

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра ТЕЛЕКОМУНІКАЦІЙ, МЕДІЙНИХ ТА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ

Освітній рівень МАГІСТР

Галузь знань 11 МАТЕМАТИКА ТА СТАТИСТИКА

Спеціальність 113 ПРИКЛАДНА МАТЕМАТИКА

Освітня програма ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА ПІДГОТОВКИ МАГІСТРА

ЗАТВЕРДЖУЮ

Зав. кафедри _____

03 вересня 2020 р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Жовнір Марія Юріївна

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод прогнозування показників виробничих процесів на основі інтелектуального аналізу даних

Керівник проекту (роботи) Кисіль Тетяна Миколаївна, к.ф.-м.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.09.2020 р. № 118

2. Строк подання студентом проекту (роботи) на кафедру 01.12.2020 р.

3. Вихідні дані до проекту (роботи) Оптимізаційна модель інтелектуального прогнозування тривалості виробництва кліматичного обладнання, результати порівняння сценаріїв прогнозування, ІС пошуку залежностей в емпіричних даних і використання таких залежностей при прогнозуванні технічних показників нових об'єктів.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) Аналіз особливостей прогнозування інтервалу тривалості виробництва складних технічних об'єктів. Формалізація завдання інтелектуального аналізу даних. Розробка моделі інтелектуального планування. Розробка методу вирішення задач прикладного типу прогнозування. Розробка алгоритмів та вибір програмного забезпечення для реалізації поставленої задачі.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) Діаграма пошуку гіпотез, діаграма класів об'єктної моделі ІС, таблиці статистики використання гіпотез, схеми реалізації компонентів індукції, абдукції, результати класифікації модифікованої навчальної вибірки тривалості виробництва проектів.

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
РОЗДІЛ 1	канд. ф-м-н., доцент, Кисіль Т.М.	02.03.2020	09.03.2020
РОЗДІЛ 2	канд. ф-м-н., доцент, Кисіль Т.М.	11.09.2020	15.09.2020
РОЗДІЛ 3	канд. ф-м-н., доцент, Кисіль Т.М.	01.10.2020	20.10.2020
РОЗДІЛ 4	канд. ф-м-н., доцент, Кисіль Т.М.	09.11.2020	12.11.2020

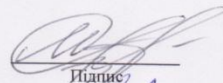

7. Дата видачі завдання «__» _____ 20__ р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1. Вибір напрямку дослідження та узгодження тематики ДРМ з керівником	03.08.2020 – 09.09.2020	Виконано
2. Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	15.08.2020 – 08.09.2020	Виконано
3. Робота над розділом 1 – аналіз відомих моделей, методів за темою	30.08.2020 – 10.09.2020	Виконано
4. Робота над розділом 2 – розробка моделей і методів для вирішення поставленої задачі	12.09.2020 – 27.09.2020	Виконано
5. Робота над науковою статтею	29.09.2020 – 02.10.2020	Виконано
6. Робота над розділом 3 – розробка алгоритмів та технологій, їх аналіз	03.10.2020 – 23.10.2020	Виконано
7. Робота над розділом 4 – проектування ПЗ для вирішення поставленої задачі	28.10.2020 – 09.11.2020	Виконано
8. Оформлення пояснювальної записки	05.11.2020 – 19.11.2020	Виконано
9. Оформлення графічної частини	07.11.2020 – 20.11.2020	Виконано
10. Попередній захист ДРМ	18.11.2020 – 29.11.2020	Виконано
11. Захист ДРМ на засіданні ЕК	03.12.2020 – 19.12.2020	Виконано

Студент

Керівник проекту (роботи)


 Підпис
 
 Підпис

Жовнір_М. Ю.

Кисіль Т. М.

АНОТАЦІЯ

Тема дипломної роботи: Метод прогнозування показників виробничих процесів на основі інтелектуального аналізу даних

Автор роботи: Жовнір Марія Юріївна

Керівник роботи: Кисіль Тетяна Миколаївна

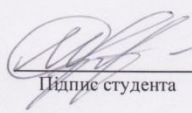
Загальний обсяг роботи: 145 сторінок, 33 рисунків, 10 таблиць, 2 додатки, 27 посилань.

ІНФОРМАЦІЙНА СИСТЕМА, КЛІМАТИЧНІ ШАФИ, БАЗИ ДАНИХ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, АВТОМАТИЧНЕ ПОРОДЖЕННЯ ГІПОТЕЗ.

Метою даної роботи є аргументований вибір інструментарію для вирішення спеціального класу реальних прикладних задач - швидкого прогнозування інтервалу тривалості виробництва складних технічних об'єктів в умовах неповної інформації про їх детальну внутрішню структуру, а саме до виконання їх детального проектування і розробки.

Основними теоретичними результатами даної роботи є розробка формалізованої моделі та системи алгоритмів для вирішення поставленого завдання прогнозування. Практичним результатом є розробка на основі створених алгоритмів інформаційної системи (ІС), технологія формування та архітектура якої надають можливості для гнучкого розширення аномодифікації функціоналу системи при виникненні нових вимог, а також при появі нових теоретичних результатів, чи розвитку системи при необхідності до корпоративної рівня (інформаційної системи, що використовується в якості штатного інструментального засобу при вирішенні виробничих завдань вербального типу), а також перенесення частини розробленого функціоналу системи на мобільні платформи.

Дата


Підпис студента

ANNOTATION

Thesis topic: Method of forecasting indicators of production processes based on data mining

Author of the work: Zhovnir Maria Yuriyivna

Mentor: Kisil Tetyana Mykolayivna

Total volume of work: 145 pages, 33 figures, 10 tables, 2 appendices, 27 references.

INFORMATION SYSTEM, CLIMATE CABINETS, DATABASES,
SOFTWARE, AUTOMATIC GENERATION OF HYPOTHESES.

The purpose of this work is a reasoned choice of tools (mathematical models and software) to solve a special class of real applied problems - rapid prediction of the duration of production of complex technical objects in terms of incomplete information about their detailed internal structure, namely to perform their detailed design and development.

The main theoretical results of this work are the development of a formalized model and system of algorithms for solving the problem of forecasting. The practical result is the development based on the created algorithms of information system (IS), the technology of formation and architecture of which provide opportunities for flexible expansion of a modification of system functionality in the event of new requirements, as well as new theoretical results or system development if necessary to the corporate level. system used as a regular tool in solving production problems of the verbal type), as well as the transfer of part of the developed functionality of the system to mobile platforms.

Date



Signature

Зміст

ВСТУП	8
СПИСОК СКОРОЧЕНЬ.....	11
1 ПОСТАНОВКА ЗАДАЧІ.....	12
1.1 Опис початкових даних	12
1.2 Змістовна постановка задачі та її особливості.....	16
1.3 Формальна постановка задачі	17
2 ОГЛЯД ВІДОМИХ МЕТОДІВ ВИРІШЕННЯ ЗАДАЧ ДАНОГО ТИПУ ..	19
2.1 ДСМ-метод автоматичного породження гіпотез.....	18
2.2 Аналіз виконання умови застосування ДСМ-методу до...._розв'язуваної задачі.....	19
2.3 Вибір алгоритму класифікації	22
3 МОЖЛИВОСТІ ТА ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ДСМ – МЕТОДУ В ЗАДАЧАХ РОЗГЛЯНУТОГО ТИПУ	28
3.1 Неформальна семантика ДСМ-методу АПГ	28
3.2 Алгоритміка ДСМ- методу АПГ	31
3.2.1.Опис алгоритму пошуку глобальних подібностей	32
3.2.1.1 Основні вимоги до алгоритму глобальних подібностей.....	32
3.2.1.2 Модифікований алгоритм пошуку подібностей.....	34
3.2.1.2.1 Основні ідеї, покладені в основу алгоритму.....	34
3.2.1.2.3 Функціональна схема розробленого алгоритму пошуку .	37
3.2.1.2.4 Доведення коректності розробленого алгоритму	39
3.2.1.2.5 Оцінка обчислювальної складності алгоритму.....	40
3.2.2 Опис алгоритмів основних етапів ДСМ методу	42

3.2.2.1	Опис алгоритму породження гіпотез	43
3.2.2.2	Опис алгоритму класифікації	47
4	ПРИКЛАДНА ЗАДАЧА: ПОСТАНОВКА ТА РОЗВ'ЯЗАННЯ	53
4.1	Опис обраного інструментарію	53
4.1.1	Формування основних вимог до технології розробки	53
4.1.2	Вибір інструментарію, відповідного пред'явленим вимогам	55
4.2	Загальна архітектура і структура даних ІС	56
4.3	Архітектура розроблених компонентів ІС	57
4.4	Характеристики функціонування ІС	62
4.5	Апробація програмної системи на задачах спеціальної прикладної предметної області	63
4.5.1	Підготовка вихідних даних для застосування ДСМ-методу	63
4.5.2	Уточнення основних понять і зв'язків, які використовуються в процесі застосування ДСМ-методу, стосовно до розв'язування задачі	64
4.5.3	Уточнення основних етапів ДСМ-методу стосовно розв'язуваної прикладної задачі	67
4.6	ДСМ - метод як платформа ІАД	77
4.6.1	Логічні та алгоритмічні засоби як інструментарій формалізації класу евристик для використання в комп'ютерних системах ІАД	77
4.6.2	Демонстрація можливостей поліпшення якості ДСМ прогноза на основі спрямованого розширення вихідних даних	78
	ВИСНОВКИ	83
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	85
	ДОДАТОК А	88
	ДОДАТОК Б	96

ВСТУП

В даний час відбувається інтенсивний розвиток галузі телекомунікації та зв'язку. В основі цього розвитку лежить поява нових телекомунікаційних технологій, наявність потреби у збільшенні зони покриття вже існуючих стільникових мереж і в надійності використовуваного телекомунікаційного обладнання. Для надійного функціонування використовуваного стільниковими операторами телекомунікаційного обладнання необхідно забезпечити оптимальні кліматичні умови експлуатації технологічного обладнання незалежно від умов зовнішнього середовища або особливостей конкретної місцевості.

Даний фактор обумовлює необхідність залучення в проекти по впровадженню стільникових мереж також і компаній виробників кліматичного обладнання. Кількість необхідних кліматичних рішень, впроваджуваних даними компаніями, в залежності від чисельності клієнтської бази і кліматичних умов регіонів, може бути досить чисельною. Результати роботи ІС, а саме прогноз термінів виробництва вироби, також можуть бути використані при прийнятті рішення керівництвом компанії про участь або неучасть підприємства в конкретному уже згадуваному проекті, а також, в разі небажаність прийняття рішення про відмову від участі, завчасно вжити заходів щодо збільшення виробничих потужностей (при відповідній необхідності).

Виявлені чинники допоможуть визначити слабкі та сильні сторони підприємства, детальний облік яких в організації процесу виробництва може зробити істотний позитивний вплив при виконанні нових проектів.

В даній роботі запропонований підхід, на базі якого ведеться розробка основних алгоритмів, що дозволяють вирішити поставлену задачу, а саме алгоритми пошуку глобальних хостів у вихідних даних, алгоритмів збірки гіпотезів з отриманих результатів, а також алгоритми класифікації нових об'єктів на основі отриманих гіпотезів, що і обумовлює наукову новизну роботи.

Актуальність роботи: потреба у збільшенні зони покриття вже існуючого телекомунікаційного обладнання, а саме забезпечення сталої та безперебійної роботи техніки в незалежності від кліматичних чинників вимагає нових підходів у проектуванні та розробці надійного кліматичного обладнання (кліматичних шаф).

Метою даної роботи є виконати аргументований вибір інструментарію (математичних моделей і програмних засобів) для вирішення спеціального класу задач прогнозування інтервалу тривалості виробництва складних технічних об'єктів (кліматичних шаф) в умовах неповної інформації про їх детальну внутрішню структуру, а саме на етапі передконтрактних переговорів, тобто до виконання їх детального проектування і розробки.

Для досягнення поставленої мети мають бути вирішені наступні завдання:

- 1) Аналіз особливостей прогнозування інтервалу тривалості виробництва складних технічних об'єктів;
- 2) Формалізація завдання інтелектуального аналізу даних;
- 3) Розробка методу вирішення задач прогнозування;
- 4) Розробка моделі інтелектуального планування;
- 5) Розробка алгоритмів, вибір та реалізація програмного забезпечення для вирішення поставленої задачі;

Об'єкт: Об'єктом дослідження є процес прогнозування показників виробничих процесів на основі інтелектуального аналізу даних.

Предмет: Предметом дослідження є оптимізаційні методи та практичні рекомендації щодо прогнозування термінів виробництва виробів

Крім вирішення завдання прогнозування, розроблена система також повинна забезпечувати виявлення чинників, що пояснюють підстави перенесення інтервалу тривалості виробництва виробу саме до періоду спрогнозованої тривалості. Виявлені чинники допоможуть визначити слабкі та сильні сторони підприємства, детальний облік яких в організації процесу виробництва може зробити істотний позитивний вплив при виконанні нових проектів.

СПИСОК СКОРОЧЕНЬ

ІС – Інформаційна система

БД – База даних

ПЗ – Програмне забезпечення

АПГ – Автоматичне породження гіпотез

ІАД – Інтелектуальний аналіз даних

1 ПОСТАНОВКА ЗАДАЧІ

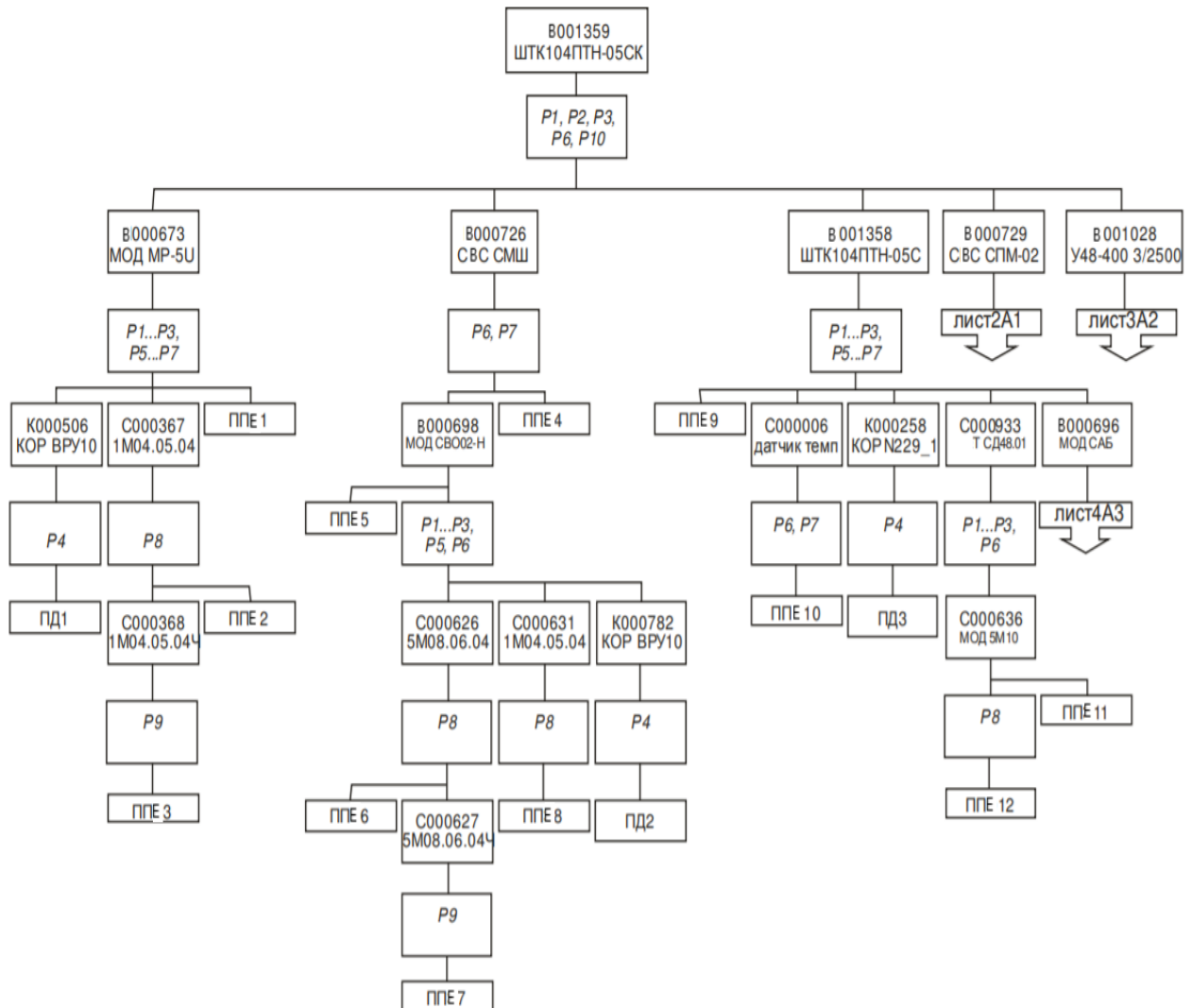
1.1 Опис початкових даних

Одними із спеціалізованих телекомунікаційних кліматичних пристроїв прийнято вважати кліматичні шафи. Призначенням кліматичних шаф являється, розміщення активного телекомунікаційного обладнання в будь-яких кліматичних умовах, не зважаючи на температуру навколишнього середовища чи інші погодні чинники, тобто використання кліматичних шаф захищає систему від перегріву чи переохолодження. Саме кліматичні шафи забезпечують безперебійне функціонування телекомунікаційних систем за будь-яких температури.

Найчастіше використовуються двохкамерні кліматичні шафи, тобто сама камера шафи оснащена двома дверцятами, де перші (зовнішні) дверцята забезпечують безпеку (запобігання спробам злому) і захист від екстремальних погодних умов. Окрім цього кожна кліматична шафа обладнана терморегуляторами. Терморегулятор – це основний елемент кожної кліматичної шафи. Саме з його допомогою здійснюється налаштування всіх необхідних параметрів. При установці кліматичної шафи встановлюються граничні температури при яких, за потреби, буде ввімкнена чи вимкнута системи охолодження, а також температура, при якій будуть ввімкнуті чи вимкнуті нагрівальні елементи шафи. Тобто, при перевищенні показника +25 градусів за Цельсієм буде вмикатись внутрішній вентиляторів. За аналогічним принципом при досягненні встановленого внутрішнього нижнього показника активується і системи обігріву шафи в холодні пори року. В загальному кожна кліматична шафа здатна забезпечити безпеку обладнання при температурі від +40°C до -40°C. За потреби в якості додаткового утеплюючого матеріалу може використовуватись алюфна прошивка товщиною до 10 мм, якою покриваються лише стінки кліматичних шаф зсередини. Також варто зазначити, що зовні шафи покриваються двома спеціальними фарбами, що роблять їх стійкими до корозій.

Кліматичне обладнання складається з різної кількості модулів, архітектура і порядок взаємодії яких можуть бути представлена у вигляді ієрархічної структури

(дерева). Нижче (рисунок 1.1) представлений приклад структурної схеми одної з



кліматичних шаф.

Рисунок 1.1 – Структурна схема кліматичної шафи

В схемі використовуються наступні позначення:

1. **Виріб** (Позначення - Vxxxxxx, наприклад В000001) – готовий виріб, що виготовляється на підприємстві. Наприклад: шафа, модульний каркас, контролер і т.д. Виріб може містити в собі інші вироби і входити до складу інших виробів.

Наприклад, модуль розподілу може продаватися окремо або входити до складу кліматичної шафи.

2.Збірна одиниця (Позначення Схxxxxxx, наприклад С000001) – виготовляється на підприємстві, може продаватися тільки в складі виробу. Наприклад: електронний модуль, складна деталь корпусу, що вимагає при виготовленні операцій зварювання, клепання і т.д.

3. Корпус (Позначення - Кхxxxxxx, наприклад К000001) – корпус виробу або збірної одиниці виробу.

4. Корпусні деталі (Позначення - Дхxxxxxx, наприклад Д000001) – деталь корпусу, яка не має в складі покупних складових, крім металу та фарби.

5. Перелік деталей корпусу (Позначення - ПДх, наприклад ДД1) – набір корпусних деталей.

6. Модуль покупної деталі (Позначення - Пхxxxxxx, наприклад П000001) – готовий куплений модуль або компонент виробу.

7. Перелік придбаних елементів (Позначення - ППЕх, наприклад ППЕ1) – набір готових придбаних модулів.

8. Заказний модуль (Позначення - Зхxxxxxx, наприклад З000001) – виріб, що виготовляється по ТЗ підприємства стороннім виробником. Інформація про всі проекти виробництва виробів і вироблених виробих зберігається в базі даних. Структура бази даних представлена на рисунку 1.2.

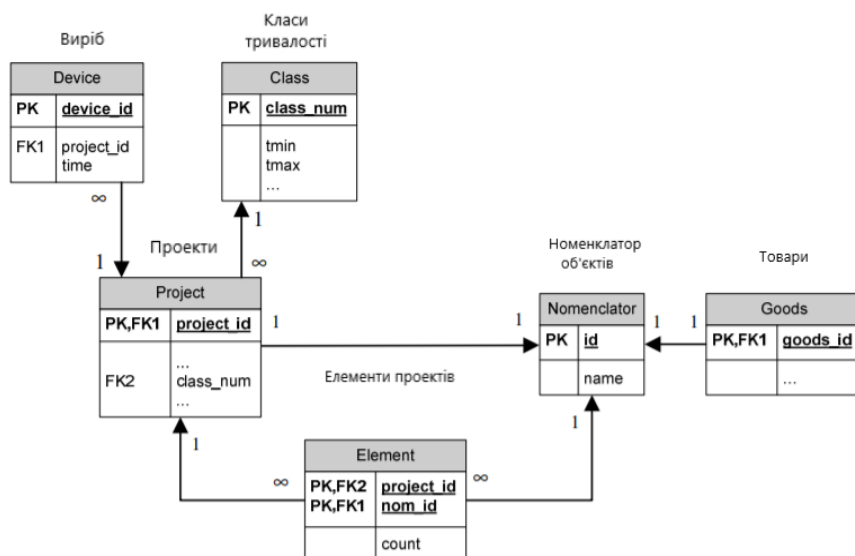


Рисунок 1.2 – Концептуальна модель сутність-зв'язок бази даних виконаних виробів.

Кожен виріб виконується за проектом виробництва виробу. У проекті представлений тільки склад виробів. До складу виробів входять різні об'єкти з номенклатора: інші вироби, модулі, корпусу, а також комплектуючі. У базі даних також містяться відомості про час виробництва кожного готового виробу. Крім того, кожен проект, за яким виконувалися вироби, фахівцями підприємства віднесений до одного з класів тривалості, що характеризує час виконання виробів за даним проектом. Інформація про класи тривалості проектів також представлена в базі даних.

Нижче представлено формальний теоретико-множинний опис представлених вище сутностей.

Введемо наступні позначення:

$$O - \text{перелік всіх об'єктів } o \in O \quad (1.1)$$

номенклатора

$$P - \text{перелік проектів } p \in P \text{ виробів} \quad (1.2)$$

Кожен проект $p \in P$ можна представити як список пар (об'єкт з номенклатора, кількість об'єктів) (відповідно до рисунку 1.3).

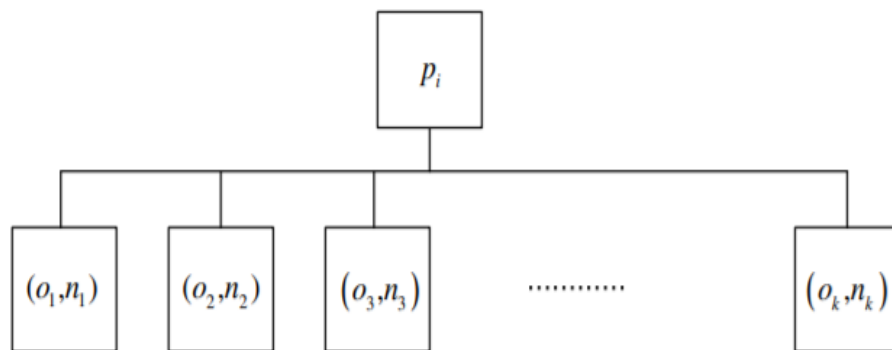


Рисунок 1.3 – Представлення проекту виробів

Будемо називати таку пару елементом проекту e , тобто $e = (o, n) \in p$, де $o \in O$, n – кількість об'єктів $n \in N$. Введемо також наступні позначення:

E – сукупність усіх елементів проектів, де кожен елемент $e \in E$ має вигляд $e = (o, n)$. Тоді $P \in \wp(E)$, де $\wp(E)$ – безліч підмножин множини E .

Тепер неважко записати зв'язок між введеними множинами:

$$p_i \subseteq E \subseteq O \times N \quad (1.3)$$

Введемо наступні позначення:

$$D = \{d_1, \dots, d_m\} \text{ – вироби, що зберігаються в базі даних} \quad (1.4)$$

$$T = \{t_1, \dots, t_m\}, \text{ де } t \in T; \quad (1.$$

5)

Кожен виріб $d \in D$ являє собою пару (проект, час виробництва) $d = (t, p)$, де $p \in P$ – проект по якому виконується виріб, де $t \in T$ – час виконання виробу. Тепер неважко записати зв'язок між введеними множинами: $D \subseteq P \times T$.

Проект $p \in P$ може бути віднесений до одного з класів тривалості, що характеризує час виконання виробів за даним проектом. Введемо наступні позначення:

$$C = \{c_1, c_2, c_3, c_4, c_5\} \text{ – безліч класів тривалості виробництва} \quad (1.6)$$

Класу $c \in C$ відповідає відфільтрована пара $\{t_c \min t_c \max\}$, що відповідає інтервалу тривалості виробництва $\{t_c \min t_c \max\}$.

$P' \subset P$, т.е. $P' = \{p_1, \dots, p_h\}$, $h < n$ підмножина проектів, за якими були виконані вироби. Таким чином між підмножиною $P' \subseteq P$ і більшістю класів C існує функціональна відповідність $\mu: P' \rightarrow C$, яка відображає той факт, що кожен проект, по якому були виконані вироби, може бути віднесений до певного класу тривалості $c \in C$ виробництва виробів за проектом.

1.2 Змістовна постановка задачі та її особливості

Одним із завдань даної роботи є розробка і практична реалізація ІС, яка на основі неповної вихідної інформації про конфігурацію виробу (тобто за проектом виробництва виробу p) буде здійснювати прогноз інтервалу тривалості виробництва виробу $\{t_{c \min} t_{c \max}\}$, тобто відносити проект нового виробу до певного класу тривалості c .

Особливостями завдання є:

1. Необхідність оперувати об'єктами складної структури (див пункт 1.1);
2. Наявність неповної інформації про конфігурацію об'єкта на етапі прийняття рішення;
3. Мала (як правило статистично значуща) вибірка вже завершених проектів;
4. Наявність прихованих (неявно виражених в описах відповідних прецедентів) причинно-наслідкових залежностей в накопиченій безлічі фактів (даних про прецеденти виробництва виробів за певний термін).

У ситуації, що розглядається властивості 2 і 3 визначають той факт, що коректне рішення даного завдання, взагалі кажучи, не може бути породжене чисельними методами або ж методами статистичного аналізу. Однак, беручи до уваги властивість 3, можна припустити, що можливими методами вирішення даного класу задач є методи ІАД (Інтелектуальний Аналіз Даних), що дозволяють виявити закономірності, закладені у вихідних даних, і проаналізувати їх наявність в новому проекті виробу.

Нижче запропонований варіант формалізації (формальної постановки) обговорюваної задачі і проведений огляд методів ІАД, придатних для вирішення поставленої задачі.

1.3 Формальна постановка задачі

Існує безліч проектів виробництва виробів P (див.(1.2)) і кінцева множина імен класів тривалості проектів виробництва виробів C (див.(1.6)). Передбачається, що існує деяка невідома залежність відображення з множини

проектів в множину імен класів: $c^*: P \rightarrow C$. Ця цільова залежність відома тільки на невеликій, кінцевій підмножині проектів, яка називається навчальна вибірка:

$P' = \{p_1 \dots p_m\} \subseteq P$ – навчальна вибірка;

$c_i = c^*(p_i)$, $i = 1, \dots, k$ – відомі відповіді;

Нехай $\varphi_c : P \rightarrow \{0,1\}$ - деякий предикат, визначений на багатьох проектах P , який показує належність чи неналежність проекту $p \in P$ до класу $c \in C$. Скажемо, що предикат φ_c покриває проект $p \in P$, якщо $\varphi_c(p) = 1$. Предикат φ_c будемо називати закономірністю. Таким чином, передбачається існування деяких логічних закономірностей $\varphi_c(p)$ між складом проектів і класом тривалості виробництва $c \in C$, до якого вони належать.

Завданням даної роботи є виявлення в наявних вихідних даних інтерпретованих (на наявному емпіричному матеріалі) логічних закономірностей $\varphi_c(p)$ і побудова алгоритму класифікації $a: P \rightarrow C$ апроксимуючого c^* на безлічі P на основі знайдених закономірностей.

Для вирішення поставленого завдання будемо використовувати логічні алгоритми класифікації. Основні вимоги, що пред'являються до шуканого логічного алгоритму класифікації:

1. Надання методології вирішення поставленого завдання класифікації.
2. Можливість інтерпретації результатів класифікації в змістовних термінах розв'язуваної прикладної задачі.
3. Можливість виявляти закономірність на невеликій навчальній вибірці і вихідних даних зі складною структурою.

У Додатках А1-А2 представлений аналіз можливих методів вирішення поставленого завдання, на основі якого, з урахуванням описаних вище вимог, буде прийнято рішення про використання певного алгоритму класифікації.

2 ОГЛЯД ВІДОМИХ МЕТОДІВ ВИРІШЕННЯ ЗАДАЧ ДАНОГО ТИПУ

2.1 ДСМ-метод автоматичного породження гіпотез

Предметом вивчення ДСМ-методу є каузальні відносини між фрагментами структури об'єктів і множиною властивостей, притаманних цим об'єктам, якими досліджувані об'єкти можуть володіти, або не володіти. Головне питання, на яке ДСМ-метод дає відповідь, неформально звучить так: які особливості структури, наявні в досліджуваних об'єктах, що змушують їх володіти або не володіти певними властивостями? Ця залежність (*структура властивості*) встановлюється шляхом використання спеціальної техніки правдоподібних міркувань, формалізованих засобами багатозначних логік певного класу.

Один з принципів, що лежить в основі таких міркувань, неформально виглядає так «схожість описів об'єктів тягне схожість їх властивостей»: на основі аналізу подібності об'єктів шукається причина (структурний «носій») досліджуваних властивостей, яка потім використовується для прогнозування наявності цих якостей в інших об'єктах. Більш докладний опис ДСМ — методу, а саме неформальна семантика, що лежить в його основі, і формальну структуру представлено в розділі 3.1 і Додатку В.1 відповідно.

Основні переваги ДСМ методу АПГ:

- будучи не статистичним методом аналізу даних, він враховує індивідуальні особливості об'єктів дослідження.
- здатність породжувати гіпотези на малих масивах даних завдяки виявленню схожості об'єктів, що характеризуються істотними параметрами.
- можливості працювати з відкритими (поповнювальними) масивами даних, розпізнаючи необхідність керованого розширення БД, якщо таке виникає.
- породження отриманих правил (причинних залежностей) в явному виді, те що забезпечує інтерпретацію результатів навчання.
- недоліки ДСМ методу АПГ:
 - непридатність за умови наявності протиріччя в поточних прикладах.
 - велика обчислювальна складність.

- необхідність коректного визначення операції подібності на прикладі.

ДСМ метод АПГ активно застосовується при вирішенні наступних задач:

- автоматизованої побудови формалізації знань,
- технічної діагностики,
- дослідження детермінант соціальної поведінки,
- медична діагностика,
- прогнозування шляхів біотрансформації,
- прогнозування хімічної канцерогенності та токсичності,
- навчання мобільних роботів

2.2 Аналіз виконання умови застосування ДСМ-методу до розв'язуваної задачі

Безперечно, застосування ДСМ-методу пред'являє певні вимоги до досліджуваної предметної області, що вивчається. Проаналізуємо виконання трьох основних умов застосування ДСМ-методу до розв'язуваної задачі.

Основною вимогою є наступне:

У базі фактів повинні в неявному вигляді міститися залежності причинно-наслідкового типу, які можуть бути представлені висловлюваннями виду «підоб'єкт В є причина наявності (відсутності) властивостей А».

Для встановлення наявності причинно-наслідкових залежностей у використовуваній базі фактів була проаналізована вибірка, що складається з 24 проектів виробів. За результатами аналізу можна зробити висновок, що в структурних схемах виробів міститься достатня кількість «каузальних» («каузальна» роль відповідних фрагментів встановлювалася по їх наявності в архітектурі виробів - представників певного класу тривалості і відсутності в інших класах тривалості) архітектурних фрагментів, що забезпечує можливість виділення причинно-наслідкових залежностей у вихідних даних.

ДСМ-метод, як було сказано вище, має справу з описами об'єктів та їх властивостями. Однак, для того щоб зробити висновок про причини наявності або відсутності властивостей, ДСМ-метод повинен вивчити факти, подані як об'єкти, що володіють, так і об'єктами, що не володіють досліджуваними властивостями - вони називаються (+) - прикладами і (-) - прикладами, відповідно. Відсутність (-) - прикладів або (+) - прикладів або ж їх непропорційне представлення може привести до неінтерпретованості результатів ДСМ-міркування. Таким чином, 2-га умова застосування ДСМ-методу наступна:

Предметна область, опис якої представлено в базі фактів, повинна містити позитивні приклади, негативні приклади і приклади невизначеності досліджуваного ефекту.

У зроблених нами вихідних припущеннях наявність п'яти класів тривалості, на які в базі фактів поділяються проекти всіх виконаних виробів, автоматично забезпечує наявність як позитивних прикладів для кожного з класів тривалості (безпосередньо проекти-представники даного класу), так і негативних прикладів для кожного класу (проекти-представники класів тривалості, відмінних від розглянутого). Прикладам невизначеності досліджуваного явища відповідають проекти ще не виконаних виробів, для яких необхідно спрогнозувати клас тривалості.

Таким чином, третя умова застосовності ДСМ - методу полягає в наступному:

Потрібно визначити, що означає «мають в загальному лише одну обставину», іншими словами, повинна бути визначена операція подібності на об'єктах: результат цієї операції на деякій множині об'єктів (при виконанні певних умов) стає гіпотезою про причини наявності або відсутності досліджуваної властивості.

Мабуть, саме ця, третя, умова є ключовою при підготовці та структуруванні даних для застосування ДСМ-методу в якій б то не було предметній області. Визначення операції подібності тягне за собою необхідність вирішення й інших проблем: зокрема - уточнення того, що вважати формальним описом об'єкта

(уявлення об'єкта нерозривно пов'язане з описом відносини «об'єкт володіє властивістю»)

Іншими словами, потрібно побудувати адекватну модель опису фактів і причинних залежностей в аналізованій предметній області. Опис моделі розглянутої предметної області, а також визначення операції подібності, необхідної для виконання 3-ї умови застосування ДСМ-методу.

2.3 Вибір алгоритму класифікації

Такі методи ІАД як нейронні мережі, лінійна і логічна регресія, байєсівська класифікація, класифікація методом опорних векторів, метод найближчого сусіда не були розглянуті, тому що не задовольняють умови інтерпретації отриманих результатів, тобто не виявляється прихованих (але змістовно інтерпретованих) логічних закономірностей в даних.

Таким чином, основними можливими методами рішення поставленої задачі є логічні методи класифікації, з яких у наведеному вище огляді були розглянуті вирішальні списки, дерева рішень, класифікація методом голосування, ДСМ метод автоматичного породження гіпотез.

Вирішальні списки задовольняють умови інтерпретації результатів, простоті алгоритму, обробці різнотипних даних, проте даний метод не передбачає вирішення завдання пошуку закономірностей в вихідних даних, вирішуючи тільки завдання класифікації нового об'єкта. Незважаючи на зазначені вище недоліки, ідея, що лежить в основі даного методу, може бути використана при класифікації, але задача пошуку закономірностей залишатиметься невирішеною.

Вирішальні дерева задовольняють умови інтерпретації результатів, простоті алгоритму, масштабованості. Крім того, даний алгоритм на відміну від вирішальних списків, які використовують масив заздалегідь знайдених закономірностей, вирішує завдання пошуку закономірностей в час побудови дерева. Однак даний метод, крім очевидної жадібності і, як наслідок, неоптимальності, володіє ще одним істотним недоліком: при зменшенні розміру

навчальної вибірки даний метод стає статистично ненадійним, що неприйнятно виходячи з вимог, що пред'являються до шуканого алгоритму.

Метод зваженого голосування спільно з алгоритмом бустінга дозволяє, на відміну від вирішального списку, компенсувати неточності закономірностей, має високу узагальнюючу здатність, інтерпретованість, можливість фільтрації шуму, але даний метод, також як і вирішальні списки, не надає рішення задачі пошуку самих закономірностей.

ДСМ метод АПГ забезпечує швидке навчання, можливість пошуку закономірностей на малих вибірках, інтерпретацію результатів. Однак даний алгоритм має недолік, пов'язаний з неможливістю класифікації нового об'єкта в разі наявності шуму і протиріч в навчальних прикладах.

Таким чином, оптимальним способом вирішення поставленого завдання в умови поточного стану навчальної виборки, а саме в умовах відсутності протиріч, є використання ДСМ-методу АПГ.

Всі алгоритми, вирішальні завдання перерахування всіх подібностей, породжуваних заданим набором прикладів, мають експонентну обчислювальну складність: число шуканих подібностей зростає експоненціально швидко з ростом вихідної вибірки. При цьому неважко переконатися, що число породжуваних подібностей як і число можливих перетинів зростає експоненціально швидко [1]. Ці обставини роблять необхідним побудову алгоритмів, які вирішували б зазначену проблему найбільш економним (ефективним) чином. Далі проаналізуємо переваги та недоліки алгоритмів, що реалізують кожну їх стратегію.

Згідно зі стратегією 1, в загальному випадку задача пошуку всіх глобальних подібностей щодо довільної множини об'єктів S може бути вирішена за допомогою послідовного перебору всіх нетривіальних вершин дерева всіх перетинань. В ході виконання даного проходу для визначення глобальних подібностей необхідно обчислити ДСМ-замикання локальної схожості кожної вершини, у якій $h \neq \emptyset$. У тому випадку, коли орбіта локальної схожості збігається

з його ДСМ - замиканням, дана локальна схожість є одночасно глобальною схожістю і заноситься в список результатів (див алгоритм «ВП-bt дерево» [2]).

Однак, для виявлення всіх локальних подібностей в загальному випадку не обов'язково здійснювати перебір всіх вершин дерева всіх перетинань [3]. Таким чином, даний спосіб вирішення завдання є неоптимальним, так як в загальному випадку може здійснювати побудову зайвих вершин дерева всіх перетинів і виконувати операцію пошуку ДСМ-замикання для локальної схожості для кожної з подібних вершин. Даний недолік нівелюється при застосуванні алгоритмів, що реалізують стратегію 2.

Згідно зі стратегією 2, основним завданням при побудові алгоритму пошуку глобальних подібностей є виключення необхідності перебору всіх вершин дерева всіх перетинань. Кількість перебору вершин можна скоротити, зробивши стрибок з поточної вершини в вершину, відповідну ДСМ-замикання локальної схожості поточної вершини. Однак, при такому стрибку в піддереві поточної вершини будуть «пропущені» всі вершини, локальна подібності яких відповідає ще не знайденим глобальним подібностям. Пропуск даних вершин може привести до знаходження алгоритмом не всіх глобальних подібностей (тобто до втрати повноти при перерахуванні всіх подібностей).

Додатковою проблемою даного підходу є повторне породження однакових глобальних подібностей через те, що вершин, з яких повинен бути проведений стрибок, тобто вершин з однаковим набором, може бути будь-яка кількість, при цьому стрибок не гарантує виключення розкриття всіх таких вершин.

Таким чином, стратегією, що лежить в основі використовуваного в даній роботі алгоритму пошуку глобальних подібностей повинна бути стратегія 2, як оптимальна. При використанні стратегії 2 при розробці алгоритму пошуку всіх глобальних подібностей необхідно вирішити дві наступні завдання:

1. Забезпечити можливість стрибка в вершину, відповідну ДСМ замикання локальної схожості поточної вершини, і врахувати вершини з пропущеними під час стрибка об'єктами.

2. Обмежити кількість стрибків до одного в вершину, відповідну глобальній подібності.

На сьогоднішній день існує велика кількість алгоритмів, що реалізують дерева рішень: CART, C4.5, CHAID, CN2, NewId, ITrule і інші [29].

Алгоритми побудови дерев рішень відрізняється такими характеристиками:

- вид розщеплення - бінарне (binary), множинне (multi-way);
- критерії розщеплення - ентропія, Gini, інші;
- можливість обробки пропущених значень;
- процедура скорочення гілок або відсікання;
- можливості вилучення правил з дерев.

Жоден алгоритм побудови дерева не можна апріорі вважати найкращим або досконалим, підтвердження доцільності використання конкретного алгоритму має бути перевірено і підтверджено експериментом.

Переваги дерев рішень:

- простота та інтерпретація класифікації. Алгоритм здатний не тільки класифікувати об'єкт, але і видати пояснення класифікації в термінах предметної області. Пояснення будується шляхом вписування послідовності умов, перевічених для даного об'єкта на шляху від кореня дерева до листа v . Ці умови утворюють кон'юнкцію K_v , тобто легко інтерпретується логічне правило.

- алгоритм конструювання дерева рішень не вимагає від користувача вибору вхідних атрибутів (незалежних змінних). На вхід алгоритму можна подавати всі існуючі атрибути, алгоритм сам вибере найбільш значущі серед них, і тільки вони будуть використані для побудови дерева.

- точність моделей, створених за допомогою дерев рішень, порівнянна з іншими методами побудови класифікаційних моделей (статистичні методи, нейронні мережі).

- швидкий процес навчання. На побудову класифікаційних моделей за допомогою алгоритмів конструювання дерев рішень потрібно значно менше часу, ніж, наприклад, на навчання нейронних мереж.

- більшість алгоритмів конструювання дерев рішень мають можливість спеціальної обробки пропущених значень.

- класичні статистичні методи, за допомогою яких вирішуються завдання класифікації, можуть працювати тільки з числовими даними, в той час як дерева рішень працюють і з числовими, і з категоріальними типами даних.

Недоліки дерев рішень:

- застосований для навчання дерев рішень алгоритм рекурсивного секціонування за своїм характером є мізерним. У разі невдалого вибору алгоритм не здатний повернутися на рівень вгору і замінити невдалий предикат. Цей алгоритм показує відносно високу продуктивність, але зазвичай не дозволяє досягти оптимальних результатів з точки зору якості і розміру дерева.

- в процесі навчання може статися надмірно ретельна підгонка. Для усунення цього недоліку може використовуватися ще один етап обробки дерева, на якому здійснюється його усічення, але для цього потрібні додаткові обчислення і додаткові дані, що дозволяють провести атестацію результатів. Але, на жаль, інтегровані алгоритми втрачають таку перевагу, як простота.

- основна причина недоліків - неоптимальність жадібної стратегії нарощування дерева. Для їх усунення застосовують різні евристичні прийоми: редукцію, елементи глобальної оптимізації, «прогахунок вперед» (look ahead), побудова сукупності дерев.

Для застосування даного алгоритму всі об'єкти множини S повинні бути пронумеровані. Порядок на об'єктах з S індукує також лексикографічне упорядкування множин з $2S$.

В описі алгоритму додатково використовуються наступні умовні позначення:

Y - поточна вершина.

Нижче представлена функціональна схема алгоритму П-3-bt.

Породження перетинів даним алгоритмом являє собою процес побудови деякого дерева, що відбувається зверху вниз (вершинам дерева відповідають глобальні подібності) відповідно до таких кроків (в процесі побудови і обходу

дерева об'єкти з S можуть стати поміченими або залишатися непоміченими незалежно в кожній вершині) [3]:

Крок 0. Коренева вершина дерева пов'язана з m вершинами, кожна з яких взаємно однозначно відповідає якому - якого об'єкта з S . Всі об'єкти у всіх вершинах непомечени, $Y := \emptyset$.

Крок 1. Беремо перший з непомічених об'єктів безлічі S для поточної вершини Y . Нехай – це X_i . Вираховуємо $(\{Y \cup X_i\})'$ і $(\{Y \cup X_i\})''$. Утворити нову вершину відповідну $(\{Y \cup X_i\})''$. Зв'яжемо її ребром з вершиною, що відповідає Y .

Крок 2. Якщо в замиканні $(\{Y \cup X_i\})''$ є хоча б один новий по відношенню до Y об'єкт з номером меншим ніж i , то глобальне схожість, відповідне замикання $(\{Y \cup X_i\})''$, було вже породжене раніше. Помічаємо всі об'єкти множини S в вершині $(\{Y \cup X_i\})''$. Ця гілка дерева таким чином не продовжуватиметься. Якщо глобальне схожість які раніше не породжувалося, то помітимо додатково в вершині Y об'єкт X_i і в вершині $(\{Y \cup X_i\})''$ всі об'єкти безлічі $(\{Y \cup X_i\})''$.

Крок 3. Якщо всі об'єкти множини S виявилися поміченими в $(\{Y \cup X_i\})''$, то переходимо до кроку 4. Інакше - $(\{Y \cup X_i\})''$, переходимо до Кроку 1.

Крок 4. Повертаємося по дереву вгору аж до найближчої вершини, в якій є непомічені об'єкти з S . Якщо така вершина є, і вона відповідає безлічі об'єктів Z , то $Y := Z$, переходимо до кроку 1. Якщо такої вершини немає, то всі глобальні подібності побудовані, і алгоритм закінчує роботу.

Переваги алгоритму «замикає по одному»:

Алгоритм здійснює пошук всіх глобальних подібностей, не перебираючи всі вершини дерева всіх перетинів (можливість здійснення переходу (тобто зміна поточної вершини стрибком вперед) і обліку вершин з пропущеними під час переходу об'єктами дозволяють здійснити пошук всіх глобальних подібностей щодо багатьох об'єктів, виключивши з перебору вершини, множина утворюючих яких збігаються з множиною утворюючих вже розглянутих вершин).

Недоліки алгоритму «замикай по одному»:

1. Алгоритм має підвищені вимоги до пам'яті через необхідність обробки додаткових масивів даних, що зберігають інформацію про помічені і не помічені об'єкти множини S для кожної вершини.
2. Алгоритм має додаткові обчислювальні витрати для визначення унікальності породженої глобальної подібності (Здійснюється перебір індексів ведених об'єктів отриманої глобальної подібності [3]).
3. Алгоритм не передбачає дій в тому випадку, коли перетин об'єктів вершини відповідає порожній безлічі.
4. Таким чином, алгоритм «замикай по одному» не відповідає висунутим до алгоритму пошуку глобальних подібностей вимогам, так як:
5. Має значні додаткові обчислювальні витрати і підвищені вимоги до обсягу необхідної пам'яті;
6. Потребує доопрацювання для застосування на практиці, так як не передбачає дій в тому випадку, коли перетин об'єктів вершини відповідає порожній множині.

3 МОЖЛИВОСТІ ТА ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ДСМ – МЕТОДУ В ЗАДАЧАХ РОЗГЛЯНУТОГО ТИПУ

3.1 Неформальна семантика ДСМ-методу АПГ

Уміння знаходити закономірності є однією з фундаментальних людських якостей, мабуть її можна порівняти лише зі здатністю до членороздільної мови. Для того, щоб виявити емпіричну закономірність існує потреба у проведенні деякої розумової роботи, відмінної за характером від обчислень і математичних доказів: необхідно зіставити факти, знайти схожість в предметах і явищах, зробити індуктивні узагальнення і висновки по аналогії. Ця розумова робота являє собою правдоподібні міркування.

Як правило, в правдоподібних міркуваннях, спрямованих на виявлення причинно-наслідкових зв'язків (відносин), загальний висновок робиться на підставі особистісних припущень. Тобто в цьому випадку ми маємо справу з індуктивними міркуваннями. Класичне формулювання таких міркувань належить Д.С. Міллю.

У 1843 році вийшла книга Д.С. Мілля «Система силлогістичної та індуктивної логіки» [6], в якій він систематизував і розвинув вчення про індукцію як логічний засіб пізнання. Д.С. Мілля вважав засновником філософії індукції Ф. Бекона [8], попередниками систематичної теорії індукції були також Д. Гершель [13] і У. Уевелл [14], хоча вони були сучасниками Д.С. Мілля.

На відміну від Д. Гершеля і У. Уевелла, Д.С. Мілля намагався створити і обґрунтувати не тільки загальні принципи індукції, але і сформулювати правила індуктивних міркувань як логічні засоби («фігури виведення») подібні до доказам цих тверджень, що представляють знання, витягнуте з фактів. Ці факти утворюють множину принципів індуктивного міркування (reasoning).

В третій книзі, VIII главі («Чотири методу вдалого дослідження») Д.С. Мілля сформулював п'ять своїх знаменитих правил індуктивного міркування, що відрізняються від схеми виведення перераховування індукції. Він назвав їх, відповідно, методом подібності, методом відмінності, сполученим методом

подібності-відмінності, методом залишків і методом супутніх змін. Нижче наведені формулювання даних методів:

1. Метод подібностей. Якщо два або більше випадків мають лише одну загальну обставину, то ця обставина, в якій узгоджуються всі ці випадки, є причиною чи наслідком цього явища.

2. Метод відмінності. Якщо випадок, в якому досліджуване явище відбувається, і випадок, в якому воно не відбувається, подібні у всіх обставинах, крім одної, що зустрічається лише в першому випадку, то ця обставина, в якій тільки і різняться ці два випадки, є наслідок, або причина, або необхідна частина причини явища.

3. Сполучний метод подібності та відмінності. Якщо два або більше випадки виникнення явища мають лише одну загальну обставину, і два або більше випадків не виникнення того ж явища.

4. Метод залишків. Якщо з явища відняти ту його частину, яка, як відомо з колишньої індукції, є наслідком деяких попередньо визначених обставин, то залишок цього явища повинен бути наслідком попередніх обставин.

5. Метод супутніх. Будь-яке явище, що змінюється певним чином щоразу, коли змінюється інше явище - є причиною або наслідком цього явища, або пов'язане з ним будь-яким причинним зв'язком.

За наступні більш ніж півтора століття було зроблено цілий ряд спроб формалізувати семантику Міллевських канонів засобами математичної логіки. Так деякі з методів Д.С. Мілля були формалізовані засобами двозначної логіки Г. Грневскім [15, 16]. В.К. Фіном була висловлена ідея використання багатозначних логік для формалізації індуктивних методів Д.С. Мілля, були опубліковані результати застосування комп'ютерної програми, що реалізує формалізацію методу подібності, для передбачення біологічної активності хімічних сполук. Ця програма, а також наступні її посилення представляли метод автоматичного породження гіпотез в базах даних з неповною інформацією. Цей метод був названий на честь Д.С. Мілля ДСМ-методом автоматичного породження гіпотез

(ДСМ-метод АПГ), описані вище правила індуктивних міркувань Д.С. Мілля послужили відправною точкою для розробки ДСМ-методу.

Досить докладний опис поточного стану розвитку ДСМ-методу АПГ можна знайти, наприклад, в роботах [20, 22].

ДСМ-метод АПГ має п'ять основних компонентів:

1) умови застосування [18];

2) ДСМ-міркування;

3) уявлення знань у вигляді відкритих квазіаксіоматичних теорій (КАТ);

4) метатеоретичні принципи і засоби дослідження міркувань і предметних областей (в тому числі дедуктивна імітація міркувань, процедурна семантика [20] і препроцесінг, результатом якого є вибір стратегій міркування і відповідної їм процедурної семантики);

5) інтелектуальні системи типу ДСМ (ІС-ДСМ) [17]

Важливо відзначити, що однією з головних ідей ДСМ-методу АПГ є формалізація взаємодії трьох пізнавальних процедур (трьох різновидів правдоподібних міркувань) - індукції, аналогії і абдукції. Ця взаємодія здійснює узгодження ідей Д.С. Мілля про індукції з абдукції Ч.С. Пірса [23, 24], вимогою фальсифікації породжуваних гіпотез К.Р. Поппера [21] і прагненням використовувати правдоподібні міркування для knowledge discovery згідно Д. Пойа [25].

Формальна структура ДСМ-методу АПГ приведена в Додатку В1.

3.2 Алгоритміка ДСМ- методу АПГ

Основні визначення і позначки, що використовуються в даному розділі зазначені в Додатку В.2.

3.2.1.Опис алгоритму пошуку глобальних подібностей

3.2.1.1 Основні вимоги до алгоритму пошуку глобальних подібностей

Вибір стратегії 2 (аналіз основних стратегій пошуку гіпотез представлений в Додатку В.4) породження гіпотез є підставою для висунення таких вимог, яким повинен задовольняти використовуваний алгоритм пошуку глобальних подібностей:

1.Забезпечити можливість скачка в вершину, відповідну ДСМ-заміку локальної схожості поточної вершини, і враховувати вершини з пропущеними під час стрибка об'єктами .

2. Обмежувати кількість стрибків до одного у вершину, відповідну глобальній подібності .

Крім того, необхідно забезпечити виконання наступних додаткових вимог:

1.Мінімізувати обчислювальні витрати і необхідний обсяг пам'яті при реалізації завдань 1 і 2.

2. Забезпечити можливість практичного застосування даного алгоритму без додаткової модифікації.

У Додатку В.4 представлено опис чинних алгоритмів пошуку глобальних подібностей, що реалізують стратегію 2, а також проведений аналіз даних алгоритмів на задоволення вимог, що пред'являються до алгоритму пошуку глобальних подібностей.

Умовами застосування ДСМ - методу АПГ є:

1. Припущення про існування в базі фактів (БФ) ІС-ДСМ позитивних прикладів досліджуваного явища ((+) - фактів) і його негативних прикладів ((-) - фактів);

2 . Структурування явища за допомогою його уявлення як об'єкта, що володіє ефектом (безліччю властивостей), таким чином, що визначно схожість (+) - фактів і (-) - фактів;

3. Припущення про те, що в БФ в неявному вигляді існують позитивні і негативні залежності причинно-наслідкового типу (відповідно, (+) - і (-) - причини досліджуваних в БФ ефектів);

4. Числа наявних у БФ (+) - фактів і (-) - фактів представлені параметром k таким, що $k^+ \geq 2$ і $k^- \geq 2$, де k є нижня межа якого призначає параметра, змінюваного відповідно до умов експерименту. Умови 1 ° - 4 ° використовуємо для формалізації міллевського методу подібності [6, стор. 305-307].

Для формалізації методу подібності була сформульована наступна мова подання знань (ДСМ-мова), за допомогою якої виражається як схожість (+) - фактів і (-) - фактів, ставлення причина - наслідок, так і оцінки (ступеня правдоподібності) фактів і гіпотез (породжувані гіпотези є наслідками цього правила):

- X, Y, V (можливо, з нижніми індексами) - змінні для об'єктів і підоб'єктів;
- $C, C_1, C_2 \dots$ - константи (множини елементів), що є значенням змінних для об'єктів і підоб'єктів;
- Y, U, W (можливо, з нижніми індексами) - змінного для ефектів (множини властивостей);
- $Q, Q_1, Q_2 \dots$ - константи (безлічі властивостей), які є значеннями змінних Y, U, W , і т.д. ;
- m, n, l, k, r, s (можливо, з нижніми індексами) - змінні, значеннями яких є натуральні числа ($n \in \mathbb{N}$);
- \cup, \cap - операції алгебри множин;
- $=$ - предикати рівності для змінних, наведених вище трьох сортів змінних;
- \leq, \geq - предикати для числових змінних;
- \square предикат включення для множин (підоб'єктів і об'єктів і множин властивостей);
- $X \Rightarrow_1 Y$ предикат «об'єкт X має безліч властивостей Y »;
- $X \Rightarrow_2 W$ предикат « V є причина W »;
- \neg, \wedge, \vee - логічні зв'язки двозначної логіки;

– J_v - J-оператори Россера-Тюркетта [18], де $v = \langle v, n \rangle$ або $v = \langle \tau, n \rangle$, $v \in \{1, -1, 0\}$, $n \in \mathbb{N}$ (\mathbb{N} - безліч натуральних чисел); 1, -1, 0, τ - типи значень «фактична істина», «фактична брехня», «фактичне протиріччя» і «невизначеність», відповідно;

– \forall, \exists - квантори загальності та існування (відповідно, для наведених вище трьох сортів індивідуальних змінних).

Терми і формули ДСМ-мови визначають стандартним чином, але з істотною зміною формул «змінної довжини» і кванторів по кортежам «змінної довжини».

Справа в тому, що при пошуку емпіричних залежностей в БФ потрібно встановити подібність або відмінність фактів на кінцевому, але заздалегідь безмежну кількість прикладів. Число таких прикладів k , отже, є змінною величиною (k називається параметром емпіричної індукції). Ця обставина вимагає розширити мову логіки предикатів 1-го порядку, ввівши формули «змінної довжини» і квантори по кортежам [26], [18, Частина I, Глава 3: Д.П. Скворцов «Про деякі способи побудови логічних мов з кванторами по кортежам», стор. 214-232]. ДСМ-мову з кванторами по кортежам є мовою слабкою логіки предикатів 2-го порядку [27], в якому виразність транзитивне замикання.

3.2.1.2 Модифікований алгоритм пошуку глобальних подібностей

3.2.1.2.1 Основні ідеї, покладені в основу модифікованого алгоритму

Для застосування даного алгоритму, також як і для описаних вище алгоритмів, всі об'єкти множини S повинні бути пронумеровані. Порядок на об'єктах з S індукує також лексикографічне упорядкування множин з 2^S , тому об'єкти орбіт всіх шуканих глобальних подібностей і локальних подібностей, відповідних вершин дерева всіх перетинів, також впорядковані. Нижче представлено опис стратегії проходу вершин з пропущеними під час стрибка об'єктами, що не вимагає додаткових обчислювальних витрат і додаткових обсягів пам'яті.

Нижче представлено опис стратегії проходу вершин з пропущеними під час стрибка об'єктами, що не вимагає додаткових обчислювальних витрат і додаткових обсягів пам'яті.

В орбіті кожної вершини, відповідної глобальній подібності, вже «закодовані» всі пропущені об'єкти. Таким чином, «декодуючи» орбіту глобальної подібності, можна «розширювати» поточну вершину по траєкторії, «що обгинає» враховані вершини (тобто вершини, набір утворюючих локальних подібностей, які відповідають набору схожості вихідної вершини стрибка) і враховує вершини з пропущеними під час стрибка об'єктами. Для цього достатньо розширювати поточну вершину об'єктами, відсутніми в її орбіті, тоді набори утворюють отриманих подібностей будуть збігатися з наборами утворюють вершин з пропущеними під час стрибка об'єктами. Для цього достатньо розширювати поточну вершину об'єктами, відсутніми в її орбіті, тоді набори будуть збігатися з наборами утворених вершин з пропущеними під час стрибка об'єктами. Для визначення безлічі всіх пропущених об'єктів \bar{Y} достатньо знати орбіту Y поточної вершини і вихідна безліч об'єктів P .

Таким чином, описана стратегія проходу вершин з пропущеними під час стрибка об'єктами не вимагає створення додаткових масивів даних. Крім того, вона вимагає значно менших обчислювальних витрат, ніж стратегії розглянутих алгоритмів, так як «розширення вершини на основі «декодування» орбіти проводиться за рахунок пошуку об'єктів, відсутніх в орбіті вершини, на основі простих логічних правил, що визначають присутність поточного розширення об'єкта в орбіті вершини чи ні.

Це вимагає перебору тільки присутніх в орбіті глобальних подібностей об'єктів, причому, починаючи з наступного за останнім знайденим розширеним об'єктом, що значно менш затратно (в частині необхідних обсягів обчислень), ніж перебір в алгоритмі «замикає по одному» при пошуку непоміченого об'єкта.

Нижче представлено опис стратегії проходу вершин з пропущеними під час стрибка об'єктами, що не вимагає додаткових обчислювальних витрат і додаткових обсягів пам'яті. За логікою алгоритму "замикай по одному", єдиність

стрибка в вершину відповідну глобальному подібністю можна забезпечити, порівнюючи ведені об'єкти орбіти глобальної подібності з провідним об'єктом. Однак, даний підхід можна в значній мірі оптимізувати.

На відміну від алгоритму "замикай по одному" об'єкти орбіти глобальної подібності завжди впорядковані. При появі веденого об'єкта, індекс якого менше, ніж індекс розширення, ведений об'єкт займає позицію в орбіті глобальної подібності лівіше, ніж розширення, тим самим "виштовхуючи" розширюваний об'єкт вправо в порівнянні з позицією в орбіті вихідної вершини стрибка. Таким чином, для реалізації обмеження кількості стрибків у вершину, відповідну глобальній подібності, до одного, досить порівнювати позиції розширюваного об'єкта у вихідній вершині стрибка і знайденої глобальної схожості. Стрибок здійснюється тільки в разі, якщо дана позиція не змінилася, тобто не з'явилося ведених об'єктів з меншим індексом, ніж розширюваний.

Таким чином, контроль єдиності стрибка в алгоритмі реалізується однією операцією порівняння.

3.2.1.2.2 Модифікований алгоритм пошуку глобальних подібностей:

основні визначення та умовні позначення

Нижче наведені основні умовні позначення і визначення, що використовуються при описі модифікованого алгоритму:

P - кінцева безліч об'єктів, щодо яких здійснюється пошук глобальних подібностей;

Дерево Перетинань - дерево, вершинами якого є локальні і глобальні подібності щодо багатьох об'єктів.

Для будь-якої побудованої вершини Дерева Перетинань зберігається наступна інформація:

- Y - безліч об'єктів даної вершини;
- h - безліч утворюють, відповідне перетину (Операційно

обчислюваному подібністю) об'єктів вершини;

- i - індекс об'єкта розширення вершини в P ;
- j - індекс, використовується для визначення об'єкта розширення;

Об'єкт розширення - об'єкт X безлічі P , що використовується для переходу від вершини $\{\{Y, h\} i, j\}$ до вершини $\{\{Y \cup X, h \cup X\}, i, j\}$.

При описі алгоритму використовуються три типи вершин Дерева Перетинань:

Коренева вершина - вершина, безліч об'єктів і безліч утворюючих одиниць якої порожні, в початковому стані індекс об'єкта розширення дорівнює нулю, а індекс j дорівнює одиниці.

Поточна вершина - вершина, яку необхідно «розширити» об'єктом розширення (умовне позначення - $\{\{Y, h\} i, j\}$).

Наступна вершина - прямий нащадок кореневої вершини, причому орбіті даної вершини належить тільки один об'єкт.

Вершина замикання - вершина, яка утворилася після «Розширення» поточної вершини об'єктом розширення (умовне позначення - $\{\{Y_c, h_c\} i_c, j_c\}$).

Для застосування даного алгоритму всі об'єкти множини повинні бути пронумеровані, починаючи з 1.

$X_i = P(i)$ – об'єкт з індексом i в множині P ;

P^{-1} – безліч індексів об'єктів з множини P ;

$i = P^{-1}(X)$ – індекс, що відповідає об'єкту X із P ;

3.2.1.2.3 Функціональна схема розробленого алгоритму пошуку глобальних подібностей

Вхідний алгоритм:

$P = \{p_1, \dots, p_n\}$ - множина об'єктів.

Поточною вершиною є коренева вершина $\langle Y, h, i=1, j=0 \rangle$.

Вихід алгоритму: безліч всіх непустих вичерпних перетинів - подібностей (разом з породжуючими їх об'єктами), тобто множина всіх глобальних подібностей для елементів множини P .

Породження глобальних подібностей даним алгоритмом являє собою процес побудови дерева перетину відповідно до наступних кроків:

КРОК 1: Пошук індексу розширюючого об'єкта для поточної вершини.

Якщо $i < P$, то об'єкт розширення для поточної вершини $\{Y, h, i, j\}$

існує, необхідно:

- задати можливий індекс об'єкта розширення i : $i = i + 1$;

Якщо $j < Yi = P^{-1}[Y(j)]$, то індекс об'єкта розширення відмінний від індексу i , необхідно:

- задати індекс j вершині: $j = j + 1$;
- перейти до повторного виконання кроку 1;

Коментар: індекс j поточної вершини зберігає позицію чергового з об'єктів, які присутні в орбіті вершини, якщо $j < |Y|$, а значить, були враховані під час стрибка. Якщо індекс розширює об'єкта збігається з індексом вже врахованого об'єкта ($i = P^{-1}[Y(i)]$), розширення вершини об'єктом з індексом i не має сенсу, так як отримана вершина буде мати той же набір утворюючих що i поточна. Необхідно здійснити перехід до початку кроку 1.

Інакше, якщо знайдений індекс пропущеного під час стрибка об'єкта, необхідно здійснити перехід до кроку 2.

Інакше, коли розширення об'єкта для поточної вершини $\{Y, h, i, j\}$ не існує, здійснюємо бектрекінг, тобто повернення до останньої з побудованих вершин дерева перетинів, у якій $i < |P|$ і встановлюємо її як поточну. Переходимо до виконання кроку 1. Якщо таких вершин немає, то алгоритм завершує роботу.

КРОК 2: Визначення глобального подібності.

Визначимо об'єкт розширення для поточної вершини: $X_i = P(i)$;

Якщо, Y то поточна вершина не є кореневою, визначимо набір глобальних подібностей, відповідних вершині замикання: $h_c = h \cap X_i$

Інакше, поточна вершина є кореневою, необхідно:

- створити таку вершину $\{Y_n = \{X_i\}; h_n = X_i; i_n = i; j_n = 2\}$;
- встановити посилання з поточної вершини $\{\{Y, h\}, i, j\}$ на наступну вершину $\{Y_n; h_n; i_n; j_n\}$;
- встановити наступну вершину $\{Y_n; h_n; i_n; j_n\}$; в якості поточної і перейти до виконання кроку 1.

Коментар: прямі нащадки кореневої вершини будуть містити тільки один об'єкт в орбіті ($Y_c = \{X_i\}$). При пошуку замикання для об'єктів орбіти вершини може виявитися, що об'єкти орбіти відповідають замиканню. У цьому випадку алгоритм породить глобальну схожість, орбіту яка буде складатися з одного об'єкта. Щоб виключити даний випадок, здійснюється перехід до пошуку об'єкта розширення для даної вершини замість пошуку замикання об'єктів її орбіти.

Якщо $h_c = \emptyset$, то продовжуємо розширення поточної вершини новим об'єктом розширення, тобто переходимо до кроку. Інакше, коли множина h_c не порожня, необхідно:

- визначити орбіту множини, відповідної вершині замикання;

Якщо $Y_c(j) = X_i$ – позиція розширюючого об'єкта не змінилася в орбіті вершини замикання в порівнянні з позицією в орбіті, відповідній розширенню поточної вершини, необхідно:

- визначити індекси для вершини замикання можна за допомогою наведеного способу: $i_c = i; i_c = j + 1$;
- створити вершину замикання $\{Y_c; h_c; i_c; j_c\}$
- встановити зв'язок з поточної вершини $\{\{Y, h\}, i, j\}$ на вершину замикання $\{Y_c; h_c; i_c; j_c\}$
- додати глобальнт $\{Y_c; h_c\}$ в список результатів $R = R \cup \{Y_c; h_c\}$
- встановити вершину замикання $\{Y_c; h_c; i_c; j_c\}$ в якості поточної і перейти до виконання кроку 1.

Коментар: якщо позиція розширюваного об'єкта не змінилася в орбіті вершини замикання в порівнянні з позицією в орбіті, відповідної розширенню поточної вершини $Y_c(j) = X_i$, то така глобальна схожість знайдена вперше.

Тобто якщо позиція розширюваного об'єкта змінилася в орбіті вершини замикання в порівнянні з позицією в орбіті, відповідної розширенню поточної вершини, необхідно здійснити перехід до кроку 1.

Коментар: глобальна схожість $\{Y_c; h_c\}$, вже була знайдена раніше.

3.2.1.2.4 Доведення коректності розробленого алгоритму

Розроблений алгоритм коректний, тобто забезпечує повноту (перераховуючи всі шукані подібності) і точність (не формуючи об'єктів, які не є подібностями) породження множини глобальних подібностей щодо безлічі об'єктів P .

1. Розроблений алгоритм проходить всі вершини дерева перетинів з поданої на вході множини P , яким відповідають всі варіанти наборів.

Алгоритм виключає побудову частини вершин, неунікальність набору яких може бути визначена. Частина вершин, в яких повторюються вже знайдені варіанти набору, відсіюються ще в ході першого стрибка. Інша частина вершин відсіюється за рахунок виключення можливості повторного потрапляння в вершину вже знайденої глобальної подібності.

Незважаючи на введення можливості виключення побудови частини вершин, розроблений алгоритм забезпечує повноту породження глобальних подібностей, так як не пропускає неунікальні вершини, навіть якщо ці вершини були пропущені під час стрибка.

Набір утворює і відповідна йому орбіта (множина об'єктів, в які входять всі елементи поточного набору) визначаються в список результатів тільки в тому випадку, коли орбіта є замиканням набору і коли шлях знаходження вершини, яка відповідає цьому замиканню, є канонічною. Таким чином, кожен елемент списку результатів - глобальна схожість, що гарантує точність алгоритму. Крім того,

контроль канонічності шляху забезпечує унікальність кожного із знайдених глобальних подібностей в списку результатів.

3.2.1.2.5 Оцінка обчислювальної складності розробленого алгоритму

Нехай на вхід алгоритму надходить безліч об'єктів потужністю $|P| = m$, при цьому кількість глобальних подібностей рівна K . Розроблений алгоритм з усіх гілок дерева «всіх перетинань», що призводять до певної глобальної подібності, продовжує лише одну гілку, відповідну канонічним шляхом. Таким чином, згідно з [3] кількість глобальних подібностей, вироблених даним алгоритмом, є $O(mK)$. При підрахунку тимчасової складності необхідно врахувати складність породження кожної вершини, якій відповідає глобальна схожість. Таким чином, оцінка швидкодії алгоритму пошуку глобальних подібностей складається з часу породження подібності і часу перевірки того, що дана глобальна подібність не була породжена раніше. Породження глобальної подібності вимагає виконання однієї операції перетину \cap і не більше m перевірок на вкладення, тому кількість цих операцій для всіх глобальних подібностей має оцінки $O(mK)$ і (Om^2K) відповідно. Для перевірки канонічності шляху потрібно одна операція порівняння тобто всього mK операцій порівняння за всю роботу алгоритму загальна тимчасова складність розробленого алгоритму має оцінку (3.1):

$$t_{cost} = O(mK) * t_{int} + O(m^2K) * t_{incl} + mK * t_{comp} \quad (3.1)$$

де t_{cost} – загальна тимчасова складність;

t_{int} – оцінка часу виконання операції перетину;

t_{incl} – оцінка часу виконання операції перевірки;

t_{comp} – час перевірки унікальності глобального подібності;

Алгоритм «замикає по одному» має наступну тимчасову складність (3.2) [3]:

$$t_{cost} = O(mK) * t_{int} + O(m^2K) * t_{incl} + O(m^2K) * t_{0-1} \quad (3.2)$$

яка перевищує складність розробленого алгоритму на $O(m^2K) * t_{0-1} - m * K * t_{comp}$ так як для контролю унікальності глобальної подібності в даному алгоритмі потрібно m^2K – операції з 0-1 векторами.

3.2.2 Опис алгоритмів основних етапів ДСМ методу

Слідуючи ідеям, викладеними в [7], в ДСМ-методі як в інтелектуальній системі було виділено дві частини *Leaner* і *Solver*. Границю між *Leaner* и *Solver* можна описати так:

- *Leaner* породжує деякі знання, а *Solver* їх використовує;
- *Solver* може породжувати нові стани вибірок для повторного використання *Leaner*;
- в сенсі ДСМ-методу розділити роботу *Leaner* і *Solver* можна наступним чином:
- *Leaner* породжує всі глобальні подібності (+) - і (-) - прикладів і застосовує правила першого роду, тобто формує базу знань;
- *Solver* на основі породженої *Leaner* бази знань застосовує правила другого роду і визначає об'єкти, породжуючи тим самим новий стан, на якому може знову працювати *Leaner*.

Традиційно важливу частину при описі алгоритмики ДСМ методу займає опис алгоритму пошуку всіх перетинів (побудова всіх глобальних подібностей). Дійсно, це центральний алгоритм всієї системи, він має найбільшу складність, і час роботи системи здебільшого визначається часом роботи цього алгоритму.

Детально цей алгоритм, оцінка його тимчасової складності і порівняння його з іншими алгоритмами пошуку всіх глобальних подібностей описані в пункті 3.2.1.

Нижче представлено опис алгоритмів, реалізованих в *Leaner* і *Solver*

3.2.2.1 Опис алгоритму породження гіпотез

Для породження (+) - гіпотез всіх рангів і (-) - гіпотез для кожного класу необхідно вирішити такі завдання:

1. Завдання знаходження глобальних подібностей (+) - прикладів для кожного з класів;
2. Завдання знаходження глобальних подібностей (-) - прикладів для кожного їх класів;
3. Завдання визначення (+) - гіпотез всіх рангів для кожного з класів;
4. Завдання визначення (-) - гіпотез для кожного з класів;

Завдання 1 і 2 вирішуються застосуванням алгоритму пошуку глобальних подібностей, описаного в пункті 3.2.1, до відповідних множин (+) - і (-) - прикладів для кожного класу.

Завдання 3 вирішується шляхом перевірки умов, описаних в правилах першого роду 1-5 (див. П. 5.3).

Завдання 4 вирішується шляхом перевірки умов, описаних в 6-му правилі першого роду (див. П. 5.3).

Розглянемо розроблений алгоритм, який реалізує пошук гіпотез, тобто який реалізує правила правдоподібного виведення першого роду.

Введемо наступні позначення:

C – множина класів;

$c_i \in C$ – клас множини класів;

$L_j - j$ – схожість, відповідна розглянутому класу;

H_i^{k+} – результуюча безліч (+) - гіпотез i - того класу з рангом k ;

H_i^{-} – результуюче безліч (-) - гіпотез i - того класу;

Нижче представлена функціональна схема розробленого алгоритму пошуку гіпотез.

Вхід алгоритму: множина класів C і відповідні класам множини подібностей (+) - і (-) - прикладів.

Вихід алгоритму:

- множини (+) - гіпотез H_i^{k+} ;
- множини (-) - гіпотез H_i^{-} ;

Пошук гіпотез розробленим алгоритмом відбувається відповідно до наступних кроків:

КРОК 1. Розглядаємо черговий клас $c_i \in C$;

КРОК 2. Розглядаємо чергове (+) - схожість L_j , що відповідає класу c_i ;

Якщо розглядається (+) - схожість присутній серед (-) - подібностей, відповідних поточному класу, необхідно перевірити наступні умови:

Якщо розглянути (+) - подібність містить всього один клас, що відповідає умові правила першого роду, тобто схожість є (+) - гіпотезою другого рангу, заносимо гіпотезу в відповідну множину $H_i^{+2} = H_i^{+2} \cup \{L_j\}$.

Якщо розглядається (+) - подібність містить два класи, що відповідає умові правила першого роду 3, тобто схожість є (+) - гіпотезою третього рангу, заносимо гіпотезу в відповідну результуючу множину $H_i^{+3} = H_i^{+3} \cup \{L_j\}$.

Якщо розглядається (+) - подібність містить більш ніж два класам, що відповідає умові правила першого роду 4, тобто схожість є (+) - гіпотезою четвертого рангу, заносимо гіпотезу в відповідну множину $H_i^{+4} = H_i^{+4} \cup \{L_j\}$.

Інакше, якщо розглядається (+) - не присутній серед (-) - подібностей поточного класу, що відповідає умові правила першого роду 1, тобто схожість є (+) - гіпотезою першого рангу, заносимо гіпотезу в відповідне результуюче множини $H_i^{+1} = H_i^{+1} \cup \{L_j\}$.

Якщо залишилися нерозглянуті (+) - подібності в класі c_i , переходимо до КРОКУ2. Якщо залишилися нерозглянуті класи в безлічі C переходимо до КРОКУ1.

КРОК 3. Розглядаємо черговий клас $c_i \in C$

КРОК 4. Розглядаємо чергове (-) – схожість L_j , відповідно класу c_i .

Якщо розглядається (-) - схожість відсутній серед (+) - подібностей розглянутого класу необхідно перевірити наступні умови:

Якщо розглядається (-) - схожість присутній серед (+) - подібностей молодших класів, що відповідає умові правила першого роду 5, тобто схожість є (+) - гіпотезою п'ятого рангу, заносимо гіпотезу в відповідну результуючу множину $H_i^{+5} = H_i^{+5} \cup \{L_j\}$.

Якщо розглядається (-) - схожість присутній серед (+) - подібностей старших класів, що відповідає умові правила першого роду 6, тобто схожість є (-) - гіпотезою, заносимо гіпотезу в відповідну множину $H_i^- = H_i^- \cup \{L_j\}$.

Якщо залишилися нерозглянуті гіпотези в класі c_i , переходимо на КРОК4. Якщо залишилися нерозглянуті класи в безлічі C , переходимо до КРОК3.

На малюнку 3.1 приведена діаграма дій, що ілюструє роботу розробленого алгоритму пошуку гіпотез. Переконаємося в коректності розробленого алгоритму пошуку гіпотез: цей алгоритм коректний, тобто забезпечує повноту і точність знаходження всіх (+) - і (-) - гіпотез кожного класу тривалості, тому що розглядає в процесі повного перебору все (+) - і (-) - подібності всіх класів тривалості. Таким чином гарантується повнота алгоритму.

3.2.2.2 Опис алгоритму класифікації

Для класифікації нового об'єкта необхідно застосувати правило виведення другого роду. Нижче представлено опис алгоритму, що реалізує правило другого роду.

Правило другого роду: об'єкт класифікується тим класом, в якому немає (-) - гіпотез, а наявні (+) - гіпотези володіють найбільшою силою, якщо таких класів кілька - вибирається найбільший з цих класів. Основна ідея алгоритму: здійснюючи перебір класів від старшого до молодшого можна уникнути перебору всіх гіпотез, що належать класам. Так, якщо в процесі такого перебору буде знайдений клас, який містить тільки (+) - гіпотези першого рангу і не містить (-) - гіпотез, відповідних досліджуваного об'єкта, то даний клас виявиться старшим з тих, які містять (+) - гіпотезу рангу максимальної сили і не містять (-) - гіпотез, тобто за визначенням буде задовольняти умові правила другого роду.

В інших випадках здійснюється повний перебір гіпотез. Якщо по завершенню повного перебору умови правила другого роду залишаються невиконаними, тобто не буде знайдено жодного класу, що містить (+) - гіпотезу і не містить (-) - гіпотез, об'єкт залишається не класифікованим.

Введемо наступні позначення:

C – множина класів тривалості;

H_i^{+k} – множина (+) - гіпотез i - того класу з рангом k ;

H_i^{-} – множина (-) - гіпотез i - того класу;

X – досліджуваний об'єкт;

$H_i^{+k}(X)$ – множина (+) - гіпотез i - того класу с рангом k , набори яких вкладаються в набір досліджуваного об'єкта X , $H_i^{+k}(X) \sqsubseteq X$;

$H_i^{-}(X)$ – множина (-) - гіпотез i - того набору, набори яких вкладаються в набір досліджуваного об'єкта X , $H_i^{-}(X) \sqsubseteq X$;

Нижче представлена функціональна схема розробленого алгоритму пошуку гіпотез. Вхід алгоритму: досліджуваний об'єкт X , який необхідно класифікувати в один з класів безлічі C .

Вихід алгоритму: прогнозований клас проекту.

КРОК 1. Розглядаємо черговий клас $c_i \in C$ (від старшого класу до молодшого).

$n = 0$ – встановлюємо значення змінної, яка в процесі роботи алгоритму зберігає номер старшого з класів, що містять відповідну для класифікації (+) - гіпотезу максимального рангу зі знайдених і не містить (-) - гіпотез.

$m = 6$ – встановлюємо значення змінної, за допомогою якої обчислюється максимальний ранг знайдених (+) - гіпотез, відповідних досліджуваного об'єкта X .

КРОК 2. Розглядаємо множину (+) – гіпотез H_i^{+k} чергового рангу k (починаючи з $k = 1$), що відповідає класу c_i .

Якщо розглянутий клас з номером i , містить (+) - гіпотези рангу k і при цьому не містить (-) - гіпотез, відповідно досліджуваного об'єкта ($H_i^{+k}(X) \neq \emptyset \wedge H_i^{-}(X) = \emptyset$), необхідно перевірити наступні умови:

Якщо ранг знайдених (+) - гіпотез є максимальним ($k = I$), відносимо досліджуваний об'єкт X до класу c_i тобто встановлюємо оцінку стосовно $X \Rightarrow_2^* c_i$ рівній $\langle I, n + I \rangle$ і завершуємо класифікацію.

Якщо $k < m$, тобто ранг знайдених (+) - гіпотез розглянутого класу вище, ніж ранг знайдених (+) - гіпотез старших класів.

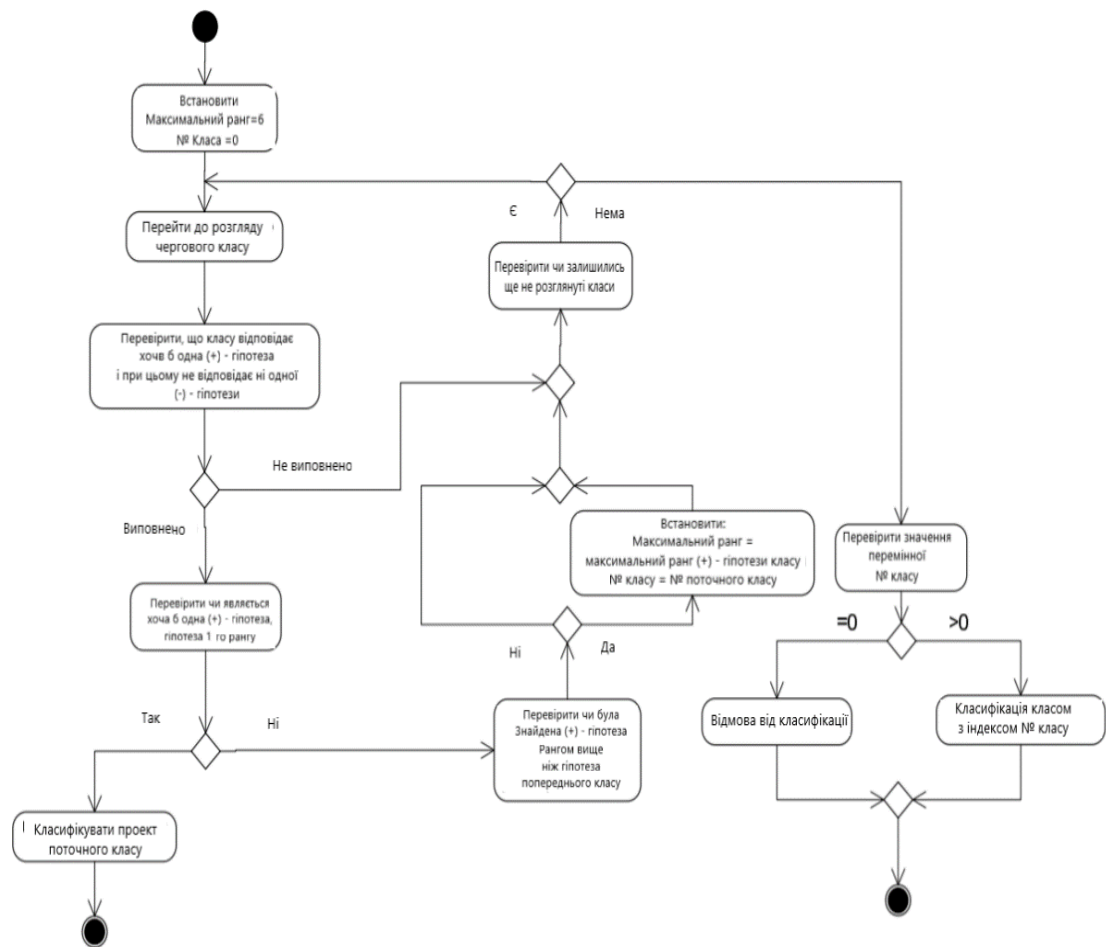
$m = k$ – зберігаємо ранг знайдених (+) - гіпотез;

$n = i$ – зберігаємо номер класу, який, на даний момент містить (+) - гіпотези найвищого рангу зі знайдених.

Якщо залишилися нерозглянуті класи в безлічі C , то переходимо до кроку 1. Якщо $n > 0$, тобто за час роботи алгоритму були знайдені відповідні для класифікації (+) - гіпотези, відносимо досліджуваний об'єкт X до класу c_n , тобто встановлюємо оцінку відношенню $X \Rightarrow_2^* c_n$ рівне $\langle I, n + I \rangle$, завершуємо

класифікацію. Інакше, тобто якщо не знайдено відповідних для класифікації гіпотез, проект залишається некласифікованим.

На малюнку 3.2 приведена діаграма дій, що ілюструє роботу розробленого алгоритму пошуку гіпотез. Нескладно перекоонатися в коректності розробленого алгоритму пошуку гіпотез: цей алгоритм забезпечує повноту і точність класифікації, тому що здійснює пошук всіх (+) і (-) - гіпотез, необхідних для класифікації досліджуваного об'єкта за допомогою правила другого роду, і безпосередньою класифікацію об'єкта або відмову від класифікації, в разі якщо не знайдено відповідних за умовою правила другого роду гіпотез. Процес пошуку гіпотез може бути завершений до завершення повного перебору, якщо певний клас тривалості містить хоча б одну (+) - гіпотезу 1-го рангу і при цьому не



містить (-) - гіпотез, відповідних досліджуваного об'єкта.

Рисунок 3.2 - Діаграма дій алгоритму, що реалізує пошук алгоритм класифікації об'єкта, в нотації UML2.0

Класифікацію досліджуваного об'єкта певним класом алгоритм здійснює тільки в тому випадку, якщо виконуються умови правила другого роду. Об'єкт залишається некласифікованих, якщо ця умова не виконується. Таким чином гарантується точність алгоритму.

Для застосування даного алгоритму всі об'єкти множини S повинні бути пронумеровані. Порядок на об'єктах з S індукує також з лексикографічним упорядкуванням множин з $2S$.

Нижче представлена функціональна схема алгоритму П-3-bt.

Послідовним перебором уздовж всієї множини об'єктів S реалізується наступна процедура:

- спершу в список ПОТОЧНОГО НАБОРУ УТВОРЮЮЧИХ поміщається відповідний об'єкт з вихідної вибірки, потім послідовним перебором (в рамках стратегії перегляду спершу вглиб, потім вшир, уздовж відповідного піддерева) породжуються його перетину (подібності) з усіма іншими об'єктами;
- при цьому кожен раз, коли на поточному кроці обчислення подібностей отримано непорожній перетин ПОТОЧНОГО НАБОРУ УТВОРЮЮЧИХ і відповідного об'єкта, обчислюється замикання орбіти для щойно знайденої подібності;
- далі, якщо це замикання, а значить і відповідна глобальна схожість - породжені вперше, воно заноситься в список РЕЗУЛЬТАТІВ, здійснюється перехід (тобто "стрибок") в вершину дерева усіх перетинань з найбільшим відповідним знайденому замиканню орбіти номером. Одночасно в списках скачок ВПЕРЕД і скачок НАЗАД (мають індекс - номер поточного стрибка) запам'ятовуються номери тих об'єктів, які були "пропущені" при активному стрибку, і далі алгоритм знову породжує схожість ПОТОЧНОГО НАБОРУ УТВОРЮЮЧИХ і поточного (тобто наступного) об'єкта;
- якщо така схожість виявляється порожньою, то здійснюється

BACKTRACKING, в ході якого при завершеному поверненні за всіма раніше проведеними стрибками робиться повернення в попередню (по відношенню до поточного значення ПОТОЧНОГО НАБОРУ УТВОРЮЮЧИХ) вершину дерева всіх перетинань. При незавершеному поверненні за поточним стрибком за допомогою списків скачок ВПЕРЕД і скачок НАЗАД робиться перехід в попередню вершину з тих, які були "пропущені" при стрибку вперед, і процес породження поточних подібностей ітерується (по вже представленим вище "звичайним правилам", але з урахуванням того, які з об'єктів вже були раніше "прийняті до уваги" при стрибку і повинні бути виключені, щоб уникнути "заиклення процедури перегляду вершин дерева" з множини, що міститься - на поточному кроці алгоритму - всіх кандидатів на роль наступного об'єкта. Такий облік ведеться за допомогою списків НАСТУПНІ ДЛЯ ОБ'ЄКТА). При кожному поверненні в вихідну вершину стрибка відбувається оновлення відповідних допоміжних списків скачок ВПЕРЕД, скачок НАЗАД, НАСТУПНІ ДЛЯ ОБ'ЄКТА та ін .;

- якщо ж знайдене замикання орбіти поточного набору утворюють дублікат, перспектива стрибка вперед ігнорується і просто розглядається подібність ПОТОЧНОГО НАБОРУ УТВОРЮЮЧИХ і наступного об'єкта;
- в ситуації, коли схожість ПОТОЧНОГО НАБОРУ УТВОРЮЮЧИХ і наступного об'єкта виявляється порожньою, обчислюється схожість цього ж ПОТОЧНОГО НАБОРУ УТВОРЮЮЧИХ і наступного далі об'єкта, або ж (у разі, коли в ролі наступного був останній з початково заданих об'єктів), робиться повернення в попередню по відношенню до поточного значення ПОТОЧНОГО НАБОРУ УТВОРЮЮЧИХ вершину;
- при завершенні перегляду поточного піддерева дерева всіх перетинів також відбувається оновлення всіх названих допоміжних списків, робиться заміна поточного об'єкта на наступний, потім процес породження і аналізу подібностей ітерується до потрапляння в останню вершину Дерева.

Переваги алгоритму П-3-bt:

Алгоритм здійснює пошук всіх глобальних подібностей і облік вершин з пропущеними під час переходу об'єктами дозволяючи здійснити пошук всіх глобальних подібностей щодо багатьох об'єктів, виключивши з перебору вершини, безліч утворюючих яких збігається з множиною вже розглянутих вершин.

Недоліки алгоритму П-3-bt:

- Алгоритм має підвищені вимоги до пам'яті через необхідність зберігання додаткових масивів даних (тобто списків «Стрибок вперед», «Стрибок назад», «Наступні для об'єкта»).
- Алгоритм має додаткові обчислювальні витрати на обробку списків для вирішення завдання обліку вершин з пропущеними під час стрибка об'єктами.
- Алгоритм не виключає декількох стрибків у один і ту ж глобальну схожість.

Таким чином, алгоритм П-3-bt не відповідає висунутим до алгоритму пошуку глобальних подібностей вимогам, так як:

- не скорочує кількість стрибків в глобальну схожість до одного;
- має значні додаткові обчислювальні витрати і підвищені вимоги до обсягу необхідної пам'яті;

4 ПРИКЛАДНА ЗАДАЧА: ПОСТАНОВКА ТА РОЗВ'ЯЗАННЯ

4.1 Опис обраного інструментарію

4.1.1 Формування основних вимог до технології розробки

програмного засобу

На сьогоднішній день існує ряд ефективних технологій, застосовуваних для розробки ПО. Дані технології орієнтовані або на рішення однорідних завдань (наприклад, розробка тільки Web - додатків) або на вирішення комплексних завдань (наприклад, розробка Web, настільних і мобільних додатків). Можливість вирішення комплексних завдань вимагає наявності в даних технологіях розробки ПО ряду універсальних інструментальних засобів. При розробці обговорюваної в даній роботі ІС потрібно вирішити наступні завдання:

- розробка компонентів для операцій зі складними структурами даних;
- розробка компонентів, які забезпечують введення вихідних даних, їх редагування, обробку і розвинені засоби представлення результатів інтелектуального аналізу даних;
- забезпечення незалежності ІС від використовуваної реляційної СУБД.

Таким чином, неоднорідна множина завдань, які необхідно вирішити в ході розробки ІС, вимагає вибору технології розробки ПЗ, що підходить для вирішення комплексних завдань розробки і налагодження ПО.

Необхідність розробки графічного інтерфейсу для забезпечення можливості введення вихідних даних і представлення результатів їх інтелектуального аналізу вимагає великих затрат часу. Таким чином, обрана технологія розробки ПЗ повинна містити бібліотеку для розробки графічного інтерфейсу, яка надає широкий спектр стандартних графічних компонентів.

Крім того, здійснюючи вибір технології для розробки ПО, необхідно врахувати можливість виникнення потреб в модифікації розробленої ІС в майбутньому в процесі її експлуатації. Можливі зміни можуть бути пов'язані:

- з необхідністю розширення наявного функціоналу компонентів через нові вимоги;

- з переходом програми на рівень корпоративного додатка або корпоративного Web-дodatку (даний перехід може бути обумовлений необхідністю децентралізованого використання даної системи багатьма користувачами);
- із забезпеченням можливості перенесення розробленої ІС на нові платформи (мобільність співробітників може стати причиною виникнення потреби в розробці версії інформаційної системи для мобільної платформи, наприклад для таких платформ як Android, iOS, Windows Mobile, і.т.д.).

Обрана технологія розробки ПО повинна забезпечувати можливість реалізації всі представлені вище зміни.

Розумно припустити, що технологія розробки ПЗ, забезпечує реалізацію всіх вище перерахованих вимог, може виявитися дорогою і / або, що ще гірше, її вивчення і впровадження може потребувати великих тимчасових витрат. Дані факти можуть стати причиною виникнення неприйнятних тимчасових і матеріальних витрат на розробку обговорюваного ПО. Таким чином, вартість застосування обраної технології повинна бути мінімізована. Крім того, обрана технологія повинна забезпечувати мінімізацію часових витрат на розробку, впровадження та модифікацію створюваного програмного засобу. Виконання останньої вимоги можливо, якщо розробка всіх необхідних компонентів, пов'язаних з розробкою поточної версії ІС і її майбутніх версій, які враховують можливі модифікації, буде базуватися на одній мові програмування і буде відбуватися в єдиному середовищі розробки.

Резюмуючи вище викладене, можна сформулювати наступні основні вимоги до технології розробки ПЗ:

1. Наявність коштів, що забезпечують взаємодію з СУБД і підтримують організацію незалежності від конкретної реляційної СУБД.
2. Наявність бібліотеки для розробки графічного інтерфейсу, яка має широкий спектр стандартних графічних компонентів.

3. Обрана технологія повинна надавати можливість гнучкого розширення ІС, а так само можливість її еволюції до корпоративної системи і перенесення частини функціоналу на мобільні платформи.
4. Дана технологія повинна базуватися на одній мові програмування.
5. Вся необхідна розробка повинна відбуватися в єдиному середовищі розробки.

4.1.2 Вибір інструментарію, відповідного пред'явленим вимогам

На основі аналізу пред'явлених вимог, можна зробити висновок, що на даний момент існує дві комплексні технології розробки ПЗ, а саме .NET і Java, які слід проаналізувати при виборі підходящої технології розробки обговорюваного тут проблемно-орієнтованого ПО. Розроблені за допомогою даних технологій варіанти прикладного ПО не мають принципових функціональних відмінностей з точки зору якості та продуктивності. Крім того, обидві технології базуються на одній мові програмування, забезпечують наявність необхідних коштів, що реалізують технології взаємодії з СУБД і розробки графічного інтерфейсу і задовольняють висунутим вимогам. Однак технологія Java забезпечує кросплатформенних розроблених додатків, в той час як технологія .NET забезпечує функціонування тільки на платформі Windows. Крім того, середовище розробки і деякі технології .NET є платними і розповсюджуються з закритим вихідним кодом, що може ускладнити розробку. При цьому для застосування технології Java існує кілька безкоштовних середовищ розробки. Таким чином, в якості технології розробки ПЗ за доцільне використовувати технологію Java, так як вона повністю задовольняє всім висунутим вимогам (див пункт 4.1.1).

Нижче представлений вибір основних технологій, реалізованих в технології Java і забезпечують виконання пред'явлених вимог.

Для роботи з різними СУБД в Java існує технологія JPA, що реалізує ідею об'єктно-реляційного проектування (ORM). Застосування об'єктно-реляційного проектування дозволяє додатку здійснювати роботу не з самою базою даних, а з

об'єктною моделлю структури даних. Синхронізація даної об'єктної моделі і конкретної БД здійснюється за допомогою провайдерів, що реалізують інтерфейс, що надається JPA. Провайдер здатний забезпечити роботу з однієї або декількома СУБД. При виборі СУБД, яку не підтримує використовуваний провайдер, здійснюється зміна провайдера без модифікації вихідного коду ПЗ. Таким чином, в якості технології взаємодії з СУБД видається природним вибрати технологію JPA як технологію, що повністю задовольняє вимогу 1 п. 4.1.1.

На даний момент для розробки графічних інтерфейсів настільних додатків на Java найбільш поширені бібліотеки Swing і SWT. Бібліотека Swing була розроблена компанією Sun Microsystems, а бібліотека SWT компанією IBM. Обидві ці бібліотеки мають розвинену систему графічних компонентів, а значить, задовольняють вимогу 2 в п.6.1.1. Основна відмінність даних бібліотек полягає в тому, що бібліотека Swing промальовує свої компоненти без участі операційної системи, а бібліотека SWT використовує компоненти операційної системи, на якій працює додаток. Через відсутність принципових відмінностей в даних бібліотеках з точки зору сформованих вимог, виберемо бібліотеку SWT через наявність наявного досвіду роботи з її компонентами.

Всі вибрані вище технології безкоштовні і інтегровані в безкоштовні середовища розробки Eclipse і NetBeans. При цьому обидві дані середовища дозволяють розробляти настільні, корпоративні та WEB - додатки, а так само додатки для мобільних пристроїв, що повністю відповідають вимогам 3 і 5 в п.4.1.1. Через відсутність принципових відмінностей з точки зору сформованих вимог обрано середовище Eclipse через наявність досвіду роботи з нею.

4.2 Загальна архітектура і структура даних ІС

Розробляється інформаційна система структурно розділена на модулі, представлені на малюнку 4.1. Такий поділ забезпечує гнучкість і надійність її функціонування. Більшість модулів («Індукція», «Класифікація», «Абдукція», «Управління БД» і модуль «Людино - машинна взаємодія») були розроблені в

процесі створення цільової ІС. Наявність компонентів JPA і JPA-провайдер, що забезпечують об'єктно-реляційне проектування (ORM), є стандартним при застосуванні технології JPA в Java.

Компоненти «Індукція», «Класифікація» і «Абдукції» надають набір функцій, необхідних для реалізації алгоритмів ДСМ - методу, описаних в пункті 3.2. Незалежність модулів, що реалізують набір функцій ДСМ-методу, від модуля, що реалізує логіку ДСМ-методу (тобто від модуля «Людино-машинна взаємодія»), дозволяє гнучко модифікувати і розширювати набір алгоритмів, необхідних для реалізації ДСМ - методу, без зміни логіки самого ДСМ-методу.

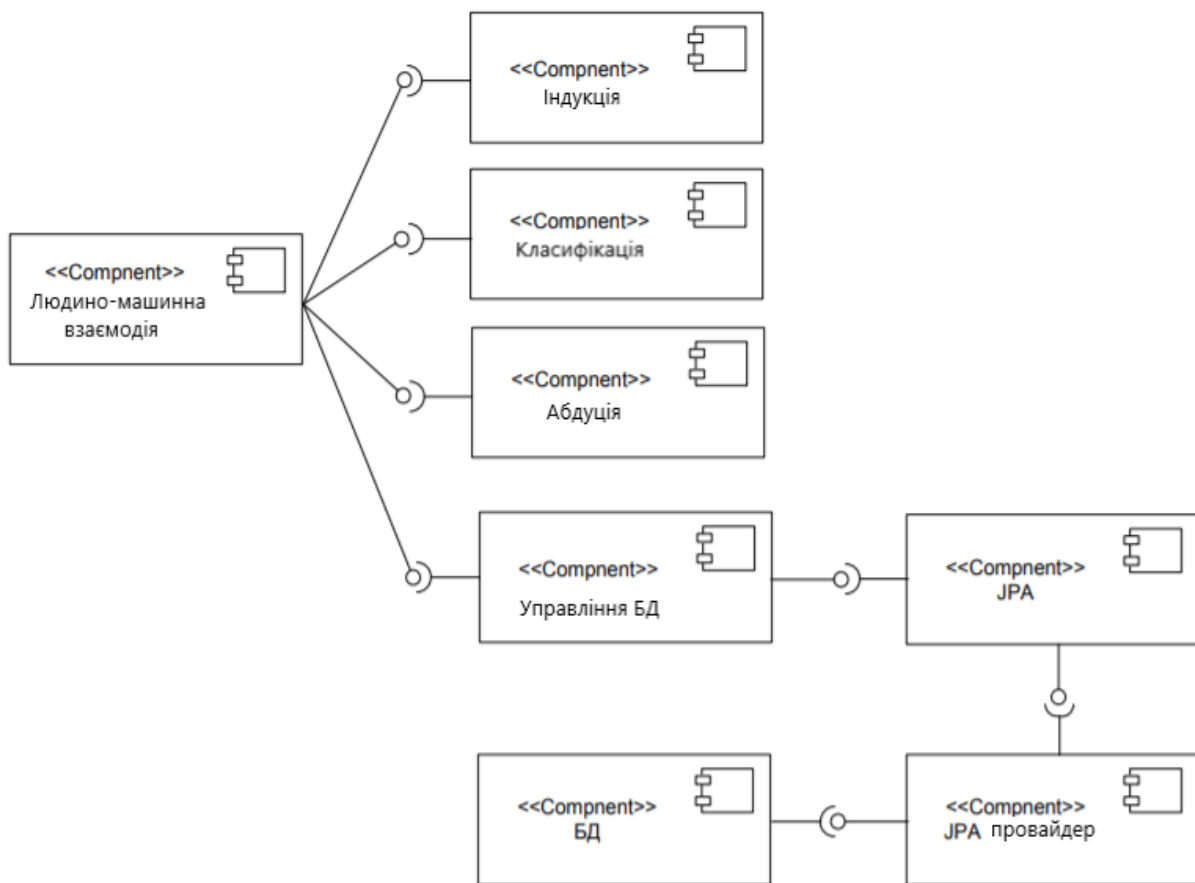


Рисунок 4.1 - Діаграма компонентів в нотації UML 2.0

Модуль «Людино-машинна взаємодія» реалізує графічний інтерфейс користувача, за допомогою якого здійснюється звернення до модулів «Управління

БД», «Індукція», «Класифікація», «Абдукції» та забезпечує користувача наступною функціональністю:

- введення вихідних даних;
- редагування вихідних даних;
- здійснення інтелектуального аналізу вихідних даних за допомогою ДСМ-методу;
- перегляд вихідних даних і результатів інтелектуального аналізу;

Крім того, всі взаємодії між компонентами «Індукція», «Класифікація», «Абдукції» і «Управлінням БД» здійснюється за допомогою компонента «Людино - машинна взаємодія», який забезпечує як передачу результатів інтелектуального аналізу вихідних даних після їх обробки модулями «Індукція», «Класифікація» і «Абдукції» в модуль «Управління БД» для збереження їх в БД, так і передачу вихідних даних з БД в дані модулі для обробки. Всі модулі інформаційної системи використовують єдину об'єктну модель даних, структурна схема якої представлена на рисунку 4.2.

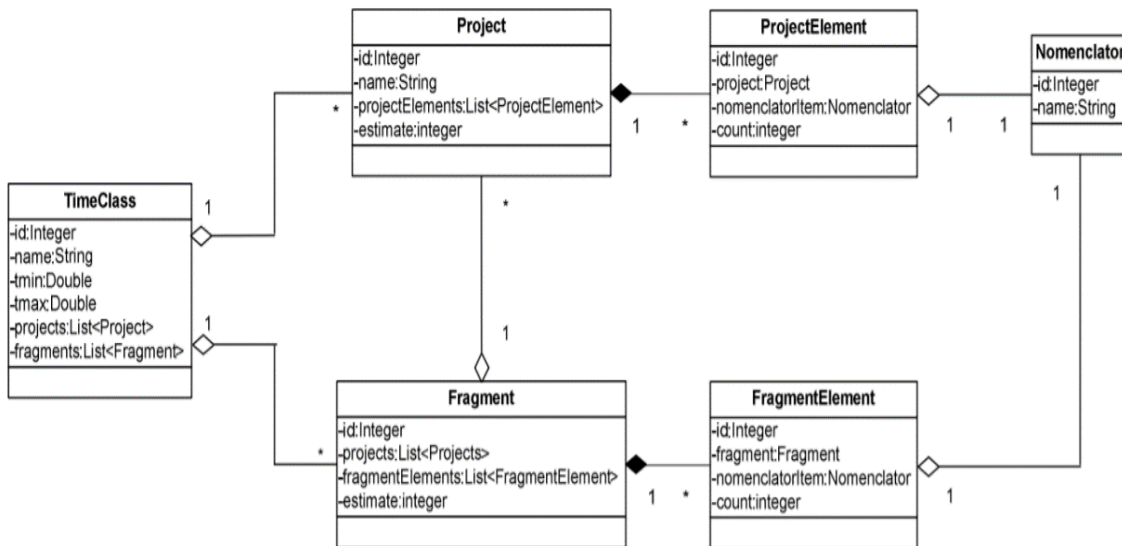


Рисунок 4.2 - Діаграма класів об'єктної моделі ІС в нотації UML 2.0

Дана об'єктна модель даних відображається в СУБД за допомогою інтерфейсу JPA і JPA - провайдера, що реалізують об'єктно-реляційне

проектування (ORM). Модуль «Управління БД» надає API для збереження, редагування і завантаження даних в БД. Відзначимо, що модуль «Управління БД» не залежить від використовуваної СУБД завдяки наявності стандартного інтерфейсу JPA, реалізованого конкретним JPA провайдером, взаємодіє з СУБД.

4.3 Архітектура розроблених компонентів ІС

Компонент «Індукція» (відповідно до малюнком 4.3) здійснює породження всіх необхідних глобальних подібностей і застосування правила першого роду для визначення (+) і (-) - гіпотез кожного класу тривалості. Даний компонент містить інтерфейс `Learner` і клас `LearnerImpl` (відповідно до малюнком 4.3), який представляє собою реалізацію даного інтерфейсу. Метод даного класу `findAllHypotesesInClasses (List <TimeClass> tcList)` здійснює пошук (+) - і (-) - гіпотез кожного класу тривалості за допомогою об'єктів класу `GLAlgorithm`, що реалізує алгоритм пошуку глобальних подібностей, описаний в п.3.2.1. Доступ до функції `findAllHypotesesInClasses (List <TimeClass> tcList)` здійснюється через інтерфейс `Learner`. При такому поділі можна створювати нові реалізації інтерфейсу `Learner`, не змінюючи при цьому зовнішні класи використовують інтерфейс `Learner`. Лістинг класів і інтерфейсів компонента `Learner` наведені в додатку В.1. Лістинг класів, що реалізують алгоритм пошуку глобальних подібностей, наведені в додатку А.

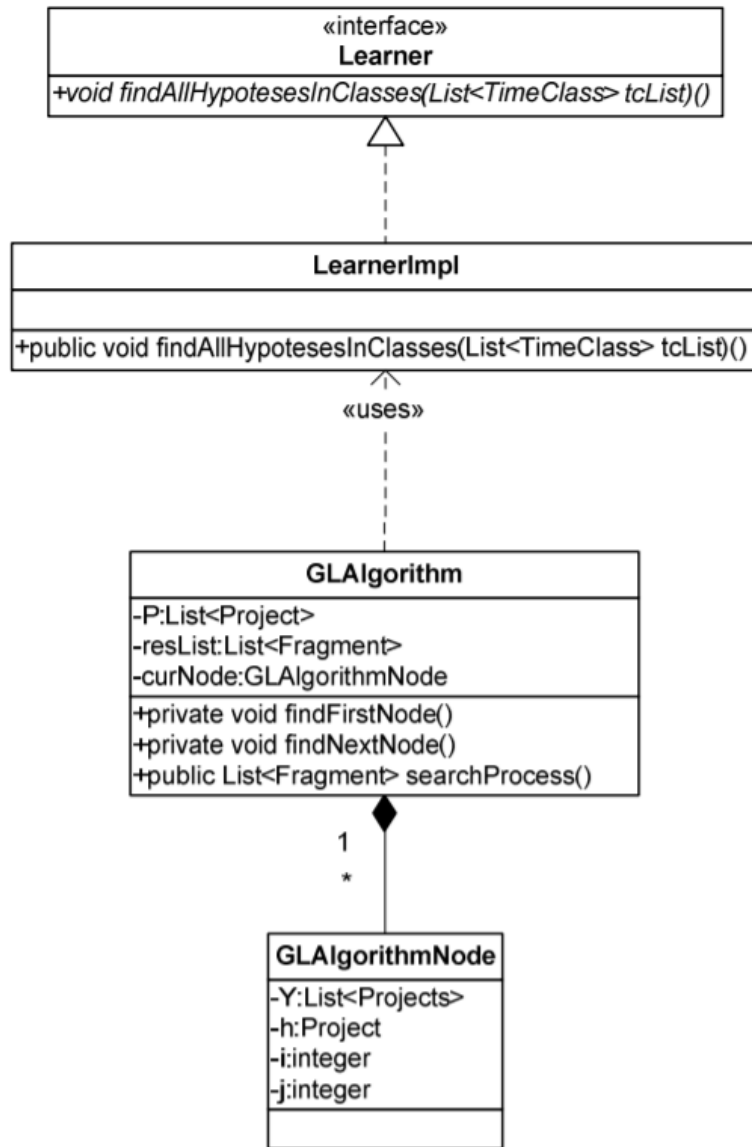


Рисунок 4.3 - Опис реалізації компонента «Індукція»

Компонент «Класифікація» (відповідно до рисунку 4.4) здійснює класифікацію об'єктів за допомогою застосування правила 2-ого роду на основі сформованих гіпотез. Даний компонент містить інтерфейс Solver і клас SolverImpl (відповідно до малюнком 4.4), який представляє собою реалізацію даного інтерфейсу. У класі SolverImpl є метод classifyProject (Project p, List <TimeClass> classes), який реалізує запуск правила 2-ого роду ДСМ-методу. Лістинг класів і інтерфейсів компонента «Класифікація» наведені в Додатку В.2.

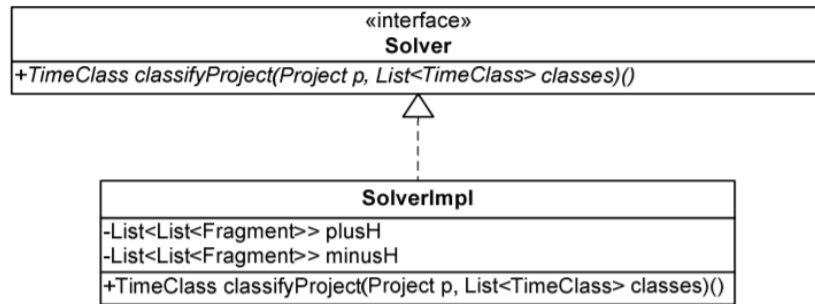


Рисунок 4.4 - Опис реалізації компонента «Класифікація»

Компонент «абдукції» (відповідно до рисунку 4.5) здійснює перевірку виконання умови каузальної повноти. Даний компонент містить інтерфейс **Abduction** і клас **AbductionImpl** (відповідно до рисунку 4.5), який представляє собою реалізацію даного інтерфейсу. Даний компонент використовує компонент «Класифікація» для здійснення класифікації всіх наявних в базі фактів проектів з метою перевірки виконання умови каузальної повноти. Конкретна реалізація інтерфейсу **Solver** компонента «Класифікація» передається компоненту «абдукції» через інтерфейс **Abduction**, що робить незалежним компонент «абдукції» від конкретної реалізації інтерфейсу **Solver**. Лістинг класів і інтерфейсів компонента «абдукції» наведені в Додатку В.3.

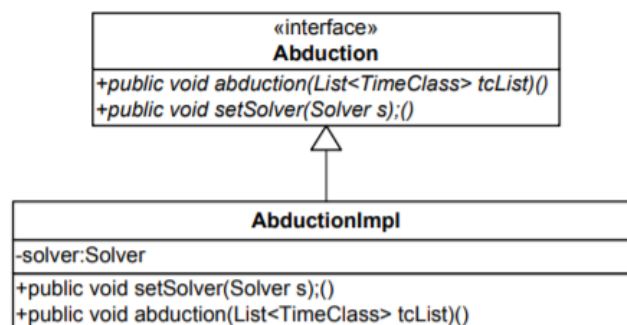


Рисунок 4.5- Опис реалізації компонента «абдукції»

Опис функціонування розробленого програмного засобу представлено в додатку В.5.

4.4. Характеристики функціонування ІС

Найважливішими особливостями функціонування ІС є:

- продуктивність;
- час відгуку;
- стабільність;
- надійність;
- гнучкість.

Виробничий аналіз ІС з даними клітинного поділу характеристики даних.

Найбільш ресурсозатратним з модулів ІС з точки зору обчислювальних витрат є модуль «Індукція», який здійснює пошук глобальних подібностей. Так як обчислювальні витрати пошуку глобальних подібностей залежать від розмірності вхідних даних, час відгуку системи зростає з ростом розмірності вхідних даних. Для того щоб забезпечити швидкий відгук ІС і високу продуктивність, розроблений алгоритм швидкого пошуку глобальних подібностей (див.п. 5.3). Застосування даного алгоритму дозволяє значно знизити крутичність зростання функції, що виражає залежність часу відгуку системи від розмірності вхідних даних, при збільшенні кількості вхідних даних.

Стійкість функціонування ІС забезпечується за допомогою контролю даних, що вводяться користувачем, завдяки чому ІС зберігає працездатність при появі непередбаченого формату вхідних даних. Цілісність даних забезпечується використовуваною СУБД.

Вбудований в Java контроль типів і об'єктно - орієнтований підхід підвищує надійність ІС в цілому. Вбудований контроль типів гарантує контроль переповнення масивів даних. Об'єктно-орієнтований підхід дозволяє добре структурувати код і запобігти більшості помилок.

Гнучкість ІС забезпечена завдяки організації незалежності від СУБД, наявності добре структурованої архітектури ІС і застосування об'єктно-орієнтованого підходу до проектування.

Таким чином, на основі виробленого аналізу можна стверджувати, що розроблена ІС має:

- високу продуктивність і швидкий відгук;
- стійкість функціонування;
- висока надійність;
- гнучкість розширення.

4.5 Апробація програмної системи на задачах спеціальної прикладної предметної області

4.5.1 Підготовка вихідних даних для застосування ДСМ-методу

Емпіричні дані, на основі яких проводиться дане дослідження, являють собою інформацію про проекти виробів (відомості про архітектурні, проектно-конструкторські та інші особливості розроблюваних виробів). При цьому проект виробу відображає тільки склад виробу (до складу «архітектурного» опису виробів входять різні об'єкти в різній кількості). Кожен проект, за яким були зроблені вироби, віднесений до одного з п'яти класів тривалості. Більш повний опис емпіричних даних було представлено в пункті 1.1 «Опис вихідних даних».

Для можливості застосування ДСМ-методу до емпіричним даним необхідне введення таких понять: об'єкт, ознаки об'єкта (що характеризують особливості його структури), властивості об'єкта і подібність об'єктів.

Проект розробки виробу в термінах ДСМ-методу буде характеризувати відповідний об'єкт. Згідно з пунктом 1.1 «Опис вихідних даних», проект виробу являє собою безліч елементів номенклатора O , причому один і той же елемент номенклатора може повторюватися кілька разів в складі об'єкта (таким чином, мова буде йти про так звану нумеровану множину). Кількість кожного з об'єктів номенклатора в складі виробу є значущою характеристикою, тому для обліку кількості елементів в складі проекту визначимо наступну множину $O^*=(O^*=\{o_{i,n}\})$, в якому перший індекс елемента буде відповідати індексу

елемента в множині елементів номенклатора, а другий індекс вказувати на кількість даних елементів (тобто, $i=1, \dots, |O|$, $n \in N$, N - множина натуральних чисел).

Використовуючи введені позначення, можна уявити проект як підмножину безлічі O^* , тобто $p = \{o_{i,n} \mid o_{i,n} \in O^*\} \subseteq O^*$. O^* -множина ознак об'єктів. Таким чином, об'єкт є підмножиною множини ознак об'єктів $X_i \subseteq O^*$.

Клас тривалості проекту в термінах ДСМ-методу буде характеризувати властивості об'єкта. Таким чином, безліч властивостей об'єктів W відповідає безлічі класів проектів C :

$C = \{c_1, c_2, c_3, c_4, c_5\}$, де кожній властивості відповідає впорядкована пара $\{t_{c \min}, t_{c \max}\}$, відповідна інтервалу тривалості $\{t_{c \min}, t_{c \max}\}$, виробництва виробів.

З огляду на природу опрацьованих об'єктів, а саме уявлення про їх (нумеровані) множини, схожість об'єктів буде представлено як теоретико-множинне перетинання описів об'єктів, тобто через перетин множин ознак об'єктів.

4.5.2 Уточнення основних понять і зв'язків, які використовуються в процесі застосування ДСМ-методу, стосовно до розв'язування задачі

У стандартній інтерпретації ДСМ - методи фактів представлені за допомогою частково визначеного відношення «володіння багатьма властивостями», якому відповідає предикат $X \Rightarrow W$. В розв'язуваній задачі в якості об'єкта X розглядається проект розробки виробу, представлене за допомогою деякої безлічі ознак об'єкта, тобто $X \subseteq O^*$, а в якості множини ознак W безліч класів тривалості $W \subseteq C$. При вирішенні даного завдання кожен з проектів може відноситися тільки до одного з розглянутих класів тривалості. Так, відношення \Rightarrow_1^+ «володіння безліччю властивостей» можна в даному випадку перейменувати в відношення «належить класу».

Тоді $J_{<v,n>}(X \Rightarrow W)$ означає, що твердження «об'єкт X належить класу W » на n -му кроці міркування має оцінку v .

Всі об'єкти в вихідній базі фактів розділені на 6 множин, п'ять з яких (P_i | $i \in [1..5]$, $P_i \sqsubseteq P$) відповідають п'яти класам тривалості, а шосте (P_6 , $P_6 \sqsubseteq P$) є безліччю об'єктів, для яких клас не визначений. Множина утворена множиною P_6 розбиттям безлічі P . Таким чином, безліч об'єктів P може бути розкладено на наступні 6 класів розбиття:

1. Множина об'єктів класу 1, для кожного об'єкта X якого виконується предикат $C_1(X) := X \Rightarrow_1^+ 1$;
2. Множина об'єктів класу 2, для кожного об'єкта X якого виконується предикат $C_2(X) := X \Rightarrow_1^+ 2$;
3. Множина об'єктів класу 2, для кожного об'єкта X якого виконується предикат $C_3(X) := X \Rightarrow_1^+ 3$;
4. Множина об'єктів класу 2, для кожного об'єкта X якого виконується предикат $C_4(X) := X \Rightarrow_1^+ 4$;
5. Множина об'єктів класу 2, для кожного об'єкта X якого виконується предикат $C_5(X) := X \Rightarrow_1^+ 5$;
6. Множина об'єктів класу 6, для кожного об'єкта X якого оцінка кожного з наведених вище предикатів дорівнює τ .

Множина об'єктів (P_i | $i \in [1..5]$, $P_i \sqsubseteq P$) являють собою множини (+) - прикладів для кожного з розглянутих властивостей, тобто класів тривалості, відповідно. Множина (P_6 , $P_6 \sqsubseteq P$) є множиною об'єктів, чий клас або властивість необхідно спрогнозувати. Однак, згідно з умовами застосування ДСМ-методу, для кожного з розглянутих властивостей має бути також сформовано безліч (-) - прикладів.

Так як множина всіх об'єктів (за винятком тих об'єктів, чий клас невідомий) розбита на 5 класів тривалості, то видається природним припустити, що множиною (-) - прикладів певного класу тривалості c_k буде об'єднання множин (+) - прикладів інших класів тривалості $\bigcup_{i=1, i \neq k}^5 P_i$, таким чином, для виділення

множин (-) - прикладів для кожного з розглянутих властивостей безліч об'єктів P (за винятком об'єктів безлічі P_6) може бути розбите на наступні 5 підмножин:

1. Множина об'єктів, для кожного об'єкта X якого не виконується предикат $C_1(X) := X \Rightarrow_1^+ 1$ - множина (-) - прикладів 1 властивості.
2. Множина об'єктів, для кожного об'єкта X якого не виконується предикат $C_2(X) := X \Rightarrow_1^+ 2$ - множина (-) - прикладів 2 властивості.
3. Множина об'єктів, для кожного об'єкта X якого не виконується предикат $C_3(X) := X \Rightarrow_1^+ 3$ - множина (-) - прикладів 3 властивості.
4. Множина об'єктів, для кожного об'єкта X якого не виконується предикат $C_4(X) := X \Rightarrow_1^+ 4$ - множина (-) - прикладів 4 властивості.
5. Множина об'єктів, для кожного об'єкта X якого не виконується предикат $C_5(X) := X \Rightarrow_1^+ 5$ - множина (-) - прикладів 5 властивості.

Таким чином, безліч всіх пар <проект, клас проекту> для кожного класу тривалості c_i розбивається в залежності від істинності оцінки, приписується атомарному висловом «об'єкт X належить класу W ($W = c_i$)» відповідному елементу відношення \Rightarrow_1^+ на дві підмножини: W^+ , W^- називаються, відповідно, (+) – прикладами (Об'єкти, що належать класу c_i) (-) - прикладами (об'єкти, які не належать класу c_i). Об'єкти, що залишилися, про які невідомо, якому класу вони належать, будуть (τ) -Теана для кожного класу тривалості.

Одним з результатів роботи ДСМ-методу є безліч гіпотез про причини наявності або відсутності досліджуваних властивостей. У стандартній інтерпретації ДСМ-методу факти представляються за допомогою частково певного ставлення «є причиною» якому відповідає предикат $V \Rightarrow_2^* W$. Це відношення є безліччю пар виду <підоб'єкти, безліч властивостей>. У розв'язуваній задачі в якості V виступає результат виконання операції подібності \cap на об'єктах, тобто підмножина безлічі ознак об'єкта $V \square O^*$. Так як при вирішенні даного завдання властивістю об'єкта є його клас, то предикат $V \Rightarrow_2^* W$ можна інтерпретувати як: «підоб'єкти V є причина належності / неналежності до класу W ». Таким чином, в розв'язуваній задачі відношення \Rightarrow_2^* висловлює

залежність класу проекту від його ознак, тобто від елементів, що входять до його складу.

Отже, підготовлені і проінтерпретовані визначення і відносини, необхідні для застосування ДСМ-методу до задачі прогнозування класу тривалості виробництва виробу за його проектом, показана також адекватність застосування цього методу до такого завдання.

Резюмуючи все вищесказане, ми маємо:

- множина ознак O^* ;
- множина об'єктів $X = \{X_1 \dots X_n\}$, де $X_i \in O^*$, для $1 < i < n$;
- безліч властивостей об'єктів W , відповідних їх класам
 $W = \{c_1, c_2, c_3, c_4, c_5\}$;
- для кожного з класів дві множини W^+, W^- , що називаються, відповідно, (+) - прикладами, (-) - прикладами, а також безліч (τ) - Теана;
- частково певне відношення \Rightarrow_1^+ «належить класу»;
- частково певне відношення \Rightarrow_2^* «є причина належності / неналежності до класу »;

4.5.3 Уточнення основних етапів ДСМ-методу стосовно розв'язуваної прикладної задачі

Послідовність дій, що визначає використовуваний в даній роботі варіант ДСМ-міркування, полягає в наступному:

(1) Застосовується алгоритм, який породжує всі можливі глобальні подібності окремо для (+) - прикладів і (-) - прикладів кожного класу (визначення глобальної подібності див. пункт 3.2.1). Знайдені глобальні подібності для кожного класу будуть проаналізовані на наступному кроці.

(2) Застосовуються правила правдоподібного виведення першого роду, які породжують гіпотези першого роду про причини приналежності (неналежності) до певного класу, тобто породжують ставлення \Rightarrow_2^* Оцінка кожної з породжених

гіпотезів є пара $\langle v, n \rangle$, яка складається з типу істинного значення $v \in \{1, 2, 3, 4, 5, -1\}$ і номера кроку n , при якому виконуються дані правила. Правила формулюються за допомогою двох предикатів M^+ і M^- , виконуючих парами елементів \langle подібний проект, клас довжини \rangle . Данні предикатів затверджують наявність загальної подібності: M^+ для деякої підмножини (+) - прикладів, а M^- - (-) – прикладів. M^+ висловлює наступне твердження:

(a) Для пари $\langle V, W \rangle$ (V – підоб'єкт, W - певний клас тривалості) вірно те, що для даного класу тривалості існує така безліч позитивних прикладів ((+ - об'єктів) $X_1 \dots X_k$, що пара $\langle X_1 \dots X_k, V \rangle$ являєть глобальною схожістю $V \neq \emptyset, k > 2$;

(b) Для всіх $X_1 \dots X_k$ вірно, що істинні твердження $J_{\langle n, l \rangle}(X_1 \Rightarrow_1^+ W), \dots, J_{\langle n, l \rangle}(X_k \Rightarrow_1^+ W), W \neq \emptyset$.

Предикат M^- формулюється симетричним чином (формальне визначення предикатів M^+ і M^- див. в Додатку В.1). Правила правдоподібного виведення першого роду виглядають наступним чином:

Правило першого роду 1: Якщо пара, $\langle V, W_i \rangle$ виконує M^+ , а M^- не виконує, то породжується гіпотеза виду «підоб'єкта V є причина приналежності до класу W_i і оцінка висловлювання, відповідного відношенню \Rightarrow_2^* стає рівною $\langle +1, n + 1 \rangle$; Формально це правило можна представити у вигляді двомісного предиката $M^+(V, W_i) \wedge \neg M^-(V, W_i)$

Правило другого роду 2: Якщо пара, $\langle V, W_i \rangle$ виконує M^+ , і M^- ні одна з пар $\langle V, W_j \rangle$, таких що $j \neq i$ не виконує M^+ то породжується гіпотеза виду «підоб'єкта V є причина приналежності до класу W_i » і оцінка висловлювання, відповідного відношенню \Rightarrow_2^* стає рівною $\langle +2, n + 2 \rangle$; Формально це правило можна представити у вигляді двомісного предиката

$$M^+(V, W_i) \wedge M^-(V, W_i) \wedge \neg \left(\bigvee_{j=1, j \neq i}^N M^+(V, W_j) \right)$$

Правило першого роду 3: Якщо пара $\langle V, W_i \rangle$ виконує M^+ , і M^- і тільки одна з пар $\langle V, W_j \rangle$, таких що $j \neq i$ виконує M^+ , то породжується гіпотеза виду

«підоб'єкта V є причина приналежності до класу W_i » і оцінка висловлювання, відповідного відношенню \Rightarrow_2^* , стає рівною $\langle +3, n + 1 \rangle$; Формально це правило можна представити у вигляді двомісного предиката:

$$M^+(V, W_i) \wedge M^-(V, W_i) \wedge \left(\sum_{j=1, j \neq i}^N M^+(V, W_j) = 1 \right)$$

Правило першого рода 4: Якщо пара $\langle V, W_i \rangle$ виконує M^+ , і M^- і тільки одна з пар $\langle V, W_i \rangle$, таких що $j \neq i$ виконує M^+ , то породжується гіпотеза виду «підоб'єкта V є причина приналежності до класу W_i » і оцінка висловлювання, відповідного відношенню \Rightarrow_2^* , стає рівною $\langle +4, n + 4 \rangle$; Формально це правило можна представити у вигляді двомісного предиката:

$$M^+(V, W_i) \wedge M^-(V, W_i) \wedge \left(\sum_{j=1, j \neq i}^N M^+(V, W_j) > 1 \right)$$

Правило першого роду 5: Якщо пара, $\langle V, W_i \rangle$ не виконує M^+ , а M^- виконує одну пару $\langle V, W_i \rangle$, таких, що $j < i$ виконує M^+ , то породжується гіпотеза виду «підоб'єкта V є причина приналежності до класу W_i » і оцінка висловлювання, відповідного відношенню \Rightarrow_2^* стає рівною $\langle +5, n + 5 \rangle$; Формально це правило можна представити у вигляді двомісного предиката:

$$\neg M^+(V, W_i) \wedge M^-(V, W_i) \wedge \sum_{j=1}^{i-1} M^+(V, W_j) \geq 1$$

Правило шостого роду 6: Якщо пара, $\langle V, W_i \rangle$ не виконує M^+ , а M^- виконує одну пару $\langle V, W_i \rangle$, таких, що $j > i$ виконує M^+ , то породжується гіпотеза виду «підоб'єкта V є причина приналежності до класу W_i » і оцінка висловлювання, відповідного відношенню \Rightarrow_2^* стає рівною $\langle -1, n + 1 \rangle$; Формально це правило можна представити у вигляді двомісного предиката:

$$\neg M^+(V, W_i) \wedge M^-(V, W_i) \wedge \sum_{j=i+1}^N M^+(V, W_j) \geq 1$$

Правило сьомого роду 7: Якщо пара, $\langle V, W_i \rangle$ не виконує M^+ , а M^- виконує одну пару $\langle V, W_i \rangle$, таких, що $j \neq i$ виконує M^+ , то породжується гіпотеза

виду «підоб'єкта V є причина приналежності до класу W_i » і оцінка висловлювання, відповідного відношенню \Rightarrow_2^* стає рівною $\langle \tau, n + 1 \rangle$; Формально це правило можна представити у вигляді двомісного предиката:

$$(\neg M^+(V, W_i) \wedge M^-(V, W_i)) \vee (\neg M^+(V, W_i) \wedge M^-(V, W_i) \wedge \sum_{j=1, j \neq i}^N M^+(V, W_j) = 0)$$

Таким чином, правила першого роду стосовно до розв'язуваної задачі мають такий вигляд (5.1):

$$(I)^{+1} \frac{J_{(\tau, n)}(V \Rightarrow_2 W), M^+(V, W_i) \wedge \neg M^-(V, W_i)}{J_{(1, n+1)}(V \Rightarrow_2 W)}$$

$$(I)^{+2} \frac{J_{(\tau, n)}(V \Rightarrow_2 W), M^+(V, W_i) \wedge M^-(V, W_i) \wedge \neg \bigvee_{j=1, j \neq i}^N M^+(V, W_j)}{J_{(2, n+1)}(V \Rightarrow_2 W)}$$

$$(I)^{+3} \frac{J_{(\tau, n)}(V \Rightarrow_2 W), M^+(V, W_i) \wedge M^-(V, W_i) \wedge (\sum_{j=1, j \neq i}^N M^+(V, W_j) = 1)}{J_{(3, n+1)}(V \Rightarrow_2 W)}$$

$$(I)^{+4} \frac{J_{(\tau, n)}(V \Rightarrow_2 W), M^+(V, W_i) \wedge M^-(V, W_i) \wedge (\sum_{j=1, j \neq i}^N M^+(V, W_j) > 1)}{J_{(4, n+1)}(V \Rightarrow_2 W)}$$

$$(I)^{+5} \frac{J_{(\tau, n)}(V \Rightarrow_2 W), M^+(V, W_i) \wedge M^-(V, W_i) \wedge \sum_{j=1}^{i-1} M^+(V, W_j) \geq 1}{J_{(5, n+1)}(V \Rightarrow_2 W)}$$

$$(I)^{-} \frac{J_{(\tau, n)}(V \Rightarrow_2 W), M^+(V, W_i) \wedge M^-(V, W_i) \wedge \sum_{j=i+1}^N M^+(V, W_j) \geq 1}{J_{(-, n+1)}(V \Rightarrow_2 W)}$$

$$(I)^\tau \frac{J_{(\tau, n)}(V \Rightarrow_2 W), (\neg M^+(V, W_i) \wedge M^-(V, W_i)) \vee (\neg M^+(V, W_i) \wedge M^-(V, W_i) \wedge \sum_{j=1, j \neq i}^N M^+(V, W_j) = 0)}{J_{(\tau, n+1)}(V \Rightarrow_2 W)}$$

(4.5.1) – правила першого роду стосовно розв'язуваної задачі

(3) Застосовуються правила правдоподібного виведення другого роду, які породжують гіпотези другого роду про те, чи належить даний об'єкт до даного класу чи ні, тобто породжують ставлення \Rightarrow_1^* . Оцінка породжених гіпотез є пара $\langle v, n \rangle$, яка складається з істинностного значення $v \in \{1, -1, \tau\}$ і номера кроку n , на якому виконуються дані правила. Дані правила формулюються за допомогою двох двомісних предикатів Π^+ , Π^- виконуваних парами елементів $\langle \text{об'єкт, клас тривалості} \rangle$. Предикат Π^+ , висловлює наступне твердження:

Для пар $\langle X, W \rangle$, остаточно, що об'єкт X містить гіпотези $V_1 \dots V_k$ про позитивні причини належності класу W , породженні правила першого рода, де $V_1 \Rightarrow_2^* W, \dots, V_k \Rightarrow_2^* W$;

Предикат Π^- визначається симметричним образом.

Правила другого рода:

1. Якщо для всіх пар $(X, W_i), i = 1 \dots N$ оцінка $X \Rightarrow_1^* W_i$ рівна $\langle \tau, n \rangle$ і виконується попередник Π^+ , а Π^- не виконуються, то $X \Rightarrow_1^* W_i$ оцінюється як $\langle +1, n + 1 \rangle$ для пар, якому відповідають (+) - гіпотези, що мають ранг вище, (+) - гіпотези, що відповідають останньому з розглянутих пар. Якщо пар з однаковим найбільшим рангом (+) - гіпотез кілька, то вибирається пара з максимальним значенням індексу i .

2. Якщо ні для однієї з пар $(X, W_i), i = 1 \dots N$ маючих оцінку $X \Rightarrow_1^* W_i$ рівну $\langle \tau, n \rangle$ предикату Π^+ вивнюється, а Π^- не виконує, то $X \Rightarrow_1^* W$ оцінюється як $\langle \tau, n \rangle$, тобто оцінка не міняється.

(4) Перевірка виконання умови каузальної повноти. Умова формулюється так: якщо всі об'єкти вихідної бази фактів можуть бути вірно визначені за допомогою правил другого рода, то умова каузальної повноти виконується, інакше - не виконується. Якщо умова каузальної повноти не виконується, то вихідну базу даних (прикладів і контрприкладів для ДСМ-навчання) слід поповнити новими прикладами так, щоб ця умова стала б виконуватися.

У застосуванні до розв'язуваної задачі ДСМ-роздум можна описати таким чином:

1. Аналізуючи все (+) - приклади для кожного класу тривалості, отримуємо за допомогою алгоритму пошуку глобальних подібностей для кожного класу тривалості набори ознак, які є загальними для декількох об'єктів числом не менше двох. З (-) - прикладами для кожного класу тривалості робимо аналогічним чином.
2. Оголошуємо для кожного класу тривалості набір ознак об'єкта, отриманий на попередньому кроці, (+) -, (-)-, (τ) - гіпотезою відповідно

- до правил першого роду. Породжені гіпотези є причинами приналежності (неналежності) об'єкта до певного класу тривалості.
3. Переглядаємо всі (τ) -Теани. Класифікуємо (τ) -приклад з допомогою правил другого роду. Якщо класифікація була виконана за допомогою 2-ого правила другого роду, то об'єкт залишається некласифікованим.
 4. Здійснюємо перевірку каузальної повноти. Припустимо, що ні для якого об'єкта невідомо значення його класу тривалості (тобто всі об'єкти стали (τ) об'єктами), і здійснимо класифікацію даних об'єктів за допомогою правила 2-ого роду. Якщо після цього хоча б один з (τ) -об'єктів буде некласифікованим або буде класифікований невірно, то критерій каузальної повноти не виконано. Це означає, що потрібно збільшити або змінити вибірку, так як занадто мало показових (навчальних) прикладів, на яких алгоритм може навчитися породжувати причини приналежності об'єкта до певного класу.

Крок (4) формує абдуктивне пояснення [5] початкового стану вихідних даних. На цьому кроці або завершується ДСМ-роздум (якщо пояснені всі (τ) -Теани і вірно пояснені всі $(+)$ - приклади кожного класу), або в іншому випадку в інтерактивному режимі до всіх непояснених або невірно пояснених $(+)$ - прикладів кожного класу тривалості додаються додаткові $(+)$ - приклади, і ДСМ-роздум ітерується щодо нового початкового стану бази даних.

Аналіз різних способів визначення вирішальних предикатів і правдоподібного виведення і вибір найкращого варіанту, що лежить в основі даного формального опису представлений в додатку В.6. Крім того, в цьому додатку також наводиться змістовна інтерпретація отриманих гіпотез.

4.5.4 Змістовна інтерпретація результатів класифікації

На малюнку 5.1 представлені результати класифікації навчальної вибірки і нових проектів за допомогою правил правдоподібного виведення другого роду на основі бази отриманих гіпотез, описаних в Додатку В 6.

ІС прогнозування тривалості виробництва					
Дані методу ДСМ					
Проекти в класах					
Клас 1	Клас 2	Клас 3	Клас 4	Клас 5	Не класифіковані
1.1 ШТК 102ВН-03С: (1,2,3): 1	2.1 ШТК 104ВН-01С: (2,3,13): 2	3.1 ШТК 102ВН-04С: (1,2,3,20): 3	4.1 ШТК 102ВН-06С: (1,2,3,33,34): 5	5.1 ШТК 103ВН-03С: (2,3,22,33,34): 5	ШТК 100ВН-22А: (3,5,7) клас: 1
1.2 ШТК 100ВН-01С: (3,4,5): 1	2.2 ШТК 104ВН-02С: (2,3,14): 2	3.2 ШТК 102ТН-02С: (1,2,6,21): 3	4.2 ШТК 102ТН-03С: (2,6,33,34,40): 5	5.2 ШТК 103ВН-02С: (5,33,34,29): 5	ШТК 100ТН-21С: (4,6,8) клас: 1
1.3 ШТК 102ТН-01С: (1,2,6): 1	2.4 ШТК 100: (1,2,6,16): 2	3.3 ШТК 103ВН-01С: (2,3,22): 3	4.3 ШТК 103ВН-02С: (2,3,22,24): 4	5.3 ШТК 104ТН-01С: (2,3,34,32,30,35): 5	ШТК 102ТН-24С: (1,6,8,19) клас: 2
1.5 ШТК 100ВН-02А: (7,8,9): 1	2.7 ШТК 104ВН-03С: (2,18,19): 2	3.4 ШТК 103ТН-01С: (2,6,23): 3	4.4 ШТК 103ВН-01С: (2,3,24,29): 4	5.4 ШТК 103ВН-01С: (33,34,29,36): 5	ШТК 104ВН-23С: (8,16,18) клас: 2
1.6 ШТК 100ВН-03С: (9,11,10): 1	2.8 ШТК 100ВН-05С: (3,10,19,28): 2	3.5 ШТК 102ВН-05С: (2,7,9,24): 4	4.5 ШТК 104ВН-05С: (2,3,20,25): 4	5.5 ШТК 103ТН-02С: (94,32,35,37,39,38): 5	ШТК 104ВН-26С: (2,9,15,17,25) клас: 3
1.7 ШТК 100ВН-04С: (3,5,10): 1	2.9 ШТК 104ВН-11С: (5,9,13): 2	3.7 ШТК 104ВН-05С: (2,3,16,17,25): 3	4.6 ШТК 104ВН-06С: (2,9,20,25): 4	5.7 ШТК 104ВН-07С: (2,3,13,33,34,36): 5	ШТК 102ВН-24С: (1,6,9,20) клас: 3
1.8 ШТК 102ВН-10С: (1,5,9): 1	2.10 ШТК 100-1: (4,5,6,16): 2	3.8 ШТК 104ВН-06С: (5,9,17,18,19): 3	4.7 ШТК 102ВН-07С: (2,3,14,24,31): 4	5.8 ШТК 102ТН-05С: (2,19,33,34,40,32): 5	ШТК 103ТН-22С: (2,21,34,29,32,35) клас: 5
1.9 ШТК 102ВН-11С: (2,3,4): 1	2.13 ШТК 104ВН-12С: (5,9,14): 2	3.9 ШТК 104ВН-04С: (2,25,26,27): 3	4.10 ШТК 103ВН-12С: (3,5,22,24): 4	5.9 ШТК 103ВН-11С: (2,33,34,29): 5	ШТК 103ВН-11С: (2,33,34,29): 5
1.10 ШТК 100ВН-12С: (1,3,5): 1	2.14 ШТК 104ВН-13С: (5,9,14): 2	3.10 ШТК 103ВН-11С: (5,9,23): 3	4.12 ШТК 102ВН-14С: (5,9,14,24,31): 4	5.10 ШТК 102ТН-12С: (5,7,19,33,34,32): 5	ШТК 104ВН-25С: (20,30,36,37) клас: 3
1.4 ШТК 100ТН-01С: (4,5,6): 1	2.15 ШТК 104ВН-14С: (5,18,19): 2	3.11 ШТК 102ВН-12С: (4,5,9,20): 3	4.13 ШТК 102ТН-15С: (1,2,3,34,38): 4	5.11 ШТК 104ТН-13С: (3,5,18,32,35,38): 5	ШТК 104ВН-21С: (23,24,37) клас: 4
1.11 ШТК 100ВН-13С: (2,3,7): 1	2.16 ШТК 100ВН-15С: (2,7,9,19): 2	3.12 ШТК 104ВН-13С: (5,9,16,17,18): 3	4.14 ШТК 102ТН-16С: (5,7,9,34,38): 4	5.12 ШТК 104ВН-14С: (5,9,14,33,34,36): 5	ШТК 104ВН-21С: (14,15,17,33) клас: 3
1.12 ШТК 102ВН-14С: (1,2,9): 1	2.17 ШТК 104ВН-16С: (2,3,18): 2	3.13 ШТК 104ВН-14С: (2,9,17,18,19): 3	4.15 ШТК 102ВН-17С: (3,5,24,29): 4	5.13 ШТК 103ВН-15С: (5,9,22,33,34): 5	
1.13 ШТК 102ТН-15С: (2,4,6): 1	2.18 ШТК 104ВН-17С: (2,3,25): 3	3.14 ШТК 102ВН-15С: (2,4,6,24): 4			
1.14 ШТК 102ВН-16С: (4,5,9): 1		3.16 ШТК 102ТН-16С: (6,11,10,21): 3			
1.15 ШТК 100ТН-17С: (1,5,6): 1		3.15 ШТК 103ВН-17С: (5,9,22): 3			
1.16 ШТК 102ВН-18С: (2,4,9): 1					
1.17 ШТК 102ВН-19С: (5,7,9): 1					

Рисунок 5.1 - Результати класифікації навчальної вибірки і нових проектів

Таким чином, з 69 проектів, що становлять навчальну вибірку, вірно було класифіковано 64 проекти. Решта 5 проектів класифіковані невірно, не класифікованих проектів немає.

Той факт, що більша частина проектів навчальної вибірки була класифікована вірно, говорить про те, що була вірно виявлена (реконструйована за вихідної навчальної вибірки проектів) більша частина справжніх причин (гіпотез) належності / неналежності проекту певного класу тривалості.

У таблиці 5.1 наведені дані, що відображають використання (+) - гіпотез певного рангу при класифікації.

Таблиця 5.1 - Статистика використання гіпотез

Ранг(+)-гіпотези	Кількість класифікованих проектів	
	Правильно	Неправильно
1	51	5
2	13	0
3	0	0
4	0	0
5	0	0

Таким чином, можна зробити висновок, що всі правильно класифіковані проекти класифікувалися за допомогою (+) - гіпотез рангу 1 або 2. Даний факт підтверджує визначальну роль даних гіпотез при класифікації і, як наслідок, обґрунтованість розгляду даних гіпотез як гіпотез найбільшої сили (див. Додаток В.6)

Розглянемо

змістовну інтерпретацію результатів класифікації проектів навчальної вибірки на кількох прикладах:

Класифікації проекту 1.9 {2,3,4}.

Даний проект класифікований вірно в клас тривалості 1 (в відповідності до рисунку 5.2). Гіпотезами, на основі яких була проведена класифікація, є гіпотези {2,4} і {3,4}. В склад проекту входять модуль нагрівача, модуль вентилятора і корпус першої складності. У складі класифікуючих гіпотез відображені комбінації модуля нагрівача і корпусу першої складності і модуля вентилятора і корпусу першої складності. Комбінація елементів {2,3,4} не зустрічається серед (+) - гіпотез 1-2 рангу в старших класах, отже, немає достатньої кількості фактів, які говорять про те, що дана комбінація здатна вплинути на збільшення термінів тривалості перевищують тривалість виробництва виробів 1-го класу, тобто проект обґрунтовано класифікований в 1-й клас.

Класифікація проекту		Клас
Шкаф (1.9 ШТК102ВН-11С: {2,3,4})		
Підходящі гіпотези		
(+1):(3,4): { 1.2, 1.9 }		1
(+1):(2,4): { 1.9, 1.1, 1.1 }		1
(+3):(4): { 1.2, 1.9, 1.4, 1.1, 1.1, 1.1 }		1
(+4):(3): { 1.1, 1.2, 1.7, 1.9, 1.1, 1.1 }		1
(+4):(2): { 1.1, 1.3, 1.9, 1.1, 1.1, 1.1, 1.1 }		1
(+4):(2,3): { 1.1, 1.9, 1.1 }		1
(+4):(2,3): { 2.1, 2.2, 2.1, 2.1 }		2
(+4):(2): { 2.1, 2.2, 2.4, 2.7, 2.1, 2.1, 2.1 }		2
(+4):(3): { 2.1, 2.2, 2.8, 2.1, 2.1 }		2
(+5):(3,4): { 1.2, 1.9 }		2
(+5):(4): { 1.2, 1.9, 1.4, 1.1, 1.1, 1.1, 3.1, 3.1 }		2
(+5):(2,4): { 1.9, 1.1, 1.1, 3.1 }		2
(+3):(4): { 3.1, 3.1 }		3
(+4):(2): { 3.1, 3.2, 3.3, 3.4, 3.5, 3.7, 3.9, 3.1, 3.1 }		3
(+4):(2,3): { 3.1, 3.3, 3.7 }		3
(+5):(3): { 2.1, 2.2, 2.8, 2.1, 2.1, 1.1, 1.2, 1.7, 1.9, 1.1, 1.1, 4.1, 4.3, 4.4, 4.5, 4.7, 4.1, 4.1, 4.1, 5.1, 5.7, 5.1 }		3
(+5):(3,4): { 1.2, 1.9 }		3
(+5):(2,4): { 1.9, 1.1, 1.1 }		3
(+4):(2): { 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.1 }		4
(+4):(2,3): { 4.1, 4.3, 4.4, 4.5, 4.7, 4.1 }		4
(+4):(3): { 4.1, 4.3, 4.4, 4.5, 4.7, 4.1, 4.1, 4.1 }		4
(+5):(4): { 3.1, 3.1, 2.1, 1.2, 1.9, 1.4, 1.1, 1.1, 1.1 }		4
(+5):(2,4): { 3.1, 1.9, 1.1, 1.1 }		4
(+5):(3,4): { 1.2, 1.9 }		4
(+4):(3): { 5.1, 5.7, 5.1 }		5
(+5):(2): { 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.1, 3.1, 3.2, 3.3, 3.4, 3.5, 3.7, 3.9, 3.1, 3.1, 2.1, 2.2, 2.4, 2.7, 2.1, 2.1, 2.1, 1.1, 1.3, 1.9, 1.1, 1.1, 1.1, 1.1 }		5
(+5):(2,3): { 4.1, 4.3, 4.4, 4.5, 4.7, 4.1, 3.1, 3.3, 3.7, 2.1, 2.2, 2.1, 2.1, 1.1, 1.9, 1.1 }		5
(+5):(4): { 3.1, 3.1, 2.1, 1.2, 1.9, 1.4, 1.1, 1.1, 1.1 }		5
(+5):(2,4): { 3.1, 1.9, 1.1, 1.1 }		5
(+5):(3,4): { 1.2, 1.9 }		5
Класифікуючі гіпотези		
(+1):(3,4): { 1.2, 1.9 }	Клас	1
(+1):(2,4): { 1.9, 1.1, 1.1 }	Клас	1
Клас проекту : 1		

Рисунок 5.2 - Результати класифікації проекту 1.9 {2,3,4}

Класифікація проекту 2.2 {2,3,14}

Даний проект класифікований вірно в клас тривалості 2 (з відповідністю до рисунка 5.3). Гіпотезою, на основі якої була проведена класифікація, є гіпотеза {14}. До складу проекту входять модуль нагрівача, модуль вентилятора і модуль корпусу другої складності. До складу класифікованої гіпотези входить модуль корпусу другої складності. Корпус другої складності не зустрічався в проектах першого класу. Крім того, в проектах старших класів зустрічається або корпус третьої складності, або комбінація корпусу другої складності з іншими елементами, що не зустрічаються в другому класі. даний факт підтверджує, що гіпотеза {14} є причиною збільшення терміну тривалості виробництва до 2-ого класу тривалості, отже, проект обґрунтовано класифікований у 2-й клас.

Класифікація проекту	
Шкаф (2.2 ШТК104ВН-02С: {2,3,14})	
Підходящі гіпотези	Клас
{++}{3}: { 1.1, 1.2, 1.7, 1.9, 1.1, 1.1 }	1
{++}{2}: { 1.1, 1.3, 1.9, 1.1, 1.1, 1.1, 1.1 }	1
{++}{2,3}: { 1.1, 1.9, 1.1 }	1
{-}{14}: { 2.2, 2.1, 2.1, 4.7, 4.1, 5.1 }	1
{+2}{14}: { 2.2, 2.1, 2.1 }	2
{++}{2,3}: { 2.1, 2.2, 2.1, 2.1 }	2
{++}{2}: { 2.1, 2.2, 2.4, 2.7, 2.1, 2.1, 2.1 }	2
{++}{3}: { 2.1, 2.2, 2.8, 2.1, 2.1 }	2
{++}{2}: { 3.1, 3.2, 3.3, 3.4, 3.5, 3.7, 3.9, 3.1, 3.1 }	3
{++}{2,3}: { 3.1, 3.3, 3.7 }	3
{+5}{3}: { 2.1, 2.2, 2.8, 2.1, 2.1, 1.1, 1.2, 1.7, 1.9, 1.1, 1.1, 4.1, 4.3, 4.4, 4.5, 4.7, 4.1, 4.1, 4.1, 5.1, 5.7, 5.1 }	3
{+5}{14}: { 2.2, 2.1, 2.1, 4.7, 4.1, 5.1 }	3
{++}{2}: { 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.1 }	4
{++}{2,3}: { 4.1, 4.3, 4.4, 4.5, 4.7, 4.1 }	4
{++}{3}: { 4.1, 4.3, 4.4, 4.5, 4.7, 4.1, 4.1, 4.1 }	4
{+5}{14}: { 2.2, 2.1, 2.1, 5.1 }	4
{+4}{3}: { 5.1, 5.7, 5.1 }	5
{+5}{2}: { 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.1, 3.1, 3.2, 3.3, 3.4, 3.5, 3.7, 3.9, 3.1, 3.1, 2.1, 2.2, 2.4, 2.7, 2.1, 2.1, 2.1, 1.1, 1.3, 1.9, 1.1, 1.1, 1.1, 1.1 }	5
{+5}{2,3}: { 4.1, 4.3, 4.4, 4.5, 4.7, 4.1, 3.1, 3.3, 3.7, 2.1, 2.2, 2.1, 2.1, 1.1, 1.9, 1.1 }	5
{+5}{14}: { 4.7, 4.1, 2.2, 2.1, 2.1 }	5
Класифікація гіпотез	Клас
{+2}{14}: { 2.2, 2.1, 2.1 }	2
Клас проекту : 2	

Рисунок 5.3 - Результати класифікації проекту 2.2 {2,3, 4}

Проаналізуємо причини невірно класифікованих проектів навчальної вибірки на прикладі проекту 2. 8 {2,3,25}.

Даний проект класифікований невірно в клас тривалості 3 (в відповідності до рисунку 5.4). Справжнім класом даного проекту є 2-ий клас тривалості. Гіпотеза, на основі якої була проведена класифікація, є гіпотеза {2,25}. До складу проекту входять модуль нагрівача, модуль вентилятора і модуль корпусу другої складності. В склад класифікуючої гіпотези входить модуль корпусу другої

складності і модуль нагрівача. Класифікуюча гіпотеза {2,25} по причині відсутності фактів помилково є гіпотезою 1-ого рангу для 3-го класу тривалості (див. Додаток В.6). Дана гіпотеза отримана на підставі подібності проектів 3.7 {2,3, 6, 7,25} і 3.9 {2,25,26,27}, проте реальними причинами потрапляння проектів в третій клас, є комбінації складних виробів {6, 7,25} і {25,26,27}, при цьому приклади, що дозволяють виявити дані достовірні гіпотези відсутні в 3-му класі тривалості. З іншого боку 2-й клас тривалості теж не містить достатнього кількості (+) - прикладів для виявлення гіпотези {2,25} через брак фактів.

Підходящі гіпотези	Клас
{(+4):{3} : { 1.1, 1.2, 1.7, 1.9, 1.1, 1.1}}	1
{(+4):{2} : { 1.1, 1.3, 1.9, 1.1, 1.1, 1.1, 1.1}}	1
{(+4):{2,3} : { 1.1, 1.9, 1.1}}	1
{(-):{2,25} : { 2.1, 3.7, 3.9, 4.5, 4.6}}	1
{(+4):{2,3} : { 2.1, 2.2, 2.1, 2.1}}	2
{(+4):{2} : { 2.1, 2.2, 2.4, 2.7, 2.1, 2.1, 2.1}}	2
{(+4):{3} : { 2.1, 2.2, 2.8, 2.1, 2.1}}	2
{(-):{2,25} : { 3.7, 3.9, 4.5, 4.6}}	2
{(+2):{2,25} : { 3.7, 3.9}}	3
{(+4):{2} : { 3.1, 3.2, 3.3, 3.4, 3.5, 3.7, 3.9, 3.1, 3.1}}	3
{(+4):{2,3} : { 3.1, 3.3, 3.7}}	3
{(+5):{3} : { 2.1, 2.2, 2.8, 2.1, 2.1, 1.1, 1.2, 1.7, 1.9, 1.1, 1.1, 4.1, 4.3, 4.4, 4.5, 4.7, 4.1, 4.1, 4.1, 5.1, 5.7, 5.1}}	3
{(+4):{2} : { 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.1}}	4
{(+4):{2,3} : { 4.1, 4.3, 4.4, 4.5, 4.7, 4.1}}	4
{(+4):{3} : { 4.1, 4.3, 4.4, 4.5, 4.7, 4.1, 4.1, 4.1}}	4
{(+5):{2,25} : { 3.7, 3.9, 2.1}}	4
{(+4):{3} : { 5.1, 5.7, 5.1}}	5
{(+5):{2} : { 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.1, 3.1, 3.2, 3.3, 3.4, 3.5, 3.7, 3.9, 3.1, 3.1, 2.1, 2.2, 2.4, 2.7, 2.1, 2.1, 2.1, 1.1, 1.3, 1.9, 1.1, 1.1, 1.1, 1.1}}	5
{(+5):{2,3} : { 4.1, 4.3, 4.4, 4.5, 4.7, 4.1, 3.1, 3.3, 3.7, 2.1, 2.2, 2.1, 2.1, 1.1, 1.9, 1.1}}	5
{(+5):{2,25} : { 4.5, 4.6, 3.7, 3.9, 2.1}}	5

Класифікуючі гіпотези	Клас
{(+2):{2,25} : { 3.7, 3.9}}	3

Клас проекту : 3

Рисунок 5.4 - Результати класифікації проекту 2.18 {2,3,25}

Проаналізуємо також результати класифікації нових проектів

На основі аналізу результатів класифікації нових проектів можна зробити висновок, що всі нові проекти були класифіковані. Однак, розроблена ІС також може класифікувати невірно або НЕ виконати класифікацію нових проектів в тих випадках, коли в структурі нових виробів містяться нові або унікальні елементи, що не використовувалися до цього при створенні об'єктів. Даний факт не говорить про помилки в моделі класифікації, а лише відображає необхідність своєчасного поповнення бази фактів новими прикладами.

Таким чином, аналіз результатів класифікації показав, що результати класифікації об'єктів за допомогою розробленої ІС навіть на невеликих

навчальних вибірках переважно є вірними, при цій гіпотезі, на основі якої відбувається класифікація є змістовно інтерпретованими і, як наслідок, інформативними і обґрунтованими.

4.6 ДСМ - метод як платформа ІАД

4.6.1 Логічні та алгоритмічні засоби ДСМ-методу як інструментарій формалізації класу евристик для використання в комп'ютерних системах ІАД

Інтелектуальний аналіз даних - це комп'ютерні технології, що базуються на математичному (модель і алгоритми) і комп'ютерному (Інструментальні системи ПО для вирішення важких завдань) моделюванні нетривіальних пізнавальних функцій дослідника (в тому числі його здатності вести коректні міркування) при вирішенні задач аналізу емпіричних даних і вилучення з цих даних (взагалі кажучи - прихованих) закономірностей (асоціативних, причинних і т.п залежностей). Особливу роль в цьому процесі відіграє використання комп'ютерних систем - «підсилювачів» інтелектуальних можливостей дослідника при його роботі з реальними даними в практичних додатках ІАД. Основними завданнями, які розв'язуються в рамках досліджень і розробок, що проводяться в цій галузі є:

- Вибір неформальної семантики (нетривіальних «фігур міркування») для використання в інструментальних ПО-системах обговорюваного типу.
- Коректна математична формалізація таких «фігур міркування» для використання в інструментальних ПО-системах обговорюваного типу.
- Розробка ефективних алгоритмічних моделей, що дозволяють реалізувати запропоновані математичні моделі в інструментальних ПО-системах обговорюваного типу.
- Розробка засобів «налаштування» інструментальних систем обговорюваного типу на специфічні особливості предметних областей їх застосування.

- Експериментальна апробація і доопрацювання математичних моделей а також алгоритміки пропонованого інструментарію комп'ютерного моделювання міркувань з урахуванням конкретних особливостей аналізованої предметної області.

В рамках даної роботи, крім вибору неформальній семантики (див п. 3.1), побудови математичної формалізації "фігур міркування", лежать в основі обраної неформальній семантики, розробки ряду алгоритмічних моделей (див. п. 3) були проведені:

- налаштування стандартних ДСМ-засобів на особливості обраної предметної області. (В рамках проведення такого налаштування була визначена формалізація подання емпіричних даних, формалізація поняття подібності, модифікація логічних і алгоритмічних засобів ДСМ-методу);
- досвідчена перевірка застосовності розробленого математичного та програмного інструментарію, в рамках якої було проведено ряд експериментів, де метою було визначення та формалізація (в тому числі - у вигляді налаштувань відповідного ПО) найбільш істотних змістовних особливостей (характеристик) предметної області (в тому числі -способи вибору прикладів і контрприкладів, гіпотез і правил виведення);

В результаті виконаних налаштувань стандартних ДСМ методів на особливості предметної області і проведення ряду експериментів вдалося розробити проблемно-орієнтовану ІС, результати застосування якої в обраній прикладної предметної області (і перш за все - отримані гіпотези і результати класифікації) є обґрунтованими і легко інтерпретуються.

Таким чином, на підставі проведених експериментів можна аргументовано стверджувати, що ДСМ-метод (як система математичних моделей і алгоритмічних засобів) є зручною платформою як для формалізації цікавого класу евристик («канони» індуктивного виводу в стилі Д.С. Мілль, абдуктивні висновки в стилі Ч. С. Пірса і міркування за аналогією в стилі Д.Пойа - див п. 3.1), так і для їх практичного застосування в прикладних проблемно-орієнтованих системах ІАД.

4.6.2 Демонстрація можливостей поліпшення якості ДСМ прогноза на основі спрямованого розширення вихідних даних

Продемонструємо можливості поліпшення якості ДСМ-прогнозу, а саме - поліпшення інтерпретується гіпотез і результатів класифікації (див п. 5.4 та Додаток В.6) в розглянутій тут прикладній задачі, на основі спрямованого розширення бази фактів, яка може відбуватися природним чином в процесі накопичення досвіду виробництва виробів з проектів.

У Додатку В.6 був представлений приклад помилкової класифікації, а саме: приклад класифікації проекту 2.18 {2,3,25}. Відзначимо ще раз, що причиною некоректної класифікації даного проекту, був той факт, що 2-й клас, до якого належить проект 2.18, не містить відповідних (+) - гіпотез для вірної класифікації проекту, а 3-й клас містить помилкову (+) - гіпотезу 1-ого рангу, за допомогою якої і була проведена класифікація (відповідно до рисунку 6.1).

Клас 1	Клас 2	Клас 3	Клас 4	Клас 5
1.1 ШТК 102ВН-03С: {1,2,3}: 1	2.1 ШТК 104ВН-01С: {2,3,13}: 2	3.1 ШТК 102ВН-04С: {1,2,3,20}: 3	4.1 ШТК 102ВН-06С: {1,2,3,33,34}: 5	5.1 ШТК 103ВН-03С: {2,3,22,33,34}: 5
1.2 ШТК 100ВН-01С: {3,4,5}: 1	2.2 ШТК 104ВН-02С: {2,3,14}: 2	3.2 ШТК 102ТН-02С: {1,2,6,21}: 3	4.2 ШТК 102ТН-03С: {2,6,33,34,40}: 5	5.2 ШТК 103ВН-02С: {5,33,34,29}: 5
1.3 ШТК 102ТН-01С: {1,2,6}: 1	2.4 ШТК 100: {1,2,6,16}: 2	3.3 ШТК 103ВН-01С: {2,3,22}: 3	4.3 ШТК 103ВН-02С: {2,3,22,24}: 4	5.3 ШТК 104ТТН-01С: {2,33,34,32,30,35}: 5
1.5 ШТК 100ВН-02А: {7,8,9}: 1	2.7 ШТК 104КН-03С: {2,18,19}: 2	3.4 ШТК 103ТН-01С: {2,6,23}: 3	4.4 ШТК 103КН-01С: {2,3,24,29}: 4	5.4 ШТК 103К-01С: {33,34,29,36}: 5
1.6 ШТК 100ВН-03С: {9,11,10}: 1	2.8 ШТК 100ВН-05С: {3,10,19,28}: 2	3.5 ШТК 102ВН-05С: {2,7,9,24}: 4	4.5 ШТК 104КН-05С: {2,3,20,25}: 4	5.5 ШТК 103ТТН-02С: {34,32,35,37,39,38}: 5
1.7 ШТК 100ВН-04С: {3,5,10}: 1	2.9 ШТК 104ВН-11С: {5,9,13}: 2	3.7 ШТК 104ВН-05С: {2,3,16,17,25}: 3	4.6 ШТК 104КН-06С: {2,9,20,25}: 4	5.7 ШТК 104ВН-07С: {2,3,13,33,34,36}: 5
1.8 ШТК 102ВН-10С: {1,5,9}: 1	2.10 ШТК 100-1: {4,5,6,16}: 2	3.8 ШТК 104ВН-06С: {5,9,17,18,19}: 3	4.7 ШТК 102ВН-07С: {2,3,14,24,31}: 4	5.8 ШТК 102ТН-05С: {2,19,33,34,40,32}: 5
1.9 ШТК 102ВН-11С: {2,3,4}: 1	2.13 ШТК 104ВН-12С: {5,9,14}: 2	3.9 ШТК 104КН-04С: {2,25,26,27}: 3	4.10 ШТК 103ВН-12С: {3,5,22,24}: 4	5.9 ШТК 103КН-11С: {2,33,34,29}: 5
1.10 ШТК 100ВН-12С: {1,3,5}: 1	2.14 ШТК 104ВН-13С: {5,6,14}: 2	3.10 ШТК 103ВН-11С: {5,9,23}: 3	4.12 ШТК 102ВН-14С: {5,9,14,24,31}: 4	5.10 ШТК 102ТН-12С: {5,7,19,33,34,32}: 5
1.4 ШТК 100ТН-01С: {4,5,6}: 1	2.15 ШТК 104КН-14С: {5,18,19}: 2	3.11 ШТК 102ВН-12С: {4,5,9,20}: 3	4.13 ШТК 102ТН-15С: {1,2,3,34,38}: 4	5.11 ШТК 104ТТН-13С: {3,5,18,32,35,38}: 5
1.11 ШТК 100ВН-13С: {2,3,7}: 1	2.16 ШТК 100ВН-15С: {2,7,9,19}: 2	3.12 ШТК 104ВН-13С: {5,9,16,17,18}: 3	4.14 ШТК 102ТН-16С: {5,7,9,34,38}: 4	5.12 ШТК 104ВН-14С: {5,9,14,33,34,36}: 5
1.12 ШТК 102ВН-14С: {1,2,9}: 1	2.17 ШТК 104ВН-16С: {2,3,18}: 2	3.13 ШТК 104ВН-14С: {2,9,16,17,18,19}: 3	4.15 ШТК 103ВН-17С: {3,5,24,29}: 4	5.13 ШТК 103ВН-15С: {5,9,22,33,34}: 5
1.13 ШТК 102ТН-15С: {2,4,6}: 1	2.18 ШТК 104ВН-17С: {2,3,25}: 3	3.14 ШТК 102ВН-15С: {2,4,6,24}: 4		
1.14 ШТК 102Н-16С: {4,5,9}: 1		3.16 ШТК 102ТН-16С: {6,11,10,21}: 3		
1.15 ШТК 100ТН-17С: {1,5,6}: 1		3.15 ШТК 103ВН-17С: {5,9,22}: 3		
1.16 ШТК 102ВН-18С: {2,4,9}: 1				
1.17 ШТК 102Н-19С: {5,7,9}: 1				

2 клас

3 клас

Підходящі (+) - гіпотези		{+1}: {2,25}; {3,7, 3,9,}
--------------------------	--	---------------------------

Рисунок 6.1 - Причини некоректної класифікації проекту 2.18 {2,3,25}

Щоб система була здатна правильно класифікувати даний проект необхідно виключити відображення помилкової (+) - гіпотези 1-ого рангу {2,25} в класі 3 і сформулювати необхідну (+) - гіпотезу у 2-му класі.

Уявімо можливий спосіб поліпшення результату класифікації даного проекту і, як наслідок, підвищення інтерпретації гіпотез, за допомогою

спрямованого розширення бази фактів. Щоб сформувавши необхідну (+) - гіпотезу у 2-му класі досить мати досвід виробництва, як мінімум, ще одного проекту другого класу, до складу якого буде входити комплектуюча 25. У цьому випадку схожість даного проекту і проекту 2.18 породить (+) - гіпотезу 2-го класу, що містить комплектуючу 25.

Щоб виключити появу (+) - гіпотези {2,25} як гіпотезу 1-ого рангу в третьому необхідно знизити ранг даної гіпотези, для чого необхідно мати досвід виробництва як мінімум ще одного проекту 2-го класу, що містить набір комплектуючих {2, 25} і не містить комплектуючу 3. У цьому випадку схожість даного проекту і проекту 2.18 породить (+) - гіпотезу 2-го класу з набором утворюють {2,25}, що знизить ранг помилковою гіпотези 3-го класу. Таким чином, у 2-му класі необхідно додати два додаткових проекту (цілком конкретної конфігурації):

- перший (проект 2.20 {5,6,25}) призначений для формування в другому класі коректної (+) - гіпотези першого рангу ({25}), яка буде використана при класифікації;
- другий (проект 2.19 {2,9,25}) призначений для формування в другому класі коректної (+) - гіпотези третього рангу ({2,25}), поява якої спричинить за собою зниження рангу помилкової гіпотези {2,25} в третьому класі, що знизить можливість використання даної гіпотези при класифікації.

Отримані результати проілюстровані на малюнку 6.2. З малюнка видно, що проект 2.18 {2,3,25} класифікований коректно, а інтерпретованість гіпотез підвищилася.

ІС прогнозування тривалості виробництва				
Вихідні дані		ДСМ метод		
Проекти в класах :				
Клас 1	Клас 2	Клас 3	Клас 4	Клас 5
1.1 ШТК 102ВН-03С: {1,2,3}: 1	2.1 ШТК 104ВН-01С: {2,3,13}: 2	3.1 ШТК 102ВН-04С: {1,2,3,20}: 3	4.1 ШТК 102ВН-06С: {1,2,3,33,34}: 5	5.1 ШТК 103ВН-03С: {2,3,22,33,34}: 5
1.2 ШТК 100ВН-01С: {3,4,5}: 1	2.2 ШТК 104ВН-02С: {2,3,14}: 2	3.2 ШТК 102ТН-02С: {1,2,6,21}: 3	4.2 ШТК 102ТН-03С: {2,6,33,34,40}: 5	5.2 ШТК 103КН-02С: {5,33,34,29}: 5
1.3 ШТК 102ТН-01С: {1,2,6}: 1	2.4 ШТК 100: {1,2,6,16}: 2	3.3 ШТК 103ВН-01С: {2,3,22}: 3	4.3 ШТК 103ВН-02С: {2,3,22,24}: 4	5.3 ШТК 104ПТН-01С: {2,33,34,32,30,35}: 5
1.5 ШТК 100ВН-02А: {7,8,9}: 1	2.7 ШТК 104КН-03С: {2,18,19}: 2	3.4 ШТК 103ТН-01С: {2,6,23}: 3	4.4 ШТК 103КН-01С: {2,3,24,29}: 4	5.4 ШТК 103К-01С: {33,34,29,36}: 5
1.6 ШТК 100ВН-03С: {9,11,10}: 1	2.8 ШТК 100ВН-05С: {3,10,19,28}: 2	3.5 ШТК 102ВН-05С: {2,7,9,24}: 4	4.5 ШТК 104КН-05С: {2,3,20,25}: 4	5.5 ШТК 103ТН-02С: {34,32,35,37,39,38}: 5
1.7 ШТК 100ВН-04С: {3,5,10}: 1	2.9 ШТК 104ВН-11С: {5,9,13}: 2	3.7 ШТК 104ВН-05С: {2,3,16,17,25}: 3	4.6 ШТК 104КН-06С: {2,9,20,25}: 4	5.7 ШТК 104ВН-07С: {2,3,13,33,34,36}: 5
1.8 ШТК 102ВН-10С: {1,5,9}: 1	2.10 ШТК 100-1: {4,5,6,16}: 2	3.8 ШТК 104ВН-06С: {5,9,17,18,19}: 3	4.7 ШТК 102ВН-07С: {2,3,14,24,31}: 4	5.8 ШТК 102ТН-05С: {2,19,33,34,40,32}: 5
1.9 ШТК 102ВН-11С: {2,3,4}: 1	2.13 ШТК 104ВН-12С: {5,9,14}: 2	3.9 ШТК 104КН-04С: {2,25,26,27}: 2	4.10 ШТК 103ВН-12С: {3,5,22,24}: 4	5.9 ШТК 103КН-11С: {2,33,34,29}: 5
1.10 ШТК 100ВН-12С: {1,3,5}: 1	2.14 ШТК 104ВН-13С: {5,6,14}: 2	3.10 ШТК 103ВН-11С: {5,9,23}: 3	4.12 ШТК 102ВН-14С: {5,9,14,24,31}: 4	5.10 ШТК 102ТН-12С: {5,7,19,33,34,32}: 5
1.4 ШТК 100ТН-01С: {4,5,6}: 1	2.15 ШТК 104КН-14С: {5,18,19}: 2	3.11 ШТК 102ВН-12С: {4,5,9,20}: 3	4.13 ШТК 102ТН-15С: {1,2,3,34,38}: 4	5.11 ШТК 104ПТН-13С: {3,5,18,32,35,38}: 5
1.11 ШТК 100ВН-13С: {2,3,7}: 1	2.16 ШТК 100ВН-15С: {2,7,9,19}: 2	3.12 ШТК 104ВН-13С: {5,9,16,17,18}: 3	4.14 ШТК 102ТН-16С: {5,7,9,34,38}: 4	5.12 ШТК 104ВН-14С: {5,9,14,33,34,36}: 5
1.12 ШТК 102ВН-14С: {1,2,9}: 1	2.17 ШТК 104ВН-16С: {2,3,18}: 2	3.13 ШТК 104ВН-14С: {2,9,17,18,19}: 3	4.15 ШТК 103ВН-17С: {3,5,24,29}: 4	5.13 ШТК 103ВН-15С: {5,9,22,33,34}: 5
1.13 ШТК 102ТН-15С: {2,4,6}: 1	2.18 ШТК 104ВН-17С: {2,3,25}: 2	3.14 ШТК 102ВН-15С: {2,4,6,24}: 4		
1.14 ШТК 102ТН-16С: {4,5,9}: 1	2.19 ШТК 104ВН-18С: {2,9,25}: 2	3.16 ШТК 102ТН-16С: {6,11,10,21}: 3		
1.15 ШТК 100ТН-17С: {1,5,6}: 1	2.20 ШТК 104ТН-19С: {5,6,25}: 2	3.15 ШТК 103ВН-17С: {5,9,22}: 3		
1.16 ШТК 102ВН-18С: {2,4,9}: 1				
1.17 ШТК 102ТН-19С: {5,7,9}: 1				
	2 клас	3 клас		
Підходящі (+) - гіпотези до зміни даних		{+1}: {2,25}: {3,7, 3,9, }		
Підходящі (+) - гіпотези до зміни даних	{+1}: {25}: {2,18, 2,19, 2,20, }	{+3}: {2,25}: {3,7, 3,9, }		
	{+3}: {2,25}: {2,19, 2,18, }			

Рисунок 6.2 - Результати класифікації модифікованої навчальної вибірки

Однак, незважаючи на те що проект 2.18 тепер класифікований коректно, кількість неправильно класифікованих проектів не змінилося, так як проект 3-го класу 3.9 {2,25,26,27} вихідної вибірки, який раніше класифікувався вірно помилковою («штучною», що не інтерпретується коректними засобами) гіпотезою {2,25} (рис.6.1), внаслідок зниження рангу даної гіпотези, тепер класифікується невірно 2-м класом (відповідно до рисунком.6.2).

Для виправлення такого результату, необхідно отримати відповідну (+) - гіпотезу в третьому класі для коректної класифікації даного проекту. Щоб сформулювати необхідну (+) - гіпотезу в 3-му класі досить мати досвід виробництва як мінімум ще одного проекту третього класу, до складу якого буде входити комплектуючі 26 і 27, тоді схожість даного проекту і проекту 3.9 породить (+) - гіпотезу 3 го класу {26, 27}, в результаті чого буде здійснена коректна класифікація проекту. Отримані результати проілюстровані на малюнку 6.3. З малюнка видно, що додавання проекту 3.17 {5, 14, 26, 27} дозволило поліпшити як результати класифікації, так і інтерпретируемість гіпотез.

Клас 1	Клас 2	Клас 3	Клас 4	Клас 5
1.1 ШТК 102ВН-03С: {1,2,3}: 1	2.1 ШТК 104ВН-01С: {2,3,13}: 2	3.1 ШТК 102ВН-04С: {1,2,3,20}: 3	4.1 ШТК 102ВН-06С: {1,2,3,33,34}: 5	5.1 ШТК 103ВН-03С: {2,3,22,33,34}: 5
1.2 ШТК 100ВН-01С: {3,4,5}: 1	2.2 ШТК 104ВН-02С: {2,3,14}: 2	3.2 ШТК 102ТН-02С: {1,2,6,21}: 3	4.2 ШТК 102ТН-03С: {2,6,33,34,40}: 5	5.2 ШТК 103КН-02С: {5,33,34,29}: 5
1.3 ШТК 102ТН-01С: {1,2,6}: 1	2.4 ШТК 100: {1,2,6,16}: 2	3.3 ШТК 103ВН-01С: {2,3,22}: 3	4.3 ШТК 103ВН-02С: {2,3,22,24}: 4	5.3 ШТК 104ТТН-01С: {2,33,34,32,30,35}: 5
1.5 ШТК 100ВН-02А: {7,8,9}: 1	2.7 ШТК 104КН-03С: {2,18,19}: 2	3.4 ШТК 103ТН-01С: {2,6,23}: 3	4.4 ШТК 103КН-01С: {2,3,24,29}: 4	5.4 ШТК 103К-01С: {33,34,29,36}: 5
1.6 ШТК 100ВН-03С: {9,11,10}: 1	2.8 ШТК 100ВН-05С: {3,10,19,28}: 2	3.5 ШТК 102ВН-05С: {2,7,9,24}: 4	4.5 ШТК 104КН-05С: {2,3,20,25}: 4	5.5 ШТК 103ТТН-02С: {34,32,35,37,39,38}: 5
1.7 ШТК 100ВН-04С: {3,5,10}: 1	2.9 ШТК 104ВН-11С: {5,9,13}: 2	3.7 ШТК 104ВН-05С: {2,3,16,17,25}: 3	4.6 ШТК 104КН-06С: {2,9,20,25}: 4	5.7 ШТК 104ВН-07С: {2,3,13,33,34,36}: 5
1.8 ШТК 102ВН-10С: {1,5,9}: 1	2.10 ШТК 100-1: {4,5,6,16}: 2	3.8 ШТК 104ВН-06С: {5,9,17,18,19}: 3	4.7 ШТК 102ВН-07С: {2,3,14,24,31}: 4	5.8 ШТК 102ТН-05С: {2,19,33,34,40,32}: 5
1.9 ШТК 102ВН-11С: {2,3,4}: 1	2.13 ШТК 104ВН-12С: {5,9,14}: 2	3.9 ШТК 104КН-04С: {2,25,26,27}: 3	4.10 ШТК 103ВН-12С: {3,5,22,24}: 4	5.9 ШТК 103КН-11С: {2,33,34,29}: 5
1.10 ШТК 100ВН-12С: {1,3,5}: 1	2.14 ШТК 104ВН-13С: {5,6,14}: 2	3.10 ШТК 103ВН-11С: {5,9,23}: 3	4.12 ШТК 102ВН-14С: {5,9,14,24,31}: 4	5.10 ШТК 102ТН-12С: {5,7,19,33,34,32}: 5
1.4 ШТК 100ТН-01С: {4,5,6}: 1	2.15 ШТК 104КН-14С: {5,18,19}: 2	3.11 ШТК 102ВН-12С: {4,5,9,20}: 3	4.13 ШТК 102ТН-15С: {1,2,3,34,38}: 4	5.11 ШТК 104ТТН-13С: {3,5,18,32,35,38}: 5
1.11 ШТК 100ВН-13С: {2,3,7}: 1	2.16 ШТК 100ВН-15С: {2,7,9,19}: 2	3.12 ШТК 104ВН-13С: {5,9,16,17,18}: 3	4.14 ШТК 102ТН-16С: {5,7,9,34,38}: 4	5.12 ШТК 104ВН-14С: {5,9,14,33,34,36}: 5
1.12 ШТК 102ВН-14С: {1,2,9}: 1	2.17 ШТК 104ВН-16С: {2,3,18}: 2	3.13 ШТК 104ВН-14С: {2,9,17,18,19}: 3	4.15 ШТК 103ВН-17С: {3,5,24,29}: 4	5.13 ШТК 103ВН-15С: {5,9,22,33,34}: 5
1.13 ШТК 102ТН-15С: {2,4,6}: 1	2.19 ШТК 104ВН-18С: {2,9,25}: 2	3.14 ШТК 102ВН-15С: {2,4,6,24}: 4		
1.14 ШТК 102Н-16С: {4,5,9}: 1	2.20 ШТК 104ТН-19С: {5,6,25}: 2	3.16 ШТК 102ТН-16С: {6,11,10,21}: 3		
1.15 ШТК 100ТН-17С: {1,5,6}: 1	2.18 ШТК 104ВН-17С: {2,3,25}: 2	3.15 ШТК 103ВН-17С: {5,9,22}: 3		
1.16 ШТК 102ВН-18С: {2,4,9}: 1		3.17 ШТК 103КН-17С: {5,14,26,27}: 3		
1.17 ШТК 102Н-19С: {5,7,9}: 1				

	2 клас	3 клас
Підходящі (+) - гіпотези до зміни даних	(+1): {25,}: {2.18, 2.19, 2.20,} (+3): {2,25,}: {2.19, 2.18, }	(+3): {2,25,}: {3.7, 3.9, }
Підходящі (+) - гіпотези до зміни даних	(+1): {25,}: {2.18, 2.19, 2.20,} (+3): {2,25,}: {2.19, 2.18, }	(+1): {26,27,}: {3.9, 3.17, } (+3): {2,25,}: {3.7, 3.9, }

Рисунок 6.3- Результати класифікації модифікованої навчальної вибірки

Таким чином, для поліпшення якості класифікації слід формувати відсутні гіпотези за допомогою спрямованого додавання нових проектів в ті класи, де немає достатнього обсягу даних для коректного (див. приклад, здійсненність умови каузальної повноти і ін.) навчання ДСМ-системи, і зниження ранги некоректних гіпотез в інших класах також за допомогою додавання нових фактів (також - цілком певного виду). Пропоноване в такій схемі спрямоване розширення навчальної вибірки слід вести до її «насичення» - ситуації, коли всі наявні в ній (наприклад, на поточний момент ДСМ міркування) емпірична залежності причини досліджуваних явищ повністю (нехай і в неявному вигляді - через опису відповідних прецедентів) представлені наявними даними для ДСМ-навчання.

ВИСНОВКИ

У даній роботі був розроблений ряд алгоритмів, що дозволяють вирішити поставлену задачу прогнозування класу тривалості виробництва технічного об'єкта (кліматичної шафи) зі складною структурою до його детального проектування і розробки. Дані рішення відображені у вигляді закінченого програмного засобу аналізу і класифікації описів розглянутих технічних об'єктів, яке спільно з використовуваною базою даних становить проблемну ІС.

Аналіз результатів роботи даної системи показав, що гіпотези (каузальні підстави обговорюваної класифікації), виявлені у вихідних даних розробленої ІС, є легко інтерпретованими і обгрунтованими. Очевидним чином простежується тенденція ускладнення елементів, що входять до складу гіпотез, відповідних певного класу тривалості виробництва, при збільшенні номера класу, а, отже, ускладненні структури технічних об'єктів, що відповідають даному класу. Перевірка класифікації вихідних даних за допомогою виявлених гіпотез, яка є стандартною операцією в рамках ДСМ- методу, що проводиться для перевірки обгрунтованості гіпотез, показала, що більше частина об'єктів у вихідній базі даних були класифіковані вірно.

Крім того, коректність роботи даної системи була досліджена на нових проектах виробництва виробів. Результати показали, що ІС класифікує некоректно або не виробляє класифікацію тільки в тих випадках, коли в структурі нових виробів містяться нові (для неї) або унікальні елементи, що не використалися до цього при створенні жодного об'єкта. Даний факт не говорить про помилки в моделі класифікації, а лише відображає необхідність поповнення бази фактів новими прикладами.

Таким чином, аналіз і оцінка результатів роботи даної системи показали, що результати класифікації об'єктів за допомогою розробленої ІС переважно є вірними, а гіпотези, на основі яких відбувається класифікація є змістовно інтерпретованими і, як наслідок, інформативними і обгрунтованими.

Розроблена ІС може бути легко адаптована для вирішення завдання прогнозування будь-яких оціночних характеристик об'єкта (представленого безліччю бінарних ознак - наявних і \ або відсутніх архітектурно-функціональних характеристик) в умовах неповної інформації про властивості аналізованих об'єктів (тобто до їх детального проектування) внесенням лише невеликих змін, налаштувань в інструментальні засоби розробленої ІС.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Забезжайло М.І. Про деякі переборні завдання, що виникають при автоматичному породженні гіпотез ДСМ-методом// НТІ. Сер. 2. 1988.
2. Забезжайло М.І., Івашко В.Г., Кузнецов С.О., Міхеєнкова М.А., Хазановський К.П., Аншаков О.М. Алгоритмічні та програмні засоби ДСМ-методу автоматичного породження гіпотез // НТІ. Сер. 2. 1987. №10. 1-14 с.
3. Кузнецов С.О. Швидкий алгоритм побудови всіх перетинів об'єктів// НТИ. Сер.2. 1993. №1. С. 17-20.
4. Фінн В.К. Міркування в інтелектуальних системах типу ДСМ // Підсумки науки і техніки, сер. "Інформатика". 1991. Т. 15. 54-101 с.
5. Фінн В.К. Синтез пізнавальних процедур і проблема індукції. // НТІ. Сер. 2. 1999. № 1-2. 8-45 с.
6. Мілль Дж. С Система логіки силлогістичної і індуктивної. М.; Книжкова справа, 1900. - 781 с.
7. Hutchinson A. Algorithmic learning. Clarendon Press, Oxford, 1994.
8. Бекон Ф. Новий Органон. Ленінське відділення, 1935.
9. Фінн В.К. Про можливості формалізації правдоподібних міркувань засобами багатозначних логік // Всесоюз. симпозіум з логіки та методології науки. Київ: Наукова думка, 1976. 82-83 с.
10. Фінн В.К. Бази даних з неповною інформацією і новий метод автоматичного породження гіпотез // Діалогові та фактографічні системи інформаційного забезпечення. М. 1981. 153 – 156 с.
11. Чубукова І. А. Data Mining: учбовий посібник / І.А. Чубукова. 2 –е вид., М.: Інтернет – Університет Інформаційних технологій; БІНОМ. Лабораторія знань, 2008. – 382 с
12. Pierce C.S. Philosophical writings/ Ed. J. Buchel. – N.Y.: Dover Publ. Co., 1955.
13. Гершель Д. Філософія природознавства. Про загальний характер, користь і принципах дослідження природи, 1989.

14. Уевел У. Історія індуктивних наук від найдавнішого і до теперішнього часу: В 3 Т. СПб.: Російська книжкова торгівля, 1867-1869.
15. Greniewski H. Elementy logiki indukcji. Warszawa, 1955.
16. Greniewski H.H. Milla Kanon Zmian towarzyszcych. *Studia Logica*. T. V. 1957. P. 109-126.
17. Автоматичне породження гіпотез в інтелектуальних системах. Сост. Е.С. Панкратова, В. К. Фінн. /під. заг. ред. В.К. Фінна. Прередм. Ю.М. Арского. М.: Книжковий дім «Ліброком», 2009.
18. ДСМ – метод автоматичного породження гіпотез: Логічні та епістемологічні основи. Сост. О.М. Аншаков, Е.Ф. Фабрикантова / під. заг. ред. О.М. Аншакова. М.: Книжковий дім «Ліброком», 2009.
19. Фінн В.К. Індуктивні методи Д.С. Мілля в системах штучного інтелекту. Ч.1// Штучний інтелект і прийняття рішень / заг. ред. акад. С.В. Емельянов. М.: Ленанд/ URSS. 2010. №3. 3-21 с.
20. Фінн В.К. Індуктивний метод сполученого подібності - відмінності і процедурна семантика ДСМ-метода// Науково-технічна інформація, сер.2. 2010. №4. 1-17 с.
21. Поппер К.П. Об'єктивне знання. Еволюційний підхід. М.:УРСС. 2002.
22. Арський Ю.М., Фінн В.К. Принципи конструювання інтелектуальних систем // Інформаційні технології та обчислювальні системи. 2008. №4. 4- 37 с.
23. Pierce C.S. *Abdoction and induction. Philosophical writings of Pierce*. Ed. Buchel. Dover Publication. N 4. P. 150-156.
24. *Abductive Inference: Computation, Philosophy, Technology*. Ed. J.R. Josephson.- Cambridge Univ. Press, 1994.
25. Пойа Д. Математика і правдоподібні міркування. М.: Наука, 1975.
26. Скворцов Д.П. Про деякі способи побудови логічних мов з кванторами по кортежам // Семіотика і інформатика. 1982. № 2. 102- 126 с.
27. Барвайс Д. Введення в логіку першого порядку. Довідкова книга з математичної логіки. Частина I, Теорія моделі. М.: Наука. 1982. 51-52 с.

28. Бочвар Д. А. Про один трифазному обчисленні і його застосуванні до аналізу парадоксів класичного розширеного функціонального числення // Математичний збірник. Т.4. Вип.2. 1938. 287-308 с.
29. Баргесян А.А Аналіз даних і процесів: навч. посібник /А.А. Баргесян, М.С. Куприянов, И.И. Холод, М.Д. Тесс, С.И. Елизарову – 3-е вид. БХВ – Петербург, 2009. - 512 с.
30. Лбов Г.С. Методи обробки різнотипних експериментальних даних. Новосибирск: Наука, 1981.
31. Белецкий Н.К. Застосування комітетів для многокласовой класифікації. Чисельний аналіз вирішення завдань лінійного і опуклого програмування. Свердловськ. 1983. 156-162 с.
32. Marchand M., Shah M., Shawe-Taylor J., Sokolova M. The set covering machine with data-dependent half-spaces // Proc. 20th International Conf. on Machine Learning. Morgan Kaufmann, 2003. P. 520-527.
33. Аншаков О.М. Про одну інтерпретації ДСМ-методу автоматичного породження гіпотез. Науково - технічна інформація. Сер.2. Інформаційні процеси та системи / Всеросійський інститут наукової і технічної інформації РАН. 1999. № 1-2. 45-53 с.
34. Фінн В.К. Штучний інтелект: Методологія, застосування, філософія. М.: КРАСАНД, 2011. - 448 с.
35. Костянтин Воронцов. Курс лекцій «Математичні методи навчання по прецедентах». МФТІ, 2004-2008.
36. Quinlan J.R. Induction on Treasures Treeds // Machine Learning, 1986. V. 1. P. 81- 106.
37. MACHINELEARNING.URL: <http://www.machinelearning.ru>
38. Quinlan J.R. Improved Use of Continuous Attributes in C 4.5 // J. of Artif. Intell. Res, 1996. V.4. P. 77-90.
39. J. Shafer, R. Agrawal, and M. Mehta. «SPRINT: A Scalable Parallel Classifier for Data Mining», Proc. 22nd Int'l Conf. Very Large Data Bases, Morgan Kaufman, San Francisco, 1996. P. 544-555.

40. Варшавський П.Р., Єремєєв А.П. Моделювання міркувань на основі прецедентів в інтелектуальних системах підтримки прийняття рішень // РАН, Штучний інтелект і прийняття рішень. 2009. №2. 45-57 с.

41. Дюк В., Самойленко А. Data mining: навчальний курс. СПб: Пітер, 2001. - 368 с.

42. Фінн В.К. Індуктивний метод сполученого подібності - відмінності і процедурна семантика ДСМ-метода// Науково-технічна інформація, сер.2. 2010. №4. 1-17 с.

ДОДАТОК А

(обов'язковий)

Дерева та списки рішень

В даному розділі будуть додатково використовуватися наступні позначення:

X – множина об'єктів;

$Y = \{1, \dots, M\}$ – множина імен класів;

$X^l = \{x_1, \dots, x_l\} * X$ – навчальна вибірка;

$y_i = y * (x_i), i = 1, \dots, l$ – відомі відповіді;

$\varphi: X \rightarrow \{0,1\}$ – деякий предикат, визначений на множині X .

А.1 Списки

Вирішальний список (decisionlist, DL) - це найбільш простий логічний алгоритм, як за своєю структурою, так і за способом побудови.

Вирішальний список - це алгоритм класифікації $a: X \rightarrow Y$, який задається набором закономірностей $\varphi_1(x), \dots, \varphi_T(x)$, приписаних до класів $c_1, \dots, c_T \in Y$ відповідно, і обчислюється згідно з алгоритмом, представленим на рисунку А.1.

1: для всіх $t=1, \dots, T$

2: якщо $\varphi_1(x)=1$ то

3: повернути c_t ;

4: повернути c_0 .

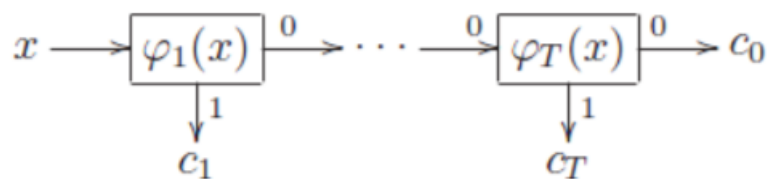


Рисунок А.1- Класифікації об'єкта $x \in X$ вирішальним списком

Логіку вирішального списку або, комітету старшинства, мають багато алгоритмів, що пропонувалися в різний час різними авторами під різними назвами. Численні варіанти відрізняються вибором сімейства предикатів, використовуваними критеріями інформативності предикатів і методами пошуку інформативних предикатів. Приклади алгоритмів, що реалізують логіку вирішального списку:

1. Найбільш поширені вирішальні списки кон'юнкцій. Вони майже ідеально відповідають людській логіці прийняття рішень, заснованої на послідовній перевірці досить простих правил.
2. Алгоритм дрібних еталонів ДРЕТ [30] заснований на покритті вибірки кулями, і відрізняється тим, що список будується від кінця до початку. На першому кроці для кожного з класів визначається куля мінімального радіусу, що включає всі навчальні об'єкти даного класу. Якщо кулі різних класів перетинаються, то для об'єктів, що потрапили в перетин, знову будуються покривають кулі, але вже меншого радіуса. Процес побудови куль триває, поки в перетинах куль залишаються представники різних класів. Побудовані кулі утворюють вирішальний список в порядку зростання їх радіусів.
3. Алгоритм Білецького [31] будує півплощини, що відокремлюють якомога більше об'єктів одного класу, що рівносильно максимізації інформативності. Для цього застосовуються методи лінійного програмування. Після побудови кожної напівплощини робиться спроба знайти кращі положення попередніх напівплощин.
4. В алгоритмі Маршанда [32] перебираються різноманітні гіперплощини, що розділяють три точки (data dependent halfspaces), і з них вибирається напівплощина з максимальною інформативністю.

Основні переваги вирішальних списків:

- інтерпретація і простота класифікації.
- гнучкість: в залежності від вибору безлічі предикатів можна будувати досить різноманітні алгоритмічні конструкції.

- можливість обробки різнотипних даних і даних з пропусками.

Основні недоліки вирішальних списків:

- якщо безліч правіл вибрано невдало, список може не побудуватися. При цьому можливий високий відсоток відмов від класифікації.
- можлива втрата інтерпретується, якщо список довгий і правила різних класів слідуєть упереміж. У цьому випадку правила не можуть бути інтерпретовані окремо, без урахування попередніх правил, і логіка їх взаємодії стає досить заплутаною.
- кожен об'єкт класифікується лише одним правилом, що не дозволяє правилам компенсувати неточності один одного. Даний недолік усувається шляхом голосування правил, але це вже інший алгоритм.

А.2 Древа розв'язання

Вирішальне дерево (decision tree, DT) - це ще один логічний алгоритм класифікації, заснований на пошуку кон'юнктивних закономірностей. Але, на відміну від вирішального списку, при синтезі дерева всі кон'юнкції будуються одночасно.

Вперше дерева рішень були запропоновані Ховілендом і Хантом (Hoveland, Hunt) в кінці 50-х років минулого століття.

Бінарне вирішальне дерево - це алгоритм класифікації, задати бінарним деревом, в якому кожної внутрішньої вершині $v \in V$ приписаний предикат $\beta_v: X \rightarrow \{0,1\}$ кожній вершині $v \in V$ приписано ім'я класу $c_v \in Y$. При класифікації об'єкта $x \in X$ він проходить по дереву шлях від кореня до деякого листа, відповідно до алгоритму, представленому на рисунку А.2.

1: $v := v_0$;

2: поки вершина v внутрішня

3: якщо $\beta_v(x) = 1$ то

- 4: $v := R_v$; перехід вправо
- 5: інакше
- 6: $v := L_v$; перехід вліво
- 7: повернути c_v .

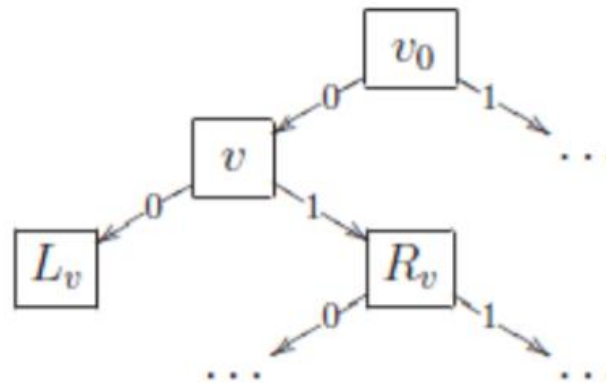


Рисунок А.2 - Класифікація об'єкта $x \in X$ бінарним вирішальним деревом

Завдання побудови дерева мінімальної складності, правильно класифікуючого задану вибірку, в загальному випадку є NP-повною задачею. На практиці застосовують різні, більш чи менш вдалі, евристики, націлені на побудову якомога більше простого дерева, що володіє якомога кращими якостями класифікації. Вибір евристик неоднозначний, як зазвичай буває в таких випадках.

А.3 Зважене голосування правил

Припустимо, є консиліум експертів, кожному члену якого може припуститися помилка. Процедура голосування - це спосіб підвищення якості прийнятих рішень, при якому помилки окремих експертів компенсують один одного.

Далі розглянемо застосування принципу голосування для побудови композиції, що складаються з логічних закономірностей.

Принцип голосування:

Нехай для кожного класу $c \in Y$ побудовано безліч логічних закономірностей (правил), що спеціалізуються на розрізненні об'єктів даного класу:

$$R_c = \{\varphi_c^t: X \rightarrow \{0,1\} | t = 1, \dots, T_c\}.$$

Вважається, що якщо $\varphi_c^t(x) = 1$, то правило φ_c^t відносить об'єкт $x \in X$ до класу c . Якщо ж $\varphi_c^t(x) = 0$, то правило φ_c^t утримується від класифікації об'єкта x .

Алгоритм простого голосування (simplevoting) підраховує частку правил в наборах R_c , що відносять об'єкт x до кожного з класів.

$R_c(x) = \frac{1}{T_c} \sum_{t=1}^{T_c} \varphi_c^t(x)$, $c \in Y$, і відносить об'єкт x до того класу, за який подана найбільша частка голосів: $a(x) = \arg \max R_c(x)$.

Якщо максимум досягається одночасно на декількох класах, вибирається той, для якого ціна помилки менше. Нормуючий множник $\frac{1}{T_c}$ вводить для того, щоб набори з великим числом правил не перетягували об'єкти в свій клас.

Алгоритм зваженого голосування (weightedvoting, WV) діє більш тонко, з огляду на, що правила можуть мати різну цінність. Кожному правилу φ_c^t приписується вага ж $\alpha_c^t \geq 0$, і при голосуванні береться зважена сума голосів:

$$R_c(x) = \sum_{t=1}^{T_c} \alpha_c^t \varphi_c^t(x), \alpha_c^t \geq 0.$$

Основні переваги алгоритму голосування:

- інтерпретування і простота класифікації.
- гнучкість: в залежності від вибору безлічі предикатів можна будувати досить різноманітні алгоритмічні конструкції.
- правила компенсують неточності один одного.

Лістинг А.1 Клас описує вершину дерева пошуку

```
public class GLAlgorithmNode {
    List<Project> Y;
    List<ProjectElement> h;
    int i, j;
    GLAlgorithmNode parent;
    public GLAlgorithmNode() {
    }
    public GLAlgorithmNode(List<Project> Y, List<ProjectElement> h,
int i, int j) {
        this.Y = Y;
        this.h = h;
        this.i = i;
        this.j = j;
    }
    public List<Project> getY() {
        return Y;
    }
}
```

Лістинг А.2 Клас реалізує алгоритм пошуку глобальних подібностей

```

public class GLAlgorithm {
    private List<Project> P;
    private List<GLAlgorithmNode> resList = new ArrayList();
    private GLAlgorithmNode curNode;
    private void prepareProjects(List<Project> projects) {
        for (int i=0;i<projects.size();i++) {
            projects.get(i).setDSMId(i);
        }
    }
    public GLAlgorithm(List<Project> P) {
        this.P=P;
        prepareProjects(P);
    }
    public GLAlgorithmNode findFirstNode() {
        //ШАГ 0
        if (P.isEmpty()) {
            return null;
        }
        GLAlgorithmNode rootNode = new GLAlgorithmNode();//Корень
        rootNode.Y = new ArrayList();
        rootNode.h = new ArrayList();
        rootNode.i = -1;
        rootNode.j = 1;
        rootNode.parent=null;
        return rootNode;
    }

    public GLAlgorithmNode findNextNode() {
        //ШАГ 1
        if (curNode.i<P.size()-1) {
            curNode.i++;
            if (curNode.j < curNode.Y.size()) {
                if (curNode.i
curNode.Y.get(curNode.j).getDSMId()) {
                    curNode.j++;
                    return curNode;
                }
            }
        } else {
            return backTracking();
        }
        Project Xi = P.get(curNode.i);
        //ШАГ 2
        List<Project> Yc;
        List<ProjectElement> hc;
        int ic,jc;
        ic = curNode.i;
        if (!curNode.Y.isEmpty()) {
            hc=intersect(curNode.h, Xi.getElements());
        } else {
            hc=Xi.getElements();
        }
        if (curNode.Y.size()<1) {
            Yc = new ArrayList();
            Yc.add(Xi);
            jc=Yc.size();
            GLAlgorithmNode extNode = new
GLAlgorithmNode(Yc,hc,ic,jc);
            extNode.parent=curNode;
            return extNode;
        }
        if (hc.isEmpty()) {
            return curNode;
        }
        Yc = getProjectsWithElements(hc);
        if (Yc.get(curNode.j).equals(Xi)) {
            jc = curNode.j+1;
            GLAlgorithmNode closingNode = new
GLAlgorithmNode(Yc,hc,ic,jc);
            closingNode.parent = curNode;
            resList.add(closingNode);
            return closingNode;
        } else {
            return curNode;
        }
    }
}

```

```

public List<Project> getProjectsWithElements(List<ProjectElement> e) {
    List<Project> res = new ArrayList();
    for (int i =0;i<P.size();i++) {
        if (P.get(i).getElements().containsAll(e)) {
            res.add(P.get(i));
        }
    }
    return res;
}

public List<ProjectElement> intersect( List<ProjectElement> set1,
List<ProjectElement> set2) {
    List<ProjectElement> res = new ArrayList();
    res.addAll(set1);
    res.retainAll(set2);
    return res;
}

private GLAlgorithmNode backTracking() {
    GLAlgorithmNode parent = curNode.parent;
    while (parent!=null) {
        if (parent.i<P.size()-1) {
            return parent;
        }
        parent = parent.parent;
    }
    return null;
}

private List<Fragment> convertResListToFragments() {
    List<Fragment> fList = new ArrayList();

    Fragment f;
    FragmentElement e;
    GLAlgorithmNode node;
    for (int i=0;i<resList.size();i++) {
        node = resList.get(i);
        f = new Fragment();
        for(int j=0;j<node.getH().size();j++) {
            e = new FragmentElement();

e.setNomenclatorItem(node.getH().get(j).getNomenclatorItem());
            e.setCount(node.getH().get(j).getCount());
            f.addFragmentElement(e);
        }
        f.addProjects(node.getY());
        fList.add(f);
    }
    return fList;
}

public List<Fragment> searchProcess() {
    curNode = findFirstNode();
    while (curNode != null) {
        curNode = findNextNode();
    }
    return convertResListToFragments ();
}
}
}

```

ДОДАТОК Б

(обов'язковий)

Класи реалізації компонентів

Лістинг Б.1 Інтерфейс і клас реалізації компонента Learner

```

public interface Learner {
    public void findAllHypotesesInClasses(List<TimeClass> tcList);
}
public class LearnerImpl implements Learner {
    @Override
    public void findAllHypotesesInClasses(List<TimeClass> tcList) {
        for (TimeClass cur: tcList) {

            List<Project> plusExamples = cur.getProjects();
            cur.setFragments(findExamplesLikeness(plusExamples,+1));
            List<Project> minusExamples =
            findLowMinusExamples(cur,tcList);
            minusExamples.addAll(findHighMinusExamples(cur,tcList));
            cur.addFragments(findExamplesLikeness(minusExamples,-
1));
        }
        find1234PlusHypoteses(tcList);
        findPlus5AndMinusHypoteses(tcList);
    }

private List<TimeClass> findClassesWithPlusLikeness(Fragment l,
List<TimeClass> tcList) {
    List<TimeClass> classes = new ArrayList();
    for (TimeClass c:tcList) {
        for(Fragment f: c.getFragments()) {
            if (f.getEstimate())>0) {
                if (f.equals(l)) {
                    classes.add(c);
                }
            }
        }
    }
    return classes;
}

private int findIndexofMinusLikeness(TimeClass c, Fragment l) {
    for(int i=0;i<c.getFragments().size();i++) {
        Fragment f = c.getFragments().get(i);
        if (f.getEstimate())<0) {
            if (f.equals(l)) {
                return i;
            }
        }
    }
    return -1;
}

private int findIndexofPlusLikeness(Fragment l,TimeClass c) {
    for(int i=0;i<c.getFragments().size();i++) {
        Fragment f = c.getFragments().get(i);
        if (f.getEstimate())>0 && f.getEstimate())<5) {

            if (f.equals(l)) {
                return i;
            }
        }
    }
    return -1;
}
}

```

```

private void find1234PlusHypoteses(List<TimeClass> tcList) {
    for (TimeClass cur: tcList) {
        for (int i=0; i<cur.getFragments().size(); i++) {
            Fragment curF = cur.getFragments().get(i);
            if (curF.getEstimate() > 0) {
                {
                    Int index = findIndexofMinusLikeness(cur,curF);
                    if (index > -1) {
                        curF.setEstimate(2);
                        cur.getFragments().remove(index);
                        List<TimeClass> tmp = new ArrayList();
                        tmp.addAll(tcList); tmp.remove(cur);
                    int count = findClassesWithPlusLikeness(curF,tmp).size();
                    if (count == 1) {
                        curF.setEstimate(3);
                    }
                    if (count > 1) {
                        curF.setEstimate(4);
                    }
                }
            }
        }
    }
}

private void findPlus5AndMinusHypoteses(List<TimeClass> tcList) {
    for (int i = 0; i < tcList.size(); i++) {
        TimeClass cur = tcList.get(i);
        for (int k = 0; k< cur.getFragments().size(); k++) {
            Fragment curF = cur.getFragments().get(k);
            if (curF.getEstimate() == -1) {
                {
                    boolean findedPlusLikness = false;
                    for (int j = 0; j < tcList.size(); j++) {
                        if (i != j) {
                            TimeClass other = tcList.get(j);
                            if (findIndexofPlusLikeness(curF,other)>-1) {
                                if (j<i) {
                                    curF.setEstimate(5);
                                }
                                findedPlusLikness = true;
                                break;
                            }
                        }
                    }
                    if (!findedPlusLikness) {
                        cur.getFragments().remove(k);
                        k--;
                    }
                }
            }
        }
    }
}

private List<Project> findHighMinusExamples(TimeClass cur,
List<TimeClass> tcList) {
    List<Project> minusExamples = new ArrayList();
    for (int j=cur.getId();j<tcList.size();j++) {
        minusExamples.addAll(tcList.get(j).getProjects());
    }
    return minusExamples;
}

```

```

private List<Project> findLowMinusExamples(TimeClass cur,
List<TimeClass> tcList) {
    List<Project> minusExamples = new ArrayList();
    for (int j=cur.getId()-2;j>=0;j--) {
        minusExamples.addAll(tcList.get(j).getProjects());
    }
    return minusExamples;
}

private List<Fragment> findExamplesLikeness(List<Project> examples,
int estimate) {
    List<Fragment> resList;
    GLAlgorithmDebug gla = new GLAlgorithmDebug(examples);
    resList = gla.searchProcess();
    for (Fragment f : resList) {
        f.setEstimate(estimate);
    }
    return resList;
}
}
}

```

Лістинг Б.2 Інтерфейс і клас реалізації компонента Solver

```

public interface Solver {
    public TimeClass classifyProject(Project p, List<TimeClass>
classes);
}

public class SolverImpl implements Solver{
    List<List<Fragment>> plusH;
    List<List<Fragment>> minusH;

    @Override
    public TimeClass classifyProject(Project p, List<TimeClass> classes) {
        return runRules(p,classes);
    }

    private TimeClass runRules(Project p, List<TimeClass> C) {
        plusH = new ArrayList();
        minusH = new ArrayList();
        Fragment X = Fragment.fragmentFromProject(p);
        for (int i=0;i<6;i++) {

            plusH.add(new ArrayList());
        }
        for (int i=0;i<C.size();i++) {
            TimeClass c = C.get(i);
            for (int j=0;j<C.size();j++) {
                plusH.get(j).add(new ArrayList());
            }
            minusH.add(new ArrayList());
            for (Fragment V : c.getFragments()) {
                if (V.isSubSetOf(X)) {
                    if (V.getEstimate() > 0) {
                        plusH.get(V.getEstimate()-1).get(i).add(V);
                    } else
                    if (V.getEstimate() < 0) {
                        minusH.get(i).add(V);
                    }
                }
            }
        }
    }
}
}

```


Б.1 Формальна структура ДСМ-методу АПГ

В роботі [19] розглянуті формалізації всіх п'яти індуктивних методів Д.С. Мілля як засобів когнітивних міркувань, що є компонентами ДСМ-методу АПГ.

В [17] і [18] були розглянуті формалізації (засобами спеціального класу багатозначних логік) методу подібності та деяких його підсилень, а також їх застосування в інтелектуальних системах типу ДСМ. Нижче наведено варіант формального опису методу подібності. Саме цей метод лежить в основі розглянутого в даній роботі використання ДСМ-методу АПГ.

Вже згадана версія ДСМ-методу АПГ заснована на процедурній семантиці PrSem з булевою структурою даних для БФ [20].

Нехай $U^{(1)}$ і $U^{(2)}$ вихідні множини об'єктів і властивостей, відповідно, а $B_i = \langle 2^U, \emptyset, U^{(1)}, -, \cup, \cap \rangle$, булеві алгебри ($i = 1, 2$) утворюють структуру даних ДСМ-методу АПГ. Предикати $x \Rightarrow_1 Y$ і $x \Rightarrow_2 Y$ визначаються за допомогою відображень: $\Rightarrow_i: 2^{U^{(1)}} \times 2^{U^{(2)}} \rightarrow V_{in}$, де $i=1, 2$, а $V_{in} = \{ \langle v, n \rangle \mid (v \in \{1, -1, 0\}) \wedge (n \in N) \} \cup \{ \langle \tau, n \rangle \mid n \in N \}$; N - безліч натуральних чисел, 1, -1, 0, τ - типи істиннісних значень, відповідно; $\langle v, n \rangle$ - істинності значення (n - їх ступінь правдоподібності, що виражає число застосувань правил правдоподібного виведення; а $\langle \tau, n \rangle$ - безліч істиннісних значень. $\langle \tau, n \rangle$ характеризується рекурентним співвідношенням $\langle \tau, n \rangle = \{ \langle 1, n+1 \rangle, \langle -1, n+1 \rangle, \langle 0, n+1 \rangle, \langle \tau, n+1 \rangle \}$ виражає можливі істинності значення гіпотез, породжуваних правдоподібним ДСМ судженням; V_{in} - множина внутрішніх (емпіричних) істиннісних значень в сенсі Д.А. Бочвара [28].

Ці істинності значення є оцінками фактів, якщо $n = 0$, і є оцінками гіпотез, якщо $n > 0$.

За допомогою ж V_{ex} позначимо безліч зовнішніх істиннісних значень в сенсі Д.А. Бочвара: $V_{ex} = \{f, t\}$ де f і t істинні значення двозначної логіки «істина» і «брехня», відповідно. Вони приписуються формулами, побудованим з термів, операцій і відносин булевої алгебри множин таким, що все входження термів

знаходяться в сфері дії J-операторів. J-оператори [18] визначаються стандартним чином:

$$J_v = \begin{cases} t, \text{ якщо } V[\varphi] = v \\ f, \text{ якщо } V[\varphi] \neq v \end{cases}$$

Структура даних SD розглянутої версії ДСМ-методу АПГ представима реляційної системою: $SD = \langle B_1, B_2, \Rightarrow_1, \Rightarrow_2 \rangle$ де предикати \Rightarrow_1 і \Rightarrow_2 мають істинності значення з V_{in} , а формули булевою структури даних і J-формули мають істинності значення з V_{ex} .

Першому правилу Д.С. Мілля (індуктивному методу подібності) в ДСМ методі АПГ відповідають правила правдоподібного виведення 1-го роду (п.п.в.-1), які визначаються за допомогою двох предикатів подібності $M_{a,n}^+(V, W)$ і $M_{a,n}^-(V, W)$. Ці M-предикати визначаються, відповідно, за допомогою параметричних предикатів $M_{a,n}^+(V, W)$ де k - параметр, що виражає число подібних (σ) -Теана.

M - предикати виражають такі умови, уточнюючі і формалізують міллевську характеристику індуктивних методів:

(ЕУ) - екзистенціальні умови (існування (+) - або (-) -прикладів);

(СХ) - умова подібності (σ) -Теана ($\sigma \in \{+ -\}$);

(ЕЗ) - емпірична залежність, яка представляє причинне змушення відповідного слідства;

(УВ) - умова вичерпності безлічі подібних прикладів (їх максимальне угруповання);

k - нижня межа числа розглянутих прикладів ($k \geq 2$).

Нижче визначимо предикат позитивного подібності $M_{a,n}^+(V, W, k)$ залежить від параметра k для n -го ($n \geq 0$) застосування правил правдоподібних висновків:

$$M_{a,n}^+(V, W, k) = \exists Z_1 \dots \exists Z_k \exists U_1 \dots \exists U_k$$

$$\begin{aligned} \tilde{M}_{a,n}^+(V,W,k) &\Leftrightarrow \exists Z_1 \dots \exists Z_k \exists U_1 \dots \exists U_k ((J_{(1,n)}(Z_1 \Rightarrow_1 U_1) \wedge \dots \wedge J_{(1,n)}(Z_k \Rightarrow_1 U_k)) \wedge \\ &(Z_1 \cap \dots \cap Z_k = V) \wedge (V \neq \emptyset) \wedge \forall i \forall j (((i \neq j) \wedge (1 \leq i, j \leq k)) \rightarrow (Z_i \neq Z_j)) \wedge \\ &\forall X \forall Y ((J_{(1,n)}(X \Rightarrow_1 Y) \wedge (V \subset X)) \rightarrow ((W \subseteq Y) \wedge (W \neq \emptyset) \wedge (\bigvee_{i=1}^k (X = Z_i)))) \wedge \\ &(k \geq 2). \end{aligned}$$

Позитивний предикат схожості $M_{a,n}^+(V,W) = kM_{a,n}^+(V,W,k)$ де a - ім'я предиката подібності, а n -число попередніх застосувань правил правдоподібного виведення, що представляє ступінь правдоподібності породжуваних гіпотез з істиннісними оцінками $\langle v,n \rangle$ де $v \in \{1, -1, 0\}$ або з безліччю можливих істиннісних значень (τ, n) .

Аналогічно визначимо негативний предикат схожості, який можна застосовувати до $(-)$ - прикладів:

$$\begin{aligned} M_{a,n}^-(V,W) &\Leftrightarrow \exists k \tilde{M}_{a,n}^-(V,W,k), \text{ где} \\ \tilde{M}_{a,n}^-(V,W,k) &\Leftrightarrow \exists Z_1 \dots \exists Z_k \exists U_1 \dots \exists U_k ((J_{(-1,n)}(Z_1 \Rightarrow_1 U_1) \wedge \dots \wedge J_{(-1,n)}(Z_k \Rightarrow_1 U_k)) \wedge \\ &(Z_1 \cap \dots \cap Z_k = V) \wedge (V \neq \emptyset) \wedge \forall i \forall j (((i \neq j) \wedge (1 \leq i, j \leq k)) \rightarrow (Z_i \neq Z_j)) \wedge \\ &\forall X \forall Y ((J_{(-1,n)}(X \Rightarrow_1 Y) \wedge (W \subseteq Y)) \rightarrow ((V \subset X) \wedge (\bigvee_{i=1}^k (X = Z_i)))) \wedge (k \geq 2). \end{aligned}$$

Сформулюємо формальні уточнення методу подібності Д.С. Мілля, які є правилами правдоподібного (індуктивного) виведення (п.п.в.-1) в ДСМ-методі АПГ:

$$\begin{aligned} (I)^+ &\frac{J_{(\tau,n)}(V \Rightarrow_2 W), M_{a,n}^+(V,W) \wedge \neg M_{a,n}^-(V,W)}{J_{\langle 1,n+1 \rangle}(V \Rightarrow_2 W)}, \\ (I)^- &\frac{J_{(\tau,n)}(V \Rightarrow_2 W), \neg M_{a,n}^+(V,W) \wedge M_{a,n}^-(V,W)}{J_{\langle -1,n+1 \rangle}(V \Rightarrow_2 W)}, \\ (I)^0 &\frac{J_{(\tau,n)}(V \Rightarrow_2 W), M_{a,n}^+(V,W) \wedge M_{a,n}^-(V,W)}{J_{\langle 0,n+1 \rangle}(V \Rightarrow_2 W)}, \\ (I)^\tau &\frac{J_{(\tau,n)}(V \Rightarrow_2 W), \neg M_{a,n}^+(V,W) \wedge \neg M_{a,n}^-(V,W)}{J_{\langle \tau,n+1 \rangle}(V \Rightarrow_2 W)}. \end{aligned}$$

Предикати $M_{a,n}^\sigma(V,W)$, де $\sigma \in \{+, -\}$ будемо називають предикатами простого схожості.

Вище було сказано, що ДСМ-метод АПГ в якості однієї зі своїх складових має ДСМ-міркування [17, 18], які реалізують взаємодію трьох пізнавальних процедур - індукції, аналогії і абдукції. Простий метод подібності є елементарною складовою всіх версій ДСМ-методу, в котором п.п.в.-1 (міллевская індукція) формалізується за допомогою предикатів простої схожості $M_{a,n}^+(V, W)$ і $M_{a,n}^-(V, W)$.

Нижче ми визначимо предикати $\Pi_n^\sigma(V, W)$, де $\sigma \in \{+, -, 0, \tau\}$, що представляють посилки висновків за аналогією за допомогою правил правдоподібного виведення - п.п.в.-2.

$\Pi_n^\sigma(V, W)$, де $\sigma \in \{+, -, 0, \tau\}$ визначаються за допомогою параметричних Предикатів $\Pi_n^\sigma(V, W, k)$, де k-параметр, що виражає число породжених гіпотез, представлених формулами $J_{(n,l)}(X_i \Rightarrow_2 Y_i)$ і $J_{(-1,n)}(X_i \Rightarrow_2 Y_i)$, які є під формулами $\Pi_n^\sigma(V, W, k)$, де $i=1, \dots, k$.

Предикат $\Pi_n^+(V, W, k)$ висловлює умова таке, що об'єкт V містить позитивні причини X, \dots, X_k для множин властивостей Y, \dots, Y_k , відповідно, а безліч властивостей W, що представляє досліджуваний ефект, покривається безліччю Y, \dots, Y_k , де k - параметр.

Ця умова виражається формулою (1):

$$\left(\bigwedge_{i=1}^k \exists X_i (J_{(1,n)}(X_i \Rightarrow_2 Y_i) \wedge (X_i \subseteq V) \wedge (\bigcup_{i=1}^k Y_i = W)) \right) \quad (1)$$

Другою умовою, що міститься в $\Pi_n^+(V, W)$, є умова, яка стверджує, що V не містить ні негативних причин Z, ні Z таких, що $J_{(0,n)}(Z \Rightarrow_2 U)$ для будь-якого непорожньої підмножини U і безлічі W. Ця умова виражається формулою (2):

$$\forall U ((U \subseteq W) \wedge (U \neq \emptyset)) \rightarrow \neg \exists Z (J_{(-1,n)}(Z \Rightarrow_2 U) \vee J_{(0,n)}(Z \Rightarrow_2 U) \wedge (Z \subseteq V)) \quad (2)$$

Визначимо тепер предикат, що виражає процедуру виведення з аналогії для (+) - прикладів з БФ:

$$\Pi_n^+(V, W) \iff \exists k \tilde{\Pi}_n^+(V, W, k) .$$

$\Pi_n^-(V, W)$, визначається аналогічно з заміною в (1) $J_{(1,n)}$ на $J_{(-1,n)}$ і з заміною $J_{(-1,n)}$ на $J_{(1,n)}$.

Таким чином, отримуємо такі визначення $\Pi_n^+(V, W, k)$ і $\Pi_n^+(V, W)$ відповідно.

$$\tilde{\Pi}_n^+(V, W, k) \Leftrightarrow \exists Y_1 \dots \exists Y_k ((\bigwedge_{i=1}^k \exists X_i (J_{(1,n)}(X_i \Rightarrow_2 Y_i) \wedge (X_i \subset V)) \wedge (\bigcup_{i=1}^k Y_i = W)) \wedge \forall U (((U \subseteq W) \wedge (U \neq \emptyset)) \rightarrow \neg \exists Z ((J_{(-1,n)}(Z \Rightarrow_2 U) \vee J_{(0,n)}(Z \Rightarrow_2 U)) \wedge (Z \subseteq V)))$$

Предикати $\Pi_n^0(V, W)$ і $\Pi_n^\tau(V, W)$ визначаються наступним чином:

$$\tilde{\Pi}_n^0(V, W) \Leftrightarrow \exists X_1 \exists Y_1 \exists X_2 \exists Y_2 (J_{(1,n)}(X_1 \Rightarrow_2 Y_1) \wedge J_{(-1,n)}(X_2 \Rightarrow_2 Y_2) \wedge (Y_1 \cap Y_2 \neq \emptyset) \wedge (X_1 \subset V) \wedge (X_2 \subset V) \wedge (Y_1 \subseteq W) \wedge (Y_2 \subseteq W) \vee \exists X \exists Y (J_{(0,n)}(X \Rightarrow_2 Y) \wedge (X \subset V) \wedge (Y \subseteq W)),$$

$$\tilde{\Pi}_n^\tau(V, W) \Leftrightarrow \neg(\Pi_n^+(V, W) \vee \Pi_n^-(V, W) \vee \Pi_n^0(V, W)).$$

Аналогічно п.п.в.-1 формулюються п.п.в.-2 (правила виводу по аналогії):

$$\begin{aligned} (\text{II})^+ & \frac{J_{(\tau,n)}(V \Rightarrow_1 W), \Pi_{a,n}^+(V, W)}{J_{\langle 1, n+1 \rangle}(V \Rightarrow_1 W)}, \\ (\text{II})^- & \frac{J_{(\tau,n)}(V \Rightarrow_1 W), \Pi_{a,n}^-(V, W)}{J_{\langle -1, n+1 \rangle}(V \Rightarrow_1 W)}, \\ (\text{II})^0 & \frac{J_{(\tau,n)}(V \Rightarrow_1 W), \Pi_{a,n}^0(V, W)}{J_{\langle 0, n+1 \rangle}(V \Rightarrow_1 W)}, \\ (\text{II})^\tau & \frac{J_{(\tau,n)}(V \Rightarrow_1 W), \Pi_{a,n}^\tau(V, W)}{J_{\langle \tau, n+1 \rangle}(V \Rightarrow_1 W)}. \end{aligned}$$

Результатом застосування п.п.в.-2 є гіпотези про наявність досліджуваного ефекту W у відповідних об'єктів V , щодо яких була проведена оцінка «невизначено».

Закон Д.С. Міллем одноманітності в природі (умова міллеровської індукції при вилученні з масивів явищ істотних причин), звичайно, є лише філософською ідеєю, яка не може бути достатньою підставою індуктивних висновків. Але для коректного формулювання достатності підстав для прийняття результатів

індуктивних висновків потрібно формулювати додаткові умови, такі, що здійсненність їх робить висновок обґрунтованим, а нездійсненність є формалізацією його фальсифікації відповідно до критерію демаркації К.Р. Поппера [21].

В ДСМ-методі АПГ таким достатньою підставою ДСМрассужденія, що включає як індукцію, так і аналогію, є аксіоми каузальної повноти $АКП^\sigma$, де $\sigma \in \{+, -\}$. За допомогою $АКП^\sigma$ формалізується абдукція Ч.С. Пірса [23] - прийняття породжуваних гіпотез за допомогою пояснення початкових даних (в ДСМ-методі АПГ ними є БФ).

Використання $АКП^\sigma$ [22] є ще однією підставою для характеристики індукції в ДСМ-методі АПГ (тобто п.п.в.-1) як контекстнозавісімою [18, 22].

Введення і використання $АКП^\sigma$ передбачає:

1. Характеризацію предметної області (універсуму моделей), що містить (+) - факти і (-) - факти, а також позитивні і негативні причинно-наслідкові залежності;
2. Завдання відкритої теорії - квазіаксіоматичної теорії (КАТ) [17,18,20].

КАТ є трійка $K = \langle \Sigma, \Sigma', R \rangle$ де Σ _ відкрите безліч аксіом, лише частково характеризують предметну область, Σ' - відкрите безліч фактів і гіпотез, а R - безліч правдоподібних і достовірних правил виведення (наприклад, п.п .в.-1, п.п.в.-2 і правил дедуктивного виведення).

Послідовність станів $K_0, K_1 \dots K_n$ квазіаксіоматичної теорії K , така, що K_0 - початковий стан, K_n - заключний стан, так що застосування правил правдоподібного виведення до K_n породжує K_{n+1} , причому $K_{n+1} = K_n$, називається р-міркуванням. Застосування правил достовірного висновку з R до результатів р міркування називається рR – міркуванням.

Наведемо нижче формулювання $АКП^+$ і $АКП^-$, сенс яких полягає в тому, що наявність кожного ефекту і його відсутність виникають (+) - і (-) - причинами відповідно:

$$\text{АКП}^{(+)} \quad \forall X \forall Y \exists k \exists V_1 \dots \exists V_k \exists W_1 \dots \exists W_k (J_{\langle 1,0 \rangle} (X \Rightarrow_1 Y) \rightarrow \exists n (\bigwedge_{i=1}^k J_{\langle 1,n \rangle} (V_i \Rightarrow_2 W_i) \wedge (V_i \subset X) \wedge (V_i \neq \emptyset) \wedge (W_i \neq \emptyset)) \wedge (\bigcup_{i=1}^k W_i = Y))),$$

$$\text{АКП}^{(-)} \quad \forall X \forall Y \exists k \exists V_1 \dots \exists V_k \exists W_1 \dots \exists W_k (J_{\langle -1,0 \rangle} (X \Rightarrow_1 Y) \rightarrow \exists n (\bigwedge_{i=1}^k J_{\langle -1,n \rangle} (V_i \Rightarrow_2 W_i) \wedge (V_i \subset X) \wedge (V_i \neq \emptyset) \wedge (W_i \neq \emptyset)) \wedge (\bigcup_{i=1}^k W_i = Y))).$$

Охарактеризуємо тепер структуру ДСМ міркування [20].

Кроком ДСМ-міркування будемо називати одноразове застосування п.п.в.-1 (індукції) або п.п.в.-2 (аналогії).

Тактом ДСМ-міркування будемо називати впорядковане послідовне застосування п.п.в.-1 і п.п.в.-2.

Етапом I ДСМ-міркування будемо називати послідовне застосування тактів таке, що безліч породжених гіпотез на такті p збігається з безліччю гіпотез, породжених на такті $p + 1$, де p - номер першого такого збігу [14]. Цей такт з номером p назвемо тактом стабілізації Етапу I ДСМ-міркування.

Етапом II ДСМ-міркування будемо називати перевірку здійсненності АКП $^\sigma$ для встановлення абдуктивної збіжності або розбіжності процесу ДСМ-міркування, результатом яких є прийняття чи неприйняття породжуваних гіпотез.

Відзначимо особливості формалізації індуктивного методу подібності Д.С. Мілля як початкової компоненти ДСМ-міркування:

1. Першим правилом для міллевського методу подібності відповідають чотири правила ДСМ-міркування I^σ , де $\sigma \in \{+, -, 0, \tau\}$, які визначаються за допомогою предикатів позитивного і негативного подібності $\Pi_{a,n}^+(V, W)$ і $\Pi_{a,n}^-(V, W)$ відповідно. Ці предикати утворюють мінімальну (по виразну силу) версію ДСМ-методу АПГ. Ця мінімальна версія посилюється за допомогою додавання до неї додаткових умов [20].
2. Формалізація індуктивного методу подібності здійснюється за допомогою його взаємодії з висновками по аналогії (п.п.в.-2) і абдуктивним прийняттям гіпотез в процесі ДСМ-міркування (Етапи I і II).

3. Достатньою умовою прийняття гіпотез є аксіоми каузальної повноти $АКП^\sigma$, де $\sigma \in \{+, -\}$. За допомогою цих аксіом реалізується непряма, контекстно-залежна індукція на достатньому підставі.
4. В силу того, що ДСМ-роздум реалізується на Етапі II як процес поповнення БФ під управлінням $АКП^\sigma$, є два рівня індуктивних процедур - породження гіпотез про (\pm) -причини допомогою п.п.в.-1 в складі ітерацій Етапу I і повторення Етапу I після розширення БФ в процесі ДСМ-міркувань на Етапі II.

Б.2 Алгоритм пошуку глобальних подібностей: основні визначення та умовні позначення

Ключовим елементом алгоритмики ДСМ-методу є механізм породження емпіричних залежностей («гіпотез-кандидатів»). Дані механізми, беручи до уваги використовувану структуру правил правдоподібного виведення ДСМ-методу, вимагають знаходження подібності описів об'єктів.

Подібність описів розглянутих об'єктів (див. Докладніше, п. 5.1) може виражатися за допомогою теоретико множинного перетину описів об'єктів.

Нижче наведені основні умовні позначення і визначення, що використовуються в даному розділі:

S - кінцева множина об'єктів, для яких визначена операція подібності як перетин;

O - конечное множество всех образующих, входящих в состав объектов;

$Y = \{X_1 \dots X_k\}$ - кінцева довільна множина об'єктів, для яких виконана така умова $(X | X \in Y, X \in S), Y+S$.

H - множина утворюючих $h \in O$.

$(Y)'$ - множина утворюючих, що відповідає перетину (результату обчислення використовуваної тут операції подібності) всіх об'єктів множини Y : $((Y)' = (\{X_1 \dots X_k\})' = X_1 \cap \dots \cap X_k$, де $X_1 \dots X_k \in Y$).

$(h)'$ - множина всіх об'єктів, що мають в своєму складі множину утворюючих h : $((h)' = (X|X \in S, X \square h))$;

$(Y)''$ - безліч об'єктів, відповідних замиканню множини Y : $((Y)'' = (\{X \dots X_k\})'' = \{X|X \in S, X \square X_1 \cap \dots \cap X_k\}, \text{де } X_1 \dots X_k \in Y)$;

Визначення 1.

Пару множин $\langle Y, h \rangle$, де $h = (Y)'$ будемо називати локальною схожістю, якщо $|h'| < |Y|$.

Визначення 2.

Пару множин $\langle Y, h \rangle$, де $h = (Y)'$ будемо називати глобальною схожістю, якщо $|h'| < |Y|$.

Визначення 3.

Для пари множин $\langle Y, h \rangle$, що є локальною або глобальною схожістю, безліч Y будемо називати орбітою.

Визначення 4.

Для локального схожості $\langle Y, h \rangle$, ДСМ - замиканням $\langle Y, h \rangle''$ є безліч об'єктів X з S , для яких виконується така умова $X \ni h$. Дерево Всіх Перетинань - дерево, вершинами якого є всі можливі локальні і глобальні подібності щодо багатьох об'єктів S .

Б.2 Аналіз двох базових моделей породження гіпотез

Можливі, принаймні, дві базові моделі породження гіпотез - відповідно до рисунку В.1 і В.2:

Стратегия M1 :



Реализует следующую последовательность действий:

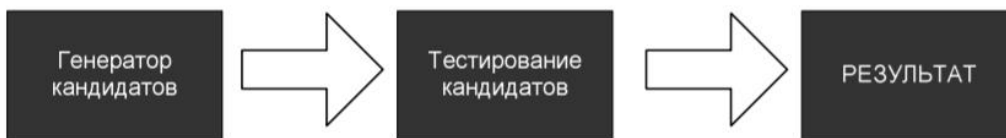


Рисунок Б.1 - Стратегия M1

Стратегия M2 :



Реализует следующую последовательность действий:

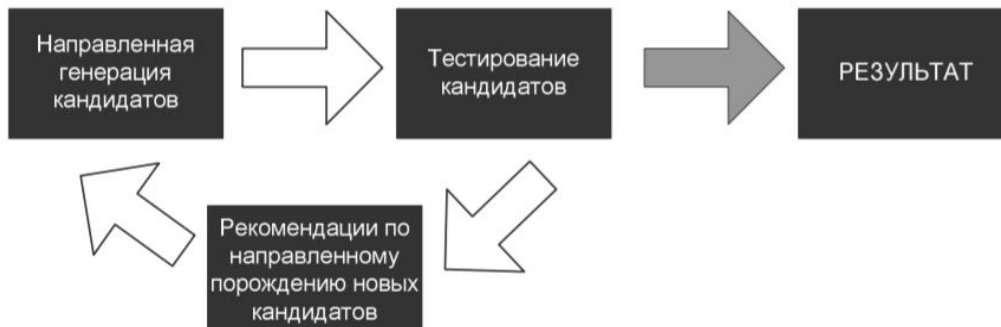


Рисунок Б.1 - Стратегия M2

Кожна із цих моделей, в свою чергу, реалізує обхід вершин дерева всіх перетинань з метою пошуку тих вершин, у яких зіставлена їм локальна схожість є також глобальною схожістю. Саме вони породжують ті об'єкти, які можуть служити вихідним матеріалом для перевірки інших умов, які використовуються при формулюванні вирішальних предикатів ДСМ - методу. Перша стратегія здійснює повний перебір всіх можливих непустих вершин дерева всіх перетинань. В рамках другої стратегії пошук подібностей має спрямований характер за рахунок аналізу вже породжених глобальних подібностей, що може служити причиною скорочення кількості породжених вершин.

Б.3 Аналіз структурних особливостей алгоритмів пошуку глобальних подібностей

Б.3.1 Результати аналізу алгоритмів пошуку глобальних перетинів

На основі проведеного аналізу алгоритмів пошуку глобальних подібностей, що реалізують стратегію 2 породження гіпотез, на задоволення вимог, що пред'являються до алгоритму пошуку глобальних подібностей, можна зробити висновок, що жоден з розглянутих алгоритмів не відповідає висунутим вимогам.

Алгоритм П-3-bt не задовольняє одному з основних вимог, що пред'являються, а саме не здійснює рішення задачі обмеження кількості стрибків у вершину, відповідну глобальній подібності, до одного.

Для обох алгоритмів характерні підвищені вимоги до обсягу необхідної пам'яті і додаткові обчислювальні витрати при реалізації обходу вершин з пропущеними під час стрибка об'єктами. Також в алгоритмі «замикай по одному» виникають додаткові обчислювальні витрати при вирішенні задачі обмеження кількості стрибків у вершину, відповідну глобальній подібності.

Таким чином, неможливість задоволення всіх поставлених вимог жодним з існуючих алгоритмів пошуку глобальних подібностей, вимагає розробку нового алгоритму, в основу якого може бути покладено алгоритм «замикай по одному» яка задовольняє основним вимогам, що пред'являються.

Основними завданнями модифікації алгоритму «замикай по одному» є наступні завдання:

- Стратегія проходу вершин з пропущеними під час стрибка об'єктами не повинна вимагати додаткових обчислювальних витрат і додаткових обсягів пам'яті.
- Стратегія обмеження стрибків у вершину, відповідну глобальній подібності, до одного не повинна вимагати проходу списків, зокрема перебору індексів об'єктів в знайденому глобальному схожості.

Б.3.2 Опис функціонування розробленого програмного засобу

У цьому розділі містяться сценарії використання розробленої інформаційної системи. Основні сценарії використання: «Читання, видалення, додавання і редагування вихідних даних», «Класифікація нового об'єкта». Опис даних сценаріїв представлено в пунктах В.2.1 та В.2.2 відповідно.

Б.3.2.1 Сценарій використання 1: «Читання, видалення, додавання і редагування даних»

Розроблена ІС дозволяє користувачеві формувати і змінювати вміст БД, тобто додавати, видаляти і редагувати як вже класифіковані проекти (тобто проекти, клас тривалості виробництва виробів з яким відомий), так і нові, некласифіковані проекти.

Для додавання класифікованого проекту користувачеві необхідно вибрати комірку, відповідну колонку бажаного класу в таблиці проектів і вибрати в контекстному меню команду «Додати».

Для додавання нового некваліфікованого проекту користувачеві необхідно вибрати комірку в таблиці проектів, відповідну колонку «Не класифіковані», і вибрати в контекстному меню команду «Додати».

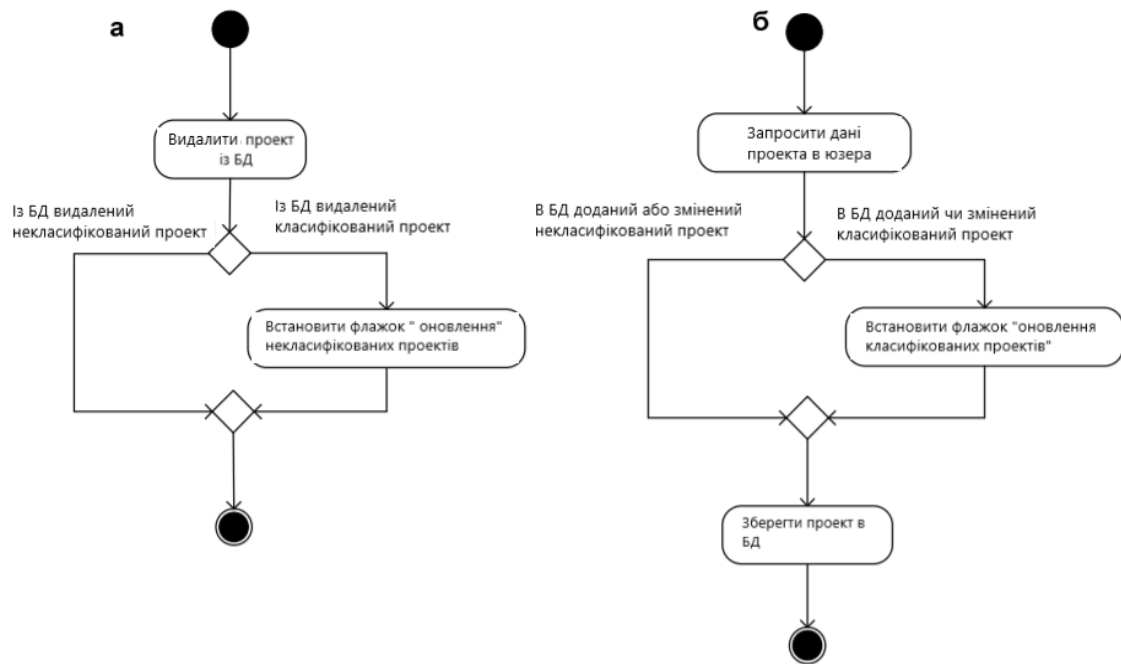


Рисунок В.4 - Діаграми дій в нотатції UML 2.0, відповідні командам а) видалити і б) додати або редагувати

При редагуванні або додаванні проекту (відповідно до малюнком В.4 а)) здійснюється запит даних проекту у користувача за допомогою діалогу «Редагування проекту» (відповідно до малюнком В.5). У разі якщо в результаті введення даних користувачем змінюється склад класифікованої проекту або додається новий класифікований проект, даний факт реєструється установкою прапора «оновлення класифікованих проектів», який враховується при класифікації нового проекту, визначаючи необхідність повторення кроків ДСМ-методу перед безпосередньою класифікацією проекту.

Додавання і модифікація нового некласифікованого проекту не супроводжується установкою прапора «оновлення класифікованих проектів» (відповідно до малюнком В.4 б)).

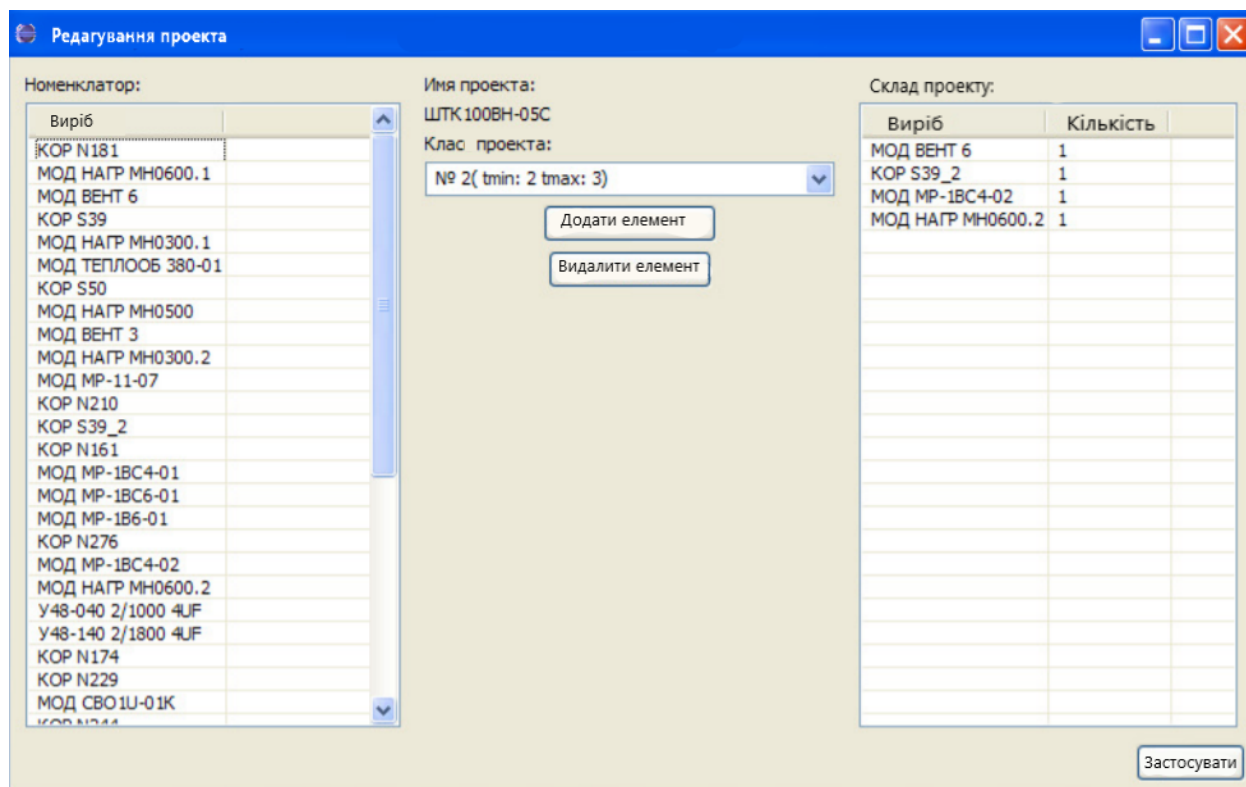


Рисунок Б.5 - Вікно редагування проекту графічного інтерфейсу ІС

Б.3.2.2 Сценарій використання 2: «Класифікація нового проекту»

Для класифікації нового проекту необхідно вибрати даний об'єкт в стовпці «Не класифіковані» і здійснити виклик команди «Впорядкувати новий проект» за допомогою натискання відповідної кнопки (відповідно до рисунку В.6). Діаграма дій, відповідних даної команди, наведена на рисунку В.7. Спочатку здійснюється перевірка прапора «Оновлення класифікованих проектів». Установка даного прапора говорить про те, що база даних класифікованих проектів була оновлена.

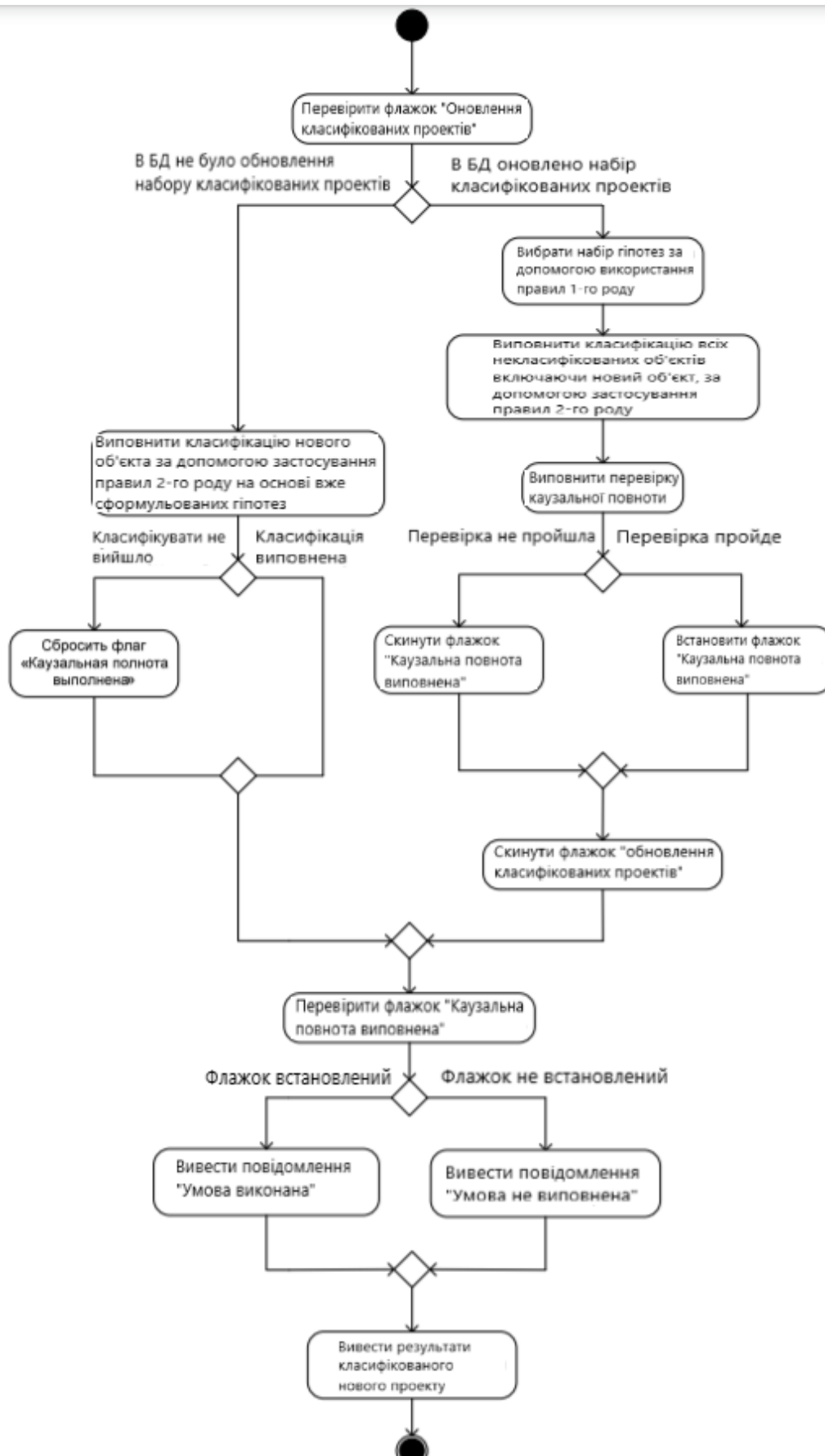


Рисунок Б.7 - Діаграма діяльності в нотації UML 2.0, що відповідає команді «Впорядкувати новий проект»

Для обліку даних змін необхідно послідовно виконати кроки ДСМ - методу, а саме оновити базу гіпотез за допомогою застосування правил 1-ого роду, класифікувати всі некласифіковані об'єкти, включаючи новий, і виконати перевірку умови каузальної повноти. Якщо ж прапор «Оновлення класифікованих проектів» не встановлено, а отже, ніяких змін в базі класифікованих проектів не було виконано, то необхідно здійснити класифікацію нового проекту. Після виконання даної класифікації необхідно оновити дані про здійсненності умови каузальної повноти. Так, якщо об'єкт був класифікований невірно, необхідно скинути прапор «Каузальна повнота». Далі після виконання класифікації та перевірки здійсненності умови каузальної повноти необхідно вивести для користувача наступну інформацію:

- повідомлення про здійсненності умови каузальної повноти (крім того, проекти, класифіковані вірно за допомогою правила 2-ого роду, будуть виділені синім кольором) (відповідно до малюнку В.8);
- результати класифікації нового проекту (відповідно до рисунку В.9).

Далі в залежності від переваги користувача в разі не здійсненності умови каузальної повноти база даних може бути довізначити новими проектами (відповідно до малюнком В.7) і процес класифікації запущений заново. Істинний клас проекту, який може відрізнитися від прогнозованого, встановлюється після виробництва виробу за проектом за допомогою редагування проекту. Тільки після установки істинного класу проекту і, як наслідок, переміщення проекту з шпальти «Не класифіковані» в стовпець, відповідний істинному класу, даний проект може бути використаний при знаходженні гіпотез.

Клас 1	Клас 2	Клас 3	Клас 4	Клас 5	Не класифіковані
1.1 ШТК102ВН-03С: (1,2,3): 1	2.1 ШТК104ВН-01С: (2,3,13): 2	3.1 ШТК102ВН-04С: (1,2,3,20): 3	4.1 ШТК102ВН-06С: (1,2,3,33,34): 5	5.1 ШТК103ВН-03С: (2,3,22,33,34): 5	ШТК100ВН-22А: (3,5,7) клас: 1
1.2 ШТК100ВН-01С: (3,4,5): 1	2.2 ШТК104ВН-02С: (2,3,14): 2	3.2 ШТК102ВН-05С: (1,2,6,21): 3	4.2 ШТК102ВН-03С: (2,6,33,34,40): 5	5.2 ШТК103ВН-02С: (5,33,34,29): 5	ШТК100ТН-21С: (4,6,8) клас: 1
1.3 ШТК102ВН-01С: (1,2,6): 1	2.3 ШТК100: (1,2,6,16): 2	3.3 ШТК103ВН-01С: (2,3,22): 3	4.3 ШТК102ВН-02С: (2,3,22,24): 4	5.3 ШТК104ТН-01С: (2,33,34,32,30,35): 5	ШТК102ТН-24С: (1,6,8,19) клас: 2
1.5 ШТК100ВН-02А: (7,8,9): 1	2.7 ШТК104ВН-03С: (2,18,19): 2	3.4 ШТК102ТН-01С: (2,6,23): 3	4.4 ШТК103ВН-01С: (2,3,24,29): 4	5.4 ШТК103К-01С: (33,34,29,36): 5	ШТК104ВН-23С: (8,16,18) клас: 2
1.6 ШТК100ВН-03С: (9,11,10): 1	2.8 ШТК100ВН-09С: (3,10,19,28): 2	3.5 ШТК102ВН-05С: (2,7,9,24): 4	4.5 ШТК104ВН-05С: (2,3,20,25): 4	5.5 ШТК102ТН-02С: (34,32,35,37,39,38): 5	ШТК104ВН-26С: (2,9,15,17,25) клас: 3
1.7 ШТК100ВН-04С: (3,5,10): 1	2.9 ШТК104ВН-11С: (5,9,13): 2	3.7 ШТК104ВН-05С: (2,3,16,17,25): 3	4.6 ШТК104ВН-06С: (2,9,20,25): 4	5.7 ШТК104ВН-07С: (2,3,13,33,34,36): 5	ШТК102ВН-24С: (1,8,9,20) клас: 3
1.8 ШТК102ВН-10С: (1,5,9): 1	2.10 ШТК100-1: (4,5,6,16): 2	3.8 ШТК104ВН-06С: (5,9,17,18,19): 3	4.7 ШТК102ВН-07С: (2,3,14,24,31): 4	5.8 ШТК102ТН-05С: (2,19,33,34,40,32): 5	ШТК102ТН-22С: (2,21,34,29,32,35) клас: 4
1.9 ШТК102ВН-11С: (2,3,4): 1	2.13 ШТК104ВН-12С: (5,9,14): 2	3.9 ШТК104ВН-04С: (2,25,26,27): 3	4.10 ШТК103ВН-12С: (3,5,22,24): 4	5.9 ШТК103ВН-11С: (2,33,34,29): 5	ШТК103ВН-21С: (23,24,37) клас: 4
1.10 ШТК100ВН-11С: (1,3,5): 1	2.14 ШТК104ВН-13С: (5,6,14): 2	3.10 ШТК103ВН-11С: (5,9,23): 3	4.12 ШТК102ВН-14С: (5,9,14,24,31): 4	5.10 ШТК102ТН-12С: (5,7,19,33,34,32): 5	ШТК104ВН-25С: (20,30,36,37) клас: 3
1.4 ШТК100ТН-01С: (1,4,5,6): 1	2.15 ШТК104ВН-14С: (5,18,19): 2	3.11 ШТК102ВН-12С: (4,5,9,20): 3	4.13 ШТК102ТН-15С: (1,2,3,34,38): 4	5.11 ШТК104ТН-13С: (2,5,18,32,35,38): 5	ШТК104К-21С: (14,15,17,33) клас: 3
1.11 ШТК100ВН-13С: (2,3,7): 1	2.16 ШТК100ВН-15С: (2,7,9,19): 2	3.12 ШТК104ВН-13С: (5,9,16,17,18): 3	4.14 ШТК102ТН-16С: (5,7,9,34,38): 4	5.12 ШТК104ВН-14С: (5,9,14,33,34,36): 5	
1.12 ШТК102ВН-14С: (1,2,9): 1	2.17 ШТК104ВН-16С: (2,3,18): 2	3.13 ШТК104ВН-14С: (2,9,17,18,19): 3	4.15 ШТК103ВН-17С: (3,5,24,29): 4	5.13 ШТК103ВН-15С: (5,9,22,33,34): 5	
1.13 ШТК102ТН-15С: (2,4,6): 1	2.18 ШТК104ВН-17С: (2,3,25): 3	3.14 ШТК102ВН-15С: (2,4,6,24): 4			
1.14 ШТК102ВН-16С: (4,5,9): 1		3.16 ШТК102ТН-16С: (6,11,10,21): 3			
1.15 ШТК100ТН-17С: (1,5,6): 1		3.15 ШТК103ВН-17С: (5,9,22): 3			
1.16 ШТК102ВН-18С: (2,4,9): 1					
1.17 ШТК102ВН-19С: (5,7,9): 1					

Фрагмент для кожного класу:

Клас 1	Клас 2	Клас 3	Клас 4	Клас 5
(-):(13): {2,1,2,9,5,7}	(-):(20): {3,1,3,1,4,5,4,6}	(-):(23,34): {4,1,4,2,5,1,5,3,5,7,5,8...}	(-):(33,34): {5,1,5,2,5,3,5,4,5,7,5,8, ...}	(+):(33,34): {5,1,5,2,5,3,5,4,5,7,5,8, ...}
(-):(14): {2,2,2,1,2,1,4,7,4,1,5,1}	(-):(6,21): {3,2,3,1}	(-):(2,34): {4,1,4,2,4,1,5,1,5,3,5,7,5,8...}	(-):(2,3,33,34): {5,1,5,7}	(+):(2,3,33,34): {5,1,5,7}
(-):(6,16): {2,4,2,1}	(-):(22): {3,3,3,1,4,3,4,1,5,1,5,1}	(-):(34): {4,1,4,2,4,1,4,1,5,1,5,2,5,3...}	(-):(22,33,34): {5,1,5,1}	(+):(22,33,34): {5,1,5,1}
(-):(19): {2,7,2,8,2,1,2,1,3,8,3,1,5,8,...}	(-):(23): {3,4,3,1}	(-):(33,34): {4,1,4,2,5,1,5,2,5,3,5,4, ...}	(-):(33,34,29): {5,2,5,4,5,9}	(+):(33,34,29): {5,2,5,4,5,9}
(-):(18,19): {2,7,2,1,3,8,3,1}	(-):(2,24): {3,5,3,1,4,3,4,4,4,7}	(-):(1,2,3,34): {4,1,4,1}	(-):(5,33,34): {5,2,5,1,5,1,5,1}	(+):(5,33,34): {5,2,5,1,5,1,5,1}
(-):(18): {2,7,2,1,2,1,3,8,3,1,3,1,5,1}	(-):(24): {3,5,3,1,4,3,4,4,4,7,4,1,4,1...}	(-):(2,3,33,34): {4,1,5,1,5,7}	(-):(34,32,35): {5,3,5,5}	(+):(34,32,35): {5,3,5,5}
(-):(2,19): {2,7,2,1,3,1,5,8}	(-):(17): {3,7,3,8,3,1,3,1}	(-):(2,3,24): {4,3,4,4,4,7}	(-):(34,32): {5,3,5,5,5,8,5,1}	(+):(34,32): {5,3,5,5,5,8,5,1}
(-):(2,18): {2,7,2,1,3,1}	(-):(2,25): {3,7,3,9,4,5,4,6}	(-):(3,24): {4,3,4,4,4,7,4,1,4,1}	(-):(32): {5,3,5,5,5,8,5,1,5,1}	(+):(32): {5,3,5,5,5,8,5,1,5,1}
(-):(5,14): {2,1,2,1,4,1,5,1}	(-):(16,17): {3,7,3,1}	(-):(24): {4,3,4,4,4,7,4,1,4,1,4,1}	(-):(32,35): {5,3,5,5,5,1}	(+):(32,35): {5,3,5,5,5,1}
(-):(2,25): {2,1,3,7,3,9,4,5,4,6}	(-):(2,17): {3,7,3,1}	(-):(3,22,24): {4,3,4,1}	(-):(2,33,34,32): {5,3,5,8}	(+):(2,33,34,32): {5,3,5,8}
(-):(20): {3,1,3,1,4,5,4,6}	(-):(5,9,17,18): {3,8,3,1}	(-):(2,24,29): {4,4,4,1}	(-):(33,34,32): {5,3,5,8,5,1}	(+):(33,34,32): {5,3,5,8,5,1}
(-):(6,21): {3,2,3,1}	(-):(9,17,18): {3,8,3,1,3,1}	(-):(20,25): {4,5,4,6}	(-):(33,34,36): {5,4,5,7,5,1}	(+):(33,34,36): {5,4,5,7,5,1}
(-):(22): {3,3,3,1,4,3,4,1,5,1,5,1}	(-):(9,17,18,19): {3,8,3,1,1}	(-):(14,24,31): {4,7,4,1}	(-):(32,35,38): {5,5,5,1}	(+):(32,35,38): {5,5,5,1}
(-):(23): {3,4,3,1}	(-):(23,34): {4,1,4,2,5,1,5,3,5,7,5,8...}	(-):(5,24): {4,1,4,1,4,1}	(-):(19,33,34,32): {5,8,5,1}	(+):(19,33,34,32): {5,8,5,1}
(-):(2,24): {3,5,3,1,4,3,4,4,4,7}	(-):(34): {4,1,4,2,4,1,5,1,5,2,5,3...}	(-):(3,24): {4,1,4,1,4,1}	(-):(5,32): {5,1,5,1}	(+):(5,32): {5,1,5,1}
(-):(24): {3,5,3,1,4,3,4,4,4,7,4,1,4,1...}	(-):(34): {4,1,4,2,4,1,4,1,5,1,5,2,5,3...}	(-):(34,38): {4,1,4,1,5,9}	(-):(5,9,33,34): {5,1,5,1}	(+):(5,9,33,34): {5,1,5,1}
(-):(17): {3,7,3,8,3,1,3,1}	(-):(33,34): {4,1,4,2,5,1,5,2,5,3,5,4, ...}	(-):(22,33,34): {5,1,5,1}	(+):(2,34): {4,1,4,2,4,1}	(+):(2,34): {4,1,4,2,4,1}
(-):(16,17): {3,7,3,1}	(-):(1,2,3,34): {4,1,4,1}	(-):(33,34,29): {5,2,5,4,5,9}	(+):(1,2,3,34): {4,1,4,1}	(+):(1,2,3,34): {4,1,4,1}
(-):(2,17): {3,7,3,1}	(-):(2,33,34): {4,1,5,1,5,7}	(-):(5,33,34): {5,2,5,1,5,1,5,1}	(+):(2,3,24): {4,3,4,4,4,7}	(+):(2,3,24): {4,3,4,4,4,7}
(-):(5,9,17,18): {3,8,3,1}	(-):(2,3,24): {4,3,4,4,4,7,4,1,4,1}	(-):(34,32,35): {5,3,5,5}	(+):(3,24): {4,3,4,4,4,7,4,1,4,1}	(+):(3,24): {4,3,4,4,4,7,4,1,4,1}
(-):(9,17,18): {3,8,3,1,3,1}	(-):(3,24): {4,3,4,4,4,7,4,1,4,1}	(-):(34,32): {5,3,5,5,5,8,5,1}	(+):(24): {4,3,4,4,4,7,4,1,4,1,4,1}	(+):(24): {4,3,4,4,4,7,4,1,4,1,4,1}
(-):(9,17,18,19): {3,8,3,1,1}	(-):(3,24): {4,3,4,4,4,7,4,1,4,1}	(-):(32): {5,3,5,5,5,8,5,1,5,1}	(+):(3,22,24): {4,3,4,1}	(+):(3,22,24): {4,3,4,1}
(-):(2,33,34): {4,1,4,2,5,1,5,3,5,7,5,8...}	(-):(3,24,29): {4,4,4,1}	(-):(32,35): {5,3,5,5,5,1}	(+):(3,24,29): {4,4,4,1}	(+):(3,24,29): {4,4,4,1}
(-):(2,34): {4,1,4,2,4,1,5,1,5,3,5,7,5...}	(-):(2,20,25): {4,5,4,6}	(-):(2,33,34,32): {5,3,5,8}	(+):(2,20,25): {4,5,4,6}	(+):(2,20,25): {4,5,4,6}
(-):(34): {4,1,4,2,4,1,4,1,5,1,5,2,5,3...}	(-):(14,24,31): {4,7,4,1}	(+):(14,24,31): {4,7,4,1}	(+):(33,34,36): {5,4,5,7,5,1}	(+):(14,24,31): {4,7,4,1}
(-):(33,34): {4,1,4,2,5,1,5,2,5,3,5,4...}	(-):(5,24): {4,1,4,1,4,1}	(-):(33,34,36): {5,4,5,7,5,1}	(+):(5,24): {4,1,4,1,4,1}	(+):(5,24): {4,1,4,1,4,1}
(-):(1,2,3,34): {4,1,4,1}	(-):(3,5,24): {4,1,4,1}	(-):(32,35,38): {5,5,5,1}	(+):(3,5,24): {4,1,4,1}	(+):(3,5,24): {4,1,4,1}
(-):(2,3,33,34): {4,1,5,1,5,7}	(-):(34,38): {4,1,4,1,5,5}	(-):(19,33,34,32): {5,8,5,1}	(+):(34,38): {4,1,4,1}	(+):(34,38): {4,1,4,1}

Рисунок Б.8 - Результати перевірки здійсненності умови каузальної повноти

Класифікація проекту	
Шкаф (3.1 ШТК102ВН-04С: (1,2,3,20))	
Підходящі гіпотези	Клас
(+):(1): {1,1,1,3,1,8,1,1,1,1,1}	1
(+):(1,3): {1,1,1,1}	1
(+):(1,2): {1,1,1,3,1,1}	1
(+):(3): {1,1,1,2,1,7,1,9,1,1,1,1}	1
(+):(2): {1,1,1,3,1,9,1,1,1,1,1,1,1}	1
(+):(2,3): {1,1,1,9,1,1}	1
(-):(20): {3,1,3,1,4,5,4,6}	1
(+):(2,3): {2,1,2,2,2,1,2,1}	2
(+):(2): {2,1,2,2,2,4,2,7,2,1,2,1,2,1}	2
(+):(3): {2,1,2,2,2,8,2,1,2,1}	2
(+):(1,2): {1,1,1,3,1,1,3,1,3,2,4,1,4,1}	2
(+):(1): {1,1,1,3,1,8,1,1,1,1,1,1,3,1,3,2,4,1,4,1}	2
(+):(1,3): {1,1,1,1,3,1,4,1,4,1}	2
(-):(20): {3,1,3,1,4,5,4,6}	2
(+):(1): {3,1,3,1}	3
(+):(1,2): {3,1,3,2}	3
(+):(2): {3,1,3,2,3,3,3,4,3,5,3,7,3,9,3,1,3,1}	3
(+):(2,3): {3,1,3,3,3,7}	3
(+):(3): {2,1,2,2,2,8,2,1,2,1,1,1,1,2,1,7,1,9,1,1,1,1,4,1,4,3,4,4,4,5,4,7,4,1,4,1,4,1,5,1,5,7,5,1}	3
(+):(1): {2,4,1,1,1,3,1,8,1,1,1,1,1,1,4,1,4,1}	3
(+):(1,3): {1,1,1,1,4,1,4,1}	3
(+):(2): {4,1,4,2,4,3,4,4,4,5,4,6,4,7,4,1}	4
(+):(2,3): {4,1,4,3,4,4,4,5,4,7,4,1}	4
(+):(3): {4,1,4,3,4,4,4,5,4,7,4,1,4,1,4,1}	4
(+):(1,2): {3,1,3,2,2,4,1,1,1,3,1,1}	4
(+):(1): {3,1,3,2,2,4,1,1,1,3,1,8,1,1,1,1,1,1,1}	4
(+):(20): {3,1,3,1}	4
(+):(1,3): {3,1,1,1,1,1}	4
(+):(3): {5,1,5,7,5,1}	5
(+):(2): {4,1,4,2,4,3,4,4,4,5,4,6,4,7,4,1,3,1,3,2,3,3,3,4,3,5,3,7,3,9,3,1,3,1,2,1,2,2,2,4,2,7,2,1,2,1,2,1,1,1,1,3,1,9,1,1,1,1,1,1,1}	5
(+):(2,3): {4,1,4,3,4,4,4,5,4,7,4,1,3,1,3,3,3,7,2,1,2,2,2,1,2,1,1,1,1,9,1,1}	5
(+):(1,2): {4,1,4,1,3,1,3,2,2,4,1,1,1,3,1,1}	5
(+):(1): {4,1,4,1,3,1,3,2,2,4,1,1,1,3,1,8,1,1,1,1,1,1,1}	5
(+):(1,3): {4,1,4,1,3,1,1,1,1,1}	5
(+):(20): {4,5,4,6,3,1,3,1}	5
Класифікуючі гіпотези	Клас
(+):(1): {3,1,3,1}	3

Клас проекту: 3

Рисунок Б.9 - Результати класифікації нового проекту

Б.4 Визначення вирішальних предикатів і правил правдоподібного виведення

У Додатку В.1 була приведена формалізація засобами спеціальних багатозначних логік методу подібності, що лежить в основі розглянутого в даній роботі ДСМ - методу АПГ. Пропонована формалізація включала в себе:

- 1) опис предикатів простої схожості $\Pi_{a,n}^+(V, W)$ і $\Pi_{a,n}^-(V, W)$ за допомогою яких формулюються п.п.в - 1, що є формальними уточненнями (аналогами) методу подібності Д.С. Мілля;
- 2) предикатів $\Pi_n^\sigma(V, W)$, де $\sigma \in \{+, -, 0, \tau\}$, що представляють висновки за аналогією за допомогою правил правдоподібного виведення - п.п.в.-2.

Можливість застосування результатів представленої (стандартної) формалізації без внесення модифікацій обмеження вузьким колом завдань. Специфіка поставленого завдання, а саме наявність кількох класів тривалості, яскраво вираженої тенденції ускладнення структури проекту від класу класу, вимагає модифікації стандартних визначень правил першого і другого роду, а також предикатів, що представляють посилки висновків за аналогією.

В даному розділі буде представлено два варіанти модифікації стандартних визначень гіпотез і правил виведення: перший варіант максимально наближений до стандартної формалізації ДСМ-методу, а другий являє собою його модифікацію відповідно до особливостей поставленої прикладної задачі. Визначення основних понять і відносин, які використовуються в процесі застосування ДСМ-методу для розглянутих нижче варіантів ідентичні і відповідають визначенням, представленим в пункті 5.2. Варіанти відрізняються способом визначення гіпотез (тобто правилами правдоподібного виведення 1-ого роду) і правилами класифікації (тобто правилами правдоподібного виведення 2-ого роду). Формальний опис предикатів і правил виведення, відповідних гіпотез і правилам виведення найкращого (другого) варіанту, представлено в пункті 5.3.

Б.4.1 Аналіз різних способів визначення вирішальних предикатів і правил правдоподібного виведення і вибір найкращого варіанта

Б.4.1.1 Опис першого варіанту модифікації

Для даного варіанту модифікації характерний наступний спосіб визначення (+) - і (-) - прикладів:

- (+) - прикладами певного класу тривалості є всі проекти, які виконані в інтервал часу, який відповідає цьому класу тривалості.
- (-) - прикладами певного класу тривалості є всі проекти, що належать до всіх інших класів тривалості. Для даного варіанту модифікації характерний наступний спосіб визначення (+) -,

(-) -, (0) - гіпотез фіксованого класу тривалості:

(+) - гіпотеза - схожість, яка є серед подібностей (+) - прикладів і відсутній серед подібностей (-) прикладів;

(+) - гіпотеза - схожість, яка є серед подібностей (+) - прикладів, є серед подібностей (-) прикладів, але відсутній серед подібностей (+) - прикладів будь-якого іншого класу.

(-) - гіпотеза - схожість, яка є серед подібностей (-) - прикладів даного класу, відсутня серед подібностей (+) - прикладів даного класу, але є подібністю (+) - прикладів будь-якого іншого класу.

(0) - гіпотеза - схожість, яка є серед подібностей (+) прикладів даного класу, є серед подібностей (-) прикладів даного класу, є подібністю (+) - прикладів будь-якого іншого класу.

Для спрощення аналізу основні властивості гіпотез, отриманих за допомогою застосування описаних вище визначень, простежимо на обмеженому наборі гіпотез, що містять підструктуру {2} (див. Таблиця В.1).

Таблиця Б.1 - Набір гіпотез, що містять подструктуру {2}

1 клас	2 клас	3 клас	4 клас	5 клас
(+): {2,4}	(+): {2,18} (+): {2,19}	(+): {2, 17} (+): {2, 24} (+): {2, 25}	(+): {2, 34} (+): {1,2,3,34} (+): {2, 3, 24} (+): {2,20,25}	(+): {2,3,33,34} (+): {2,32,33,34}
(0): {2} (0): {1,2} (0): {2,3} (0): {2,6} (0): {2,9}	(0): {2} (0): {2,3}	(0): {2} (0): {1,2} (0): {2,3} (0): {2,6}	(0): {2} (0): {2,3} (0): {2,33,34}	(-): {2,33,34}
(-): {2,20,25} (-): {2, 34} (-): {2, 3, 24} (-): {1,2,3,34} (-): {2,33,34} (-): {2,3,33,34} (-): {2,32,33,34} (-): {2, 17} (-): {2, 24} (-): {2,25} (-): {2,18} (-): {2,19}	(-): {2,20,25} (-): {2, 34} (-): {2, 3, 24} (-): {1,2,3,34} (-): {2,33,34} (-): {2,3,33,34} (-): {2,32,33,34} (-): {2, 17} (-): {2, 24} (-): {2,25} (-): {2,6} (-): {2,9} (-): {1,2} (-): {2,4}	(-): {2,20,25} (-): {2, 34} (-): {2, 3, 24} (-): {1,2,3,34} (-): {2,33,34} (-): {2,3,33,34} (-): {2,32,33,34} (-): {2,18} (-): {2,19} (-): {2,4} (-): {2,9}	(-): {2,3,33,34} (-): {2,32,33,34} (-): {2, 17} (-): {2, 24} (-): {2,25} (-): {2,18} (-): {2,19} (-): {1,2} (-): {2,6} (-): {2,9} (-): {2,4}	(-): {2,20,25} (-): {2, 34} (-): {2, 3, 24} (-): {1,2,3,34} (-): {2, 17} (-): {2, 24} (-): {2,25} (-): {2,18} (-): {2,19} (-): {2,4} (-): {2} (-): {1,2} (-): {2,3} (-): {2,6} (-): {2,9}

1. Поява в молодших класах (0) - гіпотез, очевидно (виходячи з простоти елементів, що входять до складу даних гіпотез) відповідних (+) - гіпотезам молодших класів (дані гіпотези виділені зеленим кольором). Факт присутності таких (0) -гіпотез може істотно погіршити змістовну інтерпретацію гіпотез і результати класифікації.
2. Поява в старших класах (наприклад: 2,3,4,5 класи) (0) -гіпотез, очевидно (виходячи з простоти елементів, що входять до складу гіпотез) відповідних (+) - гіпотезам молодших класів (дані гіпотези виділені жовтим кольором). Дані гіпотези не можуть бути підставою для прийняття рішення про неналежність об'єкта відповідного класу. Факт присутності таких (0) - і (-) - гіпотез може істотно погіршити змістовну інтерпретацію гіпотез і результати класифікації.
3. Поява в старших класах (наприклад: 2,3,4,5 класи) (-) - гіпотез, відповідних (+) - гіпотезам молодших класів (дані гіпотези виділені фіолетовим кольором). Дані гіпотези не можуть бути підставою для прийняття рішення про не належність об'єкта до старших класів. Факт присутності таких (-) - гіпотез

може істотно погіршити змістовну інтерпретацію гіпотез і результати класифікації.

Проаналізуємо причини цих недоліків. Однією з причин появи браку 3 є визначення в якості (-) - прикладів молодших класів. Загальною причиною появи всіх недоліків є можливість виявлення в старших класах подібностей (підструктур виробів), характерних для молодших класів. Даний факт є причиною «визначення» здебільшого (+) - гіпотез молодших класів (0) - гіпотезами і появи в старших класах необґрунтованих (0) -гіпотез.

Очевидно, що зміна способу вибору (+) - і (-) - приклад не дозволить позбутися від виявлених недоліків (семантичних «дефектів» використовуваної формалізації). Так, вибір як (-) - прикладів тільки молодших класів дозволить позбутися тільки від нестачі 1, а вибір в якості (-) - прикладів проектів тільки старших класів дозволить позбутися тільки від нестачі 3.

Таким чином, єдиним способом поліпшення змістовної інтерпретації гіпотез є модифікація (перевизначення з урахуванням виявлених семантичних «дефектів») їх правил виведення, а саме:

1. визначення подібностей, характерних одночасно для декількох класів в якості (+) - гіпотез слабкої сили;
2. визначення подібностей проектів молодших класів в якості (+) - гіпотез найменшої сили;

Модифікація 2 дозволить виключити недолік 3. Модифікація 1 дозволить виключити недоліки 1 і 2. Так, якщо в старших класах як гіпотез будуть виявлені підструктури виробів, характерних для молодших класів, дані підструктури будуть (+) - гіпотезами слабкої сили у всіх класах, що дозволить виключити їх вплив на результати класифікації в старших класах і, тим не менш, залишить їх в якості (+) - гіпотез в молодших класах. Даний варіант визначення гіпотез і відповідний спосіб класифікації розглянуті в наступному варіанті.

Б.4.1.2 Опис другого варіанту модифікації

Спосіб визначення (+) - і (-) - прикладів для даного варіанту збігається зі способом першого варіанту.

Для даного варіанту модифікації характерний наступний спосіб визначення (+) - і (-) - гіпотез фіксованого класу тривалості ((+) - гіпотези ранжовані, з збільшення їх рангу убуває обґрунтованість): (+1) - гіпотеза - схожість, яка є серед подібностей (+) - прикладів і відсутній серед подібностей (-) - прикладів.

(+2) - гіпотеза - схожість, яка є серед подібностей (+) - прикладів, є серед подібностей (-) - прикладів, але відсутній серед подібностей (+) - прикладів будь-якого іншого класу.

(+3) - гіпотеза - схожість, яка є серед подібностей (+) - прикладів даного класу, є серед подібностей (-) - прикладом даного класу, є подібність (+) - прикладів будь-якого іншого класу.

(+4) - гіпотеза - схожість, яка є серед подібностей (+) - прикладів даного класу, є серед подібностей (-) - прикладом даного класу, є подібність (+) - прикладів декількох інших класів.

(-) - гіпотеза - схожість, яка є серед подібностей (-) - приклад даного класу, відсутній серед подібностей (+) - прикладом даного класу є подібність (+) - прикладів старших класів (одного або декількох). Для ілюстрації виконуваного аналізу основні властивості гіпотез, отриманих за допомогою застосування описаних вище визначень, простежимо на обмеженому наборі гіпотез, що містять підструктуру {2} (див. Таблиця В.2).

Таблиця Б.2 - Набір гіпотез, що містять підструктуру {2}

1 клас	2 клас	3 клас	4 клас	5 клас
(+1): {2,4}	(+1): {2,18}	(+1): {2,25} (+1): {2, 17} (+1): {2, 24}	(+1): {2, 34} (+1):{1,2,3,34} (+1): {2, 3, 24} (+1):{2,20,25}	(+1): {2,3,33,34} (+1): {2,32,33,34}
	(+2): {2,19}			
(+3): {1,2} (+3): {2,6} (+3): {2,9}		(+3): {1,2} (+3): {2,9} (+3): {2,6}	(+3): {2,33,34}	(+3): {2,33,34}
(+4): {2,3} (+4): {2}	(+4): {2,3} (+4): {2}	(+4): {2,3} (+4): {2}	(+4): {2,3} (+4): {2}	
	(+5): {1,2} (+5): {2,6} (+5): {2,9} (+5): {2,4}	(+5): {2,18} (+5): {2,19} (+5): {2,4}	(+5): {2,18} (+5): {2,19} (+5): {1,2} (+5): {2,6} (+5): {2,9} (+5): {2,4} (+5): {2, 17} (+5): {2, 24}	(+5): {2,25} (+5): {2,18} (+5): {2,19} (+5): {2,3} (+5): {2} (+5): {1,2} (+5): {2,6} (+5): {2,9} (+5): {2,4} (+5): {2,20,25} (+5): {2, 34} (+5): {2, 3, 24} (+5): {1,2,3,34} (+5): {2, 17} (+5): {2, 24}
(-): {2,20,25} (-): {2, 34} (-): {2, 3, 24} (-): {1,2,3,34} (-): {2,33,34} (-): {2, 17} (-): {2, 24} (-): {2,3,33,34} (-): {2,32,33,34} (-): {2, 17} (-): {2, 24} (-): {2,25} (-): {2,18}	(-): {2,20,25} (-): {2, 34} (-): {2, 3, 24} (-): {1,2,3,34} (-): {2,33,34} (-): {2,3,33,34} (-): {2,32,33,34} (-): {2, 17} (-): {2, 24} (-): {2,25}	(-): {2,20,25} (-): {2, 34} (-): {2, 3, 24} (-): {1,2,3,34} (-): {2,33,34} (-): {2,3,33,34} (-): {2,32,33,34}	(-): {2,3,33,34} (-): {2,32,33,34}	

Порівнюючи отримані результати з результатами першого варіанту, можна зробити висновок, що основні недоліки першого варіанта усунені:

1. Гіпотези, які в першому варіанті помилково були (0) - гіпотезами в молодших класах, в даному варіанті присутні в молодших класах у вигляді (+) - гіпотез рангу 2-3 (виділені зеленим кольором).
2. Гіпотези, які в першому варіанті помилково були (0) - гіпотезами в старших класах, в даному варіанті присутні в старших класах у вигляді (+) - гіпотез рангу 2-3 (виділені жовтим кольором).
3. Гіпотези, які в першому варіанті помилково були (-) - гіпотезами в старших класах в даному варіанті присутні в старших класах у вигляді (+) - гіпотез рангу 5 (виділені фіолетовим кольором).

Аналіз отриманих гіпотез, а саме ранжування за силою отриманих (+) - гіпотез, дозволяє вважати прийнятним наступне правило класифікації: об'єкт класифікується тим класом, в якому немає (-) - гіпотез, а наявні (+) - гіпотези мають найбільшу силу, якщо таких класів кілька - вибирається найбільший з цих класів. У розділі 5.3 наведено формальне подання визначень правил першого і другого роду, а також предикатів, що представляють посилки висновків за аналогією, що відповідає даному варіанту модифікації. Розділ В.6.2 містить більш детальну змістовну інтерпретацію отриманих гіпотез. У розділі 5.4 наведені результати класифікації вихідної навчальної вибірки і нових проектів за допомогою обраного правила класифікації на основі бази сформованих гіпотез.

Б.4.2 Змістовна інтерпретація отриманих гіпотез

Вихідні дані, а саме - множина класифікованих і нових (некласифікованих) проектів розробки виробів, представлено на малюнку В.10. На малюнку в.11 наведена таблиця відповідності номерів елементів в складі проектів їх найменувань.

Клас 1	Клас 2	Клас 3	Клас 4	Клас 5	Не класифіковані
1.1 ШТК 102ВН-03С: {1,2,3,4}: 0	2.1 ШТК 104ВН-01С: {2,3,13}: 0	3.1 ШТК 102ВН-04С: {1,2,3,20}: 0	4.1 ШТК 102ВН-06С: {1,2,3,33,34}: 0	5.1 ШТК 103ВН-03С: {2,3,22,33,34}: 0	ШТК 102КН-03С: {40,37}: 0 клас: 0
1.2 ШТК 100ВН-01С: {3,4,5}: 0	2.2 ШТК 104ВН-02С: {2,3,14}: 0	3.2 ШТК 102ТН-02С: {1,2,6,21}: 0	4.2 ШТК 102ТН-03С: {2,6,33,34,40}: 0	5.2 ШТК 103КН-02С: {2,33,34,29}: 0	ШТК 102ТН-06С: {1,6,37}: 0 клас: 0
1.3 ШТК 102ТН-01С: {1,2,6}: 0	2.3 ШТК 102ТН-04С: {1,5,6,15}: 0	3.3 ШТК 103ВН-01С: {2,3,22}: 0	4.3 ШТК 103ВН-02С: {2,3,22,24}: 0	5.3 ШТК 104ТН-01С: {2,33,34,32,30,35}: 0	ШТК 102ВН-06С: {2,3,14}: 0 клас: 0
1.4 ШТК 100ТН-01С: {4,5,6}: 0	2.4 ШТК 100: {1,2,6,16}: 0	3.4 ШТК 103ТН-01С: {2,6,23}: 0	4.4 ШТК 103КН-01С: {2,24,29}: 0	5.4 ШТК 103К-01С: {33,34,29,36}: 0	ШТК 100ВН-09С: {2,3,10}: 0 клас: 0
1.5 ШТК 100ВН-02А: {7,8,9}: 0	2.5 ШТК 104ВН-03С: {2,3,13,16}: 0	3.5 ШТК 102ВН-05С: {1,2,3,24}: 0	4.5 ШТК 104КН-05С: {2,20,25,26}: 0	5.5 ШТК 103ТН-02С: {34,32,35,37,39,38}: 0	
1.6 ШТК 100ВН-03С: {9,11,10}: 0	2.6 ШТК 104ВН-04С: {2,3,13,17}: 0	3.6 ШТК 100ВН-02С: {3,4,5,24}: 0	4.6 ШТК 104КН-06С: {2,20,25}: 0	5.6 ШТК 104К-01С: {17,18,33,36}: 0	
1.7 ШТК 100ВН-04С: {3,5,12,10}: 0	2.7 ШТК 104КН-03С: {2,18,19}: 0	3.7 ШТК 104ВН-05С: {2,3,16,17,25}: 0	4.7 ШТК 102ВН-07С: {2,3,14,24,31}: 0	5.7 ШТК 104ВН-07С: {2,3,13,33,34,36}: 0	
	2.8 ШТК 100ВН-05С: {3,10,19,28}: 0	3.8 ШТК 104ВН-06С: {2,3,17,19,25}: 0	4.8 ШТК 102ТН-04С: {2,3,14,24,32}: 0	5.8 ШТК 102ТН-05С: {2,19,33,34,40,32}: 0	
		3.9 ШТК 104КН-04С: {2,25,26,27}: 0			

Рисунок Б.10 - Вихідні дані

ІС прогнозування тривалості виробництва

Вихідні дані ДСМ методу

Проект Номенклатор Клас

Номенклатор:

Номер	Найменування
1	КОР N181
2	МОД НАГР МН0600.1
3	МОД ВЕНТ 6
4	КОР S39
5	МОД НАГР МН0300.1
6	МОД ТЕПЛООБ 380-01
7	КОР S50
8	МОД НАГР МН0500
9	МОД ВЕНТ 3
10	КОР S39_2
11	МОД НАГР МН0300.2
12	МОД МР-11-07
13	КОР N210
14	КОР N161
15	МОД МР-1ВС4-01
16	МОД МР-1ВС6-01
17	МОД МР-1В6-01
18	КОР N276
19	МОД МР-1ВС4-02
20	У48-040 2/1000 4UF
21	У48-140 2/1800 4UF
22	КОР N174
23	КОР N229
24	МОД СВО1У-01К
25	КОР N244
26	МОД МР-1ВРСФ4-33
27	МОД МР-1В4-01
28	МОД НАГР МН0600.2
29	КОР N229_1
30	КОР N244_1
31	МОД МР-1ВРС4-02
32	МОД ТЕПЛООБ 380-02
33	У48-400 3/2500 4U
34	МОД СВО-02Н
35	МОД ОХЛ250.1
36	МОД МР-1ВРС9-01
37	МОД НАГР МН1000
38	У48-400 4/2500 4U
39	КОР N229_2
40	КОР N240

Добавить Удалить

Рисунок Б.11 - Дані номенклатора

На рисунку В.12 представлені гіпотези кожного класу тривалості, отримані за допомогою застосування правил правдоподібного виведення першого роду.

2. (+) - гіпотези 3-го і 4-го рангу, які зустрічаються в одному / кількох класах, включаючи розглянутий. Очевидно, що такі гіпотези не повинні відігравати вирішальну роль при класифікації, а й (-) - і (0) - гіпотезами бути не можуть, тому визначення їх в якості (+) - гіпотез слабкої сили цілком доречно.
3. (+) - гіпотези 5 типу - гіпотези, які зустрічаються в одному / кількох менших класах, але не зустрічаються в даному класі. Очевидно, що такі гіпотези не повинні відігравати вирішальну роль при класифікації, а й (-) - і (0) - гіпотезами бути не можуть (так як виходячи з тенденції ускладнення гіпотез в старших класах можуть зустрічатися підструктури проектів молодших класів), тому визначення їх в якості (+) - гіпотез найслабшої сили цілком доречно.
4. (-) - гіпотези - це залежності, які зустрічаються в одному або декількох старших класах і не зустрічаються в даному класі. Дані гіпотези є більш складними ніж гіпотези, характерні для даного класу, тому їх наявність в проекті може бути підставою для неналежності до розглянутого класу. Таким чином, визначення їх як (-) - гіпотез цілком доречно.

Проведемо більш детальну змістовну інтерпретацію виявлених гіпотез 1-ого типу.

На основі виявлених гіпотез, можна зробити висновок, що для першого класу найбільш характерні комбінації елементів, представлені в таблиці В.2.

Таблиця В.2 - Гіпотези 1 рангу 1 класу тривалості і відповідні їм елементи

Комбінації елементів	Гіпотези
Корпус першої складності	{1} {7} {10}
Модуль вентиляції + Корпус першої складності	{1,3} {1,9} {7,9} {3,4} {4,9}

Модуль нагрівача + Корпус першої складності	{1,5} {2,4} {4,5}
Модуль нагрівача + Модуль вентиляції	{3,5}
Модуль теплообмінника першої складності + Корпус першої складності	{1,6} {4,6}

На основі виявлених гіпотез, можна зробити висновок, що для другого класу найбільш характерні комбінації елементів, представлені в таблиці В.3.

Таблиця Б.3- Гіпотези 1 рангу 2 класу тривалості і відповідні їм елементи

Комбінації елементів	Гіпотези
Корпус другої складності	{13}{18}{14}
Модуль нагрівача + Корпус другий складності	{2,18} {5,14}
Модуль розподілу другої складності	{19}
Модуль теплообмінника + модуль розподілу другої складності	{6,16}
Модуль нагрівача + модуль розподілу другої складності	{2,19}
Корпус другий складності + модуль розподілу другої складності	{18,19}

Таким чином, виявлені гіпотези відображають наступні причини збільшення часу виробництва виробів і, як наслідок, приналежності об'єкта до другого класу тривалості:

1. Заміна корпусу першої складності на корпус другої складності ($\{5,3,4\} \rightarrow \{2,3,13\}$).

Коментар: до корпусів першої складності відносяться корпуси з одинарними стінками без утеплення, незалежно від габаритів. До корпусів другої складності відносяться корпуси з подвійними стінками і утеплювачем. Час складання корпусів з подвійними стінками істотно перевищує час складання корпусів з одинарними стінками.

2. Додавання модуля розподілу другої складності ($\{4,5,6\} \rightarrow \{1,2,6,16\}$).

Коментар: додавання модуля розподілу збільшує обсяг і складність виконуваних робіт, оскільки наявність модуля розподілу вимагає монтажу складнішої електричної схеми.

3. Заміна корпусу першої складності на корпус другої складності і додавання модуля розподілу другої складності ($\{1,2,3\} \rightarrow \{2,18,19\}$).

Таким чином, аналіз елементів, що відповідають отриманим гіпотезам, показав, що дані елементи можуть обґрунтовано вважатися причинами збільшення часу виробництва виробів до часу, відповідного 2-го класу тривалості.

На основі виявлених гіпотез, можна зробити висновок, що для третього класу найбільш характерні комбінації елементів, представлені в таблиці В.4.

Таблиця В.4- Гіпотези 1 рангу 3 класу тривалості і відповідні їм елементи

Комбінації елементів	Гіпотези
Корпус другий складності + нагрівач	{2,25}
Модуль нагрівача + модуль розподілу другої складності	{2,17}
Модуль нагрівача + модуль вентилятора + модуль розподілу другої складності + модуль корпусу другої складності	{5,9,17,18}
модуль вентилятора + модуль розподілу другої складності + модуль корпусу другої складності	{9,17,18}

Модуль розподілу другої складності	{17}
Модуль вентилятора + 2 модуля розподілу другої складності + Модуль корпусу другої складності	{9,17,18,19}
Модуль нагрівача + Модуль супервізора другий складності	{2,24}
Установка харчування	{20}
Установка харчування + модуль теплообмінника	{6,21}
Два модуля розподілу другої складності	{16,17} {9,17,18,19}
Корпус третьої складності	{23} {22}

Таким чином, виявлені гіпотези дозволяють простежити наступні причини збільшення часу виробництва виробів:

1. Заміна корпусу другої складності на корпус третьої складності ($\{5,6,4\} \rightarrow \{2,6,23\}$).

Коментар: до корпусів третьої складності відносяться корпуси з подвійними стінками і утеплювачем, що складаються з 2-х відсіків, що істотно збільшує час збирання в порівнянні з корпусами другий складності.

2. Додавання установки харчування ($\{1,2,3\} \rightarrow \{1,2,3,20\}$).

Коментар: додавання установки харчування збільшує обсяг і складність виконуваних робіт, оскільки наявність установки харчування вимагає виготовлення і пайки друкованих плат, а також об'ємного монтажу складної електричної схеми.

3. Додавання супервізора об'єкта другої складності ($\{1,2,3\} \rightarrow \{7,2,9,24\}$).

Коментар: додавання модуля супервізора другої складності збільшує обсяг і складність виконуваних робіт, оскільки наявність модуля супервізора вимагає виготовлення і пайки друкованих плат, а також об'ємного монтажу електричної схеми підключення датчиків.

4. Наявність двох модулів розподілу другої складності ($\{2,3,13,16\}$ -> $\{2,3,16,17,25\}$).

Коментар: наявність двох різних модулів розподілу збільшує складність виготовлення шафи як за рахунок збільшення обсягу робіт по монтажу модулів розподілу, так і за рахунок необхідності монтажу схеми з'єднань модулів розподілу.

Таким чином, аналіз елементів, що відповідають отриманим гіпотезам, показав, що частина виявлених гіпотез (виділені синім) можуть обґрунтовано вважатися причинами збільшення часу виробництва виробів до часу, відповідного 3-ому класу тривалості.

Однак частина виявлених гіпотез 1-ого типу (наприклад, гіпотези $\{5,9,17,18\}$, $\{9,17,18\}$, $\{2,25\}$, $\{17\}$) більше характерні причин приналежності об'єктів до 2-го класу. Їх поява як гіпотез 1-ого рангу в третьому класі є наслідком відсутності виробів, вироблених з такими елементами в другому класі, тобто є наслідком недостатньої репрезентабельності вибірки, що відповідає вихідним даним.

На основі виявлених гіпотез, можна зробити висновок, що для четвертого класу найбільш характерні комбінації елементів, представлені в таблиці В.5.

Таблиця В.5 - Гіпотези 1 рангу 4 класу тривалості і відповідні їм елементи

Комбінації елементів	Гіпотези
Модуль нагрівача + Модуль супервізора першої складності	$\{2,34\}$
Модуль нагрівача + Модуль супервізора другий	$\{5,24\}$

складності	
Корпус першої складності + Модуль нагрівача + модуль вентилятора + Модуль супервізора першої складності	{1,2,3,34}
Модуль нагрівача + модуль вентилятора + модуль супервізора другий складності	{2,3,24} {5,3,24}
Модуль вентилятора + Модуль супервізора другий складності	{3,24}
Модуль супервізора другий складності	{24}
Модуль нагрівача + Модуль корпусу третьої складності + Модуль супервізора другий складності	{3,22,24} {3,24,29}
Модуль нагрівача + Установка харчування + Модуль корпусу другої складності	{2,20,25}
Модуль корпусу другої складності + Модуль супервізора другий складності + Модуль розподілу другої складності	{14,24,31}
Модуль супервізора об'єкта першої складності + установка харчування	{38,34}

Таким чином, виявлені гіпотези дозволяють простежити наступні причини збільшення часу виробництва виробів:

1. Заміна корпусу першої складності на корпус другої складності при наявності установки харчування ($\{1,2,3,20\} \rightarrow \{2,9,20,25\}$).

Коментар: Сумарний обсяг робіт з монтажу установки харчування і складанні корпусу другої складності переводить проект в четвертий клас.

2. Заміна корпусу першої складності на корпус третьої складності при наявності супервізора об'єкта другої складності (1,2,3,24-> 2,3,22,24).

Коментар: Сумарний обсяг робіт з монтажу супервізора об'єкта другої складності, підключення датчиків і збірці корпусу другої складності переводить проект в четвертий клас.

3. Додавання супервізора об'єкта першої складності при наявності установки харчування (4,5,9,20-> 1,2,3,33,34).

Коментар: Додавання супервізора об'єкта першої складності (простішого в збірці, ніж супервізор об'єкта другої складності) вимагає також монтажу схеми підключення датчиків, що істотно збільшує складність проекту в цілому.

4. Додавання модуля розподілу другої складності до проекту і заміна корпусу першої складності на корпус другої складності при наявності модуля супервізора другої складності (1,2,3,24-> 2,3,14,24,31).

Коментар: додавання модуля розподілу другої складності до проекту і заміна корпусу першої складності на корпус другої складності збільшує складність виробництва вироби за рахунок монтажу модуля розподілу і ускладнення збірки корпусу.

Таким чином, аналіз елементів, що відповідають отриманим гіпотезам, показав, що частина виявлених гіпотез (виділені синім) може обґрунтовано вважатися причинами збільшення часу виробництва виробів до часу, відповідного 4-му класу тривалості.

Однак частина виявлених гіпотез 1-ого типу (наприклад, гіпотези {5,24}, {2,3,24}, {3,24}, {24}) більше характерні причин приналежності об'єктів до 3-ого класу. Їх поява як гіпотез 1-ого типу в четвертому класі є наслідком відсутності появи таких гіпотез як гіпотез 1-ого типу у другому класі, тобто є наслідком недостатньої репрезентабельності вибірки, що відповідає вихідним даних.

На основі виявлених гіпотез, можна зробити висновок, що для п'ятого класу найбільш характерні комбінації елементів, представлені в таблиці В.6.

Таблиця Б.6 - Гіпотези 1 рангу 5 класу тривалості і відповідні їм елементи

Комбінації елементів	Гіпотези
Модуль супервізора першої складності + установка харчування	{33,34}
Модуль нагрівача + Модуль супервізора першій складності + установка харчування	{5,33,34}
Модуль нагрівача + модуль вентилятора + Модуль супервізора першої складності + установка харчування	{3,2,33,34,} {5,9,33,34}
Модуль супервізора першої складності + установка харчування + Корпус третьої складності	{33,34,29} {33,34,22}
Модуль теплообмінника другий складності	{32}
Модуль теплообмінника другий складності + модуль нагрівача	{5,32}
Модуль теплообмінника другий складності + модуль супервізора першої складності	{34,32}
Установка харчування + Модуль теплообмінника другий складності + модуль супервізора першої складності	{33,34,32}
Модуль нагрівача + установка харчування + Модуль теплообмінника другий складності + модуль супервізора першої складності	{2,33,34,32}

Установка харчування + Модуль теплообмінника другий складності + модуль супервізора першої складності + Модуль розподілу другої складності	{19,33,34,32}
Модуль теплообмінника другий складності + модуль охолодження	{32,35}
Модуль теплообмінника другий складності + модуль супервізора першої складності + модуль охолодження	{32,34,35,}
Модуль теплообмінника другий складності + модуль охолодження + установка харчування	{32,35,38}
Модуль супервізора першої складності + установка харчування + модуль розподілу третьої складності	{33,34,36,}

Таким чином, виявлені гіпотези дозволяють простежити наступні причини збільшення часу виробництва виробів:

1. Заміна корпусу другої складності на корпус третьої складності при наявності установки харчування і супервізора об'єкта першої складності (40,2,6,33,34-> 29,2,33,34).

Коментар: Сумарний обсяг робіт з монтажу установки харчування, складанні супервізора об'єкта першої складності, монтажу схеми підключення датчиків і збірці корпусу третьої складності переводить проект в п'ятий клас.

2. Додавання модуля теплообмінника другої складності і модуля розподілу другої складності при наявності установки харчування і супервізора об'єкта першої складності (1,2,3,33,34-> 40,2,33,34,32,19).

Коментар: додавання модуля теплообмінника другої складності істотно збільшує обсяг робіт (за рахунок збирання теплообмінника) і переводить проект в п'ятий клас тривалості.

3. Додавання модуля охолодження, установки харчування, при наявності модуля теплообмінника (2,3,14,24,32-> 39,37,32,33,35,34).

Коментар: додавання модуля охолодження на елементах Пельтьє підвищує складність проекту за рахунок необхідності збирання, електричного монтажу та наладки модуля охолодження. До складу модуля охолодження входить складний корпус, модулі Пельтьє і плата контролера охолоджувача. Модуль охолодження може застосовуватися тільки разом з супервизором об'єкта.

4. Додавання модуля розподілу третьої складності при наявності установки харчування і супервізора об'єкта першої складності (40,2,6,33,34-> 2,3,13,33,34,36).

Коментар: додавання модуля розподілу третьої складності до проекту, має в складі модуль супервізора першої складності і установку харчування, переводить проект в п'ятий клас тривалості за рахунок необхідності збирання і електричного монтажу модуля розподілу третьої складності і схеми його підключення.

Таким чином, аналіз елементів, що відповідають отриманим гіпотезам (залежностям), показав, що частину виявлених залежностей можна обґрунтувати причинами збільшення часу виробництва виробів до термінів, відповідності 5-ому класу тривалості.

Однак частина виявлених гіпотез 1-ого типу (наприклад, гіпотези {33,34}, {5,33,34}) більше характерні причинам приналежності об'єктів до 4-ого класу. Їх поява як гіпотез 1-ого типу в п'ятому класі є наслідком відсутності появи таких гіпотез як гіпотез 1-ого типу в четвертому класі, тобто є наслідком недостатньої репрезентабельності вибірки, що відповідає вихідним даних.

ДОДАТОК В
(обов'язковий)
Матеріали публікацій

Міністерство освіти і науки України
Хмельницький національний університет



ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XII всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2020»

9-10 листопада 2020

Хмельницький 2020

Гордійчук Б. Г., Манзюк Е. А., Скрипник Т. К. Виявлення аномалій в даних	72
Городний М. С., Тітова В. Ю. Розробка архітектури додатку на основі технологій «розумний будинок» та «інтернет речей»	75
Гребінчук А. Д., Поліщук В. Ю., Форкун І. В. Модель багаторівневої автоматизованої системи керування будівельним виробництвом	78
Грипинська Н. В., Дяблов Б. В. Автоматизована система планування рекламної кампанії для малого та середнього бізнесу	82
Грипинська Н. В., Коломієць О. В. Автоматизована система виявлення та класифікації твердих побутових відходів на зображеннях	86
Демчук Б. Р. Динамічна модель перебігу вірусного захворювання	91
Долгополов С. Ю., Цюцюра М. І. Інноваційність використання технології глибокого навчання у контрольно-вимірювальному приладі будівельного спрямування «Builder of the Future»	97
Драпатий О. В., Драч І. В. Методи мережевого моделювання. Сучасні напрямки	102
Євдокімов О. В., Татаревська О. Г., Радельчук Г. І. Автоматизація і комп'ютерно-інтегровані технології моніторингу сонячних панелей у реальному масштабі часу	113
Живага В. В., Шевченко Д. О. Інтегрована Internet of Things система на основі одноплатного комп'ютеру	115
Жовнір М. Ю., Кисіль Т. М. Неформальне пояснення ДСМ-методу автоматичного породження гіпотез в задачах адаптивної поведінки ІС	120
Злотаренчук О. І., Кучерук О. Я. Сучасні підходи до організації маршрутів комплектації замовлень на складі	123
Казлаускайте А. С., Шендрік С. О. Інформаційна технологія визначення впливу погодних умов на продуктивність альтернативних джерел енергії	127

УДК 004.4

Жовнір М. Ю., Кисіль Т. М.

Хмельницький національний університет

НЕФОРМАЛЬНЕ ПОЯСНЕННЯ ДСМ-МЕТОДУ АВТОМАТИЧНОГО ПОРОДЖЕННЯ ГІПОТЕЗ В ЗАДАЧАХ АДАПТИВНОЇ ПОВЕДІНКИ ІС

Розглянуто застосування математичного моделювання методом ДСМ автоматичного породження гіпотез у галузі проектування інформаційних систем. У статті висвітлюються сучасні досягнення у галузі проектування та навчання інформаційних систем, отримані результати дозволяють спрогнозувати оптимальний час виконання завдань системою.

Applied aspects of mathematical modeling by the DSM method of automatic generation of hypotheses in the field of information systems design is considered. The article highlights the current achievements in the field of design and training of information systems, the results allow us to predict the optimal execution time of tasks by the system.

ДСМ-метод автоматичного породження гіпотез був запропонований В.К.Фінном на початку 1980-х років. В даний час ДСМ-метод розглядається як оригінальна сукупність логіко-комбінаторних технологій інтелектуального аналізу даних, що використовує формалізовані правила правдоподібних міркувань. В основі ДСМ-методу АПГ лежить синтез трьох методів прогнозування «емпірична індукція - структурна аналогія - абдукція». Саме тому ДСМ-системи засновані на принципі: «схожість фактів викликає наявність ефекту і його повторюваність, а не навпаки».

ДСМ-метод:

- виявляє закономірності в даних за допомогою індукції в стилі Д.С. Мілля,
- породжує правила усунення невизначеності в даних, використовуючи фальсифікацію в стилі К. Поппера,
- формує передбачення за допомогою оригінальної версії міркувань за аналогією,
- оцінює правдоподібність породжених гіпотез з допомогою варіантів абдукції Ч.

С. Пірса

Абревіатура «ДСМ» - це ініціали Джона Стюарта Мілля, формалізовані правила індуктивної логіки якого є ідеологічним фундаментом ДСМ-методу.

Дані предметної області повинні бути добре структуровані, що передбачає подання їх у вигляді окремих об'єктів з описом, що допускає визначення операції подібності, вкладення, об'єднання і різниці на об'єктах-фактах. Серед ознак, що описують факт, повинна бути присутня деяка множина властивостей («ефектів»), за

яким ми можемо розділити факти на позитивні (+) і негативні (-) приклади, окрім цього непотрібно забувати і про можливі суперечливі приклади (0, тобто приклади, що володіють і не володіють ефектом одночасно) і невизначені (τ) - Теана (факти, що вимагають дослідження).

ДСМ-система призначена для виявлення зв'язку між структурою об'єкта і його поведінкою. Цей зв'язок інтерпретується як причинно-наслідковий зв'язок. Як правило, характерні риси структури об'єкта вважаються причинами особливостей його поведінки, але в деяких випадках, прийнята зворотна інтерпретація: сукупність елементів поведінки вважається причиною безлічі характеристик його структури.

Формування гіпотез про можливі причини відбувається з допомогою формалізованих індуктивних міркувань. Формування гіпотез про наявність (відсутність) властивостей можна інтерпретувати як застосування формалізованих міркувань за аналогією.

В даний час переважна більшість інформаційних систем в області створення моделей поведінки, використання методів ДСМ для управління різними об'єктами мають в основному теоретичний, абстрактний характер. Причин цього багато, однієї з них є затратність на реалізацію проектів, які працюють з реальними технічними пристроями. Наприклад в ході реалізації проекту "адаптантов-2005" (див. [Добринін та інш., 2005], [Добринін, Карпов, 2005]) був створений універсальна мобільна система "Амур", що є полігоном для відпрацювання і демонстрації різних алгоритмів і методів керування.

У даній роботі мова йде про реалізацію на базі створеного пристрою деяких найпростіших форм адаптивного поведінки, починаючи з моделювання умовно-рефлекторного поведінки і закінчуючи застосуванням еволюційного моделювання і ДСМ-методу для вирішення конкретних задач. Класифікацію вхідних сигналів можна здійснювати за допомогою ДСМ методу [Фінн, 1991].

$$E = \{e_i\} = \{(X_i, u^i)\}, \quad (1)$$

де X_i - вектор сигналів, u^i - вектор управління (стан виконавчих механізмів). Елементи векторів сигналів і управління представляються парами довічних значень: $\text{вкл} = \{01\}$, $\text{викл} = \{10\}$.

В робочому режимі досліджувана ДСМ система отримує на вході сигнали рецепторів, з яких формується тестовий вектор. Ухвалення рішення відбувається шляхом перевірки вкладення гіпотез в цей вектор. Якщо в тестовий вектор сигналів рецепторів вкладається гіпотеза, то система повинна діяти відповідно до неї. Якщо ж жодної гіпотези, не знайдено, то це невідомий стан, для якого потрібно сформувати випадковий вектор управління. Якщо база фактів повна, то характер

поведінки системи в робочому режимі під управлінням ДСМ системи нічим не повинен відрізнятися.

Отже, у даній роботі мова йде про реалізацію на базі створеного пристрою деяких найпростіших форм адаптивного поведінки, починаючи з моделювання умовно-рефлекторного поведінки і закінчуючи застосуванням еволюційного моделювання і ДСМ-методу для вирішення конкретних задач. Досліджувана ДСМ-система призначена для виявлення зв'язку між структурою об'єкта і його поведінкою. Цей зв'язок інтерпретується як причинно-наслідковий зв'язок. Як правило, характерні риси структури об'єкта вважаються причинами особливостей його поведінки, але в деяких випадках, прийнята зворотна інтерпретація: сукупність елементів поведінки вважається причиною безлічі характеристик його структури

ДСМ система в робочому режимі отримує на вході сигнали рецепторів, з яких формується тестовий вектор. Ухвалення рішення відбувається шляхом перевірки вкладки гіпотез в цей вектор.

Якщо ж жодної гіпотези, не знайдено, то це невідомий стан, для якого потрібно сформувані випадковий вектор управління, а в випадку, коли база фактів повна, то характер поведінки системи в робочому режимі під управлінням ДСМ системи нічим не повинен відрізнятися.

Перелік посилань

1. Фінн В.К. Правдоподібні міркування в інтелектуальних системах типу ДСМ // Підсумки науки і техніки. Сер. "Інформатика". Т. 15. - М.: 1991. - С. 54-101..
2. Карпов В.Е. Еволюційне моделювання. Проблеми форми і змісту // Новини штучного інтелекту №5, 2003
3. Добринін Д.А., Карпов В.Е., Мещерякова Т.В., Степанов С.М. Моделювання деяких найпростіших форм поведінки: від умовних рефлексів до індуктивної адаптації // Зб. наукових праць I Міжнародної конференції "Системний аналіз та інформаційні технології САІТ-2005", М.: КомКнига, Т.1, стр. 188-193
4. Добринін Д.А., Карпов В.Е., Мещерякова Т.В., Степанов С.М. Адаптивний мобільний робот // Мобільні роботи і мехатронні системи: Матеріали наукової школи-конференції, Москва, 21-25 березня 2005 року Частина 1. М.: Изд-во Моск. ун-ту, 2005., стор. 137-143

Додаток Г
Презентаційні матеріали

Слайд 1:

Презентація до дипломної роботи на тему:
“Метод прогнозування показників виробничих процесів на основі
інтелектуального аналізу даних”

Виконала студентка ПМм-19-1
Жовнір М.Ю
Керівник канд.ф-м.н., доцент
Т.М. Кисіль

Слайд 2:

Мета: Виконати аргументований вибір інструментарію (математичних моделей і програмних засобів) для вирішення спеціального класу задач прогнозування інтервалу тривалості виробництва складних технічних об'єктів (кліматичних шаф) в умовах неповної інформації про їх детальну внутрішню структуру, а саме на етапі передконтрактних переговорів, тобто до виконання їх детального проектування і розробки.

Об'єкт: Об'єктом дослідження є процес прогнозування показників виробничих процесів на основі інтелектуального аналізу даних.

Предмет: Предметом дослідження є оптимізаційні методи та практичні рекомендації щодо прогнозування термінів виробництва виробів

Слайд 3:

Для досягнення поставленої мети мають бути вирішені наступні завдання:

- 1) Аналіз особливостей прогнозування інтервалу тривалості виробництва складних технічних об'єктів;
- 2) Формалізація завдання інтелектуального аналізу даних;
- 3) Розробка методу вирішення задач прогнозування;
- 4) Розробка моделі інтелектуального планування;
- 5) Розробка алгоритмів, вибір та реалізація програмного забезпечення для вирішення поставленої задачі;

Слайд 4:

Опис початкових даних:

Кліматичне обладнання складається з різної кількості модулів, архітектура взаємодії яких представлена у вигляді ієрархічної структури (дерева).



Для вирішення поставленого завдання потрібно використовувати логічні алгоритми класифікації. Основні вимоги до шуканого логічного алгоритму класифікації:

1. Надання методології вирішення поставленого завдання класифікації.
2. Можливість інтерпретації результатів класифікації в змістовних межах розв'язуваної прикладної задачі.
3. Можливість виявляти закономірність на невеликій вибірці вихідних даних.

Слайд 5:

ДСМ-метод автоматичного породження гіпотез

Основні переваги ДСМ методу АПГ:

- він враховує індивідуальні особливості об'єктів дослідження;
- здатність породжувати гіпотези на малих масивах даних;
- можливість працювати з відкритими (поповнювальними) масивами даних;

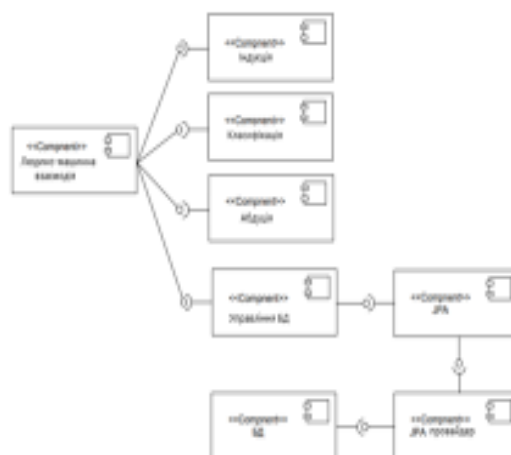
ДСМ метод АПГ активно застосовується при вирішенні наступних задач:

- автоматизованої побудови формалізації знань,
- технічної діагностики,
- дослідження детермінант соціальної поведінки,
- медична діагностика,
- навчання мобільних роботів

Слайд 6:

Розроблено інформаційну модель структурно поділену на модулі («Індукція», «Класифікація», «Абдукція», «Управління БД» і модуль «Людино - машинна взаємодія»).

Загальна архітектура і структура ІС



Модуль «Людино-машинна взаємодія» реалізує графічний інтерфейс користувача, за допомогою якого здійснюється звернення до модулів «Управління БД», «Індукція», «Класифікація», «Абдукція» та забезпечує наступною функціональністю:

- введення вхідних даних;
- редагування вхідних даних;
- здійснення інтелектуального аналізу вхідних даних за допомогою ДСМ-методу;
- перегляд вихідних даних і результатів інтелектуального аналізу;

Слайд 7:

Для обліку даних змін необхідно послідовно виконати кроки ДСМ - методу, а саме оновити базу гіпотез за допомогою застосування правил 1-ого роду, класифікувати всі некласифіковані об'єкти, включаючи нові, і виконати перевірку умови каузальної повноти

Результати перевірки здійсненності умови каузальної повноти

В результаті виконаних налаштувань стандартних ДСМ методів вдалося розробити проблемно-орієнтовану ІС, результати застосування якої в обраній області є обґрунтованими і легко інтерпретуються.

Таким чином, на підставі проведених експериментів можна аргументовано стверджувати, що ДСМ-метод (як система математичних моделей і алгоритмічних засобів) є зручною платформою для практичного застосування в прикладних проблемно орієнтованих системах ІАД.

Слайд 8:

Висновки

- 1) Розроблений ряд алгоритмів, що дозволяють вирішити поставлену задачу прогнозування класу тривалості виробництва технічного об'єкта;
- 2) Розроблено модель швидкого пошуку глобальних подібностей;
- 3) Розроблено схему і алгоритми функціонування системи планування дій на основі прецедентів;
- 4) Реалізовано програмні засоби, що дозволяють реалізувати систему на базі розроблених методів та алгоритмів;

Розроблена ІС може бути легко адаптована для вирішення завдання прогнозування будь-яких оцінних характеристик об'єкта в умовах неповної інформації про властивості аналізованих об'єктів (тобто до їх детального проектування) внесенням лише невеликих змін і налаштувань в інструментальні засоби розробленої ІС.

Загальна готовність дипломної роботи станом на 09.11.2020 складає – 85%

Було подано та рекомендовано для включення до Збірника наукових праць Конференції АПКН-2020, статтю: «НЕФОРМАЛЬНЕ ПОЯСНЕННЯ ДСМ-МЕТОДУ АВТОМАТИЧНОГО ПОРОДЖЕННЯ ППОТЕЗ В ЗАДАЧАХ АДАПТИВНОЇ ПОВЕДІНКИ ІС»



Имя пользователя:
Kafedra TMIT KhNU

Дата проверки:
01.12.2020 21:10:47 EET

Дата отчета:
01.12.2020 21:21:33 EET

ID проверки:
1005321432

Тип проверки:
Doc vs Internet + Library

ID пользователя:
100005657

Название файла: Жовнір_ПМи-19-1

Количество страниц: 76 Количество слов: 15456 Количество символов: 114812 Размер файла: 3.16 MB ID файла: 1005444323

1.27% Совпадения

Наибольшее совпадение: 0.41% с Интернет-источником (https://studopedia.com.ua/1_15058_perevagi-derev-rishen.htm..)

0.96% Источники из Интернета 41 Страница 78

0.38% Источники из Библиотеки 30 Страница 78

0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

0% Исключений

Нет исключенных источников

Модификации

Обнаружены модификации текста. Подробная информация доступна в онлайн-отчете.

Замененные символы 59

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом **0.0%**

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 8%**

ID: 81109 Название: Метод прогнозування показників виробничих процесів на основі інтелектуального аналізу даних Добавлено в БД: 2020-11-24 Авторы: Жовнір Марія Юрївна Руководители: Кисіль Тетяна Миколаївна Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	95451	710	293 (0%)	6 (1%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник _____ студентка групи ПМм-19-1 Жовнір М. Ю. _____

Тема Метод прогнозування показників виробничих процесів на основі інтелектуального аналізу даних _____

Спеціальність 113 – Прикладна математика

Обсяг дипломної роботи:

Кількість листів креслень 0; кількість сторінок записки 145

1. Короткий зміст ДР та прийнятих рішень Представлена робота присвячена актуальній темі в області прогнозування виробничих показників на основі інтелектуального аналізу даних, та складається з наступних розділів: вступ, формальна постановка задачі, огляд відомих методів розв'язання задач прогнозування, можливості та особливості застосування ДСМ-методу, прикладна задача: постановка та розв'язання, висновки, додатки.

2. Висновок про відповідність ДР поставленому завданню Магістерська кваліфікаційна робота виконана у відповідності з завданням із дотриманням всіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі студент провів детальний аналіз предметної області, дослідив метод інтелектуального аналізу даних, представив та описав особливості існуючих рішень в розглянутій сфері, на основі яких довів актуальність роботи і визначив вимоги для створюваної системи. В другому розділі було обрано та обгрунтовано вибір методу автоматичного породження гіпотез, розроблено модель інтелектуального планування. Відповідно в третьому і четвертому розділах автором проведено проектування та розробку системи, здійснено опис програмної реалізації програмного продукту та представлено результати роботи системи.

4. Позитивні сторони роботи До позитивних сторін роботи слід віднести актуальність даного напрямлення дослідження, деталізацію аналізу усіх розглянутих стратегій вирішення проблеми та поглиблене опрацювання всіх аспектів реалізації з практичним використанням запропонованого рішення.

5. Негативні сторони роботи До негативних сторін роботи слід віднести недоліки по оформленню представленого матеріалу, що були виправлені.

6. Оцінка графічного оформлення та пояснювальної записки роботи Дані матеріали роботи є структурованими у чіткій та логічній формі та відображають послідовність виконання поставлених завдань. І хоча й в них було знайдено декілька стилістичних та орфографічних помилок, вони були пізніше усунені. Тому дане виконання пояснювальної записки та графічного оформлення заслуговує оцінки «добре».

7. Відгук про роботу в цілому Загалом, зміст представленої роботи в повній мірі розкриває обрану тему. Дослідження, проведені в матеріалах є достатньо аргументованими. Прослідковуються високі теоретичні та практичні рівні у даному виконанні. Результатом проведення досліджень стали відповідні висновки і конкретні пропозиції щодо вдосконалення процесу прогнозування шляхом залучення ДСМ методу інтелектуального аналізу даних.

8. Інші зауваження _____

9. Оцінка дипломної роботи Робота заслуговує оцінки «добре», а її автор – присвоєння кваліфікації «магістра» з прикладної математики.

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи) Лисенко Сергій Миколайович, доктор технічних наук, доцент кафедри комп'ютерної інженерії та системного програмування ХНУ

“ ”


(підпис)

2020 р.

Завідувачу кафедри ТМІТ
д-р.техн.наук Підченку С.К.


ПІБ здобувача вищої освіти

ФПКТС, 2 курсу, групи ПМм-19-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

09.12.2020

дата


підпис

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ ТЕЛЕКОМУНІКАЦІЙ, МЕДІЙНИХ ТА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод прогнозування показників виробничих процесів на основі інтелектуального аналізу даних

Автор: Жовнір Марія Юріївна

Спеціальність: 113 – прикладна математика

Освітня програма: освітньо-професійна

Науковий керівник: Кисіль Тетяна Миколаївна, к.ф.-м.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	+
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) співпадіння розміщені в розділах, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення мають фрагментарний характер, або мають належним чином оформленні посилання;
- 3) більшість джерел запозичення дублюють одне одного.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 1,27% і адресується до 41 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Дата _____

Підченко С.К.

Підпис

Кисіль Т.М.

Підпис