МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ САДІВНИЦТВА

Факультет економіки і підприємництва

Кафедра інформаційних технологій

Допущено до захисту
"_____" _____ 2024 р.
Зав. кафедри, к.т.н., доцент
_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Інформаційна технологія для ідентифікації об’єктів в системах прийняття рішень

Виконав – студент 2 курсу, групи 21м-кн
освітньої програми "Комп’ютерні науки"
спеціальності 122 Комп’ютерні науки
Сторож Віталій Васильович

Науковий керівник – д.т.н., професор
Кучерук Володимир Юрійович

Умань – 2024

Реферат
на кваліфікаційну роботу «Інформаційна технологія для ідентифікації об’єктів в системах прийняття рішень»
студента факультету економіки та підприємництва
Уманського національного університету садівництва
Сторожа Віталія Васильовича

Структура та обсяг роботи - пояснювальна записка до кваліфікаційної роботи магістра складається з 3 розділів, містить 69 сторінок, 15 рисунків, 1 додаток, 61 літературних джерел.

У кваліфікаційній роботі магістра розроблено вдосконалені методи кластерного аналізу та класифікації даних для ідентифікації об’єктів, що сприяють покращенню процесу прийняття рішень у комп’ютерно-інтегрованих системах управління.

Проведено аналіз сучасних методів машинного навчання (ML), які використовуються для ідентифікації об’єктів, і визначено їх роль у підвищенні ефективності управлінських рішень. Обґрунтовано використання кластеризації та класифікації для задач ідентифікації.

Запропоновано метод кластеризації, заснований на фізичному принципі поверхневого натягу, та модифікований класифікатор SVM, побудований на його основі.

Реалізовано програмний додаток ClusterBoundsSearcher_v0.0.2, який включає модуль для кластерного аналізу з визначенням граничних точок кластерів, модифікований класифікатор SVM і засіб зменшення розмірності простору ознак для візуалізації результатів. Машинний експеримент підтвердив ефективність розроблених методів і програмного забезпечення.

Запропоновані рішення покращили швидкість і якість прийняття рішень у системах управління.

Анотація

В кваліфікаційній роботі магістра розроблено методи кластерного аналізу і класифікації даних, що використовуються під час ідентифікації об’єктів для підтримки прийняття рішень в комп’ютерно-інтегрованих системах управління. Запропоновані метод кластеризації даних, який використовує фізичний принцип поверхневого натягу, і метод машинного навчання класифікатора типу SVM, які дозволили підвищити ефективність ідентифікації об’єктів і станів систем. Надано математичне обґрунтування методу кластерного аналізу на основі методу градієнтного спуску. Розроблено алгоритми і програми ідентифікації даних запропонованими методами. Проведений машинний експеримент підтвердив адекватність розроблених методів і програмного забезпечення.

Ключові слова: системи управління, методи ідентифікації, кластерний аналіз, класифікація даних, прийняття рішень.

Annotation

In the master's qualification work, methods of cluster analysis and data classification used during object identification to support decision-making in computer-integrated control systems were developed. A data clustering method using the physical principle of surface tension and a machine learning method of the SVM type classifier were proposed, which allowed to increase the efficiency of object and system state identification. A mathematical justification of the cluster analysis method based on the gradient descent method was provided. Algorithms and programs for data identification using the proposed methods were developed. A machine experiment was conducted to confirm the adequacy of the developed methods and software.

Keywords: control systems, identification methods, cluster analysis, data classification, decision-making.

ЗМІСТ

Перелік умовних позначень, скорочень і термінів 5
ВСТУП 6
РОЗДІЛ 1. АНАЛІЗ СТАНУ ПРОБЛЕМИ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ В СИСТЕМАХ АВТОМАТИЗАЦІЇ 9
1.1. Сучасний стан проблеми ідентифікації об'єктів 9
1.2. Аналіз методів класифікації образів 20
1.3. Огляд методів кластерного аналізу 28
РОЗДІЛ 2. МАТЕМАТИЧНЕ ОБГРУНТУВАННЯ ОБРАНИХ МЕТОДІВ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ 32
2.1. Математичне обґрунтування методу класифікації 32
2.2. Розробка методу кластеризації даних 39
2.3. Розробка алгоритму кластеризації даних 43
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ 46
3.1. Обґрунтування вибору програмної платформи 46
3.2. Розробка структури програмного забезпечення 48
3.3. Інструкція користувачу програми 51
ВИСНОВКИ 57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 58
ДОДАТКИ ..... **Помилка! Закладку не визначено.**
Додаток А. Лістинги програм ..... **Помилка! Закладку не визначено.**

Перелік умовних позначень, скорочень і термінів

АСУП	Автоматизована система управління підприємством
ML	Machine Learning
SVM	Support vector machine
DBSCAN	Density-based spatial clustering of applications with noise

ВСТУП

Сучасний розвиток науки і техніки поставив на порядок денний необхідність автоматизації процесів обробки інформації для ідентифікації об’єктів на технологічних дільницях та під час оперативного прийняття рішень на адміністративному рівні сучасних комп’ютерно-інтегрованих систем автоматизації управління виробництвом і бізнесом. В більшості робіт з даної проблематики задача ідентифікації розв’язувалась з використанням лінійної апроксимації і критерія мінімуму квадратичного відхилення. Але практика застосування такої методики до об’єктів різної природи показала, що не завжди такий вибір рівняння апроксимації і критерія якості призводить до задовільних результатів. На сьогодні розв’язання цієї задачі пов’язано з впровадженням інтелектуальних методів в комп’ютерні системи управління, до яких відносяться, зокрема, методи машинного навчання (Machine Learning ML). Машинне навчання - це підрозділ науки штучного інтелекту, присвячений розробці методів і алгоритмів опрацювання інформації для прийняття рішень на адміністративному і технологічному рівнях АСУП, які покращують якість управління на основі сумісного аналізу навчальних даних і отриманих результатів від виконання прийнятих рішень на управління. Перевагою застосування машинного навчання в комп’ютерно-інтегрованих системах управління є звільнення від рутинної нецікавої роботи осіб, що приймають рішення, звільняючи їм простір для творчості і розвитку.

Методи ML дозволяють розробляти ефективні системи прийняття рішень на адміністративному рівні підприємства для управління бізнесом, логістикою і транспортними перевезеннями, проектуванням і моделюванням виробництва продукції, а на технологічному рівні - розпізнавання виробничих ситуацій, зображень об’єктів різного типу та осіб за їх голосом.

В задачах прийнятті маркетингових рішень і розпізнавання образів, які посідають значний обсяг в технологічних та бізнес процесах сучасних комп’ютерно-інтегрованих виробництв, важливу роль відіграють такі

методи ідентифікації об’єктів за їх параметрами шляхом аналізу і класифікації даних, як кластеризація і класифікація. Найбільш розповсюдженими на сьогодні є такі методи кластерного аналізу даних, як метод кластеризації на основі щільності (DBSCAN), метод *k*-внутрішньо групових середніх (*k*-means), кластеризації з використанням гаусових моделей сумішей (GMM), метод середнього зсуву (Mean-Shift), методи ієрархічної агломеративної кластеризації. На сьогодні найбільш досконалим методом класифікації даних є метод опорних векторів (SVM). Даний метод за рахунок використання узагальнених вирішувальних функцій здатний оперувати не тільки з лінійними, а й з нелінійно розподіленими зображеннями об’єктів в просторі ознак.

Недоліком існуючих методів ідентифікації об’єктів за їхніми параметрами шляхом класифікації та кластеризації є відсутність такого зв’язку між ними, що орієнтує класифікатор на використання граничних точок кластерів з метою підвищення його ефективності. Тому задача розробки покращених методів кластеризації і класифікації даних, які використовуються під час ідентифікації об’єктів для підтримки прийняття управлінських рішень, що вирішується в даній магістерській кваліфікаційній роботі, є актуальною.

Метою даної роботи є збільшення ефективності систем прийняття рішень шляхом розробки ефективних методів ідентифікації об’єктів за рахунок застосування покращених методів класифікації і кластеризації.

Об’єктом дослідження є процеси аналізу даних в інтелектуальних комп’ютерних системах управління.

Предметом дослідження є методи, моделі та алгоритми ідентифікації об’єктів шляхом класифікації і кластеризації даних.

Методи дослідження. У процесі дослідження застосовувалися: теорія інформації; методи машинного навчання; теорія алгоритмів; теорія ймовірностей; комп’ютерне моделювання для оцінки адекватності отриманих теоретичних результатів.

Наукова новизна отриманих результатів.

1. Вперше запропоновано метод групування подібних об’єктів в кластери з використанням принципу поверхневого натягу рідин.

2. Вперше запропоновано ідентифікувати об’єкти за їх параметрами шляхом використання модифікованого методу опорних векторів, який використовує координати поверхневих точок кластерів, отриманих на етапі кластеризації методом поверхневого натягу.

3. За результатами дослідження розроблено новий ефективний алгоритм ідентифікації об’єктів в системах прийняття рішень.

Практична цінність. Розроблені методи. моделі та програмне забезпечення дозволили підвищити ефективність управлінських рішень в комп’ютерно-інтегрованих системах управління виробництвом.

Достовірність теоретичних положень підтверджується коректним застосуванням математичних методів, строгістю постановки задач, порівнянням результатів, отриманих за допомогою розроблених у роботі методів, з відомими, та співпадінням результатів, отриманих в машинному експерименті під час тестування програмного забезпечення, з результатами математичного моделювання.

Особистий внесок здобувача. Автором роботи більшість результатів отримана самостійно.

РОЗДІЛ 1. АНАЛІЗ СТАНУ ПРОБЛЕМИ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ В СИСТЕМАХ АВТОМАТИЗАЦІЇ

1.1. Сучасний стан проблеми ідентифікації об'єктів

Під поняттям ідентифікації розуміють процедуру ототожнювання об'єктів, встановлення відповідності об'єкта, що розпізнається, своєму образу [1]. Висловлюючись більш стисло, ідентифікація - це дія встановлення ідентичності. У техніці під ідентифікацією розуміють встановлення відповідності розпізнаваного предмета чи явища своєму образу.

Проблема ідентифікації в сучасних комп'ютерно-інтегрованих системах управління виникає в зв'язку з необхідністю аналізу виробничих ситуацій на їх адміністративному та розпізнавання образів (зорових і сигнальних) на технологічному рівнях. На адміністративному рівні такий аналіз проводиться для моделювання і прийняття рішень з ефективного управління комерційною та організаційною діяльністю підприємств, а на технологічному – для реалізації оптимального виробництва продукції.

Проблемі ідентифікації присвячена велика кількість робіт, які відрізняються як об'єктами ідентифікації, так і алгоритмами і методами її виконання [2, 3]. Безпосередньо самі алгоритми ідентифікації об'єктів залежать від їх природи, тому методи, які лежать в їх основі, пов'язані з вибором апроксимаційного рівняння, що використовує спостережувані дані, вибором критерія якості апроксимації, і відповідного йому методу оптимізації. Внаслідок цього виникли питання: як вибирати ефективний метод ідентифікації для об'єктів різної природи, який би забезпечував гарантоване отримання найкращого в заданих умовах рішення задачі ідентифікації. Для вибору методу ідентифікації об'єктів в інтегрованих системах управління виробництвом розглянемо їх класифікацію.

Термін «метод» використовується для позначення системи прийомів і правил, що використовуються під час вивчення явищ природи, суспільства, мислення [2].

Під способом розуміють дію чи систему дій, які використовуються під час виконання роботи для досягнення поставленої мети [2].

Під час ідентифікації об'єктів застостовуються в більшості такі способи:

- уніфікованих найменувань;
- числових номерів;
- штрихового кодування;
- умовних позначок;
- класифікаційний;
- посилальний;
- описовий;
- описово-посилальний.

Найдавнішим із вказаних способів є **спосіб уніфікованих найменувань**. Для прикладу, найменування планет, морів, рік, озер, гір є унікальними і застосовуються в сполученні "ім'я - об'єкт" чи "об'єкт - ім'я": Чорне море, Маріанська впадина, планета Марс, ріка Буг, озеро Свитязь, гора Арарат тощо. Ці надані унікальні найменування є ідентифікаторами і широко застосовуються в системах обробки інформації про ці об'єкти та у системах керування ними, наприклад: Новодністровська ГЕС, міжнародна космічна станція "Мир", товариство "Люстдорф".

Використання синтагми "об'єкт - ім'я" є обов'язковою умовою ідентифікації, оскільки одне і те ж ім'я може бути надане різним об'єктам: річка Дніпро, кафе "Дніпро", місто "Дніпро" і таке інше.

Досить широко для ідентифікації застосовується **спосіб числових номерів**, які надаються різним об'єктам. Наданий об’єкту номер у сполученні з його найменуванням дозволяє однозначно ідентифікувати об'єкт. На практиці застосовуються два основних способи ідентифікації за допомогою цифрового номера: серійно-порядковий і порядковий.

Поширеним прикладом серійно-порядкового способу є ідентифікація дат. Наприклад, дата 17.12.2022 ідентифікує 17-те число грудня місяця 2022 року. Серійно-порядковий спосіб широко

застосовується у вітчизняній паспортній системі для ідентифікації паспортів. В ньому кожен паспорт має у межах виділеної серії уніфікований шестизначний номер, наприклад, паспорт має серію АА і порядковий номер 628 284.

Порядковий номер, що привласнюється об'єкту, встановлює орган, який здійснює нумерацію. Наприклад, нумерацію будинків на вулиці встановлюють комунальні служби, автобусів - керівництво автотранспортним рухом, залізничних потягів - встановлює керівництво залізничним транспортом. Номери об'єктам часто надають у міру їх появи, обліку чи реєстрації: номер автомобіля в черзі на заправці, номер поштомату в «Новій пошті», нумерація шкіл у міру їх появи, табельний номер працюючого на деякому підприємстві по мірі його реєстрації, тощо.

Перевага порядкового номерного способу полягає в забезпеченні короткої і простої (за кількістю знаків) ідентифікації об'єкта. До недоліків можна віднести неінформативність, викликану відсутністю ознак, що характеризують об'єкти з наданими порядковими номерами. В якійсь мірі цей недолік усувається при використанні серійно-порядкового ідентифікаційного номера. Наприклад, використання цього методу ідентифікації для нумерації кімнат шляхом доповнення номера кімнати номером поверху: номери 2.13 і 4.13 означають кімнату номер 13 на другому поверсі та кімнату номер 13 на четвертому поверсі.

Інформаційно більш насиченою є ідентифікація способом з використанням порядкових номерів у поєднанні з класифікаційними угрупованнями, яка прийнята у ЗКП - Загальнодержавному класифікаторі продукції. Наприклад, для ідентифікації автомобіля вантажного ГАЗ 53А служить код ЗКП 45 1113 2111. В ньому є класифікаційна частина коду -перші шість знаків 45 1113, яка дозволяє з класу 45 "Вироби автомобільної промисловості", до якого відносять також велосипеди, мотоцикли та автобуси, виділити конкретно автомобілі 451, вантажні 4511, загального призначення 45111, вантажопідйомністю від 1,5 т до 3 т (451113). Кінцеві чотири знаки коду 2111 включають номер, що

надається в автомобільній галузі автомобілю з врахуванням числа ведучих коліс.

Спосіб штрихового кодування згідно з ДСТУ 3144-95 описується як присвоєння товарам умовних позначень у вигляді цифрового, літерно-цифрового та штрихового коду з метою їх ідентифікації.

Найбільшою цариною ефективного використання технологій штрихового кодування є споживчий ринок товарів і послуг. Основне функціональне призначення таких технологій полягає в ідентифікації товарів з використання в оперативному режимі актуальної інформації.

Під час ідентифікації продукції та документів широко використовується **спосіб умовних позначок**. В переважній мірі застосовуються такі три способи: мнемонічний, класифікаційний і мнемо-класифікаційний.

Мнемонічний спосіб умовних позначок дозволяє полегшує людині запам'ятовування та розуміння інформації про продукцію або документи. Так позначення "Електронасос ГНОМ 100-25" містить поряд з назвою об'єкта ("Електронасос") мнемонічне позначення, в якому літера Г означає для брудної води, літера Н - насос, літера О - одноступінчастий, літера М - моноблочний, число 100 - продуктивність 100 $м^3$/годину і число 25 - з тиском 25 мм.

Класифікаційний спосіб застосовується тоді, коли інформація опрацьовується в комп'ютерних системах. Наприклад, з його використанням побудована унікальна класифікаційна система позначення конструкторських документів і виробів, яка виглядає таким чином.

Чотиризначний літерний код організації-виробника (ХХХХ) надається згідно спеціального кодифікатора конструкторських організацій; код класифікаційної характеристики, (наприклад 303 125), який є інформаційною частиною повідомлення, видається згідно класифікатора виробів і конструкторських документів машинобудування та приладобудування (Класифікатор Єдиної Системи Конструкторської Документації - ЕСКД). За такою класифікаційною характеристикою можна

знайти, наприклад, інформацію про "групу однорідних виробів, що розроблювалися різними конструкторськими організаціями з метою запозичення", оскільки в коді класифікаційної характеристики міститься наступна інформація:

- 30 - вироби загального машинобудування; 303 - пристрої, що передають рух; 3031 - редуктори;

- 30311 - циліндричні одноступінчасті;

- 303115 - з міжосьовою відстанню пари зубчастих коліс від 63 до 315 мм включно.

Порядковий реєстраційний номер забезпечує ідентифікацію конкретного редуктора і надається згідно класифікаційної характеристики у межах коду організації-виробника від 001 до 999.

Класифікаційний спосіб зручно використовувати у випадках, в яких ідентифікуються групи однорідних, схожих за своїми описовими параметрами, об'єктів. Його перевага полягає в інформативності - він дозволяє зі значної кількості об'єктів виділяти саме ті, що мають певні ознаки. Він досить широко застосовується в багатьох сферах діяльності, оскільки забезпечує систематизацію об'єктів. Цей метод особливо ефективний для обробки інформації в системах управління, наприклад, якщо є необхідність зібрати інформацію про вантажні автомобілі в логістиці підприємства, види продукції, що користується найбільшим попитом або фактори шкідливих умов виробництва тощо.

Посилальний спосіб застосовується у тих випадках з метою ідентифікації об'єктів, в яких інформація про конкретні характеристики міститься в технічних чи нормативних документах. Він здебільшого використовується для визначення конкретної продукції під час її замовлення. У ньому ідентифікація містить найменування виробу, умовне його позначення та посилання на документ, в якому описані визначені вимоги до виробу, наприклад:

- маргарин за ДЕСТ 240-85;

- соляна кислота за ДЕСТ 3118-77;

- бутилкаучук БК-1675У за ТУ 2294-010-17187505-95.

Якщо документ не надає ідентифікації якогось виробу, то в класифікацію додається інформація про виробника продукції, для прикладу: миючий синтетичний порошкоподібний засіб "Віка" згідно ТУ В6 - 00209651 Вінницького ЗАТ "Реактив". Спосіб має той недолік, що не розкриває використанні основних характеристик та особливостей продукції.

У випадках, коли потрібно ідентифікувати заданий об'єкт шляхом опису його характеристик (показників, властивостей, параметрів), як правило, використовується **описовий спосіб ідентифікації**. В такому способі однорідні об'єкти, які мають однакові найменування, близьку номенклатуру показників і сферу застосування можуть відрізнятися між собою значеннями цих показників. Даний спосіб ідентифікації передбачає використання усіх головних характеристик об'єкта і з їх допомогою відрізняє його від всіх інших однорідних об'єктів.

Описові методи ідентифікації широко застосовуються в закладах медицини у вигляді медичної карти пацієнта, у криміналістиці у вигляді опису злочинця та характеру злочину, у геології як опис мінералу тощо.

До однієї з головних переваг описового методу є можливість повести порівняльний аналіз об'єктів однорідного типу шляхом порівняння їх характеристик, які використовуються під час їхньої ідентифікації. При цьому порівняння можна проводити різними способами, зокрема, автоматизованим. При порівнянні встановлюється ступінь одинаковості чи відмінності об'єктів за їх основними характеристиками. При цьому приймається рішення про вибір, звичайно того об'єкта, який має найкращі показники для даних умов застосування. Даний спосіб також використовується під час вибору однорідних об'єктів з метою їхньої стандартизації та систематизації.

Описово-посилальний спосіб на відміну від описового для ідентифікації використовує тільки частину характеристик об'єкта, при цьому посилається на документ, де наведені всі його основні характеристики.

Цей метод найбільш широко застосовується при створенні баз даних про різні об'єкти, наприклад, про інформаційні видання у вигляді каталогів, кадастрів, покажчиків, тощо. Він дозволяє суттєво скоротити обсяг інформації, потрібний для ідентифікації об'єктів, що істотно економить комп'ютерну пам'ять та скорочую обсяги видання. У каталогах приводять найменування продукції, її призначення та сферу використання, умовні позначення, що використовуються під час замовлення, позначення тих документів, які містять усі вимоги до даної продукції, адресу та найменування виробника, та основні характеристики з їх встановленими значеннями.

Як випливає з проведеного огляду методів ідентифікації об’єктів в найбільшій мірі для розв’язання вказаних раніше задач прийняття рішень на управління в комп’ютерно-інтегрованій системі управління підходять класифікаційний і описовий способи. Описовий спосіб дозволяє шляхом опису його характеристик (ознак) ідентифікувати заданий об'єкт, визначивши його схожість до інших об’єктів і сформувати з них групу (кластер) однорідних. Така процедура називається кластеризацією даних, а групи схожих між собою об’єктів – кластерами (іноді таксонами). При цьому віднесення нового об’єкта до одного з встановлених кластерів здійснюється класифікаційним способом за допомогою процедури класифікації, яка визначає, до якої групи об’єктів він найбільш похожий згідно встановленого критерію близькості.

Таким чином, для вирішення проблеми розробки ефективних методів ідентифікації об’єктів в системах управління проведемо вибір оптимальних для умов автоматизованого виробництва методів кластеризації та класифікації. Оскільки вони відносяться сучасною наукою про штучний інтелект до методів машинного навчання, то проведемо аналіз сучасного стану цих методів.

В сучасних інтелектуальних системах управління переважним напрямком аналізу і опрацювання інформації для підтримки прийняття рішень є використання методів машинного навчання [3, 4]. Переваги

використання методів машинного навчання для підтримки прийняття рішень, а також основні області їх застосування в системах управління соціально- економічними системами та технічними уже були вказані у Вступі до даної роботи. Охарактеризуємо сучасний стан проблеми аналізу даних методами машинного навчання та їх особливості. Зокрема, розглянемо особливості існуючих на сьогодні методів кластеризації і класифікації даних з метою вирішення в магістерській роботі завдання покращення процедури ідентифікації об’єктів для підвищення ефективності прийняття рішень в комп’ютерно-інтегрованих системах управління.

Machine Learning, ML, або машинне навчання є розділом теорії штучного інтелекту, в якому здійснюється пошук методів вирішення різноманітних задач шляхом використання навчання під час вирішення подібних задач [5, 6, 7].

Побудова цих методів здійснюється з використанням засобів алгебри, дискретної математики, математичної статистики, чисельних методів, теорії оптимізації та інших розділів математики.

Машинне навчання останнім часом є головним напрямком теперішніх комп'ютерних технологій. Для фахівця з інтелектуальних технологій (IT) якраз знання основ і практичного застосування методів ML, окрім уміння програмувати, стає стандартом. Терміном Mashine learning на разі визначають клас тих методів штучного інтелекту, для яких характерно навчання в процесі застосування розв’язків множини схожих задач, а не пряме рішення задачі [8, 9, 10]. Таким чином, машинне навчання полягає в отриманні знань з даних. Це область науки, що знаходиться на перетині штучного інтелекту, комп'ютерних наук та статистики, вона також відома як статистичне навчання або прогнозна аналітика . Оскільки поняття штучного інтелекту здебільшого застосовується під час розв’язання задачі розробки інтелектуальних роботів, то термін «машинне навчання» (ML) на сьогодні є більш поширеним, ніж словосполучення «штучний інтелект». Застосування методів машинного навчання стало звичним явищем не тільки в комп’ютерних інтелектуальних системах управління, а й в

повсякденному житті. Велика кількість сучасних веб-застосунків і пристроїв широко використовують алгоритми машинного навчання, починаючи з простого випадку автоматичних рекомендацій з покупки продуктів чи замовлення, перегляду фільмів, і закінчуючи складними процедурами розпізнаванням осіб на фотографіях і персоналізованими онлайн-радіотрансляціями. В роботі таких складних сайтів як Facebook, Netflix або Amazon можна побачити, наявність якихось моделей машинного навчання в їхніх розділах.

На сьогодні методи машинного навчання використовують не тільки в управлінській діяльності підприємств і комерційних за стосунках, а й в наукових дослідженнях, пов’язаних з обробкою даних [11, 12] До таких наукових проблем можна віднести дослідження космосу, відкриття нових частинок, аналіз ДНК, діагностику захворювань в медичній практиці.

Навчання можна розділити два типи:

1. Дедуктивне, що використовує формалізацію експертних знань та їх занесення в комп'ютерну систему прийняття рішень у формі бази знань.

2. Індуктивне навчання, або навчання по прецедентах, яке ґрунтується на знаходженні закономірностей, що існують в даних.

Перший тип навчання використовується в експертних системах, в зв’язку з чим під терміном Machine Learning на разі розуміють навчання на прецедентах.

Сформулюємо проблему навчання з використанням прецедентів таким чином. Існує набір об'єктів чи ситуацій або явищ та низка альтернативних відповідей (реакцій) [13]. Прогнозується, що між об'єктами та відповідями існує залежність, яка апріорі є невідомою. Встановлено тільки визначений набір прецедентів - пар «об'єкт, реакція», названий навчальною вибіркою. Ґрунтуючись на цих даних треба встановити неявну залежність за допомогою побудови алгоритму, здатного достатньо точно прийняти рішення про його класифікацію для всякого можливого вхідного об'єкта. Методи ML дозволяють реалізувати емпірично сформоване рішення там, де залежність аналітично не обов'язково

виражається. Тут є важливою властивість навченої системи до узагальнення, тобто правильної реакції на ті дані, які до навчальної вибірки не належать. Для визначення адекватності розроблених моделей відповідей використовується вибраний критерій якості.

З однієї сторони розділ Machine Learning історично увібрав в себе методи математичної статистики, а з іншої сторони завдячує поділу нейро-мережевих технологій на два напрямки: топології архітектури нейро-мереж та методів їх навчання [14]. На сьогодні відомі способи класифікації методів навчання випливають лише зі способу використання нейро-мереж і на теперішній час є застарілими. Класифікацію методів ML потрібно виконати для того, щоб мати можливість орієнтуватися у виборі потрібного методу для розв'язання конкретно поставленої задачі. На рис. 1.1 наведено один із альтернативних варіантів класифікації відомих на теперішній час методів машинного навчання. Зі схеми рис. 1.1 витікає, що поставлені в роботі задачі розробки ефективних методів і кластеризації та класифікації даних відносяться до класичного навчання.

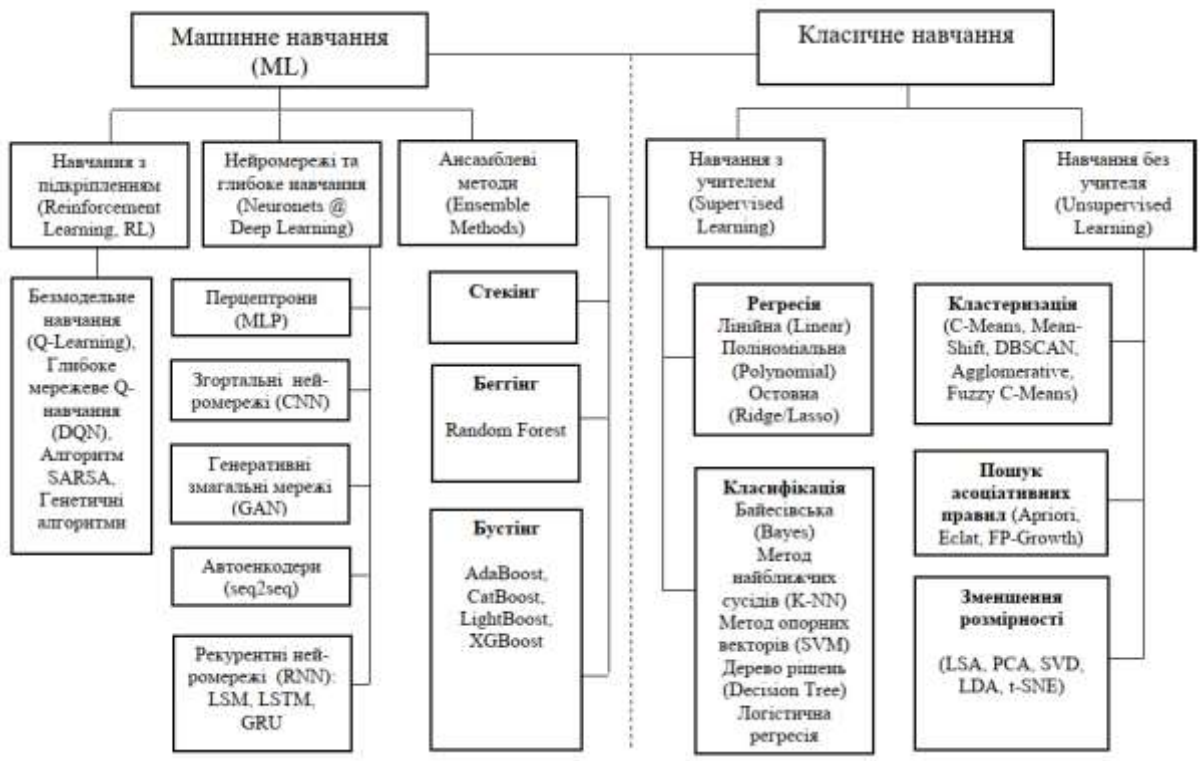


Рисунок 1.1 – Класифікаційна схема методів машинного навчання

Методи класичного навчання відрізняються від методів суто машинного навчання тим, що для їх реалізації потребують традиційне програмування.

Розглянемо відмінності між ними більш детально.

Для вияснення того, як здійснити реалізацію ефективної ідентифікації об’єктів в системах управління за допомогою запропонованих методів кластеризації і класифікації наявних в розробника даних про ці об’єкти, проаналізуємо їх відмінності від методів суто машинного навчання [15, 16].

Основна відмінність класичного навчання від суто машинного полягає в необхідності використання тля його реалізації традиційного програмування. При цьому слід відзначити, що машинне навчання не завжди може замінити традиційне програмування. Наприклад, створити сайт інженеру веб-розробнику за допомогою алгоритмів машинного навчання неможливо. Тому машинне навчання і штучний інтелект використовують в якості доповнення до стандартних інструментів програмування. Для прикладу, для системи трейдингу прогнозний алгоритм можна розробити з використанням машинного навчання, а інтерфейс, візуалізацію даних та інші частини реалізовують однією з мові програмування (Java, Python, R та інш).

Тому машинне навчання як правило використовують там, де стандартні методи програмування для вирішення задачі не є ефективними. Щоб отримати рішення задачі шляхом традиційного програмування, інженеру потрібно розробити алгоритм і написати програму. Після цього він задає початкові дані програми і отримує результат. Отже під час рішення задачі традиційним програмуванням інженер-розробник сам створює алгоритм розв’язання, який вводить набір параметрів і на підставі вхідних даних має можливість спрогнозувати результат роботи моделі.

Оскільки людині-розробнику дуже складно працювати з великим набором параметрів, тоді вона може за допомогою традиційного програмування побудувати тільки дуже просту модель. Під час

використання суто методу машинного навчання для вирішення тієї ж задачі інженер-розробник використовує інший підхід. Алгоритм вирішення задачі при цьому ним самостійно не розробляється, а спочатку збирається навчальна вибірка апріорних даних, за тим її використовують для побудови моделі машинного навчання.

Зібрана представницька вибірка навчальних даних завантажується в різні алгоритми машинного навчання і використовуються різні «регулятори», щоб налагодити алгоритм навчання на отримання різних моделей. З отриманих моделей вибирається та, яка володіє узагальнюючими властивостями, тобто може створювати адекватні прогнози при поданні на вхід уже нових даних, які не входили до навчальної вибірки. Використання готових моделей у випадку машинного навчання подібне до процесу, що має місце в стандартному програмуванні, але замість алгоритму використовується готова розроблена модель. Модель, отримавши вхідні дані, виробляє результат. Якраз використання готової відомої моделі є недоліком машинного навчання в тому випадку, коли потрібно реалізувати суттєво нову ідею в моделі обробки даних. Оскільки в нашій магістерській кваліфікаційній роботі поставлена задача розробки більш ефективних, порівняно з існуючими, методів ідентифікації об’єктів, за рахунок нової ідеї кластерного аналізу методом поверхневого натягу і відповідної до нього модифікації методу класифікації типу SVM, то в даній роботі ми обираємо шлях вирішення цієї задачі покращеними методами класичного навчання з використанням традиційного програмування.

1.2. Аналіз методів класифікації образів

Класифікація об’єктів за принципом мінімуму відстані Класифікація образів за допомогою функцій відстані - одна з перших ідей автоматичного розпізнавання образів. Цей простий метод класифікації виявляється дуже ефективним інструментом при рішенні таких задач, у яких класи характеризуються ступенем мінливості, обмеженої в розумних межах.

У ситуаціях, коли вектори вимірів (образи) у кожнім класі будуть

майже ідентичні, класифікатори, що діють за принципом мінімальної відстані, можуть виявитися надзвичайно ефективним засобом рішення задачі класифікації [17, 18, 19]. Найбільш наглядним способом введення міри подібності для векторів об‘єктів, які зображуються точками в просторі ознак, є визначення відстані між ними. Наприклад, на рис. 1.2, а) видно, що інтуїтивно точку *x* можна віднести до класу Ω_1 виключно з тих міркувань, що цей вектор знаходиться ближче до точок цього класу.

Використання відстані для класифікації дає задовільні практичні результати тільки в випадку компактного розміщення кластерів (рис. 1.2, а). В іншому випадку (рис. 1.2, б) зарахування зображень в один із класів на ґрунті оцінки його близькості до об‘єктів відповідного класу викликає ускладнення.

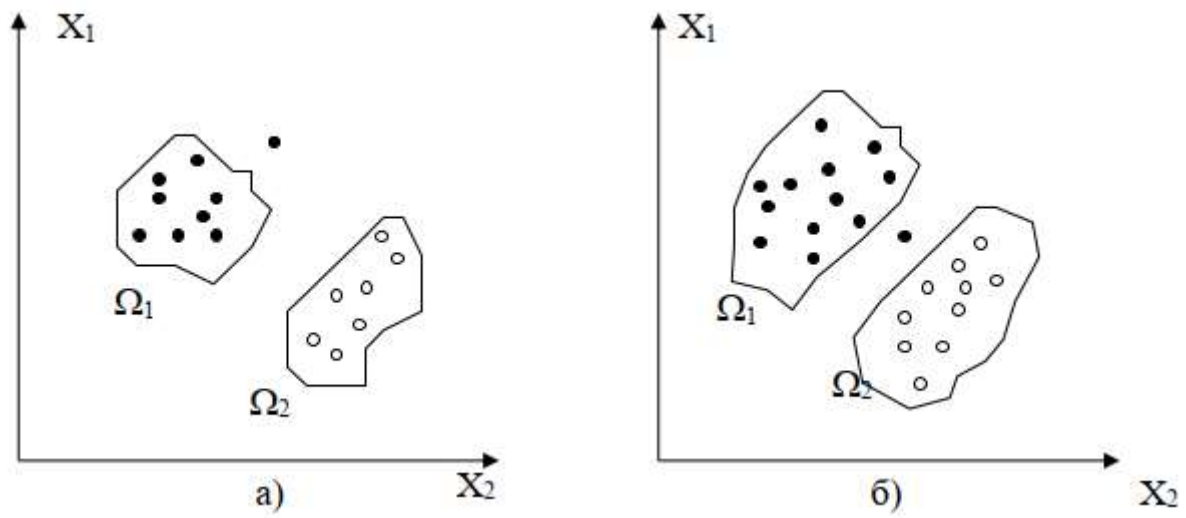


Рисунок 1.2 – Класифікація об‘єктів за принципом мінімуму відстані: а – сприятливе розміщення об’єктів *x*; б – несприятливе

Байесівська класифікація образів.

Байесівський підхід до класифікації образів передбачає побудову класифікатора на основі статистичних властивостей класів [20, 21, 22]. Статистичний аналіз надає можливість побудувати правило класифікації, яке є оптимальним з точки зору гарантування в середньому найменшої імовірності здійснення помилки класифікації. Це оптимальне в статистичному сенсі правило класифікації слугує зазвичай стандартом, з

яким почасти здійснюється порівняння якості інших класифікаційних алгоритмів.

Байєсівський класифікатор забезпечує віднесення образа $\boldsymbol{x}$ до класу з найменшим значенням середніх втрат $\boldsymbol{r}$. Тому образ $\boldsymbol{x}$ зараховується в клас ω_1, якщо виконується умова $r_1(x) < r_2(x)$; це повинно означати, що

$$L_{11}p(x|\omega_1)p(\omega_1) + L_{21}p(x|\omega_2)p(\omega_2) < L_{12}p(x|\omega_1)p(\omega_1) + L_{22}p(x|\omega_2)p(\omega_2), \quad (1.1)$$

чи, що те ж саме,

$$(L_{21} - L_{22})p(x|\omega_2)p(\omega_2) < (L_{12} - L_{11})p(x|\omega_1)p(\omega_1). \quad (1.2)$$

У формулах (1.1) і (1.2) елемент матриці L_{ij} визначає втрати від прийнятого рішення на класифікацію i-го об’єкта як j, $p(x|\omega_1)$ - функція правдоподібності для класу $\omega_i, r_i(x)$ - середні втрати для пред'явленого образа x, $r_j(x) = \sum_{i=1}^{M} L_{ij}(x|\omega_i)\, p(\omega_i)$, $p(\omega_i)$ – ймовірність появи образів класу.

Перевагою байєсівского методу є те, що є можливість оцінки ймовірності помилки для кожного рішення. До недоліком потрібно віднести потребу в деякій попередній інформації, а саме знання платіжної матриці втрат від прийняття відповідних рішень та апріорних ймовірностей класів і функцій щільностей ймовірності розподілу ознак для кожного з них.

Класифікація а допомогою дерева рішень.

Одним із широко розповсюджених методів прийняття рішень є використання моделі дерева рішень. Під “деревом рішень” розуміють графічне зображення послідовності рішень відносно станів навколишнього 28 середовища із зазначенням на кожному кроці (у вузлах дерева) умов і прийнятих відповідних рішень і іноді і виграшів для них (на гілках дерева, що виходять з вузлів для всіх можливих комбінацій альтернатив і станів середовищ. Приклад дерева рішень наведено на рис. 1.3.

Одним із поширених методів прийняття рішень є використання моделі дерева рішень. «Дерево рішень» розуміється як графічне представлення серії рішень, пов’язаних із станом навколишнього середовища, із

зазначенням на кожному кроці (у вузлах дерева) умов і відповідних прийнятих рішень, а іноді і результатів для них (на гілках дерева). Рис. 1.3 показує приклад структури дерева, що містить вузли і гілки для усіх можливих комбінацій альтернатив.

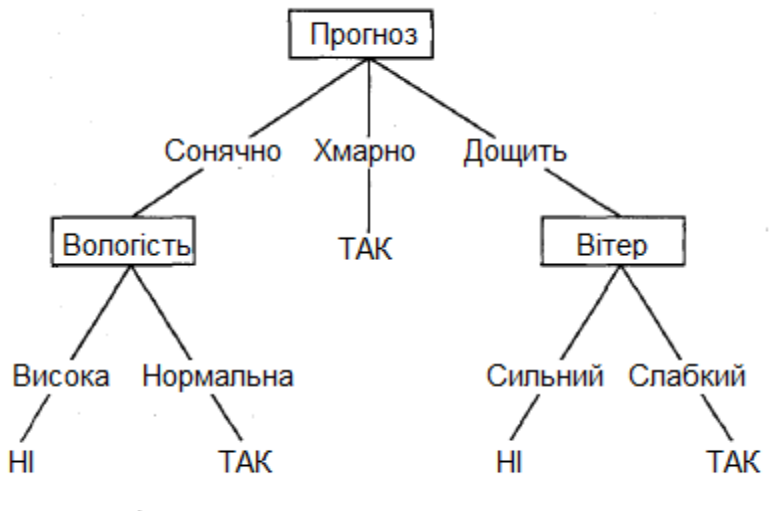


Рисунок 1.3 – Дерево прийняття рішення “Грати в теніс”

Під час побудови вказаної моделі дерева використовують його навчання на навчальній вибірці даних, опис яких залежить від природи предметної області, для якої вирішується задача. Отримана модель навченого дерева рішень відповідає деякій дискретній цільовій функції.

Іншими словами: кажуть, що комп’ютерна програма навчається при розв’язанні задачі, вибраної з набору класів задач *T*, якщо її ефективність *P* підвищується з накопиченням досвіду *E* відповідно до обраного критерію *P*.

До найпопулярніших класів задач *T* у машинному навчанні відносяться вже розглянуті в попередніх розділах:

- Класифікація - віднесення об'єктів до однієї з груп (класів) на основі їх характеристик;

- Регресія - прогнозування кількісних характеристик об'єкта на основі інших характеристик об'єкта;

- Кластеризація – поділяє набір об’єктів на групи на основі характеристик цих об’єктів таким чином, щоб об’єкти всередині групи були

схожі один на одного, а об’єкти за межами тієї самої групи менш подібні.

Окрім названих, поширеним класом задач є знаходження аномалій в наборах даних, тобто об’єктів або їх груп, сильно несхожих на інші в цьому наборі (це так званий клас методів “Data Mining”), та багато інших [24].

На додаток до описаних вище, розповсюдженим класом завдань є виявлення аномалій у наборах даних, тобто об’єктів чи груп об’єктів, які дуже відрізняються від інших у цій колекції (це так звані методи «інтелектуального аналізу даних Data Mining»), і багато інших [24].

Досвід *E* алгоритмом навчання отримує шляхом аналізу впливу даних на одержаний результат навчання. Алгоритми машинного навчання як було попередньо визначено можна поділені на навчання з учителем (контрольоване навчання) і навчання без учителя (неконтрольоване навчання) в залежності від використання у навчальній вибірці маркованих чи немаркованих даних відповідно. Тобто в контрольованому навчанні (supervised lerning) належність до того чи іншого класу для кожного об’єкта навчальної вибірки відома, в неконтрольованому навчанні (unsupervised learning) така інформація невідома. В практичному прикладі в завданні прогнозу неповернення чи повернення кредиту деяким позичальником під досвідом *E* буде отримана з попередньої кредитної історії навчальна вибірка з таких даних: набір позичальників, описаних таким набором ознак як тип кредиту, вік, зарплата і інші, а також цільовим класом, який описується фактом неповернення (0) або повернення (1) кредиту. В даному прикладі вирішується задача бінарної класифікації. У випадку наявності інформації про час відтермінування клієнтом повернення кредиту на основі чого теж саме треба спрогнозувати для нових клієнтів, вирішується задача регресії.

Ще однією, третьою складовою у наведеному визначенні машинного навчання є критерій (метрика) *P* оцінки ефективності алгоритму. Різні задачі і відповідним алгоритми використовують різні критерії, вигляд яких залежить від опису вирішуваної проблеми. З них найпростішою

метрикою якості класифікаційного алгоритму точність (accuracy), тобто частка правильних прогнозів даного алгоритму на тестовій вибірці.

Оцінка якості класифікаційної моделі у вигляді дерева рішень, характеризується основними двома показниками: точністю розпізнавання і помилкою.

Точність розпізнавання визначається відношенням правильно класифікованих в процесі навчання об'єктів до їх загальної кількості у навчальній вибірці.

Помилка розраховується відношенням неправильно класифікованих в процесі навчання об'єктів до їх загальної кількості у навчальній вибірці.

Як правило, навчання моделі дерева прийняття рішень підходить найкраще для проблем з такими характеристиками:

- Об’єкти, що класифікуються, описані у вигляді ознака-значення, при цьому використовується фіксований набір ознак (наприклад, на рис. 1.3 Вологість) і їх значеннями (наприклад, Висока). В найпростішому випадку навчання дерева рішень кожна ознака має невелику кількість розбіжних можливих значень (наприклад, Висока, Нормальна, Середня). При цьому є можливість використати ті розширення базового алгоритму, що дозволяють працювати з реальними величинами ознак (для прикладу, числове представлення вологості).

- Функція цілі чи критерій мають дискретні вихідні значення. На рис. 1.3 дерево рішень надає логічну класифікацію Так чи Ні кожному прикладу. Моделі дерева рішень також зручно розповсюдити на функції з більш ніж двома дискретними вихідними значеннями. Суто більше розширення дає можливість навчати цільові функції з реальними величинами, хоча застосування дерев рішень у такому випадку є менш поширеним.

- В тих випадках, де потрібне використання диз'юнктивних ознакових описів. Таким чином, дерева рішень можна представляти диз'юнктивними логічними виразами.

- Для випадків, коли дані для навчання містять помилки, методи

навчання дерева рішень дають можливість усунути помилки у класифікації навчальних об’єктів і помилки в значеннях їх ознак.

- В тих випадках, в яких навчальні дані мають відсутніми значення деяких ознак, використання моделі дерева рішень не приводить до негативних результатів.

Багато практичних завдань відповідають вказаним характеристикам. Навчання за допомогою моделі дерева прийняття рішень використовується до розв’язання таких проблем, як визначення несправності обладнання за відомою причиною, класифікації медичних пацієнтів за історіями їх захворювань, визначення позичальників кредитів за ймовірністю сплати платежів. Проблеми, в яких треба класифікувати приклади в одну з можливих категорій дискретного набору, називають проблемами дискретної класифікації. Оскільки в системах управління ідентифікуються здебільшого об’єкти з неперервними значеннями параметрів, що їх описують, то даний метод для вирішення нашої проблеми не підходить.

Класифікація образів з використанням опорних векторів.

Машину опорних векторів (Support Vector Mashine - SVM) на сьогодні визнана одним із найкращих класифікаторів, що дозволяє реалізувати і лінійну і нелінійну класифікацію [25, 26].

Метод опорних векторів (SVM) є алгоритмом контрольованого навчання, і він використовується в більшості для завдань класифікації. У цьому алгоритмі кожен елемент даних зображається точкою в n-вимірному просторі (де n - кількість ознак), при цьому величина кожної ознаки задає значення відповідної координати. SVM здійснює класифікацію, використовуючи гіперплощину, що розділяє класи оптимальним способом. *Опорними векторами* називаються ті об'єкти вибірки, які лежать на границях класів [27, 28, 29]. Розподіл між класами буде оптимальним у тому випадку коли відстань між граничними гіперплощинами, які лежать на опорних точках, буде максимальною (рис. 1.4).

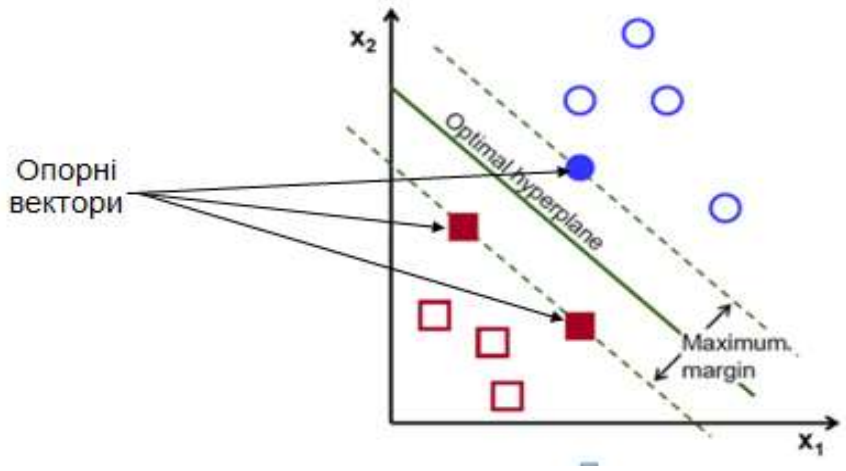


Рисунок 1.4 – Геометрична ілюстрація методу опорних векторів

Навчання класифікатора методом опорних векторів зводиться до задачі оптимізації, що розв'язується застосуванням евристичних алгоритмів. Недоліком цього методу є обчислювальна складність знаходження опорних точок, що є кінцями тих векторів, через які проводяться граничні для класів гіперплощини, для яких будується розподільна площина. Пошук опорних точок здійснюється шляхо розв'язанням оптимізаційної задачі з використанням Лагранжіана такого вигляду [30, 31].

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \tilde{x}_i^T \vec{x}_j, \quad (1.3)$$

де α_i - множники Лагранжа, x_i - вектори навчальної вибірки, $y_i = \pm 1$ – індекси належності точок навчальної вибірки до першого чи другого класів. Пошуку оптимального значення функції L_D шляхом багатократного обчислення скалярного добутку векторів породжує його недолік, що полягає в обчислювальній складність методу SVM.

Проведений в даному розділі огляд методів класифікації, що використовуються для ідентифікації об'єктів в системах прийняття управлінських рішень, показав, що на сьогодні найбільш досконалим є класифікація за алгоритмом опорних векторів SVM. Оскільки для навчання класифікатора даного типу потрібно реалізувати складну оптимізаційну процедуру для виявлення граничних точок в кластерах, яка є неефективною з точки зору обчислювальних затрат, то усунути цей

недолік можна шляхом розробки методів кластерного аналізу, які фіксують поверхневі точки кластера під час навчання [32]. Проаналізуємо особливості найкращих методів кластерного аналізу, відомих на сьогодні.

1.3. Огляд методів кластерного аналізу

Задача кластерного аналізу даних полягає в об’єднанні деякої множини об’єктів в групи (кластери) так, аби об’єкти одного кластера були більш похожі згідно визначеної міри подібності один на іншого, ніж на об’єкти в інших кластерах. Завдання кластерного аналізу відносяться до “навчання без учителя”, тобто до неконтрольованого навчання.

Завдання кластеризації можна сформулювати наступним чином. Задана навчальна вибірка X і міра відстані ρ між об'єктами. Треба розподілити задану вибірку X на групи об’єктів (кластерів) таким чином, щоб до кожного кластера входили об'єкти, близьких по метриці ρ, а об'єкти в різних кластерах істотно відрізнялися. Кожен об’єкт при цьому маркується міткою класу, тобто для всякого $x_i \in X$ приписується номер кластера y_i.

До найбільш популярних на сьогодні алгоритмів кластеризації, відноситься такі, як метод кластеризації за середнім зсувом (*k*-medians), метод *k*-внутрішньо-групових середніх, просторова кластеризація даних з шумом на основі щільності **(** DBSCAN **)**, кластеризація з використанням моделей Гаусової суміші (GMM), агломеративна ієрархічна кластеризація [33, 34].

Проаналізуємо переваги і недоліки на сьогодні найбільш популярних методів кластеризації [35, 36, 37].

Кластеризації даних за середнім зсувом (*k*-medians).

Кластеризація середніх зсувів - це алгоритм, який на основі ковзного вікна намагається визначити щільні області точок даних. Подібно до описаного нижче методу *k*-means, він ґрунтується на центроїді, і має за мету знаходження центральних точок кожного класу. Він працює шляхом послідовної зміни кандидатів на центральні точки таким чином, щоб вони

були середнім значенням точок у розширюваному вікні. Затим утворені вікна- кандидати на етапі подальшої обробки фільтруються для усунення близьких дублікатів, і утворюють кінцевий набір центральних точок та відповідні їм групи близьких до них точок.

Реалізація методу здійснюється алгоритмом методу в такий спосіб.

1. Вибирається за ядро з радіусом *r* початкове кругове розширюване вікно з центром у деякій точці *C* (яка вибирається випадковим чином). Середній зсув реалізується як алгоритм сходження по схилу, що передбачає послідовне переміщення даного ядра в напрямку більш високої щільності на кожному ітераційному кроці до злиття.

2. В кожному ітераційному кроці ковзне вікно зміщується у напрямку областей з більш високою щільністю, при цьому розраховуючи центральну точку як середнє значення точок в даному вікні. Щільність всередині поширюваного вікна є пропорційною кількості точок в ньому. Зміщуючись до середнього значення точок, вікно поступово рухатиметься в напрямку областей точок більшої щільності.

3. Зсув розширюваного вікна здійснюється згідно з середнім значенням до тих пір, поки не буде визначено напрямку, в якому зміщення може вмістити всередині ядра більше точок.

4. Крок з 1 по 3 виконуються пробно з кількома розширюваними вікнами, аж поки всі точки не увійдуть до вікна. З декількох розсувних вікон, що перекриваються зберігається одне вікно, що містить найбільшу кількість точок. Точки даних групуються відповідно до того розсувного вікна, де вони знаходяться.

Перевагою методу порівняно з методом *k*-середніх, є те, що в ньому немає необхідності задавати кількість кластерів, оскільки їх автоматично виявляє середній зсув. Недоліком є початкова невизначеність у виборі розміру радіуса *r* вікна.

Метод *k*– внутрішньо групових середніх.

Алгоритм методу *k*-means можна описати таким чином:

1. На першому кроці задається кількість кластерів і випадковим чином

вибирають їхні центральні точки. Кількість потрібних кластерів визначається з умов задачі, або шляхом застосування нескладного алгоритму кластеризації, наприклад, методу максимальної відстані.

2. На другому етапі для кожної точки з заданого набору обчислюється відстань до кожного з вибраних центрів кластерів. Точка відносять до того центру, до якого вона найближча.

3. На третьому кроці всі точки, зараховані до визначених центрів, використовують для обчислення координат нового центру, знаходячи середнє значення координат цих точок.

4. Вказані кроки повторюють до тих пір, поки не буде отримана задана необхідна точність, або доки центри кластерів не перестануть змінюватися в черговій ітерації.

Перевагою даного методу є простота його реалізації. Основними недоліками є те, що необхідно знати наперед кількість кластерів і те, що він не фіксує поверхневі точки кластеру, координати яких можуть бути корисними в алгоритмах класифікації.

Просторова кластеризація даних з шумом з використанням щільності (DBSCAN).

1. Алгоритм методу DBSCAN починає з довільної точки даних. До цієї точки задають за допомогою відстані епсилон ε деякий окіл. Точки, що знаходяться в заданому околі ε, є точками сусідства.

2. Якщо в цьому околі є достатня відповідно до заданого параметру minPoints кількість точок, то алгоритм розпочинає процес кластеризації, і першою точкою в новому кластері стає поточна точка даних. В протилежному випадку ця точка буде позначена як шум. І в тому і в іншому випадках точка позначається як “відвідана”.

3. Для знайденої першої точки в новому скупченні точки, що лежать в межах відстані ε, теж стають частиною цього скупчення. Після цього процедура, згідно з якою всі точки в околі є входять до одного кластера, повторюється для всіх тих нових точок, які щойно були додані до групи кластерів.

4. Процес для кроків 2 і 3 повторюється до тих пір, доки всі точки в кластері не будуть визначені, що означає, що всі точки в межах околу ε кластера були відвідані та позначені.

5. Після закінчення роботи з поточним кластером, береться нова невідвідана точка і обробляється, і алгоритм виявляє наступний кластер чи шум. Процес повторюється до тих пір, доки всі точки не будуть позначені як відвідані. Оскільки після того, як всі точки будуть відвідані, кожна з них позначається як та, що належить до кластеру, або як шум.

Алгоритм DBSCAN має ряд великих переваг. Перша полягає в тому, що не потрібно задавати певну кількість кластерів. Наступна полягає в тому, що він також приймає викиди за шуми, що відрізняє його від алгоритму від середнього зсуву, який відносить їх до кластера, навіть якщо ця точка даних сильно відрізняється. Ще одна перевага в тому, що він здатний досить добре знаходити кластери довільної величини та різної форми.

Головним недоліком алгоритму є те, що він працює гірше інших для кластерів з різною щільність. Це пояснюється тим, що вибір порогу відстані ε та параметра minPoints кількості точок для визначення точок сусідства буде змінюватися залежно від зміни щільності точок в кластері [38, 39].

РОЗДІЛ 2. МАТЕМАТИЧНЕ ОБГРУНТУВАННЯ ОБРАНИХ МЕТОДІВ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ

2.1. Математичне обґрунтування методу класифікації

Опис класифікації даних методом SVM.

В результаті попередньо проведеного аналізу в якості класифікатора даних, що використовується для ідентифікації об'єктів, було обрано метод опорних векторів SVM. Він широко використовує при розв'язанні завдань регресійного аналізу даних та розпізнавання образів. В інтелектуальних комп'ютерних системах управління розпізнавання образів використовується для ідентифікації спостережуваного об'єкта чи технологічної ситуації з метою виконання алгоритмів, призначених для обробки відомих категорій ситуацій чи об'єктів. В даному застосуванні завдання класифікації образів є центральним, для його вирішення використовується цілий ряд методів і алгоритмів. Як було вказано в розділі 1 обраний метод опорних векторів є одним з найдосконаліших сучасних алгоритмів класифікації, оскільки може працювати як лінійний, так і нелінійний класифікатор при умові застосування відповідних функцій ядра для нелінійного випадку [25, 26].

Алгоритм реалізації методу опорних векторів SVM є дискримінантним класифікатором, робота якого математично ґрунтується на понятті розподільної гіперплощини [27, 28]. Для отриманої навчальної вибірки маркованих даних він будує оптимальну гіперплощину, яка класифікує найкращим способом з точки зору точності нові прецеденти об'єктів. Простий приклад, показаний на рис. 2.1 надає наглядну можливість розуміння оптимальності побудованого у вигляді гіперплощини класифікатора. Для випадку лінійно розподільної множини точок на площині показано принцип пошуку прямої, яка розподіляє два класи найкращим чином [29, 30].

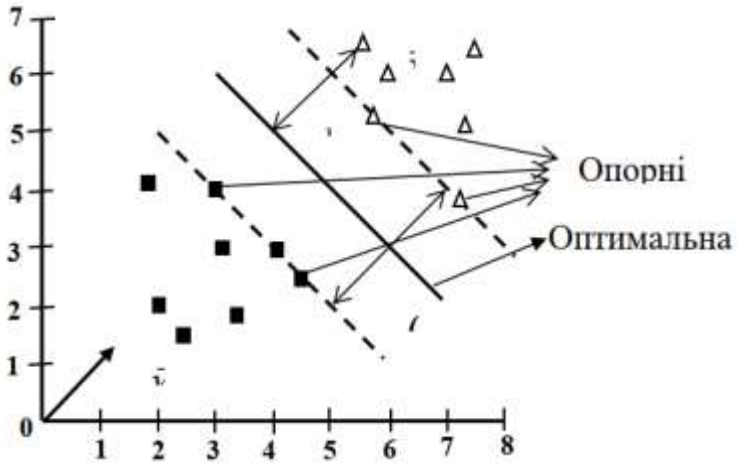


Рисунок 2.1 – Приклад побудови методом SVM оптимальної гіперплощини: r – відстань до точки $\vec{x}$, $\vec{\omega}$ - вектор нормалі до гіперплощини, ρ - маржа

На рис. 2.1 для спрощення розуміння проблеми показано опис об’єктів двох класів у двовимірному просторі ознак (на площині) у вигляді точок, що є кінцями векторів, що є зображеннями даних об’єктів. В цьому простому випадку представлений прямими лініями лінійний класифікатор спрощує наглядність уяви про розділення об’єктів на два класи, тому що тут замість векторів і гіперплощин у просторі вищої розмірності ми маємо справу з точками і прямими. Дане спрощення не виключає можливості застосування таких самих понять до задач класифікації у просторах вищої розмірності [30].

З наведеного прикладу зрозуміло, що є декілька прямих, які дозволяють провести класифікацію. Для вирішення питання, яка з них найкраща, можна інтуїтивно критерій оцінки якості дискримінантних прямих вибрати таким чином: пряма не якісна, якщо проходить занадто близько до точок, оскільки вона буде чутливою до впливу завад і не буде мати узагальнюючого характеру під час класифікації нових точок, яких не було у навчальній вибірці, тобто адекватно їх класифікувати не буде. Тому потрібно знайти такий спосіб побудови, який знайде лінію, яка буде проходити якнайдалі від усіх точок.

В такій постановці, робота алгоритму SVM, по-перше, ґрунтується

на пошуку гіперплощини, що має найбільшу мінімальну відстань ρ до опорних точок. По-друге, в теорії SVM дана відстань отримує назву маржі. Звідси розподільна гіперплощина, що є оптимальною, повинна максимізувати запас маржі для навчальних даних [31, 32].

Математичні засади класифікатора SVM.

Для n-мірного простору узагальнений математичний вираз лінійної дискримінантної (вирішувальної) функції має такий вигляд:

$$f(\vec{x}) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_n x_n = \omega_0 + \vec{\omega}_a^T \vec{x}, \quad (2.1)$$

де: $\vec{\omega}_a = (\omega_1, \omega_2, \ldots, \omega_n)^T$ - вектор нормалі, він же ваговий вектор; ω_0 - зміщення відносно початку координат розподільної гіперплощини; $\vec{x} = (x_1, x_2, \ldots, x_n)^T$- вектор зображення об'єкта.

Проекцію розподільної гіперплощини (2.1) на двовимірний простір зображено на рис. 2.2. На ній показано: $\vec{u}$ - додатньо-орієнтована одинична нормаль; $\vec{p}$ і $\vec{x}$ - деякі точки гіперплощини і простору; 0 – початок координат.

З геометричної точки зору рівняння гіперплощини записується як

$$\vec{u}^T(\vec{u} - \vec{p}), \quad (2.1)$$

або

$$\vec{u}^T \vec{x} = \vec{u}^T \vec{p}. \quad (2.3)$$

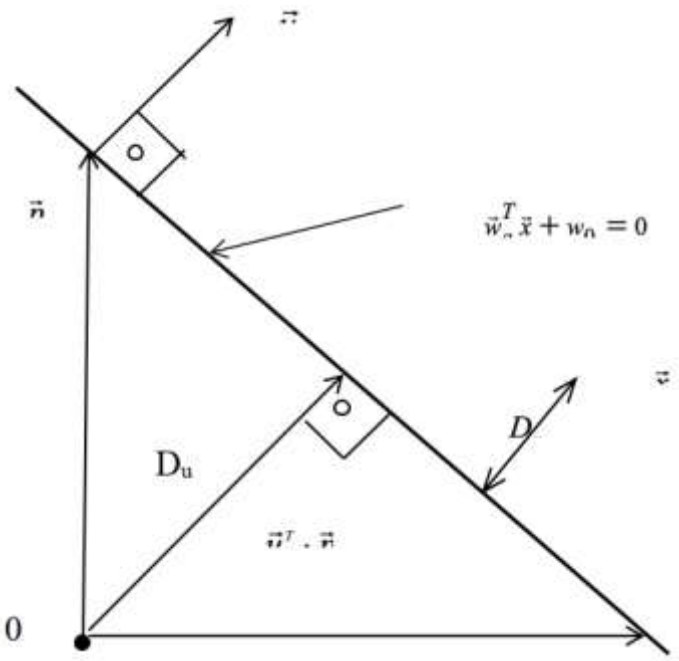


Рисунок 2.2 – Зображення розподільної гіперплощини у вигляді проекції на двовимірний простір

З рівняння (2.1) рівняння гіперплощини можна записати як

$$\vec{\omega}_a^T \vec{x} + \vec{\omega}_0 = 0, \tag{2.4}$$

або

$$\vec{\omega}_a^T \vec{x} = -\vec{\omega}_0 \tag{2.5}$$

Якщо пронормувати ваговий вектор діленням (2.5) на його норму (довжину), то отримаємо

$$\frac{\vec{\omega}_a^T \vec{x}}{\|\vec{\omega}_a^T\|} = -\frac{\omega_0}{\|\vec{\omega}_a^T\|}. \tag{2.6}$$

Шляхом порівняння (2.3) і (2.6) отримаємо вираз одиничної нормалі $\vec{u} = \frac{\vec{\omega}_a}{\|\vec{\omega}_a\|}$, вираз відстані від початку координат до гіперплощини $D_u = |\vec{u}^T \cdot \vec{p}| = \frac{\vec{\omega}_0}{\|\vec{\omega}_a\|}$, вираз відстані від точки простору до гіперплощини $D_x = |\vec{u}^T \vec{x}^* - \vec{u}^T \vec{p}| = \left|\frac{\vec{\omega}_a^T \vec{x}^* + \vec{\omega}_0}{\|\vec{\omega}_a\|}\right|$. Ці визначення будуть використані під час побудови принципів оптимізації процедури навчання класифікатора типу SVM.

Постановка задачі навчання класифікатора за методом SVM.

Задачу навчання класифікатора SVM математичну сформулюємо таким чином [31].

Існують вхід дані для навчання:

$$U = \{(\vec{x}_i, y_i)\}, \vec{x}_i \in R^n, y_i \in \{-1, +1\}, i = 1,2, \dots, m, k = 1,2, \dots, n,, \tag{2.7}$$

де через U позначена навчальна вибірка векторів об'єктів $\vec{x}_i$, для яких відомі марки y_i цих векторів, що визначаються наступним чином: якщо $\vec{x}_i \in \Omega_1$, то $y_i = 1$, а якщо $\vec{x}_i \in \Omega_2$, то $y_i = -1$, отже y_i визначає, до якого класу входить вектор $\vec{x}_i$; R^n - - простір параметрів (ознак) розмірності n; m – число векторів у навчальній вибірці.

За таких умов задачу навчання сформулюємо таким чином: за прецедентами навчальної вибірки побудувати вирішувальну функцію $f(\vec{x})$ таким чином, щоб вона відповідала наступним вимогам:

$$\begin{cases} f(\vec{x}_i) > 0, \vec{x}_i \in \Omega_1 \\ f(\vec{x}_i) < 0, \vec{x}_i \in \Omega_2 \end{cases} \text{ або } y_i \cdot f(\vec{x}_i) > 0, \vec{x}_i \in U\ . \quad (2.8)$$

Одне узагальнене для двох рівняння у виразі (2.8), дозволяє записати лінійний класифікатор на 2 класи таким чином:

$$f(\vec{x}_i) = sign(\vec{w}^T \vec{x}_i + \omega_0). \quad (2.9)$$

Рівність +1 значення виразу (2.9) означає належність об'єкта до першого класу, а якщо -1, тоді до другого. Так як під час навчання значення +1 і -1 для векторів навчальної вибірки є відомими та задаються значеннями y_i, то вираз (2.9) можна записати:

$$f(\vec{x}_i) = y_i \cdot (\vec{w}^T \vec{x}_i + \omega_0). \quad (2.10)$$

Останній вираз визначає відстань (ненормовану) від точок 1-го і 2-го класу до гіперплощини $\vec{w}^T \vec{x}_i$. Цю відстань назвемо маржою, або функціональним зазором. Як було показано в попередньому розділі, нормалізація цього зазору діленням на норму вектора нормалі дозволяє усунути в різних задачах проблему масштабування даних з точки зору відстані. При цьому геометрична маржа буде незалежною від масштабування. Для прикладу, при заміні параметрів $\vec{\omega}$ на $5\vec{\omega}$ і $\vec{\omega}_0$ на $5\vec{\omega}_0$ геометричний зазор не зміниться. При встановленому обмеженні $\|\omega\| = 1$ геометричний зазор збігається з функціональним.

Помноживши для масштабування $f(\vec{x})$ на деяке додатне число з того факту, що $f(\vec{x})$ - лінійна функція, з виразу (2.10) отримаємо:

$$y_i \cdot (\vec{w}^T \vec{x}_i + \omega_0) \geq 1, i = 1, \dots, m. \quad (2.11)$$

З цього витікає, що промасштабована відстань між граничними гіперплощинами $\vec{w}^T \vec{x}_i + \omega_0 - 1 = 0$ і $\vec{w}^T \vec{x}_i + \omega_0 + 1 = 0$, які проходять через опорні точки, дорівнює

$$r = \frac{2}{\|\omega\|}. \quad (2.12)$$

Останній вираз показує, що мінімізація величини $\|\omega\|$ приводить до максимізації маржі. Оскільки норма $\|\omega\|$ включає в себе квадратний корінь, то вирішити в такому вигляді задачу оптимізації такої функції важко.

Процедуру оптимізації можна спростити, якщо замінити $\|\omega\|$ на $\|\omega\|^2$ та п о множити на коефіцієнт 1/2, що не приведе до зміни результату. В такому вигляді маємо квадратичну задачу оптимізації [32, 33].

Задача оптимізації, таким чином, зведеться до пошуку вагового вектора, який максимізує *r*:

$$\frac{1}{2}\|\omega\|^2 \to min \text{ при обмеженнях } y_i \cdot (\vec{w}^T \vec{x}_i + \omega_0) \geq 1, i = 1, \dots, m. \tag{2.13}$$

Задачу з обмеженнями можна представити у вигляді задачі без обмежень шляхом введення множників Лагранжа:

$$\min_{\omega,\omega_0} \max_{\alpha \geq 0} \left\{ \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\vec{w}^T \vec{x}_i + \omega_0) - 1) \right\}. \tag{2.14}$$

Задачу пошуку оптимальних опорних точок розв'язують шляхом введення функції-лагранжіана:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \vec{x}_i^T \vec{x}_j^T \tag{2.15}$$

при умовах

$$\sum_i \alpha_i y_i = 0, \alpha_i \geq 0 \; \forall_i (i = \overline{1, \dots, m}),$$

де α_i - множники Лагранжа.

Розв'язок поставленої задачі має такий вигляд:

$$\vec{w} = \sum \alpha_i y_i \vec{x}_i. \tag{2.16}$$

З (2.16) витікає, що ваговий вектор для оптимальної гіперплощини можна отримати з виразу:

$$\vec{w} = \sum_{i=1}^{k} \alpha_i y_i \vec{x}_i, \tag{2.17}$$

де i – номер опорної точки (вектора), *i*=1…*k*; *k* – число шуканих опорних точок, α_i - множники Лагранжа, визначені для опорних точок процедурою оптимізації.

Алгоритмізація процесу навчання класифікатора SVM

В обчислювальному плані алгоритм пошуку величин множників Лагранжа для побудови оптимальної гіперплощини за виразом (2.17) зводиться до розв'язку системи лінійних алгебраїчних рівнянь, яку отримують шляхом визначення частинних похідних для заданої функції

Лагранжа по вибраних множниках α_i при відомих значеннях координат опорних точок x_{ij} та значень марок y_i:

$$\begin{vmatrix} n & \sum_{i=1}^{n} x_{1i} & \sum_{i=1}^{n} x_{2i} & \dots & \sum_{i=1}^{n} x_{mi} \\ \sum_{i=1}^{n} x_{1i} & \sum_{i=1}^{n} x_{1i}x_{1i} & \sum_{i=1}^{n} x_{1i}x_{2i} & \dots & \sum_{i=1}^{n} x_{1i}x_{mi} \\ \sum_{i=1}^{n} x_{2i} & \sum_{i=1}^{n} x_{2i}x_{1i} & \sum_{i=1}^{n} x_{2i}x_{2i} & \dots & \sum_{i=1}^{n} x_{2i}x_{mi} \\ \dots & \dots & \dots & \dots & \dots \\ \sum_{i=1}^{n} x_{mi} & \sum_{i=1}^{n} x_{mi}x_{mi} & \sum_{i=1}^{n} x_{mi}x_{2i} & \dots & \sum_{i=1}^{n} x_{mi}x_{mi} \end{vmatrix} \cdot \begin{vmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_m \end{vmatrix} = \begin{vmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} y_i\, x_{1i} \\ \sum_{i=1}^{n} y_i\, x_{2i} \\ \dots \\ \sum_{i=1}^{n} y_i\, x_{mi} \end{vmatrix}. \quad (2.18)$$

Алгоритм навчання класифікатора типу SVM наведений на рис. 2.3.

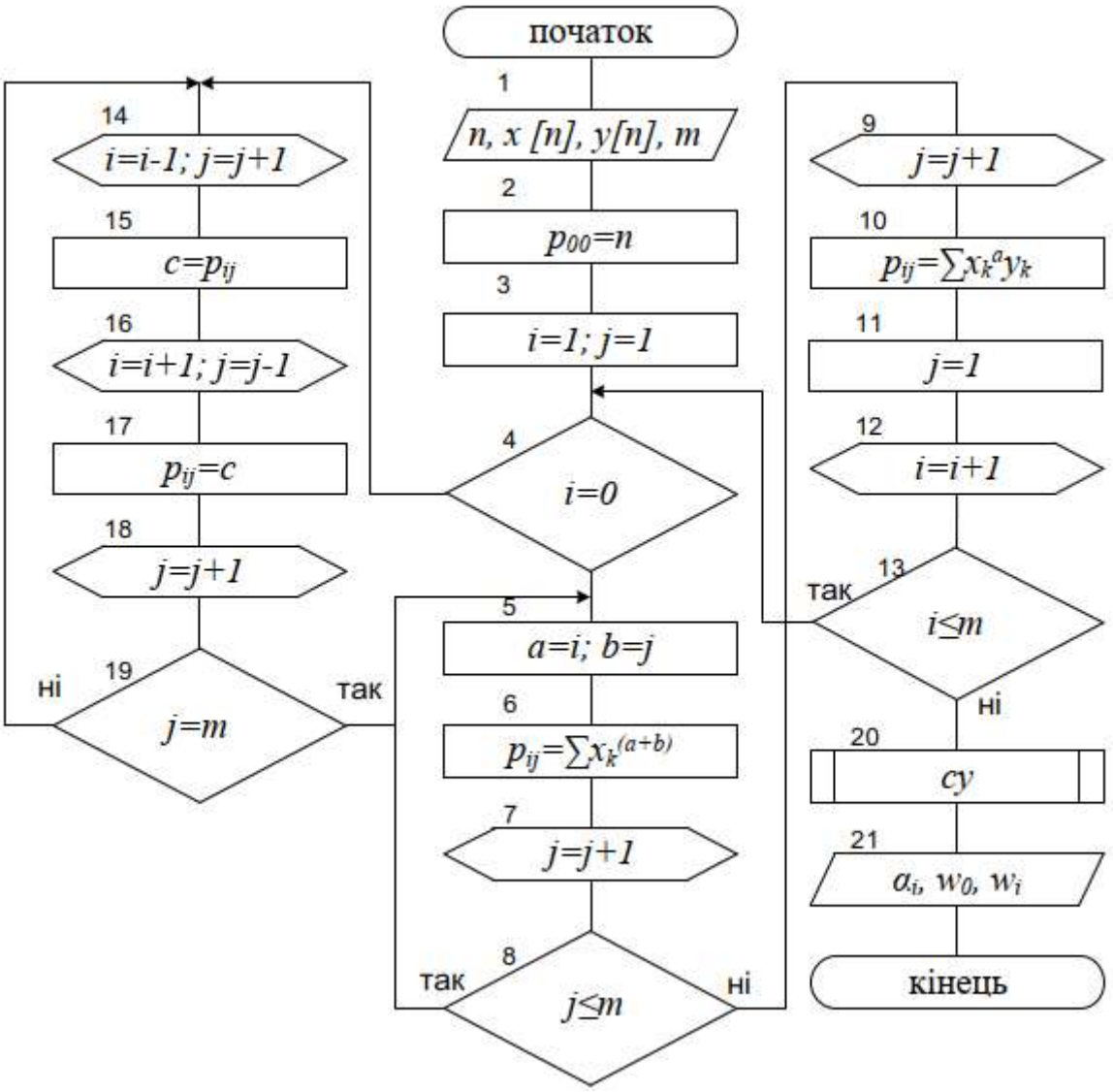


Рисунок 2.3 – Схема алгоритму навчання класифікатора типу SVM

Вхідними даними до алгоритму навчання є навчальна вибірка, що складається з масивів векторів об'єктів $\vec{x}[n]$ і їх міток $y[n]$, де n задає кількість елементів в навчальній вибірці, а m – розмірність ознакового простору.

Алгоритм складається з чотирьох основних кроків: введення даних вхідних даних; обчислення сум, що визначають коефіцієнти перед невідомим в системі (2.18); розв'язання одержаної системи та друк результатів, якими є числові значення множників Лагранжа α_i та коефіцієнтів w вагового вектора. Детальний вигляд алгоритма показаний на рис. 2.3. Під час обчислення коефіцієнтів p_{ij} при невідомих в блоці 6 алгоритму використовується властива для систем (2.18) закономірність

$$p_{ij} = \sum_{k=0}^{n} x_k^{(i+j)}.$$

Вільні члени рівнянь обчислюються в блоці 10 алгоритму за формулою

$$p_{ij} = \sum_{k=0}^{n} x_k^i y_k.$$

Скорочення кількості обчислень досягнуто за рахунок того, що повністю обчислюються: 1-ша стрічка, остання і передостання колонки матриці системи, що показано в блоки 4 і 13. В блоках 1, 6 і 10 дії наведені в спрощеному вигляді. В програмі вони реалізуються у вигляді циклічних процесів. В алгоритмі використані наступні позначення: i, j – індекси, якими позначаються номер стрічки и колонки відповідно; k – індекс, що позначає номер елемента в масиві x або y; p_y – коефіцієнти, які замінюють відповідні суми в системі рівнянь (2.18).

Після знаходження множників Лагранжа α_i алгоритмом обчислюється за формулою (2.17) ваговий вектор $\vec{\omega}$, що визначає в просторі ознак кут нахилу гіперплощини. Величина зміщення ω_0 оптимальної гіперплощини обчислюється за всіма граничними опорними векторами як медіана з:

$$\omega_0 = med\{(\vec{\omega}, \vec{x}_i) - y_i: \quad \alpha_i > 0, i = \overline{1,k}\}. \tag{2.19}$$

2.2. Розробка методу кластеризації даних

Завдання кластерного аналізу даних. Модель класифікації за

алгоритмом SVM, що описана в попередньому розділі, так само як і моделі інших класифікаційних методів, можна будувати при наявності навчальних вибірок, що представляють зразки відомих класів. В зв'язку з цим перед машинним навчанням моделі класифікатора потрібно виконати процедуру кластеризації даних. Її реалізують шляхом неконтрольоване навчання.

Як попередньо уже було визначено, під кластерним аналізом в загальному випадку розуміють групування деякої множини об'єктів таким чином, щоб об'єкти однієї групи (яку називають кластером) були більш похожі (у визначеному сенсі) один до одного, ніж до об'єктів в інших групах (кластерах) [30]. Якщо для заданої множини об'єктів відсутня будь-яка інформація відносно груп (класів), на які вони поділяються, задачу ідентифікації класів можна сформулювати як задачу неконтрольованого навчання, або "навчання без учителя".

Вказану задачу навчання без учителя вирішують шляхом застосування до заданої вибірки даних алгоритмів кластеризації. Отримані при цьому кластери можна вважати класами зображень. Наявність цих класів дає можливість використати їх для отримання дискримінантних (вирішувальних) функцій шляхом використання алгоритмів контрольованого навчання ("навчання з учителем"). Також отримані на стадії навчання без учителя центри кластерів можна безпосередньо використати для побудови класифікатора, що функціонує за принципом мінімуму відстані [32].

Оскільки більшість алгоритмів кластеризації базуються на апріорному досвіді розробника і його інтуїції, отриманих в результаті проведених раніше експериментів, то їх можна назвати неконтрольованим навчанням лише умовно.

Розробка алгоритму кластеризації за принципом поверхневого натягу.

Наразі існує значна кількість методів і відповідних їм алгоритмів кластеризації даних (більше 100). До них можна віднести як класичні методи, які базуються на узагальненому алгоритмі К. С. Фу [24], так і новітні, що зумовлені структурою і природою даних і ціллю їх

використання [25, 26]. Всім цим методам і алгоритмам кластерного аналізу властиві свої недоліки і переваги, які зумовлені орієнтацією на розв’язання вибраного класу задач. До загального недоліку можна віднести те, що вони у просторі ознак формують кластери точок, але їхні границі не фіксують. Остання характеристика Наявність маркованих граничних точок кластера є дуже корисною під час побудови класифікатора. В роботі [34] авторами запропоновано метод пошуку опорних точок для класифікатора SVM шляхом знаходження граничних точок кластера за аналогією з фізичним принципом поверхневого натягу, властивого рідинам.

Сутність казаного явища ілюструється на рис. 2.4, на якому зображена фізична природа натягу в поверхневому шарі рідини, зумовленого нескомпенсованим притяганням внутрішніми молекулами поверхневих [35].

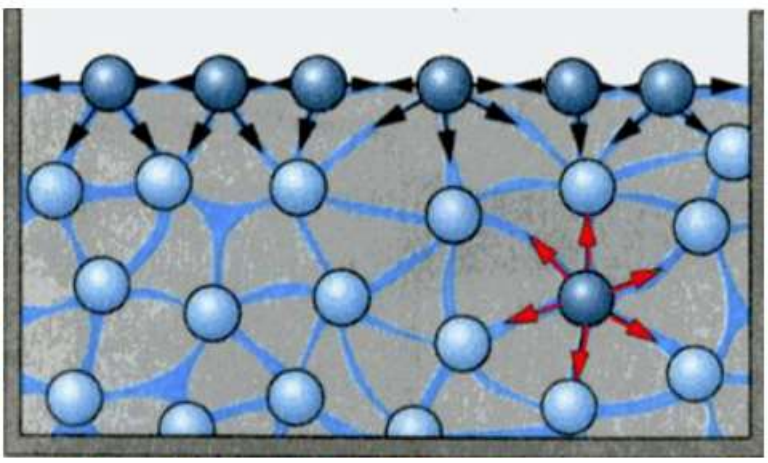


Рисунок 2.4 – Демонстрація поверхневого натягу молекул в рідині

В рідині на кожну молекулу діють сили притягання від сусідніх молекул. Для молекул, що знаходяться всередині рідини, ці сили взаємно скомпенсовані. Рівнодія сил притягання, що діє на молекули, що знаходяться на поверхні розділення середовищ рідина-повітря, напрямлена усередину рідини. В поверхневому шарі, товщина якого дорівнює радіусу дії молекулярних сил, молекули чинять тиск на рідину, стягуючи її поверхню до мінімуму. Це явище називається поверхневим натягом.

Ідея запропонованого авторами підходу до кластерного аналізу

даних полягає в використанні аналогії з описаним явищем повеневого натягу рідини. В якості поверхневих точок кластера вибираються ті, що неврівноважені силами взаємного впливу оточуючими точками (оскільки вони на поверхні кластера). Із цих знайдених точок вибирають точки сусідніх класів, між якими відстані найменші, вони і будуть опорними. У вказаному підході приймаються такі припущення:

а) вважається, що модулі сил взаємодії точок простору обернено пропорціональні відстані між ними:

$$f(\vec{x}_i, \vec{x}_j) = \frac{p}{d}, \tag{2.20}$$

де p є ваговим коефіцієнтом; d - відстань між точками, яка обчислюється за формулою

$$d = \sqrt{\sum_{k=1}^{n} x_{ik}^2}. \tag{2.21}$$

У виразі (2.21) через x_{ik} позначено k-у координату i-ої точки вибірки.

б) на вибрану окрему точку діють сили притягання тільки від найближчих точок.

Проведений в роботі [32] аналіз результатів машинного експерименту, показав, що запропонований підхід можна використовувати як для оптимізації процедури навчання класифікатора, так і для кластерного аналізу даних шляхом фіксації границі кластерів поверхневими точками.

В даній роботі проведена модифікація запропонованого у згаданій роботі алгоритму з метою його використання для програмної реалізації побудови кластерів шляхом використання методом поверхневого натягу для визначення поверхневих точок.

Задачу кластеризації даних шляхом фіксації граничних точок кластера в загальному випадку можна сформулювати наступим чином [40].

Відомі масив об’єктів X і масив номерів (міток) кластерів Y. Задана у просторі ознак $\mathcal{R}^{\mathrm{T}}$ вибірка зображень об’єктів деякою підмножиною точок $X^m = \{x_1, \dots, x_m\} \subset X$, де n – розмірність простору ознак, m – кількість точок у вибірці. Також задана метрика $D(x_1, x_2)$ для відстані міру відстані між точками і задана функція натягу (стрес) $\Delta(x_k, z_k)$ між вибраною поточною

точкою x_k і групою найближчих до неї точок, де z_k - центр вибраної групи точок. По цій заданій інформації треба розділити задану вибірку об'єктів на непересічні кластери таким чином, щоб в кожному кластері були тільки близькі за метрикою *D* точки. Також потрібно при цьому фіксувати в масиві S_{Y_i} точки, які лежать на поверхні кластера (тобто є найбільш віддалені від його центру), де Y_i - кластер з номером *i*.

Алгоритм класифікації розробляється у вигляді вирішувальної функції $f: X \to Y$, яка ставить у відповідність номер кластера $y \in Y$ будь-якому об'єкту $x \in X$, причому для кожного кластера фіксується підмножина поверхневих точок $S_{Y_i} \in X$. В більшості випадків множина кластерів *Y* наперед не є відомою, при цьому потрібно вирішити задачу пошуку оптимальної кількості кластерів.

Побудова такого алгоритму при узагальненому підході вимагає знаходження для кожного кластеру множини поверхневих точок застосуванням ітераційної процедури градієнтного спуску, оптимальної за критерієм максимізації поверхневого натягу. Задача в такій узагальненій постановці не розглядається за мету даної магістерської кваліфікаційної роботи. Вона в даній роботі зведена до розробки алгоритму ітераційного пошуку оптимального натягу (стресу) для визначення поверхневих точок тільки для одного з побудованих кластерів. Отримані результати дають можливість розширити роботу алгоритму також на випадок декількох кластерів.

2.3. Розробка алгоритму кластеризації даних

Наведемо словесний опис розробленого в роботі алгоритму. Для заданого масиву даних $MAS(M) = \{\vec{X}_1, \vec{X}_2, \dots, \vec{X}_m, \dots, \vec{X}_M\}$ *N*-мірного простору ознак він описується таким чином.

1) Для всіх різних пар точок обчислюємо масив відстаней за виразом:

$$D(\vec{X}_i, \vec{X}_j) = \sqrt{\sum_{n=1}^{N} (x_{in} - x_{jn})^2}, \qquad (2.22)$$

в якому кількість різних пар відстаней дорівнює $C_M^2 = \frac{M(M-1)}{2}$.

2) Переглядаємо послідовно всі точки масиву $MAS(M)$ шляхом зміни індексу точок m від 1 до M, і виконуємо для кожної з них такі дії:

а) Використовуємо масив відстаней, в ньому вибираємо 5 найближчих точок до поточної k-ої. На вході алгоритму передбачена можливість зміни кількості найближчих точок. Досліджувалися випадки при 3, 4, 5 і 6 найближчих точках, що відповідало разом з взятою k-ою точкою 4, 5, 6 і 7 точкам в ядрі, при цьому оптимальною виявилася кількість з 5 найближчих точок

б) Для вибраної групи точок визначаємо середню відстань за формулою:

$$d_m = \frac{\sum_{p=1}^{5} D(\vec{X}_m, \vec{X}_p)}{5}, \quad (2.23)$$

де p – індекси чотирьох точок, найближчих до k-ої.

в) з урахуванням координат вибраної поточної точки $\vec{X}_m$ обчислюємо координати $\{z_{m1}, z_{m2}, \dots, z_m, \dots, z_M\}$ центра $\vec{Z}_m$ даної сукупності точок

$$z_{m1} = \frac{\sum_{p=1}^{6} x_{p1}}{6}, z_{m2} = \frac{\sum_{p=1}^{6} x_{p2}}{6}, \dots, z_{mn} = \frac{\sum_{p=1}^{6} x_{pn}}{6}, \dots, z_{mN} = \frac{\sum_{p=1}^{6} x_{pN}}{6}, \quad (2.24)$$

де p – номери точок, які належать поточній сукупність точок.

г) Відносне зміщення точки відносно центру визначаємо з виразу:

$$\Delta_m = d(\vec{X}_m, \vec{Z}_m) = \frac{\sqrt{\sum_{n=1}^{N} (x_{mn} - z_{mn})^2}}{d_m}, \quad (2.25)$$

де n – номери координат точок $\vec{X}_m$ і $\vec{Z}_m$, для заданого масиву $MAS(M)$ точок.

д) Перевіряємо умову ; якщо вона виконується, то точка є поверхневою в кластері.

Ітеративно повторюємо процедуру для різних значень поверхневого натягу і порогу σ до тих пір, поки не отримаємо для чергового набору поверхневих точок максимального значення стресу, який використовується за умову припинення пошуку граничних (поверхневих) точок кластера.

Блок-схема алгоритму кластеризації даних запропонованим методом поверхневого натягу зображена на рис. 2.5.

Для реалізації описаного алгоритму розроблено програми мовами Python і Java.

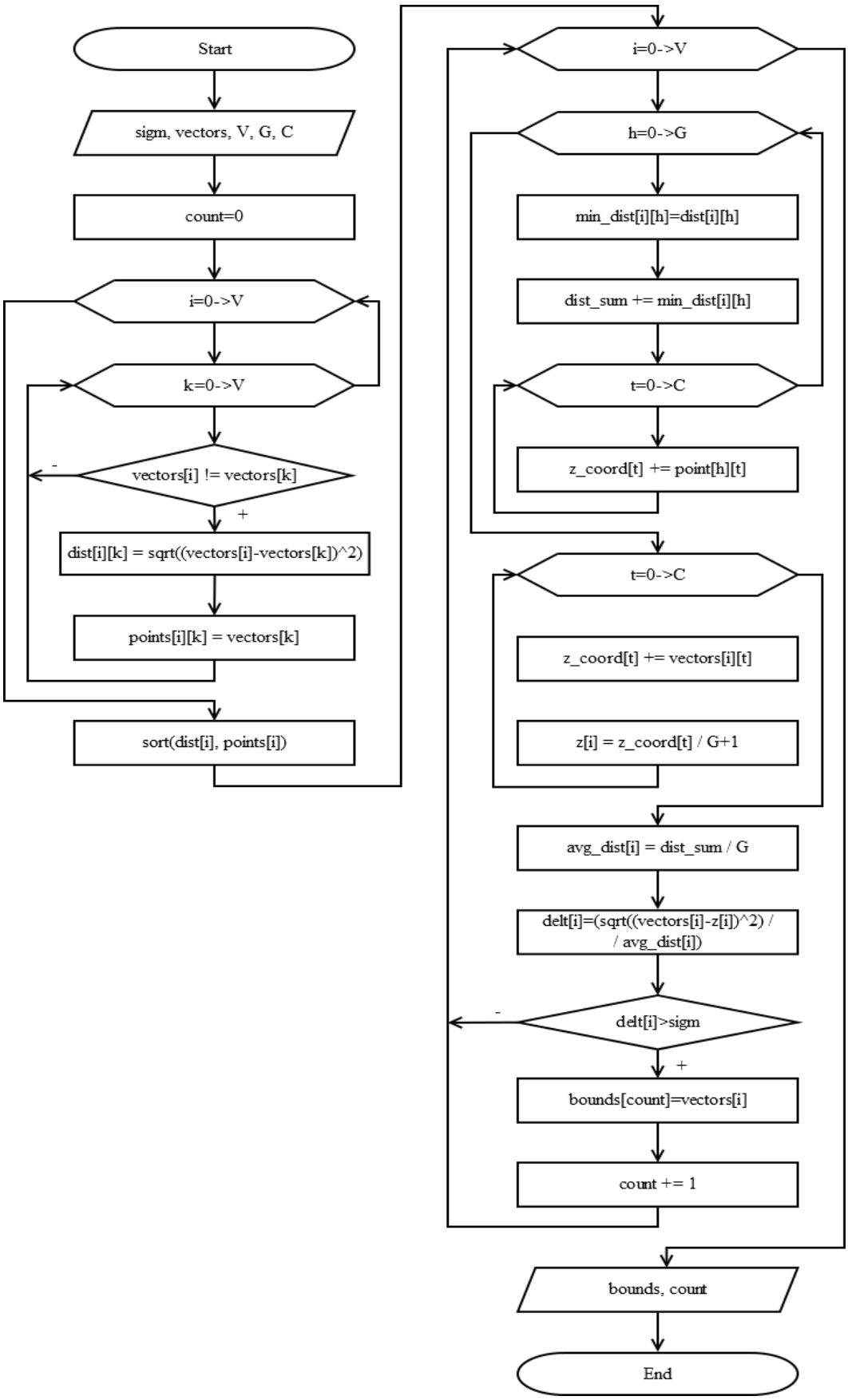


Рисунок 2.5 – Схема алгоритму знаходження поверхневих точок кластера

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1. Обґрунтування вибору програмної платформи

При розробці програмних продуктів ефективність процесу програмування і якість його результатів в великій мірі визначається вибором програмної платформи. Під час вирішення даної проблеми впливають такі основні фактори:

1. Можливість комерційної реалізації розробленого програмного продукту.

2. Наявність в середовищі програмування засобів для реалізації в алгоритмах стандартних підпрограм, що дозволяє розробнику пришвидшити процес програмування, а також тестування адекватності розроблених алгоритмів та якості програм шляхом проведення машинного експерименту.

Для визначення програмних середовищ, які задовольняють вказаним вимогам, проведемо аналіз програмних платформ, які використовуються для реалізації задач машинного навчання.

На сьогоднішній день існує декілька поширених мов програмування і значна кількість бібліотек до них, що використовуються для реалізації задач машинного навчання [41]. До найбільш відомих мов такого застосування можна віднести мови Python та R.

Мова R є досить популярною в середовищі дата-аналітиків. Початково вона розроблювалася для розв’язання задач статистичного аналізу даних, а на разі широко застосовується в областях Big Data і Data Mining. Основним її недоліком є складнощі її застосування для вирішення задач, які не пов'язані з аналізом і візуалізацією даних.

Мова Python, на відміну від мови R, є мовою загального призначення і може бути успішно використана під час вирішення завдань різного характеру, в тому числі і задач машинного навчання. Через це ця мова в останні кілька років набрала великої популярності серед фахівців з ML і стала головною в реалізації задач даного напрямку в IT. Мова Python є дуже

просто і легкою для вивчення. Її застосування в машинному навчанні не вимагає знання всі тонкощів програмування, оскільки підтримується значною кількістю бібліотек, призначених для реалізації з їх допомогою функцій окремих задач і методів машинного навчання, зокрема, регресійного аналізу, зменшення розмірності даних, кластеризації і класифікації об'єктів, графічної візуалізації результатів розрахунків, генерації випадкових величин, тощо [42].

Більшість сучасних бібліотек глибокого (DL) і машинного (ML) навчання надають мові Python додатки API. До найбільш популярних бібліотек відносяться [43, 44]:

- NumPy: задачі обробки багатовимірних масивів і лінійної алгебри; використовується в якості допоміжної бібліотеки під час виконання різних математичних операцій зі структурами даних.
- Scikit-learn: найкраща бібліотека для класичних ML алгоритмів; підтримує більшість алгоритмів навчання, таких як: лінійна і логістична регресія, метод SVM, Байесовський класифікатор, бустінг, кластеризація, метод найближчого сусіда, k-внутрішньо-групових середніх та багато інш.
- Jupyter Notebook: робота з даними, кодом і графіками; дозволяє писати код на Python і інших мовах, додавати текстові описи, вбудовувати графіки та діаграми безпосередньо в інтерактивну веб-сторінку.
- Tensorflow і Keras: бібліотеки глибокого навчання; містить такі компоненти, як багатовимірні масиви (тензори), оператори лінійної алгебри та обчислення похідних.
- Pandas: витяг і підготовка даних; дозволяє завантажувати дані з різних джерел, таких як бази даних SQL, у вигляді CSV, Excel, JSON файлів і інших форматів. Для завантажених в пам'ять даних дозволяє виконувати безліч різних операцій для аналізу, трансформації, заповнення відсутніх значень і очищення набору даних.

З урахуванням вказаних можливостей мови Python її було вибрано для реалізації і тестування запропонованих в роботі ефективних методів

ідентифікації об’єкті з використанням класифікації і кластерного аналізу.

З урахуванням 1-го фактора, з огляду на заплановану комерційну реалізацію розроблених методів і алгоритмів, частина алгоритмів також реалізована мовою програмування Java, яка на сьогодні найчастіше застосовується для розробки комерційних додатків.

3.2. Розробка структури програмного забезпечення

Для вирішення поставлених в індивідуальному завданні на магістерську роботу завдань підвищення ефективності ідентифікації об’єктів шляхом покращення методів кластеризації і класифікації даних, необхідних для підтримки прийняття рішень в інтелектуальних комп’ютерних системах управління, в роботі були розроблені алгоритм кластеризації даних з фіксацією граничних точок кластера, модифікований алгоритм класифікації даних SVM з використанням знайдених алгоритмом кластеризації поверхневих точок, і алгоритм візуалізації отриманих результатів обробки даних у двовимірному просторі шляхом зменшення розмірності описового простору ознак. Вказані алгоритми реалізовані у вигляді одного програмного додатку ClusterBoundsSearcher_v0.0.2. На рис. 3.1 зображена UML-діаграма, в якій показана схема взаємодії програм в додатку.

На рис. 3.1 розроблена в роботі UML-діаграма відображує етапи використання розроблених програми кластеризації, на вхід подається навчальна вибірка заданої розмірності, програми класифікації отриманих даних удосконаленим класифікатором SVM та програми графічної візуалізації результатів шляхом пониження їхньої розмірності для кращого розуміння до двомірного простору (площини).

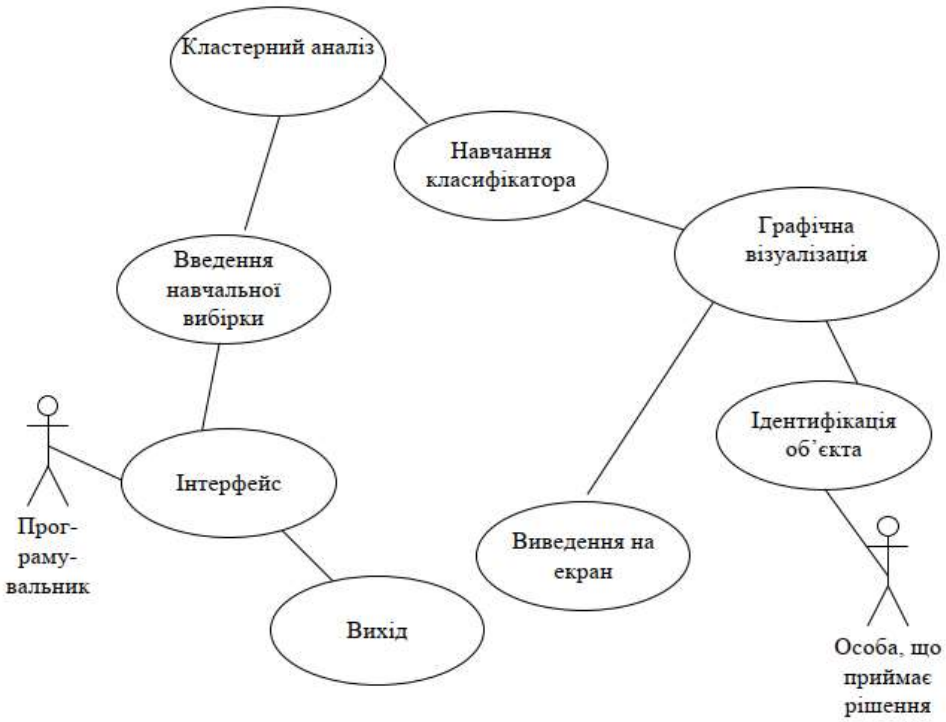


Рисунок 3.1 – UML-діаграма використання програмного забезпечення

Склад програмного додатку ClusterBoundsSearcher_v0.0.2 показано на рис. 3.2.

scripts		<Папка>	29.05.2021 14:47
ClusterBoundsSearcher	jar	26 133	29.05.2021 14:45
Instruction_Cluster_bound_points	pdf	149 472	29.05.2021 15:01
run	bat	94	30.04.2021 12:18
run_additional	bat	87	30.04.2021 12:20

Рисунок 3.2 – Програмний додаток ClusterBoundsSearcher_v0.0.2

В програмний додаток входить папка “scripts” з скриптами модулів мовою Python для кластерного аналізу даних, класифікації даних (ідентифікації) та їх візуалізації. Екран частини тексту програмного модуля ChartBuilder.py для класифікації та кластеризації даних мовою Python показано на рис. 3.3.

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.figure
import numpy as np
import csv
from sklearn.manifold import MDS
from sklearn.decomposition import PCA
from sklearn import svm
import argparse


parser = argparse.ArgumentParser(description='Process data from File')
parser.add_argument('path')
parser.add_argument('svm_gamma')
args = parser.parse_args()
path = args.path
svm_gamma = float(args.svm_gamma)


def transform_to_form(list, dimension:int):
    transformed = [0]*len(list[0])
    for idx in range(0, len(list[0])):
```

Рисунок 3.3 - Скрін екрана з частиною тексту програми мовою Python

Рис. 3.3 показує, що в заголовках (хідерах) програми розміщені 4 оператори імпорту бібліотек Matpotlib, необхідних для виконання алгоритму по створенню графіків та візуалізації даних; бібліотеки Numpy, необхідної для обробки масивів та розв'язання систем лінійних алгебраїчних рівнянь в процедурі знаходження множників Лагранжа під час побудови оптимального класифікатора; функції CSV, що використовується для завантаження введених в графічному інтерфейсі вибіркових даних; бібліотеки Scicit-leran, необхідної для завантаження функцій багатовимірного метричного масштабування MDS, та методу PCA аналізу головних компонент, що реалізує зниження розмірності простору ознак перед візуалізацією даних.

В програмному блоці вставлено модуль ClusterBoundsSearcher.jar мовою Java, призначеного для реалізації розроблених алгоритмів кластеризного аналізу і класифікації, з метою комерціалізації розробленого додатку в майбутньому. Запуск модуля на мові Java здійснюється файлом запуску run_additional.bat, а модуля на мові Java – файлом запуску

run_additional.bat Python.

3.3. Інструкція користувачу програми

Розроблений програмний додаток ClusterBoundsSearcher_v0.0.2 дозволяє реалізувати наступні функції:

- вводити вибірку навчальних даних на кроції навчання та масиви даних, які підлягають класифікації під час етапу експлуатації;

- виконувати кластер ний аналіз даних з фіксацією граничних точок кластера;

- виконувати класифікацію даних за методом SVM з використанням в якості опорних знайдених поверхневих точок знайдених кластерів;

-за необхідності проводитити багатомірне масштабування та здійснювати візуалізацію точок кластерів шляхом проектування на площину.

Результати кластеризації записуються у файли “cluster_bounds_result.txt” та “cluster_bounds.dat”. Перший містить координати усіх точок зі вхідних даних, але вони розбиті на дві групи: точки, які знаходяться в середині кластеру та точки, які формують контур кластера, відповідно є поверхневими точками. Другий містить поверхневі точки кластера у форматі вхідного файлу. Формати даних у файлах описані більш детально нижче в пункті “Формати”.

Графічний інтерфейс програми зображено на рис. 3.4. Він містить п’ять пунктів меню, що відносяться до підготовки даних і параметрів роботи, та три пункти, що відносяться до керування роботою програми:

1 – “Choose a file” - в поле потрібно задати повний шлях до файлу чи натиснути клавішу Open та обрати необхідний файл. Формат запису векторів кластеру в файлі повинен відповідати формату, який наведено нижче в пункті “Формат файлу”.

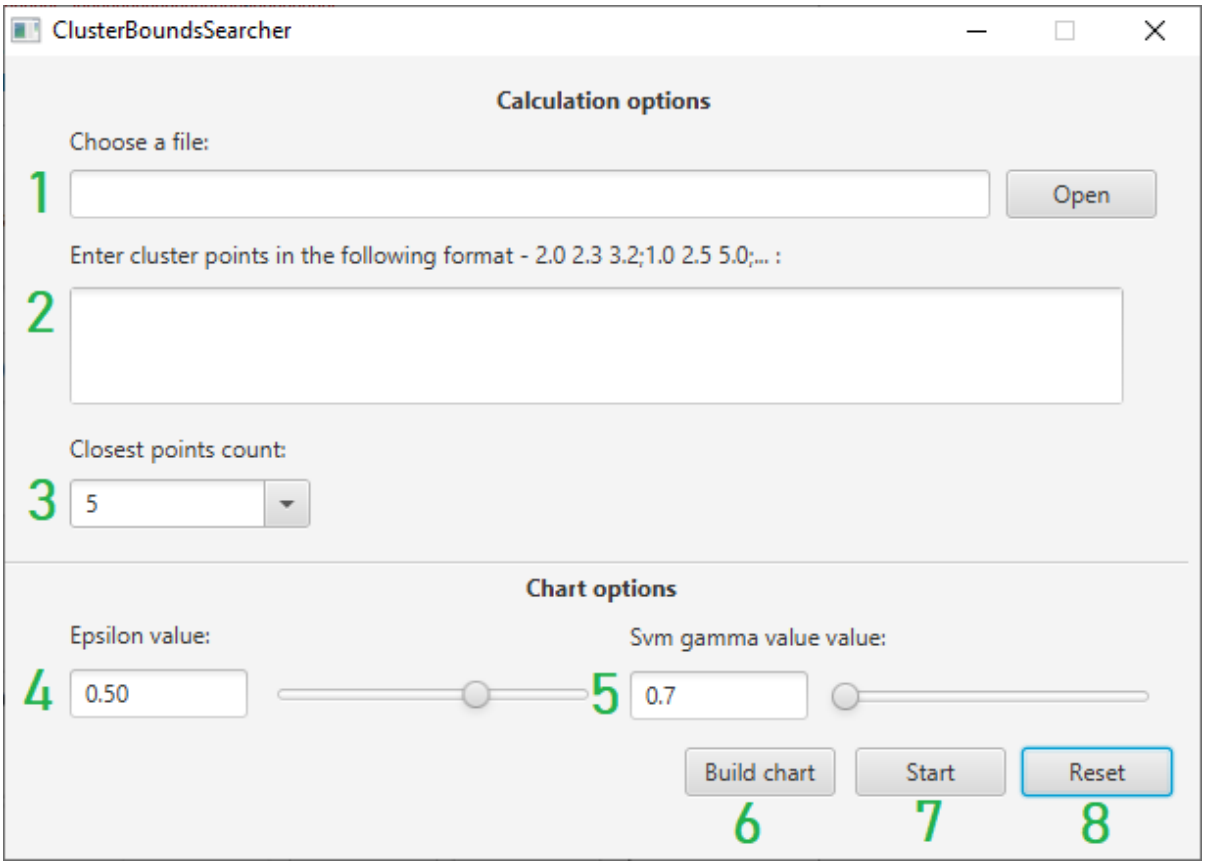


Рисунок 3.4 – Графічний інтерфейс розробленої програми

2 - Введення даних з клавіатури “Enter clusters point in the following format”- в поле потрібно ввести точки кластеру згідно формату, який наведено в нижче в пункті “Формат даних введених з клавіатури”.

3 –Вибір кількостінайближчих точок для розрахунку центрів сукупностей точок під час кластеризації “Closest point count”- можна обрати значення зі випадаючого списку чи виконати введення числового значення з клавіатури.

4 - Порогове значення відхилення “Epsilon value” поверхневого натягу (відносного зміщення точки відносно розрахованого центру) - значення обирається від 0 до 1 зі кроком 0.01.

5 - Введення параметру впливу точок кластера на результати класифікації “SVM gamma value”для методу SVM з типом ядра RBF - значення обирається від 0 до 100 зі кроком 0.05.

Керування:

6 - Клавіша Build chart - запускає процес побудови графіків.

7 - Клавіша Start - запускає процес кластеризації і класифікації та запис результатів у відповідні файли.

8 - Клавіша Reset - повертає усі значення полів графічного інтерфейсу на значення за замовченням.

Примітка: при заповненні обох полів вхідних даних (п. 1 та 2), пріоритет надаватиметься полю з файлом.

Вимоги до форматів даних, з якими працює програма, такі:

1. Формат файлу

Кожен рядок містить координати однієї точки кластеру. Координати розділені символом пробілу (‘ ’) та представляють числове значення зі плаваючою точкою. Приклад формату файлу зі координатами в трьох вимірному просторі:

3.0 5.0 3.
3.0 6.0 3.0
4.0 3.0 4.0
4.0 7.0 4.0

2. Формат даних, введених з клавіатури

Координати точок потрібно задати у вигляді одного рядку, де координати точок розділені символом крапки з комою, координати однієї точки розділені символом пробілу (‘ ’). Значення координати представляє числове значення зі плаваючою точкою. Приклад формату даних введених з клавіатури зі координатами в трьох вимірному просторі:

3.0 5.0 3.0;3.0 6.0 3.0;4.0 3.0 4.0;4.0 7.0 4.0

3. Формат файлу з поверхневими точками (“cluster_bounds.dat”)

Даний файл міститиме масив координат точок, які є поверхневими. Формат відповідає формату вхідного файлу.

3.0 5.0 3.0
3.0 6.0 3.0
4.0 3.0 4.0

4. Формат файлу зі результатом (“cluster_bounds_result.txt”)

Даний файл потрібен для побудови графіків. Перший рядок відповідає значенню розмірності простору координат. Далі в залежності від розмірності простору координат записані рядки, де кожен рядок містить

координату точки. Записані дві групи точок: точки, які знаходяться в середині кластеру та точки, які формують контур кластера, відповідно є поверхневими точками. Приклад формату файлу з результатом з координатами в тривимірному просторі:

3.0
4.0, 5.0, 5.0, 6.0, 7.0
7.0, 4.0, 6.0, 5.0, 6.0
4.0, 5.0, 5.0, 6.0, 7.0
3.0, 4.0, 7.0, 8.0, 3.0, 6.0, 9.0
6.0, 3.0, 3.0, 4.0, 5.0, 7.0, 6.0
3.0, 4.0, 7.0, 8.0, 3.0, 6.0, 9.0

Підготовка програми до запуску:

1. Заповнити один блок зі вхідних даних (п. 1 чи 2 див. Меню ПЗ);

2. Опціонально змінити додаткові параметри (п. 3-4 див. Меню ПЗ);

3. Натиснути клавішу Start для запуску процесу розрахунку відносного зміщення точок відносно відповідних розрахованих центрів.

4. Для побудови графіків необхідно натиснути клавішу Build charts. Дана клавіша стає активною після виконання розрахунку.

Примітка: значення відхилення (епсілон) та коефіцієнта ядра (гамма) можна змінити та повторно натиснути клавішу Build charts, для того щоб дослідити її вплив на кінцеву картину розташування поверхневих точок кластеру. Дана опція не вимагає повного перерахунку вхідних даних на відміну від параметру значення найближчих точок (п. 3 меню).

Результати машинного експерименту

Для оцінки адекватності та роботоздатності програмного забезпечення на етапі кластеризації був проведений експеримент з застосуванням випадкової вибірки навчальних даних, що вводилася з клавіатури. На етапі класифікації з цією метою проводився експеримент з стандартним файлом “iris.dat”. На рис. 3.5 показано результати роботи програми кластеризації навчальної вибірки з двомірних векторів з фіксацією граничних точок кластера.

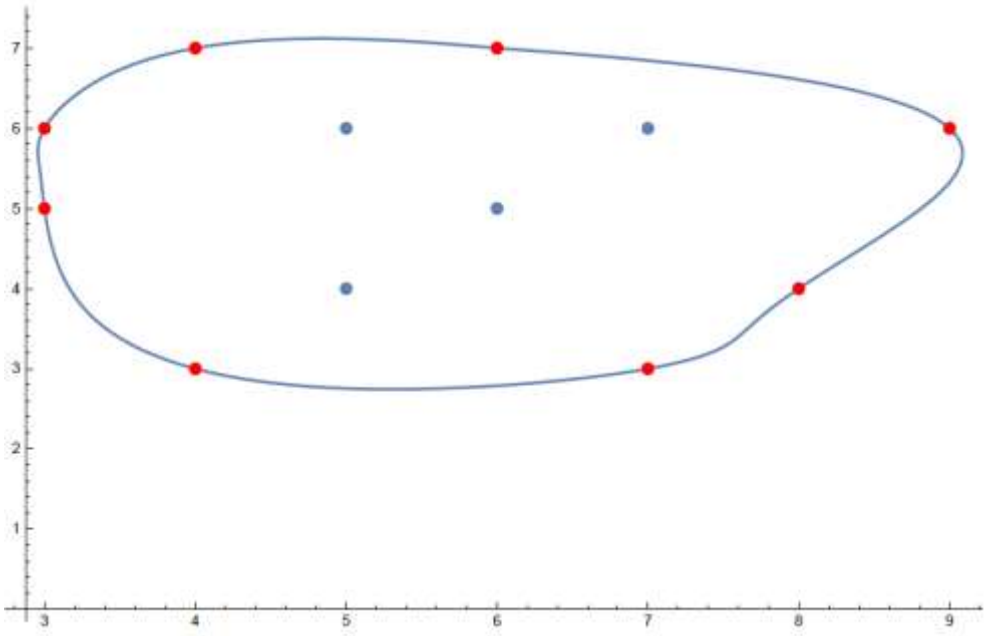


Рисунок 3.5 – Результати кластеризації даних у двомірному просторі ознак при значеннях closest point count=5 і epsilon=0,35

На рис. 3.5 червоними крапками показані поверхневі точки кластера. Проведені на різних навчальних вибірках з використанням даної програми дослідження продемонстрували, що розроблений алгоритм кластеризації правильно фіксує поверхневі точки кластера при заданні порогу натягу epsilon в діапазоні 0,35÷0,45 і кількості точок в ядрі Closest point=5.

Результати класифікації векторів файлу “iris.dat”, спроектованих на двомірну площину, наведені на рис. 3.6.

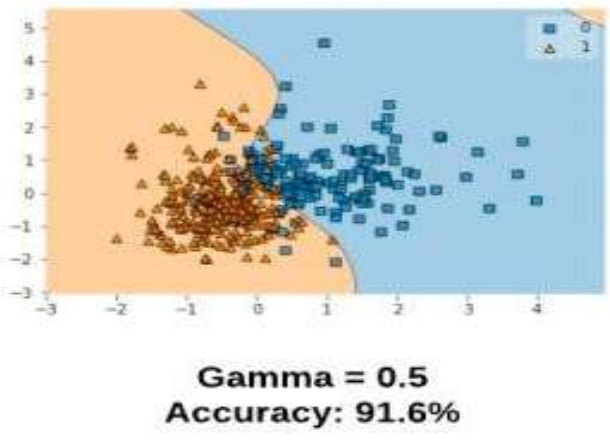


Рисунок 3.6 – Проекція класифікованих 4-х мірних векторів файлу “iris.dat” на площину

Проведений машинний експеримент показав, що класифікатор SVM, модифікований за рахунок використанням поверхневих точок кластерів, отриманих на кроці кластеризації, дає найвищу точність класифікації за умови використаної функціонального ядра RBF та параметра Gamma=0,5. Ця точність співрозмірна з точністю класичного алгоритму SVM, але використання знайдених на етапі кластеризації поверхневих точок процес навчання здійснюється в 3 рази швидше. Такий результат підтверджує ефективність розроблених алгоритмів кластеризації і класифікації об’єктів.

ВИСНОВКИ

В кваліфікаційній роботі м а г і с т р а розроблено покращені методи кластерного аналізу і класифікації даних, що використовуються під час ідентифікації об’єктів для підтримки прийняття рішень в комп’ютерно-інтегрованих системах управління.

Проведено аналіз сучасного стану проблеми ідентифікації об’єктів методами машинного навчання (ML) і їх використання методів ML для підвищення ефективності прийняття рішень в системах управління. В результаті проведеного аналізу обґрунтовано метод ідентифікації з використанням кластеризації і класифікації.

Проведено математичне обґрунтування запропонованого методу кластеризації даних, що використовує фізичний принцип поверхневого натягу, та класифікатора типу SVM, модифікованого на його основі.

Проведена програмна реалізація запропонованих в роботі модифікованих алгоритмів кластерного аналізу і класифікації даних, що використовуються для ідентифікації об’єктів. Розроблено програмний додаток ClusterBoundsSearcher_v0.0.2 у складі: програмного модуля для кластерного аналізу даних з фіксацією граничних точок кластера, модуля модифікованого класифікатора SVM, модуля пониження розмірності простору ознак до двомірного з метою візуалізації отриманих результатів кластеризації і класифікації. Проведений машинний експеримент підтвердив адекватність розроблених методів і програмного забезпечення.

Розроблені методи і програмне забезпечення для ідентифікації об’єктів в системах управління дозволили підвищити такі показники ефективності прийняття рішень, як швидкість і якість.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. https://uk.wikipedia.org/wiki/Ідентифікація

2. Березін Б. О. Підтримка прийняття рішень при побудові систем довготермінового зберігання інформаці / Б.О. Березін, Качанов П. Т., В.В. Циганок, О. В.Андрійчук // Проблеми розвитку інформаційного суспільства: матеріали Міжнародного форуму. -2009.- С.145-153.

3. Rao C., Govindaraju V. Handbook of Statistics: Machine Learning: Theory and Applications / C. Rao, V. Govindaraju. – Cambridge University Press, - 2013. - 552 c.

4. Nils J. Nilson. Introduction to Mashine Learning / - [Електронний ресурс]. – Режим доступу: http://ai.stanford.edu/people/nilsson/mlbook.html.

5. Looney C.G. Pattern Recognition Using Neural Networks / C. G. Looney –New York: Oxford University Press, 2004. – 449 p.

6. MacKay D. Informatiom Theory, Inference and Learning Algorithms / D. MacKay. – Cambridge University Press, 2003. – 640 p.

7. Глибовець М.М., Олецький О.В. Системи штучного інтелекту. - К.: КМ Академія, 2002. - 366 с.

8. Smith L. An Introduction to Neural Networks / Leslie Smith – University of Stirling, 2008. – [Електронний ресурс]. - Режим доступу: http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html

9. Mitchell T.M. Mashine Learning / T.M. Mitchell. – McGraw-Hill Science, 2008. – 414 p.

10. Tom M. Mitchell. Mashine Learning.- McGraw-Hill: Science/ Engineering/ Math, 1997. – p. 432.

11. N.M. Bykov, I.V. Kuzmin, A.I. Yakovenko. Development of effective strategy of pattern recognition. - Proceedings of SPIE, 2001, Vol. 4225, pp.76- 83.

12. Steinwart I., Christmann A. Support Vector Machines / Ingo Steinwart, Andreas Christmann. - Springer-Verlag, New York, 2008. - 602 p.

13. Vapnik V., Chapelle O. Bounds on error expectation for support vector machines // Neural Computation. - 2000. - Vol. 12, no. 9. - Pp. 2013–2036.

14. Steinwart I., Christmann A. Support Vector Machines / Ingo

Steinwart, Andreas Christmann. - Springer-Verlag, New York, 2008. - 602 p.

15. Ray S. Understanding Support Vector Machine algorithm from examples / Sunil Ray. – [Електронний ресурс]. – Режим доступу: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/.

16. [Електронний ресурс].– Режим доступу: https://habr.com/ru/post/428503/

17. [Електронний ресурс]. – Режим доступу: http://datascientist.one/support-vector-machines/

18. Fine S., Scheinberg K. INCAS: An incremental active set method for SVM: Tech. rep.: 2002. http://citeseer.ist.psu.edu/fine02incas.html.

19. Биков М.М., Волоський Б.О. Розробка ефективного класифікатора даних в інтелектуальних системах управління [Електронний ресурс] / М.М. Биков, Б.О. Волоський // Матеріали XLIX науково-технічної конференції підрозділів ВНТУ, Вінниця, 27-28 квітня 2020 р. – 2020. – Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2020/paper/view/9730

20. Биков М.М., Задачін А.С. Розробка програмного забезпечення кластеризації даних в інтелектуальних системах прийняття рішень. - Електронне наукове видання матеріалів конференції, м. Вінниця, 2021. – Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2021/paper/view/12437.

21. Rao C., Govindaraju V. Handbook of Statistics: Machine Learning: Theory and Applications, 2013, 552 c.

22. Bishop C.M. Pattern Recognition and Machine Learning. Springer, 2006, pp. 213-232.

23. Duda, R. O., Hart, P. E., & Stork, D. G. Pattern Classification. Wiley-Interscience, 2000, pp. 354-367.

24. Mitchell, T. M. Machine Learning. McGraw-Hill, 1997, pp. 157-180.

25. Hastie, T., Tibshirani, R., & Friedman, J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 2009, pp. 431-460.

26. Ripley, B. D. Pattern Recognition and Neural Networks. Cambridge University Press, 1996, pp. 125-143.

27. Goodfellow, I., Bengio, Y., & Courville, A. Deep Learning. MIT Press, 2016, pp. 457-479.

28. Breiman, L. Random Forests. Machine Learning, 45(1), 2001, pp. 5-32.

29. Russell, S., & Norvig, P. Artificial Intelligence: A Modern Approach. Prentice Hall, 2010, pp. 567-589.

30. Fawcett, T. An Introduction to ROC Analysis. Pattern Recognition Letters, 27(8), 2006, pp. 861-874.

31. Cover, T. M., & Thomas, J. A. Elements of Information Theory. Wiley-Interscience, 2006, pp. 221-239.

32. Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993, pp. 102-119.

33. Vapnik, V. N. Statistical Learning Theory. Wiley-Interscience, 1998, pp. 299-315.

34. Kohavi, R., & Provost, F. Glossary of Terms. Machine Learning, 30(2-3), 1998, pp. 271-274.

35. Aggarwal, C. C. Data Mining: The Textbook. Springer, 2015, pp. 423-439.

36. Hand, D. J., & Yu, K. Idiot's Bayes: Not So Stupid After All? International Statistical Review, 69(3), 2001, pp. 385-398.

37. Hastie, T., & Tibshirani, R. Discriminant Adaptive Nearest Neighbor Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(6), 1986, pp. 607-616.

38. Freund, Y., & Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences, 55(1), 1997, pp. 119-139.

39. Pearl, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, 1988, pp. 189-213.

40. James, G., Witten, D., Hastie, T., & Tibshirani, R. An Introduction to Statistical Learning: with Applications in R. Springer, 2013, pp. 245-258.

41. Cover, T., & Hart, P. Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory, 13(1), 1967, pp. 21-27.

42. Bishop, C. M. Neural Networks for Pattern Recognition. Oxford University Press, 1995, pp. 198-215.

43. Langley, P. Elements of Machine Learning. Morgan Kaufmann, 1996, pp. 131-146.

44. Koller, D., & Friedman, N. Probabilistic Graphical Models: Principles and Techniques. MIT Press, 2009, pp. 492-515.

45. Schölkopf, B., & Smola, A. J. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, 2002, pp. 315-335.

46. Zhang, T. Statistical Behavior and Consistency of Classification Methods Based on Convex Risk Minimization. Annals of Statistics, 32(1), 2004, pp. 56-85.

47. Sutton, R. S., & Barto, A. G. Reinforcement Learning: An Introduction. MIT Press, 2018, pp. 371-393.

48. Hinton, G. E., & Salakhutdinov, R. R. Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786), 2006, pp. 504-507.

49. Witten, I. H., Frank, E., & Hall, M. A. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2011, pp. 423-437.

50. Murphy, K. P. Machine Learning: A Probabilistic Perspective. MIT Press, 2012, pp. 567-592.

51. Raudys, S. Statistical and Neural Classifiers: An Integrated Approach to Design. Springer, 2001, pp. 157-176.

52. Hastie, T., Tibshirani, R., & Buja, A. Flexible Discriminant Analysis by Optimal Scoring. Journal of the American Statistical Association, 89(428), 1994, pp. 1255-1270.

53. Shi, X. Object Recognition and Clustering Using Neural Networks. IEEE Transactions on Neural Networks, 16(5), 2005, pp. 1213-1233.

54. Kohonen, T. Self-Organizing Maps. Springer, 2001, pp. 357-374.

55. Yang, C. Y., & Yang, X. Object Identification in Intelligent Systems.

IEEE Transactions on Systems, Man, and Cybernetics, Part A, 38(2), 2008, pp. 297-310.

56. Ripley, B. D. Pattern Recognition and Neural Networks. Cambridge University Press, 1996, pp. 198-220.

57. Manning, C. D., & Schütze, H. Foundations of Statistical Natural Language Processing. MIT Press, 1999, pp. 421-443.

58. Amari, S.-I., & Wu, S. Improving Support Vector Machine Classifiers by Modifying Kernel Functions. Neural Networks, 12(6), 1999, pp. 783-789.

59. Zhang, Z., & Xie, Y. Object Detection in Decision-Making Systems. IEEE Transactions on Neural Networks, 20(3), 2009, pp. 451-467.

60. Liu, B. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer, 2006, pp. 297-318.

61. LeCun, Y., Bengio, Y., & Hinton, G. Deep Learning. Nature, 521(7553), 2015, pp. 436-444.