Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп‘ютерних наук
Кафедра Програмної інженерії

АТЕСТАЦІЙНА РОБОТА (ПРОЕКТ)

Пояснювальна записка

другий (магістерський)

«Дослідження різницевих рівнянь для моделювання та аналізу дискретних динамічних систем»

Виконав: студент 2 курсу, групи ІПЗм-18-3
спеціальності
121 – Інженерія програмного забезпечення
Освітньо-наукова програма
Інженерія програмного забезпечення
Кам’янський І. А.

Керівник: проф. Власенко Л. А.

Допускається до захисту
Зав. кафедри, проф. _____ Дудар З. В.
(підпис)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп‘ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення

Тип програми Освітньо-наукова програма

Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«___»_____ 2020 р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Кам’янському Ігорю Андрійовичу

1. Тема роботи проекту «Дослідження різницевих рівнянь для моделювання та аналізу дискретних динамічних систем» затверджена наказом по університету від «16» березня 2020 р. № ___473___

2. Термін подання студентом роботи (проекту): 10 травня 2020 р.

3. Вихідні дані до роботи: інтелектуальний аналіз даних, метод, дискретне моделювання, динамічна система, різницеві рівняння, алгоритм.

4. Перелік питань, що потрібно опрацювати в роботі: аналіз предметної галузі та постановка задачі, аналіз існуючих методів інтелектуальної обробки даних, дослідження методів аналізу часових рядів за допомогою різницевих рівнянь, висновки.

5. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім’я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз предметної галузі	27 січня 2020 р.	Виконано
2.	Аналіз методів інтелектуальної обробки даних	20 лютого 2020 р.	Виконано
3.	Дослідження методів аналізу та моделювання часових рядів	8 березня 2020 р.	Виконано
4.	Підготовка пояснювальної записки	22 березня 2020 р.	Виконано
5.	Підготовка презентації та доповіді	15 квітня 2020 р.	Виконано
6.	Попередній захист	8 травня 2020 р.	Виконано
7.	Нормоконтроль, рецензування	13 травня 2020 р.	Виконано
8.	Занесення диплома в електронний архів	15 травня 2020 р.	Виконано
9.	Допуск до захисту у зав. кафедри	16 травня 2020 р.	Виконано

Дата видачі завдання 19.03.2020 р.

Студент _____ Кам’янський І. А.
(підпис)

Керівник роботи _____ проф. Власенко Л. А.
(підпис)

РЕФЕРАТ / ABSTRACT

Атестаційна робота магістра містить: 44 с., 12 рис., 12 джерел.

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, МЕТОД, ДИСКРЕТНЕ МОДЕЛЮВАННЯ, ДИНАМІЧНА СИСТЕМА, КЛАСТЕРИЗАЦІЯ АЛГОРИТМ.

Метою роботи є дослідження методів обробки даних за допомогою різницевих лінійних рівнянь в дискретних динамічних системах та аналіз методів моделювання дискретних динамічних систем.

В роботі проведений аналіз предметної галузі, досліджені основні методи, моделі, алгоритми інтелектуального аналізу даних за допомогою різницевих рівнянь, область Data Mining або - інтелектуальний аналіз даних.

DATA MINING, METHOD, DISCRETE MODELING, DYNAMIC SYSTEM, CLUSTERING ALGORITHM

The aim of the work is to study the methods of data processing using difference linear equations in discrete dynamical systems and the analysis of methods for modeling discrete dynamical systems.

The analysis of the subject area is carried out in the work, the basic methods, models, algorithms of intellectual analysis of data by means of difference equations, area Data Mining or - intellectual analysis of data are investigated.

ЗМІСТ

ВСТУП

В процесі розвитку інформаційних технологій, а також систем збору і зберігання даних - баз даних, сховищ даних, і з недавніх пір, хмарних сховищ, виникла проблема аналізу великих обсягів даних, коли аналітик або керівник не в змозі вручну обробити великі масиви даних і прийняти рішення. Зрозуміло, що аналітику необхідно якимось чином представити вихідну інформацію в більш компактному вигляді, з якої може впорається людський мозок за прийнятний час.

Стрімка технологічна еволюція останніх років у сфері інформаційно-комунікаційних технологій дозволила сформувати істотний доробок у частині розвиненою програмно-апаратної інфраструктури, що підтримує накопичення і постійне поповнення архівів даних різної природи і призначення. Тому в останні роки стрімкий розвиток отримала область Data Mining або - інтелектуальний аналіз даних в дискретних динамічних системах.

Data Mining - збірна назва, що використовується для позначення сукупності методів виявлення в даних раніше невідомих, практично корисних і доступних інтерпретації знань, необхідних для прийняття рішень в різних сферах людської діяльності.

Найбільший інтерес до технологій інтелектуальної обробки даних, в першу чергу, проявляють компанії, що працюють в умовах високої конкуренції та мають чітку групу споживачів (роздрібна торгівля, фінанси, зв'язок, маркетинг). Вони використовують будь-яку можливість для підвищення ефективності власного бізнесу через ухвалення більш ефективних управлінських рішень.

Існує безліч різних методів інтелектуального аналізу даних. За типом використовуваного математичного апарату виділяють різні групи методів Data Mining, серед яких опис процесів, що протікає в часі. Для них можна використовувати теорію різницевих рівнянь.

1 АНАЛІЗ ПРОБЛЕМНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальні відомості про інтелектуальний аналіз даних

В наш час розвиток методів запису і зберігання даних привело до стрімкого зростання обсягів інформації, що збирається і аналізується. Обсяги даних настільки значні, що людині просто не під силу проаналізувати їх самостійно, хоча необхідність проведення такого аналізу цілком очевидна, адже в цих даних укладені знання, які можуть бути використані при прийнятті рішень. Для того щоб провести автоматичний аналіз даних, використовується Data Mining.

Метою роботи є вивчення основних методів інтелектуальної обробки даних і, зокрема, методів аналізу часових рядів даних. З'ясувати, як здійснюється моделювання часових рядів за допомогою різницевих рівнянь, які представляють собою складові частини авторегресійних моделей часових рядів. Навчитися будувати математичні моделі часових рядів, що дозволяють адекватно описувати досліджувані процеси за допомогою різницевих рівнянь [12].

Data Mining - це процес виявлення в "сирих" даних раніше невідомих нетривіальних практично корисних і доступних інтерпретації знань, необхідних для прийняття рішень в різних сферах людської діяльності. Data Mining є одним з кроків Knowledge Discovery in Databases.

Knowledge Discovery in Databases - це процес пошуку корисних знань в "сирих" даних. Knowledge Discovery in Databases включає в себе питання: підготовки даних, вибору інформативних ознак, очищення даних, застосування методів Data Mining, пост обробки даних та інтерпретації отриманих результатів. Безумовно, "серцем" всього цього процесу є методи DM, що дозволяють виявляти знання. Цей процес не задає набір методів обробки або придатні для аналізу алгоритми, він визначає послідовність дій, яку необхідно виконати для того, щоб з вихідних даних отримати знання. Даний підхід універсальний і не залежить від предметної області, що є його безперечною гідністю.

Deductor - повнофункціональна платформа для вирішення завдань Knowledge Discovery in Databases, що дозволяє провести всі представлені нижче кроки:

1. Підготовка вихідного набору даних. До складу системи входить Deductor Warehouse - багатовимірне сховище даних, орієнтоване на вирішення завдань консолідації інформації з різнорідних джерел і швидкого вилучення цікавить набору даних. Deductor Warehouse підтримує багатий семантичний шар, що дозволяє кінцевому користувачеві оперувати бізнес термінами для отримання потрібних даних. Крім власного сховища Deductor підтримує роботу і з іншими джерелами: Oracle, DB2, MS SQL, Informix, Sybase, Interbase, DBase, FoxPro, Paradox, MS Access, CSV (текстові файли з роздільниками), ODBC, ADO. Для забезпечення максимальної швидкодії Deductor підтримує прямий доступ до більшості найбільш популярних баз даних.
2. Перед обробка. Deductor містить великий набір механізмів предобробки і очищення даних: заповнення пропусків, редагування аномалій, очищення від шумів, згладжування, фільтрація і безліч інших з можливістю комбінування методів попередньої обробки.
3. Трансформація, нормалізація даних. Deductor включає великий набір механізмів трансформації даних, що дозволяють провести всю підготовчу роботу для подальшого аналізу. Крім цього, система містить широкий спектр механізмів нормалізації для всіх типів даних: числових, рядкових, дата / час і логічних.
4. Data Mining. До складу пакету включені алгоритми, що реалізують популярні і ефективні методи Data Mining: нейронні мережі, дерева рішень, самоорганізуються карти Кохонена, асоціативні правила та інше.
5. Постобробка даних. Результати пост обробки можуть бути відображені за допомогою великого набору механізмів візуалізації: OLAP, таблиці, діаграми, дерева і безліч інших. Для деяких механізмів передбачені спеціалізовані візуалізатори, що забезпечують легкість інтерпретації результатів. Результати можуть бути експортовані для подальшої обробки за допомогою інших

додатків. Це дає можливість ефективно використовувати отримані знання або моделі на інших даних.

Інформація, знайдена в процесі застосування методів Data Mining, повинна бути нетривіальною і раніше невідома, наприклад, середні продажі не є такими. Знання повинні описувати нові зв'язки між властивостями, передбачати значення одних ознак на основі інших і т.д. Знайдені знання повинні бути застосовані і на нових даних з деякою мірою вірогідності. Корисність полягає в тому, що ці знання можуть приносити певну вигоду при їх застосуванні. Знання повинні бути зрозумілі для користувача не в математичному вигляді. Наприклад, найпростіше сприймаються людиною логічні конструкції "якщо ... то ...". Більш того, такі правила можуть бути використані в різних СУБД в якості SQL-запитів. У разі, коли витягнуті знання непрозорі для користувача, повинні існувати методи обробки поста, що дозволяють привести їх до інтерпретованих увазі.

Алгоритми, що використовуються в Data Mining, вимагають великої кількості обчислень. Раніше це було стримуючим фактором широкого практичного застосування Data Mining, проте сьогоднішнє зростання продуктивності сучасних процесорів зняв гостроту цієї проблеми. Тепер за прийнятний час можна провести якісний аналіз сотень тисяч і мільйонів записів.

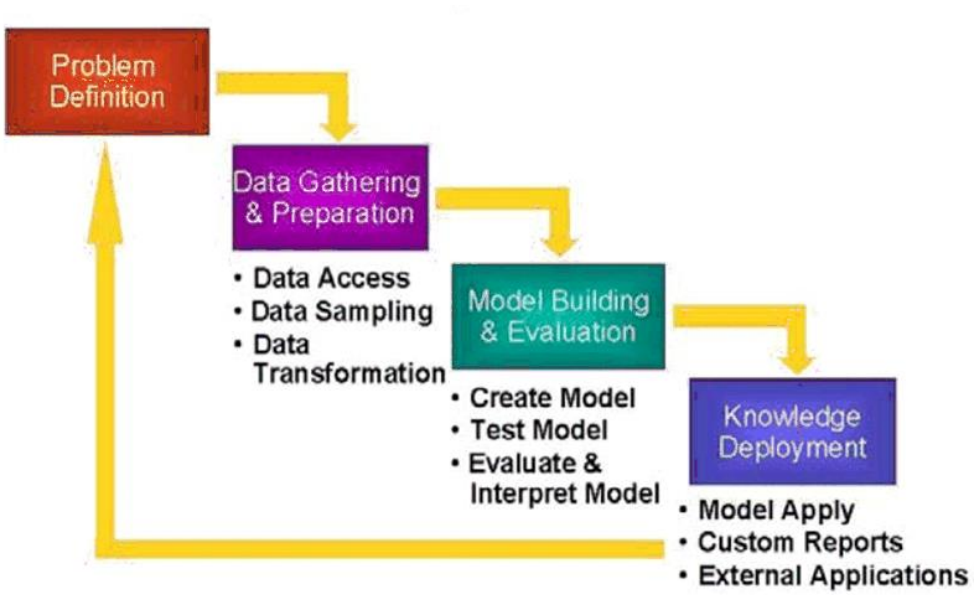

Рис. 1.1 – Процес видобутку даних Data Mining

Завдання, які вирішуються методами Data Mining:

– класифікація - це віднесення об'єктів спостережень або подій до одного з заздалегідь відомих класів;

– регресія, в тому числі завдання прогнозування. Встановлення залежності безперервних вихідних від вхідних змінних;

– кластеризація - це групування об'єктів (спостережень, подій) на основі даних (властивостей), що описують сутність цих об'єктів. Об'єкти усередині кластера повинні бути "схожими" один на одного і відрізнятися від об'єктів, які увійшли в інші кластери. Чим більше схожі об'єкти усередині кластера і чим більше відмінностей між кластерами, тим точніше кластеризація;

– асоціація - виявлення закономірностей між пов'язаними подіями. Прикладом такої закономірності служить правило, яке вказує, що з події X слід подія Y. Такі правила називаються асоціативними. Вперше ця задача була запропонована для знаходження типових шаблонів покупок, що здійснюються в супермаркетах, тому іноді її ще називають аналізом ринкової корзини або market basket analysis;

– послідовні шаблони - встановлення закономірностей між пов'язаними в часі подіями, тобто виявлення залежності, що якщо відбудеться подія X, то через заданий час відбудеться подія Y;

– аналіз відхилень - виявлення найбільш нехарактерних шаблонів.

Проблеми бізнес аналізу формулюються по-іншому, але рішення більшості з них зводиться до тієї чи іншої задачі Data Mining або до їх комбінації. Наприклад, оцінка ризиків - це вирішення завдання регресії або класифікації, сегментація ринку - кластеризація, стимулювання попиту - асоціативні правила. Фактично, завдання Data Mining є елементами, з яких можна зібрати рішення переважної більшості реальних бізнес завдань.

Для вирішення вищеописаних завдань використовуються різні методи і алгоритми Data Mining. З огляду на те, що Data Mining розвивалася і розвивається на стику таких дисциплін, як статистика, теорія інформації, машинне навчання,

теорія баз даних, цілком закономірно, що більшість алгоритмів і методів Data Mining були розроблені на основі різних методів з цих дисциплін. Наприклад, процедура кластеризації k-means була просто запозичена з статистики. Велику популярність отримали такі методи Data Mining: нейронні мережі, дерева рішень, алгоритми кластеризації, в тому числі і масштабовані алгоритми виявлення асоціативних зв'язків між подіями і т.д.

Deductor є аналітичною платформою, в яку включено повний набір інструментів для вирішення завдань Data Mining: лінійна регресія, нейронні мережі з учителем, нейронні мережі без вчителя, дерева рішень, пошук асоціативних правил і безліч інших. Для багатьох механізмів передбачені спеціалізовані візуалізатори, що значно полегшують використання отриманої моделі і інтерпретацію результатів. Сильною стороною платформи є не тільки реалізація сучасних алгоритмів аналізу, але і забезпечення можливості довільним чином комбінувати різні механізми аналізу.

1.2 Кластеризація в Data Mining

Кластеризація - об'єднання в групи схожих об'єктів - є однією з фундаментальних завдань в галузі аналізу даних і Data Mining. Список прикладних областей, де вона застосовується, широкий: сегментація зображень, маркетинг, боротьба з шахрайством, прогнозування, аналіз текстів і багато інших. На сучасному етапі кластеризація часто виступає першим кроком при аналізі даних. Після виділення схожих груп застосовуються інші методи, для кожної групи будується окрема модель.

Завдання кластеризації в тому чи іншому вигляді формулювали в таких наукових напрямках, як статистика, розпізнавання образів, оптимізація, машинне навчання. Звідси розмаїття синонімів поняття кластер - клас, таксон, згущення.

На сьогоднішній момент число методів розбиття груп об'єктів на кластери досить великий - кілька десятків алгоритмів і ще більше їх модифікацій. Однак нас цікавлять алгоритми кластеризації з точки зору їх застосування в Data Mining.

Кластеризація в Data Mining набуває цінність тоді, коли вона виступає одним з етапів аналізу даних, побудови закінченого аналітичного рішення. Аналітику частіше легше виділити групи схожих об'єктів, вивчити їх особливості і побудувати для кожної групи окрему модель, ніж створювати одну загальну модель на всіх даних.

Дуже часто дані, з якими стикається технологія Data Mining, мають такі важливі особливості:

– висока розмірність (тисячі полів) і великий обсяг (сотні тисяч і мільйони записів) таблиць баз даних і сховищ даних (надвеликі бази даних);

– набори даних містять велику кількість числових і категорійних атрибутів.

Всі атрибути, або ознаки об'єктів діляться на числові (numerical) і категорійні (categorical). Числові атрибути - це такі, які можуть бути впорядковані в просторі, відповідно категорійні - яке не можуть бути впорядковані. Наприклад, атрибут "вік" - числовий, а "колір" - категорійний. Приписування атрибутам значень відбувається під час вимірювань обраним типом шкали, а це, взагалі кажучи, являє собою окрему задачу.

Більшість алгоритмів кластеризації припускають порівняння об'єктів між собою на основі певної міри близькості (подібності).

Мірою близькості називається величина, що має межу що зростає зі збільшенням близькості об'єктів. Заходи подібності "винаходяться" за спеціальними правилами, а вибір конкретних заходів залежить від завдання, а також від шкали вимірювань. В якості запобіжного близькості для числових атрибутів дуже часто використовується евклідова відстань (Формула 1.1):

$$d_{pq} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \qquad (1.1)$$

Для категорійних атрибутів поширена міра подібності Чеканівського-Серенсена і Жаккар (|t1∩t2| / |t1∪t2|).

Потреба в обробці великих масивів даних в Data Mining привела до формулювання вимог, яким, по можливості, повинен задовольняти алгоритм кластеризації:

– Мінімально можлива кількість проходів по базі даних;

– Робота в обмеженому обсязі оперативної пам'яті комп'ютера;

– Роботу алгоритму можна перервати зі збереженням проміжних результатів, щоб продовжити обчислення пізніше;

– Алгоритм повинен працювати, коли об'єкти з бази даних можуть вилучатись тільки в режимі односпрямованого курсора (тобто в режимі навігації по записах).

Алгоритм, що задовольняє даним вимогам (особливо другого), будемо називати масштабуємим. Масштабованість - найважливіша властивість алгоритму, залежне від його обчислювальної складності та програмної реалізації. Є і більш ємне визначення. Алгоритм називають масштабуємим, якщо при незмінній місткості оперативної пам'яті зі збільшенням числа записів в базі даних час його роботи зростає лінійно.

Але далеко не завжди потрібно обробляти надвеликі масиви даних. Тому на зорі становлення теорії кластерного аналізу питань масштабованості алгоритмів уваги практично не приділялося. Передбачалося, що всі оброблювані дані будуть уміщатися в оперативній пам'яті, головний акцент завжди робився на поліпшення якості кластеризації. Тому в ідеалі в арсеналі Data Mining повинні бути присутніми як ефективні алгоритми кластеризації мікромассивів, так і масштабовані для обробки надвеликих баз даних.

За способом розбиття на кластери алгоритми бувають двох типів:

– ієрархічні;

– неієрархічні.

Класичні ієрархічні алгоритми працюють тільки з категорійними атрибутами, коли будується повне дерево вкладених кластерів. Тут поширені агломеративні методи побудови ієрархій кластерів - в них проводиться послідовне об'єднання вихідних об'єктів і відповідне зменшення числа кластерів. Ієрархічні алгоритми забезпечують порівняно високу якість кластеризації і не вимагають попереднього завдання кількості кластерів. Більшість з них мають складність $O(n^2)$.

Неієрархічні алгоритми засновані на оптимізації деякої цільової функції, яка визначає оптимальний в певному сенсі розбиття множини об'єктів на кластери. У цій групі популярні алгоритми сімейства k-середніх, які в якості цільової функції використовують суму квадратів зважених відхилень координат об'єктів від центрів шуканих кластерів. Кластери шукаються сферичної або еліпсоїдної форми. У канонічної реалізації мінімізація функції проводиться на основі методу множників Лагранжа і дозволяє знайти тільки найближчий локальний мінімум. Використання методів глобального пошуку значно збільшить обчислювальну складність алгоритму (Рис. 1.2).

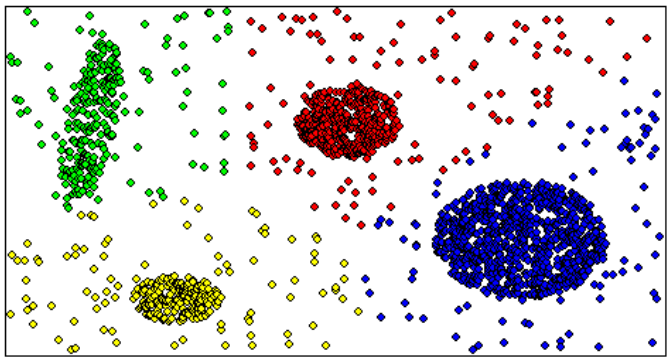

Рис 1.2 – Гістограма двох розбиттів

Серед неієрархічних алгоритмів, які не грунтуються на відстані, слід виділити EM-алгоритм (Expectation-Maximization). У ньому замість центрів кластерів передбачається наявність функції щільності ймовірності для кожного кластеру з відповідним значенням математичного очікування і дисперсією. В суміші розподілів ведеться пошук їх параметрів за принципом максимуму

правдоподібності. Алгоритм EM і є одна з реалізацій такого пошуку. Проблема полягає в тому, що перед стартом алгоритму висувається гіпотеза про вид розподілів, які оцінити в загальній сукупності даних складно (Рис. 1.3).

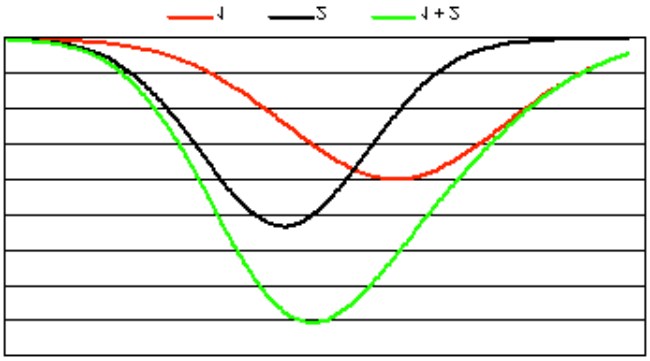

Рисунок 1.3 – Розподіл та його суміш

Ще одна проблема з'являється тоді, коли атрибути об'єкта змішані - одна частина має числовий тип, а інша частина - категорійний. Наприклад, нехай потрібно обчислити відстань між наступними об'єктами з атрибутами (Вік, Стать, Освіта):

{23, чол, вища} (1.2)
{25, жін, середня}.

Перший атрибут є числовим, інші - категорійними. Якщо ми захочемо скористатися класичним ієрархічним алгоритмом з будь-якою мірою подібності, нам доведеться якимось чином зробити дискредитацію атрибута "Вік" (Формула 1.3).

{до 30 р, чол, вища} (1.3)
{до 30 р, жін, середня}

При цьому частину інформації, ми, безумовно, втратимо. Якщо ж ми будемо визначати відстань в евклідовому просторі, то виникнуть питання з категорійними атрибутами. Зрозуміло, що відстань між "Стать чоловік" і "Стать жінка" дорівнює 0, тому що значення цієї ознаки знаходяться в шкалі найменувань. А атрибут "Освіта" можна виміряти як в шкалі найменувань, так і в шкалі порядку, присвоївши кожному значенню певні бали. Крім того, при використанні алгоритму k-середніх і йому подібних виникають труднощі з розумінням центрів кластерів у категорійних атрибутів, апріорним завданням кількості кластерів.

Алгоритм оптимізації цільової функції в неієрархічних алгоритмах, заснованих на відстанях, носить ітеративний характер, і на кожній ітерації потрібно розраховувати матрицю відстаней між об'єктами. При великому числі об'єктів це неефективно і потребує серйозних обчислювальних ресурсів. Обчислювальна складність 1-ї ітерації алгоритму k-means оцінюється як O(kmn), де k, m, n - кількість кластерів, атрибутів і об'єктів відповідно. Але ітерацій може бути дуже багато! Доведеться робити багато проходів по набору даних.

Має масу недоліків в k-means сам підхід з ідеєю пошуку кластерів сферичної або еліпсоїдної форми. Підхід добре працює, коли дані в просторі утворюють компактні згустки, що добре відрізняються одне від одного. А якщо дані мають вкладену форму, то жоден з алгоритмів сімейства k-means ніколи не впорається з таким завданням. Також алгоритм погано працює в разі, коли один кластер значно більше за інших, і вони знаходяться близько один від одного - виникає ефект "розщеплення" великого кластера (рис. 1.4).

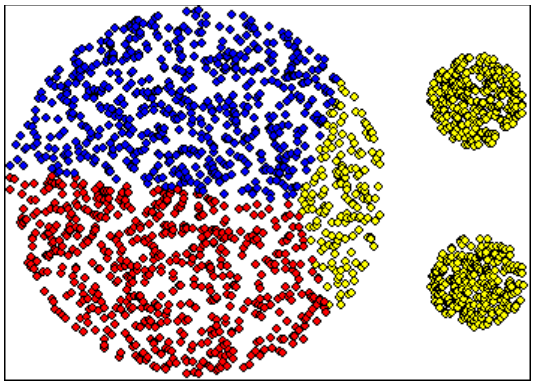

Рисунок 1.4 – Ефект розщеплення великого кластера

Втім, дослідження в галузі вдосконалення алгоритмів кластеризації йдуть постійно. Розроблено цікаві розширення алгоритму k-means для роботи з категорійними атрибутами (k-modes) і змішаними атрибутами (k-prototypes). Наприклад, в k-prototypes розрахунок відстаней між об'єктами здійснюється по-різному в залежності від типу атрибута.

На ринку масштабованих алгоритмів кластеризації боротьба йде за зниження кожного додаткового проходу по набору даних під час роботи будь-якого алгоритму. Розроблено масштабовані аналоги k-means та EM, масштабовані агломеративні методи. В цілому на даний момент сучасні алгоритми вимагають від двох до десяти ітерацій сканувань бази даних до отримання фінальної кластеризації.

Отримання масштабованих алгоритмів засноване на ідеї відмови від локальної функції оптимізації. Парне порівняння об'єктів між собою в алгоритмі k-means є не що інше, як локальна оптимізація, тому що на кожній ітерації необхідно розраховувати відстань від центру кластера до кожного об'єкта. Це веде до великих обчислювальних витрат. При завданні глобальної функції оптимізації додавання нової точки в кластер не вимагає великих обчислень: воно розраховується на основі старого значення, нового об'єкта і так званих кластерних характеристик. Конкретні кластерні характеристики залежать від того чи іншого алгоритму. Так з'явилися алгоритми BIRCH, LargeItem, CLOPE і багато інших.

Таким чином, не існує єдиного універсального алгоритму кластеризації. При використанні будь-якого алгоритму важливо розуміти його достоїнства і недоліки, враховувати природу даних, з якими він краще працює і здатність до масштабованості.

1.3 Machine learning в Data Mining

Data mining і machine learning в основному зосереджені на тому, щоб допомагати компаніям розробляти інструменти прийняття рішень без особливої участі людини. Більш того, прийняті рішення можуть стати основою для дій в тому чи іншому напрямку. Програми спочатку вивчають ваші звички і розробляють алгоритми прийняття рішень, які можуть передбачати ваші дії, направляти до потенційно цікавим для вас сферам розвитку або корисним лідам.

Сотні проблем вирішуються за секунди завдяки можливості провести глибокий всебічний аналіз даних, які зазвичай зберігаються хаотично і не структуровано.

Технологія data mining допомагає у всіх питаннях, пов'язаних з пошуком даних. Будь то інформація про людей, концепції, поведінку або про пристрої, якими користуються споживачі для взаємодії з брендом. При цьому у відносно короткі терміни ви можете проаналізувати терабайти даних.

Найчастіше для зручності компанії використовують сховища даних. Таким чином можна в будь-який момент провести потрібний аналіз і отримати робочі інсайти для прийняття рішень.

За допомогою інструментів data mining ви можете провести глибокий пошук потрібних даних і відшукати непомітні на перший погляд патерни і зв'язки. Те, з чим людський мозок просто фізично не може впоратися самотужки.

А саме вони важливі в аналізі закономірностей поведінки споживача і для передбачення можливого фідбеку.

З технологією machine learning справи йдуть трохи складніше. По суті, це система програм на основі штучного інтелекту, створена для розуміння роботами природи ходу людської думки. У підсумку, вчені та інженери сподіваються отримати механізм для прийняття рішень без участі людини.

На даний момент силами штучного інтелекту можна передбачити реакцію споживача на ваші дії. Все, що вам потрібно це база даних, яку технологія використовує як джерело знань про минулі звички цільової аудиторії.

Також зараз активно розвивається нова технологія - deep learning. Глибоке навчання намагається повторити роботу мозку людини. Зрештою, вчені хочуть дійти до тієї точки, коли в базах даних потреби і зовсім не буде. Весь процес передбачення поведінки буде автоматизовано.

Основні відмінності між технологіями:

– функціонал data mining строго обмежений збором інформації з різних ресурсів. Сама технологія не приймає рішення і не здатна робити якісь дії без участі людини. Основна мета - пошук корисних способів застосування даних, які були знайдені.

– machine learning працює з масивами даних, які технологія data mining сформувала. За допомогою заздалегідь змодельованих алгоритмів дій, технологія II використовує дані для прийняття рішень і подальших дій. Без постійного бекапу актуальної інформації ця технологія не існує.

У підсумку ми отримуємо свою екосистему прийняття обґрунтованих рішень. Обидві технології доповнюють один одного, використовувати їх поодинці означає обмежувати їх потенціал.

Кейси використання data mining:

– ретейл використовує технологію для аналізу цільової аудиторії. Потенційних клієнтів можна знайти орієнтуючись на основні характеристики, які об'єднують вже існуючу базу користувачів. Також data mining допомагає проаналізувати ефективність послуги або продукту бренду і прийняти рішення щодо необхідних доробок. Зорієнтувати потенційно зацікавлених споживачів також можливо з цієї технології;

– E-commerce процвітає саме завдяки глибокому аналізу попередньої історії активності користувача;

Кейси використання machine learning:

– business intelligence використовує технологію для вирішення різних питань. Від прийняття рішень щодо різних транзакцій, вибору потенційно сприятливих сфер розвитку для бізнесу до формування висновків щодо результатів продажів. Технологія допомагає постійно стежити за "станом здоров'я" вашої компанії і пропонує альтернативи в розвитку і пошуку нових ніш;

– управління спамом на пошті засноване на ШІ. Деякі програми можуть навіть видалити листи з пошти користувача, що мають віруси в пересланих файлах, не допускаючи можливості зараження вашого ПК;

– онлайн обслуговування клієнтів за допомогою чат ботів також засновано на ШІ. Для скорочення часу очікування на телефонні дзвінки відповідають боти.

1.4 Дерева рішень

Дерева рішень є одним з найбільш ефективних інструментів інтелектуального аналізу даних і самий корінь аналітики, які дозволяють вирішувати завдання класифікації і регресії.

Вони являють собою ієрархічні деревоподібні структури, що складаються з вирішальних правил виду «Якщо ..., то ...». Правила автоматично генеруються в процесі навчання на навчальній множині і, оскільки вони формулюються практично на природній мові, дерева рішень як аналітичні моделі більш вербалізуємі і інтерпретовані, ніж, скажімо, нейронні мережі.

Оскільки правила в деревах рішень виходять шляхом узагальнення безлічі окремих спостережень, що описують предметну область, то за аналогією з відповідним методом логічного висновку їх називають індуктивними правилами, а сам процес навчання - індукцією дерев рішень.

Основні ідеї, що послужили поштовхом до появи і розвитку дерев рішень, були закладені в 1950-х роках в області досліджень моделювання людської поведінки за допомогою комп'ютерних систем.

Основна сфера застосування дерев рішень - підтримка процесів прийняття управлінських рішень, що використовується в статистиці, аналізі даних і машинному навчанні. Завдання, які розв'язуються за допомогою даного апарату, є:

– класифікація - віднесення об'єктів до одного з заздалегідь відомих класів. Цільова змінна повинна мати дискретні значення.

– регресія (чисельне пророкування) - прогноз числового значення незалежної змінної для заданого вхідного вектора.

– опис об'єктів - набір правил в дереві рішень дозволяє компактно описувати об'єкти. Тому замість складних структур, що описують об'єкти, можна зберігати дерева рішень.

Переваги алгоритму дерев рішень:

– швидкий процес навчання;

– генерація правил в областях, де експерту важко формалізувати свої знання;

– витяг правил на природній мові;

– інтуїтивно зрозуміла класифікаційна модель;

– висока точність передбачення, порівнянна з іншими методами аналізу даних (статистика, нейронні мережі);

– побудова непараметричних моделей.

Недоліки алгоритму дерев рішень:

– проблема отримання оптимального дерева рішень є NP-повною з точки зору деяких аспектів оптимальності навіть для простих завдань;

– в процесі побудови дерева рішень можуть створюватися занадто складні конструкції, які недостатньо повно представляють дані;

– існують концепти, які складно зрозуміти з моделі, так як модель описує їх складним шляхом;

– для даних, які включають категоріальні змінні з великим набором рівнів більша інформаційна вага присвоюється тим атрибутам, які мають більшу кількість рівнів.

1.5 Алгоритми виявлення асоціативних зв’язків

Останнім часом неухильно зростає інтерес до методів 'виявлення знань в базах даних'. Обсяги сучасних баз даних, які дуже значні, викликали стійкий попит на нові масштабовані алгоритми аналізу даних. Одним з популярних методів виявлення знань стали алгоритми пошуку асоціативних зв’язків.

Асоціативні правила дозволяють знаходити закономірності між пов'язаними подіями. Прикладом такого правила, є твердження, що покупець, що придбає 'Хліб', придбає і 'Молоко' з ймовірністю 72%. Перший алгоритм пошуку асоціативних правил, що називався AIS був розроблений в 1993 році співробітниками дослідницького центру IBM Almaden. З цієї піонерської роботи зріс інтерес до асоціативних правил; на середину 90-х років минулого століття припав пік дослідних робіт в цій області, і з тих пір кожен рік з'являлося по кілька алгоритмів.

Завдання знаходження асоціативних правил розбивається на дві підзадачі:

– знаходження всіх наборів елементів, які задовольняють порогу minsupport. Такі набори елементів називаються часто зустрічаються;

– якщо підтримка має велике значення, то алгоритми знаходитимуть правила, добре відомі аналітикам або настільки очевидні, що немає ніякого сенсу проводити такий аналіз. З іншого боку, низьке значення підтримки веде до генерації величезної кількості правил, що, звичайно, вимагає істотних обчислювальних ресурсів. Тим не менше, більшість цікавих правил знаходиться саме при низькому значенні порогу підтримки. Хоча занадто низьке значення підтримки веде до генерації статистично необґрунтованих правил.

Пошук асоціативних правил зовсім не тривіальне завдання, як може здатися на перший погляд. Одна з проблем - алгоритмічна складність при знаходженні часто зустрічаючих наборів елементів, тому що з ростом числа елементів експоненціально зростає число потенційних наборів елементів.

Для реалізації правил виявлення асоціативних зв'язків існують такі алгоритми:

– Apriori - масштабований алгоритм пошуку асоціативних правил. Сучасні бази даних мають дуже великі розміри, що досягають Гігабайтів і Терабайтів, і тенденцію до подальшого збільшення. Тому, для знаходження асоціативних правил потрібні ефективні масштабовані алгоритми, що дозволяють вирішити задачу за прийнятний час. Одним з таких алгоритмів є алгоритм Apriori;

– ECLAT - розшифровується як кластеризація класів еквівалентності і обхід решітки знизу вгору. Це один з популярних методів об'єднання правил майнінгу. Це більш ефективна і масштабована версія алгоритму Apriori. Хоча алгоритм Apriori працює в горизонтальному сенсі, імітуючи пошук графіка по ширині, алгоритм ECLAT працює по вертикалі, як пошук графіка по глибині. Цей вертикальний підхід алгоритму ECLAT робить його швидшим алгоритмом, ніж алгоритм Apriori;

– Frequent Pattern-Growth - в основі методу лежить перед обробка бази транзакцій, в процесі якої ця база даних перетворюється в компактну деревоподібну структуру, frequent часте візерункове дерево-дерево популярних предметних наборів (звідки і назва алгоритму).

2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ ДАНИХ

2.1 Загальний аналіз методів інтелектуальної обробки даних в Data Mining

Інтелектуальний аналіз даних – одна з найактуальніших тем в сучасному світі. Я більш ніж впевнений в цьому, так-як Бізнес-аналіз – тільки мала частина сфери застосування цього найпотужнішого інструменту.

Наприклад існує хрестоматійна історія про те, як команда Google допомогла передбачити географію поширення грипу AH1N1 – за допомогою аналізу десятків терабайтів даних, отриманих від користувачів пошуковика. Прекрасно розуміючи перспективи досліджень в цій сфері, над розробкою і вдосконаленням методів інтелектуального аналізу даних сьогодні працюють цілі інститути по всьому світу.

В наш час розвиток методів запису і зберігання даних привело до стрімкого зростання обсягів інформації, що збирається і аналізується. Обсяги даних настільки значні, що людині просто не під силу проаналізувати їх самостійно, хоча необхідність проведення такого аналізу цілком очевидна, адже в цих даних укладені знання, які можуть бути використані при прийнятті рішень. Для того щоб провести автоматичний аналіз даних, використовується Data Mining.

З огляду на те, що Data Mining розвивалася і розвивається на стику таких дисциплін, як статистика, теорія інформації, машинне навчання, теорія баз даних, цілком закономірно, що більшість алгоритмів і методів Data Mining були розроблені на основі різних методів з цих дисциплін. Наприклад, процедура кластеризації k-means була просто запозичена з статистики. Велику популярність отримали такі методи Data Mining: нейронні мережі, дерева рішень, алгоритми кластеризації, в тому числі і масштабовані алгоритми виявлення асоціативних зв'язків між подіями. І кожен з цих алгоритмів здобув великих успіхів у певних сферах. Наприклад порівняння методів виявлення асоціативних зв’язків (рис. 2.1).

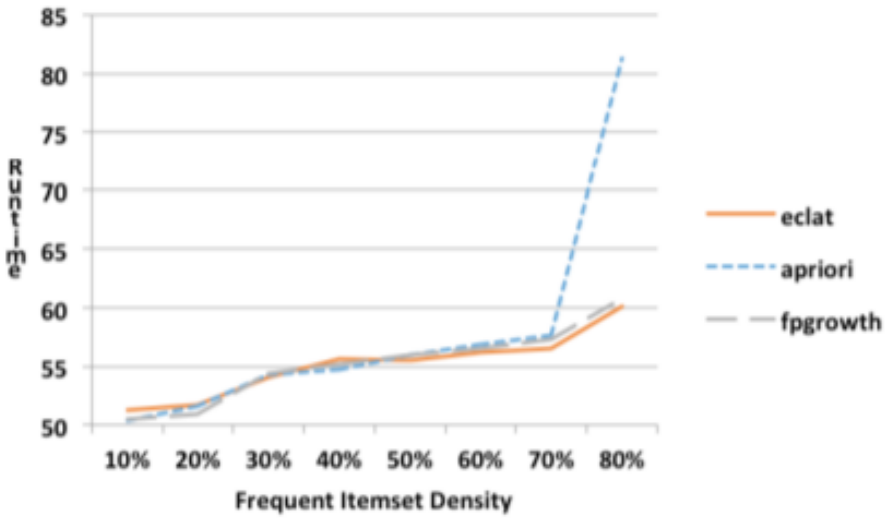

Рисунок 2.1 – Порівнення алгоритмів виявлення асоціативних зв’язків ECLAT, Apriori і FP-Growth.

Як видно на Рис. 2.1 алгоритм Apriori значно програє двом своїм собратам при збільшенні runtime-значення, але з іншої сторони він виграє в роботі з неструктурованими даними.

2.2 Кластеризація та Евклідова відстань

Кластеризація - це групування об'єктів на основі даних, що описують сутність цих об'єктів. Об'єкти усередині кластера повинні бути "схожими" один на одного і відрізнятися від об'єктів, які увійшли в інші кластери. Чим більше схожі об'єкти усередині кластера і чим більше відмінностей між кластерами, тим точніше кластеризація.

Кластерний аналіз - це сімейство алгоритмів, розроблених для формування груп таким чином, щоб члени групи були найбільш схожими один на одного і не схожими на елементи, що не виходять в групу. Кластер і група-це синоніми в світі кластерного аналізу.

Список прикладних областей, де вона застосовується, широкий: сегментація зображень, маркетинг, боротьба з шахрайством, прогнозування, аналіз текстів і багато інших. На сучасному етапі кластеризація часто виступає першим кроком при аналізі даних. Після виділення схожих груп застосовуються інші методи, для кожної групи будується окрема модель.

Кластеризація в Data Mining набуває цінність тоді, коли вона виступає одним з етапів аналізу даних, побудови закінченого аналітичного рішення. Аналітику частіше легше виділити групи схожих об'єктів, вивчити їх особливості і побудувати для кожної групи окрему модель, ніж створювати одну загальну модель на всіх даних. Таким прийомом постійно користуються в маркетингу, виділяючи групи клієнтів, покупців, товарів і розробляючи для кожної з них окрему стратегію.

Дуже часто дані, з якими стикається технологія Data Mining, мають такі важливі особливості:

- висока розмірність і великий обсяг таблиць баз даних і сховищ даних
- набори даних містять велику кількість числових і категорійних атрибутів.

Всі атрибути, або ознаки об'єктів діляться на числові і категорійні. Числові атрибути - це такі, які можуть бути впорядковані в просторі, відповідно категорійні - які не можуть бути впорядковані. Наприклад, атрибут "вік" - числовий, а "колір" - категорійний. Приписування атрибутам значень відбувається під час вимірювань обраним типом шкали, а це, взагалі кажучи, являє собою окрему задачу.

Застосування кластерного аналізу в загальному вигляді зводиться до наступних етапів:

- відбір вибірки об'єктів для кластеризації;
- визначення безлічі змінних, за якими будуть оцінюватися об'єкти у вибірці. При необхідності – нормалізація значень змінних;
- обчислення значень міри подібності між об'єктами;
- застосування методу кластерного аналізу для створення груп подібних об'єктів (кластерів);
- представлення результатів аналізу.

Процедура кластеризації-залежить від міри подібності або несхожості. Такі заходи виражаються у вигляді функцій відстаней, виражених у вигляді тієї чи іншої функції.

Однією з основних заходів подібності або відмінності може бути використано евклідова відстань. А саме, кожен об'єкт характеризується набором параметрів. Можна розрахувати відстані між об'єктами в багатовимірному просторі за допомогою теореми Піфагора, тільки з урахуванням поправки на багатовимірність простору. P=(p1, p2,....pn), Q=(q1, q2,....qn) що обчислюється за формулою (2.1).

$$d_{pq} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{2.1}$$

Наприклад, відстань Евкліда між двома точками a і b в 3-мірному просторі (XYZ) (Рис. 2.2) розраховується за формулою (2.2).

$$d_{ab} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2} \tag{2.2}$$

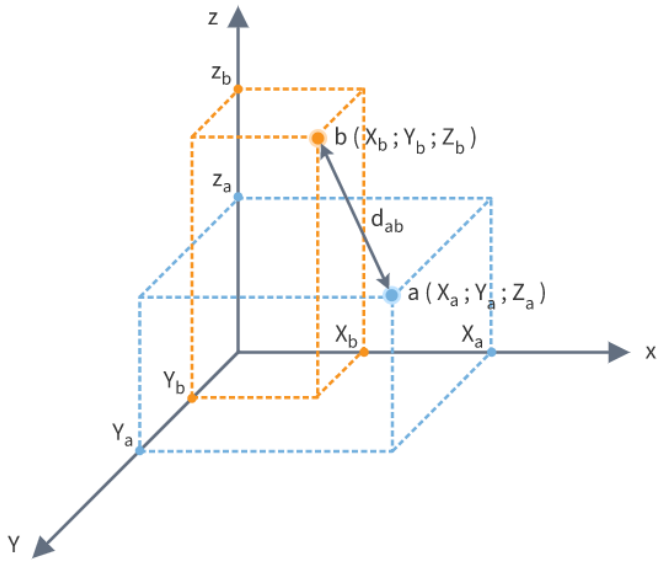

Рисунок 2.2 – Евклідова відстань між двома точками a і b в 3-мірному просторі

Евклідова відстань є найбільш зрозумілою і інтерпретованою мірою відмінності або близькості об'єктів, представлених векторами ознак в багатовимірному просторі, відображаючи інтуїтивні властивості відстані між точками. Тому вона широко використовується в аналізі даних в якості критерію для об'єднання спостережень в класи і кластери, оцінки помилок в прогнозній аналітиці, а також методах візуалізації, наприклад картах Кохонена.

Карта Кохонена - це різновид самоорганізованої карти, яка дозволяє не тільки проводити кластеризацію об'єктів, а й виконувати багатовимірну візуалізацію її результатів.

Відмінність самоорганізованої карти від звичайної мережі Кохонена полягає в кількості вихідних нейронів. В мережі Кохонена воно повинно відповідати кількості кластерів, а в карті – кількості сегментів, з якого вона повинна складатися. Тобто розміру карти. Чим більше число сегментів в карті, тим детальніше вона представляє розподіл об'єктів в просторі ознак.

Число вхідних нейронів карти, як і мережі Кохонена, має дорівнювати числу ознак, за якими проводиться кластеризація (Рис. 2.3).

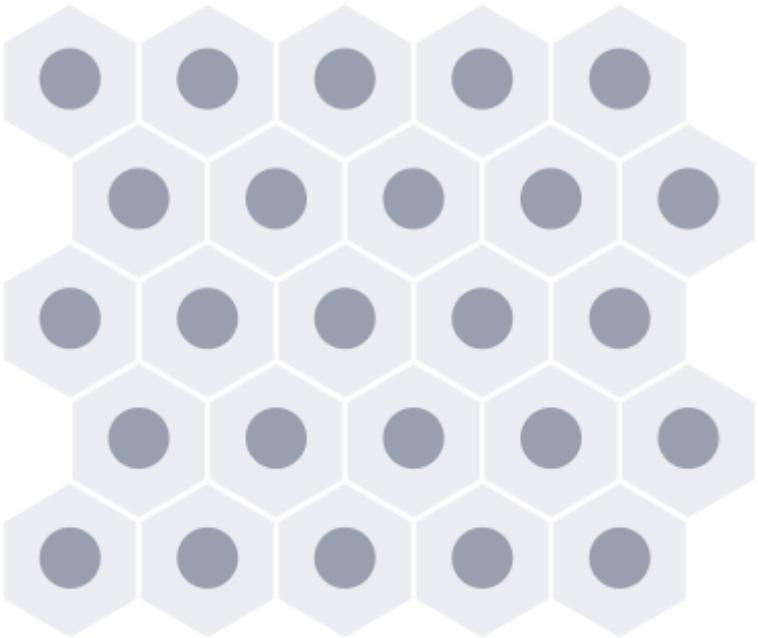

Рисунок 2.3 – Приклад карти розміром 5x5, що містить 25 вихідних нейронів

Зменшивши число сегментів карти до числа кластерів, ми повернемося до звичайної мережі Кохонена.

Карта Кохонена складається з сегментів прямокутної або шестикутної форми, званих осередками. Кожна з них пов'язана з певним вихідним нейроном карти і являє собою свого роду його «сферу впливу». Розподіл векторів ваг нейронів карти виходить так само, як і в мережі Кохонена, тобто на основі конкурентного навчання.

Об'єкти, вектори ознак яких виявляються ближче до вектора ваг даного нейрона карти, потрапляють в осередок, пов'язану з цим нейроном. Тоді розподіл об'єктів на карті в цілому відповідає розподілу векторів ваг нейронів в просторі ознак. Отже, якщо об'єкти на карті розташовані близько один до одного, тобто потрапили в одну клітинку або хоча б в сусідні, то і вектори ознак цих об'єктів близькі. І навпаки, якщо об'єкти потрапили в осередки, розташовані на карті далеко один від одного, то і вектори їх ознак розрізняються сильно.

Також існує існує метод, що дозволяє придати вагу для більш віддалених об’єктів один від одного, під назвою Квадрат Евклідової відстані, який розраховується за наступною формулою (2.3).

$$d_{pq} = \sum_{i=1}^{n} (p_i - q_i)^2 \qquad (2.3)$$

Але на цьому можливості карт Кохонена не закінчуються. Вони дозволяють також представити отриману інформацію в простій і наочній формі шляхом нанесення розмальовки. Для цього ми розфарбовуємо вузли отриманої карти кольорами, що відповідають зацікавленими для нас ознаками об'єктів. Тобто області з осередками, близькими за кольором, містять об'єкти, схожі за ознакою, що відповідає проекції карти (Рис. 2.4).

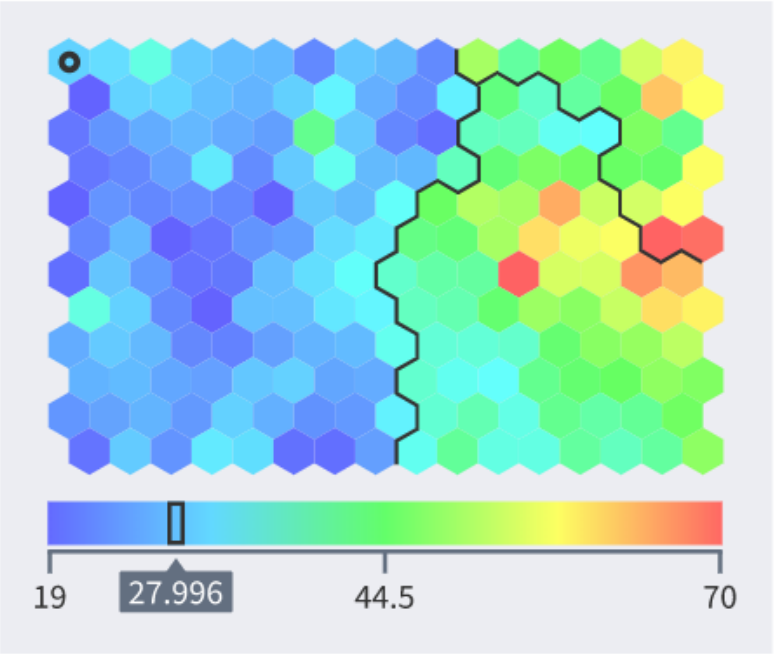

Рисунок 2.4 – Приклад можливого варіанту візуалізації самоорганізованої карти

При всьому цьому описана технологія є універсальним методом аналізу. З її допомогою можна аналізувати різні стратегії діяльності, проводити аналіз результатів маркетингових досліджень, перевіряти кредитоспроможність клієнтів та інше.

Таким чином, маючи перед собою карту і знаючи інформацію про деяку з частини досліджуваних об'єктів, ми можемо досить достовірно судити про об'єкти, з якими ми мало знайомі.

Недоліки, властиві кластерному аналізу:

– склад і кількість кластерів залежать від вибраних критеріїв розбиття;

– при зведенні вихідного масиву даних до більш компактного вигляду можуть виникати певні спотворення, а також можуть губитися індивідуальні риси окремих об'єктів за рахунок заміни їх характеристиками узагальнених значень параметрів кластера;

– при проведенні класифікації об'єктів ігнорується дуже часто можливість відсутності в розглянутій сукупності будь-яких кластерів.

2.2 Алгоритм K-means

Кластеризація в Data Mining набуває цінності тоді, коли вона виступає одним з етапів аналізу даних, побудови закінченого аналітичного рішення. Аналітику часто легше виділити групи схожих об'єктів, вивчити їх особливості і побудувати для кожної групи окрему модель, ніж створювати одну загальну модель для всіх даних. Таким прийомом постійно користуються в маркетингу, виділяючи групи клієнтів, покупців, товарів і розробляючи для кожної з них окрему стратегію.

Найбільш поширений серед неієрархічних методів є алгоритм k-середніх, також званий швидким кластерним аналізом. На відміну від ієрархічних методів, які не вимагають попередніх припущень щодо числа кластерів, для цього методу необхідно мати гіпотезу про найбільш ймовірну кількість кластерів.

K-means найбільш простий, але в той же час досить неточний метод кластеризації в класичній реалізації. Він розбиває безліч елементів векторного простору на заздалегідь відоме число кластерів k. Дія алгоритму така, що він прагне мінімізувати середньоквадратичне відхилення на точках кожного кластера. Основна ідея полягає в тому, що на кожній ітерації перевичислюється центр мас для кожного кластера, отриманого на попередньому кроці, потім вектори розбиваються на кластери знову відповідно до того, який з нових центрів виявився ближче за обраною метрикою. Алгоритм завершується, коли на якійсь ітерації не відбувається зміни кластерів. Алгоритм k-means розбиває набір X на K наборів S1, S2,..., Sk, таким чином, щоб мінімізувати суму квадратів відстаней від кожної точки кластера до його центру (центр мас кластера). Де S={S1, S2,...,Sk}, μ_i-центри кластерів, i=1,..., k, $\rho(x,\mu^i)$ - функція відстані між x і μ_i. Тоді дія алгоритму k-means рівносильно пошуку. В математичному описанні k-means, алгоритм має такий вид (Формула 2.4).

$$\arg\min_S \sum_{i=1}^{k} \sum_{\mathbf{x}\in S_i} \rho(\mathbf{x}, \mu_i)^2 \qquad (2.4)$$

Кластеризація здійснюється за наступним алгоритмом:

а) Вибирається число кластерів . Кожному кластеру відповідає один центр. Вибір початкових центрів може здійснюватися одним з таких способів:

1. вибір спостережень з умови максимізації відстані між ними;
2. випадковий вибір спостережень;
3. вибір перших спостережень.

б) Первісний розподіл об'єктів по кластерах. Кожен об'єкт приєднується до того кластеру, відстань до якого є найменшою.

в) Ітеративний процес. Обчислюються центри кластерів. Об'єкти знову перерозподіляються. Процес обчислення центрів і перерозподілу триває до тих пір, поки не буде виконано одну з умов зупинки:

1. кластерні центри стабілізувалися, тобто всі спостереження належать кластеру, якому належали до поточної ітерації;
2. число ітерацій дорівнює максимальному можливому заданому числу ітерацій.

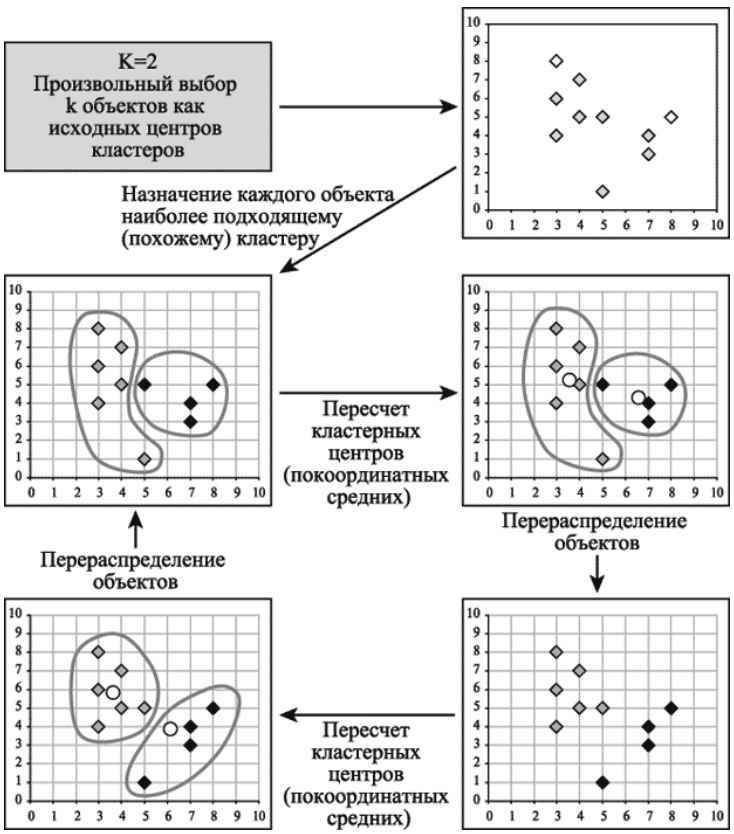

Рисунок 2.5 – приклад роботи алгоритму k-means, де k=2

Після отримання результатів кластерного аналізу методом k-середніх слід перевірити правильність кластеризації (тобто оцінити, наскільки кластери відрізняються один від одного). Для цього розраховуються середні значення для кожного кластера (Формула 2.5), де c_k - центр мас нечіткого кластера k. При хорошій кластеризації повинні бути отримані сильно відрізняються середні для всіх вимірювань або хоча б більшої їх частини.

$$c_k = \sum_{i=1}^{\min s} S_k x_i \tag{2.5}$$

Переваги алгоритму-середніх:

- простота використання;
- швидкість використання;
- зрозумілість і прозорість алгоритму.

Недоліки алгоритму k-means:

- алгоритм занадто чутливий до викидів, які можуть спотворювати середнє.
- алгоритм може повільно працювати на великих базах даних. Можливим вирішенням даної проблеми є використання вибірки даних.

Існують методи кластеризації, які можна розглядати як такі, що походять від k-means. Наприклад, в методі k-medoids для обчислення центроїдів використовується не середнє значення, а медіана, що робить алгоритм більш стійким до аномальних значень в даних.

Алгоритм G-means будує кластери, розподіл даних в яких прагне до нормального і знімає невизначеність вибору початкових кластерів. Алгоритм С-середніх використовує елементи нечіткої логіки, враховуючи при обчисленні центроїдів не тільки відстані, але і ступінь приналежності спостереження до безлічі об'єктів в кластері. Також відомий алгоритм Ллойда, який в якості початкового розбиття використовує не безлічі векторів, а області векторного простору.

3 ДОСЛІДЖЕННЯ МЕТОДІВ АНАЛІЗУ ЧАСОВИХ РЯДІВ

3.1 Загальний аналіз методів моделювання часових рядів

В останні роки стрімкий розвиток отримав інтелектуальний аналіз даних. Найбільший інтерес до технологій інтелектуальної обробки даних, в першу чергу, проявляють компанії, що працюють в умовах високої конкуренції та мають чітку групу споживачів, таких як роздрібна торгівля, фінанси, зв'язок та маркетинг. Вони використовують будь-яку можливість для підвищення ефективності власного бізнесу через ухвалення більш ефективних управлінських рішень.

Існує безліч різних методів інтелектуального аналізу даних. Які з них застосувати для аналізу своїх даних, а які можна використовувати в поєднанні з уже наявним програмним забезпеченням та інфраструктурою. Наприклад, аналіз часових рядів і, зокрема, динамічні моделі часових рядів.

Метою роботи є вивчення основних методів інтелектуальної обробки даних і, зокрема, методів аналізу часових рядів даних. З'ясувати, як здійснюється моделювання часових рядів за допомогою різницевих рівнянь, які представляють собою складові частини авторегресійних моделей часових рядів. Навчитися будувати математичні моделі часових рядів, що дозволяють адекватно описувати досліджувані процеси за допомогою різницевих рівнянь.

Моделювання часових рядів і їх аналіз необхідні фахівцям, діяльність яких пов'язана з комп'ютерною обробкою даних - інженерам, аналітикам і т.д.

Серед різних математичних моделей, що застосовуються для описання динамічних систем, важливе місце займають різницеві рівняння, причому їх роль постійно зростає. Вони широко використовуються в науці і техніці при описі самих різних процесів і систем - електричних, механічних, біологічних, демографічних, економічних та ін.

В якості прикладів можна назвати аналіз ланцюгових схем в теорії ланцюгів [12], моделі довгих ліній в електротехніці, методи чисельного інтегрування в обчислювальної математики, методи сіток і кінцевих елементів в математичній

фізиці. До різницевих рівнянь призводять багато екологічних завдань і моделі популяційної динаміки (Формула 3.1), економічні завдання, а також демографічні моделі.

$$x_{n+1} = \alpha x_n(1 - x_n) \tag{3.1}$$

Різницеве рівняння – це рівняння, що зв'язує значення деякої невідомої функції в будь-якій точці з її значенням в одній або декількох точках, віддалених від даної на певний інтервал. Застосовується для опису дискретних динамічних систем.

Часовий ряд - це послідовність значень, що описують що протікає в часі процес. Значення процесу вимірюються в послідовні моменти часу. Якщо час протікає безперервно, то часовий ряд - безперервний. Якщо вимірювання процесу здійснюються в дискретні моменти часу, то ряд є дискретним. Зазвичай вимірювання проводяться через рівні проміжки. Тимчасові ряди, як правило, виникають в результаті вимірювання деякого показника. Це можуть бути як показники або характеристики технічних систем, так і показники природних, соціальних, економічних та інших систем (наприклад, погодні дані). Графічно ряд зображується в декартовій системі координат, де вісь абсцис - час, вісь ординат - значення членів ряду.

Аналіз часових рядів - це сукупність математичних методів, призначених для виявлення структури часових рядів і для їх прогнозування. Для аналізу часового ряду потрібно визначити функціональну залежність, яка відображатиме зв'язок між минулими і майбутніми значеннями цього ряду, тобто побудувати математичну модель. В якості таких моделей використовують, наприклад, регресивні моделі.

В основу авторегресійних моделей закладено припущення про те, що значення процесу X (t) лінійно залежить від деякої кількості попередніх значень того ж процесу X (t-1), ..., X (t-p) - Авторегрессійний процес порядку p. Модельне припущення полягає в наступному (Формула 3.2) де α0, α1, ... αp - параметри моделі, $\alpha p \neq 0$ [12].

$$X(t) = \alpha_0 + \alpha_1 X(t-1) + \cdots + \alpha_p X(t-p) \quad (3.2)$$

У разі дискретного часу розглянемо лінійне однорідне різницеве рівняння p-го порядку з постійними коефіцієнтами (Формула 3.3).

$$x_{n+p} = \alpha_0 + \alpha_1 x_{n+p-1} + \alpha_2 x_{n+p-2} + \cdots + \alpha_p x_{n+p}\,, \quad n = 0,1,2,\ldots \quad (3.4)$$

Багатовимірні тимчасові ряди моделюються за допомогою систем різницевих рівнянь [12]. Наприклад, неоднорідна лінійна система де b-заданий ненульовий числовий m-мірний вектор, A - задана матриця, має вигляд (Формула 3.4).

$$\vec{x}_{n+1} = A\vec{x}_n + b, \;\; n = 0,1,2,\ldots, \quad (3.5)$$

Однорідна система має вид (Формула 3.6).

$$\vec{x}_{n+1} = A\vec{x}_n, \;\; n = 0,1,2,\ldots, \quad (3.6)$$

3.2 Динамічна павутинообразна модель

Павутинообразна модель - мікроекономічна модель, механізм якої при досконалої конкуренції встановлює ціни на основі коливань попиту і пропозиції, виробництво і ціни на товари з невеликим терміном зберігання, вийшовши зі стану рівноваги, не обов'язково повертаються до нього.

Модель отримала свою назву в 1934 році завдяки економісту Ніколасу Калдором на підставі того, що графік кривих, що відображають зміни цін, утворює павутину.

Модель дозволяє дослідити стійкість цін і обсягів виробництва на ринку, що описується кривими попиту та пропозиції деяких товару. Функція попиту S (p) характеризує залежність обсягу попиту на товар від ціни p товара в даний період і. Функція пропозиції D (p) характеризує обсяг пропозиції товару в залежності від ціни товару. Рівноважна ціна p, ринку визначається рівністю попиту та пропозиції S (p) = D (p).

Наприклад нехай ринок будь-якого окремого товару характеризується наступними функціями попиту і пропозиції (Формула 3.7).

$$D = D(P), S = S(P) \tag{3.7}$$

Для існування рівноваги ціна повинна бути такою, щоб товар на ринку був розпроданий, або D (P) = S (P).

Ціна рівноваги $\bar{P}$ задається цим рівнянням, а відповідний обсяг покупок-продажів, що позначається через $\bar{X}$, - таким рівнянням (Формула 3.8).

$$\bar{X} = D(\bar{P}) = S(\bar{P}) \tag{3.8}$$

Динамічна модель виходить при наявності запізнювання попиту або пропозиції. Найпростіша модель в дискретному аналізі включає незмінне запізнювання або відставання пропозиції на один інтервал (Формули 3.9 та 3.10).

$$D_t = D(P_t) \tag{3.9}$$

$$S_t = S(P_{t-1}) \tag{3.10}$$

Це може статися, якщо для виробництва даного товару потрібен певний період часу, обраний за інтервал. Дія моделі така, що при заданому P_{t-1} попереднього періоду обсяг пропозиції на ринку в поточному періоді буде S (P_{t-1}), і величина P_t повинна встановитися так, щоб був куплений весь обсяг

запропонованого товару. Іншими словами, P_t і обсяг покупок-продажів X_t характеризуються рівнянням (Формула 3.11).

$$X_t = D(P_t) = S(P_{t-1}) \tag{3.11}$$

Знаючи вихідну ціну P_o, за допомогою цих рівнянь ми можемо отримати значення P1 і X1. Потім, використовуючи наявну ціну P1, з відповідних рівнянь отримаємо значення P2 і X2 і т.д. Загалом зміна Pt характеризується різницевим рівнянням першого порядку (Формула 3.12).

$$D(P_t) = S(P_{t-1}) \tag{3.12}$$

Рішення можна проілюструвати діаграмою (Рис. 3.1), де D і S - відповідно криві попиту і пропозиції, а положення рівноваги, зі значеннями і, відповідає точці їх перетину Q. Ціна в початковий момент часу дорівнює P_o. Відповідна точка Q_o на кривій S дає обсяг пропозиції в період 1. Весь цей запропонований обсяг товару розкуповується при ціні P_1, заданої точкою Q_1 на кривій D з тієї ж ординатою X_1, що і Q_o. У другій період часу рух відбувається спочатку по вертикалі від точки Q_1 до точки на кривій S, що дає X_2, а потім по горизонталі - до точки Q_2 на кривій D. Остання точка характеризує P_2. Продовження цього процесу і дає графік павутини, показаний на Рис. 3.1. Ціни і обсяги (покупок - продажів) в послідовні періоди часу є відповідно координатами точок Q_1, Q_2, Q_3, ... на кривій попиту D. В даному випадку послідовність точок прагне до Q. При цьому точки по черзі розташовуються на лівій і правій стороні від Q. Отже, і значення ціни P_t прагнуть до, розташовуючись по черзі з обох боків від. Точно так само йде справа і з обсягами покупок - продажів (X_t).

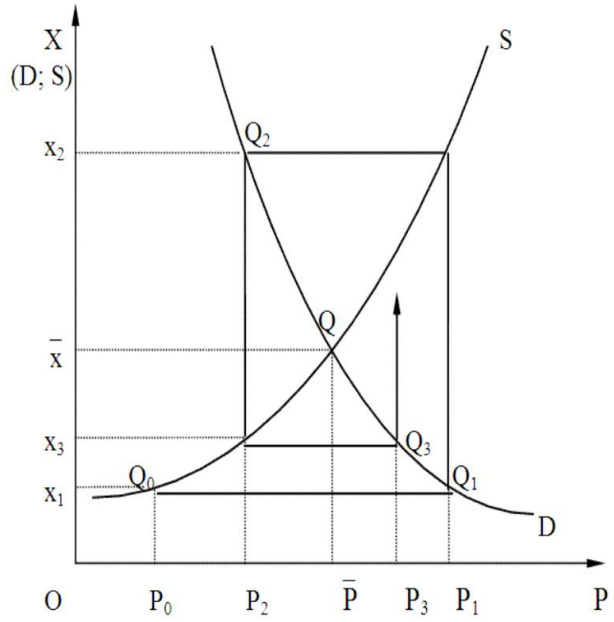

Рисунок 3.1– Діаграма відношень попиту і пропозиції для D і S.

Рішення можна отримати алгебраїчно для випадку лінійних функцій попиту і пропозиції D = a + aP, S = b + bP. Значення рівноваги і будуть задані наступними рівняннями $\bar{X} = a + a\bar{P} = b + b\bar{P}$. Дискретна динамічна модель задається рівнянням (Формула 3.13).

$$X_t = a + aP_t = b + bP_{t-1} \tag{3.13}$$

3.3 Авторегрессійна модель часового ряду

Прогнозування з використанням моделі авторегресії спирається на попередні значення продажів. Слово авторегресія означає залежність подальшого значення продажу від попередніх продажів. Залежність в разі авторегресії передбачається лінійно, тобто прогноз є сумою продажів за попередні дні з деякими коефіцієнтами, які є постійними і визначають параметри моделі авторегресії. Скільки днів таких продажів з минулого ми будемо брати, щоб намагатися спрогнозувати майбутні продажі називається порядком моделі авторегресії p.

Модель часового ряду, в якій його поточне значення лінійно залежить від попередніх значень цього ж ряду. Лінійна залежність означає, що поточне значення дорівнює зваженій сумі кількох попередніх значення ряду де C - константа, яку для простоти часто вважають рівною 0, n - число ретроспективних значень ряду, що враховуються в моделі (порядок моделі), в і - коефіцієнти (параметри) моделі, які потрібно оцінити при її побудові, ε(t) - випадкова складова, що відображає імовірнісний характер моделі. (Формула 3.14).

$$Y(t) = C + b_1 Y(t-1) + b_2 Y(t-2) + \ldots + b_n (t-n) + \varepsilon(t) = C + \sum_{i=1}^{n} b_i Y_{t-n} + \varepsilon_t \quad (3.14)$$

Якщо часовий ряд представляє собою щоденні продажі, то Y(t) - продажу сьогодні, Y(t-1) - продажу, які були вчора, Y(t-2) - позавчора і т.д., ε(t) - враховує вплив на продажу випадкових факторів, які неможливо доля в моделі або так званий білий шум (Рисунок 3.2). Можливе використання і інших шкал спостережень - щотижневі, щоквартальні та інші.

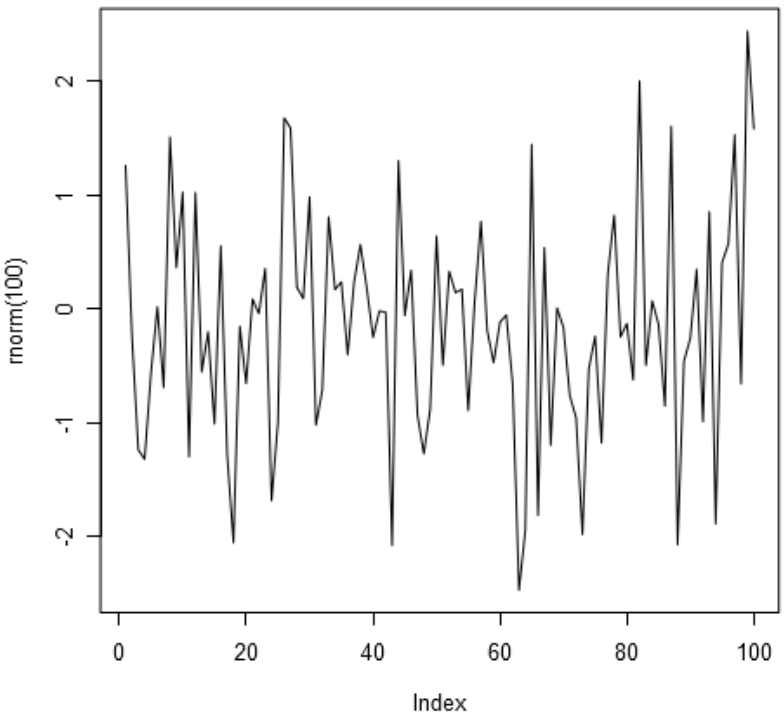

Рисунок 3.2 – Білий шум в авторегрессії

Таким чином, знаючи параметри моделі і відповідні попередні значення часового ряду, ми можемо передбачити його майбутні значення. Тому основне призначення авторегрессійної моделі - прогнозування. Крім цього, з її допомогою можна проводити аналіз часових рядів - виявляти тенденції, сезонність та інші особливості.

Для того щоб користуватися всім цим необхідно вибрати всі перераховані параметри. Звичайно, для цього існують спеціальні програми, які визначає оптимальні значення параметрів авторегрессії. Кінцевий користувач просто може скористатися готовою моделлю і отримати прогноз.

На плечі користувача лягає відповідальність за вибір порядку моделі авторегрессії. Скільки і які дні включати в модель. Для прогнозування на завтра враховувати продажу за кожен день попереднього тижня або тільки за кілька, а може краще враховувати продажу рівно тиждень тому? Ось тут можливо кілька підходів. Повний перебір всіх моделей в надії знайти хорошу модель або проаналізувати ряди, застосувати хитрі статистичні прийоми і зрозуміти які продажі найбільше впливають на те, що буде продано в наступному періоді.

Чи можна враховувати сезонність в моделі авторегрессії? Виявляється, що можна. Для того щоб враховувати сезонність в моделі авторегрессії необхідно додати в модель продажу за минулий сезон. Наприклад, якщо сезонність тижнева, то ми додамо продажу за 7 днів назад. Якщо річна сезонність, а ми прогнозуємо по місяцях, то в модель авторегрессії ми включимо продажу за місяць рік назад.

Розглянемо в загальному вигляді модель авторегрессії, де Y_n . Нехай ми хочемо дізнатися прогноз на день t (Формула 3.15).

$$Y_t = c + \varepsilon_t + a_1 * Y_{t-1} + a_2 * Y_{t-2} + a_3 * Y_{t-3} + \cdots \quad (3.15)$$

Розглянемо, як цей метод працює в реальних умовах. Весь процес починається з «підгонки» обраної моделі, а саме її порядку, до вихідних даних. Вихідні дані містять багато шуму, провалів, викидів. Якщо все це потрапить в модель, то авторегрессія як прогноз нічого доброго не видасть. Вона не зможе

узагальнити наявні тенденції, а просто запам'ятає ті дані, що були з усіма їх недоліками. І спрогнозує не попит, а проблеми зі складом і різкі викиди.

В цілому те як прогнозуємий метод поводиться на історичних, відомих йому даних називається моделлю.

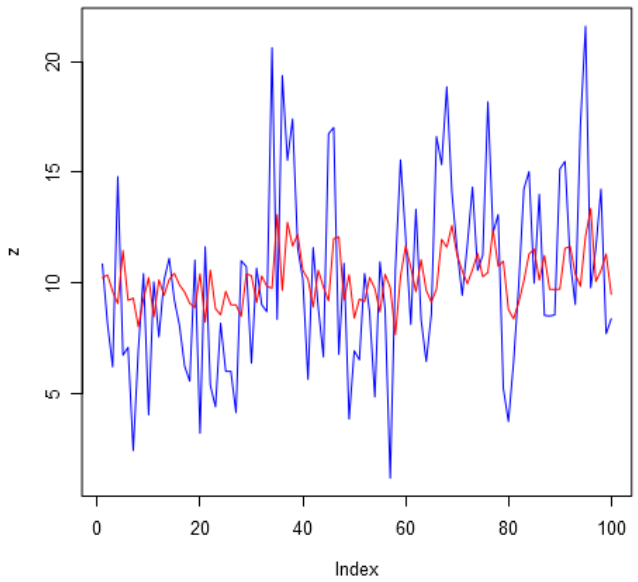

Рисунок 3.3 – Модель авторегресії першого порядку

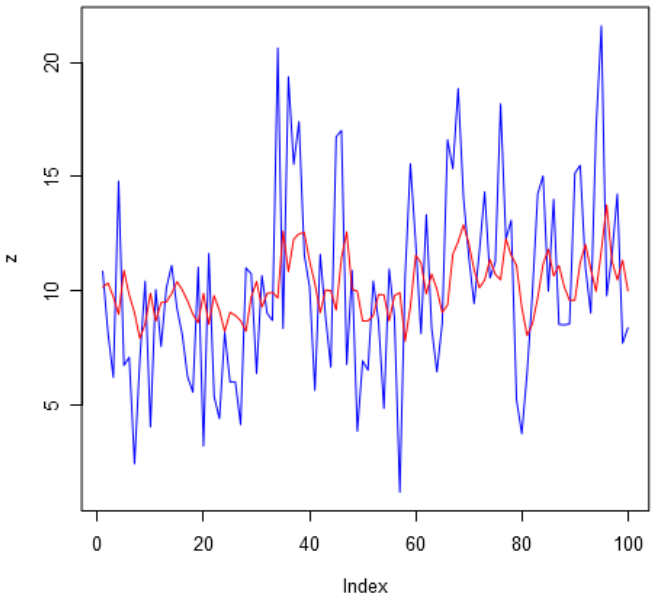

Рисунок 3.3 – Модель авторегресії другого порядку.

Математичні моделі дискретних систем управління використовують різницеві рівняння для рішення оптимізаційних задач запропоновані в роботі [12].

ВИСНОВКИ

В ході виконання науково-дослідницької роботи був виконаний аналіз існуючих методів моделювання та аналізу дискретних динамічних систем за допомогою різницевих рівнянь, що необхідні для прийняття рішень в різних сферах людської діяльності в області інтелектуального аналізу даних або Data Mining. Було розглянуто методи кластеризації в інтелектуальному аналізі даних, машинному навчанні, методи на основі дерев рішень і алгоритми на основі асоціативних зв'язків.

Досліджуючи та проводячи дослідження різницевих рівнянь в області інтелектуального аналізу даних було виявлено, що дана сфера розвивалася і продовжує розвивається на стику багатьох дисциплін, як статистика, теорія інформації, машинне навчання, теорія баз даних, і що більшість алгоритмів і методів інтелектуального аналізу даних були розроблені або запозичені на основі різних методів з цих дисциплін.

Також був проведений аналіз методів для моделювання дискретних динамічних систем за допомогою різницевих рівнянь і часових рядів.

Було побудовано такі моделі дискретних динамічних систем:

- Динамічна павутинообразна модель;
- Авторегрессійна модель часового ряду.

Крім цього моделі дискретних динамічних систем були побудовані таким чином, що їх можна використовувати у вигляді інструкції при моделюванні часових рядів за допомогою різницевих рівнянь.

В наступному часі необхідно вдосконалити побудовані моделі шляхом фільтрації шумів та звищенням рівнів в різницевих рівняннях.

ПЕРЕЛІК ПОСИЛАНЬ

1. Шумейко А. А. Інтелектуальний аналіз даних (Введення в Data Mining): навч. посіб. / А. А. Шумейко, С. Л. Сотнік. – Дніпропетровськ: Вдавництво Бєлая Е. А., 2012. – 212 с.

2. Башмаков А.І. Інтелектуальні інформаційні технології: навч. посіб./А.И. Башмаков, И.А. Башмаков. − М.: Вдав-во МГТУ ім. Н.Э. Баумана, 2005. − 304 с.

3. Методи і моделі аналізу даних: OLAP и Data Mining / А.А. Барсегян, М.С. Купріянов, В.В. Степаненко, И.И. Холод. – СПб.: «БХВ-Петербург», 2004. – 336 с.

4. Data Science: an introduction. Wikibooks [Electronic resource]. – 2017. – Mode of access: https://en.wikibooks.org/wiki/Data_Science:_An_Introduction. – Date of access: 01.02.2017.

5. Афанасієв В.Н., Юзбашев М.М. Аналіз часових рядів і прогнозування, М: Финансы и статистика. - 2001.

6. Баселлі Ф., Кохен Дж., Куадрат Дж. П. Синхронізація та лінійність: алгебра для дискретних систем подій. Чичестер, Вілі, 2009. 514 с.

7. Olshausen, B. A. Emergence of simple-cell receptive field properties by learning a sparse code for natural images, 1996. – p. 607-609.

8. Ткіндат В., Білуат Дж. С. Багатокритерійне планування: теорія, моделі та Алгоритми. Спрингер, Берлін, 2016. 25 с.

9. Романко В. К. Різницеві рівняння. – М.: Біном. Лабораторія знань. – 2015.

10. Халанай А., Векслер Д. Якісна теорія імпульсних систем. –Мир, М., 1971, 309 с.

11. Різницеві рівняння. Z-преобразування і його застосування: учбово-методичний посібник / С.В. Подолян, И.В. Юрченко – Могильов: МГУП, 2014. – 24 с.

12. М. Ф. Бондаренко, Власенко Л. А. Задача линейного квадратичного регулятора для дискрипторных сосредоточенных и распределенных систем с дискретным временем. – М.: Проблемы управления и информатики – 2015. 76-84 с.