УДК 004.9

ОБҐРУНТУВАННЯ ШВИДКИХ АЛГОРИТМІВ КЛАСИФІКАЦІЇ НА МНОЖИНАХ BIG DATA ЗА КРИТЕРІЯМИ НАДІЙНОСТІ І ПРОДУКТИВНОСТІ

DOI 10.36994 / 2788-5518-2023-01-05-16

***Одегов М.А.**, к.т.н.,доц. Державний університет інтелектуальних технологій і зв'язку, Одеса, Україна. onick_64@ukr.net*

***Гаджиєв М.М.**, д.т.н.,проф. Державний університет інтелектуальних технологій і зв'язку, Одеса, Україна. gadjievmm@ukr.net*

***Буката Л.М.** Державний університет інтелектуальних технологій і зв'язку, Одеса, Україна. ygrikluda@gmail.com*

***Глазунова М.В.**, к.ф-м.н. Державний університет інтелектуальних технологій і зв'язку, Одеса, Україна. lvglazun@gmail.com*

***Кочеткова М.В.** Державний університет інтелектуальних технологій і зв'язку, Одеса, Україна., jubdvg@gmail.com*

***Анотація:** У роботі розглянуто швидкі алгоритми класифікації, що можуть бути застосовані для розв'язку задач типу Big Data. Розглянуті алгоритми. засновані на принципах M-means. При цьому всі можливі представники класів замінюються невеликою кількістю характеристик цих класів. Приведено теоретичне обґрунтування алгоритмів адаптивних метрик та адаптивних правил. Показано принципи реалізації даних алгоритмів для розв'язку досить широкої сфери практичних задач. Наведено приклад порівняння ряду алгоритмів класифікації за критеріями надійності та продуктивності.*

***Ключові слова:** швидкі алгоритми класифікації, множини елементів, центри класів, метрики, розподілення ймовірностей, вирішальні правила, показники надійності та продуктивності*

JUSTIFICATION OF FAST CLASSIFICATION ALGORITHMS ON BIG DATA SETS WITH RELIABILITY AND PERFORMANCE CRITERIA

***Nick Odegov**, Ph.D., Ass. Prof. State University of Intellectual Technologies and Communication, Odesa, Ukraine. onick_64@ukr.net*

***Matin Hadzhyiev**, Dr. habil., Prof. State University of Intellectual Technologies and Communication, Odesa, Ukraine. gadjievmm@ukr.net*

***Liudmyla Bukata.** State University of Intellectual Technologies and Communication, Odesa, Ukraine. ygrikluda@gmail.com*

***Liudmyla Glazunova**, Ph.D., Ass. Prof. State University of Intellectual Technologies and Communication, Odesa, Ukraine. lvglazun@gmail.com*

***Marina Kochetkova.** State University of Intellectual Technologies and Communication, Odesa, Ukraine., jubdvg@gmail.com*

***Abstract:** Classification methods are among the simplest and "oldest" methods of artificial intelligence. This article discusses fast algorithms that can be used to solve Big Data problems. Big Data is a case when the methods and tools used do not allow solving the problem in a pleasant time. Therefore, when solving this type of problem, an important criterion is the performance of classification algorithms. Productivity in this sense refers to potential decision-making time. This time depends on the constructive dimension of the algorithm - the number of typical operations for making a decision. In addition, the execution time of the program depends on the typical operations of the algorithm. The article considers productive algorithms based on M-means principles. At the same time, all possible representatives of classes are replaced by a small number of characteristics of these classes. In the simplest form, these algorithms boil down to the fact that classes are replaced by class centers based on the results of training. Such centers are defined as vectors of average values over all one-dimensional projections of the factor space. Distance measures from these centers in Euclidean and other metric spaces are used to classify unknown objects. Methods of rapid adaptation of these characteristics in learning algorithms are considered. The theoretical justification of the algorithms of adaptive metrics and adaptive rules is given. It is shown that M-means algorithms can be effective in the case that the number of class representatives is extremely large, and the number of classes and the dimension of the factor space are relatively small. The advantages and disadvantages of the considered algorithms are noted. The scope of their practical application is outlined. The*

principles of spreading these algorithms to a wide range of practical problems are also shown. An example of a comparison of a number of classification algorithms based on reliability and performance criteria is given. It was concluded that the algorithm of adaptive rules is the most effective for a significant number of problems.

Key words: *fast classification algorithms, element sets, class centers, metrics, probability distribution, decision rules, reliability and performance indicators*

Вступ

Поняття Big Data («великі дані») є невизначеним, умовним. У даній роботі ми дотримуємось ніби інтуїтивного розуміння цього поняття як ситуацію, коли наявні алгоритми або технічні засоби не дозволяють вирішити конкретну задачу за прийнятний час. Наприклад, дихотомія «свій - чужий» потребує часу вирішення у долі секунди. В іншій ситуації, наприклад при вирішенні проблеми: що вносить найбільший вклад у потепління на Землі «діяльність людини чи природні циклічні процеси» – прийнятним часом рішення може бути зовсім інший, більш тривалий час. В будь-якому випадку якість алгоритмів у випадку Big Data слід визначати за двома основними показниками: надійністю (частотою помилок класифікації) та продуктивністю (часом вирішення задачі).

Наша *перша парадигма* полягає в наступному: у випадку Big Data не існує універсального алгоритму, однаково ефективного для різних випадків. Тому розглянемо лише кілька варіантів таких, найбільш типових випадків.

Перший варіант: надзвичайно велика кількість класів, мінімальна кількість представників (елементів) у кожному класі, невелика кількість характеристик (розмірність фактор-простору) елементів. Типовий приклад: дактилоскопія – скільки людей на світі, стільки ж різних відбитків пальців.

Другий варіант: дуже велика кількість класів, можливо велика кількість елементів класів, відносно невелика кількість факторів. Приклад: встановлення особистості по фотографії. Тут розпізнаване обличчя відноситься до певного класу «двійників», а класів таких може бути багато.

Третій варіант: середня кількість класів, значна кількість елементів у кожному класі, середня кількість факторів. Приклад: розподіл громадян за неформальним критерієм якості життя, який ми спростимо до дихотомії: «задовільно - не задовільно». Ясна річ, що суб'єктивна оцінка якості життя може залежати не тільки від доходів. Як кажуть: «У когось борщ рідкий – у когось діаманти маленькі». Тому у сукупність факторів треба додати наявність житла, сімейний стан, стан медичного забезпечення та т.ін. Якщо населення вже визначилось на деякий час та розподілилось на «задоволених» та «незадоволених», то маємо базові набори представників класів типу А і типу В. У певний момент умовна держава щось змінює у правилах існування суспільства. Яка буде при цьому динаміка розподілу між «задоволеними» та «незадоволеними»?.

Саме для третього варіанту і можуть бути ефективними швидкі алгоритми, що розглядаються у даній статті.

На даний час саме на задачах Data Mining в умовах Big Data зосереджуються значні зусилля світової наукової спільноти. В роботах [1,2] наведено огляд методів класифікації. При цьому в [2] є положення, що будь-який алгоритм для вирішення задач Big Data повинен бути «однопрохідним» тобто не застосовувати вкладених циклів у процедури прийняття рішень. Для радикального вирішення проблеми Р. Белмана «прокляття розмірності» в умовах Big Data пропонуються різні прийоми [3-8].

Одним з типових є розв'язок задач не на загальній множині елементів класів, а лише на вибіркових даних [3]. В алгоритмах типу K-найближчих сусідів пропонується обмежені множини тих самих «найближчих сусідів» визначати ще на етапі навчання алгоритму, як сукупність елементів, що близькі до границь класів [4].

Також, ефективним і достатньо природним є принцип скорочення розмірності фактор-простору за результатами попереднього навчання [5] або встановлення залежностей між факторами [6]. Втім, ще одним способом суттєвого зменшення обсягу обчислювань є заміна всіх множин елементів класів характерними представниками даних класів. Історично першим методом, що реалізує даний спосіб був метод M-means (у звичайному розумінні M центрів класів, але дослівно M засобів), який використовується як у задачах класифікації, так і у задачах кластеризації [7]. При цьому відмічають й недоліки даного методу, але розвиток алгоритмів його реалізації продовжується завдяки надзвичайної продуктивності цього методу [8]. У даній роботі

принцип M-means розуміється дещо ширше: на етапі навчання для класів визначається дуже обмежена кількість характеристик, які дозволяють швидко розв'язати задачу класифікації.

Метою досліджень є обґрунтування швидких алгоритмів класифікації, що засновані на принципах M-means Data Mining і відповідають наступним неформальним вимогам:

- дуже незначний час навчання;
- дуже мала кількість характеристик класів;
- дуже малий час прийняття рішень;
- достатня надійність для вирішення певних типів задач класифікації.

Виконання досліджень

1. Формалізація моделі класів

Класи з умовними номерами 1, 2,…, M задаються базовими послідовностями елементів в одному й тому ж координатному просторі (фактор-просторі) розмірності D. Кожен елемент кожного класу визначається вектором значень факторів всього фактор-простору. Таким чином, базова модель задається сукупністю множин:

$$F_{m,n} = \{[f_{m,1}, f_{m,2}, ..., f_{m,D}](n)\}, m = 1,2,...,M, n = 1,2,...,N_k, \quad (1)$$

де n – номер елементу у класі m.

Зауважимо, що кількість елементів у різних класах може значно відрізнятись, втім в усіх класах кількість елементів є дуже великою: умовно десятки тисяч, мільйонів або мільярдів в залежності від розмірності фактор-простору. Визначимо також додаткові умови та обмеження, як формальні, так і неформальні:

- розподілення значень факторів для всіх класів приблизно відповідають формальним умовам закону великих чисел [9], тобто є статистично та динамічно стійкими;
- накопичення даних у базових послідовностях виконується з урахуванням попередньо вирішеної задачі кластеризації, тобто класи співпадають з кластерами;
- всі фактори вважаються рівноправними з точки зору прикладної задачі;
- всі класи вважаються рівноправними з точки зору прикладної задачі.

Дві останні умови формалізують нашу *другу парадигму*: ніякий машинний алгоритм не може відповідати за прийняття рішень, що є прерогативою людини, оскільки людина має інтереси та потреби, а машина – ні. Наприклад, якщо вже встановлено на рівні законодавства, що розмір дорожнього збору залежить від об'єму двигуна, то навіщо крім цього фактора включати у процедури класифікації колір автомобіля, тип підвіски, діаметр коліс та ін..? Якщо класифікація «пропуск цілі чи хибна тривога» потребує як можливо зменшувати кількість саме пропусків цілей, то який формальний алгоритм може визначити вагові коефіцієнти помилок першого та другого роду? У таких задачах відповідальність залишається за людиною.

2. Задача дихотомії: алгоритм М-центрів

У найпростішому випадку класифікатор приймає рішення віднесення невизначеного об'єкту X до одного з класів: A або B. При цьому об'єкт X задано у тому ж самому фактор-просторі, що і елементи класів A і B. Рішення класифікатора можуть бути 4-х типів, які ми умовно позначимо:

- A=A – вірне рішення віднесення об'єкту X до класу A;
- B=B – вірне рішення віднесення об'єкту X до класу B;
- B/A – помилка 1-го роду: прийнято рішення віднесення об'єкту X до класу B, тоді як об'єкт відноситься до класу A;
- A/B – помилка 2-го роду: прийнято рішення віднесення об'єкту X до класу A, тоді як об'єкт відноситься до класу B.

В алгоритмах навчання дані варіанти рішень зручно асоціювати з умовними номерами:

(A=A) => 0; (B=B) => 1; (B/A) => 2; (A/B) => 3.

Відповідно, за результатами навчання можна приблизно визначити умовні частоти даних рішень. Зважуючи на велику кількість елементів класів та умову їх статистичної стійкості, в роботі не завжди будемо розрізняти поняття «ймовірності» та «частоти» (емпіричної ймовірності), до

яких будемо використовувати позначення P. Саме таким символом будемо позначати інтегральні функції розподілення, а позначкою p – диференціальні функції розподілення.

Таким чином, принцип рівноправності класів потребує хоча б приблизного виконання рівності: P(B/A) ≈ P(A/B). Зрозуміло, що при цьому обидві ці ймовірності треба мінімізувати.

У даному випадку самий швидкий алгоритм може бути побудований за принципами методу M-means (дослівно: M-засобів). Кількість таких «засобів» пропорційна кількості класів, а самими засобами можуть бути обмежені набори характеристик класів, вирішальних правил та т.ін.

Тут треба зробити термінологічне зауваження. В літературі по Data Mining [7] міцно закріпились терміни «K-найближчих сусідів» та «K-means». При цьому мова йде зовсім про різні по суті числа K. У першому випадку – про кількість елементів класів, у другому – про кількість класів. Тому у даній роботі ми все-ж таки будемо користуватись у другому випадку терміном «M-means» (саме від слова «Means»).

У найпростішому вигляді алгоритм M-means зводиться до алгоритму M-представників класів (лише по одному на кожний клас) [7]. Тоді множини елементів (1) замінюються представниками класів:

$$F_m = [f_{m,1}, f_{m,2}, ..., f_{m,D}],\ m = 1,2,...,M,\ n = 1,2,...,N_k, \tag{2}$$

де значення факторів відносяться вже не до окремих елементів, а до їх представників.

У задачі дихотомії обидва класи замість всієї базової сукупності елементів замінюються лише двома «центрами класів», які ми також позначимо іменами класів. Відповідно, вирішальне правило формулюється так:

$$\rho(A,X) < \rho(B,X) \Rightarrow 0;\quad \rho(A,X) > \rho(B,X) \Rightarrow 1, \tag{3}$$

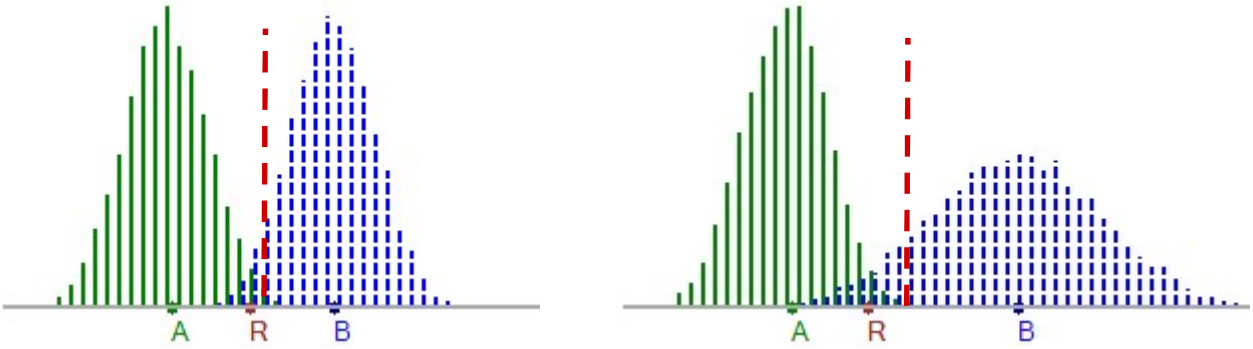
де ρ – обрана метрика у фактор-просторі.

Метрики можуть обиратись різними способами [7]. Втім, і поняття центрів класів також має бути пов'язане з типом метрики. Якщо обрана Евклідова метрика, то центри класів логічно визначати як вектори середніх значень по факторам, оскільки середнє мінімізує суму квадратів відстаней від нього. Якщо обрана метрика Хемінга, то логічно визначати центри класів як вектори медіан значень по факторам, оскільки медіана мінімізує суму модулів відстаней від неї.

Медіани є робасними статистиками, оскільки слабо чутливі до великих «викидів» у вибірках. Якщо задачі вирішуються на відносно малих масивах даних, то перевагу слід віддати метричним просторам Хемінга. Втім, для Big Data процедури навчання для визначення медіан вже ніяк не можна вважати «однопрохідними», оскільки дані алгоритми потребують ранжирування вибірок. Тобто, процес навчання може займати значний час. На відміну від цього, визначення середніх при нарощуванні обсягів елементів класів вирішується ітераційними алгоритмами за один крок. Наприклад, для класу A при додаванні нового, (N+1)-го елементу нове середнє по 1-му фактору розраховується за елементарною формулою:

$$A_{N+1,1} = \frac{1}{N+1}\left(f_{A,1}(n+1) + \sum_{n=1}^{N} f_{A,1}(n) \right) = \frac{1}{N+1} f_{A,1}(n+1) + \frac{N}{N+1} A_{N,1}. \tag{4}$$

Розглянемо найпростіший випадок дихотомії: класифікація в одномірному просторі. При цьому допускаємо, що функції розподілу ймовірностей для класів A і B відрізняються лише значеннями середніх (рис. 1.а).



а) б)

Рис. 1. Варіанти розподілів одного фактору для класів A і B

Такий алгоритм застосовується практично у задачі виявлення сигналів певного рівня на фоні шумів. Клас A – лише шум; Клас B – сигнал + шум. Втім, цим прикладом сфера застосування даного алгоритму і обмежується: навіть у 2-мірному просторі складно уявити ситуацію, коли два класи відрізняються лише середніми. Навіть у дуже простій ситуації, коли розподілення значень факторів відрізняються лише масштабом (мірою розсіювання відносно середнього), даний алгоритм буде давати очевидний перекіс ймовірності помилок 2-го типу (рис. 1.б). Невідомі об'єкти будуть скоріше відноситись до класу A, ніж до класу B.

Також відомі інші недоліки даного алгоритму:

- чутливість до наявності факторів з сильною кореляцією: пара чи більше таких факторів будуть надавати найбільш суттєвий вплив на загальне рішення; цей недолік усувається на етапі попереднього аналізу даних – частина таких факторів або відбраковується, або дані фактори агрегуються у якусь суспільну характеристику;

- велика чутливість до масштабів значень по факторам; для усунення цього недоліку використовується приведення факторів до єдиного масштабу.

Остання операція найпростіше і логічніше виконується, якщо функції розподілення значень факторів відрізняються лише масштабами: для цього метрики зважуються зворотно пропорційно стандартним відхиленням. В евклідовому просторі метрики щодо правил (3) уточнюються, як правило, у такому алгоритмі:

$$\begin{cases} \rho_{кв}(A,X) = \sum\limits_{d=1}^{D} \dfrac{1}{\sigma_d^2}\left(f_{A,d} - f_{X,d}\right)^2; \quad \rho_e(A,X) = \sqrt{\rho_{кв}(A,X)}; \\ \rho_{кв}(B,X) = \sum\limits_{d=1}^{D} \dfrac{1}{\sigma_d^2}\left(f_{B,d} - f_{X,d}\right)^2; \quad \rho_e(B,X) = \sqrt{\rho_{кв}(B,X)}; \end{cases} \tag{5}$$

де $\rho_{кв}$ – квадратична метрика; ρ_e – евклідова метрика ; σ_d – однакове для обох класів стандартне відхилення для фактору з номером d.

Таким чином, даний алгоритм поширюється на клас задач, де класи відрізняються середніми значеннями, але масштаби даних можуть бути різними для різних факторів.

Стосовно випадків Big Data зробимо необхідне зауваження, що крок обчислення саме Евклідової метрики (тобто, операція взяття кореню з квадратичної метрики) є не тільки зайвою, але й дуже шкідливою з точки зору продуктивності алгоритмів, оскільки така операція потребує значної кількості машинних тактів. Втім, ці метрики є еквівалентними у тому сенсі, що якщо: $\rho_{кв}(A,X) = 0$, то і $\rho_e(A,X) = 0$; якщо: $\rho_{кв}(A,X) < \rho_{кв}(B,X)$, то і $\rho_e(A,X) < \rho_e(B,X)$. Доказ цього положення опускаємо за очевидністю.

Також, зауважимо, що при обробці Big Data досвідчений програміст ніколи не буде використовувати функції піднесення у другу ступень тієї чи іншої мови програмування типу X^2 або power(X,2). Як правило, він завжди використає замість цього простий вираз типу X*X.

Таким чином, результати навчання зводяться до зберігання у пам'яті лише 2D значень середніх по факторам класів та ще D значень стандартних відхилень. При цьому алгоритми навчання є такими, що оцінки цих значень сходяться зі швидкістю 1/N, що відомо з факту збіжності емпіричних середніх незалежних іспитів до математичних очікувань [9].

3. Задача дихотомії: алгоритм адаптивних метрик

Розглянемо швидкий алгоритм для вирішення задачі дихотомії у випадку, що показано на рис. 1.б. Спочатку обмежимось випадком лише одного фактору. Теоретична задача: знайти оптимальну межу між класами A і B, яка буде відповідати принципу рівноправності класів у тому сенсі, що ймовірність помилок P(A/B) та P(B/A) буде однаковою.

У даному випадку встановимо вирішальну границю R між класами (рис. 2). Оптимальне значення R буде таким, що ймовірності помилок відповідатимуть рівності:

$$\frac{P(B<R)}{P(A<R)} = P(A/B) = P(B/A) = \frac{P(A>R)}{P(B>R)}. \qquad (6)$$

Оскільки події *P(B<R)* та *P(B>R)* є несумісні та утворюють повну групу (аналогічно для подій *P(A<R)* та *P(A>R)*), то рівності (6) можна представити у еквівалентній формі:

$$\frac{P(B<R)}{1-P(A>R)} = \frac{P(A>R)}{1-P(B<R)}. \qquad (7)$$

Із співвідношення (7) витікає, що для виконання умови рівності помилок 1-го та 2-го роду достатньо рівності чисельників, оскільки рівність знаменників слідує з цього.

Нехай у даному фактор-просторі задана якась симетрична диференційна функція розподілення $p(x)$ з нульовим середнім та стандартним відхиленням σ_X. Функції розподілення класів A і B відрізняються від $p(x)$ зсувом та масштабом, тобто:

$$p_A(x) = p(q_A x - f_A) \;; \quad p_B(x) = p(q_B x - f_B) \;. \qquad (8)$$

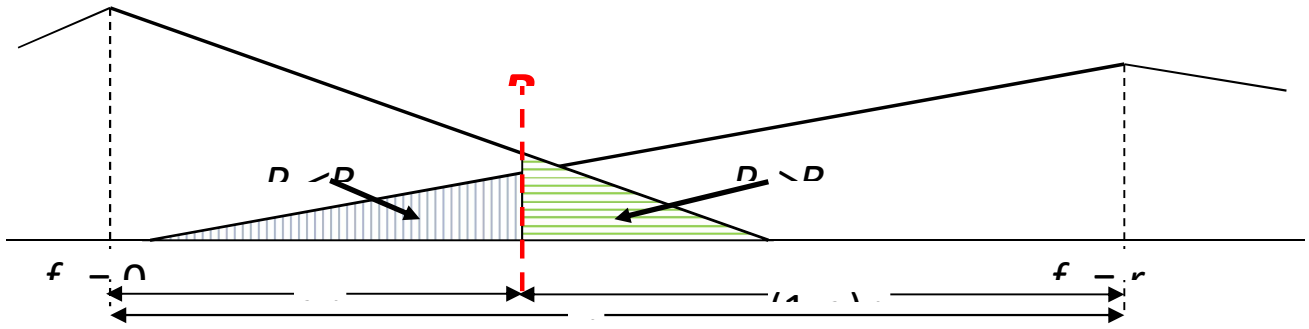


Рис. 2. Розподілення однакової форми, але з різними масштабами

Оскільки всі перетворення (8) є лише перетвореннями зсуву та масштабування, то без втрат спільності будемо вважати, що: $\sigma_X = 1$, $f_A = 0$, $f_B = r > 0$.

Можна довести, що ймовірність виходу випадкової величини X за межу « q сигм» від центру f залежить лише від виду функції розподілення та від числа q і не залежить від масштабу. Тоді із умов (7) витікає співвідношення:

$$q=\frac{\alpha r}{\sigma_A}=\frac{(1-\alpha)r}{\sigma_B}\Rightarrow \alpha r\sigma_B=\sigma_A-\alpha r\sigma_A\Rightarrow \alpha=\frac{\sigma_A}{\sigma_A+\sigma_B};\ 1-\alpha=\frac{\sigma_B}{\sigma_A+\sigma_B}. \quad (9)$$

Оптимальна межа буде визначатись як $R=\alpha r$, а вирішальне правило теж саме, як і у правилі (3), але відстані від центрів класів розраховуються наступним чином:

$$\rho_{кв}(A,X)=\sum_{d=1}^{D}\frac{1}{\sigma_{A,d}^2}\left(f_{A,d}-f_{X,d}\right)^2;\quad \rho_{кв}(B,X)=\sum_{d=1}^{D}\frac{1}{\sigma_{B,d}^2}\left(f_{A,d}-f_{X,d}\right)^2. \quad (10)$$

Порівняно з алгоритмом по п. 2 даний алгоритм призводить до необхідності додати лише значення стандартних відхилень для обох класів по всім факторам. Тобто для дихотомічного класифікатора треба зберігати в оперативній пам'яті 2D значень середніх та 2D значень стандартних відхилень.

Адаптація алгоритму за даними навчальних вибірок також може здійснюватися ітераційними процедурами типу (4). При цьому знов таки оцінки всіх параметрів виконуються з високою швидкістю порядку 1/N.

Втім, порівняно з алгоритмом по п. 2 даний алгоритм може бути використаний до більш широкого класу практичних задач. Так, наприклад ріст людей може бути не тільки у середньому різний для різних народів, але й відрізнятися значеннями стандартних відхилень.

Окрім цього, даний алгоритм також може застосовуватись у випадках, коли розподілення по факторам відрізняються не тільки зсувом та масштабом. З де якими уточненнями даний алгоритм можна застосовувати і для більш широкого класу практичних задач. Виходячи з нерівності Чебишева [9], можна вважати, що даний алгоритм буде давати невелику додаткову кількість помилок, якщо розподілення все-ж таки відрізняються за формою, але несуттєво.

4. Задача дихотомії: алгоритм адаптивних правил

У даному алгоритмі за результатами навчання заздалегідь встановлюються вирішальні границі між класами по окремим факторам згідно виразів (9). Нехай середні значення по 1-му фактору для класів А і В розміщено саме так, як на рис. 2, тобто в середньому фактор класу А менший за клас В ($f_{B,d}>f_{A,d}$). Тоді вирішальна границя буде в абсолютних значеннях знаходитись у точці

$$R_1=f_{A,1}+\alpha r=f_{A,1}+\frac{\sigma_{A,1}(f_{B,1}-f_{A,1})}{\sigma_{A,1}+\sigma_{B,1}}. \quad (11)$$

Тоді для прийняття рішення по даному фактору можна використовувати бінарну метрику:

$$\begin{cases} f_{X,1}<R_1\Rightarrow \rho_1(A,X)=0,\ \rho_1(B,X)=1 \\ f_{X,1}>R_1\Rightarrow \rho_1(A,X)=1,\ \rho_1(B,X)=0 \end{cases}, \quad (12)$$

по всій сукупності факторів визначаються відстані:

$$\rho(A,X)=\sum_{d=1}^{D}\rho_d(A,X),\quad \rho(A,X)=\sum_{d=1}^{D}\rho_d(B,X), \quad (13)$$

а рішення приймається за тим самим правилом (3).

Зауважимо, що операція порівняння у виразах (12) виконується за найскорішою машинною процедурою. Дійсно, для рішення треба визначити лише знак різниці ($R_k - f_{X,1}$), тобто визначити значення біту знака: якщо 0, то різниця позитивна, якщо 1 – негативна.

Тому даний алгоритм не потребує навіть операцій множення і є найшвидшим з попередньо розглянутих. Для задачі дихотомії за результатами навчання визначається при цьому лише пара чисел для кожного фактору:

$$.\lfloor f_{A,d}, R_d \rfloor, \quad d = 1,2,..,D .$$

Зрозуміло, що у випадку, якщо $f_{B,d} < f_{A,d}$ (центри класів міняються місцями порівняно з рис. 2), то змінюється і визначення вирішальної границі:

$$R_d = f_{B,d} + \alpha r = f_{B,1} + \frac{\sigma_{B,1}(f_{A,1} - f_{B,1})}{\sigma_{A,1} + \sigma_{B,1}} \tag{14}$$

та відповідно змінюються визначення метрик у виразах (12) на протилежні.

Зауважимо, що даний алгоритм дає однозначне рішення у випадку непарної кількості факторів. Також зауважимо, що визначені метрики (11-13) задають над простором класів метричний простір, тобто є суто метриками у розумінні аксіом:

$$\begin{gathered} \rho(A,A) = 0 \quad (аксіома\ тотожності) \\ \rho(A,B) \geq 0 \quad (аксіома\ позитивності) \\ \rho(A,B) = \rho(B,A) \quad (аксіома\ симетрії) \\ \rho(A,C) \leq \rho(A,B) + \rho(B,C) \quad (аксіома\ трикутника) \end{gathered}, \tag{15}$$

що доводиться тривіально.

Таким чином, для даного алгоритму також визначаються параметри, що залежать лише від середніх значень та значень стандартних відхилень по факторам.

5. Узагальнення сфери застосування швидких алгоритмів типу M-means

Випадок нестрогих нерівностей. У виразах вище для визначення правил класифікації навмисно всюди використано лише строгі нерівності. Якщо можливі значення факторів по всім класам мають дуже високу варіабельність, то ймовірність таких випадків близька до 0. Проте, якщо ці фактори можуть приймати дуже обмежену кількість значень наприклад, лише 10, то при невеликій розмірності фактор-простору ймовірність рівності метрик може бути високою. У даному випадку всі ці алгоритми корегуються: визначається особливий тип можливих рішень: відмова від класифікації, оскільки даний класифікатор не може прийняти будь-якого рішення. У таких випадках відмови класифікації або відправляються на «ручну» обробку, або використовується окремий алгоритм виключно для таких випадків. Наприклад, якщо алгоритм адаптивних правил дає відмову від класифікації, то застосовується алгоритм адаптивних метрик.

Випадок складних деформацій функцій розподілення ймовірностей. У даному випадку розподілення значень по факторам для різних класів відрізняються не тільки деформаціями типу зсуву та зміни масштабу, а й самі функції розподілення можуть мати дуже різну форму для різних класів. Втім, в окремих типах задач і такий випадок можна передбачити. Нехай є дві ***різні*** інтегральні функції розподілення $P_A(x)$ та $P_B(X)$. І нехай вони у метриці Чебишева [9] відрізняються на якусь невелику величину:

$$\max_{-\infty<x<\infty} \left| P_A(x) - P_B(x) \right| < \varepsilon ,$$

тоді ймовірність помилок 1-го та 2-го роду порівняно з методом по п.3 зросте не більше ніж на ε. Дане положення доводиться за допомогою нерівності Чебишева.

Випадок значної кількості класів. Тут передбачається, що кількість класів $M > 2$. Оскільки всі алгоритми за п.п. 2-5 є дуже швидкими, то навіть для значній кількості класів можливо їх використовувати у найпростішому варіанті дерев рішень [7].

Нехай задано класи A, B, C, D … Обираємо якийсь алгоритм з п.п. 2-5 (в залежності від особливостей розподілень факторів у класах). Далі використовуємо послідовне рішення задач дихотомії:

- відносно «невідомого» об'єкту X приймаємо рішення, до якого класу він ближчий – до A або B;
- допустимо, що рішення прийняте на користь класу A;
- тоді клас B виключається з подальшої обробки;
- вирішується задача дихотомії між класами A та C;
- один з цих класів виключається з подальшої обробки;
- і так далі до вичерпання всіх класів.

Очевидно, що кількість задач дихотомії буде складати $M - 1$, тобто алгоритм класифікації залишається «однопрохідним». Коректність даного алгоритму дерев рішень витікає з того, що всі задачі дихотомії вирішуються в єдиному метричному просторі. Те, що не треба повертатися до попередніх рішень саме цим і гарантовано: саме виконанням аксіоми трикутника у виразах (15).

При значній кількості класів найшвидший алгоритм адаптивних правил може мати недолік – необхідність зберігати у пам'яті попарні значення вирішальних границь R (заздалегідь, числа з плаваючою точкою типу float, double та т.ін.). Для вибору між вирішальними границями (11) або (14) треба зберігати ще пари ознак S (sign) співвідношення середніх по факторам (A>B або B>A) – дані типу boolean. Для прикладу 5-ти класів структура матриць даних лише для одного фактору представлена у табл. 1.

У даному випадку для кожного фактора, таким чином, треба зберігати 20 значень. Для фактор-простору розмірності 10 треба буде зберігати за результатами навчання вже 200 значень параметрів вирішальних правил. Тут знову ж таки проявляється характерний ефект Big Data – «прокляття розмірності», оскільки для M класів у просторі D факторів треба буде зберігати D x M x (M-1) даних.

Тобто об'єм оперативної пам'яті для зберігання цих параметрів збільшується у квадратичній залежності від кількості класів. Наприклад, якщо є відносно невелика кількість класів M = 10000, які задано у фактор-просторі розмірності D = 10, то треба буде зберігати 999900000 (майже мільярд!) значень параметрів.

Таблиця 1.

Приклад структури матриць даних для одного фактору та для 5 класів

Класи	A	B	C	D	A	B	C	D
	Вирішальні границі				Ознаки рішень			
B	R_{AB}				S_{AB}			
C	R_{AC}	R_{BC}			S_{AC}	S_{BC}		
D	R_{AD}	R_{BD}	R_{CD}		S_{AD}	S_{BD}	S_{CD}	
E	R_{AE}	R_{BE}	R_{CE}	R_{DE}	S_{AE}	S_{BE}	S_{CE}	S_{DE}

Вихід з даного положення – розрахунок вирішальних границь під час виконання програми. Тоді кількість даних в оперативній пам'яті буде аналогічна до кількості даних алгоритму адаптивних метрик. При вирішенні конкретних задач, таким чином, треба буде приймати компромісне рішення між швидкістю та розмірами масивів параметрів класифікаторів.

Випадок динамічних класів. Дуже велика кількість практичних задач вирішується для класів, характеристики яких змінюються з часом. Втім, завдяки незначної кількості характеристик класів, методи M-means можуть бути розповсюджені і на даний випадок. Відмінність застосування полягає лише у тому, що під час навчання алгоритмів встановлюються не фіксовані значення

параметрів розподілень, а параметри їх динамічних моделей виду $f_{m,d}(t)$ та $\sigma_{m,d}(t)$. У найпростішому вигляді, наприклад, для короткострокової екстраполяції, дані залежності можна представити лінійними функціями:

$$f_{m,d}(t) = f_{0,m,d} + f_{1,m,d} \cdot t, \quad \sigma_{m,d}(t) = \sigma_{0,m,d} + \sigma_{1,m,d} \cdot t. \qquad (16)$$

Задача класифікації у даному випадку вирішується за допомогою прогнозованих значень характеристик класів. При цьому для моделей залежностей виду (16) обсяг пам'яті порівняно з випадком статичних класів підвищується лише вдвічі.

У практичних задачах залежності характеристик класів від часу можуть бути більш складними, ніж лінійні функції (16). Для таких випадків можна застосовувати відомі методи штучного інтелекту, наприклад, метод групового врахування аргументів [10]: такі методи дозволяють як мінімізувати кількість вільних параметрів моделей, але й також вибрати структури моделей аз критеріями екстраполяційної стійкості.

6. Порівняння алгоритмів класифікації за показниками надійності та продуктивності

У даному пункті наведемо лише один приклад порівняння характеристик надійності та продуктивності деяких алгоритмів класифікації. Порівняння здійснюється у задачі дихотомії, де фактор-простір має розмірність D = 6. Розглядається, таким чином 2 класи: А (умовно зелений) та В (умовно синій). Задача вирішується за допомогою імітаційної моделі (більш докладно розглянуто у статті [11]).

Окрім алгоритмів адаптивних метрик та адаптивних правил також розглядається відомий алгоритм К найближчих сусідів та алгоритм К випадкових сусідів. Останній відрізняється тим, що з серед представників класів обирається незначна кількість випадкових елементів. Рішення приймається шляхом обчислення відстаней від невизначеного об'єкту Х до всіх К випадкових елементів по класам А і В. Порівнюються або суми відстаней, або середнє значення відстаней.

Приклад апріорних розподілень значень факторів наведено у вигляді гістограм в розрізі факторів (рис. 3). Тут червоним показано в умовному масштабі гістограми помилок.

Як бачимо, задача вирішується у дуже складних умовах. Масштаби значень для «зеленого» та «синього» класів відрізняються суттєво: вдвічі! Окрім того, половина значень «синього» класу перекриває всі можливі значення «синього» класу по всім факторам.

Задача вирішувалась на множинах елементів N=10000 для кожного класу. Тестування відбувалось на 1000 «невідомих» об'єктів для кожного класу. Результати класифікації для даного прикладу дано у табл. 2.

Аналіз даних у табл. 2 дозволяє зробити такі висновки:

- найменшу загальну кількість помилок дає все-ж таки алгоритм К найближчих сусідів;
- найбільшу кількість помилок дає алгоритм К випадкових сусідів;
- алгоритми К найближчих сусідів, К випадкових сусідів центрів класів у даному випадку дають «перекіс» частоти помилок у бік помилок 2-го типу: перевага частіше віддається класу А, хоча тестовий об'єкт насправді відносився до класу В;
- алгоритм адаптивних метрик та адаптивних правил у найбільший ступені забезпечують виконання принципу «рівноправності» класів;
- загальна кількість помилок алгоритмів К найближчих сусідів, адаптивних метрик та адаптивних правил мають приблизно однаковий порядок;
- що стосується часу прийняття рішень, то тут, як кажуть, «коментарі зайві»: алгоритми, засновані на принципах M-means суттєво більш ефективні за критерієм продуктивності;
- найбільш швидким алгоритмом із розглянутих є алгоритм адаптивних правил.

Таблиця 2.

Результати визначення характеристик надійності та продуктивності різних алгоритмів класифікації

Метод класифікації	Кількість К сусідів	Час рішення, мс	Похибки B/A	Похибки A/B	Загальний процент помилок
К найближчих сусідів	100	155560	2 / 998	109 / 891	5,55
К випадкових сусідів	100	1618	5 / 995	484 / 516	24,45
Центрів класів	1	11	12 / 988	137 / 863	7,45
Адаптивних метрик	1	16	61 / 939	58 / 942	5,95
Адаптивних правил	1	1	59 / 941	59 / 941	5,70

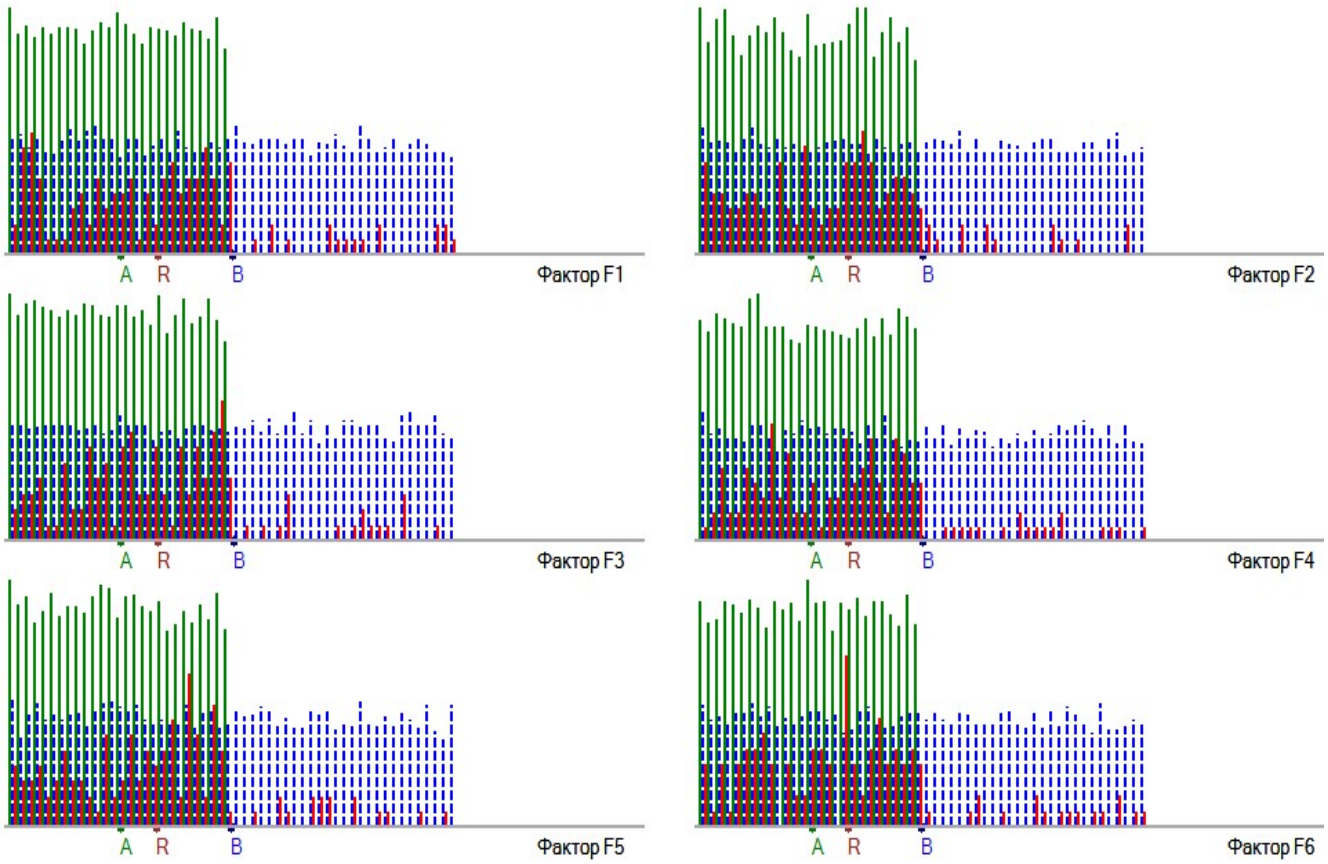


Рис. 3. Гістограми розподілу значень для 6-ти факторів

Різниця часу прийняття рішень всіма цими алгоритмами відрізняється, як бачимо, суттєво. Причини цього очевидні: алгоритм адаптивних метрик витрачає значно більше значення часу прийняття рішень, ніж адаптивних правил тому що використовує більш тривалі операції обчислення метрик – потребує операцій множення.

Алгоритми К найближчих сусідів та К випадкових сусідів використовують по суті ти ж самі метрики, що і алгоритм адаптивних метрик. Втім, кількість операцій підвищується приблизно пропорційно тій самій кількості К сусідів. Тому у табл. 2 і спостерігаються відповідні співвідношення: час прийняття рішень алгоритмом К випадкових сусідів приблизно у 100 разів більший, ніж час прийняття рішень алгоритмом адаптивних метрик. Відповідно, час прийняття рішень алгоритмом К найближчих сусідів вже у 10000 разів більший.

Висновки

1. Головний висновок полягає у тому, що різні методи (алгоритми) класифікації заздалегідь не є «поганими» чи «добрими»: вирішення практичних задач потребує вибору найбільш ефективного алгоритму для даної конкретної задачі.

2. У задачах класифікації великих обсягів даних (Big Data) поряд з критерієм ефективності алгоритмів – надійності, слід обов'язково враховувати критерій продуктивності. Якщо по критерію надійності алгоритми приблизно рівні, то однозначно слід надавати перевагу більш продуктивним алгоритмам.

3. Для задач класифікації Big Data з надзвичайно великою кількістю представників класів, середньою кількістю класів, невеликою розмірністю фактор-простору значні переваги мають алгоритми, побудовані за принципами M-means. При цьому всі елементи класів замінюються незначною кількістю характеристик класів.

4. В простих випадках, оцінюючи ефективність за всіма критеріями, слід віддати перевагу алгоритмам адаптивних метрик або адаптивних правил. При цьому останній навіть при кількості класів порядку десятків тисяч може давати суттєвий виграш у часі прийняття рішень, що видно вже з наведеного прикладу.

5. Принципи M-means дозволяють адаптувати алгоритми класифікації до дуже великої сфери практичних задач, у тому числі на випадок динамічних класів. Тому такі алгоритми мають перспективи широкого застосування для ефективного вирішення зазначеного типу задач, а продовження досліджень у цьому напрямку є актуальною науковою задачею.

Література

1. Pijush Kanti Dutta Pramanik, Saurabh Pal, Moutan Mukhopadhyay, Simar Preet. Big Data classification: techniques and tools. *Applications of Big Data in Healthcare*, Academic Press, 2021. P. 1-43.
2. Chenxia Jin, Fachao Li, Shijie Ma, Ying Wang. Sampling scheme-based classification rule mining method using decision tree in big data environment. *Knowledge-Based Systems*, Vol. 244, 2022.
3. Nooshin Hanafi, Hamid Saadatfar. A fast DBSCAN algorithm for big data based on efficient density calculation. *Expert Systems with Applications*, Vol. 203, 2022, 117501, ISSN 0957-4174.
4. Chaoyu Gong, Zhi-gang Su, Pei-hong Wang, Qian Wang, Yang You. Evidential instance selection for K-nearest neighbor classification of big data. *International Journal of Approximate Reasoning*, Vol. 138, 2021. Pages 123-144.
5. Ayman Taha, Ali S. Hadi, Bernard Cosgrave, Susan McKeever. A multiple association-based unsupervised feature selection algorithm for mixed data sets. *Expert Systems with Applications*, Vol. 212, 2023, 118718, ISSN 0957-4174..
6. Qian Tao, Chunqin Gu, Zhenyu Wang, Daoning Jiang. An intelligent clustering algorithm for high-dimensional multiview data in big data applications. *Neurocomputing*, Vol. 393, 2020, Pages 234-244, ISSN 0925-2312,
комп'ютерних наук та кібернетики. Київ: КНУ, 2017. 150 с.
8. Abiodun M. Ikotun, Absalom E. Ezugwu, Laith Abualigah, Belal Abuhaija, Jia Heming. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, Vol. 622, 2023, Pages 178-210, ISSN 0020-0255..
9. Огірко О.І., Галайко Н.В. Теорія ймовірностей та математична статистика: навч. посібн. Львів: ЛьвДУВС, 2017. 292 с.
10. Ивахненко А.Г., Степашко В.С. Помехоустойчивость моделирования. Киев: Наукова думка, 1985. 216 с.
11. Одегов М.А., Гаджиєв М.М., Буката Л.М., Глазунова Л.В., Кочеткова М.В. Порівняння алгоритмів класифікації Big Data методами імітаційного моделювання. *Інфокомунікаційні та комп'ютерні технології*. Київ, 2023. No 1 (стаття у даному випуску журналу).

References

1. Pijush Kanti Dutta Pramanik, Saurabh Pal, Moutan Mukhopadhyay, Simar Preet Singh. 1 - Big Data classification: techniques and tools. *Applications of Big Data in Healthcare*, Academic Press, 2021. P. 1-43.
2. Chenxia Jin, Fachao Li, Shijie Ma, Ying Wang. Sampling scheme-based classification rule mining method using decision tree in big data environment, *Knowledge-Based Systems*, Vol. 244, 2022.
3. Nooshin Hanafi, Hamid Saadatfar. A fast DBSCAN algorithm for big data based on efficient density calculation. Expert *Systems with Applications*, Vol. 203, 2022, 117501, ISSN 0957-4174.
4. Chaoyu Gong, Zhi-gang Su, Pei-hong Wang, Qian Wang, Yang You. Evidential instance selection for K-nearest neighbor classification of big data. *International Journal of Approximate Reasoning*, Vol. 138, 2021. Pages 123-144..
5. Ayman Taha, Ali S. Hadi, Bernard Cosgrave, Susan McKeever. A multiple association-based unsupervised feature selection algorithm for mixed data sets. *Expert Systems with Applications*, Vol. 212, 2023, 118718, ISSN 0957-4174..
6. Qian Tao, Chunqin Gu, Zhenyu Wang, Daoning Jiang. An intelligent clustering algorithm for high-dimensional multiview data in big data applications. *Neurocomputing*, Vol. 393, 2020, Pages 234-244, ISSN 0925-2312,

7. Marchenko O.O., Rossada T.V. Aktualni problemi Data Mining: navch. posibn. dlya studentiv fakultetu komp'yuternih nauk ta kibernetiki. Kiyiv: KNU, 2017. 150 s.
8. Abiodun M. Ikotun, Absalom E. Ezugwu, Laith Abualigah, Belal Abuhaija, Jia Heming. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, Vol. 622, 2023, Pages 178-210, ISSN 0020-0255.
9. Ogirko O.I., Galajko N.V. Teoriya jmovirnostej ta matematichna statistika: navch. posibn. Lviv: LvDUVS, 2017. 292 s.
10. Ivahnenko A.G., Stepashko V.S. Pomehoustojchivost modelirovaniya. Kiev: Naukova dumka, 1985. 216 s.
11. Odegov M.A., Hadzhyiev M.M., Bukata L.M., Glazunova L.V., Kochetkova M.V. Porivnyannya algoritmiv klasifikaciyi Big Data metodami imitacijnogo modelyuvannya. *Infokomunikacijni ta komp'yuterni tehnologiyi.* Kiyiv, 2023. No 1 (stattya u danomu vipusku zhurnalu).