2005.10.010
52. Zhang, S., Cheng, D., Deng, Z. at. al.: A novel KNN algorithm with data-driven k parameter computation. In: *Pattern Recognition Letters*, vol. 109, pp. 44-54. Elsevier (2018). doi: 10.1016/j.patrec.2017.09.036
53. Parsons, V., L.: Stratified Sampling. In: *Wiley StatsRef: Statistics Reference Online*. John Wiley & Sons (2017). doi: 10.1002/9781118445112.stat05999.pub2
54. Lyon, R., J., Stappers, B., W., Cooper, S. at. al.: Fifty Years of Pulsar Candidate Selection: From simple filters to a new principled real-time classification approach. In: *Monthly Notices of the Royal Astronomical Society*, vol. 459(1), pp. 1104-1123. Oxford (2016). doi: 10.1093/mnras/stw656
55. Alcalá-Fdez J., Fernandez A., Luengo J. at. al.: KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. In: *Journal of multiple-valued logic and soft computing*, vol. 17(4), pp. 255-287. Old city publishing (2010)