TECHNICAL SCIENCES AND TECHNOLOGIES
УДК 004.8: 004.89:519.7
DOI: 10.25140/2411-5363-2019-3(17)-131-139

Ігор Повхан

ПИТАННЯ ГНУЧКОСТІ ЛОГІЧНИХ ДЕРЕВ КЛАСИФІКАЦІЇ В ЗАДАЧАХ РОЗПІЗНАВАННЯ ОБРАЗІВ

Актуальність теми дослідження. *Сучасні тенденції розвитку теорії штучного інтелекту вимагають ефективних підходів та методів у задачах розпізнавання (класифікації) образів, але принциповою проблемою побудови логічних дерев класифікації є відсутність алгоритмів та методів, які б дозволили одноманітно описувати різні алгоритми розпізнавання у вигляді дерев класифікації. Робота присвячена проблемі логічних дерев класифікації. Запропоновано ефективний механізм донавчання та усунення помилок класифікації у структурі логічного дерева.*

Постановка проблеми. *Нині відомі різні методи та алгоритми побудови логічних дерев класифікації, проте всі вони здебільшого зводяться до побудови одного дерева класифікації за даними початкової навчальної вибірки, а в літературі дуже мало алгоритмів побудови логічних дерев для вибірок великого об'єму. Зрозуміло, що це має під собою об'єктивні фактори, які пов'язані з особливостями генерації таких складних структур, методиками роботи з ними та зберігання. Ця робота має намір хоча б частково подолати ці обмеження та присвячена розробці ефективного механізму донавчання та усунення помилок класифікації у структурі логічного дерева.*

Аналіз останніх досліджень і публікацій. *Були розглянуті останні публікації у відкритому доступі, які присвячені проблемі методів та алгоритмів логічних дерев класифікації в задачах розпізнавання образів.*

Виділення недосліджених частин загальної проблеми. *Можливість ефективної та економної роботи запропонованого методу зміни структури логічного дерева з масивами навчальних вибірок великого об'єму.*

Постановка завдання. *Розробка простого та якісного методу роботи з великими масивами початкових вибірок шляхом синтезу мінімальних форм дерев класифікації та розпізнавання, які забезпечують ефективну апроксимацію навчальної інформації.*

Виклад основного матеріалу. *Виявлення механізму, за допомогою якого можна було б будувати логічне дерево класифікації за неповною початковою інформацією (і за кількістю об'єктів, і за кількістю ознак). Таке логічне дерево буде безпомилково розпізнавати частину навчальної вибірки, за якою побудоване дерево, а на інших наборах давати помилки (уникнення такої ситуації пропонується за рахунок застосування схеми алгоритму усунення помилок у структурі дерева).*

Висновки відповідно до статті. *Запропонований метод донавчання та усунення помилок у структурі логічного дерева класифікації дає змогу працювати з навчальними вибірками великого об'єму та забезпечує високу швидкість та економність апаратних ресурсів у процесі генерації кінцевої схеми класифікації.*

Ключові слова: *задачі розпізнавання; логічні дерева класифікації; усунення помилок класифікації; схема розпізнавання; дискретний об'єкт.*

Табл.: 1. Бібл.: 10.

Актуальність теми дослідження. Важливою проблемою, яка часто постає перед інженером, є задача автоматичної побудови систем обробки великих масивів інформації та систем прийняття рішень. Ефективний розв'язок цих задач дасть змогу перекласти важку роботу з проектування складної системи розпізнавання на комп'ютер та звільнити творчий потенціал інженера для розв'язання інших більш важливих задач. Крім того, автоматизації алгоритмічного та програмного конструювання конкретних систем розпізнавання є запорукою їх високої ефективності для кожної реальної задачі, а отже, забезпечить швидкий розвиток різних галузей науки й техніки [1].

Основні наявні методи обробки навчаючих вибірок при побудові функції розпізнавання не дозволяють досягнути потрібного рівня точності системи розпізнавання та регулювати їхню складність у процесі конструювання цих систем. Цей недолік відсутній у методах побудови систем розпізнавання, які ґрунтуються на методах логічних дерев класифікації (ЛДК) [2]. При цьому особливістю методу логічного дерева є можливість комплексного використання для розв'язання кожної конкретної задачі побудови схеми розпізнавання багатьох відомих алгоритмів (методів) розпізнавання. В основі лежить єдина методологія – оптимальна апроксимація навчаючої вибірки набором узагальнених ознак (автономних алгоритмів), які входять у деяку схему (оператор), побудовану у процесі навчання [3].

Можливість представлення функції розпізнавання (правила класифікації) у вигляді логічного дерева має великі переваги в порівнянні з іншим представленням схем класифікації [4]. Зауважимо що запропонований алгоритм генерації дерев класифікації за даними НВ доповнює методологію підходу ЛДК та дозволяє будувати прості та ефективні правила класифікації дискретних об'єктів [5]. У цьому дослідженні зупинимось саме на описі та особливостях алгоритму побудови, корекції ЛДК для масиву початкових даних навчальної вибірки (НВ).

Постановка проблеми. Нині відомі різні алгоритми побудови ЛДК [6]. Проте всі вони здебільшого зводяться до побудови одного дерева класифікації за даними НВ. Зазначимо також, що в літературі дуже мало алгоритмів побудови ЛДК для НВ великого об'єму. Зрозуміло, що це має під собою об'єктивні фактори, пов'язані з особливостями генерації таких складних структур, методиками роботи з ними та зберігання [7]. Однак основним недоліком у питанні побудови ЛДК є відсутність алгоритмів та методів, які б дозволили одноманітно описувати різні алгоритми у вигляді ЛДК.

Нехай на першому кроці побудови логічного дерева розпізнавання використовується довільний алгоритм розпізнавання, у результаті застосування якого отримуємо деяку формулу (узагальнену ознаку). Ця формула реалізує визначений рівень розпізнавання. Функція приймає декілька значень залежно від значень ознак. Ці значення характеризують собою шляхи (класи), причому є шляхи, по яких формула «працює добре», а є і такі, по яких – «погано» і покращення рівня розпізнавання далі немає. Зрозуміло, що саме на цих значеннях ознак (шляхах) необхідно взяти інший алгоритм, який створить іншу формулу (узагальнену ознаку) і т. ін. Таким чином, у методах та алгоритмах розпізнавання на основі логічних дерев необхідно повторювати такий вибір алгоритмів доти, доки ми не отримуємо необхідний рівень якості розпізнавання [2].

Аналіз останніх досліджень і публікацій. Дослідження продовжує цикл робіт, які присвячені проблемі деревоподібних схем розпізнавання (класифікації) дискретних об'єктів [2–7]. У них порушуються питання побудови, використання та оптимізації логічних дерев. Так, з [8] відомо, що результуюче правило класифікації (схема), яке побудоване довільним методом або алгоритмом розгалуженого вибору ознак, має деревоподібну логічну структуру. Логічне дерево складається з вершин (ознак), які групуються по ярусах і які отримані на певному кроці (етапі) побудови дерева розпізнавання [6]. Важливою задачею, яка виникає з [5], є задача синтезу дерев розпізнавання, які будуть представлятися фактично деревом (графом) алгоритмів. На відміну від наявних методів, головною особливістю деревоподібних систем розпізнавання є те, що важливість окремих ознак (групи ознак чи алгоритмів) визначається відносно функції, яка задає розбиття об'єктів на класи [9].

Виділення недосліджених частин загальної проблеми. Можливість ефективної та економної роботи запропонованого методу зміни структури логічного дерева з масивами навчальних вибірок великого об'єму.

Мета роботи. Метою цієї роботи є вивчення особливостей генерації правил розпізнавання в задачах розпізнавання на основі логічних дерев (ЛДК), розробка загальної схеми донавчання та виправлення помилок у структурі ЛДК навчальних вибірок великого об'єму. Результатом роботи є простий метод усунення помилок класифікації в ЛДК.

Виклад основного матеріалу. Нехай $H_0, H_1, \dots, H_{k-1}$ система класів (образів) яка задана на множині G, що складається з об'єктів $x_i, (i = 1, \dots, m)$. Характер розбиття множини G на відповідні класи задається за допомогою НВ такого вигляду:

$$\Big(\big(x_1, f_R(x_1)\big), \big(x_2, f_R(x_2)\big), \dots, \big(x_m, f_R(x_m)\big) \Big), \tag{1}$$

де $x_h \in G, f_R(x) \in \{0,1, \dots, k-1\}, (h = 1,2, \dots, m), k -$ кількість класів НВ, $m -$ загальна кількість об'єктів НВ, а $f_R(x)$ – деяка скінчено значна функція, що задає розбиття множини G на відповідні образи. Співвідношення $f_R(x_h) = l, (l = 0,1, \dots, k-1)$ означає що $x_h \in H_l$.

Отже, ЛДК – це зв'язний граф без циклів, у некінцевих вершинах якого знаходяться ознаки (тут ознаки – відповідні компоненти об'єктів $x_1, \dots, x_m$), тобто:

$$x_1 = (x_1^1, x_2^1, \dots, x_n^1), \dots, x_m = (x_1^m, x_2^m, \dots, x_n^m);$$
$$P_i(x) = x_i^j, (i = 1, \dots, n), (j = 1, \dots, m).$$

Зауважимо, що тут n – кількість ознак об’єктів НВ вигляду (1), а ребра нумеруються значеннями цих ознак. У кінцевих вершинах ЛДК знаходяться значення функції $f_R(x_h)$. Причому на довільному шляху ЛДК та ж сама ознака трапляється тільки один раз.

Зауважимо, що кожну НВ вигляду (1) можна поставити у відповідальність (за допомогою деякого алгоритму або методу представлення) цілком визначене ЛДК, яке об’єктам $x_h, (h = 1, \dots, m)$ НВ (1) ставить у відповідність значення функції $f_R(x_h)$, що задає розбиття R на множині G [5].

Нехай маємо ЛДК, побудоване за даними НВ (1) та набори такого вигляду:

$$\Big(\big(x_{m+1}, f_R(x_{m+1})\big), \big(x_{m+2}, f_R(x_{m+2})\big), \dots, \big(x_{m+\delta}, f_R(x_{m+\delta})\big)\Big). \quad (2)$$

У формулі (2) набір об’єктів тестової вибірки, на яких знайдені помилки розпізнавання, де $2 \le m \le K_1 * K_2 * \dots * K_n, \delta -$ кількість об’єктів ТВ, на яких знайдені помилки розпізнавання, $K_i -$ кількість значень ознаки $P_i(x)$, $(i = 1, \dots, n)$, $n -$ кількість ознак об’єктів НВ.

Нехай маємо ситуацію, коли $x_h \in H_l, 0 \le l \le k - 1$ у процесі розпізнавання з різним l, то будемо говорити, що відбулася помилка розпізнавання $x_h \in H_l$. Якщо об’єкт зустрічається у ТВ декілька раз, то будемо вважати, що це та сама помилка (помилка першого роду).

Загальна схема алгоритму усунення помилок розпізнання. Побудуємо таке ЛДК, щоб воно безпомилково розпізнавало об’єкти з НВ (1) та об’єкти з (2), тобто опишемо наступний алгоритм усунення помилок розпізнавання (УПР). У парі $\big(x_{m+1}, f_R(x_{m+1})\big)$ об’єкту $x_{m+1} = (x_1^{m+1}, \dots, x_n^{m+1})$ відповідає свій шлях $a_1, a_2, \dots, a_r; (r \in \{1, \dots, n\})$, який закінчується значенням $f_R(x_{m+1})$.

$$P_{i_1} = q(a_1), a_2 = (a_1)_{x_{i_1}^h}, P_{i_2} = q(a_2), a_3 = (a_2)_{x_{i_2}^h}, \dots$$
$$\dots, P_{i_{r-1}} = q(a_{r-1}), a_r = (a_{r-1})_{x_{i_{r-1}}^h},$$

де $f_R(x_h) = q(a_r), (h = 1,2, \dots, m), 1 \le r \le n$, відповідно $i_1, i_2, \dots, i_{r-1}$ номера ознак, що знаходяться у вершинах $a_1, a_2, \dots, a_{r-1}$ та $f_R(x_{m+1}) = q(a_r)$ (такий запис означає, що значення функції $f_R(x_{m+1})$ не збігається зі значенням $f_R(x_h)$, що знаходиться в некінцевій вершині a_r тобто виникла помилка розпізнавання). Запам’ятовуємо, що $f_R(x_h) = q(a_r)$ та після цього здійснюємо таку процедуру: $P_{i_r} = q(a_r), a_{r+1} = (a_r)_{x_{i_r}^{m+1}}$ (це означає, що з вершини a_r виходить стрілка $x_{i_r}^{m+1}$, відповідна значенню ознаки P_{i_r} (об’єкту x_{m+1}) та входить у наступну за нею вершину a_{r+1}.

Далі отримаємо:

$$P_{i_{r+1}} = q(a_{r+1}), a_{r+2} = (a_{r+1})_{x_{i_{r+1}}^{m+1}}, \dots, P_{i_t} = q(a_t), a_{t+1} = (a_t)_{x_{i_t}^{m+1}},$$
$$P_{i_{t+1}} = q(a_{t+1}), a_{t+2} = (a_{t+1})_{x_{i_{t+1}}^{m+1}}, \dots, P_{i_n} = q(a_n), a_{n+1} = (a_n)_{x_{i_n}^{m+1}}.$$

де $f_R(x) = q(a_{n+1})$, $i_1, i_{r+1}, \dots, i_t, i_{t+1}, \dots, i_n -$ номери ознак, що знаходяться у вершинах ЛДК $a_r, a_{r+1}, \dots, a_t, a_{t+1}, \dots, a_n, r \le t \le n - 1$.

На наступному етапі будемо вважати, що:

$$a_{r+1}^{k_r} = (a_r)_{k_r} \text{ та } f_R(x_h) = q\big(a_{r+1}^{k_r}\big), \forall k_r,$$
$$a_{r+2}^{k_{r+1}} = (a_{r+1})_{k_{r+1}} \text{ та } f_R(x_h) = q\big(a_{r+2}^{k_{r+1}}\big), \forall k_{r+1},$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$a_t^{k_{t-1}} = (a_{t-1})_{k_{t-1}} \text{ та } f_R(x_h) = q\big(a_t^{k_{t-1}}\big), \forall k_{t-1},$$
$$a_{t+1}^{k_t} = (a_t)_{k_t} \text{ та } f_R(x_h) = q\big(a_{t+1}^{k_t}\big), \forall k_t,$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$a_n^{k_{n-1}} = (a_{n-1})_{k_{n-1}} \text{ та } f_R(x_h) = q\big(a_n^{k_{n-1}}\big), \forall k_{n-1},$$
$$a_{n+1}^{k_n} = (a_n)_{k_n} \text{ та } f_R(x_h) = q\big(a_{n+1}^{k_n}\big), \forall k_n,$$

де $r \le t \le n-1$, $k_1, k_{r+1}, \dots, k_t, k_{t+1}, \dots, k_{n-1}, k_n -$ множина значень ознак $P_{i_r}, P_{i_{r+1}}, \dots, P_{i_t}, P_{i_{t+1}}, \dots, P_{i_{n-1}}, P_{i_n}$ відповідно.

Усі інші знайдені помилки розпізнавання у вибірці $\left((x_{m+2}, f_R(x_{m+2})), \dots, (x_{m+\delta}, f_R(x_{m+\delta}))\right)$ усуваються аналогічно.

ЛДК*, побудоване за допомогою алгоритму УПР, буде безпомилково розпізнавати не тільки об'єкти основної НВ (1), але й об'єкти з (2). Це випливає із вищеописаної загальної процедури побудови алгоритму УПР.

Якщо описану вище схему УПР включити в довільний алгоритм розпізнавання, який пов'язаний із концепцією ЛДК, наприклад [10], то це забезпечить велику гнучкість даних алгоритмів.

Отже, алгоритм УПР усуває знайдені помилки розпізнавання а ЛДК, використовуючи при цьому тільки правило класифікації, побудоване за даними НВ та об'єкти з ТВ, для яких відбулася помилка розпізнавання. При цьому правило класифікації змінюється таким чином, що правильно розпізнає як об'єкти НВ, так і об'єкти ТВ, на яких раніше виникали помилки розпізнавання. Нижче приведемо ще кілька прикладів практичних застосувань алгоритму УПР.

Для цього спочатку визначимо кількість можливих помилок розпізнавання на кожному шляху в ЛДК. При розпізнаванні дискретних наборів за допомогою ЛДК інколи доцільно знайти на дереві шляхи, на яких найбільш можливе виникнення помилок розпізнавання. Залежно від ситуації можна поступити по-різному – відмовитися розпізнавати об'єкти, розпізнавати об'єкти тощо. Нехай НВ приведена до допустимого вигляду (виконується гіпотеза адекватності та несуперечливості). Відомо, що ЛДК об'єкти з початкової НВ розпізнає безпомилково. Загальна кількість всіх можливих об'єктів, що містять n ознак, не перевищує $K_1 * K_2 * \dots * K_n$, де K_i – множина значень ознаки $P_i(x), (i = 1,2, \dots, n)$.

Звідси випливає, що ЛДК у процесі розпізнавання може допустити не більше ніж $K_1 * K_2 * \dots * K_n - m$ помилок, де $m -$ кількість об'єктів НВ.

Нехай $M^j_{\eta_{i_1} \dots \eta_{i_\xi}}$ – кількість пар $(x_1, f_R(x_1)), (i = 1,2, \dots, m)$, що входять до НВ, які задовольняють таким умовам:

$$x_i \in G_{\eta_{i_1} \dots \eta_{i_\xi}}, (1 \le \xi \le n) \text{ та } f_R(x) = j, (j = 0,1, \dots, k-1),$$

$$\text{де } G_{\eta_{i_1} \dots \eta_{i_\xi}} = \{x \in G / P_{i_1}(x) = \eta_{i_1}, \dots, P_{i_\xi}(x) = \eta_{i_\xi}\}.$$

Твердження 1. Якщо $\max\limits_{0 \le j \le k-1} M^j_{\eta_{i_1} \dots \eta_{i_\xi}} = S, (1 \le \xi \le n)$ то на шляху $\eta_{i_1} \dots \eta_{i_\xi}$ ЛДК у процесі розпізнавання можливе не більше ніж $K_{n-\xi} - S$ помилок, де $K_{n-\xi} = K_1 * K_2 * \dots * K_n / K_{i_1} * \dots * K_{i_\xi}$.

Доведення. Розглянемо регулярне ЛДК (тобто дерево, у якого в вершинах ξ-го ярусу знаходяться тільки ознаки $P_{i_\xi}(x)$, $(i = 1,2, \dots n)$ називається регулярним). Кількість вершин на ξ-му ярусі регулярного ЛДК дорівнює $K_{i_1} * \dots * K_{i_\xi}$. А кількість кінцевих вершин регулярного ЛДК дорівнює $K_1 * K_2 * \dots * K_n$, де $K_{i_\xi} -$ кількість значень ознаки $P_{i_\xi}(x)$, $(i = 1,2, \dots n)$, $n -$ кількість ознак НВ. Звідси, якщо кожну вершину ξ – ярусу регулярного ЛДК вважати початковою, то таке регулярне ЛДК розіб'ється на $K_{i_1} * \dots * K_{i_\xi}$ регулярних ЛДК, кожне з яких буде мати по $K_{n-\xi} = K_1 * K_2 * \dots * K_n / K_{i_1} * \dots * K_{i_\xi}$ кінцевих вершин. Таким чином, якщо кінцева вершина деякого ЛДК виявляється на ξ-му ярусі, то вона являє собою $K_{n-\xi}$ кінцевих вершин регулярного ЛДК. Якщо задано шлях $\eta_{i_1} \dots \eta_{i_\xi}, (1 \le \xi \le n)$, який приводить в кінцеву вершину, то ця вершина являє собою $K_{n-\xi}$ об'єктів. Але якщо в цю вершину вже попадає S об'єктів з НВ то кількість помилок, які можуть виникнути на даному шляху не перевищує $K_{n-\xi} - S$.

Із цього твердження випливає, що якщо $\max\limits_{0 \le j \le k-1} M^j_{\eta_{i_1} \dots \eta_{i_\xi}} = K_{i_1} * \dots * K_{i_\xi}$, то на шляху $\eta_{i_1} \dots \eta_{i_\xi}$ в ЛДК всі об'єкти будуть розпізнаватися правильно. Якщо $K_1 = K_2 = \dots = K_n = 2$, то твердження буде звучати таким чином: якщо $\max\limits_{0 \le j \le k-1} M^j_{\eta_{i_1} \dots \eta_{i_\xi}} = S$ на шляху $\eta_{i_1} \dots \eta_{i_\xi}$ в ЛДК в процесі розпізнавання можливе не більше ніж $2^{n-\xi} - S$ помилок.

Задача побудови ЛДК за неповною НВ. Інколи доцільно мати алгоритми, за допомогою яких можна було би побудувати ЛДК за неповною НВ, за m ознаками $(m < n)$, $n -$ кількість ознак об'єктів у НВ або по l_1 об'єктам з l, заданих об'єктів НВ $(l_1 < l)$. В другому випадку, коли ЛДК будується за l_1 об'єктами з НВ, можна скористатися любим з відомих алгоритмів [7]. Побудоване таким чином ЛДК буде безпомилково розпізнавати об'єкти l_1 з НВ, а на інших об'єктах НВ можливі помилки розпізнавання. В цьому випадку використовуємо алгоритм УПР, описаний вище. Побудоване таким способом ЛДК буде безпомилково розпізнавати об'єкти НВ. У ряді випадків описаний вище метод істотно прискорює процес побудови ЛДК за даними НВ, що дуже важливо для задач РО.

Для побудови ЛДК за m ознаками з n, якими задаються об'єкти НВ впорядкуємо ознаки по важливості (мірі інформативності або ціні). Для цієї задачі можна використовувати методи та підходи описані в роботі [9]. Після цього береться m впорядкованих по важливості ознак та будується ЛДК за допомогою любого відомого алгоритму дерева та алгоритму УПР. Це суттєво прискорює процес побудови ЛДК, що дуже важливо для практичного застосування.

Задача відновлення значень ознак за допомогою ЛДК та алгоритм УПР. Спочатку побудуємо ЛДК для об'єктів НВ, для яких всі значення ознак визначені. Після цього за допомогою побудованого дерева розпізнаємо об'єкти НВ, для яких значення ознак визначені не повністю.

Крок 1. На першому кроці зафіксуємо один з таких об'єктів тестової вибірки.

Крок 2. Додатково визначаємо невідомі значення ознак, $P_{i_1}, \dots, P_{i_\xi}$, $(1 \le \xi \le n)$ всякими значеннями. Проведемо розпізнавання цих об'єктів за побудованим ЛДК. Нехай об'єкт, для якого значення ознак $P_{i_1}, \dots, P_{i_\xi}$ не визначені на НВ, належить класу H_l, $(1 \le l \le k-1)$, k – кількість класів НВ. Якщо ЛДК віднесе цей об'єкт до класу H_l, то вважають, що ознаки $P_{i_1}, \dots, P_{i_\xi}$, визначені значеннями $x_{i_1}, \dots, x_{i_\xi}$, можуть приймати ці значення, а в противному випадку – ні.

Крок 3. На наступному кроці беремо (фіксуємо) наступний з об'єктів тестової вибірки та переходимо до кроку 1. Якщо таких об'єктів більше нема, то процес відновлення значень ознак закінчуємо.

Визначення 1. Інформація, що вказує тільки на порядок розміщення ознак в логічному дереві, будемо називати схемою ЛДК, а кількість ознак, що входять у цю схему назвемо її порядком (потужністю).

Теорема 1. Кількість схем ЛДК визначається рекурсивним відношенням: $N_{n+1} = N_n^2 * (n+1)$, де $N_2 = 2$, $n -$ кількість ознак (ознак об'єкта у НВ) задачі розпізнавання.

Доведення. Для $n = 2$ теорема очевидна. Для довільного n задача розбивається на прості випадки, при $n = 3$ маємо дві схеми порядку 2. Оскільки ці схеми незалежні, то маємо $N_3 = N_2^2 * 3$, при $n = 4$ схема ЛДК розбивається на дві незалежні схеми порядку 3, тому $N_4 = N_3^2 * 4$ і так далі.

Задача побудови мінімального ЛДК відносно кількості вершин. Запропонуємо алгоритм знаходження мінімальних тестів та використання їх для побудови ЛДК. Введемо деякі позначення, які необхідні для роботи.

$$G_{r_1 \dots r_n} = \left\{ x \in \frac{G}{P_1(x)} = r_1, \dots, P_n(x) = r_n \right\}, \quad (3)$$

де $P_1(x), \dots, P_n(x)$ – деяка система ознак, $r_i \in \{0,1,\dots,k-1\}$, $(i = 1,\dots,n)$, $M_{r_1 \dots r_n}$ – кількість тих пар НВ $(x_1, f_R(x_i))$, $(1 \le i \le m)$, які задовольняють умову: $x_i \in G_{r_1 \dots r_2}$.

$M^j_{r_1 \dots r_n}$ – кількість з (4) таких наборів, які задовольняють таким умовам:

$$x_i \in G_{r_1 \dots r_n} \text{ та } f_R(x) = j \quad (4)$$

$$t^j_{r_1 \dots r_n} = M^j_{r_1 \dots r_n} / M_{r_1 \dots r_n} \quad (5)$$

$a_{r_1 \dots r_n} = \max\limits_{1 \le j \le k-1} t^j_{r_1 \dots r_n}$, де $r_1 \dots r_n \in \{0,\dots,k-1\}$, $(j = 0,1,\dots,k-1)$, $(i = 1,\dots,n)$ тоді:

Теорема 1.3 Набір ознак із номерами $i_1, \dots, i_\xi, (\xi \in 1,\dots,n)$ є тестом тоді і тільки тоді, коли для них виконується:

$$\sum\nolimits_{r_{i_1} \dots r_{i_\xi}} \left(\frac{M_{r_{i_1} \dots r_{i_\xi}}}{m} \right) * a_{r_{i_1} \dots r_{i_\xi}} = 1. \quad (6)$$

Формула (6) дає можливість знаходити тести заданої довжини (під довжиною тесту будемо розуміти кількість ознак, що входять у нього). При $\xi = 1$ формула (6) дає можливість знаходити тести довжиною «1», при $\xi = 2$ – тести довжини «2», а при $\xi = \eta$ – тести довжиною "η" $(1 \le \eta \le n)$. Формула (6) дає можливість знаходити тести у випадку коли в НВ присутня довільна кількість класів $f_R(x) \in \{0,1,\dots,k-1\}$ а ознаки $P_i(x)$ приймають довільні значення $P_i(x) \in \{0,1,\dots,k-1\}$.

Зауваження. Якщо хоча би один член виразу (6) не дорівнює одиниці, то набір ознак з номерами $r_{i_1} \dots r_{i_\xi}$ не є тестом.

Це зауваження дозволяє ефективно знаходити тести за допомогою формули (6). Мінімальні тести за формулою (6) знаходяться, зважаючи на визначення поняття мінімального тесту.

Виникає питання, як побудувати ЛДК за мінімальним тестом та чи буде таке побудоване дерево мінімальним (ЛДК побудоване за даними НВ, називається мінімальним, якщо немає іншого ЛДК*, побудованого за цією ж НВ, у якої кількість вершин менша за початкове дерево). Відповідь на першу частину питання досить тривіальна. Якщо група (набір) ознак із номерами $r_{i_1} \dots r_{i_\xi}$ є мінімальним тестом, то побудова ЛДК за цими ознаками не є важкою. Це можна зробити за допомогою алгоритмів побудови ЛДК, описаних, наприклад, у [7].

Відповідь на другу частину питання не є простою. Виявляється, що ЛДК, побудоване за мінімальним тестом, не завжди є мінімально можливим. Це можна побачити з таблиці 1.

Таблиця 1

Залежність складність дерева від кількості ознак

$KL \backslash D$	1	2	3	4	5	6	7	8	…	n
Max	3	7	15	31	63	127	255	511	…	$2^{n+1}-1$
Min	3	5	7	9	11	13	15	17	…	$2n+1$

Зауважимо, що тут D – довжина тупикового тесту, KL – кількість можливих вершин, Max – вказує на максимальну кількість вершин ЛДК (складність), при заданій довжині тупикового тесту, Min – вказує на мінімальну кількість вершин ЛДК, можливу при заданій довжині тупикового тесту.

З таблиці бачимо, що можливий випадок, коли ЛДК, побудоване за мінімальним тестом, може мати більше вершин, ніж ЛДК, побудоване за тупиковим тестом більшої довжини. Наприклад, ЛДК, побудоване за мінімальним тестом завдовжки 4, може мати 31 вершину, а ЛДК, побудоване за тупиковим тестом завдовжки 14, може мати 29 вершин, ЛДК, побудоване за мінімальним тестом завдовжки 10, може мати 2047 вершин, а ЛДК, побудоване за тупиковим тестом завдовжки 100, може мати 201 вершину (за тупиковим тестом завдовжки 1000 – 2001 вершину).

Зі сказаного вище випливає дуже важливе питання: як за заданою НВ побудувати ЛДК, яке мало б мінімальну кількість вершин. Особливу важливість це питання набуває в тих випадках, коли треба побудувати деяку технічну схему, модель, систему, яка б правильно класифікувала фіксовану кількість наборів. Вирішення такого питання дозволить у цих випадках будувати найпростішу процедуру (схему, систему) розпізнавання.

Складність ЛДК залежить від кількості кінцевих вершин у дереві.

Якщо кількість кінцевих вершин у ЛДК позначити через K, то загальна кількість вершин N в ЛДК можна розрахувати за формулою:

$$N = 2K - 1. \tag{7}$$

Це справедливо для випадку, коли $(P_i(x) \in \{0,1\})$, $2 \le K \le 2^n$, де n – кількість ознак об'єктів НВ.

Кількість кінцевих вершин ЛДК залежить від того, скільки наборів НВ відповідають одній кінцевій вершині. Виявляється, що можна знайти за заданою НВ таке K, яке відповідає мінімальній кількості вершин деякого можливого ЛДК. Зрозуміло, що таке ЛДК буде мінімальним за складністю (кількістю вершин). Це випливає з формули (7).

Нехай вже є набір тупикових тестів, які знайдені за формулою (6). Зафіксуємо деякий тупиковий тест $P_{i_1} \vee P_{i_2} \vee \dots \vee P_{i_\xi}$. Для кожної ознаки, що входить у даний тупиковий тест, знайдемо: $\max\limits_{1\le j\le k-1} M^j_{r_{i_\varepsilon}}$, де $(\varepsilon = 0,1,\dots,\xi)$, $(\xi = 1,\dots,n)$, для кожної ознаки.

Розглянемо два можливі випадки:

1. Існує таке значення ознаки P_{i_ε}, що для неї буде $\max\limits_{1\le j\le k-1} M^j_{r_{i_\varepsilon}} < M_{r_{i_\varepsilon}}$.

2. Немає такого значення ознаки P_{i_ε}, що для нього $\max\limits_{1\le j\le k-1} M^j_{r_{i_\varepsilon}} < M_{r_{i_\varepsilon}}$.

У першому випадку будемо говорити, що для цього значення ознаки P_{i_ε} є деякий шлях у ЛДК (ЛДК побудоване за таким тупиковим тестом) довжиною один крок і цей шлях приводить у кінцеву вершину.

У другому випадку в ЛДК немає шляху довжиною в один крок для цього значення ознаки P_{i_ε} такого, який би приводив у кінцеву вершину дерева.

У першому випадку – це кількість наборів НВ, на яких ознака $P_{i_\varepsilon}(x)$ набуває фіксоване значення з множини $\{0,\dots,k-1\}$, а функція $f_R(x)$ набуває на всіх цих наборах фіксоване значення із множини $\{0,\dots,k-1\}$. Усі ці набори входять в одну кінцеву вершину ЛДК. Якщо у вершинах ЛДК стоїть ознака P_{i_ε}, то далі знаходимо всі $\max\limits_{1\le j\le k-1} M^j_{r_{i_1} r_{i_2}}$ для цього тупикового тесту і так далі доти, поки рівність $\max\limits_{1\le j\le k-1} M^j_{r_{i_1}\dots r_{i_\xi}} = M_{r_{i_1}\dots r_{i_\xi}}$ не виконається для всіх наборів $r_{i_1} \dots r_{i_\xi}$ з НВ. Після цього підраховуємо кількість ситуацій, коли рівність $\max\limits_{1\le j\le k-1} M^j_{r_{i_1}\dots r_{i_\varepsilon}} = M_{r_{i_1}\dots r_{i_\varepsilon}}$, $(\varepsilon = 1,\dots,\xi)$ виконувалась для деяких наборів $r_{i_1} \dots r_{i_\xi}$ з НВ. Позначимо це число через η, η – це кількість можливих кінцевих вершин ЛДК, яке може бути побудовано за цим тупиковим тестом.

Виконаємо таку ж саму процедуру для кожного тупикового тесту, знайденого за формулою (6) та порівняємо знайдене η для кожного тупикового тесту. Тест, для якого η буде мінімальним, і дозволяє побудувати мінімальне ЛДК.

Висновки відповідно до статті. Отже, зважаючи на все вищезазначене, у роботі можна зафіксувати такі положення:

1. Навіть у ситуації виникнення помилок у ЛДК на етапі роботи з тестовою вибіркою є можливість виправлення помилок (алгоритм УПР) шляхом корекції структури дерева, таким чином правильно класифікується як НВ, так і ТВ.
2. У довільному ЛДК побудованому за даними початкової НВ на деякому фіксованому шляху $\eta_{i_1} \dots \eta_{i_\xi}$ (шляху в ЛДК) у процесі розпізнавання можливе не більше ніж $K_{n-\xi} - S$ помилок.
3. Інколи доцільно мати алгоритм, за допомогою якого можна було б побудувати ЛДК за неповною НВ (і за кількістю об’єктів, і за кількістю ознак). Побудоване таким способом ЛДК буде безпомилково розпізнавати частину НВ, за якою побудоване дерево, а на інших наборах – давати помилки (уникнути цього можна, застосувавши алгоритм УПР).
4. Довільне ЛДК, побудоване за даними НВ за мінімальним тестом, не завжди є мінімально можливим.
5. Загальну кількість вершин у ЛДК, яке побудоване за даними НВ, можна розрахувати за формулою $N = 2K - 1$.

Список використаних джерел

1. Srikant R., Agrawal R. Mining generalized association rules. *Future Generation Computer Systems.* 1997. Vol. 13, № 2. P. 161–180.
2. Василенко Ю. А., Василенко Е. Ю., Повхан І. Ф., Ващук Ф. Г. Концептуальна основа систем розпізнавання образів на основі метода розгалуженого вибору ознак. *Науково технічний журнал «Східно-Європейський журнал передових технологій».* 2004. Т. 1., № 7. С. 13–15.
3. Василенко Ю. А., Повхан І. Ф., Ващук Ф. Г. Проблема оцінки складності логічних дерев розпізнавання та загальний метод їх оптимізації. *Науково технічний журнал «Східно-Європейський журнал передових технологій».* 2011. Т. 6, № 4(54). С. 24–28.
4. Василенко Ю.А., Повхан І.Ф., Ващук Ф.Г. Загальна оцінка мінімізації деревоподібних логічних структур. *Науково технічний журнал «Східно-Європейський журнал передових технологій».* 2012, Т. 1, № 4(55). С. 29–33.
5. Povhan I. General scheme for constructing the most complex logical tree of classification in pattern recognition discrete objects. *Електроніка та інформаційні технології.* 2019. Вип. 11. С. 112–117.
6. Василенко Ю. А., Василенко Е. Ю., Повхан І. Ф., Ковач М. Й., Нікарович О. Д. Мінімізація логічних деревоподібних структур в задачах розпізнавання образів. *Науково технічний журнал «Східно-Європейський журнал передових технологій».* 2004. Т. 3, №. 9. С. 12–16.
7. Лавер В. О., Повхан І. Ф. Алгоритми побудови логічних дерев класифікації в задачах розпізнавання образів. *Вчені записки Таврійського національного університету. Серія: технічні науки.* 2019. Т. 30(69), № 4. С.100–106.
8. Vtogoff P. E. Incremental Induction of Decision Trees. *Machine Learning.* 2009. № 4. P. 161–186.
9. Повхан І. Ф. Проблема функціональної оцінки навчальної вибірки в задачах розпізнавання дискретних об’єктів. *Вчені записки Таврійського національного університету. Серія: технічні науки.* 2018. Т. 29(68), № 6. С. 217–222.
10. Whitley D. An overview of evolutionary algorithms: practical issues and common pitfalls. *Information and Software Technology.* 2001. Vol. 43, №14. P. 817–831.

References

1. Srikant, R., Agrawal, R. (1997). Mining generalized association rules. *Future Generation Computer Systems*, 13 (2), 61–180.
2. Vasilenko, Yu. A., Vasilenko, E. Yu., Povkhan, I. F, Vashchuk, F. G. (2004). Conceptual basis of pattern recognition systems based on the method of branched feature selection. *Scientific and technical journal "Eastern-European Journal of Enterprise Technologies*, 1/7, 13-15 [in Ukrainian].

3. Vasilenko, Yu. A., Vashchuk, F. G., Povkhan, I. F. (2011). The problem of estimating the complexity of the logic trees, recognition, and a general method of optimization. *Scientific and technical journal "Eastern-European Journal of Enterprise Technologies"*, 6/4(54), 24-28 [in Ukrainian].
4. Vasilenko, Yu. A., Povkhan, I. F., Vashchuk, F. G. (2012). General estimation of tree logical structures minimization. *Scientific and technical journal "Eastern-European Journal of Enterprise Technologies",* 1/4 (55), 29-33 [in Ukrainian].
5. Povkhan, I. (2019). General scheme for constructing the most complex logical tree of classification in pattern recognition of discrete objects. *Collection of scientific papers «Electronics and information technology»*, 11, 112-117.
6. Vasilenko, Yu. A., Vasilenko, E. Yu., Povkhan, I. F., Kovach, M. I., Nikarovych, O. D. (2004). Minimization of logic tree structures in pattern recognition problems. *Scientific and technical journal "Eastern-European Journal of Enterprise Technologies",* 3/9, 12-16 [in Ukrainian].
7. Laver, V. O., Povkhan, I. F. (2019). Algorithms for constructing logical classification trees in pattern recognition problems. *Scientific notes of Tauride national University. Series: technical Sciences,* 30(69), 4, 100-106 [in Ukrainian].
8. Vtogoff, P. E. (2009). Incremental Induction of Decision Trees. *Machine Learning,* 4, 61−186.
9. Povkhan, I. F. (2018). The problem of functional evaluation of the training sample in the problems of recognition of discrete objects. *Scientific notes of Taurida national University. Series: technical Sciences*, 29(68), 6, 217-222 [in Ukrainian].
10. Whitley, D. (2001). An overview of evolutionary algorithms: practical issues and common pitfalls. *Information and Software Technology*, 43 (14), 817–831.

UDC 004.8:004.89:519.7

Igor Povkhan

FLEXIBILITY OF LOGICAL CLASSIFICATION TREES IN PATTERN RECOGNITION PROBLEMS

Urgency of the research. *Modern trends in the development of the theory of artificial intelligence require effective approaches and methods in the problems of recognition (classification) of images, but the fundamental problem of constructing logical classification trees is the lack of algorithms and methods that would allow uniformly describe different recognition algorithms in the form of classification trees. The work is devoted to the problems of logical classification trees, offers an effective mechanism of learning and elimination of classification errors in the structure of the logical tree.*

Target setting. *At present, various methods and algorithms for constructing logical classification trees are known, but all of them, as a rule, are reduced to the construction of a single classification tree according to the initial training sample, and in the literature there are very few algorithms for constructing logical trees for large samples. It is clear that this is based on objective factors related to the peculiarities of the generation of such complex structures, methods of working with them and storage. This paper intends to at least partially overcome these limitations and is devoted to the development of an effective mechanism for the completion and elimination of classification errors in the structure of the logical tree.*

Actual scientific researches and issues analysis. *The recent publications in the open access, which are devoted to the problems of methods and algorithms of logical classification trees in the problems of pattern recognition, were considered.*

The research objective. *The possibility of efficient and economical operation of the proposed method of changing the structure of the logical tree with arrays of training samples of large volume.*

The statement of basic materials. *Identification of the mechanism by which it would be possible to build a logical classification tree by incomplete initial information (both by the number of objects and by the number of features). This logical tree will accurately recognize the part of the training sample on which the tree is built, and give errors on other sets (to avoid this situation is proposed by applying the algorithm scheme to eliminate errors in the tree structure).*

Conclusions. *The proposed method of completion and elimination of errors in the structure of the logical classification tree allows you to work with training samples of large volume and provides high speed and efficiency of hardware resources in the process of generating the final classification scheme.*

Keywords: *recognition tasks; logical classification trees; elimination of classification errors; recognition scheme; discrete object.*

References: 10.

Повхан Ігор Федорович – кандидат технічних наук, доцент, доцент кафедри програмного забезпечення систем, ДВНЗ «Ужгородський національний університет» (вул. Заньковецької 89Б, м. Ужгород, 88000, Україна).
Povkhan Igor – PhD in Technical Sciences, Associate Professor, Associate Professor of Department of software, Uzhgorod national University (89B Zankovetskoy Str., 88000 Uzhgorod, Ukraine).
E-mail: igor.povkhan@uzhnu.edu.ua
ORCID: http://orcid.org/0000-0002-1681-3466

Повхан І. Питання гнучкості логічних дерев класифікації в задачах розпізнавання образів. *Технічні науки та технології*. 2019. № 3 (17). С. 131–139.