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)

Використання методів адаптивного нечіткого кластерування
для вирішення завдання опрацювання матричних даних
(тема)

Виконав:
студент 2 курсу, групи СШІм-20-3
Анісімов В. Е.
(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 проф., каф. ШІ, Кулішова Н. Є.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2022 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 Анісімову Владиславу Едуардовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Використання методів адаптивного нечіткого кластерування для вирішення завдання опрацювання матричних даних

затверджена наказом університету від 24 березня 20 22 р. № 414Ст

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

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

4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної області та постановка задачі дослідження, способи вирішення задачі розпізнавання образів, методи для кластерування пакетів даних, методи нечіткого кластерування, ЕМ-алгоритм, можливісний метод кластерування (PCM), матрична модифікація методу кластерування нечітких С-середніх, матричні модифікації можливісного алгоритму С-середніх та комбінацій можливістного та нечіткого алгоритму С-середніх, адаптивний матричний алгоритм кластерування нечітких С-середніх, імітаційне моделювання

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – два непересічних класи, Рисунок 2 – Змінення функції належності відносно від значення параметру m, Рисунок 3 – Колоколоподібна функція, Рисунок 4 – Цифрове зображення, що надходить на опрацювання, Рисунок 5 – Зображення перетворено до Grayscale моделі, Рисунок 6 – Візуалізація кластерування набору даних Iris з фінальним розташуванням координат центроїдів кластерів, Рисунок 7 – Візуалізація кластерування набору даних Wine з фінальним розташуванням координат центроїдів кластерів, Рисунок 8 – Візуалізація кластерування набору даних Wine з фінальним розташуванням координат центроїдів кластерів, Рисунок 9 – Візуалізація кластерування набору даних Dermatology з фінальним розташуванням координат центроїдів кластерів, Рисунок 10 – Зображення після кластерування, Рисунок 11 – Виходове зображення після кластерування адаптивним методом нечітких С-середніх, Рисунок 12 – Результати кластерування набору даних Mall_Customers адаптивним нечітким алгоритмом С-середніх

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Отримання завдання на кваліфікаційне проектування	28.03.2022	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	12.05.2022	виконано
9.	Представлення на рецензування	13.05.2022	виконано
10.	Представлення кваліфікаційної роботи		

Дата видачі завдання 28 березня 20 22 р.

Студент _____
(підпис)

Керівник роботи _____ _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 77 с., 22 рис., 2 табл., 53 формул, 2 дод., 35 джерел.

АДАПТИВНА КЛАСТЕРИЗАЦІЯ, АЛГОРИТМ С-СЕРЕДНІХ, АНАЛІЗ ДАНИХ, КЛАСТЕРНИЙ АНАЛІЗ, МАТРИЦЯ, МАТРИЧНИЙ МЕТОД, МОЖЛИВІСНА КЛАСТЕРИЗАЦІЯ, НАВЧАННЯ БЕЗ ВЧИТЕЛЯ, НЕЧІТКА КЛАСТЕРИЗАЦІЯ.

Об’єкт дослідження є нечітке кластерування даних, представлених у багатовимірному просторі.

Предмет дослідження: алгоритми, підходи та методи кластерування даних високої розмірності.

Метою роботи є розробка пакетних та адаптивних матричних алгоритмів кластерування, які розширюють метод нечітких С-середніх, а також розробка програмного забезпечення, що реалізує запропоновані модифіковані алгоритми.

Методологічна основа кваліфікаційної роботи полягає в застосуванні методів аналізу, класифікації, узагальнення та опису досліджених методів нечіткого кластерування, а також у постановці експерименту на основі розроблених модифікацій алгоритму нечітких С-середніх.

У ході виконання кваліфікаційної роботи розроблені матричні модифікації алгоритму нечітких С-середніх і програмне забезпечення, що реалізує їх. Введені методи дозволяють працювати безпосередньо з матричними даними, уникаючи громіздких операцій векторизації-девекторизації, покращуючи час кластерування при однаковій якості з алгоритмом, що модифікується. Запропоновані методи застосовні для дослідження та обробки електромагнітних, теплових і оптичних полів вимірювань, областей забруднення повітряного басейну, медичних спостережень, цифрових зображень та відеорядів.

РЕФЕРАТ

Пояснительная записка: 77 с., 22 рис., 2 табл, 53 формул, 2 прил., 35 источников.

АДАПТИВНАЯ КЛАСТЕРИЗАЦИЯ, АЛГОРИТМ С-СРЕДНИХ, АНАЛИЗ ДАННЫХ, ВОЗМОЖНОСТНАЯ КЛАСТЕРИЗАЦИЯ, КЛАСТЕРНЫЙ АНАЛИЗ, МАТРИЦА, МАТРИЧНЫЙ МЕТОД, НЕЧЁТКАЯ КЛАСТЕРИЗАЦИЯ, ОБУЧЕНИЕ БЕЗ УЧИТЕЛЯ.

Объект исследования: нечёткая кластеризация данных, представленных в многомерном пространстве.

Предмет исследования: алгоритмы, подходы и методы кластеризации данных высокой размерности.

Целью работы является разработка пакетных и адаптивных матричных алгоритмов кластеризации, которые расширяют метод нечётких с-средних, а также разработка программного обеспечения, реализующего предложенные модифицированные алгоритмы.

Методологическая основа магистерской квалификационной работы состояла в применении методов анализа, классификации, обобщения и описания исследованных методов нечёткой кластеризации, а также в постановке эксперимента на основе разработанных модификаций алгоритма нечётких с-средних.

Предложенные методы применимы для исследования и обработки электромагнитных, тепловых и оптических полей измерений, областей загрязнения воздушного бассейна, медицинских наблюдений, цифровых изображений и видеорядов.

ABSTRACT

Explanatory note: 77 p., 22 fig., 2 tabl., 53 formulas, 2 ann., 35 sources.

ADAPTIVE CLUSTERING, CLUSTER ANALYSIS, C-MEANS ALGORITHM, DATA MINING, FUZZY CLUSTERING, MATRIX, MATRIX METHOD, POSSIBILISTIC CLUSTERING, UNSUPERVISED LEARNING.

The object of research: fuzzy clustering of high-dimensional data. Subject of research: algorithms, approaches and methods of high-dimensional data clustering.

The objective of research is to develop a batch and adaptive matrix clustering algorithms, which extend the fuzzy c-means method, and development of software that implements the proposed modified algorithms.

The methodological basis of Master Thesis was to use methods of analysis, classification, synthesis and description of the investigated methods of fuzzy clustering, as well as in the experiment on the basis of the developed modifications of fuzzy c-means algorithm.

In the course of Master Thesis matrix modifications of fuzzy c-means algorithm and software implementing them were developed. The introduced methods allow working directly with the matrix data, avoiding the bulky operations of vectoring and an opposite operation of vectoring, and improving the time of clustering (the quality of the developed and modifiable algorithms is the same). The proposed methods are applicable to research and processing of electromagnetic, thermal and optical fields, areas of air pollution, medical observations, digital images and video sequences.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

FCM – Fuzzy C-means – Нечіткі С-Середні;

FPCM – Fuzzy possibilistic C-means – Нечіткі імовірні С-Середні;

PCM – Possibilistic C-means – Можливісні С-Середні;

PFCM – Possibilistic fuzzy C-means – Можливісні нечіткі С-Середні.

ВСТУП

З різким розвитком інформаційних технологій у другій половині XX століття у світі інформації почала значно збільшуватися кількість даних. З'явилася потреба зберігати та обробляти великі обсяги інформації. Особливо значущою ця проблема стала на початку XXI століття у зв'язку з різким зростанням кількості користувачів мережі Інтернет, і на даний момент вона залишається не менш актуальною.

Кластерний аналіз є важливим завданням в межах створення систем штучного інтелекту, оскільки оцінка результатів кластерування не може бути об'єктивною, і різні методи дають різні результати для тих самих даних. Також, крім великої кількості існуючих проблем, з'являються нові, що вимагають нових методів рішень або вдосконалення старих. Наприклад, такі проблеми можуть стосуватися представлення даних у матричній, а не векторній формі, або великої розмірності входового вектору даних, а також необхідності обробляти дані в послідовному режимі, коли спостереження на опрацювання надходять одне за одним і їх кінцева кількість апріорі невідома.

Для вирішення задачі кластерування крім класичних методів та алгоритмів машинного навчання можуть застосовуватись штучні нейронні мережі. Штучна нейронна мережа є концептуальною моделлю біологічної нейронної мережі і складається з пов'язаних між собою різним чином шарів штучних нейронів, які організовують загальну активну структуру і функціонально впливають на роботу один одного. У більшості архітектур штучних нейромереж активність нейрона визначається перетворенням зовнішнього сумарного впливу інших нейронів на даний нейрон.

З моменту свого зародження технології штучних нейронних мереж розвивалися досить відокремлено від класичних методів, нерідко докорінно змінюючи уявлення про предмет і проблематику теорії машинного навчання і розпізнавання об’єктів, залишаючи значний вплив на теоретичний,

термінологічний і методологічний апарати цих дисциплін. Через деякий час після розвитку базових моделей штучних нейронних мереж, відбувся значний поділ науки про нейромережі на види топологій архітектури мереж і методи навчання мереж.

У більшості архітектур штучних нейронних мереж функції активації нейронів фіксовані, а ваги синапсів є параметрами мережі. Деякі входи нейронів є зовнішніми входами сукупної мережі, а деякі виходи нейронів – виходами сукупної мережі. Завдання нейромережі полягає в перетворенні вхідного вектора у вихідний вектор, що здійснюється вагою і топологією мережі.

Ця кваліфікаційна робота присвячена розробці матричних нечітких алгоритмів кластеруваннгя в пакетному та online режимах, що є модифікаціями методу нечітких С-середніх.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Теорія розпізнавання образів існує як розділ інформатики та суміжних дисциплін, що розвиває методи класифікації та ідентифікації об'єктів різної природи: сигналів, ситуацій, предметів, що характеризуються вичерпною кількістю деяких ознак [1]. Проблема розпізнавання об’єктів також виділена в розділ міждисциплінарних досліджень – в тому числі включаючи роботу зі створення штучного інтелекту, а також часто використовується при вирішенні практичних завдань у галузі комп'ютерного зору.

При постановці класичної задачі розпізнавання об’єктів заведено застосовувати математичну мову, ґрунтуючись на логічних міркуваннях і математичних принципах. В протилежність до цього підходу, існують методи розпізнавання об’єктів з використанням машинного навчання і штучних нейронних мереж, сформовані на не настільки формалізованих підходах до розпізнавання, і демонструють не гірший, а в деяких випадках і значно кращий результат.

Часто можна зустріти хибне зіствлення термінів «розпізнавання» та «класифікація», де вони розглядаються як синоніми, але не є повністю взаємозамінюваними. Кожен з цих двох термінів може мати свої сфери застосування, в залежності від поставленої задачі. Розглянемо загальні елементи моделі класифікації.

Клас – множина об'єктів, що мають спільні властивості. Для об'єктів одного класу передбачається наявність «схожості». Для задачі розпізнавання може бути визначено довільну кількість класів, більше одного. Кількість класів позначається числом S. Кожен клас має свою ідентифікуючу мітку класу [1].

Класифікація – процес призначення міток класу об'єктам, відповідно до певного опису властивостей цих об'єктів. Класифікатор – це інструмент для присвоєння міток класам, який в якості вхідних даних отримує перелік

ознак об'єкта. До одного з найпоширеніших способів класифікації можна віднести спосіб, що базується на описі об'єктів з використанням ознак, де кожен об'єкт характеризується набором числових або нечислових ознак. Проте існують типи даних, для яких відкриті ознаки не дають високої точності класифікації, наприклад, колір точок зображень або цифровий звуковий сигнал. Загальна класифікація зображень собак і автомобілів є дуже простою для людини і водночас складною для машини. Причиною цього є можливість людини сприймати «приховані ознаки», недоступні для машини, такі як морда собаки або колеса автомобіля [1].

Верифікація – процес зіставлення досліджуваного об'єкта із однією моделлю об'єкта або описом класу [1]. Під образом будемо розуміти найменування області в просторі ознак, в якій відображається безліч об'єктів або явищ матеріального світу. Ознакою можна назвати опис тієї чи іншої властивості, що має пряме відношення до предмета або явища.

Простір ознак – це N-вимірний простір, визначений для даної задачі розпізнавання, де N – фіксоване число ознак, що були виміряні для будь–яких об'єктів. Вектор з простору ознак x, відповідний об'єкту задачі розпізнавання, це N-вимірний вектор з компонентами $(x_1, x_2, \dots, x_N)$, що утворюють значення ознак для даного об'єкта [1].

Іншими словами, розпізнавання образів можна визначити, як віднесення вихідних даних до певного класу за допомогою виділення істотних ознак або властивостей, які характеризують ці дані, із загальної маси несуттєвих деталей.

Прикладами завдань класифікації є розпізнавання символів, встановлення медичного діагнозу, прогноз погоди, розпізнавання осіб, класифікація документів, тощо.

Найчастіше вихідним матеріалом служить отримане з камери зображення. Постановку завдання можна сформулювати, як одержання векторів, що складаються з ознак для кожного класу на зображенні. Процес

можна розглядати як процес кодування, що полягає в присвоєнні значення кожної ознаки із простору ознак для кожного класу.

Якщо розглянути 2 класи об'єктів: дорослі і діти, в якості значення ознак можна вибрати зріст і вагу (рисунок 1.1).

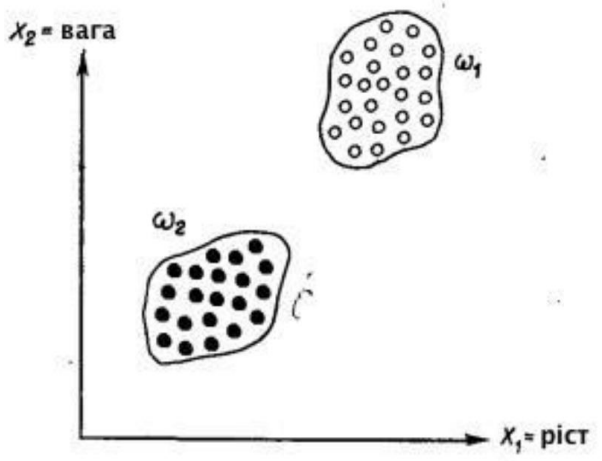

Рисунок 1.1 – Два непересічних класи

На рисунку 1.1 видно, що ці два класи утворюють дві непересічні множини, це можна пояснити обраними ознаками. Проте не завжди є можливість вибрати правильні вимірювані параметри в якості ознак класів. Наприклад, вибрані параметри не підійдуть, щоб створити непересічні класи хокеїстів та волейболістів.

Іншим завданням розпізнавання є виділення із вихідних зображень характерних ознак та/або властивостей. Це завдання можна віднести до попередньої обробки. Ознака повинна являти собою характерну властивість конкретного класу, при цьому загальну для цього класу. Міжкласові ознаки – це ознаки, що визначають відмінності між класами. Загальні ознаки, що властиві усім класам, не несуть корисної інформації, тому для задачі розпізнавання об’єктів не розглядаються як характерні. Вибрати правильні ознаки – одна із важливих задач побудови систем розпізнавання.

1.1 Способи вирішення задач розпізнавання образів

Комп'ютерний зір, у свою чергу, розвивається як теорія і технологія створення машин, які можуть виявляти, відстежувати та класифікувати об'єкти. Як наукова дисципліна, комп'ютерний зір тісно пов'язаний з машинним навчанням та теорією розпізнавання образів, але відноситься до більш спеціалізованої технології створення штучних систем, які отримують інформацію та оперують інформацією з зображень. Автоматичне планування або прийняття рішень на основі підсистем комп'ютерного зору так само займає важливу частину в області штучного інтелекту, оскільки автономні системи досить складного рівня організації, які виконують деякі механічні дії (наприклад, переміщення робота), потребують високорівневих даних, що представляють інформацію про середовища, в яких вони функціонують.

Класичні методи комп'ютерного зору, розпізнавання об’єктів і машинного навчання можна умовно розділити на три групи: методи фільтрації, методи логічного аналізу, методи навчання.

1.2 Методи фільтрації

У завданнях комп'ютерного зору фільтрація найчастіше використовується для попередньої обробки зображення перед аналізом його внутрішніх морфологічних ознак, але зустрічаються і завдання, в яких достатнім і бажаним буде використання тільки фільтрації (наприклад, в задачах машинного зору).

Бінаризація по порогу [2], вибір області гістограми. Для RGB зображень і зображень в градаціях сірого порогом є значення кольору. Встановлення порогу, за яким і відбудеться бінаризація, багато в чому визначає процес бінаризації. Зазвичай бінаризація здійснюється за допомогою адаптивного алгоритму, що вибирає поріг. Таким алгоритмом

може бути вибір математичного очікування, моди або піків гістограми. При роботі з гістограмою бінаризація ефективна для сегментації кольорів.

Перетворення Фур'є майже не використовується при обробці зображень в чистому вигляді, оскільки для аналізу зображень одновимірного перетворення зазвичай не вистачає і виникає необхідність використання куди більш ресурсного двовимірного перетворення. Цей метод застосовується тільки в разі, якщо необхідний аналіз спектра, оскільки використання згортки цікавить області з уже готовим фільтром. Проте, одномірне перетворення Фур'є застосовується при компресії зображень.

Фільтри частот. Найпростіший приклад фільтра низьких частот – фільтр Гаусса, фільтра високих частот – фільтр Габора. Для кожної точки зображення вибирається вікно, в рамках якого виконується множення вихідних даних з фільтром того ж розміру (згортка). Такий підхід отримав досить широке поширення в практиці, дозволяючи виділяти на зображенні необхідну інформацію, відсіюючи зайву інформацію. Зокрема, підхід поширений як одна з реалізацій швидкого шумозаглушення в багатьох областях науки і техніки [2].

Вейвлет-перетворення [2]. Для згортки з сигналом (областю зображення) використовується деяка характеристична функція, названа вейвлетом, що визначається як вейвлет-перетворення. Вейвлети – це сімейства функцій, локальних за часом і по частоті, в яких всі функції виходять з однієї з функцій, за допомогою її змін положення і розтягування по осі, що відображає час. Існує набір класичних функцій, залучених в вейвлет-аналізі. До них відносяться вейвлет Хаара, вейвлет Морлі, вейвлет MHat, вейвлет Добеши. На практиці вейвлет-аналізом називається пошук довільного патерну на зображенні за допомогою згортки зображення з моделлю цього патерну. Класичні вейвлети використовуються для стиснення або класифікації зображень.

Метод обчислення кореляції [2], що лежить в основі вейвлет–перетворення, сам по собі є незамінним інструментом в системах комп'ютерного зору і часто використовується в своєму природному вигляді, наприклад, для знаходження зрушень або оптичних потоків (кореляція відеопотоку). На основі обрахованої корелятором різниці реалізується найпростіший детектор зсуву.

Метод фільтрів функцій [2]. У цьому підході використовуються математичні фільтри, що дозволяють виявляти прості математичні функції на зображенні, для чого формується акумулююче зображення (накопичувальний простір), в якому для кожної точки вихідного зображення будується множина породжуючих її функцій. Класичним прикладом є узагальнене перетворення Хафа, що застосовується до бінаризованих зображень, що дозволяє знаходити функції на зображенні. Модифіковане перетворення Хафа дозволяє шукати будь-які фігури, але в обробці зображень його використання пов'язане з недостатньою стабільністю: висока чутливість до якості бинаризації і низька швидкість роботи змушують шукати більш ефективні методи.

1.3 Методи логічного аналізу

За допомогою фільтрації можна отримати набір даних, що буде придатний для обробки, але більшість завдань галузі комп'ютерного зору вимагають аналізу внутрішньої структури зображення і морфологічних ознак зображених об'єктів, для цього досліджуються і впроваджуються методи логічного аналізу зображень.

Методи математичної морфології є результатом теорії та техніки аналізу й обробки геометричних структур, заснованих на теорії множин, топології та випадкових функціях. Ці методи реалізуються базовими операціями. Операція двійкової морфології є деяким перетворенням впорядкованої множини або підмножини (області зображення) за

допомогою структурного елементу. Структурним елементом є двійкове зображення довільного розміру та довільної структури, але найчастіше використовуються симетричні елементи, такі як прямокутник фіксованого розміру або коло фіксованого діаметру. Результатом перетворення також є двійкове зображення [2].

Методи математичної морфології дозволяють видаляти шуми з двійкових зображень, а також реалізують алгоритми пошуку контурів, але на практиці ці методи використовуються в поєднанні з іншими алгоритмами.

Контурний аналіз – це потужний математичний апарат, що дозволяє описувати, зберігати і знаходити об'єкти, які знаходяться в формі зовнішніх контурів. Вище розглядалися фільтри контурів, результатом застосування яких природним чином є контури об'єктів на зображенні без застосування додаткової бінаризації. У контурному аналізі попередні стадії фільтрації і додаткової бінаризації є обов'язковими етапами завдання.

Передбачається, що контур містить необхідну інформацію про форму об'єкта, а внутрішні точки до уваги не приймаються, що обмежує область застосування алгоритмів контурного аналізу. Проте отримані з його допомогою контури дозволяють перейти від двовимірного простору образу до простору контурів, в деяких завданнях це значно зменшує складність алгоритму. Методи контурного аналізу інваріантні щодо перенесення, повороту і масштабування зображення об'єкта. Серед методів контурного аналізу можна виділити алгоритми, в яких контур об'єкта відстежується і векторизується; скануючі алгоритми, засновані на перегляді всього зображення і виділення контурних точок без відстеження; алгоритми відстеження контурів на напівтонових зображеннях, дослідження кривизни функцій.

На практиці методи контурного аналізу досить чутливі до умов середовища, що може викликати складність їх використання в реальних умовах для більшості завдань комп'ютерного зору, проте вони корисні в

задачах машинного зору, коли умови середовища досить строго визначені. У таких ситуаціях методи контурного аналізу є неперевершеними лідерами по швидкодії, володіючи зрозумілою і простою логікою, що обумовлює зручність їх використання в спеціалізованих областях.

Пошук особливих точок (feature detection) [2], на цій темі варто зупинитися докладніше. Пошук особливих точок є одним з найбільш поширених методів в класичному комп'ютерному зорі. Особливі точки надають унікальні характеристики об'єкта, дозволяючи порівнювати різні зображення одного об'єкта або одного класу об'єктів між собою. Тому на практиці особливі точки найбільш актуальні в задачах, в яких можливим рішенням є обробка серії зображень або відео потоку і подальший аналіз отриманих масивів особливих точок. Алгоритми feature detection можна умовно класифікувати за ступенем стабільності точок при переходах від одного зображення (кадру) об'єкта або класу об'єктів до іншого. Складність алгоритмів пошуку зростає з необхідним рівнем стабільності шуканих точок.

Результатом роботи таких алгоритмів є множина особливих точок, зокрема, кутів, для яких необхідно побудувати математичний опис. Формування математичного опису – це завдання дескриптора. Багато дескрипторів одночасно вирішують завдання пошуку особливих точок і побудови описів цих точок за допомогою вбудованих алгоритмів або оригінальним власним способом. Ознаки (описи) будуються на основі інформації про інтенсивність, про кольори та текстуру особливих точок.

1.4 Застосування штучних нейронних мереж при вирішенні задачі кластерування

Мета задачі кластерування полягає у розбитті сукупності об'єктів на кластери на основі певної міри. Не існує «кращого» критерію для всіх видів кластерування. Отже, критерій кластерування повинен вибиратися окремо

для вирішення кожного конкретного випадку, таким чином, щоб результат об'єднання в кластери задовольнив потреби користувача.

Кластерний аналіз займає одне з центральних місць серед методів аналізу даних і є сукупністю підходів, методів і алгоритмів, призначених для знаходження деякого розбиття досліджуваної сукупності об'єктів на підмножини відносно подібних, схожих між собою об'єктів. При цьому вихідним припущенням для виділення таких підмножин, що отримали спеціальну назву кластер, які також іноді називають таксонами або просто класами, служить лише неформальне припущення про те, що об'єкти, що відносяться до одного кластеру, повинні мати більшу схожість між собою, ніж з об'єктами з інших кластерів [2].

Виявлення або знаходження кластерів у наборі даних досліджуваної сукупності має відповідати наступним вимогам:

– кожен кластер має бути концептуально однорідною категорією і містити схожі об'єкти з близькими значеннями властивостей або ознак;

– сукупність всіх кластерів має бути вичерпною, тобто охоплювати всі об'єкти досліджуваної сукупності;

– кластери не повинні перетинатись між собою, тобто жоден з об'єктів досліджуваної сукупності не повинен одночасно належати двом різним кластерам.

Формально під завданням кластерного аналізу заданої множини об'єктів розуміють задачу знаходження деякого розбиття цієї множини об'єктів на підмножини, що не перетинаються, таким чином, щоб елементи, що відносяться до однієї підмножини, відрізнялися між собою значно меншою мірою, ніж елементи з різних підмножин.

Методи кластерування на основі штучних нейронних мереж є різновидом класичних методів кластерування.

Методи кластерування на основі мереж Т. Кохонена (ці мережі ще мають назву самоорганізовні мапи Т. Кохонена) включають найпопулярніший метод k-середніх, але в послідовному його варіанті.

Нейронні мережі Кохонена, або самоорганізовні мапи Кохонена (Kohonen's Self-Organizing Maps), призначені для вирішення завдань класифікації без вчителя (самонавчання) [3]. Це двошарова нейронна мережа, що містить вхідний шар (шар входових нейронів) та шар Кохонена (шар функцій активації).

Шар Кохонена може бути одновимірним, двовимірним або тривимірним. У першому випадку активні нейрони розташовані в ланцюжок. У другому випадку вони утворюють двовимірну сітку (зазвичай у формі квадрата або прямокутника), а у третьому випадку вони утворюють тривимірну конструкцію.

Завдяки тому, що це – мережі, які самонавчаються, для їх навчання непотрібна навчальна послідовность образів, кожному з яких від вчителя відома належність до того чи іншого класу, визначення ваг нейронів шару Кохонена грунтується на використанні алгоритмів кластерування для яких непотрібно знати вірні мітки кластерування, система (нейронна мережа) повинна на основі певних внутрішніх закономірностей «зрозуміти», до якого класу належіть то чи інше спостереження.

Нейронні мережі мають перевагу при роботі із зашумленими даними та великими (надвеликими) наборами даних, що мають надмірну розмірність.

1.5 Метод для кластерування пакетів даних К-середніх

Метод К-середніх розбиває n спостережень на $k \leq n$ кластерів, при цьому кожне спостереження належить тому кластеру, до центру якого воно найближче. Термін «К-середніх» вперше з'явився у роботі [4]. Метод простий у реалізації, але водночас потребує великих обчислювальних ресурсів. Також цей метод використовує ітеративний підхід уточнення, а кластерні центри використовуються для моделювання даних. Метод К-

середніх намагається знайти кластери, що не перетинаються між собою, тобто є лінійно роздільними, а також мають опуклу форму.

Нехай задана множина спостережень $X = (x_1, \dots, x_n)$, де $x_i \in R^d$, $i = 1, \dots, n$. Необхідно розділити множину спостережень X на k кластерів, що не перетинаються між собою і мають опуклу форму $S_1, \dots, S_k$, $S_i \cap S_j = \emptyset$, $i \neq j$, $\bigcup_{i=1}^{k} S_i = X$, таким чином, щоб мінімізувати суму квадратів відстаней від кожної точки кластера до його центроїду (центр мас кластеру), що по факту відповідає пошуку

$$\underset{s}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2, \tag{1.1}$$

де μ_i – центри кластерів S_i, $i = 1, \dots, k$, $\|x - \mu_i\|^2 = \sum_{x \neq y \in S_i} (x - \mu_i)(x - \mu_i)$.

Процедура методу К-середніх складається з послідовності наступних кроків. Випадковим чином обираються центри кластерів $\mu_1^{(1)}$,…, $\mu_k^{(1)}$, далі виконується циклічна послідовність двох етапів:

1. кожне спостереження x_p належить до того кластеру, центроїд котрого найближчий до спостереження, тобто як зазначено формулою:

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2, \text{для будь} - \text{якого } j = 1, \dots, k \right\} \tag{1.2}$$

де x_p належіть виключно до одного $S_i^{(t)}$, навіть якщо його можна віднести до двох або більше кластерів;

2. переобчислення центроїдів кластера для тих спостережень, що вже належать до різних кластерів:

$$m_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{x \in S_i^{(t)}} x_j. \quad (1.3)$$

Метод зупиняється, коли $m_i^{(t)} = m_i^{(t+1)}$ для будь-якого i.

Необхідно зауважити, що кількість кластерів k необхідно знати заздалегідь. Невірний вибір k може призвести до поганих та нерелевантних результатів рішення задачі. Тому, перед тим, як безпосередньо починати моделювання методу К-середніх, важливо виконати попередню аналітику і виявити кількість кластерів, використовуючи методи статистичного візуального аналізу даних.

Також до недоліків цього методу можна віднести чутливість до початкових значень координат центроїдів майбутніх кластерів. Як вже було зазначено вище, метод К-середніх є дуже чутливий до аномалій, викидів та шумів різної фізичної природи, завдяки тому, що при розрахунках центроїдів містить стандартну міру оцінки середньої тенденції – середнє арифметичне. Щоб покращити цей простий в імплементації метод, треба використовувати його модифікацію К-медоідів, який замість середнього арифметичного використовує медіану, яка є робастною оцінкою, тобто нечутлива до викидів та аномалій. Або необхідно звернути свою увагу на самоорганізовну мапу Кохонена, яка є послідовною модифікацією класичного методу К-середніх, але здатна опрацьовувати потоки спостережень. Але, якщо природа кластерів є такою, що вони апріорі не є лінійно роздільними, в такому випадку краще скористатись певним нечітким методом кластерування.

1.6 Метод нечіткої самоорганізації С-середніх

Метод нечіткої самоорганізації С-середніх (Fuzzy C-Means – FCM) це метод кластерування, в якому кожне зі спостережень n може належати

відразу кільком кластерам з різним рівнем належності. Таким чином, дані, розташовані на рецепторних полях кластерів, не повинні повністю належати одному кластеру, а можуть з певним рівнем належності відноситись до багатьох або всіх кластерів одразу. Рівень належності вимірюється в межах від 0 до 1, а сума всіх коефіцієнтів належності кожного окремого спостереження має дорівнювати 1.

В 1965 році Лотфі Заде [6] представив аксіоматичну структуру – нечітка множина. Нечітка множина була задумана, щоб розібратися з проблемою розпізнавання образів у контексті класів, що є лінійно нероздільними у початковому просторі ознак, а кожне спостереження з певною ймовірністю може належати до всіх класів одночасно. В 1969 році Е. Руспіні [6] опублікував статтю, яка стала основою більшості методів та систем для нечіткого кластерування. Він вперше застосував теорію нечітких множин до вирішення завдання розпізнавання лінійно нероздільних образів. Проте, лише після появи робіт Дж. Бездек [4] та Дж. Дамма [7] методи нечіткого кластерування стали важливою віхою в теорії розпізнавання образів, оскільки була чітко встановлена актуальність теорії нечітких множин для вирішення задачі розпізнавання образів як зі вчителем так і в режимі самонавчання. Метод нечітких С-середніх дуже схожий на метод К-середніх.

Нехай задана множина спостережень $X = (x_1, \dots, x_n)$, де $x_i \in R^d$, $i = 1, \dots, n$. Необхідно розділити множину спостережень X на c нечітких кластерів $(S_1, \dots, S_c)$ із центрами $(\beta_1, \dots, \beta_c) = \beta$ таким чином, щоб мінімізувати функції втрат, яка представлена виразом

$$arg\min_{(U,\beta)} \sum_{i=1}^{n} \sum_{j=1}^{c} w_{ij}^{m} \left\| x_i - \beta_j \right\|^2, \tag{1.4}$$

де $w_{ij} \in [0,1]$ – ступінь (рівень) належності елементу x_i кластеру S_j із центром β_j, який задовольняє обмеження

$$w_{ij} \in [0,1] \text{ для всіх } i, j, \tag{1.5}$$

$$0 < \sum_{j=1}^{n} w_{ij} < n \text{ для всіх } i, \tag{1.6}$$

$$\sum_{i}^{c} w_{ij} = 1 \text{ для всіх } j. \tag{1.7}$$

Число $m \in [1, +\infty)$ у функції (1.7) – експоненціальна вага, яка визначає «розмитість» границь рецепторних полів кластерів. Виходячи із необхідних умов локального екстремуму отримуємо співвідношення:

$$w_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - \beta_j\|}{\|x_i - \beta_k\|} \right)^{2/m-1}}, \quad i = 1, \dots, n, j = 1, \dots, c, \tag{1.8}$$

$$\beta_j = \frac{\sum_{i=1}^{n} w_{ij}^m x_i}{\sum_{i=1}^{n} w_{ij}^m}, j = 1, \dots, c. \tag{1.9}$$

Обмеження (1.7) узагальнює співвідношення, запропоноване спочатку Лотфі Заде [7], для визначення ступеня належності будь-якої точки x нечіткої множини S та її доповнення S'. А саме, доповнення S' до нечіткої множини S визначається рівнянням:

$$f_{S'} = 1 - f_s, \tag{1.10}$$

де $f_s: X \to [0,1]$ – характеристична функція нечіткої множини S, яка ставить у відповідність кожній точці з X дійсне число з відрізка [0, 1].

Через свою схожість з адитивним законом ймовірностей, співвідношення (1.6) часто називають ймовірнісним обмеженням. Однак закон (1.6) описує природу класифікованого набору даних, а не статистичне припущення про випадковий процес, що генерує набір даних.

Процедура методу нечіткої самоорганізації полягає в послідовному виконані наступних кроків. Випадковим чином необхідно згенерувати матрицю нечіткого розбиття $W^{(1)} = \left\{w_{ij}^{(1)}\right\}, \quad i = 1, \dots, n, \quad j = 1, \dots, c.$ Обчислити центри кластерів за формулою (1.8). Далі йде циклічне повторення кроків:

1. обчислити відстань від кожного спостереження x_i до центроїдів кластерів β_j, тобто $\|x_i - \beta_j\|$;

2. перерахувати елементи матриці нечіткого розбиття W за формулою (1.7);

3. перевстановити центроїди кластерів за формулою (1.7) відповідно до нових елементів матриці W із пункту 2;

4. порівняти $W^{(t+1)}$ та $W^{(t)}$, де t – номер ітерації. Якщо $\|W^{(t+1)} - W^{(t)}\| < \varepsilon$ (для встановленого апріорі ε), то метод зупиняється, у всіх інших випадках – повертаємось до першого кроку навчання.

Загальну кількість кластерів c, як і у випадку К-середніх, необхідно знати заздалегідь. Чим більша експоненціальна вага m, тим більш «розмитою» стає кінцева матриця нечіткого розбиття W. Якщо $m \to \infty$ елементи матриці W будуть дорівнювати $\frac{1}{c}$. Це буде показником того, що всі спостереження належать до всіх кластерів з однаковою ступеню $\frac{1}{c}$. Якщо $m \to 1$, елементи матриці W будуть збігатись до 0, або до 1, що є свідченням того, що розбиття спостережень по кластерам є чітким, а функція мінімізації буде співпадати з функцією мінімізації в методі К-середніх.

Крім того, експоненціальна вага m дозволяє при обчислені координат центроїдів кластерів посилювати вплив об’єктів з більшими значеннями ступенів (рівнів) належності і, відповідно, зменшувати вплив об’єктів з меншим значенням рівнів належності. На сьогоднішній день не існує теоретично обґрунтованого правила вибору експоненціальної ваги. Зазвичай її встановлюють $m = 2$. Не дивлячись на успіхи методу нечіткої самоорганізації С-середніх в більш природньому розділенні даних на кластери, проблема чутливості до спотворених викидами та аномаліями даних залишається актуальною.

Достатньо буде розглянути простий приклад. Припустимо, що є два кластери. Якщо спостереження x_k рівно віддалені від центроїдів обох кластерів, то в залежності від віддалення від цих центроїдів, їх ступені належності з умови (1.6) співпадають і будуть дорівнювати 0.5. Таким чином, аномальним точкам, які знаходяться далеко, але рівно віддалені від центрів двох кластерів, все-рівно може бути присвоєно мітку обох кластерів. Хоча більш природнім мало бути виключення цих точок із розгляду або присвоєння їм найменшого ступеню належності.

До недоліків методу нечіткої самоорганізації С-середніх також можна віднести чутливість до початково вибору показника ступеню належності w_{ij}, а також збіжність до локального екстремуму.

Форма кластерів у будь-якому алгоритмі кластерування визначається функцією, яка досліджується на мінімум, у якій у свою чергу бере участь відстань, що індукує топологічну метрику в R^d. Тому в методі нечіткої самоорганізації кластери можуть приймати форму, наближену до сферичної, як і у випадку методу К-середніх. Для того, щоб подолати це обмеження в межах нечітких методів кластерування існують декілька стратегій для розвитку систем нечіткої самоорганізації.

Перший напрям відноситься до методів, в основі яких лежить робота Д. Густафсона та В. Кесселя [8]. В цій роботі запропоновано замінити норму

відстані $\|x_i - \beta_j\|^2$ в функції, яка досліджується на мінімум, на альтернативну норму $\|x_i - \beta_j\|_{A_j}^2 = (x_i - \beta_j)^T A_j (x_i - \beta_j)$, де A_j – симетричні позитивно визначені матриці та $detA_j = \rho_j,\ \rho_j > 0$. Таким чином, кожен кластер приймає форму, яка закладена в матриці A_j.

Другий напрям, за яким розвиваються методи нечіткого кластерування, вперше було розглянуто в роботі [5], в якому нечіткі кластери мають форму лінії. З цієї роботи виріс цілий напрям різних методів нечіткої самоорганізації даних, в яких центри кластерів замінюються на більш загальні структури, типу ліній, площин, гіперкубів тощо.

1.7 Постановка задачі

На сьогоднішній день не існує алгоритмів кластерування, які б безпосередньо опрацьовували матричні об’єкти, що викликає необхідність використовувати операції векторизації-девекторизації в тих випадках, коли входові дані представлені у вигляді двовимірних масивів. Прикладами таких даних є електромагнітні, теплові та оптичні поля вимірювань, області забруднення повітряного басейну та водної поверхні, цифрові зображення та відеоряди.

Основною метою магістерської кваліфікаційної роботи є розробка модифікацій класичних методів кластерування для опрацювання наборів даних, які надходять на опрацювання як у векторному, так і в матричному вигляді, в умовах пакетної та послідовної обробки входової інформації.

Досягнення поставленої мети здійснюється шляхом вирішення наступних основних завдань:

– необхідно розробити та реалізувати набір матричних методів нечітких С-середніх (Matrix FCM, Matrix PCM, Matrix FPCM та Matrix PFCM), а також їх модифікації – адаптивні матричні методи нечітких С-середніх (Adaptive Matrix FCM, PCM, FPCM и PFCM);

– провести тестування створеної моделі для кластерування, провести аналіз переваг та недоліків, та зробити висновки щодо шляхів подолання виявлених недоліків;

– також необхідно провести порівняльний аналіз ефективності розроблених матричних алгоритмів з результатами векторних аналогів.

Для програмної реалізації розглянутих алгоритмів буде використовуватись мова програмування Python та основні бібліотеки для вирішення задач аналізу даних.

Основу для перевірки розробленої системи для кластерування складатиме тестові набори даних із репозиторієв машинного навчання та набір цифрових супутникових зображень міста Харкова.

Таким чином, в межах кваліфікаційної роботи магістра буде розроблено та протестовано пакетний алгоритм кластерування даних, які надходять на опрацювання у вигляді двовимірного масиву. Також буде запропонована модифікація для поширення цього підходу на випадок опрацювання даних, що подаються до системи у послідовному режимі.

Об'єкт дослідження – процес обробки даних, що надходять на опрацювання як у векторній, так і у матричній формі за допомогою адаптивного нечіткого алгоритму С-середніх.

Предмет дослідження – методи нечіткого кластерування, які використовують як ймовірністний, так і можливістний підходи для розділення даних нелінійної природи.

Методи дослідження – теорія обчислювального інтелекту і м'яких обчислень; теорія штучних нейронних мереж; теорія нечіткої логіки; теорія оптимізації і статистичний аналіз; імітаційне моделювання.

Розроблені в роботі методи дозволять уникнути громіздких операцій векторизації-девекторизації входових вибірки і, таким чином, зменшити час кластерування. Провадження методів кластерування на основі матричних процедур дозволяє полегшити загальну роботу систем, що використовуються для вирішення завдань розпізнавання образів. Матричні

алгоритми, які будуть синтезовані в кваліфікаційній роботі магістра, на відміну від глибинних нейронних мереж для свого навчання не потребують великої кількості спостережень в навчальній множині та при «тонкому» налаштуванні треба налаштовувати значно меншу кількість параметрів. Ще однією великою перевагою саме матричних алгоритмів кластерування є те, що вони навчаються значно швидше ніж класичні глибинні мережі, які зараз використовуються при опрацюванні зображень.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

Кластерний аналіз – це сукупність методів, алгоритмів та систем для розділення даних на класи об’єктів, в кожному з яких об’єкти мають схожі ознаки. В загальному поданні кластерний аналіз – це не один загальний метод, а спільна задача, яку необхідно вирішувати за допомогою різних методів та систем. Не існує об'єктивно «правильного» алгоритму кластерування. Найбільш підходящий алгоритм кластерування необхідно обирати експериментально, в залежності від набору даних, якщо тільки не існує математичної причини віддати перевагу одному алгоритму перед іншим.

Методи кластерування можна розділити за принципом опрацювання даних, за способом аналізу даних, за масштабованістю, за часом виконання тощо. Методи за способом аналізу даних, в своєю чергою, поділяють на чіткі (чи традиційні) і нечіткі. До чітких методів належать методи та системи, у яких кожен об'єкт даних належить одному кластеру. До нечітких систем кластерування відносяться ті, в яких кожен об'єкт із входового набору даних належить більш ніж до одного кластера з певним ступенем належності.

2.1 EM-алгоритм

EM-алгоритм – це ітеративний метод знаходження оцінок максимальної правдоподібності параметрів статистичної моделі, коли модель залежить від прихованих закономірностей, що знаходяться у змінних. Кожна ітерація цього методу складається із послідовності двох кроків. На етапі очікування (expectation) обчислюється очікуване значення функції правдоподібності з використанням поточної оцінки параметрів. На етапі максимізації (maximization) розраховуються параметри, які максимізують очікувану функцію максимальної правдоподібності, знайдену

на етапі очікування. Вперше така назва методу з'явилась в роботі [7], але подібна процедура розглядалась значно раніше багатьма авторами, і наприклад, приводиться у роботі А. МакКендріка та М. И. Шлезингера [3].

Припустимо, що є X та Y – випадкові змінні, які приймають значення в просторах R^n і R^m відповідно, де $n, m \geq 1$. Нехай θ – параметр із деякої множини Θ довільної природи. Щільність спільного розподілу $(n+m)$-вимірного випадкового вектору (X, Y) позначимо як

$$f_\theta(x, y), x \in R^n, y \in R^m, \theta \in \Theta. \tag{2.1}$$

Умовна щільність випадкової змінної Y при умові, що $X = x$ визначається, як показано у формулі (2.1)

$$f_\theta(x|y) = \frac{f_\theta(x, y)}{f_\theta^X(x)}, \quad y \in R^m, \tag{2.1}$$

де

$$f_\theta^X(x) = \int_{R^m} f_\theta(x, y)\mu_Y(dy) \tag{2.2}$$

є маргінальною щільністю випадкової змінної X відносно міри μ_Y.

Вираз (2.2) має сенс, якщо $f_\theta^X(x) \neq 0$. Аналогічно визначається умовна щільність випадкової змінної X за умовою $Y = y$ (формула 2.3):

$$f_\theta(x|y) = \frac{f_\theta(x, y)}{f_\theta^Y(x)}, \qquad x \in R^n. \tag{2.3}$$

Вираз (2.3) має сенс, якщо маргінальна щільність випадкової змінної Y відносно міри μ_X (формула 2.4)

$$f_\theta^Y(y) = \int_{R^n} f_\theta(x,y)\mu_X(dx). \tag{2.4}$$

В якості значень μ_Y та μ_X можна використовувати міру Лебега, або іншу міру, яка може бути формальним еквівалентом кількості елементів множини. Відносно до співвідношень (2.2) та (2.3) йде наступне:

$$f_\theta(x,y) = f_\theta(x|y)f_\theta^X(x) = f_\theta(x|y)f_\theta^Y(y). \tag{2.5}$$

Випадкова змінна X розглядається як спостереження, що досліджуються, в той час як прихована (змінна, яка не спостерігається) випадкова змінна Y грає роль допоміжної. Знаючи спільну щільність $f_\theta(x,y)$ та значення x змінної X, що спостерігається, можна формально визначити загальну функцію правдоподібності

$$L(\theta;x,y) = f_\theta(x,y), \qquad \theta \in \Theta, \tag{2.6}$$

як функцію параметру θ.

При цьому

$$L(\theta;x) = f_\theta^X(x), \qquad \theta \in \Theta \tag{2.7}$$

функція правдоподібності параметру θ за неповних даних.

Загальна мета EM-алгоритму – знайти значення параметру θ, які максимізують функції (2.6) або (2.7) за невідомим значенням Y або, іншими словами, необхідно знайти оцінки максимальної правдоподібності

параметру θ. Процедура ЕМ-алгоритму складається із обчислення послідовності значень $\{\theta^{(m)}\}$, $m \geq 1$ параметру θ. Якщо задано деяке значення $\theta^{(m)}$, то необхідно обчислити наступне значення $\theta^{(m+1)}$. Цю процедуру можна розділити на два окремих етапи:

1. (Е-етап) необхідно визначити функцію $Q(\theta, \theta^{(m)})$, як умовне математичне очікування логарифму повної функції правдоподібності при відомому значенні компоненти X, що спостерігається:

$$Q(\theta, \theta^{(m)}) = E_{\theta^{(m)}}(logf_{\theta}(X,Y)|X) \tag{2.8}$$

де θ – аргумент шуканої функції;

$\theta^{(m)}$ та X – параметри функції.

За відомих значень $X = x$ символ $E_{\theta^{(m)}}$ являє собою середнє значення випадкової змінної Y відносно умовного розподілення $f_{\theta^{(m)}}(x|y)$, тобто:

$$Q(\theta, \theta^{(m)}) = \int_{R^m}^{\cdot} (logf_{\theta}(X,Y)) f_{\theta^{(m)}}(x|y)\mu_Y(dy). \tag{2.9}$$

2. (М-етап) на цьому етапі обчислюється

$$\theta^{(m+1)} = \text{argmix}_{\theta} Q(\theta, \theta^{(m)}). \tag{2.10}$$

Далі обирається метрика $\rho(\cdot,\cdot)$ та фіксується мале позитивне значення ε. Ітераційний процес зупиняється на m-ом кроці, якщо $\rho(\theta^{(m)}, \theta^{(m+1)}) < \varepsilon$.

Також необхідно відзначити монотонність ЕМ-алгоритму, ця властивість вперше була описана в роботі [9]. Але цього замало, для того щоб стверджувати, що послідовність оцінок параметрів, яка побудована ЕМ-алгоритмом, гарантовано дає збіжність до локального екстремуму функції правдоподібності. Щоб встановити таку збіжність, необхідно

робити припущення, що розподіли, які розглядаються, задовольняють додаткові умови регулярності, і, зокрема, умови гладкості [1], [11]. Одночасно монотонність ЕМ-алгоритму свідчить про його сильну залежність від вибору початкового (стартового) наближення.

Задача пошуку найбільш правдоподібних оцінок параметрів змішаних розподілів ймовірностей є одним з найпопулярніших застосувань ЕМ-алгоритму. Припускається, що дані в кожному із кластерів підлягають певному закону розподілу. В межах даної задачі щільність розподілу шуканої випадкової змінної X має вигляд

$$f_\theta^X(x) = \sum_{i=1}^{k} p_i \varphi_i(x; t_i), \tag{2.11}$$

де $k \geq 1$ – відоме натуральне число;

$\varphi_1, \dots, \varphi_k$ – відомі щільності розподілів;

$\theta = (p_1, \dots, p_k,\ t_1, \dots, t_k)$ – невідомий параметр, причому кожне $p_j \geq 0$;

$p_1 + \dots + p_k = 1$, t_i, $i = 1, \dots, k$ – багатовимірні параметри.

$\varphi_1, \dots, \varphi_k$ – щільності, які будуть мати назву компонентів суміші (при вирішенні конкретно цієї задачі);

$p_1, \dots, p_k$ – ваги відповідних компонент.

Завданням розділення суміші (2.11) називається задача статистичного оцінювання параметрів $\theta = (p_1, \dots, p_k,\ t_1, \dots, t_k)$ за відомими реалізаціями випадкової величини X.

Припустимо, що є невідома вибірка значень $x = (x_1, \dots, x_n)$ випадкової змінної X. В межах моделі (2.11) логарифм класичної (неповної) функції правдоподібності параметру θ буде мати наступний вигляд формули:

$$logL(\theta; x) = log \prod_{j=1}^{n} f_\theta^X(x_j) = \sum_{j=1}^{n} log\left(\sum_{i=1}^{k} p_i \varphi_i(x_j; t_i)\right). \quad (2.12)$$

Безпосередньо пошук точки максимуму цієї функції є важким. Проте, якщо трактувати спостереження x як неповні, то функцію правдоподібності можна записати у зручнішому вигляді.

Припустимо, що поряд з випадковою величиною X, що спостерігається, задана випадкова величина Y, що не спостерігається, зі значеннями $(y_1, \dots, y_n)$, де $y_j\{1,2, \dots, k\}$ містять інформацію щодо номерів компонент, відносно до яких «генерується» спостереження $x = (x_1, \dots, x_n)$. Можна вважати, що пари значень (x_j, y_j) є стохастично незалежними реалізаціями пари випадкових змінних (X, Y).

Спільна щільність випадкових змінних X та Y позначається як $f_\theta(x, y)$. Так як дискретна випадкова змінна Y є суто неперервною відносно розрахованої міри та приймає значення $i = 1,2, \dots, k$, тому її маргінальна щільність дорівнює

$$f_\theta^Y(i) = p_i, i = 1,2, \dots, k, \quad (2.13)$$

а умовна щільність випадкової змінної X при фіксованому значенні $Y = 1$ дорівнює

$$f_\theta(x|i) = \varphi_i(x; t_i). \quad (2.14)$$

Тому, якщо значення $y = (y_1, \dots, y_n)$ були відомі наперед, то логарифм загальної функції правдоподібності мав би вигляд:

$$logL(\theta; x, y) = log \prod_{j=1}^{n} f_\theta(x_j, y_j) = \sum_{j=1}^{n} log f_\theta(x_j, y_j) =$$
$$= \sum_{j=1}^{n} log\left(f_\theta(x_j|y_j) f_\theta^Y(y_j)\right) = \sum_{j=1}^{n} log p_{y_j} + \sum_{j=1}^{n} log\varphi_{y_j}\left(x_j; t_{y_j}\right). \quad (2.15)$$

Після деяких перетворень умовне математичне очікування логарифма загальної функції правдоподібності при фіксованих значеннях $x = (x_1, \dots, x_n)$ випадкової змінної X, що спостерігається, має вигляд:

$$Q\left(\theta, \theta^{(m)}\right) = \sum_{l=1}^{k} \sum_{j=1}^{n} f_\theta(l|x_j)\, log p_l + \sum_{l=1}^{k} \sum_{j=1}^{n} f_\theta(l|x_j)\, log\varphi_l(x_j; t_l). \quad (2.16)$$

Для пошуку максимуму функції (2.16) по $\theta = (p_1, \dots, p_k,\ t_1, \dots, t_k)$ можна максимізувати доданки в правій частині рівняння (2.16) незалежно один від одного, тому що вони залежать від різних параметрів: перше залежить тільки від ваг $p_1, \dots, p_k$, а друге – тільки від параметрів $t_1, \dots, t_k$ компонент суміші. Враховуючи обмеження

$$\sum_{l=1}^{k} p_l = 1, \quad (2.17)$$

використовуючи метод невизначених множників Лагранжу, можна знайти значення $\theta = (p_1, \dots, p_k,\ t_1, \dots, t_k)$, яке утворює максимум функції (2.16). Відносно цього можна зробити припущення, що значення $\theta^{(m)} = \left(p_1^{(m)}, \dots, p_k^{(m)}, t_1^{(m)}, \dots, t_k^{(m)}\right)$ параметру θ на m-ой ітерації відомі, можна знайти $p_1^{(m)}, \dots, p_k^{(m)}$ на $m + 1$-ой ітерації ЕМ-алгоритму.

Треба відмітити те, що в прикладних застосунках ЕМ-алгоритм частіше за все застосовується щодо дослідження моделі (2.15), де $\varphi_l(x; t_l)$–

щільність нормального розподілу випадкових змінних. Хоча, саме ця модель не задовольняє умовам, які гарантують коректну роботу ЕМ-алгоритму. А саме, збіжність ЕМ-алгоритму доведена за обов'язковою умовою обмеженості логарифму функції правдоподібності.

Для сумішей нормальних розподілів зазначена умова взагалі не виконується. Також наявність великої кількості локальних максимумів логарифму функції правдоподібності для моделі (2.16) з великою кількістю $(k \geq 2)$ нормальних компонентів призводить до значної нестійкості по відношенню до початкового наближення та вихідних даних.

2.2 Можливістний метод кластерування (PCM)

Завдання кластерування багатовимірних спостережень, які поступають на опрацювання в реальному часі (послідовно, одне за одним), достатньо часто зустрічається в багатьох застосунках, які пов’язані з інтелектуальним аналізом потоків даних. Традиційний підхід до вирішення цих задач полягає в тому, що кожне спостереження може відноситись тільки до одного кластеру [12, 13], але більш природнім є випадок, коли вектор ознак, який опрацьовується з різними рівнями ймовірності або можливості, належить одразу до декількох класів. Такого роду ситуація є ключовою складовою нечітких методів кластерування і в останні роки в межах нечітких систем самоорганізації даних цікавим є підхід заснований на можливістному аналізі кластерів.

Так Р. Крішнапурам та Дж. Келлер [11] запропонували ідею послаблення обмеження на суму всіх рівнів належності, що дозволяє вирішити проблему спотворених викидами та аномаліями даних.

Нехай задана множина спостережень $X = (x_1, \dots, x_N)$, де $x_i \in R^d$, $i = 1, \dots, N$, $\beta = (\beta_1, \dots, \beta_C)$ – центри кластерів, d_{ij}^2 – відстань від точки x_i до центру β_j, а $U = \{u_{ij}\}$ – матриця розмірністю $C \times N$, елементи якої є характеристичними значеннями елементу x_i відповідно до кластеру S_i.

Необхідно розділити множину спостережень X на C нечітких кластерів $S_1, \dots, S_C$, таким чином, щоб мінімізувати функцію втрат

$$\underset{(U,\beta)}{\operatorname{argmin}} \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m} d_{ij}^{2} + \sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} \left(1 - u_{ij}\right)^{m}, \tag{2.18}$$

де $m \in [1, +\infty)$ – експоненціальна вага;

η_i – параметр кроку навчання.

На характеристичні значення рівнів належності u_{ij} замість (2.15)-(2.17) накладаються наступні обмеження:

$$u_{ij} \in [0,1] \text{ для всіх } i, j, \tag{2.19}$$

$$0 < \sum_{j=1}^{N} u_{ij} \leq N \text{ для всіх } i, \tag{2.20}$$

$$\max_{i} u_{ij} > 0 \text{ для всіх } j. \tag{2.21}$$

Мінімізація функції (2.18) передбачає, щоб у першому доданку відстань від точки x_i до центру кластеру β_j була якнайменшою, у той час, як у другому доданку u_{ij} має бути якомога ближчою до 1. Якби в (2.18) не було б другого доданку, то без обмеження виду (2.16) на u_{ij}, мінімізація функції призводила б до тривіального рішення $u_{ij} = 0$ для всіх i, j.

Необхідно зазначити, що рядки та стовпці матриці $U = \{u_{ij}\}$ є незалежними один від одного. Тому мінімізацію функції (2.18) можна звести до мінімізації CN незалежних функцій

$$u_{ij}^m d_{ij}^2 + \eta_i (1 - u_{ij})^m. \quad (2.22)$$

Згідно до обов'язкових умов локального екстремуму, отримуємо:

$$u_{ij} = 1 \Big/ 1 + \left(d_{ij}^2 \Big/ \eta_i \right)^{1/m-1}, i = 1, \dots, C, j = 1, \dots, N, \quad (2.23)$$

$$\beta_j = \sum_{j=1}^N u_{ij}^m x_i \Big/ \sum_{j=1}^N u_{ij}^m, i = 1, \dots, C. \quad (2.24)$$

Елементи матриці $U = \{u_{ij}\}$ сильно залежать від вибору параметру η_i. Якщо η_i маленьке, то відповідно і u_{ij} буде малим. Якщо η_i велике, то і значення u_{ij} буде великим. Також η_i визначає ступінь, з якою другий доданок (2.18) порівнюється з першим. Якщо обидва доданків (2.18) рівноважні, то η_i має бути порядку d_{ij}^2. Р. Крішнапуран та Дж. Келлер запропонували наступні співвідношення для η_i ([11, 12]):

$$\eta_i = \sum_{j=1}^N u_{ij}^m d_{ij}^2 \Big/ \sum_{j=1}^N u_{ij}^m, i = 1, \dots, C, \quad (2.25)$$

$$\eta_i = \sum_{u_{ij} > \alpha} d_{ij}^2 \Big/ \sum_{u_{ij} > \alpha} 1, i = 1, \dots, C. \quad (2.26)$$

де $0 < \alpha < 1$.

Параметр η_i може бути фіксованим (заданий апріорі константою) для всіх кроків навчання, якщо кластери мають схожу форму. В загальному випадку η_i змінюється на кожній ітерації алгоритму навчання, що може призвести до нестійкості, так як необхідні умови локального екстремуму отримані для фіксованого η_i. Тому, частіше за все, спочатку застосовують

метод нечіткого кластерування для ініціалізації u_{ij}, далі вже обчислюються η_i за формулою (2.25), після чого використовують алгоритм кластерування, в якому η_i обчислюється за формулою (2.26).

Значення m грає важливу роль у визначенні характеристичних значень u_{ij}. На рисунку 2.1 можна побачити, що при $m \to 1$ характеристичні значення u_{ij} наближуються до нуля для тих точок x_j, до яких d_{ij}^2 більше, чим η_i. При $m \to \infty$ характеристичні значення припиняють наближуватись до нуля. Значення $m = 2$ дає хороші результати в алгоритмі нечіткої самоорганізації. Але, у можливісному підході до вирішення завдання кластерування до такого значення m характеристичні функції зменшуються недостатньо швидко для великих значень d_{ij}^2. Тому в можливістному алгоритмі кластерування краще використовувати $m = 1.5$ [12].

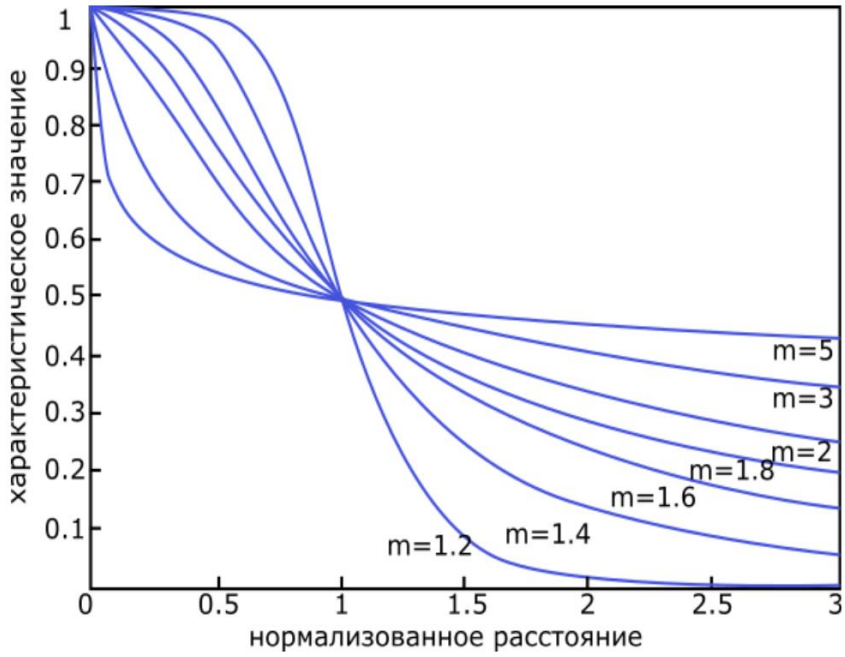

Рисунок 2.1 – Змінення функції належності відносно від значення параметру m

Загальна процедура можливістного кластерування виглядає наступним чином. Для початку необхідно сгенерувати елементи матриці $U = \{u_{ij}\}$. Далі необхідно обчислити центри кластерів за формулою (2.24). Далі виконується ітеративне повторення кроків:

1. розраховуються відстані d_{ij} від кожного спостереження x_i до центрів кластерів β_j;

2. обчислюється η_i за формулою (2.25) або (2.26);

3. перераховуються елементи матриці U за формулою (2.23);

4. перераховуються центри кластерів β_j за формулою (2.24) для нових елементів матриці U з пункту 3;

5. порівнюються $u_{ij}^{(t+1)}$ з$u_{ij}^{(t)}$, де t – номер ітерації. Якщо $\left\| u_{ij}^{(t+1)} - u_{ij}^{(t)} \right\|^2 < \varepsilon$ (для заданого ε), то процес навчання зупиняється, у всіх інших випадках – процес навчання повертається на першу ітерацію.

Запропонований Р. Крішнапурам та Дж. Келлером можливістний алгоритм – це виключно деяка ідея в межах загального підходу можливістного кластерування. Можливістний підхід означає, що характеристичне значення точки відносно кластера – це можливість точки належати кластеру або не належати.

Так як мінімізація функції (2.18) зводиться до мінімізації CN незалежних функцій (2.22), тому можуть виникнути кластери, що будуть співпадати. Ця проблема є загальною для функцій, які можна виразити як суму незалежних функцій. Причина криється не у поганому виборі доданку у (2.18), а скоріше за все у відсутності додаткових обмежень на u_{ij}. З одного боку обмеження (2.16) в алгоритмі нечіткого кластерування занадто жорстке – воно змушує викиди та аномальні точки належати одному або декільком кластерам з достатньо високим рівнем належності.

З іншого боку, обмеження (2.20) у можливістному алгоритмі дуже слабке, так як матриця U сильно залежить від вибору параметрів m та η_i. І хоча можливістний алгоритм кластерування більш робастний до викидів та

аномальних значень, так як ці аномальні значення все-таки будуть належати кластерам з маленьким рівнем належності, розраховуватись за це необхідно буде співпадінням координат центрів деяких кластерів.

Для того, щоб вирішити проблему чутливості до аномальних спостережень та викидів, а також проблему співпадіння центрів кластерів, було запропоновано декілька алгоритмів. Наприклад, в роботі [13] було запропоновано можливістно-нечіткий метод кластеризації (possibilistic fuzzy C-means – PFCM), в якому функція, яку досліджують на мінімум, включає і характеристичні значення u_{ij} та ступені належності w_{ij}. Але цей метод і досі зіштовхується з проблемами ініціалізації та вибору параметрів моделі.

Ще один алгоритм, запропонований в [14], базується на концепції, що на початковому етапі всі спостереження є по факту центрами кластерів. Після цього відбувається процедура автоматичного об’єднання точок певним чином відповідно до початкової природи даних. При цьому число кластерів знаходиться автоматично зі збереженням робастності методу. Той факт, що всі точки використовуються в якості початкових центрів кластерів, є серйозною проблемою при масштабуванні цього методу для великих об’ємів даних та опрацювання даних високої розмірності.

2.3 Матрична модифікація методу кластерування нечітких С-середніх

Проблема кластерування багатовимірних даних дуже часто зустрічається в багатьох задачах, які пов’язані з інтелектуальним аналізом даних, як представлених у вигляді фіксованого пакету, так і у вигляді потоків даних, що надходять на опрацювання одне за одним. Традиційний підхід до вирішення таких завдань дозволяє припущення, відносно того, що кожне спостереження може відноситись до одного кластеру [13, 14], хоча більш природньою є ситуація, коли вектор ознак, який подається на опрацювання, з різними рівнями належності або ймовірності, або

можливості може належати одразу декільком класам. Такий випадок є предметом, що можна розглянути відносно концепції нечітких систем для самоорганізації даних, який на даний час має величезне поширення в межах розробки систем для кластерування даних довільної природи [8], [15-19].

Традиційно входовою інформацією для задачі кластерування є набор спостережень, який складається з N n-вимірних 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 розділення входового масиву спостережень на m класів з певним рівнем належності $u_j(k)$ k-го вектору ознак j-му кластеру.

В той самий час існує великий клас задач, в яких входова інформація надходить на опрацювання не у векторному, а в матричному вигляді, 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 [16], коли входова $(N_1 \times N_2)$-матриця розбивається на $N = N_1 \cdot N_2 \cdot (n_1 \cdot n_2)^{-1}$ $(n_1 \times n_2)$-матриць-фрагментів, які необхідно кластерувати, в результаті чого формуються одномірні в деякому сенсі сегменти цього зображення. Традиційно ця задача вирішується шляхом попередньої векторизації фрагментів та у використанні вже відомих процедур, найбільш популярною з яких є метод кластерування нечітких С-середніх [8], [15].

Для опрацювання матричних даних необхідно ввести матричні методи кластерування. Тому до розгляду вводиться матричний метод нечітких С-середніх, який є узагальненням класичного методу самоорганізації, дозволяє позбутись надлишкових операцій векторизації-девекторизації при опрацюванні даних, які задані у формі двовимірного масиву, а також дозволяє опрацьовувати інформацію в послідовному режимі.

Нехай задана вибірка спостережень $x(k) = \{x_{i_1, i_2}(k)\} \in R^{n_1 \times n_2}$, $k = 1,2, \dots, N$, при цьому для зручності подальшого опрацювання ці дані попередньо центруються відносно середнього:

$$\bar{x} = {}^{1}/_{N} \sum_{k=1}^{N} x(k) \tag{2.27}$$

та нормуються на свою сферичну норму (Frobenius norm):

$$\|x(k)\| = \sqrt{Trx(k)x^{T}(k)}. \tag{2.28}$$

В якості цільової функції кластерування використовується ймовірністний критерій

$$\begin{aligned} E\big(u_j(k), c_j\big) &= \sum_{k=1}^{N} \sum_{j=1}^{m} u_j^{\beta}(k)\, D^2\big(x(k), c_j\big) = \\ &= \sum_{k=1}^{N} \sum_{j=1}^{m} u_j^{\beta}(k) Tr\big(x(k) - c_j\big)\big(x(k) - c_j\big)^{T} \end{aligned} \tag{2.29}$$

за наявністю обмежень:

$$\sum_{j=1}^{m} u_j(k) = 1, \tag{2.30}$$

або

$$\sum_{j=1}^{m} u_j(k) - 1 = 0, \tag{2.31}$$

де $k = 1,2,\dots,N, 0 < \sum_{j=1}^{m} u_j(k) < N,\ \ j = 1,2,\dots,m$;

$u_j(k) \in [1,2]$ – рівень належності спостереження $x(k)$ до j-го кластеру;

c_j – центроїд j-го кластеру;

β – параметр фаззіфікації;

$D^2\big(x(k), c_j\big)$ – міра відстані (квадрат норми) між $x(k)$ та c_j.

Результатом кластерування є $(N \times m)$-матриця $U = \{u_j(k)\}$, яка має назву матриці нечіткого розбиття.

Вводячи функцію Лагранжу:

$$L\big(u_j(k), c_j, \lambda(k)\big) = \sum_{k=1}^{N}\sum_{j=1}^{m} u_j^{\beta}(k) D^2\big(x(k), c_j\big) + \sum_{k=1}^{N} \lambda(k)\left(\sum_{j=1}^{m} u_j - 1\right) =$$

$$= \sum_{k=1}^{N}\left(\sum_{j=1}^{m} u_j^{\beta}(k) D^2\big(x(k), c_j\big) + \lambda(k)\left(\sum_{j=1}^{m} u_j(k) - 1\right)\right), \qquad (2.32)$$

де $\lambda(k)$ – невизначені множники Лагранжу, вирішують систему рівнянь Каруша-Куна-Таккеру

$$\begin{cases} \dfrac{\partial L\big(u_j(k), c_j, \lambda(k)\big)}{\partial u_j(k)} = \beta u_j^{\beta-1}(k) D^2\big(x(k), c_j\big) + \lambda(k) = 0, \\ \dfrac{\partial L\big(u_j(k), c_j, \lambda(k)\big)}{\partial \lambda_j(k)} = \sum_{j=1}^{m} u_j(k) - 1 = 0, \\ \left\{\dfrac{\partial L\big(u_j(k), c_j, \lambda(k)\big)}{\partial c_j(k)}\right\} = -2\sum_{k=1}^{N} u_j^{\beta}(k)\big(x(k) - c_j\big) = O, \end{cases} \qquad (2.33)$$

де $\left\{\frac{\partial L\left(u_j(k), c_j, \lambda(k)\right)}{\partial c_j(k)}\right\}$ – $(n_1 \times n_2)$-матриця, яка утворена частковими похідними $\frac{\partial L\left(u_j(k), c_j, \lambda(k)\right)}{\partial c_{ji_1i_2}}$;

O – матриця тієї ж самої розмірності, яка утворена нулями.

Після розв’язання системи рівнянь отримуємо результат в наступному вигляді:

$$\begin{cases} u_j(k) = \left(D^2\left(x(k), c_j\right)\right)^{1/1-\beta} \Big/ \sum_{l=1}^{m} \left(D^2\left(x(k), c_j\right)\right)^{1/1-\beta}, \\ \lambda(k) = -\left(\sum_{l=1}^{m} \left(\beta D^2(x(k), c_l)^{1/1-\beta}\right)^{1-\beta}\right), \\ c_j = \sum_{k=1}^{N} u_j^{\beta}(k)x(k) \Big/ \sum_{k=1}^{N} u_j^{\beta}(k). \end{cases} \quad (2.34)$$

Отримана система породжує широкий клас процедур кластерування. Так, обираючи $\beta = 2$, отримуємо простий та ефективний алгоритм матричного кластерування, який є узагальненням популярної процедури Дж. Бездека [8]:

$$\begin{cases} u_j(k) = \dfrac{\left(Tr\left(x(k) - c_j\right)\left(x(k) - c_j\right)^T\right)^{-1}}{\sum_{l=1}^{m} \left(Tr\left(x(k) - c_j\right)\left(x(k) - c_j\right)^T\right)^{-1}}, \\ c_j = \dfrac{\sum_{k=1}^{N} u_j^2(k)x(k)}{\sum_{k=1}^{N} u_j^2(k)}, \end{cases} \quad (2.35)$$

де Tr – символ відбитку матриці.

Функціонування алгоритму кластерування починається із задання початкової (зазвичай випадкової) матриці нечіткого розбиття U^0. На основі її розраховується початковий набір центроїдів c_j^0, які далі використовуються для обчислення нової матриці U^1.

Спостереження в пакетному режимі переобчислюються c_j^1, U^2,…, c_j^{t-1}, U^t, поки різниця $\|U^t - U^{t-1}\|$ не стане меншою за деякий апріорі заданий поріг ε. Таким чином весь масив спостережень опрацьовується декілька разів.

2.4 Матричні модифікації можливістного алгоритму С-середніх та комбінацій можливістного та нечіткого алгоритму С-середніх

Загальна різниця між ймовірністним та можливістним підходами полягає в тому, що ймовірністні алгоритми використовують відносні збіжності між об'єктами та кластерами, в той час як можливістні методи використовують абсолютні збіжності.

Замість матриці нечіткого розбиття в алгоритмі нечітких С-середніх, можливістний алгоритм С-середніх використовує $(N \times m)$-матрицю можливостей (matrix of possibilities или typicality matrix) $T = \{t_j(k)\}$, де $t_j(k) \in [0,1] -$ можливість того, що об'єкт $x(k)$ належить кластеру j.

Матриця можливостей має наступні обмеження:

$$0 < \sum_{j=1}^{m} t_j(k) \leq m,\ k = 1,2,\dots,N. \tag{2.36}$$

Це означає, що об'єкт може мати вектор можливостей, який містить тільки значення, що близькі до нуля (зазвичай такі об'єкти вважаються аномаліями або викидами), або тільки одиниці.

В запропонованому можливістному методі нечітких С-середніх формула (2.35) була замінена наступним виразом:

$$\begin{cases} t_j(k) = {1}\Big/{1 + \left(Tr\big(x(k) - c_j\big)\big(x(k) - c_j\big)^T \Big/ \gamma_j \right)^{1/\beta - 1}}, \\ c_j = {\sum_{k=1}^{N} t_j^{\beta}(k) x(k)}\Big/{\sum_{k=1}^{N} t_j^{\beta}(k)}, \end{cases} \tag{2.37}$$

де $\gamma_j > 0$ – константа, яка задається користувачем.

Можна помітити, що обчислення центроїду кластеру в формулах (2.35) та (2.37) ідентичні, з тією лиш різницею, що матриця нечіткого розбиття замінена на матрицю можливостей. Обчислення можливості належності об’єкту кластеру у формулі (2.37) може мати сенс дзвіноподібної (дзвонуватої) функції, яка має вигляд зображений на рисунку 2.2.

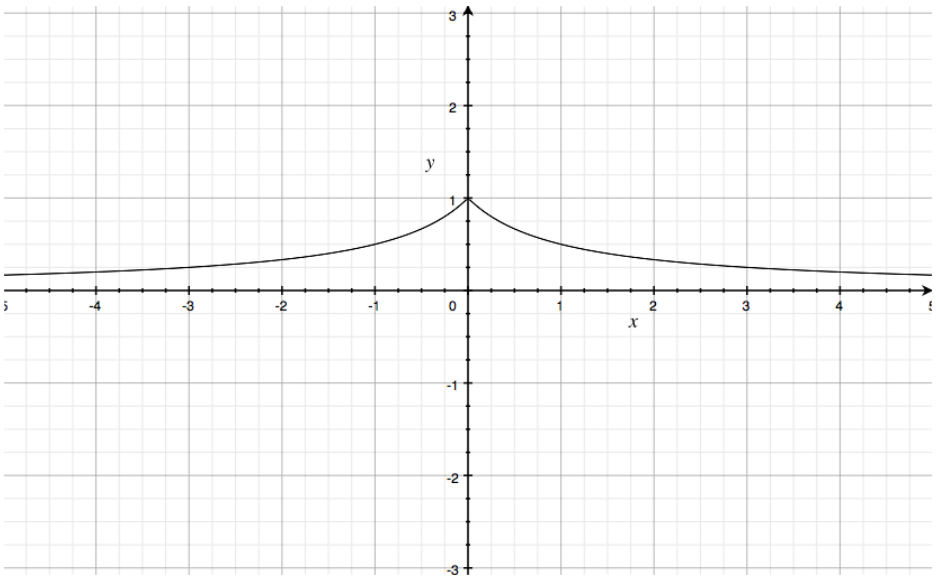

Рисунок 2.2 – Дзвіноподібна функція, яка показує залежність між відстанню та можливістю належності (для $\beta = 2$ та $\gamma_j = 1$)

Келлер та Крішнапурам запропонували обирати параметр γ_j обчислюючи рівняння (2.38)

$$\gamma_j = K \frac{\sum_{k=1}^{N} u_j^{\beta}(k) Tr\big(x(k) - c_j\big)\big(x(k) - c_j\big)^T}{\sum_{k=1}^{N} u_j^{\beta}(k)}, \tag{2.38}$$

де $K > 0$ (частіше за все обирається $K = 0$).

Але обчислення γ_j за формулою (3.9) потребує пам’ять для зберігання матриці нечіткого розбиття, а також часу для її обчислення.

Алгоритм PCM добре себе зарекомендував в ідентифікації збурень і зазвичай може бути використовуваний, коли необхідно покращити результати, які були отримані за допомогою інших методів та систем. Також цей алгоритм може поєднувати близькі кластери в один, якщо початкова кількість класів задана користувачем є надлишковою (в той же час, алгоритм PCM може об'єднувати кластери, які мають бути розділені).

Алгоритми FPCM та PFCM використовують як матрицю нечіткого розбиття, так і матрицю можливостей, намагаючись використовувати переваги обох підходів.

FPCM алгоритм використовує наступні співвідношення:

$$\begin{cases} u_j(k) = \dfrac{\left(Tr\left(x(k)-c_j\right)\left(x(k)-c_j\right)^T\right)^{1/1-\beta}}{\sum_{l=1}^{m}\left(Tr\left(x(k)-c_j\right)\left(x(k)-c_j\right)^T\right)^{1/1-\beta}}, \\ t_j(k) = \dfrac{\left(Tr\left(x(k)-c_j\right)\left(x(k)-c_j\right)^T\right)^{1/1-\eta}}{\sum_{l=1}^{N}\left(Tr\left(x(k)-c_j\right)\left(x(k)-c_j\right)^T\right)^{1/1-\eta}}, \\ c_j = \dfrac{\sum_{k=1}^{N}\left(u_j^{\beta}(k)+t_j^{\eta}(k)\right)x(k)}{\sum_{k=1}^{N}\left(u_j^{\beta}(k)+t_j^{\eta}(k)\right)}, \end{cases} \quad (2.39)$$

де $\eta > 0$ (частіше за все $\eta = 2$).

Алгоритм FPCM використовує стандартну процедуру обчислення матриці нечіткого розбиття, але матриця можливостей обчислюється за новою формулою. Центроїди кластерів розраховуються, використовуючи суму обох матриць.

Метод PFCM використовує стандартну процедуру розрахунку матриці нечіткого розбиття (як наведено у формулі (2.35)). Процедура обчислення матриці можливостей була взята з алгоритму PCM (2.27) та незначно модифікована. Центроїди обчислюються так само як і в алгоритмі FPCM, але обидві матриці мають свої ваги:

$$\begin{cases} u_j(k) = \dfrac{\left(Tr\left(x(k)-c_j\right)\left(x(k)-c_j\right)^T\right)^{1/1-\beta}}{\sum_{l=1}^{m}\left(Tr\left(x(k)-c_j\right)\left(x(k)-c_j\right)^T\right)^{1/1-\beta}}, \\ t_j(k) = \dfrac{1}{1+\left(b\dfrac{Tr\left(x(k)-c_j\right)\left(x(k)-c_j\right)^T}{\gamma_j}\right)}, \\ c_j = \dfrac{\sum_{k=1}^{N}\left(au_j^{\beta}(k)+bt_j^{\eta}(k)\right)x(k)}{\sum_{k=1}^{N}\left(au_j^{\beta}(k)+bt_j^{\eta}(k)\right)}, \end{cases} \quad (2.40)$$

де $a > 0$, $b > 0$.

Константи a та b визначають відносну важливість матриці нечіткого розбиття та матриці можливостей у функції обчислення центроїдів. Обравши $a = 0$, можна перетворити (2.40) в PCM, а обравши $b = 0$ (2.40) набуває вигляду класичного методу FCM.

Аналізуючи всі розглянуті вище методи та алгоритми, можна зробити низку висновків. По-перше, функція належності алгоритму FCM з його обмеженнями є достатньо потужною, що дозволяє відносити аномальні спостереження до одного або більше кластерів, що, в свою чергу, може сильно впливати на загальну структуру набору даних. З іншого боку, обмеження методу PCM для можливостей є дуже слабким – він дозволяє належати об’єктам до кластерів незалежно від інших даних.

Також PCM дуже чутливий до початкової ініціалізації матриці можливостей. Метод PFCM є дієвою комбінацією двох підходів і результати кластерування, використовуючи його залежать від вибору параметрів a, b, β, η. Подальші дослідження необхідні для того, щоб зробити висновки щодо вірного вибору цих параметрів.

2.5 Адаптивний матричний алгоритм кластерування нечітких С-середніх

Алгоритм (2.35) розширяється, якщо дані на опрацювання надходять у вигляді послідовності об’єктів. Для цього, застосовуючи до лагранжіану (2.33) процедуру пошуку сідлової точки Ерроу-Гурвіца-Удзави, при надходженні $(k+1)$-го спостереження оцінок рівнів належності та центроїдів можливо уточнити за допомогою рекурентного співвідношення

$$\begin{cases} u_j(k+1) = \dfrac{\left(D^2\left(x(k+1), c_j(k)\right)^{\frac{1}{1-\beta}}\right)}{\sum_{l=1}^{m}\left(D^2\left(x(k+1), c_j(k)\right)^{\frac{1}{1-\beta}}\right)}, \\ c_j(k+1) = c_j(k) - \eta(k)\left\{\dfrac{\partial L\left(u_j(k+1), c_j, \lambda(k+1)\right)}{\partial c_j}\right\} = \\ = c_j(k) - \eta(k) u_j^{\beta}(k+1)\left(x(k+1) - c_j(k)\right). \end{cases} \tag{2.41}$$

де β – параметр фаззіфікації, який має довільне значення;

$\eta(k)$ – параметр кроку навчання (learning rate parameter).

Для параметру фаззіфікації $\beta = 2$ можна записати наступну систему

$$\begin{cases} u_j(k+1) = \dfrac{\left(Tr\left(x(k+1) - c_j(k)\right)\left(x(k+1) - c_j(k)\right)^T\right)^{-1}}{\sum_{l=1}^{m}\left(Tr\left(x(k+1) - c_j(k)\right)\left(x(k+1) - c_j(k)\right)^T\right)^{-1}}, \\ c_j(k+1) = c_j(k) - \eta(k) u_j^2(k+1)\left(x(k+1) - c_j(k)\right). \end{cases} \tag{2.42}$$

Помітно, що рівняння (2.41) є адаптивною версією процедури (2.34), а (2.42) – співпадає з формулою (2.35). Також, процедури (2.41), (2.42) є узагальненням рекурентних алгоритмів нечіткого кластерування, що було розглянуто у [17, 18], та відрізняється чисельною реалізацією.

3 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

Головною метою імітаційного моделювання є підтвердження теоретичних досліджень та аналіз отриманих результатів, а так ж порівняння створених моделей для кластерування матричних даних з існуючими класичними алгоритмами кластерування даних.

3.1 Вибір програмних засобів для реалізації моделей методів кластерування як векторних, так і матричних даних

Мовою програмування для виконання цієї робити була обрана мова Python 3 версії. Була обрана саме третя версія цієї мови, бо у 2020 році підтримка другої версії Python була зупинена, а ми сподіваємося використовувати нашу систему довгий час.

Головними перевагами мови програмування Python є:

Зрозумілість. Python – високорівнева і зрозуміла мова програмування, з якою зручно працювати. Завдяки її лаконічності і зручності читання вона добре підходить для навчання розробці ПЗ.

Крім того, Python добре підходить для машинного навчання, тому що самі алгоритми машинного навчання складні для розуміння. При роботі з Python розробнику не потрібно приділяти багато уваги безпосередньо написання коду: всю увагу він може зосередити на вирішенні більш складних завдань, пов'язаних з машинним навчанням. Простий синтаксис мови Python допомагає розробнику тестувати складні алгоритми з мінімальною витратою часу на їх реалізацію.

Велика підтримка. Ще одна перевага Python – це велика підтримка і якісна документація. Існує безліч корисних ресурсів про Python, на яких програміст може отримати допомогу і консультацію, перебуваючи на будь-якому етапі розробки.

Гнучкість. Наступна перевага Python в машинному навчанні полягає в його гнучкості: наприклад, у розробника є вибір між об'єктно-орієнтованим підходом і скриптами. Python допомагає об'єднувати різні типи даних. Більш того, Python особливо зручний для тих розробників, які більшу частину коду пишуть за допомогою IDE.

Популярність. Як вже відзначили, Python набрав популярність завдяки простій і зрозумілій структурі синтаксису. Саме тому на ринку багато Python-розробників, які готові працювати над проектами, пов'язаними з машинним навчанням.

Великий вибір бібліотек і фреймворків. Одна з основних причин, чому Python використовується для машинного навчання полягає в тому, що у нього є безліч фреймворків спеціально розроблених для машинного навчання, які спрощують процес написання коду і скорочують час на розробку.

Перераховані вище фактори пояснюють, чому Python був обраний мовою програмування для цієї роботи. Його простота допомагає працювати над складними алгоритмами машинного навчання.

3.2 Кластерування матричних даних

Для підтвердження запропонованих в магістерській кваліфікаційній роботі матричних модифікацій алгоритму кластерування нечітких С-середніх було використано декілька наборів тестових даних з UCI-репозіторію, а також з Kaggle.com. Крім цього, в тестуванні розробленого методу кластерування було використано набір супутникових знімків (цифрові зображення) міста Харкова.

Після запуску програми в операційну пам’ять середовища завантажуються дані, з яких формується вибірка векторних або матричних спостережень. Після цього необхідно задати основні параметри алгоритму:

– кількість кластерів m;

– поріг зупинення алгоритму ε;

– початкові координати центроїдів;

– задати співвідношення за яким генеральна сукупність входових даних буде ділитись на навчальну та тренувальну підмножини.

За необхідністю, це залежіть від вибірки, яка опрацьовується необхідно провести попередній аналіз входових даних (EDA – explorationary data analysis). Цей етап додається для виявлення додаткових закономірностей в даних та у випадку, коли кількість кластерів апріорі невідома. Перед тим, як переходити до етапу навчання моделей, необхідно додатково ще провести центрування відносно середнього значення та нормування даних на сферичну норму.

Після того, як вибірка спостережень завантажена та передоброблена, задається випадкова матриця нечіткого розбиття U^0. Далі алгоритм послідовно в пакетному режимі обчислює нові значення прототипів майбутніх центроїдів c_j, де j – це номер кластера і матриці нечіткого розбиття U, цей процес циклічно повторюється до моменту коли різниця між значеннями $\|U^t - U^{t-1}\|$ не стане меншою за значення порогу ε.

Адаптивний алгоритм працює за схожим принципом, з тією лиш різницею, що кожний об’єкт опрацьовується лише один раз, та це виконується не у пакетному режимі, а в режимі online, коли спостереження на опрацювання надходять одне за одним.

Після отримання кінцевого набору прототипів центроїдів, для спостережень, які не приймали участь в процесі навчання, розраховуються вектори належності за формулою (2.35) для пакетного алгоритму кластерування, та за формулою (2.37) для адаптивного (послідовного) алгоритму. Кінцевим результатом роботи алгоритму є кінцева матриця нечіткого розбиття для всіх об’єктів із досліджуваної генеральної сукупності даних та зафіксовані вже координати центроїдів.

При обробці цифрових зображень, об’єкти (матриці або вектори однакової розмірності) утворюються із фрагментів цього зображення, а

кожен піксель із кольорової моделі RGB (Red-Green-Blue) перекодовується в модель Grayscale, де яскравість пікселю описується скалярним значенням в інтервалі [0, 1]. Перетворення зображень із моделі RGB до Grayscale моделі виконується за формулою (3.1):

$$Y = \frac{0.299R + 0.587G + 0.114B}{255}, \quad (3.1)$$

де Y – яскравість пікселю;

R, G, B – канали червоного, зеленого та синього кольору, відповідні значення яких знаходяться в інтервалі [0, 255].

Набори, що сформовано з цифрових хображень, опрацьовуються за тим самим принципом, що і стандартні вибірки, які мають вигляд таблиць Об’єкт-Властивість.

Після опрацювання кожного зображення, кожному кластеру присвоюється колір відносно до Grayscale моделі, а кожний об’єкт фарбується в колір найближчого до нього кластеру.

Приклад цифрового зображення, що надходить на опрацювання наведено на рисунку 3.1, а приклад перетвореного до Grayscale моделі зображено на рисунку 3.2.

Рисунок 3.1 – Цифрове зображення, що надходить на опрацювання

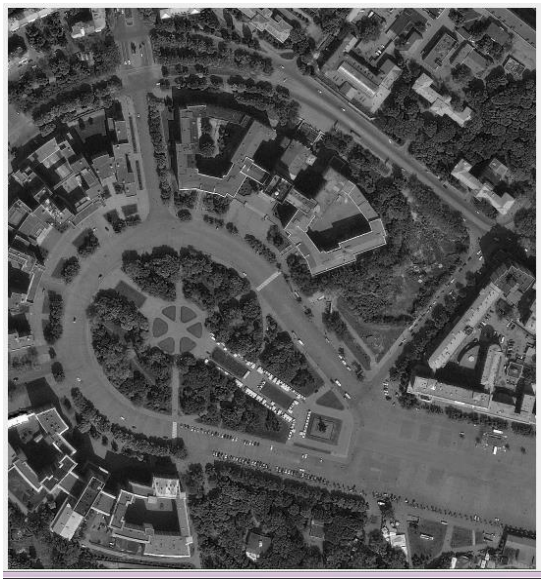

Рисунок 3.2 – Зображення перетворено до Grayscale моделі

3.3 Аналіз отриманих результатів

Для оцінки якості роботи алгоритму вжито наступні критерії: Partition Coefficient (PC), Classification Entropy (CE), Partition Index (PI) при однаковій початково ініціалізованої матриці нечіткого розбиття U^0.

В табл. 3.1 наведені результати точності та часу роботи алгоритмів кластерування на наборі даних Iris, а у табл. 3.2 – на супутниковому цифровому зображенні міста Харкова. Час зазначено в середньому для однієї ітерації з урахуванням операції векторизації-девекторизації.

Таблиця 3.1 – Результати кластерування набору даних Iris

Алгоритм кластерування	Критерії оцінки якості алгоритму			Час, хв
	PC	CE	PI	
Нечітких С-середніх	0.531	0.811	12.19	0.003
Матричний алгоритм нечітких С-середніх	0.531	0.811	12.19	0.0025

Таблиця 3.2 – Результати кластерування цифрового супутникового зображення м. Харкова

Алгоритм кластерування	Критерії оцінки якості алгоритму			Час, хв
	PC	CE	PI	
Нечітких С-середніх	0.697	0.419	8.23	1.9
Матричний алгоритм С-середніх	0.697	0.419	8.23	1.8

На рисунках 3.3-3.6 наведено візуалізацію остаточного етапу кластерування наборі даних з UCI-репозіторію.

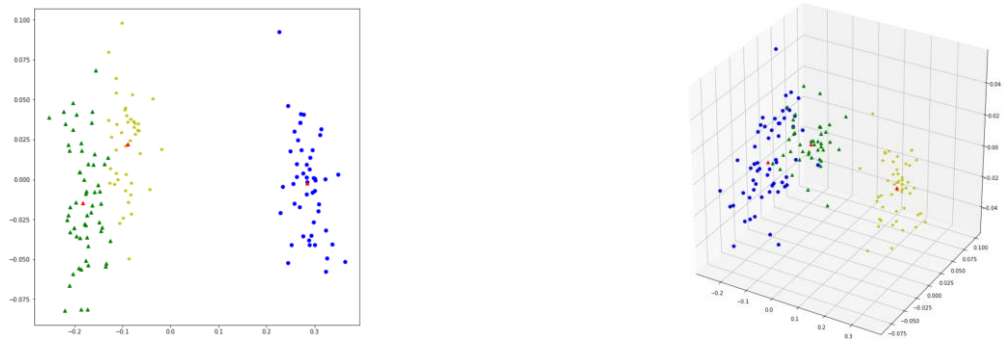

Рисунок 3.3 – Візуалізація кластерування набору даних Iris з фінальним розташуванням координат центроїдів кластерів

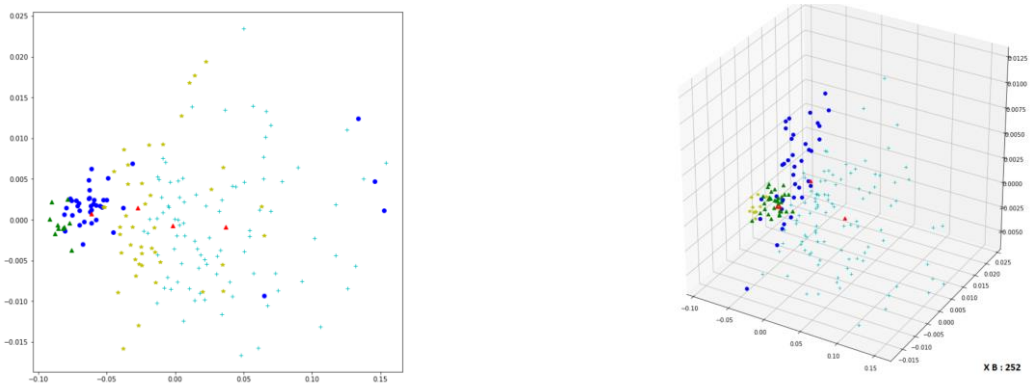

Рисунок 3.4 – Візуалізація кластерування набору даних Wine з фінальним розташуванням координат центроїдів кластерів

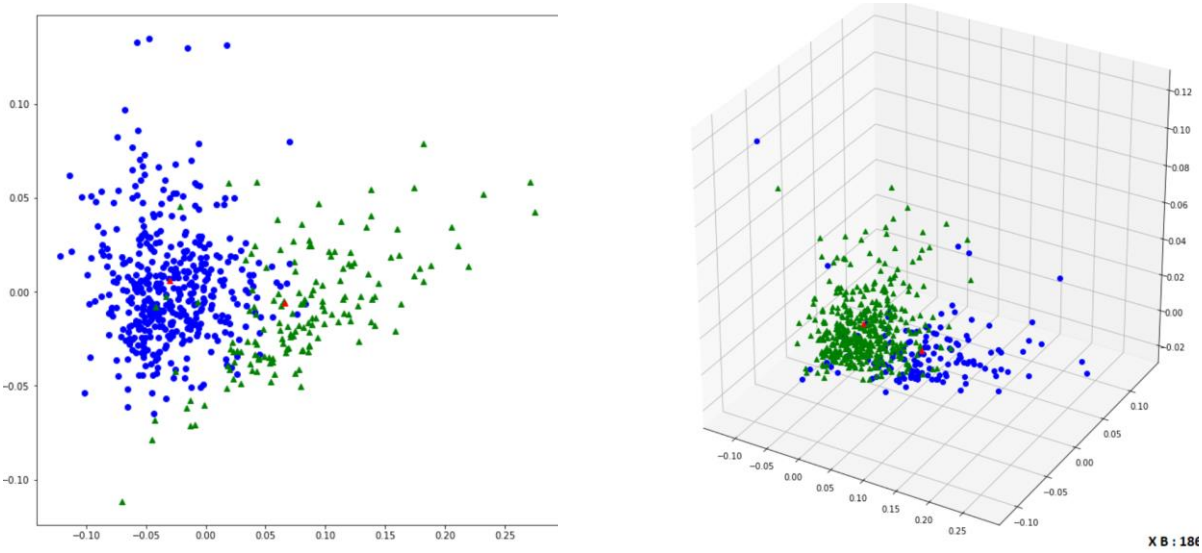

Рисунок 3.5 – Візуалізація кластерування набору даних Wine з фінальним розташуванням координат центроїдів кластерів

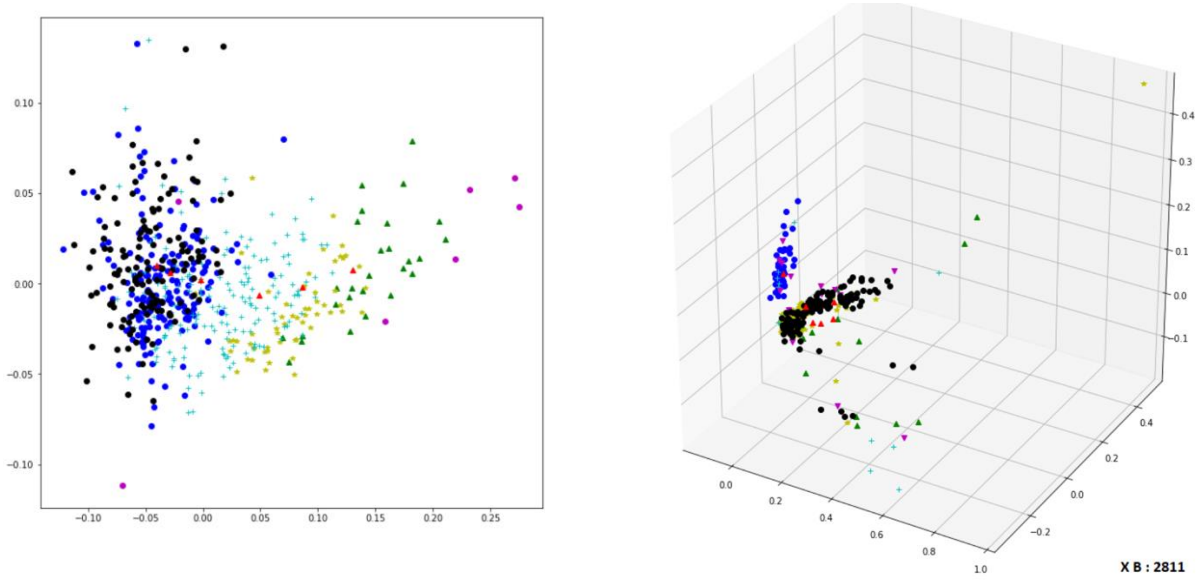

Рисунок 3.6 – Візуалізація кластерування набору даних Dermatology з фінальним розташуванням координат центроїдів кластерів

Для отримання фінальних візуалізацій всі набори даних були скомпресовані в дві та три головні компоненти.

На рисунках 3.7-3.10 відповідно представлені входове зображення, предоброблена вибірка (20% об’єктів), результат кластерування та процес роботи алгоритму.

Рисунок 3.7 – Входове цифрове зображення

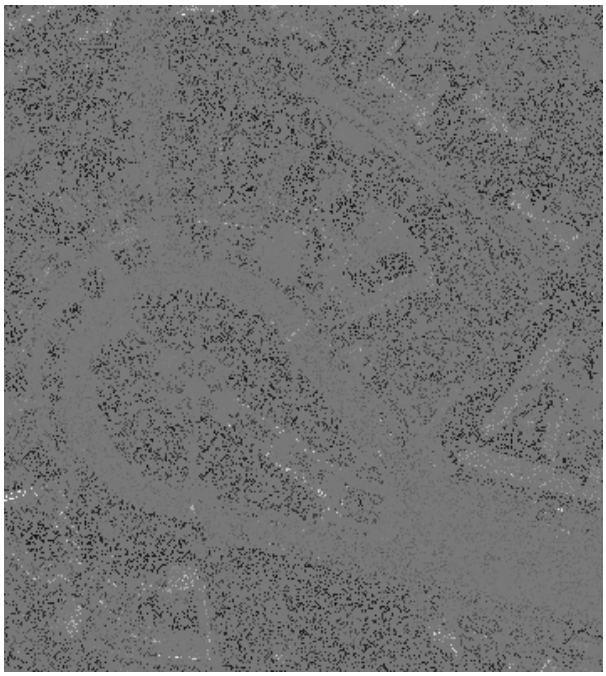

Рисунок 3.8 – Предоброблена вибірка (20% об’єктів)

Рисунок 3.9 – Зображення після кластерування

```
Iteration: 1    ε: 24,154805  Time: 1,86880 s
Iteration: 2    ε: 1,835973   Time: 1,90209 s
Iteration: 3    ε: 0,628801   Time: 2,60592 s
Iteration: 4    ε: 0,353193   Time: 2,27173 s
Iteration: 5    ε: 0,215733   Time: 2,23635 s
Iteration: 6    ε: 0,150384   Time: 2,26352 s
Iteration: 7    ε: 0,109181   Time: 2,35191 s
Iteration: 8    ε: 0,080619   Time: 2,26822 s
Iteration: 9    ε: 0,060460   Time: 2,62709 s
Iteration: 10   ε: 0,046043   Time: 2,48717 s
Iteration: 11   ε: 0,035485   Time: 2,28367 s
Iteration: 12   ε: 0,027542   Time: 2,19080 s
Iteration: 13   ε: 0,021493   Time: 2,24285 s
Iteration: 14   ε: 0,016868   Time: 2,26129 s
Iteration: 15   ε: 0,013294   Time: 2,39989 s
Iteration: 16   ε: 0,010520   Time: 2,19403 s
Iteration: 17   ε: 0,008349   Time: 2,18732 s
```

Рисунок 3.10 – Тривалість процесу роботи алгоритму кластерування

На рисунку 3.11 наведено результат кластерування цифрового зображення адаптивним матричним алгоритмом нечітких С-середніх.

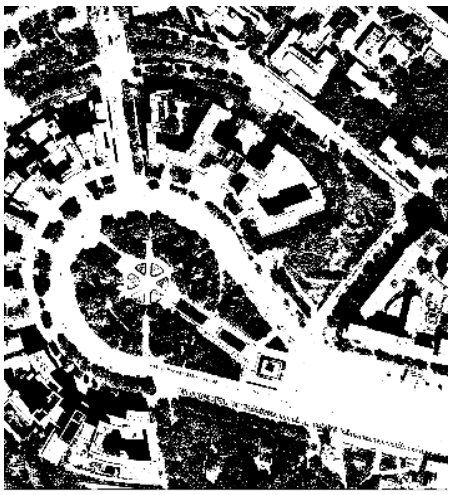

Рисунок 3.11 – Виходове зображення після кластерування адаптивним методом нечітких С-середніх

На рисунку 3.12 наведено приклад кінцевої матриці нечіткого розбиття, а на рисунку 3.13 кінцеві значення координат центроїдів кластерів, які були отримані в результаті кластерування набору даних Iris.

	A	B	C	D
1		Cluster # 0	Cluster # 1	Cluster # 2
2	Object # 0	0.000496325	0.00067804	0.99882563
3	Object # 1	0.006529864	0.01019459	0.98327555
4	Object # 2	0.003900382	0.0059351	0.99016452
5	Object # 3	0.009136773	0.01450453	0.97635869
6	Object # 4	0.000934492	0.00126564	0.99779986
7	Object # 5	0.015379749	0.01832698	0.96629327
8	Object # 6	0.004997068	0.00735954	0.98764339
9	Object # 7	0.000170108	0.00024099	0.9995889
10	Object # 8	0.01783695	0.03004381	0.95211924
11	Object # 9	0.004897069	0.00754933	0.9875536
12	Object # 10	0.008003043	0.01008681	0.98191015
13	Object # 11	0.00259657	0.00378084	0.99362259
14	Object # 12	0.007549432	0.01196828	0.98048229
15	Object # 13	0.013171975	0.02146973	0.96535829
16	Object # 14	0.029986294	0.03415834	0.93585537
17	Object # 15	0.04761057	0.0500123	0.90237713
18	Object # 16	0.013894244	0.01671872	0.96938704
19	Object # 17	0.000628996	0.00085571	0.99851529
20	Object # 18	0.023032024	0.02694006	0.95002792

Рисунок 3.12 – Матриця нечіткого розбиття набору даних Iris

```
Centroid # 0
[0,3631      -0,0708
 0,8561      0,3609     ]
Centroid # 1
[-0,2986     -0,8378
 0,4570      0,0063     ]
Centroid # 2
[-0,3095     0,1376
 -0,8692     -0,3602    ]
```

Рисунок 3.13 – Значення координат центроїдів кластерів набору даних Iris

В останньому експерименті, для фінального підтвердження працездатності розробленого адаптивного методу нечіткого кластерування С-середніх, було використано набір даних Mall_Customers зі змагання на Kaggle.com. Цей набір даних цікавий тим, що на відміну від використаних тестових даних з репозіторію, він не містить міток правильної класифікації, і для того, щоб встановити приблизну кількість кластерів, було проведено попередній аналіз цього набору даних.

Основні статистичні характеристики набору даних Mall_Customers наведені на рисунку 3.14.

index	CustomerID	Age	Income	SpendingScore
count	200.0	200.0	200.0	200.0
mean	100.5	38.85	60.56	50.2
std	57.879184513951124	13.969007331558883	26.264721165271254	25.823521668370162
min	1.0	18.0	15.0	1.0
25%	50.75	28.75	41.5	34.75
50%	100.5	36.0	61.5	50.0
75%	150.25	49.0	78.0	73.0
max	200.0	70.0	137.0	99.0

Рисунок 3.14 – Основі статистичні характеристики набору даних Mall_Customers

Також було проведено перевірку на наявність пропущених значень по кожній змінній досліджуваного набору даних.

```
CustomerID      0
Gender          0
Age             0
Income          0
SpendingScore   0
dtype: int64
```

Рисунок 3.15 – Перевірка набору даних Mall_Customers на наявність пропущених значень

Завдяки статистичному аналізу можна зробити висновок, що змінна CustomerID не впливає на аналіз даних тому її можна видалити.

На рисунку 3.16 наведені діаграми розподілу по змінним Age , Income та SpendingScore. На рисунку 3.17 представлені співвідношення змінних Age , Income та SpendingScore відносно змінної Gender. З наведених діаграм можна зробити висновок, що змінна Gender не грає суттєвої ролі в рівні заробітку, рівню витрат чи якійсь особливій закономірності.

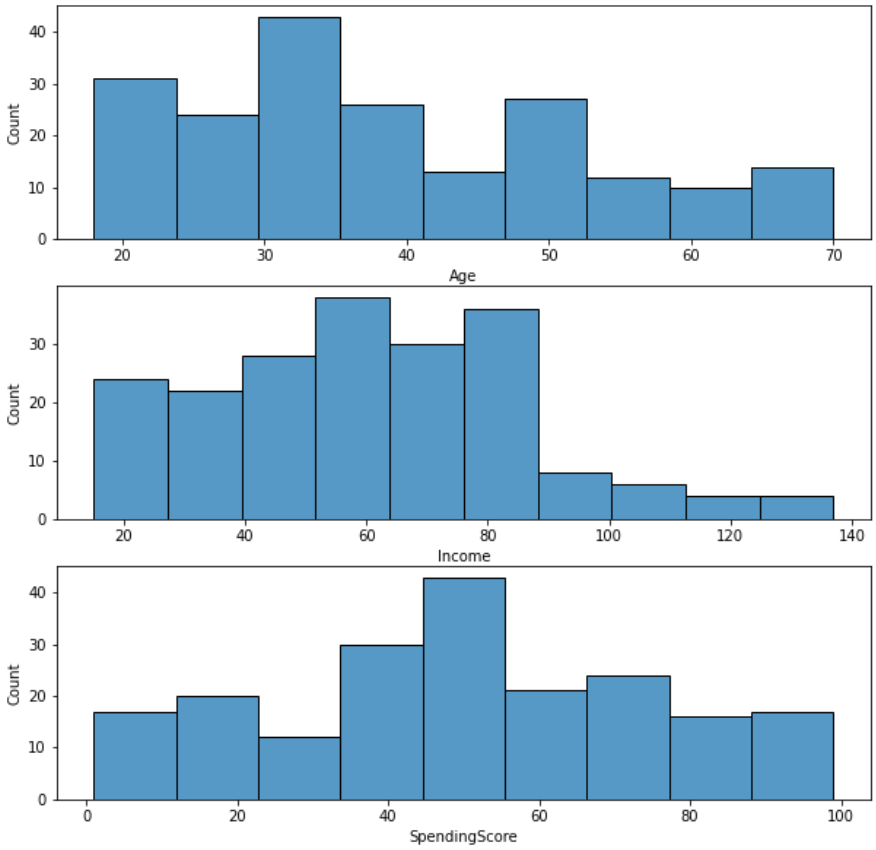

Рисунок 3.16 – Діаграми розподілу за змінними Age , Income та SpendingScore

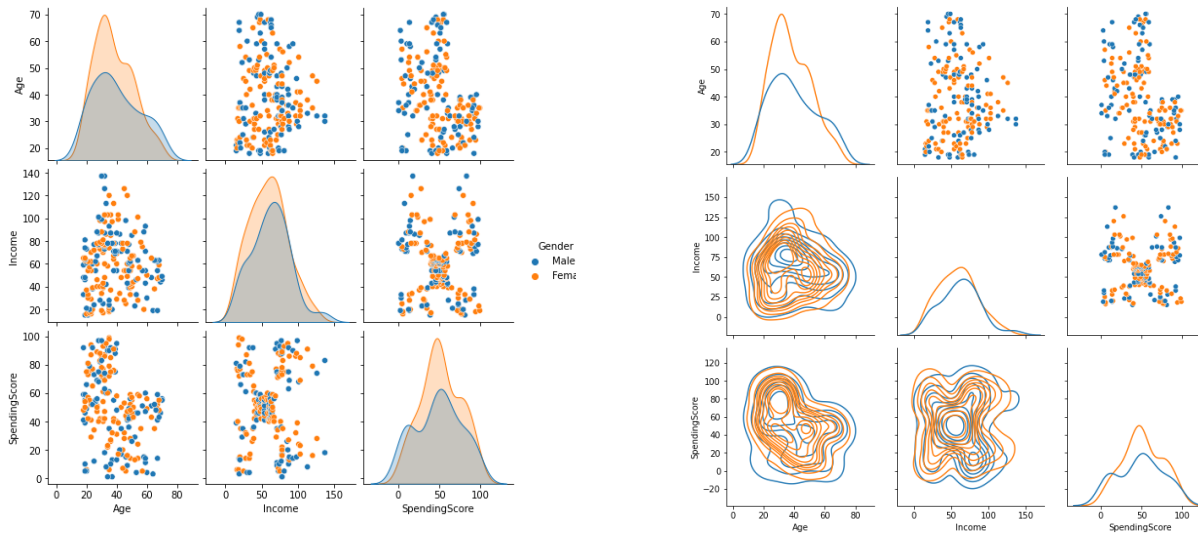

Рисунок 3.17 – Співвідношення змінних Age , Income та SpendingScore відносно змінної Gender

На рисунку 3.18 зображено гістограми змінних Age та Income відносно змінної Gender.

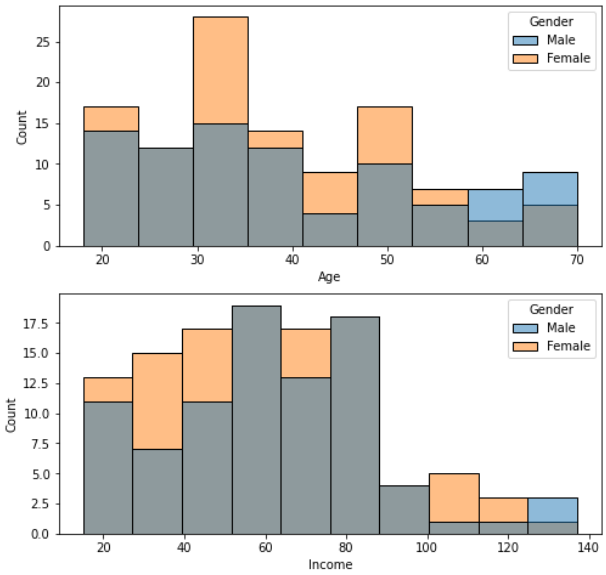

Рисунок 3.18 – Гістограми змінних Age та Income відносно змінної Gender

В межах проведеного попереднього статистичного аналізу набору даних, було зроблено припущення, що цей набір даних містить 5 або 6 кластерів. На рисунку 3.19 наведені фінальні візуалізації кластерування набору даних Mall_Customers адаптивним нечітким алгоритмом С-середніх.

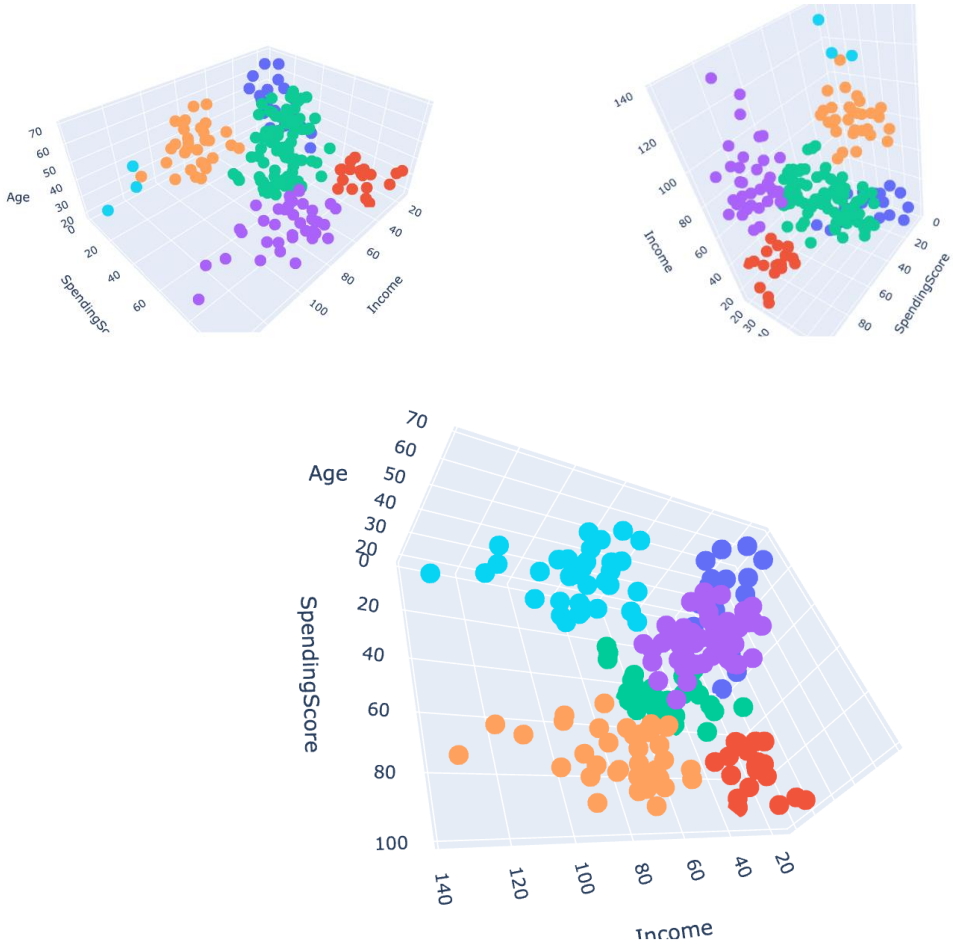

Рисунок 3.19 – Результати кластерування набору даних Mall_Customers адаптивним нечітким алгоритмом С-середніх

Імітаційне моделювання експериментальних досліджень, які було проведено у попередньому розділі магістерської кваліфікаційної роботи, підтверджує працездатність введеного адаптивного нечіткого алгоритму кластервання С-середніх. Запропонований алгоритм можна використовувати як для опрацювання векторних даних, так і для вирішення завдання кластерування матричних даних.

ВИСНОВКИ

В кваліфікаційній роботі було розроблено чисельно прості модифікації ймовірністного та можливісного алгоритмів нечітких С-середніх, які можуть бути використані для опрацювання інформації в матричній формі як в пакетному так і в послідовному режимі опрацювання інформації.

Дослідження предметної області кластерування великих наборів даних у багатовимірному просторі та оцінка базових методів рішення цієї задачі показує, що більша частина часу витрачається на вирішення додаткової задачі попередньої векторизації матричних даних.

Застосування розроблених алгоритмів дозволяє покращити час кластерування базових векторних методів, дозволивши відмовитись від операції векторизації-девекторизації входових сигналів.

Проведені експериментальні дослідження підтвердили доцільність використання матричних модифікацій алгоритмів кластерування нечітких С-середніх. Швидкість роботи модифікованих алгоритмів перевищує швидкість класичних методів FCM, PCM, FPCM и PFCM, при цьому точність кластерування залишається тією самою.

Запропоновані алгоритми можуть бути застосовані в задачах аналізу даних різної природи, наприклад, біомедичних спостережень, екологічного моніторингу, сателітних (супутникових) знімків.

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: ТЕЛЕТЕХ, 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. Hoeppner 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 algoritms. *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.