

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Послідовне нечітке кластерування потоків даних
на основі ансамблевого підходу
(тема)

Виконав:
студент 2 курсу, групи СШМ-19-2
Романенко А. Р.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник к.т.н., доцент, Чала Л. Е.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)
ініціали)

В.О. Філатов
(прізвище,

2021 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)
Кафедра _____ Штучного інтелекту _____
(повна назва)
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)
Тип програми _____ освітньо-наукова _____
(освітньо-професійна або освітньо-наукова)
Освітня програма _____ Системи штучного інтелекту (СШІ) _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Романенку Андрію Романовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Послідовне нечітке кластерування потоків даних _____
_____ на основі ансамблевого підходу _____

затверджена наказом університету від 29 березня 2021 р. № 390 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 20 ____ р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет та відомих наукових _____
проектів, електронні документації, тестові набори даних _____

4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної області та постановка _____
задачі дослідження, нечітка кластеризація, адаптивна кластеризація, ансамблі систем _____
обчислювального інтелекту, труднощі та особливості відомих методів кластерування даних, _____
нечітке послідовне кластерування та критерії дійсності, Адаптивне навчання вузлів каскадної _____
нейро-фаззі системи, що еволюціонує, середовище реалізації, опис алгоритму, аналіз _____
отриманих результатів _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – Архітектура ансамблю систем обчислювального інтелекту, Рисунок 2 – Каскадна адаптивна нейронна мережа для послідовного опрацювання даних, Рисунок 3 – Штучно сгенеровані набори даних, Рисунок 4 – Кластерування набору даних з чітко вираженими класами (Crisp dataset), Рисунок 5 – Кластерування набору даних з нечіткими межами класів (Fuzzy dataset), Рисунок 6 – Розбиття даних з кластерами, що перетинаються (Extra fuzzy dataset), Рисунок 7 – Ієрархічне кластерування датасету «Іриси Фішера», Рисунок 8 – Розкластерований датасет «Іриси Фішера», Рисунок 9 – Точність кластерування датасету «Іриси Фішера» від кількості кластерів та параметру фаззифікації, Рисунок 10 – Датасет «Знання студентів про електричні машини постійного струму», Рисунок 11 – Розкластерований датасет «Знання студентів про електричні машини постійного струму»

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. каф. ШІ Чала Л.Е.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів	Примітка
1.	Отримання завдання на кваліфікаційне проектування	29.03.2021	виконано
2.	Аналіз завдання та пошук літератури за темою	30.03-05.04	виконано
3.	Опрацювання літератури та аналіз об'єкту	06.04-12.04	виконано
4.	Вибір програмних засобів для розробки системи	13.04-19.04	виконано
5.	Розробка програмного засобу	20.04-03.05	виконано
6.	Аналіз отриманих результатів	04.04-06.05	виконано
7.	Оформлювання пояснювальної записки	06.05-11.05	виконано
8.	Оформлення презентаційних матеріалів	12.05.2021	виконано
9.	Представлення на рецензування	13.05.2021	виконано
11.	Представлення кваліфікаційної роботи	20.05.2021	

Дата видачі завдання 29 березня 2021 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Чала Л. Е.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 86 с., 20 рис., 9 табл., 69 формул, 2 дод., 35 джерел.

АНАСАМБЛЕВА АРХІТЕКТУРА, АНАЛІЗ ПОТОКІВ ДАНИХ, ЕВОЛЮЦІОНУЮЧА СИСТЕМА, НЕЧІТКЕ КЛАСТЕРУВАННЯ, НЕЧІТКИЙ С-СЕРЕДНІХ.

Об'єктом дослідження є процес послідовного кластерування потоків даних, за умов часткового перетинання кластерів

Предметом дослідження є ансамблева архітектура, яка складається з нечітких систем, що еволюціонують, призначена для вирішення задач кластерування даних, що послідовно надходять на опрацювання.

Метою даного дослідження є розробка ансамблю систем обчислювального інтелекту для вирішення задачі нечіткого кластерування потоків даних в умовах невідомої кількості класів.

Методи дослідження. Теорія штучних нейронних мереж, що дозволила синтезувати нові еволюційні архітектури нейронних мереж, нечітка логіка, що дала можливість реалізувати нечітке висновування на основі розроблених архітектур, теорія оптимізації, що забезпечила розробку методів налаштування синаптичних вагових коефіцієнтів з підвищеною швидкістю і стійкістю до зашумлених даних.

На сьогоднішній день актуальною є задача розробки нових адаптивних методів навчання ансамблів систем обчислювального інтелекту, заснованих на різних принципах, що дозволяють налаштовувати їх параметри в online режимі, що дозволяє забезпечити оптимальну точність вихідного сигналу в умовах апріорної та поточної невизначеності.

РЕФЕРАТ

Пояснительная записка: 86 с., 20 рис., 9 табл, 69 формул, 2 прил., 35 источников.

АНАСАМБЛЕВАЯ АРХИТЕКТУРА, АНАЛИЗ ПОТОКОВ ДАННЫХ, ЭВОЛЮЦИОНИРУЮЩАЯ СИСТЕМА, НЕЧЕТКАЯ КЛАСТЕРИЗАЦИЯ, НЕЧЕТКИХ С-СРЕДНИХ.

Объектом исследования является процесс последовательного кластерування потоков данных, в условиях частичного пересечения кластеров

Предметом исследования является ансамблевая архитектура, состоящая из нечетких систем, эволюционирующих, предназначенная для решения задач кластерування данных, последовательно поступают на обработку.

Целью данного исследования является разработка ансамбля систем вычислительного интеллекта для решения задачи нечеткого кластерування потоков данных в условиях неизвестного количества классов.

Методы исследования. Теория искусственных нейронных сетей, которая позволила синтезировать новые эволюционные архитектуры нейронных сетей, нечеткая логика, дала возможность реализовать нечеткое висновування на основе разработанных архитектур, теория оптимизации, обеспечила разработку методов настройки синаптических весовых коэффициентов повышенным быстродействием и устойчивостью к зашумленным данным.

На сегодняшний день является актуальной задача разработки новых адаптивных методов обучения ансамблей систем вычислительного интеллекта, основанных на различных принципах, позволяющих настраивать их параметры в online режиме, что позволяет обеспечить оптимальную точность выходного сигнала в условиях априорной и текущей неопределенности.

ABSTRACT

Explanatory note: 86 p., 20 fig., 9 tabl., 69 formulas, 2 ann., 35 sources.

ENSEMBLE ARCHITECTURE, EVOLVING SYSTEM, STEAM DATA MINING, FUZZY CLUSTERING, FUZZY C-MEANS.

The object of research is the process of sequential clustering of data streams, under conditions of partial intersection of clusters

The subject of research is the ensemble architecture, which consists of fuzzy evolving systems, designed to solve the problem of clustering data that are consistently submitted for processing.

The purpose of research is to develop an ensemble of computational intelligence systems to solve the problem of fuzzy clustering of data streams in an unknown number of classes.

Research methods. The theory of artificial neural networks, which allowed to synthesize new evolutionary architectures of neural networks, fuzzy logic, which allowed to realize fuzzy inference based on developed architectures, optimization theory, which provided the development of methods for adjusting synaptic weights with increased speed and resistance to data.

Today, the scientific task of developing new adaptive teaching methods for ensembles of computational intelligence systems based on various principles that allow you to adjust their parameters online, which allows to ensure optimal accuracy of the output signal in a priori and current uncertainty.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів..	8
Вступ.....	9
1 Аналіз предметної галузі та постановка завдання дослідження.....	10
1.1 Завдання класифікації та кластеризації.....	10
1.2 Нечітка кластеризація.....	14
1.3 Адаптивна кластеризація.....	18
1.4 Метрики кластерного аналізу.....	20
1.5 Задача кластеризації векторних даних.....	20
1.6 Постановка задачі дослідження.....	21
2 Ансамблі систем обчислювального інтелекту.....	24
3 Каскадна нейронна мережа, що еволюціонує, для послідовного нечіткого кластерування потоків даних.....	37
3.1 Труднощі та особливості відомих методів кластерування даних..	37
3.2. Нечітке послідовне кластерування та критерії дійсності.....	40
3.3. Архітектура каскадної мережі, що еволюціонує, для нечіткого кластерування.....	43
3.4. Адаптивне навчання вузлів каскадної нейро-фаззі системи, що еволюціонує.....	46
3.5. Керування каскадами самонавчання нейро-фаззі системи, що еволюціонує.....	50
4 Імітаційне моделювання самонавчання нейро-фаззі системи на основі ансамблевого підходу.....	54
Висновки.....	75
Перелік джерел посилання.....	76
Додаток А Вихідний код програми.....	78
Додаток Б Відомість кваліфікаційної роботи.....	86

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

ШНМ – штучна нейрона мережа;

FCM – fuzzy c-means – нечітких c-середніх;

PCM – possibilistic c-means – можливістьний алгоритм c-середніх;

PFCM – possibilistic fuzzy c-means – возможностью-нечіткий
алгоритм c-середніх.

ВСТУП

З різким розвитком інформаційних технологій в другій половині ХХ століття в світі інформації стало значно збільшуватися кількість даних. З'явилася потреба зберігати і обробляти великі обсяги інформації. Особливо значимою ця проблема стала на початку ХХІ століття в зв'язку з різким зростанням кількості користувачів мережі Інтернет і на даний момент вона залишається не менше актуальною. Змагальне машинне навчання є новим полем для досліджень, що виникло на перетині машинного навчання та кібербезпеки. Причиною для цього стала виявлена вразливість інтелектуальних моделей до ледве помітних збурень, які однак здатні вплинути на значення вихідного сигналу. Однією з відкритих задач даної області є розробка та дослідження змагальних атак, що дозволять оцінити вразливості інтелектуальних моделей.

Під час інтелектуального аналізу і обробки даних вирішується безліч завдань, в тому числі і завдання кластерного аналізу, метою якого є виділення з даних деяких структур або груп подібних об'єктів, що становлять кластери, при цьому відсіваючи помилкові, нерелевантні структури. Однак часто об'єкти не можна чітко віднести до того чи іншого кластеру, тому інтенсивно розвивається нечітка (фаззи) кластеризація, яка розбиває вихідну вибірку на перекриваються класи. Кластерний аналіз є важливим завданням, оскільки оцінка результатів кластеризації не може бути об'єктивною, і різні методи дають різні результати для одних і тих же даних. Також, крім великої кількості існуючих проблем з'являються нові, що вимагають нових методів рішень або удосконалення старих. Наприклад, до таких проблем можуть ставитися уявлення даних в матричній, а не векторній формі або велика розмірність вектора даних, а також необхідність обробляти дані в режимі реального часу.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ

Одне з найбільш поширених завдань для нейронних мереж та машинного навчання – завдання класифікації. На сьогодні розроблено велике число підходів до рішення завдань класифікації зі вчителем та в режимі самонавчання. Для вирішення завдання класифікації зі вчителем потрібно створити класифікуючу функцію, яка привласнює набору вхідних екземплярів значення мітки одного з класів. Цей метод вимагає навчання. А якщо розглядати задачу класифікації без вчителя, тобто задачу кластеризації тут досить часто виникає проблема коли немає інформації стосовно кількості кластерів, а дані перетинаються і не можуть бути лінійно роздільними.

1.1 Завдання класифікації та кластеризації

Достатньо багато досліджень ставлять за мету організацію отриманих даних у наглядні структури. Так, в біології часто метою є розбиття сукупності тварин на види і підвиди, у психології – класифікація видів поведінки, у педагогіці – таксономія виховних цілей тощо. Допомогти це зробити може кластерний аналіз. Фактично, кластерний аналіз є набором різноманітних алгоритмів класифікації. Відомі також ряд прикладів застосування кластерного аналізу в сфері маркетингу. Загалом, коли необхідно класифікувати великі масиви інформації на групи, які придатні для подальшого аналізу – кластерний аналіз є незамінним інструментом. Цікаві можливості кластерного аналізу у сфері психологічних досліджень групових процесів та явищ.

Класифікація являє собою одну з найбільш часто виконуваних задач. При її вирішенні дані, одержувані в серії експериментів, використовуються для поділу досліджуваних зразків на кілька класів.

Класифікація – завдання упорядкування по деякому принципом множини об'єктів, які мають подібні класифікаційні ознаки (одна або кілька властивостей), обраних для визначення подібності або відмінності між цими об'єктами [1]. Об'єкти, що утворюють один клас, є в певному сенсі «схожими» один на одного, а об'єкти, що належать різним класам – «відмінними» один від одного. Розрізняють:

– допоміжну (штучну) класифікацію, яка проводиться по зовнішній ознаці і служить для додання множини спостережень (процесів, явищ) потрібного порядку;

– природну класифікацію, яка проводиться за істотними ознаками, що характеризує внутрішню спільність предметів і явищ. Вона є результатом і важливим засобом наукового дослідження, так як передбачає і закріплює результати вивчення закономірностей класифікуються об'єктів.

Під класифікацією розуміється віднесення об'єктів (спостережень, подій) до одного з заздалегідь відомих класів.

Класифікація відноситься до стратегії навчання зі вчителем (supervised learning), яке також називається контрольованим або керованим навчанням.

Завданням класифікації часто називають передбачення категоріальної залежної змінної (тобто залежною змінною, яка є категорією) на основі вибірки безперервних і / або категоріальних змінних.

Класифікація може бути одновимірної (за однією ознакою) і багатовимірної (за двома і більше ознаками).

Кластеризація – це задача класифікації без вчителяб тобто це самонавчання система, яка має сама прийняти рішення до якого класу (кластеру) віднести те чи інше спостереження. Задача кластеризації є одною з головних задач аналізу даних та машинного навчання. В рамках вирішення цієї задачі треба встановити міру подібності, щоб віднести спостереження на до певного кластеру. В якості такої міри подібності в

кластерном аналізі прийнято використовувати відстань між об'єктами та центроїдами.

У випадку, представленому на рисунку 1.1, мірою подібності є відстань: два або більше об'єктів належать до одного кластеру, якщо вони «близькі» відповідно до заданого відстанню. Це кластеризація на підставі відстані.

Інший тип кластеризації – концептуальна кластеризація: два або більше об'єктів належать до одного кластеру, якщо вони визначають поняття, загальне для всіх об'єктів. Іншими словами, об'єкти групуються так, щоб відповідати певному концепту.

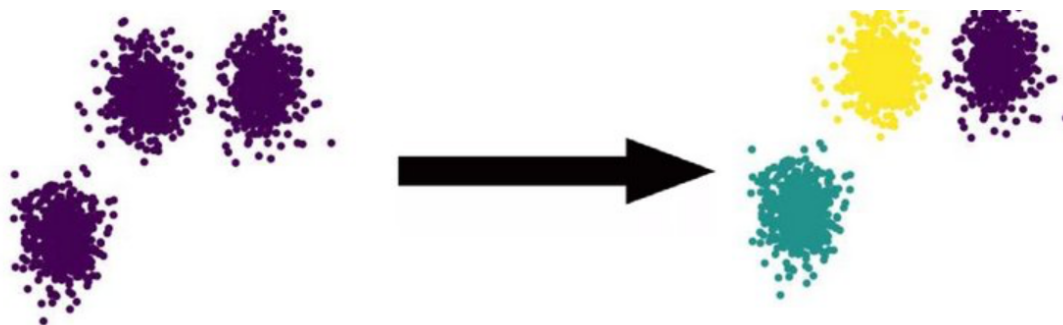


Рисунок 1.1 – Кластеризація

Основна мета кластеризації полягає в розбитті заданої вибірки об'єктів (ситуацій) або ж потоку даних на апіорі відоме (довільне) число підмножин опуклої, лінійно роздільної (довільної) форми, які називаються кластерами, так, щоб кожен кластер складався зі схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися.

Наразі не існує універсального методу для вирішення цієї задачі, є певні початкові умови, такі як форма кластерів, кількість спостережень, їх розмірність та багато інших додаткових умов які потребують використання специфічного апарату для вирішення конкретної задачі.

Кластеризація даних має одну суттєву особливість – він не є звичайним статистичним методом, оскільки до нього у більшості випадків незастосовні процеси перевірки статистичної значимості. Кластеризація дає найбільш можливо-значиме рішення. Саме тому досить часто його використовують тоді, коли дослідник має набір даних, але не має жодної апріорної гіпотези про класи цих даних.

Виявлення або знаходження кластерів в множині даних досліджуваної сукупності повинно відповідати таким вимогам:

- кожен кластер повинен являти собою концептуально однорідну категорію і містити схожі об'єкти з близькими значеннями властивостей або ознак;

- сукупність всіх кластерів повинна бути вичерпною, тобто охоплювати всі об'єкти досліджуваної сукупності;

- кластери повинні бути взаємно виключають, тобто жоден з об'єктів досліджуваної сукупності не повинен одночасно належати двом різним кластерам.

Формально під завданням кластерного аналізу заданої множини об'єктів розуміється завдання знаходження деякого розбиття цієї множини об'єктів на підмножини таким чином, щоб елементи, що відносяться до одній підмножині, відрізнялися між собою в значно меншій мірі, ніж елементи з різних підмножин.

У таблиці 1.1 наведено порівняння деяких параметрів задачі класифікації та кластеризації.

Виходячи з наведених характеристик, класифікація відносить кожен об'єкт до однієї з заздалегідь визначених груп, а кластеризація призначена для розбиття множини об'єктів на групи. Задача кластерного аналізу подібна до завданням класифікації, є її логічним продовженням, але її відмінність в тому, що класи досліджуваного набору даних заздалегідь не визначені або можливо буде відомо ймовірнісну їх кількість. Тому ці дві задачі часто використовуються разом, спочатку модель навчається на

даних з вірними мітками, а потім натренована модель на основі алгоритмів самонавчання починає кластеризацію вже на даних без міток належності до кластерів. Такий підхід до навчання має назву напівконтрольоване навчання.

Таблиця 1.1 – Порівняння задач класифікації та кластеризації

Характеристика	Класифікація	Кластеризація
Парадигма навчання	Контрольоване навчання	Самонавчання
Стратегія	Навчання зі вчителем	Навчання без вчителя
Наявність мітки класу	Навчальна множина супроводжується міткою, яка вказує клас, до якого належить спостереження	Мітки класу навчальної множини невідомі
Підстава для класифікації	Нові дані класифікуються на підставі навчальної множини	Дано безліч даних з метою встановлення існування класів або кластерів даних

1.2 Нечітка кластеризація

Теорія нечітких множин, основні ідеї якої були запропоновані американським математиком Лотфі Заде в 1965 році, дозволяє описувати якісні, неточні поняття і наші знання про навколишній світ, а також оперувати цими знаннями з метою отримання нової інформації [2]. Засновані на цій теорії методи побудови інформаційних моделей істотно розширюють традиційні сфери застосування комп'ютерних систем і утворюють самостійний напрям науково-прикладних досліджень, яке отримало спеціальну назву – нечітке моделювання. Воно виявляється особливо ефективним, коли в описі технічних систем і процесів присутня

невизначеність, яка ускладнює або навіть виключає застосування точних кількісних методів.

В області аналізу даних нечітке моделювання часто дозволяє отримувати більш адекватні результати в порівнянні з результатами, які ґрунтуються на використанні традиційних аналітичних моделей і алгоритмів.

Нечітке безліч (fuzzy set) являє собою сукупність елементів довільної природи, щодо яких не можна з повною впевненістю стверджувати – чи належить той чи інший елемент розглянутої сукупності даній безлічі чи ні.

Взаємозв'язок між кластерним аналізом і теорією нечітких множин заснована на тій обставині, що при вирішенні завдань структуризації складних систем більшість об'єктів виявляються розмитими за своєю природою. Ця розмитість полягає в тому, що перехід від належності до неналежності елементів до даних класах швидше поступовий, ніж стрибковий.

Вимога знаходження однозначної кластеризації елементів досліджуваної проблемної області є досить грубим і жорстким, особливо при вирішенні погано або слабо структурованих задач системного аналізу. Методи нечіткої кластеризації послаблюють цю вимогу. Ослаблення вимоги здійснюється за рахунок введення в розгляд нечітких кластерів і відповідних їм функцій належності, що приймають значення з інтервалу $[0, 1]$. На рисунку 1.2 зображена нечітка кластеризація з різними ступенями належності спостережень.

Для будь-якої міри схожості величина належності спостереження кластеру залежить від схожості об'єкта і прототипу цього кластера. У разі якщо мірою подібності є відстань, то величина належності об'єкта обернено пропорційна його відстані до центроїда кластера. Сума всіх функцій належності спостереження кластерам в будь-який момент часу повинна дорівнювати 1.

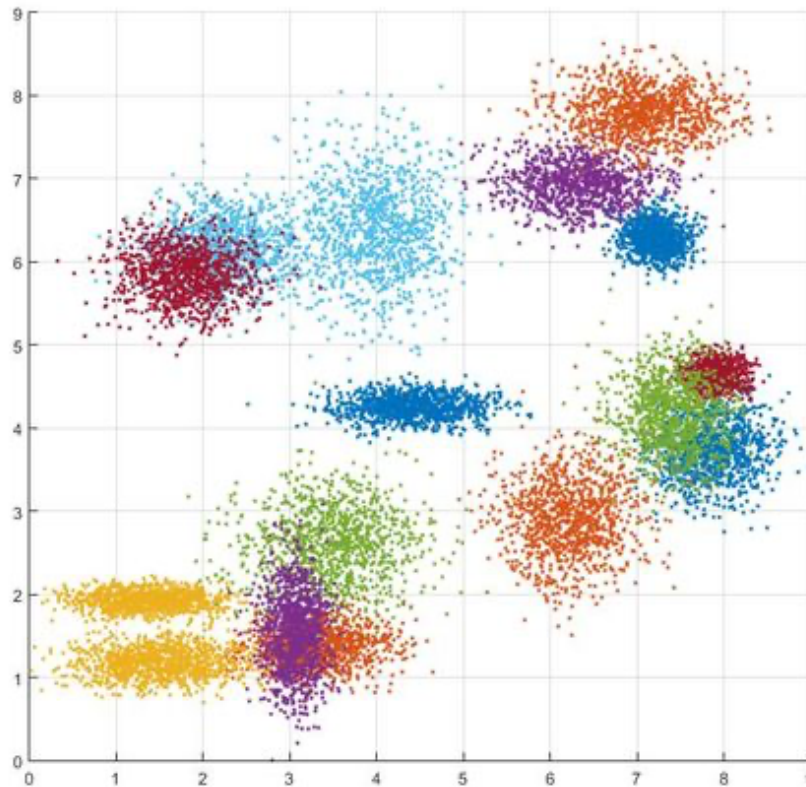


Рисунок 1.2 – Нечітка кластеризація

Таким чином, в загальному випадку завданням нечіткої кластеризації полягає в знаходженні нечіткого розбиття або нечіткого покриття множини елементів досліджуваної сукупності, які утворюють структуру нечітких кластерів, присутніх в розглянутих даних. Це завдання зводиться до знаходження ступенів належності елементів шуканим нечітким кластерам, які в сукупності і визначають нечітке розбиття або нечітке покриття вихідної множини розглянутих елементів.

Основні ідеї алгоритму для вирішення завдання нечіткої кластеризації були запропоновані Дж. К. Даному в 1973 році [3]. Надалі алгоритм був доопрацьований Джеймсом Бездеком і отримав назву нечітких C -середніх (FCM) [4].

Поряд з традиційним імовірнісним підходом до нечіткої кластеризації, коли кожен об'єкт з певною ймовірністю належить до кожного з кластерів, існує можливістьний підхід до кластерного аналізу.

Можливісна кластеризація також розглядає нечіткі кластери і відповідні їм функції належності, що приймають значення з інтервалу $[0, 1]$. Різниця полягає в тому, що імовірнісна кластеризація має на увазі наявність суворого обмеження, що сума всіх функцій належності об'єкта до всіх кластерів дорівнює 1, а можливісний кластерний аналіз не має на увазі подібного обмеження.

Перевага можливісного кластерного аналізу над імовірнісним полягає в тому, що об'єкти, які мають низький рівень подібності з будь-яким з кластерів (outliers), будуть мати значення належності близьке нулю для всіх кластерів, в той час як імовірнісний нечіткий кластерний аналіз буде явно віддавати перевагу одному або декільком кластерам (хоча всі вони повинні бути досить погані).

Часто результати можливісної кластеризації перетворюють в імовірнісний вигляд за допомогою нормалізації, попередньо позбувшись від outlier-об'єктів. Зазвичай, результати отримані таким чином, краще результатів, отриманих при прямому використанні імовірнісної кластеризації (проте, це також залежить від вихідних даних, від конкретних алгоритмів нечіткої кластеризації і ініціалізації їх параметрів).

Алгоритм FCM має ітеративний характер послідовного поліпшення деякого нечіткого розбиття, яке задається користувачем або формується автоматично за деяким евристичним правилом. На кожній з ітерацій рекуррентно перераховуються значення функцій належності об'єктів нечітким кластерам і їх типові представники (центроїди).

Алгоритм закінчить роботу в разі, коли станеться виконання заданого апріорі деякого кінцевого числа ітерацій, або коли мінімальна абсолютна різниця між значеннями функцій належності (або центроїдами кластерів) на двох фінальних ітераціях не стане менше деякого апріорі заданого значення.

Формально алгоритм FCM визначається в формі ітеративного виконання наступній послідовності кроків:

а) задається кількість шуканих нечітких кластерів c , максимальна кількість ітерацій алгоритму s , параметр збіжності алгоритму ε . Як поточного нечіткого розбиття на першій ітерації алгоритму для матриці даних D задається деякий вихідне нечітке розбиття на c непустих нечітких кластерів, які описуються сукупністю функцій належності;

б) для вихідного поточного нечіткого розбиття розраховуються центри нечітких кластерів і значення цільової функції. Кількість виконаних ітерацій встановлюється в 1;

в) формується нове нечітке розбиття вихідного безлічі об'єктів кластеризації на c непустих нечітких кластерів, що характеризуються сукупністю функцій належності;

г) для нового нечіткого розбиття розраховуються центри нечітких кластерів і значення цільової функції;

д) якщо кількість виконаних ітерацій перевищує вказану кількість s або ж значення модуля різниці між значеннями функцій належності (або центроїдами кластерів) на двох останніх ітераціях менше значення параметра збіжності алгоритму ε , то в якості бажаного результату нечіткої кластеризації приймається останнім нечітке розбиття і виконання алгоритму припиняється. В іншому випадку відбувається перехід на крок (в) алгоритму, збільшуючи на 1 кількість виконаних ітерацій.

Алгоритм FCM за своїм характером відноситься до наближених алгоритмам пошуку екстремуму для цільової функції при наявності обмежень. В результаті виконання даного алгоритму визначається локально-оптимальне нечітке розбиття, яке описується сукупністю функцій належності.

Можливістний алгоритм з-середніх (Possibilistic c-means, PCM) був запропонований Р. Крішнапурамом і Дж. Келлером в 1993 році [5] і доопрацьований ними в 1996 році [6]. Робота алгоритму PCM не відрізняється від FCM, а різниця полягає лише в тому, що імовірнісна матриця нечіткого розбиття замінюється на можливістну i , відповідно,

змінюється процедура обчислення даної матриці. У 2005 році Н. Р. Пал і інші запропонували поєднати FCM і PCM і отримали дві модифікації: нечітко-можливісного алгоритм с-середніх (FPCM) і возможностью-нечіткий алгоритм с-середніх (PFPCM) [11].

1.3 Адаптивна кластеризація

Адаптивність – здатність системи змінювати свої властивості відповідно до змін параметрів середовища і вхідних даних. Подібні системи адаптують значення своїх параметрів виходячи з інформації, отриманої із зовнішнього середовища.

Адаптивність алгоритму кластеризації має на увазі, що даний алгоритм послідовно обробляє спостереження, що надходять для кластерного аналізу, при цьому зміщуючи центри кластерів ближче до нового об'єкту на певну величину.

Кожне вхідне спостереження опрацьовується один раз, на відміну від алгоритмів, що працюють ітераційно з пакетом даних. Центроїд найближчого кластеру зміщується в бік нового об'єкту на відстань, яке прямо пропорційно залежить від параметра кроку пошуку $\eta(k)$ та відстані між даним об'єктом і прототипом кластера.

В нечітких адаптивних алгоритмах зміщення центроїда кожного кластера також прямо пропорційно значенню належності спостереження даного класу. Параметр кроку пошуку в більшості випадків зменшується зі збільшенням кількості вхідних спостережень, тому адаптивні алгоритми частіше використовуються для невеликих вибірок даних, так як кожне нове спостереження все менше впливає на зміну значень центроїдів.

У випадках, коли необхідно обробляти великі обсяги даних в реальному часі, параметр кроку пошуку перестає зменшуватися і стає фіксованим починаючи з обробки n -ого об'єкта.

Існують адаптивні модифікації методів К-середніх і нечітких С-середніх. Адаптивний аналіз даних зупиняється тоді, коли нові об'єкти перестають надходити на обробку.

Адаптивні алгоритми можна поєднувати з пакетними, використовуючи результати кластерного аналізу пакетного алгоритму для подальшої адаптації центроїдів кластерів спостереженнями, які надходять на обробку в режимі реального часу.

Результати адаптивного кластерного аналізу можуть істотно відрізнятися від результатів пакетної кластеризації. У більшості випадків адаптивні алгоритми працюють швидше завдяки відсутності повторної обробки тих же самих об'єктів.

1.4 Метрики кластерного аналізу

Подібність або відмінність між об'єктами кластеризації встановлюється в залежності від обраного метричного відстані між ними. Якщо об'єкт описується d властивостями (ознаками), то він може бути представлений як точка в d -мірному просторі, і схожість з іншими спостереженнями буде визначатися як відповідне відстань.

Мірою подібності (схожості) називається будь-яка функція, яка може визначити значення подоби або відстань між двома об'єктами в даному просторі. Подібність між двома об'єктами назад пропорційно відстані між ними.

Метрика – функція, що визначає відстань в метричному просторі. В даному просторі визначено відстань між будь-якою парою елементів. Міри схожості для кількісних і якісних (номінальних) даних можуть істотно відрізнятися один від одного.

Міра подібності називається метрикою, якщо задовольняє наступним умовам:

– аксіома тотожності – відстань між двома тотожними об'єктами дорівнює нулю ($d(x, y) = 0 \Leftrightarrow x = y$);

– аксіома симетрії – відстань між об'єктами x та y дорівнює відстані між об'єктами x та y ($d(x, y) = d(y, x)$);

– нерівність трикутника – відстань між об'єктами x та z не перевищує суми відстаней між об'єктами x та y і об'єктами y та z ($d(x, z) \leq d(x, y) + d(y, z)$).

1.5 Задача кластеризації векторних даних

Проблема кластеризації багатовимірних спостережень досить часто зустрічається в багатьох задачах, пов'язаних з інтелектуальним аналізом даних. Традиційний підхід до вирішення цих завдань допускає, що кожне спостереження може відноситися тільки до одного кластеру [7], хоча більш природною є ситуація, коли вектор ознак, який надходить на обробку, з різними рівнями належності або ймовірності, або можливостей може належати одночасно до окремих класів. Така ситуація є предметом розгляду нечіткого (фаззи) кластерного аналізу, який в даний час досить інтенсивно розвивається [8].

Традиційно вихідною інформацією для завдання кластеризації є вибірка спостережень, сформована з N n -мірних векторів ознак $X = \{x(1), x(2), \dots, x(k), \dots, x(N)\}$, $x(k) = (x_1(k), \dots, x_n(k))^T \in R^n$, $k = 1, 2, \dots, N$, а результатом роботи алгоритму є поділ вихідного масиву даних на m класів з деяким рівнем $u_j(k)$ належності k -ого вектора ознак до j -го кластеру.

У той же час існує широкий клас задач, коли вихідна інформація надходить не в векторній, а в матричній формі, тобто $x(k) = \{x_{i_1 i_2}(k)\}$; $i_1 = 1, 2, \dots, n_1$; $i_2 = 1, 2, \dots, n_2$; $k = 1, 2, \dots, N$.

Така ситуація є характерною, наприклад, при обробці зображень [20], коли вихідна $(N_1 \times N_2)$ -матриця розбивається на $N = N_1 \cdot N_2 \cdot (n_1 \cdot n_2)^{-1}$ $(n_1 \times n_2)$ -матриць-фрагментів, які підлягають кластеризації, в результаті якої формуються однорідні в деякому сенсі сегменти цього зображення.

Традиційно це завдання вирішується шляхом попередньої векторизації фрагментів і використання вже відомих процедур, найбільш популярною з яких є метод кластеризації нечітких C-середніх [8].

1.6 Постановка задачі дослідження

Як відомо, на сьогоднішній день розроблено безліч алгоритмів і методів кластеризації, при цьому більшість з них є по своїй суті чіткими процедурами, де передбачається, що кластери лінійно-роздільні і мають опуклу форму. У практичних задачах досить часто виникає ситуація, коли кластери можуть перекриватися і мати довільну форму, в таких випадках можуть бути використані алгоритми нечіткої кластеризації в їх рекурентній формі. Разом з тим, досить часто виникає задача, коли кількість кластерів невідома апіорі. Дана проблема може бути вирішена на основі ансамблевого підходу [9], [10].

Можливості сучасної обчислювальної техніки дозволяють обробляти великі масиви інформації в послідовному режимі в умовах апіорної та поточної невизначеності, коли відсутня інформація про функціональні залежності всередині даних, про зміну цих залежностей і про обсяг всієї вибірки. При роботі в режимі послідовної обробки інформації гостро постає проблема підвищення швидкості обробки даних з мінімальною похибкою. Для розв'язання задач інтелектуального аналізу даних є доцільним використання методів наукового напрямку, що розвивається в межах штучного інтелекту й отримав назву «обчислювальний інтелект».

Переваги цих методів полягають у тому, що процес обробки інформації за їх допомогою відбувається подібно до процесів, які відбуваються у біологічних системах.

Таким чином, на сьогоднішній день актуальною є наукова задача розробки нових адаптивних методів навчання ансамблів систем обчислювального інтелекту, заснованих на різних принципах, що дозволяють налаштовувати їх параметри в on-line режимі, що дозволяє забезпечити оптимальну точність вихідного сигналу в умовах апріорної та поточної невизначеності.

Об'єктом дослідження є процес послідовного кластерування потоків даних, за умов часткового перетинання кластерів

Предметом дослідження є ансамблева архітектура, яка складається з нечітких систем, що еволюціонують, призначена для вирішення задач кластерування даних, що послідовно надходять на опрацювання.

Методи дослідження. Теорія штучних нейронних мереж, що дозволила синтезувати нові еволюційні архітектури нейронних мереж, нечітка логіка, що дала можливість реалізувати нечітке висновування на основі розроблених архітектур, теорія оптимізації, що забезпечила розробку методів налаштування синаптичних вагових коефіцієнтів з підвищеною швидкістю і стійкістю до зашумлених даних.

Метою даного дослідження є розробка ансамблю систем обчислювального інтелекту для вирішення задачі нечіткого кластерування потоків даних в умовах невідомої кількості класів. Відповідно до поставленої мети необхідно розв'язати такі задачі:

- аналіз існуючих методів та підходів до кластерування та класифікації даних;
- аналіз існуючих методів нечіткого кластерування даних;
- розробка ансамблю систем обчислювального інтелекту для вирішення задачі нечіткого кластерування потоків даних в умовах невідомої кількості класів.

Припускається, що процес класифікації та кластеризації за допомогою ансамблів вирішує задачу визначення оптимальної кількості класів у випадку коли дані на опрацювання надходять в потоковому режимі та суттєво перетинаються, а їх кількість невідома заздалегідь. Треба відмітити актуальність пропонованого в атестаційній роботі підходу до нечіткого кластервання поточкових даних та простоту реалізації ансамблевої системи.

2 АНСАМБЛІ СИСТЕМ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ

В попередньому розділі було розглянуто основні підходи до вирішення задач класифікації та кластеризації. На даний час немає одного універсального методу або підходу до вирішення цих задач. Все дуже відносно і треба дивитись на умови в яких будуть опрацьовуватись масиви даних, на природу даних. Тому на сьогодні немає універсального методу і для кожної окремої задачі треба використовувати той чи інший підхід.

Якість рішення певної задачі кластеризації або класифікації можна істотно підвищити за допомогою ансамблів (комітетів, банків) методів, алгоритмів та нейронних мереж [11], в яких одні і ті ж дані паралельно обробляються декількома системами, вихідні сигнали яких далі деяким чином комбінуються в об'єднану оцінку, що перевершує за якістю результати, одержані за допомогою локальних систем, що входять в ансамблі так, як це показано на рисунку 2.1.

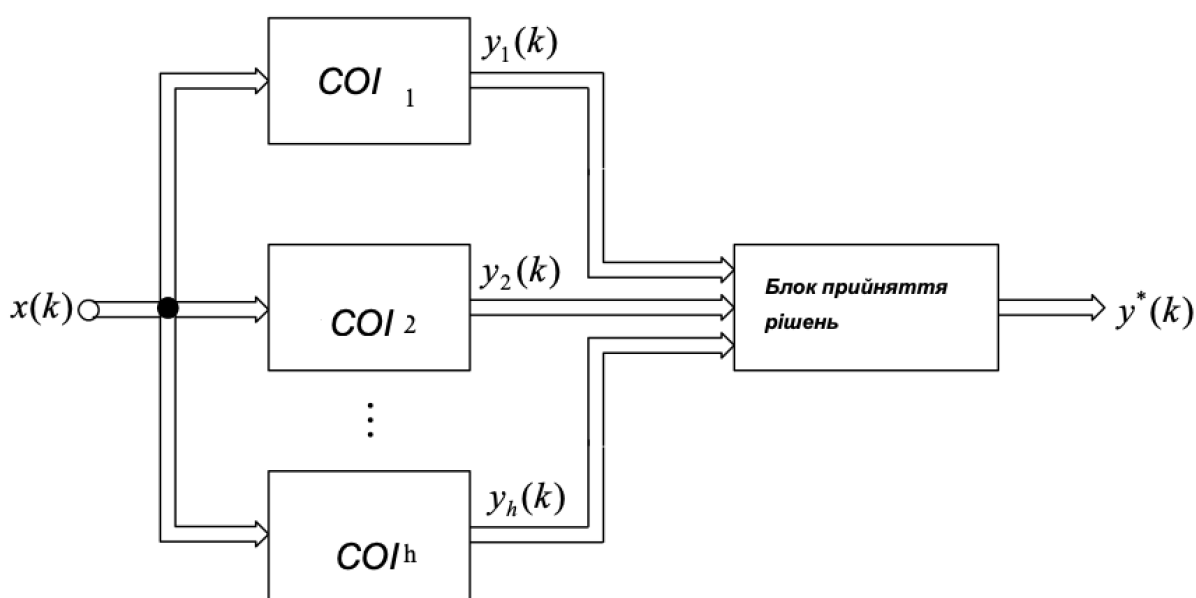


Рисунок 2.1 – Архітектура ансамблю систем обчислювального інтелекту

На практиці найбільшого поширення набули два підходи до об'єднання систем в ансамблі: модульний [4] і заснований на зваженому усередненні і хоча змістовно вони досить відрізняються один від одного, їх об'єднує те, що обидва вони використовують лінійну комбінацію вихідних сигналів своїх членів в тій чи іншій формі [12]. Модульний підхід має досить евристичний характер на відміну від більш математично строгого зваженого усереднення, хоча і тут залишається елемент суб'єктивізму, пов'язаний з вибором членів ансамблю. Це завдання зазвичай вирішується за допомогою тих чи інших евристик, хоча і є більш-менш суворі результати, засновані на генетичному програмуванні [13] або поступове нарощування складності систем-членів ансамблю [4].

У цьому розділі магістерської роботи буде розглянуто адаптивний підхід до об'єднання систем-членів ансамблю [13], призначений для роботи в реальному часі в темпі надходження нових даних, який дозволяє автоматично виявляти «найкращі» локальні блоки, найбільшим чином пристосовані для вирішення завдання послідовної кластеризації, за умов класів, що перетинаються.

Нехай сигнал, який треба опрацювати, задано у вигляді n -мірної часової послідовності $x(k)$, $k = 1, 2, \dots$, а вихідний сигнал l -того блоку ансамбля це m -мірна послідовність $y_l(k)$, $l = 1, 2, \dots, h$. Ведемо у розгляд узагальнюючу оцінку

$$y^*(k) = \sum_{l=1}^h w_l y_l(k) = y(k)w, \quad (2.1)$$

де $w = (w_1, w_2, \dots, w_h)^T$ – вектор невідомих вагових коефіцієнтів, який визначає близькість сигналів $y_l(k)$ до навчального сигналу $d(k)$ і відповідають умові незміщеності [13]

$$E_h^T w = 1, \quad (2.2)$$

де $y(k) = (y_1(k), y_2(k), \dots, y_h(k)) - (m \times h)$ – матриця;

$E_h - (h \times 1)$ – вектор, який складається з одиниць.

Вектор вагових коефіцієнтів w може бути знайдено за допомогою методу невизначених множників Лагранжа, для чого слід ввести $(k \times m)$ – матрицю навчальних сигналів, $(k \times mh)$ – матрицю вихідних сигналів ансамблю:

$$D(k) = \begin{pmatrix} d^T(1) \\ d^T(2) \\ \vdots \\ d^T(k) \end{pmatrix}, \quad Y(k) = \begin{pmatrix} y_1^T(1) & y_2^T(1) & \cdots & y_h^T(1) \\ y_1^T(2) & y_2^T(2) & \cdots & y_h^T(2) \\ \vdots & \vdots & \ddots & \vdots \\ y_1^T(k) & y_2^T(k) & \cdots & y_h^T(k) \end{pmatrix}, \quad (2.3)$$

$(k \times m)$ – матриця похибок

$$e^k = D(k) - Y(k)I_m \otimes w \quad (2.4)$$

та функцію Лагранжа

$$\begin{aligned} L(w, \lambda) &= \frac{1}{2} \text{Tr}(e^{kT} e^k) + \lambda(w^T E_h - 1) = \\ &= \frac{1}{2} \text{Tr}(D(k) - Y(k)I_m \otimes w)^T (D(k) - Y(k)I_m \otimes w) + \lambda(w^T E_h - 1) = \\ &= \frac{1}{2} \sum_{p=1}^k \|d(p) - y(p)w\|^2 + \lambda(w^T E_h - 1). \end{aligned} \quad (2.5)$$

$I_m - (m \times m)$ – одинична матриця;

\otimes – символ тензорного твору;

λ – невизначений множник Лагранжу.

Рішення системи рівнянь Куна-Таккера

$$\begin{cases} \nabla_w L(w, \lambda) = \sum_{p=1}^k (-y^T(p)d(p) + y^T(p)y(p)w) + \lambda E_h = 0, \\ \partial L(w, \lambda)/\partial \lambda = w^T E_h - 1 = 0 \end{cases} \quad (2.6)$$

дозволяє знайти шуканий вектор ваг у вигляді

$$w = w^* + P(k) \frac{1 - E_h^T w^*}{E_h^T P(k) E_h} E_h, \quad (2.7)$$

де

$$\begin{cases} P(k) = \left(\sum_{p=1}^k y^T(p)y(p) \right)^{-1}, \\ w^* = P(k) \sum_{p=1}^k y^T(p)d(p) = P(k)r(k) \end{cases} \quad (2.8)$$

– оцінка стандартного методу найменший квадратів.

З тим, щоб оцінити властивості отриманого вектору ваг (2.7), перепишемо цей вираз в дещо іншій формі. З урахуванням того, що послідовність помилок може бути представлена у вигляді

$$e(k) = d(k) - y(k)w = d(k)E_h^T w - y(k)w = (d(k)E_h^T - y(k))w = v(k)w, \quad (2.9)$$

лагранжіан (2.5) можна переписати у вигляді

$$L(w, \lambda) = \frac{1}{2} \sum_{p=1}^k w^T v^T(p)v(p)w + \lambda(w^T E_h - 1) = \frac{1}{2} w^T V(k)w + \lambda(w^T E_h - 1), \quad (2.10)$$

після чого, вирішуючи систему рівнянь

$$\begin{cases} \nabla_w L(w, \lambda) = V(k)w + \lambda E_h = 0, \\ \partial L(w, \lambda) / \partial \lambda = w^T E_h - 1 = 0, \end{cases} \quad (2.11)$$

отримуємо

$$\begin{cases} w = V^{-1}(k)E_h(E_h^T V^{-1}(k)E_h)^{-1}, \\ \lambda = -E_h^T V^{-1}(k)E_h, \end{cases} \quad (2.12)$$

при цьому функція Лагранжа (2.10) в сідлової точці має значення

$$L^*(w, \lambda) = (E_h^T V^{-1}(k)E_h)^{-1}. \quad (2.13)$$

Далі розглянемо довільну пару векторів a та b і запишемо нерівність Коши-Шварца у формі

$$\begin{aligned} (a^T b)^2 &= (a^T V^{\frac{1}{2}}(k)V^{-\frac{1}{2}}(k)b)^2 = ((V^{\frac{1}{2}}(k)a)^T (V^{-\frac{1}{2}}(k)b))^2 \leq \\ &\leq \left\| V^{\frac{1}{2}}(k)a \right\|^2 \left\| V^{-\frac{1}{2}}(k)b \right\|^2 = (a^T V(k)a)(b^T V^{-1}(k)b). \end{aligned} \quad (2.14)$$

Також ведемо $(h \times 1)$ -вектор $E_{h,l}$, утворений нулями крім l -го елемента, який дорівнює одиниці, та перепишемо (2.14) у вигляді

$$(E_h^T E_{h,l})^2 \leq (E_{h,l}^T V(k)E_{h,l})(E_h^T V^{-1}(k)E_h), \quad (2.15)$$

звідки виходить

$$1 \leq v_{ll}(k)(E_h^T V^{-1}(k)E_h), \quad (2.16)$$

або

$$v_{ll}(k) = \sum_{p=1}^k \|d(p) - y_l(p)\|^2 = \sum_{p=1}^k \|v_l(p)\|^2 \geq (E_h^T V^{-1} E_h)^{-1} = L^*(w, \lambda), \quad (2.17)$$

де $v_{ll}(k)$ – діагональний елемент матриці $V(k)$.

З (2.17) виходить, що об'єднаний вихідний сигнал $y^*(k)$ не поступається по точності найкращому з локальних виходів $y_l(k)$, сформованому l -ю мережею, що входить в ансамбль.

З тим, щоб забезпечити обробку інформації в реальному часі, вираз (2.7) має бути поданий в рекурентній формі, яка за допомогою формули Шермана-Моррісона-Вудбері набуває вигляду

$$\begin{cases} P(k+1) = P(k) - P(k)y^T(k+1)(I_m + y(k+1)P(k)y^T(k+1))^{-1}y(k+1)P(k) = \\ \quad = (I_m - P(k)y^T(k+1)y(k+1))^{-1}P(k), \\ r(k+1) = r(k) + y^T(k+1)d(k+1), \\ w^*(k+1) = P(k+1)r(k+1), \\ w(k+1) = w^*(k+1) + P(k+1)(E_h^T P(k+1)E_h)^{-1}(1 - E_h^T w^*(k+1))E_h. \end{cases} \quad (2.18)$$

Громіздкість алгоритму (2.18), що є по суті гауссівсько-ньютонівської процедурою нелінійного програмування, пов'язана перш за все з необхідністю звернення на кожному такті часу k матриць розмірності $(m \times m)$, що змушує вдаватися до більш простих градієнтних алгоритмів налаштування вектору ваг $w(k)$. Досить ефективна процедура може бути отримана, якщо для пошуку сідлової точки лагранжіана використовувати градієнтний алгоритм Ерроу-Гурвіца, який має вигляд [14]

$$\begin{cases} w(k+1) = w(k) - \eta_w(k) \nabla_w L(w, \lambda), \\ \lambda(k+1) = \lambda(k) + \eta_\lambda(k) \partial L(w, \lambda) / \partial \lambda, \end{cases} \quad (2.19)$$

або для певної функції (2.5) –

$$\begin{cases} w(k+1) = w(k) + \eta_w(k)(y^T(k)e_w(k) - \lambda(k)E_h), \\ \lambda(k+1) = \lambda(k) + \eta_\lambda(k)(w^T(k)E_h - 1), \end{cases} \quad (2.20)$$

де

$$e_w(k) = d(k) - y^*(k) = d(k) - y(k)w(k). \quad (2.21)$$

Процедура Ерроу-Гурвіца сходиться до сідлової точки лагранжіану при досить загальних припущеннях про значення параметрів кроку $\eta_w(k)$, $\eta_\lambda(k)$, однак для скорочення часу налаштування, можна спробувати ці параметри оптимізувати. Домножимо зліва перший співвідношення (2.19) на $y(k)$ і віднімемо обидві частини отриманого виразу з $d(k)$, тобто

$$y(k)w(k+1) = y(k)w(k) - \eta_w y(k) \nabla_w L(w, \lambda), \quad (2.22)$$

та

$$d(k) - y(k)w(k+1) = d(k) - y(k)w(k) + \eta_w y(k) \nabla_w L(w, \lambda). \quad (2.23)$$

Ліва частина (2.23) описує апостеріорну похибку $e_w^A(k)$, отриману після одного такту налаштування параметрів, тобто

$$e_w^A(k) = e_w(k) + \eta_w(k) y(k) \nabla_w L(w, \lambda). \quad (2.24)$$

Вводячи квадрат норми цього співвідношення

$$\|e_w^A(k)\|^2 = \|e_w(k)\|^2 + 2\eta_w(k)e_w^T(k)y(k)\nabla_w L(w, \lambda) + \eta_w^2(k)\|y(k)\nabla_w L(w, \lambda)\|^2 \quad (2.25)$$

та мінімізуючи його по $\eta_w(k)$, тобто вирішуючи диференційне рівняння

$$\partial \|e_w^A(k)\|^2 / \partial \eta_w = 0, \quad (2.26)$$

отримуємо оптимальне значення кроку пошуку

$$\eta_w(k) = -\frac{e_w^T(k)y(k)\nabla_w L(w, \lambda)}{\|y(k)\nabla_w L(w, \lambda)\|^2}. \quad (2.27)$$

Тоді алгоритм (2.19), (2.20) остаточно може бути записано у вигляді

$$\begin{cases} \nabla_w L(w, \lambda, k) = -(y^T(k)e_w(k) - \lambda(k)E_h), \\ w(k+1) = w(k) + \frac{e_w^T(k)y(k)\nabla_w L(w, \lambda, k)}{\|y(k)\nabla_w L(w, \lambda, k)\|^2} \nabla_w L(w, \lambda, k), \\ \lambda(k+1) = \lambda(k) + \eta_\lambda(k)(w^T(k)E_h - 1). \end{cases} \quad (2.28)$$

Нескладно бачити, що процедура (2.28) з обчислювальної точки зору набагато простіше ніж (2.18), а при відсутності обмежень (2.2) являє собою узагальнення на багатовимірний випадок алгоритму навчання Качмажа-Уйроу-Хоффа.

Елементом вагового вектору можна надати сенс ймовірностей або рівнів належності, що визначають «внесок» кожного з членів ансамблю в об'єднаний результат, якщо в оптимізується лагранжیان ввести додаткове обмеження на від'ємність одержуваних ваг, тобто

$$E_h^T \mu = \sum_{l=1}^h \mu_l = 1, \quad 0 \leq \mu_l \leq 1, \quad (2.29)$$

де $\mu = (\mu_1, \mu_2, \dots, \mu_h)^T$ – вектор невід'ємних вагових коефіцієнтів в об'єднаному сигналі

$$y_{\mu}^*(k) = \sum_{l=1}^h \mu_l y_l(k) = y(k)\mu. \quad (2.30)$$

Вводячи до розгляду лагранжیان

$$\begin{aligned} L(\mu, \lambda, \rho) &= \frac{1}{2} \text{Tr} e^{kT} e^k + \lambda(\mu^T E_h - 1) - \rho^T \mu = \\ &= \frac{1}{2} \text{Tr}(D(k) - Y(k)I_m \otimes \mu)^T (D(k) - Y(k)I_m \otimes \mu) + \lambda(\mu^T E_h - 1) - \rho^T \mu = \\ &= \frac{1}{2} \sum_{p=1}^k \|d(p) - y(p)\mu\|^2 + \lambda(\mu^T E_h - 1) - \rho^T \mu \end{aligned} \quad (2.31)$$

(тут $\rho - (h \times 1)$ – вектор невід'ємних вагових коефіцієнтів в об'єднаному сигналі) і систему рівнянь Куна-Таккера

$$\begin{cases} \nabla_{\mu} L(\mu, \lambda, \rho) = 0, \\ \partial L(\mu, \lambda, \rho) / \partial \lambda = 0, \\ \rho_l \geq 0, \quad l = 1, 2, \dots, h, \end{cases} \quad (2.32)$$

рішення якого має вигляд

$$\begin{cases} \mu = P(k)(r(k) - \lambda E_h + \rho), \\ \lambda = \frac{E_h^T P(k)r(k) - 1 + E_h^T P(k)\rho}{E_h^T P(k)E_h}, \end{cases} \quad (2.33)$$

і застосувавши далі процедуру Ерроу-Гурвіца-Удзави, отримуємо алгоритм настройки вектору вагових коефіцієнтів μ у вигляді

$$\left\{ \begin{array}{l} \mu(k+1) = w^*(k+1) - P(k+1) \frac{E_h^T w^*(k+1) - 1 + E_h^T P(k+1) \rho(k)}{E_h^T P(k+1) E_h} \times \\ \quad \times E_h + P(k+1) \rho(k), \\ \rho(k+1) = \text{Pr}_+(\rho(k) - \eta_\rho(k) \mu(k)), \end{array} \right. \quad (2.34)$$

де $\text{Pr}_+(\bullet)$ – проектор на позитивний ортант.

Перше співвідношення (2.34) можна перетворити до вигляду

$$\begin{aligned} \mu(k+1) &= w(k+1) - P(k+1) \frac{E_h^T P(k+1) \rho(k)}{E_h^T P(k+1) E_h} E_h + P(k+1) \rho(k) = \\ &= w(k+1) + \left(I_h - \frac{P(k+1) E_h E_h^T}{E_h^T P(k+1) E_h} \right) P(k+1) \rho(k), \end{aligned} \quad (2.35)$$

де $w(k+1)$ визначається співвідношенням (2.7);

$I_h - (h \times h)$ – одинична матриця;

$I_h - P(k+1) E_h E_h^T (E_h^T P(k+1) E_h)^{-1}$ – проектор на гіперплощину $\mu^T E_h = 1$.

Нескладно побачити також, що вектор $(I_h - P(k+1) E_h E_h^T (E_h^T P(k+1) E_h)^{-1}) P(k+1) \rho(k)$ ортогонален вектору E_h , що дозволяє переписати (2.35) в простій формі

$$\left\{ \begin{array}{l} \mu(k+1) = w(k+1) + \text{Pr}_{\mu^T E_h = 1} (P(k+1) \rho(k)), \\ \rho(k+1) = \text{Pr}_+(\rho(k) - \eta_\rho(k) \mu(k)). \end{array} \right. \quad (2.36)$$

Тоді остаточно рекурентний алгоритм налаштування вагових векторів w^* , w та μ набуває вигляду

$$\left\{ \begin{array}{l} P(k+1) = P(k) - P(k)y^T(k+1)(I_m + y(k+1)P(k)y^T(k+1))^{-1}y(k+1)P(k) = \\ \quad = (I_m - P(k)y^T(k+1)y(k+1))^{-1}P(k), \\ r(k+1) = r(k) + y^T(k+1)d(k+1), \\ w^*(k+1) = P(k+1)r(k+1), \\ w(k+1) = w^*(k+1) + P(k+1)(E_h^T P(k+1)E_h)^{-1}(1 - E_h^T w^*(k+1))E_h, \\ \mu(k+1) = w(k+1) - P(k+1)\frac{E_h^T P(k+1)\rho(k)}{E_h^T P(k+1)E_h}E_h + P(k+1)\rho(k), \\ \rho(k+1) = \text{Pr}_+(\rho(k) - \eta_\rho(k)\mu(k)). \end{array} \right. \quad (2.37)$$

Процедура обчислення оцінок вкладів членів ансамблю (2.34) може бути істотно спрощена, якщо для пошуку сідлової точки лагранжіана (2.31) аналогічно попередньому скористатися градієнтним алгоритмом

$$\left\{ \begin{array}{l} \mu(k+1) = \mu(k) - \eta_\mu(k)\nabla_\mu L(\mu, \lambda, \rho), \\ \lambda(k+1) = \lambda(k) + \eta_\lambda(k)(\mu^T(k)E_h - 1), \\ \rho(k+1) = \text{Pr}_+(\rho(k) - \eta_\rho(k)\mu(k)), \end{array} \right. \quad (2.38)$$

або

$$\left\{ \begin{array}{l} \mu(k+1) = \mu(k) + \eta_\mu(k)(y^T(k)e_\mu(k) - \lambda(k)E_h + \rho(k)), \\ \lambda(k+1) = \lambda(k) + \eta_\lambda(k)(\mu^T(k)E_h - 1), \\ \rho(k+1) = \text{Pr}_+(\rho(k) - \eta_\rho(k)\mu(k)), \end{array} \right. \quad (2.39)$$

де

$$e_\mu(k) = d(k) - y(k)\mu(k) = d(k) - y_\mu^*(k). \quad (2.40)$$

Проводячи перетворення такі ж самі як в (2.22) – (2.26), можна отримати оптимальне значення кроку пошуку

$$\eta_{\mu}(k) = -\frac{e_{\mu}^T(k)y(k)\nabla_{\mu}L(\mu, \lambda, \rho)}{\|y(k)\nabla_{\mu}L(\mu, \lambda, \rho)\|^2}, \quad (2.41)$$

після чого остаточно записати алгоритм налаштування вектору $\mu(k)$:

$$\begin{cases} \nabla_{\mu}L(\mu, \lambda, \rho, k) = -(y^T(k)e_{\mu}(k) - \lambda(k)E_h + \rho(k)), \\ \mu(k+1) = \mu(k) + \frac{e_{\mu}^T(k)y(k)\nabla_{\mu}L(\mu, \lambda, \rho, k)}{\|y(k)\nabla_{\mu}L(\mu, \lambda, \rho, k)\|^2} \nabla_{\mu}L(\mu, \lambda, \rho, k), \\ \lambda(k+1) = \lambda(k) + \eta_{\lambda}(k)(\mu^T(k)E_h - 1), \\ \rho(k+1) = \text{Pr}_+(\rho(k) - \eta_{\rho}(k)\mu(k)). \end{cases} \quad (2.42)$$

Нескладно побачити, що у випадку невід'ємності всіх компонент вектору $\mu(k)$, алгоритм (2.42) автоматично набуває вигляду процедури (2.28). На рисунку 2.2 наведена архітектура системи обчислювального інтелекту, яка вирішує задачу синтезу ансамблю локальних систем.

На вхід ансамблю надходить для опрацювання багатовимірний сигнал $x(k)$, що подається далі на перший шар, утворений h паралельно функціонуючими системами, які по факту вирішують одну і туж саму задачу (класифікації, кластеризації і т.п.), що відрізняються один від одного нейронами, архітектурою, алгоритмами навчання і формують набір векторних локальних виходів $y_l(k)$, $l = 1, 2, \dots, h$.

Ці локальні сигнали надходять на вихідний шар, утворений двома нейронами (ALA), що представляють собою по суті багатовимірні аналоги стандартних адаптивних лінійних асоціатор і обчислює об'єднані вихідні сигнали

$$\begin{cases} y^*(k) = y(k)w(k), & ALA_w, \\ y_\mu^*(k) = y(k)\mu(k), & ALA_\mu. \end{cases} \quad (2.43)$$

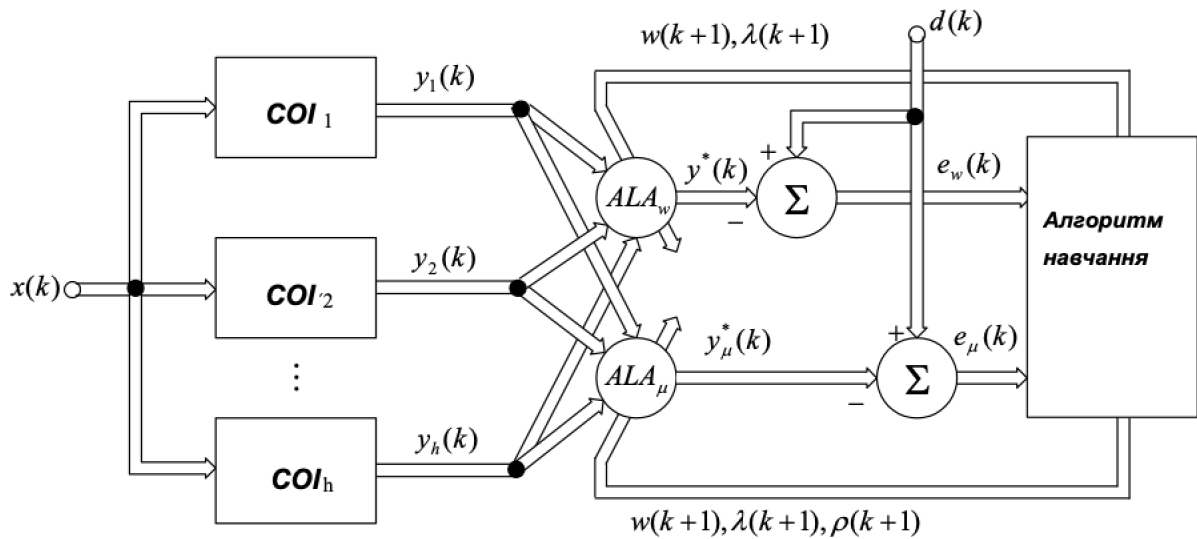


Рисунок 2.2 – Архітектура ансамблю систем обчислювального інтелекту

Навчання нейронів ALA_w та ALA_μ , виконується за допомогою алгоритмів (2.28) та (2.42) відповідно. Використання такого підходу дозволяє підвищити якість обробки інформації на основі ансамблю систем обчислювального інтелекту і визначити системи-переможці, які найкращим чином пристосовані до вирішення даної задачі і вносять найбільший внесок у її рішення.

Розглянутий підхід доцільно застосувати та перенести його ідею на систему, яка буде складатися з певної кількості систем для нечіткої кластеризації з блоком прийняття рішень, в якому буде проводитись вибір найкращого з окремих блоків для кластеризації. В якості складових цієї системи буде використано нечітку систему Дж. Безденка FCM (Fuzzy C-means) з різним показником коефіцієнту фазифікації.

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

У цьому розділі описані архітектура та метод навчання каскадної нейромережі для нечіткого кластерування, зокрема потоків даних і розглянуті особливості та труднощі послідовного кластерування, та описані два підходи, переваги яких поєднує у собі пропонована система: нечітке та ієрархічне кластерування.

3.1 Труднощі та особливості відомих методів кластерування даних

Завдання кластерування (класифікування 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, пропонована у цьому розділі самонавчання 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 сприйнятливості до локальних мінімумів, адже вони ґрунтуються на оптимізації певної цільової функції. Типові

методи розподільного кластерування мають обчислювальну складність $O(N)$ для тренувальної вибірки розміру N .

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 пропонується самонавчання система є у певному сенсі альтернативою системам висхідного ієрархічного кластерування, що здатна працювати в онлайн режимі).

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 винятком 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 належати іншому, що унеможлиблює створення модифікацій методів для послідовного кластерування, на відміну від розподільного кластерування.

Методи ієрархічного кластерування здебільшого мають обчислювальну складність принаймні $O(N^2)$, що робить їх використання недоцільним для великих масивів даних. Також ці методи використовують евристичний підхід до встановлення оптимальної кількості кластерів, що може привести до невірної розбиття даних.

3.2. Нечітке послідовне кластерування та критерії дійсності

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 [16], а для його вирішення широко використовується апарат обчислювального інтелекту [17] і, насамперед, нейро-фаззі підхід. 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 $x(1), x(2), \dots, x(N)$, 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 ($J = 1, 2, \dots, M$). 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 (Dynamic Data Mining, Data Stream Mining) [17] 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.

Самоорганізовані мапи Кохонена [18] 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».

Самоорганізованні мапи також відомі своєю ефективністю вирішення задачі кластерування класів, що перетинаються. 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 самоорганізованні мапи Кохонена (SOM) та метод нечітких c -середніх Бездека [19].

Такі гібридні системи володіють обширною функціональністю завдяки використанню спеціальних методів налаштування, що ґрунтуються на процедурах оптимізації прийнятої цільової функції, але потребують попередньо заданої кількості кластерів та фіксованого значення фазіфікатора.

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

Індекс розбиття ентропії PE – ще один критерій дійсності нечіткого кластерування, запропонований у [20], що залежить лише від значень функції належності

$$PE = -\frac{1}{N} \sum_{l=1}^M \sum_{i=1}^N u_{li} \log_a(u_{li}). \quad (3.1)$$

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.

Фукуяма та Сугено запропонували індекс дійсності нечіткого

кластерування, залежний як від рівнів належності так і від самих вхідних даних:

$$FS = \sum_{i=1}^N \sum_{l=1}^M u_{li}^{\beta} (\|x_i - z_l\|^2 - \|z_l - z\|^2) \quad (3.2)$$

де z та z_l – середнє арифметичне усієї виборки та образів віднесених до кластеру M_l відповідно.

З визначення (3.2) видно, що малі значення FS говорять про компактні добре визначені кластери.

Нечітка множина i -ого образу визначається як

$$\tilde{A}_l = \sum_{i=1}^N \frac{u_{li}}{x_i}, l = 1, 2, \dots, M. \quad (3.3)$$

Ступінь, в якій \tilde{A}_l є підмножиною \tilde{A}_p визначається таким чином

$$\begin{cases} S(\tilde{A}_l, \tilde{A}_p) = \frac{S(\tilde{A}_l, \tilde{A}_p) + S(\tilde{A}_p, \tilde{A}_l)}{2}, \\ U(\tilde{A}_j) = \sum_{i=1}^N U_{ji}. \end{cases} \quad (3.4)$$

Зважаючи на (3.4), можна запропонувати такі варіанти обчислення міри подібності:

$$N_1(\tilde{A}_l, \tilde{A}_p) = \frac{S(\tilde{A}_l, \tilde{A}_p) + S(\tilde{A}_p, \tilde{A}_l)}{2}, \quad (3.5a)$$

$$N_2(\tilde{A}_l, \tilde{A}_p) = \min(S(\tilde{A}_l, \tilde{A}_p), S(\tilde{A}_p, \tilde{A}_l)), \quad (3.5b)$$

$$N_3(\tilde{A}_l, \tilde{A}_p) = S(\tilde{A}_l \cup \tilde{A}_p, \tilde{A}_p \cap \tilde{A}_l). \quad (3.5c)$$

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

$$FSim = \max_{1 \leq l \leq M} \min_{\substack{1 \leq p \leq M, \\ p \neq l}} N(\tilde{A}_l, \tilde{A}_p), \quad (3.6)$$

де міру нечіткої подібності $N(\tilde{A}_l, \tilde{A}_p)$ можна знайти за будь-яким виразом (3.6).

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

Архітектура каскадної нейронної мережі для опрацювання даних в послідовному режимі, наведена на рисунку 3.1. Її вузлами є модифіковані нейрони Оя (рисунок 3.1), крім того, вона містить додатковий акумулятор і блок прийняття рішень (DM), в якому оцінюється необхідність додавання додаткового нейрона в мережу.

Мережа починає свою роботу з єдиного нейрона MON_1 , при цьому, якщо точність відновленого сигналу $\hat{x}_1(k) = w_1(k)y_1(k)$ нижче деякого заданого порогу ε_{EST} , приймається рішення про додавання другого нейрона MON_2 так, як це показано на рисунку 3.1.

При цьому слід зауважити, що додавання другого і наступних нейронів (каскадів) ніяк не впливає на роботу попередніх вузлів. Процес нарощування каскадів триває до тих пір, поки відновлений сигнал

$$\hat{x}_1(k) = \sum_{j=1}^m y_j(k)w_j(k) \quad (3.7)$$

по точності не задовольнятиме критерієм оцінювання

$$E_{EST}^k = \frac{1}{k} \sum_k \frac{\|\tilde{x}(k) - \hat{x}(k)\|^2}{\|\tilde{x}(k)\|^2} \leq \varepsilon_{EST} \quad (3.8)$$

Як тільки значення E_{EST}^k стає менше рівня порога ε_{EST} , блок DM припиняє нарощування числа каскадів. У разі істотної зміни властивостей сигналу $\tilde{x}(k)$ і порушення нерівності (3.15), процес зростання мережі може бути відновлений.

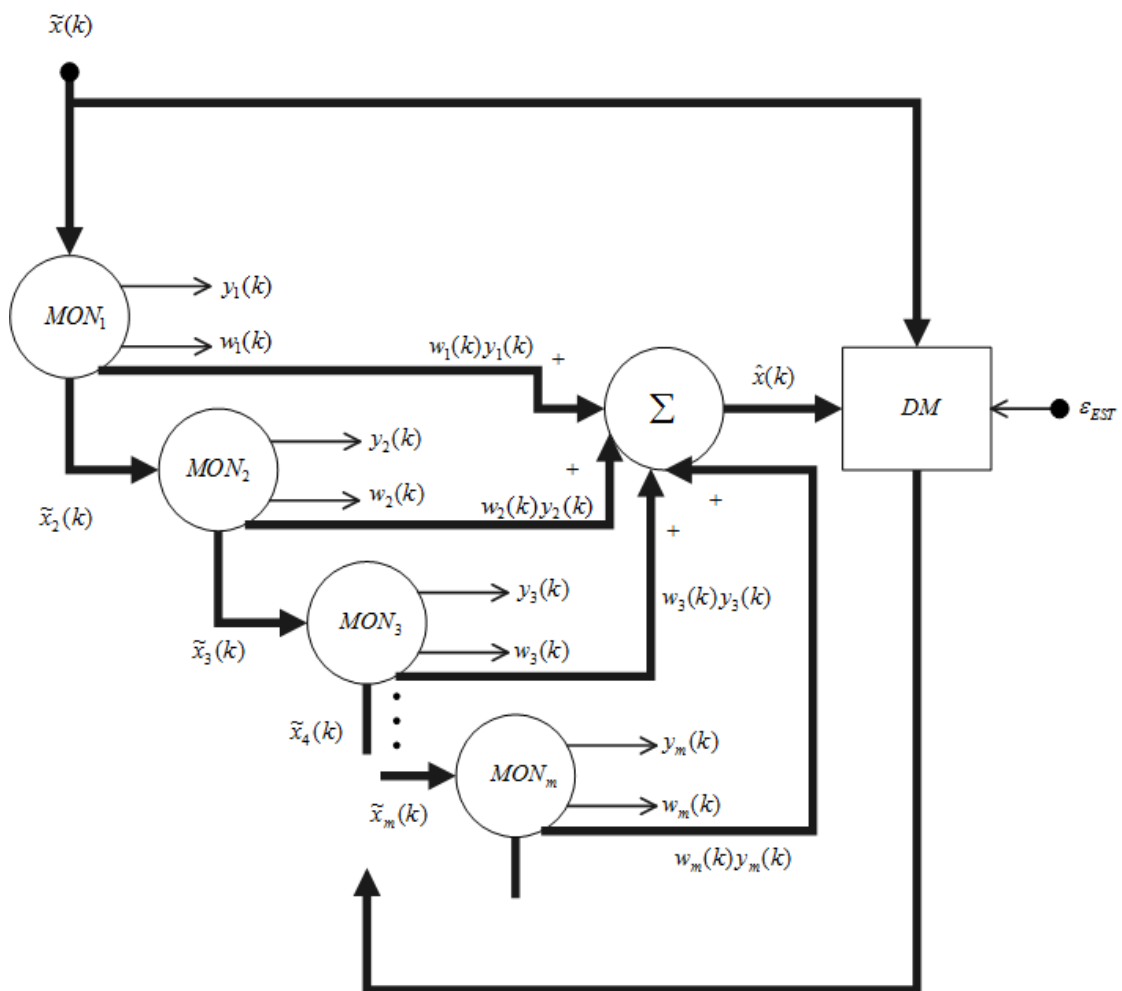


Рисунок 3.1 – Каскадна адаптивна нейронна мережа для послідовного опрацювання даних

Слід зазначити, що архітектура на рисунку 3.1 будується в рамках конструктивного підходу, тобто може нарощувати число своїх нейронів від одного до $n-1$. Для вирішення завдання послідовного аналізу головних компонент може бути використаний і деструктивний підхід, коли вихідна мережа створюється з максимально можливого числа нейронів $n-1$, яке скорочується в процесі самонавчання.

До нульового шару запропонованої системи послідовно передаються дані у формі векторного сигналу $x(k) = (x_1(k), x_2(k), \dots)^T$. Вхідні сигнали надходять до всіх вузлів системи $N_j^{[m]}$, де $j = 1, 2, \dots, q$ – кількість вузлів у пулі-ансамблі, $m = 1, 2, \dots$ – номер каскаду. Вузол кожного каскаду призначений для онлайн кластерування потоку даних і відрізняється від вузлів-сусідів методом 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 загальну якість кластеризації пулу, враховуючи прийнятну кількість кластерів $m + 1$.

Таким чином, система розв'язує задачу кластерування нестационарного потоку даних в умовах невизначеності щодо кількості кластерів, а також їх вигляду і рівню взаємного перекриття. І, нарешті, вихідний вузол системи XB^* , порівнюючи якість кластерування 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, який надходить до системи, може оброблятися у паралельному режимі вузлами системи $N_j^{[m]}$ [21].

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

В основі методів навчання вузлів системи лежать алгоритми нечіткого кластерування, засновані на цільових функціях, такі, що вирішують задачу їх оптимізації за деяких апріорних припущень. Найбільш поширеним є ймовірнісний підхід, заснований на мінімізації цільової функції

$$E(u_{jl}^{[m]}(k), c_{jl}^{[m]}) = \sum_{k=1}^N \sum_{l=1}^{m+1} (u_{jl}^{[m]}(k))^{\beta} \|x(k) - c_{jl}^{[m]}\|^2 \quad (3.9)$$

при обмеженнях

$$\sum_{l=1}^{m+1} u_{jl}^{[m]}(k) = 1, \quad 0 \leq \sum_{k=1}^N u_{jl}^{[m]}(k) \leq N, \quad (3.10)$$

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

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

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

N – кількість образів у вхідній виборці, що, у рамках класичного підходу Бездека, вважається незмінною та такою, що задана апріорі.

Вводячи функцію Лагранжа

$$\begin{aligned}
L\left(u_{jl}^{[m]}(k), c_{jl}^{[m]}, \lambda_j^{[m]}(k)\right) = \\
= \sum_{k=1}^N \sum_{l=1}^{m+1} \left(u_{jl}^{[m]}(k)\right)^\beta \left\|x(k) - c_{jl}^{[m]}\right\|^2 \\
+ \sum_{k=1}^N \lambda_j^{[m]}(k) \left(\sum_{l=1}^{m+1} u_{jl}^{[m]}(k) - 1\right),
\end{aligned} \tag{3.11}$$

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

$$\left\{ \begin{aligned}
u_{jl}^{[m]}(k) &= \frac{\left(\left\|x(k) - c_{jl}^{[m]}\right\|^2\right)^{\frac{1}{1-\beta}}}{\sum_{l=1}^{m+1} \left(\left\|x(k) - c_{jl}^{[m]}\right\|^2\right)^{\frac{1}{1-\beta}}}, \\
c_{jl}^{[m]} &= \frac{\sum_{k=1}^N \left(u_{jl}^{[m]}(k)\right)^\beta x(k)}{\sum_{k=1}^N \left(u_{jl}^{[m]}(k)\right)^\beta}, \\
\lambda_j^{[m]}(k) &= - \left(\left(\sum_{l=1}^{m+1} \beta \left\|x(k) - c_{jl}^{[m]}\right\|^2 \right)^{\frac{1}{1-\beta}} \right)^{\frac{1}{1-\beta}},
\end{aligned} \right. \tag{3.12}$$

що при $\beta = 2$ збігається з алгоритмом нечітких c -середніх Бездека (FCM) [21] і приймає форму

$$\begin{cases} u_{jl}^{[m]}(k) = \frac{\left(\|x(k) - c_{jl}^{[m]}\|^2\right)^{-2}}{\sum_{l=1}^{m+1} \left(\|x(k) - c_{jl}^{[m]}\|^2\right)^{-2}}, \\ c_{jl}^{[m]} = \frac{\sum_{k=1}^N \left(u_{jl}^{[m]}(k)\right)^2 x(k)}{\sum_{k=1}^N \left(u_{jl}^{[m]}(k)\right)^2}. \end{cases} \quad (3.13)$$

Тут варто відзначити, що вибір фаззифікатору $\beta = 2$ в (3.13) не дає жодних переваг порівняно з довільним значенням β у (3.12), у зв'язку з чим пропонується використовувати різні значення параметра фаззифікації для кожного вузла пулу каскаду, після чого вибирати найкращий результат залежно від прийнятого критерію якості нечіткого кластерування [22].

Для послідовної обробки потоку даних, що надходять в онлайн режимі, у [23] були запропоновані рекурентні алгоритми, в основі яких лежить процедура нелінійного програмування Ерроу-Гурвіца-Удзави [23]. Так, пакетному алгоритмові (3.13) відповідає вираз

$$\begin{cases} u_{jl}^{[m]}(k+1) = \frac{\|x(k+1) - c_{jl}^{[m]}(k)\|^{\frac{1}{1-\beta}}}{\sum_{l=1}^{m+1} \|x(k+1) - c_{jl}^{[m]}(k)\|^{\frac{1}{1-\beta}}} \\ c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \eta(k+1) \left(u_{jl}^{[m]}(k+1)\right)^{\beta_j} \left(x(k+1) - c_{jl}^{[m]}(k)\right), \end{cases} \quad (3.14)$$

де $\eta(k+1)$ – параметр кроку навчання), що є узагальненням алгоритму навчання Чанга-Лі і при $\beta = 2$ близьке до градієнтної процедури Парка-Дегера [24]:

$$\begin{cases} u_{jl}^{[m]}(k+1) = \frac{\|x(k+1) - c_{jl}^{[m]}(k)\|^{-2}}{\sum_{l=1}^{m+1} \|x(k+1) - c_{jl}^{[m]}(k)\|^{-2}}, \\ c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \eta(k+1) \left(u_{jl}^{[m]}(k+1)\right)^{-2} \left(x(k+1) - c_{jl}^{[m]}(k)\right). \end{cases} \quad (3.15)$$

Варто зауважити, що, розглянувши співвідношення (3.15) з позицій навчання Кохоненової самоорганізовної мапи (SOM) [24], можна помітити, що множник $\left(u_{jl}^{[m]}\right)^{\beta_i}$ відповідає функції сусідства в правилі навчання на основі принципу «переможцю дістається більше», маючи при цьому дзвонуватий вигляд.

Вочевидь, у випадку, коли $\beta = 1$ та $u_{jl}^{[m]}(k) \in [0, 1]$, процедура (3.15) збігається з чітким алгоритмом c -середніх (НСМ), коли ж $\beta = 0$, маємо стандартне правило навчання Кохонена «переможцю дістається все» [24]

$$c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \eta(k+1) \left(x(k+1) - c_{jl}^{[m]}(k)\right), \quad (3.7)$$

запропоноване Каш'япом та Блейдоном [25] у шістдесятих роках минулого століття. Легко побачити, що процедура (3.16) оптимізує цільову функцію

$$E(c_{jl}^{[m]}) = \sum_{k=1}^N \|x(k) - c_{jl}^{[m]}\|^2, \quad \sum_{l=1}^{m+1} N_l = N, \quad (3.17)$$

мінімум якої збігається із середнім арифметичним

$$c_{jl}^{[m]} = \frac{1}{N} \sum_{k=1}^{N_l} x(k), \quad (3.18)$$

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

Якщо записати (3.17) у рекурентній формі, отримаємо оптимальний алгоритм самонавчання Ципкіна [26]

$$c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \frac{1}{N_l(k+1)} \left(x(k+1) - c_{jl}^{[m]}(k) \right), \quad (3.19)$$

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

У загальному випадку метод навчання (3.19) 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 (3.18).

Вузли $N_{jl}^{[m]u}$ обчислюють рівні належності згідно першому співвідношенню (3.19). Вузли $N_{jl}^{[m]}$ кожного з каскадів відрізняються тільки фазифікатором алгоритму самонавчання, а вузол кожного наступного каскаду містить додатково один нейрон Кохонена і один елемент для розрахунку рівнів належності.

3.5. Керування каскадами самонавчання нейро-фаззі системи, що еволюціонує

Якість кластерування кожного вузла системи може бути оцінена за допомогою будь-якого з індексів, що використовується у задачах нечіткого кластерування [27]. Одним 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

$$PC_j^{[m]} = \frac{1}{N} \sum_{k=1}^N \sum_{l=1}^{m+1} \left(u_{jl}^{[m]}(k) \right)^2 \quad (3.8)$$

Цей коефіцієнт має ясний фізичний зміст: що краще 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 онлайн режимі

$$PC_j^{[m]}(k + 1) = PC_j^{[m]}(k) + \frac{1}{k + 1} \left(\sum_{l=1}^{m+1} \left(u_{jl}^{[m]}(k + 1) \right)^2 - PC_j^{[m]}(k) \right). \quad (3.21)$$

Розрахунок коефіцієнту розбиття проводиться для кожного вузла системи разом з налаштуванням їх параметрів, тобто співвідношення (3.20) та (3.21) 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 [28], який для фіксованої вибірки з N спостережень може бути записаний у вигляді

$$XB_j^{[m]} = \frac{\left(\sum_{k=1}^N \sum_{l=1}^{m+1} \left(u_{jl}^{[m]}(k) \right)^2 \left\| x(k) - c_{jl}^{[m]} \right\|^2 \right) / N}{\min_{l \neq q} \left\| c_{jl}^{[m]} - c_{jq}^{[m]} \right\|^2} = \frac{NXB_j^{[m]}}{DXB_j^{[m]}}. \quad (3.22)$$

Вираз (3.22) також можна записати у рекурентній формі

$$XB_j^{[m]}(k+1) = \frac{NXB_j^{[m]}(k+1)}{DXB_j^{[m]}(k+1)} = \frac{NXB_j^{[m]}(k)}{\min_{l \neq q} \left\| c_{jl}^{[m]}(k+1) - c_{jq}^{[m]}(k+1) \right\|^2} + \frac{\sum_{l=1}^{m+1} \left(u_{jl}^{[m]}(k+1) \right)^2 \left\| x(k+1) - c_{jl}^{[m]}(k+1) \right\|^2 - NXB_j^{[m]}(k)}{(k+1) \min_{l \neq q} \left\| c_{jl}^{[m]}(k+1) - c_{jq}^{[m]}(k+1) \right\|^2}, \quad (3.23)$$

при цьому рекурентні вирази (3.22) та (3.23) обчислюються 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 (3.22) та (3.23). 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 фаззифікатору, ефективність роботи кожного каскаду доцільно оцінювати за допомогою розширеного індексу Ксі-Бені EXB [28]

$$EXB_j^{[m]} = \frac{\left(\sum_{k=1}^N \sum_{l=1}^{m+1} \left(u_{jl}^{[m]}(k) \right)^{\beta_{[m]}} \left\| x(k) - c_{jl}^{[m]} \right\|^2 \right) / N}{\min_{l \neq q} \left\| c_{jl}^{[m]} - c_{jq}^{[m]} \right\|^2} = \frac{NEXB_j^{[m]}}{DEXB_j^{[m]}}. \quad (3.24)$$

або його рекурентної форми

$$\begin{aligned} EXB_j^{[m]}(k+1) &= \frac{NEXB_j^{[m]}(k+1)}{DEXB_j^{[m]}(k+1)} \\ &= \frac{NEXB_j^{[m]}(k)}{\min_{l \neq q} \left\| c_{jl}^{[m]}(k+1) - c_{jq}^{[m]}(k+1) \right\|^2} + \end{aligned} \quad (3.25)$$

$$+ \frac{\sum_{l=1}^{m+1} \left(u_{jl}^{[m]}(k+1) \right)^{\beta_{[m]}} \left\| x(k+1) - c_{jl}^{[m]}(k+1) \right\|^2 - NEXB_j^{[m]}(k)}{(k+1) \min_{l \neq q} \left\| c_{jl}^{[m]}(k+1) - c_{jq}^{[m]}(k+1) \right\|^2},$$

де $\beta_{[m]}$ – фаззифікатор найкращого з вузлів m -го каскаду.

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

В даному розділі було розглянуто завдання нечіткого кластерування у режимі послідовного надходження даних до системи та запропоновано метод визначення локально оптимальної кількості кластерів і значення параметру фаззифікації для послідовного кластерування потоків даних.

4 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ САМОНАВЧАННОЇ НЕЙРО-ФАЗЗИ СИСТЕМИ НА ОСНОВІ АНСАМБЛЕВОГО ПІДХОДУ

Одна з основних переваг, притаманних пропонованій самонавчанній нейро-фаззі системі, що еволюціонує, полягає в автоматичному визначенні оптимальної кількості кластерів та значення фаззифікатора на кожному етапі оброблення даних. Першу серію експериментів було проведено на штучно зсинтезованих наборах даних з різним ступенем розмитості та перекриття класів, аби дослідити вплив значення параметру фаззифікації на якість кластерування в режимі реального часу відповідно до обраного критерію дійсності.

Кожен з наборів даних, що їх наведено на рисунку 4.1, містить вісімдесят спостережень з двома ознаками (для очності) у кожному спостереженні.

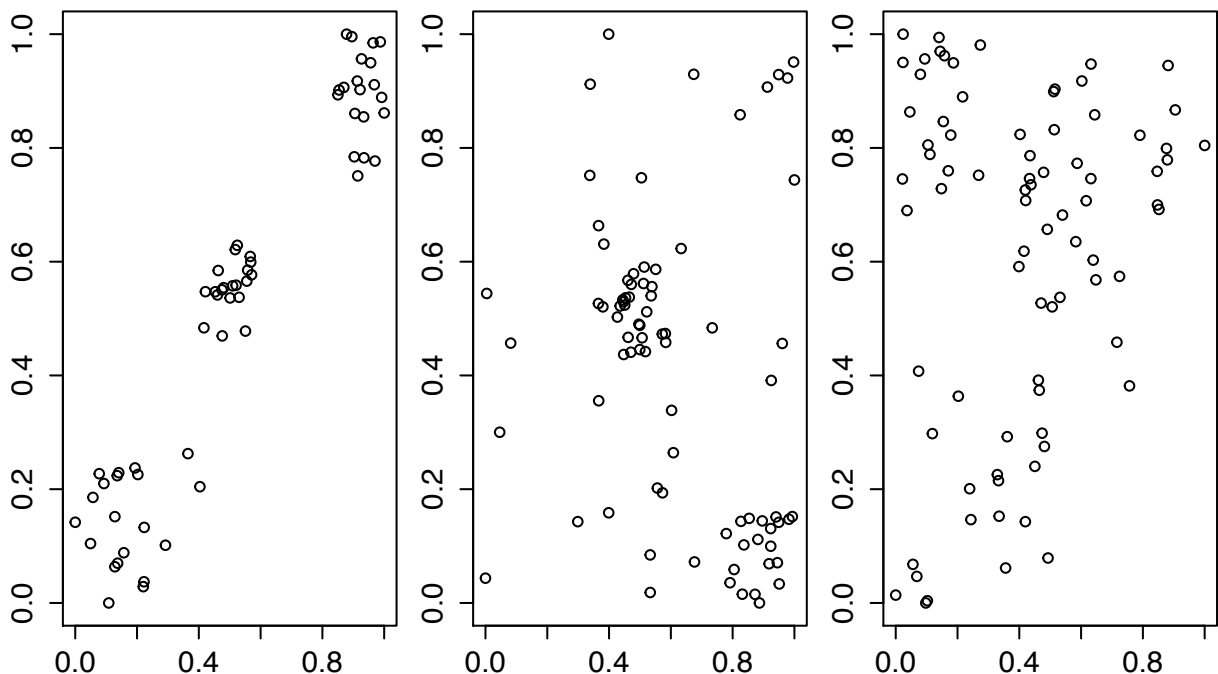


Рисунок 4.1 – Штучно сгенеровані набори даних

Тестові дані були сгенеровані таким чином, аби у першому наборі класи були чітко розподілені (crisp dataset), у другому наборі кластерні границі були дещо розмиті (fuzzy dataset), у третьому випадку класи сильно перетиналися (extra fuzzy dataset). Логічно припустити, що система, яка тестується, надасть перевагу менше значення параметру фаззифікації для першого датасету та більше для останнього, де границі класів спостережень є більш розмитими [29].

Спостереження надходили до нейро-фаззи мережі у послідовному режимі, вагові коефіцієнти нейронів були проініціалізовані, використовуючи пакетну модифікацію обраного алгоритму кластерування на датасеті з довільних двадцятьох спостережень відповідного набору даних (адже система, як і класичний fuzzy c-means, досить чутлива до параметрів ініціалізації).

Локально оптимальні кількість кластерів та значення параметру фаззифікації обумовлювалися максимальним середнім значенням рекурентних коефіцієнту розбиття РС (3.19) та Ксі-Бені індексу (3.21)

$$f\left(PC_j^{[m]}(k), XB_j^{[m]}(k)\right) = \max \frac{PC_j^{[m]}(k) + 1 - XB_j^{[m]}(k)}{2}$$

(у даному випадку використовувалося від'ємне значення Ксі-Бені індексу $1 - XB(k)$, оскільки що меншим є $XB_j^{[m]}(k)$, то ліпшим є розбиття даних на кластери).

Оптимальне розбиття першого набору даних (crisp dataset) наведено на рисунку 4.2.

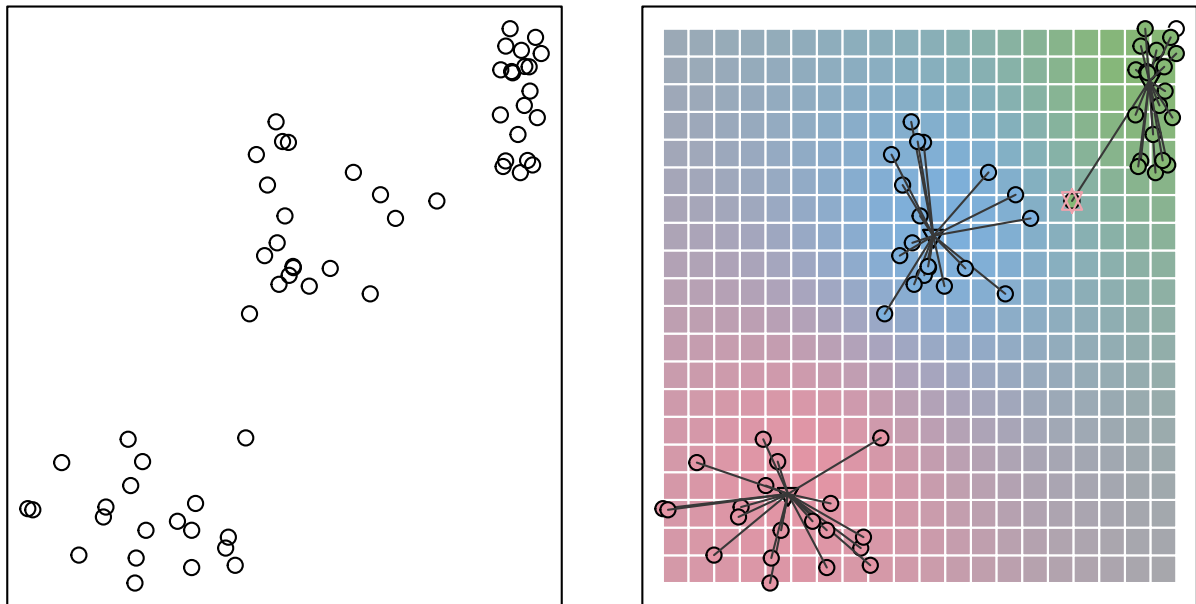


Рисунок 4.2 – Кластерування набору даних з чітко вираженими класами (Crisp dataset)

Для цього випадку, як і передбачалося, оптимальним виявився другий каскад $m = 3$ з трьома кластерами і нейроном-переможцем із найменшим значенням параметру фазифікації $\beta = 2$.

Така конфігурація є оптимальною відповідно до обох використовуваних індексів валідності – найменше значення Ксі-Бені індексу $XV_j^{[m]}$ та найбільший коефіцієнт розбиття $PC_j^{[m]}$: $PC_1^{[2]} = 0.9009951$, $XV_1^{[2]} = 0.03349166$

Лише одне спостереження у цьому датасеті (його позначено багряним квадратом) не належить жодному кластерові зі ступенем більшим від 0.6. Індекси валідності нейронів системи наведені у таблиці 4.1.

Найліпше кластерування (відповідно обраним критеріям якості) другого датасету наведено на рисунку 4.3.

Для набору даних з середньою вираженістю класів найліпшим виявився нейрон другого каскаду $m = 3$ і фаззифікатором $\beta = 3$ (таблиця 4.1).

Таблиця 4.1 – Індеси валідності кластерування I датасету

Каскад I ($m = 2$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.91758	0.7446	0.64787	0.59236
Індекс Ксі-Бені	0.052129	0.061034	0.092235	0.1294
Каскад II ($m = 3$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.92643	0.6609	0.50214	0.43305
Індекс Ксі-Бені	0.027232	0.06872	0.17281	0.26914
Каскад III ($m = 4$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.87218	0.5256	0.37605	0.31993
Індекс Ксі-Бені	0.15687	0.4153	0.84699	1.1765
Каскад IV ($m = 5$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.73909	0.45445	0.32428	0.27063
Індекс Ксі-Бені	0.12985	0.30637	0.68584	1.0551

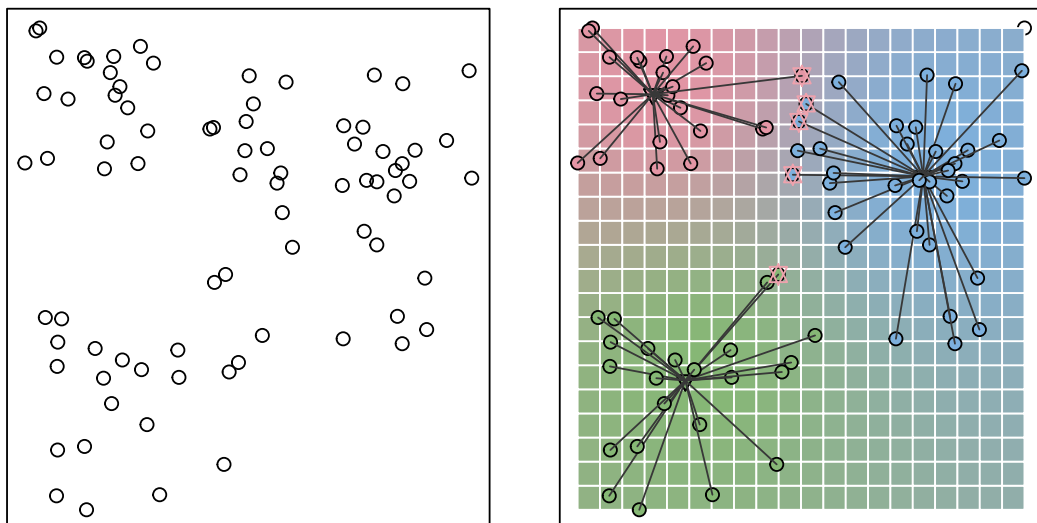


Рисунок 4.3 – Кластерування набору даних з нечіткими межами класів (Fuzzy dataset)

Як показано на рисунку 4.3, декілька спостережень у центрі (позначені багряними квадратами) можна віднести до 2 кластерів з відносно високим ступенем належності, проте більшість спостережень можна чітко розкластерувати, що ілюструється високим значенням коефіцієнту розбиття, та дуже низьким Ксі-Бені індексом: $PC_1^{[2]} = 0.9009951$, $XB_1^{[2]} = 0.03349166$.

Таблиця 4.2 – Індеси валідності кластерування II датасету

Каскад I ($m = 2$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.78414	0.58928	0.53853	0.52239
Індекс Ксі-Бені	0.16668	0.30834	0.3745	0.38723
Каскад II ($m = 3$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.50084	0.71164	0.97275	0.4191
Індекс Ксі-Бені	0.009751	0.031235	0.087474	0.1323
Каскад III ($m = 4$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.91888	0.47532	0.32777	0.28912
Індекс Ксі-Бені	0.052563	0.1757	0.27516	0.33766
Каскад IV ($m = 5$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.85618	0.34327	0.24778	0.22445
Каскад I ($m = 2$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Індекс Ксі-Бені	0.048316	0.19887	0.34307	0.41228
Каскад V ($m = 6$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.81295	0.30709	0.21636	0.19214
Індекс Ксі-Бені	0.060896	0.19702	0.31393	0.38668

Розбиття останнього датасету наведено на рисунку 4.4. на рисунку можна побачити як система розкласифікувала вибірку на чотири кластери.

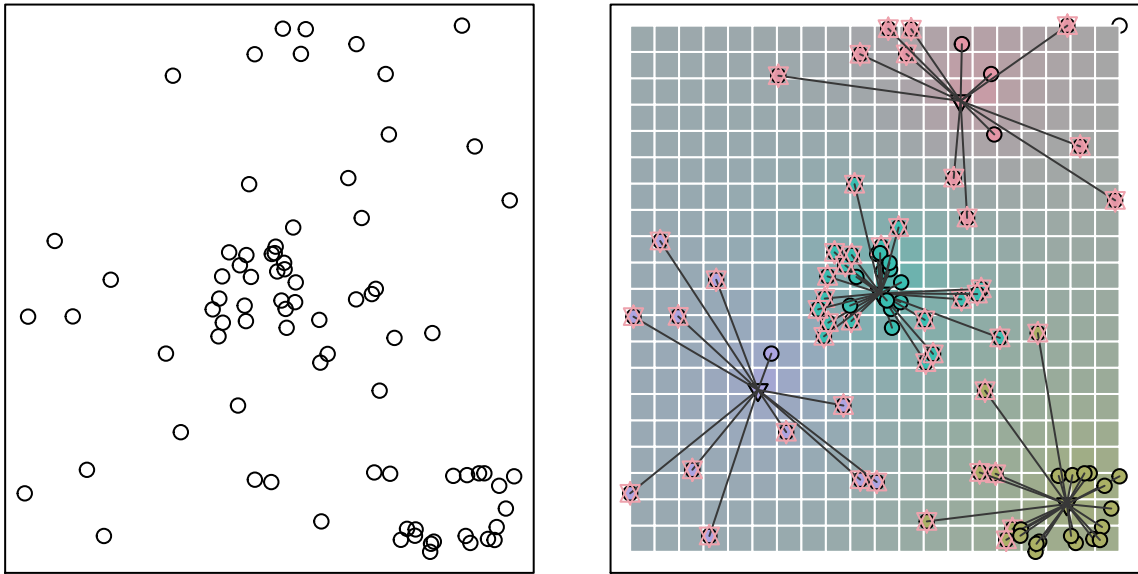


Рисунок 4.4 – Розбиття даних з кластерами, що перетинаються
(Extra fuzzy dataset)

У цьому випадку система обрала нейроном-переможцем вузол третього каскаду ($m = 4$) з високим параметром фазифікації $\beta = 4$: $PC_1^{[2]} = 0.335525$, $XB_1^{[2]} = 0.2128333$. Індеси валідності кластерування останнього датасету наведені у таблиці 4.3.

Для очності у всіх наведених цьому розділі рисунках кольором позначені не тільки розкластеровані спостереження і центри кластерів, а й задній план (фон) малюнків, що дозволяє візуально визначити, до якого кластеру система віднесла б нові спостереження.

Не дивно, що, тоді як для перших двох датасетів важко визначити домінуючий колір, оскільки кластери їх спостережень більш менш компактні та явно виражені, для останнього набору даних домінуючий колір — сірий, сформований кольорами усіх кластерів, що ілюструє великий ступінь перекриття класів і, відповідно, високе значення оптимального параметру фазифікації β , що обрала система.

Таблиця 4.4 – Індокси валідності кластерування III датасету

Каскад I ($m = 2$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.85094	0.71415	0.61734	0.57085
Індекс Ксі-Бені	0.10584	0.11462	0.13797	0.16101
Каскад II ($m = 3$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.61668	0.42848	0.37779	0.35884
Каскад II ($m = 3$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Індекс Ксі-Бені	0.1754	0.20364	0.22364	0.23995
Каскад III ($m = 4$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.33458	0.44082	0.79405	0.29615
Індекс Ксі-Бені	0.20989	0.129	0.051039	0.26282
Каскад IV ($m = 5$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.50244	0.33067	0.26029	0.23318
Індекс Ксі-Бені	0.37268	0.61417	0.79695	0.93626
Каскад V ($m = 6$)	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Коефіцієнт розбиття	0.53279	0.29731	0.22648	0.19858
Індекс Ксі-Бені	0.27407	0.47298	0.60569	0.70716

Ця низка експериментів проілюструвала, як важливо вірно визначати параметр фазифікації, оптимальне значення якого у випадку оброблення даних у послідовному режимі з високою вірогідністю змінюється у часі, а саме здатність визначати оптимальне значення цього параметру в онлайн режимі є відмінною особливістю пропонованої самонавчанняї нейро-системи.

Наступна низка експериментів була проведена на наборі даних «Іриси Фішера» (Fisher's Iris data set). Це багатовимірний датасет для задачі класифікації, на прикладі якого англійський статистик та біолог Рональд

Фішер в 1936 році продемонстрував роботу розробленого ним методу дискримінантного аналізу. Іноді його також називають «Ірисами Андерсона» (через те, що дані були зібрані американським ботаніком Едгаром Андерсоном).

Цей набір даних став класичним і часто використовується в літературі для ілюстрації роботи різних статистичних алгоритмів. Проте цей датасет рідко використовується у кластерному аналізі, адже межі класів «*Verginica*» та «*Versicolor*» не можна чітко визначити, ґрунтуючись на даних, що їх використовував Фішер (що легко продемонструвати за допомогою ієрархічного кластерування).

Саме цим і цікавий для нас цей набір даних: коли класичні методи чіткого кластерного аналізу не справляються з задачею, може стати у нагоді система, що реалізує нечітке кластерування зі змінним параметром фаззифікації та кількістю кластерів. Для більшості методів кластерного аналізу, зокрема для методу нечітких середніх (fuzzy c-means), необхідно заздалегідь задати кількість кластерів, і очевидним рішенням є прийняти $m = 3$, адже маємо три класи: *Iris Verginica*, *Iris Versicolor* та *Iris Setosa* (рисунок 4.6).

Дані по цьому датасету були попередньо відцентровані та пронормовані, а також було проведено ділення на навчальну, тестову та валідаційну підвибірki в співвідношенні 70 на 30.

Точність кластерування за допомогою методу нечітких середніх за таких умов ($m = 3$, $\beta = 2$) рідко перевищує 83% (таблиця 4.5). (Оскільки для обраного датасету існують мітки з вірною класифікацією, ефективність кластеризації вимірювалася у відсотках точності щодо еталонного розбиття після дефаззифікації.)

- virginica
- versicolor
- setosa

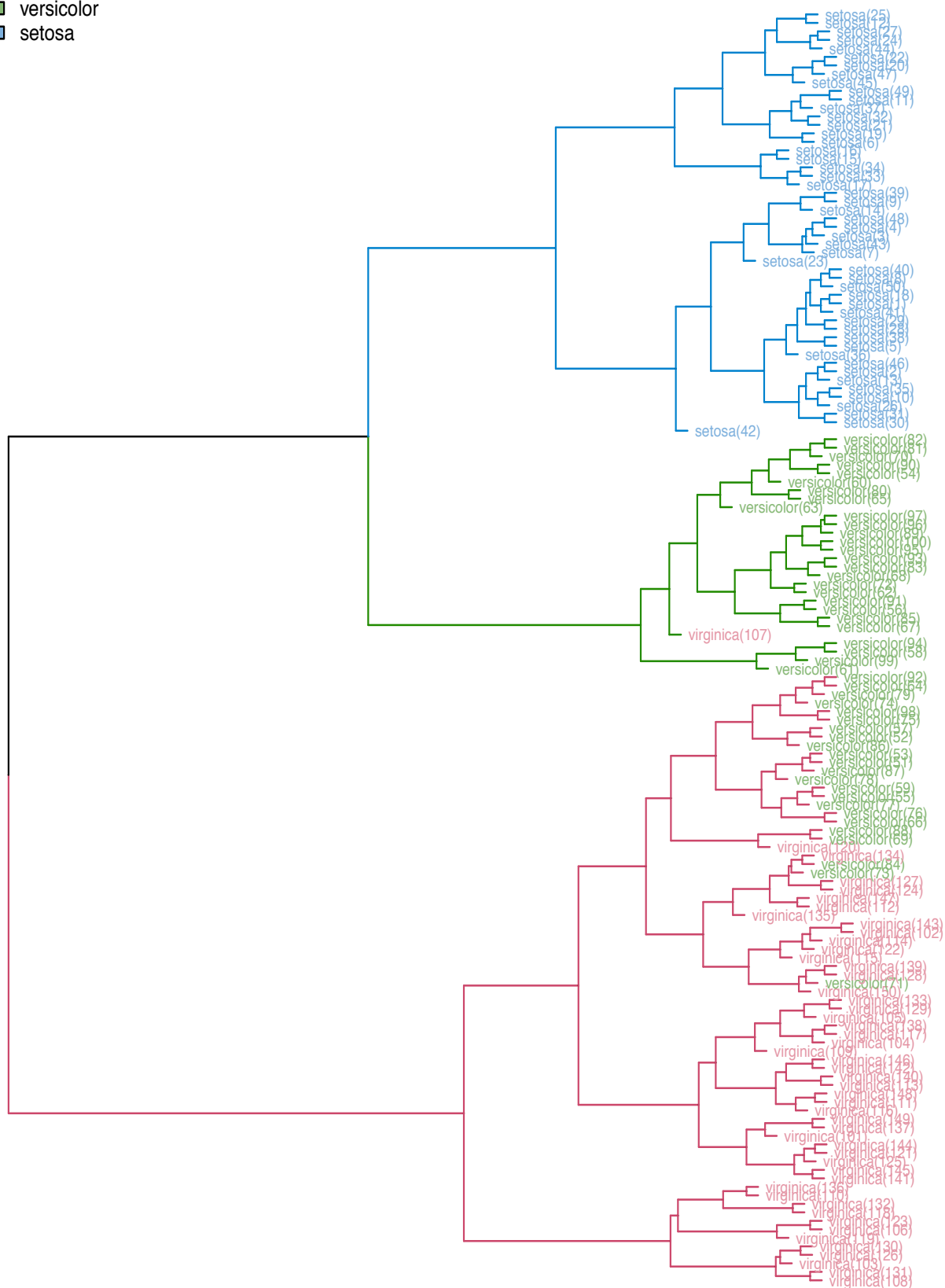


Рисунок 4.5 – Ієрархічне кластерування датасету «Іриси Фішера»

Проте, якщо не обмежувати пропоновану систему у кількості кластерів (система ініціалізується інтервалом допустимих значень m (кількість кластерів) та параметру фаззифікації β), вельми цікавими є результати кластерування нейронів кожного з каскадів, що наведені у таблиці 4.7.

Таблиця 4.6 – Точність кластерування датасету «Іриси Фішера»,

$m = 3$

Точність кластерування	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Середня	0.8313073	0.8741245	0.8709475	0.8888124
Точність кластерування	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Мінімальна	0.7859722	0.7533766	0.6615745	0.7656498
Максимальна	0.8534013	0.9166667	0.935051	0.9604701

Таблиця 4.7 – Точність кластерування датасету «Іриси Фішера» для

$m \in [7, 13]$, $\beta \in [2, 5]$

$m = 7$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Середня	0.8972948	0.9150268	0.9242503	0.9178207
Мінімальна	0.8536056	0.8461905	0.8723182	0.8600289
Максимальна	0.9621849	0.9736172	0.9810146	0.9663462
$m = 8$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Середня	0.9065560	0.9296311	0.9243606	0.9248976
Мінімальна	0.8217056	0.8562179	0.8577202	0.8590278
Максимальна	0.9474588	0.9789402	0.9848214	0.9747899
$m = 9$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Середня	0.9258154	0.9282887	0.9308971	0.9229753

Продовження таблиці 4.7

$m = 9$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Мінімальна	0.8689921	0.8270525	0.8684641	0.8556390
Максимальна	0.9849170	0.9806397	0.9664112	0.9748284
$m = 10$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Середня	0.9332528	0.9213191	0.9285106	0.9282907
Мінімальна	0.8663370	0.8722271	0.8652272	0.8766667
Максимальна	0.9663420	0.9838095	0.9723656	0.9756335
$m = 11$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Середня	0.9295315	0.9408977	0.9317242	0.9295800
Мінімальна	0.8520268	0.8964924	0.8890781	0.8788656
Максимальна	0.9716166	0.9848485	0.9704892	0.9798627
$m = 12$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Середня	0.9349407	0.9433244	0.9337934	0.9306632
Мінімальна	0.8815133	0.8949802	0.8798160	0.8486111
Максимальна	0.9795274	0.9783497	0.9630952	0.9772727
$m = 13$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Середня	0.9420998	0.9398127	0.9375204	0.9357708
Мінімальна	0.8823175	0.8614025	0.8882479	0.8828348
Максимальна	0.9807518	0.9788034	0.9753452	0.9748873

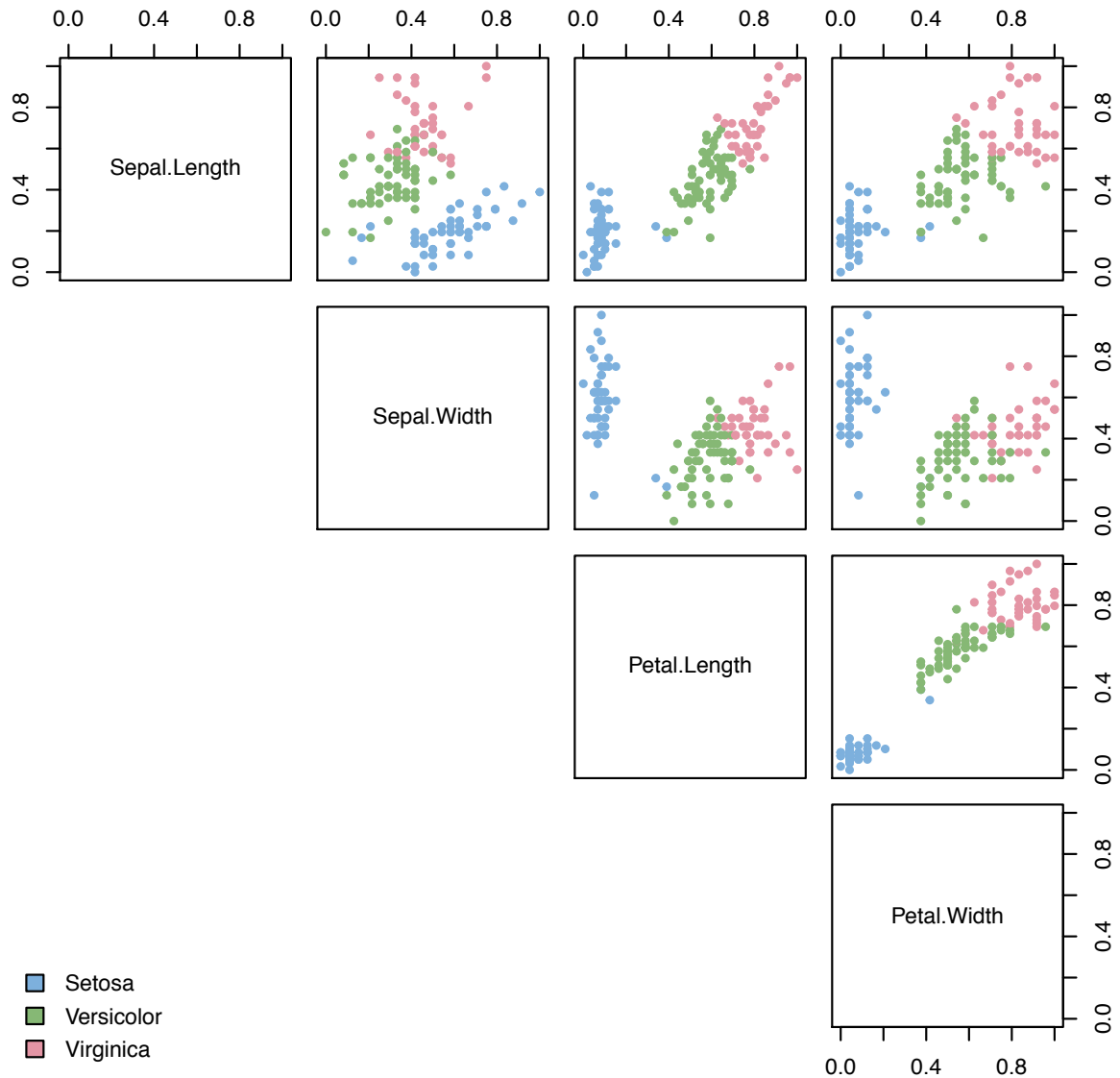


Рисунок 4.6 – Розкластерований датасет «Іриси Фішера» при $t = 3$, $\beta = 2$
(Точність– 96%)

Цікаво, що при, здавалося б, очевидному рішенні обрати кількість кластерів рівною трьом, отримуємо чи не найгіршу точність кластерування (при $\beta = 2$) після дефаззифікації щодо еталонного розбиття. Для порівняння на рисунку 4.7 та рисунку 4.8 наведені розбиття, що їх запропонували нейрони-переможці деяких каскадів, де $t \gg 3$. Цьому легко знайти пояснення, адже метод нечітких k -середніх (а саме цей метод у цьому експерименті реалізують вузли пулів кожного каскаду) добре розпізнає кластери лише гіперсферичної форми.

Проте кластер довільної (негіперсферичної) форми, можна розбити на декілька гіперсферичних підкластерів, що й відбувається у каскадах, де $m > 3$, гіперсферичних підкластерів, що й відбувається у каскадах, де $m > 3$, що пропонують розбиття на дрібні кластери.

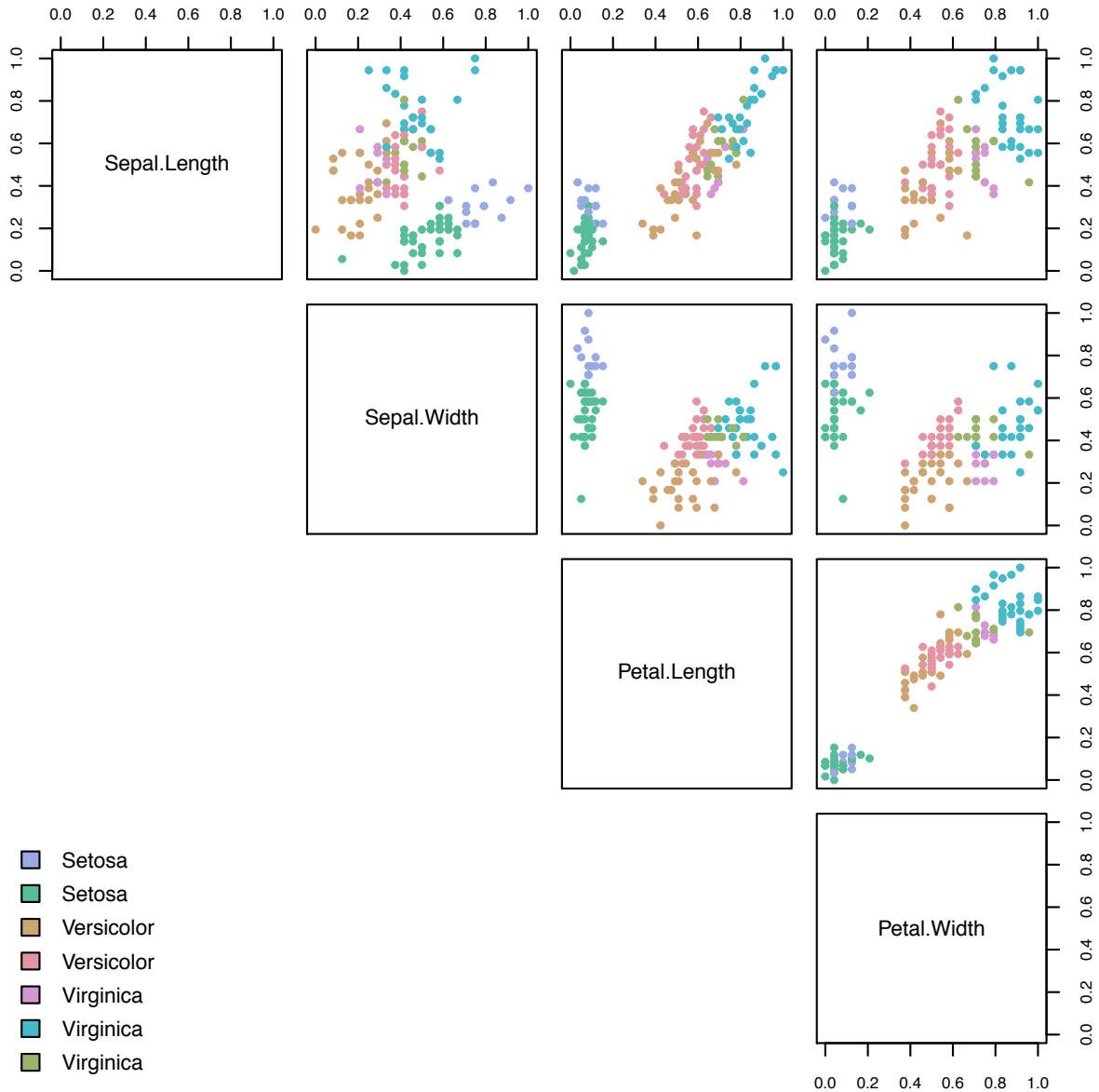


Рисунок 4.7 – Розкластерований датасет «Іриси Фішера» при $m = 7$, $\beta = 5$
(Точність кластерування $\approx 93\%$)

На рисунку 4.9 та рисунку 4.10 наведені розбиття деяких каскадів, де кількість кластерів більша від кількості класів еталонної вибірки; тут

можна побачити, що декілька кластерів, що після дефаззифікації будуть віднесені до одного класу.

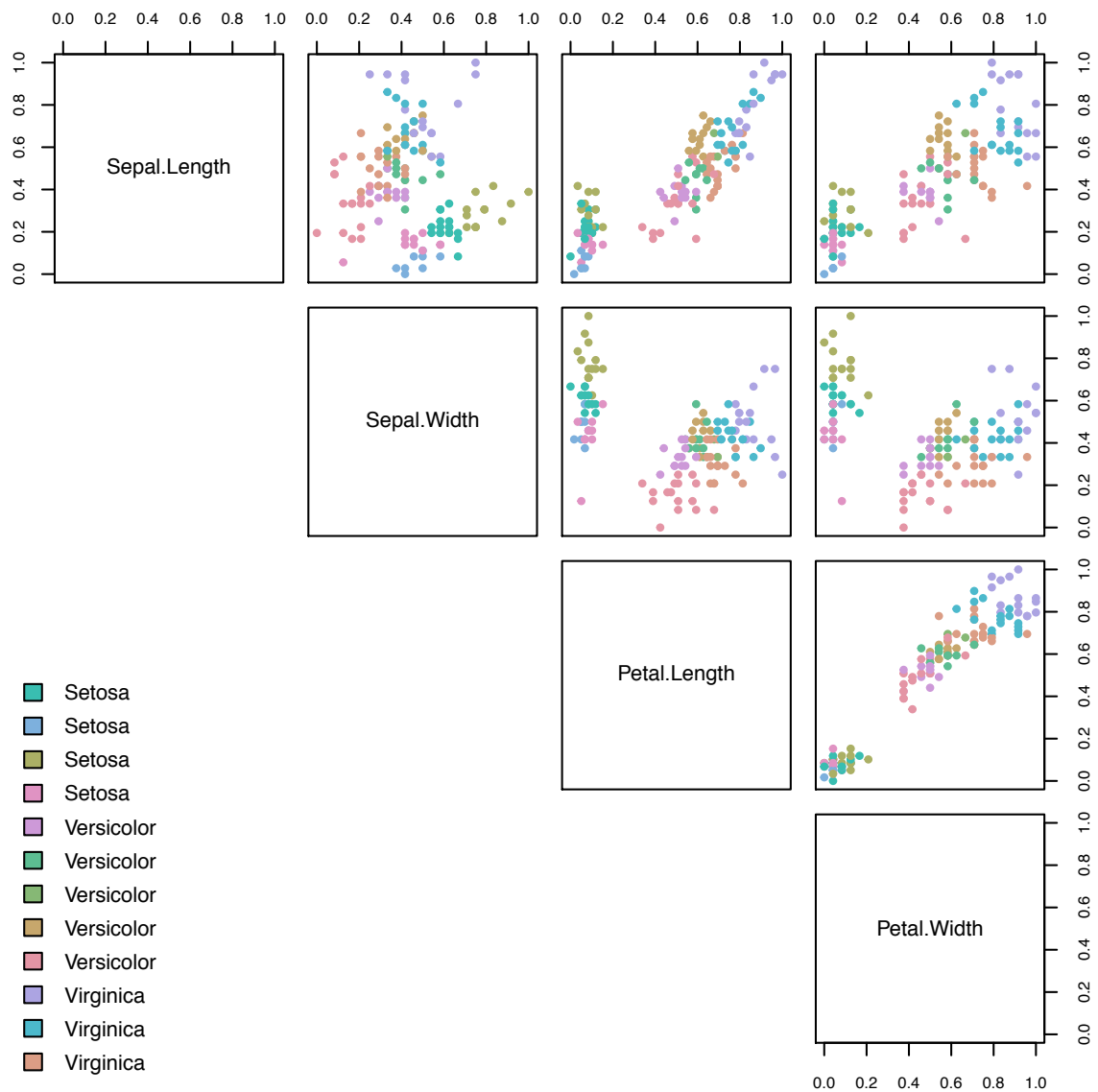


Рисунок 4.8 – Розкластерований датасет «Іриси Фішера» при $m = 12$,
 $\beta = 4$ (Точність кластерування $\approx 96\%$)

Найкращі результати були отримані нейроном 13-ого каскаду, що розбиває вхідний датасет на 14 кластерів. Точність кластерування нейронів цього каскаду наведена у таблиці 4.8.

Таблиця 4.8 – Точність кластерування датасету «Іриси Фішера»,
 $m = 14$

Точність кластерування	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Середня	0.9369168	0.9448829	0.9383179	0.9403416
Точність кластерування	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
Мінімальна	0.8847819	0.8953380	0.8787879	0.9069805
Максимальна	0.9731262	0.9754579	0.9918301	0.9762515

Наприклад, *Iris Virginica*, розташовані поруч один з одним, тобто є складовими більшого кластеру негіперсферичної форми.

На рисунку 4.11 зображено залежність точності кластерування від кількості кластерів. Таким чином, видається доречним навіть у випадку, коли відоме еталонне розбиття датасету, дозволити системі обрати кінцеву кількість кластерів самостійно, а особливо, коли вузли системи реалізують однаковий метод кластерування. Варто зауважити, що у цьому випадку для визначення локально оптимального розбиття доцільно використовувати модифіковані індекси валідності, чи такі, що не залежать від відстані центрів кластерів, наприклад ті, що ґрунтуються на щільності (density-based).

Наступну серію експериментів було проведено на датасеті «Знання студентів про електричні машини постійного струму». Цей датасет було додано до UCI репозиторію у 2013 році, він містить 403 патерни, кожен з п'ятьма атрибутами:

- STG: кількість часу, що його витратив(витратила) студент(ка) на вивчання цільового матеріалу;
- SCG: кількість повторювань вивчання цільового матеріалу студентом(студенткою);
- STR: кількість часу, що його використав(використала) студент(ка) на вивчання матеріалу, пов'язаного з цільовим матеріалом;

- LPR: оцінка, що її отримав(отримала) студент(ка) на іспиті з предмету, пов'язаного з цільовим предметом;
- PEG: оцінка, що її отримав(отримала) студент(ка) на іспиті з цільового предмету.

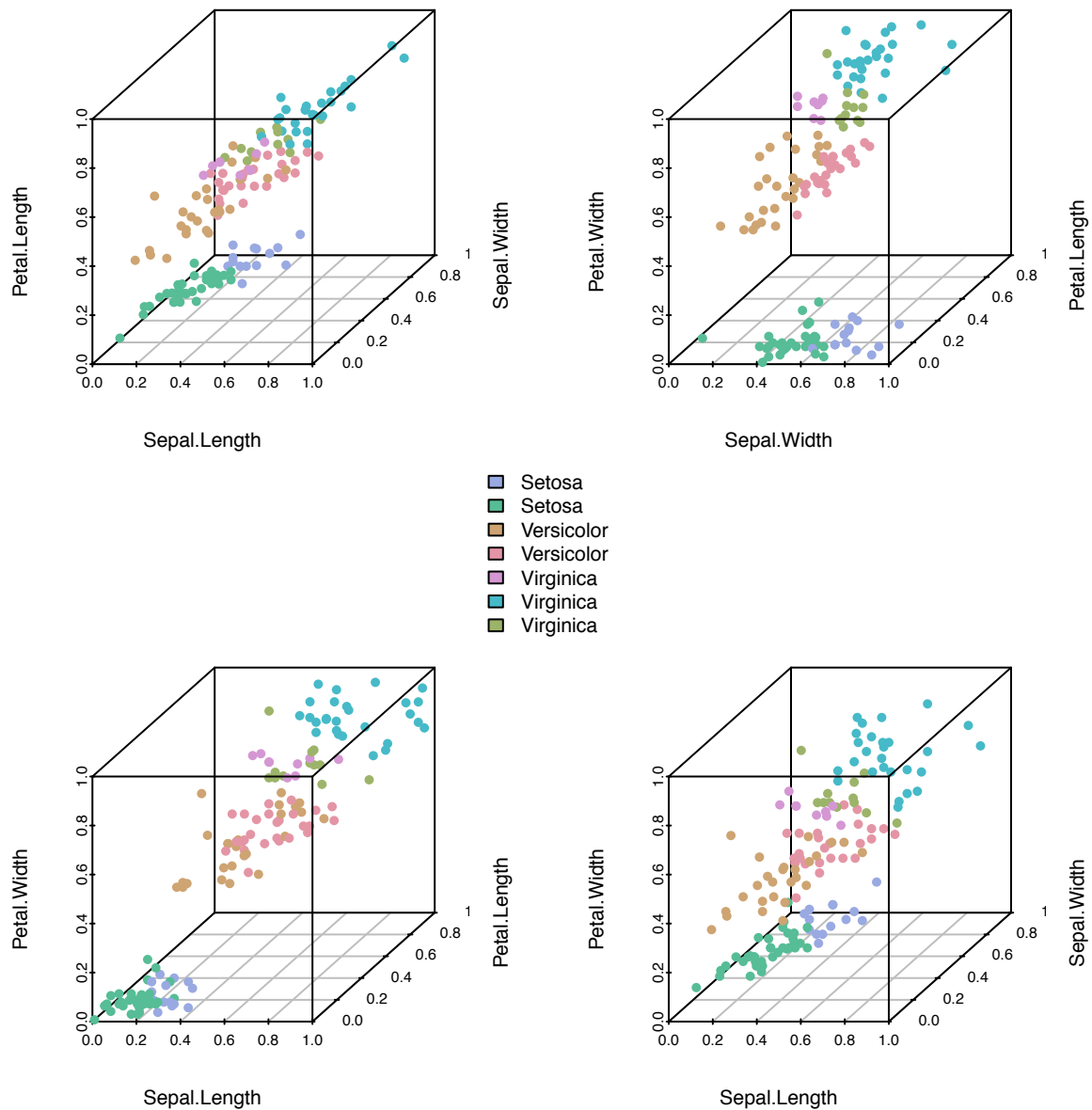


Рисунок 4.9 – Розкластерований датасет «Іриси Фішера» при $m = 7$, $\beta = 5$
(Точність кластерування $\approx 93\%$)

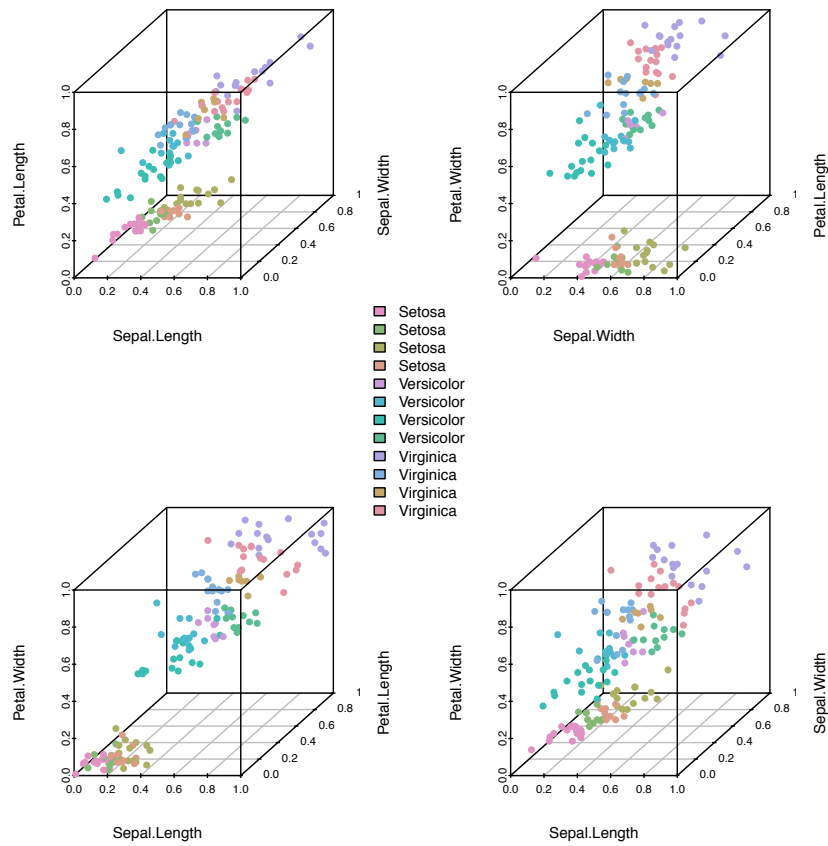


Рисунок 4.10 – Розкластерований датасет «Іриси Фішера» при $m = 12$, $\beta = 4$ (Точність кластерування $\approx 96\%$)

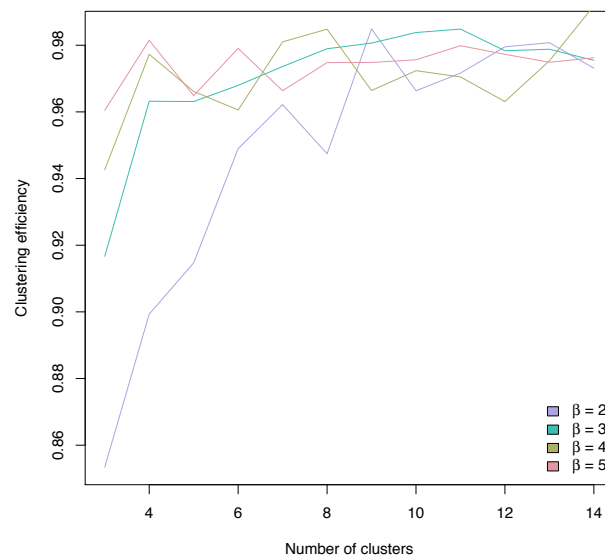


Рисунок 4.11 – Точність кластерування датасету «Іриси Фішера» від кількості кластерів та параметру фазифікації

Попарні графіки атрибутів наведено на рисунку 4.12 та у тримірному просторі на рисунку 4.13.

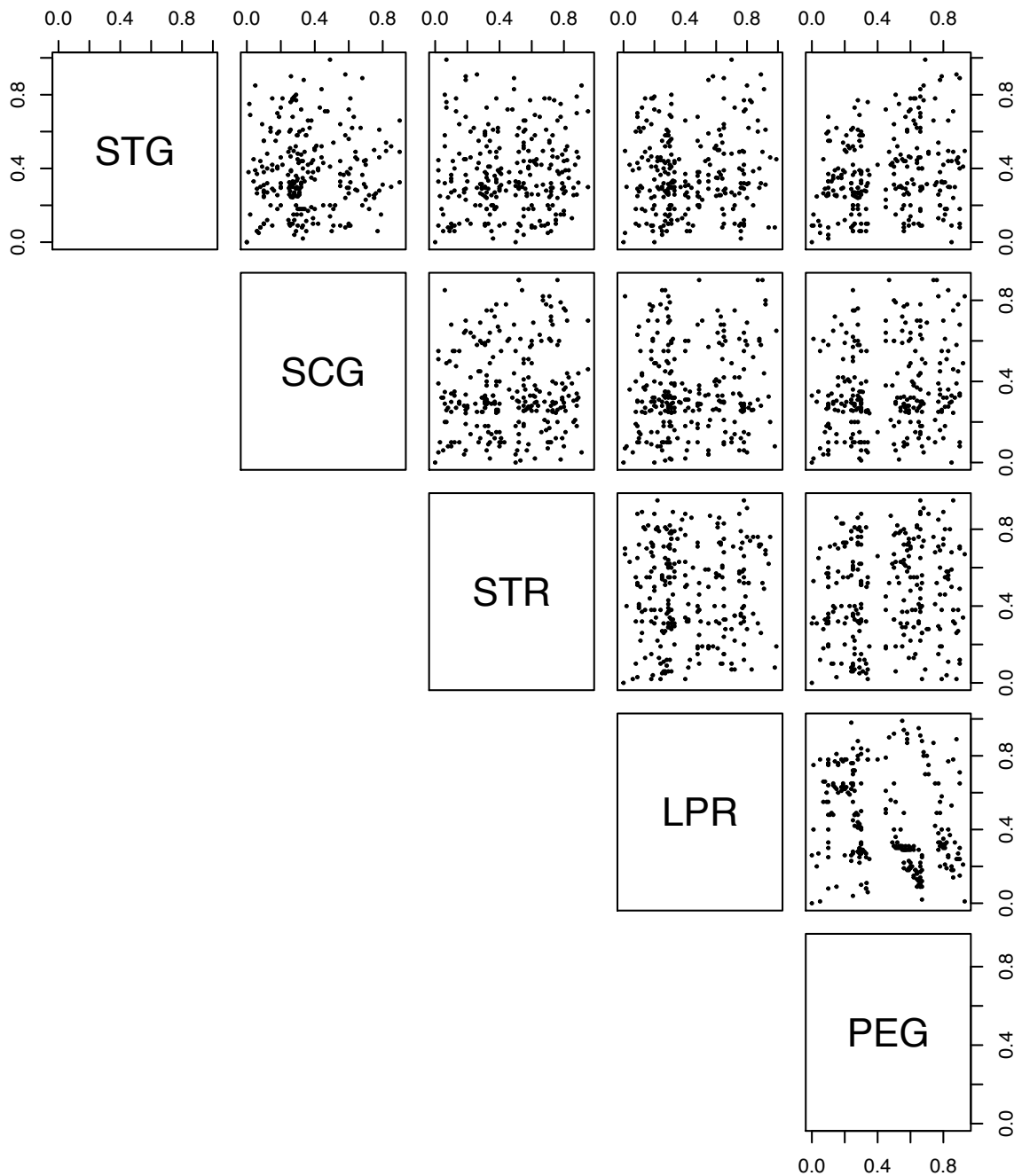


Рисунок 4.12 – Датасет «Знання студентів про електричні машини постійного струму»

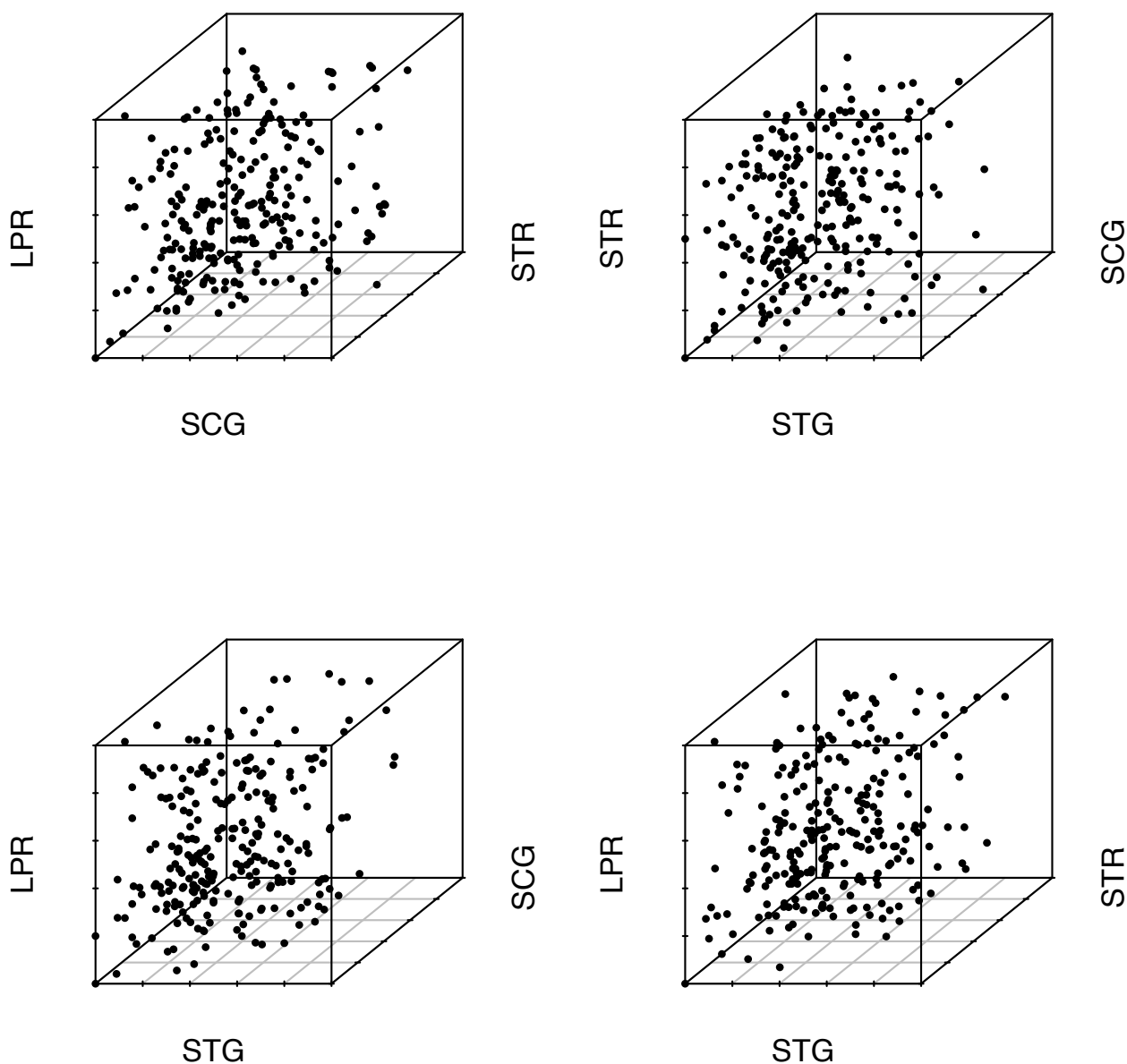


Рисунок 4.13 – Датасет «Знання студентів про електричні машини постійного струму»

Для цього експерименту нейрони-узагальнювачі керувалися рекурентним Ксі-Бені Індексом при визначанні локально-оптимального нейрона (з найліпшим параметром фаззифікації) та каскаду (з оптимальною кількістю кластерів).

Оптимальне розбиття, що його наведено на рисунок 4.14, надав другий нейрон третього каскаду з $m = 4, \beta = 2$ з коефіцієнтом Ксі-Бені $XB_2^{[3]} = 0.38155$.

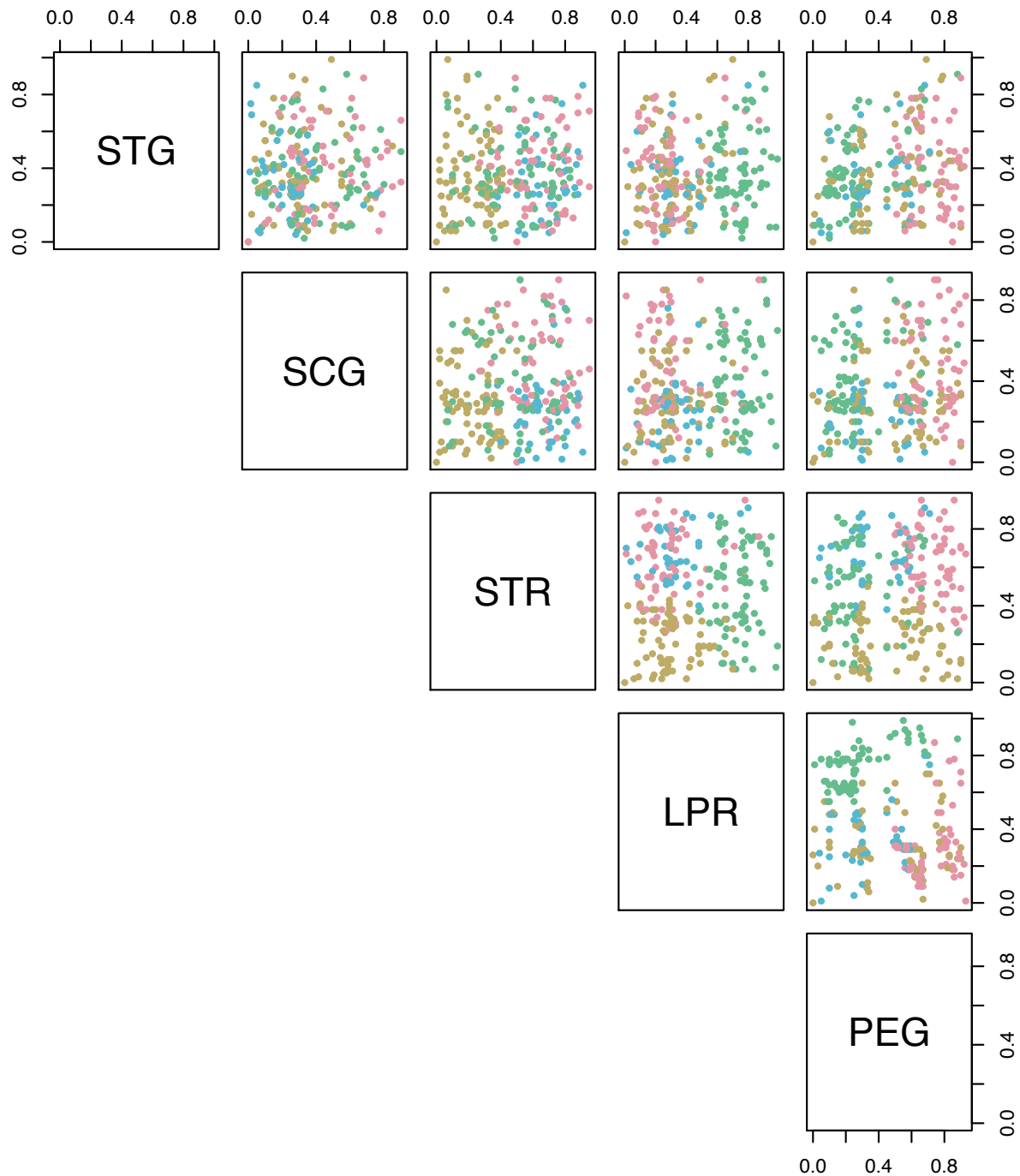


Рисунок 4.14 – Розкластерований датасет «Знання студентів про електричні машини постійного струму», $m = 4, \beta = 2$

Результати кластерування системи у тривимірному просторі наведено на рисунку 4.15.

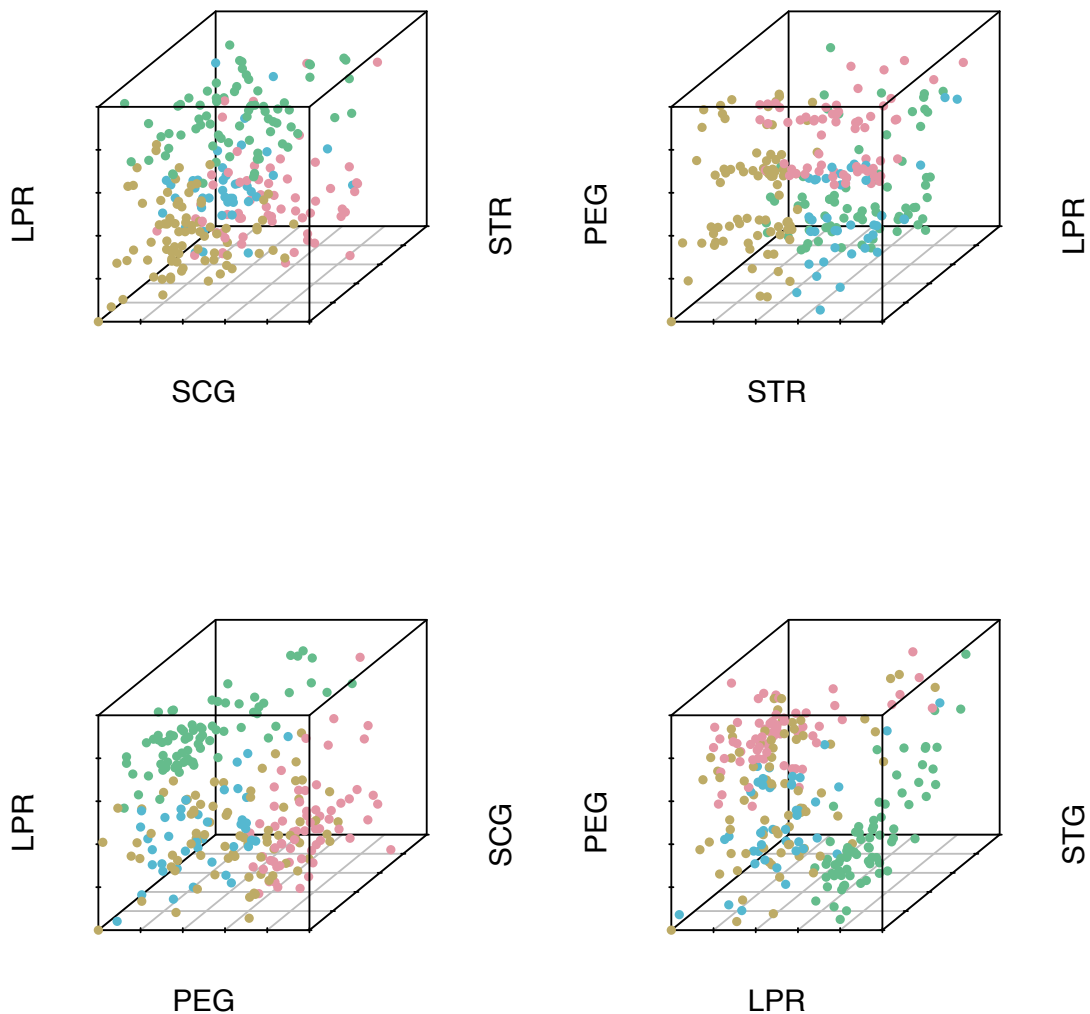


Рисунок 4.15 – Розкластерований датасет «Знання студентів про електричні машини постійного струму», $m = 4, \beta = 2$

В рамках імітаційного моделювання було розглянуто завдання нечіткого кластерування у режимі послідовного надходження даних до системи. Розв'язано низку практичних задач за умови невизначеності щодо кількості кластерів та рівня їх розмитості за допомогою запропонованої самонавчання гібридної системи.

ВИСНОВКИ

В роботі представлено результати, що відповідають меті дослідження, а саме – розробці методу для послідовного нечіткого кластерування потоків даних на основі ансамблевого підходу. Проведені дослідження дозволили зробити такі висновки:

1. Розглянуто завдання нечіткого кластерування у режимі послідовного надходження даних до системи.

2. Запропоновано метод визначення локально оптимальної кількості кластерів і значення параметру фаззифікації 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 кластерування 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 відшукування 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 фаззифікатору, і оптимальну кількість кластерів на кожному етапі оброблення даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Bodyanskiy Ye., Pliss I., Tyschenko O., Kopaliani D. A cascade neuro-fuzzy system for high-dimensional data clustering in a sequential mode. *Advances in Data Science. Proc. International Workshop and Networking Event, Poland, Hołny Mejera, May 6-8, Bialystok: BUT, 2015. P.11.*
2. Bodyanskiy Ye., Tyschenko O., Kopaliani D. An evolving neuro-fuzzy system for online fuzzy clustering. *Computer Science & Information Technologies. Proc. International Conference, Ukraine, Lviv, September 14-17, 2015. Lviv Polytechnic National University, 2015. P. 158-161.*
3. Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети: архитектуры, обучение, применения. Харьков: ТЕЛТЕХ, 2004. 372 с.
4. Haykin S. *Neural Networks. A Comprehensive Foundation.* Upper Saddle River, NJ: Prentice Hall, Inc., 1999. 842 p.
5. Zadeh, L. Fuzzy sets. *Information and Control.* 1965. Vol. 8. P.338-353.
6. Zadeh, L. Fuzzy logic – a personal perspective. *Fuzzy Sets and Systems.* 2015. Vol. 281. P. 4-20.
7. Dumitras A., Moschytz G. Understanding Fuzzy Logic: An Interview with Lotf i Zadeh [DSP History]. *IEEE Signal Processing Magazine.* 2007. 24(3). P.102-105.
8. Jang J.-S. R., Sun G.-T., Mizutani E. *Neuro-Fuzzy and Soft Computing.* Upper Saddle River, NJ: Prentice Hall, 1997. 614 p.
9. Chin-Teng Lin A neural fuzzy control system with structure and parameter learning. *Fuzzy Sets and Systems.* 1995. Vol. 70 (2-3). P. 183-212.
10. Chin-Teng Lin, George Lee C. S. *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems.* Prentice Hall, 1996. 797 p.
11. Shie-Jue Lee, Chen-Sen Ouyang A neuro-fuzzy system modeling with self-constructing rule generation and hybrid SVD-based lerning. *IEEE Transactions on Fuzzy Systems.* 2003. Vol. 11 (3). P. 341-353.

24. Cordon O., Gomide F., Herrera F., Hoffmann F., Magdalena L. Ten years of genetic fuzzy systems: current framework and new trends. *Fuzzy Sets and Systems*. 2004. Vol. 141 (1). P. 5-31.
12. Angelov P., Filev D. P. An approach to online identification of takagi-sugeno fuzzy models. *Systems, Man, and Cybernetics, Part B: Cybernetics*. 2004. Vol. 34(1). P. 484-498.
13. Angelov P., Lughofer E. Data-driven evolving fuzzy systems using eTS and FLEXFIS: comparative analysis. *Int. J. General Systems*. 2008. Vol. 37(1). P. 45-67.
14. Bezdek J. C. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press. New York. 1981.
15. Hoepfner F., Klawonn F., Kruse R., Runkler T. Fuzzy Clustering Analysis: Methods for Classification, Data Analysis and Image Recognition. Chichester: John Willey & Sons. 1999. 289 p.
16. Pedrycz W., Oliveira J. S. Advances in Fuzzy Clustering and its Applications. Chichester: John Wiley and Sons. 2008. 434 p.
17. Xu R., Wunsch D. Survey of clustering algorithms. *IEEE Trans on Neural Networks*. 2005. No3 (16). P. 645-678.
18. Abonyi J., Feil B. Cluster Analysis for Data Mining and System Identification. Basel: Birkhaeuser. 2007. 303p.
19. Dai B.-R., Huang J.-W., Yeh M.-Y., Chen M.-S. Adaptive clustering for multiple evolving streams. *IEEE Trans. Knowl. Data Eng.* 2006. Vol. 18(9). P. 1166-1180.
20. Beringer J., Huller E. Online clustering of parallel data streams. *meier Data Knowl. Eng.* 2006. Vol. 58(2). P. 180-204.
21. Gan G., Ma C., Wu J. Data Clustering: Theory, Algorithms, and Applications. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2007. 466 p.
22. Han J., Kamber M. Data Mining: Concepts and Techniques. Amsterdam: Morgan Kaufman Publ. 2006. 743p.

23. Kohonen T. *Self-Organizing Maps*.– Berlin: Springer-Verlag, 1995. 501p.
24. Park D., Dagher I. Gradient based fuzzy c-means (GBFCM) algorithm Proc. IEEE Int. Conf. on Neural Networks. 1984. P.1626-1631.
25. Vuorimaa P. Fuzzy self-organizing maps. *Fuzzy Sets and Systems*. 1994. Vol. 66. P.223-231.
26. Vuorimaa P. Use of the fuzzy self-organizing map in pattern recognition. Proc. 3-rd IEEE Int. Conf. Fuzzy Systems “FUZZ-IEEE’94”. Orlando. USA. 1994. P.798-801.
27. Chung F. L., Lee T. Fuzzy competitive learning. *Neural Networks*. 1994. No3. P.539-552.
28. Tsao E. C.-K., Bezdek J. C., Pal N. R. Fuzzy Kohonen clustering networks. *Pattern Recognition*. 1994. Vol. 27. P. 757-764.
29. Айзерман М.А., Браверман Э.М., Розоноэр Л.И. Метод потенциальных функций в теории обучения машин. М.: Наука, 1970. 384 с.
30. Parzen E. On the estimation of a probability density function and the mode. *Ann. Math. Statist.* 1962. №38. P. 1065–1076.
31. Надарая Э.А. О непараметрических оценках плотности вероятности и регрессии. *Теория вероятностей и ее применение*. 1965. 10. № 1. С. 199–203.
32. Живоглядов В.П., Медведев А.В. Непараметрические алгоритмы адаптации. Фрунзе: Илим, 1974. 214 с.
33. Медведев А.В. Адаптация в условиях непараметрической неопределенности. *Адаптивные системы и их приложения*. Новосибирск: Наука, 1978. С. 4–34.
34. Раудис Ш.Ю. Оптимизация непараметрического алгоритма классификации. *Адаптивные системы и их приложения*. Новосибирск: Наука, 1978. С. 57–61.
35. Poggio T., Girosi F., Networks for approximation and learning, Proc. IEEE, vol. 78, no. 9, pp. 1481-1497, Sep. 1990.