

УДК 519.7:004.8
КП
№ДР 0113U000361
Інв. №

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
(ХНУРЕ)
61166, м. Харків, пр. Науки, 14
тел. (057) 702-13-97; телефакс (057) 702-14-13

ЗАТВЕРДЖУЮ
Т.в.о. проректора з наукової роботи ХНУРЕ
д. ф.-м. н., проф.

В. І. Хаханов

ЗВІТ
ПРО НАУКОВО-ДОСЛІДНУ РОБОТУ
НЕЙРО-ФАЗЗИ СИСТЕМИ ДЛЯ ПОТОЧНОЇ КЛАСТЕРИЗАЦІЇ ТА
КЛАСИФІКАЦІЇ ПОСЛІДОВНОСТЕЙ ДАНИХ ЗА УМОВ ЇХ
ВИКРИВЛЕНOSTІ ВІДСУТНІМИ ТА АНОМАЛЬНИМИ
СПОСТЕРЕЖЕННЯМИ
№ 273
(заключний)

Керівник НДР
д-р техн. наук, професор

Є. В. Бодянський

2015

Рукопис завершено 21 грудня 2015 р.

Результати цієї роботи розглянуто Вченою радою ХНУРЕ
протокол № _____ від _____

СПИСОК АВТОРІВ

Керівник НДР, проф., г.н.с., д.т.н., проф.	Бодяньський Є.В. (реферат, вступ, висновки, розд. 1-4)
Відповідальний виконавець, п.н.с., к.т.н., с.н.с.	Плісс І.П. (підрозд. 1.1-1.3, 2.1, 2.2)
г.н.с., д.т.н, проф.	Винокурова О.А. (розд. 3.2)
с.н.с., к.т.н.	Тищенко О.К. (підрозд. 1.2)
с.н.с., к.т.н, с.н.с.	Ламонова Н.С. (підрозд. 1.1)
с.н.с., к.т.н.	Тесленко Н.О. (підрозд. 2.1)
н.с., к.т.н.	Чепенко Т.Є. (підрозд. 2.2)
с.н.с., к.т.н.	Долотов А.І. (підрозд. 2.2)
м.н.с., к.т.н.	Дейнеко А.О. (підрозд. 3.1)
м.н.с., к.т.н.	Шафроненко А.Ю. (підрозд. 1.1, 1.2)
м.н.с., к.т.н.	Колчигін Б.В. (підрозд. 2.2)
н.с., к.т.н.	Шкуро К.О. (підрозд. 3.2)
асп.	Татарінова Ю.Є. (підрозд. 3.2)

асп.	Копаліані Д.С. (підрозд. 2.2)
асп.	Бойко О.О. (підрозд. 1.2)
асп.	Куценко Я.В. (підрозд. 1.1)
асп.	Самітова В.О. (підрозд. 1.1)
асп.	Дрюк О.Д. (підрозд. 1.2.3)
магістрант	Кобилін І.О. (підрозд. 1.3)
магістрант	Харченко О.О. (підрозд. 3.3)
магістрант	Алексенко В.С. (підрозд. 1.1)
магістрант	Малишева Д.М. (підрозд. 1.2)
магістрант	Скуратов М.В. (підрозд. 1.3)
магістрант	Машошин Д.А. (підрозд. 2.1)
магістрант	Стольнікова М.З. (підрозд. 2.1)
магістрант	Тихонов А.О. (підрозд. 2.2)
магістрант	Ушков К.І. (підрозд. 1.1)
магістрант	Нечитайло А.Ю. (підрозд. 1.2)

магістрант	Сапожников Ю.А. (підрозд. 1.3)
магістрант	Цибулько В.І. (підрозд. 2.3)
студент	Езе Ф.М. (підрозд. 1.2)
студент	Заїка О.А. (підрозд. 2.1)
студент	Агафонов В.В. (підрозд. 2.2)
студент	Ганжа Д.Д. (підрозд. 2.2)
Г.Н.С., Д.Т.Н., С.Н.С.	Попов С.В. (розд. 4)
К.Т.Н., доц.	Машталір С.В. (розд. 5, підрозд. 5.1-5.3)
Д.Т.Н., проф.	Кіріченко Л.О. (підрозд. 5.4)
асп.	Богучарський С.І. (підрозд. 5.1)
асп.	Кобицька Ю.А. (підрозд. 5.4)
асп.	Коляда М.О. (підрозд. 5.2)
Керівник розділу НДР, Д.Т.Н., проф.	Корабльов М.М. (розд. 6)
П.Н.С., К.Т.Н.	Сотніков О.М.

с.н.с.	(розд. 6.1) Єлаков С.Г.
с.н.с.	(розд. 6.2) Єлісеєв А.О. (розд. 6.2)
д.т.н., проф.	Удовенко С.Г. (підрозд. 6.1)
к.т.н., доц.	Аксак Н.Г. (підрозд. 6.2)
докторант, к.т.н.	Безсонов О.О. (підрозд. 6.1)
доц., к.т.н.	Барковская О.Ю. (підрозд. 6.2)
ст.викл., к.т.н.	Сорокіна І.В. (підрозд. 6.1)
ст.викл.	Фомічов А.А. (підрозд. 6.2)
ст. викл.	Іващенко Г.С. (підрозд. 6.1)
асп.	Кушнар'ов М.В. (підрозд. 6.1)
асп.	Легедіна О.В. (підрозд. 6.2)
асп.	Ілюнін О.О. (підрозд. 6.2)
асп.	Шкловець А.В. (підрозд. 6.1)
асп.	Гришко А.О. (підрозд. 6.2)

асистент	Соковікова Н.С. (підрозд. 6.1)
студент	Єськов Р.Г. (підрозд. 6.1)
студент	Житник І.О. (підрозд. 6.2)
Керівник розділу НДР, г.н.с., д.т.н., проф.	Тімофєєв В.О. (розд. 7)
н.с.	Костин Д.Ю. (підрозд. 7.1)
асп.	Лещенко О.В. (підрозд. 7.1)
асп.	Денищук П.М. (підрозд. 7.2)
асп.	Краснояружська К.Г. (підрозд. 7.3)

РЕФЕРАТ

Звіт про НДР: 360 с., 64 рис., 5 табл., 278 посилання.

Об'єкт дослідження: гібридні системи обчислювального інтелекту, засновані на м'яких та рідких обчисленнях.

Предмет дослідження: створення нейро-фаззі систем реального часу, що здатні оброблювати суттєво викривлену вихідну інформацію за умов короткої навчальної вибірки.

Загальною фундаментальною проблемою, на розв'язання якої спрямовано проект, є створення нових підходів та засобів обчислювального інтелекту на основі нейро-фаззі підходів першого і другого типів (Туре-2), що дають можливість ефективної обробки інформації за умов її дефіциту та суттєвої викривленості.

Конкретною фундаментальною задачею в межах загальної проблеми, розв'язання якої передбачалося в проекті, є створення нейро-фаззі систем, що здатні навчатися у послідовному режимі, розв'язуючі задачі кластеризації та класифікації, при цьому, як навчання, так і сама кластеризація та класифікація, повинні протікати у on-line режимі (кількість даних співрозмірна з кількістю настроюваних параметрів системи), а самі дані містять викиди та пропуски (загублені спостереження). Основна відмінність запропонованого підходу від відомих полягає в тому, що традиційні нейро-фаззі системи навчаються на репрезентативних вибірках неспотворених («чистих») даних, при цьому передбачається, як правило, off-line настроювання у пакетному режимі, засноване на оптимізаційному підході.

Метою проекту є подальший розвиток теорії, нових методів та засобів обчислювального інтелекту, призначених для розв'язання широкого класу задач інтелектуального аналізу даних і перш за все кластеризації та класифікації з використанням найсучасніших досягнень у цій галузі (рідкі обчислення, спайк-нейронні мережі, фаззі-системи типу-2) у реальному часі

за умов дефіциту та викривленості вихідної інформації. Основними завданнями є:

- створення on-line нейро-фаззі систем для кластеризації та класифікації даних у формі таблиць «об'єкт-властивість» та коротких часових рядів за наявності пропусків у спостереженнях з використанням процедур відновлення та спеціальних метрик або функцій подібності;

- створення on-line нейро-фаззі систем для кластеризації та класифікації даних, що містять аномальні викиди невідомої (нестатистичної) природи з використанням спеціальних цільових функцій та правил навчання;

- створення багат шарових нейро-фаззі систем для інтелектуального аналізу даних в on-line режимі з використанням спеціалізованих архітектур та правил навчання, орієнтованих на малу вибірку викривлених даних та нечітке виведення із змінним фаззіфікатором або нечіткими функціями належності типу-2.

ОБЧИСЛЮВАЛЬНИЙ ІНТЕЛЕКТ, М'ЯКІ ОБЧИСЛЕННЯ, САМОНАВЧАННЯ, ГІБРИДНІ СИСТЕМИ, НЕЙРО- НЕО- ФАЗЗИ-ВЕЙВЛЕТ- МЕРЕЖІ, ІНТЕЛЕКТУАЛЬНА ОБРОБКА ТА АНАЛІЗ ДАНИХ В УМОВАХ НЕВИЗНАЧЕНОСТІ; АНОМАЛЬНІ ВИКИДИ; НЕЧІТКИМИ ФУНКЦІЯМИ НАЛЕЖНОСТІ ТИПУ-2.

ЗМІСТ

ВСТУП.....	12
1 АДАПТИВНІ МЕТОДИ КРАСТЕРИЗАЦІЇ ТА КЛАСИФІКАЦІЇ ДАНИХ У ВИГЛЯДІ ТАБЛИЦЬ АБО ПОСЛІДОВНОСТЕЙ, ЩО МІСТЯТЬ АПРІОРИ НЕВІДОМУ КІЛЬКІСТЬ ПРОПУСКІВ.....	16
1.1 Адаптивне відновлення пропущених даних в таблицях «об’єкт- властивість» і часових рядах.....	16
1.1.1 Прості алгоритми заповнення пропусків у таблицях «об’єкт-властивість».....	18
1.1.2 Складні алгоритми заповнення пропусків у таблицях.....	21
1.1.3 Адаптивний лінійний елемент Адаліна.....	24
1.1.4 Ансамблі нейронних мереж.....	26
1.1.5 Адаптивне відновлення даних з пропусками.....	27
1.1.6 Адаптивне відновлення даних з пропусками на основі ансамблю нео-фаззі нейронів.....	32
1.1.7 Метод багатовимірної нечіткої екстраполяції.....	36
1.1.8 Відновлення викривлених даних в часових рядах.....	39
1.1.9 Архітектура ортогональної нейронної мережі.....	49
1.2 Адаптивні нейро-фаззі методи для кластеризації викривлених даних.....	54
1.2.1 Формальна постановка задачі кластеризації.....	55
1.2.2 Функції відстані.....	56
1.2.3 Методи кластеризації даних.....	58
1.3 Адаптивна нечітка кластеризація спотворених даних з використанням оптимізаційних процедур.....	86
1.3.1 Адаптивна ймовірнісна нечітка кластеризація даних з пропусками на основі стратегії оптимального розширення.....	87
1.3.2 Адаптивна можливісна кластеризація даних з пропусками	

	10
на основі стратегії оптимального розширення.....	89
1.3.3 Адаптивна нечітка кластеризація даних з відсутніми спостереженнями з використанням стратегії найближчого прототипу – центроїду.....	93
2 АДАПТИВНІ МЕТОДИ КЛАСТЕРИЗАЦІЇ ТА КЛАСИФІКАЦІЇ ДАНИХ У ВИГЛЯДІ ТАБЛИЦЬ АБО ПОСЛІДОВНОСТІ ЗА НАЯВНОСТІ АНОМАЛЬНИХ ВИКИДІВ.....	99
2.1 Адаптивні методи нечіткої кластеризації.....	100
2.1.1 Цільова функція нечіткої кластеризації.....	100
2.1.2 Оптимізація цільової функції.....	102
2.1.3 Лінійний змінний фаззифікатор.....	103
2.1.4 Адаптивна нечітка кластеризація зі змінним фаззифікатором.....	107
2.1.5 Можливісна нечітка кластеризація зі змінним фаззифікатором.....	108
2.1.6 Процедура придушення в нечіткій кластеризації.....	111
2.1.7 Методи, що засновані на підході Густафсона-Кеселя.....	112
2.1.8 Базовий метод Густафсона-Кеселя.....	114
2.1.9 Можливісний варіант метода Густафсона-Кеселя.....	116
2.1.10 Адаптивні версії алгоритма Густафсона-Кеселя.....	117
2.1.11 Робастний метод нечіткої кластеризації на основі критерію Коші.....	120
2.1.12 Імовірнісний підхід.....	121
2.1.13 Можливісний підхід.....	124
2.2 Мапи Кохонена та їх ансамблі для вирішення задач нечіткої кластеризації.....	125
2.2.1 Конкурентне навчання.....	126
2.2.2 Модифіковані нейромережі Кохонена.....	130
2.2.3 Ансамблі та їх навчання.....	137

	11
2.2.4 Основні поняття теорії систем колективного висновку.....	139
2.2.5 Методи побудови ансамблів.....	141
2.2.6 Ансамблі нейро-фаззі мереж Кохонена.....	143
2.2.7 Нечітка кластеризація типу-2 з використанням ансамблів модифікованих нейро-фаззі мереж Кохонена.....	147
3 НЕЙРО-ФАЗЗІ СИСТЕМИ КЛАСТЕРИЗАЦІЇ ТА КЛАСИФІКАЦІЇ ДАНИХ У ON-LINE РЕЖИМІ НА ОСНОВІ СПЕЦІАЛІЗОВАНИХ АРХІТЕКТУР ТА ПРАВИЛ НАВЧАННЯ.....	151
3.1 Системи на основі ядерних функцій активації-належності.....	151
3.2 Багатошарова нейро-фаззі система для діагностування- класифікації нестационарних нелінійних даних.....	169
3.2.1 Архітектура діагностуючої нейро-фаззі системи.....	171
3.2.2 Навчання діагностуючої нейро-фаззі системи.....	174
4 НАВЧАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ НА НЕПОВНИХ ВИКРЕВЛЕНИХ ДАНИХ.....	180
5 МОДЕЛІ І МЕТОДИ АДАПТИВНОГО ПРОГНОЗУВАННЯ ТА КЛАСИФІКАЦІЇ ЧАСОВИХ РЯДІВ.....	189
5.1 Методи адаптивного прогнозування на основі багатовимірної лінійної екстраполяції і використанні метрик і функцій близькості...	190
5.2 Адаптивне прогнозування часових рядів при рівновіддалених спостереженнях.....	197
5.3 Адаптивна поліноміальна нейромережева прогноуюча модель часових рядів і її навчання.....	205
5.4 Розробка системи підтримки прийняття рішень для аналізу, класифікації та прогнозування фрактальних часових рядів.....	214
5.4.1 Опис СППР, аналіз фрактальних і мультифрактальних характеристик ряду.....	215
5.4.2 Аналіз інформаційних характеристик ряду.....	219
5.4.3. Вхідні модельні дані.....	224

	12
5.4.4. Експериментальні фізіологічні дані.....	225
5.4.5 Результати досліджень.....	226
5.4.6 Дослідження зашумлених фрактальних часових рядів.....	232
6 АДАПТИВНІ МЕТОДИ ТА МОДЕЛІ КЛАСИФІКАЦІЇ ДАНИХ І ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ ЗА УМОВ ЇХ ВИКРИВЛЕНОСТІ ВІДСУТНІМИ ТА АНОМАЛЬНИМИ СПОСТЕРЕЖЕННЯМИ НА ОСНОВІ ШТУЧНИХ ІМУННИХ СИСТЕМ.....	235
6.1. Імунний метод класифікації даних, що мають апіорі невідому кількість пропусків.....	235
6.1.1 Постановка задачі класифікації даних.....	236
6.1.2 Загальна схема класифікації даних на основі імунного підходу.....	237
6.1.3 Відновлення пропущених ознак вихідних даних.....	241
6.1.4 Імунне навчання класифікатора.....	243
6.1.5 Критерії класифікації та угруповання даних.....	251
6.1.6 Експериментальні дослідження.....	253
6.2 Прогнозування часових рядів, що мають пропущені та аномальні значення, в режимі реального часу.....	258
6.2.1 Постановка задачі прогнозування часових рядів.....	260
6.2.2 Виявлення аномалій за допомогою моделей ШС.....	261
6.2.3 Гібридна модель прогнозування часових рядів на основі моделі клонального відбору.....	264
6.2.4 Здобуття прогнозу та процес навчання ШС.....	266
6.2.5 Розподілений алгоритм прогнозування.....	269
6.2.6 Результати експериментальних досліджень.....	274
7 МЕТОДИ ПОТОЧНОГО КОНТРОЛЮ ПОСЛІДОВНОСТЕЙ ДАНИХ ЗА УМОВ ЇХ ВИКРИВЛЕНОСТІ НА ОСНОВІ КРИТИЧНОГО ПІДХОДУ.....	280

7.1 Розробка концепції поточного контролю послідовностей даних за умов викривленості вихідних спостережень.....	280
7.1.1 Аналіз впливу зовнішнього середовища на процедуру контролю.....	288
7.2 Процедури контролю за умів аномальних спостережень на основі критичного підходу.....	290
7.3 Адаптивні процедури обробки результатів спостережень за умів аномальних викідів.....	297
ВИСНОВКИ.....	329
ПЕРЕЛІК ВИКОРИСТАННИХ ДЖЕРЕЛ	331

ВСТУП

Загальною фундаментальною проблемою, на розв'язання якої було спрямовано проект, є створення нових підходів та засобів обчислювального інтелекту на основі нейро-фаззі підходів першого і другого типів, що дають можливість ефективної обробки інформації за умов її дефіциту та суттєвої викривленості.

Конкретною фундаментальною задачею в межах загальної проблеми, розв'язання якої було запропоновано в проекті, є створення нейро-фаззі систем, що здатні навчатися у послідовному режимі, розв'язуючи задачі кластеризації та класифікації, при цьому як навчання, так і сама кластеризація та класифікація, повинні протікати у on-line режимі (кількість даних співрозмірна з кількістю настроюваних параметрів системи), а самі дані містять пропуски (загублені спостереження).

Основна відмінність запропонованого підходу від відомих полягає в тому, що традиційні нейро-фаззі системи навчаються на репрезентативних вибірках неспотворених («чистих») даних, при цьому передбачається, як правило, off-line настроювання у пакетному режимі, засноване на оптимізаційному підході.

Метою проекту є подальший розвиток теорії, нових методів та засобів обчислювального інтелекту, призначених для розв'язання широкого класу задач інтелектуального аналізу даних і перш за все кластеризації та класифікації з використанням найсучасніших досягнень у цій галузі у реальному часі за умов дефіциту та викривленості вихідної інформації. Основними завданнями є:

- створення on-line нейро-фаззі систем для кластеризації та класифікації даних у формі таблиць «об'єкт-властивість» та коротких часових рядів за наявності пропусків у спостереженнях з використанням процедур відновлення та спеціальних метрик або функцій подібності;

- створення on-line нейро-фаззі систем для кластеризації та класифікації даних, що містять аномальні викиди невідомої (нестатистичної) природи з використанням спеціальних цільових функцій та правил навчання;

- створення багатосарових нейро-фаззі систем для інтелектуального аналізу даних в on-line режимі з використанням спеціалізованих архітектур та правил навчання, орієнтованих на малу вибірку викривлених даних та нечітке виведення із змінним фаззіфікатором або нечіткими функціями належності типу-2.

Таким чином, передбачалося створення принципово нових методів обчислювального інтелекту для обробки інформації, аналіз якої є неможливим з використанням відомих традиційних нейро- та нейро-фаззі підходів.

Останні роки для розв'язання задач інтелектуального аналізу даних і знань досить часто використовуються методи наукового напрямку, що розвивається в межах комп'ютерних наук та отримав назву «обчислювальний інтелект». Переваги методів обчислювального інтелекту пояснюються, перш за все, тим, що обробка інформації в них реалізується за аналогією з процесами, що відбуваються у нейро-біологічних системах. Найбільш ефективними системами обчислювального інтелекту є так звані гібридні системи і, перш за все, нейро-фаззі-, вейвлет-нейро-фаззі- та нео-фаззі-мережі, що об'єднують в собі універсальні апроксимуючі властивості нейронних мереж, прозорість та інтерпретуємість систем нечіткого виведення та компактне представлення локальних особливостей сигналів на основі вейвлет-перетворення. Останні десять років цим системам у всьому науковому світі приділяють багато уваги, в тому числі попередні розробки авторів цього проекту були присвячені розробці нових вузлів, архітектур та алгоритмів навчання гібридних вейвлет-нейро-нео-фаззі систем, що мають покращені властивості в порівнянні з відомими.

В той же час передбачається, що навчання або самонавчання таких систем реалізується на основі репрезентативної вибірки, сформованої з

"чистих" даних у вигляді пакету, а якщо у даних і присутні збурення, то вони мають суто випадковий характер та обмежений другий момент. Якщо це не так, і дані викривлені аномальними викидами, або деякі спостереження взагалі відсутні, що притаманне багатьом реальним задачам, навчання у більшості випадків стає просто неможливим. Подоланню цієї проблеми і було присвячено проект, що пропонується до розгляду.

Результати досліджень, проведених у рамках виконання теми, були опубліковані у низці вітчизняних та закордонних монографій або розділів монографій [1-5], статтях у виданнях, що індексуються у Scopus та Web of Science [6-15], статтях у закордонних та вітчизняних виданнях [16-40].

1 АДАПТИВНІ МЕТОДИ КРАСТЕРИЗАЦІЇ ТА КЛАСИФІКАЦІЇ ДАНИХ У ВИГЛЯДІ ТАБЛИЦЬ АБО ПОСЛІДОВНОСТЕЙ, ЩО МІСТЯТЬ АПРІОРИ НЕВІДОМУ КІЛЬКІСТЬ ПРОПУСКІВ

У розділі проведено дослідження та створені адаптивні гібридні on-line методи кластеризації та класифікації даних, що можуть бути представлені у вигляді таблиць «об'єкт-властивість» або багатовимірних часових рядів. Розроблено нові швидкодіючі методи кластеризації-класифікації, що на відміну від існуючих дозволяють оброблювати інформацію у послідовному режимі реального часу, при цьому як кількість пропусків, так і їх розташування у масивах даних припускаються невідомими. Передбачені режими як відновлення вихідних масивів, так і прямої їх обробки, при цьому кількість пропусків є відновлення вихідних масивів, так і прямої їх обробки, при цьому кількість пропусків є співрозмірною із кількістю невикревлених даних. За умов надкородкої вибірки даних передбачено використання нового методу просторової екстраполяції вкупі із набором різного типу метрик, що орієнтовані на роботу за умов викривленої інформації. Запропоновані нейро-фаззі методи здатні обробляти суттєво викривлену інформацію в on-line режимі та на сьогодні не мають аналогів, що підтверджено публікаціями у низці провідних закордонних видань та в матеріалах міжнародних конференцій.

1.1 Адаптивне відновлення пропущених даних в таблицях «об'єкт-властивість» і часових рядах

Задача відновлення пропущених значень полягає в заповненні пропусків в таблицях «об'єкт-властивість» і часових рядах найбільш правдоподібними значеннями з найменшою похибкою.

Задачі відновлення таких пропущених спостережень приділялося достатньо уваги [41-43], при цьому досить ефективними в даній ситуації виявилися підходи, що засновані на математичному апараті обчислювального інтелекту і, насамперед, штучних нейронних мережах [44-47]. Разом з тим, відомі підходи до відновлення пропусків працездатні лише у випадках, коли вихідна таблиця даних задана апріорно і кількість її рядків або стовпців не може змінюватися в процесі обробки.

Алгоритми заповнення пропусків у таблицях даних можна розділити на дві групи: складні і прості.

До простих алгоритмів відносяться такі методи заповнення таблиць як:

- метод заповнення таблиць середнім арифметичним;
- алгоритм найближчого сусіда;
- метод багатовимірної лінійної регресії.

Складні алгоритми, в свою чергу, діляться на дві частини: глобальні та локальні.

До глобальних алгоритмів відносяться:

- метод Барлетта;
- алгоритм *resampling*;
- алгоритми максимальної правдоподібності (МП);
- алгоритми максимального очікування (ЕМ – *expectation maximization*);
- локальний – алгоритм ZET.

Схематично алгоритми заповнення пропусків у таблицях даних демонструє рисунок 1.1.

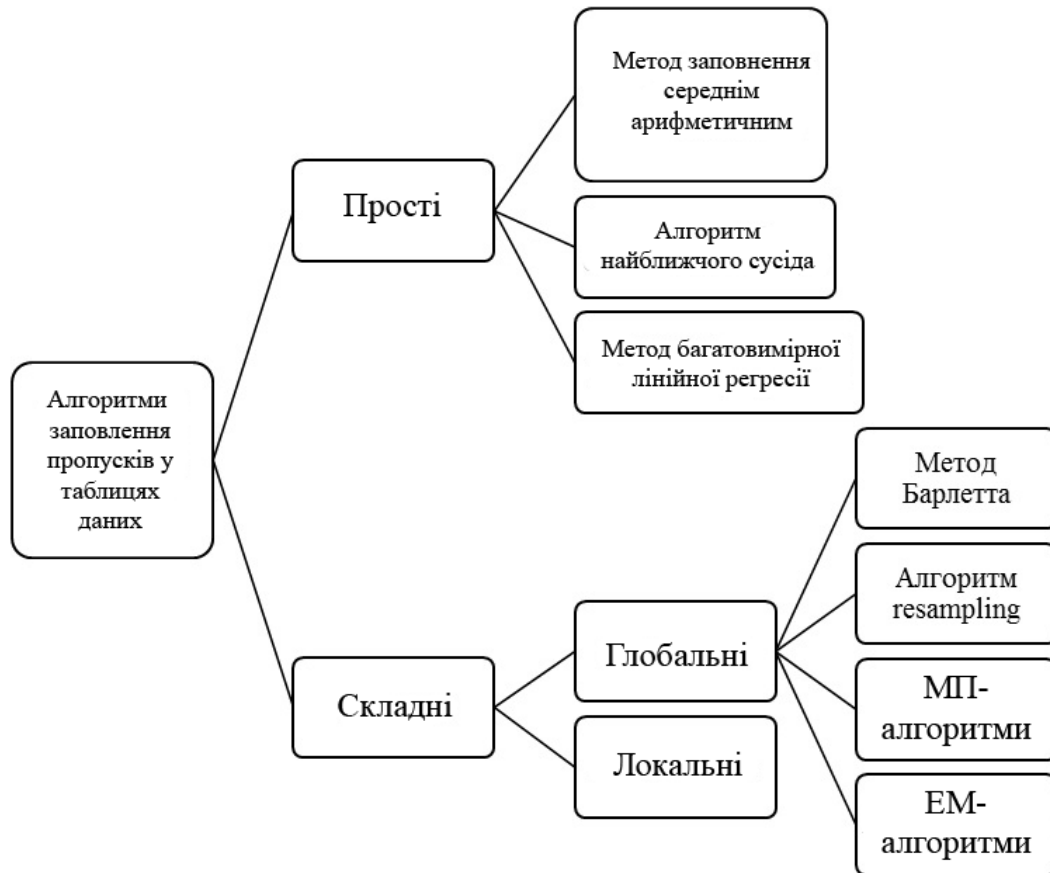


Рисунок 1.1 – Алгоритми заповнення пропусків у таблицях даних

Розглянемо більш докладно загальний принцип роботи алгоритмів заповнення пропусків у таблицях даних «об’єкт-властивість».

Як відзначено вище, дані алгоритми діляться на дві основні підгрупи.

1.1.1 Прості алгоритми заповнення пропусків у таблицях «об’єкт-властивість»

Найбільш поширені алгоритми заповнення таблиць ґрунтуються на методах інтелектуального аналізу даних (Data Mining).

Метод заповнення середнім арифметичним. Цей метод достатньо широко використовується у більшості статистичних пакетів як засіб боротьби з пропусками в таблицях, найпростіший, але не найточніший метод заповнення пропущених даних. Згідно з цим методом, рекомендується

заповнювати пропущені значення таблиць середніми значеннями величин, які містяться в даному стовпці.

Таким чином, невідоме значення ϵ_{ki} в матриці X розміру $N \times n$ відновлюється за формулою:

$$\epsilon_{ki} = \frac{\sum_{i=1}^n \tilde{x}_{ki}}{n-1}. \quad (1.1)$$

Експериментальні дослідження показали неспроможність даного методу навіть на простих наборах даних, не кажучи вже про роботу з даними, які мають більшу вимірність та які надходять на обробку в реальному часі.

Метод k найближчих сусідів (англ. k -nearest neighbor algorithm) – метод автоматичної класифікації об'єктів. Принцип роботи цього методу полягає в присвоєнні об'єкта, що відновлюється, до класу, який є найбільш поширеним серед сусідів цього елемента (об'єкта). Сусіди визначаються з множини об'єктів з відомими класами. У даному методі розраховується найбільш розповсюджений клас k серед об'єктів.

В основі алгоритму k найближчих сусідів лежить припущення, що якщо об'єкти близькі за значеннями $n-1$ властивостей, то вони близькі за значенням n -ої властивості.

Відновлення пропущених даних у таблиці «об'єкт-властивість» методом k найближчих сусідів полягає в знаходженні k рядків (об'єктів), які найбільш «схожі» на рядок, який містить пропуски. У якості міри схожості рядків використовують декартову відстань між рядками (об'єктами) у просторі стовпців (властивостей). Чим менша декартова відстань між об'єктами у просторі властивостей, тим більше вони «схожі» один на одного. Стовпець, який містить значення, що передбачається, називається цільовим стовпцем. Для отримання передбачення невідомого елемента значення

цільової властивості у k найближчих сусідів усереднюються з вагами, обернено пропорційними декартовій відстані до рядка, що містить пропуск.

$$\epsilon_{ki} = \frac{\sum_{l=1}^q \tilde{x}_{li} w_l}{\sum_{l=1}^q w_l}, \quad (1.2)$$

де w_l – вага (компетентність) l -го найближчого сусіда,

D_{li} – декартова відстань між l -тим і i -тим рядком

$$D_{li} = \sum_{p \neq l} (\tilde{x}_{lp} - \tilde{x}_{jp})^2 \quad (1.3)$$

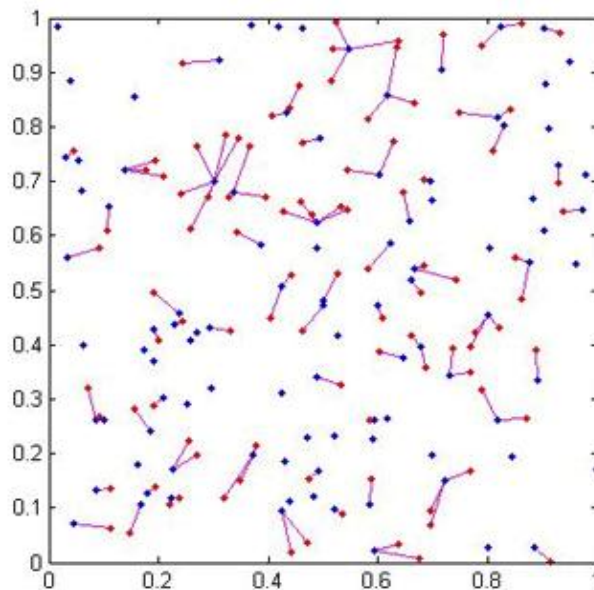


Рисунок 1.2 – Схематичне зображення роботи методу k найближчих сусідів

Алгоритм найближчого сусіда заповнює пропуск значенням об'єкта, який є найближчим за подібністю з об'єктом (пропуском), що передбачається, як зображено на рисунку 1.2.

Основною особливістю, якою виділяється метод k - NN від інших – це відсутність навчання, тобто відсутність побудови функції, що класифікує. До переваг даного методу варто віднести можливість оновлення вибірки без перенавчання класифікатора. Це зручно, коли дані надходять і поповнюються новими спостереженнями. Недоліком алгоритму є запам'ятовування попередніх спостережень та їхнє порівняння з новими об'єктами, що потребує достатньо багато часу на етапі класифікації.

Метод багатовимірної лінійної регресії. На першому етапі роботи методу список незалежних змінних пустий. Спочатку розглядаються всі лінійні залежності від однієї змінної і для кожної розраховуються стандартні похибки і значення F -статистики. Далі обирається незалежна змінна, розраховується мінімальна похибка і включається до списку, аналогічним чином відбувається з другою змінною. Такі ж дії відбуваються і у двовимірній лінійній моделі. Далі йде порівняння моделей і обирається найкраща серед них. Таким чином на кожному кроці додається нова незалежна змінна. Цей процес продовжується доти, доки певний рівень достовірності не припинить покращуватися у порівнянні з попереднім. Закінчення процесу додавання нових змінних означає, що всі змінні, що значним чином впливають на залежну змінну, вже включені до списку (якщо, звісно, список не залишився пустим). До недоліків даного методу варто віднести те, що даний метод погано працює з даними, в яких мало чисел і багато дискретних полів, а також може зовсім не виявити наявності якоїсь залежності в даних.

1.1.2 Складні алгоритми заповнення пропусків у таблицях

Існують і більш складні алгоритми заповнення пропусків. Метод Барлетта і *resampling* метод ґрунтуються на рівняннях багатовимірної лінійної регресії.

У методі Барлетта спочатку пропущеному значенню присвоюється початкове значення (наприклад, середнє арифметичне стовпця), далі йде побудування рівняння багатовимірної лінійної регресії з використанням початкового значення пропуску, і після аналізу цього рівняння аналізується пропуск.

Метод *resampling* є ітеративним. У даному методі пропуск заповнюється набором значень і для кожного значення будується рівняння лінійної регресії. На наступному кроці аналізуються результати розв'язків рівнянь, на основі чого передбачається значення пропущеного елемента таблиці «об'єкт-властивість».

Метод максимальної правдоподібності. В основі даного методу лежить оцінювання невідомого параметра шляхом максимізації функції правдоподібності. До переваг даного методу варто віднести унікальність і простоту визначення рішення, однак метод працює на фіксованому наборі даних і базовій імовірнісній моделі. Таким чином, суттєвим недоліком методу є робота в пакетному режимі за умови, що дані у процесі обробки не змінюються і не надходять в реальному часі.

Метод максимального очікування (EM – *expectation maximization*). Являє собою процедуру ітерацій, яка призначена для вирішення задач оптимізації деякого функціонала, за допомогою аналітичного пошуку екстремумів цільової функції. Даний метод ділиться на два етапи. На першому етапі «очікування» (E – *expectation*) за сукупністю наявних спостережень (об'єктів) розраховуються очікувані значення для кожного неповного спостереження. Після отримання заповненого масиву даних оцінюються основні статистичні параметри. На другому етапі «максимізація» (M – *maximization*) відбувається максимізація ступеня відповідності очікуваних даних і даних, що реально підставляються.

Методи, згадані вище, діють глобально: в них припускається, що залежність заданого (наприклад, лінійного) типу реалізована на всіх об'єктах, тому і в оцінюванні залежності беруть участь всі рядки і стовпці.

Алгоритм ZET. Одним з найбільш ефективних локальних алгоритмів, призначених для заповнення пропущених значень у таблицях «об'єкт-властивість», є алгоритм ZET [41]. В основу алгоритму входить три складника: гіпотеза надмірності, гіпотеза аналогічності і гіпотеза локальної компетентності.



Рисунок 1.3 – Гіпотези алгоритму ZET

Гіпотеза надмірності полягає в аналізі реальних таблиць, які мають надмірність у схожості між собою рядків (об'єктів) і стовпців (властивостей), залежних один від одного. У випадку відсутності надмірності віддання переваги одному прогнозу над іншим є неможливим.

Гіпотеза аналогічності встановлює близькість об'єктів за n -тою властивістю у випадку, коли деяка пара об'єктів близька за значеннями $(n-1)$.

Гіпотеза локальної компетентності стверджує, що надмірність має локальний характер, тобто у кожного об'єкта є своя підмножина об'єктів-аналогів і у кожної властивості є своя підмножина властивостей-аналогів. У передбаченнях повинні брати участь тільки «компетентні» рядки і стовпці, які обираються для кожного елемента, що передбачується, окремо.

Умовно робота алгоритму складається з трьох кроків.

На першому кроці за обраним пропущеним значенням з вихідної таблиці «об'єкт-властивість» обирається підмножина «компетентних» рядків, а потім для обраних рядків – підмножина «компетентних» стовпців.

На другому кроці відбувається автоматичний підбір параметрів для передбачення мінімальної похибки.

На третьому кроці відбувається прогнозування.

Як було відзначено вище, існує цілий ряд підходів до обробки і відновлення даних, викривлених пропусками, при цьому найбільш ефективними в даній ситуації виявилися підходи, що ґрунтуються на математичному апараті обчислювального інтелекту і, насамперед, штучних нейронних мережах.

Разом з тим, описані підходи до відновлення пропусків працездатні лише у випадках, коли вихідна таблиця даних задана апріорно і кількість її рядків або стовпців не може змінюватися в процесі обробки. Існує багато прикладних задач, коли дані надходять на обробку послідовно в реальному часі, при цьому заздалегідь невідомо, який з векторів-образів, що надходить на обробку, містить пропуски.

Відновлення пропусків у таблицях «об'єкт-властивість» зручно реалізувати за допомогою штучної нейронної мережі. Для більш зручної роботи з даними, які надходять на обробку в реальному часі, пропонується використовувати адаптивні нейронні мережі. У даному випадку використовується нейронна мережа, утворена з n паралельно працюючих елементарних нейронів, що навчаються, а саме адаптивних лінійних елементів (ADALINE), які навчаються на основі модифікованої процедури Качмажа-Уїдроу-Хоффа (Kaczmarz-Widrow-Hoff).

1.1.3 Адаптивний лінійний елемент Адаліна

Одним з найпростіших нейронів, що навчаються, є адаптивний лінійний елемент (ADALINE), запропонований Б. Уїдроу і наведений на

рисунку 1.4.

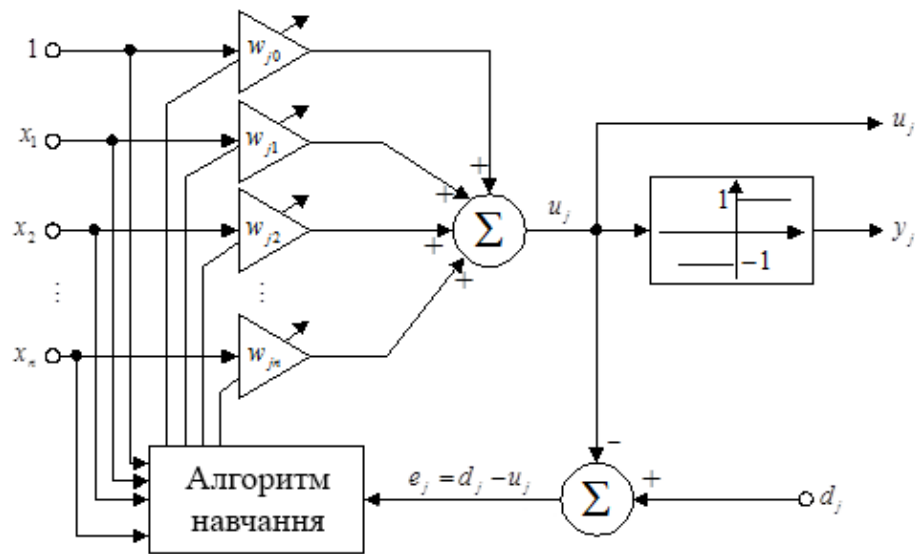


Рисунок 1.4 – Адаптивний лінійний елемент

Структурно адаліна дуже нагадує нейрон МакКалоха-Пітса з активаційною сигнум-функцією і складається з двох основних частин: адаптивного лінійного асоціатора і нелінійної активаційної функції. Адаліна має $n + 1$ входів – x_0, x_1, \dots, x_n і два виходи – аналоговий u_j і бінарний y_j . Крім того, є додатковий вхід, на який подається навчальний сигнал d_j , що показує, яка повинна бути бажана реакція нейрона на кожний конкретний набір вхідних сигналів. Аналоговий вихід u_j являє собою зважену суму входів x_j

$$u_j = \sum_{i=0}^n w_{ji} x_i = w_j^T x, \quad (1.4)$$

а бінарний вихід y_j може приймати значення $+1$ або -1 в залежності від полярності аналогового сигналу u_j . Вихідний сигнал u_j порівнюється із зовнішнім навчальним сигналом d_j , і сигнал похибки $e_j = d_j - u_j$, що виникає

при цьому, надходить до алгоритму навчання, який переналаштовує синаптичні ваги так, щоб мінімізувати деяку функцію похибки e_j , яка називається критерієм навчання. У якості такої функції найчастіше використовують квадратичну, що дозволяє використовувати для навчання не тільки «рідний» алгоритм, синтезований Б. Уїдроу і М. Хоффом спеціально для адаліни, але і безліч рекурентних процедур адаптивної ідентифікації.

Адаліна може використовуватися як в якості елементарного нейрона у складі штучних нейронних мереж (ШНМ), так і самостійно у задачах розпізнавання образів, обробки сигналів, реалізації логічних функцій.

У даному випадку, Адаліни підходять для роботи з даними, які надходять на обробку в послідовному режимі.

1.1.4 Ансамблі нейронних мереж

Якість рішення різних задач прогнозування, розпізнавання образів, зворотнього моделювання, керування, відновлення даних тощо може бути підвищена за рахунок використання ансамблів нейронних мереж, в яких одні й ті самі дані обробляються одразу декількома n паралельно працюючими штучними нейронними мережами. Вихідні сигнали деяким чином комбінуються в загальну оцінку, яка дає уявлення про якість отриманих результатів за допомогою локальних мереж, що входять до ансамблів, як це показано на рис. 1.5.

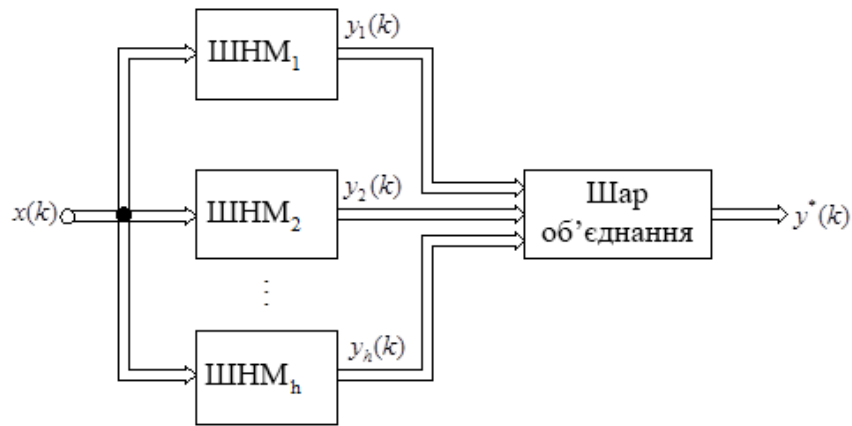


Рисунок 1.5 – Ансамбль h паралельно організованих штучних нейронних мереж

Найбільше розповсюдження до поєднання ШНМ в ансамблі отримали два підходи, такі як модельний і заснований на зваженому усередненні. Дані підходи достатньо відрізняються один від іншого, однак вони також мають і схожість. Вони обидва використовують лінійну комбінацію вихідних сигналів своїх членів в тій або іншій формі.

Модульний підхід має достатньо евристичний характер, на відміну від більш математично строгого зваженого усереднення, хоча і тут залишається елемент суб'єктивізму, пов'язаний з вибором членів ансамблю. Ця задача зазвичай вирішується за допомогою тих або інших евристик, хоча і існують більш або менш строгі результати, що засновані на генетичному програмуванні або поступовому нарощуванні складності мереж-членів ансамблю.

1.1.5 Адаптивне відновлення даних з пропусками

Вихідною інформацією є $N \times n$ таблиця «об'єкт-властивість» вигляду

Таблиця 1.1 - Таблиця «об'єкт-властивість»

	1	...	p	...	j	...	n
1	x_{11}	...	x_{1p}	...	x_{1j}	...	x_{1n}
...
i	x_{i1}	...	x_{ip}	...	x_{ij}	...	x_{in}
...
k	x_{k1}	...	x_{kp}	...	x_{kj}	...	x_{kn}
...
N	x_{N1}	...	x_{Np}	...	x_{Nj}	...	x_{Nn}

Така таблиця містить інформацію про N об'єктів, кожен з яких описується $(1 \times n)$ -вектором-рядком ознак $\underline{x}_i = (x_{i1}, \dots, x_{ip}, \dots, x_{ij}, \dots, x_{in})$, при цьому припускається, що N_G рядків можуть мати по одному пропуску, а $N_F = N - N_G$ заповнені повністю. У процесі обробки таблиці необхідно заповнити пропуски так, щоб відновлені елементи були б в певному сенсі «найбільш правдоподібні» або «близькі» до апіорі невідомих закономірностей, які містяться у таблиці.

Представимо таблицю 1.1 у вигляді $(N \times n)$ -матриці X , в якій відсутній один елемент x_{kj} або, в більш загальному випадку, відсутні N_G елементів. Припускається [41], що між стовпцями $\vec{x}_j = (x_{1j}, \dots, x_{ij}, \dots, x_{kj}, \dots, x_{Nj})^T$ існує лінійна кореляція, на підставі якої і виконується відновлення пропуску за допомогою регресії

$$\epsilon_{kj} = w_{j0} + w_{j1}x_{k1} + w_{j2}x_{k2} + w_{j,j-1}x_{k,j-1} + w_{j,j+1}x_{k,j+1} + \dots + w_{jn}x_{kn}, \quad (1.5)$$

або

$$\epsilon_{kj} = \underline{X}_{kj} w_j, \quad (1.6)$$

де $w_j = (w_{j0}, w_{j1}, \dots, w_{jn})^T$ – $(n \times 1)$ -вектор параметрів, які підлягають визначенню,

$\underline{X}_{kj} = (1, x_{k1}, \dots, x_{k,j-1}, x_{k,j+1}, \dots, x_{kn})$ – вектор-рядок ознак k -того об'єкта без kj -того елемента і з одиницею на першій позиції.

Вектор невідомих параметрів w_j може бути знайдений за допомогою стандартного методу найменших квадратів, для чого з матриці X треба вилучити k -ий рядок, j -й стовпець, додати зліва стовпець з одиниць і на основі отриманої $((N-1) \times n)$ матриці X_j розрахувати оцінки параметрів

$$w_j = (X_j^T X_j)^+ X_j^T \bar{X}_j, \quad (1.7)$$

де $\bar{X}_j = (x_{1j}, \dots, x_{ij}, \dots, x_{k-1,j}, x_{k+1,j}, \dots, x_{Nj})^T$,

$(\bullet)^+$ – символ псевдообернення за Муром-Пенроузом [48].

Якщо ж пропуски є в N_G рядках і в різних стовпцях, з матриці X вилучаються всі ці рядки, і на основі отриманої зрізаної $(N_F \times n)$ матриці n разів знаходяться вектори параметрів (1.7) для всіх $j = 1, 2, \dots, n$.

Далі за допомогою рівняння (1.5) або (1.7) заповнюються всі пропуски отриманими оцінками \hat{x}_{kj} .

Розглянутий метод зручно представити у вигляді структурної схеми, наведеної на рис.1.6.

Цей метод неважко поширити на випадок, коли дані про об'єкти у таблицю 1.1 надходять послідовно об'єкт за об'єктом.

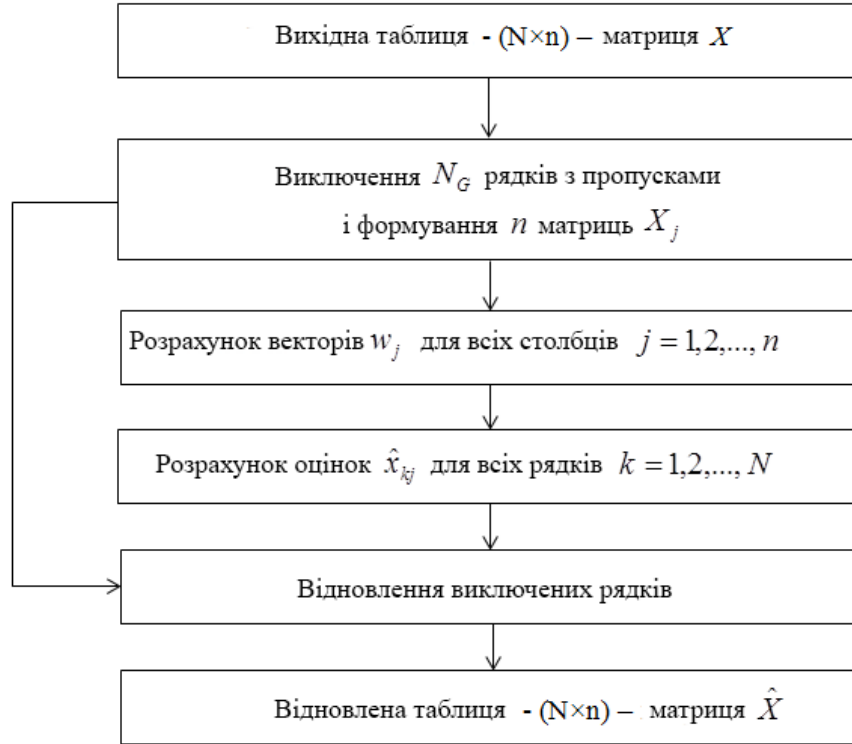


Рисунок 1.6 – Пакетний метод заповнення пропусків

Припустимо без втрати загальності, що спочатку задана інформація у вигляді таблиці 1.1 з N рядками, з яких N_F заповнені повністю. На основі цих даних може бути побудована оцінка

$$w_j(N_F) = (X_j^T(N_F)X_j(N_F))^{-1} X_j^T(N_F)\bar{X}_j(N_F), \quad (1.8)$$

за допомогою якої відновлюється вихідна таблиця. При появі $(N+1)$ -го спостереження у формі повністю заповненого рядка \underline{x}_{N+1} оцінка (1.8) може бути відкоригована за допомогою рекурентного методу найменших квадратів

$$\begin{cases} w_j(N_F + 1) = w_j(N_F) + \frac{P_j(N_F)(x_{N+1,j} - \underline{X}_{N+1,j}w_j(N_F))}{1 + \underline{X}_{N+1,j}P_j(N_F)\underline{X}_{N+1,j}^T} \underline{X}_{N+1,j}^T, \\ P_j(N_F + 1) = P_j(N_F) - \frac{P_j(N_F)\underline{X}_{N+1,j}^T \underline{X}_{N+1,j} P_j(N_F)}{1 + \underline{X}_{N+1,j}P_j(N_F)\underline{X}_{N+1,j}^T}, \end{cases} \quad (1.9)$$

після чого уточнюються і відновлені значення ϵ_{kj} .

Якщо ж $(N + 1)$ -ий рядок містить пропуск, він пропускається і алгоритм (1.9) очікує появи повністю заповненого рядка, наприклад x_{N+2} , після чого розраховується $w_j(N_F + 1)$ за допомогою $(N + 2)$ -го спостереження і коригуються всі значення ϵ_{kj} , включаючи $\epsilon_{N+1,j}$.

На початкових етапах обробки таблиці 1.1, коли кількість повністю заповнених рядків N_F сумірна з кількістю стовпців n , оцінки, отримані за допомогою методу найменших квадратів, характеризуються низькою точністю. У цій ситуації для послідовної обробки ефективнішим є застосування адаптивних алгоритмів навчання, що мають як фільтруючі, так і слідкуючі (для нестационарних ситуацій) властивості [49, 50].

Даний метод можна переписати із застосуванням адаптивних алгоритмів навчання у вигляді

$$\begin{cases} w_j(N_F + 1) = w_j(N_F) + r_j^{-1}(N_F + 1)(x_{N+1,j} - \underline{\mathbf{X}}_{N+1,j} w_j(N_F)) \underline{\mathbf{X}}_{N+1,j}^T, \\ r_j(N_F + 1) = \alpha r_j(N_F) + \|\underline{\mathbf{X}}_{N+1,j}\|^2, \end{cases} \quad (1.10)$$

де $0 \leq \alpha \leq 1$ – параметр згладжування, який задає компроміс між згладженням і слідкуванням за характеристиками об'єктів, що змінюються.

Обробку інформації в режимі послідовного надходження даних зручно організувати на основі нейромережевої системи, основними елементами якої є n паралельно працюючих адаптивних лінійних асоціаторів (ALA-Adaline) [51], що навчаються за допомогою формули (1.10).

На рисунку 1.7 наведена схема цієї системи, яка не потребує додаткових пояснень.

Зауважимо тільки, що індекс k тут позначає номер об'єкта, характеристики якого в поточний момент часу подаються на обробку.

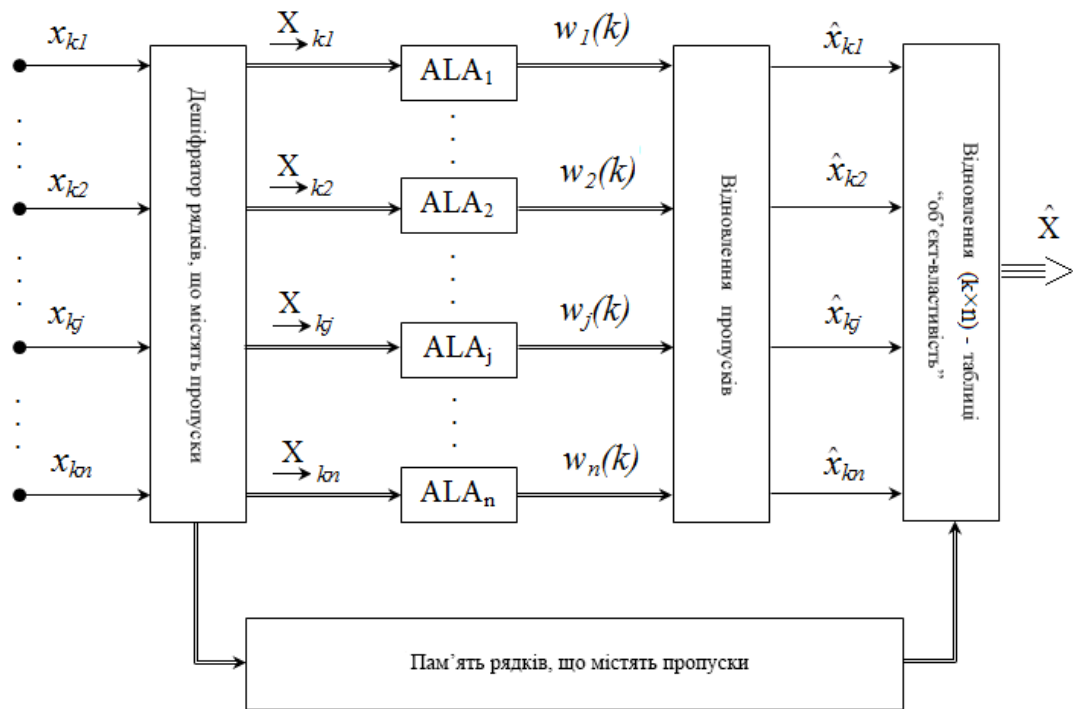


Рисунок 1.7 – Адаптивна неймережева система для відновлення пропусків

1.1.6 Адаптивне відновлення даних з пропусками на основі ансамблю нео-фаззі нейронів

Розглянута в попередньому розділі задача відновлення даних з пропусками на основі ансамблю ALA – Адалін була вирішена у припущенні, що між стовпцями \vec{x}_j матриці «об'єкт-властивість» (Таблиця 1.1) існує лінійна кореляція, що описується рівнянням (1.5), параметри якого оцінюються за допомогою або стандартного методу найменших квадратів (1.7), або його рекурентного варіанта (1.9), або адаптивного алгоритму навчання (1.10).

Ситуація різко ускладнюється, якщо зв'язок між стовпцями \vec{x}_j не може бути адекватно описаний лінійними співвідношеннями, а має більш складний, нелінійний характер, при чому характер цієї нелінійності апріорі невідомий. Звісно, в загальному випадку довільний тип нелінійності може

бути наскільки завгодно точно відновлений за допомогою нейромережових технологій [45, 51], однак, по-перше, штучні нейронні мережі потребують для свого навчання достатньо великих за об'ємом навчальних вибірок, нехай і з пропусками [44], а, по-друге, різко ускладнюється реалізація системи відновлення і знижується її швидкодія, не кажучи вже про неможливість роботи такої мережі в on-line режимі.

Подолати вказані труднощі можна, скориставшись у якості «будівельних блоків» системи відновлення, так званими, нео-фаззі нейронами [52-54], які, з одного боку, є нейро-фаззі системами, що мають високі апроксимуючі властивості, а, з іншого боку – можуть розглядатися як нелінійні розширення Адалін, для яких можуть бути використані оптимальні за швидкістю лінійні алгоритми навчання.

Структурна схема нео-фаззі нейрона наведена на рис. 1.8, і, як видно, крім синаптичних ваг w_{li} , що налаштовуються, вона містить нелінійні функції належності $\mu_{li}(x_i)$, що забезпечують нео-фаззі нейрону необхідні апроксимуючі властивості.

При подачі на вхід нео-фаззі нейрона векторного сигналу $\tilde{x}_k = (\tilde{x}_{k1}, \tilde{x}_{k2}, \dots, \tilde{x}_{kn})^T$ (тут вхідні дані попередньо перетворюються так, що $0 \leq \tilde{x}_{ki} \leq 1$ для всіх $1 \leq k \leq N$ і $1 \leq i \leq n$), на його виході з'являється скалярне значення

$$y_k = \sum_{i=1}^n f_i(\tilde{x}_{ki}) = \sum_{i=1}^n \sum_{l=1}^h w_{li}(k-1) \mu_{li}(\tilde{x}_{ki}), \quad (1.11)$$

що визначається як синаптичними вагами $w_{li}(k-1)$ (подібно Адаліні), що налаштовуються, так і функціями належності $\mu_{li}(\tilde{x}_{ki})$, у якості яких можуть бути використані будь-які прийняті в теорії нечітких систем конструкції: трикутні, дзвонуваті, сплайн-функції тощо.

В якості критерію навчання нео-фаззі нейрона використовується та сама квадратична функція, що й для Адаліні:

$$E(k) = \frac{1}{2}(d_k - y_k)^2 = \frac{1}{2}e_k^2 = \frac{1}{2}\left(d_k - \sum_{i=1}^n \sum_{l=1}^h w_{li}(k-1)\mu_{li}(\tilde{x}_{ki})\right)^2, \quad (1.12)$$

що мінімізується за допомогою градієнтного алгоритму

$$w_{li}(k+1) = w_{li}(k) + \eta e_k \mu_{li}(\tilde{x}_{ki}), \quad (1.13)$$

де η – скалярний параметр кроку пошуку, питання вибору якого авторами нео-фаззі нейрона не розглядалося.

Введемо в розгляд $(hn \times 1)$ -вектори функції належності

$$\mu(\tilde{x}_k) = (\mu_{11}(\tilde{x}_{k1}), \mu_{21}(\tilde{x}_{k1}), \dots, \mu_{h1}(\tilde{x}_{k1}), \mu_{21}(\tilde{x}_{k2}), \dots, \mu_{hm}(\tilde{x}_{kn}))^T$$

і синаптичних ваг $w = (w_{11}, w_{21}, \dots, w_{h1}, w_{12}, \dots, w_{hm})^T$ і перепишемо рівняння (1.11) у векторній формі (1.14), яка має ту саму структуру, що і відображення (1.4), що реалізується Адаліною.

$$y_k = w^T(k-1)\mu(\tilde{x}_k), \quad (1.14)$$

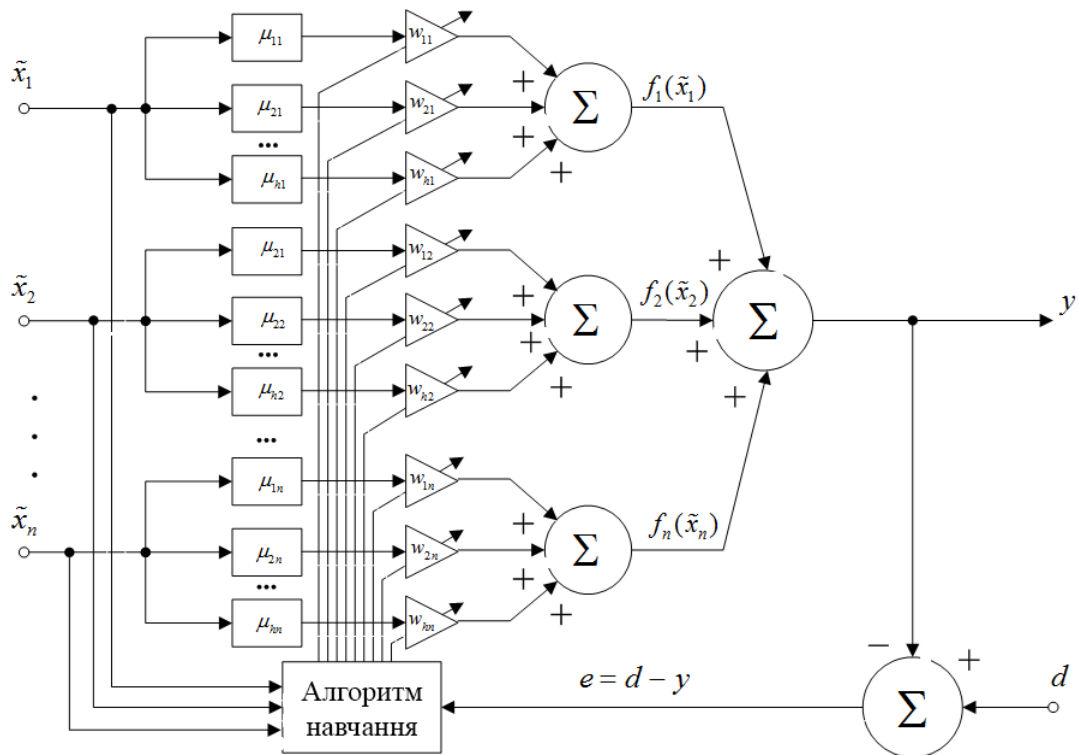


Рисунок 1.8 – Нео-фаззі нейрон

Неважко помітити, що оцінювання вектора синаптичних ваг може бути проведено як за допомогою стандартного методу найменших квадратів (якщо таблиця 1.1 задана апріорно), так і в послідовному режимі за допомогою алгоритму типу (1.10), який у даному випадку набуває вигляду

$$\begin{cases} w(k+1) = w(k) + r^{-1}(k+1)(d_{k+1} - w^T(k)\mu(\tilde{x}_{k+1}))\mu(\tilde{x}_{k+1}), \\ r(k+1) = \alpha r(k) + \|\mu(\tilde{x}_{k+1})\|^2 \end{cases} \quad (1.15)$$

для окремого нео-фаззі нейрона i

$$\begin{cases} w_j(N_F + 1) = w_j(N_F) + r^{-1}(N_F + 1)(\tilde{x}_{N+1,j} - \mu(\tilde{x}_{N+1,j})w_j(N_F))\mu^T(\tilde{x}_{N+1,j}), \\ r_j(N_F + 1) = \alpha r(N_F) + \|\mu(\tilde{x}_{N+1,j})\|^2 \end{cases} \quad (1.16)$$

- для нео-фаззі нейрона у складі адаптивної нейро-фаззі системи для відновлення пропусків, що наведена на рис. 1.9.

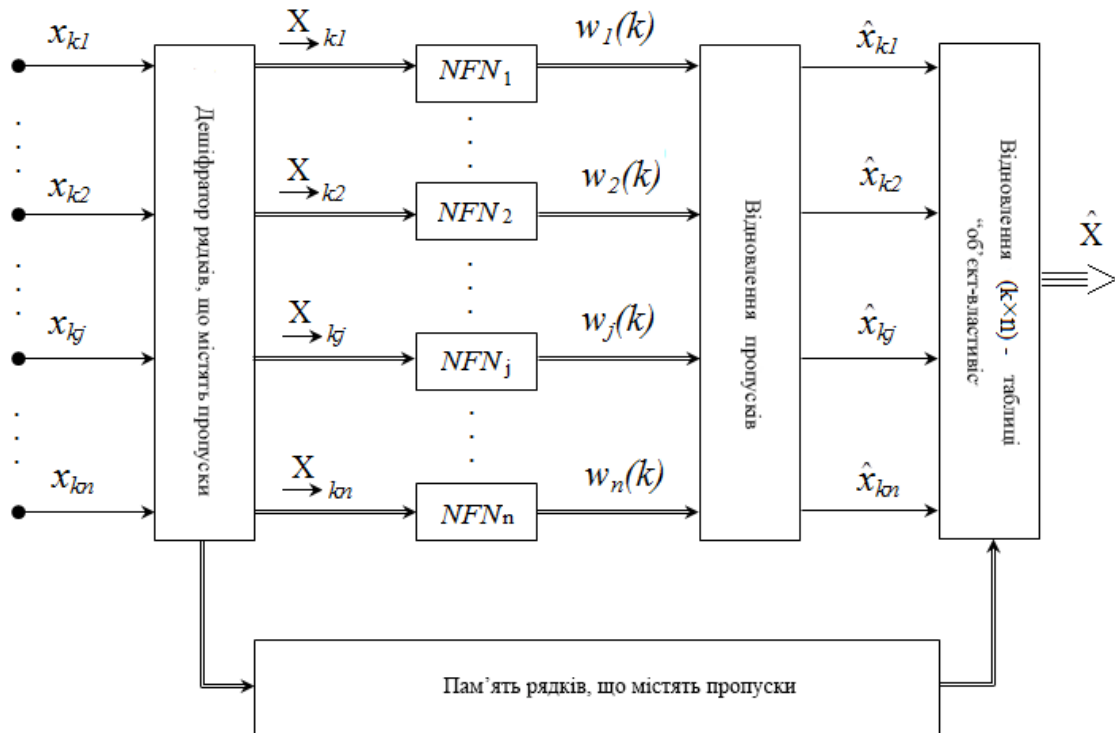


Рисунок 1.9 – Адаптивна нейро-фаззі система для відновлення пропусків

Неважко помітити, що ця ж система є розширенням адаптивної нейромережевої системи, наведеної на рис. 1.7, в якій Адаліни (ALA_j) замінені на нелінійні нео-фаззі нейрони (NFN_j).

Як видно, нелінійне розширення задачі з теоретичної точки зору жодних проблем не викликає, а «платою» за нелінійні властивості є у h разів більша у порівнянні з системою на рис. 1.7 кількість синаптичних ваг, які налаштовуються, що у ряді випадків веде до зниження швидкодії процесу обробки інформації.

1.1.7 Метод багатовимірної нечіткої екстраполяції

Метод багатовимірної лінійної екстраполяції (БЛЕ) дозволяє вирішити широке коло задач проектування і оперативного керування і за своєю евристичною математичною суттю і універсальністю знаходиться в одному

рангу з такими математичними парадигмами, як метод сплайнів, метод групового урахування аргументів, нейронні мережі. За простотою реалізації, можливістю використання в адаптивних системах і високою ефективністю у ряді випадків БЛЕ займає більш високі позиції.

В основі вихідного алгоритму БЛЕ покладено припущення про те, що простір спостережень сигналу, на якому відбувається навчання системи, є фіксованим (нерухомим). Однак в нашому випадку необхідно обробляти дані, що надходять на обробку послідовно.

Представимо таблицю 1.1 у вигляді $N \times n$ -матриці X , в якій в найпростішому випадку відсутній елемент x_{ip} , або, в більш загальному випадку, $\sum_{i=1}^N n_i$ елементів. Усі інші наявні в матриці елементи перетворюються так, щоб спостереження, які підлягають обробці, належали гіперкубу $[-1,1]^n$, тобто $-1 \leq \tilde{x}_{kp} \leq 1, k = 1, 2, \dots, N; p = 1, 2, \dots, n$. Для кожного рядка \tilde{x}_i , що містить пропуски, розраховуються відстані між ним і всіма іншими рядками, використовуючи поняття «часткової відстані» (PD), прийнятого в нечіткій кластеризації [55] і модифікованого у вигляді

$$D_P^2(\tilde{x}_i, \tilde{x}_k) = \frac{n}{n_i + n_k - n_{ik}} \sum_{j=1}^n (\tilde{x}_{ij} - \tilde{x}_{kj})^2 \delta_j,$$

де $\delta_j = \begin{cases} 0, & \text{якщо чи у } \tilde{x}_i, \text{ чи у } \tilde{x}_k \text{ в } j \text{ позиції є пропуск,} \\ 1 & \text{в протилежному випадку,} \end{cases}$

n_{ik} – кількість загальних пропусків в одній і тій самій позиції в \tilde{x}_i і \tilde{x}_k .

При цьому з розглядання виключаються ті \tilde{x}_k , для яких $\sum_{j=1}^n \delta_j = 0$.

Проаналізуємо далі $\tilde{N} \leq N - 1$ розрахованих відстаней так, щоб

$$0 \leq D_P^{2[\min]} = D_P^{2[1]} < D_P^{2[2]} < \dots < D_P^{2[\tilde{N}]} \leq 4n$$

(тут індекс у квадратних дужках означає ранг), і залишимо для подальшої обробки тільки $\tilde{N} \leq \tilde{N} \leq N - 1$ спостережень, для яких

$$\frac{D_p^{2[l]}}{4n} \leq \varepsilon, \quad l = 1, 2, \dots, \tilde{N},$$

де ε – деякий поріг ($0 < \varepsilon < 1$).

Використовуючи поняття належності, прийняте у стандартному методі с-середніх [56], розрахуємо рівень близькості \tilde{x}_i до \tilde{N} векторів \tilde{x}_l , що залишилися в розгляді, у вигляді

$$U_l(i) = \frac{D_p^{-2[l]}}{\sum_{q=1}^{\tilde{N}} D_p^{-2[q]}}, \quad l = 1, 2, \dots, \tilde{N},$$

при цьому, якщо $D_p^{2[1]} = 0$, то автоматично отримуємо $U_1(i) = 1$.

Таким чином, кожний вектор \tilde{x}_i апроксимується виразом

$$\underline{x}_i = \sum_{l=1}^{\tilde{N}} U_l(i) \tilde{x}_l. \quad (1.17)$$

Зауважимо, що у БЛЕ також використовується апроксимація типу (1.17), однак замість рівнів належності $U_l(i)$ використовуються вагові коефіцієнти, які отримуються в результаті вирішення задачі оптимізації, яка до того ж не завжди є такою, що можна розв'язати.

І, нарешті, останній етап – власне заповнення пропусків. Нескладно побачити, що оцінка пропущеного елемента \tilde{x}_{ip} може бути записана у вигляді

$$\underline{x}_{ip} = \sum_{l=1}^{\mathcal{N}} U_l(i) \tilde{x}_{lp}.$$

Запропонований метод зручно представити у вигляді схеми, наведеної на рисунку 2.10.

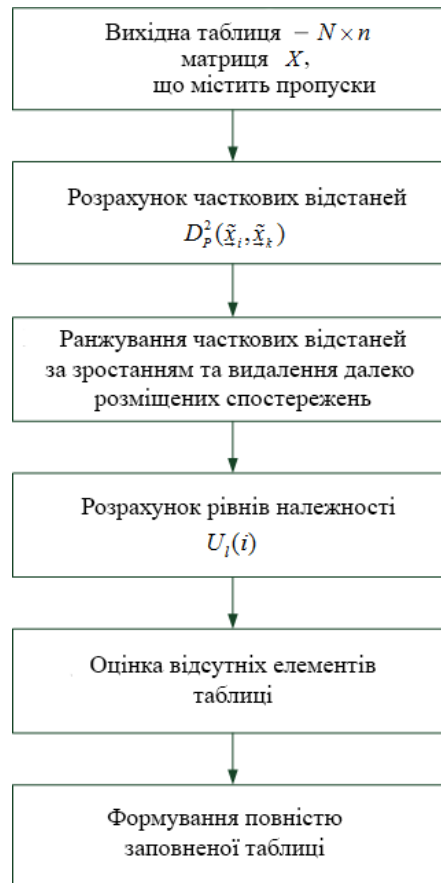


Рисунок 1.10 – Метод багатовимірної нечіткої екстраполяції в задачі відновлення пропусків

1.1.8 Відновлення викривлених даних в часових рядах

Аналіз часових рядів поєднує методи вивчення часових рядів, як ті, що намагаються зрозуміти природу точок даних, так і ті, що намагаються побудувати прогноз. Прогнозування часових рядів полягає в побудованні моделі для передбачення майбутніх подій, ґрунтуючись на відомих подіях минулого, передбачення майбутніх даних до того, як вони будуть виміряні.

Модель часового ряду в загальному сенсі відображує ідею, що близькі у часі спостереження будуть тісніше пов'язані, ніж віддалені. Крім того, моделі часових рядів часто використовують однонаправлений порядок за часом в тому сенсі, що значення в ряді виражаються в деякому вигляді через попередні значення, а не через наступні.

Методи аналізу часових рядів часто поділяють на два класи:

- аналіз у частотній області,
- аналіз у часовій області.

Перший ґрунтується на спектральному аналізі і останнім часом вейвлет-аналізі, і може розглядатися в якості методів аналізу, що не використовують моделі, які добре підходять для досліджень на етапі розвідки.

Методи аналізу у часовій області також мають безмодельну підмножину, яка складається з крос-кореляційного аналізу та автокореляційного аналізу, але саме тут з'являються частково або повністю визначені моделі часових рядів.

Як показано Боксом і Дженкінсом, моделі часових рядів можуть мати різні форми і представляти різні стохастичні процеси. При моделюванні змін рівня процесу можна виділити три широких класи, що мають практичну цінність:

- авторегресійні моделі,
- інтегральні моделі,
- моделі ковзного середнього.

Ці три класи лінійно залежать від попередніх даних. На їх основі побудовані моделі авторегресійного ковзного середнього (Autoregressive Moving Average, ARMA) і авторегресійного інтегрованого ковзного середнього (Autoregressive Integrated Moving Average, ARIMA). Ці моделі в свою чергу узагальнює модель авторегресійного дрібноінтегрованого

ковзного середнього (autoregressive fractionally integrated moving average, ARFIMA).

Розширення моделей на випадки, коли дані представляються не скалярно, а векторно, називають моделями багатовимірних часових рядів. Для таких моделей у скорочених назвах з'являється літера «v» від слова «vector». Існують розширення моделей на випадок, коли часовий ряд, що досліджується, є веденим для деякого ряду, що «змушує» (який, однак, може не бути причиною виникнення ряду, що досліджується). Відмінність від багатовимірного ряду полягає в тому, що ряд, що змушує, може бути детермінованим або керуватися дослідником, що проводить експеримент. Для таких моделей у скороченні з'являється літера «x» від «exogenous» (екзогенний, викликаний зовнішніми причинами).

Нелінійна залежність рівня ряду від попередніх точок цікава, почасти через можливість генерації хаотичних часових рядів. Але головним все ж є те, що досвідчені дослідники вказують на перевагу прогнозів, отриманих від нелінійних моделей, над прогнозами лінійних моделей.

Серед інших типів нелінійних моделей часових рядів можна виділити моделі, які описують зміни дисперсії ряду з часом (гетероскедатичність). Такі моделі називають моделями авторегресійної умовної гетероскедатичності (AutoRegressive Conditional Heteroscedasticity, ARCH). У цих моделях зміни дисперсії пов'язують з найближчими попередніми даними. Противагою такому підходу є представлення локально мінливої дисперсії, при якому дисперсія може бути змодельована залежною від окремого мінливого з часом процесу, як у бістохастичних моделях.

Останнім часом значну увагу отримали дослідження в області безмодельного аналізу і методи, що базуються на вейвлетних перетвореннях (наприклад, локально стаціонарні вейвлети) зокрема. Методи кратномасштабного аналізу розкладають заданий часовий ряд на складові частини, щоб показати залежність від часу з різним масштабом.

В основі підходу до відновлення отриманих спостережень, що пропонується, лежить використання класичних ортогональних поліномів [57] і, насамперед, поліномів Чебишева (Т-систем) [58, 59], що мають цілий ряд корисних властивостей з позиції задачі апроксимації за квадратичним критерієм [58].

У загальному випадку поліноми Чебишева можуть бути компактно представлені за допомогою тригонометричних функцій

$$T_l(x) = \cos(l \arccos x), \quad x \in [-1, 1], \quad l = 0, 1, 2, \dots \quad (1.18)$$

або за допомогою рекурентного співвідношення

$$T_l(x) = 2xT_{l-1}(x) - T_{l-2}(x). \quad (1.19)$$

У ряді випадків зручніше вирішувати задачу на інтервалі $x \in [0, 1]$, для чого можуть бути використані зміщені поліноми Чебишева:

$$T_l^B(x) = T_l(2x - 1), \quad (1.20)$$

$$T_l^B(x) = (4x - 2)T_{l-1}^B(x) - T_{l-2}^B(x). \quad (1.21)$$

Нескладно також записати зміщені поліноми для довільного інтервалу $x \in [x_{\min}, x_{\max}]$.

Для апроксимації довільної функції скалярного аргументу $y = f(x)$, заданої на деякій множині вузлів x_1, x_2, \dots, x_N ; y_1, y_2, \dots, y_N , перетворимо вихідні дані за допомогою простих співвідношень

$$\tilde{x}_k = 2 \frac{x_k - x_{\min}}{x_{\max} - x_{\min}} - 1;$$

$$\tilde{y}_k = 2 \frac{y_k - y_{\min}}{y_{\max} - y_{\min}} - 1, \quad k = 1, 2, \dots, N$$

для поліномів (1.18), (1.19) і

$$\tilde{x}_k = \frac{x_k - x_{\min}}{x_{\max} - x_{\min}}; \quad \tilde{y}_k = \frac{y_k - y_{\min}}{y_{\max} - y_{\min}}, \quad k = 1, 2, \dots, N$$

для (1.20), (1.21) і запишемо апроксимуючу Т-систему у вигляді

$$T_h^\Sigma(\tilde{x}) = \sum_{l=0}^h w_l T_l(\tilde{x}),$$

параметри w_l якої визначаються простим співвідношенням

$$w_l = \frac{\sum_{k=1}^N \tilde{y}_k T_l(\tilde{x}_k)}{\sum_{k=1}^N T_l^2(\tilde{x}_k)},$$

яке є з врахуванням умов ортогональності

$$\sum_{k=1}^N T_l(\tilde{x}_k) T_j(\tilde{x}_k) = 0 \quad \text{при } k \neq j$$

є стандартною оцінкою найменших квадратів. При цьому значення $T_l(\tilde{x}_k)$ розраховуються за допомогою рекурентних співвідношень

$$T_0(\tilde{x}_k) = \tilde{x}_k - b_1,$$

$$T_l(\tilde{x}_k) = (\tilde{x}_k - b_l)T_{l-1}(\tilde{x}_k) - a_l T_{l-2}(\tilde{x}_k),$$

$$a_l = \frac{\sum_{k=1}^N \tilde{x}_k^{l-1} T_{l-1}(\tilde{x}_k)}{\sum_{k=1}^N \tilde{x}_k^{l-2} T_{l-2}(\tilde{x}_k)},$$

$$b_l = \frac{\sum_{k=1}^N \tilde{x}_k^l T_{l-1}(\tilde{x}_k)}{\sum_{k=1}^N \tilde{x}_k^{l-1} T_{l-1}(\tilde{x}_k)} - \frac{\sum_{k=1}^N \tilde{x}_k^{l-1} T_{l-2}(\tilde{x}_k)}{\sum_{k=1}^N \tilde{x}_k^{l-2} T_{l-2}(\tilde{x}_k)},$$

а точність апроксимації, що досягається, – на основі дисперсії

$$\sigma_h^2 = \sigma_{h-1}^2 - w_h^2 \sum_{k=1}^N \tilde{x}_k^h T_h(\tilde{x}_k),$$

$$\sigma_0^2 = \sum_{k=1}^N \tilde{y}_k^2 - \frac{1}{N} \left(\sum_{k=1}^N \tilde{y}_k \right)^2.$$

Розглянемо далі часовий ряд \tilde{x}_k , $k = 1, 2, \dots, N$ і введемо апроксимуючу Т-систему

$$T_h^\Sigma(k) = \sum_{l=0}^h w_l T_l(k), \quad (1.22)$$

при цьому будемо припускати, що в деякі моменти дискретного часу k вимірювання або не були проведені, або були загублені.

Введемо в розгляд дві підмножини: $X_P = \{\tilde{x}_k, k\}$, що містить $N_P \leq N$ наявних спостережень, і $X_G = \{k\}$, що містить моменти часу з відсутніми спостереженнями.

Тоді коефіцієнти апроксимуючого полінома можна розрахувати за допомогою співвідношень:

$$w_l = \frac{\sum_{\substack{k=1 \\ k \in X_P}}^N \tilde{x}_k T_{lN}(k)}{\sum_{\substack{k=1 \\ k \in X_P}}^N T_{lN}^2(k)},$$

$$w_0 = \frac{1}{N_P} \sum_{\substack{k=1 \\ k \in X_P}}^N \tilde{x}_k,$$

де індекс N в T_{lN} означає, що відповідні поліноми ортогональні на інтервалі $k \in [1, N]$.

Далі нескладно відновити відсутні спостереження у вигляді

$$\mathfrak{E}_k = \sum_{l=0}^h w_l T_{lN}(k) \quad \forall k \in X_G.$$

Поєднуючи наявні ($\tilde{x}_k \in X_P$) і відновлені ($\mathfrak{E}_k \in X_G$) значення часового ряду, можна сформулювати послідовність, яка містить всі N значень, що позначається далі $\mathfrak{E}_1, \mathfrak{E}_2, \dots, \mathfrak{E}_N$.

Введемо далі вектор параметрів $w = (w_0, w_1, \dots, w_h)^T$ і Т-систему ортогональних поліномів $T_N(k) = (T_{0N}(k), T_{1N}(k), \dots, T_{hN}(k))^T$ і перепишемо (1.22) у векторній формі

$$T_h^\Sigma(k) = w^T T_N(k),$$

де вектор w розраховується за допомогою стандартного методу найменших квадратів

$$w(N) = \left(\sum_{\substack{k=1 \\ k \in X_p}}^N T_N(k) T_N^T(k) \right)^{-1} \sum_{\substack{k=1 \\ k \in X_p}}^N T_N(k) \tilde{x}_k = P(N) \sum_{\substack{k=1 \\ k \in X_p}}^N T_N(k) \tilde{x}_k, \quad (1.23)$$

після чого можна остаточно записати

$$\begin{cases} \mathfrak{E}_k = w^T(N) T_N(k) \forall k \in X_G, \\ \mathfrak{E}_k = \tilde{x}_k \forall k \in X_P. \end{cases} \quad (1.24)$$

Використання співвідношень (2.23), (2.24) припускає обробку даних в пакетному режимі при фіксованій кількості точок N . Якщо ж дані надходять на обробку послідовно, варто організувати on-line обробку інформації. В [59] для цього пропонується використовувати рекурентний метод найменших квадратів у формі

$$\begin{cases} w(N) = w(N-1) + \frac{P(N-1)(\tilde{x}_N - w^T(N-1)T_N(N))}{1 + T_N^T(N)P(N-1)T_N(N)} T_N(N), \\ P(N) = P(N-1) - \frac{P(N-1)T_N(N)T_N^T(N)P(N-1)}{1 + T_N^T(N)P(N-1)T_N(N)}, \end{cases} \quad (2.25)$$

однак при цьому необхідно відзначити, що з приходом нового $(N+1)$ -го спостереження \tilde{x}_{N+1} , суттєвим чином змінюється структура апроксимуючих поліномів так, що $T_N(k) \neq T_{N+1}(k)$. Зрозуміло, що цей факт суттєво ускладнює процес on-line обробки.

Щоб зберегти структуру апроксимуючих поліномів, можна організувати обробку даних на ковзному вікні s , при цьому в кожний момент

часу k в цьому вікні можуть бути присутніми як всі s спостережень, так і пропуски.

Введемо в розгляд Т-систему поліномів на ковзному вікні s , яка відрізняється від розглянутих вище тим, що замість реального часу k ми будемо використовувати машинний час $\tilde{k} = 1, 2, \dots, s$, пов'язаний з реальним так, що в розрахунках використовуються тільки моменти $k - s + 1$, $k - s + 2, \dots, k$, при цьому

$$\left\{ \begin{array}{l} T_{0,s}(\tilde{k}) = \frac{1}{\sqrt{s}}, \\ T_{1,s}(\tilde{k}) = \sqrt{\frac{3}{s(s^2-1)}}(2\tilde{k} - s - 1), \\ T_{l,s}(\tilde{k}) = (a_l \tilde{k} + b_l)T_{l-1,s}(\tilde{k}) - c_l T_{l-2,s}(\tilde{k}), l = 2, 3, \dots, h, \\ a_l = \frac{2}{l} \sqrt{\frac{d_l(d_l-2)}{g_l}}, \\ b_l = -\frac{s+1}{l} \sqrt{\frac{d_l(d_l-2)}{g_l}} = -\frac{s+1}{2} a_l, \\ c_l = \frac{l-1}{l} \sqrt{\frac{d_l(d_l+g_l-2)}{(d_l-4)g_l}}, \\ d_l = 2l+1, \\ g_l = s^2 - l^2. \end{array} \right. \quad (1.26)$$

Тоді оцінка типу (1.23) з фіксованою структурою поліномів $T_s(N)$ може бути записана у вигляді

$$w_s(N) = \left(\sum_{\substack{\tilde{k}=1, \\ k=N-s+1 \\ k \in X_p}}^{s,N} T_s(\tilde{k}) T_s^T(\tilde{k}) \right)^{-1} \sum_{\substack{\tilde{k}=1, \\ k=N-s+1 \\ k \in X_p}}^{s,N} T_s(\tilde{k}) \tilde{x}_k = P_s(N) \sum_{\substack{\tilde{k}=1, \\ k=N-s+1 \\ k \in X_p}}^{s,N} T_s(\tilde{k}) T_s^T(\tilde{k}), \quad (1.27)$$

а оцінки відсутніх спостережень

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{\tilde{k}} = \mathbf{w}_s^T(N) T_s(\tilde{k}) \quad \forall k \in X_G.$$

Варто відзначити також, що в силу ортогональності поліномів Чебишева, матриці $P_s(N)$ є діагональними (або наближаються до них), завдяки чому забезпечується чисельна простота і стійкість оцінки (1.30), що дозволяє розраховувати її в on-line режимі в будь-який момент реального часу $k > s$.

Реалізувати описаний підхід досить зручно за допомогою апарата штучних нейронних мереж, використовуючи, так званий, орто-синапс [60], що наведений на рис. 1.11

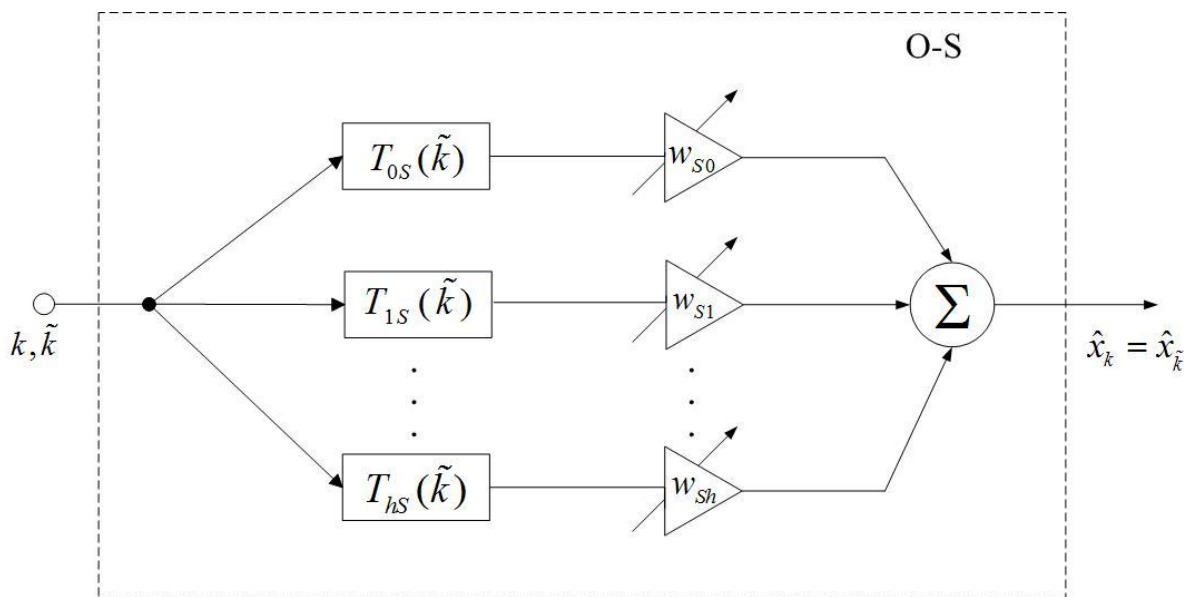


Рисунок 1.11 – Орто-синапс

і що навчається за допомогою алгоритму (2.30). Орто-синапс за структурою співпадає з нелінійним синапсом нео-фаззі нейрона [53, 54], однак замість функцій належності містить ортогональні активаційні функції T_{ls} , що робить процес навчання більш простим і швидким. Важливо також, що завдяки використанню ковзного вікна, ці активаційні функції не змінюються у

процесі навчання. Розмір ковзного вікна обирається з емпіричних міркувань $s \geq h + 1$, при цьому якщо послідовність, що апроксимується, є нестационарною, значення s не повинно суттєво перевищувати кількість оцінок параметрів.

1.1.9 Архітектура ортогональної нейронної мережі

Використання у якості активаційних функцій ортогональних поліномів привело до створення цілої групи ортогональних нейронних мереж [61-69], що мають хороші апроксимуючі властивості і високу швидкість налаштування синаптичних ваг. В [70-73] були запропоновані ортогональні мережі, що ростуть, які побудовані на основі орто-синапсів і орто-нейронів [60] та які характеризуються простотою навчання як синаптичних ваг, так і архітектури.

На рисунку 1.12 наведена архітектура ортогональної нейронної мережі для обробки даних з втраченими спостереженнями, яка реалізує нелінійне відображення

$$\mathfrak{E}_k = \mathcal{F}(\mathfrak{E}_k) = \sum_{i=1}^n \sum_{l=0}^h w_{li} T_{li}(\mathfrak{E}_{ki}), \quad (1.28)$$

де $k = 1, 2, \dots$ - поточний дискретний час,

w_{li} – синаптичні ваги, що налаштовуються,

$T_{li}(\bullet)$ – l -а ортогональна активаційна функція типу (1.18) або (1.20) для відповідного сигналу $\mathfrak{E}_{ki}, i = 1, 2, \dots, n$.

На нульовий (рецепторний) шар мережі послідовно надходять спостереження \tilde{x}_{ki} , які містять апріорі невідому кількість пропусків. Зауважимо також, що зовнішній навчальний сигнал \tilde{y}_k також може містити пропуски.

Перший прихований шар мережі утворений n орто-синапсами $O-S_i^M$ і слугує для відновлення відсутніх спостережень так, як це було описано в попередньому розділі.

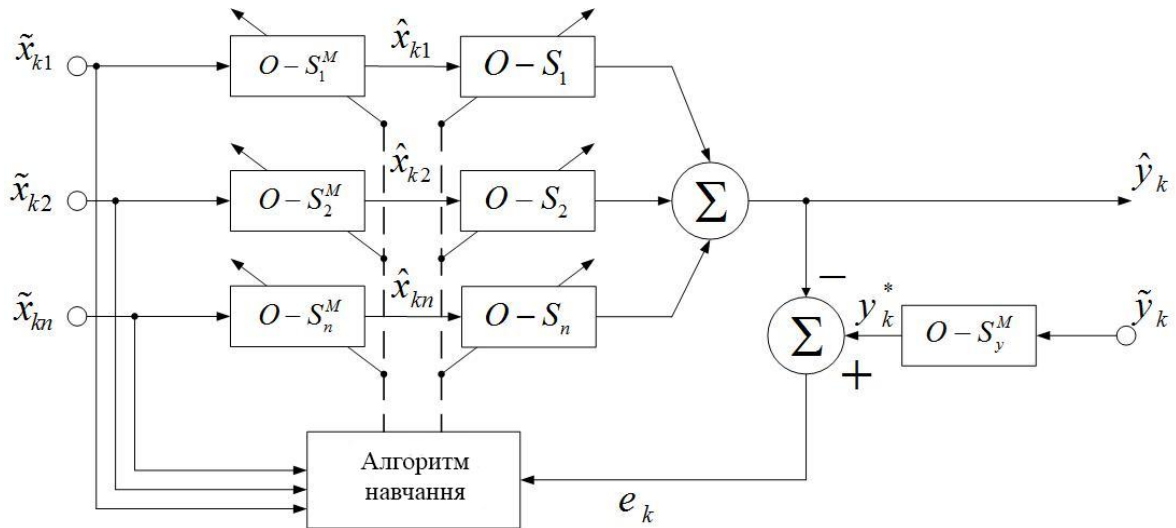


Рисунок 1.12 – Ортогональна нейронна мережа для обробки даних з втраченими спостереженнями

Ту саму функцію для навчального сигналу \tilde{y}_k виконує орто-синапс $O-S_y^M$. У результаті в on-line режимі формуються часові ряди ϵ_{ki}, y_k^* . Вихідний шар утворений орто-нейроном, який співпадає за архітектурою з нео-фаззі нейроном, але містить ортогональні активаційні функції замість звичайних функцій належності. Похибка

$$e_k = y_k^* - \hat{y}_k$$

використовується алгоритмом навчання для налаштування як ваг, так і архітектури.

Наведена архітектура містить $2n+1$ орто-синапсів і $(h+1)(2n+1)$ синаптичних ваг, які підлягають визначенню, при цьому, що дуже важливо, вихідний сигнал \mathfrak{f}_k лінійно залежить від ваг w_{li} .

Введемо в розгляд стандартний критерій навчання

$$E_k = \sum_{k=1}^N e_k^2 = \sum_{k=1}^N \left(y_k^* - \sum_{i=1}^n \sum_{l=0}^h w_{li} T_{li}(\mathfrak{f}_{ki}) \right)^2,$$

$(h+1)n \times 1$ -вектори

активаційних

функцій

$$T(\mathfrak{f}_k) = (T_{01}(\mathfrak{f}_{k1}), T_{11}(\mathfrak{f}_{k1}), \dots, T_{h1}(\mathfrak{f}_{k1}), T_{02}(\mathfrak{f}_{k2}), \dots, T_{h2}(\mathfrak{f}_{k2}), \dots, T_{hn}(\mathfrak{f}_{kn}))^T$$

$$= (T_1^T(\mathfrak{f}_{k1}), T_2^T(\mathfrak{f}_{k2}), \dots, T_n^T(\mathfrak{f}_{kn}))^T$$

і

синаптичних

ваг

$$w = (w_{01}, w_{11}, \dots, w_{h1}, w_{02}, w_{h2}, \dots, w_{hn})^T = (w_1^T, w_2^T, \dots, w_n^T)^T$$

і перепишемо вихідний сигнал мережі (1.28) у компактній формі

$$\mathfrak{f}_k = w^T T(\mathfrak{f}_k) = \sum_{i=1}^n w_i^T T_i(\mathfrak{f}_k). \quad (1.29)$$

Для налаштування вектора синаптичних ваг w в послідовному режимі в принципі можна використовувати рекурентний метод найменших квадратів (1.25), однак при обробці нестационарних сигналів доцільніше скористатися адаптивними процедурами, які мають слідкуючі властивості, наприклад, тим самим методом найменших квадратів на ковзному вікні, який в даній ситуації може бути записаний у вигляді

$$\left\{ \begin{array}{l} \tilde{w}(N) = w(N-1) + \frac{P(N-1)(y_N^* - w^T(N-1)T(\boldsymbol{\epsilon}_N))}{1 + T^T(\boldsymbol{\epsilon}_N)P(N-1)T(\boldsymbol{\epsilon}_N)} T(\boldsymbol{\epsilon}_N), \\ \tilde{P}(N) = P(N-1) - \frac{P(N-1)T(\boldsymbol{\epsilon}_N)T^T(\boldsymbol{\epsilon}_N)P(N-1)}{1 + T^T(\boldsymbol{\epsilon}_N)P(N-1)T(\boldsymbol{\epsilon}_N)}, \\ P(N) = \tilde{P}(N) + \frac{\tilde{P}(N)T(\boldsymbol{\epsilon}_{N-s})T^T(\boldsymbol{\epsilon}_{N-s})\tilde{P}(N)}{1 - T^T(\boldsymbol{\epsilon}_{N-s})\tilde{P}(N)T(\boldsymbol{\epsilon}_{N-s})}, \\ w(N) = \tilde{w}(N) - \frac{\tilde{P}(N)(y_{N-s}^* - \tilde{w}^T(N)T(\boldsymbol{\epsilon}_{N-s}))}{1 - T^T(\boldsymbol{\epsilon}_{N-s})\tilde{P}(N)T(\boldsymbol{\epsilon}_{N-s})} T(\boldsymbol{\epsilon}_{N-s}), \end{array} \right. \quad (1.30)$$

або

$$w(N) = \left(\sum_{k=N-s+1}^N T(\boldsymbol{\epsilon}_k)T^T(\boldsymbol{\epsilon}_k) \right)^{-1} \sum_{k=N-s+1}^N T(\boldsymbol{\epsilon}_k)y_k^* = P_s(N) \sum_{k=N-s+1}^N T(\boldsymbol{\epsilon}_k)y_k^*, \quad (1.31)$$

при цьому, завдяки діагональності матриці $P_s(N)$, розрахунок оцінки (1.31) навіть при великих n та h не викликає труднощів.

Кількість активаційних функцій $h+1$ в кожному орто-синапсі $O-S_i$ вихідного шару обирається достатньо довільно, тому, якщо буде встановлено, що синтезована нейронна мережа не забезпечує потрібну якість обробки інформації, кількість цих функцій може бути збільшена (або зменшена, якщо необхідно) в on-line режимі безпосередньо в процесі навчання. Таким чином мережа набуває еволюційних властивостей, адаптуючи свою структуру, при цьому завдяки ортогональності активаційних функцій, корекція для розрахованих синаптичних ваг відбувається достатньо просто.

Нехай в момент часу $N = N^*$ було прийнято рішення, що в кожному орто-синапсі $O-S_i$ необхідно додати активаційну функцію $T_{h+1,i}$ так, що в мережу повинно бути введено n додаткових синаптичних ваг $w_{h+1,i}$.

При цьому оцінка $w(N)$ (1.31) повинна бути відкоригована наступним чином:

$$w^*(N) = \left(\begin{array}{c|c} \sum_{k=N-s+1}^N T(\boldsymbol{\epsilon}_k)T^T(\boldsymbol{\epsilon}_k) & R_{l,h+1}(\boldsymbol{\epsilon}_k) \\ \hline R_{l,h+1}^T(\boldsymbol{\epsilon}_k) & R_{h+1,h+1}(\boldsymbol{\epsilon}_k) \end{array} \right)^{-1} \left(\begin{array}{c} \sum_{k=N-s+1}^N T(\boldsymbol{\epsilon}_k)y_k^* \\ \hline r_{h+1}(\boldsymbol{\epsilon}_k) \end{array} \right), \quad (1.32)$$

де $((h+1)n \times n)$ -матриця $R_{l,h+1}(\boldsymbol{\epsilon}_k)$ утворена елементами

$$\sum_{k=N-s+1}^N T_{li}(\boldsymbol{\epsilon}_{ki})T_{h+1,j}(\boldsymbol{\epsilon}_{kj}), \quad l=0,1,\dots,h; \quad i=1,2,\dots,n; \quad j=1,2,\dots,n;$$

$(n \times n)$ -матриця $R_{h+1,h+1}(\boldsymbol{\epsilon}_k)$ утворена елементами

$$\sum_{k=N-s+1}^N T_{h+1,i}(\boldsymbol{\epsilon}_{ki})T_{h+1,j}(\boldsymbol{\epsilon}_{kj});$$

$(n \times 1)$ -стовпець $r_{h+1,i}(\boldsymbol{\epsilon}_{ki})$ утворений елементами $\sum_{k=N-s+1}^N T_{h+1,i}(\boldsymbol{\epsilon}_{ki})y_k^*$.

Застосовуючи формулу Фробеніуса, оцінку (1.32) можна переписати у вигляді

$$\begin{aligned}
w^*(N) &= \left(\begin{array}{c|c} P_s(N) + P_s(N)R_{l,h+1}(\mathbf{x}_k)H^{-1}(\mathbf{x}_k)R_{l,h+1}^T(\mathbf{x}_k)P_s(N) & -P_s(N)R_{l,h+1}(\mathbf{x}_k)H^{-1}(\mathbf{x}_k) \\ \hline -H^{-1}(\mathbf{x}_k)R_{l,h+1}(\mathbf{x}_k)P_s(N) & H^{-1}(\mathbf{x}_k) \end{array} \right) \bullet \\
& \bullet \left(\begin{array}{c} \sum_{k=N-s+1}^N T(\mathbf{x}_k)y_k^* \\ \hline r_{h+1}(\mathbf{x}_k) \end{array} \right) = \\
& = \left(\begin{array}{c|c} w(N) + P_s(N)R_{l,h+1}(\mathbf{x}_k)H^{-1}(\mathbf{x}_k)R_{l,h+1}^T(\mathbf{x}_k)w(N) - P_s(N)R_{l,h+1}(\mathbf{x}_k)H^{-1}(\mathbf{x}_k)r_{h+1}(\mathbf{x}_k) \\ \hline -H^{-1}(\mathbf{x}_k)R_{l,h+1}^T(\mathbf{x}_k)w(N) + H^{-1}(\mathbf{x}_k)r_{h+1}(\mathbf{x}_k) \end{array} \right),
\end{aligned}$$

$$\text{де } H^{-1}(\mathbf{x}_k) = R_{h+1,h+1}^{-1}(\mathbf{x}_k) - R_{l,h+1}^T(\mathbf{x}_k)P_s(N)R_{l,h+1}(\mathbf{x}_k).$$

Зрозуміло, що архітектурна адаптація з обчислювальної точки зору складніша за синаптичну, однак можливість її реалізації в on-line режимі, завдяки використанню ортогональних активаційних функцій, дозволяє домогтися потрібних апроксимуючих властивостей в процесі навчання.

1.2 Адаптивні нейро-фаззі методи для кластеризації викривлених даних

Задача кластеризації масивів даних, що описуються наборами векторів-образів, достатньо часто зустрічається в багатьох прикладних задачах, пов'язаних з інтелектуальним аналізом даних, при цьому останнім часом особлива увага приділяється, так званій, нечіткій кластеризації [74-75], коли вектор-образ ознак, що обробляється, з різними рівнями імовірностей, можливостей або належностей може відноситися одночасно до декількох класів.

Разом з тим, в багатьох задачах Data Mining, включаючи, звичайно, кластеризацію, вихідні масиви даних можуть містити пропуски, інформація в

яких з тих або інших причин відсутня. Для роботи в подібних умовах дуже ефективним може виявитися використання математичного апарата обчислювального інтелекту [76] і, насамперед, штучних нейронних мереж [43], які вирішують задачу відновлення втрачених спостережень, і модифікацій популярного методу нечітких c -середніх [77], які вирішують задачу кластеризації без такого відновлення.

Крім відзначених, існує ще цілий ряд підходів [78, 79] до обробки даних, які містять пропуски, однак всі вони працездатні тільки у випадках, коли масив вихідних спостережень заданий заздалегідь в повному об'ємі і не змінюється в процесі обробки. В той же час існує широкий клас задач, коли дані надходять на обробку послідовно в режимі реального часу так, як це відбувається при навчанні самоорганізованих мап Кохонена [80] або їх модифікацій [41].

1.2.1 Формальна постановка задачі кластеризації

Вихідною інформацією для стандартної задачі кластеризації є вибірка даних, сформована з N n -вимірних векторів (образів) ознак $X = \{x_1, x_2, \dots, x_N\} \subset R^n, x_k \in X, k = 1, 2, \dots, N$ – або номер спостереження у таблиці «об'єкт-властивість», або поточний дискретний час в задачах on-line обробки.

Результатом кластеризації є розбиття X на m класів, що перетинаються, з деяким рівнем належності $U_q(k)$ k -го вектора ознак x_k до q -го кластера.

Вихідні дані в процесі обробки центруються і стандартизуються за всіма компонентами так, щоб всі спостереження належали гіперкубу $[-1, 1]^n$.

Таким чином вихідні дані набувають вигляду $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N\} \subset R^n, \tilde{x}_k = \{\tilde{x}_{k1}, \tilde{x}_{k2}, \dots, \tilde{x}_{kn}\}^T, -1 \leq \tilde{x}_{ki} \leq 1, 1 \leq k \leq N, 1 < q < m,$

$1 \leq i \leq n, 1 \leq k \leq N$, при цьому N в загальному випадку не обмежене і може збільшуватися з часом. Відмітимо, що традиційно прийняте в мапах Кохонена (SOM) перетворення даних до вигляду $\|\tilde{x}_k\| = 1$ в даному випадку не має сенсу, оскільки якщо x_k містить пропуск, розрахунок норми такого вектора є неможливим, а якщо x_k містить викид в одній з компонент, то \tilde{x}_k буде практично співпадати з відповідним ортом простору ознак. Перетворення $-1 \leq \tilde{x}_{ki} \leq 1$ веде до того, що невикривлені дані концентруються в околі нуля, а дані з викидами – біля -1 та +1. Крім того, в розгляд вводяться додаткові підмасиви даних [81]:

$$X_F = \{\tilde{x}_k \in \tilde{X} \mid \tilde{x}_k \text{ - вектор, що містить усі компоненти}\};$$

$$X_P = \{\tilde{x}_{ki}, 1 \leq i \leq n, 1 \leq k \leq N \mid \text{усі значення } \tilde{x}_k, \text{ що містяться у } \tilde{X}\};$$

$$X_G = \{\tilde{x}_{ki} = ?, 1 \leq i \leq n, 1 \leq k \leq N \mid \text{усі значення } \tilde{x}_k, \text{ що відсутні в } \tilde{X} \text{ (пропуски)}\}.$$

1.2.2 Функції відстані

Вибір відстані між об'єктами є вузловим моментом дослідження, від нього багато в чому залежить остаточний варіант розбиття об'єктів на класи при даному алгоритмі розбиття.

Евклідова відстань. Найбільш простий шлях розрахунку відстані між об'єктами у багатовимірному просторі полягає в розрахунку евклідових відстаней. Якщо є дво- або тривимірний простір, то ця міра є реальною геометричною відстанню між об'єктами у просторі.

$$D^2(\tilde{x}_k, w_q) = \|\tilde{x}_k - w_q\|_2^2 \quad (3.1)$$

Зауважимо, що евклідова відстань (та його квадрат) розраховується за вихідними, а не за стандартизованими даними. Це звичайний спосіб його

розрахунку, який має певні переваги. Тим не менш, на відстані можуть сильно впливати відмінності між осями, за координатами яких розраховуються ці відстані.

Відстань міських кварталів (манхеттенська відстань). Ця відстань є просто сумою модулів різниць за координатами. У більшості випадків ця міра відстані приводить до тих самих результатів, як і для звичайної відстані Евкліда. Однак відмітимо, що для цієї міри вплив окремих великих різниць (викидів) зменшується (оскільки вони не підносяться до квадрату). Манхеттенська відстань розраховується за формулою:

$$D(\tilde{x}_k, w_q) = \|\tilde{x}_k - w_q\|_1 = \sum_{i=1}^n |\tilde{x}_{ki} - w_{qi}|. \quad (3.2)$$

Відстань Чебишева. Ця відстань може виявитися корисною при визначенні двох об'єктів як «різні», якщо вони відрізняються за якоюсь однією координатою (якимсь одним виміром). Відстань Чебишева розраховується за формулою:

$$D(\tilde{x}_k, w_q) = \max |\tilde{x}_{ki} - w_{qi}| \quad (3.3)$$

Часткова відстань. Дана міра відстані розраховується за формулою (3.4)

$$D_P^2(\tilde{x}_k, w_q) = \frac{n}{\delta_{k\Sigma}} \sum_{i=1}^n (\tilde{x}_{ki} - w_{qi})^2 \delta_{ki}, \quad (3.4)$$

де w_{qi} – i -а компонента q -го прототипу (центроїда) відповідного кластера ($q = 1, 2, \dots, m$),

$$\delta_{ki} = \begin{cases} 0 & | \tilde{x}_{ki} \in X_G, \\ 1 & | \tilde{x}_{ki} \in X_F, \end{cases}$$

$$\delta_{k\Sigma} = \sum_{i=1}^n \delta_{ki}.$$

Часткова відстань (PD) дозволяє врахувати пропущені значення, що дозволяє кластеризувати пошкоджені дані. Жоден з наведених вище методів розрахунку відстаней не надає такої можливості.

1.2.3 Методи кластеризації даних

При великій кількості спостережень ієрархічні методи кластерного аналізу не придатні. У таких випадках використовують неієрархічні методи, що ґрунтуються на розділенні, які являють собою ітеративні методи розбиття вихідної сукупності. У процесі ділення нові кластери формуються доти, доки не буде виконано правило зупинки.

Така неієрархічна кластеризація полягає в розділенні набору даних на певну кількість окремих кластерів. Існують два підходи. Перший полягає у визначенні границі кластерів як найбільш щільних ділянок у багатовимірному просторі вихідних даних, тобто визначення кластера там, де є велике «скупчення точок». Другий підхід полягає у мінімізації міри відмінності об'єктів.

Адаптивний метод кластеризації. Найбільш поширеним серед неієрархічних методів є алгоритм k -середніх, який також називають швидким кластерним аналізом. Повний опис алгоритму можна знайти у роботі Хартигана і Вонга. На відміну від ієрархічних методів, які не потребують попередніх припущень відносно кількості кластерів, для можливості використання цього методу необхідно мати гіпотезу про найбільш імовірну кількість кластерів.

Алгоритм k -середніх будує k кластерів, розташованих на максимально можливо великих відстанях один від іншого. Основний тип

задач, які вирішує алгоритм k -середніх, – наявність припущень (гіпотез) відносно кількості кластерів, при цьому вони повинні бути різними настільки, наскільки це можливо. Вибір числа k може базуватися на результатах попередніх досліджень, теоретичних міркуваннях або інтуїції.

Загальна ідея алгоритму: задана фіксована кількість k кластерів, спостереження ставлять у відповідність до кластерів так, що середні у кластері (для всіх змінних) максимально можливо відрізняються один від іншого. На рисунку 1.13 наведений приклад роботи алгоритму k -середніх.

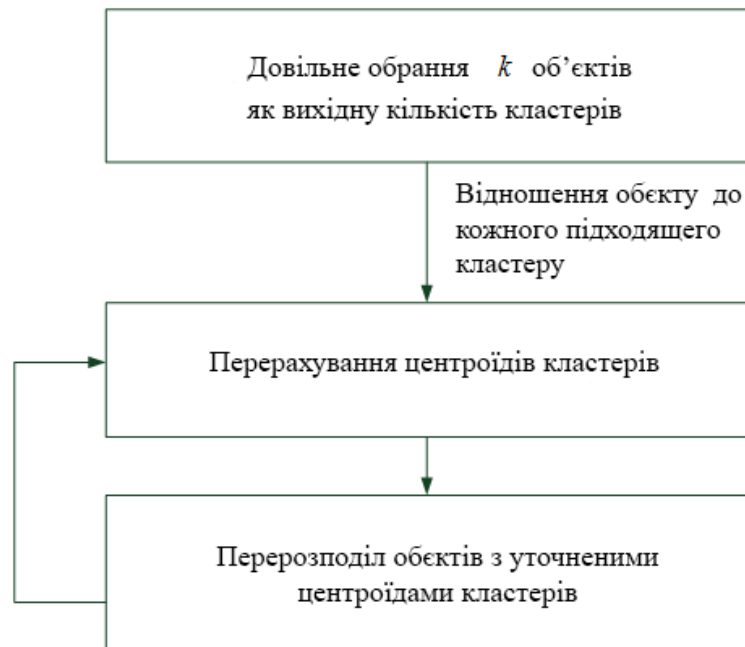


Рисунок 1.13 – Приклад роботи алгоритму k -середніх

Для вирішення задачі кластеризації в on-line режимі доцільно скористатися самоорганізовною мапою Т. Кохонена [80], яка має просту архітектуру з прямою передачею інформації і яка крім нульового рецепторного шару містить єдиний шар нейронів, найчастіше тих самих адаптивних лінійних асоціаторів (ALA). Кожний нейрон зв'язаний з кожним рецептором нульового шару прямими зв'язками і з всіма іншими нейронами поперечними внутрешаровими (латеральними) зв'язками. Саме латеральні зв'язки забезпечують збудження одних нейронів і гальмування інших.

Завдяки такій організації мережі, кожний нейрон-ALA отримує всю інформацію про вектор-образ, що аналізується, і генерує на своєму виході відповідний відгук, після чого між нейронами виникає конкуренція, в результаті якої визначається єдиний нейрон-переможець з максимальним вихідним сигналом. Цей сигнал латеральними зв'язками забезпечує збудження найближчих «сусідів» переможця і заглушення реакції віддалених нейронів. Таким чином в процесі конкурентного самонавчання формуються групи нейронів, кожен з яких максимальним чином реагує на образи «своїх» кластерів-підобластей вхідного простору сигналів.

Розглянемо елементарну мапу Кохонена, яка має 1D топологію, n рецепторів і m (за кількістю можливих кластерів) нейронів, кожен з яких характеризується власним вектором синаптичних ваг $w_q^K, q = 1, 2, \dots, m$, які визначають центроїд відповідного кластера.

Кожний нейрон мережі в k -й момент часу отримує на вході n -вимірний вхідний вектор

$$\tilde{x}_k = \frac{\mathbf{x}_k^T}{\|\mathbf{x}_k\|} \quad (1.37)$$

і генерує на своєму виході сигнал

$$y_{kq} = w_q^{KT} (k-1) \tilde{x}_k, \quad (1.38)$$

який залежить від вектора синаптичних ваг $w_q^K (k-1)$, налаштованих за даними, отриманими до моменту k , на певну область (кластер) вхідного простору. Близькі у сенсі метрики, що використовується, вхідні вектори \tilde{x}_k і \tilde{x}_i можуть збуджувати або один і той самий нейрон w_q^K , або два нейронисусіди w_q^K і w_{q+1}^K або w_q^K і w_{q-1}^K , $q = 1, 2, \dots, m$.

В основі алгоритму самоорганізації мапи Кохонена лежать принципи конкурентного самонавчання [51], при цьому як і будь-яка інша процедура налаштування нейронної мережі, робота алгоритму починається з ініціалізації синаптичних ваг мережі, які зазвичай обираються за допомогою генератора випадкових чисел, при цьому для кожного з нейронів повинна виконуватися умова

$$\|w_q^K(0)\| = 1, l = 1, 2, \dots, m. \quad (1.39)$$

Процедура самоорганізації реалізується в три основні етапи: конкуренції, кооперації і синаптичної адаптації і починається з аналізу образу \tilde{x}_k , який надходить з рецепторного шару на всі нейрони мережі. Для кожного з нейронів розраховується відстань

$$D(\tilde{x}_k, w_q^K(k-1)) = \|\tilde{x}_k - w_q^K(k-1)\|, \quad (1.40)$$

при цьому, якщо вхідні вектори попередньо пронормовані за допомогою виразу (1.37) так, що

$$\|\tilde{x}_k\| = 1, \quad (1.41)$$

а в якості відстані (1.40) використовується евклідова метрика, то мірою відстані між \tilde{x}_k та $w_q^K(k-1)$ може слугувати

$$D(\tilde{x}_k, w_q^K(k-1)) = 2(1 - \tilde{x}_k^T w_q^K(k-1)) = 2(1 - \cos(\tilde{x}_k, w_q^K(k-1))), \quad (1.42)$$

а мірою близькості – скалярний добуток

$$-1 \leq \tilde{x}_k^T w_q^K(k-1) = \cos(\tilde{x}_k, w_q^K(k-1)) \leq 1. \quad (1.43)$$

Далі визначається нейрон-переможець, «найближчий» до вхідного образу \tilde{x}_k такий, що

$$D(\tilde{x}_k, w_q^K(k-1)) = \min_q D(\tilde{x}_k, w_q^K(k-1)), \quad (1.44)$$

після чого, тимчасово опускаючи етап кооперації, можна налаштувати синаптичні ваги мережі за допомогою модифікованого правила самонавчання

$$w_q^K(k) = \begin{cases} \frac{w_q^K(k-1) + \eta(k)(\tilde{x}_k - w_q^K(k-1))}{\|w_q^K(k-1) + \eta(k)(\tilde{x}_k - w_q^K(k-1))\|} & , \text{ якщо } q\text{-й нейрон переміг,} \\ w_q^K(k-1) & \text{в протилежному випадку.} \end{cases} \quad (1.45)$$

Нескладно побачити, що процедура (3.13) реалізує принцип «переможець отримує все», при цьому вектор синаптичних ваг $w_q^K(k-1)$ нейрона-переможця «підтягується» до вхідного вектора \tilde{x}_k на відстань, що визначається кроком налаштування $\eta(k)$.

Регулювання значення кроку зазвичай проводиться, виходячи з емпіричних міркувань, при цьому досить зручною є процедура [66]

$$\eta(k) = r^{-1}(k), r(k) = \alpha r(k-1) + \|\tilde{x}_k\|^2 = \alpha r(k-1) + 1. \quad (1.46)$$

Однією з особливостей мапи Кохонена є наявність етапу кооперації в процесі самоорганізації, коли нейрон-переможець визначає локальну область топологічного сусідства, в якій збуджується не тільки він сам, але і його найближче оточення, при цьому більш близькі до переможця нейрони збуджуються сильніше, ніж віддалені. Ця топологічна область визначається функцією сусідства ϕ_{ql} , яка залежить від відстані $D(w_q^K(k-1), w_l^K(k-1))$ між переможцем $w_q^K(k-1)$ і будь-яким з нейронів мережі $w_l^K(k-1)$. Як правило, ϕ_{ql} – це ядерна функція, яка є симетричною відносно максимуму в точці $D(w_q^K(k-1), w_q^K(k-1)) = 0$, приймає в ній одиничне значення, монотонно спадає зі збільшенням відстані і наближається до нуля при $D(w_q^K(k-1), w_l^K(k-1)) = 2$. При цьому найчастіше в якості функції сусідства використовуються гаусіан, конус, функції Єпанечнікова, «мексиканський капелюх» тощо.

Використання функції сусідства приводить до модифікованого правила навчання

$$\begin{cases} w_l^K(k) = \frac{w_l^K(k-1) + r^{-1}(k)\phi_{ql}(\tilde{x}_k - w_l^K(k-1))}{\|w_l^K(k-1) + r^{-1}(k)\phi_{ql}(\tilde{x}_k - w_l^K(k-1))\|} & \forall l, \\ r(k) = \alpha r(k-1) + 1, \end{cases} \quad (1.47)$$

який реалізує принцип «переможець отримує більше». При $\phi_{ql} = \delta_{ql} = 1$ при $q=l$ і 0 в інших випадках, знову приходимо до алгоритму (1.45), який забезпечує на кожному такті налаштування єдиного нейрона-переможця $w_q^K(k-1)$.

В принципі, для навчання самоорганізовна мапи можна взагалі обійтися без знаходження переможця, а в якості функції сусідства використовувати деяку конструкцію, яка залежить від вихідних сигналів нейронів (1.41) так, що

$$\begin{cases} w_l^K(k) = \frac{w_l^K(k-1) + r^{-1}(k)\phi(y_{kl})(\tilde{x}_k - w_l^K(k-1))}{\|w_l^K(k-1) + r^{-1}(k)\phi(y_{kl})(\tilde{x}_k - w_l^K(k-1))\|} \quad \forall l, \\ r(k) = \alpha r(k-1) + 1. \end{cases} \quad (1.48)$$

Оскільки

$$y_{kl} = w_l^{KT}(k-1)\tilde{x}_k = \cos(\tilde{x}_k, w_l^K(k-1)), \quad (1.49)$$

то найпростішою такою конструкцією є

$$\phi(y_{kl}) = \frac{1 + y_{kl}}{2}, \quad (1.50)$$

яка задовольняє всім умовам, що висуваються до ядерних функцій сусідства. Тоді остаточно алгоритм навчання мапи Кохонена для знаходження центроїдів m кластерів має вигляд

$$\begin{cases} w_l^K(k) = \frac{w_l^K(k-1) + r^{-1}(k)\frac{1 + y_{kl}}{2}(\tilde{x}_k - w_l^K(k-1))}{\|w_l^K(k-1) + r^{-1}(k)\frac{1 + y_{kl}}{2}(\tilde{x}_k - w_l^K(k-1))\|} \quad \forall l, \\ r(k) = \alpha r(k-1) + 1. \end{cases} \quad (1.51)$$

Нейромережева архітектура, яка реалізує процес кластеризації даних з пропусками, наведена на рисунку 1.14.

Як видно, введена нейронна мережа утворена однотипними нейронами – адаптивними лінійними асоціаторами, які налаштовуються за допомогою достатньо простих і наочних алгоритмів навчання.

Метод адаптивної нечіткої кластеризації даних з пропусками.
Власне нечітка кластеризація реалізується шляхом мінімізації цільової функції [43, 82]

$$E(U_q(k), w_q) = \sum_{k=1}^N \sum_{q=1}^m U_q^\beta(k) D^2(\tilde{x}_k, w_q) \quad (1.52)$$

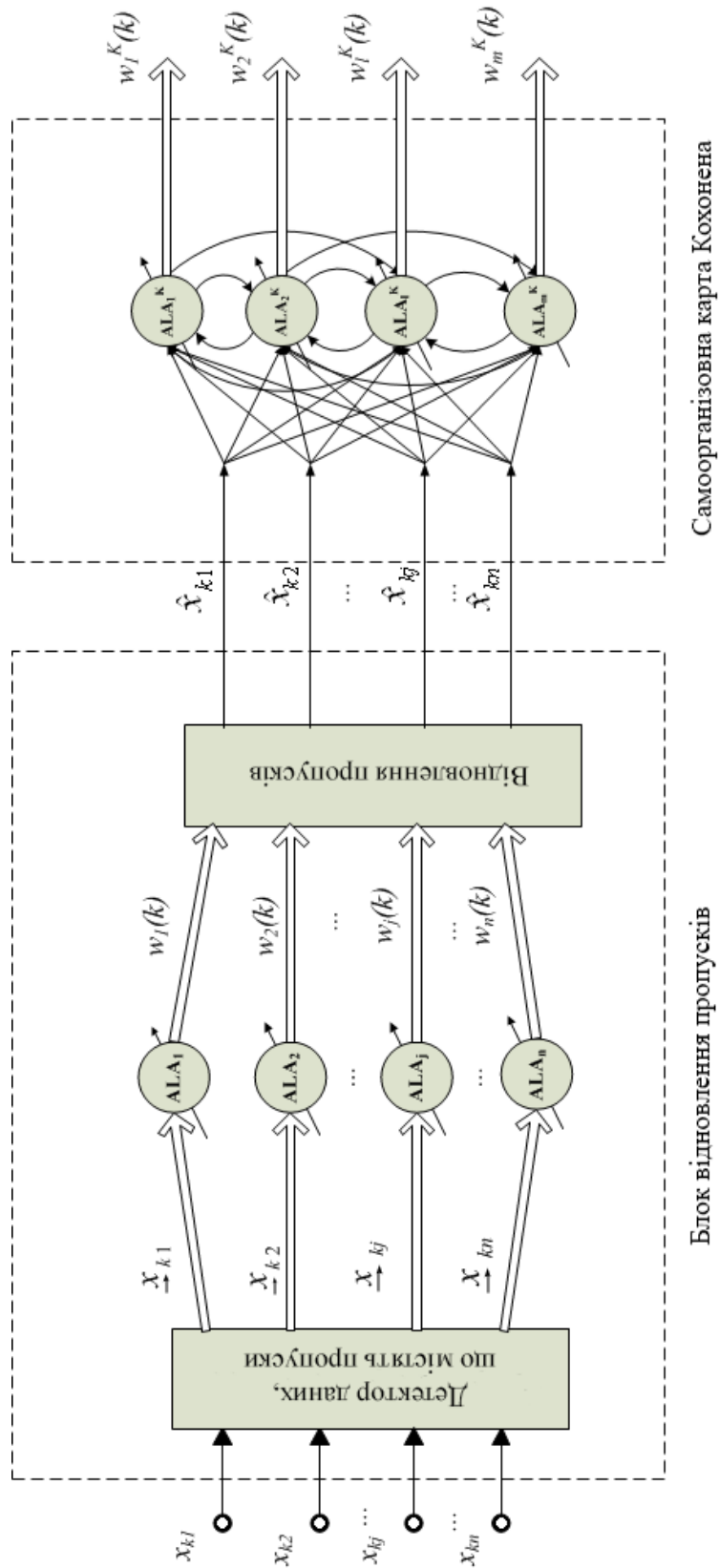


Рисунок 1.14 – Штучна мережа для адаптивної кластеризації даних з пропущеними значеннями

при обмеженнях

$$\sum_{q=1}^m U_q(k) = 1, \quad k = 1, \dots, N, \quad (1.53)$$

$$0 \leq \sum_{k=1}^N U_q(k) \leq N, \quad q = 1, \dots, m. \quad (1.54)$$

Тут $U_q(k) \in [0, 1]$ – рівень належності вектора \tilde{x}_k до q -го кластера,

w_q – прототип (центроїд) q -го кластера

β – невід’ємний параметр, що називається «фаззіфікатором» (зазвичай $\beta = 2$),

$D^2(\tilde{x}_k, w_q)$ – відстань між \tilde{x}_k і w_q в прийнятій метриці.

Результатом кластеризації є $(N \times m)$ -матриця $W = \{U_q(k)\}$, яка називається «матрицею нечіткого розбиття».

Розв’язок задачі нелінійного програмування за допомогою невизначених множників Лагранжа веде до відомого алгоритму нечіткої імовірнісної кластеризації [67]

$$\left\{ \begin{array}{l} U_q(k) = \frac{(D^2(\tilde{x}_k, w_q))^{-\frac{1}{\beta}}}{\sum_{l=1}^m (D^2(\tilde{x}_k, w_l))^{-\frac{1}{\beta}}}, \\ w_q = \frac{\sum_{k=1}^N U_q^\beta(k) \tilde{x}_k}{\sum_{k=1}^N U_q^\beta(k)}, \end{array} \right. \quad (1.55)$$

який приймає при $\beta = 2$, $D^2(\tilde{x}_k, w_q) = \|\tilde{x}_k - w_q\|^2$ форму популярної процедури, що була введена Дж. Бездеком:

$$\left\{ \begin{array}{l} U_q(k) = \frac{\|\tilde{x}_k - w_q\|^{-2}}{\sum_{l=1}^m \|\tilde{x}_k - w_l\|^{-2}}, \\ w_q = \frac{\sum_{k=1}^N U_q^2(k) \tilde{x}_i}{\sum_{k=1}^N U_q^2(k)}. \end{array} \right. \quad (1.56)$$

Для роботи в on-line режимі, коли дані на обробку надходять послідовно, замість пакетної форми (1.55) пропонується використовувати рекурентний алгоритм [83]

$$\left\{ \begin{array}{l} w_q(k) = w_q(k+1) - \eta(k) U_q^\beta(k+1) D(\tilde{x}_k, w_q(k+1)) \nabla_{w_q} D(\tilde{x}_k, w_q(k+1)), \\ U_q(k) = \frac{(D^2(\tilde{x}_k, w_q(k)))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (D^2(\tilde{x}_k, w_l(k)))^{\frac{1}{1-\beta}}}, \end{array} \right. \quad (1.57)$$

або адаптивну модифікацію FCM алгоритму Бездека [56]

$$\left\{ \begin{array}{l} w_q(k) = w_q(k+1) + \eta(k) U_q^2(k+1) (\tilde{x}_k - w_q(k+1)), \\ U_q(k) = \frac{\|\tilde{x}_k - w_q(k)\|^{-2}}{\sum_{l=1}^m \|\tilde{x}_k - w_l(k)\|^{-2}}. \end{array} \right. \quad (1.58)$$

Тут $\eta(k)$ – параметр кроку навчання.

Варто відзначити, що хоча алгоритми (1.57), (1.58) досить близькі до процедур Park-Dagher і Chung-Lee [84], однак отримані вони з інших міркувань.

Адаптивний метод кластеризації даних з пропусками на основі нейро-фаззі підходу. Вводячи цільову функцію кластеризації

$$E(U_q(k), w_q) = \sum_{k=1}^N \sum_{q=1}^m U_q^\beta(k) \frac{n}{\delta_{k\Sigma}} \sum_{i=1}^n (\tilde{x}_{ki} - w_{qi})^2 \delta_{ki}$$

і розв'язуючи задачу нелінійного програмування, приходимо до алгоритму

$$\left\{ \begin{array}{l} U_q^{(\tau+1)} = \frac{(D_P^2(\tilde{x}_k, w_q^{(\tau)}))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (D_P^2(\tilde{x}_k, w_l^{(\tau)}))^{\frac{1}{1-\beta}}}, \\ w_{qi}^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta \delta_{ki} \tilde{x}_{ki}}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta \delta_{ki}}, \end{array} \right.$$

де $\tau=0,1,2,\dots$ - індекс епохи обробки інформації, яка організується у вигляді послідовності $w_q^{(0)} \rightarrow U_q^{(1)} \rightarrow w_q^{(1)} \rightarrow U_q^{(2)} \rightarrow \dots$

Процес розрахунків продовжується до виконання умови

$$\|w_q^{(\tau+1)} - w_q^{(\tau)}\| \leq \varepsilon \quad \forall 1 \leq q \leq m,$$

де ε – наперед заданий поріг точності розрахунків.

Даний алгоритм може бути переписаний у рекурентній формі:

$$\begin{cases} U_q(k+1) = \frac{(D_P^2(\tilde{x}_{k+1}, w_q(k)))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (D_P^2(\tilde{x}_{k+1}, w_q(k)))^{\frac{1}{1-\beta}}}, \\ w_{qi}(k+1) = w_{qi}(k) + \eta(k+1)U_q^\beta(k+1)(\tilde{x}_{k+1,i} - w_{qi}(k))\delta_{ki}, \end{cases}$$

при цьому друге співвідношення може бути представлене у вигляді алгоритму самонавчання нейро-фаззи мережі Кохонена:

$$w_q(k+1) = w_q(k) + \eta(k+1)\phi_q(k+1)(\tilde{x}_{k+1} - w_q(k)) \square \delta_k,$$

де $\phi_q(k+1) = U_q^\beta(k+1)$ – дзвонувата функція сусідства, $\delta_k = (\delta_{k1}, \dots, \delta_{kn})^T$,

\square – символ прямого (скоттова) добутку.

Таким чином, використовуючи архітектуру стандартної мережі Кохонена і введений алгоритм її налаштування, можна в on-line режимі розв'язати задачу нечіткої кластеризації даних з пропусками.

Адаптивний метод імовірнісної нечіткої кластеризації даних.
Вводячи цільову функцію кластеризації [56]

$$E(U_q(k), w_q) = \sum_{k=1}^N \sum_{q=1}^m U_q^\beta(k) D^2(\tilde{x}_k, w_q)$$

при обмеженнях $\sum_{q=1}^m U_q(k) = 1$, $0 < \sum_{k=1}^N U_q(k) < N$ і розв'язуючи задачу

нелінійного програмування, приходимо до алгоритму імовірнісної нечіткої кластеризації [76, 82]

$$\left\{ \begin{array}{l} U_q^{(\tau+1)}(k) = \frac{(D^2(\tilde{x}_k, w_q^{(\tau)}))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (D^2(\tilde{x}_k, w_l^{(\tau)}))^{\frac{1}{1-\beta}}}, \\ w_q^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau+1)})^\beta \tilde{x}_k}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta}, \end{array} \right. \quad (1.59)$$

де w_q - прототип (центроїд) q -го кластера,

$\beta > 1$ - параметр, що називається фаззифікатором і що визначає «розмитість» границь між класами,

$D^2(\tilde{x}_k, w_q)$ - відстань між \tilde{x}_k і w_q в прийнятій метриці,

$\tau = 0, 1, 2, \dots$ - індекс епохи обробки інформації, яка організується у вигляді послідовності $w_q^{(0)} \rightarrow U_q^{(1)} \rightarrow w_q^{(1)} \rightarrow U_q^{(2)} \rightarrow \dots$

Процес розрахунків продовжується до виконання умови

$$\|w_q^{(\tau+1)} - w_q^{(\tau)}\| \leq \varepsilon \quad \forall 1 \leq q \leq m,$$

де ε – наперед заданий поріг точності розрахунків.

Обираючи $\beta = 2$ і приймаючи евклідову відстань

$D^2(\tilde{x}_k, w_q) = \|\tilde{x}_k - w_q\|^2$, отримуємо популярний алгоритм нечітких s -середніх

(FCM) Дж. Бездека:

$$\left\{ \begin{array}{l} U_q^{(\tau+1)}(k) = \frac{\|\tilde{x}_k - w_q^{(\tau)}\|^{-2}}{\sum_{l=1}^m \|\tilde{x}_k - w_l^{(\tau)}\|^{-2}}, \\ w_q^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^2 \tilde{x}_k}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^2}. \end{array} \right.$$

Процес нечіткої кластеризації може бути організований в on-line режимі по мірі послідовного надходження даних на обробку. При цьому пакетний алгоритм (1597) може бути переписаний в рекурентній формі

$$\left\{ \begin{array}{l} U_q(k+1) = \frac{(D^2(\tilde{x}_{k+1}, w_q^{(k)}))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (D^2(\tilde{x}_{k+1}, w_l^{(k)}))^{\frac{1}{1-\beta}}}, \\ w_q(k+1) = w_q(k) + \eta(k+1) U_q^\beta(k+1) (\tilde{x}_{k+1} - w_q(k)), \end{array} \right. \quad (1.60)$$

(тут $\eta(k+1)$ - параметр кроку навчання), що є узагальненням градієнтної процедури кластеризації Парка-Деггера [85] і алгоритму навчання Чанга-Лі.

Якщо дані надходять на обробку з високою частотою, перерахування за епохами не виконується, якщо ж ця частота невелика, то між моментами часу k та $k+1$ можна організувати декілька епох у прискореному часі, при цьому моменту k відповідає $\tau=0$, а в $(k+1)$ -й момент це перерахування закінчується.

Цікаво також зауважити, що перший вираз (1.60) може бути переписаний у формі

$$\begin{aligned}
U_q(k+1) &= \frac{(D^2(\tilde{x}_k, w_q(k)))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (D^2(\tilde{x}_k, w_l(k)))^{\frac{1}{1-\beta}}} = \\
&= \frac{(D^2(\tilde{x}_k, w_q(k)))^{\frac{1}{1-\beta}}}{(D^2(\tilde{x}_k, w_q(k)))^{\frac{1}{1-\beta}} + \sum_{\substack{l=1 \\ l \neq q}}^m (D^2(\tilde{x}_k, w_l(k)))^{\frac{1}{1-\beta}}} = \\
&= \frac{1}{1 + (D^2(\tilde{x}_k, w_q(k)))^{\frac{1}{\beta-1}} \sum_{\substack{l=1 \\ l \neq q}}^m (D^2(\tilde{x}_k, w_l(k)))^{\frac{1}{1-\beta}}},
\end{aligned}$$

для евклідової метрики і при $\beta = 2$ він приймає вигляд функції Коші з параметром ширини σ^2 :

$$U_q(k+1) = \frac{1}{1 + \frac{\|\tilde{x}_k - w_q(k)\|^2}{\sigma^2}},$$

$$\sigma^2 = \left(\sum_{\substack{l=1 \\ l \neq q}}^m \|\tilde{x}_k - w_l(k)\|^{-2} \right)^{-1}.$$

Ця обставина дозволяє переписати другий вираз (1.60) при $\beta = 2$ у формі

$$\begin{aligned}
w_q(k+1) &= w_q(k) + \eta(k+1)U_q^2(k+1)(\tilde{x}_{k+1} - w_q(k)) = \\
&= w_q(k) + \eta(k+1)\varphi_q(k+1)(\tilde{x}_{k+1} - w_q(k)),
\end{aligned}$$

де $U_q^2(k+1) = \varphi_q(k+1)$ - дзвонувата функція сусідства нейро-фаззи мережі Кохонена [86], яка призначена для розв'язку задач нечіткої кластеризації [87] на основі принципу «переможець отримує більше» (WTM).

У тому випадку, якщо дані в масиві \tilde{X} містять пропуски, розглянутий вище підхід повинен бути відповідним чином модифікований. Так, в [80] запропонована модифікація FCM-процедури, що заснована на стратегії часткових відстаней (PDSFCM). При цьому в розгляд вводяться додаткові масиви. Приймаючи замість традиційної евклідової метрики часткову відстань (PD)

$$D_P^2(\tilde{x}_k, w_q) = \frac{n}{\delta_{k\Sigma}} \sum_{i=1}^n (\tilde{x}_{ki} - w_{qi})^2 \delta_{ki},$$

цільову функцію кластеризації

$$E(U_q(k), w_q) = \sum_{k=1}^N \sum_{q=1}^m U_q^\beta(k) \frac{n}{\delta_{k\Sigma}} \sum_{i=1}^n (\tilde{x}_{ki} - w_{qi})^2 \delta_{ki}$$

і розв'язуючи все ту саму задачу нелінійного програмування, приходимо до алгоритму

$$\left\{ \begin{array}{l} U_q^{(\tau+1)} = \frac{(D_P^2(\tilde{x}_k, w_q^{(\tau)}))^{1-\beta}}{\sum_{l=1}^m (D_P^2(\tilde{x}_k, w_q^{(\tau)}))^{1-\beta}}, \\ w_{qi}^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta \delta_{ki} \tilde{x}_{ki}}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta \delta_{ki}}, \end{array} \right. \quad (1.61)$$

що є узагальненням стандартної FCM-процедури (1.59).

Алгоритм (1.61) може бути переписаний у рекурентній формі по типу (1.60):

$$\begin{cases} U_q(k+1) = \frac{(D_P^2(\tilde{x}_{k+1}, w_q(k)))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (D_P^2(\tilde{x}_{k+1}, w_q(k)))^{\frac{1}{1-\beta}}}, \\ w_{qi}(k+1) = w_{qi}(k) + \eta(k+1)U_q^\beta(k+1)(\tilde{x}_{k+1,i} - w_{qi}(k))\delta_{ki}, \end{cases} \quad (1.62)$$

при цьому друге співвідношення (1.62) може бути представлене у вигляді алгоритму самонавчання нейро-фаззи мережі Кохонена:

$$w_q(k+1) = w_q(k) + \eta(k+1)\phi_q(k+1)(\tilde{x}_{k+1} - w_q(k)) \square \delta_k, \quad (1.63)$$

де $\phi_q(k+1) = U_q^\beta(k+1)$ - дзвонувата функція сусідства, $\delta_k = (\delta_{k1}, \dots, \delta_{kn})^T$,

\square - символ прямого (скоттова) добутку.

Таким чином, використовуючи архітектуру стандартної мережі Кохонена і алгоритм її налаштування (1.64), можна в on-line режимі розв'язувати задачу нечіткої кластеризації даних з пропусками.

Адаптивний алгоритм можливісної нечіткої кластеризації даних. Основний недолік імовірнісних алгоритмів нечіткої кластеризації пов'язаний з обмеженням на суму належностей, яка повинна дорівнювати одиниці. Ця обставина привела до появи можливісних алгоритмів нечіткої кластеризації [88].

У можливісних алгоритмах цільова функція кластеризації має вигляд

$$E(U_q(k), w_q, \mu_q) = \sum_{k=1}^N \sum_{q=1}^m U_q^\beta(k) D^2(\tilde{x}_k, w_q) + \sum_{q=1}^m \mu_q \sum_{k=1}^N (1 - U_q(k))^\beta, \quad (1.64)$$

де скалярний параметр $\mu \geq 0$ визначає відстань, на якій рівень

належності приймає значення 0,5, тобто якщо $D^2(\tilde{x}_k, w_q) = \mu_q$, то $w_q(k) = 0,5$.

Мінімізація (1.64) за $U_q(k)$, w_q і μ_q дає очевидний розв'язок

$$\left\{ \begin{array}{l} U_q^{(\tau+1)}(k) = \frac{1}{1 + \left(\frac{D^2(\tilde{x}_k, w_q^{(\tau)})}{\mu_q^{(\tau)}} \right)^{\frac{1}{\beta-1}}}, \\ w_q^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^{\beta} \tilde{x}_k}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^{\beta}}, \\ \mu_q^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^{\beta} D^2(\tilde{x}_k, w_q^{(\tau+1)})}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^{\beta}}, \end{array} \right. \quad (1.65)$$

що при $\beta = 2$ і евклідовій метриці приймає вигляд

$$\left\{ \begin{array}{l} U_q^{(\tau+1)}(k) = \frac{1}{1 + \frac{\|\tilde{x}_k - w_q^{(\tau)}\|^2}{\mu_q^{(\tau)}}}, \\ w_q^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau)}(k))^2 \tilde{x}_k}{\sum_{k=1}^N (U_q^{(\tau)}(k))^2}, \\ \mu_q^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau)}(k))^2 \|\tilde{x}_k - w_q^{(\tau+1)}\|^2}{\sum_{k=1}^N (U_q^{(\tau)}(k))^2}. \end{array} \right. \quad (1.66)$$

У режимі on-line обробки інформації співвідношення (1.65), (1.66) можуть бути представлені у вигляді [89, 86]

$$\left\{ \begin{array}{l} U_q(k+1) = \frac{1}{1 + \left(\frac{D^2(\tilde{x}_{k+1}, w_q(k))}{\mu_q(k)} \right)^{\frac{1}{\beta-1}}}, \\ w_q(k+1) = w_q(k) + \eta(k+1)U_q^\beta(k+1)(\tilde{x}_{k+1} - w_q(k)), \\ \mu_q(k+1) = \frac{\sum_{p=1}^{k+1} U_q^\beta(p)D^2(\tilde{x}_p, w_q(k+1))}{\sum_{p=1}^{k+1} U_q^\beta(p)} \end{array} \right. \quad (1.67)$$

і

$$\left\{ \begin{array}{l} U_q(k+1) = \frac{1}{1 + \frac{\|\tilde{x}_k - w_q(k)\|^2}{\mu_q(k)}}, \\ w_q(k+1) = w_q(k) + \eta(k+1)U_q^2(k+1)(\tilde{x}_{k+1} - w_q(k)), \\ \mu_q(k+1) = \frac{\sum_{p=1}^{k+1} U_q^2(p)\|\tilde{x}_p - w_q(k+1)\|^2}{\sum_{p=1}^k U_q^2(p)}. \end{array} \right. \quad (1.68)$$

Неважко помітити, що другі співвідношення в (1.67), (1.68) є не що інше, як WTM-правила самонавчання Кохонена з кошанами в якості функції сусідства.

Приймаючи замість евклідової метрики часткову відстань, запишемо цільову функцію кластеризації типу (1.64) у вигляді

$$E(U_q(k), w_q, \mu_q) = \sum_{k=1}^N \sum_{q=1}^m U_q^\beta(k) \frac{n}{\delta_{k\Sigma}} \sum_{i=1}^n (\tilde{x}_{ki} - w_{qi})^2 \delta_{ki} + \sum_{q=1}^m \mu_q \sum_{k=1}^N (1 - U_q(k))^\beta,$$

після чого, розв'язуючи систему рівнянь

$$\left\{ \begin{array}{l} \frac{\partial E(U_q(k), w_q, \mu_q)}{\partial U_q(k)} = 0, \\ \frac{\partial E(U_q(k), w_q, \mu_q)}{\partial \mu_q} = 0, \\ \nabla_{w_q} E(U_q(k), w_q, \mu_q) = \vec{0}, \end{array} \right.$$

приходимо до процедури типу (1.65), яка може бути переписана у рекурентній формі

$$\left\{ \begin{array}{l} U_q(k) = \frac{1}{1 + \left(\frac{D_p^2(\tilde{x}_{k+1}, w_q(k))}{\mu_q(k)} \right)^{\frac{1}{\beta-1}}}, \\ w_{qi}(k+1) = w_{qi}(k) + \eta(k+1) U_q^\beta(k+1) (\tilde{x}_{k+1,i} - w_{qi}(k)) \delta_{ki}, \\ \mu_q(k+1) = \frac{\sum_{p=1}^{k+1} U_q^\beta(p) D_p^2(\tilde{x}_p, w_q(k+1))}{\sum_{p=1}^{k+1} U_q^\beta(p)}, \end{array} \right.$$

друге співвідношення якої можна переписати у вигляді

$$w_q(k+1) = w_q(k) + \eta(k+1) U_q^\beta(k+1) (\tilde{x}_{k+1} - w_q(k)) \square \delta_k,$$

що співпадає з процедурою самонавчання (1.63).

Таким чином, процес нечіткої можливісної кластеризації даних з пропусками також може бути реалізований за допомогою нейро-фаззі мережі Кохонена.

Адаптивна нечітка робастна кластеризація даних на основі міри подібності. Як вже відмічалось, для рішення задачі нечіткої кластеризації

даних, які містять викиди, можуть бути використані цільові функції спеціального виду [89-92], які тим або іншим способом придумують ці аномалії, а сама задача пов'язана з мінімізацією цих функцій. З практичної точки зору більш зручним представляється використання замість цільових функцій, що засновані на метриках, так званих, мір подібності (SM) [93], до яких висуваються м'якші, ніж до метрик, умови:

$$\begin{cases} S(\tilde{x}_k, \tilde{x}_p) \geq 0, \\ S(\tilde{x}_k, \tilde{x}_p) = S(\tilde{x}_p, \tilde{x}_k) \\ S(\tilde{x}_k, \tilde{x}_k) = 1 \geq S(\tilde{x}_k, \tilde{x}_p) \end{cases}$$

(відсутня нерівність трикутника), а задача кластеризації може бути «прив'язана» до максимізації цих мір.

Якщо дані перетворені так, що $-1 \leq \tilde{x}_{ki} \leq 1$, то міра подібності може бути сконструйована так, щоб заглушити небажані дані, що лежать на кінцях інтервалу $[-1, 1]$.

Рисунок 1.15 ілюструє використання в якості міри подібності функції Коші з різними параметрами ширини $\sigma^2 < 1$.

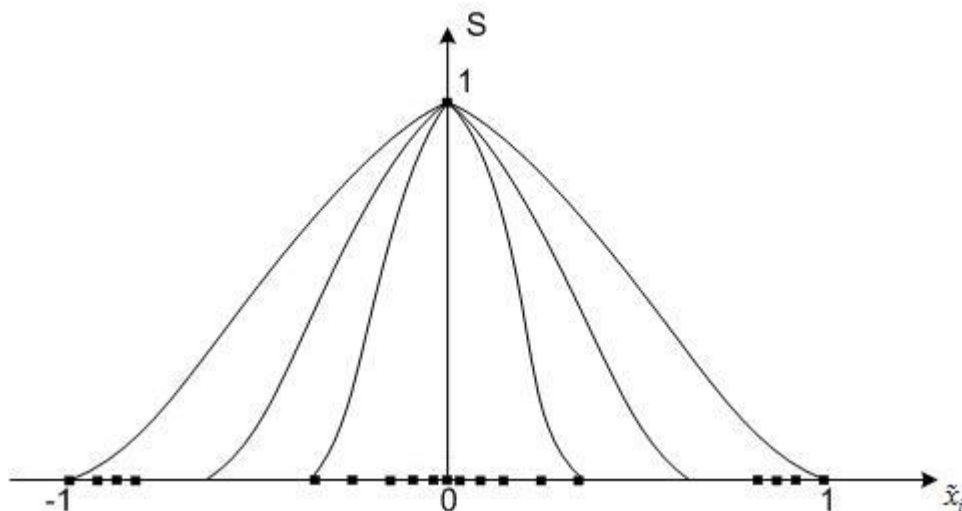


Рисунок 1.15 – Міра подібності на основі функції Коші

Підбираючи параметр ширини σ^2 функції

$$S(\tilde{x}_k, w_q) = \frac{1}{1 + \frac{\|\tilde{x}_k - w_q\|^2}{\sigma^2}} = \frac{\sigma^2}{\sigma^2 + \|\tilde{x}_k - w_q\|^2} = \frac{\sigma^2}{\sigma^2 + D^2(\tilde{x}_k, w_q)}, \quad (1.69)$$

можна виключити вплив аномальних спостережень, що в принципі неможливо зробити за допомогою евклідової метрики

$$D^2(\tilde{x}_k, w_q) = \|\tilde{x}_k - w_q\|^2, \quad (1.70)$$

при цьому можна помітити, що $S(\tilde{x}_k, w_q)$ є величина обернена до $D^2(\tilde{x}_k, w_q)$.

Далі, вводячи в розгляд цільову функцію, що заснована на мірі подібності (1.69),

$$E_S(U_q(k), w_q) = \sum_{k=1}^N \sum_{q=1}^m U_q^\beta(k) S(\tilde{x}_k, w_q) = \sum_{k=1}^N \sum_{q=1}^m \frac{U_q^\beta(k) \sigma^2}{\sigma^2 + \|\tilde{x}_k - w_q\|^2},$$

імовірнісні обмеження

$$\sum_{q=1}^m U_q(k) = 1,$$

функцію Лагранжа

$$L_S(U_q(k), w_q, \lambda(k)) = \sum_{k=1}^N \sum_{q=1}^m \frac{U_q^\beta(k) \sigma^2}{\sigma^2 + \|\tilde{x}_k - w_q\|^2} + \sum_{k=1}^N \lambda(k) \left(\sum_{q=1}^m U_q(k) - 1 \right) \quad (1.71)$$

(тут $\lambda(k)$ - невизначені множники Лагранжа) і розв'язуючи систему рівнянь Каруша-Куна-Таккера, приходимо до розв'язку

$$\begin{cases} U_q(k) = (S(\tilde{x}_k, w_q))^{\frac{1}{\beta-1}} / \sum_{l=1}^m (S(\tilde{x}_k, w_l))^{\frac{1}{\beta-1}}, \\ \lambda(k) = -\left(\sum_{l=1}^m (\beta S(\tilde{x}_k, w_l))^{\frac{1}{\beta-1}}\right)^{\beta-1}, \\ \nabla_{w_q} L_S(U_q(k), w_q, \lambda(k)) = \sum_{k=1}^N U_q^\beta(k) \frac{\tilde{x}_k - w_q}{(\sigma^2 + \|\tilde{x}_k - w_q\|^2)^2} = \vec{0}. \end{cases} \quad (1.72)$$

Останнє рівняння системи (1.72) не має аналітичного розв'язку, тому для знаходження сідлової точки лагранжіана (1.71) можна скористатися процедурою Ерроу-Гурвіца-Удзави, в результаті застосування якої приходимо до алгоритму

$$\begin{cases} U_q(k+1) = \frac{(S(\tilde{x}_{k+1}, w_q(k)))^{\frac{1}{\beta-1}}}{\sum_{l=1}^m (S(\tilde{x}_{k+1}, w_l(k)))^{\frac{1}{\beta-1}}}, \\ w_q(k+1) = w_q(k) + \eta(k+1) U_q^\beta(k+1) \frac{\tilde{x}_{k+1} - w_q(k)}{(\sigma^2 + \|\tilde{x}_{k+1} - w_q(k)\|^2)^2} = \\ = w_q(k) + \eta(k+1) \varphi_q(k+1) (\tilde{x}_{k+1} - w_q(k)), \end{cases} \quad (1.73)$$

де $\varphi_q(k+1) = \frac{\tilde{x}_{k+1} - w_q(k)}{(\sigma^2 + \|\tilde{x}_{k+1} - w_q(k)\|^2)^2}$ - функція сусідства робастного

WTM-правила самонавчання.

Задаючи значення фаззифікатора $\beta = 2$, приходимо до робастного варіанта FCM:

$$\begin{cases} U_q(k+1) = \frac{S(\tilde{x}_{k+1}, w_q(k))}{\sum_{l=1}^m S(\tilde{x}_{k+1}, w_l(k))}, \\ w_q(k+1) = w_q(k) + \eta(k+1) \frac{U_q^2(k+1)}{(\sigma^2 + \|\tilde{x}_{k+1} - w_q(k)\|^2)^2}. \end{cases}$$

Використовуючи далі концепцію прискореного часу по типу (1.65), можна ввести робастну адаптивну імовірнісну процедуру нечіткої кластеризації виду (1.74), при цьому рішення про належність кожного \tilde{x}_k до конкретного кластера приймається за максимальним значенням міри подібності.

$$\begin{cases} U_q^{(\tau+1)}(k) = \frac{(S(\tilde{x}_k, w_q^{(\tau)}(k)))^{\frac{1}{\beta-1}}}{\sum_{l=1}^m (S(\tilde{x}_k, w_l^{(\tau)}(k)))^{\frac{1}{\beta-1}}}, \\ w_q^{(0)}(k) = w_q^{(0)}(k+1), \\ w_q^{(\tau+1)}(k+1) = w_q^{(\tau)}(k+1) + \eta(k+1) \frac{(U_q^{(0)}(k))^\beta}{(\sigma^2 + \|\tilde{x}_{k+1} - w_q^{(\tau)}(k+1)\|^2)^2} (\tilde{x}_{k+1} - w_q^{(\tau)}(k+1)), \end{cases} \quad (1.74)$$

Аналогічним чином може бути синтезований алгоритм робастної адаптивної можливісної нечіткої кластеризації.

Вводячи цільову функцію

$$E_S(U_q(k), w_q, \mu_q) = \sum_{k=1}^N \sum_{q=1}^m U_q^\beta(k) S^{-1}(\tilde{x}_k, w_q) + \sum_{q=1}^m \mu_q \sum_{k=1}^N (1 - U_q(k))^\beta$$

і розв'язуючи задачу її оптимізації, приходимо до розв'язку:

$$\left\{ \begin{array}{l} U_q(k+1) = \left(1 + \left(\frac{S^{-1}(\tilde{x}_{k+1}, w_q(k))}{\mu_q(k)} \right) \right)^{-1}, \\ w_q(k+1) = w_q(k) + \eta(k+1) U_q^\beta(k+1) \frac{\tilde{x}_{k+1} - w_q(k)}{(\sigma^2 + \|\tilde{x}_{k+1} - w_q(k)\|^2)^2}, \\ \mu_q(k+1) = \frac{\sum_{p=1}^{k+1} U_q^\beta(p) S^{-1}(\tilde{x}_p, w_q(k+1))}{\sum_{p=1}^{k+1} U_q^\beta(p)}, \end{array} \right. \quad (1.75)$$

що при $\beta = 2$ набуває вигляду

$$\left\{ \begin{array}{l} U_q(k+1) = \frac{1}{1 + \frac{S^{-1}(\tilde{x}_{k+1}, w_q(k))}{\mu_q(k)}}, \\ w_q(k+1) = w_q(k) + \eta(k+1) \frac{U_q^2(k+1)}{(\sigma^2 + \|\tilde{x}_{k+1} - w_q(k)\|^2)^2} (\tilde{x}_{k+1} - w_q(k)), \\ \mu_q(k+1) = \frac{\sum_{p=1}^{k+1} U_q^2(p) S^{-1}(\tilde{x}_p, w_q(k+1))}{\sum_{p=1}^{k+1} U_q^2(p)}. \end{array} \right.$$

I, нарешті, вводячи прискорений час, отримуємо процедуру

$$\left\{ \begin{array}{l} U_q^{(\tau+1)}(k) = \frac{1}{1 + \left(\frac{S^{-1}(\tilde{x}_k, w_q^{(\tau)}(k))}{\mu_q^{(\tau)}(k)} \right)^{\frac{1}{\beta-1}}}, \\ w_q^{(Q)}(k) = w_q^{(0)}(k+1), \\ w_q^{(\tau+1)}(k+1) = w_q^{(\tau)}(k+1) + \eta(k+1) \frac{(U_q^{(Q)}(k))^{\beta}}{(\sigma^2 + \|\tilde{x}_{k+1} - w_q^{(\tau)}(k+1)\|^2)^2} (\tilde{x}_{k+1} - w_q^{(\tau)}(k+1)), \\ \mu_q^{(\tau+1)}(k) = \frac{\sum_{p=1}^k (U_q^{(\tau+1)}(p))^{\beta} S^{-1}(\tilde{x}_p, w_q^{(\tau+1)}(k))}{\sum_{p=1}^k (U_q^{(\tau+1)}(p))^{\beta}}. \end{array} \right. \quad (1.76)$$

Адаптивна нечітка робастна кластеризація даних з пропусками на основі міри подібності. Для вирішення задачі робастної кластеризації даних з пропусками введемо в розгляд часткову міру подібності (PCM), яка є гібридом часткової відстані (PD) (1.36) і міри подібності (SM) (1.69). Нескладно побачити, що така PSM має вигляд

$$S_P(\tilde{x}_k, w_q) = \frac{\sigma^2}{\sigma^2 + D_P^2(\tilde{x}_k, w_q)}, \quad (1.77)$$

що дозволяє отримати алгоритми з бажаними властивостями на основі процедур, що описані вище.

Так, на основі процедур (1.74) і (1.77) можна ввести адаптивний робастний імовірнісний алгоритм нечіткої кластеризації даних з пропусками:

$$\left\{ \begin{array}{l}
U_q^{(\tau+1)}(k) = \frac{(S_P(\mathbf{x}_k^{(\tau)}, w_q^{(\tau)}(k)))^{\frac{1}{\beta-1}}}{\sum_{l=1}^m (S_P(\mathbf{x}_k^{(\tau)}, w_l^{(\tau)}))^{\frac{1}{\beta-1}}}, \\
\mathbf{x}_{ki}^{(\tau)} = w_{qi}^{(\tau)}, w_q^{(\tau)}(k) = \arg \max_q \{S_P(\tilde{x}_k^{(\tau)}, w_1^{(\tau)}(k)), \dots, S_P(\tilde{x}_k^{(\tau)}, w_m^{(\tau)}(k))\}, \\
w_q^{(Q)}(k) = w_q^{(0)}(k+1), \\
w_q^{(\tau+1)}(k+1) = w_q^{(\tau)}(k+1) + \eta(k+1) \frac{(U_q^{(Q)}(k))^\beta}{(\sigma^2 + \|\mathbf{x}_{k+1}^{(\tau)} - w_q^{(\tau)}(k+1)\|^2)^2} (\mathbf{x}_{k+1}^{(\tau)} - w_q^{(\tau)}(k+1)),
\end{array} \right. \quad (1.78)$$

а на основі процедур (1.75) і (1.78) нескладно записати адаптивний робастний можливісний алгоритм нечіткої кластеризації даних з пропусками. Таким чином, використання часткової міри подібності, що ґрунтується на частковій відстані (1.35), дозволяє вирішувати задачі нечіткої кластеризації даних, які містять як пропуски, так і аномальні спостереження.

$$U_q^{(\tau+1)}(k) = \frac{1}{1 + \left(\frac{S^{-1}(\mathbf{x}_k^{(\tau)}, w_q^{(\tau)}(k))}{\mu_q^{(\tau)}(k)} \right)^{\frac{1}{\beta-1}}},$$

$$\mathbf{x}_{ki}^{(\tau)} = w_{qi}^{(\tau)}, w_q^{(\tau)}(k) = \arg \max_q \{S_P(\tilde{x}_k^{(\tau)}, w_1^{(\tau)}(k)), \dots, S_P(\tilde{x}_k^{(\tau)}, w_m^{(\tau)}(k))\}$$

$$w_q^{(Q)}(k) = w_q^{(0)}(k+1),$$

$$w_q^{(\tau+1)}(k+1) = w_q^{(\tau)}(k+1) + \eta(k+1) \frac{(U_q^{(Q)}(k))^\beta}{(\sigma^2 + \|\mathbf{x}_{k+1}^{(\tau)} - w_q^{(\tau)}(k+1)\|^2)^2} (\mathbf{x}_{k+1}^{(\tau)} - w_q^{(\tau)}(k+1)),$$

$$\mu_q^{(\tau+1)}(k) = \frac{\sum_{p=1}^k (U_q^{(\tau+1)}(p))^\beta S_P^{-1}(\mathbf{x}_p, w_q^{(\tau+1)}(k))}{\sum_{p=1}^k (U_q^{(\tau)}(p))^\beta}.$$

1.3 Адаптивна нечітка кластеризація спотворених даних з використанням оптимізаційних процедур

На сьогодні є цілий ряд підходів для обробки даних, які містять пропуски, але всі вони здатні працювати лише у тих випадках, коли масив початкових спостережень заданий заздалегідь у повному обсязі і не змінюється в процесі обробки. В той же час є широкий клас задач, коли дані надходять на обробку послідовно у режимі реального часу так, як це відбувається при навчанні самоорганізованих мап Кохонена або його модифікацій. У зв'язку з цим вище була наведена адаптивна нейро-фазі мережа Кохонена, яка призначена для вирішення задачі кластеризації даних з пропусками на основі стратегії часткових відстаней (PDSFCM). В той же час у ситуаціях, коли кількість таких пропусків занадто велика, стратегія часткових відстаней може виявитися не ефективною, у зв'язку з чим може виникнути необхідність поряд з вирішенням задачі нечіткої кластеризації одночасно відновлювати відсутні спостереження. У цій ситуації найбільш ефективним може виявитися підхід, в основі якого лежить, так названа, стратегія оптимального розширення (OCSFCM) [55]. У зв'язку з цим цікава сама задача on-line кластеризації даних з використанням стратегії оптимального розширення адаптивної на випадок, коли інформація оброблюється у послідовному режимі, а її обсяг заздалегідь не визначено.

1.3.1 Адаптивна ймовірнісна нечітка кластеризація даних з пропусками на основі стратегії оптимального розширення

Стратегія оптимального розширення полягає у тому, що елементи підмасива X_G розглядаються як додаткові змінні, по яким також здійснюється оптимізація прийнятої цільової функції E . Таким чином, паралельно з кластеризацією (оптимізація E по $U_q(k)$ и w_q) здійснюється

оцінювання відсутніх спостережень (оптимізація E по $\tilde{x}_{ki} \in X_G$). У цьому випадку алгоритм нечітких с-середніх на основі стратегії оптимального розширення може бути записан у вигляді наступної послідовності кроків [55]:

1. Завдання початкових умов для роботи алгоритма: $\beta > 0; 1 < m < N$; $\varepsilon > 0$; $w_q^{(0)}$; $1 \leq q \leq m$; $\tau = 0, 1, 2, \dots, Q$; $X_G^{(0)} = \{-1 \leq \mathfrak{E}_{ki}^{(0)} \leq 1\}$,

де $X_G^{(0)} - N_G (1 \leq N_G \leq (n-1)N)$ початкових довільних оцінок $\mathfrak{E}_{ki}^{(0)}$ відсутніх спостережень $\tilde{x}_{ki} \in X_G$;

2. Розрахунок рівней належності шляхом вирішення задачі оптимізації:

$$\begin{aligned} U_q^{(\tau+1)}(k) &= \arg \min_{U_q(k)} E(U_q(k), w_q^{(\tau)}, X_G^{(\tau)}) = \frac{(D^2(\mathfrak{E}_k^{(\tau)}, w_q^{(\tau)}))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (D^2(\mathfrak{E}_k^{(\tau)}, w_l^{(\tau)}))^{\frac{1}{1-\beta}}} = \\ &= \frac{(\|\mathfrak{E}_k^{(\tau)} - w_q^{(\tau)}\|^2)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (\|\mathfrak{E}_k^{(\tau)} - w_l^{(\tau)}\|^2)^{\frac{1}{1-\beta}}}, \end{aligned}$$

(тут вектор $\mathfrak{E}_k^{(\tau)}$ відрізняється від \tilde{x}_k тим, що відсутні спостереження $\tilde{x}_{ki} \in X_G$ замінені оцінками $\mathfrak{E}_{ki}^{(\tau)}$, розрахованими на τ -ой епосі обробки даних);

3. Розрахунок центроїдів кластерів шляхом вирішення задачі оптимізації:

$$w_q^{(\tau+1)} = \arg \min_{w_q} E(U_q^{(\tau+1)}(k), w_q, X_G^{(\tau)}) = \frac{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^{\beta} \mathfrak{E}_k^{(\tau)}}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^{\beta}};$$

4. Перевірка умов зупину:

якщо $\|w_q^{(\tau+1)} - w_q^{(\tau)}\| < \varepsilon \quad \forall 1 \leq q \leq m$ або $\tau = Q$, алгоритм закінчує роботу, в протилежному випадку повертатися до кроку 5;

5. Відновлення відсутніх спостережень шляхом вирішення задачі оптимізації

$$X_G^{(\tau+1)} = \arg \min_{X_G} E(U_q^{(\tau+1)}(k), w_q^{(\tau+1)}, X_G),$$

або, що є теж саме

$$\frac{\partial E(U_q^{(\tau+1)}(k), w_q^{(\tau+1)}, X_G)}{\partial \xi_{ki}} = 0,$$

відкіль

$$\xi_{ki}^{(\tau+1)} = \frac{\sum_{q=1}^m (U_q^{(\tau+1)}(k))^\beta w_{qi}^{(\tau+1)}}{\sum_{q=1}^m (U_q^{(\tau+1)}(k))^\beta}.$$

Обробка інформації за допомогою цього алгоритма організовується у вигляді послідовності

$$w_q^{(0)} \rightarrow U_q^{(1)} \rightarrow \xi_{ki}^{(1)} \rightarrow w_q^{(1)} \rightarrow U_q^{(2)} \rightarrow \dots \rightarrow w_q^{(\tau)} \rightarrow U_q^{(\tau+1)} \rightarrow \xi_{ki}^{(\tau+1)} \rightarrow w_q^{(\tau+1)} \rightarrow \dots$$

$$\dots \rightarrow w_q^{(Q)}.$$

1.3.2 Адаптивна можливісна кластеризація даних з пропусками на основі стратегії оптимального розширення

Основний недолік ймовірносних алгоритмів нечіткої кластеризації

пов'язаний з обмеженням на суму належностей, яка повинна дорівнюватись одиниці, тобто належностям неявно надається сенс ймовірностей належностей кожного вектора-образа до кожного з можливих класів. Для ослаблення цього обмеження були наведені, так названі, можливісні алгоритми нечіткої кластеризації [88].

Цільова функція можливої нечіткої кластеризації має вигляд

$$E(U_q(k), w_q, \mu_q) = \sum_{k=1}^N \sum_{q=1}^m U_q^\beta(k) D^2(\tilde{x}_k, w_q) + \sum_{q=1}^m \mu_q \sum_{k=1}^N (1 - U_q(k))^\beta$$

(тут скалярний параметр $\mu \geq 0$ визначає відстань, на якій рівень належності приймає значення 0,5, тобто, якщо $D^2(\tilde{x}_k, w_q) = \mu_q$, то $U_q(k) = 0,5$), а її мінімізація по $U_q(k)$, w_q , μ_q веде до результату:

$$\begin{cases} U_q^{(\tau+1)}(k) = 1 / \left(1 + (D^2(\tilde{x}_k, w_q^{(\tau)}) / \mu_q^{(\tau)})^{\frac{1}{\beta-1}} \right), \\ w_q^{(\tau+1)} = \sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta \tilde{x}_k / \sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta, \\ \mu_q^{(\tau+1)} = \sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta D^2(\tilde{x}_k, w_q^{(\tau+1)}) / \sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta. \end{cases} \quad (1.79)$$

У режимі on-line обробки інформації рекурентна форма алгоритму (1.79) може бути записана у вигляді [94]

$$\left\{ \begin{array}{l} U_q(k+1) = \frac{1}{1 + \left(\frac{D^2(\tilde{x}_{k+1}, w_q(k))}{\mu_q(k)} \right)^{\frac{1}{\beta-1}}}, \\ w_q(k+1) = w_q(k) + \eta(k+1)U_q^\beta(k+1)(\tilde{x}_{k+1} - w_q(k)), \\ \mu_q(k+1) = \frac{\sum_{p=1}^{k+1} U_q^\beta(p)D^2(\tilde{x}_p, w_q(k+1))}{\sum_{p=1}^{k+1} U_q^\beta(p)}, \end{array} \right. \quad (1.80)$$

при цьому другі співвідношення у (1.80) відрізняються тільки виглядом функції сусідства $U_q^\beta(k+1)$. Таким чином, рекурентна нечітка можливісна кластеризація також ґрунтується на принципах конкурентного навчання по Кохонену.

Розглядаючи ситуацію з відсутніми спостереженнями і використовуючи стратегію оптимального розширення, у якості цільової функції можливої нечіткої кластеризації будемо використовувати вираз

$$E(U_q(k), w_q, \mu_q, X_G) = \sum_{k=1}^N \sum_{q=1}^m U_q^\beta(k) D^2(\mathfrak{X}_k, w_q) + \sum_{q=1}^m \mu_q \sum_{k=1}^N (1 - U_q(k))^\beta,$$

а його мінімізація веде до системи очевидних співвідношень

$$\left\{ \begin{array}{l} \frac{\partial E(U_q(k), w_q, \mu_q, X_G)}{\partial U_q(k)} = 0, \\ \nabla_{w_q} E(U_q(k), w_q, \mu_q, X_G) = \vec{0}, \\ \frac{\partial E(U_q(k), w_q, \mu_q, X_G)}{\partial \mathfrak{X}_{ki}} = 0, \\ \frac{\partial E(U_q(k), w_q, \mu_q, X_G)}{\partial \mu_q} = 0, \end{array} \right.$$

або

$$\left\{ \begin{array}{l} U_q^{(\tau+1)}(k) = 1 / \left(1 + \left(\frac{D^2(\mathfrak{E}_k^{(\tau)}, w_q^{(\tau)})}{\mu_q^{(\tau)}} \right)^{\frac{1}{\beta-1}} \right), \\ w_q^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta \mathfrak{E}_k^{(\tau)}}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta}, \\ \mathfrak{E}_{ki}^{(\tau+1)} = \frac{\sum_{q=1}^m (U_q^{(\tau+1)}(k))^\beta w_{qi}^{(\tau+1)}}{\sum_{q=1}^m (U_q^{(\tau+1)}(k))^\beta}, \\ \mu_q^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta D^2(\mathfrak{E}_k^{(\tau+1)}, w_q^{(\tau+1)})}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta}. \end{array} \right. \quad (1.81)$$

Аналогічно рекурентній нечіткій можливісній кластеризації на основі стратегії оптимального розширення можна організувати процес можливої кластеризації. При цьому аналогом алгоритма (1.79) є процедура у вигляді

$$\left\{ \begin{array}{l} U_q^{(\tau+1)}(k+1) = \frac{1}{1 + \left(\frac{\|\mathfrak{E}_{k+1}^{(\tau)} - w_q(k)\|^2}{\mu_q^{(\tau)}} \right)^{\frac{1}{\beta-1}}}, \\ \mathfrak{E}_{k+1,i}^{(\tau+1)} = \frac{\sum_{q=1}^m (U_q^{(\tau+1)}(k+1))^\beta w_{qi}(k)}{\sum_{q=1}^m (U_q^{(\tau+1)}(k+1))^\beta}, \\ \mu_q^{(\tau+1)}(k+1) = \frac{\sum_{p=1}^{k+1} (U_q^{(\tau+1)}(p))^\beta \|\mathfrak{E}_p^{(\tau+1)} - w_q(k)\|^2}{\sum_{p=1}^{k+1} (U_q^{(\tau+1)}(p))^\beta}, \\ w_q(k+1) = w_q(k) + \eta(k+1)(U_q^{(Q)}(k+1))^\beta (\mathfrak{E}_{k+1}^{(Q)} - w_q(k)), \end{array} \right. \quad (1.82)$$

або у прискореному часі

$$\left\{ \begin{array}{l}
 U_q^{(\tau+1)}(k+1) = \frac{1}{1 + \left(\frac{\|\epsilon_{k+1}^{(\tau)} - w_q^{(\tau)}(k+1)\|^2}{\mu_q^{(\tau)}} \right)^{\frac{1}{\beta-1}}}, \\
 w_q^{(0)}(k+1) = w_q^{(Q)}(k), \\
 w_q^{(\tau+1)}(k+1) = w_q^{(\tau)}(k+1) + \eta(k+1)(U_q^{(\tau+1)}(k+1))^\beta * \\
 \quad * (\epsilon_{k+1}^{(\tau)} - w_q^{(\tau)}(k+1)), \\
 \epsilon_{k+1,i}^{(\tau+1)} = \frac{\sum_{q=1}^m (U_q^{(\tau+1)}(k+1))^\beta w_{qi}^{(\tau+1)}(k+1)}{\sum_{q=1}^m (U_q^{(\tau+1)}(k+1))^\beta}, \\
 \mu_q^{(\tau+1)}(k+1) = \frac{\sum_{p=1}^{k+1} (U_q^{(\tau+1)}(p))^\beta \|\epsilon_p^{(\tau+1)} - w_q^{(\tau+1)}(k+1)\|^2}{\sum_{p=1}^{k+1} (U_q^{(\tau+1)}(p))^\beta}.
 \end{array} \right. \quad (1.83)$$

Зрозуміло, що алгоритм (1.83) з обчислювальної точки зору є найбільш громіздким, але до його переваг можна віднести те, що з його допомогою можна в on-line режимі виявляти появу нових кластерів, а, окрім цього, в процедурах (1.82), (1.83) немає необхідності накопичення оброблюваної вибірки, що важливо у задачах (наприклад, Web Mining), де обсяги оброблюваної інформації є великими.

1.3.3 Адаптивна нечітка кластеризація даних з відсутніми спостереженнями з використанням стратегії найближчого прототипу - центроїду

Стратегія ближчого прототипа-центроїда може бути розглянута у якості гібрида стратегії оптимального розширення і часткових відстаней та складається з послідовності кроків.

1. Задання початкових умов: фазифікатора $\beta > 0; m$, необхідної точності $\varepsilon > 0$, прототипів (центроїдів) кластерів $w_q^{(0)}$, кількість епох обробки $\tau \leq Q$, $X_G^{(0)} = \{-1 \leq \mathfrak{F}_{ki}^{(0)} \leq 1\} - N_G$ довільних оцінок відсутніх спостережень $\tilde{x}_{ki} \in X_G$;

2. Розрахунок рівнів належності:

$$U_q^{(\tau+1)}(k) = \left(\sum_{l=1}^m (\|\mathfrak{F}_k^{(\tau)} - w_l^{(\tau)}\|^2)^{\frac{1}{1-\beta}} \right)^{-1} (\|\mathfrak{F}_k^{(\tau)} - w_q^{(\tau)}\|^2)^{\frac{1}{1-\beta}},$$

(тут вектор $\mathfrak{F}_k^{(\tau)}$ відрізняється від \tilde{x}_k тим, що відсутні спостереження \tilde{x}_{ki} замінені оцінками $\mathfrak{F}_k^{(\tau)}$);

3. Розрахунок прототипів (центроїдів) кластерів:

$$w_q^{(\tau+1)} = \left(\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta \right)^{-1} \sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta \mathfrak{F}_k^{(\tau)};$$

4. Перевірка умов зупинки:

якщо $\|w_q^{(\tau+1)} - w_q^{(\tau)}\| < \varepsilon \forall q$ або $\tau = Q$, зупинка; інакше йти до кроку 5;

5. Оцінка відсутніх спостережень шляхом знаходження прототипа $w_q^{(\tau+1)}$ ближчого до \tilde{x}_k у сенсі часткової відстані

$$D_P^2(\tilde{x}_k, w_q) = \frac{n}{\delta_{k\Sigma}} \sum_{i=1}^n (\tilde{x}_{ki} - w_{qi})^2$$

знаходження

$$w_q^{(\tau+1)} = \arg \min_q \{D_P^2(\tilde{x}_k, w_1^{(\tau+1)}), \dots, D_P^2(\tilde{x}_k, w_m^{(\tau+1)})\}$$

та заміна відсутніх спостережень \tilde{x}_{ki} координатами $\mathbf{x}_{ki}^{(\tau+1)} = w_{qi}^{(\tau+1)}$.

Далі йти до кроку 2.

Нескладно бачити, що на п'ятому кроці, якщо розглядати його з позиції самоорганізованих мап Кохонена [80], знаходиться нейрон-переможець, найближчий до надійшовшого спостереження \tilde{x}_k у сенсі часткових відстаней D_p^2 .

Далі обставина дозволяє переписати стратегію найближчого прототипа у вигляді

$$\begin{cases} U_q^{(\tau+1)}(k) = \left(\sum_{l=1}^m (\|\mathbf{x}_k^{(\tau)} - w_l(k)\|^2)^{\frac{1}{1-\beta}} \right)^{-1} (\|\mathbf{x}_k^{(\tau)} - w_q(k)\|^2)^{\frac{1}{1-\beta}}, \\ \text{зде } \mathbf{x}_{ki}^{(\tau)} = w_{qi}(k), \quad w_q(k) = \arg \min_q \{D_p^2(\tilde{x}_k, w_1(k)), \dots, D_p^2(\tilde{x}_k, w_m(k))\}, \\ w_q(k+1) = w_q(k) + \eta(k+1)(U_q^{(Q)}(k))^\beta (\mathbf{x}_k^{(Q)} - w_q(k)) \quad \forall q = 1, 2, \dots, m, \end{cases} \quad (1.84)$$

при цьому належності між двома тактами реального часу k и $k+1$ налаштовуються у прискореному часі $\tau = 0, 1, 2, \dots, Q$, а центроїди у on-line режимі $k = 0, 1, 2, \dots, N, \dots$ за допомогою правила самоонавчання Кохонена «Переможець отримує більше» з функцією сусідства $(U_q^{(Q)}(k))^\beta$.

Таким чином, процес нечіткої кластеризації даних з пропусками може бути організований у послідовному режимі за допомогою звичайної мапи Кохонена зі спеціальною функцією сусідства $(U_q^{(Q)}(k))^\beta$, яка має форму Кошиана.

Центроїди-прототипи можна також перерахувати у прискореному часі по типу послідовного співвідношення в (1.84), хоча це ускладнює обчислювальну реалізацію процедури кластеризації, яка у цьому випадку приймає вигляд:

$$\left\{ \begin{array}{l} U_q^{(\tau+1)}(k) = \frac{(\|\mathfrak{X}_k^{(\tau)} - w_q^{(\tau)}(k)\|^2)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (\|\mathfrak{X}_k^{(\tau)} - w_l^{(\tau)}(k)\|^2)^{\frac{1}{1-\beta}}}, \\ \text{зде } \mathfrak{X}_{ki}^{(\tau)} = w_{qi}^{(\tau)}(k), \quad w_q^{(\tau)}(k) = \arg \min \{D_P^2(\tilde{x}_k, w_1^{(\tau)}(k)), \dots, D_P^2(\tilde{x}_k, w_m^{(\tau)}(k))\}, \\ w_q^{(Q)}(k) = w_q^{(Q)}(k+1), \\ w_q^{(\tau+1)}(k+1) = w_q^{(\tau)}(k+1) + \eta(k+1)(U_q^{(Q)}(k))^\beta (\mathfrak{X}_k^{(\tau)} - w_q^{(\tau)}(k+1)). \end{array} \right. \quad (1.85)$$

1.3.4 Можливісна адаптивна стратегія найближчого центроїда

Основний недолік ймовірнісних алгоритмів нечіткої кластеризації (FCM та йому подібні процедури) пов'язаний з жорстким обмеженням на суму належностей кожного вектора-образу, яка повинна дорівнюватися одиниці, тобто неявно належностям надається сенс ймовірностей, що далеко не завжди коректно з точки зору розглядуваної проблеми. Для ослаблення цього обмеження були наведені можливісні алгоритми нечіткої кластеризації, при цьому базова процедура (PCM) має вигляд:

$$\left\{ \begin{array}{l} U_q^{(\tau+1)}(k) = 1 / \left(1 + \left(\frac{\|\tilde{x}_k - w_q^{(\tau)}\|^2}{\mu_q^{(\tau)}} \right) \right), \\ w_q^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta}, \\ \mu_q^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta \|\tilde{x}_k - w_q^{(\tau+1)}\|^2}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta}, \end{array} \right.$$

де $\mu_q \geq 0$ визначає відстань, на якій рівень належності приймає значення 0,5, тобто якщо $\|\tilde{x}_k - w_q\|^2 = \mu_q$, то $U_q(k) = 0,5$.

Можливісна стратегія ближчого прототипа-центроїда у цьому випадку може бути записана у вигляді наступної послідовності кроків:

1. Задання початкових умов: фаззифікатора $\beta > 0$, m , необхідної точності $\varepsilon > 0$, прототипів (центроїдів) кластерів $w_q^{(0)}$, кількість епох обробки $\tau \leq Q$, $X_G^{(0)} = \{-1 \leq \mathbf{x}_{ki}^{(0)} \leq 1\} - N_G$ довільних оцінок відсутніх спостережень $\tilde{x}_{ki} \in X_G$;

2. Розрахунок рівнів належності:

$$U_q^{(\tau+1)}(k) = \frac{1}{1 + \left(\frac{\|\mathbf{x}_k^{(\tau)} - w_q^{(\tau)}\|^2}{\mu_q^{(\tau)}} \right)^{\frac{1}{\beta-1}}};$$

3. Розрахунок прототипів (центроїдів) кластерів:

$$w_q^{(\tau+1)}(k) = \frac{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^{\beta} \mathbf{x}_k^{(\tau)}}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^{\beta}};$$

4. Перевірка умов зупинки: якщо $\|w_q^{(\tau+1)} - w_q^{(\tau)}\| < \varepsilon \quad \forall q$ або $\tau = Q$, зупинка; інакше йти до кроку 5;

5. Оцінка відсутніх спостережень шляхом знаходження прототипу $w_q^{(\tau+1)}$ найближчого до \tilde{x}_k у сенсі часткової відстані

$$D_P^2(\tilde{x}_k, w_q) = \frac{n}{\delta_{k\Sigma}} \sum_{i=1}^n (\tilde{x}_{ki} - w_{qi})^2 \delta_{ki}$$

знаходження

$$w_q^{(\tau+1)} = \arg \min_q \{D_p^2(\tilde{x}_k, w_1^{(\tau+1)}), \dots, D_p^2(\tilde{x}_k, w_m^{(\tau+1)})\}$$

і заміна відсутніх спостережень \tilde{x}_{ki} координатами $\mathfrak{E}_{ki}^{(\tau+1)} = w_{qi}^{(\tau+1)}$.

6. Розрахунок скалярного параметру відстані

$$\mu_q^{(\tau+1)} = \frac{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta \|\mathfrak{E}_k^{(\tau+1)} - w_q^{(\tau+1)}\|^2}{\sum_{k=1}^N (U_q^{(\tau+1)}(k))^\beta}.$$

Далі йдемо до кроку 2.

Аналогічно ймовірнісній адаптивній кластеризації на основі стратегії найближчого центроїда можна організувати процес можливісної кластеризації. При цьому аналогом алгоритма (1.84) є процедура вигляду

$$\left\{ \begin{array}{l} U_q^{(\tau+1)}(k) = \frac{1}{1 + \left(\frac{\|\mathfrak{E}_k^{(\tau)} - w_q(k)\|^2}{\mu_q^{(\tau)}} \right)^{\frac{1}{\beta-1}}}, \\ \text{зде } \mathfrak{E}_{ki}^{(\tau)} = w_{qi}(k), \quad w_q(k) = \arg \min_q \{D_p^2(\tilde{x}_k, w_1(k)), \dots, D_p^2(\tilde{x}_k, w_m(k))\}, \\ w_q(k+1) = w_q(k) + \eta(k+1)(U_q^{(Q)}(k))^\beta (\mathfrak{E}_k^{(Q)} - w_q(k)) \quad \forall q = 1, 2, \dots, m, \\ \mu_q^{(\tau+1)} = \frac{\sum_{p=1}^k (U_q^{(\tau+1)}(p))^\beta \|\mathfrak{E}_k^{(\tau)} - w_q(k)\|^2}{\sum_{p=1}^k (U_q^{(\tau+1)}(p))^\beta} \end{array} \right. \quad (1.86)$$

а алгоритма (1.85) –

$$\left\{ \begin{array}{l}
 U_q^{(\tau+1)}(k) = \frac{1}{1 + \left(\frac{\|\mathfrak{E}_k^{(\tau)} - w_q(k)\|^2}{\mu_q^{(\tau)}} \right)^{\frac{1}{\beta-1}}}, \\
 \text{где } \mathfrak{E}_{ki}^{(\tau)} = w_{qi}^{(\tau)}(k), \quad w_{qi}^{(\tau)}(k) = \arg \min_q \{D_P^2(\tilde{x}_k, w_1^{(\tau)}(k)), \dots, D_P^2(\tilde{x}_k, w_m^{(\tau)}(k))\}, \\
 w_q^{(\tau)}(k+1) = w_q^{(\tau)}(k) + \eta(k+1)(U_q^{(\tau)}(k))^\beta (\mathfrak{E}_{k+1}^{(\tau)} - w_q^{(\tau)}(k+1)) \quad \forall q = 1, 2, \dots, m, \\
 \mu_q^{(\tau+1)} = \frac{\sum_{p=1}^k (U_q^{(\tau+1)}(p))^\beta \|\mathfrak{E}_k^{(\tau+1)} - w_q^{(\tau+1)}(k)\|^2}{\sum_{p=1}^k (U_q^{(\tau+1)}(p))^\beta}.
 \end{array} \right. \quad (1.87)$$

З обчислювальної точки зору можливісні процедури є більш громіздкими, але, до їх переваг можна віднести те, що на їх основі не складно організувати процес виявлення нових кластерів.

2 АДАПТИВНІ МЕТОДИ КЛАСТЕРИЗАЦІЇ ТА КЛАСИФІКАЦІЇ ДАНИХ У ВИГЛЯДІ ТАБЛИЦЬ АБО ПОСЛІДОВНОСТІ ЗА НАЯВНОСТІ АНОМАЛЬНИХ ВИКИДІВ

У цьому розділі введено та досліджено методи on-line кластеризації та класифікації даних у вигляді таблиць «об'єкт-властивість» та коротких часових рядів, включаючи багатовимірні векторні та матричні, за наявності аномальних викидів невідомої природи. Методи описані у вигляді рекурентних процедур або правил нечіткого виведення та орієнтовані на реалізацію в якості алгоритмів навчання-самонавчання гібридних нейро-фаззі систем.

Відмінністю підходу, що пропонується, є використання замість традиційних робастних критеріїв навчання або цільових функцій кластеризації спеціального типу функцій подібності, що придушують аномальні викиди. Ці функції засновані а так званих «часткових метриках», що дозволяє використовувати їх для оброблення даних з пропусками. Таким чином, запропоновані методи є двічі робастними як відносно викидів, так і відносно пропусків, в яких містяться загублені спостереження.

Ще однією особливістю введеного підходу є можливість роботи за умов класі, що перетинаються. Це означає, що одне спостереження може одночасно належати двом або більше класам з різними рівнями належності (в сенсі теорії нечітких множин). Рекурентне представлення методів дозволяє використовувати їх для навчання нечітких самоорганізованих мап, при цьому важливо, що як функції подібності, так і функції належності, так і функції сусідства самоорганізованих мап є за своєю сутністю ядерними (дзвонуватими) конструкціями, що можуть мати однотипну форму кошанів. Це значно спрощує чисельну реалізацію методів та підвищує швидкодію процесів опрацювання викривлених даних.

2.1 Адаптивні методи нечіткої кластеризації

Кластеризація зазвичай використовується в якості апарату розвідувального аналізу даних, а це означає, що дослідник завжди стикається з фундаментальними складнощами, пов'язаними з тим, що деякі характеристики даних невідомі заздалегідь. Зазвичай до них відносять частку і масштаб викидів, число і щільність кластерів, ступінь їх перетину і вкладеності тощо. Складнощі, пов'язані з подібного роду невизначеностями, виникають особливо гостро в разі необхідності виконувати аналіз даних, що послідовно надходять в on-line режимі. Окрім технічних труднощів, пов'язаних з неможливістю «згадати» довгу вибірку для повторного аналізу з іншими параметрами моделі, виникають складнощі фундаментального характеру, зумовлені змінами характеристик досліджуваного об'єкта в часі.

У даному розділі буде розглянуто ряд підходів і методів, заснованих на цих підходах, які спрямовані на обробку основних видів невизначеностей у даних. Узагальнення та універсалізація параметрів, що використовуються для отримання цих методів, дозволяють створювати моделі з мінімальною кількістю налаштовуваних вручну параметрів, що здатні гнучко пристосовуватися до змін характеристик досліджуваних даних у широких межах. Окрему увагу приділено схожості схем роботи основних методів, що дозволяє згодом розглядати ансамблі подібних моделей, що працюють паралельно задля досягнення консенсусу про властивості даних.

2.1.1 Цільова функція нечіткої кластеризації

Нечітка логіка майже відразу стала використовуватися у задачах класифікації та кластеризації. Інтуїтивним рішенням є описання класів або кластерів нечіткими множинами – тоді завданням кластеризації стає відновлення оптимальних з деякого погляду функцій належності

спостережень до цих множин. Найпростіша цільова функція нечіткої кластеризації має вигляд

$$E(u_j, c_j) = \sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) d(x(k), c_j) \quad (2.1)$$

за обмеження

$$\sum_{j=1}^m u_j(k) = 1, \quad (2.2)$$

де $k = 1, 2, \dots, N$ – номер спостереження;

$u_j(k) \in [0, 1]$ – рівень належності вектора $x(k)$ до j -го кластеру;

c_j – центроїд (прототип) j -го кластеру;

β – невід’ємний параметр фаззифікації (фаззифікатор), що визначає розмитість меж між кластерами;

N – об’єм вибірки даних;

m – кількість кластерів.

Результатом кластеризації є $(N \times m)$ - матриця $U = \{u_j(k)\}$ (так звана «матриця нечіткого розбиття»), яка описує ступінь належності кожного спостереження до кожного з виділених кластерів.

Зауважимо, що в силу обмеження (2.2), елементи матриці U , які відносяться до одного спостереження, можуть розглядатися як імовірності гіпотез належності цього спостереження до різних кластерів, через що процедури, породжувані мінімізацією (2.1), називаються імовірнісними алгоритмами нечіткої кластеризації.

2.1.2 Оптимізація цільової функції

Використовуючи у якості метрики квадрат евклідової відстані, тобто прийнявши в (2.1)

$$d(x(k), c_j) = \sqrt{(x(k) - c_j)^T (x(k) - c_j)} = \|x(k) - c_j\|,$$

можна записати функцію Лагранжа для цільової функції (2.1):

$$L(u_j(k), c_j, \lambda(k)) = \sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) \|x(k) - c_j\|^2 + \sum_{k=1}^N \lambda(k) \left(\sum_{j=1}^m u_j(k) - 1 \right), \quad (2.3)$$

де $\lambda(k)$ – невизначений множник Лагранжа.

Розв'язуючи систему рівнянь Каруша-Куна-Таккера, визначимо стаціонарні точки:

$$\begin{cases} \frac{\partial L(u_j(k), c_j, \lambda(k))}{\partial u_j(k)} = 0, \\ \nabla_{c_j} L(u_j(k), c_j, \lambda(k)) = \vec{0} \\ \frac{\partial L(u_j(k), c_j, \lambda(k))}{\partial \lambda(k)} = 0. \end{cases} \quad (2.4)$$

У даному випадку неважко отримати аналітичний розв'язок у вигляді:

$$\left\{ \begin{array}{l} u_j(k) = \frac{\left(\|x(k) - c_j\|^2\right)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m \left(\|x(k) - c_l\|^2\right)^{\frac{1}{1-\beta}}}, \\ c_j = \frac{\sum_{k=1}^N \hat{u}_j^\beta(k) x(k)}{\sum_{k=1}^N \hat{u}_j^\beta(k)}, \\ \lambda(k) = - \left(\left(\sum_{l=1}^m \beta \|x(k) - c_l\|^2 \right)^{\frac{1}{1-\beta}} \right)^{1-\beta} \end{array} \right. \quad (2.5)$$

яке при $\beta = 2$ співпадає із популярним алгоритмом нечітких С-середніх Бездека (fuzzy C-means, FCM, [56]):

$$\left\{ \begin{array}{l} u_j(k) = \frac{\|x(k) - c_j\|^{-2}}{\sum_{l=1}^m \|x(k) - c_l\|^{-2}} \\ c_j = \frac{\sum_{k=1}^N u_j^2(k) x(k)}{\sum_{k=1}^N u_j^2(k)}, \end{array} \right. \quad (2.6)$$

а при $\beta \rightarrow 1$ наближається до результатів, отриманих за допомогою звичайної (чіткої) кластеризації K -середніх (*Hard K-Means, НКМ*).

2.1.3 Лінійний змінний фаззіфікатор

Алгоритм (2.5) не можна в повному розумінні назвати нечітким, оскільки він містить точне значення фаззіфікатора $1 < \beta < \infty$, що зазвичай обирається з емпіричних міркувань та суттєвим чином впливає на отримувані результати. При цьому варіація β від 1 до ∞ відповідає переходу від цілком чітких меж ($\beta \rightarrow 1$) до їх повного розмиття ($\beta \rightarrow \infty$), коли всі спостереження з однаковим рівнем належності належать усім кластерам [95].

Ефект впливу класифікатора на «розмиття» меж продемонстровано на рис. 2.1.

Роль фаззіфікатора було докладно розглянуто Клавонном та Хьопнером у [96], де показано, що досягнення мінімуму функціоналу $E(u_j, c_j, x(k))$ значно простіше, якщо використовувати лінійну вагову функцію фаззіфікації замість експоненційної ваги.

Використовуючи рекомендації з [96], отримуємо найпростіший варіант цільової функції з лінійним змінним фаззіфікатором:

$$E(u_j, c_j) = \sum_{k=1}^N \sum_{j=1}^m (\alpha u_j^2(k) + (1 - \alpha) u_j(k)) \|x(k) - c_j\|^2 \quad (2.7)$$

де $0 \leq \alpha \leq 1$ – лінійний фаззіфікатор.

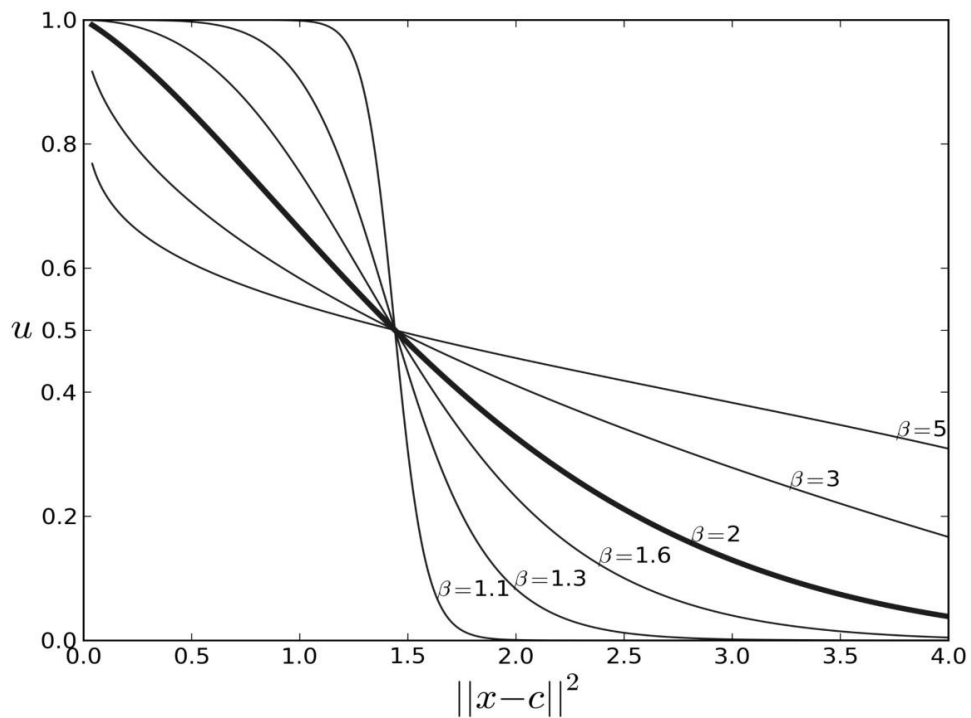


Рисунок 2.1 – Функції належності за різних значень фаззіфікатора

Вводячи функцію Лагранжа

$$L(u_j(k), c_j, \lambda(k)) = \sum_{k=1}^N \sum_{j=1}^m (\alpha u_j^2(k) + (1 - \alpha)u_j(k)) \|x(k) - c_j\|^2 + \sum_{k=1}^N \lambda(k) \left(\sum_{j=1}^m u_j(k) - 1 \right) \quad (2.8)$$

та розв'язуючи для неї систему рівнянь Каруша-Куна-Таккера

$$\begin{cases} \frac{\partial L(u_j(k), c_j, \lambda(k))}{\partial u_j(k)} = (\alpha u_j(k) + 1 - \alpha) \|x(k) - c_j\|^2 + \lambda(k) = 0, \\ \nabla_{c_j} L(u_j(k), c_j, \lambda(k)) = - \sum_{k=1}^N 2(\alpha u_j^2(k) + (1 - \alpha)u_j(k)) (x(k) - c_j) = \vec{0} \\ \frac{\partial L(u_j(k), c_j, \lambda(k))}{\partial \lambda(k)} = \sum_{j=1}^m u_j(k) - 1 = 0, \end{cases} \quad (2.9)$$

приходимо до розв'язку

$$\begin{cases} u_j(k) = -\frac{1 - \alpha}{2\alpha} + \frac{1 + m \frac{1 - \alpha}{2\alpha}}{\sum_{l=1}^m \frac{\|x(k) - c_j\|^2}{\|x(k) - c_l\|^2}}, \\ c_j = \frac{\sum_{k=1}^N (\alpha u_j^2(k) + (1 - \alpha)u_j(k)) x(k)}{\sum_{k=1}^N (\alpha u_j^2(k) + (1 - \alpha)u_j(k))} \\ \lambda(k) = -\frac{1 + m \frac{1 - \alpha}{2\alpha}}{\sum_{l=1}^m (2\alpha \|x(k) - c_l\|^2)^{-1}}. \end{cases} \quad (2.10)$$

Неважко помітити, що при $\alpha = 1$ розв'язок зводиться до FCM-алгоритму Бездека (2.6), з урахуванням чого перше співвідношення (2.10) можна переписати у вигляді

$$u_j(k) = -\frac{1 - \alpha}{2\alpha} + \left(1 + m \frac{1 - \alpha}{2\alpha} \right) u_j^{FCM}(k) \quad (2.11)$$

Для порівняння на рис. 2.2 наведено характер зміни рівней належності в залежності від значень параметра α , при цьому від'ємні значення належності не суперечать загальним положенням теорії систем нечіткого висновування та свідчать про «неналежність» конкретного спостереження до даного кластеру [97, 98].

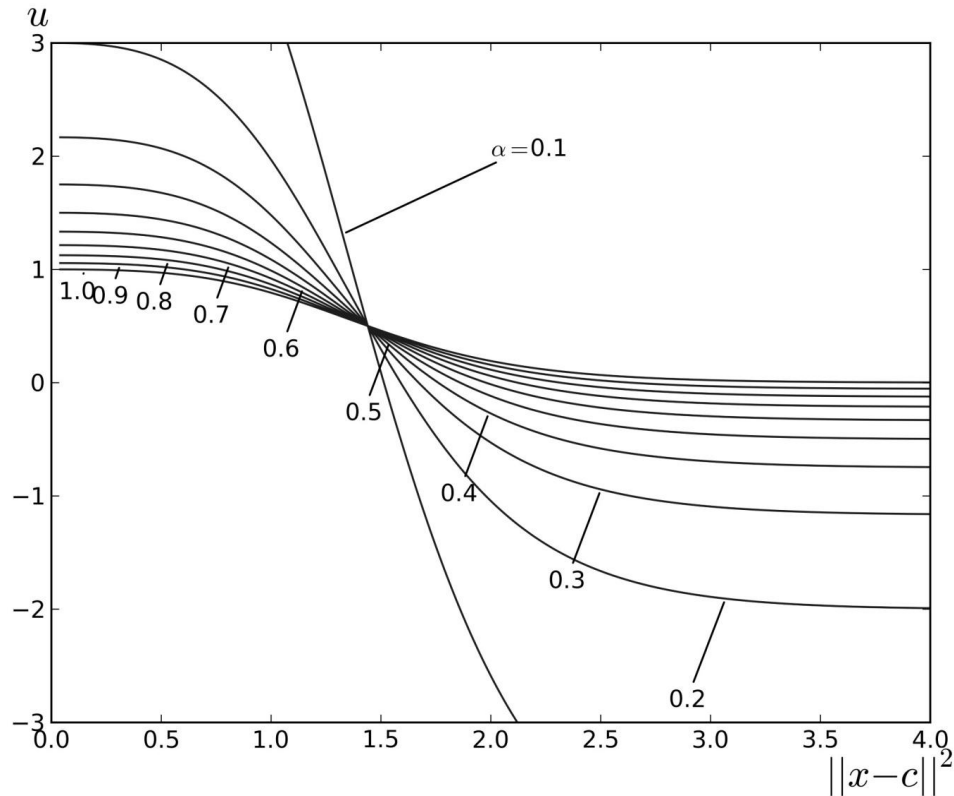


Рисунок 2.2 – Функції належності за різних значень параметра α

Цікаво помітити, що при $\alpha = \frac{1}{3}$ приходимо до компактного виразу

$$u_j(k) = (1 + m)u_j^{FCM}(k). \quad (2.12)$$

Таким чином, лінійний фаззифікатор позбавлений недоліків низької толерантності до невдалого вибору та не зовсім перебдачуваної поведінки, що притаманна ступеневому параметру, але й набуває у якості додаткових переваг ясний фізичний смисл та широкі можливості для спрощення виразів, що містять його.

2.1.4 Адаптивна нечітка кластеризація зі змінним фаззифікатором

Використання процедур, що їх було розглянуто вище, має на увазі, що вибірку, що підлягає кластеризації, задано заздалегідь у розмірі N спостережень та вона не може змінюватись у процесі обробки. У [84, 85, 99, 100] було введено групу інкрементних адаптивних процедур, що дозволяють розв'язувати задачу кластеризації в online режимі.

Для отримання інкрементної процедури за критерієм (2.7), оптимізуємо (2.8) за допомогою процедури нелінійного програмування Ерроу-Гурвіца-Удзави. У результаті отримуємо систему рівнянь, що здійснюють рекурентну мінімізацію цільової функції:

$$\begin{cases} u_j(k+1) = -\frac{1-\alpha}{2\alpha} + \frac{1+m\frac{1-\alpha}{2\alpha}}{\sum_{l=1}^m \frac{\|x(k)-c_j\|^2}{\|x(k)-c_l\|^2}}, \\ c_j(k+1) = c_j(k) + \\ \quad + \eta(k) (\alpha u_j^2(k+1) + (1-\alpha) u_j(k+1)) (x(k+1) - c_j(k)) \end{cases} \quad (2.13)$$

де $\eta(k)$ – параметр кроку налаштування, що визначає швидкість збіжності та зазвичай обирається з емпіричних міркувань у відповідності з умовами Дворецького [101].

Можна побачити, що функціонуюча за таким алгоритмом система приймає на вхід лише одне спостереження $x(k+1)$, розраховує тільки його належність до j -го кластеру $u_j(k+1)$ та одразу же перераховує нове положення відповідного центроїда $c_j(k+1)$, що відповідає принципам інкрементного навчання.

2.1.5 Можливісна нечітка кластеризація зі змінним фаззифікатором

Основні недоліки імовірнісного підходу пов'язані з обмеженням (2.2), що потребує рівності одиниці суми належностей усім кластерам кожного спостереження. Така вимога призводить до некоректного опрацювання викидів (для яких адекватна сума належностей до існуючих кластерів значно менше одиниці) та кластерів, що щільно перетинаються (у цьому випадку спостереженням в області перетину було б адекватно надати рівень належності близький до одиниці для кожного з кластерів, що перетинаються). Цих недоліків у значній мірі позбавлені методи *можливісної* нечіткої кластеризації, запропоновані Крішнапурамом та Келлером [87, 95]. Базова цільова функція можливісної кластеризації має вигляд

$$E(u_j, c_j) = \sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) d(x(k), c_j) + \sum_{j=1}^m \mu_j \sum_{k=1}^N (1 - u_j(k))^\beta, \quad (2.14)$$

при цьому обмеження (2.2) не застосовується, а скалярний параметр $\mu_j > 0$ визначає нечітку межу кластера, тобто відстань від центроїда, на якому рівень належності приймає значення $\frac{1}{2}$ (якщо $d(x(k), c_j) = \mu_j$, то $u_j(k) = 0.5$).

Пряма мінімізація (2.14) за c_j та $u_j(k)$ при використанні в якості метрики квадрата евклідової відстані дає відоме рішення РСМ (Possibilistic C-Means, [87]):

$$\left\{ \begin{array}{l} u_j(k) = \left(1 + \left(\frac{\|x(k) - c_j\|^2}{\mu_j} \right)^{\beta-1} \right)^{-1} \\ c_j = \frac{\sum_{k=1}^N \hat{u}_j^\beta(k) x(k)}{\sum_{k=1}^N \hat{u}_j^\beta(k)}, \\ \mu_j(k) = \frac{\sum_{k=1}^N \hat{u}_j^\beta(k) \|x(k) - c_j\|^2}{\sum_{k=1}^N \hat{u}_j^\beta(k)}. \end{array} \right. \quad (2.15)$$

Цей алгоритм є узагальненням алгоритму навчання Чанга-Лі [84, 102], а при $\beta = 2$ співпадає з градієнтною кластеризацією Парка-Деггера [85]. При $\beta \rightarrow 1$ процедура (2.15) наближається до класичного алгоритму К-середніх, як і (2.5).

Для отримання можливісних процедур кластеризації зі змінним фаззифікатором на основі принципів, що описані в підрозділі 2.1.3, змінимо цільову функцію (2.14) наступним чином:

$$E(u_j, c_j) = \sum_{k=1}^Q \sum_{j=1}^m (\alpha u_j^2(k) + (1 - \alpha) u_j(k)) \|x(k) - c_j\|^2 + \sum_{j=1}^m \mu_j \sum_{k=1}^Q (\alpha(1 - u_j(k)))^2 \quad (2.16)$$

Її мінімізація шляхом розв'язання системи рівнянь Каруша-Куна-Таккера призводить до пакетної процедури можливісної кластеризації з лінійним фаззифікатором:

$$\left\{ \begin{array}{l} u_j(k) = \frac{\left(\alpha \mu_j + (1 - \alpha) \|x(k) - c_j\|^2\right) + \mu_j}{2 \alpha \left(\|x(k) - c_j\|^2 + \mu_j\right)}, \\ c_j = \frac{\sum_{k=1}^N \left(\alpha u_j^2(k) + (1 - \alpha) u_j(k)\right) x(k)}{\sum_{k=1}^N \left(\alpha u_j^2(k) + (1 - \alpha) u_j(k)\right)}, \\ \mu_j = \frac{\sum_{k=1}^N \left(\alpha u_j^2(k) + (1 - \alpha) u_j(k)\right) \|x(k) - c_j\|^2}{\sum_{k=1}^N \left(\alpha u_j^2(k) + (1 - \alpha) u_j(k)\right)} \end{array} \right. \quad (2.17)$$

В адаптивному варіанті при обробці даних, що послідовно надходять, приходимо до інкрементної процедури

$$\left\{ \begin{array}{l} u_j(k+1) = \frac{\left(\alpha \mu_j(k) + (1 - \alpha) \|x(k+1) - c_j(k)\|^2\right) + \mu_j(k)}{2 \alpha \left(\|x(k+1) - c_j(k)\|^2 + \mu_j(k)\right)}, \\ c_j(k+1) = c_j(k) + \eta(k) \left(\alpha u_j^2(k+1) + (1 - \alpha) u_j(k+1)\right) \times \\ \quad \times \left(x(k+1) - c_j(k)\right), \\ \mu_j(k+1) = \frac{\sum_{p=1}^{k+1} \left(\alpha u_j^2(p) + (1 - \alpha) u_j(p)\right) \|x(p) - c_j(k+1)\|^2}{\sum_{p=1}^{k+1} \left(\alpha u_j^2(p) + (1 - \alpha) u_j(p)\right)} \end{array} \right. \quad (2.18)$$

Треба відзначити, що в (2.13) та (2.18) алгоритми налаштування центроїдів кластерів співпадають, а відмінність знаходиться у обчисленнях рівнів належностей до конкретного кластеру. Важливо також, що на відміну від імовірнісних алгоритмів можливісні процедури дозволяють під час обробки даних виявляти появу нових кластерів. Так, якщо рівень належності спостереження $x(k+1)$ до всіх кластерів виявиться нижчим за декотрий заздалегідь заданого порогу, можна говорити про появу $(m+1)$ -го кластеру з початковими координатами центроїда $c_{m+1} = x(k+1)$.

2.1.6 Процедура придушення в нечіткій кластеризації

У системах нечіткої кластеризації з придушенням [103] результати, отримані за допомогою нечітких процедур, модифікуються таким чином, що посилюється значущість результату з найвищим рівнем належності, решта ж значень належностей придушується. Нехай для спостереження $x(k)$ обчислено значення $u_1(k), u_2(k), \dots, u_m(k)$, серед яких знаходиться найбільше $u_p(k)$. Далі це найбільше значення посилюється так, що

$$u_p^S(k) = 1 - \sigma \sum_{\substack{j=1 \\ j \neq p}}^m u_j(k) = 1 - \sigma + \sigma u_p(k), \quad 0 \leq \sigma \leq 1 \quad (2.19)$$

решта ж належностей придушуються у відповідності до виразу

$$u_j^S(k) = \sigma u_j(k), \quad \forall j \neq p. \quad (2.20)$$

Якщо узагальнити вираз (2.19) на можливий випадок, то без обмеження

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

воно приймає вигляд:

$$\begin{aligned}
u_p^S(k) &= \sum_{j=1}^m u_j(k) - \sigma \sum_{\substack{j=1 \\ j \neq p}}^m u_j(k) = \\
&= \sum_{j=1}^m u_j(k) - \sigma \left(\sum_{j=1}^m u_j(k) - u_p(k) \right) = \\
&= \sigma u_p(k) + (1 - \sigma) \sum_{j=1}^m u_j.
\end{aligned} \tag{2.21}$$

Неважко помітити, що зі зменшенням σ результати стають «більш чіткими», а з ростом σ наближаються до рішення, що можна отримати при фаззифікаторі $\beta = 2$. Як видно, параметр σ у (2.19), (2.20), (2.21) має той же зміст регулювання рівня нечіткості, що й α у цільовій функції (2.7). Однак використання обох параметрів не є надлишковим, бо за допомогою параметра фаззифікації α можна маніпулювати компактністю кластерів, а за допомогою параметра придушення σ – їх розделимістю (використовуючи терміни методу Ксі-Бені для оцінювання якості кластеризації [104]). До практичних ефектів введення окремих параметрів для компактності та розделимісті відноситься можливість розділяти кластери з густиною, що сильно відрізняються, що в свою чергу буде розглянуто в розділі експериментальної перевірки.

2.1.7 Методи, що засновані на підході Густафсона-Кеселя

Роздивимось базову цільову функцію нечіткої кластеризації, яку вже було згадано раніше та що має вигляд

$$E(u_j, c_j) = \sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) d(x(k), c_j) \tag{2.22}$$

при обмеженні

$$\sum_{j=1}^m u_j(k) = 1, \quad (2.23)$$

де $k = 1, 2, \dots, N$ – номер спостереження;

$u_j(k) \in [0, 1]$ – рівень належності вектора $x(k)$ до j -го кластера;

c_j – центроїд (прототип) j -го кластера;

β – невід’ємний параметр фаззіфікації (фаззіфікатор), що визначає розмитість меж між кластерами;

N – об’єм вибірки даних;

m – кількість кластерів;

До неї входить функція відстані (метрика) $d^2(x(k), c_j)$. Важливо звернути увагу на те, що функція відстані є компонентою функції, що її оптимізують, а, отже, її зміна безпосередньо в алгоритмі неможлива: зміна метрики породжує новий алгоритм.

Найбільш популярні методи кластеризації, такі як K-means, FCM та інші, отримані за використання евклідової метрики

$$d(x(k), c_j) = \|x(k) - c_j\|^2.$$

Неважко помітити, що значення належності u_j , обчислене за алгоритмами (2.5), (2.10), (2.15), (2.18) та інше значення, що базується на евклідовій метриці, будуть однаковими для всіх точок, що знаходяться на однаковій відстані від центроїда. Межі кластерів, що виділяються такими алгоритмами, завжди мають форму гіперсфери, що природним чином звужує їх застосовність у реальних задачах, у яких форма скупчень сильно залежить від гіперсфери, або ж шкали значень окремих компонент векторів спостережень мають різний масштаб. Априорне припущення щодо сферичної

форми кластерів робить у таких випадках глобальний мінімум цільового функціоналу недосяжним.

2.1.8 Базовий метод Густафсона-Кеселя

Відомим компромісом є алгоритм Густафсона-Кеселя [105], що дозволяє оперувати еліпсоїдальними кластерами різного ступеня витягнутості, не вимагаючи при цьому глибокої переробки алгоритму. Для цього у формулу розрахунку відстаней між векторами вводиться матриця масштабування A , що змінює масштаб простору за кожною координатою:

$$d_A^2(x, c) = (x - c)^T A (x - c) = \|x - c\|_A^2. \quad (2.24)$$

Мінімізуючи цільовий функціонал нечіткої кластеризації (2.22) з метрикою (2.24), отримуємо відому процедуру мінімізації, що включає, окрім розрахунку центрів кластерів c_j , $1 \leq j \leq m$ та рівнів належності u_j , також обчислення коваріаційної матриці кожного кластеру S_j і на її основі матриці масштабування кластеру A_j :

$$\left\{ \begin{array}{l} u_j(k) = \frac{((x(k) - c_j(k))^T A_j (x(k) - c_j(k)))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m ((x(k) - c_l(k))^T A_l (x(k) - c_l(k)))^{\frac{1}{1-\beta}}} \\ c_j = \frac{\sum_{k=1}^N \hat{u}_j^\beta(k) x(k)}{\sum_{k=1}^N \hat{u}_j^\beta(k)}, \\ S_j = \sum_{k=1}^N \hat{u}_j^\beta(k) (x(k) - c_j)(x(k) - c_j)^T, \\ A_j = |S_j|^{\frac{1}{n}} S_j^{-1}. \end{array} \right. \quad (2.25)$$

Отримана таким чином матриця A_j є симетричною і позитивно визначеною, а масштаб, що задається нею, визначається для кожної

координати квадратним коренем відповідного власного вектора, що є оптимальним [106] з точки зору критерію (2.22). Значення власних векторів матриці A_j здатні змінювати форму відповідного кластера, що значно розширює область застосовності алгоритму.

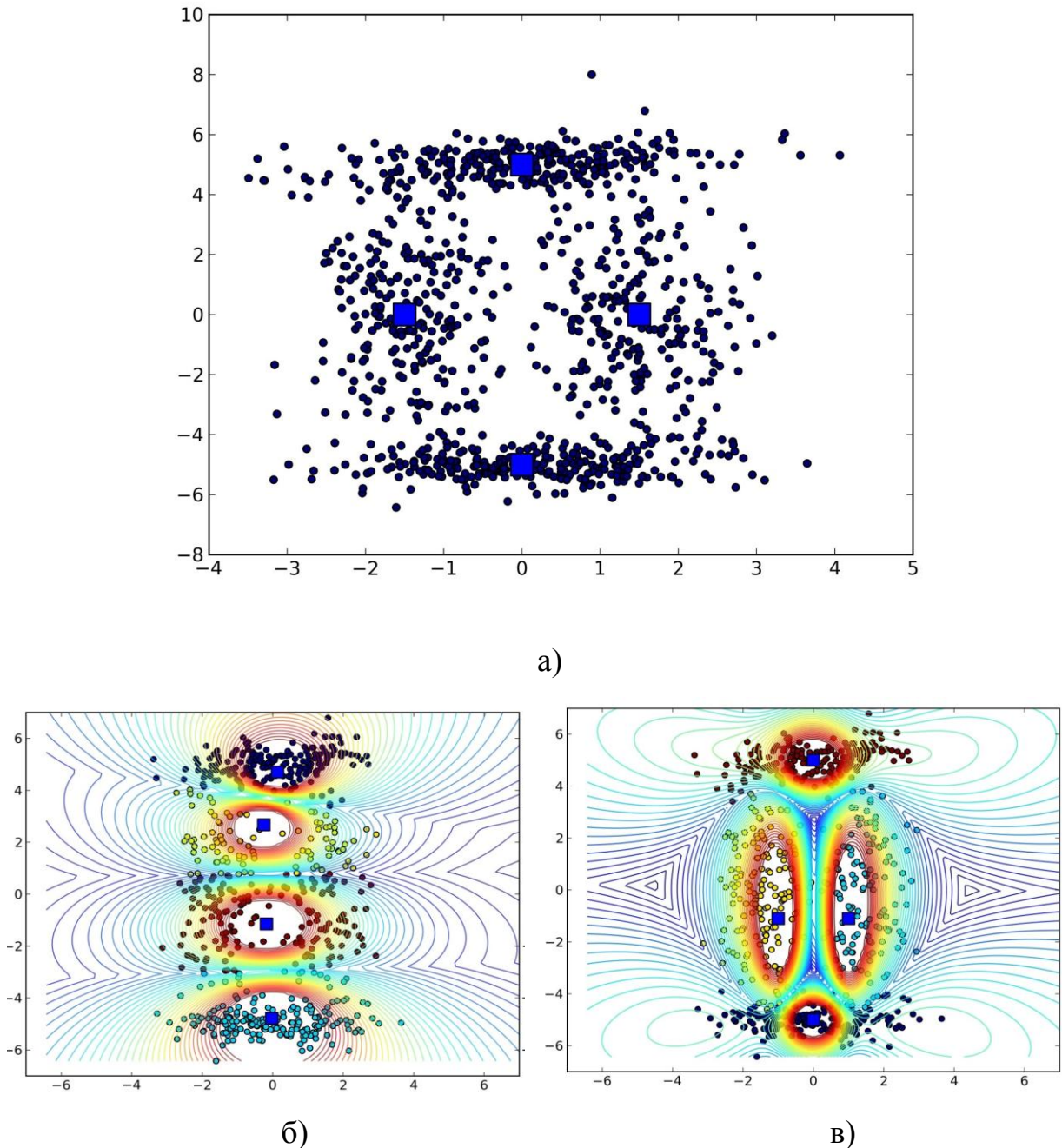


Рисунок 2.3 – Кластеризація даних з витягнутими скупченнями: а – вихідна вибірка; б – розбиття методом FCM; в – розбиття методом Густафсона-Кесея

Ілюстрація різниці роботи алгоритмів показана на рисунку 2.3. Для цього створено штучну вибірку з 1000 точок, що нормально розподілена в околицях 4 центрів (рисунок 2.3а). Особливістю даної вибірки є те, що граничні дані витягнутих кластерів лежать ближче до центру сусіднього кластера, ніж до свого власного. Тому застосування алгоритму нечітких С-середніх не призводить до бажаного розбиття (рисунок 2.3б). У той же час помітно, що центри і параметри розподілу (коваріаційні матриці), відтворені алгоритмом Густафсона-Кеселя (рисунок 2.3в) максимально близькі до заданих спочатку.

2.1.9 Можливісний варіант метода Густафсона-Кеселя

Як вже відзначалось у розділі 2.3.2, алгоритми, що дотримують обмеження (2.23), називаються імовірнісними та володіють суттєвими недоліками, основними з яких є низька стійкість до викидів у даних, а також до кластерів, що суттєво перетинаються. Вочевидь умова (2.23) є занадто жорсткою для багатьох реальних задач: викиди в даних повинні мати близький до нуля ступінь належності до всіх кластерів, а спостереженням, що знаходяться в центрі груп, що істотно перекриваються, доречно було б присвоїти належність близьку до одиниці до кожної з груп. Алгоритми, які не використовують це обмеження, називаються можливісними.

Щоб отримати можливісний алгоритм Густафсона-Кеселя, треба використати цільову функцію, аналогічну (2.14) з точністю до метрики:

$$E(u_j, c_j) = \sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) \|x(k) - c_j\|_A^2 + \sum_{j=1}^m \mu_j \sum_{k=1}^N 1 - u_j(k)^\beta. \quad (2.26)$$

Алгоритм, отриманий завдяки її мінімізації, має вигляд [87]

$$\left\{ \begin{array}{l} u_j(k) = \left(1 + \left(\frac{(x(k) - c_j)^T A_j (x(k) - c_j)}{\mu_j} \right)^{\frac{1}{1-\beta}} \right)^{-1} \\ c_j = \frac{\sum_{k=1}^N \hat{u}_j^\beta(k) x(k)}{\sum_{k=1}^N \hat{u}_j^\beta(k)}, \\ S_j = \sum_{k=1}^N \hat{u}_j^\beta(k) (x(k) - c_j) (x(k) - c_j)^T, \\ A_j = |S_j|^{\frac{1}{n}} S_j^{-1}, \\ \mu_j(k) = \frac{\sum_{k=1}^N \hat{u}_j^\beta(k) (x(k) - c_j)^T A_j (x(k) - c_j)}{\sum_{k=1}^N \hat{u}_j^\beta(k)}. \end{array} \right. \quad (2.27)$$

2.1.10 Адаптивні версії алгоритма Густафсона-Кеселя

Наявність операцій обчислення коваріаційної матриці та її обернення (псевдообернення) призводять до великої обчислювальної складності алгоритму Густафсона-Кеселя, що швидко зростає разом з розміром вибірки; до принципової нестійкості алгоритму до викидів (у цьому випадку матриця виходить погано обумовленою); до пакетної обробки даних.

Використання формули Шермана-Моррісона для накопичувального обернення зростаючої матриці та процедури нелінійного програмування Ерроу-Гурвіца-Удзави для отримання рекурентної процедури оптимізації функціоналу (2.26) дозволяє вивести адаптивну форму алгоритму Густафсона-Кеселя, що працює повністю в рекурентному режимі:

$$\left\{ \begin{array}{l}
u_j(k+1) = \frac{\|x(k+1) - c_j(k)\|_{A_j(k)}^{\frac{2}{1-\beta}}}{\sum_{l=1}^m \|(x(k+1) - c_l(k))\|_{A_j(k)}^{\frac{2}{1-\beta}}}, \\
S_j(k+1) = S_j(k) + \hat{u}_j(k+1) (x(k+1) - c_j(k)) (x(k+1) - c_j(k))^T, \\
S_j^{-1}(k+1) = S_j^{-1}(k) - \\
\frac{\hat{u}_j(k+1) S_j^{-1}(k) (x(k+1) - c_j(k)) (x(k+1) - c_j(k))^T S_j^{-1}(k)}{1 + \hat{u}_j(k+1) (x(k+1) - c_j(k))^T S_j^{-1}(k) (x(k+1) - c_j(k))}, \\
|S_j(k+1)| = |S_j(k)| (1 + \hat{u}_j(k+1) \|x(k+1) - c_j(k)\|^2), \\
A_j(k+1) = |S_j(k+1)|^{\frac{1}{n}} S_j^{-1}(k+1), \\
c_j(k+1) = c_j(k) + \eta(k) \hat{u}_j(k+1) A_j(k+1) (x(k+1) - c_j(k)).
\end{array} \right. \quad (2.28)$$

Особливостями даного алгоритму є рекурентне обернення коваріаційної матриці $S_j^{-1}(k+1)$ за формулою Шермана-Моррісона та рекурентне обчислення її визначника $|S_j(k+1)|$.

Адаптивній формі імовірнісних алгоритмів, незважаючи на зручність використання, притаманні ті ж недоліки, що й їх пакетній формі – у даному випадку це вразливість до викидів і складність з поділом кластерів, що сильно перетинаються. Крім того, такі алгоритми не можуть бути названі у повній мірі алгоритмами навчання без учителя, оскільки експериментатор все ще повинен заздалегідь задати кількість кластерів, використовуючи знання про природу вибірки. Для компенсації цих недоліків логічно привести до адаптивної форми можливісний алгоритм Густафсона-Кесея (2.27). Процедура виведення такого алгоритму аналогічна процедурі виведення алгоритму (2.28), тільки в якості цільового функціоналу використовується вираз (2.26). У результаті отримуємо адаптивний можливісний алгоритм Густафсона-Кесея:

$$\left\{ \begin{array}{l}
u_j(k+1) = \left(1 + \left(\frac{(x(k+1) - c_j(k))^T A_j(k) (x(k+1) - c_j(k))}{\mu_j(k)} \right)^{\frac{1}{1-\beta}} \right)^{-1} \\
S_j(k+1) = S_j(k) + \hat{u}_j^\beta(k+1) (x(k+1) - c_j(k)) (x(k+1) - c_j(k))^T, \\
S_j^{-1}(k+1) = S_j^{-1}(k) - \\
\quad - \frac{\hat{u}_j^\beta(k+1) S_j^{-1}(k) (x(k+1) - c_j(k)) (x(k+1) - c_j(k))^T S_j^{-1}(k)}{1 + \hat{u}_j^\beta(k+1) (x(k+1) - c_j(k))^T S_j^{-1}(k) (x(k+1) - c_j(k))}, \\
|S_j(k+1)| = |S_j(k)| (1 + \hat{u}_j^\beta(k+1) \|x(k+1) - c_j(k)\|^2), \\
A_j(k+1) = |S_j(k+1)|^{\frac{1}{n}} S_j^{-1}(k+1), \\
c_j(k+1) = c_j(k) + \eta(k) \hat{u}_j^\beta(k+1) A_j(k+1) (x(k+1) - c_j(k)), \\
\mu_j(k+1) = \frac{\sum_{p=1}^{k+1} \hat{u}_j^\beta(p) (x(p) - c_j(k+1))^T A_j(k+1) (x(p) - c_j(k+1))}{\sum_{p=1}^{k+1} \hat{u}_j^\beta(p)}.
\end{array} \right. \quad (2.29)$$

Незважаючи на громіздкість, при найближчому розгляді структура виразу аналогічна процедурам (2.28) та (2.18).

Важливо відзначити, що у можливісних алгоритмів є й свої недоліки. У першу чергу, за відсутності обмеження (2.23) ніщо не заважає центроїдам зближуватися або віддалятися необмежено. Внаслідок цього, результат таких алгоритмів може відрізнитися нестабільністю, тобто малі збурення у вихідних даних (а також в ініціалізації центроїдів) призведуть до значної різниці в результуючому розбитті. Основною рекомендацією для боротьби з цією обставиною є використання для ініціалізації можливісного алгоритму результатів роботи декількох ітерацій імовірнісного алгоритму [107, 108].

Тут слід звернути увагу на те, що за винятком виду функції належності $u_j(k+1)$ та обчислення пов'язаного з нею параметра нечіткої межі $\mu_j(k+1)$, адаптивний можливісний алгоритм (2.29) співпадає з адаптивним імовірнісним алгоритмом Густафсона-Кеселя (2.28), що робить зручним їх спільне використання для підвищення стійкості.

2.1.11 Робастний метод нечіткої кластеризації на основі критерію Коші

Протягом останніх десятиліть зростало розуміння того факту, що деякі найбільш поширені процедури (у тому числі ті, які оптимальні у припущенні про нормальність розподілу) досить чутливі до досить малих відхилень від припущень. Ось чому активний розвиток отримали так звані «робастні» процедури (від англ. robust – міцний, здоровий, дужий).

Під терміном робастність розуміється нечутливість моделі до відносно малих відхилень від апріорних припущень про характер даних. Найбільш поширеною проблемою реальних завдань є наявність викидів в даних, що при використанні неробастних процедур призводить до зміщення центроїдів і радіусів кластерів. Чим більше таке зміщення, тим менш робастний процедура

Раніше в даному розділі були розглянуті можливості підходи до кластеризації, здатні виробити адекватно малі рівні належності для аномальних даних. Однак процедура розрахунку центроїдів можливісних методів така ж, як і для імовірнісних, хоча й мала належність послаблює роль викидів на положення центроїдів, цього непрямого впливу не завжди достатньо.

Наявність викидів відбивається на розподілі, що породжується, повільним (у порівнянні з нормальним) спаданням його «хвостів». У роботах [109-112] було запропоновано різні модифікації класичних процедур кластеризації для обробки даних з великою дисперсією. Основний використовуваний для цього підхід - заміна метрики в цільовій функції (2.1) на критерій близькості, що спадає повільніше, ніж квадрат евклідової відстані. Прикладами таких критеріїв можуть служити модуль евклідової відстані, функція Гріна, функція Гауса та інші. У [89] запропоновано використання функції Коші в якості критерію близькості, яка проста в розрахунку та може бути легко виражена в інкрементному вигляді.

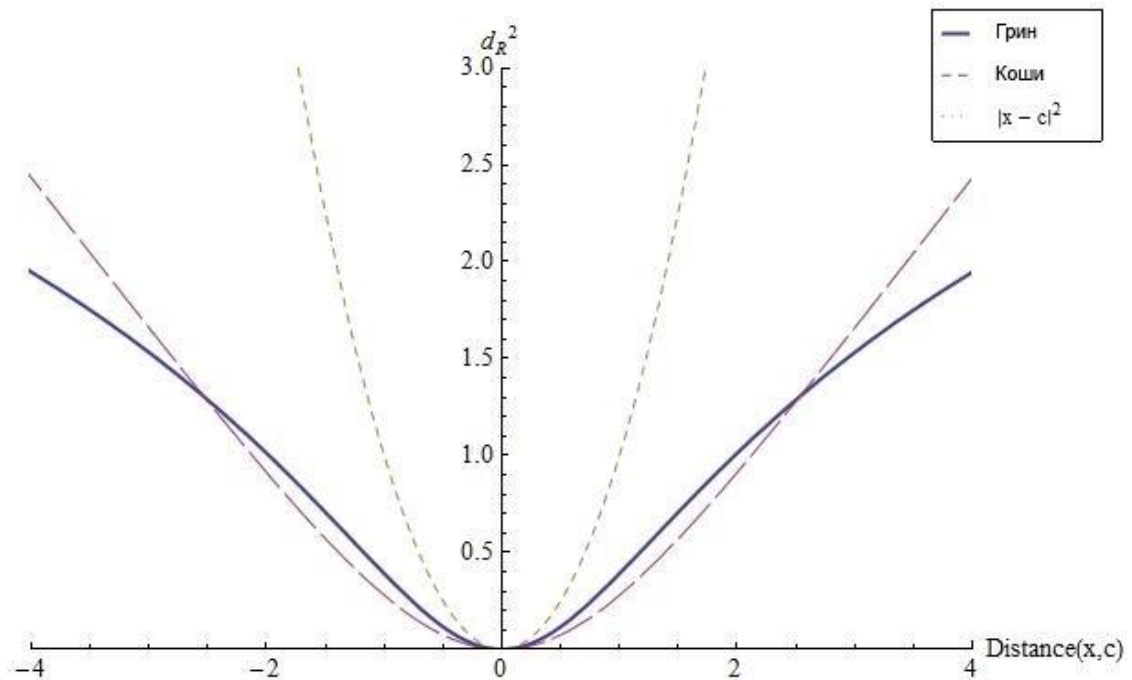


Рисунок 2.4 – Порівняння деяких критеріїв відстані: евклідової, Гріна, Коші

На рисунку 2.4 наведено порівняння деяких критеріїв відстані. На рисунку одразу видно, що функція Коші не є метрикою, оскільки має точки перегину, тобто не дотримується правило трикутника. Це означає, що, розглядаючи функції на основі цього критерію, ми переходимо від поняття «відстані» до менш суворого поняття «міри близькості» або «схожості», що, втім, цілком природно при використанні нечіткої логіки.

2.1.12 Імовірнісний підхід

Міра близькості, заснована на функції Коші, має вигляд

$$d_R^2(x, c) = \frac{\sigma_C^2}{2} \ln \left(1 + \frac{\|x(k) - c\|^2}{\sigma_C^2} \right) \quad (2.30)$$

Введемо цільову функцію робастної нечіткої кластеризації на підставі критерію (2.30)

$$\begin{aligned}
 E_R(u_j, c_j) &= \sum_{k=1}^N \sum_{j=1}^m \hat{u}_j^\beta(k) d_R^2(x(k), c_j) = \\
 &= \sum_{k=1}^N \sum_{j=1}^m \hat{u}_j^\beta(k) \frac{\sigma_C^2}{2} \ln \left(1 + \frac{\|x(k) - c_j\|^2}{\sigma_C^2} \right). \quad (2.31)
 \end{aligned}$$

де $k = 1, 2, \dots$,

N – номер спостереження;

N – обсяг вибірки даних;

m – кількість кластерів;

$u_j(k) \in [0, 1]$ – рівень належності вектора $x(k)$ до j -го кластеру;

c_j – центроїд (прототип) j -го кластеру;

β – невід'ємний параметр фаззіфікації (фаззіфікатор), що визначає розмитість меж між кластерами;

σ – налаштовний параметр «ширини дзвону» (толерантності).

Відповідна їй функція Лагранжа має вигляд:

$$\begin{aligned}
 L_R(u_j(k), c_j, \lambda(k)) &= \sum_{k=1}^N \sum_{j=1}^m \hat{u}_j^\beta(k) \frac{\sigma_C^2}{2} \ln \left(1 + \frac{\|x(k) - c_j\|^2}{\sigma_C^2} \right) + \\
 &+ \sum_{k=1}^N \lambda(k) \left(\sum_{j=1}^m u_j(k) - 1 \right). \quad (2.32)
 \end{aligned}$$

Сідлова точка функції Лагранжа може бути знайдена шляхом рішення системи рівнянь Каруша-Куна-Таккера, аналогічно тому, як це було зроблено для виведення процедур (2.5), (2.10), (2.15) та інших:

$$\left\{ \begin{array}{l} \frac{\partial L_R(u_j(k), c_j, \lambda(k))}{\partial u_j(k)} = 0 \\ \frac{\partial L_R(u_j(k), c_j, \lambda(k))}{\partial \lambda(k)} = 0, \\ \nabla_{C_j} \partial L_R(u_j(k), c_j, \lambda(k)) = \vec{0}. \end{array} \right. \quad (2.33)$$

Перші два рівняння системи (2.33) мають аналітичний розв'язок:

$$\left\{ \begin{array}{l} u_j(k) = \frac{(d_R^2(x(k), c_j))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (d_R^2(x(k), c_l))^{\frac{1}{1-\beta}}}, \\ \lambda(k) = - \left(\sum_{l=1}^m (d_R^2(x(k), c_l))^{\frac{1}{1-\beta}} \right)^{1-\beta}, \end{array} \right. \quad (2.34)$$

у той час як третє рівняння такого розв'язку явно не має. У зв'язку з цим, застосовуючи до (2.32) процедуру Ерроу-Гурвіца-Удзави, приходимо до адаптивного робастного методу нечіткої кластеризації, що здійснює інкрементне налаштування центроїдів і функцій належності:

$$\left\{ \begin{array}{l} u_j(k+1) = \frac{(d_R^2(x(k+1), c_j(k)))^{\frac{1}{1-\beta}}}{\sum_{l=1}^m (d_R^2(x(k+1), c_l(k)))^{\frac{1}{1-\beta}}}, \\ c_j(k+1) = c_j(k) - \eta(k) \nabla_{C_j(k)} d_R^2(x(k+1), c_j(k)) = \\ = c_j(k) + \eta(k) u_j(k+1) \frac{\sigma_C^2 (x(k+1) - c_j(k))}{\sigma_C^2 + \|x(k+1) - c_j(k)\|^2} \end{array} \right. \quad (2.35)$$

У даній процедурі за рахунок метрики частково компенсована вразливість імовірнісних методів до викидів і аномальних спостережень. Тим не менш, вона все ще не дозволяє виділяти види кластерів, що сильно перетинаються, вкладені, переплетені та інше, коректний опис яких вимагає

можливості вказати суму належностей окремих спостережень до сусідніх кластерів, що перевищує одиницю.

2.1.13 Можливісний підхід

Для отримання можливісного варіанту робастної кластеризації на основі критерію Коші найпростіше мінімізувати базову цільову функцію можливісної кластеризації (2.14), використовуючи як міру відстані критерій (2.30):

$$\begin{aligned}
 E(u_j, c_j) &= \sum_{k=1}^N \sum_{j=1}^m \hat{u}_j^\beta(k) d_R^2(x(k), c_j) + \sum_{j=1}^m \mu_j \sum_{k=1}^N (1 - u_j(k))^\beta, = \\
 &= \sum_{k=1}^N \sum_{j=1}^m \hat{u}_j^\beta(k) \frac{\sigma_C^2}{2} \ln \left(1 + \frac{\|x(k) - c_j\|^2}{\sigma_C^2} \right) + \sum_{j=1}^m \mu_j \sum_{k=1}^N (1 - u_j(k))^\beta.
 \end{aligned} \tag{2.36}$$

Розв'язуючи систему рівнянь у частинних похідних аналогічну (2.33), можна знайти аналітичний розв'язок для розрахунку належностей $u_j(k)$ та нечіткої межі $\mu_j(k)$:

$$\begin{cases} u_j(k) = \left(1 + \left(\frac{d_R^2(x(k), c_j)}{\mu_j} \right)^{\beta \frac{1}{\beta-1}} \right)^{-1} \\ \mu_j(k) = \frac{\sum_{k=1}^N \hat{u}_j^\beta(k) d_R^2(x(k), c_j)}{\sum_{k=1}^N \hat{u}_j^\beta(k)}, \end{cases} \tag{2.37}$$

однак рівняння оптимального положення центроїдів c_j не має аналітичного розв'язку. Як і у можливісному випадку, для розв'язання доцільно використати ітеративну процедуру Эрроу-Гурвіца-Удзави, в результаті чого аналогічно до (2.35) отримуємо метод адаптивної можливісної кластеризації за критерієм Коші:

$$\begin{cases} u_j(k+1) = \frac{1}{1 + \left(\frac{d_R^2(x(k+1), c_j(k))}{\mu_j(k)} \right)^{\beta-1}}, \\ c_j(k+1) = c_j(k) + \eta(k) u_j(k+1) \frac{\sigma_C^2 (x(k+1) - c_j(k))}{\sigma_C^2 + \|x(k+1) - c_j(k)\|^2} \\ \mu_j(k+1) = \frac{\sum_{p=1}^{k+1} u_j^2(p) \|x(p) - c_j(k+1)\|^2}{\sum_{p=1}^{k+1} u_j^2(p)}. \end{cases} \quad (2.38)$$

2.2 Мapi Кохонена та їх ансамблі для вирішення задач нечіткої кластеризації

Більшість методів навчання для класичних архітектур штучних нейронних мереж вимагають наявності чисельних значень вихідних сигналів мережі в початковій вибірці даних, тобто відносяться до класу навчання з вчителем. Однак, існують реальні завдання, для яких у початковій вибірці відсутні необхідні вихідні значення мережі і є тільки набір спостережень $x(k)$. Отже, для їх вирішення, тобто для навчання мережі, необхідно витягувати значиму інформацію тільки на основі цього набору $x(k)$. У таких випадках говорять про методи навчання без вчителя або процедурах самонавчання, самоорганізації. Як правило, процедури навчання без вчителя для ШНМ засновані на принципі конкуренції між нейронами.

В якості прикладів згаданих задач можна навести такі:

- кластеризація: у випадку, коли вхідні дані групуються в кластери, їх необхідно виділити для подальшої класифікації, як чіткої, так і нечіткої при істотному перетині кластерів;
- квантування векторів: ця задача виникає, коли потрібно вхідний векторний простір представити у дискретному вигляді оптимальним чином, розбивши його на непересічні підпростори;

- зниження розмірності: у випадку, коли навчальна інформація міститься в підпросторі меншої розмірності, ніж вихідна вибірка. Метою зниження розмірності є отримання оптимального відображення в простір меншої розмірності з найменшими втратами інформації (найчастіше інформативними ознаками вважаються ознаки з найбільшою дисперсією даних);

- вибір ознак: на відміну від задачі зниження розмірності, тут потрібно виділити набір найбільш значимих ознак, не роблячи їх перетворення.

Основна відмінність методу квантування векторів від кластеризації полягає в тому, що при кластеризації процедурою знаходяться області вхідного простору, що містять «схожі» спостереження [56], а у разі квантування векторів основною метою є розбиття всього вхідного простору [51].

2.2.1 Конкурентне навчання

У рамках концепції конкурентного навчання розглядаються процедури самонавчання, які вирішують задачу кластеризації даних в традиційному сенсі [56].

В основі процедури самоорганізації лежать принципи конкурентного самонавчання, а її робота, як і будь-якого іншого методу навчання, починається з ініціалізації синаптичних ваг мережі, обраних або випадково, або за допомогою будь-якого з відомих методів. Сама стандартна процедура самоорганізації реалізується в три основних етапи [51]: конкуренції, кооперації та синаптичної адаптації.

Стандартна архітектура конкурентної мережі (competitive neural network) являє собою повнозв'язний шар з m нейронів, що містять по n рецепторів, які характеризуються n -мірними векторами синаптичних ваг.

$$w_j(k) = w_{j,1}(k), w_{j,2}(k), \dots, w_{j,n}(k)^T, \quad j = 1, 2, \dots, m. \quad (2.39)$$

На вхід мережі, а далі з рецепторного поля на всі нейрони, подається вхідний вектор-спостереження $x(k), k = 1, 2, \dots$. При цьому, на етапі конкуренції, коли подається вхідний вектор, на виході активується тільки один нейрон, так званий нейрон-переможець. У коректно навченої мережі для всіх векторів $x(k)$, які належать одному кластеру, а, отже, близьких у сенсі прийнятої метрики, буде активуватися один і той же нейрон-переможець $w^*(k)$. У якості функцій активації нейронів конкурентної мережі використовуються лінійні функції. Додатково, нейрони мережі містять ваги v_j , що задають так звану топологію карти i , крім безпосереднього значення $y(k)$ вихід мережі, результатом класифікації служить номер (або координати v) нейрона-переможця.

Існує два традиційних метода визначення нейрона-переможця як найближчого до вектору поточного вхідного спостереження $x(k)$ згідно з прийнятою метрикою. Розглянемо кожен з них більш докладно.

Вибір переможця по скалярному добутку. Тут передбачається, що для кожного з нейронів і вхідних образів виконувалася умова нормування на гіперсферу

$$\begin{aligned} \|w_j(k)\|^2 &= w_j(k)^T w_j(k) = 1, \\ \|x(k)\|^2 &= x(k)^T x(k) = 1. \end{aligned} \quad (2.40)$$

Вихід кожного нейрона обчислюється як скалярний добуток векторів синаптичних ваг і вхідного спостереження:

$$y_j(k) = \sum_{l=1}^n w_{j,l}(k)x_l(k) = w_j^T(k)x(k), \quad j = 1, 2, \dots, m. \quad (2.41)$$

Нейрон-переможець $w^*(k)$ обирається відповідно з найбільшим значенням вихідного сигналу нейронів, тобто

$$w^*(k) = w_o(k), o = \underset{j=1, \dots, m}{\operatorname{argmax}} y_j(k) \quad (2.42)$$

Відповідно до правила «переможець отримує все» (winner-takes-all), на виході нейрона-переможця $w^*(k)$ встановлюється значення $y(k) = y_o(k) = 1$, а на виходах решти нейронів $y_j(k) = 0, j \neq o$.

Необхідно відмітити, що при виконанні умов нормування (2.40) скалярний добуток (2.41) можна розглядати у якості міри близькості векторів $x(k)$ і $w_j(k)$, тобто відстань між будь-якими векторами a і b з одиничною нормою $\|a\| = \|b\| = 1$ буде визначатися як

$$D_{\cos}(a, b) = \sqrt{2 - a^T b} = \sqrt{2 - \cos(a, b)} \quad (2.43)$$

Вибір переможця по відстані в евклідовому просторі. У разі ненормованих даних і векторів синаптичних ваг скалярний добуток векторів не може бути мірою їх близькості. Отже, для вибору нейрона-переможця природно використовувати евклідову метрику відстані

$$D_E(a, b) = \|a - b\| = \sqrt{(a - b)^T (a - b)}, \quad (2.44)$$

відповідно до якої

$$w^*(k) = w_o(k), o = \underset{j=1, \dots, m}{\operatorname{argmin}} \|w_j(k) - x(k)\|, \quad (2.45)$$

що при нормуванні (2.40) еквівалентно (2.42).

Налаштування синаптичних ваг. У загальному випадку вибір нейрона-переможця, найближчого до вхідного спостереження згідно з прийнятою метрикою, можна записати як

$$D(w^*(k), x(k)) = \min_j D(w_j(k), x(k)). \quad (2.46)$$

Після етапу вибору переможця відбувається налаштування синаптичних ваг нейрона-переможця у відповідності з наступними процедурами синаптичної адаптації.

Для ненормованого простору ваг:

$$w^*(k+1) = w^*(k) + \eta(k) (x(k) - w^*(k)), \quad (2.47)$$

і не може не впадати в очі схожість цієї процедури з процедурами розрахунку центроїдів методів кластризації, розглянутих у розділі 2.1.

Для нормованого простору ваг:

$$w^*(k+1) = \frac{w^*(k) + \eta(k) (x(k) - w^*(k))}{\|w^*(k) + \eta(k) (x(k) - w^*(k))\|} \quad (2.48)$$

де поділ на норму забезпечує виконання умови (2.40).

Процедури налаштування ваг (2.47), (2.48) «підтягують» вектор синаптичної ваги нейрона-переможця до вхідного вектору $x(k)$ на відстань, яка визначається параметром величини кроку пошуку η (рисунок 2.5а). У

випадку нормованого простору налаштування ваги виражається в його повороті у бік $x(k)$ (рисунок 2.5 б).

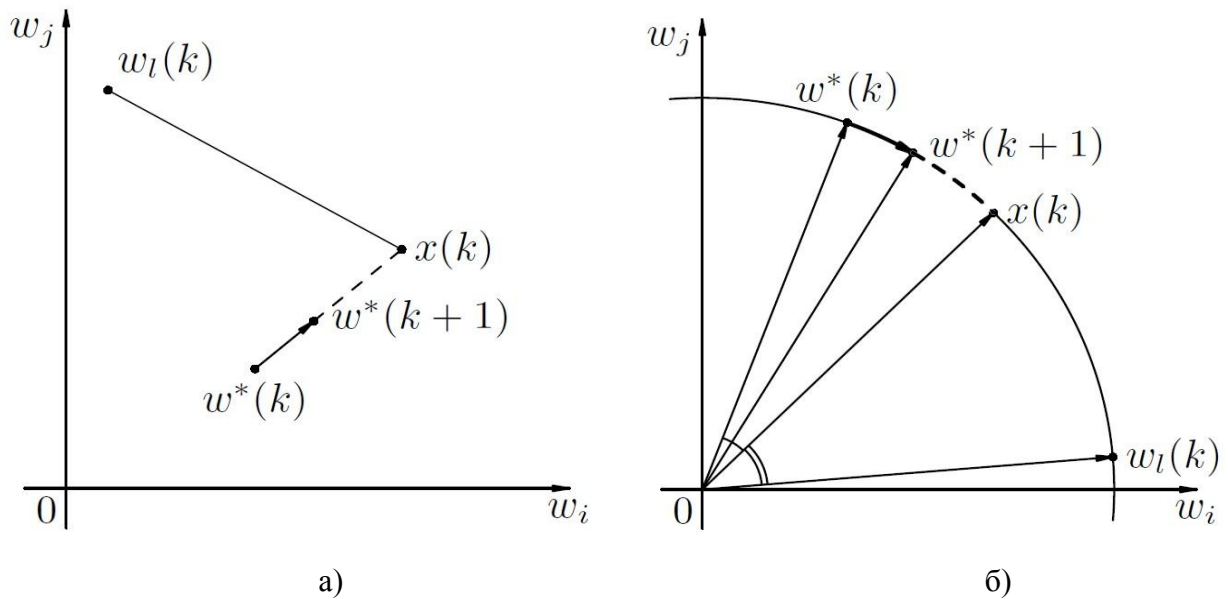


Рисунок 2.5 – Зміщення нейрона-переможця: а – у випадку ненормованого простору; б – у випадку нормованого простору

Оскільки процес навчання карти Кохонена відбувається значно швидше, ніж налаштування багат шарової мережі за допомогою зворотного поширення похибок [113], ці ШНМ найбільш ефективні для роботи у реальному часі, коли налаштування синаптичних ваг і обробка вхідних сигналів протікають паралельно.

2.2.2 Модифіковані нейромережі Кохонена

Самоорганізовані мапи Кохонена, спочатку призначені для зниження розмірності даних, застосовуються в задачах візуалізації, стиснення, моделювання і т.п. Особливістю SOM є використання крім матриці ваг $W = \{w_{ji}\}_{1 \leq j \leq N, 1 \leq i \leq n}$ додаткових неналаштованих векторних параметрів v_j , кожен з яких є відображення відповідного вузла (вектор-стовпця ваг матриці W) в простір зниженої розмірності. Відображення вузлів організовані в

регулярну решітку, топологія якої може бути різною, але найчастіше прямокутної або гексагональної, як зображено на рисунку 2.6.

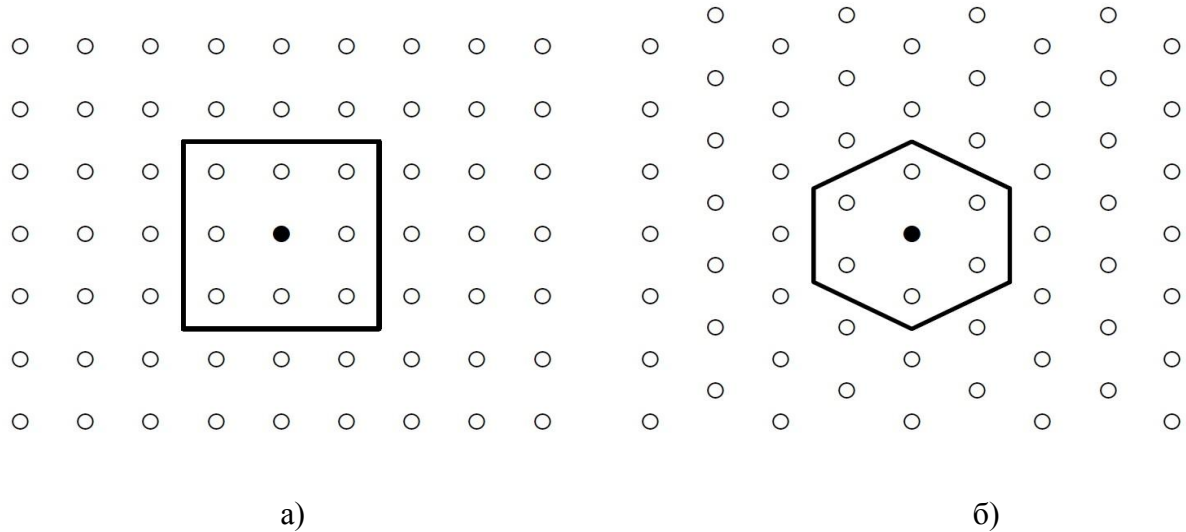


Рисунок 2.6 – Поширені види топології самоорганізовної мапи Кохонена: а – прямокутна; б – гексагональна

Використання такої топології дозволяє задати на множині v функцію сусідства $\varphi(j,l)$. Зазвичай $\varphi(j,l)$ – це ядерна (радіально-базисна) функція, симетрична відносно максимуму в точці $w^*(k)$ і спадаюча зі зростанням відстані $d(w^*, w_j)$. Найчастіше у якості функцій сусідства використовуються гавсиан, мексиканська шляпа, конусоподібна та інші дзвонкуваті функції, деякі з них показані на рисунку 2.7.

В процесі роботи мапа Кохонена реагує на отримання спостереження $x(k)$ наближаючи до нього вузол-переможець $w^*(k)$ відповідно до принципів конкурентного навчання. При цьому інші вузли $w_l, 1 \leq l \leq m, l \neq j$ наближаються до вузла-переможця w^* пропорційно ступеня сусідства між ними $\varphi(l, w^*)$.

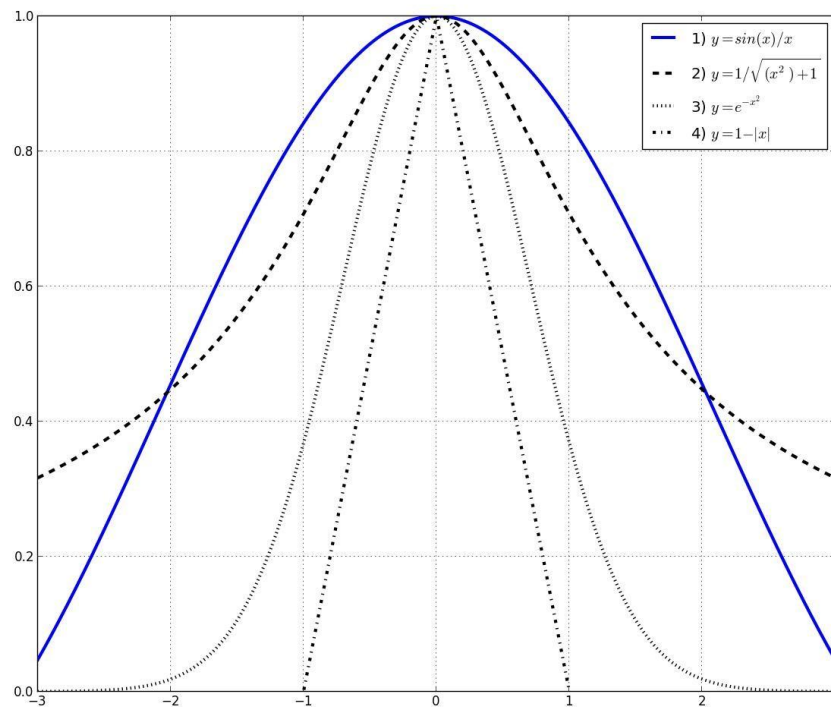


Рисунок 2.7 – Приклад дзвонуватих функцій: 1 – конусоподібна; 2 – мультіквадратична; 3 – гаусіан; 4 – трикутна

Графічно архітектура самоорганізовної мапи зображена на рисунку 2.8.

Для надходячих на вхід спостережень $x(k)$ визначається найближчий вузол $w^*(k)$ і наближається до надійшовшого спостереження за формулою

$$w^*(k+1) = w^*(k) + \eta(k) x(k) - w^*(k) , \quad (2.49)$$

де $\eta(k)$ задає коефіцієнт навчання (крок пошуку).

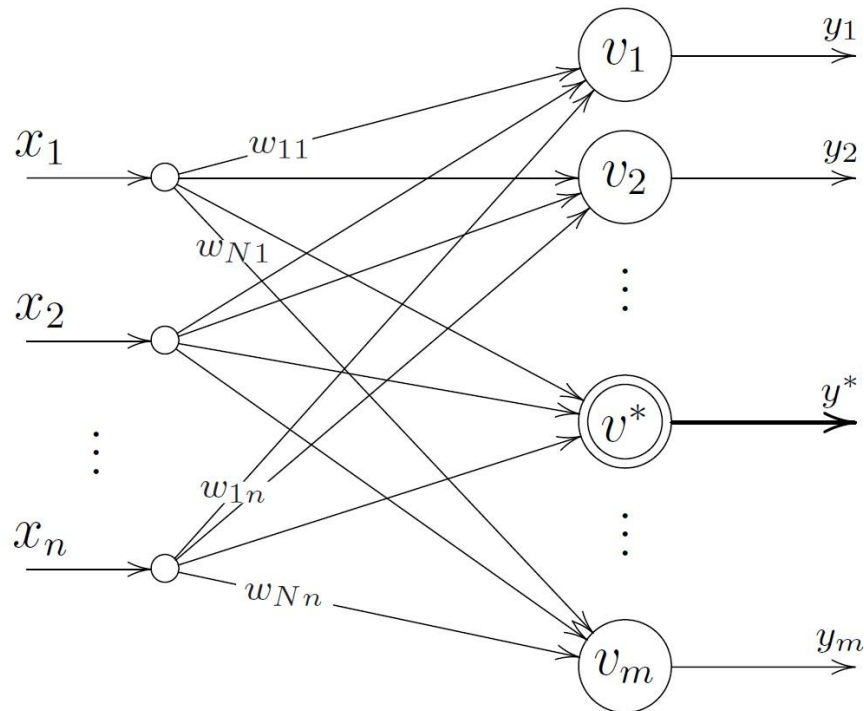


Рисунок 2.8 – Архітектура самоорганізовної мапи Кохонена

В процесі роботи SOM реагує на вхідне спостереження активацією відображення вузла-переможця $v^*(k)$. Таким чином, після виконання самоорганізації на деякій множині спостережень, близькі спостереження будуть, завдяки функції сусідства, активувати близькі відображення вузлів. Що стосується виконання задач нечіткої кластеризації, вихід $y_j(k) = 1$, є спостереження $x(k)$ яке належить j -му кластеру, але ясно, що і інші відображення вузлів можуть бути активовані пропорційно близькості до вузла-переможця, забезпечуючи таким чином нечітку кластеризацію.

Не складно побачити, що процедура (2.49) мінімізує критерій

$$E_l^K = \frac{1}{2} \sum_{p=1}^k \phi(j, l) \|x(p) - w_j\|^2 \quad (2.50)$$

Очевидним є схожість критерію (2.50) з основною цільовою функцією задачі кластеризації, а процедура його мінімізації (2.49),

відповідно, з процедурами адаптивної кластеризації Парка-Деггера [85], так само структурна подібність з процедурами розрахунку центроїдів в адаптивних методах.

Дана обставина дозволяє провести аналогії між параметрами самоорганізовних мап Кохонена і методами кластеризації та модифікувати мапи Кохонена для реалізації даних методів. Перші конструкції такого роду наведені в [114, 115] і інтерпретуються як нейро-фаззі мережі Кохонена, оскільки матриця ваг W інтерпретуються як шар конкуруючих нейронів, що обчислюють центроїди, а замість відображень v , які не навчаються, використовується другий шар нейронів, що обчислює нечіткі значення належності u , при цьому функція належності виступає аналогом функції сусідства $\varphi(l,j)$. Базова архітектура адаптивної кластеризуючої нейро-фаззі мережі Кохонена (*adaptive fuzzy Kohonen clustering network, AFKCN*) показана на рисунку 2.9.

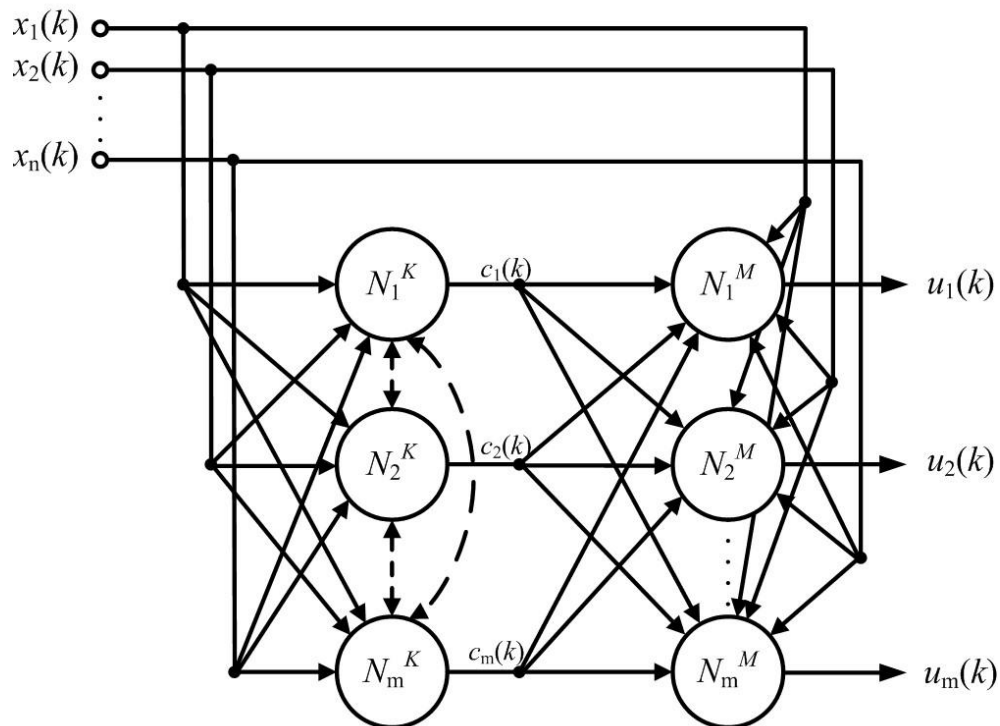


Рисунок 2.9 – Узагальнена архітектура адаптивної кластеризуючої нейро-фаззі мережі Кохонена (AFKCN)

Мережа з архітектурою, показана на рисунку 2.9, здатна реалізовувати як ймовірнісну, так і можливісну адаптивну кластеризацію. При цьому, як згадувалося раніше, численні джерела рекомендують використовувати результати ймовірнісної кластеризації для стабілізації процесу можливісної кластеризації. Використовуючи парадигму нейромереж Кохонена, легко сформулювати принцип паралельної можливісної і ймовірнісної кластеризації.

Нейрони шару Кохонена знаходяться в тому ж просторі, що й вхідні дані, і при вирішенні задачі кластеризації інтерпретуються як центроїди кластерів. Це означає, що кілька мереж, що обробляють одні й ті ж дані, здатні легко обмінюватися обчисленими центроїдами, тобто, по суті, мати об'єднаний перший шар нейронів. Аналітично, метод кластеризації комбінованої адаптивної нейро-фаззі мережі Кохонена можна представити таким чином:

$$\left\{ \begin{array}{l} c_j^{PR}(k+1) = c_j^{PS}(k) + \eta(k) (u_j^{PR}(k))^{\beta} (x(k+1) - c_j^{PS}(k)), \\ u_j^{PR}(k+1) = \frac{\|x(k+1) - c_j^{PR}(k+1)\|^{\frac{2}{1-\beta}}}{\sum_{l=1}^m \|x(k+1) - c_l^{PR}(k+1)\|^{\frac{2}{1-\beta}}}, \\ c_j^{PS}(k+1) = c_j^{PR}(k+1) + \eta(k) (u_j^{PS}(k))^{\beta} (x(k+1) - c_j^{PR}(k+1)), \\ u_j^{PS}(k+1) = \frac{1}{1 + \left(\frac{\|x(k) - c_j^{PS}(k+1)\|^2}{\mu_j(k)} \right)^{\frac{1}{1-\beta}}}, \\ \mu_j(k+1) = \frac{\sum_{p=1}^{k+1} (u_j^{PS}(p))^{\beta} \|x(p) - c_j^{PS}(k+1)\|^2}{\sum_{p=1}^{k+1} (u_j^{PS}(p))^{\beta}}, \end{array} \right. \quad (2.51)$$

де індекс PR значить розрахунок в рамках ймовірнісного (probabilistic) підходу;

індекс PS значить розрахунок в рамках можливісного (possibilistic) підходу;

$k = 1, 2, \dots, N$ – номер спостереження;

$u_j(k) \in [0, 1]$ – рівень належності вектору $x(k)$ к j -му кластеру;

c_j – центроїд (прототип) j -го кластеру;

β – ненегативний параметр фазифікації (фазифікатор), який визначає розмитість кордонів між кластерами;

m – число кластерів.

Графічно принципова схема методу (2.51) представлена на рисунку 2.10.

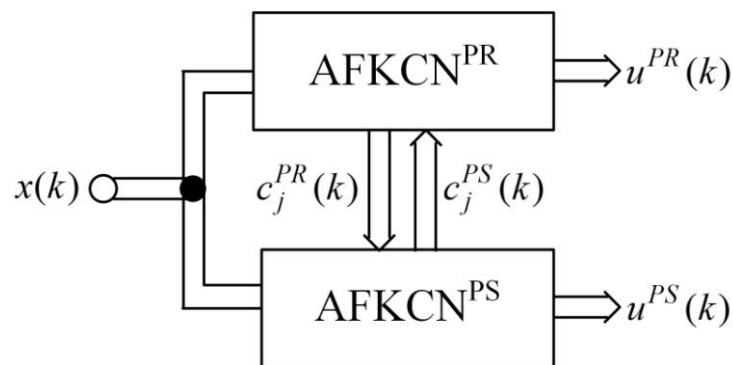


Рисунок 2.10 – Складова адаптивна нечітка кластеризуюча мережа Кохонена

Така схема відповідає багатомодельному підходу до вирішення задачі виявлення змін властивостей стохастичних послідовностей [116, 117], суть якого полягає в тому, що діагностуючий сигнал проходить через безліч моделей, кожна з яких заснована на своїй певній гіпотезі щодо характеру можливих змін. Якщо певні гіпотези дійсно виконуються, то сигнали оновлень на виході відповідних моделей повинні бути малими [118]. Таким чином, вирішальний механізм по суті заснований на знаходженні тієї моделі, на виході якої поновлення мінімальні, а ймовірність відповідної гіпотези максимальна.

2.2.3 Ансамблі та їх навчання

Якість рішення конкретної поставленої задачі (емуляції, прогнозування, розпізнавання образів, зворотного моделювання, управління тощо) може бути істотно підвищено за допомогою ансамблів (колективів, комітетів, сумішей, банків, груп) нейромереж [51, 116, 119, 120], в яких одні й ті ж дані паралельно обробляються декількома штучними нейронними мережами (ШНМ вихідні сигнали яких далі деяким чином комбінуються в об'єднану оцінку, переважаючи за якістю результати, одержувані за допомогою локальних мереж, що входять в ансамблі. Концептуальна схема нейромережевого ансамблю представлена на рисунку 2.11.

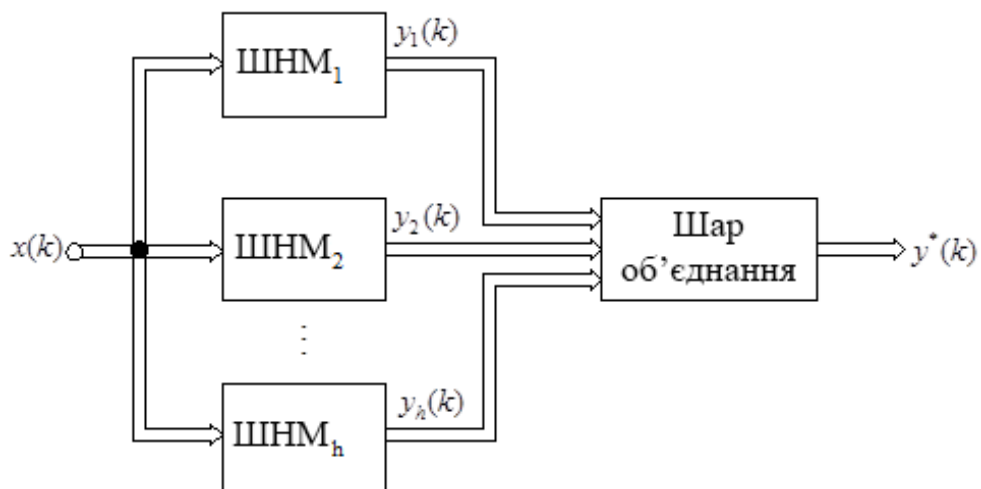


Рисунок 2.11 – Ансамбль нейронних мереж

Системи машинного навчання, засновані на ансамблях, мається на увазі паралельне навчання багатьох моделей для вирішення одного завдання. На відміну від звичайного підходу машинного навчання, замість того, щоб намагатися сформулювати за даними єдину найкращу гіпотезу,

ансамбль виробляє цілий набір гіпотез на основі різних моделей і далі комбінує їх в узагальнена рішення.

Використання нейронної метамережі дозволяє підвищити якість обробки інформації на основі ансамблю ШНМ і визначити мережі-переможці, найкращим чином пристосовані до вирішення розглянутої задачі і які вносять найбільший внесок у її рішення.

В оглядовій статті [121] наводяться три причини для застосування ансамблів: статистична, обчислювальна та репрезентативна. На підставі цього можна виділити три класи ситуацій, в яких застосування ансамблів доцільно:

1) Статистична причина відповідає ситуації, коли навчальний набір даних занадто малий для адекватного уявлення генеральної сукупності. У цьому випадку використання ансамблю еквівалентно узагальненню процедури бутстреппінга, тобто штучному розширенню вибірки шляхом виділення з неї випадкових непересічних підвбірок і додавання їх до початкового набору, як ніби це нові дані. В [122] показано, що в якості методу предобробки даних бутстреппінг ефективніше неінтелектуальних (не використовують знання про предметну область) методів інтерполяції даних. У разі використання ансамблю, ефективність бутстреппінга підвищується, якщо використовувати кожен підвбірку для навчання окремої моделі.

2) Обчислювальна причина асоціюється з проблемою вибору моделі, коли існує занадто багато моделей або їх варіацій, здатних вирішити задачу, і недостатньо відомостей, щоб визначити, яка з них буде оптимальною. Паралельний або послідовний перебір моделей обчислювально невиправданий, оскільки всі невдалі спроби будуть відкинуті без корисного ефекту. При використанні ансамблю кожна з паралельно працюючих моделей вносить свій внесок у рішення, що виправдовує обчислювальні витрати, а також і скорочує їх, оскільки

моделі ансамблю не вимагають настільки ретельного навчання, як індивідуально працюючі моделі.

3) Нарешті, використання властивості репрезентативності мається на увазі, коли жодна з наявних моделей не може належним чином відобразити складність або різноманітність даних. Краща гіпотеза може виявитися надто складною або ж лежати за межами області визначення окремої моделі. У подібних випадках доцільно використовувати ансамблі моделей, кожна з яких простіше підсумкової або визначена лише в частині діапазону вибірки (принцип «розділяй і володарюй»). Безліч рішень, отриманих такими моделями, дає при об'єднанні, гіпотезу краще будь-якої окремої моделі.

Аналогічним чином ансамблі доцільні в разі використання інформації з безлічі різних джерел (скажімо, результати різних обстежень одного пацієнта). Об'єднання множини даних в один вектор ознак може швидко привести до переускладнення моделі. У цьому випадку доцільно використовувати частини ансамблю для обробки окремих підмножин ознак (підпросторів даних), а потім поєднувати результати на більш високому рівні [123].

2.2.4 Основні поняття теорії систем колективного висновку

Впевненість. Якщо відразу кілька моделей, що входять в ансамбль, приходять до єдиного або схожого рішенням, кажуть про впевненість ансамблю в цьому рішенні. При оцінці впевненості розумно розглядати не частку моделей, що призвели до схожих рішень, а співвідношення часток для різних рішень. Так, скажімо, якщо половина ансамблю голосує за одне рішення, а половина за друге, корисно вважати, що впевненість ансамблю в обох рішеннях низька (попри високу частку голосуючих за кожне окреме рішення).

Важливо зауважити, що висока впевненість в загальному випадку не означає хороше рішення, а низька впевненість не означає погане. Тим не менш, показано [124], що рішення правильно навченого ансамблю зазвичай тим краще, чим вище впевненість в ньому. Використовуючи цей підхід, рішення ансамблю можуть бути використані для оцінки апостеріорних ймовірностей різних рішень.

Диверсифікація. Ступінь успішності роботи ансамблю (його здатність компенсувати внутрішні помилки) прямо залежить від ступеня різноманітності (диверсифікації) моделей які до нього входять. У граничному випадку, якщо серед моделей немає ніякого різноманіття і всі вони продукують однаковий вихід, корекція помилок неможлива. Таким чином, різні моделі, що входять в ансамбль, повинні робити різні помилки, щоб була можлива реалізація стратегії зменшення загальної помилки. У контексті кластеризації, диверсифікація має на увазі продукування різних розбиттів.

Диверсифікація може бути досягнута декількома шляхами.

По-перше, це використання різних навчальних вибірок для навчання різних частин ансамблю. Різноманітність вибірок часто забезпечується техніками ресемплірованія, такими як *bagging* і *bootstrapping*, при яких нові вибірки складаються як випадкові пересічні підвибірки вихідних даних [125].

Альтернативою або доповненням до методів ресемплірованія є використання різних підмножин ознак для навчання різних частин ансамблю. Використання випадкових підмножин ознак відомо як метод випадкових підпросторів [126], невідомі ж розділення зазвичай ґрунтуються на природі даних (при злитті даних з різних джерел і т.п.).

По-друге, для досягнення різноманітності рішень навіть на загальній вибірці при складанні ансамблю віддається перевага менш стабільним

моделям, які генерують істотно різне розбиття навіть при невеликій зміні параметрів навчання або ініціалізації [127].

По-третє, різноманітність досягається за рахунок включення в ансамбль різних по будові моделей (у разі використання нейромереж - різних архітектур і / або алгоритмів навчання).

Інкрементне навчання ансамблів. Інкрементне навчання мається на увазі можливість методу навчатися на нових даних, отриманих після того, як модель вже була згенерована. Навчання називається інкрементним, якщо на послідовності навчальних вибірок (або окремих спостережень) воно продукує послідовність моделей, кожна наступна з яких залежить тільки від попередньої моделі і нових даних. Для збереження інформації про попередні етапи навчання, в системах, заснованих на ансамблях, практикується підключення додаткових моделей по мірі надходження даних та їх навчання тільки на нових даних, що робить систему в цілому інкрементною навіть при використанні пакетних алгоритмів.

2.2.5 Методи побудови ансамблів

Базова ідея ансамблів передбачає, що всі вхідні в ансамбль моделі працюють незалежно і їх результати використовуються тільки для формування спільного рішення. Таким чином, з розгляду виключаються багаторівневі, ієрархічні, каскадні і т.п. системи, принципи побудови яких можуть бути вельми варіативні.

При розгляді лише класичних ансамблів, вся варіативність їх побудови полягає тільки в способі комбінування результатів роботи які входять в ансамбль моделей для остаточного висновку.

Очевидною і вичерпною є ієрархія способів комбінування рішень, впорядкована за зростанням складності:

Алгебраїчна комбінація. Алгебраїчні методи виведення колективного рішення є найпростішими, ненавченими методами комбінування відповідей. Вони мають на увазі застосування до рішень, запропонованим моделями які входять до ансамблю, простих алгебраїчних операторів, таких як обчислення середнього, медіани, мінімуму або максимуму, суми, твори і т.д.

У контексті кластеризації та класифікації ці методи часто називаються «просте голосування», тому що вони можуть бути зведені до вибору мітки для кожного спостереження тим чи іншим методом голосування які входять в ансамбль моделей. Незважаючи на поширеність цього поняття, воно коректно тільки у випадках, коли висновком моделей є дискретні і впорядковані мітки класів / кластерів. Оскільки всі розглянуті в роботі системи використовують, по-перше, нечіткі і невпорядковані заходи належності, а, по-друге, продукують, крім них, центроїди кластерів, надалі це поняття використовуватися не буде.

Алгебраїчна комбінація гарантує тільки те, що найгірший результат, отриманий всередині ансамблю, не використовуватиметься в якості спільного висновку. Це найменший ступінь корекції помилок, на яку здатний ансамбль, так званий «краще гіршого». Незаперечною перевагою цього способу висновку є те, що для його використання не потрібна ніяка додаткова інформація про задачу або системи - ні навчальний сигнал, ні яка-небудь ще оцінка якості. Це робить його доцільним, а іноді і безальтернативним при вирішенні завдань навчання без вчителя і розвідувальному аналізу даних.

Зважена комбінація. Очевидним розвитком нездібної алгебраїчної комбінації є комбінація з налаштованим вагами. У цьому випадку вихід кожної моделі бере участь у підсумковому рішенні з деяким коефіцієнтом, який може налаштовуватися в ході навчання або інкрементно. Процедура настройки коефіцієнтів тривіальна, якщо є навчальний сигнал. У разі

навчання без вчителя вибір критеріїв навчання вихідних ваг може бути заснований лише на непрямих оцінках якості роботи моделей, що часто вимагає індивідуального підходу і знання предметної області.

Правильно навчена зважена комбінація дозволяє отримати на виході ансамблю результат кращий, ніж результат будь-якої з вхідних в ансамбль моделей («краще кращого») [128] .

Більш складні системи колективного виводу. Для більш тонкої настройки залежності загального висновку від висновків які входять в ансамбль моделей використовуються більш складні системи ваг. Найбільш поширеними нелінійними системами колективного висновку є нейромережі (багатошаровий персептрон, радіально-базисні мережі, мапи Кохонена.

Навіть за наявності навчального сигналу існують різні методики оцінки якості, не кажучи вже про ситуації, коли успішність вхідних в ансамбль моделей треба оцінювати побічно. Якщо немає твердої підстави для вибору однієї з моделей / архітектур колективного виведення, найбільш природним є об'єднання декількох моделей в новий ансамбль. Доцільність і правила побудови такого надбудованого ансамблю залишаються тими ж, що і для однорівневого ансамблю, а розвиток концепції багаторівневих ансамблів призводить до появи численних погано формалізуючих систем колективного виведення, таких як каскадні, фрактальні і т.п., загальна теорія яких тільки починає формуватися [129, 130-132]

2.2.6 Ансамблі нейро-фаззі мереж Кохонена

За структурою методу (2.51) видно, що результатом кластеризації при подібному підході є дві матриці розбиття - за версією ймовірнісного і можливісного методів. Зазвичай пропонується приймати в якості

остаточної відповіді можливісне розбиття як більш універсальне, однак повне ігнорування результатів ймовірнісної кластеризації - явно нераціональне використання обчислювальних ресурсів. Використовуючи техніки колективного виведення, описані в вище, можна спробувати поліпшити загальний результат за рахунок використання альтернативного розбиття U^{PR} для поліпшення загального результату.

Найпростіший варіант колективного виведення - усереднення U^{PR} і U^{PS} - не буде працювати коректно через примусове нормування U^{PR} на одиницю. Застосування більш складних моделей виведення вимагає якоїсь оцінки якості елементів антецедента.

Існує ряд індексів оцінки якості кластеризації, найпопулярніші з яких - індекси Данна [133], Ксі-Бені [104], Фукуяма-Сугено [134], Девіса-Болдуїна [135] та інші. Усі вони, однак, засновані на оцінці загальних характеристик розбиття, а значить принципово незастосовні для підтримки адаптивного навчання, коли оцінювати якість необхідно згідно з частковим результатом.

Найпростішим індексом з доведеною ефективністю можна вважати так званий коефіцієнт розбиття (*partition coefficient*), обговорюваний [56,136–138]. Він являє собою просте середнє квадратів належностей всіх спостережень до кластеру, і в пакетному вигляді записується простим виразом

$$Q_U = \frac{1}{N} \sum_{j=1}^m \sum_{k=1}^N u_j^2(k) \quad (3.15)$$

Коефіцієнт має ясний фізичний зміст: чим краще виражені кластери, тим більше буде коефіцієнт розбиття, а його мінімум $Q \rightarrow \frac{1}{m}$ досягається, якщо дані належать усім кластерам рівномірно (що є очевидно марним

рішенням). У цього коефіцієнта є деякі недоліки, в основному стосуються занадто високої толерантності до викидів (їх наявність не надто впливає на оцінку якості, хоча зазвичай вважається бажаним її в таких випадках знижувати), однак для пропонованого застосування найважливішим є те, що на відміну від більш складних оцінок, цей індекс дуже просто отримати в ітеративній формі, використовуючи формулу рекурентного розрахунку середнього:

$$Q_{u_j}(k) = Q_{u_j}(k-1) + \frac{1}{k} (u_j^2(k) - Q_{u_j}(k-1)). \quad (3.16)$$

Такий вигляд дозволяє тривіально отримати колективну відповідь як зважене середнє:

$$u_j(k) = \frac{Q_{u_j^{PS}}(k)u_j^{PS}(k) + Q_{u_j^{PR}}(k)u_j^{PR}(k)}{u_j^{PS}(k) + u_j^{PR}(k)}, \quad (3.17)$$

причому поки Q_{u_j} рахується тільки для U^{PR} , коефіцієнт завжди буде менше або дорівнюватися одиниці, що відповідає початкової концепції, по якій імовірнісний метод є допоміжним по відношенню до можливісного.

Використовуючи процедуру (2.51) в буквальному вигляді, ми знову стикаємося з проблемою експоненціальної ваги β , яка до того ж значно погіршується тим, що така форма фаззифікатора безпосередньо впливає на оцінку Q_U . Великі значення фаззифікатора β безпосередньо збільшують розмитість меж кластерів, яку і оцінює коефіцієнт розбиття.

Способи вирішення цієї проблеми вже були розглянуті вище і можуть бути легко застосовані до процедури (2.51) заміною параметрів з індексом PR відповідними параметрами з процедури (2.13), а параметри з індексами PS – параметрами з процедури (2.18). Окрім цього, на відміну

від параметра β , який зазвичай має тільки одне оптимальне значення для конкретного набору даних, вплив лінійного фаззифікатора α більш м'який і змінює скоріше характер розбиття, ніж його якість, що дозволяє варіювати його в широких межах. Як вже було згадано, фаззифікатор α і параметр подавлення σ мають чіткий фізичний сенс і впливають на компактність і поділ кластерів. Кожне їх поєднання відповідає якомусь апріорному припущенню про властивості вибірки, а в умовах, коли характеристики вибірки змінюються з часом, повинні змінюватися і ці параметри. Це призводить до схеми модифікованої адаптивної нейро-фаззі мережі Кохонена, що має макропараметри фаззифікації і подавлення, як показано на рисунку 2.12.

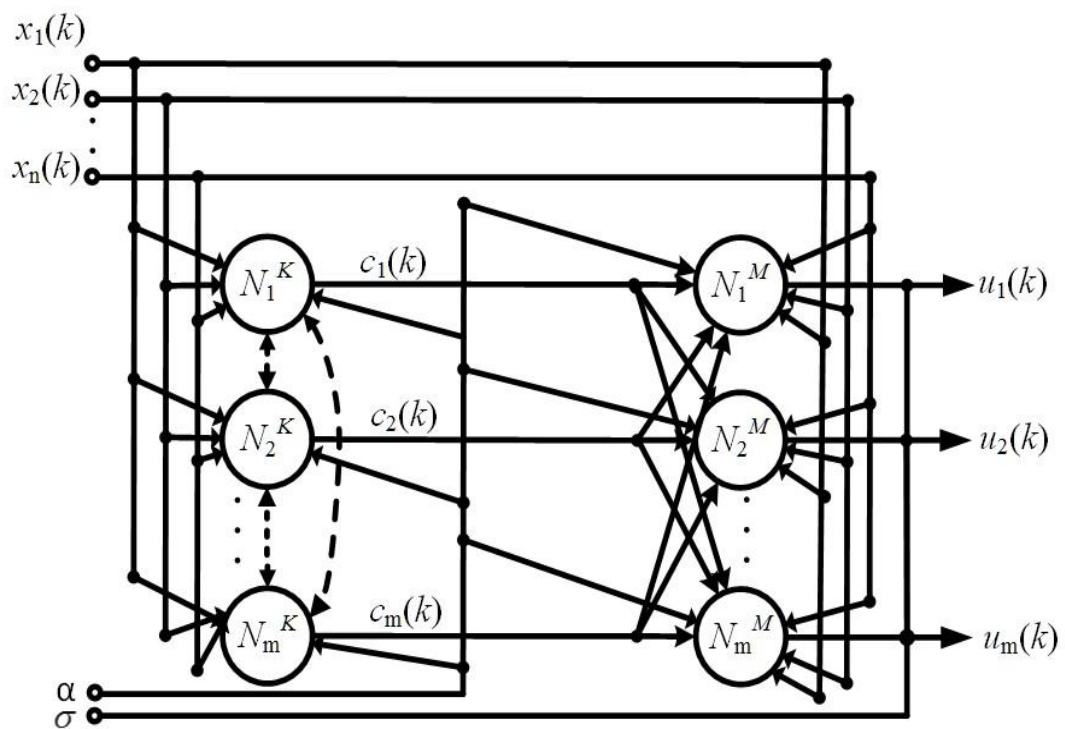


Рисунок 2.12 – Адаптивна нейро-фаззі мережа Кохонена зі змінним фаззифікатором

2.2.7 Нечітка кластеризація типу-2 з використанням ансамблів модифікованих нейро-фаззі мереж Кохонена

Використання в одній системі кластеризації нечіткої функції належності і нечіткої функції якості цієї належності приводить нас до концепції кластеризації, заснованої на нечіткій логіці типу-2.

При цьому застосування двох мереж, як на схемі 2.10, використовуючих загальні центроїди, які відрізняються значеннями фазифікації, призводить до характерного результату - кластерам з нечітким кордоном, що є нічим іншим, як результатом роботи системи на інтервальній нечіткій логіці типу-2. У цьому випадку операція (2.54) є простим варіантом процедури редукції моделі, причому виконуваної ітеративно.

Немає жодних обмежень на те, щоб використовувати в паралельній роботі більшу кількість адаптивних нейро-фаззі мереж Кохонена з різними значеннями фазифікатора. Беручи до уваги той факт, що більше значення α завжди призводить до більшого розмиття кордонів, ми можемо стверджувати, що результати роботи AFKCN з різними фазифікаторами є зрізами по порозу α деякої неявно заданої вторинної функції належності.

Редукція центроїдів. Робота в ансамблі вимагає диверсифікації вхідних до нього мереж за параметрами i , відповідно, за результатами роботи. Ця обставина, а також те, що при паралельній роботі великої кількості мереж обмін інформацією про центроїди за типом процедури (2.51) для кожної їх пари стає обчислювально громіздким, призводить до необхідності працювати з великим числом центроїдів, майже незалежно породженими різними мережами.

Просте обчислення положення результуючого центроїда кластера як усередненого положення всіх центроїдів, розрахованих для даного кластера, при найближчому розгляді виявляється принципово неможливим

через непорівнянності різних розбиттів. Навіть при рівній кількості кластерів (i , відповідно, центроїдів) у кожній мережі їх асоціація між собою стає необґрунтована або неможливою вже при помірній диверсифікації параметрів або нестационарності даних.

Це призводить до необхідності ввести для центроїдів процедуру редукції, аналогічну редукції функції належності (2.54) і в цілому є частиною процедури редукції моделі, необхідної будь-якій системі нечіткої логіки типу-2.

Виходячи з того, що у разі об'єктивної наявності в даних скупчень, будь-яка процедура кластеризації, породжена цільовою функцією (2.22) або родинними, буде прагнути помістити центроїди в ці скупчення [139], було припущено, що безліч центроїдів, розраховане множиною паралельно працюючими над одними і тими ж даними нейромережами Кохонена, буде прагнути до скупчень даних незалежно від параметрів настроювання і умов ініціалізації. У цьому випадку редукція центроїдів може бути зведена до виділення кластерів серед усього безлічі центроїдів, розрахованих ансамблем, і усереднення цих скупчень згідно з правилом колективного виводу. Найпростіше рішення, яке задовольняє цьому формулюванні - провести процедуру кластеризації над вибіркою, що складається виключно з центроїдів, з тією лише різницею, що головним результатом цієї кластеризації буде не матриця розбиття, а отримані в центроїди.

Логічним буде використовувати для цього адаптивну нейромережу Кохонена, аналогічну наведеним раніше. Загальна архітектура ансамблю, що реалізує повну процедуру адаптивної нечіткої кластеризації типу-2, наведена на малюнку 2.13.

Така архітектура може включати довільне число паралельних нейромереж Кохонена, що здійснюють кластеризацію будь-якими методами, розглянутими вище. Отримані цими методами центроїдами у

свою чергу кластеризуються вихідною мапою Кохонена, щоб знайти стійкі скупчення, а функції належності зважено підсумовуються в рамках підходу колективного висновку за узагальненим виразом (2.54).

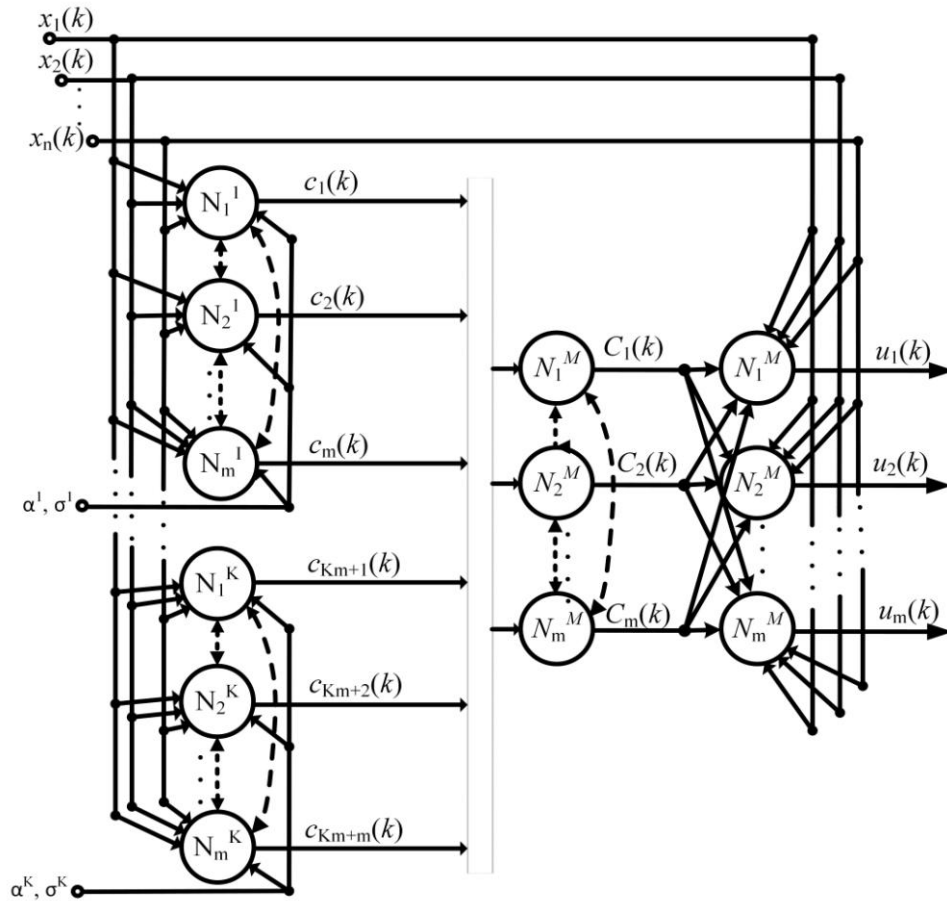


Рисунок 2.13 – Архітектура ансамблю нейро-фаззі мереж Кохонена для нечіткої кластеризації типу-2

$$u_j(k) = \frac{\sum_{t=1}^K Q^t(k) u_j^t(k)}{\sum_{t=1}^K u_j^t(k)} \quad (2.55)$$

Здійснювана таким чином рекурентная нечітка кластеризація типу-2 дозволяє отримувати адекватні результати кластеризації після надходження кожного нового спостереження навіть в умовах невизначеності і мінливості вхідних даних. Система в on-line режимі пристосовується до змін характеру поступаючих даних, а в залежності від

вибору початкового поєднання кластеризаторів для побудови ансамблю може бути спочатку підготовлена до тих чи інших основних видів невизначеності (різного ступеня забрудненості викидами, різниці в характерних масштабах кластерів, їх числа і формі і т.д.) без шкоди для якості кластеризації в стаціонарних умовах.

Додаткові оцінки, обчислювані в рамках використання нечіткої логіки типу-2, найбільш часто інтерпретуються як показники якості або «впевненості» в оцінці типу-1. Зазвичай вважається достатнім виправданням доцільності таких систем, що дана інформація допомагає в ухваленні рішення оператором системи, проте в поєднанні з адаптивним підходом стає можливо абсолютно по-новому розкрити потужність і гнучкість нечітких систем типу-2. Адаптивна система, здатна до «самокритики» на підставі таких мета-характеристик рішень, отримує широкі можливості пристосовуватися до вхідного потоку даних на відразу багатьох рівнях з гнучкістю, недоступною раніше.

Таким чином вперше стало можливо здійснювати нечітку кластеризацію типу 2 в online режимі, що відкриває широкі можливості застосування такого підходу в системах управління, ідентифікації, асоціативної пам'яті та ін.

З НЕЙРО-ФАЗЗИ СИСТЕМИ КЛАСТЕРИЗАЦІЇ ТА КЛАСИФІКАЦІЇ ДАНИХ У ON-LINE РЕЖИМІ НА ОСНОВІ СПЕЦІАЛІЗОВАНИХ АРХІТЕКТУР ТА ПРАВИЛ НАВЧАННЯ

Цей розділ присвячено синтезу спеціалізованих архітектур та відповідних методів навчання/самонавчання, що здатні у режимі реального часу вирішувати задачі класифікації та кластеризації потоку даних. При цьому підхід, що розвивається базується як на концепції традиційних штучних нейронних мереж, так і на більш «просунутих» підходах і, перш за все, нейро-фаззи-системи, що поєднують в собі здатності до навчання та апроксимації штучних нейронних мереж, так і прозорості та інотерпретуємості систем нечіткого виведення.

3.1 Системи на основі ядерних функцій активації-належності

Найбільш популярними серед існуючих штучних нейронних мереж (ШНМ) є радіально-базисні нейронні мережі (Radial Basis Function Neural Networks - RBFN). Основні ідеї радіально-базисних нейронних мереж пов'язані з методом потенційних функцій [140], оцінками Парзена [141, 142], ядерною [51] і непараметричною [143] регресіями. Важливою особливістю цих мереж є те, що вони володіють універсальними апроксимуючими властивостями і здатні навчатися в реальному часі.

Основним недоліком радіально-базисної мережі є її схильність до «прокльону розмірності», що породжує експоненційне зростання кількості радіально-базисних нейронів (R-нейронів) із зростанням розмірності вхідного простору, у зв'язку з чим, пропонується підхід до навчання RBFN з обмеженням кількості нейронів в on-line режимі обробки інформації.

На рис. 3.1 наведена стандартна архітектура радіально-базисної мережі, прихований шар якої реалізує деяке нелінійне перетворення простору входів

R^n у прихований простір R^h більш високої розмірності $h > n$, а вихідний шар, утворений адаптивним лінійним асоціатором, формує відгук мережі, здійснюючи нелінійне перетворення виду

$$\mathcal{F}^R(k) = w_0 + \sum_{l=1}^h w_l \varphi_l(x(k)) = \sum_{l=0}^h w_l \varphi_l(x(k)) = w^T \tilde{\varphi}(x(k)),$$

де $x(k) = [x_1(k), x_2(k), \dots, x_n(k)]^T$, $\varphi_l(x(k)) = \varphi_l(\|x(k) - c_l\|)$, σ_l – радіально-базисні функції, залежні від відстані $\|x(k) - c_l\|$ між вектором входів $x(k)$ і центром функції c_l і параметром ширини σ_l , k – поточний дискретний час, $\tilde{\varphi}(x(k)) = [1, \varphi_1^T(x(k)), \varphi_2(x(k)), \dots, \varphi_h(x(k))]^T$.

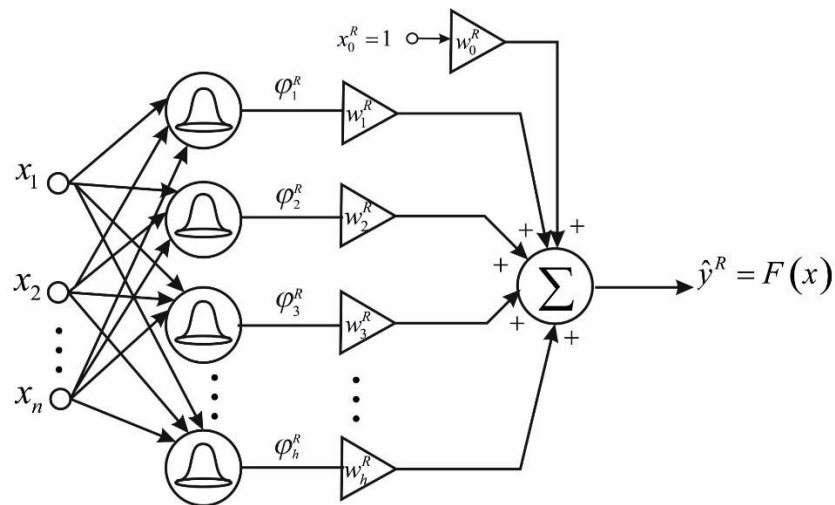


Рисунок 3.1 – Радіально-базисна нейронна мережа (Radial Basis Function Neural Networks - RBFN)

В якості активаційних функцій радіально-базисних ШНМ найчастіше використовується стандартний Гауссіан

$$y_l(x, k) = \exp\left(-\frac{\|x, k - c_l\|^2}{\sigma_l^2}\right), \quad l = 1, 2, \dots, h, \quad (3.1)$$

при цьому параметри центрів c_l і ширин σ_l , як правило, задаються заздалегідь і не налаштовуються в процесі навчання. Саме ж навчання зводиться до налаштування вектора синаптичних ваг $w = w_0, w_1, \dots, w_h^T$, для чого зазвичай використовуються різні модифікації методу найменших квадратів.

Покращити апроксимуючі властивості мережі можна, використовуючи замість гауссіана (3.1) багатовимірну конструкцію

$$y_l(x, k) = \exp\left[-(x, k - c_l)^T \Sigma_l^{-1} (x, k - c_l)\right] = \exp\left[-\|x, k - c_l\|_{\Sigma_l^{-1}}^2\right], \quad (3.2)$$

де коваріаційна матриця Σ_l визначає форму, розмір і орієнтацію рецепторного поля 1-ї радіально-базисної функції.

При $\Sigma_l = \sigma_l^2 I$ (тут $I - n \times n$ – одинична матриця) рецепторне поле являє собою гіперсферу з центром c_l і радіусом σ_l ; при $\Sigma_l = \text{diag } \sigma_1^2, \sigma_2^2, \dots, \sigma_n^2$ – це гіперелліпсоїд, чий осі збігаються з осями вхідного простору і мають довжину σ_i по i -ї осі, і, нарешті, при Σ_l – довільній додатно визначеній матриці

$$\Sigma_l = Q_l^T \Lambda_l Q_l$$

діагональна матриця власних значень Λ_l визначає розмір рецепторного поля, а ортогональна матриця обертання Q_l – його орієнтацію.

Кажучи про навчання радіально-базисної ШНМ, слід зауважити, що налаштуватися можуть не тільки вектор синаптичних ваг w , але й центри c_l і матриці Σ_l . Так, вводячи перетворення, реалізоване нейронною мережею, у вигляді

$$\mathcal{E}^R k = \sum_{l=0}^h w_l \varphi_l \left\| x k - c_l \right\|_{\Sigma_l^{-1}}^2 ,$$

критерій навчання

$$E k = \frac{1}{2} e^2 k = \frac{1}{2} \left(y k - \sum_{l=0}^h w_l \varphi_l \left\| x k - c_l \right\|_{\Sigma_l^{-1}}^2 \right)^2$$

(тут $y k$ – зовнішній навчальний сигнал) і похідні за всіма параметрами, що підлягають налаштуванню

$$\left\{ \begin{array}{l} \frac{\partial E k}{\partial w_l} = -e k \varphi_l \left\| x k - c_l \right\|_{\Sigma_l^{-1}}^2 , \\ \nabla_{c_l} E k = 2e k w_l \varphi_l' \left\| x k - c_l \right\|_{\Sigma_l^{-1}}^2 \Sigma_l^{-1} x k - c_l , \\ \left\{ \frac{\partial E k}{\partial \Sigma_l^{-1}} \right\} = -e k w_l \varphi_l' \left\| x k - c_l \right\|_{\Sigma_l^{-1}}^2 x k - c_l x k - c_l^T , \end{array} \right. \quad (3.3)$$

можна записати алгоритм навчання [144]:

$$\left\{ \begin{array}{l} w_l k+1 = w_l k + \eta_w k+1 e k+1 \varphi_l \|x k+1 - c_l k\|_{\Sigma_l^{-1} k}^2, \\ c_l k+1 = c_l k - \eta_c k+1 e k+1 w_l k+1 \varphi_l' \|x k+1 - c_l k\|_{\Sigma_l^{-1} k}^2 \times \\ \times \Sigma_l^{-1} k x k+1 - c_l k, \\ \Sigma_l^{-1} k+1 = \Sigma_l^{-1} k + \eta_\Sigma k+1 e k w_l k+1 \varphi_l' \|x k+1 - c_l k+1\|_{\Sigma_l^{-1} k}^2 \times \\ \times x k+1 - c_l k+1 x k+1 - c_l k+1^T, \end{array} \right. \quad (3.4)$$

де $\eta_w k+1, \eta_c k+1, \eta_\Sigma k+1$ – параметри кроку навчання для відповідних змінних.

Використання в якості активаційних функцій гауссіанів (3.1) і (3.2) призводить до того, що процедура навчання (4) стає занадто громіздкою з обчислювальної точки зору, що природно уповільнює швидкість навчання. У зв'язку з цим пропонується ввести в розгляд багатовимірну модифікацію функції В. Єпанечнікова [145] у вигляді

$$\varphi_l \|x k+1 - c_l k\|_{\Sigma_l^{-1}}^2 = 1 - \|x k+1 - c_l k\|_{\Sigma_l^{-1}}^2,$$

чії похідні мають форму

$$\left\{ \begin{array}{l} \nabla_{c_l} \varphi_l \|x k - c_l\|_{\Sigma_l^{-1}}^2 = -2 \Sigma_l^{-1} x k - c_l, \\ \left\{ \frac{\partial \varphi_l \|x k - c_l\|_{\Sigma_l^{-1}}^2}{\partial \Sigma_l^{-1}} \right\} = x k - c_l x k - c_l^T. \end{array} \right. \quad (3.5)$$

Співвідношення (3.5) дозволяють переписати систему (3.3) у вигляді

$$\begin{cases} \frac{\partial E^k}{\partial w_l} = -e^k \left(1 - \|x^k - c_l\|_{\Sigma_l^{-1}}^2 \right) \\ \nabla_c E^k = 2e^k w_l \Sigma_l^{-1} (x^k - c_l) , \\ \left\{ \frac{\partial E^k}{\partial \Sigma_l^{-1}} \right\} = -e^k w_l (x^k - c_l) (x^k - c_l)^T , \end{cases}$$

а алгоритм навчання –

$$\begin{cases} w_l^{k+1} = w_l^k + \eta_w^{k+1} e^{k+1} \left(1 - \|x^{k+1} - c_l^k\|_{\Sigma_l^{-1}^k}^2 \right) , \\ c_l^{k+1} = c_l^k - \eta_c^{k+1} e^{k+1} w_l^{k+1} \Sigma_l^{-1} (x^{k+1} - c_l^k) , \\ \Sigma_l^{-1}{}^{k+1} = \Sigma_l^{-1}{}^k + \eta_\Sigma^{k+1} e^{k+1} w_l^{k+1} \times \\ \times (x^{k+1} - c_l^{k+1}) (x^{k+1} - c_l^{k+1})^T . \end{cases} \quad (3.6)$$

Нескладно помітити, що процедура (6) з обчислювальної точки зору істотно простіша за алгоритм (4).

Питання вибору кількості нейронів в мережі n і початкового розташування центрів c_l є надзвичайно актуальним. Найпростішим варіантом вирішення цієї проблеми є використання алгоритму Subtractive clustering [146], який досить ефективний при роботі в пакетному режимі, але при цьому вимагає вибору цілого набору вільних параметрів. Якщо розв'язувана задача пов'язана з обробкою нестационарних процесів, то необхідно час від часу переналаштовувати мережу.

Dynamic Decay Adjustment (DDA), теж є одним з можливих методів навчання радіально-базисних нейронних мереж [147]. Він відноситься до алгоритмів конструктивного навчання і працює досить швидко. Однак, при роботі в on-line режимі при обробці нестационарних сигналів цей метод стає неефективним.

Resource Allocation Network [148] використовує комбіноване навчання, базується як на оптимізації, так і на пам'яті (принцип «нейрони в точках

даних)), з використанням елементів конкуренції. При цьому в процесі навчання за допомогою градієнтних процедур налаштовуються як синаптичні ваги, так і параметри центрів нейронів, найближчих до спостереження яке тільки що надійшло на обробку. Можна помітити, що в якості активаційних, в цій мережі замість традиційних гауссіанів використовуються стандартні функції Єпанечнікова. Недоліком Resource Allocation Network є висока обчислювальна складність.

У зв'язку з цим є доцільною розробка штучної радіально-базисної нейронної мережі, яка сама налаштовує не тільки всі свої параметри, а й визначає автоматично кількість нейронів в on-line режимі з високою швидкістю надходження і обробки даних.

На рис. 3.2 наведена структурна схема гібридної еволюційної штучної нейронної мережі, основу якої складає радіально-базисна нейронна мережа з перемінною кількістю нейронів і самоорганізовна мапа Т. Кохонена (SOM) [80], яка управляє їх кількістю і підлаштовує розташування центрів в режимі самонавчання.

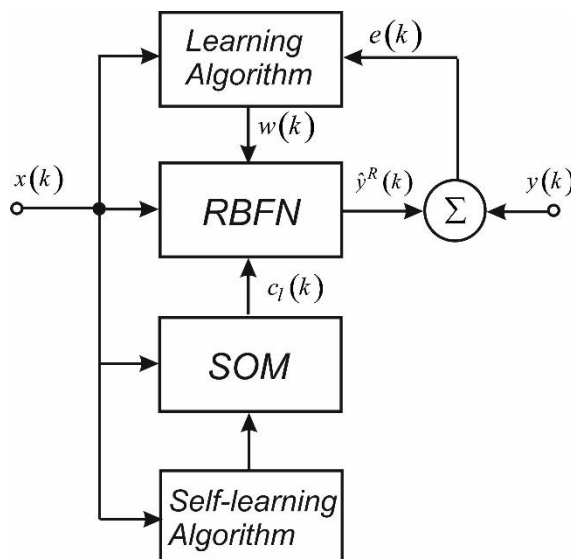


Рисунок 3.2 – Структурна схема гібридної еволюційної мережі

Процес функціонування цієї системи відбувається наступним чином. При надходженні першого спостереження x_1 воно подається на вхід радіально-базисної мережі, де формується перший нейрон за принципом «нейрони в точках даних», тобто практично миттєво. При подальшому надходженні даних вони спочатку поступають в SOM, де відбувається порівняння з вже існуючими центроїдами, а потім, якщо збігів не виявилось, формується новий центр радіально-базисної функції і відповідно і новий нейрон в RBFN.

В рамках цього підходу введемо в розгляд наступний метод управління кількістю нейронів у мережі:

Крок 1₁: закодувати всі значення вхідних змінних в інтервал $-1 \leq x_i \leq 1$ та задати радіус рецепторного поля функції сусідства в інтервалі $r \leq 0,33$;

Крок 2₁: при надходженні спостереження x_1 задати $c_1 = x_1$;

Крок 3₁: при надходженні спостереження x_2 :

- якщо $\|x_2 - c_1\| \leq r$, тоді c_1 коректується за правилом

-

$$c_1 = \frac{c_1 + x_2}{2};$$

- якщо $r < \|x_2 - c_1\| \leq 2r$, c_1 коректується згідно до правила самонавчання самоорганізовної мапи Кохонена за принципом «переможець отримає більше» (WTM) [11]

-

$$c_1 = c_1 + \eta \psi_1(x_2 - c_1)$$

з функцією сусідства

$$\psi_1(x) = \max \left\{ 0, 1 - \left(\frac{\|x - c_1\|}{2r} \right)^2 \right\}$$

(функція Єпанечнікова з рецепторним полем радіусом $2r$)

- якщо $\|x - c_1\| > 2r$, формується нова радіально-базисна функція з центром $c_2 = x$.

На цьому перша ітерація формування активаційних функцій радіально-базисної нейронної мережі закінчується. Нехай до k -го моменту часу сформовано $p \leq h$ активаційних функцій $\varphi_l(x)$ з центрами c_l і на обробку надійшло спостереження x^{k+1} . Далі формування радіально-базисних функцій проводиться таким чином:

Крок 1_{k+1}: визначити нейрон-переможець, для якого відстань $\|x^{k+1} - c_l\|$ мінімальна серед всіх $l = 1, 2, \dots, p$;

Крок 2_{k+1}:

- якщо $\|x^{k+1} - c_l\| \leq r$, тоді

$$c_l^{k+1} = \frac{c_l^k + x^{k+1}}{2};$$

- якщо $r < \|x^{k+1} - c_l\| \leq 2r$, тоді

$$c_l^{k+1} = c_l^k + \eta^{k+1} \psi_l(x^{k+1}) (x^{k+1} - c_l^k);$$

$$\psi_l^{k+1} = \max \left\{ 0, 1 - \left(\frac{\|x^{k+1} - c_l^k\|}{2r} \right)^2 \right\};$$

- якщо $\|x_{k+1} - c_l^k\| > 2r$, тоді формується радіально-базисна функція з центром $c_{p+1}^{k+1} = x_{k+1}$;

- якщо ж у процесі формування радіально-базисних функцій виникає ситуація $\|x_{k+1} - c_l^k\| > 2r$, а $p = h$, то необхідно збільшити радіус рецепторного поля і повернутися до кроку 2_{k+1} зі збільшеним радіусом функції ψ_l^{k+1} .

Усі наступні кроки реалізуються аналогічно [149, 150].

Як видно дана процедура є гібрид еволюційного алгоритму Н. Касабова [81] і самоорганізовної мапи Т. Кохонена. Однак запропонована нейронна мережа розроблена не тільки для вирішення завдань кластеризації, але для керування кількістю нейронів у радіально-базисній нейронній мережі.

Даний підхід дозволяє забезпечити високу якість обробки інформації на заданій кількості спостережень в послідовному on-line режимі в умовах істотного дефіциту апріорної та поточної інформації.

У порівнянні з нейронними мережами більшими можливостями володіють нейро-фаззі ситеми (NFS) [151-154], що поєднують у собі можливості навчання, апроксимації та лінгвістичної інтерпретації отриманих результатів. Найбільше розповсюдження серед NFS отримали ANFIS і TSK-системи, вихідний сигнал яких також залежить лінійно від синаптичних ваг, що дозволяє використовувати для їх навчання множину оптимальних адаптивних алгоритмів лінійної ідентифікації. В той же час, для уникнення появи «дір» у просторі входів, породжуваного розсіяним розбиттям (scatter partitioning) [155], використовуваним в ANFIS і TSK-системах, проводиться налаштування параметрів функцій належності в першому прихованому шарі NFS. Для цього використовується алгоритм зворотного поширення похибок, реалізований на множині епох [156] так, що говорити про навчання в реальному часі в даному випадку не доводиться.

Для управління параметрами активаційних функцій (центрів і матриць рецепторних полів) RBFN в режимі реального часу в [149, 150, 157] було

запропоновано використовувати самоорганізовану мапу Т. Кохонена [80], що забезпечує в процесі свого самонавчання в on-line режимі налаштування цих параметрів.

Тому, є доцільним поширити цей підхід для адаптивної настройки параметрів функцій належності нейро-фаззі систем.

Архітектура системи, що розглядається, наведена на рис. 3.3 та складається з п'яти послідовно з'єднаних шарів.

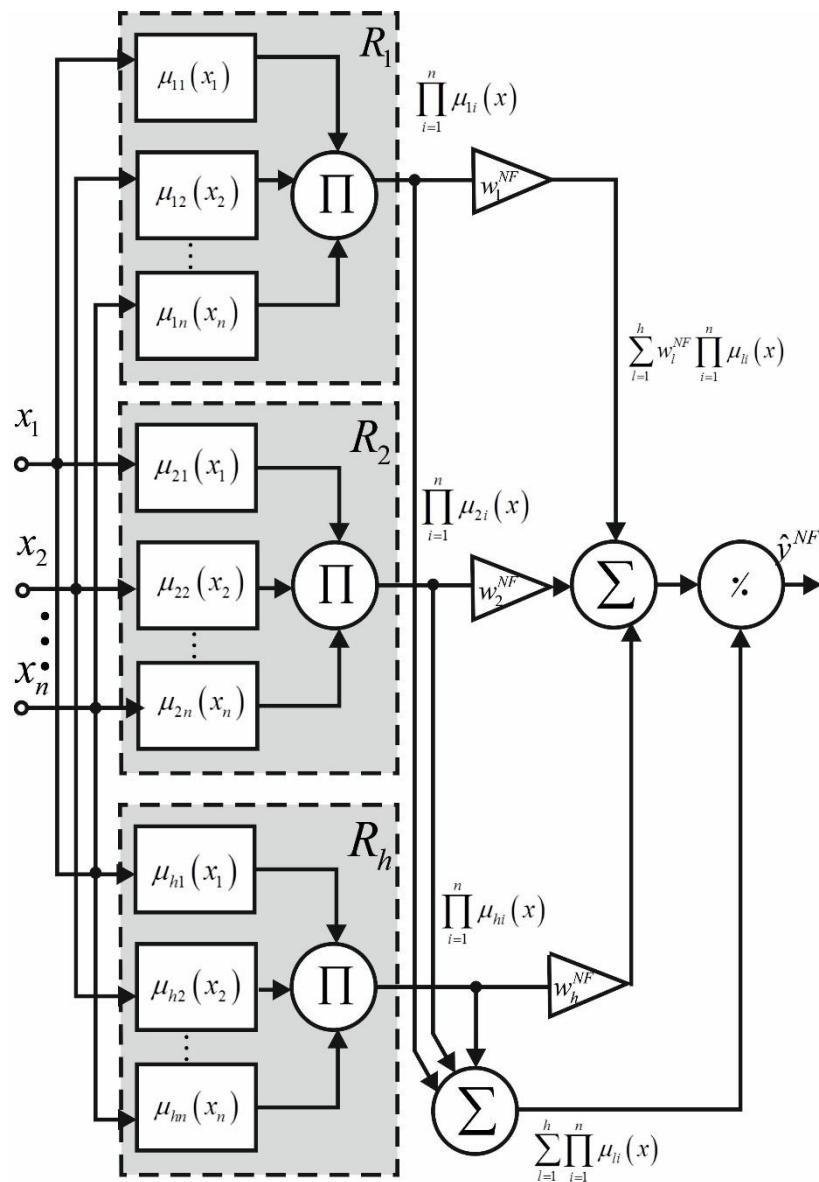


Рисунок 3.3 – Архітектура нейро-фаззі системи

На вхідний(нульовий) шар NFS подається $n \times 1$ –вимірний вектор вхідних сигналів-образів $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T$ (тут $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T$ поточний дискретний час), що підлягають обробці. Перший прихований шар містить nh (по h на кожний вхід) функцій належності $\mu_{il}(x)$, $i=1,2,\dots,n$; $l=1,2,\dots,h$ і виконує фаззифікацію вхідного простору. Другий прихований шар забезпечує агрегацію рівнів належності, розрахованих в першому шарі, і складається з h блоків множення. Третій прихований шар - це шар синаптичних ваг, що підлягають визначенню в процесі навчання. Четвертий шар утворений двома суматорами і обчислює суми вихідних сигналів другого і третього шарів. І, нарешті, в шостому (вихідному) шарі проводиться нормалізація, в результаті якої обчислюється вихідний сигнал системи \mathcal{F} .

Таким чином, якщо на вхід NFS поданий векторний сигнал $x k$, елементи першого прихованого шару обчислюють рівні належності $0 \leq \mu_{li} x k \leq 1$, при цьому в якості функцій належності найчастіше використовуються традиційні Гауссіани

$$\mu_{li} x_i k = \exp\left(-\frac{x_i k - c_{li}}{2\sigma_i^2}\right),$$

де c_{li} , σ_i – параметри центрів і ширини відповідно.

У другому прихованому шарі обчислюються агреговані значення

$\prod_{i=1}^n \mu_{li} x_i k$, при цьому для гауссіанів з однаковими параметрами ширини σ

$$\prod_{i=1}^n \mu_{li} x_i k = \prod_{i=1}^n \exp\left(-\frac{x_i k - c_{li}}{2\sigma^2}\right) = \exp\left(-\frac{\|x k - c_l\|^2}{2\sigma^2}\right),$$

(тут $c_l = c_{l1}, c_{l2}, \dots, c_{ln}^T$), тобто сигнали на виходах блоків множення другого прихованого шару аналогічні сигналам на виходах нейронів першого прихованого шару радіально-базисних нейронних мереж.

Виходами третього прихованого шару є значення $w_l \prod_{i=1}^n \mu_{li} x_i k$ (тут $w_l, l=1,2,\dots,h$ – синаптичні ваги, що підлягають визначенню), четвертого – $\sum_{l=1}^h w_l \prod_{i=1}^n \mu_{li} x_i k$ і, нарешті, на виході системи (п'ятого шару) з'являється сигнал

$$\begin{aligned} \mathcal{F}(x, k) &= \frac{\sum_{l=1}^h w_l \prod_{i=1}^n \mu_{li} x_i k}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li} x_i k} = \sum_{l=1}^h w_l \frac{\prod_{i=1}^n \mu_{li} x_i k}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li} x_i k} = \\ &= \sum_{l=1}^h w_l \varphi_l(x, k) = w^T \varphi(x, k), \end{aligned}$$

де

$$\varphi_l(x, k) = \frac{\prod_{i=1}^n \mu_{li} x_i k}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li} x_i k},$$

$$w = w_1, w_2, \dots, w_h^T, \varphi(x, k) = \varphi_1(x, k), \varphi_2(x, k), \dots, \varphi_h(x, k)^T.$$

Нескладно помітити, що розглянута система реалізує нелінійне відображення вхідного простору в скалярний вихідний сигнал подібно нормалізованій RBFN [158], а по архітектурі співпадає з системою Такагі-

Сугено-Канга нульового порядку, тобто архітектурою Л. Ванга - Дж. Менделя [153].

Як уже зазначалося, для налаштування синаптичних ваг NFS можуть бути використані відомі адаптивні алгоритми ідентифікації-навчання такі, як, наприклад, експоненційно-зважений рекурентний метод найменших квадратів.

$$\left\{ \begin{aligned} w_k &= w_{k-1} + \frac{P_{k-1} y_k - w_{k-1}^T \varphi_{xk}}{\beta + \varphi_{xk}^T P_{k-1} \varphi_{xk}} \varphi_{xk} = \\ &= w_{k-1} + \frac{P_{k-1} y_k - \hat{y}_k}{\beta + \varphi_{xk}^T P_{k-1} \varphi_{xk}} \varphi_{xk} , \\ P_k &= \frac{1}{\beta} \left(P_{k-1} - \frac{P_{k-1} \varphi_{xk} \varphi_{xk}^T P_{k-1}}{\beta + \varphi_{xk}^T P_{k-1} \varphi_{xk}} \right), \quad 0 < \beta \leq 1, \end{aligned} \right. \quad (3.7)$$

(тут y_k – зовнішній навчальний сигнал, β – параметр забування застарілої інформації), оптимальний за швидкодією градієнтний однокроковий алгоритм Качмажа-Уїдрон-Хоффа:

$$w_k = w_{k-1} + \frac{y_k - w_{k-1}^T \varphi_{xk}}{\|\varphi_{xk}\|^2} \varphi_{xk} , \quad (3.8)$$

алгоритм навчання, що володіє як слідкуючими, так і згладжуючими властивостями [159, 160]

$$\left\{ \begin{aligned} w_k &= w_{k-1} + \beta^{-1} (y_k - w_{k-1}^T \varphi_{xk}) \varphi_{xk} , \\ p_k &= \beta p_{k-1} + \|\varphi_{xk}\|^2, \quad 0 \leq \beta \leq 1, \end{aligned} \right. \quad (3.9)$$

і тому подібні процедури.

Необхідно відзначити, що процедура (3.9) пов'язана з алгоритмом (3.7) співвідношенням

$$p_k = \text{Tr } P_k,$$

а при $\beta = 0$ – набуває форму алгоритму (3.8).

Процес налаштування функцій належності першого прихованого шару проілюстровано на прикладі двовимірного вектора входів $x_k = (x_{1k}, x_{2k})^T$ і п'яти функцій належності $\mu_{li}(x_{ik})$, $l=1,2,3,4,5$; $i=1,2$ на кожному із входів. В такому випадку NFS містить $nh=10$ функцій належності. Початкове положення центрів $c_{li} = 0$ рівномірно розподілені по координатним осям x_1, x_2 , а відстань між ними визначається співвідношенням

$$\Delta_0 = \frac{x_{i\max} - x_{i\min}}{h-1} = \frac{2}{h-1} = 0,5$$

для $-1 \leq x_i \leq 1$.

Така ситуація проілюстрована на рис. 3.4.

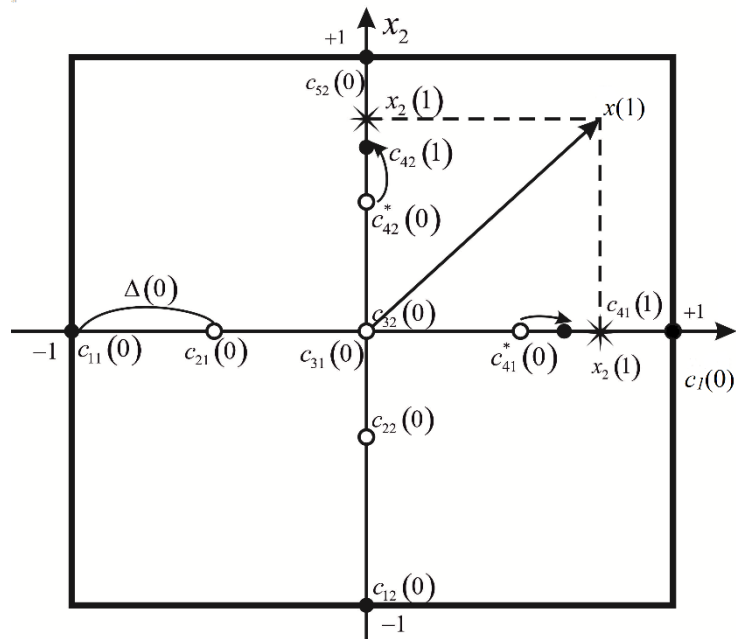


Рисунок 3.4 – Самонавчання функцій належності на першому кроці

В випадку багатовимірного вектора входів $x \in R^n$, центри c_{li} розподіляються рівномірно по осям гіперкубу $[-1, 1]^n$.

Нехай на вхід системи поданий перший вектор-образ x_1 (на рис. 2 – $x_1 = [x_{11}, x_{22}]^T$). Далі на кожній з осей знаходяться центри-«переможці» c_{li}^* найближчі до x_i в сенсі відстані

$$d_{li} = |x_i - c_{li}|,$$

тобто

$$c_{li}^* = \arg \min \{d_{li}, d_{2i}, \dots, d_{hi}\}.$$

Зауважимо, що дана процедура по суті є реалізацією процесу конкуренції по Т. Кохонену [80] з тією лише різницею, що «переможці» по

кожній з осей можуть належати функціям належності з різними індексами l . На рис. 3.4 такими «переможцями» виявилися $c_{41}^* = 0$ і $c_{42}^* = 0$.

Далі ці «переможці» підтягуються до компонентів вхідного сигналу $x_i = 1$ згідно з правилом самонавчання Т. Кохонена «Переможець отримує все» (WTA), яке для ситуації на рис. 3.4 може бути записано у вигляді

$$c_{li} = 1 = \begin{cases} c_{li}^* = 0 + \eta_{li} = 1 - x_i = 1 - c_{li}^* = 0 & \text{— для переможця } l=4, \\ c_{li} = 0 & \text{— для всіх інших, } l=1,2,3,5; \end{cases}$$

а для загального випадку:

$$c_{li} = k = \begin{cases} c_{li}^* = k-1 + \eta_{li} = k - x_i = k - c_{li}^* = k-1 & \text{— для переможця } l=1,2,\dots,h; i=1,2,\dots,n; \\ c_{li} = k-1 & \text{— для всіх інших.} \end{cases}$$

При цьому в якості значення параметру кроку навчання в простому випадку може бути прийнято значення

$$\eta_{li} = k = \frac{1}{k_{li}},$$

де k_{li} — кількість разів, коли $c_{li} = k$ був «переможцем», що відповідає популярному методу кластеризації К-середніх.

У найбільш загальному випадку може бути використана оцінка, запропонована для стандартної мапи Т. Кохонена, виду [161]:

$$\begin{cases} \eta_{li} = k = p_{li}^{-1}, \\ p_{li} = k = \beta p_{li} = k-1 + x_i^2, \quad 0 \leq \beta \leq 1. \end{cases}$$

Як можна помітити, запропонований підхід є модифікацією правила самонавчання Т. Кохонена з тією різницею, що стандартне самонавчання реалізується на гіпершарі $\|x(k)\| = 1$, а в нашому випадку - на гіперкубі $[-1, 1]^n$.

На рис. 3.5 наведена структурна схема комбінованого навчання-самонавчання нейро-фаззи системи.

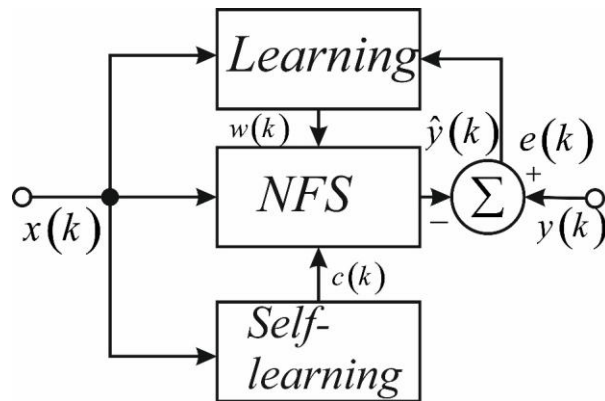


Рисунок 3.5 - Комбіноване навчання - самонавчання.

Процес функціонування цієї системи відбувається наступним чином. При надходженні на вхід вектора-образу $x(k)$ спочатку в блоці самонавчання відбувається корекція функцій належності $\mu_{li}(x_i(k))$, тобто розраховуються центри $c_{li}(k)$. Далі на підставі уточнених функцій належності

$$\mu_{li}(x_i(k)) = \exp\left(-\frac{(x_i(k) - c_{li}(k))^2}{2\sigma^2}\right)$$

і раніше розрахованого вектора синаптичних ваг $w(k-1)$ за допомогою алгоритмів навчання з вчителем ((1) або (2), або (3)) відбувається розрахунок ваг вихідного шару NFS $w(k)$.

Таким чином, запропоновано підхід до комбінованого навчання синаптичних ваг і центрів функцій належності нейро-фаззі систем, заснований на використанні як правил навчання з учителем, так і самонавчання. Основною перевагою підходу, що розвивається є можливість його використання в on-line режимі, коли навчальна вибірка на вхід системи подається послідовно, а її обсяг заздалегідь не фіксований. Отримані результати можуть бути використані при вирішенні широкого класу задач, що виникають у Dynamic Data Mining.

3.2 Багатошарова нейро-фаззі система для діагностування-класифікації нестационарних нелінійних даних

Для вирішення широкого кола задач інтелектуального аналізу медичних даних (Medical Data Mining), що пов'язані насамперед з діагностикою, класифікацією, компресією, розпізнаванням образів тощо, все частіше використовується штучні нейронні мережі [51], завдяки їх універсальним апроксимувальним можливостям і здатності навчання за наявними експериментальними даними. І хоча для вирішення подібних задач найбільшого поширення набув класичний багатошаровий персептрон, не можна не відзначити такі його основні недоліки, як досить великий обсяг навчальної вибірки, низька швидкість збіжності процесу зворотного поширення похибок, необхідність використання великої кількості епох навчання. І якщо суто обчислювальні проблеми можна обійти, необхідність наявності репрезентативної навчальної вибірки істотно ускладнює використання цієї нейронної мережі при вирішенні багатьох практичних задач. Особливо гостро ця проблема постає в медико-біологічних дослідженнях, де вибірка даних коротка і при цьому об'єкт описується безліччю різнопланових характеристик [162, 163].

У даній ситуації більш кращими є радіально-базисні нейронні мережі [51], чий вихідний сигнал лінійно залежить від налаштованих синаптичних

ваг. Данна обставина дозволяє використовувати для навчання цих мереж широкий арсенал добре відомих підходів, від стандартного методу найменших квадратів до популярних алгоритмів лінійної адаптивної ідентифікації [164]. І хоча специфіка завдань діагностування-класифікації обмежує можливості використання традиційного квадратичного критерію навчання, застосування спеціалізованого критерію Дж. Шинка [165], орієнтованого на задачі розпізнавання образів з двійковим навчальним сигналом, дозволило синтезувати досить просту і ефективну діагностуючу радіально-базисну нейронну мережу [144, 166].

Незважаючи на всі свої переваги, радіально-базисні мережі все ж не є панацеєю, оскільки їх можливості обмежуються, так званим, «прокляттям розмірності», що веде до експоненціального зростання числа налаштованих синаптичних ваг в залежності від розмірності простору вхідних сигналів-образів.

Для подолання цієї проблеми найчастіше використовується процедура попереднього розміщення центрів радіально-базисних функцій за допомогою того чи іншого методу кластеризації. Таким чином, контрольоване навчання мережі доповнюється самонавчанням її центрів, що робить таке навчання надмірно громіздким. В [167] для вирішення задач обробки текстових документів у рамках проблеми Text Mining, що характеризується високою розмірністю вхідних сигналів, була введена ієрархічна радіально-базисна нейронна мережа з багатошаровою архітектурою, що в кожному вузлі використовує звичайну радіально-базисну нейронну мережу, на вхід якої подається тільки деяка частина ознак, що дозволило подолати проблему «прокляття розмірності». Головне, що вдалося домогтися в даній ситуації - це можливість роботи за умов, коли розмірності вхідних образів співрозмірні з обсягом навчальної вибірки. Разом з тим, не можна не відзначити громіздкість цієї системи, неможливість роботи в послідовному on-line режимі, високий рівень суб'єктивізму при розбитті вхідного образу на множині підвекторів-входів для кожного вузла мережі.

Так чи інакше, проблема обробки даних з високою розмірністю вектору ознак за умов, коли обсяг навчальної вибірки співрозмірний з цією розмірністю, представляє безперечний інтерес і особливо для задач класифікації та діагностики для Text Mining, Web Mining, медико-біологічних застосувань.

Досить перспективним тут є використання нейро-фаззі-систем [152], які поряд з апроксимувальними можливостями і здатністю до навчання, дозволяють забезпечити лінгвістичну інтерпретацію результатів. Не можна також не відзначити і їх еквівалентність з точки зору отриманих результатів радіально-базисних нейронних мереж [168], що дозволяє використовувати для їх навчання ті ж алгоритми, що і в радіально-базисних нейронних мережах. Синтезу такої діагностуючої нейро-фаззі системи для ситуацій, коли розмірність вхідних сигналів має один порядок з об'ємом навчальної вибірки, а самі дані надходять на вхід системи в послідовному on-line режимі, присвячений цей розділ.

3.2.1 Архітектура діагностуючої нейро-фаззі системи

Архітектура розглянутої нейро-фаззі системи наведена на рис. 3.6 і складається з шести послідовно з'єднаних шарів [169, 170-172]. На вхідний (нульовий, рецепторний) шар нейро-фаззі системи подається $(n+1)$ -вимірний вектор вхідних сигналів-образів $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T$, де $k = 1, 2, \dots, N$ - номер спостереження у вихідному масиві даних. При цьому передбачається, що всі компоненти $x_i(k)$ попередньо перетворені так, що

$$0 \leq x_i(k) \leq 1, \quad i = 1, 2, \dots, n,$$

а двійкові вхідні ознаки приймають значення 0 або 1.

Перший прихований шар містить nh функцій належності $\mu_{li}(x_i(k))$, $i = 1, 2, \dots, n$; $l = 1, 2, \dots, h$ і виконує фаззифікацію вхідних змінних, при цьому чим більше число h , тим краще апроксимувальні властивості нейро-фаззи системи, хоча для двійкових ознак достатньо мати $h = 2$.

Другий прихований шар реалізує агрегування рівнів належності, обчислених в першому шарі, і складається з h блоків множення P .

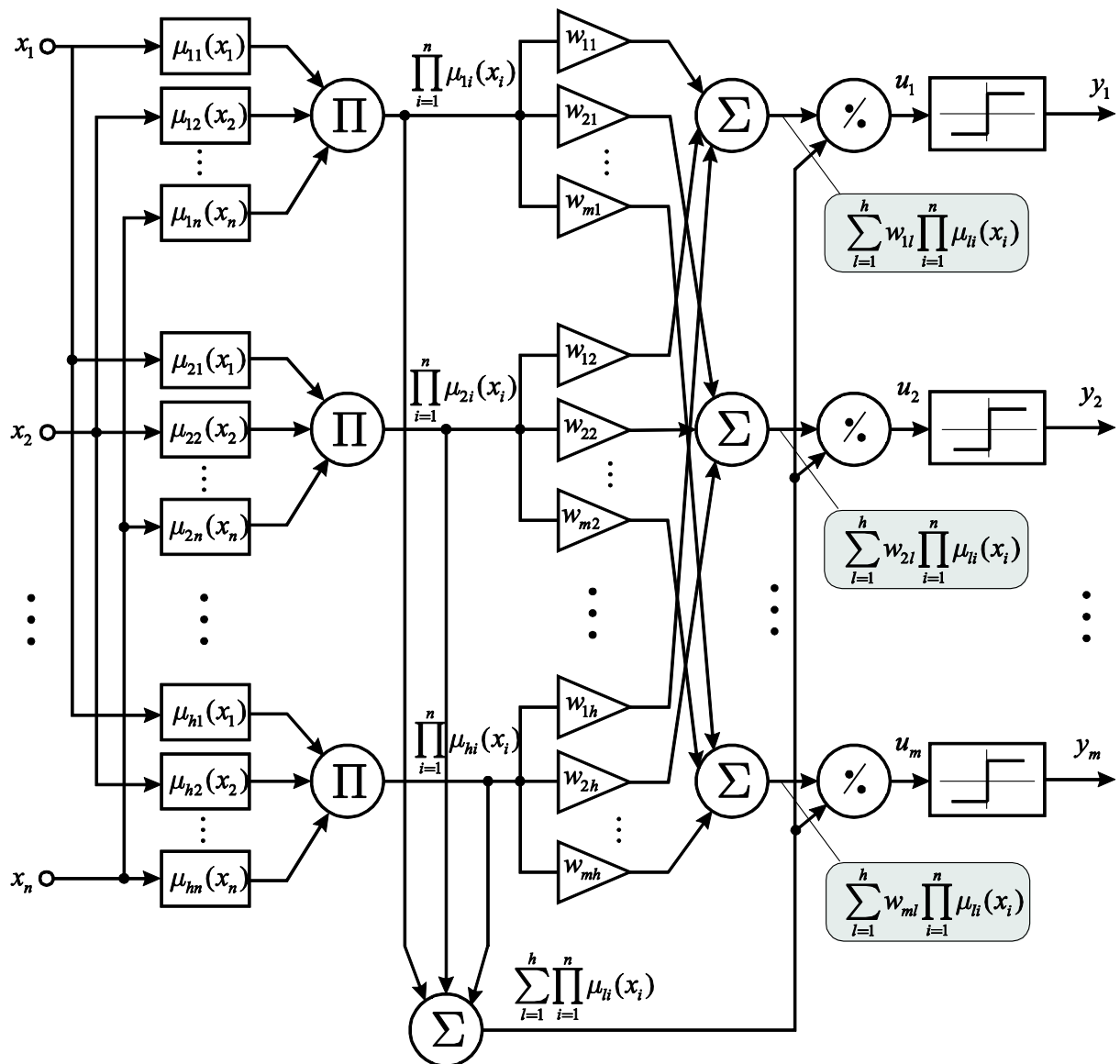


Рисунок 3.6 – Діагностуюча нейро-фаззи система

Третій прихований шар - це шар синаптичних ваг w_{jl} , $j = 1, 2, \dots, m$, що підлягають навчанню. Розглянута нейро-фаззи система містить mh

налаштовних ваг, де m - кількість можливих класів-діагнозів, по одному на кожен вихід системи. Зрозуміло, що $mh \ll e^n$, тобто кількість ваг нейро-фаззі системи істотно менше числа ваг радіально-базисної нейронної мережи.

Четвертий прихований шар утворений $m+1$ суматорами S , що обчислюють суми сигналів на виходах другого та третього прихованого шарів.

У п'ятому прихованому шарі, що утворений m блоками ділення \cdot/\cdot , проводиться нормалізація вихідних сигналів четвертого шару.

І, нарешті, вихідний (шостий) шар містить m нелінійних активаційних функцій, при цьому в задачах діагностики доцільне використання найпростіших сігнум-функцій, що приймають значення $+1$ у разі правильного діагнозу і -1 - в іншому випадку. Отже, вихідні сигнали системи $y_j(k)$ можуть приймати тільки два значення ± 1 .

Таким чином, при подачі на вхід нейро-фаззі системи векторного сигналу $x(k)$, елементи першого шара обчислюють рівні належності $m_i(x_i(k))$, при цьому зазвичай в якості функцій належності використовуються дзвонуваті (ядерні) конструкції з нестрого локальним рецепторним полем, що дозволяє уникнути виникнення «дірок» у фаззіфікованому просторі [155]. Найчастіше - це традиційні гаусіани

$$m_i(x_i(k)) = \exp\left\{-\frac{(x_i(k) - c_{li})^2}{2s_i^2}\right\} \quad (3.10)$$

де c_{li} - параметр центру (у простому випадку центри рівномірно розміщуються на інтервалі $[0,1]$ з кроком $(h-1)^{-1}$),

s_i - параметр ширини, обраний емпірично або налаштовний за допомогою зворотного поширення похибок [156].

На рис. 3.7 показаний вигляд таких функцій належності.

Зрозуміло, що для двійкових змінних $x_i(k)$ достатньо всього двох трикутних функцій належності

$$\begin{aligned} \mu_{1i}(x_i(k)) &= 1 - x_i(k), \\ \mu_{2i}(x_i(k)) &= x_i(k), \end{aligned} \quad (3.11)$$

що наведені на рис. 3.8.

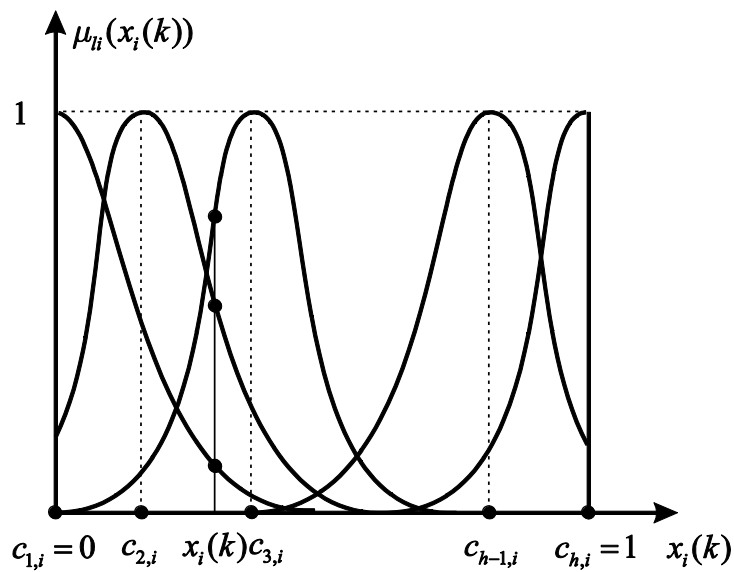


Рисунок 3.7 – Дзвонуваті функції належності

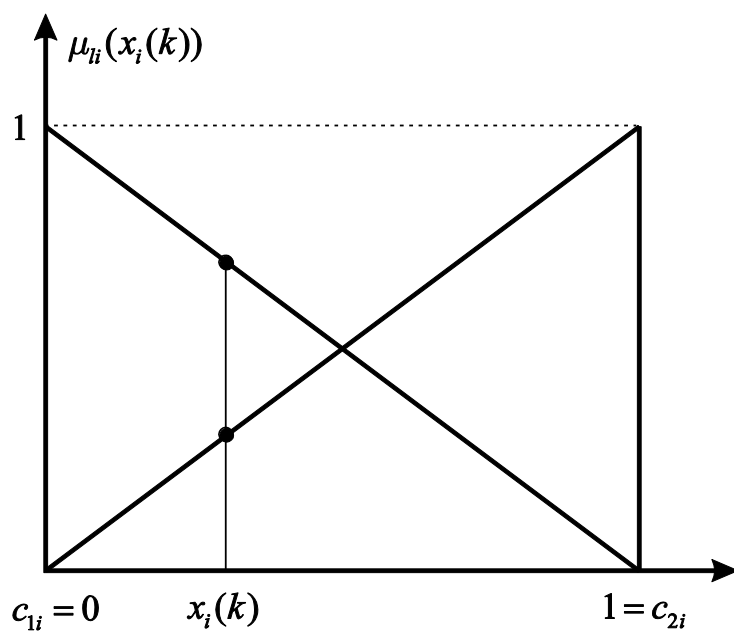


Рисунок 3.8 – Функції належності для двійкових змінних

Зауважимо також, що функції належності в ряді випадків з успіхом можуть бути використані і для ознак, що приймають довільну кількість значень (див. рис. 3.8), а число синаптичних ваг, що налаштовуються, приймає мінімально можливе значення $2m$.

На виходах другого прихованого шару з'являються агреговані значення $\sum_{i=1}^n m_i(x_i(k))$, при цьому нескладно помітити, що, якщо параметри ширини s_i однакові для всіх ознак, тобто $s_i = s$, тоді

$$\sum_{i=1}^n m_i(x_i(k)) = \sum_{i=1}^n \exp\left\{-\frac{(x_i(k) - c_{li})^2}{2s^2}\right\} = \sum_{i=1}^n \exp\left\{-\frac{\|x_i(k) - c_{li}\|^2}{2s^2}\right\},$$

(тут $c_l = (c_{l_1}, c_{l_2}, \dots, c_{l_n})^T$), тобто реалізується нелінійне перетворення аналогічне радіально-базисній нейронній мережі.

Виходами третього прихованого шару є сигнали $w_{jl} \sum_{i=1}^n m_i(x_i(k))$, четвертого $\sum_{l=1}^h w_{jl} \sum_{i=1}^n m_i(x_i(k))$ та $\sum_{l=1}^h \sum_{i=1}^n m_i(x_i(k))$, п'ятого

$$\begin{aligned} u_j(k) &= \frac{\sum_{l=1}^h w_{jl} \sum_{i=1}^n m_i(x_i(k))}{\sum_{l=1}^h \sum_{i=1}^n m_i(x_i(k))} = \sum_{l=1}^h w_{jl} \frac{\sum_{i=1}^n m_i(x_i(k))}{\sum_{i=1}^n m_i(x_i(k))} = \\ &= \sum_{l=1}^h w_{jl} j_l(x(k)) = w_j^T j(x(k)) \end{aligned}$$

$$\text{(тут } j_l(x(k)) = \sum_{i=1}^n m_i(x_i(k)) \exp\left(-\sum_{l=1}^h \sum_{i=1}^n m_i(x_i(k))\right), \quad w_j = (w_{j1}, w_{j2}, \dots, w_{jh})^T,$$

$j(x(k)) = (j_1(x(k)), j_2(x(k)), \dots, j_h(x(k)))^T$) і, нарешті, шостого

$$y_j(k) = \text{sign} u_j(k).$$

Нескладно бачити, що розглянута нейро-фаззі система є модифікацією системи Л.Ванга-Дж. Менделя [173], що орієнтована на вирішення завдань on-line діагностики-класифікації.

3.2.2 Навчання діагностуючої нейро-фаззі системи

Для навчання синаптичних ваг введеної системи будемо використовувати метод навчання, що заснований на спеціалізованому критерії [165], який призначено для вирішення задач розпізнавання образів, класифікації, діагностики, тощо.

Введемо до розгляду m похибок навчання

$$e_j(k) = d_j(k) - y_j(k) = d_j(k) - \text{sign} u_j(k)$$

та m критеріїв, що засновані на цих похибках

$$\begin{aligned} E_j(k) &= e_j(k) u_j(k) = d_j(k) u_j(k) - |u_j(k)| = \\ &= (d_j(k) - \text{sign} w_j^T(x(k))) w_j^T(x(k)), \end{aligned} \quad (3.12)$$

де $d_j(k) \in \{-1, 1\}$ - навчальний сигнал, що приймає значення 1, якщо вхідний вектор $x(k)$ належить до j -го діагнозу, і -1 в іншому випадку.

Для налаштування синаптичних ваг використовуємо стандартну градієнтну процедуру мінімізації критерію (4.3)

$$w_{jl}(k+1) = w_{jl}(k) - h(k) \frac{\partial E_j(k)}{\partial w_{jl}}$$

(тут $h(k)$ - параметр кроку навчання), яка у векторній формі може бути переписана у вигляді

$$\begin{aligned} w_j(k+1) &= w_j(k) + h(k) e_j(k) j(x(k)) = \\ &= w_j(k) + h(k) (d_j(k) - \text{sign } w_j^T(k) j(x(k))) \Psi(x(k)), \end{aligned} \quad (3.13)$$

$j = 1, 2, \dots, m.$

Вводячи далі загальний критерій для всіх виходів системи

$$E(k) = \sum_{j=1}^m E_j(k) = \sum_{j=1}^m e_j(k) u_j(k),$$

можна записати алгоритм навчання всіх синаптичних ваг системи у вигляді

$$W(k+1) = W(k) + h(k) (d(k) - \text{sign } W(k) j(x(k))) \Psi^T(x(k)), \quad (3.14)$$

де $\text{sign}(u_1(k), u_2(k), \dots, u_m(k))^T = (\text{sign } u_1(k), \text{sign } u_2(k), \dots, \text{sign } u_m(k))^T$,

$d(k) = (d_1(k), d_2(k), \dots, d_m(k))^T$,

$W(k) = \begin{pmatrix} w_1^T(k) \\ w_2^T(k) \\ \vdots \\ w_m^T(k) \end{pmatrix} \quad (m \times h)$ - матриця налаштованих синаптичних ваг.

Відомо, що градієнтні алгоритми типу (4.3) - (4.5) забезпечують збіжність в досить широкому діапазоні варіювання параметра кроку $h(k)$, однак при цьому швидкість збіжності може виявитися недостатньою. Збільшити цю швидкість можна, скориставшись квазін'ютонівськими процедурами навчання [174], наприклад,

$$w_j(k+1) = w_j(k) + (j(x(k))j^T(x(k)) + hI)^{-1} e_j(k)j(x(k)), \quad (3.15)$$

де $h > 0$ - регуляризуючий параметр, I - $(n \times n)$ - одинична матриця.

Скориставшись далі лемою обернення матриць, нескладно показати, що [144]

$$(j(x(k))j^T(x(k)) + hI)^{-1} j(x(k)) = \frac{j(x(k))}{h + \|j(x(k))\|^2},$$

і переписати (3.15) в компактній формі

$$w_j(k+1) = w_j(k) + \frac{e_j(k)j(x(k))}{h + \|j(x(k))\|^2}, \quad (3.16)$$

або

$$W(k+1) = W(k) + \frac{d(k) - \text{sign}W(k)j(x(k))}{h + \|j(x(k))\|^2} j^T(x(k)), \quad (3.17)$$

що при $h = 0$ є багатовимірним варіантом оптимального за швидкістю алгоритму навчання, введеного в [175].

Запропонована діагностувальна нейро-фаззі система і адаптивний метод її навчання призначені для вирішення завдань розпізнавання образів, класифікації, діагностики тощо в умовах коли обсяг навчальної вибірки є

близькими до розмірності вхідних образів, а самі ці образи надходять на обробку послідовно в on-line режимі. Особливістю пропонованої системи є істотно менша кількість параметрів, що настроюються в порівнянні з штучними нейронними мережами, які вирішують ту же задачу.

Система характеризується простотою чисельної реалізації, високою швидкістю при навчанні, можливістю обробки інформації, що задається в різних шкалах (кількісній, ранговій, бінарній), що дуже часто виникає в задачах інтелектуальної обробки медичних даних.

4 НАВЧАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ НА НЕПОВНИХ ВИКРЕВЛЕНИХ ДАНИХ

У теорії ШНМ за замовчуванням мається на увазі, що вибірки даних, на яких провадиться навчання, тестування, а потім і постійне функціонування ШНМ, є повними, тобто вони не мають пропущених значень. Таке припущення справедливо для більшості випадків застосування ШНМ, однак є ряд використань, де наявність пропусків у даних є невід'ємною властивістю розв'язуваної задачі. Як приклад можна вказати біомедичні використання [176, 177], де для різних пацієнтів виконується не повний набір аналізів через їх дорожнечу або відсутності необхідності. У технічних задачах дані можуть губитися внаслідок несправності датчиків, збоїв в каналах передачі інформації або помилок при зверненнях до баз даних.

Всі стандартні процедури навчання ШНМ не здатні обробляти неповні вибірки. Суть проблеми полягає в тому, що для обчислення виразів, які входять в алгоритми навчання, всі змінні повинні прийняти конкретні числові значення. Коли значення у вибірці пропущено, вираз або взагалі не може бути обчислено, або він має бути змінений так, щоб виключити це пропущене значення, тобто повністю відмовитися від використання даної змінної в процесі вирішення задачі. Коли пропущені значення розташовані у вибірці випадково, вирази не можуть бути змінені так, щоб врахувати всі можливі варіанти, тому необхідно знайти спосіб заповнення пропусків.

Існує декілька простих методів відновлення повноти вибірок:

- якщо кількість елементів вибірки з пропущеними даними незначна, повністю видалити ці елементи (розмір вибірки при цьому відповідно зменшується);
- заповнити пропуски середніми або допустимими для цих параметрів значеннями;

– якщо це допускається умовами розв'язуваної задачі, інтерполювати пропущені значення по сусідніх елементах вибірки.

Названі методи мають загальні недоліки:

– якщо частка пропусків у вибірці висока, ці методи можуть виявитися повністю непридатними;

– заміна (заповнення) пропущених значень веде до зміщення оцінок параметрів ШНМ;

– коли відновлена вибірка піддається подальшій обробці, то алгоритм навчання «не знає», які значення реальні, а які заповнені, тому вони мають однакову значущість при впливі на синаптичні ваги ШНМ.

Для подолання зазначених недоліків доцільно розробити метод навчання ШНМ, який безпосередньо працював би з неповними вибірками. Великий інтерес такий алгоритм представляє, наприклад, для задач пониження розмірності за допомогою автоасоціативних ШНМ. Це пов'язано з тим, що в таких завданнях відбувається стиснення даних з певними втратами. При наявності неповної вибірки необхідно, щоб наявна інформація була максимально збережена, а втрати відбувалися, в першу чергу, за рахунок пропущених значень.

Для стандартного методу головних компонент подібний алгоритм існує, це добре відомий алгоритм максимізації математичного очікування (Expectation Maximization – EM) [178]. Однак метод головних компонент виконує тільки лінійне перетворення, тому виникає задача синтезу подібного алгоритму для нелінійних методів зниження розмірності на основі автоасоціативної ШНМ [179, 180].

Оскільки на вхідний і вихідний шари автоасоціативної ШНМ подаються одні й ті ж дані, то задача виключення впливу пропусків розділяється на прямий (обробка інформації) і зворотний (навчання) шляхи поширення сигналів в мережі. Єдиний шар, який безпосередньо отримує вхідні значення x_k – це перший прихований шар. Зважені вхідні сигнали

підсумовуються для отримання сигналів активації нейронів наступним чином

$$u_j^k = \sum_{i=1}^n w_{ji} x_i^k, \quad (4.1)$$

де u_j^k – сигнали активації j -го нейрона;

n – розмірність вхідного вектора x^k ;

x_i^k – i -й елемент вхідного вектора x^k ;

w_{ji} – відповідна синаптична вага.

Якщо вхідне значення x_i^k пропущено, то перед початком навчання змінюємо його середнім значенням по всіх наявних (непропущених) даних для i -го входу мережі. Якщо вибірка центрована, то таким значенням буде 0, що еквівалентно виключенню відповідного члена із суми (4.1).

Навчання автоасоціативної ШНМ, по суті, є процедурою оптимізації квадратичного критерію

$$E = \sum_{k=1}^N \|x^k - \hat{x}^k\|^2 = \sum_{k=1}^N \|e^k\|^2, \quad (4.2)$$

де N – довжина навчальної вибірки;

\hat{x}^k – відновлений після пониження розмірності сигнал x^k ;

e^k – сигнал похибки навчання.

При цьому, чим більше сигнал похибки, тим більше змінюються ваги мережі в процесі навчання. Якщо похибка дорівнює нулю, зміна ваг не відбувається. Таким чином, для того, щоб виключити вплив пропущених спостережень на процес навчання, тобто зворотний шлях поширення сигналу (похибки), необхідно обнуляти похибки e_i^k на тих виходах мережі, де відсутні

(пропущені) навчальні сигнали x_i^k .

Така модифікація процедури уточнення ваг еквівалентна наступній модифікації локального критерію навчання E^k :

$$E^{\delta k} = \sum_{i=1}^n \delta_i^k (x_i^k - \hat{x}_i^k)^2, \quad (4.3)$$

де δ_i^k дорівнює 1, якщо навчальний сигнал x_i^k присутній, та 0 в протилежному випадку.

Таким чином, «увага» навчальної процедури концентрується тільки на реальних даних, а пропущені значення повністю ігноруються, в результаті чого досягається максимальне збереження корисної інформації в ході зниження розмірності вихідних даних.

Після кожної епохи навчання мережі пропущені дані у вихідній вибірці x_i^k замінюються їх апроксимаціями \hat{x}_i^k , отриманими на виході мережі. В результаті чого в ході навчання мережі йде ітеративний процес відновлення пропущених значень. Аналогічним чином можна обробляти й аномальні значення, за винятком того, що їх можна не замінювати середніми перед початком навчання.

Запропонований підхід до обробки неповних вибірок може бути використаний як для лінійного, так і для нелінійного варіантів автоасоціативною ШНМ і не залежить від її конкретної архітектури та алгоритму навчання. Даний метод можна застосувати й до будь-якого іншого типу ШНМ, що навчають з учителем, за винятком того, що вихідні сигнали не можна подавати на вхід мережі для відновлення пропущених значень.

Запропонований підхід був застосований до задачі візуалізації біомедичних даних (рис. 4.1-4.4). Перша вибірка містить результати аналізів крові (35 параметрів) 26 пацієнтів, взятих до і після лікування (всього 52 заміри). Зниження розмірності виконується з 35-мірного простору в

двовимірний. Для порівняння різних алгоритмів з вибірки випадковим чином виключено 20% даних, тобто є і повна, і неповна вибірка.

Еталонна візуалізація отримана за допомогою стандартного методу головних компонент на повній вибірці (рис. 4.1). Кожна лінія представляє одного пацієнта: кінець з чорною крапкою відповідає аналізу крові, взятому до лікування, а кінець з білою крапкою – після лікування.

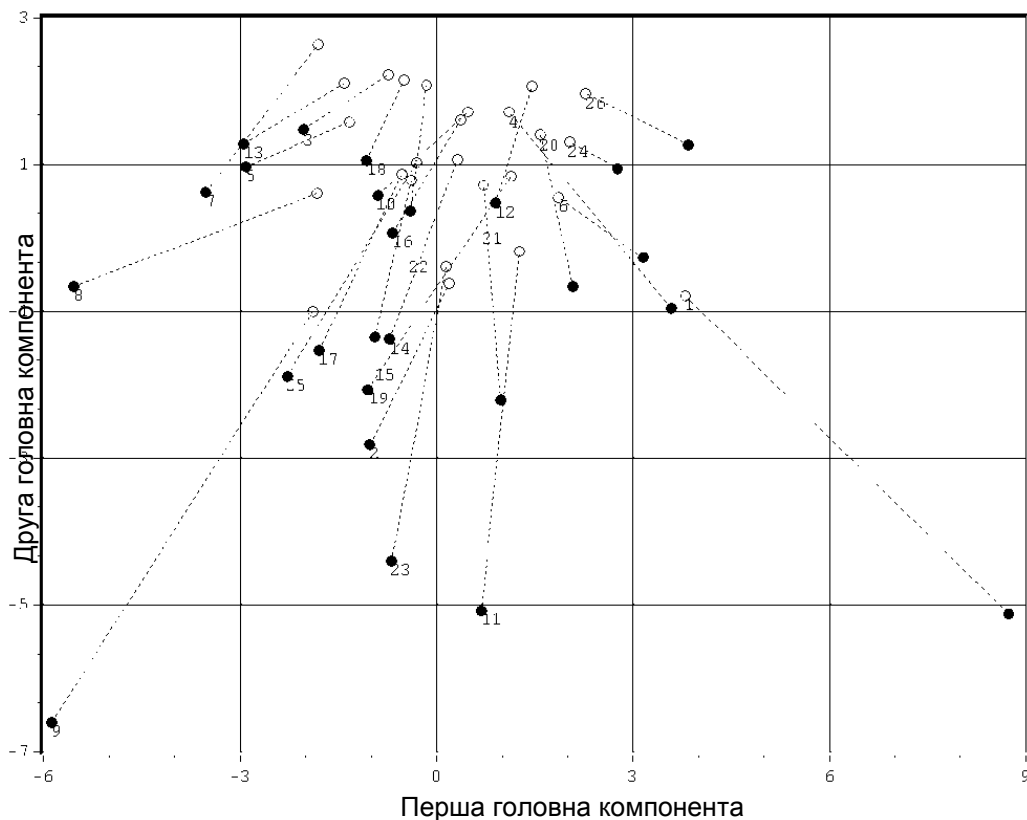


Рисунок 4.1 - Візуалізація повної вибірки

Неповна вибірка візуалізована з використанням трьох методів: рис. 4.2 – пропущені значення замінені середніми значеннями відповідних параметрів, потім застосований стандартний метод головних компонент; рис. 4.3 – алгоритм максимізації математичного очікування безпосередньо застосований до неповної вибірки; рис. 4.4 – запропонований метод безпосередньо застосований до неповної вибірки.

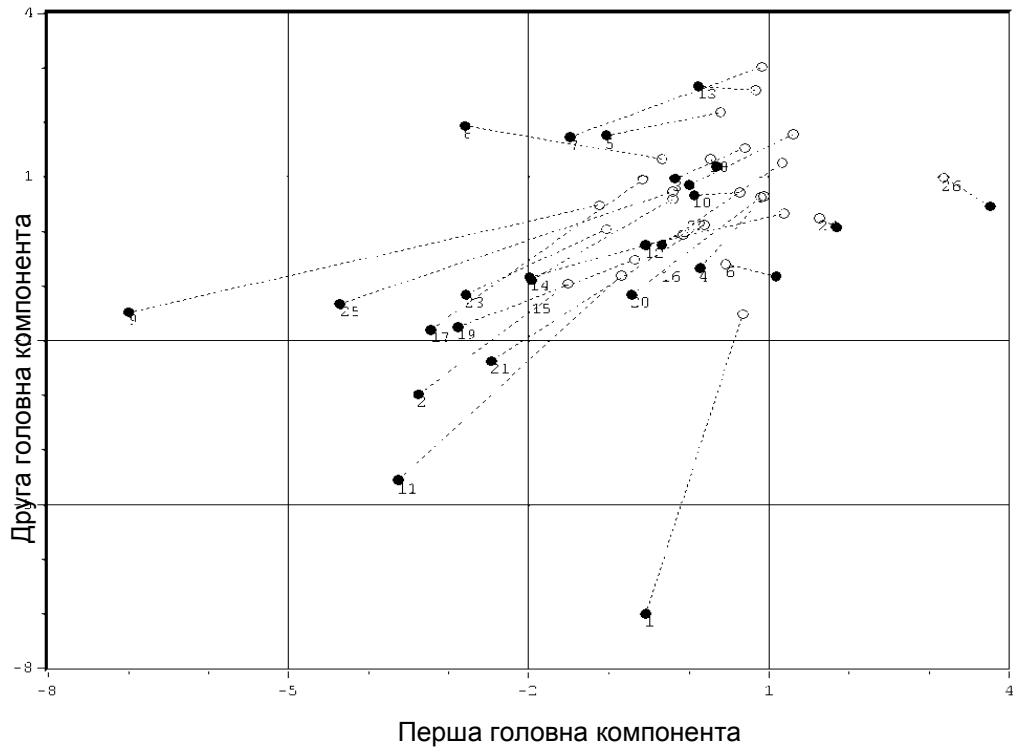


Рисунок 4.2 - Візуалізація неповної вибірки за допомогою методу головних компонент (пропущені значення замінені середніми)

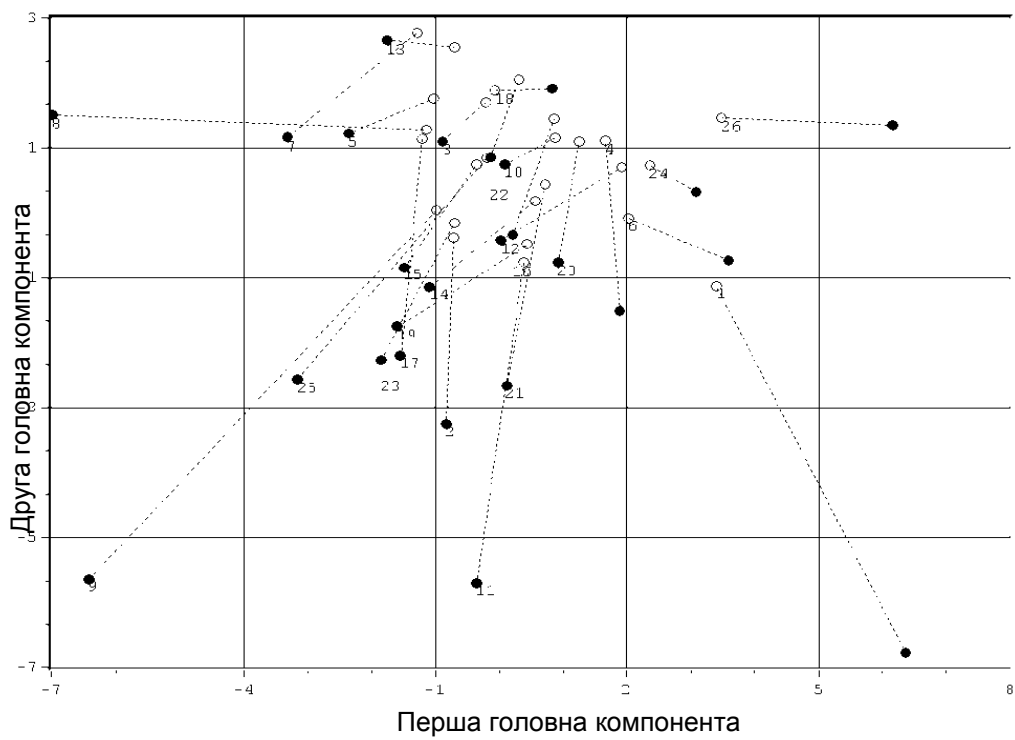


Рисунок 4.3 - Візуалізація неповної вибірки за допомогою алгоритму максимізації математичного очікування

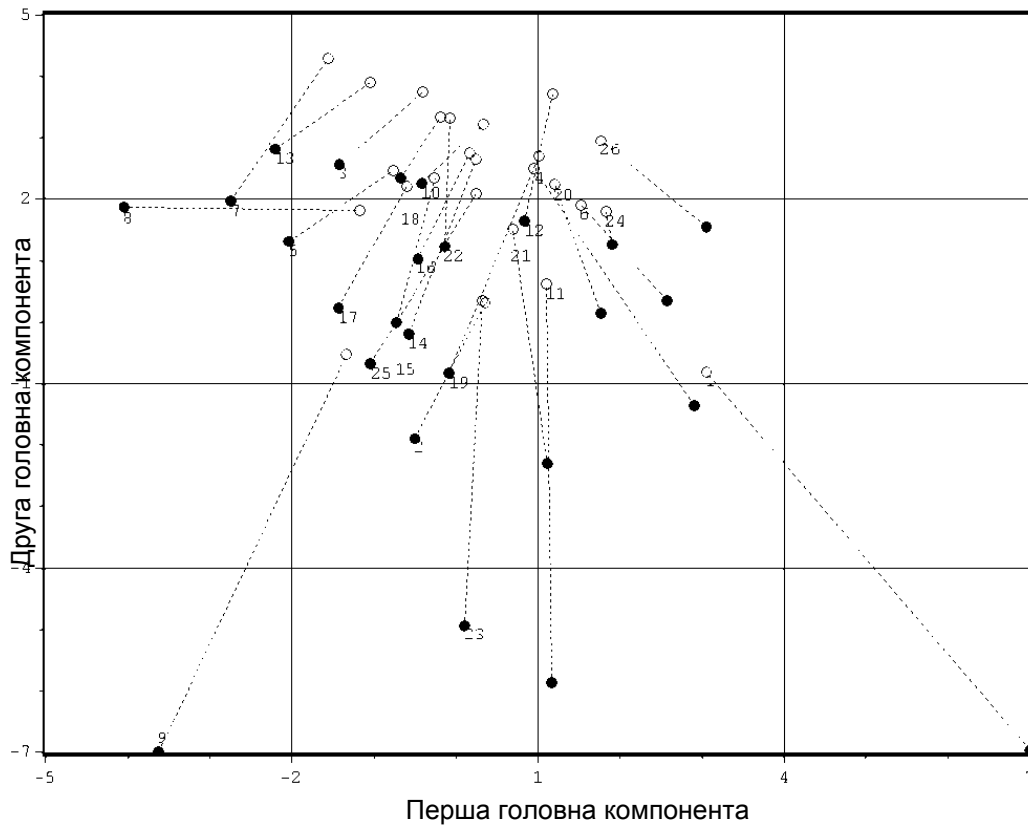


Рисунок 4.4 - Візуалізація неповної вибірки за допомогою запропонованого підходу

Тут використана лінійна автоасоціативна ШНМ (архітектура 35-2-35), оскільки порівняння проводиться з лінійним методом головних компонент.

Аналіз отриманих результатів показує наступне:

- заміна пропущених даних середніми значеннями призвела до суттєвого спотворення форми всієї вибірки і взаємного розташування точок в ній, це може призвести до невірних висновків, якщо візуалізація використовується для прийняття рішень або експрес-діагностування;
- алгоритм максимізації математичного очікування дав значно кращий результат, ніж попередній метод: загальна форма лише частково спотворена, але взаємне розташування точок все ще невірне в багатьох випадках і ближче до попереднього результату, ніж до еталону;

– запропонований підхід дозволив отримати найкращу візуалізацію в плані близькості до еталонної візуалізації, отриманої на повній вибірці за допомогою методу головних компонент: форма вибірки збережена майже точно, а взаємне розташування точок має лише незначне спотворення.

Друга вибірка допомагає продемонструвати всю потужність запропонованого підходу. Вона містить результати 135 біомедичних аналізів для 202 пацієнтів. Складність полягає в тому, що пропущені 70% вимірів, оскільки для кожного пацієнта виконувалася тільки частина аналізів. Були отримані дві візуалізації: лінійна (рис. 4.5) з використанням автоасоціативної ШНМ з архітектурою 135-2-135 і нелінійна (рис. 4.6) з використанням автоасоціативної ШНМ з архітектурою 135-25-2-25-135.

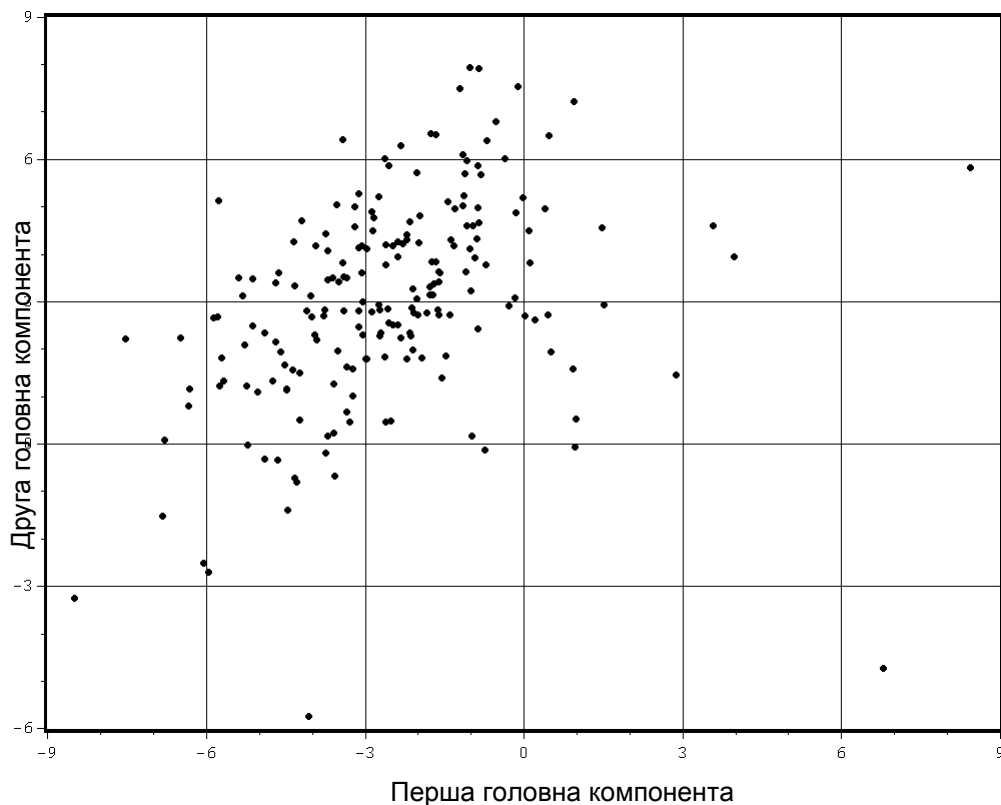


Рисунок 4.5 - Лінійна візуалізація другої вибірки

Нелінійна візуалізація забезпечила значно краще представлення даних: чітко видно два кластери, які не виявила лінійна візуалізація, але вони

насправді присутні в даних.

В [181] запропонований метод був використаний для візуалізації результатів кластеризації багатовимірних вибірок біомедичних даних.

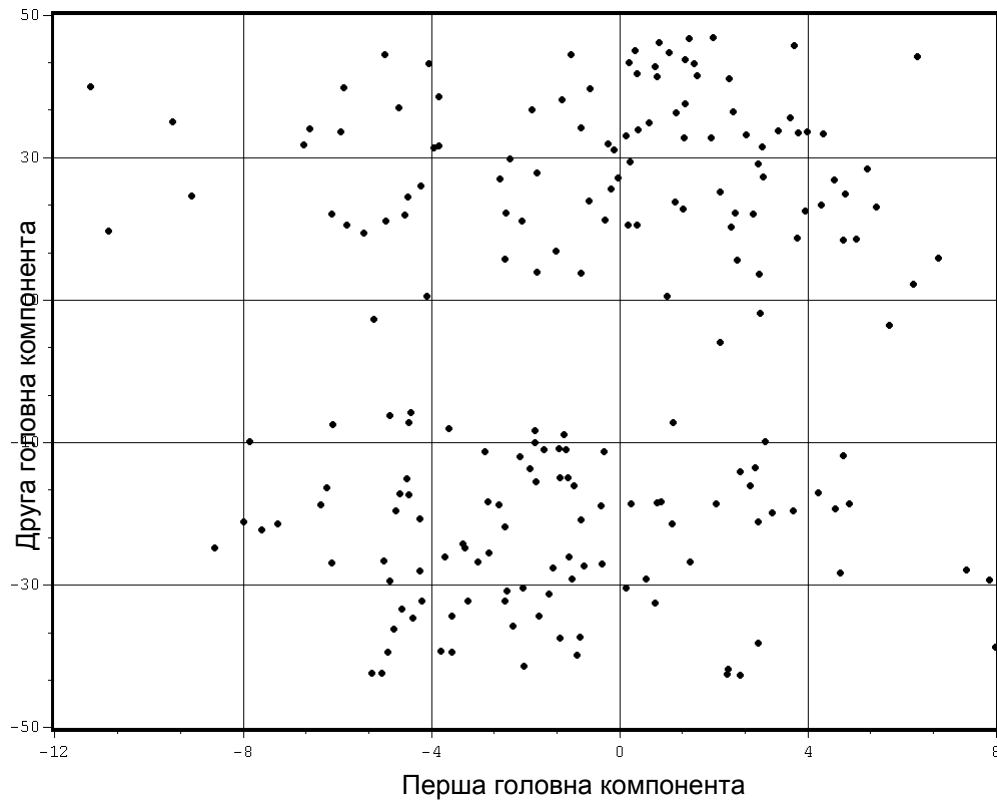


Рисунок 4.6 - Нелінійна візуалізація другої вибірки

5 МОДЕЛІ І МЕТОДИ АДАПТИВНОГО ПРОГНОЗУВАННЯ ТА КЛАСИФІКАЦІЇ ЧАСОВИХ РЯДІВ

Як правило, для успішного використання більшості процедур прогнозування необхідно мати досить представницьку вибірку спостережень, що дозволяє побудувати прийнятну по точності прогнозуючу модель. Разом з тим досить часто виникають ситуації, коли ця вибірка або мала, або прогнозований процес нестационарний так, що його передісторія не може бути використана для знаходження параметрів моделі. Зрозуміло, що в цій ситуації говорити про ефективну прогнозуючу моделі взагалі не доводиться, і в розгляд повинні бути введені методи прогнозування, які не використовують самого поняття моделі.

В основі традиційних математичних методів прогнозування (статистичних, адаптивних, нейромережових тощо) лежать математичні моделі того чи іншого виду, одержувані в результаті вирішення задачі структурної та параметричної ідентифікації. Саме на основі математичних моделей вирішується завдання часової екстраполяції, при цьому в моделі явно чи неявно присутній аргумент дискретного часу. Якщо ж даних для побудови математичної моделі недостатньо, синтез прогнозуючої моделі просто неможливий. У цьому випадку замість часової екстраполяції може бути використано просторове прогнозування (екстраполяція), яке зводиться до оцінки значень векторного поля по окремих спостереженнями. Серед методів просторової екстраполяції в якості одного з найбільш перспективних слід зазначити багатовимірну лінійну екстраполяцію [182], що підтвердила свою ефективність при вирішенні ряду реальних завдань проектування й управління складними багатовимірними нелінійними об'єктами.

5.1 Методи адаптивного прогнозування на основі багатовимірної лінійної екстраполяції і використанні метрик і функцій близькості

Розглянемо метод багатовимірної лінійної екстраполяції стосовно до задачі однокрокового прогнозування n -мірної нелінійної нестационарної часової послідовності $y(k)$, $k=1,2,\dots,N$, де k – дискретний час. Нехай аналізований ряд (i -я його компонента) теоретично може бути описана нелінійною залежністю (NARX-модель) виду

$$\begin{aligned} \hat{y}_i(k) &= f_i(y_i(k-1), \dots, y_i(k-n_{A,i}), x_1(k-1), \dots, x_1(k-n_B), \\ &\quad x_2(k-1), \dots, x_2(k-n_B), \dots, x_p(k-l), \dots, x_q(k-n_B)) = \\ &= f_i(z_{i1}(k), \dots, z_{i,n_{A,i}}(k), \dots, z_{i,n_{A,i}+n_Bq}(k)), \end{aligned} \quad (5.1)$$

де $f_i(\circ)$ – апріорі невідома нелінійна залежність, що підлягає відновленню на підставі наявних спостережень;

$\hat{y}_i(k)$ – оцінка (прогноз) контрольованої послідовності $y_i(k)$ за даними, наявними до $(k-1)$ -го моменту часу; $i=1,\dots,n$; $n_{A,i}$ – глибина обліковуємої передісторії за контрольованою послідовністю;

$x_p(k-l)$ – p -я компонента багатовимірного екзогенного сигналу, що впливає на $y_i(k)$; $l=1,\dots,n_B$; $p=1,\dots,q$. Вираз (5.1) може бути представлений і в векторно-матричній формі

$$\hat{\mathbf{y}}(k) = F(y(k-1), \dots, y(k-n_A), x(k-1), \dots, x(k-n_B)) = F(z(k)), \quad (5.2)$$

де $\hat{\mathbf{y}}(k) = (\hat{y}_1(k), \hat{y}_2(k), \dots, \hat{y}_n(k))^T$,

$y(k) = (y_1(k), \dots, y_n(k))^T$,

$$x(k-1) = (x_1(k-1), \dots, x_q(k-1))^T,$$

$$z(k) = (y^T(k-1), \dots, y^T(k-n_A), x^T(k-1), \dots, x^T(k-n_B))^T - (nn_A + qn_B) \times 1$$

- вектор передісторії.

При наявності репрезентативної навчальної вибірки нелінійні перетворення $f_i(\circ)$ та $F(\circ)$ можуть бути відновлені в процесі навчання тієї чи іншої штучної нейронної мережі, проте якщо ця вибірка мала, нейромережевої підхід виявляється непрацездатним, в той час як багатовимірною лінійною екстраполяцією дозволить отримати цілком прийнятні по точності результати.

Задачу багатовимірної лінійної екстраполяції стосовно проблемі прогнозування багатовимірних часових рядів можна описати, використовуючи позначення (5.1), (5.2), наступним чином [182]. Нехай задана матриця прецедентів (передісторії) у вигляді

$$P = \begin{pmatrix} z^T(1), & y^T(1) \\ z^T(2), & y^T(2) \\ \vdots & \\ z^T(N), & y^T(N) \end{pmatrix}$$

розмірності $N \times (nn_A + qn_B + n)$. Сама екстраполяція зводиться до знаходження в момент часу N оцінки

$$\hat{y}(N+1) = \psi(z(N+1), P),$$

де $\psi(\circ)$ – власне алгоритм екстраполяції, який повинен задовільняти ряду вимог, основною з яких є те, що в результаті його використання повинні точно відновлюватися всі наслідки матриці прецедентів, тобто

$$\mathfrak{f}(k) = y(k) = \psi(z(k), P), \quad k = 1, 2, \dots, N.$$

Важливо відзначити, що методи прогнозування, засновані на тих чи інших математичних моделях, практично ніколи не забезпечують виконання умови (5.1).

З інших вимог можна відзначити те, що алгоритм $\psi(\circ)$ повинен бути векторним, тобто

$$\Psi = (\psi_1, \dots, \psi_n), \quad y_i = \psi_i(Z, P), \quad i = 1, \dots, n;$$

трудомісткість реалізації повинна зростати по n та N не швидше ніж лінійно, алгоритм повинен бути працездатним при будь-яких N (навіть при $N = 1$). Зауважимо також, що при малих N , не кажучи вже про $N = 1$, досить точна математична модель в принципі не може бути побудована. Тоді розглянемо алгоритм багатовимірної лінійної екстраполяції з евклідової метрикою.

Слідуючи [182], представимо алгоритм багатовимірної лінійної екстраполяції у вигляді такої послідовності кроків:

1. Формування передісторії прогнозованого процесу у вигляді матриць

$$Z = (z(1), z(2), \dots, z(N)) - (nn_A + qn_B) \times N \text{-матриця,}$$

$$Y = (y(1), y(2), \dots, y(N)) - n \times N \text{-матриця.}$$

2. Знаходження вектора вагових коефіцієнтів $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_N)^T$, які доставляють мінімум нормі

$$\|z(N+1) - \sum_{k=1}^N \lambda(k)z(k)\|^2 = \|Z(N+1) - Z\lambda\|^2. \quad (5.3)$$

3. Формування оптимального прогнозу у вигляді лінійної комбінації

$$\hat{y}(N+1) = \sum_{k=1}^N \lambda_k y(k) = Y\lambda. \quad (5.4)$$

Мінімізація виразу (5.4) може бути зроблена за допомогою стандартного методу найменших квадратів, в результаті чого отримуємо відоме рішення

$$\lambda = (Z^T Z)^{-1} Z^T z(N+1), \quad (5.5)$$

яке існує тільки тоді, коли матриця $Z^T Z$ невироджена. Оскільки при малих N ($N < n n_A + q n_B$) це не так, замість операції обернення матриць пропонується використовувати операцію псевдообернення [183], в результаті чого приходимо до виразу

$$\lambda = Z^+ z(N+1),$$

звідки випливає, що фактично задача зводиться до знаходження ортогональної проєкції вектора $z(N+1)$ на лінійну оболонку, утворену векторами передісторії. З обчислювальної точки зору ніяких труднощів при реалізації даної процедури не виникає, однак, рішення задачі ускладнюється, якщо дані на обробку надходять послідовно в реальному часі $k = 1, 2, \dots$. При цьому всі попередні співвідношення можна переписати в такій формі:

$$\begin{aligned} Z_k &= (z(1), z(2), \dots, z(k)), \\ Y_k &= (y(1), y(2), \dots, y(k)), \\ \|z(k+1) - \sum_{l=1}^k \lambda_l z(l)\|^2 &= \|z(k+1) - Z_k \Lambda_k\|^2, \end{aligned} \quad (5.6)$$

$$\Lambda_k = (\lambda_1, \lambda_2, \dots, \lambda_k)^T,$$

$$\mathfrak{F}(k+1) = \sum_{l=1}^k \lambda_l y(l) = Y_k \Lambda_k, \quad (5.7)$$

$$\Lambda_k = Z_k^+ z(k+1). \quad (5.8)$$

В [182] для розрахунку матриці Z_{k+1}^+ за наявною Z_k^+ і надійшовшими на обробку $z(k+1)$ та $y(k+1)$ пропонується використовувати формулу Гревіль, хоча набагато простіше застосувати регуляризоване версію (5.5) у вигляді

$$\Lambda_k = (Z_k^T Z_k + \gamma I_k)^{-1} Z_k z(k+1) \quad (5.9)$$

де γ – параметр регуляції,

I – $(k \times k)$ одинична матриця.

Для обробки нестационарних часових рядів, характеристики яких непередбачувано змінюються в часі, замість обробки всієї наявної вибірки, доцільно вирішувати задачу на «ковзному вікні», що складається з останніх спостережень. При цьому співвідношення (5.6) - (5.8) можуть бути переписані таким чином:

$$Z_{k,\chi} = (z(k-\chi+1), z(k-\chi+2), \dots, z(k)),$$

$$Y_{k,\chi} = (y(k-\chi+1), y(k-\chi+2), \dots, y(k)),$$

$$\|z(k+1) - \sum_{l=k-\chi+1}^k \lambda_l z(l)\|^2 = \|z(k+1) - Z_{k,\chi} \Lambda_{k,\chi}\|^2, \quad (5.10)$$

$$\Lambda_{k,\chi} = (\lambda_{k-\chi+1}, \dots, \lambda_k)^T,$$

$$\mathfrak{F}(k+1) = \sum_{l=k-\chi+1}^k \lambda_l y(l) = Y_{k,\chi} \Lambda_{k,\chi}, \quad (5.11)$$

$$\Lambda_{k,\chi} = Z_{k,\chi}^+ z(k+1). \quad (5.12)$$

Для реалізації цієї процедури в реальному часі в [182] було запропоновано рекурентний алгоритм псевдообернення на «ковзному вікні», що відрізняється громіздкістю і обчислювальною складністю.

У зв'язку з цим доцільно замість операції рекурентного псевдообернення використовувати модифікацію (5.9) в «віконному» варіанті, при цьому його пакетна форма може бути записана у вигляді

$$\Lambda_{k,\chi} = (Z_{k,\chi}^T Z_{k,\chi} + \gamma I_\chi)^{-1} Z_{k,\chi} z(k+1), \quad (5.13)$$

а рекурентна [184] –

$$\left\{ \begin{array}{l} \Lambda_{k+1,\chi} = \Lambda_{k,\chi} + \frac{P(k)(z(k+1) - \Lambda_{k,\chi}^T z(k))}{1 + z^T(k)P(k)z(k)} z(k), \\ \tilde{P}(k-1) = P(k-1) + \frac{P(k-1)z(k-\chi)z^T(k-\chi)P(k-1)}{1 - z^T(k-\chi)P(k-1)z(k-\chi)}, \\ P(k) = \tilde{P}(k-1) - \frac{\tilde{P}(k-1)z(k)z^T(k)\tilde{P}(k-1)}{1 + z^T(k)\tilde{P}(k-1)z(k)}, \\ \Lambda_{0,\chi} = 0, \quad P(k) = \gamma^{-1}I_\chi. \end{array} \right.$$

Власне ж прогноз обчислюється згідно співвідношенню (5.10).

Зрозуміло, що застосування для прогнозування співвідношень (5.10), (5.12) різко спрощує використання методу, однак залишається питання обґрунтувати ванного вибору величини вікна χ , яке, як правило, задається з суто суб'єктивних міркувань, що знижує ефективність підходу в цілому.

Пропонований нижче метод багатовимірної екстраполяції заснований на використанні деякої функції близькості (відстані) останнього вектора

передісторії $z(N+1)$ до всіх попередніх даними $z(1), \dots, z(N)$ і формуванні прогнозу $\hat{z}(N+1)$ за допомогою цієї ж функції.

Реалізація методу складається з послідовності наступних кроків:

1. Розрахунок відстані між вектором $z(N+1)$ і усіма попередніми $z(k)$ на основі деякої функції близькості $d(N+1, k)$ - в простому випадку евклідової метрики

$$d(N+1, k) = \|z(N+1) - z(k)\| \quad \forall k.$$

2. Впорядкування (ранжування) цих відстаней в порядку зростання так, що

$$d^1(N+1, k_1) < d^2(N+1, k_2) < \dots < d^N(N+1, k_N).$$

3. Відбір перших χ векторів, для яких виконується умова

$$d^\chi(N+1, k_\chi) \leq \varepsilon,$$

де ε – деякій поріг.

4. Формування набору вагових коефіцієнтів λ_l в вигляді

$$\lambda_l = \frac{(d^l)^{-1}}{\sum_{l=1}^{\chi} (d^l)^{-1}}, 1 \leq l \leq \chi,$$

які відповідають умові $\sum_{l=1}^{\chi} \lambda_l = 1$.

5. Розрахунок прогнозу

$$\hat{y}(N+1) = \sum_{l=1}^{\chi} \lambda_l z(l).$$

При надходженні нового спостереження процесу $y(N+1)$ всі ітерації повторюються.

Таким чином, на кожному кроці у формуванні прогнозу також беруть участь χ спостережень, однак це значення може змінюватися, при цьому ясно, що чим менше χ , тим більше нестационарний сигнал. Нескладно також зауважити, що якщо $y(k) = const$, то, $\chi = N$, $\lambda_l = 1/N$.

Тим самим, можна резюмувати, що запропонований метод прогнозування багатомірних нестационарних часових рядів на основі ідей просторової екстраполяції. Метод не вимагає великих обсягів апіорної інформації (мала навчальна вибірка), немає необхідності вирішення додаткових завдань оптимізації або псевдообернення, вкрай простий в обчислювальній реалізації.

При поступлении нового наблюдения процесса все итерации повторяются.

5.2 Адаптивне прогнозування часових рядів при рівновіддалених спостереженнях

Для успішного використання всіх традиційних методів прогнозування, вихідна вибірка спостережень - часовий ряд - повинна бути досить репрезентативною. При цьому, чим більше параметрів містить прогнозуюча модель, тим більша за обсягом повинна бути вхідна інформація. Разом з тим в реальних задачах досить часто виникає ситуація, коли ця вибірка або мала за обсягом, або прогнозований процес нестационарний - містить як нерегулярні тренди, так і раптові стрибки, тому його передісторія не може бути використана для знаходження параметрів моделі. Зрозуміло, що багато відомі методи синтезу моделей в цьому випадку неефективні, а сама

побудована модель повинна містити невелике число оцінюваних параметрів. Ще більш складна ситуація виникає, коли спостереження розподілені нерівномірно на часовій шкалі, тобто квантування контрольованого процесу відбувається із змінним кроком. У цьому випадку спектр можливих підходів ще більше звужується.

У зв'язку з цим пропонується підхід до синтезу адаптивних прогнозуючих моделей, який виробляється в умовах обмеженої навчальної вибірки, при цьому дані на обробку можуть подаватися через довільні заздалегідь невідомі інтервали часу.

Для вирішення даної задачі з успіхом можуть бути використані ортогональні поліноми Чебишева [185-188], що володіють рядом корисних властивостей.

Математична модель на основі цих поліномів може бути побудована на основі малої вибірки і містить невелике число оцінюваних параметрів. При додаванні нового члена в модель немає необхідності перераховувати вже існуючі параметри, які, у свою чергу, можуть розраховуватися на основі звичайного методу найменших квадратів, а похибка апроксимації розподіляється рівномірно по інтервалу спостережень. Крім того, багато відомих поліноми, наприклад, Лагерра, Ерміта та ін. Є окремим випадком поліномів Чебишева.

$$Q_m(x) = \sum_{l=0}^m c_l f_l(x), \quad (5.14)$$

$$c_l = \frac{\sum_{k=0}^N y(k) f_l(x(k))}{\sum_{k=0}^N f_l^2(x(k))},$$

де $y(0), y(1), \dots, y(N)$ – апроксимуємої функція, задана на довільних вузлах $x(0), x(1), \dots, x(N)$,

$$f_0(x) = x - b_1, f_{l+1}(x) = (x - b_{l+1})f_l(x) - a_{l+1}f_{l-1}(x), \quad l = 0, 1, \dots, N,$$

$$a_{l+1} = \frac{\sum_{k=0}^N x^l(k) f_l(x(k))}{\sum_{k=0}^N x^{l-1}(k) f_{l-1}(x(k))},$$

$$b_{l+1} = \frac{\sum_{k=0}^N x^{l+1}(k) f_l(x(k))}{\sum_{k=0}^N x^l(k) f_l(x(k))} - \frac{\sum_{k=0}^N x^l(k) f_{l-1}(x(k))}{\sum_{k=0}^N x^{l-1}(k) f_{l-1}(x(k))}.$$

При цьому найкращий порядок полінома m , який забезпечує необхідну точність $\sigma_m < \varepsilon$, можна визначити за формулами [187]

$$\sigma_m = \sqrt{\frac{\sigma_m^2}{N - m - 1}},$$

$$\sigma_m^2 = \sigma_{m-1}^2 - c_m^2 \sum_{k=0}^N x^m(k) f_m(x(k)),$$

$$\sigma_0^2 = \sum_{k=0}^N y^2(k) - \frac{1}{N} \left(\sum_{k=0}^N y(k) \right)^2.$$

Аргумент $x(k)$ для вирішення задачі прогнозування можна перевести в часову шкалу, при цьому безперервний час, в якому відбувається контрольований процес, перетвориться в дискретне у формі

$$k = \frac{x(k) - x(0)}{\Delta},$$

де Δ – мінімальний такт квантування.

При цьому можна перейти від уявлення полінома у формі (5.14) до вираження

$$Q_m(k) = c_0 \phi_{0N}(k) + c_1 \phi_{1N}(k) + \dots + c_m \phi_{mN}(k), \quad (5.15)$$

де коефіцієнти розкладання можуть бути знайдені шляхом мінімізації критерію

$$I^I(k) = \sum_{k=0}^N (y(k) - Q_m(k))^2,$$

що веде до простих співвідношень [187]

$$c_l = \frac{\sum_{k=0}^N y(k)\phi_{lN}(k)}{\sum_{k=0}^N \phi_{lN}^2(k)},$$

$$c_0 = \frac{1}{N} \sum_{k=0}^N y(k).$$

Самі ж ортогональні функції $\phi_{lN}(k)$ відомі заздалегідь [185, 186] і мають такий вигляд

$$\phi_{0N}(k) = 1,$$

$$\phi_{1N}(k) = 1 - 2\frac{k}{N},$$

$$\phi_{2N}(k) = 1 - 6\frac{k}{N} + 6\frac{k(k-1)}{N(N-1)},$$

$$\phi_{3N}(k) = 1 - 12\frac{k}{N} + 30\frac{k(k-1)}{N(N-1)} - 20\frac{k(k-1)(k-2)}{N(N-1)(N-2)}$$

і т.д.

Тут важливо зазначити, що в опис всіх ортогональних поліномів входить обсяг вибірки N , тому наведені вище вирази повинні бути відповідним чином модифіковані для того, щоб організувати процеси адаптивної обробки та прогнозування.

У задачах, коли дані надходять на обробку послідовно, доцільно організувати процес адаптивної настройки параметрів прогнозуючої моделі, а оскільки параметри, що підлягають оцінюванню, входять в опис (5.15) лінійно, то в цьому випадку можна скористатися методами класичної теорії ідентифікації [189].

Вводячи в розгляд критерій ідентифікації

$$I^1(k) = \sum_{k=0}^N \alpha(k) \left(y(k) - \sum_{l=0}^m c_l \phi_{lN}(k) \right)^2$$

(тут $\alpha(k)$ - система ваг, яка визначає процес забування застарілої інформації) і мінімізуючи його за параметрами c_l приходимо до стандартної оцінки зважених найменших квадратів.

$$c_l(N) = \frac{\sum_{k=0}^N \alpha(k) y(k) \phi_{lN}(k)}{\sum_{k=0}^N \alpha(k) y_{lN}^2(k)}. \quad (5.16)$$

Вводячи далі вектори

$$c = (c_0, c_1, \dots, c_m)^T,$$

$$\phi(k) = (\phi_{0N}(k), \phi_{1N}(k), \dots, \phi_{mN}(k))^T,$$

оцінку (5.16) можна переписати у векторно-матричній формі

$$c(N) = \left(\sum_{k=0}^N \alpha(k) \phi(k) \phi^T(k) \right)^{-1} \sum_{k=0}^N \alpha(k) \phi(k) y(k) \quad (5.17)$$

або

$$\begin{cases} c(N) = c(N-1) + \frac{\alpha(N)P(N-1)(y(N) - c^T(N-1)\phi(N))\phi(N)}{1 + \alpha(N)\phi^T(N)P(N-1)\phi(N)}, \\ P(N) = P(N-1) + \frac{\alpha(N)P(N-1)\phi(N)\phi^T(N)P(N-1)}{1 + \alpha(N)\phi^T(N)P(N-1)\phi(N)}, \end{cases} \quad (5.18)$$

при цьому в якості компонентів $\phi_{lN}(k)$ зручно скористатися системою ортонормованих поліномів Чебишева виду [186]:

$$\begin{aligned} \phi_{0N}(k) &= \frac{1}{\sqrt{N}}, \\ \phi_{1N}(k) &= \sqrt{\frac{3}{N(N^2-1)}}(2k-N-1), \\ \left\{ \begin{aligned} \phi_{lN}(k) &= (A_l k + B_l)\phi_{l-1,N}(k) - H_l \phi_{l-2,N}(k), l = 2, 3, \dots, m, \\ A_l &= \frac{2}{l} \sqrt{\frac{D_l(D_l-2)}{G_l}}, \\ B_l &= -\frac{N+1}{l} \sqrt{\frac{D_l(D_l-2)}{G_l}} = -\frac{N+1}{l} A_l, \\ H_l &= \frac{l-1}{l} \sqrt{\frac{D_l(D_l+G_l-2)}{(D_l-4)G_l}}, \\ D_l &= 2l+1, \\ G_l &= N^2 - l^2. \end{aligned} \right. \quad (5.19)$$

Тут слід зауважити, що при спільному використанні рекурентного методу найменших квадратів (5.18) і співвідношень (5.19), на кожному такті необхідно перераховувати не тільки параметри моделі $c(k)$, але й параметри поліномів A_l, B_l, H_l, D_l, G_l , які визначаються обсягом навчальної вибірки. Зрозуміло, що в адаптивному режимі обробки це ускладнює чисельну реалізацію алгоритму.

Виключити цю незручність можна, застосовуючи в цьому випадку в якості вагової функції $\alpha(k)$ в (5.17) «ковзне вікно» розміру s

$$\alpha(k) = \begin{cases} 1 & \text{при } N-s+1 \leq k \leq N, \\ 0 & \text{інакше,} \end{cases}$$

що веде до оцінки

$$c(N) = \sum_{\chi=N-s+1}^N \varphi(\chi) \varphi^T(\chi)^{-1} \sum_{\chi=N-s+1}^N \varphi(\chi) y(\chi).$$

Тоді замість рекуррентного зваженого методу найменших квадратів (5.18), приходимо до адаптивної процедури виду

$$\begin{cases} P(N) = P_S(N-1) - \frac{P_S(N-1) \varphi(N) \varphi^T(N) P_S(N-1)}{1 + \varphi^T(N) P_S(N-1) \varphi(N)}, \\ P_S(N) = P(N) + \frac{P(N) \varphi(N-s) \varphi^T(N-s) P(N)}{1 - \varphi^T(N-s) P(N) \varphi(N-s)}, \\ p_s(N) = p_s(N-1) + \varphi(N) y(N) - \varphi(N-s) y(N-s), \\ c(N) = P_S(N) p_s(N). \end{cases}$$

При цьому співвідношення (5.19) можуть бути переписані у формі

$$\left\{ \begin{array}{l} \varphi_{0S}(k) = \frac{1}{\sqrt{s}}, \\ \varphi_{1S}(k) = \sqrt{\frac{3}{s(s^2-1)}}(2k-s-1), \\ \varphi_{lS}(k) = (A_l k + B_l)\varphi_{l-1,S}(k) - H_l \varphi_{l-2,S}(k), l = 2, 3, \dots, m, \\ A_l = \frac{2}{l} \sqrt{\frac{D_l(D_l-2)}{G_l}}, \\ B_l = -\frac{1+s}{l} \sqrt{\frac{D_l(D_l-2)}{G_l}} = -\frac{1+s}{l} A_l, \\ H_l = \frac{l-1}{l} \sqrt{\frac{D_l(D_l+G_l-2)}{(D_l-4)G_l}}, \\ D_l = 2l+1, \\ G_l = s^2 - l^2, s > m. \end{array} \right.$$

Як видно, система ортонормальних поліномів може бути розрахована заздалегідь і не коригуватися в процесі настройки адаптивної моделі.

Таким чином, задачу прогнозування коротких часових рядів з нерівновіддаленими спостереженнями пропонується вирішувати за допомогою адаптивної моделі, яка заснована на системі поліномів Чебишева, а також алгоритму ідентифікації на ковзному вікні. Використання даної моделі дозволяє не коректувати структуру поліномів в процесі налаштування параметрів моделі.

Відмінною особливістю запропонованої процедури прогнозування є те, що вона проста з погляду чисельної реалізації, дозволяє значно скоротити час на виконання операції, а також дає можливість обробляти істотно нестационарні процеси, що містять як нерегулярні тренди, так і раптові скачки.

5.3 Адаптивна поліноміальна нейромережева прогноуюча модель часових рядів і її навчання

Задача прогнозування часових рядів часто зустрічається в багатьох застосунках, а для її вирішення розроблено цілий арсенал методів від найпростіших емпіричних до дуже складних, заснованих на інтелектуальному підході. Можливість використання того чи іншого методу визначається обсягом апіорної інформації про розв'язувану задачу, при цьому задача істотно ускладнюється такими факторами як нелінійність, нестационарність, стохастичність, хаотичність досліджуваного явища. У цих умовах з найкращого боку себе проявили штучні нейронні мережі (ШНМ), завдяки високим апроксимується і екстраполюються властивостям і можливості перебудовувати свої характеристики безпосередньо по ходу рішення задачі.

Найбільш часто для вирішення задач прогнозування використовуються такі ШНМ, як багатошаровий перцептрон і його модифікації, що лежать в основі рекурентних нейронних мереж, а також радіально-базисні ШНМ [190]. Незважаючи на те, що ці мережі забезпечують високу якість передбачення, вони мають цілу низку недоліків, що обмежують їх використання при вирішенні багатьох практичних завдань. Так, багатошарові перцептрони, яких навчають на основі зворотного поширення помилок, досить повільно налаштовують свої параметри і вимагають великих обсягів навчальної вибірки. Радіально-базисні ШНМ навчаються з високою швидкістю завдяки тому, що їх вихідний сигнал лінійно залежить від настроюються синаптичних ваг, однак наявністю великої кількості вільних параметрів рецепторних полів радіально-базисних активаційних функцій змушує вирішувати додаткові задачі кластеризації, а крім того не слід забувати про досить часто виникаюче у реальних задачах «прокляття розмірності».

Гідною альтернативою багатошаровим і радіально-базисним ШНМ є поліноміальні нейронні мережі [191], що використовують як активаційних

функцій ті чи інші поліноми і мають цілий ряд переваг перед традиційними мережами.

В якості активаційних поліноміальних функцій вельми перспективним представляється використання ортогональних поліномів, що лежать в основі, так званих, ортогональних ШНМ [192-194], основною перевагою яких є простота навчання на основі методу найменших квадратів з діагональною ковариационної матрицею. А серед безлічі можливих ортогональних поліномів найбільш ефективними представляються поліноми Чебишева [189], що володіють цілим рядом корисних властивостей, пов'язаних з традиційним в навчанні ШНМ квадратичним критерієм.

На рис. 5.1 наведена архітектура поліноміальної нейронної мережі, що реалізує прогнозуючу модель нелінійної авторегресії з екзогенними входами (NARX-модель) виду

$$\hat{y}(k) = f(y(k-1), \dots, y(k-n_A), x(k-1), \dots, x(k-n_B))$$

або, переобозначивши змінні,

$$\hat{y}(k) = f(z_1(k), \dots, z_{n_A}(k), z_{n_A+1}(k), \dots, z_p(k), \dots, z_{n_A+n_B}(k)),$$

де $\hat{y}(k)$ – прогнозована часова послідовність в момент дискретного часу $k = 1, 2, \dots$;

$\hat{y}(k)$ – її прогноз за наявними на $(k-1)$ -й момент реальним спостереженням;

$x(k-1), \dots, x(k-n_B)$ – значення екзогенної (зовнішньої) змінної, що визначають поведінку прогнозованого ряду;

n_A, n_B – глибина використовуваної передісторії;

$z_p(k) = y(k-p)$, если $p \leq n_A$ и $z_p(k) = x(k-p)$ при $p > n_A$.

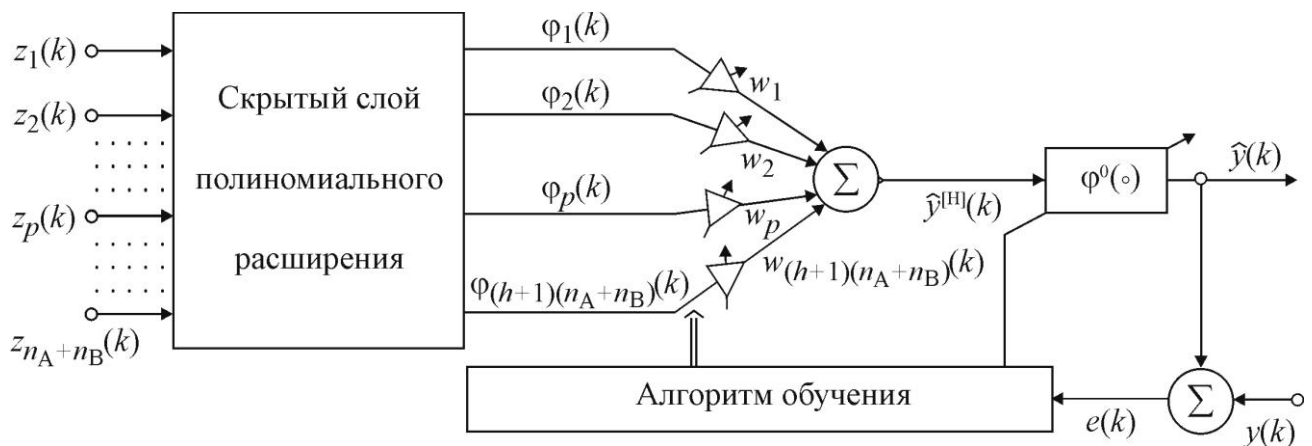


Рисунок 5.1 – Поліноміальна ортогональна нейронна мережа

Вхідні сигнали $z_1(k), \dots, z_p(k), \dots, z_{n_A+n_B}(k)$ з рецепторного шару подаються на перший прихований шар мережі, іменованій шаром полиномиального розширення, де реалізується нелінійне чебишовських перетворення

$$\begin{aligned}
 f_0(z_p(k)) &= 1, \\
 f_1(z_p(k)) &= z_p(k), \\
 f_2(z_p(k)) &= 2z_p^2(k), \\
 &\dots \\
 f_h(z_p(k)) &= 2z_p(k)f_{h-1}(z_p(k)) - f_{h-2}(z_p(k)),
 \end{aligned}
 \tag{5.20}$$

на основі якого вхідні сигнали $z_p(k)$, $p = 1, 2, \dots, n_A + n_B$ перетворюються так, що на виходах першого прихованого шару формуються сигнали $\phi_1(k) = f_0(z_1(k)), \dots, \phi_{(h+1)(n_A+n_B)}(k) = f_h(z_{n_A+n_B}(k))$. Другий прихований шар утворений адаптивним лінійним асоціатором [190] з $(h+1)(n_A+n_B)$ входами і такою ж кількістю настраюються синаптичних ваг $w = (w_1, w_2, \dots, w_{(h+1)(n_A+n_B)})^T$. На виході другого прихованого шару в

результаті формується скалярний сигнал

$$\mathcal{E}^{[H]}(k) = \sum_{l=1}^{(h+1)(n_A+n_B)} w_l \phi_l(k) = w^T \phi(k),$$

$$\text{де } \phi(k) = (\phi_1(k), \phi_2(k), \dots, J_{(h+1)(n_A+n_B)}(k))^T.$$

У ортогональній нейронній мережі, введеної в [194], замість адаптивного лінійного асоціатора пропонується використовувати елементарний перцептрон Розенблатта з фіксованою активаційною функцією гіперболічного тангенса

$$\mathcal{E}(k) = \varphi^0(\mathcal{E}^{[H]}(k)) = \tanh(\mathcal{E}^{[H]}(k)).$$

Зрозуміло, що єдина активаційна функція не може забезпечити ніякі додаткові апроксимуючі властивості, а сама крива гіперболічного тангенса може виконувати тільки роль обмежувача

$$-1 < \mathcal{E}(k) < 1.$$

При цьому, при довільному нелінійному перетворенні $\varphi^0(\circ)$, нейронна мережа в цілому виконує

$$\mathcal{E}(k) = \varphi^0(w^T \phi(k)) = \varphi^0 \left(\sum_{l=1}^{(h+1)(n_A+n_B)} w_l \phi_l(k) \right). \quad (5.21)$$

Для поліпшення апроксимуючих і екстраполюючих властивостей мережі ми пропонуємо формувати вихідний шар не на основі єдиної фіксованої активаційної функції, а на основі, так званого, орто-синапсу

[195], схема якого наведена на рис. 5.2.

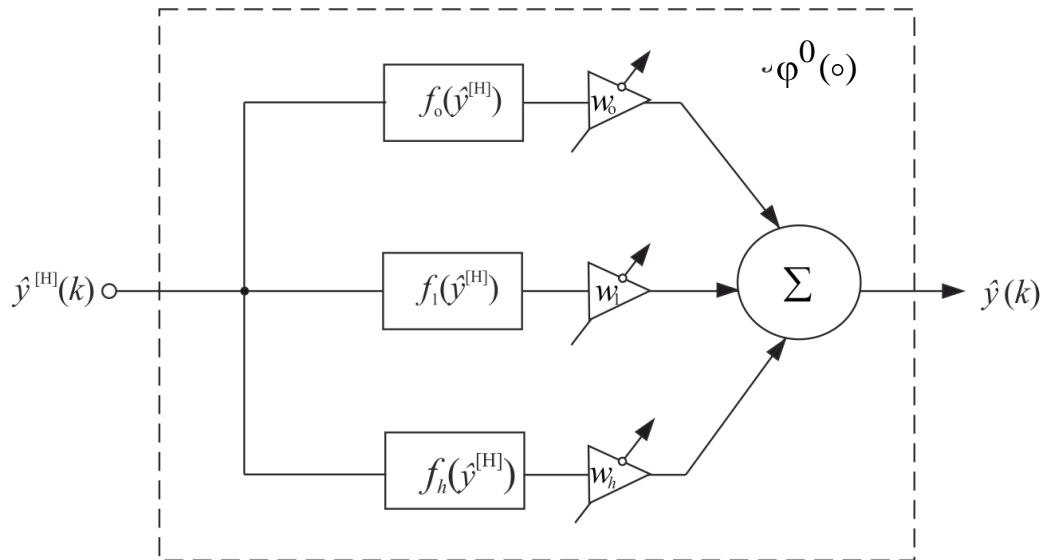


Рисунок 5.2 – Орто-синапс вихідного шару поліноміальної нейронної мережі

В якості активаційної функції тут використовуються ті ж поліноми Чебишева (5.20), а крім того є $h+1$ настроюються синаптичних ваг, що покращує якість рішення розглянутої задачі.

Перетворення, що реалізується орто-синапсом, може бути записано у вигляді

$$\mathfrak{f}(k) = \sum_{j=0}^h w_j^0 f_j^0(\mathfrak{y}^{[H]}(k)), \quad (5.22)$$

а відображення (5.25), реалізоване мережею в цілому:

$$\mathfrak{f}(k) = \sum_{j=0}^h w_j^0 f_j^0 \left(\sum_{l=1}^{(h+1)(n_A+n_B)} w_l \varphi_l(k) \right). \quad (5.23)$$

Така мережа містить $(h+1)(n_A+n_B)+h+1$ настроюваних в процесі навчання синаптичних ваг.

Навчання мережі проводиться на основі зворотного поширення

помилки [92], при цьому спочатку налаштовуються ваги $w^0 = (w_1^0, \dots, w_h^0)^T$ вихідного шару, а потім вектор ваг другому прихованого шару. Перший прихований шар мережі є непараметричним і не містить параметрів, що настроюються.

Для налаштування вектора ваг вихідного шару w^0 доцільно скористатися оптимальними за швидкістю алгоритмами, наприклад, адитивно-мультиплікативної модифікацією алгоритму Качмажа [53]

$$\begin{aligned} w^0(k) &= w^0(k-1) + \gamma \frac{y(k) - \hat{y}(k)}{\beta + \|F(k)\|^2} F(k) = w_0(k-1) + \gamma \frac{e(k)F(k)}{\beta + \|F(k)\|^2} = \\ &= w_0(k-1) + \gamma \frac{y(k) - w^{0T}(k-1)F(k)}{\beta + \|F(k)\|^2} F(k), \end{aligned} \quad (5.24)$$

де $F(k) = f_0^0(\mathcal{E}^{[H]}(k)), f_1^0(\mathcal{E}^{[H]}(k)), \dots, f_h^0(\mathcal{E}^{[H]}(k))^T$,

$0 < \gamma < 2$, $\beta \geq 0$ – параметри алгоритму, обрані з емпіричних міркувань.

Тут важливо зазначити, що, оскільки вихідний сигнал мережі $\hat{y}(k)$ лінійно залежить від вектора синаптичних ваг вихідного шару w^0 , їх налаштування може бути проведена максимально швидко.

Оскільки в реальних задачах аналізований ряд $y(k)$ «забруднений» різними збуреннями і перешкодами, для їх фільтрації замість однокрокового алгоритму (5.24) можна скористатися багатокроковими процедурами типу методу найменших квадратів на ковзному вікні. Тоді $w^0(k)$ оцінка на ковзному вікні зі s спостережень може бути записана у вигляді

$$w^0(k) = \sum_{\alpha=k-s+1}^k F(\alpha)F^T(\alpha)^{-1} \sum_{\alpha=k-s+1}^k F(\alpha)y(\alpha)$$

або в рекуррентній формі:

$$\begin{cases} P(k) = P_s(k-1) - \frac{P_s(k-1)F(k)F^T(k)P_s(k-1)}{1 + F^T(k)P_s(k-1)F(k)}, \\ P_s(k) = P(k) + \frac{P(k)F(k-s)F^T(k-s)P(k)}{1 - F^T(k-s)P(k)F(k-s)}, \\ p_s(k) = p_s(k-1) + F(k)y(k) - F(k-s)y(k-s), \\ w^0(k) = P_s(k)p_s(k). \end{cases} \quad (5.25)$$

Синаптичні ваги другого прихованого шару в опис мережі (5.23) входять нелінійно, отже, для їх налаштування необхідно використовувати алгоритми, засновані на зворотному поширенні помилок.

Введемо в розгляд цільову функцію навчання

$$\begin{aligned} J(k) &= \frac{1}{2}e^2(k) = \frac{1}{2}(y(k) - \mathfrak{f}(k))^2 = \frac{1}{2}\left(y(k) - \sum_{j=0}^h w_j^0 f_j^0(\mathfrak{f}^{[H]}(k))\right)^2 = \\ &= \frac{1}{2}\left(y(k) - w^{0T}F(k)\right)^2 = \frac{1}{2}\left(y(k) - \sum_{j=0}^h w_j^0 f_j^0\left(\sum_{l=1}^{(h+1)(n_A+n_B)} w_l \varphi_l(k)\right)\right)^2 = \\ &= \frac{1}{2}\left(y(k) - w^{0T}F(w^T\varphi(k))\right)^2, \end{aligned}$$

яка може бути мінімізована по w_l за допомогою градієнтного алгоритму

$$\begin{aligned} w_l(k) &= w_l(k-1) - \eta(k) \frac{\partial J(k)}{\partial w_l} = w_l(k-1) + \eta(k) e(k) \frac{\partial \mathfrak{f}(k)}{\partial w_l} = \\ &= w_l(k-1) + \eta(k) e(k) \sum_{j=0}^h w_j^0(k) \frac{\partial f_j^0(\mathfrak{f}^{[H]}(k))}{\partial \mathfrak{f}} \varphi_l(k), \end{aligned}$$

або у векторній формі:

$$w(k) = w(k-1) + \eta(k)e(k)w_0^T(k) \left(\frac{\partial \vec{F}(k)}{\partial \mathfrak{f}} \right) \varphi(k), \quad (5.26)$$

де $\eta(k) > 0$ – параметр кроку навчання,

$$\left(\frac{\partial \vec{F}(k)}{\partial \mathfrak{f}} \right) - (h+1) \times 1 - \text{ вектор, утворений приватними похідними}$$

$$\left(\frac{\partial f_j^0(\mathfrak{f}^H k)}{\partial \mathfrak{f}} \right).$$

Алгоритм (5.26) може бути також записаний у більш звичній в навчанні нейронних мереж формі δ -правила навчання [92]:

$$w(k) = w(k-1) + \eta(k)\delta(k)\varphi(k),$$

де $\delta(k) = e(k)w_0^T(k) \left(\frac{\partial \vec{F}(k)}{\partial \mathfrak{f}} \right)$ – так звана локальна помилка (δ -помилка)

навчання багатошарової мережі.

Необхідно відзначити також, що в силу нелінійної залежності виходу мережі $\mathfrak{f}(k)$ від синаптичних ваг $w(k)$ другого прихованого шару, швидкість збіжності алгоритму (5.25) істотно нижче, ніж швидкість процедури (5.24), тому для навчання нейронної мережі в цілому можна використовувати навчання за епохами, вживане в навчанні багатошарових мереж. Тим не менше, оскільки навчається тільки один прихований шар, розглянута нейромережева модель все одно налаштовується швидше, ніж стандартний тришаровий персептрон. В роботах [196-198] розглянуті експериментальні результати запропонованих методів. Застосування методів прогнозування зводиться до пошуку ситуацій, коли кадр, що спостерігається, представлений в просторі ознак або власне в просторі зображень, відрізняється від прогнозу,

що і означає зміну властивостей відеоряду і забезпечує його структуризацію. Специфіка сегментації відеорядів полягає в необхідності оперування зі зростаючими вибірками, що в підсумку дозволяє здійснювати аналіз в режимі on-line. Приклади пошуку границь різних відеосегментів наведені на рис. 5.3-5.4.

Таким чином, запропонована адаптивна поліноміальна нейромережева прогноуюче модель нестационарних часових рядів і процедура її навчання, засновані на використанні ортогональних поліномів в якості активаційних функцій, що забезпечує обчислювальну простоту її реалізації, забезпечує високу швидкість навчання, дозволяє ускладнювати архітектуру нейронної мережі без необхідності перерахунку вже налаштованих синаптичних ваг.



Рисунок 5.3 – Приклад границь сегменту, кадри 537-540 відеоряду «Tsunami»

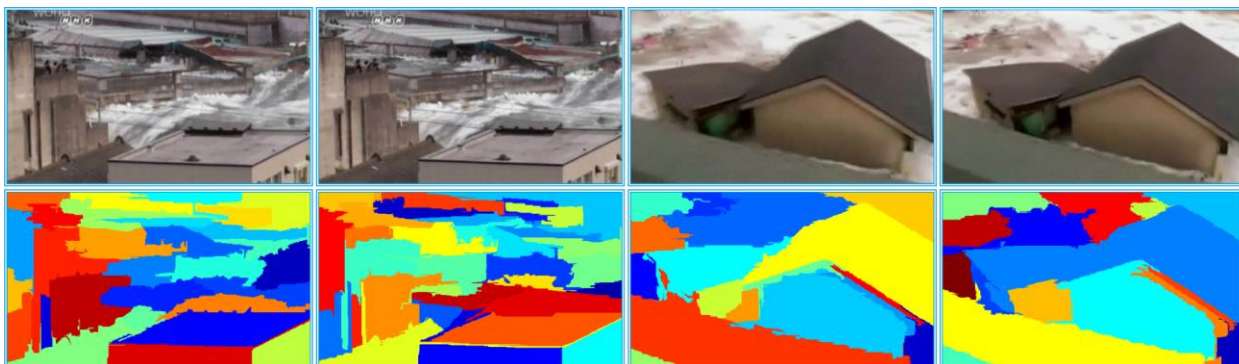


Рисунок 5.4 – Приклад границь відеосегменту і сегментовані кадри 575, 576, 578,579 відеоряду «Tsunami»

5.4 Розробка системи підтримки прийняття рішень для аналізу, класифікації та прогнозування фрактальних часових рядів

Численні дослідження, проведені в останні десятиліття, показали, що багато інформаційних, біологічних, фізичних, технологічних процесів мають складну фрактальну структуру. Такі процеси вивчаються самими різними науками - від розподілу галактик в астрономії до формування хмар, клімату і погоди, передбачення землетрусів у метеорології; від полімеризації і окислення в хімії до структури легень і судинної системи людини, у фізіології зорової системи і картин зростання багатьох рослин в біології; від «язиків», що заважають видобутку нафти і структури притоків, областей водозбору річкових систем, частоти повеней в геофізиці до власне фізики, де фрактали і статистична самоподібність зустрічаються в броунівському русі, перколяції, в дослідженнях турбулентності і в інших явищах.

Усе вищезазначене призвело до появи ряду моделей фрактальних стохастичних процесів. До їх характерних особливостей відносяться високі піки і «важкі хвости» розподілів, довгострокова залежність, умовна гетероскедастичність та інші. Слід відзначити відсутність універсальних моделей, які могли б використовуватися для опису фрактальних процесів різної природи. І навпаки, один і той же процес, залежно від цілей дослідження, може описуватися декількома моделями. У загальному випадку вибір моделі базується на характеристиках досліджуваного часового ряду (ЧР). Проте ще не існує єдиного універсального методу визначення фрактальних характеристик для ЧР і єдиного підходу до проведення досліджень, які повинні передувати проведенню фрактального аналізу.

На даний час все більше застосування знаходять методи інтелектуального аналізу ЧР, спрямовані на виявлення закономірностей у динаміці систем на тлі випадковостей, виявлення прихованих кореляцій і неперіодичних циклів, що дозволяє робити обґрунтовані висновки і прогнози. Одним із способів реалізації методів аналізу та прогнозування ЧР є

розробка системи підтримки прийняття рішень (СППР). В останні роки запропоновані і розроблені СППР для ЧР, що володіють різними характерними особливостями, зокрема різними нестационарними властивостями. Проте, не дивлячись на те, що фрактальний аналіз знаходять все більш практичне застосування при аналізі та прогнозуванні ЧР, розробці математичних моделей та проведенні імітаційного моделювання в найрізноманітніших галузях науки, техніки і медицини, ще не розроблені СППР, що реалізують моделі та методи фрактального аналізу. Таким чином, актуальною є задача розробки СППР, яка дозволяє користувачеві досліджувати структуру фрактальних ЧР і класифікувати їх або вибрати кращу модель з деякої безлічі для аналізу та прогнозування.

5.4.1 Опис СППР, аналіз фрактальних і мультифрактальних характеристик ряду

Системи підтримки прийняття рішень – це прикладні системи штучного інтелекту, в яких база знань являє собою формалізовані емпіричні знання висококваліфікованих фахівців в якій-небудь вузькій предметній області. Зазвичай СППР містить наступні компоненти: механізм логічного виводу (інтерпретатор, вирішувач), робочу базу даних (БД), базу знань (БЗ), підсистему придбання та поповнення знань, підсистему пояснення і підсистему діалогу.

Механізм логічного виводу (МВ) призначений для отримання нових фактів на основі зіставлення вихідних даних з БД (часові ряди і опис предметної області) і знань з БЗ (даних, що описують властивості фрактальних процесів). Для цього зазвичай використовується програмно реалізований механізм дедуктивного логічного виводу. МВ в усій структурі СППР займає найбільш важливе місце. Розглянемо основні етапи роботи МВ в СППР, що дозволяє досліджувати структуру фрактального ЧР і вибрати

найбільш відповідну модель стохастичного процесу. На рис. 5.5 представлена блок-схема основних етапів функціонування МВ.

На першому етапі (блок 1) необхідно провести ряд досліджень, перед виконанням фрактального аналізу деякого випадкового процесу, представленого тимчасовим рядом $X(t)$ [199, 200]:

1. З апріорно відомої інформації необхідно з'ясувати, чи є ЧР кумулятивним (наприклад, курс валюти) чи ні (наприклад, ряд волатильності).

2. Визначити інтервали різних скейлінгів. Якщо ЧР володіє декількома скейлінгами, залежними від часових інтервалів (наприклад, денні та годинні дані для валютних рядів), то на кожному такому інтервалі динаміка ЧР $X(t)$ визначається відповідним показником Херста.

3. Виявити і видалити короткострокову авторегресійну залежність. Наявність авторегресійної залежності зміщує значення показника Херста і демонструє помилкову довготривалу пам'ять. Тому для коректного оцінювання ступеня самоподібності короткострокову залежність необхідно видалити відповідно до формули $S(t) = X(t) - (a + b \cdot X(t-1))$. Якщо вихідний ряд мав довгострокову залежність, то вона зберігається (показник Херста), у той час як короткострокова залежність усувається. Якщо авторегресійна залежність є значущою, то подальші етапи фрактального аналізу проводяться для залишкового часового ряду.

Усі результати попередніх досліджень, у тому числі в графічному вигляді, надаються користувачеві за допомогою підсистеми пояснення. Найчастіше ці результати можуть мати самостійну значимість для подальшого аналізу та прогнозування ЧР.

Наступним етапом (блок 2) є виявлення наявності фрактальних властивостей ЧР. Якісної перевіркою наявності властивостей статистичної самоподібності є побудова агрегованих часових рядів, відповідних досліджуваного ряду $X(t)$ і побудова для них вибірових функцій розподілу.

У разі самоподібності агреговані ряди мають однаковий розподіл, що підтверджується статистичними критеріями. Після цього виконується оцінювання ступеня самоподібності (показника Херста H). Якщо значення H близько до 0,5, необхідно перевірити гіпотезу про наявність самоподібності відповідними методами.

При відсутності фрактальних властивостей ЧР користувачеві пропонується проводити подальший аналіз і моделювання ЧР методами, які не використовують фрактальний аналіз. Підсистема пояснення пояснює хід міркувань, і які знання були при цьому використані.

Наступним етапом є визначення, чи має ЧР монофрактальні або мультифрактальні властивості (блок 3). Розрізнення властивостей моно- і мультифрактальних по часових рядів малої довжини є непростим завданням. В роботі [201] запропонований метод, що дозволяє приймати або відкидати гіпотезу про наявність монофрактальних властивостей для часового ряду. В основі методу лежать дослідження статистичних характеристик вибіркового значень узагальненого показника Херста $h(q)$, які отримані методом мультифрактального детрендорованого флуктуаційного аналізу. Значення випадкової величини $\Delta h = h(q1) - h(q2)$ можна використовувати в якості статистичного критерію для прийняття гіпотези про монофрактальність ряду. Користувачеві надаються результати досліджень, у тому числі ймовірність прийняття гіпотези.

Після цього виконується розрахунок статистичних характеристик (блок 4), необхідних для вибору і побудови математичних моделей (блок 5), що володіють властивостями монофрактальності (фрактальний броунівський рух, фрактальні точкові процеси, фрактальний процес FARIMA та ін.) і мультифрактального (фрактальний стійкий рух Леві, стохастичні мультиплікативні каскади, фрактальное рух в мультифрактального часу та ін.). Користувачеві надаються рекомендовані моделі, яких може бути

декілька. В рамках СППР можливе проведення тестування і перевірка адекватності моделей (блок 6).

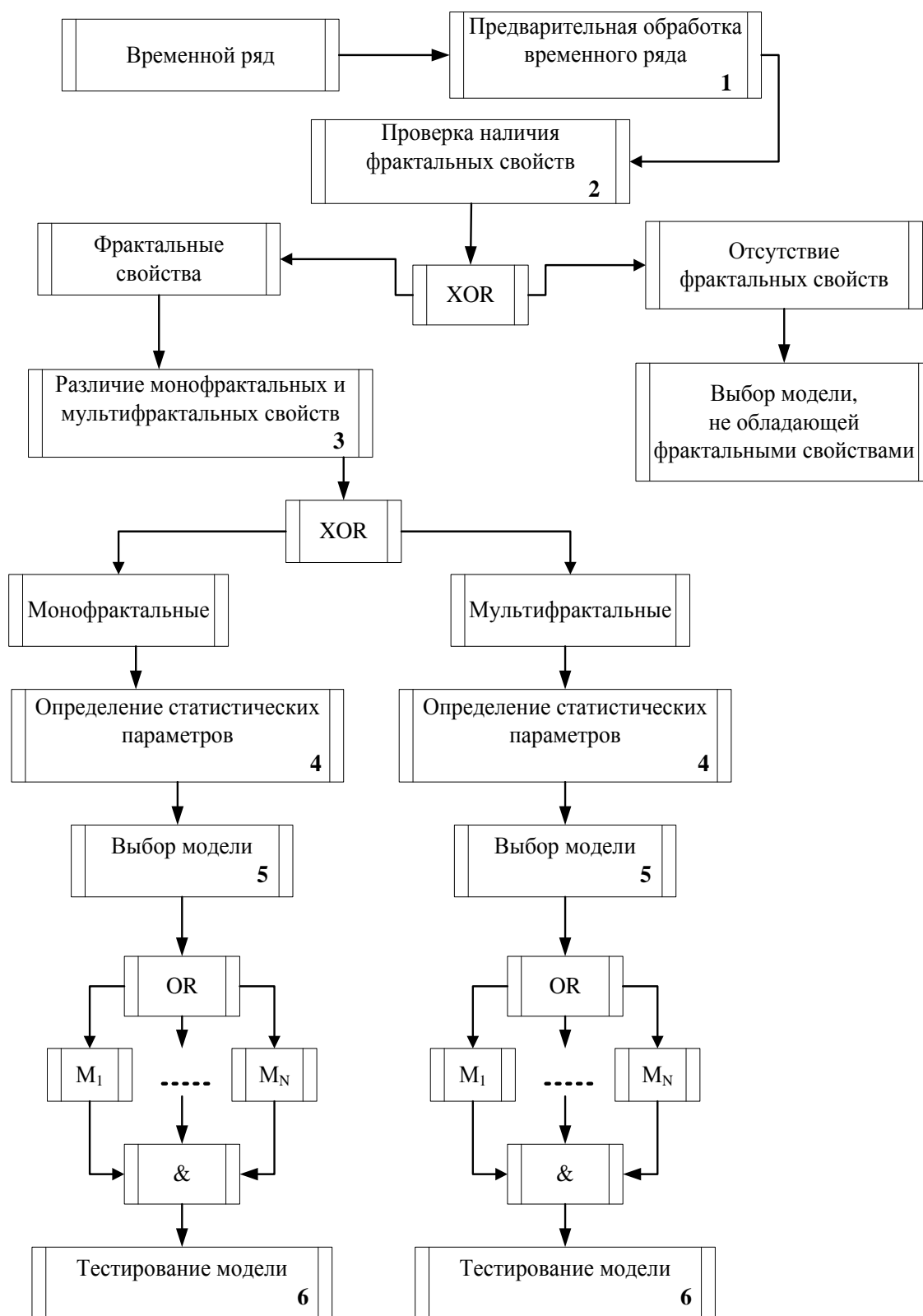


Рисунок 5.5 – Блок-схема основных этапов функционирования МВ

5.4.2 Аналіз інформаційних характеристик ряду

СППР призначена для підтримки багатокритеріальних рішень у складному інформаційному середовищі. При цьому під многокритеріальністю розуміється той факт, що результати прийнятих рішень оцінюються не по одному, а по сукупності багатьох показників (критеріїв) розглянутих одночасно. Інформаційна складність визначається необхідністю врахування великого обсягу даних, обробка яких без допомоги сучасної обчислювальної техніки практично нездійсненна. У цих умовах число можливих рішень, як правило, дуже велике, і вибір найкращого з них "на око", без всебічного аналізу може призводити до грубих помилок.

Головною функцією СППР є вивід рекомендацій на підставі вивчення історичного та поточного стану об'єкта дослідження та порівняння їх з інформацією, що зберігається в БЗ системи. Одним з головних елементів системи є БЗ - сховище даних, що описують властивості і характеристики самоподібних і мультифрактальних процесів.

З точки зору архітектури СППР знання в БЗ мають наступну структуру: інтерпретовані знання (використовуються вирішувачем) і не інтерпретовані знання, необхідні для підсистеми пояснень та діалогу. Інтерпретовані знання в БЗ запропонованої СППР можна умовно розділити на два блоки: стандартні статистичні характеристики ЧР (спектр, кореляційна функція, моменти різних порядків та ін.) і фрактальні характеристики (показник Херста, мультифрактальний спектр, тяжкість хвоста розподілу та ін.). Для більш якісного дослідження та класифікації ЧР в базу знань додано блок знань, що визначає інформаційну складність системи (рис. 5.6).

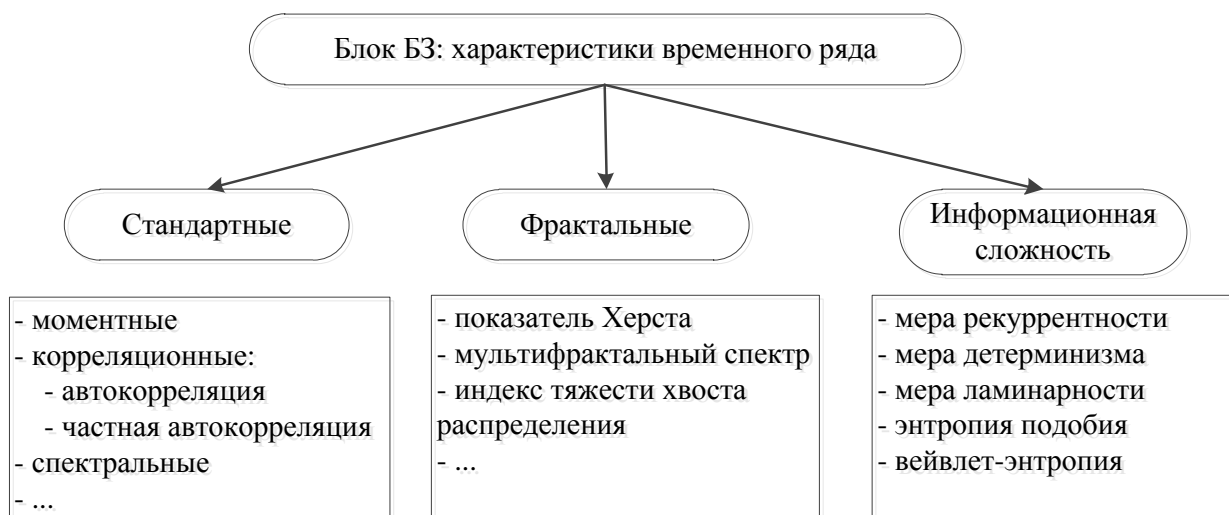


Рисунок 5.6 – Блок характеристик інформаційної складності

Існує безліч підходів щодо дослідження інформаційної складності ЧР, які базуються як на традиційному статистичному аналізі, так і на методах нелінійної хаотичної динаміки. В даний час основними характеристиками складності динаміки систем можна вважати різні типи ентропії і міри складності рекурентних діаграм [202].

Рекурентний аналіз базується на фундаментальній властивості дисипативних динамічних систем - рекурентності (повторюваності станів). Даний метод аналізу, заснований на представленні властивостей процесів у вигляді геометричних структур, є інструментом для виявлення прихованих залежностей в ЧР, які спостерігаються. Кількісними мірами складності рекурентних діаграм є міри детермінізму, рекурентності, ламінарності, тренда і т.д. [203-205]. Характеристикою складності поведінки системи традиційно є ентропія. Існує різні типи ентропії: ентропія подібності, ентропія шаблонів, багатомасштабна ентропія, вейвлет-ентропія та ін. Розрахунок ентропії і побудова рекурентних діаграм засновані на методах нелінійної динаміки, зокрема, процедурі Паккарда-Такенса, що дозволяє відновити фазову траєкторію системи по одній реалізації [206, 207].

Головна ідея застосування методів нелінійної динаміки до аналізу траєкторії динамічної системи полягає в тому, що основна структура, яка

містить в собі всю інформацію про систему, а саме, атрактор системи, може бути відновлено через вимір тільки однієї компоненти цієї системи. Широко використовувана процедура Паккарда-Такеенса дозволяє відновити фазову траєкторію динамічної системи по одній реалізації:

$$F(t) = [x(t), x(t + \tau), \dots, x(t + m\tau)],$$

де: $F(t)$ – m -вимірний псевдо фазовий простір, $x(t)$ – часова реалізація системи, τ – період запізнювання.

Рекурентна діаграма є проекцією m -мірного псевдофазового простору на площину. Нехай точка x_i відповідає точці фазової траєкторії $x(t)$, яка описує динамічну систему в m -мірному просторі в момент часу $t = i$, для $i = 1, \dots, N$, тоді рекурентна діаграма RP є масив точок, де ненульовий елемент з координатами (i, j) відповідає випадку, коли відстань між x_j і x_i менше ε :

$$RP_{i,j} = \Theta(\varepsilon - \|x_i - x_j\|), \quad x_i, x_j \in R^m, \quad i, j = 1, \dots, N,$$

де ε - розмір околу точки x_i , $\|x_i - x_j\|$ - відстань між точками, $\Theta(\cdot)$ - функція Хевісайда.

Аналіз топології діаграми дозволяє класифікувати процеси, які спостерігаються: визначати однорідні процеси з незалежними випадковими значеннями; процеси з параметрами, які повільно змінюються періодичні чи осцилюючі процеси, які відповідають нелінійним системам і т.д. Чисельний аналіз рекурентних діаграм дозволяє обчислювати міри складності структур рекурентних діаграм, такі як міра рекурентності, міра детермінізму, міра ентропії та ін. Міра рекурентності RR показує щільність рекурентних точок:

$$RR = \frac{1}{N^2} \sum_{i,j} RP_{i,j}. \quad \text{Міра детермінізму } Det \text{ є характеристикою}$$

передбачуваності поведінки процесу і дорівнює відношенню числа рекурентних точок, складових діагональні лінії, до загальної кількості

рекурентних точок: $Det = \sum_{l=l_{\min}}^N P(l) / \sum_{i,j} RP_{i,j}$, де: l_i - довжина i -ї

діагональної лінії, $P(l) = l_i, i=1, \dots, N_l$ - частотний розподіл довжин діагональних ліній в діаграмі, N_l - кількість діагональних ліній.

Ентропія подібності $ApEn$ є статистикою регулярності часового ряду, що визначає можливість його прогнозування [208, 209]. Розглянемо часовий ряд $\{x_i\}, i=1, \dots, N$. Нехай вектор $P_m(i)$ - підпоследовність значень ряду $\{x_i, x_{i+1}, \dots, x_{i+m}\}$ завдовжки m . Два вектора $P_m(i)$ і $P_m(j)$ будуть подібними, якщо виконується $|x_{i+k} - x_{j+k}| < \varepsilon, 0 \leq k < m$.

Для кожного значення $i=1, \dots, N-m+1$ обчислюється величина

$C_{im}(\varepsilon) = \frac{n_{im}(\varepsilon)}{N-m+1} f$, де: $n_{im}(\varepsilon)$ - число векторів, подібних вектору $P_m(i)$.

Ентропія подібності $ApEn$ визначається за формулою

$$ApEn(m, \varepsilon) = \ln \frac{C_m(\varepsilon)}{C_{m+1}(\varepsilon)}, \quad C_m(\varepsilon) = \frac{\sum_{i=1}^{N-m+1} C_{im}(\varepsilon)}{N-m+1}.$$

Одним із потужних інструментів дослідження та класифікації часових рядів є вейвлет-аналіз, зокрема кратномасштабного аналіз, що дозволяє проводити декомпозицію часового ряду на складові з різними частотними діапазонами [210, 211]. Використання вейвлет-характеристик в якості знань для СППР дає можливість розпізнавання характерних особливостей фрактальних сигналів.

Дискретне вейвлет-перетворення (ДВП) будується за допомогою кратномасштабного аналізу, основна ідея якого полягає в розбитті

досліджуваного сигналу $X(t)$ на дві складові: апроксимуючу та деталізуючу, з подальшим аналогічним дробленням апроксимуючої до заданого рівня декомпозиції сигналу N . У результаті ДВП сигнал $X(t)$ представляється у вигляді:

$$X(t) = \sum_{k=1}^{N_a} apr(N,k) \varphi_{Nk}(t) + \sum_{j=1}^N \sum_{k=1}^{N_j} det(j,k) \psi_{jk}(t),$$

де $\psi_{jk}(t)$ – вейвлет-функція, $\varphi_{Nk}(t)$ – відповідна скейлінг-функція, $apr(N,k)$ – апроксимуючі вейвлет-коефіцієнти рівня N , $det(j,k)$ – деталізуючі вейвлет-коефіцієнти рівня j , N_j – кількість деталізуючих коефіцієнтів на рівні j , N_a – кількість апроксимуючих коефіцієнтів на рівні N .

Величина вейвлет-енергії на заданому рівні вейвлет-розкладання j визначається як $E_j = \frac{1}{N_j} \sum_{k=1}^{N_j} det^2(j,k)$. Набір величин E_j для кожного рівня розкладання становить спектр вейвлет-енергії ряду. Повна вейвлет-енергія спектра являє собою суму енергій кожного рівня $E_{tot} = \sum_{j=1}^N E_j$. Відносна

вейвлет-енергія показує розподіл енергії за рівнями розкладання: $p_j = \frac{E_j}{E_{tot}}$.

Вейвлет-ентропія WE є кількісною мірою впорядкованості сигналу і визначається за формулою $WE = - \sum_{j=1}^N p_j \ln(p_j)$.

5.4.3. Вхідні модельні дані

Дослідження біологічних сигналів може бути проведено за допомогою методів, які розвинуто в теорії динамічного хаосу при припущеннях, що сигнал є реалізацією деякої нелінійної динамічної системи, чутливої до початкових умов. Таким чином, передбачається, що біоелектрична активність описується неявно заданою хаотичною системою і, незважаючи на відсутність системи рівнянь, що моделюють біоелектричну активність, можна вивчати поведінку системи за її вихідним даним.

Хаос являє собою складну форму поведінки детермінованої системи в сталому режимі. Основною властивістю таких систем є чутлива залежність режиму функціонування до як завгодно малих змін початкових умов. Ця обставина веде до втрати детермінованої передбачуваності та хаотичної поведінки. Кількісною мірою хаотичності системи є показник Ляпунова λ , що характеризує швидкість розбігання спочатку близьких траєкторій.

Одними з найпростіших і наочних математичних моделей, що демонструють хаотичну поведінку, є ітерувемі відображення. Найбільш відомим прикладом хаотичних відображень є логістичне відображення. Це одномірне квадратичне відображення, яке визначається наступним чином $x_{n+1} = Ax_n(1 - x_n)$, де A - керуючий параметр, $A \in (0, 4]$, а значення $x_n \in [0, 1]$.

Розглянемо реалізації даного відображення при різних хаотичних режимах, які визначаються показником Ляпунова λ . На рис. 5.7 показані реалізації при значеннях керуючого параметра $A = 3.7$ і 4 . Відповідні значення показника Ляпунова рівні $\lambda = 0.37, 0.69$. Більше значення показника Ляпунова відповідає більшій мірі хаотичності системи.

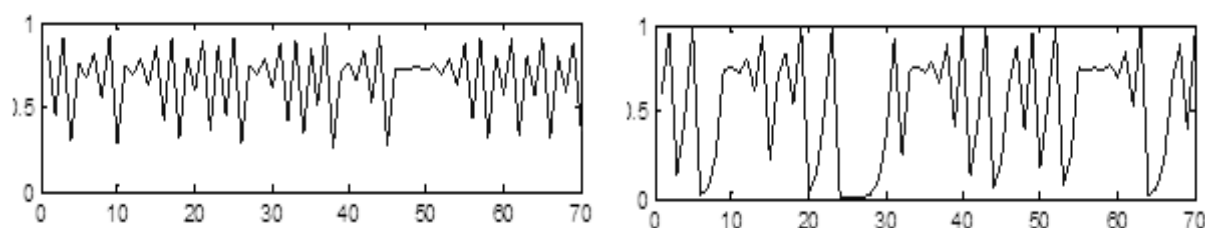


Рисунок 5.7 – Реалізація логістичного відображення

5.4.4 Експериментальні фізіологічні дані

Реалізації електроенцефалограм. У роботі були досліджені записи електроенцефалограм (ЕЕГ) лабораторних тварин, які були розбиті на фази бадьорості та сну. На рис. 5.8 представлені типові реалізації ЕЕГ у фазі повільного сну (sws) і бадьорості (awake).

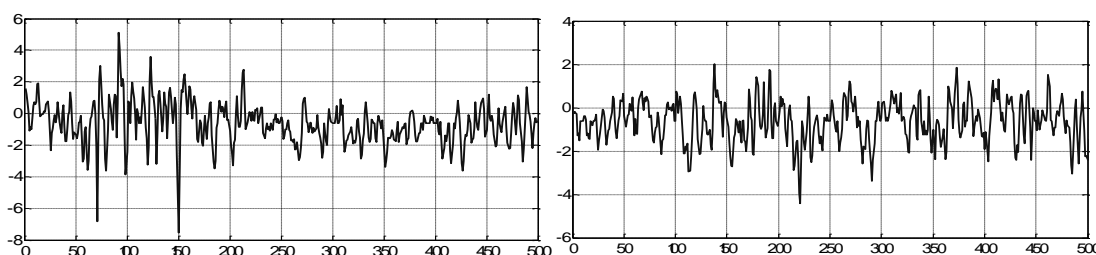


Рисунок 5.8 – Реалізація ЕЕГ sws та awake

Реалізації RR-інтервалів. Одним із методів діагностики серцевих захворювань, є аналіз рядів, побудованих за RR-інтервалам. RR-інтервал являє собою проміжок часу між сусідніми зубцями електрокардіограми і дорівнює тривалості серцевого циклу. Вихідні дані були отримані на спеціалізованому сайті [212]. В ході медичного дослідження розглядалися пацієнти, що мають серцеву аритмію. Наведено дані RR-інтервалів до і після прийому медичного препарату, що застосовується для лікування і запобігання тахікардії. На рис. 5.9 наведені типові послідовності RR-інтервалів до і після застосування препарату.

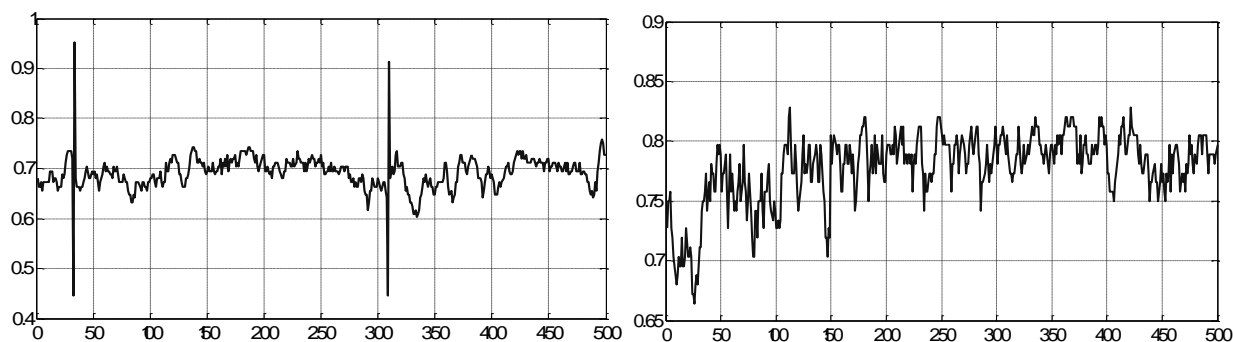


Рисунок 5.9 – Реалізації RR-інтервалів до (ліворуч) та після (праворуч) прийому препарату

5.4.5 Результати досліджень

Результати досліджень опубліковані в роботах [213-215].

Модельні сигнали. Проведений в роботі аналіз виявив сильні відмінності, як у візуальній топології, так і в числових характеристиках реалізацій вищеописаних процесів. Для наочності розглянемо приклад абсолютно різних за складністю процесів: гармонійного сигналу і некорельованого білого шуму. Очевидно, що характеристики вищеописаних процесів повинні перебувати всередині діапазону значень характеристик, розрахованих для періодичних і повністю випадкових траєкторій. На рис. 5.10 наведені рекурентні діаграми для реалізацій синусоїди і незалежних значень нормальної випадкової величини.

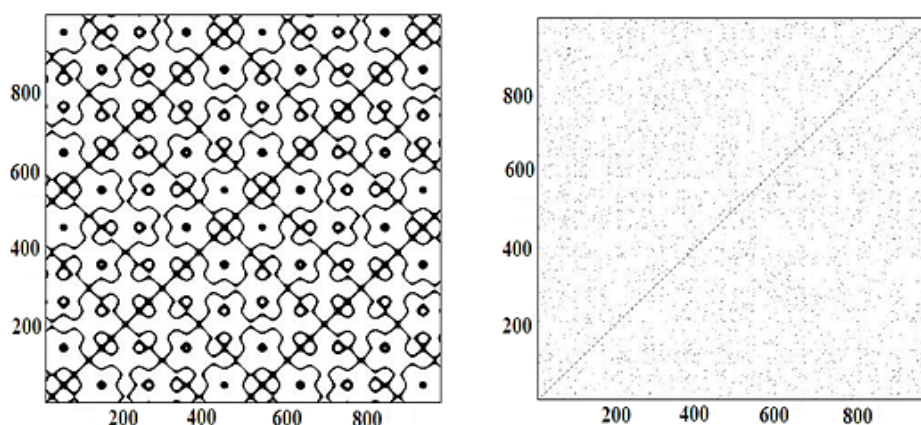


Рисунок 5.10 – Рекурентні діаграми для гармонійного сигналу (ліворуч) і некоррелірованими шуму (праворуч)

Рис.5.11 демонструє відповідні спектри вейвлет-енергії реалізацій. Очевидно, що для гармонійного сигналу основна енергія зосереджена на одному рівні декомпозиції. У разі некоррельованого шуму вейвлет-енергія реалізації розподіляється по частотах досить рівномірно.

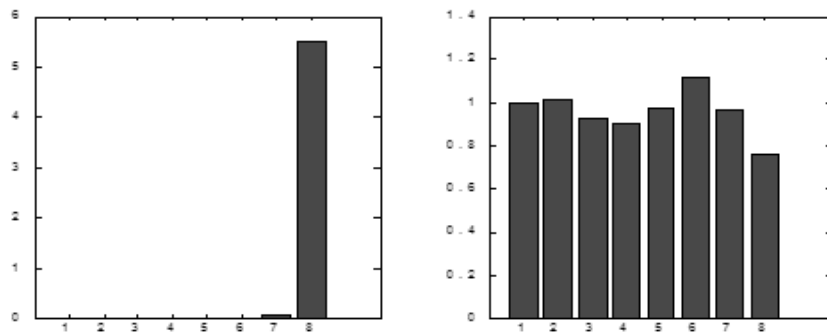


Рисунок 5.11 – Спектр вейвлет-енергії гармонійного сигналу (ліворуч) і некорельованого шуму (праворуч)

У таблиці 5.1 наведені середні значення міри RR рекурентності, міри детермінізму Det , ентропії подібності $ApEn$ і вейвлет- ентропії W , які розраховані для реалізацій довжиною 1000 значень.

Таблиця 5.1 - Характеристики складності найпростіших сигналів

Процес	$ApEn$	Det	RR	W
Синусоїда	0.03	0.998	0.03	0.62
Некорельований шум	1.7	0.025	1.7	1,7

Розглянемо реалізації логістичного відображення, приклади яких були наведені на рис.5.7. На рис. 5.12 показані відповідні рекурентні діаграми. Більше значення показника Ляпунова відповідають більшій хаотичності

системи, що наочно проявляється на рекурентних діаграмах: наявність певної структури діаграми змінюється рівномірним заповненням.

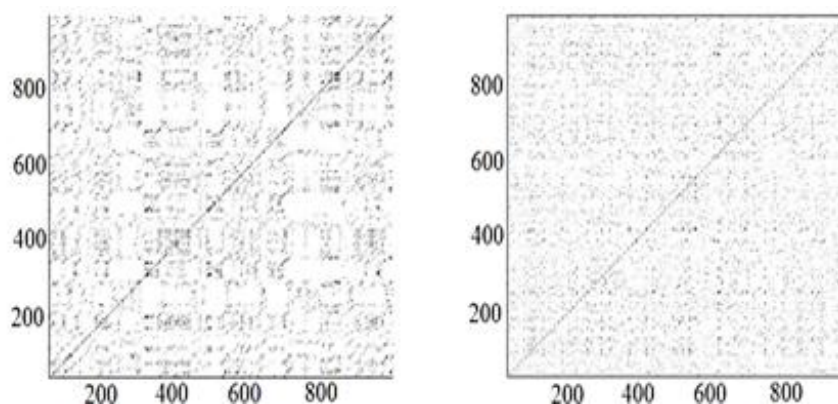


Рисунок 5.12 – Рекурентні діаграми для реалізацій відображення при $\lambda=0.37$ (ліворуч) та $\lambda=0.69$ (праворуч)

На рис.5.13 показані спектри вейвлет-енергії логістичного відображення при різних значеннях керуючого параметра. Очевидно, що при менших значеннях показника Ляпунова основна енергія процесу зосереджена в високочастотних компонентах (початкові рівні декомпозиції). У разі $\lambda = 0.69$ дана система досягає максимального рівня хаотичності і вейвлет-енергія реалізацій розподіляється по частотах досить рівномірно.

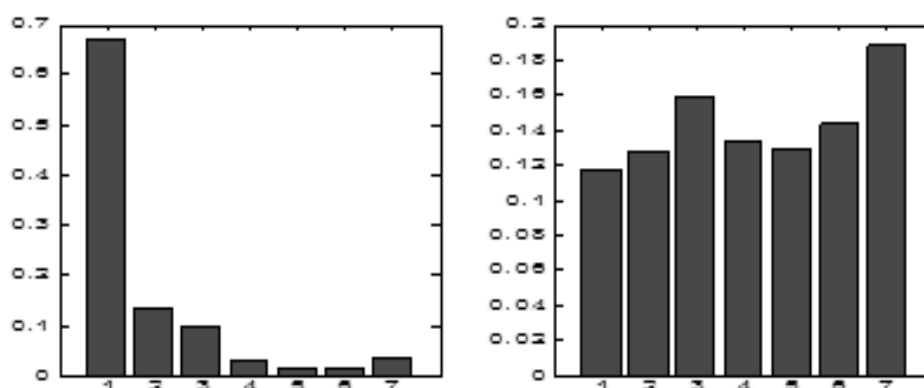


Рисунок 5.13 – Спектри вейвлет-енергії для відображення при $\lambda=0.37$ (ліворуч) та $\lambda=0.69$ (праворуч)

У таблиці 5.2 наведено середні значення характеристик, отриманих за рекурентним діаграмами: міра детерменизму Det , міра рекурентності RR , ентропія подібності $ApEn$ і вейвлет-ентропія WE .

Таблиця 5.2 - Характеристики складності для хаотичного відображення

A	λ	WE	ApEn	RR	Det
3.7	0.37	1.22	0.35	0.008	0.1
4	0.69	1.86	0.62	0.002	0.05

Виходячи з отриманих результатів можна відмітити, що зі зростанням хаотичності або некорельованості процесу характеристики, які отримано за рекурентними діаграмами зменшуються, а у випадку з величинами ентропії збільшуються.

Експериментальні сигнали. На рис. 5.14 показані типові рекурентні діаграми реалізацій ЕЕГ у фазі повільного сну і бадьорості. ЕЕГ у фазі бадьорості відповідає більш структурована рекурентна діаграма.

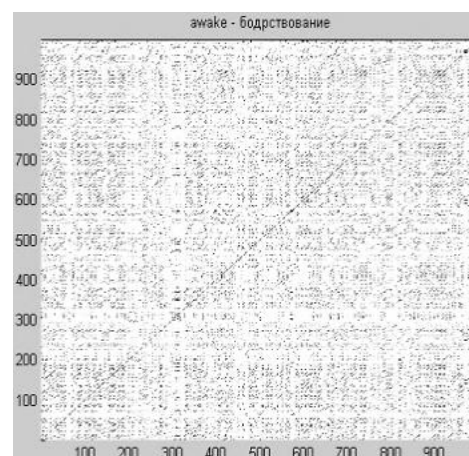
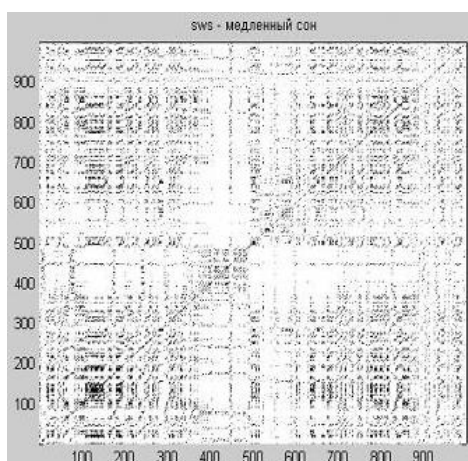


Рисунок 5.14 – Рекурентні діаграми для реалізацій ЕЕГ: sws (ліворуч) та awake (праворуч)

Спектри вейвлет-енергії ЕЕГ представлені на рис. 5.15. У разі бадьорості вейвлет-енергія реалізацій зосереджена на високих рівнях, а в разі фази сну більш рівномірно розподіляється за рівнями.

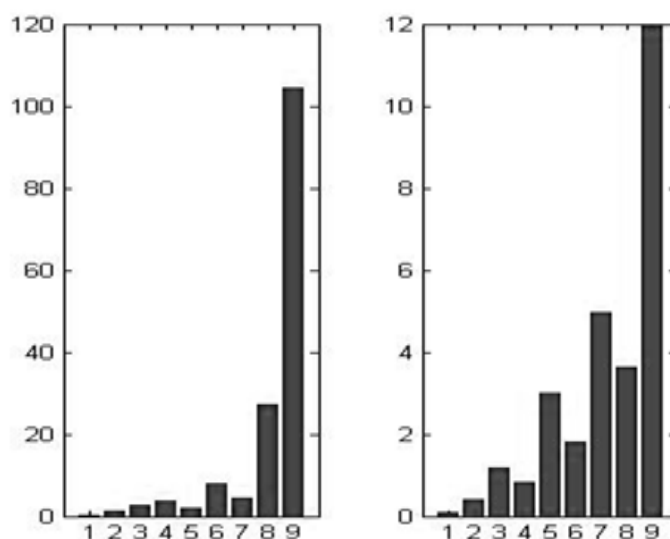


Рисунок 5.15 – Спектр вейвлет-енергії для реалізацій ЕЕГ: sws (ліворуч) та awake (праворуч)

У табл. 5.3 наведені середні значення характеристик складності для реалізацій ЕЕГ довжиною 1000 значень. Треба відзначити, що значення інформаційних характеристик для різних станів організму мають суттєві відмінності.

Таблиця 5.3 - Характеристики складності для реалізацій ЕЕГ

Стан	ApEn	Det	RR	W
Повільний сон	1.393	0,2644	0,0148	1,41
Бадьорість	1.502	0,3717	0,365	1,35

На рис. 5.16 показані рекурентні діаграми послідовностей RR-інтервалів до і після застосування препарату. Очевидно, що рекурентна структура сигналів має великі відмінності.

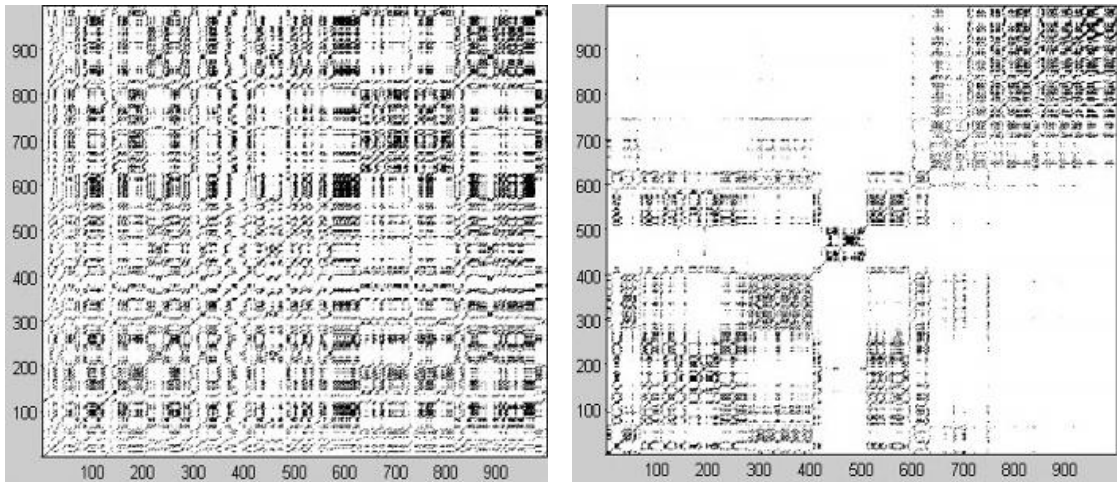


Рисунок 5.16 –Рекурентні діаграми для RR-інтервалів до (ліворуч) і після (праворуч) застосування препарату

Спектри вейвлет-енергії RR-інтервалів представлені на рис. 5.17. У разі послідовностей RR-інтервалів, отриманих після прийому медичного препарату, вейвлет-енергія реалізацій в основному міститься на високих рівнях, а в разі RR-інтервалів пацієнтів до лікування, вейвлет-енергія розподіляється нерівномірно по всіх рівнях.

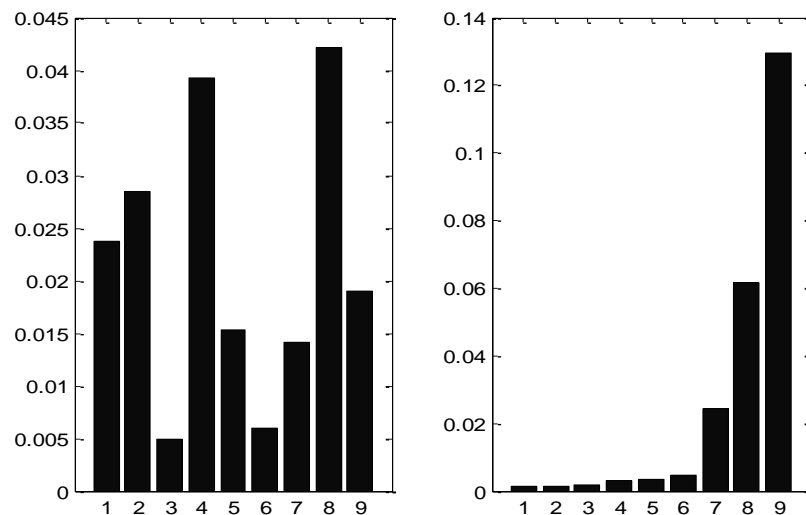


Рисунок 5.17 – Спектр вейвлет-енергії для RR-інтервалів до (ліворуч) і після (праворуч) застосування препарату

У табл. 5.4 наведені середні значення характеристик складності для реалізацій RR-інтервалів довжиною 1000 значень. Значення інформаційних характеристик для різних пацієнтів до і після застосування препарату значно відрізняються за значеннями.

Таблиця 5.4 - Характеристики складності для RR-інтервалів

Стан	ApEn	Det	RR	WE
до прийняття	0.242	0.0707	0.133	2.018
після прийняття	0.797	0.627	0.023	1.214

Важливим аспектом є те, що проведені дослідження виявили некоррелірованність один з одним величин вейвлет-ентропії W , ентропії подібності $ApEn$ та рекурентних характеристик. Це дозволяє використовувати їх як незалежні параметри при розпізнаванні часових рядів за допомогою СППР.

5.4.6 Дослідження зашумлених фрактальних часових рядів

Боротьба з шумами і перешкодами є актуальним завданням в багатьох прикладних областях науки і техніки. Оскільки зазвичай в корисному сигналі завжди присутня деяка кількість шумів, важливо знати, як змінюються фрактальні характеристики сумарних сигналів. В дослідженні було розглянуто різні модельні сигнали – часові реалізації, представлені сумою мультифрактального часового ряду і шумовими складовими з різними функціями розподілу і різними автокореляційними залежностями.

Сумарний сигнал можна представити як $X(t) = Y(t) + Z(t)$, де $Y(t)$ – мультифрактальний часовий ряд, $Z(t)$ - адитивний шум. Як величина, що

характеризує співвідношення мультифрактального сигналу і шуму, було використано коефіцієнт $SNR = \frac{D(Y(t))}{D(z(t))}$.

За модельні мультифрактальні часові ряди були вибрані реалізації стохастичного біноміального мультиплікативного каскаду. При побудові стохастичних каскадів ваговими коефіцієнтами є незалежні значення деякої заданої випадкової величини. У даній роботі в якості випадкової величини була використана величина, що має бета-розподіл. Це дозволяє отримувати реалізації з різним ступенем неоднорідності, тобто з великим діапазоном мультифрактальних властивостей. На реалізації мультифрактального каскаду адитивно накладалися такі види шуму: білий шум, рівномірний некорельований шум, корельований шум (авторегресія першого порядку), самоподібний шум (фрактальний гаусовський шум).

У роботі було виявлено, що при невеликому співвідношенні сигнал / шум ($SNR \leq 0.1$), узагальнений показник Херста $h(q)$ зашумленого ряду і показник $h(q)$ ряду без шуму практично збігаються при позитивних значеннях параметра q . Був чисельно досліджений узагальнений показник Херста $h(q)$ при різному відношенні сигнал / шум, і показано, що при зменшенні значення SNR від 1 до 0.1 показник $h(q)$ зашумленого ряду прагне до $h(q)$ вихідного ряду. Показано, що ця залежність має місце для будь-яких видів шумового сигналу. На рис. 5.18 наведені функції для початкового мультифрактального ряду і зашумленого ряду при різних значеннях коефіцієнта SNR .

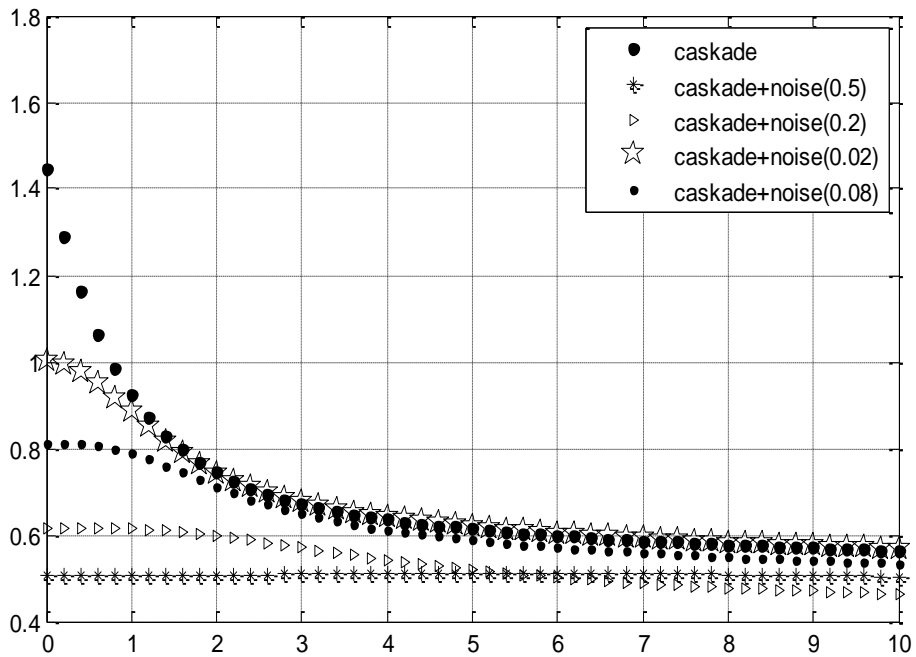


Рисунок 5.18– Узагальнений показник Херста початкового (.) та зашумлених рядів при різних SNR

Результати досліджень мають прикладне значення в таких областях як радіотехніка та цифрова сейсмологія, де однією з основних проблем є виділення корисного сигналу в зашумленому середовищі.

6 АДАПТИВНІ МЕТОДИ ТА МОДЕЛІ КЛАСИФІКАЦІЇ ДАНИХ І ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ ЗА УМОВ ЇХ ВИКРИВЛЕНOSTІ ВІДСУТНІМИ ТА АНОМАЛЬНИМИ СПОСТЕРЕЖЕННЯМИ НА ОСНОВІ ШТУЧНИХ ІМУННИХ СИСТЕМ

6.1. Імунний метод класифікації даних, що мають апріорі невідому кількість пропусків

Задача класифікації може бути вирішена різними методами, що функціонують як на основі чисто математичних моделей [216-218], так і на основі біологічних принципів організації обчислень, представлених апаратом теорії штучних нейронних мереж, генетичних алгоритмів, імунних систем та ін. [219-221]. Використання біологічних принципів організації обчислень при класифікації зазвичай застосовується для підвищення швидкості навчання, особливо при автоматичній класифікації даних.

Специфікою імунних методів класифікації є їх здатність відновлювати вихідні дані при навчанні, які представляються у вигляді популяції антитіл. Серед імунних моделей, які використовуються при класифікації об'єктів, можна виділити три основних [219-221]:

- 1) модель клонального відбору;
- 2) модель імунної мережі;
- 3) модель RLAIS (resources limited artificial immune system).

Основною відмінністю між моделлю клонального відбору і моделлю імунної мережі є мережева взаємодія між об'єктами. У методах, що функціонують на основі моделі імунної мережі, об'єкти, що використовуються при навчанні системи (антитіла) взаємодіють не лише з антигенами навчальної вибірки, але і між собою. Для методів, що функціонують на основі моделі клонального відбору, мережева взаємодія антитіл виключається. Модель RLAIS поєднує в собі деякі особливості

моделі клонального відбору і моделі імунної мережі, і відрізняється використанням областей розпізнавання ARB (artificial recognition balls), з яких в результаті навчання формуються кінцеві класи.

Класифікація ускладнюється за умови втрати цілісності опису об'єктів, що класифікуються, наприклад, появи шумів, або відсутності ознак, що характеризують груповані об'єкти. Крім того, більшості методів імунної класифікації характерний один загальний недолік – велика кількість надлишкових обчислень, що виникає в ході імунного навчання, внаслідок чого зростає час, що витрачається на навчання та класифікацію даних.

Метою проведеного дослідження є розробка імунного методу класифікації, здатного групувати дані, що мають апріорі невідому кількість пропусків, і вирішального проблему надлишкових обчислень, що виникає на етапі обчислень.

6.1.1 Постановка задачі класифікації даних

Дана множина об'єктів $OBJ = \{obj_1; \dots; obj_n\}$, кожен з яких характеризується масивом векторів метричних ознак змінної довжини $k, z = \overline{1, N}, D_{k \times z}$, де кожен характеризується вагою w_k . В разі класифікації без навчання або класифікації з контрольованим навчанням разом з вихідними об'єктами надається також навчальна вибірка з класифікованими об'єктами $SMP = \{s_1; \dots; s_m\}$ та набір вихідних класів $C = \{c_1; \dots; c_c\}$. При цьому ознаки об'єктів, що класифікуються, або об'єктів навчальної вибірки характеризуються відсутністю цілісності ознак, тобто ознаки у векторах об'єктів множини OBJ або множини SMP можуть бути частково відсутніми.

Відповідно до принципів функціонування імунних методів обробки даних, в якості міри близькості між ознаками імунних об'єктів використовується функція афінності [219-221]:

$$aff_{ij} = \left(+ d_{ij} \right)^{-1}, \quad (6.1)$$

де d_{ij} - евклідова або манхеттенська відстань між ознаками i -го та j -го імунних об'єктів.

Необхідно розробити адаптивний імунний метод класифікації початкової множини об'єктів OBJ , що характеризуються відсутністю цілісності даних при використанні навчальних вибірок SMP , або за відсутності інформації про класи.

6.1.2 Загальна схема класифікації даних на основі імунного підходу

Роботу імунного методу класифікації об'єктів, що характеризуються частковою відсутністю, ознак можна умовно розділити на три основних етапи: 1) нормалізація об'єктів, що класифікуються, з множини OBJ та навчальної вибірки SMP ; 2) перевірка необхідності навчання та проведення імунного навчання; 3) визначення критеріїв класифікації та приналежності початкових об'єктів класам великої кількості CL . Основні етапи класифікації даних на основі імунного підходу представлені на рис. 6.1 [223-226].

На першому етапі класифікації робиться нормалізація об'єктів, що класифікуються, з множини OBJ та об'єктів навчальної вибірки SMP . Відповідно до [222] існує декілька основних підходів, вживаних для обробки об'єктів, що характеризуються пропусками невідомої кількості ознак:

- виключення відсутніх ознак та класифікація об'єкту по наявних атрибутах;
- заповнення відсутніх ознак середніми для усієї множини об'єктів значеннями;
- заповнення відсутніх ознак середніми для найближчих об'єктів значеннями.

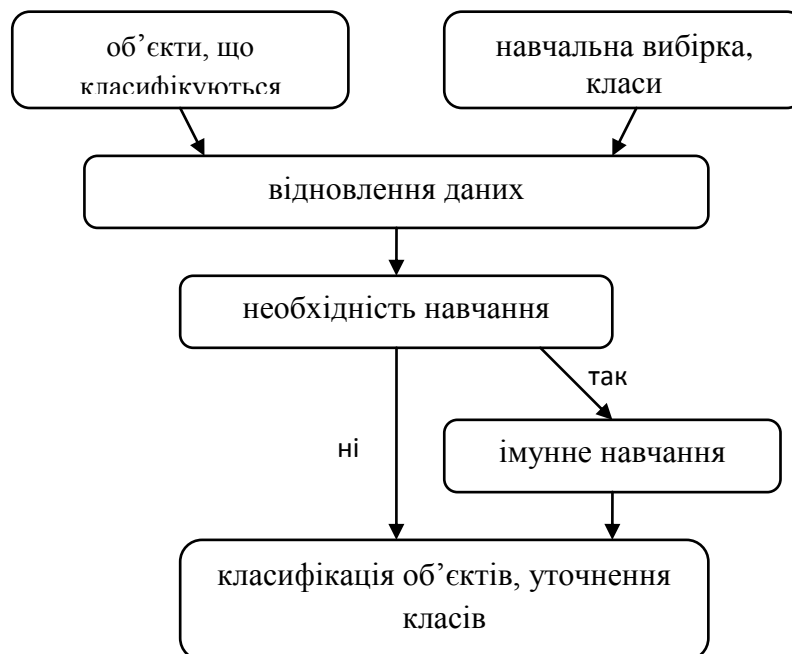


Рисунок 6.1 - Етапи класифікації даних на основі імунного підходу

При використанні першого підходу нормалізації об'єктів, що класифікуються, відсутні ознаки не враховуються в процесі імунного навчання та при визначенні приналежності об'єктів класам. Основною перевагою застосування першого підходу є підвищення швидкості класифікації даних. Головним недоліком є падіння точності класифікації.

При використанні другого підходу відсутні ознаки визначаються як середнє арифметичне відповідних ознак об'єктів, що класифікуються. Відповідно до цього, процес відновлення ознак відбувається таким чином: на першому етапі формується масив ознак, що підлягають відновленню. Ознака потрапляє в масив у тому випадку, якщо він відсутній хоч би у одного з об'єктів, що класифікуються. Потім для кожної ознаки із сформованого масиву пропущених значень визначається середнє арифметичне ознак неушкоджених об'єктів. Після цього значенням відновленої ознаки заповнюються відповідні пропуски в пошкоджених об'єктах, що класифікуються. Використання цього підходу призводить до збільшення часу роботи, в порівнянні з першим підходом, але підвищує точність класифікації.

При використанні третього підходу відсутні ознаки визначаються як середнє арифметичне відповідних ознак об'єктів, найближчих по афінності до пошкодженого об'єкту. Відповідно до цього, відновлення відсутніх ознак відбувається таким чином: для кожного об'єкту, що має пропуски, визначаються найближчі по афінності неушкоджені об'єкти. Після цього значення кожної пропущеної ознаки визначається як середнє арифметичне відповідних ознак найближчих неушкоджених об'єктів. Використання цього підходу призводить до збільшення часу класифікації та характеризується найбільшими тимчасовими витратами, в порівнянні з походами, розглянутими раніше, проте характеризується найбільшою точністю класифікації. Слід зазначити, що при використанні цього підходу кількість найближчих сусідів для пошкодженого об'єкту зазвичай є вхідним параметром, аналогічно тому, як це відбувається в методі kNN [216-218].

Після відновлення ознак для об'єктів, що класифікуються, і об'єктів навчальної вибірки, робиться перевірка необхідності імунного навчання. При цьому для об'єктів навчальної вибірки (антигенів) визначається афінність усередині популяції NAT , аналогічно [219-221]. Після цього визначається міжпопуляційна афінність $aff \langle AB, AG \rangle$ відповідно до виразу (6.2):

$$aff \langle AB, AG \rangle = \frac{1}{n} \sum_{i=1}^n \left(\frac{1}{m} \sum_{j=1}^m aff \langle b_i, ag_j \rangle \right), \quad (6.2)$$

де $aff \langle b_i, ag_j \rangle$ - афінність між i -им антитілом (об'єктом, що класифікується) та j -им антигеном (об'єктом навчальної вибірки) визначена відповідно до (6.1).

У випадку якщо між популяційна афінність більше або дорівнює афінності усередині навчальної вибірки NAT – імунне навчання не робиться і антитіла визначають приналежності до класів. У випадку якщо між популяційна афінність $aff \langle AB, AG \rangle$ менше афінності навчальної вибірки NAT – робиться контрольоване імунне навчання, об'єкти (антитіла), що при цьому

класифікуються, шляхом клонування, мутації і відбору клонів відтворюють ознаки об'єктів навчальної вибірки (антигенів). Подальша класифікація антитіл робиться на підставі результатів відновлення антигенів та їх приладдя початковим класам. У випадку якщо афінність навчальної вибірки *NAT* не може бути визначена, або навчальна вибірка відсутня – робиться неконтрольоване навчання [226] (автоматична класифікація, кластеризація), в результаті якого буде отриманий набір класів для об'єктів, що групуються.

В ході імунного навчання відбувається відновлення ознак об'єктів навчальної вибірки (антигенів) антитілами та їх клонами з популяції об'єктів, що класифікуються. Для організації імунного навчання використовується модифікований метод CLONALG, що функціонує на основі моделі клонального відбору. Модифікації в цьому методі піддався принцип обробки антитіл, оператор мутації клонів, оператор відбору клонів і оператор старіння. У початковому методі CLONALG використовується популяційна обробка антитіл, тобто уся популяція антитіл формує одну загальну множину клонів, які потім піддаються мутації та відбору, при цьому виникають складнощі при відборі клонів і старінні клонованих антитіл. Процес клонування, мутації та відбору клонів відбувається з меншими тимчасовими втратами у разі послідовної обробки популяції антитіл. При цьому для кожного клонованого антитіла послідовно формується окрема популяція клонів, яка потім піддається мутації та відбору, внаслідок чого відбувається скорочення операції сортування безлічі клонів. Оператор мутації клонів використовує афінність далекого предка [227] при визначенні коефіцієнта мутації. При відборі клонів використовується конкурентно-цільовий відбір [228], застосування якого підвищує вірогідність відновлення ознак антигенів.

Після імунного навчання робиться уточнення меж класів та визначення критеріїв класифікації, по яких визначатиметься приналежність об'єктів тому або іншому класу. Відповідно до цього, в якості критеріїв класифікації для кожного класу визначається гранична афінність входження об'єктів в клас, а також внутрішня середня афінність між усіма об'єктами, що входять в цей

клас. Після визначення критеріїв класифікації всі некласифіковані об'єкти визначають приналежності до класів шляхом зіставлення афінностей та критеріїв класифікації. Класифікація об'єктів полягає у визначенні приналежності одному з існуючих класів для об'єктів, що не відновили ознаки одного з антигенів навчальної вибірки. Для тих об'єктів, що мають низькі афінності до існуючих класів, тобто афінність яких менше критеріїв класифікації, визначених раніше, формуються додаткові класи.

6.1.3 Відновлення пропущених ознак вихідних даних

При відновленні пропущених ознак об'єктів, що класифікуються, в цілях підвищення швидкості класифікації без втрати точності використовується не один, а декілька підходів, використовуваних для відновлення ознак, наведених в [222-223]. Слід зазначити, що при ухваленні рішення про необхідність відновлення відсутніх ознак доцільно враховувати значення ваги групи з відсутніми ознаками. Таким чином, якщо пошкоджена група ознак характеризується мінімальною вагою – необхідність відновлення пропущених ознак в групі відсутня. При цьому необхідність відновлення пропущених ознак в групі зростає при збільшенні ваги групи. Відповідно до цього відсутні ознаки об'єкту не відновлюються при виконанні наступної умови:

$$\sum_{i=1}^k w_i \ll \sum_{j=1}^k w_j \ll x \ll \sum_{f=1}^k w_f \quad (6.3)$$

де $w_i \ll$ – вага i -ої групи неушкоджених ознак, $w_j \ll x \ll$ – вага j -ої групи частково пошкодженої групи ознак.

Якщо умова (6.3) не виконується, то необхідне відновлення ознак пошкоджених груп імунних об'єктів, яке виробляється в разі виконання умови (6.4):

$$\sum_{i=1}^k w_i \llbracket \varphi \rrbracket \not\geq \sum_{j=1}^k w_j \llbracket \varphi x \rrbracket \not\geq \sum_{f=1}^k w_f \llbracket \varphi \rrbracket \quad (6.4)$$

У випадку якщо умови (4.3) та (4.4) не виконуються, робиться перевірка об'єкта на повну відсутність у нього ознак. При цьому ознаки об'єкта не відновлюються та робиться висновок про неможливість його класифікації, перевірка чого здійснюється при виконанні виразу (6.5):

$$\sum_{i=1}^k w_i \llbracket \varphi \rrbracket \not\geq \sum_{j=1}^k w_j \llbracket \varphi x \rrbracket \not\geq \sum_{f=1}^k w_f \llbracket \varphi \rrbracket \quad (6.5)$$

Таким чином об'єкти, в яких повністю відсутні ознаки в групах, що характеризуються максимально скупюю вагою, не беруть участь в класифікації і не можуть бути класифіковані.

Відновлення пропущених ознак відбувається завдяки імунному механізму мережевої взаємодії. Відповідно до цього антитіла взаємодіють між собою, що виражається у визначенні афінностей по наявних ознаках. Після визначення афінностей для кожного пошкодженого антитіла визначається множина найближчих по афінності неушкоджених антитіл, в яких присутні необхідні для відновлення ознаки. При формуванні даної множини неушкоджених антитіл використовується критерій *NAT* (6.2), що характеризує середню афінність між всіма антитілами. Відповідно до цього, антитіло, що характеризується присутністю ознак, включається в множину найближчих антитіл при виконанні наступної умови:

$$ab_j \in AB'_i, \text{aff} \llbracket ab_i, ab_j \rrbracket \geq NAT, \quad (6.6)$$

де ab_i - антитіло, що характеризується пошкодженими ознаками, для якого формується множина найближчих неушкоджених антитіл AB'_i ; ab_j - антитіло, що характеризується цілісністю ознак.

Після формування множини найближчих неушкоджених антитіл, для кожного пошкодженого антитіла виробляється відновлення відсутніх ознак.

Відповідно до цього пошкоджені ознаки антитіл відновлюються наступним чином:

$$ab_i.attr_j = \frac{1}{n} \sum_{k=1}^n ab'_k.attr_j, \quad ab'_k \in AB'_i, \quad (6.7)$$

де $ab_i.attr_j$ – пошкоджена j -а ознака i -го антитіла, що характеризується відсутніми ознаками; $ab'_k.attr_j$ – відповідна неушкоджена j -а ознака k -го антитіла з множини найближчих неушкоджених антитіл AB'_i ; n – кількість найближчих неушкоджених антитіл, що містяться в множині AB'_i .

При використанні даного підходу відновлюються не лише пропущені ознаки антитіл в частково пошкоджених групах, але і повністю відсутні групи ознак. Слід зазначити, що відновлення ознак може вироблятися не лише для об'єктів (антитіл), що класифікуються, але і для об'єктів навчальної вибірки (антигенів), які також можуть мати пропущені ознаки. Процес відновлення ознак пошкоджених об'єктів навчальної вибірки відбувається аналогічно тому, як це відбувається для популяції антитіл, що класифікуються. На початку процесу відновлення для всіх антигенів визначається значення NAT , що є середньою афінністю у середині об'єктів навчальної вибірки. Після цього для кожного пошкодженого антигену на підставі афінностей до інших неушкоджених об'єктів навчальної вибірки і критерію NAT формується множина найближчих антигенів, після чого відбувається процес відновлення пропущених ознак середніми значеннями відповідних ознак, найближчих по афінності антигенів.

6.1.4 Імунне навчання класифікатора

Імунне навчання – це процес досягнення специфічності вихідних антитіл класифікованим антигенам шляхом їх клонування, мутації і відбору

клонів, що характеризуються високою афінністю до антигенів навчальної вибірки. Після того, як антитіло або його клон, що мутував, досяг стану специфічності одному з антигенів, його приналежність класу визначається по приналежності специфічного антигена. Таким чином, висновок про приналежність класам вихідної популяції об'єктів, що класифікуються, робиться за результатами обробки останньої популяції антитіл.

Слід зазначити, що класифікація вихідних об'єктів може вироблятися і без етапу імунного навчання. Висновок про необхідність імунного навчання робиться на підставі афінностей популяції антитіл, що класифікуються, до популяції антигенів навчальної вибірки. При цьому якщо афінність популяції антитіл до популяції антигенів не менше значення NAT , визначеного для об'єктів навчальної вибірки, то імунне навчання не виробляється, і антитіла, що класифікуються, визначають приналежність класам по максимальним афінностям до їх центрів. Імунне навчання виробляється при виконанні однієї з наступних умов: 1) за відсутності навчальної вибірки і вихідних класів; 2) у разі, коли середня афінність антитіла, що класифікуються, менше значення NAT для антигенів навчальної вибірки. В першому випадку відбувається контрольоване навчання (навчання з вчителем), а в другому – навчання без вчителя і подальша кластеризація об'єктів.

При контрольованому імунному навчанні антигенами є об'єкти навчальної вибірки, а антитілами – об'єкти, що класифікуються. На рівні імунних операторів етап контрольованого імунного навчання представляється в такий спосіб [223, 230]:

$$AISsl \langle AB, AG \rangle \Rightarrow \left[\begin{array}{l} NatCalculation \langle AG \rangle \Rightarrow \\ AgPresentation \langle AB, AG \rangle \Rightarrow \\ ClassSelection \langle AB, AG \rangle \end{array} \right] \Rightarrow \left\langle \begin{array}{l} Cloning \langle AB, CL \rangle \Rightarrow \\ Mutation \langle CL \rangle \Rightarrow \\ ClSelection \langle CL, AG \rangle \Rightarrow \\ Apoptosis \langle AB, CL \rangle \end{array} \right\rangle^T \quad (6.8)$$

де $NatCalculation \langle AG \rangle$ – оператор визначення значення NAT для

популяції антигенів навчальної вибірки; $AgPresentation \langle AB, AG \rangle$ – оператор представлення антигенів антитілам, визначення між ними афінностей і виділення цільових антигенів для кожного антитіла; $ClassSelection \langle AB, AG \rangle$ – оператор визначення класу для антитіла по приналежностях цільових агентів; $Cloning \langle AB, CL \rangle$ – оператор клонування некласифікованих антитіл; $Mutation \langle CL \rangle$ – оператор мутації клонів; $ClSelection \langle CL, AG \rangle$ – оператор відбору клонів; $Apoptosis \langle AB, CL \rangle$ – оператор старіння клонованих антитіл; T – критерій зупинки навчання.

При представленні навчальних антигенів антитілами в ході роботи оператора $AgPresentation \langle AB, AG \rangle$, кожне антитіло визначає афінність до всієї множини антигенів. Після цього для кожного антитіла визначається множина цільових антигенів, що характеризуються приналежністю до одного з вихідних класів. Антитіло стає цільовим, якщо його афінність з даним антитілом перевищує значення NAT . При цьому кількість цільових антигенів визначається відповідно до виразу (6.9):

$$AG_i^t = ClassNumber \cdot k, \quad (6.9)$$

де AG_i^t – множина цільових антигенів i -го антитіла, $ClassNumber$ – кількість класів, k – коефіцієнт збільшення зони пошуку $k \in N$.

Після визначення цільових антигенів кожне антитіло, що класифікується, визначає приналежність до класу в ході роботи оператора $ClassSelection \langle AB, AG \rangle$. При цьому антитіло визначає приналежність до класу в тому випадку, якщо всі його цільові антигени належать даному класу. Якщо дана умова виконується – антитіло стає кліткою імунної пам'яті і не піддається клонуванню і мутації. Антитіла, що не є клітками імунної пам'яті, піддаються дії оператора пропорційного клонування $Cloning \langle AB, CL \rangle$. Кількість створюваних клонів визначається відповідно до виразу (6.10):

$$Ncl_i = n \cdot \max aff \left(\left[b_i, ag_j^t \right] \right), \quad (6.10)$$

де Ncl_i – кількість клонів, що створюються i -им антитілом; n – загальна кількість антитіл, що класифікуються, $\max aff \left(\left[b_i, ag_j^t \right] \right)$ – максимальна афінність i -го клонованого антитіла до j -го цільового антигена.

Популяція клонів, яка сформована в результаті роботи оператора клонування, піддається дії оператора мутації $Mutation \left(\left[CL \right] \right)$, який використовується для зміни ознак клонів з метою досягнення стану специфічності цільовим антигенам. У даному методі класифікації використовується обернено пропорційна мутація на основі афінності далекого предка, запропонованого в [229, 230]. Основна відмінність даного оператора мутації полягає в засобі визначення коефіцієнта мутації, що впливає на характер змін ознак клонів. При визначенні коефіцієнта мутації як нижній поріг використовується афінність антитіла – батька по відношенню до антитіла, клони якого піддаються мутації. Дане антитіло – батько є далеким предком по відношенню до клонів, що мутують, для яких обчислюється коефіцієнт мутації. Визначення коефіцієнта мутації на основі афінності далекого предка відбувається в спосіб:

$$\mu = rand \left[-aff \left(\left[b_i^*, AG_i^t \right] \right); 1 - aff \left(\left[b_i, AG_i^t \right] \right) \right] \quad (6.11)$$

де $aff \left(\left[b_i^*, AG_i^t \right] \right)$ – афінність далекого предка клонованого i -го антитіла до цільових антигенів, $aff \left(\left[b_i, AG_i^t \right] \right)$ – афінність клонованого антитіла до цільових антигенів. Після визначення коефіцієнта мутації зміна ознаки i -го клону $cl_i.attr$ виробляється відповідно до виразу (6.12):

$$cl_i.attr = cl_i.attr \pm \mu \cdot \left(\frac{1}{aff \left(\left[b_i, AG_i^t \right] \right)} - 1 \right), \quad (6.12)$$

де μ – коефіцієнт мутації, визначуваний відповідно до (6.11), при цьому, знак \pm у виразі (6.12) визначається випадковим чином.

Після проведення мутації виконується стиснення популяції клонів при використанні оператора клонального відбору $ClSelectio n \langle CL, AG \rangle$. У даному методі класифікації використовується конкурентно-цільовий відбір [223, 228, 231], який виконується при використанні множини цільових антигенів, виділеної на етапі представлення антигенів популяції антитіл. В результаті конкурентно-цільового відбору для кожного клонованого антитіла зі всієї множини клонів виживає один клон, що характеризується максимальною афінністю до одного з цільових антигенів. Слід зазначити, що використання даного підходу при організації клонального відбору підвищує швидкість імунного навчання. Це зумовлюється скороченням кількості обчислювальних операцій при визначенні афінностей до антигенів. При цьому скорочення обчислювальних операцій досягається за рахунок скорочення кількості антигенів, які представляються клонам, оскільки кількість цільових антигенів завжди менше загальної кількості навчальних антигенів.

Оператор старіння антитіл $Apoptosys \langle AB, CL \rangle$ використовується для заміни клонованих антитіл клонами, що характеризуються високими афінностями до цільових антигенів. При цьому клоноване антитіло замінюється клоном, відібраним в результаті роботи оператора клонального відбору, у випадку, якщо максимальна афінність клона до одного з цільових антигенів перевищує максимальну афінність клонованого антитіла.

В якості критерія зупинки імунного навчання використовується максимальна кількість популяцій антитіл, використовуваних при імунному навчанні. Окрім цього, зупинка процесу навчання відбувається у випадку, якщо всі антитіла визначали свої приналежності до кластерів по приналежності цільових антигенів, або у разі, коли всі антитіла стали специфічними популяції початкових антигенів.

Алгоритм контрольованого імунного навчання представляється у

вигляді послідовності наступних основних дій [223]:

1. Підготовчий етап.
 - 1.1. Визначення порогового значення *NAT* для популяції навчальних антигенів.
 - 1.2. Визначення афінностей між антигенами та антитілами та виділення цільових антигенів.
 - 1.3. Аналіз цільових антигенів і визначення класів для антитіл
2. Для кожного некласифікованого антитіла виконати:
 - 2.1. Пропорційне клонування.
 - 2.2. Обернено- пропорційна мутація.
 - 2.3. Конкурентно-цільовий відбір клонів.
 - 2.4. Старіння антитіл.
 - 2.5. Перевірка критерію зупинки.

Таким чином, класифікація антитіл відбувається на різних етапах імунного навчання. На підготовчому етапі імунного навчання антитіла можуть бути класифіковані у випадку, якщо всі їх цільові антигени належать одному класу. Такі антитіла стають клітками імунної пам'яті та не беруть участь в процесі імунного навчання. Окрім цього, антитіла визначають приналежність до класу досягнувши специфічності одному з цільових антигенів. При цьому таке антитіло належить тому ж класу, що і специфічний антиген, стає кліткою імунної пам'яті та не бере участь в процесі імунного навчання. Класифікація антитіл, що не досягли стану специфічності антигенам, відбувається після визначення критеріїв класифікації.

При навчанні без вчителя навчальна вибірка та вихідна інформація про класи відсутня. Відповідно до цього процес класифікації полягає не лише у визначенні приналежності об'єктів тому або іншому класу, але і в безпосередньому виділенні класів в ході імунного навчання. Імунне навчання робиться послідовно для кожного антитіла, що класифікується [224-226]. Для кожного антитіла антигенами є антитіла, що залишилися в популяції. В процесі навчання кожне антитіло визначає множину цільових антигенів,

найближчих до нього по афінності, формуючих з ними новий центр класу. Після завершення процесу імунного навчання робиться укрупнення класів за рахунок злиття отриманих раніше груп антитіл, що характеризуються між собою максимальними афінностями. На рівні імунних операторів процес неконтрольованого імунного навчання представляється таким чином:

$$\begin{aligned}
 AISnsl \langle AB \rangle \approx NatCalculation \langle AG \rangle \Rightarrow & \left\{ \begin{array}{l} AgPresentation \langle AB, AG \rangle \xrightarrow{T} \\ Cloning \langle AB, CL \rangle \xrightarrow{} \\ Mutation \langle CL \rangle \xrightarrow{} \\ AG^t Selection \langle AB, AG \rangle \xrightarrow{} \\ ClSelection \langle CL, AG \rangle \xrightarrow{} \\ Apoptosis \langle AB \rangle \xrightarrow{} \end{array} \right\} \Rightarrow \\
 \Rightarrow & \left[\begin{array}{l} ClassCenterSelection \langle AB, C \rangle \xrightarrow{} \\ ClassNatCalculation \langle C \rangle \xrightarrow{} \\ ClassMerging \langle C, NAT \rangle \xrightarrow{} \end{array} \right]
 \end{aligned} \tag{6.13}$$

де $AG^t Selection \langle AB, AG \rangle$ – оператор відбору цільових антигенів; $ClassCenterSelection \langle AB, C \rangle$ – оператор визначення центрів класів; $ClassNatCalculation \langle C \rangle$ – оператор визначення значення NAT для кожного класу; $ClassMerging \langle C, NAT \rangle$ – оператор злиття класів.

Робота оператора відбору цільових антигенів $AG^t Selection \langle AB, AG \rangle$ полягає у скороченні кількості цільових антигенів для прискорення процесу формування кластерів. На етапі представлення антигенів антитілам після формування множини цільових антигенів між ними і антитілом, що класифікується, визначається середня афінність NAT^t аналогічно тому, як це відбувається при визначенні значення NAT . Скорочення множини цільових антигенів відбувається шляхом зіставлення афінностей між антитілом і цільовими антигенами зі значенням середньої афінності NAT^t . Антиген покидає множини цільових антигенів у випадку, якщо його афінність до антитіла менше значення NAT^t . Слід зазначити, що на етапі конкурентно-цільового відбору клонів при виконанні оператора $ClSelection \langle CL, AG \rangle$,

клонам, що мутовали, представляється відредагована множина цільових антигенів. Обробка антитіла при навчанні припиняється в разі досягнення стану специфічності, або скорочення множини цільових антигенів до двох.

Робота оператора виділення центрів $ClassCenterSelection(A, B, C)$ у сформованих класах полягає в зіставленні афінностей антитіл, що належать одному класу, та вибору центрального антитіла, що характеризується максимальною афінністю до всіх об'єктів, що належать даному класу. Після цього відбувається визначення критерію NAT^c для кожного сформованого класу при виконанні оператора $ClassNatCalculation(C)$. Дане значення визначається аналогічно значенню NAT для популяції антигенів. Скорочення кількості класів, що мають мінімальну кількість антитіл, відбувається при виконанні оператора $ClassMerging(C, NAT)$. При об'єднанні класів відбувається визначення афінності між їх центральними об'єктами. Відповідно до цього класи об'єднуються у випадку, якщо афінність між їх центрами перевищує значення NAT^c . Клас, сформований в результаті такого об'єднання, отримує всі антитіла, що належать цим класам.

Алгоритм неконтрольованого імунного навчання представляється у вигляді наступної послідовності дій [225, 226, 232, 233]:

1. Підготовчий етап.
 - 1.1. Визначення порогового значення NAT для популяції антигенів.
 - 1.2. Визначення афінностей між антигенами і антитілами та виділення цільових антигенів.
2. Для кожного антитіла виконати:
 - 2.1. Пропорційне клонування.
 - 2.2. Обернено-пропорційна мутація.
 - 2.3. Скорочення кількості цільових антигенів.
 - 2.4. Конкурентно-цільовий відбір клонів.
 - 2.5. Старіння антитіл.
 - 2.6. Перевірка критерію зупинки.

3. Об'єднання класів.
- 3.1. Визначення антитіл, класів, що є центрами.
- 3.2. Визначення *NAT* між антитілами усередині кожного класу.
- 3.3. Об'єднання класів.

Таким чином, на першому етапі формування кластерів відбувається об'єднання антитіл у велику кількість дрібних груп, кожна з яких є окремим класом. Після цього відбувається об'єднання найближчих по афінності дрібних класів в кінцевий набір кластерів.

6.1.5 Критерії класифікації та угруповання даних

Після проведення імунного навчання відбувається класифікація антитіл, які в процесі клонування, мутації та відбору клонів не досягли стану специфічності навчальним антигенам, або не увійшли ні до одного з сформованих кластерів. Для цього визначаються критерії класифікації, за допомогою яких виробляється угруповання некласифікованих антитіл, а також формування нових класів. Під критеріями класифікації розуміються значення граничних афінностей, що дозволяють віднести антитіло до того або іншого класу, або сформувати новий клас з некласифікованих об'єктів. При цьому використовуються два види критеріїв, що використані для класифікації даних [223, 226]:

- 1) критерії, які використані для визначення приналежності до існуючих класів некласифікованих антитіл;
- 2) критерії, які використані для виділення нових класів з множини некласифікованих антитіл.

При визначенні критеріїв першої групи усередині кожного існуючого класу визначається афінність між центром класу та найбільш віддаленим від нього антитілом, що належить даному класу. Після цього для кожного некласифікованого антитіла виробляється визначення афінностей до центрів існуючих класів. Визначення класів для некласифікованих антитіл

відбувається при виконанні наступних умов:

$$ab_i \in CL_j = \begin{cases} aff \langle b_i, ab_j^c \rangle \geq \min aff \langle b_j^c, ab_j \rangle; ab_j^c, ab_j \in CL_j \\ aff \langle b_i, ab_j^c \rangle \geq \max aff \langle b_i, ab^{CL} \rangle; CL_j \notin CL \end{cases}, \quad (6.14)$$

де ab_i - некласифіковане i -е антитіло; CL_j – клас, до якого визначається приналежність; $aff \langle b_i, ab_j^c \rangle$ – афінність між некласифікованим i -м антитілом та антитілом ab_j^c , що є центром класу CL_j ; $\min aff \langle b_j^c, ab_j \rangle$ – афінність між центром класу ab_j^c та найбільш віддаленим від нього антитілом ab_j , що належить даному класу; $\max aff \langle b_i, ab^{CL} \rangle$ – максимальна афінність до центру одного з останніх існуючих класів.

При визначенні критеріїв для виділення нових класів з множини некласифікованих антитіл, відбувається визначення афінностей між антитілами, центрами даних класів, що є. Критерієм, при використанні якого робиться виділення центру нового класу, є мінімальна афінність між центрами існуючих класів. Формування нових класів відбувається в разі існування множини некласифікованих антитіл, після проведення класифікації при використанні критеріїв першої групи. При цьому в результаті мережевої взаємодії некласифікованих антитіл, між ними виробляється визначення афінностей, після чого серед них виділяються антитіла, що характеризуються максимальними афінностями до останньої множини некласифікованих об'єктів. Дані антитіла є центрами класів, що формуються. При цьому для формування нового класу афінність антитіла, що є центром формованого класу, повинна задовольняти критерію формування класів:

$$\exists ab_i^c \in CL_i \Leftrightarrow \max aff \langle b_i^c, ab_j^c \rangle \geq \min aff \langle CL \rangle, \quad (6.15)$$

де ab_i^c - некласифіковане антитіло, що є центром формованого класу; CL_i – клас, що формується для некваліфікованих антитіл, $aff \langle b_i^c, ab_j^c \rangle$ –

максимальна афінність між центром сформованого класу та центром існуючого класу, $min\,aff(\mathcal{C}L)$ – мінімальна афінність між центрами існуючих класів антитіл.

Після виділення центрів сформованих класів для всіх неклассифікованих антитіл визначаються афінності до центрів сформованих класів аналогічно тому, як це відбувається при визначенні критеріїв класифікації антитіл після етапу імунного навчання.

6.1.6 Експериментальні дослідження

Тестування імунних методів класифікації даних, що характеризуються апріорі невідомою кількістю пропусків, виконувалося на декількох наборах вихідних даних, при використанні різних імунних методів класифікації та різних підходів відновлення даних. Як еталонні імунні методи класифікації використовувалися методи CLOANLG, BCA, aiNet та метод RLAIIS. Для порівняння результатів класифікації та виявленні погрішності визначення класів, вихідні набори даних піддавалися пошкодженням в різній мірі від 10% до 80%, при цьому під пошкодженнями розуміються внесення пропусків до вихідних даних. Крім того, як контрольна група використовувалися дані з неушкодженими ознаками. Вихідні набори даних представлені в табл. 6.1.

Таблиця 6.1 – Характеристики набору даних

Характеристики даних	Набір 1	Набір 2	Набір 3	Набір 4	Набір 5	Набір 6
Кількість об'єктів	100	500	800	1000	1200	1500
Матриця ознак	3×3	6×6	8×8	10×10	10×10	12×12
Міра пошкодження	0 %	10 %	30 %	60 %	70 %	80 %

Класифікація наборів даних проводилася при використанні навчальних

вибірок, які використовувалися для наборів 1, 2, 3 та 5. При обробці наборів 4 і 6 навчальні вибірки не використовувалися, тобто відбувалася автоматична класифікація (кластеризація) даних з неконтрольованим імунним навчанням. Опис навчальних вибірок, використаних для перерахованих наборів даних, представлений в табл. 6.2.

Таблиця 6.2 – Характеристики навчальних вибірок

Характеристики навчальних вибірок	Набір 1	Набір 2	Набір 3	Набір 5
Кількість об'єктів у вибірці	40	400	1000	1000
Кількість класів	3	7	5	10
Міра пошкодження об'єктів	0 %	5 %	10 %	15 %

Перед проведенням класифікації об'єктів відповідно до схеми, представленої на рис. 6.1, в разі виявлення пропусків у групованих об'єктах, проводиться відновлення відсутніх ознак. Вирішення задачі відновлення ознак відбувається на основі найбільш поширених методів, приведених в [222], та при використанні комбінованого методу, що функціонує на основі імунного підходу. Результати класифікації наводяться в табл. 6.3.

У табл. 6.3 використовуються такі скорочень: параметр В характеризує швидкодію методу, тобто загальний час, витрачений на класифікацію вихідних об'єктів, параметр К характеризує кількість класів, виділених в результаті класифікації, параметр Т характеризує точність класифікації, тобто кількість виділених класів, в разі відсутності навчальних вибірок, та розподіл об'єктів між класами. За результатами класифікації наборів даних (табл. 4.1) метод ВСА характеризується найгіршою швидкодією та найкращою точністю формування класів. З табл. 4.3 витікає, що при класифікації без відновлення пропущених ознак точність класифікації знижується в середньому на 20-35%, а швидкодія підвищується в середньому

на 5-10%. При цьому підвищення розмірності об'єктів, що класифікуються, призводить до зниження точності в середньому на 3-5% при підвищенні швидкості в середньому на 3-4%.

Таблиця 6.3 – Результати класифікації наборів даних

Иммунні методи класифікації			Набір 1	Набір 2	Набір 3	Набір 4	Набір 5	Набір 6
BCA	Без відновлення ознак	<i>B</i>	100 %	88.5 %	82.8 %	77.4 %	74.9 %	75.2 %
		<i>K</i>	3	7	5	6	10	9
		<i>T</i>	100 %	80.4 %	72.9 %	65.3 %	60.3 %	60.7 %
	Відновлення середніми по множині	<i>B</i>	–	94.7 %	89.3 %	84.6 %	71.3 %	72.8 %
		<i>K</i>	–	7	5	6	10	9
		<i>T</i>	–	94.3 %	86.7 %	79.8 %	73.5 %	74.1 %
	Відновлення середніми по афінності	<i>B</i>	–	100 %	100 %	100 %	100 %	100 %
		<i>K</i>	–	7	5	6	10	9
		<i>T</i>	–	100 %	100 %	100 %	100 %	100 %
CLONALG	Без відновлення ознак	<i>B</i>	79.8 %	50.6 %	44.2 %	39.5 %	37.4 %	37.2 %
		<i>K</i>	3	7	5	8	10	6
		<i>T</i>	82.3 %	58.3 %	51.8 %	39.2 %	34.9 %	34.5 %
	Відновлення середніми по множині	<i>B</i>	–	56.8 %	50.3 %	45.5 %	36.7 %	35.6 %
		<i>K</i>	–	7	5	5	10	8
		<i>T</i>	–	72.1 %	71.4 %	58.8 %	52.6 %	51.8 %
	Відновлення середніми по афінності	<i>B</i>	–	62.1 %	55.6 %	50.9 %	40.8 %	39.6 %
		<i>K</i>	–	7	5	6	10	8
		<i>T</i>	–	77.9 %	76.8 %	64.5 %	58.4 %	57.8 %
aiNET	Без відновлення ознак	<i>B</i>	75.1 %	47.9 %	40.7 %	31.3 %	29.2 %	28.8 %
		<i>K</i>	3	7	5	12	10	5
		<i>T</i>	78.7 %	51.5 %	45.3 %	33.5 %	29.8 %	29.1 %
	Відновлення середніми по множині	<i>B</i>	–	54.1 %	47.2 %	41.2 %	30.9 %	29.8 %
		<i>K</i>	–	7	5	9	10	7
		<i>T</i>	–	65.4 %	64.2 %	52.8 %	46.3 %	45.1 %
	Відновлення середніми по	<i>B</i>	–	59.4 %	52.9 %	42.6 %	33.8 %	30.1 %
		<i>K</i>	–	7	5	7	10	8

	афінності	<i>T</i>	–	71.1 %	69.6 %	58.1 %	51.8 %	51.4 %
RLAIS	Без відновлення ознак	<i>B</i>	93.2 %	60.2 %	57.8 %	53.2 %	49.9 %	47.8 %
		<i>K</i>	3	7	5	6	10	9
		<i>T</i>	95.2 %	73.2 %	71.8 %	64.7 %	59.4 %	58.6 %
	Відновлення середніми по множині	<i>B</i>	–	65.4 %	59.4 %	54.4 %	50.2 %	49.7 %
		<i>K</i>	–	7	5	6	10	9
		<i>T</i>	–	87.1 %	84.8 %	78.2 %	77.1 %	75.6 %
	Відновлення середніми по афінності	<i>B</i>	–	70.7 %	64.9 %	61.8 %	57.1 %	55.8 %
		<i>K</i>	–	7	5	6	10	9
		<i>T</i>	–	92.8	91.3 %	91 %	90.2 %	89.8 %
Пропонований метод	Без відновлення ознак	<i>B</i>	92.8 %	53.9 %	50.6 %	48.4 %	46.2 %	45.9 %
		<i>K</i>	3	7	5	6	10	9
		<i>T</i>	97.1 %	75.5 %	74.9 %	73.8 %	71.1 %	70.7 %
	Відновлення середніми по множині	<i>B</i>	–	57.1 %	55.8 %	51.9 %	49.8 %	48.3 %
		<i>K</i>	–	7	5	6	10	9
		<i>T</i>	–	89.9 %	88.2 %	86.9 %	83.5 %	79.1 %
	Відновлення середніми по афінності	<i>B</i>	–	60.4 %	57.6 %	53.9 %	50.5 %	49.4 %
		<i>K</i>	–	7	5	6	10	9
		<i>T</i>	–	95.1 %	93.8 %	93.3 %	92.6 %	92.3 %

Слід зазначити, що метод aiNet характеризується найкращою швидкістю та перевершує по даній характеристиці метод ВСА на 35-40%, проте при цьому поступається йому в точності на 27-30%. Метод CLONALG поступається методу aiNet по швидкості на 3-5%, але перевершує його по точності на 5-8%. Таким чином, методи aiNet і CLONALG співставні за своїми основними характеристиками, не дивлячись на різні моделі, на основі яких реалізуються дані методи. Метод RLAIS поступається методам aiNet і CLONALG по швидкодії на 10-15%, проте перевищує їх в точності, класифікації та по даному параметру співставний з методом ВСА, та поступається йому на 7-10%. Слід зазначити, що пропонуванний метод класифікації перевершує метод RLAIS по точності на 1-3%, і по швидкодії на

5-10%. Таким чином, запропонований метод співставний з методом ВСА, поступається йому по точності на 5-8% та перевершує його по швидкодії в середньому на 30-37%, що практично порівняно з методами aiNet і CLONALG.

Підсумовуючи вищенаведені результати класифікації даних, що мають апріорі невідому кількість пропусків, можна зробити наступні висновки.

1. Розглянуто вирішення задачі класифікації даних, що мають апріорі невідому кількість пропусків, на основі моделі клонального відбору.

2. У запропонованому методі класифікація виконується після автоматичного відновлення пропущених даних на основі імунного підходу.

3. Використання цільових антигенів на етапі імунного навчання приводить до підвищення швидкодії при класифікації. Це відбувається завдяки тому, що кількість антигенів, що представляються антитілами на різних етапах імунного навчання, знижується та обмежується невеликою кількістю цільових антигенів. Надалі цільові антигени використовуються на етапі відбору клонів після мутації, що також приводить до підвищення швидкості імунного навчання.

4. Вживання мутації на основі афінності далекого предка підвищує швидкість імунного навчання та точність класифікації завдяки зміні принципу визначення коефіцієнта мутації, використаного для зміни ознак клонів.

3. З результатів експериментів, проведених для різних імунних методів, витікає, що використання імунного підходу для відновлення відсутніх ознак підвищує точність класифікації в середньому на 5-8%, але при цьому призводить до зниження швидкодії на 2-3% в порівнянні з не імунними методами відновлення ознак.

Таким чином, завдяки використанню цільових антигенів і мутації на основі афінності далекого предка при імунному навчанні, а також імунного методу відновлення пропущених ознак, запропонований метод класифікації співставний з методами RLAIIS і ВСА по точності формування класів, а з

методами aiNet і CLONALG – за швидкістю імунного навчання і класифікації вихідних об'єктів.

6.2 Прогнозування часових рядів, що мають пропущені та аномальні значення, в режимі реального часу

Прогнозування часових рядів дозволяє вирішити завдання визначення майбутнього стану різних систем на основі аналізу вже наявних ретроспективних даних. Здобуття достовірного прогнозу необхідне для ефективного прийняття рішень, та залишається актуальною науково-технічною проблемою [234].

При цьому пред'являються підвищені вимоги до якості рядів, у яких нерідкі пропуски за організаційними або технічними причинами, внаслідок чого в безперервному ряді спостережень створюються розриви різної величини, заповнення яких за наявними даними вносить залежність між спостереженнями і приводить до таких недоліків, як неспроможність оцінок параметрів [234-235]. Аналіз часового ряду після виключення спостережень з пропусками може привести до одержання некоректного прогнозу, і як наслідок, до прийняття невірних рішень.

Наявність аномальних значень у ретроспективних даних є однією з проблем завдання прогнозування, для вирішення якої необхідно використовувати підхід, що дозволяє обробляти викривлені дані таким чином, щоб вони мали мінімальний вплив на результат роботи моделі, що прогнозує. Виявлення раніше невідомих аномалій у режимі реального часу для більшості існуючих методів (статистичні методи аналізу, регресійний аналіз, аналіз сингулярного спектру та інші) є важко здійснюваним завданням [237].

Одним з визначальних факторів одержання достовірного прогнозу є обґрунтований вибір серед існуючих методу прогнозування. На різних фрагментах часового ряду один метод може мати різну ефективність, тому

доцільно проведення сегментації вихідного ряду та використання деякої множини різних предикторів [238]. Результати, отримані в [239], показали, що комбінація простих методів часто показує результат точніший, ніж трудомісткі методи прогнозування.

Перспективним підходом для вирішення завдання прогнозування є використання методу виводу на основі прецедентів (case based reasoning – CBR) [240], у якому висновок відносно поточного завдання виконується за результатами пошуку аналогій – прецедентів, наявних в базі. Проте у прогнозування за прецедентами є ряд недоліків, основні з яких – вимога до кількості відомих значень часового ряду (інакше в базі може бути відсутнім необхідний прецедент) і припущення про те, що закономірності, що діяли у минулому, зберігаються і в майбутньому.

Для вирішення завдання прогнозування часових рядів активно розвиваються підходи на основі засобів штучного інтелекту, таких як штучні нейронні мережі і штучні імунні системи (ШИС) [241-246], які можуть інтегруватися з іншими підходами та відрізняються швидкодією та адаптаційними можливостями. Існують різні моделі, засновані на принципах роботи імунної системи: модель клонального відбору, модель імунної мережі та інші, які можна використовувати для вирішення завдання прогнозування часових рядів.

Не зважаючи на те, що розвиток апаратних та програмних засобів надає можливість використовувати більш потужні обчислювальні платформи, на яких можлива реалізація складних алгоритмів прогнозування, при зростаючому об'ємі вихідних даних існує необхідність у вирішенні задачі короткострокового прогнозування в умовах обмежених обчислювальних і часових ресурсів. Тому актуальним напрямом вирішення даної проблеми є паралельне програмування для батаядерних та багатопроцесорних кластерних систем. Проте при написанні паралельних програм алгоритми вирішення навіть простих завдань перестають бути тривіальними [247-249].

Метою проведеного дослідження є розробка та дослідження

ефективності гібридного методу короткострокового прогнозування часових рядів за умов наявності пропусків та викривленістю аномальними значеннями, на основі моделі клонального відбору, що використовує різнорідні антитіла, побудовані з використанням методу виводу за прецедентами та простих методів прогнозування.

6.2.1 Постановка задачі прогнозування часових рядів

Набір даних, що описує протікання тривалого процесу, може бути представлений у вигляді часового ряду $Z(t) = z_1, z_2, \dots, z_N$ довжини N , і S зовнішніх факторів, представлених у вигляді рядів $X_1(t), X_2(t), \dots, X_S(t)$, значення яких набуті в моменти часу t_1, t_2, \dots, t_N . Набір послідовних значень $Z_t^L = z_t, z_{t+1}, \dots, z_{t+L-1}$, що лежать усередині часового ряду, назвемо вибіркою із цього ряду, яка має довжину L , з моментом початку відліку t , $L \in \overline{[N-1]}$, $t \in \overline{[N-L]}$. Пропуск визначається як елемент вибірки, що не має значення, але не може бути виключеним з вибірки. Аномальний викид визначається як значення часового ряду, що істотно відрізняється від інших елементів вибірки [237]. Необхідно на підставі наявних даних побудувати модель, яка дозволить визначати викривлення, та виконувати прогнозування ряду.

Прогнозування за допомогою СВР ґрунтується на гіпотезі [250]: якщо мірило подоби між вибірками значень прогнозованого ряду Z_t^L та Z_{t-k}^L , а також між відповідними вибірками значень зовнішніх факторів X_t^L і X_{t-k}^L , має значення, близьке до одиниці, то міра подоби між вибірками довжини P , наступними за вибірками вихідного ряду Z_{t+L}^P та Z_{t-k+L}^P , також близька до одиниці. Тобто шляхом визначення вибірки, максимально відповідної останнім значенням часового ряду, можлива оцінка його майбутніх значень.

Ставиться завдання короткострокового прогнозування часового ряду $Z_1^N = z_1, z_2, \dots, z_N$, що має невідому кількість пропущених значень і аномалій,

в режимі реального часу, за допомогою моделей ШС, зокрема, моделі клонального відбору, що використовує різномірні антитіла, побудовані з використанням виводу за прецедентами та простих методів прогнозування.

6.2.2 Виявлення аномалій за допомогою моделей ШС

Перспективним підходом для виявлення аномальних викидів у часових рядах є використання моделей штучних імунних систем. Зокрема, підхід на основі моделі позитивного відбору [251], полягає в генерації детекторів, відповідних вибіркам значень часового ряду, що не містять аномальних викидів. Метод включає етапи навчання та розпізнавання. В ході навчання випадковим чином створюються детектори, які зіставляються з вибірками значень, що складають часовий ряд, що аналізується. Якщо значення афінності менше заданого граничного значення, новий детектор усувається з популяції. Створення випадкових детекторів та їх подальший відбір проводиться до формування набору детекторів, що описують аналізований ряд. В процесі розпізнавання вибірка приймається такою, що містить аномальні викиди в тому випадку, якщо її афінність детекторам менше граничного значення. Недоліком даного підходу є необхідність навчальної вибірки, інформацію, що включає аномальні викиди [252].

В разі використання підходу CBR для вирішення завдання виявлення аномалій, прецедент містить вибірку значень часового ряду і її характеристику (містить дана вибірка аномальні викиди чи ні).

Модель клонального відбору ґрунтується на пошуку антитіл (варіантів рішення), найбільш відповідних антигену (поставленому завданню), ґрунтуючись на значенні функції афінності (мірі подоби між антитілом та антигеном). Антигеном є фрагмент часового ряду $Ag_j = ag_1, ag_2, ag_3, \dots, ag_L$, де j – індекс антигена в популяції, L – довжина вибірки значень ряду, що перевіряються. Антитіло – основний елемент ШС, в термінах підходу CBR,

виконує роль прецеденту, складається з двох частин. Перша частина ab_1, ab_2, \dots, ab_L аналогічна по структурі антигену, визначає міру близькості антитіла антигену, за допомогою визначення функції афінності.

Афінність – скалярна величина, міра близькості між антитілом та антигеном (основний критерій відбору антитіл), що визначається як

$$Aff = \frac{\sum_{k=1}^L |ab_k - ag_k|}{L} \in [0, 1], \quad (6.16)$$

Друга частина антитіла містить його ідентифікатор (отриманий при створенні), значення міри відхилення D_{Ab} , що показує близькість антитіла до аномального, $D_{Ab} \in (0, 1]$, та ідентифікатор того антитіла, на підставі порівняння з яким визначається поточне значення D_{Ab} . Дана частина не бере участь у визначенні афінності.

Алгоритм виявлення аномальних значень наступний:

1. Створення початкової популяції антитіл на основі значень часового ряду, що аналізується. При цьому друга частина створених антитіл на даному етапі містить лише їх власний ідентифікатор. Передбачається, що створені антитіла не містять аномальних значень. Невживані значення вихідного ряду служать повчальною та контрольною вибірками. За наявності тренда у вихідному часовому ряду рекомендується використовувати ряд приростів.

2. Формування навчальної вибірки. Можливе налаштування моделі як шляхом навчання з вчителем (коли відомо, які антитіла з навчальної вибірки відповідають аномальним фрагментам ряду), так і за допомогою навчання без вчителя. У тому випадку, коли невідомо, які дані є аномальними, ШС самостійно формує навчальну вибірку шляхом обчислення матриці афінностей між антитілами. Ті з них, середня афінність яких найменша (менше певного порогу, що говорить про те, що дані антитіла істотно відрізняються від останніх в популяції), приймаються аномальними та

складають навчальну вибірку.

3. Додавання в популяцію антитіл, що містять аномальні значення. При цьому відбувається перевірка афінностей між ними і присутніми в популяції антитілами, значення D_{Ab} яких досі не задане або менше одиниці. У випадку, якщо $D_{Ab'} * Aff < D_{Ab}$, $D_{Ab} = D_{Ab'}$. Після додавання усіх аномальних антитіл, кожне антитіло в популяції має своє значення міри близькості до аномального, $D_{Ab} \in (0,1]$.

4. Перевірка антигена, що представляє собою вибірку часового ряду, про яку невідомо, чи містить вона аномальні значення. Виконується відбір антитіл, що мають найбільшу афінність до антигена. Відбір так само проходять антитіла, на які посилаються (згідно із значенням ідентифікатора антитіла, з яким порівнювались) відібрані. Якщо значення для антитіла, що має найвищу афінність, вище граничного, приймається рішення про присутність в антигені аномальних значень. Висновок робиться не про одне аномальне значення ряду, а про аномальну послідовність, оскільки перевіряється весь антиген, а не його окрема частина.

5. Після отримання інформації про те, є антиген аномальним чи ні, виконується корекція моделі, залежно від прийнятого рішення:

6а. Антиген правильно визначений як той, що містить, або не містить аномальні викиди. Слідує клонування відібраних антитіл, вживання не напрямленої зворотно пропорційної мутації, значення клонів розраховується на підставі афінності з антигеном, що перевіряється, або наявними в популяції антитілами з найбільшими значеннями міри відхилення.

6б. Антиген помилково визначений як той, що містить аномалії. Відібрані антитіла клонуються і після цього усуваються з популяції. Значення D_{Ab} клонів розраховується на підставі афінності з присутніми в популяції антитілами, що містять аномальні значення.

6г. Антиген помилково визначений як той, що не містить аномалій. У популяцію додається нове антитіло, що має значення $D_{Ab} = 1$, та відбувається

перерахунок міри відхилення для всіх антитіл популяції.

7. За наявності нового антигена, що перевіряється, відбувається повернення до пункту 4.

Даний підхід дозволяє виявляти аномалії в часових рядах в умовах відсутності прикладів аномальних значень і недостатньої інформації про критерій, по якому можливо розрізнити нормальні і аномальні вибірки значень часового ряду.

6.2.3 Гібридна модель прогнозування часових рядів на основі моделі клонального відбору

Прогнозування на основі моделі клонального відбору засноване на пошуку антитіл (варіантів рішення), найбільш відповідних антигену (поставленому завданню), ґрунтуючись на значенні функції афінності (міри близькості між антитілом і антигеном) [252-255].

Антиген включає вибірку відомих значень часового ряду, безпосередньо попередніх прогнозованим, та відповідні вибірки значень всіх зовнішніх чинників:

$$\begin{aligned}
 Ag_j &= ag_1, ag_2, ag_3, \dots, ag_L, \\
 ag_{\langle 1 \rangle}, ag_{\langle 2 \rangle}, ag_{\langle 3 \rangle}, \dots, ag_{\langle L \rangle}, \\
 &\dots \\
 ag_{\langle S \rangle 1}, ag_{\langle S \rangle 2}, ag_{\langle S \rangle 3}, \dots, ag_{\langle S \rangle L},
 \end{aligned}
 \tag{6.17}$$

де Ag_j – антиген, j – його індекс в популяції антигенів, S – число зовнішніх факторів, L – довжина вибірки. Тобто антигеном є сукупність вибірок значень прогнозованого і супутніх рядів. Дані вибірки можуть включати невідоме число пропусків.

Антитіло складається з двох частин. Перша частина

$ab_1, ab_2, \dots, ab_L, ab'_1, ab'_2, \dots, ab'_L$ за структурою аналогічна антигену (але включає вибірку значень лише одного зовнішнього фактора $ab'_1, ab'_2, \dots, ab'_L$), є набором параметрів, що описують поставлене завдання (у нашому випадку це вибірки відомих значень ряду, включаючи пропущені значення) та використовується при визначенні афінності.

Друга частина $ab_{L+1}, \dots, ab_{L+f}$, довжина якої дорівнює горизонту прогнозування, не впливає на обчислювальне значення афінності та описує пропонований антитілом прогноз для тієї вибірки значень часового ряду, що складає його першу частину.

Загальне число антитіл в популяції за відсутності пропущених значень часового ряду визначається таким чином:

$$n = \sum_{i=L_{min}}^{L_{max}} (N - C_i + k) \cdot 1 \cdot N_m \cdot F, \quad (6.18)$$

де N – кількість відомих значень прогнозованого часового ряду, L – довжина частини антитіла, що бере участь у визначенні афінності, $L \in [L_{min}, L_{max}]$; k – величина горизонту прогнозування (довжина вибірки прогнозованих значень ряду); N_m – число методів прогнозування використаних в моделі [6]; F – кількість зовнішніх факторів, що потенційно впливають на величину, що прогнозується.

Кожне антитіло має свій «вік» – кількість запусків алгоритму (спроб отримати прогноз) з моменту його появи в популяції. Якщо «вік» вище за граничне значення та антитіло не входить до числа кліток пам'яті, воно усувається з популяції в ході роботи алгоритму.

Афінність визначається з урахуванням кількості пропущених значень у складі антитіла та вагових коефіцієнтів для вибірок, що представляють різні зовнішні фактори:

$$Aff \left(\mathbb{A}b \right) \approx \left(-n_m^{-1} \right) * \eta * \left(\eta_{Ab} * Aff_{Ab} + \eta_{Ab'} * Aff_{Ab'} \right) \quad (6.19)$$

де n_m – число пропусків у вибірці, на основі якої створюється антитіло; η – значення коефіцієнта відбору; η_{Ab} і $\eta_{Ab'}$ – коефіцієнти, що визначають вплив вибірок вихідного та супутнього рядів на афінність антитіла, при цьому $\eta_{Ab} + \eta_{Ab'} = 1$. У антигені можуть бути не представлені деякі зовнішні чинники, а в окремому антитілі представлена лише одна супутня вибірка.

Коефіцієнт відбору η призначений для визначення пріоритету антитіл різних типів, оскільки антитіла, створені на основі однієї і тієї ж вибірки, матимуть однакове значення афінності $Aff_{Ab'}$. Міра подоби вибірок прогнозованого часового ряду Aff_{Ab} і вибірок рядів значень зовнішніх чинників $Aff_{Ab'}$ визначається згідно (1). Як результат (пропонований варіант прогнозу) в даному поколінні приймається вибірка значень $ab_{L+1}, ab_{L+2}, \dots, ab_{L+f}$, що входять до складу антитіл, афінність яких $Aff \left(\mathbb{A}b \right) \rightarrow 1$.

6.2.4 Здобуття прогнозу та процес навчання ШС

На початковому етапі здобуття прогнозу потрібно виявити аномальні значення у часовому ряді. Для створення популяції антитіл використовується частина відомих значень ряду. Невживані значення виконують роль навчальної вибірки (якщо відомо, які антитіла містять аномальні фрагменти) та контрольної вибірки. Якщо навчання із вчителем неможливе, ШС формує навчальну вибірку шляхом обчислення матриці афінностей між антитілами.

Після визначення аномальних значень формується антиген на основі вибірки значень ряду, попередніх прогнозованим, та відбувається вибір антитіл, що мають значення афінності вище граничного. Результатом є прогноз антитіла, що має найбільшу афінність до заданого антигену.

Після набуття реальних значень прогнозу відбувається корекція коефіцієнтів, що впливають на розрахунок значення афінностей антитіл. Той тип антитіл, представник якого серед популяції відібраних показав найменшу помилку прогнозу (навіть якщо не його результат був прийнятий), отримує тимчасовий приріст коефіцієнта відбору η протягом віку антитіла. При подальших відборах антитіл представники даного типа матимуть переваги перед іншими при визначенні афінності, тобто перевага буде віддана тим антитілам, які використовували для здобуття свого варіанту прогнозу метод, який успішно використовувався на попередніх ітераціях.

У разі врахування зовнішніх факторів відбувається корекція значень вагових коефіцієнтів, що визначають афінність вибірок, що складають антитіло. Корекція вагового коефіцієнта в ході навчання ШС дозволяє знизити вплив того чи іншого зовнішнього чинника на пропонований варіант прогнозу, шляхом витіснення з популяції антитіл, які включають вибірку значень саме цього зовнішнього чинника.

При визначенні афінності пріоритет віддається антитілам, побудованим на основі вибірок з меншим числом пропусків, та в процесі навчання шаблони, що мають пропуски, будуть заміщені близькими до них, але з меншою кількістю пропущених значень.

Зміна популяції антитіл в поточному поколінні, що відбувається в результаті вживання операторів клонування, мутації та відбору, в загальному вигляді можна представити таким чином:

$$Ab^{gen+1} = Edit \left(Mutate \left(Clone \left(Sel \left(Ab^{gen} \right) \right) \right) \right) \quad (6.20)$$

де

$$Clone : Ab^{gen} \rightarrow Ab_C^{gen}; \quad (6.21)$$

$$Mutate : Ab_C^{gen} \rightarrow Ab_{MC}^{gen}; \quad (6.22)$$

$$Edit : (Ab_{MC}^{gen}, Ab^{gen}) \rightarrow Ab^{gen+1}. \quad (6.23)$$

Для антитіл, які використовують CBR, застосовується направлена прямо пропорційна мутація, якої піддається лише та частина антитіла, що визначає його прогноз та не бере участь у визначенні афінності. Для антитіл, що обчислюють свій варіант прогнозу самостійно, не напрямленій мутації піддається лише перша частина (і відповідно змінюється пропонований варіант прогнозу), що частково вирішує проблему недоліку прецедентів в базі.

При значному розмірі популяції антитіл доцільне вживання оператора супресії – визначення афінності між антитілами та подальше зменшення їх надмірності:

$$Suppres : Ab^{gen+1} \rightarrow Ab_{sup}^{gen+1}, \quad (6.24)$$

де Ab_{sup}^{gen+1} – популяція антитіл після виконання супресії.

Антитіла, що залишилися – клітки пам'яті, популяція яких формується в процесі навчання ШС. Ці антитіла мають найбільше значення афінності, включають найбільш затребуваний зовнішній чинник та найменше число пропусків. Клітки пам'яті – шаблони, що описують аналізований ряд.

В ході навчання ШС змінюється доля антитіл різних типів, присутніх в популяції. На рис. 6.2 показана зміна числа антитіл, створених за допомогою виводу по прецедентах (CBR), «наївной» моделі (Naive1), простого середнього (Avg) та експоненціального середнього (Exavg) в процесі навчання ШС. Можна спостерігати скорочення кількості антитіл, що використовують методи прогнозування на основі середніх значень.

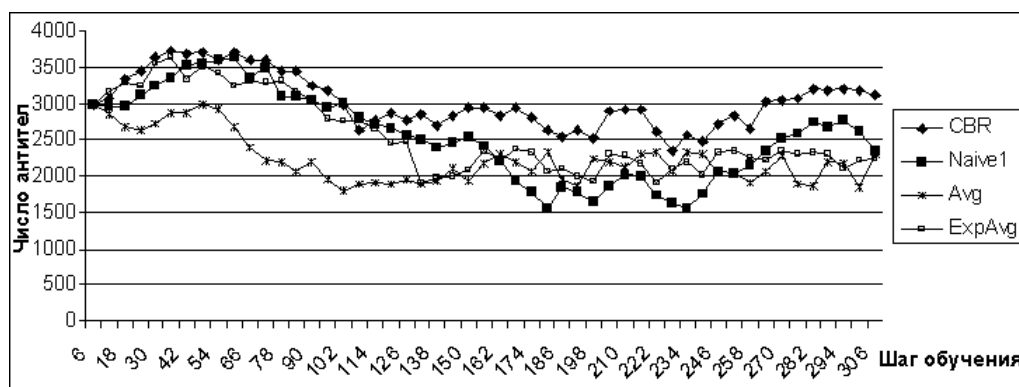


Рисунок 6.2 – Зміна числа антитіл різного типу в ході навчання

В результаті налаштування коефіцієнта відбору в ході навчання моделі для кожного сегменту вихідного ряду будуть вибрані методи прогнозування, що пропонують прогноз з найменшою погрішністю, можливою на даному етапі навчання моделі. Своєчасна зміна методу прогнозування дозволяє врахувати зміни в основній структурі ряду, що дозволяє досягти більшої точності в порівнянні із вживанням окремо кожного предиктора із заданого набору для прогнозування всього вихідного часового ряду.

Навчання ШС повторюється для кожного антигена з навчальної вибірки задане число разів, або до досягнення деякого заданого значення середньої абсолютної помилки.

6.2.5 Розподілений алгоритм прогнозування

Прогнозування часових рядів за допомогою запропонованого імунного алгоритму є важкою обчислювальною процедурою, оскільки спочатку потрібно вивчити систему на великій навчальній вибірці, що вимагає популяції антитіл значного розміру. Для вирішення поставленого завдання доцільно використовувати засоби для організації розподілених обчислень [247], такі як бібліотека MPI, та бібліотеку розпаралелювання завдань TPL [248]. TPL дозволяє істотно полегшити написання легко підтримуваних програм, що автоматично використовують переваги багатоядерних систем

без переробки архітектури додатка.

У даній роботі розглянуто два способу паралельної реалізації імунного алгоритму, які побудовані на основі острівної моделі генетичного алгоритму [249]. Використання паралелізму за даними у вирішуваному завданні полягає в рівномірному (за винятком вузла, що управляє) розподілі популяції антитіл по обчислювальних вузлах та подальшому незалежному виконанні кожним вузлом еволюційного процесу. Це дозволяє обробляти популяцію антитіл великого розміру, для якої вживання послідовної версії імунного алгоритму потребує значних часових витрат. Загальним принципом запропонованих у [254] засобів розпаралелювання є розподіл етапів імунного короткострокового прогнозування між одним вузлом, що управляє, та множиною обчислювальних вузлів. Процес, що управляє, розподіляє початкові дані та формує результат після закінчення роботи N_{cn} розрахункових вузлів, і не має своєї частини популяції антитіл для обробки. Алгоритм першого способу розпаралелювання імунного алгоритму наступний:

1. На кожному розрахунковому вузлі формується унікальна популяція антитіл розміру n_1 .

2. Вузол, що управляє, здійснює розсилку антигена всім обчислювальним вузлам.

3. Кожен розрахунковий вузол виконує відбір антитіл з найбільшою афінністю та відправляє вузлу, що управляє, популяцію клонів розміру $n_c n'$

4. Вузол, що управляє, порівнює запропоновані варіанти прогнозу, вибирає антитіло з найбільшою афінністю та формує результат.

5. За наявності реального значення прогнозованої величини вузол, що управляє, порівнює варіанти прогнозу відібраних антитіл з реальними значеннями. Виконується мутація антитіл з популяції клонів, корекція коефіцієнтів, що визначають вплив використовуваного в антитілі методу та зовнішнього чинника на афінність антитіла [239].

6. Вузол, що управляє, приймає рішення про включення (або видалення) клонів в популяцію антитіл. Формуються популяції кліток пам'яті Ab^M для кожного вузла.

7. Популяції кліток пам'яті та змінені параметри для розрахунку афінності розсилаються розрахунковим вузлам.

8. Кроки 2-7 повторюються для кожного антигена з навчальної вибірки до досягнення одного з критеріїв припинення роботи ШС – формування заданого числа поколінь або зменшення помилки прогнозу до заданого користувачем значення.

Після завершення етапу навчання порядок взаємодії між вузлами залишається тим самим. Даний засіб розпаралелювання дозволяє підтримувати унікальні популяції антитіл на кожному розрахунковому вузлі, розподіливши між вузлами лише операції визначення афінностей та відбору антитіл.

Для оцінки часу обміну даними між обчислювальними вузлами використана модель, запропонована Хокні [247]. оскільки об'ємом контрольних даних (корекція коефіцієнтів відбору) можна нехтувати, тривалість виконання комунікаційних операцій в процесі навчання ШС для першого варіанту паралельного алгоритму може бути виражена як

$$T_1 = N_{cn} - 1 \left(N_g \left(\alpha + \frac{V_{ag} + V_{ab}n_c n' + V_{ab}n_m}{\beta} \right) \right), \quad (6.25)$$

де N_{cn} – число обчислювальних вузлів, N_g – число ітерацій алгоритму в ході навчання ШС, V_{ag} – об'єм даних для передачі одного антигену, V_{ab} – об'єм даних для передачі одного антитіла, n_c – число клонів одного антитіла, n' – число антитіл, відібраних для клонування, n_m – розмір популяції кліток пам'яті, n_m – кількість антиген навчальної вибірки, α і β – параметри

моделі Хокні.

К перевагам цього підходу відносяться унікальність антитіл в популяціях на різних вузлах, кожен з яких має незалежну популяцію кліток пам'яті, та можливість контролю параметрів алгоритму для кожного вузла на кожній ітерації. Проте контроль процесом, що керує, над популяціями на розрахункових вузлах приводить до значного завантаження мережі, оскільки на кожній ітерації алгоритму відбувається пересилка великої кількості даних.

З метою скорочення комунікаційних операцій, у другому способі розпаралелювання імунного алгоритму пропонується розподіл процесу навчання ШС та управління популяцією між окремими обчислювальними вузлами, без обміну проміжними результатами з вузлом, що керує:

1. На кожному розрахунковому вузлі формується унікальна популяція антитіл розміру n_1 .

2. Вузол, що управляє, здійснює розсилку навчальної вибірки (множини антигенів та реальних значень прогнозованої величини) всім розрахунковим вузлам. На кожному вузлі виконуються кроки 3-7:

3. Відбір і клонування антитіл, що мають найбільшу афінність антигену.

4. Мутація антитіл з популяції клонів розміру $n_c n'$.

5. Корекція коефіцієнтів, що визначають вплив використаного в антитілі методу прогнозування та представленого зовнішнього чинника на розрахунок афінності антитіл.

6. Вузол, що керує, приймає рішення про включення або видалення клонів з популяції незалежно від популяцій антитіл на інших розрахункових вузлах.

7. Формування популяції кліток пам'яті Ab^M .

8. Кроки 3-7 повторюються для кожного антигена з навчальної вибірки до досягнення одного з критеріїв припинення роботи ШС.

Після завершення етапу навчання взаємодія між розрахунковими

вузлами та вузлом, що керує, відбувається аналогічно крокам 2-7 першого варіанту паралельного алгоритму. Розрахункові вузли, завершивши роботу, відправляють кращий результат вузлу, що керує, який формує результат роботи імунного алгоритму та регулює параметри відбору на розрахункових вузлах.

Основною перевагою цього підходу є мале навантаження мережі в процесі навчання – передача даних обмежується навчальною вибіркою, що складається з m антиген:

$$T_2 = N_{cn} - 1 \left(\alpha + \frac{mV_{ag}}{\beta} \right). \quad (6.26)$$

Недоліком даного підходу є можливість появи однакових антитіл в популяціях на різних вузлах у ході навчання ШС, а вживання операції супресії потребує додаткових обмінів даними між вузлами. Крім того, спостерігається нерівномірне навантаження вузлів в кінці етапу навчання: після досягнення заданої середньої афінності одним процесом, виникає зупинка в очікуванні завершення роботи інших.

Оскільки запропоновані варіанти паралельної реалізації імунного алгоритму відрізняються, головним чином, кількістю інформаційних обмінів в процесі навчання ШС, а надалі обмін даними відбувається однаково, то для порівняння підходів досить визначити тривалість операцій пересилки даних. Порівняння отриманих виразів 6.25 та 6.26 показує, що другий розроблений засіб паралельних обчислень має істотно менші комунікаційні витрати і допускає кращу масштабованість при збільшенні кількості використовуваних процесорів. Тому подальші дослідження проводяться із застосуванням другого засобу розпаралелювання імунного алгоритму.

6.2.6 Результати експериментальних досліджень

В ході експериментальних досліджень було проведено короткострокове прогнозування рядів, використовуваних в M3-competition, і порівняння отриманих результатів з приведеними в [239] результатами прогнозування за допомогою експоненціального згладжування, варіантів моделі Хольта (RobustTrend, HoltWinters, CombSHD), моделі ARIMA, експертних систем (ForecastPro і SmartFcs), штучних нейронних мереж. Також було виконано прогнозування ряду середньодобових значень свідчень температури Meteo, з врахуванням внесених пропущених значень. Аналізувалися два варіанти розташування пропусків – у всьому ряду, і лише в контрольній частині (для випадку, коли була можливість перед використанням створити та навчити ШС на неушкоджених даних). Симетричні середні абсолютні помилки прогнозування приведені в табл. 6.4, для ряду Meteo вказана середня абсолютна помилка (MAE, °C).

Серед використовуваних для аналізу рядів лише ряд середньодобових показань температури Meteo містить кількість значень, достатню для створення популяції антитіл, відповідній великій базі прецедентів, і проведення навчання ШС. Результати прогнозування даного ряду підтверджують перевагу використання запропонованого підходу на основі моделі клонального відбору. Проте, при малій кількості значень часового ряду, імовірна відсутність в базі антитіл з високою афінністю, а величина навчальної вибірки не дозволяє повною мірою навчити ШС. Внаслідок цього на деяких рядах спостерігається перевага традиційних методів прогнозування перед запронованим підходом.

Наявність пропусків веде до збільшення помилки на 2-8%,але якщо пропущені значення розташовані лише в контрольній частині ряду (результати для ClonAlg (10%),control), помилка збільшується лише на 1-3% відсотка.

Таблиця 6.4 – Симетрична середня абсолютна помилка (SMAPE) при використанні різних методів при різній кількості пропущених значень

Метод	N704 (44)	N736 (44)	N1366 (63)	N2830 (104)	N2841 (104)	N2867 (79)	Meteo (21337)
Exp.Smooth	4,08	12,11	0,42	2,47	0,5	20,52	4,56
RobustTrend	4,76	8,88	0,41	2,18	0,52	18,83	–
HoltWinters	4,92	10,68	1,04	3,27	0,39	20,08	2,9
CombSHD	4,36	9,65	0,5	2,73	0,46	20,09	–
Box–Jenkins	3,66	7,35	0,57	2,45	0,5	26,13	2,99
ForecastPro	3,13	6,5	0,41	2,47	0,5	20,52	–
SmartFcs	5,18	8,62	0,29	2,47	0,5	23,85	–
AutoANN	3,42	8,73	0,27	1,56	0,53	26,11	–
ClonAlg	5,20	7,65	0,41	1,83	0,14	16,26	2,44
ClonAlg (10%)	5,92	9,74	0,91	2,34	0,21	18,01	-
ClonAlg (10%), cntrl	5,20	8,45	0,41	1,83	0,23	18,35	2,44
ClonAlg (15%)	6,42	7,31	0,88	2,98	0,21	19,01	2,56
ClonAlg (20%)	6,30	8,15	0,56	3,31	0,53	20,09	3,77

Аномальні значення для експериментів (результати приведені в таблиці 6.5) вносилися шляхом збільшення або зменшення випадкового значення на 20% від $z_{max} - z_{min}$. Експерименти проводилися як для випадку розміщення аномальних значень у всьому прогнозованому ряду, так і лише в контрольній частині (дозволяючи виконувати навчання з вчителем).

Унаслідок виявлення більшості аномалій, результати, як правило, відповідають результатам прогнозування часових рядів, що містять пропущені значення.

Таблиця 6.5 – Симетрична середня абсолютна помилка (SMAPE) при використанні різних методів при різній кількості аномальних значень

Метод	N704 (44)	N736 (44)	N1366 (63)	N2830 (104)	N2841 (104)	N2867 (79)	Meteo (21337)
Exp.Smooth	4,08	12,11	0,42	2,47	0,5	20,52	4,56
Exp.Smooth (10%)	4,3	12,5	0,45	2,48	0,59	21,13	4,7
HoltWinters	4,92	10,68	1,04	3,27	0,39	20,08	2,9
HoltWinters (10%)	5,36	12,04	1,18	3,51	0,44	22,31	3,13
Box–Jenkins	3,66	7,35	0,57	2,45	0,5	26,13	2,99
Box–Jenkins (10%)	4,02	7,85	0,65	2,48	0,53	27,76	3,35
ClonAlg	5,2	7,65	0,41	1,83	0,14	16,26	2,44
ClonAlg (10%)	5,92	9,74	0,91	2,34	0,21	18,01	-
ClonAlg (10%), cntrl	5,2	8,45	0,41	1,83	0,23	18,35	2,44
ClonAlg (15%)	6,42	7,31	0,9	2,98	0,21	20,19	2,56
ClonAlg (20%)	6,39	8,26	0,56	3,31	0,53	21,73	3,77

Для перевірки ефективності паралельного алгоритму був обраний ряд середньодобових значень температури (Meteo, більше 20000 значень), який дозволяє створити обширну навчальну вибірку. Для обчислювального експерименту в даній роботі був використаний кластер, побудований на основі Microsoft Windows Compute Cluster Server 2003 (MS CCS). Ця система забезпечує безпечне та легко масштабоване управління ресурсами кластера, планування завдань та надає інтерфейс передачі повідомлень для розподіленого програмування [247].

Обчислювальний експеримент полягав у послідовному запуску паралельної версії алгоритму з однаковим критерієм припинення роботи ШС та максимальним (для послідовної версії) розміром популяції на різному числі доступних обчислювальних вузлів кластера. Використовувалося від

одного до двадцяти обчислювальних вузлів.

На рис. 6.3 показана залежність часу навчання моделі прогнозування часових рядів від задіяного числа процесорів, що свідчить про прискорення, близьке до лінійного. При зменшенні числа антитіл в популяціях на окремих розрахункових вузлах потенційно можливе надлінійне прискорення.

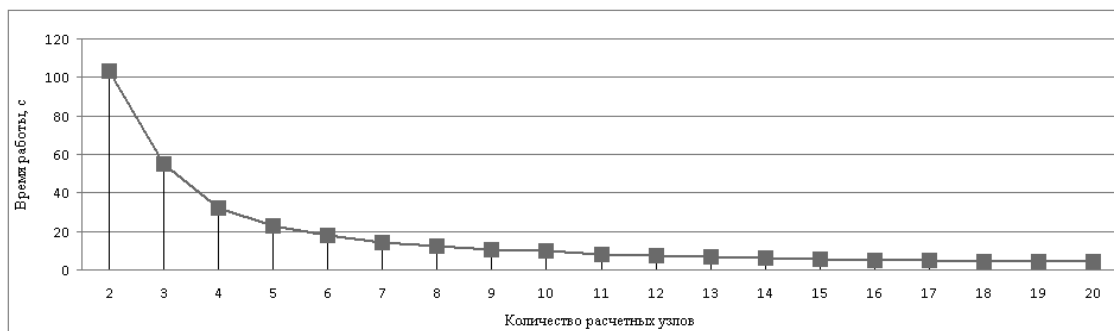


Рисунок 6.3 – Залежність часу навчання від числа процесорів

В результаті вживання імунного алгоритму, адаптованого для використання технології TPL, середнє прискорення досягло 7.5%. Рис. 6.4 ілюструє різницю часу навчання послідовної та адаптованої для TPL версій алгоритму для 2-ядерного процесора моделі Intel Pentium D, 3000 МГц.

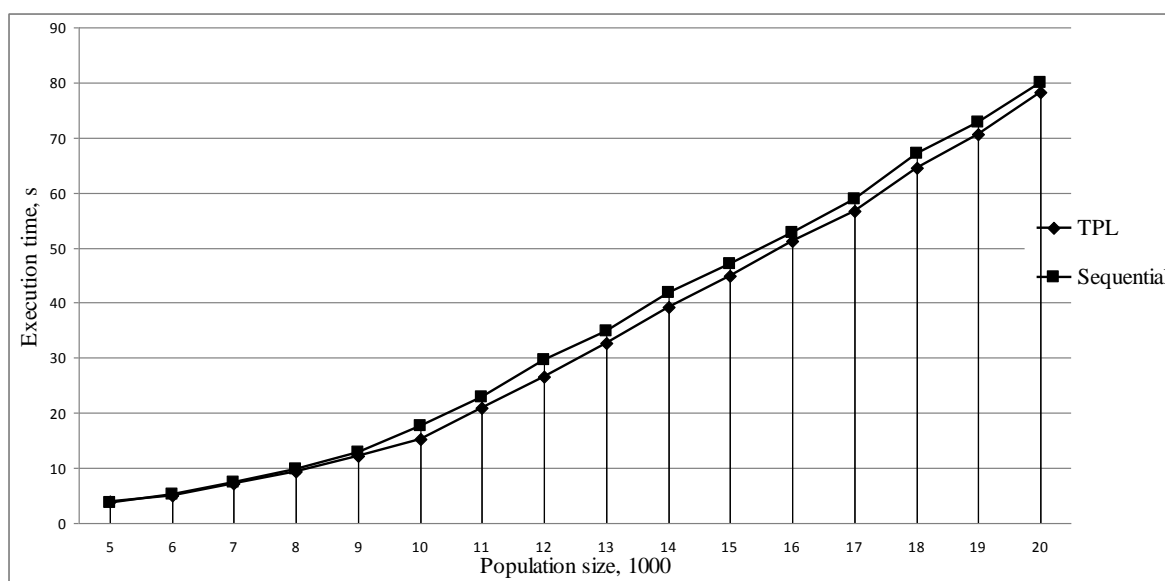


Рисунок 6.4 – Залежність часу навчання імунного алгоритму від розміру популяції

Низька ефективність вживання технології TPL пояснюється тим, що в розглянутих реалізаціях імунного алгоритму змінні більшості циклів є залежними одні від одного, що веде до складності оптимізації керованого коду для багатоядерних процесорів. При цьому, для того, щоб отримати перевагу в швидкодії паралельної реалізації, необхідно, щоб трудомісткість паралельних процесів в областях розпаралелювання істотно перевершувала трудомісткість породження паралельних потоків.

Підсумовуючи вищенаведені результати прогнозування часових рядів, що мають пропущені та аномальні значення, можна зробити наступні висновки:

1. Запропонована гібридна модель короткострокового прогнозування викривлених часових рядів на основі моделі клонального відбору, що використовує різномірні антитіла. Розглянутий метод виявлення аномалій на основі моделі клонального відбору, шляхом визначення афінностей антигена (досліджуваного фрагмента часового ряду) та кліток пам'яті. Можливе налаштування моделі виявлення аномалій як шляхом навчання з учителем, так і за допомогою навчання без учителя, шляхом розрахунку матриці афінності між антитілами. Надалі, в ході отримання прогнозу, при визначенні афінності антитіл використовуються отримані на етапі пошуку аномалій значення міри відхилення антитіла. Даний підхід дозволяє виявляти аномалії в часовому ряду в процесі вирішення задачі короткострокового прогнозування на основі моделі клонального відбору, що використовує метод виводу по прецедентах (CBR).

2. Розглянуто два способи розпаралелювання імунного алгоритму на основі острівної моделі генетичного алгоритму. Проведений аналіз розглянутих засобів розпаралелювання на основі обліку комунікаційної трудомісткості. Особливістю першого засобу розпаралелювання є контроль еволюції ШС на розрахункових вузлах, що дозволяє підтримувати унікальні популяції антитіл, але веде до значних комунікаційних навантажень. Другий

засіб розпаралелювання характеризується мінімізацією комунікаційних витрат усередині кластера шляхом організації незалежного еволюційного процесу на окремих розрахункових вузлах в ході навчання ШС.

3. Для реалізації обміну даними між обчислювальними вузлами використовується технологія MPI.NET, для розрахунків в межах популяцій окремих вузлів застосовується оптимізація керованої коди для багатоядерних процесорів з використанням технології TPL. Використання розподілених обчислень дозволяє добитися прискорення, близького до лінійного, тоді як оптимізація коди для багатоядерної архітектури дозволяє дістати прискорення близько 7%.

7 МЕТОДИ ПОТОЧНОГО КОНТРОЛЮ ПОСЛІДОВНОСТЕЙ ДАНИХ ЗА УМОВ ЇХ ВИКРИВЛЕНОСТІ НА ОСНОВІ КРИТИЧНОГО ПІДХОДУ

7.1 Розробка концепції поточного контролю послідовностей даних за умов викривленості вихідних спостережень

Вирішення задач контролю, ранньої діагностики часто доводиться вирішувати в умовах невизначеності, пов'язаної з неповнотою, недостовірністю і несвоєчасним надходженням інформації. Управління виробничими процесами здійснюється за різною інтенсивністю завад, статистичні характеристики яких не завжди можуть бути визначені і враховані при виробленні керуючих впливів.

Існує значна кількість джерел та форм прояву невизначеності. Вкажем деякі важливі типи невизначеності [256-258]: 1) невизначеність, викликана браком інформації та її достовірності в силу організаційних, технологічних та інших причин; 2) невизначеність, пов'язана з обмеженнями в ситуації прийняття рішення - за часом рішення, заданої точності; 3) невизначеність, що обумовлена занадто високою або недоступний платою за визначеність; 4) невизначеність, що виникає через неадекватність моделі з причин невизначеності мети і критерія функціонування, невизначеність прояви зовнішнього середовища по відношенню до даної системи, невизначеність при виборі раціональної складності моделі; 5) невизначеність, породжувана діями людей в процесі виробництва і прийняття рішень; 6) невизначеність, організована навмисно для приховування поведінки системи та її ресурсів.

Прийняття управлінських рішень при детермінованій інформації у багатьох випадках зводиться до вирішення завдання

$$x^* = \operatorname{argmin} \{ G(x) \mid G(x) \geq b, x \geq 0 \}, \quad (7.1)$$

де x^* - вектор шуканих змінних;

a, b, c - параметри моделі.

Рішення задачі (7.1) на практиці здійснюється в умовах неповної та недостовірної інформації, яка проникає в модель через невизначеність мети, внутрішніх зв'язків і прояви зовнішнього середовища. Наприклад, вектор ресурсів залежить від множини заздалегідь непередбачуваних факторів. При цьому чим більше період планування, тим більше невизначеність в оцінці можливого значення b . Сказане про ресурси $b = \{b_j\}, j = 1, m$ повною мірою відноситься до норм витрат $a = \{a_{ij}\}, i = 1, n, j = 1, m$ і коефіцієнтів цільовій функції $c = \{c_i\}, i = 1, n$.

При верогідніснім завданні параметрів задачу (7.1) можна представити моделлю стохастичного програмування [259]

$$x^* = \underset{x \in X}{\operatorname{argmin}} \left\{ E_{\omega} Q(\omega, x) \mid E_{\omega} G(\omega, x) \geq E_{\omega} b(\omega), x \geq 0 \right\}, \quad (7.2)$$

де x - вектор шуканих змінних;

$E_{\omega}(\bullet)$ - оператор усереднення в умовах дії випадкового фактора ω ; $Q(c(\omega), x)$, $G(a(\omega), x)$, $c(\omega)$, $a(\omega)$, $b(\omega)$ - цільова функція, функція обмежень, коефіцієнти цільової функції, функції обмежень і умов функціонування, що залежать від випадкового фактора, ω відповідно.

Існують завдання, в яких порушення обмежень при якійсь реалізації випадкових умов призводить до дуже небажаних наслідків. У таких випадках під планом розуміються лише вектори x , що задовольняють обмеженням при всіх технологічних умовах $a(\omega)$ і векторах ресурсних обмеженнях $b(\omega)$, що з'являються з ненульовою ймовірністю. Подібного роду жорсткі постановки

визначають клас одноетапних завдань стохастичного програмування. Умови одноетапною завдання мають вигляд

$$a(\omega)x \leq b(\omega), \omega \in \Omega, x > 0, \quad (7.3)$$

де ω - випадковий параметр стану природи, що визначає реалізацію випадкових елементів умов завдання;

Ω - кінцева множина можливих станів природи, що з'являються з ненульовою ймовірністю.

Процедуру пониження невизначеності розглянемо на прикладі методології [260] вирішення двоетапної задачі стохастичного програмування

$$x^* = \underset{x \in X}{\operatorname{argmin}} \{ c(\omega)x \mid A(\omega)x = b(\omega), Ax = b, x \geq 0 \}, \quad (7.4)$$

де $A(\omega), b(\omega), c(\omega)$ - матриця і вектори з випадковими елементами;

A, b, c - те ж з детермінованими елементами.

На першому етапі визначається вектор x з умови мінімізації неявно заданого функціоналу $Q(\omega) = (c(\omega), x)$ при дотриманні детермінованих обмежень $Ax = b, x \geq 0$. Сукупність цих обмежень висікає опуклу множину допустимих попередніх планів. Для вирішення цього завдання необхідно зафіксувати реалізацію випадкової події на підставі апріорної інформації, а за відсутності такої вибирається довільна точка, що належить множині припустимих рішень. Отже, для пониження невизначеності потрібне введення додаткової інформації, яка в свою чергу містить невизначеність, можливо, меншого порядку.

На другому етапі фіксується реалізація ω випадкової події ω (оцінка фактора і здійснюється без якої б то не було інформації про закон розподілу ω , що вносить додаткову невизначеність) і відповідні йому значення $A(\omega)$ і

$b(\omega)$, потім обчислюється вектор нев'язок $y(\omega)$ (згідно виразу $B(\omega)y(\omega) = b(\omega) - A(\omega)x$, де $B(\omega)$ - матриця компенсації, елементи якої в загальному випадку мають стохастичний характер. За порушення умов завдання стягується штраф, залежний від величин складових вектора $y(\omega)$ з урахуванням масштабного множника $q(\omega)$.

Вектор $y(\omega)$ вибирається з умови мінімального штрафу за компенсацію нев'язок умов завдання щодо попереднього плану x завдання.

Математична модель двоетапної задачі лінійного стохастичного програмування

$$x^* = \underset{x \in X}{\operatorname{argmin}} \mathbb{E}_\omega \left[\sum_{i=1}^m q_i(y_i) \right] + \min \left\{ \sum_{i=1}^m q_i(y_i) \mid B(\omega)y(\omega) = b(\omega) - A(\omega)x, y(\omega) \geq 0 \right\} \mid Ax = b, x \geq 0$$

містить ряд невизначеностей, пов'язаних із завданням вихідних даних; вибором структури моделі, можливою її коригуванням у разі несумісних обмежень неєдиності або нестійкості рішення відхиленням вихідних передумов моделювання та оптимізації; необхідністю залучення додаткової інформації в процесі виконання завдання.

На практиці часто зустрічаються випадки, коли не всі компоненти вектора стану доступні вимірюванню через випадкових викидів і викривлення вхідних спостережень. У цій ситуації інформацію про стан системи доводиться витягати з результатів спостереження за вихідними координатами. Завдання відновлення вектора стану $x(t)$ розмірності n динамічної системи

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), x(0) = x_0 \quad (7.5)$$

$$y(t) = C(t)x(t) \quad (7.6)$$

де $y(t)$ - вектор вихідний спостерігається змінної розмірності l , причому $l < n$, полягає у визначенні оператора $F(-)$, що реалізує співвідношення

$$\hat{x}(t) = F(y(\tau), t_0 \leq \tau \leq t, t_0 \leq t) \quad (7.7)$$

де $x(t)$ - відновлений стан, який залежить від попередніх результатів спостережень $y(\tau)$, $t_0 \leq \tau \leq t$ і не залежить від майбутніх спостережень $y(\tau)$, $\tau > t$.

Для відновлення стану $x(t)$ системи (7.6) при відомих матрицях $A(t)$, $B(t)$, $C(t)$ по спостережуваній змінній $y(t)$, яка визначається співвідношенням (7.2), скористаємося спостерігачами Луенберга [261], що представляють собою систему диференціальних рівнянь, структура і поведінка якої вибираються таким чином, щоб отримати відновлюване значення $x(t)$, що має певні властивості.

Твердження 1. Система диференціальних рівнянь n -го порядку

$$\dot{\hat{x}} = A_n \hat{x} + B_n u + K(y - C\hat{x}) \quad (7.8)$$

описує поведінку спостерігача для динамічної системи (7.5), (7.6), якщо при

$$\hat{x}(t_0) = x(t_0) \quad (7.9)$$

система (7.8) забезпечує

$$\hat{x}(t) = x(t) \quad (7.10)$$

для всіх значень $u(t)$, $t \geq 0$, що вимагає дотримання наступних умов:

$$A_n \triangleq A - KCC \quad (7.11)$$

$$B_n \triangleq B \quad (7.12)$$

$$\dot{\tilde{x}} \triangleq A_n \tilde{x} \quad (7.13)$$

де

$$\tilde{x} \triangleq x - \hat{x} \quad (7.14)$$

Структура спостерігача визначається виразом

$$\hat{x} \triangleq A \hat{x} + B \hat{u} + K \left[y - C \hat{x} \right] \quad (7.15)$$

Найважливішою бажаною властивістю спостерігача є асимптотична стійкість, яка визначається поведінкою матриці $A(t) - K(t)C(t)$, що входить в диференціальне рівняння помилки відновлення вектора стану

$$\dot{\tilde{x}} \triangleq A_n \tilde{x} \quad (7.16)$$

Розрахунок спостерігача зводиться до визначення такої матриці коефіцієнтів підсилення $K(t)$ при $t \geq t_0$, яка забезпечить стійкість диференціального рівняння помилки відновлення (7.16), забезпечивши прагнення до нуля $x(t)$ незалежно від початкового стану. Для збільшення запасу стійкості бажано матричний коефіцієнт посилення брати великим, але це в свою чергу робить спостерігач досить чутливим до завад у

спостереженнях. Тому зазвичай виходять з компромісного рішення, що враховує належний запас стійкості і заводо захищеності спостерігача.

Синтез структури спостерігача повного порядку для динамічної системи (7.5) з постійними коефіцієнтами можна виконати згідно з рівнянням

$$\dot{\hat{x}} = [A - LC]\hat{x} + Ly + Bu, \quad (7.17)$$

яке отримано із співвідношення (7.15) шляхом заміни матричного коефіцієнта посилення $K(t)$ векторним коефіцієнтом посилення $L(t)$ компонентами $L_i(t)$, вибраними таким чином, щоб забезпечити бажаний процес відновлення вектора стану.

Для того, щоб синтезуємий спостерігач, крім запасу стійкості, забезпечував задану якість процесу відновлення вектора стану, необхідно виконати канонічне перетворення. Ідея канонічного перетворення виходить з того, що вектор стану може бути утворений різними комбінаціями змінних стану. Застосовуючи до рівнянь

$$\dot{\hat{x}} = A\hat{x} + Bu \quad (7.18)$$

$$y = C\hat{x} \quad (7.19)$$

лінійне перетворення

$$x^* = Tx, \quad (7.20)$$

можна отримати різні вектори стану і відповідні їм вектори спостереження, які є лінійними комбінаціями n компонент вихідного вектора $x(t)$ і T компонент вектора $y(t)$, де T - матриця перетворення.

Нова лінійно перетворена система рівняння (7.18) - (7.20) набуде вигляду

$$\dot{x}^* = A^* x^* + B^* u; \quad (7.21)$$

$$y^* = C^* x^*, \quad (7.22)$$

де

$$A^* = TA T^{-1}; \quad (7.23)$$

$$B^* = T^{-1} B; \quad (7.24)$$

$$C^* = C T. \quad (7.25)$$

Важлива особливість перетворення (7.20) полягає в тому, що власні значення вихідного рівняння (7.18) збігаються з власними значеннями перетвореного рівняння (7.20). Окремим випадком лінійного перетворення (7.20) є канонічне перетворення, що використовує у якості матриці T^{-1} матрицю власних векторів V . Ця матриця виходить з вирішення однорідного лінійного рівняння

$$\dot{x} = A x, \quad x_0 = x_0. \quad (7.26)$$

Рішення рівняння (7.21) можна записати у вигляді

$$x = V \exp(At) = V_1 \exp(\lambda_1 t) + \dots + V_n \exp(\lambda_n t). \quad (7.27)$$

де V_1, \dots, V_n - власні вектори.

Для знаходження невідомих V_1, \dots, V_n , продиференціюємо (7.27) і отриманий результат підставимо в ліву частину рівняння (7.26), замінивши при цьому праву частину (7.26) виразом (7.22). Після деяких перетворень одержимо рівняння

$$\mathbb{A} - \lambda_i I \dot{Y}_i = 0, i = \overline{1, n}. \quad (7.28)$$

7.1.1 Аналіз впливу зовнішнього середовища на процедуру контролю

В задачах комп'ютерної інженерії та управління досить часто виникає ситуація, коли замкнута система управління, що знаходиться під впливом зовнішнього возмущаючого сигналу ω , повинна підтримувати характеристики об'єкта управління v (вихідний сигнал об'єкта, помилка управління тощо) всередині деяких апріорно задаються кордонів так, що

$$|V(t, \omega)| \leq E, \quad t \in R, \quad (7.29)$$

де t - безперервний або дискретний час.

У тому випадку, якщо порушення нерівності (7.29) в принципі не припустимо, наприклад, веде до катастрофічних наслідків, закон керування, який забезпечує жорстке підтримання (7.29), називається критичним, а система управління, його реалізує, - критичною [262, 263].

Задача підтримки вихідних сигналів об'єкта в заданих межах виникла досить давно і для її вирішення був розвинений цілий ряд підходів. Так в [264] було запропоновано статистичний підхід, максимізує ймовірність того, що виходи об'єкта не вийдуть за певні межі при довільних випадкових входах. Відомий метод, заснований на теоретико-множині підході [265], що

використовує концепцію "цільової труби", всередині якої повинні залишатися фазові змінні об'єкту, що знаходиться під впливом невідомих, але обмежувальних збурень.

У загальному випадку мета будь-якої системи управління зі зворотним зв'язком полягає в забезпеченні необхідного поведіння об'єкта шляхом відповідної обробки вхідних і вихідних сигналів, обчислення керуючих впливів і їх подачі на виконавчі органи. Головною проблемою при цьому є проектування власне регулятора, з теоретичної точки зору представляє собою формальний алгоритм, результат роботи якого є чисельне значення керуючого сигналу. Проблема синтезу розпадається на дві відносно незалежні підзадачі, першою з яких є визначення мети управління та його формального представлення - критерію. Друга підзадача полягає в знаходженні формального опису регулятора, що забезпечує необхідне значення цього критерію. Крім основного критерію в розгляд зазвичай вводиться ряд додаткових подцелей, що вимагають, наприклад, щоб замкнута система була стійкою, керуючі сигнали або деякі функції від них не були б занадто великими, ефекти шуму і збурень були б малими, а сама система прагнула до необхідного стану деяким певним чином. Ці критерії звичайно є деякими функціями від вхідних і вихідних сигналів або станів об'єкта управління, при цьому сигнали звичайно покладаються стохастическими процесами з деякою апріорно заданою ймовірнісною структурою. Найбільш часто використовується гіпотеза про гауссовий розподілі корисних сигналів і перешкод. Саме на цій гіпотезі ґрунтується популярна ЛКГ-проблема, H^2 - оптимізаційна проблема і класична стохастична теорія управління. У практичних завданнях гаусові процеси зустрічаються не так вже часто, а їх використання пов'язано в основному з математичними зручностями.

З меншими обмеженнями на статистичний характер сигналів пов'язана H^∞ - оптимізаційна проблема, що вимагає лише, щоб сигнали об'єкта були інтегровними з квадратом і мали обмежену енергію.

Певної гнучкості при створенні систем управління можна досягти, якщо припустити, що сигнали належать деякому функціональному простору. Такий простір може бути визначене шляхом завдання кордонів на амплітуду, швидкість зміни, енергію та інші характеристики сигналів. Такий опис сигналів значно простіше статистичного, має ясний фізичний зміст і в загальному підсумку полегшує процес проектування критичної системи управління.

7.2 Процедури контролю за умів аномальних спостережень на основі критичного підходу

Контроль, здійснюється, як правило, при різних проявах зовнішнього середовища. Важливим фактором, що впливає на якість контролю є наявність аномальних спостережень. Крім того, до основних факторів можна віднести зашумленість, корельованість і мультиколінеарність результатів спостережень. Обчислювальна процедура оцінки ефективності будується таким чином, що кожен раз вибираємо один з алгоритмів параметричного контролю, який аналізується при різному дії зовнішнього середовища. За результатами імітаційного моделювання визначають якісні показники узагальненого критерію ефективності. У результаті статистичної обробки визначається область пріоритетним застосування кожного з алгоритмів.

Для порівняння одержуваних різними способами оцінок в теорії статистичного оцінювання використовуються наступні основні показники [272, 273]: незміщеність, ефективність, спроможність.

При вирішенні практичних завдань оцінювання виникає необхідність у визначенні характеристик точності одержуваних оцінок. Очевидно, що ці характеристики не можуть бути знайдені без помилок, і фактично мова може йти лише про оцінку точності одержуваних оцінок контрольованої величини.

Для прийнятої математичної моделі ідеальне співвідношення між вектором вимірювань $b = \{b_1, b_2, \dots, b_m\}$ і вектором стану $q = \{q_1, q_2, \dots, q_m\}$ має вигляд

$$b = F \bar{q}. \quad (7.30)$$

Проте насправді вимірювання виходять в результаті спостереження за станом реальної системи. Внаслідок цього з'являється помилка

$$\xi_2 = b_n - F \bar{q}_n. \quad (7.31)$$

У вимірах контрольованої величини присутній похибка вимірювань, тому існує різниця ξ_1 між виміряним b і істинним b_n значеннями параметра

$$\xi_1 = \tilde{b} - b_n. \quad (7.32)$$

З (7.31) і (7.32) випливає, що

$$\tilde{b} = F \bar{q}_n + \xi_1 + \xi_2. \quad (7.33)$$

Розглянемо першу складову помилки. Найбільш повно вектор ξ_1 описується n - мірним законом розподілу. Однак у вирішенні практичних завдань складання точного закону розподілу є нездійсненним завданням чинності обмеженого обсягу вибірки вимірюваної величини. Для подолання цієї труднощі в даний час існує два підходи. Перший полягає в тому, що закон розподілу помилок передбачається відомим. При цьому параметри розподілу задаються або визначаються за даними експерименту. Зазвичай в цьому випадку використовується нормальний закон розподілу. Однак

контрольована величина може бути розподілена і за іншими законами, наприклад, рівномірному або Лапласа. Тому більш кращим і більш загальним є другий підхід, при якому задаються перші два моменту розподілу: математичне очікування $M[\xi_l]$ і коваріаційна матриця $D[\xi_l]$.

Якщо помилки вимірювань між собою некорреліровані, матриця приймає вигляд

$$D[\xi_l] = \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n^2 \end{pmatrix} \quad (7.34)$$

Якщо дисперсія помилки по всій довжині вибірки незмінна (випадок гомоскедастичності), матриця (7.44) зводиться до вигляду $D[\xi_l] = \sigma^2 I$.

Поряд з некоррелірованими помилками при вирішенні завдань оцінювання доводиться мати справу з систематичними помилками, що виникають внаслідок того, що вимірювання b залежать не тільки від стану q , а й від вектора заважаючих параметрів $\phi = \{\phi_l\}$, $l = \overline{1, k}$. В цьому випадку

$$b = F(q, \phi) \quad (7.35)$$

Вважаючи $M[\phi] = 0$, розкладемо в ряд Тейлора, обмежуючись першими двома членами

$$b = F(q) + B\phi \quad (7.36)$$

де

$$F(\Phi) = \Phi(\Phi, 0); \quad B = \left(\frac{\partial \Phi}{\partial \varphi} \right)_0 = \left(\frac{\partial b}{\partial \varphi} \right)_0 = \left(\frac{\partial \Phi_i}{\partial \varphi_j} \right)_0.$$

Якщо в якості моделі прийняти співвідношення виду

$$\xi_I^* = B\varphi, \quad (7.37)$$

то з (7.36) і (7.37) під систематичною помилкою будемо мати на увазі величину

$$\xi_I^* = B\varphi. \quad (7.38)$$

З властивостей оператора математичного очікування знаходимо вираз для ковариационної матриці

$$D(\xi_I^*) = BD\Phi B^T. \quad (7.39)$$

Користуючись (7.10), матрицю B представимо у вигляді

$$B = (\beta_1, \beta_2, \dots, \beta_k), \quad (7.40)$$

де β_i - вектор i - й приватної похідної, а вирази (7.39) і (7.40) перепишемо у вигляді

$$\xi_i^* = \sum_{j=1}^k \beta_j \varphi_j \quad D(\xi_i^*) = \sum_{j=1}^k \beta_j \beta_j^T \quad (7.41)$$

Таким чином, систематична помилка являє собою суму похибок, що виникають внаслідок дії кожного з заважають параметрів. Якщо заважають параметри не корельовані між собою, коваріаційна матриця помилки являє собою суму коваріаційних помилок складових.

Поряд з помилками вимірювання на точність одержуваних оцінок в значній мірі впливають помилки ξ_2 моделі, джерелами яких є неповнота використовуваного вектора q стану і похибка модельних залежностей (7.41). Фактичний стан системи визначається нескінченним числом параметрів, з яких частина включена в вектор q , а частина з включається внаслідок недоцільності (ускладнення використовуваних алгоритмів, можливе погіршення точності). Параметри, не включені в q , можна розділити на дві групи: а) параметри, вплив які на вимірювання b незначно; б) параметри вплив яких слабо вивчено. З першої групи можна скласти вектор заважаючих параметрів. Відповідна йому помилка ξ_2 визначається аналогічно

$$\xi_2 = B\varphi. \quad (7.42)$$

При визначенні імовірнісних ξ_2 характеристик, можуть виникнути труднощі, які полягають в тому, що вектор не завжди може розглядатися як випадковий. У ряді випадків він має постійне, хоча і невідоме значення. Таким чином, всі заважають параметри можна розділити на випадкові і постійні. Для урахування впливу перших можливе застосування викладеної вище методики для систематичних помилок вимірювання. Для урахування другу імовірнісний підхід не дає позитивного результату. У цьому випадку можна, наприклад, здався максимально можливими значеннями, оцінити максимальне значення помилки ξ_2 .

Поряд з помилками моделі виду (7.42) існують помилки, вплив яких на ξ_2 вектор, що не піддається урахуванню. Ці похибки є приватними

реалізаціями деяких випадкових послідовностей, або не змінюються від вимірювання до вимірювання. У першому випадку виникає завдання визначення імовірнісних характеристик, а в другому - оцінки максимального значення помилок. Крім впливу самих помилок вимірів і помилок моделі на якість одержуваних оцінок в значній мірі впливає вибір алгоритму оцінювання залежно від закону розподілу помилки. Аналогічно (7.32) залежна і незалежна змінна пов'язані співвідношенням

$$y = \alpha x + \xi, \quad (7.43)$$

де α - вектор шуканих параметрів.

Для знаходження параметрів існує велике число алгоритмів, різних за складністю реалізації та статистичних властивостях одержуваних оцінок. Найбільш часто в даній ситуації застосовується метод найменших квадратів (МНК), однією з передумовою якого є припущення щодо нормального розподілу ξ . Однак це припущення в ряді випадків виявляється помилковим, що веде до втрати оцінками своїх оптимальних властивостей. У цій ситуації для різних законів розподілу існують методи, оцінки яких мають кращі в порівнянні з МНК статистичні властивості [261,262]. Розглянемо ці ситуації і відповідні їм алгоритми.

Нехай ξ має щільність розподілу

$$f_k(\epsilon) = \frac{C(\epsilon)}{2\sigma_k \exp\left(-\frac{1}{k} \left| \frac{\epsilon}{\sigma_k} \right|^k\right)} \quad (7.44)$$

У цьому випадку для отримання оцінок, методом максимальної правдоподібності необхідно мінімізувати вираз

$$\sum_{j=1}^n \left| y_j - \sum_{i=0}^m l_i x_{ij} \right|^k \rightarrow \min_{\alpha_i}. \quad (7.45)$$

Нехай $k = 1$. У цьому випадку розподіл ξ зводиться до розподілу Лапласа, а вираз (7.45) приймає вигляд

$$\sum_{j=1}^n P_j \left| y_i - \partial x_j \right| \rightarrow \min_{\alpha_j}, \quad (7.46)$$

де p_j - вага j -го спостереження у разі неравноточних спостережень. Оцінки, отримані з (7.46), називаються оцінками метод найменших модулів (МНМ).

Покладемо $k = 2$. У цьому випадку маємо нормальний ξ розподіл, а вектор α знаходимо з умов мінімуму суми квадратів невязки, тобто приходимо до процедури МНК

$$\sum_{j=1}^n \left(y_i - \alpha x_j \right)^2 \rightarrow \min_{\alpha_i}. \quad (7.47)$$

Значенню $k = \infty$ відповідає рівномірний розподілів Коші. В якості найкраще го методу оцінювання виступає чебишовських наближення, а (7.45) приймає вигляд

$$\hat{\alpha} = \arg \min_{\alpha_i} \max_i \left| y_j - \alpha x_i \right|. \quad (7.48)$$

У роботах [274, 275] наведено порівняльний аналіз алгоритмів (7.46) - (7.48), а також розподілу (7.54) при $k = 4$. В роботі [276] показано, що якщо

апріорна інформація про розподіл помилок відсутній, то найкращою оцінкою буде оцінка МНМ, а якщо відомо, що помилки мають обмежену дисперсію, слід застосовувати МНК.

Традиційно для отримання оцінок лінійних моделей використовується МНК, оскільки його оцінки є незміщеними, ефективними і заможними при наступних припущеннях. Однак на практиці названі припущення повністю не виконуються (тобто не виконуються вихідні передумови теореми Гауса-Маркова [263]), в силу чого оцінки МНК втрачають свою ефективність.

7.3 Адаптивні процедури обробки результатів спостережень за умов аномальних викідів

Оцінки контрольованих параметрів містять помилку, що складається з помилки моделі і помилки вимірювань. Величина помилки моделі залежить від обраного виду моделі і способу обробки вихідної інформації, і її величину можна зменшити лише шляхом ускладнення моделі і, відповідно, ускладнення застосовуваних алгоритмів. Абсолютне значення цієї помилки невідомо, також невідомо, яким чином змінюється вона в міру ускладнення моделі і алгоритмів, тому надалі під помилкою або перешкодою будемо мати на увазі помилку вимірювань. Задачу відшукування істинних значень контрольованих параметрів ускладнює дрейф характеристик об'єкта, тобто існує нестационарність характеристик процесу, що спостерігається. Таким чином, для знаходження значень контрольованих параметрів за результатами спостережень необхідно вирішувати задачу фільтрації або згладжування.

Існує досить широке коло алгоритмів, що дозволяють отримати оцінки в умовах зашумленості вимірів: від алгоритмів, які не вимагають ніякої інформації про зовнішнє середовище і статистичних характеристиках ряду спостережень (алгоритм ковзної середньої), до алгоритмів, що вимагають значної інформації, отримати яку на практиці неможливо.

Найпростішим способом отримання оцінок є згладжування ковзними середніми, при якому поточна оцінка виходить як середнє арифметичне попередніх вимірювань

$$y_i = \frac{1}{m} \sum_{j=i-p}^{i+p} y_j, \quad (7.49)$$

де $m = 2p + 1$ - інтервал згладжування.

У ряді випадків згладжування ряду можна розглядати як допоміжний засіб з подальшим застосуванням більш точних методів. З (7.49) видно, що ваги попередніх вимірювань вибираються однаковими і рівними $\frac{1}{m}$.

Даний метод є вельми грубим прийомом згладжування, оскільки тенденція зміни параметра при цьому виявляється лише в найзагальнішому вигляді і деякі вигини в тренді можуть взагалі зникнути. Більш гнучким прийомом є метод зважених ковзних середніх, при якому кожному виміру в межах інтервалу згладжування приписується своє значення вагового коефіцієнта. В [277] показано, як з умови мінімуму суми квадратів відхилень оцінок параметрів від дійсних значень знаходиться один, два або три значення параметрів моделі

$$y_t = a_0 + a_1 t + a_2 t^2. \quad (7.50)$$

Так, при згладжуванні параболою за $m = 5, 7, 9$ точкам отримані наступні співвідношення:

$$\hat{y}_t = \frac{1}{35} (3y_{t-2} + 12y_{t-1} + 17y_t + 12y_{t+1} - 3y_{t+2}); \quad (7.51)$$

$$\hat{y}_t = \frac{1}{21} (2y_{t-3} + 3y_{t-2} + 6y_{t-1} + 7y_t + 6y_{t+1} + 3y_{t+2} - 2y_{t+3}); \quad (7.52)$$

$$\hat{y}_t = \frac{1}{231} (21y_{t-4} + 14y_{t-3} + 39y_{t-2} + 54y_{t-1} + 59y_t + 54y_{t+1} + 39y_{t+2} + 14y_{t+3} - 21y_{t+4}) \quad (7.53)$$

З формул (7.51) - (7.53) видно, що ваги симетричні щодо центрального вимірювання та сума їх дорівнює знаменника множника, винесеного за дужки. Оскільки вагами є позитивні і негативні числа згладжена крива зберігає вигини кривої тренда. Формулу (7.49) можна представити в іншому вигляді:

$$\hat{y}_i = \hat{y}_{i-1} + \frac{1}{m} (y_i - y_{i-m}). \quad (7.54)$$

У цьому випадку ми приходимо до так званої адаптивної ковзної середньої, в якій перша складова враховує інерцію зміни параметра, а друга - нову інформацію про його фактичну зміну.

На практиці метод ковзної середньої не дає задовільних результатів, оскільки в силу динаміки процесу на кожному кроці є різна залежність між попередніми результатами спостережень. Більш гнучкою є методика, відома як метод експоненціального згладжування, при якому оцінювання параметрів здійснюється за спостереженнями, ваги яких експоненціально убивають у міру віддалення від даного моменту часу t і, крім того, в розрахунки вводиться параметр згладжування, значення якого підбирається з умов мінімізації помилки прогнозу. Ітераційна процедура експоненціального згладжування виглядає наступним чином:

$$S_t = S_{t-1} + \alpha (y_t - S_{t-1}), \quad (7.55)$$

де S_t - оцінка параметра в момент часу t ,

S_{t-1} - оцінка параметра в момент $t - 1$,

x_t - поточне спостереження параметра,

α - постійна згладжування.

Оцінювання параметрів за методикою експоненціального згладжування виконується в залежності від обраного раніше кількості складових моделі (7.50). На практиці використовуються моделі типу

$$y = a + \xi, \quad (7.56)$$

$$y = a_0 + a_1 t + \xi. \quad (7.57)$$

Оскільки в [264] показано, що експоненціальне згладжування дає незміщену оцінку, то оцінку параметра моделі (7.55) можна записати

$$\hat{a}_t = S_t. \quad (7.58)$$

Для визначення коефіцієнтів моделі (7.55) використовується поняття багаторазового експоненціального згладжування порядку

$$S_t^{(n)} = \alpha S_t^{(n-1)} + (1 - \alpha) S_{t-1}^{(n)}. \quad (7.59)$$

Параметри моделі знаходимо з теореми про те, що $n + 1$ невідомих коефіцієнтів полінома n -го порядку, можуть бути оцінені за допомогою лінійних комбінацій $n + 1$ значення $S_t^{(n)}$, $i = \overline{1, n + 1}$. Для $n = 1$, тобто для моделі (7.57) параметри знаходяться з подвійного експоненціального згладжування

$$S_t^{\ominus} = S_{t-1}^{\ominus} + \alpha (y_t^{\ominus} - S_{t-1}^{\ominus}), \quad (7.60)$$

$$S_t^{\ominus} = S_{t-1}^{\ominus} + \alpha (y_t^{\ominus} - S_{t-1}^{\ominus}), \quad (7.61)$$

звідки

$$\hat{a}_0 = 2S_t^{\ominus} - S_t^{\ominus} \quad (7.62)$$

$$\hat{a}_1 = \frac{\alpha}{1-\alpha} (y_t^{\ominus} - S_t^{\ominus}). \quad (7.63)$$

Модель (7.57) з коефіцієнтами (7.62), (7.63) дозволяє більш гнучко реагувати на зміни контрольованого процесу, однак при цьому виникає проблема вибору величини α , оскільки малі значення забезпечують більшу точність оцінки при незмінній моделі, більше вага попередніх вимірювань, а великі значення зменшують час реакції на зміни моделі, зменшуючи вагу попередніх спостережень.

Ефективним методом оцінювання параметрів в умовах наявності завад, а також обмеженого часу обчислень є використання алгоритмів стохастичною апроксимації. Крім того, використання алгоритмів стохастичною апроксимації доцільно для вирішення зазначеної задачі в силу їх непараметричності, пов'язаної з можливістю статистичного оцінювання параметрів при обмеженій апріорній інформації і рекурентності, тобто можливості отримання нової оцінки за новим спостереженням і старій оцінці.

З відомих традиційних методів стохастичною апроксимації розглядалися алгоритми Кіфера-Вольфовица і Сакса [278]. Алгоритм Кіфера-Вольфовица має вигляд

$$X_{t+1} = X_t - \alpha \frac{y_t^{\ominus} - \Delta x_t^{\ominus}}{\Delta x_t^{\ominus}}, \quad (7.64)$$

де α_k - параметр згладжування, який вибирається з умов

$$\lim_{k \rightarrow \infty} \alpha_k = 0 ; \quad (7.65)$$

$$\sum_{i=1}^{\infty} \alpha_i = \infty ; \quad (7.66)$$

$$\sum_{i=1}^{\infty} \alpha_i^2 < \infty . \quad (7.67)$$

Константи α_k і Δx_k рекомендується вибирати наступними:

$$\Delta x_k ; \quad -0.5 \leq p < 0 ; \quad (7.68)$$

$$\alpha_k = \frac{1}{A + BK} , \quad (7.69)$$

причому A, B, F, G підбираються з умови збіжності алгоритму.

При визначенні оцінки градієнта за формулою (7.64) з'являється додаткова похибка, пов'язана з асиметрією у виборі точок вимірювання, що в кінцевому підсумку веде до збільшення часу збіжності алгоритму (7.64). Для виключення даної похибки може бути використано співвідношення

$$X_{k+1} = X_k - \alpha_k \frac{y(x_k + \Delta x_k) - y(x_k - \Delta x_k)}{2\Delta x_k} . \quad (7.70)$$

Алгоритм (7.70) дозволяє збільшити швидкості збіжності в порівнянні з алгоритмом (7.64), однак при цьому збільшується кількість точок вимірювання функції $y(x)$ ($n+1$ і $2n$ вимірювань для алгоритмів (7.64) і (7.70) відповідно).

Внаслідок того, що гіперфункція якості в просторі станів має складну форму, можлива ситуація, що подальша ітерація не дасть кращий результат у

порівнянні з попередньою. Ця обставина значно збільшує час збіжності алгоритмів стохастичною апроксимації.

Для подолання цього недоліку пропонується використовувати у процедурі оцінювання параметрів модифікації алгоритму (7.70), в яких ітерационная процедура враховує як напрям градієнта на k -му кроці, так і напрям градієнта на $k-1$ -му кроці, за аналогією з методом спряжених градієнтів. Метод сполучених градієнтів об'єднує в собі переваги одноступінчатого і двоступінчатого градієнтних методів. Одноступінчатий градієнтний метод простий в обчислювальному відношенні, але дає повільну збіжність поблизу точки екстремуму. Двоступінчастий метод забезпечує швидку збіжність в околицях точки екстремуму, проте у разі великої розмірності вектора виникають серйозні труднощі в обігу $\nabla^2 Q(x)$ матриці, крім того, для забезпечення гарної збіжності важливе значення має вибір початкового наближення.

Алгоритм сполучених градієнтів на початкових ітераціях поводить як одноступінчатий градієнтний метод, однак поблизу точки екстремуму його поведінка стає схожою на поведінку двоступеневого методу. Основна ідея методу сполучених градієнтів ґрунтується на припущенні близькості $Q(x)$ і квадратичної функції і полягає у формуванні послідовності рекуррентно обчислюваних векторів (напрямів ρ_1, \dots, ρ_m спуску), ортогональних до $\nabla^2 Q(x)$ так що

$$\rho_k^T \nabla Q(x) = 0, 0 \leq j \leq k. \quad (7.71)$$

Потім проводиться пошук в сполучених напрямках, що визначаються кожним з векторів ρ_k з метою вибору оптимальної величини кроку. Алгоритм сполучених градієнтів можна представити співвідношенням

$$x_{k+1} = x_k - \alpha_k \rho_k, \quad (7.72)$$

де α_k - скалярний множник, що визначає величину кроку в напрямку \hat{p} з умови

$$Q(\hat{x}) - \alpha_k \hat{p}^T \nabla Q(\hat{x}) = \min_{\alpha \geq 0} Q(\hat{x}) - \alpha \hat{p}^T \nabla Q(\hat{x}), \quad (7.73)$$

де \hat{p} - напрямок пошуку, яке визначається співвідношенням

$$\rho_0 = \nabla Q(\hat{x}), \rho_k = \nabla Q(\hat{x}) - \beta_k \rho_{k-1}. \quad (7.74)$$

Залежно від способу обчислення коефіцієнтів \hat{p} стосовно до задачі оцінювання параметрів технологічних процесів були досліджені наступні конструкції [264]:

а) алгоритм Флетчера-Рівза [265]

$$\beta_k = \frac{\|\nabla Q(\hat{x})\|^2}{\|\nabla Q(\hat{x}) - \nabla Q(\hat{x})\|^2}; \quad (7.75)$$

б) алгоритм Поляка [266]

$$\beta_k = \frac{\|\nabla Q(\hat{x}) - \nabla Q(\hat{x})\|^2}{\|\nabla Q(\hat{x}) - \nabla Q(\hat{x})\|^2}; \quad (7.76)$$

в) алгоритм Мелешко [267]

$$\beta_k = \frac{\|\nabla Q(\hat{x}) - \nabla Q(\hat{x})\|^2}{\|\nabla Q(\hat{x}) - \nabla Q(\hat{x})\|^2}, \quad (7.77)$$

де

$$\rho'_{k-1} = \rho_{k-1} = \frac{\mathbf{q}_{k-1}^T \nabla Q}{\|\nabla Q\|^2}. \quad (7.78)$$

Багато задач обробки експериментальних даних, ідентифікації та прогнозування пов'язані з побудовою множинної регресії виду

$$\vec{Y} = X\vec{a} + \vec{\varepsilon},$$

де $\vec{Y} \in \mathbb{R}^n$ - вектор вихідної змінної;

$X \in \mathbb{R}^{n \times m}$ - матриця спостережень;

$\vec{a} \in \mathbb{R}^m$ - вектор оцінюваних параметрів;

$\varepsilon \in \mathbb{R}^n$ - завада вимірювань.

Найбільш часто для оцінювання параметрів використовуються алгоритми методу найменших квадратів (МНК). Якщо перешкоди вимірювань розподілені по нормальному закону, МНК - оцінки стають оцінками максимальної правдоподібності, а також незміщеними, заможними, ефективними і достатніми. Наявність коррелированности вхідних змінних, що приводить до поганої обумовленості інформаційної матриці, різко погіршує властивості МНК - оцінок.

Зменшення середньоквадратичної помилки (СКО) оцінювання може бути досягнуто переходом від МНК до двоступінчастим МНК. Двоступінчастий метод, заснований на МНК, був запропонований в [269] для аналізу залишків лінійного регресійного рівняння. Задача оцінювання параметрів за допомогою такого методу розглянута в [270, 271]. Крім того, двоступінчастий МНК (ДМНК), маючи ті ж зміщення, що і МНК, дозволяє отримати меншу в порівнянні з МНК середньоквадратичну помилку оцінювання. Ефективність ДМНК істотно зростає при наявності перешкод і

при сильній корелюваності вхідних змінних, тобто при поганій обумовленості матриці спостережень.

У ряді випадків, зокрема, при наявності кореляції корисного сигналу і завади кращим виявляється метод інструментальних змінних (МІЗ). Однак, відомий МІЗ має ряд недоліків, тому є об'єктом отримання такої модифікації цього методу, яка крім незміщеної оцінки давала б ще й оцінку з меншою середньоквадратичною помилкою в порівнянні з МП.

Для отримання модифікованого МІЗ вчинимо так. Розіб'ємо матрицю X і вектор \vec{a} на блоки

$$X = (X_1, X_2), \vec{a}^T = (\vec{a}_1^T, \vec{a}_2^T). \quad (7.79)$$

Тут $X_1 \in R^{n \times p}$, $X_2 \in R^{n \times q}$, $\vec{a}_1 \in R^p$, $\vec{a}_2 \in R^q$, $p + q = m$.

Рівняння регресії можна записати у вигляді

$$\vec{Y} = x_1 \vec{a}_1 + x_2 \vec{a}_2 + \vec{\varepsilon}. \quad (7.80)$$

Позначимо $x_2 \vec{a}_2 + \vec{\varepsilon}_t = \vec{\chi}_t$ і оцінимо з отриманого рівняння \vec{a}_1 . У цьому випадку оцінка МІЗ буде мати вигляд

$$\vec{\alpha}_1 = (W_1^T X_1)^{-1} W_1^T Y. \quad (7.81)$$

Тут матриця W розбита також на блоки $W = (W_1, W_2)$, $W_1 \in R^{n \times p}$, $W_2 \in R^{n \times q}$.

Позначимо $\vec{Y} = \vec{Y} - x_1 \vec{\alpha}_1$. Тоді оцінка МІЗ вектора \vec{a} запишеться наступним чином

$$\alpha_2 = (W_2^T X_2)^{-1} W_2^T \bar{Y}$$

або з урахуванням виразів для $\bar{\alpha}_1$ і

$$\bar{Y} \alpha_2 = (W_2^T X_2)^{-1} W_2^T (\bar{Y} - X_1 \bar{\alpha}_1) = (W_2^T X_2)^{-1} W_2^T P \bar{Y} \quad (7.82)$$

де $P = I - X_1 (W_1^T X_1)^{-1} W_1^T I$ - одинична матриця розмірності $n \times n$.

Таким чином, оцінки ДМІЗ мають вигляд

$$\bar{\alpha}_1 = (W_1^T X_1)^{-1} W_1^T \bar{Y}; \quad (7.83)$$

$$\bar{\alpha}_2 = (W_2^T X_2)^{-1} W_2^T P \bar{Y}. \quad (7.84)$$

Порівняємо СКО і ДМІЗ і МІЗ

$$\begin{aligned} & M\{\|\hat{\epsilon}_1 - \bar{a}_1\|^2\} - M\{\|\bar{\alpha}_1 - \bar{a}_1\|^2\} = M\{\|\hat{\epsilon}_1 - M\{\hat{\epsilon}_1\}\|^2\} = \\ & = \sigma^2 \text{tr} D_{11} - \sigma^2 \text{tr} (W_1^T X_1)^{-1} W_1^T W_1 (X_1^T W_1)^{-1} - \|A_1 \bar{a}_1\|^2; \\ & M\{\|\hat{\epsilon}_1 - \bar{a}_1\|^2\} - M\{\|\bar{\alpha}_1 - \bar{a}_1\|^2\} = M\{\|\hat{\epsilon}_1 - M\{\hat{\epsilon}_1\}\|^2\} = \\ & = \sigma^2 \text{tr} D_{11} - \sigma^2 \text{tr} (W_1^T X_1)^{-1} W_1^T W_1 (X_1^T W_1)^{-1} - \|A_1 \bar{a}_1\|^2; \end{aligned}$$

Підставляючи в дані формули вирази для D_{11} , D_{22} , A_1 , A_2 і P , після нескладних перетворень отримаємо, що оцінки ДМІЗ мають меншу СКП в порівнянні з МІЗ при виконанні умов

$$\text{tr} \tilde{W}_1^T [\sigma^2 (X_1 N^{-1} W_2^T P - I) P^T W_2 \tilde{N}^{-1} X_2^T - X_2 (\sigma^2 N^{-1} W_2^T P + \bar{a}_2 \bar{a}_2^T X_2^T)] \tilde{W}_1 > 0 \quad (7.85)$$

$$\text{tr}[\sigma^2 N^{-1} W_2^T P(I - P^T) W_2 \tilde{N}^{-1} - \sigma^2 \tilde{W}_2^T P \tilde{W}_2 - A_2 A_1 \tilde{\alpha}_2 \tilde{\alpha}_2^T A_2^T A_2] > 0. \quad (7.86)$$

Тут

$$\tilde{W}_1^T = (W_1^T X_1)^{-1} W_1^T$$

$$\tilde{W}_2^T = (W_2^T X_2)^{-1} W_2^T.$$

При побудові ДМНК слід прийняти $W = X$, тобто в якості матриці I_3 використовується сама матриця спостережень X . В цьому випадку $W^T X = X^T W = X^T X$, $(W^T X)^{-1} = (X^T W)^{-1} = (X^T X)^{-1}$, і отримані формули істотно спрощуються. Так

$$D\{\tilde{\alpha}_1\} = \sigma^2 (X_1^T X_1)^{-1}; \quad D\{\tilde{\alpha}_2\} = \sigma^2 (I - A_2 A_2)(X_2^T X_2)^{-1};$$

$$D_{\text{МНК}}\{\alpha\alpha = \sigma^2 (X^T X)^{-1}.$$

Крім того, спрощуються елементи ковариационної $D\{\alpha\}$ матриці, оскільки $N^{-1} = N^{-1}$, $P = P^T$, $PP^T = P$.

За результатами розрахунків можна зробити наступні висновки. Помилка оцінювання всіма алгоритмами зростає в міру збільшення рівня перешкод у вихідній вибірці. Помилки оцінок, що отримано алгоритмами змінного середнього і експоненційного згладжування, практично не залежать від виду розподілу помилок вимірювань. Для алгоритму стохастичною апроксимації похибка оцінювання при одному і тому ж рівні перешкод кілька нижче для нормально розподілених помилок текстової послідовності. З перерахованих алгоритмів більш точні результати були отримані алгоритмом (7.70) з модифікацією (7.77). Трохи гірші результати дає процедура експоненціального згладжування (7.60) - (7.63). Поліноміальні форми (7.47) - (7.52) не дали відчутного виграшу в порівнянні з методом ковзної середньої

чинності динаміки досліджуваної послідовності, особливо при збільшенні рівня завади.

Традиційно для знаходження невідомих параметрів регресійних моделей за наявності достатнього числа спостережень використовується МНК

$$\hat{\Theta} = (X^T X)^{-1} X^T Y. \quad (7.87)$$

Особливістю використання МНК є та обставина, що для оцінювання параметрів не потрібно робити ніяких попередніх обчислень. Проте використання МНК може бути ефективним лише при дотриманні умов, описаних вище.

Співвідношення (7.87) дозволяє здійснити оцінювання параметрів безпосередньо за результатами вимірювань. Однак при цьому виникає ряд незручностей, пов'язаних з необхідністю набору великої кількості вимірювань. Крім того, МНК не враховує фактор стирання інформації. Для отримання оцінок параметрів за результатами оцінювання на попередньому кроці з урахуванням поточного вимірювання використовується рекурентний МНК [268]. У цьому випадку вираз для оцінювання параметрів має вигляд

$$\hat{Q}_{n+1} = \hat{Q}_n + R_n h_{n+1}^T (R_{n+1} h_{n+1}^T + \sigma^2)^{-1} (y_{n+1} - h_{n+1}^T \hat{\Theta}_n), \quad (7.88)$$

де

$$R_{n+1} = R_n - R_n h_{n+1}^T (R_{n+1} h_{n+1}^T + \sigma^2)^{-1} R_n. \quad (7.89)$$

З (7.88), (7.89) видно, що для отримання поточної оцінки необхідно знати значення попередньої \hat{Q}_n оцінки, поточне y_{n+1} вимірювання, а також значення елементів матриці R , причому ці елементи можуть бути розраховані заздалегідь для будь-якої розмірності матриці R .

Методика оцінювання параметрів за допомогою МНК придатна для знаходження параметрів статичних процесів, в той час як більшість контрольованих процесів транспорту газу є динамічними. У даній ситуації більш кращим є застосування процедури Калмана (ФК). Відмінною особливістю даної процедури від рекурентного МНК є обчислення коефіцієнта посилення фільтра (матриці K) на кожному кроці. Алгоритм знаходження невідомих параметрів при цьому описується рівняннями

$$\hat{X}_{k+1} = F \hat{X}_k - K_{k+1} (Z_{k+1} - H^T F \hat{X}_k); \quad (7.90)$$

$$K_{k+1} = \gamma_{k+1} H / \sigma_0^2; \quad (7.91)$$

$$V_{k+1} = Q_k \gamma_k Q_k + \Gamma \Gamma^T; \quad (7.92)$$

$$\gamma_{k+1} = V_{k+1} - \frac{V_{k+1} H H^T V_{k+1}}{H^T V_k H + \sigma_0^2}. \quad (7.93)$$

Основною перевагою фільтра Калмана перед МНК є висока точність одержуваних оцінок в умовах нелінійності і нестационарності. До недоліків можна віднести слабку збіжність обчислювальної процедури, труднощі у виборі кореляційної матриці помилок оцінок.

У випадку, якщо перешкода має розподіл Лапласа, кращі результати можуть бути отримані за допомогою МНМ, сутність якого полягає в мінімізації виразу

$$H \Theta \approx \sum_{i=1}^n P_i |x_i - \bar{y}_i \Theta_1, \dots, \Theta_n|, \quad (7.94)$$

де P_i - вагова матриця i -го спостереження;

X - вектор вимірювань;

\bar{y}_i - відома функція.

Перевагою МНМ перед МНК є можливість оцінювання параметрів при корельованих вимірах. У першому випадку мінімізується вираз

$$H(\theta) = \sum_{i=1}^n P_i |e_i - \varphi(\theta)|^2, \quad (7.95)$$

а в другому -

$$\Phi(\theta) = \sum_{i=1}^n P_i \left| \Delta x_i - \sum_{j=1}^m b_{ij} z_j \right|, \quad (7.96)$$

де

$$z_j = \theta_j - \theta_j^0, \Delta x_i = x_i - \varphi_i(\theta); b_{ij} = \left. \frac{\partial \varphi_i}{\partial \theta_j} \right|_{\theta = \theta^0}. \quad (7.97)$$

Для знаходження невідомих параметрів використовується симплексний метод лінійного програмування або метод варіаційно-зважених квадратичних наближень. Перший метод особливо ефективний для лінійних моделей і застосовується у випадках, коли неприпустимий вихід параметра за встановлені межі. Другий метод більш кращий для нелінійного вигляду функції y_i , припускає значне збільшення часу перебування оцінки в порівнянні з МНК. Точність знаходження невідомих параметрів оцінювалася величиною залишкової дисперсії. Аналіз отриманих кривих дозволяє зробити висновок, що алгоритм ФК є найменш чутливим до зашумленості вихідних даних, у той час як дисперсія помилки по МНК зі збільшенням рівня завади значно зростає. Алгоритм ковзної середньої є найгіршим з досліджуваних алгоритмів з точки зору точності оцінювання. Алгоритм стохастичною апроксимації успішно поєднує в собі швидкодію і фільтруючі властивості.

Таким чином, при знаходженні невідомих параметрів в умовах аномальних викидів результатів спостережень прийнятні результати можуть бути отримані за допомогою сукупності алгоритмів, причому, якщо оцінювання ведеться в реальному масштабі часу, переважним є використання методу стохастичною апроксимації, а в ситуаціях, коли необхідна висока точність, кращі результати дає алгоритм Калмана.

При вирішенні завдань контролю результати спостережень надходять для обробки у вигляді часового ряду. Обробка інформації може бути здійснена різними алгоритмами, однак найбільш широко для цієї мети використовується МНК, в основі якого закладено низку важливих обмежуючих умов, одним з яких є некоррелірованні результатів спостережень між собою, помилок спостережень між собою, помилок спостережень і результатів спостережень. Ця передумова, як правило, порушується, і при використанні МНК веде до додаткової похибки.

При дотриманні основних гіпотез МНК точність одержуваних оцінок прийнято характеризувати середньоквадратичної помилкою (СКП), яка визначається співвідношенням

$$\sigma_{\bar{a}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\epsilon_i - a)^2}, \quad (7.98)$$

з якого випливає, що при збільшенні числа вимірів точність повинна необмежено зростати. Однак на практиці це положення не виконується, оскільки порушується одна з основних передумов МНК - відсутність кореляції між результатами спостережень. Наслідком цього є той факт, що зі збільшенням числа вимірювань похибка флюктує поблизу встановленого значення й у ряді випадків навіть зростає. Тому має сенс говорити про вибір оптимального числа експериментів, яке забезпечить достатню показність і необхідну точність при певному впливі зовнішнього середовища. Розглянемо це положення докладніше.

Уявімо помилку оцінки ξ_i , $i = \overline{1, n}$ у вигляді суми складових

$$\xi_i = \xi'_i + \xi''_i, \quad (7.99)$$

де ξ'_i - помилки вимірювань, що є випадковою складовою помилок;

ξ_i , ξ''_i - помилки, викликані неадекватністю моделі, що представляють систематичну складову помилок ξ_i .

Оскільки складова помилок ξ''_i є детермінованою, то основна увага надалі приділимо дослідженню випадкової складової ξ'_i , відносно якої зробимо припущення

$$M \xi'_i = 0, D \xi'_i = \sigma^2, i = \overline{1, n}, \quad (7.100)$$

а корельованість динамічного ряду задана коефіцієнтом кореляції

$$K \xi'_i, \xi'_j = \begin{cases} = 0, & \text{при } i = j \\ \neq 0, & \text{при } i \neq j \end{cases}. \quad (7.101)$$

Розглянемо вплив корельованості результатів спостережень на похибки одержуваних оцінок. Нехай є n рівноточних вимірювань зі середньоквадратичної помилкою σ . Припустимо, що між результатами спостережень має місце позитивна кореляційна залежність з коефіцієнтом кореляції, рівним

$$K \xi'_i, \xi'_j = K, i \neq j, 0 < K < 1. \quad (7.102)$$

Для визначення середньоквадратичного відхилення σ помилки справедливе співвідношення

$$\sigma_{\hat{\beta}} = \frac{1}{n} \sqrt{\sum_{i=1}^n \sum_{j=1}^n \sigma_{ij}^2}, \quad (7.103)$$

де σ_{ij}^2 - коефіцієнти ковариационної матриці помилок вимірів, які визначаються співвідношенням

$$\sigma_{ij}^2 = \begin{cases} \sigma^2 & \text{при } i = j \\ K\sigma^2 & \text{при } i \neq j \end{cases}. \quad (7.104)$$

Оскільки діагональних елементів ковариационної матриці помилок n , а недіагональних $n(k-1)$, то співвідношення (7.103) з урахуванням (7.104) набуде вигляду

$$\sigma_{\hat{\beta}} = \frac{1}{n} \sqrt{n\sigma^2 + n(k-1)K\sigma^2} = \sigma \sqrt{\frac{1-k}{n} + k}. \quad (7.105)$$

Порівнюючи співвідношення (7.105) і (7.103), приходимо до висновку про неспроможність оцінок МНК при корельованості результатів спостережень, оскільки

$$\sigma_{\hat{\beta}} \rightarrow \sigma \sqrt{k} \text{ при } n \rightarrow \infty. \quad (7.106)$$

Скориставшись співвідношенням (7.105), досліджуємо величину відносної помилки оцінки як функцію коефіцієнта кореляції при різному числі результатів спостережень, тобто

$$\frac{\sigma_{\hat{\beta}}}{\sigma_0} = f(k) \text{ при } n = \text{const} \quad (7.107)$$

За результатами досліджень можна зробити наступні висновки:

1) При фіксованому числі результатів спостережень зі збільшенням з від 0 до 1 спостерігається збільшення відносної помилки оцінки за законом

$$\frac{\sigma_{\hat{\theta}}}{\sigma_0} \approx \sqrt{kn} \text{ при } n > 20.$$

2) При наявності навіть малої кореляційної залежності (порядку 0,1) між результатами спостережень зі збільшенням довжини динамічного ряду відносна похибка може скільки завгодно зростати. Наприклад, при $n = 1000$, $\sigma_{\hat{\theta}}/\sigma_0 = 10$, що спростовує співвідношення (7.107). Таким чином, неврахування кореляції між результатами спостережень призводить до різкого зростання похибки оцінки контрольованих параметрів і порушенню заможності статистичних оцінок.

За наявності корельованості між результатами спостережень підвищити точність оцінок тільки за рахунок збільшення числа вимірів не уявляється можливим. Для цього необхідне залучення нових членів динамічного ряду, що містять додаткову інформацію щодо параметрів або станів процесу.

У даній ситуації для визначення оцінок параметрів моделі використовуються видозмінені методи оцінювання. Стосовно до оцінювання МНК розроблений ряд методів оцінювання на основі так званого узагальненого МНК. Основна ідея його полягає у включенні в процедуру оцінювання ковариаційної матриці $R = M \left[\xi, \xi' \right] = \sigma^2 \Omega$. Вираз для оцінки параметрів узагальненим МНК виглядає наступним чином:

$$a = \left(X' R^{-1} X \right)^{-1} X' R^{-1} Y. \quad (7.108)$$

Основна трудність застосування (7.108) полягає в тому, що ми не володіємо інформацією ні про порядок кореляції, ні про коефіцієнти у відповідній авторегресійній схемі. Тому при вирішенні практичних завдань оцінювання обмежуються тій чи іншою умовною формою її подання. У найпростішому випадку $R = \sigma^2 I$. Однак таке уявлення хоч і дасть вигреш в порівнянні зі звичайним МНК, все ж дозволяє говорити про ефективність оцінок

Якщо завада задовольняє схемою авторегресії першого порядку $\xi_t = \rho \xi_{t-1} + \varepsilon_t$, то для випадку гомоскедастичність можна припустити

$$R = \sigma^2 \begin{bmatrix} 1 & \rho & \rho^2 & \rho^{m-1} \\ \rho & 1 & \rho & \rho^{m-2} \\ \dots & \dots & \dots & \dots \\ \rho^{m-1} & \dots & \dots & 1 \end{bmatrix}. \quad (7.109)$$

Однак матриця R з виразу (7.109) хоч і містить всі елементи, відмінні від нуля, може внести додаткову погрешність в оцінки, оскільки коефіцієнт парної кореляції між сусідніми вимірами по всій довжині вибірки не є величиною постійною, причому ця похибка може бути вище, ніж відповідна похибка при більш простих формах матриці R . У силу цього при дослідженнях обмежимося двома формами подання R - діагональною і стрічковою, що містить головну і дві симетричні їй діагоналі, що відповідає кореляції між сусідніми вимірами.

Іншим підходом, що дозволяє отримати ефективні оцінки при корельованих вимірах, є попереднє усунення кореляції шляхом перетворення вихідного ряду з подальшим використанням традиційних методів. В даний час відомий ряд методів виключення кореляції. Перший спосіб, відомий як метод Фріша-Боу, припускає введення в рівняння регресії часу в якості додаткового фактора. При цьому лінійна модель має вигляд

$$y = a_0 + a_1 x + a_2 t \quad (7.110)$$

У разі використання МНК для отримання параметрів моделі мінімізується вираз

$$L = \sum_{i=1}^n (y_i - f(x_i, t_i))^2 \rightarrow \min_a \quad (7.111)$$

Параметри моделі знаходяться з системи нормальних рівнянь

$$\begin{aligned} \sum_i y_i &= a_0 n + a_1 \sum x_i + a_2 \sum t_i; \\ \sum x_i y_i &= a_0 \sum x_i + a_1 \sum x_i^2 + a_2 \sum t_i x_i; \\ \sum t_i y_i &= a_0 \sum t_i + a_1 \sum t_i x_i + a_2 \sum t_i^2. \end{aligned} \quad (7.112)$$

Другим методом усунення кореляції є метод кінцевих різниць, при якому в якості вихідного для обробки ряду виступає ряд, сформований з послідовних різниць порядку. Для лінійних моделей обмежуються різницями першого порядку

$$\begin{aligned} \Delta y_t &= y_t - y_{t-1}; \\ \Delta y_{t-1} &= y_{t-1} - y_{t-2}; \\ &\dots\dots\dots \\ \Delta y_{t-k} &= y_{t-k} - y_{t-k-1}. \end{aligned} \quad (7.113)$$

При цьому виходять із припущення, що всі різниці будуть містити таку випадкову компоненту. Надалі всі обчислення проводяться з різницями з наступним перерахунком до рівнів динамічного ряду. Для оцінювання в умовах коррелированности результатів спостережень відомий рекурентний

алгоритм Хенріксону, в якому фільтруються не власними вимірювання, а зважена різниця послідовних вимірювань

$$\delta_k = Z_{k+1} \Psi_k Z_k^{-1} \quad (7.114)$$

яка може бути представлена у вигляді

$$\delta_k = H_k^* x_k + \omega_k, \quad (7.115)$$

$$H_k^* = H_k \Phi - \Psi_k H_k, \quad (7.116)$$

де x_k - вектор стану.

При цьому алгоритм ФК набуває вигляду

$$\hat{x}_{k+1} = \Phi x_k + K_{k+1} \left[\delta_{k+1} - H_{k+1}^* \Phi \hat{Y}_k \right], \quad (7.117)$$

$$P_{k+1} = P_{k+1}' - K_{k+1} H_{k+1}^* P_{k+1}', \quad (7.118)$$

$$K_{k+1} = \left(P_{k+1}' H_{k+1}^{*T} H_{k+1}^* + \sigma_{\omega k}^2 \right)^{-1}, \quad (7.119)$$

$$P_{k+1} = \Phi P_k \Phi^T, \quad (7.120)$$

де x_k позначає оцінку вектора стану за k вимірами, а P_k - розрахункову ковариаційну матрицю помилок оцінок.

Перевагою запропонованого методу є відсутність погано обумовлених матриць в процесі знаходження оцінки параметрів.

Найбільш часто задача оцінювання параметрів лінійної регресійної моделі зводиться до наступної процедури

$$Y_n = X_n \cdot C_n^* + \Xi_n, \quad (7.121)$$

де $Y_n = (y_1, y_2, \dots, y_n)^T$ - вектор вихідних сигналів $n \times 1$;

$X_n = (x_{11}, x_{12}, \dots, x_{1n})$ - матриця вхідних змінних $N \times n$;

$C_n^* = (c_1, c_2, \dots, c_n)^T$ - вектор оцінюваних параметрів $N \times 1$;

$\Xi_n = (\xi_1, \xi_2, \dots, \xi_n)^T$ - вектор завад $N \times 1$.

При виконанні звичайних уявлень класичного регресійного аналізу успішно вирішується за допомогою методу найменших квадратів (МНК). Одним з істотних припущень є те, що Ξ_n - випадковий вектор з $M\{\Xi_n\} = 0$ і $\text{cov}(\Xi_n) = \sigma_\xi^2 \cdot I_n$. Тут σ_ξ^2 - дисперсія завади, I_n - одинична матриця $n \times n$.

Якщо ж завади корельовані, тобто $\text{cov}(\Xi_n) = K$, де K - відома невироджена кореляційна матриця завади, то найменшою дисперсією і матрицею коваріації володіє оцінка узагальненого або зваженого МНК

$$c_n = (X_n^T \cdot K_n^{-1} \cdot X_n)^{-1} \cdot X_n^T \cdot K_n^{-1} \cdot Y_n. \quad (7.122)$$

Оцінка (7.122) є спроможною, якщо вхідні змінні x_i вимірюються без перешкод. Якщо ж змінні x_i вимірюються з завадами σ_{i_i} і існує кореляція, оцінки ОМНК будуть зміщені. У цьому випадку доцільно застосування методів, заснованих на використанні інструментальних змінних, удовлетворяющий певним вимогам. Оцінка методу інструментальних змінних (МІЗ) має вигляд

$$c_n = (W_n^T \cdot X_n)^{-1} \cdot W_n^T \cdot Y_n, \quad (7.123)$$

де W_n - матриця інструментальних змінних $N \times n$.

За аналогією з оцінкою (7.122) розглянемо наступну модифікацію МІЗ:

$$c_n = (W_n^T \cdot K_n^{-1} \cdot X_n)^{-1} \cdot W_n^T \cdot K_n^{-1} \cdot Y_n, \quad (7.124)$$

враховує кореляцію завад $\Xi_i (i = 1, 2, \dots, n)$.

Нехай помилки виміру ξ_i мають нульове математичне очікування і кореляційну функцію

$$K = \sigma_\xi^2 \cdot e^{-\alpha t}, \quad (7.125)$$

де α - постійний параметр.

Крім того, прийmemo звичайне в задачах ідентифікації припущення, що вимірювання проводяться через рівні проміжки часу $\Delta t = t_{i+1} - t_i (i = 0, 1, \dots, n-1)$.

У цьому випадку матриця K^{-1} з елементами, обумовленими виразом (7.125), являє собою трехдіагональну матрицю

$$K^{-1} = \frac{1}{\sigma_\xi^2(1-d^2)} \begin{vmatrix} 1 & -d & 0 & 0 & \dots & 0 & 0 & 0 \\ -d & 1+d^2 & -d & 0 & \dots & 0 & 0 & 0 \\ 0 & -d & 1+d^2 & -d & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & -d & 1+d^2 & -d \\ 0 & 0 & 0 & 0 & \dots & 0 & -d & 1 \end{vmatrix},$$

де $d = e^{-\alpha \Delta t}$.

Таким чином, елементи матриці K^{-1} обчислюються за формулами

$$K_{i,i}^{-1} = \begin{cases} \frac{1}{\sigma_{\xi}^2(1-d^2)} & \text{при } i=1; i=n \\ \frac{1+d^2}{\sigma_{\xi}^2(1-d^2)} & \text{при } i=2,3,\dots,n \end{cases};$$

$$K_{i,i+1}^{-1} = -\frac{d}{\sigma_{\xi}^2(1-d^2)} \quad \text{при } i=1,2,\dots,n-1;$$

$$K_{i,i-1}^{-1} = -\frac{d}{\sigma_{\xi}^2(1-d^2)} \quad \text{при } i=2,3,\dots,n. \quad (7.126)$$

Для отримання рекуррентной форми цікавлячох нас оцінки перепишемо (7.124) наступним чином:

$$W_n^T \cdot K_n^{-1} \cdot X_n \cdot c_n = W_n^T \cdot K_n^{-1} \cdot Y_n, \quad (7.127)$$

При надходженні інформації щодо $(n+1)$ -й вимір вхідних і вихідних сигналів рівняння (7.124) набуде вигляду

$$W_{n+1}^T \cdot K_{n+1}^{-1} \cdot X_{n+1} \cdot c_{n+1} = W_{n+1}^T \cdot K_{n+1}^{-1} \cdot Y_{n+1}, \quad (7.128)$$

Розглянемо праві і ліві частини рівнянь (7.127), (7.128). Можна записати для n -вимірів

$$W_n^T \cdot K_n^{-1} \cdot Y_n = \frac{1}{\sigma_{\xi}^2(1-d^2)} \left[w_1 \cdot y_1 + w_n \cdot y_n + (1+d^2) \cdot \sum_{i=1}^{n-1} w_i \cdot y_i - d \sum_{i=1}^{n-1} w_i \cdot y_{i+1} - d \sum_{i=2}^n w_i \cdot y_{i-1} \right]; \quad (7.129)$$

Аналогічно для $(n + 1)$ - вимірювання маємо

$$W_{n+1}^T \cdot K_{n+1}^{-1} \cdot Y_{n+1} = \frac{1}{\sigma_{\xi}^2(1-d^2)} \left[w_{n+1} \cdot y_{n+1} + (1+d^2) \cdot \sum_{i=1}^n w_i \cdot y_i - d \sum_{i=1}^n w_i \cdot y_{i+1} - d \sum_{i=2}^{n+1} w_i \cdot y_{i-1} \right];$$

$$W_{n+1}^T \cdot K_{n+1}^{-1} \cdot X_{n+1} = \frac{1}{\sigma_{\xi}^2(1-d^2)} \left[w_{n+1} \cdot x_{n+1} + (1+d^2) \cdot \sum_{i=1}^n w_i \cdot x_i - d \sum_{i=1}^n w_i \cdot x_{i+1} - d \sum_{i=2}^{n+1} w_i \cdot x_{i-1} \right]. \quad (7.130)$$

Додамо до правої і лівої частини співвідношення (7.129) вектор

$$\frac{1}{\sigma_{\xi}^2(1-d^2)} \left[w_{n+1} \cdot x_{n+1} + d^2 \cdot w_n \cdot x_n - d \cdot w_n \cdot x_{n+1} - d \cdot w_{n+1} \cdot x_n \right] c_n.$$

Для спрощення запису введемо наступні позначення:

$$\frac{1}{\sigma_{\xi}^2(1-d^2)} \left[d^2 \cdot w_n \cdot x_n - d \cdot w_{n+1} \cdot x_n \right] = \frac{1}{\sigma_{\xi}^2(1-d^2)} \left[\mathbf{V} \cdot w_n - w_{n+1} \right] x_n = q \cdot x_n; \quad (7.131)$$

$$\frac{1}{\sigma_{\xi}^2(1-d^2)} \left[w_{n+1} - d \cdot w_n \right] x_{n+1} = r \cdot x_{n+1}. \quad (7.132)$$

Це дозволяє записати перетворене співвідношення (7.126) у вигляді

$$\mathbf{V} \cdot K_n^{-1} \cdot X_n + q \cdot x_n + r \cdot x_{n+1} \cdot c_n = W_n^T \cdot K_n^{-1} \cdot Y_n + (q \cdot x_n + r \cdot x_{n+1}) \cdot c_n. \quad (7.133)$$

Відніmemo з правої і лівої частини рівняння (7.128) відповідні частини рівняння (7.133)

$$\begin{aligned} W_{n+1}^T \cdot K_{n+1}^{-1} \cdot X_{n+1} \cdot (c_{n+1} - c_n) = \\ = W_{n+1}^T \cdot K_{n+1}^{-1} \cdot Y_{n+1} - W_n^T \cdot K_n^{-1} \cdot Y_n - (q \cdot x_n^T + r \cdot x_{n+1}^T) \cdot c_n. \end{aligned} \quad (7.134)$$

Але з урахуванням формул (7.129), (7.130) і введених позначень (7.131), (7.132)

$$\begin{aligned} W_{n+1}^T \cdot K_{n+1}^{-1} \cdot Y_{n+1} - W_n^T \cdot K_n^{-1} \cdot Y_n = \frac{1}{\sigma_{\xi}^2(1-d^2)} (w_{n+1} \cdot y_{n+1} + d^2 \cdot w_n \cdot y_n - d \cdot w_n \cdot y_{n+1} - d \cdot w_{n+1} \cdot y_n) \\ = r \cdot y_{n+1} + q \cdot y_n. \end{aligned}$$

Тоді з рівняння (7.134) випливає, що

$$c_{n+1} = c_n + (W_{n+1}^T \cdot K_{n+1}^{-1} \cdot X_{n+1})^{-1} \cdot \begin{bmatrix} Y_{n+1} + q \cdot Y_n - (q \cdot x_n^T + r \cdot x_{n+1}^T) c_n \end{bmatrix}$$

або

$$\begin{aligned} c_{n+1} = c_n + (W_{n+1}^T \cdot K_{n+1}^{-1} \cdot X_{n+1})^{-1} \cdot \\ \times \begin{bmatrix} (Y_{n+1} - c_n \cdot x_{n+1}^T) + q \cdot (Y_n - c_n \cdot x_n^T) \end{bmatrix}. \end{aligned} \quad (7.135)$$

Зворотню матрицю, що входить в даний вираз, також можна обчислювати рекуррентно. Позначимо $P_{n+1} = W_{n+1}^T \cdot K_{n+1}^{-1} \cdot X_{n+1}$. З (7.130) випливає, що

$$P_{n+1} = P_n + q \cdot x_n^T + r \cdot x_{n+1}^T. \quad (7.136)$$

Введемо ще одне позначення

$$\hat{P}_n^{-1} = P_n^{-1} + q \cdot x_n^T. \quad (7.137)$$

Тоді

$$P_{n+1}^{-1} = \hat{P}_n^{-1} + r \cdot x_{n+1}^T. \quad (7.138)$$

Застосовуючи до (7.137) і (7.138) лему про звернення матриць, отримуємо наступні рекурентні співвідношення

$$\hat{P}_n = P_n - \frac{P_n \cdot q \cdot x_n^T \cdot P_n}{1 + x_n^T \cdot P_n \cdot q}, \quad (7.139)$$

$$P_{n+1} = \hat{P}_n - \frac{\hat{P}_n \cdot r \cdot x_{n+1}^T \cdot \hat{P}_n}{1 + x_{n+1}^T \cdot \hat{P}_n \cdot r}, \quad (7.140)$$

Перепишемо оцінку (7.137) у вигляді

$$c_{n+1} = c_n + P_{n+1} \left[(y_{n+1} - c_n \cdot x_{n+1}^T) + q \cdot (y_n - c_n \cdot x_n^T) \right]. \quad (7.141)$$

Таким чином, співвідношення (7.139) - (7.141) є алгоритмом оцінки вектора c^* за наявності корельованих завад. Цей алгоритм відрізняється від відомих модифікацій МІЗ тим, що в ньому замість векторів інструментальних змінних w_n використовуються перетворені вектори r і q .

Якщо в (7.136) ввести позначення

$$\tilde{P}_n^{-1} = P_n^{-1} + r \cdot x_{n+1};$$

тобто змінити порядок обчислення елементів матриці P_{n+1} , то, як неважко в цьому переконатися, в алгоритмі (7.139) - (7.141) зміняться лише формули (7.139), (7.140). В цьому випадку

$$\tilde{P}_{n+1} = P_n - \frac{P_n \cdot r \cdot x_{n+1}^T \cdot P_n}{1 + x_n^T \cdot P_n \cdot r}; \quad (7.142)$$

$$P_{n+1} = \tilde{P}_n - \frac{\tilde{P}_{n+1} \cdot q \cdot x_n^T \cdot \tilde{P}_{n+1}}{1 + x_n^T \cdot \tilde{P}_{n+1} \cdot q}; \quad (7.143)$$

Ефективність вирішення завдань контролю в значній мірі залежить від якості використовуваних математичних моделей досліджуваних об'єктів. Зазвичай реальні об'єкти описуються нелінійними залежностями і характеризуються значним рівнем завад і їх корельованості, відсутністю апріорної інформації про параметри об'єкта. Всі ці фактори суттєво ускладнюють вибір адекватної математичної моделі. Тому найчастіше виправдані спрощення, що застосовуються при побудові та дослідженні моделей. Зокрема, досить широко використовується модель досліджуваного об'єкта у вигляді

$$\vec{Y} = X\vec{a} + \vec{\varepsilon}, \quad (7.144)$$

де $\vec{Y} \in R^n$ - вектор вихідної змінної;

$X \in R^{n \times m}$ - матриця спостережень;

$\vec{a} \in R^n$ - вектор оцінюваних параметрів;

$\varepsilon \in R^n$ - завада вимірювань;

n - число спостережень.

У цьому випадку найбільш часто для оцінювання параметрів на практиці використовуються алгоритми методу найменших квадратів (МНК). Однак такий недолік відомого МІЗ, як значна середньоквадратична помилка обмежує сфери його застосування. У зв'язку з цим запропонована двоступенева модифікація МІЗ (ДМІЗ), оцінки якої мають меншу в порівнянні з МІЗ середньоквадратичну помилку. З цією метою матриця спостережень X і оцінюваний вектор \vec{a} розбивалися на блоки

$$X = (X_1, X_2), \vec{a}^T = (\vec{a}_1^T, \vec{a}_2^T), \quad (7.145)$$

а рівняння (7.144) записувалося у вигляді

$$\vec{Y} = X_1 \vec{a}_1 + X_2 \vec{a}_2 + \vec{\varepsilon}. \quad (7.146)$$

Тут $X_1 \in R^{n \times p}$, $X_2 \in R^{n \times q}$, $\vec{a}_1 \in R^p$, $\vec{a}_2 \in R^q$, $p + q = m$, - символ транспонування.

Позначаючи $X_2 \vec{a}_2 + \vec{\varepsilon} = \vec{\chi}_i$ і оцінюючи з отриманого рівняння, отримуємо

$$\vec{\alpha}_1 = (W_1^T X_1)^{-1} W_1^T \vec{Y}$$

Тут матриця ІЗ W розбита також на блоки $W = (W_1, W_2)$, $W_1 \in R^{n \times p}$, $W_2 \in R^{n \times q}$.

Якщо ввести позначення $\vec{Y} = \vec{Y} - X_1 \vec{\alpha}_1$, то оцінка МІЗ вектора запишеться наступним чином:

$$\vec{\alpha}_2 = (W_2^T X_2)^{-1} W_2^T \vec{Y}$$

або з урахуванням виразів для $\vec{\alpha}_1$ і \vec{Y}_1

$$\vec{\alpha}_2 = (W_2^T X_2)^{-1} W_2^T (\vec{Y} - X_1 \vec{\alpha}_1) = (W_2^T X_2)^{-1} W_2^T P \vec{Y}$$

де $P = I - X_1 (W_1^T X_1)^{-1} W_1^T$; I - одинична матриця розмірності $n \times n$.

Таким чином, оцінки ДМІЗ мають вигляд

$$\vec{\alpha}_1 = (W_1^T X_1)^{-1} W_1^T \vec{Y}; \quad (7.147)$$

$$\vec{\alpha}_2 = (W_2^T X_2)^{-1} W_2^T P \vec{Y}. \quad (7.148)$$

Як випливає з (7.147) і (7.148), ДМІЗ включає дві процедури: процедуру оцінювання параметра a_1 і процедуру оцінювання параметра a_2 . Рекурентна процедура оцінювання a_1 може бути отримана за аналогією з рекурентной процедурою МНК.

Записавши оцінку (7.144) для n і $(n - 1)$ -го виміру у вигляді (7.146), після перетворень можна отримати рекурентну процедуру оцінювання $a_{1,n}$.

Для отримання рекурентной процедури оцінювання \vec{a}_2 слід скористатися регресійним рівнянням (7.150), записаним для n і

$(N - 1)$ - го вимірювань. Після перетворень отримуємо

$$\vec{\alpha}_{2,n} = \vec{\alpha}_{2,n-1} + \sum_{i=1}^n (\vec{w}_{2,i} \vec{x}_{2,i}^T)^T [\vec{w}_{2,n} Y_n - W_{2,n} \vec{x}_{1,n} \vec{\alpha}_{1,n-1} - W_{2,n} \vec{x}_{2,n} \vec{\alpha}_{2,n-1} - L_n (\vec{\alpha}_{1,n} - \vec{\alpha}_{1,n-1})],$$

Висновки

1. Розроблено концепцію поточного контролю послідовностей даних за умов викривленості вихідних спостережень.

2. Проведено аналіз існуючих процедур обробки результатів спостережень.

3. Здійснено аналіз впливу зовнішнього середовища на процедуру контролю.

4. Розроблено стратегію контролю на підставі критичного підходу.

5. Запропоновано процедури контролю за умов аномальних спостережень на основі критичного підходу.

6. Розроблено та досліджено методи обробки результатів спостережень на основі критичного підходу.

7. Розроблено адаптивні процедури обробки результатів спостережень за умов аномальних викидів на основі критичного підходу.

ВИСНОВКИ

Отримані результати відповідають світовому рівню досягнень та на сьогодні не мають аналогів, оскільки всі відомі аналоги орієнтовані на обробку даних у пакетному режимі, причому завдання обробки процесів та аномальних викидів розглядаються окремо на основі статистичних методів, що жорстко прив'язані до конкретних типів розподілів. On-line версії сумісної обробки нелінійних сигналів на сьогодні невідомі.

Проведено дослідження та створені адаптивні гібридні online методи кластеризації та класифікації даних, що можуть бути представлені у вигляді таблиць «об'єкт-властивість» або багатовимірних часових рядів. Розроблено нові швидкодіючі методи кластеризації–класифікації, що на відміну від існуючих дозволяють оброблювати інформацію у послідовному режимі реального часу, при цьому як кількість пропусків, так і їх розташування у масивах даних припускаються невідомими. Передбачені режими як відновлення вихідних масивів, так і прямої їх обробки, при цьому кількість пропусків є співрозмірною із кількістю невикривлених даних. За умов надкороткої вибірки даних передбачено використання нового методу просторової екстраполяції вкупі із набором різного типу метрик, що орієнтовані на роботу за умов викривленої інформації. Запропоновані нейро-фаззі методи здатні обробляти суттєво викривлену інформацію в online режимі та на сьогодні не мають аналогів, що підтверджено публікаціями у низці провідних закордонних видань та в матеріалах міжнародних конференцій.

Розроблено та досліджено методи online кластеризації та класифікації даних у вигляді таблиць «об'єкт-властивість» та коротких часових рядів, включаючи багатовимірні векторні та матричні, за наявності аномальних викидів невідомої природи. Методи описані у вигляді рекурентних процедур

або правил нечіткого виведення та орієнтовані на реалізацію в якості алгоритмів навчання-самонавчання гібридних нейро-фаззі систем.

Відмінністю підходу, що пропонується, є використання замість традиційних робастних критеріїв навчання або цільових функцій кластеризації спеціального типу функцій подібності, що придушують аномальні викиди. Ці функції засновані на так званих «часткових метриках», що дозволяє використовувати їх для оброблення даних з пропусками. Таким чином, запропоновані методи є двічі робастними як відносно викидів, так і відносно пропусків, в яких містяться загублені спостереження.

Ще однією особливістю введеного підходу є можливість роботи за умов класів, що перетинаються. Це означає, що одне спостереження може одночасно належати двом або більше класам з різними рівнями належності (в сенсі теорії нечітких множин). Рекурентне представлення методів дозволяє використовувати їх для навчання нечітких самоорганізованих мап, при цьому важливо, що як функції подібності, так і функції належності, так і функції сусідства самоорганізованих мап є за своєю сутністю ядерними (дзвонуватими) конструкціями, що можуть мати однотипну форму кошіанів. Це значно спрощує чисельну реалізацію методів та підвищує швидкодію процесів опрацювання викривлених даних.

Були розроблені архітектури та правила on-line навчання нейро-фаззі систем кластеризації, класифікації та діагностування на основі викривлених даних, що послідовно надходять на обробку, при чому ці дані містять як пропуски, так і аномальні викиди. При чому в процесі навчання-самонавчання передбачено налаштування не лише синаптичних ваг, але й функцій активації-належності та власне архітектури.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бодянский Е.В., Кучеренко Е.И., Михалев А.И., Филатов В.А., Гасик М.М., Куцин В.С. Интеллектуальное управление технологическими процессами. - Днепропетровск: НМАУ, 2013. – 213 с.
2. Bodyanskiy Ye., Shafronenko A. Robust adaptive fuzzy clustering for data with missing values // Eds. by G. Setlak, K. Markov “Computational Models for Business and Engineering Domains”. – Rzeszow-Sofia: ITHEA, 2014. – P. 34-43.
3. Bodyanskiy Ye., Tyshchenko O., Kopalani D. The least squares support vector machine based on a neo-fuzzy neuron // Eds. by G. Setlak, K. Markov “Computational Models for Business and Engineering Domains”. – Rzeszow-Sofia: ITHEA, 2014. – P. 44-51.
4. Bodyanskiy Ye., Vynokurova O., Pliss I., Peleshko D. Multilayer neuro-fuzzy system for solving on-line diagnostic tasks // Eds. by G. Setlak, K. Markov “Computational Models for Business and Engineering Domains”. – Rzeszow-Sofia: ITHEA, 2014. – P. 52-59.
5. Bodyanskiy Ye., Tyshchenko O., Deineko A. Evolving Neuro-Fuzzy Systems with Kernel Activation Functions. – Saarbrucken: LAP LAMBERT Academic Publishing, 2015. – 64 p.
6. Bodyanskiy Ye., Vynokurova O. Hybrid adaptive wavelet-neuro-fuzzy system for chaotic time series identification // Information Sciences. – 2013. – 220. - P.170-179.
7. Bodyanskiy Ye., Vynokurova E.A., Dolotov A.I. Self-learning cascade spiking neural network for fuzzy clustering based on Group Method of Data Handling // J. of Automation and Information Sciences. – 2013. – 45. – №3. – P.23-33.
8. Bodyanskiy Ye., Kolchygin B. Adaptive fuzzy clustering with a variable fuzzifier // Cybernetics and System Analysis. – 2013. – 49. - №3. – P.176-181.

9. Bodyanskiy Ye., Kalynychenko O., Chalyi S., Golian V., Golian N. Implementation of search mechanism for implicit dependences in process mining // Proc. IEEE 7th Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS). – V. 1. – Berlin, Germany, Sept. 12-14, 2013. – P. 138-142.
10. Bodyanskiy Ye., Dolotov A., Vynokurova O. Evolving spiking wavelet-neuro-fuzzy self-learning system // Applied Soft Computing. – 2014. – 14. – P. 252-258.
11. Bodyanskiy Ye., Tyshchenko O.K., Kopalani D.S. A hybrid cascade neural network with an optimized pool in each cascade // Soft Computing. A Fusion of Foundations, Methodologies and Applications. – Soft Computing. - DOI 10.1007/s00500-014-1344-3.
12. Bodyanskiy Ye., Boiko O.O., Pliss I.P. Adaptive method of hybrid learning for an evolving neuro-fuzzy system // Cybernetics and Systems Analysis. – 2015. – 51. – 4. – P. 13-18.
13. Bodyanskiy Ye., Tyshchenko O.K., Deineko A.O. An Evolving radial basis neural network with adaptive learning of its parameters and architecture // Automatic Control and Computer Science. – 2015. – 49. – 5. – P. 255-260.
14. Черненко П.О., Мартинюк О.В., Попов С.В., Бодянский Є.В. Порівняльний аналіз двох підходів до вирішення задач короткострокового прогнозування сумарного електричного навантаження електроенергетичної системи // Технічна електродінаміка. – 2013. - №3. – С. 61-72.
15. Bodyanskiy Ye., Setlak G., Peleshko D, Vynokurova O. Hybrid Generalized Additive Neuro-Fuzzy System and its Adaptive Learning Algorithms // The 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 24-26 September 2015, Warsaw, Poland. – P. 328-333.
16. Бодянский Е.В., Винокурова Е.А., Долотов А.И. Самообучающаяся каскадная спайк-нейронная сеть для нечеткой кластеризации на основе метода группового учета аргументов // Проблемы управления и информатики. – 2013. – №2. – С. 25-34.

17. Бодянский Е.В., Колчигин Б.В. Адаптивная нечеткая кластеризация с переменным фаззификатором // Кибернетика и системный анализ. – 2013. – №3. – С.47-55.
18. Бодянский Е.В., Колчигин Б.В., Волкова В.В., Плисс И.П. Адаптивная нечеткая кластеризация данных на основе метода Густафсона-Кесселя // Управляющие системы и машины. – 2013. - №2. – С.40-46.
19. Bodyanskiy Ye., Shafronenko A., Volkova V. Adaptive fuzzy probabilistic clustering of incomplete data // Int. J. “Information Models and Analyses”. – 2013. – 2. - №2. – P.112-117.
20. Bodyanskiy Ye., Tyshchenko O., Pliss I. Reservoir forecasting neuro-fuzzy network and its learning // Int. J. “Information Theories and Applications”. – 2013. – 20. - №3. – P.203-209.
21. Bodyanskiy Ye., Tyshchenko O., Wojcik W. Multivariate non-stationary time series predictor based on an adaptive neuro-fuzzy approach // Elektronika – konstrukcje, technologie, zastosowania. – № 8. – 2013. – P. 10-13.
22. Bodyanskiy Ye., Pliss I., Vynokurova O. Flexible neo-fuzzy neuron and its adaptive learning algorithms // Proc. 20th East-West Zittau Fuzzy-Kolloquium. – Zittau-Goerlitz: HS, 2013. – P. 47-54.
23. Bodyanskiy Ye., Shafronenko A., Volkova V. Neuro-fuzzy Kohonen network for incomplete data clustering using optimal completion strategy // Proc. 20th East-West Zittau Fuzzy-Kolloquium. – Zittau-Goerlitz: HS, 2013. – P. 214-223.
24. Bodyanskiy Ye., Dolotov A. A spiking neuron model based on the Lambert W-function // Proc. 5th Int. Joint Conf. on Computational Intelligence. – Vilamoura, Algarve, Portugal, 20-22 Sept., 2013. – P. 542-546.
25. Bodyanskiy Ye.V., Ryabova N.V., Zolotukhin O.V. Multilayer adaptive fuzzy probabilistic neural network in classification problems // Radioelectronics, Computer Science, Control. – 2014. - №1. – P. 38-45.

26. Бодянский Е.В., Колчигин Б.В., Волкова В.В., Плисс И.П. Адаптивная нечеткая кластеризация данных на основе метода Густафсона-Кесселя // Управляющие системы и машины. – 2013. - №2. – С.40-46.
27. Bodyanskiy Ye., Pliss I., Vynokurova O., Flexible neo-fuzzy-neuron and neuro-fuzzy network for monitoring time-series properties // Sci. J. of Riga Technical University: Information Technology and Management Science. – 2013. – 16. – P.47-52.
28. Bodyanskiy Ye., Samitova V., Tyshchenko O. Robust fuzzy data clustering in an ordinal scale based on a similarity measure // Int. J. of Research in Engineering and Science. – 2014. – 2. - №4. – P.21-25.
29. Bodyanskiy Ye., Tyshchenko O., Kopaliani D. A multidimensional cascade neuro-fuzzy system with neuron pool optimization in each cascade // Int. J. Information Technology and Computer Science. – 2014. – 6. - №8. – P. 11-17.
30. Бодянский Е.В., Винокурова Е.А., Пелешко Д.Д. Информационная технология кластеризации данных в условиях короткой обучающей выборки на основе ассоциативной вероятностной нейро-фаззи системы // Управляющие системы и машины. – 2014. – 4. – С. 73-76.
31. Бодянский Е.В., Мулеса П.П., Винокурова О.А., Плисс И.П. Алгоритм навчання діагностуючої нейро-фаззі-системи в задачах біомедичних досліджень // Proc. XXIII Int. Conf. Problems of Decision Making Under Uncertainties. – Mukachevo-Kyiv. – 2014. – P. 66-67.
32. Bodyanskiy Ye., Tyshchenko O.K., Kopaliani D.S. An Extended Neo-Fuzzy Neuron and its Adaptive Learning Algorithm // I.J. of Intelligent Systems and Applications (IJISA). - Vol.7, No. 2. - 2015. – P. 21-26.
33. Bodyanskiy Ye., Tyshchenko O.K., Deineko A.O. An Evolving Neuro-Fuzzy System with Online Learning/Self-learning // I.J. of Modern Education and Computer Science (IJMECS). - Vol.7, No. 2. - 2015. – P. 1-7.
34. Bodyanskiy Ye., Kharchenko O., Vynokurova O. Least squares support vector machine based on wavelet-neuron // Sci. J. of Riga Technical University: Information Technology and Management Science. – 17. – 2014. – P. 19-24.

35. Бодяньський Є.В., Плісс І.П., Винокурова О.А. Швидкодiюча узагальнена адитивна вейвлет-нейро-фаззи система для задач динамiчного iнтелектуального аналізу даних// Proc. XXV Int. Conf. “Problems of Decision Making under uncertainties” (PDMU-2015). – Skhidnytsia, Ukraine, 2015. – P. 71-72.

36. Bodyanskiy Ye., Peleshko D., Vynokurova O., Tatarinova Yu. Architecture of Hybrid Generalized Additive Neuro-Fuzzy System in Modelling Technological Process // Proc. XIII-th Int. Conf. “The Experience of Designing and Application of CAD Systems in Microelectronics”. – Lviv-Polyana, Ukraine, 2015. – P. 333-335.

37. Bodyanskiy Ye., Setlak G., Pliss I.P., Vynokurova O. Hybrid neuro-neo-fuzzy system and its adaptive learning algorithm // Proc. Xth International Scientific and Technical Conference «Computer science and information technologies». – Lviv. – Ukraine. – 2015. – P. 111-114

38. Kulishova N., Bodyanskiy Ye. Flexible 2D membership functions for images filtering using fuzzy peer group approach // Proc. Xth International Scientific and Technical Conference «Computer science and information technologies». – Lviv. – Ukraine. – 2015. – P. 82-84

39. Bodyanskiy Ye., Tyshchenko O., Kopaliani D. An evolving neuro-fuzzy system for on-line fuzzy clustering // Proc. Xth International Scientific and Technical Conference «Computer science and information technologies». – Lviv. – Ukraine. – 2015. – P. 158-161.

40. Bodyanskiy Ye., Tyshchenko O., Kopaliani D. An evolving cascade neuro-fuzzy system for data stream fuzzy clustering // Int. J. of Computer Science and Mobile Computing. – 2015. – 4. – 9. – P.270-275.

41. Загоруйко, Н.Г. Эмпирические предсказания / Н.Г. Загоруйко. – Новосибирск: Наука, 1979. – 120с.

42. Gorban, A. Principal Manifolds for Data Visualization and Dimension Reduction. Lecture Notes in Computational Science and Engineering/ A. Gorban,

B. Keg1, B. Wunsch, A. Zinovyev (Eds.). – Berlin-Heidelberg - New York: Springer, 2007. - Vol. 58. – 330 p.

43. Han, J. Data Mining: Concepts and Techniques / J. Han, M. Kamber – Amsterdam: Morgan Kaufman Publ., 2006. – 743p.

44. Marwala, T. Computational Intelligence for Missing Data Imputation, Estimation, and Management: Knowledge Optimization Techniques / T. Marwala. – Hershey - New York: Information Science Reference, 2009. – 303p.

45. Bishop, C.M. Neural Networks for Pattern Recognition / C.M. Bishop.– Oxford: Clarendon Press, 1995. – 482 p.

46. Gorban, A.N. Neural network modeling of data with gaps / A.N. Gorban, A.A. Rossiev, D.C. Wunsch // Радиоэлектроника. Информатика. Управление, 2000. – №1 (3). – С. 47 - 55.

47. Tkacz, M. Artificial neural networks in incomplete data sets processing / M. Tkacz // Eds. M.A. Klopotek, S.T. Wierzchon, K. Trojanowski Intelligent Information Processing and Web Mining. – Berlin - Heidelberg: Springer – Verlag, 2005. – P.577 – 583.

48. Шафроненко, А.Ю. Кластеризация данных с пропусками на основе нечеткого возможностного подхода / А.Ю. Шафроненко, И.П. Плисс, Е.В. Бодянский // Международная научно-техническая конференция «Автоматизация: проблемы, идеи, решения», 3-7 сентября 2012 г.: матер. конф. – Севастополь, 2012. - С. 246-247.

49. Бодянский, Е.В. Многошаговые оптимальные учредители многомерных нестационарных стохастических процессов / Е.В. Бодянский, И.П. Плисс, Т.В. Соловьева // Доклады АН УССР. – №12. - 1986. – С. 47-49.

50. Bodyanskiy, Ye. An adaptive learning algorithm for a neuro – fuzzy network / Ye. Bodyanskiy, V. Kolodyazhniy, A. Stephan // Ed. by B.Reusch “Computational Intelligence. Theory and Applications”. – Berlin – Heidelberg – New York: Springer, 2001. – P.68 – 75.

51. Haykin, S. Neural Networks. A Comprehensive Foundation / S. Haykin. - Upper Saddle River, N.J.: Prentice Hall, Inc., 1999. – 842 p.

52. Yamakawa, T. F neo-fuzzy neuron and its applications to system identification and prediction of the system behavior / T. Yamakawa, E. Uchino, T. Miki, H. Kusanagi //Proc. 2-nd Int. Conf. on Fuzzy Logic and Neural Networks "IIZUKA-92", 17-22 July. -Japan, 1992. – P.477-483.

53. Uchino, E. Soft computing based signal prediction, restoration, and filtering / E. Uchino, T. Yamaka // Ed. Da Ruan "Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks". – Boston: Kluwer Academic Publ., 1997. –P. 331-349.

54. Miki, T. Analog implementation of neo-fuzzy neuron and its on-board learning / T. Miki, T. Yamakowa // Ed. N.E. Mastorakis "Computational Intelligence and Applications". – Piraeus: WSES Press, 1999. – P.144-149.

55. Hathaway, R.J. Fuzzy c-means clustering of incomplete data / R.J. Hathaway, J.C Bezdek // IEEE Trans on Systems, Man, and Cybernetics. - 2001.- №5 (31). - P. 735-744.

56. Bezdek, J.C. Pattern Recognition with Fuzzy Objective Function Algorithms / J.C. Bezdek. - Plenum Press, New York, 1981.

57. Суетин, П.К. Классические ортогональные многочлены / П.К. Суетин. – Москва: Наука, 1976. – 328с.

58. Karlin, S. Systems with Applications in Analysis and Statistics / S. Karlin, W. Studden, J. Tchebycheff – N.Y. - London - Sydney: Interscience Publishes, 1966. – 586р.

59. Семесенко, М.П. Методы обработки и анализа измерений в научных исследованиях / М.П. Семесенко. – Киев-Донецк: Вища школа, 1983. – 240с.

60. Бодянский, Е.В. Ортосинапс, ортонейроны и нейропредиктор на их основе / Е.В. Бодянский, Е.А. Викторов, А.Н. Слипченко // Системи обробки інформації. – 2007. – Вип.4(62). – С.139-143.

61. Yang, S.-S. An orthonormal neural network for function approximation / S.-S. Yang, C.- S. Tseng // IEEE Trans. on Syst., Man, and Cybern. – 1996. - №10. – P.779-784.

62. Lee, T.T. The Chebychev polynomial-based unified model neural networks for functional approximation / T.T. Lee, J.T. Jeng // IEEE Trans. on Syst., Man, and Cybern.– 1998. – №12(28). – P.925-935.

63. Andras, P. Orthogonal RBF neural network approximation / P. Andras // Neural Processing Letters.– 1999. – 9(2). – P.141-151.

64. Chien, F.Sh. Properties and performance of orthogonal neural network in function approximation / F.Sh. Chien, Ch.-Ch. Tseng, Ch.-S. Chen // Int. J. Intelligent Systems.– 2001. – 16. – P.1377-1392.

65. Patra, J.C. Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks / J.C. Patra, A.C. Kot // IEEE Trans. on Syst., Man, and Cybern. – 2002. - №4.(32). – P.505-511.

66. Bodyanskiy, Ye. Artificial neural network with orthogonal activation functions for dynamic system identification / Ye. Bodyanskiy, V.Kolodyazhniy, O. Slipchenko // Eds. by O.Sawodny, P.Scharff “Sinergies between Information Processing and Automation”. – Aachen: Shaker - Verlag, 2004. – P.122-127.

67. Statiak, B. Fast orthogonal neural network / B. Statiak, M.Yatsymirskyy // Artificial Intelligence and Soft Computing.– 2006. – 4029. – P. 142-149.

68. Rodriguez, N. Orthogonal neural network based prediction for OFDM systems / N. Rodriguez, C. Cubillos // Proc. Electronics, Robust, and Automotive Mechanics Conf., 25-28 Sept., 2007. – CERMA, 2007. -P.225-228.

69. Hongwei, W. Tracking control of robot manipulators based on orthogonal neural network / W. Hongwei, Yu. Shuanghe // Int. J. Modeling, Identification, and Control. – 2010. – 11(1-2).– P.130-136.

70. Bodyanskiy, Ye. Structural and synaptic adaptation in the artificial neural networks with orthogonal activation functions / Ye. Bodyanskiy, V. Kolodyazhniy, O. Slipchenko // Sci. Proc. of Riga Technical University. Comp. Sci., Inf. Technology and Management Sci. – 2004. –№20. – P.69-76.

71. Bodyanskiy, Ye. Ontogenic neural networks using orthogonal activation functions / Ye. Bodyanskiy, O. Slipchenko // Prace naukowe Akademii Ekonomicznej we Wroclawiu. – 2006. -№21. – P.13-20.

72. Bodyanskiy, Ye. Growing neural networks using nonconventional activation functions / Ye. Bodyanskiy, I. Pliss, O. Slipchenko // *Int. J. Information Theories & Applications*. - 2007. - №3 (14). – P.275.

73. Bodyanskiy, Ye. The cascade orthogonal neural network / Ye. Bodyanskiy, A. Dolotov, I. Pliss, Ye. Victorov // In “Advanced Research in Artificial Intelligence”. – Sofia: FOI ITHEA. - 2008. – Vol.2. – P.13-20.

74. Jain, A. Dataclustering: A review / A. Jain, M. Murty, P.Flynn // *ACM Computing Surveys*. – 1999. - №3 (31) – P.264 – 323.

75. Дюк, В., *Data Mining* / В.Дюк, А.Самойленко. – Санкт-Петербург: Питер, 2001. – 368 с.

76. Xu, R. Survey of clustering algorithms / R. Xu, D.Wunsch II // *IEEE Trans on Neural Networks*. – 2005. -№3 (16). – P. 645 – 678.

77. Gan, G. *Data Clustering: Theory, Algorithms and Applications* / G. Gan, Ch. Ma, J. Wu. – Philadelphia: SIAM, 2007. – 466 p.

78. Abonyi, J. *Cluster Analysis for Data Mining and System Identification* / J. Abonyi, B Feil. – Basel: Birkhaeuser, 2007. – 303p.

79. Olson, D.Z. *Advanced Data Mining Techniques* / D.Z. Olson, D. Dursun. – Berlin: Springer, 2008. – 180 p.

80. Kohonen, T. *Self-Organizing Maps* / Kohonen T. - Berlin: Springer-Verlag, 1995 - 501p.

81. Kasabov, N. *Evolving Connectionist Systems* / Kasabov N. – London: Springer-Verlag, 2003. – 307p.

82. Hoppner, F. *Fuzzy Cluster Analysis* / F. Hoppner, F. Klawonn, R. Kruse, T. Runkler. - Wiley, Chichester, 1999. -289 p.

83. Плісс, І.П. Нейромережеве відновлення пропусків у таблицях даних / І.П. Плісс, А.Ю. Шевякова (А.Ю. Шафроненко), Ю.Ю. Шевякова // *Наукові праці: науково-методичний журнал*. – Миколаїв: Вид-во ЧДУ ім. Петра Могили, 2011. – Вип.148. Т.160. Комп'ютерні технології. - С.59-61.

84. Chung, F.L *Fuzzy competitive learning* / F.L. Chung, T. Lee // *Neural Networks*. - 1994. - №3. -P.539-552.

85. Park, D. Gradient based fuzzy c-means (GBFCM) algorithm [Текст] / D.C. Park, I. Dagher // Proc. IEEE Int. Conf. on Neural Networks. – 1984. – P.1626-1631.

86. Шевякова, А.Ю. (Шафроненко, А.Ю.) Адаптивная обработка данных с пропусками / А.Ю. Шевякова (А.Ю. Шафроненко), Ю.Ю.Шевякова, М.З. Стольникова // Международная научная конференция «Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта», 17-21 мая 2011 г.: матер. конф. – Евпатория, 2011. - Том 2. - С. 250-253.

87. Krishnapuram, R. A possibilistic approach to clustering / R. Krishnapuram, J.M. Keller // Fuzzy Systems. – 1993. – №2(1).– P.98-110.

88. Bodyanskiy, Ye. Computational intelligence techniques for data analysis / Ye. Bodyanskiy // Lecture Notes in Informatics.-Bonn: GI.– 2005. -Vol. 72. – P. 15-36.

89. Dave, R.N. Robust clustering methods: A unified view / R.N. Dave, R. Krishnapuram // IEEE Trans. on Fuzzy Systems.– 1997. – №2(5). – P.270-293.

90. Bodyanskiy, Ye. Robust recursive fuzzy clustering algorithms / Ye. Bodyanskiy, Ye. Gorshkov, I. Kokshenev, V. Kolodyazhniy //Proc. East West Fuzzy Coll. – Zittau/Goerlitz: HS, 2005. – P. 301-308.

91. Kokshenev, I. Outlier resistant recursive fuzzy clustering algorithm / I. Kokshenev, Ye. Bodyanskiy, Ye. Gorshkov, V. Kolodyazhniy / In “Computational Intelligence: Theory and Application”. – Ed. by B.Reusch-Advances in Soft Computing.– Berlin-Heidelberg: Springer-Verlag, 2006. -Vol.38. – P. 647-652.

92. Keller, L. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing / L. Keller, R. Krishnapuram, N.R. Pal. – N.Y.:Springer Science + Business Media, Inc., 2005. – 776p.

93. Keller, A. Fuzzy clustering with weighting of data variables / A. Keller, F. Klawonn // Int. J. of Uncertainty, Fuzziness and Knowledge - Based Systems. – 2008. – №6(8). -P.735 - 746.

94. Волкова, В.В. Нечітка кластеризації масивів даних з пропущеними значеннями / В.В.Волкова, А.Ю. Шафроненко // Збірник наукових праць «Індуктивне моделювання складних систем». – 2011. -№ 3. - С. 27-32.

95. Krishnapuram, R. The possibilistic C-means algorithm: Insights and recommendations / R. Krishnapuram, J.M. Keller // IEEE Trans. on Fuzzy Systems. – 1996. – 4(3). – P. 385-393.

96. Klawonn, F. What is fuzzy about fuzzy clustering? Understanding and improving the concept of the fuzzifier./ Klawonn F., Hoëppner F. // Lecture Notes in Computer Science. – Berlin / Heidelberg: Springer, 2003. – 2811. – P. 254-264.

97. Mitaim, S. What is the best shape for a fuzzy set in function approximation? / S. Mitaim, B. Kosko // Proc. 5th IEEE Int. Conf. on Fuzzy Systems «Fuzzy96». – 1996. – 2. – P. 1233-1237.

98. Mitaim, S. Adaptive joint fuzzy sets for function approximation / S. Mitaim, B. Kosko // Proc. Int. Conf. on Neural Networks «ICNN-97». – 1997. – P.537-542.

99. Bodyanskiy, Ye. Computational intelligence techniques for data analysis / Ye. Bodyanskiy // Lecture Notes in Informatics. – Bonn, Germany. – 2005. – 72. – P.15-36.

100. Бодянский, Е.В. Об адаптивном алгоритме нечёткой кластеризации данных / Е.В. Бодянский, Е.В. Горшков, И.В. Кокшенев, В.В. Колодяжный // Адаптивні системи автоматичного управління. – 5(25). – Дніпропетровськ: Системні технології. – 2002. – С. 108-117.

101. Dvoretzky, A. On stochastic approximation / A. Dvoretzky // Proc. 3-rd Berkley Symp. Math. Statistics and Probability. – 1956. – 1. – P. 39-55.

102. Chung, F.L. Unsupervised fuzzy competitive learning with monotonically decreasing fuzziness / F.L. Chung, T. Lee // Proc. Int. Joint Conf. on Neural Networks, 1993. – P. 2929-2932.

103. Fan, J.-L. Supressed fuzzy C-means clustering algorithm / J.-L. Fan, W.-Z.Zhen, W.-X. Xie // Pattern Recognition Letters. – 2003. – 23. – P. 1607-1612.
104. Xie, X.L. A Validity Measure for Fuzzy Clustering / X. L. Xie, G. Beni //IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1991. – 13. – P. 841–847.
105. Gustafson, D.E. Fuzzy clustering with a fuzzy covariance matrix / D.E.Gustafson, W.C. Kessel // IEEE Conference on Decision and Control incl. the 17th Symp. on Adaptive Processes. – 1978. – P. 761-766.
106. Krishnapuram, R. A note on the Gustafson-Kessel and adaptive fuzzyclustering algorithms / R. Krishnapuram, K. Jongwoo // IEEE Transactions on Fuzzy Systems. – 7. – 4. – Aug 1999. – P.453-461.
107. Krishnapuram, R. Fuzzy and possibilistic clustering methods for computervision / R. Krishnapuram, J.M. Keller // Neural Fuzzy Systems. – 1994. – 12. – P. 133-159.
108. Klawonn, F. Constructing a fuzzy controller from data / F. Klawonn, R. Kruse// Fuzy Sets and Systems. – 1997. – 85. – P. 117-193.
109. Hoëppner, F. Fuzzy clustering of sampled functions / F. Hoëppner, F. Klawonn // Proc. 19-th Int. Conf. North American Fuzzy Information Processing Society (NAFIPS). – Atlanta, USA, 2000. – P. 251–255.
110. Georgieva, O. A clustering algorithm for identification of single clusters inlarge data sets / O. Georgieva, F. Klawonn // Proc. 11-th East-West Fuzzy Coll. – Zittau/Goerlitz: HS, 2004. – P. 118–125.
111. Winkler, R. M-Estimator induced Fuzzy Clustering Algorithms / R. Winkler, F. Klawonn, R. Kruse. // European Society for Fuzzy Logic and Technology – «les rencontres francophones sur la Logique Floue et ses Applications» (EUSFLAT-LFA). – Aix-les-Bains, France, 2011. – p.298-304.
112. Winkler, R. A new Distance Function for Prototype based Clustering Algorithms in High Dimensional Spaces / R. Winkler, F. Klawonn, R. Kruse. //

Statistical Models for Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization. – Springer, 2013. – P.371-378.

113. Tsoukalas, L. H. Fuzzy and Neural Approaches in Engineering / L.H.Tsoukalas, R.E. Uhrig. – N.Y.: John Wiley & Sons, Inc. – 1997. – 587 p.

114. Bodyanskiy, Ye. Combined learning algorithm for a self-organizing map with fuzzy inference / Ye. Bodyanskiy, Ye. Gorshkov, V. Kolodyazhnyy, A. Stephan // Computational intelligence: theory and applications. – Advanced in Soft Computing. – Berlin-Heidelberg: Springer-Verlag, 2005. – 3. – P.641-650.

115. Gorshkov Ye. New recursive learning algorithms for fuzzy Kohonenclustering network / Ye. Gorshkov, V. Kolodyazhnyy, Ye. Bodyanskiy // Proc. 17th Int. Workshop on Nonlinear Dynamics of Electronic Systems. – Rapperswil, Switzerland, 2009. – P. 58-61.

116. Rojas, R. Neural Networks. A Systematic Introduction / R. Rojas // Berlin:Springer-Verlag. – 1996. – 502 p.

117. Pouliezos, A.D. Real Time Fault Monitoring of Industrial Processes / A.D. Pouliezos, G.S. Stavrakakis. – Dordrecht:Kluwer Academic Publishers. – 1994. – 542 p.

118. Islam, M.M. A constructive algorithm for training cooperative neural networkensembles. / M.M. Islam, Xin Yao, K. Murase // IEEE Transactions on Neural Networks. – 2003. – 14(4). – P.820-834.

119. Opitz, D.W. Actively searching for an effective neural network ensemble /D.W. Opitz, J.W. Shavlik // Connect. Sci. – 1996. – 8. – P.337-353.

120. Naftaly, U. Optimal ensemble averaging of neural networks / U. Naftaly, N.Intrator, D. Horn // Network: Comput. Neural Syst. – 1997. – 8. – P. 283-296.

121. Dietterich, T.G. Ensemble Methods in Machine Learning / T.G. Dietterich // Int. Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science. – Springer-Verlag. – 1857. – 2000. – P.1-15.

122. Efron, B. Bootstrap methods: another look at the jackknife / B. Efron // The Annals of Statistics. – 1979. – 7(1). – P. 1-26.

123. Polikar, R. Bootstrap inspired techniques in computational intelligence: ensemble of classifiers, incremental learning, data fusion and missing features / R. Polikar // *IEEE Signal Processing Mag.* – 2007. – 24(4). – P.59-72.

124. Muhlbaier, M. Ensemble confidence estimates posterior probability / M.Muhlbaier, A. Topalis, R. Polikar // *6th Int. Workshop on Multiple Classifier Systems (MCS 2005)*. – Springer Lecture Notes in Computer Science (LNCS). – Monterey, CA: Seaside, 2005. – 3541. – P.326-335.

125. Witten, I.H. Data mining: practical machine learning tools and techniques. —3rd ed. / I.H. Witten, F. Eibe, M.A. Hall. – Elsevier. – 2011. – 665 p.

126. Ho, T.K. Random subspace method for constructing decision forests / T.K. Ho // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 1998. – 20(8). – P.832-844.

127. Das, S. Metaheuristic Clustering / S. Das, A. Abraham, A. Konar // *Studies in Computational Intelligence*. – 2009. – 178. – 265 p.

128. Sharkey, A.J.C. On combining artificial neural nets / A.J.C. Sharkey // *Connect. Sci.* – 1996. – 8. – P.299-313.

129. Steeb, W.-H. The Nonlinear Workbook: Chaos, Fractals, Cellular Automata, Genetic Algorithms, Gene Expression Programming, Support Vector Machine, Wavelets, Hidden Markov Models, Fuzzy Logic with C++, Java and SymbolicC++ Programs, 5th Edition / W.-H. Steeb. – Singapore:World Science Publ. – 2011. – 644 p.

130. Bayat, H. Improving Estimation of Specific Surface Area by Artificial Neural Network Ensembles Using Fractal and Particle Size Distribution Curve Parameters as Predictors / H. Bayat, S. Ersahin, E.N. Hepper // *Environmental Modeling & Assessment*. – Springer, Netherlands. – 2013 – 18(5). – P.605-614.

131. Pulido, M. Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange / M. Pulido, P. Melin, O. Castillo // *Information Sciences*. – 2014. – 280. – P.188204.

132. Gerhard, W. Fractals in the nervous system: conceptual implications for theoretical neuroscience // *Frontiers in Physiology*. – 2010. – 1(15). – P.1-15.

133. Dunn, J.C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters / J.C. Dunn // *Journal of Cybernetics*. – American Society for Cybernetics, Washington, DC, USA. – 1973. – 3. – P.32–57.

134. Fukuyama, Y. A New Method of Choosing the Number of Clusters for the Fuzzy c-Means Method. / Y. Fukuyama, M. Sugeno // *Proc. 5th Fuzzy Systems Symposium (in Japanese)*. – Kobe : Japan Society for Fuzzy Sets and Systems, 1989. – P.247–256.

135. Davies, D.L. A Cluster Separation Measure / D.L. Davies D.W. Bouldin // *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. – Piscataway, NJ:IEEE Press, 1979. – 1(4). – P.224–227.

136. Bezdek, J.C. Cluster Validity with Fuzzy Sets / J.C. Bezdek // *Journal of Cybernetics*. – 1973. – 3(3). – P. 58–73.

137. Windham, M.P. Cluster Validity for the Fuzzy c-Means Algorithm / M.P.Windham // *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. – 1982. – 4(4). – P. 357–363.

138. Stutz, C. Partially Supervised Fuzzy c-Means Clustering with Cluster Merging/ C. Stutz // *Proc. 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT'98, Aachen, Germany)*. – 1998. – P.1725–1729.

139. Daniely, A. Clustering is difficult only when it does not matter / A. Daniely, N. Linial, M. Saks // *arXiv:1205.4891v1 [cs.LG]*. – 2012.

140. Айзерман М. А. Метод потенциальных функций в теории обучения машин. / Айзерман М. А., Браверман Э. М., Розоноэр Л. И. – М.: Наука, 1970. – 384 с.

141. Parzen E. On the estimation of a probability density function and the mode / Parzen E. // *Ann. Math. Statist.* – 1962. – 38. – P. 1065-1076.

142. Надарая Э. А. О непараметрических оценках плотности вероятности и регрессии / Надарая Э. А. // Теория вероятностей и ее применение. – 1965. – 10. – № 1. – С. 199-203.

143. Варядченко Т. В. Непараметрический метод обращения функций регрессии / Варядченко Т. В., Катковник В. Я. // Стохастические системы управления. – Новосибирск: Наука, 1979. – С. 4-14.

144. Бодянский Е.В. Нейро-фаззи сети Петри в задачах моделирования сложных систем / Бодянский Е.В., Кучеренко Е.И., Михалев А.И. – Днепропетровск: Системные технологии, 2005. – 311 с.

145. Епанечников В.А. Непараметрическая оценка многомерной плотности вероятности / Епанечников В.А. // Теория вероятностей и ее применение – 1968 – 14 – №1 – С. 156-161.

146. Chiu S. Fuzzy model identification based on cluster estimation / Chiu S. // Journal of intelligent & fuzzy systems. – Vol. 2 – №3 –Sept. 1994.

147. Paetz J Reducing the number of neurons in radial basis function networks with dynamic decay adjustment. / Paetz J. // Neurocomputing – №62 – 2004 – P. 79-91.

148. Platt J. A resource allocating network for function interpolation / Platt J. // Neural Comp. –1991 – 3– P. 213-225.

149. Бодянский Е.В. Адаптивное обучение архитектуры и параметров радиально-базисной нейронной сети / Бодянский Е.В., Дейнеко А.А. // Системные технологии. – Днепропетровск, 2013. – 4. – 87. – С. 166-173.

150. Бодянский Е.В. Эволюционирующая радиально-базисная нейронная сеть и ее обучение с помощью карты Кохонена / Бодянский Е.В., Дейнеко А.А. // Научно-техническая конференция: «Информационные технологии в металлургии и машиностроении.» – Днепропетровск, 2013. – С. 75-77.

151. Rutkowski L. Computational Intelligence. Methods and Tehniques. / Rutkowski L. // Berlin-Heidelberg: Springer-Verlag, – 2008. – 514 p.

152. Jang J.-S. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Maching Intelligence. / Jang J.-S., Sun C.-T., Mizutani E. // Upper Saddle River: Prentice Hall – 1997. – 640 p.

153. Wang L.-X. Fuzzy basis functions, universal approximation and orthogonal least squares learning. / Wang L.-X., Mendel J. M. // IEEE Trans. on Neural Networks – 1993. – 3. – P. 807-814.

154. Cios K. J. Neuro-fuzzy algorithms. / Cios K. J., Pedrycz W. – Oxford: IOP Publishing Ltd and Oxford University Press. // Handbook of Neural Computation, – 1997. – D1. 3:1 – D1. 3:7.

155. Friedman J. The Elements of Statistical Learning. Data Mining, Inference and Prediction. / Friedman J., Hastie T., Tibshirani R. // Berlin: Springer, – 2003.–552 p.

156. Осовский С. Нейронные сети для обработки информации. / Осовский С. // М.: Финансы и статистика. – 2002. – 344 с.

157. Torres J.A. The problem of organizing and partition large data sets in learning algorithms for SOM-RBF mixed structure / Torres J.A., Martinez S., Martinez F.J., Peralta M. // Proc. 5th Int. Joint Conf. on Computational Intelligence. – Vilamoura, Algarve, Portugal – 2013 – P 497-501.

158. Nelles O. Nonlinear System Identification. / Nelles O. // Berlin: Springer, – 2001. – 785p.

159. Bodyanskiy Ye. An adaptive learning algorithm for a neuro-fuzzy network / Bodyanskiy Ye., Kolodyazhniy V., Stephan A. / Ed. by B. Reusch "Computational Intelligence. Theory and Applications." – Berlin – Heidelberg – New York: Springer, 2001. – P. 68-75.

160. Otto P. / A new learning algorithm for a forecasting neuro-fuzzy network. / Otto P., Bodyanskiy Ye., Kolodyazhniy V. // Integrated Computer-Aided Engeneering. – 2003. – 10. – № 4. – P. 399-409.

161. Бодянский Е.В. Искусственные нейронные сети: архитектуры, обучение, применение. / Бодянский Е.В., Руденко О.Г. – Харьков: ТЕЛЕТЕХ. – 2004. – 372 с.

162. Rizzo, R. Computational Intelligence Methods for Bioinformatics and Biostatistics / R. Rizzo. - In Lecture Notes in Bioinformatics (7th International Meeting, CIBIB 2010, Palermo, Italy, September 16-18, 2010). – Springer. – 2011. - 301 p.

163. Kountchev, R. Advances in Intelligent Analysis of Medical Data and Decision Support Systems (Studies in Computational Intelligence) / R. Kountchev and etc - Springer. – 2013. – 246 p.

164. Ljung, L. System Identification: Theory for the User / L. Ljung. - Upper Saddle River, N.J.: PTR Prentice Hall. - 1999. - 672 p.

165. Shynk, J.J. Performance surfaces of a single-layer perceptron / J.J. Shynk // IEEE Trans. on Neural Networks. – 1990. – 1. – P. 268-274.

166. Бодянский, Е.В. Диагностика и прогнозирование временных рядов с помощью многослойной радиально-базисной нейронной сети / Е.В. Бодянский, Е.И. Кучеренко, А.П. Чапланов // Труды 8 Всероссийской конф. с междунар. Участием «Нейрокомпьютеры и их применение». – Москва. – 2002. – С. 209-213.

167. Бодянский, Е.В. Семантическое аннотирование текстовых документов на основе иерархической радиально-базисной нейронной сети / Е.В. Бодянский, О.В. Шубкина // Восточно-европейский журнал передовых технологий. – 2010. - Вып. 9 (90). – С. 70-74.

168. Jang, J.S.R. Functional equivalence between radial basis function networks and fuzzy inference systems / J.S.R. Jang, C.T. Sun // IEEE Trans. on Neural Networks. - 1993. - 4. - P. 156-159.

169. Бодянский, Е.В. Диагностирующая нейро-фаззи-система и ее адаптивное обучение в задачах интеллектуальной обработки данных медико-биологических исследований / Е.В. Бодянский, П.П. Мулеса, И.Г. Перова, Е.А. Винокурова // Системні технології. Регіональний міжвузівський збірник наукових праць. – Дніпропетровськ: ІВК «Системні технології». – 2014. – Вип. 2 (91). – С. 125-135.

170. Бодянський, Є.В. Діагностуюча нейро-фаззі-система в задачах інтелектуальної обробки даних великої розмірності / Є.В. Бодянський, П.П. Мулеса, О.О. Харченко, Е.А. Винокурова // Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту: матер. міжнар. наук. конф., Залізний Порт, 28-31 травня 2014 р. – Херсон: ХНТУ. – 2014. – С. 254-256.

171. Бодянський, Є.В. Алгоритм навчання діагностуючої нейро-фаззі-системи в задачах біомедичних досліджень / Є.В. Бодянський, П.П. Мулеса, І.П. Плісс, О.А. Винокурова // Problems of Decision Making Under Uncertainties: proc. of the XXIII int. conf., Mukachevo, May 12-16, 2014. – Kyiv. – 2014. – P. 66-67.

172. Mulesa, P. Multilayer neuro-fuzzy system and its learning algorithm for diagnostic tasks / P. Mulesa, Ye. Bodyanskiy, O. Vynokurova // Теорія прийняття рішень: матер. VII-ї міжнар. школи-семінару, Ужгород, 29 вересня-4 жовтня 2014 р. – Ужгород: УжНУ. – 2014. – С. 15-17.

173. Wang, L.-X. Adaptive Fuzzy Systems and Control: Design and Stability Analysis / L.-X. Wang. - New Jersey: Prentice Hall. – 1994. – 256 p.

174. Shepherd, A.J. Second-Order Methods for Neural Networks / A.J. Shepherd. - London: Springer-Verlag. – 1997. – 145 p.

175. Цыпкин, Я.З. Основы теории обучающихся систем / Я.З. Цыпкин. – М.: Наука. – 1970. – 252 с.

176. Medjanik Z. Neural network compression of biomedical data / Medjanik Z., Popov S., Teslenko N. // 2-й Международный радиоэлектронный Форум «Прикладная радиоэлектроника. Состояние и перспективы развития» МРФ-2005. Сборник научный трудов. – Харьков, 2005. – Т. III. Международная конференция «Информационные системы и технологии». – С. 318-320.

177. Попов С. В. Обучение автоассоциативной нейронной сети при наличии пропусков в данных / Попов С. В., Тесленко Н. А., Беловол Ю. С. //

Автоматика-2005 : Материалы 12-й Международной конференции по автоматическому управлению. – Харьков, 2005. – Т. 1. – С. 124-125.

178. Roweis S. EM algorithm for PCA and SPCA / Roweis S. // *Neural Information Processing Systems*. – 1997. – Vol. 10. – P. 626-632.

179. Popov S. Large-scale data visualization with missing values / Popov S. // *Technological and economic development of economy*. – Vilnius : Technika, 2006. – Vol. XII, № 1. – P. 44-49.

180. Popov S. Nonlinear Visualization of Incomplete Data Sets / Popov S. // *Lecture Notes in Computer Science* – Berlin : Springer, 2006. – Vol. 3967. – P. 524-533.

181. Пат. 79573 UA, МПК G06F 19/00, G06F 17/00, G06F 7/00, G01N 33/48. Спосіб оцінки біологічних станів, заснований на інтелектуальному аналізі даних множини вимірюваних показників / Бодянский Є. В., Божков А. І., Бондаренко М. Ф., Дьомін О. О., Клімова О. М., Кушнар'ов В. М., Леонтьєва Ф. С., Попов С. В. ; опубл. 25.06.2007, Бюл. №9.

182. Растрин, Л.А. Экстраполяционные методы проектирования и управления / Л.А. Растрин, Ю.П. Пономарев. – М.: Машиностроение, 1986. – 120 с.

183. Алберт, А. Регрессия, псевдоинверсия и рекуррентное оценивание; пер с англ. под ред. Я.З. Цыпкина. – М.: Наука, 1977. – 22 с.

184. Бодянский, Е.В. Искусственные нейронные сети: архитектуры, обучение, применения / Е.В. Бодянский, О.Г. Руденко. – Харьков: ТЕЛЕТЕХ, 2004. – 372 с.

185. Ахиезер, Н.И. Лекции по теории аппроксимации / Н.И. Ахиезер – М.: Наука. – 408 с.

186. Карлин, С. Чебышевские системы и их применение в анализе и статистике / С. Карлин, В. Стадден; пер с англ. под ред. С.М. Ермакова. – М.: Наука. – 568 с.

187. Семесенко, М.П. Методы обработки и анализа измерений в научных исследованиях / М.П. Семесенко. – К.: Выща школа, 1983. – 240 с.

188. Чебышев, П.Л. Об интерполировании. Избранные труды / П.Л. Чебышев. – М.: Физматгиз, 1965. – 661 с.
189. Льюнг, Л. Идентификация систем. Теория для пользователя / Л. Льюнг; пер с англ. под ред. Я.З. Цыпкина. – М.: Наука, 1991. – 432 с.
190. Haykin, S. Neural Networks: A Comprehensive Foundation / S. Haykin. – Upper Saddle River, N. J.: Prentice-Hall, Inc., 1999. – 842 p.
191. Pao, Y.H. Adaptive pattern recognition and neural networks / Y.H. Pao. – Addison-Wesley: Reading, 1989. – 320 p.
192. Lee, T.T. The Chebyshev polynomial-based unified model neural networks for function approximation / T.T. Lee, J.T. Jeng // IEEE Transactions on Systems, Man, and Cybernetics. – 1998. – Vol. 28. – No. 12. – P. 925-935.
193. Yang, S.-S. An ortonormal neural network for function approximation / S.-S. Yang, C.-S. Tseng // IEEE Transactions on Systems, Man, and Cybernetics. – 1996. – Vol. 26. – No. 12. – P. 925-935.
194. Patra, J.C. Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks / J.C. Patra, A.C. Kot // IEEE Transactions on Systems, Man, and Cybernetics. – 2002. – Vol. 32, No. 4. – P.505-511.
195. Бодянский, Е.В. Ортосинапс, ортонейроны и нейропредиктор на их основе. / Е.В. Бодянский, Е.А. Викторов, А.Н. Слипченко // Системи обробки інформації. – 2007. – Вип.4(62). – С.139-143.
196. Мантула, Е.В. Адаптивная полиномиальная нейросетевая прогнозирующая модель временных рядов и ее обучение / Е.В. Мантула, С.В. Машталир // Восточно-Европейский журнал передовых технологий. Математика и кибернетика – прикладные аспекты. – 2014. – № 2/4(68) – С.16-20.
197. Мантула, Е.В. Адаптивное прогнозирование временных рядов при неравностоящих наблюдениях / Е.В. Мантула, С.В. Машталир // Біоніка інтелекту. – 2013. – № 2 (81). – С.53-57.

198. Mantula, E. Method of adaptive forecasting based on multidimensional linear extrapolation / E. Mantula., S. Mashtalir // International Journal of Research in Engineering and Science. – 2013. – Vol., 1 No. 4. – P. 31-36.

199. Кириченко Л.О. Разработка алгоритмов принятия решений в экспертной системе фрактального анализа / Л.О. Кириченко, А.В. Стороженко, Ю.А. Кобицкая // «Системні технології» – збірник наукових праць. – №3 (86). -2013. – С.54-61.

200. КириченкоЛ.О. Комплексный подход к исследованию фрактальных временных рядов / Л.О. Кириченко, Л.Э. Чалая // Information Technologies & Knowledge, International Journal. - 2014. – Vol. 8. – № 1. – P. 22-28.

201.Kirichenko L.The method of distinction monofractal and multifractal process from time series / L. Kirichenko // Системні технології. – 2014. – №6 (95). – С. 121–127.

202. Дербенцев В.Д. Синергетичні та еконофізичні методи дослідження динамічних та структурних характеристик економічних систем / В.Д. Дербенцев, О.А. Сердюк, В.М. Соловійов, О.Д. Шарапов // Монографія. – Черкаси: Брама-Україна, 2010. – 287 с.

203. Eckmann J.P. Recurrence Plots of Dynamical Systems / J.P. Eckmann, S.O. Kamphorst, D. Ruelle / J.P. Eckmann // Europhysics Letters 5. – 1987. – P. 973–977.

204. Marwan N. Recurrence-plots-based measures of complexity and application to heart-rate-variability data / N. Marwan, N. Wessel, U. Meyerfeldt, A. Schirdewan, J. Kurths // Physical Review. – E66. – 2002.

205. Zbilut J.P. Recurrence quantification based Liapunov exponent for monitoring divergence in experimental data / J.P. Zbilut, J.-M. Zaldivar-Comenges, F. Stozzi // Phys. Lett. A. – V. 297, – 2002. – P. 173–181.

206. Шустер Г. Детерминированный хаос: Введение. / Г. Шустер. – М.: Мир, 1988. – 240 с.

207. Мун Ф. Хаотические колебания / Ф. Мун. – М.: Мир, 1990. –304 с.

208. Pincus S.M. Approximate entropy as a measure of system complexity / S.M. Pincus // *Natl. Acad. Sci.* – Vol.88. – P. 2297-2301.
209. Joshua S. Physiological time–series analysis using approximate entropy and sample entropy / S. Joshua, J. Richman, R. Moorman // *Am. J Physiol. Heart Circ. Physiol.* 278: H2039-H2049, 2000.
210. Малла С. Вейвлеты в обработке сигналов: Пер. с англ / С. Малла. – М.: Мир, 2005. – 671 с.
211. Смоленцев Н. К. Основы теории вейвлетов. Вейвлеты в MATLAB / Н.К. Смоленцев. – М. : ДМК Пресс, 2005. – 304 с.
212. PhysioNet: the research resource for complex physiologic signals [Электронный ресурс] – Режим доступа: www.physionet.org.
213. Kirichenko L. Comparative Analysis of the Complexity of Chaotic and Stochastic Time Series / L. Kirichenko, Yu. Kobitskaya, A. Nabacheva // «Радіоелектроніка. Інформатика. Управління» – №2 (31). – 2014 – С. 126–134.
214. Кириченко Л.О. Использование вейвлет-характеристик временных рядов в экспертной системе / Л.О. Кириченко, Ю.А. Кобицкая, А.В. Стороженко // *Системні технології*. – 2015. – Вып. 3(98). – С. 72-78.
215. Кириченко Л.О. Анализ и распознавание реализаций сигналов, обладающих фрактальными свойствами / Л.О. Кириченко, Ю.А. Кобицкая, Н.А. Дёмина // *Бионика интеллекта*. – 2015. – № 1(84). – С. 49-55.
216. Mirkin B.G. Clustering for Data Mining. A Data Recovery Approach / B.G. Mirkin // Taylor & Francis Group, LLC. – 2005. – 278 p.
217. Duda R.O. Pattern classification / R.O. Duda, P.E. Hart, D.G. Stork. – Wiley & Sons, 2010. – 738 p.
218. Kaski P. Classification Algorithms for Codes and Designs / P. Kaski, P.R.J. Ostergard // Springer. – 2006. – 413 p.
219. Dasgupta D., Immunological computation, theory and applications / D. Dasgupta, L. F. Nino / CRC Press. – 2009. – 298 p.
220. Timmis J. A resource limited AIS for data analysis / J. Timmis, M. Neal // *ELSEVIER, Knowledge Based Systems*. – 2001. – № 3. – Vol. 14 – P. 121-

130.

221. Watkins A.B. A resource limited artificial immune classifier / A.B. Watkins, L.C. Boggess // IEEE, World Congress on Computational Intelligence. – 2002. – P. 926-931.

222. Литтл Р.Дж.А. Статистический анализ данных с пропусками / Р.Дж.А. Литтл, Д.Б. Рубин. – М.: Финансы и статистика, 1991. – 430 с.

223. Кораблев Н.М. Классификация объектов на основе искусственных иммунных систем / Н.М. Кораблев, А.А. Фомичев // Системы обработки информации. – 2010. – Вып. 6 (87). – С. 13-17.

224. Кораблев Н.М. Решение задачи кластеризации данных при помощи искусственных иммунных систем / Н.М. Кораблев, А.А. Фомичев // Сборник трудов IX международной конференции «Интеллектуальный анализ информации ИАИ-2009». – К.: Просвіта, 2009. – С. 171-177.

225. Кораблев Н.М. Кластеризация данных на основе искусственных иммунных систем / Н.М. Кораблев, А.А. Фомичев // Системи обробки інформації: збірник наукових праць. – 2009. – Вип. 4 (78). – С. 77-82.

226. Кораблев Н.М. Автоматическая классификация данных на основе иммунного подхода / Н.М. Кораблев, А.А. Фомичев // Бионика интеллекта: науч.-техн. журнал. – 2014. – № 2 (83). – С. 83-90.

227. Кораблев Н.М. Исследование иммунных операторов в задаче кластеризации объектов / Н.М. Кораблев, А.А. Фомичев // Бионика интеллекта: науч.-техн. журнал. – 2010. – № 1 (72). – С. 70-74.

228. Korablyov M. The immune method for classifying objects on the basis of the target clonal selection (Immunologiczne metody klasyfikacji obiektów bazujące na selekcji klonalnej) / M. Korablyov, O. Fomichov, M. Kushnaryov, W. Wójcik // Elektronika (LIV). – 2013. – № 8. – P. 36-39.

229. Кораблев Н.М. Модификация иммунного метода RLAIIS для автоматической классификации объектов / Н.М. Кораблев, А.А. Фомичев, М.В. Кушнарев // Проблеми інформаційних технологій: наук.-техн. журнал. – 2014. – № 02(016). – С. 29-38.

230. Кораблёв Н.М. Классификация объектов с помощью иммунного метода ближайших соседей / Н.М. Кораблев, А.А. Фомичев // Системи управління, навігації та зв'язку – К.: ЦНДІ НУ, 2011. – Вип. 2 (18). – С. 114 – 118.

231. Кораблёв Н.М. Нечеткая классификация объектов на основе искусственных иммунных систем / Н.М. Кораблев, А.А. Фомичев, М.В. Кушнарєв // Наук. вісник Чернівецького ун-ту «Комп'ютерні системи та компоненти»: збірник наукових праць. – 2010. – Том 1. Вип.2. – С. 88-94.

232. Кораблев Н.М. Нечеткая кластеризация данных на основе искусственных иммунных систем / Н.М. Кораблев, А.А. Фомичев // Вестник ХНТУ. – 2010. – № 2 (38). – С. 323-328.

233. Кораблёв Н.М. Кластеризация данных методом k-means с использованием иммунных операторов / Н.М. Кораблев, А.А. Фомичев // Бионика интеллекта: науч.-техн. журнал. – 2011. – № 3 (77). – С. 102-106.

234. Лукашин Ю. П. Адаптивные методы краткосрочного прогнозирования временных рядов / Лукашин Ю.П. – М. : Финансы и статистика, 2003. – 416 с.

235. Han J. Data Mining Concepts and Techniques. Second Edition / J. Han, M. Kamber – Elsevier, 2006. – 772 p.

236. Снитюк В. Е. Эволюционный метод восстановления пропусков в данных / В. Е. Снитюк // Сборник трудов VI-й Международной конференции «Интеллектуальный анализ информации». – Киев, 2006. – С. 262–271.

237. Бучацкая В. В. Обработка аномальных значений уровней временного ряда как этап комплексной оценки информации в подсистеме прогнозирования для ситуационного центра / Бучацкая В. В. // Вестник Адыгейского государственного университета. Сер.: Естественно-математические и технические науки. - 2013. – Вып. 3. - С. 98-102.

238. Батуро А. П. Финансовые временные ряды: кусочное прогнозирование и проблема обнаружения предвестников существующего

изменения закономерности / А. П. Батуро, Н. М. Еременко // *Банковские технологии.* – 2001.– № 12.– С. 70–77.

239. Makridakis S. The M-3 Competition: Results, Conclusions and Implications / S. Makridakis, M. Hibon // *International of Forecasting.* – 2000. – Amsterdam: Elsevier. – № 16. – P. 451–476.

240. Черный С. Г. Применение case based reasoning для поддержки принятия решений / С. Г. Черный // *Вестник ХНТУ.* – 2010.– № 2 [38].– С. 336–342.

241. Дасгупта Д. Искусственные иммунные системы и их применение / Дасгупта Д.; пер. с англ. под ред. А. А. Романюхи. – М. : ФИЗМАТЛИТ, 2006. – 344 с.

242. Бидюк П. И. Алгоритм клонального отбора для прогнозирования нестационарных динамических систем / П. И. Бидюк, В. И. Литвиненко, И. В. Баклан, А. А. Фефелов // *Искусственный интеллект.* – Херсон, 2004. – № 4.– С. 89–99.

243. Прасолов С. В. Постановка задачи прогнозирования, основанная на применении упрощенной модели искусственной иммунной сети / С.В. Прасолов, Д.В. Шойтов // *Ученые записки. Электронный научный журнал Курского государственного университета.* – Курск, 2009. – Т. 2. – С. 1–3.

244. Кораблев Н. М. Применение искусственных иммунных сетей для прогнозирования временных рядов / Н. М. Кораблев, Г. С. Иващенко // *Системаи обробки інформації.* – Харьков, 2012. – № 9 [107]. – С. 42–45.

245. Кораблев Н. М. Применение модели клонального отбора, использующей вывод по прецедентам, для прогнозирования временных рядов / Н. М. Кораблев, Г. С. Иващенко // *Бионика интеллекта.* – Харьков, 2013. – № 1 [80]. – С. 108–111.

246. Демидова Л.А. Модифицированный алгоритм клонального отбора для анализа временных рядов с короткой длиной актуальной части / Л.А. Демидова, А.В. Корячко, Т.С. Скворцова // *Системы управления и информационные технологии.* – 2010. – № 4. – С. 131-136.

247. Воеводин В.В. Параллельные вычисления. / В.В. Воеводин, Вл.В. Воеводин – СПб.: БХВ-Петербург, 2002. – 475 с.
248. Timothy G.M. Patterns for Parallel Programming / G. Timothy, V. Mattson, A. Sanders, L. Berna // Massingill. – Addison-Wesley, 2004. – 384 p.
249. Монахов, О. Г. Параллельный генетический алгоритм для оптимизации торговых стратегий / О. Г. Монахов // Сибирский журнал вычислительной математики РАН. Сибирское отделение. – Новосибирск, 2008. – Т. 11, № 4. – С. 423-432.
250. Чучуева И. А. Модель экстраполяции временных рядов по выборке максимального подобия / И. А. Чучуева // Информационные технологии. – Москва, 2010. – № 12.– С. 43–47.
251. Кораблев Н.М. Гибридный метод клонального отбора для прогнозирования временных рядов, имеющих пропущенные значения / Н.М. Кораблев, Г.С. Иващенко // Электротехнические и компьютерные системы: науч.-техн. журнал. – К.: Техника – 2014. – № 13 (89). – С. 170-177.
252. Кораблев Н.М. Гибридный метод краткосрочного прогнозирования временных рядов на основе модели клонального отбора / Н.М. Кораблев, Г.С. Иващенко // 16-я всероссийская научно-техн. конференция с международным участием «Нейроинформатика-2014», 27-31 января 2014 г.: сборник научных трудов. – Москва, 2014. – Часть 1. – С. 79-89.
253. Ivaschenko G. S. Time series forecasting on the basis of the case-based reasoning using the models of artificial immune systems / G. S. Ivaschenko, N. M. Korablev // System technologies. – 2014. – № 6 [95]. – P. 43-51.
254. Korablev N. M. Parallel immune algorithm of short-term forecasting based on model of clonal selection / N. M. Korablev, G. S. Ivaschenko // Radio Electronics, Computer Science, Control. – 2014. – № 2. – P. 73-78.
255. Кораблев Н.М. Краткосрочное прогнозирование временных рядов, содержащих аномальные значения, при помощи моделей

искусственных иммунных систем / Н.М. Кораблев, Г.С. Иващенко // Бионика интеллекта: науч.-техн. журнал. – 2015. – № 2 (83). – С. 83-90.

256. Беляев Л.С. Решение сложных оптимизационных задач в условиях неопределенности / Л.С. Беляев. – Новосибирск : Наука, 1978. – 128 с.

257. Трухаев Р.И. Модели принятия решений в условиях неопределенности / Р.И. Трухаев. – М. : Наука, 1981. – 258 с.

258. Вентцель Е.С. Исследование операций: задачи, принципы, методология / Е.С. Вентцель. – М. : Наука, 1980. – 208 с.

259. Юдин Д.Б. Экстремальные модели в управлении / Д.Б. Юдин, А.Д. Юдин. – М. : Экономика, 1979. – 288 с.

260. Деруссо П. Пространство состояний в теории управления / П. Деруссо, Р. Рой, Ч. Клауз. – М. : Наука, 1970. – 620 с.

261. Данжело Г. Линейные системы с переменными параметрами / Г. Данжело. – М. : Машиностроение, 1974. – 386 с.

262. Браммер К, Зиффлинг Г. Фильтр Калмана-Бьюси / К. Браммер, Г. Зиффлинг. – М. : Наука, 1982. – 200 с.

263. Luenberger D.G. Observers for multivariable system / D.G. Luenberger // IEEE Trans. on Automatic Control. – 1966. – AC-11. – P. 190-197.

264. Zakian V. Critical systems and torelable inputs / V. Zakian // Int.J.Control. – 1989. – 49. – № 4. – P. 1285-1289.

265. Whidbone J.F. Critical Control Systems / J.F. Whidbone, G.P. Liu. – N.Y.: John Wiley & Sons Inc., 1993. – 187 p.

266. Андреев Н.И. Корреляционная теория статистических оптимальных систем / Н.И. Андреев. – М. : Наука, 1966. – 456 с.

267. Whidborne J.F. Robust controller design using H^∞ loop-shaping and the method of inequalities / J.F. Whidborne, J. Postlethwaite, D.W. Lie. – Technical Report 92-33, Leicester-University Engineering Department, Leicester, U.K. – 1992. – 123 p.

268. Brammer K., Siffing G. Stochastische Grundlagen des Kalman-Bucy-Filters / K. Brammer, G. Siffing. – München: Ulderburg, 1975. – 124 p.
269. Ицкович Э.Л. Контроль производства с помощью вычислительных машин / Э.Л. Ицкович. – М. : Энергия, 1975. – 416 с.
270. Петрович М.Л. Регрессионный анализ и его математическое обеспечение на ЕС ЭВМ / М.Л. Петрович. – М. : Финансы и статистика, 1982. – 199 с.
271. Устойчивые статистические методы оценки данных / Под ред. Р. Лонера, Г.Н. Уилкинсона. – М. : Машиностроение, 1984. – 232 с.
272. Смоляк С.А. Устойчивые методы оценивания / С.А. Смоляк, Б.П. Титаренко. – М. : Статистика, 1980. – 206 с.
273. Мудров В.И. Методы обработки измерений / В.И. Мудров, В.Л. Кушко. – М. : Радио и связь, 1983. – 304 с.
274. Алгоритмы и программы восстановления зависимостей / Под ред. В.Н. Вапнина. – М. : Наука, 1984. – 816 с.
275. Мостеллер Ф. Анализ данных и регрессия / Ф. Мостеллер, Дж. Тьюки. – М. : Финансы и статистика, 1982. – вып. 1. – 317 с, вып. 2. – 239 с.
276. Чуев Ю.В. Прогнозирование количественных характеристик процессов / Ю.В. Чуев, Ю.Б. Михайлов, В.И. Кузьмин. – М. : Советское радио, 1975. – 400 с.
277. Brown R.G. Smoothing, Forecasting and Prediction of Discrete Time Series / R.G. Brown. – New Jersey: Prentice Hall, 1963. – 468 p.
278. Тимофеев В.А. Двухступенчатый алгоритм оценивания параметров, основанный на методе инструментальных переменных / В.А. Тимофеев // Радиоэлектроника и информатика. – 1999. – № 4. – С. 39–41.