UDC 519.87

Vestnik SibGAU
Vol. 16, No. 1, P. 131–136

SELF-CONFIGURING HYBRID EVOLUTIONARY ALGORITHM FOR MULTI-CLASS UNBALANCED DATASETS

V. V. Stanovov[*], O. E. Semenkina

Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation
[*]E-mail: vladimirstanovov@yandex.ru

This paper describes a modification of the self-configuring hybrid evolutionary algorithm for solving classification problems. The algorithm implements a hybridization of Pittsburg and Michigan approaches, where Michigan part is used together with mutation operator. The rule bases use fixed fuzzy terms, and the number of rules in the rule base can change during the algorithm run. Also, the applied algorithm uses a set of heuristics to determine the weights and class labels for every fuzzy rule, using the confidence values, which are calculated using the training sample. A special initialization procedure allows getting more accurate fuzzy rule bases on the first generations. The modification changes the procedure of determining the most appropriate class number for the fuzzy rule. It uses the number of instances of different classes, as a weighting coefficient to avoid confidence values bias. Also, we apply two classification quality measures, the classical accuracy value and the average accuracy among classes. The modification, combined with different classification quality measures, allows improvement in the classification results. The self-configuring algorithm is tested on a set of unbalanced classification problems with several classes using cross-validation and a stratified sampling procedure. The test problems included image segment classification, bank client classification, phoneme recognition, classification of page contents, and satellite image classification. For one of the problems, the confusion matrixes are provided to show the increasing balance over the class accuracies. The presented method has efficiently solved the satellite images classification problem and can be applied for many real-life problems, including the problems from aerospace area.

Keywords: fuzzy classification system, unbalanced data, evolutionary algorithm, self-configuration.

Вестник СибГАУ
Т. 16, № 1. С. 131–136

САМОКОНФИГУРИРУЕМЫЙ ГИБРИДНЫЙ ЭВОЛЮЦИОННЫЙ АЛГОРИТМ ДЛЯ ЗАДАЧ С НЕСБАЛАНСИРОВАННЫМИ ДАННЫМИ И МНОЖЕСТВОМ КЛАССОВ

В. В. Становов[*], О. Э. Семенкина

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, 660014, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31
[*]E-mail: vladimirstanovov@yandex.ru

Рассматривается модификация самоконфигурируемого гибридного эволюционного алгоритма для решения задач классификации. В алгоритме реализована гибридизация Питсбургского и Мичиганского подходов, где Мичиганская часть используется вместе с оператором мутации. Базы правил используют фиксированные нечеткие термы, а число правил в базе может меняться в ходе работы алгоритма. Также примененный алгоритм использует набор эвристик для определения весов и номеров классов для каждого нечеткого правила с использованием значений достоверности (confidence), которые рассчитываются по обучающей выборке. Особая процедура инициализации позволяет получать более точные нечеткие базы правил на первых поколениях. Модификация изменяет процедуру определения наиболее подходящего номера класса для нечеткого правила. Она использует число объектов различных классов в качестве весовых коэффициентов, чтобы избежать смещения значений достоверности. Модификация в комбинации с другими мерами качества классификации позволяет улучшить результаты классификации. Самоконфигурируемый алгоритм был протестирован на ряде задач классификации с несбалансированными данными и несколькими классами с применением процедуры кросс-валидации и стратифицированным разбиением выборки. Тестовые задачи включали классификацию сегментов изображения, классификацию клиентов банка, распознавание фонем, классификацию содержимого страниц и классификацию снимков со спутника. Для одной из задач были приведены матрицы ошибок, для того чтобы показать увеличение баланса точности по классам. Представленный подход успешно решил задачу классификации снимков со спутника и может быть применен для множества реальных задач, включая задачи из аэрокосмической области.

Ключевые слова: нечеткие системы классификации, несбалансированные данные, эволюционный алгоритм, самоконфигурирование.

Introduction. Classification problems are typical problems for data analysis, and a vast number of approaches and techniques have been proposed to solve them. These techniques, such as, for example, artificial neural networks (ANN), support vector machines (SVM) and genetic programming (GP) might show very good results on some datasets, but there are still some problems to be resolved. These include processing big datasets with many instances, feature selection problems, missing values, and so on. Among them, there is also the problem of unbalanced data; it has been recently said to be one of the main obstacles preventing good classification results [1; 2].

The imbalance in classification datasets means that the number of instances of different classes is not the same for all classes. For example, for two-class datasets, this problem occurs when the number of instances of one class is much lower than the number of instances of the other class. The class that has more instances is called the majority class, while the other one is called the minority class. The higher the unbalance ratio is, the more difficult the classification problem becomes.

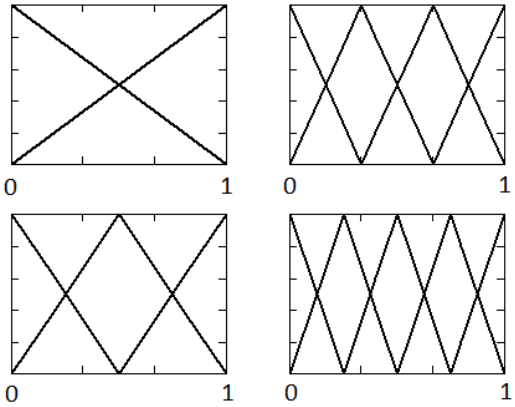
During the classifier learning process, especially when learning is conducted by evolutionary algorithms, this unbalance may lead to bias in classification results. That means that the classifier focuses on the majority class, and most of its instances are classified correctly, while the minority class is classified incorrectly. The reason for this is the learning procedure, and basically the classification quality criterion, which in most cases is simply the overall accuracy on the whole dataset. Although the overall accuracy may seem to be high, actually only the majority class may be classified correctly, so that for unbalanced datasets accuracy cannot be an adequate criterion. One more important thing is that in the vast number of real-world problems the minority class is the class-of-interest for the researcher; classifying it correctly is actually solving the problem.

The known methods for dealing with unbalance problems are divided into two groups: external and internal approaches. External approaches are used to change the learning sample so that it becomes balanced; these approaches use some sampling methods and can be very helpful in some situations. Yet in this paper we will focus on internal approaches, which are used to change the learning criterion so that it takes the unbalance ratio into account. Moreover, we will introduce the different modified initialization procedure for the algorithm that we used in our previous works.

Hybrid fuzzy evolutionary algorithm. The algorithm used in this work is based on ideas by Ishibuchi [3; 4], and the main idea of learning fuzzy classifiers from data was presented by Wang and Mendel [5]. This algorithm combines the Pittsburg and Michigan approaches for evolutionary classifier learning so that the Michigan-style algorithm is used as a mutation operator. We will give a short description of the main features of this algorithm here.

Each individual in the population is a rule base and the number of rules is not fixed, although the maximum number of rules cannot be exceeded. Each rule is an integer string with numbers from [0, 14], each meaning the number of a certain fuzzy set. For each variable several fuzzy partitions are used as shown in picture. There are 4 partitions into 2, 3, 4, and 5 fuzzy sets. Also the "don't care" term is used and named as a zero fuzzy set. During the fuzzy inference procedure the "don't care" condition always returned the membership function value equal to 1, whatever the value of the variable is. There are also possible some other approaches for setting the fuzzy partitions, for example, 2-tuples representation [6].

There are two types of selection used (rank and tournament), one crossover operator and three mutation operators (weak, medium and strong). The tournament size was equal to 3, and the mutation probability depended on the number of rules in the rule base. After the mutation operator, the Michigan part is applied to each rule base, so that each rule in it is an individual. There are three types of Michigan operators that could be applied to the rules in the base. The first one adds new rules, if the size of the rule base was not exceeded, the second deletes the worst rules from the base, and the third firstly deletes rules and then adds new rules into the base. Adding new rules is performed by heuristic and genetic approaches. The heuristic approach generates new rules from the misclassified instances, and the genetic approach generates rules from existing ones by genetic operators.



Four fuzzy partitions used in the algorithm

For selection, mutation, the Michigan part and the type of rules adding the self-configuration procedure described in [7; 8] is applied. This self-configuration is based on the success rates of the operators, calculated using the fitness improvements.

For each rule the class number and the rule weight are calculated, and these values are not coded in the chromosome, but determined heuristically. The confidence value is used in both cases so we used the approaches proposed in [9]. During the initialization procedure, after the antecedent parts of rules are set at random, the confidence value for the new rule is calculated as follows:

$$Conf(A_q \to \text{Class } k) = \frac{\sum_{x_p \in \text{Class } k} \mu_{A_q}(x_p)}{\sum_{p=1}^{m} \mu_{A_q}(x_p)},$$

where $\mu_{A_q}(x_p)$ is the membership function value for instance x_p and antecedent part A_q. In the original algorithm, the class number having the highest confidence value was attached to the rule, but for unbalanced datasets, the confidence values become biased as they calculate the sum of membership functions over all instances of every class. This means that for the minority class the confidence values can be smaller not because the rule covers the instances of this class worse than the majority class, but simply because the number of instances is not balanced. This leads to the problem that most of the generated rules are tied with the majority class, although they may describe an important part of the minority class.

To avoid this bias, a modified procedure of class number definition is proposed, which uses the number of class instances for every class as a weight:

$$\text{Class } q = \arg\max_{k}\left(\frac{Conf(A_q \to \text{Class } k)}{m_k} \cdot m\right),$$

where m_k is the number of instances in class k. This modification is supposed to improve classification quality and accuracy in case of unbalanced datasets. For the balanced datasets, this equation does not change the results of choosing the class number.

***Acc* and *Ave* values for estimating the classification quality.** The typical approach for the estimation of classification quality is a use of the accuracy, as has been said previously. The overall accuracy, *Acc* is determined as the number of incorrectly classified instances, or instances that were not classified anywhere, divided by the total number of instances in the learning sample. The other way of estimating the classification quality is by calculating the accuracy separately for every class. The accuracies on all classes are summed and then divided by the number of classes, resulting in the *Ave* measure. The following two equations are provided to formally describe these measures:

$$Acc = \frac{\sum_{i=0}^{k} E_i}{\sum_{i=0}^{k} m_i},,$$

$$Ave = \sum_{i=0}^{k} \frac{E_i}{m_i \cdot k},$$

where E_i is the number of errors of the classifier for the i-th true class. Although the *Ave* measure is not the only one used for unbalanced datasets, it is simple enough and allows significant improvement in the classification quality as will be shown later. The *Ave* measure can be also used as a quality criterion in other algorithms, for example, for algorithms that automatically design artificial neural networks [10; 11] or support vector machines [12]. Also, each part of the sum in *Ave* measure equation can be used as a separate quality criterion in a multi-objective algorithm for automated design of ANNs [13] or other machine learning techniques. The fitness function was a combination of three parameters – *Acc* or *Ave* value, number of rules and total length of all the rules in the base. Some other approaches apply multi-objective methods, which can be very helpful [14].

Algorithm testing and results. To see how the algorithm performs on complex classification problems, we have chosen some problems from the UCI Machine Learning Repository [15] and used four algorithm configurations: with modified class definition and without, with *Acc* and with *Ave* as the main quality estimation used in the fitness function.

The problems on which we tested the algorithms are the following:

1. Page-blocks, 5472 instances, 10 variables, 5 classes.
2. German credit, 1000 instances, 24 variables, 2 classes.
3. Phoneme, 5404 instances, 5 variables, 2 classes.
4. Segment, 2310 instances, 19 variables, 7 classes.
5. Satimage, 6435 instances, 36 variables, 6 classes.

The tab. 1 below shows the number of instances for each class in these problems.

To receive adequate results, we used stratified sample splitting into 10 folds to perform 10-fold cross validation. The cross validation procedure was performed three times and the results were averaged over all 30 runs of the algorithm. The algorithm resources were the following: 100 individuals, 500 generations, 40 rules maximum.

The next tab. 2 shows the results for the first problem, the *Acc* and *Ave* values for four parameter combinations.

When using the *Acc* value as the classification quality, the accuracies on both learning and test samples are the best, while the *Ave* values are very low. Applying the *Ave* measure significantly improves *Ave* values, but the accuracy decreases. Adding unbiased class numbering not only increases *Ave* values, but also improves the accuracy (tab. 3).

For the German dataset the results are almost the same, but the learning and test accuracy does not change very much. Again, unbiased class numbering gives the best test *Ave* values (tab. 4).

This problem is rather interesting because here the *Ave* values appeared to be higher than the accuracy for configurations #2 and #4, and this is true for both the learning and testing samples. This can be due to the fact that the minority class was easier to describe for the learning algorithm than the majority class, so that the accuracy on it was much higher. The unbiased class numbering does not give any advantages for this problem (tab. 5).

Although this problem has 7 classes, all of them are balanced. The results show that in this case there is no difference in classification results for all modifications as the quality criteria are not sensitive to biased classification in this case (tab. 6).

The last problem appears to be not very sensitive to different quality criteria, but a slight improvement in the *Ave* measure can be tracked when using it in the fitness function. An important thing about this problem is that in most of the cases, the algorithm was not able to build any significantly good rule for the third class, so that it was always misclassified into the second class.

To show the real classification results here we also provide averaged confusion matrixes for the first, second and fourth configurations for the test sample to show the difference. First, two confusion matrixes are provided for the page-blocks problem (tab. 7–9).

Table 1

Class instances for all problems

Problem/Class	1	2	3	4	5	6	7
Page-blocks	4913	329	28	87	115	–	–
German	700	300	–	–	–	–	–
Phoneme	3818	1586	–	–	–	–	–
Segment	330	330	330	330	330	330	330
Satimage	1533	703	1358	626	707	1508	–

Table 2

Results for Page-blocks problem

Configuration	Learning *Acc*	Learning *Ave*	Test *Acc*	Test *Ave*
Acc+biased	**0.959**	0.600	**0.955**	0.584
Ave+biased	0.879	0.810	0.874	0.767
Acc+unbiased	0.951	0.602	0.947	0.573
Ave+unbiased	0.891	**0.855**	0.887	**0.822**

Table 3

Results for German problem

Configuration	Learning *Acc*	Learning *Ave*	Test *Acc*	Test *Ave*
Acc+biased	0.797	0.681	**0.726**	0.591
Ave+biased	0.794	0.759	0.715	0.633
Acc+unbiased	**0.802**	0.694	0.721	0.596
Ave+unbiased	**0.802**	**0.777**	0.719	**0.678**

Table 4

Results for Phoneme problem

Configuration	Learning *Acc*	Learning *Ave*	Test *Acc*	Test *Ave*
Acc+biased	**0.825**	0.781	**0.815**	0.768
Ave+biased	0.803	0.824	0.792	0.811
Acc+unbiased	0.821	0.778	0.810	0.764
Ave+unbiased	0.803	**0.826**	0.791	**0.812**

Table 5

Results for Segment problem

Configuration	Learning *Acc*	Learning *Ave*	Test *Acc*	Test *Ave*
Acc+biased	0.912	0.900	0.912	0.900
Ave+biased	0.909	0.896	0.909	0.896
Acc+unbiased	0.909	0.895	0.909	0.895
Ave+unbiased	0.911	0.903	0.911	0.903

Table 6

Results for Satimage problem

Configuration	Learning *Acc*	Learning *Ave*	Test *Acc*	Test *Ave*
Acc+biased	0.837	0.756	0.829	0.748
Ave+biased	0.838	**0.767**	0.830	**0.757**
Acc+unbiased	0.836	0.755	0.826	0.745
Ave+unbiased	0.837	**0.763**	0.828	**0.755**

Table 7

Confusion matrix for Page-blocks, *Acc*+biased

	Predicted 1	Predicted 2	Predicted 3	Predicted 4	Predicted 5	Unknown
True 1	**487,26**	2,80	0,06	0,53	0,60	0,03
True 2	4,33	**28,03**	0,00	0,16	0,33	0,03
True 3	**1,83**	0,00	0,93	0,00	0,00	0,03
True 4	3,00	0,13	0,00	**5,36**	0,13	0,06
True 5	**9,83**	0,00	0,13	0,00	1,50	0,03

Table 8

Confusion matrix for Page-blocks, *Ave*+biased

	Predicted 1	Predicted 2	Predicted 3	Predicted 4	Predicted 5	Unknown
True 1	**432,06**	22,16	5,96	10,93	20,06	0,10
True 2	1,36	**29,66**	0,30	0,43	1,13	0,00
True 3	0,90	0,00	**1,53**	0,03	0,33	0,00
True 4	0,53	0,53	0,13	**6,83**	0,63	0,03
True 5	1,86	0,80	0,23	0,26	**8,30**	0,03

Table 9

Confusion matrix for Page-blocks, *Ave*+unbiased

	Predicted 1	Predicted 2	Predicted 3	Predicted 4	Predicted 5	Unknown
True 1	**437,83**	21,56	2,73	12,23	16,86	0,06
True 2	1,43	**29,80**	0,03	0,50	1,06	0,06
True 3	0,53	0,00	**2,00**	0,00	0,23	0,03
True 4	0,30	0,13	0,16	**7,43**	0,63	0,03
True 5	1,73	0,63	0,23	0,26	**8,63**	0,00

From these tables one may see that using the *Ave* measure together with unbiased class numbering allows the classifying of most of the class instances correctly, although the accuracy on the majority class, as well as the overall accuracy becomes lower. From the comparison of tab. 8 and 9, it can be seen that although the unbiased procedure does not change significantly the accuracy on the majority class, for classes 3 and 4 there is a significant change in the average number of correctly classified instances.

Conclusion. In this paper an improvement for the hybrid evolutionary genetics-based classifier forming algorithm was proposed, which allowed a significant improvement in testing *Ave* values for most of the tested unbalanced datasets. This improvement balances the class numbering procedure by adding information about the number of class instances so that every class has the same chance to be in the right side of the rule, only depending on the membership values. Together with the *Ave* measure in the fitness function, the unbalanced class numbering provides the most balanced results on both test and learning samples. Moreover, this improvement does not influence balanced datasets and does not change the accuracy on them. Further improvements for the more adequate classification of unbalanced datasets for this algorithm may include the using of more complex quality measures and the developing of an unbiased rule weighting procedure.

Acknowledgment. This work was supported by the Ministry of Education and Science of the Russian Federation, Project 140/14.

Благодарности. Работа поддержана Министерством образования и науки Российской Федерации, НИР 140/14.

References

1. Bhowan U. Genetic Programming for Classification with Unbalanced Data. Victoria University of Wellington. 2012, 270 pp.

2. Fernández A., García S., Jesus M., Herrera F. A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced datasets. *Fuzzy Sets and Systems.* 2008, vol. 159, p. 2378–2398.

3. Ishibuchi H., Yamamoto T. Rule weight specification in fuzzy rule-based classification systems. *IEEE Trans. on Fuzzy Systems*. 2005, vol. 13, no. 4, p. 428–435.

4. Ishibuchi H., Yamamoto T. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems.* 2004, vol. 141, p. 59–88.

5. Wang L., Mendel J. Generating fuzzy rules by learning from examples. *IEEE Transactions on systems, man and cybernetics*. 1992, vol. 22, no. 6, p. 1414–1427.

6. Semenkina M. E. [Self-adaptive evolutionary algorithms for design of intelligent data analysis information technologies]. *Iskusstvennyy intellekt i prinyatie resheniy.* 2013, no. 1, p. 13–23 (In Russ.).

7. Alcala R., Alcala-Fdez J., Herrera F., Otero J. Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation. *International Journal of Approximate Reasoning*. 2007, no. 44, p. 45–64.

8. Semenkin E., Semenkina M. Self-configuring genetic algorithm with modified uniform crossover operator. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2012, 7331 LNCS (PART 1), p. 414–421.

9. Ishibuchi H., Mihara S., Nojima Y. Parallel Distributed Hybrid Fuzzy GBML Models With Rule Set Migration and Training Data Rotation. *IEEE Transactions on fuzzy systems*. 2013, vol. 21, no. 2, p. 355–368.

10. Akhmedova S. A., Semenkin E. S. Co-Operation of Biology Related Algorithms Meta-Heuristic in ANN-Based Classifiers Design. *Proceedings of the World Congress on Computational Intelligence (WCCI'14).* 2014, p. 867–872.

11. Khritonenko D. I., Semenkin E. S. Distributed Self-Configuring Evolutionary Algorithms For Artificial Neural Networks Design. *Vestnik SibGAU.* 2013, no. 4 (50), p. 112–116 (In Russ.).

12. Akhmedova S. A., Semenkin E. S., Gasanova T., Minker. W. Co-Operation of Biology Related Algorithms for Support Vector Machine Automated Design. *Engineering and Applied Sciences Optimization (OPT-i'14).* 2014, p. 1831–1837.

13. Brester C., Semenkin E. Development of adaptive genetic algorithms for neural network models multicriteria design. *Vestnik SibGAU.* 2013, no. 4 (50), p. 99–103 (In Russ.).

14. Asuncion A., Newman D. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, 2007, Available at: http://www.ics.uci.edu/~mlearn/MLRepository.html.

15. Ishibuchi H., Nojima Y. Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. *International Journal of Approximate Reasoning*. 2007, no. 44, p. 4–31.

Библиографические ссылки

1. Bhowan U. Genetic Programming for Classification with Unbalanced Data. Victoria University of Wellington, 2012, 270 pp.

2. A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets / A. Fernández [et al.] // Fuzzy Sets and Systems. 2008. № 159. P. 2378–2398.

3. Ishibuchi H., Yamamoto T. Rule weight specification in fuzzy rule-based classification systems // IEEE Trans. on Fuzzy Systems. 2005. Vol. 13, no. 4. P. 428–435.

4. Ishibuchi H., Yamamoto T. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining // Fuzzy Sets and Systems. 2004. № 141. P. 59–88.

5. Wang L., Mendel J. Generating fuzzy rules by learning from examples // IEEE Transactions on systems, man and cybernetics. 1992. Vol. 22, No. 6. Pp. 1414–1427.

6. Семенкина М. Е. Самоадаптивные эволюционные алгоритмы проектирования информационных технологий интеллектуального анализа данных // Искусственный интеллект и принятие решений. 2013. № 1. С. 13–23.

7. Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation / R. Alcala [et al.] // International Journal of Approximate Reasoning. 2007. № 44. Pp. 45–64.

8. Semenkin E., Semenkina M. Self-configuring genetic algorithm with modified uniform crossover operator // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2012. 7331 LNCS (PART 1). P. 414–421.

9. Ishibuchi H., Mihara S., Nojima Y. Parallel Distributed Hybrid Fuzzy GBML Models With Rule Set Migration and Training Data Rotation // IEEE Transactions on fuzzy systems. 2013. Vol. 21, No. 2. P. 355–368.

10. Akhmedova S. A., Semenkin E. S. Co-Operation of Biology Related Algorithms Meta-Heuristic in ANN-Based Classifiers Design // Proceedings of the World Congress on Computational Intelligence (WCCI'14). 2014, p. 867–872.

11. Khritonenko D. I., Semenkin E. S. Distributed Self-Configuring Evolutionary Algorithms For Artificial Neural Networks Design // Вестник СибГАУ. 2013. № 4 (50). С. 112–116.

12. Co-Operation of Biology Related Algorithms for Support Vector Machine Automated Design / S. A. Akhmedova [et al.] // Engineering and Applied Sciences Optimization (OPT-i'14). 2014, p. 1831–1837.

13. Brester C., Semenkin E. Development of adaptive genetic algorithms for neural network models multicriteria design // Вестник СибГАУ. 2013. № 4 (50). С. 99–103.

14. Asuncion A., Newman D. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, 2007. URL: http://www.ics.uci.edu/~mlearn/MLRepository.html.

15. Ishibuchi H., Nojima Y. Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning // International Journal of Approximate Reasoning. 2007. № 44. Pp. 4–31.