УДК 519.254
Doi: 10.31772/2712-8970-2021-22-1-18-31

Для цитирования: Денисов М. А., Сопов Е. А. Генетический алгоритм условной оптимизации для проектирования информативных признаков в задачах классификации // Сибирский аэрокосмический журнал. 2021. Т. 22, № 1. С. 18–31. Doi: 10.31772/2712-8970-2021-22-1-18-31.

For citation: Denisov M. A., Sopov E. A. Constraint handling genetic algorithm for feature engineering in solving classification problems // Siberian Aerospace Journal. 2021, Vol. 22, No. 1, P. 18–31. Doi: 10.31772/2712-8970-2021-22-1-18-31.

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ УСЛОВНОЙ ОПТИМИЗАЦИИ ДЛЯ ПРОЕКТИРОВАНИЯ ИНФОРМАТИВНЫХ ПРИЗНАКОВ В ЗАДАЧАХ КЛАССИФИКАЦИИ

М. А. Денисов[*], Е. А. Сопов

Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева
Российская Федерация, 660037, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31
[*]E-mail: denisov.maksim.work@gmail.com

Проектирование признаков в машинном обучении является перспективным, но недостаточно изученным направлением. Создание нового пространства признаков из исходного набора позволяет повысить эффективность алгоритма машинного обучения, применяемого для решения сложных задач интеллектуального анализа данных. Некоторые методы отбора часто способны одновременно при увеличении точности классификации уменьшить исходное пространство, что особенно актуально в эпоху больших данных.

В работе предлагается новый подход машинного обучения к решению задачи классификации на основе методов проектирования информативных признаков. Проектирование информативных признаков осуществляется с помощью методов извлечения и отбора. На основании исходных данных созданы новые множества признаков, которые включают исходные признаки и признаки, полученные методом главных компонент. Выбор эффективного подмножества информативных признаков реализуется с использованием генетического алгоритма. Для того чтобы избежать переобучения и создания тривиальных классификаторов, на функцию пригодности генетического алгоритма накладываются ограничения, требующие определенного количества признаков исходной выборки, а также определенного количества признаков, полученных методом главных компонент. Проведен сравнительный анализ эффективности следующих алгоритмов классификации: k-ближайших соседей, метод опорных векторов и случайный лес. Эксперименты по исследованию эффективности проводятся путем решения прикладных задач бинарной классификации из репозитория задач машинного обучения UCI Machine Learning. В качестве критерия эффективности выбрана мера macro F1-score.

Результаты численных экспериментов показали, что точность классификации предложенным подходом превосходит решения, полученные на исходном наборе признаков и при случайном отборе (оценка границы снизу). Причем, увеличение точности характерно для всех типов задач (выборки, у которых количество признаков больше числа объектов, а также объемом 500 значений и более). Подтверждена статистическая значимость результатов.

Ключевые слова: отбор признаков, извлечение признаков, генетический алгоритм, условная оптимизация

CONSTRAINT HANDLING GENETIC ALGORITHM FOR FEATURE ENGINEERING IN SOLVING CLASSIFICATION PROBLEMS

M. A. Denisov*, E. A. Sopov

Reshetnev Siberian State University of Science and Technology
31, Krasnoiarskii Rabochi Prospekt, Krasnoyarsk, 660037, Russian Federation
*E-mail: denisov.maksim.work@gmail.com

Feature engineering in machine learning is a promising but still insufficiently studied direction. Creating new feature space from an original set allows to increase accuracy of the machine learning algorithm chosen to solve complex data mining problems. Some existing selection methods are capable of simultaneously increasing accuracy and reducing feature space. The reduction is an urgent task for big data problems.

The paper considers a new machine learning approach for solving classification problems based on feature engineering methods. The design of informative features is carried out using extraction and selection methods. Based on the initial data, new sets of characteristics have been created, which include the original characteristics and characteristics obtained by the method of principal components. The choice of an effective subset of informative features is implemented using a genetic algorithm. In order to avoid overfitting and the creation of trivial classifiers, restrictions are imposed on the fitness function of the genetic algorithm, requiring a certain number of features of the original sample, as well as a certain number of features obtained by the principal component method. A comparative analysis of efficiency of the following classification algorithms is carried out: k-nearest neighbors, support vector machine, and a random forest. Efficiency research experiments are carried out by solving applied binary classification problems from the UCI Machine Learning repository of machine learning problems. The macro F1-score was chosen as an efficiency criterion.

The results of numerical experiments show that the proposed approach outperforms the solutions obtained using the original data set and the performance of random feature selection (the low bound for the results). Moreover, the accuracy enhancement is obtained for all types of problems (data sets that have more features than values). All results are proved to be statistically significant.

Keywords: feature selection, feature construction, genetic algorithm, constraint optimization

Introduction. Machine learning is an integral part of modern information technology and is widely used in many areas. For example, for handwriting recognition, image classification and spam filtering are used [1-3]. Science and technology, medicine, economics, and other industries also actively use machine learning algorithms in solving complex applied problems [4; 5]. Learning data is a key part for machine learning algorithms. In practice, when analyzing the data, it may turn out that some of the features are not informative. Such features are either unrepresentative or strongly correlated with each other. With the availability of unrepresentative features, whose contribution to the final accuracy is insignificant or absent, methods from the Feature Selection class are usually used [6; 7]. In situations where the features are strongly interrelated, that is, they influence the predictive ability of the system in the same way, the methods for constructing features (Feature Construction) or their extraction (Feature Extraction) are used [8, 9]. At the present stage, these approaches are summarized in a single term - Feature Engineering [10; eleven].

Recently, feature design methods have been actively researched and developed. With the emergence of big data, the problem of reducing the dimension of feature space has become even more urgent [12]. Feature selection methods can significantly reduce the required computing power of a computer while maintaining or increasing the forecast accuracy. At the same time, attempts are made to reduce the original dimension of the feature space by transforming it into a new one of lesser dimension [13]. However, research in this direction is still insufficient. This paper proposes to combine feature extraction and selection techniques together to obtain a new representation of raw data that increases predictive power. The problem of binary classification is considered. The Principal Component Analysis (PCA) is used as an extraction technique [14]. Further, the obtained features are combined with the original sample. The last step is the selection of informative features using a genetic algorithm (GA), which is additionally subject to restrictions set by a user, taking into account the practical goals of solving a problem, software or hardware implementations.

The article is organized as follows. The first section examines the existing works on the research topic. The second section is aimed at a detailed description of the proposed method for designing features using PCA and GA. The third section describes computational experiments. The conclusion summarizes and discusses further research prospects.

1. Analysis of literature on the topic. Even though the problems of design and extraction of features have been dealt with since the second half of the twentieth century, the terminology is still not well established. Some authors use a single term "feature construction", also meaning "feature extraction". Others give preference to "feature extraction". In this work, it is decided to separate these two concepts, since they solve fundamentally different and, in general case, independent problems.

1.1 Feature construction. By design we mean the process of creating new features using some transformations. The role of such transformations can be both mathematical operations (addition, subtraction, multiplication, and others) and logical operations (conjunction, disjunction, implication, etc.). Usually, the selected set of mathematical operators is unique for each specific problem and cannot be generalized [15, 16]. In [17], a special criterion is used to search for features that, when combined, could form a new one capable of giving better response accuracy. In [18], for an applied economic problem, a classification algorithm using a sample of constructed features shows better results compared to a classifier using the initial data. However, all these approaches cannot be generalized to arbitrary problems.

In this regard, at the end of the 20th early 21st century algorithms which can be used in various applied problems are being developed. They are, for example, *FRINGE* [19] and *CITRE* [20], which use binary operations and decision trees to create new features. The authors of *FICUS* [15] decided to improve the existing approaches and, in addition to binary operations, added standard mathematical and other functions that can be suggested by a subject area expert. The disadvantage of such methods is their computational complexity. At each iteration, more and more features are added to the original sample, which must be fed to the decision tree. As a result, the tree becomes too large.

Around the same period, algorithms based on genetic programming began to develop. For example, in works [21; 22], the population consists of individuals representing a coded set of arithmetic and logical operators. During evolution, with their help, a new space of features is formed, which is subsequently submitted to the classifier.

There is also a method for constructing features using inductive logic programming to generate predicates based on some a priori knowledge. In applied problems, it is used to eliminate semantic ambiguity of words in the process of processing and analysis of a natural language by a computer [23].

1.2. Feature extraction. The second type of this class of problems is feature extraction. Extraction means change in the original feature space by decreasing its dimension. The classical method is PCA and its variations [24]. In a general sense, this technique, using singular value decomposition of the data matrix, allows one to construct new features that are a linear combination of the original ones. The obtained features are uncorrelated, and the initial sample does not contain redundant information, which is a significant advantage of the method. This approach is classified as unsupervised learning. It does not require additional knowledge of a subject area. The disadvantage is that new data no longer reflects an original view, that is, it becomes almost impossible to interpret it.

The authors of this article in their work use the PCA method to extract features that are subsequently added to the original set. The logic of this manipulation lies in the principle of the algorithm. In the process of transformation of space, the first main component reflects the largest part of the variance of the entire set of data. Each subsequent component reflects this dispersion to a lesser extent. If you select the first few (most significant) components and add them to an original sample, you can enhance its predictive power. Compared to design methods, this approach does not require any additional knowledge of a subject area and thus can be used for any applied problem.

1.3 Feature selection. The task of feature selection implies a reduction in their number in order to increase computational performance, improve interpretability of the model while maintaining high classification accuracy. The fulfillment of these requirements is carried out by removing from the data redundant, irrelevant features, as well as those that can be attributed to noise. Works [25; 26] show that with the help of selection it is possible not only to preserve, but also to increase the forecast accuracy.

Search strategies are usually subdivided into subtypes: Filter methods, Wrapper Methods, and Embedded methods. The filtering methods are independent of the classifier (they are not built into it in any way). Despite the fact that this approach usually requires little computational power and can also be relatively easily generalized to various applied problems, it does not take into account the final accuracy obtained as a result of selection. Wrapping and embedded methods evaluate the resulting set of features using the forecast accuracy metric and, based on this estimate, improve the selection process during, for example, the evolution process. A distinctive feature of embedded methods is the search for the optimal set of features embedded in the structure of the classifier. In other words, selection is part of the learning process [27]. For wrapping methods, criterion for selecting features is accuracy of a classifier, that is, a classifier is a “wrapper” for a search algorithm.

It is important to note that feature selection is inherently challenging. The search space contains (2^n – 1) possible solutions, where n is a number of features in the problem. Considering that humanity has entered the era of big data [12], when signs can be counted not even in thousands, but in millions, one can imagine how much labor intensity is increasing. The study [28] shows that genetic algorithms are effectively used to solve the problem of feature selection.

1.4. Adaptive penalty. Генетические алгоритмы для решения задач условной оптимизации исследованы достаточно подробно. В работе [29] показаны различные варианты, включающие использование штрафных функций, специальных генетических операторов или алгоритмов на основе коэволюции.

Genetic algorithms for solving constrained optimization problems have been studied in sufficient detail. In [29], various options are shown, including the use of penalty functions, special genetic operators, or algorithms based on coevolution.

The article discusses the problem of optimizing the form:

$$f(x) \to \max_{x \in B_2^n}, \tag{1}$$

where $x = (x_1,\ldots, x_n)$, $x_i \in \{0,1\}$, $i = \overline{1,n}$.

В своей работе мы используем адаптивный штраф [30], который в среднем превосходит другие методы на основе штрафов. Математическое представление функции пригодности выглядит следующим образом In our work, we use an adaptive penalty [30], which, on average, is superior to other penalty-based methods. The mathematical representation of fitness function is as follows:

$$F(x) = \begin{cases} f(x), & \text{if } x \text{ is valid,} \\ \tilde{f}(x) - \sum_{j=1}^{l} k_j v_j(x), & \text{else,} \end{cases} \tag{2}$$

where $F(x)$ is the fitness function obtained by the penalty function method, $f(x)$ is the objective function, l in the adder is equal to the number of constraints imposed on the objective function, v_j is the numerical size of the violation of constraint j, k_j is the parameter of the penalty function for constraint j which is calculated as:

$$k_j = \frac{\left|\sum_{i=1}^{pop} f(x^i)\right|}{\sum_{s=1}^{l}\left[\sum_{i=1}^{pop} v_s(x^i)\right]^2}\sum_{i=1}^{pop} v_j(x^i), \tag{3}$$

where pop is population size. $\tilde{f}(x)$ is defined as:

$$\tilde{f}(x) = \begin{cases} f(x), & \text{if } f(x) > \langle f(x)\rangle, \\ \langle f(x)\rangle, & \text{else,} \end{cases} \tag{4}$$

where $\langle f(x)\rangle = \sum_{i=1}^{pop} f(x^i) \Big/ pop$.

2. Proposed approach. Limiting the search space

The paper investigates a classification problem, mathematical formulation of which can be presented as follows. Let U be a set of attributes, and Y - a set of class labels, that is, class names. It is assumed that there is an unknown transformation:

$$y^* : U \to Y, \tag{5}$$

whose values are known only on the objects of the training set:

$$U^m = \{(u_1, y_1), \dots, (u_m, y_m)\}. \tag{6}$$

It is required to build an algorithm:

$$a : U \to Y, \tag{7}$$

able to classify an arbitrary value $u \in U$ formed by a set of features $u \in \{\overline{Attr} \in attr_i,\ i = 1, \dots, n_1\}$. Let us denote the set of features obtained using PCA as $\overline{Attr}^{МГК} \in attr_i$, $i = 1, \dots, n_2$.

The paper considers a classification using several approaches to design features:

1) Feature extraction using PCA;
2) Creation of a new space of features by combining the initial ones with PCA;
3) Selection of features from the space obtained in 2 (initial with PCA) by genetic algorithm;
4) Selection of features of the initial set in a random way;
5) Selection of features randomly from the space obtained in 2 (initial with PCA).

Let's describe the approaches in more detail. In the first experiment, the accuracy of the classification of objects described by constructed features of PCA is estimated. For the second experiment, a new feature space is supplied to a classifier, obtained by combining original features of the sample with PCA: $\overline{Attr} \cup \overline{Attr}^{PCA}$. In the third experiment, the selection of features is carried out by a genetic algorithm. A sample similar to the second experiment is used as input data: $\overline{Attr} \cup \overline{Attr}^{PCA}$. The search strategy used in the third experiment is a wrapper method, where the classifier is a wrapper for a genetic search algorithm. The restrictions imposed on the objective function are described as follows:

$$\begin{cases} r_1 - \sum\limits_{i\in\overline{Attr}} x_i \le 0, \\ \sum\limits_{i\in\overline{Attr}} x_i - w_1 \le 0, \\ r_2 - \sum\limits_{i\in\overline{Attr}^{МГК}} x_i \le 0, \\ \sum\limits_{i\in\overline{Attr}^{МГК}} x_i - w_2 \le 0, \end{cases} \tag{8}$$

where r_1, r_2, w_1, w_2 are parameters indicating the number of features that will remain in the sample, x_i, i=1,…,n s the chromosome of the genetic algorithm. The size of chromosome is the sum of potency of a set $\overline{Attr}$ and a set $\overline{Attr}^{МГК}$: $n=n_1+n_2$. Zero in the chromosome denotes a trait that will not be taken into account in the classifier, and one is vice versa. In (8), the first two conditions require the presence of features from r_2 to w_2 from the initial sample, the third and fourth conditions require additional availability from r_2 to w_2 of PCA features. The final classification accuracy is defined as median value of accuracy results obtained from a series of 40 runs of the third experiment.

In the last two experiments, selection is carried out at random. A mask that is filled with 0 or 1 random number generator with a probability of p=0.5 is used. Similar to GA, this operation is repeated 40 times for the entire set of features, the comparison is based on the median.

3. Results of the experiment. We use tasks from the UCI Machine Learning repository [31]. Their main characteristics are presented in table 1.

Table 1

Main characteristics of the data selected for the study

	Number of classes	Number of features	Sample size
Breast Cancer	2	30	569
LSVT Voice Rehabilitation	2	310	126
Australian Credit	2	14	690
Heart Disease	2	13	270

The classifiers used in the work, the hyperparameters of which are tuned in the learning process, are given below:

– k-nearest neighbour (kNN). The number of neighbors is configurable in the interval [2, 100];

– Support Vector Mashine (SVM). Kernel type is configurable among [‘linear’, ’poly’, ’rbf’, ’sigmoid’];

– Random forest (RFC). The number of trees is configurable in the interval [1, 100].

Accuracy of mentioned classifiers is high on average and configuring their hyperparameters does not require a lot of time, which mainly allows you to concentrate on the task of finding informative features. These properties determine their use in work.

The best hyperparameter is the one with the highest median value. When setting up classifiers, the stratified *k-Fold* cross-validation method is used with the number of partitions k=5. The *macro F1-score* [32], which calculates the unweighted average for each class, was chosen as metric for assessing accuracy. Initial data are pre-normalized in the interval [0, 1]. The parameters r_1, r_2 and w_1, w_2 that limit the objective function when selecting features by a genetic algorithm are equal to $r_{1,2}$=2 and $w_{1,2}$=4. When using PCA, only the first 4 features with the highest values of the explanatory variance are taken into account, which subsequently form a set$\overline{Attr}^{PCA}$.

The described approaches are implemented using the Python 3.8.2 programming language and the Scikit-learn library [33] version 0.23.2. For classification, the KNeighborsClassifier functions with the specified default parameters, SVC with the specified default parameters, except for max_iter = 1000000 and RandomForestClassifier with the random_state = 1 parameter, and the rest are used by default. To calculate PCA features, functions of the PCA class in the sklearn.preprocessing module are used. The number of n_components calculated by the function is equal to the number of features in the original sample. In the case of using LSVT Voice Rehabilitation data, where the number of features exceeds the number of points, the value of n_components = 126. Cross-validation is performed using the StratifiedKFold function, and the normalization of samples is done using the MinMaxScaler function.

The parameters and features of a genetic algorithm are described below, with the help of which the required number of features increasing the classification accuracy is selected:

1) Initialization happens as follows. The features are selected equally in two stages. The used sample was constructed from the initial features and PCA. At the first stage, no more than 4 features of the initial sample are selected. On the second one, no more than 4 signs of PCA are selected. This is necessary to ensure the convergence of the algorithm. In the case when search space is large, as, for example, for the LSVT Voice Rehabilitation sample (310 features), convergence of the algorithm is slow if standard random initialization is used.

2) Tournament selection is used. The tournament size is 2.

3) Mating is single point.

4) The probability of a gene mutation is inversely proportional to the number of traits in the original sample.

5) Fitness function is the classification accuracy value obtained after stratified *k-Fold* cross-validation with the adaptive penalty described earlier.

6) The population size is 100, the number of individuals in the population is 100.

The results of solving the problems are presented in tables 2, 3, where the columns represent the sample and the classifier used for it, and the rows represent the type of experiment. Each cell contains the values of the macro F1-score classification accuracy metric obtained on the test sample (the median value of a series of 40 experiments). Experiment No. 4, where the features for classification were selected by a genetic algorithm with constraints, showed an advantage over other approaches.

Table 2

Results of computational experiments (part 1)

Experiment No. *	Breast Cancer			LSVT Voice Rehabilitation		
	kNN	SVM	RFC	kNN	SVM	RFC
1	0.965	0.973	0.960	0.795	0.838	0.836
2	0.960	0.963	0.958	0.784	0.785	0.808
3	0.966	0.975	0.957	0.788	0.827	0.834
4	0.975	0.977	0.974	0.888	0.870	0.883
5	0.961	0.967	0.955	0.778	0.827	0.802
6	0.960	0.969	0.958	0.779	0.832	0.811

* Note: 1 – all features; 2 – PCA; 3 – construction of PCA features with initial; 4 – features selected by GA from initial + PCA; 5 – random selection from source; 6 – random selection from initial + PCA.

Table 3

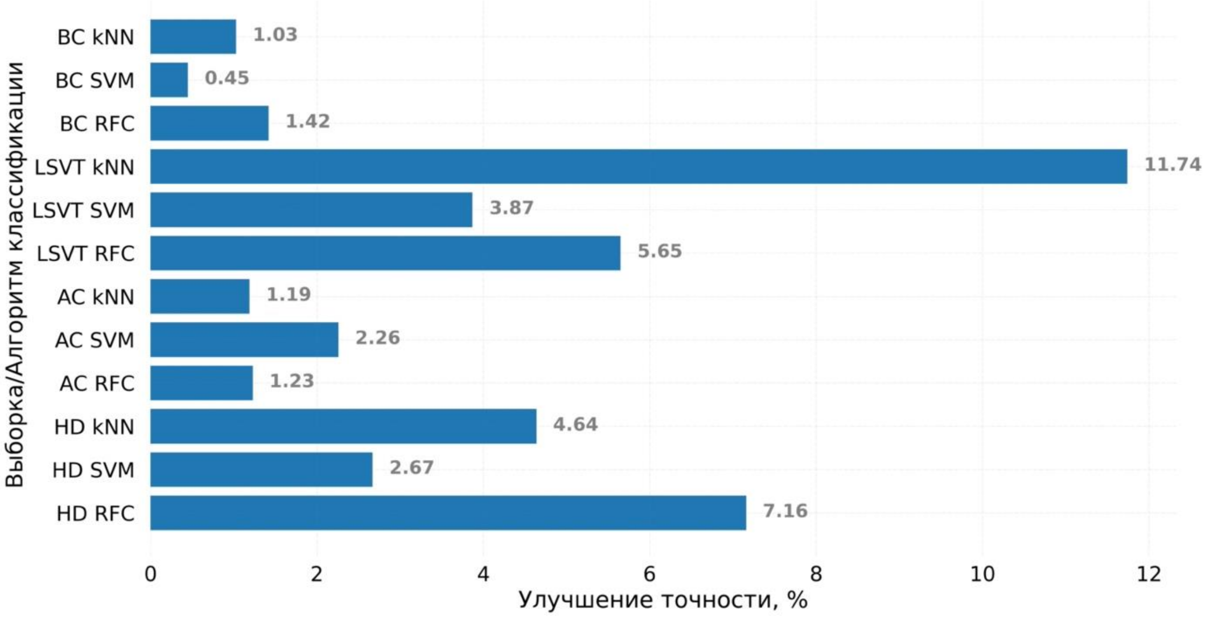
Results of computational experiments (part 2)

Experiment No. *	Australian Credit	Heart Disease

	kNN	SVM	RFC	kNN	SVM	RFC
1	0.871	0.854	0.872	0.837	0.845	0.820
2	0.877	0.855	0.840	0.832	0.819	0.815
3	0.870	0.856	0.873	0.833	0.845	0.860
4	0.881	0.873	0.884	0.876	0.868	0.879
5	0.855	0.854	0.803	0.806	0.786	0.775
6	0.859	0.854	0.856	0.814	0.823	0.812

* Note: 1 – all features; 2 – PCA; 3 – construction of PCA features with initial; 4 – features selected by GA from initial + PCA; 5 – random selection from source; 6 – random selection from initial + PCA.

Fig. 1 in the form of a diagram shows the increase in accuracy in percent of the experiment 4 compared to the experiment 1 for each sample from the Table 1 and considered in the work classifiers.

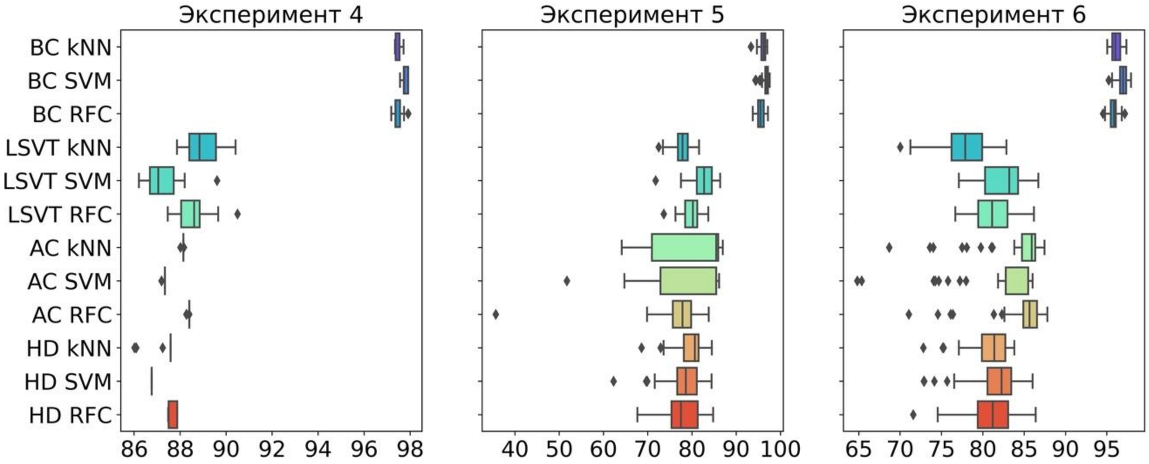


Выборка/Алгоритм классификации - Sample /Classification algorithm
Улучшение точности - Accuracy improving

Рис. 1. Прирост точности в процентах для 4 эксперимента по сравнению с 1
Fig. 1 Percentage accuracy increase for the experiment 4 in comparison with the experiment 1

Based on the results shown in Fig. 1, an increase in the classification accuracy is fixed for all samples. Its highest value can be noted for the LSVT Voice Rehabilitation sample, which in the original version contains 310 features. It is important to note that increase in accuracy was achieved with a significantly smaller number of features - 8 (4 of them are PCA features).

Below is a chart of the magnitude of the 40-run accuracy results for the experiments 4, 5, 6:



Эксперимент - Experiment

Рис. 2. Диаграмма размаха для экспериментов 4, 5, 6

Fig. 2. Box plot for experiments 4, 5, 6

In Fig. 2 for the experiments 5, 6 the range of distributions of accuracy values is higher than for the experiment 4. Hence, we can conclude that the approach proposed in this work has a stable solution. In addition, on the basis of the one-sided nonparametric Mann-Whitney U-test for the experiment 4 in relation to the experiments 5 and 6, the statistical significance of differences in the results was revealed.

Conclusion. In this paper, it is proposed to combine the techniques of feature extraction and selection in order to obtain a new representation of initial sample to increase the classification accuracy. The described approach of feature extraction using PCA with their subsequent addition to initial data and selection by a genetic algorithm with constraints showed greater efficiency compared to other methods of feature design used in the work. An increase in accuracy was recorded when classifying samples of different sizes.

In addition, the statistical significance of the results of proposed approach was confirmed in comparison with the selection of features at random (lower bound of accuracy limit). The proposed approach has a smaller spread of the *macro F1-score* metric values over a series of independent launches.

The restrictions imposed on a fitness function for feature selection may be of practical applicability in cases when it is required by a software or hardware component of the project being implemented. For example, under certain limitations of the communication channel in the process of transferring information or insufficient memory capacity.

In the future, it is planned to conduct a study of other approaches to the design of features. For example, an autoencoder type neural network [34] for feature extraction. Unlike PCA, such a network can operate with nonlinear dependencies, which can contribute to an increase in accuracy. Another approach is a genetic programming method for constructing features, which allows not only to create an effective (in terms of accuracy) set of features, but also to “justify” the obtained solution in the form of a mathematical function, which subsequently allows increasing not only the interpretability of solution, but also the amount of knowledge about initial data.

Библиографические ссылки

1. Guzella T. S., Caminhas W. M. A review of machine learning approaches to spam filtering // Expert Systems with Applications. 2009. Vol. 36, No. 7. P. 10206–10222.

2. Ballestar M. T., Grau-Carles P., Sainz J. Predicting customer quality in e-commerce social networks: a machine learning approach // Review of Managerial Science. 2019. Vol. 13, No. 3. P. 589–603.

3. Bahlmann C., Haasdonk B., Burkhardt H. Online handwriting recognition with support vector machines-a kernel approach // Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition. 2002. P. 49–54.

4. Kononenko I. Machine learning for medical diagnosis: history, state of the art and perspective // Artificial Intelligence in medicine. 2001. Vol. 23, No. 1. P. 89–109.

5. Kouziokas G. N. Machine learning technique in time series prediction of gross domestic product // Proceedings of the 21st Pan-Hellenic Conference on Informatics. 2017. P. 1–2.

6. John G. H., Kohavi R., Pfleger K. Irrelevant features and the subset selection problem // Machine Learning Proceedings. 1994. P. 121–129.

7. Kira K., Rendell L. A. A practical approach to feature selection // Machine Learning Proceedings. 1992. P. 249–256.

8. Rendell L., Seshu R. Learning hard concepts through constructive induction: Framework and rationale // Computational Intelligence. 1990. Vol. 6, No. 4. P. 247–270.

9. Liu H., Motoda H. Feature extraction, construction and selection: A data mining perspective. Massachusetts : Kluwer Academic Publishers, 1998. 453 p.

10. Duboue P. The Art of Feature Engineering: Essentials for Machine Learning. Cambridge : Cambridge University Press. 2020. 270 p.

11. Zheng A., Casari A. Feature engineering for machine learning: principles and techniques for data scientists. Sebastopol : O'Reilly Media Inc., 2018. 193 p.

12. Feature selection: A data perspective / Li J., Cheng K., Morstatter F. et al. // ACM Computing Surveys (CSUR). 2017. Vol. 50, No. 6. P. 1–45.

13. Park M. S., Na J. H., Choi J. Y. PCA-based feature extraction using class information // 2005 IEEE International Conference on Systems, Man and Cybernetics. 2005. Vol. 1. P. 341–345.

14. Abdi H., Williams L. J. Principal component analysis // Wiley interdisciplinary reviews: computational statistics. 2010. Vol. 2, No. 4. P. 433–459.

15. Markovitch S., Rosenstein D. Feature generation using general constructor functions // Machine Learning. 2002. Vol. 49, No. 1. P. 59–98.

16. Hirsh H., Japkowicz N. Bootstrapping training-data representations for inductive learning: A case study in molecular biology // AAAI-94 Proceedings. 1994. P. 639–644.

17. Sutton R. S., Matheus C. J. Learning polynomial functions by feature construction // Machine Learning Proceedings. 1991. P. 208–212.

18. Zhao H., Sinha A. P., Ge W. Effects of feature construction on classification performance: An empirical study in bank failure prediction // Expert Systems with Applications. 2009. Vol. 36, No. 2. P. 2633–2644.

19. Pagallo G. Haussler D. Boolean feature discovery in empirical learning // Machine learning. 1990. Vol. 5, No. 1. P. 71–99.

20. Matheus C. J., Rendell L. A. Constructive Induction on Decision Trees // IJCAI'89: Proceedings of the 11th international joint conference on Artificial intelligence. 1989. Vol. 89. P. 645–650.

21. Krawiec K. Genetic programming-based construction of features for machine learning and knowledge discovery tasks // Genetic Programming and Evolvable Machines. 2002. Vol. 3, No. 4. P. 329–343.

22. Smith M. G., Bull L. Genetic programming with a genetic algorithm for feature construction and selection // Genetic Programming and Evolvable Machines. 2005. Vol. 6, No. 3. P. 265–281.

23. An investigation into feature construction to assist word sense disambiguation / Specia L., Srinivasan A., Sachindra J. et al. // Machine Learning. 2009. Vol. 76, No. 1. P. 109–136.

24. Khalid S., Khalil T., Nasreen S. A survey of feature selection and feature extraction techniques in machine learning // 2014 Science and Information Conference. 2014. P. 372–378.

25. Кривенко М. П. Критерии значимости отбора признаков классификации // Информатика и её применения. 2016. Т. 10, №. 3. С. 32–40.

26. Miao J., Niu L. A survey on feature selection // Procedia Computer Science. 2016. Vol. 91. P. 919–926.

27. Chandrashekar G., Sahin F. A survey on feature selection methods // Computers & Electrical Engineering. 2014. Vol. 40, No. 1. P. 16–28.

28. A survey on evolutionary computation approaches to feature selection / Xue B., Zhang M., Browne W. et al. // IEEE Transactions on Evolutionary Computation. 2015. Vol. 20, No. 4. P. 606–626.

29. Coello C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art // Computer methods in applied mechanics and engineering. 2002. Vol. 191, No. 11–12. P. 1245–1287.

30. Barbosa H. J. C., Lemonge A. C. C. An adaptive penalty method for genetic algorithms in constrained optimization problems // Frontiers in Evolutionary Robotics, 2008.

31. UCI Machine Learning Repository [Электронный ресурс]. URL: https://archive.ics.uci.edu/ml/index.php (дата обращения: 09.01.2021).

32. Opitz J., Burst S. Macro f1 and macro f1. Препринт: arXiv:1911.03347. [Электронный ресурс]. URL: https://arxiv.org/abs/1911.03347 (дата обращения: 25.02.2021).

33. Scikit-learn: Machine learning in Python / Pedregosa F., Varoquaux G., Gramfort A. et al. // Journal of machine Learning research. 2011. Vol. 12. P. 2825–2830.

34. Dong G, Liao G., Liu H, Kuang G. A review of the autoencoder and its variants: A comparative perspective from target recognition in synthetic-aperture radar images // IEEE Geoscience and Remote Sensing Magazine. 2018. Vol. 6, No. 3. P. 44–68.

References

1. Guzella T. S., Caminhas W. M. A review of machine learning approaches to spam filtering. *Expert Systems with Applications*. 2009, Vol. 36, No. 7, P. 10206–10222. Doi: 10.1016/j.eswa.2009.02.037.

2. Ballestar M. T., Grau-Carles P., Sainz J. Predicting customer quality in e-commerce social networks: a machine learning approach. *Review of Managerial Science*. 2019, Vol. 13, No. 3, P. 589–603. Doi: 10.1007/s11846-018-0316-x.

3. Bahlmann C., Haasdonk B., Burkhardt H. Online handwriting recognition with support vector machines-a kernel approach. *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*. 2002, P. 49–54. Doi: 10.1109/IWFHR.2002.1030883.

4. Kononenko I. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*. 2001. Vol. 23, No 1, P. 89–109. Doi: 10.1016/S0933-3657(01)00077-X.

5. Kouziokas G. N. Machine learning technique in time series prediction of gross domestic product. *Proceedings of the 21st Pan-Hellenic Conference on Informatics*. 2017, P. 1–2. Doi: 10.1145/ 3139367.3139443.

6. John G. H., Kohavi R., Pfleger K. Irrelevant features and the subset selection problem. *Machine Learning Proceedings*. 1994, P. 121–129. Doi: 10.1016/B978-1-55860-335-6.50023-4.

7. Kira K., Rendell L. A. A practical approach to feature selection. *Machine Learning Proceedings*. 1992, P. 249–256. Doi: 10.1016/B978-1-55860-247-2.50037-1.

8. Rendell L., Seshu R. Learning hard concepts through constructive induction: Framework and rationale. *Computational Intelligence*. 1990, Vol. 6, No. 4, P. 247–270. Doi: 10.1111/j.1467-8640. 1990.tb00298.x.

9. Liu H., Motoda H. *Feature extraction, construction and selection: A data mining perspective.* Massachusetts : Kluwer Academic Publishers, 1998, 453 p.

10. Duboue P. *The Art of Feature Engineering: Essentials for Machine Learning.* Cambridge : Cambridge University Press, 2020, 270 p. Doi: 10.1017/9781108671682.

11. Zheng A., Casari A. *Feature engineering for machine learning: principles and techniques for data scientists.* Sebastopol : O'Reilly Media Inc., 2018, 193 p.

12. Li J., Cheng K., Morstatter F. et al. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*. 2017, Vol. 50, No. 6, P. 1–45. Doi: 10.1145/3136625.

13. Park M. S., Na J. H., Choi J. Y. PCA-based feature extraction using class information. *2005 IEEE International Conference on Systems, Man and Cybernetics*. 2005, Vol. 1, P. 341–345. Doi: 10.1109/ICSMC.2005.1571169.

14. Abdi H., Williams L. J. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*. 2010, Vol. 2, No. 4, P. 433–459. Doi: 10.1002/wics.101.

15. Markovitch S., Rosenstein D. Feature generation using general constructor functions. *Machine Learning*. 2002, Vol. 49, No. 1, P. 59–98. Doi: 10.1023/A:1014046307775.

16. Hirsh H., Japkowicz N. Bootstrapping training-data representations for inductive learning: A case study in molecular biology. *AAAI-94 Proceedings,* 1994, P. 639–644.

17. Sutton R. S., Matheus C. J. Learning polynomial functions by feature construction. *Machine Learning Proceedings*. 1991, P 208–212. Doi: 10.1016/B978-1-55860-200-7.50045-3.

18. Zhao H., Sinha A. P., Ge W. Effects of feature construction on classification performance: An empirical study in bank failure prediction. *Expert Systems with Applications*. 2009, Vol. 36, No. 2, P. 2633–2644. Doi: 10.1016/j.eswa.2008.01.053.

19. Pagallo G. Haussler D. Boolean feature discovery in empirical learning. *Machine learning*. 1990, Vol. 5, No 1, P. 71–99. Doi: 10.1023/A:1022611825350.

20. Matheus C. J., Rendell L. A. Constructive Induction on Decision Trees. *IJCAI'89: Proceedings of the 11th international joint conference on Artificial intelligence*. 1989, Vol. 89, P. 645–650.

21. Krawiec K. Genetic programming-based construction of features for machine learning and knowledge discovery tasks. *Genetic Programming and Evolvable Machines*. 2002, Vol. 3, No. 4, P. 329–343. Doi: 10.1023/A:1020984725014.

22. Smith M. G., Bull L. Genetic programming with a genetic algorithm for feature construction and selection. *Genetic Programming and Evolvable Machines*. 2005, Vol. 6, No. 3, P. 265–281. Doi: 10.1007/s10710-005-2988-7.

23. Specia L., Srinivasan A., Sachindra J., et al. An investigation into feature construction to assist word sense disambiguation. *Machine Learning*. 2009, Vol. 76, No 1, P. 109–136. Doi: 10.1007/ s10994-009-5114-x.

24. Khalid S., Khalil T., Nasreen S. A survey of feature selection and feature extraction techniques in machine learning. *2014 Science and Information Conference*. 2014, P. 372–378. Doi: 10.1109/SAI. 2014.6918213.

25. Krivenko M. P. [Significance tests of feature selection for classification]. *Informatics and Applications*. 2016, Vol. 10, No. 3, P. 32–40. Doi: 10.14357/19922264160305. (In Russ.)

26. Miao J., Niu L. A survey on feature selection. *Procedia Computer Science*. 2016, Vol. 91, P. 919–926. Doi: 10.1016/j.procs.2016.07.111.

27. Chandrashekar G., Sahin F. A survey on feature selection methods. *Computers & Electrical Engineering*. 2014, Vol. 40, No. 1, P. 16–28. Doi: 10.1016/j.compeleceng.2013.11.024.

28. Xue B., Zhang M., Browne W. et al. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*. 2015, Vol. 20, No. 4, P. 606–626. Doi: 10.1109/TEVC.2015.2504420.

29. Coello C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering*. 2002, Vol. 191, No. 11–12, P. 1245–1287. Doi: 10.1016/S0045-7825(01)00323-1.

30. Barbosa H. J. C., Lemonge A. C. C. An adaptive penalty method for genetic algorithms in constrained optimization problems. *Frontiers in Evolutionary Robotics*. 2008. Doi: 10.5772/5446.

31. UCI Machine Learning Repository Available at: https://archive.ics.uci.edu/ml/index.php (accessed 09.01.2021).

32. Opitz J., Burst S. Macro f1 and macro f1. Preprint arXiv:1911.03347. Available at: https://arxiv.org/abs/1911.03347 (accessed 25.02.2021).

33. Pedregosa F., Varoquaux G., Gramfort A. et al. Scikit-learn: Machine learning in Python. *Journal of machine Learning research*. 2011, Vol. 12, P. 2825–2830.

34. Dong G., Liao G., Liu H, Kuang G. A review of the autoencoder and its variants: A comparative perspective from target recognition in synthetic-aperture radar images. *IEEE Geoscience and Remote Sensing Magazine*. 2018. Vol. 6, No. 3, P. 44–68. Doi: 10.1109/MGRS.2018.2853555.

Денисов Максим Андреевич – аспирант; Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева. E-mail: max_denisov00@mail.ru.

Сопов Евгений Александрович – кандидат технических наук, доцент; Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева. E-mail: evgenysopov@gmail.com.

Denisov Maksim Andreevich – postgraduate; Reshetnev Siberian State University of Science and Technology. E-mail: max_denisov00@mail.ru.

Sopov Evgenii Aleksandrovich – PhD (CS), associate professor; Reshetnev Siberian State University of Science and Technology. E-mail: evgenysopov@gmail.com.