

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ

С. О. Субботін, А. О. Олійник

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

Навчальний посібник

під загальною редакцією
д-ра техн. наук, професора С. О. Субботіна

*Рекомендовано Міністерством освіти і науки України
як навчальний посібник для студентів
вищих навчальних закладів, які навчаються
за напрямом підготовки «Програмна інженерія»*

*Видання здійснено за підтримки міжнародного проекту
«Centers of Excellence for young RESearchers» (CeRes)
програми «Tempus» Європейської комісії
(реєстраційний номер
544137-TEMPUS-1-2013-1-SK-TEMPUS-JPHES)*

Запоріжжя
2014

УДК 004.4:004.05:004.93

ББК 32.973

C-89



Tempus



Education and Culture DG

*Гриф надано Міністерством освіти і науки України
(лист № 1/11-18206 від 27.11.2013 р.)*

Колектив авторів:

С. О. Субботін, д-р техн. наук, професор – розділи 1, 2, 4, 5;

А. О. Олійник, канд. техн. наук, доцент – розділ 3

Рецензенти:

*Єрохін А. Л. – доктор технічних наук, професор, професор кафедри програмної інженерії Харківського національного університету радіоелектроніки;
Гоменюк С. І. – доктор технічних наук, професор, декан математичного факультету Запорізького національного університету;*

Литвиненко В. І. – доктор технічних наук, доцент, завідувач кафедри кафедри інформатики та комп’ютерних наук Херсонського національного технічного університету

Субботін С. О.

**C-89 Інтелектуальні системи : навч. посіб. / С. О. Субботін,
А. О. Олійник; під заг. ред. проф. С. О. Субботіна. – Запоріжжя :
ЗНТУ, 2014. – 218 с.**

ISBN 978-617-529-095-8

Книга містить систематизований виклад основних понять, моделей і методів штучного інтелекту, які можуть використовуватися при вирішенні практичних завдань розпізнавання образів, прийняття рішень, аналізу та класифікації даних різної природи та прогнозування. Розглянуто питання побудови інтелектуальних систем, заснованих на знаннях, а також методи пошуку і виведення в інтелектуальних системах.

Видання призначено для студентів комп’ютерних спеціальностей вищих навчальних закладів, а також може використовуватися педагогічними працівниками та практичними фахівцями.

УДК 004.4:004.05:004.93

ББК 32.973

ISBN 978-617-529-095-8

© Запорізький національний
технічний університет (ЗНТУ), 2014
© Субботін С. О., Олійник А. О. 2014

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1	
ОСНОВНІ ПОНЯТТЯ ШТУЧНОГО ІНТЕЛЕКТУ	8
1.1 Інтелектуальні системи.....	8
1.2 Проблемні області та їхні властивості	10
1.3 Агентний підхід до інтелектуальних систем.....	12
1.4 Історія розвитку штучного інтелекту	14
1.5 Контрольні питання	21
1.6. Практичні завдання.....	21
1.7 Література до розділу	22
РОЗДІЛ 2	
РОЗПІЗНАВАННЯ ОБРАЗІВ	23
2.1 Задачі розпізнавання образів. Навчання з учителем	23
2.2 Лінійна роздільність та нелінійна роздільність класів	25
2.3 Класифікація і порівняльна характеристика методів розвізнавання образів	26
2.4 Метод метричної класифікації.....	40
2.5 Кластерний аналіз. Навчання без учителя	42
2.6. Приклади виконання завдань.....	56
2.7 Контрольні питання	58
2.8 Практичні завдання.....	59
2.9 Література до розділу	61
РОЗДІЛ 3	
ВІДБІР ІНФОРМАТИВНИХ ОЗНАК	62
3.1 Загальна постановка задачі відбору інформативних ознак для синтезу розпізнавальних моделей	62
3.2 Структура методів відбору ознак	64
3.3 Методи відбору інформативних ознак	67
3.4 Критерії оцінювання інформативності ознак.....	81
3.5 Метрики, використовувані при кластеризації ознак.....	96
3.6 Методи формування штучних ознак	99
3.7 Приклади виконання завдань	101
3.8 Контрольні питання	115
3.9 Практичні завдання.....	116
3.10 Література до розділу	119

РОЗДІЛ 4

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ, ЗАСНОВАНІ НА ЗНАННЯХ	120
4.1 Знання та їх властивості	120
4.2 Принципи побудови систем, заснованих на знаннях.....	126
4.3 Експертні системи.....	128
4.4 Логічне виведення	146
4.5 Чітке логічне виведення	147
4.6 Нечітке логічне виведення.....	158
4.7 Пошук у просторі станів	169
4.8 Приклади виконання завдань	185
4.9 Контрольні питання	189
4.10 Практичні завдання.....	191
4.11 Література до розділу	193

РОЗДІЛ 5

ПРОГРАМНІ ЗАСОБИ ДЛЯ ПОБУДОВИ

ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ.....	194
5.1 Подання і введення-виведення даних	195
5.2 Кластер-аналіз даних	197
5.3 Відбір інформативних ознак.....	199
5.4 Побудова моделей прийняття рішень	200
5.5 Приклади виконання завдань	201
5.7 Практичні завдання.....	204
5.8 Література до розділу	208
Алфавітно-предметний покажчик.....	209
ЛІТЕРАТУРА	216

ВСТУП

Розв'язання завдань автоматизації прийняття рішень, керування складними технічними об'єктами і процесами, обробки і аналізу великих обсягів даних, прогнозування, побудови системи технічного і біомедичного діагностування, спрощення спілкування людини та ЕОМ, пов'язано з необхідністю використання інтелектуальних інформаційних технологій, що містять методи, моделі та програмні засоби штучного інтелекту.

Актуальність створення, вивчення та використання методів, моделей та комп'ютерних систем, що в своїй роботі застосовують методи штучного інтелекту, підтверджується Державною науково-технічною програмою «Нові вітчизняні інтелектуальні комп'ютерні засоби», Указом Президента України №102/2003 від 12.02.2003 «Про Концепцію державної промислової політики», що визначають як пріоритет інформатизації створення систем підтримки прийняття рішень та штучного інтелекту; Постановою Кабінету Міністрів України №1896 від 10.12.2003, яка передбачає «... розроблення методології інтелектуального аналізу даних ... на основі застосування сучасних методів нечіткої логіки, штучного інтелекту та добування знань із баз даних»; Постановою Кабінету Міністрів України №789 від 15 липня 1997 р. «Про першочергові заходи інформатизації», Законом України №75/98–ВР від 04.02.1998 «Про Концепцію Національної програми інформатизації», які передбачають, зокрема, «створити діючі зразки та прототипи конкурентоспроможних засобів та систем: методичне та програмне забезпечення проектування і розроблення комп'ютеризованих систем для застосування в управлінні, програмно-технічні засоби підтримки експертного прийняття рішень, програмне забезпечення інформаційно-аналітичної обробки ... фактографічних та статистичних даних, ... засоби інтелектуалізації широкого застосування».

Метою даної книги є систематизований виклад основних понять, моделей і методів штучного інтелекту, які можуть використовуватися для побудови інтелектуальних систем при вирішенні практичних завдань розпізнавання образів, прийняття рішень, класифікації та прогнозування, технічного і біомедичного діагностування.

У першому розділі розглянуто основні поняття теорії штучного інтелекту. Описано задачі та дано класифікацію систем штучного інтелекту. Наведено короткий історичний нарис історії розвитку штучного інтелекту. Описано сучасний агентний підхід щодо створення інтелектуальних систем.

Другий розділ присвячено методам теорії розпізнавання образів. Наведено загальні постановки задач розпізнавання образів, введено необхідні поняття з теорії розпізнавання. У розділі подана порівняльна характеристика основних груп методів розпізнавання образів. Розглянуто метод еталонів. Значну увагу приділено питанням чіткого і нечіткого кластер-аналізу.

У третьому розділі розглянуто методи пошуку найбільш інформативної комбінації ознак у навчальній вибірці. Наведено постанову завдання відбору ознак, проаналізовано основні методи вибору інформативної комбінації (метод повного перебору, пошук у глибину, пошук вглибину, метод гілок і границь, метод групового врахування аргументів, ранжирування ознак, випадковий пошук з адаптацією та ін.), розглянуто критерії оцінювання індивідуальної та групової значущості ознак.

Четвертий розділ присвячено системам, заснованим на знаннях. Розглянуто основні види знати. Наведено опис основних елементів систем, заснованих на знаннях. Детально розглянуто принципи побудови експертних систем. Описано методи виведення та пошуку рішень у просторі станів.

У п'ятому розділі розглянуто програмні засоби для побудови інтелектуальних систем. Особливу увагу приділено автоматизації вирішення завдань розпізнавання образів, кластерного аналізу, відбору інформативних ознак.

Для спрощення **самостійного опрацювання** та кращого за своєння матеріалу книги наприкінці кожного розділу наведено контрольні питання, а також практичні та тестові завдання.

Видання орієнтоване на студентів комп’ютерних спеціальностей вищих навчальних закладів, а також може використовуватися педагогічними працівниками та практичними фахівцями.

Матеріал, наведений у книзі, призначений для вивчення курсу «Інтелектуальні системи». Наведені у книзі теоретичні відомості та методичні матеріали розроблені й апробовані авторами при читанні курсів «Інтелектуальні системи», «Інтелектуальні системи інже-

нерного забезпечення виробництва», «Системи штучного інтелекту» у Запорізькому національному технічному університеті. Книга також може використовуватися при вивчені окремих розділів дисциплін «Інтелектуальний аналіз даних», «Основи обчислювально-го інтелекту», «Технологія та використання штучних нейронних мереж» та інших.

Більш детальна інформація про використання матеріалу книги у навчальному процесі, а також посилання корисні літературні джерела та рекомендовані програмні засоби доступні на веб-сайті авторів за адресою: <http://csit.narod.ru>.

Терміни та назви методів, визначення яких наводиться у наступному тексті виділено курсивом, назви тематичних груп термінів або методів виділено жирним курсивом, заголовки розділів і підрозділів та підзаголовки всередині підрозділів виділено напівжирним шрифтом. Тексти програм виділено монотипічним шрифтом.

Для спрощення пошуку навчальних завдань, прикладів, контрольних питань і тестів для самоперевірки використовується така система умовних позначень:



– приклади,



– контрольні питання,



– практичні завдання, що мають бути виконані з використанням комп’ютера,



– завдання, що виконуються вручну (допускається набір у текстових редакторах на ЕОМ),



– літературні джерела для вивчення розділу.

РОЗДІЛ 1

ОСНОВНІ ПОНЯТТЯ ШТУЧНОГО ІНТЕЛЕКТУ

Штучний інтелект – це галузь комп’ютерних наук, що займається автоматизацією розумного поводження агентів, які одержують у результаті актів сприйняття інформацію про навколошне середовище і виконують дії, що реалізують функцію від результатів сприйняття і попередніх дій.

Метою штучного інтелекту є розробка комп’ютерних систем, що мають можливості, які традиційно пов’язуються з людським розумом: розуміння мови, навчання, здатність мркувати, вирішувати проблеми, планувати і т. д.

Основні напрями досліджень в галузі штучного інтелекту:

– подання знань і робота з ними – створення спеціалізованих моделей і мов для подання знань в ЕОМ, а також програмних і апаратних засобів для їхнього перетворення (поповнення, логічної обробки і т. д.);

– планування доцільного поводження – дослідження зі створення методів формування цілей і вирішення задач планування дій агента, що функціонує в складному зовнішньому середовищі;

– спілкування людини з ЕОМ – задачі створення мовних засобів, що дозволяють ефективно взаємодіяти з ЕОМ непрограмуючому користувачу. Ведуться дослідження в області синтаксису і семантики природних мов, способів збереження знань про мову в пам’яті машини і побудови спеціальних процесорів, що здійснюють переклад текстової інформації у внутрішнє машинне подання;

– розпізнавання образів і навчання – дослідження зі сприйняття зорової, слухової і інших видів інформації, методів її обробки, формування відповідних реакцій на впливи зовнішнього середовища і способів адаптації штучних систем до середовища шляхом навчання.

1.1 Інтелектуальні системи

Інтелектуальною системою називають кібернетичну систему, призначенну для вирішення інтелектуальних задач.

Інтелектуальна задача – це задача, точний алгоритмізований метод вирішення якої априорі є невідомий. При цьому під *рішенням задачі* розуміється будь-яка діяльність, пов’язана з розробкою планів і виконанням дій, необхідних для досягнення визначеної мети.

Властивість інтелектуальності, що істотно відрізняє інтелектуальні системи від інших кібернетичних систем, характеризується набором таких ознак.

1. Наявність у системі власної внутрішньої моделі зовнішнього світу, що забезпечує індивідуальність, самостійність системи в оцінці вхідного запиту, можливість значеннєвої (семантичної) і прагматичної інтерпретації запиту відповідно до власних знань і вироблення відповіді (реакції), семантично і прагматично правильної з точністю до адекватного моделювання зовнішнього світу (предметної області).

2. Здатність системи поповнювати наявні знання, засвоювати нові знання, навчатися, здійснюючи будовування нової інформації в систему подання знань.

3. Здатність системи виділяти значні якісні характеристики ситуації.

4. Здатність до дедуктивного виведення, генерації рішення, що у явному і готовому вигляді не міститься в самій системі.

5. Здатність до прийняття рішень на основі нечіткої, неточної, недостатньої або погано визначеної інформації і застосування формалізмів подань, що допомагають програмісту справлятися з циминедоліками.

6. Еволюційність і адаптивність моделей штучного інтелекту: набуття знань системою здійснюється за допомогою навчання, заснованого на адаптації (пристосуванні) її структури і параметрів у процесі еволюційного розвитку до умов зовнішнього середовища, що змінюються.

За сферою застосування інтелектуальні системи поділяють на системи загального призначення і спеціалізовані системи.

Інтелектуальні системи загального призначення – системи, що не тільки виконують задані процедури, але на основі метапроцедур пошуку генерують і виконують процедури рішення нових конкретних задач. Технологія використання таких систем полягає в наступному. Користувач (експерт) формує знання (дані і правила), що описують обране застосування. Потім на підставі цих знань, заданої мети і вихідних даних метапроцедури системи генерують і виконують процедуру вирішення конкретної задачі. Дану технологію називають *технологією систем, заснованих на знаннях*,

або технологією інженерії знань. Вона дозволяє фахівцю, що не знає програмування, розробляти гнучкі прикладні системи. Найбільше широко використовуваним типом інтелектуальних систем загального призначення є оболонки експертних систем.

Спеціалізовані інтелектуальні системи – вирішують фіксований набір задач, визначений при проектуванні системи. Для використання таких систем потрібно наповнити їх даними, що відповідають обраному застосуванню.

1.2 Проблемні області та їхні властивості

Проблемна область (предметна область) – сукупність взаємозалежних відомостей, необхідних і достатніх для вирішення даної інтелектуальної задачі. Знання про предметну область включають описи об'єктів, явищ, фактів, подій, а також відношень між ними.

Уявно предметна область складається з реальних або абстрактних об'єктів, що звуться *сутностями*. Сутності предметної області знаходяться у визначених *відношеннях* (асоціаціях) одна до одної, які також можна розглядати як сутності і включати в предметну область. Між сутностями спостерігаються різні *відношення подоби*. Сукупність подібних сутностей складає *клас сутностей*, що є новою сутністю предметної області.

Для вирішення інтелектуальних задач необхідно використовувати знання з конкретної предметної області, подані в певній стандартній формі і скласти програму їхньої обробки.

Класифікація проблемних середовищ може бути зроблена за такими вимірами.

– Середовище, яке повністю або частково спостерігається.

Повністю спостережене – середовище, у якому датчики агента надають йому доступ до повної інформації про стан середовища в кожний момент часу, необхідної для вибору агентом дії.

Частково спостережене – середовище, у якому через датчики, що створюють шум і є неточними, або через те, що окремі характеристики його стану просто відсутні в інформації, отриманій від датчиків, агент не може мати доступ до повної інформації про стан середовища в кожний момент часу.

– *Детерміноване* або *стохастичне середовище*.

Детерміноване – середовище, наступний стан якого цілком визначається поточним станом і дією, виконаною агентом.

Сттохастичне – середовище, наступний стан якого цілком не визначається поточним станом і дією, виконаною агентом.

– *Епізодичне* або *послідовне середовище*.

В *епізодичному середовищі* досвід агента складається з нерозривних епізодів, кожний з яких містить у собі сприйняття середовища агентом, а потім виконання однієї дії, при цьому наступний епізод не залежить від дій, виконаних у попередніх епізодах.

У *послідовному середовищі* поточне рішення може вплинути на всі майбутні рішення.

– *Статичне* або *динамічне середовище*.

Динамічне середовище для агента – середовище, що може змінитися в ході того, як агент обирає чергову дію.

Статичне середовище для агента – середовище, що не може змінитися в ході того, як агент обирає чергову дію. Статичність області означає незмінність вихідних даних, що її описують. При цьому похідні дані (виведені з вихідних) можуть і з'являтися заново, і змінюватися (не змінюючи, однак, вихідних даних).

Напівдинамічне середовище для агента – середовище, що з часом не змінюються, а змінюються тільки показники продуктивності агента в середовищі.

– *Дискретне* або *неперервне середовище*. Розходження між дискретними і неперервними варіантами середовища може відноситися до стану середовища, способу обліку часу, а також сприйняттям і діям агента.

– *Одноагентне середовище* (коли в середовищі діє тільки один агент) або *мультиагентне середовище* (коли діють два агенти і більше).

Крім того, предметні області можна характеризувати такими аспектами: числом і складністю сущностей; їхніх атрибути і значень атрибутів; зв'язністю сущностей та їхніх атрибутів; повнотою знань; точністю знань (знання точні або правдоподібні; правдоподібність знань подається певним числом або висловленням).

Задачі, що вирішуються інтелектуальними системами у проблемній області, класифікують:

– за ступенем зв'язності правил: *зв'язні* (задачі, що не вдається розбити на незалежні задачі) та *малозв'язні* (задачі, що вдається розбити на деяку кількість незалежних підзадач);

– з точки зору розробника: *статичні* (якщо процес вирішення задачі не змінює вихідні дані про поточний стан предметної області) і *динамічні* (якщо процес вирішення задачі змінює вихідні дані про поточний стан предметної області);

– за класом *вирішуваних задач*: розширення, довизначення, перетворення.

1. *Задачі розширення* – задачі, у процесі вирішення яких здійснюється тільки збільшення інформації про предметну область, що не призводить ні до зміни раніше виведених даних, ані до вибору іншого стану області. Типовою задачею цього класу є задача класифікації.

2. *Задачі довизначення* – задачі з неповною або неточною інформацією про реальну предметну область, мета вирішення яких – вибір з множини альтернативних поточних станів предметної області того, що є адекватний вихідним даним. У випадку неточних даних альтернативні поточні стани виникають як результат ненадійності даних і правил, що призводить до різноманіття різних доступних висновків з тих самих вихідних даних. У випадку неповних даних альтернативні стани є результатом довизначення області, тобто результатом припущені про можливі значення відсутніх даних.

3. *Задачі перетворення* – задачі, які здійснюють зміни вихідної або виведеної раніше інформації про предметну область, що є наслідком змін або реального світу, або його моделі.

Задачі розширення і довизначення є статичними, а задачі перетворення – динамічними.

1.3 Агентний підхід до інтелектуальних систем

Агент – це сутність для вирішення поставлених задач, що має цілеспрямоване поводження: взаємодіє з зовнішнім складним середовищем, що динамічно-розвивається, здатним модифікуватися у залежності від конкретних умов.

Взаємодія має на увазі сприйняття динаміки середовища; дії, що змінюють середовище; міркування з метою інтерпретації явищ, що спостерігаються, вирішення задач, виведення висновків і визначення дій.

Функція агента визначає дію, що починається агентом у відповідь на будь-яку послідовність актів сприйняття.

Показники продуктивності агента оцінюють поводження агента у середовищі. *Раціональний агент* діє так, щоб максимізувати очікувані

значення показників продуктивності, з урахуванням послідовності актів сприйняття, отриманої агентом до даного моменту.

Інтелектуальний агент – це агент, взаємодія якого з навколошнім середовищем є адекватним визначеній системі вимог:

- навчатися і розвиватися в процесі взаємодії з навколошнім середовищем;
- пристосовуватися у режимі реального часу;
- швидко навчатися на основі великого обсягу даних;
- покроково пристосовувати нові способи вирішення проблем;
- мати базу прикладів з можливістю її поповнення;
- мати параметри для моделювання швидкої і довгої пам'яті, віку і т. д.;
- аналізувати себе у термінах поводження, помилки й успіху.

Інтелектуальний агент, як показано на рис. 1.1, містить чотири концептуальні компоненти:

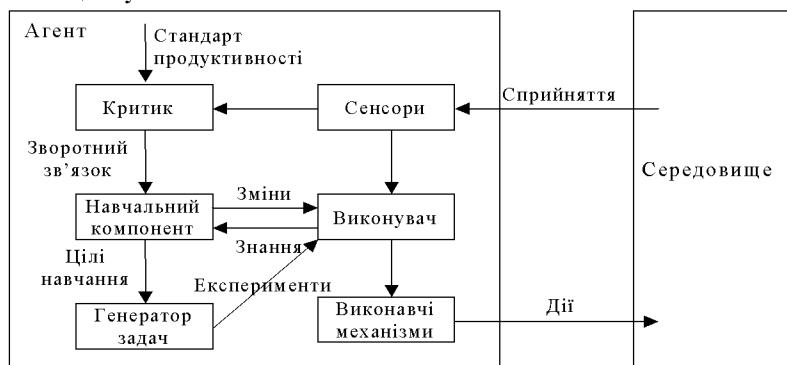


Рисунок 1.1 – Схема узагальненого інтелектуального агента

– *сенсори* – датчики, пристрой що дозволяють сприймати характеристики середовища;

– *критик* – компонент, що оцінює ефективність дій агента з урахуванням постійного *стандарту продуктивності* (системи вимог до функціонування агента);

– *виконавець* – забезпечує вибір зовнішніх дій: одержує сприйману інформацію і приймає рішення про виконання дій;

– *виконавчі механізми* – програмно-апаратні компоненти, що дозволяють здійснити дії у зовнішньому середовищі відповідно до прийнятих рішень виконавцем;

– навчальний компонент – відповідає за внесення удосконалень: використовує інформацію зворотного зв’язку від критика з оцінкою того, як діє агент, і визначає, яким чином повинен бути модифікований виконавець для того, щоб він більш успішно діяв у майбутньому;

– генератор задач – пропонує дії, що повинні привести до одержання нового й інформативного досвіду з метою вивчення середовища і функціонування агента в ній.

Процес навчання, здійснюваний в інтелектуальних агентах, можна в цілому охарактеризувати як процес модифікації кожного компонента агента для забезпечення більш точної відповідності цих компонентів доступної інформації зворотного зв’язку і тим самим поліпшення загальної продуктивності агента.

1.4 Історія розвитку штучного інтелекту

У період архаїки та античності з’являються перші ідеї створення штучних пристроїв, що здатні замінити людей, а також перші спроби категоризувати навколошній світ та формалізувати принципи людських міркувань. Найважливіші події цього періоду подані нижче.

Період	Події
ІІІ тисячоріччя до н. е.	Давньоєгипетський папірус (куплений у Луксорі Е. Смітом у 1882 р.) описує хірургічне обстеження у вигляді «симптом-діагноз-лікування-прогноз», комбінуючи конструкції «якщо, то» – перша відома база знань («експертна система»).
I тисячоріччя до н. е.	Створення механічних статуй у Єгипті та Греції, приблизно здатних до демонстрації «міркувань» і «емоцій». Давньогрецькі міфи про Гефеста і Пігмаліона, що містять ідеї інтелектуальних роботів (таких, як Талос) і штучних створень (Галатея і Пандора).
940 р. до н. е.	Ян Ши у Давньому Китаї (Yan Shi) розробив механічну «плодину».
384-322 р. до н. е.	Аристотель описав силогізм – метод формальних механічних міркувань.
10 р. н. е.	Герон Олександрийський створив автоматичні машини, включаючи механічну «плодину».
260 р. н. е.	Порфирій Тирський написав працю «Isagogē» про категоризацію знань і логіку.

У період середньовіччя значно поширюються ідеї створення штучних людей та здійснюються спроби створення механічних автоматів. Найважливіші події цього періоду подані нижче.

Рік	Події
~800	Гебер (Geber) розробив алхімічну теорію створення життя в лабораторії.
1206	Аль-Джазарі (Al-Jazari) створив програмований оркестр механічних «людій».
1275	Р. Ллуль (R. Llull) створив механічний пристрій, здатний генерувати ідеї (комбінувати концепти).
~1500	Парацельс (Paracelsus) висунув ідею створення штучної людини.
~1580	Рабин Й. Лоев (Judah ben Bezalel Loew) – ідея створення штучного людиноподібного захисника Голема (Golem).

У новий час зароджуються механістичні уявлення про будову людини, а також закладаються основи сучасної науки. Найважливіші події цього періоду подані нижче.

Рік	Події
1626	Р. Декарт (R. Descartes) висунув припущення, що тіла тварин – не більше, ніж складні машини, а також по-суті окреслив поняття рефлексу.
1651	Т. Гоббс (T. Hobbes) висунув механістичну комбінаторну теорію пізнання. Він припустив, що люди спільно за допомогою організації і машин можуть створити новий інтелект.
1672 – 1694	М. Лейбніц (G. Leibniz) створив вирішальний калькулятор – механічний пристрій для інтерпретації й обчислення концептів, а також запропонував дводжому систему для універсальних міркувань.
1726	Дж. Свіфт (J. Swift) висунув ідею автоматичного написання книг.
1750	Ж. де Ламетрі (J. de La Mettrie) висунув ідею про те, що людські міркування – строго механічні.
1763	Т. Байес (T. Bayes) висунув теорему про імовірності.
1769	В. фон Кемпелен (W. von Kempelen) побудував автомат для гри в шахи.
1805	Ж. Жаккард (J. Jacquard) винайшов перший програмований пристрій.
1822	Ч. Баббідж (Ch. Babbage) і А. Лавлейс (A. Lovelace) працювали над програмувальними механічними обчислювальними машинами.
1859	
1837	Б. Больцано (B. Bolzano) зробив спробу формалізувати семантику.
1847	Дж. Буль (G. Boole) розробив математичну символічну логіку (алгебру Буля) для міркувань про категорії об'єктів.
1863	С. Батлер (S. Butler) запропонував використовувати теорію еволюціїDarвіна стосовно до машин.
1879	М. Фреже (G. Frege) розробив логіку предикатів для доказу загальних теорем за правилами.
19 ст.	Ч. Тремо (C. Trémaux) розробив метод пописку у глибину.
1898	Е. Тхорндік (E. Thorndike) висунув теорію бхевіоризму: усі дії, думки або бажання викликані рефлексивно стимулами, разом – пасивний асоціативний механізм.
1914	Е. Голдберг (E. Goldberg) розробив машину, здатну читувати символи та перетворювати їх у телеграфний код.
1915	Л. Торрес (L. Torres) створив автомат для гри в шахи.

Новітній час ознаменувався власне виникненням штучного інтелекту як області комп’ютерних наук. Інтелектуальні методи і технології одержали впровадження в задачах аналізу даних, інтелектуалізації людино-машинного інтерфейсу, планування, керування. Найважливіші події цього періоду подані нижче.

Рік	Події
1920 – 1930 рр.	Л. Вітгенштейн (L. Wittgenstein) і Р. Карнар (R. Carnap) зводять філософію до логічного аналізу знань.
1921	К. Чапек (K. Čapek) винайшов термін «робот» для опису інтелектуальної машини, що повстасе проти людини.
1928	Дж. фон Нейман (J. von Neumann) запропонував min-max теорему, що стала основою для ігрових програм.
1931	К. Гёдель (K. Gödel) продемонстрував, що відомі вірні математичні теореми є недовідними, тобто людина може розпізнавати правильне значення деяких пропозицій, але їхня істинність не може бути витягнута з якої-небудь логічної системи.
1936	Р. Фішер (R. Fisher) заклав основи дискримінантного аналізу
1937	А. Тьюрінг (A. Turing) запропонував ідею універсальної машини, що може імітувати роботу будь-якої іншої обчислювальної машини. Але він також визнав, що існують визначені види розрахунків, які жодна машина не може виконати. А. Тьюрінг (A. Turing) і А. Чърч (A. Church) висунули припущення, що всі проблеми, що може вирішувати людина, могуть бути зведені до набору алгоритмів.
1940	Дж. фон Нейман (J. von Neumann) зазначив відмінність між даними та інструкціями.
1941	К. Цузе (K. Zuse) побудував перший прапоочий програмно-керований комп'ютер.
1943	У. Піттс (W. Pitts) та У. МакКаллох (W. McCulloch) розробили модель штучного нейрона.
1944	Дж. фон Нейман (J. von Neumann) і О. Моргенштерн (O. Morgenstern) заснували теорію ігор і розвили мінімаксний аналіз.
1947	Дж. фон Нейман (J. von Neumann) запропонував автомати, що самовідврояться.
1948	Н. Віннер (N. Wiener) розробив теорію кібернетики – науки про керування та комунікації у тварині та машині.
1949	Д. Хебб (D. Hebb) запропонував принципи навчання нейронних мереж.
1949	У. Вівер (W. Weaver) виділив напрям «машинний переклад».
1949	У. Грей-Уолтер (W. Grey-Walter) розробив роботів Elmer та Elsie.
1950	А. Тьюрінг (A. Turing) запропонував тест для перевірки машинного інтелекту.
1950	І. Азімов (I. Asimov) опублікував три закони робототехніки.
1950	К. Шеннон (C. Shannon) опублікував детальний аналіз гри у шахи як попук.
1950	М. Маастерман (M. Masterman) розробила семантичні мережі для машинного перекладу.
1951	С. Бар-Гілель (Y. Bar-Hillel) – перші дослідження з машинного перекладу.
1951	М. Мінський (M. Minsky) та Д. Едмондс (D. Edmonds) побудували першу штучну нейронну мережу, що моделювала попук пляху пацюком у лабіринті.
1951	Створено перші програми для гри у шашки (К. Страчей – C. Strachey) та гри у шахи (Д. Принц – D. Prinz).
1952	К. Шеннон (C. Shannon) створив механічну мишу «Theseus», здатну навчатися навігації у лабіринті.
1954	Й. Енгельбергер (J. Engelberger) розробив промислового робота Unimate.

Рік	Події
1954	Л. Достерт (L. Dostert) розробив систему машинного перекладу.
1954	Створено перші програми з машинного перекладу у СРСР та Японії.
1955	Вперше вжито термін «штучний інтелект».
1956	А. Ньюелл (A. Newell), Дж. Шоу (J. Shaw) та Г. Саймон (H. Simon) продемонстрували програму Logic Theorist – одну з перших програм штучного інтелекту.
1957	Г. Гелернте (H. Gelernter) та Н. Рочестер (N. Rochester) написали Geometry Theorem Prover – програму для доказу теорем з геометрії.
1957	А. Ньюелл (A. Newell) та Г. Саймон (H. Simon) розробили програму General Problem Solver, засновану на аналізі засобів досягнення цілей та спрямовану на скорочення різниці між прогнозуванням і бажаним результатами шляхом зміни контролюючих факторів.
1957	А. Семюел (A. Samuel) написав програму для гри у шашки.
1957	Е. Фейгенбаум розробив ЕРАМ (елементарний пристрій розпізнавання і запам'ятовування) – модель того, як люди запам'ятовують безглазді склади.
1957	Н. Хомський (N. Chomsky) розробив граматику перетворень.
1957	Р. Соломонов (R. Solomonoff) створив концепцію машинного навчання.
1958	Дж. МакКарті (J. McCarthy) розробив мову програмування LISP.
1958	Ф. Розенблatt (F. Rosenblatt) створив персепtron – нейронну мережу, здатну навчатися.
1959	Д. Хьюбель (D. Hubel) та Т. Візель (T. Wiesel) дослідили роботу зорової кори та показали, що окрім нейронів відповідають за стимули, що відбиваються певною рецепторною зоною.
1960	Б. Уідроу (B. Widrow) розробив нейронну мережу Adaline.
1961	М. М. Бонгард розробив програму розпізнавання образів «Кора».
1962	В. М. Глупков, В. А. Ковалевський та В. І. Рибак розробили комплекс для розпізнавання тексту.
1963	Л. Ухр (L. Uhr) та Ч. Восслер (C. Vossler) розробили програму для розпізнавання образів, здатну автоматично обирати ознаки та навчатися.
1964	Б. Рафаель (B. Raphael) розробив програму SIR – систему відповідей на запитання на основі логічного подання знань.
1964	Д. Бобров (D. Bobrow) показав, що комп'ютери можуть розуміти природну мову.
1964	Компанія IBM розробила програму для розпізнавання мови «Shoebox».
1965	Дж. Вейzenbaum (J. Weizenbaum) створив інтерактивну програму підтримки діалогу (спілкування) ELIZA.
1965	Дж. Робісон (J. Robinson) запропонував процедуру механічного доведення.
1965	Е. Фейгенбаум (E. Feigenbaum) та Р. Ліндсей (R. Lindsay) створили першу експертну систему DENDRAL.
1965	Л. Заде (L. Zadeh) розробив основи нечіткої логіки.
1966	Р. Квілліан (R. Quillian) продемонстрував семантичні мережі.
1966	Ю. І. Журавльов розробив метод АВО.
1967	Б. Хейс-Рот (B. Hayes-Roth) розробила систему розуміння мови Hearsay.
1967	Т. К. Вінцюк запропонував генеративну модель розпізнавання образів (Dynamic Time Warping – DTW), що знайшла застосування в теорії розпізнавання мовленнєвих і зорових образів, а також в моделюванні нелінійних процесів у радіофізиці та в біоінформатиці

Рік	Події
1968	Д. Бултон (D. Boulton) та К. Уоллес (C. Wallace) розробили програму Snob для класифікації без учителя.
1968	Ж. Мозес (J. Moses) розробив програму Macsyma, засновану на знаннях для символічних міркувань у математиці.
1968	О. Г. Івахненко створив метод групового урахування аргументів.
1968	П. Тома (P. Toma) розробив програму машинного перекладу Systran.
1968	П. Харт (P. Hart), Н. Нільсон (N. Nilsson) та Б. Рафаель (B. Raphael) розробили метод А*.
1968	Р. Грінблatt (R. Greenblatt) створив програму для гри у шахи MacHack, засновану на знаннях.
1968	Т. Виноград (T. Winograd) створив програму SHRDLU, що керувала роботом-руковою в обмеженому світі дитячих кубиків за допомогою інструкцій надрукованих англійською мовою.
1968	Я. З. Ципкін започаткував теорію адаптивних систем.
1969	Й. Уілкс (Y. Wilks) розробив подання семантичної узгодженості мови, втілене у першій програмі машинного перекладу, керованій семантикою.
1969	М. Мінський (M. Minsky) та С. Пайперт (S. Papert) описали недоліки двопарових персепtronів, чим сприяли зменшенню фінансування досліджень у галузі штучного інтелекту.
1969	Р. Шенк (R. Schank) створив модель концептуальної залежності для розуміння природної мови.
1969	У Стенфордському інституті розроблено робота Shakey, здатного комбінувати сприйняття, рух та вирішення проблем.
1970	Б. Вудс (B. Woods) розробив подання знань для розуміння природної мови – розширену мережу переходів (<i>augmented transition network</i>).
1970	Ж. Карбонел (J. Carbonell) розробив інтерактивну програму SCHOLAR для навчання за допомогою комп'ютера на основі семантичних мереж.
1970	М. А. Айзерман, Е. М. Браверман, Л. Й. Розоноер запропонували метод потенційних функцій.
1970	П. Уінston (P. Winston) розробив програму ARCH, здатну вивчати концепти з прикладів.
1971	А. Колмерое (A. Colmerauer) та Ф. Русセル (Roussel) створили комп'ютерну мову PROLOG.
1971	І. Рехенберг (I. Rechenberg) запропонував еволюційні стратегії як оптимізаційні методи.
1971	Р. Файк (R. Fikes) та Н. Нільсон (N. Nilsson) розробили систему автоматичного планування STRIPS.
1972	В. А. Ковалевський, М. І. Шлезінгер створили систему розпізнавання тексту «ЧАРС».
1972	Е. Шорліфф (E. Shorliffe) створив медичну експертну систему MYCIN.
1972	І. Сандерлоті (E. Sacerdoti) розробив одну з перших ієрархічних програм для планування ABSTRIPS.
1973	В університеті Единбурга створено робота Freddy, здатного використовувати візуальне сприйняття для знаходження та збирання моделей.

Рік	Події
1974	П. Вербос (P. Werbos) розробив метод зворотного поширення помилки для навчання нейронних мереж.
1975	Дж. Холланд (J. Holland) запропонував генетичні алгоритми.
1975	М. Мінський (M. Minsky) створив апарат фреймів.
1975	Програма Meta-Dendral відкрила нові знання з хімії.
1979	Г. Моравець (H. Moravec) розробив перший автономний транспортний засіб, керований комп'ютером, Stanford Cart.
1979	Д. МакДермотт (D. McDermott), Дж. Дойл (J. Doyle) та Дж. МакКарти (J. McCarthy) розробили немонотонну логіку та формальні аспекти підприємки істинності.
1979	Д. Марр (D. Marr) розробив теорію бачення.
1979	Дж. Майєрс (J. Myers) та Г. Попле (H. Pople) розробили медичну діагностичну систему INTERNIST, засновану на знаннях.
1979	Комп'ютерна програма BKG Г. Берлінера (H. Berliner) перемогла чемпіона світу з гри у народі Л. Віллу.
1979	Н. М. Амосов проводить дослідження з розробки біологічних нейронних мереж.
1980	Дж. МакДермотт (J. McDermott) розробив експертну систему Xcon для допомоги у підборі компонентів ЕОМ.
1980	К. Фукушима (K. Fukushima) розробив нейронну мережу неокогніtron.
1980	Перше промислове застосування нечіткого контролера виробником цементу F. L. Smidth & Co.
1981	Д. Хілліс (D. Hillis) розробив коннекціоністську машину.
1982	Дж. Хопфілд (J. Hopfield) розробив нейронні мережі Хопфілда, засновані на моделюванні віддалу.
1982	Т. Кохонен (T. Kohonen) розробив карти, що самоорганізуються, (Self-Organized Maps – SOM) для навчання без учителя.
1983	Д. Аллен (J. Allen) розробив інтервальні обчисlenня, які уперше використані для формалізації подій у часі.
1983	Д. Хінтон (G. Hinton) та Т. Сейновські (T. Sejnowski) розробили машину Больцмана для навчання без учителя.
1984	Компанія General Electrics створила експертну систему для діагностики локомотивів.
1985	Ю. Перл (J. Pearl) розробив мережі Байсса.
1987	Б. Коско (B. Kosko) розробив модель асоціативної пам'яті
1987	Е. Дікманнс (E. Dickmanns) створив перший автомобіль-робот, що пересувався по порожніх вулицях.
1987	М. Мінський (M. Minsky) розробив опис розуму як співтовариства взаємодіючих агентів.
1987	С. Гросберг (S. Grossberg) створив теорію адаптивного резонансу (Adaptive Resonance Theory – ART) для навчання без учителя.
1988	Д. Брумхед (D. Broomhead) та Д. Лоуе (D. Lowe) запропонували штучну нейронну мережу радіальних базисних функцій.
1989	Д. Померле (D. Pomerleau) створив автономний автомобіль, керований на основі нейронної мережі ALVINN.

Рік	Події
1989	Д. Янг розробив систему оптичного розпізнавання тексту Finereader.
1990	Дж. Ельман (J. Elman) запропонував нейронну мережу Ельмана.
1990	К. Мід (C. Mead) описав нейроморфний процесор.
1991	Програма для планування DART за період війни у Перській затоці окупила інвестиції США у штучний інтелект за 30 попередніх років.
1991	Створено машинний перекладач PROMT (з. 1992 – STYLUS).
1991	У Харківському державному університеті розроблено першу комерційну програму з машинного перекладу російська-англійська-українська.
1992	Т. Рей (T. Ray) розробив віртуальний світ «Тетта».
1993	Я. Хорсвілл (I. Horswill) розробив робота Polly, здатного до навігації на основі зору зі швидкістю тварини.
1994	Автономні роботи-автомобілі VaMP і VITA-2 з пасажирами на борту проїхали більш тисячі кілометрів по швидкісному шосе.
1995	В. Н. Вапник розробив машину опорних векторів (Support Vector Machine – SVM).
1995	Дж. Хінтон (G. Hinton) розробив машину Гельмгольца.
1997	Проведено перший офіційний футбольний чемпіонат роботів RoboCup.
1997	Суперкомп'ютер IBM Deep Blue переміг чемпіона світу з грі у шахи Г. Каєпрова.
1998	Компанія Tiger Electronics розробила домашнього робота Furby.
1999	Інтелектуальна система Remote Agent уперше застосована для повного управління космічним кораблем.
1999	Компанія Sony представляє домашнього робота. Робот АІВО стає одним з перших автономних роботів «домашніх тварин».
2000	Інтерактивні інтелектуальні іграшки (роботи-тварини) стали комерційно доступні.
2000	Робот Nomad досліджує регіони Антарктики.
2000	С. Брезіль (C. Breazeal) висунула ідею емоційного робота «Kismet» з обличчям, здатним виражати емоції.
2001	Н. Хансен (N. Hansen) запропонував еволюційну стратегію «адаптація коваріаційної матриці» (Covariance Matrix Adaptation) для чисельної оптимізації.
2004	М. Тільден (M. Tilden) створив біоморфного робота Robosapien.
2004	Розроблено мову веб-онтології OWL Web Ontology.
2005	Компанія Honda розробила інтелектуального робота ASIMO, що зданий пересуватися як людина.
2005	Почато проект Blue Brain для емуляції мозку на молекулярному рівні.
2006	Дж. Хінтон (G. Hinton) розробив швидкий алгоритм навчання для машини Больцмана.
2007	Компанія Google запровадила власну технологію машинного перекладу Google Translate.
2008	Компанія IBM почала розробку нейроморфного процесора.
2010	Система NELL запущена в університеті Карнегі-Меллон для демонстрації машинного навчання семантиці.
2011	Н. д'Алоїсіо (N. D'Aloisio) випустив узагальнюючий засіб Trinit для смартфонів.
2011	О. Хасегава (O. Hasegawa) створив робота, що навчається функціям, які не були запрограмовані.
2011	Суперкомп'ютер IBM «Watson» переміг людину у грі Jeopardy на телебаченні.
2012	Р. Брукс (R. Brooks) створив програмованого робота «Baxter».



1.5 Контрольні питання

1. Що таке штучний інтелект, яка в нього мета?
2. Які основні напрями досліджень в галузі штучного інтелекту ви знаєте?
3. Що таке інтелектуальна система, інтелектуальна задача?
4. Що таке властивість інтелектуальності?
5. Що таке інтелектуальні системи загального призначення?
6. Що таке спеціалізовані інтелектуальні системи?
7. Які ви знаєте проблемні області та їхні властивості?
8. Класифікація проблемних середовищ.
9. Класифікація задач, що вирішуються інтелектуальними системами.
10. Агент.
11. Інтелектуальний агент.
12. Основні компоненти інтелектуального агента.
13. Основні етапи історії інтелектуальних систем.

1.6. Практичні завдання

- Завдання 1.** Написати реферат на одну з тем.
1. Системи розпізнавання тексту
 2. Системи розпізнавання мови.
 3. Системи розпізнавання музичних фрагментів.
 4. Системи інтелектуального діагностування у техніці.
 5. Системи інтелектуального діагностування в медицині.
 6. Інтелектуальні системи в економіці.
 7. Інтелектуальні системи в юриспруденції.
 8. Інтелектуальні системи машинного перекладу.
 9. Інтелектуальні системи пошуку.
 10. Інтелектуальні системи планування.
 11. Інтелектуальні системи проектування.
 12. Експертні системи у медицині.
 13. Експертні системи у техніці.
- Завдання 2.** Провести дослідження літератури з історії штучного інтелекту та підготувати реферат на одну з тем.
1. Сучасні досягнення та тенденції розвитку систем розпізнавання образів.

2. Сучасні досягнення та тенденції розвитку систем, що засновані на знаннях.
3. Історія виникнення передумов для створення інтелектуальних систем.
4. Історія виникнення і розвитку систем штучного інтелекту у 40-х – 60-х роках ХХ сторіччя.
5. Історія розвитку систем штучного інтелекту у 70-х роках ХХ сторіччя.
6. Історія розвитку систем штучного інтелекту у 80-х роках ХХ сторіччя.
7. Історія розвитку систем штучного інтелекту у 90-х роках ХХ сторіччя.
8. Історія розвитку систем штучного інтелекту на початку ХХІ сторіччя.
9. Історія, сучасний стан і перспективи розвитку штучного інтелекту в Україні.



1.7 Література до розділу

Загальні питання з теорії інтелектуальних систем наведено в [1–6, 10, 15].

Агентному підходу присвячено [2, 4, 17].

Питання історії штучного інтелекту висвітлюються в [2, 4].

РОЗДІЛ 2

РОЗПІЗНАВАННЯ ОБРАЗІВ

Для побудови сенсорних (розвізнавальних) інтелектуальних систем використовують як базис теорію розпізнавання образів, яка містить методи побудови моделей прийняття рішень за прецедентами, що характеризуються ознаками.

Цей розділ присвячено розгляду основних питань теорії розпізнавання образів та аналізу найбільш відомих методів розпізнавання.

2.1 Задачі розпізнавання образів. Навчання з учителем

Генеральною сукупністю будемо називати множину всіх екземплярів (об'єктів) визначеного типу. Екземпляри характеризуються значеннями *ознак* (атрибутив, властивостей).

Вибіркою будемо називати кінцеву підмножину екземплярів визначеного типу. Вона витягається з генеральної сукупності.

Нехай задана вибірка спостережень (екземплярів, прецедентів) $\langle x, y \rangle$, що характеризуються набором N ознак $\{x_j\}$, $j = 1, 2, \dots, N$, де N – кількість ознак. Вибірка $\langle x, y \rangle$ складається з S екземплярів: $x = \{x^s\}, s = 1, 2, \dots, S$, кожний з яких характеризується набором значень ознак $x^s = \{x_j^s\}$, де x_j^s – значення j -ї вхідної (описової) ознаки s -го екземпляра вибірки. При цьому кожному екземпляру також зіставлене значення вихідної (цільової) ознаки y , множину значень якої для екземплярів вибірки позначимо $y = \{y^s\}$.

Задача апроксимації залежності полягає у визначенні для розпізнаваного екземпляра x^s розрахункового значення вихідної ознаки y^s на основі моделі залежності $y = f(x)$.

Якщо вихідна ознака є неперервною, то говорять про *задачу оцінювання*, а якщо дискретною, то говорять про *задачу класифікації (розвізнавання образів)*. У задачі класифікації розпізнаваний екземпляр x^s необхідно віднести до одного з класів на основі побудованої розпізнавальної моделі $f(x)$.

Класом будемо називати множину екземплярів, об'єднаних загальними властивостями. У просторі ознак екземпляри різних класів можуть розташовуватися по-різному в різних задачах. Найкращою ситуацією є та, де класи є *компактними*, тобто екземпляри кожного класу розташовані в окремій області простору ознак і не змішуються між собою. Така ситуація зустрічається

рідко на практиці, однак ряд методів розпізнавання виходять з неї, і тоді говорять про те, що метод заснований на *гіпотезі про компактність класів*.

Навчальною вибіркою будемо називати ту вибірку, на основі якої будується розпізнавальна модель.

Важливою властивістю навчальної вибірки є її *репрезентативність* щодо генеральної сукупності. Вибірка є репрезентативною щодо генеральної сукупності, якщо вона містить усі типові випадки (види екземплярів), подані у генеральній сукупності, та їхні частоти у вибірці близькі до частот у генеральній сукупності.

Тестовою (перевірочною) вибіркою будемо називати ту вибірку, що використовується для перевірки працевздатності розпізнавальної моделі.

У загальному випадку навчальна і тестова вибірки не повинні збігатися. Якщо як тестову вибірку використовувати навчальну вибірку, то оцінки критерію якості моделі, отримані по ній, можуть виявитися надмірно завищеними.

Задача побудови розпізнавальної моделі (*задача навчання з учителем*, *задача структурно-параметричного синтезу, ідентифікації*) на основі заданої навчальної вибірки $\langle x, y \rangle$, що описує залежність $y=f(w, x)$, полягає в знаходженні (ідентифікації) такої її структури f (*задача структурного синтезу розпізнавальної моделі*) і таких значень її параметрів w (*задача параметричного синтезу розпізнавальної моделі*), що забезпечують оптимальне значення заданого функціонала якості розпізнавальної моделі $Q_m(f(w, x))$.

У випадку, коли не відомі ані структура залежності f , ані значення її параметрів w , говорять про побудову моделі типу «*чорна скриня*».

У випадку, коли відома структура залежності f і значення її параметрів w , говорять про побудову моделі типу «*біла скриня*».

У випадку, коли відома або структура залежності f , або значення її параметрів w , говорять про побудову моделі типу «*сіра скриня*».

Як критерій оптимальності $Q_m(f(w, x))$ традиційно використовують функції помилки E :

– для випадку, коли вихідна ознака є бінарною (число класів дорівнює двом):

$$E = \sum_{s=1}^S |y^s - f(x^s)| \rightarrow \min,$$

– для випадку, коли вихідна ознака є дискретною:

$$E = \sum_{s=1}^S \{1 | y^s \neq f(x^s)\} \rightarrow \min.$$

При цьому для попереднього і даного випадків також можна оцінити імовірність прийняття помилкових рішень: $P_{\text{пом.}} = E/S$ і імовірність прийняття правильних рішень $P_{\text{пр.}} = 1 - P_{\text{пом.}}$. Помітимо, що дані оцінки будуть відрізнятися для навчальної і тестової вибірок;

– для загального випадку (для дискретних і неперервних виходів):

$$E = \frac{1}{2} \sum_{s=1}^S (y^s - f(x^s))^2 \rightarrow \min.$$

Для оцінки заданого показника якості побудованої моделі Q_m застосовуються експериментальні способи:

– розбиття вихідної вибірки на навчальну (за якою далі будуєть модель) і контрольну (тестову, перевірочну – за якою роблять перевірку моделі);

– одночасне використання однієї і тієї ж вибірки в якості навчальної і контрольної (дає завищену оцінку якості розпізнавання в порівнянні з тією же оцінкою якості за незалежними від навчання даними);

– метод ковзного інституту (багаторазовий послідовний або випадковий витяг з вихідної вибірки об'єктів по одному або групами як контрольних вибірок і побудова за об'єктами, що залишилися, моделей, на основі яких здійснюється розпізнавання контрольних вибірок; дає найменшу дисперсію помилки, але є обчислювально складним).

2.2 Лінійна роздільність та нелінійна роздільність класів

Два класи у просторі ознак називають лінійно роздільними, якщо можна екземпляри цих класів розділити однією гіперплощиною (у двовимірному просторі ознак – прямою лінією) так, що по один бік від неї будуть розташовані екземпляри тільки одного класу, а по другий бік – іншого класу. Якщо класи не є лінійно роздільними, то їх називають нелінійно роздільними.

Приклад лінійно роздільних класів зображенено на рис. 2.1 *a*, а нелінійно роздільних класів – на рис. 2.1 *b*. Тут чорними кругами позначено екземпляри одного класу, а білими – другого.

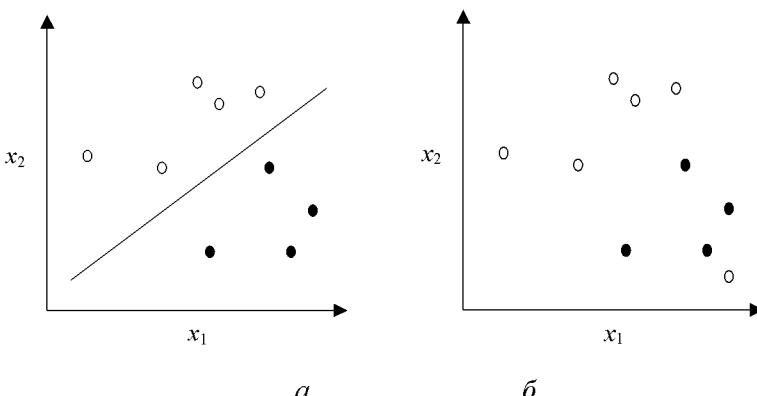


Рисунок 2.1 – Лінійна роздільність (*а*) та лінійна нероздільність (*б*) класів

Для того, щоб розділити нелінійно роздільні класи потрібно або використати комбінацію декількох гіперплощин (прямих ліній у проекціях), або використати нелінійну поверхню (криву лінію у проекціях).

2.3 Класифікація і порівняльна характеристика методів розпізнавання образів

Класифікація методів розпізнавання образів можлива за різними критеріями. Методи розпізнавання образів виділяють:

- *за типом одержуваної моделі*: *якісні* (дозволяють одержувати одне з кінцевої множини рішень або оперують дискретними даними) і *кількісні* (дозволяють одержувати кількісну оцінку параметра);
- *у залежності від виду використовуваних знань*: *параметричні* (використовують щільності розподілів ознак, їхнє застосування в реальних задачах пов’язано з накладенням сильних обмежень на структуру даних, що приводять до лінійних моделей з дуже приблизними оцінками їхніх параметрів), *непараметричні* (аналізують відносні кількості об’єктів, що попадають у задані багатомірні обсяги, і використовують різні функції відстані між екземплярами навчальної вибірки і розпізнаваних екземплярів, застосовуються тоді, коли вид кривої щільності розподілу невідомий і не можна зробити ніяких припущень про її характер) і

евристичні (ґрунтуються на знаннях, що важко формалізуються, та інтуїції дослідника, що визначає, яку інформацію і який образ потрібно використовувати для досягнення необхідного ефекту розпізнавання);

– за *специфікою способу подання знань про предметну область*: *інтенсіональні* (засновані на операціях з ознаками, як елементи яких вони використовують різні характеристики ознак і їхніх зв'язків, фіксують закономірності і зв'язки, які визначають структуру даних в аналітичній чи конструктивній формі, не припускають виконання операцій над конкретними об'єктами, що виступають лише в ролі індикаторів для оцінки взаємодії і поводження своїх ознак) і *екстенсіональні* (засновані на операціях з конкретними екземплярами, кожному з яких додають самостійне значення, використовують як основні операції визначення подібності і розходження екземплярів, за якими і розрізняються методи даної групи, іхне застосування не пов'язане з певними припущеннями про структуру експериментальної інформації, крім того, що екземпляри одного класу повинні бути чимось схожі, а екземпляри різних класів повинні чимось відрізнятися один від одного).

Основні групи методів розпізнавання образів виділяють такі.

Статистичні методи ґрунтуються на оцінках щільностей розподілу значень ознак.

Застосовність: задачі з відомими розподілами ознак, (як правило, нормальним).

Незастосовність: задачі з невідомими розподілами ознак, необхідність пояснення рішень.

Переваги: можливість одержання оптимальної моделі.

Недоліки: необхідність набору великої статистики для апроксимації щільностей розподілів і необхідність перебору усієї навчальної вибірки при прийнятті рішень, висока чутливість до непрепрезентативності навчальної вибірки й артефактам.

До статистичних методів відносяться методи регресійного аналізу і методи статистичних рішень.

Методи регресійного аналізу дозволяють моделювати залежності за вимірюваними даними, що складаються з пар значень залежної змінної (змінної відгуку) і незалежних змінних (пояснюючих змінних) при який модель є функцією випадкової вели-

чини, незалежних змінних і параметрів, що налагоджуються таким чином, що модель щонайкраще наближає дані, а критерієм якості наближення (цільовою функцією) звичайно є середньоквадратична помилка. Для оцінки параметрів одновимірних лінійних залежностей використовують неітеративний метод найменших квадратів, для випадку багатомірної лінійної регресії вирішують систему лінійних рівнянь зі звертанням матриці; для випадку багатовимірної нелінійної регресії вирішують задачу мінімізації середньоквадратичної помилки регресійних залишків на основі градієнтних методів оптимізації.

Застосовність: великі обсяги безперервних даних.

Незастосовність: істотно нелінійні задачі високої розмірності з малим обсягом даних, якщо потрібно пояснення прийнятих рішень, символільні дані.

Переваги: простота одержуваних моделей.

Недоліки: необхідність перевірки гіпотези породження даних (визначає характер розподілу випадкової величини, реалізується на основі статистичних тестів, називаних аналізом залишків), висока чутливість до помилок і погрішностей у даних, для одномірної лінійної регресії – придатність тільки для простих задач, для багатовимірної лінійної регресії – необхідність нормування ознак і можливість виродження матриці, що звертається, для багатомірної нелінійної регресії - невизначеність у виборі структури моделі і початкових значень її параметрів, а також локальний характер пошуку градієнтних методів.

Методи статистичних рішень використовують припущення про те, що існує багатомірна функція щільності імовірності, яка характеризує кожен клас, і розглядають об'єкти як реалізації багатовимірної випадкової величини, розподіленої в просторі ознак за визначенним законом. Такі теоретичні передумови ведуть до діапазону стратегій класифікації від випадку повного знання ап-ріорних розподілів класів до повного їхнього незнання, крім тих розподілів, що можуть бути виведені з вибірок (непараметричний випадок). Вирішальне правило обирається виходячи з деяких умов (критерію) оптимальності.

Застосовність: великі обсяги даних, що добре розуміються, з добре сформульованими гіпотезами.

Незастосовність: розвідницький аналіз даних із залежними змінними.

Переваги: можливість одночасного урахування ознак різної фізичної природи, оскільки вони характеризуються безрозмірними величинами – ймовірностями їхньої появи при різних станах системи.

Недоліки: суворо вимагають виконання ряду припущенень, що часто не беруть до уваги кінцеві користувачі (наприклад, незалежність ознак), труднощі урахування загальних знань, проблема узагальнення на багатовимірний випадок

Методи поділу у просторі ознак засновані на гіпотезі про компактність класів: точки (екземпляри) одного класу групуються в одній області простору ознак.

Застосовність: там, де потрібно узагальнення набору прецедентів великого розміру.

Незастосовність: складні символільні дані, коли потрібно пояснення прийнятих рішень.

Переваги: добре розроблений математичний апарат і геометрична природа методів.

Недоліки: необхідність перетворення простору ознак для спрощення поділу класів, недоведеність гіпотези про компактність, складність сприйняття й аналізу одержуваної моделі людиною, складність урахування експертних знань при побудові моделі.

До методів поділу у просторі ознак відносять метод дискримінантних функцій, метод потенційних функцій і метод потенціалів.

Метод дискримінантних функцій ставить задачею знаходження за навчальною вибіркою функцій, що моделює поверхню, яка щонайкраще розділяє (розмежовує) у багатовимірному просторі ознак класи. Розпізнавання здійснюють шляхом підстановки значень ознак об'єкта у модель і визначення, з якого боку поділяючої поверхні знаходиться об'єкт, тобто до якого класу він належить.

Застосовність: задачі невеликої розмірності; для складно структурованих і символічних даних.

Незастосовність: для лінійних методів – для нелінійно роздільних класів.

Переваги: використовує максимум додаткової інформації і забезпечує високу надійність і відносну простоту класифікації у випадку невеликої кількості лінійно роздільних класів.

Недоліки: складність реалізації для великої кількості класів і ознак.

Метод потенційних функцій є розвитком ідеї перетворення простору ознак і базується на гіпотезі про характер функцій, що розділяють множини, які відповідають різним образам (класам). Як дискриміантні обираються функції, що мають найбільше значення для точок області діагнозу й спадні у міру видалення від неї. Метод визначає близькість розпізнаваного об'єкта до об'єктів навчальної вибірки; при цьому вважається, що об'єкти з навчальної вибірки «заряджені» своїм класом, а міра важливості кожного з них при класифікації залежить від його заряду і відстані до класифікованого об'єкта. Функцію, що описує розподіл потенціалу, можна використовувати як вирішальне правило (або для його побудови). Побудова поділяючої функції здійснюється за допомогою ітеративної процедури, що самонавчається.

Застосовність: для складно структурованих і символічних даних.

Незастосовність: якщо потрібно пояснення прийнятого рішення й узагальнення.

Переваги: зведення до методу багатьох методів побудови моделей.

Недоліки: складність вибору потенційних функцій і необхідність зберігати вихідну вибірку даних.

Метод потенціалів заснований на тих же припущеннях, що і метод потенційних функцій, однак він є не таким, що самонавчається, а заздалегідь обраним, детермінованим: у ньому потенційні функції знаходять на основі нерекурентної процедури за наявною попередньою інформацією.

Застосовність: для складно структурованих і символічних даних.

Незастосовність: якщо потрібно пояснення прийнятого рішення й узагальнення.

Переваги: простота і наявність фізичної аналогії.

Недоліки: складність вибору потенційних функцій і необхідність зберігати вихідну вибірку даних.

Метричні методи засновані на кількісній оцінці близькості між точками (екземплярами) у просторі ознак за відстанню між точками.

Застосовність: там, де потрібно узагальнення набору прецедентів великого розміру.

Незастосовність: складні символічні дані, коли потрібно пояснення прийнятих рішень.

Переваги: добре розроблений математичний апарат і геометрична природа методів.

Недоліки: необхідність вибору виду метрики, пригнічення інформативних ознак малоінформативними, вимога компактності класів.

До метричних методів відносять метод найближчих сусідів, метод АВО (алгоритм вирахування оцінок), метод CBR (Case Based Reasoning – міркування за прецедентами) і метод порівняння з еталоном.

Метод найближчих сусідів при класифікації невідомого об'єкта знаходить задане число k геометрично найближчих до нього у просторі ознак інших об'єктів (найближчих сусідів) із уже відомою належністю до розпізнаваних класів. Рішення про віднесення невідомого об'єкта до того чи іншого класу приймається шляхом аналізу інформації про цю відому належність його найближчих сусідів, наприклад, за допомогою простого підрахунку голосів.

Застосовність: задачі невеликої розмірності за числом класів і ознак.

Незастосовність: задачі великої розмірності.

Переваги: зрозумілість процесу прийняття рішень.

Недоліки: відсутність узагальнення, висока залежність результатів класифікації від міри відстані (метрики), необхідність повного перебору навчальної вибірки при розпізнаванні, велика обчислювальна трудомісткість, необхідність аналізу багатовимірної структури експериментальних даних для мінімізації числа об'єктів, які представляють класи, що може зменшити репрезентативність навчальної вибірки.

Методи АВО є розширенням методу k -найближчих сусідів і засновані на обчисленні пріоритетів (оценок подібності), що характеризують близькість розпізнаваного й еталонного об'єктів за системою ансамблів ознак, що являє собою систему підмножин (опорних множин) заданої множини ознак. Об'єкти існують одночасно у різних підпросторах простору ознак, а ступінь близькості об'єктів обчислюється як комбінація близькостей розпізнаваного об'єкта з еталонними об'єктами, обчислених на множинах часткових описів, отриманих при зіставленні всіх можливих або визначених сполучень ознак, що входять в описи об'єктів. Задача визначення подібності і розходження об'єктів формулюється як

параметрична і виділяється етап настроювання за навчальною вибіркою, на якому підбираються оптимальні значення введених параметрів. Критерієм якості слугує помилка розпізнавання, а параметризуються правила обчислення близькості об'єктів.

Застосовність: задачі невеликої розмірності за числом класів і ознак.

Незастосовність: задачі великої розмірності.

Переваги: теоретично за можливостями може бути не гірше інших методів, оскільки за його допомогою можуть бути реалізовані всі уявні операції з досліджуваними об'єктами.

Недоліки: відсутність узагальнення, залежність результатів класифікації від міри відстані (метрики), необхідність повного перебору навчальної вибірки при розпізнаванні, комбінаторна складність настроювання моделей, що обумовлює низьку швидкість роботи для задач великої розмірності, а також необхідність введення евристичних обмежень і допущень для застосування на практиці при вирішенні задач великої розмірності.

Методи CBR використовують явні функції подібності, визначені на основі відстані, а також стратегію методу *k*-найближчих сусідів. Характеристики розв'язуваної задачі зіставляються з прецедентами, раніше занесеними у базу прецедентів, звідкіля витягаються один або декілька прецедентів, подібних до розв'язуваної задачі, що перевіряються на успіх. Якщо не знайдений схожий прецедент або схожий прецедент не забезпечив успіх рішення, то на основі опису задачі створюється новий прецедент, що зберігається в базі прецедентів. Індукція як логічний процес прийняття рішень формує основу для побудови історії прецедентів, виявлення характерних шаблонів правил серед прецедентів, упорядкування критеріїв прийняття рішень за їхньою поділяючою здатністю і розбиття прецедентів на групи.

Застосовність: проблемні області, що розуміються погано, зі складно структурованими даними, які повільно змінюються в часі, коли потрібно пояснення прийнятих рішень, задачі з неповною інформацією, з інтерпретацією інформації, що варієється у залежності від контексту, а також без чітко визначених цілей і переваг користувача.

Незастосовність: коли немає прецедентів або якщо потрібно складна адаптація або точна й оптимальна відповідь, якщо відсутня інформація про діагнози в минулому.

Переваги: дозволяють користувачу ефективно здійснювати пошук у бібліотеці прецедентів і витягати знання з неї.

Недоліки: відсутність узагальнення, їхня застосовність без утручання людини в основному тільки для пошуку (керування базою прецедентів, наприклад, адаптація, здійснюється, як правило, людьми-менеджерами бази).

Метод порівняння з еталоном застосовується, коли розпізнавані класи відображаються в просторі ознак компактними геометричними угрупованнями. У такому випадку звичайно як крапку – еталона вибирається центр геометричного угруповання класу (чи найближчий до центра об'єкта). Для класифікації невідомого об'єкта знаходиться найближчий до нього еталон, і об'єкт відноситься до того ж класу, що і цей еталон. Як міра близькості можуть застосовуватися різні типи відстаней. При цьому вирішальне правило класифікації об'єктів еквівалентно лінійній вирішальній функції.

Застосовність: задачі невеликої розмірності простору ознак із компактно розташованими класами у просторі ознак.

Незастосовність: задачі зі складним поділом класів і ознаками, поданими у шкалах різних масштабів.

Переваги: простота і компактність одержуваної моделі.

Недоліки: низьке узагальнення, висока залежність результатів класифікації від міри відстані (метрики), необхідність і невизначеність її вибору, необхідність аналізу багатовимірної структури експериментальних даних для мінімізації кількості об'єктів, які описують класи, що може зменшити репрезентативність навчальної вибірки.

Методи на основі м'яких обчислень – неточні, наближені методи вирішення задач, що найчастіше не мають рішення за поліноміальний час.

Застосовність: там, де потрібно узагальнення набору прецедентів великого розміру.

Незастосовність: складні символільні дані.

Переваги: можливість побудови моделі за зашумленими даними, що містять погрішності, масований паралелізм обчислень мережних моделей.

Недоліки: висока ітеративність і тривалість побудови моделі, невизначеність у початковій структурі і значеннях параметрів моделі.

До методів на основі м'яких обчислень відносяться нейронні мережі (НМ), системи на основі нечіткої логіки (нечіткі системи) і нейро-нечіткі мережі (ННМ).

НМ – група (як правило, нелінійних) моделей, що встановлюють множину взаємозалежних функціональних зв'язків між входними стимулами і бажаними результатами, де параметри функціонального зв'язку необхідно налагодити для досягнення оптимальної продуктивності. Структура НМ задається множиною функціональних вузлів – нейронів і множиною зв'язків між нейронами, кожний з яких має напрям і вагу (число, що масштабує сигнал). Ваги зв'язків є налагоджуваними параметрами мережі. Кожен вузол мережі виконує нелінійне перетворення множини входних сигналів у вихідні за допомогою обчислення значення дискримінантної (вагової, постсинаптичної) функції, на основі якої розраховується значення активаційної (передатної) функції, видаване вузлом на вихід.

Вид апроксиматора: числовий.

Здатність навчатися за прикладами: є.

Принцип навчання моделі: оптимізація. Настроювання НМ звичайно здійснюється шляхом впливу на мережу множиною прикладів, спостереженням реакції мережі і перенастроюванням параметрів так, щоб звести до мінімуму помилку. Для настроювання (навчання, тренування) цих параметрів може бути використаний ряд методів оптимізації, зокрема, градієнтних.

Рівень автоматизації побудови моделі: високий.

Залучення людини у процес побудови моделі: завдання вибірки даних, вибір типу моделі, завдання числа шарів, числа вузлів, визначення виду функцій активації і критерію якості навчання.

Рівень узагальнення моделі: високий.

Інтерпретабельність моделі: низька.

Можливість використання експертних знань для побудови моделі: відсутня.

Проблеми при побудові моделі: невизначеність числа шарів і вузлів у мережі, невизначеність у встановленні початкових значень ваг зв'язків.

Розмірність моделі: мала при високому рівні узагальнення (визначається кількістю зв'язків і кількістю вузлів).

Тип моделі: «чорна скриня».

Застосовність: якщо задача забезпечена достатнім числом спостережень і немає експертних знань, дані не можуть бути подані у символльній формі, а також відсутні експертні знання.

Незастосовність: складні символальні дані, коли потрібно пояснення прийнятих рішень.

Переваги: масований паралелізм обчислень і розподілене подання даних, висока швидкість роботи при паралельній реалізації, адаптивність, надійність і стійкість до відмов окремих елементів при апаратній реалізації, стійкість до шумів у вхідних даних.

Недоліки: висока ітеративність і низька швидкість процесу навчання, локальний характер пошуку для градієнтних методів навчання і можливість їхнього зациклення у локальних оптимумах цільової функції.

Нечіткі моделі (*системи нечіткого виведення*) ґрунтуються на нечіткій логіці, що є однією з форм багатозначної логіки, отриманих з теорії нечітких множин для виведення наближених суджень.

Вид апроксиматора: лінгвістичний.

Здатність навчатися за прикладами: немає.

Принцип навчання моделі: на основі правил.

Рівень автоматизації побудови моделі: низький.

Залучення людини у процес побудови моделі: виділення нечітких термів, задавання виду і значень параметрів функцій належності до нечітких термів, задавання правил прийняття рішень, визначення методу дефазифікації.

Рівень узагальнення моделі: низький.

Інтерпретабельність моделі: висока.

Можливість використання експертних знань для побудови моделі: є.

Проблеми при побудові моделі: невизначеність кількості і значень параметрів функцій належності до нечітких термів, необхідність наявності експерта.

Розмірність моделі: велика (визначається числом нечітких термів і кількістю правил).

Тип моделі: «біла скриня»

Застосовність: якщо є експертні знання, які можна сформулювати тільки в лінгвістичній формі.

Незастосовність: якщо необхідно об'єднати експертні знання й експериментальні спостереження, для витягу знань з даних.

Переваги: дозволяють поліпшити прийняття рішень в умовах невизначеності (роздільності ознаки), мають стійкість до змін значень ознак за рахунок могутніх властивостей механізму нечіткого виведення, що інтерполює.

Недоліки: є істотні обмеження на кількість входних змінних унаслідок необхідності розбиття універсальних множин на окремі області, входний набір нечітких правил формулюється експертом-людиною і може виявиться неповним чи суперечливим, входні і вихідні змінні повинні бути описані лінгвістично.

НМ – це подання системи нечіткого виведення у вигляді НМ, зручної для навчання, поповнення, аналізу і використання, структура якої відповідає основним блокам системи нечіткого виведення.

Вид априксиматора: лінгвістичний.

Здатність навчатися за прикладами: є.

Принцип навчання моделі: на основі правил і оптимізації.

Рівень автоматизації побудови моделі: середній.

Залучення людини у процес побудови моделі: задавання вибірки даних, вибір типу моделі, задавання критерію якості навчання, виділення нечітких термів, задавання виду і значень параметрів функцій належності до нечітких термів.

Рівень узагальнення моделі: середній.

Інтерпретабельність моделі: висока.

Можливість використання експертних знань для побудови моделі: є.

Проблеми при побудові моделі: невизначеність кількості і значень параметрів функцій належності до нечітких термів.

Розмірність моделі: велика (визначається числом нечітких термів і кількістю правил).

Тип моделі: «сіра скриня».

Застосовність: якщо потрібно пояснення прийнятих рішень, якщо необхідно об'єднати експертні знання й експериментальні спостереження, для витягу знань з даних.

Незастосовність: складні символільні дані.

Переваги: поєднують переваги НМ і нечітких систем.

Недоліки: висока ітеративність і низька швидкість процесу навчання, локальний характер пошуку для градієнтних методів

навчання і можливість їхнього зациклення в локальних оптимумах цільової функції, істотні обмеження на число входних змінних унаслідок необхідності розбиття універсальних множин на окремі області.

Методи, засновані на припущеннях про клас вирішальних функцій, припускають, що загальний вид вирішальної функції відомий (найчастіше лінійні й узагальнені нелінійні поліноми) і заданий функціонал її якості (звичайно пов'язують з помилкою класифікації), на основі якого за навчальною послідовністю шукається найкраще наближення вирішальної функції шляхом вирішення оптимізаційної задачі за допомогою градієнтних методів.

Застосовність: класи повинні бути добре роздільними, система ознак – ортонормована.

Незастосовність: там, де потрібно мати високий рівень узагальнення і не відомий вид вирішальної функції.

Переваги: чіткість математичної постановки задачі побудови моделі.

Недоліки: відсутність узагальнення, невизначеність вибору виду вирішальної функції для погано вивчених задач, локальний характер і висока ітеративність використовуваних методів оптимізації, необхідність при вирішенні задач з високою розмірністю простору ознак звертання дослідника до лінійних моделей через те, що при підвищенні ступеня поліноміальної вирішальної функції відбувається величезний ріст кількості її членів при проблематичному супутному підвищенні якості розпізнавання.

До методів, заснованих на припущеннях про клас вирішальних функцій, відносять метод групового урахування аргументів (МГУА) і метод стохастичної апроксимації.

МГУА використовує принцип евристичної самоорганізації (уводяться зовнішні доповнення, обирають евристично), і відтворює схему масової селекції, за допомогою чого з наростиючим ускладненням синтезуються і відбираються члени узагальненого полінома Колмогорова-Габора: спочатку розглядаються прості попарні комбінації вихідних ознак, з яких складаються рівняння вирішальних функцій (як правило, не вище другого порядку), кожне рівняння аналізується як самостійна вирішальна функція, і за навчальною вибіркою визначеним способом знаходяться значення параметрів

складених рівнянь, після чого з отриманого набору вирішальних функцій відбирається частина кращих у певному розумінні і на контрольній вибірці здійснюється перевірка якості окремих вирішальних функцій (принцип зовнішнього доповнення), далі відібрані часткові вирішальні функції розглядаються як проміжні змінні, що слугують вихідними аргументами для аналогічного синтезу нових.

Застосовність: задачі оцінювання.

Незастосовність: якщо потрібно витяг знань з даних (зокрема, знань про механізми взаємодії ознак), що обумовлюється вибором форми вирішальних функцій.

Переваги: можливість будувати на його основі моделі високої складності й одержувати практично прийнятні результати, а також рішення задачі добору інформативних ознак у процесі побудови моделі.

Недоліки: залежність моделі від використовуваних евристик, від поділу об'єктів на навчальну і тестову вибірки, виду критерію якості розпізнавання, заданого числа змінних, що пропускаються у наступний ряд селекції і т. д.

Метод стохастичної апроксимації призначений для вирішення задач статистичного оцінювання. Кожне наступне значення оцінки отримується у вигляді заснованої лише на новому спостереженні виправленні до раніше побудованої оцінки. Даний метод узагальнює градієнтні методи (зокрема, метод Ньютона), методи персепtronного типу та ін., що розрізняються використовуваними функціоналами якості вирішального правила (наприклад, мінімізується математичне сподівання ризику або емпіричний ризик за заданою навчальною послідовністю) і оптимізаційними методами пошуку екстремуму.

Застосовність: задачі оцінювання.

Незастосовність: задачі обробки символічних даних.

Переваги: непараметричність (і істотно менша, ніж у параметричних методів залежність від неузгодженості теоретичних уяв про закони розподілу об'єктів у просторі ознак з емпіричною реальністю), рекурентність (простота перерахування оцінки при надходженні нового результату спостережень).

Недоліки: залежність ефективності роботи від якості навчальної вибірки і початкових значень параметрів, що налагоджуються у процесі оптимізації.

Логічні методи базуються на апараті алгебри логіки і дозволяють оперувати інформацією, укладеною в сполученнях значень ознак для здійснення пошуку за навчальною вибіркою логічних закономірностей і формування деякої системи логічних вирішальних правил (наприклад, у вигляді кон'юнкцій елементарних подій – значень ознак), кожне з яких має власну вагу.

Застосовність: задачі невеликої розмірності простору ознак.

Незастосовність: задачі великої розмірності простору ознак.

Переваги: простота апаратної реалізації синтезованих моделей.

Недоліки: відсутність узагальнення, висока обчислювальна складність унаслідок того, що при відборі логічних вирішальних правил (кон'юнкцій) необхідний повний перебір, що спричиняє високі вимоги до ефективної організації обчислювального процесу, детерміністський опис за допомогою двійкових змінних, характерний для логічних методів, є дуже грубою і наближеною моделлю реальної ситуації.

До логічних методів відносять методи витягу асоціативних правил, методи побудови дерев розв'язків і лінгвістичні (структурні) методи.

Методи витягу асоціативних правил призначенні для виявлення у великих масивах даних схованих причинно-наслідкових зв'язків у вигляді продукційних правил (правил виду «Якщо <умови>, то <дії>><>»), для яких визначаються імовірності або коефіцієнти вірогідності, дозволяючи робити відповідні висновки.

Застосовність: як допоміжний засіб для підготовки навчальних даних для методів, керованих даними.

Незастосовність: якщо потрібні висока точність і надійність моделі.

Переваги: дозволяють в автоматичному режимі аналізувати великі масиви даних і узагальнювати дані.

Недоліки: низька швидкість, обумовлена перебором великого числа сполучень спостережень і можливість витягу великого числа тривіальних правил, не істотних для рішення задачі.

Методи побудови дерев розв'язків дозволяють витягати з вибірки продукційні правила і подавати їх у вигляді дерева розв'язків – ієрархічної, послідовної структури, де кожному об'єкту відповідає єдиний вузол, що дає розв'язок.

Застосовність: для задач класифікації і регресії, де потрібно пояснення процесу прийняття рішення.

Незастосовність: для апроксимації істотно нелінійних залежностей.

Переваги: комплексне узагальнення без загального знання, швидкість процесу навчання, генерація правил в областях, де експерту важко формалізувати свої знання, інтерпретабельність моделі.

Недоліки: проблема перебору великої кількості варіантів за прийнятний час, відсутність гарантії одержання узагальнених закономірностей, можливість одержання надзвичайно гіллястих дерев, складність як для аналізу, так і застосування.

Лінгвістичні (структурні) методи засновані на використанні спеціальних граматик (правил), що породжують мови, за допомогою яких може описуватися сукупність властивостей розпізнаваних об'єктів. Шляхом синтаксичного аналізу визначається, чи може деяка фіксована граматика, що описує клас, породити наявний опис об'єкта.

Застосовність: задачі невеликої розмірності простору ознак.

Незастосовність: задачі з великим числом ознак.

Переваги: дозволяють враховувати структуру множини ознак (наприклад, їхню упорядкованість).

Недоліки: відсутність узагальнення, нерозв'язаність ряду теоретичних проблем, труднощі формалізації задачі відновлення (визначення) граматики по деякій множині висловлень (описів об'єктів).

2.4 Метод метричної класифікації

Метод метричної класифікації (метод еталонів) є одним з найпростіших та найпопулярніших методів розпізнавання образів. Він виходить із гіпотези про компактність класів та містить такі фази.

Фаза навчання. Задати навчальну вибірку $\langle x, y \rangle$, де $y \in \{1, 2, \dots, K\}$, K – кількість класів, $K > 1$. Знайти координати центрів класів (еталонів класів – усереднених екземплярів-представників класів) як центрів мас точок-екземплярів:

$$C_j^q = \frac{1}{S^q} \sum_{s=1}^S \{x_j^s \mid y^s = q\}, j = 1, 2, \dots, N, q = 1, 2, \dots, K.$$

Тут C_j^q – значення j -ї координати (ознаки) центра q -го класу, S^q – кількість екземплярів q -го класу у вибірці.

Повернути як результат множину центрів класів $w = \{C^q\}$.

Фаза розпізнавання. Задати розпізнаваний екземпляр x^s та множину центрів класів $w = \{C^q\}$. Знайти відстані від розпізнаваного екземпляра x^s до центру кожного класу:

$$R(x^s, C^q) = \sqrt{\sum_{j=1}^N (x_j^s - C_j^q)^2}, \quad q = 1, 2, \dots, K.$$

Після чого віднести розпізнаваний екземпляр x^s до того класу, до еталону якого відстань від нього є найменшою:

$$y^s = f(w, x^s) = \arg \min_{q=1,2,\dots,K} \{R(x^s, C^q)\}.$$

У випадку, якщо розпізнаваний екземпляр за відстанню буде однаково близьким до центрів декількох класів, то діють одним зі способів:

- видають відмову від розпізнавання;
- відносять екземпляр до того з найближчих класів, екземпляри якого частіше зустрічаються;
- враховують цінні перейменування класів і відносять екземпляр до того класу, потенційні втрати від віднесення до якого є найменшими.

На рис. 2.2 зображене геометричну інтерпретацію методу еталонів для випадку двовимірного простору ознак. Тут кружками позначені екземпляри навчальної вибірки (чорними та білими – для різних класів), знайдені центри класів (еталони) позначені чотирипроменевими зірками (чорними та білими – для різних класів), ромбом позначені розпізнаваний екземпляр.

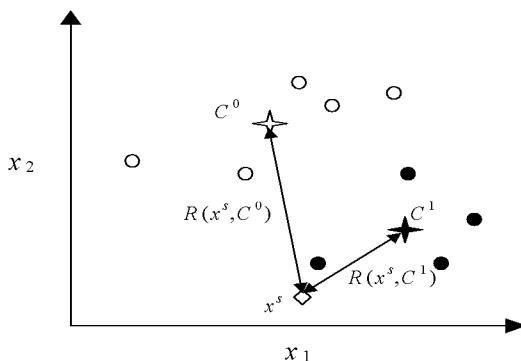


Рисунок 2.2 – Геометрична інтерпретація методу еталонів

Очевидно, що для наведеного прикладу розпізнаваний екземпляр x^s слід віднести до класу чорних точок, оскільки відстань від розпізнаваного екземпляра до еталона класу чорних точок є меншою від відстані від розпізнаваного екземпляра до еталона класу білих точок.

Зauważення. Розглянутий метод може безпомилково вирішувати задачі розпізнавання лише за умови виконання гіпотези про компактність класів та лінійної роздільності класів. У випадку, коли класи не є компактними та лінійно роздільними, метод еталонів можна застосовувати, але він буде працювати з помилками. Для підвищення точності розпізнавання доцільно замість фази навчання методу еталонів використати чіткий кластер-аналіз для виділення еталонів для кожної групи компактно розташованих екземплярів кожного класу та застосувати фазу розпізнавання методу еталонів, виходячи з множини еталонів, сформованої кластер-аналізом.

2.5 Кластерний аналіз. Навчання без учителя

Кластерний аналіз (кластеризація) – це технологія, що дозволяє розподілити вхідні дані на *класи* – групи однотипних екземплярів вибірки, або *кластери* – компактні області групування екземплярів вибірки у просторі ознак.

Вихідною інформацією для кластеризації є вибірка спостережень $x=\{x_j^s\}$, де x_j^s – значення j -ї ознаки s -го екземпляра вибірки, $s = 1, 2, \dots, S$; $j=1, 2, \dots, N$, S – кількість екземплярів вибірки, N – кількість ознак, що характеризують екземпляри вибірки.

Задача кластеризації полягає в розбитті об'єктів з x на декілька кластерів, у яких об'єкти більш схожі між собою, ніж з об'єктами інших кластерів. У метричному просторі «схожість» звичайно визначають через відстань.

Методи кластеризації можна класифікувати на чіткі та нечіткі. Чіткі методи кластеризації розбивають вихідну множину об'єктів x на декілька непересічних підмножин. При цьому будь-який об'єкт із x належить тільки одному кластеру. Нечіткі методи кластерного аналізу дозволяють будь-якому екземпляру одночасно належати до всіх визначених кластерів, але з різним ступенем.

Чіткий кластер-аналіз. Методи даної групи можна умовно розділити на дві категорії: методи кластеризації даних без указівки учителя та методи, що ураховують номер класу при кластеризації.

Якщо здійснюють кластер-аналіз даних за відсутності вказівок учителя (тобто за відсутності виходу y), то говорять про *навчання без учителя*. В іншому випадку (за наявності y) фактично йде мова про *навчання розпізнаванню образів з учителем*.

Серед методів чіткого кластер-аналізу, що оперують тільки вхідними ознаками розглянемо метод пікового групування та метод субтрактивної кластеризації.

Метод пікового групування (гірської кластеризації), запропонований Р. Ягером та Д. Фільовим, не вимагає задавання кількості кластерів. Ідея методу полягає у тому, що спочатку визначають точки, які можуть бути центрами кластерів. Далі для кожної такої точки розраховується значення потенціалу, що показує можливість формування кластера в її околиці. Чим щільніше розташовані об'єкти в околиці потенційного центра кластера, тим вище значення його потенціалу. Після цього ітераційно вибираються центри кластерів серед точок з максимальними потенціалами. Метод гірської кластеризації можна записати як послідовність таких кроків.

Крок 1. Сформувати потенційні центри кластерів, число яких Q повинно бути кінцевим. Центрами кластерів можуть бути об'єкти кластеризації – екземпляри вибірки x , тоді $Q = S$, де S – кількість екземплярів у вибірці x . Другий спосіб вибору потенційних центрів кластерів полягає в дискретизації простору вхідних ознак. Для цього діапазони зміни вхідних ознак розбивають на кілька інтервалів. Проводячи через точки розбиття прямі, паралельні координатним осям, одержуємо «гратовий» гіперкуб. Вузли цих граторів будуть відповідати центрим потенційних кластерів. Позначимо через q_r – кількість значень, що можуть приймати центри кластерів за r -ю координатою, $r = 1, 2, \dots, N$. Тоді кількість можливих кластерів буде дорівнювати: $Q = \prod_{r=1}^N q_r$.

Крок 2. Розрахувати потенціал центрів кластерів за формулою:

$$P(C^q) = \sum_{s=1}^S e^{-\alpha d(C^q, x^s)}, \quad q = 1, 2, \dots, Q,$$

де $C^q = \{C_j^q\}$ – потенційний центр q -го кластера, C_j^q – значення j -ї ознаки для центра q -го кластера; α – позитивна константа, $d(C^q, x^s)$ – відстань (наприклад, евклідова) між потенційним центром кластера C^q та об'єктом кластеризації x^s .

У випадку, коли об'єкти кластеризації задані двома ознаками ($N = 2$), графічне зображення розподілу потенціалу буде являти

собою поверхню, що нагадує гірський рельєф. Звідси і назва – гірський метод кластеризації.

Крок 3. Вибрати як центри кластерів координати «гірських» вершин. Для цього, центром першого кластера призначають точку з найбільшим потенціалом. Звичайно, найвища вершина оточена декількома досить високими піками. Тому призначення центром наступного кластера точки з максимальним потенціалом серед вершин, що залишилися, привело б до виділення великого числа близько розташованих центрів кластерів.

Щоб вибрати наступний центр кластера необхідно спочатку виключити вплив тільки що знайденої кластера. Для цього значення потенціалу для можливих центрів кластерів, що залишилися, перераховується в такий спосіб: від поточних значень потенціалу віднімають внесок центра тільки що знайденої кластера (тому кластеризацію за цим методом іноді називають субтрактивною). Перерахунок потенціалу відбувається за формулою:

$$P_2(C^q) = P_1(C^q) - P_1(C^{v_1})e^{-\beta d(C^q, C^{v_1})},$$

де P_1 – потенціал на 1-й ітерації; P_2 – потенціал на 2-й ітерації; v_1 – номер першого знайденої центра кластера:

$$v_1 = \arg \max_{q=1,2,\dots,Q} P_1(C^q),$$

де β – позитивна константа.

Номер центра другого кластера визначається за максимальним значенням оновленого потенціалу:

$$v_2 = \arg \max_{q=1,2,\dots,Q} P_2(C^q)$$

Потім знову перераховується значення потенціалів:

$$P_3(C^q) = P_2(C^q) - P_2(C^{v_2})e^{-\beta d(C^q, C^{v_2})}.$$

Крок 4. Якщо максимальне значення потенціалу перевищує деякий поріг, перейти до кроку 2, у протилежному випадку – зупинення.

Метод гірської кластеризації є ефективним, якщо розмірність вхідного вектора не є занадто великою. У протилежному випадку (при

великій кількості ознак) число потенційних центрів наростиє лавиноподібно, і процес розрахунку чергових пікових функцій стає за- надто тривалим, а процедура кластеризації – малоефективною.

Метод різницевого групування (субтрактивної кластеризації, subtractive clustering) на відміну від попереднього методу, як потенційні центри кластерів розглядає екземпляри навчальної вибірки.

Пікова функція $P(x^s)$ задається у вигляді:

$$P(x^s) = \sum_{\substack{g=1 \\ g \neq s}}^S \exp \left(-\frac{d(x^s, x^g)^{2b}}{0,5r_a^2} \right), \quad s = 1, 2, \dots, S.$$

Значення коефіцієнта r_a визначає сферу сусідства. На значення $P(x^s)$ істотно впливають тільки ті вектори x^g , що розташовані в межах цієї сфери. При великій щільноті точок навколо x^s (потенційного центра) значення функції $P(x^s)$ буде великим. Навпаки, мале її значення свідчить про те, що в околиці x^s знаходиться незначна кількість даних. Така точка вважається «невдалим» кандидатом у центри.

Після розрахунку значень пікової функції серед усіх точок відбирається вектор x^s , для якого міра щільноті $P(x^s)$ виявилася найбільшою. Саме ця точка стає першим відібраним центром C^1 .

Вибір наступного центра можливий після виключення попереднього центра і всіх точок, що лежать у його околиці. Подібно до методу пікового групування, перевизначається пікова функція:

$$P_2(x^s) = P_1(x^s) - P_1(C^1) \exp \left(-\frac{d(x^s, C^1)^{2b}}{0,5r_b^2} \right).$$

При визначенні $P_2(x^s)$ коефіцієнт r_b позначає нове значення константи, що задає сферу сусідства чергового центра. Звичайно дотримуються умови $r_b \geq r_a$. Пікова функція $P_2(x^s)$ приймає нульове значення при $x^s = C^1$ і близька до нуля в найближчій околиці цієї точки.

Після модифікації значень пікової функції шукається наступна точка x^s , для якої величина $P_2(x^s)$ виявляється максимальною. Ця точка стає наступним центром кластера C^2 .

Процес пошуку чергового центра відновляється після виключення компонентів, що відповідають уже відібраним точкам, і

завершується в момент фіксації всіх центрів, передбачених початковими умовами.

Відповідно до описаного методу відбувається самоорганізація множини векторів x , що полягає у знаходженні оптимальних значень центрів, які відповідають множині даних з мінімальною погрішністю.

Якщо ми маємо справу з множиною навчальних даних у вигляді пар $\langle x^s, y^s \rangle$, то для знаходження центрів, що відповідають множині векторів y^s , достатньо сформувати розширену версію векторів x : $x_{N+1}^s = y^s$. Процес групування, проведений із пред'явленням розширених векторів x^s , дозволяє визначити також розширені версії центрів C^q , в описі яких легко виділити частину, що відповідає векторам x (перші N компонентів), і залишок, що відповідає вектору y . У такий спосіб можна одержати центри як вхідних змінних, так очікуваних вихідних значень.

Розглянемо окремо прості методи, що враховують номер класу при кластеризації.

Метод чіткого кластер-аналізу з додаванням кластерів.

Крок 1. Задати навчальну вибірку $\langle x, y \rangle$ та невелике ціле позитивне число ε – припустиму помилку розпізнавання.

Крок 2. Знайти центри кластерів $\{C^q\}$ та співставлені ним номери класів $\{Y^q\}$ на основі методу еталонів (вважаємо, що кожен клас поданий усього одним кластером). Покласти: $Q=K$.

Крок 3. Розпізнати екземпляри вибірки $\langle x, y \rangle$ відносно центрів кластерів $\{C^q\}$ – для кожного екземпляра отримати $y_{\text{розр.}}^s$.

Крок 4. Оцінити критерій якості розпізнавання:

$$E = \sum_{s=1}^S \{1 | y^s \neq y_{\text{розр.}}^s\}.$$

Крок 5. Якщо $E > \varepsilon$ та $Q < S$, тоді виділити усі екземпляри, що помилково розпізнані та занести кожний такий екземпляр x^s як центр нового кластеру: $C^{Q+1} = x^s$, $Y^{Q+1} = y^s$, $Q = Q + 1$ та перейти до кроku 3, у протилежному випадку – закінчити пошук і повернути отримані центри кластерів.

Перевагою даного методу є те, що він намагається отримати максимальне узагальнення даних з самого початку пошуку, а недоліком – те, що він робить це за рахунок компромісу за точністю (помилкою).

Метод чіткого кластер-аналізу з видаленням кластерів.

Крок 1. Задати навчальну вибірку $\langle x, y \rangle$.

Крок 2. Занести усі екземпляри вихідної навчальної вибірки у центри кластерів $\{C^q\}$: $C^q = x^q, Y^q = y^q, q = 1, 2, \dots, S$. Покласті: $Q = S$.

Крок 3. Розпізнати екземпляри вибірки $\langle x, y \rangle$ відносно центрів кластерів $\{C^q\}$ – для кожного екземпляра отримати $y^s_{\text{розв.}}$.

Крок 4. Оцінити критерій якості розпізнавання:

$$E = \sum_{s=1}^S \{1 | y^s \neq y^s_{\text{розв.}}\}.$$

Крок 5. Якщо $E < \varepsilon$ та $Q > K$, тоді знайти відстані (квадрати відстаней) між усіма центрами кластерів:

$$R(C^q, C^p) = \sum_{j=1}^N (C_j^q - C_j^p)^2,$$

після чого знайти два найближчих кластера q та p .

Якщо $Y^q = Y^p$, тоді поєднати кластери:

$$C_j^q = \frac{C_j^q + C_j^p}{2},$$

після чого видалити кластер C^p , встановити: $Q = Q - 1$ та перейти до кроку 3, у протилежному випадку – закінчити пошук і повернути отримані центри кластерів.

Перевагою даного методу є те, що він з самого початку намагається отримати максимальну точність розпізнавання, а недоліком – те, що це досягається за рахунок зменшення узагальнення даних.

Гібридний (комбінований) метод чіткого кластер-аналізу поєднує запропоновані вище методи: спочатку виконується кластер-аналіз з додаванням кластерів, після чого – кластер-аналіз із видаленням кластерів, або навпаки.

Нечіткий кластер-аналіз використовується при побудові нейро-нечітких систем для визначення нечітких множин, якщо вони невідомі апріорі. Нечіткі множини знаходяться як проекції кластерів на кожну розмірність. Можливо поєднувати апріорні знання з кластерним аналізом, використовуючи його для уточ-

нення параметрів функції належності. Недоліком такого методу визначення нечітких множин є складність їхньої інтерпретації.

Більшість методів нечіткої кластеризації спрямовані на мінімізацію суми:

$$J(x, u, C) = \sum_{s=1}^S \sum_{v=1}^V \left(\left(u_v^s \right)^m d^2(x^s, C^v) \right)$$

при виконанні умов:

$$V > 1, \quad \sum_{s=1}^S u_v^s > 0, \quad \sum_{v=1}^V u_v^s = 1,$$

де S – кількість екземплярів, N – кількість параметрів, що описують один екземпляр (або кластер), V – кількість кластерів; $x = (x^1, x^2, \dots, x^S)^T$ – це матриця входів для екземплярів навчальної вибірки, $x^s = (x_1^s, x_2^s, \dots, x_N^s)$ – входи s -го екземпляра, $s = 1, 2, \dots, S$, $u = (u^1, u^2, \dots, u^S)^T$ – матриця належностей екземплярів до кожного з кластерів, $u^s = (u_1^s, u_2^s, \dots, u_V^s)$ – вектор належностей s -го екземпляра до кожного з кластерів, $u_v^s \in [0, 1]$, $C = (C^1, C^2, \dots, C^V)^T$ – матриця центрів кластерів, $C^v = (C_1^v, C_2^v, \dots, C_N^v)$ – центр v -го кластера, $v = 1, 2, \dots, V$, $m > 1$ – ступінь нечіткості отриманого розподілу (зазвичай обирається рівним 2), $d(x^s, C^v)$ – відстань між s -м екземпляром та центром v -го кластера.

Координати центрів кластерів визначають за формулою:

$$C_j^v = \frac{\sum_{s=1}^S (u_v^s)^m x_j^s}{\sum_{s=1}^S (u_v^s)^m}.$$

Найбільш простим є метод, в якому відстань між екземпляром та кластером знаходиться як евклідова відстань:

$$d(x^s, C^v) = \sqrt{\sum_{j=1}^N (x_j^s - C_j^v)^2}.$$

Такий метод шукає кластери як сфери однакового розміру.

Більш складні методи кластеризації шукають кластери як гіпереліпсоїди різного розміру. Такі методи називають *частковими*, вони не можуть вірно опрацьовувати шуми та викиди і віднаходити кластери з неопуклими поверхнями. Для проведення кластерного аналізу за допомогою часткового методу необхідно задати його параметри: діапазон значень змінних, кількість кластерів для кожної із змінних (або їх ширину), функцію належності, що описує кластери та інші параметри в залежності від обраного методу кластеризації.

За допомогою *ієрархічних методів* можна віднайти кластери, об'єднуючи менші кластери та розподіляючи більші. Таким чином знаходить дерево кластерів, на різних рівнях якого можна отримати різне розподілення на кластери.

Цільносні методи та сіткові методи дозволяють розподіляти на кластери різного розміру довільно розподілені екземпляри. Вони також добре впізнають шуми та викиди, але потребують ретельного вибору параметрів, необхідних для реалізації методу.

Метод FCM (Fuzzy c-means – нечітких c-середніх), заснований на ідеях Дж. Дана (J. Dunn) та Дж. Беждека (J. Bezdek), для вирішення задачі нечіткої кластеризації має ітеративний характер послідовного поліпшення певного вихідного нечіткого розбиття $R(A) = \{A_v | A_v \subseteq A\}$, що задається користувачем або формується автоматично за певним евристичним правилом. На кожній з ітерацій рекурентно перераховуються значення функцій належності нечітких кластерів та їхніх типових представників.

Метод FCM закінчить роботу у випадку, коли відбудеться виконання заданого априорі деякого кінцевого числа ітерацій, або коли мінімальна абсолютна різниця між значеннями функцій належності на двох послідовних ітераціях не стане менше деякого априорі заданого значення.

Формально метод FCM визначається у формі ітеративного виконання такої послідовності кроків.

Крок 1. Попередньо необхідно задати такі значення: кількість шуканих нечітких кластерів $V (V > 1)$, максимальну кількість ітерацій методу *Epochs*, параметр збіжності методу ϵ , а також експонентну вагу розрахунку цільової функції і центрів кластерів m (як правило, $m = 2$). Як поточне нечітке розбиття на першій ітерації методу для вибірки даних x задати деяке вихідне нечітке розбиття $R(A) = \{A_v | A_v \subseteq A\}$ на

V непорожніх нечітких кластерів, що описуються сукупністю функцій належності u^s_v , $s = 1, 2, \dots, S; v = 1, 2, \dots, V$. Нечітке розбиття отримують шляхом генерації випадковим чином елементів u^s_v , що задовольняють умовам цільової функції.

Крок 2. Для вихідного поточного нечіткого розбиття $R(A) = \{A_v | A_v \subseteq A\}$, розрахувати центри нечітких кластерів C_j^v , $v = 1, 2, \dots, V; j = 1, 2, \dots, N$, та значення цільової функції $J(x, u, C)$. Кількість виконаних ітерацій покласти рівною 1.

Крок 3. Сформувати нове нечітке розбиття $R'(A) = \{A_v | A_v \subseteq A\}$ вихідної множини об'єктів кластеризації A на V непорожніх нечітких кластери, що характеризуються сукупністю функцій належності $u^{s'}_v$, $v = 1, 2, \dots, V; x^s \in A$, які визначаються за формулою:

$$u^{s'}_v = \left(\sum_{g=1}^V \left(\frac{d(x^s, C^g)}{d(x^s, C^v)} \right)^{\frac{2}{m-1}} \right)^{-1}, \quad v = 1, 2, \dots, V.$$

Крок 4. При цьому якщо для деякого v та деякого x^s значення $d(x^s, C^v) = 0$, то для відповідного нечіткого кластера встановимо $u^{s'}_v = 1$, а для інших кластерів $u^{h'}_g = 0$, $g = 1, 2, \dots, V; g \neq v; h = 1, 2, \dots, S$. Якщо ж таких v для деякого x^s виявиться декілька, тобто для них значення $d(x^s, C^v) = 0$, то евристично для меншого з v встановимо $u^{s'}_v = 1$, а для інших встановимо $u^{h'}_g = 0$, $g = 1, 2, \dots, V; g \neq v; h = 1, 2, \dots, S$.

Крок 5. Для нового нечіткого розбиття $R'(A) = \{A_v | A_v \subseteq A\}$ розрахувати центри нечітких кластерів C_j^v та значення цільової функції $J'(x, u, C)$.

Крок 6. Якщо кількість виконаних ітерацій перевищує задане число $Epochs$ або ж модуль різниці $|J(x, u, C) - J'(x, u, C)| \leq \epsilon$, то за шуканий результат нечіткої кластеризації прийняти нечітке розбиття $R'(A)$ і закінчити виконання методу. У протилежному випадку вважати поточним нечітким розбиттям $R(A) = R'(A)$ і перейти до кроку 3 методу, збільшивши на одиницю кількість виконаних ітерацій.

Вибір кількості кластерів V є однією з найважливіших проблем у розглянутому методі. Правильно вибрати кількість кластерів для реальних задач без будь-якої апріорної інформації про структуру даних досить складно. Існує два формальних підходи до вибору кількості кластерів.

Перший підхід заснований на *критерії компактності та роздільності* отриманих кластерів. Логічно припустити, що при правильному виборі кількості кластерів дані будуть розбиті на компактні і добре віддільні одна від іншої групи. У іншому випадку, кластери, імовірно, не будуть компактними і добре віддільними.

Існує кілька критеріїв оцінки компактності кластерів, однак питання про те, як формально і вірогідно визначити правильність вибору кількості кластерів для довільного набору даних залишається відкритим. Для методу FCM рекомендується використовувати *індекс Хіє-Бені* (Xie-Beni index):

$$\chi = \frac{\sum_{v=1}^V \sum_{s=1}^S (u_v^s)^m d^2(x^s, C^v)}{S \min d^2(x^s, C^v)}.$$

Другий підхід заснований на *редукції кількості кластерів* і пропонує починати кластеризацію при досить великій кількості кластерів, а потім послідовно поєднувати схожі суміжні кластери. При цьому використовуються різні формальні критерії схожості кластерів.

Експонентна вага m є також важливим параметром методу FCM. Чим більше m , тим кінцева матриця нечіткого розбиття $\{u_v^s\}$ стає більш «розмазаною», та при $m \rightarrow \infty$ її елементи: $u_v^s = V^{-1}$, що є дуже поганим рішенням, тому що всі об'єкти належать до всіх кластерів з одним і тим же ступенем. Крім того, експонентна вага дозволяє при формуванні координат центрів кластерів підсилити вплив об'єктів з великими значеннями ступенів належності й зменшити вплив об'єктів з малими значеннями ступенів належності. На сьогодні не існує теоретично обґрутованого правила вибору значення експонентної ваги. Звичайно встановлюють: $m = 2$.

Вибір норми для визначення близькості є ще однією проблемою для методів кластер-аналізу. У базовому методі FCM відстань між об'єктом $x^s = \{x_j^s\}$ і центром кластера $C^v = \{C_j^v\}$, $j = 1, 2, \dots, N$, розраховується через стандартну Евклідову норму. У кластерному аналізі застосовуються й інші норми, серед яких часто використовується діагональна норма і норма Махалонобіса.

Норму в загальному виді можна задати через симетричну по-зитивно визначену матрицю $B = \{B_{i,j}\}$, $i, j = 1, 2, \dots, N$, як:

$$d^2(x^s, C^v)_B = (x^s - C^v) \cdot B \cdot (x^s - C^v)^T,$$

де x^s та C^v – вектори з координатами екземпляра та центра клас-тера, T – операція транспонування.

Евклідова норма дозволяє виділяти кластери у вигляді гіперсфер. Для Евклідової норми матриця B являє собою одиничну матрицю:

$$B_{i,j} = \begin{cases} 1, & i = j; \\ 0, & i \neq j. \end{cases}$$

Діагональна норма дозволяє виділяти кластери у вигляді гіпереліпсоїдів, орієнтованих уздовж координатних осей. Для діагональної норми матриця B задається у такий спосіб:

$$B_{i,j} = \begin{cases} w_i, & i = j; \\ 0, & i \neq j, \end{cases}$$

де w_i – елементи головної діагоналі матриці, що інтерпретуються як ваги координат.

Норма Махаланобіса дозволяє виділяти кластери у вигляді гіпереліпсоїдів, вісі яких можуть бути орієнтовані в довільних напрямках. Для норми Махаланобіса матриця B розраховується через коваріаційну матрицю від x :

$$B = G^{-1}, \quad G = \frac{1}{S} \sum_{s=1}^S (x^s - \bar{x})^T (x^s - \bar{x}), \quad \bar{x} = \frac{1}{S} \sum_{s=1}^S x^s,$$

де G – коваріаційна матриця; \bar{x} – вектор середніх значень даних.

У результаті застосування методів кластеризації з фіксованою нормою форма всіх кластерів виходить однаковою. Методи кластеризації ніби нав'язують даним невластиву їм структуру, що приводить не тільки до неоптимальних, але й іноді до принципово неправильних результатів. Для усунення цього недоліку запропоновано декілька методів, серед яких виділимо метод Густавсона-Кесселя.

Метод Густавсона-Кесселя (Gustafson-Kessel method) використовує адаптивну норму для кожного кластера, тобто для кожного v -го кластера існує своя норм-породжуюча матриця B^v . У цьому методі при кластеризації оптимізуються не тільки координати центрів кластерів і матриця нечіткого розбиття, але також і норм-породжуючі матриці для всіх кластерів. Це дозволяє виділяти кластери різної

геометричної форми. Критерій оптимальності $J(x, u, C)$ є лінійним відносно B^v , тому для одержання ненульових рішень уводять певні обмеження на норм-породжуючі матриці. В методі Густавсона-Кесселя це обмеження на значення визначника норм-породжуючих матриць: $\det(B^v) > 0, v = 1, 2, \dots, V$.

Метод Густавсона-Кесселя може бути поданий таким чином.

Крок 1. Зробити початкове розміщення центрів кластерів у просторі ознак. Ця ініціалізація може бути випадковою або заснованою на результатах пікового або різницевого групування даних. Створити елементарну форму масштабувальної матриці B^v .

Крок 2. Сформувати матрицю коефіцієнтів належності усіх векторів $x^s, s = 1, 2, \dots, S$, до центрів $C^v, v = 1, 2, \dots, V$, шляхом розрахунку u_v^s :

$$u_v^s = \left(\sum_{g=1}^V \left(\frac{d(x^s, C^g)_B}{d(x^s, C^g)_B} \right)^{\frac{2}{m-1}} \right)^{-1}, v = 1, 2, \dots, V.$$

Якщо $\exists k : d(x^k, C^v) = 0$, тоді прийняти: $u_v^k = 1, u_v^s = 0, s = 1, 2, \dots, S, s \neq k, v = 1, 2, \dots, V$.

Крок 3. Розрахувати нові координати центрів кластерів:

$$C_j^v = \frac{\sum_{s=1}^S (u_v^s)^m x_j^s}{\sum_{s=1}^S (u_v^s)^m}.$$

Крок 4. Знайти для кожного центра кластера матрицю коваріації Φ^v :

$$\Phi^v = \sum_{s=1}^S (u_v^s)^m (x^s - C^v)(x^s - C^v)^T, v = 1, 2, \dots, V.$$

Крок 5. Для всіх $v = 1, 2, \dots, V$, розрахувати нову масштабувальну матрицю $B^v = \sqrt[N]{\det(\Phi^v)} (\Phi^v)^{-1}$, де N – кількість ознак, що характеризують екземпляри навчальної вибірки.

Крок 6. Якщо останні зміни положень центрів кластерів і матриці коваріації є досить малими відносно попереднього значення і не перевищують попередньо заданої граничної величини ϵ , тоді завершити ітераційний процес; у протилежному випадку – перейти до кроку 2.

Метод Густавсона-Кесселя має значно більшу обчислювальну трудомісткість у порівнянні з методом FCM.

Адаптивний метод нечіткої самоорганізації сформульований для гаусівської функції і дозволяє визначати кількість центрів кластерів і їхнє розташування в частині, що відповідає умовам (множина векторів x^s) і висновкам (множина скалярних очікуваних значень y^s). Цей метод можна описати у такий спосіб.

Крок 1. При старті з першої пари даних $\langle x^1, y^1 \rangle$ створюється перший кластер з центром $C^1 = x^1$. Приймається, що $w_1 = y^1$ і що потужність множини $L_1 = l$. Нехай r позначає граничну евклідову відстань між вектором x та центром, при якому дані будуть трактуватися як належні до створеного кластера. Для збереження загальності рішення приймається, що в момент початку навчання існують V кластерів з центрами $C^v, v=1, 2, \dots, V$, та відповідні ним значення w_v та $L_v, v=1, 2, \dots, V$.

Крок 2. Після зчитування s -ї навчальної пари $\langle x^s, y^s \rangle$ розраховуються відстані між вектором x^s та всіма існуючими центрами $d(x^s, C^v), v = 1, 2, \dots, V$. Допустимо, що найближчий центр – це C^q . У такому випадку в залежності від значення $d(x^s, C^q)$ може виникнути одна з двох ситуацій:

- якщо $d(x^s, C^q) > r$, тоді створюється новий кластер $C^{q+1} = x^s$, причому $w_{q+1}(s) = y^s, L_{q+1}(s) = 1$. Параметри створених до цього кластерів не змінюються, тобто: $w_v(s) = w_v(s-1), L_v(s) = L_v(s-1), v = 1, 2, \dots, V$. Збільшується кількість кластерів: $V = V + 1$;
- якщо $d(x^s, C^q) \leq r$, тоді дані включаються в кластер C^q , параметри якого слід уточнити відповідно до формул:

$$w_q(s) = w_q(s-1) + y^s, L_q(s) = L_q(s-1) + 1, C^q(s) = (C^q(s-1) L_q(s-1) + x^s) / L_q(s).$$

В іншій версії методу фіксується положення центрів C^q після ініціалізації, і їхні координати вже не змінюються. У багатьох випадках такий прийом поліпшує результати адаптації.

Крок 3. Після уточнення параметрів нечіткої системи функція, що апроксимує вхідні дані системи, визначається як:

$$y^{s*} = \frac{\sum_{v=1}^V w_v(s) e^{-\sigma^{-2} d^2(x^s, C^v)}}{\sum_{v=1}^V L_v(s) e^{-\sigma^{-2} d^2(x^s, C^v)}},$$

де σ – певна константа, тоді як інші кластери не змінюються.

При повторенні перерахованих кроків методу до $s = S$ з уточненням щоразу значення V простір даних розподіляється на V кластерів, при цьому потужність кожного з них визначається як $L_v = L_v(s)$, центр – як $C^v = C^v(s)$, а значення приписаної йому накопиченої функції y^s – як $w_v = w_v(s)$.

Цей метод є таким, що самоорганізується, оскільки поділ простору даних на кластери відбувається самостійно і без участі людини, відповідно до заданого значення порога r . При малому значенні r кількість кластерів зростає, у результаті чого апроксимація даних стає більш точною, однак це досягається за рахунок більш складної функції і збільшення обсягу необхідних обчислень при одночасному погрішенні узагальнюючих властивостей моделі. Якщо значення r є занадто великим, то обчислювальна складність зменшується, однак зростає погрішність апроксимації. При підборі оптимальної величини порога r повинний дотримуватися компроміс між точністю відображення й обчислювальною складністю. Як правило, оптимальне значення r підбирається методом проб і помилок з використанням обчислювальних експериментів.

Результати нечіткого кластер-аналізу можна використовувати для синтезу нечітких правил. Кожен кластер буде являти собою деяке нечітке правило, що узагальнює підмножину екземплярів навчальної вибірки, найбільш тісно розташованих у просторі ознак.

Функції належності термів у посилках правила отримують проектуванням ступенів належності відповідного кластера на вхідні змінні. Потім отримані множини ступенів належностей апроксимують придатними параметричними функціями належності.

Як висновок правила сінглтонної бази знань вибирають координату центра кластера. Висновки правил бази знань Мамдані знаходять також як і функції належності термів вхідних змінних. Висновки правил бази знань Сугено знаходять за методом найменших квадратів. При кластеризації з використанням норми Махалонобіса як висновки правил типу Сугено можуть бути обрані рівняння довгих осей гіпереліпсоїдів.

Метод поступово зростаючого розбиття (incremental decomposition algorithm – IDA) полягає у наступному.

На першій ітерації є одне продукційне правило, що має як область свого впливу (область, у якій значення результиуючої функції належності передумови нечіткого правила перевищує задану величину) усю множину припустимих вхідних значень (рис. 3.2).

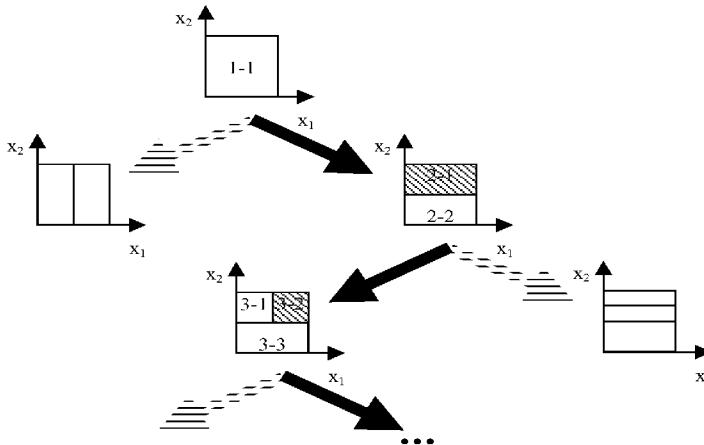


Рисунок 3.2 – Схема роботи методу IDA

На другій ітерації дане правило розбивається на два двома способами (показані стрілками). Проводиться навчання і вибирається, який зі способів розбиття дає найменшу погрішність (даний переход відзначений чорною стрілкою). Серед наявних правил вибирається те, для якого складової погрішності в загальній погрішності є найбільшою (область його впливу заштриховано). Воно і підлягає розбиттю на два правила двома способами (ітерація 3). Описаний процес продовжується до досягнення необхідної точності або поки не буде згенеровано задане число продукційних правил.

2.6. Приклади виконання завдань

Приклад 1. Для навчальної вибірки $x = \{<2\ 4\ 1>, <3\ 3\ 2>, <1\ 0\ 0>, <-2\ 1\ -1>\}$, екземплярам якої співставлено вихідний вектор $y = \{0\ 0\ 1\ 1\}$ визначити координати еталонів класів C^0 та C^1 на основі методу еталонів.

Загуваження. Іноді класи нумерують, починаючи не з одиниці, а з нуля. Тоді у відповідних формулах корегують K відповідним чином.

Підставимо відповідні дані у формули:

$$C_1^0 = \frac{1}{2}(2+3) = 2,5; C_2^0 = \frac{1}{2}(4+3) = 3,5; C_3^0 = \frac{1}{2}(1+2) = 1,5;$$

$$C_1^1 = \frac{1}{2}(1 - 2) = -0,5; C_2^1 = \frac{1}{2}(0 + 1) = 0,5; C_3^1 = \frac{1}{2}(0 - 1) = -0,5.$$

У результаті отримаємо координати еталонів класів: $C^0 = <2,5\ 3,5\ 1,5>$ та $C1 = <-0,5\ 0,5\ -0,5>$.

Приклад 2. За заданими еталонами класів $C^0 = <2,5\ 3,5\ 1,5>$ та $C^1 = <-0,5\ 0,5\ -0,5>$ розпізнати екземпляр $x = <2\ 4\ 1>$ на основі методу еталонів із використанням евклідової відстані.

Спочатку визначимо відстані від кожного екземпляра до еталона кожного класу:

$$R(x, C^0) = \sqrt{(2 - 2,5)^2 + (4 - 3,5)^2 + (1 - 1,5)^2} = \sqrt{0,75} \approx 0,866,$$

$$R(x, C^1) = \sqrt{(2 + 0,5)^2 + (4 - 0,5)^2 + (1 + 0,5)^2} = \sqrt{19,0625} \approx 4,366.$$

Після чого віднесемо розпізнаваний екземпляр до того класу, до еталону якого відстань від нього є найменшою: оскільки $R(x, C^0) < R(x, C^1)$, то встановимо: $y = 0$.

Приклад 3. Нехай побудовано розпізнавальну модель на основі методу еталонів на основі навчальної вибірки для якої вихід $y = \{0\ 0\ 0\ 1\ 1\ 1\}$. При розпізнаванні навчальної вибірки отримано розрахункові значення $f(x) = \{0\ 0\ 1\ 0\ 1\ 1\}$. Оцінити критерій якості моделі.

Оскільки вихідна ознака є бінарною, можна оцінити помилку:

$$E = \sum_{s=1}^S |y^s - f(x^s)| = |0 - 0| + |0 - 0| + |0 - 1| + |1 - 0| + |1 - 1| + |1 - 1| = 2;$$

При цьому також можна оцінити імовірність прийняття помилкових рішень: $P_{\text{пом.}} = E/S = 2/6 \approx 0,33$ і імовірність прийняття правильних рішень $P_{\text{пр.}} = 1 - P_{\text{пом.}} \approx 1 - 0,33 = 0,67$.

Також можна оцінити середньоквадратичну помилку:

$$\begin{aligned} E &= \frac{1}{2} \sum_{s=1}^S (y^s - f(x^s))^2 = \\ &= \frac{1}{2} ((0 - 0)^2 + (0 - 0)^2 + (0 - 1)^2 + (1 - 0)^2 + (1 - 1)^2 + (1 - 1)^2) = \frac{2}{2} = 1. \end{aligned}$$

?

2.7 Контрольні питання

1. Використання кластер-аналізу при розпізнаванні образів.
2. Вимоги до навчальних виброк даних.
3. Задача кластер-аналізу.
4. Задача розпізнавання образів.
5. Кластерний аналіз, задача кластеризації.
6. Лінгвістичні (структурні) методи.
7. Лінійна роздільність і лінійна нерозділеність класів.
8. Логічні методи.
9. Метод CBR.
10. Метод АВО.
11. Метод групового урахування аргументів.
12. Метод дискримінантних функцій.
13. Метод найближчих сусідів.
14. Метод порівняння з еталоном.
15. Метод потенціалів.
16. Метод потенційних функцій.
17. Метод стохастичної апроксимації.
18. Методи витягу асоціативних правил.
19. Методи метричної класифікації.
20. Методи метричної класифікації.
21. Методи на основі м'яких обчислень.
22. Методи побудови дерев розв'язків.
23. Методи поділу у просторі ознак.
24. Методи регресійного аналізу.
25. Методи статистичних рішень.
26. Методи, засновані на припущеннях про клас вирішальних функцій.
27. На які різновиди класифікують методи кластер-аналізу?
28. Навчання без учителя.
29. Навчання з учителем.
30. Нейро-нечіткі мережі.
31. Нейронні мережі.
32. Нечіткий кластер-аналіз.
33. Норма, евклідова норма, діагональна норма, норма Махalanобіса.
34. Основні поняття теорії розпізнавання образів.

35. Подібність кластер-аналізу і метричної класифікації.
36. Статистичні методи.
37. У чому полягають методи нечітких *c*-середніх, Густавсона-Кесселя, гірської та субтрактивної кластеризації?
38. Характеристики методів розпізнавання: помилка навчання / класифікації, час навчання / класифікації, цільова функція навчання.
39. Чи впливає кількість використаних ознак на швидкість кластер-аналізу?
40. Чи впливає кількість використаних ознак на швидкість кластер-аналізу? Відповідь обґрунтуйте.
41. Чи впливає обсяг навчальної вибірки на швидкість кластер-аналізу?
42. Чи впливає обсяг навчальної вибірки на швидкість навчання методу метричної класифікації?
43. Чи впливає обсяг навчальної вибірки на швидкість навчання? Відповідь обґрунтуйте.
44. Чи впливає репрезентативність навчальної вибірки на точність класифікації екземплярів тестової вибірки? Відповідь обґрунтуйте.
45. Чи впливає репрезентативність тестової вибірки на точність класифікації екземплярів тестової вибірки? Відповідь обґрунтуйте.
46. Чи залежить якість навчання від якості та обсягу навчальної вибірки? Відповідь обґрунтуйте.
47. Чи повинна навчальна вибірка бути репрезентативною?
48. Чи повинна тестова вибірка бути репрезентативною?
49. Чіткий кластер-аналіз.
50. Що таке генеральна сукупність, вибірка, екземпляр, ознака?
51. Що таке репрезентативна вибірка даних?
52. Що таке: клас, кластер,

2.8 Практичні завдання

 Завдання 1. Написати реферат на одну з таких тем.

1. Статистичні методи.
2. Методи регресійного аналізу.
3. Методи статистичних рішень.
4. Методи поділу у просторі ознак.
5. Метод дискримінантних функцій.

6. Метод потенційних функцій.
7. Метод потенціалів.
8. Метричні методи.
9. Метод найближчих сусідів.
- 10.Метод АВО.
- 11.Метод СВР.
- 12.Метод порівняння з еталоном.
- 13.Методи на основі м'яких обчислень.
- 14.Нейронні мережі.
- 15.Системи на основі нечіткої логіки.
- 16.Нейро-нечіткі мережі.
- 17.Методи, засновані на припущеннях про клас вирішальних функцій.
- 18.Метод групового обліку аргументів.
- 19.Метод стохастичної апроксимації.
- 20.Логічні методи.
- 21.Методи витягу асоціативних правил.
- 22.Методи побудови дерев розв'язків.
- 23.Лінгвістичні (структурні) методи.
- 24.Чіткий кластерний аналіз.
- 25.Нечіткий кластерний аналіз.

 **Завдання 2.** Для навчальної вибірки $x = \{<0\ 3\ 2>, <0,3\ 5\ 4>, <-1\ 1\ 0>, <1\ 6\ 3>\}$, екземплярам якої співставлено вихідний вектор $y = \{0\ 1\ 0\ 1\}$ визначити координати еталонів класів C^0 та C^1 на основі методу еталонів.

 **Завдання 3.** За заданими еталонами класів $C^0 = \{0\ 1\ 0\}$ та $C^1 = \{1\ 0\ 1\}$ для навчальної вибірки $x = \{<0\ 2\ 1>, <0\ 2\ 0>, <2\ 0\ 1>\}$ визначити номери класів екземплярів y на основі методу еталонів із використанням евклідової відстані.

 **Завдання 4.** Програмно реалізувати метод еталонів. За допомогою програмної реалізації методу еталонів дослідити його властивості при вирішенні практичних задач, використовуючи вибірки даних з ресурсу <http://archive.ics.uci.edu/ml/>.

Позначимо: S_h – обсяг (кількість екземплярів) навчальної вибірки, S_t – обсяг тестової вибірки, t_h – час навчання методу, $t_{p.h.}$ – час розпізнавання навчальної вибірки, E_h – помилка розпізнаван-

ня навчальної вибірки, $t_{\text{п.т.}}$ – час розпізнавання тестової вибірки, $E_{\text{п.т.}}$ – помилка розпізнавання тестової вибірки.

Результати досліджень занести до таблиці

Назва задачі	Характеристики задачі				Результати роботи методу				
	N	K	$S_{\text{н.}}$	$S_{\text{т.}}$	$t_{\text{н.}}$	$t_{\text{п.н.}}$	$E_{\text{н.}}$	$t_{\text{п.т.}}$	$E_{\text{п.т.}}$
Задача 1									
....									

Зробити висновки про вплив кількості екземплярів, ознак та класів у навчальній вибірці на час та помилку навчання.

Зробити висновки про вплив кількості екземплярів, ознак та класів у тестовій вибірці на час та помилку розпізнавання.



Завдання 5. Використовуючи вибірки даних для задач розпізнавання (наприклад, з ресурсу <http://archive.ics.uci.edu/ml/>) дослідити за допомогою пакету Matlab різні методи чіткої та нечіткої кластеризації. Результати досліджень занести до таблиці.

Назва задачі	Характеристики задачі			Результати роботи методів кластер-аналізу				
	N	S	K	Метод 1		...	Метод Z	
				t	Q		t	Q
Задача 1								
....								

Тут позначено: t – час, витрачений на кластеризацію вибірки, Q – кількість сформованих кластерів.

Зробити висновки про вплив кількості ознак, класів та екземплярів у вибірці на час та кількість виділених кластерів для методів кластер-аналізу.



2.9 Література до розділу

Огляд основних методів теорії розпізнавання образів наведено в [1–4, 6–8, 10–12, 16–18]. Методи чіткого кластерного аналізу розглянуто в [6–8]. Нечіткий кластер-аналіз описано в [5, 6–8, 13, 16].

РОЗДІЛ 3

ВІДБІР ІНФОРМАТИВНИХ ОЗНАК

Для синтезу розпізнавальних моделей використовують навчальну вибірку, що складається з великого набору ознак, які характеризують досліджуваний об'єкт або процес. Масиви даних великого розміру, як правило, містять надлишкові й неінформативні ознаки, які ускладнюють не тільки процес синтезу моделі, але й призводять до її надлишковості, що збільшує час класифікації за такою моделлю. Таким чином, при вирішенні задач розпізнавання образів важливим етапом є процес редукції вхідного набору ознак.

Складність вирішення задачи вибору максимально значимої комбінації ознак полягає в її комбінаторному характері. Використання повного перебору всіх можливих комбінацій при великій кількості ознак приводить до комбінаторного вибуху. Тому такий підхід на практиці виявляється неприйнятним, у результаті чого були розроблені методи скороченого перебору комбінацій ознак.

У даному розділі приводиться загальна постановка задачі відбору інформативних ознак, опис відомих підходів до її вирішення. Розглядається структура методів відбору ознак, а також критерії оцінювання індивідуальної та спільної значущості ознак.

3.1 Загальна постановка задачі відбору інформативних ознак для синтезу розпізнавальних моделей

Нехай задана вибірка вихідних даних у вигляді: $\langle X = \{X_1, X_2, \dots, X_L\} = \{X\}, Y = \{y_1, y_2, \dots, y_m\} = \{y\} \rangle$, де X – вихідний набір значень ознак, що характеризують досліджуваний об'єкт або процес, Y – масив значень вихідного параметра в заданій вибірці, $X_i = \{x_{ij}\}$ – i -та ознака у вихідній вибірці, $i = 1, 2, \dots, L$, x_{ij} – значення i -ї ознаки для j -го екземпляра вибірки, $j = 1, 2, \dots, m$, y_j – значення прогнозованого параметра для j -го екземпляра, L – загальна кількість ознак у вихідному наборі, m – кількість екземплярів вибірки.

Тоді задача відбору інформативних ознак може бути подана одним з таких способів.

1. Ідеалізована постановка: виділити комбінацію ознак X^* з вихідного масиву даних, при якій досягається мінімум заданого критерію оцінювання набору ознак:

$$J(X^*) = \min_{Xe \in XS} J(Xe),$$

де Xe – елемент множини XS ; $J(Xe)$ – критерій оцінювання значимості набору ознак Xe ; XS – множина всіх можливих комбінацій ознак, отримана з вихідного набору ознак X .

2. Класична постановка: відібрати з множини вхідних L ознак комбінацію, що складається не більш, ніж з L_0 ознак ($L_0 < L$), при якій досягається оптимум заданого критерію:

$$J(X^*) = \min_{Xe \in XS, |Xe| \leq L_0} J(Xe),$$

де $|Xe|$ – кількість елементів у множині Xe .

3. Знайти набір ознак мінімального розміру, що забезпечує досягнення заданого значення критерію оцінювання значимості набору ознак:

$$|X^*| = \min_{Xe \in XS, J(Xe) < \varepsilon} |Xe|,$$

де ε – задане значення критерію оцінювання набору ознак J .

Результатом виконання процедури відбору ознак є оптимальний набір ознак X^* , що має достатню інформативність. Інформативність ознак (набору ознак) – це величина, що відображає ступінь взаємозв'язку ознаки (набору ознак) із прогнозованим параметром. Інформативність комбінації ознак дорівнює сумі інформативності окремих ознак тільки при їхній незалежності. Якщо ознаки є залежними одна від одної, то інформативність набору не виражається через інформативність окремих ознак.

Таким чином, отриманий у результаті відбору ознак оптимальний набір X^* , маючи достатню інформативність, найбільш повно відображає досліджуваний об'єкт або процес. При цьому з вихідного набору X виключаються:

- незначущі ознаки – ознаки, що не впливають на вихідний параметр;
- надлишкові ознаки – ознаки, значення яких залежать від інших ознак. Такі ознаки не приводять до поліпшення якості прогнозування по синтезованій моделі.

3.2 Структура методів відбору ознак

Методи відбору ознак у своїй структурі містять такі складові:

- початкова точка пошуку;
- процедура пошуку оптимального набору ознак;
- стратегія оцінювання набору ознак;
- критерії зупинення.

Початкова точка пошуку – початкова комбінація ознак, з якої починається пошук набору ознак, з максимальною інформативністю. Вона генерується методами відбору ознак на етапі ініціалізації.

Як початкова точка пошуку можуть бути використані:

- порожня множина. При цьому метод на кожній ітерації додає ознаку (один або трохи), максимально поліпшуючий критерій оцінювання набору ознак;
- вихідний набір ознак, з якого ітеративно видаляються ознаки, виключення яких призводить до мінімального погіршення критерію оцінювання комбінації ознак;
- випадково згенерована або отримана за певними правилами множина.

Процедура пошуку оптимального набору ознак генерує новий набір ознак за певними правилами з метою оптимізації заданого критерію оцінювання комбінації ознак. Найбільш відомими є наступні методи генерації комбінацій ознак:

- *методи перебору* – послідовно генерують всі або більшість із $2^L - 1$ можливих комбінацій ознак. У випадку застосування недостатньої кількості методів, які перебирають не всі можливі комбінації, генерація нових рішень відбувається за певними правилами, що відкидає малоефективні комбінації;
- *евристичні методи* – група методів, що використовують евристичні процедури для визначення напрямку пошуку;
- *ранжування ознак* – підхід, при якому генерується одна комбінація ознак, що складається з ознак з максимальною індивідуальною значимістю;
- *методи випадкового пошуку* – генерують набори ознак випадковим або випадково спрямованим чином. Процедура випадкової генерації нових рішень триває доти, поки не буде знайдена

комбінація ознак, що задовольняє заданим критеріям, або поки не буде досягнута максимально припустима кількість ітерацій.

Стратегії оцінювання набору (комбінації) ознак залежать від типу використованого критерію:

– *фільтруючі методи* (filters) – методи, при яких відбір ознак відбувається незалежно від побудови моделі;

– *вбудовуючі методи* (wrappers) – методи, які в процесі пошуку оптимальної комбінації ознак синтезують моделі на основі оцінюваного набору ознак;

– *вбудовувані методи* (embedded methods) – методи, при яких процедура відбору ознак вбудовується в процедуру побудови оптимальної моделі.

Фільтруючі методи передбачають виключення неінформативних ознак з вихідного набору до побудови математичної моделі, що описує досліджуваний об'єкт або процес. При цьому використовується критерій оцінювання набору ознак, що не залежить від точності моделі, синтезованої на його основі.

Одним з переваг таких методів є те, що вони не мають потреби в повторному запуску якщо буде потреба синтезу нової моделі по вже відобраних ознаках.

Фільтри є обчислювально більш простими в порівнянні з іншими методами й ефективно можуть застосовуватися для відбору інформативних ознак з масивів даних дуже великого розміру.

Однак у результаті використання фільтруючих методів можуть бути отримані такі комбінації ознак, на основі яких не вдається побудувати модель, що забезпечує необхідну точність. Це викликано тим, що такі методи безпосередньо не пов'язані з математичною моделлю, що буде використовуватися для опису досліджуваного об'єкта, процесу або системи.

Критерії оцінювання ознак у фільтруючих методах, можуть бути класифіковані на:

– критерії оцінювання індивідуальної інформативності, що застосовуються при відборі ознак за допомогою ранжирування;

– критерії оцінювання групової інформативності ознак, що використовуються в більш складних пошукових процедурах.

Вбудовуючі методи оцінюють набір ознак за допомогою помилки прогнозування або класифікації по моделі, побудованої

на основі ознак з набору, що аналізується. Як синтезовані моделі можуть використовуватися регресійні, нечіткологічні, нейромрежеві, нейронечіткологічні та інші.

Використання помилок синтезованих моделей для оцінювання інформативності набору ознак є більше ресурсомісткою процедурою, оскільки синтез математичних моделей на основі оцінюваної комбінації ознак займає значно більший час у порівнянні з оцінюванням ознак шляхом застосування критеріїв оцінювання спільногого впливу ознак, використовуваних у фільтрах.

Як правило, такі методи приводять до кращих результатів у порівнянні з фільтруючими методами, оскільки вони орієнтовані на пошук інформативної комбінації ознак для конкретної моделі, що надалі буде застосовуватися на практиці.

Однак це приводить до зменшення гнучкості результатів у вигляді набору інформативних ознак. І у випадку прийняття рішення про зміну типу моделі, що використовується для опису досліджуваного об'єкта або процесу, необхідно буде запускати метод для повторного пошуку комбінації інформативних ознак, що відповідає новій моделі.

Вбудовувальні методи відбору ознак впроваджуються в процес побудови оптимальної моделі. Прикладами таких методів можуть служити методи ID3, C4.5 і CART, що використовуються для побудови дерев вирішальних правил. Вбудовувані методи, при побудові дерев вирішальних правил на кожному кроці використовують функцію оцінювання інформативності для вибору ознаки, за якою піде розбиття вихідної множини на підмножини.

Критерії зупинення визначають умови закінчення пошуку. В якості таких критеріїв можуть бути використані наступні:

- досягнення заданого прийнятного значення критерію оцінювання набору ознак;
- неможливість генерації нового набору ознак, що поліпшує досягнуте на поточній ітерації значення оцінки комбінації ознак;
- перевищення максимально можливої кількості ітерацій або часу функціонування методу.

Як правило виконання методів відбору ознак здійснюється у такій послідовності кроків.

Крок 1. Згенерувати початкове рішення.

Крок 2. Оцінити поточну комбінацію ознак.

Крок 3. Перевірити критерії закінчення пошуку. У випадку, якщо такі критерії задоволені, тоді виконати перехід до кроку 5.

Крок 4. Згенерувати нове рішення. Виконати перехід до кроку 2.

Крок 5. Зупинення.

У залежності від обраних способів реалізації процедур генерації і відбору та формування нових рішень, критеріїв інформативності та закінчення пошуку розрізняють конкретні методи відбору ознак.

3.3 Методи відбору інформативних ознак

Методи відбору ознак виділяють:

- переборні – здійснюють перегляд комбінацій ознак у визначеному порядку;
- евристичні – здійснюють перегляд комбінацій ознак із використанням певних гіпотез розробника про вибір більш перспективної комбінації ознак;
- методи ранжирування та класифікації ознак – певним чином визначають оцінки важливості ознак та групують їх за принципом пов’язаності, виділяючи з кожної групи подібних ознак найбільш інформативні ознаки;
- на основі випадкового (стохастичного) пошуку – використовують спеціальні оператори формування нових рішень на основі розглянутих раніше із елементами випадковості;

Класифікація методів відбору ознак наведена на рис. 3.1.

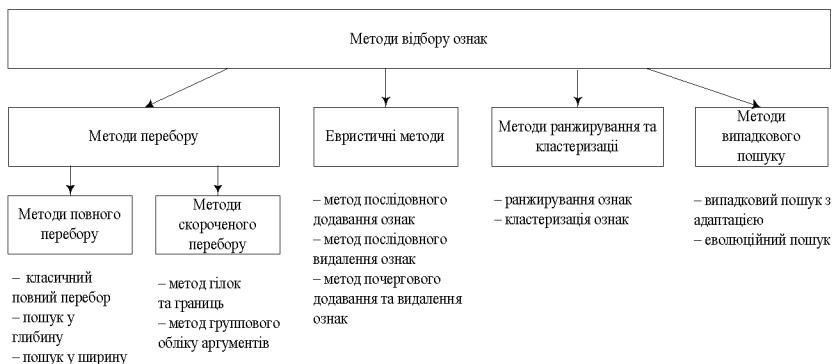


Рисунок 3.1 – Класифікація методів відбору ознак

Метод повного перебору (exhaustive search) аналізує всі можливі комбінації ознак, серед яких вибирає найкращу.

Крок 1. Згенерувати всі можливі набори ознак.

Крок 2. Оцінити всі отримані на попередньому кроці комбінації X_e , обчисливши для кожної з них значення критерію оцінювання набору ознак: $J(X_e)$.

Крок 3. Визначити оптимальне значення критерію оцінювання набору ознак: $J_{opt} = \min(J(X_e))$.

Крок 4. Визначити оптимальний набір ознак:

$$X^* = \arg \min J(X_e).$$

Крок 5. Зупинення.

Класичний повний перебір є найбільш простим для реалізації й гарантує одержання оптимального рішення. Недоліком такого методу є його обмеженість при рішенні практичних завдань, зумовлена величими обчислювальними витратами при його використанні.

Виконати повний перебір всіх можливих комбінацій ознак можливо за допомогою обходу дерева можливих наборів ознак, у якому вузли відповідають наборам ознак. Кореневий вузол відповідає порожньому набору. Кожний наступний набір утвориться шляхом приєднання деякої ознаки до попереднього батьківського вузла. З метою уникання появи в дереві вузлів, що відповідають однаковим наборам, що відрізняються тільки порядком ознак, до дочірніх вузлів додають тільки ті ознаки, номери яких перевищують максимальний номер ознак в батьківському наборі.

При використанні *методу пошуку в глибину* (depth-first search) обхід дерева можливих наборів ознак відбувається по напрямку від кореневого (батьківського) вузла до вузла-нащадка, що характеризується найбільшою кількістю ознак.

Для обходу дерева використовується рекурсивний виклик процедури нарощування поточного вузла Наростити(X_e), у такий спосіб дерево в явному виді не будеться. Процедура нарощування по черзі приєднує до поточного набору по одній означі, і для кожного з отриманих наборів спочатку обчислює значення критерію оцінювання, а потім викликає себе рекурсивно.

Крок 1. Ініціалізувати початковий набір ознак (кореневий вузол): $X_e = \emptyset$. Виконати: $X^* = X_e$, $J_{opt} = J(X^*)$, де X^* – оптимальний набір ознак; J_{opt} – оптимальне значення критерію оцінювання набору ознак.

Крок 2. Виконати нарощування поточного вузла дерева, викликавши процедуру Наростити (Xe).

Крок 3. Зупинення. Результатом виконання методу є оптимальний набір ознак X^* і оптимальне значення критерію оцінювання набору ознак $J_{\text{опт}}$, отримані в ході рекурсивного виконання процедури Наростити (Xe).

Процедура Наростити (Xe) виконується в наступній послідовності.

Крок 1. З метою усунення можливості генерації наборів ознак, що відрізняються тільки порядком ознак, визначити максимальний номер ознак \max у вихідному наборі Xe : $\max = \max_{X_i \in Xe} i$. У випадку,

якщо $\max = L$, тоді виконати перехід до кроku 9.

Крок 2. Встановити лічильник включених у переглянуті набори ознак: $i = \max + 1$.

Крок 3. Згенерувати новий набір ознак шляхом додавання i -ї ознак до поточного набору: $Xt = \{Xe \cup X_i\}$.

Крок 4. Оцінити комбінацію ознак Xt , обчисливши значення критерію оцінювання набору ознак: $J(Xt)$.

Крок 5. У випадку, якщо $J(Xt) < J_{\text{опт}}$, тоді виконати: $J_{\text{опт}} = J(Xt)$ і $X^* = Xt$.

Крок 6. Наростити вузол дерева Xt , для чого рекурсивно перейти до виконання процедури нарощування поточного вузла дерева: Наростити (Xt).

Крок 7. Виконати: $i = i + 1$.

Крок 8. У випадку, якщо $i \leq L$, тоді виконати перехід до кроku 3.

Крок 9. Зупинення.

При обході дерева можливих наборів ознак за допомогою *пошуку в ширину* (breadth-first search) відбувається послідовний перегляд вузлів дерева по рівнях, тобто комбінації ознак аналізуються по збільшенню їхнього розміру. Таким чином, на початку проглядаються всі можливі одноознакові комбінації, потім комбінації, що складаються із двох ознак і т. д.

Крок 1. Ініціалізувати початковий набір ознак: $Xe = \emptyset$. Виконати: $X^* = Xe$, $J_{\text{опт}} = J(X^*)$, де X^* – оптимальний набір ознак; $J_{\text{опт}}$ – оптимальне значення критерію оцінювання набору ознак.

Крок 2. Встановити лічильник аналізованого рівня дерева (кількості ознак в аналізованому наборі): $k = 1$.

Крок 3. Згенерувати всі можливі набори ознак розміром k .

Крок 4. Оцінити всі отримані на попередньому кроці комбінації Xe_k , обчисливши для кожної з них значення критерію оцінювання набору ознак: $J(Xe_k)$.

Крок 5. У випадку, якщо знайдено комбінацію ознак зі значенням критерію оцінювання, що краще поточного оптимального значення ($J(Xe_k) < J_{\text{опт}}$), тоді виконати: $J_{\text{опт}} = J(Xe_k)$ і $X^* = Xe_k$.

Крок 6. Збільшити лічильник кількості ознак в аналізованому наборі: $k = k + 1$.

Крок 7. У випадку, якщо $k \leq L$, тоді виконати перехід до кроку 3.

Крок 8. Зупинення.

Аналогічно методу повного перебору недоліком методів пошуку в глибину й пошуку в ширину є їхня обмеженість практичного застосування.

Пошук оптимальних комбінацій за допомогою дерева наборів ознак може бути легко перетворений від повного перебору рішень до скороченого, заснованому на використанні деякої додаткової інформації, отриманої з вихідного масиву даних.

До методів скороченого перебору відносяться метод гілок і границь (скорочений пошук у глибину) і метод групового урахування аргументів (скорочений пошук в ширину).

Скорочення кількості комбінацій, що перебираються, у методі гілок і границь (branch and bound method) досягається за рахунок відмови від нарощування гілки дерева у випадку, якщо вже є краща гілка. Тобто вузол, що відповідає набору ознак Xe , не нарощується, якщо значення критерію оцінювання набору ознак $J(Xe)$ виявиться гірше, ніж на найкращому із уже оцінених наборів меншої розмірності.

З метою пошуку більш оптимального набору спочатку ознаки ранжируються в порядку убування їхньої індивідуальної значимості.

Крок 1. Ініціалізувати початковий набір ознак (кореневий вузол): $Xe = \emptyset$. Виконати: $X^* = Xe$, $J_{\text{опт}} = J(X^*)$, де X^* – оптимальний набір ознак; $J_{\text{опт}}$ – оптимальне значення критерію оцінювання набору ознак.

Крок 2. Упорядкувати ознаки по убуванню інформативності.

Крок 3. Виконати нарощування поточного вузла дерева, викликавши процедуру Наростили (Xe).

Крок 4. Зупинення. Результатом виконання методу є оптимальний набір ознак X^* і оптимальне значення критерію оцінювання

набору ознак $J_{\text{опт}}$, отримані в ході рекурсивного виконання процедури Наростити (Xe).

Процедура Наростити(Xe) виконується в такій послідовності.

Крок 1. У випадку, якщо із числа вже оцінених наборів ознак найдеться така комбінація Xk , що $J(Xe) > J(Xk)$ і $|Xe| \geq |Xk|$, тоді виконати перехід до кроку 10.

Крок 2. З метою уникнення можливості генерації наборів ознак, що відрізняються тільки порядком ознак, визначити максимальний номер ознак \max у вихідному наборі Xe : $\max = \max_{X_i \in Xe} i$.

У випадку, якщо $\max = L$, тоді виконати перехід до кроку 10.

Крок 3. Встановити лічильник включених у переглянуті набори ознак: $i = \max + 1$.

Крок 4. Згенерувати новий набір ознак шляхом додавання i -ї ознаки до поточного набору: $Xt = \{Xe \cup X_i\}$.

Крок 5. Оцінити комбінацію ознак Xt , обчисливши значення критерію оцінювання набору ознак: $J(Xt)$.

Крок 6. У випадку, якщо $J(Xt) < J_{\text{опт}}$, тоді виконати: $J_{\text{опт}} = J(Xt)$ і $X^* = Xt$.

Крок 7. Наростити вузол дерева Xt , для чого рекурсивно перейти до виконання процедури нарощування поточного вузла дерева: Наростити(Xt).

Крок 8. Виконати: $i = i + 1$.

Крок 9. У випадку, якщо $i \leq L$, тоді виконати перехід до кроку 4.

Крок 10. Зупинення.

Застосування методу гілок і границь дозволяє скоротити час, необхідний для пошуку. Недоліком такого методу є послідовне додавання ознак до оцінюваного набору ознак, що часто приводить до виключення з розгляду комбінацій ознак, що володіють максимальною інформативністю.

У методі групового урахування аргументів (МГУА) на кожній t -й ітерації оцінюється не один набір ознак, а множина P_t ($|P_t| = N$) наборів, що називається t -м рядом. Для переходу від поточного P_t до наступного P_{t+1} ряду від кожного набору $Xe \in P_t$ породжується $L - t$ нових наборів шляхом приєднання одного з ознак, що не належать набору Xe . Зі згенерованих $N(L - t)$ наборів ознак у наступний ряд відирається не більше N наборів, кращих за значенням критерію

оцінювання набору ознак. Таким чином, на кожній ітерації розмір наборів ознак збільшується на одиницю.

Кількість наборів ознак на кожній ітерації N називається шириною пошуку. Зокрема при $N=1$ метод групового обліку аргументів являє собою евристичний метод послідовного додавання ознак.

Крок 1. Встановити лічильник ітерацій (рядів): $t = 1$. Ініціалізувати початкову множину наборів ознак одноознаковими комбінаціями: $P_t = \{Xe \mid Xe = \{X_i\}, i = 1, 2, \dots, L\} \dots$ Виконати: $X^* = \emptyset$, $J_{\text{опт}} = \infty$, де X^* – оптимальний набір ознак; $J_{\text{опт}}$ – оптимальне значення критерію оцінювання набору ознак.

Крок 2. Обчислити значення критерію оцінювання кожного набору ознак з t -го ряду: $J(Xe)$, $Xe \in P_t$.

Крок 3. У випадку, якщо найдеться такий набір $Xe \in P_t$, що $J(Xe) < J_{\text{опт}}$, тоді виконати: $J_{\text{опт}} = J(Xe)$ і $X^* = Xe$.

Крок 4. Відсортувати ряд P_t по зростанню значення критерію оцінювання набору ознак.

Крок 5. Залишити в ряді P_t N кращих наборів ознак. Всі інші набори видалити.

Крок 6. Збільшити лічильник ітерацій (рядів): $t = t + 1$.

Крок 7. Перевірити критерій закінчення пошуку (досягнення максимально можливого розміру набору ознак L_0 , перевищення припустимої кількості ітерацій T і т. п.). У випадку, якщо такі критерії досягнуті, тоді виконати перехід до кроку 10.

Крок 8. Згенерувати наступний ряд. Для цього сформувати для кожного набору $Xe \in P_{t-1}$ $L - t$ нових наборів шляхом приєднання одного з ознак, що не належать набору Xe .

Крок 9. Виконати перехід до кроку 2.

Крок 10. Зупинення.

Такий метод дозволяє позбутися необхідності оцінювання кожної з $2^L - 1$ можливих комбінацій ознак, але є більше складним у порівнянні з методом гілок і границь.

Евристичні методи відбору ознак використовують жадібні стратегії (greedy strategy) для додавання або видалення ознак на кожній ітерації. До евристичних методів відносяться метод послідовного додавання ознак, метод послідовного видалення ознак, а також метод почергового додавання й видалення ознак.

У методі послідовного додавання ознак (forward selection) на основі оптимального набору ознак, знайденого на попередній іте-

рації X_{t-1}^* , на поточній ітерації формуються всі можливі комбінації ознак $Xe_{t,k}$ шляхом додавання однієї ознаки, ще не включеного в набір X_{t-1}^* . У наступній ітерації формування нових рішень відбувається на основі оптимального набору $X_t^* = \operatorname{argmin} J(Xe_{t,k})$, знайденого на поточній ітерації.

Крок 1. Встановити лічильник ітерацій (часу): $t = 0$. Ініціалізувати початковий набір ознак: $Xe_t = \emptyset$. Виконати: $X^* = Xe_t$, $J_{\text{опт}} = J(X^*)$, де X^* – оптимальний набір ознак; $J_{\text{опт}}$ – оптимальне значення критерію оцінювання набору ознак.

Крок 2. Збільшити лічильник ітерацій: $t = t + 1$. Згенерувати всі можливі нові набори ознак шляхом додавання однієї ознаки до набору X^* , одержавши в такий спосіб $M = L - |X^*|$ пробних комбінацій ознак $Xe_{t,k}$, $k = 1, 2, \dots, M$, де L – кількість ознак у вихідному масиві даних; $|X^*|$ – кількість відібраних ознак в оптимальному наборі, отриманому на попередньому кроці.

Крок 3. Оцінити кожний з $Xe_{t,k}$ набір ознак, розрахувавши значення критерію оцінювання набору ознак $J(Xe_{t,k})$.

Крок 4. Перевірити критерій закінчення пошуку.

Як такі критерії можуть бути використані:

- виникнення ситуації, при якій всі згенеровані на поточній ітерації набори ознак гірше оптимального набору, отриманого раніше: $J_{\text{опт}} < \min(J(Xe_{t,k}))$;

- досягнення максимально можливого розміру набору ознак L_0 ;

- перевищення припустимої кількості ітерацій T .

У випадку, якщо такі критерії досягнуті, тоді виконати переход до кроку 7.

Крок 5. Виконати: $J_{\text{опт}} = \min(J(Xe_{t,k}))$ і $X^* = \operatorname{argmin} J(Xe_{t,k})$.

Крок 6. Перейти до виконання кроку 2.

Крок 7. Зупинення.

Метод послідовного додавання ознак є простим у реалізації, а також не вимагає значних тимчасових витрат при його використанні: обчислювальна складність методу $O(L^2)$, що значно нижче, ніж у методів повного перебору.

Недолік такого методу викликаний неоптимальністю жадібної стратегії пошуку: при використанні методу послідовного додавання ознак часто в оптимальний набір включаються надлишкові ознаки.

У методі послідовного видалення ознак (backward selection) на основі оптимального набору ознак, знайденого на попередній ітерації X^*_{t-1} , на поточній ітерації формуються всі можливі комбінації ознак $Xe_{t,k}$ шляхом видалення однієї ознаки, що з набору X^*_{t-1} . У наступній ітерації формування нових рішень відбувається на основі оптимального набору $X_t^* = \operatorname{argmin} J(Xe_{t,k})$, знайденого на поточній ітерації.

Таким чином, у методі послідовного видалення ознак на кожній ітерації виключаються ознака, що мінімально погіршує критерій оцінювання набору ознак.

Крок 1. Встановити лічильник ітерацій (часу): $t = 0$. Ініціалізувати початковий набір ознак комбінацією із всіх можливих ознак, що містяться у вихідному наборі даних: $Xe_t = \{X_1, X_2, \dots, X_L\} \dots$ Виконати: $X^* = Xe_t, J_{\text{опт}} = J(X^*)$, де X^* – оптимальний набір ознак; $J_{\text{опт}}$ – оптимальне значення критерію оцінювання набору ознак.

Крок 2. Збільшити лічильник ітерацій: $t = t + 1$. Згенерувати всі можливі нові набори ознак шляхом видалення однієї ознаки з набору X^* , одержавши в такий спосіб $|X^*|$ пробних комбінацій ознак $Xe_{t,k}, k = 1, 2, \dots, |X^*|$, де $|X^*|$ – кількість відібраних ознак в оптимальному наборі, отриманому на попередньому кроці.

Крок 3. Оцінити кожний з $Xe_{t,k}$ набір ознак, розрахувавши значення критерію оцінювання набору ознак $J(Xe_{t,k})$.

Крок 4. Перевірити критерій закінчення пошуку.

Як такі критерії можуть бути використані:

- виникнення ситуації, при якій значення критеріїв оцінювання всіх згенерованих на поточній ітерації наборів ознак гірше максимально припустимої величини ε , заданої користувачем: $\min(J(Xe_{t,k})) > \varepsilon$. Величина (ε) визначає мінімально прийнятне значення критерію оцінювання набору ознак;

- виникнення ситуації, при якій всі згенеровані на поточній ітерації набори ознак є значно гірше оптимального набору, отриманого раніше: $|\min(J(Xe_{t,k})) - J_{\text{опт}}| > \xi$, де ξ – величина, що визначає максимально можливе погіршення значення критерію оцінювання набору ознак на одній ітерації;

- досягнення максимально можливого розміру набору ознак L_0 ;
- перевищенння припустимої кількості ітерацій T .

У випадку, якщо такі критерії досягнуті, тоді виконати перевід до кроку 7.

Крок 5. Виконати: $J_{\text{опт}} = \min(J(Xe_{t,k}))$ і $X^* = \operatorname{argmin} J(Xe_{t,k})$.

Крок 6. Перейти до виконання кроку 2.

Крок 7. Зупинення.

Метод послідовного видалення ознак також, як і попередній метод, простий у реалізації. До основного недоліку такого методу варто віднести неоптимальність жадібної стратегії. Важливо відзначити, що метод послідовного видалення ознак працює повільніше в порівнянні з методом послідовного додавання ознак, оскільки на початкових ітераціях необхідно оцінювати набори ознак, що складаються із всіх або майже всіх ознак. Такий метод застосовують у випадках, коли відомо, що інформативних ознак значно більше, ніж малоінформативних або надлишкових.

Метод послідовного додавання та видалення ознак (combined selection) сполучає в собі ідеї двох розглянутих раніше методів, що діють протилежно, у результаті чого виходить нежадібна стратегія пошуку.

Ідея такого методу полягає в тому, щоб використовувати стратегію додавання ознак доти, поки не виникне ситуація, при якій збільшення кількості ознак в оптимальному наборі не приводить до поліпшення значення критерію оцінювання набору ознак $J(X^*)$. Після цього запускається процедура видалення ознак, що припиняє своє функціонування за тих самих умов, що й процедура додавання ознак. Процедури додавання та видалення ознак чергуються доти, поки значення критерію $J(X^*)$ в оптимальних точках X^* не перестане поліпшуватися.

Крок 1. Встановити лічильник ітерацій (часу): $t = 0$. Ініціалізувати початковий набір ознак: $Xe_t = \emptyset$. Виконати: $X^* = Xe_t$, $J_{\text{опт}} = J(X^*)$, де X^* – оптимальний набір ознак; $J_{\text{опт}}$ – оптимальне значення критерію оцінювання набору ознак.

Крок 2. Виконати процедуру додавання ознак.

Крок 2.1. Збільшити лічильник ітерацій: $t = t + 1$. Згенерувати всі можливі нові набори ознак шляхом додавання однієї ознаки до набору X^* , одержавши в такий спосіб $M = L - |X^*|$ пробних комбінацій ознак $Xe_{t,k}$, $k = 1, 2, \dots, M$, де L – кількість ознак у вихідному масиві даних; $|X^*|$ – кількість відібраних ознак в оптимальному наборі, отриманому на попередньому кроці.

Крок 2.2. Оцінити кожний з $X_{e_t, k}$ набір ознак, розрахувавши значення критерію оцінювання набору ознак $J(X_{e_t, k})$.

Крок 2.3. Перевірити критерій закінчення виконання процедури додавання ознак. У випадку, якщо виникла ситуація, при якій всі згеровані на поточній ітерації набори ознак гірше оптимального набору, отриманого раніше: $J_{\text{опт}} < \min(J(X_{e_t, k}))$, тоді завершити процедуру додавання ознак і перейти до кроku 3.

Крок 2.4. Виконати: $J_{\text{опт}} = \min(J(X_{e_t, k}))$ і $X^* = \operatorname{argmin} J(X_{e_t, k})$.

Крок 2.5. Виконати перехід до кроku 2.1.

Крок 3. Виконати процедуру видалення ознак.

Крок 3.1. Збільшити лічильник ітерацій: $t = t + 1$. Згенерувати всі можливі нові набори ознак шляхом видалення однієї ознаки з набору X^* , одержавши в такий спосіб $|X^*|$ пробних комбінацій ознак $X_{e_t, k}, k = 1, 2, \dots, |X^*|$.

Крок 3.2. Оцінити кожний з $X_{e_t, k}$ набір ознак, розрахувавши значення критерію оцінювання набору ознак $J(X_{e_t, k})$.

Крок 3.3. Перевірити критерій закінчення виконання процедури видалення ознак.

Як такі критерії можуть бути використані:

– виникнення ситуації, при якій значення критеріїв оцінювання всіх згенерованих на поточній ітерації наборів ознак гірше максимально припустимої величини ϵ , заданої користувачем: $\min(J(X_{e_t, k})) > \epsilon$. Величина (ϵ) визначає мінімально прийнятне значення критерію оцінювання набору ознак;

– виникнення ситуації, при якій всі згенеровані на поточній ітерації набори ознак є значно гірше оптимального набору, отриманого раніше: $|\min(J(X_{e_t, k})) - J_{\text{опт}}| > \xi$, де ξ – величина, що визначає максимальне можливе погіршення значення критерію оцінювання набору ознак на одній ітерації;

У випадку, якщо критерій закінчення виконання процедури видалення ознак виконуються, тоді завершити процедуру видалення ознак і перейти до кроku 4.

Крок 3.4. Виконати: $J_{\text{опт}} = \min(J(X_{e_t, k}))$ і $X^* = \operatorname{argmin} J(X_{e_t, k})$.

Крок 3.5. Виконати перехід до кроku 3.1.

Крок 4. Перевірити критерій закінчення пошуку (досягнення максимально можливого розміру набору ознак L_0 , перевищення припустимої кількості ітерацій T і т. п.). У випадку, якщо такі критерії досягнуті, тоді виконати перехід до кроku 6.

Крок 5. Перейти до виконання кроку 2.

Крок 6. Зупинення.

Метод додавання й видалення ознак є більше складним у реалізації й працює довше в порівнянні з методами послідовного додавання й послідовного видалення ознак окремо й також не гарантує оптимальності знайденого рішення. Однак рішення, отримані за допомогою такого методу, як правило, виявляються більш оптимальними в порівнянні з рішеннями, отриманими шляхом застосування методів послідовного додавання й послідовного видалення ознак.

Методи ранжирування й кластеризації не використовують критерії оцінювання спільноговпливу набору ознак на вихідний параметр. Так при ранжируванні використовуються критерії оцінювання індивідуальної значимості ознак. У методах кластеризації застосовуються метрики відстані на ознаках.

Для відбору ознак також використовується *ранжирування ознак*. При такому підході виконується сортування ознак за обраним критерієм індивідуальної значимості. Після індивідуального оцінювання кожної ознаки і їхнього сортування відбувається вибір певної кількості ознак, індивідуальна значимість яких задоволяє заданим умовам.

Для оцінювання індивідуальної значимості ознак можуть використовуватися критерії кореляції (парний, Фехнера, знаків), інформаційний критерій, ентропія ознак.

Крок 1. Обчислити значення критерію оцінювання індивідуальної значимостіожної ознаки у вихідному наборі ознак.

Крок 2. Упорядкувати ознаки по убуванню інформативності.

Крок 3. Сформувати комбінацію з перших L_0 ознак або з ознак, значення індивідуальної значимості яких вище граничного. Отримана комбінація вважається оптимальним набором.

Крок 4. Зупинення.

Перевагою такого підходу є простота реалізації, а недоліком – можливість застосування тільки у випадку, якщо ознаки у вихідному наборі є статистично незалежними. Як правило, при рішенні практичних завдань ознаки статистично залежать друг від друга, у результаті чого при використанні такого підходу для відбору ознак виходять комбінації, що містять надлишкові ознаки, отже, такі комбінації виявляються далеко не оптимальними.

Виділити максимальну значиму комбінацію можна за допомогою **кластеризації ознак** (unsupervised learning for feature selection).

Методи кластеризації застосовують для розбиття вибірки на кластери, що складаються зі схожих екземплярів, і виділення в кожній групі одного найбільш типового екземпляра. Аналогічні дії можна виконати не над екземплярами, а над ознаками, якщо ввести функцію відстані на ознаках.

Крок 1. Для кожної ознаки у вихідному наборі обчислити відстань d_{ab} від нього до інших ознак.

Крок 2. На основі розрахованих на попередньому кроці відстаней d_{ab} між ознаками виконати кластеризацію ознак, згрупувавши їх по кластерах.

Крок 3. Виділити типових представників у кожному кластері.

Крок 4. Сформувати інформативну комбінацію з ознак, виділених на попередньому етапі.

Крок 5. Зупинення.

Недоліком кластеризації ознак є можливість існування кластерів, що цілком складаються з неінформативних ознак, у результаті чого типові представники таких кластерів можуть увійти в комбінацію ознак, що вважається найбільш інформативною.

Кластеризацію ознак доцільно застосовувати на початкових етапах інших методів відбору ознак для формування груп (кластерів) схожих ознак. Після цього в процесі пошуку оптимальної комбінації занадто схожим ознакам, що належать одному класу, забороняється входити в той самий набір.

У методі **випадкового пошуку з адаптацією** (adaptive stochastic search) на кожній t -й ітерації використовується популяція P_t , що складається з N рішень. Формування нових рішень відбувається шляхом випадкової генерації N наборів ознак залежно від розподілу ймовірностей включення ознак у генеровані набори.

Ідея адаптації полягає у тому, щоб при генерації нових наборів ознак імовірність включення в них ознак, які частіше входять у кращі набори, була більшою в порівнянні з імовірністю мало використовуваних у кращих наборах ознак.

Крок 1. Встановити лічильник ітерацій (часу): $t = 0$. Встановити рівніймовірності включення ознак у генеровані набори: $p_1 = p_2 = \dots = p_L = 1/L$, де p_i – ймовірність включення i -ї ознаки у формований набір.

Крок 2. Ініціалізувати початкову множину наборів ознак P_t , згенерувавши N наборів ознак відповідно до розподілу $\{p_1, p_2, \dots, p_L\} \dots$ Обчислити значення критерію оцінювання кожного набору ознак з P_t : $J(Xe)$, $Xe \in P_t$. Виконати: $J_{\text{опт}} = \min(J(Xe))$ і $X^* = \operatorname{argmin} J(Xe)$, де X^* – оптимальний набір ознак; $J_{\text{опт}}$ – оптимальне значення критерію оцінювання набору ознак.

Крок 3. Визначити найкращу $X_{\min} = \operatorname{argmin} J(Xe)$ і найгіршу $X_{\max} = \operatorname{argmax} J(Xe)$ комбінації ознак у поточній популяції. У випадку, якщо $J(X_{\min}) < J_{\text{опт}}$, тоді виконати: $J_{\text{опт}} = J(X_{\min})$ і $X^* = X_{\min}$.

Крок 4. Встановити $s = 0$, де s – сумарне значення зміни ймовірностей включення ознак у генеровані набори.

Крок 5. Зменшити ймовірності включення в нові набори ознак, що входять у найгірший набір X_{\max} .

Крок 5.1. Якщо $p_i > h$ ($X_i \in X_{\max}$), тоді виконати: $p_i = p_i - h$ і $s = s + h$, де $h << 1/L$ – крок зміни ймовірності включення ознак у генеровані набори.

Крок 5.2. Якщо $p_i \leq h$ ($X_i \in X_{\max}$), тоді виконати: $p_i = 0$ і $s = s + p_i$.

Крок 6. Збільшити ймовірності включення в нові набори ознак, що входять у найкращий набір X_{\min} : $p_i = p_i + h/|X_{\min}|$ ($X_i \in X_{\min}$).

Крок 7. Перевірити критерій закінчення пошуку (досягнення прийнятного значення критерію оцінювання набору ознак $J(Xe)$, перевищення припустимої кількості ітерацій T і т. п.). У випадку, якщо такі критерії досягнуті, тоді виконати перехід до кроку 10.

Крок 8. Збільшити лічильник ітерацій: $t = t + 1$.

Крок 9. Одержані нову множину наборів ознак P_t , згенерувавши N наборів ознак відповідно до розподілу $\{p_1, p_2, \dots, p_L\} \dots$ Обчислити значення критерію оцінювання кожного набору ознак з P_t : $J(Xe)$, $Xe \in P_t$. Виконати перехід до кроку 3.

Крок 10. Зупинення.

Кількість наборів ознак N у кожній ітерації повинне бути по можливості мінімальним, але одночасно достатнім для того, щоб

крацій набір X_{\min} був близький до оптимального, а гірший X_{\max} – до самого неінформативного набору ознак.

Параметр h , що задає ступінь адаптації, вибирається таким чином, щоб імовірність включення ознаки не могла стати нульовою. При $h = 0$ метод випадкового пошуку з адаптацією перетвориться до неадаптивного випадкового пошуку.

Перевага методу випадкового пошуку з адаптацією полягає в тому, що в більшості випадків він знаходить більш оптимальні набори ознак у порівнянні з методом почергового додавання й видалення ознак. Недоліком такого методу є досить повільна збіжність.

Для відбору інформативних ознак з вихідного масиву, що містить L ознак, за допомогою **методів еволюційного пошуку** (evolutionary search) комбінація ознак-рішення (хромосома) подається бітовим рядком розміру L . Якщо біт хромосоми приймає одиничне значення, то відповідна йому ознака вважається інформативною і враховується при оцінюванні набору ознак, що відповідає хромосомі. У протилежному випадку, коли біт приймає нульове значення, ознака вважається неінформативною і не використовується при оцінюванні комбінації ознак.

Перевага такого подання полягає в тому, що класичні еволюційні оператори скрещування й мутації можуть бути застосовані для відбору ознак без внесення в них яких-небудь змін.

Для рішення задачі відбору інформативних ознак еволюційний пошук здійснюється шляхом виконання таких кроків.

Крок 1. Встановити лічильник ітерацій (часу): $t = 0$.

Крок 2. Згенерувати початкові комбінації ознак у вигляді хромосом – бітових рядків H_j розмірності L , де $j = 1, 2, \dots, N$ – номер хромосоми в популяції (множині рішень); N – кількість згенерованих хромосом; L – кількість ознак.

Крок 3. Обчислити значення фітнес-функції хромосом $H_j \in P_t$. У якості фітнес-функції використовується значення критерію оцінювання набору ознак Xe_j , що відповідає хромосомі H_j .

Крок 4. Перевірити умови закінчення пошуку, у якості яких можуть бути використані: досягнення обмеження часу, кількості ітера-

цій, прийнятного значення критерію оцінювання набору ознак. Якщо критерії закінчення вдоволені, тоді перейти до кроку 9.

Крок 5. Збільшити лічильник ітерацій (часу): $t = t + 1$.

Крок 6. Згенерувати нові рішення шляхом застосування еволюційних операторів схрещування й мутації до особин поточного покоління.

Крок 7. Обчислити значення фітнес-функції хромосом $H_j \in P_t$.

Крок 8. Відібрати N кращих хромосом для переходу в наступне покоління. Виконати переход до кроку 4.

Крок 9. Зупинення.

Перевагою еволюційного пошуку є те, що він має можливості для виходу з локальних оптимумів і пристосований для знаходження нових рішень за рахунок об'єднання кращих рішень, отриманих на різних ітераціях. Крім того, еволюційний пошук адаптується до особливостей цільової функції. Створені в процесі схрещування, нові рішення тестують усе більше широкі області простору ознак і переважно розташовуються в області оптимуму. Відносно рідкі мутації перешкоджають виродженню популяції, що рівносильно рідкому, але не припиняєму пошуку оптимуму у всіх інших областях простору ознак.

Недоліками еволюційного пошуку є відносно повільна збіжність і залежність від початкових умов пошуку.

У табл. 3.1 наведено порівняльну характеристику проаналізованих методів відбору ознак.

3.4 Критерії оцінювання інформативності ознак

Критерії, що використовуються для оцінювання інформативності ознак, можуть бути класифіковані на критерії оцінювання індивідуальної інформативності ознак (оцінюють вплив однієї окремо взятої ознаки на вихідний параметр) і критерії оцінювання групової інформативності (оцінюють спільний вплив набору ознак на вихідний параметр).

Критерії оцінювання індивідуальної інформативності ознак застосовуються, як правило, при відборі ознак за допомогою ранжирування або на початкових етапах більше складних методів відбору ознак. Індивідуальна інформативність ознак може бути оцінена за допомогою таких критеріїв.

1. Коефіцієнт парної кореляції $r_{\text{п}}(X_i)$ ознаки X_i і вихідного параметра Y дозволяє оцінити наявність лінійного зв'язку між ними й розраховується за формулою:

$$r_{\text{п}}(X_i) = \frac{\sum_{p=1}^m (x_{ip} - \bar{x}_i)(y_p - \bar{y})}{\sqrt{\sum_{p=1}^m (x_{ip} - \bar{x}_i)^2 \sum_{p=1}^m (y_p - \bar{y})^2}},$$

де $\bar{x}_i = \frac{1}{m} \sum_{p=1}^m x_{ip}$ й $\bar{y} = \frac{1}{m} \sum_{p=1}^m y_p$ – середні значення ознаки X_i і відгуку Y . Для оцінювання інформативності ознак використовується модуль значення коефіцієнта парної кореляції.

2. Коефіцієнт кореляції знаків $r_3(X_i)$ визначається за формулою:

$$r_3(X_i) = \frac{C_{x,y}^+ - C_{x_i}^+ C_y^+}{\sqrt{C_{x_i}^+ C_y^+ (1 - C_{x_i}^+) (1 - C_y^+)}},$$

де $C_{x,y}^+$ – кількість збігів позитивних знаків різниць $(x_{ip} - \bar{x}_i)$ і $(y_p - \bar{y})$, поділена на m – кількість екземплярів у вибірці; $C_{x_i}^+$, C_y^+ – частки від розподілу кількості позитивних знаків різниць $(x_{ip} - \bar{x}_i)$ і $(y_p - \bar{y})$ на m для кожної змінної X_i і Y окремо.

3. Коефіцієнт кореляції Фехнера $r_{\Phi}(X_i)$ може бути розрахований таким чином:

$$r_{\Phi}(X_i) = \frac{C_i - D_i}{C_i + D_i} = \frac{2C_i}{m} - 1,$$

де C_i – кількість збігів однакових, як позитивних, так і негативних знаків різниць $(x_{ip} - \bar{x}_i)$ і $(y_p - \bar{y})$ для i -ї ознаки відповідно; D_i – кількість розбіжностей знаків різниць $(x_{ip} - \bar{x}_i)$ і $(y_p - \bar{y})$ для i -ї ознаки.

Таблиця 3.1 – Порівняльна характеристика методів відбору ознак

Критерій порівняння	Методи відбору ознак					
	Методи перевороту		Евристичні методи		Методи ранжування та класифікації	
	Метод повного перевороту	Методи скороченого перевороту	Метод послідовного додавання та видалення ознак	Метод послідовного поєднання та видалення ознак	Ранжування ознак	Випадковий пошук з алгоритмом Европейського пошуку
Оптимальність пошуку (оптимальність знайдених рішень)	Класичний головний генетичний алгоритм	Популяційну генетичну завданнями	Метод глюок і гранінь	Метод МГВА	Кластеризація ознак	Кластеризація ознак
Кількість рішень на кожній ітерації	Оптимальний	Оптимальний	Субоптимальний	Субоптимальний	Субоптимальний	Субоптимальний
Дегерманованість пошуку	Одне	Одне	Одне	Одне	Одне	Одне
Теоретичне обґрунтування	Дегерманіваний	Дегерманіваний	Дегерманіваний	Дегерманіваний	Дегерманіваний	Дегерманіваний
Відносна складність реалізації	Низька	Низька	Висока	Середня	Середня	Середня
Принцип формування початкового рішення (початкова точка пошуку)	Порожня множина	Порожня множина	Порожня множина	Декілька (N) одиниць	Множина, що містить всі ознаки викінченої набору	Порожня множина
Критерій оптимізовання інформативності ознак	Групові	Групові	Групові	Групові	Групові	Групові
Оригінальність спільність	О(2^d)	О(2^d)	О(2^d)	О(L^d)	О(L^d)	О(NLT)

4. Дисперсійне (кореляційне) відношення $\eta(X_i)$ є мірою зв'язку вихідного параметра Y і ознаки X_i , що враховує зміну величини умовного математичного сподівання $M(Y/X_i)$:

$$\eta(X_i) = \sqrt{\frac{DM(Y/X_i)}{DY}} = \sqrt{\frac{\frac{1}{m} \sum_{k=1}^{N_i} n_k (\bar{y}_k - \bar{y})^2}{\frac{1}{m} \sum_{p=1}^m (\bar{y}_p - \bar{y})^2}} = \sqrt{\frac{\sum_{k=1}^{N_i} n_k (\bar{y}_k - \bar{y})^2}{\sum_{p=1}^m (\bar{y}_p - \bar{y})^2}},$$

де $DM(Y/X_i)$ – дисперсія умовного математичного сподівання $M(Y/X_i)$ – характеризує ту частину коливань значень вихідної змінної Y , що викликана впливом вихідної змінної X_i ; n_k – кількість значень ознаки X_i , що потрапили в k -й інтервал групування; N_i – кількість інтервалів з діапазону зміни i -ї ознаки, в які він може потрапити; $\bar{y}_k = \frac{1}{n_k} \sum_{j=1}^{n_k} y_{k,j}$ – середнє значення вихідного параметру для k -го інтервалу діапазону зміни i -ї ознаки; $y_{k,j}$ – j -те значення вихідного параметра Y за умови, якщо вихідна змінна X_i потрапить в k -й інтервал; $\bar{y} = \frac{1}{m} \sum_{p=1}^m \bar{y}_p$ – середнє значення вихідної змінної Y ; DY – дисперсія вихідного параметра Y .

5. Коефіцієнт зв'язку $\xi(X_i)$ у порівнянні з дисперсійним відношенням дозволяє одержати більш об'єктивну оцінку інформативності ознаки X_i , оскільки характеризує залежність вихідного параметра Y від ознаки X_i не тільки в результаті зміни величини умовного математичного сподівання $M(Y/X_i)$, але й у результаті зміни величини умовної дисперсії $D(Y/X_i)$:

$$\xi(X_i) = \sqrt[4]{\frac{D(M(Y/X_i))^2}{DY^2 - 2B}},$$

$$D(M(Y/X_i))^2 = \frac{1}{m} \sum_{k=1}^{N_i} n_k \left((\bar{y}_k)^2 - \bar{y}^2 \right)^2,$$

$$DY^2 = \frac{1}{m} \sum_{p=1}^m \left(y_p^2 - \overline{y^2} \right)^2,$$

$$\overline{y^2} = \frac{1}{m} \sum_{p=1}^m y_p^2,$$

$$\overline{y^2}_k = \frac{1}{n_k} \sum_{j=1}^{n_k} y_{k,j}^2,$$

$$B = \text{cov}\left(\left(M(Y/X_i)\right)^2, \left(D(Y/X_i)\right)\right) = \\ = \frac{1}{m} \sum_{k=1}^{N_i} n_k \left(\left(\overline{y}_k\right)^2 - \overline{y^2} \right) \left(\overline{y^2}_k - \left(\overline{y}_k\right)^2 - \frac{1}{m} \sum_{a=1}^{N_i} n_a \left(\overline{y^2}_a - \left(\overline{y}_a\right)^2 \right) \right).$$

6. *Інформаційний критерій* $I(X_i)$ – передбачає використання кількості інформації, що одержує система в процесі розпізнавання об'єктів у результаті використання оцінюваної ознаки:

$$I(X_i) = - \sum_{l=1}^{N_y} p_l \log_2 p_l + \sum_{l=1}^{N_y} p_l \sum_{q=1}^{N_y} \sum_{k=1}^{N_i} p_{lk} p_{kq} \log_2 p_{kq},$$

де N_y – кількість інтервалів з діапазону зміни вихідного параметра y ; $p_l = \frac{n_l}{m}$ – імовірність попадання значення вихідного параметра в l -й інтервал діапазону його зміни; n_l – кількість значень вихідного параметра, що належать l -му інтервалу діапазону його зміни; $p_{lk} = p(X_i \in [x_{i,k}; x_{i,k+1}) / y \in [y_l; y_{l+1}]) = \frac{n_{lk}}{n_l}$ – умовна імовірність попадання значення i -ї ознаки в k -й інтервал діапазону його зміни за умови, що вихідний параметр у потрапить в l -й інтервал; n_{lk} – кількість значень i -ї ознаки належних k -му інтервалу діапазону його зміни за умови, що значення вихідного параметра у належить l -му інтервалу;

$p_{kq} = \frac{n_{kq}}{n_k}$ – умовна імовірність влучення

значення вихідного параметра y в q -й інтервал, за умови, що i -а ознака потрапить в k -й інтервал; n_{kq} – кількість значень вихідного параметра y , що належать q -му інтервалу діапазону його зміни за умови, що значення i -ї ознаки належить k -му інтервалу діапазону його зміни; n_k – кількість значень i -ї ознаки, що належать k -му інтервалу діапазону її зміни.

Перевагою інформаційного підходу для оцінювання значимості ознак є наявність строгого математичного обґрунтування. Недолік обчислення інформаційного критерію складається в необхідності дискретизації ознак, що приймають безперервні значення.

7. *Теоретико-інформаційний критерій $TI(X_i)$* – передбачає використання кількості інформації, що одержує система в процесі розпізнавання об'єктів у результаті використання оцінюваної ознаки:

$$TI(X_i) = \sum_{a=1}^{N_y} \sum_{b=1}^{N_i} p_{ab} \log_2 \frac{p_{ab}}{p_a p_b},$$

де $p_{ab} = p(X_i \in [x_{i,b}; x_{i,b+1}), y \in [y_a; y_{a+1}]) = \frac{n_{ab}}{m}$ – імовірність одночасного влучення значення i -ї ознаки в b -й інтервал діапазону його зміни й вихідного параметра y потрапить в a -й інтервал; n_{ab} – кількість екземплярів вибірки, для яких виконуються умови: $X_i \in [x_{i,b}; x_{i,b+1})$ і $y \in [y_a; y_{a+1})$; p_a – імовірність попадання значення вихідного параметра в a -й інтервал діапазону її зміни; p_b – імовірність попадання значення i -ї ознаки в b -й інтервал діапазону її зміни.

8. *Ентропія ознаки $e(X_i)$* також використовує інформаційний підхід до визначення його значимості й розраховується за формулою:

$$e(X_i) = - \sum_{k=1}^{N_i} \left(p_k \sum_{l=1}^{N_y} p_{kl} \log_2 p_{kl} \right),$$

де $p_k = \frac{n_k}{m}$ – імовірність влучення значення i -ї ознаки в k -й інтервал діапазону її зміни; n_k – кількість значень i -ї ознаки, що належать k -му інтервалу діапазону її зміни; N_i – кількість інтервалів з діапазону зміни i -ї ознаки, у які вона може потрапити; N_y – кількість інтервалів з діапазону зміни вихідного параметра y ;

$p_{kl} = \frac{n_{kl}}{n_k}$ – умовна ймовірність влучення значення вихідного параметра y в l -й інтервал, за умови, що i -а ознака потрапить в k -й інтервал; n_{kl} – кількість значень вихідного параметра y , що належать l -му інтервалу діапазону його зміни за умови, що значення i -ї ознаки належить k -му інтервалу діапазону її зміни.

У теорії інформації передбачається, що значення ймовірностей станів систем точно відомі. Однак, як правило, ці ймовірності визначаються на основі статистичних даних і являють собою випадкові величини. Тому тільки при нескінченно великому обсязі вибірок їхнього значення можна вважати точними.

9. Критерій, заснований на імовірнісному підході, $Z(X_i)$ заснований на тім, що ознаки можуть бути умовно підрозділені на дві групи. До першої групи відносяться ознаки, значення яких незначно змінюються при переході від одного екземпляра даного класу до іншого об'єкта й досить помітно змінюються при переході від екземпляра одного класу до екземплярів інших класів. До другої групи відносяться ознаки, значення яких чутливі до переходів від одного екземпляра даного класу до іншого екземпляра й лише незначно змінюються при переходах від екземплярів одного класу до екземплярів інших класів. Ознаки, що відносяться до першої групи, є більше інформативними в порівнянні з ознаками, що відносяться до другої групи.

Для оцінювання інформативності ознак використовують $Z(X_i)$:

$$Z(X_i) = \frac{M(d_{il})}{D(m_{il})} = \frac{\sum_{l=1}^{N_y} d_{il} p_l}{\frac{1}{N_y} \sum_{l=1}^{N_y} (m_{il} - M_i)^2} = \frac{\sum_{l=1}^{N_y} \left(\frac{\sum_{j=1}^{\eta_l} (x_{il,j} - m_{il})^2}{\eta_l} \cdot \frac{\eta_l}{m} \right)}{\frac{1}{N_y} \sum_{l=1}^{N_y} (m_{il} - M_i)^2} = \frac{\frac{1}{m} \sum_{l=1}^{N_y} \sum_{j=1}^{\eta_l} (x_{il,j} - m_{il})^2}{\frac{1}{N_y} \sum_{l=1}^{N_y} (m_{il} - M_i)^2},$$

де $M(d_{il})$ – математичне сподівання дисперсії i -ї ознаки по класах. Очевидно, що чим менше величина $M(d_{il})$, тим більше значимою є ознака X_i , оскільки невисокі значення $M(d_{il})$ характеризують більш компактне розташування екземплярів уздовж осі i -ї ознаки; $D(m_{il})$

– дисперсія математичного сподівання розподілу i -ї ознаки при переході від класу до класу. Чим більше дисперсія $D(m_{il})$, тим далі уздовж осі i -ї ознаки розташовуються екземпляри, що відносяться до різних класів, що характеризує високу значимість i -ї ознаки; d_{il} – дисперсія i -ї ознаки за умови, що вихідний параметр у потрапить в l -й інтервал діапазону своєї зміни;

$$m_{il} = m(X_i / y \in [y_l; y_{l+1})) = \frac{1}{n_l} \sum_{j=1}^{n_l} x_{il,j} \quad \text{математичне сподівання } i\text{-ї}$$

ознаки за умови, що вихідний параметр у потрапить в l -й інтервал діапазону своєї зміни; $x_{il,j}$ – j -те значення i -ї ознаки за умови, що вихідний параметр у потрапить в l -й інтервал; n_l – кількість значень i -ї ознаки за умови, що вихідний параметр у потрапить в l -й інтервал, $l = 1, 2, \dots, N_y$; N_y – кількість інтервалів з діапазону зміни вихідного параметра y ; $M_i = \frac{1}{m} \sum_{j=1}^m x_{ij}$ – математичне сподівання i -ї ознаки; m – кількість екземплярів у вибірці.

10. Критерій, заснований на статистичному підході, $S(X_i)$ використовується при наявності досить великого обсягу статистичних даних. У випадку нормального розподілу значень ознаки X_i у кожному класі критерій інформативності $S(X_i)$ може бути обчислений з виразу:

$$S(X_i) = \frac{1}{2} \left(\sigma_{j,1}^2 - \sigma_{j,2}^2 \right) \cdot \left(\frac{1}{\sigma_{j,1}^2} - \frac{1}{\sigma_{j,2}^2} \right) + \frac{1}{2} \left(\frac{1}{\sigma_{j,1}^2} + \frac{1}{\sigma_{j,2}^2} \right) \cdot (\bar{x}_{j,1} - \bar{x}_{j,2})^2,$$

де $\sigma_{j,1}$ і $\sigma_{j,2}$ – середньоквадратичні відхилення j -ї ознаки, за умови, що екземпляр ставиться до першого й другого класів, відповідно; $\bar{x}_{j,1}$ і $\bar{x}_{j,2}$ – середні значення j -ї ознаки, за умови, що екземпляр відноситься до першого й другого класів, відповідно.

Недоліками статистичного критерію є необхідність нормального розподілу в кожному класі, а також неможливість оцінювання інформативності при рішенні завдання класифікації для випадку декількох класів.

Зauważення. Коєфіцієнт парної кореляції застосовується у випадках, коли значення досліджуваної ознаки й вихідного параметра є безперервними.

Якщо ознака й вихідний параметр приймають дискретні значення, тоді для аналізу індивідуальної інформативності застосовують інші критерії. Такі критерії можуть бути використані також для аналізу інформативності безперервних ознак, однак значення таких ознак необхідно попередньо дискретизувати. Для дискретизації весь діапазон зміни ознаки X_i розбивається на N_i однакових інтервалів, де N_i залежить від кількості екземплярів у вибірці даних i , як правило, визначається за формулою: $N_i = \text{Ціле}(\log_2(m)) + 1$, де m – кількість екземплярів у вибірці.

Для відбору інформативних ознак при використанні критерію, заснованого на імовірнісному підході, або ентропії ознаки виконують пошук мінімуму, у випадку використання інших критеріїв виконують їхню максимізацію.

Критерій оцінювання групової інформативності ознак залежно від використовуваної стратегії оцінювання набору ознак застосовують:

- критерій, використовувані у фільтруючих методах;
- помилки синтезованих за допомогою оцінюваної комбінації ознак моделей.

Критерій фільтруючих методів.

1. *Множинний коефіцієнт кореляції* $R(Xe)$ узагальнює поняття парної кореляції на багатомірний випадок і служить для виміру ступеня залежності між однією випадковою величиною Y і множиною випадкових величин Xe . Цей показник множинного зв'язку є аналогом абсолютної величини парного коефіцієнта кореляції й вимірює якість найкращого лінійного наближення:

$$r(Xe) = \frac{\sum_{p=1}^m (y_{\text{оп}, p} - \bar{y})^2}{\sqrt{\sum_{p=1}^m (y_p - \bar{y})^2}},$$

де $y_{\text{оп}, p}$ – p -й елемент вектора $Y_{\text{оп}} = A(A^T A)^{-1} A^T Y$ оцінок значень вихідного параметра Y , обчисленого при припущенні, що залежність між Y і Xe є лінійною; A – матриця, перший стовпець якої містить одиничні значення, а інші значення визначаються в такий спосіб: $A(b, c) = Xe(b, c - 1)$.

2. Коефіцієнт кореляції Пірсона $r_{\text{Пп}}(Xe)$ може бути розрахований за формулою:

$$r_{\text{Пп}}(Xe) = \frac{k \bar{r}_y}{\sqrt{k + k(k-1)r_x}},$$

де $k = |Xe|$ – кількість ознак у розглянутому наборі Xe ; \bar{r}_y – середнє значення коефіцієнта кореляції між кожною ознакою й вихідним параметром; \bar{r}_x – середнє значення коефіцієнта кореляції між ознаками.

3. Множинне дисперсійне відношення $\eta(Xe)$ ураховує зміну величини умовного математичного сподівання $M(Y/Xe)$:

$$\eta(Xe) = \sqrt{\frac{DM(Y/Xe)}{DY}} = \sqrt{\frac{\frac{1}{m} \sum_{k=1}^{N_{Xe}} n_k (\bar{y}_k - \bar{y})^2}{\frac{1}{m} \sum_{p=1}^m (\bar{y}_p - \bar{y})^2}} = \sqrt{\frac{\sum_{k=1}^{N_{Xe}} n_k (\bar{y}_k - \bar{y})^2}{\sum_{p=1}^m (\bar{y}_p - \bar{y})^2}},$$

де $DM(Y/Xe)$ – дисперсія умовного математичного сподівання $M(Y/Xe)$ – характеризує ту частину коливань значень вихідної змінної Y , що

викликана впливом набору вхідних змінних Xe ; $N_{Xe} = \sum_{l=1}^{|Xe|} N_l$ –

кількість комбінацій інтервалів діапазону зміни значень набору ознак Xe . Ця величина визначається як добуток кількостей інтервалів розбивки для кожної ознаки, що входить у набір Xe ; n_k – кількість значень набору ознак Xe , що потрапили в k -й інтервал групування;

$\bar{y}_k = \frac{1}{n_k} \sum_{j=1}^{n_k} y_{k,j}$ – середнє значення вихідного параметра для k -го інтервалу діапазону значень набору ознак Xe ; $y_{k,j}$ – j -те значення вихідного параметра Y за умови, що значення набору ознак Xe потраплять в k -й інтервал; $\bar{y} = \frac{1}{m} \sum_{p=1}^m \bar{y}_p$ – середнє значення вихідної змінної Y ;

DY – дисперсія вихідного параметра Y .

4. Множинний коефіцієнт зв'язку $\xi(Xe)$ для оцінювання інформативності набору ознак Xe стосовно вихідного параметра Y за допомогою оцінки зміни умовного математичного сподівання $M(Y/Xe)$ і умовної дисперсії $D(Y/Xe)$:

$$\xi(Xe) = \sqrt{\frac{D(M(Y/Xe))^2}{DY^2 - 2B}},$$

$$B = \text{cov}((M(Y/Xe))^2, (D(Y/Xe))) =$$

$$= \frac{1}{m} \sum_{k=1}^{N_e} n_k \left((\bar{y}_k)^2 - \bar{y}^2 \right) \left(\bar{y}^2_k - (\bar{y}_k)^2 - \frac{1}{m} \sum_{a=1}^{N_e} n_a \left(\bar{y}_a^2 - (\bar{y}_a)^2 \right) \right),$$

де $D(M(Y/Xe))^2 = \frac{1}{m} \sum_{k=1}^{N_e} n_k \left((\bar{y}_k)^2 - \bar{y}^2 \right)^2$ – дисперсія квадрата умовного математичного сподівання $M(Y/Xe)$; $\bar{y}^2 = \frac{1}{m} \sum_{p=1}^m y_p^2$ – середнє

значення квадрата вихідної змінної Y ; $DY^2 = \frac{1}{m} \sum_{p=1}^m \left(y_p^2 - \bar{y}^2 \right)^2$ – дисперсія квадрата вихідного параметра Y ; $\bar{y}^2_k = \frac{1}{n_k} \sum_{j=1}^{n_k} y_{k,j}^2$ –

квадрат середнього значення вихідного параметра для k -го інтервалу діапазону зміни i -ї ознаки.

5. Інформаційний критерій $I(Xe)$ розраховується за формулою:

$$I(Xe) = - \sum_{l=1}^{N_y} p_l \log_2 p_l + \sum_{l=1}^{N_y} p_l \sum_{q=1}^{N_y} \sum_{k=1}^{N_{Xe}} p_{lk} p_{kq} \log_2 p_{kq},$$

де N_y – кількість інтервалів з діапазону зміни вихідного параметра y ; $p_l = \frac{n_l}{m}$ – імовірність попадання значення вихідного параметра в l -й інтервал діапазону його зміни; n_l – кількість значень вихідного параметра, що належать l -му інтервалу діапазону його зміни; N_{Xe} – кількість комбінацій інтервалів діапазону зміни набору ознак Xe , у

які можуть потрапити значення ознак набору Xe ; p_{ik} – умовна ймовірність влучення комбінації значень набору ознак Xe в k -ту комбінацію інтервалів діапазону зміни набору ознак за умови, що вихідний параметр y потрапить в l -й інтервал; p_{kq} – умовна ймовірність влучення значення вихідного параметра y в q -й інтервал, за умови, що комбінації значень набору ознак Xe потрапить в k -ту комбінацію інтервалів діапазону зміни набору ознак.

6. Ентропія $e(Xe)$ набору ознак Xe визначається за формулою

$$e(Xe) = - \sum_{k=1}^{N_{Xe}} p_k \left(\sum_{l=1}^{N_y} p_{kl} \log_2 p_{kl} \right),$$

де p_k – імовірність влучення комбінації значень набору ознак Xe в k -ту комбінацію інтервалів діапазону зміни набору ознак; N_{Xe} – кількість комбінацій інтервалів діапазону зміни набору ознак Xe , у які можуть потрапити значення ознак набору Xe ; N_y – кількість інтервалів з діапазону зміни вихідного параметра y ; p_{kl} – умовна ймовірність влучення значення вихідного параметра y в l -й інтервал, за умови, що набір ознак Xe потрапить в k -ту комбінацію інтервалів діапазону зміни набору ознак.

7. Критерій, заснований на статистичному підході, $S(Xe)$ може бути застосований за тих самих умов, що й аналогічний критерій, що використовується для оцінювання індивідуальної інформативності ознак. Інформативність сукупності ознак за допомогою статистичного критерію може бути визначена за формулою:

$$S(Xe) = \frac{1}{2} \text{tr} \left((V_1 - V_2)(V_1^{-1} - V_2^{-1}) \right) + \frac{1}{2} \text{tr} \left((V_1^{-1} + V_2^{-1})(\bar{Xe}_1 - \bar{Xe}_2)(\bar{Xe}_1 - \bar{Xe}_2)^T \right),$$

де $V_1 = \{v_{ij}\}_1$ і $V_2 = \{v_{ij}\}_2$ – коваріаційні матриці для випадків, коли екземпляр відноситься до першого й другого класів, відповідно; $v_{ij} = M((x_i - \bar{x}_i)(x_j - \bar{x}_j))$ – загальний елемент коваріаційної матриці, що прораховується окремо для кожного класу; M – оператор математичного сподівання; V_1^{-1} і V_2^{-1} – зворотні матриці до V_1 та V_2 , відповідно; $\text{tr}(V)$ – слід матриці V , обчислений як сума діагональних елементів; $\bar{Xe} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{|Xe|})$ – вектор середніх значень ознак; A^T – матриця, транспонована до A .

8. Критерій оцінювання набору ознак на основі теорії чітких множин використовується для відбору ознак при рішенні завдання класифікації. Якщо буде потреба застосування такого критерію для виділення інформативної комбінації ознак при прогнозуванні значення ознак і вихідного параметра необхідно квантувати.

Критерій оцінювання набору ознак Xe на основі теорії чітких множин розраховується за формулою:

$$\gamma_Y(Xe) = \frac{|POS_{Xe}(Y)|}{|U|},$$

де U – кінцева множина екземплярів; $|U| = m$ – потужність множини U , рівна кількості екземплярів у вихідній вибірці; $|POS_{Xe}(Y)|$ – величина, що визначає кількість екземплярів, які можуть бути вірно класифіковані за допомогою набору ознак Xe . При цьому в множину $POS_{Xe}(Y)$ включаються номери вірно класифікованих екземплярів за допомогою набору ознак Xe . Таким чином, у множину $POS_{Xe}(Y)$ не включаються номери тих екземплярів, які мають однакові значення ознак з набору Xe , але різні значення вихідного параметра.

Математично множина $POS_{Xe}(Y)$ може бути визначене в такий спосіб:

$$POS_{Xe}(Y) = \bigcup_{A \in U/Y} Xe(A),$$

де $U/Y = IND(Y)$ – множина, кожний l -й елемент якої містить номенклатуру екземплярів, для яких значення класу вихідного параметра Y дорівнює l ; $Xe(A) = \{x | [a]_{Xe} \subseteq A\}$ – нижнє наближення множини A до множини Xe ; $[a]_{Xe}$ – еквівалентні класи Xe -нерозрізленого зв'язку, тобто такі, які визначаються операцією $IND(Xe)$: $IND(Xe) = U/Xe = \{(x, y) \in U^2 | \forall a \in Xe \ a(x) = a(y)\}$.

Якщо $\gamma_Y(Xe) = 1$, тоді вихідний параметр Y повністю прогнозується (класифікується) за допомогою набору ознак Xe .

9. Оцінка групової інформативності на основі теорії нечітких множин для набору ознак Xe може бути розрахована у такий спосіб:

$$\beta_Y(Xe) = \frac{\sum_{l=1}^m \mu_{POS_{Xe}(Y)}(l)}{|U|},$$

$$\mu_{POS_{Xe}(Y)}(l) = \max_{A=y_1, y_2, \dots, y_m} \mu_{\underline{XeA}}(l),$$

$$\mu_{\underline{XeA}}(l) = \max_{b=1, 2, \dots, B_{Xe}} \left[\min \left(\mu_{Xe_b}(l); \min_{a=1, 2, \dots, m} \max(l - \mu_{Xe_b}(a); \mu_A(a)) \right) \right],$$

$$B_{Xe} = \prod_{X_i \in Xe} |X_i|,$$

де U – кінцева множина екземплярів; $|U| = m$ – потужність множини U , рівна кількості екземплярів у вихідній вибірці; l – номер екземпляра; $\mu_{POS_{Xe}(Y)}(l)$ – функція належності нечіткої множини $POS_{Xe}(Y)$; B_{Xe} – кількість різних комбінацій значень, які можуть приймати ознаки з множини Xe ; Xe_b – b -та комбінація значень нечітких термів множини Xe .

Критерій оцінювання групової інформативності ознак на основі помилок синтезованих моделей поділяють на:

- критерій оцінювання спільного впливу набору ознак у задачах прогнозування;
- критерій оцінювання спільного впливу комбінації ознак при класифікації.

Критерій оцінювання спільного впливу набору ознак у за- вданнях прогнозування.

1. Середньоеквадратична помилка MSE (mean squared error), що розраховується за формулою:

$$MSE(Xe) = \frac{1}{m} \sum_{p=1}^m (y^p - y_{Xe}^p)^2,$$

де y^p_{Xe} – значення вихідного параметра p -го екземпляра, розраховане по синтезованій моделі; y^p – реальне значення вихідного параметра p -го екземпляра; m – кількість екземплярів у вихідній вибірці даних.

2. Сума квадратів відхилень SSE (sum squared error):

$$SSE(Xe) = m \cdot MSE(Xe) = \sum_{p=1}^m (y^p - y_{Xe}^p)^2.$$

3. Середня абсолютна помилка MAE (mean absolute error) – характеризує абсолютні відхилення реальних значень вихідного параметра від значень, отриманих за допомогою синтезованої моделі:

$$MAE(Xe) = \frac{1}{m} \sum_{p=1}^m |y^p - y_{Xe}^p|.$$

4. Сума значень абсолютних відхилень SAE (sum absolute error):

$$SAE(Xe) = m \cdot MAE(Xe) = \sum_{p=1}^m |y^p - y_{Xe}^p|.$$

5. Максимальне абсолютне відхилення $MaxAE$ (maximum absolute error):

$$MaxAE(Xe) = \max_{p=1,2,\dots,m} |y^p - y_{Xe}^p|.$$

6. Середня відносна помилка $MARE$ (mean absolute relative error) або $MAPE$ (mean absolute percentage error) – оцінює відносні відхилення реальних значень вихідного параметра від значень, отриманих за допомогою синтезованої моделі:

$$MARE(Xe) = \frac{1}{m} \sum_{p=1}^m \left| \frac{y^p - y_{Xe}^p}{y^p} \right|, \quad y^p \neq 0.$$

7. Сума відносних відхилень SRE (sum relative error):

$$SRE(Xe) = m \cdot MARE(Xe) = \sum_{p=1}^m \left| \frac{y^p - y_{Xe}^p}{y^p} \right|, \quad y^p \neq 0.$$

8. Максимальне відносне відхилення $MaxRE$ (maximum relative error):

$$MaxRE(Xe) = \max_{p=1,2,\dots,m} \left| \frac{y^p - y_{Xe}^p}{y^p} \right|, \quad y^p \neq 0.$$

Критерій оцінювання групової інформативності ознак при класифікації.

1. Ймовірність прийняття помилкових рішень E за моделлю, що відповідає оцінюваній комбінації ознак:

$$E(Xe) = \frac{n}{m},$$

де n – кількість екземплярів вибірки, невірно класифікованих за допомогою побудованої моделі.

2. Критерій Фішера, що розраховується за формулами:

$$F(Xe) = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$$

або

$$F(Xe) = \frac{|\mu_1 - \mu_2|}{\sigma_1^2 + \sigma_2^2},$$

де μ_1 , μ_2 і σ_1^2 , σ_2^2 – математичні сподівання й дисперсії належності вихідного параметра u до першого й другого класу, відповідно, обчислені за допомогою моделі, синтезованої на основі набору ознак Xe .

Порівняльна характеристика критеріїв визначення інформативності ознак наведена у табл. 3.2. Тут позначено: «д» – дискретні значення, «б» – безперервні, «бін» – бінарні, «неч» – нечіткі.

3.5 Метрики, використовувані при кластеризації ознак

При кластеризації ознак застосовуються метрики відстані, що дозволяють оцінити не інформативність ознаки або набору ознак, а визначити міру близькості їхнього розташування по відношенню друг до друга в просторі екземплярів.

До метрик, використовуваних при кластеризації й дозволяючих оцінити відстань між ознаками x_a і x_b , що характеризуються m екземплярами, відносяться:

1) Евклідову відстань:

$$d = \left(\sum_{j=1}^m (x_{aj} - x_{bj})^2 \right)^{\frac{1}{2}};$$

2) відстань Хеммінга (відстань Манхеттена, метрика міських кварталів):

$$d = \sum_{j=1}^m |x_{aj} - x_{bj}|;$$

3) *відстань Мінковського* (узагальнену відстань):

$$d = \left(\sum_{j=1}^m |x_{aj} - x_{bj}|^v \right)^{\frac{1}{v}},$$

де v – деяке ціле число, що задається дослідником;

4) *відстань у неізотропному просторі* ознак:

$$d = \sum_{j=1}^m (\alpha_j)^2 (x_{aj} - x_{bj})^2,$$

де α_j – коефіцієнт значимості j -го екземпляра;

5) *діагностичну міру відстані*:

$$d = \left(\sum_{j=1}^m |x_{aj} - x_{bj}|^v \right)^{\frac{\mu}{v}},$$

де v, μ – деякі цілі числа, що задаються дослідником;

6) *узагальнену відстань у просторі* ознак:

$$d_k = \left(\sum_{j=1}^m \alpha_{jk}^v |x_{aj} - x_{bj}|^v \right)^{\frac{\mu}{v}},$$

де α_{jk} – коефіцієнт значимості j -го екземпляра для k -го класу;

7) *відстань у нелінійному просторі*:

$$d = \sum_{j=1}^m (\alpha_j)^p \left| (x_{aj})^p - (x_{bj})^p \right|,$$

де p – ступінь нелінійності, призначена зменшити помилку класифікації (як правило, $p = 2, 3$);

Таблиця 3.2 – Порівняльна характеристика критеріїв оцінювання інформативності ознак

Критерій інформативності	Тип значень ознак	Тип значень вихідного параметра	Необхідність побудови моделі для оцінювання інформативності	Можливість опанування наборів, що складаються з декількох ознак	Горизонтальне обтуртування	Відносна складність реалізації	Час обчислень
Коефіцієнт парної кореляції	б, л	б, д	немас	немас	високе	нізька	нільзький
Коефіцієнт кореляції знаків	б, л	б, д	немас	немас	середнє	нізька	нільзький
Коефіцієнт кореляції Фехтера	б, л	б, д	немас	немас	середнє	нізька	нільзький
Дисперсійне відхилення	л	б, д	немас	так	високе	середній	нільзький
Коефіцієнт зв'язку	л	б, д	немас	так	високе	середній	нільзький
Інформаційний критерій	л	д	немас	так	високе	висока	середній
Теоретико-інформаційний критерій	л	д	немас	так	високе	висока	середній
Ентропія	л	д	немас	так	високе	висока	середній
Критерій, заснований на імовірностному підході	б, л	д	немас	немас	середнє	середній	нільзький
Критерій, заснований на статистичному підході	б, л	бін	немас	так	середнє	нізька	нільзький
Множинний коефіцієнт кореляції	б, л	б, д	немас	так	високе	нізька	нільзький
Коефіцієнт кореляції Греона	б, л	б, д	немас	так	середнє	нізька	нільзький
Критерій, заснований на теорії чітких множин	неч	неч	немас	так	середнє	середній	нільзький
Критерій, заснований на теорії нечітких множин	б, л	б, д	так	так	високе	висока	середній
Середньоквадратична помилка	б, л	б, д	так	так	високе	висока	високий
Сума квадратів відхилень	б, л	б, д	так	так	високе	висока	високий
Середня абсолютнона помилка	б, л	б, д	так	так	високе	висока	високий
Сума значень абсолютноних відхилень	б, л	б, д	так	так	середнє	висока	високий
Максимальне абсолютное відхилення	б, л	б, д	так	так	високе	висока	високий
Середня відносна помилка	б, л	б, д	так	так	високе	висока	високий
Сума відносних відхилень	б, л	б, д	так	так	високе	висока	високий
Максимальне відносне відхилення	б, л	б, д	так	так	середнє	висока	високий
Імовірність прийняття істотикових рішень	б, л	д	так	так	високе	висока	високий
Критерій Фішера	б, л	бін	так	так	високе	висока	високий

8) узагальнену (зважену) відстань Махалонобіса:

$$d = \sqrt{(\bar{x}_a - \bar{x}_b)^T \Lambda^T \Sigma^{-1} \Lambda (\bar{x}_a - \bar{x}_b)},$$

де Σ – коваріаційна матриця генеральної сукупності ознак; Λ – деяка симетрична ненегативно визначена матриця вагових коефіцієнтів, що найчастіше вибирається діагональною;

9) відстань Камберра: $d = \sum_{j=1}^m \frac{|x_{aj} - x_{bj}|}{|x_{aj} + x_{bj}|};$

10) відстань Чебишєва: $d = \max_j |x_{aj} - x_{bj}|;$

11) кореляційну відстань: $d = \left(\sum_{j=1}^m x_{aj} x_{bj} \right) - \frac{1}{m} \left(\sum_{j=1}^m x_{aj} \right) \left(\sum_{j=1}^m x_{bj} \right);$

12) кутова відстань: $d = \cos \gamma = \frac{x_a x_b}{|x_a| |x_b|} = \frac{\sum_{j=1}^m x_{aj} x_{bj}}{\sqrt{\left(\sum_{j=1}^m x_{aj}^2 \right) \left(\sum_{j=1}^m x_{bj}^2 \right)}}.$

Вибір певної метрики при класифікації ознак спирається на апріорні відомості про можливі особливості розподілу значень ознак в обраній сукупності екземплярів, а також залежить від результатів, отриманих раніше, і обмежень на обчислювальні витрати при проведенні експериментів.

3.6 Методи формування штучних ознак

У випадку, якщо вихідний набір ознак $x = \{x_j\}, j = 1, 2, \dots, N$, має великий обсяг N , а самі ознаки x_j індивідуально є малоінформативними, виникає задача конструювання або витягу (feature extraction) з набору первинних ознак x N' більш інформативних штучних ознак $x' = \{x'_i\}, i = 1, 2, \dots, N'$: $x' = g(x)$, де g – деяке функціональне перетворення, при якому $N' < N$ і $Q_n(x') \geq Q_n(x)$, де Q_n – функціонал якості набору ознак вибірки.

Методи витягу (конструювання) ознак виділяють:

– аналіз головних компонентів (Principal Component Analysis, PCA) – використовує ортогональне перетворення набору можливо

корельованих змінних у множину некорельованих змінних, назива-
них головними компонентами, число яких менше або дорівнює числу
вихідних змінних, ранжируваних за убуванням дисперсії (недолік –
чутливість до відносного масштабування вихідних змінних);

– *напізвисначене вкладення* (Semidefinite Embedding, SDE) – ви-
користовує напізвисначене програмування для нелінійного скорочен-
ня розмірності даних, прагнучи відобразити багатомірні дані в Евклі-
довий простір малої розмірності, використовуючи локальну ліній-
ність різноманітті і створюючи відображення, що зберігає локальні
околиці в кожній точці;

– *багатофакторне скорочення розмірності* (Multifactor Dimensionality Reduction, MDR) – метод виявлення й опису комбінації
дискретних вихідних змінних, що спільно впливають на бінарну вихід-
ну змінну, заснований на конструктивному алгоритмі індукції, що
змінює простір представлення даних для виявлення такого представ-
лення, що полегшує виявлення нелінійних взаємодій між ознаками;

– *нелінійне скорочення розмірності* (Non-linear Dimensionality Reduction, NLDR) – група методів, заснованих на припущення про те,
що дані, які становлять інтерес, лежать на вкладеному нелінійному
різноманітті в просторі більш високої розмірності. Якщо різноманіт-
тя має досить низьку розмірність, то дані можуть бути візуалізовані у
просторі меншої розмірності. Найбільше широко застосовуваним
методом є *Isomap* – метод, у якому об’єднані геодезичні відстані на
зваженому графі з метричним багатовимірним шкачуванням, вико-
ристовується для обчислення квазізометричного, низькорозмірного
вкладення множини точок даних високої розмірності;

– *часткові найменні квадрати* (Partial Least Squares, PLS) –
метод, що шукає модель лінійної регресії, проектуючи вхідні і
вихідні змінні в новий простір;

– *аналіз незалежних компонентів* (Independent Component Analysis, ICA) – метод поділу багатомірного сигналу на аддитивні
компоненти, виходячи з припущення взаємної статистичної неза-
лежності сигналів;

– *теорія редукції* ставить метою максимальне скорочення
простору ознак, здійснюване шляхом ітеративного конструюван-
ня штучних ознак на основі первинних ознак з найкращими поді-
ляючими властивостями;

— метод генеральної узагальненої змінної перетворює багатомірний простір ознак в одну змінну, одержувану у виді мультиплікативної чи аддитивної функції усього вихідного набору ознак.

Загальними недолікам усіх розглянутих груп методів конструктування ознак є відсутність у загальному випадку гарантії поліпшення роздільності класів, втрата частини інформації, що містилася у вихідній вибірці, зміна топологічних і статистичних властивостей вибірки, практично повна утрата фізичного змісту вихідних змінних і складність інтерпретації результатів проектування навчальної множини з N -вимірного у M -вимірний простір і, як наслідок, неінтерпретабельність синтезованих моделей, одержуваних на основі сформованих штучних ознак. Також дані методи є обчислювально витратними і припускають участь користувача для оцінки якості побудованого перетворення, або завдання керованих параметрів методів, що обмежує рівень автоматизації процесу синтезу моделі.

У ряді практичних застосувань для витягу ознак також можуть застосовуватися різні методи цифрової обробки сигналів, наприклад, перетворення Фур'є- і вейвлет-перетворення. Однак можливість їхнього застосування визначається специфічними особливостями розв'язуваної задачі.

3.7 Приклади виконання завдань

 *Приклад 1.* Для заданої вибірки (табл. 3.3), ознаки й вихідний параметр якої приймають безперервні значення, обчислити значення:

Таблиця 3.3 – Вибірка з безперервними значеннями ознак і вихідного параметра

№ екземпляра	X_1	X_2	X_3	X_4	X_5	X_6	Y
1	1,2	7,2	8,6	5,6	2,3	6,4	2,5
2	2,3	3,4	5,4	8,9	4,5	4,2	4,7
3	3,7	5,6	3,2	11,5	7,9	3,8	8,1
4	4,5	6,9	1,1	12,6	9,3	3,5	9,1
5	5,9	8,1	0,8	20,4	10,8	2,7	11,6

а) коефіцієнта парної кореляції для ознак X_1 , X_2 і X_4 : $r_{\text{п}}(X_1)$, $r_{\text{п}}(X_2)$ і $r_{\text{п}}(X_4)$;

б) множинного коефіцієнта кореляції для набору ознак $X_e = \{X_1, X_2, X_4\}$: $r(X_e)$.

в) коефіцієнта кореляції Пірсона для набору ознак $X_e = \{X_1, X_2, X_4\}$: $r_{\text{Пир}}(X_e)$.

г) метрики, використовувані при класифікації ознак: Евклідова відстань, відстань Хеммінга, відстань Мінковського (при $v = 3$), відстань у неізотропному просторі ознак (при $\alpha_1 = 0,3$, $\alpha_2 = 0,1$, $\alpha_3 = 0,4$, $\alpha_4 = 0,05$, $\alpha_5 = 0,15$), діагностичну міру відстані (при $v = 3$, $\mu = 2$), відстань у нелінійному просторі (при $p = 2$ і $\alpha_1 = 0,3$, $\alpha_2 = 0,1$, $\alpha_3 = 0,4$, $\alpha_4 = 0,05$, $\alpha_5 = 0,15$), відстань Камберра, відстань Чебишева, кореляційна відстань, кутова відстань між ознаками X_3 і X_5 .

Розв'язання:

а) для розрахунку значень коефіцієнтів парної кореляції ознак X_1 , X_2 і X_4 і вихідний параметри обчислимо попередньо середні значення \bar{x}_1 , \bar{x}_2 , \bar{x}_4 , \bar{y} :

$$\bar{x}_1 = \frac{1}{5}(1,2 + 2,3 + 3,7 + 4,5 + 5,9) = 3,52,$$

$$\bar{x}_2 = \frac{1}{5}(7,2 + 3,4 + 5,6 + 6,9 + 8,1) = 6,24,$$

$$\bar{x}_4 = \frac{1}{5}(5,6 + 8,9 + 11,5 + 12,6 + 20,4) = 11,8,$$

$$\bar{y} = \frac{1}{5}(2,5 + 4,7 + 8,1 + 9,1 + 11,6) = 7,2.$$

Розрахуємо значення коефіцієнтів парної кореляції ознак:

$$r_{\text{п}}(X_1) = \frac{(1,2 - 3,52)(2,5 - 7,2) + (2,3 - 3,52)(4,7 - 7,2) + \dots + (5,9 - 3,52)(11,6 - 7,2)}{\sqrt{[(1,2 - 3,52)^2 + (2,3 - 3,52)^2 + \dots + (5,9 - 3,52)^2][(2,5 - 7,2)^2 + (4,7 - 7,2)^2 + \dots + (11,6 - 7,2)^2]}} = \\ = \frac{26,45}{\sqrt{13,528 \cdot 52,12}} = 0,9961.$$

Аналогічно $r_{\text{н}}(X_2) = 0,435$ і $r_{\text{н}}(X_4) = 0,9484$. Оскільки $|r_{\text{н}}(X_1)| > |r_{\text{н}}(X_4)| > |r_{\text{н}}(X_2)|$, то можна стверджувати, що X_1 впливає на Y . При цьому перевіряється наявність лінійної залежності між X_1 і Y .

б) розрахуємо значення множинного коефіцієнта кореляції для набору ознак $Xe = \{X_1, X_2, X_4\}$: $r(Xe)$. Для цього визначимо вектор $Y_{\text{он}} = A(A^T A)^{-1} A^T Y$:

$$Y_{\text{он}} = \begin{pmatrix} 1 & 1,2 & 7,2 & 5,6 \\ 1 & 2,3 & 3,4 & 8,9 \\ 1 & 3,7 & 5,6 & 11,5 \\ 1 & 4,5 & 6,9 & 12,6 \\ 1 & 5,9 & 8,1 & 20,4 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1,2 & 2,3 & 3,7 & 4,5 & 5,9 \\ 7,2 & 3,4 & 5,6 & 6,9 & 8,1 \\ 5,6 & 8,9 & 11,5 & 12,6 & 20,4 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1,2 & 7,2 & 5,6 \\ 1 & 2,3 & 3,4 & 8,9 \\ 1 & 3,7 & 5,6 & 11,5 \\ 1 & 4,5 & 6,9 & 12,6 \\ 1 & 5,9 & 8,1 & 20,4 \end{pmatrix}^{-1} \times \\ \times \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1,2 & 2,3 & 3,7 & 4,5 & 5,9 \\ 7,2 & 3,4 & 5,6 & 6,9 & 8,1 \\ 5,6 & 8,9 & 11,5 & 12,6 & 20,4 \end{pmatrix} \cdot \begin{pmatrix} 2,5 \\ 4,7 \\ 8,1 \\ 9,1 \\ 11,6 \end{pmatrix} = \begin{pmatrix} 2,5015 \\ 4,8619 \\ 7,6822 \\ 9,3352 \\ 11,6192 \end{pmatrix}.$$

Обчислимо значення множинного коефіцієнта кореляції:

$$r(Xe) = \frac{(2,5015 - 7,2)^2 + (4,8619 - 7,2)^2 + (7,6822 - 7,2)^2 + (9,3352 - 7,2)^2 + (11,6192 - 7,2)^2}{(2,5 - 7,2)^2 + (4,7 - 7,2)^2 + (8,1 - 7,2)^2 + (9,1 - 7,2)^2 + (11,6 - 7,2)^2} = \\ = \frac{51,86}{52,12} = 0,99;$$

в) для розрахунку значення коефіцієнта кореляції Пірсона для набору ознак $Xe = \{X_1, X_2, X_4\}$ спочатку визначимо значення k , \bar{r}_x і \bar{r}_y .

Оскільки набір Xe складається із трьох ознак, то $k = 3$.

\bar{r}_y – середнє значення коефіцієнта кореляції між кожною ознакою оцінюваного набору й вихідним параметром. Використовуючи отримані раніше дані, одержуємо:

$$\bar{r}_y = \frac{r_{\text{н}}(X_1) + r_{\text{н}}(X_2) + r_{\text{н}}(X_4)}{3} = \frac{0,9961 + 0,435 + 0,9484}{3} = 0,7932.$$

З метою розрахунку середнього значення коефіцієнта кореляції між ознаками \bar{r}_x розрахуємо коефіцієнти парної кореляції між всіма ознаками в оцінюваному наборі Xe : $r_n(X_1; X_2)$, $r_n(X_1; X_4)$ і $r_n(X_2; X_4)$, де $r_n(X_i; X_j)$ – коефіцієнт кореляції між i -ю і j -ю ознаками в наборі Xe : $r_n(X_1; X_2) = 0,4621$, $r_n(X_1; X_4) = 0,9648$, $r_n(X_2; X_4) = 0,4727$.

Таким чином:

$$\bar{r}_x = \frac{r_n(X_1; X_2) + r_n(X_1; X_4) + r_n(X_2; X_4)}{3} = \frac{0,4621 + 0,9648 + 0,4727}{3} = 0,6332.$$

Підставляючи знайдені значення k , \bar{r}_x і \bar{r}_y у формулу для розрахунку коефіцієнта кореляції Пірсона, одержуємо:

$$r_{\text{Пир}}(Xe) = \frac{k \bar{r}_y}{\sqrt{k + k(k-1)\bar{r}_x^2}} = \frac{3 \cdot 0,7932}{\sqrt{3 + 3(3-1) \cdot 0,6332}} = 0,9126;$$

г) розрахуємо різні відстані між ознаками X_3 і X_5 :

– Евклідову відстань:

$$d = \sqrt{(8,6 - 2,3)^2 + (5,4 - 4,5)^2 + (3,2 - 7,9)^2 + (1,1 - 9,3)^2 + (0,8 - 10,8)^2} = 15,16;$$

– відстань Хеммінга:

$$d = |8,6 - 2,3| + |5,4 - 4,5| + |3,2 - 7,9| + |1,1 - 9,3| + |0,8 - 10,8| = 30,1;$$

– відстань Мінковського (при $v = 3$):

$$d = \left(|8,6 - 2,3|^3 + |5,4 - 4,5|^3 + |3,2 - 7,9|^3 + |1,1 - 9,3|^3 + |0,8 - 10,8|^3 \right)^{\frac{1}{3}} = 12,4;$$

– відстань у неізотропному просторі ознак (при $\alpha_1 = 0,3$, $\alpha_2 = 0,1$, $\alpha_3 = 0,4$, $\alpha_4 = 0,05$, $\alpha_5 = 0,15$):

$$d = 0,3^2(8,6 - 2,3)^2 + 0,1^2(5,4 - 4,5)^2 + 0,4^2(3,2 - 7,9)^2 + 0,05^2(1,1 - 9,3)^2 + 0,15^2(0,8 - 10,8)^2 = 39,19;$$

– діагностична міра відстані (при $v = 3$, $\mu = 2$):

$$d = \left(|8,6 - 2,3|^3 + |5,4 - 4,5|^3 + |3,2 - 7,9|^3 + |1,1 - 9,3|^3 + |0,8 - 10,8|^3 \right)^{\frac{2}{3}} = 153,72;$$

– відстань у нелінійному просторі (при $p = 2$ і $\alpha_1 = 0,3$, $\alpha_2 = 0,1$, $\alpha_3 = 0,4$, $\alpha_4 = 0,05$, $\alpha_5 = 0,15$):

$$d = 0,3^2 |8,6^2 - 2,3^2| + 0,1^2 |5,4^2 - 4,5^2| + 0,4^2 |3,2^2 - 7,9^2| + 0,05^2 |1,1^2 - 9,3^2| + 0,15^2 |0,8^2 - 10,8^2| = 17,44;$$

– відстань Камберра:

$$d = \frac{|8,6 - 2,3| + |5,4 - 4,5| + |3,2 - 7,9| + |1,1 - 9,3| + |0,8 - 10,8|}{|8,6 + 2,3| + |5,4 + 4,5| + |3,2 + 7,9| + |1,1 + 9,3| + |0,8 + 10,8|} = 0,558;$$

– відстань Чебишева:

$$d = \max(|8,6 - 2,3|; |5,4 - 4,5|; |3,2 - 7,9|; |1,1 - 9,3|; |0,8 - 10,8|) = 10;$$

– кореляційна відстань:

$$d = 8,6 \cdot 2,3 + 5,4 \cdot 4,5 + 3,2 \cdot 7,9 + 1,1 \cdot 9,3 + 0,8 \cdot 10,8 - \frac{1}{5} (8,6 + 5,4 + 3,2 + 1,1 + 0,8) (2,3 + 4,5 + 7,9 + 9,3 + 10,8) = -44,71;$$

– кутова відстань:

$$d = \frac{8,6 \cdot 2,3 + 5,4 \cdot 4,5 + 3,2 \cdot 7,9 + 1,1 \cdot 9,3 + 0,8 \cdot 10,8}{\sqrt{8,6^2 + 5,4^2 + 3,2^2 + 1,1^2 + 0,8^2} \sqrt{2,3^2 + 4,5^2 + 7,9^2 + 9,3^2 + 10,8^2}} = 0,0026.$$

 *Приклад 2.* Для заданої вибірки (табл. 3.4), ознаки й вихідний параметр якої приймають дискретні значення, обчислити значення:

Таблиця 3.4 – Вибірка з дискретними значеннями ознак і вихідного параметра

№ екземпляра	X_1	X_2	X_3	X_4	X_5	X_6	Y
1	2	0	2,2	2	1	3	0
2	1	1	4,7	1	0	5	2
3	1	0	4	0	2	5	1
4	2	1	5,6	0	1	3	2
5	0	0	3,3	2	1	2	1
6	1	2	4,1	0	2	5	1
7	1	1	6,2	1	2	5	2
8	0	1	4,3	1	0	2	1

- а) коефіцієнта кореляції знаків для ознак X_2 і X_4 : $r_s(X_2)$ і $r_s(X_4)$;
 б) коефіцієнта кореляції Фехнера для ознак X_2 і X_4 : $r_{\Phi}(X_2)$ і $r_{\Phi}(X_4)$;

- в) дисперсійного відношення для ознак X_1 і X_3 : $\eta(X_1)$ і $\eta(X_3)$;
- г) коефіцієнта зв'язку для ознаки X_3 : $\xi(X_3)$;
- д) інформаційного критерію для ознаки X_4 : $I(X_4)$;
- е) ентропії ознак X_2 і X_4 : $e(X_2)$ і $e(X_4)$;
- ж) ентропії набору ознак $Xe = \{X_2, X_4\}$: $e(Xe)$;
- з) критерію, заснованого на імовірнісному підході, для ознак X_5 і X_6 : $Z(X_5)$ і $Z(X_6)$;

и) критерію, заснованого на статистичному підході, для ознаки X_5 : $S(X_5)$, вважаючи, що значення вихідного параметра для першого екземпляра дорівнює одиниці, а також ознаки по класах $(1, 2, 1, 2, 0)$ і $(0, 1, 2)$ розподілені нормальню;

к) критерію оцінювання набору ознак $Xe = \{X_2, X_4\}$ на основі теорії чітких множин: $\gamma(Xe)$;

Розв'язання:

а) для розрахунку значень коефіцієнтів кореляції знаків для ознак X_2 і X_4 визначимо попередньо значення \bar{x}_2 , \bar{x}_4 , \bar{y} , $C_{x_2,y}^+$, $C_{x_4}^+$ і C_y^+ :

$$\bar{x}_2 = 0,75, \quad \bar{x}_4 = 0,875, \quad \bar{y} = 1,25, \quad C_{x_2,y}^+ = 3/8 = 0,375, \quad C_{x_4}^+ = 5/8 = 0,625,$$

$$C_y^+ = 3/8 = 0,375, \quad C_{x_4,y}^+ = 2/8 = 0,25, \quad C_{x_4}^+ = 5/8 = 0,625.$$

Таким чином:

$$r_3(X_2) = \frac{0,375 - 0,625 \cdot 0,375}{\sqrt{0,625 \cdot 0,375(1 - 0,625)(1 - 0,375)}} = 0,6,$$

$$r_3(X_4) = \frac{0,25 - 0,625 \cdot 0,375}{\sqrt{0,625 \cdot 0,375(1 - 0,625)(1 - 0,375)}} = 0,0667;$$

б) з метою розрахунку коефіцієнта кореляції Фехнера необхідно визначити C_i – кількість збігів однакових, як позитивних, так і негативних знаків різниць $(x_{ip} - \bar{x}_i)$ і $(y_p - \bar{y})$ для i -ї ознаки: $C_2 = 6$, $C_4 = 4$. Тому:

$$r_\Phi(X_2) = \frac{2 \cdot 6}{8} - 1 = 0,5,$$

$$r_\Phi(X_4) = \frac{2 \cdot 4}{8} - 1 = 0;$$

в) для обчислення значення дисперсійного відношення $\eta(X_1)$ ознаки X_1 урахуємо, що він 2 рази приймає значення, рівне нулю, 4 рази дорівнює одиниці й 2 рази дорівнює двом.

При $X_1 = 0$ вихідний параметр два рази приймає значення, рівне одиниці. При $X_1 = 1$ вихідний параметр два рази приймає значення, рівне одиниці, і два рази дорівнює двом. При $X_1 = 2$ вихідний параметр один раз дорівнює нулю й один раз дорівнює двом.

$$\text{Тому: } \bar{y}_1 = \bar{y}(X_1 = 0) = \frac{1+1}{2} = 1,$$

$$\bar{y}_2 = \bar{y}(X_1 = 1) = \frac{2+1+1+2}{4} = 1,5 \text{ та } \bar{y}_3 = \bar{y}(X_1 = 2) = \frac{0+2}{2} = 1.$$

Підставляючи отримані значення у формулу для розрахунку дисперсійного відношення, і з огляду на, що $\bar{y} = 1,25$, одержуємо:

$$\eta(X_1) = \sqrt{\frac{2 \cdot (1-1,25)^2 + 4 \cdot (1,5-1,25)^2 + 2 \cdot (1-1,25)^2}{(0-1,25)^2 + (2-1,25)^2 + (1-1,25)^2 + (2-1,25)^2 + \dots + (1-1,25)^2}} = \sqrt{\frac{0,5}{3,5}} = 0,38.$$

Для розрахунку дисперсійного відношення $\eta(X_3)$ ознаки X_3 необхідно попередньо її дискретизувати. Для цього визначимо кількість інтервалів розбиття його значень:

$$N_3 = \text{Ціле}(\log_2(m)) + 1 = \text{Ціле}(\log_2(8)) + 1 = 4,$$

і довжину кожного інтервалу:

$$\Delta = \frac{\max(X_3) - \min(X_3)}{N_3} = \frac{6,2 - 2,2}{4} = 1.$$

Таким чином, одержуємо 4 інтервали: $[2,2; 3,2)$, $[3,2; 4,2)$, $[4,2; 5,2)$ і $[5,2; 6,2]$.

В інтервал $[2,2; 3,2)$ попадає одне значення X_3 ($X_3 = 2,2$), при цьому $Y = 0$. В інтервал $[3,2; 4,2)$ попадає три значення X_3 ($X_3 = 3,3; 4; 4,1$), при цьому вихідний параметр три рази дорівнює одиниці. В інтервал $[4,2; 5,2)$ попадає два значення X_3 ($X_3 = 4,3; 4,7$), при цьому вихідний параметр одного разу дорівнює одиниці й один раз дорівнює двом. В інтервал $[5,2; 6,2]$ попадає два значення X_3 ($X_3 = 5,6; 6,2$), при цьому вихідний параметр два рази дорівнює двом.

Тому:

$$\bar{y}_1 = \bar{y}(X_3 \in [2,2;3,2]) = \frac{0}{1} = 0,$$

$$\bar{y}_2 = \bar{y}(X_3 \in [3,2;4,2]) = \frac{1+1+1}{3} = 1,$$

$$\bar{y}_3 = \bar{y}(X_3 \in [4,2;5,2]) = \frac{1+2}{2} = 1,5,$$

$$\bar{y}_4 = \bar{y}(X_3 \in [5,2;6,2]) = \frac{2+2}{2} = 2.$$

Таким чином:

$$\eta(X_3) = \sqrt{\frac{1 \cdot (0 - 1,25)^2 + 3 \cdot (1 - 1,25)^2 + 2 \cdot (1,5 - 1,25)^2 + 2 \cdot (2 - 1,25)^2}{(0 - 1,25)^2 + (2 - 1,25)^2 + (1 - 1,25)^2 + (2 - 1,25)^2 + \dots + (1 - 1,25)^2}} = \sqrt{\frac{3}{3,5}} = 0,93;$$

г) при розрахунку коефіцієнта зв'язку $\xi(X_3)$ для ознаки X_3 скористаємося результатами, отриманими вище, а також обчислимо середнє значення квадрата \bar{y}^2 вихідної змінної Y і середні значення квадратів \bar{y}^2_k вихідного параметра для k -го інтервалу діапазону зміни третьої ознаки.

Середнє значення квадрата вихідної змінної Y :

$$\bar{y}^2 = \frac{1}{m} \sum_{p=1}^m y_p^2 = \frac{1}{8} (0^2 + 2^2 + 1^2 + 2^2 + 1^2 + 1^2 + 2^2 + 1^2) = 2.$$

Середні значення квадратів \bar{y}^2_k вихідного параметра для k -го інтервалу:

$$\bar{y}^2_1 = \bar{y}^2(X_3 \in [2,2;3,2]) = \frac{0^2}{1} = 0,$$

$$\bar{y}^2_2 = \bar{y}^2(X_3 \in [3,2;4,2]) = \frac{1^2 + 1^2 + 1^2}{3} = 1,$$

$$\bar{y}^2_3 = \bar{y}^2(X_3 \in [4,2;5,2]) = \frac{1^2 + 2^2}{2} = 2,5,$$

$$\bar{y}^2_4 = \bar{y}^2(X_3 \in [5,2;6,2]) = \frac{2^2 + 2^2}{2} = 4.$$

Дисперсія квадрата умовного математичного сподівання:

$$D(M(Y/X_i))^2 = \frac{1}{m} \sum_{k=1}^{N_i} n_k \left(\left(\bar{y}_k \right)^2 - \bar{y}^2 \right)^2 = \\ = \frac{1}{8} \left(1 \cdot (0^2 - 2)^2 + 3 \cdot (1^2 - 2)^2 + 2 \cdot (1,5^2 - 2)^2 + 2 \cdot (2^2 - 2)^2 \right) = 1,391$$

Дисперсія квадрата вихідного параметра Y :

$$DY^2 = \frac{1}{8} \left((0^2 - 2)^2 + (2^2 - 2)^2 + (1^2 - 2)^2 + (2^2 - 2)^2 + (1^2 - 2)^2 + (1^2 - 2)^2 + (2^2 - 2)^2 + (1^2 - 2)^2 \right) = 2,5. \\ A = \frac{1}{8} \left(1 \cdot (0 - 0^2) + 3 \cdot (1 - 1^2) + 2 \cdot (2,5 - 1,5^2) + 2 \cdot (4 - 2^2) \right) = 0,0625.$$

$$B = \frac{1}{8} \left(1 \cdot (0^2 - 2)(0 - 0^2 - 0,0625) + 3 \cdot (1^2 - 2)(1 - 1^2 - 0,0625) + \right. \\ \left. + 2 \cdot (1,5^2 - 2)(2,5 - 1,5^2 - 0,0625) + 2 \cdot (2^2 - 2)(4 - 2^2 - 0,0625) \right) = 0,15625.$$

Підставляючи отримані значення у формулу для розрахунку коефіцієнта зв'язку, одержуємо:

$$\xi(X_3) = \sqrt{\frac{1,391}{2,5 - 2 \cdot 0,15625}} = 0,893;$$

д) для розрахунку значення інформаційного критерію для ознаки X_4 необхідно врахувати, що він приймає три різних значення: 0, 1 і 2. При цьому ознака X_4 три рази дорівнює нулю, три рази – одиниці й два рази – двом. Вихідний параметр Y один раз приймає нульове значення, чотири рази дорівнює одиниці й три рази приймає значення, рівне двом.

Визначимо безумовні й умовні ймовірності різних подій, необхідні для обчислення значення інформаційного критерію:

– безумовні ймовірності віднесення вихідного параметра до класів 0, 1 і 2:

$$p_1 = p(Y = 0) = \frac{1}{8}; \quad p_2 = p(Y = 1) = \frac{4}{8}; \quad p_3 = p(Y = 2) = \frac{3}{8};$$

– імовірності того, що ознака X_4 прийме у k -те з можливих значень за умови, що вихідний параметр Y прийме l -те значення:

$$p(X_4 = 0/Y = 0) = 0; \quad p(X_4 = 0/Y = 1) = \frac{2}{4}; \quad p(X_4 = 0/Y = 2) = \frac{1}{3};$$

$$p(X_4 = 1/Y = 0) = 0; \quad p(X_4 = 1/Y = 1) = \frac{1}{4}; \quad p(X_4 = 1/Y = 2) = \frac{2}{3};$$

$$p(X_4 = 2/Y = 0) = 1; \quad p(X_4 = 2/Y = 1) = \frac{1}{4}; \quad p(X_4 = 2/Y = 2) = 0;$$

– імовірності того, що вихідний параметр y прийме q -е можливе значення за умови, що ознака X_4 прийме в k -е з можливих значень:

$$p(Y = 0/X_4 = 0) = 0; \quad p(Y = 0/X_4 = 1) = 0; \quad p(Y = 0/X_4 = 2) = \frac{1}{2};$$

$$p(Y = 1/X_4 = 0) = \frac{2}{3}; \quad p(Y = 1/X_4 = 1) = \frac{1}{3}; \quad p(Y = 1/X_4 = 2) = \frac{1}{2};$$

$$p(Y = 2/X_4 = 0) = \frac{1}{3}; \quad p(Y = 2/X_4 = 1) = \frac{2}{3}; \quad p(Y = 2/X_4 = 2) = 0.$$

Таким чином:

$$\begin{aligned} I(X_4) &= -\left(\frac{1}{8}\log_2\frac{1}{8} + \frac{4}{8}\log_2\frac{4}{8} + \frac{3}{8}\log_2\frac{3}{8}\right) + \frac{1}{8}\left(0+0+1\cdot\left[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2} + 0\right]\right) + \\ &+ \frac{4}{8}\left(\frac{2}{4}\left[0+\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right] + \frac{1}{4}\left[0+\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right] + \frac{1}{4}\left[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2} + 0\right]\right) + \\ &+ \frac{3}{8}\left(\frac{1}{3}\left[0+\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right] + \frac{2}{3}\left[0+\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right] + 0\right) = \\ &= 1,4056 + \frac{1}{8}\cdot(-1) + \frac{4}{8}\cdot\left(\frac{2}{4}\cdot(-0,9183) + \frac{1}{4}\cdot(-0,9183) + \frac{1}{4}\cdot(-1)\right) + \\ &+ \frac{3}{8}\cdot\left(\frac{1}{3}\cdot(-0,9183) + \frac{2}{3}\cdot(-0,9183)\right) = 1,4056 - 0,125 - 0,4069 - 0,3444 = 0,5293; \end{aligned}$$

е) при розрахунку значення ентропії ознаки X_2 врахуємо, що вона приймає 3 різні значення (0, 1 і 2). При цьому ознака X_2 три рази приймає нульове значення, чотири рази дорівнює одиниці й один раз – двом. Тому:

$$\begin{aligned}
e(X_2) &= -\left(\frac{3}{8}\left(\frac{1}{3}\log_2 \frac{1}{3} + \frac{2}{3}\log_2 \frac{2}{3} + 0\right) + \frac{4}{8}\left(0 + \frac{1}{4}\log_2 \frac{1}{4} + \frac{3}{4}\log_2 \frac{3}{4}\right) + \frac{1}{8}\left(0 + \frac{1}{1}\log_2 \frac{1}{1} + 0\right)\right) = \\
&= -\left(\frac{1}{8}\log_2 \frac{1}{3} + \frac{1}{4}\log_2 \frac{2}{3} + \frac{1}{8}\log_2 \frac{1}{4} + \frac{3}{8}\log_2 \frac{3}{4}\right) = 0,75.
\end{aligned}$$

$$\begin{aligned}
\text{Аналогічно міркуючи, одержуємо значення ентропії ознаки } X_4: \\
e(X_4) &= -\left(\frac{3}{8}\left(0 + \frac{2}{3}\log_2 \frac{2}{3} + \frac{1}{3}\log_2 \frac{1}{3}\right) + \frac{3}{8}\left(0 + \frac{1}{3}\log_2 \frac{1}{3} + \frac{2}{3}\log_2 \frac{2}{3}\right) + \frac{2}{8}\left(\frac{1}{2}\log_2 \frac{1}{2} + \frac{1}{2}\log_2 \frac{1}{2} + 0\right)\right) = \\
&= -\left(\frac{1}{8}\log_2 \frac{1}{3} + \frac{1}{4}\log_2 \frac{2}{3} + \frac{1}{8}\log_2 \frac{1}{4} + \frac{3}{8}\log_2 \frac{3}{4}\right) = 0,75.
\end{aligned}$$

ж) для обчислення значення ентропії набору ознак $X_e = \{X_2, X_4\}$ необхідно врахувати, що набір ознак $\{X_2, X_4\}$ два рази приймає комбінацію значень $\{0, 2\}$, три рази – комбінацію $\{0, 0\}$, по одному разу – комбінації значень $\{0, 0\}, \{1, 0\}$ і $\{2, 0\}$. Тоді:

$$e(X_e) = -\left(\frac{2}{8}\left(\frac{1}{2}\log_2 \frac{1}{2} + \frac{1}{2}\log_2 \frac{1}{2} + 0\right) + \frac{3}{8}\left(0 + \frac{1}{3}\log_2 \frac{1}{3} + \frac{2}{3}\log_2 \frac{2}{3}\right) + 0 + 0 + 0\right) = 0,59436;$$

з) оцінимо інформативність ознак X_5 і X_6 за допомогою імовірнісного підходу $Z(X_5)$ і $Z(X_6)$.

Вихідний параметр Y приймає 3 різні значення (0, 1 і 2), тому $N_y = 3$.

Визначимо безумовне й умовні математичні сподівання ознаки X_5 :

$$M_5 = M(X_5) = \frac{1+0+2+1+1+2+2+0}{8} = 1,125,$$

$$m_{51} = m(X_5/y=0) = \frac{1}{1} = 1, \quad m_{52} = m(X_5/y=1) = \frac{2+1+2+0}{4} = 1,25 \quad \text{та}$$

$$m_{53} = m(X_5/y=2) = \frac{0+1+2}{3} = 1.$$

Підставляючи отримані значення у формулу для оцінювання інформативності за допомогою імовірнісного підходу, одержуємо:

$$\begin{aligned}
Z(X_5) &= \frac{\frac{1}{8}\left((1-1)^2 + (2-1,125)^2 + (1-1,125)^2 + (2-1,125)^2 + (0-1,125)^2 + (0-1)^2 + (1-1)^2 + (2-1)^2\right)}{\frac{1}{3}\left((1-1,125)^2 + (1,25-1,125)^2 + (1-1,125)^2\right)} = 38.
\end{aligned}$$

Аналогічно знаходимо інформативність шостої ознаки $Z(X_6)$:

$$M_6 = M(X_6) = \frac{3+5+5+3+2+5+5+2}{8} = 3,75,$$

$$m_{61} = m(X_6 / y = 0) = \frac{3}{1} = 3, \quad m_{62} = m(X_6 / y = 1) = \frac{5+2+5+2}{4} = 3,5 \quad \text{та}$$

$$m_{63} = m(X_6 / y = 2) = \frac{5+3+5}{3} = 4,33.$$

$$Z(X_6) = \frac{\frac{1}{8}((3-3)^2 + (5-3)^2 + (2-3)^2 + (5-3)^2 + (2-3)^2 + (5-4,33)^2 + (3-4,33)^2 + (5-4,33)^2)}{\frac{1}{3}((3-3,75)^2 + (3,5-3,75)^2 + (4,33-3,75)^2)} = 4,53.$$

Оскільки $Z(X_6) < Z(X_5)$, то при використанні імовірнісного підходу можна зробити висновок про те, що ознака X_6 є більш інформативною у порівнянні з ознакою X_5 .

и) оцінимо інформативність ознакої X_5 за допомогою критерію, заснованого на статистичному підході: $S(X_5)$. Будемо думати, що для 1-й, 3-й, 5-й, 6-й і 8-й екземпляри відносяться до першого класу, а 2-й, 4-й, і 7-й екземпляри – до другого класу.

Визначимо середні значення п'ятої ознакої, за умови, що екземпляр відноситься до першого й другого класів:

$$\overline{X_{5,1}} = \frac{1+2+1+2+0}{5} = 1,2 \quad \text{та} \quad \overline{X_{5,2}} = \frac{0+1+2}{3} = 1.$$

Розрахуємо квадрати середньоквадратичних відхилень ознакої X_5 , за умови, що екземпляр відноситься до першого й другого класів:

$$\sigma_{5,1}^2 = D(X_5 / Y = 1) = \frac{1}{5}((1-1,2)^2 + (2-1,2)^2 + (1-1,2)^2 + (2-1,2)^2 + (0-1,2)^2) = 0,56,$$

$$\sigma_{5,2}^2 = D(X_5 / Y = 2) = \frac{1}{3}((0-1)^2 + (1-1)^2 + (2-1)^2) = 0,667.$$

Інформативність ознакої X_5 за допомогою критерію, заснованого на статистичному підході:

$$S(X_5) = \frac{1}{2}(0,56 - 0,667)\left(\frac{1}{0,56} - \frac{1}{0,667}\right) + \frac{1}{2}\left(\frac{1}{0,56} + \frac{1}{0,667}\right)(1,2 - 1) = 0,344;$$

к) при використанні набору ознак $X_e = \{X_2, X_4\}$ однозначно можуть бути класифіковані третій, четвертий і шостий екземпляри. Тому: $POS_{X_e}(Y) = \{3, 4, 6\}$, $|POS_{X_e}(Y)| = 3$. Номера інших екземплярів не ввійшли в множину $POS_{X_e}(Y)$, оскільки комбінація з ознаками X_2 і X_4 не дозволяє їх вірно класифікувати. Так, наприклад, перший і п'ятий екземпляри мають одинакові значення ознак

$x_{21} = x_{25} = 0$ і $x_{41} = x_{45} = 5$, але різні значення ознак: $y_1 = 0$, $y_5 = 1$. Таким чином:

$$\gamma_Y(Xe) = \frac{|POS_{Xe}(Y)|}{m} = \frac{3}{8} = 0,375.$$

 **Приклад 3.** Задано вибірку, що складається з дев'яти екземплярів, що характеризуються трьома ознаками, що приймають нечіткі значення. При цьому ознака X_1 може приймати значення x_{11}, x_{12}, x_{13} , $X_2 - x_{21}, x_{22}, x_{23}$, $X_3 - x_{31}, x_{32}$, вихідний параметр Y приймає значення y_1, y_2, y_3 . У табл. 3.5 наведені значення ймовірності того, що i -та ознака j -го екземпляра прийме значення x_{ijd} .

Необхідно розрахувати значення критерію оцінювання інформативності ознак X_1, X_2, X_3 , а також набору ознак $Xe = \{X_1, X_2\}$ на основі теорії нечітких множин: $\beta_Y(X_1), \beta_Y(X_2), \beta_Y(X_3)$ і $\beta_Y(Xe)$.

Таблиця 3.5 – Вибірка з нечіткими значеннями ознак і вихідного параметра

Номер екземпляра	X_1			X_2			X_3		Y		
	x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}	x_{31}	x_{32}	y_1	y_2	y_3
1	0,3	0,7	0	0,2	0,7	0,1	0,3	0,7	0,1	0,9	0
2	1	0	0	1	0	0	0,7	0,3	0,8	0,2	0
3	0	0,3	0,7	0	0,7	0,3	0,6	0,4	0	0,2	0,8
4	0,8	0,2	0	0	0,7	0,3	0,2	0,8	0,6	0,3	0,1
5	0,5	0,5	0	1	0	0	0	1	0,6	0,8	0
6	0	0,2	0,8	0	1	0	0	1	0	0,7	0,3
7	1	0	0	0,7	0,3	0	0,2	0,8	0,7	0,4	0
8	0,1	0,8	0,1	0	0,9	0,1	0,7	0,3	0	0	1
9	0,3	0,7	0	0,9	0,1	0	1	0	0	0	1

Розрахуємо значення критерію інформативності нечіткої ознаки X_1 за формулою:

$$\beta_Y(X_1) = \frac{\sum_{l=1}^m \mu_{POS_{X_1}(Y)}(l)}{m}.$$

Визначимо значення функції належності нечіткої множини $POS_{X_1}(Y)$ першого екземпляра $\mu_{POS_{X_1}(Y)}(1) = \max(\mu_{X_1 y_1}(1); \mu_{X_1 y_2}(1); \mu_{X_1 y_3}(1))$.

$$\mu_{X_1y_1}(1) = \max_{b=1,2,3} \left[\min \left(\mu_{X_{1b}}(1); \min_{a=1,2,\dots,9} \max(1 - \mu_{X_{1b}}(a); \mu_{y_1}(a)) \right) \right].$$

При $b = 1$:

$$\text{Аналогічно при } b = 2: \min_{a=1,2,\dots,9} \max(1 - \mu_{x_{12}}(a); \mu_{y_1}(a)) = 0,2,$$

$$\text{при } b = 3: \min_{a=1,2,\dots,9} \max(1 - \mu_{x_{13}}(a); \mu_{y_1}(a)) = 0,2.$$

Таким чином:

$$\begin{aligned} \mu_{X_1y_1}(1) &= \max(\min(\mu_{x_{11}}(1); 0,6); \min(\mu_{x_{12}}(1); 0,2); \min(\mu_{x_{13}}(1); 0,2)) = \\ &= \max(\min(0,3; 0,6); \min(0,7; 0,2); \min(0; 0,2)) = 0,3. \end{aligned}$$

$$\text{Аналогічно } \mu_{X_1y_2}(1) = 0,2 \text{ і } \mu_{X_1y_3}(1) = 0,3.$$

$$\text{Тому } \mu_{POS_{Xe}(Y)}(1) = \max(0,3; 0,2; 0,3) = 0,3.$$

Міркуючи аналогічно, одержуємо:

$$\mu_{POS_{Xe}(Y)}(2) = 0,6; \quad \mu_{POS_{Xe}(Y)}(3) = 0,3; \quad \mu_{POS_{Xe}(Y)}(4) = 0,6;$$

$$\mu_{POS_{Xe}(Y)}(5) = 0,5; \quad \mu_{POS_{Xe}(Y)}(6) = 0,3; \quad \mu_{POS_{Xe}(Y)}(7) = 0,6;$$

$$\mu_{POS_{Xe}(Y)}(8) = 0,3; \quad \mu_{POS_{Xe}(Y)}(9) = 0,3.$$

Разом:

$$\beta_Y(X_1) = \frac{0,3 + 0,6 + 0,3 + 0,6 + 0,5 + 0,3 + 0,6 + 0,3 + 0,3}{9} = \frac{3,8}{9} = 0,422.$$

У такий же спосіб знаходимо $\beta_Y(X_2)$ і $\beta_Y(X_3)$:

$$\beta_Y(X_2) = \frac{2,1}{9} = 0,233, \quad \beta_Y(X_3) = \frac{2,7}{9} = 0,3.$$

При оцінюванні набору ознак $Xe = \{X_1, X_2\}$ необхідно врахувати, що ознаки такої комбінації можуть приймати $B_{Xe} = \prod_{X_i \in Xe} |X_i| = 3 \cdot 3 = 9$

значень: $\{x_{11}, x_{21}\}, \{x_{11}, x_{22}\}, \{x_{11}, x_{23}\}, \{x_{12}, x_{21}\}, \{x_{12}, x_{22}\}, \{x_{12}, x_{23}\}, \{x_{13}, x_{21}\}, \{x_{13}, x_{22}\}, \{x_{13}, x_{23}\}$. Для кожного з таких значень необхідно визначити $\mu_{x_{1i}x_{2j}y_k}(a), i, j, k = 1, 2, 3, a = 1, 2, \dots, 9$, після чого обчислити

$\mu_{POS_{Xe}(Y)}(a)$. У результаті одержимо:

$$\beta_Y(Xe) = \frac{4}{9} = 0,444.$$



3.8 Контрольні питання

1. З якою метою виконують відбір інформативних ознак?
2. Наведіть постановку задачі відбору ознак.
3. Що є результатом виконання процедури відбору ознак?
4. Дайте визначення таких понять: інформативність ознаки, незначущі ознаки, надлишкові ознаки.
5. Проаналізуйте методи відбору інформативних ознак.
6. Порівняйте методи повного перебору.
7. Наведіть переваги та недоліки методів скороченого перебору.
8. Для оцінювання якого типу зв'язку використовується коефіцієнт парної кореляції?
9. До чого призводить наявність надлишкових та неінформативних ознак?
10. На скільки змінюється розмір наборів ознак на кожній ітерації у методі групового врахування аргументів?
11. Наведіть послідовність виконання методів послідовного додавання та видалення ознак.
12. Порівняйте критерії оцінювання індивідуальної інформативності.
13. Проаналізуйте процедури пошуку оптимального набору ознак.
14. У чому полягає основна ідея методу випадкового пошуку з адаптацією?
15. У чому полягає основна ідея методу групового врахування аргументів?
16. У яких випадках доцільним є використання методів класифікації ознак?
17. Який з методів виділення набору ознак відноситься до евристичних методів?
18. Який критерій може бути використаний в якості критерію закінчення пошуку у методі послідовного додавання ознак?
19. Який недолік має метод класичного повного перебору?
20. Який тип можуть мати значення ознак при оцінюванні інформативності ознак за допомогою коефіцієнта кореляції знаків?
21. Яким може бути подана постановка задачі відбору інформативних ознак?

22. Яким чином визначається середньоквадратична помилка, для розрахунку помилки синтезованої моделі у випадку відбору ознак при вирішенні задачі прогнозування?

23. Яким чином визначається середня відносна помилка, для розрахунку помилки синтезованої моделі у випадку відбору ознак при рішенні задачі прогнозування?

24. Яким чином можуть використовуватися методи еволюційного пошуку до відбору інформативних ознак?

25. Яким шляхом відбувається формування нових рішень у методі випадкового пошуку з адаптацією?

26. Які критерії використовуються для оцінювання спільногопливу набору ознак?

27. Які критерії використовуються у методі ранжирування ознак?

28. Які критерії оцінювання інформативності ознак використовує метод гілок і границь при відборі ознак?

29. Які множини можуть бути використані як початкова точка у методах відбору ознак?

30. Які стратегії використовують евристичні методи відбору ознак?

31. Які стратегії використовуються для оцінювання інформативності набору ознак?

3.9 Практичні завдання



Завдання 1. Проведіть порівняння методів відбору інформативних ознак. Сформуйте набір критеріїв порівняння та складіть порівняльну таблицю.



Завдання 2. Критерії оцінювання індивідуальної інформативності.

2.1 За допомогою середовища Matlab згідно з номером студента за журналом для відповідного номера варіанту V сформувати навчальну вибірку $\langle x, y \rangle$ обсягом m екземплярів, $j = 1, 2, \dots, m$, що характеризуються L ознаками x_{ij} , $i = 1, 2, \dots, L$, та зіставити кожному екземпляру значення цільової ознаки y_j :

$$x_{ij} = \begin{cases} iV - 0,1j, & i = 1, 5, 9, \dots, \\ 0,01iV^{-1} + 0,3j, & i = 2, 4, 6, \dots, \\ i \text{ rand}, & i = 3, 7, 11, \dots; \end{cases} \quad y_j = 2x_{1j} + 0,1x_{2j};$$

$$m = \begin{cases} 10V, & V < 10, \\ 5V, & 10 \leq V < 20, \\ 3V, & V \geq 20; \end{cases} \quad L = \begin{cases} 5V, & V < 7, \\ 4V, & 7 \leq V < 10, \\ 3V, & 10 \leq V < 20, \\ 2V, & V \geq 20; \end{cases}$$

де $rand$ – випадкове число в діапазоні $[0, 1]$.

2.2 На алгоритмічній мові програмування пакету Matlab написати програму, що дозволяє оцінювати значення критеріїв індивідуальної інформативності ознак:

- коефіцієнт парної кореляції;
- коефіцієнт кореляції знаків;
- коефіцієнт кореляції Фехнера;
- дисперсійне відношення;
- коефіцієнт зв'язку;
- інформаційний критерій;
- теоретико-інформаційний критерій;
- ентропія ознаки;
- критерій, заснований на імовірнісному підході;
- критерій, заснований на статистичному підході.

2.3 За допомогою розробленої у п.2.2 програми оцінити інформативність ознак екземплярів вибірки, використовуючи різні критерії індивідуальної інформативності. При необхідності, значення вихідної ознаки можна дискретизувати.

2.4 Побудувати таблицю з оцінками інформативності ознак відносно вихідного параметру y , стовпці якої повинні мати назви:

- 1) номер ознаки;
- 2) коефіцієнт парної кореляції;
- 3) коефіцієнт кореляції знаків;
- 4) коефіцієнт кореляції Фехнера;
- 5) дисперсійне відношення;
- 6) коефіцієнт зв'язку;
- 7) інформаційний критерій;
- 8) теоретико-інформаційний критерій;
- 9) ентропія ознаки;
- 10) критерій, заснований на імовірнісному підході;
- 11) критерій, заснований на статистичному підході;

2.5 Проаналізувати за побудованими таблицями оцінки інформативності ознак. Зробити висновки щодо важливості ознак відповідно до вихідного параметру y .



Завдання 3. Методи відбору інформативних ознак.

3.1 За допомогою мови пакету Matlab згідно з номером студента за журналом розробити програму, що реалізує два методи відбору інформативних ознак:

- 1) метод повного перебору;
- 2) пошук у глибину;
- 3) пошук завширшки;
- 4) метод гілок і границь або скорочений пошук у глибину;
- 5) метод групового врахування аргументів або скорочений пошук завширшки;
- 6) метод послідовного додавання ознак;
- 7) метод послідовного видалення ознак;
- 8) метод почергового додавання й видалення ознак;
- 9) ранжирування ознак;
- 10) кластеризація ознак;
- 11) випадковий пошук з адаптацією;
- 12) еволюційний пошук.

3.2 Для вибірки, сформованої у попередньому завданні, використовуючи розроблену програму, за допомогою різних методів виділити комбінацію інформативних ознак.

При цьому для оцінювання значущості набору ознак (груповій інформативності) використовувати такі критерії:

- 1) середньоквадратична помилка;
- 2) сума квадратів відхилень;
- 3) середня абсолютна помилка;
- 4) сума значень абсолютних відхилень;
- 5) максимальне абсолютноне відхилення;
- 6) середня відносна помилка;
- 7) сума відносних відхилень;
- 8) максимальне відносне відхилення.

В якості моделі для обчислення значущості набору ознак з використанням критеріїв, наведених вище, обрати лінійну багатовимірну регресію.

3.3 Проаналізувати отримані набори ознак. Зробити висновки щодо доцільності використання того чи іншого методу відбору ознак.

 **Завдання 4.** Використовуючи пакет Matlab, написати програму для синтезу інформативних ознак за допомогою методів:

- метод головних компонентів (Principal Component Analysis);
- метод незалежних компонентів (Principal Independent Analysis).

Для вибірки, сформованої у завданні № 2, використовуючи розроблену програму, синтезувати набір інформативних ознак.

Виконати аналіз отриманих результатів.

 **Завдання 5.** Написати реферат на одну з таких тем.

1. Методи відбору інформативних ознак.
2. Критерій оцінювання індивідуальної значущості.
3. Стохастичний підхід до відбору ознак.
4. Методи синтезу інформативних ознак.
5. Програмні засоби відбору та синтезу ознак.

 **3.10 Література до розділу**

Питання відбору інформативних ознак для побудови розпізнавальних моделей розглянуто в [3, 6–8, 10, 11, 14, 16, 17].

РОЗДІЛ 4

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ, ЗАСНОВАНІ НА ЗНАННЯХ

При використанні традиційних структурних мов програмування необхідні знання вміщувалися безпосередньо в прикладну програму й утворювали з нею одне ціле. Однак такий підхід ускладнює розуміння того, яким чином використовуються знання і яку роль вони виконують, тобто знання, закладені в програму, і сама програма їхньої обробки виявляються жорстко пов'язаними між собою і дозволяють одержувати тільки ті висновки з наявних знань, що передбачені програмою їхньої обробки. Для вирішення цієї проблеми використовують системи, засновані на знаннях.

4.1 Знання та їх властивості

Інформація (у широкому розумінні) – це будь-які відомості про певний об'єкт, процес або явище.

Факти – це інформація, що розглядається як надійна.

Ієархію способів подання інформації (у порядку збільшення рівня ієархії) виділяють таку.

1. *Шум* – відсутність видимих ознак інформації, складається з інформаційних елементів, що не становлять інтересу і можуть лише ускладнити сприйняття й подання інформації.

2. *Дані* – потенційне джерело інформації – елементи інформації, що у принципі можуть становити певний інтерес. Даними називають інформацію фактичного характеру, що є фіксованою певним способом та описує об'єкти, процеси і явища предметної області, а також їхні властивості. Дані являють собою ізольовані факти, відносини між якими з зовнішнім світом у них самих не зафіксовані.

3. *Інформація* (у вузькому розумінні) – потенційне джерело знань – оброблені дані, що явно становлять інтерес для користувачів.

4. *Знання* – це формалізована система суджень із принциповою і єдиною організацією, заснована на об'єктивній закономірності, що спостерігається у визначеній предметній області (принципи, зв'язки, закони), встановлений в результаті розумової діяльності людини, спрямованої на узагальнення досвіду, отриманого нею у результаті практичної діяльності, яка дозволяє ставити і вирішувати задачі в цій області. Знання визначають здатність вико-

ристовувати інформацію і являють собою добре структуровані дані або *метадані* (дані про дані) – елементи інформації, зв’язані між собою і з зовнішнім світом.

5. *Метазнання* – спеціальним чином організовані знання про знання з метою реалізації процесу їхньої інтерпретації і планування виведення. Метазнання дозволяють інтелектуальній системі виправляти або доповнювати свої знання в міру навчання в процесі вирішення конкретних задач.

Онтологія в експертних системах являє собою метазнання, що описують усе, що відомо про розглянуту предметну область. В ідеальному випадку онтологія повинна бути подана у формальному виді задля того, щоб можна було легко виявляти несумісності і невідповідності.

Пояснення в експертних системах – вид метазнань, знання системи про саму себе, про спосіб використання своїх знань для одержання рішення. Крім того, за метазнаннями, отриманими пояснюючої підсистемою, можна згенерувати дерево пояснень для формування підказки, повторного розширеного питання, навчального матеріалу або безпосереднього пояснення етапів рішення й у цілому результатів.

6. *Мудрість* – здатність використовувати знання щонайкраще – це метазнання, що дозволяють визначати найкращі цілі в житті і знаходити шляхи їхнього досягнення.

Питаннями збору, обробки, збереження даних та інформації займається *інформатика*. Мудрість є філософською категорією. Тому подальшим предметом розгляду є знання.

Епістемологія – наука про знання. У рамках цієї науки розглядаються характер, структура і походження знань.

Типи знань виділяють такі.

1. *Базові елементи знання* – інформація про властивості об’єктів реального світу. Пов’язані з безпосереднім сприйняттям, не вимагають обговорення і використовуються в тому виді, у якому отримані.

2. *Твердження і визначення* – засновані на базових елементах і заздалегідь розглядаються як достовірні.

3. *Концепції* – перегрупування або узагальнення базових елементів. Для побудови кожної концепції використовуються

свої прийоми (приклади, контрприклади, окрім випадки, більш загальні випадки, аналогії).

4. *Відношення* – виражаютъ як елементарні властивості базових елементів, так і відношення між концепціями. До властивостей відношень відносять їхні більші або менші правдоподібність і зв'язок з даною ситуацією.

5. *Теореми і правила перезапису* – окремий випадок продукційних правил (правил виду «якщо..., то..., інакше...») з цілком визначеними властивостями. Теореми не мають користі без експертних правил їхнього застосування.

6. *Алгоритми рішення* – необхідні для виконання визначених задач. В усіх випадках вони пов'язані зі знанням особливого типу, оскільки обумовлена ними послідовність дій виявляється оформленою в блок у строго визначеному порядку, на відміну від інших типів знань, де елементи знання можуть з'являтися і розташовуватися без зв'язку один з одним.

7. *Стратегії й евристика* – уроджені або придбані правила поведінки, що дозволяють у конкретній ситуації прийняти рішення про необхідні дії. Людина постійно користається цим типом знань при формуванні концепцій, вирішенні задач і формальних розсудах.

8. *Метазнання* – є присутнім на багатьох рівнях і подає знання того, що відомо, визначає значення коефіцієнта довіри до цього знання, важливість елементарної операції стосовно всієї множини знань. Сюди ж відносяться питання організації різного типу знань і вказівки, де, коли і як вони можуть бути використані.

Різновиди знань виділяють такі.

1. *Ап'riорні знання* (від лат. *a priori* – з попереднього) передують знанням, отриманим за допомогою органів почуттів, і не залежать від них. Ап'riорні знання розглядаються як універсально істинні, і ці знання неможливо спростувати, не впадаючи в протиріччя.

2. *Апостерiорні знання* (від лат. *a posteriori* – з наступного) – знання, отримані за допомогою органів почуттів – є протилежними стосовно ап'riорних знань. Істинність або хибність апостерiорних знань може бути перевірена на пiдставi чуттєвого досвiду. Але чуттєвий досвiд не завжди може виявитися надiйним, тому iснує iмовiрнiсть того, що апостерiорнi знання будуть спростованi на основi нових знань.

Виділяють такі *види знань*.

1. *Процедурні знання* – знання, що задають послідовності дій, які мають бути виконані, і послідовності цілей, які мають бути досягнуті – відносяться до процедур обробки інформації і методів логічного виведення.

2. *Декларативні знання* – знання про те, чи є певне твердження істинним або помилковим – включають факти або аксіоми і правила, що відносяться до цих фактів. Термін *декларативний* застосовується до знань, виражених у формі декларативних тверджень. Для декларативних форм є особливістю організація бази знань, при якій у ній зберігаються тільки описи об'єктів і їхніх семантичних відношень і відсутня інформація про те, як можуть бути використані дані описи.

3. *Неявні знання* – підсвідомі знання, що не можуть бути виражені за допомогою мови. Якщо мова йде про комп’ютерні системи, то знання, подані в штучній нейронній системі, нагадують неявні знання, оскільки звичайно нейронна мережа нездатна безпосередньо пояснити суть знань, що містяться в ній, але могла б придбати таку здатність за наявності відповідної програми.

Логічною прозорістю системи називають здатність системи пояснювати методику прийняття рішення. Під цим розуміється, наскільки просто людині з’ясувати, що робить система і чому.

Експертні знання – спеціалізований різновид знань і навичок, яким володіють експерти. Експертні знання можуть відноситися до рівнів знань, метазнань і мудрості. Вони являють собою ті неявні знання і навички експерта, що повинні бути витягнуті і перетворені в явні задля того, щоб їх можна було подати в експертній системі. Причина, через яку знання є неявними, полягає в тому, що справжній експерт володіє цими знаннями настільки добре, що вони перетворилися в його другу натуру і не вимагають міркувань.

У залежності від часу існування знання виділяють:

– *статичні знання* – включають логічні правила рішення задач, правила реалізації процедур опитування експертів і користувачів, правила побудови функціональних та діагностичних моделей і аналізу результатів роботи, а також безпосередньо програмні модулі процедур опитування і математичних моделей, факти і дані про предметну область;

– *динамічні знання* – являють собою сукупність фактів і даних, одержуваних у ході рішення задачі, а також висновків (логічних висновків і строгих аналітичних рішень), вироблених у процесі рішення задачі.

У загальному вигляді знання подаються певною *семіотичною* (знаковою) *системою*. З поняттям «знак» безпосередньо зв'язані поняття денотат і концепт. *Денотат* – це об'єкт, що позначається даним знаком, а *концепт* – властивість денотата.

Інтенсіонал знака визначає зміст пов'язаного з ним поняття через його властивості, тобто значенневий зміст поняття. *Інтенсіонал* відокремлює знання від даних, що завжди задаються екстенсіонально.

Екстенсіонал знака визначає конкретний клас усіх його приступимих денотатів. *Екстенсіонал* поняття – набір конкретних фактів, що відповідають даному поняттю.

Інтенсіональні знання описують абстрактні об'єкти, події, відношення.

Екстенсіональні знання являють собою дані, що характеризують конкретні об'єкти, їхній стан, значення параметрів у визначені моменти часу.

Зазначені відмінності призвели до появи спеціальних формалізмів у вигляді *моделей подання знань*.

Аспекти семіотичної системи виділяють такі.

1. *Синтаксис* описує внутрішній пристрій знакової системи, тобто правила побудови і перетворення складних знакових виразів. Для природної мови, як відомо, синтаксис визначає правила побудови речень і зв'язаного тексту.

2. *Семантика* визначає відношення між знаками і їх концептами, тобто задає зміст або значення конкретних знаків.

3. *Прагматика* визначає знак з погляду конкретної сфери його застосування або суб'єкта, що використовує дану знакову систему.

Відповідно до перерахованих аспектів семіотичних систем можна виділити три *типи знань*:

– *синтаксичні знання* – характеризують синтаксичну структуру описаного об'єкта або явища, що не залежить від смислу і змісту використовуваних при цьому понять;

– *семантичні знання* – містять інформацію, безпосередньо пов'язану зі значеннями і змістом описуваних явищ і об'єктів;

– *прагматичні знання* – описують об'єкти і явища з погляду розв'язуваної задачі, наприклад, з урахуванням діючих у даній задачі специфічних критерій.

Знання, якими володіє фахівець у якій-небудь області (дисципліні), можна розділити на формалізовані (точні) і неформалізовані (неточні).

Формалізовані знання формулюються в книгах і посібниках у виді загальних і строгих суджень (законів, формул, моделей, алгоритмів і т. п.), що відбувають універсальні знання.

Неформалізовані знання, як правило, не попадають до книг і посібників у зв'язку з їхньою конкретністю, суб'єктивністю і приближністю. Знання цього роду є результатом узагальнення багаторічного досвіду роботи й інтуїції фахівців. Вони звичайно являють собою різноманіття емпіричних (евристичних) прийомів і правил.

У залежності від того, які знання переважають у тій чи іншій області (дисципліні), її відносять до формалізованих (якщо переважають точні знання) або до неформалізованих (якщо переважають неточні знання) описових областей.

Задачі, розв'язувані на основі точних знань, називають *формалізованими*, а задачі, розв'язувані за допомогою неточних знань, – *неформалізованими*. Тут мова йде не про такі задачі, які не можна формалізувати, а про такі задачі, формалізація яких є невідомою.

До неформалізованих задач відносять ті, котрі мають одну чи декілька з таких особливостей: алгоритмічне рішення задачі є невідомим (хоча, можливо, й існує) або не може бути використане через обмеженість ресурсів ЕОМ (часу, пам'яті); задача не може бути визначена в числовій формі (потрібно символічне подання); цілі задачі не можуть бути виражені в термінах точно визначеної цільової функції. Як правило, неформалізовані задачі мають неповноту, помилковість, неоднозначність та (або) суперечливість знань (як даних, так і використовуваних правил перетворення).

Форми існування знань в інтелектуальних системах:

- вихідні знання (правила, виведені на основі практичного досвіду, математичні й емпіричні залежності, що відбувають взаємні зв'язки між фактами; закономірності і тенденції, що описують зміну фактів з часом; функції, діаграми, графи);

- опис вихідних знань засобами обраної моделі подання знань (множина логічних формул або продукційних правил, семантична мережа, ієархії фреймів і т. п.);

- подання знань структурами даних, що призначенні для збереження й обробки в ЕОМ;

- бази знань на машинних носіях інформації.

Особливості знань виділяють такі.

1. *Внутрішня інтерпретованість* – властивість знань, що забезпечує можливість їхньої змістової інтерпретації без використання відповідної програми, що відрізняє їх від даних, які у відриві від програми не несуть ніякої змістової інформації і можуть змістовою інтерпретуватися лише відповідною програмою.

2. *Наявність класифікуючих відношень (структурованість)* – властивість знань, що визначає можливість довільного встановлення між окремими одиницями знань відношень типу «частина – ціле», «рід – вид», «елемент – клас», «клас – підклас», «тип – підтип», « ситуація – під ситуація» для забезпечення рекурсивної вкладеності одних одиниць знань в інші. Кожна одиниця знань може бути включена до складу будь-якої іншої, і з кожної інформаційної одиниці можна виділити деякі складові її інформаційні одиниці. Це дозволяє записати і зберігати окремо інформацію, однакову для всіх елементів множини. При необхідності цю інформацію можна автоматично передати опису будь-якого елемента множини. Такий процес передачі називається *спайдкуванням інформації*.

3. *Наявність ситуативних зв'язків* – здатність знань відбивати закономірності щодо фактів, процесів, явищ і причинно-наслідкові відношення між ними. Ситуативні зв'язки допомагають будувати процедури аналізу знань на сумісність, суперечливість і інші, котрі важко реалізувати при збереженні традиційних масивів даних.

4. *Шкаловання* використовується для фіксації співвідношень окремих інформаційних одиниць на основі різних шкал.

5. *Активність* – властивість знань впливати на інформаційні процеси і дії інтелектуальної системи, що відрізняє їх від даних, які є пасивними.

4.2 Принципи побудови систем, заснованих на знаннях

Система, заснована на знаннях – система програмного забезпечення, основними структурними елементами якої є бази знань і механізм (машина) логічних виведень. У системи, засновані на знаннях, також входять як модулі: підсистема набуття знань, підсистема пояснень та інтерфейсна підсистема.

У системі, заснованій на знаннях, знання подаються в конкретній формі в базі знань, що дозволяє їх легко визначати, модифікувати і поповнювати; функції вирішення задач реалізуються автономним механізмом логічних виведень, що робляться на

знаннях, які зберігаються в базі. Саме вибір методів подання й одержання знань визначає архітектуру системи знань і на практиці виражається у відповідній організації бази знань і схеми керування машиною виведення.

Невід'ємними характеристиками систем, заснованих на знаннях, із практичної точки зору вважаються їхня здатність пояснювати лінію суджень та можливість набуття і нарощування знань.

Система, заснована на знаннях, може бути описана такою *iерархією рівнів* (у порядку зменшення рівнів ієрархії).

1. *Рівень знань* (knowledge level) – пов'язаний зі змістом інформації, а також способами її використання і визначає можливості інтелектуальної системи. Самі знання не залежать від формалізмів, використовуваних для їхнього подання, а також виразності обраної мови програмування. На рівні знань зважуються питання про те, які запити є припустимими в системі, які об'єкти і відношення відіграють важливу роль у даній предметній області, як додати в систему нові знання, чи будуть факти згодом змінюватися, як у системі будуть реалізовані розсуди про знання, чи має дана предметна область добре зрозумілу систематику, чи є в ній незрозуміла або неповна інформація.

2. *Рівень символів* (symbol level) – пов'язаний з конкретними формалізмами, застосовуваними для подання знань у процесі вирішення задач. На цьому рівні здійснюється вибір конкретного способу подання знань і визначається мова подання для бази знань, зокрема, логічні або продукційні правила. Відділення рівня символів від рівня знань дозволяє програмісту вирішувати проблеми виразності, ефективності і простоти програмування, що не відноситься до більш високих рівнів поводження системи.

3. *Рівень алгоритмів і структур даних* – визначає структури даних для подання знань і алгоритми їхньої обробки.

4. *Рівень мов програмування* – визначає використовуваний стиль програмування. Хоча гарний стиль програмування припускає поділ конкретних властивостей мови програмування і вищестоящих рівнів, специфіка задач штучного інтелекту вимагає їхнього глибокого взаємозв'язку.

5. *Рівень компонування* – визначає архітектуру і функціональність операційної системи.

6. Рівень апаратних засобів – визначає архітектуру апаратних засобів, обсяг пам'яті і швидкодію процесора. Багаторівневий підхід дозволяє програмісту відволіктися від складності, що відноситься до нижніх рівнів, і сконцентрувати свої зусилля на питаннях, що відповідають даному рівню абстракції. Такий підхід дозволяє виділити теоретичні основи штучного інтелекту й абстрагуватися від деталей конкретної реалізації або мови програмування. Він дозволяє модифікувати реалізацію, підвищуючи її ефективність, або виконати портирування на іншу платформу, не торкаючись поводження системи на більш високих рівнях.

4.3 Експертні системи

Експертна система – це програмний засіб, що використовує експертні знання у певній предметній області з метою ефективного вирішення задач у предметній області, яка цікавить користувача, на рівні середнього професіонала (експерта).

Усі експертні системи є системами, заснованими на знаннях і програмами штучного інтелекту, але не навпаки. Інтелектуальні системи – найбільш загальний клас систем, які демонструють інтелектуальне поводження вмілим застосуванням евристик; системи, засновані на знаннях – підклас інтелектуальних систем, що роблять знання предметної області явними і відокремлюють їх від іншої частини системи; експертні системи – підклас систем, заснованих на знаннях, що застосовують експертні знання до складніших задач реального життя.

Експертні системи використовуються для вирішення так званих *неформалізованих задач*, загальним для яких є те, що: задачі недостатньо добре розуміються або вивчені; задачі не можуть бути задані в числовій формі, але можуть бути дослідженні за допомогою механізму символічних суджень; цілі не можна виразити в термінах точно визначеної цільової функції; не існує відомого алгоритмічного рішення задачі; якщо алгоритмічне рішення є, то його не можна використовувати через обмеженість ресурсів (час, пам'ять). Крім того неформалізовані задачі мають помилковість, неповноту, неоднозначність і суперечливість як вихідних даних, так і знань про розв'язувану задачу.

Властивості експертних систем, що відрізняють їх від звичайних програм:

- накопичення й організація знань про предметну область у процесі побудови й експлуатації експертної системи;
- явність і доступність знань;
- застосування для вирішення проблем високоякісного досвіду, що подає рівень мислення найбільш кваліфікованих експертів у даній області, який веде до творчих, точних і ефективних рішень;
- моделювання не стільки фізичної (чи іншої) природи визначеної проблемної області, скільки механізму мислення людини стосовно до вирішення задач у цій проблемній області;
- наявність прогностичних можливостей, за яких експертна система видає відповіді не тільки для конкретної ситуації, але і показує, як змінюються ці відповіді в нових ситуаціях, з можливістю докладного пояснення яким чином нова ситуація призвела до змін;
- забезпечення такої нової якості, як *інституціональна пам'ять*, за рахунок бази знань, що входить до складу експертної системи та розроблена в ході взаємодії з фахівцями організації, і являє собою поточну політику цієї групи людей. Цей набір знань стає зведенням кваліфікованих думок і постійно поновлюваним довідником найкращих стратегій і методів, використовуваних персоналом. Провідні спеціалісти ідуть, але їхній досвід залишається;
- можливість використання для навчання і тренування керівників, забезпечуючи нових службовців великим багажем досвіду і стратегій, за якими можна вивчати політику, що рекомендується, і методи;
- явний поділ засобів керування і даних;
- слабка детермінованість керування;
- використання при вирішенні задач евристичних і наближених методів, які, на відміну від алгоритмічних, не завжди гарантують успіх. *Евристика* є правилом впливу (rule of thumb), що у машинному вигляді подає деяке знання, набуте людиною в міру накопичення практичного досвіду вирішення аналогічних проблем. Такі методи є приблизними в тому змісті, що, по-перше, вони не вимагають вичерпної вихідної інформації, і, по-друге, існує визначений ступінь упевненості (чи непевності) у тому, що пропоноване рішення є вірним;
- здатність до символічних суджень: здатність подавати знання в символному вигляді і переформулювати символні знання;
- прийняття рішень на основі правил і логічного виведення;

- організація способу керування ходом виконання з використанням машини логічного виведення;
- *самосвідомість* – здатність досліджувати свої судження (тобто перевіряти їхню правильність) і пояснювати свої дії;
- здатність до навчання на своїх помилках;
- можливість застосування неповні чи неправильні вхідні дані;
- *компетентність* – здатність досягати експертного рівня рішень (в конкретній предметній області мати той же рівень професіоналізму, що й експерти-люди), бути вмілою (застосовувати знання ефективно і швидко, уникаючи, як і люди, непотрібних обчислень), мати *адекватну робастність* (здатність лише поступово знижувати якість роботи у міру наближення до меж діапазону або компетентності припустимої надійності даних);
- *логічна адекватність* – здатність подання знань експертної системи розпізнавати усі відмінності, що складаються у вихідні сутності;
- *євристична потужність* – наявність поряд з виразною мовою подання деякого засобу використання подань, сконструйованих і інтерпретованих таким чином, щоб з їхньою допомогою можна було вирішити проблему;
- *природність нотації* – зручність і простота виразів, якими формально описуються знання в експертній системі, зрозумілість їхнього змісту навіть тим, хто не знає, яким чином комп’ютер інтерпретує ці вирази;
- *логічна прозорість* – здатність експертної системи пояснити методику прийняття рішення, яка обумовлює те, наскільки просто персоналу з’ясувати, що робить програма і чому;
- *глибина* – здатність експертної системи працювати в предметній області, що містить важкі задачі, і використовувати складні правила (використовувати складні конструкції правил або велику кількість);
- *корисність* – здатність експертної системи в ході діалогу визначати потреби користувача, виявляти й усувати причини невдач у роботі, а також вирішувати поставлені задачі;
- *гнучкість* – здатність системи налагоджуватися на різних користувачів, а також враховувати зміни в кваліфікації одного й того ж самого користувача;
- *зручність роботи* – природність взаємодії з експертною системою (спілкування в звичному виді, що втомлює користувача),

її гнучкість і стійкість системи до помилок (здатність не виходити з ладу при помилкових діях недосвідченого користувача).

Класифікація експертних систем можлива за різними критеріями. Виділяють такі види експертних систем.

1. *За метою створення*: для навчання фахівців, для вирішення задач, для автоматизації рутинних робіт, для тиражування знань експертів.

2. *За основним користувачем*: для не фахівців в галузі експертиз, для фахівців, для учнів.

3. *За типами розв'язуваних задач*:

– *інтерпретуючі системи* – призначені для формування опису ситуацій за результатами спостережень або даними, одержуваними від різного роду сенсорів. Приклади: розпізнавання образів і визначення хімічної структури речовини;

– *прогнозуючі системи* – призначені для логічного аналізу можливих наслідків заданих ситуацій або подій. Приклади: прогнозування погоди і ситуацій на фінансових ринках;

– *діагностичні системи* – призначені для виявлення джерел несправностей за результатами спостережень за поведінкою контролюваної системи (технічної або біологічної). У цю категорію входить широкий спектр задач у всіляких предметних областях – медицині, механіці, електроніці і т. д.;

– *системи проектування* – призначені для структурного синтезу конфігурації об'єктів (компонентів проектированої системи) при заданих обмеженнях. Приклади: синтез електронних схем, компонування архітектурних планів, оптимальне розміщення об'єктів в обмеженому просторі;

– *системи планування* – призначені для підготовки планів проведення послідовності операцій, що призводить до заданої мети. Приклади: задачі планування поведінки роботів і складання маршрутів пересування транспорту;

– *системи моніторингу* – аналізують поведінку контролюваної системи і, порівнюючи отримані дані з критичними точками заздалегідь складеного плану, прогнозують імовірність досягнення поставленої мети. Приклади: контроль руху повітряного транспорту і спостереження за станом енергетичних об'єктів;

- *налагоджуvalні системи* – призначені для вироблення рекомендацій з усунення несправностей у контролюваній системі. До цього класу відносяться системи, що допомагають програмістам у налагодженні програмного забезпечення, і консультуючі системи;
- *системи надання допомоги при ремонті устаткування* – виконують планування процесу усунення несправностей у складних об’єктах, наприклад, у мережах інженерних комунікацій;
- *навчальні системи* – проводять аналіз знань студентів за визначенім предметом, відшукують пробіли в знаннях і пропонують засоби для їхньої ліквідації;
- *системи контролю* – забезпечують адаптивне керування поведінкою складних людино-машинних систем, прогнозуючи появу можливих збоїв і плануючи дії, необхідні для їхнього попередження. Приклади: керування повітряним транспортом, воєнними діями і діловою активністю в сфері бізнесу.

4. За ступенем складності структури:

- *поверхневі системи* – подають знання про область експертизи у вигляді правил (умова → дія). Умова кожного правила визначає зразок деякої ситуації, при дотриманні якої правило може бути виконано. Пошук рішення полягає у виконанні тих правил, зразки яких зіставляються з поточними даними. При цьому передбачається, що в процесі пошуку рішення послідовність формованих у такий спосіб ситуацій не обірветься до одержання рішення, тобто не виникне невідомої ситуації, що не зіставиться з жодним правилом;
- *глибинні системи* – крім можливостей поверхневих систем, мають здатність при виникненні невідомої ситуації визначати за допомогою деяких загальних принципів, справедливих для області експертизи, які дії варто виконати.

5. За типом використовуваних методів і знань:

- *традиційні системи* – використовують в основному неформалізовані методи інженерії знань і неформалізовані знання, отримані від експертів;
- *гіbridні системи* – використовують методи інженерії знань і формалізовані методи, а також дані традиційного програмування та математики.

6. За видами використовуваних даних і знань: з детермінованими і невизначеними знаннями. Під невизначеністю знань і даних розуміються їхня неповнота, ненадійність, нечіткість.

7. За способом формування рішення:

– *аналізуючі системи* – вибір рішення здійснюється з мно-
жини відомих рішень на основі аналізу знань;

– *синтезуючі системи* – рішення синтезується з окремих фраг-
ментів знань.

8. За способом урахування часової ознаки:

– *статичні системи* – призначенні для вирішення задач з не-
змінними в процесі рішення даними і знаннями;

– *динамічні системи* – допускають зміни даних і знань у
процесі рішення.

9. За рівнем складності:

– *прості системи*: поверхневі, традиційні (рідше гібридні)
системи, виконані на персональних ЕОМ, з комерційною вартіс-
тю від 100 до 25 тисяч доларів, з вартістю розробки від 50 до 300
тисяч доларів, з часом розробки від 3 міс. до одного року, що міс-
тять від 200 до 1000 правил;

– *складні системи*: глибинні, гібридні системи, виконані або
на символічних ЕОМ, або на потужній універсальній ЕОМ, або на
інтелектуальній робочій станції, з комерційною вартістю від 50
тисяч до 1 мільйона доларів, із середньою вартістю розробки 5–
10 мільйонів доларів, часом розробки від 1 до 5 років, що містять
від 1,5 до 10 тисяч правил.

10. За стадією існування (ступенем пропрацьованності і нала- годженості):

– *демонстраційний прототип* – система, що вирішує частину
необхідних задач, демонструючи життєздатність методу інжене-
рії знань. При наявності розвитих інструментальних засобів для
розробки демонстраційного прототипу потрібно в середньому
приблизно 1–2 міс., а при відсутності – 12–18 міс. Демонстрацій-
ний прототип працює, маючи 50–100 правил;

– *дослідницький прототип* – система, що вирішує всі необ-
хідні задачі, але хитлива в роботі та не є цілком перевіреною. На
доведення системи до стадії дослідницького прототипу йде 3–6
міс. Дослідницький прототип звичайно має 200–500 правил, що
описують проблемну область;

– *діючий прототип* – надійно вирішує всі задачі, але для вирі-
шення складних задач може знадобитися занадто багато часу та (або)

пам'яті. Для доведення системи до стадії діючого прототипу потрібно 6–12 міс., при цьому кількість правил збільшується до 500–1000.

– *система промислової стадії* – забезпечує високу якість вирішення всіх задач при мінімумі часу і пам'яті. Звичайно процес перетворення діючого прототипу в промислову систему полягає в розширенні бази знань до 1000–1500 правил і переписуванні програм з використанням більш ефективних інструментальних засобів. Для доведення системи від початку розробки до стадії промислової системи потрібно 1–1,5 року;

– *комерційна система* – система, придатна не тільки для власного використання, але і для продажу різним споживачам. Для доведення системи до комерційної стадії потрібно 1,5–3 роки та 0,3–5 млн доларів. При цьому в базі знань системи – 1500–3000 правил.

11. За поколінням:

– *системи першого покоління* – статичні поверхневі системи;

– *системи другого покоління* – статичні глибинні системи (іноді до другого покоління відносять також гіbridні системи);

– *системи третього покоління* – динамічні системи, що, як правило, є глибинними і гіbridними.

12. За узагальненим показником – класом:

– *класифікуючі системи* – вирішують задачі розпізнавання ситуацій. Основним методом формування рішень у таких системах є дедуктивне логічне виведення;

– *довизначальні системи* – використовуються для вирішення задач з не цілком визначеними даними і знаннями. У таких системах виникають задачі інтерпретації нечітких знань і вибору альтернативних напрямків пошуку в просторі можливих рішень. Як методи обробки невизначених знань можуть використовуватися байесівський імовірнісний підхід, коефіцієнти впевненості, нечітка логіка;

– *трансформуючі системи* – відносяться до синтезуючих динамічних експертних систем, у яких передбачається повторюване перетворення знань у процесі вирішення задач. У системах даного класу використовуються різні способи обробки знань: генерація і перевірка гіпотез, логіка припущенень і умовчань (коли за неповними даними формулюються подання про об'єкти визначеного класу, що згодом адаптуються до конкретних умов ситуацій, що змінюються), використання метазнань (більш загальних закономірностей) для усунення невизначеностей у ситуаціях;

– *мультиагентні системи* – динамічні системи, засновані на інтеграції декількох різномірних джерел знань, що обмінюються між собою одержуваними результатами в ході вирішення задач. Системи даного класу мають можливості реалізації альтернативних міркувань на основі використання різних джерел знань і механізму усунення протиріч, розподіленого вирішення проблем, що декомпозуються на паралельно розв’язувані підзадачі із самостійними джерелами знань, застосування різних стратегій виведення рішень у залежності від типу розв’язуваної проблеми, обробки великих масивів інформації з баз даних, використання математичних моделей і зовнішніх процедур для імітації розвитку ситуацій.

Розробка й експлуатація експертної системи передбачає участь таких фахівців (склад і взаємодію учасників побудови й експлуатації експертних систем зображенено на рис. 4.1).

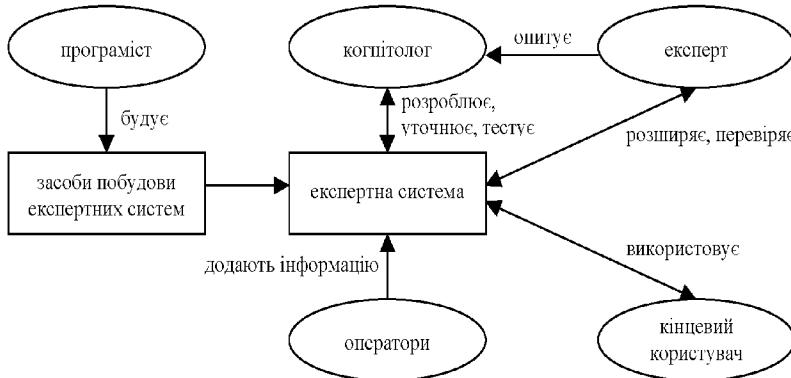


Рисунок 4.1 – Взаємозв’язки основних учасників побудови та експлуатації експертних систем

– *Експерт* – високоекспертний фахівець у проблемній області, задачі якої повинна вирішувати експертна система, який визначає знання, що характеризують проблемну область, забезпечує повноту і правильність введених в систему знань.

– *Інженер зі знань (когнітолог)* – фахівець з розробки експертних систем, який виступає в ролі проміжного буфера між експертом і базою знань. Він допомагає експерту виявляти і структурувати знання, необхідні для роботи експертної системи, здійснює вибір того інструментального засобу, що найбільше підходить для даної

проблемної області, визначає спосіб подання знань у цьому інструментальному засобі, виділяє і програмує (традиційними засобами) стандартні функції (типові для даної проблемної області), що будуть використовуватися в знаннях, які вводяться експертом.

– *Програміст* – фахівець з розробки інструментальних засобів, що здійснює їхнє створення і сполучення з тим середовищем, у якому вони будуть використовуватися.

– *Користувач* – людина, що використовує вже побудовану експертну систему. Термін користувач є трохи неоднозначним. Звичайно він позначає *кінцевого користувача*. Однак користувачем може бути програміст, що відлагоджує засіб побудови експертної системи, інженер зі знань, що уточнює існуючі в системі знання, експерт, що додає в систему нові знання, оператор, що заносить у систему поточну інформацію.

Засіб побудови експертної системи – це програмний засіб, використовуваний інженером зі знань або програмістом для побудови експертної системи. Цей інструмент відрізняється від звичайних мов програмування тим, що забезпечує зручні способи подання складних високорівневих понять. Важливо розрізняти інструмент, що використовується для побудови експертної системи, і саму експертну систему. Інструмент побудови включає як мову, використовувану для доступу до знань, що містяться в системі, і їхнього подання, так і підтримуючі засоби – програми, що допомагають користувачам взаємодіяти з компонентом експертної системи, яка вирішує проблему.

Концепція «швидкого прототипу» використовується для розробки експертних систем і полягає в тому, що розробники не намагаються відразу створити кінцевий продукт. На початковому етапі вони створюють прототип (прототипи) експертної системи, що повинний задовольняти двом суперечливим вимогам: з одного боку, вирішувати типові задачі конкретного застосування, а з іншого боку – час і трудомісткість його розробки повинні бути дуже незначними, щоб можна було максимально запаралеліти процес накопичення і відлагодження знань (здійснюваний експертом) із процесом вибору (розробки) програмних засобів (здійснюваним інженером зі знань і програмістом). Для задоволення зазначених вимог при створенні прототипу, як правило, використовуються різноманітні інструментальні засоби, що прискорюють процес проектування.

Прототип повинний продемонструвати придатність методів інженерії знань для даного застосування. У випадку успіху експерт за допомогою інженера зі знань розширює знання прототипу про проблемну область. При невдачі може знадобитися розробка нового прототипу або розробники можуть прийти до висновку про непридатність методів інженерії знань для даного застосування. В міру збільшення знань прототип може досягти такого стану, коли він успішно вирішує всі задачі даного застосування. Перетворення прототипу в кінцевий продукт звичайно приводить до перепрограмування експертної системи на мовах низького рівня, що забезпечують як підвищення її швидкодії, так і зменшення необхідної пам'яті.

Методологія розробки експертних систем включає такі етапи (див. рис. 4.2).

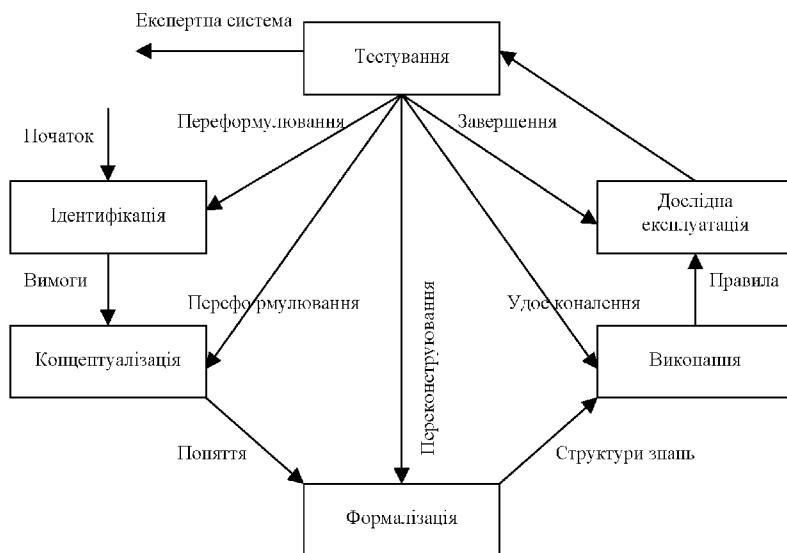


Рисунок 4.2 – Етапи розробки експертних систем

1. *Етап ідентифікації*: визначаються задачі, що підлягають вирішенню, складаються їхні неформальні описи (загальні характеристики задачі; підзадачі, що виділяються усередині даної задачі; ключові поняття, характеристики і відношення; вхідні і ви-

хідні дані; можливий вид рішення; знання, релевантні розв'язуваній задачі; ситуації, що перешкоджають вирішенню задачі; приклади рішення задачі; критерії оцінки якості рішень), виявляються цілі розробки (формалізація неформальних знань експертів; поліпшення якості рішень, прийнятих експертом; автоматизація рутинних аспектів роботи експерта або користувача; тиражування знань експерта), ресурси (джерела знань, трудомісткість, час розробки, обчислювальні засоби й обсяг фінансування), експерти, інженери зі знань і категорії користувачів, визначаються форми взаємин учасників розробки.

2. *Етап концептуалізації*: експерт та інженер зі знань виконують змістовний аналіз проблемної області, виявляють використовувані поняття і їхні взаємозв'язки (типи використовуваних відношень: ієрархія, причина – наслідок, частина – ціле і т. п.), визначають гранулярність (рівень деталізації) знань, визначають особливості задачі (типи доступних даних; вихідні і вихідні дані, підзадачі загальної задачі), визначають методи вирішення задач (використовувані стратегії і гіпотези; процеси, використовувані в ході рішення задачі; типи обмежень, що накладаються на процеси, використовувані в ході рішення; склад знань, використовуваних для рішення задачі; склад знань, використовуваних для пояснення рішення).

3. *Етап формалізації*: визначаються способи подання й інтерпретації усіх видів знань, формалізуються (подаються формальною мовою) основні поняття і відношення, подається структура простору станів і характер методів пошуку в ньому, моделюється робота системи, вибираються програмні засоби розробки, оцінюється адекватність цілям і повнота системи зафікованих понять, методів рішення, засобів подання і маніпулювання знаннями.

4. *Етап виконання* (реалізація): здійснюється набуття знань системою, яке розділяють на витяг знань з експерта, організацію знань, що забезпечує ефективну роботу системи, і подання знань у вигляді, зрозумілому експертній системі. Мета цього етапу – створення одного або декількох прототипів, що вирішують необхідні задачі.

5. *Етап тестування*: експерт (та інженер зі знань) в інтерактивному режимі, використовуючи діалогові та пояснювальні засоби, перевіряє компетентність експертної системи на великій кількості репрезентативних задач. Процес тестування продовжується доти,

поки експерт не вирішить, що система досягла необхідного рівня компетентності.

6. *Етап дослідної експлуатації*: перевіряється придатність експертної системи для кінцевих користувачів, яка визначається в основному зручністю роботи із системою та її корисністю. За результатами цього етапу може знадобитися істотна модифікація експертної системи. Після успішного завершення етапу дослідної експлуатації і використання різними користувачами експертна система може класифікуватися як комерційна.

Модифікація експертної системи здійснюється майже постійно в ході її створення. Виділяють такі види модифікації системи:

– *Удосконалення прототипу* – здійснюється в процесі циклічного проходження через етапи виконання і тестування для налагодження правил і процедур виведення. Цикли повторюються доти, поки система не буде поводитися очікуваним чином. Зміни, здійснювані при удосконаленні, залежать від обраного способу подання і класу задач, розв'язуваних системою. Якщо в процесі удосконалення бажане поводження не досягається, то необхідно здійснити більш серйозні модифікації архітектури системи і бази знань;

– *Переконструювання подання* – перегляд обраного раніше способу подання знань, здійснюваний у результаті повернення від етапу тестування до етапу формалізації;

– *Переформулювання понять*, використовуваних у системі – проектування всієї системи практично заново, здійснюване в результаті повернення на етапи концептуалізації й ідентифікації після невдачі на етапі тестування.

Структура експертної системи (рис. 4.3) містить такі основні компоненти, як: машина логічного виведення (вирішувач, інтерпретатор правил), база знань, підсистема набуття знань, підсистема пояснення рішень, інтерфейсна підсистема (діалоговий компонент), робоча пам'ять (база даних).

Такі експертні системи називають статичними експертними системами – вони використовуються в тих застосуваннях, де можна не враховувати зміни навколошнього світу за час рішення задачі.

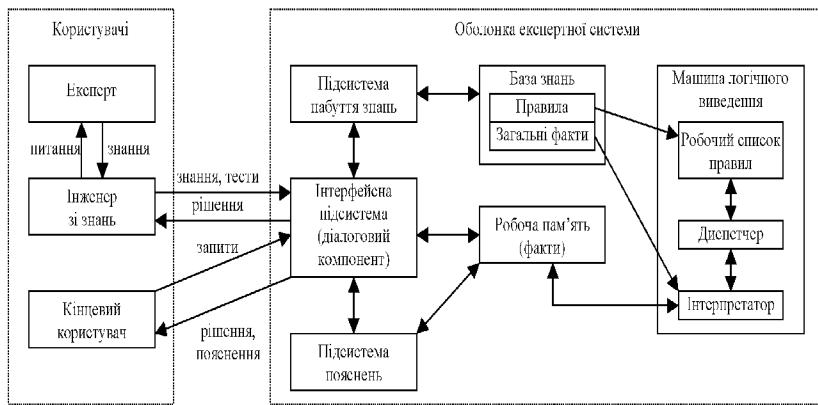


Рисунок 4.3 – Схема ідеальної статичної експертної системи

База знань – частина експертної системи, що містить подання знань, яке стосується визначененої предметної області. У статичній частині бази знань зберігаються довгострокові знання, що описують розглянуту предметну область у вигляді загальних *фактів* (фраз без умов, що містять твердження, які завжди є абсолютно вірними) і *правил* (тверджень, істинність яких залежить від деяких умов, що утворюють тіло правила), які описують доцільні перетворення фактів цієї області з метою породження нових фактів або гіпотез.

Робоча пам’ять (база даних) – динамічна частина бази знань, що змінює свій стан під впливом правил, призначена для збереження вихідних даних (фактів, що описують поточну ситуацію) і проміжних даних розв’язуваної в поточний момент задачі (фактів, що були встановлені до визначеного моменту в результаті виведення, яке полягає в застосуванні правил до наявних фактів).

Машина логічного виведення (механізм логічного виведення, вирішувач) – основна частина експертної системи, яка, використовуючи інформацію з бази знань, на основі стратегії, тісно пов’язаної зі способом подання знань в експертній системі і характером розв’язуваних задач, генерує рекомендації з вирішення задачі та містить:

– *інтерпретатор* – компонент, який, вишиковуючи правила в ланцюжок для досягнення поставленої користувачем мети, послідовно визначає, які правила можуть бути активовані в залеж-

ності від умов, що у них містяться, вибирає одне з застосовних у даній ситуації правил і виконує його;

– *диспетчер* – компонент, що установлює порядок застосування активованих правил;

– *робочий список правил* – створений машиною логічного виведення і впорядкований за пріоритетами список правил, шаблони яких задовільняють фактам або об'єктам, що знаходяться в робочій пам'яті.

Оболонка експертної системи – програма, що забезпечує взаємодію між базою знань та машиною логічного виведення. Кінцевий користувач взаємодіє з оболонкою через інтерфейсну підсистему, передаючи їй запити. Остання активізує машину логічного виведення, яка звертається до бази знань, витягає з неї і генерує в процесі логічного виведення знання, необхідні для відповіді на конкретне питання, і передає сформовану відповідь користувачу або як рішення проблеми, або у формі рекомендації чи поради.

Інтерфейсна підсистема (діалоговий компонент) розподіляє ролі користувачів і експертної системи, а також організує їхню взаємодію в процесі кооперативного вирішення задачі за допомогою перетворення запитів користувачів у внутрішню мову подання знань експертної системи і перетворення повідомлень системи, поданих внутрішньою мовою, у повідомлення мовою, звичною для користувача (обмеженою природною мовою або мовою графіки).

Підсистема набуття знань – автоматизує процес наповнення експертної системи знаннями експертом або інженером зі знань через редактор бази знань без залучення інженера зі знань до вирішення задачі явного кодування знань або автоматично витягає знання з наборів даних у процесі навчання на основі дерев рішень, методів виділення асоціативних правил, штучних нейронних або нейро-нечітких мереж.

Редактор бази знань – складова частина підсистеми набуття знань, що являє собою транслятор з деякої підмножини природної мови, використовуваної інженером зі знань і експертом, у спеціальний код, орієнтований на роботу механізму логічного виведення.

Підсистема пояснень – дозволяє контролювати хід суджень експертної системи і пояснювати її рішення або їхню відсутність з указівкою використаних знань, а також виявляти неодно-

значності і протиріччя в базі знань експертної системи, що погоджує експерту тестування системи і підвищує довіру користувача до отриманого результату, а також дозволяє навчати користувача рішенню відповідних задач.

Динамічні експертні системи – більш високий клас програмних засобів у порівнянні зі статичними експертними системами, що враховують динаміку змін навколошнього світу за час виконання програми. У порівнянні зі статичною експертною системою в динамічну вводяться ще два компоненти: підсистема моделювання зовнішнього світу і підсистема сполучення з зовнішнім світом (рис. 4.4).

Динамічна експертна система здійснює зв'язки з зовнішнім світом через систему контролерів і датчиків. Крім того компоненти бази знань і механізму виведення істотно змінюються, щоб відбити тимчасову логіку подій, які відбуваються в реальному світі.

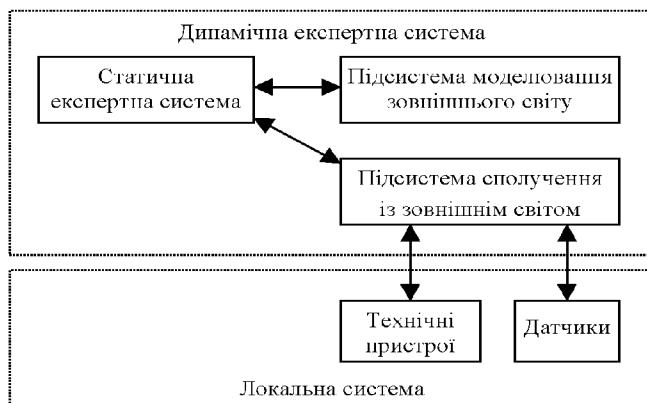


Рисунок 4.4 – Динамічна експертна система

Функціонування експертної системи здійснюється в одному з режимів роботи експертної системи.

– **Режим набуття знань:** спілкування з експертною системою здійснює експерт за посередництвом інженера зі знань. Експерт описує проблемну область у вигляді сукупності фактів і правил. Факти визначають об'єкти, їхні характеристики і значення, що існують в області експертизи. Правила визначають способи маніпулювання даними, характерні для розглянутої проблем-

ної області. Експерт, використовуючи компонент набуття знань, наповнює систему знаннями, що дозволяють експертній системі в режимі рішення самостійно (без експерта) вирішувати задачі з проблемної області.

Важливу роль у режимі набуття знань грає пояснювальний компонент. Саме завдяки йому експерт на етапі тестування локалізує причини невдалої роботи експертної системи, що дозволяє експерту цілеспрямовано модифікувати старі або вводити нові знання. Звичайно пояснювальний компонент повідомляє таке: як правила використовують інформацію користувача; чому використовувалися або не використовувалися дані чи правила; які були зроблені висновки і т. п. Усі пояснення робляться, як правило, обмеженою природною мовою або мовою графіки.

– *Режим консультації* (рішення): спілкування з експертною системою здійснює кінцевий користувач, якого цікавить результат та (або) спосіб одержання рішення. Користувач у залежності від призначення експертної системи може не бути фахівцем у даній проблемній області, у цьому випадку він звертається до системи за порадою, не вміючи одержати відповідь самостійно, або бути фахівцем, у цьому випадку він звертається до системи, щоб або прискорити процес одержання результату, або покласти на систему рутинну роботу. У режимі консультації дані про задачу користувача обробляються діалоговим компонентом. Після обробки дані надходять у робочу пам'ять. На основі вхідних даних з робочої пам'яті, загальних даних про проблемну область і правил з бази знань вирішувач (інтерпретатор) формує рішення задачі. На відміну від традиційних програм експертна система в режимі рішення задачі не тільки виконує запропоновану послідовність операцій, але і попередньо формує її. Якщо відповідь системи є незрозумілою користувачу, то він може зажадати пояснення того, як відповідь отримана.

Переваги і недоліки експертних систем.

Перевагами експертних систем є:

– сталість: знання експертної системи зберігаються протягом невизначено довгого часу і нікуди не зникають, у той час як людська компетенція слабшає із часом, перерва у діяльності людини-експерта може серйозно відбитися на її професійних якостях, крім того експерти-люди можуть піти на пенсію, звільнитися з роботи або вмерти, тобто їхні знання можуть бути втрачені;

- легкість передачі або відтворення: передача знань від однієї людини до іншої – довгий і дорогий процес, передача штучної інформації – це простий процес копіювання програми або файлу даних;
- підвищена доступність: експертна система – засіб масового виробництва експертних знань, що дозволяє багатьом користувачам одержати доступ до експертних знань;
- можливість одержання й об'єднання експертних знань з багатьох джерел: за допомогою експертних систем можуть бути зібрані знання багатьох експертів і притягнуті до роботи над задачею, виконуваної одночасно і безупинно у будь-яку годину дня і ночі; рівень експертних знань, скомбінованих шляхом об'єднання знань декількох експертів, може перевищувати рівень знань окремо узятого експерта-людини;
- стійкість і відтворюваність результатів: експерт-людина може приймати в тотожних ситуаціях різні рішення через емоційні фактори або утому, у той час, як результати експертних систем стабільні і являють собою незмінно правильні, позбавлені емоцій і повні відповіді за будь-яких обставин;
- низька вартість: експерти, особливо висококваліфіковані, обходяться дуже дорого, у той час, як експертні системи, навпаки, є порівняно недорогими – їхня розробка є дорогою, але вони є дешевими в експлуатації: вартість надання експертних знань у розрахунку на окремого користувача істотно знижується;
- зменшена небезпека: експертні системи можуть використовуватися в таких варіантах середовища, що можуть виявитися небезпечними для людини;
- швидкий відгук: експертна система може реагувати швидше і бути більш готовою до роботи, ніж експерт-людина, особливо в деяких екстремальних ситуаціях, де може знадобитися більш швидка реакція, ніж у людини;
- підвищена надійність: застосування експертних систем дозволяє підвищити ступінь довіри до того, що прийнято правильне рішення, шляхом надання ще однієї обґрунтованої думки людині-посереднику за наявності неузгоджених думок між декількома експертами-людьми;
- можливість пояснення рішень: експертна система здатна докладно пояснити свої рішення, що привели до визначеного висновку, а лю-

дина може виявитися занадто втомленою, не скильною до пояснень або нездатною робити це постійно;

– можливість застосування в якості інтелектуальної навчальної програми: експертна система може діяти як інтелектуальна навчальна програма, передаючи учню на виконання приклади програм і пояснюючи, на чому засновані судження системи;

– можливість застосування у якості інтелектуальної бази даних: експертні системи можуть використовуватися для доступу до баз даних за допомогою інтелектуального способу доступу;

– формалізація і перевірка знань: у процесі розробки експертної системи знання експертів-людей перетворяться в явну форму для введення в комп'ютер, у результаті чого вони стають явно відомими і з'являється можливість перевіряти знання на правильність, несуперечність і повноту.

Недоліки експертних систем:

– експертні системи погано вміють: подавати знання про часові та просторові відношення, розмірковувати, виходячи зі здорового глузду, розпізнавати межі своєї компетентності, працювати із суперечливими знаннями;

– інструментальні засоби побудови експертних систем погано вміють: виконувати набуття знань, уточнювати бази знань, працювати зі змішаними схемами подання знань;

– побудова експертних систем не під силу кінцевому користувачу, який не володіє експертними знаннями про проблемну область;

– необхідність залучення людини-експерта з проблемної області, що є носієм знань; неможливість повного відмовлення від експерта-людини;

– можливі труднощі взаємодії експерта зі спеціалістом-когнітологом, який шляхом діалогу з експертом оформляє отримані від експерта знання в обраному формалізмі подання знань;

– необхідність повної переробки програмного інструментарію, у випадку, якщо наявна оболонка експертної системи та / або використовувана нею модель подання знань погано підходять для обраної проблемної області, задачі;

– тривалість процесів витягу знань з експерта, їхньої формалізації, перевірки на несуперечність і усунення протиріч.

Експертні системи залучають значні грошові інвестиції і людські зусилля. Спроби вирішити занадто складну, малозрозумілу чи, іншими словами, не відповідну наявній технології проблему можуть привести до дорогих і ганебних невдач.

Використання експертної системи є доцільним, якщо розробка експертної системи можлива, виправдана і доречна.

Розробка експертної системи можлива, якщо: задача не вимагає загальнодоступних знань, задача вимагає тільки інтелектуальних навичок, експерти можуть описати свої методи природною мовою, існують справжні експерти, експерти одностайні щодо рішень, задача не є занадто важкою, задача є цілком зрозумілою, проблемна область є добре структурованою і не вимагає розмірковувань на основі здорового глузду – широкого спектру загальних відомостей про світ та спосіб його функціонування, котрі знає та вміє використовувати будь-яка нормальна людина.

Розробка експертної системи виправдана, хоча б в одному з випадків, якщо: одержання рішення є високорентабельним, втрачається людський досвід, експертів мало, досвід потрібний одночасно у багатьох місцях, досвід необхідно застосовувати у ворожих людині умовах, людський досвід відсутній у ситуаціях, де він є необхідним.

Застосування експертної системи розумно, якщо: задача вимагає оперування символами, задача не може бути вирішена традиційними обчислювальними методами і вимагає евристичних рішень, задача не є занадто простою, задача становить практичний інтерес, задача має розміри, що допускають реалізацію.

4.4 Логічне виведення

Логічне виведення – процес одержання з вихідних фактів за заданими правилами нових фактів, що логічно випливають з вихідних.

Типи логічного виведення виділяють такі.

– *Дедукція* – логічний розсуд, у якому висновки повинні випливати з відповідних ним посилок.

– *Абдукція* – метод формування суджень у зворотному напрямку, від істинного висновку до посилок, що могли привести до одержання цього висновку.

– *Індукція* – логічне виведення від окремого випадку до загального – один з основних методів машинного навчання, у якому

комп’ютери навчаються без утручення людини. До цих методів відноситься кластер-аналіз і нейронні та нейро-нечіткі мережі, що самоорганізуються.

– *Інтуїція* – метод, не заснований на перевірений теорії. Відповідь являє собою усього лише припущення, можливо, сформульоване шляхом підсвідомого розпізнавання якогось основного образа. Логічне виведення такого типу ще не реалізовано в експертних системах.

– *Евристика* – емпіричні (отримані на основі досвіду) правила виведення.

– *Метод породження і перевірки* – метод проб і помилок. Часто використовується в сполученні з плануванням для досягнення максимальної ефективності.

– *Судження, застосовувані за замовчуванням* – судження, що допускають можливість під час відсутності конкретних знань приймати за замовчуванням загальноприйняті чи загальновідомі знання.

– *Автоеністемічні судження* – самопізнання, або міркування про те, яким людині уявляється деякий об’єкт, його властивість.

– *Судження за аналогією* – логічне виведення виходячи з наявності ознак, подібних до ознак іншої ситуації.

– *Монотонні судження* – судження, застосовувані в тих умовах, коли раніше отримані знання не можуть виявитися неправильними після одержання нового свідчення.

– *Немонотонні судження* – судження, застосовувані в тих умовах, коли раніше отримані знання можуть виявитися неправильними після одержання нового свідчення.

Якщо факти оброблюють, розглядаються тільки як істинні, або хибні, то говорять про чітку логіку і, відповідно, чітке виведення.

Якщо факти і правила, що їх оброблюють, розглядаються як істинні або хибні лише у певній мірі (з певною імовірністю, впевненістю), то говорять про нечітку логіку і, відповідно, нечітке виведення.

4.5 Чітке логічне виведення

Чітке логічне виведення – процес отримання рішень на основі звичайної (булевої, чіткої) логіки.

Звичайне (булеве) логічне виведення висновків на основі відомих фактів і правил широко використовується в експертних системах і базується на таких *тавтологіях*:

– *модус поненс* (modus ponens): $(A \wedge (A \rightarrow B)) \rightarrow B$. Модус поненс виводить висновок « B є істинно», якщо відомо, що « A є істинно» й існує правило «Якщо A , то B », де A та B – чіткі логічні твердження.

– *модус толленс* (modus tollens): $((\neg B) \wedge (A \rightarrow B)) \rightarrow \neg A$. Модус толленс виводить висновок « A є хибним», якщо відомо, що « B є хибним» та існує правило «Якщо A , то B ».

– *силогізм* (syllogism): $((A \rightarrow B) \wedge (B \rightarrow C)) \rightarrow (A \rightarrow C)$. Силогізм виводить висновок «Якщо A , то C » (« $\exists A$ випливає C »), якщо відомо, що «Якщо A , то B » (« $\exists A$ випливає B ») і «Якщо B , то C » (« $\exists B$ випливає C »).

– *контрапозиція* (contraposition): $(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$. Контрапозиція виводить висновок «Якщо B є хибним, то A є хибним», якщо відомо, що «Якщо A , то B ».

Дедуктивне логічне виведення – прямий метод виведення висновків (метод висхідного виведення, метод прямого ланцюжка суджень, classic forward-chaining reasoning), заснований на використанні правила виведення модус поненс, що дозволяє визначити істинність висновку правила при відомій істинності його умови. Прямий метод виведення реалізується за допомогою перетворення окремих фактів проблемної області в конкретні значення істинності умов правил. Після цього перетворення ті з правил, для яких стають істинними відповідні умови, генерують висновки своїх правильних частин. Ці висновки приймаються як істинні і стають новими фактами, що можуть бути використані як умови в розглянутій базі правил. При цьому правила, для яких є істинними умови, називають *активними*.

Процес виведення прямим методом має рекурсивний характер і може бути зупинений або у випадку відсутності нових активних правил, або у випадку одержання висновку, що є цільовим у контексті рішення вихідної проблеми. Подібне підтвердження цільового висновку характеризує успіх процесу виведення, оскільки тільки в цьому випадку використання системи правил характеризує рішення поставленої проблеми.

Ланцюжок прямого виведення висновків (forward chaining) – шлях суджень, що будеться, відштовхуючись від фактів (умов,

про які відомо, що вони задовольняються), до гіпотез (стану проблеми, що випливає з цих умов).

Абдуктивне логічне виведення є необґрунтованим правилом виведення $((A \rightarrow B) \wedge B) \rightarrow A$ і означає, що висновок не є обов'язково істинним для кожної інтерпретації, при якій істинні передумови. Абдуктивні судження часто називають найкращим поясненням даних B .

Абдукція заснована на зворотному виведенні висновків (метод спадного виведення, classic backward-chaining reasoning), що заснований на використанні правила модус толенс, що дозволяє визначити хибність умови правила при відомій хибності його висновку.

Зворотний метод виведення в експертних системах реалізується в модифікованому вигляді за допомогою дослідження можливості застосування правил для підтвердження деяких заздалегідь заданих висновків: заперечення висновку заміняється питанням про його істинність. Символічно це записується у вигляді: $(B? \wedge (A \rightarrow B)) \rightarrow A?$, що означає, що у випадку істинності імплікації $A \rightarrow B$ достатньою умовою істинності формули B є істинність формули A . Таким чином, якщо метою виведення є доказ істинності висновку B , то для цього досить довести істинність умови A , розглянутої як підціль. Тому зворотний метод служить обґрунтуванням достатніх умов для істинності висновків правил.

Процес зворотного виведення починається з підстановки окремих цікавлячих нас висновків у праві частини відповідних правил, які у цьому випадку стають активними. Після аналізу кожного з активних правил фіксуються умови, що підтверджують ці правила. Ці умови приймаються як істинні і стають новими фактами, що можуть бути використані в якості нових цільових висновків у розглянутій базі правил. Процес виведення зворотним методом має рекурсивний характер і може бути зупинений або у випадку відсутності нових активних правил, або у випадку одержання підтвердження умов, що є істинними чи відомими фактами проблемної області.

Ланцюжок зворотного виведення (backward chaining) – шлях суджень, що будується, відштовхуючись від заданої цілі (гіпотез, що подають цільовий стан системи) до умов, при яких можливе досягнення цієї цілі (до фактів).

Традиційна логіка має обмеження на використання в умовах

неповної і невизначеної інформації є монотонною.

Монотонна логіка заснована на множині аксіом, прийнятих за істинні, з яких виводяться наслідки. При цьому додавання в систему нової інформації може викликати тільки збільшення множини істинних тверджень, що приводить до проблем при моделюванні суджень, заснованих на довірі і припущеннях.

Немонотонна логіка вирішує проблему моделювання суджень, заснованих на довірі і припущеннях за рахунок того, що, на відміну від математичних аксіом, міра довіри і висновки можуть мінятися в міру накопичення інформації.

Система немонотонних суджень керує ступенем невизначеності, роблячи найбільш визначені припущення в умовах невизначеності інформації. Потім виконується виведення на основі цих припущень, прийнятих за істинні. Пізніше міра довіри може змінитися і зажадати повторного перегляду усіх висновків, виведених з її використанням.

До методів абдуктивного виведення відносять стенфордську теорію коефіцієнта впевненості, нечітку логіку, теорію Демпстера-Шафера, метод Байеса, метод Нейлора.

Стенфордська теорія коефіцієнта впевненості заснована на неформальних оцінках фактів і висновків.

Коефіцієнт упевненості (довіри, вірогідності, certainty factor) – це число, що означає імовірність або ступінь упевненості, з яким можна вважати даний факт або правило достовірним або справедливим.

Для визначення коефіцієнта упевненості використовуються методи математичної статистики, а також суб'єктивні оцінки експерта.

Коефіцієнт упевненості антецедента (умови правила) $CF(X, Y)$, що містить посилки X та Y , обчислюється за формулою:

$$CF(X, Y) = \begin{cases} 1, & CF(X) = 1 \text{ або } CF(Y) = 1; \\ CF(X) + CF(Y) - CF(X)CF(Y), & CF(X) > 0, CF(Y) > 0; \\ \frac{CF(X) + CF(Y)}{1 - \min\{|CF(X)|, |CF(Y)|\}}, & CF(X)CF(Y) \leq 0, CF(X) \neq \pm 1, CF(Y) \neq \pm 1; \\ CF(X) + CF(Y) + CF(X)CF(Y), & CF(X) < 0, CF(Y) < 0; \\ -1, & CF(X) = -1 \text{ або } CF(Y) = -1, \end{cases}$$

де $CF(X)$ – коефіцієнт упевненості посилки X , $CF(Y)$ – коефіцієнт упевненості посилки Y .

Коефіцієнт упевненості консеквента $CF(C)$, визначається на основі коефіцієнта впевненості антецедента $CF(X, Y)$ і коефіцієнта впевненості правила $CF(R)$ як $CF(C) = CF(X, Y) CF(R)$.

Коефіцієнт впевненості приймає значення в діапазоні $[-1, +1]$. Якщо він дорівнює $+1$, то це означає, що при дотриманні всіх обговорених умов укладач правила абсолютно упевнений у правильності висновку, а якщо він дорівнює -1 , то виходить, що при дотриманні всіх обговорених умов існує абсолютно впевненість у помилковості цього висновку. Відмінні від $+1$ позитивні значення коефіцієнта вказують на ступінь впевненості в правильності висновку, а негативні значення – на ступінь впевненості в його помилковості.

Коефіцієнт упевненості є комбінованою оцінкою. Його основне призначення полягає в тому, щоб керувати ходом виконання програми при формуванні суджень, керувати процесом пошуку мети в просторі станів (якщо коефіцієнт упевненості гіпотези виявляється в діапазоні $[-0,2, +0,2]$, то пошук блокується), ранжирувати набір гіпотез після обробки всіх ознак.

Якщо обидві гіпотези підтверджують висновок або, навпаки, обидві гіпотези його спростовують, то коефіцієнт упевненості їхньої комбінації зростає за абсолютною величиною. Якщо ж одна гіпотеза підтверджує висновок, а інша його спростовує, то наявність знаменника у відповідному виразі згладжує цей ефект. Якщо виявилось, що гіпотез декілька, то їх можна по черзі пропускати через цю формулу, причому, оскільки вона має властивість комутативності, то порядок, у якому обробляються гіпотези, значення не має.

Незважаючи на відсутність строгого теоретичного обґрунтування, коефіцієнти впевненості знаходять широке застосування в експертних системах продукційного типу завдяки простоті сприйняття їх інтерпретації одержуваних результатів, які непогано узгоджуються з реальністю.

Теорія Демпстера–Шафера розроблена з метою узагальнення імовірнісного підходу до опису невизначеності та пов’язана зі спробою звільнитися від догматів аксіом теорії ймовірностей при описі суб’ективної віри людей.

Розглянемо фрейм розрізнення – кінцеву множину можливостей θ , які взаємно виключають одна одну.

На множині всіх підмножин θ як на множині елементарних подій задамо базисний розподіл ймовірностей m , визначений на множині 2^θ значень з інтервалу $[0, 1]$, такий, що: $m(\emptyset) = 0$ та

$$\sum_{A_i \subseteq \theta} m(A_i) = 1,$$

де $m(A_i)$ – міра довіри, приписана гіпотезі A_i .

Міра загальної довіри, приписана A , визначається співвідношенням:

$$Bel(A) = \sum_{B \subseteq A} m(B),$$

де $Bel(A) \in [0, 1]$ – функція довіри, що характеризує віру суб'єкта в істинність події A , називана нижньою імовірністю.

Міра правдоподібності події A , називана верхньою імовірністю, визначається як: $Pl(A)=1-Bel(\text{not}(A))$.

Теорія Демпстера-Шафера заснована на двох ідеях. Перша – одержання ступеня довіри для даної задачі із суб'єктивних свідчень про пов'язані з нею проблеми та друга – використання правила поєднання свідчень, якщо вони засновані на незалежних спостереженнях.

Правила Демпстера дозволяють обчислити нове значення функції довіри за двома її значеннями, що базуються на різних спостереженнях.

Метою правила є приписати деяку міру довіри m різним підмножинам A множини θ ; m іноді називають імовірнісною функцією чутливості (probability density function) підмножини θ .

Реально свідчення підтримують не всі елементи θ . В основному підтримуються різні підмножини A множини θ . Більш того, оскільки елементи θ передбачаються як взаємовиключні, доказ на користь одного з них може впливати на довіру іншим елементам. У системі Демпстера-Шафера ці взаємодії враховуються прямо шляхом безпосереднього маніпулювання множинами гіпотез. Величина $m_n(A)$ означає ступінь довіри, пов'язаний з підмножиною гіпотез A , а n подає число джерел свідчень.

Правило Демпстера має вигляд:

$$m_n(A) = \frac{\sum_{\substack{X \cap Y = A}} m_{n-2}(X)m_{n-1}(Y)}{1 - \sum_{\substack{X \cap Y = \emptyset}} m_{n-2}(X)m_{n-1}(Y)}.$$

X та Y поширюються на всі підмножини в Θ , перетинанням яких є A . Якщо в таблиці перетинань буде виявлено порожній елемент, то виконується нормалізація: визначається значення k як сума всіх ненульових значень, присвоєних у множині Θ , потім установлюють $m_n(\emptyset)=0$, а значення m_n для всіх інших множин гіпотез поділяються на $(1-k)$.

Метод виведення Байеса заснований на припущення про наявність практично для будь-якої події апріорних ймовірностей того, що дана подія є істинною. Задача полягає в зміні імовірнісних оцінок цієї події з появою інформації про настання деякої іншої події, називаних апостеріорними ймовірностями.

Апріорна імовірність події (безумовна імовірність, prior probability) – це імовірність, привласнена події при відсутності знання, що підтримує її настання, тобто це імовірність події, що передує якій-небудь основі. Апріорна імовірність позначається $P(\text{подія})$.

Апостеріорна імовірність події (умовна імовірність, posterior probability) – це імовірність події при деякій заданій основі. Вона позначається $P(\text{подія} | \text{основа})$.

Правило добутку пов'язує апостеріорні й апріорні імовірності:

$$P(A \wedge B) = P(A|B) P(B) = P(B|A) P(A).$$

Правило Байеса дозволяє обчислювати невідомі імовірності з відомих умовних ймовірностей, засновано на правилі добутку і має вигляд:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \text{ або } P(B|A) = \frac{P(A|B)P(B)}{P(A)}.$$

Теорема Байеса в загальному виді дозволяє визначити апостеріорну імовірність, залежить від апріорної імовірності і від інформації, що надійшла:

$$P(A_i | B) = \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^N P(B|A_j)P(A_j)}, i = 1, 2, \dots, N,$$

де $\{A_i\}$ – послідовність непересічних подій (гіпотез), B – довільна подія (симптом), для якої $P(B) > 0$, N – число можливих гіпотез, $P(A_i|B)$ – апостеріорна імовірність гіпотези A_i за наявності симптуму B ; $P(B|A_i)$ – імовірність появи симптуму B за наявності гіпотези A_i ; $P(A_i)$ –aprіорна імовірність гіпотези A_i . Імовірності $P(B|A_i)$ та $P(A_i)$, $i = 1, \dots, N$, задаються експертом і не змінюються в процесі вирішення задачі.

Метод виведення Нейлора полягає в приписуванні кожному симптуму ціни, що відбиває роль у процесі виведення, і задавання в першу чергу того питання, для якого ціна виявляється найбільшою.

Судження проводяться за такою послідовністю кроків.

Крок 1. Оцінити aprіорні імовірності $P(A_i)$ для всіх гіпотез.

Крок 2. Обчислити ціни симптомів.

Ціна симптуму визначається за формулою:

$$C(B_j) = \sum_{i=1}^N |P(A_i | B_j) - P(A_i | \neg B_j)|,$$

$$P(A_i | B_j) = \frac{P(B_j | A_i)P(A_i)}{P(B_j)}, \quad P(A_i | \neg B_j) = \frac{1 - P(B_j | A_i)P(A_i)}{1 - P(B_j)},$$

$$P(A_i) = P(B_j | A_i) P(A_i) + P(\neg B_j | \neg A_i) P(\neg A_i), \quad P(\neg B_j) = 1 - P(B_j),$$

де $C(B_j)$ – ціна симптуму B_j – сума максимальних змін ймовірностей подій, що можуть відбутися у всіх N гіпотезах, до яких цей симптом може бути застосовним.

Крок 3. Знайти симптом з максимальною ціною:

$$B_m: m = \arg \max_{j=1,2,\dots,N} C(B_j).$$

Крок 4. Поставити запитання користувачу для симптуму B_m і одержати на нього відповідь R_m , $R_m \in \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$, де -5 відповідає «ні», 0 – «не знаю», а 5 – «так».

Крок 5. Обчислити апостеріорні імовірності для актуальних гіпотез:

$$P(A_i | R_m) = \begin{cases} 0,2R_m(P(A_i | B_m) - P(A_i)) + P(A_i), & 0 \leq R_m \leq 5; \\ -0,2R_m(P(A_i | \neg B_m) - P(A_i)) + P(A_i), & -5 \leq R_m \leq 5. \end{cases}$$

Установити: $P(A_i) = P(A_i | R_m)$.

Крок 6. Перерахувати $C(B_j)$ для нових значень $P(A_i)$.

Крок 7. Для кожної гіпотези A_i знайти значення поточної мінімальної імовірності гіпотези $P_{\min}(A_i)$ і поточної максимальної імовірності гіпотези $P_{\max}(A_i)$.

Крок 8. Знайти найбільший з можливих досяжних максимумів імовірностей для всіх гіпотез:

$$PM = \max_{i=1,2,\dots,N} P_{\min}(A_i).$$

Крок 9. Якщо існує такий номер k , для якого $P_{\max}(A_k) > PM$, то перейти до кроку 3; у протилежному випадку – вибрати гіпотезу:

$$A_m : m = \arg \max_{i=1,2,\dots,N} P(A_i),$$

як найбільш імовірний результат.

Крок 10. Видати як результат гіпотези A_m і завершити роботу.

Індуктивне логічне виведення поєднує у собі усі методи машинного навчання за прецедентами, зокрема методи розпізнавання образів. Розглянемо як приклад один з найвідоміших методів індукції.

Метод ДСМ (метод Джона Стюарта Міля) запропонований у середині XIX століття і є методом індуктивного виведення. Способи встановлення причинно-наслідкових відношень, запропоновані Мілем, ґрунтуються на ідеях виявлення подібності та розходження в ситуаціях, що спостерігаються. Здатність уловлювати подібність і виділяти розходження – фундаментальна здатність, властива, очевидно, усім живим істотам. Спираючись на цю здатність, Міль сформулював такі *принципи індукції*:

1. *Принцип одного розходження*: якщо після введення якого-небудь фактора з'являється (чи після його видалення зникає) відоме явище, причому ми не вводимо і не видаляємо ніякої іншої обставини, що могла б мати вплив, то зазначений фактор складає причину явища. Цей принцип можна проілюструвати схемою:

$$A, B, C \rightarrow D,$$

$$A, B, C \rightarrow D,$$

.....

$$A, B, C \rightarrow D,$$

$$B, C \not\rightarrow D,$$

де знак « \rightarrow » трактується як поява D при наявності A, B, C . При достатній кількості експериментів принцип єдиного розходження дозволяє стверджувати, що A є причиною, а D – наслідком.

2. *Принцип єдиної подібності*: якщо всі обставини явища, крім одної, можуть бути відсутні, не знишуючи цим явища, та ця обставина є причиною даного явища. Схема принципу така:

$$A, B, C \rightarrow D,$$

$$A, B, C \rightarrow D,$$

.....

$$A, B \rightarrow D,$$

$$A, C \rightarrow D,$$

.....

$$A \rightarrow D.$$

З цієї схеми випливає, що A та D пов'язані причинно-наслідковим відношенням.

3. *Принцип єдиного залишку*: якщо відняти з якого-небудь явища ту його частину, що є наслідком відомих причин, то залишок явища є наслідком інших причин. Розглянемо схему:

$$A, B, C \rightarrow D, E$$

$$A, B, C \rightarrow D, E$$

.....

$$B, C \rightarrow E.$$

Після того як із прикладів $A, B, C \rightarrow D, E$ було «відняте» причинно-наслідкове відношення $A \rightarrow D$, були отримані спостереження $B, C \rightarrow E$, на підставі яких можна припустити, що B та C є можливими причинами явища E . Для подальшого уточнення потрібно перевірити, чи приводить виключення B до появи E . Якщо так, то причиною явища E слугує C , у противному випадку – B . Можливо також, що явище E обумовлене одночасною наявністю B та C , тобто поява деякого елемента ситуації може визначатися не окремими факторами, а їхньою сукупністю.

Схеми Міля справедливі лише за умови, що в описі ситуації присутня повна множина фактів і явищ, що спостерігаються.

Нехай задана множина причин $A = \{A_1, A_2, \dots, A_p\}$, множина наслідків $B = \{B_1, B_2, \dots, B_m\}$ і множина оцінок $Q = \{q_1, q_2, \dots, q_r\}$.

Вираз виду $A_i \rightarrow B_j$ називається позитивною гіпотезою, що виражає твердження « A_i є причиною B_j , з оцінкою вірогідності q_k ». Негативною гіпотезою називається вираз $A_i \not\rightarrow B_j$, що формулюється « A_i не є причиною B_j з оцінкою вірогідності q_k ». Позитивні гіпотези будемо позначати $h_{i,j,k}^+$, негативні — $h_{i,j,k}^-$.

Серед значень виділимо два спеціальних, котрі можна інтерпретувати як «хибність» (0) та «істина» (1). Гіпотези з цими оцінками можна розглядати як явища, істинність або хибність яких твердо встановлено. Інші значення між 0 та 1 будемо позначати раціональними числами k/n , де $k = 1, \dots, n-1$, а n характеризує число прикладів.

У загальнений ДСМ-метод включає такі кроки.

Крок 1. На основі вихідної множини позитивних і негативних прикладів (спостережень) формується набір гіпотез, що записуються в матриці M^+ та M^- . Гіпотези формуються на основі виявлення подібності і розходження в прикладах. Матриці мають вигляд:

$$M^+ = \begin{array}{c|c|c|c} & B_l & \dots & B_w \\ \hline A_l & h_{l,i,k}^+ & \dots & h_{l,w,m}^+ \\ \dots & \dots & \dots & \dots \\ A_r & h_{r,i,s}^+ & \dots & h_{r,w,d}^+ \end{array}; \quad M^- = \begin{array}{c|c|c|c} & B_j & \dots & B_v \\ \hline A_x & h_{x,j,k}^- & \dots & h_{x,v,t}^- \\ \dots & \dots & \dots & \dots \\ A_z & h_{z,j,m}^- & \dots & h_{z,v,f}^- \end{array}.$$

Крок 2. До вихідної множини прикладів додаються нові спостереження, що можуть або підтверджувати висунуті гіпотези, або спростовувати їх, при цьому оцінки гіпотез змінюються в такий спосіб. Якщо деяка гіпотеза $h_{i,j,k}$ мала оцінку $q_k = k/n$, то з появою нового приклада $(n+1)$ проводиться перевірка на підтвердження цієї гіпотези. У випадку позитивної відповіді оцінка $q_k = (k+1)/(n+1)$, інакше $q_k = (k-1)/(n+1)$. У процесі накопичення інформації оцінки висунутих гіпотез можуть наблизятися до 1 або 0. Зміна оцінок може також мати коливальний характер, що, як правило, веде до виключення таких гіпотез з множин M^+ або M^- .

Крок 3. Циклічне додавання прикладів, що супроводжується зміною оцінок вірогідності гіпотез з періодичною зміною множин M^+ та M^- .

Крок 4. Завершення процесу індуктивного виведення при виконанні умов закінчення циклу. Як такі умови можуть викорис-

товуватися міри близькості значень q_i до 0 або 1, а також додаткові умови, що можуть бути пов'язані з обмеженням часу (кількості нових прикладів) виведення і т. п.

У сучасних модифікаціях ДСМ-методу використовується виведення за аналогією, враховується контекст реалізації причинно-наслідкових відношень, застосовуються нечіткі описи фактів і т. д.

4.6 Нечітке логічне виведення

Нечітким логічним виведенням (fuzzy logic inference) називається апроксимація залежності $y = f(x_1, x_2, \dots, x_n)$ за допомогою нечіткої бази знань і операцій над нечіткими множинами.

Нехай E – універсальна множина, x – елемент E , а G – деяка властивість. Звичайна (*чітка*) *підмножина* A універсальної множини E , елементи якої мають властивість G , визначається як множина впорядкованих пар $\{\langle \mu_A(x) | x \rangle\}$, де $\mu_A(x)$ – характеристична функція належності, що приймає значення 1, якщо x має властивість G , та 0 – у протилежному випадку.

Нечітка підмножина відрізняється від звичайної тим, що для елементів x з E немає однозначної відповіді «ні» або «так» щодо властивості G . У зв'язку з цим *нечітка підмножина* A універсальної множини E визначається як множина впорядкованих пар $A = \{\langle \mu_A(x) | x \rangle\}$, де $\mu_A(x)$ – характеристична функція належності (або просто функція належності), що приймає значення в деякій цілком впорядкованій множині M (наприклад, $M = [0; 1]$).

Функція належності вказує ступінь належності елемента x підмножині A . Множину M називають *множиною належностей*. Якщо $M = \{0, 1\}$, то нечітка підмножина A може розглядатися як чітка множина.

Функції належності нерозривно пов'язані із нечіткими множинами. Тип функції належності в значному ступені визначає властивості нечіткої системи. Задавання функцій належності можна здійснювати у вигляді списку з явним перерахуванням усіх елементів та відповідних їм значень функції належності (наприклад, використовуючи відносні частоти за даними експерименту як значення належності), або аналітично у вигляді формул (наприклад, використовуючи типові форми кривих для задання функцій належності з уточненням їхніх параметрів відповідно до даних експерименту).

Нечітка змінна визначається як $\langle a, E, A \rangle$, де a – найменування змінної, $E = \{x\}$ – область визначення змінної, набір можливих зна-

чень x , $A = \{\langle \mu_A(x) | x \rangle\}$ – нечітка множина, що описує обмеження на можливі значення змінної a (семантику). Нечітка змінна – це теж саме, що і нечітке число, тільки з додаванням імені, яким формалізується поняття, що описується цим числом.

Лінгвістична змінна визначається як $\langle B, T, X, G, M \rangle$, де B – найменування змінної, T – множина її значень (базова терм-множина), що складається з найменувань нечітких змінних, областю визначення кожної з яких є множина X ; G – синтаксична процедура (граматика), що дозволяє оперувати елементами терм-множини T , зокрема – генерувати нові осмислені терми; $T' = T \cup G(T)$ задає розширену терм-множину (\cup – знак об’єднання); M – семантична процедура, що дозволяє приписати кожному новому значенню лінгвістичної змінної нечітку семантику, шляхом формування нової нечіткої множини.

Лінгвістична змінна – це множина нечітких змінних, вона використовується для того, щоб дати словесний опис деякому нечіткому числу, отриманому в результаті деяких операцій.

Терм-множина – це множина всіх можливих значень лінгвістичної змінної.

Терм – будь-який елемент терм-множини. У теорії нечітких множин терм формалізується нечіткою множиною за допомогою функції належності.

Нечіткий терм – це нечітка множина, яка має властивість, якій відповідає певне поняття.

Операції над нечіткими множинами використовують для отримання на їх основі нової множини. Розглянемо найбільш важливі операції.

Нехай A та B – нечіткі множини на універсальній множині E .

Доповнення (заперечення множини) \bar{A} : інвертується належність кожного елемента: $\forall x \in E, M = [0, 1]: \mu_{\bar{A}}(x) = 1 - \mu_A(x)$. Тут доповнення визначене для $M = [0, 1]$, але очевидно, що його можна визначити для будь-якого упорядкованого M .

Об’єднання (логічна сума множин): $A \cup B$ – найменша нечітка підмножина, що містить як A , так і B , з функцією належності: $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$ – створюється нова множина з елементів вихідних множин, причому для одинакових елементів належність береться максимальною. Операція об’єднання моделює логічне зв’язування «АБО».

Перетинання (логічний добуток множин): $A \cap B$ – найбільша нечітка підмножина, що міститься одночасно в A та B : $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$ – створюється нова множина з однакових елементів вихідних множин, належність яких береться мінімальною. Операція перетинання моделює логічне зв'язування «ТА».

Нечітка імплікація $A \rightarrow B$ визначає причинно-наслідкове відношення між умовами та наслідками правил. Okремо виділяють нечіткі імплікації, які визначаються:

- за Заде: $\mu_{A \rightarrow B}(x) = \max\{\min\{\mu_A(x), \mu_B(x)\}, (1-\mu_A(x))\}$,
- за Мамдані: $\mu_{A \rightarrow B}(x) = \min\{\mu_A(x), \mu_B(x)\}$.

Нечіткі відношення дозволяють задавати зв'язки між множинами.

Нечітке n-арне відношення визначається як нечітка підмножина R на E , що приймає свої значення в M , де $E = E_1 \times E_2 \times \dots \times E_n$ – прямий добуток універсальних множин, M – деяка множина належностей (наприклад, $M = [0; 1]$).

У випадку $n = 2$ і $M = [0, 1]$, бінарним нечітким відношенням R між множинами $X = E_1$ і $Y = E_2$ буде називатися функція $R: (X, Y) \rightarrow [0, 1]$, що ставить у відповідність кожній парі елементів $(x, y) \in X \times Y$ величину $\mu_R(x, y) \in [0; 1]$.

Нечітке відношення на $X \times Y$ записується у вигляді: $x \in X$, $y \in Y: x R y$.

У випадку, коли $X = Y$, тобто X і Y збігаються, нечітке відношення $R: X \times X \rightarrow [0, 1]$ називається *нечітким відношенням на множині* X .

Операції над нечіткими відношеннями задаються подібно до операцій над нечіткими множинами.

Об'єднання двох відношень $R_1 \cup R_2$ визначається з виразу:

$$\mu_{R_1 \cup R_2}(x, y) = \max(\mu_{R_1}(x, y), \mu_{R_2}(x, y)).$$

Перетинання двох відношень $R_1 \cap R_2$ визначається з виразу:

$$\mu_{R_1 \cap R_2}(x, y) = \min(\mu_{R_1}(x, y), \mu_{R_2}(x, y)).$$

(Max-min)-композицією або *(max-min)-згорткою* нечітких відношень $R_1: (X \times Y) \rightarrow [0, 1]$ між X і Y та $R_2: (Y \times Z) \rightarrow [0, 1]$ між Y та Z , називається $R_1 \bullet R_2$ – нечітке відношення між X і Z , визначене через R_1 та R_2 як:

$$\mu_{R_1 \bullet R_2}(x, z) = \max_y [\min(\mu_{R_1}(x, y), \mu_{R_2}(y, z))].$$

(Max-) – композиція відношень R_1 та R_2 :*

$$\mu_{R_1 * R_2}(x, z) = \max_y [\mu_{R_1}(x, y) * \mu_{R_2}(y, z)],$$

де $*$ – будь-яка операція, для якої виконуються ті ж обмеження, що і для \min : асоціативність і монотонність (у змісті неубування) за кожним аргументом. Зокрема, операція \min може бути замінена алгебраїчним множенням $prod$ – тоді говорять про *(max-prod)-композицію*, або операцією максимуму max – тоді говорять про *(max-max)-композицію*, або операцією середнього арифметичного $average$ – тоді говорять про *(max-average)-композицію*.

(Min-) – композиція відношень R_1 та R_2 :*

$$\mu_{R_1 * R_2}(x, z) = \min_y [\mu_{R_1}(x, y) * \mu_{R_2}(y, z)],$$

де $*$ – будь-яка операція, для якої виконуються ті ж обмеження, що і для max : асоціативність і монотонність (у змісті неубування) за кожним аргументом. Зокрема, операцією $*$ може бути операція максимуму max – тоді говорять про *(min-max)-композицію*, або операція мінімуму \min – тоді говорять про *(min-min)-композицію*.

(Sum-prod)-композиція відношень R_1 та R_2 :

$$\mu_{R_3}(x, z) = f \left(\sum_y (\mu_{R_1}(x, y) \mu_{R_2}(y, z)) \right),$$

де $f()$ – певна логістична функція типу сигмоїдної, що обмежує значення функції числом з інтервалу $[0; 1]$.

Нечітка база знань. Нехай $\mu_{jp}(x_i)$ – функція належності входу x_i нечіткому терму a_i^{jp} , $i = 1, 2, \dots, n; j = 1, 2, \dots, m; p = 1, 2, \dots, k_j$; $\mu_{d_j}(y)$ – функція належності виходу y нечіткому терму d_j , $j = 1, 2, \dots, m$. Тоді ступінь належності конкретного вхідного вектора $x^* = \{x_1^*, x_2^*, \dots, x_n^*\}$ нечітким термам d_j з бази знань визначається такою системою нечітких логічних рівнянь:

$$\mu_{d_j}(x^*) = \max_{p=1, 2, \dots, k_j} \min_{i=1, 2, \dots, n} \{\mu_{jp}(x_i^*)\}, \quad j = 1, 2, \dots, m.$$

Нечітка множина y , відповідна вхідному вектору x^* , яка має верхню і нижню межі діапазону значень \bar{y} та \underline{y} , відповідно, визначається як:

$$y = \bigcup_{j=1}^m \int_{\underline{y}}^{\bar{y}} \min \left(\mu_{d_j}(x^*), \mu_{d_j}(y) \right) dy.$$

Чітке значення виходу y^* , що відповідає вхідному вектору x^* , визначається в результаті дефаззифікації нечіткого y .

Знання експерта $A \rightarrow B$ відбиває нечітке причинне відношення передумови і висновку, тому його можна назвати нечітким відношенням і позначити через R : $R = A \rightarrow B$, де « \rightarrow » називають *нечіткою імплікацією*.

Відношення R можна розглядати як нечітку підмножину прямого добутку $X \times B$ повної множини передумов X і висновків B . Таким чином, процес одержання (нечіткого) результату виведення B' з використанням даного спостереження A' і знання $A \rightarrow B$ можна подати у виді композиційного правила *нечіткий «modus ponens»*: $B' = A' \bullet R = A' \bullet (A \rightarrow B)$, де « \bullet » – операція згортки.

Як операцію композиції, так і операцію імплікації в алгебрі нечітких множин можна реалізовувати по-різному (при цьому буде відрізнятися її одержуваний результат), але в будь-якому випадку загальне логічне виведення здійснюється за такі чотири етапи.

1) *Уведення нечіткості* (фаззифікація – fuzzification). Функції належності, визначені на вхідних змінних, застосовуються до їхніх фактичних значень для визначення ступеня істинності кожної передумови кожного правила.

2) *Логічне виведення*. Обчислене значення істинності для передумов кожного правила застосовується до висновків кожного правила. Це приводить до однієї нечіткої підмножини, що буде призначена кожній змінній виведення для кожного правила. У якості правил логічного виведення звичайно використовуються тільки операції *min* (мінімум) або *prod* (множення). У логічному виведенні мінімуму функція належності виведення «відтінається» за висотою, що відповідає обчисленню ступеня істинності передумови правила (нечітка логіка «ТА»). У логічному виведенні множення функція належності виведення масштабується за допомогою обчисленого ступеня істинності передумови правила.

3) *Композиція*. Усі нечіткі підмножини, призначені до кожної змінної виведення (у всіх правилах), поєднуються разом, щоб сформувати одну нечітку підмножину для всіх змінних виведення. При подібному об'єднанні звичайно використовуються операції *max* (максимум) або *sum* (сума). При композиції максимуму комбіноване виведення нечіткої підмножини конструюється як поточковий максимум по всіх нечітких підмножинах (нечітка логіка «АБО»). При композиції суми комбіноване виведення нечіткої підмножини формується як поточкова сума по всіх нечітких підмножинах, призначених змінній виведення правилами логічного виведення.

4) *Приведення до чіткості* (дефузифікація – defuzzification) використовується, якщо потрібно перетворити нечіткий набір виведень у чітке число.

Система нечіткого виведення складається з п'яти функціональних блоків (рис. 4.5):

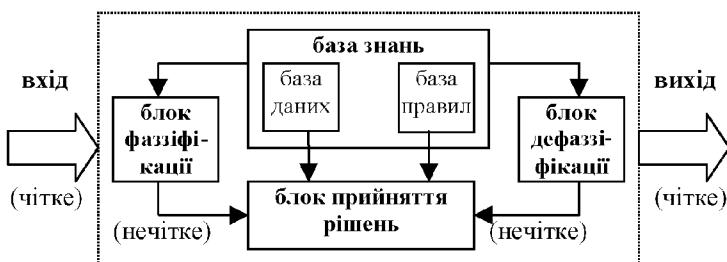


Рисунок 4.5 – Система нечіткого виведення

- блок *фаззіфікації*, що перетворює чисельні вхідні значення в ступінь відповідності лінгвістичним змінним;
- *база правил*, що містить набір нечітких правил типу якщо-то;
- *база даних*, у якій визначені функції належності нечітких множин, що використовуються в нечітких правилах;
- блок *прийняття рішень*, який виконує операції виведення на основі існуючих правил;
- блок *дефаззіфікації*, що перетворює результати виведення в чисельні значення.

Виділяють три основних типи систем нечіткого виведення:

- 1-й тип: вихідне значення знаходиться як зважене середнє результатів виконання кожного правила, для кожного з яких *дефаззіфікація* проводиться окремо; для таких систем вихідні функції належності повинні бути монотонно-неспадаючими;
- 2-й тип: вихідне нечітке значення – це результат об’єднання нечітких виходів кожного правила; кожний нечіткий вихід зважено за допомогою ваг спрацьовування правил; чітке вихідне значення знаходиться в результаті *дефаззіфікації* об’єднаного нечіткого виходу;
- 3-й тип: система, побудована на правилах типа Сугено; вихідне значення є лінійною комбінацією вхідних значень плюс деяке постійне значення, загальний вихід є середнім зваженим всіх правил.

В загальному випадку в якості значень вхідних та вихідних змінних правил можна використовувати нечіткі множини, з якими не пов’язано ніяке поняття – оскільки при проведенні нечіткого виведення нечіткі терми все одно представляються нечіткими множинами і пов’язане з нечітким термом поняття не відіграє ніякої ролі.

Фаззіфікація (fuzzification) – це визначення ступеня виконання антецедентів правил. За допомогою фаззіфікації чіткому значенню ставляється у відповідність ступені його належності до нечітких множин.

Дефаззіфікація (defuzzification) – процедура перетворення нечіткої множини в чітке число за ступенем належності.

У теорії нечітких множин процедура дефаззіфікації є аналогічною заходженню характеристик положення (математичного сподівання, моди, медіані) випадкових величин у теорії ймовірностей. Найпростішим способом виконання процедури дефаззіфікації є вибір чіткого числа, що відповідає максимуму функції належності. Однак придатність цього способу обмежується лише одноекстремальними функціями належності.

В системах нечіткого виведення функції консеквенту, отримані в результаті виконання правил, об’єднуються в одну функцію $\mu(y)$. Існують різні методи дефаззіфікації цієї об’єднаної функції належності.

Нехай y – нечітка змінна, Y – область визначення змінної y , y^* – чітке значення нечіткої змінної y .

Методи дефаззіфікації можна записати у такому вигляді:

—середній з максимальних (MOM – mean of maximum):

$$y^* = \frac{1}{|\text{MAX}(\mu_Y)|} \sum_{y \in \text{MAX}(\mu_Y(y))} y,$$

де $\text{MAX}(\mu_Y) = \{y \in Y \mid \forall y' \in Y : \mu_Y(y') \leq \mu_Y(y)\}$ – це множина значень вихідної змінної, при яких функція належності приймає максимальне значення, ця множина має бути непустою; $\text{MAX}(\mu_Y)$ – кількість елементів множини $\text{MAX}(\mu_Y)$;

—найбільший з максимальних (LOM – largest of maximum):

$$y^* = \max(\text{MAX}(\mu_Y(y)));$$

—найменший з максимальних (SOM – smallest of maximum):

$$y^* = \min(\text{MAX}(\mu_Y(y)));$$

—максимум функції належності:

$$y^* = \arg \sup_y \mu_Y(y),$$

де $\mu_Y(y)$ – унімодальна функція;

—центр тяжіння (COG – center of gravity, центроїд – centroid):

$$y^* = \frac{\sum_{i=1}^k y_i \mu_Y(y_i)}{\sum_{i=1}^k \mu_Y(y_i)};$$

де y_i – i -ий синглтон (одноточкова нечітка множина), $\mu_Y(y_i)$ – значення функції належності для i -го елемента нечіткої множини Y ;

—центр площини (center of area): чітке значення вихідної змінної y^* визначається з рівняння:

$$\int_{Y_{\min}}^{y^*} \mu_Y(y) dy = \int_{y^*}^{Y_{\max}} \mu_Y(y) dy;$$

- метод медіані (bisector):

$$y^* = \min_{\forall j: \sum_{i=1}^j \mu_Y(y_i) \geq \frac{1}{2} \sum_{i=1}^k \mu_Y(y_i)} (y_i);$$

- висотна дефаззифікація (height defuzzification):

$$y^* = \frac{\sum_{Y_\alpha} y_i \mu_Y(y_i)}{\sum_{Y_\alpha} \mu_Y(y_i)},$$

де A_α – нечітка множина α -рівня. Елементи нечіткої множини, для котрих значення функції належності менше, ніж певний рівень α , до розрахунків не беруться.

Методи нечіткого виведення.

Прямий (вихідний) метод нечіткого виведення заснований на використанні нечіткого узагальнення правила *modus ponens*, який стосовно до систем нечітких продукцій реалізується тим, що окремі факти проблемної області перетворюються у конкретні значення функцій належності умов нечітких продукцій. Після чого за одним з методів нечіткої композиції знаходяться значення функцій належності висновків правих частин за кожним з правил нечітких продукцій. Ці значення функцій належності або є шуканим результатом виведення, або можуть бути використані як додаткові умови у базі правил продукцій, що розглядається.

Розглянемо найбільш вживані методи нечіткого виведення.

Метод Мамдані (E. Mamdani) використовує базу знань із правилами у Мамдані та передбачає виконання таких дій:

1) Уведення нечіткості. Знаходяться ступені істинності для передумов кожного правила: $\mu_{jp}(x_i^*)$, $i = 1, 2, \dots, n; j = 1, 2, \dots, m; p = 1, 2, \dots, k_j$.

2) Логічне виведення. Знаходяться рівні «відтинання» для передумов кожного з правил (з використанням операції мінімум):

$$\mu_j(y) = \min_{i,p} \mu_{jp}(x_i^*), \quad i = 1, 2, \dots, n; j = 1, 2, \dots, m; p = 1, 2, \dots, k_j.$$

Потім знаходяться «усічені» функції належності:

$$\mu_j'(y) = \min(\mu_j(y), \mu_{d_j}(y)), j = 1, 2, \dots, m.$$

3) Композиція. Здійснюється об'єднання знайдених усічених функцій з використанням операції максимум, що приводить до одержання підсумкової нечіткої підмножини для змінної виходу з функцією належності:

$$\mu(y) = \max_j(\mu_j'(y)), j = 1, 2, \dots, m.$$

4) Приведення до чіткості – проводиться для отримання y^* , наприклад, центроїдним методом.

Метод Цукамото (Y. Tsukamoto): Вихідні посили – як у попереднього методу, але тут передбачається, що функції $\mu_{d_j}(y)$ є монотонними.

1) Уведення нечіткості. Знаходяться ступені істинності для передумов кожного правила: $\mu_{jp}(x_i^*), i = 1, 2, \dots, n; j = 1, 2, \dots, m; p = 1, 2, \dots, k_j$.

2) Нечітке виведення. Знаходяться рівні «відтинання» для передумов кожного з правил:

$$\mu_j(y) = \min_{i,p} \mu_{jp}(x_i^*), i = 1, 2, \dots, n; j = 1, 2, \dots, m; p = 1, 2, \dots, k_j,$$

а потім для кожного вихідного правила визначаються чіткі значення y_j шляхом розв'язку рівняння: $\mu_j(y) = \mu_{d_j}(y_j), j = 1, 2, \dots, m$.

3) Визначається чітке значення змінної y^* на основі центроїдного методу.

Метод Ларсена (H. Larsen): нечітка імплікація моделюється з використанням оператора множення.

1) Уведення нечіткості. Знаходяться ступені істинності для передумов кожного правила: $\mu_{jp}(x_i^*), i = 1, 2, \dots, n; j = 1, 2, \dots, m; p = 1, 2, \dots, k_j$.

2) Нечітке виведення. Знаходяться рівні «відтинання» для передумов кожного з правил:

$$\mu_j(y) = \min_{i,p} \mu_{jp}(x_i^*), i = 1, 2, \dots, n; j = 1, 2, \dots, m; p = 1, 2, \dots, k_j,$$

а потім визначаються часткові нечіткі підмножини:

$$\mu_j(y)\mu_{d_j}(y), j = 1, 2, \dots, m.$$

3) Знаходиться підсумкова нечітка підмножина:

$$\mu(y) = \max_j(\mu_j(y)\mu_{d_j}(y)), j = 1, 2, \dots, m.$$

4) При необхідності здійснюється приведення до чіткості.

Спрощений метод нечіткого виведення: Вихідні правила в даному випадку задаються у виді: Якщо $(x_1 = a_1^{j1})$ та $(x_2 = a_2^{j1})$ та ... та $(x_n = a_n^{j1})$, то $y = d_j$ для всіх $j = 1, 2, \dots, m$. Тут d_j – нечіткі числа.

1) Уведення нечіткості. Знаходяться ступені істинності для передумов кожного правила: $\mu_{jp}(x_i^*), i = 1, 2, \dots, n; j = 1, 2, \dots, m; p = 1, 2, \dots, k_j$.

2) Нечітке виведення. Знаходяться числа

$$\mu_j(y) = \min_{i,p} \mu_{jp}(x_i^*), i = 1, 2, \dots, n; j = 1, 2, \dots, m; p = 1, 2, \dots, k_j.$$

3) Визначається чітке значення вихідної змінної y^* для нечіткої множини $y = \{\mu_j(y)|d_j\}$ на основі центроїдного методу.

Метод Сугено: М. Сугено (M. Sugeno) та Т. Такагі (T. Takagi) використовували набір правил у формі: Якщо $(x_1 = a_1^{j1})$ та $(x_2 = a_2^{j1})$ та ... та $(x_n = a_n^{j1})$, то $y = w_1x_1 + w_2x_2 + \dots + w_nx_n$, для всіх $i = 1, 2, \dots, n; j = 1, 2, \dots, m$. Тут w_i – деякі вагові коефіцієнти.

1) Уведення нечіткості. Знаходяться ступені істинності для передумов кожного правила: $\mu_{jp}(x_i^*), i = 1, 2, \dots, n; j = 1, 2, \dots, m; p = 1, 2, \dots, k_j$.

2) Нечітке виведення. Знаходяться рівні «відтинання» для передумов кожного з правил:

$$\mu_j(y) = \min_{i,p} \mu_{jp}(x_i^*), i = 1, 2, \dots, n; j = 1, 2, \dots, m; p = 1, 2, \dots, k_j,$$

а також індивідуальні виходи правил:

$$d_j = w_1x_1 + w_2x_2 + \dots + w_nx_n$$

3) Визначається чітке значення змінної виведення y^* для нечіткої множини $y = \{\mu_j(y)|d_j\}$ на основі центроїдного методу.

Раніше розглянуті нечіткі логічні виведення є *вихідними виведеннями* від передумов до висновків. У діагностичних нечітких системах часто застосовуються спадні виведення.

Зворотний (спадний) метод нечіткого логічного виведення заснований на використанні нечіткого узагальнення правила виведення *modus tollens*.

Нехай заданий повний простір передумов $X = \{x_1, \dots, x_m\}$ і повний простір висновків $Y = \{y_1, \dots, y_n\}$. Між x_i та y_j існують нечіткі причинні відношення $x_i \rightarrow y_j$, які можна подати у вигляді певної матриці R з елементами $r_{ij} \in [0, 1]$, $i = 1, \dots, m; j = 1, \dots, n$. Передумови і висновки можна розглядати як нечіткі множини A та B на просторах X та Y , відношення яких можна подати у вигляді: $B = A \bullet R$, де « \bullet » позначає правило композиції нечітких виведень, наприклад, (max-min)-композицію. У даному випадку направок виведень є зворотним для правил, тобто задана матриця R (знання експерта), спостерігаються виходи B (висновки) і визначаються входи A (передумови).

4.7 Пошук у просторі станів

При проектуванні інтелектуальної системи, заснованої на знаннях, серйозну увагу має бути приділено тому, як здійснюється доступ до знань і як вони використовуються при пошуку рішення.

Пошук – процес формування системою послідовності дій, що дозволяють їй досягти своїх цілей. Перш ніж система зможе приступити до пошуку рішень, повинна бути сформульована мета, а потім ця мета може використовуватися для формулювання задачі. Процес вирішення задач пошуку рішень в інтелектуальних системах, заснованих на знаннях, як правило, являє собою перебір досить великої кількості різних варіантів рішень, кожне з яких можна зіставити деякому стану.

Стан – це колекція характеристик, що можуть використовуватися для визначення стану або статусу об'єкта.

Простір станів – це множина станів, за допомогою якої подаються переходи між станами, у яких може бути об'єкт. У результаті переходу об'єкт попадає з одного стану до іншого.

Простір станів зручно подавати у вигляді гіперграфа.

Гіперграф складається з множини вершин (вузлів) N і множини гіпердрит H .

Вузол – це облікова структура даних, застосовувана для подання дерева пошуку. Кожен вузол має батьківський вузол, містить дані про стан і має різні допоміжні поля.

Листовий вузол – вузол, що не має нащадків у орієнтованому графі.

Периферія – колекція вузлів, що були сформовані, але ще не розгорнуті. Кожен елемент периферії являє собою лист.

Гіпердуга задається упорядкованою парою, у якій перший елемент є окремою вершиною з N , а другий – підмножиною множини N . Гіпердуги також називають *k-коннекторами*, де k – потужність множини породжених вершин.

Граф ТА/АБО (And/Or-Graph) – окремий випадок гіперграфа, у якому вузли з'єднані не окремими дугами, а множиною дуг. Щоб подати різні відношення графічно, на графах ТА/АБО розрізняють вузли: ТА (*and*) і АБО (*or*). (рис. 4.6).



Рисунок 4.6 – ТА/АБО-графи

ТА-вузли графа – вузли, що подають імплікацію передумов, зв'язаних оператором ТА. Дуги, що ведуть до цих вузлів, з'єднуються кривою. Крива, що з'єднує дуги, означає, що для доказу вузла повинні бути істинними усі його передумови, з'єднані дугою.

АБО-вузли графа – вузли, що подають імплікацію передумов, зв'язаних оператором АБО. Дуги, які ведуть до цих вузлів, не з'єднуються кривою, що вказує на те, що істина кожної з передумов є достатньою умовою для істинності висновку.

Задачі пошуку в просторі станів можна сформулювати в термінах трьох найважливіших компонентів:

- *вихідний стан проблеми*;
- *тест завершення* — перевірка, чи досягнуто необхідний кінцевий стан або знайдено рішення проблеми;
- множина операцій, які можна використовувати для зміни поточного стану проблеми.

Простір пошуку може бути визначено такими підаспектами:

– *розміром простору пошуку* – дає узагальнену характеристику складності задачі. Виділяють малі (до 10! станів) і великі (понад 10! станів) простори пошуку;

– *глибиною простору пошуку* – характеризується середнім числом послідовно застосовуваних правил, що перетворюють вихідні дані в кінцевий результат;

– *шириною простору пошуку* – середнім числом правил, придатних до виконання в поточному стані.

Стратегія пошуку в просторі станів – порядок, у якому відбувається розгортання станів. Стратегія пошуку повинна бути виражена у вигляді функції, що вибирає певним чином з периферії наступний вузол, який підлягає розгортанню. Хоча даний підхід концептуально є нескладним, він може виявитися дорогим з обчислювальної точки зору, оскільки функцію, передбачену в цій стратегії, можливо, прийдеться застосовувати до кожного елемента в зазначеній множині для вибору найкращого з них. Тому часто передбачається, що колекцію вузлів реалізовано у вигляді черги.

Стратегія прямого пошуку (forward chaining) – відповідає руху від вихідних вершин графа до цільової вершини.

Стратегія зворотного пошуку (backward chaining) – відповідає руху від цільової вершини до вихідних вершин. Якщо вершин-цілей мало, а вихідних багато, то зворотний пошук є більш природним і ефективним.

Стратегія двонаправленого пошуку (bi-directional chaining) – поєднує прямий пошук (рух від вихідних вершин до цільової) і зворотний пошук (рух від цільової вершини до вихідної) та намагається досягти деякого загального для обох пошуків стану, зупиняючись після того, як два процеси пошуку зустрінуться на середині. У стратегії двонаправленого пошуку передбачається перевірка в одному чи в обох процесах пошуку кожного вузла перед його розгортанням для визначення того, чи не знаходиться він на периферії іншого дерева пошуку; у випадку позитивного результату перевірки вважається, що рішення знайдено. Перевірка належності вузла до іншого дерева пошуку може бути виконана за постійний час за допомогою хеш-таблиці.

Продуктивність методів пошуку оцінюють за допомогою таких показників:

- *повнота* – визначає гарантію виявлення методом рішення, якщо воно існує;
- *оптимальність* – властивість забезпечення методом знаходження оптимального рішення;
- *часова складність* – оцінка часу, за який метод знаходить рішення;
- *просторова складність* – оцінка обсягу пам'яті, необхідного для здійснення пошуку.

Пошуковий метод с припустимим, якщо для будь-якого графа він завжди вибирає оптимальний шлях до рішення.

Ефективність методів пошуку визначається вартістю пошуку, що звичайно залежить від часової складності, але може також включати вираз для оцінки використання пам'яті, або сумарною вартістю, у якій поєднуються вартість пошуку і вартість шляху знайденого рішення.

Схема пошуку на ТА/АБО-графі – спосіб руху по графу в напрямку, заданому стратегією пошуку. Розрізняють схеми *сліпого (нє-інформованого)* і *спрямованого (інформованого) пошуків* на графі, пов'язані з перебором альтернативних вершин-підцілей і організацією повернення.

На практиці сліпі методи пошуку використовуються рідко, оскільки вони пов'язані з великими витратами ресурсів через комбінаторний вибух. Спрямовані методи пошуку використовують як оцінювання придатних альтернатив, так і точок повернення, тобто забезпечують керування поверненням.

Стратегії нєінформованого (сліпого) пошуку – стратегії, що не використовують додаткову інформацію про стани, крім тієї, котра подана у визначені задачі. Вони здатні тільки виробляти нові стани-нащадки і відрізняти цільовий стан від нецільового.

Метод породження і перевірки (generate-and-test).

Крок 1. Генерувати новий стан, модифікуючи існуючий.

Крок 2. Перевірити, чи не є стан кінцевим (рішенням). Якщо це так, то завершити роботу, інакше перейти до кроку 1.

Метод породження і перевірки має два основних варіанти: пошук у глибину та пошук у ширину. Відрізняються варіанти порядком формування станів на кроці 1.

Метод пошуку в ширину (breadth-first search) – це стратегія, у якій простір станів послідовно проглядається по рівнях: на кожному рівні, у свою чергу, послідовно проглядаються стани, подані вузлами цього рівня один за іншим, і тільки якщо станів на даному рівні більше немає, метод переходить до наступного рівню (див. рис. 4.7 – пунктиром показаний порядок перегляду вузлів).

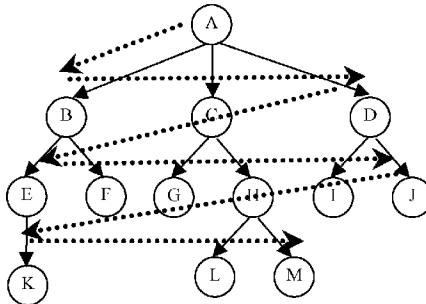


Рисунок 4.7 – Граф, що демонструє роботу методу пошуку в ширину

Нехай s – вузол початкового стану, $Open$ – список, що містить обрані, але необроблені вузли; $Closed$ – список, що містить оброблені вузли. Тоді метод пошуку в ширину буде полягати у виконанні таких кроків.

Крок 1. Ініціалізація. Установити: $Open = \{s\}$, $Closed = \emptyset$.

Крок 2. Якщо $Open = \emptyset$, то припинити виконання – шляху до цільового стану на графі не існує; у протилежному випадку – перейти до кроку 3.

Крок 3. Видалити з $Open$ крайній ліворуч стан x .

Крок 4. Якщо x – ціль, то закінчити пошук і сформувати результат – шлях, породжений простежуванням покажчиків від вузла x до вузла s ; у протилежному випадку – виконати кроки 4.1–4.4.

Крок 4.1 Згенерувати стан-нащадок x .

Крок 4.2 Помістити x у список $Closed$.

Крок 4.3 Виконати перевірку на цикл – виключити нащадок x , якщо він вже є в списку $Open$ або $Closed$.

Крок 4.4 Помістити інші нащадки в правий кінець списку $Open$, створюючи в такий спосіб чергу.

Крок 5. Перейти до кроку 2.

На кожній ітерації методу пошуку в ширину генеруються всі дочірні вершини стану x і записуються в список *Open*, що діє як черга й обробляє дані в порядку надходження: стани додаються в список праворуч, а видаляються ліворуч. У такий спосіб у пошуку беруть участь стани, що знаходяться в списку *Open* довше всього, забезпечуючи пошук у ширину. Дочірні стани, що були вже записані в списки *Open* або *Closed*, відкидаються.

Коли ціль знайдено, метод може відновити шлях рішення, просліджуючи його в зворотному напрямку від мети до початкового стану по батьківських станах. Оскільки пошук у ширину знаходить кожний стан за найкоротшим шляхом і зберігає першу версію кожного стану, цей шлях від початку до цілі є найкоротшим. Методи, що мають таку властивість, називають *розв'язними* (admissible).

Метод пошуку в глибину (depth-first search) – це стратегія, у якій переглядаються стани на одному шляху – розгортається найглибший вузол у поточній периферії дерева пошуку, у результаті чого пошук безпосередньо переходить на найглибший рівень дерева пошуку, на якому вузли не мають нащадків. У міру того, як ці вузли розгортаються, вони видаляються з периферії, тому надалі пошук переходить до наступного найповерхневого вузла, що усе ще має недосліджених нащадків (див. рис. 4.8 – пунктиром показаний порядок перегляду вузлів).

Нехай s – вузол початкового стану, *Open* – список, що містить обрані, але необроблені вузли; *Closed* – список, що містить оброблені вузли. Тоді метод пошуку в глибину буде полягати у виконанні таких кроків.

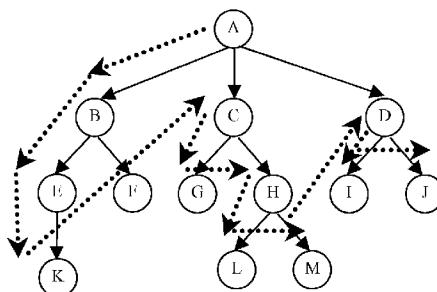


Рисунок 4.8 – Граф, що демонструє роботу методу пошуку в глибину

Крок 1. Ініціалізація. Установити: $Open = \{s\}$, $Closed = \emptyset$.

Крок 2. Якщо $Open = \emptyset$, то припинити виконання – шляху до цільового стану на графі не існує; у противному випадку – перейти до кроку 3.

Крок 3. Видалити з $Open$ крайній ліворуч стан x .

Крок 4. Якщо x – ціль, то закінчити пошук і сформувати результат – шлях, породжений простежуванням покажчиків від вузла x до вузла s ; у противному випадку – виконати кроки 4.1–4.4.

Крок 4.1 Згенерувати стан-нащадок x .

Крок 4.2 Помістити x у список $Closed$.

Крок 4.3 Виконати перевірку на цикл – виключити нащадок x , якщо він вже є в списку $Open$ або $Closed$.

Крок 4.4 Помістити інші нащадки в лівий кінець списку $Open$, створюючи в такий спосіб стек.

Крок 5. Перейти до кроку 2.

При пошуку в глибину після дослідження стану спочатку необхідно оцінити всіх його нащадків та їхніх нащадків, а потім досліджувати кожну з вершин-сестер. Пошук у глибину за можливістю поглибується в область пошуку. Якщо подальші нащадки стану не знайдені, розглядаються вершини-сестри. У цьому методі стани-нащадки додаються і видаляються з лівого кінця списку $Open$, тобто список $Open$ реалізований як стек магазинного типу. При організації списку $Open$ у вигляді стека перевага віддається самим «молодим» (нешодавно згенерованим) станам.

Пошук у глибину швидко проникає в глибини простору.

Якщо відомо, що шлях рішення буде довгим, то пошук у глибину не буде витрачати час на пошук великої кількості поверхневих станів на графі.

Пошук у глибину ефективний для областей пошуку з високим ступенем зв'язаності, оскільки йому не потрібно запам'ятовувати усі вузли даного рівня в списку $Open$.

Недоліком пошуку в глибину є те, що в ньому може бути зроблено неправильний вибір і перехід у тупикову ситуацію, пов'язану з проходженням униз по дуже довгому (чи навіть нескінченному) шляху, при тому, що інший варіант міг би привести до рішення, яке знаходиться недалеко від кореня дерева пошуку.

Після вибору стратегії пошуку (на основі даних або від мети), оцінки графа і вибору методу пошуку подальший хід рішення буде залежати від конкретної задачі.

Результатом застосування будь-якого методу рішення задачі є або невдале завершення, або одержання рішення. Деякі методи можуть входити в нескінчений цикл і не повернати ніякого результату.

Методи пошуку на графах використовують ідеї пошуку з поверненнями.

Метод пошуку з поверненнями (backtracking) – це метод систематичної перевірки різних шляхів у просторі станів при якому формується список недосліджених станів *Open*, для того щоб мати можливість повернутися до кожного з них; підтримується список переглянутих «невдалих» станів *Closed*, щоб відгородити метод від перевірки марних шляхів; підтримується список вузлів поточного шляху, що повертається по досягненні мети; кожен новий стан перевіряється на входження в ці списки, щоб запобігти зацикленню.

Пошук з поверненнями запускається з початкового стану і проводиться по деякому шляху доти, поки не досягне мети або не упреться в тупик. Якщо ціль досягнуто, пошук завершується, і як рішення задачі повертається шлях до цілі. Якщо ж пошук привів у тупикову вершину, то метод повертається в найближчу з пройдених вершин і досліджує всі її вершини-сестри, а потім спускається по одній з гілок, що ведуть від вершини-сестри. Цей процес описується таким рекурсивним правилом.

Якщо вихідний стан не задовольняє вимогам цілі, то зі списку його нащадків вибираємо першого нащадка, і до цієї вершини рекурсивно застосовуємо процедуру пошуку з поверненнями. Якщо в результаті пошуку з поверненнями в підграфі з коренем першого нащадка ціль не виявлено, то повторюємо процедуру для вершини-сестри – другого нащадка вихідного стану. Ця процедура продовжується доти, поки один з нащадків розглянутої вершини-сестри не виявиться цільовим вузлом, або не з'ясується, що розглянуто уже всіх можливих нащадків (усіх вершин-сестер). Якщо ж жодна з вершин-сестер вихідної вершини не привела до цілі, то повертаємося до предка вихідної вершини і повторюємо процедуру з вершиною-сестрою вихідної вершині і т. д.

Оскільки сліпий пошук можливий тільки в невеликому просторі варіантів, є необхідним деякий спосіб спрямованого пошуку.

Стратегії інформованого (евристичного) пошуку – стратегії, що дозволяють визначити, чи є один нецільовий стан більш перспективним у порівнянні з іншим. Ці стратегії використовують при пошуку шляху на графі в просторі станів деякі знання, специфічні для конкретної предметної області.

Найкраще розглядати *евристику* як деяке правило впливу, що, хоча і не гарантує успіху (як детермінований метод або процедура прийняття рішення), у більшості випадків виявляється дуже корисним.

У загальному вигляді *евристичний пошук* можна податі як послідовність таких етапів.

Етап 1. Обрати з множини можливих дій деяку дію:

- з урахуванням відповідності до цілі:
 - зменшення деякого небажаного розходження,
 - безпосереднє вирішення тієї чи іншої підзадачі;
- з урахуванням досвіду:
 - повторення минулого,
 - виявлення ключової дії;
- з урахуванням необхідних умов:
 - рішення, обумовлене аналізом ситуації,
 - виключення нездійсненного варіанта;
- з урахуванням фактора випадковості:
 - перевага віддається розмаїтості.

Етап 2. Здійснити обрану дію і змінити поточну ситуацію.

Етап 3. Оцінити ситуацію:

- за аналогією:
 - відома сама задача,
 - відома підзадача;
- за величиною відстані до цілі:
 - відстань між двома ситуаціями,
 - кількість зусиль, затрачуваних на пошук;
- за математичним критерієм:
 - складання переліку необхідних та / або достатніх для одержання даного рішення характеристик,
 - чисельна оцінна функція,
 - верхні і нижні межі,
 - сума вартостей, обраних придатним способом;

- за очікуваним виграшем (критерій, пов'язаний з минулим досвідом):
 - простота ситуації,
 - коефіцієнт розширення пошуку,
 - інший критерій (складність задачі, затрачуваний на її рішення час і т. д.).

Етап 4. Відкинути некорисні ситуації.

Етап 5. Якщо досягнуто кінцеву ситуацію – кінець; у противному випадку – вибрати нову, вихідну ситуацію і почати спочатку:

- рухатися тільки вперед:
- систематичний розвиток останньої породженої ситуації;
- виконувати всі дії паралельно:
 - почергове виконання всіх дій;
- як вихідну обирати найбагатообіцяючу ситуацію:
 - у відношенні оцінкою функції,
 - у відношенні незначного числа входних у неї дій;
- йти на компроміс між:
 - глибиною пошуку,
 - оцінкою ситуації.

Метод пошуку зі сходженням до вершини (пошук екстремуму, hill climbing) – проста форма евристичного пошуку, що є фундаментальним методом локального пошуку. У процесі пошуку використовується певна *оцінна функція*, за допомогою якої можна грубо оцінити, наскільки гарним або поганим є поточний стан. При цьому оцінюють не тільки поточний стан пошуку, але і його нащадків. Для подальшого пошуку вибирається найкращий нащадок; при цьому про його братів і батьків просто забивають. Пошук припиняється, коли досягається стан, що краще ніж кожний з його нащадків.

Метод сходження до вершини, можна сформулювати у такий спосіб.

Крок 1. Знаходячись у даній точці простору станів, застосувати правила породження нової множини можливих рішень.

Крок 2. Якщо один з нових станів є рішенням проблеми, припинити процес. У противному випадку перейти в той стан, що характеризується найвищим значенням оцінної функції. Повернутися до кроку 1.

Пошук зі сходженням до вершини іноді називають *жадібним*

локальним пошуком, оскільки в процесі його виконання відбувається захоплення самого гарного сусіднього стану без попередніх міркувань про те, куди варто відправитися далі. Якщо кількість найкращих нащадків вузла більше одного, то метод звичайно передбачає випадковий вибір нащадка серед них.

У методі зі сходженням до вершини не здійснюється прогнозування за межами станів, що є безпосередньо сусідніми стосовно поточного стану.

Жадібні методи часто показують дуже високу продуктивність, але застосування цієї стратегії наштовхується на такі труднощі:

- необхідність формульовання оцінної функції, яка б адекватно відбивала якість поточного стану;

- тенденція зупинятися в *локальному максимумі* – рішенні, з якого можливий тільки спуск, тобто погіршення стану. Як тільки метод досягає стану, що має кращу оцінку, ніж його нащадки, він зупиняється. Якщо цей стан не є рішенням задачі, а тільки локальним максимумом, то такий метод неприйнятний для даної задачі. Це значить, що рішення може бути оптимальним на обмеженій множині, але через форму всього простору, можливо, ніколи не буде обране найкраще рішення.

Оскільки в цій стратегії дані про попередні стани не зберігаються, то вона без механізмів повернення або інших прийомів відновлення не може відрізнити локальний максимум від глобального, у наслідок чого метод не може бути відновлений із точки, що привела до невдачі.

Існують методи наближеного вирішення цієї проблеми, наприклад, випадкове збурювання оцінки. Однак гарантовано вирішувати задачі з використанням даної стратегії пошуку не можна.

Незважаючи на ці обмеження, метод пошуку зі сходженням до вершини може бути досить ефективним, якщо оцінна функція дозволяє уникнути локального максимуму і зациклення методу.

Метод пошуку за першим найкращим збігом (жадібний пошук, best-first search) – форма евристичного пошуку, що використовує оцінну функцію, за допомогою якої можна порівнювати стани в просторі станів.

Кожному стану n відповідає значення оцінної функції $f = g(n) + h(n)$, де $g(n)$ – глибина стану n у просторі пошуку, оцінка яко-

го утримує метод від завзятого проходження по невірному шляху, $h(n)$ – деяка евристична оцінка відстані від стану n до цілі, що веде метод пошуку до евристично найбільш перспективних станів.

Метод порівнює не тільки ті стани, у які можливий переход з поточного стану, але і всі, до яких можна дістати. Такий метод вимагає великих обчислювальних ресурсів, але його ідея полягає в тому, щоб брати до уваги не тільки найближчі стани, тобто локальну обстановку, а переглянути як можна більшу ділянку простору станів і бути готовим, у разі потреби, повернутися туди, де він уже був, і піти іншим шляхом, якщо найближчі претенденти не обіцяють істотного прогресу в досягненні мети. Саме ця можливість відмовитися від частини пройденого шляху заради глобальної мети і дозволяє знайти більш ефективний шлях.

Метод використовує списки збережених станів: список *Open* відслідковує поточний стан пошуку, а в *Closed* записуються вже перевірені стани. На кожному кроці метод записує в список *Open* стан з урахуванням деякої евристичної оцінки його близькості до цілі. Таким чином, на кожній ітерації розглядаються найбільш перспективні стани зі списку *Open*. Стани в списку *Open* сортуються у відповідності зі значеннями оцінної функції f . Зберігаючи всі стани в списку *Open*, метод зможе уберегти себе від невдалого завершення. У жадібному методі пошуку при використанні пріоритетної черги можливе відновлення точки локального максимуму.

Метод припускає виконання таких кроків.

Крок 1. Установити: $Open=\{S\}$, $Closed=\emptyset$, де S – початковий стан.

Крок 2. Якщо $Open = \emptyset$, то перейти до кроку 9, у протилежному випадку – перейти до кроку 3.

Крок 3. Видалити перший стан X зі списку *Open*.

Крок 4. Якщо X – цільовий стан, то зупинення, повернути шлях від S до X ; у протилежному випадку – перейти до кроку 5.

Крок 5. Згенерувати нащадків X .

Крок 6. Для кожного нащадка X виконувати кроки 6.1–6.3.

Крок 6.1 Якщо нащадок не міститься в списках *Open* та *Closed*, то зіставити нащадку евристичне значення і додати нащадка в список *Open*.

Крок 6.2 Якщо нащадок міститься в списку *Open* і нащадок був досягнутий по найкоротшому шляху, то записати цей стан у список *Open*.

Крок 6.3 Якщо нащадок міститься в списку *Closed* і нащадок був досягнутий по найкоротшому шляху, то видалити стан зі списку *Closed* і додати нащадок у список *Open*.

Крок 7. Помістити X у список *Closed*.

Крок 8. Переупорядкувати стани в списку *Open* відповідно до евристики «країці ліворуч».

Крок 9. Зупинення. Повернути відмову – список *Open* порожній.

На кожній ітерації метод видаляє перший елемент зі списку *Open*. Досягнувши мети, він повертає шлях, що веде до рішення. Помітимо, що кожен стан зберігає інформацію про попередній стан, щоб згодом відновити його і дозволити методу знайти найкоротший шлях до рішення.

Якщо перший елемент у списку *Open* не є рішенням, то метод використовує правила перевірки відповідності й операції, щоб згенерувати всіх можливих нащадків даного елемента. Якщо нащадок уже знаходиться в списку *Open* або *Closed*, то метод вибирає найкоротший із двох можливих шляхів досягнення цього стану.

Потім метод обчислює евристичну оцінку станів у списку *Open* і сортує список відповідно до цих евристичних значень. При цьому країці стани ставляться на початок списку. Помітимо, що через евристичну природу оцінювання наступний стан повинний перевірятися на будь-якому рівні простору станів. Відсортований список *Open* часто називають *пріоритетною чергою* (priority queue).

На відміну від пошуку екстремуму, що не зберігає пріоритетну чергу для відбору наступних станів, даний метод відновлюється після помилки і знаходить цільовий стан.

Компонент $g(n)$ функції оцінки додає жадібному пошуку властивості пошуку в ширину. Він запобігає можливості заблукання через помилкову оцінку: якщо евристика безупинно повертає гарні оцінки для станів на шляху, що не веде до мети, то значення g буде рости і домінувати над h , повертаючи процедуру пошуку до найкоротшого шляху. Це рятує метод від зациклення.

Пошук за першим найкращим збігом нагадує пошук у глибину в тому відношенні, що цей метод надає переваги на шляху до мети постійному крокуванню по єдиному шляху, але метод повертається до попередніх вузлів після влучення в тупик. Даний метод страждає від тих же недоліків, що і пошук у глибину: він

не є оптимальним, до того ж він – неповний (оскільки здатний відправитися по нескінченному шляху та так і не повернутися, щоб випробувати інші можливості)..

Метод рекурсивного пошуку за першим найкращим збігом (Recursive Best-First Search – RBFS) – простий рекурсивний метод, у якому робляться спроби імітувати роботу стандартного пошуку за першим найкращим збігом, але з використанням тільки лінійного простору. Метод може бути описаний як така функція.

Функція *RBFS* (вузол *node*, межа *f*-вартості f_{limit}) – повертає рішення *result* або індикатор невдачі «відмова» і нову межу *f*-вартості f_{limit} .

Крок 1. Якщо вузол *node* – цільовий, то повернути *result* = *node*.

Крок 2. Сформувати для вузла *node* множину вузлів-нащадків *Successors*.

Крок 3. Якщо *Successors* = \emptyset , то повернути: *result* = «відмова» та $f_{limit} = \infty$.

Крок 4. Для кожного вузла *s* у *Successors* установити: $f(s) = \max(g(s) + h(s), f(node))$.

Крок 5. Установити: *best* = вузол з найменшим *f*-значенням у множині *Successors*.

Крок 6. Якщо $f(best) > f_{limit}$, то повернути *result* = «відмова» та $f_{limit} = f(best)$.

Крок 7. Знайти *alternative* – друге після найменшого *f*-значення для елементів у множині *Successors*.

Крок 8. Установити: *result*, $f(best) = RBFS(best, \min(f_{limit}, alternative))$.

Крок 9. Якщо *result* ≠ «відмова», то повернути *result*.

Перший виклик функції: *RBFS* (*node*=початковий вузол, $f_{limit}=\infty$).

Цей метод контролює *f*-значення найкращого альтернативного шляху, доступного з будь-якого предка поточного вузла. Якщо поточний вузол перевищує дану межу, то поточний етап рекурсії скасовується і рекурсія продовжується з альтернативного шляху. Після скасування даного етапу рекурсії відбувається заміна *f*-значення кожного вузла уздовж даного шляху найкращим *f*-значенням його дочірнього вузла. Завдяки цьому запам'ятовується *f*-значення найкращого листового вузла з забутого піддерева і тому в деякий наступний момент часу може бути прийняте рішення про те, чи варто знову розгорнати це піддерево.

Метод RBFS є оптимальним, якщо евристична функція $h(n)$ є припустимою. Метод RBFS страждає від недоліку, пов'язаного з занадто частим повторним формуванням вузлів.

Метод пошуку A^* – різновид пошуку за першим найкращим збігом. Метод припускає використання для кожного вузла n на графі простору станів *оцінної функції* виду $f(n) = g(n) + h(n)$, де $g(n)$ відповідає відстані на графі від вузла n до початкового стану, $h(n)$ – оцінка відстані від n до вузла, що подає кінцевий (цільовий) стан. Чим менше значення оцінної функції $f(n)$, тим «краще», тобто вузол n лежить на більш короткому шляху від вихідного стану до цільового.

Ідея методу полягає в тому, щоб за допомогою $f(n)$ відшукати найкоротший шлях на графі від вихідного стану до цільового. Звідси випливає, що якщо $h(n)$ – нижня оцінка дійсної відстані до цільового стану, тобто якщо $h(n)$ ніколи не дає завищеної оцінки відстані, то метод A^* завжди відшукав оптимальний шлях до цілі за допомогою оцінної функції $f(n)$.

Нехай s – вузол початкового стану; g – вузол цільового стану; *Open* – список, що містить, обрані, але необроблені вузли; *Closed* – список, що містить оброблені вузли. Тоді метод A^* буде полягати у виконанні таких кроків.

Крок 1. Установити: $Open = \{s\}$, $Closed = \emptyset$.

Крок 2. Якщо $Open = \emptyset$, то припинити виконання – шляху до цільового стану на графі не існує; у противному випадку – перейти до кроku 3.

Крок 3. Видалити зі списку *Open* вузол n , для якого $f(n) < f(m)$ для будь-якого вузла m , що вже знаходиться у списку *Open*, і перенести його в список *Closed*.

Крок 4. Сформувати список чергових вузлів, у які можливий перехід з вузла n , і видалити з нього осі вузли, що утворять петлі; з кожним із вузлів, що залишилися зв'язати покажчик на вузол n .

Крок 5. Якщо в сформованому списку чергових вузлів присутній вузол g , то завершити виконання і сформувати результат – шлях, породжений простежуванням покажчиків від вузла g до вузла s ; у противному випадку для кожного чергового вузла n_i , включеного в список, виконати таку послідовність операцій, що задається кроками 5.1–5.3.

Крок 5.1 Обчислити $f(n_i)$.

Крок 5.2 Якщо n_i не є присутнім ані в списку *Open*, ані в списку *Closed*, то додати його в список *Open*, приєднати до нього оцінку $f(n_i)$ і установити зворотний покажчик на вузол n .

Крок 5.3 Якщо n_i вже є присутнім у списку *Open* або в списку *Closed*, то порівняти нове значення $new = f(n_i)$ з колишнім $old = f(n_i)$. Якщо $old < new$, то припинити обробку нового вузла. Якщо $new < old$, замінити новим вузлом колишній у списку, причому, якщо колишній вузол був у списку *Closed*, перенести його в список *Open*.

Крок 6 Перейти до кроку 2.

Всі методи A^* є припустимими. Пошук A^* є оптимальним, за умови, що $h(n)$ являє собою припустиму евристичну функцію, тобто за умови, що $h(n)$ ніколи не переоцінює вартість досягнення мети.

Метод A^* з ітеративним поглибленням (Iterative-Deepening A^{*} – IDA^{*}) створений для скорочення потреб у пам'яті для пошуку A^* на основі ідеї ітеративного поглиблення в контексті евристичного пошуку. Основне розходження між IDA^{*} і стандартним методом ітеративного поглиблення полягає в тому, що застосовуваною умовою зупинення розгортання слугує f -вартість ($g+h$), а не глибина; на кожній ітерації цим зупинним значенням є мінімальна f -вартість будь-якого вузла, що перевищує зупинне значення, досягнуте в попередній ітерації. Метод IDA^{*} є практично застосовним для рішення багатьох задач з одиничними вартостями етапів і дозволяє уникнути істотних витрат, пов'язаних з підтримкою відсортованої черги вузлів.

Метод IDA^{*} є небагато менш ефективним у порівнянні з методом RBFS. Обидва ці методи піддаються потенційному експонентному збільшенню складності, пов'язаної з пошуком у графах, оскільки вони не дозволяють визначати наявність повторюваних станів, відмінних від тих, котрі знаходяться в поточному шляху. Тому дані методи здатні багато разів досліджувати один і той самий стан. Методи IDA^{*} та RBFS страждають від того недоліку, що в них використовується занадто мало пам'яті. Між ітераціями IDA^{*} зберігає тільки одне – поточну межу f -вартості. RBFS зберігає в пам'яті більше інформації, але кількість використовуваної в ньому пам'яті вимірюється лише значенням $O(bd)$: навіть якби було доступно більше пам'яті, RBFS не здатний нею скористатися.

4.8 Приклади виконання завдань

 **Приклад 1.** Нехай відомо, що у пацієнта: температура тіла є нормальнюю (із коефіцієнтом впевненості 1), місце болю – у животі (із коефіцієнтом впевненості 0,8), вид болю – слабкий (із коефіцієнтом впевненості 0,6) та відомо, що правило позитивного сценарію з попереднього прикладу, характеризується коефіцієнтом впевненості 0,9. Визначити коефіцієнт впевненості консеквента (наслідку) правила для пацієнта на основі стенфордської теорії коефіцієнта впевненості.

Коефіцієнти впевненості посилок антецедента з умови прикладу: $CF(X_1) = 1$, $CF(X_2) = 0,8$, $CF(X_3) = 0,6$.

Коефіцієнт упевненості першої та другої посилок антецедента визначаємо з відповідної формули: $CF(X_1, X_2) = 1$.

Коефіцієнт упевненості $CF(X_1, X_2)$ та третьої посилки визнаємо з відповідної формули: $CF(C) = CF(X_1, X_2, X_3) = 1$.

Коефіцієнт впевненості правила з умови прикладу: $CF(R) = 0,9$.

Коефіцієнт упевненості консеквента $CF(C) = CF(X, Y) CF(R) = 0,9$.

 **Приклад 2.** За допомогою правила Байєса оцінити імовірність діагнозу «гострий апендицит» за наявності симптому «підвищена температура», якщо відомо, що апріорна імовірність діагнозу «апендицит» – 0,001, апріорна імовірність наявності симптому «підвищена температура» – 0,1, а умовна імовірність наявності симптому «підвищена температура» за наявності діагнозу «апендицит» – 0,8.

Позначимо діагноз «апендицит» як подію A , а симптом «підвищена температура» як подію B . Тоді $P(A) = 0,001$, $P(B) = 0,1$, $P(B|A) = 0,8$. Необхідно знайти $P(A|B)$.

Використовуючи правило Байєса $P(A|B) = P(B|A) P(A) / P(B)$, отримаємо: $P(A|B) = 0,8 \cdot 0,001 / 0,1 = 0,008$.

 **Приклад 3.** Нехай $E = \{x_1, x_2, x_3, x_4, x_5\}$, $M = [0, 1]$; A – нечітка множина, для якої $\mu_A(x_1) = 0,3$; $\mu_A(x_2) = 0$; $\mu_A(x_3) = 1$; $\mu_A(x_4) = 0,5$; $\mu_A(x_5) = 0,9$. Тоді A можна подати у вигляді: $A = \{0,3|x_1; 0|x_2; 1|x_3; 0,5|x_4; 0,9|x_5\}$ або $A = 0,3/x_1 + 0/x_2 + 1/x_3 + 0,5/x_4 + 0,9/x_5$ або $A = \{0,3|x_1; 0|x_2; 1|x_3; 0,5|x_4; 0,9|x_5\}$ або $A = \{<0,3; x_1>; <0; x_2>; <1; x_3>; <0,5; x_4>; <0,9; x_5>\}$ або

$$A = \begin{vmatrix} & 1 & 2 & 3 & 4 & 5 \\ 1 & | & | & | & | & | \\ ,3 & | & | & | & | & ,9 \end{vmatrix} \text{ або } A = \begin{vmatrix} ,3 & & & ,5 & ,9 \\ 1 & | & | & | & | \\ & 2 & & 3 & 4 \\ & | & | & | & | \\ & ,3 & & ,5 & ,9 \end{vmatrix}$$

Тут знак «+» не є позначенням операції додавання, а має сенс об'єднання.

 **Приклад 4.** Нехай $E = \{1, 2, 3, \dots, 100\}$ і відповідає поняттю «вік», тоді нечітка множина «молодий», може бути визначена у такий спосіб:

$$\mu_{\text{молодий}}(x) = \begin{cases} 1, x \in [1, 25] \\ 1 + (0,2(x - 25))^{-2}, x > 25. \end{cases}$$

Нечітка множина «молодий» на універсальній множині $E' = \{\text{Іваненко, Петренко, Сидоренко, ...}\}$ задається за допомогою функції належності $\mu_{\text{молодий}}(x)$ на $E = \{1, 2, 3, \dots, 100\}$ (вік), названої стосовно E' функцією сумісності, при цьому: $\mu_{\text{молодий}}(\text{Сидоренко}) = \mu_{\text{молодий}}(x)$, де x – вік Сидоренка.

 **Приклад 5.** Нехай ми маємо нечіткі множини $A = \{0,1|1; 0,5|2; 1,0|3; 0,8|3,2\}$ та $B = \{0,6|1; 0,1|2; 0,1|3\}$. Визначимо їхні властивості та результати операцій над ними. У якості скалярного значення будемо використовувати число $V_1 = 3$, або, за потреби, $V_1^{-1} \approx 0,3$.

Унарні операції з A : доповнення: $\{0,9|1; 0,5|2; 0,2|3,2\}$;

Унарні операції з B : доповнення: $\{0,4|1; 0,9|2; 0,9|3\}$;

Бінарні операції з A та B : об'єднання: $\{0,6|1; 0,5|2; 1,0|3; 0,8|3,2\}$; перетинання: $\{0,1|1; 0,1|2; 0,1|3\}$;

 **Приклад 6.** Нехай $X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2, y_3, y_4\}$, $M = [0, 1]$. Нечітке відношення $R = X R Y$ може бути задане таблицею:

R	y_1	y_2	y_3	y_4
x_1	0	0	0,1	0,3
x_2	0	0,8	1	0,7
x_3	1	0,5	0,6	1

 **Приклад 7.** Нехай $X = Y = (-\infty, \infty)$, тобто множина усіх дійсних чисел. Відношення $x \succ y$ (x забагато більший ніж y) можна задати функцією належності:

$$\mu_R = \begin{cases} 0, & \text{якщо } x \leq y, \\ \frac{1}{1+(x-y)^2}, & \text{якщо } y > x. \end{cases}$$

 **Приклад 8.** Відношення R , для котрого $\mu_R(x,y) = e^{-k(x-y)^2}$ при достатньо великих k можна інтерпретувати так: « x та y близькі одне до одного числа».

 **Приклад 9.** Нехай нечіткі відношення R_1 та R_2 задаються такими таблицями:

R_1	y_1	y_2	y_3
x_1	0,1	0,7	0,4
x_2	1	0,5	0

R_2	z_1	z_2	z_3	z_4
y_1	0,9	0	1	0,2
y_2	0,3	0,6	0	0,9
y_3	0,1	1	0	0,5

Тоді нечітке відношення R_1

• R_2 може бути задане таблицею:

$R_1 \bullet R_2$	z_1	z_2	z_3	z_4
x_1	0,3	0,6	0,1	0,7
x_2	0,9	0,5	1	0,5

При цьому:

$$\begin{aligned}
 \mu_{R_1 \bullet R_2}(x_1, z_1) &= \\
 &= [\mu_{R_1}(x_1, y_1) \cap \mu_{R_2}(y_1, z_1)] \cup [\mu_{R_1}(x_1, y_2) \cap \mu_{R_2}(y_2, z_1)] \cup \\
 &\cup [\mu_{R_1}(x_1, y_3) \cap \mu_{R_2}(y_3, z_1)] = \\
 &= (0,1 \cap 0,9) \cup (0,7 \cap 0,3) \cup (0,4 \cap 0,1) = \\
 &= 0,1 \cup 0,3 \cup 0,1 = 0,3; \\
 \mu_{R_1 \bullet R_2}(x_1, z_2) &= (0,1 \cap 0) \cup (0,7 \cap 0,6) \cup (0,4 \cap 1) = \\
 &= 0 \cup 0,6 \cup 0,4 = 0,6.
 \end{aligned}$$

У цьому прикладі спочатку використано «аналітичний» спосіб композиції відношень R_1 та R_2 , тобто i -й рядок R_1 «збільшується» на j -й стовпець R_2 з використанням операції \cap , потім отриманий результат «згортається» з використанням операції \cup в $\mu(x_i, z_j)$.



Приклад 10. Нехай ми маємо нечітку множину для поняття «чоловік середнього росту»: $A = \{0|155, 0,1|160, 0,3|165, 0,8|170, 1|175, 1|180, 0,5|185, 0|190\}$. Проведемо дефаззифікацію нечіткої множини «чоловік середнього росту» за методом центра тяжіння:

$$a = (0 \cdot 155 + 0,1 \cdot 160 + 0,3 \cdot 165 + 0,8 \cdot 170 + 1 \cdot 175 + 1 \cdot 180 + 0,5 \cdot 185 + 0 \cdot 190) / (0 + 0,1 + 0,3 + 0,8 + 1 + 1 + 0,5 + 0) = 175,4.$$



Приклад 11. Спадне нечітке логічне виведення. Нехай задана модель діагностики системи, що складається з двох передумов і трьох висновків: $X = \{x_1, x_2\}$, $Y = \{y_1, y_2, y_3\}$, а, матриця нечітких відношень має вигляд:

$$R = \begin{bmatrix} 0,8 & 0,2 & 0,3 \\ 0,7 & 0,4 & 0,5 \end{bmatrix}.$$

Припустимо, що в результаті діагностики системи були отримані такі висновки: $Y = 0,8/y_1 + 0,2/y_2 + 0,3/y_3$. Необхідно знайти передумови, що до цього призвели: $A = a_1/x_1 + a_2/x_2$.

З урахуванням конкретних даних відношения між передумовами та висновками будуть подані в транспонованому вигляді у такий спосіб:

$$\begin{bmatrix} 0,8 \\ 0,2 \\ 0,3 \end{bmatrix} = \begin{bmatrix} 0,8 & 0,7 \\ 0,2 & 0,4 \\ 0,3 & 0,5 \end{bmatrix} \bullet \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}.$$

При використанні *max-min*-композиції останнє співвідношення перетвориться до вигляду: $0,8 = (0,8 \cap a_1) \cup (0,7 \cap a_2)$, $0,2 = (0,2 \cap a_1) \cup (0,4 \cap a_2)$, $0,3 = (0,3 \cap a_1) \cup (0,5 \cap a_2)$. При вирішенні даної системи помітимо, що в першому рівнянні другий член правої частини не впливає на ліву частину: $0,8 = 0,8 \cap a_1$, $a_1 \geq 0,8$. З другого рівняння одержимо: $0,2 = 0,4 \cap a_2$, $a_2 \leq 0,2$. Отримане рішення задовільняє третьому рівнянню. У такий спосіб: $0,8 < a_1 < 1$, $0 < a_2 < 0,2$.

При вирішенні практичних задач можуть одночасно використовуватися різні правила композиції нечітких виведень. Сама схема виведень може бути багатокаскадною. В даний час загальних методів вирішення подібних задач, очевидно, не існує.

 **Приклад 12.** Створити на мові пакету *MATLAB* м-функцію, що реалізує колоколоподібну функцію належності $\mu(x) = \frac{1}{1 + ((x - b)/a)^2}$.

```
function mu=bellmf(x, params)
%bellmf - колоколоподібна функція належності;
%x - вхідний вектор;
%params(1) - коефіцієнт концентрації (>0);
%params(2) - координата максимуму.
a=params(1); b=params(2);
mu=1./ (1 + ((x - b)/a).^2);
```

4.9 Контрольні питання

1. У чому полягає відмінність знань від даних?
2. Які властивості мають знання?
3. Що таке інформація?
4. Що таке факт?
5. Ієрархія способів подання інформації.
6. Що таке дані, знання, метазнання?
7. Класифікація знань.
8. Особливості знань.
9. Поняття екстенсіоналу та інтенсіоналу.
10. Що таке система, заснована на знаннях?
11. Актуальність інтелектуальних систем, заснованих на знаннях.
12. Ієрархія рівнів систем, заснованих на знаннях.
13. Що таке експертна система?
14. Які властивості мають експертні системи?
15. Класифікація експертних систем.
16. Життєвий цикл та методологія розробки експертних систем.
17. Хто такий експерт, інженер зі знань, програміст, користувач?
18. Що таке засіб побудови експертної системи?
19. У чому полягає концепція «швидкого прототипу»?
20. У чому полягають структура та функціонування експертної системи?
21. Що таке база знань, факт, правило, робоча пам'ять, машина логічного виведення, оболонка експертної системи, інтерфейсна підсистема, підсистема набуття знань, редактор бази знань, підсистема пояснень?

22. Що таке динамічна експертна система?
23. Які ви знаєте режими роботи експертної системи?
24. Які ви знаєте переваги і недоліки експертних систем?
25. Що таке логічне виведення?
26. Які ви знаєте типи логічного виведення?
27. Що таке дедукція, індукція, абдукція?
28. У чому полягає звичайне (булеве) логічне виведення?
29. У чому полягає прямий метод виведення?
30. У чому полягає зворотний метод виведення?
31. У чому полягає метод Джона Стюарта Міля?
32. Що таке монотонна логіка, немонотонна логіка?
33. У чому полягають Стенфордська теорія коефіцієнта впевненості, теорія Демпстера–Шафера, метод виведення Байеса, метод виведення Нейлора?
34. Що таке пошук у просторі станів?
35. Що таке граф ТА/АБО?
36. Які ви знаєте стратегії пошуку в просторі станів?
37. У чому полягає метод пошуку з поверненнями?
38. Продуктивність та ефективність методів пошуку.
39. Які ви знаєте стратегії неінформованого (сліпого) пошуку?
40. У чому полягає метод породження і перевірки?
41. У чому полягає метод пошуку в ширину?
42. У чому полягають метод пошуку в глибину, метод пошуку з поверненнями, метод пошуку з обмеженням глибини, метод пошуку в глибину з ітеративним поглибленням?
43. Які ви знаєте стратегії інформованого (евристичного) пошуку?
44. У чому полягає метод пошуку зі сходженням до вершини?
45. Що таке оцінна функція?
46. У чому полягають метод пошуку за першим найкращим збігом, метод рекурсивного пошуку за першим найкращим збігом, методи пошуку A^* ?
47. Що таке пояснення процесу прийняття рішень?
48. Які ви знаєте види пояснень?
49. Які ви знаєте методи формування пояснень?
50. Склад розроблювачів експертної системи, роль і задачі кожного з них.
51. Які властивості предметної області (об'єкта автоматизації) є передумовою для створення експертної системи.

52. Області людської діяльності в яких застосовуються експертні системи.

53. Чим експертні системи відрізняються від звичайних програмних додатків та типових програм штучного інтелекту?

54. Чи може програма, яка не використовує методи штучного інтелекту, мати такі ж властивості як експертна система?

55. У чому різниця між експертною системою та системою, заснованою на знаннях?

56. Що таке: нечітка множина, чітка підмножина, функція належності, нечітка підмножина, множина належностей, нечітка змінна, лінгвістична змінна, терм-множина, терм, нечіткий терм?

57. Які існують методи побудови та запису функцій належності?

58. Дайте визначення операцій над нечіткими множинами: до-повнення, об'єднання, перетинання.

59. Що таке: нечітке n -арне відношення, нечітке відношення на множині?

60. Які існують способи задавання нечітких відношень?

61. Як визначаються операції над нечіткими відношеннями?

62. Як визначаються згортки: ($\max\text{-}*$)-композиція, ($\min\text{-}*$)-композиція та ($\text{sum}\text{-prod}$)-композиція? Які властивості вони мають?

63. Що таке фаззіфікація та дефаззіфікація? Дайте визначення основних методів дефаззіфікації: середній з максимальних, найбільший з максимальних, найменший з максимальних, центр тяжіння, метод медіани, висотна дефаззіфікація.

64. Що таке нечітка база знань, правила Мамдані, правила Такагі-Сугено, антецедент і консеквент правила, імплікація, нечітке логічне виведення, нечіткий «modus ponens»?

65. Які існують етапи та з яких компонентів складаються системи нечіткого виведення? На які типи класифікують системи нечіткого виведення?

66. У чому полягають висхідний та низхідний методи нечіткого виведення? Опишіть основні етапи методів Мамдані, Цукамото, Сугено та Ларсена.

4.10 Практичні завдання

✉ Завдання 1. Написати реферат на одну з таких тем.

1. Застосування експертних систем.
2. Методи логічного виведення.

3. Методи пошуку у просторі станів та особливості їхнього застосування.
4. Байесівські мережі виведення.
5. Методи пояснення рішень в експертних системах.
6. Принципи аналізу предметної області.
7. Моделі подання знань для побудови експертних систем.
8. Мови для опису та обробки знань.
9. Сучасні оболонки експертних систем.
10. Методи та засоби видобування знань з даних.
11. Методи витягу знань з експертів.
12. Методи визначення функцій належності нечітких множин.
13. Операції над нечіткими множинами та відношеннями.
14. Закони чіткої та нечіткої логік.
15. Узагальнення нечітких операцій.
16. Нечіткі числа та операції з ними.
17. Методи нечіткого виведення.
18. Методи дефазифікації.
19. Нечіткий кластерний аналіз.

 **Завдання 2.** Нехай V – номер варіанта студента. Визначимо:

$$V_1 = \begin{cases} 4V, & \text{якщо } V < 5; \\ V, & \text{якщо } 5 \leq V \leq 10; \\ \text{round}(0,3V), & \text{якщо } V > 10, \end{cases}$$

де $\text{round}(x)$ – функція округлення.

Сформувати нечіткі множини:

$$A = \left\{ \left. < \frac{k}{V_1} \right| k \right> \}, \quad B = \left\{ \left. < \frac{2k}{V_1} \right| 0,5k \right> \}, \quad k = 0, 1, \dots, V.$$

 **Завдання 3.** Визначити результати виконання над нечіткими множинами A та B операцій: доповнення, об'єднання, перетинання.

 **Завдання 4.** Придумати приклад лінгвістичної змінної та варіантів її значень. Для лінгвістичної змінної запропонувати нечіткі змінні. Навести приклади можливих значень нечітких змінних. Визначити терм-множину лінгвістичної змінної.

 **Завдання 5.** Побудувати декілька разів графіки функцій належності у залежності від номеру варіанту із різними значеннями па-

раметрів. Параметри функцій належності задати самостійно. Дослідити, як впливають значення параметрів на зміну значення функцій належності. Завдання виконати у пакеті MATLAB із використанням функцій модуля Fuzzy Logic Toolbox.

V	Функції		V	Функції	
1.	dsigmf	gbellmf	14.	psigmf	gaussmf
2.	dsigmf	zmf	15.	psigmf	pimf
3.	dsigmf	trimf	16.	sigmf	dsigmf
4.	gauss2mf	gbellmf	17.	sigmf	pimf
5.	gauss2mf	trimf	18.	sigmf	trimf
6.	gauss2mf	trapmf	19.	smf	gauss2mf
7.	gaussmf	gauss2mf	20.	smf	gbellmf
8.	gaussmf	pimf	21.	trapmf	gaussmf
9.	gaussmf	trapmf	22.	trimf	gauss2mf
10.	gbellmf	pimf	23.	trimf	gbellmf
11.	gbellmf	smf	24.	zmf	smf
12.	pimf	gaussmf	25.	zmf	dsigmf
13.	pimf	sigmf	26.	zmf	pimf

 **Завдання 6.** Придумати приклад нечіткого відношення $A \ R \ B$, а також його доповнення \bar{R} . Визначити результати операцій:

$$C = R \cup \bar{R}, \ D = R \cap \bar{R}, \ E = (C \cap D) \cup R.$$

 **Завдання 7.** Для нечітких множин A та B відповідно до V знайти чіткі значення A^* та B^* , використовуючи методи дефазіфікації: середній з максимальних, найбільший з максимальних, найменший з максимальних, центр тяжіння, метод медіани, висотна дефазіфікація. Порівняти результати методів для кожної нечіткої множини.



4.11 Література до розділу

Питання побудови інтелектуальних систем, заснованих на знаннях висвітлено в [2, 4, 5, 6, 15, 19]. Методи чіткого виведення наведено в [2, 4, ,5, 6, 18]. Нечітке виведення та основи нечіткої логіки описано в [5, 6, 12, 13, 16, 20].

РОЗДІЛ 5

ПРОГРАМНІ ЗАСОБИ ДЛЯ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

Побудова інтелектуальної системи, як правило, вимагає тривалих експериментів з підбором описів і форматів вхідних даних, архітектури і методів побудови моделей та їх алгоритмічних реалізацій, потребуючи при цьому багаторазової переробки програмного коду, розробки інтерфейсу і часто навіть зміни мови програмування. Це суттєво ускладнює і здорожчує процес побудови інтелектуальних систем.

Тому дуже популярним у середовищі фахівців з інтелектуальних систем є підхід, що передбачає отримання «швидкого прототипу» шляхом перенесення якомога більшої кількості експериментів у середовища спеціалізованих пакетів, що мають добре реалізовані засоби введення-виведення, обробки і візуалізації даних та широкий набір бібліотечних функцій, які реалізують найвідоміші методи обробки даних. При цьому акцент робиться на алгоритмічних аспектах, а не на особливостях апаратно-програмної платформи. Після отримання «швидкого прототипу» – програмної моделі вирішення задачі вона автоматично або уручну перекладається мовою найкращого для розробника середовища розробки. При цьому вже здійснюється вирішення платформозалежних питань.

Одним з найбільш використовуваних програмних кросплатформених засобів побудови «швидких прототипів» інтелектуальних систем обробки даних є пакет Matlab компанії Mathworks.

Matlab – широко відома і перевірена часом система автоматизації математичних розрахунків, побудована на розширеному поданні та застосуванні матричних операцій. Це знайшло відображення в назві системи – MATrix LABoratory – «матрична лабораторія». Однак синтаксис мови програмування системи продуманий настільки ретельно, що ця орієнтація майже не відчувається тими користувачами, яких не цікавлять безпосередньо матричні обчислення.

Коди програм у системі Matlab пишуться мовою високого рівня, яка досить зрозуміла для користувачів помірної кваліфікації в області програмування. Пакет Matlab є типовим інтерпретатором. Це означає, що кожна інструкція програми розпізнається і відразу виконується, що полегшує забезпечення діалогового режиму спілкування із системою.

Важливими перевагами системи є її відкритість і розширюваність. Більшість команд і функцій системи реалізовані у вигляді текстових m-файлів (з розширенням «.m»), причому усі файли доступні для модифікації. Користувачеві дана можливість створювати не тільки окремі файли, але і бібліотеки файлів для реалізації специфічних задач.

5.1 Подання і введення-виведення даних

Для побудови інтелектуальних моделей за прецедентами потрібно отримати навчальну вибірку даних-спостережень. Цю вибірку можна ввести до ЕОМ на зовнішніх носіях у вигляді файлів відповідних форматів або за допомогою спеціальних редакторів уручну. Пакет Matlab має широкий спектр функцій для зчитування даних як з файлів на диску, так і безпосередньо з вимірювального обладнання.

Змінні і визначення нових функцій у системі Matlab зберігаються в особливій області пам'яті, що називається робочою областю. Її вміст можна переглянути і відредактувати у вікні спеціального браузера робочої області – *Workspace Browser*. Для виводу вмісту об'єкта досить виділити його ім'я за допомогою миші і клацнути на кнопці *Open* (Відкрити). Об'єкт можна відкрити і подвійним щицликом на його імені в списку. Відкриється вікно редактування масиву *Array Editor*.

Вікно редактування масиву (матриці) дає зручний доступ для редактування будь-якого елемента матриці за правилами, прийнятими при роботі з електронними таблицями. Основне з них – швидкий доступ до будь-якого елемента матриці. Можна також змінювати тип значень елементів, обираючи його зі списку, наданого меню *Numeric format* (Формат чисел). У вікні також виводиться дані про число рядків і стовпців матриці.

Слід зазначити, що перегляд робочої області можливий і в командному режимі, без звертання до браузера *Workspace Browser*. Команда *who* виводить список визначених змінних, а команда *whos* – список змінних із вказівкою їхнього розміру й обсягу займаної пам'яті.

Зауваження. З урахуванням орієнтації цієї книги на вирішення задач побудови інтелектуальних систем, що здатні навча-

тися за прецедентами, окрім зазначимо, що, нажаль, у пакеті Matlab функції різних модулів мають різне подання вибірки даних у пам'яті (наприклад у модулі *Neural Network Toolbox* екземпляри розташовуються у стовпцях, а ознаки – у рядках, а у модулях *Fuzzy Logic Toolbox* і *Statistics Toolbox* екземпляри розташовуються у рядках, а ознаки – у стовпцях). При використанні як функцій, так і засобів графічного інтерфейсу користувача необхідно обачливо ставитися для вибору змінних та перевіряти коректність їх використання. У разі невідповідності ситуацію легко виправити, використовуючи за потреби операцію транспонування, що для матриці X реалізується викликом `X'`.

Найбільш широко застосованою функцією для завантаження даних з файлу до робочої області є `load('fname')` – здійснює завантаження раніше збережених у файлі `fname.mat` визначень зі специфікаціями на місці крапок, подібно описаним нижче для команди `save` (включаючи ключ `-mat` для завантаження файлів з розширенням `.mat` звичайного бінарного формату, що використовується за замовченням). Якщо функція `load` використовується в ході проведення сесії, то відбудеться заміна поточних значень змінних тими значеннями, що були збережені в `mat`-файлі, що читається.

Для збереження даних із середовища Matlab у файл на диску використовується функція `save`:

`save('fname')` – усі змінні робочої області записуються у файлі бінарного формату з ім'ям `«fname.mat»`;

`save('fname', 'X', 'Y')` – до файлу бінарного формату з ім'ям `«fname.mat»` записуються тільки значення змінних `X` та `Y`.

Для уточнення формату запису файлів після списку змінних можна вказувати ключі:

`'-mat'` – двійковий Mat-формат, що використовується за замовченням;

`'-ascii'` – ASCII-формат одиничної точності (8 цифр);

`'-ascii', '-double'` – ASCII-формат подвійної точності (16 цифр);

`'-ascii', '-double', '-tabs'` – формат з роздільником і мітками табуляції;

`'-append'` – додавання в існуючий Mat-файл.

5.2 Кластер-аналіз даних

Пакет Matlab містить різноманітні засоби для чіткого кластерного аналізу у модулі Statistics Toolbox та нечіткого кластер-аналізу – у модулі Fuzzy Logic Toolbox.

Функція $[idx, c, sumd, D] = kmeans(X, k)$ здійснює чіткий кластер-аналіз матриці $n \times p$ даних X (рядки відповідають екземплярам, а стовпці – ознакам), розбиваючи її на k кластерів із використанням Евклідової відстані за замовчуванням, де IDX – вектор $n \times 1$ номерів кластерів до яких належать екземпляри, C – матриця $k \times p$ координат центрів кластерів, $sumd$ – вектор $1 \times k$ внутрікластерних сум відстаней від екземплярів до центрів кластерів, D – матриця $n \times k$ відстаней між екземплярами та центрами кластерів. Це ітеративне розбиття мінімізує суму за усіма кластерами внутрікластерних сум відстаней від екземплярів до центрів кластерів.

Функція $T = clusterdata(X, Name, Value)$ здійснює агломеративний кластер-аналіз даних, де T – вектор з m елементів, що містять номери кластерів до яких відносяться екземпляри, X – матриця з двох і більше рядків (рядки містять екземпляри, а стовпці – ознаки), $Name-Value$ – набір пар аргументів, що налаштовують параметри кластер-аналізу. $Name$ задає назву параметра і має бути в одинарних лапках, а тип $Value$ залежить від $Name$: ‘cutoff’ – максимальна кількість кластерів, ‘distance’ – спосіб визначення відстані (‘euclidean’ – Евклідова, ‘seuclidean’ – стандартизована Евклідова, ‘cityblock’ – міських кварталів, ‘minkowski’ – Мінковського, ‘chebychev’ – Чебишева, ‘mahalanobis’ – Махалонобіса, ‘cosine’ – кутова відстань між точками як векторами, ‘correlation’ – вибіркова кореляція, ‘spearmann’ – вибіркова рангова кореляція Спірмена, ‘hamming’ – Хеммінга, ‘jaccard’ – коефіцієнт Жаккарда, ‘linkage’ – метод (критерій) визначення зв’язування екземплярів (‘average’ – середнє, ‘centroid’ – центроїдне, ‘complete’ – повне, ‘median’ – медіанне, ‘single’ – одинарне, ‘ward’ – Уорда, ‘weighted’ – зважене), ‘maxclust’ – максимальна кількість кластерів.

Функція $subclust$ визначає координати центрів кластерів шляхом чіткої кластеризації зі зменшенням кількості кластерів. Вона знаходить оптимальну точку даних для визначення центра кластера, ґрунтуючись на щільності оточення точок даних. Усі точки даних у межах відстані $RADI$ до цієї точки видаляються, щоб визначити наступний кластер даних та його центр. Цей про-

цес повторюється поки усі дані не знаходяться у межах відстані RADII до центра кластера.

[C] = SUBCLUST (X, RADII) кластеризує точки даних $S \times N$ матриці X, де S – кількість точок даних, N – кількість координат точок даних, RADII – значення між 0 та 1, що визначає розмір кластера в кожному з вимірювань даних, приймаючи, що дані знаходяться у межах діапазону [0, 1] (Встановлення меншого радіуса кластера буде звичайно створювати більше менших за розміром кластерів. Коли RADII є скаляром, він застосовується до усіх вимірів даних. Коли RADII є вектором, він має окреме значення для кожного виміру даних), та повертає центри кластерів як рядки матриці C, що має розмір $V \times N$, де V – кількість кластерів.

[C] = SUBCLUST (..., XBOUNDS) також визначає матрицю XBOUNDS, розміром $2 \times N$, що використовується для нормалізації даних X у діапазон [0, 1]. Кожний стовпець XBOUNDS містить мінімальні та максимальні значення для відповідної розмірності даних. Якщо XBOUNDS – порожня матриця або не використовується, тоді за замовчуванням використовуються мінімальні та максимальні значення даних X.

[C] = SUBCLUST (..., OPTIONS) визначає вектор для зміни значень за замовчуванням параметрів алгоритму: OPTIONS(1) – коефіцієнт, що використовується для множення на значення RADII для визначення осередку центру кластера, у межах якого існування інших центрів кластерів заборонено; OPTIONS(2) – коефіцієнт прийняття, що встановлює потенціал як частку потенціалу центра першого кластера, вище якої інша точка даних буде прийнята як центр кластера; OPTIONS(3) – коефіцієнт відхилення, що встановлює потенціал як частку потенціалу центра першого кластера, нижче якої інша точка даних буде відхиlena як центр кластера; OPTIONS(4) – ознака відображення поточної інформації, якщо не встановлена як 0. Значеннями вектора OPTIONS за замовчуванням є [1.25 0.5 0.15 0].

Функція [CENTER, U, OBJ_FCN] = fcm(DATA, N_CLUSTER, OPTIONS) здійснює нечітку кластеризацію на основі методу нечітких c-середніх, де N_CLUSTER – кількість кластерів в наборі даних масиву DATA, який має розміри $S \times N$, S – кількість точок даних, N – кількість координат точок, CENTER – матриця з координатами центрів кластерів (кластери містяться у рядках, ознаки – у стовпцях), U – мат-

риця функції належності, що містить рівні належності кожної точки масиву DATA до кожного кластера, OBJ_FCN – значення цільової функції для центрів кластерів, OPTIONS – необов'язковий параметр, що задає вектор опцій для процесу кластеризації: OPTIONS(1) – експонента для матриці U (за замовчуванням: 2.0), OPTIONS(2) – максимальна кількість ітерацій (за замовчуванням – 100), OPTIONS(3) – мінімально прийнятне покращення цільової функції (за замовчуванням – 10^{-5}), OPTIONS(4): ознака відображення проміжних результатів (за замовчуванням – 1).

На кожній ітерації цільова функція мінімізується для знаходження кращого розташування кластерів. Процес кластеризації зупиняється, коли максимально прийнятна кількість ітерацій є досягнутою, або коли покращення цільової функції між двома послідовними ітераціями зміна є меншою ніж мінімально прийнятний приріст.

5.3 Відбір інформативних ознак

У пакеті Matlab відбір інформативних ознак можна здійснити за допомогою модуля Statistics Toolbox.

Функція [inmodel, history] = sequentialfs(fun, X, y, param1, val1, param2, val2,...) виділяє підмножину ознак (логічний вектор) inmodel з матриці даних X, які найкращим чином апроксимують дані в у шляхом послідовного додавання ознак, що відбувається доти, поки не припиниться покращення точності апроксимації.

Рядки в X відповідають екземплярам, а стовпці – ознакам, у – вектор-стовпець віхідних значень для кожного екземпляра, param1, val1, param2, val2,... – набір параметрів, що регулюють процес відбору ознак, history – скалярна структура (містить такі поля: Crit – вектор, що містить значення критерія, розраховані на кожному кроці, In – логічна матриця, де у рядках розташовані ознаки, обрані на ітерації з номером, що відповідає номеру рядка), fun – посилання на функцію, що визначає критерій відбору ознак та критерій зупинення, у форматі: criterion = fun(XTRAIN, ytrain, XTEST, ytest), де XTRAIN та ytrain – відповідно, масиви входів та виходу для навчальної вибірки, XTEST та ytest – відповідно, масиви входів та виходу для тестової вибірки, яка повертає скалярне значення критерію criterion (при цьому функція fun використовує масиви XTRAIN та

`ytrain` для навчання моделі, після чого емулює її (виконує розпізнавання на основі навченої моделі), подаючи на її входи `XTEST`, та у результаті повертає значення помилки або втрат для розрахованих значень відносно `ytest`.

Параметри (не обов'язково) задаються набором пар «назва»–«значення»: ‘cv’ – метод перевірки для розрахунку критерія для кожного піднаборку ознак, ‘mcdeps’ – кількість випадкових повторів при крос-перевірці, ‘direction’ – напрям пошуку (за замовчуванням – ‘forward’ – пошук з додаванням ознак, значення ‘backward’ дозволяє здійснювати відбір ознак із виключенням ознак, доки критерій збільшується), ‘keepin’ – логічний вектор, що визначає ознаки, що мусять бути включені, ‘keepout’ – логічний вектор, що визначає ознаки, що мусять бути виключені, ‘nfeatures’ – кількість ознак, на якій потрібно закінчити пошук, ‘options’ – структура, що містить параметри для настроювання ітеративного пошуку (`Display` – обсяг інформації, що відображується на екрані, `MaxIter` – максимальна кількість дозволених ітерацій пошуку, `TolFun` – чутливість до значень цільової функції (за замовчуванням: 10^{-6} для ‘direction’ – ‘forward’ та 0 – для ‘direction’ – ‘backward’), `TolTypeFun` – тип функції оцінювання толерантності).

Для кожної комбінації ознак `sequentialfs` здійснює 10-кратну перевірку, ітеративно викликаючи `fun` з різними підмножинами `X` та `y`, `XTRAIN` та `ytrain`, а також `XTEST` та `ytest`.

5.4 Побудова моделей прийняття рішень

Побудова моделей прийняття рішень за спостереженнями-прецедентами може бути реалізована різними засобами пакету Matlab.

Модуль Neural Network Toolbox містить функції для побудови та навчання штучних нейронних мереж, а також засіб для створення нейромереж у графічному режимі користувача `nnptool`.

Модуль Fuzzy Logic Toolbox містить функції та графічні засоби користувальницього інтерфейсу для створення нейро-нечітких моделей як за преседентами, так і на основі наявних експертних знань продукційного типу. Фактично цей модуль дозволяє будувати гібридні інтелектуальні системи, що поєднують експертні знання та знання, що витягаються з наборів даних.

Модуль Statistics Toolbox містить засоби для вирішення завдань побудови моделей прийняття рішень на основі дискримінантного аналізу, а також дерев розв'язків.

Перелік функцій модулів постійно розширюється з виходом нових версій пакету Matlab. У середовищі пакету доцільно використовувати команди `help` та `demos` для ознайомлення з наявним функціоналом засобів для побудови інтелектуальних систем.

5.5 Приклади виконання завдань

 *Приклад 1.* Виконання послідовного відбору ознак з нарощуванням у пакеті Matlab.

```
% завантажуємо стандартний приклад даних
% для класифікації ірисів Фішера
load fisheriris;
% створюємо на основі завантажених даних
% та випадкових чисел матриці вхідних
X = randn(150,10); X(:,[1 3 5 7])= meas;
% і вихідних даних
y = species;
% створюємо об'єкт класу, що визначає випадкове розбиття
% для k-кратної перевірки (k=10) за спостереженнями у.
% Розбиття розподіляє спостереження з утворенням k
% випадковим чином обраних підвибірок приблизно
% однакового розміру, що не перетинаються
c = cvpartition(y,'k',10);
% задаємо значення опцій Statistics Toolbox:
% показувати ітерації пошуку в командному вікні
opts = statset('display','iter');
% створюємо посилання на функцію-критерій пошуку,
% яка визначає кількість поелементних неспівпадінь
% значень yt та результатів розпізнавання вибірки Xt
% відносно груп (кластерів) yt у навчальній вибірці Xt
fun = @(XT, yt, Xt, yt)... %sum(~strcmp(yt, classify(Xt, XT, yt,'quadratic'))));
% виконуємо запуск функції відбору ознак
[fs, history] = sequentialfs(fun, X, y,'cv',...
    'options', opts)
```

Результати роботи програми (у дужках – переклад українською мовою):

```
Start forward sequential feature selection:
(Початок послідовного відбору ознак з нарощуванням)
Initial columns included: none
(Початкові стовпці включені: немає)
Columns that can not be included: none
(Стовпці, що не будуть включатися: немає)
(Kрок ..., додано стовпець..., значення критерію...)
```

```

Step 1, added column 7, criterion value 0.04
Step 2, added column 5, criterion value 0.0266667
(У результаті обрано ознаки-стовпці)
Final columns included: 5 7

```

```

fs =
    0   0   0   0   1   0   1   0   0   0
history =
    In: [2x10 logical]
    Crit: [0.0400 0.0267]

history. In
ans =
    0   0   0   0   0   0   1   0   0   0
    0   0   0   0   1   0   1   0   0   0

```

Приклад 2. Використання функції чіткого кластер-аналізу `subclust`.

```

% генеруємо вибірку даних
X1 = 10*rand(50,1); X2 = 5*rand(50,1);
X3 = 20*rand(50,1)-10; X = [X1 X2 X3];
[C] = subclust(X,0.5); % знаходимо центри кластерів

```

Приклад 3. Використання функції чіткого кластер-аналізу `clusterdata`.

```

% створюємо масив даних
X = [gallery('uniformdata',[10 3],12);...
      gallery('uniformdata',[10 3],13)+1.2;...
      gallery('uniformdata',[10 3],14)+2.5];
% виконуємо кластеризацію даних з побудовою дерева
% ієрархії кластерів, де екземпляри групуються
% у три кластери
T = clusterdata(X,'maxclust',3);
% виводимо перелік номерів усіх екземплярів,
% що потрапили до другого кластеру
find(T==2)

```

Приклад 4. Використання функції нечіткого кластер-аналізу `fcm`.

```

data = rand(100,2); % генеруємо вибірку
% виконуємо кластер-аналіз
[center, U, obj_fcn] = fcm(data,2);
% зображення дані на графіку
plot(data(:,1), data(:,2),'o'); hold on;
% знаходимо максимальне значенн належностей
maxU = max(U);

```

```

% Знаходимо точки з найвищим рівнем належності
% до першого кластера
index1 = find(U(1,:) == maxU);
% Знаходимо точки з найвищим рівнем належності
% до другого кластера
index2 = find(U(2,:) == maxU);
% оформлення графіка
line(data(index1,1), data(index1,2),...
'marker','*','color','g');
line(data(index2,1), data(index2,2),...
'marker','*','color','r');
% Зображення центри кластерів на графіку
plot([center([1 2],1)],...
[center([1 2],2)],'*','color','k');

```

? 5.6 Контрольні питання

1. Архітектура та характеристики пакету MATLAB.
2. Вимоги до навчальних вибірок даних.
3. Задача відбору ознак.
4. Критерії оцінювання інформативності ознак на основі евристичного, інформаційного, статистичного та імовірнісного підходів.
5. Оцінювання ознак за допомогою коефіцієнта кореляції знаків, коефіцієнта кореляції Фехнера, кількості інтервалів зміни номера класу.
6. Задача розпізнавання образів. Навчання з учителем.
7. Основні поняття теорії розпізнавання образів.
8. Методи метричної класифікації.
9. Задача кластер-аналізу. Навчання без учителя.
10. Чіткий кластер-аналіз.
11. Нечіткий кластер-аналіз.
12. Використання кластер-аналізу при побудові систем розпізнавання образів.
13. Подібність кластер-аналізу і метричної класифікації.
14. Функції кластер-аналізу у пакеті MATLAB.
15. Чи впливає кількість використаних ознак на швидкість кластер-аналізу?
16. Чи впливає обсяг навчальної вибірки на швидкість кластер-аналізу?
17. Чи впливає обсяг навчальної вибірки на швидкість навчання методу метричної класифікації?

18. Чи впливає репрезентативність навчальної вибірки на точність класифікації екземплярів тестової вибірки?
19. Чи впливає репрезентативність тестової вибірки на точність класифікації екземплярів тестової вибірки?
20. Чи впливає репрезентативність тестової вибірки на точність навчання персептрона за навчальною вибіркою?
21. Чи залежить якість навчання від якості та обсягу навчальної вибірки?
22. Чи повинна навчальна вибірка бути репрезентативною?
23. Чи повинна тестова вибірка бути репрезентативною?
24. Що таке генеральна сукупність, вибірка, екземпляр, ознака?
25. Що таке репрезентативна вибірка даних?

5.7 Практичні завдання

 **Завдання 1. Самостійна робота «Програмні засоби для побудови інтелектуальних систем»**

Мета роботи: освоїти засоби середовища MATLAB для створення на його основі прикладних інтелектуальних систем.

Завдання до роботи.

1. Використовуючи рекомендовану літературу самостійно вивчити основні принципи побудови програм для пакету MATLAB.

2. Розробити на мові для пакету MATLAB функції, що демонструють роботу з матрицями (із використанням матричних та масивних операцій), будують графіки, зберігають змінні на диск. Розроблені функції інтегрувати до графічної форми, яку створити за допомогою *GUI*.

3. Порівняти можливості пакету MATLAB із відомими системами програмування.

Зміст звіту.

1. Мета роботи.
2. Короткі теоретичні відомості.
3. Тексти програм та інтерфейсні форми, розроблені студентом.
4. Вхідні дані та результати роботи програми.
5. Висновки, що відображують результати виконання роботи та їх критичний аналіз.



Завдання 2. Лабораторна робота «Розпізнавання образів на основі метричної класифікації»

Мета роботи: вивчити та засвоїти на практиці метричні методи розпізнавання образів у просторі ознак, навчитися створювати програмні засоби, що реалізують методи метричної класифікації.

Завдання до роботи.

1. Ознайомитися з рекомендованою літературою. На алгоритмічній мові програмування пакету MATLAB написати програму, що реалізує процедури для навчання та емуляції (розпізнавання) за методом еталонів.

2. Згідно з номером студента за журналом для відповідного номера варіанта V сформувати навчальну вибірку $\langle x, y \rangle$ обсягом S екземплярів $x^s, s=1, 2, \dots, S$, що характеризуються N ознаками $x_j^s, j=1, 2, \dots, N$, та зіставити кожному екземпляру значення цільової ознаки y^s :

$$x_j^s = \begin{cases} jV - 0,1s, & j = 1, 5, 9, \dots, \\ 0,01 jV^{-1} + 0,3s, & j = 2, 4, 6, \dots, \\ jrand, & j = 3, 7, 11, \dots; \end{cases} \quad y^s = \begin{cases} 0, (x_1^s)^2 + (x_2^s)^2 < V^2 + 0,04S^2, \\ 1, (x_1^s)^2 + (x_2^s)^2 \geq V^2 + 0,04S^2; \end{cases}$$

$$S = \begin{cases} 10V, & V < 10, \\ 5V, & 10 \leq V < 20, \\ 3V, & V \geq 20; \end{cases} \quad N = \begin{cases} 5V, & V < 7, \\ 4V, & 7 \leq V < 10, \\ 3V, & 10 \leq V < 20, \\ 2V, & V \geq 20; \end{cases}$$

де $rand$ – випадкове число в діапазоні $[0, 1]$.

Замість штучної вибірки на цьому етапі роботи можливо використати навчальну вибірку для певної реальної практичної задачі.

3. Виконати нормування навчальної вибірки даних.

4. На основі пронормованої вибірки побудувати розпізнавальну модель за методом еталонів, тобто визначити координати центрів класів (еталонів) у просторі ознак C_j^q , де q – номер класу, j – номер ознаки. Зафіксувати час навчання.

5. На основі побудованої моделі для екземплярів навчальної вибірки виконати розпізнавання, тобто визначити розрахункові номери класів y^{**} . Зафіксувати час розпізнавання.

6. Обчислити помилку розпізнавання, визначити ймовірність прийняття правильного рішення та ймовірність прийняття помилкового рішення для побудованої моделі.

7. У попередньо сформованій вибірці залишити тільки одну ознаку, номер якої у попередній вибірці дорівнює V , а також у центрів еталонів класів залишити тільки V -ту координату C_V^q .

8. Для нової вибірки та нових еталонів виконати розпізнавання, тобто визначити розрахункові номери класів y^{**} . Зафіксувати час розпізнавання.

9. Обчислити помилку розпізнавання, визначити ймовірність прийняття правильного рішення та ймовірність прийняття помилкового рішення для нової моделі.

10. Порівняти результати проведених експериментів, зробити висновки щодо впливу параметрів навчальної вибірки на характеристики процесів навчання та розпізнавання.

Зміст звіту.

1. Мета роботи.
2. Короткі теоретичні відомості до роботи.
3. Текст розробленої програми.
4. Сформована навчальна вибірка $\langle x, y \rangle$.
5. Побудована модель – координати еталонів (центрів зосередження екземплярів) класів. Час навчання.
6. Результати розпізнавання: розраховані значення номеру класу для екземплярів y^{**} , помилка розпізнавання, імовірності прийняття правильного та помилкового рішень, час розпізнавання.
7. Навчальна вибірка із однією ознакою.
8. Еталони класів з однією ознакою.
9. Результати розпізнавання скороченої вибірки за скороченими еталонами: розраховані значення номеру класу для екземплярів y^{**} , помилка розпізнавання, імовірності прийняття правильного та помилкового рішень, час розпізнавання.
10. Висновки, що містять критичний аналіз результатів роботи.

Завдання 3. Лабораторна робота «Методи відбору ознак для побудови розпізнаючих моделей»

Мета роботи: вивчити та засвоїти на практиці методи оцінювання інформативності та відбору ознак, для побудови розпізнаючих моделей.

Завдання до роботи.

1. Ознайомитися з рекомендованою літературою.
2. На алгоритмічній мові програмування пакету MATLAB написати програму, що реалізує методи оцінювання інформативності

ознак: на основі модуля коефіцієнта парної кореляції, коефіцієнта кореляції знаків, коефіцієнта кореляції Фехнера, кількості інтервалів зміни номера класу, інформаційного підходу, статистичного підходу.

3. Згідно з номером студента за журналом для відповідного номера варіанта V сформувати навчальну вибірку $\langle x, y \rangle$ за формулами, наведеними у завданні попередньої роботи. Також для екземплярів вибірки визначити значення другої цільової ознаки y_2^s :

$$y_2^s = 2x_1^s + 0,1x_2^s, s=1,2,\dots, S.$$

4. На основі сформованої вибірки по відношенню до дискретного виходу y^s оцінити інформативність ознак екземплярів вибірки за допомогою коефіцієнта кореляції знаків, коефіцієнта кореляції Фехнера, кількості інтервалів зміни номера класу, інформаційного підходу, статистичного підходу.

5. На основі сформованої вибірки по відношенню до дійсного виходу y_2^s оцінити інформативність ознак екземплярів вибірки за допомогою на основі модуля коефіцієнта парної кореляції, кількості інтервалів зміни номера класу (для цього попередньо дискретизувати y_2^s).

6. Побудувати таблицю з оцінками інформативності ознак відносно дискретного y^s , стовпці якої повинні мати назви: номер ознаки, коефіцієнт кореляції знаків, коефіцієнт кореляції Фехнера, оцінка за кількістю інтервалів зміни номера класу, оцінка за інформаційним підходом, оцінка на основі статистичного підходу.

7. Побудувати таблицю з оцінками інформативності ознак відносно дійсного y_2^s , стовпці якої повинні мати назви: номер ознаки, модуль коефіцієнта парної кореляції, оцінка за кількістю інтервалів зміни номера класу

8. Проаналізувати за побудованими таблицями оцінки інформативності ознак. Зробити висновки щодо важливості ознак окремо для y^s та y_2^s .

Зміст звіту

1. Мета роботи.
2. Короткі теоретичні відомості до роботи.
3. Текст розробленої програми.
4. Сформована навчальна вибірка $\langle x, y, y_2 \rangle$.

5. Таблиці з оцінками інформативності ознак по відношенню до y^s та y^s_2 .

6. Рішення щодо важливості ознак окремо для y^s та y^s_2 .

7. Висновки. У висновках треба проаналізувати результати роботи, а також лаконічно відповісти на контрольні питання.



Завдання 4. Лабораторна робота «Самоорганізація та навчання без учителя. кластер-аналіз»

Мета роботи: вивчити та засвоїти на практиці методи кластер-аналізу та його використання для навчання з учителем та без учителя.

Завдання до роботи.

1. Ознайомитися з конспектом лекцій та рекомендованою літературою. На алгоритмічній мові програмування пакету MATLAB написати програму, що використовує функції кластер-аналізу.

2. Згідно з номером студента за журналом для відповідного номера варіанта V сформувати навчальну вибірку за допомогою формул лабораторної роботи № 1.

3. На основі вибірки виділити центри кластерів шляхом чіткої та нечіткої кластеризації.

4. Виконати нормування навчальної вибірки даних.

5. На основі пронормованої вибірки виділити центри кластерів шляхом чіткої та нечіткої кластеризації.

6. Порівняти результати для чіткої та нечіткої кластеризації для нормованої та ненормованої вибірок.

Зміст звіту.

1. Мета роботи.

2. Короткі теоретичні відомості до роботи.

3. Текст розробленої програми.

4. Сформована навчальна вибірка.

5. Пронормована вибірка.

6. Координати центрів кластерів для ненормованої та нормованої вибірок після чіткої та нечіткої кластеризації.

7. Висновки, що містять критичний аналіз отриманих результатів.



5.8 Література до розділу

Огляд основних можливостей, компонентів середовища та основі мови програмування пакету Matlab наведено в [5, 9, 13, 20].

АЛФАВІТНО-ПРЕДМЕТНИЙ ПОКАЖЧИК

А

- Агент, 12
 - раціональний, 12
 - інтелектуальний, 13
- Активні правила, 148
- Активність, 126
- Алгоритми рішення, 122

Б

- База даних, 162
- База знань 126, 140
 - нечітка, 160
- База правил, 162
- Блок
 - дефаззіфікації, 162
 - прийняття рішення, 162
 - фаззіфікації, 162

В

- Вибірка, 25
 - навчальна, 26
 - тестова, 26
 - перевірочна, 26
- Виконавець, 13
- Виконавчі механізми, 14
- Вихідний стан проблеми, 169
- Відношення, 10, 122
 - подоби, 10
- Відстань
 - діагностична, 98
 - Евклідова, 49, 96
 - Камберра, 99
 - кореляційна, 99
 - кутова, 99
 - Манхеттена, 96
 - Махалонобіса, 98
 - Мінковського, 98
 - міських кварталів, 96
 - у неізотропному просторі ознак, 98
 - у нелінійному просторі, 98
 - узагальнена, 98
 - Хеммінга, 96
 - Чебишева, 99

Властивість інтелектуальності, 9

- Вузол, 168
 - АБО, 169
 - листовий, 168
 - ТА, 169

Г

- Генеральна сукупність, 25
- Гіперграф, 168
- Гіпердуга, 168
- Гіпотеза про компактність класів, 26
- Граф ТА/АБО, 168

Д

- Дані, 120
- Денотат, 124
- Дефаззіфікація, 162, 163
- Диспетчер, 140

Е

- Евристика, 122
- Експерт, 135
- Експертна система, 128
 - аналізуюча, 132
 - властивості, 128
 - – адекватна робастність, 130
 - – глибинна, 130
 - – гнучкість, 130
 - – евристика, 129
 - – евристична потужність, 130
 - – зручність, 130
 - – інституціональна пам'ять, 129
 - – компетентність, 129
 - – корисність, 130
 - – логічна адекватність, 130
 - – логічна прозорість, 130
 - – природність нотації, 130
 - – самосвідомість, 129
 - – глибинна, 132
 - – демонстраційний прототип, 133
 - – динамічна, 132, 141
 - – діагностична, 131
 - – діючий прототип, 133
 - – довизначальна системи, 134

- дослідницький прототип, 133
- другого покоління, 134
- інтерпретуюча, 131
- класифікуюча системи, 134
- комерційна, 133
- контролю, 132
- моніторингу, 131
- навчальна, 131
- надання допомоги при ремонті, 131
- налагоджувальна, 131
- першого покоління, 134
- планування, 131
- поверхнева, 132
- прогнозуюча, 131
- проектування, 131
- промислової стадії, 133
- проста, 133
- режим роботи, 142
- набуття знань, 142
- – консультації, 142
- синтезуюча, 132
- складна, 133
- статична, 132
- традиційна, 132
- трансформуюча системи, 134
- третього покоління, 134
- функціонування, 142
- Експонентна вага, 52
- Екстенсіонал, 124
- Епістемологія, 121
- Етап
 - виконанн, 138
 - дослідної експлуатації, 138
 - ідентифікації, 137
 - концептуалізації, 138
 - тестування, 138
 - формалізації, 138
- 3**
- Задача
 - апроксимації залежності, 25
 - відбору інформативних ознак, 63
 - ідентифікації, 26
 - інтелектуальна, 8
 - класифікації, 25
 - кластеризації, 44
 - навчання з учителем, 26
- неформалізована, 125
- оцінювання, 25
- параметричного синтезу, 26
- побудови розпізнавальної моделі, 26
- пошуку, 169
- розпізнавання образів, 8, 25
- структурного синтезу, 26
- структурно-параметричного синтезу, 26
- формалізована, 125
- Задачі інтелектуальних систем, 11
 - розширення, 12
 - довізначення, 12
 - перетворення, 12
- Засіб побудови експертної системи, 136
- Змінна
 - лінгвістична, 158
 - нечітка, 157
- Знання, 120
 - апостеріорні, 122
 - априорні, 122
 - базові елементи,
 - види, 122
 - внутрішня інтерпретованість, 126
 - декларативні, 123
 - динамічні, 123
 - експертні, 123
 - екстенсіональні, 124
 - інтенсіональні, 124
 - неформалізовані знання, 125
 - неявні, 123
 - особливості, 125
 - прагматичні, 124
 - процедурні, 122
 - різновиди, 122
 - семантичні, 124
 - синтаксичні, 124
 - статичні, 123
 - типи,
 - типи, 124
 - формалізовані, 125
 - форми існування, 125
- I**
- Ієрархія рівнів знань, 127

- Імовірність
 – апостеріорна, 152
 – апріорна, 152
- Індекс Хіс-Бені, 52
- Інженер зі знань, 135
- Інженерія знань, 9
- Інтелектуальна система, 8
 – загального призначення, 9
 – спеціалізована, 10
- Інтенсіонал, 124
- Інтерпретатор, 140
- Інтерфейсна підсистема, 141
- Інформатика, 121
- Інформація, 120
 – способи подання, 120
- К**
- Клас сутностей, 10
- Клас, 25, 43
- Класи
 – лінійно роздільні, 27
 – нелінійно роздільні, 27
- Класифікуючі відношення, 126
- Кластер, 43
- Кластерний аналіз, 43
 – чіткий, 44
 – нечіткий, 44, 49
- Когнітолог, 135
- Коефіцієнт упевненості, 149
- Контрапозиція, 147
- Концепт, 124
- Концепція, 121
 – «швидкого прототипу», 136
- Користувач, 136
- Критерій
 – зупинення, 67
 – компактності та роздільності, 52
 – оптимальності, 26
 – оцінювання інформативності ознак, 82
 – оцінювання ознак, 66
- Критерій оцінювання групової інформативності ознак, 89
 – ентропія, 92
 – заснований на статистичному підході, 92
 – інформаційний критерій, 91
- ймовірність прийняття помилкових рішень, 95
 – коефіцієнт кореляції Пірсона, 90
 – критерій Фішера, 96
 – множинне дисперсійне відношення, 90
 – множинний коефіцієнт кореляції, 89
 – множинний коефіцієнт зв'язку, 91
 – на основі теорії чітких множин, 93
 – на основі теорії нечітких множин, 94
 – на основі помилок синтезованих моделей, 94
 – при класифікації, 95
 – у завданнях прогнозування, 94
 – фільтруючих методів, 89
- Критерій оцінювання індивідуальної інформативності ознак, 82
 – заснований на імовірнісному підході, 87
 – заснований на статистичному підході, 88
 – дисперсійне відношення, 84
 – ентропія ознаки, 87
 – інформаційний критерій, 85
 – коефіцієнт парної кореляції, 82
 – коефіцієнт кореляції знаків, 84
 – коефіцієнт кореляції Фехнера, 84
 – коефіцієнт зв'язку, 85
 – теоретико-інформаційний критерій,
- Критик, 13
- Л**
- Ланцюжок виведення
 – зворотного, 149
 – прямого, 148
- Логіка
 – монотонна, 149
 – немонотонна логіка, 149
- Логічна прозорість, 123
- Логічне виведення, 146, 147, 161
 – абдукція, 146, 148
 – аутоепістемічні судження, 146
 – булеве, 147
 – дедукція, 146, 147
 – евристика, 146
 – загальне, 161

- звичайне, 147
- зворотне, 148
- індукція, 146, 154
- інтуїція, 146
- метод породження і перевірки, 146, 171
- нечітке, 157
- судження
- – за аналогією, 147
- – застосовані за замовчуванням, 146
- – монотонні, 147
- – немонотонні, 147
- чітке, 147

M

- Максимальне абсолютне відхилення, 95
 Максимальне відносне відхилення, 95
 Машинологічного виведення, 126, 140
 Метадані, 121
 Метазнання, 121
 Метазнання, 122
 Метод дефазіфікації, 163
 - середній з максимальних, 163
 - найбільший з максимальних, 164
 - найменший з максимальних, 164
 - максимум функції належності, 164
 - центр тяжіння, 164
 - сінглтон, 164
 - центр площини, 164
 - метод медіані, 164
 - висотна дефазіфікація, 164
 Метод ковзного іспиту, 27
 Метод нечіткого виведення, 165
 - вихідний, 165
 - зворотний, 167
 - Ларсена, 166
 - Мамдані, 165
 - прямий, 165
 - спадний, 167
 - спрощений, 166
 - Сугено, 167
 - Цукамото, 165
 Метод чіткого виведення
 - Байсса, 152
 - ДСМ, 154
 - Нейлора, 153
 Методи відбору інформативних

- ознак, 68
- вбудовувані методи, 66, 67
- вбудовуючі методи, 66
- видалення ознак, 75
- випадкового пошуку, 65
- випадкового пошуку з адаптацією, 80
- гілок і границь, 71
- групового урахування аргументів, 72
- додавання ознак, 74
- додавання та видалення ознак, 76
- еволюційного пошуку, 81
- евристичні, 65
- евристичні, 73
- кластеризації ознак, 78
- обхід дерева, 69
- перебору, 65
- повного перебору, 68
- пошуку в глибину, 69
- пошуку в ширину, 70
- ранжирування ознак, 78
- ранжирування, 65
- скороченого перебору, 71
- фільтруючі методи, 65, 66
- Методи кластеризації, 44
 - FCM , 50
 - аддитивний нечіткої самоорганізації, 54
 - гіbridний, 48
 - гірської кластеризації, 44
 - Густавсона-Кесселя, 53
 - з видаленням кластерів, 48
 - з додаванням кластерів, 47
 - ієрархічні, 50
 - поступово зростаючого розбиття IDA, 56
 - пікового групування, 44
 - різницевого групування, 45
 - редукція кількості кластерів, 52
 - сіткові, 50
 - субтрактивної кластеризації, 45
 - часткові, 50
 - цільносні, 50
- Методи розпізнавання образів, 28
 - CBR, 34
 - АВО, 33
 - витягу асоціативних правил , 40

- дискримінантних функцій, 31
 - евристичні, 28
 - екстенсіональні, 29
 - еталонів, 42
 - засновані на припущеннях про клас вирішальних функцій, 38
 - інтенсіональні, 29
 - кількісні, 28
 - лінгвістичні, 41
 - логічні, 40
 - МГУА, 39
 - метричні, 32
 - метричної класифікації, 42
 - на основі м'яких обчислень, 35
 - найближчих сусідів, 33
 - нейро-нечіткі мережі, 37
 - нейронні мережі, 35
 - непараметричні, 28
 - нечіткі моделі, 37
 - параметричні, 28
 - побудови дерев розв'язків, 41
 - поділу у просторі ознак, 31
 - порівняння з еталоном, 34
 - потенціалів, 32
 - потенційних функцій, 31
 - регресійного аналізу, 29
 - системи нечіткого виведення, 37
 - статистичних рішень, 30
 - статистичні, 29
 - стохастичної апроксимації, 40
 - структурні, 41
 - якісні, 28
- Методи формування штучних ознак, 99
- Isomap, 100
 - аналіз головних компонентів, 99
 - аналіз незалежних компонентів, 100
 - багатофакторне скорочення розмірності, 100
 - витягу ознак, 99
 - генеральної узагальненої змінної, 100
 - конструювання ознак, 99
 - напізвисначене вкладення, 99
 - нелинейне скорочення розмірності, 100
 - теорія редукції, 100
 - цифрової обробки сигналів, 101
- часткові найменші квадрати, 100
 - Методологія розробки експертних систем, 137
 - Метрики при класифікації ознак, 96
 - Механізм логічних виведень, 126
 - Міра
 - загальної довіри, 151
 - правдоподібності, 151
 - Множина
 - належностей, 157
 - нечітка, 157
 - чітка, 157
 - Моделі подання знань, 124
 - Модифікація експертної системи, 138
 - Модус поненс, 147
 - нечіткий, 161
 - Модус толенс, 147
 - Мудрість, 121
 - Мультиагентна система, 134
- Н**
- Навчальний компонент, 14
 - Навчання, 44
 - без учителя, 44
 - з учителем, 44
- Нечітке відношення, 159
- Норма, 52, 53
- Евклідова, 53
 - діагональна норма, 53
 - Махalanобіса, 53
- О**
- Оболонка експертної системи, 140
 - Ознака, 25
 - Онтологія, 121
 - Операції над нечіткими відношеннями, 159
 - композиція, 159, 161
 - нечітка імплікація, 161
 - об'єднання, 159
 - перетинання, 159
- Операції над нечіткими множинами, 158
- доповнення, 158
 - заперечення, 158
 - логічна сума, 159
 - логічний добуток, 159

- нечітка імплікація, 159
- об’єднання, 158
- перетинання, 159
- Оцінна функція, 76
- П**
- Пакет Matlab, 192
 - Array Editor, 193
 - Fuzzy Logic Toolbox, 193
 - Neural Network Toolbox, 193
 - Statistics Toolbox, 193
 - Workspace Browser, 193
- Переконструювання подання, 139
- Переформулювання понять, 139
- Периферія, 168
- Підсистема набуття знань, 141
- Підсистема пояснень, 141
- Планування доцільного поводження, 8
- Подання знань, 8
- Помилка
 - середньоквадратична, 94
 - середня абсолютна, 95
 - середня відносна, 95
 - сума відносних відхилень, 95
 - сума значень абсолютних відхилень, 95
 - сума квадратів відхилень, 94
- Початкова точка пошуку, 65
- Пошук у просторі станів, 169
 - IDA*, 182
 - RBFS, 180
 - A* з ітеративним поглибленим, 182
 - A*, 181
 - в ширину, 171
 - в глибину, 172
 - двонаправлений, 170
 - евристичний, 175
 - ефективність, 170
 - жадібний локальний, 177
 - з поверненнями, 174
 - за першим найкращим збігом, 178
 - зворотний, 169
 - зі сходженням до вершини, 176
 - інформований, 175
 - неінформований, 170
 - оптимальність, 170
 - повнота, 170
 - породження і перевірки, 171
 - припустимий, 170
 - продуктивність, 170
 - просторова складність, 170
 - пряний, 169
 - рекурсивний за першим найкращим збігом, 180
 - сліпий, 170, 171
 - спрямований, 170
 - часова складність, 170
- Пошук, 167
- Пояснення, 121
- Правило, 140
 - Байеса, 152
 - Демпстера, 151
 - добутку, 152
- Прагматика, 124
- Принцип
 - единого залишку, 155
 - единого розходження, 154
 - одної подібності, 155
- Принцип індукції, 154
- Проблемна область, 10
- Проблемне середовище, 10
 - детерміноване, 10
 - динамічне, 11
 - дискретне, 11
 - епізодичне, 11
 - мультиагентне, 11
 - напівдинамічне, 11
 - неперервне, 11
 - одноагентне, 11
 - повністю спостережне, 10
 - послідовне, 11
 - статичне, 11
 - стохастичне, 11
 - частково спостережне, 10
- Програміст, 135
- Простір пошуку, 169
 - глибина, 169
 - розмір, 169
 - ширина, 169
- Простір станів, 168
- Процедура пошуку оптимального набору ознак, 65
- Р**
- Редактор бази знань, 141

- Репрезентативність, 26
Рівень
– алгоритмів і структур даних, 127
– апаратних засобів, 127
– знань, 127
– компонування, 127
– мов програмування, 127
– символів, 127
Рішення, 8
Робоча пам'ять, 140
Робочий список правил, 140
C
Семантика, 124
Семіотична система, 123
Семіотична система, 124
Сенсор, 13
Силогізм, 147
Синтаксис, 124
Синтез нечітких правил, 56
Система нечіткого виведення, 162
Система, заснована на знаннях, 126
Ситуативні зв'язки, 126
Скриня
– чорна, 26
– біла, 26
– сіра, 26
Спілкування людини з ЕОМ, 8
Стан, 168
Стандарт продуктивності, 13
Стенфордська теорія коефіцієнта впевненості, 149
Стратегії оптимізації набору ознак, 65
Стратегія, 122
Структура експертної системи, 139
Структурованість, 126
Сутність, 10
T
Тавтологія, 147
Твердження і визначення, 121
Теорема Байсса, 153
Теореми і правила перезапису, 122
Теорія Демпстера–Шафера, 151
Терм нечіткий, 158
Терм, 158
Терм-множина, 158
Тест завершення, 169
Типи систем нечіткого виведення, 162
У
Удоосконалення прототипу, 139
Ф
Фаззіфікація, 161, 163
Факт, 120, , 140
Фрейм розрізnenня, 151
Функція належності, 157
Ціна симптому, 153
Ш
Шкаловання, 126
Штучний інтелект, 8
– історія, 14
Шум, 120

ЛІТЕРАТУРА

Основна література

1. Зайченко Ю. П. Основи проектування інтелектуальних систем : навч. посіб. / Ю. П. Зайченко. – К. : Слово, 2004.– 352 с.
2. Люгер Дж. Ф. Искусственный интеллект : стратегии и методы решения сложных проблем / Дж. Ф. Люгер. – М. : Вильямс, 2005. – 864 с.
3. Олійник А. О. Інтелектуальний аналіз даних : навч. посіб. / А. О. Олійник, С. О. Субботін, О. О. Олійник. – Запоріжжя : ЗНТУ, 2011. – 271 с.
4. Рассел С. Искусственный интеллект : современный подход / С. Рассел, П. Норвиг ; пер с англ. – М. : Вильямс, 2006. – 1408 с.
5. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень : навч. посіб. / С. О. Субботін. – Запоріжжя : ЗНТУ, 2008. – 341 с.

Додаткова література

6. Encyclopedia of artificial intelligence / eds.: J. R. Dopico, J. D. de la Calle, A. P. Sierra. – New York : Information Science Reference, 2009. – Vol. 1–3. – 1677 p.
7. Барсегян А. А. Технологии анализа данных : Data Mining, Visual Mining, Text Mining, OLAP: Уч. пос. / А. А. Барсегян. – СПб : BHV, 2007. – 384 с.
8. Брянцев И. Н. Data Mining. Теория и практика / И. Н. Брянцев. – М. : БДЦ-Пресс, 2006. – 208 с.
9. Дьяконов В. Matlab 6: Учебный курс / В. Дьяконов. – СПб.: Питер, 2002. – 592 с.
10. Интеллектуальные информационные технологии проектирования автоматизированных систем диагностирования и распознавания образов : монография / [С. А. Субботин, Ан. А. Олейник, Е. А. Гофман, С. А. Зайцев, Ал. А. Олейник]; под ред. С. А. Субботина. – Харьков : Компания СМИТ, 2012. – 318 с.
11. Интеллектуальные средства диагностики и прогнозирования надежности авиадвигателей : монография / [В. И. Дубровин, С. А. Субботин, А. В. Богуслаев, В. К. Яценко]. – Запорожье : ОАО «Мотор-Сич», 2003. – 279 с.

12. Кричевский М. Л. Интеллектуальные методы в менеджменте. / М. Л. Кричевский. – СПб. : Питер, 2005.– 304 с.
13. Леоненков А. В. Нечеткое моделирование в среде MATLAB и FuzzyTech / А. В. Леоненков. – СПб.: БХВ–Петербург, 2003. – 736 с.
14. Прогрессивные технологии моделирования, оптимизации и интеллектуальной автоматизации этапов жизненного цикла авиационных двигателей : монография / [А. В. Богуслаев, Ал. А. Олейник, Ан. А. Олейник, Д. В. Павленко, С. А. Субботин] ; под ред. Д. В. Павленко, С. А. Субботина. – Запорожье: ОАО «Мотор Сич», 2009. – 468 с.
15. Рідкокаша А. А. Основи систем штучного інтелекту : . навч. посіб. / А. А. Рідкокаша, К. К. Голдер. – Черкаси, «Відлуння–Плюс», 2002.–240 с.
16. Ротштейн А. П. Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети / А. П. Ротштейн. – Винница : УНИВЕРСУМ–Винница, 1999. – 320 с.
17. Субботін С. О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей: монографія / С. О. Субботін, А. О. Олійник, О. О. Олійник ; під заг. ред. С. О. Субботіна. – Запоріжжя : ЗНТУ, 2009. – 375 с.
18. Черноруцкий И. Г. Методы принятия решений / И. Г. Черноруцкий. – СПб. : БХВ–Петербург, 2005. – 416 с.
- Електронні ресурси*
19. Комп'ютерне моделювання та інтелектуальні системи : веб-сайт. – Режим доступу: <http://www.csit.narod.ru>
20. Exponenta. ru : веб-сайт. – Режим доступу: <http://www.exponenta.ru>

Навчальне видання

СУББОТІН Сергій Олександрович
ОЛІЙНИК Андрій Олександрович

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

Навчальний посібник

під загальною редакцією С. О. Субботіна

Комп'ютерний набір Субботін С. О.

Оригинал-макет підготовлено
в редакційно-видавничому відділі ЗНТУ

Підписано до друку 26.02.2014. Формат 60×84/16. Ум. друк. арк. 12,8.
Тираж 300 прим. Зам. № 181.

Запорізький національний технічний університет
Україна, 69063, м. Запоріжжя, вул. Жуковського, 64
Тел.: (061) 769–82–96, 220–12–14

Свідоцтво суб'єкта видавничої справи ДК № 2394 від 27.12.2005.