The object of this study is the process of automatic formation of fuzzy production rules on the basis of a training sample for solving the classification problem. The problem of automatically creating and then checking the correctness of a fuzzy inference model for a classification task is solved. The result is an automatically constructed correct database of rules for solving the classification problem. Analysis of the correctness of the knowledge base is carried out using the criteria of completeness, minimality, coherence, and consistency. To prove the completeness of the rule base, Hoare logic and the resolution method are used. The quality of the classification is assessed using such metrics as accuracy, precision, recall, f1-score. The dependence of the classification result on the size of the training sample is considered.

The expert system has the following features: the ability to learn from data; high level of accuracy; the correct knowledge base. The knowledge base is formed using the objects of the training sample on the basis of linguistic variables and term sets. A production model of knowledge representation is applied, combining the Mamdani and Takagi-Sugeno-Kang models. It is assumed that the left parts of the production rules describe combinations of the features of objects, and the right parts correspond to classes. The matrix representation of the antecedents of the rules is used. Consequents are represented as a column vector. For the automatic construction of the matrix of antecedents, it is proposed to use the Cartesian product. The formation of the consequent vector is carried out automatically using domain data and a training procedure.

The resulting expert system can be used to solve the problems of classification, clustering, data mining, and big data analysis

Keywords: expert system, fuzzy logic, fuzzy classification, Hoare logic, resolution method

UDC 621.391
DOI: 10.15587/1729-4061.2022.268908

AUTOMATIC CONSTRUCTION OF A FUZZY SYSTEM WITH A MATRIX REPRESENTATION OF RULES AND A CORRECT KNOWLEDGE BASE

Danylo Yehoshkin
Corresponding author
Postgraduate Student*
E-mail: KnightDanila@i.ua
Natalia Guk
Doctor of Physical and Mathematical Sciences, Professor, Head of Department*
*Department of Computer Technology
Oles Honchar Dnipro National University
Gagarina ave., 72, Dnipro, Ukraine, 49010

Received date 27.09.2022
Accepted date 29.11.2022
Published date

How to Cite: *Yehoshkin, D., Guk, N. (2022). Automatic construction of a fuzzy system with a matrix representation of rules and a correct knowledge base. Eastern-European Journal of Enterprise Technologies, 6 (4 (120)), 14–22. doi: https://doi.org/10.15587/1729-4061.2022.268908*

1. Introduction

It is known that most modern approaches to machine learning have had a great impact on many industries [1]. The development of technologies has made it possible to widely apply artificial intelligence systems to support decision-making in such areas of knowledge as economy, industry, medicine, science, trade, construction, transport. Domain information is a large amount of data that needs to be processed to make the right decisions. Therefore, to solve a wide class of practical problems, such methods of artificial intelligence as fuzzy logic, neural networks, genetic algorithms, and others are used. This avoids the use of accurate mathematical models based on the apparatus of mathematical equations and classical decision theory. With the help of artificial intelligence, systems are being created to simulate the activities of experts in various fields. The tasks of control, identification, modeling of complex physical phenomena, classification, pattern recognition are successfully solved using elements of fuzzy logic [1]. The use of fuzzy logic is due to the fact that it intuitively corresponds to the process of human reasoning under conditions of fuzziness and incompleteness of the conditions of the problem. Since the system of production rules can be incomplete, incoherent, and contradictory, it is necessary to check this system for correctness after its formation.

A number of works consider various models for the representation of production knowledge bases and examine the correctness of the expert system after its formation [2–7]. However, the proposed approaches for checking the correctness of the rule base do not allow for automatic verification, taking into account information about the object of study, which predetermines the need for theoretical and practical research in this field of knowledge.

2. Literature review and problem statement

Automatic formation of the production base of rules in expert systems and proof of its correctness is an urgent task since it makes it possible to ensure the simplicity of the development of such systems and the high quality of logical inference. In the literature, different approaches are developing to solve such problems.

To analyze the reliability of the knowledge base of fuzzy logical inference, work [2] considers the representation of the system of rules in the form of a meta graph. At the same time, a fuzzy knowledge base is represented using the Mamdani model. After the verification procedure, the database of rules must satisfy the following conditions: non-redundancy,

linguistic consistency, absence of looping, and linguistic completeness. It is proposed to carry out static verification of bases of fuzzy knowledge on the basis of the structure of the meta graph by finding looping in the structure of the graph. The paper considers the knowledge base built by the expert; it makes sense to test this approach on automatically generated knowledge bases with a large number of rules.

Paper [3] deals with the issue of automating the verification of the correctness of the knowledge base of product rules. A quantum model of encoding symbols of the multivalued alphabet using cubic calculus is proposed. The Mamdani algorithm is used as the output algorithm. A computer program is being developed that makes it possible to perform a formal check of the database of production rules for correctness using the resolution method. Set-theoretic operations on alphabet characters are reduced to corresponding bitwise logical operations on their qubits. The disadvantages of this approach include the following: input linguistic variables must have the same number of terms, and the terms of different input linguistic variables must have the same ranges of values. In addition, the production rules must be fully defined, i.e., any combination of term values of input linguistic variables corresponds to a certain value of the terms of the output linguistic variable.

Acyclic graphs are also used to represent the knowledge base. In work [4], an algorithm for constructing a directed acyclic graph is used to accumulate large sets of knowledge in real time. Subsequent translation of the graph is carried out using a specialized language TLC (Target Language Compiler). The paper proposes to check the knowledge base for correctness at the translation stage. Both classical methods are used, using disjunctive and conjunctive normal forms, and using the normal form of negation NNF (Negation Normal Form). This allows the user to display counterexamples if the knowledge base is contradictory or incomplete. The paper discusses the construction of a knowledge base in real time but does not consider the effect of retraining.

If it is necessary to represent data in tabular form for the application of matrix and vector processing operations, the decision-making table (DMT) is used [5]. In the subsequent work of the author, changes were made to the algorithm of formation, but the representation with the help of decision-making tables remained unchanged [6]. A big advantage of DMT is the visibility and convenient representation of data for the expert in the process of analyzing and filling out the database of rules. Another advantage is that the main operations of the production cycle are logical (vector and matrix) operations that make it possible to get a high processing speed. This makes it possible to use the tabular model of knowledge representation very effectively. To check the knowledge base, a disjunctive normal form is used, which is convenient for automatic proof of theorems. The proof process is based on the logic of propositions and the logic of predicates. In works [5, 6], when using DMT, antecedents and consequents are represented compositionally, it is possible to consider antecedents as a separate matrix, and consequents as a vector.

To generate a database of Mamdani-type rules, paper [7] proposes the determination of optimal sequences based on the use of multiagent optimization algorithms, in particular ants. This approach makes it possible to effectively generate rule bases in the following cases:

1) if there is insufficient amount of initial information;

2) with a sufficiently large number of rules, for which the compilation of a database of fuzzy rules based on the knowledge of experts is not always effective;

3) with different levels of qualification of experts.

This does not address the question of a sufficient number of generation cycles and time to achieve the required level of model accuracy.

These approaches in [2–7] significantly increase the efficiency of using intelligent systems. Make it possible to form and verify the database of product rules, however:

1) impose limits on the number of terms of linguistic variables;

2) require the presence of an expert to analyze the verification work;

3) impose limits on the number of logical conclusions.

The task of classification is considered. To solve the problem, an approach based on fuzzy logic is used.

Information about the subject area is represented in the form of a production model under the assumption that the left parts of the product rules describe combinations of features of objects, and the right ones correspond to classes.

In the formation of the left parts of products, the attributes of objects from some finite (quasi-finite, if it is permissible to replenish the model) set are used, the conjunction of the true values of which determines the conditions for the applicability of the product. On the right side, you specify classes from some finite allowable set of feature classes.

To ensure the reliability of inference based on the formulated knowledge base, it is necessary that the knowledge base has the properties of completeness, minimality (not redundancy), consistency, and coherence. An approach is being developed that provides validation for these properties.

3. The aim and objectives of the study

The purpose of the study is to develop an approach to the automatic generation of fuzzy production rules based on a training sample for solving the classification problem with the subsequent verification of the correctness of the constructed model. This will create fuzzy expert systems that can learn from test datasets. The production bases of the rules of such trained systems will allow experts to find hidden dependences between the features of objects and their classes.

To accomplish the aim, the following tasks have been set:

– to create a production model for the representation of knowledge about objects of the subject area, combining the Mamdani and Takagi-Sugeno-Kang models, based on the results of observation of objects of the subject area:

– to develop an algorithm for automatic generation and choose a way to represent the database of rules;

– to develop an approach to verify the correctness of the rule base using Hoare logic and the resolution method. Apply Simplify software for automatic consistency proof;

– to select metrics to assess the quality of the resulting fuzzy base of rules;

– to apply the developed approach to check the correctness of the rule base for solving the classification problem.

4. The study materials and methods

The object of the study is the process of automatic formation of fuzzy production rules on the basis of a training sample for solving the classification problem. The task is stated in the following way: it is required to build fuzzy production rules for the MISO system on the basis of the training sample. At the same time, the basis of the hypothesis

of the formation of rules is that the rules are built on the basis of linguistic variables and their term sets.

To construct the antecedents of the rules, the attributes of objects from the finite term set are used, the conjunction of the true values of which determines the conditions for the applicability of the product. The antecedents of the rules are represented as a matrix, this is necessary for the automatic formation of the matrix using the Cartesian product. For consequents, you specify classes from some finite allowable set of feature classes. Rule consequents are represented as a column vector, the values of the elements of the vector depend on the matrix of antecedents and the objects of the training sample. The formation of the vector of consequents is carried out using the training procedure of the knowledge base.

To check the obtained knowledge base for correctness, criteria are used: completeness, minimality, coherence, and consistency. To prove completeness, Hoare logic, the resolution method, and the Simplify application are used, which automatically check the system for consistency using first-order logic.

The developed approach is applied to solving the classification problem, the quality assessment of the resulting fuzzy base of rules is carried out using accuracy, precision, recall, f1-score metrics.

5. Results of the development of a methodology for analyzing the correctness of the product model

5. 1. Production model of knowledge representation and inference method

Domain objects in the classification problem are described by a feature system $\tilde{k}_1,\tilde{k}_2,...,\tilde{k}_L$ for each object $x_i \in X$, the features correspond to the linguistic variables k_1, k_2, ..., k_L, wherein each linguistic variable contains a term set A_{lt}, where l is the number of feature $l=\overline{1,L}$, and $t=\overline{1,T_l}$; T_l is the number of terms for feature k_l. The output variable y corresponds to the class to which the object of the subject area belongs. To solve the classification problem, we will build a production model of knowledge representation that combines the Mamdani and Takagi-Sugeno-Kang models [8].

Rules P_p of the MISO type take the form:

$$\Pi_p: \text{IF } k_1 \text{ is } \tilde{a}_{p1} \text{ AND } k_2 \text{ is } \tilde{a}_{p2} \text{ AND } ... \text{ AND } k_L \text{ is } \tilde{a}_{pL}, \quad (1)$$

$$\text{THEN } y \text{ is } f(k_1,...,k_L,p)\cdot c_p,$$

where k_1, k_2,..., k_L are the input linguistic variables; $\tilde{a}_{pl}$ is a variable that takes fuzzy values A_{lt} for the rule p; A_{lt} is a fuzzy term set of the linguistic variable k_l for the rule p; t is the number of term for k_l, $t=\overline{1,T_l}$; T_l is the number of terms for the feature k_l; y is the output variable; p is the rule number in the rule database, $p=\overline{1,P}$; P is the total number of rules P; c_p is a variable that takes fuzzy values of C_m for the rule p; C_m is the label of the fuzzy class to which the object $x_i \in X$ belongs; m is the number of class C, $m=\overline{1,M}$; M is the number of classes C.

The real, non-negative, normalized function $f(k_1, ..., k_L, p)$ is continuous over the interval $[a, b]$ and is used as a weighting factor for the consequent of the rule. It determines the degree to which the output variable y belongs to the term set C_m and is calculated as follows:

$$f(k_1,...,k_L,p)=\frac{\sum_{l=1}^{L}\mu_{a_{pl}}(k_l)}{L}. \quad (2)$$

The fuzzification procedure is defined as follows:

$$a_{pl}=\int_{\underline{k}_l}^{\overline{k}_l} < \mu_{a_{pl}}(k_l) \,|\, k_l > \mathrm{d}k, \quad (3)$$

where $\mu_{a_{pl}}(k_l)$ is the function of attributing a clear value of the input variable k_l to a fuzzy term a_{pl}.

To automate the process of building rules, it is convenient to represent the antecedents of the rules in the form of a matrix A:

$$A=\begin{pmatrix} a_{11} & a_{12} & ... & a_{1l} \\ a_{21} & a_{22} & ... & a_{2l} \\ ... & ... & ... & ... \\ a_{P1} & a_{P2} & ... & a_{PL} \end{pmatrix}, \quad (4)$$

where a_{pl} is an element of the matrix that takes the value $\tilde{a}_{pl}$ of a variable from the rule system, and is equal to the fuzzy value A_{lt} for the rule p; P – number of rules; L is the dimensionality of the feature vector $\overline{K}=(k_1,k_2,...,k_L)$ of the object $x_i \in X$.

We will represent the rule consequents as a column vector $\vec{C}$:

$$\vec{C}=\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ ... \\ c_p \end{pmatrix}. \quad (5)$$

Further, the fuzzy set of classes $\tilde{C}$ that correspond to the output variable y is defined as:

$$\tilde{C}_m = < \sum_{p=1}^{P} f(k_1,...,k_L,p) \,|\, C_m \in \Pi_p, C_m >.$$

Thus, the set $\tilde{C}$ will consist of ordered pairs $<f_m|C_m>$ such that:

$$f_m = \sum_{p=1}^{P} f(k_1,...,k_L,p) \,|\, C_m \in \Pi_p,$$

where the pair $<f_m|C_m>$ determines the degree of belonging f_m of object y_i to class C_m.

For $\tilde{C}$ defuzzification, the Mamdani's algorithm with the centroid method is used. The fuzzy set $\tilde{C}$ of classes corresponding to the input object X is defined as:

$$\tilde{C}=\underset{m=\overline{1,M}}{\mathrm{agg}}\left(\int_{\underline{C}}^{\overline{C}} \mathrm{imp}\left(f_m\cdot\mu_{C_m}(C),\mu_{C_m}(C)/C\right)\mathrm{d}C\right),$$

wherein imp implication and agg aggregation operations are implemented by finding min and max, respectively.

$$C^*=\int_{\underline{C}}^{\overline{C}} C\cdot\mu_{\tilde{C}}(C)\mathrm{d}C \,/\, \int_{\underline{C}}^{\overline{C}} \mu_{\tilde{C}}(C)\mathrm{d}C.$$

The clear value of the logical inference C^* is determined as a result of the defuzzification of the fuzzy set $\tilde{C}$ by the center of gravity method.

5. 2. Algorithm of automatic formation of the rule database

Below is the process of automatic formation of a knowledge representation model in the form of a matrix of antecedents A and a vector of consequents $\vec{C}$.

For each linguistic variable k_l, there is a finite (quasi-finite, if it is permissible to replenish the model) fuzzy term set $A_l=(A_{l1}, A_{l2}, A_{l3}, ..., A_{lt})$, which defines the finite alphabet for describing the states or sets of states of the linguistic Variable.

The number of fuzzy classes C_m to which the object $x_i \in X$ belongs is also limited. Thus, the elements A_{lt} and C_m define the alphabet of the production model.

Formally, the matrix of antecedents A is given by an ordered set $<A_{11}, ..., A_{lt}>$, where each vector-row of matrix A is a set of conditions of the production rule. The vector of consequents $\tilde{C}$ consists of classes C_m and has a dimensionality equal to the number of rules, sets the correspondence between the production rule in the matrix A and the fuzzy class C_m.

The set of production rules P can be represented as: $P_p=<a_p, c_p>$, where a_p is the vector-string of the matrix A. Term-set $A_l=(A_{l1}, A_{l2}, A_{l3,...,} A_{lt})$ is ordered with respect to the semantic order and boundaries of the terms. To construct the matrix A, it is proposed to use the Cartesian product:

$$A_1 \times A_2 \times A_3 \times ... \times A_l = A^{xl}.$$

Then, this set A^{xl} can be represented as the following matrix A:

$$A = \begin{pmatrix} A_{11} & A_{21} & A_{31} & ... & A_{l1} \\ A_{12} & A_{21} & A_{31} & ... & A_{l1} \\ A_{11} & A_{22} & A_{31} & ... & A_{l1} \\ ... & ... & ... & ... & ... \\ A_{1t} & A_{2t} & A_{3t} & ... & A_{lt} \end{pmatrix}. \quad (6)$$

The number of rules P is determined from the properties of the Cartesian product and is equal to the product of the number of elements of the multiplier sets:

$$P = \prod_{l=1}^{L} |A_l| = \prod_{l=1}^{L} T_l. \quad (7)$$

The formation of the vector of consequents is carried out using the training procedure of the knowledge base. Each object from the training sample X^{Train} undergoes a fuzzification procedure $fuzz\left(x_i^{Train}\right)$, after which the $\vec{C}=\left(c_1, c_2, ..., c_p\right)$ vector is determined as follows:

$$c_p = C_m : \max\left(\underset{i}{cord}^{I^{Train}} \left(fuzz\left(x_i^{Train}\right) \in a_p, C_m \right) \right), \quad (8)$$

where *cord*() *is a* function of the power of the set.

The vector $\tilde{C}$ must contain all C_m classes, otherwise an incomplete or incorrect training sample is fed as input, and the system must be retrained on a different dataset.

5. 3. Knowledge base correctness analysis

The knowledge base should meet uniform formal requirements of correctness, namely, be complete, minimal, coherent, and consistent [3].

Completeness is understood as any incoming combination of values of the terms of input linguistic variables corresponds to a certain rule in the rule base. The minimum database of rules is the base from which none of the production rules can be removed without violating its completeness. A knowledge base is consistent if it does not contain inconsistent rules – rules with the same linguistic conditions but different conclusions. The minimality and consistency of the formed knowledge base is ensured by the properties of the Cartesian product. The vector $\tilde{C}$, built in the learning process on the basis of the training sample (8), is responsible for coherence. Each row of matrix (4) differs only in one of l subconditions since at the beginning of the formation of the matrix A the term-set $A_l=(A_{l1}, A_{l2}, A_{l3},..., A_{lt})$ is ordered relative to the semantic order and boundaries of the terms.

To automatically prove the completeness of the database of rules, Hoare logic is used [9]. Define the Hoare triple: $\{Q\}\cdot S\cdot\{R\}$, where Q is the predicate of the precondition, S is the set of commands, R is the predicate of the postcondition. By the set of commands S, we will understand the set of production rules $P_p=<a\cdot p, c_p>$:

$$S: <A, \vec{C}>.$$

A precondition in this predicate describes all possible initial values that linguistic variables can take to close the set of input values:

$$Q: \left(\underset{l=1}{\overset{L}{\wedge}} \underset{t=1}{\overset{T_l}{\vee}} k_l \in A_{lt} \right) \wedge \underset{m=1}{\overset{M}{\vee}} \text{result} = C_m.$$

The predicate $Q=T$ is true because $\underset{l=1}{\overset{L}{\forall}} k_l$ belongs to at least one A_{lt}.

The postcondition predicate describes the expected result of S:

$$R: \underset{p=1}{\overset{P}{\vee}} \left(\left(\underset{l=1}{\overset{L}{\wedge}} k_l \in a_{pl} \right) \wedge \text{result} = c_p \right).$$

If the precondition Q is met, the command S renders true the postcondition R [10] true.

This statement is proved using the predicate of the weakest precondition $WP(S, R)$ [11]:

$$Q \Rightarrow WP(S, R). \quad (9)$$

The WP predicate performs a substitution $<A, \vec{C}>$ from S to R.

$$Q \Rightarrow R_S. \quad (10)$$

To prove the truth of predicate (10), let's use the resolution method:

$$Q \models R_S,$$

$$\left(\underset{l=1}{\overset{L}{\wedge}} \underset{t=1}{\overset{T_l}{\vee}} k_l \in A_{lt} \right) \wedge \underset{m=1}{\overset{M}{\vee}} \text{result} = C_m \models$$

$$= \underset{p=1}{\overset{P}{\vee}} \left(\left(\underset{l=1}{\overset{L}{\wedge}} k_l \in a_{pl} \right) \wedge \text{result} = c_p \right) \Big|_S,$$

$$\left\{ \begin{array}{l} \left(\underset{l=1}{\overset{L}{\wedge}} \underset{t=1}{\overset{T_l}{\vee}} k_l \in A_{lt} \right), \underset{m=1}{\overset{M}{\vee}} \text{result} = C_m, \\ \neg \underset{p=1}{\overset{P}{\vee}} \left(\left(\underset{l=1}{\overset{L}{\wedge}} k_l \in a_{pl} \right) \wedge \text{result} = c_p \right) \Big|_S \end{array} \right\},$$

$$\left\{ \begin{array}{l} \left(\underset{l=1}{\overset{L}{\wedge}} \underset{t=1}{\overset{T_l}{\vee}} k_l \in A_{lt} \right), \underset{m=1}{\overset{M}{\vee}} \text{result} = C_m, \\ \neg \left(\underset{p=1}{\overset{P}{\vee}} \left(\underset{l=1}{\overset{L}{\wedge}} k_l \in a_{pl} \right) \Big|_S \wedge \underset{p=1}{\overset{P}{\vee}} \left(\text{result} = c_p \right) \Big|_S \right) \end{array} \right\},$$

$$\left\{\begin{pmatrix}\overset{L}{\underset{l=1}{\wedge}}\overset{T_l}{\underset{t=1}{\vee}} k_l \in A_{lt}\end{pmatrix}, \overset{M}{\underset{m=1}{\vee}} \text{result} = C_m, \\ \overset{P}{\underset{p=1}{\wedge}}\left(\overset{L}{\underset{l=1}{\vee}} \neg\left(k_l \in a_{pl}\right)\right)\Big|_S \vee \neg\left(\overset{P}{\underset{p=1}{\vee}} \text{result} = c_p\right)\Big|_S\right\}.$$

Since c_p take values from C_m, after substituting S, we obtain:

$$\left\{\begin{pmatrix}\overset{L}{\underset{l=1}{\wedge}}\overset{T_l}{\underset{t=1}{\vee}} k_l \in A_{lt}\end{pmatrix}, \overset{M}{\underset{m=1}{\vee}} \text{result} - C_m, \\ \overset{P}{\underset{p=1}{\wedge}}\left(\overset{L}{\underset{l=1}{\vee}} \neg\left(k_l \in a_{pl}\right)\right)\Big|_S \vee \neg\left(\overset{M}{\underset{m=1}{\vee}} \text{result} = C_m\right)\right\}.$$

Further, the rule of the resolution applies to the second and third disjuncts:

$$\left\{\begin{pmatrix}\overset{L}{\underset{l=1}{\wedge}}\overset{T_l}{\underset{t=1}{\vee}} k_l \in A_{lt}\end{pmatrix}, \overset{P}{\underset{p=1}{\wedge}}\left(\overset{L}{\underset{l=1}{\vee}} \neg\left(k_l \in a_{pl}\right)\right)\Big|_S\right\}.$$

Considering (7), we obtain:

$$\left\{\begin{pmatrix}\overset{L}{\underset{l=1}{\wedge}}\overset{T_l}{\underset{t=1}{\vee}} k_l \in A_{lt}\end{pmatrix}, \overset{\prod_{l=1}^{L} T_l}{\underset{p=1}{\wedge}}\left(\overset{L}{\underset{l=1}{\vee}} \neg\left(k_l \in a_{pl}\right)\right)\Big|_S\right\}.$$

Let's apply the law of distribution to the second disjunct and perform substitution S:

$$\left\{\begin{pmatrix}\overset{L}{\underset{l=1}{\wedge}}\overset{T_l}{\underset{t=1}{\vee}} k_l \in A_{lt}\end{pmatrix}, \overset{L}{\underset{l=1}{\vee}}\left(\overset{T_l}{\underset{t=1}{\wedge}} \neg\left(k_l \in A_{lt}\right)\right)\right\},$$

$$\left\{\begin{pmatrix}\overset{L}{\underset{l=1}{\wedge}}\overset{T_l}{\underset{t=1}{\vee}} k_l \in A_{lt}\end{pmatrix}, \neg\left(\overset{L}{\underset{l=1}{\wedge}}\overset{T_l}{\underset{t=1}{\vee}}\left(k_l \in A_{lt}\right)\right)\right\}.$$

As a result of applying the rule of resolutions to the first and second disjuncts, we obtain: {_}. The above proof shows the completeness of the postconditions based on the WP predicate.

To automatically check the rule database, we will use the Simplify software [12], which implements the resolution method to prove the truth of given predicates based on first-order logic. Formal calculus allows statements about variables, fixed functions, and predicates, which in turn extends the logic of propositions. Simplify never forms an infinite loop in the process of proof and believes that predicates whose truth cannot be proved are false [13].

Let's formulate the Hoare triple:

$$Q: \left(\overset{L}{\underset{l=1}{\wedge}}\overset{T_l}{\underset{t=1}{\vee}} k_l \in A_{lt}\right) \wedge \overset{M}{\underset{m=1}{\vee}} \text{result} = C_m;\ S: <A, \vec{C}>;$$

$$R: \overset{P}{\underset{p=1}{\vee}}\left(\left(\overset{L}{\underset{l=1}{\wedge}} k_l \in a_{pl}\right) \wedge \text{result} = c_p\right).$$

Next, to prove correctness, we will formulate predicates using the directives of the Simplify language and conduct an automatic proof.

The Simplify axiom for the first conjunct of the precondition Q will be:

$$\left(BG\_PUSH\left(\begin{pmatrix}FORALL\ (k\_i\ ai1\ ai2\ ai3\ ...\ ait),\\ \begin{pmatrix}OR\ (EQ\ k\_i\ ai1)(EQ\ k\_i\ ai2)\\ (EQ\ k\_i\ ai3)...(EQ\ k\_i\ ait)\end{pmatrix}\end{pmatrix}\right)\right).$$

The axiom for the second conjunct of the precondition Q is represented as:

$$\left(BG\_PUSH\left(\begin{pmatrix}FORALL(result\ c1\ c2\ c3\ ...\ cm),\\ \begin{pmatrix}OR\ (EQ\ result\ c1)\\ (EQ\ result\ c2)\\ (EQ\ result\ c3)...(EQ\ result\ cm)\end{pmatrix}\end{pmatrix}\right)\right).$$

After that, some body needs to translate Q and R into the Simplify language. Using the introduced axioms and predicates Q, S, R, automatic proof of the correctness of the knowledge base is organized.

5. 4. Selecting metrics to assess the quality of a rule base

Since an expert system with a fuzzy base of rules is used to solve the problem, the result of the classification is approximate, so it is necessary to be able to assess the quality of the result obtained. To assess the quality of the classification result, carried out using a fuzzy base of rules, the following metrics are used in the work:

$$\text{accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}};$$

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}};\ \text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}};$$

$$\text{f1-score} = \frac{2\times(\text{precision}\times\text{recall})}{\text{precision}+\text{recall}},$$

where TP – True Positive; FP – False Positive; FN – False Negative; TN – True Negative.

5. 5. Checking the correctness of the rule base for solving the classification problem

The proposed approach was applied to the problem of classifying the species population of Arctic penguins [14]. The dataset contains data on 344 individuals of three species Chinstrap, Adelie, Gentoo. The attributes of individuals: bill_length_mm – the length of the beak; bill_depth_mm is the depth of the beak; flipper_length_mm is the length of the fin; body_mass_g is body weight. For the classification shown in Fig. 3, 4, a training and test sample of 300/44 (training/test) were formed. Fig. 1 depicts the result of the program for calculating the limit distribution diagram.

Fig. 2 shows the result of checking the knowledge base for completeness using the weakest postcondition predicate and the Simplify program.

Fig. 3, 4 show the results of the classification of the developed program before and after the training of a fuzzy knowledge base.

The dependence of the classification result on the size of the training sample before/after training is considered. To do this, a training and test sample of 200/144 (training/test) were formed. Fig. 5, *a* presents a matrix of contradictions before training; Fig. 5, *b* – after training the system. Tables 1, 2 give values of quality metrics before and after training, respectively. The contradiction matrices in Fig. 5, 6 are built by the developed program in Python.

To analyze the impact of the learning process on the classification result, the sample sizes were changed 250/94 (training/test). Fig. 6, *a* shows the matrix of contradictions before training; Fig. 6, *b* – after training the system. Tables 3, 4 give values of quality metrics before and after training, respectively.

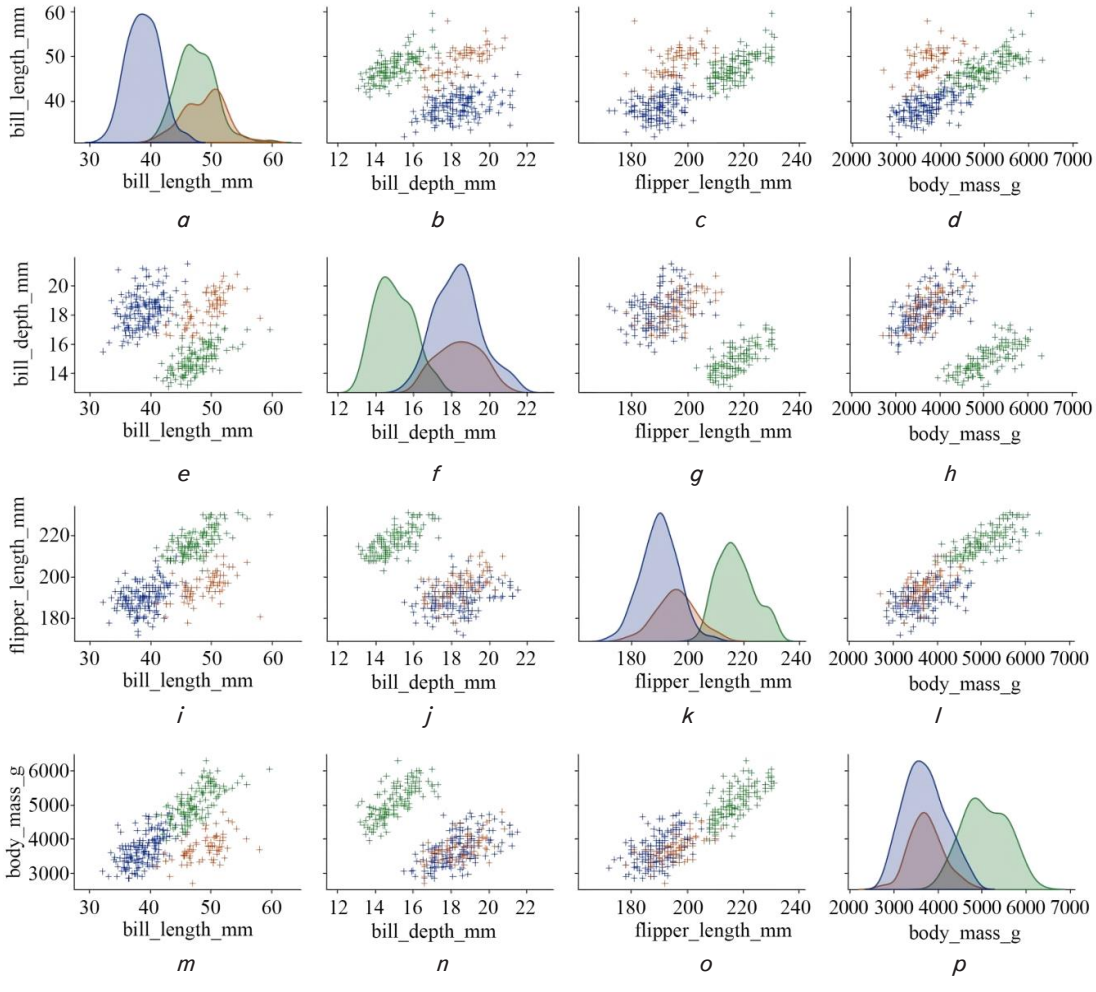


Fig. 1. Class distribution limit chart: ● – Chinstrap, ● – Adelie, ● – Gentoo

```
...
(OR
(EQ k_L a_L1) (EQ k_L a_L2) (EQ k_L a_L3)
(EQ k_L a_L4) (EQ k_L a_L5) (EQ k_L a_L6)
(EQ k_L a_L7) (EQ k_L a_L8) (EQ k_L a_L9)
...
(EQ k_L a_Lt)
)
)
(OR
(EQ result c1)
(EQ result c2)
(EQ result c3)
...
(EQ result cm)
)
)
;R_S
(OR
(AND (EQ k_1 a_11)(EQ k_2 a_21)(EQ k_3 a_31)...(EQ k_L a_Lt) (EQ result c1))
(AND (EQ k_1 a_12)(EQ k_2 a_21)(EQ k_3 a_31)...(EQ k_L a_Lt) (EQ result c1))
(AND (EQ k_1 a_11)(EQ k_2 a_22)(EQ k_3 a_31)...(EQ k_L a_Lt) (EQ result c1))
...
(AND (EQ k_1 a_1t)(EQ k_2 a_2t)(EQ k_3 a_3t)...(EQ k_L a_Lt) (EQ result cm))
)
)
1: Valid.
```

Fig. 2. The result of the completeness check, «Valid» is the truth

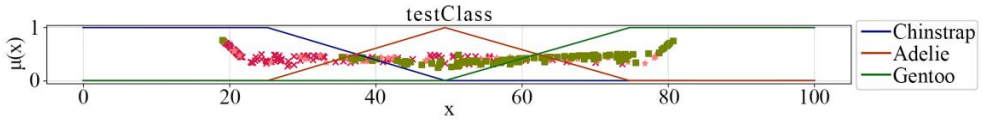


Fig. 3. Before training

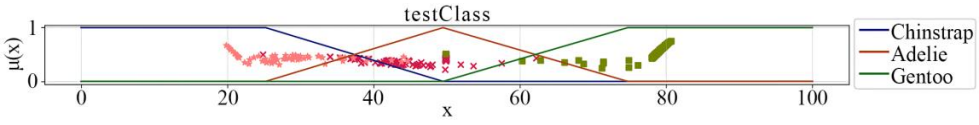
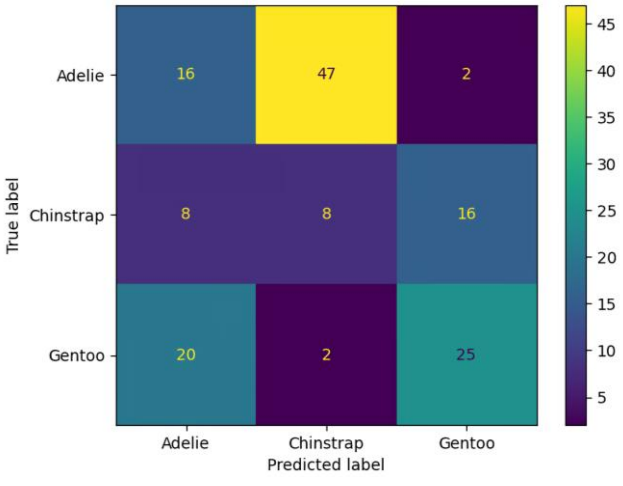
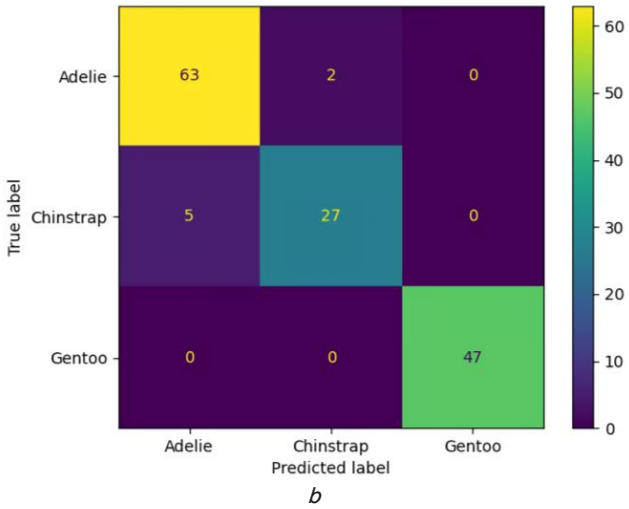


Fig. 4. After training

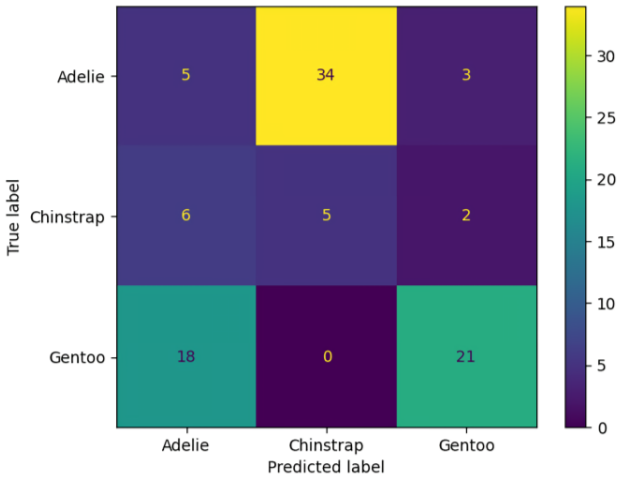


a

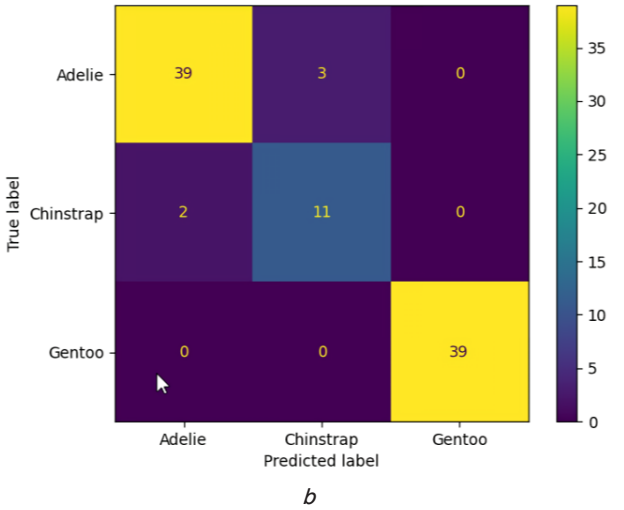


b

Fig. 5. Contradiction matrices — 200/144 (training/test): *a* — before training; *b* — after training



a



b

Fig. 6. Contradiction matrices — 250/94 (training/test): *a* — before training; *b* — after training

Table 1

Before training: 200/144 (training/test)

Species	precision	recall	f1-score	support
Adelie	0.36	0.25	0.29	65
Chinstrap	0.14	0.25	0.18	32
Gentoo	0.58	0.53	0.56	47
avg accuracy	0.34			

Table 2

After training: 200/144 (training/test)

Species	precision	recall	f1-score	support
Adelie	0.93	0.97	0.95	65
Chinstrap	0.93	0.84	0.89	32
Gentoo	1.00	1.00	1.00	47
avg accuracy	0.90			

Table 3

Before training: 250/94 (training/test)

Species	precision	recall	f1-score	support
Adelie	0.17	0.12	0.14	42
Chinstrap	0.13	0.38	0.19	13
Gentoo	0.81	0.54	0.65	39
avg accuracy	0.33			

Table 4

After training: 250/94 (training/test)

Species	precision	recall	f1-score	support
Adelie	0.95	0.93	0.94	42
Chinstrap	0.79	0.85	0.81	13
Gentoo	1.00	1.00	1.00	39
avg accuracy	0.93			

The dependence of the classification result on the size of the training sample before/after training is considered. Samples 200/144 and 250/94 (training/test) were used.

6. Discussion of pre- and post-training classification results as a function of training sample size

Analysis of the limit distribution diagram (Fig. 1) showed that the Adelie and Chinstrap species are sufficiently related, and their classification can be difficult. This is especially noticeable for the attributes of body_mass_g and bill_depth_mm – Fig. 1, *g*, *i*.

Fig. 2 shows the result of checking the knowledge base for completeness using the predicate of the weakest postcondition. The «Valid» result shows the truth of the predicate $Q=>WP(S,R)$, and, consequently, the completeness of the knowledge base.

Applying the acquired knowledge base to training in Fig. 3 shows that the classes are not defined and the values in the results are mixed. There is no grouping of objects by class. But, after training in Fig. 4, you can see that the data are grouped, and each class is within the boundaries of its membership function. It should be noted that all 344 objects are present on the chart at the same time.

Next, the influence of the size of the training sample on the classification result and the quality of the expert system was considered. To do this, a training and test sample of 200/144 (training/test) was formed. Before studying, Fig. 5, *a* presents a matrix of contradictions and analysis of metrics in Table 1. According to the results, you can see that the accuracy is quite low, 34 %. After training, Fig. 5, *b* and the analysis of metrics in Table 2 demonstrate that the accuracy of the system has increased to 90 %, and the number of incorrect answers in the matrix of contradictions has significantly decreased.

Next, a training and test sample was formed in the amount of 250/94 (training/test). According to Fig. 6, *a* and Table 3, one can also see a low accuracy of 33 %, but, after training, Fig. 6, *b*, the accuracy increased to 93 % like the rest of the indicators in Table 4.

The analysis of Tables 1–4 reveals that increasing the volume of the training sample makes it possible to obtain an acceptable classification accuracy for the samples, as well as to adjust the vector of the consequents to solve the classification problem.

Our results differ from those in works [2–7] in that a method has been developed for automatically constructing a system of rules using the Cartesian product and the matrix representation of antecedents and consequents of rules, followed by automatic proof of the correctness of the rule system based on the method of resolutions using Hoare logic and the weak precondition predicate $Q=>WP(S,R)$.

The developed approach will make it possible to find hidden dependences between a set of input parameters and output data of the model, automatically create a database of rules of the expert system and prove its correctness. This approach can be used to solve practical tasks of multiclass classification, control, decision-making, clustering, data mining.

The limitations of the application of this method include the fact that in the case of classification of objects whose features are strongly related, the necessary accuracy of classification is not achieved.

The disadvantage of this approach is the significant size of the rule base. To eliminate this problem, it is recommended to use the methods of reducing the knowledge base, which will make it possible to apply the proposed approach to the problems of classifying objects with a large number of attributes.

7. Conclusions

1. The paper reports the constructed production model of knowledge representation that combines the Mamdani and Takagi-Sugeno-Kang models and uses the function as a weighting factor for the rule's consequent. This has made it possible to determine the degree of influence of each rule on the output result of the production system.

2. A distinctive feature of the constructed model is the automatic formation of a database of rules based on the data of the domain model. An antecedent matrix is created using features of objects from some finite set using a Cartesian product. A training sample is used to form the consequent vector. Thanks to this, it is possible to create a database of rules and configure the model under an automatic mode.

3. An approach has been developed to check the correctness of production rules based on the resolution method using Hoare logic and the weak precondition predicate $Q=>WP(S,R)$. This approach makes it possible to automate the verification of the correctness of the rule base at the training stage.

4. To assess the quality of solving the classification problem, accuracy, precision, recall, f1-score metrics were used. High quality scores were obtained for the trained system (accuracy=93 %, precision=0.91, recall=0.93, f1-score=0.92) while, before training, the quality of the model was significantly lower (accuracy=33 %, precision=0.37, recall=0.35, f1-score=0.33). Before training, the accuracy of the model was insignificant, 33 %; after training, the accuracy of the model reaches 93 %. It is established that increasing the volume of training sample makes it possible to get an acceptable classification accuracy, as well as adjust the vector of consequents to solve the classification problem. Also, after training, the quality assessments of the model increase: accuracy, precision, recall, f1-score.

5. The proposed approach is used to solve the classification problem. The correctness of the automatically constructed fuzzy database of rules for solving the classification problem was checked. An assessment of the quality of the obtained fuzzy base of rules was carried out, as a result of which the accuracy of the model reaches 93 %. The analysis of the results of the computational experiment was carried out; it was established that the size of the training sample affects the classification result and the quality of the expert system.

Conflicts of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

The study was conducted without financial support.

Data availability

All data are available in the main text of the manuscript.

References

1. Zadeh, L. A., Abbasov, A. M., Yager, R. R., Shahbazova, S. N., Reformat, M. Z. (Eds.) (2014). Recent Developments and New Directions in Soft Computing. Studies in Fuzziness and Soft Computing. doi: https://doi.org/10.1007/978-3-319-06323-2
2. Ternovoi, M. Yu., Shtohryna, E. S. (2015). Formalnaia spetsyfykatsyia svoistv baz nechetkykh znanyi Mamdany na osnove metahrafa. Visnyk Kharkivskoho natsionalnoho universytetu imeni V. N. Karazina. Seriya: Matematychne modeliuvannia. Informatsiyni tekhnolohiyi. Avtomatyzovani systemy upravlinnia, 27, 157–171. Available at: http://nbuv.gov.ua/UJRN/VKhIMAM_2015_27_17
3. Krivulya, G. F., Shkil', A. S., Kucherenko, D. E. (2013). Analiz korrektnosti produkcionnyh pravil v sistemah nechetkogo logicheskogo vyvoda s ispol'zovaniem kvantovyh modelej. ASU i pribory avtomatiki, 165, 42–53. Available at: https://openarchive.nure.ua/server/api/core/bitstreams/eeb5b66f-7eb9-4db4-b045-0e774308ee6d/content
4. Darwiche, A., Marquis, P. (2002). A Knowledge Compilation Map. Journal of Artificial Intelligence Research, 17, 229–264. doi: https://doi.org/10.1613/jair.989
5. Sugiura, A., Riesenhuber, M., Koseki, Y. (1993). Comprehensibility Improvement of Tabular Knowledge Bases. AAAI-93 Proceedings, 716–721. Available at: https://www.aaai.org/Papers/AAAI/1993/AAAI93-107.pdf
6. Sugiura, A., Koseki, Y. (1995). Comprehensibility Improvement of Tabular Knowledge Bases. Journal of the Japanese Society for Artificial Intelligence, 10 (4), 628–635. doi: https://doi.org/10.11517/jjsai.10.4_628
7. Kondratenko, Y. P., Kozlov, A. V. (2019). Generation of Rule Bases of Fuzzy Systems Based on Modified Ant Colony Algorithms. Journal of Automation and Information Sciences, 51 (3), 4–25. doi: https://doi.org/10.1615/jautomatinfscien.v51.i3.20
8. Zheldak, T. A., Koriashkina, L. S. (2020). Nechitki mnozhyny v systemakh upravlinnia ta pryiniattia rishen. Dnipro: NTU «DP», 222–227.
9. Hoare, C. A. R. (1969). An axiomatic basis for computer programming. Communications of the ACM, 12 (10), 576–580. doi: https://doi.org/10.1145/363235.363259
10. Gries, D. (1981). The Predicate Transformer wp. The Science of Programming, 108–113. doi: https://doi.org/10.1007/978-1-4612-5983-1_8
11. Dijkstra, E. W. (1975). Guarded commands, nondeterminacy and formal derivation of programs. Communications of the ACM, 18 (8), 453–457. doi: https://doi.org/10.1145/360933.360975
12. Simplify. ESC/Java2 Summary. Available at: https://www.kindsoftware.com/products/opensource/escjava2
13. Khizha, A. L., Vysokopoyasnyj, I. G. (2017). Avtomaticheskaya proverka semanticheskoy pravil'nosti resheniy zadach po programmirovaniyu. Pytannia prykladnoi matematyky i matematychnoho modeliuvannia, 17, 234–246. Available at: http://nbuv.gov.ua/UJRN/Ppmmm_2017_17_29
14. Gorman, K. B., Williams, T. D., Fraser, W. R. (2014). Ecological Sexual Dimorphism and Environmental Variability within a Community of Antarctic Penguins (Genus Pygoscelis). PLoS ONE, 9 (3), e90081. doi: https://doi.org/10.1371/journal.pone.0090081