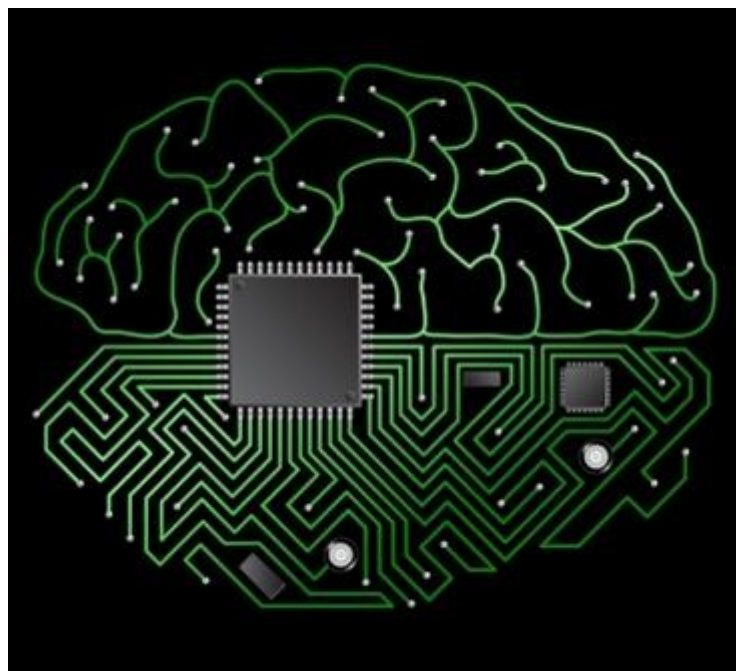


М. М. Биков, В. В. Ковтун, В. О. Гаврилюк

ОСНОВИ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ

Частина 1. Технології розпізнавання



Міністерство освіти і науки України
Вінницький національний технічний університет

М. М. Биков, В. В. Ковтун, В. О. Гаврилюк

ОСНОВИ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ

Частина 1. Технології розпізнавання

Електронний навчальний посібник
комбінованого (локального та мережного) використання

Вінниця
ВНТУ
2023

УДК 681.3.07

Б60

Рекомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 12 від 04.05.2023 р.)

Рецензенти:

С. Д. Штовба, доктор технічних наук, професор

О. В. Бісікало, доктор технічних наук, професор

О. М. Козачко, кандидат технічних наук, доцент

Биков, М. М.

Б60 Основи інтелектуальних технологій. Частина 1. Технології розпізнавання : електронний навчальний посібник комбінованого (локального та мережного) використання [Електронний ресурс] / Биков М. М., Ковтун В. В., Гаврилюк В. О. – Вінниця : ВНТУ, 2023. – 229 с.

Навчальний посібник складається з Частини 1 – Технології розпізнавання і Частини 2 – Технології машинного навчання.

У Частині 1 посібника наводяться теоретичні відомості з важливого для спеціальності 174 напрямку штучного інтелекту – розпізнавання образів. Теоретичний матеріал супроводжується прикладами задач і їх розв’язанням, а також алгоритмами вирішення окремих задач з інтелектуальних технологій розпізнавання.

У Частині 2 наводяться математичні і алгоритмічні основи розробки технологій навчання традиційним програмуванням і з застосуванням машинних методів.

УДК 681.3.07

© ВНТУ, 2023

ЗМІСТ

| | |
|---|-----------|
| ВСТУП..... | 6 |
| РОЗДІЛ 1 ОСНОВНІ НАПРЯМИ І ЕТАПИ ДОСЛІДЖЕНЬ В ГАЛУЗІ ШТУЧНОГО ІНТЕЛЕКТУ | 8 |
| 1.1 Напрями застосування систем штучного інтелекту | 8 |
| 1.2 Коротка історія розвитку систем штучного інтелекту і їх використання | 9 |
| 1.3 Структура сучасних систем штучного інтелекту | 10 |
| 1.4 Розв'язування задач | 12 |
| 1.5 Застосування систем ШІ в іграх..... | 13 |
| 1.6 Експертні системи..... | 15 |
| 1.7 Робототехніка | 19 |
| 1.8 Машинний переклад мови..... | 22 |
| 1.9 Інтелектуальний аналіз даних (Data Mining)..... | 24 |
| 1.10 Нейронні мережі..... | 25 |
| 1.11 Прийняття рішень в умовах невизначеності і браку інформації .. | 28 |
| 1.12 Розпізнавання образів | 32 |
| Контрольні питання та завдання | 34 |
| РОЗДІЛ 2 ЗАГАЛЬНІ ПОНЯТТЯ ТЕОРІЇ РОЗПІЗНАВАННЯ ОБРАЗІВ | 36 |
| 2.1 Визначення терміну «розпізнавання образів»..... | 36 |
| 2.2 Основні поняття теорії розпізнавання образів | 38 |
| 2.3 Основні задачі теорії розпізнавання образів | 42 |
| 2.4 Класифікація систем розпізнавання образів..... | 54 |
| Контрольні питання та завдання | 62 |
| РОЗДІЛ 3 КЛАСИФІКАЦІЯ ОБРАЗІВ | 63 |
| 3.1 Методи класифікації | 63 |
| 3.2 Лінійні вирішувальні функції та їх властивості..... | 66 |
| 3.3 Узагальнені вирішувальні функції | 74 |
| 3.4 Класифікація образів за критерієм мінімуму відстані | 78 |
| 3.5 Класифікація образів за методом опорних векторів SVM..... | 80 |
| Контрольні питання та завдання | 95 |
| РОЗДІЛ 4 ІМОВІРНІСНИЙ ПІДХІД ДО РОЗПІЗНАВАННЯ ОБРАЗІВ..... | 99 |
| 4.1 Класифікація образів як задача теорії статистичних рішень..... | 99 |
| 4.2 Байєсівська класифікація образів | 101 |
| 4.3 Байєсівський класифікатор для образів з нормальною щільністю розподілу ймовірностей ознак | 107 |
| 4.4 Оцінка функцій щільності розподілу і їх параметрів..... | 110 |
| 4.5 Оцінка помилок байєсівського класифікатора..... | 121 |

| | | |
|---|---|------------|
| 4.6 | Апроксимація щільностей розподілу функціями | 130 |
| 4.7 | Алгоритм побудови Байєсівського класифікатора для образів з невідомим законом щільності розподілу ймовірностей ознак | 135 |
| | Контрольні питання та завдання | 139 |
| РОЗДІЛ 5 КЛАСТЕРНИЙ АНАЛІЗ ДАНИХ..... | | 141 |
| 5.1 | Міри подібності | 141 |
| 5.2 | Загальні принципи і критерії кластеризації..... | 145 |
| 5.3 | Алгоритм кластеризації за пороговою відстанню | 146 |
| 5.4 | Алгоритм максимінної відстані (алгоритм кластеризації Бетчелора-Уілкінса) | 150 |
| 5.5 | Алгоритм К-внутрішньогрупових середніх | 156 |
| 5.6 | Алгоритм ІСОМАД (ISODATA) | 160 |
| 5.7 | Алгоритм динамічних ядер | 170 |
| 5.8 | Графо-теоретичні алгоритми кластеризації | 174 |
| | Контрольні питання та завдання | 178 |
| РОЗДІЛ 6 ВИБІР МНОЖИНИ ОЗНАК В СИСТЕМАХ РОЗПІЗНАВАННЯ..... | | 179 |
| 6.1 | Характеристика проблеми вибору множини ознак образів..... | 179 |
| 6.2 | Використання відстаней в задачі виділення ознак | 180 |
| 6.3 | Зменшення розмірності простору ознак методом РСА..... | 182 |
| 6.4 | Вплив лінійного перетворення простору ознак на визначення їх важливості | 190 |
| 6.5 | Вибір ознак за принципом мінімізації ентропії | 192 |
| 6.6 | Вибір ознак за методом розкладу Карунена-Лоева | 197 |
| 6.7 | Розклад Карунена-Лоева для випадкових процесів..... | 203 |
| | Контрольні запитання та завдання | 206 |
| РОЗДІЛ 7 СИНТАКСИЧНЕ РОЗПІЗНАВАННЯ ОБРАЗІВ..... | | 208 |
| 7.1 | Основні ідеї синтаксичного підходу до розпізнавання | 208 |
| 7.2 | Структурна схема системи синтаксичного розпізнавання образів... | 212 |
| 7.3 | Поняття теорії формальних мов | 214 |
| 7.4 | Структурна схема системи синтаксичного розпізнавання образів... | 220 |
| 7.5 | Застосування граматики в розпізнаванні образів | 222 |
| | Контрольні запитання та завдання | 227 |
| ЛІТЕРАТУРА..... | | 228 |

ВСТУП

Автоматизація процесів обробки інформації на основі використання обчислювальної техніки в системах управління і автоматики поставила на порядок денний питання розв'язання найскладніших задач з автоматизації операцій, що відносяться до складних інтелектуальних форм діяльності людини – розпізнавання образів, вирішення задач, формування складних логічних понять та доведення теорем, переклад з однієї мови на іншу, прийняття рішень в умовах невизначеності та нечіткого опису об'єктів управління та інших. Математичні методи та методи програмування, призначені для вирішення таких задач, є змістом галузі науки під назвою **«Штучний інтелект»**.

Задачі штучного інтелекту вирішуються за допомогою інформаційних технологій штучного інтелекту. Під інформаційною технологією в загальному випадку розуміють процес, що використовує сукупність засобів і методів збирання, опрацювання, зберігання і передавання даних (первинної інформації) для отримання інформації нової якості про стан об'єкта, процесу чи явища (інформаційного продукту). Мета інформаційної технології — виробництво інформації для її аналізу людиною і прийняття на його основі рішення щодо виконання якої-небудь дії. Зокрема, інформаційна технологія штучного інтелекту, або інтелектуальна технологія (ІТ), являє собою набір програмних засобів, що базуються на математичних методах подання і опрацювання інформації, які описують механізми розумової діяльності людей та інших біологічних організмів, й на методах навчання, які відтворюють роботу нейронних мереж.

Метою вивчення дисципліни «Основи інтелектуальних технологій» (ОІТ) є формування у студентів цілісного уявлення про суть інтелектуальних технологій та їх місце в автоматизації та робототехніці.

Згідно з вимогами освітньо-професійної програми студенти мають:

вміти:

- обґрунтувати доцільність застосування інтелектуальних технологій в прикладних комп'ютерних системах управління;
- обрати доцільний тип інтелектуальної технології для конкретної комп'ютерної системи управління;
- спроектувати інтелектуальну технологію для конкретної комп'ютерної системи управління із застосуванням сучасного програмного інструментарію;
- експлуатувати системи автоматизації та робототехніку, що використовують в функціонуванні інтелектуальні технології.
- здійснювати вибір програмних засобів для вирішення задач штучного інтелекту; вирішувати задачі автоматизації підтримки прийняття рішень, розпізнавання образів, діагностики, класифікації та аналізу даних.

знати:

- класифікацію задач та видів систем штучного інтелекту;
- суть і класифікацію інтелектуальних технологій;
- принципи розробки інтелектуальних технологій;
- місце і роль інтелектуальних технологій в системах автоматизованого управління та в робототехнічних системах;
- принципи розробки інтелектуальних технологій;
- основні означення та методи теорії розпізнавання образів;
- принципи створення інтелектуальних систем на основі розпізнавання образів;
- застосування дисперсійного аналізу та теорії кореляції під час вирішення задач інтелектуального аналізу даних та розпізнавання образів;
- теоретичні основи інтелектуальних технологій на основі дерев рішень та інформаційні технології їх проектування;
- інтелектуальні технології пошуку з використанням дерев;
- теоретичні основи інтелектуальних технологій на основі нечітких баз знань та інформаційні технології їх проектування;
- теоретичні основи інтелектуальних технологій на основі штучних нейронних мереж та інформаційні технології їх проектування.

Компетентності, на оволодіння якими спрямоване вивчення дисципліни:

- Здатність до пошуку, опрацювання та аналізу інформації з різних джерел.
- Здатність розробляти та досліджувати інтелектуальні системи управління і обробки даних.
- Здатність застосовувати спеціальні знання для створення ефективних систем автоматизації складних технологічних об'єктів та комплексів на основі інтелектуальних методів управління й комп'ютерних технологій з використанням баз даних, баз знань та методів штучного інтелекту.
- Здатність до ідентифікації, збереження та аналізу великих даних у рамках інформаційних систем.

Навчальний посібник складається з двох частин:

Частина I – Технології розпізнавання,

Частина II – Технології машинного навчання.

В цій частині посібника, яка складається з 7-и розділів, розглядаються основні напрями і етапи досліджень в галузі штучного інтелекту, загальні питання теорії розпізнавання образів, методи класифікації образів і кластерного аналізу даних, принципи вибору інформативних ознак для опису образів та технології розпізнавання, що використовують синтаксичний підхід.

РОЗДІЛ 1

ОСНОВНІ НАПРЯМИ І ЕТАПИ ДОСЛІДЖЕНЬ В ГАЛУЗІ ШТУЧНОГО ІНТЕЛЕКТУ

1.1 Напрями застосування систем штучного інтелекту

Штучним інтелектом називається галузь науки і техніки, яка вивчає теоретичні основи та методи практичної реалізації інтелектуальних систем (ІС). Під *інтелектуальною* розуміють таку систему, яка під час обробки інформації та прийняття рішень для виконання дій, спрямованих на досягнення поставленої мети, наслідує механізми людського мислення. В більш широкому розумінні інтелектуальною називають таку систему, яка здатна адаптувати своє функціонування до зміни зовнішніх умов. Основна частина роботи в ІС виконується комп'ютерною програмою, через що цю програму часто називають ІС, хоча це і не зовсім відповідає дійсності. До основних галузей застосування систем штучного інтелекту можна віднести:

- Розв'язування задач і доведення теорем;
- Ігри;
- Експертні системи (інженерія знань);
- Робототехніка;
- Машинний переклад мови;
- Інтелектуальний аналіз неопрацьованих даних (Data Mining);
- Нейронні мережі;
- Прийняття рішень в умовах невизначеності та браку інформації;
- Розпізнавання образів.

Сучасний підхід **визначає штучний інтелект як науку про агентів**, що сприймають впливи зовнішнього середовища, в якому вони функціонують, й виконують дії, направлені на досягнення поставленої мети, причому кожен такий агент реалізує функцію, що відображає послідовності актів сприйняття в дії. **Агентом** вважається все, що діє (слово агент пішло від латинського слова *agere* — діяти). Але передбачається, що комп'ютерні агенти мають деякі інші атрибути, які відрізняють їх від звичайних «програм», таких як здатність функціонувати автономно, сприймати своє середовище, існувати протягом тривалого періоду часу, адаптуватися до змін і мати здатність узяти на себе досягнення мети, поставленої іншими. **Раціональним агентом** називається агент, який діє так, щоб можна було досягти якнайкращого результату або, в умовах невизначеності, якнайкращого очікуваного результату.

Існують різні способи подання цих функцій, зокрема продукційні системи, реактивні агенти, умовні планувальники в реальному масштабі часу, нейронні мережі і системи, що діють на основі теорії рішень.

Агентний підхід дає можливість застосувати для проектування ІС всі напрацьовані раніше методи оброблення інформації, призначені для прийняття ефективних рішень. В такому аспекті робототехніка та розпізнавання образів розглядаються не як незалежно визначені наукові напрямки, а як галузі знань, що дозволяють забезпечити більш успішне досягнення цілей, що стоять перед агентами, і підкреслюють важливість врахування того, в якому середовищі агент вирішує поставлені перед ним завдання.

1.2 Коротка історія розвитку систем штучного інтелекту і їх використання у вказаних напрямках застосування

На *першому етапі* (1956–1973) розвитку методів штучного інтелекту дослідницьким полігоном стали всілякі ігри, головоломки, математичні задачі. Деякі з цих задач стали класичними в літературі по ШІ (задачі про мавпу і банани, місіонерів і людоджерів, Ханойські вежі, гра в 15 та інші). Вибір таких задач обумовлювався простотою і ясністю проблемного середовища (середовища, у якому розгортається вирішення задачі), її відносно малою громіздкістю, можливістю достатньо легкого відбору і навіть штучного конструювання «під метод». Основний розквіт такого роду досліджень припадає на кінець 60-х років, після цього стали з'являтися перші спроби застосування розроблених методів до задач, вирішуваних не в штучних, а в реальних проблемних середовищах.

Другий етап (1974–1980) досліджень з ШІ був викликаний необхідністю дослідження інтелектуальних систем за умови їхнього функціонування в реальному середовищі з метою створення інтегральних роботів. У Стенфордському університеті, Стенфордському дослідницькому інституті і деяких інших закладах були розроблені експериментальні роботи, що функціонують у лабораторних умовах. Проведення цих експериментів показало необхідність вирішення кардинальних питань, пов'язаних із проблемою подання знань про середовище функціонування, і одночасно недостатню дослідженість таких проблем, як зорове сприйняття, побудова складних планів поведінки в динамічних середовищах, спілкування з роботами природною мовою.

Ці проблеми були більш ясно сформульовані і поставлені перед дослідниками в середині 70-х рр. Цей період вважають початком *третього етапу* (1980–1993) досліджень систем ШІ. Його характерною рисою стало перенесення центра уваги дослідників зі створення автономно функціонуючих систем, що самостійно вирішують у реальному середовищі поставлені перед ними задачі, на створення людино-машинних систем, що інтегрують у єдине ціле інтелект людини і можливості обчислювальної машини для досягнення загальної мети – розв'язання задачі, поставленої перед інтегральною людино-машинною системою прийняття рішень. Можна зауважити дві причини такої зміни акцентів:

- до цього часу з'ясувалося, що навіть прості на перший погляд задачі, що виникають перед інтегральним роботом під час його функціонування в реальному часі, не можуть бути вирішені методами, розробленими для експериментальних задач в спеціально сформованих проблемних середовищах.

- стало зрозумілим, що поєднання доповнюючих одна одну можливостей людини й ЕОМ дозволяє обійти гострі кути шляхом перекладання на людину тих функцій, які поки що є недоступними для ЕОМ. На перший план стали висувати не розробку окремих методів машинного розв'язання задач, а розробку методів і засобів, що забезпечують тісну взаємодію людини й обчислювальної системи протягом усього процесу розв'язування задачі з можливістю оперативного внесення людиною змін у ході цього процесу.

Розвиток досліджень з ШІ в цьому напрямку обумовлювався також різким зростанням виробництва засобів обчислювальної техніки і також різким їхнім здешевленням, що зробило їх потенційно доступними для більш широкого кола користувачів. Це привело до виникнення в кінці 20 - на початку 21 століття нового, агентного, підходу до побудови інтелектуальних систем, який знаменує початок *четвертого етапу* (1994-донині) в розвитку штучного інтелекту.

1.3 Структура сучасних системи штучного інтелекту

Для розуміння структури системи ШІ розглянемо характерні риси людського інтелекту, які наслідуються комп'ютерними програмами таких систем.

Першою істотною рисою таких програм є відтворення ними послідовності процесів мислення людини і її діяльності, а саме:

- постановка мети;
- визначення об'єктів навколишнього середовища, взаємодія з якими дозволяє досягнути поставленої мети;
- визначення властивостей об'єктів, що підлягають зміні, і відповідних правил для їх зміни;
- застосування правил з метою досягнення поставленої цілі.

Метою називається кінцевий результат, на який спрямовано процеси мислення і діяльності людини. Як тільки мета досягнута, перед людським мозком одразу виникають нові цілі.

Приклад. Ранком дзвонить будильник, який сповіщає про те, що час підніматися і збиратися на роботу. Перша мета, яка виникає в мозку людини – виключити його, і мозок віддає команду руці на виконання цієї дії.

Як тільки це буде виконано, мозку людини одразу постають нові цілі – піти в ванну кімнату, почистити зуби, вмитися, одягнутися, поснідати, вийти на автобусну зупинку, сісти на потрібний автобус і т. д. Здійснення

всіх цих цілей веде до досягнення головної мети – вчасно потрапити на роботу. Кожен крок на шляху до головної мети має свою локальну ціль. Ціль примушує людину мислити. Таким чином, людина робить щось не тому, що думає, а думає тому, що має щось робити. Тому під час проектування ІС завжди потрібно орієнтуватися на мету, для досягнення якої вони призначені. Крім того, в них потрібно відтворювати механізми вирішення людиною повсякденних задач. Для цього штучний інтелект має являти собою сховище відомостей про об'єкти, описані у вигляді сукупності їх властивостей, і сукупностей правил використання цих властивостей об'єктів для досягнення поставленої мети.

Сукупність цих трьох понять (об'єкт, властивості об'єкта – параметри, факти, правила використання і зміни властивостей) описує **фрагмент (фрейм) знань**. Набір фреймів знань в пам'яті штучного інтелекту утворює базу знань. Чим більша база знань, тим більша кількість цілей, які може ставити і досягати інтелектуальна система.

Приклад 1.1

Розглянемо такі фрейми:

1. а) Об'єкт – кухонна плита;
б) Факт, властивість – запалена плита гаряча;
в) *Правило 1.* Якщо покласти руку на плиту, то можна обпектися.
2. а) Об'єкт – вулиця;
б) властивість, факт – в годину «пік» на вулиці багато машин;
в) *Правило 2.* Якщо намагатися перейти вулицю в годину «пік», то можна потрапити під машину.

В наведених прикладах всі правила відображені відношенням слідування ЯКЩО – ТО. Це означає, що ЯКЩО виконується деяка умова, ТО з неї виводиться певна дія чи якась інша реакція. Ці правила також часто називають правилами виведення, тому що з їх допомогою можна виводити (усвідомлювати) нові факти. Частина інтелекту, яка допомагає з'ясувати нові факти, називається механізмом виведення.

Приклад 1.2

Відомі такі факти.

Факт 1. Джейн і Пітер – батьки Карла.

Факт 2. Джейн і Пітер – батьки Мері.

Ставимо за мету визначити, ким є один для одного Карл і Мері. Для її досягнення механізм пошуку знаходить в пам'яті фрагмент знань про Карла і Мері та звертається до правила – ЯКЩО у хлопчика і дівчинки одні і ті самі батьки, ТО вони – брат і сестра. Мета досягнута. В процесі її досягнення було отримано раніше невідомий факт – Карл і Мері є братом та сестрою.

Механізм виведення дозволяє людині та інтелектуальній системі вчитися на досвіді, тобто генерувати нові факти на основі уже існуючих знань.

Компоненти інтелектуальної системи, що відтворюють описані процеси, подано на рис. 1.1.

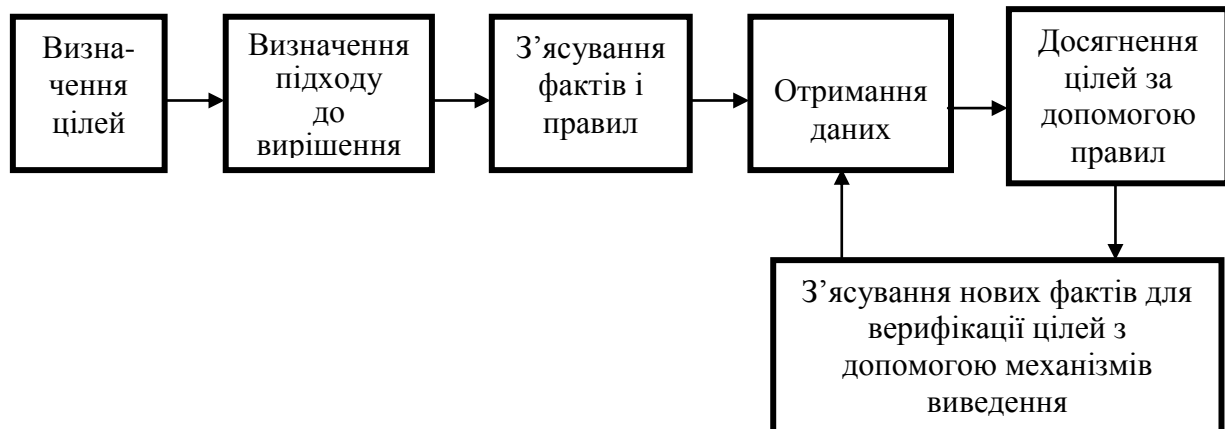


Рисунок 1.1 – Основні компоненти інтелектуальної системи

Другою істотною рисою програм, що реалізують штучний інтелект, є їх модульна структура. Вона зумовлена фреймовою структурою бази знань і дозволяє, аналогічно людському мозку, поповнювати базу знань новими фреймами, не змінюючи уже існуючих. Різні компоненти структури ІС визначаються окремо і модифікація будь-якої її частини не чіпає загальної структури. Аналіз показує, що перші комп'ютерні програми, які вирішували інтелектуальні задачі, мали модульну структуру, де кожен модуль призначений для роботи зі своїм фреймом знань, а ініціалізація потрібного модуля здійснювалась з супервізорного модуля. Така концепція сьогодні називається об'єктно-орієнтованим програмуванням і її на сьогодні також використовують більшість програм, які не відносяться до ІС.

Охарактеризуємо коротко основні галузі використання систем штучного інтелекту. Не дивлячись на їх різноманітність, задачі, які в них вирішуються, можна описати такою формальною схемою: відомо цільовий стан деякої системи, її поточний стан і операції (правила), з допомогою яких можна переходити від одного стану до наступного. Відомі також обмеження, які накладаються на ті чи інші переходи. На основі цих знань інтелектуальна система має перевести систему з поточного стану в цільовий оптимальним шляхом.

1.4 Розв'язування задач

До цієї галузі відносяться розв'язування математичних і логічних задач, доведення теорем. Роботи в цьому напрямку були розпочаті в кінці 50-х років американськими вченими А. Ньюеллом, П. Саймоном і Дж. Шоу, які досліджували процеси розв'язання різних задач. Результатами їхніх робіт стали такі програми, як «ЛОГІК-ТЕОРЕТИК»,

призначена для доведення теорем у численні висловлень, і «УНІВЕРСАЛЬНИЙ ВИРІШУВАЧ ЗАДАЧ» (GPS – General Problem Solver). За допомогою програми ЛТ було доведено 38 із 52 теорем з книги «Принципи математики» авторів Уайтхеда і Рассела. Для виведення цільових тверджень (теорем), з відомих фактів (аксіом) в цій програмі використовується не алгоритмічні, а евристичні правила. Програма GPS показала вміння вирішувати задачі взагалі без зв'язку з конкретною предметною галуззю. Для цього попередньо потрібно задати структуру характерних станів і операторів (фактів і правил) на абстрактному рівні (наприклад, в задачі про козу, капусту і вовка потрібно задати можливі розміщення і пересування човна). Така процедура специфікації називається описом проблемного середовища. За допомогою програми GPS вирішувались логічні задачі, шахові задачі, шкільні алгебраїчні задачі та інші.

Евристичний метод розв'язання задачі авторами наведених вище програм розглядався як притаманний людському мисленню «взагалі», для якого характерним є виникнення здогадів про шляхи розв'язання задачі з наступною їх перевіркою. Приблизно в той час, коли роботи Ньюелла і Саймона стали привертати до себе увагу, в Масачусетському технологічному інституті, Стенфордському університеті і Стенфордському дослідницькому інституті також сформувалися дослідницькі групи в галузі ШІ. На противагу раннім роботам Ньюелла і Саймона ці дослідження більше відносилися до формальних математичних подань. Способи розв'язання задач у цих дослідженнях розвивалися на основі розширення математичної і символічної логіки. Починаючи з 1960 р., було розроблено ряд програм, здатних знаходити доказ теорем у численні предикатів першого порядку. Ці програми мають, за словами американського фахівця в галузі ШІ Дж. Маккарті, «здоровим глуздом», тобто здатністю робити дедуктивні висновки. У програмі К. Гріна й ін., що реалізує систему типу «запит – відповідь», знання записуються мовою логіки предикатів у вигляді набору аксіом, а питання, що задаються машині, формулюються у вигляді теорем, що підлягають доведенню. Великий інтерес становить «інтелектуальна» програма американського математика Хао Ванга. Ця програма за 3 хвилини роботи ІВМ-704 вивела 220 відносно простих лем і теорем з фундаментальної математичної монографії, а потім за 8,5 хвилин видала доведення ще 130 більш складних теорем, частина з яких ще не була виведена математиками. Правда, дотепер жодна програма не сформулювала і не довела ні однієї теореми, яка б, що називається, «конче» була б потрібна математикам і була б принципово новою.

1.5 Застосування систем ШІ в іграх

Найпершими інтелектуальними задачами, що стали вирішуватися за допомогою ЕОМ, були логічні ігри (шашки, шахи). Американський

кібернетик А. Самуель склав для обчислювальної машини програму, що дозволяє їй грати в шашки, причому в ході гри машина навчається, чи, принаймні, створює враження, що навчається, поліпшуючи свою гру на основі накопиченого досвіду. У 1962 р. ця програма змагалася з Р. Нілі, найсильнішим шашкістом у США, і перемогла. В машину було програмно закладено правила гри так, що вибір чергового ходу підлягав цим правилам. На кожній стадії гри машина вибирала черговий хід з множини можливих ходів відповідно до деякого критерію якості гри. У шашках (як і в шахах) звичайно не вигідно втрачати свої фігури, і, навпаки, вигідно брати фігури супротивника. Гравець (це може бути людина чи машина), що зберігає рухливість своїх фігур і право вибору ходів та в той самий час тримає під ударом велику кількість полів на дошці, звичайно грає краще свого супротивника, що надає значення цим елементам гри. Описані критерії гарної гри зберігають свою силу протягом усієї гри, але є й інші критерії, що відносяться до окремих її стадій – дебюту, міттельшпілю, ендшпілю.

Розумно поєднуючи такі критерії (наприклад, у вигляді лінійної комбінації з коефіцієнтами, що експериментально підбираються, чи більш складним чином), можна для оцінення чергового ходу машини одержати деякий числовий показник ефективності – оцінну функцію. Тоді машина, порівнявши між собою показники ефективності чергових ходів, вибере хід, що відповідає найбільшому показнику. Подібна автоматизація вибору чергового ходу не обов'язково забезпечує оптимальний вибір, але все-таки це якийсь вибір, і на його основі машина може продовжувати гру, удосконалюючи свою стратегію (спосіб дії) у процесі навчання на минулому досвіді. Формальне навчання складається в підстроюванні параметрів (коефіцієнтів) оцінної функції на основі аналізу проведених ходів і ігор з урахуванням їх результату. На думку А. Самуеля, машина, що використовує цей вид навчання, може навчитися грати краще, ніж середній гравець, за відносно короткий період часу.

Прикладом складної інтелектуальної гри можна вважати шахи. Перший міжнародний шаховий турнір машин, забезпечених відповідними програмами, відбувся у 1974 р. Як відомо, перемогу на цьому турнірі одержала радянська машина із шаховою програмою «Каїсса». Іншим відомим шаховим комп'ютером є Deep Blue – шаховий суперкомп'ютер, розроблений компанією IBM, який 11 травня 1997 здобув перемогу в матчі з 6 партій над чемпіоном світу з шахів Гаррі Каспаровим з рахунком 3,5:2,5. Перед цим у лютому 1996 року Каспаров виграв матч у попередньої версії цього комп'ютера з рахунком 4:2. В основі Deep Blue II знаходиться сервер RS/6000 фірми IBM, у якого є 32 процесори. До кожного з цих процесорів підключено по 8 спеціалізованих шахових процесорів. Таким чином всього Deep Blue II містить 256 шахових процесорів.

На сьогодні чемпіон світу з шахів серед комп'ютерів є Deep Fritz. Його встановлено на восьмипроцесорний суперкомп'ютер. Зросла не тільки обчислювальна потужність комп'ютера, але й, якщо можна так висловитись, його досвід. В 2002 Deep Fritz зіграв внічию 4:4 у матчі в Бахреїні проти чемпіона світу з класичних шахів Володимира Крамника. У листопаді 2003 року, Fritz X3D, версія Deep Fritz з тривимірним інтерфейсом, зіграв внічию в чотирьох партіях проти Гаррі Каспарова. З 25 листопада по 5 грудня 2006 Deep Fritz зіграв шість ігрових партій проти Крамника в Бонні і переміг з рахунком 4:2.

Підсумовуючи матч з машиною «Deep Junior» в 2003 році Каспаров поділився думкою про те, що через кілька років у людини вже не буде жодних шансів у протистоянні з шаховими програмами

1.6 Експертні системи

На початку вісімдесятих років у дослідженнях зі штучного інтелекту сформувався самостійний напрямок, що одержав назву «експертні системи» (ЕС). Мета досліджень в галузі ЕС полягає в розробці програм, що під час розв'язання задач, важких для експерта-людини, одержують результати, що не поступаються за якістю й ефективністю перед рішеннями, одержуваними екпертом. Дослідники в галузі ЕС для назви своєї дисципліни часто використовують також термін «інженерія знань», введений Е. Фейгенбаумом як «привнесення принципів і інструментарію досліджень з галузі штучного інтелекту в розв'язанні важких прикладних проблем, що потребують знань експертів».

Експертною системою називається система штучного інтелекту, призначена для вирішення задач в конкретній предметній області, яка використовує знання і правила виведення для вирішення задач, що є важкими для людей-експертів. Джерелом знань для неї є експерти – висококваліфіковані і вузькоспеціалізовані спеціалісти, які є визнаними особистостями в цій предметній області діяльності

Програмні засоби (ПЗ), що базуються на технології експертних систем чи інженерії знань (надалі будемо використовувати ці поняття як синоніми), одержали значне поширення у світі. Важливість експертних систем полягає в такому:

- технологія експертних систем істотно розширює коло практично значимих задач, розв'язуваних на комп'ютерах, вирішення яких приносить значний економічний ефект;

- технологія ЕС є найважливішим засобом у вирішенні таких глобальних проблем традиційного програмування, як тривалість і, отже, висока вартість розробки складних додатків;

- об'єднання технології ЕС з технологією традиційного програмування додає нові якості до програмних продуктів за рахунок: забезпечення динамічної модифікації додатків користувачем, а не

програмістом; більшої «прозорості» додатка; кращої графіки, інтерфейсу і взаємодії.

На сьогодні існує більше 150 різних напрямків, в яких успішно використовуються експертні системи. Приблизно в 20 % випадків їх використання визнано, що без їх застосування вирішення поставлених задач було б неможливим.

Головними напрямками застосування експертних систем є вирішення задач діагностування в медицині, техніці, біології і хімії. На наступному місці стоять експертні системи для автоматизації проектування та контролю електронних, механічних і хімічних систем. До найбільш цікавих галузей застосування експертних систем відносяться планові ігри з аналізом комплексних «сценаріїв» під час врахування різноманітних і важкодоступних до пізнання внутрішніх та зовнішніх взаємодій і залежностей.

Наразі перспективні ЕС знаходять такі застосування:

- ЕС відіграють провідну роль у всіх фазах проектування, розробки, виробництва, розподілу, продажу, підтримки і надання послуг;
- технологія ЕС комерційного призначення забезпечує революційний прорив в інтеграції додатків з готових інтелектуально-взаємодіючих модулів.

Типова ЕС складається з компонентів, зображених на рис. 1.2.

- вирішувача (інтерпретатора);
- робочої пам'яті (РП), яку також називають базою даних (БД);
- бази знань (БЗ);
- компонентів придбання знань;
- пояснювального компонента;
- діалогового компонента.

База даних (робоча пам'ять) призначена для збереження вихідних і проміжних даних розв'язуваної в поточний момент задачі. Цей термін збігається за назвою, але не за змістом з терміном, використовуваним в інформаційно-пошукових системах (ІПС) і системах керування базами даних (СУБД) для позначення всіх даних (насамперед довгострокових), збережених у системі.

База знань (БЗ) у ЕС призначена для збереження довгострокових даних, що описують розглянуту область (а не поточних даних), і правил, що описують доцільні перетворення даних цієї області.

Вирішувач, використовуючи вихідні дані з робочої пам'яті і знання з БЗ, формує таку послідовність правил, що, будучи застосованими до вихідних даних, приводять до вирішення задачі.

Компонент **придбання знань** автоматизує процес наповнення ЕС знаннями, здійснюваний користувачем-експертом.

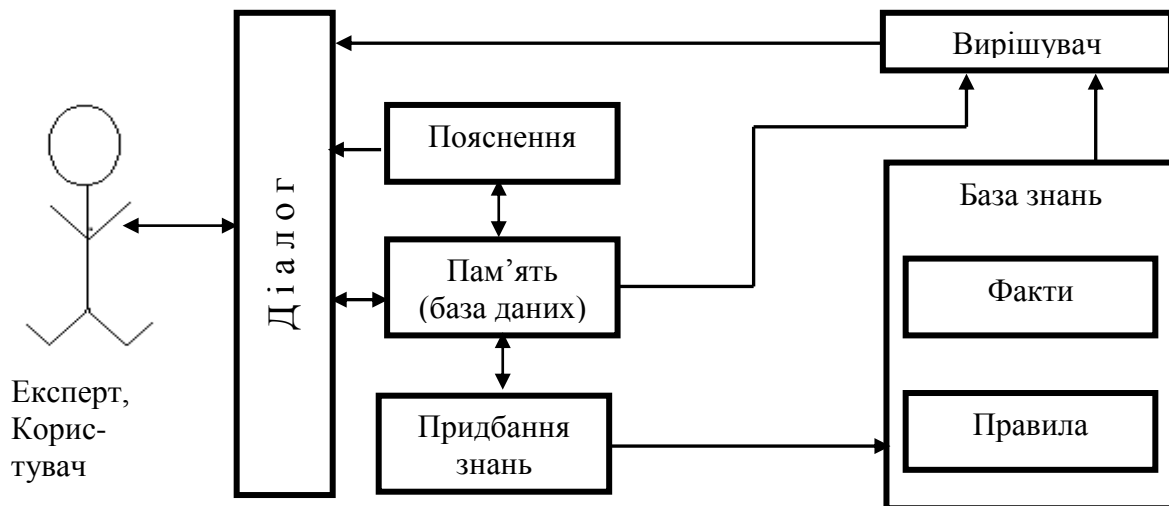


Рисунок 1.2 – Узагальнена схема експертної системи

Пояснювальний компонент пояснює, як система одержала рішення задачі (чи чому вона не одержала рішення) і які знання вона під час вирішення використовувала, що полегшує експерту тестування системи і підвищує довіру користувача до отриманого результату.

Діалоговий компонент орієнтований на організацію дружнього спілкування з усіма категоріями користувачів як в процесі вирішення задач, так і набуття знань.

Розглянемо характеристики окремих ЕС.

Експертна система MYSIN використовується для діагностики інфекційних захворювань в умовах стаціонарного лікування і вибору антимікробної теорії. Вона спочатку запитує необхідні дані про пацієнта, симптоми захворювання, реєструє гіпотези відносно можливих причин захворювання. Є однією з перших в світі ефективних експертних систем. Розроблена в 1972 році в Стенфордському університеті США. Реалізує приблизно 1200 правил і побудована на основі мови ЛІСП.

Експертна система DART розроблена в 1981 році і призначена для діагностики несправностей в електронних схемах.

ЕС SYN призначена для проектування великих інтегральних схем і розроблена в 1983 році.

Експертна система XCON базується на мові програмуванні OPS-5 і призначена для конструювання конфігурації обчислювальної системи і вибору її компонентів. Розроблена в 1981 році.

Система ISIS призначена для керування виробничими процесами, складання технологічних процесів. Розроблена в 1982 році на базі мови IRL.

Експертні системи ART і TWAICE призначені для програмування експертних систем. Оболонка експертної системи TWAICE реалізована

мовою ПРОЛОГ. За її допомогою можуть бути реалізовані системи для вирішення проблем діагностування і класифікації в техніці, медицині, народному господарстві.

ЕС, призначені для розробки експертних систем в конкретній предметній області, нині набули найбільшого поширення. До найвідомішої із них можна віднести G2 (Gensym, США); наступні місця зі значним відставанням (реалізовано менше 50 % можливостей G2) займають ЕКО (РосНДІ ІТ та АП, Росія), RTWorks (Talarian США), COMDALE/C (Comdale Techn., Канада), COGSYS (SC, США), ILOG Rules (ILOG, Франція).

Класи задач, для яких призначена G2 і подібні їй системи:

- моніторинг у реальному масштабі часу;
- системи керування верхнього рівня;
- системи виявлення несправностей;
- діагностика;
- складання розкладів;
- планування;
- оптимізація;
- системи - порадики оператора;
- системи проектування.

Інструментальні засоби фірми Gensym є еволюційним кроком у розвитку традиційних експертних систем від статичних предметних областей до динамічних. Чималу частку успіху фірмі Gensym забезпечують основні принципи, яких вона дотримується у своїх нових розробках:

- проблемно/предметна орієнтація;
- відповідність стандартам;
- незалежність від обчислювальної платформи;
- сумісність знизу-вверх з попередніми версіями;
- універсальні можливості, незалежні від розв'язуваної задачі;
- забезпечення технологічної основи для прикладних систем;
- комфортне середовище розробки;
- пошук нових шляхів розвитку технології;
- розподілена архітектура клієнт-сервер;
- висока продуктивність.

Основною перевагою оболонки експертних систем G2 для є можливість застосовувати її як інтегровальний компонент, що дозволяє за рахунок відкритості інтерфейсів і підтримки широкого спектра обчислювальних платформ легко об'єднати вже існуючі, розрізнені засоби автоматизації в єдину комплексну систему керування, що охоплює всі аспекти виробничої діяльності – від формування портфеля замовлень до керування технологічним процесом і відвантаження готової продукції. Це особливо важливо для вітчизняних підприємств, парк технічних і

програмних засобів яких формувався переважно безсистемно, під впливом різких коливань в економіці.

Крім системи G2, як базового засобу розробки, фірма Gensym пропонує комплекс проблемно/предметно-орієнтованих розширень для швидкої реалізації складних динамічних систем на основі спеціалізованих графічних мов, що містять параметризовані операторні блоки для подання елементів технологічного процесу і типових задач обробки інформації. Набір інструментальних середовищ фірми Gensym, згрупований за проблемною орієнтацією, охоплює всі стадії виробничого процесу і виглядає в такий спосіб:

- інтелектуальне керування виробництвом – G2, G2 Diagnostic Assistant (GDA), NeurOn-Line (NOL), Statistical Process Control (SPC), BatchDesign_Kit;

- оперативне планування – G2, G2 Scheduling Toolkit (GST), Dynamic Scheduling Packadge (DSP);

- розробка і моделювання виробничих процесів – G2, ReThink, BatchDesign_Kit;

- керування операціями і корпоративними мережами – G2, Fault Expert.

G2 можна назвати бестселером на ринку програмних продуктів – на початок 1996 р. у світі було встановлено більше 5000 її копій. Фірма Gensym обслуговує більш 30 галузей – від аерокосмічних досліджень до виробництва харчових продуктів. Список користувачів G2 виглядає як довідник Who-Is-Who у світовій промисловості. 25 найбільших індустріальних світових корпорацій використовують G2. На базі G2 написано більш 500 діючих додатків.

Прикладом інструментальних засобів для створення статичних ЕС на території СНД є комплекс ЕКО, розроблений в РосНДІ ІТ і АП. Найбільш успішно комплекс застосовується для створення ЕС, що вирішують задачі діагностики (технічної і медичний), евристичного оцінювання (ризик, надійності і т. д.), якісного прогнозування, а також навчання.

Комплекс ЕКО використовується: для створення комерційних і промислових експертних систем на персональних ЕОМ, а також для швидкого створення прототипів експертних систем з метою визначення застосовності методів інженерії знань у деякій конкретній проблемній області. На основі комплексу ЕКО було розроблено більш 100 прикладних експертних систем.

1.7 Робототехніка

Термін робототехніка (Robotics) уведений письменником-фантастом Айзеком Азімовим у 1941 році в оповіданні «Lisag» («Брехун»). **Робототехніка** – прикладна наука, що займається розробкою автоматизованих технічних систем (роботів). Вона орієнтована на

створення роботів і робототехнічних систем, призначених для автоматизації складних технологічних процесів і операцій, зокрема таких, що виконуються в недетермінованих умовах, для заміни людини під час виконання важких, втомливих і небезпечних робіт. Інше означення терміна робототехніка (таке, що відповідає сучасному агентному підходу до систем штучного інтелекту) : **робототехніка** – це науково-технічний напрямок, присвячений створенню інтелектуальних агентів, які виконують свої маніпуляції у фізичному світі.

Слово «робот» з'явилося вперше у п'єсі Карела Чапека «R.U.R.» (Россумські Універсальні Роботи) в 1921 році. Три основні правила, яким має підкорятися поведінка робота, сформульовані в 1942 році автором-фантастом Айзеком Азімовим в оповіданні «Хоровод»:

- робот не може зашкодити людині або, не діючи, допустити, щоб людині було завдано шкоди;

- робот має підкорятися наказам, які дає йому людина за винятком випадків, коли такі накази суперечать першому правилу

- робот має захищати своє існування до тих пір, поки цей захист не суперечить першому чи другому правилам.

Роботи – це фізичні агенти, які виконують поставлені перед ними завдання, проводячи маніпуляції у фізичному світі. Робот може безпосередньо підкорятися командам оператора, може працювати за задалегідь складеною програмою або слідувати набору загальних вказівок з допомогою технології штучного інтелекту.

Для виконання поставленої мети роботів оснащують виконавчими механізмами, такими як ноги, колеса, шарніри і схоплювачі. Виконавчі механізми мають єдине призначення – докладати фізичні зусилля до середовища. Крім того, роботів оснащують датчиками, які дозволяють їм сприймати дані про навколишнє середовище. У сучасних роботах застосовуються різні види датчиків, включно й ті, що призначені для вимірювання характеристик середовища (наприклад, відеокамери й ультразвукові далекоміри), і ті, які вимірюють характеристики руху самого робота (наприклад, гіроскопи і акселерометри).

Більшість сучасних роботів відноситься до однієї з трьох основних категорій. **Роботи-маніпулятори**, або **роботи-руки**, фізично «прив'язані» до свого робочого місця, наприклад на заводському складальному конвеєрі або на борту Міжнародної космічної станції. У русі маніпулятора зазвичай бере участь увесь ланцюжок керованих шарнірів, що дозволяє таким роботам встановлювати свої виконавчі механізми в будь-яку позицію в межах свого робочого простору. Маніпулятори відносяться до типу найбільш поширених промислових роботів, оскільки в усьому світі встановлено понад мільйон таких пристроїв. Деякі мобільні маніпулятори використовуються в лікарнях як асистенти хірургів. Без робототехнічних маніпуляторів в наші дні не зможуть продовжувати свою виробничу

діяльність більшість автомобільних заводів, а деякі маніпулятори використовувалися навіть для створення оригінальних художніх творів.

До другої категорії відносяться **мобільні роботи**. Роботи такого типу пересуваються в межах свого середовища з використанням коліс, ніг або аналогічних механізмів. Вони знайшли своє застосування для доставки обідів в лікарнях, під час переміщення контейнерів у вантажних доках, а також у разі виконання аналогічних завдань. До мобільних роботів відносяться також автоматичні транспортні засоби: наземний (Unmanned Land Vehicle – ULV) NavLab; повітряний (Unmanned Air Vehicle – UAV); підводний (Autonomous Underwater Vehicle – AUV). Перший здатний автономно пересуватися по автомагістралях в режимі самоводіння. Другий зазвичай використовується для повітряного спостереження, хімічної обробки земельних ділянок і військових операцій. Третій використовується в глибоководних морських дослідженнях. До мобільних також відноситься робот-планетоход, такий як робот Sojourner.

До третього типу відносяться **гібридні пристрої** – мобільні роботи, обладнані маніпуляторами. До них відносяться роботи-гуманоїди, які за своєю фізичною конструкцією нагадують людське тіло (як приклад можна назвати роботи японської корпорації Honda). Гібридні роботи здатні поширити дію своїх виконавчих елементів на більш широку робочу область порівняно з прикріпленими до одного місця маніпуляторами, але змушені виконувати завдання, що стоять перед ними, з великими зусиллями, оскільки не мають такої жорсткої опори, яку надає вузол кріплення маніпулятора.

Робототехніка буває будівельною, промисловою, побутовою, авіаційною, екстремальною (військовою, космічною, підводною). До сфери робототехніки відносяться також протезні пристрої (штучні кінцівки, вушні і очні протези для людей), інтелектуальні системи життєзабезпечення (наприклад, цілі будинки, обладнані датчиками і виконавчими механізмами), а також багатоелементні системи, в яких робототехнічні дії здійснюються з використанням цілих полчищ невеликих роботів, які об'єднують свої зусилля.

Робототехніка в усьому світі розвивається дуже стрімко. Уже сьогодні роботи літають в космос і беруть участь в бойових діях, водять автомобілі та вчать дітей в школах, роблять хірургічні операції і грають на театральній сцені. Збільшується їх кількість. Підвищується якість виконуваних функцій. Багато держав вважають робототехніку ключовим напрямком економічного розвитку країни і вкладають у відповідні дослідження мільйони доларів. Виробництво роботів в ХХІ столітті може стати найбільшою галуззю промисловості.

Експерти Міжнародної федерації робототехніки відзначають, що в промисловості використовується найбільше роботів – приблизно 770 000. Причому половина з них – 350 000 працюють в Японії. В Європі ж використовується 233 000, а в Північній Америці – 104 000 промислових

роботів і використовуються вони, переважно, на складальних конвеєрах (дані за 2009 рік). Роботи до 2025 року зможуть виконувати роботу, для якої у Японії потрібно було б зайняти 3,5 мільйона чоловік.

1.8 Машинний переклад мови

Проблема машинного перекладу мови (МП) виникла в 50-х роках практично одночасно з появою перших комп'ютерів через необхідність автоматизації обробки користувачами великих обсягів інформації, поданої іншомовними текстами. З тих пір вона пройшла через декілька стадій розвитку, внаслідок чого нині розроблено і використовується декілька десятків комерційних систем машинного перекладу мови, з яких найбільш відомими є STILUS і PROMT (ПРОМТ, Росія), Power Translator (компанія Globalink, США), Language Assistant (компанія MicroTas, США), TRANSEND (компанія Intergraph, США), Pragma (компанія Trident Softwer, Україна), ATLAS (фірма «Фудзіцу», Японія), PENSEE (корпорація Oki Computer, Японія), PIVOT (компанія Nippon Electric, Японія) та інші.

В найзагальнішому випадку під **машинним перекладом** розуміють передавання змісту висловлювань, сформульованих однією мовою, засобами іншої мови за допомогою комп'ютерної програми. Вперше думку про можливість машинного перекладу висловив Чарльз Беббідж (1791–1871), який розробив у 1836–1848 роках проект цифрової аналітичної машини – механічного прототипу електронних цифрових обчислювальних машин, що з'явилися через 100 років. Фактично історія машинного перекладу починається з «Джорджтаунського експерименту». У січні 1954 р. відбулася перша публічна демонстрація машинного перекладу з російської мови на англійську, здійсненого на машині IBM-701. Повідомлення про цю подію було опубліковано в журналі Computers and Automation, 1954, № 2.

Перші роботи з машинного перекладу в Європі були виконані в тогочасному СРСР у Всесоюзному Інституті наукової і технічної інформації під керівництвом Д. Ю. Панова і реалізовані на машині БЭСМ в 1955 році, а також в Математичному інституті АН СРСР під керівництвом Ляпунова А. А. і реалізовані на ЕОМ «Стрела». Ці роботи поклали початок **першому поколінню систем машинного перекладу мови**. Системи першого покоління базувались на алгоритмах послідовного перекладу «слово за словом», «фраза за фразою». Можливості таких систем визначалися доступними розмірами словників, що прямо залежали від обсягу пам'яті комп'ютера. Переклад тексту здійснювався окремими реченнями, змістовні зв'язки між ними ніяк не враховувались. Такі системи називаються системами прямого перекладу. Перший період розвитку систем машинного перекладу закінчився в 1967 році з появою доповіді Комісії Національної Академії наук США. Виходячи з реальної ситуації з перекладами в Сполучених Штатах і з оцінки показників

вартості різних способів перекладу, комісія прийшла до висновку про нерентабельність застосування машинного перекладу в умовах США. Хоча комісія і рекомендувала проводити подальші дослідження в цій галузі, її рішення призвели до різкого зниження інтересу до проблеми машинного перекладу, що істотно загальмувало його розвиток загалом.

Другий період розвитку машинного перекладу почався з нового підйому робіт у цій галузі в середині 70-х років. В цей час з'явилась ціла низка практично діючих дослідницьких і комерційних систем. На користь відродження МП в цей період діяла низка об'єктивних факторів, таких як, з одного боку, зростання потреби в перекладах, зумовлене розширенням співробітництва різних країн, а з іншого, бурхливим розвитком можливостей комп'ютерної техніки. Вісімдесяті роки стали часом переходу до широкого практичного використання систем машинного перекладу в багатьох країнах, утворився ринок комерційних систем. Розроблено ряд глобальних проєктів, таких як EUROTRA (європейське економічне співтовариство), ARIANE (Франція), METAL (США і Німеччина), KANT (США), SUSY (Німеччина), MU (Німеччина). Появився значний досвід експлуатації, обговоренню якого було присвячено багато конференцій. В цей час на зміну системам прямого перекладу прийшли системи наступних поколінь, в яких переклад тексту з мови на мову здійснюється на рівні синтаксичних структур. В алгоритмах перекладу використовувався набір операцій, який дозволяв шляхом аналізу речення, що перекладалось, побудувати його синтаксичну структуру за правилами граматики вхідного речення, а потім перетворити її в синтаксичну структуру вихідного речення і синтезувати вихідне речення шляхом підстановки потрібних слів із словника. Такі системи називаються Transfer- або T- системи (transfer – перетворення).

З початку 90-х років почався **третій період розвитку технології машинного перекладу**, який був ініційований черговим зростанням інтересу до машинного перекладу, пов'язаним з бурхливим розвитком Інтернет і необхідністю online-перекладу. В цей період продовжують розвивати започаткований в деяких попередніх проєктах підхід до побудови систем машинного перекладу на основі отримання деякого, незалежного від мов, подання змісту вхідного речення шляхом його семантичного аналізу (перехід до глибинної форми) з наступним синтезом після цього вихідного речення (перекладу) з отриманого смислового подання. Такі системи МП називають I-системами (аббревіатура від слова «Інтерлінгва»). Системи типу INTERLINGUA припускають апріорі наявність деякої метамови структур, на якій можна описати всі структури як вхідної, так і вихідної мов у загальному випадку; тому алгоритм перекладу в системі типу INTERLINGUA передбачається як більш простий: аналіз вхідного речення в термінах метамови і потім синтез з метаструктури відповідного речення вихідної мови. «Єдина» складність у цьому випадку – розробити саму метамову і описати природну мову у

відповідних термінах. Тому нині нічого не відомо про практично діючі системи типу INTERLINGUA.

З кінця 90-х років почався **новий період в розробці систем МП**, який характеризується повторенням на більш високому рівні принципу прямого перекладу, що використовувався в перших системах МП. Цей підхід започатковано в лабораторії статистичного машинного перекладу Університету Джона Хопкінса під керівництвом професора Девіда Яровскі. Він полягає в використанні відповідностей між так званими n-грамами вхідного і вихідного текстів для прямого перекодування першого в другий. Ці відповідності знаходяться статистичними методами шляхом навчання на вибірках масивів вхідного і правильно перекладеного тексту, в цьому разі лінгвістичні характеристики текстів не використовуються. На такому принципі працює система машинного перекладу Google Translator.

1.9 Інтелектуальний аналіз даних (Data Mining)

Data Mining переводиться як «видобуток» чи «розкопка даних». Нерідко поруч з Data Mining зустрічаються слова «виявлення знань у базах даних» (knowledge discovery in databases) і «інтелектуальний аналіз даних». Їх можна вважати синонімами Data Mining. Виникнення всіх зазначених термінів пов'язано з новим витком у розвитку засобів і методів обробки даних.

Мета Data Mining полягає у виявленні прихованих правил і закономірностей у наборах даних. Справа в тому, що людський розум сам по собі не пристосований для сприйняття великих масивів різномірної інформації. Людина до того ж не здатна уловлювати більше двох-трьох взаємозв'язків навіть у невеликих вибірках. Але і традиційна математична статистика, яка довгий час претендувала на роль основного інструмента аналізу даних, також нерідко пасує під час розв'язання задач з реального складного життя. Вона оперує усередненими характеристиками вибірки, що часто є фіктивними величинами (типу середньої температури пацієнтів у лікарні, середньої висоти будинку на вулиці, що складається з палаців і халуп і тощо). Тому методи математичної статистики виявляються корисними переважно для перевірки заздалегідь сформульованих гіпотез (verification-driven data mining).

Сучасні технології Data Mining (discovery-driven data mining) перелопачують інформацію з метою автоматичного пошуку шаблонів, характерних для яких-небудь фрагментів неоднорідних багатовимірних даних. На відміну від оперативної аналітичної обробки даних (online analytical processing, OLAP) у Data Mining тягар формулювання гіпотез і виявлення незвичайних (unexpected) шаблонів перекладено з людини на комп'ютер.

Таблиця 1.1 – Приклади формулювань задач за використання методів OLAP і Data Mining

| OLAP | Data Mining |
|---|--|
| Які середні показники травматизму для курящих і некурящих? | Які фактори найкраще пророкують нещасливі випадки? |
| Які середні розміри телефонних рахунків існуючих клієнтів у порівнянні з рахунками колишніх клієнтів (що відмовилися від послуг телефонної компанії)? | Які характеристики відрізняють клієнтів, що, цілком ймовірно, збираються відмовитися від послуг телефонної компанії? |
| Яка середня величина щоденних покупок по украденій і неукраденій кредитній картці? | Які схеми покупок характерні для шахрайства з кредитними картками? |

В постановці задачі Data Mining немає особливої новизни. Подібні задачі вирішувались протягом декількох останніх десятиріч років («пошук емпіричних закономірностей», «евристичний пошук у складних середовищах», «індуктивний висновок» і т. п.). Але тільки зараз суспільство повністю дозріло для розуміння практичної важливості і широти цих задач. По-перше, з огляду на розвиток технологій запису і збереження даних сьогодні на людей обрушилися колосальні потоки інформаційної руди у всіляких галузях, що без продуктивної переробки загрожують перетворитися в нікому не потрібні смітники. По-друге, засоби і методи обробки даних стали доступними і зручними, а їхні результати зрозумілими будь-якій людині.

Насамперед методи Data Mining сьогодні користуються попитом у комерційних підприємств, що розгортають проекти на основі інформаційних сховищ даних (Data Warehousing). Досвід багатьох таких підприємств показує, що віддача від використання Data Mining може досягати 1000 %. Наприклад, відомі повідомлення про економічний ефект, що у 10–70 разів перевищив початкові затрати від 350 до 750 тис. дол. Наводяться дані про проект у 20 млн. дол., що окупився усього за 4 місяці. Інший приклад – річна економія 700 тис. дол. за рахунок упровадження Data Mining у мережі універсамів у Великобританії.

Data Mining становлять велику цінність для керівників і аналітиків у їхній повсякденній діяльності. Ділові люди усвідомили, що за допомогою методів Data Mining вони можуть отримати відчутні переваги в конкурентній боротьбі.

1.10 Нейронні мережі

У наші дні зростає необхідність в системах, які здатні не тільки виконувати один раз запрограмовану послідовність дій над задалегідь

визначеними даними, але і здатні самі аналізувати інформацію, що заново надходить, знаходити в ній закономірності, здійснювати прогнозування та ін. У цій галузі застосувань найкращим чином зарекомендували себе так звані нейронні мережі (НМ) – системи, що імітують діяльність людського мозку і здатні до самонавчання.

Нейронні мережі перевершують послідовні машини у вирішенні тих задач, в яких машину перевершує людина. Задачі, що потребують великого обсягу обчислень або високої точності, краще виконуються звичайним комп'ютером. До завдань, що успішно вирішуються НМ на цьому етапі їх розвитку, відносяться:

- розпізнавання зорових, слухових образів. До цього напряму відноситься широка область застосувань: від розпізнавання тексту і цілей на екрані радара до систем голосового управління;
- асоціативний пошук інформації та створення асоціативних моделей;
- синтез мовлення та формування природної мови;
- формування моделей різних нелінійних та інших систем, що важко піддаються описанню, та прогнозування розвитку цих систем в часі;
- застосування на виробництві;
- прогнозування розвитку циклонів і інших природних процесів, прогнозування змін курсів валют та інших фінансових процесів;
- системи управління і регулювання з прогнозом;
- управління роботами, іншими складними пристроями – різноманітні кінцеві автомати;
- системи масового обслуговування та комутації, телекомунікаційні системи;
- прийняття рішень та діагностика, що виключають логічний висновок, особливо в галузях, де відсутні чіткі математичні моделі: медицині, криміналістиці, фінансовій сфері.

Крім того, сучасні нейронні мережі мають ряд додаткових можливостей – вони дозволяють оцінювати порівняльну важливість різних видів вхідної інформації, зменшувати її обсяг без втрати істотних даних, розпізнавати симптоми наближення критичних ситуацій і т. д.

В найзагальнішій формі **під нейромережею розуміють автомат, призначений для моделювання способів, в які мозок виконує окремі завдання і функції**, що становлять інтерес. Нейромережі зазвичай реалізуються у вигляді електронної схеми або шляхом програмної імітації на комп'ютерах.

Нейронна мережа являє собою сукупність елементів двох типів – процесорів, що називаються нейронами, і елементів зв'язку (в загальному випадку також процесорів) між нейронами.

Дослідження, пов'язані з розробкою пристроїв переробки інформації на основі принципів роботи природних нейронних систем, відносяться до

галузі нейроінформатики або нейрообчислень (нейрокомп'ютинга). Терміни ці з'явилися недавно – в середині 80-х років.

Суть всіх підходів нейроінформатики – розробка методів створення (синтезу) нейронних схем, які вирішують ті чи інші завдання. Нейрон в цьому випадку виглядає як пристрій дуже простий, щось на кшталт підсилювача з великим числом входів і одним виходом. Різниця між підходами і методами – в деталях уявлень про роботу нейрона, і, звичайно, в уявленнях про роботу зв'язків. Власне, як вже зазначалося вище, пристрої нейроінформатики є зв'язковими системами. На відміну від цифрових мікропроцесорних систем, що являють собою складні комбінації процесорних і запам'ятовувальних блоків, нейропроцесор містить пам'ять, розподілену в зв'язках між дуже простими процесорами. Тим самим основне навантаження на виконання конкретних функцій процесорами лягає на архітектуру системи, деталі якої теж визначаються міжнейронними зв'язками.

Навчання зазвичай будується так: існує збірник задач – набір прикладів із заданими відповідями. Ці приклади надаються системі. Нейрони отримують з вхідних зв'язків сигнали – «умови прикладу», перетворюють їх, кілька разів обмінюються перетвореними сигналами і, нарешті, видають відповідь – також набір сигналів. Відхилення від правильної відповіді штрафуються. Навчання складається в мінімізації штрафу як (неявної) функції зв'язків.

Можна назвати такі раціональні основи широкого розвитку в останні роки нейроінформатики і нейрокомп'ютинга:

- нейрокомп'ютери дають стандартний спосіб вирішення багатьох нестандартних задач. І не важливо, що спеціалізована машина краще вирішить деякий один клас задач. Важливіше, що один нейрокомп'ютер вирішить і цю задачу, і іншу, і третю – і не потрібно кожен раз проектувати спеціалізовану ЕОМ, нейрокомп'ютер зробить все сам і майже не гірше;

- замість програмування – навчання. Нейрокомп'ютер вчиться – потрібно лише формувати навчальні збірники задач. Праця програміста заміщується новою роботою – вчителя (можна сказати – тренера чи дресирувальника). Програміст задає машині всі деталі роботи, вчитель – створює «освітнє середовище», до якого пристосовується нейрокомп'ютер;

- нейрокомп'ютери особливо ефективні там, де потрібна подібність до людської інтуїції – для розпізнавання образів (впізнавання осіб, читання рукописних текстів), переведення з однієї природної мови на іншу тощо. Саме для таких задач звичайно важко скласти явний алгоритм;

- гнучкість структури: можна різними способами комбінувати прості складові нейрокомп'ютерів – нейрони і зв'язки між ними. За рахунок цього на одній елементній базі і навіть усередині «тіла» одного нейрокомп'ютера можна створювати абсолютно різні машини. З'являється ще одна нова професія – «нейроконструктор» (конструктор мізків);

- нейронні мережі дозволяють створити ефективне програмне забезпечення для високопаралельних комп'ютерів. Для високопаралельних машин добре відома проблема: як їх ефективно використовувати, тобто як домогтися, щоб усі елементи одночасно і без зайвого дублювання обчислювали що-небудь корисне? Створюючи математичне забезпечення на базі нейронних мереж можна для широкого класу задач вирішити цю проблему.

Головна перевага нейронних мереж – це їхні здібності до навчання та їхня розподілена структура. Головними недоліками є неможливість інтегрування знань в нейронну мережу та витягування їх з неї й опис їхніх параметрів. Кількість нейронів та їх зв'язки, параметри процесу навчання можуть бути вибрані тільки експериментально.

1.11 Прийняття рішень в умовах невизначеності і браку інформації

В процесі своєї життєдіяльності кожна людина вимушена приймати рішення, пов'язані з вибором того чи іншого варіанта дій (альтернативи), необхідного для досягнення поставленої мети. Вибір найкращого, оптимального рішення має такий зміст: вибрати найкраще рішення, що допускається обставинами.

Під **прийняттям рішень (ПР)** (англійською – «decision-making») розуміють процес людської діяльності, спрямований на вибір найкращого варіанта дій (тобто такого, який приводить до отримання найкращого результату).

Теорія прийняття рішень – галузь науки, яка використовує поняття і методи математики, статистики, економіки, менеджменту і психології з метою вивчення закономірностей вибору людьми шляхів вирішення різного роду задач, а також способів пошуку найбільш вигідних з можливих рішень.

В процесах прийняття рішень можна виділити **такі етапи**:

- 1) пошук інформації;
- 2) пошук та знаходження множини допустимих альтернатив (варіантів);
- 3) порівняння та вибір найкращої альтернативи;
- 4) реалізація обраної альтернативи;
- 5) контроль результатів.

Перші два етапи є слабоформалізованими, але на третьому етапі можливо застосування наукових підходів.

В загальному випадку проблеми прийняття рішень можуть бути структуровані, тобто описані, такою п'ятіркою:

$$\langle Q, A, I, B, D \rangle,$$

де Q – критерій;

A – множина варіантів (альтернатив);

I – наявна інформація;

B – особливості особи, яка приймає рішення (ОПР);

D – правила вибору (прийняття рішення – ПР).

Якщо деякі елементи цієї п'ятірки не описані, або описані не повністю, проблема прийняття рішення є слабо (погано) структурованою. Якщо проблема ПР взагалі не структурована, то неможливо застосувати наукові методи до її вирішення.

Теорія прийняття рішень припускає, що рішенням, пов'язаним з ризиком, завжди властиві елементи невідомості конкретної поведінки вихідних параметрів, які не дозволяють чітко детермінувати значення кінцевих результатів цих рішень. Залежно від ступеня невідомості майбутньої поведінки вихідних параметрів прийняття рішень розрізняють **умови ризику**, в яких імовірність настання окремих подій, що впливають на кінцевий результат, може бути встановлена з тим або іншим ступенем точності, та **умови невизначеності**, в яких через відсутність необхідної інформації така ймовірність не може бути встановлена.

Задача прийняття рішень (ПР) виникає, коли існує декілька варіантів дій (альтернатив) для досягнення заданого або бажаного результату. В цьому разі потрібно вибрати найкращу (в певному сенсі) альтернативу.

Загальну постановку задачі прийняття рішень можна сформулювати таким чином:

Нехай X – множина альтернатив, Y – множина можливих наслідків (результатів) і між вибором певної альтернативи $x_i \in X$ та настанням відповідного результату $y_i \in Y$ існує деякий причинний зв'язок. Крім того, передбачається наявність механізму оцінення якості такого вибору – звичайно оцінюється якість результату. В деяких випадках доцільно вважати, що ми маємо можливість безпосередньо оцінювати якість альтернативи x_i , тоді множину можливих наслідків Y можна виключити з розгляду. Потрібно вибрати найкращу альтернативу, для якої відповідний результат має найкращу оцінку якості.

Теорія вибору і прийняття рішень, а також теорія оптимізації можуть бути використані в таких областях, як:

- задача вибору оптимальної номенклатури товару в торгових та інших організаціях;

- задача вибору персоналу у фірмі (наприклад, під час прийому на роботу);

- задача раціональної організації розробки програмного забезпечення для комп'ютерних систем;

- задачі, що вирішуються в ріелторських фірмах, які надають послуги населенню на ринку нерухомості (наприклад, підбір квартир);

- задача оптимального вибору параметрів (числових характеристик) будь-якої системи (або організації) – проектованої або реально існуючої;

- формування оптимальних стратегій поведінки на ринку цінних паперів;
- задача прийняття рішень на фінансовому ринку в умовах ризику і невизначеності;
- задача реалізації оптимальних стратегій заміни обладнання;
- задача максимізації доходів в умовах аукціонних торгів та ін.

Як правило, більшість реальних інженерних задач містить у тому чи іншому вигляді невизначеність. Можна навіть стверджувати, що рішення задач з урахуванням різного виду невизначеностей є загальним випадком, а прийняття рішень без їх урахування – окремим. Однак, через концептуальні і методичні труднощі нині не існує єдиного методологічного підходу до вирішення таких задач. А втім, накопичено достатньо велику кількість методів формалізації постановки і прийняття рішень з урахуванням невизначеностей. Під час використання цих методів потрібно мати на увазі, що всі вони носять рекомендаційний характер і вибір остаточного рішення завжди залишається за людиною (ОПР – особою, що приймає рішення).

Під час вирішення конкретних задач з урахуванням невизначеностей інженер стикається з різними їх типами. У дослідженні операцій прийнято розрізняти три типи невизначеностей:

- невизначеність цілей;
- невизначеність наших знань про навколишнє оточення і діючі в цьому явищі фактори (невизначеність природи);
- невизначеність дій активного або пасивного партнера або противника.

Окрім розглянутої вище класифікації невизначеностей потрібно враховувати їх тип з погляду відношення до випадковості.

За цією ознакою можна розрізняти стохастичну (імовірнісну) невизначеність, коли невідомі чинники статистично стійкі і тому є звичайні об'єкти теорії ймовірностей – випадкові величини (або випадкові функції, події тощо). Водночас мають бути відомі або визначені в процесі постановки завдання всі необхідні статистичні характеристики (закони розподілу та їх параметри). Прикладом таких завдань можуть бути, зокрема, система технічного обслуговування і ремонту будь-якого виду техніки, система організації робіт з її огляду тощо.

Іншим крайнім випадком може бути невизначеність нестохастичного вигляду, за якої ніяких припущень про стохастичну стійкість не існує. Нарешті, можна говорити про проміжний тип невизначеності, коли рішення приймається на підставі будь-яких гіпотез про закони розподілу випадкових величин. Водночас ОПР потрібно мати на увазі небезпеку незбіжності її результатів з реальними умовами. Ця небезпека незбіжності формалізується за допомогою коефіцієнтів ризику.

Невизначеність цілей потребує залучення будь-яких гіпотез, що допомагають в отриманні однозначних рішень. В цьому випадку

врахування фактора невизначеності мети призводить до необхідності розгляду іншої проблеми, яка формулюється у вигляді проблеми прийняття оптимальних багатоцільових рішень.

Методологія прийняття рішення в умовах ризику та невизначеності передбачає побудову в процесі обґрунтування ризикових рішень так званої «матриці рішень», яка має такий вигляд (табл. 1.2).

Таблиця 1.2 – Матриця рішень

| Варіанти альтернатив прийняття рішень | Варіанти ситуацій розвитку подій | | | |
|---------------------------------------|----------------------------------|----------|-----|----------|
| | Y_1 | Y_2 | ... | Y_n |
| X_1 | E_{11} | E_{12} | ... | E_{1n} |
| X_2 | E_{21} | E_{22} | ... | E_{2n} |
| ... | | | ... | |
| X_n | E_{n1} | E_{n2} | ... | E_{nn} |

У наведеній матриці значення X_1, X_2, \dots, X_n характеризують кожен з варіантів альтернатив прийняття рішення; значення Y_1, Y_2, \dots, Y_n – кожен з можливих варіантів результатів; значення $E_{11}, E_{12}, E_{1n}, E_{21}, E_{22}, E_{2n}, E_{n1}, E_{n2}, \dots, E_{nn}$ – конкретний рівень ефективності рішення, відповідний певній альтернативі за певного результату (певної ситуації розвитку подій).

Наведена матриця рішень характеризує один з її видів, що позначається як «матриця виграшів», оскільки вона розглядає показник ефективності. Можлива також побудова матриці рішень іншого виду, що позначається як «матриця ризиків», в якій замість показника ефективності використовується показник фінансових втрат, що відповідають певним сполученням альтернатив прийняття рішень і можливих ситуацій розвитку подій (результатів).

На основі зазначеної матриці розраховується найкраще з альтернативних рішень за обраним критерієм. Методика цього розрахунку диференціюється для умов ризику та умов невизначеності.

I. Прийняття рішень в умовах ризику ґрунтується на тому, що для кожної альтернативної ситуації розвитку подій може бути задана певна ймовірність її здійснення. Це дозволяє зважити кожне з конкретних значень ефективності (втрат) значенням ймовірності кожної окремої альтернативи та отримати на цій основі інтегральний показник рівня ризику, що відповідає кожній з альтернатив прийняття рішень. Порівняння цього інтегрального показника за окремими альтернативами дозволяє

обрати для реалізації ту з них, яка призводить до обраної мети (заданому показнику ефективності) з найменшим рівнем ризику.

Оцінка ймовірності реалізації окремих ситуацій розвитку подій може бути отримана експертним шляхом.

Виходячи з матриці рішень, побудованої в умовах ризику з урахуванням ймовірності реалізації окремих ситуацій, розраховується інтегральний рівень ризику за кожною з альтернатив прийняття рішень.

II. Прийняття рішень в умовах невизначеності оснований на тому, що ймовірності різних варіантів ситуацій розвитку подій суб'єкта, що приймає ризикове рішення, не відомі. В цьому випадку під час вибору альтернативи прийнятого рішення суб'єкт керується, з одного боку, своїми ризиковими перевагами, а з іншого – відповідним критерієм вибору з усіх альтернатив за складеною ним «матрицею рішень».

Задача обґрунтування рішень в умовах невизначеності всіх типів, крім апріорного, зводиться до звуження вихідної множини альтернатив на основі інформації, якою володіє особа, що приймає рішення (ОПР). Якість рекомендацій для прийняття рішень в умовах стохастичної невизначеності підвищується за умови врахування таких характеристик особистості ОПР, як ставлення до своїх вигравів і програшів, схильність до ризику. Обґрунтування рішень в умовах апріорної невизначеності можливо побудовою алгоритмів адаптивного управління.

Для вирішення багатьох нестандартних завдань потрібні нетрадиційні методи, які використовують як формалізовані методи вирішення (математичні), так і неформалізовані (методи штучного інтелекту). В останні роки було розроблено добре обґрунтовані методології прийняття рішень. В сфері формалізованих задач розвивається новий напрям – математична теорія ризику і безпеки. В сфері погано формалізованих задач активно розвиваються методи штучного інтелекту, основані на формалізації знань. Розвиток цих методів на першому етапі було багато в чому пов'язаний зі створенням експертних систем. Нині також широко використовуються методи нечіткої математики (Fuzzy Mathematic).

1.12 Розпізнавання образів

В сучасних умовах глобальної автоматизації виробництва потрібно, щоб машина виконувала замість людини деякі функції розпізнавання. Наприклад, в таких традиційних застосуваннях, як збирання механічних вузлів необхідно, щоб робот розпізнавав потрібні деталі. Автоматичне сортування деталей чи предметів (яблук, томатів і т. ін.) теж потребує наявності у технічних пристроїв властивостей розпізнавання. В деяких випадках швидкість і вірогідність розпізнавання людиною недостатня для реалізації тих чи інших задач. Зокрема, комп'ютер, забезпечений пристроєм введення флюорограм і алгоритмом розпізнавання, набагато

швидше і точніше ставить діагноз можливого захворювання. Потреби в комплексній механізації і автоматизації виробництва, створенні роботів, у вирішенні задач технічної та медичної діагностики, метеорологічного прогнозу, розпізнаванні зорових і мовних образів та ін. викликали до життя значні за обсягом наукові і технічні роботи, спрямовані на вирішення теоретичних і прикладних питань проблеми розпізнавання.

Розпізнавання образів – це віднесення деяких об'єктів або явищ до певної групи (класу) за допомогою виділення їх істотних ознак із загальної маси несуттєвих даних. **Образ, клас** – класифікаційне групування в системі класифікації, що об'єднує (виділяє) певну групу об'єктів за ознаками їх схожості. Задача розпізнавання образів є основною в більшості інтелектуальних систем.

Предмет розпізнавання образів об'єднує ряд наукових дисциплін, їх пов'язує пошук рішення загальної задачі – виділити елементи, що належать конкретному класу, серед множини розмитих елементів, що відносяться до декількох класів.

Перші спроби вивчення можливостей автоматизації процесу розпізнавання образів відносяться до початку 50-х років, коли цифрові обчислювальні машини поступово стали загальнодоступним засобом обробки інформації. Частина цих початкових робіт в сфері, галузі розпізнавання образів була присвячена розробці програм автоматичного прийняття рішень та створення спеціалізованої апаратури, призначеної для автоматичного «читання» образів типу друкованих літерно-цифрових знаків. В кінці 50-х років Розенблатт запропонував перцептронний алгоритм, який являв собою одну з перших моделей процесів запам'ятовування і організації інформації, реалізованих мозком. Дослідження в сфері синтезу систем розпізнавання набирали темп протягом 60-х років у міру того, як розширювалося використання обчислювальних машин і ставала очевидною потреба в більш швидкому і ефективному зв'язку людини з ЕОМ. Для того щоб під час вирішення деяких типів задач розпізнавання зорових образів можна було користуватися результатами теорії машинних мов і відповідними можливостями обробки інформації, був запропонований синтаксичний підхід як додаток до аналітичних методів.

На сьогодні основні концепції теорії розпізнавання набувають все більшого визнання як фактора, істотного для побудови сучасних інформаційних систем, реалізованих на основі обчислювальних машин. Інтерес до цього напрямку продовжує швидко зростати – відповідні задачі є об'єктами міждисциплінарних досліджень, що проводяться в рамках таких різних галузей, як техніка, обчислювальна математика і кібернетика, теорія інформації, статистика, фізика, хімія, лінгвістика, психологія, біологія, фізіологія і медицина. У кожній з них основна увага приділяється якомусь конкретному аспекту задачі розпізнавання – від моделювання

фізіологічних процесів до розвитку аналітичних методів автоматичного прийняття рішень.

Як приклад найпоширеніших застосувань розпізнавання образів можна назвати:

- оптичне розпізнавання символів (OCR);
- розпізнавання штрих-кодів;
- розпізнавання автомобільних номерів;
- розпізнавання осіб;
- розпізнавання мови;
- розпізнавання зображень;
- розпізнавання локальних ділянок земної кори, в яких знаходяться родовища корисних копалин;
- класифікація документів;
- наділення чуттєвими органами (слухом, зором, нюхом) роботів.

Контрольні питання та завдання

1. Дайте означення штучного інтелекту і інтелектуальної системи.
2. Назвіть основні напрями застосування систем штучного інтелекту.
3. В чому суть агентного підходу до штучного інтелекту?
4. Які завдання виконує раціональний агент?
5. Які етапи можна виділити в історії розвитку штучного інтелекту та в чому їх зміст?
6. Які характерні риси людського інтелекту наслідуються системами штучного інтелекту?
7. З яких основних компонент складається інтелектуальна система?
8. Охарактеризуйте основні досягнення інтелектуальних систем в сфері розв'язання задач.
9. Назвіть приклади застосування систем штучного інтелекту в іграх.
10. Що таке експертна система і з яких основних компонент вона складається?
11. Назвіть приклади застосування експертних систем.
12. Дайте означення понять «робототехніка» та «робот».
13. Які категорії роботів існують і які особливості різних категорій роботів?
14. Назвіть основні напрямки застосування роботів.
15. Що розуміють під машинним перекладом мови?
16. Наведіть приклади сучасних комерційних систем машинного перекладу.
17. Які періоди можна виділити в історії розвитку систем машинного перекладу мови?

18. Назвіть основні характеристики різних поколінь систем машинного перекладу мови.
19. В чому полягає суть статистичного підходу до машинного перекладу мови?
20. В чому полягає основна мета технології Data Mining?
21. Поясніть відмінність обробки даних за технологією Data Mining від статистичної обробки під час пошуку закономірностей в їх структурі.
22. Назвіть сфери застосування технології Data Mining.
23. Завдання якого типу вирішуються нейронними мережами успішніше, ніж комп'ютерами?
24. Що таке нейромережа і в якому вигляді вона реалізується?
25. Дайте означення нейроінформатики.
26. З яких компонентів складається нейрокомп'ютер?
27. Назвіть основні причини широкого застосування нейрокомп'ютинга в останні роки.
28. В чому полягають головні переваги і недоліки нейромереж?
29. Що називають прийняттям рішень?
30. Що є предметом науки «теорія прийняття рішень»?
31. Опишіть структури проблеми прийняття рішень.
32. Чим відрізняються умови ризику від умов невизначеності?
33. Назвіть основні типи невизначеностей.
34. Зробіть загальну постановку задачі прийняття рішень.
35. Що називається розпізнаванням образів?
36. Назвіть основні сфери застосування теорії розпізнавання образів.

РОЗДІЛ 2 ЗАГАЛЬНІ ПОНЯТТЯ ТЕОРІЇ РОЗПІЗНАВАННЯ ОБРАЗІВ

2.1 Означення терміна «розпізнавання образів»

Під розпізнаванням образів (РО) розуміється віднесення об'єктів до деяких виділених груп за їх схожістю. Під об'єктами маються на увазі предмети, явища, процеси, ситуації, дії тощо. Однією з найцікавіших властивостей мозку людини є його здатність відповідати на нескінченну множину станів зовнішнього середовища скінченною кількістю реакцій. Завдяки цій властивості інформація про зовнішній світ зберігається в згорнутому стані, що дає можливість збільшити її обсяг для запам'ятовування. Наприклад, одне і те саме слово, вимовлене різними особами, має для людини один і той самий зміст.

Приклад 2.1. Визначити, яку літеру алфавіту означає подана множина символів: $\{\alpha, a, A, A, A\}$.

Розв'язання: зі свого досвіду читання різних текстів з різними стилем напису ми знаємо, що наведені символи означають літеру алфавіту А.

В цьому випадку зрозуміло, що розпізнавання людиною образів зводиться до оцінення відносного шансу на те, що вхідні дані відповідають тій чи іншій з відомих множин статистичних сукупностей, які визначаються минулим досвідом людини і становлять апіорну інформацію для розпізнавання.

Приклад 2.2. Об'єднати між собою подібні зображення, наведені на рис. 2.1

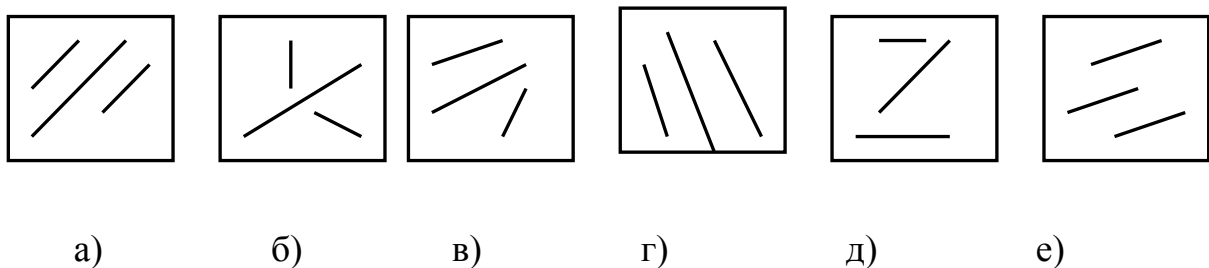


Рисунок 2.1 – Приклади зображень

Розв'язання: На зображеннях а), г), е) лінії або паралельні, або перетинаються під дуже малим кутом; на зображеннях б), в), д) лінії перетинаються під великим кутом.

З цього прикладу випливає, що людина об'єднує об'єкти і явища зовнішнього світу в групи за деякими спільними властивостями і присвоює цим групам однаково поняття.

Властивість людини відносити низку подібних явищ до одного і того самого поняття називається **розпізнаванням**. Поняття, яке формується

людиною про деяку множину подібних між собою об'єктів зовнішнього світу, називається **образом**. Розпізнавання полягає, таким чином, в перетворенні вхідної інформації, в ролі якої доречно розглядати деякі параметри, ознаки об'єктів, в вихідну, що являє собою висновок про те, до якого образу відноситься об'єкт, що підлягає розпізнаванню.

Перед тим, як визначити основні опорні поняття теорії розпізнавання, розглянемо на деякому прикладі змістовну інтерпретацію задачі розпізнавання. За такий приклад візьмемо задачу розпізнавання стороною А літаків сторони В.

Для побудови системи розпізнавання літаків сторони В сторона А має зібрати і проаналізувати всю доступну інформацію про авіацію противника, визначити їх тактико-технічні характеристики і визначити ті рішення, які вона може приймати щодо знешкодження літаків нападаючої сторони своїми засобами протиповітряної оборони. Наприклад, якщо літаки поділяються на три класи: 1 – бомбардувальники, 2 – винищувачі, 3 – штурмовики, то шляхом математичного моделювання для мінімізації втрат сторона А призначила класи засобів протидії S1 проти літаків 1-го класу, S2 проти літаків 2-го класу і сумісно S1 і S2 проти літаків 3-го класу.

Після проведення класифікації літаків сторони В потрібно визначити, за допомогою яких параметрів чи ознак можна описати задані класи літаків, тобто із знайденого переліку вилучити ті ознаки, які не можуть бути реально визначені за допомогою технічних засобів спостереження за літаками (радіолокатори, оптичні пристрої, лазери, акустичні засоби, тепलोкатори та інш.). І, врешті-решт, на основі отриманих апріорно ознак (наприклад, крейсерська та максимальна швидкості, гранична висота польоту, кількість двигунів і їх тип, довжина фюзеляжу, розлогість крил, тощо) потрібно в алфавіті вибраних ознак описати кожен клас літаків. Вигляд опису залежить від типів ознак, якими описується той чи інший клас. На цьому підготовча робота закінчується. Тепер припустимо, що з допомогою технічних засобів спостереження знайдено ознаки виявленого невідомого літака. Порівняння отриманих апостеріорних даних з апріорними описами класів літаків, що зберігаються у вигляді еталонів в пам'яті системи розпізнавання, дозволяє визначити, до якого класу відноситься цей літак, тобто виконати його розпізнавання.

Наведений приклад дає можливість скласти уявлення про задачу розпізнавання в такій інтерпретації. Є в наявності деяка сукупність об'єктів чи явищ. Згідно з обраним принципом класифікації вона поділена на низку класів, тобто складено алфавіт класів. Вибрано алфавіт ознак, мовою якого описується кожен клас об'єктів. Створено технічні засоби, що забезпечують вимірювання параметрів об'єктів, їх попередню обробку, виділення ознак і формування еталонних описів класів в пам'яті обчислювальної системи. В обчислювальній системі також реалізовано алгоритм розпізнавання, який дозволяє порівнювати отриманий ознаковий

опис невідомого об'єкта (апостеріорну інформацію) з еталонними описами всіх класів літаків (апріорною інформацією) і на основі порівняння визначати, до якого класу він може бути віднесений. Коли з'являється об'єкт, що підлягає розпізнаванню, технічні засоби спостереження здійснюють вимірювання і визначають його ознаки. Ознаковий опис надходить на вхід алгоритму розпізнавання, який порівнює його з еталонами і приймає рішення про його класифікацію. Таким чином, узагальнена схема системи розпізнавання має вигляд, зображений на рис. 2.2.

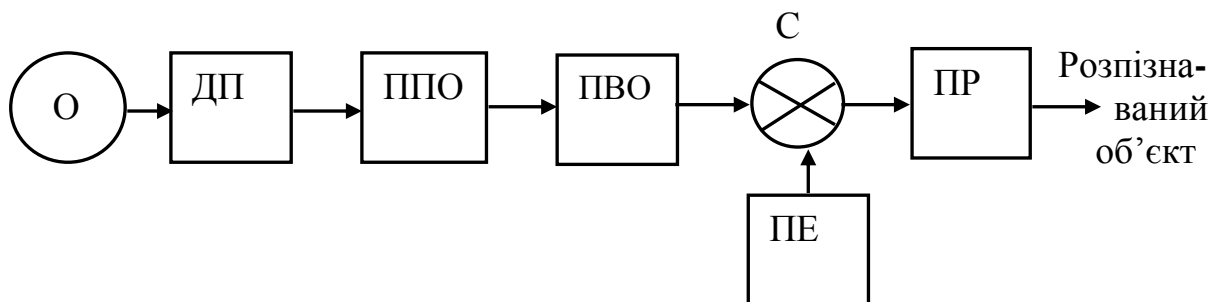


Рисунок 2. 2 – Узагальнена структурна схема системи розпізнавання образів: О – об'єкт; ДП – детектор параметрів об'єкта; ППО – пристрій попередньої обробки параметрів об'єкта; ПВО – пристрій виділення ознак; СП – схема порівняння; ПЕ – пам'ять еталонів; ПР – схема прийняття рішень на розпізнавання

2.2 Основні поняття теорії образів

Дамо змістовне означення основних понять теорії розпізнавання образів.

В кожній з задач розпізнавання аналізуються деякі явища, процеси, стани навколишнього середовища, які далі будуть називатися об'єктами зовнішнього світу ω_i , а їх сукупність – множиною об'єктів

$$\Omega = \{\omega_1, \omega_2, \dots, \omega_n, \dots, \omega_N\}. \quad (2.1)$$

Нехай чуттєві органи (давачі) системи вимірювання дозволяють вимірювати деяку множину Y параметрів об'єкта.

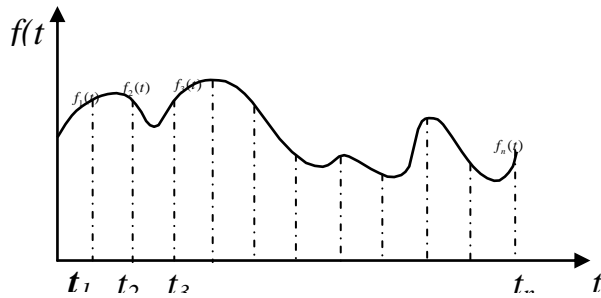
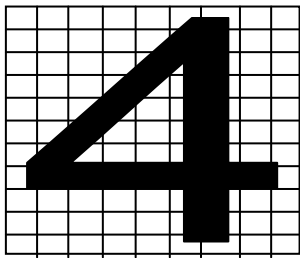
$$Y = \{y_1, y_2, \dots, y_l, \dots, y_L\}. \quad (2.2)$$

Тоді вектор $\vec{y}_i = (y_{i1}, y_{i2}, \dots, y_{il}, \dots, y_{iL})^T$ конкретних значень цих параметрів (де T – символ транспонування) для деякого об'єкта ω_i з множини Ω називається його **образом** або реалізацією:

$$\omega_i \rightarrow \vec{y}_i.$$

Для зручності запису матричних і векторних операцій в подальшому будемо вважати \vec{y}_i вектором-стовпцем.

На рис. 2.3 наведено приклади зображень \vec{y} різних об'єктів – символу як просторового образу на площині та сигналу як часової функції $f(t)$.



а)

$$\vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{99} \end{pmatrix}, y_i = 0 \text{ або } 1$$

б)

$$\vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, y_i = f(t_i)$$

Рисунок 2.3 – Приклади зображень об'єктів: а) – символа, б) – неперервного сигналу

Координати $y_1, y_2, \dots, y_b, \dots, y_L$ утворюють простір вимірювань, а точки, що відповідають кінцям векторів \vec{y}_i , утворюють зображення об'єктів множини Ω в просторі вимірювань.

Далі нехай на основі емпіричного досвіду або математичними методами розроблено деяке відображення F простору вимірювань Y в простір інформативних параметрів X , $X = \{x_1, x_2, \dots, x_q, \dots, x_Q\}$:

$$X = F(Y). \quad (2.3)$$

В теорії розпізнавання параметри X , які містять інформацію про характерні спільні властивості окремих об'єктів ω_i з множини Ω , називаються **ознаками**.

Відношення подібності, задане ознаками X на множині Ω , задає розбиття об'єктів цієї множини на класи образів:

$$\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_i, \dots, \Omega_k\}, k \leq N; \quad \Omega_i = \{\omega_{i1}, \omega_{i2}, \dots, \omega_{im}\}, m \leq N; \quad (2.4)$$

$$\forall_{i \neq j} (\Omega_i \wedge \Omega_j = \emptyset), \quad i, j = \overline{1, k}$$

Класом називають множину об'єктів, подібних між собою за деякими ознаками.

Кожен об'єкт ω_i у просторі ознак X зображається точкою, що є кінцем вектора X_i .

Кластером або **таксоном** називається область простору ознак, в яку потрапляють вектори об'єктів тільки одного класу.

Процедура віднесення невідомого зображення \vec{x} до одного з класів Ω_i називається **класифікацією**, а процедура, що складається з визначення множини класів (алфавіту образів) в просторі ознак – **кластеризацією**.

Під **розпізнаванням** розуміють процедуру, що складається з кластеризації і класифікації.

Ілюстрацію введених понять дамо на прикладі, зображеному на рисунку 2.4:

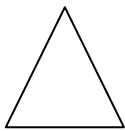
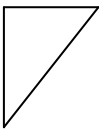
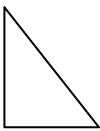
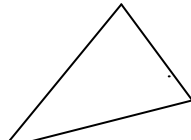
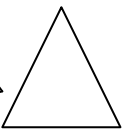
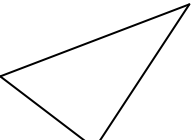
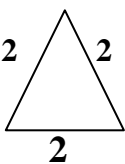
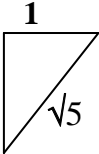
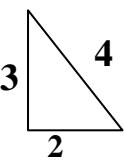
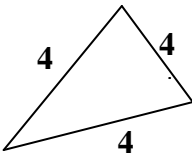
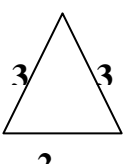
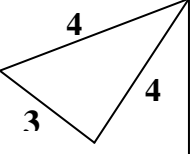
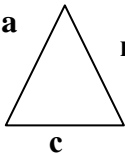
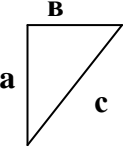
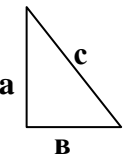
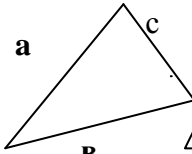
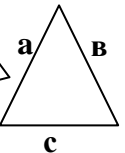
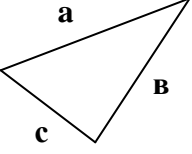
| | | | | | | |
|----------------|---|---|---|--|---|---|
| $\{\omega_i\}$ |  |  |  |  |  |  |
| $\{y_L\}$ |  |  |  |  |  |  |
| $\{x_j\}$ |  |  |  |  |  |  |

Рисунок 2.4 – Приклад класифікації множини трикутників:
 $\{\omega_i\}$ – досліджувані об'єкти; $\{y_L\}$ – вимірювані параметри; $\{x_j\}$ – ознаки

На рис. 2.4 показано шість геометричних фігур (трикутників), що становлять множину об'єктів $\Omega = \{\omega_1, \omega_2, \dots, \omega_6\}$.

За ознаки цих об'єктів взято довжини їх сторін: $x_1 = a$, $x_2 = b$, $x_3 = c$, які утворюють простір ознак $X = \{x_1, x_2, x_3\}$.

Виміри сторін кожного трикутника дають їх значення у просторі вимірювань:

$$\vec{y}_1 = (2,2,2)^T$$

$$\vec{y}_2 = (2,1,\sqrt{5})^T$$

$$\vec{y}_3 = (3,4,2)^T$$

$$\vec{y}_4 = (4,4,4)^T$$

$$\vec{y}_5 = (3,3,2)^T$$

$$\vec{y}_6 = (4,4,3)^T$$

Отримані дані зображуємо в тривимірному просторі ознак x_1, x_2, x_3 (рис. 2.5)

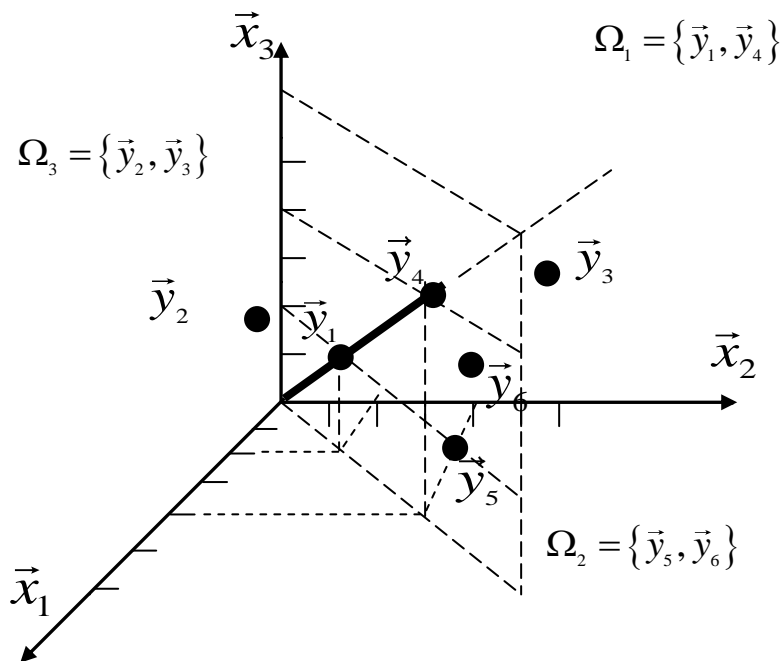


Рисунок 2.5 – Зображення об'єктів у просторі ознак

З цього рисунка видно, що точки $\{\vec{y}_1, \vec{y}_4\}$ діагоналі тривимірного куба $x_1 = x_2 = x_3$ утворюють клас Ω_1 рівносторонніх трикутників, точки $\{\vec{y}_5, \vec{y}_6\}$ площини $x_1 = x_2$, що не лежать на діагоналі – клас Ω_2 рівнобедрених трикутників, інша частина тривимірного простору являє собою клас $\Omega_3 = \{\vec{y}_2, \vec{y}_3\}$ всіх інших трикутників.

В цьому прикладі простір вимірювань збігається з простором ознак, тобто $\mathbf{X} = \mathbf{Y}$ і відображення $F = 1$, що характерно для множин об'єктів природи з явною структурованістю. В тих випадках, коли достатня

регулярність, яка виявляється у вигляді постійних зв'язків, закономірностей, відсутня, для надійної класифікації потрібно виконувати деяке перетворення простору вимірювань в простір ознак, в якому спостерігалась би структурованість множини об'єктів Ω у вигляді класів. Приклад такого перетворення показано на рис. 2.6.

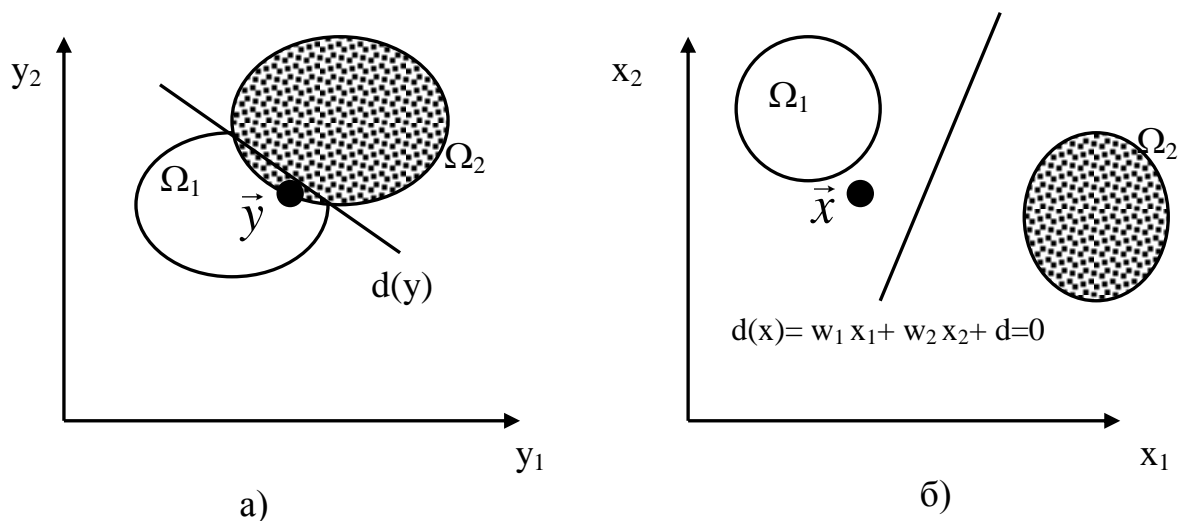


Рисунок 2.6 – Геометрична інтерпретація впливу перетворення простору вимірювань в простір ознак: а) в просторі вимірювань класи Ω_1 і Ω_2 перетинаються; б) класи в просторі ознак не перетинаються

З рис. 2.6 видно, що деформація простору вимірювань в простір ознак $\mathbf{X} = F(\mathbf{Y})$ дає можливість виконати безпомилковий розподіл об'єктів на класи Ω_1 і Ω_2 .

2.3 Основні задачі теорії розпізнавання образів

Задача 1. Визначення апріорного словника ознак

Ця задача призначена визначенню повного переліку параметрів, що можуть характеризувати об'єкти або явища, для розпізнавання яких ця система розробляється.

У вирішенні цієї задачі головне – знайти всі параметри, які описують суть об'єктів розпізнавання. Будь-які обмеження, будь-яка неповнота, можуть призвести до помилок або повної неможливості правильної класифікації об'єктів.

Можемо собі уявити таку неповноту у вже розглянутій нами задачі розпізнавання літаків як використання тільки однієї ознаки – стелю висоти польоту літаків. Результат – бомбардувальники не вдасться відрізнити від винищувачів (під час створення бомбардувальників прагнуть забезпечити максимально можливу висоту польоту, а під час створення винищувачів намагаються, щоб вони могли знищувати бомбардувальників).

Реально навіть ціла група ознак може виявитися неефективною.

Визначення повного словника апіорних ознак дає можливість після апостеріорного оцінення ефективності рішень системи розпізнавання, виявивши обмеженість обраних ознак, не повертатися до цієї задачі на наступних етапах розробки.

Вибір ознак на сьогодні здебільшого здійснюється людиною, тобто є евристичною операцією. Цей вибір можна здійснювати, маючи уявлення про їхні загальні властивості. З цих позицій достатньо прийняти, що ознаки можуть поділятися на:

- детерміновані;
- імовірносні;
- логічні;
- структурні;
- нечіткі.

Детерміновані ознаки – це такі характеристики об'єктів або явищ, що мають конкретні і постійні числові значення.

Прикладами детермінованих ознак можуть бути, наприклад, тактико-технічні характеристики бомбардувальників і винищувачів, що стоять на озброєнні в деякій державі. Числові значення ознак щодо кожного з літаків можна інтепретувати як координати точок, що являють їх у просторі ознак заданої розмірності.

Необхідно мати на увазі, що в задачах розпізнавання з детермінованими ознаками похибки вимірювання цих ознак не відіграють ніякої ролі, якщо, наприклад, точність вимірів такої ознаки як розмах крил літака значно вища (наприклад, 1 см), ніж відмінність цієї ознаки в різних класів літаків (наприклад, 10 м).

Уявити таку систему, де використовуються детерміновані ознаки, не так важко:

- розпізнавання монет за їх розміром у монетоприймачі автомата продажу води;
- розпізнавання на конвеєрі деталей за відмінностями геометричних характеристик, якщо похибки вимірювання істотно менші «розкидів» розмірів цих деталей.

Розпізнавання здійснюється шляхом порівняння отриманих розмірів із існуючими в базі даних характеристиками деталей.

Імовірнісні ознаки – це характеристики об'єкта (явища), що носять випадковий характер. З такими ознаками переважно і мають справу в природі та техніці.

Відрізняються ці ознаки тим, що в силу випадковості відповідного розміру ознака одного класу може набувати значення з області значень інших класів, кожний з яких підлягає розпізнаванню в системі.

Якщо ознака не може набути значення в області відповідних значень для інших класів, то маємо справу не з імовірсною, а з тією самою

детермінованою ознакою. Це підкреслює, чому імовірнісні системи є системами більш загального порядку.

Для того, щоб можна було для імовірнісних ознак говорити про можливість розпізнавання, варто зажадати, щоб можливості спостереження значень ознаки у своєму класі були значно більші, ніж у чужих. У протилежному випадку ця ознака не дозволить побудувати СР, що використовує опис класів на її основі. Ефективність її недостатня для достовірного рішення і необхідно шукати інші ознаки, що мають велику розподільчу здатність. Наведемо приклади імовірносних ознак розпізнавання:

- середнє значення потужності сигналу радіолокатора, відбитого від літака (причина – порізаність кругової діаграми розсіювання сигналу радіолокатора літаком і електронні й атмосферні завади в тому самому радіолокаційному діапазоні);

- розмір листка рослини (причини – відмінності в живленні, освітленості, вологості тощо);

- розмір патологічної зміни якогось органа людини (причини – різноманітні стадії захворювання за його виявлення, різноманітні ракурси і перетини спостережень утворення тощо).

Логічні ознаки розпізнавання – це характеристики об'єкта або явища, подані у вигляді елементарних висловлень про істинність («так», «ні») або «істина», «неправда»).

Ці ознаки, як ми розуміємо, не мають кількісного вираження, тобто є якісними судженнями про наявність або про відсутність деяких властивостей чи складових в об'єктів або явищ.

Приклади логічних ознак:

- наявність турбореактивного двигуна на літаку;

- біль у горлі пацієнта;

- кашель;

- нежить;

- розчинність реактиву і т. д.

Тут за кожною ознакою можна сказати тільки те, що вона є або її немає.

До логічних можна віднести також такі ознаки, у яких не важлива величина, а лише факт потрапляння або непотрапляння її в заданий інтервал (наприклад, крейсерська швидкість літака більша або менша за 2000 км/год).

Структурні ознаки – непохідні (тобто елементарні, не взяті з інших елементарних ознак) елементи (символи), примітиви зображення об'єкта розпізнавання.

Поява структурних ознак зобов'язана виникненню проблеми розпізнавання зображень із її специфічними особливостями і труднощами.

Приклади структурних ознак:

- а) для зображення прямокутника:

- горизонтальний відрізок прямої;
 - вертикальний відрізок прямої.
- б) для будь-якого зображення на екрані дисплея:
- піксел.

Оперуючи обмеженим числом атомарних (непохідним) елементів (примітивів), можна одержати опис різноманітних об'єктів. Тобто, для об'єктів, що відрізняються, можна мати набір однакових непохідних елементів. Але для того, щоб опис можна було б виконати, поряд із визначенням непохідних елементів мають вводитись правила комбінування, що визначають засоби побудови об'єкта зі згаданих непохідних елементів. Ці правила називаються граматиною. Результат – два однакових непохідних елементи різноманітних об'єктів можуть бути об'єднані один з одним за різними правилами. Це і буде їх відрізняти.

Загалом для опису якого-небудь об'єкта непохідні елементи об'єднуються в ланцюжки (речення) за своїм, характерним тільки для цього об'єкта, набором правил.

Внаслідок зв'язків із непохідних елементів (структурних ознак) утворюється об'єкт, аналогічний тому, як речення мови будуються шляхом об'єднання слів, які й собі складаються з літер. У цьому структурні методи проявляють аналогію із синтаксисом звичайної мови. Звідси структурні ознаки носять ще назву лінгвістичних або синтаксичних.

Нечіткі ознаки характеризують якісні знання людини про деякий об'єкт.

Для формального подання таких ознак об'єкта запропоновано формальний апарат нечіткої (fuzzy) логіки.

Нечітка підмножина A універсальної множини U визначається як множина упорядкованих пар $A = \{\mu_A(x)/x\}$, де x – елемент множини U , $x \subset U$, а $\mu_A(x) \rightarrow [0,1]$ – характеристична функція належності (або просто функція належності). Значення функції належності для конкретного елемента називається ступенем належності. Ступінь належності знаходиться в діапазоні $[0, 1]$. Чим вище ступінь належності, тим більшою мірою елемент x універсальної множини відповідає властивостям нечіткої підмножини A . Таким чином, нечітку множини A можна записати як

$$A = \sum_{i=1}^n (\mu(X_i) / X_i), \quad (2.5)$$

де X_i – i -тий елемент універсальної множини U , а знак « Σ » не є позначенням операції додавання, а має смисл об'єднання.

Лінгвістична змінна (ЛЗ) в теорії нечітких множин визначається як змінна, значення якої задається набором словесних характеристик деякої властивості. Наприклад, ЛЗ «вік» може мати такі значення (терми):

$$\text{ЛЗ} = \text{МлВ, ДВ, ОВ, ЮВ, МВ, ЗВ, ПВ, СВ,}$$

що означає вік немовляти, дитячий, підлітковий, юнацький, молодий, зрілий, похилий і старий, відповідно. Універсальна множина U в цьому випадку – це шкала прожитих людиною років $[0..120]$. Функція належності в цьому випадку визначає нашу впевненість у тому, що відому кількість прожитих років можна віднести до заданого значення ЛЗ. Нехай експертом до молодого віку віднесені люди у віці 20 років зі ступенем упевненості 0,8, у віці 25 років зі ступенем упевненості 0,95, у віці 30 років зі ступенем упевненості 0,95 і у віці 35 років зі ступенем упевненості 0,7. Отже:

$$\mu(X_1)=0,8; \mu(X_2)=0,95; \mu(X_3)=0,95; \mu(X_4)=0,7. \quad (2.6)$$

Значення ЛЗ=МВ можна записати:

$$\begin{aligned} \text{МВ} &= \mu(X_1)/X_1 + \mu(X_2)/X_2 + \mu(X_3)/X_3 + \mu(X_4)/X_4 = \\ &= 0,8/X_1 + 0,95/X_2 + 0,95/X_3 + 0,7/X_4 . \end{aligned} \quad (2.7)$$

де «+» – нечітка операція «АБО».

Таким чином, нечіткі ознаки дозволяють враховувати суб'єктивні думки окремих експертів про окремі характеристики об'єктів.

З вищенаведеного в розв'язанні задачі вибору апріорного словника ознак можна відзначити такі основні моменти:

1. Вибір, призначення ознак розпізнавання – евристична операція, що залежить від досвіду, творчості, винахідливості розробника.

2. Склад ознак, що вибираються на цьому етапі, має бути якомога різнобічнішим і повнішим, незалежно від того, можна чи не можна ці ознаки одержати.

3. Вибір ознак має здійснюватися в групах детермінованих, імовірнісних, логічних, структурних і нечітких.

Задача 2. Початкова класифікація об'єктів (явищ), які підлягають розпізнаванню, складання апріорного алфавіту класів

В попередніх розділах ця задача уже була подана на описовому рівні і полягає у виборі (призначенні) класів об'єктів розпізнавання. Найчастіше її розв'язання, як і вибір ознак розпізнавання, здійснюється евристично, під час цього послідовність виконання її етапів така:

- на першому етапі визначається мета розпізнавання, тобто які рішення можуть прийматися за результатами розпізнавання або людиною, або автоматичною системою керування об'єктом;

- на другому етапі відповідно до визначеної мети формулюються вимоги до системи розпізнавання, що дозволяють вибрати принцип класифікації;

- на третьому етапі складається апріорний алфавіт класів об'єктів.

Для прикладу візьмемо, що за результатами деякого методу медичної діагностики стану печінки людини необхідно приймати рішення про застосування одного з методів лікування за відомими протоколами

(відповідає 1-му пункту у розглянутій послідовності вирішення задачі апріорної класифікації – визначенню мети). Наскільки відповідальним є прийняття такого рішення, з огляду на можливість хірургічного втручання, зрозуміло.

Тому очевидною вимогою до системи є надійне (із високою імовірністю) діагностування кожного захворювання печінки (відповідає 2-му етапу послідовності).

Отже, в апріорний алфавіт класів необхідно внести всі можливі захворювання печінки, а їх – 11. Тобто, класів захворювань печінки, що розпізнаються і які діагностуються деякою гіпотетичною системою розпізнавання, має бути 11. До цих класів, які визначені попередньою медичною практикою, відносяться:

1. Гострий гепатит.
2. Хронічний гепатит.
3. Жирова інфільтрація.
4. Цирроз.
5. Кіста проста.
6. Кіста паразитарна.
7. Абсцес.
8. Пухлина.
9. Метастази.
10. Гематома.
11. Конкременти.

Потрібно зауважити, що окрім розглянутої в цьому прикладі ситуації, можливі й інші задачі, в яких кількість класів, за якими надійно розпізнаються деякі об'єкти (явища), заздалегідь невідома і має визначатися самою системою розпізнавання. Ця задача називається задачею кластеризації, розв'язання якої здійснюється як за допомогою евристичних підходів, так і з використанням формальних критеріїв.

Задача 3. Розробка словника інформативних ознак

В задачі № 1 потрібно було знайти всі можливі ознаки розпізнавання заданих об'єктів або явищ. Аналогічно під час розв'язання задачі № 2 визначався склад класів, об'єкти яких підлягають розпізнаванню.

За наявності цієї інформації на наступному кроці проводиться аналіз можливості забезпечення вимірювання ознак наявними в розробника технічними засобами, та їх інформативності, яка визначається вкладом кожної з ознак на точність розпізнавання. В разі необхідності розробником вносяться пропозиції про створення нових засобів вимірювань ознак для забезпечення необхідної ефективності розпізнавання, а гарний чи поганий набір ознак розпізнавання був отриманий внаслідок виконаних дій можна зрозуміти, виконавши випробування системи розпізнавання загалом й оцінивши ефективність розпізнавання. Оскільки системи розпізнавання на зазначеному етапі розробки ще не існує, то оцінити ефективність вибраного набору інформативних ознак на цьому кроці можна тільки

шляхом математичного моделювання. Завершений вибір оптимального з погляду швидкодії і точності розпізнавання алфавіту інформативних ознак можна реалізувати тільки на основі сумісної оптимізації всіх етапів побудови системи розпізнавання, включно і задачі побудови оптимального ознакового опису класів та розробки оптимальних алгоритмів розпізнавання.

Задача 4. Опис класів апріорного алфавіту мовою апріорного словника ознак

Апріорний опис класів – найбільш трудомістка і творча задача у процесі створення системи розпізнавання, що потребує глибокого вивчення властивостей об'єктів розпізнавання.

У рамках цієї задачі необхідно кожному класу поставити у відповідність числові параметри детермінованих, імовірнісних, логічних, структурних чи нечітких ознак.

Значення цих параметрів описів можна одержати із сукупності таких робіт і дій:

- спеціально поставлені експериментальні роботи або експериментальні постереження;
- результати обробки експериментальних даних;
- математичні розрахунки;
- результати математичного моделювання;
- витяги з літературних джерел.

Розглянемо принципи ознакового опису класу окремо для детермінованих, імовірносних, логічних, структурних і нечітких ознак.

Якщо ознаки об'єктів розпізнавання **детерміновані**, то описом класу може бути точка в M -вимірному просторі детермінованих ознак з апріорного словника, сума відстаней якої від точок, що представляють об'єкти цього класу, мінімальна.

Легко собі уявити такий еталон, повернувшись до розглянутих нами таблиць тактико-технічних характеристик літаків. Тут ми маємо справу з 11-вимірним простором ознак. Кожна координата – це одна якась характеристика, наприклад «екіпаж». Якщо розглядати тільки одну координату «екіпаж», то точкою еталону для винищувачів буде 1, для бомбардувальників – 4. Це точки, суми відстаней яких від усіх винищувачів і всіх бомбардувальників, що є представниками цих двох класів, мінімальні.

Точно так це можна зробити за всіма 11 координатами (тобто «стеля», «розмах крил», «бомбове навантаження» тощо), внаслідок чого будемо вже мати справу з точками еталонів у 11-вимірному просторі.

Якщо ознаки розпізнавання **логічні**, то для опису кожного класу необхідно насамперед мати повний набір елементарних логічних висловлювань A, B, C , що входять до складу апріорного словника. Для опису класів необхідно встановити відповідність між набором значень наведених ознак A, B, C і класами $\Omega_1, \Omega_2, \dots, \Omega_m$.

Для кращого розуміння (без претензій на медичну достовірність) розглянемо такий приклад:

необхідно розпізнати два захворювання – звичайна застуда й ангіна (Ω_1, Ω_2). Як логічні ознаки виберемо такі:

A – підвищена температура (A=0 – ні, A=1 – так);

B – нежить (B=0 – ні, B=1 – так);

C – нариви в горлі (C=0 – ні, C=1 – так).

Тоді так зване булеве співвідношення між класом W_1 (звичайне простудне захворювання) і значеннями ознак (а ці значення – бінарні) виглядає так:

$$\Omega_1 = ABC\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}BC\bar{C}.$$

Тут множення відповідає логічному «І», а додавання – «АБО». Для другого класу захворювань отримаємо такий опис

$$\Omega_2 = A\bar{B}C + \bar{A}\bar{B}C.$$

Якщо розподіл об'єктів розпізнавання, що подаються числовими значеннями їх ознак за областями відповідного простору **імовірнісний**, то для опису класів необхідно визначити характеристики цих розподілів. З теорії імовірності відомо, що такими характеристиками є:

- функції щільності розподілу імовірностей (ЩРІ) ознак $F_i(x_1, x_2, \dots, x_n)$, де x_1, x_2, \dots, x_n – імовірнісні ознаки, i – номер класу;

- $P(\Omega_i)$ – апіорна імовірність того, що об'єкт, випадково обраний із загальної сукупності, належить до класу Ω_1 .

Щоб одержати ЩРІ класів системи розпізнавання розробник СР може використати три засоби:

- експериментальне визначення за статистичними даними;
- теоретичний висновок;
- моделювання.

Те саме стосується і апіорної імовірності класу $P(\Omega_i)$.

Якщо ознаки розпізнавання – **структурні**, то описом кожного класу має бути набір речень (ланцюжків із непохідних елементів з правилами з'єднання). Кожне з речень класу – характеристика структурних особливостей об'єктів цього класу. За приклад можна навести код Фрімена.

Для нечітких ознак класифікатор будується так, як це показано на рис. 2.8.

Для побудови системи класифікації на нечіткій логіці необхідно мати таку інформацію:

1) множину класів $\{b_1, b_2, \dots, b_m\}$, за якими відбувається класифікація;

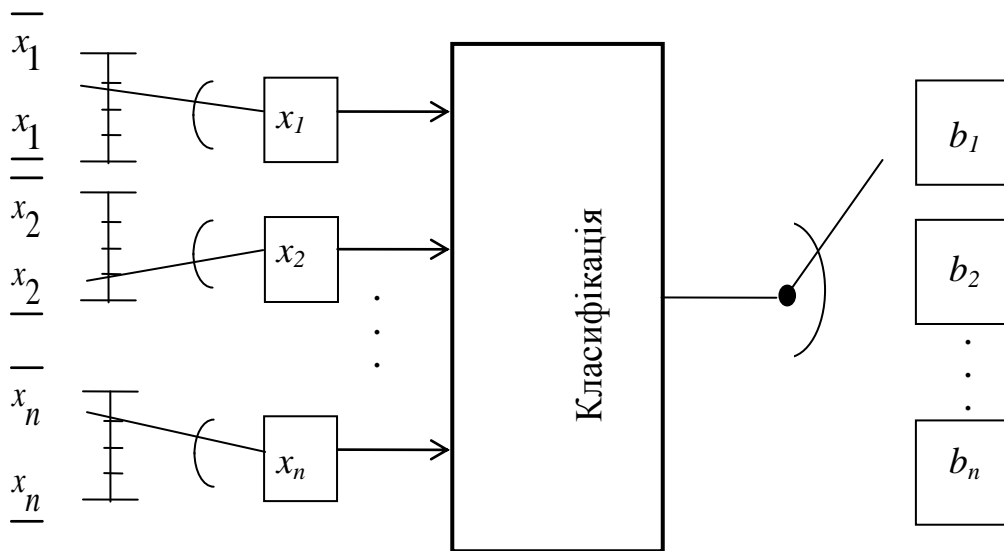


Рисунок 2.8 – Структурна схема нечіткого класифікатора

2) перелік властивостей (фрагментів зображення) об'єкта, що підлягає класифікації $\{x_1, x_2, \dots, x_n\}$;

3) нечітку базу знань – сукупність правил типу

$$\bigcup_{p=1, k_j} \bigcap_{i=1, n} (x_i = a_i^{jp} \text{ з вагою } c_{pj}) \rightarrow y = b_j, \quad j = \overline{1, m}, \quad (2.8)$$

за якими здійснюється класифікація;

4) перелік лінгвістичних термів a_i^{jp} для лінгвістичної оцінки властивості x_i ($i = \overline{1, n}$, $j = \overline{1, m}$, $p = \overline{1, k_j}$);

5) функції належностей $\mu_{a_i^{jp}}(x_i)$ нечітких термів a_i^{jp} ($i = \overline{1, n}$, $j = \overline{1, m}$, $p = \overline{1, k_j}$).

Задача 5. Вибір алгоритму класифікації, що забезпечує віднесення об'єкта розпізнавання або явища до відповідного класу

Безпосереднє вирішення задачі розпізнавання на основі використання словника ознак і алфавіту класів об'єктів або явищ фактично полягає в розподілі простору значень ознак розпізнавання на області D_1, D_2, \dots, D_n , що відповідають класам $\Omega_1, \Omega_2, \dots, \Omega_n$.

Зазначений розподіл має бути виконаний таким чином, щоб забезпечувалися мінімальні значення помилок віднесення класифікованих об'єктів або явищ до «чужих» класів.

Результатом такої операції є віднесення об'єкта, що має набір ознак x_1, x_2, \dots, x_n (точка в n -вимірному просторі), до класу Ω_i , якщо зазначена точка лежить у відповідному кластері простору ознак D_i .

Розбивку простору ознак можна подати як побудову дискримінантних функцій $F_i(x_1, x_2, \dots, x_n)$ між областями простору ознак D_i , що належать різним класам.

Потрібно, щоб згадані функції мали таку властивість – якщо об'єкт, що має вектор ознак \vec{x}_q , фактично відноситься до класу Ω_q , то значення дискримінантної функції $F_q(\vec{x}_q)$ має бути більшим, ніж її значення для деякого іншого класу Ω_g . Вираз вирішувальної межі між областями D_q і D_g , що відповідають класам Ω_q і Ω_g , знаходиться з виразу:

$$F_q(\vec{x}) - F_g(\vec{x}) = 0$$

Для двох класів розпізнавання розбивка двовимірного простору виглядає так (рис. 2.9).

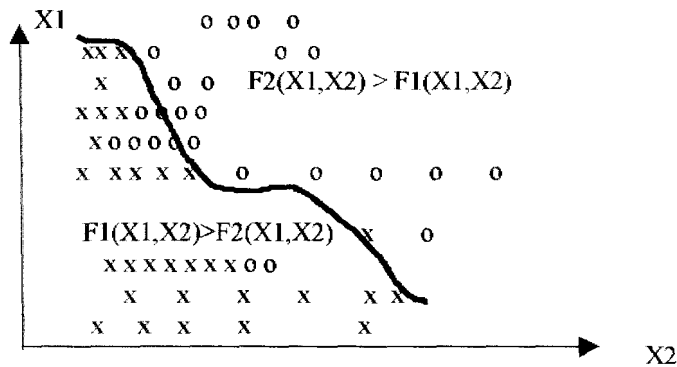


Рисунок 2. 9 – Вигляд дискримінантної функції для двох класів

Фізично розпізнавання ґрунтується на порівнянні значень тієї або іншої міри близькості об'єкта розпізнавання з кожним класом. Водночас, якщо значення обраної міри близькості L цього об'єкта ω із яким-небудь класом ω_g досягає екстремума щодо її значень за іншими класами, тобто

$$L(\omega, \Omega_g) = \underset{i}{extr} L(\omega, \Omega_i), \quad (2.9)$$

то приймається рішення про належність цього об'єкта класу Ω_g , тобто $\omega \in \Omega_g$.

Якщо міра близькості не має екстремума, то ми знаходимось на межі, де не можемо віддати перевагу жодному з класів.

У алгоритмах розпізнавання, що використовують детерміновані ознаки як міри близькості, використовується середньоквадратична (Евклідова) відстань між цим об'єктом ω і сукупністю об'єктів $(\omega_1, \omega_2, \dots, \omega_k)$, які належать деякому класу. Відстань від зображення об'єкта ω до класу Ω_g записується, як:

$$L(\omega, \Omega_g) = \sqrt{\left(\frac{1}{k_g} \cdot \sum_{g=1}^{k_g} d^2(\omega, \omega_g) \right)} \quad (2.10)$$

де k_g – кількість об'єктів, що є представниками Ω_g -го класу.

В цьому випадку як відстані між об'єктами $d(\omega, \omega_g)$ потрібно використовувати такі, які відповідають природі об'єктів, що розпізнаються.

В термінах координат (ознак) відстань має такий вигляд:

$$d(\omega, \omega_g) = \sqrt{\sum_{l=1}^L (X_{\omega_l} - X_{\omega_g})^2}, \quad (2.11)$$

де L – розмірність ознакового простору.

У алгоритмі розпізнавання, що використовує детерміновані ознаки, можна враховувати і їх ваги V_j (встановлювати ступінь довіри або важливості). Тоді розглянута середньоквадратична відстань набуває такого вигляду:

$$L(\omega, \Omega_g) = \sqrt{\frac{1}{k_g} \sum_{j=1}^{k_g} \sum_{k=1}^N V_j^2 (X_{\omega} - X_{\omega_g})}. \quad (2.12)$$

Якщо об'єктами розпізнавання є сигнали, то як відстані часто застосовують кутову відстань:

$$d(\omega, \omega_g) = \arccos \frac{\bar{X}_{\omega} \cdot \bar{X}_{\omega_g}}{\|\bar{X}_{\omega}\| \cdot \|\bar{X}_{\omega_g}\|}, \quad (2.13)$$

де $\|\bar{X}_{\omega}\|$ і $\|\bar{X}_{\omega_g}\|$ – норми відповідних векторів.

У алгоритмах розпізнавання, що використовують імовірнісні ознаки, як міра близькості використовується середній ризик, пов'язаний із ймовірністю належності об'єкта до класу ω_i , де i – номер класу. ($i = 1, 2, \dots, k$), і платіжною матрицею.

Для алгоритмів, основаних на логічних ознаках, використовують відстань Танімото або значення істинності логічної функції. Покажемо це на прикладі, розглянутому вище для логічних ознак захворювань (простой застуди й ангіни).

Якщо є значення ознак A, B, C , достатньо підставити їх у булеві співвідношення між класами й ознаками, щоб відразу одержати результат як істинність або хибність булевої функції опису того або іншого класу.

Дійсно, нехай ознаки набули таких значень:

- підвищена температура: $A=1$;
- нежить: $B=0$;
- нариви в горлі: $C=1$.

Тоді підстановка їх у булеві співвідношення дасть такий результат:

$$F(\Omega_1) = ABC + \bar{A}\bar{B}\bar{C} + \bar{A}BC = 0$$

$$F(\Omega_2) = A\bar{B}C + \bar{A}\bar{B}C = 1$$

Тобто, істиною є друге співвідношення, що відповідає розпізнаванню ангіни як діагностовного класу, з двох захворювань.

Для алгоритмів, оснований на структурних (лінгвістичних) ознаках, поняття «міри близькості» більш специфічно.

Задача класифікації за умови використання нечітких ознак може бути поставлена таким чином: об'єкт X^* з характеристиками $\{x_1^*, x_2^*, \dots, x_n^*\}$ відноситься до класу $b^* \in \{b_1, b_2, \dots, b_m\}$, для якого

$$\mu_{b^*}(X^*) = \max_{j=1, m}(\mu_{b_j}(X^*)), \quad (2.14)$$

де

$$\mu_{b_j}(X^*) = \bigvee_{p=1}^{k_j} [c_{pj} \cdot \bigwedge_{i=1}^n (\mu_{a_i^{jp}}(x_i^*))], \quad j = \overline{1, m}, \quad (2.15)$$

$\bigvee (\bigwedge)$ – операція максимуму (мінімуму).

Алгоритм нечіткого логічного виведення такий:

1. З бази знань отримати систему нечітких логічних рівнянь, які мають вигляд (2.2);

2. Визначити ступінь належності значення вхідної змінної X^* термам a_i^{jp} , ($i = \overline{1, n}$, $j = \overline{1, m}$, $p = \overline{1, k_j}$), k_j – число диз'юнкції (або) в j -му логічному висловлюванні;

3. Визначити ступінь належності вхідного вектора X^* термам b_j ($j = \overline{1, m}$) шляхом підстановки значень, знайдених на 2-му кроці в систему нечітких логічних рівнянь;

4. Визначити елемент d^* , для якого справедливий вираз (2.14).

З урахуванням того, що кожен клас описується сукупністю пропозицій, що характеризують структурні особливості об'єктів відповідних класів, розпізнавання невідомого об'єкта здійснюється ідентифікацією пропозиції, що описує цей об'єкт, з одним із пропозицій у складі опису якогось класу.

Водночас ідентифікація може розуміти найбільшу подібність пропозиції, що описує об'єкт розпізнавання із пропозиціями з наборів опису кожного класу.

Розглянувши задачу № 5, ми фактично завершили розгляд кола задач створення СР. У той самий час вже відзначалося, що створення СР здійснюється послідовними наближеннями в міру одержання додаткової інформації. У цьому ряду послідовних наближень головну роль відіграють ознаки розпізнавання. Від ефективності їхнього набору залежить ефективність системи загалом. У процесі удосконалювання системи зазначений набір поповнюється, неефективні ознаки виключаються. Тому однією з задач створення СР має бути і задача переходу від апріорного словника ознак до робочого. Те саме стосується й апріорного алфавіту класів.

Задача 6. Розробка ефективної системи розпізнавання

Ця задача на рівні розробки, що пройшла етапи вирішення задач 1–5, принаймні вже можливо поставлена, тому що внаслідок виконання попередніх задач створено систему розпізнавання першого наближення (апріорний алфавіт класів і апріорний словник ознак, обраний алгоритм розпізнавання).

Суть поставленої задачі – розробка такого (робочого) алфавіту класів, такого (робочого) словника ознак і такого алгоритму класифікації, які б забезпечили максимальне значення показника ефективності розпізнавання. Тобто, з апріорного словника ми маємо вибрати ознаки, що дозволяють за всіх існуючих обмежень на їх одержання (вимірювання) доставити максимум імовірності правильної класифікації об'єктів (явищ) і (або) мінімальні імовірності помилкових класифікацій, що утворюються системою. Такий вибір не може не припускати оцінення зазначених показників до того, як створена система. Зазначена суть задачі змушує знову звернути увагу на можливість одержання оцінки ефективності системи розпізнавання шляхом її моделювання.

Що ж стосується прийомів, які забезпечують відбір у процесі оптимізації систем розпізнавання, то вони є також предметом окремого розгляду.

2.4 Класифікація систем розпізнавання образів

Можлива класифікація систем розпізнавання об'єктів та явищ, що базується на різних властивостях інформації, яка використовується в процесі розпізнавання, показана на рис. 2.10.

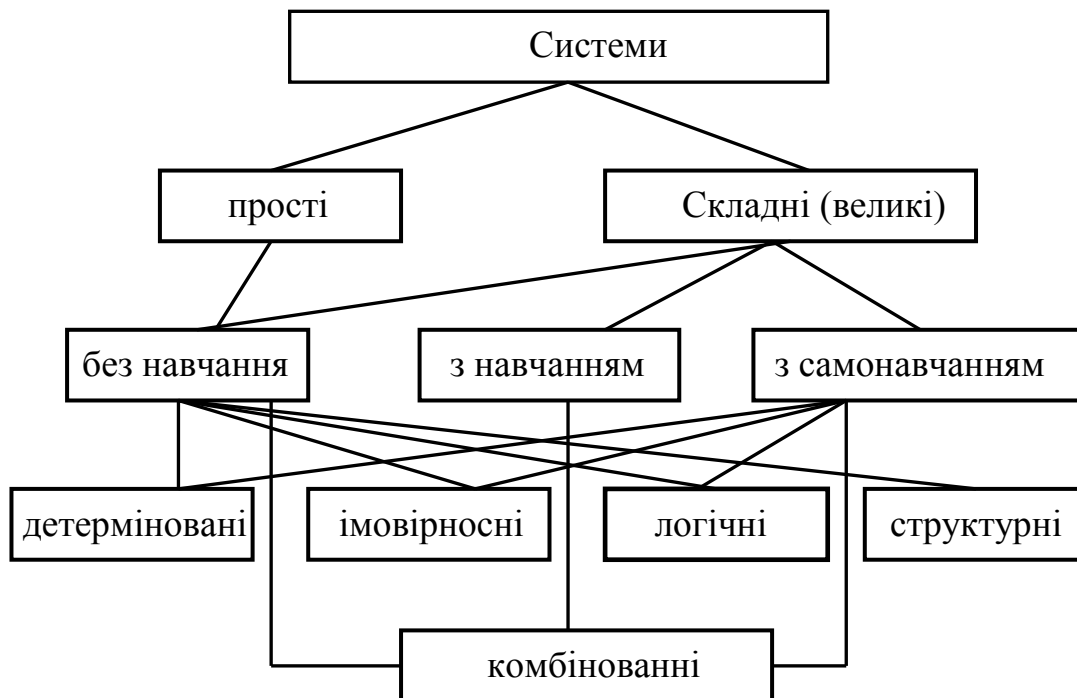


Рисунок 2.10 – Схема класифікації систем розпізнавання образів

Класифікація – це розподіл предметів, явищ по класах, відділах, розрядах залежно від їх загальних властивостей.

У основі класифікації лежать певні принципи. Для класифікації СР будемо використовувати такі принципи:

1. Однорідність інформації для опису об'єктів розпізнавання або явищ.
2. Засіб одержання апостеріорної інформації.
3. Кількість початкової апріорної інформації.
4. Характер інформації про ознаки розпізнавання.

Розглянемо 1-й принцип (Однорідність інформації).

Тут під однорідністю варто розуміти – різноманітну або єдину фізичну природу інформації (ознак). За цим принципом СР поділяються на прості і складні.

Прості системи розпізнавання. Прості СР характеризуються єдиною фізичною природою ознак. До них відносять, наприклад, зчитувальні автоматичні пристрої розпізнавання, в яких ознаки робочого словника являють собою лише ті чи інші лінійні розміри об'єктів, що розпізнаються; автомати для розміну монет, де в як ознаки береться їх вага; автоматичні прилади, які призначені для обробки деталей, в яких як ознаки, що застосовуються для опису класів бракованих і небракованих деталей, використовуються або деякі лінійні розміри, або вага і т. д.

Складні системи розпізнавання. До них відносять, наприклад, системи медичної діагностики, в яких як ознаки (симптоми) можуть використовуватись дані аналізу крові і кардіограма, температура і динаміка тиску крові тощо; системи, які призначені для розпізнавання зразків геологічної розвідки, в яких як ознаки беруться різні фізичні та хімічні властивості, або зразки воєнної техніки можливого ворога тощо.

Другий принцип класифікації (Спосіб одержання апостеріорної інформації). Якщо як принципи класифікації використовувати спосіб отримання апостеріорної інформації, то складні системи можна розділити на однорівневі та багаторівневі.

Однорівневі складні системи. В цих системах апостеріорна інформація про ознаки об'єктів, що розпізнаються, визначається прямими вимірами на основі обробки результатів дослідів (рис. 2.11, де АІ – апріорна інформація; БАР – блок алгоритмів розпізнавання; БКРЗ – блок керування роботи засобів).

Багаторівневі складні системи. В цих системах апостеріорна інформація про ознаки визначається на основі другорядних вимірів. Для таких вимірів використовуються спеціальні локальні системи розпізнавання (рис. 2.12, де АІ – апріорна інформація; БАР – блок алгоритмів розпізнавання; БКРЗ – блок керування роботи засобів).

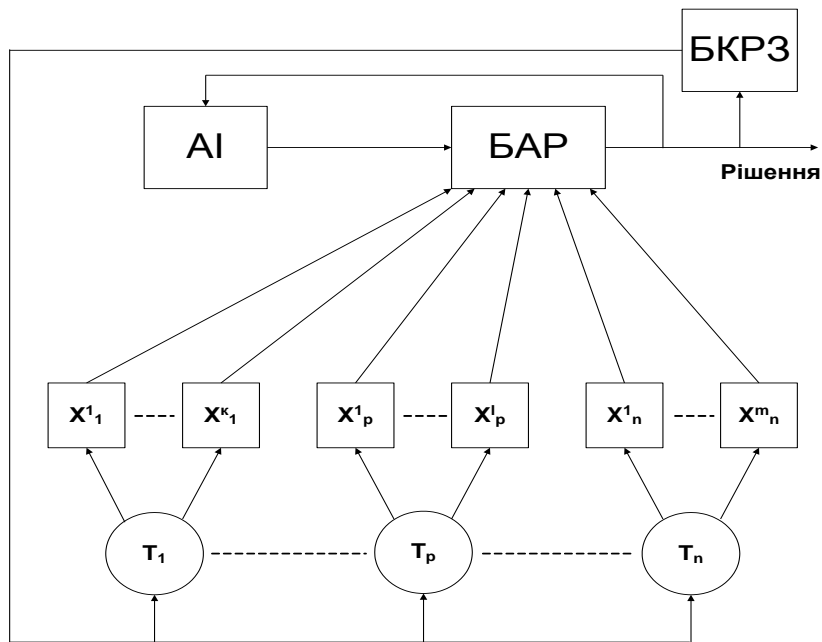


Рисунок 2.11 – АІ – апіорна інформація; БАР – блок алгоритмів розпізнавання; БКРЗ – блок керування роботою засобів; Т - технічні вимірювальні засоби

В однорівневих системах (рис. 2.11) за даними технічних засобів $T_1, \dots, T_p, \dots, T_n$ на основі обробки отриманих реалізацій безпосередньо знаходять ознаки $X_1^1, \dots, X_1^k; X_p^1, \dots, X_p^l; X_n^1, \dots, X_n^m$ невідомих об'єктів або явищ, які використовуються для їх розпізнавання.

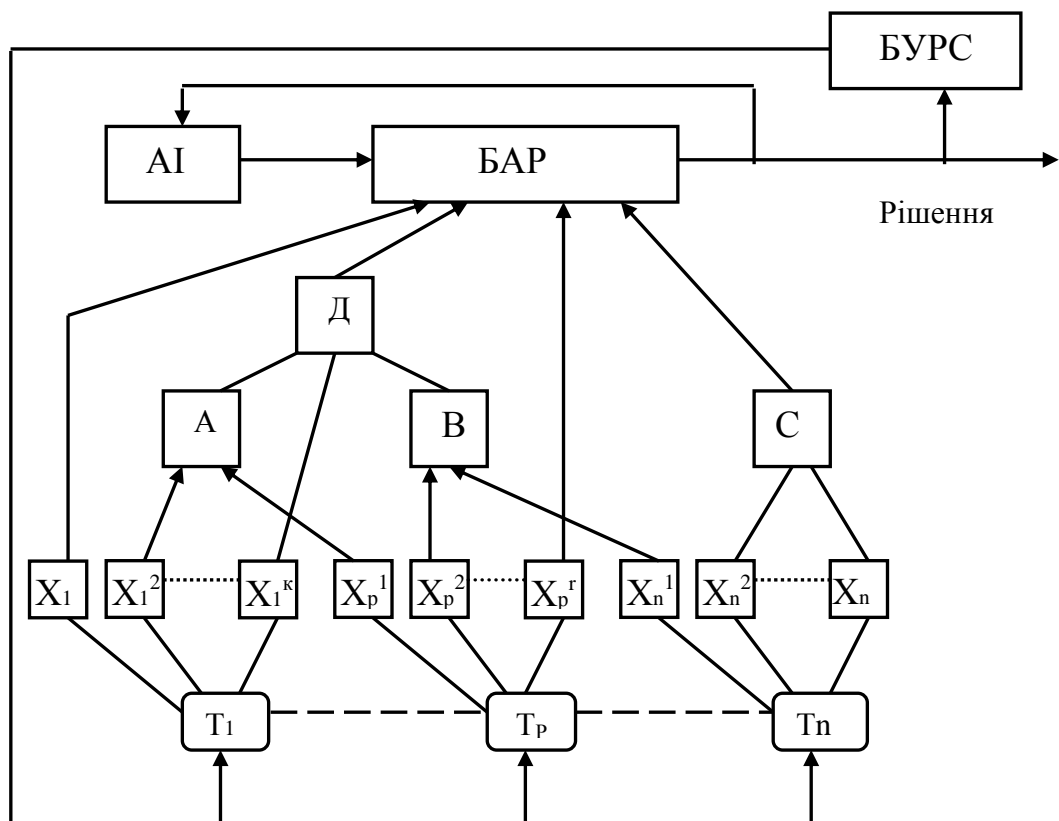


Рисунок 2.12 – Багаторівнева система розпізнавання

В багаторівневих системах (рис. 2.12) за даними технічних засобів $T_1, \dots, T_p, \dots, T_n$ визначаються ознаки $X_1^1, \dots, X_1^K; X_p^1, \dots, X_p^l; X_n^1, \dots, X_n^m$ (назвемо їх первинними), які поділяються на такі групи.

Група 1. До неї відносять ознаки, що використовуються в локальних пристроях першого (нижнього) рівня (назвемо ці ознаки ознаками першого рівня) для визначення ознак другого рівня. На рис. 2.12 такими ознаками є $X_1^2, X_p^1, X_p^2, X_n^1, X_n^2, X_n^m$. На основі цих ознак розпізнавальні пристрої першого рівня А, В, С визначають ознаки другого рівня X_2^A, X_2^B, X_2^C .

Група 2. До неї відносяться ознаки, що безпосередньо використовуються в обчислювальних пристроях другого рівня для визначення ознак третього рівня. На рис. 2.12 такою ознакою є X_1^k , що використовується одночасно з ознаками другого рівня X_2^A, X_2^B в розпізнавальному пристрої другого рівня D для визначення ознаки третього рівня X_3^D .

Група 3. До неї відносять ознаки, що використовуються в розпізнавальних пристроях третього рівня для визначення ознак четвертого рівня.

До останньої групи відносять ознаки, що безпосередньо використовуються в процесі розпізнавання невідомих об'єктів, тобто ознаки, що входять в робочий словник ознак системи розпізнавання. На рис. 2.12 такими ознаками є X_1^1, X_p^l (назвемо ці ознаки ознаками верхнього рівня).

Таким чином, в однорівневих системах розпізнавання інформація про ознаки формується шляхом прямих вимірювань безпосередньо на основі обробки даних дослідів. В багаторівневих системах інформація про зазначені ознаки формується на основі другорядних вимірів як результат функціонування допоміжних розпізнавальних пристроїв. На входи таких пристроїв надходить попередньо оброблена вимірювальна інформація, а на виходах утворюється або безпосередньо інформація про ознаки об'єктів чи явищ, або проміжна інформація, що використовується розпізнавальними пристроями наступних рівнів для формування інформації про ознаки об'єктів розпізнавання.

Третій принцип класифікації (Кількість початкової апріорної інформації). За цим принципом класифікації системи розпізнавання (прості та складні) можна розділити на системи без навчання, системи, що навчаються, та самонавчальні системи.

Багаторівневі системи розділяються на вказані класи, тому що кожна з їх локальних систем може, також, являти собою систему без навчання; систему, що навчається, та самонавчальну.

Система без навчання. В цих системах апріорної інформації достатньо для того, щоб визначити апріорний словник класів, побудувати апріорний словник ознак та на основі безпосередньої обробки початкових даних провести опис кожного класу мовою цих ознак, тобто в першому

наближенні достатньо визначити вирішувальні функції та вирішувальні правила.

Будемо вважати, що для побудови цього класу систем необхідно мати повну початкову апріорну інформацію. На рис. 2.13 (де ТЗ – технічні засоби, АІ – апріорна інформація, БАР – блок алгоритмів розпізнавання) подано структурну схему системи без навчання.

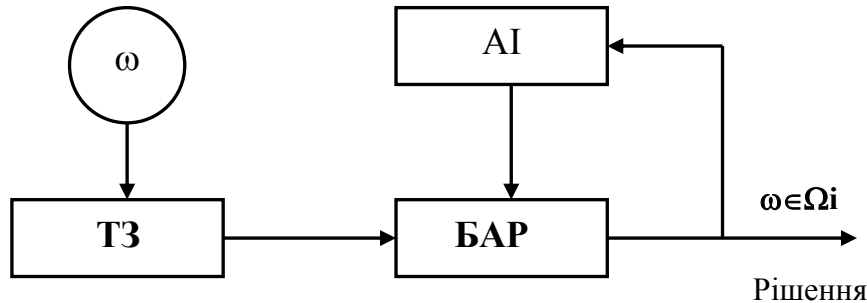


Рисунок 2.13 – Система розпізнавання без навчання

Система, що навчається. В цих системах початкової апріорної інформації достатньо для того, щоб визначити апріорний алфавіт класів та побудувати апріорний словник ознак, але не достатньо для опису класів мовою ознак. Початкову інформацію, необхідну для побудови такої системи, подано у вигляді конкретних об’єктів, що належать різним класам:

$$\left. \begin{array}{l} (\omega_1, \omega_2, \dots, \omega_r) \in \Omega_{1i} ; \\ \dots \dots \dots \\ (\omega_{r+1}, \omega_{r+2}, \omega_q) \in \Omega_{2i} ; \\ (\omega_{g+1}, \omega_{g+2}, \dots, \omega_l) \in \Omega_m \end{array} \right\} \quad (2.16)$$

Об’єкти $\omega_1, \dots, \omega_l$ є елементами навчальної вибірки. Мета процедури навчання – визначення вирішувальних функцій $F_i(x_1, \dots, x_L), l = 1, \dots, L$, шляхом багаторазового показу системі розпізнавання різних об’єктів з вказанням на класи, до яких ці об’єкти належать.

Дані системи на стадії розробки працюють з «вчителем». Ця робота полягає в тому, що «вчитель» багаторазово надає системі об’єкти всіх визначених класів та вказує, до яких класів вони належать. Потім «вчитель» починає «екзаменувати» систему розпізнавання, корегуючи її відповіді до тих пір, доки кількість помилок в середньому не досягне бажаного рівня.

Самонавчальна система. В цих системах початкової апріорної інформації достатньо лише для визначення словника ознак x_1, \dots, x_N , але недостатньо для проведення класифікації об’єктів. На стадії формування системи їй надають початкову сукупність об’єктів $\omega_1, \dots, \omega_l$, що задані

значеннями своїх ознак для $\omega_l = (x_1^l, \dots, x_N^l)$. Але через обмежений обсяг початкової інформації система в цьому випадку не отримує вказівок про те, до якого класу належать об'єкти початкової сукупності.

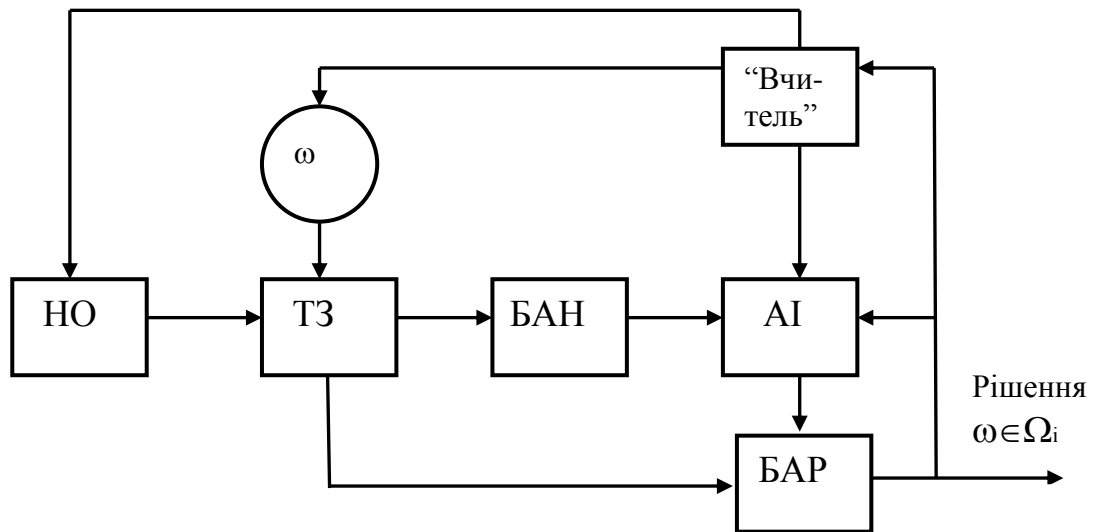


Рисунок 2.14 – Система розпізнавання з навчанням

На рис. 2.14 використано такі позначення: НО – об'єкти, що навчають, ТЗ – технічні засоби, БАН – блок алгоритмів навчання, АІ – апіорна інформація, БАР – блок алгоритмів розпізнавання.

Всі вказівки замінюються набором правил, згідно з якими на стадії самонавчання система розпізнавання сама розробляє класифікацію, яка може відрізнитись від природної, і в подальшому її дотримується.

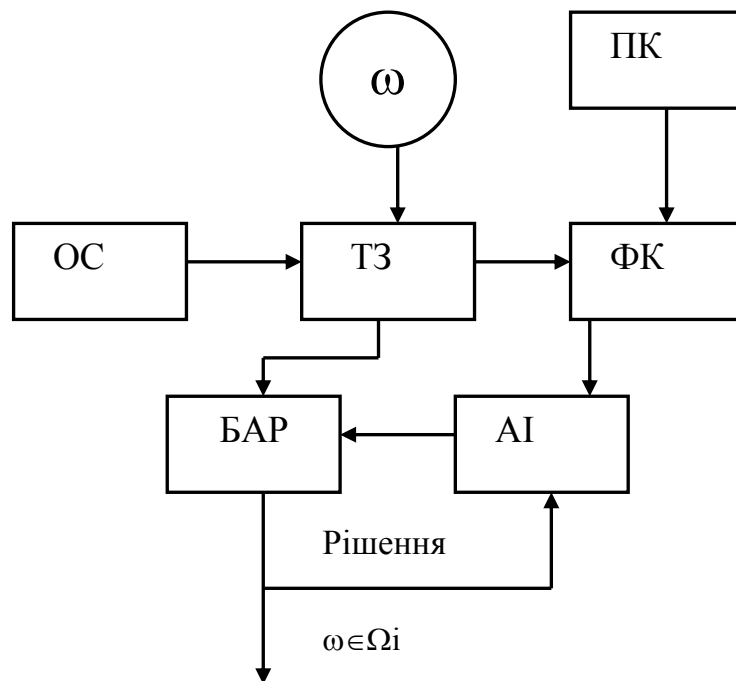


Рисунок 2.15 – Самонавчальна система розпізнавання

На рис. 2.15 використано такі позначення: ОС – об'єкти для самонавчання, ТЗ – технічні засоби, БАР – блок алгоритмів розпізнавання, АІ – апріорна інформація, ПК – правила класифікації, ФК – формування класів.

Термін «повна початкова апріорна інформація» характеризує не абсолютну, а відносну кількість необхідної інформації. Він вказує на те, що в системах без навчання на рівні з іншими умовами, кількість початкової інформації більша, ніж в розпізнавальних системах, що навчаються.

Системи, що навчаються або самонавчальні системи отримують недостатню апріорну інформацію в процесі навчання або самонавчання. Більш того, мета цього – виробити таку кількість інформації, яка необхідна для функціонування системи розпізнавання.

Створення систем, що навчаються, та самонавчальних систем розпізнавання обумовлено тим, що на практиці дуже часто зустрічаються ситуації, коли відсутня повна початкова апріорна інформація.

Під час побудови систем розпізнавання доцільно використовувати принцип зворотного зв'язку. Це питання полягає в тому, що результати вирішення задачі розпізнавання невідомих об'єктів потрібно використовувати для уточнення апріорного опису класів. На рис. 2.11 – 2.15 видно зворотні зв'язки в розглянутих типах систем розпізнавання.

Четвертий принцип класифікації (Характер інформації про ознаки розпізнавання). Якщо як принцип класифікації використовувати характер інформації про ознаки об'єктів, що розпізнаються, які розділили на детерміновані, імовірнісні, логічні та структурні, то залежно від того, мовою яких ознак проводиться опис цих об'єктів, по-іншому – залежно від того, який алгоритм розпізнавання реалізовано, системи розпізнавання можуть бути розділені на детерміновані, логічні, імовірнісні, структурні та комбіновані.

Детерміновані системи. В цих системах для побудови алгоритмів розпізнавання використовуються «геометричні» міри наближення, основані на вимірюванні відстаней між об'єктом та еталонами класів. В загальному випадку застосування детермінованих методів розпізнавання передбачає наявність координат еталонів класів в просторі ознак або координат об'єктів, що належать до відповідних класів.

Імовірнісні системи. В цих системах для побудови алгоритмів розпізнавання використовуються імовірнісні методи розпізнавання, які ґрунтуються на теорії статистичних рішень. В загальному випадку застосування імовірнісних методів розпізнавання передбачає наявність імовірнісних залежностей між ознаками об'єктів, що розпізнаються, та класами, до яких ці об'єкти відносяться.

Логічні системи. В цих системах для побудови алгоритмів розпізнавання використовуються логічні методи розпізнавання, що базуються на дискретному аналізі та основаних на ньому висловлюваннях. В загальному випадку застосування логічних методів розпізнавання передбачає наявність логічних зв'язків, виражених через систему булевих рівнянь, в яких змінні – логічні ознаки об'єктів, що розпізнаються, а невідомі величини – класи, до яких ці об'єкти відносяться.

Структурні (лінгвістичні) системи. В цих системах для побудови алгоритмів розпізнавання використовуються спеціальні граматики, з яких породжують мови, що складаються з речень, кожне речення описує об'єкти, які належать певному класу.

Застосування структурних методів розпізнавання потребує наявності сукупностей речень, що описують всю множину об'єктів, які належать всім класам алфавіту класів системи розпізнавання. Водночас множина речень має бути розділена на підмножини за кількістю класів системи. Елементами підмножини є речення, що описують об'єкти, які належать цій підмножині (класу). Таким чином, апріорними описами класів є сукупності речень, кожне з яких відповідає конкретному об'єкту, що належить цьому класу.

Комбіновані системи. В цих системах для побудови алгоритмів розпізнавання використовується спеціально розроблений метод обчислення оцінок. Такі алгоритми розпізнавання називають алгоритмами обчислення оцінок (АОО). Їх застосування потребує наявності таблиць, де знаходяться об'єкти, які належать відповідним класам, а також значення ознак, якими характеризуються ці об'єкти.

Ознаки можуть бути детермінованими, логічними, імовірнісними, структурними та нечіткими.

Контрольні питання та завдання

1. Дайте означення понять «розпізнавання» і «образ».
2. Наведіть приклади задач розпізнавання.
3. Зарисуйте структурну схему системи розпізнавання образів і опишіть призначення її складових частин.
4. Наведіть змістовне означення основних понять теорії розпізнавання образів.
5. Що називається ознакою?
6. Як математично описуються об'єкти, що розпізнаються?
7. Дайте змістовне означення таких понять: клас, кластер, класифікація, кластеризація.
8. Наведіть геометричну інтерпретацію впливу перетворення простору вимірювань в простір ознак.
9. Назвіть основні задачі теорії розпізнавання образів.
10. Опишіть задачу визначення апріорного словника ознак. Назвіть основні типи ознак, що використовуються в розпізнаванні.
11. Які властивості інформації покладено в основу принципу класифікації систем розпізнавання?
12. Назвіть основні типи систем розпізнавання образів за ознакою однорідності інформації.
13. Наведіть приклади простих і складних систем розпізнавання.
14. На які типи поділяються системи розпізнавання за принципом способу одержання інформації?
15. Чим відрізняються одна від одної системи розпізнавання без навчання, з навчанням і самонавчальні?
16. Який принцип класифікації лежить в основі поділу систем розпізнавання на детерміновані, імовірнісні, логічні, структурні, комбіновані?

РОЗДІЛ 3 КЛАСИФІКАЦІЯ ОБРАЗІВ

3.1 Методи класифікації

В попередньому розділі було вказано, що в розпізнаванні образів розрізняють такі основні задачі:

1. Визначення повного апріорного набору ознак (первинних параметрів), що описують об'єкти розпізнавання.
2. Розподіл множини об'єктів за їх описами на класи (систему підмножин, що не перетинаються) – кластеризація.
3. Виділення інформативних ознак з апріорного словника ознак і зменшення розмірності векторів зображень.
4. Розробка способу подання образів вибраними ознаками.
5. Побудова правил віднесення сприйнятих образів до одного з класів – класифікація.
6. Розробка ефективних алгоритмів розпізнавання.

Центральною задачею розпізнавання образів є задача класифікації. Вона може бути вирішена за допомогою евристичних або математичних методів. В основу евристичного підходу покладені інтуїція і досвід людини; в ньому використовуються принципи спільності властивостей і перерахування членів класу.

Приклад 3.1. Розробити алгоритм розподілу на класи такої множини літер: П, Л, А, Д.

Розв'язування. За ознакою нахилу лівої риски ці літери можна розбити на два класи: Ω_1 і Ω_2 . В класі Ω_1 нараховуємо одну літеру П, тобто $\Omega_1 = \{\text{П}\}$, а в класі Ω_2 – літери Л, А і Д, тобто $\Omega_2 = \{\text{Л}, \text{А}, \text{Д}\}$. За наявності горизонтальної лінії клас Ω_2 можна поділити на два: $\Omega_3 = \{\text{Л}\}$ і $\Omega_4 = \{\text{А}, \text{Д}\}$. За місцеположенням горизонтальної риски клас Ω_4 можна розбити на два класи: $\Omega_5 = \{\text{А}\}$, $\Omega_6 = \{\text{Д}\}$.

В цьому прикладі ми на основі визначених для цієї множини спільних ознак визначаємо різні класи літер шляхом їх перерахування. Просту процедуру класифікації цих літер зображено на рис. 3.1.

З наведеного прикладу зразу ж можна зробити висновок про недоліки евристичного підходу – він має вузьке застосування, тому що орієнтований на конкретну задачу розпізнавання і не дає загальних принципів синтезу систем розпізнавання. Наприклад, якщо до множини вказаних літер додати літеру Н, то наведена процедура буде працювати неправильно, оскільки вона не аналізує наявність верхньої горизонтальної риски.

Тому під час розгляду задачі класифікації будемо опиратися на математичні методи, в основу яких покладено правила класифікації, що

формулюються і виводяться в рамках математичного формалізму з допомогою принципів спільності властивостей і кластеризації.

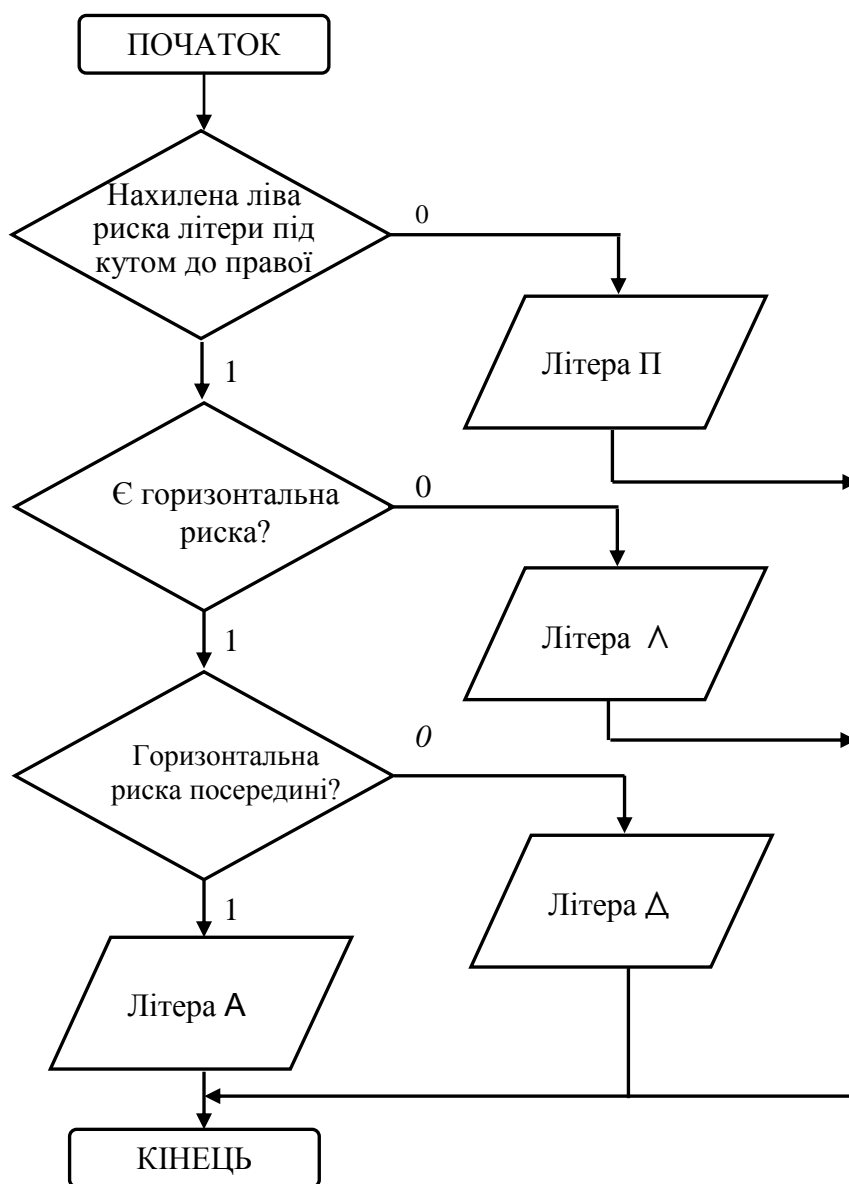


Рисунок 3.1 – Схема простої евристичної процедури класифікації

Цим математичний підхід відрізняється від евристичного, в якому правила рішень тісно пов'язані з характером вирішуваної задачі.

Математичні методи можна поділити на такі типи:

- детерміновані;
- статистичні (імовірнісні);
- структурні (лінгвістичні);
- логічні;
- нечіткі;
- на базі нейронних мереж.

Останній клас математичних методів виділений умовно, оскільки неважко показати, що нейронну мережу можна розглядати як різновид нелінійного класифікатора 1-го або 2-го типу.

В цьому розділі будуть розглянуто основні положення детермінованого методу класифікації.

Повернемося до розділу 2 і розглянемо рис. 2.4. З нього видно, що два класи Ω_1 і Ω_2 зручно розмежувати прямою. Її рівняння можна записати у вигляді:

$$d(\vec{x}) = w_0 + w_1x_1 + w_2x_2 = 0, \quad (3.1)$$

де w_0, w_1, w_2 – вагові параметри,

x_1, x_2 – ознакові координати вектора зображення в двовимірному просторі.

З рис. 2.4 зрозуміло, що підстановка в $d(\vec{x})$ будь-якого вектора \vec{x} , який належить класу Ω_1 , дасть додатне значення, а будь-якого вектора з класу Ω_2 – від’ємне значення, тобто:

$$\forall \vec{x} \in \Omega_1 \quad d(\vec{x}) > 0 \quad \text{і} \quad \forall \vec{x} \in \Omega_2 \quad d(\vec{x}) < 0. \quad (3.2)$$

Таким чином, функцію $d(\vec{x})$ можна використовувати за вирішувальну (дискримінантну), оскільки за її знаком можна вирішити, до якого класу відноситься об’єкт з відомим ознаковим описом \vec{x} . Структурну схему лінійного класифікатора на два класи, що відповідає лінійній функції (3.1), зображено на рис. 3.2.

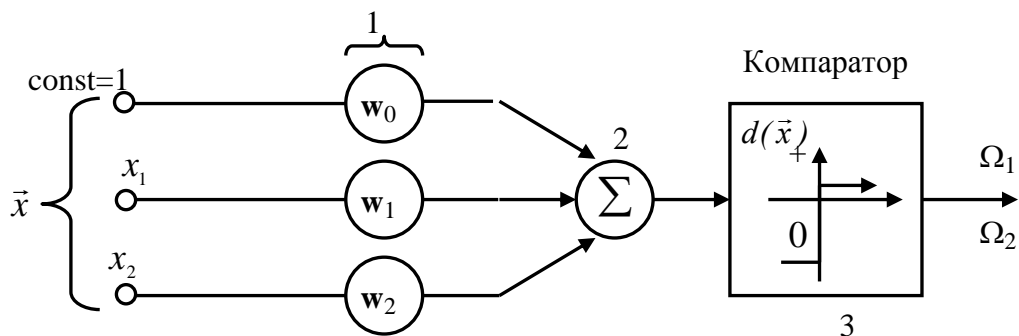


Рисунок 3.2 – Схема лінійного класифікатора на два класи: 1 – вагові коефіцієнти, 2 – суматор, 3 – компаратор нуля

Приклад 3.2. Лінійна вирішувальна функція задана у вигляді площини з рівнянням:

$$d(\vec{x}) = 2x_1 + 3x_2 - x_3 + 5,$$

що розділяє два класи Ω_1 і Ω_2 . Визначити, до якого класу відноситься вектор $\vec{x} = (0.5; 1; 3)$.

Розв'язання: підставляємо координати вектора \vec{x} в рівняння гіперплощини:

$$d(\vec{x}) = 2 * 0.5 + 3 * 1 - 1 * 3 + 5 = +6.$$

Функція $d(\vec{x})$ – додатна, тому робимо висновок, що $\vec{x} \in \Omega_1$.

В подальшому буде показано, що цей метод класифікації неважко узагальнити на довільну кількість класів, описаних в просторі ознак довільної розмірності.

Ефективність використання вирішувальної функції для класифікації визначається двома факторами – виглядом функції $d(\vec{x})$ і практичною можливістю визначення її коефіцієнтів.

Перший фактор пов'язаний з геометричними властивостями простору образів. Наприклад, з рис. 3.3 видно, що для розподілу заданої сукупності об'єктів на класи потрібні границі, значно складніші, ніж в розглянутому на рис. 2.4 випадку лінійної розмежованості.

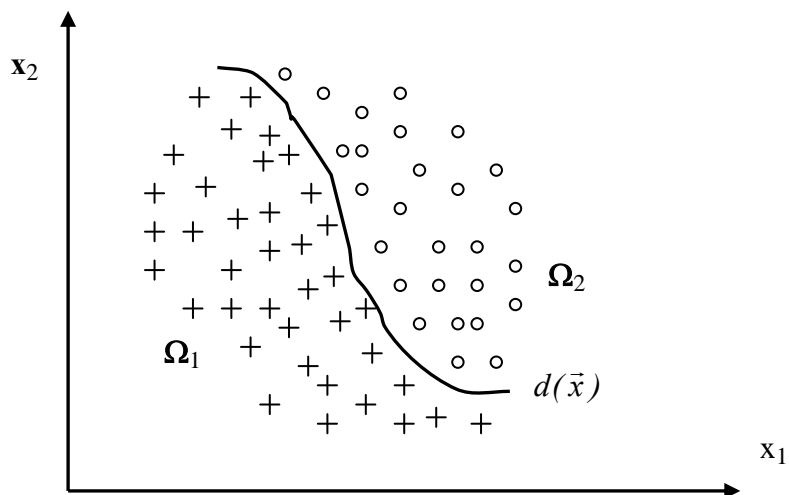


Рисунок 3.3 – Випадок нелінійної вирішувальної функції

Крім того, за розмірності простору ознак більше трьох зорва уява не дозволяє нам взагалі визначити вигляд вирішувальної функції з розміщення точок в просторі. Тому в більшості випадків для вибору вигляду вирішувальної функції звертаються до аналітичних процедур.

Після вибору її вигляду переходять до визначення вагових коефіцієнтів. Для цього використовують адаптивні процедури навчання, застосовані до заданої вибірки об'єктів.

3.2 Лінійні вирішувальні функції та їх властивості

Узагальнимо простий варіант лінійної вирішувальної функції, зображеної на рис. 2.4, на n -вимірний випадок. Загальний вигляд лінійної вирішувальної функції задається формулою

$$d(\vec{x}) = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n = w_0 + \vec{w}_a^T \vec{x}, \quad (3.3)$$

де вектор $\vec{w}_a = (w_1, w_2, \dots, w_n)^T$ називається ваговим або параметричним, а вектор $\vec{x} = (x_1, x_2, \dots, x_n)^T$ – вектор об'єкта.

Співвідношення (3.3) можна також подати у вигляді:

$$d(\vec{x}) = \vec{w}^T \vec{x}, \quad (3.4)$$

де $\vec{w} = (w_0, w_1, w_2, \dots, w_n)^T$, і $\vec{x} = (1, x_1, x_2, \dots, x_n)^T$ – поповнені вектори ваг та зображень, відповідно.

Розглянемо основні геометричні властивості гіперплощин (3.3), які є лінійними вирішувальними функціями в n-вимірному ознаковому просторі. У випадку двох класів рівняння розподільної гіперплощини буде мати вигляд:

$$d(\vec{x}) = w_0 + \vec{w}_a^T \cdot \vec{x} = 0. \quad (3.5)$$

Зобразимо схематично гіперплощину так, як це показано на рис. 3.4.

Нехай \vec{u} – одинична додатно орієнтована нормаль, тобто одиничний вектор, перпендикулярний до гіперплощини в деякій точці \vec{p} і напрямлений в додатну сторону гіперплощини. З геометричного погляду рівняння гіперплощини можна записати у вигляді

$$\vec{u}^T (\vec{x} - \vec{p}) = 0, \quad (3.6)$$

або

$$\vec{u}^T \vec{x} = \vec{u}^T \vec{p}. \quad (3.7)$$

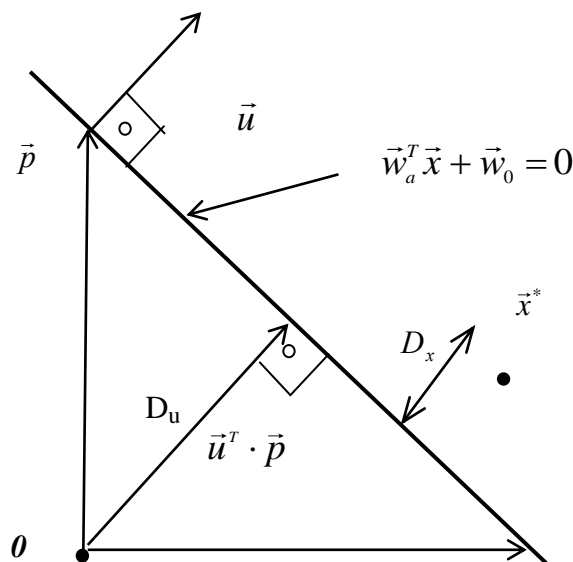


Рисунок 3.4 – Ілюстрація деяких геометричних властивостей гіперплощин

Поділимо рівняння (3.5) на довжину вагового вектора $\|\vec{w}_a^T\| = \sqrt{w_1^2 + w_2^2 + \dots + w_n^2}$, отримаємо:

$$\frac{\vec{w}_a^T \vec{x}}{\|\vec{w}_a^T\|} = -\frac{w_0}{\|\vec{w}_a^T\|}. \quad (3.8)$$

Порівняємо (3.8) і (3.7), звідки запишемо

$$\vec{u} = \frac{\vec{w}_a}{\|\vec{w}_a\|}. \quad (3.9)$$

З цього виразу видно, що за рівності 0 будь-якої координати вектора \vec{w}_a гіперплощина паралельна цій координаті.

$$\vec{u}^T \cdot \vec{p} = -\frac{w_0}{\|\vec{w}_a\|}. \quad (3.10)$$

З рисунка 3.4 видно, що добуток $\vec{u}^T \cdot \vec{p}$ характеризує відстань D_u від початку координат до гіперплощини і дорівнює чисельно:

$$D_u = \frac{|w_0|}{\|\vec{w}_a\|}. \quad (3.11)$$

З (3.11) витікає, що за $w_0 = 0$ гіперплощина проходить через початок координат. З рис. 3.4 також видно, що відстань D_x від гіперплощини до довільної точки \vec{x}^* ознакового простору визначається рівнянням:

$$D_x = \left| \vec{u}^T \vec{x}^* - \vec{u}^T \vec{p} \right| = \left| \frac{\vec{w}_a^T \cdot \vec{x}^* + w_0}{\|\vec{w}_a\|} \right|. \quad (3.12)$$

Ще однією з характеристик лінійної вирішувальної функції є кількість способів класифікації заданої множини об'єктів, які можна здійснити з її допомогою. Розглянемо для прикладу рис. 3.5, на якому наведено всі можливі розподіли на два класи чотирьох двовимірних зображень $\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4$.

Пряма 1, наприклад, розбиває всі зображення на два класи: $\Omega_1 = \{\vec{x}_1\}$, $\Omega_2 = \{\vec{x}_2, \vec{x}_3, \vec{x}_4\}$. Оскільки назви класів можна поміняти ($\Omega_1 = \{\vec{x}_2, \vec{x}_3, \vec{x}_4\}$, а $\Omega_2 = \{\vec{x}_1\}$), то пряма 1 визначає два можливих розподіли на два класи.

Те саме можна сказати про прямі 2, 3, 4, 5, 6 і 7. Тому загальна кількість розподілів на 2 класи (число дихотомій) в цьому випадку дорівнює 14. Як відомо, в загальному випадку кількість можливих

способів розподілу чотирьох об'єктів за двома класами дорівнює $2^4=16$. Очевидно з рис 3.5, що 2 з цих 16 дихотомій (а саме $\Omega_1 = \{\vec{x}_1, \vec{x}_3\}$ і $\Omega_2 = \{\vec{x}_2, \vec{x}_4\}$) лінійно реалізувати неможливо.

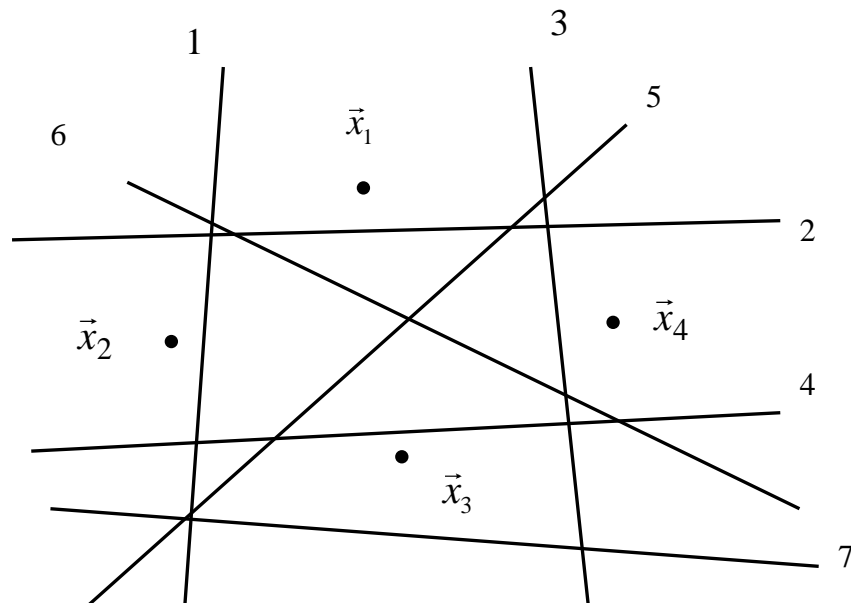


Рисунок 3.5 – Дихотомія лінійними вирішувальними функціями для чотирьох «добре» розташованих зображень в двовимірному просторі

Кількість лінійних дихотомій N точок в n -вимірному евклідовому просторі дорівнює подвоєному числу способів розподілу цих точок $(n-1)$ -вимірною площиною. Відомо, що при «доброму» розташуванні точок число лінійних дихотомій (дихотомізаційна потужність лінійних функцій) для N зображень розмірності n визначається таким виразом

$$D(N, n) = \begin{cases} 2 \sum_{k=0}^n C_{N-1}^k, \text{ якщо } N > n \\ 2^N, \text{ якщо } N \leq n \end{cases}, \quad (3.13)$$

де C_{N-1}^k – число сполучень із $N-1$ за k точками. Множина N точок називається «добре» розташованою в n -вимірному просторі, якщо ні одна із її підмножин, що містить $n+1$ точок, не лежить на $(n-1)$ -вимірній гіперплощині. Наприклад, N двовимірних точок «добре» розташовані, якщо ніякі три точки не лежать на одній прямій (одновимірній гіперплощині).

Розглянемо три можливих випадки розподілу об'єктів сукупності $\Omega = \{\omega_1, \omega_2, \dots, \omega_m\}$ на декілька класів $\Omega_1, \Omega_2, \dots, \Omega_k$.

Випадок 1. Кожен клас можна відділити від всіх інших однією розподільною поверхнею (гіперплощиною). Це спостерігається для достатньо рознесених в просторі класів (рис. 3.6).

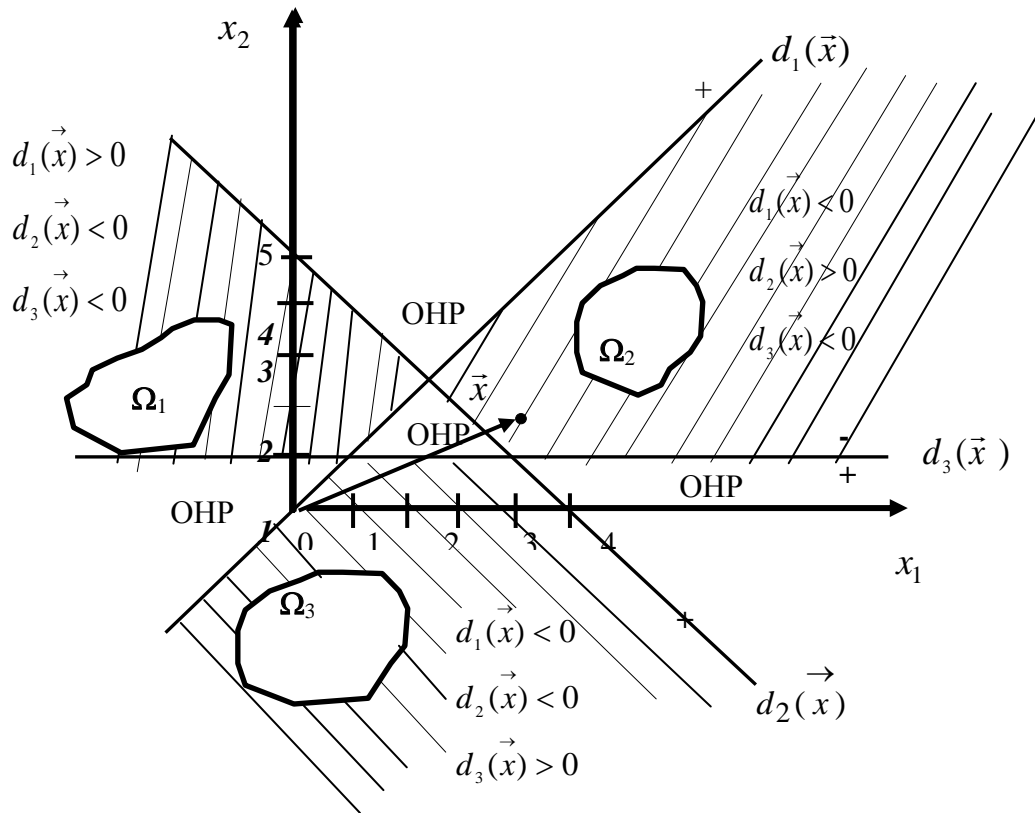


Рисунок 3.6 – Геометрична ілюстрація розподілу на класи в випадку 1 (ОНР – область невизначених рішень)

В цьому випадку існує k лінійних вирішувальних функцій таких, що

$$d_i(\vec{x}) = \vec{w}_i^T \vec{x} \begin{cases} > 0, \text{ якщо } \vec{x} \in \Omega_i, \\ < 0, \text{ якщо } \vec{x} \notin \Omega_i, \end{cases} \quad i = 1, 2, \dots, k. \quad (3.14)$$

Наприклад, якщо об'єкт \vec{x} належить образу Ω_1 , то $d_1(\vec{x}) > 0$. Границя, що відділяє клас Ω_1 від всіх інших класів, визначається значеннями \vec{x} , для яких $d_1(\vec{x}) = 0$.

З рис. 3.6 можна визначити рівняння розподільних границь:

$$d_1(\vec{x}) = -x_1 + x_2; \quad d_2(\vec{x}) = x_1 + x_2 - 5; \quad d_3(\vec{x}) = -x_2 + 1.$$

Віднесення об'єкта, що класифікується, здійснюється безпосередньо підстановкою його координат в рівняння вирішувальної функції.

Приклад 3.3. Виконати класифікацію зображення $\vec{x}^T = (4; 2)$, використовуючи правило (3.14).

Розв'язування. Виконавши підстановку координат вектора в рівняння вирішувальних функцій, отримаємо:

$$d_1(\vec{x}) = -2, \quad d_2(\vec{x}) = 1, \quad d_3(\vec{x}) = -1.$$

Оскільки $d_2(\vec{x}) > 0$, то відповідно з (3.14) приймаємо рішення, що $\vec{x} \in \Omega_2$.

Недоліком такого розподілу на класи є те, що в просторі ознак існують області невизначених рішень (ОНР), в яких вирішувальна функція $d_i(\vec{x})$ більше нуля, більше ніж для одного значення i .

Випадок 2. Кожен клас можна відділити від будь-якого іншого взятого окремо класу «індивідуальною» розподільною поверхнею, тобто класи попарно роздільні. Це характерно для близько згрупованих в просторі класів (рис. 3.7).

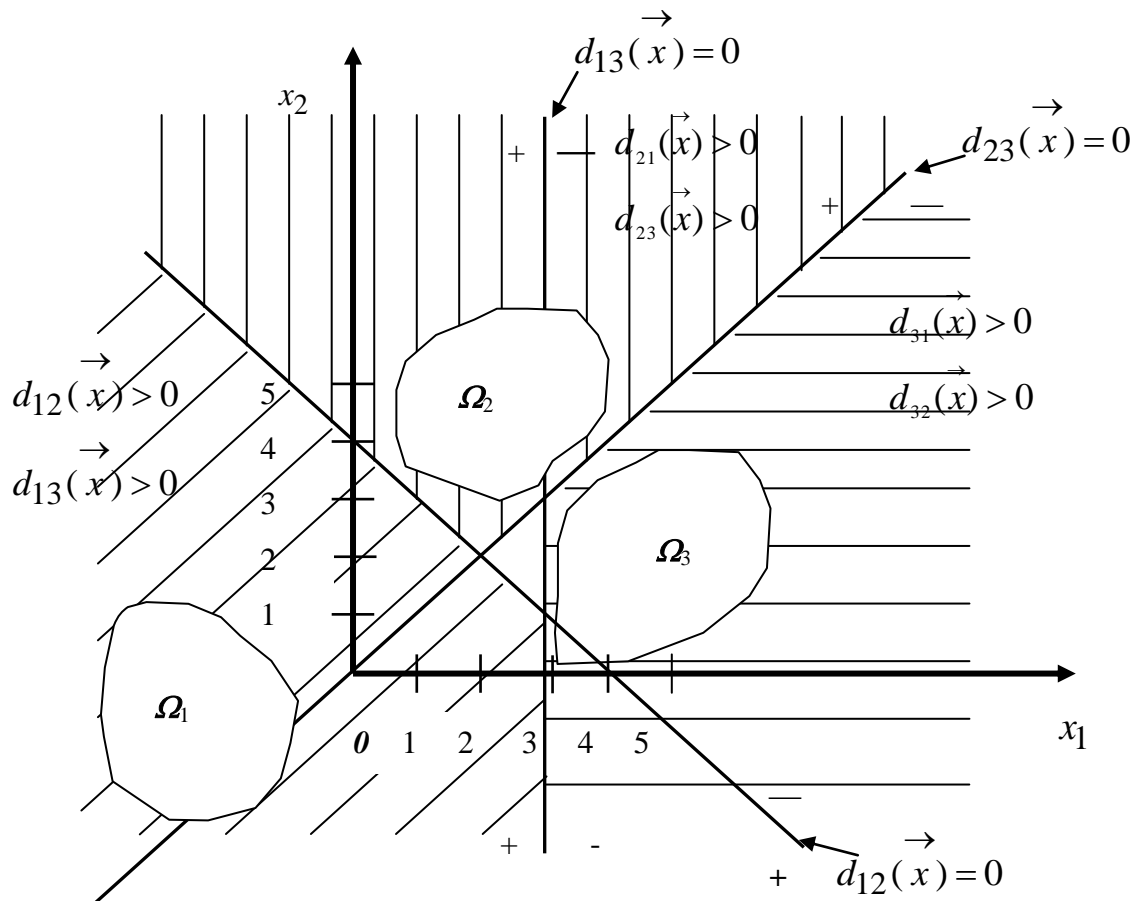


Рисунок 3.7 – Геометрична ілюстрація розподілу класів у випадку 2

В цьому випадку існує $C_k^2 = \frac{k(k-1)}{2}$ розподільних поверхонь (кількість сполучень із k класів по два). Вирішувальні функції мають вигляд: $d_{ij}(\vec{x}) = \vec{w}_{ij}^T \cdot \vec{x}$ і здійснюють класифікацію за правилом

$$\text{якщо } \forall_{i \neq j} d_{ij}(\vec{x}) > 0, \text{ то } \vec{x} \in \Omega_i, \quad (3.15)$$

що читається таким чином: для всякого \vec{x} , що належить класу Ω_i , виконується нерівність $d_{ij}(\vec{x}) > 0$ для всіх $i \neq j$. Крім того $d_{ij}(\vec{x}) = -d_{ji}(\vec{x})$.

З рис. 3.7 видно, що ні один клас неможливо відділити від всіх інших за допомогою єдиної розподільної поверхні. Кожна з проведених границь забезпечує розділ точно двох класів.

Наприклад, границя $d_{13}(\vec{x})=0$ проходить через клас Ω_2 , вона ефективно розділяє тільки класи Ω_1 і Ω_3 .

Наведені на рис. 3.7 вирішувальні функції мають такий вигляд:

$$d_{12}(\vec{x}) = -x_1 - x_2 + 4; \quad d_{13}(\vec{x}) = -x_1 + 3; \quad d_{23}(\vec{x}) = -x_1 + x_2.$$

Області рішень в цьому випадку, як видно з рисунка 3.7, можуть містити декілька зон, де вказані функції – додатні. Зокрема, область, що відповідає класу Ω_1 , визначається значеннями зображень \vec{x} , за яких $d_{12}(\vec{x}) > 0$ і $d_{13}(\vec{x}) > 0$. Значення вирішувальної функції $d_{23}(\vec{x})$ в цій області неістотне, оскільки ця вирішувальна функція ніяк не пов'язана з класом Ω_1 . З рисунка 3.7 також видно, що області рішень необмежені, і що існують області неприйняття рішень, для яких умови (3.14) випадку 2 не виконуються.

Приклад 3.4. Виконати класифікацію об'єкта, заданого вектором $\vec{x} = (5; 2)^T$, використавши правило (3.15).

Розв'язування. Підставимо значення координат вектора в вирішувальні функції, отримаємо:

$$d_{12}(\vec{x}) = -3; \quad d_{13}(\vec{x}) = -2; \quad d_{23}(\vec{x}) = -3,$$

що автоматично приводить до значень:

$$d_{21}(\vec{x}) = 3; \quad d_{31}(\vec{x}) = 2; \quad d_{32}(\vec{x}) = 3.$$

Оскільки $d_{3j}(\vec{x}) > 0$ для $j=1, 2$ і в область невизначеності ми не потрапили, то згідно з правилом (3.15) $\vec{x} \in \Omega_3$.

Випадок 3. Існує k вирішувальних функцій таких, що, якщо образ \vec{x} належить класу Ω_i , то виконується умова

$$\forall_{j \neq i} d_i(\vec{x}) > d_j(\vec{x}), \quad i = \overline{1, k}. \quad (3.16)$$

Ця ситуація є різновидом випадку 2, оскільки можна вважати, що

$$d_{ij}(\vec{x}) = d_i(\vec{x}) - d_j(\vec{x}) = (\vec{w}_i - \vec{w}_j)^T \vec{x} = \vec{w}_{ij}^T \vec{x}. \quad (3.17)$$

Легко впевнитися в тому, що із справедливості умови $d_i(\vec{x}) > d_j(\vec{x})$ для всіх $j \neq i$ витікає справедливність виразу $d_{ij}(\vec{x}) > 0$ для всіх $j \neq i$,

тобто, якщо класи розділяються так, як у випадку 3, то вони автоматично розділяються і як у випадку 2. Границя між класами Ω_i і Ω_j визначається тими значеннями вектора \vec{x} , за яких $d_i(\vec{x}) = d_j(\vec{x})$, тобто $d_i(\vec{x}) - d_j(\vec{x}) = 0$. Таким чином, у разі виведення рівняння розподільної границі для образів Ω_i і Ω_j значення вирішувальних функцій використовують сумісно.

Геометричну ілюстрацію випадку 3 для трьох класів ($k=3$) наведено на рис. 3.8.

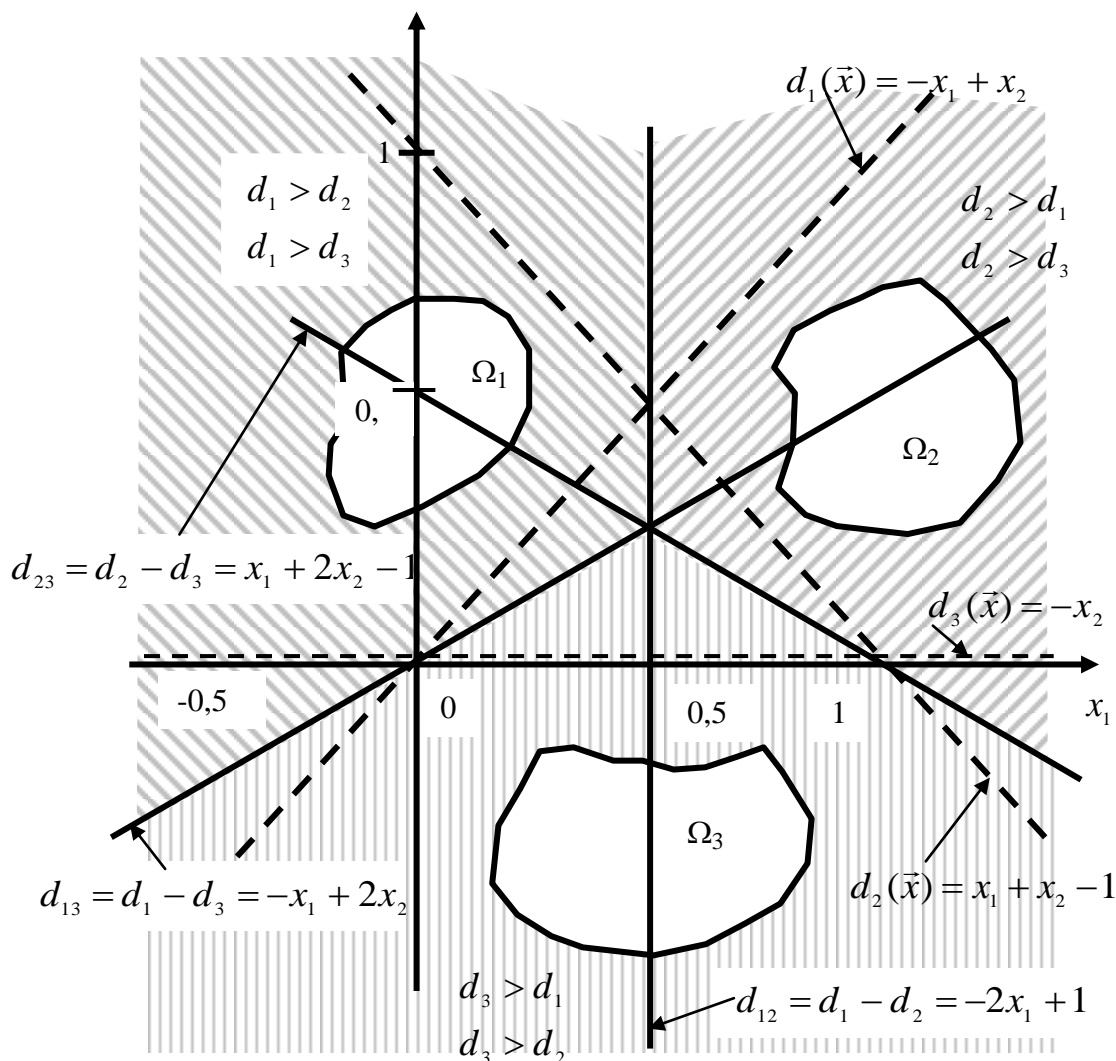


Рисунок 3.8 – Геометрична ілюстрація розподілу класів для випадку 3

В цьому прикладі вирішувальні функції, що відділяють кожен клас, мають вигляд:

$$d_1(\vec{x}) = -x_1 + x_2; \quad d_2(\vec{x}) = x_1 + x_2 - 1; \quad d_3(\vec{x}) = -x_2.$$

Відповідно до вказаного в (3.16) для цього випадку правила класифікації, для зображень, що належать класу Ω_1 , мають виконуватися

умови $d_1(\bar{x}) > d_2(\bar{x})$ і $d_1(\bar{x}) > d_3(\bar{x})$, для зображень класу $\Omega_2 - d_2(\bar{x}) > d_1(\bar{x})$ і $d_2(\bar{x}) > d_3(\bar{x})$, для зображень класу $\Omega_3 - d_3(\bar{x}) > d_1(\bar{x})$ і $d_3(\bar{x}) > d_2(\bar{x})$.

Це еквівалентно вимозі того, щоб зображення, які входять в клас Ω_i , розміщувались в додатних зонах поверхонь $d_i(\bar{x}) - d_j(\bar{x}) = 0$, $j = \overline{1, k}$, $j \neq i$. В цьому випадку додатна зона границі $d_i(\bar{x}) - d_j(\bar{x}) = 0$ збігається з від'ємною $d_j(\bar{x}) - d_i(\bar{x}) = 0$.

Розподільні границі для трьох класів будуть мати в цьому прикладі такий вигляд:

$$\begin{aligned} d_1(\bar{x}) - d_2(\bar{x}) &= -2x_1 + 1 = 0; & d_1(\bar{x}) - d_3(\bar{x}) &= -x_1 + 2x_2 = 0; \\ d_2(\bar{x}) - d_3(\bar{x}) &= x_1 + 2x_2 - 1 = 0. \end{aligned}$$

Як видно з рис 3.8, область рішень, що відповідає класу Ω_1 , збігається з додатними зонами для прямих $-2x_1 + 1 = 0$ і $-x_1 + 2x_2 = 0$. Область прийняття рішень про належність зображення до класу Ω_2 збігається з додатними зонами прямих $2x_1 - 1 = 0$ і $x_1 + 2x_2 - 1 = 0$. Область класу Ω_3 визначається додатними зонами для прямих $x_1 - 2x_2 = 0$ і $-x_1 - 2x_2 + 1 = 0$.

Використання правила класифікації (3.15) для випадку 3 дає можливість позбавитися областей неприйняття рішень, які мають місце для правила класифікації (3.13) у випадку 1.

Приклад 3.5. Виконаємо класифікацію об'єкта, зображеного вектором $\bar{x} = (1; 1)^T$.

Розв'язування: підставимо компоненти вектора в задані функції $d_1(\bar{x}), d_2(\bar{x}), d_3(\bar{x})$, маємо $d_1(\bar{x}) = 0$, $d_2(\bar{x}) = 1$, $d_3(\bar{x}) = -1$

Оскільки $d_2(\bar{x}) > d_j(\bar{x})$, $j = 1, 3$, то згідно з правилом (3.15) зображення відноситься до класу Ω_2 .

Класи, які можна розділити функціями, описаними для випадків 1, 2 і 3, називаються такими, що лінійно розділяються. Основна проблема, яка виникає після визначення набору вирішувальних функцій, полягає в знаходженні вагових коефіцієнтів. Як уже раніше підкреслювалось, для їх визначення використовується доступна вибірка зображень об'єктів.

3.3 Узагальнені вирішувальні функції

Зрозуміло, що для класів, які не містять однакових зображень об'єктів, можна завжди побудувати роздільні границі. Їх складність може змінюватись від найпростіших лінійних до складних, суто нелінійних, які описуються великою кількістю членів. Обчислення складних нелінійних

функцій може потребувати великих затрат часу і пам'яті, які в багатьох випадках є недоступними, і в цьому випадку образи в істинному розумінні не є розподілюваними. В таких випадках шукають їх наближення у вигляді композиції деяких простіших вирішувальних функцій $\{f_i(\vec{x})\}$, $i = 1, 2, \dots, M$.

Щоб узагальнити на такі випадки математичні результати для лінійних вирішувальних функцій, введемо вирішувальні функції виду

$$d(\vec{x}) = w_1 f_1(\vec{x}) + w_2 f_2(\vec{x}) + \dots + w_M f_M(\vec{x}) + w_{M+1} = \sum_{i=1}^{M+1} w_i f_i(\vec{x}), \quad (3.18)$$

де $\{f_i(\vec{x})\}$, $i = 1, 2, \dots, M$ – дійсні однозначні функції зображення \vec{x} , $f_{M+1}(\vec{x}) = 1$, а $M+1$ – кількість членів в розкладі. Співвідношення (3.18) репрезентує нескінченну множину вирішувальних функцій, вигляд яких залежить від вибору функцій $\{f_i(\vec{x})\}$ і кількості членів, використаних в розкладі.

Використання перетворення (3.18) дозволяє працювати зі складними вирішувальними функціями як з лінійними. Покажемо це. Визначимо вектор \vec{x}' таким чином:

$$\vec{x}' = \begin{bmatrix} f_1(\vec{x}) \\ f_2(\vec{x}) \\ \dots \\ f_M(\vec{x}) \\ 1 \end{bmatrix}. \quad (3.19)$$

Тоді вираз (3.18) можна записати як

$$d(\vec{x}') = \vec{w}^T \vec{x}', \quad (3.20)$$

де $\vec{w} = (w_1, w_2, \dots, w_M, w_{M+1})^T$.

Перетворення всіх початкових зображень \vec{x} в зображення \vec{x}' шляхом обчислення всіх M значень функції $f_i(\vec{x})$ перетворює задачу класифікації в лінійну. Незважаючи на те, що в M -вимірному просторі вирішувальні функції можна вважати лінійними, в n -вимірному просторі початкових зображень вони повністю зберігають свій принципово нелінійний характер.

Найбільш поширеним є спосіб задання узагальнених вирішувальних функцій у вигляді многочленів. Многочлен першого степеня задає лінійну вирішувальну функцію вигляду (3.3). В цьому випадку $f_i(\vec{x}) = x_i, i = 1, 2, \dots, n$. Многочлен другого степеня задає вирішувальні

функції другого порядку (квадратичні). Наприклад, для двовимірного простору ($\vec{x} = (x_1, x_2)^T$) вирішувальні функції мають вигляд:

$$d(\vec{x}) = w_{11}x_1^2 + w_{12}x_1x_2 + w_{22}x_2^2 + w_1x_1 + w_2x_2 + w_3. \quad (3.21)$$

Лінійне зображення функції (3.21) можна отримати, поклавши $\vec{x}' = (x_1^2, x_1x_2, x_2^2, x_1, x_2, 1)^T$ і $\vec{w} = (w_{11}, w_{12}, w_{22}, w_1, w_2, w_3)^T$, тоді $d(\vec{x}') = \vec{w}^T \vec{x}'$.

В загальному випадку для n -вимірного випадку квадратна функція за аналогією з (3.21) запишеться у вигляді:

$$d(\vec{x}) = \sum_{j=1}^n w_{jj}x_j^2 + \sum_{j=1}^{n-1} \sum_{k=j+1}^n x_jx_k + \sum_{j=1}^n w_jx_j + w_{n+1}. \quad (3.22)$$

Кількість членів, а отже, і параметрів в (3.22) дорівнює

$$n + \frac{n(n-1)}{2} + n + 1 = \frac{(n+1)(n+2)}{2}.$$

Вирішувальні функції, що задаються многочленом r -го степеня, можна записати рекурентним співвідношенням:

$$d^r(\vec{x}) = \left(\sum_{p_1=1}^n \sum_{p_2=p_1}^n \dots \sum_{p_r=p_{r-1}}^n w_{p_1 p_2 \dots p_r} x_{p_1} x_{p_2} \dots x_{p_r} \right) + d^{r-1}(\vec{x}), \quad (3.23)$$

де $d^0(\vec{x}) = w_{n+1}$

Приклад 3.6. Записати квадратичну вирішувальну функцію, скориставшись формулою (3.23), для двовимірного простору ознак, $n=2$.

Розв'язування: за умовою маємо $r=2$, $n=2$, тоді з (3.22) можемо записати

$$d^2(x) = \sum_{p_1=1}^2 \sum_{p_2=p_1}^2 w_{p_1 p_2} \cdot x_{p_1} \cdot x_{p_2} + d^1(\vec{x}), \quad (3.24)$$

де $d^1(x) = \sum_{p_1=1}^2 w_{p_1} x_{p_1} + d^0(x) = w_1x_1 + w_2x_2 + w_3$.

Підставляємо це значення в (3.24) замість $d^1(\vec{x})$ і розкриваємо суми:

$$d^2(x) = w_{11}x_1^2 + w_{12}x_1x_2 + w_{22}x_2^2 + w_1x_1 + w_2x_2 + w_3.$$

Останній вираз збігається з (3.20), що і потрібно було показати.

Число членів N_w , необхідне для подання поліноміальної вирішувальної функції, є функцією від порядку r і розмірності n та дуже

швидко зростає під час їх збільшення. Використовуючи метод математичної індукції, неважко показати, що

$$N_w = C_{n+r}^r = \frac{(n+r)!}{r!n!}, \quad (3.25)$$

де C_{n+r}^r – число сполучень із $n+r$ за r . Потрібно визначити, що немає необхідності завжди використовувати всі члени, що визначають загальний вигляд розкладу (3.23). Так, під час побудови вирішувальної функції другого порядку можна відмовитися від всіх членів, лінійних відносно компонент зображення \vec{x} .

Для визначення форми розподільної поверхні, що задається квадратною вирішувальною функцією (3.22), подамо її у матричній формі. Для цього введемо такі позначення:

$$a_{jj} = w_{jj}, a_{jk} = \frac{w_{jk}}{2}, b_j = w_j, c = w_{n+1}, j, k = 1, 2, \dots, n.$$

Тоді вираз (3.21) можна подати у вигляді

$$d(x) = \vec{x}^T \mathbf{A} \vec{x} + \vec{x}^T \vec{b} + c, \quad (3.26)$$

де $\mathbf{A} = \|a_{jk}\|, b = (b_1, b_2, \dots, b_n)^T$.

Властивості матриці \mathbf{A} для квадратичної вирішувальної функції визначають форму розподільної границі. Як відомо з аналітичної геометрії, якщо \mathbf{A} – одинична матриця, то вирішувальна функція являє собою гіперсферу, якщо \mathbf{A} – додатно визначена матриця – гіпереліпсоїд, напрямком осей якого задається власними векторами матриці \mathbf{A} , якщо \mathbf{A} – від'ємно визначена матриця, то границя є гіпергіперболоїдом.

Визначимо дихотомізаційну потужність узагальнених вирішувальних функцій. Для N перетворених образів з «добрим» розташуванням існує 2^N дихотомій, з яких лінійно можна реалізувати тільки $\mathcal{D}(N, M)$ (3.13), де $M+1$ – число вагових параметрів в формулі узагальненої вирішувальної функції (3.18). Тоді ймовірність $P_{N, K}$ того, що варіант дихотомії, вибраний випадковим чином, можна лінійно реалізувати, визначиться виразом:

$$P_{N, K} = \frac{\mathcal{D}(N, M)}{2^N} = \begin{cases} 2^{1-N} \sum_{j=0}^M C_{N-1}^j, & \text{якщо } N > M \\ 1, & \text{якщо } N \leq M \end{cases}. \quad (3.27)$$

З формули (3.27) можна зробити висновок, що для випадку, коли число класифікованих об'єктів N менше або дорівнює числу вагових параметрів M (M – це також число допоміжних функцій $f_i(\vec{x})$ у виразі

(3.17)), то незалежно від способу від способу групування заданих об'єктів їх можна лінійно класифікувати в M -вимірному просторі зображень.

Проведений аналіз показує, що дихотомізаційна потужність узагальненої вирішувальної функції C_D дорівнює:

$$C_D = 2(M + 1), \quad (3.28)$$

тобто подвоєному числу ступенів свободи (регульованих параметрів) узагальненої вирішувальної функції.

3.4 Класифікація образів за критерієм мінімуму відстані

Найбільш наочним способом введення міри подібності для векторів об'єктів, які зображаються точками в просторі ознак, є визначення відстані між ними. Наприклад, на рис. 3.9, а) видно, що інтуїтивно точку \bar{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. 3.9, а). В іншому випадку (рис. 3.9, б) зарахування зображень в один із класів на ґрунті оцінки його близькості до об'єктів відповідного класу викликає ускладнення.

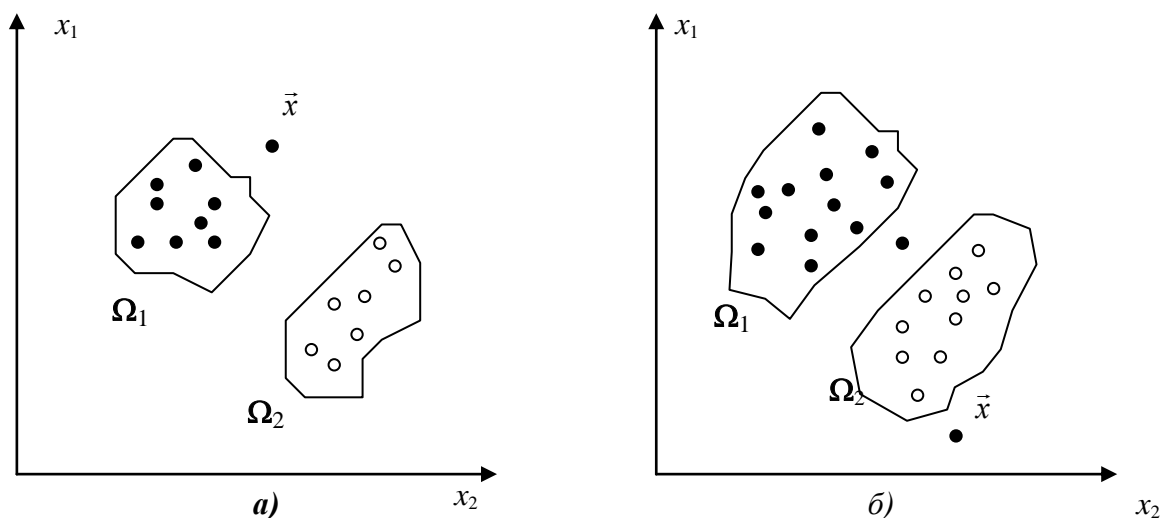


Рисунок 3.9 – Класифікація об'єктів за функцією відстані: а) – сприятливе розміщення об'єктів для класифікації за мінімумом відстані; б) – несприятливе

Оскільки близькість зображення, що класифікується, до деякого класу використовується за критерій його класифікації, то такий підхід називається класифікацією за критерієм мінімуму відстані. Така класифікація зручна в випадках, коли зображення об'єктів будь-якого з

класів проявляють тенденцію до тісного згрупування відносно деякої точки, яку називають еталоном відповідного класу.

Розглянемо K класів, що репрезентуються еталонами $\vec{z}_1, \vec{z}_2, \dots, \vec{z}_K$. Евклідова відстань між довільним вектором об'єкта \vec{x} і i -им еталоном визначається таким виразом:

$$D_i = \|\vec{x} - \vec{z}_i\| = \sqrt{(\vec{x} - \vec{z}_i)^T (\vec{x} - \vec{z}_i)}. \quad (3.29)$$

Введена таким чином відстань має такі властивості:

$D(\vec{x}_1, \vec{x}_1) = 0$ – ідентичність;

$D(\vec{x}_1, \vec{x}_2) = D(\vec{x}_2, \vec{x}_1)$ – симетрія;

$D(\vec{x}_1, \vec{x}_3) \leq D(\vec{x}_1, \vec{x}_2) + D(\vec{x}_2, \vec{x}_3)$ – нерівність трикутника,

де $\vec{x}_1, \vec{x}_2, \vec{x}_3$ – три різних точки простору. Така відстань називається метричною.

Вираз (3.29) можна розписати через координати векторів \vec{x} і \vec{z}_i у вигляді, зручному для обчислення:

$$D_i = \sqrt{\sum_{j=1}^n (x_j - z_{ij})^2}, \quad (3.30)$$

де $j=1, 2, \dots, n$ – номери координат.

Образ \vec{x} зараховується до класу Ω_i за умови:

$$\vec{x} \in \Omega_i, \text{ якщо } D_i = \min\{D_i\} \quad (3.31)$$

де $i=1, 2, \dots, K$.

Виконаємо ряд перетворень виразу (3.29). Піднесемо всі її члени до квадрата:

$$\begin{aligned} D_i^2 &= \|\vec{x} - \vec{z}_i\|^2 = (\vec{x} - \vec{z}_i)^T (\vec{x} - \vec{z}_i) = \vec{x}^T \vec{x} - 2\vec{x}^T \vec{z}_i + \vec{z}_i^T \vec{z}_i = \\ &= \vec{x}^T \vec{x} - 2(\vec{x}^T \vec{z}_i - \frac{1}{2} \vec{z}_i^T \vec{z}_i). \end{aligned} \quad (3.32)$$

Вибір мінімального D_i еквівалентний вибору мінімального D_i^2 , оскільки відстані є величинами невід'ємними. З формули (3.32) видно, що вибір мінімального значення D_i^2 рівносильний вибору максимального значення різниці $(\vec{x}^T \vec{z}_i - \frac{1}{2} \vec{z}_i^T \vec{z}_i)$, оскільки член $\vec{x}^T \vec{x}$ однаковий для всіх i . Отже, вирішувальні функції можна визначити таким чином:

$$d_i(\vec{x}) = \vec{x}^T \vec{z}_i - \frac{1}{2} \vec{z}_i^T \vec{z}_i, \quad i = 1, 2, \dots, K, \quad (3.33)$$

і правило класифікації має вигляд

$$\forall d_l(\vec{x}) > d_i(\vec{x}) \rightarrow \vec{x} \in \Omega_l, \quad l = 1, 2, \dots, K.$$

Якщо покласти $i=1, 2, \dots, K$ – номер класу, а $j=1, 2, \dots, n$ – номер координати простору ознак, то (3.33) можна переписати у вигляді:

$$d_i(\vec{x}) = \vec{w}_i^T \vec{x}, \quad (3.34)$$

де $\vec{w}_i = (w_{i1}, w_{i2}, \dots, w_{i,n+1})$, $w_{ij} = z_{ij}$, $\vec{x}^T = (x_1, x_2, \dots, x_n, 1)^T$.

Звідси видно, що $d_i(\vec{x})$ – лінійна вирішувальна функція. Вона є гіперплощиною, яка є геометричним місцем точок, рівновіддалених від двох еталонних точок, що являють собою образи Ω_i і Ω_j .

Таким чином ми впевнились, що класифікатор, який базується на принципі мінімальної відстані, є окремим випадком лінійного класифікатора. Оскільки він класифікує образи, виходячи з найбільш повного збігу зображення з еталонами відповідних класів, то його називають кореляцією з кластером.

3.5 Класифікація образів за методом опорних векторів SVM

3.5.1 Загальний опис ідеї класифікації за методом опорних векторів

В попередніх пунктах цього розділу було розглянуто математичні основи побудови детермінованих класифікаторів лінійного типу і таких, що до них зводяться за допомогою узагальнених вирішувальних функцій за нелінійних границь класів. Одним з найкращих класифікаторів такого типу на сьогодні є машина опорних векторів, скорочено SVM (Support Vector Mashine).

Порівняно з більш новими алгоритмами, такими як нейронні мережі, методи SVM мають дві основних переваги: високу швидкість і кращу продуктивність для даних невисокого і середнього обсягу – до декількох тисяч елементів. До мінусів можна віднести повільну роботу у разі високих розмірів даних, що пов'язано зі збільшенням часу на навчання, а також наявність помилок за наявності викидів в наборах даних, тобто під час перекривання цільових класів. Крім того, метод безпосередньо не визначає ймовірності помилки класифікації даних.

«Машина опорних векторів» або метод опорних векторів – це контрольований алгоритм машинного навчання, який може використовуватися для задач класифікації та регресії. В переважній

більшості він використовується в задачах класифікації. В таких задачах кожен елемент даних зображається точкою в n -вимірному просторі (де n – це кількість ознак для опису об'єкта, що підлягає класифікації), в якому значення кожної ознаки є значенням певної координати. Отже, алгоритм навчання методом опорних векторів для класифікації даних шукає гіперплощину, яка дуже добре розділяє два класи.

Гіперплощина – це підплощина з розмірністю $n-1$ в n -вимірному Евклідовому просторі, яка розділяє цей простір на дві окремі частини. Наприклад, якщо набір даних лежить на одній прямій (наприклад, на числовій осі), то «гіперплощина» буде подана точкою, якщо дані подано точками на площині, то «гіперплощиною» буде лінія. В 3-вимірному просторі це буде уже площина, а в просторах з розмірністю більше 3 – дійсно гіперплощина.

Машина опорних векторів (SVM) – це дискримінантний класифікатор, алгоритм якого якраз і ґрунтується формально на понятті розподільної гіперплощини. Іншими словами, отримавши під час навчання марковані дані (контрольоване навчання), алгоритм буде оптимальну гіперплощину, яка класифікує нові прецеденти об'єктів оптимальним чином з погляду точності. Оптимальність побудованого класифікатора у вигляді гіперплощини виясномо на простому прикладі, який ілюструє рис. 3.10. Для лінійно розподільної множини точок на 2-вимірному просторі (на площині) потрібно знайти розподільну (дискримінантну) пряму.

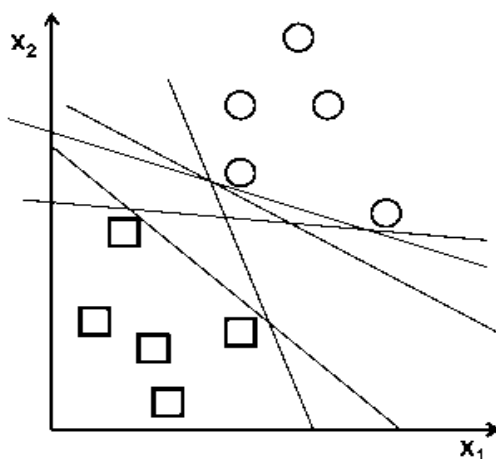


Рисунок 3.10 – Приклад пошуку розподільної прямої на площині для двох класів

Для спрощення розуміння проблеми тут вибрано приклад опису об'єктів двох класів на декартовій площині (двовимірний простір ознак) у вигляді точок, які є кінцями векторів зображень цих об'єктів. В цьому випадку лінійний класифікатор подано прямими лініями, що спрощує нашу зорову уяву про розподіл об'єктів на два класи, оскільки тут ми маємо справу з прямими і точками замість гіперплощин і векторів у

просторі вищої розмірності. Це не виключає можливості застосування таких самих понять до класифікаційних задач у просторі, розмірність якого перевищує два.

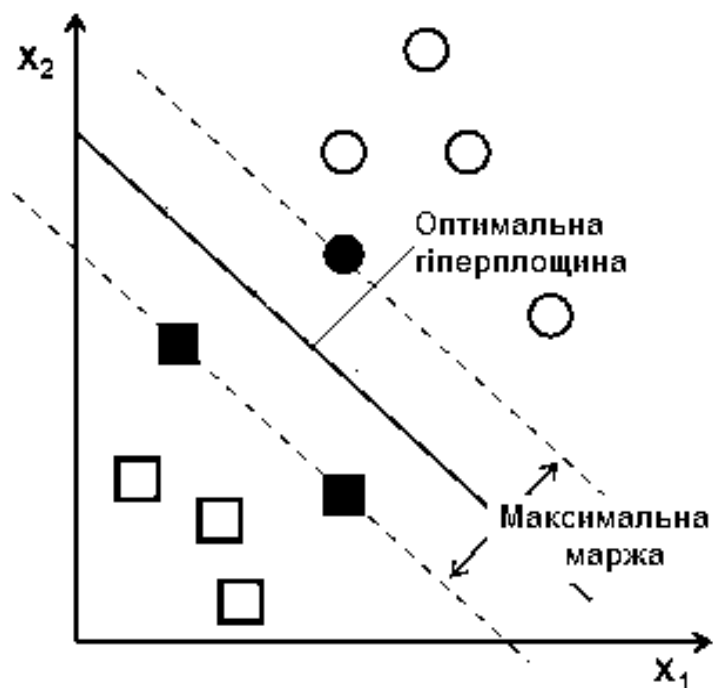


Рисунок 3.11 – Приклад оптимальної розподільної прямої на площині

На рис. 3.10 видно, що існує кілька прямих, які дозволяють виконати класифікацію. Виникає питання, яка з них найкраща. Інтуїтивно можна вибрати критерій для оцінення якості дискримінантних (вирішувальних) прямих таким чином: пряма погана, якщо вона проходить занадто близько до точок, оскільки він буде чутливим до завад і не мати узагальнювального характеру в разі класифікації нових точок, які не брали участь у навчанні, тобто не буде класифікувати їх правильно. Тому мета дослідження полягає в тому, щоб знайти лінію, яка проходить якомога далі від усіх точок (рис. 3.11).

В цьому плані, по-перше, робота алгоритму SVM основана на пошуку гіперплощини, яка дає найбільшу мінімальну відстань до навчальних прикладів.

По-друге, ця відстань отримує важливу назву **маржі** в теорії SVM. Тому оптимальна розподільна гіперплощин максимізує запас **маржі** за навчальними даними (рис. 3.11).

Приклад 3.7. Розглянемо процес створення класифікатора за методом SVM на основі дуже малого набору даних, який містить дві точки першого класу (чорні кружечки) з координатами (1,1) і (2,0) та одну точку другого класу (трикутник) з координатами (2, 3), поданого на рис. 3.12.

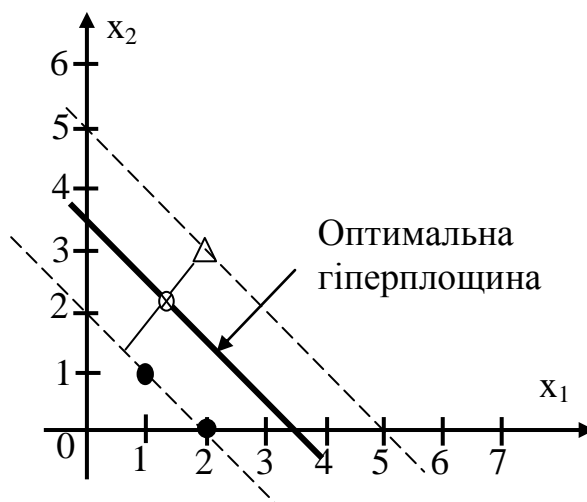


Рисунок 3.12 – Геометрична ілюстрація розв’язання прикладу 3.7

Розв’язування: вважаючи опорними обидві точки 1-го класу і точку 2-го класу, проведемо границю першого класу у вигляді прямої лінії, що проходить через дві точки 1-го класу, а границю 2-го класу у вигляді прямої лінії, що проходить через точку другого класу паралельно уже проведеній. Опустимо з точки другого класу нормаль на граничну лінію 1-го класу і через середину отриманого відрізка проведемо пряму, паралельну границям класів. Її рівняння і буде рівнянням оптимальної гіперплощини з максимальним проміжком (маржею).

З погляду аналітичної геометрії рівняння прямої $d_1(\vec{x}) = w_1x_1 + w_2x_2 - b_1$, що проходить через точки (1,1) і (2,0) і є межею 1-го класу, запишеться як: $x_1 + x_2 - 2 = 0$, звідки ваговий вектор $\vec{w} = (1,1)$.

Тоді рівняння прямої $d_2(\vec{x}) = w_1x_1 + w_2x_2 - b_2$, що є межею 2-го класу, знайдемо з умови, що її ваговий вектор однаковий з ваговим вектором першої граничної лінії (оскільки вона паралельна їй), і підстановка координат опорної точки 2-го класу в її рівняння дає значення 0: $w_1x_1 + w_2x_2 - b_2 = 0$; $1 \cdot 2 + 1 \cdot 3 = b_2$, звідки $b_2 = 5$, і в остаточному вигляді маємо рівняння $x_1 + x_2 - 5 = 0$.

Оптимальна гіперплощина $d_{opt}(\vec{x}) = w_1x_1 + w_2x_2 + b_{opt}$ проходить посередині між проведеними прямими, тобто її зміщення b_{opt} відносно 0 по осі ординат дорівнює середньому арифметичному зміщень b_1 і b_2 і має такий самий ваговий вектор:

$$b_{opt} = \frac{b_1 + b_2}{2} = \frac{2 + 5}{2} = 3.5.$$

Отже, рівняння оптимальної розподільної прямої для цього набору даних має вигляд: $x_1 + x_2 - 3.5 = 0$.

Насправді метод SVM спочатку шукає шляхом навчання опорні точки на границях обох класів за допомогою оптимізаційної процедури, а після цього виконує процедуру побудови оптимальної гіперплощини.

3.5.2 Постановка задачі навчання класифікатора за методом SVM

Сформулюємо математичну задачу навчання класифікатора SVM.

Початкові дані для навчання класифікатора SVM:

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

де U – навчальна вибірка векторів об'єктів \vec{x}_i , причому існують такі марки y_i цих векторів, які визначаються таким чином:

якщо $\vec{x}_i \in \Omega_1$, тоді $y_i = 1$, якщо $\vec{x}_i \in \Omega_2$, тоді $y_i = -1$, тобто y_i визначає, до якого класу належить об'єкт, описаний вектором ознак \vec{x}_i ;

R^n – n -вимірний простір ознак (параметрів);

m – кількість векторів навчальної вибірки.

Задача навчання: за навчальною вибіркою побудувати таку дискримінантну функцію $f(\vec{x})$, яка б відповідала вимогам:

$$\begin{cases} f(\vec{x}_i) > 0, & \vec{x}_i \in \Omega_1 \\ f(\vec{x}_i) < 0, & \vec{x}_i \in \Omega_2 \end{cases}, \text{ або } y_i \cdot f(\vec{x}_i) > 0, \quad \vec{x}_i \in U. \quad (3.35)$$

Узагальнюючи два рівняння на одне у виразі (3.35), ми можемо записати, що лінійний класифікатор на 2 класи записується таким чином:

$$f(\vec{x}_i) = \text{sign}(\vec{w}^T \vec{x}_i + w_0). \quad (3.36)$$

Якщо значення виразу (3.36) дорівнює $+1$, то об'єкт належить до першого класу, якщо -1 , то до другого. Оскільки під час навчання класифікатора значення $+1$ і -1 для векторів навчальної вибірки x_i відомі і задаються значеннями y_i , то для етапу навчання класифікатора вираз (3.36) можна записати у вигляді:

$$f(\vec{x}_i) = y_i \cdot (\vec{w}^T \vec{x}_i + w_0). \quad (3.37)$$

Вираз (3.37) визначає відстань від точок 1-го і 2-го класу до гіперплощини $\vec{w}^T \vec{x} = 0$. Назвемо цю відстань функціональним зазором, або **маржою**. Як уже було показано в розділі 3.2, нормалізація цього зазору шляхом його ділення на норму вектора нормалі \vec{w} дозволяє усунути проблему масштабування даних в різних задачах з погляду геометричної

відстані. Завдяки цьому геометричний зазор не буде залежати від масштабування, наприклад, заміна параметрів \vec{w} на $5\vec{w}$ і w_0 на $5w_0$ не приведе до його зміни. Якщо установити обмеження $\|w\|=1$, то в цьому випадку геометричний зазор збігається з функціональним.

Таким чином, якщо ми для масштабування домножимо $f(\vec{x})$ на деяке додатне число і враховуючи, що $f(\vec{x})$ – лінійна функція, з (3.37) маємо:

$$y_i \cdot (\vec{w}^T \vec{x}_i + w_0) \geq 1, \quad i = 1, \dots, m. \quad (3.38)$$

Промасштабована відстань між граничними гіперплощинами $\vec{w}^T \vec{x}_i + w_0 - 1 = 0$ і $\vec{w}^T \vec{x}_i + w_0 + 1 = 0$, що проходять через опорні точки:

$$r = \frac{2}{\|w\|}. \quad (3.39)$$

З виразу (3.39) видно, що для максимізації маржі необхідно мінімізувати величину $\|w\|$. Однак в такому вигляді задачу оптимізації важко вирішити, оскільки вона залежить від норми $\|w\|$, до складу якої входить квадратний корінь. Її можна спростити, замінивши $\|w\|$ на $\|w\|^2$ і вибравши коефіцієнт $1/2$, що не змінить рішення. Це квадратична задача оптимізації.

Задача оптимізації – пошук вагового вектора, що максимізує r :

$$\frac{1}{2} \|w\|^2 \rightarrow \min \text{ за обмежень: } y_i \cdot (\vec{w}^T \vec{x}_i + w_0) \geq 1, \quad i = 1, \dots, m. \quad (3.40)$$

Шляхом введення множників Лагранжа задача оптимізації з обмеженнями може бути подана у вигляді задачі безумовної оптимізації:

$$\min_{w, w_0} \max_{\alpha \geq 0} \left\{ \frac{1}{2} \|w\|^2 - \sum_{i=1}^m \alpha_i (y_i (\vec{w}^T \vec{x}_i + w_0) - 1) \right\}. \quad (3.41)$$

Можна показати, що задачу пошуку оптимальних опорних точок можна розв'язати шляхом використання лагранжіана:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \vec{x}_i^T \vec{x}_j \quad (3.42)$$

за умов

$$\sum_i \alpha_i y_i = 0,$$

$$\alpha_i \geq 0 \quad \forall (i = \overline{1, m}),$$

де α_i – множники Лагранжа.

Розв'язок цієї задачі має такий вигляд:

$$\vec{w} = \sum \alpha_i y_i \vec{x}_i. \quad (3.43)$$

З (3.43) випливає, що ваговий вектор для оптимальної гіперплощини можна отримати з виразу:

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

де i – номер опорної точки (вектора), $i=1 \dots k$;

k – число шуканих опорних точок,

α_i – множники Лагранжа, визначені для опорних точок процедурою оптимізації.

3.5.3 Алгоритмізація процесу навчання класифікатора SVM

В обчислювальному плані алгоритм пошуку величин множників Лагранжа для побудови оптимальної гіперплощини за виразом (3.43) зводиться до розв'язку системи лінійних алгебраїчних рівнянь, яку отримують шляхом визначення частинних похідних для заданої функції Лагранжа за вибраними множниками α_i у разі відомих значень координат x_{ji} опорних точок \vec{x}_i та значень марок y_i :

$$\begin{pmatrix} n & \sum_{i=1}^n x_{1i} & \sum_{i=1}^n x_{2i} & \dots & \sum_{i=1}^n x_{mi} \\ \sum_{i=1}^n x_{1i} & \sum_{i=1}^n x_{1i}x_{1i} & \sum_{i=1}^n x_{1i}x_{2i} & \dots & \sum_{i=1}^n x_{1i}x_{mi} \\ \sum_{i=1}^n x_{2i} & \sum_{i=1}^n x_{2i}x_{1i} & \sum_{i=1}^n x_{2i}x_{2i} & \dots & \sum_{i=1}^n x_{2i}x_{mi} \\ \dots & \dots & \dots & \dots & \dots \\ \sum_{i=1}^n x_{mi} & \sum_{i=1}^n x_{mi}x_{1i} & \sum_{i=1}^n x_{mi}x_{2i} & \dots & \sum_{i=1}^n x_{mi}x_{mi} \end{pmatrix} \cdot \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_m \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n y_i x_{1i} \\ \sum_{i=1}^n y_i x_{2i} \\ \dots \\ \sum_{i=1}^n y_i x_{mi} \end{pmatrix} \quad (3.44)$$

Алгоритм навчання класифікатора типу SVM наведено на рис. 3.13.

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.

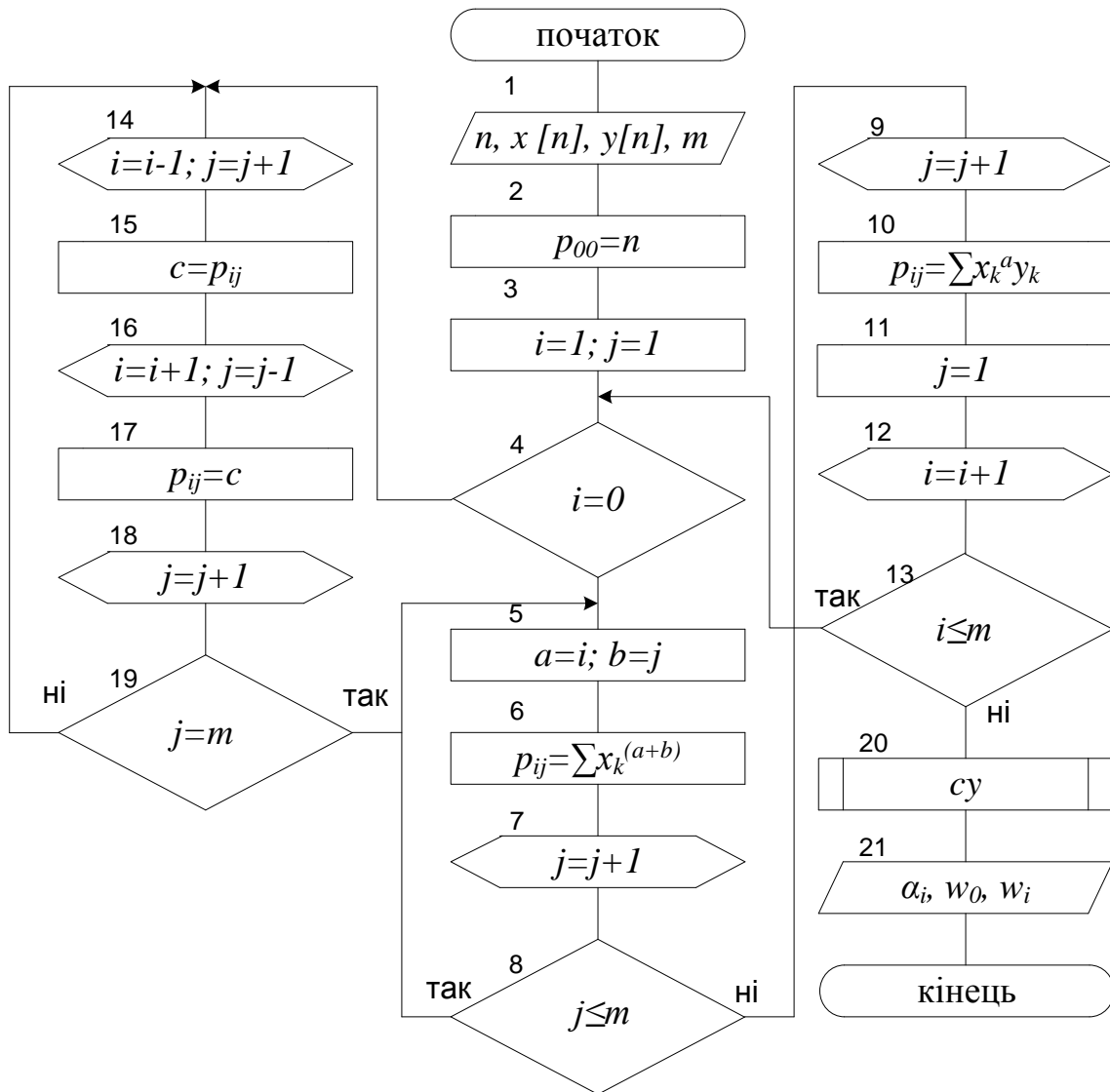


Рисунок 3.13 – Алгоритм навчання класифікатора типу SVM

Алгоритм складається з чотирьох основних кроків: введення вхідних даних; обчислення сум, що визначають коефіцієнти перед невідомим в системі (3.43); розв’язання одержаної системи та друк результатів, якими є числові значення множників Лагранжа α_i та коефіцієнтів w вагового вектора. Детальний вигляд алгоритма показано на рис. 3.13. Під час обчислення коефіцієнтів p_{ij} у разі невідомих в блоці 6 алгоритму використовується властива для систем (3.44) 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-ий рядок, останній і передостанній стовпці матриці системи, що показано в блоках 4 і 13. В блоках 1, 6 і 10 дії наведено в спрощеному вигляді. В програмі вони реалізуються у вигляді циклічних процесів. В алгоритмі використано такі позначення:

i, j – індекси, якими позначаються номер рядка і стовпця відповідно;
 k – індекс, що позначає номер елемента в масиві x або y ;
 p_y – коефіцієнти, які замінюють відповідні суми в системі рівнянь (3.44).

Після знаходження множників α_i Лагранжа алгоритмом обчислюється за формулою (3.44) ваговий вектор \vec{w} , що визначає в просторі ознак кут нахилу гіперплощини. Величина зміщення w_0 оптимальної гіперплощини обчислюється за всіма граничними опорними векторами як медіана з:

$$w_0 = \text{med} \{ (\vec{w}, \vec{x}_i) - y_i : \alpha_i > 0, i = \overline{1, k} \}. \quad (3.45)$$

На сьогодні проблема алгоритмізації і програмування класифікатора SVM не є актуальною, оскільки існує можливість застосувати його код з відкритих програмних ресурсів, наприклад, з бібліотеки SVMlib ресурсу Open CV мовою C++ або з бібліотеки Scikit Learn ресурсу GitHub мовою Python. Розглянемо особливості використання програмного коду цього класифікатора з бібліотеки Scikit learn.

3.5.3 Налаштування параметрів SVM-класифікатора під час використання бібліотечних функцій мови Python

Під час створення класифікатора SVM мовою Python потрібно вибрати параметри, регламентовані бібліотекою Scikit Learn. В офіційній документації цієї бібліотеки описано такі параметри, які потрібно вибрати розробнику:

```
options:
-s svm_type: set type of SVM (default 0)
  0 -- C-SVC
  1 -- nu-SVC
  2 -- one-class SVM
  3 -- epsilon-SVR
  4 -- nu-SVR
-t kernel_type: set type of kernel function (default 2)
  0 -- linear: u'*v
  1 -- polynomial: (gamma*u'*v + coef0)^degree
  2 -- radial basis function: exp(-gamma*|u-v|^2)
  3 -- sigmoid: tanh(gamma*u'*v + coef0)
-d degree: set degree in kernel function (default 3)
-g gamma: set gamma in kernel function (default
1/num_features)
-r coef0: set coef0 in kernel function (default 0)
```



```
-c cost: set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)
-n nu: set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)
-p epsilon: set the epsilon in loss function of epsilon-SVR (default 0.1)
-m cachesize: set cache memory size in MB (default 100)
-e epsilon: set tolerance of termination criterion (default 0.001)
-h shrinking: whether to use the shrinking heuristics, 0 or 1 (default 1)
-b probability_estimates: whether to train a SVC or SVR model for probability estimates, 0 or 1 (default 0)
-wi weight: set the parameter C of class i to weight*C, for C-SVC (default 1)
```

Параметр `-s` задає один із п'яти можливих типів класифікатора, параметр `-t` – один із чотирьох видів функції ядра, що визначає вигляд розподільної поверхні, і ще чотири параметри (`-d`, `-g`, `-r`, `-c`) задають властивості вибраної функції ядра.

За формою функції помилки, моделі SVM можна розділити на 5 окремих типів (позначимо їх номерами відповідно до їх опису в документації бібліотеки SciKit Learn). Вибір тієї чи іншої моделі класифікації залежить від поставленої задачі і типу даних, які в ній використовуються.

Якщо стоїть задача класифікації на два класи, тобто потрібно спрогнозувати, марку якого з класів необхідно присвоїти якомусь елементу даних з заданої множини, то потрібно використовувати C класифікатор (0: C-SVC) або nu-класифікатор (1: nu-SVC, ν -SVC). Різниця між класифікаторами C-SVC і nu-SVC полягає у використанні параметра C у першому випадку і параметра nu (ν) у другому. Діапазон C змінюється від нуля до нескінченності, а діапазон nu лежить в межах [0..1]. Характерною властивістю nu є те, що цей параметр пов'язаний зі співвідношенням кількості опорних векторів з величиною помилки навчання.

Якщо стоїть задача виявлення нехарактерних прецедентів (викидів) тільки в одному класі даних, то використовується модель однокласової класифікації 2: one-class SVM.

Моделі класифікації 3: epsilon-SVR і 4: nu-SVR використовуються для задач регресії, де потрібно передбачити безперервне число, наприклад, ціну житла.

Оскільки в цьому розділі предметом розгляду є класифікатори детермінованого типу, а задача регресії розглядається в інших розділах, то більш детально проаналізуємо принципи побудови перших двох типів SVM класифікаторів.

0 - Класифікація SVM Type 0 (також відома як C-SVM класифікація). Процедура навчання такого типу класифікатора передбачає мінімізацію функції помилки:

$$\frac{1}{2} \vec{w}^T \vec{w} + C \sum_{i=1}^N \xi_i \quad (3.46)$$

з урахуванням обмежень:

$$y_i (\vec{w}^T \phi(\vec{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, i = 1, \dots, N. \quad (3.47)$$

У виразі (3.46) C – константа «потужності» штрафу, параметр регуляризації. Цей параметр контролює компроміс між плавністю розподільної поверхні і правильністю класифікації навчальних прецедентів. Для великих значень C оптимізація вибирає гіперплощину з меншою маржою для отримання правильної класифікації всіх навчальних точок. Навпаки, дуже мале значення C заставить процедуру оптимізації шукати гіперплощину з більшою розподільною маржою, навіть якщо вона неправильно класифікує більше точок. Тобто, чим більше C , тим більша міра «покарання» помилки. Відштовхуючись від цих міркувань, C потрібно вибирати з обережністю, щоб уникнути надмірності.

На рис. 3.14 показано приклад двох різних значень параметра регуляризації C для заданої множини навчальних даних, поданих точками двох класів у двовимірному просторі ознак. На рис. 3.14, а) для невеликої величини параметра регуляризації C маємо просту лінійну розподільну гіперплощину, але неправильну класифікацію двох точок навчальної вибірки, а на рис. 3.14, б) більше значення параметра C дає правильну класифікацію всіх точок, але більш складний вигляд розподільної гіперплощини.

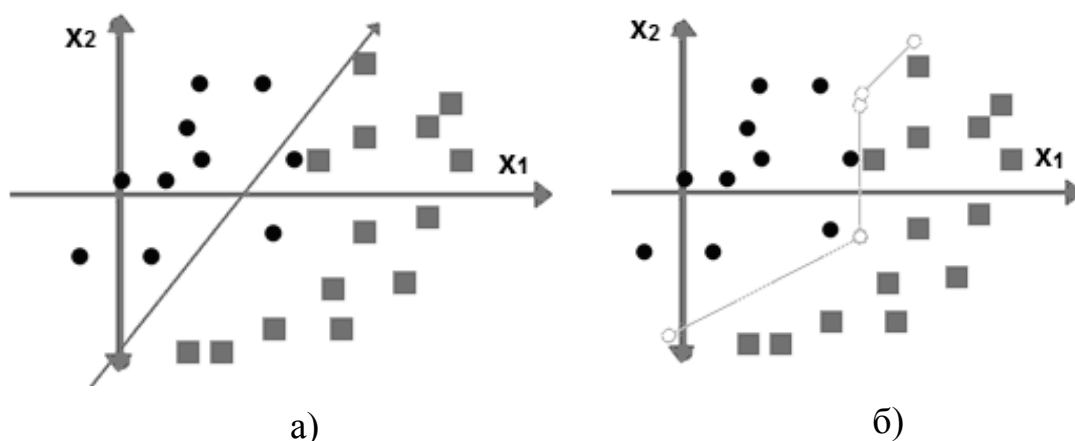


Рисунок 3.14 – Вигляд розподільної границі за різних значень C

Параметр ξ_i у виразі (3.46) визначає величину помилки класифікації, тобто кількість неправильно класифікованих елементів навчальної вибірки, а індекс i задає номер одного із N елементів навчальної вибірки.

Вектор вагових коефіцієнтів \vec{w} і константа b у виразі (3.47) визначають відповідно нахил і зсув розподільної гіперплощини в просторі ознак (див. розділ 3.2). Як було попередньо сказано, у цьому виразі величина y задає собою мітки класу $y \in \pm 1$, а \vec{x}_i – вектори зображень навчальної вибірки в цьому виразі є незалежними змінними. Функція ядра ϕ використовується для перетворення даних з вхідного простору незалежних змінних в простір об'єктів, в якому нелінійна розподільна поверхня перетворюється в лінійну.

1 – Класифікація SVM Type 1 (також відома як класифікація ν -SVM). Функція помилки, яку мінімізує класифікатор цього типу, має вигляд:

$$\frac{1}{2} \vec{w}^T \vec{w} - \nu \rho + \frac{1}{N} \sum_{i=1}^N \xi_i, \quad (3.48)$$

з урахуванням обмежень:

$$y_i (\vec{w}^T \phi(\vec{x}_i) + b) \geq \rho - \xi_i; \xi_i \geq 0, i = 1, \dots, N; \rho \geq 0. \quad (3.49)$$

Цей варіант класифікатора є модифікацією попереднього варіанта, який буде більш гнучке значення оптимальної гіперплощини з урахуванням кількості екземплярів навчальної вибірки, розміщених в області чужих класів. В цьому варіанті жорстке значення параметра штрафу C замінено на параметр $\nu \in [0, 1]$, який є нижньою та верхньою межею кількості точок, що є опорними векторами і лежать відповідно на неправильній стороні гіперплощини. Додатково в моделі також введено параметр оптимізації ρ , значення якого стає зрозумілим з огляду на те, що за умови рівності нулю помилки класифікації, $\xi_i = 0$, обмеження (3.49)

просто стверджує, що два класи розділені маржою $\frac{2\rho}{\|\vec{w}\|}$.

Ефективність методу опорних векторів також залежить від вибору функції ядра. Як уже було сказано, функції ядра дозволяють за нелінійних меж між класами у вихідному просторі ознак перейти до лінійного варіанта розподільної гіперплощини у просторі вищої розмірності, що відповідає використанню узагальнених вирішувальних функцій (3.18). Для розуміння ролі функції ядра в перетворенні нелінійної задачі класифікації в лінійну розглянемо наступний приклад.

Приклад 3.8. На рис. 3.15, а) наведено вихідний двовимірний ознаковий простір, в якому розміщено зображення об'єктів двох класів:

1-ий клас зображено трикутниками, 2-ий клас – кружечками. Побудувати лінійний класифікатор SVM для цих об'єктів.

Розв'язування: з рис. 3.15, б) очевидно, що цей набір даних не можна розділити лінійно. Ми не можемо накреслити пряму лінію, яка б класифікувала ці дані, тому що розподільна лінія являє собою коло, яке можна описати формулою $k=x_1^2+x_2^2$.

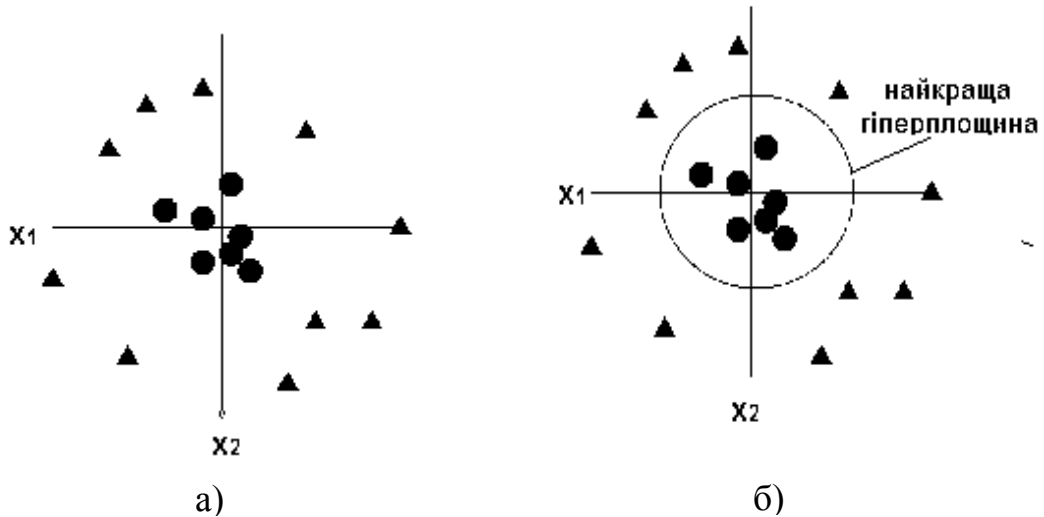


Рисунок 3.15 – Приклад двох класів, що не розділяються лінійно

Але цей набір даних можна розділити лінійно в просторі вищої розмірності, додавши додаткову ознаку z , визначивши її як $z=x_1^2+x_2^2$. Отримаємо тривимірний простір з третьою координатою z , яка подана квадратом відстані точки до початку осі. Цей прийом відповідає введенню квадратичної узагальнювальної функції, описаної в (3.21) підрозділі 3.3. На рис. 3.16 наведено візуалізацію вихідного набору даних в отриманому тривимірному просторі ознак.

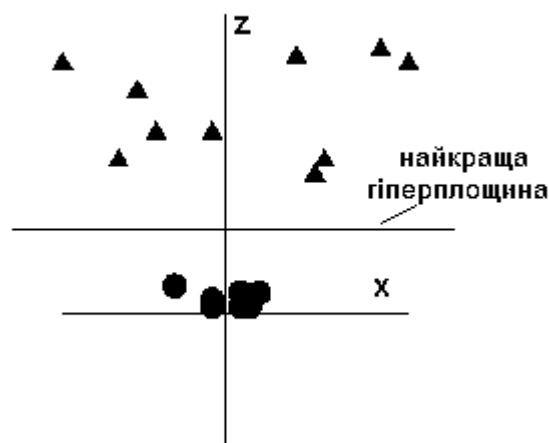


Рисунок 3.16 – Поділ двох класів прикладу 3.7 лінійною гіперплощиною

Поставлена в прикладі 3.8 задача вирішена. Тепер дані можна поділити лінійно. Результат – ми можемо класифікувати нелінійний набір

даних, додавши до нього додатковий вимір, а потім привести назад до вихідного виду, використовуючи математичне перетворення.

Гіперплощина у отриманому багатовимірному просторі визначається як набір точок, для яких скалярний добуток з вектором у цьому просторі є константою. Розглянемо, як виглядає скалярний добуток двох векторів \vec{a} і \vec{b} з координатами (x_{1a}, x_{2a}, z_a) і (x_{1b}, x_{2b}, z_b) в отриманому тривимірному просторі p з третьою координатою $z=x_1^2+x_2^2$:

$$(\vec{a} \cdot \vec{b}) = x_{1a}x_{1b} + x_{2a}x_{2b} + z_a z_b,$$

або

$$(\vec{a} \cdot \vec{b}) = x_{1a}x_{1b} + x_{2a}x_{2b} + (x_{1a} + x_{2a})^2 (x_{1b} + x_{2b})^2. \quad (3.50)$$

Описаний тут математичний прийом називається трюком ядра (kernel trick).

Існує низка функцій ядра $K(\vec{x}_i, \vec{x}_j)$, що використовуються в моделях класифікатора SVM – лінійна, поліноміальна, радіально-базисна, сигмоїдна:

$$K(\vec{x}_i, \vec{x}_j) = \begin{cases} \vec{x}_i \cdot \vec{x}_j & \text{Linear} \\ (\gamma \vec{x}_i \cdot \vec{x}_j + C)^2 & \text{Polinomial} \\ e^{-\gamma |\vec{x}_i - \vec{x}_j|^2} & \text{RBF} \\ th(\gamma \vec{x}_i \cdot \vec{x}_j + C) & \text{Sigmoid} \end{cases}, \quad (3.51)$$

де \vec{x}_i, \vec{x}_j – вхідний і опорний вектори, відповідно;

$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$ – функція ядра, яка за допомогою скалярного добутку векторів зображень вхідного простору переводить їх в ознаковий простір вищої розмірності відображенням ϕ ;

γ (gamma) – налагоджуваний параметр для поліноміальної (polinomial), радіально-базисної (RBF) та сигмоїдної (sigmoid) функцій. Параметр gamma визначає наскільки суттєво кожен із елементів навчальної вибірки даних має вплив на вигляд оптимальної гіперплощини. Чим нижче значення цього параметра, тим більше елементів, навіть досить далеких від розподільної гіперплощини, беруть участь у процесі її побудови. За високого значення gamma алгоритм «спиратиметься» тільки на ті елементи, які найбільш близькі до самої гіперплощини.

Після вибору типу SVM і функції ядра на вибраній моделі для заданого набору даних налаштовують насамперед експериментальним

шляхом значення параметрів C (ν) і γ (gamma). Найкращу комбінацію їхніх значень зазвичай вибирають шляхом пошуку по шкалі з експоненційним зростанням чисел C і γ . Наприклад,

$$C \in \{2^{-5}, 2^{-2}, \dots, 2^{13}, 2^{15}\}, \quad (3.52)$$

$$\gamma \in \{2^{-15}, 2^{-13}, \dots, 2^1, 2^3\}. \quad (3.53)$$

Для цього навчальну вибірку даних поділяють на непересічні множини даних і виконують так звану крос-перевірку шляхом тестування роботи класифікатора на кожній з цих підмножин, використовуючи різні комбінації значень C і γ . З цих комбінацій вибирається та, яка проявила себе найкраще на крос-перевірці.

Фінальна модель, яка використовує для тестування та класифікації нових даних, навчається потім на всій множині з використанням вибраних параметрів.

Нижче наведено загальну структуру програмного коду використання методу SVM мовою Python.

3.5.4 Приклад впровадження класифікатора SVM мовою Python

Для впровадження алгоритмів машинного навчання в програмах мовою Python потрібно імпортувати бібліотеку scikit-learn. В цій бібліотеці є також доступним метод SVM. Для його використання в програмі виконуються такі основні кроки: імпорт бібліотеки, створення набору об'єктів, створення моделі на наборі об'єктів та класифікація за допомогою отриманої моделі на тестовій вибірці об'єктів. Глобальну структуру такої програми подано нижче:

```
#Імпорт бібліотеки scikit-learn
from sklearn import svm
#Введення навчальної вибірки даних у вигляді об'єктів і марок
класів
x_test(predictor) of test_dataset, Y target
# Вибір моделі класифікатора SVM
model = svm.svc(kernel='linear', c=1, gamma=1) # Вибір параметрів
# моделі (ядро = 'лінійний', c = 1, гама = 1)
# Навчання моделі на навчальній вибірці (X,Y) і оцінка її якості
model.fit(X, y)
model.score(X, y)
# Прогнозування результату (класифікація)
predicted= model.predict(x_test)
```

Далі наведено код діючої програми для реалізації методу SVM мовою Python. Символом # починаються рядки коментарів. Як навчальну вибірку вибрано еталонний файл даних iris.dat, в якому для опису ірисів взято тільки дві перших ознаки.

```

import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm, datasets
# імпорт файлу даних ірисів для роботи з ним
iris = datasets.load_iris()
X = iris.data[:, :2] # Для кращої наочності графіків беремо
# тільки дві перших ознаки
y = iris.target
# Обираємо модель SVM і підбираємо дані. Не масштабуємо їх на
# цьому кроці, оскільки ми хочемо побудувати опорні вектори
C = 1.0 # SVM regularization parameter
svc = svm.SVC(kernel='linear', C=1, gamma=0).fit(X, y)
# створюємо сітку для накреслення графіка
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
h = (x_max / x_min)/100
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
    np.arange(y_min, y_max, h))
plt.subplot(1, 1, 1)
Z = svc.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
plt.contourf(xx, yy, Z, cmap=plt.cm.Paired, alpha=0.8)
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Paired)
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')
plt.xlim(xx.min(), xx.max())
plt.title('SVC with linear kernel')
plt.show ()

```

Контрольні питання та завдання

1. Зарисуйте структурну схему лінійного класифікатора.
2. Назвіть методи вирішення задачі класифікації.
3. Які принципи використовуються в евристичному підході до вирішення задачі класифікації?
4. Які принципи використовуються в математичному підході до вирішення задачі класифікації?
5. На які типи поділяються математичні методи класифікації образів?
6. Запишіть загальний вигляд лінійної вирішувальної функції і назвіть її властивості.
7. Запишіть формули одиничної нормалі до площини, відстані від початку координат до гіперплощини і відстані від заданої точки простору до гіперплощини.
8. Запишіть формулу для визначення дихотомізаційної потужності лінійних функцій для випадку «добре» розташованих точок.
9. Запишіть правило класифікації об'єкта для випадку 1, коли всі класи розподіляються окремими вирішувальними функціями.

10. Запишіть правило класифікації об'єкта для випадку 2, коли всі класи розподіляються окремими вирішувальними функціями, а класи є попарно роздільними.

11. Запишіть правило класифікації об'єкта для випадку 3.

12. Який вигляд має формула узагальненої вирішувальної функції? Які її складові і який їх зміст?

13. Знайти відстань від точки $\vec{x} = (4; 7; 1)$ до площини $d(x) = 2x_1 + 4x_2 - 5x_3 + 3$.

14. Визначити, до якого з двох класів відноситься точка $\vec{x} = (-5; -2; 3; 1)$, якщо розподільна поверхня має вигляд $d(\vec{x}) = -3x_1 + 2x_2 + 3x_3 + 5x_4 - 6$.

15. Запишіть формулу квадратичної вирішувальної функції для n -вимірного випадку.

16. Запишіть вигляд узагальненої вирішувальної функції третього степеня для двовимірного простору за рекурентною формулою 3.22.

17. Знайти еталон кластера, що містить такі точки: $\vec{x}_1 = (0; 1; 1)$; $\vec{x}_2 = (2; 2; 1)$; $\vec{x}_3 = (1; 0; 0)$; $\vec{x}_4 = (1; 2; 1)$; $\vec{x}_5 = (3; 2; 1)$; $\vec{x}_6 = (1; 1; 2)$.

18. Визначте вирішувальну функцію $d_i(\vec{x})$, що забезпечує правильну класифікацію образів, зображених на рис. 3.17.

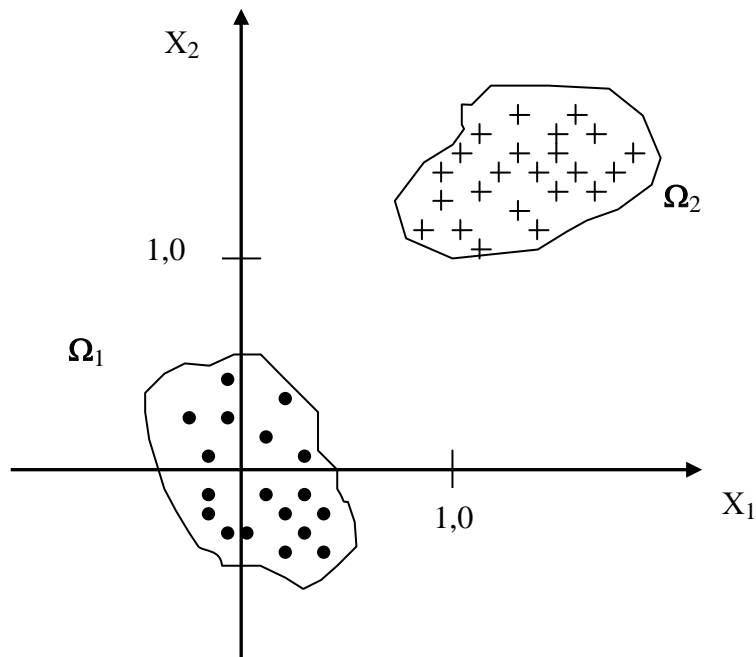


Рисунок 3.17 – Приклад розміщення образів у двовимірному просторі ознак

19. Покажіть графічно залежність найденої в п. 6 вирішувальної функції від x_1 та x_2 за умови $x_1 \geq 0$ і $x_2 \geq 0$.

20. Нехай в задачі розпізнавання образів, що передбачає розбиття на 10 класів, три класи відповідають випадку 1, а інші – випадку 2 (підрозділ 3.2). Чому дорівнює мінімальне число вирішувальних функцій, необхідних для розв’язання цієї задачі?

21. Записати рівняння одиничної нормалі до гіперплощини, рівняння якої $d(\vec{x}) = 2x_1 + 3x_2 - x_3 + 5$.

22. Визначити відстань від точки $\vec{x} = (5;4;-3)$ до площини, рівняння якої $d(\vec{x}) = 2x_1 + 3x_2 - x_3 + 5$.

23. Визначити відстань від заданої гіперплощини (п. 9) до початку координат.

24. Нехай для задачі розподілу на три класи задано такі вирішувальні функції:

$$d_1(\vec{x}) = -\vec{x}_1; \quad d_2(\vec{x}) = x_1 + x_2 - 1; \quad d_3(\vec{x}) = x_1 - x_2 - 1.$$

Накресліть розподільні границі і області, відповідні кожному класу зображень за умови, що ці функції визначені:

а) для випадку 1 розподілу на декілька класів;

б) для випадку 2 за таких умов:

$$d_{12}(\vec{x}) = d_1(\vec{x}); \quad d_{13}(\vec{x}) = d_2(\vec{x}); \quad d_{23}(\vec{x}) = d_3(\vec{x});$$

в) для випадку 3.

25. Нехай в тривимірному просторі задано два класи, кожен з яких містить по п’ять зображень. Яка мінімальна кількість коефіцієнтів потрібна для реалізації поліноміальної дискримінантної функції, що забезпечує розподіл цих класів незалежно від геометрії їх взаємного розташування, але за умови хорошого розміщення зображень?

26. Яка в загальному випадку імовірність лінійної розподілюваності на два класи вибірки з п’яти «добре» розміщених двовимірних зображень?

27. Покажіть, що за використання класифікації двох класів за критерієм мінімуму відстані, розподільною границею є гіперплощина, рівновіддалена від центрів класів \vec{z}_1 і \vec{z}_2 .

28. Визначити границі, що розділяють три класи в двовимірному евклідовому просторі за допомогою класифікації за критерієм мінімуму відстані. Класи містять такі вибіркові точки:

$$\Omega_1 : \{\vec{x}_1 = (-4;0), \vec{x}_2 = (-3;1), \vec{x}_3 = (-2;0), \vec{x}_4 = (-2;-1)\};$$

$$\Omega_2 : \{\vec{x}_5 = (2;4), \vec{x}_6 = (2;3), \vec{x}_7 = (3;3)\};$$

$$\Omega_3 : \{\vec{x}_8 = (3;-1), \vec{x}_9 = (4;0), \vec{x}_{10} = (4;-2), \vec{x}_{11} = (5;-1)\}.$$

29. Запишіть формулу дихотомізаційної потужності узагальненої вирішувальної функції.

30. Запишіть формулу Евклідової відстані і вкажіть її властивості.

31. Запишіть формулу вирішувальної функції для класифікатора за мінімумом відстані.

32. Що таке машина опорних векторів SVM (Support Vector Mashine)?

33. Назвіть переваги і недоліки методу SVM.
34. Для розв'язання яких задач використовується метод SVM?
35. В чому полягає суть роботи класифікатора типу SVM?
36. Як називається розподільна поверхня між класами в класифікаторі SVM?
37. Що називають маржою в класифікаторі SVM?
38. Проілюструйте геометрично суть роботи SVM.
39. Як формулюється задача навчання в методі SVM?
40. Для чого виконується масштабування даних під час навчання SVM?
41. Як виконується пошук оптимальних опорних точок для SVM?
42. Які види моделей SVM Вам відомі?
43. Які параметри регуляризації використовують в моделях SVM?
44. Що таке функція ядра і які ядра застосовують в SVM?
45. Яке призначення параметра регуляризації C і в якому діапазоні вибирають його числові значення?
46. На що впливає величина параметра Γ і в яких межах вибирається його значення?

РОЗДІЛ 4 ІМОВІРНІСНИЙ ПІДХІД ДО КЛАСИФІКАЦІЇ ОБРАЗІВ

4.1 Класифікація образів як задача теорії статистичних рішень

В тому випадку, коли між ознаками об'єктів і класами, до яких ці об'єкти можуть бути віднесені, існують імовірнісні зв'язки, побудова алгоритмів класифікації може ґрунтуватися на положеннях теорії статистичних рішень. Розглянемо коротко основні її ідеї.

Припустимо, що спостерігається деякий об'єкт, описаний вектором \vec{x} в просторі імовірнісних ознак. Цей об'єкт може належати, наприклад, тільки одному із двох образів (рис. 4.1).

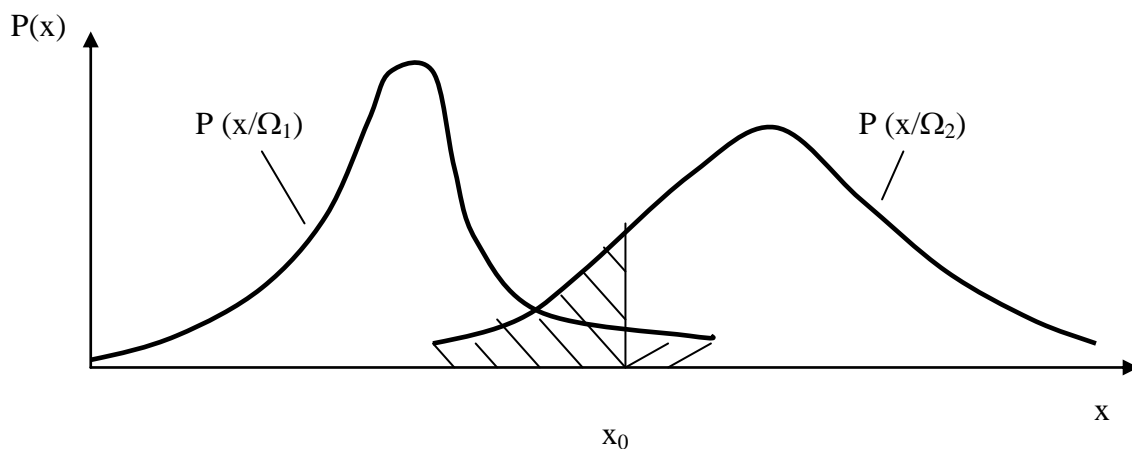


Рисунок 4.1 – Функції щільності розподілу ймовірностей ознаки x для образів Ω_1 і Ω_2

Теорія статистичних рішень дозволяє знайти метод, який після вивчення об'єкта дав би з мінімальною ймовірністю помилки відповідь на те, до якого з двох образів, Ω_1 чи Ω_2 – відноситься конкретне значення вектора \vec{x} (наприклад, x_0 на рис. 4.1).

В процесі прийняття рішення розрізняють помилки подвійного гатунку. Їх називають помилками першого і другого роду. Будемо вважати, що існує дві гіпотези: *гіпотеза 1*, що об'єкт належить першому образу Ω_1 , і *гіпотеза 2*, що цей об'єкт міститься в другому образі Ω_2 . Тоді вважається, що здійснюється помилка першого роду, коли відхиляється перша гіпотеза, в той час як вона дійсно має місце, і помилка другого роду, якщо приймається перша гіпотеза тоді, коли справедливою є друга.

Для прикладу на рис. 4.1 помилка першого роду чисельно дорівнює

$$e_1 = \int_{x_0}^{+\infty} p(x/\Omega_1) dx, \quad (4.1)$$

а другого роду

$$e_2 = \int_{-\infty}^{x_0} p(x/\Omega_2) dx. \quad (4.1, a)$$

Часто помилку першого роду називають пропуском цілі, а помилку другого роду – хибною тривоگو. З власного досвіду дуже легко впевнитися, що ці два види помилок часто не є рівноцінними для тих, хто приймає рішення. Наприклад, нехай образ Ω_1 відповідає сигналам землетрусу, а образ Ω_2 – сигналам навантаженого поїзда. Зрозуміло, що витрати від того, що сигнал землетрусу буде класифікований сейсмологами як гуркіт навантаженого поїзда (пропуск цілі), будуть набагато вищі, ніж у разі хибної тривоги.

Згідно з теорією статистичних рішень кожен раз потрібно знайти таке правило, яке б мінімізувало ризик або середню вартість прийняття рішень, що визначається в вигляді суми ймовірності помилки першого роду, помноженої на свою вартість, і ймовірності помилки другого роду, помноженої на свою вартість.

З такого погляду процес прийняття рішень на класифікацію зручно розглядати як гру статистичного характеру, яку класифікатор системи розпізнавання образів проводить з природою. Формально гра \mathbf{G} задається набором з трьох елементів :

$$\mathbf{G} = (\mathbf{X}, \mathbf{\Omega}, \mathbf{C}), \quad (4.2)$$

де $\mathbf{X} = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_Q\}$ – множина стратегій природи, причому \vec{x}_i – зображення об'єкта ω_i в ознаковому просторі;

$\mathbf{\Omega} = \{\Omega_1, \Omega_2, \dots, \Omega_K\}$ – множина стратегій класифікатора, де Ω_j – клас, до якого класифікатор відносить подане природою зображення \vec{x}_i ;

$\mathbf{C} = \|c_{ij}\|$ – платіжна матриця (або матриця втрат чи штрафів) гри, де $c_{ij} = c(\vec{x}_i, \Omega_j)$ – втрати класифікатора за віднесення зображення \vec{x}_i до класу Ω_j .

В такій постановці гра вважається грою з нульовою сумою, тобто втрати одного учасника точно дорівнюють виграшу іншого учасника. Також вважають додатні значення втрат істинними втратами, а нульові – виграшем. Нульові втрати відповідають правильній класифікації, тобто діагональним елементам c_{ij} платіжної матриці \mathbf{C} .

В іграх такого типу використовуються різні стратегії, зокрема байєсівська стратегія, мінімаксна стратегія і стратегія Неймана-Пірсона.

Задача класифікатора полягає в тому, щоб знайти таке оптимальне рішення, яке забезпечить мінімізацію середнього ризику чи вартості втрат.

Мінімаксна стратегія прийняття рішень застосовується тоді, коли апіорні ймовірності появи об'єктів відповідних класів невідомі. Мінімізувати значення середнього ризику прийняття рішень на основі байєсівської стратегії в цьому випадку не є можливим. В цій ситуації раціонально використовувати критерій, який мінімізує максимально можливе значення середнього ризику. Цей критерій називають мінімаксом критерієм.

В ситуаціях, коли апіорну ймовірність появи образів різних класів невідома і невідомі штрафи за помилки 1-го і 2-го роду, застосовують критерій Неймана-Пірсона.

Гра, що розглядається, називається статистичною, оскільки природа вибирає стратегії не усвідомлено, а ґрунтуючись на ймовірностях класів $p(\Omega_i)$, $i = 1, 2, \dots, K$. Вона їх притримується, не дивлячись на їх можливу неоптимальність з погляду максимізації виграшу. Наприклад, звуки мови, як і літери в тексті, зустрічаються з відомою ймовірністю.

Класифікатор має можливість отримати додаткову інформацію про противника у образі природи шляхом проведення експериментів за множиною зображень. Внаслідок цього в більшості випадків йому відомо апіорну інформацію про стратегію природи у вигляді ймовірностей появи образів $p(\Omega_i)$ і функцій правдоподібності $p(\vec{x}/\Omega_i)$ образів Ω_i . Цю інформацію класифікатор використовує для покращення своєї гри, тобто для підвищення точності розпізнавання.

Нехай під час реалізації гри між природою і класифікатором природа вибирає клас Ω_i і відтворює зображення \vec{x}_i . Ймовірність належності \vec{x}_i до класу Ω_i визначається як $p(\Omega_i/\vec{x})$. Якщо класифікатор приймає рішення про те, що $\vec{x} \in \Omega_j$ в тому випадку, коли в дійсності $\vec{x} \in \Omega_i$, то він має втрати c_{ij} . Оскільки образ \vec{x}_i може належати до будь-якого з K класів, то математичне сподівання втрат, пов'язане з цим рішенням, визначиться таким виразом :

$$R_j(\vec{x}) = \sum_{i=1}^K c_{ij} * p(\Omega_i/\vec{x}) . \quad (4.3)$$

В теорії статистичних рішень цю величину часто називають *умовним середнім ризиком* або умовними середніми втратами.

4. 2 Байєсівська класифікація образів

Внаслідок розпізнавання кожного образу, що подається природою, класифікатор може зарахувати його до одного із K можливих класів.

Оскільки для кожного зображення \bar{x}_i обчислюються значення умовних середніх втрат $R_1(\bar{x}), R_2(\bar{x}), \dots, R_K(\bar{x})$ і класифікатор зараховує його до того класу, якому відповідають найменші умовні втрати, то і математичне сподівання повних втрат на множині всіх рішень також буде мінімізовано. Класифікатор, що мінімізує математичне сподівання загальних втрат, називається *Байєсівським класифікатором*. Він, з погляду статистики, відповідає оптимальній якості класифікації.

Вираз (4.4) можна подати в іншому вигляді, скориставшись формулою Байєса

$$p(\Omega_i / \bar{x}) = \frac{p(\Omega_i)p(\bar{x} / \Omega_i)}{p(\bar{x})}, \quad (4.4)$$

для чого підставимо (4.4) в (4.3) замість $p(\Omega_i / \bar{x})$:

$$R_j(\bar{x}) = \frac{1}{p(\bar{x})} \sum_{i=1}^K c_{ij} p(\Omega_i)p(\bar{x} / \Omega_i), \quad (4.5)$$

де $p(\bar{x} / \Omega_i)$ – функція правдоподібності для класу Ω_i .

Оскільки вираз $1/p(\bar{x})$ входить до всіх формул обчислення умовних середніх втрат $R_j(\bar{x}), j = 1, 2, \dots, K$ спільним множником, вираз (4.5) можна на нього скоротити. Одержимо таку формулу втрат

$$R_j(\bar{x}) = \sum_{i=1}^K c_{ij} p(\bar{x} / \Omega_i)p(\Omega_i) \quad (4.6)$$

Для випадку двох класів підрахуємо втрати $R_1(\bar{x})$ і $R_2(\bar{x})$.

В процесі вибору стратегії 1, тобто за внесення \bar{x} до Ω_1 , середні втрати:

$$R_1(\bar{x}) = c_{11}p(\bar{x} / \Omega_1)p(\Omega_1) + c_{21}p(\bar{x} / \Omega_2) \cdot p(\Omega_2), \quad (4.7)$$

а у разі вибору стратегії 2:

$$R_2(\bar{x}) = c_{12} \cdot p(\bar{x} / \Omega_1)p(\Omega_1) + c_{22} \cdot p(\bar{x} / \Omega_2) \cdot p(\Omega_2). \quad (4.8)$$

Класифікатор Байєса зараховує \bar{x} до класу Ω_1 , якщо $R_1(\bar{x}) < R_2(\bar{x})$, тобто:

$$c_{11} \cdot p(\bar{x} / \Omega_1)p(\Omega_1) + c_{21}p(\bar{x} / \Omega_2) \cdot p(\Omega_2) < c_{12}p(\bar{x} / \Omega_1)p(\Omega_1) + c_{22}p(\bar{x} / \Omega_2)p(\Omega_2)$$

або, що одно й те саме,

$$p(\bar{x}/\Omega_1)p(\Omega_1)(c_{12} - c_{11}) > p(\bar{x}/\Omega_2) \cdot p(\Omega_2)(c_{21} - c_{22}). \quad (4.9)$$

Перепишемо (4.9) у вигляді

$$\frac{p(\bar{x}/\Omega_1)}{p(\bar{x}/\Omega_2)} > \frac{p(\Omega_2)(c_{21} - c_{22})}{p(\Omega_1)(c_{12} - c_{11})} \quad (4.10)$$

Ліву частину виразу (4.10) називають відношенням правдоподібності

$$\ell_{12}(\bar{x}) = \frac{p(\bar{x}/\Omega_1)}{p(\bar{x}/\Omega_2)}, \quad (4.11)$$

оскільки воно є відношенням двох функцій правдоподібності. Таким чином, для кількості класів $K=2$ байєсівське вирішувальне правило формулюється таким чином :

$$\bar{x} \in \Omega_1, \text{ якщо } \ell_{12}(\bar{x}) > \theta_{12},$$

$$\bar{x} \in \Omega_2, \text{ якщо } \ell_{12}(\bar{x}) < \theta_{12},$$

і рішення вибирається довільно, якщо $\ell_{12}(\bar{x}) = \theta_{12}$.

Величина θ_{12} називається порогом і визначається виразом

$$\theta_{12} = \frac{p(\Omega_2)}{p(\Omega_1)} \cdot \frac{(c_{21} - c_{22})}{(c_{12} - c_{11})}. \quad (4.12)$$

Приклад 4.1. Знайти оптимальне вирішувальне правило для класифікатора сигналів, поданих у вигляді одиниць і нулів на виході каналу з шумом (рис. 4.2).

В інформаційному каналі на вхідний сигнал накладається гаусовський шум з нульовим середнім значенням і дисперсією σ^2 ; результат – на вхід класифікатора подається суміш сигналу з шумом x . Потрібно знайти оптимальне вирішувальне правило.

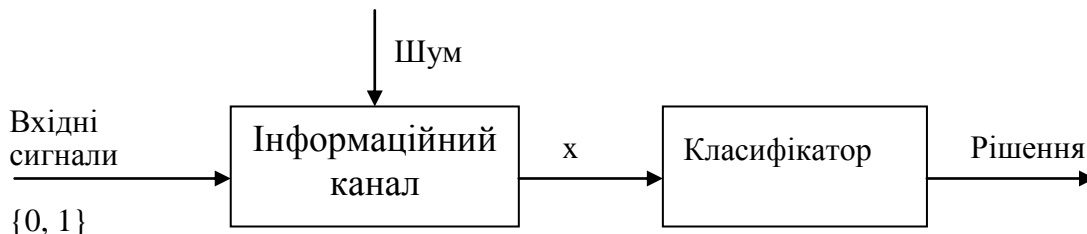


Рисунок 4.2 – Приклад простої задачі класифікації

Розв’язування: нехай Ω_1 – гіпотеза, що передано символ «0», а Ω_2 – символ «1». На основі сприйнятого сигналу x потрібно здійснити вибір

між Ω_1 і Ω_2 . Інтуїтивно можна вважати, що за значення $x < 0.5$ вибирається символ «0», за $x > 0.5$ – символ «1». Перевіримо зроблені припущення.

Нехай матриця втрат має вигляд

$$C = \begin{matrix} & a_1 & a_2 \\ \Omega_1 & \left\| \begin{matrix} 0 & c_{12} \end{matrix} \right\| \\ \Omega_2 & \left\| \begin{matrix} c_{21} & 0 \end{matrix} \right\| \end{matrix},$$

де a_1 і a_2 – рішення про те, що були передані символи 0 і 1, відповідно,
 c_{12} – втрати під час вибору рішення Ω_2 , коли істинний клас Ω_1 ,
 c_{21} – втрати під час вибору рішення Ω_1 , коли істинний клас Ω_2 .

З платіжної матриці видно, що правильним рішенням відповідають нульові втрати. Нехай $p(0)$ і $p(1)$ – апіорні ймовірності того, що передано символи 0 і 1, відповідно. Тоді порогове значення дорівнює

$$\theta_{12} = \frac{p(1)c_{21}}{p(0)c_{12}}.$$

Рішення, що передано 0, приймається в тому випадку, коли $\ell_{12}(x) > \theta_{12}$. Щоб знайти $\ell_{12}(x)$, знаходимо $p(x/\Omega_1)$ і $p(x/\Omega_2)$:

$$p(x/\Omega_1) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x-0)^2}{2\sigma^2}\right), \quad p(x/\Omega_2) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x-1)^2}{2\sigma^2}\right).$$

Підставивши ці значення в (4.11), одержимо значення функції правдоподібності

$$\ell_{12}(x) = \exp\left(\frac{1-2x}{2\sigma^2}\right).$$

Таким чином, якщо

$$\exp\left(\frac{1-2x}{2\sigma^2}\right) > \theta_{12} \quad \text{або} \quad x < 0.5 - \sigma^2 \ln \theta_{12},$$

то приймається рішення, що передано символ 0.

Цей результат збігається з інтуїтивними припущеннями в тому випадку, коли $\sigma^2 = 0$ або $\theta_{12} = 1$. Останнє справедливо за однакових величин $c_{21} = c_{12}$ платіжної матриці.

В загальному випадку розподілу на декілька класів ($K > 2$) зображення \vec{x} зараховується до класу Ω_i , якщо $R_i(\vec{x}) < R_j(\vec{x})$ для всіх $j \neq i, j = 1, 2, \dots, K$:

$$\sum_{k=1}^K c_{ki} p(\vec{x}/\Omega_k) p(\Omega_k) < \sum_{k=1}^K c_{kj} p(\vec{x}/\Omega_k) p(\Omega_k), \quad i, j = 1, 2, \dots, K; \quad i \neq j. \quad (4.13)$$

Співвідношення (4.13) можна використати для знаходження відношення правдоподібності (4.10) для загального випадку кількох класів, тобто $i = 1, 2, \dots, K$. Воно має вигляд:

$$\ell_{ij}(\vec{x}) = \frac{p(\vec{x}/\Omega_i)}{p(\vec{x}/\Omega_j)} = \frac{p(\Omega_j)(c_{ji} - c_{jj})}{p(\Omega_i)(c_{ij} - c_{ii})}, \quad (4.14)$$

$$\theta_{ij} = \frac{p(\Omega_j)(c_{ji} - c_{jj})}{p(\Omega_i)(c_{ij} - c_{ii})}. \quad (4.15)$$

Тоді правило класифікації вектора \vec{x} для випадку кількох класів має вигляд:

$$\vec{x} \in \Omega_i, \quad \text{якщо } \ell_{ij} > \theta_{ij} \quad \text{для всіх } j \neq i. \quad (4.16)$$

Більш зручною для класифікації в випадку кількох класів є функція втрат спеціального вигляду, отримати яку можна, зробивши декілька припущень.

Припущення 1. В більшості задач розпізнавання образів втрати від правильної класифікації дорівнюють нулю: $c_{ii} = 0, \quad i = 1, 2, \dots, K$.

Припущення 2. Втрати від неправильного рішення однакові: $c_{ij} = c_{kj}; \quad i, j, k, q = 1, 2, \dots, K; \quad i \neq j, \quad k \neq q$. Будемо також вважати, що всі втрати пронормовано, тобто $c_{ij} = 1$.

За таких припущень елемент матриці втрат c_{ij} можна записати у вигляді:

$$c_{ij} = 1 - \delta_{ij}, \quad (4.17)$$

де

$$\delta_{ij} = \begin{cases} 1, & \text{якщо } i = j \\ 0, & \text{якщо } i \neq j \end{cases}.$$

Підстановка виразу (4.17) в (4.6) приводить до виразу:

$$R_j(\vec{x}) = \sum_{i=1}^K (1 - \delta_{ij}) p(\vec{x} / \Omega_i) p(\Omega_i) = \sum_{i=1}^K p(\vec{x} / \Omega_i) p(\Omega_i) - p(\vec{x} / \Omega_j) p(\Omega_j). \quad (4.18)$$

Оскільки класи Ω_i , $i=1,2,\dots,K$ є незалежними подіями і утворюють повну групу подій, а $p(\vec{x} / \Omega_i)$ – умовна щільність ймовірності вектора ознак \vec{x} , то, відповідно з теорією ймовірності:

$$\sum_{i=1}^K p(\vec{x} / \Omega_i) p(\Omega_i) = p(\vec{x}). \quad (4.19)$$

Підставимо (4.19) в (4.18), одержимо:

$$R_j(\vec{x}) = p(\vec{x}) - p(\vec{x} / \Omega_j) p(\Omega_j).$$

Правило байєсівської класифікації тепер можна записати таким чином:

$$x \in \Omega_i, \text{ якщо } p(\vec{x}) - p(\vec{x} / \Omega_i) * p(\Omega_i) < p(\vec{x}) - p(\vec{x} / \Omega_j) * p(\Omega_j),$$

або

$$p(\vec{x} / \Omega_i) p(\Omega_i) > p(\vec{x} / \Omega_j) p(\Omega_j), \quad j=1,2,\dots,K, \quad j \neq i. \quad (4.20)$$

Якщо записати

$$d_i(\vec{x}) = p(\vec{x} / \Omega_i) p(\Omega_i), \quad i=1,2,\dots,K, \quad (4.21)$$

то правило (4.20) байєсівської класифікації еквівалентне вирішувальним функціям розділу 3, для яких $\vec{x} \in \Omega_i$, якщо $d_i(\vec{x}) > d_j(\vec{x})$ для всіх $j \neq i$.

Враховуючи те, що $p(\vec{x} / \Omega_i) p(\Omega_i) = p(\Omega_i / \vec{x}) p(\vec{x})$, вираз (4.21) можна переписати у вигляді:

$$d_i(\vec{x}) = p(\Omega_i / \vec{x}),$$

оскільки імовірність $p(\vec{x})$ не залежить від i .

У випадку (4.21) для класифікації використовується функція правдоподібності $p(\vec{x} / \Omega_i)$, а у випадку (4.22) використовується ймовірність $p(\Omega_i / \vec{x})$ належності зображення \vec{x} до образу Ω_i .

Одержані висновки дозволяють реалізувати схему розпізнавання на основі байєсівського класифікатора таким чином, як це показано на рисунку 4.3.

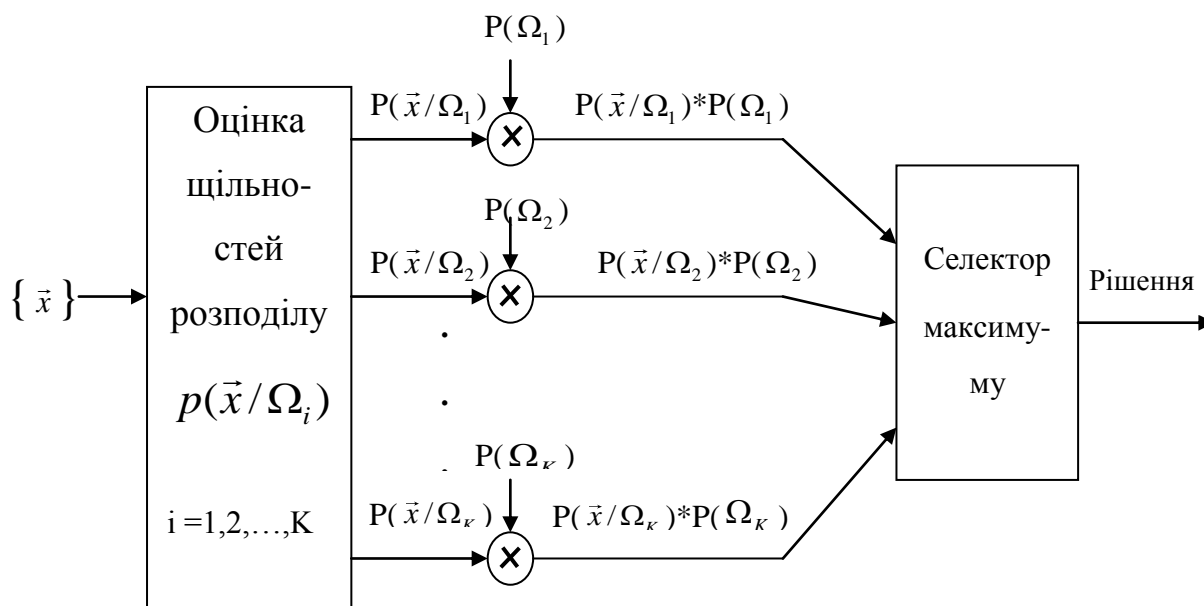


Рисунок 4.3 – Структурна схема байєсівського класифікатора

4.3 Байєсівський класифікатор для образів з нормальною щільністю розподілу ймовірностей ознак

Припущення про нормальний закон розподілу спрощує синтез вирішувальних функцій для ймовірнісних методів розпізнавання.

Щільність нормального розподілу випадкової змінної x

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right), \quad (4.22)$$

повністю визначається двома параметрами – середнім значенням m і дисперсією σ . Ці параметри визначаються таким чином:

$$m = E\{x\} = \int_{-\infty}^{\infty} x \cdot p(x) dx, \quad (4.23)$$

$$\sigma^2 = E\{(x-m)^2\} = \int_{-\infty}^{\infty} (x-m)^2 p(x) dx, \quad (4.24)$$

де $E\{\cdot\}$ – символ математичного сподівання. Для простоти щільність нормального розподілу записують як $p(x) \sim N(m, \sigma^2)$. Образи, що характеризуються нормальним розподілом, групуються відносно середнього значення m , а їх розкид пропорціональний середньоквадратичному відхиленню σ . Близько 95 % об'єктів, взятих із сукупності з нормальним законом розподілу, потраплять в інтервал 2σ в околі середнього значення m (рис. 4.4).

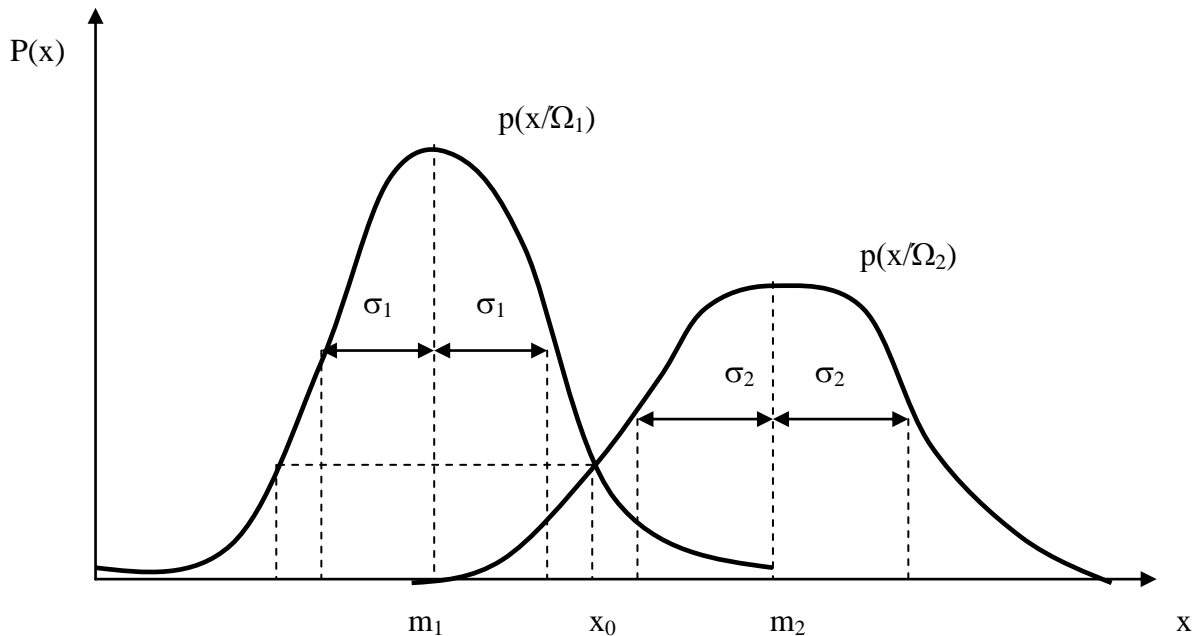


Рисунок 4.4 – Умовні щільності розподілу ймовірностей $p(x/\Omega_1)$ і $p(x/\Omega_2)$ значень ознаки образів для нормального закону

З рисунка 4.4 видно, що помилка класифікації буде дорівнювати сумі похибок 1-го і 2-го роду:

$$\varepsilon = \int_{x_0}^{+\infty} p(x/\Omega_1) dx + \int_{-\infty}^{x_0} p(x/\Omega_2) dx.$$

Для випадку K класів, що описуються багатовимірним щільностями нормального розподілу, вираз умовної щільності нормального розподілу випадкового вектора ознак для i -го класу запишеться:

$$p(\vec{x}/\Omega_i) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_i|}} \exp\left[-0.5(\vec{x} - \vec{m}_i)^T \Sigma_i^{-1} (\vec{x} - \vec{m}_i)\right] \quad (4.25)$$

Вектор середніх значень \vec{m}_i і коваріаційна матриця Σ_i задаються відповідно як

$$\bar{m}_i = E_i\{\bar{x}\}, \quad (4.26)$$

i

$$\Sigma_i = E_i\{(\bar{x} - \bar{m}_i)(\bar{x} - \bar{m}_i)^T\}, \quad (4.27)$$

де $\bar{x} = (x_1, x_2, \dots, x_n)$ і $\bar{m} = (m_1, m_2, \dots, m_K)$; $E_i\{\bar{x}\}$ – оператор математичного сподівання, визначений на зображеннях класу Ω_i ;

n – розмірність вектора образів;

$|\Sigma_i|$ – детермінант (визначник) коваріаційної матриці Σ_i .

Коваріаційна матриця Σ_i є симетричною і додатно напіввизначеною. Її діагональний елемент σ_{kk} є дисперсією k -ої координати вектора зображення. Елемент σ_{jk} , що не лежить на головній діагоналі матриці, являє собою коваріацію випадкових змінних x_j і x_k (j -ої і k -ої координат вектора \bar{x}). Якщо ознаки x_j і статистично незалежні, то елемент матриці $\sigma_{jk} = 0$. Багатовимірна щільність нормального розподілу зведеться до добутку одновимірних щільностей нормальних розподілів, якщо всі недіагональні елементи коваріаційної матриці – нулі. Вона повністю визначається $n + \frac{n(n+1)}{2}$ параметрами, якими є компоненти вектора середніх значень і незалежні елементи коваріаційної матриці. Об'єкти, вибрані із сукупності, що має нормальний розподіл, схильні до утворення одного кластера, центр якого визначається вектором середніх значень, а форма – коваріаційною матрицею. З формули (4.25) випливає, що геометричними місцями точок з постійною щільністю розподілу є гіпереліпсоїди, напрям головних осей яких визначається власними векторами коваріаційної матриці, а довжина цих осей – її власними значеннями.

Згідно з формулою (4.21) вирішувальну функцію для класу Ω_i можна вибрати у вигляді $d_i(\bar{x}) = p(\bar{x}/\Omega_i)p(\Omega_i)$.

Оскільки щільність нормального розподілу зображається експонентою, то зручніше працювати з натуральним логарифмом від цієї вирішувальної функції. Водночас її можна подати у такому вигляді:

$$d_i(\bar{x}) = \ln[p(\bar{x}/\Omega_i)p(\Omega_i)] = \ln[p(\bar{x}/\Omega_i)] + \ln[p(\Omega_i)]. \quad (4.28)$$

$$d_i(\bar{x}) = \ln p(\Omega_i) - \frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} [(\bar{x} - \bar{m}_i)^T \Sigma_i^{-1} (\bar{x} - \bar{m}_i)] \quad (4.29)$$

Оскільки вираз $\frac{n}{2} \ln 2\pi$ не залежить від i , його можна видалити, тоді вирішувальна функція набуде такого вигляду:

$$d_i(\vec{x}) = \ln p(\Omega_i) - \frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} [(\vec{x} - \vec{m}_i)^T \Sigma_i^{-1} (\vec{x} - \vec{m}_i)], \quad i=1,2,\dots,K. \quad (4.30)$$

Вираз (4.30) є байєсівською вирішувальною функцією для образів з нормальним законом розподілу, справедливу за допущень про нульові втрати від правильної класифікації і рівності втрат за неправильної класифікації. З рівняння (4.30) видно, що вирішувальна границя є квадратичною формою відносно вектора \vec{x} , оскільки вона не містить ні одного члена вище другого порядку, утвореного із компонент вектора знак \vec{x} . Звідси можна зробити висновок, що найкращі результати за байєсівської класифікації образів з нормальним законом розподілу дають узагальнені розподільні поверхні другого порядку.

В випадку однакових коваріаційних матриць $\Sigma_1 = \Sigma_2 = \dots = \Sigma_K = \Sigma$ розподільна границя стає лінійною вирішувальною функцією вектора \vec{x} :

$$d_i(\vec{x}) = \ln p(\Omega_i) + \vec{x}^T \Sigma^{-1} \vec{m}_i - \frac{1}{2} \vec{m}_i^T \Sigma^{-1} \vec{m}_i, \quad i=1,2,\dots,K. \quad (4.31)$$

На рис. 4.5 наведено приклади розподільних границь в двовимірному ознаковому просторі для образів зі щільністю нормального розподілу ймовірностей ознак.

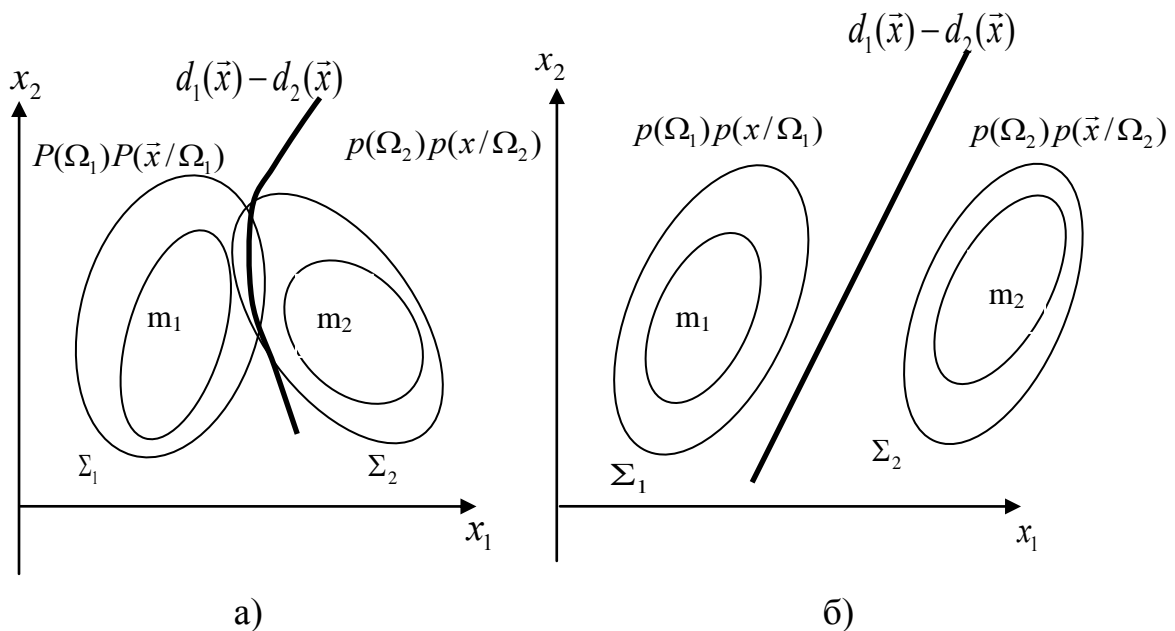


Рисунок 4.5 – Розподільні границі для 2-х класів з нормальним законом розподілу: а) – випадок неоднакових коваріаційних матриць; б) – випадок однакових коваріаційних матриць.

4.4 Оцінення функцій щільності розподілу і їх параметрів

З попередніх розділів витікає, що оцінення щільностей розподілу $p(\vec{x}/\Omega_i)$ є найважливішою задачею, що виникає під час реалізації байєсівського класифікатора образів. Розглянемо методи отримання оцінок щільностей розподілу, що спираються на задану вибірку образів.

Хорошим підґрунтям для вибору щільності розподілу деякого виду є принцип максимуму ентропії. Він стверджує, що за невідомої щільності розподілу деякої випадкової величини логічно вибрати таку щільність розподілу, яка забезпечує максимізацію ентропії випадкової величини за умови урахування всіх відомих обмежень. Для цього критерію щільність розподілу особливо легко визначати в тих випадках, коли обмеження подано в формі середніх оцінок, таких, наприклад, як математичні сподівання і дисперсії щільностей розподілу.

За відомим означенням ентропія сукупності образів з щільністю розподілів $p(\vec{x})$ визначиться, як

$$H = -\int_{\vec{x}} p(\vec{x}) \ln p(\vec{x}) d\vec{x} \quad (4.32)$$

Для спрощення форми запису в (4.32) і надалі $p(\vec{x})$ – це $p(\vec{x}/\Omega_i)$. Нехай апріорна інформація про випадкову величину задається в вигляді таких обмежень:

$$\int_{\vec{x}} p(\vec{x}) d(\vec{x}) = 1 \quad (4.33)$$

і

$$\int_{\vec{x}} B_k(\vec{x}) p(\vec{x}) d(\vec{x}) = a_k, \quad k = 1, 2, \dots, Q. \quad (4.34)$$

В (4.34) $B_k(\vec{x})$ – деяка функція від \vec{x} , a_k – моменти функції щільності розподілу ймовірностей вектора ознак \vec{x} .

За заданих обмежень (4.33) і (4.34) потрібно задати щільність розподілу $p(\vec{x})$ таким чином, щоб величина ентропії (4.32) була максимальною. Для визначення оптимального з погляду цього критерію вигляду функції щільності розподілу скористаємось методом множників Лагранжа [], який дозволяє записати функцію ентропії у вигляді

$$H_1 = -\int_{\vec{x}} p(\vec{x}) [\ln p(\vec{x}) - \sum_{k=0}^Q \lambda_k B_k(\vec{x})] d(\vec{x}) - \sum_{k=0}^Q \lambda_k a_k \quad (4.35)$$

де $\lambda_0, \lambda_1, \dots, \lambda_Q$ – множники Лагранжа, причому $a_0=1$ і $B_0(\vec{x})=1$ для всіх векторів \vec{x} .

Дослідимо функцію (4.35) на максимум, для чого візьмемо частинну похідну від функції H_1 за щільністю розподілу $p(\vec{x})$ і прирівняємо її до нуля:

$$\frac{\partial H_1}{\partial p(\vec{x})} = - \int_{\vec{x}} \left\{ \left[\ln(\vec{x}) - \sum_{k=0}^Q \lambda_k B_k(\vec{x}) \right] + 1 \right\} dx = 0,$$

звідки отримаємо

$$p(\vec{x}) = \exp\left(\sum_{k=0}^Q \lambda_k B_k(\vec{x}) - 1\right). \quad (4.36)$$

В виразі (4.36) потрібно вибрати $Q+1$ параметрів $\lambda_0, \lambda_1, \dots, \lambda_Q$ таким чином, щоб вони відповідали апріорним обмеженням (4.33) і (4.34).

З виразу (4.36) неважко довести, що в тому разі, коли випадкова величина відрізняється від нуля тільки в обмеженому інтервалі, потрібно вибрати рівномірний розподіл щільності ймовірностей. Якщо ж відомо, що випадкова величина може набувати будь-якого дійсного значення, а єдиними розумними характеристиками вважаються математичне сподівання і дисперсія, то потрібно вибрати нормальний розподіл. Після вибору вигляду функції щільностей розподілу ймовірностей ознак класів $p(\vec{x}/\Omega_i)$ переходять до оцінення параметрів вибраної функції. Розглянемо методи оцінення вектора середніх значень і коваріаційної матриці вибірки образів.

Приклад 4.4. Визначити вигляд функції щільностей розподілу ймовірностей, якщо апріорна інформація про випадкову величину x задається такими обмеженнями:

$$\alpha < x < \beta; \quad \int_0^{\infty} p(x) dx = 1.$$

Розв'язування: враховуючи те, що в обмеженні (4.34) a_i і B_i дорівнюють 0 для всіх $i > 0$, з виразу (4.36) можемо записати:

$$p(x) = \exp(\lambda_0 - 1),$$

тоді

$$\int_{\alpha}^{\beta} e^{(\lambda_0 - 1)} dx = 1,$$

звідки

$$\exp(\lambda_0 - 1) = \frac{1}{\beta - \alpha}, \quad \text{і}$$

$$p(x) = \begin{cases} \frac{1}{\beta - \alpha} & \text{за умови } \alpha < x < \beta \\ 0 & \text{в протилежному випадку.} \end{cases} \quad (4.37)$$

Вираз (4.37) характеризує рівномірну щільність розподілу в заданому інтервалі $[\alpha, \beta]$.

Приклад 4.5. Визначити вигляд функції щільностей розподілу, якщо апіорно відомо такі обмеження відносно випадкової величини x :

$$x \geq 0;$$

$$\int_0^{\infty} p(x) d(x) = 1; \quad (4.38)$$

$$\int_0^{\infty} xp(x) d(x) = m. \quad (4.39)$$

Розв'язування: з формули (4.36) витікає, що $p(x) = \exp(\lambda_0 - 1 + \lambda_1 x)$. Підставимо цей вираз в формули (4.38) і (4.39), отримаємо

$$\begin{cases} \int_0^{\infty} \exp(\lambda_0 - 1 + \lambda_1 x) dx = 1, \\ \int_0^{\infty} x \cdot \exp(\lambda_0 - 1 + \lambda_1 x) dx = m. \end{cases} \quad (4.40)$$

Розв'язуючи систему рівнянь (4.40) відносно множників λ_1 і λ_2 , отримаємо вирази

$$\exp(\lambda_0 - 1) = \frac{1}{m}; \quad \lambda_1 = -\frac{1}{m},$$

тоді

$$p(x) = \begin{cases} \frac{1}{m} \exp\left(-\frac{x}{m}\right) & \text{за } x \geq 0, \\ 0 & \text{в протилежному випадку.} \end{cases} \quad (4.41)$$

Приклад 4.6. Визначити вигляд функції щільностей розподілу, якщо апіорно відомі такі обмеження випадкової величини x :

$$-\infty < x < \infty; \quad \int_{-\infty}^{\infty} p(x)dx = 1; \quad \int_{-\infty}^{\infty} xp(x)dx = m; \quad \int_{-\infty}^{\infty} x^2 p(x)dx = \sigma^2 .$$

Розв'язування: З виразу (4.36) можна записати:

$$p(x) = \exp(\lambda_0 - 1 + \lambda_1 x + \lambda_2 x^2).$$

З огляду на відомі обмеження можна записати таку систему рівнянь:

$$\left\{ \begin{array}{l} \int_{-\infty}^{\infty} \exp(\lambda_0 - 1 + \lambda_1 x + \lambda_2 x^2) dx = 1, \\ \int_{-\infty}^{\infty} x \cdot \exp(\lambda_0 - 1 + \lambda_1 x + \lambda_2 x^2) dx = m, \\ \int_{-\infty}^{\infty} x^2 \cdot \exp(\lambda_0 - 1 + \lambda_1 x + \lambda_2 x^2) dx = \sigma^2. \end{array} \right.$$

Розв'язання цієї системи рівнянь відносно множників $\lambda_0, \lambda_1, \lambda_2$ дає

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-m)^2}{2\sigma^2}\right], \quad (4.42)$$

тобто одержали щільність нормального розподілу.

На прикладах 4.4 – 4.6 було показано, що в багатьох випадках тип щільності розподілу визначається набором параметрів. Наприклад, для випадку нормального розподілу (4.42) функція щільності розподілу повністю визначається математичним сподіванням та дисперсією (вектором середніх значень та коваріаційною матрицею для багатовимірних щільностей). Задача визначення вигляду щільності розподілу за її параметрами називається параметричною оцінкою.

Розглянемо методи оцінення вектора середніх значень і коваріаційної матриці, що характеризують деяку сукупність образів.

Нехай сукупність об'єктів, що утворюють цей образ, описується щільністю розподілу $p(\vec{x})$. Вектор середніх значень для цієї сукупності визначиться як

$$\vec{m} = E\{\vec{x}\} = \int_{\vec{x}} \vec{x} p(\vec{x}) d\vec{x}. \quad (4.43)$$

Математичне сподівання $E\{\bar{x}\}$ можна апроксимувати вибіркоvim середнім значенням, тоді

$$\bar{m} = E\{\bar{x}\} = \frac{1}{N} \sum_{i=1}^N \bar{x}_i = \frac{1}{N} \sum_{j=1}^n \sum_{i=1}^N x_{ij} \quad (4.44)$$

де N – обсяг вибірки,

n – розмірність простору ознак.

Коваріаційна матриця в векторній формі визначається виразом:

$$\begin{aligned} \Sigma &= E\{(\bar{x} - \bar{m})(\bar{x} - \bar{m})^T\} = E\left\{ \begin{bmatrix} x_1 - m_1 \\ x_2 - m_2 \\ x_3 - m_3 \\ \vdots \\ x_n - m_n \end{bmatrix} \begin{bmatrix} x_1 - m_1 & x_2 - m_2 & x_3 - m_3 & \dots & x_n - m_n \end{bmatrix} \right\} = \\ &= E\left\{ \begin{pmatrix} (x_1 - m_1)(x_1 - m_1) & (x_1 - m_1)(x_2 - m_2) & (x_1 - m_1)(x_3 - m_3) & (x_1 - m_1)(x_n - m_n) \\ (x_2 - m_2)(x_1 - m_1) & (x_2 - m_2)(x_2 - m_2) & (x_2 - m_2)(x_3 - m_3) & (x_2 - m_2)(x_n - m_n) \\ (x_3 - m_3)(x_1 - m_1) & (x_3 - m_3)(x_2 - m_2) & (x_3 - m_3)(x_3 - m_3) & (x_3 - m_3)(x_n - m_n) \\ \vdots & \vdots & \vdots & \vdots \\ (x_n - m_n)(x_1 - m_1) & (x_n - m_n)(x_2 - m_2) & (x_n - m_n)(x_3 - m_3) & (x_n - m_n)(x_n - m_n) \end{pmatrix} \right\} = \\ &= \begin{vmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \sigma_{13}^2 & \sigma_{1n}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \sigma_{23}^2 & \sigma_{2n}^2 \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_{n1}^2 & \sigma_{n2}^2 & \sigma_{n3}^2 & \sigma_{nn}^2 \end{vmatrix}. \end{aligned} \quad (4.45)$$

Елементами σ_{lk}^2 цієї матриці є числа

$$\sigma_{lk}^2 = E\{(x_l - m_l)(x_k - m_k)\}, \quad l, k = 1, 2, \dots, n \quad (4.46)$$

Діагональними елементами коваріаційної матриці є дисперсії окремих випадкових величин (ознак образу), а недіагональними елементами – коваріації пари випадкових величин (ознак x_l і x_k). Коваріаційна матриця, як видно з виразу (4.45), є симетричною, що дозволяє в подальшому використовувати результати з теорії симетричних матриць. Співвідношення (4.27) можна перетворити до такого вигляду:

$$\Sigma = E\{\bar{x}\bar{x}^T\} - E\{\bar{x}\}\bar{m}^T - \bar{m}E\{\bar{x}^T\} + \bar{m}\bar{m}^T = \mathbf{S} - \bar{m}\bar{m}^T, \quad (4.47)$$

$$\mathbf{S} = E\{\bar{\mathbf{x}}\bar{\mathbf{x}}^T\} = \begin{vmatrix} E\{x_1x_1\} & E\{x_1x_2\} & \dots & E\{x_1x_n\} \\ E\{x_2x_1\} & E\{x_2x_2\} & \dots & E\{x_2x_n\} \\ \dots & \dots & \dots & \dots \\ E\{x_nx_1\} & E\{x_nx_2\} & \dots & E\{x_nx_n\} \end{vmatrix} \quad (4.48)$$

Вираз \mathbf{S} (4.48) називають **автокореляційною матрицею**, або, іноді, матрицею розкиду вектора $\bar{\mathbf{x}}$.

Математичне сподівання в виразі (4.47) можна замінити вибіркоvim середнім значенням:

$$\Sigma = \frac{1}{N} \sum_{i=1}^N \bar{\mathbf{x}}_i \bar{\mathbf{x}}_i^T - \bar{\mathbf{m}}\bar{\mathbf{m}}^T. \quad (4.49)$$

Оцінки вектора середніх значень і коваріаційної матриці можна задати рекурентними співвідношеннями. Припустимо, що необхідно скоригувати оцінку вектора середніх значень, обчислену за вибіркою обсягу N , з урахуванням появи ще одного об'єкта.

Позначимо нову оцінку $\bar{\mathbf{m}}(N+1)$:

$$\bar{\mathbf{m}}(N+1) = \frac{1}{N+1} \sum_{i=1}^{N+1} \bar{\mathbf{x}}_i = \frac{1}{N+1} \left(\sum_{i=1}^N \bar{\mathbf{x}}_i + \bar{\mathbf{x}}_{N+1} \right),$$

або, в кінцевому вигляді

$$\bar{\mathbf{m}}(N+1) = \frac{1}{N+1} (N\bar{\mathbf{m}}(N) + \bar{\mathbf{x}}_{N+1}), \quad (4.50)$$

де $\bar{\mathbf{m}}(N)$ – оцінка, отримана за N вибілковими значеннями. На початковому кроці процедури $\bar{\mathbf{m}}(1) = \bar{\mathbf{x}}_1$. Одержане рекурентне співвідношення можна використовувати як для обчислення, так і для корекції значень вектора середніх значень.

Аналогічно можна отримати вираз для рекурентної оцінки коваріаційної матриці. Нехай $\Sigma(N)$ – оцінка коваріаційної матриці, обчислена за N вибілковими значеннями:

$$\Sigma(N) = \frac{1}{N} \sum_{i=1}^N \bar{\mathbf{x}}_i \bar{\mathbf{x}}_i^T - \bar{\mathbf{m}}(N)\bar{\mathbf{m}}^T(N). \quad (4.51)$$

Поповнення вибірки ще одним об'єктом приводить до такого виразу

$$\begin{aligned}
\sum(N+1) &= \frac{1}{N+1} \sum_{i=1}^{N+1} \bar{x}_i \bar{x}_i^T - \bar{m}(N+1) \bar{m}^T(N+1) = \\
&= \frac{1}{N+1} (\sum_{i=1}^N \bar{x}_i \bar{x}_i^T + \bar{x}_{N+1} \bar{x}_{N+1}^T) - \bar{m}(N+1) \bar{m}^T(N+1) = \\
&= \frac{1}{N+1} (N \sum(N) + N \bar{m}(N) \bar{m}^T(N) + \bar{x}_{N+1} \bar{x}_{N+1}^T) - \\
&\quad - \frac{1}{(N+1)^2} (N \bar{m}(N) + \bar{x}_{N+1}) \cdot (N \bar{m}(N) + \bar{x}_{N+1})^T.
\end{aligned} \tag{4.52}$$

Цей вираз забезпечує зручний спосіб оцінення і корекції коваріаційної матриці. На першому кроці $\sum(1) = \bar{x}_1 \bar{x}_1^T - \bar{m}(1) \bar{m}^T(1)$ і $\bar{m}(1) = \bar{x}_1$. Аналіз цієї умови показує, що $\sum(1) = 0$, тобто це нульова матриця.

Приклад 4.7. Знайти вигляд розподільної поверхні для двох класів об'єктів з нормальним розподілом, вектори зображень яких в тривимірному просторі ознак задано в табл. 4.1.

Таблиця 4.1 – Вектори зображень об'єктів в просторі ознак

| Класи | Вектори | Координати векторів | | |
|----------------|-------------|---------------------|----------------|----------------|
| | | X ₁ | X ₂ | X ₃ |
| Ω ₁ | \bar{X}_1 | 0 | 0 | 0 |
| | \bar{X}_2 | 1 | 0 | 0 |
| | \bar{X}_3 | 1 | 0 | 1 |
| | \bar{X}_4 | 1 | 1 | 0 |
| Ω ₂ | \bar{X}_5 | 0 | 0 | 1 |
| | \bar{X}_6 | 0 | 1 | 1 |
| | \bar{X}_7 | 0 | 1 | 0 |
| | \bar{X}_8 | 1 | 1 | 1 |

Для класифікації використовується вирішувальна функція (4.30) байєсівського класифікатора.

Розв'язування: з виразу (4.30) видно, що для визначення байєсівських вирішувальних функцій $d_1(\bar{x})$ і $d_2(\bar{x})$ потрібно визначити вектори середніх значень і коваріаційні матриці \bar{m}_1, Σ_1 і \bar{m}_2, Σ_2 , відповідно. Для їх оцінювання скористаємося вибірковими значеннями образів:

$$\vec{m}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \vec{x}_{ij},$$

$$\Sigma_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \vec{x}_{ij} \vec{x}_{ij}^T - \vec{m}_i \vec{m}_i^T,$$

де N_i – кількість векторів i -го класу,

\vec{x}_{ij} – j -ий вектор i -го класу.

Застосуємо ці вирази до векторів, що є зображеннями об'єктів для класів Ω_1 і Ω_2 в тривимірному просторі ознак:

$$\vec{m}_1 = \frac{1}{4} (m_1, m_2, m_3)^T,$$

де k -а координата вектора \vec{m}_k визначається як

$$\vec{m}_k = \sum_{j=1}^{N_i} x_{ik} = \sum_{j=1}^4 x_{ik},$$

де j – номер вектора класу Ω_1 ,

k – номер координати вектора.

Наприклад $m_1 = x_{11} + x_{21} + x_{31} + x_{41} = 0 + 1 + 1 + 1 = 3$. Таким самим чином знаходимо координати m_2 і m_3 , маємо:

$$\vec{m}_1 = \frac{1}{4} \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}.$$

Визначимо коваріаційну матрицю Σ_1 :

$$\vec{x}_1 \vec{x}_1^T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \cdot [0 \ 0 \ 0] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad \vec{x}_2 \vec{x}_2^T = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \cdot [1 \ 0 \ 0] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix};$$

$$\vec{x}_3 \vec{x}_3^T = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \cdot [1 \ 0 \ 1] = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}; \quad \vec{x}_4 \vec{x}_4^T = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \cdot [1 \ 1 \ 0] = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix};$$

$$E\{\vec{x}_j \vec{x}_y^T\} = \frac{1}{4} \begin{bmatrix} 3 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}; \quad \vec{m}_1 \vec{m}_1^T = \frac{1}{4} \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} \cdot \frac{1}{4} [3 \ 1 \ 1] = \frac{1}{16} \begin{bmatrix} 9 & 3 & 3 \\ 3 & 1 & 1 \\ 3 & 1 & 1 \end{bmatrix}.$$

Коваріаційна матриця $\Sigma_1 = E\{\vec{x}\vec{x}^T\} - \vec{m}_1 \vec{m}_1^T$,

$$\Sigma_1 = \frac{1}{16} \left(\begin{bmatrix} 12 & 4 & 4 \\ 4 & 4 & 0 \\ 4 & 0 & 4 \end{bmatrix} - \begin{bmatrix} 9 & 3 & 3 \\ 3 & 1 & 1 \\ 3 & 1 & 1 \end{bmatrix} \right) = \frac{1}{16} \begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & -1 \\ 1 & -1 & 3 \end{bmatrix}.$$

Аналогічно визначимо вектор середніх значень і коваріаційну матрицю для класу Ω_2 , скориставшись вибілковими значеннями таблиці 4.1:

$$\vec{m}_2 = \frac{1}{4} \begin{bmatrix} x_{51} + x_{61} + x_{71} + x_{81} \\ x_{52} + x_{62} + x_{72} + x_{82} \\ x_{53} + x_{63} + x_{73} + x_{83} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 0+0+0+1 \\ 0+1+1+1 \\ 1+1+0+1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 \\ 3 \\ 3 \end{bmatrix};$$

$$\vec{m}_2 \vec{m}_2^T = \frac{1}{4} \cdot \frac{1}{4} \begin{bmatrix} 1 \\ 3 \\ 3 \end{bmatrix} \begin{bmatrix} 1 & 3 & 3 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 1 & 3 & 3 \\ 3 & 9 & 9 \\ 3 & 9 & 9 \end{bmatrix};$$

$$\vec{x}_5 \vec{x}_5^T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \vec{x}_6 \vec{x}_6^T = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix};$$

$$\vec{x}_7 \vec{x}_7^T = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad \vec{x}_8 \vec{x}_8^T = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix};$$

$$E\{\vec{x}_j \vec{x}_j^T\} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 2 & 3 \end{bmatrix};$$

$$\Sigma_2 = E\{\vec{x}_j \vec{x}_j^T\} - \vec{m}_2 \vec{m}_2^T = \frac{1}{16} \left(\begin{bmatrix} 4 & 4 & 4 \\ 4 & 12 & 8 \\ 4 & 8 & 12 \end{bmatrix} \right) - \begin{bmatrix} 1 & 3 & 3 \\ 3 & 9 & 9 \\ 3 & 9 & 9 \end{bmatrix};$$

$$\Sigma_2 = \frac{1}{16} \begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & -1 \\ 1 & -1 & 3 \end{bmatrix}.$$

З отриманих значень робимо висновок, що коваріаційні матриці обох класів однакові, тобто $\Sigma_1 = \Sigma_2$, тому байесівські вирішувальні функції в цьому прикладі можна визначити за виразом (4.31). Для спрощення допускаємо однакові апіорні ймовірності класів $p(\Omega_1) = p(\Omega_2) = 0.5$, тоді в виразі складову $\ln p(\Omega_i)$ можна опустити, звідки

$$d_i(\vec{x}) = \vec{x}^T \Sigma^{-1} \vec{m}_i - \frac{1}{2} \vec{m}_i^T \Sigma^{-1} \vec{m}_i, \quad (4.53)$$

де Σ^{-1} – матриця, обернена до коваріаційної.

Обернену коваріаційну матрицю Σ^{-1} знайдемо за методом Жордана – Гауса:

$$\Sigma^{-1} = \begin{bmatrix} 8 & -4 & -4 \\ -4 & 8 & 4 \\ -4 & 4 & 8 \end{bmatrix}.$$

Виконаємо послідовно для отриманих величин дії, передбачені виразом (4.53) :

$$\begin{aligned} \vec{x}^T \Sigma^{-1} \vec{m}_1 &= \frac{1}{4} \cdot [x_1 \quad x_2 \quad x_3] \begin{bmatrix} 8 & -4 & -4 \\ -4 & 8 & 4 \\ -4 & 4 & 8 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} = \\ &= \frac{1}{4} \left[(8x_1 - 4x_2 - 4x_3)(-4x_1 + 8x_2 + 4x_3)(-4x_1 + 4x_2 + 8x_3) \right] \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} = \\ &= \frac{1}{4} [24x_1 - 12x_2 - 12x_3 - 4x_1 + 8x_2 + 4x_3 - 4x_1 + 4x_2 + 8x_3] = \\ &= \frac{1}{4} [16x_1 + 0x_2 + 0x_3] = (4x_1, 0, 0). \end{aligned}$$

$$\begin{aligned} \frac{1}{2} \vec{m}_1^T \Sigma^{-1} \vec{m}_1 &= \frac{1}{2} \cdot \frac{1}{4} [3 \quad 1 \quad 1] \begin{bmatrix} 8 & -4 & -4 \\ -4 & 8 & 4 \\ -4 & 4 & 8 \end{bmatrix} \cdot \frac{1}{4} \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} = \\ &= \frac{1}{32} [24 - 4 - 4 \quad -12 + 8 + 4 \quad -12 + 4 + 8] \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{32} [16 \quad 0 \quad 0] \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} = \frac{3}{2}. \end{aligned}$$

$$\text{Тоді } d_1(\vec{x}) = \vec{x}^T \Sigma^{-1} \vec{m}_1 - \frac{1}{2} \vec{m}_1^T \Sigma^{-1} \vec{m}_1 = 4x_1 - \frac{3}{2}.$$

Аналогічно знаходимо

$$d_2(\vec{x}) = -4x_1 + 8x_2 + 8x_3 - \frac{11}{2}.$$

Розподільна поверхня для двох класів визначиться з виразу:

$$d_1(\vec{x}) - d_2(\vec{x}) = 8x_1 - 8x_2 - 8x_3 + 4 = 0.$$

Зображення об'єктів цього прикладу і розподільної поверхні для двох класів наведено на рис. 4.6.

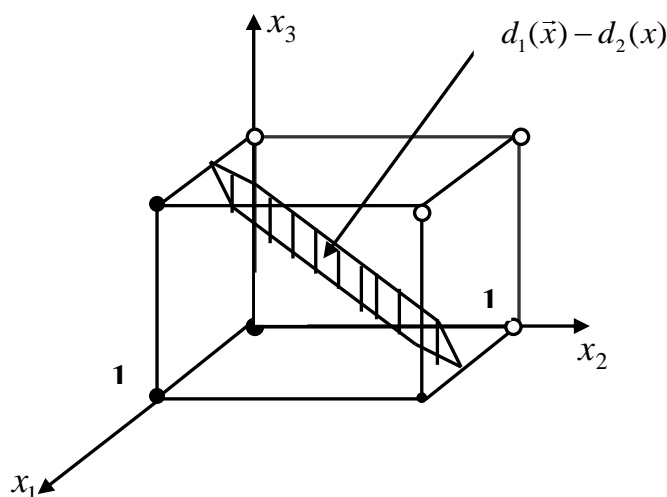


Рисунок 4.6 – Зображення об’єктів, наведених в таблиці 4.1 і розподільна поверхня для класів

4.5 Оцінення помилок байєсівського класифікатора

Як було визначено в підрозділі 4.1, байєсівський класифікатор є оптимальним для випадку наявності апріорної інформації про ймовірності класів і матрицю втрат від неправильної класифікації. Тому цікаво було б оцінити величину помилок класифікації для цього випадку, оскільки вони встановлюють верхній рівень точності для класифікаторів іншого типу. Оскільки образи описані векторами імовірнісного простору ознак, то мова буде йти не про оцінення самої помилки, а про оцінення її ймовірності.

Нехай вектори об’єктів, що відносяться до класів Ω_i і Ω_j , характеризуються такими щільностями багатовимірного нормального розподілу:

$$p(\bar{x}/\Omega_i) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left[-\frac{1}{2}(\bar{x} - \bar{m}_i)^T \Sigma_i^{-1} (\bar{x} - \bar{m}_i)\right] \quad (4.54)$$

і

$$p(\bar{x}/\Omega_j) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left[-\frac{1}{2}(\bar{x} - \bar{m}_j)^T \Sigma_j^{-1} (\bar{x} - \bar{m}_j)\right], \quad (4.55)$$

їх коваріаційні матриці однакові: $\Sigma_i = \Sigma_j = \Sigma$.

Спростимо аналіз шляхом використання логарифма відношення правдоподібності:

$$u_{ij} = \ln(l_{ij}(\bar{x})) = \ln p(\bar{x}/\Omega_i) - \ln p(\bar{x}/\Omega_j). \quad (4.56)$$

Підставляючи в останній вираз щільності розподілів з виразів (4.54) і (4.55), отримаємо:

$$u_{ij}(\bar{x}) = \bar{x}^T \Sigma^{-1}(\bar{m}_i - \bar{m}_j) - \frac{1}{2}(\bar{m}_i + \bar{m}_j)^T \Sigma^{-1}(\bar{m}_i - \bar{m}_j). \quad (4.57)$$

Якщо вибрати двійкову функцію втрат (0 – втрати для правильного рішення, 1 – втрати для хибного рішення), то правило, що визначає належність образу \bar{x} до класу Ω_i за критерієм мінімізації ймовірності помилки класифікації, набуває вигляду

$$u_{ij}(\bar{x}) > \alpha, \quad (4.58)$$

де

$$\alpha = \ln \theta = \ln \frac{p(\Omega_i)}{p(\Omega_j)}. \quad (4.59)$$

Тоді ймовірність неправильної класифікації образу, належного до класу Ω_j , визначається виразом $p(u_{ij} > \alpha)$; а ймовірність неправильної класифікації образу, належного класу Ω_i , є $p(u_{ij} < \alpha)$.

Оскільки значення вектора образу \bar{x} підлягають нормальному закону розподілу ймовірностей, то u_{ij} також описується нормальним розподілом, оскільки є лінійною комбінацією компонент образу \bar{x} . З виразу (4.57) математичне сподівання логарифма відношення правдоподібності u_{ij} для класу Ω_i можна записати у вигляді:

$$E_i \{u_{ij}\} = \bar{m}_i^T \Sigma^{-1}(\bar{m}_i - \bar{m}_j) - \frac{1}{2}(\bar{m}_i + \bar{m}_j)^T \Sigma^{-1}(\bar{m}_i - \bar{m}_j). \quad (4.60)$$

Перепишемо (4.60) таким чином:

$$E\{U_{ij}\} = \tilde{U}_{ij} = \frac{1}{2}(\bar{m}_i - \bar{m}_j)^T \Sigma^{-1}[\bar{m}_i - \bar{m}_j] = \frac{1}{2}r_{ij}, \quad (4.61)$$

де

$$r_{ij} = (\bar{m}_i - \bar{m}_j)^T \Sigma^{-1}(\bar{m}_i - \bar{m}_j) \quad (4.62)$$

називають відстанню Махаланобіса між щільностями розподілу $p(\bar{x}/\Omega_i)$ і $p(\bar{x}/\Omega_j)$.

Дисперсія u_{ij} визначається, як

$$D_i \{u_{ij}\} = E_i \left\{ (u_{ij} - \tilde{u}_{ij})^2 \right\}, \quad (4.63)$$

або

$$D \{u_{ij}\} = E_i \left\{ (\vec{m}_i - \vec{m}_j)^T \Sigma^{-1} (\vec{x} - \vec{m}_i) (\vec{x} - \vec{m}_j)^T \Sigma^{-1} (\vec{m}_i - \vec{m}_j) \right\}. \quad (4.64)$$

Спростивши останній вираз, отримаємо:

$$D_i \{u_{ij}\} = r_{ij}. \quad (4.65)$$

Таким чином, за умови $\vec{x} \in \Omega_i$ логарифм відношення правдоподібності u_{ij} має нормальний закон розподілу $N\left(\frac{r_{ij}}{2}, r_{ij}\right)$.

Аналогічно можна визначити, що за умови $\vec{x} \in \Omega_j$ відповідний логарифм відношення правдоподібності u_{ij} підлягає нормальному закону розподілу $N\left(\frac{-r_{ij}}{2}, r_{ij}\right)$.

Тоді ймовірність неправильної класифікації зображення $\vec{x} \in \Omega_j$ дорівнює

$$p(u_{ij} > \alpha) = \int_{\alpha}^{\infty} \frac{1}{\sqrt{2\pi r_{ij}}} \exp \left[-\frac{\left(u_{ij} + \frac{1}{2} r_{ij}\right)^2}{2r_{ij}} \right] du_{ij} = 1 - \Phi \left(\frac{\alpha + \frac{1}{2} r_{ij}}{\sqrt{r_{ij}}} \right), \quad (4.66)$$

а ймовірність неправильної класифікації зображення $\vec{x} \in \Omega_i$

$$p(u_{ij} < \alpha) = \int_{-\infty}^{\alpha} \frac{1}{\sqrt{2\pi r_{ij}}} \exp \left[-\frac{\left(u_{ij} + \frac{1}{2} r_{ij}\right)^2}{2r_{ij}} \right] du_{ij} = 1 - \Phi \left(\frac{\alpha + \frac{1}{2} r_{ij}}{\sqrt{r_{ij}}} \right). \quad (4.67)$$

Функція Φ (функція Лапласа) визначається як

$$\Phi(\xi) = \int_{-\infty}^{\xi} \frac{1}{\sqrt{2\pi}} \cdot \exp(-y^2 / 2) dy. \quad (4.68)$$

Тоді ймовірність похибки класифікації визначиться за формулою:

$$p(e) = p(w_i) \cdot p(u_{ij} < \alpha / w_i) + p(w_j) \cdot p(u_{ij} > \alpha / w_j),$$

або

$$p(e) = p(w_i) \cdot \Phi\left(\frac{\alpha - \frac{1}{2} r_{ij}}{\sqrt{r_{ij}}}\right) + p(w_j) \cdot [1 - \Phi\left(\frac{\alpha + \frac{1}{2} r_{ij}}{\sqrt{r_{ij}}}\right)]. \quad (4.69)$$

В тих випадках, коли апріорні ймовірності появи відповідних класів однакові, тобто: $p(w_i) = p(w_j)$, тоді $\theta = 1$, а $\alpha = 0$. Для цього випадку ймовірність похибки визначиться, як:

$$p(e) = \frac{1}{2} \cdot \Phi\left(-\frac{1}{2} \sqrt{r_{ij}}\right) + \frac{1}{2} [1 - \Phi\left(\frac{1}{2} \sqrt{r_{ij}}\right)], \quad (4.70)$$

або

$$p(e) = \int_{0.5\sqrt{r_{ij}}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp(-y^2 / 2) dy. \quad (4.71)$$

Таким чином, функція, що пов'язує відстань Махаланобіса з ймовірністю помилки класифікації, є щільністю одновимірного розподілу з нульовим середнім і одиничною дисперсією.

Приклад 4.8. Визначити вплив завад на точність розпізнавання мовних сигналів.

Розв'язування: оскільки мовний сигнал є відгуком лінійної системи з повільно змінними параметрами, то на проміжку часу стаціонарності цієї системи (звичайно $\tau=10-30$ мс), його можна вважати квазидетермінованим процесом для одних і тих самих класів зображень. Тому первинний опис мовного сигналу виконується для інтервалів його стаціонарності у вигляді векторів оцінок енергії короткочасного спектра на послідовності частотних смуг. Ці оцінки в реальних системах розпізнавання одержують за допомогою гребінки смугових фільтрів з підключеними до їхніх виходів детекторами. За наявності адитивної завади інформаційний сигнал можна подати у такому вигляді

$$y^*(t, \tau) = y(t, \tau) + \xi(t, \tau). \quad (4.72)$$

У більшості випадків завада $\xi(t, \tau)$ має нульове середнє значення і некорельована з мовним сигналом $y(t, \tau)$, тому вектор спектрального опису сигналу запишеться виразом:

$$X = X_y + X_\xi, \quad (4.73)$$

де $X_y = (x_{y1}, x_{y2}, \dots, x_{yn})^T$, $X_\xi = (x_{\xi1}, x_{\xi2}, \dots, x_{\xi n})^T$ – вектор спектрального опису мовного сигналу і завади, відповідно,

T – символ транспонування,

x_i – потужність поточного спектру в i -тій частотній смузі.

Для усунення впливу на результати розпізнавання голосності вимови, яка не несе інформації про клас сигналу, необхідно використовувати нормований вектор спектрального опису, визначивши норму у вигляді:

$$\|X\| = \sum_{i=1}^n x_i = \sigma_c^2, \quad (4.74)$$

де σ_c^2 – дисперсія (потужність) випадкового процесу на вході системи розпізнавання.

Після нормування вектора (4.73) відповідно до (4.74) одержимо:

$$\tilde{X} = \frac{r^2}{r^2 + 1} \tilde{X}_y + \frac{1}{r^2 + 1} \tilde{X}_\xi,$$

де \tilde{X} , \tilde{X}_y , \tilde{X}_ξ – нормовані вектори X , X_y , X_ξ , відповідно,

$r = \sqrt{\frac{\sigma_y^2}{\sigma_\xi^2}}$ – відношення «сигнал / шум» на вході системи розпізнавання. З

огляду на те, що в практичних випадках $r^2 \gg 1$, приходимо до виразу:

$$\tilde{X} = \tilde{X}_y + \frac{1}{r^2} \tilde{X}_\xi. \quad (4.75)$$

Не втрачаючи загальності міркувань, точність роботи класифікатора системи розпізнавання можна визначити на прикладі двох класів мовних образів Ω_1 і Ω_2 , оскільки результати легко узагальнити для випадку багатьох класів. Через обмеження на час розпізнавання й обчислювальні ресурси в реальних системах у більшості випадків використовують класифікацію за правилом «найближчого середнього». Тому оцінку точності розпізнавання мовних сигналів, поданих спектральним описом (4.75), зробимо для класифікатора цього типу. Вирішувальне правило для нього має вигляд:

$$d_e(\tilde{X}, \mu_i) = \min d_e(\tilde{X}, \mu_i) \Rightarrow X \in \Omega_i, \quad i, j = 1, 2. \quad (4.76)$$

У виразі (4.76) $\mu_i = E \{ \tilde{X} \mid \tilde{X} \in \Omega_i \}$ – середнє значення векторів, що належать класу \tilde{X} , еталон класу Ω_i , а $d_e(\tilde{X}, \mu_i)$ є евклідовою відстанню вектора \tilde{X} до вектора μ_i :

$$d_e(\tilde{X}, \mu_i) = [(\tilde{X} - \mu_i)^T(\tilde{X} - \mu_i)].$$

У термінах розподільної для класів Ω_1 і Ω_2 гіперплощини правило (4.76) для сигналу без завад запишеться як:

$$H_{12}(\tilde{X}) = (\tilde{X}_y - \mu_1)^T(\tilde{X}_y - \mu_1) - (\tilde{X}_y - \mu_2)^T(\tilde{X}_y - \mu_2) = 0, \quad (4.77)$$

або, після елементарних перетворень:

$$H_{12}(\tilde{X}) = 2\tilde{X}_y^T(\mu_2 - \mu_1) + \mu_1^T\mu_1 - \mu_2^T\mu_2 = 0. \quad (4.78)$$

Якщо $H_{12}(\tilde{X}) > 0$, то $\tilde{X}_y \in \Omega_1$, а якщо $H_{12}(\tilde{X}) \leq 0$, то $\tilde{X}_y \in \Omega_2$. Підставляючи (4.75) у (4.78), отримаємо вираз для розподільної гіперплощини за наявності завад з урахуванням того, що навчання здійснювалося на «чистому» сигналі:

$$H_{12}^r(\tilde{X}) = 2\tilde{X}_y^T(\mu_2 - \mu_1) + \frac{2}{r^2}\tilde{X}_\xi^T(\mu_2 - \mu_1) + \mu_1^T\mu_1 - \mu_2^T\mu_2 = 0, \quad (4.79)$$

або

$$H_{12}^r(\tilde{X}) = H_{12}(\tilde{X}) + \frac{2}{r^2}\tilde{X}_\xi^T(\mu_2 - \mu_1).$$

Звідси випливає, що дія завад еквівалентна зсуву розподільної площини вбік класу Ω_1 чи класу Ω_2 залежно від знака алгебраїчної суми компонентів різницевого вектора $(\mu_2 - \mu_1)$. Оскільки для квазідетермінованого мовного сигналу потужність миттєвого спектра в i -тій частотній смузі є випадковою функцією, то спектральний вектор опису \tilde{X} є випадковим вектором, що характеризується багатовимірним нормальним розподілом. Тоді вирішувальні функції (4.78) і (4.79), що є лінійними функціями вектора \tilde{X} , можна описати щільністю одновимірного нормального розподілу. Середнє значення функції (4.78) визначиться як:

$$E\{H_{12}(\tilde{X})\} = 2(\mu_2 - \mu_1)^T E\{\tilde{X}_y\} + \mu_1^T\mu_1 - \mu_2^T\mu_2, \quad (4.80)$$

а для вирішувальної функції (4.79) як

$$E\{H^r_{12}(\tilde{X})\} = 2(\mu_2 - \mu_1)^T E\{\tilde{X}_y\} + \frac{2}{r^2} \tilde{X}_\xi^T (\mu_2 - \mu_1) + \mu_1^T \mu_1 - \mu_2^T \mu_2. \quad (4.81)$$

Під час виведення формули (4.81) можна прийняти

$$\frac{2}{r^2} E\{\tilde{X}_\xi^T\} \approx \frac{2}{r^2} \tilde{X}_\xi^T,$$

тому що середнє по множині від миттєвого спектра стаціонарного випадкового сигналу завади є статистичним спектром, який для більшості класів акустичних завод досить близький до миттєвого спектра з слабкими флуктуаціями.

Дисперсія вирішувальної функції (4.78), по визначенню, дорівнює:

$$\sigma_H^2 = E \left\{ \left[2(\mu_2 - \mu_1)^T \tilde{X}_y + \mu_1^T \mu_1 - \mu_2^T \mu_2 \right] - \left[2(\mu_2 - \mu_1)^T E(\tilde{X}_y) + \mu_1^T \mu_1 - \mu_2^T \mu_2 \right] \right\}^2. \quad (4.82)$$

Після нескладних перетворень отримаємо:

$$\sigma_H^2 = 2(\mu_2 - \mu_1)^T \sum_i (\mu_2 - \mu_1), \quad i, j = 1, 2, \quad (4.83)$$

де \sum_i – коваріаційна матриця i -того класу.

Для спрощення викладок вважатимемо коваріаційні матриці обох класів однаковими $\sum_1 = \sum_2 = \sum$.

Підстановка (4.79) у (4.82) дає вираз для дисперсії вирішувальної функції $H^r_{12}(\tilde{X})$, аналогічний виразу (4.83). Таким чином, за $\tilde{X} \in \Omega_1$ вирішувальні функції $H_{12}(\tilde{X})$ і $H^r_{12}(\tilde{X})$ підкоряються нормальним законам розподілу: $N_1(m_1, \sigma_H)$ і $N^r_1(m_1 + \frac{2}{r^2} \tilde{X}_\xi^T (\mu_2 - \mu_1), \sigma_H)$, відповідно. Середнє значення m_1 визначимо, підставивши в (4.80) значення $E\{\tilde{X}_y\} = \mu_1$ і дорівнює

$$m_1 = \mu_2^T (2\mu_1 - \mu_2) - \mu_1^T \mu_1. \quad (4.84)$$

Аналогічно для $\tilde{X} \in \Omega_2$ закони розподілу для функцій (4.78) і (4.79) запишуться як:

$$N_2(m_2, \sigma_H), N^r_2(m_2 + \frac{2}{r^2} \tilde{X}_\xi^T (\mu_2 - \mu_1), \sigma_H),$$

де

$$m_2 = \mu_1^T (\mu_1 - 2\mu_2) - \mu_2^T \mu_2. \quad (4.85)$$

Тоді ймовірність помилки класифікації для вирішувальної функції (4.78) за відсутності завад запишеться як сума помилок 1-го і 2-го роду:

$$p(e) = p(\Omega_1)p(H_{12}(\tilde{X}) < \theta \left| \tilde{X} \in \Omega_1 \right.) + p(\Omega_2)p(H_{12}(\tilde{X}) > \theta \left| \tilde{X} \in \Omega_2 \right.), \quad (4.86)$$

де θ – поріг, вибраний з умови мінімуму втрат від помилок розпізнавання. Тоді:

$$p(H_{12}(\tilde{X}) < \theta \left| \tilde{X} \in \Omega_1 \right.) = \int_{-\infty}^{\theta} \frac{1}{\sigma_H \sqrt{2\pi}} \exp \frac{-[H_{12}(\tilde{X}) - m_1]^2}{2\sigma_H^2} dH_{12}(\tilde{X}), \quad (4.87)$$

$$p(H_{12}(\tilde{X}) > \theta \left| \tilde{X} \in \Omega_2 \right.) = \int_{-\infty}^{\theta} \frac{1}{\sigma_H \sqrt{2\pi}} \exp \frac{-[H_{12}(\tilde{X}) - m_2]^2}{2\sigma_H^2} dH_{12}(\tilde{X}). \quad (4.88)$$

Підставивши (4.87) і (4.88) у (4.86) і виконавши ряд перетворень, одержимо:

$$p(e) = p(\Omega_1)\Phi^*\left(\frac{\theta - m_1}{\sigma_H}\right) + p(\Omega_2)\Phi^*\left(\frac{m_2 - \theta}{\sigma_H}\right), \quad (4.89)$$

де $\Phi^*(a)$ – функція Лапласа.

Другий доданок у (4.89) отримано з урахуванням властивості функції Лапласа $\Phi^*(-a) = 1 - \Phi^*(a)$. Під час вибору двійкової функції втрат (0 – правильне рішення, 1 – помилкове), порогова величина визначається з відношення правдоподібності:

$$\frac{p(H_{12}(\tilde{X}) = \theta \left| \tilde{X} \in \Omega_1 \right.)}{p(H_{12}(\tilde{X}) = \theta \left| \tilde{X} \in \Omega_2 \right.)} = \frac{p(\Omega_1)}{p(\Omega_2)} = \lambda. \quad (4.90)$$

Після підстановки (4.87) і (4.88) у (4.90) і розв'язання отриманого рівняння відносно θ , знаходимо:

$$\theta = \frac{m_1 + m_2}{2} + \frac{\sigma_H^2 \ln \lambda}{(m_1 - m_2)}. \quad (4.91)$$

Замінюючи в (4.91) величини m_1 , m_2 , σ_H^2 їхніми значеннями, визначаємо остаточно граничну величину θ :

$$\theta = \sum \lambda, \quad (4.92)$$

де Σ – коваріаційна матриця.

За наявності завади величина θ набуде значення:

$$\theta_{\xi} = \frac{4}{r^2} \tilde{X}_{\xi}^T (\mu_2 - \mu_1) + \sum \lambda = \Delta_{\xi} + \theta. \quad (4.93)$$

Тоді ймовірність помилки за наявності завад визначається виразом:

$$p(e_{\xi}) = p(\Omega_1) \Phi^* \left(\frac{\theta + \Delta_{\xi} - m_1}{\sigma_H} \right) + p(\Omega_2) \Phi^* \left(\frac{m_2 - (\theta + \Delta_{\xi})}{\sigma_H} \right). \quad (4.94)$$

Таким чином, проведений аналіз дозволив установити теоретичну залежність помилки розпізнавання від діючого на мовний сигнал шуму, подану виразом (4.94). Перевагою цієї формули є те, що вона виключає необхідність інтегрування щільності розподілу ймовірності ознак у n -вимірному просторі.

Використовуючи формули (4.89) і (4.94), можна визначити кількісно збільшення помилки розпізнавання від величини завади. На рис. 4.7 наведено графіки залежності ймовірності помилки класифікації від величини Δ_{ξ} складової завади спектрального опису мовних образів з рівними апіорними ймовірностями.

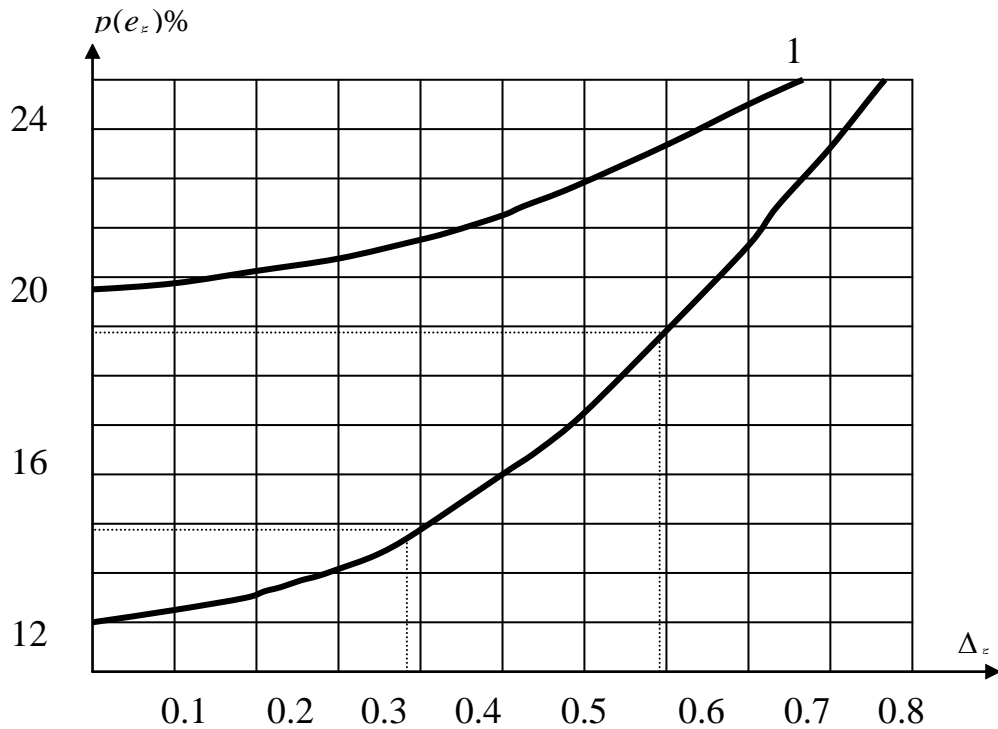


Рисунок 4.7 – Графіки залежності ймовірності помилки класифікації від величини завадової складової вектора спектрального опису мовного сигналу : 1 - $m_1 = 1, \sigma_H = 1$; 2 - $m_1 = 1, \sigma_H = 0.5$

У цьому випадку обчислення здійснювались за формулою (4.94), перетвореною до вигляду:

$$p(e_\xi) = \frac{1}{2} \left[\Phi^* \left(\frac{\Delta_\xi - m_2}{\sigma_H} \right) + \Phi^* \left(\frac{-\Delta_\xi - m_1}{\sigma_H} \right) \right]. \quad (4.95)$$

Вираз (4.95) отримано за умов однакових апіорних ймовірностей класів Ω_1 і Ω_2 , а також з урахуванням однаковості $m_2 = -m_1$, яка витікає з порівняння (4.84) і (4.85).

Аналіз отриманих графіків дозволяє зробити такі висновки:

1. Усунення впливу завад на точність розпізнавання мовних сигналів методом їх фільтрації дає високу ефективність тільки для класів образів з гарною роздільною здатністю, а більш конкретно, якщо розкид σ_H образів у класах менший або дорівнює половині відстані m_1 між їх еталонами: $\sigma_H \leq m_1/2$.

2. Фільтрація завад великої амплітуди дає більший вигравш у точності розпізнавання, чим фільтрація перешкод з малою амплітудою.

4.6 Апроксимація щільностей розподілу функціями

В попередніх розділах було розглянуто окремі випадки оцінення умовних щільностей розподілу, коли є відомим закон розподілу (підрозділ 4.4) або відома деяка апіорна інформація про випадкову величину \vec{x} у вигляді системи обмежень (підрозділ 4.5). Як було показано, в першому випадку оцінення щільностей розподілу зводилась до оцінення параметрів закону розподілу, а в другому випадку – до визначення вигляду функції і оцінення її параметрів.

В загальному випадку такі припущення є неприйнятними і виникає необхідність оцінювати щільність розподілу безпосередньо за вибіркою зображень $\mathbf{X} = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$. Нехай $p(\vec{x})$ – щільність розподілу випадкової величини \vec{x} (під $p(\vec{x})$ тут розуміємо умовну щільність розподілу вектора ознак образу Ω – $p(\vec{x}/\Omega)$). Якщо $p(\vec{x})$ відповідає умовам Діріхле, то її можна апроксимувати у вигляді суми ν членів ряду

$$\hat{p}(\vec{x}) = C_0 \varphi_0(\vec{x}) + C_1 \varphi_1(\vec{x}) + \dots + C_\nu \varphi_\nu(\vec{x}) = \sum_{j=1}^{\nu} C_j \varphi_j(\vec{x}). \quad (4.96)$$

У виразі (4.96) C_j – коефіцієнти, що підлягають визначенню, $\{\varphi_j(\vec{x})\}$ – множина вибраних базисних ортогональних функцій, а $\hat{p}(\vec{x})$ – оцінка щільності розподілу $p(\vec{x})$. Система дійсних функцій $\{\varphi_j(\vec{x})\}$

називається ортогональною відносно вагової функції $\rho(\bar{x})$ на відрізьку $[a; b]$, якщо

$$\int_a^b \rho(\bar{x}) \cdot \varphi_i(\bar{x}) \cdot \varphi_j(\bar{x}) d\bar{x} = 0 \text{ при } i \neq j. \quad (4.97)$$

Вираз (4.97) визначає попарну ортогональність функцій $\varphi_i(\bar{x})$. Величина

$$\|\varphi_i(\bar{x})\| = \sqrt{\int_a^b \varphi_i^2(\bar{x}) d\bar{x}} \quad (4.98)$$

називається нормою функції $\varphi_j(\bar{x})$. Якщо квадрат норми функції дорівнює одиниці:

$$\|\varphi_i(\bar{x})\|^2 = \int_a^b \varphi_i^2(\bar{x}) d\bar{x} = 1, \quad (4.99)$$

то вона називається нормованою, а система попарно ортогональних нормованих функцій називається ортонормованою темою.

Якщо деяка система функцій $\varphi_1^*(\bar{x}), \varphi_2^*(\bar{x}), \dots$ ортогональна в інтервалі $[a; b]$, то для одержання в цьому самому інтервалі ортонормованої системи функцій потрібно скористатися співвідношенням

$$\varphi_i(\bar{x}) = \sqrt{\frac{\rho(\bar{x})}{A_i}} \varphi_i^*(\bar{x}), \quad (4.100)$$

де коефіцієнт A_i визначається формулою

$$A_i = \int_a^b \rho(\bar{x}) \varphi_i^{*2}(\bar{x}) d\bar{x}. \quad (4.101)$$

Множина функцій $\{f_1(\bar{x}), f_2(\bar{x}), \dots, f_v(\bar{x})\}$ називається лінійно незалежною, якщо не існує коефіцієнтів

$$c_1, c_2, \dots, c_v,$$

не всі з яких дорівнюють нулю і таких, що для всіх \bar{x} виконується рівняння

$$c_1 f_1(\bar{x}) + c_2 f_2(\bar{x}) + \dots + c_v f_v(\bar{x}) = 0. \quad (4.102)$$

Всі функції, що утворюють ортогональну систему, є лінійно незалежними.

Крім того, система функцій називається повною, якщо будь-яку кусково-неперервну функцію можна в середньому як завгодно точно апроксимувати за допомогою лінійної комбінації функцій, що утворюють цю систему.

В задачі апроксимації функцій найбільш уживаними є такі ортогональні системи функцій, як поліноми Чебишова, Ерміта, Лагера, Лежандра, функції Уолша та інші. Всі ці системи є функціонально повними. З метою спрощення розуміння подальших викладок дамо означення цим системам спочатку для випадку функцій однієї змінної, а потім перейдемо до ортогональних систем функцій багатьох змінних.

Поліноміальні функції Лежандра однієї змінної можна отримати, скориставшись таким рекурентним співвідношенням:

$$(k+1)P_{k+1}(x) = (2k+1)xP_k(x) + kP_{k-1}(x) = 0, \quad k \geq 1, \quad (4.103)$$

де $P_0(x) = 1, P_1(x) = x$. Ці функції є ортогональними в діапазоні $x \in [1;1]$ відносно вагової функції $\rho(x) = 1$.

Поліноми Лежандра нижчих степенів визначаються виразами:

$$\begin{aligned} P_0(x) &= 1, \\ P_1(x) &= x, \\ P_2(x) &= \frac{3}{2}x^2 - \frac{1}{2}, \\ P_3(x) &= \frac{5}{3}x^3 - \frac{3}{2}x, \\ P_4(x) &= \frac{35}{8}x^4 + \frac{15}{4}x^2 + \frac{3}{8}. \end{aligned} \quad (4.104)$$

Ортогональні многочлени Лагерра отримують за рекурентним співвідношенням

$$L_{k+1}(x) = (2k+1-x)L_k(x) - k^2L_{k-1}(x), \quad k = 0,1,2,\dots, \quad (4.105)$$

де $L_0(x) = 1$ і $L_1(x) = -x + 1$.

Многочлени Лагерра ортогональні на додатній півосі $0 < x < \infty$ з вагою $\rho(x) = e^{-x}$.

Можна навести за приклад декілька перших многочленів Лагерра, отриманих за формулою (4.105):

$$\begin{aligned} L_0(x) &= 1; \\ L_1(x) &= -x + 1; \\ L_2(x) &= x^2 - 4x + 2; \end{aligned} \quad (4.106)$$

$$L_3(x) = -x^3 + 9x^2 - 18x + 6;$$

$$L_4(x) = x^4 - 16x^3 + 72x^2 - 96x + 24.$$

Оскільки многочлени Лагерра утворюють системи розбіжних за $x \rightarrow \infty$ функцій, то зручніше користуватися функціями Лагерра, ортогональних з одиничною вагою:

$$\varphi_k(x) = \sqrt{\rho(x)} \cdot L_k(x) = e^{-\frac{x^2}{2}} \cdot L_k(x), \quad k = 0, 1, 2, \dots \quad (4.107)$$

Поліноми Ермітта визначаються за рекурентною формулою:

$$H_{k+1}(x) = 2xH_k(x) + 2kH_{k-1}(x), \quad k \geq 1, \quad (4.108)$$

де $H_0(x) = 1$ і $H_1(x) = 2x$. Ці функції ортогональні відносно вагової функції $\rho(x) = e^{-x^2}$ на інтервалі $-\infty < x < \infty$; що робить їх достатньо зручними, оскільки звільняє від турбот відносно діапазону зміни аргументів x .

Перші п'ять поліномів Ермітта:

$$\begin{aligned} H_0(x) &= 1; \\ H_1(x) &= 2x; \\ H_2(x) &= 4x^2 - 2; \\ H_3(x) &= 8x^3 - 12x; \\ H_4(x) &= 16x^4 - 48x^2 + 12. \end{aligned} \quad (4.109)$$

Для переходу до ортонормованої системи поліномів Ермітта вводять функцію:

$$\varphi_k(x) = \frac{H_k(x)\sqrt{\rho(x)}}{\|H_k(x)\sqrt{\rho(x)}\|} = \frac{e^{-\frac{x^2}{2}} \cdot H_k(x)}{\sqrt{n!2^n \sqrt{\pi}}}, \quad k = 0, 1, 2, \dots \quad (4.110)$$

Вирази (4.104), (4.106) і (4.108) відповідають функціям однієї змінної. Оскільки в загальному випадку щільність розподілу вектора ознак $p(\vec{x})$ є функцією багатьох змінних, то потрібно знати методи її побудови з ортонормованих функцій однієї змінної. Функцію $p(\vec{x}) = p(x_1, x_2, \dots, x_n)$ можна побудувати з множини ортонормованих функцій $\varphi_1(x), \varphi_2(x), \dots, \varphi_n(x)$ шляхом формування груп добутків з n функцій однієї змінної, підставляючи відповідно змінні x_1, x_2, \dots, x_n .

Приклад 4.9. Побудувати повну систему ортонормованих функцій двох змінних x_1 і x_2 , якщо задано певну систему ортонормованих функцій однієї змінної $\varphi_1(x), \varphi_2(x), \dots, \varphi_n(x)$ на інтервалі $a \leq x \leq b$.

Розв'язування:

$$\begin{aligned} f_1(\vec{x}) &= p_1(x_1, x_2) = \varphi_1(x_1) \cdot \varphi_1(x_2); \\ f_2(\vec{x}) &= p_2(x_1, x_2) = \varphi_1(x_1) \cdot \varphi_2(x_2); \end{aligned}$$

$$\begin{aligned}
f_3(\vec{x}) &= p_3(x_1, x_2) = \varphi_2(x_1) \cdot \varphi_1(x_2); \\
f_4(\vec{x}) &= p_4(x_1, x_2) = \varphi_2(x_1) \cdot \varphi_2(x_2); \\
f_5(\vec{x}) &= p_5(x_1, x_2) = \varphi_2(x_1) \cdot \varphi_3(x_2);
\end{aligned}
\tag{4.111}$$

Приклад 4.10. Для умов попереднього прикладу побудувати множину функцій $\{f_i(\vec{x}) = f_i(x_1, x_2, x_3, x_4)\}$.

Розв'язування:

$$\begin{aligned}
f_1(\vec{x}) &= \varphi_1(x_1) \cdot \varphi_1(x_2) \cdot \varphi_1(x_3) \cdot \varphi_1(x_4); \\
f_2(\vec{x}) &= \varphi_1(x_1) \cdot \varphi_1(x_2) \cdot \varphi_1(x_3) \cdot \varphi_2(x_4); \\
f_3(\vec{x}) &= \varphi_1(x_1) \cdot \varphi_1(x_2) \cdot \varphi_2(x_3) \cdot \varphi_1(x_4); \\
f_4(\vec{x}) &= \varphi_1(x_1) \cdot \varphi_1(x_2) \cdot \varphi_2(x_3) \cdot \varphi_2(x_4); \\
f_5(\vec{x}) &= \varphi_1(x_1) \cdot \varphi_2(x_2) \cdot \varphi_1(x_3) \cdot \varphi_1(x_4).
\end{aligned}
\tag{4.112}$$

Неважко довести, що отримані функції багатьох змінних є ортонормованими на гіперкубі $a \leq \vec{x} \leq b$, $j = 1, 2, \dots, n$.

Після того, як функції $\varphi_j(\vec{x})$ вибрано, визначають коефіцієнти C_j . Коефіцієнти C_j у виразі (4.96) визначаються таким чином, щоб мінімізувати критерій середньоквадратичного відхилення отриманої оцінки функції $\hat{p}(\vec{x})$ від дійсної щільності розподілу $p(\vec{x})$:

$$R = \int_{\vec{x}} [p(\vec{x}) - \hat{p}(\vec{x})]^2 d\vec{x} = \int_{\vec{x}} \left[p(\vec{x}) - \sum_{j=1}^v C_j f_j(\vec{x}) \right]^2 d\vec{x}. \tag{4.113}$$

Коефіцієнти C_j , що мінімізують інтеграл імовірності помилки (4.113), знаходяться з умови

$$\frac{\partial R}{\partial C_k} = 0, \quad k = 1, 2, \dots, v. \tag{4.114}$$

Візьмемо частинну похідну, отримаємо:

$$\sum_{j=1}^v C_j \int_{\vec{x}} f_j(\vec{x}) f_k(\vec{x}) d\vec{x} = \int_{\vec{x}} f_k(\vec{x}) p(\vec{x}) d\vec{x}. \tag{4.115}$$

Права частина виразу (4.115) є математичним сподіванням функції $f_k(\vec{x})$, яку можна апроксимувати вибіркоvim середнім:

$$\int_{\vec{x}} f_k(\vec{x}) p(\vec{x}) d\vec{x} = \frac{1}{N} \sum_{i=1}^N f_k(\vec{x}_i), \tag{4.116}$$

де N – кількість векторів \vec{x}_i у вибірці. Вираз під інтегралом в лівій частині виразу (4.115) за умови $j = k$

$$\int_{\vec{x}} f_k^2(\vec{x}) d\vec{x} = \|f_k(\vec{x})\|^2$$

є квадратом норми k -тої ортогональної функції і для ортонормованих функцій дорівнює одиниці, як було визначено виразом (4.99). Підставляючи (4.99) в ліву частину (4.115), а (4.116) – в праву, отримаємо значення коефіцієнта C_k :

$$C_k = \frac{1}{N} \sum_{i=1}^N f_k(\vec{x}_i), \quad k = 1, 2, \dots, \nu. \quad (4.117)$$

Після визначення коефіцієнтів за допомогою формули (4.96) формуємо оцінку щільності розподілу $\hat{p}(\vec{x})$.

Користуючись результатами розділу 4.4, вираз (4.117) можна записати більш зручно у вигляді рекурентної формули:

$$C_k(N+1) = \frac{1}{N+1} \sum_{i=1}^{N+1} f_k(\vec{x}_i) = \frac{1}{N+1} [NC_k(N) + f_k(\vec{x}_{N+1})], \quad (4.118)$$

де $C_k(N)$ – коефіцієнт, визначений на вибірці обсягом N ;

$C_k(N)$ – вираз для коефіцієнта у разі збільшення вибірки на один об'єкт; $C_k(1) = f_k(\vec{x}_1)$.

Для досягнення задовільної якості апроксимації щільності розподілу $p(\vec{x})$ її оцінкою $\hat{p}(\vec{x})$ потрібно дотримуватися таких міркувань.

По-перше, якість апроксимації за допомогою вибраної системи базисних функцій залежить від кількості ν членів розкладу (4.96). Оскільки дійсний вигляд $p(\vec{x})$ невідомий, то якість оцінки $\hat{p}(\vec{x})$ можна визначити безпосередньо в експерименті з навчальною вибіркою. Якщо точність класифікації є незадовільною, то потрібно пробувати збільшувати кількість ν базисних функцій аж до насичення, тобто до припинення збільшення точності класифікації.

По-друге, за відсутності апріорної обізнаності про характер щільності розподілу базисні функції $f_k(\vec{x})$ потрібно вибирати з погляду простоти реалізації. Якщо ж, наприклад, відомо, що щільність розподілу $p(\vec{x})$ має синусоїдальний характер, то як розклад оцінки потрібно вибирати базисні синусоїдальні функції.

Загальні правила вибору базисних функцій зводяться до того, що за виконання умови лінійної незалежності і деяких інших не дуже жорстких обмежень на вигляд щільності розподілу $p(\vec{x})$ можна довести збіжність

$\hat{p}(\vec{x})^* \rightarrow p(\vec{x})$ за умов $\nu \rightarrow \infty$ і $N \rightarrow \infty$. Ортогональність базисних функцій є окремим випадком лінійної незалежності.

4.7 Алгоритм побудови Байєсівського класифікатора для образів з невідомим законом щільності розподілу ймовірностей ознак

З урахуванням вищенаведених міркувань, за умови невідомого закону щільності розподілу ознак, для побудови ймовірнісного класифікатора для K класів $\Omega_1, \Omega_2, \dots, \Omega_K$ необхідно виконати такі кроки:

1. Вибираємо систему базисних функцій. Наприклад, многочлен Ерміта:

$$H_0(x)=1; H_1(x)=2x; H_{i+1}(x) = 2xH_i(x) + 2iH_{i-1}(x).$$

2. Для k -вимірному простору ознак утворюємо з базисних функцій ортогональні функції k змінних $f_j(\vec{x})$.

3. Визначаємо коефіцієнти C_{ij}

$$C_{ij} = \frac{1}{N_i} \sum_{j=1}^{N_i} f_j(\vec{x}_{ij})$$

4. Знаходимо оцінку функцій щільності розподілу

$$\hat{p}(\vec{x} / \Omega_i) = \sum_{j=1}^N C_{ij} f_j(\vec{x})$$

5. Шукаємо вирішувальну функцію

$$d_i(\vec{x}) = \hat{p}(\vec{x} / \Omega_i) \cdot p(\Omega_i).$$

Якщо на вхід системи розпізнавання надходить вектор опису \vec{x} деякого об'єкта, то класифікацію здійснюємо таким чином – обчислюємо значення всіх вирішувальних функцій $d_i(\vec{x})$, і рішення з розпізнавання приймаємо згідно з правилом:

якщо $d_i(\vec{x}) > d_j(\vec{x})$, для всіх $i, j = \overline{1, K}$, $i \neq j$, то вектор \vec{x} належить i -му класу.

Приклад 4.11. Розробити Байєсівський класифікатор для класів, заданих навчальними вибірками, поданими на рис. 4.8.

Розв'язування: виконаємо оцінення щільності розподілу безпосередньо на заданих на рис. 4.7 навчальних вибірках, скориставшись виразом (4.96):

$$\hat{p}(\vec{x} / \Omega_i) = \sum_{k=1}^m C_{ik} f_k(\vec{x}),$$

в якому перший індекс коефіцієнта визначає клас Ω_i .

За систему базисних функцій $\{f_k(\vec{x})\}$ виберемо многочлени Ермітта, оскільки вони ортогональні на інтервалі $(-\infty, \infty)$. Для одновимірного випадку перші чотири функції мають вигляд (4.109):

$$H_0(x) = 1; H_1(x) = 2x; H_2(x) = 4x^2 - 2; H_3(x) = 8x^3 - 12x;$$

$$H_4(x) = 16x^4 - 48x^2 + 12.$$

Всі вони є ортогональними. Щоб запобігти виникненню обчислювальних складнощів з ілюстраційною метою будемо поводитися з ними, як з ортонормованими.

Для двовимірного простору ознак утворимо з них ортогональні функції двох змінних, користуючись виразами (4.109, 4.111):

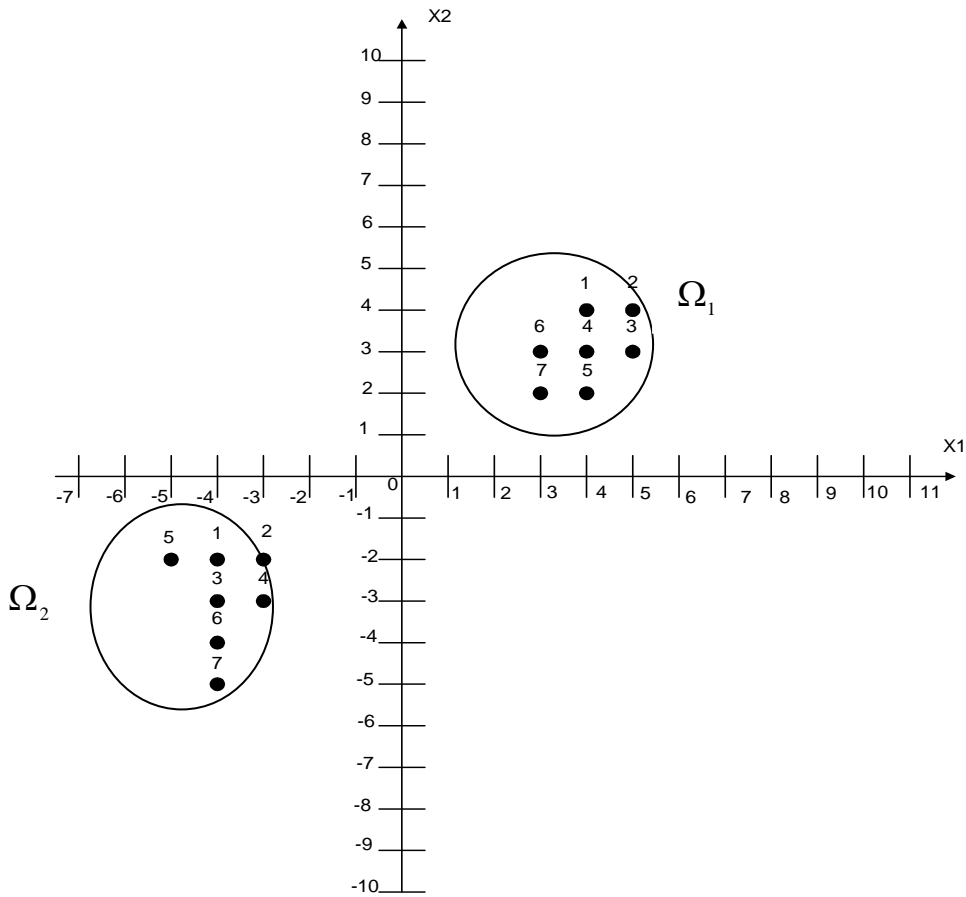


Рисунок 4.8 – Визначення розподільної границі для заданих класів з використанням Байєсівського класифікатора

$$f_1(\vec{x}) = f_1(x_1, x_2) = H_0(x_1) \cdot H_0(x_2) = 1;$$

$$f_2(\vec{x}) = f_2(x_1, x_2) = H_1(x_1) \cdot H_0(x_2) = 2x_1;$$

$$f_3(\vec{x}) = f_3(x_1, x_2) = H_0(x_1) \cdot H_1(x_2) = 2x_2;$$

$$f_4(\vec{x}) = f_4(x_1, x_2) = H_1(x_1) \cdot H_1(x_2) = 4x_1 \cdot x_2.$$

Вибір таких комбінацій функції однієї змінної зумовлений насамперед міркуваннями простоти.

Після цього визначаємо коефіцієнти C_{ik} . За припущенням ортонормованості функцій $f_k(\vec{x})$ їх можна обчислити за виразом (4.116). Для класу Ω_1 :

$$C_{1k} = \frac{1}{N} \sum_{k=1}^{N_1} f_k(\vec{x}_{1k}),$$

де N_1 – кількість об'єктів, які входять в клас Ω_1 , $N_1 = 7$;

k – номер об'єкта, k змінюється від 1 до m , $m = 4$.

Перший індекс вектора \vec{x}_{1k} означає в цьому випадку номер класу.

$$C_{11} = \frac{1}{7} [f_1(\vec{x}_{11}) + f_1(\vec{x}_{12}) + f_1(\vec{x}_{13}) + \dots + f_1(\vec{x}_{17})] = \frac{1}{7} (1 + 1 + \dots + 1) = 1,$$

оскільки $f_1(\vec{x}) = 1$.

$$\begin{aligned} C_{12} &= \frac{1}{7} [f_2(\vec{x}_{11}) + f_2(\vec{x}_{12}) + f_2(\vec{x}_{13}) + \dots + f_2(\vec{x}_{17})] = \\ &= \frac{1}{7} (2 \cdot 4 + 2 \cdot 5 + 2 \cdot 5 + 2 \cdot 4 + 2 \cdot 4 + 2 \cdot 3 + 2 \cdot 3) = \frac{1}{7} (8 + 10 + 10 + 8 + 8 + 6 + 6) = 8, \end{aligned}$$

оскільки $f_2(\vec{x}) = 2x_1$.

$$\begin{aligned} C_{13} &= \frac{1}{7} [f_3(\vec{x}_{11}) + f_3(\vec{x}_{12}) + f_3(\vec{x}_{13}) + \dots + f_3(\vec{x}_{17})] = \\ &= \frac{1}{7} (2 \cdot 4 + 2 \cdot 4 + 2 \cdot 3 + 2 \cdot 3 + 2 \cdot 2 + 2 \cdot 3 + 2 \cdot 2) = \frac{1}{7} (8 + 8 + 6 + 6 + 4 + 6 + 4) = 6, \end{aligned}$$

оскільки $f_3(\vec{x}) = 2x_2$.

$$\begin{aligned} C_{14} &= \frac{1}{7} [f_4(\vec{x}_{11}) + f_4(\vec{x}_{12}) + f_4(\vec{x}_{13}) + \dots + f_4(\vec{x}_{17})] = \\ &= \frac{1}{7} (4 \cdot 4 \cdot 4 + 4 \cdot 5 \cdot 4 + 4 \cdot 5 \cdot 3 + 4 \cdot 4 \cdot 3 + 4 \cdot 4 \cdot 2 + 4 \cdot 3 \cdot 3 + 4 \cdot 3 \cdot 2) = \\ &= \frac{1}{7} (64 + 80 + 60 + 48 + 32 + 36 + 24) = \frac{344}{7} = 49.143, \end{aligned}$$

оскільки $f_4(\vec{x}) = 4x_1x_2$.

Звідси:

$$\begin{aligned} \hat{p}(\vec{x}/\Omega_1) &= C_{11}f_1(\vec{x}) + C_{12}f_2(\vec{x}) + C_{13}f_3(\vec{x}) + C_{14}f_4(\vec{x}) = \\ &= 1 + 6x_1 + 12x_2 + 196.6x_1x_2, \end{aligned}$$

Застосувавши таку саму процедуру, знайдемо коефіцієнти C_{2k} :

$$C_{21} = 7; C_{22} = -7.71; C_{23} = -6; C_{24} = 46.29.$$

Звідси:

$$\begin{aligned} \hat{p}(\vec{x}/\Omega_2) &= C_{21}f_1(\vec{x}) + C_{22}f_2(\vec{x}) + C_{23}f_3(\vec{x}) + C_{24}f_4(\vec{x}) = \\ &= 1 - 7.71x_1 - 6x_2 + 185.16x_1x_2 \end{aligned}$$

В цьому випадку шукані вирішувальні функції мають вигляд:

$$d_1(\vec{x}) = \hat{p}(\vec{x}/\Omega_1) \cdot p(\Omega_1)$$

$$d_2(\vec{x}) = \hat{p}(\vec{x}/\Omega_2) \cdot p(\Omega_2)$$

За припущенням $p(\Omega_1) = p(\Omega_2) = 0.5$.

$$d_1(\vec{x}) = 0.5 + 8x_1 + 6x_2 + 98.3x_1x_2$$

$$d_2(\vec{x}) = 0.5 - 3.86x_1 - 3x_2 + 92.58x_1x_2$$

В цьому випадку рівняння розподільної границі має вигляд:

$$d_1(\vec{x}) - d_2(\vec{x}) = 4.14x_1 + 3x_2 + 6x_1x_2 - d_2(\vec{x}).$$

Контрольні запитання та приклади

1. В якому випадку використовується імовірнісний підхід до класифікації образів?
2. Які помилки прийняття рішень називаються помилками першого і другого роду? Навести приклади.
3. З яких елементів складається статистична гра?
4. Запишіть формулу умовного середнього ризику.
5. В яких випадках для класифікації образів використовується Байєсівська стратегія, а в яких – мінімаксна або Неймана-Пірсона?
6. Запишіть формулу умовного середнього ризику під час класифікації образу до одного з декількох класів.
7. Який вигляд має матриця штрафів (платіжна матриця)?
8. Запишіть Байєсівське правило класифікації для випадку двох класів.
9. Запишіть формулу для відношення і порога правдоподібності. Чому їх так називають?
10. Запишіть Байєсівське правило класифікації для випадку багатьох класів.
11. Запишіть вигляд Байєсівської вирішувальної функції для випадку багатьох класів і правило класифікації за допомогою цієї функції.
12. Зарисуйте структурну схему Байєсівського класифікатора.
13. Які методи оцінення функції щільності розподілу ознак за відомих обмежень вам відомі?
14. Запишіть форму і вигляд коваріаційної матриці ознак класів.
15. Побудуйте Байєсівський класифікатор для класифікації одновимірних зображень на M класів, кожен з яких описується щільністю розподілу Релея:

$$p(x/\Omega_i) = \begin{cases} \frac{x}{\sigma_i^2} \exp\left(-\frac{x}{2\sigma_i^2}\right), & \text{якщо } x \geq 0 \\ 0, & \text{якщо } x < 0 \end{cases}.$$

Вирішувальні функції потрібно будувати за умови, що елементи платіжної матриці набувають значення 0 для правильної класифікації і 1 для неправильної, а апіорні ймовірності образів $p(\Omega_i) = 1/M$.

16. Побудуйте Байєсівський класифікатор для класифікації одновимірних зображень на M класів, кожен з яких описується нормальною щільністю розподілу:

$$p(x/\Omega_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(x - m_i)^2}{2\sigma_i^2}\right]$$

а) Втрати за неправильної класифікації набувають значення 1, за правильної – 0, апіорні ймовірності класів – $p(\Omega_i) = 1/M$.

б) Побудуйте щільності розподілу для задачі розділення на два класи за умов $\sigma_1 = \sigma_2 = 2$, $m_1 = 0$, і $m_2 = 2$. Визначіть положення розподільної границі.

17. Задано класи, що характеризуються нормальним законом розподілу, $\Omega_1: \{ (0,0)^T, (2,0)^T, (2,2)^T, (0,2)^T \}$ і $\Omega_2: \{ (4,4)^T, (6,4)^T, (6,6)^T, (4,6)^T \}$. Знайдіть рівняння Байєсівської розподільної границі для цих класів за умови $p(\Omega_1) = p(\Omega_2) = 0.5$ і побудуйте цю границю.

18. Задано класи, що характеризуються нормальним законом розподілу, $\Omega_1: \{ (-1,0)^T, (0,-1)^T, (1,0)^T, (0,1)^T \}$ і $\Omega_2: \{ (-2,0)^T, (0,-2)^T, (2,0)^T, (0,2)^T \}$. Знайдіть рівняння Байєсівської розподільної границі для цих класів за умови $p(\Omega_1) = p(\Omega_2) = 0.5$ і побудуйте цю границю.

19. Який вигляд має апроксимувальна функція невідомої щільності розподілу випадкової величини \bar{x} ?

20. Які функції називаються лінійно незалежними?

21. Які ортонормовані системи функцій ви знаєте?

22. Яким чином можна побудувати повну систему ортонормованих функцій декількох змінних, якщо задано певну систему ортонормованих функцій однієї змінної?

23. Запишіть формулу обчислення коефіцієнтів оціночної функції.

24 Наведіть кроки алгоритма побудови Байєсівського класифікатора для випадку невідомої щільності розподілу образів.

25. Виконайте оцінення щільностей розподілу $p(x/\Omega_1)$ і $p(x/\Omega_2)$ для класів $\Omega_1: \{ (-5,-5)^T, (-5,-4)^T, (-4,-5)^T, (-5,-6)^T, (-6,-5)^T \}$ і $\Omega_2: \{ (5,5)^T, (5,6)^T, (6,5)^T, (5,4)^T, (4,5)^T \}$. Використайте в цьому випадку чотири перших поліноми двох змінних Ермітта. Побудуйте байєсівські вирішувальні функції за умови $p(\Omega_1) = p(\Omega_2) = 0.5$.

РОЗДІЛ 5 КЛАСТЕРНИЙ АНАЛІЗ ДАНИХ

Задачу побудови алфавіту класів образів вирішують за допомогою кластерного аналізу. Під **кластерним аналізом** або **кластеризацією** в загальному випадку розуміють групування деякої множини об'єктів таким чином, щоб об'єкти однієї групи (яку називають кластером) були більш схожі (за певними характеристиками) між собою, ніж з об'єктами в інших групах (кластерах). 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».

Розв'язання задачі навчання без учителя здійснюється шляхом застосування до заданої вибірки алгоритмів кластеризації. Отримані як результат кластери можна вважати класами зображень. Після визначення класів таким способом їх можна використати для отримання вирішувальних функцій за допомогою алгоритмів контрольованого навчання («навчання з учителем»). Крім того, центри кластерів, отримані на стадії навчання без учителя, можна використовувати безпосередньо для побудови класифікатора, що діє за принципом мінімуму відстані (розділ 3).

Перед тим, як перейти до розгляду конкретних алгоритмів кластеризації, потрібно попередньо зауважити, що більшість з них можна назвати алгоритмами навчання без учителя тільки умовно, оскільки вони використовують апріорні знання розробника («учителя») про властивості структурованості даних, отримані внаслідок експериментів і інтуїції.

5.1 Міри подібності

Ідентифікація класів на заданій вибірці зображень, або кластеризація, полягає в занесенні цих зображень в групи, ґрунтуючись на їх подібності чи розбіжності відповідно до вибраної міри подібності (схожості).

Наприклад, за міру схожості в розділі 3.4 було вибрано евклідову відстань між зображеннями \vec{x} і \vec{y} в просторі ознак:

$$D_E = \|\vec{x} - \vec{y}\| = \sqrt{(\vec{x} - \vec{y})^T (\vec{x} - \vec{y})}. \quad (5.1)$$

Чим менша ця відстань, тим більш схожими є об'єкти. Евклідова відстань D_E задовольняє такі аксіоми:

1. Ідентичності $D_E(\vec{x}, \vec{x}) = 0$; (5.2)

2. Симетрії $D_E(\vec{x}, \vec{y}) = D_E(\vec{y}, \vec{x})$ (5.3)

3. Трикутника для трьох точок \vec{x} , \vec{y} і \vec{z}

$$D_E(\vec{x}, \vec{y}) + D_E(\vec{y}, \vec{z}) \geq D_E(\vec{x}, \vec{z}) \quad (5.4)$$

Міри схожості, що задовільняють умови (5.2), (5.3), (5.4), називаються метричними.

Для n -вимірного простору ознак обчислювальна формула евклідової відстані має вигляд:

$$D_E(\vec{x}, \vec{z}) = \sqrt{(\vec{x} - \vec{z})^T (\vec{x} - \vec{z})} = \sqrt{\sum_{i=1}^n (x_i - z_i)^2}, \quad (5.5)$$

де x_i, z_i – i -ті координати векторів \vec{x}, \vec{z} .

Серед метричних мір схожості, які часто використовуються в задачах розпізнавання, можна назвати такі.

Покоординатне зміщення (відстань Манхеттена, або міська відстань):

$$D_M(\vec{x}, \vec{z}) = \sum_{i=1}^n |x_i - z_i|, \quad (5.6)$$

де $|x_i - z_i|$ – модуль різниці координат x_i і z_i .

Відстань Чебишова:

$$D_{CH}(\vec{x}, \vec{z}) = \max\{|x_i - z_i|\}, \quad i = \overline{1, n}. \quad (5.7)$$

Формули (5.5), (5.6) і (5.7) є окремими випадками функції відстані, яка описується метрикою Мінковського:

$$D_\lambda(\vec{x}, \vec{z}) = \left(\sum_{i=1}^n |x_i - z_i|^\lambda \right)^{\frac{1}{\lambda}}, \quad (5.8)$$

де λ – дійсне число.

В функціях відстані (5.6) ÷ (5.8) компоненти векторів зображень \vec{x} і \vec{z} вважались незалежними. Однак на практиці вони часто є значеннями деякого випадкового процесу, внаслідок чого стають статистично залежними. В цьому випадку використовують відстань Махаланобіса:

$$D_M = \sqrt{[(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})]} \quad (5.9)$$

де $\vec{\mu}$ є усередненим вектором заданих вибірових зображень, а Σ^{-1} – обернена коваріаційна матриця векторів цих зображень.

По суті відстань Махаланобіса є зваженою відстанню Евкліда (див. (5.9) і (5.5)), в якій ваги вибірових векторів відповідають їх апріорно визначеним важливостям, заданих середньоквадратичним відхиленням їхніх координатних компонент.

На рис. 5.1 показано на площині зображення відстаней, що відповідають описаним вище мірам схожості.

На рис. 5.1, а) зображено гіперкулю. Відстані всіх векторів \vec{x} всередині гіперкулі від її центра \vec{z} визначаються Евклідовою відстанню $r = D_E \leq |\vec{x} - \vec{z}|$ (формула 5.1).

На рис. 5.1, б) відстані всіх векторів \vec{x} , що лежать всередині гіперкуба, від його центра \vec{z} визначаються відстанню Манхеттена $r = D_1 \leq |x_1 - z_1| + |x_2 - z_2| = r_1 + r_2$. Вектори \vec{w} і \vec{u} також віддалені від \vec{z} на відстань r .

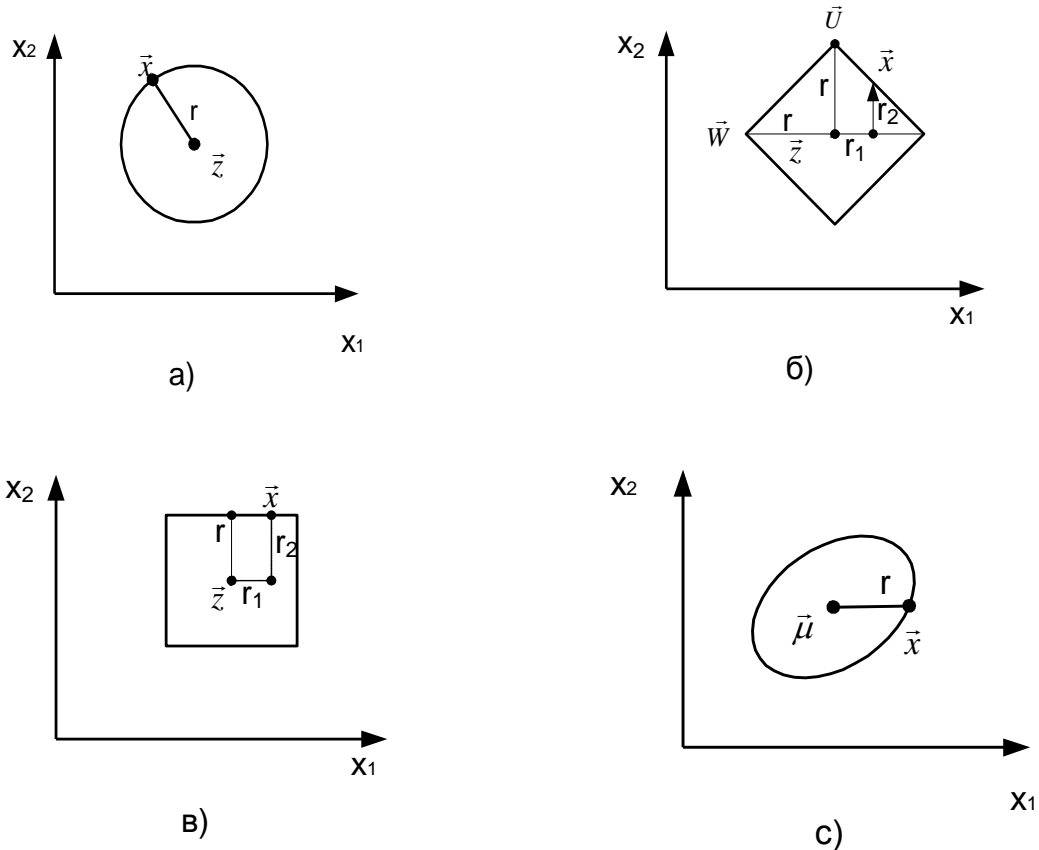


Рисунок 5.1 – Зображення відстаней, що відповідають введеним мірам схожості: а) $r = D_E$; б) $r = r_1 + r_2 = D_1$; в) $r = \max\{r_1, r_2\} = D_{CH}$;

$$\text{г) } r = \sqrt{\delta_{11}(x_1 - \mu_1)^2 + \delta_{12}(x_1 - \mu_1)(x_2 - \mu_2) + \delta_{22}(x_2 - \mu_2)^2} = D_M.$$

На рис. 5.1, в) відстані всіх векторів \vec{x} , що лежать всередині гіперквадрата, від його центра визначаються відстанню Чебишова

$$r = D_{CH} = \max\{r_1, r_2\}.$$

На рис. 5.1, г) відстані всіх векторів \vec{x} , що лежать всередині гіпереліпсоїда, від його центра визначаються відстанню Махаланобіса

$$r = D_M \leq \sqrt{\delta_{11}(x_1 - \mu_1)^2 + \delta_{12}(x_1 - \mu_1)(x_2 - \mu_2) + \delta_{22}(x_2 - \mu_2)^2},$$

де δ_{11} і δ_{22} – дисперсії,

$\delta_{12} = \delta_{21}$ – коваріації відповідних координатних компонент векторів x_1 і x_2 ,

μ_1 і μ_2 – координати вектора $\vec{\mu}$.

Для всіх вказаних випадків знак « \Leftrightarrow » у виразі для відстані відповідає розміщенню вектора на гіперповерхні.

Корисною неметричною мірою для визначення схожості двох сигналів, заданих векторами \vec{x} і \vec{z} , є косинус кута між ними, що визначається формулою:

$$D(\vec{x}, \vec{z}) = \cos\theta = \frac{\vec{x}^T \vec{z}}{\|\vec{x}\| \|\vec{z}\|}, \quad (5.9, a)$$

де $\|\vec{x}\|$ означає довжину вектора \vec{x} : $\|\vec{x}\| = \sqrt{\sum_{i=1}^n x_i^2}$.

З рисунка 5.2 видно, що функція схожості (5.9, а) досягає максимального значення в тому випадку, коли напрямки векторів \vec{x} і \vec{z} збігаються, і $\cos\theta = 1$.

$$\cos\theta_1 = \frac{\vec{x}^T \vec{z}_1}{\|\vec{x}\| \|\vec{z}_1\|}$$

$$\cos\theta_2 = \frac{\vec{x}^T \vec{z}_2}{\|\vec{x}\| \|\vec{z}_2\|}$$

$$\cos\theta_1 > \cos\theta_2$$

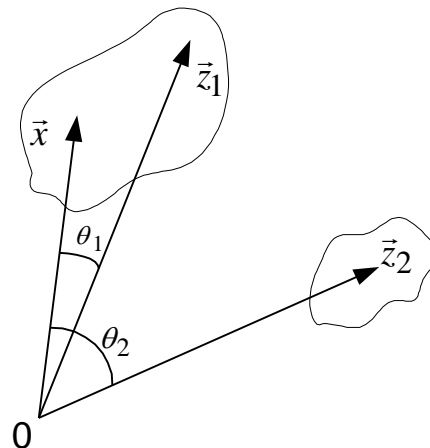


Рисунок 5.2 – Геометрична ілюстрація косинусної міри схожості

За цієї умови вектор \vec{z} дорівнює вектору \vec{x} , помноженому на довільний скаляр, тобто $\vec{z} = \alpha \cdot \vec{x}$. Якщо $\cos\theta = 0$, то говорять, що вектори \vec{x} і \vec{z} ортогональні. Описана міра часто використовується для ідентифікації акустичних спектрів (наприклад, в задачах розпізнавання мовної інформації).

Двійковим варіантом формули (5.9, а) є так звана міра схожості Танімото:

$$D_T(\vec{x}, \vec{z}) = \frac{\vec{x}^T \vec{z}}{\vec{x}^T \vec{x} + \vec{z}^T \vec{z} - \vec{x}^T \vec{z}}. \quad (5.10)$$

Якщо координатами векторів \vec{x} і \vec{z} є 1 або 0, які означають наявність або відсутність i -тої ознаки в об'єктах, то формула (5.10) визначає відношення кількості однакових ознак до кількості різних ознак в зображеннях \vec{x} і \vec{z} . Це дуже легко уявити з діаграм Венна на рис. 5.3.

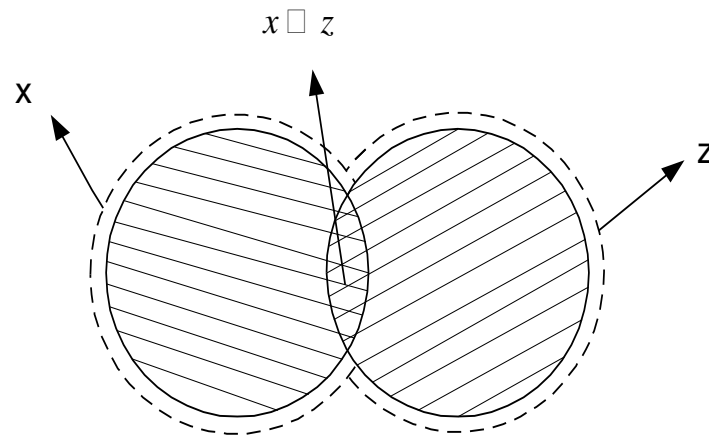


Рисунок 5.3 – Ілюстрація міри Танімото діаграмами Венна:

\mathbf{X} – множина ознак в векторі \vec{x} ; \mathbf{Z} – множина ознак в векторі \vec{z} ;
 $n(\mathbf{X})$ $n(\mathbf{X})$ – кількість ознак в множині \mathbf{X} ; $n(\mathbf{Z})$ – кількість ознак в множині \mathbf{Z} ;
 $n(\mathbf{X} \cap \mathbf{Z})$ – кількість спільних ознак в множинах \mathbf{X} і \mathbf{Z}

Для визначення схожості об'єктів, описаних двійковими векторами, часто також використовують відстань Хеммінга:

$$D_n(\vec{x}, \vec{z}) = \sum_{i=1}^n x_i \oplus z_i, \quad (5.11)$$

де символ \oplus означає логічну операцію «виключне АБО» або «сума за модулем 2». Вираз (5.11) можна записати за допомогою операцій алгебри логіки у вигляді:

$$D_n(\vec{x}, \vec{z}) = \sum_{i=1}^n (\bar{x}_i \wedge z_i \vee x_i \wedge \bar{z}_i), \quad (5.12)$$

який визначає кількість незбіжних розрядів в n – компонентних двійкових векторах \vec{x} і \vec{z} . Наприклад, для $\vec{x} = (0,0,1,1,0,1)$ і $\vec{z} = (1,0,1,0,1,1)$ $D_n(\vec{x}, \vec{z}) = 3$.

5.2 Загальні принципи і критерії кластеризації

В розділі 2.1 було визначено, що кластером в просторі ознак називається група точок (зображень об'єктів), які є більш близькими одна до одної і до їхнього центра, ніж до центрів інших груп. Узагальнена процедура кластеризації заданої вибірки векторів, згідно з К. S. Fu [], складається з таких кроків.

Крок 1. Розбиття множини вибіркових точок $S = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_a\}$ на K пробних підмножин (кластерів) шляхом використання відповідної міри схожості.

Крок 2. Тестування якості отриманого розбиття, що ґрунтується на вибраному критерії кластеризації.

Крок 3. Припинення кластеризації в випадку задовільного значення критерію, в протилежному випадку – перехід до Кроку 4.

Крок 4. Перерозподіл отриманих кластерів: розбиття тих, всередині яких спостерігається значна відмінність зображень і/або об'єднання між собою тих, в яких спостерігається подібність зображень; перенесення точок до інших класів і перехід до кроку 2.

Як витікає з описаного узагальненого алгоритму кластеризації, для тестування якості отриманого розподілу вибіркових точок на класи потрібно ґрунтуватися на вибраному належним чином критерії кластеризації. Одним з найбільш популярних показників якості кластеризації є середньоквадратичне відхилення:

$$J = \sum_{i=1}^{N_m} \sum_{\bar{x} \in \Omega_i} \|\bar{x} - \bar{\mu}_i\|^2, \quad (5.13)$$

де N_m – кількість кластерів,

Ω_i – множина точок, що відносяться до i -го класу, а

$$\bar{\mu}_i = \frac{1}{N_i} \sum_{\bar{x} \in \Omega_i} \bar{x}, \quad (5.14)$$

де μ_i – вектор вибіркових середніх значень для класу Ω_i ;

N_i – кількість зображень в класі Ω_i .

Критерій кластеризації (5.13) визначає загальну суму квадратів відхилень ознак всіх зображень, що входять до деякого кластера, від відповідних середніх значень по кластеру. Розроблювані процедури кластеризації мають забезпечити мінімізацію цього показника якості.

Крім критерію (5.13) існує велика кількість інших показників, наприклад, середнє квадратів відстаней між точками в кластері; середнє квадратів відстаней між точками, що входять до різних кластерів; мінімум і максимум дисперсії тощо.

Нерідко використовуються алгоритми знаходження кластерів, які ґрунтуються на сумісному використанні евристичного підходу та формального критерію кластеризації (наприклад, алгоритм ISODATA). Якість багатьох окремих алгоритмів значною мірою визначається здібностями їх розробників до здобування корисної інформації з даних, що аналізуються.

5.3 Алгоритм кластеризації за пороговою відстанню

Нехай задано вибірку, що містить Q зображень $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_q\}$. Задамо початковий центр першого кластера \vec{z}_1 , наприклад, $\vec{z}_1 = \vec{x}_1$, і визначимо довільно невід’ємну порогову величину T . Після цього подаємо на вхід алгоритму вектор \vec{x}_2 і обчислюємо за формулою (5.5) відстань $D(\vec{x}_2, \vec{z}_1)$. Якщо ця відстань більша значення порогової величини, то започатковуємо новий центр кластера $\vec{z}_2 = \vec{x}_2$. В іншому випадку зображення \vec{x}_2 заноситься в кластер з центром \vec{z}_1 . Припустимо, що $D(\vec{x}_2, \vec{z}_1) > T$, тобто \vec{z}_2 є центром нового кластера. На наступному кроці обчислюють відстані $D(\vec{x}_3, \vec{z}_1)$ і $D(\vec{x}_3, \vec{z}_2)$. Якщо обидві відстані більше порога T , започатковується новий центр кластера $\vec{z}_3 = \vec{x}_3$. В іншому випадку вектор \vec{x}_3 зараховується до того кластера, чий центр до нього ближче. Подібним чином обчислюються відстані від кожного нового вибіркового зображення \vec{x}_q до кожного відомого k -го центра кластера Ω_k і порівнюються з пороговою величиною – якщо ці всі відстані більші значення порогу T , утворюється новий центр кластера $\vec{z}_{k+1} = \vec{x}_q$. В іншому випадку зображення зараховується в кластер з найближчим до нього центром. Результати описаної процедури визначаються вибором першого центра кластера, порядком огляду зображень, значенням порогової величини T і геометричною структурою даних. На рис. 5.4 показано три різних варіанти вибору центрів кластерів для однієї і тієї самої вибірки даних, які виникли внаслідок зміни тільки значення порога T і початкової точки кластеризації.

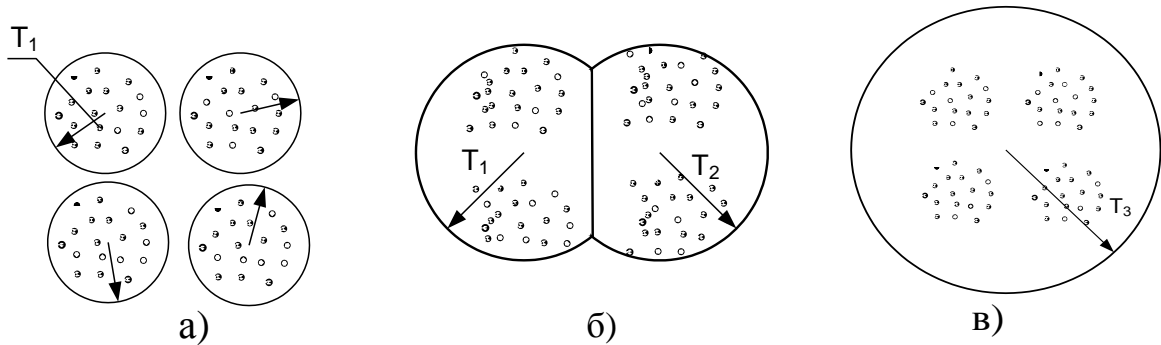


Рисунок 5.4 – Геометрична ілюстрація впливу вибору величини порога і початкових центрів в пороговій схемі кластеризації

До переваг цього алгоритму потрібно віднести можливість просто і швидко отримати наближені оцінки основних характеристик заданої вибірки даних. Він також є привабливим з погляду невеликих обчислювальних затрат, оскільки може визначити центри кластерів за один прохід. На практиці ж, для отримання задовільних результатів кластеризації, потрібно багаторазово застосовувати до даних алгоритм за різних значень порога і з різними початковими точками кластеризації.

Отриману якість кластеризації визначають, наприклад, шляхом порівняння положення нових і попередніх кластерів. Якщо їх зміщення менше заданої величини, то процедуру кластеризації припиняють. Алгоритм є збіжним для лінійно розділюваних класів.

Нижче наводиться детальний опис порогового алгоритму кластеризації.

Вхідні дані: випадкова вибірка ознакових векторів зображень $\{ \vec{x}_1, \vec{x}_2, \dots, \vec{x}_q, \dots, \vec{x}_Q \}$; значення порога T ; коефіцієнт пропорційності α .

Результати: K_p – кількість класів; Ω_k – k -тий клас, утворюється індексною функцією $Class(q) \leftarrow k$, яка присвоює вектору \vec{x}_q індекс k -го класу; $n(k)$ – кількість векторів в кожному k -ому класі ($k=1, \dots, K$); \vec{z}_k – центри (еталонні вектори кожного класу); σ_k^2 – дисперсія кожного класу

($\sigma_k^2 = \frac{1}{n(k)} \|\vec{x}_q - \vec{z}_k\|^2$); σ_T^2 – загальне середньоквадратичне відхилення

(дисперсія), $\sigma_T^2 = \sum_{k=1}^k \sigma_k^2$.

Алгоритм 5.1

Порогова кластеризація

```

 $K_p := 1$ ;  $\vec{z}_1 := \vec{x}_1$ ; /*Ініціалізація. Визначення центра першого кластера */
 $Class(1) := 1$ ;
 $Flag := True$ ; /*Установлення прапорця зупинки алгоритму */
while  $Flag$  do begin /*Цикл однієї ітерації алгоритму*/
  for  $k := 1$  to  $K_p$  do /*Цикл обнулення лічильників кількості елементів
     $n(k) := 0$ ; в класах*/
    for  $q := 1$  to  $Q$  do /*Цикл подання векторів вибірки*/
      begin  $d_{min} := 999999.9$ ; /*Знаходження мінімальної відстані від точки
        for  $k := 1$  to  $K_p$  do  $\vec{x}_q$  до центрів кластерів*/
          begin  $d := \|\vec{x}_q - \vec{z}_k\|$ ;
            if  $d < d_{min}$  then begin
               $d_{min} := d$ ;  $k_{min} := k$  end;
            end; /* Кінець циклу за  $k$  */
          if  $d_{min} > T$  then begin /*Якщо відстань більша за поріг,
             $K_p := K_p + 1$ ;  $\vec{z}_k := \vec{x}_q$ ; призначити новий центр */
             $Class(q) := K_p$ ; /*Віднести  $\vec{x}_q$  в новий клас */
             $n(K) := n(K_p) + 1$  end; /*Інкрементувати лічильник*/
          if  $d_{min} < \alpha T$  then begin /*Якщо  $\vec{x}_q$  близький до центра*/
             $Class(q) := k_{min}$ ; /*Віднести  $\vec{x}_q$  в клас  $\Omega_{k_{min}}$  */
          end;
        end;
      end;
    end;
  end;
end;

```

```

n(kmin) := n(kmin) + 1 end; /* Інкрементувати лічильник */
Call Update (kmin); /*Процедура поновлення центрів
кластерів і дисперсій кластерів */
end; /*Кінець циклу за q*/
for k := 1 to Kp do begin /*Цикл визначення величини зміщення нових
центрів кластерів відносно попередніх */
d := ||z̄k* - z̄k||;
if d > 0.0005 then /*Установлення прапорця зупинення алгоритму
в ознаку припинення алгоритму */
Flag := True else Flag := False
end /* Кінець циклу за k */
end; /* Кінець циклу while */

```

Процедура оновлення статистики переобчислює значення вектора середніх значень \bar{z}_k і дисперсії σ_k^2 згідно з формулами (4.50) і (4.51) відповідно до підрозділу 4.6:

$$\bar{z}_k \leftarrow \frac{[(n(k)-1)z_k + \bar{x}_q]}{n(k)}; \quad \sigma_k^2 \leftarrow \frac{[(n(k)-1)\sigma_k^2 + (\bar{x}_q - \bar{z}_k)^2]}{n(k)}.$$

На практиці алгоритм потрібно повторити багато разів з $\alpha = 0.5$ (якщо невідома апріорна інформація, яка дозволить визначити інше значення α) для різних значень T . Вибирають таке значення порога T , яке дає мінімальне значення загальної середньоквадратичної похибки.

Приклад 5.1. Для заданої вибірки векторів $\{\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5, \bar{x}_6\} = \{(2,1), (3,1), (2,4), (3,2), (2,3), (1,3)\}$ виконати кластеризацію за допомогою порогового алгоритму. Геометричне зображення заданої вибірки даних показано на рис. 5.5.

Розв'язування. Виберемо поріг $T=2.9$ і коефіцієнт пропорційності $\alpha = 0,5$ і виконаємо всі кроки, описані в пороговому алгоритмі.

Виберемо центр першого кластера $\bar{z}_1 = \bar{x}_1$. Точка \bar{x}_2 знаходиться до центра на відстані, меншій $\alpha T = 1.45$, тому ми відносимо її до класу Ω_1 за допомогою індексної функції $Class(2) = 1$. Відстань точки \bar{x}_3 до \bar{z}_1 $d(\bar{x}_3, \bar{z}_1) = \sqrt{(2-2)^2 + (4-1)^2} = 3$ більша за поріг $T=2.9$, тому започатковуємо новий центр кластера $\bar{z}_2 = \bar{x}_3, Class(1) = 2$. Знаходимо відстані $d(\bar{x}_4, \bar{z}_1)$ і $d(\bar{x}_4, \bar{z}_2)$ і вибираємо мінімальну:

$$d(\bar{x}_4, \bar{z}_1) = \sqrt{(3-2)^2 + (2-1)^2} \approx 1.41; d(\bar{x}_4, \bar{z}_2) = \sqrt{(3-2)^2 + (2-4)^2} = \sqrt{5} \approx 2.24.$$

Відстань від точки \bar{x}_4 до \bar{z}_1 найменша, $d_{\min} = d(\bar{x}_4, \bar{z}_1)$, тому точку \bar{x}_4 зараховуємо в клас Ω_1 : $Class(4) = 1$. Продовжуємо алгоритм для точок \bar{x}_5 і

\bar{x}_6 , отримуємо $\text{Class}(5)=2$, $\text{Class}(6)=2$. Таким чином, перший цикл алгоритму визначив класи $\Omega_1 = \{\bar{x}_1, \bar{x}_2, \bar{x}_4\}$ і $\Omega_2 = \{\bar{x}_3, \bar{x}_5, \bar{x}_6\}$. Шляхом усереднення векторів в кожному класі отримуємо координати їх еталонних центрів: $\bar{z}_1^* = (2.66, 1.33)$ і $\bar{z}_2^* = (1.66, 3.33)$, які на рис. 5.5 позначено символом «*».

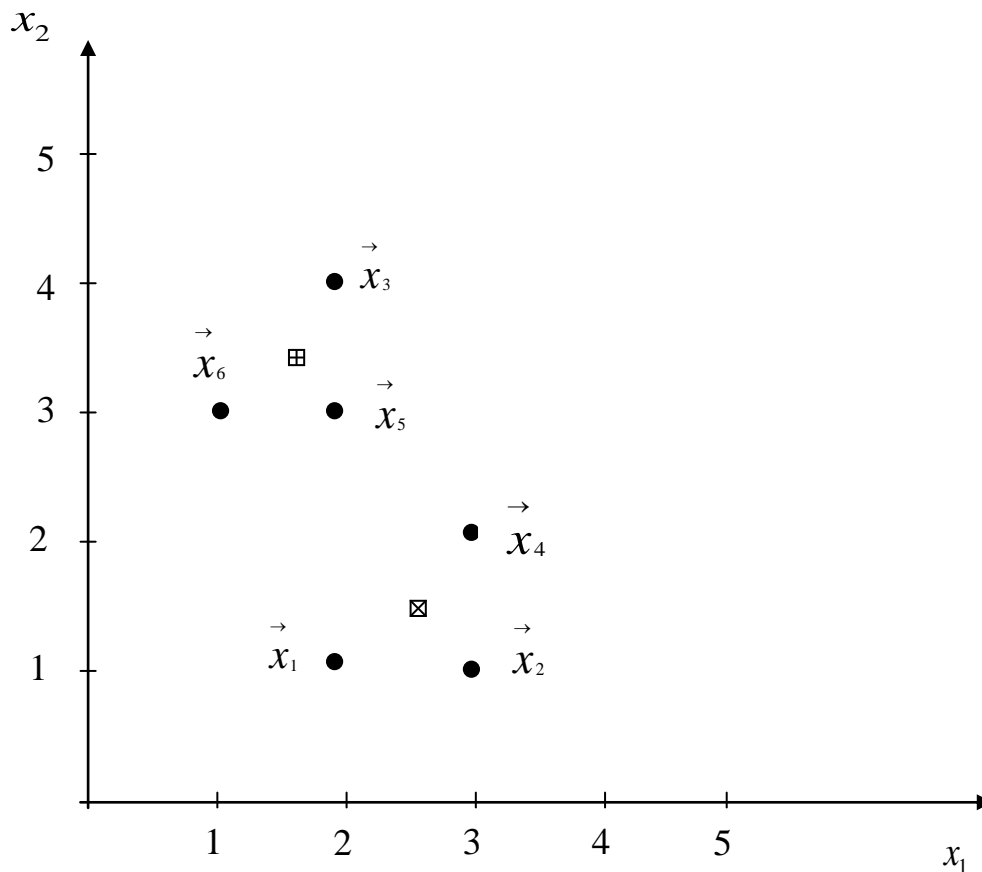


Рисунок 5.5 – Кластеризація даних пороговим алгоритмом для прикладу 5.1

Застосування ще однієї ітерації не змінює кластери, тому роботу алгоритму припиняємо.

5.4 Алгоритм максимінної відстані (алгоритм кластеризації Бетчелора-Уілкінса)

Алгоритм максимінної (максимально-мінімальної) відстані реалізує нескладну евристичну процедуру визначення класів і теж використовує евклідову відстань. На відміну від порогового алгоритму він насамперед виявляє найбільш віддалені кластери. Початковими даними до алгоритму є вибірка розміщених в просторі випадковим чином ознак векторів зображень $\{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_Q\}$ і коефіцієнт пропорційності α ($0 < \alpha < 1$), що використовується під час прийняття рішень. Початково приймають $\alpha = 0,5$.

На першому кроці алгоритму один із об'єктів, наприклад \vec{x}_1 , призначається центром першого кластера: $\vec{z}_1 = \vec{x}_1$. Наступним кроком він знаходить максимальну відстань від усіх інших зображень до знайденого центра кластера: $d_{\max} = \max \{\|\vec{x}_q - \vec{z}_1\|\}, q > 1$. Точка \vec{x}_q , яка відповідає цій відстані, вважається центром нового кластера: $\vec{z}_2 = \vec{x}_q$. Далі, для всіх точок \vec{x}_q , що не стали центрами кластерів, знаходять пари відстаней $(\|\vec{x}_q - \vec{z}_1\|, \|\vec{x}_q - \vec{z}_2\|)$, і в кожній парі виділяють мінімальну відстань d_{\min} . Після цього з цих мінімальних відстаней вибирають максимальну $d_{\max} = \max \{d_{q\min}\}$. Точку, що відповідає цій відстані, позначають як \vec{x}_q^* . Якщо ця максимальна відстань становить значну частину відстані між центрами кластерів \vec{z}_1 і \vec{z}_2 , $d_{\max} > \alpha \cdot \|\vec{z}_1 - \vec{z}_2\|$, то точка \vec{x}_q^* призначається центром нового кластера: $\vec{z}_3 = \vec{x}_q^*$. В іншому випадку виконання алгоритму припиняється за кількості кластерів $K = 2$ з центрами \vec{z}_1 і \vec{z}_2 .

На наступному кроці алгоритму обчислюють відстані між знайденими центрами і кожною з інших вибіркових точок: $d_{q1} = \{\vec{x}_q - \vec{z}_1\}$, $d_{q2} = \{\vec{x}_q - \vec{z}_2\}$, $d_{q3} = \{\vec{x}_q - \vec{z}_3\}$. В кожній такій групі з трьох відстаней вибирають мінімальну: $d_{q\min} = \min \{d_{q1}, d_{q2}, d_{q3}\}$, після чого з цих мінімальних відстаней вибирають максимальну. Вектор зображень \vec{x}_q^{**} , що відповідає цій максимальній відстані, вважається кандидатом на центр нового кластера. Якщо ця відстань $\|\vec{x}_q^{**} - \vec{z}_k\|$, $k=1$ або $k=2$, більша «типової», то точка \vec{x}_q^{**} називається новим центром $\vec{z}_4 = \vec{x}_q^{**}$.

«Типова» відстань визначається як середнє арифметичне всіх відстаней між знайденими центрами:

$$d_{\min} = \alpha \cdot \frac{d(\vec{z}_1, \vec{z}_2) + d(\vec{z}_1, \vec{z}_3) + d(\vec{z}_2, \vec{z}_3)}{3}. \quad (5.15)$$

Після припинення процедури, яка знаходить k центрів кластерів $\vec{z}_1, \vec{z}_2, \dots, \vec{z}_k$, всі інші вектори зображень (ті, які не стали центрами), заносяться в кластери, центри яких лежать на мінімальній відстані. Іншими словами, такий вектор \vec{x}_q заноситься в кластер Ω_k , якщо він розташований найближче до центра \vec{z}_k .

На рис. 5.6 наведено геометричне пояснення максимінної процедури.

На цьому рисунку вектори \vec{z}_1, \vec{z}_2 і \vec{z}_3 є визначеними на цьому кроці центрами кластерів, а вектори \vec{x}_q і \vec{x}_{q+1} є непозначеними векторами.

Відстань $d_{q\min}$ є мінімальною із відстаней від точки \vec{x}_q до всіх центрів знайдених кластерів. На цьому рисунку вона є відстанню між \vec{x}_q і \vec{z}_1 . Так само, відстань $d_{(q+1)\min}$ до центра \vec{z}_3 є мінімальною в групі із трьох відстаней від точки \vec{x}_{q+1} до центрів кластерів \vec{z}_1 , \vec{z}_2 і \vec{z}_3 . Максимальною серед відстаней $d_{q\min}$ і $d_{(q+1)\min}$ є відстань $d_{(q+1)\min}$, тобто $d_{\max} = \|\vec{x}_{q+1} - \vec{z}_3\|$. Якщо ця відстань d_{\max} більша в α разів від середньої, тобто, якщо виконується умова

$$d_{\max} > \alpha \cdot \frac{d(\vec{z}_1, \vec{z}_2) + d(\vec{z}_1, \vec{z}_3) + d(\vec{z}_2, \vec{z}_3)}{3}, \quad (5.16)$$

то \vec{x}_{q+1} позначається як новий вектор \vec{z}_4 .

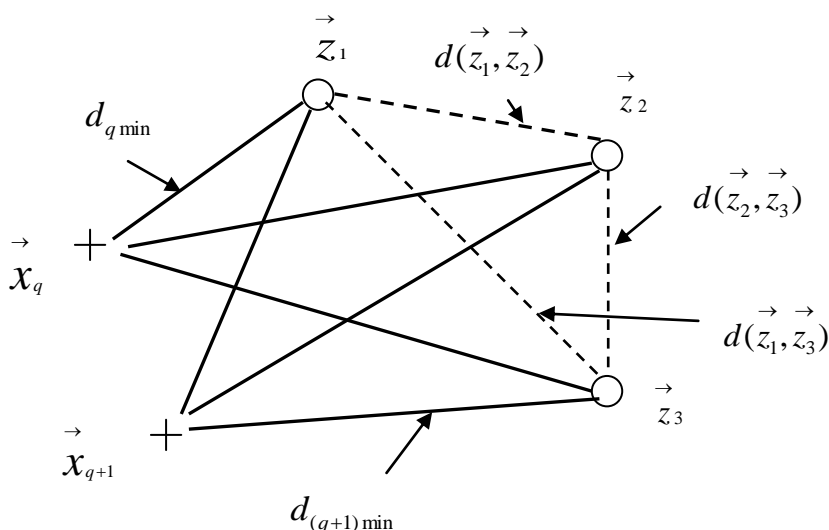


Рисунок 5.6 – Ілюстрація максимінної процедури кластеризації

В іншому випадку центр не призначається і подальші ітерації його визначити не зможуть. Тому процедура пошуку нових центрів припиняється.

Нижче наведено детальний опис максимінної процедури кластеризації. Її входами і виходами є такі величини.

Вхідні дані: послідовність вибірковок векторів зображень $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_q, \dots, \vec{x}_Q\}$ і коефіцієнт пропорційності α ($0 < \alpha < 1$).

Вихідні результати: число кластерів k ; класи $\Omega_1, \dots, \Omega_k$, отримані шляхом присвоєння точкам \vec{x}_q індексів k , $class(q) = k$, що означає їх належність до кластерів Ω_k ; $n(k)$ – кількість точок (векторів зображень) в k -му кластері Ω_k ; $cc(q)$ – індексна функція, яка визначає центри кластерів, тобто, що \vec{x}_q є центром кластера Ω_k .

Алгоритм 5.2 Максимінна кластеризація

```

for  $q:=1$  to  $Q$  do  $cc(q):=0$ ; /* Обнулення індексної функції */
 $\bar{z}_1:=\bar{x}_1$ ;  $K:=1$ ;  $n(K):=0$ ; /* Призначення першого вектора
 $cc(1):=1$ ;  $class(1):=1$ ; /* центром першого кластера */
 $d_{\max}:=0.0$ ;
for  $q:=2$  to  $Q$  do begin /*Знаходження відстаней від точок до
 $d:=\|\bar{x}_q - \bar{z}_k\|$ ; /* центра I кластера */
if  $d>d_{\max}$  then begin /*Знаходження точки з максимальною
 $d_{\max}:=d$ ;  $q_{\max}:=q$ ; /* відстанню від I центра */
end;
end;
 $K:=K+1$ ;  $\bar{z}_k:=\bar{x}_{q_{\max}}$ ; /*Призначення центра і індексу кластера*/
 $class(q_{\max}):=K$ ;  $n(K):=n(K)+1$ ; /*Переобчислення*/
 $flag:=true$ ; /*Установлення умови повторення циклу*/
while  $flag$  do begin
 $d_{\max}:=0.0$ ;  $d_{\min}:=999999.9$ ; /*Ініціалізація  $d_{\max}$  і  $d_{\min}$  */
for  $q:=1$  to  $Q$  do
if  $cc(q)=0$  then begin /*Визначення точок, що не є центрами*/
for  $k1:=1$  to  $K$  do begin /*Цикл визначення відстаней від
 $d(q,k1):=\|\bar{x}_q - \bar{z}_{k1}\|$ ; точки до всіх центрів кластерів*/
if  $d(q,k1)<d_{\min}$  then begin /*Визначення мінімальної відстані до
 $d_{\min}:=d(q,k1)$ ; центра і номер відповідної точки*/
 $q_{\min}:=q$ ; end
end; /*Кінець циклу по k*/
if  $d_{\min}>d_{\max}$  then begin
 $d_{\max}:=d_{\min}$ ; /*Знаходження максимальної відстані та її
 $q_{\max}:=q_{\min}$ ; /* індексу*/
end;
end; /*Кінець циклу за Q*/
 $average:=0.0$ ; /*Початкове значення типової відстані між
центрами*/
for  $k1:=1$  to  $K-1$  do /*Цикл знаходження типової відстані між
центрами кластерів*/
for  $k2:=k1+1$  to  $K$  do
 $average:=average + \|\bar{z}_{k1} - \bar{z}_{k2}\|$ ;
 $average:=average/K$ ;
if  $d_{\max}>d \cdot average$  then begin /*Якщо  $d_{\max}$  більша «типової» відстані,
призначити новий центр кластера і його індекс*/
 $K:=K+1$ ;  $cc(q_{\max}):=K$ ;  $\bar{z}_k:=\bar{x}_{q_{\max}}$ ;
end
else
 $flag:=false$ ; /*Інакше центрів нових кластерів більше немає*/
end; /*Кінець циклу while*/
call procedure Assign_Vect;

```

Процедура Assign_Vect відносить всі вибірккові вектори \vec{x}_q , що не є центрами кластерів, до тих кластерів, центри яких до цих точок є найближчими. Ця процедура детально описана в кроці 2 алгоритму K – внутрішньогрупових середніх, наведених в підрозділі 5.5.

Описаний евристичний алгоритм має такі переваги:

- він не потребує «учителя», тобто є неконтрольованим;
- визначає кількість центрів кластерів K ;
- є ефективним для випадку невеликої кількості кластерів.

До недоліків цього алгоритму можна віднести:

- необхідність визначення коефіцієнта пропорційності α і вплив його величини на результати кластеризації;
- порядок слідування вибірккових векторів впливає на результати;
- можливість використання тільки для лінійно розділених класів;
- центри кластерів не переобчислюються для їх уточнення;
- відсутність критерію якості класифікації.

На практиці процедуру потрібно повторити для декількох значень α , обчислити дисперсії точок σ_k^2 для кожного класу і на основі їхніх значень знайти загальну середньоквадратичну похибку для всієї вибірки точок (векторів зображень).

Оптимальним вважається таке значення α , яке мінімізує значення загальної середньоквадратичної помилки. Крім того, потрібно знайти середнє значення $\bar{\mu}_k$ кожного кластера для визначення нових центрів і знову віднести всі випадкові вектори до кластерів за критерієм мінімальної відстані від нових центрів.

Приклад 5.2. Для заданої вибірки векторів зображень (рис. 5.7) виконати кластеризацію за допомогою алгоритму максимальної відстані.

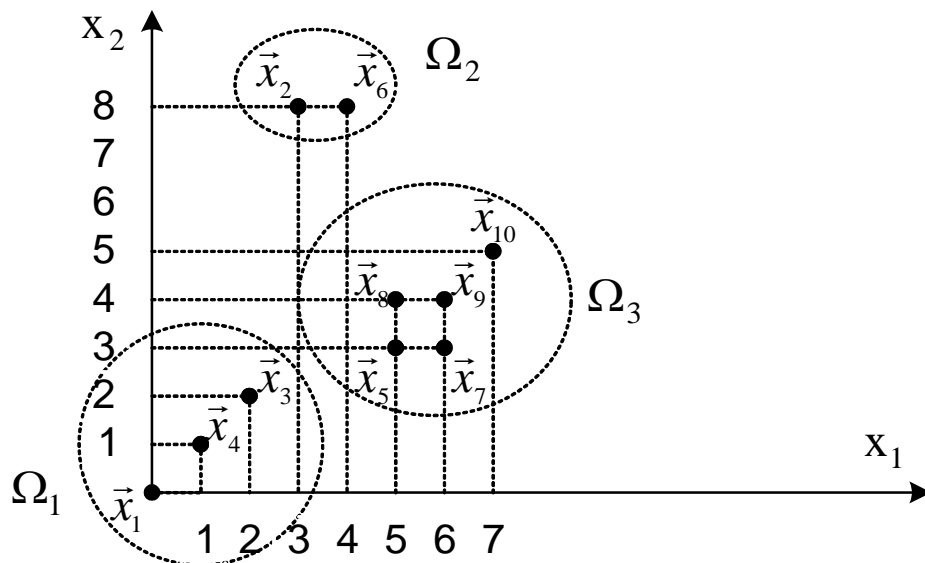


Рисунок 5.7 – Кластеризація зображень алгоритмом максимальної відстані x_1

Розв'язування: виберемо коефіцієнт пропорційності $\alpha = 0.5$ і виконаємо кроки, передбачені алгоритмом максимальної кластеризації.

Виберемо центр першого кластера $\vec{z}_1 = \vec{x}_1$, після цього знаходимо відстані від цих точок до першого центра:

$$d(\vec{x}_q, \vec{z}_1) = \|\vec{x}_q - \vec{z}_1\|.$$

Найбільшу відстань від центра \vec{z}_1 має зображення \vec{x}_6 .

Визначаємо відстані від всіх непозначених точок до центрів \vec{z}_1 і \vec{z}_2 і визначаємо мінімальні відстані в кожній парі відстаней:

| | | |
|------------------|------------------|------------------------------|
| $d(2,1) = 8.54$ | $d(2,2) = 1$ | $d_{2\min} = d(2,2) = 1$ |
| $d(3,1) = 2.83$ | $d(3,2) = 6.32$ | $d_{3\min} = d(3,1) = 2.83$ |
| $d(4,1) = 1.41$ | $d(4,2) = 7.62$ | $d_{4\min} = d(4,1) = 1.41$ |
| $d(5,1) = 5.83$ | $d(5,2) = 5.10$ | $d_{5\min} = d(5,2) = 5.10$ |
| $d(7,1) = 6.71$ | $d(7,2) = 5.39$ | $d_{7\min} = d(7,2) = 5.39$ |
| $d(8,1) = 6.40$ | $d(8,2) = 4.12$ | $d_{8\min} = d(8,2) = 4.12$ |
| $d(9,1) = 7.20$ | $d(9,2) = 4.47$ | $d_{9\min} = d(9,2) = 4.47$ |
| $d(10,1) = 8.60$ | $d(10,2) = 4.24$ | $d_{10\min} = d(10,2) = 4.2$ |

Серед мінімальних відстаней вибираємо максимальну і визначаємо індекс:

$$d_{\max} = \max\{d_{q\min}\} = d(7,2) = 5.39; \quad q_{\max} = 7.$$

Отже, точка $\vec{x}_{q_{\max}} = \vec{x}_7$ відповідає знайденому максимуму.

Обчислюємо відстань між центрами \vec{z}_1 і \vec{z}_2 :

$$d(\vec{z}_1, \vec{z}_2) = d(\vec{x}_1, \vec{x}_6) = \sqrt{(0-4)^2 + (0-8)^2} \approx 8.45.$$

Перевіряємо умову $d_{\max} > \alpha \cdot d(\vec{z}_1, \vec{z}_2)$:

$$5.39 > 0.5 \cdot 8.45 = 4.225,$$

тому точку \vec{x}_7 назначаємо центром нового кластера:

$$\vec{z}_3 = \vec{x}_7, \quad k = 3.$$

На наступному кроці алгоритму обчислюємо відстані між всіма непозначеними точками \vec{x}_q і центрами кластерів \vec{z}_1 , \vec{z}_2 і \vec{z}_3 . В кожній групі із трьох відстаней вибираємо мінімальну, а серед мінімальних – максимальну, і визначаємо її індекс: $d_{\max} = d(3,1) = 2.83$. Обчислюємо відстані між центрами \vec{z}_1 , \vec{z}_2 і \vec{z}_3 і знаходимо їх середнє значення («типову» відстань):

$$d(\vec{z}_1, \vec{z}_2) = d(\vec{x}_1, \vec{x}_6) = 8.45; \quad d(\vec{z}_1, \vec{z}_3) = d(\vec{x}_1, \vec{x}_7) = \sqrt{6^2 + 3^2} = \sqrt{45} \approx 6.72;$$

$$d(\vec{z}_2, \vec{z}_3) = d(\vec{x}_6, \vec{x}_7) = \sqrt{(4-6)^2 + (8-3)^2} = \sqrt{29} \approx 5.39;$$

Знаходимо «типову» відстань:

$$d_{min} = \frac{8.45 + 6.71 + 5.39}{3} = 6.85.$$

Перевіряємо умову $d_{max} > \alpha \cdot d_{min}$. Оскільки умова не виконується ($2.83 < 3.425$), то процедуру пошуку нових центрів припиняємо і виконуємо процедуру віднесення непозначених точок до певних кластерів за критерієм мінімуму відстані до центрів цих кластерів. Непозначеними в вибірці є точки $\{\vec{x}_2, \vec{x}_3, \vec{x}_4, \vec{x}_5, \vec{x}_8, \vec{x}_9, \vec{x}_{10}\}$. Визначаємо, наприклад, відстані від точки \vec{x}_2 до центрів \vec{z}_1 , \vec{z}_2 і \vec{z}_3 (відповідно точки \vec{x}_4 , \vec{x}_6 і \vec{x}_7):

$$d(\vec{x}_2, \vec{z}_1) = d(\vec{x}_2, \vec{x}_1) = \sqrt{(3-0)^2 + (8-0)^2} = \sqrt{73} \approx 8.54,$$

$$d(\vec{x}_2, \vec{z}_2) = d(\vec{x}_2, \vec{x}_6) = \sqrt{(3-4)^2 + (8-8)^2} = \sqrt{1} = 1,$$

$$d(\vec{x}_2, \vec{z}_3) = d(\vec{x}_2, \vec{x}_7) = \sqrt{(3-6)^2 + (8-3)^2} = \sqrt{34} \approx 5.83.$$

Найближчим до точки \vec{x}_2 є центр кластера \vec{z}_2 , тому відносимо цю точку до другого кластера: $\vec{x}_2 \in \Omega_2$.

Аналогічно визначаємо належність до кластерів інших точок. Результат кластеризації зображено на рис. 5.7: $\Omega_1 = \{\vec{x}_1, \vec{x}_3, \vec{x}_4\}$, $\Omega_2 = \{\vec{x}_2, \vec{x}_6\}$, $\Omega_3 = \{\vec{x}_5, \vec{x}_7, \vec{x}_8, \vec{x}_9, \vec{x}_{10}\}$.

5.5 Алгоритм К-внутрішньогрупових середніх

Алгоритм К-внутрішньогрупових середніх реалізує узагальнений алгоритм кластеризації Фу К.С., описаний в підрозділі 5.2. На відміну від евристичних алгоритмів, описаних в підрозділах 5.3-5.4, його роботою керує формальний критерій, визначений як сума квадратів евклідових відстаней всіх точок, що входять в кластерну область, до центра кластера. Цей алгоритм використовує за вхідні дані вибірку векторів зображень $\{\vec{x}_1, \dots, \vec{x}_q, \dots, \vec{x}_Q\}$ і кількість класів K , причому $K < Q$. Його робота виконується таким чином, щоб мінімізувати значення вибраного критерію і складається з таких кроків.

На першому кроці точки $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_K$ призначаються центрами K кластерів $\vec{z}_1, \vec{z}_2, \dots, \vec{z}_K$.

На другому кроці задана вибірка зображень розподіляється по K кластерах згідно з правилом

$$\vec{x}_q \in \Omega_k(i), \text{ якщо } \|\vec{x}_q - \vec{z}_k(i)\| < \|\vec{x}_q - \vec{z}_j\|, \quad (5.17)$$

для всіх $j, k = 1, 2, \dots, K$, де $k \neq j$, Ω_k – k -ий кластер з центром $\vec{z}_k(i)$, i – номер ітерації. В випадку рівності в (5.17) рішення приймається довільно.

На третьому кроці визначають нові центри кластерів таким чином, щоб мінімізувати критерій якості кластеризації

$$\sigma_k^2 = \sum_{\vec{x}_q \in \Omega_k(i)} \|\vec{x}_q - \vec{z}_k(i+1)\|^2, \quad k = 1, 2, 3, \dots, K. \quad (5.18)$$

Для цього новий центр $\vec{z}_k(i+1)$ визначають як вибіркове середнє векторів кластера $\Omega_k(i)$:

$$\vec{z}_k = \frac{1}{N_k} \sum_{q=1}^{N_k} \vec{x}_q, \quad (5.19)$$

де N_k – кількість векторів в кластері $\Omega_k(i)$,

\vec{x}_q – вектори, що належать класу Ω_k .

Цей спосіб корекції центрів кластерів і визначає назву алгоритму « K - внутрішньогрупових середніх».

На четвертому кроці перевіряється умова закінчення процедури кластеризації, яка полягає в відсутності зміни координат центрів відносно їх попереднього значення:

$$\vec{z}_k(i+1) = \vec{z}_k(i), \quad \text{де } k = 1, 2, \dots, K. \quad (5.20)$$

В разі виконання умови (5.20) процедура кластеризації припиняється, інакше виконується чергова ітерація. На практиці ітерації продовжують до тих пір, поки не буде отримано мінімальне значення загального середньоквадратичного відхилення σ_T^2 для всіх кластерів.

Серед переваг цього алгоритму можна відзначити його простоту, ефективність, самоорганізацію і мінімізацію середньоквадратичного відхилення по всіх кластерах.

До недоліків алгоритма потрібно віднести:

- необхідність задання кількості кластерів K , яка апіорно відома дуже рідко;

- метод можна застосувати тільки до лінійно розділюваних класів, які мають вигляд гіперкуль, достатньо рознесених в просторі ознак;

- загальне доведення збіжності цього алгоритму невідоме.

Наведемо детальний опис алгоритму. Його входами і виходами є такі величини.

Вхідні дані: послідовність вибіркових векторів зображень $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_q, \dots, \vec{x}_Q\}$ і кількість K кластерів.

Вихідні результати: кластери $\Omega_1, \dots, \Omega_K$, задані індексною функцією $class(q) := k$; кількість векторів $n(k)$ в кожному кластері Ω_k ; центри \vec{z}_k

кожного кластера Ω_k ; дисперсії (середньоквадратичні відхилення) σ_k^2 для кожного k -го кластера; загальне середньоквадратичне відхилення σ_T^2 за всіма класами (загальна дисперсія), визначене формулою (5.13).

Алгоритм 5.3

Кластеризація за методом K -внутрішньогрупових середніх

```

Крок1: for  $k1:=1$  to  $K$  do begin          /* Цикл ініціалізації центрів
       $\vec{z}_k := \vec{x}_k$ ;  $n(k1):=0$ ; end;      кластерів */
Крок2: for  $q:=1$  to  $Q$  do begin          /* Цикл занесення векторів до кла-
       $d_{\min} := 999999.9$ ;                стерів з найближчим центром */
      for  $k1:=1$  to  $K$  do begin          /* Ініціалізація  $\min$  відстані */
       $d := \|\vec{x}_q - \vec{z}_{k1}\|$ ;          /* Цикл знаходження мінімальної
      if  $d < d_{\min}$  then begin          відстані до центрів */
       $d_{\min} := d$ ;  $k_{\min} := k1$       /*Позначення вектора  $\vec{x}_q$   $k$ -им
      end;                                класом */
      end;                                /* Кінець цикла за  $k1$  */
       $class(q) := k_{\min}$ ;
       $n(k_{\min}) := n(k_{\min}) + 1$ ;    /* Інкрементація лічильника векторів
      end;                                 $k_{\min}$ -го класу */
      /* Кінець цикла за  $q$  */
Крок3: for  $k1:=1$  to  $K$  do begin          /* Цикл обчислення  $K$  нових центрів
       $\vec{z}_k^* := (1/n(k1)) * \sum_{class(q)=k} \vec{x}_q$ ; /* Знаходження середнього  $k1$ -го
       $\sigma_{k1}^2 := (1/n(k1)) * \sum_{class(q)=k1} \|\vec{x}_q - \vec{z}_{k1}^*\|^2$  /*Обчислення дисперсії в  $k1$ -му
      end;                                кластері */
       $\sigma_T^2 := \sum_{k=1}^K \sigma_k^2$ ; /* Знаходження загального середньоквадратичного
      відхилення по всіх кластерах */
Крок4:  $Flag := False$ ; /* Установлення прапорця зупинення процедури */
      for  $k1:=1$  to  $K$  do /* Цикл фіксації зміни хоча б одного*/
      if  $\|\vec{z}_{k1}^* - \vec{z}_{k1}\| > 0.0005$  then /* центра за допомогою прапорця */
       $Flag := True$ ;
      if  $Flag = True$  then begin
      for  $k1:=1$  to  $K$  do /* За наявності змін оновити центри і
       $\vec{z}_{k1} \leftarrow \vec{z}_{k1}^*$ ;          повторити процедуру */
      go to Крок2 end
      else stop /* Інакше припинити кластеризацію */

```

Приклад 5.3. Для заданої на рис. 5.5 вибірки векторів $\{\bar{x}_1, \dots, \bar{x}_6\}$ (див. приклад 5.1) виконати кластеризацію методом K – внутрішньогрупових середніх.

Розв’язування. Задаємо $K := 2$ і вибираємо початкові значення центрів кластерів:

$$\bar{z}_1(1) = \bar{x}_1 = (2;1)^T, \quad \bar{z}_2(1) = \bar{x}_2 = (3;1)^T.$$

На другому кроці обчислюємо евклідові відстані від вибірових точок до центрів цих кластерів і відносимо ці точки до кластерів з найближчими центрами. Наприклад, відстані від точки $\bar{x}_5 = (2;3)^T$ до центрів $\bar{z}_1(1) = (2;1)^T$ і $\bar{z}_2(1) = (3;1)^T$ дорівнюють:

$$d(\bar{x}_5, \bar{z}_1(1)) = \sqrt{(2-2)^2 + (3-1)^2} = \sqrt{4} = 2,$$

$$d(\bar{x}_5, \bar{z}_2(1)) = \sqrt{(2-3)^2 + (3-1)^2} = \sqrt{5} \approx 2.25.$$

Оскільки $d(\bar{x}_5; \bar{z}_1(1)) < d(\bar{x}_5; \bar{z}_2(1))$, то точку \bar{x}_5 заносимо в кластер Ω_1 . Аналогічно виконаємо дії для всіх інших точок вибірки і отримаємо:

$$\Omega_1 = \{\bar{x}_1, \bar{x}_3, \bar{x}_5, \bar{x}_6\} = \{(2;1)^T, (2;4)^T, (2;3)^T, (1;3)^T\};$$

$$\Omega_2 = \{\bar{x}_2, \bar{x}_4\} = \{(3;1)^T, (3;2)^T\};$$

На третьому кроці визначаємо нові центри кластерів шляхом усереднення кластерів:

$$\bar{z}_1(2) = ((2+2+2+1)/4; (1+4+3+3)/4)^T = (1.75; 2.75)^T,$$

$$\bar{z}_2(2) = ((3+3)/2; (1+2)/2)^T = (3; 1.5)^T.$$

На четвертому кроці перевіряємо умову припинення процедури кластеризації:

$$\|\bar{z}_1(2) - \bar{z}_1(1)\| = \sqrt{(1.75-2)^2 + (2.75-1)^2} = 0.815 > 0.0005,$$

$$\|\bar{z}_2(2) - \bar{z}_2(1)\| = \sqrt{(3-3)^2 + (1.5-1)^2} = 0.5 > 0.0005.$$

Оскільки умова припинення процедури не виконується, то координати обох центрів значно змінилися; повторюємо крок 2 для перерозподілу вибірових точок по кластерах і отримуємо:

$$\Omega_1 = \{\bar{x}_1, \bar{x}_2, \bar{x}_4\} = \{(2;1)^T, (3;2)^T, (3;1)^T\},$$

$$\Omega_2 = \{\bar{x}_3, \bar{x}_5, \bar{x}_6\} = \{(1;3)^T, (2;4)^T, (2;3)^T\}.$$

Обчислюємо нові центри кластерів шляхом усереднення їх векторів:

$$\bar{z}_1(3) = ((2+3+3)/3; (1+2+1)/3)^T = (2.66; 1.33)^T,$$

$$\bar{z}_2(3) = ((1+2+2)/3; (3+4+3)/3)^T = (1.66; 3.33)^T.$$

Після перерозподілу точок по кластерах на основі значень нових центрів склад кластерів не змінюється, положення центрів теж не змінюється, тому процедуру кластеризації припиняємо. Результат – отримуємо такі самі кластери, як і в прикладі 5.1 (порогова кластеризація).

5.6 Алгоритм ICOMAD (ISODATA)

Алгоритм ICOMAD (ISODATA) – Ітеративний Самоорганізаційний Метод Аналізу Даних (ISODATA – Iterative Self-Organizing Data Analysis Techniques) – за своїм принципом аналогічний процедурі K –внутрішньогрупових середніх. Схожість цих алгоритмів полягає в тому, що центром кластерів в них слугують вибірккові середні, визначені ітеративно. Однак, на відміну від попереднього алгоритму, ICOMAD містить великий набір евристичних процедур, старанно відібраних на підставі експериментального досвіду.

Існує велика кількість модифікацій цього алгоритму, нижче наводиться його основна версія. В роботі алгоритму використовуються: випадкова навчальна вибірка векторів $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_q, \dots, \vec{x}_Q\}$; K_p – необхідна кількість кластерів; n_{\min} – мінімально допустима кількість векторів в кластері; $\sigma_n(k)$ – стандартне (середньоквадратичне) відхилення n -ої компоненти за всіма векторами q -го кластера, $t = 1, 2, \dots, N$, де N – розмірність простору ознак; $\sigma_s(k)$ – параметр, що характеризує компактність k -го кластера; d_L – поріг злиття для відстаней між центрами кластерів, який визначає умову злиття двох кластерів з досить близькими центрами в один; L_{\max} – максимальна кількість пар центрів кластерів, які можна об'єднати; I_p – допустима кількість ітерацій. Вказані вище параметри керують процесом кластеризації. Крім того, використовуються такі величини: K_C – поточна кількість кластерів; $\Omega_1, \Omega_2, \dots, \Omega_{K_C}$ – поточні кластери з відповідними центрами $\vec{z}_1, \vec{z}_2, \dots, \vec{z}_{K_C}$.

Якщо стандартне відхилення $\sigma_t(k)$ більше за $\sigma_s(k)$, то центр \vec{z}_k розділяється на два центри, які відрізняються від попереднього тільки n -ою компонентою. Значення цієї компоненти в нових центрах дорівнюють $\vec{z}_k + \sigma_t(k)$ і $\vec{z}_k - \sigma_t(k)$, відповідно. Параметр d_L керує злиттям двох центрів в один в тому випадку, коли центри деякої пари кластерів розташовані ближче, ніж d_L . Протягом однієї ітерації допускається об'єднання тільки L_{\max} пар.

Алгоритм ISODATA складається з таких кроків.

Крок 1. Призначаються за центри кластерів перші K_C вибірккових точок:

$$\vec{z}_k := \vec{x}_k, \quad k = 1, 2, \dots, K_C.$$

Крок 2. Задані Q вибірккових точок розподіляються по кластерах за правилом найближчого центра.

Крок 3. Ліквідуються кластери, що містять менше ніж n_{\min} точок, тобто, якщо для деякого k -го кластера виконується умова $N_k < n_{\min}$, то кластер Ω_k виключається з розгляду, а значення K_c зменшується на 1.

Крок 4. Обчислюються нові центри кластерів шляхом усереднення всіх векторів, що входять в кожний k -ий кластер:

$$\bar{z}_k = \frac{1}{N_k} \sum_{class(q)=k} \bar{x}_q,$$

де N_k – кількість векторів в k -ому кластері. Обчислюються також нові значення дисперсії (квадрат стандартного відхилення)

$$\sigma_n^2 = \frac{1}{N_k} \sum_{t=1}^N \sum_{class(q)=k} (x_{tq} - z_{tk})^2.$$

всередині кожного k -го кластера і підсумовуються ці дисперсії по всіх кластерах для отримання загальної середньоквадратичної помилки σ_T^2 .

Крок 5. В кожному k -му кластері обчислюється середньоквадратична похибка (дисперсія) $\sigma_i^2(k)$ для кожної n -ої компоненти x_n і знаходиться серед них максимальна $\sigma_{t_{\max}}^2(k)$, де індекс t_{\max} означає номер максимальної компоненти.

Крок 6. Якщо кількість кластерів недостатня ($K_c < K_p/2$) і поточна ітерація не є останньою, а для деякого з кластерів виконується умова $\sigma_{az}(k) > \sigma_s(k)$, то кластер розбивається на два кластери шляхом розподілу одного кластера на два центри, що відрізняються тільки в компоненті з максимальною дисперсією. Два нових центри мають n_{\max} -ті компоненти, що дорівнюють відповідно $z_{t_{\max}} + \sigma_{\max}(k)$ і $z_{t_{\max}} - \sigma_{\max}(k)$, а всі інші компоненти залишаються незмінними.

Крок 7. Якщо ітерація є парною і $K_c > 2K_p$, то обчислюються всі відстані $D_{ij} = d(\bar{z}_i, \bar{z}_j)$ між центрами кластерів \bar{z}_i і \bar{z}_j ; $i, j=1, \dots, K_c$. Якщо деяка відстань $D_{ij} < d_L$, то i -тий і j -тий кластери зливаються в один кластер (за додаткової умови, що кількість об'єднань не перевищує L_{\max}). У цьому разі зменшують поточну кількість класів $K_c = K_c - 1$.

Крок 8. Змінюється номер ітерації $i := i + 1$. Якщо $i > I_p$, то процедура кластеризації припиняється, інакше – виконати нову ітерацію, починаючи з кроку 2.

Нижче наводиться детальний опис алгоритму ISODATA. Він використовує такі величини:

Вхідні дані: випадковим чином впорядкована вибірка ознакових векторів $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_q, \dots, \vec{x}_Q\}$; початкове значення K_C кількості кластерів; наближено вибране необхідна кількість K_p кластерів; максимальне значення середньоквадратичного відхилення (поріг) σ_s для розділення класів; порогове значення d_L мінімальної відстані для злиття кластерів; мінімальна кількість n_{\min} ознакових векторів в кластері, нижче якої кластер ліквідується; максимальна кількість пар кластерів L_{\max} , які можуть бути об'єднані на цій ітерації, максимальна кількість ітерацій I_p .

Вихідні результати: поточна кількість кластерів K_C ; кластери Ω_k , задані функцією визначення індексів $class(q) = k$ (якщо $\vec{x}_q \in \Omega_k$); кількість векторів $n(k)$ в k -ому кластері. В циклах алгоритму параметр q позначений як $q1$, щоб відрізнити його від верхнього значення Q в алгоритмічних мовах.

Алгоритм 5.4: ISODATA

Крок 1: Ініціалізація кількості ітерацій і параметрів.

```

i := 0;                                /*Ініціалізація параметрів*/
input (Kp, KC, Ip, dL, σs);
for q1 := 1 to Q do begin
    cc(q1) := 0;                        /*Ініціалізація значень центрів*/
    class(q1) := 0;                    /*Ініціалізація значень міток класів*/
end;
for k := 1 to 2Kp + 1 do              /*Ініціалізація кількості векторів в
    n(k) := 0;                          кластерах*/

```

Крок 2: Вибір перших K_C векторів за центри кластерів

```

for k := 1 to KC do begin
     $\vec{z}_k := \vec{x}_k$ ;                /*Призначення KC центрів*/
    class(q1) := k;                    /*Фіксація призначень індексною функцією*/
end;

```

Крок 3: Віднесення ознакових векторів до кластерів за мінімальним значенням відстані

```

change_cluster := False;              /*Початкові зміни кластерів відсутні*/
for q1 := 1 to Q do begin            /*Для кожного ознакового вектора  $\vec{x}_q$  */
    dmin := 999999.9;                /* знаходиться мінімальна відстань */
    for k := 1 to KC do begin        /*до центрів кластерів*/

```

```

     $d := \|\vec{x}_q - \vec{z}_q\|;$ 
    if  $d < d_{\min}$  then begin
         $d_{\min} := d;$    $k_{\min} := k;$ 
    end;
end ;
 $cc(q1) := k_{\min};$            /*Позначити центри кластерів*/
 $n(k_{\min}) := n(k_{\min}) + 1;$  /*Збільшити значення лічильника*/
if  $cc(q1) \neq class(q1)$  then /*Фіксація факту зміни центра*/
     $change\_cluster := True;$ 
    if  $change\_cluster = False$  then stop /*Зупинення за відсутності змін */
end ;                               /*Кінець циклу за Q*/
for  $q1 := 1$  to  $Q$  do               /* Оновлення індексних міток*/
     $class(q1) := cc(q1);$ 

```

Крок 4: Ліквідація кластерів з числом векторів, меншим ніж n_{\min} , перенумерація кластерів, оновлення числа кластерів

```

for  $k := 1$  to  $K_c$  do               /*Цикл перегляду кластерів*/
    if  $n(k) < n_{\min}$  then begin     /* Якщо векторів в кластері мало*/
        for  $j := k + 1$  to  $K_c$  do begin /*то ліквідувати його і узгодити*/
             $n(j - 1) := n(j); \vec{z}_{j-1} := \vec{z}_j$  end; /* індекси */
        for  $q1 := 1$  to  $Q$  do
            if  $class(q1) > k$  then
                 $class(q1) := class(q1) - 1;$  /* Змінити мітки кластерів*
             $K_c := K_c - 1;$  /* Зменшити поточне число кластерів*/
        end;                               /* Кінець if */

```

Крок 5: Обчислення нових центрів кластерів

```

for  $k := 1$  to  $K_c$  do
     $\vec{z}_k^* := (1/n(k)) \sum_{class(q)=k} \vec{x}_q;$  /*Формування нового центра усередненням */

```

Крок 6: Обчислення нового значення дисперсії для кожного кластера

$$D_k := \left(\frac{1}{n(k)}\right) * \sum_{q: class(q)=k} \|\vec{x}_q - \vec{z}_k\|^2;$$

Крок 7: Обчислення загальної дисперсії за всіма кластерами і максимальної компоненти дисперсії для кожного кластера

```

 $D_T := \left(\frac{1}{Q}\right) \sum_{k=1}^{K_c} n(k) D_k ;$  /*Сума дисперсій всіх кластерів*/
for  $k := 1$  to  $K_c$  do begin /*Для всіх кластерів обчислити*/
     $S_{\max} := 0.0;$  /*дисперсії ознакових компонент*/
     $sum := 0.0;$ 
    for  $t := 1$  to  $N$  do begin /*Для кожної компоненти знайти */
        for  $q1 := 1$  to  $Q$  do /*суму квадратів відхилень*/
            if  $class(q1) := k$  then /*за всіма векторами кластера*/
                 $sum := sum + (x_{tq} - z_{tq})^2;$ 
                 $sum := sum / n(k);$ 
            if  $sum \geq s_{\max}$  then  $s_{\max} := sum;$  /*Знайти максимальне середньо-*/
                /*квадратичне відхилення*/
                 $\sigma_{\max} := \sqrt{sum}; t_{\max} := t$  /*Обчислити максимальне  $\sigma_k$  */
                /* і номер його компоненти  $t_{\max}$  */
        end /* кінець циклу за  $t$  */
    end; /* кінець циклу за  $k$  */

```

Крок 8: Вибір умови розбиття або злиття кластерів

```

if  $i \geq I_p$  then begin /*Якщо ітерація остання*/
     $d_L := 0;$  /*то можливе злиття, але не */
    goto Крок_10; /*розбиття*/
end;
if  $K_c \leq K_p / 2$  then /*Якщо кластерів мало, то*/
    goto Крок_9; /*спробувати розділити кластери*/
if  $(i \bmod 2 = 0) \text{ and } (K_c \geq 2K_p)$  /*Якщо  $i$  – парне і кластерів багато */
then goto Крок_10; /* то спробувати злити*/

```

Крок 9: Вибір кластера і його розбиття на два кластери за наявності причин для цього

```

 $split := False;$ 
for  $k := 1$  to  $K_c$  do /*Якщо максимальна компонента*/
    if  $\sigma_{\max}(k) > \sigma_s$  then /* $\sigma_{\max}(k)$  велика і якщо є достатня*/
        if  $(n(k) > 2 \cdot n_{\min} + 1)$  and  $D_k > D_T$  /*кількість векторів для двох*/
            or  $K_c < K_p / 2$  then begin /*кластерів або дисперсія велика, */
                /* або досить мало кластерів */
                 $split := True; K_c := K_c + 1;$  /* то розбити на два кластери */
                 $z_{t_{\max}}^k := z_{t_{\max}}^k - 0.5 \cdot \sigma_{\max}(k);$  /*за певною компонентою*/
                 $z_{t_{\max}}^{K_p} := z_{t_{\max}}^{K_p} + 0.5 \cdot \sigma_{\max}(k);$ 
            end;
end;

```

```

if split = True then                                     /*Якщо розбиття відбулося, то*/
goto Крок_3 ;                                           /*перейти до перерозподілу точок по кластерах */
Крок 10: Обчислення відстаней між всіма парами центрів кластерів і
злиття достатньо близьких
for k := 1 to  $K_c - 1$  do                               /*Для всіх пар кластерів (k, j)/
  for j := k + 1 to  $K_c$  do begin                   /*визначити відстані між центрами*/
     $D_{jk} := \|\vec{z}_j - \vec{z}_k\|$ ;
    if  $D_{jk} < d_L$  then  $L(j,k) := True$  /*Порівняти з  $d_L$  і встановити*/
      else  $L(j,k) := False$                        /*умову злиття*/
  end;
Крок 11: Знайти найменшу з відстаней  $D_{jk}$ , меншу за  $d_L$  і злити два
кластери
count := 0                                             /*Обнулення лічильника злиттів*/
repeat
   $d_{min} := 999999.9$ ; empty := True;           /*Початкові значення параметрів*/
  for k := 1 to  $K_c - 1$  do
    for j := k + 1 to  $K_c$  do
      if  $L(j,k) = True$  then
        if  $D_{jk} < d_{min}$  then begin               /*Якщо відстань <  $d_L$ , то знайти*/
           $d_{min} := D_{jk}$ ;                          /*мінімальну з відстаней <  $d_L$ */
          empty := False; /*Якщо вказане злиття і черга не пуста,*/
           $j^* := j; k^* := k$  /*то зафіксувати індекси min відстані*/
        end ;
      if empty := false then                       /*Процедура злиття пар кластерів*/
        call Lump( $j^*, k^*$ );                          /*з мінімальною відстанню*/
         $L(j^*, k^*) := False$ ;                          /*Ліквідувати індикатор злиття*/
        count := count + 1;                          /*Інкрементація лічильника злиттів*/
      until empty = True or                          /* Кінець циклу repeat:*/
         $count \geq L_{max}$                              /*Якщо черга індикаторів злиття пуста*/
          /* або вже виконано  $L_{max}$  злиттів, */
          /* повторення злиттів зупиняється */
Крок 12: Вибір переходу до наступної ітерації або зупинення
if  $i < I_p$  then begin  $i := i + 1$ ; goto Крок_3 /*Збільшуємо номер ітерації*/
  end;
else stop                                               /*і зупиняємося, якщо ітерацій достатньо*/

```

Опис процедури злиття $Lump(j^*, k^*)$.

```
 $k^* := \min(j^*, k^*); j^* := \max(j^*, k^*);$  /* Використати найменший індекс  
для злитого кластера */  
 $n(k^*) := n(k^*) + n(j^*);$  /* Кількість векторів в новому кластері */  
 $\bar{z}_{k^*} := (\frac{1}{n(k^*)}) \cdot [n(j^*) \cdot \bar{z}_{j^*} + n(k^*) \cdot \bar{z}_{k^*}];$  /* Центр нового кластера */  
for  $q := 1$  to  $Q$  do /* Обновити функцію міток індексів, */  
  if  $class(q) = j^*$  then /* перепризначивши індекси нового класу */  
     $class(q) = k^*$  ;  
   $K_c := K_c - 1; n(j^*) := 0; \bar{z}_{j^*} := 0;$   
for  $k := j^*$  to  $K_c$  do begin  
   $n(k) := n(k + 1); \bar{z}_k := \bar{z}_{k+1};$   
  for  $q := 1$  to  $K_c$  do  
    if  $class(q) = k$  then  
       $class(q) := k - 1;$   
end.
```

До переваг алгоритму ICOMAD потрібно віднести:

- здатність до самоорганізації;
- високу гнучкість в ліквідації достаньох малих кластерів;
- здатність до поділу кластерів з досить несхожими векторами ознак;
- здатність до злиття кластерів з подібними векторами ознак.

Але він має і недоліки, наприклад:

- його можна застосовувати тільки до лінійно розділюваних класів, що мають вигляд гіперкуль;
- визначення параметрів $K_p, n_{\min}, d_L, \sigma_s, L_{\max}$ і I є досить складним, оскільки вони не є апріорно відомими;
- результати його роботи сильно залежать від заданих користувачем параметрів;
- менш ефективний з погляду обчислювальних затрат порівняно з іншими лінійними алгоритмами для випадку великої навчальної вибірки і великої кількості кластерів.

На практиці алгоритм потрібно повторити багато раз для різних значень K та інших параметрів, внаслідок чого вибирається варіант кластеризації з мінімальною загальною середньоквадратичною похибкою.

Хоча алгоритм ISODATA і не підходить для обчислень вручну, принцип його роботи розглянемо на досить простому прикладі.

Приклад 5.4. Для вибірки образів, поданої на рис. 5.8, виконати процедуру кластеризації згідно з алгоритмом ISODATA.

В цьому прикладі вибірка складається з $Q=8$ векторів зображень: $\{\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4, \vec{x}_5, \vec{x}_6, \vec{x}_7, \vec{x}_8\} = \{(0;0)^T, (1;1)^T, (2;2)^T, (4;3)^T, (5;3)^T, (4;4)^T, (5;4)^T, (6;5)^T\}$, розмірність векторів $N=2$.

Розв'язування: застосуємо до вибірки послідовність дій, передбачених алгоритмом ISODATA.

Крок 1. Задаємо початкові значення параметрів кластеризації: $K_p=2$ – необхідна кількість кластерів; $K_c=1$ – поточна кількість кластерів; $n_{\min}=2$ – мінімально допустима кількість векторів в кластері,

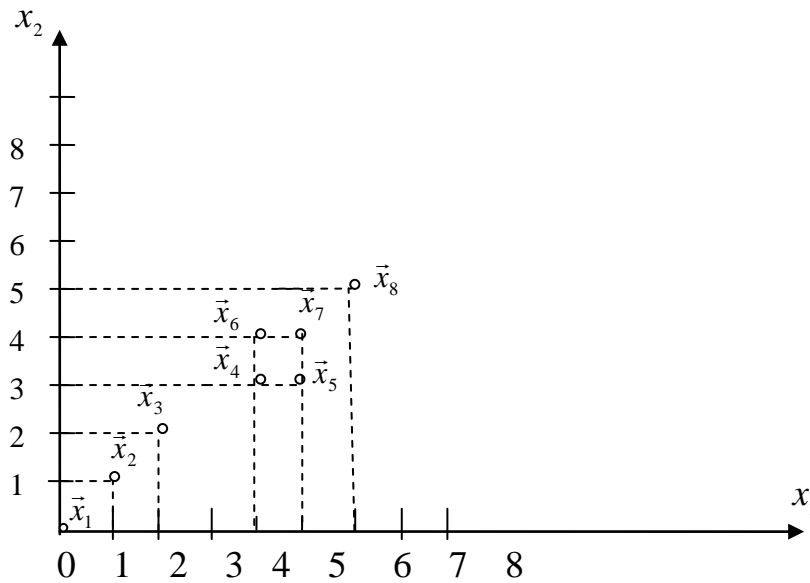


Рисунок 5.8 – Вибірка образів, використана для ілюстрації роботи алгоритму ISOMAD

$\sigma_s=1$ – пороговий параметр для стандартного відхилення (використовується як умова розбиття класів);

$d_L=4$ – пороговий параметр компактності (використовується як умова злиття класів);

$L_{\max}=1$ – кількість пар кластерів, дозволених для злиття протягом однієї ітерації;

$I_p=4$ – допустима кількість ітерацій;

$i=1$ – задаємо номер першої ітерації.

Ці параметри вибрані довільним чином, оскільки апріорна інформація про аналізовані дані є відсутньою. Вони уточнюються на кожній наступній ітерації.

Крок 2. Призначаємо перші K_C центрів кластерів:

$$\vec{z}_1 = \vec{x}_1.$$

Крок 3. За критерієм мінімальної відстані розподіляємо вибіркові вектори по K_C кластерах. Оскільки задано тільки один центр, то всі вектори вибірки входять в один клас:

$$\Omega_1 = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_8\}, \quad n(1) = 8.$$

Крок 4. Вилучаємо ті кластери, для яких виконується умова $n(k) < n_{\min}$. На цій ітерації $n(1) = 8$, $n_{\min} = 2$, і оскільки $n(1) > n_{\min}$, то ні один клас не ліквідується.

Крок 5. Обчислюємо нові координати центра кластера:

$$\vec{z}_1 = \frac{1}{n(k)} \sum_{\vec{x}_q \in \Omega_k} \vec{x}_q = ((0+1+2+4+5+4+5+6)/8; (0+1+2+3+3+4+4+5)/8)^T = (3.38, 2.75)^T$$

Крок 6. Обчислюємо нове значення середньоквадратичної помилки в кожному класі:

$$D_k = \frac{1}{n(k)} \sum_{\vec{x}_q \in \Omega_k} \|\vec{x}_q - \vec{z}_k\|^2; \quad D_1 = \frac{1}{8} \sum_{q=2}^8 \left(\sqrt{\sum_{t=1}^N (x_{tq} - z_{t1})^2} \right)^2$$

$$D_k = \frac{1}{8} \left[\begin{aligned} &((1-3.38)^2 + (1-2.75)^2 + (2-3.38)^2 + (2-2.75)^2 + (4-3.38)^2 + \\ &+ (3-2.75)^2 + (5-3.38)^2 + (3-2.75)^2 + (4-3.38)^2 + (4-2.75)^2 + \\ &+ (5-3.38)^2 + (4-2.75)^2 + (6-3.38)^2 + (5-2.75)^2) = 5.11 \end{aligned} \right]$$

Крок 7. Визначаємо дисперсію за всіма кластерами і компоненту з максимальною серед всіх кластерів дисперсією:

$$D_T = \frac{1}{Q} \sum_{k=1}^Q n(k) D_k; \quad D_T = \frac{1}{8} \cdot 8 \cdot D_1 = 5.11,$$

і координати вектора стандартного (середньоквадратичного) відхилення для кластера Ω_1 за формулою

$$\sigma_{kt} = \sqrt{\frac{1}{n(k)} \sum_{q=1}^Q (x_{tq} - z_{tk})^2} :$$

$$\sigma_{11} = \sqrt{\frac{1}{8} ((3.38-3.38)^2 + (1-3.38)^2 + (2-3.38)^2 + (4-3.38)^2 + (5-3.38)^2 + (4-3.38)^2 + (5-3.38)^2 + (6-3.38)^2)} = 1.99$$

$$\sigma_{12} = \sqrt{\frac{1}{8} ((2.75-2.75)^2 + (1-2.75)^2 + (2-2.75)^2 + (3-2.75)^2 + (3-2.75)^2 + (4-2.75)^2 + (4-2.75)^2 + (5-2.75)^2)} = 1.56$$

тобто $\vec{\sigma}_1 = (1.99; 1.56)^T$.

Оскільки на цьому етапі існує всього один кластер, то максимальна компонента стандартного відхилення дорівнює:

$$\sigma_{\max} = \sigma_{\max}(1) = 1.99.$$

Крок 8. Оскільки $K_C = \frac{K_p}{2}$, то переходимо на крок 9.

Крок 9. Оскільки $\sigma_{\max}(1) > \sigma_s$ ($1.99 > 1$), то кластер з центром \vec{z}_1 розділюється на два нових кластери:

$$\vec{z}_1^- = (3.38 - 0.5 \cdot \sigma_{\max}(1), 2.25)^T = \begin{pmatrix} 2.38 \\ 2.75 \end{pmatrix},$$

$$\vec{z}_1^+ = (3.38 + 0.5 \cdot \sigma_{\max}(1), 2.25)^T = \begin{pmatrix} 4.38 \\ 2.75 \end{pmatrix},$$

$$K_c = K_c + 1 = 1 + 1 = 2; \quad i = i + 1 = 1 + 1 = 2.$$

Переходимо на крок 3.

Крок 3. За критерієм мінімальної відстані розподіляємо вибіркві вектори по 2 кластерах:

$$\Omega_1 = \{\bar{x}_4, \bar{x}_5, \bar{x}_6, \bar{x}_7, \bar{x}_8\}, \quad \Omega_2 = \{\bar{x}_1, \bar{x}_2, \bar{x}_3\};$$

$$n(1) = 5, \quad n(2) = 3.$$

Крок 4. Оскільки обидві величини – $n(1)$ і $n(2)$ – більші за n_{\min} , то ні один із кластерів не ліквідується.

Крок 5. Знаходимо нові центри кластерів:

$$\vec{z}_1 = \frac{1}{n(1)} \sum_{\bar{x}_q \in \Omega_1} \bar{x}_q = ((4+5+4+5+6)/5; (2+3+4+4+5)/5)^T = (4.80; 3.80)^T,$$

$$\vec{z}_2 = \frac{1}{n(2)} \sum_{\bar{x}_q \in \Omega_2} \bar{x}_q = ((0+1+2)/3; (0+1+2)/3)^T = (1.00; 1.00)^T.$$

Крок 6. Визначаємо нові значення дисперсії в кожному класі:

$$D_1 = \frac{1}{n(1)} \sum_{\bar{x}_q \in \Omega_1} \|\bar{x}_q - \vec{z}_1\|^2 = \frac{1}{5} \sum_{q=4}^8 \|\bar{x}_q - \vec{z}_1\|^2 = 1.12,$$

$$D_2 = \frac{1}{n(2)} \sum_{\bar{x}_q \in \Omega_2} \|\bar{x}_q - \vec{z}_2\|^2 = \frac{1}{3} \sum_{q=1}^3 \|\bar{x}_q - \vec{z}_2\|^2 = 1.33.$$

Крок 7. Обчислюємо D_T , і σ_1, σ_2 :

$$D_T = \frac{1}{Q} \sum_{j=1}^Q N_j D_j = \frac{1}{8} (5 \cdot 1.12 + 3 \cdot 1.33) = 1.2.$$

Для кластерів Ω_1, Ω_2 знаходимо стандартні відхилення:

$$\vec{\sigma}_1 = \begin{pmatrix} 0.75 \\ 0.75 \end{pmatrix}, \quad \vec{\sigma}_2 = \begin{pmatrix} 0.82 \\ 0.82 \end{pmatrix},$$

$$\sigma_{\max}(1) = 0.75, \quad \sigma_{\max}(2) = 0.82.$$

Крок 8. Оскільки ця ітерація є парною і $K_c > 2 \cdot K_p$, то переходимо на крок 10.

Крок 10. Обчислюємо відстані між парами центрів кластерів :

$$D_{jk} = \|\vec{z}_j - \vec{z}_k\|;$$

$$D_{12} = \|\vec{z}_1 - \vec{z}_2\| = \sqrt{(4.80 - 1.00)^2 + (3.80 - 1.00)^2} = 4.72.$$

Крок 11. Перевіряється умова $D_{jk} < d_L$. Оскільки в цьому випадку умова не виконується ($4.72 > 4$), то об'єднання кластерів неможливе.

Крок 12. Оскільки цей цикл ітерації не останній, $i < I_p$, то необхідно прийняти рішення: вносити чи не вносити зміни в параметри процесу кластеризації.

Оскільки в такому (простому) випадку:

- 1) кількість знайдених кластерів відповідає заданій;
- 2) відстань між ними більша середнього розкиду, який характеризується середньоквадратичним відхиленням;
- 3) оскільки кожний кластер містить суттєву частину загальної кількості вибіркового образів, то можна зробити висновок про те, що локалізація центрів кластерів правильно відтворює специфіку аналізованих даних.

Тому параметри залишаємо незмінними і переходимо до кроку 3 нової ітерації:

$$i = i + 1 = 2 + 1 = 3.$$

Виконання кроків 3–7 дає такі самі результати, що і в попередньому циклі ітерації.

Крок 8. Оскільки $K_c = \frac{K_p}{2}$, то переходимо на крок 9.

Крок 9. Оскільки $\sigma_{\max}(1) = 0.75$ і $\sigma_{\max}(2) = 0.82$. Умови розподілу кластерів не виконуються, тому переходимо до кроку 10.

Крок 10. Обчислюємо відстань між парами класів:

$$D_{12} = \|\vec{z}_1 - \vec{z}_2\| = 4.72$$

що ідентично результату попереднього циклу ітерації.

Крок 11. Отриманий результат ідентичний результату попереднього циклу ітерації.

Крок 12. Оскільки $i < I_p$ ($3 < 4$), то виконуємо ще одну ітерацію:

$$i = i + 1 = 3 + 1 = 4$$

Кроки 3–7 дають такі самі результати, що й в попередньому циклі ітерації.

Крок 8. Оскільки цей цикл ітерації – останній, задаємо $d_L = 0$ і переходимо до кроку 10.

Крок 10. Як і раніше:

$$D_{12} = \|\vec{z}_1 - \vec{z}_2\| = 4.72$$

Крок 11. Результат ідентичний результату попереднього циклу ітерації, об'єднання кластерів неможливе.

Крок 12. Оскільки $i = I_p$, то процедуру кластеризації припиняємо.

5.7 Алгоритм динамічних ядер

Характерними недоліками алгоритмів ІСОМАД, К–внутрішньогрупових середніх та інших їм подібних є неспроможність

функціонування у випадку класів, що розділяються нелінійними границями. Алгоритм динамічних ядер, описаний нижче, позбавлений вказаного недоліку, а також містить в собі критерій якості функціонування. Він використовує декілька центрів, названих ядрами, для кожного кластера. Рисунок 5.9 пояснює цей підхід.

На рис. 5.9 E_1 і E_2 є ядрами кластерів Ω_1 і Ω_2 . Нехай число кластерів є заданим (наприклад, за допомогою інших алгоритмів). Алгоритм розпочинає роботу з такими даними:

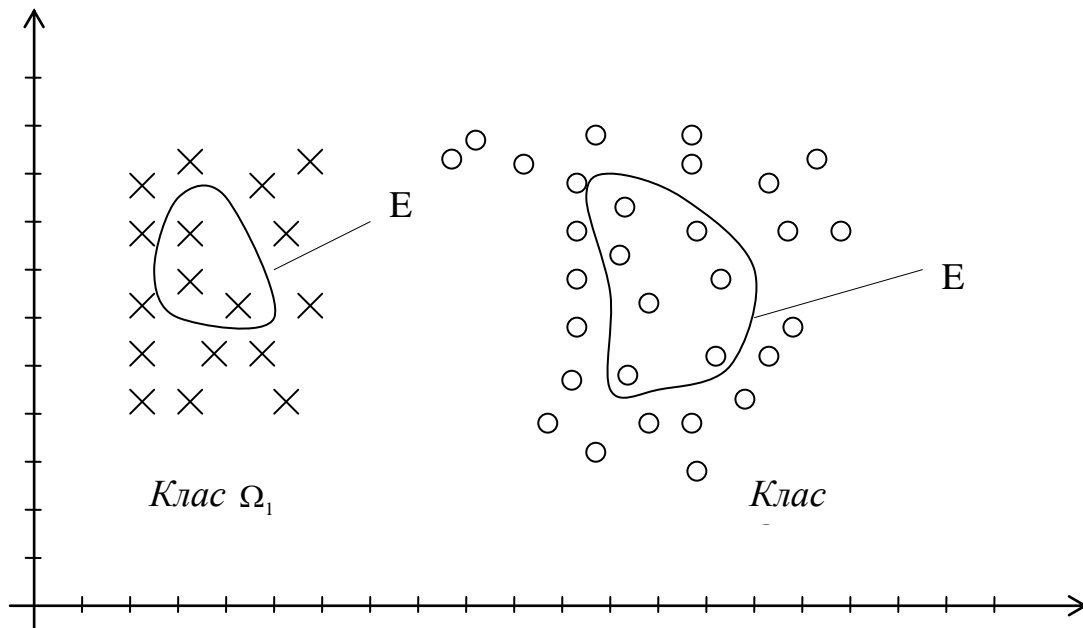


Рисунок 5.9 – Динамічні ядра кластерів

- 1) множина наборів з обмеженою кількістю векторів $\Omega = \{\Omega_1, \dots, \Omega_{K_p}\}$, що є представниками різних кластерів;
- 2) початкові набори незначного обсягу $\mathbf{E}(0) = \{E_1(0), \dots, E_{K_p}(0)\}$, що є ядрами і містять спочатку по одному елементу;
- 3) кількість векторів Q_k , необхідних для ядер E_k кластерів Ω_k , $k = 1, 2, \dots, K_p$.

Проблема кластеризації в цьому випадку полягає в пошуку такої множини ядер \mathbf{E} і віднесенню до них вибіркового векторів зображень таким чином, щоб отримати множину кластерів Ω , яка мінімізує критерій:

$$I(\Omega, \mathbf{E}) = \sum_{\vec{x}_q \in \Omega_k} \sum_{\vec{z} \in E_k} \|\vec{x}_q - \vec{z}\|, \quad (5.21)$$

де $\vec{x}_q \in \Omega_k$ і $\vec{z} \in E_k$, $k = 1, 2, \dots, K_p$.

Введемо величину

$$D(\vec{x}_q, E_k) = \sum_{\vec{z} \in E_k} \|\vec{x}_q - \vec{z}\| = \|\vec{x}_q - E_k\|. \quad (5.22)$$

Процедура мінімізації критерію (5.21) полягає в виборі N_k ознакових векторів \vec{x}_q з Ω_k для внесення їх в E_k з метою мінімізації функції

$$D_k = D(\vec{x}_q, E_k) / \sum_{j=1, K} D(\vec{x}_q, E_k), \quad (5.23)$$

причому має виконуватися умова $\Omega_r \cap \Omega_k = \emptyset$.

В описаному нижче алгоритмі динамічних ядер через I позначено задане число ітерацій, K_p відповідає бажаному числу кластерів, множина $\{\vec{x}_1, \dots, \vec{x}_Q\}$ подає вибіркові вектори зображень, $e(q)=k$ є індексною функцією належності векторів \vec{x}_q до ядра E_k , а $class(q)=k$ є індексною функцією (масивом індексів) належності \vec{x}_q до класу k .

Вхідні дані: вибірка векторів зображень $\{\vec{x}_1, \dots, \vec{x}_Q\}$; кількість кластерів K_p ; кількість ітерацій I_p ; Q_k – кількість точок (векторів) в ядрі E_k k -го кластера Ω_k .

Вихідні дані: ядра E_k , задані індексною функцією $e(q)=k$ для $\vec{x}_q \in E_k$; кластери Ω_k , $k = 1, 2, \dots, K_p$, задані індексною функцією $class(q)=k$ якщо $\vec{x}_q \in \Omega_k$.

Алгоритм 5.5

Метод динамічних ядер

Крок 1: Вибір початкових центрів, віднесення векторів в кластери за критерієм мінімуму відстані

```

input  $K_p, I_p$ ;
 $i := 0$ ;                                /*Початкова кількість ітерацій*/
for  $k := 1$  to  $K_p$  do
     $\vec{z}_k \leftarrow \vec{x}_k$ ;              /*Призначити перші  $K_p$  центрів*/
for  $q = 1$  to  $Q$  do
    if  $d(\vec{z}_k; \vec{x}_q) = d_{\min}$  then    /*Якщо відстань від  $\vec{x}_q$  до  $\vec{z}_k$  мінімальна,*/
         $class(q) := k$ ;                /*то віднести до  $\Omega_k$ */

```

Крок 2: Вибір багатоцентрових ядер для кожного кластера

```

for  $k := 1$  to  $K_p$  do                    /*Вибір  $Q_k$  векторів для  $E_k$ */
for  $j := 1$  to  $Q_k$  do begin              /*Випадковий вибір  $\vec{x}_j$  для  $E_k$ */
    select new  $\vec{x}_j$  from  $\Omega_k$ ;
     $e(j) := k$ ; end;

```

Крок 3: Віднесення однакових векторів до кластерів за критерієм мінімуму відстані;

```

for  $k := 1$  to  $K_p$  do  $D_k := 0$            /*Ініціалізація середньоквадратичної
                                           помилки для всіх кластерів */
  for  $q := 1$  to  $Q$  do begin           /*Для всіх  $\vec{x}_q$  зробити призначення до
                                           кластерів за мінімумом відстані */
     $d_{\min} \leftarrow \|\vec{x}_q - E_1\|$ ;
    for  $k := 2$  to  $K_p$  do           /* Для кожного  $k$ -го класу знайти ядро */
      if  $\|\vec{x}_q - E_k\| < d_{\min}$  then begin /* з мінімальною відстанню до  $\vec{x}_q$  */
         $k_{\min} \leftarrow k; d_{\min} = \|\vec{x}_q - E_k\|$ ; end; /*Фіксація індексів цієї відстані*/
       $class(q) := k_{\min}$ ;           /*Віднести  $\vec{x}_q$  до класу  $k_{\min}$ */
       $D_{k_{\min}} := D_{k_{\min}} + d_{\min}$ ;
  end;

```

Крок 4: Вибір нових ядер E_k в кожному кластері $k := 1, 2, \dots, K_p$ до тих пір, поки не мінімізується D_k .

```

repeat           /*Повторювати до тих пір,*/
                 /*поки  $change\_flag = FALSE$ ;*/
   $change\_flag := FALSE$ ; /*Відсутність змін у кластерах*/
  reselect new  $\xi_k$  from  $\Omega_k$ ; /*Вибір нових ядер випадковим
                                     чином з  $\Omega_k$ */
  compute  $D_k$ ; /*Обчислити нові середньоквадратичні */
                 /* відхилення в кластерах*/
  if  $D_k < D_k$  then begin /*Якщо відхилення зменшилось, то */
     $D_k := D_k$ ;  $E_k := \xi_k$ ; /* поновити відхилення і ядра*/
     $change\_flag \leftarrow TRUE$ ; /*Ядро змінилось, тому запобігти зупинці */
  until  $change\_flag = False$ ; /* Кінець циклу repeat */

```

Крок 5: Змінити номер ітерації і повторити цикл або зупинитися ;

```

 $i \leftarrow i + 1$ ;
if  $i > I_p$  then stop /*Якщо кількість ітерацій перевищує задану, */
else goto Крок_3; /* то зупинитись*/

```

Перевагою цього методу є його спроможність кластеризації даних з нелінійною розподільною границею (кластери не обов'язково мають бути у вигляді гіперкуль, а можуть мати продовгуватий або викривлений вигляд).

До недоліків алгоритму відносяться:

- необхідність визначення і програмування стратегії відбору ядер – для великих вибірок однакових векторів така стратегія стає досить складною в обчислювальному плані;

- результати кластеризації залежать від порядку розгляду вибірових векторів.

Цей алгоритм подібний до алгоритмів K -внутрішньосередніх і ІСОМАД, за винятком того, що ядра, на відміну від центрів, містять більше однієї точки.

На практиці, для визначення початкових кластерів застосовують один з попередніх алгоритмів, після чого використовується цей алгоритм, починаючи з кроку 2. Ядра можуть бути вибрані випадковим чином або отримані шляхом розбиття кластерів з подальшим використанням центрів отриманих субкластерів як ядер.

5.8 Графо-теоретичні алгоритми кластеризації

Усі розглянуті попередньо алгоритми значною мірою залежать від послідовності перегляду вибірових векторів. Графо-теоретичні методи враховують всі відстані між векторами з самого початку, що усуває вплив вказаної залежності на результати експерименту.

Базовий алгоритм, поданий нижче, еквівалентний методам мінімального остовного дерева для графів, теорія яких наведена, наприклад, в літературі. Нехай $S = \{\mathbf{x}^{(1)} \dots \mathbf{x}^{(Q)}\}$ – задана вибірка Q ознакових векторів. Для будь-якого порога відстані $\tau > 0$ формується матриця подібності графа M шляхом обчислення відстані $d_{ij} = \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|$ для всіх i та j ($i \leq j$): а якщо $d_{ij} < \tau$, то приймається значення $m_{ij}=1$ інакше – $m_{ij}=0$. Таким чином, матриця подібності M містить 0 і 1, які залежать від порога τ (для різних значень τ отримують різні матриці подібності). Звичайно, всі діагональні елементи є одиницями, тому що кожен вектор ознак має відстань від себе самого до себе, яка дорівнює 0, тобто яка менша будь-якого $\tau > 0$. Опишемо алгоритм в глобальному плані таким чином.

Пошук стовпців є рекурсивним процесом, який повертається до наступних рядків у кожному стовпці після того, як він закінчився рядками для поточного стовпця. Коли для пошуку не залишається більше стовпців, це означає, що всім знайденим векторам присвоюють номер поточного класу. Потім він оновлює номер кластера, знаходить новий вектор, який не був призначений, і починає пошук стовпця знову за номером стовпця для цього вектора. Алгоритм припиняється, коли немає більше векторів, які не були призначені.

У наступному прикладі виконується пошук за рядками; алгоритм використовує пошук стовпців.

5.8.1 Графо-теоретична кластеризація

Розглянемо матрицю подібності в таблиці 5.1. В ній i -й рядок містить запис 1 у j -му стовпці для кожного вектора $\mathbf{x}^{(j)}$, що знаходиться в межах порогової відстані τ від $\mathbf{x}^{(i)}$. Починаючи з першого рядка ($i=1$), ми бачимо, що є 1 під стовпцями для векторів $\mathbf{x}^{(1)}$, $\mathbf{x}^{(6)}$ і $\mathbf{x}^{(8)}$.

Призначення класів вибіркоким векторам в цьому алгоритмі кластеризації здійснюється за допомогою індексної функції $class(q)$, де q – номер вектора, для чого спочатку обнуляють всі її значення. Відзначені вектори $\mathbf{x}^{(1)}$, $\mathbf{x}^{(6)}$ і $\mathbf{x}^{(8)}$ поміщаємо у перший кластер, присвоюючи їм індекс першого класу індексною функцією $class(1)=class(6)=class(8)=1$. Вектори $\mathbf{x}^{(6)}$ і $\mathbf{x}^{(8)}$ заштовхуємо (шляхом запису їхніх індексів) в стек пошуку. Після цього читаємо вектор $\mathbf{x}^{(6)}$ (витагуємо його індекс) зі стека пошуку і знаходимо в шостому рядку ті одиничні елементи, що не розташовані в стовпцях з індексами $j = 1, j = 6$ і $j = 8$, бо ті вже увійшли в кластер. Виявляємо, що таких елементів у шостому рядку немає.

Таблиця 5.1 – Приклад матриці подібності

| j | i | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 6 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 8 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 10 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 12 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Далі витягуємо $\mathbf{x}^{(8)}$ зі стека пошуку (шляхом читання його індексу) і перевіряємо таким самим чином восьмий рядок. Він містить запис 1 під стовпцями $j = 1, j = 4$ та $j = 8$, тому «заштовхуємо» $\mathbf{x}^{(4)}$ у стек пошуку, а також відносимо його до першого кластера, використовуючи індексну функцію $class(4)=1$. Після читання індексу 4 зі стека, вивчаємо четвертий рядок і знаходимо одиничні значення в стовпцях з індексами $j = 4, j = 8$ і $j = 9$, після чого «заштовхуємо» $\mathbf{x}^{(9)}$ у стек пошуку і робимо присвоєння $class(9)=1$. Після чого витягуємо $\mathbf{x}^{(9)}$ зі стека пошуку і перевіряємо дев'ятий рядок, знаходимо одиничні значення в стовпцях $j = 4, j = 5$ і $j = 9$. Оскільки векторам $\mathbf{x}^{(4)}$ і $\mathbf{x}^{(9)}$ номер кластера вже було призначено, то в стек пошуку «заштовхуємо» $\mathbf{x}^{(5)}$ і призначаємо для цього вектора індекс першого кластера через індексну функцію: $class(5) = 1$. Після цього

витагуємо зі стека пошуку індекс вектора $\mathbf{x}^{(5)}$ і розглядаємо п'ятий рядок, знаходимо одиниці під стовпцями $j = 5$ і $j = 9$, але ці вектори вже були позначені. Стек пошуку пустий. Таким чином, перший кластер $\Omega_1 = \{\mathbf{x}^{(1)}, \mathbf{x}^{(4)}, \mathbf{x}^{(5)}, \mathbf{x}^{(6)}, \mathbf{x}^{(8)}, \mathbf{x}^{(9)}\}$.

Перевіряємо індексну функцію на наявність в ній нульових значень. Оскільки $class(2) = 0$ ($\mathbf{x}^{(2)}$ ще не присвоєно кластеру, то оновлюємо номер кластера з 1 до 2 і розглядаємо другий рядок. Оскільки він містить записи 1 під $j = 2$ і $j = 10$, призначаємо $\mathbf{x}^{(2)}$ і $\mathbf{x}^{(10)}$ другому кластеру індексною функцією $class(2) = class(10) = 2$ і записуємо $\mathbf{x}^{(10)}$ в стек пошуку. Після «виштовхування» $\mathbf{x}^{(10)}$ зі стека знаходимо одиниці в десятому рядку під стовпцями $j=2, j=7, j=10$ і $j=11$. Таким чином, «заштовхуємо» $\mathbf{x}^{(2)}$ і $\mathbf{x}^{(10)}$ в пошуковий стек і робимо векторам $\mathbf{x}^{(7)}$ і $\mathbf{x}^{(11)}$ призначення $class(7) = class(11) = 2$. Вилучаємо $\mathbf{x}^{(7)}$ і $\mathbf{x}^{(11)}$ зі стека і не знаходимо записів одиниць для векторів, які ще не були призначені.

Таким чином, $\Omega_2 = \{\mathbf{x}^{(7)}, \mathbf{x}^{(10)}, \mathbf{x}^{(11)}\}$.

Знову перевіряємо індексну функцію на наявність в ній нульових значень і оновлюємо номер кластера до 3, оскільки $class(3)=0$ ($\mathbf{x}^{(3)}$ ще не присвоєно кластеру. Третій рядок містить одиниці під $j = 3$ і $j = 12$, тому ми заносимо $\mathbf{x}^{(3)}$ і $\mathbf{x}^{(12)}$ в третій кластер індексною функцією $class(3) = class(12) = 3$ і записуємо $\mathbf{x}^{(12)}$ у стек пошуку. Після вибору $\mathbf{x}^{(12)}$ і перевірки дванадцятого рядка, ми не знаходимо нових записів. Таким чином, $\Omega_3 = \{\mathbf{x}^{(3)}, \mathbf{x}^{(12)}\}$.

Ще раз перевіряємо індексну функцію на наявність в ній нульових значень і бачимо, що в ній немає непозначених векторів. Отже, всі кластери було сформовано. Таким чином, алгоритм виявив такі три кластери:

$\Omega_1 = \{\mathbf{x}^{(1)}, \mathbf{x}^{(4)}, \mathbf{x}^{(5)}, \mathbf{x}^{(6)}, \mathbf{x}^{(8)}, \mathbf{x}^{(9)}\}$; $\Omega_2 = \{\mathbf{x}^{(2)}, \mathbf{x}^{(7)}, \mathbf{x}^{(10)}, \mathbf{x}^{(11)}\}$ і $\Omega_3 = \{\mathbf{x}^{(3)}, \mathbf{x}^{(12)}\}$.

Алгоритм 5.6

Графо-теоретичний алгоритм кластеризації з використанням матриці подібності графа

Входи: вибірккові значення $S = \{\mathbf{x}^{(1)} \dots \mathbf{x}^{(Q)}\}$ і поріг $\tau > 0$.

Виходи: кількість кластерів K ; кластери $\Omega_k, k=1, \dots, K$; індексна функція $class(q)=k$, яка присвоює вибіркковим векторам номер кластера.

Step 1: /* Ініціалізація параметрів */
input τ ; /* Введення порога τ */
for $q = 1$ *to* Q *do* $class(q) := 0$; /* Ініціалізація початкових значень
індексів класів до 0 */

Step 2: /* Обчислення матриці подібності M */
for $i := 1$ *to* Q *do*
for $j := 1$ *to* Q *do*

$d_{ij} = \| x^{(i)} - x^{(j)} \|;$ /* Обчислити відстані, використовуючи симетрію матриці */

if $d_{ij} < \tau$ then ($m_{ij}:=1$; $m_{ij}:=1$)
else ($m_{ij}:=1$; $m_{ij}:=1$);

Step 3: /* Рекурсивна процедура перевірки всередині повторюваного циклу */

$k:=1$; /* Ініціалізація номера класу */
 $class(1):=1$; $j:=1$; /* Призначити вектор $x^{(1)}$ до класу 1 */

repeat

$Check_column(j, k)$; /* Виклик рекурсивної функції для пошуку в стовпцях */

$k:=k+1$; /* Оновлення номера класу */

$next_col := Get_next_column()$; /* Вибрати неперевірений стовпець */

until $next_col := 0$;

Функції **Check_column(j, k)** і **Get_next_column()** є простими; їх наведено нижче. Функція **Check_column(j, k)** є рекурсивною, і вона викликає себе повторно, поки не вичерпає зв'язок, починаючи з деякого рядка.

Procedure Check_column(j, k):

 for $i := 1$ to Q do /* Check row numbers greater than column number */

 if $m_{ij}:=1$ and $class(i):=0$ then /* If close enough, then assign the new */
 $class(i):=k$; /* vector to the current class and use the */

$Check_column(j, k)$; /* row as a column to search: Depth First Search */

Procedure Get_next_column():

$key_col := 0$; /* Initialize next unassigned vector to 0 */

 for $q = t$ to Q do /* If q-th vector has not been assigned */

 if $class(q):=0$ then /* then it is a candidate for next column search */

$key_col := q$; exit loop; /* Get out of loop when unassigned vector found */

 return key_col ;

До переваг цього методу можна віднести:

- здатність до повної самоорганізації під час визначення кількості кластерів;

- спроможність кластеризації даних з нелінійною розподільною границею (кластери не обов'язково мають бути у вигляді гіперкуль, а можуть мати продовгуватий або викривлений вигляд);

- порядок подання векторів вибірки не має ефекту.

Недоліками є такі:

- для великих вибірових наборів матриця подібності стає великою (розмір вибірки $Q=1\ 000$ потребує зберігати $1\ 000\ 000$ елементів або

принаймні 500 000, якщо використовується половина симетричної матриці);

- можливо, що два або більше класів можуть бути введені в один кластер, оскільки може існувати послідовність векторів ознак, близьких між собою, але в якій два або більше класи:

- значення порога τ потрібно вибирати в умовах відсутності інформації про його величину. Вимоги до пам'яті можуть бути дещо зменшені порівняно з іншими алгоритмами, але залишається необхідність розбивати кластери на більш дрібні.

На практиці цей метод може бути використаний з різними значеннями τ для отримання, можливо, лінійної класифікації, яка мінімізує загальну відстань, що є сумою відстаней за всіма кластерами сум відстаней між всіма парами векторів у межах кожного кластера. Він також може бути використаний для отримання результатів, необхідних для роботи іншого алгоритма, як, наприклад, алгоритм динамічних кластерів.

Контрольні питання та завдання

1. Що називають кластером і кластеризацією?
2. Що таке «навчання без учителя» і коли воно застосовується?
3. Для вирішення яких завдань використовується кластерний аналіз?
4. Назвіть відомі вам міри схожості об'єктів?
5. З яких міркувань вибираються міри схожості?
6. Які неметричні міри схожості ви знаєте?
7. Як називається двійковий варіант косинусної міри схожості?
8. Назвіть етапи узагальненої процедури кластеризації згідно з К. Фу.
9. Опишіть алгоритм кластеризації за пороговою відстанню і назвіть його недоліки і переваги.
10. Чому алгоритм максимінної кластеризації має таку назву?
11. Яка відстань в алгоритмі максимінної кластеризації називається «типовою»?
12. З яких кроків складається алгоритм максимінної кластеризації і коли він застосовується?
13. Опишіть алгоритм кластеризації методом К-внутрішньогрупових середніх.
14. Які переваги і недоліки алгоритму k-means?
15. Назвіть переваги особливості алгоритму ІСОМАД і його недоліки та переваги.
16. В чому основана відмінність алгоритму динамічних ядер і які його недоліки і переваги?
17. Чи впливає послідовність перегляду вибіркового векторів в графо-теоретичних методах кластеризації?
18. Які графо-теоретичні методи кластерного аналізу вам відомі?
19. Назвіть переваги і недоліки графо-теоретичних методів кластеризації.

РОЗДІЛ 6

ВИБІР МНОЖИНИ ОЗНАК В СИСТЕМАХ РОЗПІЗНАВАННЯ

6.1 Характеристика проблеми вибору множини ознак образів.

Будь-який об'єкт або зображення, що підлягає розпізнаванню і класифікації, має низку характерних (розрізнявальних) якостей або ознак. Першим кроком всякого процесу розпізнавання є аналіз задачі вибору розрізнявальних ознак і визначення способу їх вимірювання. Цей аналіз показує, що розробка словника ознак, який використовується системою розпізнавання, є досить складною проблемою, розв'язання якої пов'язано з низкою обмежень. По-перше, кількість ознак, необхідних для опису образів, залежить від їх розподільних властивостей, а найбільш якісні з цього погляду ознаки досить часто неможливо виміряти через технічні або економічні обмеження. По-друге, біологічні системи розпізнавання (людина і інші живі істоти) використовують для розпізнавання об'єктів зовнішнього світу переважно фізичні та структурні ознаки. До фізичних ознак можна віднести такі ознаки як колір, смак, запах, а до структурних – форму, об'єм, взаємне розміщення елементів структури та інші геометричні властивості образів. Штучні системи розпізнавання «не володіють» методами сприйняття і обробки інформації, характерними для біологічних систем. Наприклад, паралельне сприйняття зображень зоровою системою людини і паралельна їх обробка нейронами мозку дають можливість одномоментно відтворити такі інтегральні ознаки як форма об'єкта, його розмір, структурні особливості у вигляді взаємного розміщення непохідних елементів та інше. Крім того, непохідні (первинні) елементи структури сприймаються такою системою розпізнавання як деякий готовий елементарний образ, а не будуються з окремих точок.

Тому штучні системи на сьогодні переважно використовують математичні ознаки, які виділяються з отриманих даними інформаційних сигналів про об'єкт шляхом математичних перетворень і обчислень деяких стійких характеристик об'єкта: статистичних середніх, коефіцієнтів кореляції, характеристичних чисел й власних векторів коваріаційних матриць та інших інваріантних ознак. В математичному підході до попередньої обробки образів і зниження розмірності даних задача виділення ефективного набору ознак займає центральне місце. В нетривіальній постановці ця задача має вирішуватись з погляду оптимізації роботи всієї системи розпізнавання загалом, шляхом паралельного вирішення задач вибору оптимального алфавіту образів і розробки оптимального алгоритму класифікації. Один із методів такого вирішення задачі описаний в роботах авторів посібника, наведених в списку літератури. В цьому розділі об'єкти описуються математичними ознаками у вигляді векторів чи точок багатовимірного простору (числового або

двійкового). Принципи використання структурних ознак для опису і розпізнавання зорових зображень будуть розглянуті в наступному розділі.

Останнім часом в теорії розпізнавання для математичного опису образів все більше використовують математичний апарат нечітких множин.

В цьому математичному підході ступінь впевненості про належність деякого зображення до певного класу (лінгвістичного терму) за заданої величини ознаки визначається значенням функції належності. Такий підхід описує особливості людського мислення в формуванні логічних висновків і відповідає заміні двозначної або k -значної логіки континуальною (неперервною) логікою. В цьому випадку розпізнавання здійснюється у разі одночасного (паралельного) збігу правил послідовного виведення ЯКЦО – ТО Тому застосування математичного апарату нечітких множин можна вважати проміжним кроком переходу від послідовної обробки інформації в штучних інтелектуальних системах, реалізованої класичним математичним апаратом, до паралельної в біологічних системах. В більш обмеженому підході, який і буде розглядатися в цьому розділі, задача виділення інформативних ознак полягає спочатку в визначенні ряду інваріантних властивостей образів, після чого на основі цих властивостей знижують розмірність векторів зображень за допомогою лінійного перетворення. В підрозділах 6.3, 6.4 і 6.5 розглянуто вибір множини ознак, оптимальної за критерієм, що не пов'язаний безпосередньо з якістю класифікації. Такий підхід можна вважати вибором ознак без врахування обмежень. Інший підхід (підрозділи 6.6 і 6.7) пов'язує вибір множини ознак з якістю класифікації, яка визначається в термінах імовірності правильного розпізнавання.

6.2 Використання відстаней в задачі виділення ознак

Поняття відстані відіграє фундаментальну роль в проблемі вибору інформативних ознак. Це пов'язано з тим, що сама категорія «ознака» визначає таку характеристику об'єкта деякого класу, яка відрізняє його від об'єктів інших класів, і ступінь його відрізняваності якраз й визначається відстанню між зображеннями цих об'єктів в просторі цих ознак. Об'єкти в просторі цих ознак зображаються векторами або точками. Група близьких між собою точок утворює образ (клас): образи розділені між собою значною відстанню. Тому середньоквадратичне відхилення відстані між точками всередині класу поряд з середньоквадратичним відхиленням відстаней між точками різних класів можуть використовуватися як критерій якості процедури зменшення розмірності простору ознак. Введемо поняття відстані між окремими точками, між точками і множиною точок, а також між множинами точок.

Відстань $D(\vec{a}, \vec{b})$ між точками \vec{a} і \vec{b} в N -вимірному евклідовому просторі визначають таким чином

$$D(\vec{a}, \vec{b}) = \|a - b\| = \sqrt{(\vec{a} - \vec{b})^T (\vec{a} - \vec{b})} = \sqrt{\sum_{n=1}^N (a_n - b_n)^2} \quad (6.1)$$

де \vec{a} і \vec{b} є N -вимірними векторами, n -і компоненти яких дорівнюють a_n і b_n , відповідно.

Відстань між точкою \vec{x} і множиною точок $\{\vec{a}^k\}$, $k=1, \dots, K$, визначається як середньоквадратична відстань між точкою \vec{x} і K точками заданого класу. Квадрат відстані між точками \vec{x} і \vec{a}^k дорівнює:

$$D^2(\vec{x}, \vec{a}^k) = (\vec{x} - \vec{a}^k)^T = \sum_{n=1}^N (x_n - a_n^k)^2, \quad (6.2)$$

Звідси відстань визначиться таким чином:

$$\overline{D^2(\vec{x}, \{\vec{a}^k\})} = \frac{1}{K} \sum_{k=1}^K D^2(\vec{x}, \vec{a}^k) = \frac{1}{K} \sum_{k=1}^K \sum_{n=1}^N (x_n - a_n^k)^2, \quad (6.3)$$

$$\overline{D^2(\{\vec{a}^k\}, \{\vec{a}^j\})}, \quad k, j = 1, 2, \dots, k-1; k \neq j. \quad (6.4)$$

Враховуючи з (6.2) те, що

$$D^2(\vec{a}^k, \vec{a}^j) = \sum_{n=1}^N (a_n^k - a_n^j)^2, \quad (6.5)$$

а з (6.3) те, що

$$\overline{D^2(\vec{a}^k, \{\vec{a}^j\})} = \frac{1}{K-1} \sum_{k=1}^K \sum_{n=1}^N (a_n^k - a_n^j)^2, \quad (6.6)$$

і усереднюючи по всіх точках множини $\{\vec{a}^k\}$, отримуємо вираз для внутрішньомножинної відстані:

$$\overline{D^2(\{\vec{a}^k\}, \{\vec{a}^j\})} = \frac{1}{K(K-1)} \sum_{k=1}^K \sum_{j=1}^K \sum_{n=1}^N (a_n^k - a_n^j)^2. \quad (6.7)$$

Неважко показати, що внутрішньомножинну відстань можна виразити через незміщену оцінку вибіркової дисперсії:

$$\overline{D^2} = 2 \sum_{n=1}^N (\sigma_k)^2. \quad (6.8)$$

Відстань між множинами точок $\{\vec{a}^k\}$ і $\{\vec{b}^j\}$, якщо містять відповідно K_a і K_b вибірових зображень, визначається, як

$$D^2(\{\vec{a}^k\}, \{\vec{b}^j\}), k = 1, 2, \dots, K_a; j = 1, 2, \dots, K_b, \quad (6.9)$$

Нажаль, цей вираз важко привести до простого замкнутого вигляду, тому його часто замінюють відстанню між центроїдами (еталонами) цих множин або відстанню Махаланобіса (5.9). Відстань Махаланобіса використовується для ознак ймовірнісного типу, тобто тоді, коли ознаки є випадковими величинами. В такому випадку зниження розмірності простору ознак базується на методі аналізу головних компонент (PCA – principal component analysis). Інші способи зменшення розмірності даних – це метод незалежних компонент, багатовимірне масштабування, а також численні нелінійні узагальнення: метод головних кривих і різноманітностей, пошук найкращої проєкції (англ. Projection Pursuit), самоорганізаційні карти Кохонена, нейромережеві методи та ін. Основні ідеї методу PCA розглянуто в наступному підрозділі.

6.3 Зменшення розмірності простору ознак методом PCA

Метод PCA ґрунтується на тих міркуваннях, що часто ознаки досить сильно залежать одна від одної. Знаючи залежності і їх силу, можна виразити кілька ознак через одну, зливу воедино, і працювати вже з простішою моделлю образу. Метод розроблений К. Пірсоном (англ. Karl Pearson) в 1901 р. Застосовується в багатьох галузях, таких як розпізнавання образів, комп'ютерний зір, стиснення даних та ін. Обчислення головних компонент зводиться до обчислення власних векторів і власних значень коваріаційної матриці вихідних даних або до сингулярного розкладання матриці даних. Іноді метод головних компонент називають перетворенням Карунена-Лоева (англ. Karhunen-Loeve) або перетворенням Хотеллінга (англ. Hotelling transform). Завдання аналізу головних компонент може реалізовуватися на використанні таких чотирьох базових версій:

- апроксимація даних лінійними просторами меншої розмірності;
- знаходження підпросторів меншої розмірності в ортогональній проєкції, на якій розкид даних (тобто середньоквадратичного відхилення від середнього значення) максимальний;
- знаходження підпросторів меншої розмірності в ортогональній проєкції, на якій середньоквадратична відстань між точками максимальна;
- побудова для цієї багатовимірної випадкової величини такого ортогонального перетворення координат, внаслідок якого кореляції між окремими координатами згорнуться в нуль.

Перші три версії оперують зі скінченними множинами даних. Вони еквівалентні і не використовують жодної гіпотези про статистичну природу (походження) даних. Четверта версія оперує випадковими величинами, в яких скінченні множини з'являються як вибірки з цього розподілу, а рішення трьох перших завдань – як наближення до «істинного» перетворення Карунена-Лоева. В такому випадку виникає додаткове і не цілком тривіальне питання про точність цього наближення.

Подальше викладення суті методу PCA орієнтується на другу базову версію – знаходження підпросторів меншої розмірності, в ортогональній проекції, на якій розкид даних (тобто середньоквадратичного відхилення від середнього значення) максимальний. Цей підхід можна коротко описати таким чином. Нехай нам дано центрований набір векторів даних, в якому середнє арифметичне значення (математичне сподівання) кожної координати дорівнює нулю. Завдання – знайти таке ортогональне перетворення в нову систему координат, для якого були б справедливі такі умови:

- вибіркова дисперсія даних уздовж першої координати максимальна (цю координату називають першою головною компонентою, рис. 6.1);



Рисунок 6.1 – Приклад побудови першої головної компоненти

- вибіркова дисперсія даних уздовж другої координати максимальна за умови ортогональності першій координаті (друга головна компонента);
- і т. д. ...;
- вибіркова дисперсія даних уздовж значень k -ої координати максимальна за умови ортогональності перших координат до $k-1$.

Обчислення головних компонент зводиться до обчислення власних векторів і власних значень коваріаційної матриці вихідних даних.

6.3.1 Застосування коваріаційної матриці в задачі аналізу інформативності ознак

Одновимірна випадкова величина повністю описується такими математичними характеристиками як її середнє значення (математичне сподівання) та розкид значень відносно нього (дисперсія). У випадку з багатовимірною випадковою величиною (випадковим вектором)

положення центра буде таким самим чином визначатися проєкціями її математичних сподівань її проєкцій на осі координат. Однак для опису її форми вже недостатньо тільки її дисперсій по осях, що видно з наведених на рис. 6.2 графіків. На них видно, що у всіх трьох випадкових величин різні форми, але математичні сподівання, дисперсії та їх проєкції на осі загалом виявляються однаковими.

Для опису форми випадкового вектора необхідна коваріаційна матриця. Як було показано в розділі 4, коваріаційна матриця для n -вимірному простору ознак визначається таким чином:

$$\sum_i = E_i \{ (\vec{x} - \vec{m}_i)(\vec{x} - \vec{m}_i)^T \}, \quad (6.10)$$

де $\vec{x} = (x_1, x_2, \dots, x_n)$ і $\vec{m} = (m_1, m_2, \dots, m_n)$;

$E_i \{ \vec{x} \}$ – оператор математичного сподівання, визначений на зображеннях класу Ω_i ;

n – розмірність вектора образів.

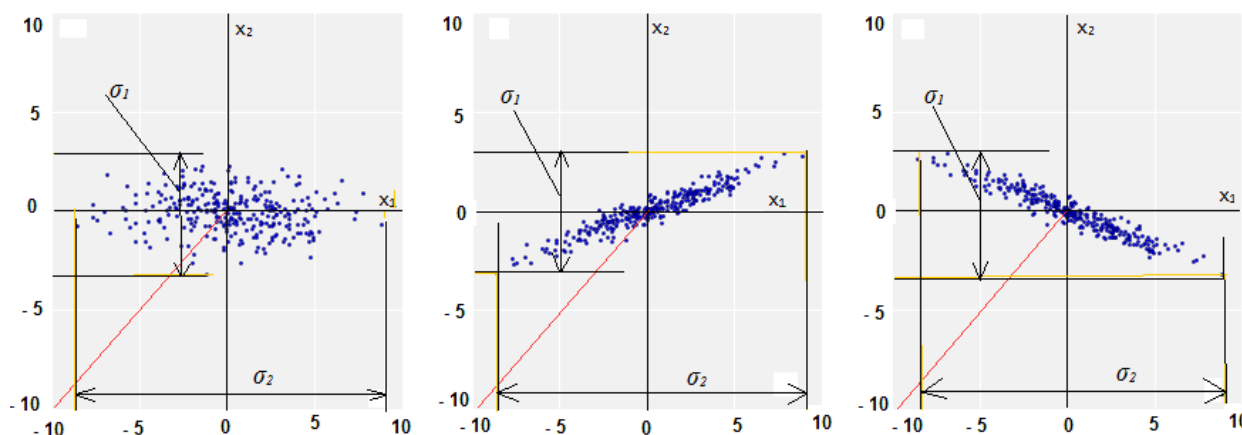


Рисунок 6.2 – Приклад випадкових величини різної форми, але з однаковими проєкціями середніх значень і дисперсій на осі

Коваріаційна матриця \sum_i є симетричною і додатно напіввизначеною. Її діагональний елемент σ_{kk}^2 є дисперсією k -ої координати вектора зображення. Елемент σ_{jk}^2 , що не лежить на головній діагоналі матриці, являє собою коваріацію випадкових змінних x_j і x_k (j -ої і k -ої координат вектора \vec{x}). Якщо ознаки x_j і x_k статистично незалежні, то елемент матриці $\sigma_{jk}^2 = 0$.

Коваріаційна матриця є узагальненням дисперсії на випадок багатовимірних випадкових величин – вона так само описує форму (розкид) випадкової величини, як і дисперсія. Дійсно, дисперсія

одновимірної випадкової величини – це коваріаційна матриця розміру 1×1 , в якій її єдиний член задано формулою $E(x, x) = \sigma_x^2$.

З допомогою коваріаційної матриці можна знайти такий вектор, за якого максимізувався б розмір (дисперсія) проекції досліджуваної вибірки на нього.

Проекція випадкового вектора \vec{x} на одиничний вектор \vec{v}^T дорівнює $\vec{v}^T \vec{x}$. Дисперсія проекції на вектор буде відповідно дорівнювати $Var(\vec{v}^T \vec{x})$. У загальному вигляді у векторній формі (для центрованих величин) дисперсія виражається так:

$$Var(\vec{x}) = \Sigma = E(\vec{x} \cdot \vec{x}^T).$$

Відповідно дисперсія проекції:

$$Var(\vec{x}^*) = \Sigma^* = E(\vec{x}^* \cdot \vec{x}^{*T}) = E((\vec{v}^T \vec{x}) \cdot (\vec{v}^T \vec{x})^T) = \vec{v}^T \Sigma \vec{v}$$

Дисперсія максимізується за максимального значення $\vec{v}^T \Sigma \vec{v}$. Для його знаходження використаємо співвідношення Релея. Співвідношення Релея для коваріаційних матриць має вигляд:

$$R(\Sigma, \vec{x}^T) = \frac{\vec{x}^T \Sigma \vec{x}}{\vec{x}^T \vec{x}} = \lambda \frac{\vec{x}^T \vec{x}}{\vec{x}^T \vec{x}} = \lambda, \quad (6.11)$$

де

$$\Sigma \cdot \vec{x} = \lambda \cdot \vec{x}.$$

Остання формула відповідає виразу розкладання матриці на власні вектори і власні значення. В ній \vec{x} є власним вектором, а λ – власним значенням. Кількість власних векторів (eigenvectors) і власних значень (eigenvalues) дорівнюють розміру матриці (значення можуть повторюватися).

Таким чином, напрямком максимальної дисперсії у проекції завжди збігається з власним вектором коваріаційної матриці, який має максимальне власне значення, що дорівнює величині цієї дисперсії.

І це справедливо також для проекцій на більшу кількість вимірювань – дисперсія (коваріаційна матриця) проекції на m -вимірний простір буде максимальна в напрямку тих m власних векторів, які мають максимальні власні значення.

Метод PCA, реалізований у вигляді бібліотечної функції, автоматично проектує на осі, максимізуючи дисперсію. Це пов'язано з тим, що найбільш інформативні ознаки мають найбільшу ентропію (розкид значень). Однак для випадку класифікації об'єктів на класи це не завжди раціонально, що видно з рис. 6.3.

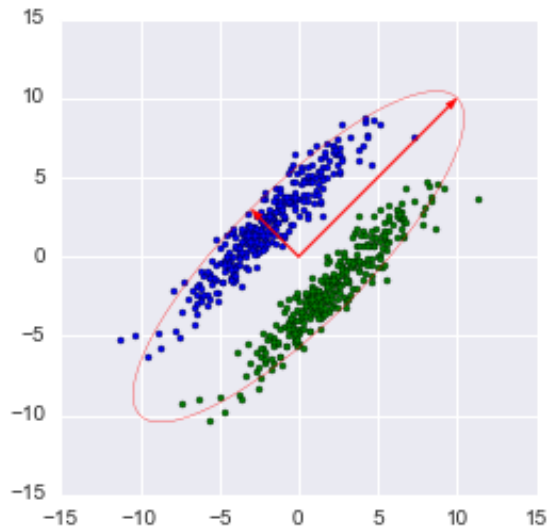


Рисунок 6.3 – Приклад зменшення розмірності простору ознак за принципом мінімуму ентропії

З рис. 6.3 видно, що зниження розмірності за принципом максимуму ентропії призведе до втрати можливості класифікації. В такому випадку виконують проектування на менший власний вектор (використовують принцип мінімуму ентропії, описаний в розділі 6.5). Як видно з рис. 6.3, проектування вибірки на менший вектор успішно знижує розмірність простору ознак і зберігає класифікатор.

6.3.2 Простий приклад зменшення розмірності простору ознак методом PCA

Приклад 6.1: Задано вибірку 10-и векторів у двовимірному просторі ознак: $\{(1.00; 2.75), (2.00; 4.35), (3.0;7.21), (4.0; 11.25), (5.0; 9.59), (6.0; 12.10), (7.0;13.79), (8.0;13.85), (9.0; 15.29), (10.0; 18.10)\}$. Точки, які є кінцями вибірових векторів, зображено на рис. 6.4.

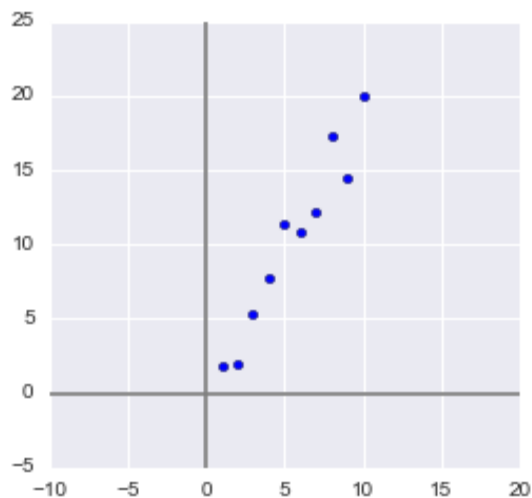


Рисунок 6.4 – Зображення точок вибірки

Завдання: використовуючи метод PCA, зменшити розмірність простору ознак з двох до одного.

Розв'язування. Для мінімізації втрат інформації під час проектування значень випадкової величини на вектор він має проходити через центр нашої вибірки, водночас процес проектування даних не буде впливати на їхні середні значення. Тому для спрощення перетворень у цьому процесі відцентруємо досліджувану вибірку, для чого лінійно зсунемо її так, щоб середні значення ознак дорівнювали 0:

$$\begin{aligned}x'_{i1} &= x_{i1} - m_1, \\x'_{i2} &= x_{i2} - m_2,\end{aligned}$$

де $(m_1; m_2)$ – координати центра множини вибіркових точок.

Отримаємо відцентровані координати точок вибірки (рис. 6.5): $\{(-4.5; -8.45), (-3.5; -8.32), (-2.5; -4.93), (-1.5; -2.57), (-0.5; 1.01), (0.5; 0.58), (1.5; 1.86), (2.5; 7.01), (3.5; 4.21), (4.5; 9.59)\}$, $m=(5.5; 10.31)$.

На відміну від математичного сподівання дисперсія сильно залежить від порядків розкиду значень випадкової величини, тобто чутлива до масштабування. Тому якщо одиниці виміру ознак сильно розрізняються своїми порядками, рекомендується їх стандартизувати шляхом ділення центрованих значень випадкових величин на середньоквадратичні відхилення:

$$x'_{i1} = (x_{i1} - m_1) / \sigma(x_1); \quad x'_{i2} = (x_{i2} - m_2) / \sigma(x_2),$$

де $i=1, 2, \dots, 10$ – номер вектора нашої вибірки;

$$m_1 = \frac{1}{10} \sum_{i=1}^{10} x_{i1}, \quad m_2 = \frac{1}{10} \sum_{i=1}^{10} x_{i2} \quad - \text{математичні сподівання 1-ої і 2-ої}$$

координат вибіркових векторів, відповідно;

$$\sigma(x_1) = \sqrt{\frac{1}{10-1} \sum_{i=1}^{10} (x_{i1} - m_1)^2}, \quad \sigma(x_2) = \sqrt{\frac{1}{10-1} \sum_{i=1}^{10} (x_{i2} - m_2)^2} \quad - \text{середньоквадратичні}$$

відхилення координат векторів від їхнього математичного сподівання.

В загальному випадку для n -вимірного простору ознак

$$u_{ij} = (x_{ij} - \mu_j(x_j)) / \sigma(x_j), \quad i = \overline{1, n}, j = \overline{1, m},$$

де $\mu_j(x_j) = \frac{1}{n} \sum_{i=1}^n x_{ij}$ – математичне сподівання j -ої координати вектора;

$$\sigma_j^2(x_j) = \sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \mu_j(x_j))^2 \quad - \text{дисперсія } j\text{-ої координати вектора.}$$

Для прикладу, який розглядається, значення не сильно різняться в порядках, тому для спрощення розуміння ця операція не виконується.

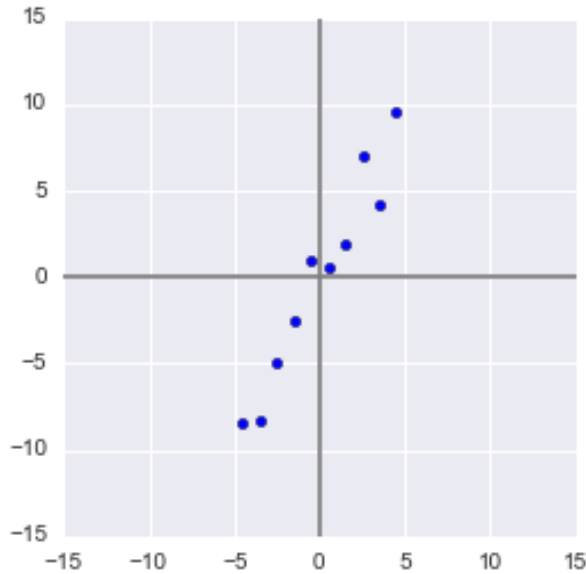


Рисунок 6.5 – Зображення точок відцентрованої вибірки

Обчислимо коваріаційну матрицю і її власні значення та власні вектори (eigenvalues та eigenvectors). Нагадуємо (підрозділ 4.4), що для відцентрованих випадкових величин коваріаційна матриця обчислюється за формулою (дивись формули (4.48), (4.49)):

$$\text{Var}(X_i, X_j) = E\{\bar{x}\bar{x}^T\} = \begin{bmatrix} E\{x_1x_1\} & E\{x_1x_2\} & \dots & E\{x_1x_n\} \\ E\{x_2x_1\} & E\{x_2x_2\} & \dots & E\{x_2x_n\} \\ \dots & \dots & \dots & \dots \\ E\{x_nx_1\} & E\{x_nx_2\} & \dots & E\{x_nx_n\} \end{bmatrix},$$

$$\Sigma = \frac{1}{N} \sum_{i=1}^N \bar{x}_i \bar{x}_i^T .$$

Отже, сформуємо коваріаційну матрицю Σ для нашої вибірки. Для цього порахуємо дисперсії координат (ознак) x_i і x_j , а також їх коваріацію. Можна скористатися вищенаведеною формулою, але в мові Python можна використати готову функцію `numpy.cov(X)` бібліотеки **numpy**. Вона приймає на вхід список всіх ознак випадкової величини і повертає її коваріаційну матрицю. В цій функції X є n -вимірним випадковим вектором (n – кількість рядків). Функція добре підходить і для розрахунку незміщеної дисперсії, і для коваріації двох величин, і для побудови коваріаційної матриці.

$$\Sigma = \begin{bmatrix} 9.17 & 17.93 \\ 17.93 & 37.26 \end{bmatrix}$$

Оскільки розмірність заданої вибірки дорівнює двом, то кількість власних векторів у неї, відповідно, дорівнює 2. Знайдемо їх за допомогою бібліотеки **numpy**, в якій реалізовано функцію `numpy.linalg.eig(X)`, де X – квадратна матриця. Вона повертає 2 масиви – масив власних значень і масив власних векторів (у вигляді векторів-стовпців). Ці вектори нормовані – їх довжина дорівнює 1, і вони задають новий базис для вибірки, такий, що його осі збігаються з півосями еліпса, який апроксимує вихідну вибірку.

На рис. 6.6 показано апроксимацію заданої вибірки векторів еліпсом з радіусами в 2 сигми (тобто він містить в собі 95 % всіх точок – що тут і спостерігається). Оскільки нам важливий напрямок, то більший вектор тут інвертований (функція `eig(X)` спрямовувала його в зворотний бік).

Для проведення проєкції на кроці 3 потрібно виконати операцію проєктування вибірки даних на вектор нормалі, тобто $v^T X$ (вектор має бути довжини 1). Або, якщо у нас не один вектор, а гіперплощина, то замість вектора v^T беремо матрицю базисних векторів V^T . Отриманий вектор (або матриця) буде масивом проєкцій наших спостережень:

$$X = (-9.56, -9.02, -5.53, -2.96, 0.69, 0.74, 2.33, 7.39, 5.32, 10.60).$$

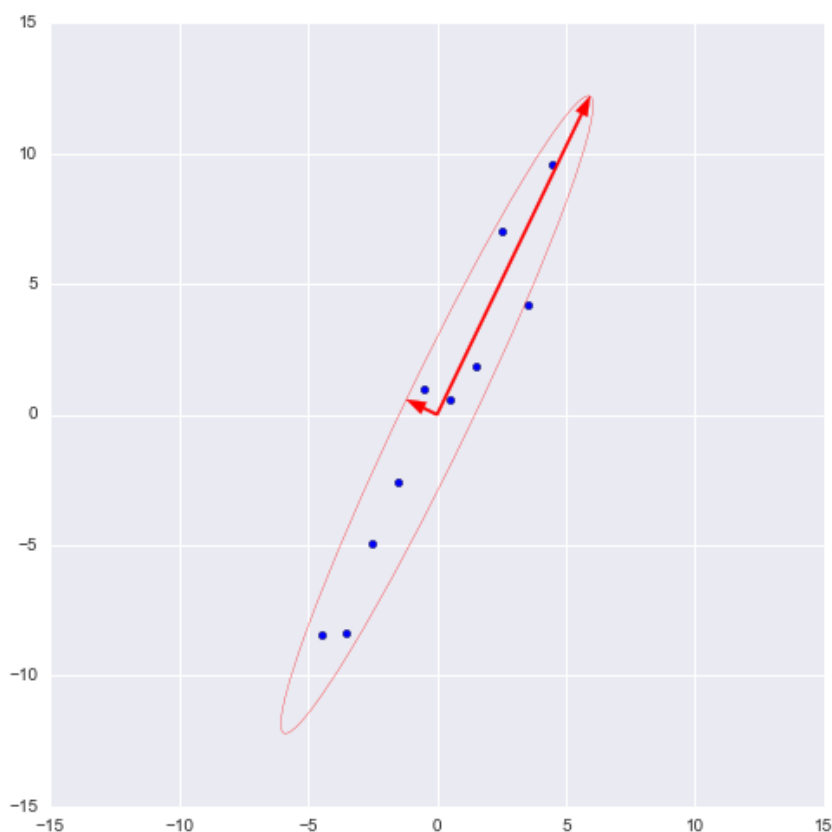


Рисунок 6.6 – Апроксимація вибірки еліпсом, півосі якого є власними векторами коваріаційної матриці

6.4 Вплив лінійного перетворення простору ознак на визначення їх важливості

На першому кроці створення системи розпізнавання збирають інформацію про об'єкти, що підлягають розпізнаванню, шляхом вимірювання їх параметрів всіма доступними технічними засобами. На наступному кроці із них відбирають найбільш важливі (інформативні) з погляду їх здатності розрізняти об'єкти одних класів від об'єктів інших класів. Це дає можливість подавати зображення об'єктів в просторі меншої розмірності і, таким чином, спростувати процедуру розпізнавання.

Призначення ваг ознакам можна здійснити шляхом лінійного перетворення, яке забезпечить більш сприятливе групування точок, що є зображеннями об'єктів, в новому просторі. Пояснимо це на простому, з погляду геометричної ілюстрації, прикладі (рис. 6.7).

Припустимо, що є множина $\{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_Q\}$ N -вимірних вибірових векторів, з яких Q_1 відноситься до класу Ω_1 , і Q_2 – до класу Ω_2 . Можна зменшити розмірність простору з N вимірів до одного шляхом проектування N -вимірних точок на пряму. Для цього утворимо лінійну комбінацію компонент вектора \bar{x} , отримаємо скалярну величину

$$y = \vec{W}^T \bar{x}, \quad (6.12)$$

де \vec{W} – вектор ваг.

За умови $\|\vec{W}\|=1$ лінійне перетворення (6.12) дасть точки y_i , які є проєкціями точок \bar{x}_i на пряму в напрямку вектора ваг \vec{W} .

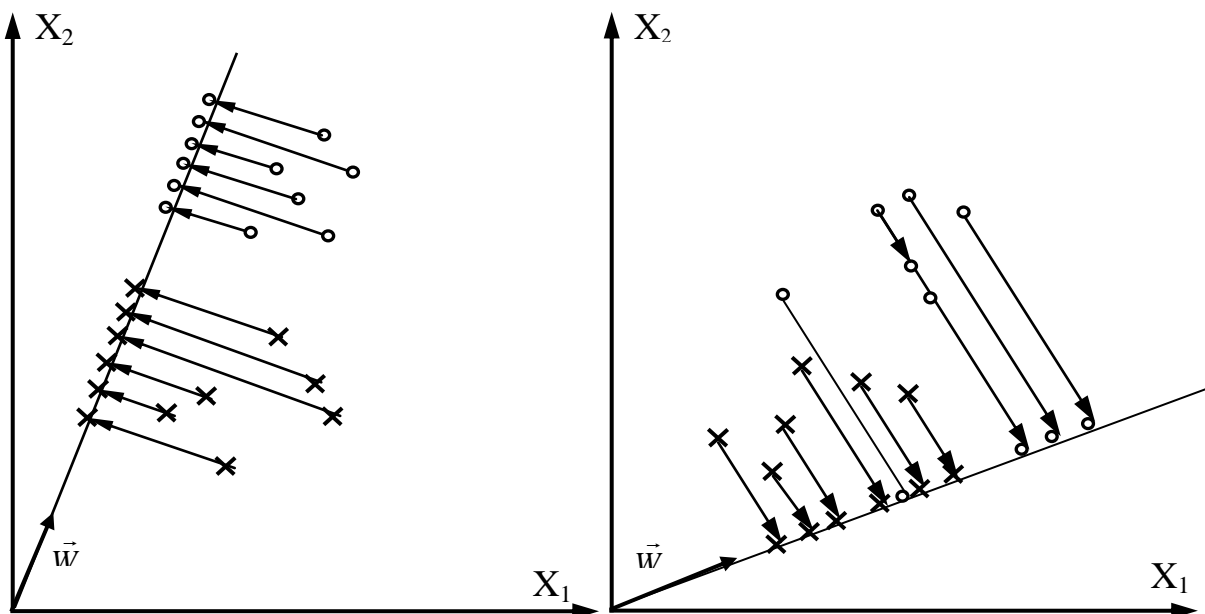


Рисунок 6.7 – Два способи перетворення простору ознак

Точки y_1, y_2, \dots, y_Q будуть якимось чином, залежно від напрямку \vec{W} , розподілені по класах Ω_1 і Ω_2 . З рисунку 6.5 видно, що в одному випадку спроектовані точки окремих класів утворюють компактні групи, які достатньо віддалені одна від одної, а в іншому – перемішані між собою. Звідси видно, що вибір вагового вектора \vec{W} впливає на якість розподілу класів в просторі ознак. Природньо вважати, що ця якість збільшується під час збільшення відстані між середніми значеннями класів і під час зменшення розкиду між точками всередині класів, тобто за більш тісного групування їх всередині класів. Тому в цьому простому випадку вигляд лінійного перетворення \vec{W} знаходять шляхом мінімізації критерію

$$I(\vec{W}) = \frac{|\vec{m}_1^* - \vec{m}_2^*|}{\tilde{\sigma}_1^2 + \tilde{\sigma}_2^2}, \quad (6.13)$$

де \vec{m}_1^*, \vec{m}_2^* – середнє вибіркве значення класів Ω_1 і Ω_2 в просторі зменшеної розмірності,

$\tilde{\sigma}_1^2$ і $\tilde{\sigma}_2^2$ – дисперсії вибірок в класах Ω_1 і Ω_2 .

Вираз (6.13) можна переписати у вигляді (6.14):

$$I(\vec{W}) = \frac{\vec{W}^T S_B \vec{W}}{\vec{W}^T S_W \vec{W}}, \quad (6.14)$$

де S_W – матриця розкиду всередині класів,

$$S_W = S_1 + S_2 = \sum_{\vec{x} \in \Omega_1} (\vec{x} - \vec{m}_1)(\vec{x} - \vec{m}_1)^T + \sum_{\vec{x} \in \Omega_2} (\vec{x} - \vec{m}_2)(\vec{x} - \vec{m}_2)^T, \quad (6.15)$$

а S_B – матриця розкиду між класами,

$$S_B = (\vec{m}_1 - \vec{m}_2)(\vec{m}_1 - \vec{m}_2)^T. \quad (6.16)$$

В виразах (6.15) і (6.16) \vec{x} – вибіркві вектори, а \vec{m}_1 і \vec{m}_2 – середні вибіркві значення класів Ω_1 і Ω_2 в початковому N -вимірному просторі ознак.

Вже з цього простого прикладу видно, що вигляд лінійного перетворення W для зниження розмірності простору ознак визначається статистичними характеристиками вибірки в початковому просторі.

Розглянемо більш загальний підхід до знаходження оптимального, відносно критерія (6.14) лінійного перетворення простору. Воно називається перетворенням кластеризації і дозволяє зменшити розмірність N простору ознак. Цей підхід полягає в виконанні таких кроків. Спочатку виконують перетворення векторів \vec{x} простору X в вектори \vec{y} простору Y за

допомогою матриці W , яка мінімізує внутрішньомножинну відстань в просторі Y :

$$\bar{y} = W\bar{x}, \quad (6.17)$$

У цьому випадку вектор математичного сподівання \bar{m}^* і коваріаційна матриця Σ^* в новому просторі відповідно дорівнюють:

$$\bar{m}^* = E\{\bar{y}\} = WE\{\bar{x}\} = W\bar{m}, \quad (6.18)$$

$$\begin{aligned} \Sigma^* &= E\{(\bar{y} - \bar{m}^*)(\bar{y} - \bar{m}^*)^T\} = E\{A(\bar{x} - \bar{m})\{W(\bar{x} - \bar{m})\}^T\} = \\ &= WE\{(\bar{x} - \bar{m})(\bar{x} - \bar{m})^T\}W^T = W\Sigma W^T \end{aligned} \quad (6.19)$$

де \bar{m} – вектор математичного сподівання, а Σ – коваріаційна матриця вибірових точок в просторі X . В процесі виведення виразу (6.18) використано відоме в матричній алгебрі співвідношення:

$$(WM)^T = M^T W^T, \quad (6.20)$$

де W і M – матриці. Якщо замість M підставити $(\bar{x} - \bar{m})$ з виразу (6.20), то матимемо

$$\{W(\bar{x} - \bar{m})\}^T = (\bar{x} - \bar{m})^T W^T. \quad (6.21)$$

Після першого перетворення (6.21) виконують друге, ортонормоване лінійне перетворення

$$\bar{y}^* = A \cdot \bar{y}, \quad (6.22)$$

яке збільшує відстань між множинами точок (розкид між класами) і здійснює декореляцію ознак шляхом діагоналізації коваріаційної матриці Σ^{**} множини точок в новому просторі.

6.5 Вибір ознак за принципом мінімізації ентропії

В попередніх розділах було визначено, що найбільш ефективними для задачі класифікації є такі ознаки, які мінімізують внутрішньогрупові відстані між зображеннями в класах і максимізують міжкласові відстані, тобто такі, які зменшують невизначеність розміщення образів в ознаковому просторі. Зручною мірою невизначеності заданої вибірки зображень є ентропія:

$$H = -E\{\ln p\}, \quad (6.23)$$

де p – щільність імовірності сукупності зображень,

E – оператор математичного сподівання щільності p .

Тому поняття ентропії зручно використовувати за критерій в процедурі вибору оптимального набору ознак. Ознаки, які зменшують невизначеність заданої ситуації, вважаються більш інформативними, ніж ті, які викликають протилежну дію. Виходячи з цих міркувань, вибір оптимальних ознак можна здійснити шляхом мінімізації ентропії класів, що утворюються заданою вибіркою зображень. Оскільки це правило еквівалентне мінімізації дисперсії в класах, то можна очікувати, що вказана процедура буде мати кластеризаційні властивості.

Нехай задано вибірку векторів зображень утворює M класів, для яких відомі щільності розподілу $p(\bar{x}/\Omega_1), p(\bar{x}/\Omega_2), \dots, p(\bar{x}/\Omega_m)$.

Згідно з (6.23) ентропія i -го класу визначиться виразом:

$$H_i = - \int_{\bar{x}} p(\bar{x}/\Omega_i) \ln p(\bar{x}/\Omega_i) d\bar{x}, \quad (6.24)$$

де інтегрування здійснюється по простору зображень.

Прийmemo також для спрощення викладок такі припущення. Нехай щільності розподілу зображень в класах описуються нормальним законом, тобто $p(\bar{x}/\Omega_i) \sim N(\bar{m}_i, \Sigma_i)$, де \bar{m}_i і Σ_i – вектор математичного сподівання і коваріаційна матриця, відповідно. Крім того, будемо вважати коваріаційні матриці Σ_i , що описують статистичні характеристики всіх класів, ідентичними. Згідно з припущенням про рівність коваріаційних матриць $\Sigma_1 = \Sigma_2 = \dots = \Sigma_M = \Sigma$ отримаємо такий вираз для щільності нормального розподілу зображень i -го класу:

$$p(\bar{x}/\Omega_i) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2} (\bar{x} - \bar{m}_i)^T \Sigma^{-1} (\bar{x} - \bar{m}_i)\right]. \quad (6.25)$$

Коваріаційна матриця Σ визначається характеристичними числами і власними векторами. Частина власних векторів описує властивості зображень меншою мірою, ніж інші вектори, тому ними можна знехтувати. Звідси впливає процедура вибору ознак, яка передбачає використання найважливіших власних векторів як координатних векторів (векторів-ознак) нового ознакового простору. Вибір векторів-ознак за принципом мінімізації ентропії полягає в знаходженні матриці лінійного перетворення A , яка переводить вибіркові вектори зображень \bar{x} в нові вектори меншої розмірності \bar{y} таким чином, що ентропія зображень в класах мінімізується:

$$\bar{y} = A\bar{x}, \quad (6.26)$$

де \bar{x} – вектор розмірності n ,

\bar{y} – вектор зображення розмірності m , причому $m < n$,
 A – матриця розмірності $m \times n$:

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_m^T \end{bmatrix}. \quad (6.27)$$

Рядками матриці A слугують m вибраних векторів-ознак. В новому просторі ознак вектор математичного сподівання \bar{m}_i^* для зображень y , згідно з (6.27), дорівнює:

$$\bar{m}_i^* = A\bar{m}_i, \quad (6.28)$$

Позначимо вектор $\bar{x} - \bar{m}_i$ з виразу (6.25) через \bar{z} : $\bar{z} = \bar{x} - \bar{m}_i$, тоді, згідно з (6.26),

$$\bar{y} - \bar{m}_i^* = A(\bar{x} - \bar{m}_i) = A\bar{z}, \quad (6.29)$$

Звідси коваріаційна матриця для зображень в новому просторі дорівнює:

$$\Sigma_i^* = E\{(\bar{y} - \bar{m}_i^*)(\bar{y} - \bar{m}_i^*)^T\} = AE\{\bar{z}\bar{z}^T\}A^T = A\Sigma A^T, \quad (6.30)$$

оскільки $E\{\bar{z} \cdot \bar{z}^T\} = E\{(\bar{x} - \bar{m}_i)(\bar{x} - \bar{m}_i)^T\} = \Sigma$.

Щільність розподілу для зображень в новому просторі ознак, згідно з (6.28) і (6.30), визначиться, як

$$p(\bar{y}/\Omega_i) = \frac{1}{\sqrt[2]{(2\pi)^m |A\Sigma A^{-1}|^{\frac{1}{2}}}} \exp\left[-\frac{1}{2}(\bar{y} - \bar{m}_i^*)^T (A\Sigma A^T)^{-1} (\bar{y} - \bar{m}_i^*)\right]. \quad (6.31)$$

Ентропія зображень в трансформованому просторі дорівнює:

$$H_i^* = -\int_{\bar{y}} p(\bar{y}/\Omega_i) \ln p(\bar{y}/\Omega_i) d\bar{y}. \quad (6.32)$$

Можна довести, що функція ентропії (6.32) у разі підстановки в неї щільності розподілу (6.31) набуває мінімального значення за такої умови: матриця перетворення A утворена з m нормованих власних векторів, що відповідають найменшим власним значенням коваріаційної матриці Σ .

Таким чином, виділення інформативних ознак за методом мінімізації ентропії полягає в визначенні власних векторів і власних значень

коваріаційної матриці Σ за навчальною вибіркою, і виборі як ознак m власних векторів, що відповідають найменшим власним значенням.

Приклад 6.2. Нехай в тривимірному просторі ознак задано вибірку векторів, що належать двом класам : $\Omega_1 = \{\bar{x}_1^1, \bar{x}_2^1, \bar{x}_3^1, \bar{x}_4^1\}$ і $\Omega_2 = \{\bar{x}_1^2, \bar{x}_2^2, \bar{x}_3^2, \bar{x}_4^2\}$, причому $\bar{x}_1^1 = (0,0,0)^T$, $\bar{x}_2^1 = (1,0,0)^T$, $\bar{x}_3^1 = (1,0,1)^T$, $\bar{x}_4^1 = (1,1,0)^T$, $\bar{x}_1^2 = (0,0,1)^T$, $\bar{x}_2^2 = (0,1,0)^T$, $\bar{x}_3^2 = (0,1,1)^T$, $\bar{x}_4^2 = (0,1,1)^T$. На рис. 6.8 показано розміщення цих вибірових векторів. Потрібно понизити розмірність простору ознак (вибравши найбільш інформативні ознаки) за допомогою перетворення, що мінімізує ентропію.

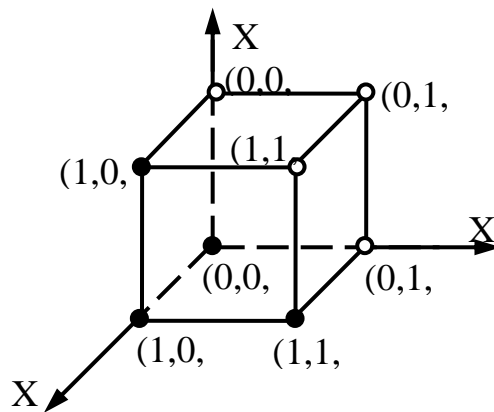


Рисунок 6.8 – Зображення вибірових векторів для прикладу 6.2

Розв’язування: насамперед визначимо за значеннями вибірових векторів оцінки вектора математичного сподівання та коваріаційної матриці, використавши відповідно формули (4.47) і (4.49) розділу 4:

$$\vec{m}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \bar{x}_j^i$$

і

$$\Sigma_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \bar{x}_j^i (\bar{x}_j^i)^T - \vec{m}_i \vec{m}_i^T.$$

Знаходимо вектори математичного сподівання для 1-го і 2-го класів :

$$\vec{m}_1 = \frac{1}{4} \begin{pmatrix} 0+1+1+1 \\ 0+0+0+1 \\ 0+0+1+0 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix}; \quad \vec{m}_2 = \frac{1}{4} \begin{pmatrix} 0+0+0+1 \\ 0+1+1+1 \\ 1+0+1+1 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 1 \\ 3 \\ 3 \end{pmatrix};$$

Знаходимо коваріаційну матрицю Σ_1 :

$$\bar{x}_1^1 * (\bar{x}_1^1)^T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} [000] = \begin{bmatrix} 000 \\ 000 \\ 000 \end{bmatrix}; \quad \bar{x}_2^1 * (\bar{x}_2^1)^T = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} [100] = \begin{bmatrix} 100 \\ 000 \\ 000 \end{bmatrix};$$

$$\bar{x}_3^1 * (\bar{x}_3^1)^T = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} [101] = \begin{bmatrix} 101 \\ 000 \\ 101 \end{bmatrix}; \quad \bar{x}_4^1 * (\bar{x}_4^1)^T = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} [110] = \begin{bmatrix} 110 \\ 110 \\ 000 \end{bmatrix}.$$

$$\bar{m}_1(\bar{m}_1)^T = \frac{1}{4} * \frac{1}{4} \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} [311] = \frac{1}{16} \begin{bmatrix} 933 \\ 311 \\ 311 \end{bmatrix}; \quad \Sigma_1 = \frac{1}{4} \begin{bmatrix} 311 \\ 110 \\ 101 \end{bmatrix} - \frac{1}{16} \begin{bmatrix} 933 \\ 311 \\ 311 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 311 \\ 13-1 \\ 1-13 \end{bmatrix}$$

Аналогічно знаходимо коваріаційну матрицю Σ_2 і отримуємо, що $\Sigma_2 = \Sigma_1 = \Sigma$.

Знайдемо власні значення коваріаційної матриці Σ , розв'язавши одним із чисельних методів характеристичне рівняння матриці

$$\det(\Sigma - \lambda I) = |\Sigma - \lambda I| = 0, \quad (6.33)$$

де I – одинична матриця,
 λ – власне значення.

Власні значення матриці Σ дорівнюють: $\lambda_1 = \frac{1}{16}$, $\lambda_2 = \lambda_3 = \frac{1}{4}$.

Нормовані власні вектори, що відповідають характеристичним числам $\lambda_1, \lambda_2, \lambda_3$, мають вигляд:

$$\bar{e}_1 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}, \quad \bar{e}_2 = \frac{1}{\sqrt{6}} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \quad \bar{e}_3 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}.$$

Підстановка транспонованих власних векторів в матрицю перетворення дає такий результат

$$A = \begin{bmatrix} \bar{e}_1^T \\ \bar{e}_2^T \end{bmatrix} = \begin{bmatrix} -1/\sqrt{3} & -1/\sqrt{3} & -1/\sqrt{3} \\ 2/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} \end{bmatrix}$$

Вектори зображень в новому просторі ознак, отримані внаслідок перетворення, мають такі значення:

$$\Omega_1 : \{ \bar{y}_1^1 = (0,0)^T; \bar{y}_2^1 = (\frac{1}{\sqrt{3}}, \frac{2}{\sqrt{6}})^T; \bar{y}_3^1 = (0, \frac{3}{\sqrt{6}})^T; \bar{y}_4^1 = (0, \frac{3}{\sqrt{6}})^T \},$$

$$\Omega_2 : \{ \bar{y}_1^2 = (-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{6}})^T; \bar{y}_2^2 = (-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{6}})^T; \bar{y}_3^2 = (-\frac{2}{\sqrt{3}}, \frac{2}{\sqrt{6}})^T; \bar{y}_4^2 = (-\frac{1}{\sqrt{3}}, \frac{4}{\sqrt{6}})^T \}$$

На рис. 6.9 подано вектори зображень в перетвореному просторі ознак.

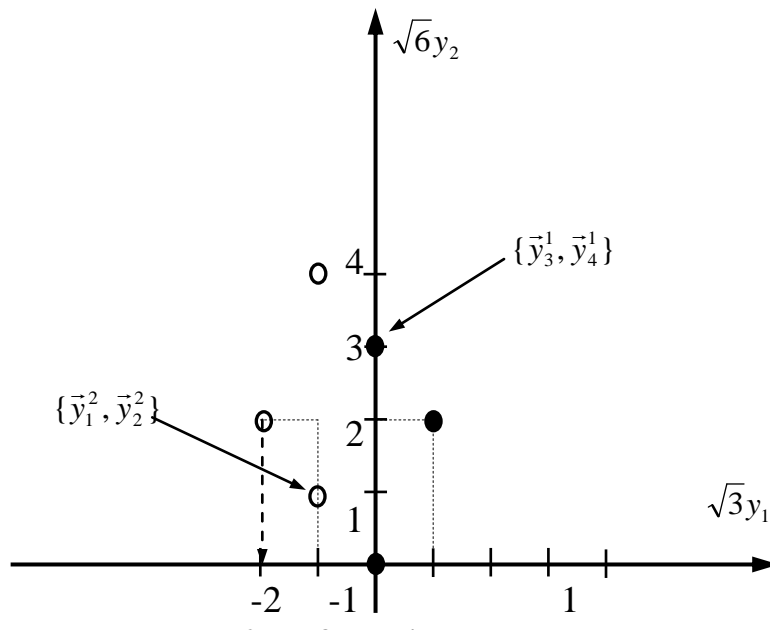


Рисунок 6.9 – Зображення в перетвореному просторі ознак

6.6 Вибір ознак за методом розкладу Карунена-Лоева

В цьому підрозділі розглядається вибірка векторів

В цьому підрозділі розглядається вибірка векторів $X = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_q, \dots, \bar{x}_Q\}$, що належить до одного класу (є тільки одна щільність розподілу випадкових векторів). Для цього випадку немає сенсу розв'язувати задачу розпізнавання, можна тільки розглянути, наскільки точно можна подати задані вибіркові об'єкти в просторі ознак меншої розмірності. Це означає, що для векторів, заданих n координатами (ознаками) $\bar{x}_q = (x_1, x_2, \dots, x_n)$, потрібно вибрати m координат ($m < n$) таким чином, щоб зображення об'єктів в новому просторі з меншою кількістю ознак залишалось якісним згідно з вибраним критерієм. Якщо з допомогою невеликої кількості ознак можна точно зобразити об'єкти, то такі ознаки

вважаються ефективними. Хоч ця задача і не пов'язана безпосередньо з розпізнаванням образів, знання характерних ознак окремих класів допомагає відрізнити їх один від одного.

На рис. 6.10 показано приклад розкладу деякого вектора \vec{x} в тривимірному просторі ознак за координатами (ознаками) x_1, x_2, x_3 .

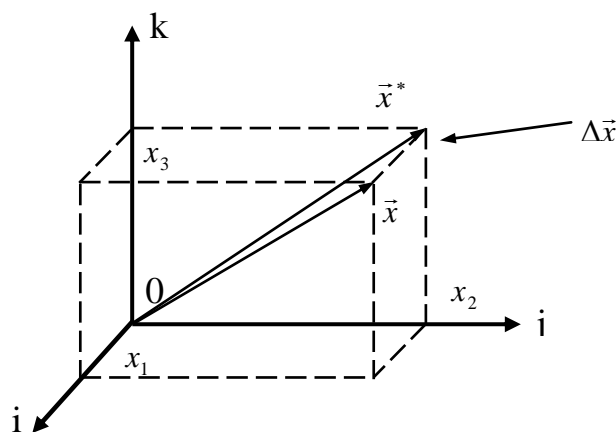


Рисунок 6.10 – Приклад розкладу вектора X в тривимірному просторі ознак (x_1, x_2, x_3) і в двовимірному просторі ознак (y_2, y_3)

На цьому самому рисунку показано його проекцію \vec{x}^* на площину x_2Ox_3 , яка є найближчою серед усіх проекцій. Можна тоді сказати, що вектор \vec{y} є наближеним значенням вектора \vec{x} , розкладеного по двох координатах (ознаках) x_2 і x_3 . З наведеного прикладу можна зробити висновок, що вибір меншої кількості ознак для опису об'єкта можна подати як розклад вектора його зображення по меншій кількості координат. В розкладі Карунена-Лоева за критерій якості зображення вибрано мінімум середньоквадратичної відстані між вектором \vec{x} початкового простору і вектором \vec{x}^* простору з меншою кількістю ознак:

$$\Delta \vec{x} = \|\vec{x} - \vec{x}^*\|^2 = \min \quad (6.34)$$

Ще одне обмеження, прийняте в розкладі Карунена-Лоева, пов'язано з тим, що шукають тільки ті ознаки, які можна отримати лінійним перетворенням простору. З урахуванням попередніх міркувань розглянемо математичне тлумачення задачі зниження розмірності простору ознак шляхом використання дискретного розкладу Карунена-Лоева.

Узагальнюючи формулу (6.12) розкладу випадкового вектора \vec{x} в n -вимірному просторі ознак, ми можемо подати його у вигляді:

$$\vec{x} = \sum_{i=1}^n y_i \vec{\Phi}_i = \Phi \vec{y}, \quad (6.35)$$

де $\Phi = [\vec{\Phi}_1, \vec{\Phi}_2, \dots, \vec{\Phi}_n]$, $\vec{y} = [y_1, y_2, \dots, y_n]^T$

Матриця Φ є детермінованою і містить n незалежних векторів–стовпців, звідки випливає, що детермінант матриці не дорівнює нулю:

$$\det \Phi = |\Phi| \neq 0. \quad (6.36)$$

Отже, лінійні комбінації стовпців матриці Φ утворюють n -вимірний простір ознак, що містить \vec{x} . Стовпці матриці Φ називаються базисними векторами. Прийmemo, що вони є ортонормованими, тобто відповідають умові:

$$\vec{\Phi}_i^T \vec{\Phi}_j = \begin{cases} 1, & \text{якщо } i = j \\ 0, & \text{якщо } i \neq j. \end{cases} \quad (6.37)$$

Вектор \vec{y} є зображенням вектора \vec{x} в перетвореному просторі ознак. Якщо умова ортонормованості (6.37) виконана, то компоненти вектора y визначаються таким чином:

$$y_i = \vec{\Phi}_i^T \vec{x}, \quad i = 1, \dots, n. \quad (6.38)$$

Таким чином, вектор \vec{y} є ортогональним перетворенням випадкового вектора \vec{x} і також є випадковим вектором. Кожна з його компонент є ознакою, яка дає свій внесок в подання спостережуваного вектора \vec{x} .

Припустимо, що для визначення оцінки вектора \vec{x} вектором $\vec{x}^*(m)$ ми використовуємо m компонент вектора \vec{y} , а інші компоненти (від $m+1$ до n) ще не знаємо і хочемо визначити, якими вони мають бути, щоб мінімізувати помилку в поданні вектора \vec{x} (назвемо ці компоненти b_i). Тоді сам вектор \vec{x} і його оцінку $\vec{x}^*(m)$ можна записати, як:

$$\vec{x} = \sum_{i=1}^m y_i \vec{\Phi}_i + \sum_{i=m+1}^n y_i \vec{\Phi}_i, \quad (6.39)$$

$$\vec{x}^*(m) = \sum_{i=1}^m y_i \vec{\Phi}_i + \sum_{i=m+1}^n b_i \vec{\Phi}_i. \quad (6.40)$$

Якщо обчислюється тільки m перших компонент вектора \vec{y} , тобто використовуються не всі ознаки, то вектор \vec{x} подається з помилкою, що визначається виразом:

$$\Delta\bar{x}(m) = \bar{x} - \bar{x}_1^*(m) = \sum_{i=m+1}^n (y_i - b_i) \bar{\Phi}_i. \quad (6.41)$$

За критерій якості для вимірювання ефективності підмножини, що складається із m ознак, використовуємо середню величину квадрата відхилення $\Delta\bar{x}(m)$:

$$\tilde{\varepsilon}^2(m) = E\{\|\Delta\bar{x}(m)\|^2\} = \sum_{i=m+1}^n E\{(y_i - b_i)^2\}. \quad (6.42)$$

Кожному набору базисних векторів і значень констант відповідає деяке значення $\tilde{\varepsilon}^2(m)$. Виберемо їх таким чином, щоб мінімізувати середньоквадратичну похибку $\tilde{\varepsilon}^2(m)$.

Оптимальний вибір констант b_i здійснюється таким чином:

$$b_i = E\{y_i\} = E\{\bar{\Phi}_i^T \bar{x}\} = \bar{\Phi}_i^T E\{\bar{x}\}. \quad (6.43)$$

Іншими словами, ті компоненти y_i , що не вимірюються, ми маємо замінити їхніми математичними сподіваннями. Цей висновок легко отримати, виконавши мінімізацію $\tilde{\varepsilon}^2(m)$ за b_i :

$$\frac{\partial}{\partial b_i} E\{(y_i - b_i)^2\} = -2[E\{y_i\} - b_i] = 0, \quad (6.44)$$

звідки $b_i = E\{y_i\}$.

Підставляючи (6.43) в (6.42), отримаємо:

$$\begin{aligned} \tilde{\varepsilon}^2(m) &= \sum_{i=m+1}^n E[(y_i - E\{y_i\})^2] = \sum_{i=m+1}^n E[(\bar{\Phi}_i^T \bar{x} - E\{\bar{\Phi}_i^T \bar{x}\}) \times (\bar{\Phi}_i^T \bar{x} - E\{\bar{\Phi}_i^T \bar{x}\})^T] = \\ &= \sum_{i=m+1}^n \bar{\Phi}_i^T E[\bar{x} - E\{\bar{x}\}] \times [(\bar{x} - E\{\bar{x}\})^T] \bar{\Phi}_i = \sum_{i=m+1}^n \bar{\Phi}_i^T \Sigma_x \bar{\Phi}_i, \end{aligned}$$

де Σ_x – коваріаційна матриця випадкового вектора \bar{x} . Оптимальний вибір матриці Φ відповідає умові:

$$\Sigma_x \bar{\Phi}_i = \lambda_i \bar{\Phi}_i, \quad (6.45)$$

тобто оптимальними базисними векторами є власні вектори коваріаційної матриці Σ_x . Підставляючи (6.45) в (6.44), отримуємо вираз для мінімальної середньоквадратичної похибки:

$$\tilde{\varepsilon}^2(m)_{opt} = \sum_{i=m+1}^n \lambda_i. \quad (6.46)$$

Вираз (6.46) дає підставу зробити висновок, що дискретний варіант розкладу Карунена-Лоева є розкладом за власними векторами коваріаційної матриці.

Коефіцієнти y_1, y_2, \dots, y_n цього розкладу вважаються ознаками, що описують вектор зображення об'єкта \vec{x} . Цим ознакам притаманні такі, варті уваги, властивості:

1. Ефективність кожної ознаки для подання вектора \vec{x} визначається відповідним власним значенням. Наприклад, якщо деяка ознака (коефіцієнт y_i) виключається з розкладу, то середньоквадратична похибка збільшується на λ_i . Тому під час зменшення кількості ознак насамперед виключається ознака з найменшим власним значенням.

2. Ознаки, про які йде мова, взаємно некорельовані, тобто коваріаційна матриця випадкового вектора \vec{y} є діагональною D_λ :

$$\Sigma_y^- = \Phi^T \Sigma_x \Phi = \begin{bmatrix} \lambda_1 & \dots & 0 \\ \vdots & \lambda_2 & \vdots \\ 0 & \dots & \lambda_n \end{bmatrix} = D_\lambda. \quad (6.47)$$

В окремому випадку, коли вектор \vec{x} має нормальний закон розподілу, ознаки y_i взаємно незалежні.

3. Власні вектори коваріаційної матриці Σ_x дають найменше значення середньоквадратичної похибки $\tilde{\varepsilon}^2(m)$ серед всіх ортонормованих базисних векторів.

Особливості вибору ефективних ознак за допомогою розкладу Карунена-Лоева розглянемо на такому прикладі.

Приклад 6.3. Задано дві множини вибіркових векторів, зображених на рис.7.5, а) і б). Потрібно зменшити розмірність простору ознак, застосувавши для цього дискретний розклад Карунена-Лоева.

Розв'язування: Запишемо вектори для рис.6.11а) за їх координатами:

$$\vec{x}_1 = (1,1)^T, \vec{x}_2 = (2,2)^T, \vec{x}_3 = (-1,1)^T, \vec{x}_4 = (-2,-2)^T,$$

а для рис. 6.11, б)

$$\vec{x}_1 = (-1,1)^T, \vec{x}_2 = (2,2)^T, \vec{x}_3 = (1,-1)^T, \vec{x}_4 = (-2,-2)^T.,$$

після чого обчислимо коваріаційні матриці Σ_a і Σ_b для розподілів векторів на рис. 6.11, а) і 6.11, б), відповідно:

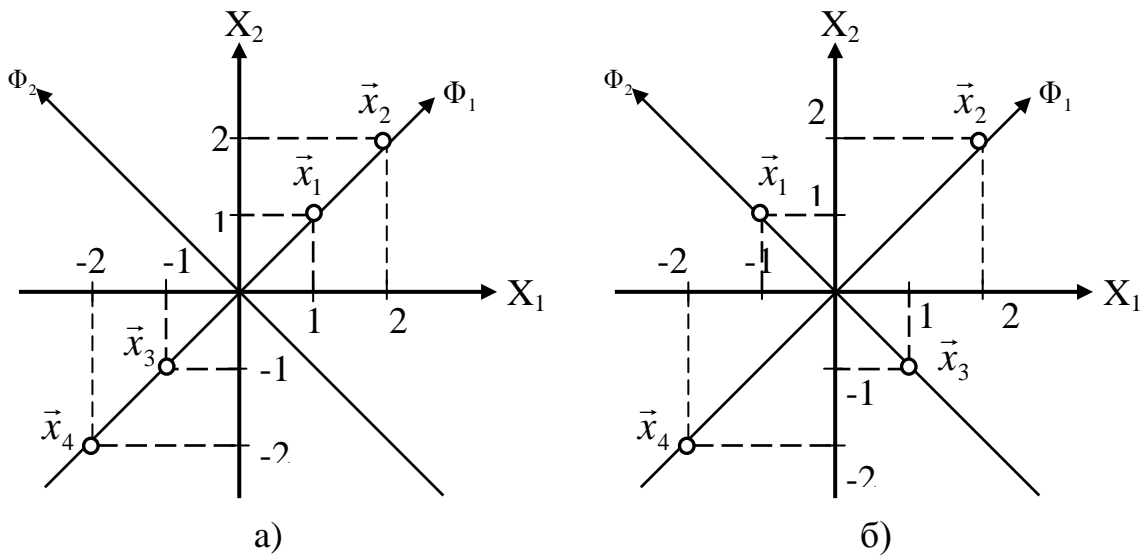


Рисунок 6.11 – Зображення вибірових точок для прикладу 6.3

$$\Sigma_a = \frac{1}{4} \sum_{i=1}^4 \vec{x}_i \vec{x}_i^T = \frac{1}{4} \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} [11] + \begin{bmatrix} 2 \\ 2 \end{bmatrix} [22] + \begin{bmatrix} -1 \\ -1 \end{bmatrix} [-1-1] + \begin{bmatrix} -2 \\ -2 \end{bmatrix} [-2-2] \right\} = \begin{bmatrix} \frac{10}{4} & \frac{10}{4} \\ \frac{10}{4} & \frac{10}{4} \end{bmatrix}; \quad (6.48)$$

$$\Sigma_b = \frac{1}{4} \sum_{i=1}^4 \vec{x}_i \vec{x}_i^T = \frac{1}{4} \left\{ \begin{bmatrix} -1 \\ +1 \end{bmatrix} [-1+1] + \begin{bmatrix} 2 \\ 2 \end{bmatrix} [22] + \begin{bmatrix} +1 \\ -1 \end{bmatrix} [+1-1] + \begin{bmatrix} -2 \\ -2 \end{bmatrix} [-2-2] \right\} = \begin{bmatrix} \frac{10}{4} & \frac{6}{4} \\ \frac{6}{4} & \frac{10}{4} \end{bmatrix}; \quad (6.49)$$

Після цього за допомогою відомих чисельних методів знаходимо власні значення і власні вектори матриці Σ_a і Σ_b :

$$\lambda_{1a} = 5, \lambda_{2a} = 0, \vec{\Phi}_{1a} = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}, \vec{\Phi}_{2a} = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}, \quad (6.50)$$

$$\lambda_{1b} = 4, \lambda_{2b} = 1, \vec{\Phi}_{1b} = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}, \vec{\Phi}_{2b} = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}, \quad (6.51)$$

Базисні вектори $\vec{\Phi}_1$ і $\vec{\Phi}_2$ обернені на кут 45° , як зображено на рис. 6.11, а) і б).

Для вибірки даних рис. 6.11, а) власне значення $\lambda_{2a} = 0$, тому двовимірний простір ознак $X_1 O X_2$ можна перетворити на двовимірний, виключивши вектор $\vec{\Phi}_2$ з розкладу Карунена-Лоева. Оскільки $\lambda_{2a} = 0$, то середньоквадратична похибка буде дорівнювати нулю. Це можна

спостерігати на рис. 6.11, а), з якого видно, що всі чотири об'єкти можна безпомилково виразити через перший базисний вектор $\vec{\Phi}_1$.

Для вибірки даних на рис. 6.11, б) власне значення $\lambda_{2b}=1$, тому у разі виключення базисного вектора $\vec{\Phi}_2$ с згідно з (6.46), помилка $\tilde{\varepsilon}^2(m)$ дорівнює:

$$\xi_1^2 = \sum_{i=m+1}^n \lambda_i = \lambda_{2b} = 1 \quad (6.52)$$

Такий самий результат можна отримати з геометричних міркувань: вектори \vec{x}_2 і \vec{x}_4 розміщені на векторі $\vec{\Phi}_1$ і тому виражаються ним без помилки, а вектори \vec{x}_1 і \vec{x}_3 розміщені на відстані $\sqrt{2}$ від вектора $\vec{\Phi}_1$. Тому середньоквадратична похибка дорівнює $(0^2 + 0^2 + \sqrt{2}^2 + \sqrt{2}^2)/4 = 1$, що відповідає виразу (6.52).

6.7 Розклад Карунена-Лоева для випадкових процесів

Використаємо отримані в розділі 6.4 результати для задачі подання випадкових процесів. Розглянемо додатково також деякі характерні властивості розкладу Карунена-Лоева, специфічні для випадкових процесів.

Випадковим процесом називається функція часу $X(t)$, $t \in T$, значення якої за будь-якого фіксованого $t = t_0$ є випадковою величиною $x(t_0)$. Приклад випадкового процесу показано на рис. 6.12.

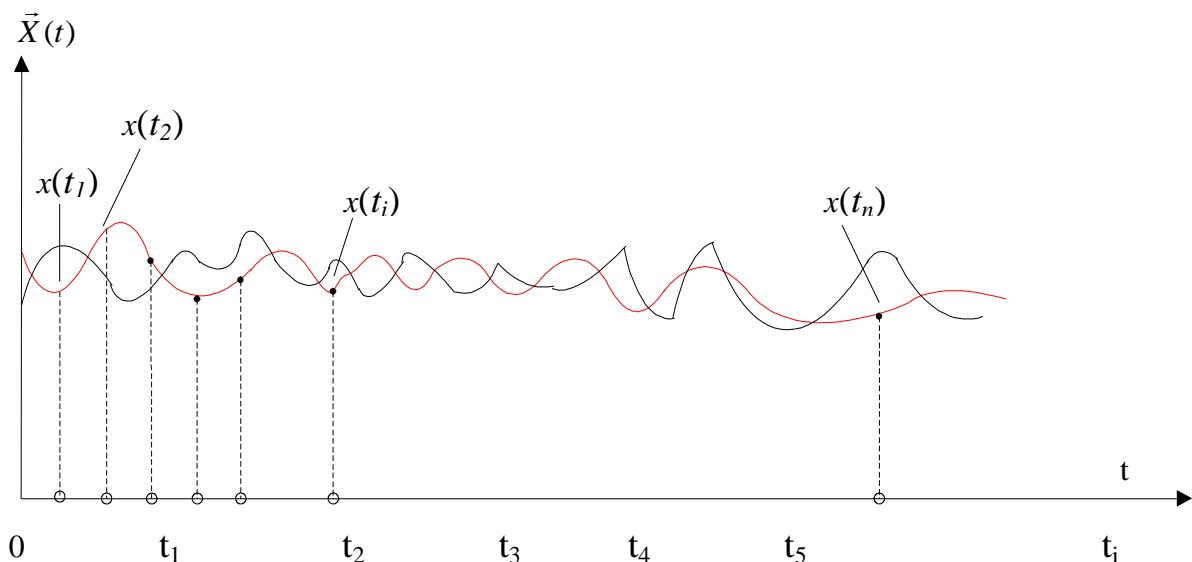


Рисунок 6.12 – Приклади реалізацій випадкового процесу

Якщо в деякому інтервалі $0 \leq t \leq t_n$ через однакові проміжки часу виконати вибірку значень $x(t_i)$, то результат можна подати в такому векторному вигляді:

$$\vec{x}(t) = \begin{bmatrix} x(t_1) \\ x(t_2) \\ \vdots \\ x(t_i) \\ \vdots \\ x(t_n) \end{bmatrix}, \quad (6.53)$$

де n – кількість спостережень функції $x(t)$, зроблених на інтервалі $[0, t_n]$.

Такий випадковий процес можна подати лінійною комбінацією базисних функцій:

$$\vec{x}(t) = \sum_{i=1}^n y_i y_i(t), \quad (6.54)$$

де базисні функції $y_i(t)$ є детермінованими функціями часу, а коефіцієнти y_i – випадковими величинами. Для утворення повної системи функцій потрібно мати нескінченну кількість базисних функцій, тому в виразі (6.54) верхня границя підсумовування має дорівнювати ∞ . Умова ортонормованості базисних функцій $y_i(t)$ має вигляд:

$$\int_0^t y_i(t) y_j^*(t) dt = \delta_{ij}, \quad (6.55)$$

де $y_j^*(t)$ і $y_i(t)$ – комплексно-спряжені функції.

Якщо $y_i(t)$ – дійсна функція, то $y_j^*(t) = y_j(t)$.

Обернена операція обчислення коефіцієнтів y_i за $\vec{x}(t)$ має вигляд:

$$y_i = \int_0^{t_n} \vec{x}(t) y_i^*(t) dt. \quad (6.56)$$

Математичне сподівання, кореляційна та коваріаційна функції випадкового процесу визначаються таким чином:

$$m(t) = E\{\vec{x}(t)\} \quad (6.57)$$

$$R(t, \tau) = E\{\vec{x}(t) \vec{x}^*(\tau)\}, \quad (6.58)$$

$$\Sigma(t, \tau) = E\{ \{\bar{x}(t) - m(t)\} \{\bar{x}(\tau) - m(\tau)\}^* \}. \quad (6.59)$$

Оскільки розклад Карунена-Лоева застосовується до вибірки векторів, які відносяться до одного класу (утворюють один розподіл ймовірностей), то для спрощення викладок можна вважати, що $m(t) = 0$ для $0 \leq t \leq t_n$. За умови того, що базисні функції $y_i(t)$ є власними функціями кореляційної функції $y_i(t)$, вони задовольняють таке інтегральне рівняння:

$$\int_0^{t_n} R(t, \tau) y_i(\tau) d\tau = \lambda_i y_i(t), \quad i = 1, 2, \dots, \quad (6.60)$$

де λ_i – власні значення кореляційної функції $R(t, \tau)$.

Рівняння 6.57, 6.58 і 6.59 для випадкових процесів мають такий самий вигляд, як і рівняння для випадкових векторів.

Труднощі використання розкладу Карунена-Лоева для неперервних випадкових процесів полягають в тому, що для отримання власних векторів і власних значень необхідно розв'язувати інтегральне рівняння (6.60). В загальному випадку розв'язок цього інтегрального рівняння в явному вигляді не вдається. Тому для його отримання чисельними методами потрібно звернутися до дискретного варіанта, тобто взяти вибірккові значення випадкового процесу, обчислити кореляційну матрицю і знайти її власні значення та власні вектори.

Дискретним аналогом рівняння (6.60) є рівняння виду:

$$S \vec{\Phi}_i = \lambda_i \vec{\Phi}_i, \quad (6.61)$$

де кореляційна матриця обчислюється за виразом:

$$S = \begin{bmatrix} R(t_1, t_1) \cdots R(t_1, t_n) \\ \vdots \\ R(t_n, t_1) \cdots R(t_n, t_n) \end{bmatrix} = \begin{bmatrix} E\{x(t_1)x^*(t_1)\} \cdots E\{x(t_1)x^*(t_n)\} \\ \vdots \\ E\{x(t_n)x^*(t_1)\} \cdots E\{x(t_n)x^*(t_n)\} \end{bmatrix} \quad (6.62)$$

$$\vec{\Phi}_i = (y_i(t_1), y_i(t_2), \dots, y_i(t_n))^T \quad (6.63)$$

є власним вектором матриці перетворення Φ :

$$\vec{y} = \Phi^T \vec{x}, \quad (6.64)$$

$$\Phi = [\vec{\Phi}_1, \vec{\Phi}_2, \dots, \vec{\Phi}_m]. \quad (6.65)$$

Вираз (6.64) є дискретним аналогом розкладу Карунена-Лоева (6.56) для неперервного випадкового процесу $\bar{x}(t)$.

Оскільки матриця кореляції S має розмірність $n \times n$, то в розкладі Карунена-Лоева за такого підходу буде використовуватися не нескінченне число, а тільки n власних значень і власних векторів.

Мінімізація середньоквадратичної помилки здійснюється за процедурою, аналогічною для дискретного випадку. А саме, можна показати, що середньоквадратична похибка подання випадкового процесу розкладом Карунена-Лоева дорівнює:

$$\hat{\varepsilon}^2 = \sum_{i=m+1}^{\infty} E\{y_i y_i^*\} = \sum_{i=m+1}^{\infty} \lambda_i. \quad (6.66)$$

Звідси можна зробити висновок, що умови оптимальності розкладу Карунена-Лоева виконується, якщо за стовпці матриці перетворення Φ вибираються m нормованих власних векторів, що відповідають найбільшим власним значенням кореляційної матриці R .

Враховуючи наведені вище міркування, алгоритм вибору ефективного набору ознак для зображення випадкового процесу відповідно до розкладу Карунена-Лоева складається з таких кроків:

1. За часовими вибірками реалізацій випадкового процесу, що входять в навчальну вибірку, будують кореляційну матрицю S згідно з виразом (6.62).

2. Визначають власні значення і власні вектори кореляційної матриці S . Виконується нормування власних векторів.

3. Використовують m власних векторів, що відповідають найбільшим власним значенням кореляційної матриці S , для формування матриці перетворення Φ за формулою (6.63).

4. За формулою (6.65) обчислюють коефіцієнти розкладу, які задають зображення з меншою розмірністю опису.

Для того, щоб розклад Карунена-Лоева давав оптимальні результати, необхідне виконання умови $E\{\bar{y}\} = 0$, що рівносильно умові $E\{\bar{x}\} = 0$. Остання умова виконується автоматично, якщо окремі класи характеризуються нульовими середніми сподіваннями. Припущення такого типу характерні для більшості статистичних методів аналізу. Успіх використання будь-якого методу залежить тільки від того, наскільки добре дані, що підлягають аналізу, відповідають основним передбаченням, прийнятим під час розробки відповідного статистичного методу.

Контрольні питання та завдання

1. Які типи ознак використовуються біологічними системами розпізнавання?
2. Назвіть приклади фізичних і структурних ознак.

3. Які типи ознак використовують системи розпізнавання зі штучним інтелектом?
4. Який критерій якості застосовують в процедурі зменшення розмірності простору ознак?
5. Запишіть формулу відстані між точкою \vec{x} і множиною точок $\{\vec{a}^k\}$ в N -вимірному просторі ознак.
6. Запишіть формулу внутрішньомножинної відстані в класі $\Omega_i = \{\vec{a}_1, \vec{a}_1, \dots, \vec{a}_k, \dots, \vec{a}_K\}$.
7. Як можна виразити внутрішньомножинну відстань через незміщену оцінку вибіркової дисперсії?
8. Назвіть основні способи зниження розмірності простору ознак. Вкажіть, на якому принципі вони базуються.
9. Які міркування лежать в основі зниження простору розмірності ознак за методом PCA?
10. Яким чином здійснюється обчислення головних компонент в методі PCA?
11. Які умови використовуються в методі PCA під час реалізації ортогонального перетворення в нову систему координат?
12. Чи достатньо таких параметрів, як математичне сподівання і дисперсія для опису форми кластера у випадку багатовимірного простору ознак ймовірнісного типу?
13. Для чого використовується коваріаційна матриця під час аналізу множини векторів випадкового типу?
14. Запишіть загальний вигляд формули дисперсії у векторній формі для центрованих величин.
15. Запишіть вираз для розкладання матриці на власні вектори і власні значення.
16. В якому випадку використовують проектування вибірки векторів кластера на менший власний вектор, тобто використовують принцип мінімуму ентропії?
17. Що впливає на якість розподілу класів під час лінійного перетворення простору ознак?
18. Запишіть формулу критерію якості лінійного перетворення простору ознак.
19. Опишіть послідовність дій під час виділення інформативних ознак за методом мінімізації ентропії.
20. Як використовується коваріаційна матриця для дискретного випадку в здійсненні оптимального варіанта розкладу Карунена-Лоева?
21. Опишіть алгоритм вибору ефективного набору ознак для зображення випадкового процесу згідно з розкладом Карунена-Лоева.

РОЗДІЛ 7 СИНТАКСИЧНЕ РОЗПІЗНАВАННЯ ОБРАЗІВ

7.1 Основні ідеї синтаксичного підходу до розпізнавання

Попередні розділи були присвячені дискримінантному математичному підходу до розпізнавання образів, який можна трактувати з позицій теорії прийняття рішень. В цьому підході об'єкти характеризуються наборами чисел, що є результатами деякої множини вимірювань, які характеризують об'єкти, і називаються ознаками. Розпізнавання образів (віднесення кожного об'єкта до деякого класу) в цьому випадку здійснюють за допомогою розбиття простору ознак на області. Перші дослідження з розпізнаванням образів були здебільшого пов'язані з дискримінантним підходом і його застосуваннями.

Інша група математичних методів до розпізнавання образів реалізована в рамках синтаксичного підходу. Часто в літературі такий підхід іноді називають лінгвістичним або іноді структурним. Синтаксичні методи розпізнавання застосовуються (іноді для вирішення окремих підзадач) у різних прикладних областях, зокрема для аналізу фотографій, отриманих за допомогою бульбашкових камер, у матеріалознавстві, у біології і медицині, для виявлення дефектів промислових об'єктів, деталей машин і механізмів, розпізнавання символів, класифікації відбитків пальців, розпізнавання мови, дистанційного обстеження навколишнього середовища, аналізу форми, для розпізнавання (двовимірних) математичних формул та ін. У синтаксичному підході вивчаються задачі розпізнавання образів, в яких важлива інформація, що описує структуру кожного об'єкта, а від процедури розпізнавання потрібно, щоб вона давала можливість не тільки віднести об'єкт до певного класу (класифікувати його), а й описати ті сторони об'єкта, які виключають його віднесення до іншого класу. Типовим прикладом таких задач слугує розпізнавання зображень або, в більш широкому аспекті, аналіз сцен. Розглянуті в цьому класі задачі об'єкти зазвичай складні, і кількість необхідних ознак часто велика. Коли об'єкти складні і кількість можливих описів велика, вважати, що кожний опис визначає клас, незручно, як, наприклад, в задачах ідентифікації портретів, відбитків пальців, в процесі розпізнавання зливої мови або китайських ієрогліфів. У таких випадках розпізнавання може бути проведено лише з використанням опису кожного об'єкта, а не просто за допомогою класифікації. Це робить привабливою ідею опису складного об'єкта у вигляді ієрархічної структури більш простих субобразів.

Для того, щоб подавати ієрархічну (деревоподібну) структурну інформацію, що міститься в кожному образі, тобто описувати образ за допомогою більш простих субобразів, а кожен субобраз знову описувати

ще більш простими субобрами і т. д., був запропонований синтаксичний, або структурний, підхід.

Приклад 7.1. Зображення на рис. 7.1 можна описати ієрархічними структурами, показаними на рис. 7.2.

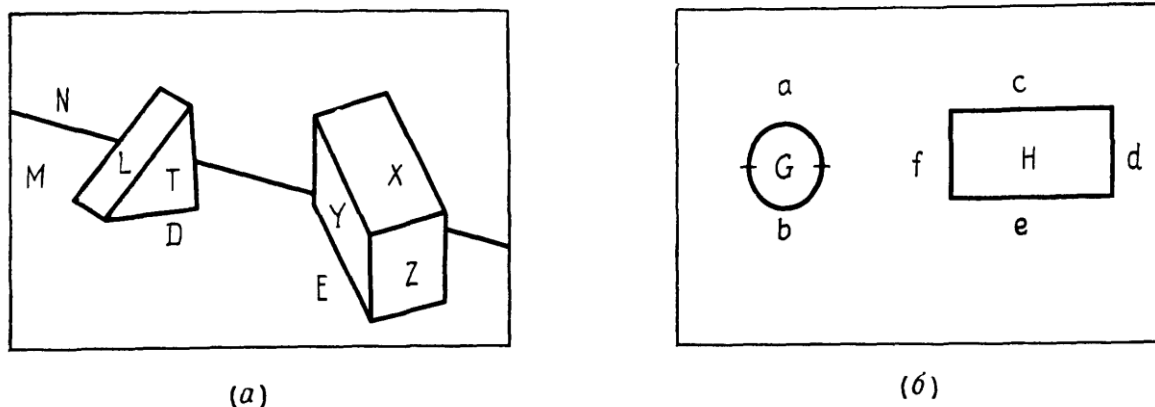


Рисунок 7.1 – Зображення для прикладу 7.1: (а) – сцена А; (б) – зображення F.

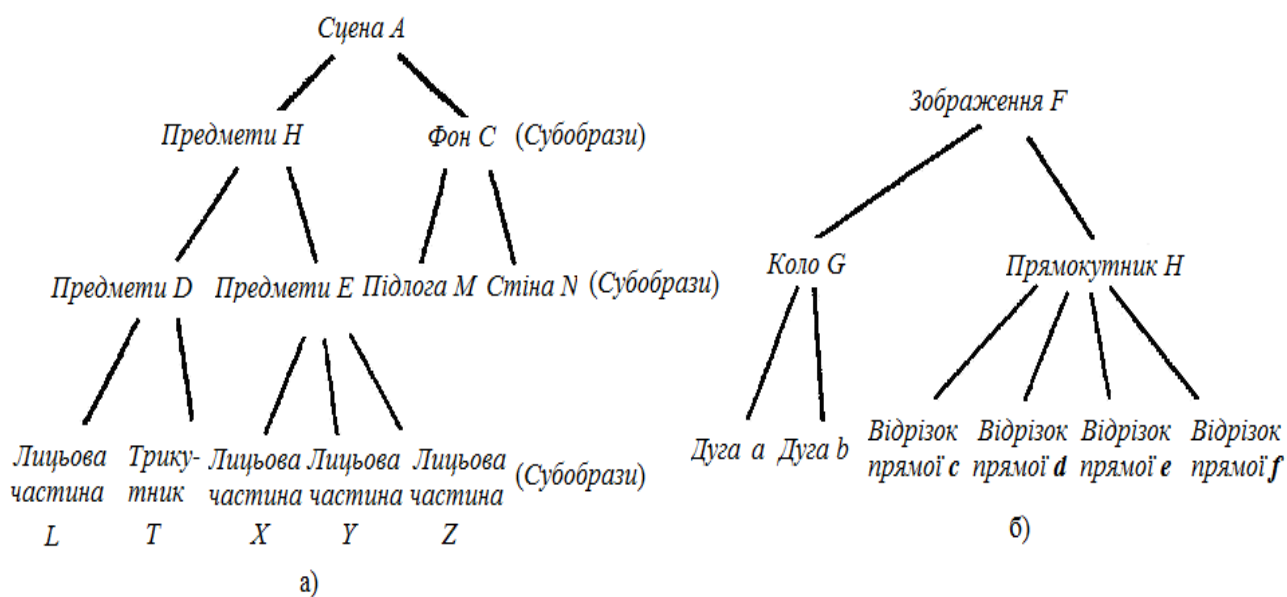


Рисунок 7.2 – Ієрархічний структурний опис сцени А – а) і зображення F – б)

Цей підхід оснований на аналогії між структурою образів (ієрархічною або деревоподібною) і синтаксисом мов. У рамках синтаксичного підходу вважається, що образи будуються із з'єднаних різними способами субобразів, так само як фрази і речення будуються шляхом сполучення слів, а слова складаються з букв. Очевидно, що такий підхід корисний тільки в тому випадку, коли розпізнавання вибраних найпростіших субобразів, які називаються непохідними елементами, легше, ніж самих образів. «Мову», яка забезпечує структурний опис

образів у термінах множини непохідних елементів та операцій композиції цих елементів, іноді називають «мовою опису образів». Правила композиції непохідних елементів зазвичай задають за допомогою так званої граматики мови опису образів. Процес розпізнавання здійснюється після ідентифікації в об'єкті непохідних елементів і складання опису об'єкта. Розпізнавання полягає в синтаксичному аналізі, або граматичному розборі, «речення», що описує цей об'єкт. Ця процедура встановлює, чи є це речення синтаксично (або граматично) правильним відносно заданої граматики. Паралельно синтаксичний аналіз дає деякий структурний опис речення (зазвичай у вигляді дерева).

Синтаксичний підхід до розпізнавання образів дає можливість описувати множину складних об'єктів шляхом використання невеликої множини непохідних елементів і граматичних правил. Як буде видно з подальшого викладу, одним з найпривабливіших аспектів цієї можливості є використання рекурсивної природи граматики.

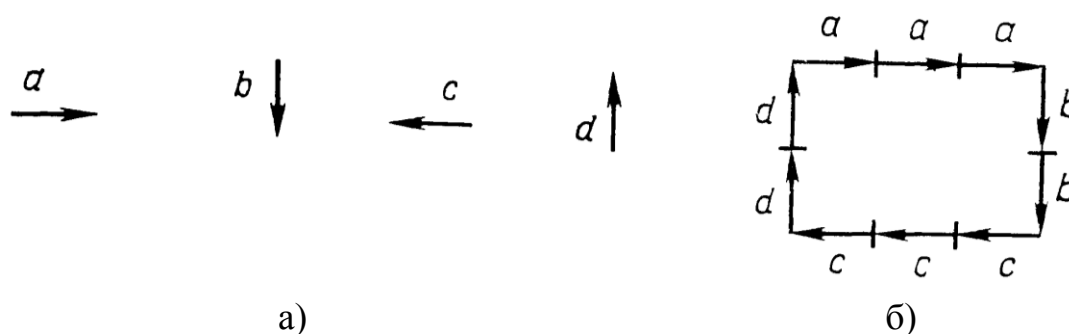


Рисунок 7.3 – Прямокутник і його непохідні елементи

Рекурсивне граматичне правило (правило підстановки) може бути застосовано будь-яку кількість разів, так що виявляється можливим дуже компактно висловити деякі основні структурні характеристики нескінченної кількості речень. Практична корисність такого підходу залежить, звичайно, від здатності розпізнавати непохідні елементи образів та їх взаємні відношення, виражені операціями композиції.

Різні відношення, визначені між субобразами, або операції композиції, звичайно можуть бути виражені логічними і/або математичними операціями. Якщо, наприклад, як єдине відношення (операції композиції) для опису взаємних образів вибрати «конкатенацію», то за непохідних елементів, показаних на рис. 7.3, а), прямокутник, зображений на рис. 7.3, б), буде подано ланцюжком $aaabbcscdd$. Якщо використовувати знак «плюс» для позначення операції «конкатенація початку одного елемента до кінця іншого», то прямокутник на рис. 7.3, б), буде подано реченням $a + a + a + b + b + c + c + c + d + d$.

Деревоподібну структуру цього речення показано на рис. 7.4, а). На рис. 7.4, б) показано аналогічний, але більш складний приклад – з використанням непохідних елементів, наведених на рис. 7.3, а). Іншим

поданням структурної інформації образу слугує «граф відношень». Як приклад, на рис. 7.5 показано граф відношень зображення F на рис. 7.1, б). Оскільки між поміченими графами і матрицями існує взаємно однозначна відповідність, граф відношень може також бути виражений через «матрицю відношень».

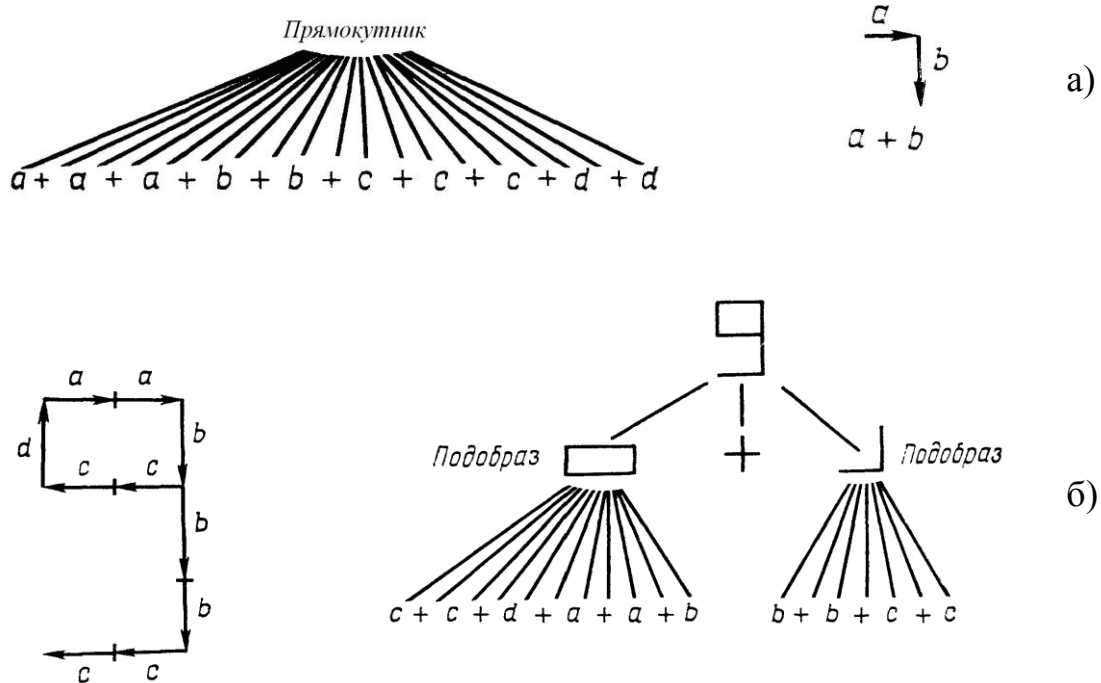


Рисунок 7.4

а) – структурний опис прямокутника, зображеного на рис. 7.3, б);
 б) – зображення цифри 9 і його структурний опис

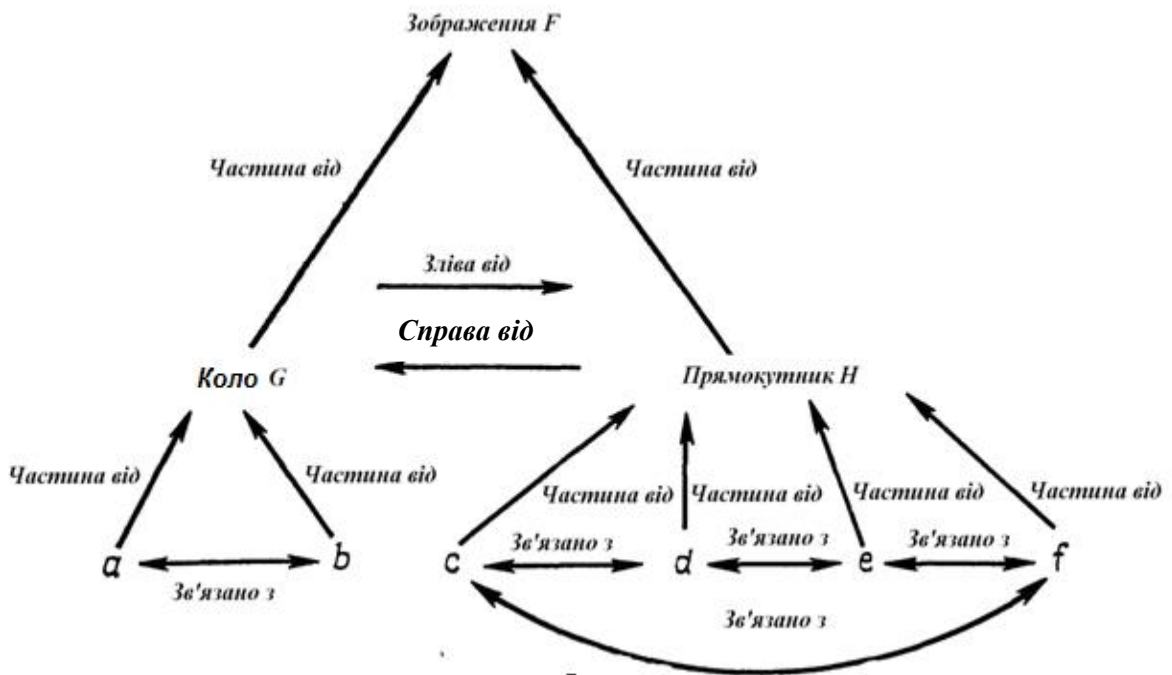


Рисунок 7.5 – Граф відношень зображення F

Шляхом використання графа відношення для опису образу можна розширити клас допустимих відношень, внівши до нього будь-яке відношення, яке зручно визначається з образу. Зауважимо, по-перше, що конкатенація – єдина природна операція для одновимірних мов і, по-друге, що граф, взагалі-то, містить цикли, тоді як дерево їх не містить. Тому за допомогою графа можна висловлювати більш багаті описи, ніж за допомогою деревоподібних структур. Проте використання деревоподібних структур дозволяє безпосередньо пристосувати методи теорії формальних мов до задачі компактного подання та аналізу зображень, що мають істотний структурний зміст. Синтаксичний підхід іноді називають лінгвістичним через те, що в ньому використовуються методи теорії формальних мов. Проте, мабуть, краще вважати, що методи теорії формальних мов є апаратом синтаксичного підходу, а не його суттю. Тому розробку ефективних структурних описів і аналізу образів не потрібно обмежувати розвитком математичної лінгвістики.

7.2 Структурна схема системи синтаксичного розпізнавання

У лінгвістичному підході до розпізнавання ознаками слугують субобрази, які називають непохідними елементами, а також відношення між ними, що характеризують структуру образу. Для опису образів через непохідні елементи і їхні відношення можна використовувати деяку «мову» образів. Правила такої мови, що дозволяють складати образи з непохідних елементів, називаються граматикою. В цьому випадку образ подається деяким реченням відповідно до діючої граматики.

Для розпізнавання деякого образу необхідно насамперед визначити його непохідні елементи і відношення між ними, після чого варто провести (синтаксичний) аналіз для того, щоб встановити, чи узгоджується опис образу з граматикою, що могла б його породити; такий аналіз часто називають «(граматичним) розбором».

Синтезувати граматику можна, спираючись на апріорні відомості про образи чи на результати аналізу деякої кінцевої множини репрезентативних образів (виведення граматики).

Структуру лінгвістичної системи розпізнавання наведено на рис. 7.6.

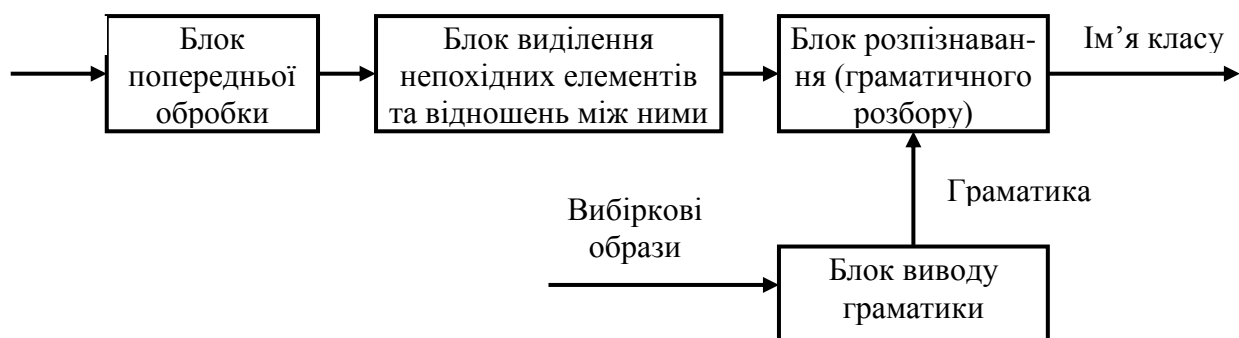


Рисунок 7.6 – Блок-схема системи синтаксичного розпізнавання образів

Систему синтаксичного розпізнавання образів можна подати у вигляді трьох основних частин, а саме блока попередньої обробки, блока опису або подання об'єкта і блока синтаксичного аналізу (розпізнавання).

Блок попередньої обробки здійснює, по-перше, функції кодування та апроксимації і, по-друге, функцій фільтрації, відновлення і поліпшення об'єкта. Зображення вихідного об'єкта спочатку кодується або апроксимується так, щоб це було зручно для подальшої обробки. Чорно-біле зображення, наприклад, можна кодувати за допомогою решітки (або матриці), що складається з нулів і одиниць, а криву можна апроксимувати відсіками в дискретні моменти часу або кінцевим набором коефіцієнтів розкладання цієї кривої в ряд Фур'є. Щоб зробити обробку в наступних блоках системи більш ефективною, в цьому блоці часто застосовують який-небудь спосіб «стиснення даних». Потім використовують методи фільтрації і відновлення та (або) поліпшення для ліквідації шуму, відновлення спотворень і (або) поліпшення якості закодованих (або апроксимованих) об'єктів. Кожен підданий попередній обробці об'єкт потім подають у вигляді структури мовного типу (наприклад, ланцюжка). Цей процес подання об'єкта полягає, по-перше, з сегментації і, по-друге, з виділення непохідних елементів (ознак). Щоб подати об'єкт через його субобрази, потрібно цей об'єкт сегментувати і в той самий час ідентифікувати (або виділити) у ньому непохідні елементи. Іншими словами, кожен підданий передобробці об'єкт поділяється на частини і непохідні елементи на основі заздалегідь заданих синтаксичних операцій (або операцій декомпозиції). Кожна виділена частина також ідентифікується відносно заданої множини непохідних елементів. На цій стадії кожен об'єкт отримує своє подання через множину непохідних елементів і певних синтаксичних операцій. Наприклад, якщо задано операцію конкатенації, то кожен об'єкт подається ланцюжком непохідних елементів.

Рішення про те, чи є подання об'єкта синтаксично правильним (тобто чи належить він до класу образів, що описуються цим синтаксисом або цією граматиною), приймається «блоком синтаксичного аналізу» або «блоком граматичного розбору». В процесі синтаксичного аналізу або граматичного розбору цей блок зазвичай може давати повний синтаксичний опис об'єкта в термінах граматичних одиниць або дерева граматичного розбору, якщо подання об'єкта синтаксично правильне. В іншому випадку об'єкт або виключають з розгляду, або аналізують на основі інших заданих граматик, які, можливо, описують інші можливі класи аналізованих образів. Найпростішою формою розпізнавання є, мабуть, «порівняння з еталоном». Ланцюжок непохідних елементів, що являє собою вихідний об'єкт, порівнюється з ланцюжками непохідних елементів, що описують кожен еталонний образ. З цих образів вибирається той, який найбільш «узгоджений» або «близький» до вихідного об'єкта відповідно, але за деяким заданим критерієм. Водночас

ієрархічна структурна інформація ігнорується. З іншого боку, за повного граматичного розбору ланцюжка, що подає, описує результатний об'єкт, досліджується повний структурний опис. Між цими крайнощами існує ряд проміжних підходів. Наприклад, можуть бути сконструйовані послідовності тестів для перевірки наявності певних субобразів або певних комбінацій підобразів (або непохідних елементів). Результати тестів (отримані, наприклад, за допомогою перегляду таблиці, дерева рішень або логічних операцій) використовуються для класифікації. Кожен тест може бути схемою порівняння з еталоном або граматичним розбором піддерева, що описує субобраз. Вибір підходу залежить від завдання. Якщо для розпізнавання потрібно повний опис об'єкта, то необхідним є граматичний розбір. В іншому випадку можна обійтися без повного граматичного розбору і для збільшення ефективності використовувати більш прості підходи.

Для того щоб отримати граматику, що описує структурну інформацію класу образів, необхідний пристрій виведення граматики, що дозволяє вивести її за заданою множиною об'єктів у вигляді, подібному мові. Функції цього пристрою аналогічні процесу «навчання» в дискримінантних системах розпізнавання образів. Внаслідок навчання на прикладах об'єктів розглянутого класу формується структурний опис цього класу. Потім отриманий у вигляді граматики опис використовується для опису образів і синтаксичного аналізу (рис. 7.6). Мабуть, для знаходження найкращої множини непохідних елементів та відповідного структурного опису класу розглядуваних образів необхідно навчання більш загального вигляду.

7.3 Поняття теорії формальних мов

7.3.1 Узагальнений опис застосування теорії формальних мов до опису і розпізнавання зображень

Виникнення теорії формальних мов в середині 50-х років пов'язане з розробкою Ноамом Хомським математичних моделей граматики в процесі дослідження природних мов. Одним з первинних завдань лінгвістів, що працюють в цій області, було створення «обчислювальних» граматики, здатних описувати природні мови, наприклад англійську. Була надія на те, що якщо задум вдасться, не буде складно навчити машину «розуміти» природні мови з метою машинного перекладу і вирішення завдань. І хоча, за загальною думкою, надії поки не виправдалися, побічні результати цих досліджень мають важливий вплив в інших областях, наприклад під час розробки компіляторів, в мовах програмування, теорії автоматів і, зовсім недавно, в розпізнаванні образів. У цьому підрозділі розглядається розвиток основних ідей теорії формальних мов через проблеми синтаксичного розпізнавання образів і навчання ЕОМ.

Граматики і мови. Граматика образів може бути використана для породження речень, що описують деякий образ, і для граматичного розбору речень, ціль якого полягає у визначенні відповідності їхньої структури застосовуваній граматиці. Розглянемо породження речень за допомогою декількох граматик, цікавих з погляду обробки зображень.

Породження речення починається з деякого початкового символу S , що належить деякій множини так званих нетермінальних символів V_N . Правило, що входить у набір правил P деякої граматики, можна застосувати для перетворення початкового символу S у деякий ланцюжок символів, наприклад $S \rightarrow aA$; це означає, що символ S замінений символами aA . Деякі символи, що входять у такий ланцюжок, можуть належати множині нетермінальних символів V_N , однак деякі інші символи можуть належати множині непохідних елементів V_T . Ці елементи будуть позначатися малими літерами і називатися термінальними символами, а нетермінальні символи – заголовними. Нетермінальні символи, що входять у ланцюжок, отриманий внаслідок перетворення, можуть бути піддані новому перетворенню відповідно до правил використовуваної граматики і т. д. Речення вважається побудованим, коли в ньому містяться тільки непохідні елементи. Граматики і типи правил визначаються теорією, що має високий ступінь формалізації.

Мовою $L(G)$ називається сукупність речень, породжених заданою граматиною.

7.3.2 Основні означення

В теорії формальних мов центральну роль відіграють поняття, описані нижче.

Алфавіт – будь-яка скінченна множина символів.

Речення в деякому алфавіті – довільний ланцюжок скінченної довжини, що складається з символів цього алфавіту.

Наприклад, для алфавіту $\{0,1\}$ допустимими є такі речення: $\{0,1,00,01,10,\dots\}$. Зазвичай для позначення речення використовують також терміни **ланцюжок** і **слово**.

Речення, що не містить жодного символу, називається порожнім реченням. Надалі пусте речення буде позначатися s_0 . Для довільного алфавіту V знак V^* буде використовуватися для позначення множини всіх речень, складених із символів алфавіту V , включно й пусте речення. Символ V^+ буде позначати множину речень $V^* - s_0$. Якщо, наприклад, задано алфавіт $V = \{a,b\}$, то

$$V^* = \{s_0, a, b, aa, ab, ba, \dots\} \text{ і } V^+ = \{a, b, aa, ab, ba, \dots\}.$$

Мова – довільна множина (не обов'язково скінченна) речень у деякому алфавіті.

Теорія формальних мов базується на граматиках та їхніх властивостях. Граматикою називається четвірка

$$G = (V_N, V_T, P, S), \quad (7.1)$$

де V_N – множина *нетермінальних символів* (змінних);

V_T – множина *термінальних символів* (констант);

P – множина *граматичних правил* або правил підстановки;

S – *початковий* або *кореневий* символ.

Передбачається, що S належить множині V_N і що V_N і V_T – множини, що не перетинаються. Алфавіт V є об'єднанням алфавітів V_N і V_T .

Приклад 7.2. Порівняти наведене вище означення формальної граматики зі стандартними поняттями граматики англійської мови, розглянувши просте речення *The boy runs* (хлопчик біжить).

Розв'язування. Подамо запис цього речення у вигляді дерева (рис. 7.7).

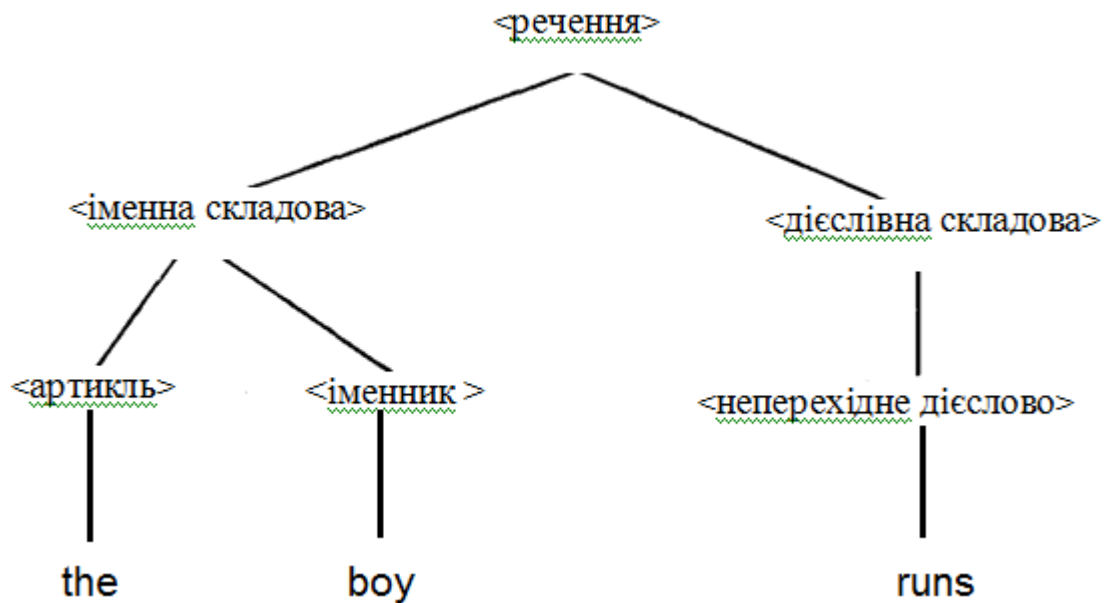


Рисунок 7.7 – Правила підстановки, використані під час породження речення «The boy runs» та відповідного семантичного дерева

<речення> → <іменна складова> <дієслівна складова>

<іменна складова> → <артикль> <іменник>

<дієслівна складова> → <неперехідне дієслово>

<артикль> → the

<іменник> → boy

<неперехідне дієслово> → runs

Породження цього речення відбувається таким чином. Процес починаємо з абстрактного поняття, яке називається структурним елементом <речення>. На цьому етапі <речення> – не більше ніж синтаксичне поняття, що подає всі правильні речення англійської мови. Потім ми замінюємо <речення> на <іменна складова> плюс <дієслівна складова>. У теорії формальних мов завжди починають з описаного вище символу S . Правила підстановки граматики G виду (7.1) відповідають в англійській мові такому, наприклад, заміщенню: <речення> замінюється на <іменна складова> і <дієслівна складова>. Як видно з рис. 7.7, внаслідок подальшого застосування граматичних правил або правил підстановки <іменна складова> зводиться до <артикль> плюс <іменник>, а <дієслівна складова> до <неперехідне дієслово>. Нарешті, застосування правил підстановки, які відображають <артикль> в «the», <іменник> в «boy», а <неперехідне дієслово> в «guns», приводить до шуканої пропозиції. **Нетермінальним символам** граматики G відповідає синтаксичним категоріям <іменна складова>, <дієслівна складова>, <артикль>, <іменник> і т. д., тоді як **термінальні символи** відповідають словам природної мови «the», «boy», «guns». Іншими словами, нетермінальні символи відіграють роль змінних, термінальні – констант.

Мова, породжувана граматиною G , позначається $L(G)$ – це множина ланцюжків, що задовольняють дві умови:

- 1) кожний ланцюжок складений тільки з термінальних символів (тобто є **термінальним**),
- 2) кожний ланцюжок може бути виведений з S шляхом відповідного застосування правил підстановки з множини P .

У цьому підрозділі використовуються такі позначення. Нетермінальні символи позначаються великими літерами S, A, B, C, \dots . Малі літери з першої половини латинського алфавіту a, b, c, \dots використовуються для термінальних символів. Ланцюжки термінальних символів позначаються малими літерами з кінця латинського алфавіту v, w, x, \dots . Змішані ланцюжки термінальних і нетермінальних символів подано малими літерами грецького алфавіту $\alpha, \beta, \gamma, \delta \dots$.

Множина P правил підстановки складається з виразів виду $\alpha \rightarrow \beta$, де α – ланцюжок в словнику V^+ і β – ланцюжок в словнику V^* . Тобто, символ \rightarrow означає заміщення ланцюжка α ланцюжком β . Символ \Rightarrow буде використаний для позначення операцій виду $\gamma\alpha\delta \Rightarrow \gamma\beta\delta$ у граматиці G , тобто \Rightarrow вказує на заміщення α на β внаслідок застосування правила підстановки $\alpha \rightarrow \beta$, водночас γ і δ залишаються незмінними. У тих випадках, коли ясно, про яку граматику йде мова, G опускається і використовується символ \Rightarrow .

Приклад 7.3. Розглянемо граматику $G = (V_N, N_T, P, S)$, де $V_N = \{S\}$, $V_T = \{a, b\}$ і $P = \{S \rightarrow aSb, S \rightarrow ab\}$. Застосовуючи перше правило $m - 1$ разів, отримуємо

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow a^3Sb^3 \Rightarrow \dots \Rightarrow a^{m-1}Sb^{m-1}.$$

Застосування другого правила приводить до ланцюжка

$$a^{m-1}Sb^{m-1} \Rightarrow a^m b^m.$$

Мова, породжувана цією граматикою, складається, як ми бачимо, виключно з ланцюжків подібного виду, причому довжина конкретного ланцюжка залежить від m . Мову $L(G)$ можна подати у вигляді $L(G) = \{a^m b^m \mid m \geq 1\}$. Варто відзначити, що проста граMATика, описана в цьому прикладі, має здатність породжувати мову з нескінченним числом ланцюжків або речень. Така граMATика згідно з класифікацією Н. Хомського, називається необмеженою (рекурсивно перечислюваною). У наступних підрозділах буде видно, як ця властивість створює труднощі під час використання цих понять у розпізнаванні образів.

7.3.2. Типи граматик

У цьому пункті розглянемо граматики, що є частковим випадком (8.1). Всі ці граматики мають форму $G = (V_N, V_T, P, S)$ і розрізняються лише за типом правил підстановки, допустимих в кожній з них.

Необмежена граMATика характеризується правилами підстановки виду $\alpha \rightarrow \beta$, де α – ланцюжок алфавіту V^+ , а β – ланцюжок алфавіту V .

ГраMATика безпосередньо складових, або контекстна, характеризується правилами підстановки виду $\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$, де α_1 і α_2 – елементи алфавіту V , β належить V^+ , а нетермінальний символ A належить V_N . Ця граMATика допускає заміщення нетермінального символу A ланцюжком β тільки в тому випадку, якщо A з'являється у контексті $\alpha_1 A \alpha_2$, складеному з ланцюжків α_1 і α_2 . Наприклад, правило виду $bB \rightarrow bb$ означає, що символ B може бути замінений символом b тільки в тих випадках, коли перед B стоїть b ; таким чином, у цьому випадку має значення контекст допоміжного символу.

Безконтекстна граMATика характеризується правилами підстановки виду $A \rightarrow \beta$, де A належить множині нетермінальних символів V_N , а β належить множині непустих ланцюжків V^+ . Сама назва «безконтекстна» вказує на те, що змінна A може заміщатися ланцюжком β незалежно від контексту, в якому з'являється A .

Нарешті, **регулярна** (або **автоматна**) **граматика** – це граматика з правилами підстановки виду $A \rightarrow aB$ або $A \rightarrow a$, де A і B – змінні з V_N , a – термінальний символ з V_T . Альтернативними допустимими правилами підстановки є $A \rightarrow Ba$ і $A \rightarrow a$. Вибір одного з цих двох типів правил виключає, однак, застосування правил іншого типу.

Ці граматиками називають іноді **граматиками типу 0, 1, 2 і 3**, відповідно. Крім того, їх часто позначають як **граматиками структури складових**.

Якщо кожне правило підстановки безконтекстної граматиками має вигляд $A \rightarrow xBw$ або $A \rightarrow w$, де A і B – нетермінальні символи, а x та w – термінальні ланцюжки, то граматика вважається **лінійною**.

Цікаво відзначити, що всі регулярні граматиками безконтекстні, всі безконтекстні граматиками є граматиками безпосередньо складових, а всі граматиками безпосередньо складових – необмежені.

Приклад 7.4. Способи функціонування обговорюваних граматик показано на таких простих прикладах граматик.

(0) Необмежена граматика

$$G = (V_N, V_T, P, S)$$

за

$$V_N = \{S, A, B\}, \quad V_T = \{a, b, c\}$$

$$P: \quad S \rightarrow aAbc \\ Ab \rightarrow bA \\ Ac \rightarrow Bbcc \\ bB \rightarrow Bb \\ aB \rightarrow aaA \\ aB \rightarrow S_0$$

породжує речення виду $x = a^n b^{n+2} c^{n+2}$, де $n \geq 0$ означає довжину ланцюжка символів. Наприклад, для породження ланцюжка $x = a^0 b^2 c^2 = bbcc$ в цьому випадку застосовуються перші чотири правила і потім останнє, тобто

$$S \Rightarrow aAbc \Rightarrow abAc \Rightarrow abBbcc \Rightarrow aBbbcc \Rightarrow bbcc.$$

Зауважимо, що останнє правило припустиме лише в необмежених граматиках.

(1) Граматика безпосередньо складових

$$G = (V_N, V_T, P, S)$$

за

$$\begin{aligned}
V_N &= \{S, A, B\}, & V_T &= \{a, b, c\} \\
P: & S \rightarrow abc \\
& S \rightarrow aAbc \\
& Ab \rightarrow bA \\
& Ac \rightarrow Bbcc \\
& bB \rightarrow Bb \\
& aB \rightarrow aaA \\
& aB \rightarrow aa
\end{aligned}$$

породжує пропозиції виду $x = a^n b^n c^n$, де $n \geq 1$

(2) Бесконтекстна граматика

$$G = (V_N, V_T, P, S)$$

за

$$\begin{aligned}
V_N &= \{S\}, & V_T &= \{a, b\}, \\
P: & S \rightarrow ab, \\
& S \rightarrow aSb
\end{aligned}$$

породжує ланцюжки виду $x = a^n b^n$, де $n \geq 1$.

(3) Регулярна граматика

$$G = (V_N, V_T, P, S)$$

за

$$\begin{aligned}
V_N &= \{S\}, & V_T &= \{a, b\}, \\
P: & S \rightarrow a, \\
& S \rightarrow b, \\
& S \rightarrow aS, \\
& S \rightarrow bS
\end{aligned}$$

породжує ланцюжки, що складаються з символів a і b .

Як і передбачалося, необмежені граматики мають значно більшу потужність, ніж граматики трьох інших типів.

Однак ступінь спільності цих граматик створює ряд серйозних труднощів у їх теоретичних і практичних додатках. Це твердження є правильним і для граматик безпосередніх складових.

Хоча в літературі часто зустрічаються й інші граматичні структури, граматики, подані тут, становлять основу для більшої частини досліджень у цій області. В наступних розділах мова буде йти про розширення цих понять і їх застосування до розпізнавання образів.

7.4 Структурна схема системи синтаксичного розпізнавання образів

Поняття, описані в підрозділі 7.3, можуть бути пов'язані з розпізнаванням таким чином. Припустимо, у нас є два класи образів w_1 і w_2 і нехай образи цих класів можуть бути побудовані з ознак, що належать деякій скінченній множині. Назвемо ці ознаки терміналами і позначимо безліч терміналів символом V_T відповідно до системи позначень, введеної в попередньому підрозділі. У синтаксичному розпізнаванні образів термінали називаються також непохідними символами (елементами). Кожен образ може розглядатися як ланцюжок або речення, оскільки він складений з термінальних символів множини V_T . Припустимо, що існує граматики G , така, що породжувана нею мова $L(G)$ складається з речень (образів), які належать виключно одному з класів, скажімо w_1 . Очевидно, що ця граматики може бути використана з метою класифікації образів, оскільки заданий образ невідомої природи може бути віднесений до w_1 в тому випадку, коли він є пропозицією мови $L(G)$. В іншому випадку образ приписується класу w_2 . Наприклад, бесконтекстна граматики $G = (V_N, V_T, P, S)$ за $V_N = \{S\}$, $V_T = \{a, b\}$ і множини правил підстановки $P = \{S \rightarrow aaSb, S \rightarrow aab\}$ має здатність породжувати лише речення, що містять вдвічі більше символів a , ніж b . Якщо ми сформулюємо синтаксичну задачу розбиття образів на два класи, причому об'єкти класу w_1 – це ланцюжки виду aab , $aaaabb$ і т. д., а об'єкти класу w_2 містять однакову кількість символів a і b (тобто ab , $aabb$ і т. д.), то очевидно, що класифікація заданого ланцюжка проводиться простим визначенням того, чи може цей ланцюжок породжуватися граматикою G , розглянутою вище. Якщо може, то об'єкт належить w_1 , якщо ні – він автоматично приписується класу w_2 . Процедура, що використовується для визначення, є чи ні цей ланцюжок граматично правильним реченням для цієї мови, називається *граматичним розбором*. Способи граматичного розбору будуть розглянуті трохи пізніше.

За наведеною вище схемою класифікації образ потрапляє в клас w_2 виключно тому, що він не належить класу w_1 . Якщо виявиться, що образ не є правильним для G реченням, передбачається, що він має належати w_2 . Однак не виключено, що образ не належить і класу w_2 . Він може являти собою зашумлений або спотворений ланцюжок, який найкраще вилучити з розпізнавання. Для того щоб забезпечити можливість вилучення, необхідно задати дві граматики G_1 і G_2 , що породжують мови $L(G_1)$ і $L(G_2)$, відповідно. Образ зараховується до класу, мова якого дозволяє вважати цей образ граматично правильним реченням. Якщо виявиться, що

образ може належати обом класам, його довільно зараховують в будь-який з цих класів. Якщо образ не є реченням ні в мові $L(G_1)$, ні в мові $L(G_2)$, він вилучається.

У разі M класів ми розглядаємо M граматик і пов'язаних з ними мов $L(G_i)$, $i = 1, 2, \dots, M$. Розпізнаваний об'єкт належить до класу w_i в тому і тільки тому випадку, якщо він є реченням мови $L(G_i)$. Якщо об'єкт є реченням більш ніж однієї мови або не належить ні до однієї з них, він може бути вилучений з розгляду або довільно віднесений до одного з класів невизначених об'єктів.

У зв'язку з викладеним вище підходом до розпізнавання виникає ряд суттєвих питань:

1. Як найкращим чином описувати об'єкти для їх класифікації подібним способом?

2. Як вибираються граматика розпізнавання?

3. Будь-яке завдання, що має практичне значення, зазвичай спотворюється випадковими змінними, наприклад шумами вимірювання; яким чином у синтаксичному підході можна використовувати статистичний апарат з тим, щоб врахувати ці випадкові змінні?

4. Однією з найбільш важливих особливостей розглянутих раніше підходів до розпізнавання образів є здатність будувати вирішувальні функції за допомогою ітеративних процедур навчання; чи можна поширити концепцію навчальної вибірки на синтаксичне розпізнавання?

7.5 Застосування граматики в розпізнаванні образів

7.5.1 Синтаксично-орієнтована граMATика

У попередньому підрозділі було зазначено, що формальні граматики можна використовувати в розпізнаванні образів, визначаючи, чи є заданий об'єкт термінальним реченням будь-якої з відповідних розглянутих задачі граматик. Основним питанням, після того, як визначено граматику, є розробка процедури, що встановлює, є чи ні цей об'єкт допустимим реченням. Процедура, що застосовується в цьому випадку, в теорії формальних мов, називається граMATичним розбором. Переважно розглядають два типи граMATичного розбору: зверху вниз і знизу вверху. Ці назви стають більш зрозумілими, якщо звернутися до семантичного дерева, такого, наприклад, як подане на рис. 7.1. Вершина або корінь (інвертованого) дерева – це початковий символ S . Термінальні речення (образи) – це нижня частину або листя дерева. Процедура розбору зверху вниз починається з кореневого символу S і полягає в спробах за допомогою застосування граMATичних правил, що повторюються, отримати задане термінальне речення. З іншого боку, процедура аналізу знизу вверху починається з конкретної пропозиції і полягає в спробах дійти до символу S за допомогою інверсії правил підстановки. У кожному з цих випадків у

разі невдалого результату граматичного аналізу заданий образ відхиляється як такий, що описує неправильну пропозицію. Очевидно, що описані вище схеми граматичного аналізу принципово неефективні, оскільки потребують повного перебору за застосування граматичних правил.

Часто немає необхідності застосовувати послідовність граматичних правил від початку до кінця, оскільки існує можливість перевіряти на відповідність поставленим цілям проміжні результати і визначати тим самим, чи здатна ця послідовність правил забезпечувати успішний граматичний розбір.

Подальше вдосконалення процесу граматичного аналізу пов'язане із застосуванням правил синтаксису граматики. Синтаксис визначається як з'єднання та конкатенація об'єктів. Синтаксичне правило встановлює деякі допустимі (чи заборонені) відношення між об'єктами. Наприклад, з'єднання qqq ніколи не зустрічається англійською мовою. У цій термінології граMATика є більш ніж безліччю синтаксичних правил, що визначають допустимі чи бажані відношення між об'єктами. Синтаксично орієнтований граматичний аналізатор, таким чином, включає в процес граматичного аналізу синтаксис граматики. Наступний приклад дозволить нам внести більшу ясність у ці поняття.

Приклад 7.5. Повернемося до структур типу квадрат, використаємо для ілюстрації змісту попереднього параграфа. Непохідними елементами, як показано на рис. 7.8, а), слугують горизонтальний і вертикальний відрізки певної довжини, позначені a_1 і a_2 , відповідно. Безконтекстна граMATика G , здатна породжувати квадрати, задається набором (7.1)

$$G = \{V_N, V_T, P, S\}$$

за

$$\begin{aligned} V_T &= \{a_1, a_2\}, V_N = \{S, O_1, O_2\}; \\ P: S &\rightarrow A(a_1, O_2), \\ O_2 &\rightarrow A(O_1, a_1), \\ O_1 &\rightarrow L(a_2, a_2), \end{aligned}$$

де $A(x, y)$ та $L(x, y)$ – дескриптори, що читаються відповідно « x розташований над y » і « x розташований зліва від y ».

Тут потрібно відзначити роль дескрипторів, які під час обробки зображень дають можливість узагальнювати граматичні правила так, щоб вони могли застосовуватись до двовимірних сполук. У цьому простому прикладі позиційний дескриптор $A(x, y)$ допустимий тільки в тому випадку, коли частина знаходиться безпосередньо над x , а дескриптор

$L(x,y)$ допустимий тільки тоді, коли частина знаходиться безпосередньо праворуч від x .

Структури, що нагадують квадрати, зображені на рис. 7.8, б), породжуються послідовністю граматичних правил

$$S \rightarrow A(a_1, O_2).$$

Це правило замінює початковий символ непохідним елементом a_1 , розташованим над деяким поки що не визначеним об'єктом O_2 . Правило

$$O_2 \rightarrow A(O_1, a_1)$$

замінює невизначений об'єкт O_2 іншим об'єктом O_1 ще не визначеним, розташованим над горизонтальним відрізком a_1 .

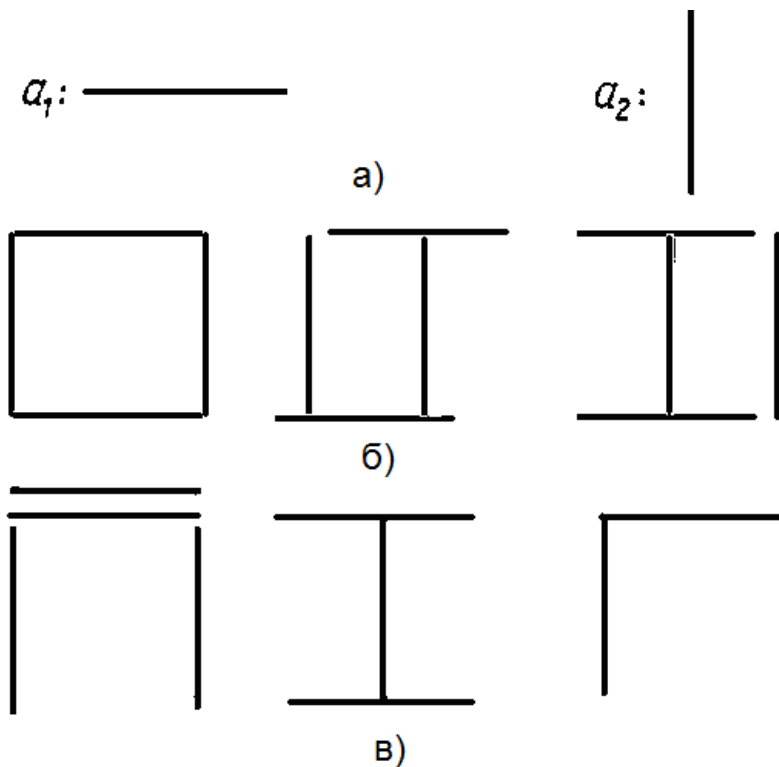


Рисунок 7.8 – Образи, використані для ілюстрації синтаксично-орієнтованого граматичного розбору: а) – непохідні елементи образів; б) – образи, що піддаються розбору за допомогою описаної схеми; в) – образи, що не піддаються розбору за допомогою описаної схеми.

Нарешті, O_1 замінюється на два вертикальних непохідних елемента за допомогою застосування правила

$$O_1 \rightarrow L(a_2, a_2).$$

Іншими словами, побудова цих структур починається з горизонтального відрізка, потім йде інший горизонтальний відрізок під ним, і завершується все розміщенням між ними двох вертикальних відрізків. Мінливістю структур можна управляти, накладаючи обмеження на позиційні дескриптори $A(x, y)$ та $L(x, y)$. Варто зазначити, що наведена вище граMATика здатна породжувати лише структури типу квадрата і що лише наведена вище послідовність правил вважається допустимою.

ГраMATичний розбір, проведений у цій простій системі, є тривіальною процедурою, оскільки використовується лише одна послідовність правил підстановки. Припустимо, наприклад, що потрібно встановити, належить чи належить ця структура до класу об'єктів, породжуваних описаною вище граMATикою. Синтаксично орієнтований розбір зверху вниз буде проводитися в такий спосіб. Перше правило підстановки починається з S і передбачає пошук деякого об'єкта O_2 нижче непохідного елемента a_1 . Якщо нижче деякого a_1 не знайдено жодного об'єкта, граMATичний розбір переривається і образ відхиляється. Якщо це правило застосовано успішно, на наступному кроці відшуковується деякий об'єкт O_1 над іншим непохідним елементом a_1 . Перший елемент a не вважається частиною O_1 . Якщо O_1 виявлено, граMATичний розбір продовжується, в протилежному випадку образ відхиляється. Нарешті, об'єкт O_1 , виявлений на попередньому кроці, має для прийняття образу розділитися на два непохідні елементи a_2 за умовою $L(a_2, a_2)$. Цій схемі граMATичного аналізу задовольняють структури, зображені на рис. 7.8, б), і не задовольняють структури, зображені на рис. 7.8, в).

Синтаксично орієнтований граMATичний розбір знизу вверху, що полягає у застосуванні правил підстановки у зворотному порядку, відбувається так.

Спочатку ми намагаємося виявити об'єкт O_1 , визначаючи, чи містить цей об'єкт непохідний елемент a_2 зліва від непохідного елемента a_2 . Якщо пошук виявився вдалим, процедура продовжується, інакше образ відхиляється.

Оскільки в цьому випадку процедура розбору знизу вверху починається з термінального речення, то спочатку мають розглядатися ті правила підстановки, застосування яких приводить виключно до термінальних символів.

Для продовження граMATичного аналізу необхідно на наступному кроці отримати об'єкт O_2 , який складається з об'єкта O_1 , розташованого над непохідним елементом a_1 . Якщо цей крок є успішним, то здійснюється вивести початковий символ S , відшукуючи непохідний елемент a_1 , розташований над O_2 . Якщо S може бути виведений, то образ приймається, інакше на цьому етапі він відхиляється. Об'єкти, зображені на рис. 8.8, б) піддаються граMATичному розбору, тоді як об'єкти, зображені на рис. 8.8, в) будуть відхилені на одному з етапів граMATичного розбору.

7.5.2 Граматика дерев

Щоб обробляти деревоподібні структури, необхідно незначно модифікувати попереднє означення граматики (7.1). Граматика дерев визначається як п'ятірка

$$G = (V_N, V_T, P, R, S), \quad (7.2)$$

де V_N і V_T , як і раніше, – множини непохідних (нетерміналів і похідних (терміналів) елементів відповідно,

S – початковий символ, який може бути деревом,

P – множина граматичних правил виду $\Omega \rightarrow \Psi$, де Ω і Ψ – дерева,

R – функція ранжування, що означає кількість прямих нащадків вузла, мітка якого є термінальним символом цієї граматики.

Приклад 7.6. Побудувати граматику виведення дерева для електричної схеми, поданої на рис. 7.9, а).

Розв'язування. У цьому випадку отриманий двовимірний об'єкт визначається зв'язками між різними компонентами схеми. На рис. 7.9, б) показано відповідне дерево. Зазначимо, кожен вузол дерева позначає вузол схеми і що корінь дерева образу довільно визначається як крайній лівий верхній вузол образу.

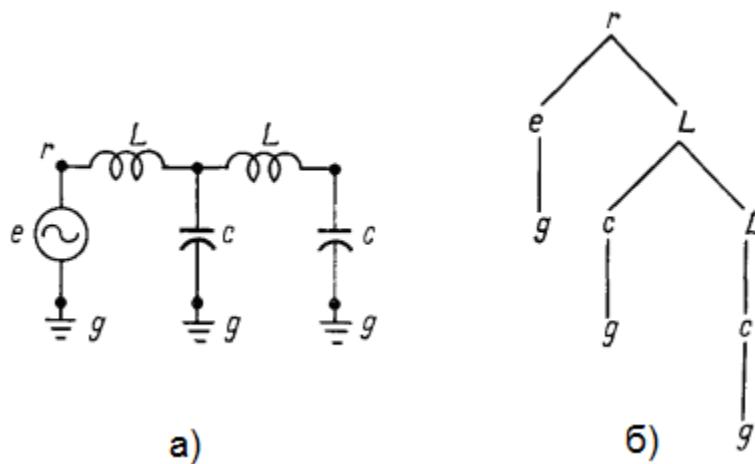
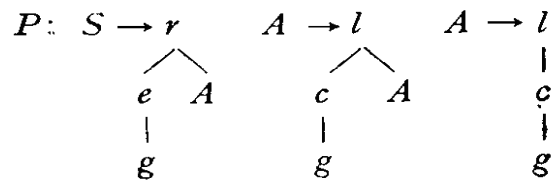


Рисунок 7.9 – Електрична схема – а)
і подання її образу у вигляді дерева – б)

Граматика, що породжує цей об'єкт, складається з таких елементів:

$$V_N = \{A, S\}, \quad V_T = \{r, e, l, c, g\}$$



$$\begin{aligned}
 R(r) &= 2 \\
 R(l) &= \{2,1\} \\
 R(e) &= 1 \\
 R(g) &= 0 \\
 R(c) &= 1
 \end{aligned}$$

Щоб породити конкретний образ, необхідно переписати всі нетермінальні символи на вузлах дерева в такий спосіб, щоб сформувати дерево, всі вузли якого мають термінальні мітки (у цій граматиці).

Розпізнавання деревоподібних структур може здійснюватися методами, обговореними раніше у цьому розділі, за винятком правил підстановки, які, відображуючи специфіку граматики дерев, «зобов'язані» мати деревоподібну структуру.

Контрольні питання та завдання

1. В чому полягає суть синтаксичного розпізнавання образів?
2. Чому синтаксичне розпізнавання інакше називають структурним?
3. Назвіть приклади задач розпізнавання, які вирішуються синтаксичним підходом.
4. Які елементи образів називають непохідними?
5. Наведіть приклади образів з непохідними елементами.
6. Що таке граMATика і як вона використовується в синтаксичному підході для розпізнавання?
7. Яку можливість надає використання графів в поданні образів?
8. Нарисуйте структурну схему системи синтаксичного розпізнавання образів і опишіть призначення її частин.
9. Дайте неформальне означення граматики і мови.
10. Запишіть формальне означення граматики і мови в синтаксичному підході до розпізнавання. Опишіть складові граматики.
11. Наведіть приклад породження речення деякою граMATикою.
12. Які типи граMATик Ви знаєте?
13. Опишіть складові необмеженої граматики.
14. Опишіть складові граматики безпосередньо складових (контекстної граматики).
15. Опишіть складові безконтекстної граматики.
16. Опишіть складові регулярної граматики.
17. Як формулюється задача синтаксичного розпізнавання образу?

18. Які питання потрібно вирішити під час розробки синтаксичного методу розпізнавання?
19. Опишіть процедуру синтаксичного розбору структурного образу методом зверху вниз.
20. Опишіть процедуру синтаксичного розбору структурного образу методом знизу вверху.
21. Чим відрізняється граматики побудови дерева від звичайної синтаксичної граматики?
22. Наведіть приклад граматики виведення дерева для деякої електричної схеми.

ЛІТЕРАТУРА

1. Patrick E. A. Fundamentals of Pattern Recognition / Edward A. Patrick. - New York: Prentice-Hall, 1972. – 408 p.
2. Duda R.O., Hart P. E. Pattern classification and scene analysis / Richard O. Duda, Peter E. Hart. - New York-London: John Wiley & Sons, 1973. – 511 p.
3. Fu K.S. Sintactic Methods in Pattern Recognition / K. S. Fu. – New Yourk-London: Academic Press, 1974. – 320 p.
4. Tou J. T., Gonzales R. C. Pattern Recognition Principles / Julius T. Tou, Rafael C Gonzalez. - London-Amsterdam-Don Mills: Addison-Weasley Publishing Center, 1977. – 411 p.
5. Winston P. H. Artificial Intelligence / P. H. Winston. – London-Amsterdam-Don Mills: Addison-Weasley Publishing Center, 1977. – 513 p.
6. Faure A. Perception et reconnaissance des formes / Alain Faure. – Paris: Editests, 1985. - 286 p.
7. Looney C.G. Pattern Recognition Using Neural Networks / C. G. Looney. –New York: Oxford University Press, 1997. – 449 p.
8. Nillson N. Artificial Intelligence. A New Synthesis / N/ Nillson. – S/ Francisco: Morgan Kaufmann Publishers, CA, 1998. – 514 p.
9. Fukunaga K. Introduction to Ststistical Pattern Recognition. 2-nd edition / Keinosuke Fukunaga. - San Diego - San Francisco - New York: Academic Press, 1990. - 609 p.
10. Bykov N. M., Kuzmin I. V., Yakovenko A. I. Development of effective strategy of pattern recognition // - *Proceedings of SPIE*, 2001, Vol. 4225, pp.76-83.
11. Russel S. J., Norvig P. Artificial Intelligence. A modern Approach. – Second Edition / Stuart J. Russel, Peter Norvig. – New Jersey: Prentice Hall, 2003. – 1408 p.
12. 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>

13. Prateek J. Artificial Intelligence with Python / Joshi Prateek. – Birmigham- Mumbai: Packt Publishing Ltd, 2019. – 423 p.
14. Биков М. М., Грищук Т. В. Розробка методів оцінки ефективності автоматизованих систем розпізнавання мови // Вісник Технологічного університету Поділля. Хмельницький, ТУП, 2003. № 3, том 1. С. 122–125.
15. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень. Запоріжжя: ЗНТУ, 2008. 341 с.
16. Методичні вказівки до виконання лабораторних робіт з дисципліни «Інтелектуальні засоби систем автоматизації» для студентів спеціальності 151 – Автоматизація та комп'ютерно- інтегровані технології / Уклад. М. М. Биков, Т. В. Грищук, Г. Ю. Дерман, В. В. Ковтун. Вінниця: ВНТУ, 2018. 48 с.
17. Глибовець М. М., Олецький О. В. Штучний інтелект К. : Видавничий дім «КМ Академія», 2022. 366 с.
18. Лі К. Ф., Цюфань Ч. Штучний інтелект Київ : BookChef, 2022. 464 с.

*Навчальне електронне видання
комбінованого використання.
Можна використовувати в локальному та мережному режимах*

**Микола Максимович Биков,
Вячеслав Васильович Ковтун
Володимир Олександрович Гаврилюк**

**ОСНОВИ ІНТЕЛЕКТУАЛЬНИХ
ТЕХНОЛОГІЙ**
Частина 1 . Технології розпізнавання

Навчальний посібник

Рукопис оформив *М. Биков*

Редактор *Т. Старічек*

Оригінал-макет підготувала *Т. Старічек*

Підписано до видання 29.08.2023 р.
Гарнітура Times New Roman.
Зам. № P2023-097.

Видавець та виготовлювач
Вінницький національний технічний університет,
Редакційно-видавничий відділ.
ВНТУ, ГНК, к. 114.
Хмельницьке шосе, 95, м. Вінниця, 21021.
Тел. (0432) 65-18-06.
press.vntu.edu.ua;
E-mail: irvc.ed.vntu@gmail.com.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.