УДК 621.391

Н. А. Гук, Д. І. Єгошкін
Дніпровський національний університет ім. Олеся Гончара

АЛГОРИТМ КЛАСИФІКАЦІЇ НА БАЗІ НЕЧІТКОЇ ЛОГІКИ З РОЗШИРЮВАНОЮ КІЛЬКІСТЮ ВИВОДІВ

Розглядається задача автоматичної генерації бази знань, що складається з продукційних правил, для об'єктів навчальної вибірки з використанням методів нечіткої логіки і правила порівняння значень вихідних змінних. Пропонується алгоритм формування нечітких продукційних правил.

Рассматривается задача автоматической генерации базы знаний, состоящей из продукционных правил, для объектов обучающей выборки с использованием методов нечеткой логики и правила сравнения значений выходной переменной. Предлагается алгоритм формирования нечетких продукционных правил.

The problem of automatic generation of a knowledge base which consist of production rules, for training sample objects using fuzzy logic methods and a rule for comparing the values of an output variable is considered. An algorithm for the formation of fuzzy production rules is proposed.

Ключові слова: *штучний інтелект, нечітка логіка, відображення, класифікація, база знань, експертна система, мова C/C ++, мова JavaScript, JSON.*

Вступ. Розвиток обчислювальних технологій і апаратної частини комп'ютера дозволив широко застосовувати експертні системи для підтримки прийняття рішень в таких галузях знань, як економіка, промисловість, медицина. Сформульовані бази знань і розроблені алгоритми логічного виведення дозволяють акумулювати великий обсяг знань, отриманих від експертів. За допомогою таких систем можна усувати труднощі формалізації знань про технологічні процеси, організувати розпізнавання нестандартних та аварійних ситуацій без використання точних математичних моделей, які базуються на апараті математичних рівнянь і класичної теорії прийняття рішень.

Побудова точної математичної моделі для погано формалізованих об'єктів і процесів є досить складною задачею у зв'язку з відсутністю повної інформації. Ситуація ще більше ускладнюється, якщо властивості об'єкта або процесу змінюються динамічно. Крім того, знання є такою структурою, що постійно змінюється і розвивається, а це в свою чергу може призвести до необхідності повної переробки математичної моделі.

Слід зазначити, що якість функціонування експертної системи залежить від повноти, несуперечності, а також розміру бази знань. Велика кількість правил призводить як до зниження швидкості виконання логічного виводу, що неприпустимо для систем, які працюють у режимі реального часу, так і до суперечливості накопичених знань.

Тому розробка математичних методів і алгоритмів, що дозволяють структурувати систему правил і визначати порядок їх викликів, контролювати несуперечність і повноту, оптимізувати кількість правил, є актуальною задачею.

При формуванні баз знань експертних систем існує 2 підходи: аналіз знань експертом (групою експертів) на основі досвіду, або автоматичне формування бази знань з використанням методів інтелектуального аналізу даних і алгоритмів машинного навчання.

Використання другого підходу дозволяє виконувати процедури створення і контролю баз знань в автоматичному режимі. В даний час існує декілька підходів до автоматизації зазначених процесів.

В [1] з використанням теорії графів продукційна база знань представляється у вигляді мультиграфа, в якому кожному продукційному правилу відповідає власний підграф. Побудована таким чином база знань структурується, для збільшення швидкодії механізму логічного виводу умови правил зв'язуються зі значеннями атрибутів в робочій пам'яті, що дозволяє врахувати вплив результатів виконання одних продукційних правил на умови реалізації інших.

В [2] розроблено математичну модель, яка представляє собою гіперграф. Всі сутності і залежності, що представлені в базі знань, об’єднуються у гіперграфі. Модифіковані алгоритми прямого та зворотного виведення здійснюють пошук на отриманому підграфі.

В роботі [3] запропоновано підхід до структурування бази знань шляхом визначення рейтингів правил з подальшим видаленням суперечливих правил і правил з найменшими рейтингами.

Методи інтелектуального аналізу даних і нейронечітких технології останнім часом знаходять застосування при створенні та аналізі баз знань. Так в [4] запропонована математична модель нейронної мережі, яка допускає пряме перетворення бази знань в мережу. З використанням генетичного алгоритму виконується параметрична оптимізація структури бази знань.

В роботі [5] пропонується поєднання методу кластерного аналізу та нечіткої моделі логічного виводу Такагі-Сугено для редукції бази знань, схожі правила логічного виведення об'єднуються в один кластер. Для оцінки схожості передумов правил при однакових значеннях логічних висновків використовуються спеціальні метрики. На число логічних висновків накладаються кількісні обмеження.

В роботі [6] задача кластеризації знань в системах штучного інтелекту розв’язується із застосуванням мурашиних алгоритмів.

Зазначені підходи істотно поліпшують роботу експертних систем, однак дозволяють працювати тільки зі статичними базами знань, накладають обмеження на кількість логічних висновків і не можуть бути застосовані для випадків, коли в існуючу систему необхідно додавати нові логічні правила.

У даній роботі розвивається підхід, який дозволяє розширювати базу знань експертної системи новими правилами в процесі експлуатації.

Постановка задачі. Припустимо, що існує сформована продукційна база знань $R=\{R_1, R_2, \dots, R_N\}$, де R_i – нечітке продукційне правило; $i=\overline{1,N}$; N – вихідна кількість правил в базі знань. При цьому передбачається, що у множені правил виділені k підмножин ($k<N$), у яких правила згруповані за

результатом логічного висновку. Підмножина правил містить правила, з використанням яких описуються приблизно однакові нечіткі закономірності в аналізованих даних і виконується класифікація об'єктів за допомогою алгоритму нечіткого логічного виведення. Передбачається, що існуюча система правил може бути розширена, коли у систему потрапляє новий об'єкт.

При цьому структура правила системи залишається незмінною. Подібні вимоги дозволяють уникати додаткових перевірок логічних правил на наявність протиріч після додавання нових правил в систему.

Необхідно розробити підхід, з використанням якого при виконанні процедури прямого логічного виведення стає можливим формування і додавання в систему нових продукційних правил для класифікації об'єктів, параметри яких відрізняються від вже використаних у навчальній вибірці при складанні бази правил.

Продукційна модель представлення знань. Розглядається задача ідентифікації об'єктів з використанням алгоритму нечіткого логічного виведення та системи продукційних правил.

Об'єкт *X* характеризується вектором параметрів $(x_1, x_2, \ldots, x_n)$,

де x_i - вхідна лінгвістична змінна. З кожною лінгвістичною змінною пов'язано її нечітке значення A_j.

З кожним об'єктом *X*, що характеризується набором вхідних параметрів, зв'язується вихідна змінна *Y*.

Вхід і вихід об'єкту, що досліджується, пов'язані між собою функціональною залежністю виду:

$$Y = f(x_1, x_2, \ldots, x_n) \qquad (1)$$

де *f* – дійсна функція від чітких значень $(x_1, x_2, \ldots, x_n)$

Функція $f(x_1, x_2, \ldots, x_n)$ набуває дискретних значень, оскільки число різних заключень логічних правил при будь-яких значеннях аргументів $x_1, x_2, \ldots, x_n$ є скінченним.

При розв'язанні задачі класифікації значеннями функції $f(x_1, x_2, \ldots, x_n)$ є константи, що вказують на клас до якого належить об'єкт.

Апріорну інформацію стосовно залежності (1) можна зобразити у вигляді сукупності продукційних правил у такий спосіб:

$$\begin{aligned} \Pi_P : Якщо\ & x_1 \in A_{P1} \wedge x_2 \in A_{P2} \wedge \ldots \wedge \\ & x_n \in A_{Pn}\ \ TO\ \ Y = Y_P, \end{aligned} \qquad (2)$$

де $p = \overline{1, P}$ – номер правила в базі правил; *P* – загальна кількість правил; A_{pj} – нечітке значення змінної x_i в термі j, $i = \overline{1, N}$; Y_p – мітка деякого класу, до якого належить об'єкт *X*.

Сформована база правил може бути розбита на систему підмножин, що перетинаються, за значенням мітки класу в логічному заключенні правила:

$$R = \{R_1, R_2, ..., R_N\} = \{Rul_1 \mid Y = Y_1 \cup Rul_2 \mid Y = Y_2 \cup ... \cup Rul_k \mid Y = Y_k\}$$

де Rul_k – підмножина правил, в яких $Y = Y_k$.

Для відображення чітких вхідних значень x_j – в нечіткі множини вводяться функції приналежності M_{pj} виду:

$$M_{pj}(x_j, a_p, b_p, c_p, d_p) = \begin{cases} 0, & x_j \le a_p \\ \dfrac{x_j - a_p}{b_p - a_p}, & a_p \le x_j \le b_p \\ 1, & b_p \le x_j \le c_p \\ \dfrac{d_p - x_j}{d_p - c_p}, & c_p \le x_j \le d_p \\ 0, & d_p \le x_j \end{cases} \quad (3)$$

де a_p, b_p, c_p, d_p - числові параметри, які визначають границі термів, набувають дійсних значень і впорядковані відношенням: $a_p \le b_p \le c_p \le d_p$.

Границі термів визначаються за формулами відносно бази даних навчальної вибірки:

$$\underline{\overline{x}}_j = \begin{cases} \left(\min\limits_{i=1}\left(x_i\right), \max\limits_{i=1}\left(x_i\right)\right), \; j = 1, J = 2 \\ \left(\min\limits_{i=1}\left(x_i\right), \max\limits_{i=1}\left(x_i\right)\right), \; j = J, J = 2 \\ \left(\min\limits_{i=1}\left(x_i\right), \min\limits_{i=1}\left(x_i\right) + \dfrac{\left(\underset{i=1}{avg}\left(x_i\right)\right)}{J-2}\right), j = 1 \\ \left(\min\limits_{i=1}\left(x_i\right) + \dfrac{(j-1)\cdot \underset{i=1}{avg}\left(x_i\right)}{J-1}, \max\limits_{i=1}\left(x_i\right) - \dfrac{(J-j)\cdot \underset{i=1}{avg}\left(x_i\right)}{J-1}\right), 1 < j < J \\ \left(\max\limits_{i=1}\left(x_i\right) - \dfrac{\left(\underset{i=1}{avg}\left(x_i\right)\right)}{J-2}, \max\limits_{i=1}\left(x_i\right)\right), j = J \end{cases}, \quad (4)$$

де $j=\overline{1,J}$, $J\geq 2$, J - кількість елементів терм-множини лінгвістичної змінної, якщо $J<2$ - немає сенсу створювати таку лінгвістичну змінну.

Метод розв'язання. Для виконання процедури нечіткого логічного виведення введемо такі операції, аналогічно [8].

Для виконання процедури фазифікації вхідних змінних x_i в вектор нечітких множин A_{pj} будемо використовувати наступну операцію:

$$A_{pj}=\int_{\underline{x_i}}^{\overline{x_i}}(M_{pj}(x_i)/x_i)dx \qquad (5)$$

де $\underline{x_i}, \overline{x_i}$ – границі термів вхідних змінних.

Для виконання процедури дефазифікації будемо використовувати:

$$Y_p=\int_{\underline{Y}}^{\overline{Y}}(M_{Y_p}(Y)/Y)dY,\ Y\in[\underline{Y};\overline{Y}] \qquad (6)$$

де $\underline{Y}, \overline{Y}$ – границі терму вихідної змінної.

Ступінь приналежності вхідного об'єкта $X^*=(x_1^*, x_2^*,..., x_n^*)$ нечітким термам Y_p з бази знань (2) описується системою нечітких логічних рівнянь:

$$M_{Y_p}(x^*)=\bigvee_{p=\overline{1,P}}\bigwedge_{i=\overline{1,N}}\left[M_{pj}(x_i^*)\right],\quad j=\overline{1,m} \qquad (7)$$

де оператори $\vee$ та $\wedge$ відповідають виконанню логічних операцій «АБО» та «І» відповідно. В роботі використані їх реалізації у вигляді знаходження $\max$ та $\min$.

Нечітка множина $\tilde{Y}^*$, що відповідає вхідному вектору X^* визначається у вигляді:

$$\tilde{Y}^*=\underset{j=\overline{1,m}}{agg}\left(\int_{\underline{Y}}^{\overline{Y}} imp(M_{Y_p}(X^*), M_{Y_p}(Y)/Y)dY\right) \qquad (8)$$

де imp – операція імплікації, agg – операція агрегування, які реалізовані операцією знаходження $\min$ та $\max$ відповідно.

Чітке значення виходу *Y** визначається в результаті дефазифікації нечіткої множини $\tilde{Y}^*$ за методом центру тяжіння:

$$Y^*=\int_{\underline{Y}}^{\overline{Y}} Y\cdot M_{\tilde{Y}}(Y)dY \Big/ \int_{\underline{Y}}^{\overline{Y}} M_{\tilde{Y}}(Y)dY \qquad (9)$$

Для виконання ідентифікації об'єкта використовується алгоритм нечіткого логічного виведення [7], модифікований введенням додаткового механізму порівняння чіткого вихідного значення *Y**, отриманого для об'єкта $X^*=(x_1^*, x_2^*,..., x_n^*)$, і вихідних значень об'єктів Y_i наявних в базі знань. Схему модифікованого алгоритму нечіткої логіки наведено на рис. 1.

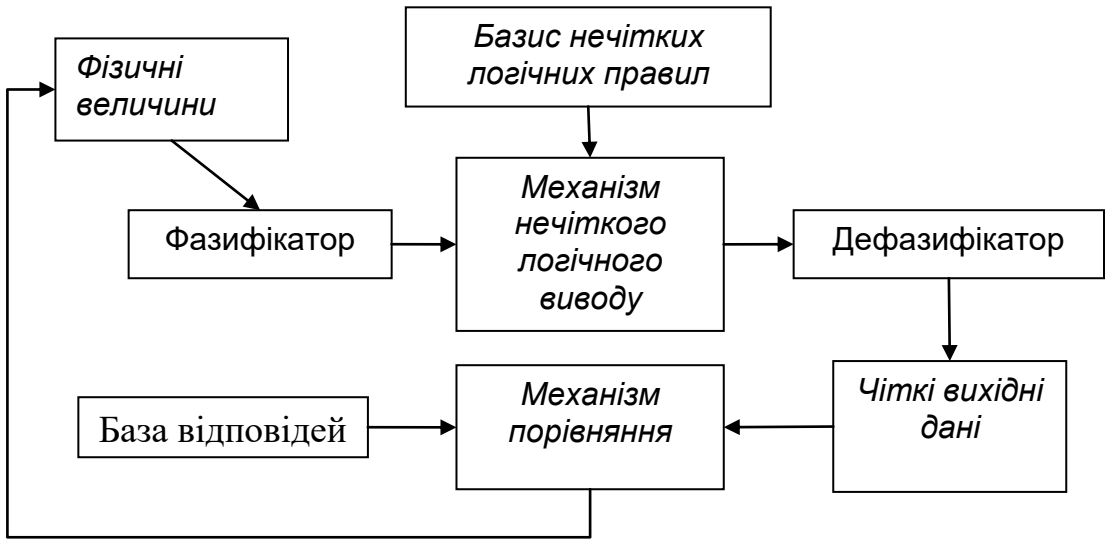

Рис. 1. Модифікований алгоритм нечіткої логіки

Відстань між об'єктами визначається на основі обраної метрики в просторі характеристик. Для оцінки міри близькості елементів використовується Евклідова відстань:

$$d(Y^*, Y_i) = \sqrt{\sum_{t=1}^{m} (Y^*_t - Y_{it})^2} \qquad (10)$$

$$d(Y^*, Y_i) < \varepsilon \qquad (11)$$

У разі, коли для об'єкта *X** умова (11) не виконується, необхідно сформулювати нове логічне правило:

$$\Pi_{P+1}: Якщо \ x_1 \in A_{k1} \wedge x_2 \in A_{k2} \wedge ... \wedge x_n \in A_{kn} \ TO \ Y_{P+1} = X^*, \qquad (12)$$

де, A_{kj} – нечіткі значення змінних x_j, які обчислюються з використанням функції приналежності (5) при значеннях $a_{k1} = x_1^*, a_{k2} = x_2^*, ..., a_{kn} = x_n^*$.

Після виконання процедури дефазифікації обчислюється *d(Y*,Y)* та перевіряється умова (11).

Вказану послідовність дій для існуючої бази правил можна описати наступним алгоритмом.

Алгоритм:

Крок 1. Задати значення ε – похибка системи, *P* – загальне число правил, ініціалізувати вхідні змінні, задати функції приналежності.

Крок 2. Для об’єкта X^* на базі сформованих продукційних правил за допомогою операцій алгоритму нечіткого логічного виведення (5) - (8) розрахувати ступені приналежності об'єкта чітким множинам $\tilde{Y}_p$.

За формулою (9) отримати чітке значення Y^* для об’єкта X^*.

Для об'єктів Y_k, які є логічними висновками правил з системи (2), так само виконати процедуру нечіткого логічного виведення та отримати відповідні значення $Y_k^*, k = \overline{1, K}$.

Крок 3. Визначити відстані між об'єктом Y^* і існуючими об'єктами Y_k, перевірити виконання умови (11).

Якщо нерівність є вірною перейти до кроку 6, інакше перейти до кроку 4.

Крок 4. Доповнити базу знань правилом (12); *p:=p+1* .

Крок 5. Перерахувати границі термів нечітких множин відповідно до формул (4).

Крок 6. Видати результат ідентифікації.

Крок 7. Перевірка правил останову:

Перевірити, чи всі об'єкти навчальної вибірки переглянуті, якщо так, то зупинитися, інакше обрати наступний об'єкт і перейти до кроку 3.

Аналіз результатів. Запропонований підхід до формування бази правил при пред'явленні елементів навчальної вибірки був протестований для задачі класифікації користувача відносно персонажа бази даних «CMD - Combat Marvel DC»[8].

В якості вхідних параметрів персонажа виступають: зріст (см), вага (кг), вік (роки). Зазначені параметри описуються нечіткими значеннями.

Формуються відповідні терми вхідних змінних:

Зріст (низький, середній, високий);

Вага (низька, середня, висока);

Вік (молодий, середній, дорослий).

Вихідна змінна характеризує класи об'єктів: Клас (I, II, III).

Для кожної вхідної та вихідної змінних вводяться функції приналежності виду (3), обчислюються границі термів за формулами (4).

Границі термів:

Зріст_низький = (155, 174.6);

Зріст_середній = (164.83, 186.33);

Зріст_високий = (174.6, 198);

Вага_низка = (50, 73.33);

Вага_середня = (61.6, 89.16);

Вага_висока = (73.33, 105);

Вік_молодий = (19, 26.33);

Вік_середній = (22.7, 30.7);

Вік_дорослий = (26.33, 35);

Клас_I = (0, 0.5);

Клас_II = (0.25, 0.75);

Клас_III = (0.5, 1);

Формується система правил, передумови яких складені з усіх можливих комбінацій значень нечітких вхідних змінних (всього 27 правил).

У таблиці представлені результати ідентифікації різних об'єктів при заданих значеннях ε=2,5, *P=27,* база даних персонажів: X_{01}(198, 105, 35), X_{02}(171, 65, 25), X_{03}(155, 50, 19).

Вхідні об’єкти: X_1(175, 70, 20), X_2(180, 73, 21). У таблиці 1 подано результати ідентифікації.

	Вхідні параметри			Дефазифі-цировані значення Y*	$\min_{j=1}(d(X_i, X_j))$	Результат класиці-кації	Заклю-чення
	Зріст (см)	Вага (кг)	Вік (роки)				
X_1	175	70	20	42,857	2,179	Клас_II	X_{02}
X_2	180	73	21	48,75	8,072	Клас_II	Новий
X_{01}	198	105	35	75	0	Клас_I	X_{01}
X_{02}	171	65	25	40,678	0	Клас_II	X_{02}
X_{03}	155	50	19	25	0	Клас_III	X_{03}

Таблиця 1

Результати ідентифікації.

Для об'єкта X_1 система повертає об'єкт X_{02}, який знаходиться найближче. Фактично ми відобразили об'єкт X_1 в множину об'єктів бази відповідей X_{01}, X_{02}, X_{03}. Для об'єкта X_2 відстань d = 8,072, тобто умова $d < \varepsilon$ не виконується, отже, потрібно додати цей об'єкт і розширити систему правил правилом з новим заключенням.

Запропонований алгоритм, було реалізовано у вигляді програмного продукту з використанням мов C/C++ та JavaScript, а також текстового формату обміну даними JSON. Для розробки використовувалися: NetBeans IDE, WhiteStarUML, GitHub, WebGL, Chrome, Mozilla Firefox, Opera.

Розроблений алгоритм має наступні переваги: висока швидкість розв'язання задачі; можливість розширення кількості відповідей системи, без зміни існуючих правил у базі знань і алгоритму логічного виведення; розширення спектру застосування алгоритмів нечіткої логіки.

Розроблений алгоритм має наступні недоліки: якщо в базі відповідей системи є об'єкти схожі один на одного, вони можуть мати однаковий центр ваги, що в свою чергу призводить до додаткових перевірок; мінімальну відстань для відображення об'єкта потрібно підбирати експериментальним шляхом.

Висновки: запропоновано підхід до автоматичної генерації продукційних правил бази знань на основі порівняння нових об'єктів з вже існуючими у системі за допомогою метрики, модифіковано алгоритм нечіткого логічного виведення шляхом додаванням блоку порівняння вихідного значення для нового об'єкту з вихідними значеннями для наявних у базі знань об'єктів.

Подальші дослідження будуть спрямовані на вдосконалення запропонованого підходу шляхом обчислення чітких вихідних даних для бази відповідей заздалегідь і внесення цих даних до бази відповідей, але для цього буде потрібно створити механізм контролю даних для постійного їх оновлення і перезапису в разі зміни або модифікації правил системи, а також в разі додавання нових відповідей системи.

Бібліографічні посилання:

1. **Домнич В.С., Иващенко В.А.** Построение базы знаний для поиска причин аварийных ситуаций при формовании листового стекла // УБС. Вып. № 33. – М.: ИПУ РАН, 2011. – 218–232 с.

2. **Иванов А. С.** Модель представления продукционных баз знаний на ЭВМ // Изв. Сарат. ун-та. Нов. сер. Сер. Математика. Механика. Информатика, 7:1. – Саратов: Изд-во Сарат. ун-та, 2007. – 83–88 с.
3. **Сергиенко М.А.** Методы проектирования нечеткой базы знаний / М.А. Сергиенко // Вест. Воронеж. гос. ун-та. Серия: Системный анализ и информационные технологии – №2. – Воронеж: Издательство Воронежского университета, 2008. – 67 – 71 с.
4. **Бухнин А.В., Бажанов Ю.С.** Оптимизация баз знаний экспертных систем с применением нейронных нечетких сетей // Нейрокомпьютеры: разработка, применение. – М.: Радиотехника, 2007. – №11.
5. **Абдулхаков А.Р., Катасёв А.С.** Кластерно-генетический метод редукции баз знаний интеллектуальных систем // Фундаментальные исследования. 2015. – № 5-3. – 471-475 с.
6. **Щуревич Е.В.** Кластеризация знаний в системах искусственного интеллекта // Информационные технологии. 2009. №2. 25-29 с.
7. **Ротштейн А.П.** "Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети". – [Электронный ресурс]. – Режим доступа: http://matlab.exponenta.ru/fuzzylogic/book5/index.php
8. **Егошкин Д.И.** "CMD - Combat Marvel DC". – [Электронный ресурс]. – Режим доступа: https://knightdanila.github.io/CMD/index.html

Дата надходження до редколегії: 13.05.2018