Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Центр Післядипломної освіти
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Використання та дослідження методів глибинного навчання при вирішенні задач динамічного аналізу даних
(тема)

Виконав:
студент 2 курсу, групи СШІмзд-20-2
Патлань К.В.
(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., д.т.н. Удовенко С.Г.
(посада, прізвище, ініціали)

Допускається до захисту

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

2022 р.

Харківський національний університет радіоелектроніки

Центр Післядипломної освіти
(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 Патлань Катерині Вікторівні
(прізвище, ім'я, по батькові)

1. Тема роботи Використання та дослідження методів глибинного навчання при вирішенні задач динамічного аналізу даних

затверджена наказом університету від __ _____ 20 __ р. № _____

2. Термін подання студентом роботи до екзаменаційної комісії 17 травня 20 22 р.

3. Вихідні дані до роботи Науково-технічні публікації щодо дослідження методів глибинного навчання, дані відомих наукових проектів щодо розробки та дослідження задач динамічного аналіз, інші інтернет джерела і література із вказаної теми

4. Перелік питань, що потрібно опрацювати в роботі
аналіз предметної області і постановка задачі дослідження; оптимізація функцій

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)_____

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 кваліфікаційну роботу	28.03.2022	Виконано
2	Аналіз завдання та пошук літератури за темою	29.03.2022–30.03.2022	Виконано
3	Опрацювання літератури та аналіз об'єкту	31.03.2022–05.04.2021	Виконано
4	Вибір програмних засобів для розробки системи	6.04.2022–10.04.2022	Виконано
5	Розробка програмного засобу	11.04.2022–15.04.2022	Виконано
6	Аналіз отриманих результатів	16.04.2022–20.04.2022	Виконано
7	Оформлювання пояснювальної записки	21.04.2022–29.04.2022	Виконано
8	Проходження нормоконтролю	30.04.2022–02.05.2022	Виконано
9	Оформлення презентаційних матеріалів	03.05.2022-04.052022	Виконано
10	Представлення кваліфікаційної роботи	17.05.22	виконано

Дата видачі завдання 28 березня__ 20 22_ р.

Студент _____
(підпис)

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

РЕФЕРАТ

Пояснювальна записка: 69 с., 13 рис., 3 табл.,1 дод., 48 джерел.

ГЛИБИННЕ САМОНАВЧАННЯ, ГЛОБАЛЬНИЙ ЕКСТРЕМУМ, ЕВОЛЮЦІЙНИЙ РОЙОВИЙ АЛГОРИТМ, КОСЯКИ РИБ, НЕЧІТКА КЛАСТЕРИЗАЦІЯ, ОБРОБКА ДАНИХ.

Об'єкт дослідження – дослідження методів глибинного самонавчання, за умов дефіциту та викривленості вихідної інформації.

Предмет дослідження – застосування модифікованого еволюційного ройового алгоритму.

Метою даного дослідження є запропонувати метод з використанням еволюційної оптимізації, який був би позбавлений недоліків традиційних підходів до кластеризації даних.

Методи дослідження – кластерний аналіз.

Припускається, що модифікований ройовий алгоритм має покращені властивості в кластерному аналізі.

РЕФЕРАТ

Объяснительная записка: 69 с., 13 рис., 3 табл.,1 прил., 48 источников.

ГЛИБИННОЕ САМООБУЧЕНИЕ, ГЛОБАЛЬНЫЙ ЭКСТРЕМУМ, КОСЯКИ РЫБ, НЕЧЕТКАЯ КЛАСТЕРИЗАЦИЯ, ОБРАБОТКА ДАННЫХ, ЭВОЛЮЦИОННЫЙ РОЙОВЫЙ АЛГОРИТМ.

Объект исследования – исследование методов глубинного самообучения, в условиях дефицита и искривленности исходной информации.

Предмет исследования – применение модифицированного эволюционного роевого алгоритма.

Целью данного исследования является предложить метод с использованием эволюционной оптимизации, который был лишен недостатков традиционных подходов к кластеризации данных.

Методы исследования – кластерный анализ.

Предполагается, что модифицированный роевой алгоритм обладает улучшенными свойствами в кластерном анализе.

ABSTRACT

Explanatory note: 69 p., 13 fig., 3 tabl., 1 ann., 48 sources.

DATA PROCESSING, DEEP SELF-LEARNING, EVOLUTIONARY SWORD ALGORITHM, FISH SCHOLL, FUZZY CLUSTERIZATION, GLOBAL EXTREME.

Object of research – research of methods of deep self-study, in conditions of deficit and distortion of source information.

Subject of study – application of a modified evolutionary swarm algorithm.

The aim of this study is to propose a method using evolutionary optimization, which would be free from the shortcomings of traditional approaches to data clustering.

Research methods – cluster analysis.

It is assumed that the modified swarm algorithm has improved properties in cluster analysis.

ЗМІСТ

ВСТУП

Нині все більше і більше даних збирається для академічних, наукових досліджень, приватного та особистого використання, інституційного використання, комерційного використання. Ці зібрані дані необхідно зберігати, сортувати, фільтрувати, аналізувати та представляти і навіть вимагати передачі даних, щоб це було корисно. Цей процес може бути простим або складним залежно від масштабу, в якому здійснюється збір даних, і складності результатів, які необхідно отримати. Час, затрачений на отримання бажаного результату, залежить від операцій, які необхідно виконати над зібраними даними, і від характеру вихідного файлу, який необхідно отримати. Ця проблема стає більш гострою при роботі з дуже великим обсягом даних.

Виникають завдання, пов’язані з необхідністю обробки великих масивів даних з метою пошуку нових закономірностей, встановлення і виявлення нових знань, що можуть бути вирішені у вигляді нових аналітичних систем, які базуючись на методах інтелектуального аналізу даних, зможуть надати користувачу обґрунтовану інформацію для прийняття рішень про асортимент, розміщення та комплектацію товарів і стратегію розвитку підприємства.

У таких випадках потреба в обробці стає все більш критичною. У таких випадках в дію вступають інтелектуальний аналіз даних і керування даними , без яких неможливо отримати оптимальні результати.

Можливості глибокого навчання – безмежні, алгоритми глибокого навчання, які стоять за лаштунками популярних програм глибокого навчання, таких як розпізнавання мови, автономні транспортні засоби та роботи глибокого навчання, і це лише деякі з них

Однак, незважаючи на вагомі переваги, традиційні алгоритми кластеризації мають недолік, який ускладнює їх використання при аналізі

даних оскільки функція кластеризації багатоекстримальна, а аналітичні метеди знаходять лише локальний екстремум.

Таким чином, вирішення задачі аналізу потоків даних потребує модифікацій наявних алгоритмів для покращення вирішення задач оптимізації.

У магістерській кваліфікаційній роботі, запропоновано метод з використанням еволюційної оптимізації, який був би позбавлений недоліків традиційних підходів до кластеризації даних.

Об’єкт дослідження:

Дослідження методів глибинного самонавчання, за умов дефіциту та викривленості вихідної інформації.

Предмет дослідження виступає застосування модифікованого еволюційного ройового алгоритму.

Розглянуто задачу нечіткої кластеризації масиву спостережень на основі нечіткого імовірнісного підходу, в основу якого покладено алгоритм нечітких С-середніх, який було переформулювало в задачу безумовної багатоекстремальної оптимізації. Для вирішення задачі використана рандомізована модифікація алгоритму оптимізації котячих зграй, що відрізняється від відомої вве денням в процеси пошуку і погоні елементів випадкового пошуку. Використання рандомізованої модифікації дозволило поліпшити точність визначення напрямку руху в режимі котячого пошуку і поліпшити глобальні властивості процедури в режимі погоні, що, в свою чергу, покращує якість кластеризації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Поняття обробки даних

Обробка даних – це перетворення даних у придатну та бажану форму для використання. Перетворення або «обробка» виконується з використанням попередньо визначеної послідовності операцій вручну або автоматично. Більшість обробки здійснюється за допомогою комп'ютерів, таким чином, відбувається автоматично. Вихідні або «оброблені» дані можна отримати в різних формах. Прикладами цих форм є зображення, графік, таблиця, векторний файл, аудіо, діаграми або будь-який інший бажаний формат. Отримана форма залежить від використовуваного програмного забезпечення або методу обробки даних. Коли ця процедура виконується, це називається автоматичною обробкою даних.

Обробка даних, в основному, синхронізує всі дані, введені в програмне забезпечення, щоб відфільтрувати з нього найбільш корисну інформацію. Це дуже важливе завдання для будь-якої компанії, оскільки воно допомагає їм отримати найбільш релевантний контент для подальшого використання. Кожен важливий сектор, будь то банки, школи, коледжі чи великі компанії, майже всі потребують такої обробки даних. Ця обробка виконується для того, щоб зберегти найбільш придатну інформацію в їхніх системах для подальшого використання. Обробка вручну займає дуже багато часу і вимагає залучення занадто великої кількості людей для цього.

Сьогодні люди залежать від потужних та ефективних програмних інструментів, які допомагають обробляти всі ці дані. Це допомагає їм досягти більшої точності та підвищити ефективність. При належній обробці даних можна сортувати все більше і більше інформації. Це допомагає отримати чіткіше уявлення про матерію та краще її зрозуміти. Це може призвести до кращої продуктивності та більшого прибутку для різних сфер бізнесу.

За допомогою належних алгоритмів і протоколів безпеки можна гарантувати, що введені дані та оброблена інформація будуть безпечними та надійно збережені без несанкціонованого доступу або змін. З належним чином обробленими даними дослідники можуть писати наукові матеріали та використовувати їх у навчальних цілях. Те ж саме можна застосувати для оцінки економічних і подібних сфер і факторів. У галузі охорони здоров’я оброблені дані можна використовувати для швидшого пошуку інформації та навіть для порятунку життя. Крім того, відомості про хворобу та записи методів лікування можуть зменшити витрати часу на пошук рішень і допомогти зменшити страждання пацієнтів [2], [3], [4], [5].

Обробка даних, щоб упорядкувати їх за типом та інформацією, може заощадити багато місця, зайнятого даними, які не впорядковані й не зберігаються випадково. Оброблені дані також можуть допомогти всім співробітникам і працівникам легко їх зрозуміти. Вони можуть реалізувати це в роботі, яка в іншому випадку може зайняти більше часу і призвести до зниження продуктивності. Це може зашкодити інтересам підприємства чи організації.

Більшість компаній і сфер потребують даних для надання якісних послуг. Наявність колекції інформації про зібрані дані та їх наслідки є дуже важливим аспектом управління ними та забезпечення статистичної вірогідності. Особливо важливо це для служб, пов'язаних із фінансовими технологіями. Обумовлено це тим, що дані транзакції та деталі платежу мають належним чином зберігатися для легкого доступу клієнтам, а також посадовим особам компанії в разі потреби. Обробка не обмежується комп’ютерами може виконуватися вручну.

У той час як ручний варіант використовує потужність мозку та інтелект, електронні методи обробки даних можуть заощадити багато часу та забезпечити безперебійний робочий процес та забезпечити дотримання термінів. Точність вища з електронною обробкою. Один із важливих аспектів цього полягає в тому, щоб переконатися, що сформовані ідеї

зберігаються для майбутнього та спільного використання, щоб заощадити обчислювальну потужність та час.

Для баз структурованих даних розрізняють три основні типи логічних моделей даних залежно від характеру підтримуваних між ними зв'язків елементами даних – мережну, ієрархічну й реляційну. Ознаками класифікації у цих моделях є: ступінь твердості (фіксації) зв'язку, математичне подання структури моделі припустимих типів даних

Рисунок 1.1 ілюструє особливості кожної моделі даних. При зіставленні моделей варто пам'ятати, що всі вони теоретично еквівалентні. Еквівалентність моделей полягає в тому, що вони можуть бути зведені одна до одної шляхом формальних перетворень [7], [8].

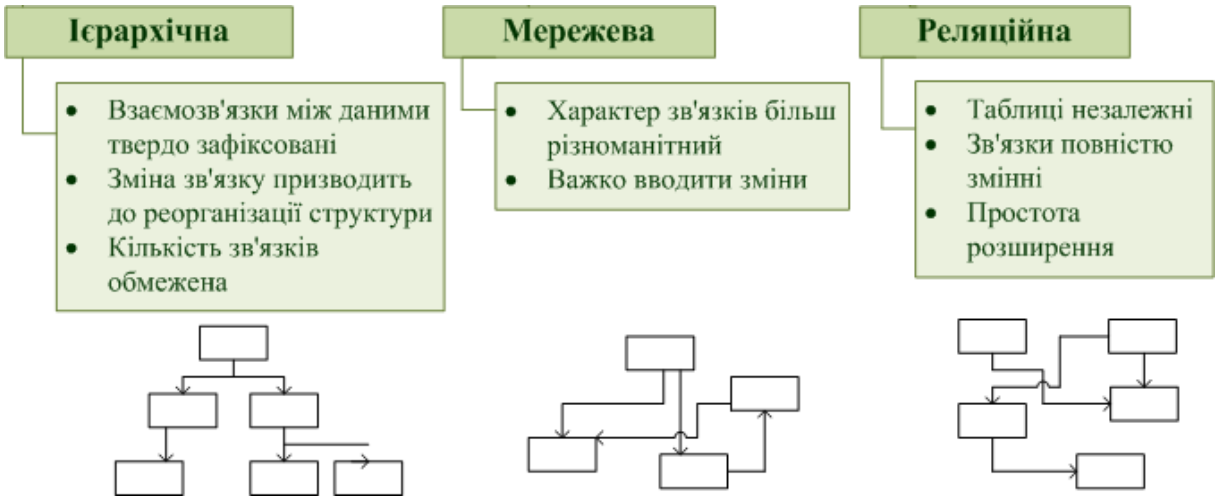


Рисунок 1.1 – Типи моделей даних

Для обробки даних використовуються в основному три методи: ручний, механічний та електронний. Особливостями цих методів є:

1) вручну – цьому методі дані обробляються вручну. Усі завдання обробки, такі як обчислення, сортування, фільтрація та логічні операції, виконуються вручну без використання будь-яких інструментів, електронних пристроїв чи програмного забезпечення для автоматизації;

2) механічний – у цьому методі дані не обробляються вручну, а обробляються за допомогою дуже простих електронних пристроїв і механічних пристроїв, наприклад, калькулятора та друкарської машинки;

3) електронний – це найшвидший метод обробки даних, а також сучасна технологія з сучасними необхідними функціями, такими як найвища надійність і точність. Цей метод досягається набором програм або програмного забезпечення, які запускаються на комп'ютерах.

1.2 Типи обробки даних

У основних областях наукової та комерційної обробки використовуються різні методи для застосування етапів обробки до даних. Три основні типи обробки даних, які ми збираємося обговорити, – це автоматична або ручна, пакетна обробка та обробка даних в реальному часі [10]:

– автоматична чи ручна обробка даних;

Це може здатися неможливим, але навіть сьогодні люди все ще використовують ручну обробку даних. Функції обробки даних бухгалтерського обліку можна виконувати з книги, опитування клієнтів можна збирати й обробляти вручну, і навіть обробка даних на основі електронних таблиць тепер вважається дещо ручною. У деяких складніших частинах обробки даних може знадобитися ручний компонент для інтуїтивного міркування.

Першою технологією, яка привела до розробки автоматизованих систем обробки даних, були перфокарти, які використовувалися при підрахунку населення.

Комп'ютери почали використовуватися корпораціями в 1970-х роках, коли почала розвиватися електронна обробка даних. Деякі з перших додатків для автоматизованої обробки даних у вигляді спеціалізованих баз

даних були розроблені для управління взаємовідносинами з клієнтами (CRM), щоб збільшити продажі.

Електронне управління даними набуло поширення з появою персонального комп'ютера в 1980-х роках. Електронні таблиці забезпечують просту електронну допомогу навіть для повсякденних функцій управління даними, таких як особистий бюджет і розподіл витрат.

Ця розробка в області автоматичної обробки даних у поєднанні з інструментами машинного навчання для оптимізації та покращення сервісу має на меті спростити доступ до даних і керування ними для кінцевих користувачів, без потреби у вузькоспеціалізованих спеціалістах з обробки даних.

– пакетна обробка;

Щоб заощадити обчислювальний час, до широкого використання архітектури розподілених систем, або навіть після нього, автономні комп'ютерні системи застосовують методи пакетної обробки. Це особливо корисно у фінансових програмах або там, де дані вимагають додаткових рівнів безпеки, наприклад, медичні записи [12].

Пакетна обробка завершує ряд процесів даних як пакет, спрощуючи окремі команди для виконання дій з кількома наборами даних. Це трохи схоже на порівняння комп'ютерної електронної таблиці з калькулятором. Обчислення можна застосувати з однією функцією, тобто одним кроком, до цілого стовпця або серії стовпців, даючи кілька результатів від однієї дії. Така ж концепція досягається при пакетній обробці даних. Серії дій або результатів можна досягти, застосовуючи функцію до цілої серії даних. Таким чином, час обробки комп'ютера значно менше.

Пакетна обробка може завершити чергу завдань без втручання людини, а системи даних можуть програмувати пріоритети для певних функцій або встановлювати час, коли пакетна обробка може бути завершена.

Банки зазвичай використовують цей процес для виконання транзакцій після завершення діяльності, коли комп’ютери більше не беруть участь у зборі даних і можуть бути призначені для функцій обробки.

– обробка даних у режимі реального часу;

Для комерційного використання багато великих програм обробки даних вимагають обробки в режимі реального часу. Тобто вони повинні отримувати результати з даних саме так, як це відбувається. Одним із застосувань цього, який більшість із нас може ідентифікувати, є відстеження тенденцій фондового ринку та валют. Дані потрібно оновлювати негайно, оскільки інвестори купують в режимі реального часу, а ціни оновлюються щохвилини. Дані про розклад авіакомпаній і програми для продажу квитків і GPS-відстеження в транспортних послугах мають подібні потреби для оновлення в режимі реального часу.

– потокова обробка;

Найпоширенішою технологією, яка використовується в обробці в реальному часі, є потокова обробка. Аналітика даних витягується безпосередньо з потоку, тобто з джерела. Якщо дані використовуються для висновків без завантаження та перетворення, процес відбувається набагато швидше.

– віртуалізація даних;

Методи віртуалізації даних є ще одним важливим розвитком в обробці даних в реальному часі, де дані залишаються у вихідній формі, єдина інформація витягується для потреб обробки даних. Принадність віртуалізації даних полягає в тому, що трансформація не потрібна, вона не виконується, тому кількість помилок зменшується.

Віртуалізація даних і потокова обробка означають, що аналітику даних можна отримувати в режимі реального часу набагато швидше, що приносить користь багатьом технічним і фінансовим додаткам, скорочуючи час обробки та кількість помилок [13].

Крім цих популярних методів обробки даних, є ще три методи обробки, які згадуються нижче:

– онлайн-обробка;

Ця техніка обробки даних є похідною від автоматичної обробки даних. Ця техніка зараз відома як негайна або нерегулярна обробка доступу. Згідно з цією технікою, діяльність фреймворком готується під час роботи/обробки. Його можна легко переглянути за допомогою постійної підготовки наборів даних. Цей метод обробки підкреслює швидкий внесок обміну даними та підключається безпосередньо до баз даних.

– розподіл часу.

Цей вид обробки даних повністю залежить від часу. При цьому один блок обробки даних використовується кількома користувачами. Кожному користувачеві призначаються встановлені таймінги, за якими він повинен працювати на одному процесорному блоці.

Інтервали розділені на сегменти, а отже, для користувачів, тому не відбувається згортання часу, що робить її системою з багатьма доступом. Ця техніка обробки також широко використовується і в основному розважається в стартапах.

1.3 Методи інтелектуального аналізу даних

Інтелектуальний аналіз даних шукає закономірності у величезних сховищах даних. Цей процес приносить корисні шляхи, і таким чином ми можемо робити висновки щодо даних. Це також генерує нову інформацію про дані, якими ми вже володіємо. Методи включають моделі відстеження, класифікацію, асоціацію, виявлення викидів, кластеризацію, регресію та передбачення. Розпізнати закономірності легко, оскільки в наведених даних може статися раптова зміна. Ми зібрали та класифікували дані на основі різних розділів для аналізу за категоріями.

Інтелектуальний аналіз даних – це процес вилучення корисної інформації чи знань із величезної кількості даних (або великих даних). Розрив між даними та споживанням було зменшено за допомогою різних інструментів аналізу даних[15].

Його можна виконувати в різних базах даних та інформаційних сховищах, таких як реляційні бази даних, сховища даних, транзакційні бази даних, потоки даних та багато іншого.

Існує багато методів, що використовуються для інтелекту даних, але найважливішим кроком є вибір з них відповідної форми відповідно до бізнесу або формулювання проблеми. Ці методи допомагають передбачати майбутнє, а потім приймати відповідні рішення. Вони також допомагають аналізувати ринкові тенденції та збільшувати дохід компанії.

Деякі методи:

– асоціація;
– класифікація;
– кластерний аналіз;
– прогнозування;
– послідовні шаблони;
– дерева рішень;
– аналіз аномалій;
– нейронна мережа.

Асоціація.

Він використовується для пошуку кореляції між двома або більше елементами шляхом виявлення прихованого шаблону в наборі даних і, отже, також називається аналізом відносин. Цей метод використовується в аналізі ринкового кошика для прогнозування поведінки клієнта [17], [18], [19].

Припустимо, менеджер з маркетингу супермаркету хоче визначити, які продукти часто купують разом. Як приклад: купує (x,” beer”) -> купує(x,“chips”) [підтримка = 1%, впевненість = 50%]

Тут x представляє клієнта, який купує разом пиво та чіпси. Впевненість показує впевненість, що якщо клієнт купує пиво, є 50% шанс, що покупець також прийме чіпси. Підтримка означає, що 1% усіх аналізованих транзакцій показали, що пиво та чіпси купувалися разом.

Можна розглянути багато подібних прикладів, як-от хліб з маслом або комп'ютер і програмне забезпечення. Існує два типи Правил асоціації:

– правило одновимірної асоціації: ці правила містять один атрибут, який повторюється;

– правило багатовимірної асоціації: ці правила містять кілька повторюваних атрибутів.

Класифікація.

Цей метод аналізу даних використовується для розрізнення елементів у наборах даних на класи або групи. Це допомагає точно передбачити поведінку сутностей у групі. Це двоетапний процес:

1) етап навчання (фаза навчання): на цьому алгоритмі класифікації будують класифікатор, аналізуючи навчальний набір;

2) етап класифікації: дані тестування використовуються для оцінки точності та точності правил класифікації [18].

Наприклад, банківська компанія використовує для визначення претендентів на позику з низьким, середнім або високим кредитним ризиком. Аналогічно, медичний дослідник аналізує дані про рак, щоб передбачити, які ліки призначити пацієнту.

Кластерний аналіз [19].

Кластеризація групує дані на основі подібності даних.

Кластеризація майже схожа на класифікацію, але в цьому кластері вносяться в залежності від подібності елементи даних. Різні групи мають різнорідні або не пов'язані між собою об'єкти. Її також називають сегментацією даних, оскільки вона поділяє величезні набори даних на групи відповідно до подібності.

Використовуються різні методи кластеризації.

Ієрархічні агломеративні методи на основі сітки:

– методи поділу;

– методи на основі моделі;

– методи на основі щільності.

Тут також можна розглянути подібний приклад претендентів на кредит. Деякі відмінності між класифікацією та кластеризацією зображені на рисунку 1.2.

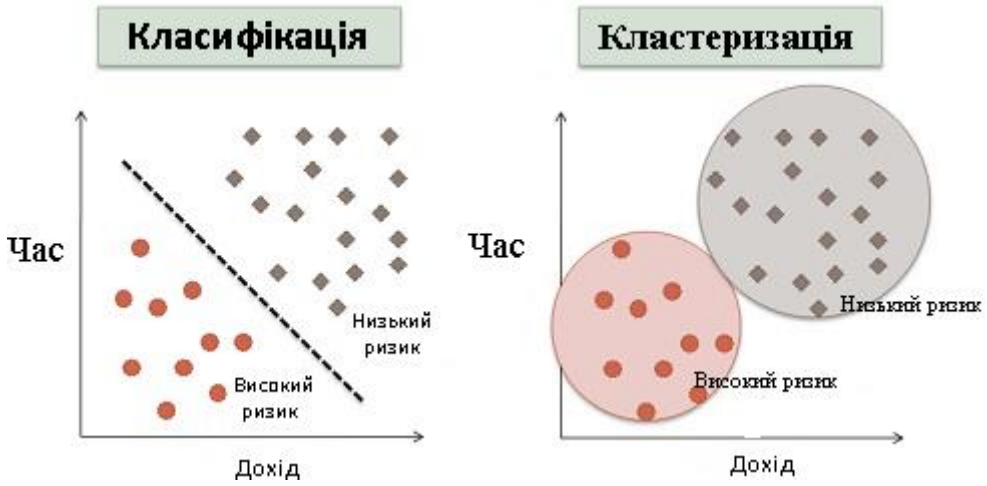


Рисунок 1.2 – Відмінності між класифікацією та кластеризацією

Передбачення.

Цей метод використовується для прогнозування майбутнього на основі минулих і теперішніх тенденцій або набору даних. Прогнозування в основному використовується для поєднання інших методів видобутку, таких як класифікація, узгодження шаблонів, аналіз тенденцій та зв’язок.

Наприклад, якщо менеджер з продажу хотів би передбачити суму доходу, який буде генерувати кожен товар на основі даних минулих продажів. Він моделює безперервну функцію, яка вказує на відсутні числові значення даних.

Регресійний аналіз є найкращим вибором для виконання передбачення. Його можна використовувати для встановлення зв’язку між незалежними змінними та залежними змінними.

Послідовні шаблони або відстеження шаблонів.

Цей метод використовується для виявлення закономірностей, які часто виникають протягом певного періоду часу. Наприклад, менеджер з продажу одягу бачить, що продажі курток, здається, збільшуються безпосередньо перед зимовим сезоном, або продажі в пекарні збільшуються під час Різдва чи Нового року.

Дерево рішень.

Дерево рішень – це деревовидна структура (як випливає з її назви), де кожен внутрішній вузол представляє тест на атрибут, відділення позначає результат тесту, термінальні вузли містять мітку класу.

Найвищий вузол – це кореневий вузол, який має просте запитання, яке має дві чи більше відповідей. Відповідно, дерево росте, і формується блок-схема.

Аналіз викидів або аналіз аномалій.

Цей метод визначає елементи даних, які не відповідають очікуваному шаблону або очікуваній поведінці. Ці несподівані елементи даних розглядаються як викиди або шум. Вони корисні в багатьох областях, до прикладу виявлення шахрайства з кредитними картками, виявлення вторгнень, виявлення неполадок тощо. Це також називається Outlier Mining .

Наприклад, припустимо, що наведений нижче графік побудовано з використанням деяких наборів даних у нашій базі даних.

Таким чином, буде проведена лінія, яка найкраще підходить. Точки, що лежать поруч із лінією, показують очікувану поведінку, тоді як кінець, далекий від лінії, є викидом. Це допоможе виявити аномалії та вжити відповідних заходів.

Нейронна мережа [20].

Цей метод або модель заснована на біологічних нейронних мережах. Це сукупність нейронів, подібних до процесорних блоків із зваженими зв’язками між ними. Вони використовуються для моделювання взаємозв'язку між входами та виходами. Він використовується для класифікації, регресійного аналізу, обробки даних тощо. Ця техніка працює на трьох основах:

1) модель;
2) алгоритм навчання (контрольований чи неконтрольований);
3) функція активації.

1.4 Поняття глибинного навчання

Штучний інтелект (ШІ) як інтелект, продемонстрований машинами, є ефективним підходом до розуміння людського навчання та формування міркувань [21]. У 1950 році тест Тьюринга був запропонований як задовільне пояснення того, як комп’ютер може відтворити когнітивні міркування людини. Як область досліджень, ЇЇ поділяють на більш конкретні напрямки. Наприклад: обробка природної мови (Natural Language Processing – NLP) [23] може покращити якість написання в різних програмах. Найбільш класичним підрозділом НЛП є машинний переклад, який розуміється як переклад між мовами. Алгоритми машинного перекладу сприяли появі різних додатків, які вивчають граматичну структуру та орфографічні помилки. Щобільше, набір слів і лексики, що стосуються теми матеріалу, автоматично використовуються як основне джерело, коли комп’ютер пропонує зміну для автора чи редактора [24].

Глибинне навчання (Deep Learning) – це підмножина машинного навчання, тобто на основі штучної нейронної мережі та навчання репрезентації, оскільки воно здатне реалізувати функцію, яка використовується для імітації функціональності мозку шляхом створення

шаблонів та обробки даних. Глибоке навчання також використовується для прийняття рішень у таких галузях, як безпілотні автомобілі (для виявлення пішоходів, вуличних ліхтарів, інших автомобілів тощо), розпізнавання мовлення, аналізу зображень (наприклад, виявлення раку в крові та пухлинах) тощо [22], [23], [24], [25].

Комп'ютерні програми, які використовують глибоке навчання, що навчається ідентифікувати собаку. Кожен алгоритм в ієрархії застосовує нелінійне перетворення до свого входу і використовує те, що він дізнався, для створення статистичної моделі як вихідну інформацію. Ітерації продовжуються, поки вихід не досягне прийнятного рівня точності. Кількість шарів обробки, через які повинні проходити дані, – це те, що надихнуло мітку deep.

У традиційному машинному навчанні процес навчання контролюється, і програміст повинен бути надзвичайно конкретним, повідомляючи комп'ютеру, які типи речей він повинен шукати, щоб вирішити, містить зображення об'єкт чи не містить його. Це трудомісткий процес, який називається вилученням функцій, і рівень успіху комп'ютера повністю залежить від здатності програміста точно визначити набір функцій для об'єкта. Перевага глибокого навчання полягає в тому, що програма створює набір функцій сама без нагляду. Навчання без нагляду не тільки швидше, але і, як правило, точніше.

Спочатку комп'ютерна програма може бути забезпечена навчальними даними – набором зображень, для яких людина позначила кожне зображення об'єкту чи ні метатегами. Програма використовує інформацію, яку вона отримує з даних дресування, щоб створити набір функцій для об'єкта та побудувати прогнозну модель. У цьому випадку модель, яку спочатку створює комп'ютер, може передбачити, що все на зображенні, що має вказані параметри, має бути позначено як бажаний об'єкт. З кожною ітерацією прогнозна модель стає складнішою та точнішою [25].

Комп’ютерній програмі, яка використовує алгоритми глибокого навчання, можна показати навчальний набір і відсортувати мільйони зображень, точно визначивши, на яких зображеннях є об’єкт за кілька хвилин [26].

Щоб досягти прийнятного рівня точності, програми глибокого навчання вимагають доступу до величезної кількості навчальних даних і потужності обробки, жодне з яких не було легко доступним для програмістів до ери великих даних і хмарних обчислень. Оскільки програмування глибокого навчання може створювати складні статистичні моделі безпосередньо з власних ітераційних результатів, воно здатне створювати точні прогнозні моделі з великої кількості немаркованих, неструктурованих даних.

Deep Learning швидко змінює навколишній світ, роблячи надзвичайні прогнози в таких областях, як безпілотні автомобілі (для виявлення пішоходів, вуличних ліхтарів, інших автомобілів тощо), виявлення токсичності для різних хімічних структур тощо.

Глибоке навчання має різноманітне застосування у фінансових сферах, комп’ютерному баченні, розпізнаванні звуку та мови, аналізі медичних зображень, техніках розробки ліків тощо.

Алгоритми глибокого навчання створюються шляхом з'єднання шарів між ними. Першим кроком є вхідний шар, за яким слідують прихований шар і вихідний шар. Кожен шар складається із взаємопов’язаних нейронів. Мережа споживає велику кількість вхідних даних для роботи з ними на кількох рівнях.

Щоб створити модель глибокого навчання, необхідно виконати наступні кроки:

1) розуміння проблеми;

2) ідентифікувати дані;

3) вибрати алгоритм;

4) тренувати модель;

5) тестувати модель.

Навчання відбувається у два етапи:

Етап 1 – застосування нелінійного перетворення вхідних даних і створення статистичної моделі як вихідної.

Етап 2 –покращети модель за допомогою похідного методу.

Ці дві фази операцій відомі як ітерації. Нейронні мережі повторюють два кроки, доки не буде отримано бажаний результат і точність:

– навчання мереж: щоб навчити мережу даних, ми збираємо велику кількість даних і розробляємо модель, яка вивчатиме особливості. Але при дуже великій кількості даних процес відбувається повільніше;

– transfer Learning в основному налаштовує попередньо навчену модель, а потім виконується нове завдання. У цьому процесі час обчислень стає меншим;

– вилучення ознак: після того, як всі шари навчені щодо особливостей об'єкта, з нього витягуються ознаки, і вихід прогнозується з точністю [24].

Переваги глибокого навчання:

– розв'язувати складні проблеми, такі як обробка аудіо, розпізнавання зображень тощо;

– зменшуйте потребу у виділенні функцій, автоматизовані завдання, де передбачення можна виконувати за менший час за допомогою Keras і Tensorflow;

– паралельні обчислення можна виконувати, таким чином зменшуючи накладні витрати;

– моделі можна навчати на величезній кількості даних, і модель стає кращою з більшою кількістю даних;

– високоякісні передбачення в порівнянні з людьми, невтомно тренуючись;

– працює з добре неструктурованими даними, такими як відеокліпи, документи, дані датчиків, дані вебкамери тощо;

Мережі глибокого навчання представлені нижче.

Попередньо навчена мережа без нагляду : це базова модель з 3 рівнями: вхідним, прихованим і вихідним. Мережу навчають реконструювати вхідні дані, а потім приховані шари навчаються на вхідних даних для збору інформації, і, нарешті, елементи витягуються із зображення.

Звичайна нейронна мережа: як стандартна нейронна мережа, вона має згортку всередині для виявлення країв і точного розпізнавання об'єктів.

Рекурентна нейронна мережа: у цій техніці вихідні дані попереднього етапу використовуються як вхідні дані для наступного або поточного етапу. RNN зберігає інформацію в контекстних вузлах, щоб дізнатися вхідні дані та отримати вихідні дані. Наприклад, щоб закінчити речення, нам потрібні слова. тобто, щоб передбачити наступне слово, потрібні попередні слова, які потрібно запам'ятати. RNN в основному вирішує цей тип проблеми.

Рекурсивні нейронні мережі: це ієрархічна модель, де вхід є деревоподібною структурою. Такий тип мережі створюється шляхом застосування того самого набору ваг до збірки входів[26].

1.5 Методи Deep Learning

Техніки Deep Learning – це методи, які використовуються для імітації функціональності людського мозку шляхом створення моделей, які використовуються в класифікаціях з тексту, зображень і звуків. Ці моделі складаються з кількох шарів прихованого шару, також відомого як нейронна мережа, яка може витягувати функції з даних, кожен шар цих нейронних мереж, починаючи з крайнього лівого шару до крайнього правого шару, витягує низькорівневі характеристики, такі як край і згодом робити точні прогнози.

Методи глибокого навчання використовують нейронні мережі. Тому їх часто називають глибокими нейронними мережами. Глибокі або приховані нейронні мережі мають кілька прихованих шарів глибоких мереж.

Глибоке навчання навчає ШІ передбачати вихідні дані за допомогою певних вхідних даних або прихованих мережевих рівнів. Мережі навчаються за допомогою великих позначених наборів даних і вивчають функції з самих даних. Навчання з наглядом і без нагляду працює для навчання даних і створення функцій.

Існує кілька різних структур з незліченними цілями, і їх функціонування також залежить від структури, і всі вони засновані на нейронних мережах (рисунок 1.3) [25], [26], [27].

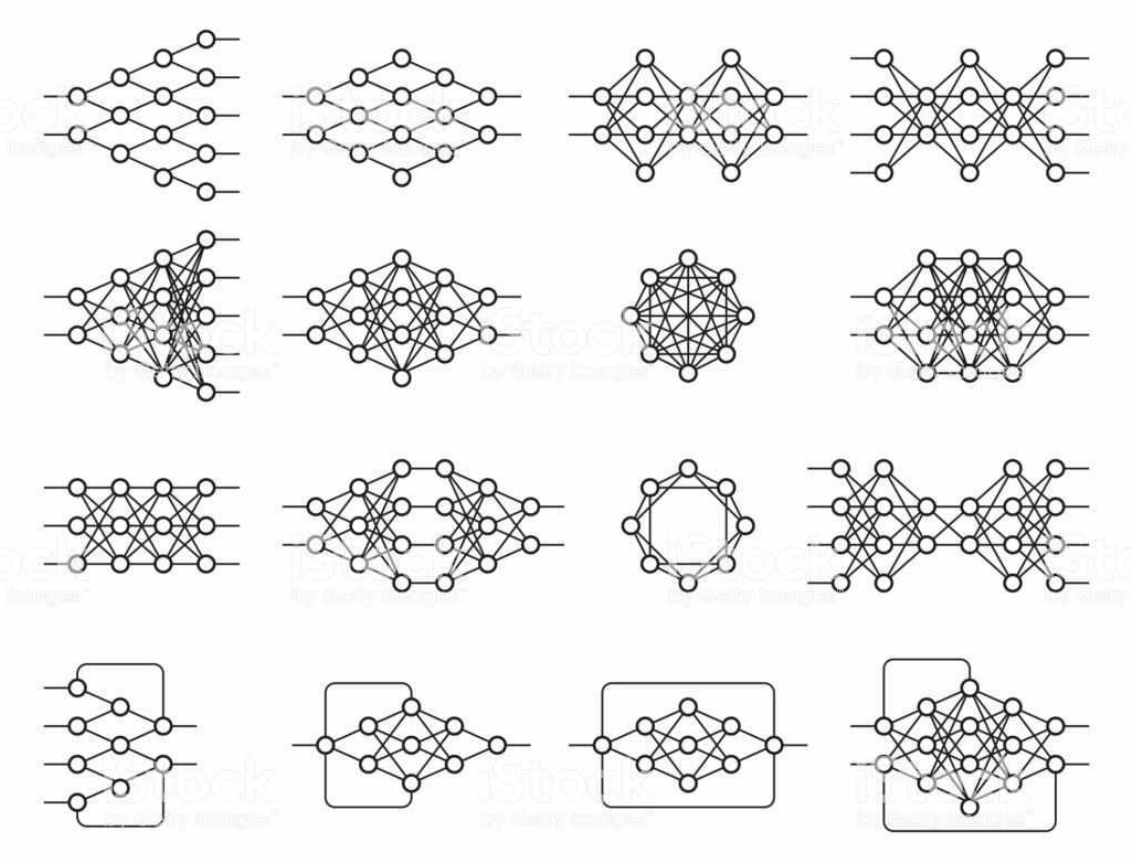


Рисунок 1.3 – Приклади структур нейронних мереж

Звичайні нейронні мережі (CNN), одна з найпопулярніших нейронних мереж, об'єднує функції, отримані з вхідних даних, і використовує двовимірні згорткові шари, щоб зробити його придатним для обробки 2D-даних, наприклад зображень. Таким чином, вона зменшує використання

ручного вилучення функцій у цьому випадку. Він безпосередньо витягує необхідні ознаки із зображень для класифікації. Завдяки цій функції автоматизації CNN є переважно точним і надійним алгоритмом машинного навчання. Кожен CNN вивчає особливості зображень із прихованого шару, і ці приховані шари збільшують складність вивчених зображень.

Важливою частиною є навчання штучного інтелекту або нейронних мереж. Для цього ми надаємо вхідні дані з набору даних і, нарешті, робимо порівняння вихідних даних за допомогою виводу набору даних. Якщо штучного інтелекту не навчений, результат може бути неправильним.

Методи глибокого навчання створюють моделі з кількома прихованими шарами нейронних мереж, щоб робити точні прогнози. Деякі з програм глибокого навчання – це безпілотні автомобілі, віртуальні помічники, виявлення новин про шахрайство, автоматизовані ігри, візуальне розпізнавання, мовний переклад та багато іншого.

Існують різні типи методів глибокого навчання, кожна з яких використовується з іншою метою. Нижче наведено 7 найкращих методів глибокого навчання:

– трансферне навчання – це процес удосконалення раніше навченої машини або моделі для виконання нових і більш конкретних завдань. Цей метод корисний, оскільки потребує набагато меншої кількості даних, ніж інші методи, і допомагає скоротити великий час обчислень.

– машини Больцмана: у цій моделі немає попередньо визначеного напрямку. Ця техніка глибокого навчання використовується для моніторингу системи, платформи бінарних рекомендацій та аналізу конкретних наборів даних. Його вузли розташовані по колу, і це унікальний метод глибокого навчання, який використовується для створення параметрів моделі. Вона значно відрізняється від інших мережевих моделей і також відома як стохастична;

– глибоке підкріплення. Цей алгоритм глибокого навчання має вхідний шар, вихідний шар та інші кілька прихованих шарів. Ця модель

спрямована на прогнозування майбутньої винагороди залежно від вхідних дій. Ця техніка глибокого навчання призначена для настільних ігор, самокерованих автомобілів, робототехніки та інших;

– зворотне поширення. Ця техніка була запущена в 1970 році. Вона також відома як зворотне поширення. Це контрольований алгоритм глибокого навчання, який використовується для навчання штучних нейронних мереж. Цей метод використовує техніку, яка називається градієнтним спуском;

– нормалізація партії; передбачається, що це одна з життєво важливих частин підготовки даних для методів глибокого навчання. Він був запущений у 2015 році і є одним із новітніх методів глибокого навчання. Ця техніка використовується для підвищення продуктивності, а також стабільності штучної нейронної мережі. Це також допомагає скоротити період навчання, необхідний для навчання глибоких нейронних мереж;

– стохастичний градієнтний спад – метод або Стохастичний градієнтний спуск або SGD також називають методом Роббінса Монро, оскільки він був винайдений американським математиком-статистиком Гербертом Роббінсом у 1951 році. Це також одна з основних технік революції ШІ, і, ймовірно, це також серед найбільш широко використовуваних методів глибокого навчання, які використовуються ентузіастами ШІ. Ця техніка використовує кілька випадково вибраних вибірок замість всього набору даних для кожної ітерації;

– ResNet, – 2015 році Каймін Він розробив ResNet, також відому як залишкова нейронна мережа. Більш глибокі нейронні мережі набагато складніші та складніші для навчання, і ось тут входить ResNet. Він використовується для покращення глибоких нейронних мереж і розбиття глибоких нейронних мереж на менші мережі. Вони додатково підключаються за допомогою пропуску або інших ярликів, щоб створити більшу мережу[28], [29], [30].

1.6 Поняття кластерного аналізу

Інформація стає все більш важливою та доступною для людей по всьому світу, розробляється все більше і більше методів науки про дані та машинного навчання. Модель кластерного аналізу, на перший погляд може здатися простою, але важливо зрозуміти, як працювати з величезними даними.

Кластерний аналіз або кластеризацію даних можна визначити як техніку машинного навчання без нагляду (не мічені дані), яка спрямована на пошук закономірностей (наприклад, багато підгруп, розмір кожної групи, загальні характеристики, згуртованість даних) під час збору зразків даних. і згрупуйте їх у подібні записи, використовуючи попередньо визначені міри відстані, як-от евклідова відстань тощо.

У кластерному аналізі спочатку розділити набір даних на групи, знайшовши схожість об’єктів у групі, а потім, якщо потрібно, присвоїти йому мітку. Основна перевага кластеризації полягає в тому, що вона намагається виділити корисні функції в наборі даних і використовує їх для розрізнення різних груп. Завдяки цьому він також пристосовується до змін.

Ексклюзивні проти невиключних: в ексклюзивному кластері ми матимемо точки даних лише в певному кластері, тоді як у невиключних точках даних може належати до двох або більше кластерів одночасно [27].

Нечіткі проти нечітких: на нечіткому рівні кластеризації точки даних належать до всіх кластерів із вагою від 0 до 1, і вся вага має дорівнювати 1. Це також відоме як ймовірнісна кластеризація. У нечіткому все навпаки, коли точка даних належить одному конкретному кластеру.

Часткове проти повного: під час часткового ми хотіли б об’єднати лише частину даних з різних бізнес-причин. Якість кластеризації залежить від методів і виявлення прихованих закономірностей.

Гетерогенні проти однорідних: у гетерогенних розмірів, форма та щільність кластера можуть змінюватися, але при неоднорідності необхідно переконатися, що кластер має однакову форму, розмір і щільність.

Кластерний аналіз широко використовується в різних програмах, таких як обробка зображень, нейро-наука, економіка, мережеві комунікації, медицина, рекомендаційні системи, сегментація клієнтів тощо. Крім того, кластеризацію можна вважати початковим кроком під час роботи з новим набором даних для отримання інформації та розуміння розподілу даних. Кластерний аналіз також можна використовувати для зменшення розмірності (наприклад, PCA). Він також може служити як попередня обробка або проміжний крок для інших алгоритмів, таких як класифікація, передбачення та інші програми для аналізу даних [28], [29].

Важливість методів кластеризації:

– наявність методів кластеризації допомагає перезапустити процедуру локального пошуку та усунути неефективність;

– кластеризація допомагає визначити внутрішню структуру даних;

– кластеризація допомагає зрозуміти природне групування в наборі даних, їх мета полягає в тому, щоб мати сенс розділити дані на певну групу логічних груп;

– відіграє широку роль у таких програмах, як маркетингові економічні дослідження та веб-журнали для визначення мір подібності, обробки зображень та просторових досліджень;

На рисунку 1.4 представлено алгоритми кластеризації.

Нечітка кластеризація дуже потужний у порівнянні з традиційним жорсткою пороговою кластеризацією, де кожній точці призначається чітка, точна мітка. Цей алгоритм працює шляхом призначення членства кожній точці даних, що відповідає кожному центру кластера, на основі відстані між центром кластера та точкою даних. Чим більше дані ближче до кластерного центру, тим більше їх приналежність до конкретного кластерного центру.

Очевидно, що підсумок приналежності кожної точки даних має дорівнювати одиниці [29], [30].

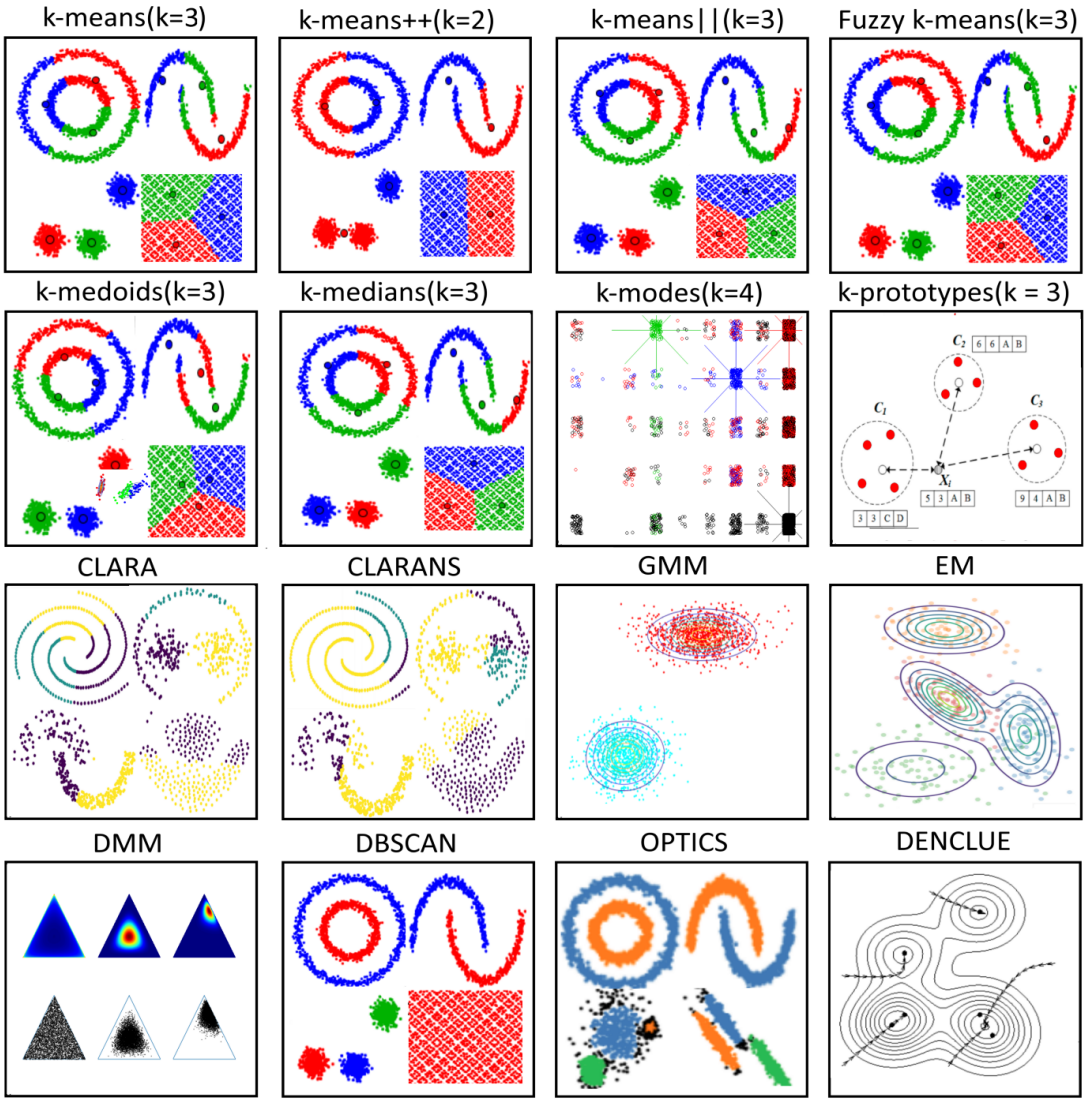


Рисунок 1.4 – Алгоритми кластерного аналізу

Це неконтрольований алгоритм кластеризації, який дозволяє нам будувати нечіткий розділ із даних. Алгоритм залежить від параметра m, який відповідає ступеню нечіткості рішення. Великі значення m розмиють класи, і всі елементи, як правило, належать до всіх кластерів.

Розв'язки оптимізаційної задачі залежать від параметра m. Тобто різний вибір m зазвичай призведе до різних розділів. Нечітка кластеризація

розглядається як кластеризація, в якій кожен елемент має ймовірність належності до кожного кластера. Іншими словами, кожен елемент має набір коефіцієнтів належності, що відповідають ступеню перебування в даному кластері.

Це відрізняється від k-середніх і k-медоїдних кластеризації, коли кожен об’єкт впливає точно на один кластер. Кластеризація K-середніх і k-медоїдів відома як жорстка або нечітка кластеризація.

У нечіткій кластеризації точки, близькі до центру кластера, можуть бути в кластері більшою мірою, ніж точки на краю кластера. Ступінь, до якої елемент належить даному кластеру, – це числове значення, яке змінюється від 0 до 1 [30].

Алгоритм нечітких c-середніх (FCM) є одним з найбільш широко використовуваних алгоритмів нечіткої кластеризації. Центроїд кластера обчислюється як середнє всіх точок, зважене за ступенем їхньої належності до кластера.

Плюси:

– дає найкращий результат для набору даних, що перекривається, і порівняно краще, ніж алгоритм k-середніх;

– на відміну від k-середніх, де точка даних повинна належати виключно одному кластерному центру, тут точці даних призначається членство в кожному центрі кластера, в результаті чого точка даних може належати більш ніж одному кластерному центру.

Мінуси алгоритму нечітких c-середніх:

– апріорне уточнення кількості кластерів;

– при меншому значенні ми отримуємо кращий результат, але шляхом більшої кількості ітерацій;

– евклідова відстань може неоднаково впливати на основні фактори;

– продуктивність алгоритму FCM залежить від вибору початкового центру кластера та початкового значення членства;

Завдання кластеризації масивів багатовимірних спостережень часто зустрічається у безлічі реальних додатків, а для її вирішення на сьогодні розроблено велику кількість методів, процедур, алгоритмів [21], [32] від суто емпіричних до строго математичних. У найбільш спільній постановці передбачається, що є група N об'єктів, що описуються n-мірними векторами-ознаками $x(k) \in Rn$, $k = 1, 2, ..., N$, яку необхідно розбити на p кластерів, причому це число може бути наперед невідомо, тобто $1 < p < N$. Зрозуміло, що така велика кількість можливих підходів до розв'язання задачі пов'язана з тим, що принципово не існує універсального алгоритму, придатного для ефективного використання в всіх ситуаціях, що виникають у реальних задачах.

Особливу групу методів кластеризації утворюють алгоритми, призначені для обробки інформації, що зберігається у надвеликих базах даних (VLDB) [28], де першому плані виходять швидкодія та простота чисельної реалізації. У цій ситуації як досить ефективні показали себе методи кластеризації, засновані на щільності розподілу даних, при цьому поняття щільності, що застосовується тут за змістом близько до щільності розподілу, що використовується в теорії ймовірностей та математичної статистики.

Саме методи, засновані на щільності, дозволяють формувати кластери довільної форми за умов, коли оброблювані дані спотворені обуреннями, а саме число кластерів заздалегідь невідоме. У рамках «щільні» підходу під кластерами розуміють області в n-мірному просторі ознак з високим рівнем концентрації даних. Ці області розділені ділянками з низькою щільністю і тут розташовуються обурення.

Таким чином, алгоритми, засновані на понятті щільності, в процесі обробки даних формують області довільної форми, де дані густо сконцентровані.

1.7 Постановка задачі дослідження

Інновацій завжди виникали на місці злиття різноманітних джерел знань. Там, де вчені експериментують з новими ідеями, щоб розширити наукові кордони.

Серед методів визначальне місце займають методи інтелектуального аналізу та їх комбінації, і для їх реалізації використовуються спеціальні технології та алгоритми, а їх використання у практичній діяльності орієнтоване на синергетичний ефект.

Зі збільшенням обсягу даних, а більшість даних є неструктурованим (зображення, відео, аудіо тощо), алгоритми глибинного навчання відіграють важливу роль в епоху революції даних.

Завдання кластеризації вирішується основуючись на парадигмі самонавчання (навчання без вчителя), тобто в умовах дефіциту інформації, відсутності сформоваої вибірки.

Мета роботи: запропонувати метод з використанням еволюційної оптимізації, який був би позбавлений недоліків традиційних підходів до кластеризації даних.

Для досягнення поставленої мети в роботі вирішуються такі задач:

– виконання аналізу традиційних підходів до обробки даних;

– застосування еволюційного ройового алгоритму, тобто покращити задачі оптимізації.

2 КЛАСИФІКАЦІЯ МАСИВІВ ДАНИХ НА ОСНОВІ КОМБІНОВАНОЇ ОПТИМІЗАЦІЇ ФУНКЦІЙ

2.1 Особливості векторних та матричних даних

Під час дослідницького аналізу даних дослідники намагаються виявити особливості з необроблених даних. Вони можуть почати з якісного дослідження, переглядаючи візуалізації та застосовуючи свої знання в області, щоб вивести ідею, яка може перетворити спостереження на вектори ознак. Наприклад, вектор ознак в аналізі даних представляє прихований шаблон у великих наборах даних, таких як сигнали купівлі або продажу акції з історичних даних про ціну та обсяги торгівлі.

У сфері обробки природної мови процес поділу речень на окремі сутності відомий як лексема. Наприклад, дослідники можуть розглядати кожне слово або фонему як унікальний маркер для створення векторів ознак для подальшого аналізу та експериментів.

Вектори ознак представляють функції, які використовуються моделями машинного навчання, у багатовимірних числових значеннях. Оскільки моделі машинного навчання можуть мати справу лише з числовими значеннями, перетворення будь-яких необхідних ознак у вектори ознак має вирішальне значення. Тут ми обговорюємо вектори функцій у різних випадках використання. Ми також пояснюємо труднощі у створенні векторів ознак і керування ними.

Вектор ознак – це впорядкований список числових властивостей спостережуваних явищ. Він представляє вхідні функції для моделі машинного навчання, яка робить прогноз [30].

Люди можуть аналізувати якісні дані, щоб прийняти рішення. Наприклад, дивитися на хмарне небо, відчуваємо вологий вітерець і вирішуємо взяти парасольку, виходячи на вулицю. Наші п’ять органів почуттів можуть перетворювати зовнішні подразники в нервову активність

нашого мозку, обробляючи численні вхідні сигнали, оскільки вони відбуваються в певному порядку.

Однак моделі машинного навчання можуть мати справу лише з кількісними даними. Таким чином, завжди потрібно перетворювати особливості спостережуваних явищ у числові значення та вводити їх у модель машинного навчання в тому ж порядку. Коротко кажучи, ми повинні представляти особливості у векторах ознак. Існують різні типи функцій і методів, які корисні для побудови вектору ознак.

Вектори функцій для класифікації тексту:

– модель мішка слів являє собою документ у векторному форматі, де кожен елемент має кількість зустрічей конкретного слова. Хоча кожен індекс у векторі відповідає слову, модель машинного навчання розглядає його як список числових значень для прогнозування;

– Tf-idf (термін частота зворотна частота документа) вимірює важливість кожного слова в документі. Розрахунок містить ділення кількості зустрічей слова на кількість документів, що містять те саме слово. Якщо в одному документі певне слово використовується дуже часто, а в інших документах – ні, то це слово має бути важливим у цьому документі;

– одночасне кодування – це вектор з нулями скрізь, за винятком одного індексу, де значення дорівнює одиниці, що унікально представляє кожне слово. На відміну від цього, формат word2vec (від слова до вектору) використовує розподілене подання, що означає, що елементи вектора часто відрізняються від нуля. Використовує набагато менше пам’яті, ніж одноразове кодування, і навіть дозволяє операціям лінійної алгебри вимірювати подібність слів. Цей тип вектору слів зазвичай називають вектором вбудовування слів[29].

Вектор може представляти багато властивостей моделей купівельної активності користувачів, наприклад час покупки, категорію продукту, ціну, ідентифікатор магазину, вік тощо. Системи рекомендацій виконують

матричні операції над великою кількістю даних клієнтів, представлених у векторах ознак.

Під час дослідницького аналізу даних дослідники намагаються виявити особливості з необроблених даних. Наприклад, вектор ознак у аналізі даних представляє прихований шаблон у великих наборах даних, таких як сигнали купівлі-продажу акції з історичних даних про ціну та обсяги торгівлі.

У сфері обробки природної мови процес поділу речень на окремі сутності відомий як лексема. Наприклад, дослідники можуть розглядати кожне слово або фонему як унікальний маркер для створення векторів ознак для подальшого аналізу та експериментів.

У комп'ютерному баченні колірна схема RGB не є єдиним способом представлення пікселів зображення. Альтернативою йому є також HSL (відтінок, насиченість, яскравість) і HSV (відтінок, насиченість, значення). Іноді з практикою навіть використовують монохромну схему, щоб зменшити шуми від кольорових зображень.

Зрештою, дослідники досліджують різні вектори ознак, щоб оцінити ефективність своїх прогнозних моделей. Як тільки дизайн функцій готовий, можна переходити до наступного етапу[32].

Вектори є корисним способом зберігання та маніпулювання даними. Однак може бути корисно створювати більш складні структури даних – і саме тут потрібно вводити матриці.

Матриці – є структурами даних, які дозволяють організувати числа. Вони являють собою квадратні або прямокутні масиви, що містять значення, організовані у двох вимірах: у вигляді рядків і стовпців. Зазвичай термін матриця в контексті математики та двовимірний масив.

У контексті матриць термін розмірність відрізняється від розмірів геометричного представлення векторів (розмірів простору). Коли ми говоримо, що матриця є двовимірним масивом, це означає, що в масиві є два напрямки рядки і стовпці.

Матриця A має два рядки і два стовпці, але ви можете уявити матриці будь-якої форми. У більш загальному вигляді, якщо матриця має m рядків і n стовпців і містить реальні значення.

Можна посилатися на записи матриці з назвою матриці без напівжирного шрифту (оскільки записи є скалярами), а потім індексом рядка та індексом стовпця, розділених комою в нижньому індексі. Наприклад, $A_{1,2}$ позначає запис у першому рядку та другому стовпці.

Вміння маніпулювати матрицями, що містять дані, є важливою навичкою для науковців даних. Перевірка форми даних важлива, щоб бути впевненим, що вони організовані так, як ви хочете. Також важливо знати форму даних, яка вам знадобиться для використання бібліотек, таких як Sklearn або Tensorflow.

Форма масиву показує кількість компонентів у кожному вимірі. Якщо матриця двовимірна (рядки та стовпці), потрібно два значення, щоб описати форму (кількість рядків і число стовпців у такому порядку).

Еквівалентна операція для матриць називається добутком матриці або множенням матриці . Він приймає дві матриці і повертає іншу матрицю. Це основна операція в лінійній алгебрі.

У машинному навчанні зазвичай мають справу з матрицями, які мають функції в якості стовпців. Тепер це може мета матриці – «діяти» на (вхідні) вектори, щоб дати (вихідні) вектори.

2.2 Кластеризація масивів спостережень

Задача кластеризації масивів спостережень довільної природи є невід’ємною частиною Data Mining, а у більш загальному випадку Data Science, а для її вирішення запропонована дуже велика кількість підходів, що відрізняються між собою як апріорними припущеннями що до фізичної

природи даних та задач, що вирішуються на їх основі, так і математичним апаратом, що використовується [14], [21], [27], [28], [34].

З обчислювальної точки зору найбільш простими є, серед яких слід відзначити процедуру k-середніх, що набрала дуже широкого розповсюдження для вирішення найрізноманітніших задач. Тут можна відзначити, що найбільш адекватним математичним апаратом для вирішення задач кластеризації є методи обчислювального інтелекту і, перш за все, штучні нейроні мережі, нечіткі системи, еволюційна оптимізація та так звані, гібридні системи обчислювального інтелекту, що об'єднують ці три напрямки. Сутність цього методу полягає в тому, що при попаданні у локальний екстремум, алгоритм реалізує інтенсивні випадкові стрибки (Jamps), що виводять процедуру з околу локального екстремум.

Всі ці методи призначені для роботи у пакетному режимі, тобто апріорі задана вибірка багаторазово опрацьовується у формі фіксованого масиву даних. Якщо ж дані надходять у послідовному он-лайн режимі, ці підходи є непрацездатними. В умовах, коли все більшого поширення набувають задачі, пов'язані з Big Data, викликає необхідність розробки нових підходів, пристосованих для нових умов.

Цікаво відзначити, що один з найпопулярніших нейронних мереж – самоорганізовані карти Кохонена фактично реалізують процедуру k-середніх, представлену у рекурентній формі. Як відомо самоорганізовні мапи Т.Кохонена, що навчаються у послідовному режимі, можуть реалізувати HCM-кластеризацію на основі правила самонавчання «Переможець отримує все»(WTA) та FCM-кластеризацію на основі правила «Переможець отримує більше» (WTM), якщо функція сусідства обирається у вигляді кошіану.

Нескладно додати у процедуру налаштування системних ваг нейронної мережі Т.Кохонена, що є за штучно градієнтними алгоритмами оптимізації, випадкових збурень, що надає алгоритму самонавчання в цілому властивостей глобального випадкового пошуку. Таким чином

реалізується но-лайн модифікація J-середніх, основною перевагою якої є простота чисельної реалізації та висока швидкість обробки даних, що надходять у послідовному режимі.

Нескладно додати у процедуру налаштування системних ваг нейронної мережі Т.Кохонена, що є за штучно градієнтними алгоритмами оптимізації, випадкових збурень, що надає алгоритму самонавчання в цілому властивостей глобального випадкового пошуку. Таким чином реалізується но-лайн модифікація J-середніх, основною перевагою якої є простота чисельної реалізації та висока швидкість обробки даних, що надходять у послідовному режимі [35].

Тут слід відзначити, що в загальному випадку вирішення задач кластеризації суттєво ускладняється, якщо вихідні вектори (у загальному випадку матриці) спостереження мають велику різноманітність, викривлені збуреннями та завалами, містять пропуски, самі вихідні масиви або занадто великі (Big Data) або занадто короткі, кластери можуть мати досить складну форму, а їх кількість апріорі невідома.

У цьому випадку найбільш ефективними (але й найбільш складними) є алгоритм, що базуються на аналізі щільностей розподілу даних, серед яких в якості одного найбільш «популярних» є DENCLUE [37] та його модифікації [38], що були запропоновані для вирішення задач кластеризації великих масивів векторних даних високої розмірності, при цьому класи, що формуються у процесі кластеризації, можуть мати будь яку складну форму.

DENCLUE застосовує метод оцінки щільності ядра для оцінки невизначеної функції щільності ймовірності випадкової величини, яка генерує вибірку даних. Оцінка базується на функції щільності ядра (наприклад, функції щільності по Гауссу), що представляє розподіл кожної точки даних. Потім оцінка щільності ядра для всіх попередніх функцій обчислюється шляхом їх підсумовування (або інтеграла).

Ядро – це математична функція, яка моделює вплив між точками даних та їхніми сусідами. Крім того, функція щільності ядра має такі властивості:

– невід’ємність: $K(x) \geq 0$;

– симетричний: $K(x) = K(-x)$;

– площа під ядром має дорівнювати одній одиниці: $K(x)=1$;

– зменшення: $K'(x) \leq 0$.

DENCLUE використовує концепцію атракторів щільності, які служать представниками для спостережень, навколо яких утворюються скупчення.

Існує два типи кластерів:

– кластери, визначені в центрі: формуються шляхом призначення щільності точок, притягнутих до заданого атрактора щільності;

– кластер довільної форми: формується шляхом злиття атракторів щільності з високою щільністю.

Алгоритм виглядає наступним чином:

1) оцініти загальну функцію щільності ядра простору даних, додавши функції щільності всіх точок даних;

2) кластери формуються шляхом ідентифікації атракторів щільності, які становлять локальні максимуми оціненої функції густини;

3) локальні максимуми обчислюються за допомогою алгоритму Hill-climbing з градієнтом функції оцінки щільності. Переваги:

– значно швидше, ніж DBSCAN;

– гнучкі для кластерів будь-якої форми;

– ефективно працює з будь-якими наборами даних. Недоліки:

– погано масштабується для набору даних великого розміру;

– залежить від кількох гіперпараметрів;

– працює лише з числовими даними.

При розв'язанні задач кластеризації завжди передбачається, що кожне багатовимірне спостереження описується n-мірним вектором x(k), а весь процес вирішення пов'язаний саме з векторними операціями. У ситуації, коли є велика колекція зображень, що підлягають кластеризації, кожне двовимірне зображення спочатку має бути піддане векторизації, далі вирішується завдання кластеризації, а її результат піддається девекторизації, що переводить векторний опис матричну форму [32].

Істотно спростити процес кластеризації масивів можна, не переводячи їх у векторну форму, а безпосередньо оперуючи з матрицями. В основі цих алгоритмів полягає пошук екстремумів-максимумів функції щільності розподілу даних у масиві що аналізується (багатоекстремальна оптимізація), при цьому ця функція формується, як суперпозиція ядерних (дзвонуватих) функцій, пов’язаних з кожним спостереженням. Фактично ця функція будується на основі вікон Парзена [28] та оцінок Надарая-Ватсона (вважатимемо, що залежність константна і тоді можна аналітично обчислити чому має дорівнювати ця константа, таким чином, складність моделі перенесена на ваги. Ваги використовують ядерне згладжування [37]).

З обчислювальної точки зору задача кластеризації перетворюється у проблему пошуку локальних екстремумів багатоекстримальної функції векторного аргументу щільності за допомогою градієнтних процедур, які багатократно запускаються з різних точок вихідного масиву даних. Зрозуміло, що це займає досить багато часу, оскільки апріорі навіть невідомо скільки ж екстремумів має сформована функція щільності.

Пришвидшити процес пошуку цих екстремумів можна, скориставшись ідеями еволюційної оптимізації, що включає в себе алгоритми, інспіровані природою, ройові алгоритми, популяційні алгоритми, тощо [36], [37], [38]. При цьому пошук ведеться одночасно групою, що діють або незалежно, або у взаємодії, що дозволяє суттєво пришвидшити процес пошуку екстремумів, кожен з яких «відповідає» тому або іншому кластеру, що формується.

2.3 Формування функцій щільності розподілу даних у масиві, що полягає кластеризації

Вихідною інформацією для вирішення задачі кластеризації традиційно є масив векторів-спостережень X= {x(1),x(2),…x(k),…,x(N)}, x(k) ={x_1(k)}ϵ R^m, при цьому дані попередньо відцентровано на гіперкуб так, що x(k) = {$x_{i1,i2}$ (k)}ϵ R^{N1*N2}. Така ситуація може виникати у випадку обробки масивів зображень.

Основними поняттями, на яких будується DENCLUE є функція щільності та атрактори щільності, що за суттю є локальними екстремумами функції щільності.

У загальному випадку функція впливу для будь-якого векторного спостереження x(•) з вихідного масиву X є ядерною (дзвонувальною) функцією $f^{x(\cdot)}(x)$, при цьому найбільш популярною є традиційна гуасівська функція.

$$f_G^{\mathrm{x}(\bullet)}(x) = \exp\left(-\frac{d^2(x,\mathrm{x}(\bullet)}{2G^2}\right) = \exp\left(-\frac{\|x-\mathrm{x}(\bullet)\|^2}{2G^2}\right) \qquad (2.1)$$

де, $d^2(x,\mathrm{x}(\bullet)$ – евклідова відстань, G^2–параметр ширини функції впливу завдяки простоті обчислення її похідних.

У матричному випадку замість евклідової можна використати матричну Флобейніуса, при цьому функція впливу набуває вигляду.

$$f_G^{\mathrm{x}(\bullet)}(x) = \exp\left(-\frac{d^2(x,\mathrm{x}(\bullet)}{2G^2}\right) = \exp\left(-\frac{Tr(x-\mathrm{x}(\bullet))(\mathrm{x}-\mathrm{x}(\bullet)^T}{2G^2}\right) \qquad (2.2)$$

де Tr(•) – символ сліду матриці

Нескладно бачити, що (2.2) є узагальненням (2.1).

На основі функцій впливу формується функція щільності розподілу даних X у вигляді:

$$f_{(x)}^{x} = \sum_{k=1}^{N} f\left(x_1 x(k)\right) \tag{2.3}$$

Що по суті є оцінкою Надарая-Ватоона. Нескладно бачити, що функція $f^{x}(x)$ Може приймати значеня в інтервалі $1 \leq f^{x}(x) \leq N$, при цьому крайні значення цього інтервалу приймаються коли вибірка містить лише 1 спостереження або усі N спостережень співпадають, тобто існує лише один кластер – вироджена ситуація.

Для знаходження m>1 кластерів необхідно ввести у розгляд деякий поріг $\xi > 1$, що дозволяє формувати дійсно значущі кластери, відстежуючі аномальні спостереження та класи що містять занадто мало даних.

Власне процес Формування кластерів пов’язаний з відшуканням усіх екстремумів функцій щільності за допомогою градієнтної процедури

$$x^{l} = x^{l-.1} + \eta^{l}; \quad \frac{\nabla f_{1}^{x}{}_{,x}{}^{l+J}}{\left\|\nabla f^{x}\left(x, x^{l-1}\right)\right\|} \tag{2.4}$$

x_0=x(k), l= 0,1,2…..; $\forall k$=1,2,…,N

Тобто кількість запусків алгоритму (4) визначається обсягом навчальної вибірки N. Зрозуміло, що при великих N процес кластеризації – пошуку локальних екстремумів може потребувати дуже багато часу. Тому запропоновані модифікації DENCLUE пов’язані із пришвидшенням процесу пошуку локальних екстремумів (2.3) шляхом модифікацій градієнтної процедури (2.4) [39].

У випадку коли спостережень x(k) у вибірці Xϵ($n_1 \times n_2$)- матрицями нескладно ввести у розгляд матричний варіант процедури (2.4):

$$x^l = x^{l-1} + \eta^l \Gamma^x(x, x^{l-1})(Tr\Gamma^x(x, x^{l-1})\Gamma^{xt}(x, x^{l-1}))^{-\frac{1}{2}} \quad (2.5)$$

де $\Gamma^x(x,x^{l-1})=\left\{\frac{\partial f^x(x,x^{l-1})}{\partial x_{i_1 j_2}}\right\} \in R^{n_1 \times n_2}$

Процес градієнтної оптимізації закінчується відшуканням m локальним екстремумів функції (2.3), при цьому чим менше значення ξ, тим більше кластерів може бути сформовано.

Пришвидшити процес відшукання локальних екстремумів можна, використовуючи замість градієнтного пошуку методи еволюційної оптимізації, серед яких в якості достатнього ефектного, чисельного простого і швидкого можна відзначити, так званий пошук на основі косяків риб [33], [37],[39], що повинен бути модифікований для вирішення задачі класифікації.

І хоча на сьогодні окрім FCM розроблено безліч методів та алгоритмів нечіткої кластеризації зі своїми перевагами та недоліками, все вони дозволяють знайти тільки локальний екстремум прийнятої цільової функції [38], що веде до того, що використання процедур оптимізації (нелінійного програмування) на основі похідних прийнятого критерію в загальному випадку не дозволяє отримати найкраще.

Подолати цю проблему можна, багаторазово вирішуючи завдання за різних початкових умовах та вибираючи найкращий варіант з множини отриманих. Зрозуміло, що подібний підхід суттєво збільшує час розв'язання задачі.

2.4 Модифікований метод оптимізації на основі косяків риб

При використані методів еволюційної оптимізації, що за суттю є методами оптимізації нульового порядку, тобто не використовують похідних, припускається, що при відшуканні екстремумів деякої функції

$f'(x)$ застосовується популяція агентів, кожен з яких діє або самостійно, , або взаємодіючи з іншими, при цьому рух кожного q-го агента (q=1,2,…Q) на l-й може бути записаний за допомогою співвідношення:

$$x_q^l = x_q^{l-1} + \eta_q^l Dir_q^l, \ (\text{q=1,2,…Q}) \qquad (2.6)$$

де $x_q^l = x_{q1}^l, x_{q2}^l, \dots, x_{qn}^l$, Dir_q^l- вектор що задає напрямок руху q-го агента на l-й ітерації пошуку.

У великій родині таких методів слід визначити метод на основі косяків риб, де кожен агент популяції імітує рух окремої риби, у косяці [39], [40]. Основною перевагою цього методу є достатня ефективність відшукання глобального екстремуму досить складних функцій, до яких можна віднести і функцію щільності розподілу даних в задачах кластеризації.

Автори методу вводять у розгляд ітерації , пов'язані із рухом косяка – агента. Чим ближче риба, тим ближче вона до екстремума – максимума. Вага кожної риби w_q налаштовується згідно з виразом:

$$w_q^l = w_q^{l-1} + \frac{f^x(x_q^l) - f^x(x_q^{l-1})}{max\{f^x(x_q^l) - f^x(x_q^{l-1})\}} \ \forall q = 1,2,\dots Q, \qquad (2.6)$$

При цьому:

$$0 < w_q^l < w_{max}^l, \quad w_l^0 = 0{,}5 w_{max}$$

Оператор планування описує як індивідуальний рух кожної риби, так і колективний рух косяка в цілому. Тут розглядається три типи руху:

- індивідуальний;
- інстинктивно-колективний;
- колективно рольовий.

Індивідуальний рух описується співвідношенням

$$x_{qi}^{l} = \begin{cases} x_{qi}^{l} + \eta_q^l Rand\{0,1\}, if\ f^x(x_q^l > f^x(x_q^{l-1}) \\ x_q^{l-1}\ else \end{cases}, \quad (2.7)$$

де, $Rand\{0,1\}$- рівномірно розподілене у інтервалі (0,1) випадкове число. Тут слід булоб відзначити, що (д) є суттю локальним випадковим порушником з поверненням, введеним Л.Растрігіним [41]. Фактично це процедура зондування функції $f'(x)$ в колі точки x_q^{l-1} при цьому крім (2.6) тут можна бути будь який інший алгоритм випадкового пошуку.

На базі зондування функції щільності за допомогою індивідуального руху (2.6) реалізується інстинктивно – колективний рух у напрямку зростання цієї функції:

$$x_q^l = x_q^{l-1} + \frac{(\sum_{p=1}^{Q}(x_p^l - x_q^{l-1}))(f^x(x_q^l) - f^x(x_q^{l-1}))}{\sum_{p=1}^{Q}(f^x(x_q^l) - f^x(x_q^{l-1})} \quad (2.8)$$

На цьому етапі відбувається зважене усереднення індивідуальних рухів з урахуванням «успішності» кожної з риб -агентів

І, нарешті, колективно – вольовий рух, коли всі риби косяка «згуртуються» до зваженого центру ваг, якщо косяк прямує до екстремуму, та «розбігаються», якщо популяція йде у невірному напрямку.

Вводячи у розгляд зважений центр ваги косяка риб:

$$\mathrm{Bar}^{\mathrm{l}} = \frac{\sum_{\mathrm{p=1}}^{\mathrm{Q}} \mathrm{x_q^l w_q^l}}{\sum_{\mathrm{p=1}}^{\mathrm{Q}} \mathrm{w_q^l}} \quad (2.9)$$

Можна записати цей рух у вигляді:

$$x_q^l = \begin{cases} x_q^l - \eta_q^l Rand\{0,1\} \frac{x_q^l - Bar^{l-1}}{\|x_q^l - Bar^{l-1}\|}, if \ \sum_{p=1}^{Q} w_p > \sum_{p=1}^{Q} w_p^{l-1} \\ x_q^l + \eta_q^l Rand\{0,1\} \frac{x_q^l - Bar^{l-1}}{\|x_q^l - Bar^{l-1}\|} \ if \ \sum_{p=1}^{Q} w_p < \sum_{p=1}^{Q} w_p^{l-1} \end{cases} \quad (2.10)$$

Для підвищення ефективності FSS у розгляд може бути введений додатковий оператор розведення, що дозволяє створювати нових риб – агентів, мають покращені характеристики у порівнянні з вже існуючими членами косяка. Для цього можна скористатися ідеями еволюційних операцій [41], серед яких з обчислювальної точки зору та ефективності – надійності відшукання екстемуму можна відзначити послідовний симплекс метод [42] та його модифікації[43], [44].

Сформуємо косяк, що містить Q=n+1 риб – агентів, при цьому ця кількість залишається незмінною у процесі пошуку, тобто популяція $x_{1,}^0 x_{2,}^0 \dots x_{Q,}^0$ генерується випадковим чином. В цій популяції знайдемо «найгарнішу» рибу x_{qworst}^0, що має найменшу вагу w_{qmin}^0 та «найкращу» рибу x_{qbest}^0з найбільшою вагою w_{qmax}^0. Основна операція руху симпплекса полягає у відображені x_{qworst}^0 через центр ваги n риб (без найгіршої), який може бути записаний вигляді:

$$\overline{x^0} = \frac{1}{n} \sum_{q=1}^{0} (x_q^0 - x_{qworst}^0) \quad (2.11)$$

В результаті цієї операції створюється нова риба:

$$x_q^1 = \overline{x^0} + a(\overline{x^0} - x_{qworst}^0) \quad (2.12)$$

Яка заміняє в косяку найгіршу особину x_{qworst}^0 . Таким чином формується нова популяція $x_{1,}^0 x_{2,}^0 \dots x_{Q,}^0$. Тут 0,5≤ a ≤2 – параметр, що

виражає форму косяка-симплекса у процесі оптимізації. При a=1 реалізується відображення симплекса через центр тяжіння $\overline{x^0}$, у випадку, якщо $(f^x(x_q^l) > f^x(x_{qbest}^0)$приймається a=2, тобто косяк «розтягується» у сприятливому напряму, якщо ж $(f^x(x_q^l) > f^x(x_{qworst}^0)$ обирається a=0,5, тобто симплекс стикається відносно невдалого напряму. Таким чином рух косяка-симплекса можже бути описаний за допомогою співвідношень:

$$\begin{cases} \bar{x}^{l-1} = \frac{1}{n}\sum_{E=1}^{Q}(x_q^l) - (x_{qworst}^0) \\ \bar{x}_q^l = x^{l-1} + a(x_q^l) - (x_{qworst}^0) \end{cases} \quad (2.13)$$

то у загальному випадку є за своєю суттю алгоритмом оптимізації Нелдера-Ліда [38]. Таким чином, з косяка у процесі пошуку екстремуму вилучаються найгірші риби з найнижчою вагою та створюються нові агенти з більшою вагою.

Таким чином, процес комбінованої оптимізації функції щільності (2.65)-(2.13) є за своєю суттю комбінацією FSS. випадкового пошуку та еволюційного планування на основі метода Неллера-Ліда.

Оскільки задача, що розглядається, є за своєю суттю проблемою багатомірної оптимізації, необхідно відшукати множину екстремумів, кожен з яких є номером деякого кластера. Тому задача оптимізації повинна вирішуватись багаторазово при різних значеннях ∂ та ξ. При знаходженні якогось з екстремумів з вихідної вибірки X виключаються спостереження, що розташовані безпосередньо в його в його околі (у [42] пропонується виключати з окола кожного знайденого центроїда 0,01N-0,02N спостережень). Після цього вилучення запропонована процедура комбінованої еволюційної оптимізації повторюється до відшукання всіх екстремумів – центроїдів.

Таким чином, у процесі пошуку крайнього, найгірша риба з найменшою вагою видаляється з нахилу і створюються нові агенти з

більшою вагою. Тому, процес комбінованої оптимізації щільності, функція (2.65)-(2.13) по суті є поєднання FSS, випадкового пошуку та еволюційного На основі методу Nelderead. Оскільки проблема, що розглядається, по суті є проблемою багатоекстремальної оптимізації, ценеобхідно знайти набір екстремумів, кожен з яких є центроїдом скупчення. Тому оптимізаційну задачу необхідно вирішувати багаторазово при різних значеннях о' і. При знаходженні будь-якого з екстремумів з вихідної вибірки X спостереження, розташовані безпосередньо в його околиці, виключаються (у роботі [37] пропонується виключити з околиці кожного спостережуваного центроїда 0,01N-0,02N спостереження). Після цього видалення запропонована процедура комбінованої еволюційної оптимізації повторюється до тих пір, поки не будуть знайдені всі екстремуми-центроїди.

3 ЕКСПЕРИМЕНТАЛЬНО -КОМП'ЮТЕРНА МОДЕЛЬ

У цьому розділі приведено результати експериментальної повірки еволюційного ройового алгоритму, щоб покращити якість вирішення задач оптимізації. Досліджено, як можна застосувати методи науки про дані для вирішення реальних проблем економічної політики.

3.1Аналіз набору даних

Працюючи з досить невеликим набором даних про 200 компаній, дослідження буде зосереджено на пошуку прихованих закономірностей, які можуть допомогти у встановленні відповідних ставок оподаткування. Швидкий погляд на інформаційну таблицю набору даних показує, що ми маємо як багату географічну інформацію, так і фінансову інформацію (рисунок 3.1).

	CompanyID	Kind	Years_Active	Growth_Potential_Index	Monthly_Revenue ($)	An_Revenue ($)	Tax_Burden	Tax_Income	Industry	Latitude	Longitude
2	132470126	Modern	6	39	375000	4500000	0,15	675000	Energy & Electricity	2,03333	45,349998
3	117696015	Traditional	7	81	375000	4500000	0,38	1710000	Agriculture & Livestock	3,1166699	43,650002
4	33917743	Modern	7	6	400000	4800000	0,13	624000	Energy & Electricity	2,03333	45,349998
5	117938475	Modern	8	77	400000	4800000	0,38	1824000	Agriculture & Livestock	8,4053602	48,484501
6	25975199	Modern	10	40	425000	5100000	0,14	714000	Energy & Electricity	11,27554	49,187901
7	10528239	Modern	7	76	425000	5100000	0,38	1938000	Agriculture & Livestock	3,7965	42,544201
8	150419880	Modern	12	6	450000	5400000	0,16	864000	Energy & Electricity	4,73597	45,203999
9	43788233	Modern	8	94	450000	5400000	0,38	2052000	Agriculture & Livestock	6,7684002	47,4296
10	114831883	Modern	21	3	475000	5700000	0,17	969000	Energy & Electricity	3,1166699	43,650002
11	97908458	Modern	10	72	475000	5700000	0,38	2166000	Agriculture & Livestock	2,3333299	42,283298
12	121417141	Modern	22	14	475000	5700000	0,15	855000	Energy & Electricity	6,7684002	47,4296
13	61892345	Modern	12	99	475000	5700000	0,38	2166000	Agriculture & Livestock	8,4053602	48,484501
14	92419477	Modern	19	15	500000	6000000	0,14	840000	Energy & Electricity	6,137	46,6259
15	135744897	Modern	8	77	500000	6000000	0,38	2280000	Agriculture & Livestock	6,137	46,6259
16	152838548	Modern	12	13	500000	6000000	0,17	1020000	Energy & Electricity	11,27554	49,187901
17	107278392	Traditional	7	79	500000	6000000	0,38	2280000	Agriculture & Livestock	1,869318	44,960175
182	56507166	Modern	12	32	2425000	29100000	0,17	4947000	Tecommunication	2,03333	45,349998
183	37872677	Modern	11	86	2425000	29100000	0,35	10185000	Manufacturing &	2,03333	45,349998
184	113100748	Traditional	15	15	2450000	29400000	0,14	4116000	Tecommunication	2,03333	45,349998
185	142548527	Modern	10	88	2450000	29400000	0,35	10290000	Manufacturing &	2,03333	45,349998
186	157023618	Modern	14	39	2475000	29700000	0,14	4158000	Tecommunication	2,03333	45,349998
187	21080245	Traditional	10	97	2475000	29700000	0,35	10395000	Manufacturing &	2,03333	45,349998
188	120441726	Modern	18	24	2525000	30300000	0,15	4545000	Tecommunication	2,03333	45,349998
189	27807852	Traditional	9	68	2525000	30300000	0,35	10605000	Manufacturing &	3,1166699	43,650002
190	86625276	Modern	14	17	2575000	30900000	0,14	4326000	Tecommunication	11,27554	49,187901
191	79514507	Modern	12	85	2575000	30900000	0,35	10815000	Manufacturing &	11,27554	49,187901
192	31610829	Modern	11	23	2575000	30900000	0,17	5253000	Tecommunication	2,03333	45,349998
193	33903756	Modern	11	69	2575000	30900000	0,35	10815000	Manufacturing &	2,03333	45,349998
194	13237210	Traditional	11	8	2825000	33900000	0,17	5763000	Tecommunication	2,03333	45,349998
195	148634837	Modern	13	91	2825000	33900000	0,35	11865000	Manufacturing &	2,03333	45,349998
196	13689934	Modern	16	16	3000000	36000000	0,14	5040000	Tecommunication	2,03333	45,349998
197	93648417	Modern	12	79	3000000	36000000	0,35	12600000	Manufacturing &	2,03333	45,349998
198	48189850	Modern	15	28	3150000	37800000	0,14	5292000	Tecommunication	2,03333	45,349998
199	157978408	Traditional	11	74	3150000	37800000	0,35	13230000	Manufacturing &	2,03333	45,349998
200	105185518	Traditional	11	18	3425000	41100000	0,15	6165000	Tecommunication	2,03333	45,349998

Рисунок 3.1 – Фрагмент інформаційної таблиці набору даних

Найважливішими стовпцями є дохід у доларах США, роки послідовної діяльності, потенціал зростання, галузь, в якій працює компанія, та її податкове навантаження. Географічне розташування, також важливе. Усі параметри які було застосовано у даних наведено нижче:

– Growth_Potential_Index (0–100 вищих значень, які вказують на те, що компанія може швидко розвиватися в коротко- та середньостроковій перспективі);

– Monthly_Revenue ($) (місячний валовий дохід компанії);

– An_Revenue ($) (місячний дохід, екстраполюваний протягом року);

– Monthly_Revenue ($) (офіційна ставка податку, яка вираховується з валового доходу);

– Tax_Income (дохід, який місцеві адміністрації отримують від компанії щорічно);

– Industry (досить очевидно);

– Geo-information(у нас є довгота і широта).

Цей набір даних можна використовувати для різних цілей, але метою буде застосування еволюційного ройового алгоритму для групування компаній і визначення того, чи оподатковуються вони надмірно чи занижена.

Але перш ніж ми розглянемо тему податкового впливу, спочатку проведено дослідницький аналіз набору даних.

Рисунок 3.2 показує, що переважна більшість компаній у нашому наборі даних працюють у секторі фінансових послуг, що відображає бум мобільних грошей та грошових переказів. Другий підграфик показує, що існують кластерні моделі, за якими фірми чітко розрізняються тим, що одні генерують високі податкові надходження, а інші – досить нижчі суми податків. Фірми також відрізняються за потенціалом зростання.

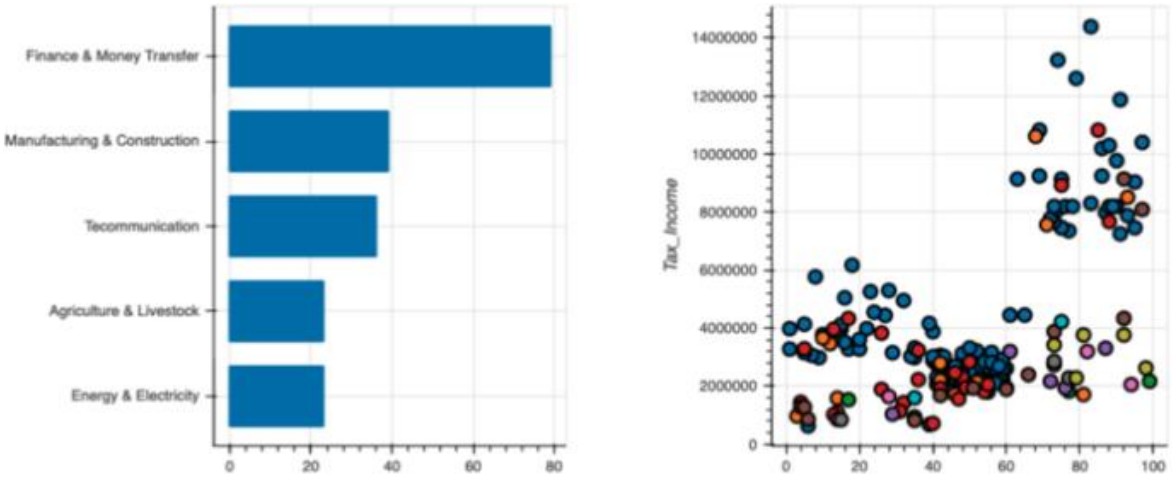


Рисунок 3.2– Популяризація галузі (ліворуч)
та потенціал зростання (праворуч)

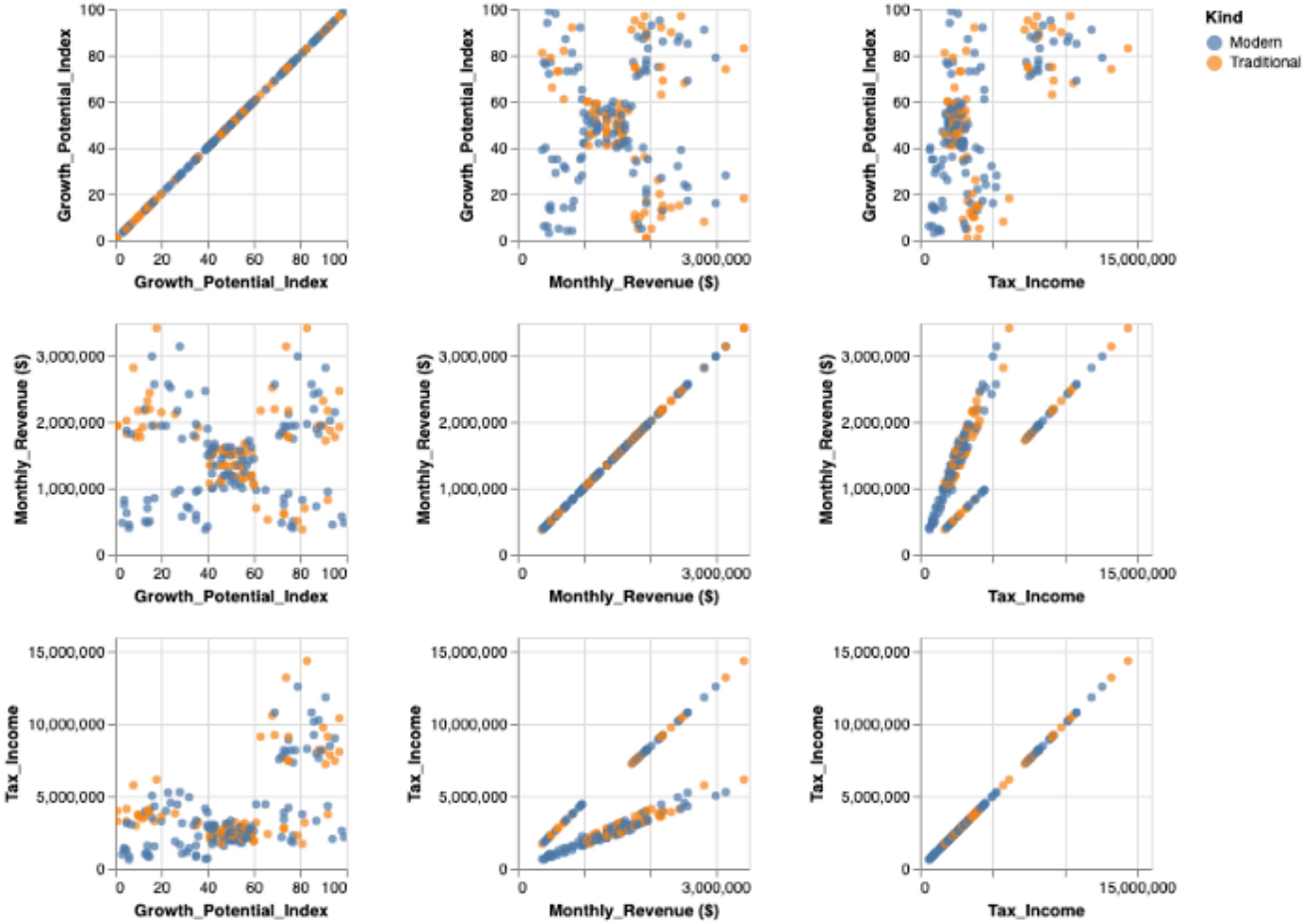


Рисунок 3.3 – Графік парної сітки між потенціалом зростання,
місячним доходом і податковим доходом

На рисунку 3.3, вказує на те, що сучасні компанії потребують підтримки, можливо, зниження податків, щоб стимулювати зростання та розблокувати економію від масштабу. На одному з діаграм розкиду видно, що існує кластер компаній з низьким потенціалом зростання і відносно мізерним місячним доходом. Можливо, це випадок ухилення від сплати податків, що зводить нанівець мою пропозицію щодо податкових стимулів для стимулювання зростання.

На рисунку 3.4, зображено об’єм податкових надходжень. Ми чітко бачимо– що сектор «Виробництво та будівництво» генерує найвищу частку податкового доходу по відношенню до річного доходу. Сільське господарство та тваринництво також демонструють непропорційну ставку податку в порівнянні з такими, як енергетика та електроенергія та фінанси та грошові перекази. Наведений нижче сюжет надає ще більшої достовірності цьому спостереженню.

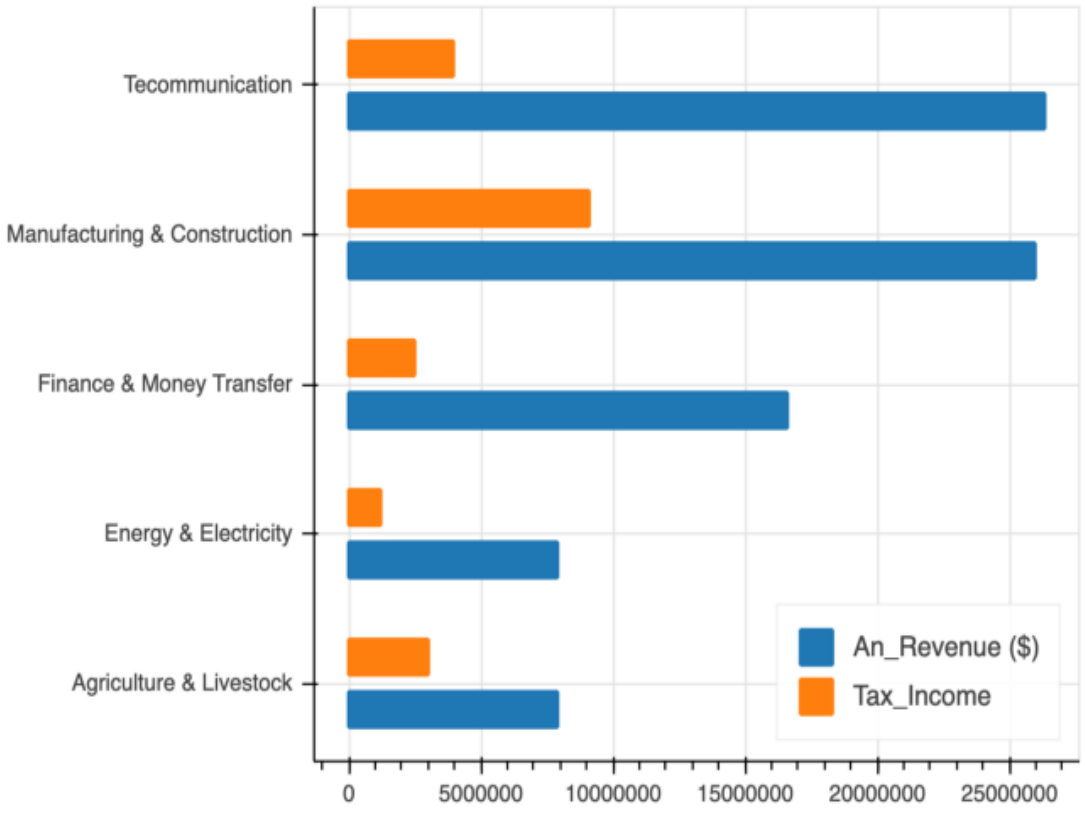


Рисунок 3.4 – Порівняння надходження податку за секторами

На рисунку 3.5 порівняно потенціал зростання за секторами з податковим тягарем за секторами в порядку спадання. Видно, що виробництво та будівництво, сільське господарство та тваринництво демонструють найвищу частку податкового тягаря серед усіх галузей, незважаючи на те, що вони мають найбільший потенціал зростання. Можливим поясненням цього може бути той факт, що обидва сектора сплачують податки на імпорт і експорт у. Тому для зняття тягаря податків та транспортних витрат необхідне втручання держави, які, безсумнівно, гальмують зростання. Крім того, алгоритми кластеризації можуть допомогти виявити приховані закономірності, які необхідно усунути.

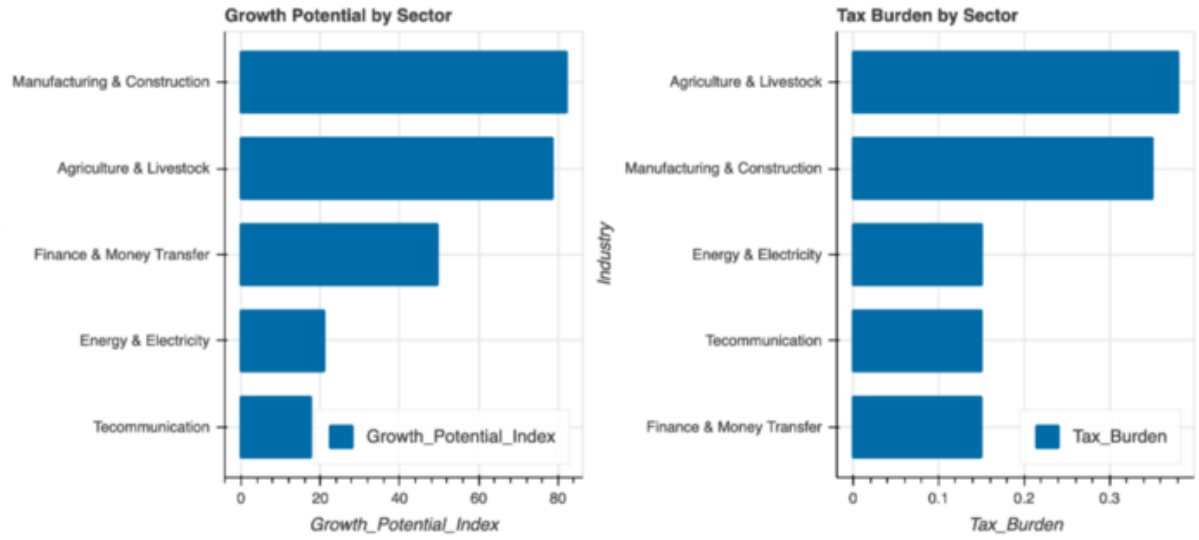


Рисунок 3.5 – Потенціалу зростання (ліворуч)
та податковий тягар (праворуч)

3.2 Аналіз результатів експериментів

Експериментальне дослідження проводилося на даних представлених на рисунку 3.1 та двох тестових багатоекстремальних функцій., а тестові мультиекстремальні функції в таблиці 3.1.

У зв’язку з тим, що функції Растрігіна (рисунок 3.6 і 3.7) та функцію Гріванґка(рисунок 3.8 і 3.9) мають багато локальних екстремальних точок у своїй області пошуку. Функція Растригіна є неопуклою функцією і часто використовується як задача перевірки продуктивності для алгоритмів оптимізації. Для алгоритму оптимізації вона є дуже складною. Його складна поведінка призводить до того, що алгоритми оптимізації часто зависають на локальних мінімумах. Наявність великої кількості косинусних коливань на площині вводить в цю функцію складну поведінку.

Таблиця 3.1 – Перевірка мультиекстремальних функцій

Назва функції	Формула	Атрибути	Крок
Растригіна	$f(x) = 20 + x^2 + y^2 - 10\cos(2\pi x) + cos(2\pi y)$	[-5,12; 5,12]	0,01
Гріванґка	$f(x) = \frac{1}{4000}x + \frac{1}{4000}y - \cos\left(\frac{x}{\sqrt{1}}\right)\cos\left(\frac{x}{\sqrt{2}}\right) + 1$	[-10; 10]	0,1

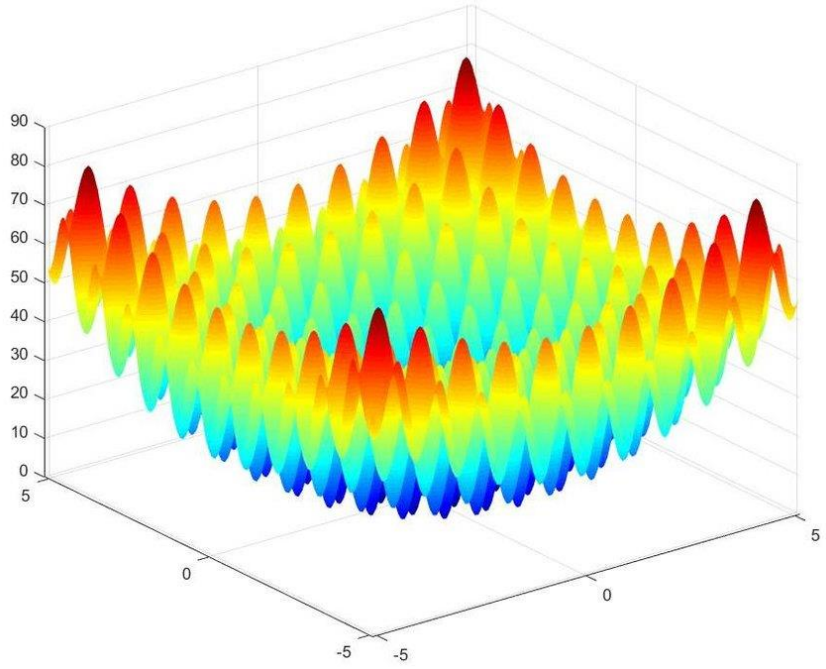


Рисунок 3.6 – Функція Растрігіна

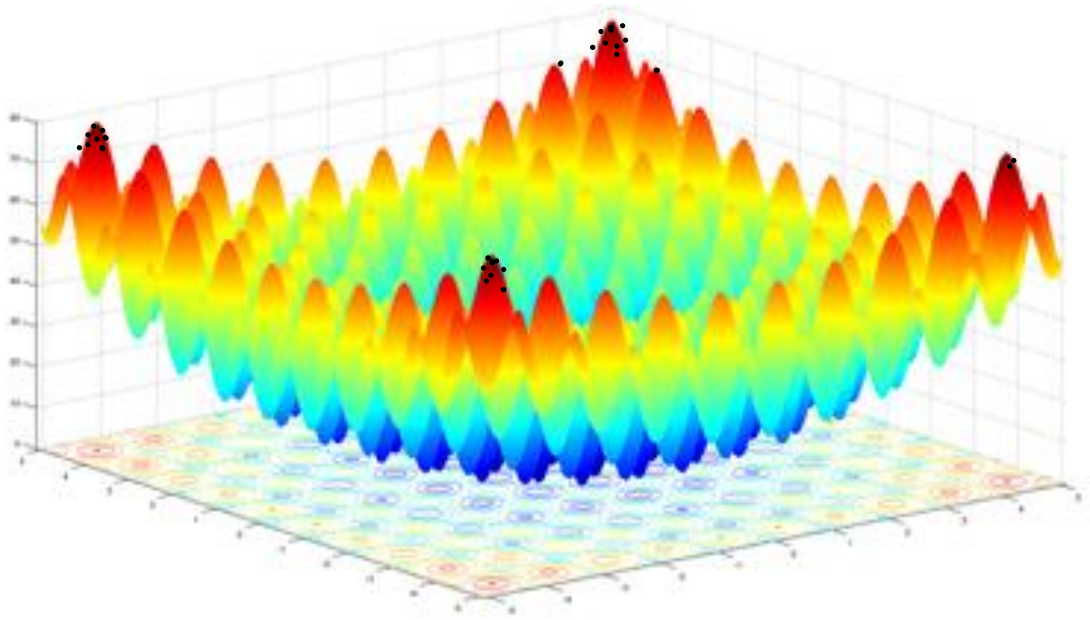
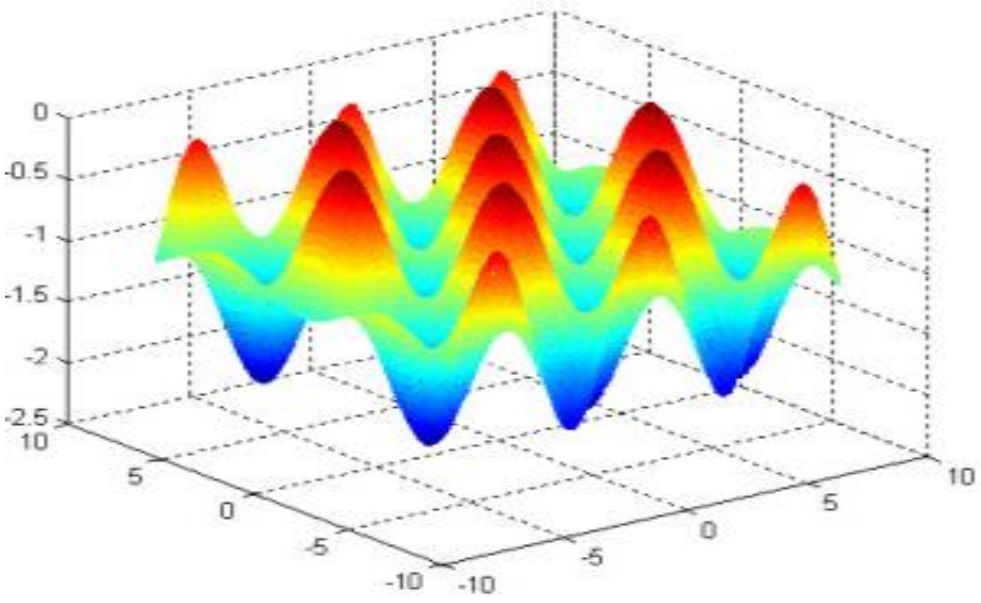


Рисунок 3.7 – Модифікований метод оптимізації на основі косяків риб (fish School) на функції Растрігіна



.

Рисунок 3.8 – функція Гріванґка

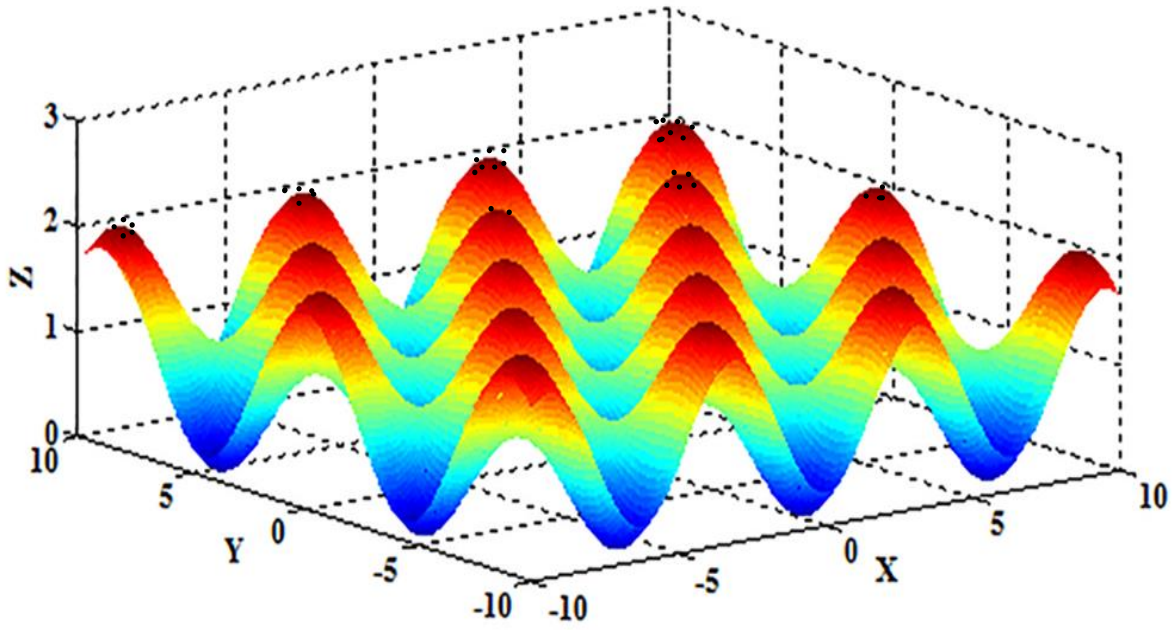


Рисунок 3.9 – Модифікований метод оптимізації функції Гріванґка, заснований на косяках риб

Функція Растрігіна грунтується на функції De Jong, або як її ще називають – «сферична модель». Вона безперервна і випукла. Визначення функції De Jong:з додаванням косинусної модуляції для створення багатьох локальних екстремумів. Таким чином, тестова функція є мультимодальною, а положення екстремумів розподілені регулярно.

Порівняння точності відомих алгоритмів оптимізації, таких як Fish School (FSS) і Cat Swarm (CSO), і запропонований метод модифікації оптимізації на основі Fish School (OMFS).

З отриманого результату, наведеного в таблиці 3.2, ми бачимо, що модифікований метод оптимізації на основі Fish School загалом працює краще, ніж оригінальний алгоритм.

Для оцінки ефективності методу кластеризації використано кілька метрик валідності: Індекс Данна DD-високе значення вказує на кращу кластеризацію; Індекс Девіса-Булдіна (DBI) - найменше значення вказує на

кращу кластеризацію; Cluster Accuracy (CA) – високе значення вказує на найкращу якість кластеризації.

Таблиця 3.2 – Порівняння точності результатів

Дані	Точність	OMFS	FSS	CSO
Растригіна	Середній	190,46	189,65	190,46
	Найкращий	195,83	195,59	195,46
Гріванґка	Середній	2,65	2,41	2,65
	Найкращий	3,82	3,12	3,81
Показник потенціалу	Середній	951,47	951,01	951,15
	Найкращий	959,64	959,43	959,55
Податковий тиск	Середній	291,77	291,17	291,77
	Найкращий	299,84	299,48	299,64

Для порівняння запропоновано метод класифікації векторних і матричних наборів даних на основі комбінованої оптимізації функцій розподілу (CODF) за класичним алгоритмом DENCLUE та DENCLUE-IM для кластеризації великих даних. Результати наведено у таблиці 3.3

Таблиця 3.3 – Алгоритми порівняння за метрикою їх валідності

Дані	Заходи	Показник потенціалу	Податковий тиск
CODF	DL	0,835	0,721
DENCLUE		0,789	0,721
DENCLUE-IM		0,831	0,693
CODF	DBI	0,768	0,764
DENCLUE		0,867	0,864
DENCLUE-IM		1,041	1,041
CODF	CA	0,718	0,920
DENCLUE		0,805	0,920
DENCLUE-IM		0,701	0,911

Усі ці результати роблять висновок, що запропонований метод класифікації векторних і матричних наборів даних на основі комбінованої оптимізації функцій розподілу має прийнятну кластеризацію.

При застосуванні кластеризації за даним ройовим алгоритмом наявні дані можна розділити на 5 кластерів та надати поради щодо економічних процесів, а саме реальних проблем економічної політики.

Сектор енергетики та електроенергетики підпадає під кластер низький дохід і низький потенціал зростання. Фірми в цьому секторі підлягають низьким податковим тягарем, і це має залишатися як є. Імовірність подальшого розвитку цього сектора висока в соціально-економічному середовищі сучасного реалій.

Сектор телекомунікацій підпадає під категорію кластеру високий дохід, але низький потенціал зростання. Фірми в цьому секторі підлягають низьким податковим тягарем, хоча вони є одними з найбільших прибутків. Схоже, що цей сектор недостатньо регулюється. Ці компанії повинні сплачувати свою справедливу частку оподаткування, що дасть більше можливостей для політики для допомоги іншим, більш хворим секторам.

Кластер високий дохід і високий потенціал зростання, безумовно, є сектором виробництва та будівництва, який наразі підлягає одному з найвищих податкових тягарів (близько 38%). Раніше ми визначали, як цей сектор страждає від ввізних мит на сировину у великих портах. Збільшення надходжень від сектору телекомунікацій може дозволити владі зробити це.

Низький дохід, але високий потенціал зростання можна визначити як сектор сільського господарства та тваринництва, який є основою економіки Сомалі. Значна частина робочої сили в країнах зайнята в цьому секторі. Крім того, він відіграє вирішальну роль для кращої продовольчої безпеки, тому його необхідно підтримувати та захищати від необґрунтованих мит і податків у портах та аеропортах. Високий податковий тягар, який зараз становить близько 40%, має знизитися для реалізації цих стратегічних цілей.

Останній сектор, – це кластер середній дохід і середній потенціал зростання. Це остання галузь, що залишилася від формулювання наших рекомендацій щодо політики. Її можна ідентифікувати як галузь фінансів та грошових переказів, яка, як ми бачили, має дуже низький податковий тягар близько 15%. Задля більш стратегічно важливих галузей, політикам рекомендується підвищити цей показник.

ВИСНОВОК

В магістерській роботі представлені результати вирішення задачі модифікації ройових алгоритмів оптимізації, що мають покращені властивості.

У випадках застосування методів нечіткої кластеризації приймають до уваги належності. Тому в цих випадках цільова функція багатоекстримальна і методи знаходять лише локальний екстремум. При використані еволюційного ройового алгоритму знаходять глобальний екстремум.

В процесі виконання атестаційної роботи отримано такі результати.

1. Проаналізовано сучасний стан теорії обробки даних. Проведений аналіз показав, що розв'язання даної задачі потребує інтеграції можливостей швидкодіючих ймовірнісних нейронних мереж для задоволення обмежень у часі при послідовній обробці великих об’ємів даних, а також переваг нечіткої логіки для виділення класів, що перетинаються в просторі ознак.

2. Розглянуто задачу кластеризації масивів даних, що описано у векторній та матричній формах на основі оптимізації функцій щільності розподілу даних у цих масивах

3. Для оптимізації цих функцій – пошуку локальних екстремумів запропонований алгоритм, що є гібридом Fish School Search, випадкового пошуку та еволюційної оптимізації. Цей алгоритм не потребує обчислення похідних функцій, що оптимізується і в загальному випадку призначений для відшукання максимумів багатоекстремальних функцій матричного аргументу (зображень).

4. Результати досліджень та публіковалися у низці публікацій [36], [39], [40], [41].

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жалдак М.І. Деякі методичні аспекти навчання інформатики в школі і педагогічному університеті. "Комп'ютерно-орієнтовані системи навчання". Випуск 9. Науковий часопис. Київ.: НПУ ім. М.П. Драгоманова. 2005. С. 3-14.

2. Бахрушин В.Є. Аналіз даних :навчальний посібник /В.Є.Бахрушин. Запоріжжя :ГУ “ЗІДМУ”, 2006. 128с

3. Буйницька О.П. Інформаційні технології та технічні засоби навчання. Навч. посіб. К.: Центр учбової літератури, 2012. 240 с.

4. Верес О. М. Класифікація методів аналізу великих даних / О. М. Верес, Р. М. Оливко // Вісник Національного університету “Львівська політехніка” 2017. Випуск 872. С.84-92.

5. Бонч-Брує́вич Г.Ф., Носенко Т.І. Інтерактивний комплекс SMART Board у навчальному процесі: Навч. посіб. К.: Київ. ун-т ім. Б. Грінченка, 2010. 108 с.

6. Gan G., Ma Ch., Wu J. Data Clustering: Theory, Algorithms and Applications. Philadelphia, Pennsilvania: SIAM, 2007. – 455 p.

7. Xu R., Wunsch D.C. II. Clustering– Hoboken, N.J.: John Wiley & Sons, Inc., 2009. – 341 p.

8. Ferenci, P. (2017). Hepatic encephalopathy. Gastroenterology Report, 5(2), 138–147. doi: http://doi.org/10.1093/gastro/gox013

9. Butterworth, R. (2016). Neurosteroids in hepatic encephalopathy: Novel insights and new therapeutic opportunities. The Journal of Steroid Biochemistry and Molecular Biology, 160, 94–97. doi: 10.1016/j.jsbmb.2015.11.006.

10. Volk, M., Tocco, R., Bazick, J., Rakoski, M., & Lok, A. (2012). Hospital Readmissions Among Patients With Decompensated Cirrhosis. The American Journal of Gastroenterology, 107(2), 247–252. doi: 10.1038/ajg.2011.314.

11. Bezdek J.C., Keller J., Krishnapuram R., Pal N.R. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. – N.Y.: Springer Science + Business Media, Inc., 2015. – 776 p.

12. Романова Ю. Д. Інформаційні технології в менеджменті (управлінні): підручник і практикум для академічного бакалаврату / під заг. ред. Д. Ю. Романової. – М: Видавництво Юрайт, 2015. 478 с. [Електронний ресурс]. Режим доступу: https://stud.com.ua/62442/menedzhment/intelektualniy_analiz_danih.

13. Грабовецький, Л. Дослідження та використання методів інтелектуального аналізу даних (ІАД) для підвищення ефективності роботи інтернет-магазину меблів / Леонтій Грабовецький // Наукові здобутки молоді – вирішенню проблем харчування людства у XXI XXI столітті: програма і матеріали 80 міжнародної наукової конференції молодих учених, аспірантів і студентів, 10–11 квітня 2014 р. – К.: НУХТ, 2014. – Ч. 2. – С. 479-480.

14. Чорноус Г. Оптимізація ціноутворення на основі моделей інтелектуального аналізу даних / Г. Чорноус, С. Рибальченко // Вісник Київського національного університету імені Тараса Шевченка. – 2015. – №7 (172). С. 52-58.

15. Adamo J.-M. Data mining for association rules and sequential patterns: sequential and parallel algorithms / J.-M. Adamo. New York: Springer-Verlag. 2001. 259 p.Grosan C., Abraham A., Chis M. Swarm intelligence in Data Mining – Studies in Computational Intelligence. 2006. 34. P. 1-20.

16. L.Rutkowski. Computational Intelli-gence. Methods and Techniques. Berlin-Heidelberg: Springer-Verlag, 2008.-514p.

17. Mumford C. L., Jain L.C. Computation-al Intelligence. Berlin: Springer-Verlag, 2009. - 729p.

18. Kroll A. Computational Intelligence. Eine Einführung in Probleme, Methoden und technishe Anwendungen – München: Olden-bourg Verlag, 2013. – 428 S.

19. Kruse R., Borgelt C., Klawonn F., Moawes C., Steinbrecher M., Held P. Computa-tional Intelligence. A Methodological Introduc-tion. - Berlin: Springer-Verlag, 2013. 488 p.

20. Kacprzyk J., Pedrycz W. Springer Handbook of Computational Intelligence. – Berlin Heidelberg: Springer-Verlag, 2015. – 1634p.

21. Abonyi J., Feil B. Cluster Analysis for Da-ta Mining and System Identification. - Basel: Birkhauser, 2007. - 303p.

22. Grosan C., Abraham A., Chis M. Swarm intelligence in Data Mining - Studies in Com-putational Intelligence. - 2006. - 34. - P. 1-20.

23. Chu S.-C., Tsai P.-W., Pan J.S. Cat swarm optimization // Lecture Notes in Artifi-cial Intelligence. 4099. - Berlin Heidelberg: Springer-Verlag, 2006. – P. 854-858.

24. Chu S.-C., Tsai P.-W. Computational In-telligence based on the behavior of cats // Int. J. of Innovative Computing, Information, and Control. 2007. 3. - №1. P.163 -173.

25. Liu Y., Wu, Shen Y. Cat swarm optimi-zation clustering (KSACSOC): A cat swarm optimization clustering algorithm. Sci. Re-seach and Essays 2012 – 7. №49. – P. 4176 – 4185.

26. Бодянский Е.В, Шафроненко А.Ю. Рандомизированная модификация метода оптимизации на основе кошачьих стай. - Системи обробки інформації. 2018. № 1(152). С. 142-147.

27. Shafronenko, A., Bodyanskiy, Ye. Pliss, I., Patlan, K.: Fuzzy Clusterization of Distorted by Missing Observations Data Sets Using Evolutionary Optimization. In: Proceedings “Advanced Computer Information Technologies (ACIT’2019)”, Česke Budejovice, Czech Republic, June 5-7, 2019, pp. 217-220 (2019). doi: 10.1109/ACITT.2019.8779888

28. Xu R., Wunsch D.C. II. Clustering Hoboken, N.J.: John Wiley & Sons, Inc., 2009. – 341p.

29. Bezdek J.C. Pattern Recognition with Fuzzy Objective Function Algorithms.-N.Y.:Plenum Press, 1981.-272p.

30. Bezdek J.C., Keller J., Krishnapuram R., Pal N.R. Fuzzy Models and Algorithms for Pat-tern Recognition and Image Processing. - N.Y.: Springer Science + Business Media, Inc., 2015. - 776p.

31. Zhukov I.A., Kravets I.M. “Organization of distribution load database in the analysis and information system”, International scientific technical conference “DESSERT-2009”, Radioelectronic and computer system. Kharkiv, KhAI, 2009. Vol. 5(39). P. 25–30.

32. Zhukov I.A., Kravets I.M. “An algorithm of fragmentation optimization in distributed database”, Advanced computer systems and networks design and application: proceedings of the 4-th International conference ACSN-2009. – Lviv, 2009. – P. 72–75.

33. Rosenbrock, H.H. “An Automatic Method for Finding the Greatest or Least Value of a Function.”, Computer J. 3, 1960. – P. 175–184.

34. Darrell Whitley. “A genetic algorithm tutorial” // Statistics and Computing. – Springer Netherlands. Vol. 4(2) / June, 1994. – PP. 65-85.

35. Sharapov R., Lapshin A. “Convergence of genetic algorithms” // Pattern Recognition and Image Analysis. MAIK Nauka/Interperiodica distributed exclusively by Springer Science+Business Media LLC. Vol. 16(3) / July, 2006. PP. 392-397Wantson G.S. Smooth regression analysis. Sankhya: The Indian Journal of Statistics. 1964 Ser. A-26-№4. - P. 359-372.

36. Бодянський Є.В., Патлань К.В «Послідовна он-лайн модернізація методу кластеризації J- середніх».- Інформаційні технології: наука, техніка, технологія, освіта, здоров’я: тези доповідей XXVII міжнародної науково-практичної конференції MicroCAD-2019, 15-17 травня 2019 р.: у 4 ч. Ч. III. / за ред. проф. Сокола Є.І. Харків: НТУ «ХПІ». 425 с.

37. Kennedy J. Eberhart R. Particle swarm optimization. Proc. IEEE Int. Conf. on Neural Networks. - Perth, Australia, 1995. - P. 1942-1948.

38. Eiben A., Smith J. Introduction to Evolutionary Computing. - Heidelberg, Springer.

39. Бодянский Е.В., Шафроненко А.Ю., Патлань Е.В.: НЕЧЕТКАЯ КЛАСТЕРИЗАЦИЯ МАССИВОВ ДАННЫХ НА ОСНОВЕ ЭВОЛЮЦИОННОГО МЕТОДА ОПТИМИЗАЦИИ КОШАЧЬИХ СТАЙ- 2018-№2(91)-С.3-8 (Web of Science)

40. Бодянський Є., Шафроненко А.,Плісс І., Патлань К., НЕЧІТКА КЛАСТЕРИЗАЦІЯ МАСИВІВ ДАНИХ ЗА ДОПОМОГОЮ ЕВОЛЮЦІЙНИХ РОЙОВИХ АЛГОРИТМІВ – Матеріали V Міжнародна науково-практичної конференції "Обчислювальний інтелект» (результати, ппроблеми, перспективи) – Ужгород, 2019 – 74-75

41. Shafronenko, A., Bodyanskiy, Ye. Pliss, I., Patlan, K.: Fuzzy Clusterization of Distorted by Missing Observations Data Sets Using Evolutionary Optimization. In: Proceedings “Advanced Computer Information Technologies (ACIT’2019)”, Česke Budejovice, Czech Republic, June 5-7, 2019, pp. 217-220 (2019). doi: 10.1109/ACITT.2019.8779888 (Scopus)

42. Karpenko A. P. Population algorithms for global continious optimization. Review of new and little - known algorithms - Приложение к журналу" Информационные технологии" N07/2012.-32p.

43. Bastos - Felino C.J.A., Lima Neto C.J.A., Lins A.J.C.C., Nascimento A.I.S., Lima M.P. Fish School Search. - Nature. - Insperiod Algorithms for Optimization. - Berlin Hedelberg: Springer Verlag, 2009. - SCI 193. - P. 261-277.

44. Cavalcanti Jr. G.M., Bastos - Felino C.J.A., Lima Neto F.B., Castro R.M.C.S. A hybrid algorithm based on fish school search and particle swarm optimization for dynamic problems.

45. Proc. Int. Conf. in Swarm Intelligence (ICSI). - 2011. - V. 2. P.543-552. (21) Janecek A., Tan Y. Feeding the fish-weight update strategies for the fish school seach algorithm. - Berlin Heidelberg: Springer - Verlag. - Lecture Nodes in Computer Scince. - 2011. - V.6729.- Part 11. - P. 553-562.

46. Box Y.E.P. Evolutionary operation: A method for increasity industrial productivity. -

47. Rodrigues F., Laio A. Clustering by fast search and find of density peacks. Science. - 2014,-34.- P.1492-1496.

48. Hinneburg, A. and H. Gabriel. «DENCLUE 2.0: Швидка кластеризація на основі оцінки щільності ядра». IDA (2007).