```
## Warning in grepl(db, input): input string 43 is invalid in this local
## Warning in grepl(db, input): input string 44 is invalid in this local
## Warning in grepl(db, input): input string 45 is invalid in this local
## Warning in grepl(db, input): input string 48 is invalid in this local
## Warning in grepl(db, input): input string 53 is invalid in this local
## Warning in grep("^\\\\bibliography.+", input, value = TRUE): input
string 43 is invalid in this locale
## Warning in grep("^\\\\bibliography.+", input, value = TRUE): input
string 44 is invalid in this locale
## Warning in grep("^\\\\bibliography.+", input, value = TRUE): input
string 45 is invalid in this locale
## Warning in grep("^\\\\bibliography.+", input, value = TRUE): input
string 48 is invalid in this locale
## Warning in grep("^\\\\bibliography.+", input, value = TRUE): input
string 53 is invalid in this locale
```

РОЗДІЛ 1
КАСКАДНА НЕЙРОННА МЕРЕЖА, ЩО ЕВОЛЮЦІОНУЄ, ДЛЯ ПОСЛІДОВНОГО НЕЧІТКОГО КЛАСТЕРУВАННЯ ПОТОКІВ ДАНИХ

У цьому розділі описані архітектура та методи навчання пропонованої каскадної нейро-мережі для нечіткого кластерування, зокрема потоків даних; проведено аналіз існуючих систем, що еволюціонують, для кластерування даних, зокрема нечіткого, і розглянуті особливості та труднощі послідовного кластерування, та описні два підходи, переваги яких поєднує у собі пропонована система: нечітке та ієрархічне кластерування.

1.1. Труднощі та особливості відомих методів кластерування даних

Завдання кластерування (класифікації без вчителя) досить часто зустрічається в багатьох додатках, пов'язаних з видобутком знань, де у режимі самонавчання необхідно розбити деякий вхідний нерозмічений масив даних на однорідні в прийнятому сенсі групи. Розглянемо деякі ієрархічні та розподільні методи кластерування, адже, як буде показано далі, пропована у цьому розділі самонавчанна система поєднує у собі переваги обох підходів.

Розподільні методи кластерування (чи то жорсткі, чи нечіткі) можна назвати динамічними у тому сенсі, що належність певного образу до певного кластеру (кластерів для нечіткої модифікації) не є постійною. Нездатність методів розподільного кластерування самостійно визначити кількість кластерів у певному сенсі компенсується тим, що знання форми чи розміру кластерів може стати у нагоді на етапі вибору відповідних прототипів та насамперед типу відстані (міри схожості) і суттєво поліпшити кінцеве розбиття вибірки. Але, варто зазначити чутливість таких методів до початкової ініціалізації,

шуму і викидів, їх сприйнятливість до локальних мінімумів, адже вони ґрунтуються на оптимізації певної цільової функції. Типові методи розподільного кластерування мають обчислювальну складність $\mathcal{O}(N)$ для тренувальної вибірки розміру N [15].

Серед методів ієрархічного кластерування виділяють два основних типи: висхідні та спадні методи. Спадні методи працюють за принципом «зверху-вниз»: на початку припускається, що всі образи належать до одного кластеру, який потім розбивається на все більш дрібні кластери. Більш поширеними є висхідні алгоритми, які на початку роботи поміщають кожен об'єкт до окремого кластеру, а потім об'єднують кластери у все більш крупні, доки усі образи не матимуть свій власний кластер. Таким чином будується система вкладених розбиттів. Результати таких алгоритмів зазвичай представляють у вигляді дерева - дендрограми (тут можна провести аналогію між висхідними та спадними методами ієрархічного кластерування та конструктивними і деструктивними системами, що еволюціонують. У цій роботі здебільшого розглядається конструктивний підхід, тому пропонована самонавчанна система є у певному сенсі альтернативою системам висхідного ієрархічного кластерування, що здатна працювати у режимі реального часу).

Для обчислення відстаней між кластерами використовуються такі відстані:

— одинарний зв'язок (відстань найближчого сусіда): відстань між двома кластерами визначається відстанню між двома найбільш близькими об'єктами (найближчими сусідами) у різних кластерах. Результуючі кластери мають тенденцію об'єднуватися в ланцюжки;

— повний зв'язок (відстань найбільш віддалених сусідів): відстані між кластерами визначаються найбільшою відстанню між будь-якими двома об'єктами різних кластерів (тобто найбільш віддаленими сусідами). Цей метод зазвичай працює дуже добре, коли об'єкти походять з окремих груп. Якщо ж кластери мають видовжену форму або їх природний тип є «ланцюжковим» цей метод непридатний;

— незважене попарне середнє: відстань між двома різними кластерами обчислюється як середня відстань між усіма парами об'єктів у них. Метод ефективний, коли об'єкти формують різні групи, проте він працює однаково добре і у випадках протяжних («ланцюжкового» типу) кластерів;
— зважене попарне середнє: метод ідентичний методу незваженого попарного середнього, за винятком того, що при обчисленнях розмір відповідних кластерів (тобто число об'єктів, що містяться в них) використовується у якості вагового коефіцієнту. Тому доцільно використовувати даний метод у випадку нерівних за розміром кластерів;
— незважений центроїдний метод: у цьому методі відстань між двома кластерами визначається як відстань між їх центрами тяжкості;
— зважений центроїдний метод (медіана): цей метод ідентичний попередньому, за вийнятком того, що при обчисленнях використовуються ваги для обліку різниці між розмірами кластерів. Тому, якщо є або підозрюються значні відмінності в розмірах кластерів, цей метод має перевагу над попереднім

Порівняно з розподільним кластеруванням, методи ієрархічного кластерування легко ідентифікують викиди, не потребують визначеної кількості кластерів та нечутливі до початкової ініціалізації чи локальних мінімумів. До недоліків варто віднести нездатність методів визначати кластери, що перекривають один одного. Крім того, ієрархічне кластерування є статичним, тобто образи віднесені до певного кластеру на ранніх стадіях не можуть бути пізніше належити іншому, що унеможливлює створення модифікацій методів для послідовного кластерування, на відміну від розподільного кластерування. Методи ієрархічного кластерування здебільшого мають обчислювальну складність принаймні $\mathcal{O}\left(N^2\right)$, що робить їх використання недоцільним для великих масивів даних.

1.1.1. Нечітке послідовне кластерування. Традиційний підхід до завдання кластерування припускає, що кожне спостереження належить лише одному кластерові, в той час як більш природною видається ситуація, коли кожен вектор-спостереження оброблюваної вибірки можна віднести відразу декільком класам з різними рівнями належності. Така ситуація є предметом розгляду нечіткого кластерного аналізу [1, 6, 16, 18, 25, 38], а для його вирішення широко використовується апарат обчислювального інтелекту [9-12] і, насамперед, нейро-фаззі підхід [13]. При цьому більшість алгоритмів нечіткої кластеризації призначені для роботи в пакетному режимі, коли усі дані, що підлягають обробці, задані апріорно. Вихідною інформацією для такої задачі є вибірка спостережень, сформована з m-вимірних векторів ознак $x(1), x(2), \ldots, x(1), ;x(N)$, при цьому для зручності чисельної реалізації вихідні дані попередньо деяким чином перетворюються, наприклад, так, щоб всі спостереження належали до гіперкубу $[-1,1]^n$ або одиничній гіперсфері $\|x(k)\|^2$.

Результатом такого кластерування є розбиття масиву вихідних даних на M кластерів з певним рівнем належності $u_J(k)$ k-ого вхідного образу $x(k)$ до J-ого кластеру $(J = 1, 2, \ldots, M)$. Передбачається, що N та M, а також параметри кластерування (в першу чергу, фаззіфікатор) задані апріорі і не змінюються під час обробки даних. Варто зауважити, що існує широкий клас задач динамічного інтелектуального аналізу даних і потоків даних (Dynamik Data Mining, Data Stream Mining) [2, 4, 10, 20, 24, 30, 36] у випадку, коли дані надходять у вигляді послідовного потоку в онлайн режимі. Отже, кількість вхідних образів N у цьому випадку не обмежується, а k набуває значення поточного дискретного часу.

Самоорганізовні мапи Кохонена [22] добре пристосовані для вирішення завдання кластерування в онлайн режимі. Ці нейронні мережі мають один шар латеральних з'єднань та навчаються за принципами «переможець отримує все» або «переможець отримує більше». Самоорганізовні мапи також відомі своєю ефективністю вирішення задачі кластерування класів, що перети-

наються. Тому, у зв'язку з дедалі більшою кількістю завдань кластерування потоків даних, з'явилися самонавчанні нейро-фазі гібридні системи, що у деякому сенсі поєднують у собі самоорганізовні мапи Кохонена (SOM) та метод нечітких c-середніх Бездека [5, 7, 12, 13, 19, 23, 27, 28, 31–34]. Такі гібридні системи володіють обширною функціональністю завдяки використанню спеціальних алгоритмів налаштування, що ґрунтуються на процедурах оптимізації прийнятої цільової функції, але потребують попередньо заданої кількості кластерів та фіксованого значення фаззіфікатора.

1.2. Критерії дійсності нечіткого кластерування

Оскільки коефіцієнт розбиття залежить лише від значень функції належності, йому властиві деякі недоліки. Коли фаззіфікатор наближається до 1, індекс дійсності буде однаковим для ус!іх c, коли фаззіфікатор наближається до ∞.

Індекс розбиття ентропії PE (Partition Entorpy Index) - ще один критерій дійсності нечіткого кластерування, запропонований (Bezdek, 1974a, 1981), що залежить лише від значень фунції належності

$$PE = -\frac{1}{N}\sum_{l=1}^{M}\sum_{i=1}^{N} u_{li}\log_a(u_{li}). \tag{1.1}$$

Індекс ентропії розбиття набуває значень у інтервалі $[0, \log_a M]$. Що ближче значення PE до 0, то жорсткіше розбиття вхідних даних. Значення PE близькі до верхньої межі вказують на відсутність будь-якої структури, притаманної набору вхідних даних, або на нездатність методу її виявити. Індекс ентропії розбиття має ті самі недоліки, що і коефіцієнт розбиття. Оптимальній кількості кластерів M^* відповідає мінімальне значення (1.1).

Фукуяма та Суґено запропонували індекс дійсності нечіткого кластерування, залежний як від рівнів належності так і від самих вхідних даних:

$$FS = \sum_{i=1}^{N}\sum_{l=1}^{M} u_{li}^{\beta}\left(\|x_i - z_l\|^2 - \|z_l - z\|^2\right), \tag{1.2}$$

де z та z_l – середнє арифметичне усієї виборки та образів віднесених до кластеру M_l відповідо. З визначення (1.2) видно, що малі значення FS говорять про компактні добре визначені кластери.

Нечітка множина i-ого образу визначається як

$$\tilde{A}_l = \sum_{i=1}^{N} \frac{u_{li}}{x_i}, l = 1, 2, \ldots, M. \tag{1.3}$$

Ступінь, в якій A_l є підмножиною A_p визначається наступним чином

$$\begin{cases} S\left(\tilde{A}_l, \tilde{A}_p\right) = \dfrac{U\left(\tilde{A}_l \cap \tilde{A}_p\right)}{U\left(\tilde{A}_l\right)}, \\ U\left(\tilde{A}_j\right) = \sum\limits_{i=1}^{N} u_{ji}. \end{cases} \tag{1.4}$$

Зважаючи на (1.4), можна запропонувати такі варіанти обчислення міри подібності:

$$N_1\left(\tilde{A}_l, \tilde{A}_p\right) = \frac{S\left(\tilde{A}_l, \tilde{A}_p\right) + S\left(\tilde{A}_p, \tilde{A}_l\right)}{2}, \tag{1.5a}$$

$$N_2\left(\tilde{A}_l, \tilde{A}_p\right) = \min\left(S\left(\tilde{A}_l, \tilde{A}_p\right), S\left(\tilde{A}_p, \tilde{A}_l\right)\right), \tag{1.5b}$$

$$N_3\left(\tilde{A}_l, \tilde{A}_p\right) = S\left(\tilde{A}_l \cup \tilde{A}_p, \tilde{A}_p \cap \tilde{A}_l\right). \tag{1.5c}$$

Тоді індекс дійсності кластерування, що грунтується на нечіткій подібності, можна визначити як

$$FSim = \max_{1\le l\le M} \max_{1\le p\le M, p\ne l} N\left(\tilde{A}_l, \tilde{A}_p\right), \tag{1.6}$$

де міру нечіткої подібності $N\left(\tilde{A}_l, \tilde{A}_p\right)$ можна знайти за будь-яким виразом (1.5).

1.3. Архітектура каскадної мережі, що еволюціонує, для нечіткого кластерування

До нульовго шару системи послідовно передаються дані у формі векторного сигналу $x(k) = (x_1(k), x_2(k), \ldots, x_1(k))^T$, де $k = 1, 2, \ldots, N, N+1, \ldots$ — індекс поточного дискретного часу. Вхідні сигнали надходять до всіх вузлів системи $N_j^{[m]}$, де $j = 1, 2, \ldots, q$ - кількість вузлів у пулі-ансамблі, $m = 1, 2, \ldots$ — номер каскаду. Вузол кожного каскаду призначений для онлайн кластерування потоку данних і відрізняється від вузлів-сусідів використаним алгоритмом навчання або, у випадку спільного методу кластерування, параметрами алгоритму. Кількість кластерів для кожного каскаду є відомою і дорівнює $m+1$. Елемент $PC_j^{[m]}$ дає оцінку якості кластерування кожного вузла у пулі, а елемент $PC^{*[m]}$ визначає найкращий елемент у пулі кожного каскаду. Елемент системи $XB^{[m]}$ оцінює загальную якість кластеризації пула, враховуючи прийняту кількість кластерів $m+1$. Таким чином, система розв'язує задачу кластерування нестаціонарного потоку даних в умовах невизначенності щодо кількості кластерів, а також їх вигляду і рівню взаємного перекриття. І, нарешті, вихідний вузол системи XB^*, порівнюючи якість кластеризації кожного з каскадів, виділяє найкращий результат — кількість кластерів, їх центроїди-прототипи та рівні належності кожного спостереження до кожного з сформованих центроїдів. Незважаючи на удавану громіздкість, чисельна реалізація запропонованої архітектури не викликає принципових труднощів завдяки тому, що потік даних, що надходить до системи, може оброблятися у паралельному режимі вузлами системи $N_j^{[m]}$[2, 4].

1.4. Адаптивне навчання вузлів каскадної нейро-фаззі системи, що еволюціонує

В основі алгоритмів навчання вузлів системи лежать алгоритми нечіткого кластерування, засновані на цільових функціях, такі, що вирішують задачу їх оптимізації при деяких апріорних припущеннях. Найбільш поширеним є

ймовірнісний підхід, заснований на мінімізації цільової функції

$$E\left(u_{jl}^{[m]}(k),\, c_{jl}^{[m]}\right) = \sum_{k=1}^{N}\sum_{l=1}^{m+1}\left(u_{jl}^{[m]}(k)\right)^{\beta}\left\|x(k) - c_{jl}^{[m]}\right\|^2 \tag{1.7}$$

при обмеженнях

$$\sum_{l=1}^{m+1}(k) = 1, \quad 0 \le \sum_{k=1}^{N} u_{jl}^{[m]}(k) \le N \tag{1.8}$$

де $u_{ij}^{[m]}(k) \in [0,1]$ — рівень належності спостереження $x(k)$ до l-ого кластеру у j-ому вузлі каскаду m,

$c_{jl}^{[m]}$ — $(n \times 1)$ - вимірній вектор-центроїд l-ого кластеру у j-ому вузлі каскаду m,

$\beta > 1$ — параметр фаззіфікації (фаззіфікатор), що визначає розмитість границь між кластерами,

$k = \overline{1, N}$ — номер образу (N — кількість образів у вхідній виборці, що, у рамках класичного подходу Бездека, вважається незмінною та такою, що задана апріорі).

Вводячи функцію Лагранжа

$$\begin{aligned} L\left(u_{jl}^{[m]}(k),\, c_{jl}^{[m]},\, \lambda_j^{[m]}(k)\right) = &\sum_{k=1}^{N}\sum_{l=1}^{m+1}\left(u_{jl}^{[m]}(k)\right)^{\beta}\left\|x(k) - c_{jl}^{[m]}\right\|^2 + \\ &+ \sum_{k=1}^{N}\lambda_j^{[m]}(k)\left(\sum_{l=1}^{m+1} u_{jl}^{[m]}(k) - 1\right) \end{aligned} \tag{1.9}$$

(тут $\lambda_j^{[m]}(k)$ — невизначений множник Лагранжа) та розв'язавши систему рівнянь Каруша-Куна-Таккера, нескладно отримати шукане рішення у вигляді

$$\begin{cases} u_{jl}^{[m]}(k) = \dfrac{\left(\left\|x(k) - c_{jl}^{[m]}\right\|^2\right)^{\frac{1}{1-\beta}}}{\sum\limits_{l=1}^{m+1}\left(\left\|x(k) - c_{jl}^{[m]}\right\|^2\right)^{\frac{1}{1-\beta}}}, \\ c_{jl}^{[m]} = \dfrac{\sum\limits_{k=1}^{N}\left(u_{jl}^{[m]}(k)\right)^{\beta} x(k)}{\sum\limits_{k=1}^{N}\left(u_{jl}^{[m]}(k)\right)^{\beta}}, \\ \lambda_j^{[m]}(k) = -\left(\left(\sum\limits_{l=1}^{m+1}\beta\left\|x(k) - c_{jl}^{[m]}\right\|^2\right)^{\frac{1}{1-\beta}}\right)^{\frac{1}{1-\beta}}, \end{cases} \tag{1.10}$$

що при $\beta = 2$ збігається з алгоритмом нечітких с-середніх Бездека (FCM) [17] і приймає форму

$$\begin{cases} u_{jl}^{[m]}(k) = \dfrac{\left\|x(k) - c_{jl}^{[m]}\right\|^{-2}}{\sum\limits_{l=1}^{m+1}\left\|x(k) - c_{jl}^{[m]}\right\|^{-2}}, \\ c_{jl}^{[m]} = \dfrac{\sum\limits_{k=1}^{N}\left(u_{jl}^{[m]}(k)^2 x(k)\right)}{\sum\limits_{k=1}^{N}\left(u_{jl}^{[m]}(k)\right)^2}. \end{cases} \tag{1.11}$$

Тут варто відзначити, що вибір фаззіфікатора $\beta = 2$ в (1.11) не дає жодних переваг порівняно з довільним значенням β у (1.10), у зв'язку з чим пропонується використовувати різні значення параметра фаззіфікації для кожного вузла пулу каскаду, після чого вибирати найкращий результат залежно від прийнятого критерію якості нечіткого кластерування [26, 29, 35].

Для послідовної обробки потоку даних, що надходять в online режимі, у [8, 9] були запропоновані рекурентні алгоритми, в основі яких лежить процедура нелінійного програмування Ерроу-Гурвіца-Удзави [3]. Так, пакетному алгоритмові (1.10) відповідає вираз

$$\begin{cases} u_{jl}^{[m]}(k+1) = \dfrac{\left\| x(k+1) - c_{jl}^{[m]}(k) \right\|^{\frac{1}{1-\beta}}}{\sum\limits_{l=1}^{m+1} \left\| x(k+1) - c_{jl}^{[m]}(k) \right\|^{\frac{1}{1-\beta}}}, \\ c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \eta(k+1) \left(u_{jl}^{[m]}(k+1) \right)^{\beta_j} \left(x(k+1) - c_{jl}^{[m]}(k) \right), \end{cases} \tag{1.12}$$

(тут $\eta(k+1)$ — параметр кроку навчання), що є узагальненням алгоритму навчання Чанга-Лі [11] і при $\beta = 2$ близьке до ґрадієнтної процедури Парка-Деера [27].

$$\begin{cases} u_{jl}^{[m]}(k+1) = \dfrac{\left\| x(k+1) - c_{jl}^{[m]}(k) \right\|^{-2}}{\sum\limits_{l=1}^{m+1} \left\| x(k+1) - c_{jl}^{[m]}(k) \right\|^{-2}}, \\ c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \eta(k+1) \left(u_{jl}^{[m]}(k+1) \right)^{2} \left(x(k+1) - c_{jl}^{[m]}(k) \right). \end{cases} \tag{1.13}$$

Варто зауважити, що, розглянувши співвідношення (1.12) з позицій навчання Кохоненової самоорганізованої мапи (SOM) [22], можна помітити, що множник $\left(u_{jl}^{[m]} \right)^{\beta_j}$ відповідає функції сусідства в правилі навчання на основі принципу «переможцю дістається більше», маючи при цьому дзвонуватий вигляд.

Вочевидь, у випадку, коли $\beta_j = 1$ та $u_{jl}^{[m]}(k) \in [0,1]$, процедура (1.12) збігається з чітким алгоритмом c-середніх (HCM), коли ж $\beta_j = 0$, маємо стандартне правило навчання Кохонена «переможцю дістається все» [22]

$$c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \eta(k+1) \left(x(k+1) - c_{jl}^{[m]}(k) \right), \tag{1.14}$$

запропоноване Каш'япом та Блейдоном [21] у шістдесятих роках минулого століття. Легко побачити, що процедура (1.14) оптимізує цільову функцію

$$E\left(c_{jl}^{[m]}\right) = \sum_{k=1}^{N} \left\| x\left(k\right) - c_{jl}^{[m]} \right\|^2, \quad \sum_{l=1}^{m+1} N_l = N, \tag{1.15}$$

мінімум якої збігається із середнім арифметичним

$$c_{jl}^{[m]} = \frac{1}{N} \sum_{k=1}^{N_l} x\left(k\right), \tag{1.16}$$

де N_l — кількість векторів, віднесених до l-го кластеру у процесі конкуренції.

Якщо записати (1.16) у рекурентній формі, отримаємо оптимальний алгоритм самонавчання Ципкіна [14]

$$c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \frac{1}{N_l(k+1)} \left(x\left(k+1\right) - c_{jl}^{[m]}(k) \right), \tag{1.17}$$

де $N_l\left(k+1\right)$ — число векторів, віднесених до l-го кластеру в $k+1$-й момент реального часу, що є стандартною процедурою стохастичної апроксимації.

У загальному випадку алгоритм навчання (1.12) вузла можна розглядати як правило самонавчання нечіткої модифікації самоорганізовної мапи Кохонена.

Тут $N_{jl}^{[m]K}$ — стандартні нейрони Кохонена, пов'язані між собою латеральними зв'язками, що налаштовуються згідно «переможцю дістається більше» правилу навчання на основі другого співвідношення (1.12). Вузли $N_{jl}^{[m]u}$ обчислюють рівні належності згідно першому співвідношенню (1.12). Вузли $N_j^{[m]}$ кожного з каскадів відрізняються тільки фаззіфікатором алгоритму самонавчання, а вузол кожного наступного каскаду містить додатково один нейрон Кохонена і один елемент для розрахунку рівнів належності.

1.5. Керування каскадами самонавчанної нейро-фаззі системи, що еволюціонує

Якість кластерування кожого вузла системи може бути оцінена за допомогою будь-якого з індексів, що використовується у задачах нечіткого кластерування [38]. Одим за найпростіших та разом з тим найефективніших індексів є так званий «коефіцієнт розбиття», який, власне, є середнім квадратів рівнів належності всіх спостережень до кожного кластеру і має вигляд

$$PC_j^{[m]} = \frac{1}{N}\sum_{k=1}^{N}\sum_{l=1}^{m+1}\left(u_{jl}^{[m]}(k)\right)^2. \tag{1.18}$$

Цей коефіцієнт має ясний фізичний зміст: щокраще виражені кластери, то більше значення $PC_j^{[m]}$ (верхня межа — $PC_j^{[m]} = 1$), а його мінімум $PC_j^{[m]} = (m+1)^{-1}$ досягається, якщо дані належать усім кластерам рівномірно, що, вочевидь, є тривіальним рішенням. Для розглянутої нами системи цей коефіцієнт зручний тим, що його легко розрахувати в online режимі

$$PC_j^{[m]}(k+1) = PC_j^{[m]}(k) + \frac{1}{k+1}\left(\sum_{l=1}^{m+1}\left(u_{jl}^{[m]}(k+1)\right)^2 - PC_j^{[m]}(k)\right). \tag{1.19}$$

Розрахунок коефіцієнту розбиття проводиться для кожного вузла системи разом з налаштуванням їх параметрів, тобто співвідношення (1.12) та (1.19) реалізуються одночасно. На кожному такті навчання вузол $PC^{*[m]}$ визначає найкращий елемент каскаду, що забезпечує максимальне значення коефіцієнта розбиття у кожний поточний момент k, при цьому не виключається ситуація, коли в різні моменти обробки інформації «переможцями» виявляться різні вузли.

Кожен з каскадів розглянутої системи відрізняється від інших числом кластерів, на які розбивається оброблюваний потік даних. Тому якщо вузли $PC_j^{[m]}$ і $PC^{*[m]}$ оцінюють якість кластеризації без урахування кількості

сформованих класів, то вузли системи, позначені $XB^{[m]}$ та XB^*, оцінюють результати з урахуванням числа кластерів у кожному каскаді. Одним з таких показників є індекс Ксі-Бені [37], який для фіксованої вибірки з N спостережень може бути записаний у вигляді

$$XB_j^{[m]} = \frac{\left(\sum\limits_{k=1}^{N}\sum\limits_{l=1}^{m+1}\left(u_{jl}^{[m]}(k)\right)^2\left\|x(k)-c_{jl}^{[m]}\right\|^2\right)\Big/N}{\min\limits_{l\neq q}\left\|c_{jl}^{[m]}-c_{jq}^{[m]}\right\|^2} = \frac{NXB_j^{[m]}}{DXB_j^{[m]}} \tag{1.20}$$

Вираз (1.20) також можна записати у рекурентній формі

$$XB_j^{[m]}(k+1) = \frac{NXB_j^{[m]}(k+1)}{DXB_j^{[m]}(k+1)} =$$

$$= \frac{NXB_j^{[m]}(k)+\frac{1}{k+1}\left(\sum\limits_{l=1}^{m+1}\left(u_{jl}^{[m]}(k+1)\right)^2\left\|x(k+1)-c_{jl}^{[m]}(k+1)\right\|^2-NXB_j^{[m]}(k)\right)}{\min\limits_{l\neq q}\left\|c_{jl}^{[m]}(k+1)-c_{jq}^{[m]}(k+1)\right\|^2}, \tag{1.21}$$

при цьому рекурентні вирази (1.19) та (1.21) реалізуються одночасно.

Індекс Ксі-Бені є по суті співвідношенням відхилення всередині кластерів $NXB_j^{[m]}$ до величини поділу кластерів $DXB_j^{[m]}$. Оптимальному числу кластерів у каскаді відповідає мінімальне значення (1.20) та (1.21). Тому процес нарощування каскадів у системі продовжується доки значення індексу не почне збільшуватися. Цей процес контролює вузол архітектури XB^*.

Варто зауважити, що оскільки вузли кожного каскаду відрізняються тільки значенням фаззіфікатору, ефективність роботи кожного каскаду доцільно оцінювати за допомогою розширеного індексу Ксі-Бені EXB [38].

$$EXB_j^{[m]} = \frac{\left(\sum\limits_{k=1}^{N}\sum\limits_{l=1}^{m+1}\left(u_{jl}^{[m]}(k)\right)^{\beta^{[m]}}\left\|x(k)-c_{jl}^{[m]}\right\|^2\right)\Big/N}{\min\limits_{l\neq q}\left\|c_{jl}^{[m]}-c_{jq}^{[m]}\right\|^2} = \frac{NEXB_j^{[m]}}{DEXB_j^{[m]}} \qquad (1.22)$$

або його рекурентої форми

$$XB_j^{[m]}(k+1) = \frac{NXB_j^{[m]}(k+1)}{DXB_j^{[m]}(k+1)} =$$
$$\frac{NXB_j^{[m]}(k)+\frac{1}{k+1}\left(\sum\limits_{l=1}^{m+1}\left(u_{jl}^{[m]}(k+1)\right)^{\beta^{[m]}}\left\|x(k+1)-c_{jl}^{[m]}(k+1)\right\|^2-NXB_j^{[m]}(k)\right)}{\min\limits_{l\neq q}\left\|c_{jl}^{[m]}(k+1)-c_{jq}^{[m]}(k+1)\right\|^2}, \qquad (1.23)$$

де $\beta^{[m]}$ — фаззіфікатор найкращого з вузлів m-го каскаду.

Таким чином, процес еволюції пропонованої системи зумовлений максимізуванням поточного значення показника якості кластерування потоку даних, що надходять на обробку в онлайн режимі.

Висновки до розділу 1

1. Розглянуто завдання нечіткого кластерування у режимі послідовного надходження даних до системи.
2. Запропоновано метод визначення локально оптимальної кількості кластерів і значення параметру фаззіфікації для послідовного кластерування потоків даних.
3. Запропоновано архітектуру і метод самонавчання каскадної нейро-фаззі системи, що еволюціонує, для послідовного кластерування потоків даних з автоматичним визначенням оптимальної кількості кластерів. Кожен вузол кожного каскаду системи вирішує завдання кла-

стерування незалежно від інших, що дозволяє організувати паралельну обробку інформації в каскадах, тобто підвищити швидкодію цього процесу. Система не містить жодних порогових параметрів, що задаються суб'єктивно, а процес оцінювання якості її функціонування визначається шляхом відшукання оптимального значення певного індексу дійсності розбиття даних на кластери (їх поточна оцінка також проводиться в режимі реального часу). Відмінною особливістю пропонованої системи є те, що вона самостійно визначає і поточне значення фаззіфікатору, і оптимальну кількість кластерів на кожному етапі обробляння даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Abonyi Janos, Feil Balazs. Cluster Analysis for Data Mining and System Identification. — Birkhäuser Basel, 2007. — ISBN: 3764379871.

[2] Adaptive Clustering for Multiple Evolving Streams / Bi-Ru Dai, Jen-Wei Huang, Mi-Yen Yeh, Ming-Syan Chen // IEEE Trans. Knowl. Data Eng. — 2006. — Vol. 18, no. 9. — P. 1166–1180. — URL: http://dx.doi.org/10.1109/TKDE.2006.137; http://doi.ieeecomputersociety.org/10.1109/TKDE.2006.137.

[3] Arrow Kenneth Joseph, Hurwicz Leonid, Uzawa Hirofumi. Studies in linear and non-linear programming. Stanford mathematical studies in the social sciences. — Stanford,Ca : Stanford university press, 1972. — ISBN: 0-8047-0562-3. — URL: http://opac.inria.fr/record=b1099343.

[4] Beringer Jürgen, Hüllermeier Eyke. Online clustering of parallel data streams // Data Knowl. Eng. — 2006. — Vol. 58, no. 2. — P. 180–204. — URL: http://dx.doi.org/10.1016/j.datak.2005.05.009.

[5] Beringer Jürgen, Hüllermeier Eyke. Online clustering of parallel data streams // Data Knowl. Eng. — 2006. — Vol. 58, no. 2. — P. 180–204. — URL: http://dx.doi.org/10.1016/j.datak.2005.05.009.

[6] Bezdek J.C. Pattern recognition with fuzzy objective function algorithms. — Kluwer Academic Publishers, 1981.

[7] Bodyanskiy Yevgeniy. Computational Intelligence Techniques for Data Analysis // Leipziger Informatik-Tage / Ed. by Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig. — Vol. 72 of LNI. — GI, 2005. — P. 15–36. — URL: http://subs.emis.de/LNI/Proceedings/Proceedings72/article3735.html.

[8] Bodyanskiy Ye., Gorshkov Ye., Kolodyazhniy V. New recursive learning algorithms for fuzzy Kohonen clustering network // Proc. 17th Int. Workshop on Nonlinear Dynamics of Electronic Systems. — 2009. — P. 58–61.

[9] Bodyanskiy Ye., Kolodyazhniy V., Stephan A. Recursive fuzzy clustering algorithms // Proc. 10th East West Fuzzy Colloquium. — 2002. — P. 276–283.

[10] Chen Y., Tu L. Density-based clustering for real-time stream data // Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. — 2007. — P. 133–142.

[11] Chung Fu Lai, Lee Tong. Fuzzy Competitive Learning // Neural Netw. — 1994. — Vol. 7, no. 3. — P. 539–551. — URL: http://dx.doi.org/10.1016/0893-6080(94)90111-2.

[12] Chung Fu-Lai, Lee Tong. Fuzzy competitive learning // Neural Networks. — 1994. — Vol. 7, no. 3. — P. 539–551. — URL: http://dx.doi.org/10.1016/0893-6080(94)90111-2.

[13] Crespo Fernando, Weber Richard. A methodology for dynamic data mining based on fuzzy clustering // Fuzzy Sets and Systems. — 2005. — Vol. 150, no. 2. — P. 267–284. — URL: http://dx.doi.org/10.1016/j.fss.2004.03.028.

[14] Cypkin Ja.Z., Kaplinskii A. I., Krasnenker A. S. Fundamentals of the theory of learning systems // Nauka. — 1970. — P. 252.

[15] Du K.L., Swamy M.N.S. Neural Networks and Statistical Learning. SpringerLink : Bücher. — Springer, 2013. — ISBN: 9781447155713. — URL: https://books.google.com.ua/books?id=wzK8BAAAQBAJ.

[16] Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition / Frank Höppner, Frank Klawonn, Rudolf Kruse, Thomas Runkler. — 1. Auflage edition. — John Wiley & Sons, 1999. — ISBN: 0471988642.

[17] Fuzzy models and algorithms for pattern recognition and image processing / James C. Bezdek, James Keller, Raghu Krisnapuram, Nikhil R. Pal. The Handbooks of fuzzy sets series. — New York : Springer, 2005. — ISBN: 0-387-24515-4. — URL: http://opac.inria.fr/record=b1102792.

[18] Gan Guojun, Ma Chaoqun, Wu Jianhong. Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Prob-

ability). — SIAM, 2007. — ISBN: 0898716233.

[19] Han H., Qiao J. A self-organizing fuzzy neural network based on a growing-and-pruning algorithm // Fuzzy Systems, IEEE Transactions on. — 2010. — Vol. 18, no. 6. — P. 1129–1143.

[20] Han J, Kamber M. Mining Stream, Time-Series and Sequence Data // Data Mining: Concepts and Techniques. — 2006. — P. 467–489.

[21] Kashyap R. L., Blaydon Colin C. Estimation of probability density and distribution functions // IEEE Transactions on Information Theory. — 1968. — Vol. 14, no. 4. — P. 549–556. — URL: http://dx.doi.org/10.1109/TIT.1968.1054184.

[22] Kohonen Teuvo. Self-organizing maps. — Springer Science & Business Media, 2001. — Vol. 30.

[23] Leng Gang, Prasad Girijesh, McGinnity T. Martin. An on-line algorithm for creating self-organizing fuzzy neural networks // Neural Networks. — 2004. — Vol. 17, no. 10. — P. 1477–1493. — URL: http://dx.doi.org/10.1016/j.neunet.2004.07.009.

[24] Liu Ying-Ho. Stream mining on univariate uncertain data // Appl. Intell. — 2013. — Vol. 39, no. 2. — P. 315–344. — URL: http://dx.doi.org/10.1007/s10489-012-0415-3.

[25] Oliveira Jose Valente de, Pedrycz Witold. Advances in Fuzzy Clustering and Its Applications. — New York, NY, USA : John Wiley & Sons, Inc., 2007. — ISBN: 0470027606.

[26] Pakhira Malay K, Bandyopadhyay Sanghamitra, Maulik Ujjwal. Validity index for crisp and fuzzy clusters // Pattern recognition. — 2004. — Vol. 37, no. 3. — P. 487–501.

[27] Park Dong C., Dagher Issam. Gradient Based Fuzzy c_means (GBFCM) Algorithm // IEEE International Conference on Neural Networks (ICNN'94). — Vol. III. — Orlando, FL : IEEE, 1994. — jun. — P. 1626–1631. — Intelligent Computing Research Lab, Department of ECE, FL.

[28] Pedrycz Witold, Rai Partab. Collaborative clustering with the use of Fuzzy C-Means and its quantification // Fuzzy Sets and Systems. — 2008. — Vol.

159, no. 18. — P. 2399–2427. — URL: http://dx.doi.org/10.1016/j.fss.2007.12.030.

[29] Roubens Marc. Fuzzy clustering algorithms and their cluster validity // European Journal of Operational Research. — 1982. — Vol. 10. — P. 294–301.

[30] Silva Bruno, Marques Nuno. Neural Network-based Framework for Data Stream Mining. // STAIRS. — 2012. — P. 294–305.

[31] Smoothly distributed fuzzy C-means: a new self-organizing map / R. D. Pascual Marqui, A. D. Pascual Montano, K. Kochi, J. M. Carazo // Pattern Recognition. — 2001. — Vol. 34, no. 12. — P. 2395–2402. — URL: http://www.sciencedirect.com/science/article/B6V14-43W07HY-B/2/8a783959537578469a5357887f06327b.

[32] Tsao E. C. K., Bezdek J. C., Pal N. R. Fuzzy Kohonen clustering networks // Pattern Recognition. — 1994. — Vol. 27, no. 5. — P. 757–764. — URL: http://www.sciencedirect.com/science/article/B6V14-48MPPXD-1YX/2/cf0e26b7498b3a4f4b7698975495f7f9.

[33] Vuorimaa Petri. Fuzzy Self–Organizing Map // Fuzzy Sets and Systems. — 1994. — Vol. 66. — P. 223–231.

[34] Vuorimaa P. Use of the Fuzzy Self-Organizing Map in pattern recognition // Proceedings of the Third IEEE Conference on Fuzzy Systems. IEEE World Congress on Computational Intelligence / Signal Process. Lab. , Tampere Univ. of Technol. , Finland. — Vol. 2. — New York, NY, USA : IEEE, 1994. — P. 798–801.

[35] Wang W, Zhang Y. On fuzzy cluster validity indices // Fuzzy Sets and Systems. — 2007. — oct. — Vol. 158, no. 19. — P. 2095–2117.

[36] Wu Yi. Network Big Data: A Literature Survey on Stream Data Mining. — 2014. — Vol. 9. — P. 2427–2434. — URL: http://ojs.academypublisher.com/index.php/jsw/article/view/12381.

[37] Xie Xuanli Lisa, Beni Gerardo. A Validity Measure for Fuzzy Clustering // IEEE Trans. Pattern Anal. Mach. Intell. — 1991. — Vol. 13, no. 8. — P. 841–847. — URL: http://dx.doi.org/10.1109/34.85677.

[38] Xu Rui, Wunsch Donald C. Clustering algorithms in biomedical research:

a review // Biomedical Engineering, IEEE Reviews in. — 2010. — Vol. 3. — P. 120–154.