

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається

В.о. завідувача кафедри

_____ О.І. Рязанцев

« ____ » _____ 2021 р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ
ПРЕЦЕДЕНТІВ

Освітньо-кваліфікаційний рівень «Магістр»

Спеціальність 122 – «Комп'ютерні науки»

Науковий керівник роботи:

(підпис)

Л. О. Шумова

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Я. О. Критська

(ініціали, прізвище)

Студент:

(підпис)

О. О. Сацук

(ініціали, прізвище)

Група:

КН-19дм

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень Магістр
Напрямок підготовки _____
Спеціальність 122 «Комп'ютерні науки»
(шифр і назва)

ЗАТВЕРДЖУЮ:

Т.в.о. завідувача кафедри
_____ В.С.Кардашук
« ____ » _____ 20__ р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Сацуку Олександр Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційні технології підтримки прийняття рішень на основі прецедентів

керівник проекту (роботи) Шумова Лариса Олександрівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом вищого навчального закладу від "05" 10 2020 р. № 140/15.15

2. Строк подання студентом роботи 11.01.2021

3. Вихідні дані до роботи Матеріали науково-дослідної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Теоретико-методологічні аспекти інтелектуальної підтримки прийняття рішень на основі прецедентів.

Аналіз концепцій CBR.

Реалізація методів прийняття рішень на основі прецедентів.

Охорона праці.

5. Перелік графічного матеріалу

Електронні плакати

6. Консультанти розділів проекту (роботи)

| Розділ | Прізвище, ініціали та посада Консультанта | Підпис, дата | |
|---------------|--|----------------|---------------------|
| | | завдання видав | завдання прийняв |
| Охорона праці | Критська Я. О. ст. викладач | | |
| | | | |
| | | | |
| | | | |

7. Дата видачі завдання 05.10.2020

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проекту (роботи) | Строк виконання етапів проекту (роботи) | Примітка |
|-------|--|---|----------|
| 1 | Аналіз завдання | 05.10.20 – 14.10.20 | |
| 2 | Літературний пошук. Визначення вимог до роботи. | 15.10.20 – 24.10.20 | |
| 3 | Огляд відомих засобів і методів рішення | 25.10.20. – 21.11.20 | |
| 4 | Дослідження і побудова рішення | 20.11.20 – 3.12.20 | |
| 5 | Експерименти моделювання | 3.12.20 – 20.12.20 | |
| 6 | Розгляд питань охорони праці та основних напрямків їх дотримання | 21.12.20 – 27.12.20 | |
| 7 | Оформлення пояснювальної записки | 02.01.21 – 05.01.21 | |
| 8 | Оформлення презентації роботи | 06.01.21 – 10.01.21 | |
| | | | |
| | | | |
| | | | |

Студент

_____ (підпис)

Сацук О.О.

_____ (прізвище та ініціали)

Науковий керівник

_____ (підпис)

Шумова Л.О.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Сацук О.О. Інформаційні технології підтримки прийняття рішень на основі прецедентів.

Розглянуті актуальні задачі інтелектуальної підтримки прийняття рішень на основі прецедентів. Проведено аналіз методів і програмних засобів для моделювання міркувань на основі прецедентів. Розроблено концептуальна модель підтримки прийняття рішень на основі прецедентів. Проведено експеримент моделювання у платформі Deductor для рішення конкретної задачі прийняття рішень.

Загальний обсяг роботи: Рис.: 14. Табл.: 4. Бібліогр.: 38.

Ключові слова: інформаційні технології, системи підтримки прийняття рішень, інтелектуальний аналіз даних, прецедентний підхід.

АНОТАЦИЯ

Сацук А.А. Информационные технологии поддержки принятия решений на основе прецедентов.

Рассмотрены актуальные задачи интеллектуальной поддержки принятия решений на основе прецедентов. Проведен анализ методов и программных средств для моделирования рассуждений на основе прецедентов. Разработана концептуальная модель поддержки принятия решений на основе прецедентов. Проведен эксперимент моделирования на платформе Deductor для решения конкретной задачи принятия решений.

Общий объем работы: Рис.: 14. Табл.: 4. Библиогр.: 38.

Ключевые слова: информационные технологии, системы поддержки принятия решений, интеллектуальный анализ данных, прецедентный подход.

ABSTRACT

Satsuk A. Information technology decision support based on precedents.

Topical tasks of intelligent decision support based on precedents are considered. The analysis of methods and software tools for modeling reasoning based on precedents. A conceptual model of decision support based on precedents has been developed. A simulation experiment was carried out on the Deductor platform for solving a specific decision-making problem.

Total scope of work: Fig.: 14. Tab.: 4. Bibliography: 38.

Keywords: information technology, decision support systems, data mining, precedent approach.

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧОК І СКОРОЧЕНЬ | 8 |
| ВСТУП | 9 |
| РОЗДІЛ 1 ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ АСПЕКТИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ В СИСТЕМАХ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ ПРЕЦЕДЕНТІВ | 11 |
| 1.1 Специфіка ІСППР | 11 |
| 1.2 Основні визначення в області ІАД | 12 |
| 1.3 Основні задачі інтелектуальної підтримки прийняття рішень | 13 |
| 1.4 Прецедентний підхід | 14 |
| 1.5 Способи подання і вилучення прецедентів | 16 |
| 1.5.1 Метод найближчого сусіда (NN - Nearest Neighbor) | 16 |
| 1.5.2 Метод вилучення прецедентів на основі дерев рішень | 16 |
| 1.5.3 Метод вилучення прецедентів на основі знань | 17 |
| 1.5.4 Метод вилучення прецедентів з урахуванням їх застосовності | 17 |
| 1.6 Програмні засоби та середовища для інтелектуальної підтримки прийняття рішень в СППР | 18 |
| 1.6.1 Protégé OWL | 18 |
| 1.6.2 Мова програмування C | 19 |
| 1.6.3 Мова програмування C++ | 19 |
| 1.6.4 Мова програмування C# | 19 |
| 1.6.5 Мова програмування Perl | 20 |
| 1.6.6 Мова програмування Python | 20 |
| 1.6.7 Мова програмування Java | 21 |
| 1.6.8 Deductor | 22 |
| 1.7 Висновок до першого розділу | 23 |
| 1.8 Перелік використаних джерел посилань до першого розділу | 23 |
| РОЗДІЛ 2 АНАЛІЗ КОНЦЕПЦІЙ СВР | 25 |
| 2.1 Концептуальна модель ІТППР на основі прецедентів | 25 |
| 2.2 Загальна постановка задачі навчання по прецедентах | 28 |
| 2.3 Аналіз процесу використання прецедентів при вирішенні нового завдання | 28 |
| 2.4 Вилучення прецедентів методом найближчого сусіда (найближчих сусідів) з використанням різних метрик | 30 |
| 2.5 Метод k-NN | 32 |

| | |
|---|----|
| 2.6 Ефективні алгоритми для витягу прецедентів | 33 |
| 2.7 Повторне використання прецедентів | 35 |
| 2.8 Адаптація та застосування прецедентів | 36 |
| 2.9 Збереження прецедентів | 36 |
| 2.10 Етапи розробки СВР систем | 37 |
| 2.11 Висновок до другого розділу | 37 |
| 2.12 Перелік використаних джерел посилань до другого розділу | 38 |
| РОЗДІЛ 3 РЕАЛІЗАЦІЯ МЕТОДІВ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ ПРЕЦЕДЕНТІВ | 39 |
| 3.1 Постановка задачі | 39 |
| 3.2 Опис можливостей АП Deductor | 40 |
| 3.2.1 Дерева рішень | 40 |
| 3.2.2 Карти Кохонена | 43 |
| 3.3 Алгоритм кластеризації | 44 |
| 3.4 Алгоритм навчання без вчителя для мереж Кохонена | 44 |
| 3.5 Використання карт Кохонена для вирішення задачі класифікації даних | 46 |
| 3.6 Аналіз отриманих результатів | 50 |
| 3.7 Висновок до третього розділу | 52 |
| 3.8 Перелік використаних джерел посилань до третього розділу | 52 |
| РОЗДІЛ 4 ОХОРОНА ПРАЦІ | 54 |
| 4.1 Аналіз стану умов праці | 54 |
| 4.1.1 Вимоги до приміщення | 54 |
| 4.1.2 Вимоги до організації місця праці | 54 |
| 4.2 Виробнича санітарія | 55 |
| 4.2.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу | 55 |
| 4.2.2 Пожежна безпека | 56 |
| 4.2.3 Електробезпека | 58 |
| 4.3 Гігієнічні вимоги до параметрів виробничого середовища | 58 |
| 4.3.1 Мікроклімат | 58 |
| 4.3.2 Освітлення | 59 |
| 4.4 Вентилювання | 60 |
| 4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій | 60 |

| | |
|--|-----------|
| 4.6 Розрахунок захисного заземлення (забезпечення електробезпеки будівлі). | 61 |
| 4.8 Екологія | 64 |
| 4.9 Висновки до четвертого розділу | 65 |
| 4.10 Перелік джерел посилань до четвертого розділу | 65 |
| ВИСНОВКИ | 67 |
| ДОДАТОК А Електронна презентація | 68 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧОК І СКОРОЧЕНЬ

| | |
|-------|--|
| БД | база даних |
| БП | бібліотека прецедентів |
| ІАД | інтелектуальний аналіз даних |
| ІСППР | інтелектуальна система підтримки прийняття рішень |
| ІТППР | інформаційні технології підтримки прийняття рішень |
| КБП | конструктор бібліотеки прецедентів |
| ОПР | особа, приймаюча рішення |
| РСМ | редактор семантичних мереж |
| СППР | система підтримки прийняття рішень |
| ШНМ | штучна нейронна мережа |
| СВР | Case-Based Reasoning |
| DM | Data Mining |
| KD | Knowledge Discovery |

ВСТУП

Актуальність теми. Інтелектуальні системи підтримки прийняття рішень (ІСППР) є затребуваним класом програмних систем. Їх створення - складне і ресурсовитратне завдання, особливо, якщо мова йде про слабоформалізованні області, що характеризуються великими обсягами використовуваних в них експертних знань [1]. Перед розробниками таких систем постає ряд проблем, від вирішення яких буде залежати ефективність готової ІСППР. Вони повинні враховувати цілі, які переслідуватиме створювана ІСППР і завдання, які вона буде вирішувати. Залежно від типів вирішуваних завдань повинні бути підібрані адекватні методи їх вирішення. Однак вибір відповідного методу - це окреме нетривіальне завдання. В рамках теорії прийняття рішень розроблено велику кількість методів підтримки прийняття рішень. Однак в них дуже важко орієнтуватися розробнику, який не має спеціальної підготовки. Велику допомогу в розробці ІСППР надають інструментарії. Але використання існуючих інструментаріїв обмежена їх жорсткою орієнтованістю на рішення певного класу задач. Крім того, вони, як правило, реалізують тільки один метод підтримки прийняття рішень, тоді як для вирішення багатьох реальних задач необхідне поєднання декількох методів.

Тому для розробки ІСППР потрібні досконаліші інформаційні технології інтелектуального аналізу даних (ІАД). У ІАД для вилучення нових знань з наявних даних застосовуються різні методи, серед яких активно просуваються методи правдоподібних міркувань на основі прецедентів (СВР – Case-Based Reasoning). [2, 3, 4]

Все вищенаведене свідчить про те, що задача розробки методів інтелектуального аналізу даних і відповідних програмних засобів є актуальною на даний момент. Тому обґрунтованою є тема магістерської роботи, у якій вирішується науково-прикладне завдання дослідження інформаційних технологій інтелектуальної підтримки прийняття рішень на основі прецедентів.

Об'єкт дослідження – процеси інтелектуальної підтримки прийняття рішень.

Предмет дослідження – програмні засоби інтелектуальної підтримки прийняття рішень на основі прецедентів.

Мета і завдання дослідження. Метою даної магістерської роботи є дослідження методів та програмних засобів інтелектуального аналізу даних на основі прецедентів в ІСППР.

Для досягнення зазначеної мети необхідно вирішити такі завдання:

- провести аналіз існуючих методів та програмних засобів аналітичної підтримки прийняття рішень в ІСППР на основі прецедентів;
- розглянути існуючі програмні рішення для ІСППР;

- створити концептуальну модель підтримки прийняття рішень на основі прецедентів;
- сформулювати загальну постановку задачі навчання по прецедентах;
- дослідити ефективні алгоритми для вилучення прецедентів;
- реалізувати і оцінити ефективність методів вилучення прецедентів для конкретної задачі.

Методи рішення поставлених задач базуються на комплексному використанні методів машинного навчання та інтелектуального аналізу даних. У якості методів математичного моделювання у роботі використовувалися методи нейро-мережевого моделювання (карта Кохонена), дерево рішень.

Апробація результатів роботи. Основні результати магістерської атестаційної роботи докладалися на VI науково-практичному молодіжному форумі «ІТ-ідея 2020» (м. Сєвєродонецьк) [5].

Практичне значення отриманих результатів. В результаті досліджень розроблена концептуальна модель підтримки прийняття рішень на основі прецедентів, проведені експерименти моделювання у платформі Deductor двома засобами (карт Кохонена і дерева рішень), виконано оцінку якості методів вилучення прецедентів.

Публікації. Основні результати магістерської атестаційної роботи опубліковані в тезах у збірнику науково-практичних праць VI молодіжного форуму «ІТ-ідея 2020» [5].

Структура і обсяг роботи. Робота складається зі вступу, чотирьох розділів, висновків і додатку. Загальний обсяг роботи складає 74 сторінок, з яких анотація на 1 сторінці, зміст на 3 сторінках, вступ на 2 сторінках, основний текст на 59 сторінках, висновки на 1 сторінці, список використаних джерел із 38 найменувань, додаток на 7 сторінках. Робота містить 4 таблиць та 14 рисунків.

РОЗДІЛ 1

ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ АСПЕКТИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ В СИСТЕМАХ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ ПРЕЦЕДЕНТІВ

1.1 Специфіка ІСППР

На сучасному етапі розвитку об'єкти управління приймають все більш складний, неоднозначний характер. При цьому вони постійно модернізуються, стають більш універсальними і надійними, ускладнюються їх організаційна, технічна та особливо інформаційна складова, що в свою чергу, призводить і до ускладнення процесу управління цими об'єктами. Велика кількість інформації як про самих складних організаційно-технічних об'єктах, так і про оточуюче їх середовище ускладнює роботу операторів відповідної системи управління: вони в змозі миттєво оцінити всі аспекти сформованої обстановки і прийняти правильне, обґрунтоване рішення в короткі терміни, особливо з урахуванням подальшого розвитку ситуації. При підвищенні вимог до якості управлінських рішень цей процес ускладнюється - чим вище відповідальність за прийняті рішення, тим більш якісною повинна бути оцінка. На допомогу операторам у цьому випадку приходять системи підтримки прийняття рішення, що дозволяють особам, які приймають рішення, здійснити повний і об'єктивний аналіз динамічно змінюється обстановки, а також прогноз її розвитку, особливо при вирішенні слабоструктурованих задач. Один з напрямків розвитку систем автоматизації процесів вироблення управлінських рішень пов'язаний з моделюванням процесів людського мислення - штучним інтелектом. Потужним інструментом в теорії штучного інтелекту вважаються експертні системи, які представляють клас комп'ютерних інформаційних систем, заснованих на роботі зі знаннями.

Так, наприклад, описав свою задачу А.В. Крилов у своїй науковій праці "Проблема вилучення знань з використанням методів на основі прецедентів" [6]. Використання прецедентного методу при побудові експертних систем управління складними організаційно-технічними об'єктами забезпечує додаткові можливості по підвищенню оперативності прийняття управлінських рішень та їх адекватності, а також прогнозування, завдяки накопиченим в базі прецедентів досвідченим даним.

В даний час досить актуальною проблемою в галузі штучного інтелекту є проблема конструювання інтелектуальних систем реального часу, типовими представниками яких є ІСППР, орієнтовані на відкриті і динамічні предметні області. В основі таких систем лежить інтеграція здатних до адаптації, модифікації і навчання моделей уявлення і оперування

знаннями, орієнтованих на специфіку проблемної (предметної) області та відповідний тип невизначеності, що відображає їх здатність до розвитку і зміни свого стану.

Система підтримки прийняття рішень вимагає трьох первинних компонентів: модулі управління, управління даними для збору і ручної обробки даних і управління діалогом для полегшення доступу користувача до СППР. Користувач взаємодіє із СППР через призначений для користувача інтерфейс, вибираючи приватну модель і набір даних, які потрібно використовувати, а потім СППР представляють результати користувачеві через той же самий призначений для користувача інтерфейс. Моделі управління і управління даними в значній мірі діють незалежно і варіюються від відносно простої типової моделі в електронній таблиці до складної комплексної моделі планування, заснованої на математичному програмуванні.

При реалізації методів моделювання міркувань в ІСППР необхідно враховувати специфіку подібних систем:

- необхідність отримання рішення в умовах часових обмежень, визначених реальним керованим процесом;
- необхідність врахування часового фактора при описі проблемної ситуації і в процесі пошуку рішення;
- неможливість отримання всієї об'єктивної інформації, необхідної для вирішення, і в зв'язку з цим використання суб'єктивної, експертної інформації;
- багатоваріантність пошуку;
- необхідність застосування методів правдоподібного виведення і активної участі в процесі пошуку ОПР;
- наявність недетермінізму, необхідність корекції і введення додаткової інформації в базу знань системи при пошуку рішення.

Методи пошуку рішення на основі прецедентів можуть застосовуватися в багатьох блоках ІСППР РВ (аналізаторі, блоках пошуку рішення, пояснення, моделювання і прогнозування) і дозволяють підвищити ефективність прийняття рішень ОПР в різних проблемних (аномальних) ситуаціях.

1.2 Основні визначення в області ІАД

ІАД - це процес виявлення в сирих даних раніше невідомих, нетривіальних, практично корисних і доступних інтерпретації знань, необхідних для прийняття рішень в різних сферах людських діяльності [7].

Під сирими даними розуміються дані, що не мають чіткої специфікації і містять необроблені (або оброблені в мінімальному ступені) фрагменти даних.

У зарубіжній літературі термін ІАД трактується як Knowledge Discovery (KD) і Data Mining (DM).

Під KD (виявлення знань) в БД розуміють будь-якої нетривіальний процес ідентифікації достовірних, нових, потенційно корисних і добре інтерпретуються зразків (структур) в даних. Таким чином, під процесом KD розуміють багатокроковому систему процедур, що включає підготовку даних, пошук зразків в БД, оцінку витягнутого знання, коригування та ітерацію процедур [8].

Під DM розуміють етап процесу KD, що складається в застосуванні специфічних алгоритмів породження зразків, витягнутих з БД.

Безліч зразків може бути відкритим, а їх перерахування реалізується спеціальним алгоритмом.

Процес ІАД включає чотири основні етапи. На першому етапі аналітик формулює постановку задачі в термінах цільових змінних. На другому етапі здійснюється підготовка даних для аналізу. На третьому етапі проводиться аналіз даних за допомогою методів DM. На четвертому етапі здійснюється верифікація та інтерпретація отриманих результатів (витягнутих знань). При верифікації застосовується тестовий набір записів, виділених з вихідних даних і не піддавалися аналізу.

Процес виявлення знань на основі ІАД включає в себе:

- вибір предметної області та релевантного знання для реалізації цілей кінцевого користувача комп'ютерної системи;
- вибір вихідного безлічі даних і підмножини змінних, які необхідні для отримання нового знання з бази фактів;
- уточнення даних і предпроцесінг;
- редукція даних;
- вибір завдання DM і алгоритмів, що реалізують DM для пошуку закономірностей в даних;
- видача результатів у формі, зручній для користувача;
- інтерпретація отриманих даних;
- огляд і узгодження виявленого знання.

1.3 Основні задачі інтелектуальної підтримки прийняття рішень

Система підтримки прийняття рішень (СППР) - це комп'ютерні інтерактивні системи, розроблені в допомогу менеджеру (або керівнику) при прийнятті рішень. СППР включають і дані, і моделі, щоб допомогти приймаючому рішення вирішити проблеми, особливо ті, які погано формалізовані.

Системи підтримки прийняття рішень допомагають знаходити відповіді не тільки на пряме запитання «що якщо?», а також на подібні. Типові питання по системам підтримки прийняття рішень (СППР):

- аналіз прикладів (case analyses) - оцінка значень вихідних величин для заданого набору вихідних змінних;
- параметричний (case analyses) аналіз - оцінка поведінки вихідних величин при зміні значень вихідних змінних;
- аналіз чутливості - дослідження поведінки результуючих змінних в залежності від зміни значення однієї або кількох вхідних змінних;
- аналіз можливостей - знаходження значень вхідної змінної, які забезпечують бажаний кінцевий результат (відомий також під назвою «пошук цільових рішень», «аналіз значень цілей», «управління за цілями»);
- аналіз впливу - виявлення для обраної результуючої змінної всіх вхідних змінних, що впливають на її значення, і оцінка величини зміни результуючої змінної при заданій зміні вхідної змінної, скажімо на 1%;
- аналіз даних - пряме введення в модель ранні відомих даних і маніпулювання при прогнозуванні;
- порівняння та узагальнення - порівняння результатів двох або більше прогнозів, зроблених при різних вхідних припущеннях, або порівняння передбачених результатів з дійсними, або об'єднання результатів, отриманих при різних прогнозах або для різних моделей;
- командні послідовності - можливість, використовувати, зберігати для подальшого використання регулярно виконуються серії команд і повідомлень;
- аналіз ризику - оцінка виконання вихідних змінних при випадкових змінах вхідних величин;
- оптимізація - пошук значень керованих вхідних змінних, що забезпечують оптимальне значення однієї або декількох результуючих змінних.

1.4 Прецедентний підхід

У більшості енциклопедичних джерел прецедент (від латинського *praecedentis* - попередній) визначається як випадок, що мав місце раніше і служить прикладом або виправданням для наступних випадків подібного роду [9]. Взагалі термін "прецедент" можна визначити як одиничне знання про попередній досвід. При цьому яку саме інформацію містить прецедент, залежить від предметної області і цілей використання прецеденту. Міркування на основі прецедентів базуються на накопиченому досвіді та подальшої адаптації рішення відомої задачі до вирішення нової. Прецедентний підхід дозволяє спростити процес прийняття рішень в умовах часових обмежень і при наявності різного роду невизначеності у вихідних даних і експертних знаннях, а також в разі виникнення різних нештатних (аномальних) ситуаціях.

Прецедентний підхід набуває широкого поширення. Наприклад, в сфері медичної діагностики рішення, пропоноване експертної системою, ґрунтується не тільки на анатомії і фізіології і теоретичних моделях діагнозів, але і на вивченні історії хвороб і лікарського досвіду, набутого при лікуванні пацієнтів (системи CASEY, PROTOS). Не менш широко прецедентний підхід використовується і в юриспруденції, де важливим компонентом правових міркувань є ідентифікація прецеденту для прийняття рішень в конкретному випадку (системи Rissland, Rissland & Ashly).

Висновок на основі прецедентів (CBR - CaseBased Reasoning) є підходом, що дозволяє вирішити нове, невідоме завдання, використовуючи або адаптуючи рішення вже відомої задачі, тобто використовуючи вже накопичений досвід вирішення подібних завдань.

Підхід на основі прецедентів виник в процесі розвитку досліджень в області створення експертних систем (систем, заснованих на знаннях). Експертні системи перших поколінь [1] були системами, що базуються на правилах (продукційного типу), що передбачали наявність досить добре формалізувати задачі. Для вирішення таких завдань застосовувалися або методи достовірного висновку, що, на основі вихідних даних відповідно до наявного в системі набором правил, формували висновок щодо поточної проблеми, або методи правдоподібного виведення для випадків невизначеності ймовірнісної природи (байесовський метод [10], метод на основі суб'єктивних ймовірностей [11] та інше).

На жаль, більшість практичних задач, орієнтованих на відкриті і динамічні предметні області, є погано формалізованими, причому невизначеність може мати неперевірною характер. При пошуку рішення таких задач необхідно застосування методів правдоподібного виведення, що дозволяють знайти якесь прийнятне (яке може і не бути оптимальним) рішення. Один з підходів базується на тому факті, що людині (експерту, ОПР) властиво на

першому етапі пошуку рішення нової (невідомої) задачі намагатися використовувати рішення, які приймалися раніше в подібних випадках, і при необхідності адаптувати їх до виниклої проблеми (поточної проблемної ситуації). Даний підхід з використанням накопиченого досвіду ліг в основу методів моделювання міркувань на основі прецедентів [5].

1.5 Способи подання і вилучення прецедентів

1.5.1 Метод найближчого сусіда (NN - Nearest Neighbor)

Метод найближчого сусіда є найбільш використовуваним для порівняння і вилучення прецедентів. Він дозволяє досить легко обчислити ступінь подібності поточної проблемної ситуації і прецедентів з БП.

Для визначення ступеня подібності на множині параметрів, використовуваних для опису прецедентів і поточної ситуації, вводиться деяка метрика. Далі відповідно до обраної метрики визначається відстань від цільової точки, відповідної поточної проблемної ситуації, до точок, що представляють прецеденти з БП, і вибирається найближча точка до цільової. Різні одифікації методу найближчого сусіда широко застосовується для вирішення завдань класифікації, кластеризації, регресії і розпізнавання образів. Зазвичай рішення вибирається на основі декількох найближчих точок (сусідів), а не однієї (метод k найближчих сусідів). Можливе використання методу, заснованого на знаннях про предметну область (певних залежностях між параметрами об'єкта та інші). Основними перевагами методу найближчого сусіда є простота реалізації та універсальність в сенсі незалежності від специфіки конкретної проблемної області. До істотних недоліків методу можна віднести складність вибору метрики для визначення ступеня подібності та пряму залежність необхідних обчислювальних ресурсів від розміру БП, а також неефективність при роботі з неповними і погано визначеними (так званими «зашумленими») вихідними даними.

1.5.2 Метод вилучення прецедентів на основі дерев рішень

Цей метод базується на знаходженні необхідних прецедентів шляхом досягнення вершин дерева рішень [12]. Кожна вершина дерева вказує, з якої її гілки слід здійснювати подальший пошук рішення. Вибір гілки здійснюється на основі інформації про поточні проблемні ситуації. Необхідно дістатися до кінцевої вершини, яка відповідає одному або декільком прецедентів. Якщо кінцева вершина пов'язана з деякою підмножиною прецедентів, то тоді для вибору найбільш підходящого з них може використовуватися метод найближчого сусіда. Такий підхід рекомендується застосовувати для великих БП, тому що основна

частина роботи по вилученню прецедентів виконується заздалегідь на етапі побудови дерева рішень, що значно скорочує час пошуку рішення.

Метод дерева рішень застосовується в задачах класифікації та прогнозування, коли рішення доводиться приймати в умовах ризику, невизначеності та результат подій залежить від ймовірностей. На кожне рішення впливають якісь певні чинники, і у кожного рішення є свої наслідки, яким притаманний імовірнісний характер. У цих умовах процес прийняття рішень є послідовним і метод дерева рішень передбачає визначати, які дії слід зробити в кожній вершині дерева.

Дерево рішень - математична модель, яка задає процес прийняття рішень так, що будуть відображені кожне можливе рішення, попередні та наступні цим рішенням події або інші рішення і наслідки кожного кінцевого рішення.

1.5.3 Метод вилучення прецедентів на основі знань

На відміну від попередніх методів, цей дозволяє врахувати знання експертів (ОПР) по конкретній предметній області (коефіцієнти важливості параметрів, виявлення залежності та інші) при вилученні прецедентів. Метод реалізує підхід, заснований на індексації прецедентів спеціальним чином (семантичної індексації). При визначенні прецедентів враховуються важливості параметрів прецедентів, задані експертом або ОПР, і інша інформація, що дозволяє врахувати знання про конкретну предметну область. За рахунок цього значно скорочується час пошуку рішення, що є істотною перевагою даного методу. Процес виконання індексації ускладнюється з ростом числа прецедентів в БП і необхідністю проводити індексацію динамічно. Для реалізації методу потрібно передбачити в структурі прецедентів і БП можливість подання і зберігання семантичної інформації, а також додаткові витрати на підтримання БП для обліку знань про конкретну предметну область. Метод може успішно застосовуватися спільно з іншими методами отримання прецедентів, особливо коли БП має великі розміри і предметна область є відкритою і динамічною. [13]

1.5.4 Метод вилучення прецедентів з урахуванням їх застосовності

У більшості систем, що використовують механізми міркувань на основі прецедентів, передбачається, що найбільш схожі з поточною проблемною ситуацією прецеденти є і найбільш застосовними у цій ситуації. Однак це не завжди так. В основі методів вилучення на основі застосування прецедентів лежить той факт, що витяг прецедентів базується не тільки на їх схожості з поточною проблемною ситуацією, але і на те, наскільки хороша для

бажаного результату модель вони собою являють. Таким чином, на вибір видобутих прецедентів впливає можливість їх успішного застосування (адаптації) в конкретній ситуації, тобто наявність відомостей про їх застосовності в ситуації, що склалася. У деяких системах ця проблема вирішується шляхом збереження прецедентів разом з коментарями щодо їх застосування. Використання даного методу дозволяє зробити пошук рішення більш ефективним, заздалегідь відкидаючи частину свідомо неперспективних прецедентів. Крім розглянутих методів вилучення прецедентів можуть успішно застосовуватися й інші методи (наприклад, апарат штучних нейронних мереж (ШНМ)). Безумовно, добре навчена ШНМ здатна успішно і досить швидко вирішувати завдання класифікації, кластеризації та визначення схожих прецедентів, але проблеми з ШНМ полягають в необхідності спользованих представницької навчальної вибірки для навчання мережі із заданою точністю і істотних витрат часу на навчання ШНМ. Крім того, виникає проблема, пов'язана з розробкою спеціальної топології ШНМ, орієнтованої на конкретну проблемну область і рішення складних багатопараметричних задач [14].

1.6 Програмні засоби та середовища для інтелектуальної підтримки прийняття рішень в СППР

Для реалізації механізмів інтелектуальної підтримки прийняття рішень у рамках СППР розглядаються два програмних компонента - конструктор бібліотеки прецедентів (КБП), що реалізує механізм міркування на основі прецедентів, і редактор семантичних мереж (PCM), що реалізує механізм пошуку рішення на основі структурної аналогії.

1.6.1 Protégé OWL

Protégé OWL [15] - це вільний, відкритий редактор онтологій для побудови баз знань, програмний продукт, розроблений Стенфордським університетом. Онтології, побудовані в цьому редакторі, експортуються в безліч форматів: RDF (RDF Schema), OWL і XML Schema. Цей редактор має відкриту і легко розширювану архітектуру.

Редактор Protégé-Frames дозволяє користувачам будувати і заповнювати онтології, засновані на фреймах, відповідно до ОКВС (Open Knowledge Base Connectivity protocol - прикладний інтерфейс програмування для доступу до баз знань систем уявлення знань). У цій моделі онтологія складається з набору класів, організованих в категоризовану ієрархію, для подання важливих понять області, набору слотів, пов'язаних з класами для опису їх

властивостей і відносин між ними, і набору примірників цих класів - окремих екземплярів понять, які мають певні значення своїх властивостей.

Редактор Protégé-OWL дозволяє користувачам будувати онтології для семантичної павутини, зокрема на OWL. OWL-онтологія може включати описи класів, властивостей і їх примірників. Даючи таку онтологію, формальна семантика OWL визначає як отримувати логічні сліdstва, тобто факти, які не присутні безпосередньо в онтології, але можуть бути виведені з існуючих за допомогою семантики. Ці висновки можуть бути засновані на одному документі або на безлічі розподілених документів, які були об'єднані з використанням певних механізмів OWL

1.6.2 Мова програмування C

C - стандартизований процедурний мову програмування, був розроблений на початку 80-х років минулого століття і використовується до цих пір. Спочатку розроблявся для використання в UNIX системах, але пізніше був перенесений на багато інших платформи і стан одним з найбільш популярних мов програмування і деякий час був одним з найбільш використовуваних мов для створення системного програмного забезпечення. Таку популярність C отримав завдяки своїй ефективності і простоті вивчення, за рахунок чого до сих пір використовується для навчання програмуванню. Також синтаксис C був узятий за основу багатьох інших мов програмування.

Незважаючи на досить великий обсяг набору операцій, C досить лаконічний.

1.6.3 Мова програмування C++

C++ - компільована статично типізована мова програмування загального призначення.

Підтримує такі парадигми програмування як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування, забезпечує модульність, роздільну компіляцію, обробку винятків, абстракцію даних, оголошення типів (класів) об'єктів, віртуальні функції. Стандартна бібліотека включає, в тому числі, загальноновживані контейнери і алгоритми. C ++ поєднує властивості як високорівневих, так і низькорівневих мов. У порівнянні з його попередником - мовою C, - найбільшу увагу приділено підтримці об'єктно - орієнтованого і узагальненого програмування

1.6.4 Мова програмування C#

C # - об'єктно-орієнтована мова програмування. Розроблено як мову розробки додатків для платформи Microsoft.NET Framework.

C # відноситься до сім'ї мов з C-подібним синтаксисом, з них його синтаксис найбільш близький до C ++ і Java. Мова має статичну типізацію, підтримує поліморфізм, перевантаження операторів (в тому числі операторів явного і неявного приведення типу), делегати, атрибути, події, властивості, узагальнені типи і методи, ітератори, анонімні функції з підтримкою замикань, LINQ, виключення, коментарі у форматі XML.

Переїнявши багато від своїх попередників - мов C ++, Delphi, Модула, Smalltalk і особливо Java - C #, спираючись на практику їх використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, C # на відміну від C ++ не підтримує множинне спадкування класів (між тим допускається множинне спадкування інтерфейсів).

1.6.5 Мова програмування Perl

Це високорівнева інтерпретуюма динамічна мова програмування загального призначення, створена Ларрі Уоллом. Основною особливістю мови вважаються її різноманітні можливості для роботи із текстом, в тому числі робота з регулярними виразами, що вбудована в синтаксис.

Perl - мова програмування загального призначення, яка була спочатку створена для маніпуляцій із текстом, але на даний момент використовується для виконання широкого спектру завдань, включаючи системне адміністрування, веб-розробку, мережеве програмування, ігри, біоінформатику, розробку графічних користувацьких інтерфейсів.

Мову можна охарактеризувати скоріше як практичну (легкість у використанні, ефективність, повнота), ніж красиву (елегантність, мінімалістичний). Головними перевагами мови є підтримка різних парадигм (процедурний, об'єктно-орієнтований і функціональний стилі програмування), контроль за пам'яттю (без збирача сміття, заснованого на циклах), вбудована підтримка обробки тексту, а також велика колекція модулів сторонніх розробників.

1.6.6 Мова програмування Python

Python - високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читаємості коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує кілька парадигм програмування, в тому числі структурну, об'єктно-орієнтовану, функціональну, імперативну і аспектно-орієнтовану. Основні архітектурні риси - динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень і зручні високорівневі структури даних. Код у Пітоні організовується у функції та класи, які можуть об'єднуватися в модулі (які в свою чергу можуть бути об'єднані в пакети).

1.6.7 Мова програмування Java

Java - об'єктно-орієнтована мова програмування, розроблений компанією Sun Microsystems (в подальшому придбаній компанією Oracle). Програми Java зазвичай компілюються в спеціальний байт-код, тому вони можуть працювати на будь-якій віртуальній Java-машині (JVM) незалежно від комп'ютерної архітектури. Програми на Java транслюються в байт-код, що виконується віртуальною машиною Java (JVM) - програмою, обробній байтовий код і передавальній інструкції обладнанню як інтерпретатор.

Перевагою подібного способу виконання програм є повна незалежність байт-коду від операційної системи і устаткування, що дозволяє виконувати Java-додатки на будь-якому пристрої, для якого існує відповідна віртуальна машина. Іншою важливою особливістю технології Java є гнучка система безпеки завдяки тому, що виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером) викликають негайне переривання.

Часто до недоліків концепції віртуальної машини відносять те, що виконання байт-коду віртуальною машиною може знижувати продуктивність програм і алгоритмів, реалізованих на мові Java. Останнім часом було внесено ряд удосконалень, які дещо збільшили швидкість виконання програм на Java:

- застосування технології трансляції байт-коду в машинний код безпосередньо під час роботи програми (JIT-технологія) з можливістю збереження версій класу в машинному кодї;

- широке використання платформенно-орієнтованого коду (Native-код) в стандартних бібліотеках;
- апаратні засоби, що забезпечують прискорену обробку байт-коду (наприклад, технологія Jazelle, підтримувана деякими процесорами фірми ARM).

Ідеї, закладені в концепцію і різні реалізації середовища віртуальної машини Java, надихнули безліч ентузіастів на розширення переліку мов, які могли б бути використані для створення програм, що виконуються на віртуальній машині. Цим ідеям знайшли також застосування в специфікації загальномовної інфраструктури CLI, закладеної в основу платформи .NET компанією Microsoft.

1.6.8 Deductor

Deductor є аналітичною платформою - основою для створення закінчених прикладних рішень в області аналізу даних. Реалізовані в Deductor технології дозволяють на базі єдиної архітектури пройти всі етапи побудови аналітичної системи: від створення сховища даних до автоматичного підбору моделей і візуалізації отриманих результатів.

Deductor Warehouse - багатовимірне сховище даних, що акумулює всю необхідну для аналізу предметної області інформацію. Використання єдиного сховища дозволяє забезпечити несуперечність даних і централізоване зберігання, а також автоматично забезпечує всю необхідну підтримку процесу аналізу. Deductor Warehouse оптимізований для вирішення саме аналітичних задач, що позитивно позначається на швидкості доступу до даних. У ряді випадків має сенс відмовитися від традиційного сховища і скористатися альтернативою - віртуальним сховищем Virtual Warehouse.

Deductor Studio - програма, яка реалізує функції імпорту, обробки, візуалізації і експорту даних. Вона може функціонувати і без сховища, отримуючи інформацію з будь-яких інших джерел, але найбільш оптимальним є їх спільне використання. В Deductor Studio включений повний набір механізмів, що дозволяє отримати інформацію з довільного джерела даних, провести весь цикл обробки (очищення, трансформацію даних, побудову моделей), відобразити отримані результати найбільш зручним чином (OLAP, таблиці, діаграми, дерева і т. Д.) і експортувати їх в найбільш поширені формати.

Deductor Client - клієнт доступу до Deductor Server. Він забезпечує доступ до сервера з сторонніх додатків і управління його роботою. Реалізована в Deductor архітектура дозволяє досягнути максимальної гнучкості при створенні закінченого рішення. Завдяки цій архітектурі можна зібрати в одному аналітичному додатку всі необхідні інструменти аналізу і реалізувати автоматичне виконання підготовленого сценарію. Технологічна платформа

включає кошти, що дозволяють максимально скоротити терміни розробки, швидко створювати і виводити на ринок нові прикладні рішення, а також адаптувати їх відповідно до вимог, що змінюються у підприємств. Створення закінченого рішення займає дуже мало часу: досить отримати дані, визначити сценарій обробки і задати місце для експорту отриманих результатів. Наявність потужного набору механізмів обробки і візуалізації дозволяє рухатися по кроках, від найпростіших способів аналізу до найбільш ефективних. Перші результати користувач отримує практично одразу, але при цьому можна легко нарощувати ефективність рішення.

1.7 Висновок до першого розділу

Використання прецедентного методу при побудові ІСППР забезпечує, завдяки накопиченим в базі прецедентів досвідченим даним, додаткові можливості підвищення оперативності прийняття рішень та їх адекватності. Для цього необхідна наявність великих баз прецедентів в конкретній предметній області, що вимагає значних витрат часу. З іншого боку, збільшення обсягу прецедентної бази призводить до збільшення ресурсів, необхідних для отримання прецедентів з бази і знаходження найбільш схожою з поточної ситуації. Комплексне використання методів вилучення прецедентів дозволяє істотно скоротити трудомісткість і час пошуку. В основу комплексу розглянутих методів може бути покладено метод найближчих сусідів, а з використанням інших методів здійснюється попередня фільтрація прецедентів в базі. Таким чином, застосування прецедентного методу в СППР дозволяє значно скоротити час на прийняття рішення і підвищити його адекватність.

Також було розглянуто існуючі програмні засоби та середовища для інтелектуальної підтримки прийняття рішень в СППР та проведено їх порівняльний аналіз для обґрунтованого вибору необхідного програмного забезпечення.

1.8 Перелік використаних джерел посилань до першого розділу

1. Первое поколение экспертных систем [электронный ресурс] - режим доступа: www.URL:itstan.ru/it-i-is/pervoe-pokolenie-ekspertnyhsistem.html#:~:text=Экспертные%20системы%20первого%20поко%D
2. Рассел С., Норвиг П. Искусственный интеллект: современный подход; [пер. с англ.]. М.: Вильямс, 2006. 1410 с.
3. Финн В.К. Об интеллектуальном анализе данных // Новости искусственного интеллекта. 2004. № 3. С. 3–19.

4. Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В. Достоверный и правдоподобный вывод в интеллектуальных системах. М.: Физматлит, 2008. 712 с.
5. Сацук О.О., Шумова Л.О. Інформаційні технології підтримки прийняття рішень на основі прецедентів // ІТ-Ідея – 2020: збірник науково-практичних праць, 11 груд. 2020 р., м. Сєвєродонецьк. Сєвєродонецьк : Видавництво СНУ ім. В. Даля, 2020. - С. 38-40.
6. А. В. КРЫЛОВ Проблема извлечения знаний с использованием рассуждений на основе прецедентов / А. В. КРЫЛОВ. - Санкт-Петербургский институт информатики и автоматизации РАН, 199178, Санкт-Петербург, Россия, 2018.
7. Рассел С., Норвиг П. Искусственный интеллект: современный подход; [пер. с англ.]. М.: Вильямс, 2006. 1410 с.
8. Финн В.К. Об интеллектуальном анализе данных // Новости искусственного интеллекта. 2004. № 3. С. 3–19.
9. Е.С. Макарова Модели и алгоритмы интеллектуальной поддержки принятия решений в области ИТ-консультирования на основе метода прецедентов / Е.С. Макарова - ФГБОУ ВО НГТУ, Новосибирск, 2017.
10. Как применять теорему Байеса для решения реальных задач [электронный ресурс] - режим доступа: www. URL: <https://neurohive.io/ru/osnovy-data-science/kak-primenjat-teoremu-bajesa-dlja-reshenija-realnyh-zadach/>
11. М.В. Кармінська-Бєлоброва Управління ризиками у підприємстві: навч. посібник /М.В. Кармінська-Бєлоброва, Є.М. Ігнатова ; Харківський інститут управління, 2014
12. Метод дерева решений и другие методы на основе графов [электронный ресурс] - режим доступа: www. URL: https://function-x.ru/graphs4_modeling_decision_tree_game_tree.html
13. Варшавский П.Р. Архитектура системы распределенного вывода на основе прецедентов для интеллектуальных систем / П.Р. Варшавский, К.Н. Бредихин // Программные продукты и системы. – 2011. № 1. – С. 50-53.
14. С.А. Ус Моделі й методи прийняття рішень / С.А. Ус, Л.С. Коряшкіна; Дніпропетровський національний гірничий університет, 2014
15. Protégé [электронный ресурс] - режим доступа: www. URL: <https://protege.stanford.edu/>

РОЗДІЛ 2

АНАЛІЗ КОНЦЕПЦІЙ СВР

2.1 Концептуальна модель ІТПР на основі прецедентів

Як правило, методи міркування на основі прецедентів включають в себе чотири основні етапи, які утворюють так званий СВР-цикл (рис. 2.1). [1].

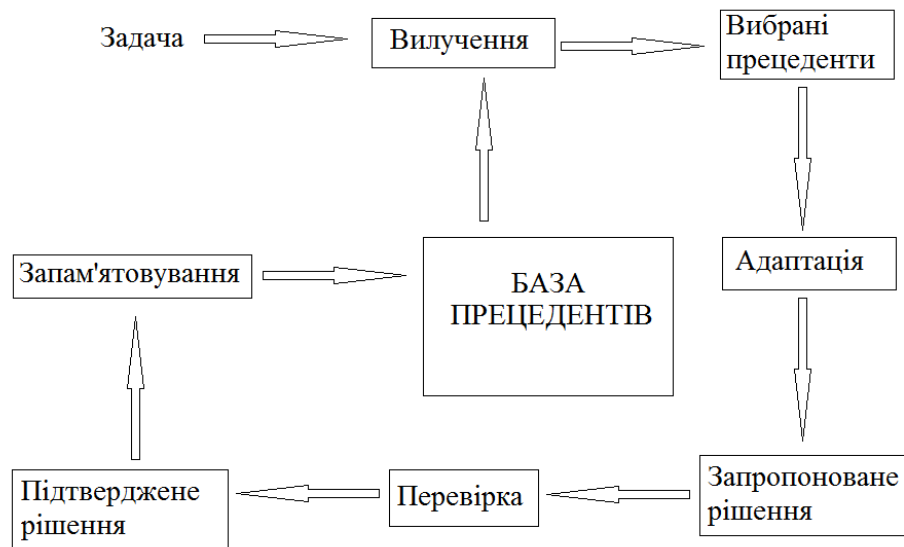


Рисунок 2.1 – Структура СВР-циклу

У роботі [2] СВР-цикл називається також циклом навчання по прецедентах. Основними етапами СВР-циклу є:

- витяг найбільш адекватного (подібного) прецеденту (або прецедентів) для ситуації, що склалася з бібліотеки прецедентів (БП);
- повторне використання вилученого прецеденту для спроби вирішення поточної проблеми;
- перегляд і адаптація в разі необхідності отриманого рішення відповідно до поточної проблемою;
- збереження (запам'ятовування) знову прийнятого рішення як частини нового прецедента.

На першому етапі СВР-циклу - добуванні прецедентів - виконується визначення ступеня подібності поточної ситуації із прецедентами з БП і подальше їх вилучення з метою вирішення цієї нової проблемної ситуації. Для успішної реалізації міркувань на основі прецедентів необхідно забезпечити коректний витяг прецедентів з БП. Вибір методу

отримання прецедентів безпосередньо пов'язаний зі способом подання прецедентів і відповідно з організацією БП.

Другий етап - повторне використання вилученого прецеденту для спроби вирішення поточної проблеми.

На третьому етапі переглядають і адаптують прецеденти в разі необхідності отриманого рішення відповідно до поточної проблеми.

Останній, четвертий етап, це збереження прийнятого рішення як частини нового прецеденту.

Найчастіше другий, третій і четвертий етапи не враховуються, але витяг прецеденту є основним етапом CBR-циклу.

БП може входити до складу бази знань інтелектуальної системи (ІСППР), але може виступати і як самостійний компонент системи. Структура БП істотно впливає на різні показники роботи системи і, зокрема, на час пошуку та вилучення прецедентів. Існують різні способи подання та зберігання прецедентів - від простих (лінійних) до складних ієрархічних. Слід зазначити, що прості способи, що базуються, як правило, на технології реляційних баз даних, вимагають значно менших витрат на реалізацію, а також на підтримку і супровід БП, ніж складні. Однак часу для пошуку рішення при простому поданні прецедентів може знадобитися значно більше в порівнянні з іншими більш складними способами представлення і зберігання прецедентів.

У загальному випадку модель уявлення прецеденту включає:

- опис ситуації;
- рішення для даної ситуації;
- результат застосування рішення.

Основною метою використання прецедентного підходу є отримання рішення для поточної ситуації на основі прецедентів, які вже мали місце в минулому. Прецедентний підхід доцільний, коли достовірні алгоритми неспроможні (присутні неповнота знань про предметну область, наявність обмежень по тимчасовим та обчислювальним ресурсам) або взагалі відсутні. З урахуванням поставлених цілей і завдань була сформована концептуальна модель інформаційної технології підтримки прийняття рішень (ІТППР) на основі прецедентів (рис. 2.2). [3]

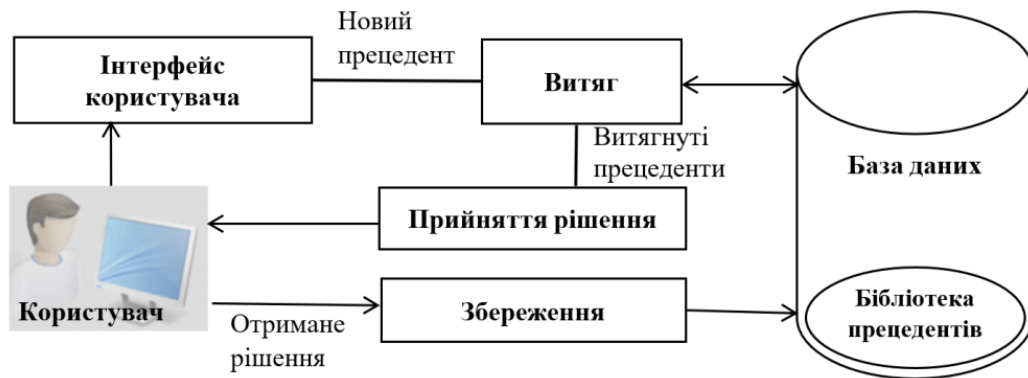


Рисунок 2.2 - Концептуальна модель ІТППР на основі прецедентів

До переваг CBR методів можна віднести:

- можливість безпосередньо використовувати досвід, накопичений системою, без інтенсивного залучення експерта в тій чи іншій предметної області;
- можливість скорочення часу пошуку рішення за рахунок використання вже наявного рішення для такого завдання;
- можливість виключення повторного отримання помилкового рішення;
- відсутність необхідності поглибленого вивчення і використання всіх наявних знань з предметної області, так як можна обмежитися урахуванням тільки істотних особливостей предметної області;
- можливе застосування евристик, що підвищують ефективність процесу пошуку рішення.

До недоліків CBR методів можна віднести наступне:

- при описі прецедентів зазвичай обмежуються поверхневими знаннями про предметну область;
- велика кількість прецедентів може привести до зниження продуктивності системи;
- проблематичним є визначення критеріїв для індексації і порівняння прецедентів;
- складності з налагодженням алгоритмів визначення подібних (аналогічних) прецедентів;
- неможливість отримання рішення задач, для яких немає прецедентів або ступінь їх подібності менше заданого порогового значення.

У загальному сенсі під прецедентом слід розуміти:

- умова задачі;
- фрагмент досвіду.

Таким чином, прецедент визначається як пара (завдання, рішення) в базі прецедентів (БП).

Для реалізації CBR систем на початковому етапі необхідно вибрати спосіб представлення прецедентів. У загальному випадку прецедент може включати наступні компоненти:

- опис завдання (проблемної ситуації);
- рішення задачі (діагноз по проблемної ситуації і рекомендації);
- результат (або прогноз) застосування рішення.

Для представлення прецедентів можуть використовуватися різні способи від простого параметричного уявлення до складних структурованих уявлень (у вигляді дерев, фреймів, мережних структур і т.д.). В роботі розглядається просте параметричне представлення прецедентів:

$$\text{CASE} = (x_1, x_2, \dots, x_n, R), \quad (2.1)$$

де x_1, \dots, x_n – параметри ситуації, яка описує даний прецедент ($x_1 \in X_1, x_2 \in X_2, \dots, x_n \in X_n$), n – кількість параметрів, R – рішення, а X_1, \dots, X_n – області допустимих значень відповідних параметрів.

2.2 Загальна постановка задачі навчання по прецедентах

Задано безліч об'єктів X , безліч допустимих об'єктів Y , і існує цільова функція $y^*: X \rightarrow Y$, значення якої $y_i = y^*(x_i)$ відомі тільки на кінцевому підмножині об'єктів $\{x_1, \dots, x_l\} \in X$. Пари «об'єкт - відповідь» (x_i, y_i) називаються прецедентами. сукупність пар $X^l = (x_i, y_i)_{i=1}^l$ називаються навчальною вибіркою. Завдання навчання по прецедентах полягає в тому, щоб за вибіркою X^l відновити залежність y^* , тобто побудувати вирішальну функцію $a: X \rightarrow Y$, яка наближала б цільову функцію $y^*(x)$, причому не тільки на об'єктах навчальної вибірки, а й на всій безлічі X . Вирішальна функція a повинна допускати ефективну комп'ютерну реалізацію, з цього будемо називати її алгоритмом.

2.3 Аналіз процесу використання прецедентів при вирішенні нового завдання

На рис.2.3 показано процес використання прецедентів при вирішенні нового завдання, реалізується в такий спосіб.

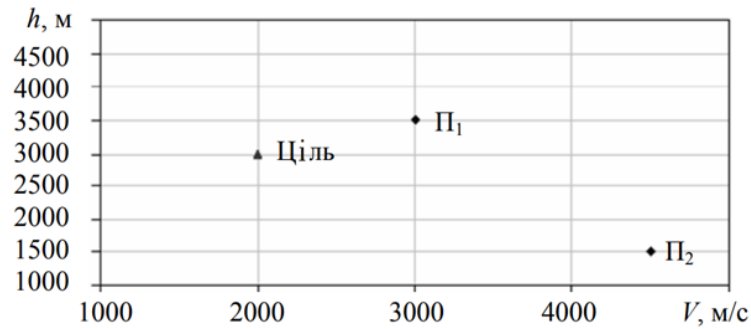


Рисунок 2.3 - Процес використання прецедентів

Спочатку необхідно застосувати метод вилучення з урахуванням застосування прецедентів, виключивши тим самим прецеденти, результати рішень по яким були негативними або призвели до негативних наслідків. Далі можна скористатися методом вилучення прецедентів на основі знань. Для цього необхідно заздалегідь побудувати шкали відносної важливості параметрів для основних видів дій, характерних для конкретної предметної області. Шаплони повинні бути підготовлені з урахуванням знань експертів в даній галузі. Застосовуючи певні шаплони для поточної ситуації, слід в черговий раз скоротити кількість прецедентів, вибравши з залишившогося набору прецеденти, які підходять за ступенем значущості параметрів для поточної ситуації.

На кінцевому етапі необхідно скористатися методом найближчих сусідів [4]. На практиці застосовуються різні модифікації цього методу. Зазвичай рішення вибирається на основі декількох найближчих точок (сусідів), а не однієї (метод k найближчих сусідів). В алгоритмі визначення найближчого сусіда використовується просте координатне зіставлення поточної ситуації з прецедентом (кожен параметр для опису прецедентів розглядається як одна з координат). Таким чином, визначається відстань D між точкою, що відповідає поточній ситуації, і точкою, що відповідає прецеденту, а також обчислюється максимальна відстань D_{max} з використанням кордонів діапазонів параметрів для опису прецедентів. Потім обчислюється значення ступеня подібності:

$$SIM = 1 - D / D_{max} \quad (2.2)$$

$$SIM = (1 - D / D_{max}) 100\% \quad (2.3)$$

Розглянемо алгоритм визначення найближчого сусіда на наступному прикладі. Нехай прецеденти і поточна ситуація описуються тільки двома параметрами:

V - швидкість об'єкта (діапазон зміни швидкості 1000 ... 5000 м / с);

h - висота об'єкта над рівнем моря (Діапазон зміни висоти 1000 ... 5000 м).

У БП є два прецеденти:

$$П1: V = 3000 \text{ м / с}, h = 3500 \text{ м}; П2: V = 4500 \text{ м / с}, h = 1500 \text{ м}$$

Для поточної ситуації (мета) $V = 2000 \text{ м / с}, h = 3000 \text{ м}$.

Далі розглянемо координатну площину (рис. 1), де по осі абсцис відкладені значення V , а по осі ординат - h . Мета має координати (2000, 3000), а прецеденти - П1 (3000, 3500), П2 (4500, 1500).

Розрахуємо відстань від мети до П1 ($D1$) і П2 ($D2$):

$$D1 = \sqrt{(2000 - 3000)^2 + (3000 - 3500)^2} = 1118 \text{ м},$$

$$D2 = \sqrt{(2000 - 4500)^2 + (3000 - 1500)^2} = 2915 \text{ м},$$

Аналогічним чином обчислюється максимальна відстань D_{max} між точками з координатами (1000, 1000) і (5000, 5000), яка складе 5657 м.

Потім обчислюються значення ступеня подібності SIM поточної ситуації з двома прецедентами з БП:

$$\text{для П1: } SIM1 = 1 - \frac{D1}{D_{max}} = 1 - \frac{1118}{5657} = 0.8 \text{ (80\%)}$$

$$\text{для П2: } SIM2 = 1 - \frac{D2}{D_{max}} = 1 - \frac{2915}{5657} = 0.48 \text{ (48\%)}$$

Таким чином, прецедент П1 є більш схожим з поточною ситуацією в порівнянні з прецедентом П2. При управлінні складними організаційно-технічними об'єктами кількість контрольованих і прийнятих до обліку параметрів іноді досягає декількох сотень, а часом і тисяч. У цих випадках порівняння прецедентів з поточною ситуацією розглядається не в двухкоординатній площині, а в n -вимірному просторі

2.4 Вилучення прецедентів методом найближчого сусіда (найближчих сусідів) з використанням різних метрик

Це найпопулярніший і часто використовуваний метод для вилучення прецедентів, який також широко застосовується для вирішення завдань класифікації, регресії, розпізнавання образів тощо. В його основі лежить певний спосіб вимірювання ступеня

схожості (близькості) прецеденту і поточної проблемної ситуації. Говорячи формально необхідно ввести метрику на просторі параметрів (ознак, властивостей) для опису прецедентів і поточної ситуації, а потім, визначивши відстань в обраній метриці між точками, які відповідають прецедентів, і точкою, що характеризує поточну ситуацію, вибрати найближчу точку - прецедент (найближчого сусіда) до поточної проблемної ситуації. Безумовно, ефективність методу найближчого сусіда багато в чому залежить від вибору метрики (міри схожості). Нехай заданий прецедент (С) і поточна проблемна ситуація (Т) в n-вимірному просторі ознак (властивостей, якостей) x_1, x_2, \dots, x_n для С і y_1, y_2, \dots, y_n для Т, тоді ступінь подібності або близькості можна визначити, використовуючи одну з перерахованих нижче основних метрик для визначення відстані між С і Т (d_{CT}):

Евклидова метрика (Евклідова відстань):

$$d_{CT} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.4)$$

Квадрат евклідової відстані:

$$d_{CT} = \sum_{i=1}^n (x_i - y_i)^2 \quad (2.5)$$

Манхэттенская метрика:

$$d_{CT} = \sum_{i=1}^n |x_i - y_i| \quad (2.6)$$

Відстань Чебишева:

$$d_{CT} = \max |x_i - y_i| \quad (2.7)$$

Вибір відповідної метрики творча і досить трудомістке завдання, від успішного вирішення якої безпосередньо залежить результативність пошуку та вилучення прецедентів. У кожному конкретному випадку цей вибір проводиться по-різному, в залежності від цілей ОПР (користувача), фізичної та статистичної природи використовуваної інформації при

управлінні складним об'єктом і інших обмежень, і чинників, що впливають на процес пошуку рішення. У деяких методах вибір відповідної метрики досягається за допомогою спеціальних алгоритмів перетворення початкового простору ознак (параметрів), а в інших випадках експерт (ЛПР) сам визначає метрику, спираючись на власні знання з предметної області або експериментальні дані. Таким чином, в методі визначення найближчого сусіда використовується просте покоординатно зіставлення поточної ситуації з прецедентом (кожен параметр для опису прецедентів розглядається як одна з координат) і визначається відстань d_{CT} між поточною ситуацією і прецедентом. Для визначення значення ступеня подібності $S(C, T)$ необхідно знайти максимальну відстань d_{MAX} в обраній метриці, використовуючи кордону діапазонів параметрів для опису прецедентів (x_{h1}^{CT} і x_{hn}^{CT} , $i = 1, \dots, n$). Потім можна обчислити значення ступеня подібності

$$S(C, T) = 1 - d_{CT}/d_{MAX} \quad (2.8)$$

Поряд з вищевказаними основними метриками на практиці часто застосовуються такі міри схожості (близькості):

Міра близькості Журавльова:

$$S(C, T) = \sum_{i=1}^n I_i^{CT} \quad (2.9)$$

де $I_i^{CT} = 1$, якщо $|x_i - y_i| < E$, інакше 0.

Міра близькості Хеммінга:

$$S(C, T) = \frac{n_{CT}}{n} \quad (2.10)$$

де n_{CT} - число співпадаючих ознак (параметрів) у прецеденту C і ситуації T .

На жаль, даний метод не дає чітких рекомендацій для випадку, коли присутні кілька прецедентів, рівновіддалених від поточної ситуації в заданій метриці. Крім того, метод неефективний в разі зашумленості вихідних даних і в разі їх неповноти. Для усунення вищевказаних недоліків існує досить велика кількість різних модифікацій методу найближчого сусіда. Найбільш відомим є метод k -NN.

2.5 Метод k-NN

Метод k-NN широко поширений метод для вирішення задач класифікації, кластеризації, дискримінантного аналізу, регресії і розпізнавання образів. Суть методу полягає у визначенні заданого числа k найближчих сусідів (прецедентів) до нової ситуації, що склалася в просторі ознак (параметрів). Що стосується виконання завдання класифікації визначається, до якого діагностичного класу належить більшість найближчих сусідів (наприклад, може застосовуватися просте голосування) і до цього класу належить поточна проблемна ситуація. Вибір значення параметра k займає ключове місце в методі k-NN. Дійсно, параметр k є одним з найбільш значущих чинників, що впливають на якість одержуваного результату. Один з підходів до оцінки необхідного числа найближчих сусідів - сприймати k як параметр згладжування. Збільшення k призводить до зменшення впливу випадкових похибок в даних, але при цьому поділ на класи стає менш чітким. Отже, для параметра k, як і для будь-якого згладжує параметра, необхідно визначити оптимальне значення, при якому б досягався бажаний компроміс. Оптимальне число k можна підбирати експериментальним шляхом або використовувати спеціальні методи оцінки невідомих параметрів (наприклад, метод крос-перевірки). Як і у випадку з методом простого найближчого сусіда в методі k-NN необхідно вирішувати задачу вибору метрики для визначення близькості діагностованих ситуацій. Це завдання в умовах великої розмірності простору ознак надзвичайно складна і передбачає виконання аналізу багатовимірної структури експериментальних даних для мінімізації числа об'єктів (прецедентів), що представляють діагностичні класи.

2.6 Ефективні алгоритми для вилучення прецедентів

В роботі для отримання прецедентів був обраний метод вилучення прецедентів на основі алгоритму k-NN [5] з можливістю вибору різних метрик (наприклад, Евклідовій, Манхеттенський, Чебишева, міри близькості Журавльова, міри схожості по Хеммінга) та обліком порогового значення ступеня подібності (рис. 2.4)

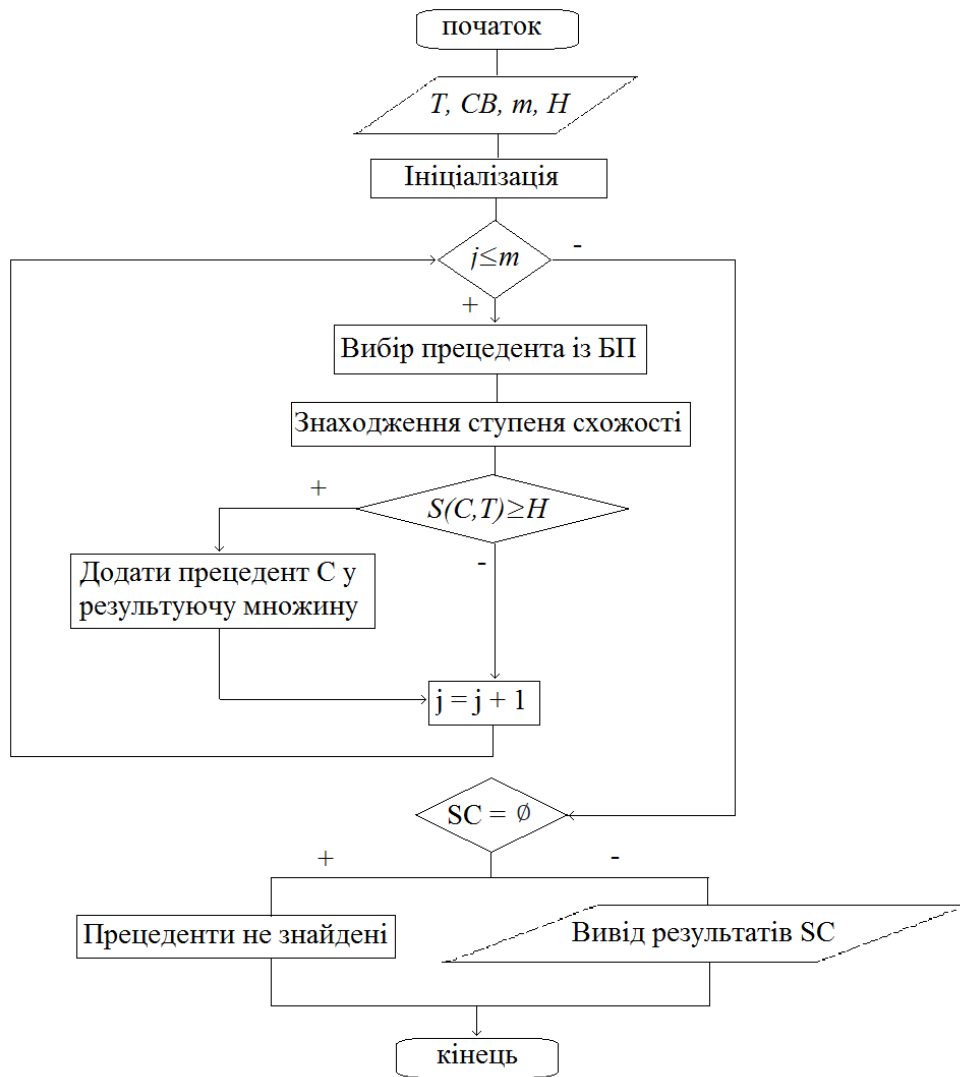


Рисунок 2.4 - Блок схема алгоритму для витягу прецедентів

Вхідні дані:

- T - поточна ситуація;
- CB - непорожня безліч прецедентів (БП);
- m - кількість розглянутих прецедентів з БП;
- S (C, T) - задана метрика (міра схожості);
- H - порогове значення ступеня подібності.

Переваги алгоритму k-NN для вилучення прецедентів:

- потрібно знати тільки функцію близькості між об'єктами;
- може бути застосований до об'єктів будь-якої структури, якщо задана функція близькості;
- проста логіка роботи і реалізація;
- розумне пояснення результатів класифікації. Недоліки алгоритму k-NN;

- з ростом БП знижується швидкодія і збільшуються вимоги по пам'яті;
- точність погіршується з ростом розмірності простору;
- проблема вибору метрики.

2.7 Повторне використання прецедентів

При повторному використанні знайденого прецеденту в контексті нової проблемної ситуації істотними є наступні моменти: відмінність між витягнутим і новим прецедентом, а також те, яку частину видобутого прецеденту можна перенести на поточну ситуацію.

У простих задачах класифікації відмінності просто ігноруються і клас рішення витягнутого прецеденту переноситися на клас рішення нового прецеденту. Це найпростіший спосіб повторного використання прецедентів.

Багато системи, однак, враховують відмінності між знайденим і наявним прецедентом, і тому рішення витягнутого прецеденту не може бути безпосередньо перенесено на нову ситуацію. Це рішення вимагає адаптації до поточної проблеми з урахуванням відмінностей між прецедентами.

Існує два методи повторного використання прецедентів: використання рішення знайденого прецеденту (трансформаційне використання) і використання методу, за допомогою якого було отримано це рішення (дериваційне використання).

При трансформаційному використанні рішення знайденого прецеденту не є безпосереднім рішенням поточної ситуації, однак існують деякі знання у вигляді трансформаційних операторів, застосування яких до старого рішення перетворює його в рішення нового прецеденту. Можна впорядкувати ці оператори, проіндексувавши їх щодо помічених відмінностей між витягнутим прецедентом і поточною проблемною ситуацією. При трансформаційному використанні увага спрямована не так на те, як вирішується проблема, а на схожість двох рішень, тому системі потрібні глибокі знання про специфіку предметної області в формі трансформаційних операторів і режим управління для організації операторів в додатку.

При дериваційному використанні прецедентів увага приділяється тому, як було знайдено рішення в витягнутому прецеденті. Знайдений прецедент містить інформацію про те, який метод використовувався при пошуку рішення, обґрунтованість використання операторів, передбачувані підцели, можливі альтернативи, вдалі і невдалі шляхи пошуку рішення і т.д. При дериваційному використанні до нової проблеми застосовується витягнутий метод і план пошуку рішення, тільки з урахуванням контексту нового завдання. При пошуку рішення вдалі гілки пошуку рішення, оператори, і шляхи вирішення застосовуються в першу чергу, а

невдалі гілки пошуку не розглядаються. Нові подцели формуються на основі старих і для їх досягнення будується новий план на основі знайденого раніше. Таким чином, дериваційне використання дозволяє знаходити рішення нової проблемної ситуації, але з мінімальними витратами часу і ресурсів.

2.8 Адаптація та застосування прецедентів

На даному етапі оцінюється рішення, отримане при повторному використанні прецеденту. Якщо рішення застосовано успішно, то формується новий прецедент, який зберігається в БП, в іншому випадку рішення виправляється з використанням знань про дану предметної області.

Оцінка рішення дається за результатами його безпосереднього застосування, або безпосередньо запитується у ЛПР. Застосування рішення зазвичай відбувається поза системою. Для отримання результатів застосування рішення може знадобитися деякий час, в залежності від предметної області, для якої будувалася система. Наприклад, в ІСППР в області медицини оцінити результат можна тільки через кілька годин або навіть місяців. Однак, незважаючи на це, прецедент може бути збережений в бібліотеці, і використовуватися в подальшому, але тільки позначений як неоцінений.

Виправлення прецедентів включає в себе вивчення помилок в поточному рішенні і знаходження причин цих помилок. Потім знайдені причини використовуються для виправлення рішення, так щоб в подальшому виключити помилки

2.9 Збереження прецедентів

Збереження прецедентів - це процес відбору інформації з вирішення нової проблемної ситуації для збереження і подальшого використання. Збереження прецедентів включає в себе відбір інформації, яку потрібно зберегти, форму в якій вона буде збережена, індексацію для майбутнього вилучення в разі виникнення схожої ситуації і те, яким чином слід включити новий прецедент в склад БП.

Якщо рішення було взято з витягнутого прецеденту, то може бути сформований новий прецедент або старий прецедент може бути узагальнено, покриваючи нову ситуацію. Якщо рішення було отримано іншими методами, наприклад, рішення знайдено самим ЛПР, то формується абсолютно новий прецедент. У будь-якому випадку має бути прийняте рішення, як поповнювати БП. Крім опису параметрів проблемної ситуації і її вирішення, до складу прецеденту може бути включено пояснення, чому дане рішення є рішенням

проблемної ситуації. У системах, в яких застосовується дериваційне використання прецедентів, зберігається метод пошуку рішення (план виведення, можливі альтернативи і т.д.).

Інформація про невдалий застосуванні отриманих рішень так само може бути збережена у вигляді окремих прецедентів. Тоді в разі невдачі, система може знайти прецедент, який зберігає інформацію про схожий невдалому випадку і використовувати її для виправлення поточної ситуації.

2.10 Етапи розробки CBR систем

Розробка CBR систем складається з наступних етапів:

- Вилучення знань для формування первинного безлічі прецедентів (збір інформації про ситуації і випадках, що мали місце в минулому) може здійснюватися на основі аналізу архіву системи і мали місце аварійних ситуацій, вивчення досвіду, накопиченого експертом (ЛПР), оперативних інструкцій, технологічного регламенту тощо.
- Визначення структури для організації БП. Структура БП безпосередньо залежить від способу представлення прецедентів і специфіки проблемної області.
- Вибір методу отримання прецедентів. Метод вилучення залежить від предметної області і типу розв'язуваних завдань, а також від способу представлення прецедентів в БП і її структури.
- Визначення способу адаптації прецедентів. Існує два типи адаптації прецедентів. Структурна адаптація застосовується для погано обумовлених прецедентів, а адаптація по відхиленню - для добре обумовлених.
- Визначення інших аспектів CBR системи, обумовлених особливостями предметної області або типом розв'язуваної задачі.

2.11 Висновок до другого розділу

1. У цьому розділі наведено основні визначення, пов'язані з прецедентним підходом (CBR методами), і концепції CBR, зокрема, те, що висновок на основі прецедентів є підходом, що дозволяє вирішити нову, невідому завдання, використовуючи або адаптуючи рішення вже відомої задачі.

2. Наведено CBR цикл, що включає в себе чотири основні етапи: вилучення найбільш відповідного (подібного) прецеденту (або прецедентів) для ситуації, що склалася з БП; повторне використання вилученого прецеденту для спроби вирішення поточної

проблеми; адаптація і застосування отриманого рішення для вирішення поточної проблеми; збереження нового прийняття рішення як частини нового прецеденту. Описано етапи і особливості розробки CBR систем (витяг знань для формування первинного безлічі прецедентів (збір інформації про ситуації і випадках, що мали місце в минулому); визначення структури для організації БП; вибір методу вилучення прецедентів; визначення способу адаптації прецедентів; визначення інших аспектів CBR системи, обумовлених особливостями предметної області або типом розв'язуваної задачі).

3. Описано ефективний алгоритми вилучення прецедентів на основі k-NN, який дозволяє підвищити якість рішення задач ІАД.

4. Розглянуто специфіку вирішення завдання класифікації даних з використанням прецедентного підходу.

2.12 Перелік використаних джерел посилань до другого розділу

1. Варшавский П.Р., Еремеев А.П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений // Искусственный интеллект и принятие решений. 2009. № 2. С. 45-47.
2. Aamodt A., Plaza E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. Artificial Intelligence Communications. IOS Press. 1994, vol. 7, no. 1, pp. 39–59.
3. Сацук О.О., Шумова Л.О. Інформаційні технології підтримки прийняття рішень на основі прецедентів // IT-Ідея – 2020: збірник науково-практичних праць, 11 груд. 2020 р., м. Сєвєродонецьк. Сєвєродонецьк : Видавництво СНУ ім. В. Даля, 2020. - С. 38-40.
4. Маннинг К.Д., Рагхаван П., Шютце Х. Введение в информационный поиск. - М.: Изд-во «Вильямс», 2011
5. S. B. Imandoust, M. Bolandraftar, 2013

РОЗДІЛ 3

РЕАЛІЗАЦІЯ МЕТОДІВ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ ПРЕЦЕДЕНТІВ

Як показав досвід розробки системи підтримки прийняття рішень (СППР) [1], використання тільки двох джерел знань, а саме: експертів і проблемно природно-мовних текстів, часто призводить до неповноти видобутих знань. В умовах придбання знань неповнота пов'язана, в основному, з тим, що експерт не знає (залишив поза увагою, або забув відзначити) будь-якої факт, необхідний для вирішення задачі. У цьому випадку можливі наступні альтернативи подолання неповноти: або проведення кількох сеансів придбання знань з одним і тим же експертом і порівняння отриманих результатів, або залучення декількох експертів і кореляція їх думок, а також використання технології вилучення знань з баз даних - Data Mining.

Одним із завдань, що вирішується Data Mining, є кластеризація даних.

Кластеризація - це групування об'єктів на основі даних, що описують властивості об'єктів.

В Data Mining для кластеризації використовуються алгоритми k-means та мережі Кохонена.

У розділі представлені результати застосування аналітичної платформи Deductor для формалізації і автоматизації процесу прийняття рішення. Для прикладу були взяті дані з загальнодоступного сховища [2], які в роботі складають базу прецедентів прямого банківського маркетингу. Реальні дані були зібрані в ході рекламної кампанії, пов'язаної з підпискою на банківський депозит. Бізнес-мета полягає в тому, щоб знайти модель, яка може пояснити успіх контакту, тобто підписку клієнта на депозит. Така модель може підвищити ефективність кампанії за рахунок визначення основних характеристик, що впливають на успіх і підтримку прийняття рішення підпишеться чи клієнта на депозит.

3.1 Постановка задачі

Є база даних прецедентів банківського маркетингу. Необхідно виявити вплив таких чинників, як вік клієнта, сімейний стан, освіту, чи є кредит в дефолті, середньорічний баланс, кількість контактів, результат попередньої маркетингової кампанії на бажану мету - підписався чи клієнт на строковий вклад?

Таким чином, входними параметрами будуть:

- вік (age, числовий);

- сімейний стан (marital, категоріальне: «одружений», «розлучений», «неодружений»); примітка: «розлучений» означає розлучений або овдовіла);
- освіту (education, категоріальне: «невідомо», «середнє», «початкова», «вища»);
- є кредит в дефолті (default, двійковий: «так», «ні»);
- середньорічний баланс в євро (balance, числовий);
- є житловий кредит (housing, двійковий: «так», «ні»);
- чи є особистий кредит (loan, двійковий: «так», «ні»);
- кількість контактів, виконаних під час цієї кампанії і для цього клієнта (campaign, числове, включаючи останній контакт);
- кількість днів, що пройшли після того, як з клієнтом в останній раз зв'язалися з попередньої кампанії (pdays, число, -1 означає, що з клієнтом які раніше не зв'язалися);
- результат попередньої маркетингової кампанії (poutcome, категоріальні: «невідомо», «інше», «невдача», «успіх»).

Вихідний параметр - бажана мета: підписався чи клієнт на строковий вклад? (Двійковий: «так», «ні»).

3.2 Опис можливостей АП Deductor

3.2.1 Дерева рішень

Дерево рішень - це логічний алгоритм класифікації, заснований на пошуку кон'юнктивні закономірностей [3].

Деревом називається кінцевий зв'язний граф з безліччю вершин V , що не містить циклів і має виділену вершину $v_0 \in V$, в яку не входить жодне ребро. Ця вершина називається коренем дерева. Вершина не має виходять ребер, називається термінальної або листом. Решта вершини називаються внутрішніми. Дерево називається бінарним, якщо з будь-якої його внутрішньої вершини виходить рівно два ребра. Вихідні ребра пов'язують кожну внутрішню вершину v з лівої дочірньої вершиною Lv і з правої дочірньої вершиною Rv .

Бінарне вирішальне дерево - це алгоритм класифікації, задати бінарним деревом, в якому кожної внутрішньої вершині $v \in V$ приписаний предикат $\beta v: X \rightarrow \{0,1\}$, кожної термінальної вершини $v \in V$ приписано ім'я класу $cv \in Y$.

Об'єкт x доходить до вершини v тоді і тільки тоді, коли виконується кон'юнкція $Kv(x)$, складена з усіх предикатів, приписаних внутрішнім вершин дерева на шляху від кореня v_0 до вершини v . Нехай T - безліч всіх термінальних вершин дерева. Безлічі об'єктів $\Omega v = \{x$

$\in X: K_v(x) = 1\}$, що виділяються термінальними кон'юнкції $v \in T$, попарно не перетинаються, а їх об'єднання збігається з усім простором X (це легко доводиться індукцією по числу вершин дерева). Звідси випливає, що алгоритм класифікації $a: X \rightarrow Y$, реалізований бінарним вирішальним деревом, можна представити у вигляді простого голосування кон'юнкція:

$$a(x) = \arg \max_{y \in Y} \sum_{v \in Y} [cv = y] K_v(x), \quad (3.1)$$

Причому для будь-якого $x \in X$ одне і тільки одне доданок у всіх цих сумах дорівнює одиниці. Замість підсумовування можна було б використовувати і диз'юнкцію.

Природна вимога максимізації інформативності кон'юнкції $K_v(x)$ означає, що кожна з них повинна виділяти якомога більше навчальних об'єктів, допускаючи при цьому якомога менше помилок. Для підвищення узагальнюючої здатності вирішального дерева число листя має бути якомога менше, і вони повинні покривати підвибірки приблизно однакової потужності $|\Omega_v \cap X^A|$.

В Deductor Studio подібний клас завдань реалізується за допомогою найбільш популярного алгоритму C4.5., В якому кількість нащадків у вузла не обмежена. Не вміє працювати з безперервним цільовим полем, тому вирішує тільки завдання класифікації.

Спочатку імпортуємо дані з файлу. Запустимо майстер обробки і виберемо дерево рішень. Виберемо в якості обробки дерево рішень. У Майстрі побудови дерева рішення на другому етапі налаштуємо поля. Віддам полях age, balance, campaign, previous і poutcome значення вхідних, полю у вихідний, а іншим - інформаційне (рис. 3.1).

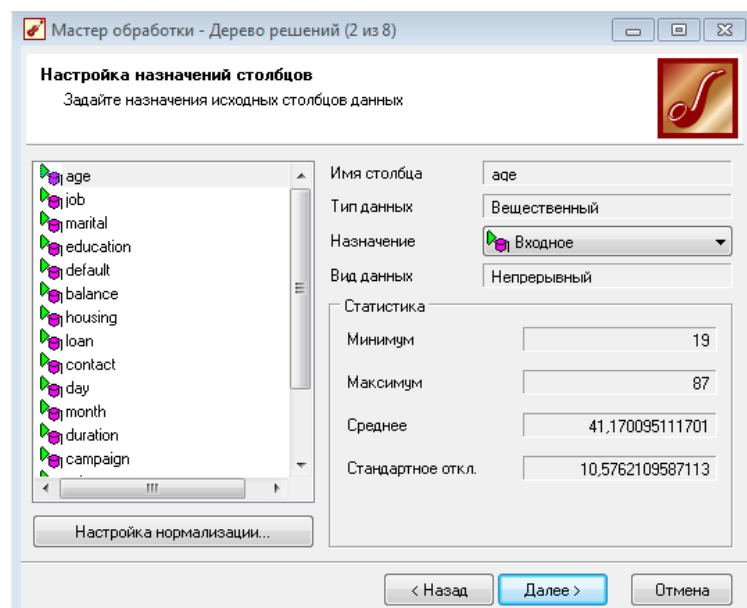


Рисунок 3.1 - Налаштування значень стовпців

Далі пропонується налаштувати спосіб розбиття вихідного безлічі даних на навчальне і тестове. Задамо випадковий спосіб розбиття, коли дані для тестового та навчальної множини беруться з вихідного набору випадковим чином. На наступному кроці Майстра пропонується налаштувати параметри процесу навчання, а саме мінімальну кількість прикладів, при якому буде створений новий вузол (нехай вузол створюється, якщо в нього потрапили два і більше прикладів), а також пропонується можливість будувати дерево з більш достовірними правилами. Включимо дані опції (рис. 3.2).

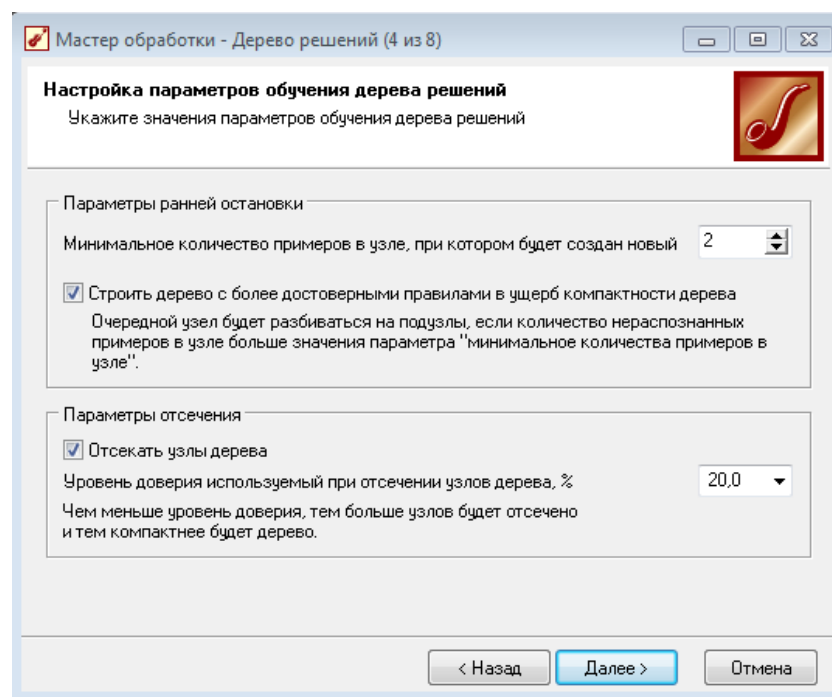


Рисунок 3.2 - Налаштування параметрів навчання

На наступному кроці Майстра запускається сам процес побудови дерева. Також можна побачити інформацію про кількість розпізнаних прикладів. Після побудови дерева можна побачити, що велика частина прикладів і на навчальній і на тестовій вибірці розпізнані (рис. 3.3).

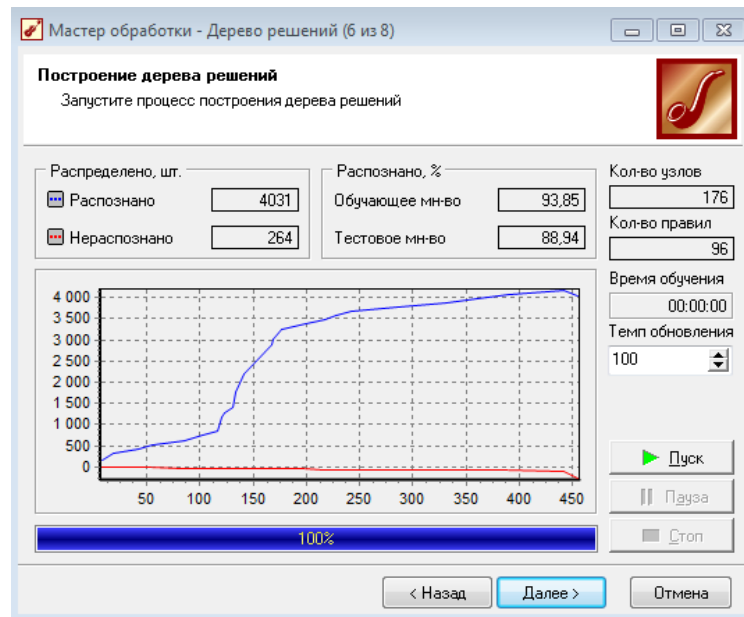


Рисунок 3.3 - Построение дерева решений

Проаналізуємо дані за допомогою наявних візуалізаторів. Для початку подивимося на таблиці спряженості (рис 3.4).

| Фактически | Классифицировано | | | Итого |
|------------|------------------|-----|-------|-------|
| | no | yes | Итого | |
| no | 3958 | 42 | 4000 | |
| yes | 436 | 85 | 521 | |
| Итого | 4394 | 127 | 4521 | |

| Фактически | Классифицировано | | | Итого |
|------------|------------------|-----|-------|-------|
| | no | yes | Итого | |
| no | 191 | 3 | 194 | |
| yes | 26 | 6 | 32 | |
| Итого | 217 | 9 | 226 | |

Рисунок 3.4 - Таблиці спряженості дерева рішень

По діагоналі таблиці розташовані приклади, які були правильно розпізнані, в інших осередках - ті, які були віднесені до іншого класу. В даному випадку дерево правильно класифікувало істотну частину прикладів.

3.2.2 Карти Кохонена

Мережі, звані картами Кохонена, - це одна з різновидів нейронних мереж, які використовують неконтрольоване навчання. При такому навчанні навчальна множина складається лише з значень вхідних змінних, в процесі навчання немає порівнювання виходів нейронів з еталонними значеннями. Можна сказати, що така мережа вчиться розуміти структуру даних. Найбільш поширене застосування мереж Кохонена - рішення задачі класифікації без вчителя, тобто кластеризації. При такій постановці завдання нам дано набір об'єктів, кожному з яких порівняна рядок таблиці (вектор значень ознак). Потрібно розбити

вихідне безліч на класи, тобто для кожного об'єкта знайти клас, до якого він належить. В результаті отримання нової інформації про класи можлива корекція існуючих правил класифікації об'єктів.

Мережі Кохонена призначені для вирішення завдань кластеризації. Постановка завдання кластеризації. Дано: $X [n \times m]$, n - кількість об'єктів, m - кількість ознак; або Дано: безліч векторів $X = \{X^1, X^2, \dots, X^p, \dots, X^n\}$, где $X^p = \{X_1^p, \dots, X_m^p\}$.

Заздалегідь відомо кількість k майбутніх кластерів, але саме розбиття на кластери не відомо.

Необхідно: 1) знайти k -ядер (Центроїд) кластерів, тобто вектор $C = \{C^1, C^2, \dots, C^p, \dots, C^n\}$, де $C^p = \{C_1^p, \dots, C_m^p\}$;

2) розбити безліч векторів X на k кластерів: $X^p = \{X_1^p, \dots, X_m^p\}$ на $K\{C^1\}$. Іншими словами, завдання полягає в тому, щоб знайти деяку функцію $l(p)$, яка дозволяє визначити номер кластера l за номером вхідного об'єкта P , що l є рішенням задачі кластеризації. При цьому ця функція повинна задовольняти наступним критерієм: мінімізації відстаней між об'єктами і ядрами кластерів.

3.3 Алгоритм кластеризації

Насамперед задаються початкові значення ядер кластерів $\{C^1\}$.

Способи завдання: а) випадковими числами; б) деякими рівними числами; в) евристичними правилами (метод головних компонент), заснованими на деякій закономірності.

Потім фіксуються ядра кластерів $\{C^1\} = const$. Знаходимо $l(p)$ - розбиття $\{X^p\}$ із ціллю $D \rightarrow \min$.

$\{C^1\}$ коригується таким чином, щоб в межах кожного кластера сумарне відстань між об'єктом цього кластера і ядром було мінімальним[4].

3.4 Алгоритм навчання без вчителя для мереж Кохонена

Труднощі при використанні алгоритму це вибір вихідних центрів кластерів і необхідність знати число кластерів заздалегідь. 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.

Самоорганізована карта Кохонена будується на основі нейромережі Кохонена.

Мережа Кохонена є однією з різновидів змагальної (конкурентної) нейронної мережі з навчанням без учителя. Її основною метою є перетворення складних багатовимірних даних в більш просту структуру малої розмірності. Мережа складається з вузлів, які об'єднуються в кластери.

Сигнал в мережу Кохонена надходить відразу на всі нейрони, ваги відповідних синапсів інтерпретуються як координати положення вузла, і вихідний сигнал формується за принципом «переможець забирає все» - тобто ненульовий вихідний сигнал має нейрон, найближчий (у сенсі ваг синапсів) до подається на вхід об'єкту. В процесі навчання ваги синапсів налаштовуються таким чином, щоб вузли решітки «розташовувалися» в місцях локальних згущень даних, тобто описували кластерну структуру хмари даних; з іншого боку - зв'язку між нейронами відповідають відносинам сусідства між відповідними кластерами в просторі ознак.

Карта Кохонена складається з осередків, центром кожної з яких є нейрон вихідного шару. У осередок потрапляє один або кілька об'єктів, вектори ваг яких виявляються ближче до вектору ваг даного нейрона.

Спочатку самоорганізована карта являє сітку з вузлів, з'єднаних між собою. Кохонен розглядав два варіанти з'єднання вузлів - в прямокутну і гексагональну сітку. Відмінність полягає в тому, що в прямокутній сітці кожен вузол з'єднаний з чотирма сусідніми, а в гексагональній - з шістьма найближчими вузлами. Шестикутні осередки більш коректно відображають відстані між об'єктами на карті, тому що у них рівні відстані між центрами суміжних осередків.

Самоорганізована карта Кохонена є методом проектування багатовимірного простору в простір з більш низькою розмірністю (найчастіше двомірне). Двомірна карта має кольорову розмальовку. Інтенсивність кольору в певній точці залежить від даних, які туди потрапили. Осередки, в які потрапили елементи з мінімальними значеннями або не потрапило взагалі жодного запису, фарбуються синім кольором, а осередки з максимальними значеннями фарбуються червоним. Розфарбування карти дозволяє оцінити і результати кластеризації. Якщо осередки з однаковою забарвленням утворюють відособлені області, то результати кластеризації хороші. Якщо осередки різних кольорів розкидані упереміш по всій карті, то результати погані.

Недоліком карти Кохонена є евристичний характер методу. Початкова ініціалізація карти, тобто завдання ваг нейронів, є довільною, що може привести до втрати однозначності результату. Якщо навчати карту Кохонена кілька разів, то виходять несхожі підсумки.

Кластери відображають групи векторів, відстань між якими менше, ніж відстань до сусідніх груп. Іншими словами, всі елементи карти, що потрапили в область одного кольору (кластер), мають схожі ознаки.

Результати кластеризації алгоритмом Кохонена можна побачити не тільки на карті, але і на спеціальному візуалізаторі Профілі кластерів. Це кросстаблиця з двома вимірами - кластери і поля.

3.5 Використання карт Кохонена для вирішення задачі класифікації даних

В Deductor Studio мережі і карти Кохонена реалізовані в обробнику Карта Кохонена, де містяться сам алгоритм Кохонена і спеціальний візуалізатор Карта Кохонена.

Алгоритм Кохонена застосовується до мережі Кохонена, що складається з осередків, упорядкованих на площині. За замовчуванням розмір карти дорівнює 16 x 12, що відповідає 192 осередків (нейронам).

Не існує строгих правил для вибору числа нейронів в карті Кохонена. Кожен раз аналітик повинен сам визначати (можливо, методом проб і помилок) оптимальна кількість нейронів, виходячи з розв'язуваної задачі і особливостей даних. Наприклад, якщо розкид значень ознак в навчальній вибірці сильний (тобто вектори об'єктів в просторі ознак розріджені), то, можливо, слід зменшити число нейронів карти, щоб уникнути великої кількості порожніх клітинок. І навпаки, якщо вектори об'єктів розташовані в просторі ознак щільно, то для отримання кращих результатів можна збільшити число нейронів.

В Deductor Studio алгоритм Кохонена орієнтований на роботу переважно з числовими типами даних, а також з впорядкованими (ординальними) типами. Обробка даних в полях, значення яких не можна впорядкувати, буде приводити до некоректних результатів. Упорядкування ординальних типів здійснюється на вкладці Налаштування нормалізації.

У обробнику «карта Кохонена» задається і вихідна поле. Це поле не братиме участь в алгоритмі Кохонена, однак після побудови по цьому полю буде зібрана статистика. Саме це і відкриває можливості для вирішення картою Кохонена завдання класифікації або регресії.

Запускаємо майстер обробки і вибираємо зі списку метод обробки Карта Кохонена. Далі слід налаштувати призначення стовпців, тобто для кожного стовпця вибрати одне з призначень: вхідний, вихідний, не використовується і інформаційне.

В якості вхідних параметрів використовуються:

- age;
- balance;
- campaign;

- previous;
- poutcome;

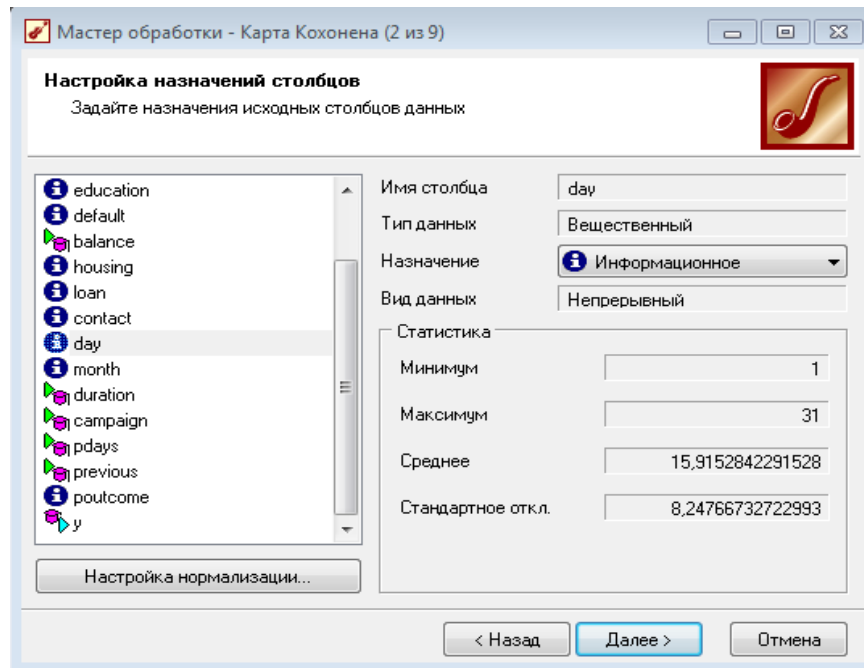


Рисунок 3.5 - Налаштування значень стовпців

На цій же вкладці при натисканні кнопки «Налаштування нормалізації» відкриється вікно, де можна задати значимість кожного вхідного поля. Залишимо значимість однаковою для всіх полів без змін.

Потім розбиваємо входить множину на навчальне і тестове. Оскільки будь-який метод кластеризації, в тому числі і алгоритм Кохонена, суб'єктивний, сенс у виділенні окремої, тестової множини, як правило, відсутня. Але ми залишимо в тестовій множині 5% записів.

На третій вкладці задаються розмір і форма карти Кохонена. Погодимося з настройками за замовчуванням - шестикутні осередки, розмір 16x12.

Зазвичай будують кілька моделей з різними настройками. Наприклад, карта зі збільшеним масштабом 24x18 може виявитися краще, так як дозволить «розгледіти» кластер, який не вдавалося виявити при розмірі карти 16x12. Тому тут універсальних рецептів немає. Зрозуміти, краще або гірше карта Кохонена, можна, лише порівнявши її з картами при інших налаштуваннях, порівнявши матриці помилок квантування і матриці щільності попадання.

На кроці «Налаштування параметрів зупинки навчання» встановлюємо параметри зупинки навчання і встановлюємо епоху (за замовчуванням - 500), по досягненню якої навчання буде припинено.

На наступному кроці, представленому на рис. 2, налаштовуються інші параметри навчання: спосіб початкової ініціалізації, тип функції сусідства.

Спосіб початкової ініціалізації карти визначає, як будуть встановлені початкові ваги нейронів карти. Вдало обраний спосіб ініціалізації може істотно прискорити навчання і привести до отримання більш якісних результатів.

Доступні три варіанти:

- випадковими значеннями - початкові ваги нейронів будуть ініційовані випадковими значеннями;
- з навчальної множини - в якості початкових ваг будуть використовуватися випадкові приклади з навчальної множини;
- з власних векторів - початкові ваги нейронів карти будуть проініціалізовані значеннями підмножини гіперплощини, через яку проходять два головних власних вектора матриці коваріації вхідних значень навчальної вибірки.

Між зазначеними параметрами і способом початкової ініціалізації існує багато залежностей. Виділимо кілька головних.

1. Якщо обсяг навчальної вибірки значно (в 100 і більше) перевищує число осередків карти і час навчання не грає першочерговим ролі, то краще вибрати ініціалізацію випадковими значеннями, так як це дасть меншу ймовірність попадання в локальний мінімум помилки кластеризації.

2. Якщо обсяг навчальної вибірки не дуже великий, час навчання обмежена або необхідно зменшити ймовірність появи після навчання порожніх клітинок, в які не потрапило жодного примірника навчальної вибірки, то слід використовувати ініціалізацію прикладами з навчальної множини.

3. Ініціалізацію з власних векторів можна використовувати при будь-якому збігу обставин. Єдине зауваження: ймовірність появи порожніх клітинок після навчання вище, ніж при ініціалізації прикладами з навчальної множини. Саме цей спосіб краще вибирати при першому ознайомленні з даними.

Швидкість навчання - задається швидкість навчання на початку і в кінці навчання карти Кохонена. Рекомендовані значення: 0,1-0,3 на початку і 0,05 0,005 в кінці навчання.

Радіус навчання - задається радіус навчання на початку і в кінці навчання карти Кохонена. Радіус на початку повинен бути досить великий - приблизно половина або менше розміру карти (максимальне лінійне відстань від будь-якого нейрона до іншого будь-якого нейрона), а в кінці - досить малим, приблизно 1 або менше. Початковий радіус в Deductor підбирається автоматично в залежності від розміру карти.

У цьому ж блоці задається Функція сусідства: Гауссова або Ступенева. Якщо функція сусідства Ступенчатая, то «сусідами» для нейрона-переможця будуть вважатися все нейрони, лінійне відстань до яких не більше поточного радіуса навчання. Якщо використовується Гауссова функція сусідства, то «сусідами» для нейрона-переможця будуть вважатися все нейрони карти, але в різному ступені повноти. При використанні Функція Гауса сусідства навчання проходить більш плавно і рівномірно, так як одночасно змінюються ваги всіх нейронів, що може дати трохи кращий результат, ніж якби використовувалася ступінчаста функція. Однак часу, необхідного на навчання, потрібно трохи більше через те, що на кожній епосі коригуються всі нейрони.

Кластеризація - в цій галузі вказуються параметри алгоритму kmeans (G-means), який запускається після алгоритму Кохонена для угруповання осередків карти. Тут потрібно тільки визначити, дозволити алгоритму автоматично визначити число кластерів (G-means) або відразу зафіксувати його (k-means). Слід знати, що автоматично підбирається число кластерів не завжди приводить до бажаного результату - число кластерів може пропонуватися занадто великим, тому розраховувати на цю опцію можна тільки на етапі дослідження даних.

Оскільки нам відомо кількість кластерів, виберемо 2.

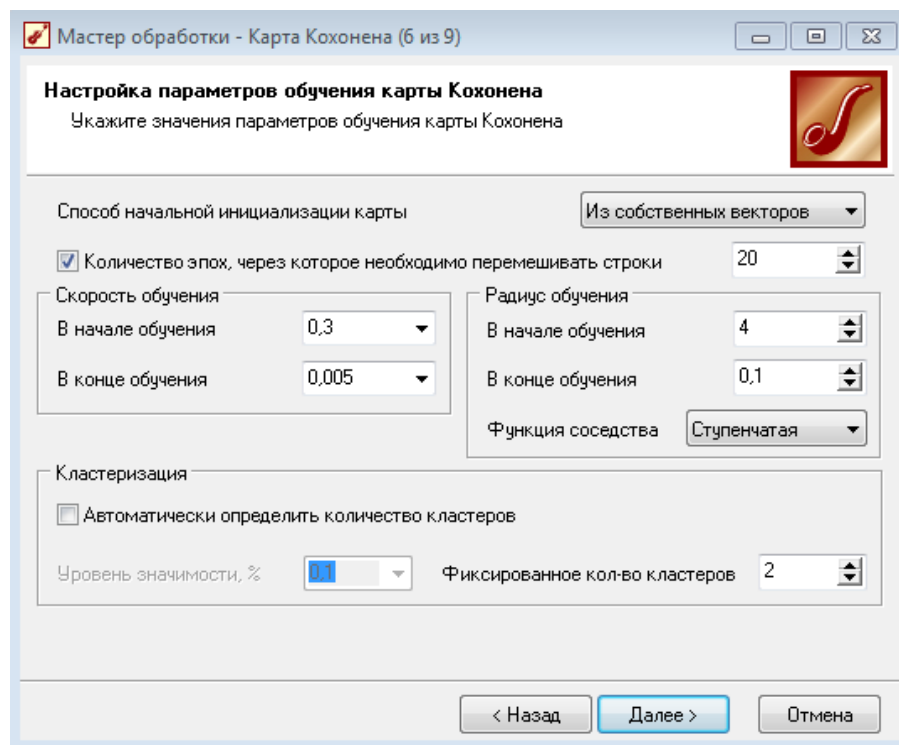


Рисунок 3.6 - Налаштування параметрів навчання

Після закінчення введення параметрів запускаємо процес навчання-необхідно натиснути на кнопку Пуск і дочекатися закінчення процесу навчання. У вікні можна буде побачити динаміку процесу навчання карти Кохонена. За замовчуванням алгоритм робить 500 ітерацій (epoch). Якщо попередньо встановити прапор Рестарт, то ваги нейронів будуть проініціалізовані відповідно до обраного на попередньому кроці способу ініціалізації, інакше навчання почнеться з поточних вагових коефіцієнтів (це справедливо тільки при повторній налаштування вузла).

Після закінчення навчання в списку способів відображення даних виберемо «Карта Кохонена», «Профілі кластерів» і візуалізатор «чтоеслі». На останньому кроці налаштуємо відображення карти Кохонена.

На рис. 3 приведена ілюстрація карт входів і виходів, остання - це карта кластерів. Тут ми бачимо кілька карт входів і сформовані кластери, кожен з яких виділено окремим кольором.

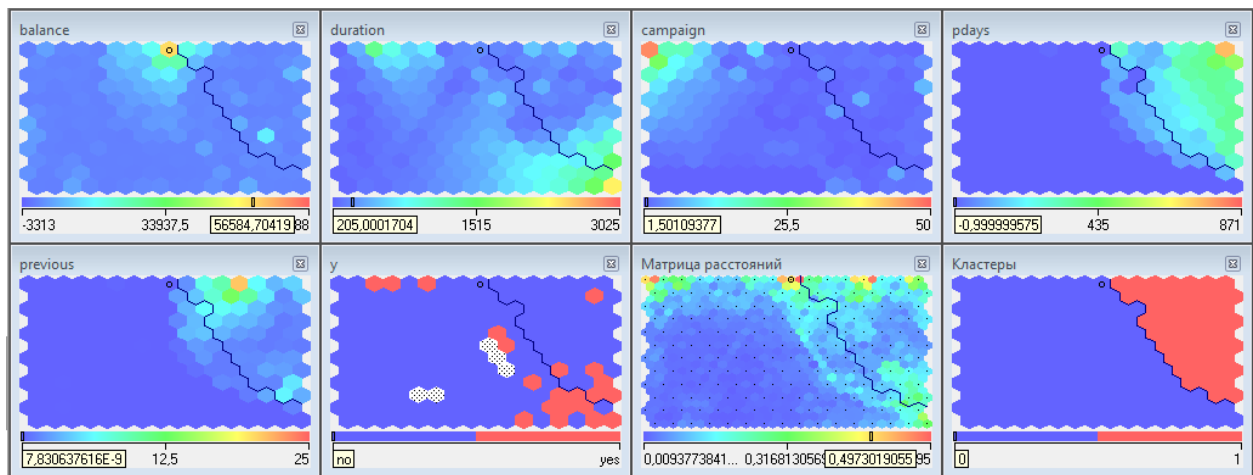


Рисунок 3.7 - Карти входів і виходів

Останнім кроком проведемо аналіз результатів, аналогічний попередньому. У майстра візуалізації вибираємо Таблиця спряженості (рис. 3.8).

| Фактически | Классифицировано | | |
|------------|------------------|-----|-------|
| | no | yes | Итого |
| no | 3898 | 102 | 4000 |
| yes | 371 | 150 | 521 |
| Итого | 4269 | 252 | 4521 |

| Фактически | Классифицировано | | |
|------------|------------------|-----|-------|
| | no | yes | Итого |
| no | 206 | | 206 |
| yes | 16 | 4 | 20 |
| Итого | 222 | 4 | 226 |

Рисунок 3.8 - Таблиці спряженості мережі Кохонена

3.6 Аналіз отриманих результатів

В результаті рішення задачі отримано 2 кластера.

При аналізі карт входів рекомендують використовувати відразу декілька карт. Досліджуємо фрагмент карти, що складається з карт п'яти входів, який наведено на рис. 3.

На карті balance виділяємо область з найбільшим значенням показника. Далі має сенс вивчити ці ж осередки на інших картах. Вони розташовані також у верхній частині і відносяться до 0 кластеру.

Також ми можемо визначити, наприклад, таку характеристику: кластер, розташований в правій частині, характеризується високими значеннями виконаних контактів попередньої кампанії.

Для знаходження конкретного об'єкта на карті необхідно натиснути правою кнопкою миші на досліджуваному об'єкті і вибрати пункт «Знайти осередок на карті». В результаті ми можемо бачити як сам об'єкт, так і значення того виміру, яке ми переглядаємо. Таким чином, ми можемо оцінити стан аналізованого об'єкта, а також порівняти його з іншими об'єктами.

Дані по банківській статистиці були класифіковані на 2 групи, для кожної з яких можливо визначення конкретних характеристик, виходячи з розмальовки відповідних показників.

Таким чином, карти Кохонена дозволяють спростити багатовимірну структуру, їх можна вважати одним з методів проектування багатовимірного простору в простір з більш низькою розмірністю. Інтенсивність кольору в певній точці карти визначається даними, які туди потрапили: осередки з мінімальними значеннями зображуються темно-синім кольором (в чорно-білій палітрі - чорним), осередки з максимальними значеннями - червоним (в чорно-білій палітрі - білим).

Інша принципова відмінність карт Кохонена від інших моделей нейронних мереж - інший підхід до навчання, а саме - некероване або неконтрольоване навчання. Цей тип навчання дозволяє даними навчальної вибірки містити значення тільки вхідних змінних. Мережа Кохонена вчиться розуміти саму структуру даних і вирішує завдання кластеризації.

Таблиця спряженості дерева рішень (рис. 3.9):

| Фактически | Классифицировано | | |
|------------|------------------|-----|-------|
| | no | yes | Итого |
| no | 3958 | 42 | 4000 |
| yes | 436 | 85 | 521 |
| Итого | 4394 | 127 | 4521 |

| Фактически | Классифицировано | | |
|------------|------------------|-----|-------|
| | no | yes | Итого |
| no | 191 | 3 | 194 |
| yes | 26 | 6 | 32 |
| Итого | 217 | 9 | 226 |

Рисунок 3.9 - Таблица спряженості дерева рішень

Таблиця спряженості мережі Кохонена (рис. 3.10):

| Фактически | Классифицировано | | |
|------------|------------------|-----|-------|
| | no | yes | Итого |
| no | 3898 | 102 | 4000 |
| yes | 371 | 150 | 521 |
| Итого | 4269 | 252 | 4521 |

| Фактически | Классифицировано | | |
|------------|------------------|-----|-------|
| | no | yes | Итого |
| no | 206 | | 206 |
| yes | 16 | 4 | 20 |
| Итого | 222 | 4 | 226 |

Рисунок 3.10 - Таблиця спряженості мережі Кохонена

Про ефективність методу ми можемо судити за таблицями пов'язаності для навчальної вибірки. По діагоналі таблиці розташовані приклади, які були правильно розпізнані, в інших осередках - ті, які були віднесені до іншого класу, тобто помилки. За ними можна робити оцінку якості моделі класифікації за допомогою порівняння реальних даних і результатів отриманих за допомогою моделі класифікації

У порівнянні з таблицею спряженості дерева рішень, в таблиці мережі Кохонена більше правильно класифікованих осередків (4048, коли як в дереві рішень 4043). З цього можна зробити висновок, що карти Кохонена більш ефективні для навчальної вибірки, ніж дерева рішень.

3.7 Висновок до третього розділу

Проведені експерименти моделювання в аналітичній платформі Deductor двома засобами: карти Кохонена і дерева рішень для конкретної задачі.

Виконано оцінку якості моделі класифікації за допомогою порівняння реальних даних і результатів отриманих за допомогою моделі класифікації. Отримали візуалізований результат роботи алгоритмів. Кожен метод має свої переваги і недоліки.

3.8 Перелік використаних джерел посилань до третього розділу

1. Варшавский П.Р., Еремеев А.П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений // Искусственный интеллект и принятие решений. 2009. № 2. С. 45-47.
2. <http://hdl.handle.net/1822/14838>
3. И.Н. Шафигуллина Интеллектуальный анализ данных с использованием системы DEDUCTOR - КПФУ, 2014

4. Н. Ю. Прокопенко Системы поддержки принятия решений на базе Deductor Studio Academic 5.3 - ННГАСУ, 2017

РОЗДІЛ 4

ОХОРОНА ПРАЦІ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

4.1 Аналіз стану умов праці

4.1.1 Вимоги до приміщення

Таблиця 4.1 - Розміри приміщення.

| Найменування | Значення |
|-------------------------|----------|
| Довжина, м | 18 |
| Ширина, м | 6 |
| Висота, м | 3,5 |
| Площа, м ² | 108 |
| Об'єм, м ³ | 378 |
| Кількість робочих місць | 8 |

Згідно з ДСН 3.3.6.042-99 [5] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

Для забезпечення потрібного рівного освітленості кімната має вікна та систему загального рівномірного освітлення, що встановлена на стелі. Для дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

4.1.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за ДСанПіН 3.3.2.007-98 [5](табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 - Характеристики робочого місця

| Найменування параметра | Фактичне Значення | Нормативне Значення |
|--|-------------------|---------------------|
| Висота робочої поверхні, мм | 700 | 680-800 |
| Висота простору для ніг, мм | 650 | не менше 600 |
| Ширина простору для ніг, мм | 540 | не менше 500 |
| Глибина простору для ніг, мм | 660 | не менше 650 |
| Висота поверхні сидіння, мм | 420 | 400-500 |
| Ширина сидіння, мм | 410 | не менше 400 |
| Глибина сидіння, мм | 420 | не менше 400 |
| Висота поверхні спинки, мм | 500 | не менше 300 |
| Ширина опорної поверхні спинки, мм | 400 | не менше 380 |
| Радіус кривини спинки в горизонтальній площині, мм | 400 | 400 |
| Відстань від очей до екрану дисплея, мм | 750 | 800 |

4.2 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

4.2.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00-7.15-18 [8], яке встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої. Основними робочими характеристиками персонального комп'ютера є:

– робоча напруга $U=+220V \pm 5\%$;

- робочий струм $I=2A$;
- споживана потужність $P=350$ Вт.

Таблиця 4.3 – Аналіз небезпечних і шкідливих виробничих факторів

| Небезпечні і шкідливі виробничі фактори | Джерела факторів (види робіт) | Кількісна Оцінка | Нормативні Документи |
|---|--|------------------|----------------------|
| Фізичні: | | | |
| підвищена або знижена вологість повітря | -//- | 3 | [4] |
| підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини | -//- | 3 | [6] [7] |
| Психофізіологічні: | | | |
| нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових) | - формулювання теми; - пошук інформації про предметну область; | 4 | [6] [7] |
| | - проектування структур та алгоритмів; - виконання роботи; - оформлення записки. | | |
| фізичні (статичне - сидіння) | порушення умов організації робочого часу (безперервна робота) | 2 | [4] [5] |

Робочі місця в обов'язковому порядку повинні відповідати вимогам до санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, що затверджені постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [5].

4.2.2 Пожежна безпека

Висока щільність елементів в електронних схемах призводить до значного підвищення температури окремих вузлів (80...100 °C). При проходженні електричного струму по провідниках і деталей виділяється тепло, що в умовах їх високої щільності може привести до перегріву, і може служити причиною запалювання ізоляційних матеріалів. Слабкий опір ізоляційних матеріалів дії температури може викликати порушення ізоляції і привести до короткого замикання між струмоведучими частинами обладнання (шини, електроди).

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), надійно

захищені діелектричними щитками та/або сітками з метою недопущення потрапляння працівника під напругу.

В приміщенні наявна затверджена «План-схема евакуації з кабінету (приміщення)».

Горючими матеріалами в приміщенні, де розташовані ЕОМ, є:

- 1) поліамід - матеріал корпусу мікросхем, горюча речовина, температура samozаймання 420 °C;
- 2) полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 °C, температура samozаймання 530 °C;
- 3) склотекстоліт ДЦ - матеріал друкарських плат, важкогорючий матеріал, показник горючості 1.74, не схильний до температурного samozаймання;
- 4) пластикат кабельний №489 - матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1;
- 5) деревина - будівельний і обробний матеріал, з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, температура запалювання 255 °C, температура samozаймання 399 °C.

Простори усередині приміщень в межах, яких можуть утворюватися або знаходитися пожежонебезпечні речовини і матеріали відповідно до ДСТУ Б В.1.1-36:2016 [10] відносяться до пожежонебезпечної зони класу П-Па. Це обумовлено тим, що в приміщенні знаходяться тверді горючі та важкозаймісті речовини та матеріали. Приміщенню, у якому розташоване робоче місце, присвоюється II ступень вогнестійкості.

Причинами можливого загоряння і пожежі можуть бути:

- 1) несправність електроустановки;
- 2) конструктивні недоліки устаткування;
- 3) коротке замикання в електричних мережах;
- 4) запалювання горючих матеріалів, що знаходяться в безпосередній близькості від електроустановки.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол [9].

4.2.3 Електробезпека

Виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

4.3 Гігієнічні вимоги до параметрів виробничого середовища

4.3.1 Мікроклімат

Мікроклімат робочих приміщень - це клімат внутрішнього середовища цих приміщень, що визначається діючою на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. В даному приміщенні проводяться роботи, що виконуються сидячи і не потребують динамічного фізичного напруження, то для нього відповідає категорія робіт 1а. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають ДСН 3.3.6.042-99 [4] і наведені в табл. 4.4:

Таблиця 4.4 – Норми мікроклімату робочої зони об'єкту

| Період Року | Категорія Робіт | Температура С ⁰ | Відносна вологість % | Швидкість руху повітря, м/с |
|-------------|-----------------|----------------------------|----------------------|-----------------------------|
| Холодна | Легка-1а | 22-24 | 40-60 | 0,1 |
| Тепла | Легка-1а | 23-25 | 40-60 | 0,1 |

4.3.2 Освітлення

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше $1/8$, в побутових - $1/10$:

$$S_b = \left(\frac{1}{5} / \frac{1}{10} \right) * S_n \quad (4.1)$$

де S_b – площа віконних прорізів, m^2 ;

S_n – площа підлоги, m^2 .

$$S_n = a \cdot b = 18 \cdot 6 = 108 \text{ м}^2,$$

$$S = 1/10 \cdot 108 = 10,8 \text{ м}^2.$$

Приймаємо 3 вікна площею $S=3,6 \text{ м}^2$.

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 18 м, ширина 6 м, світильниками ЛПО2П, оснащеними лампою типа ЛБ (одна - 80 Вт) з світловим потоком 5400 лм. Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників n виробляється по формулі (4.2):

$$n = \left(\frac{E * S * Z * K}{F * U * M} \right) \quad (4.2)$$

де E - нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S - освітлювана площа, m^2 ; $S = 108 \text{ м}^2$;

Z - поправочний коефіцієнт світильника ($Z = 1,15$ для ламп розжарювання та ДРЛ; $Z = 1,1$ для люмінесцентних ламп) приймаємо рівним 1,1;

K - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U - коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п.

- 0,575 M - число люмінесцентних ламп в світильнику - 1;

F - світловий потік лампи - 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \left(\frac{300 * 108 * 1.15 * 1.5}{5400 * 0.575 *} \right) = 18,0$$

Приймаємо освітлювальну установку, яка складається з 18 світильників, оснащених лампами типа ЛБ (одна - 80 Вт) зі світловим потоком 5400 лм.

4.4 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при V приміщення > 40 м³ на одного працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП. Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

1) Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціонування повітря:
 - а) якщо об'єм приміщення 20 м³, то потрібно подати не менш як 30 м³/год повітря;
 - б) якщо об'єм приміщення у межах від 20 до 40 м³, то потрібно подати не менш як 20 м³/год повітря;

в) якщо об'єм приміщення становить понад 40 м^3 , допускається природна вентиляція, у випадку, коли немає виділення шкідливих речовин.

2) Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

- постійно стежити за справним станом електромережі;
- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;
- не тягнути за мережевий кабель, щоб витягти вилку з розетки;
- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;
- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;
- не залишати включені електроприлади без нагляду;

4.6 Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом [6], приміщення в якому проводяться всі роботи відноситься до першого класу (без підвищеної небезпеки). Під час роботи використовуються електроустановки з напругою живлення 36 В, 220 В, та 360 В. Опір контуру заземлення повинен мати не більше 4 Ом.

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів $R_{\text{шт.з.}}$:

$$R_{\text{шт.з.}} = \left(\frac{R_{\text{д}} * R_{\text{пр.з}}}{R_{\text{пр.з}} - R_{\text{д}}} \right) \quad (4.3)$$

де $R_{\text{пр.з.}}$ - опір природних заземлювачів;

$R_{\text{д}}$ - допустимий опір заземлення.

Якщо природні заземлювачі відсутні, то $R_{\text{шт.з.}} = R_{\text{д}}$.

Підставивши числові значення у формулу (4.3), отримуємо:

$$R_{\text{шт.з.}} = \left(\frac{4 * 40}{40 - 4} \right) \approx 40 \text{ м}$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ , Ом·м.

Приблизне значення питомого опору глини приймаємо $\rho=40 \text{ Ом}\cdot\text{м}$ (табличне значення).

3) Розрахунковий питомий опір ґрунту, $R_{\text{розр}}$, $\text{Ом}\cdot\text{м}$, визначається відповідно для вертикальних заземлювачів $R_{\text{розр.в}}$, і горизонтальних $R_{\text{розр.г}}$, $\text{Ом}\cdot\text{м}$ за формулою:

$$R_{\text{розр.}} = \psi * \rho \quad (4.4)$$

де ψ - коефіцієнт сезонності для вертикальних заземлювачів I кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $R_{\text{розр.в}}=1,7$ і горизонтальних $R_{\text{розр.г}}=5,5 \text{ Ом}\cdot\text{м}$

$$R_{\text{розр.в}} = 1,7 * 40 = 68 \text{ Ом}\cdot\text{м}$$

$$R_{\text{розр.г}} = 5,5 * 40 = 220 \text{ Ом}\cdot\text{м}$$

4) Розраховується опір розтікання струму вертикального заземлювача R_B , Ом , за (4.5).

$$R_B = \left(\frac{R_{\text{розр.в}}}{2 * \pi * l_B} \right) * \left(\ln \frac{2 * l_B}{d_{\text{ст}}} + \frac{1}{2} * \ln \frac{4 * t + l_B}{4 * t - l_B} \right) \quad (4.5)$$

де l_B - довжина вертикального заземлювача (для труб - 2–3 м; $l_B=3$ м);

$d_{\text{ст}}$ - діаметр стержня (для труб - 0,03–0,05 м; $d_{\text{ст}}=0,05$ м);

t - відстань від поверхні землі до середини заземлювача, яка визначається за ф. (4.6):

$$t = h_B + \frac{l_B}{2} \quad (4.6)$$

де h_B - глибина закладання вертикальних заземлювачів (0,8 м); тоді $t = 0,8 + \frac{3}{2} = 2,3$ м

$$R_B = \left(\frac{68}{2 * \pi * 3} \right) * \left(\ln \frac{2 * 3}{0,05} + \frac{1}{2} * \ln \frac{4 * 2,3 + 3}{4 * 2,3 - 3} \right) = 18,50 \text{ м}$$

5) Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання η_B :

$$n = \left(\frac{2 * R_B}{R_d} \right) = \frac{2 * 18,5}{4} = 9,25 \quad (4.7)$$

6) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання n_B , шт:

$$n_B = \left(\frac{2 * R_B}{R_d * \eta_B} \right) = \frac{2 * 18,5}{4 * 0,57} = 16,2 \approx 16 \quad (4.8)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача l_c , м:

$$l_c = 1,05 * L_B * (n_B - 1) \quad (4.9)$$

де l_B - відстань між вертикальними заземлювачами, (прийняти за $L_B = 3$ м);

n_B - необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 * 3 * (16 - 1) \approx 48 \text{ м}$$

8) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки) R_r , Ом:

$$R_r = \frac{P_{\text{розр.в}}}{2 * \pi * l_c} * \ln \frac{2 * l_c^2}{d_{cm} * h_r} \quad (4.10)$$

де d_{cm} - еквівалентний діаметр смуги шириною b , $d_{cm} = 0,95b$, $b = 0,15$ м;

h_r - глибина закладання горизонтальних заземлювачів (0,5 м);

l_c - довжина з'єднувальної стрічки горизонтального заземлювача l_c , м

$$R_r = \frac{220}{2 * \pi * 48} * \left(\ln \frac{2 * 48^2}{0,95 * 0,15 * 0,5} \right) = 8,1 \text{ Ом}$$

9) Визначається коефіцієнт використання горизонтального заземлювача η_c . відповідно до необхідної кількості вертикальних заземлювачів n_B . Коефіцієнт використання з'єднувальної смуги $\eta_c = 0,3$ (табличне значення).

10) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{\text{заг}} = \frac{R_B * R_r}{R_B * \eta_B + R_r * n_B * \eta_B} \quad (4.11)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{\text{заг}} \leq R_{\text{д}}$, а саме:

$$R_{\text{заг}} = \frac{18,5 * 8,1}{18,5 * 0,3 + 8,1 * 16 * 0,57} = 1,9 \leq R_{\text{д}}$$

4.8 Екологія

Діяльність за темою магістерської роботи, а саме: оптимізація запитів до бази даних в процесі її виконання впливає на навколишнє природне середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища» [12], Законом України «Про оцінку впливу на довкілля» [14], Законом України «Про відходи» [13].

В процесі діяльності з виконанням дипломного проектуванням виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- Відпрацьовані люмінесцентні лампи - I клас небезпеки.
- Змінні носії інформації - IV клас небезпеки.
- Відпрацьовані вогнегасники - IV клас небезпеки.
- Макулатура - IV клас небезпеки.

4.9 Висновки до четвертого розділу

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в кваліфікаційній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника.

Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та значення температури, вологості й рухливості повітря, необхідна кількість ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

Подальше поширення інформаційних технологій не збільшить, а навпаки – зменшить техногенне навантаження на довкілля.

4.10 Перелік джерел посилань до четвертого розділу

1. Закон України "Про охорону праці". Вводиться в дію Постановою ВР № 2695-ХІІ від 14.10.92, ВВР, 1992, № 49, ст.669. - Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/2694-12](http://www.URL:https://zakon.rada.gov.ua/laws/show/2694-12)
2. Кодекс законів про працю України. Затверджується Законом № 322-VIII від 10.12.71 ВВР, 1971. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/322-08](http://www.URL:https://zakon.rada.gov.ua/laws/show/322-08)
3. Закон України "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності". Наказ від 21 грудня 2000 року N 2180-III. Режим доступу: [www. URL: https://dnaop.com/html/2065/doc-zakon-ukrajini-pro-zagalynoobovjzskove-derzhavne-socialynestrahuvannya-vid-neshhasnogo-vipadku-na-virobnictvi-ta-profesijnogo-z](http://www.URL:https://dnaop.com/html/2065/doc-zakon-ukrajini-pro-zagalynoobovjzskove-derzhavne-socialynestrahuvannya-vid-neshhasnogo-vipadku-na-virobnictvi-ta-profesijnogo-z)
4. Про затвердження Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці (НПАОП 0.00-4.12-05). Наказ від 26.01.2005 №15. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/z0231-05](http://www.URL:https://zakon.rada.gov.ua/laws/show/z0231-05)
5. Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. Постанова N 42 від 01.12.99. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/va042282-99](http://www.URL:https://zakon.rada.gov.ua/rada/show/va042282-99)

6. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98. Затверджено Постановою Головного державного санітарного лікаря України 10 грудня 1998 р. N 7. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/v0007282-98](http://www.zakon.rada.gov.ua/rada/show/v0007282-98)
7. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом. Наказ від 1 липня 2016 року N 204. Режим доступу: [www. URL: http://epicentre.co.ua/dstu/doc28522.html](http://epicentre.co.ua/dstu/doc28522.html)
8. ДБН В.2.5-28:2018 «Природне і штучне освітлення». Режим доступу: [www. URL: http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf](http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf)
9. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за № 508/31960. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/z0508-18](https://zakon.rada.gov.ua/laws/show/z0508-18)
10. ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою». Наказ від 15.06.2016 №158. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/v0158858-16](https://zakon.rada.gov.ua/rada/show/v0158858-16)
11. Закон України «Про охорону навколишнього природного середовища». Вводиться в дію Постановою ВР № 1268-ХІІ від 26.06.91, ВВР, 1991, № 41, ст.547. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/1264-12](https://zakon.rada.gov.ua/laws/show/1264-12)
12. Закони України «Про охорону навколишнього природного середовища». Вводиться в дію Постановою ВР № 4005-ХІІ від 24.02.94, ВВР, 1994, № 27, ст.219. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/4004-12](https://zakon.rada.gov.ua/laws/show/4004-12)
13. Закон України «Про відходи». Відомості Верховної Ради України (ВВР), 1998, № 36-37, ст.242. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/187/98-вр](https://zakon.rada.gov.ua/laws/show/187/98-вр)
14. Закон України "Про оцінку впливу на довкілля" (Відомості Верховної Ради (ВВР), 2017, № 29, ст.315) Із змінами, внесеними згідно із Законом [№ 733-IX від 18.06.2020](https://zakon.rada.gov.ua/laws/show/733-IX) Режим доступу: [www URL: https://zakon.rada.gov.ua/laws/show/2059-19#Text](https://zakon.rada.gov.ua/laws/show/2059-19#Text)

ВИСНОВКИ

Метою дипломної роботи визначено дослідження методів та програмних засобів інтелектуального аналізу даних на основі прецедентів в ІСППР.

В ході виконання дослідницької частини були отримані наступні результати:

1. Проведено аналіз існуючих методів та програмних засобів аналітичної підтримки прийняття рішень в ІСППР на основі прецедентів.
2. Розглянуті існуючі програмні рішення для ІСППР.
3. Розроблено концептуальна модель підтримки прийняття рішень на основі прецедентів.

В ході практичної частини були отримані наступні результати:

1. Проведені експерименти моделювання у платформі Deductor двома засобами (карт Кохонена і дерева рішень).
2. Виконано оцінку якості методів вилучення прецедентів.

Таким чином, вирішено поставлене завдання - проведено дослідження інформаційних технологій підтримки прийняття рішень на основі прецедентів.

ДОДАТОК А ЕЛЕКТРОННА ПРЕЗЕНТАЦІЯ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ ПРЕЦЕДЕНТІВ

Ст. гр. КН-19дм
Керівник

Сацук О.О.
к.т.н., доцент Шумова Л.О.

Рисунок А.1 - Перший слайд презентації

Актуальність

2

ІСППР є затребуваним класом програмних систем. Їх створення - складне і ресурсовитратне завдання, особливо, якщо мова йде про слабоформалізованні області, що характеризуються великими обсягами використовуваних в них експертних знань. Але використання існуючих інструментаріїв обмежена їх жорсткою орієнтованістю на рішення певного класу задач. Тому для розробки ІСППР потрібні досконаліші інформаційні технології ІАД.

У ІАД для вилучення нових знань з наявних даних застосовуються різні методи, серед яких активно просуваються методи правдоподібних міркувань на основі прецедентів (СВР – Case-Based Reasoning).

Мета. Дослідження методів та програмних засобів інтелектуального аналізу даних на основі прецедентів в ІСППР.

Рисунок А.2 - Другий слайд презентації

Постановка завдання

3

- провести аналіз існуючих методів та програмних засобів аналітичної підтримки прийняття рішень в ІСППР на основі прецедентів;
- розглянути існуючі програмні рішення для ІСППР;
- створити концептуальну модель підтримки прийняття рішень на основі прецедентів;
- сформулювати загальну постановку задачі навчання по прецедентах
- дослідити ефективні алгоритми для вилучення прецедентів
- реалізувати і оцінити ефективність методів вилучення прецедентів для конкретної задачі

Об'єкт дослідження – процеси інтелектуальної підтримки прийняття рішень

Предмет дослідження – програмні засоби інтелектуальної підтримки прийняття рішень на основі прецедентів.

Рисунок А.3 - Третій слайд презентації

Основні задачі інтелектуальної підтримки прийняття рішень

4

- аналіз прикладів (case analyses)
- параметричний (case analyses) аналіз
- аналіз чутливості
- аналіз можливостей
- аналіз впливу
- аналіз даних
- порівняння та узагальнення
- командні послідовності
- аналіз ризику
- оптимізація

Рисунок А.4 - Четвертий слайд презентації

Методи подання і вилучення прецедентів

- **Метод найближчого сусіда (NN - Nearest Neighbor)**
- **Метод вилучення прецедентів на основі дерев рішень**
- **Метод вилучення прецедентів на основі знань**
- **Метод вилучення прецедентів з урахуванням їх застосовності**

Рисунок А.5 - П'ятий слайд презентації

Програмні засоби для інтелектуальної підтримки прийняття рішень в СППР

- Protégé OWL
- Мова програмування C
- Мова програмування C++
- Мова програмування C#
- Мова програмування Perl
- Мова програмування Python
- Мова програмування Java
- Deductor

Рисунок А.6 - Шостий слайд презентації

Концептуальна модель підтримки прийняття рішень на основі прецедентів

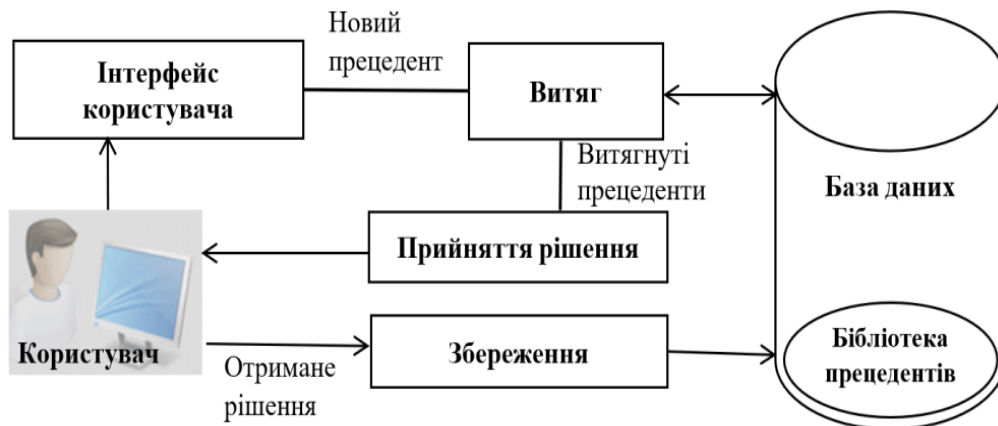


Рисунок А.7 - Сьомий слайд презентації

Загальна постановка задачі навчання по прецедентах

- **Задано** безліч об'єктів X , безліч допустимих об'єктів Y , і існує цільова функція $y^*: X \rightarrow Y$, значення якої $y_i = y^*(x_i)$ відомі тільки на кінцевому підмножині об'єктів $\{x_1, \dots, x_l\} \in X$. Пари «об'єкт - відповідь» (x_i, y_i) називаються прецедентами. сукупність пар

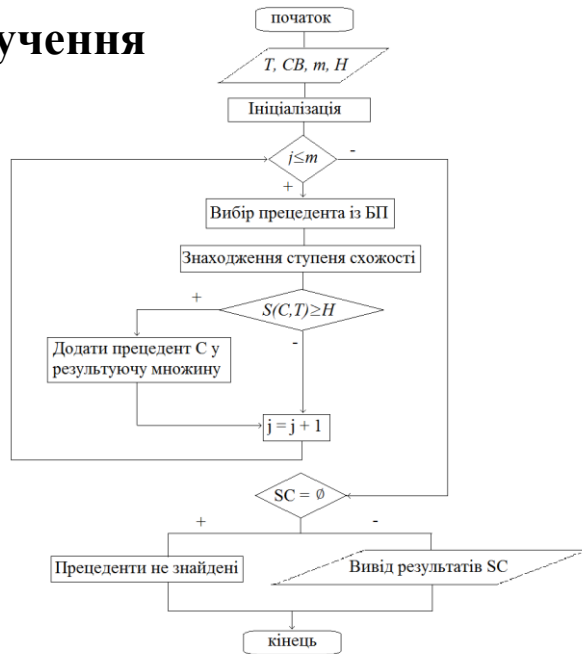
$$X^l = (x_i, y_i)_{i=1}^l \text{ називаються навчальною вибіркою.}$$

- **Завдання навчання по прецедентах** полягає в тому, щоб за вибіркою X^l відновити залежність y^* , тобто побудувати вирішальну функцію $a: X \rightarrow Y$, яка наближала б цільову функцію $y^*(x)$, причому не тільки на об'єктах навчальної вибірки, а й на всій безлічі X . Вирішальна функція a повинна допускати ефективну комп'ютерну реалізацію, з цього будемо називати її алгоритмом.

Рисунок А.8 - Восьмий слайд презентації

Алгоритм k-NN для вилучення прецедентів

- T - поточна ситуація;
- CB - непорожня безліч прецедентів (БП);
- m - кількість розглянутих прецедентів з БП;
- $S(C, T)$ - задана метрика (міра схожості);
- H - порогове значення ступеня подібності.



9

Рисунок А.9 - Дев'ятий слайд презентації

Реалізація методів прийняття рішень на основі прецедентів

10

Постановка задачі:

Є БД прецедентів банківського маркетингу.

Необхідно виявити вплив таких чинників:

вік клієнта, сімейний стан, освіту, чи є кредит в дефолті, середньорічний баланс, кількість контактів, результат попередньої маркетингової кампанії

на бажану мету - підписався чи клієнт на строковий вклад?

Рисунок А.10 - Десятий слайд презентації

Моделювання за допомогою Дерева рішень в АП Deductor

11

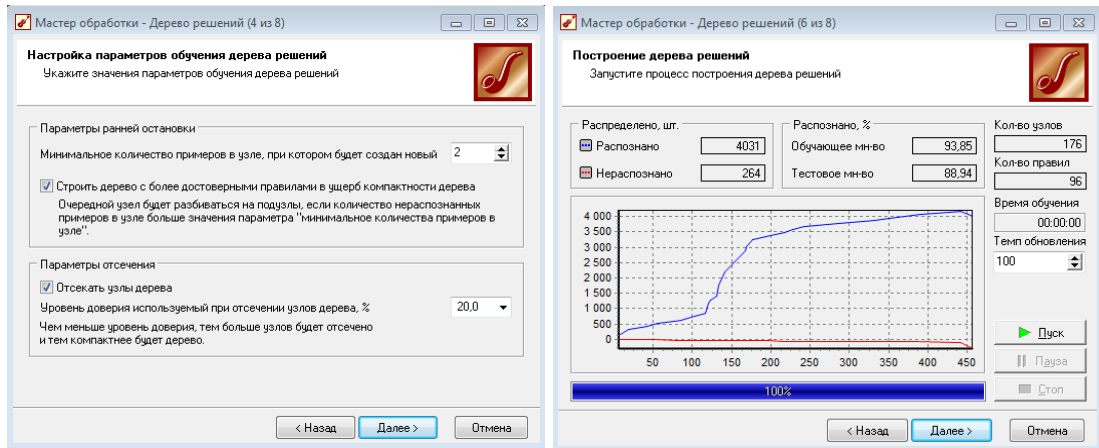


Рисунок А.11 - Одинадцятий слайд презентації

Моделювання за допомогою Карти Кохонена в АП Deductor

12

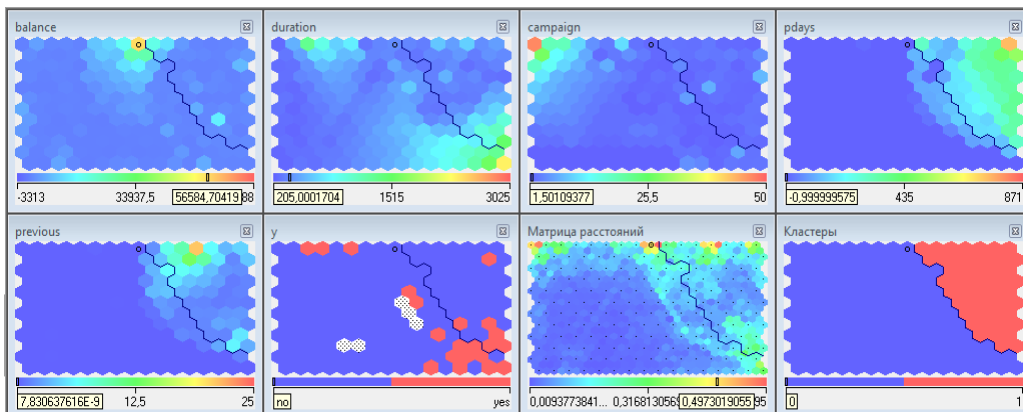


Рисунок А.12 - Дванадцятий слайд презентації

Оцінка ефективності методів вилучення прецедентів для конкретної задачі

13

Таблиця спряженості Дерева рішень для навчальної та тестової вибірки:

| Фактически | Класифіковано | | | Итого |
|------------|---------------|-----|-------|-------|
| | no | yes | Итого | |
| no | 3958 | 42 | 4000 | |
| yes | 436 | 85 | 521 | |
| Итого | 4394 | 127 | 4521 | |

| Фактически | Класифіковано | | | Итого |
|------------|---------------|-----|-------|-------|
| | no | yes | Итого | |
| no | 191 | 3 | 194 | |
| yes | 26 | 6 | 32 | |
| Итого | 217 | 9 | 226 | |

Таблиця спряженості Карти Кохонена для навчальної та тестової вибірки:

| Фактически | Класифіковано | | | Итого |
|------------|---------------|-----|-------|-------|
| | no | yes | Итого | |
| no | 3898 | 102 | 4000 | |
| yes | 371 | 150 | 521 | |
| Итого | 4269 | 252 | 4521 | |

| Фактически | Класифіковано | | | Итого |
|------------|---------------|-----|-------|-------|
| | no | yes | Итого | |
| no | 206 | | 206 | |
| yes | 16 | 4 | 20 | |
| Итого | 222 | 4 | 226 | |

Рисунок А.13 - Тринадцятий слайд презентації

Висновки

14

- проведено аналіз існуючих методів та програмних засобів аналітичної підтримки прийняття рішень в ІСППР на основі прецедентів;
- розглянуті існуючі програмні рішення для ІСППР;
- створено концептуальну модель підтримки прийняття рішень на основі прецедентів;
- сформульовано загальну постановку задачі навчання по прецедентах;
- досліджено ефективні алгоритми для вилучення прецедентів;
- реалізовано і оцінено ефективність методів вилучення прецедентів для конкретної задачі.

Рисунок А.14 - Чотирнадцятий слайд презентації