МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В.Н.Каразіна
Факультет математики і інформатики
Кафедра теоретичної та прикладної інформатики

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В.Н.Каразіна
Факультет математики і інформатики
Кафедра теоретичної та прикладної інформатики

Кваліфікаційна робота

бакалавр

на тему: “ Формальний концептуальний аналіз за неповними даними ”

Виконав: студент 4 курсу, групи МФ-41
спеціальність 122 «Комп’ютерні науки»
освітньо-професійна програма «Інформатика»
Мустафа Р.А.

Керівник: Волков І.В
Рецензент:

_____
(прізвище та ініціали)

Харків 2023

Зміст

ВСТУП

Найбільш поширеним методом пошуку інформації в даний час є безсумнівно, використання так званих пошукових систем в Інтернеті. Це популярний сервіс, який дозволяє користувачеві знайти веб-сторінки, які містять потрібну йому інформацію про терміни. Принцип роботи пошукових систем в Інтернеті зараз очевидний практично кожній людині - по-перше, необхідно ввести ключові слова, що характеризують шукану інформацію, і пошукова система відразу ж видасть список посилань на основі своєї бази даних. Отже, мета пошукових систем в Інтернеті полягає в тому, щоб максимально наблизити результат пошуку до заданих критеріїв і таким чином надати користувачеві найбільш релевантну інформацію у відповідь на його запит. Для цього, однак, необхідно оцінювати вміст різними способами та релевантність веб-сайтів, які пошукові системи мають у своїй базі даних.

Ринок пошукових систем - це дуже складне економічне середовище, в якому окремі компанії постійно змушені впроваджувати різні технологічні та маркетингові інновації, тільки щоб утримати свої позиції на ринку або розширити свою частку ринку.

Сучасною тенденцією для максимально точного опису реальних явищ є міждисциплінарний підхід, який тягне за собою велику кількість змінних, що підлягають аналізу, які необхідно певним чином статистично обробити. Для цього пропонується широкий спектр методів, одним з яких є метод формального концептуального аналізу. Це сучасний метод аналізу даних в основі якого лежить ідея групування досліджуваних об'єктів за спільними для них характеристиками, який може бути успішно використаний для

генерування дослідницьких гіпотез у багатьох природничих науках. Цей метод має складне математичне підґрунтя, яке випливає з робіт німецького математика Рудольфа Вілле у 1980-х роках.

Цей метод дозволяє висувати припущення на основі даних, отриманих, наприклад, за допомогою анкетування або інструментальних вимірювань, так званий концептуальний пакет - структуру даних, яку користувачеві легко візуалізувати, а експерту легше знайти цікаві зв'язки в даних.

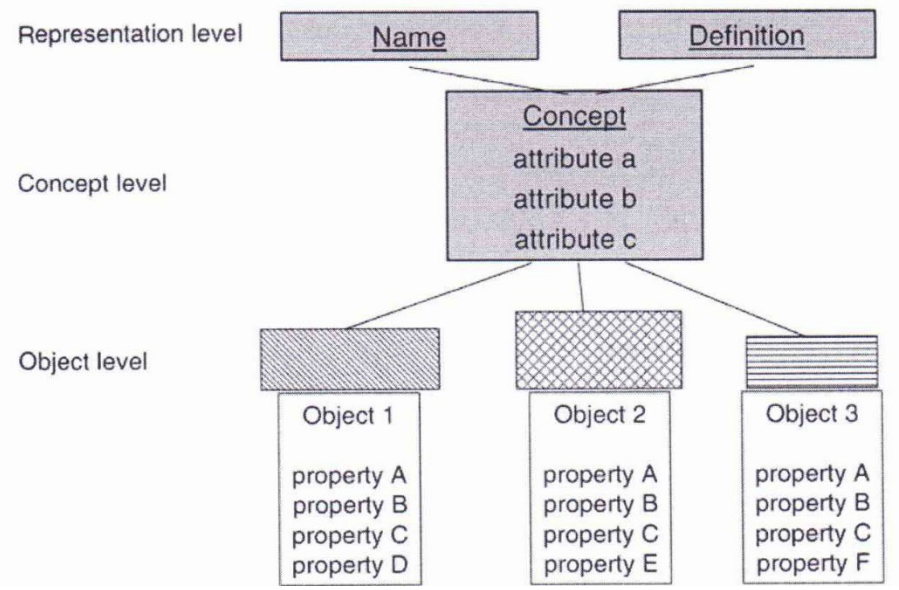
Метою дипломної роботи було дослідити проблеми неповних даних з точки зору ФКА та узагальнено кілька методів для роботи з такими даними.

Актуальність теми: Проблема неповноти даних стає все більш актуальною, оскільки потрібно обробляти все більші обсяги даних, а повні дані не завжди доступні. У реальних додатках неповні дані є поширеним явищем, і з ними потрібно працювати. У більшості випадків базові методи обробки даних призначені для повних даних, а неповні дані потребують попередньої обробки або обробки в зовсім інший спосіб. Попередня обробка зазвичай полягає у видаленні неповних записів або доповненні їх. Існує кілька методів доповнення даних, які оцінюють невідомі значення на основі відомих даних або за допомогою знань користувача. Не є винятком і формальний концептуальний аналіз (ФКА), основні методи якого працюють виключно з повними даними.

РОЗДІЛ 1. ФОРМАЛЬНИЙ КОНЦЕПТУАЛЬНИЙ АНАЛІЗ

Формальний концептуальний аналіз (ФКА) [1] як методологія аналізу даних, управління інформацією та представлення знань має потенціал для застосування до багатьох лінгвістичних проблем. ФСА - це математична теорія концептів та ієрархій концептів, яка відображає розуміння концепту і була встановлена в німецькому стандарті "DIN 2330 – Concepts and terms, general principles" [2]. Формальний концептуальний аналіз чітко формалізує розширення та зміст поняття, їхні взаємні зв'язки, а також той факт, що збільшення змісту означає зменшення обсягу, і навпаки. Заснований на теорії решіток, він дозволяє вивести ієрархію концептів із заданого набору даних.

Формальний аналіз концептів відрізняється від інших формалізмів представлення знань (RDF - Resource Description Framework) [3], OWL (Web Ontology Language) [4] або концептуальних графів [5]. Стандарт DIN 2330 [2] базується на різниці між ними. Він розрізняє три рівні: рівень об'єкта, рівень концепції та рівень представлення (мал. 1). Цей зв'язок (немає безпосереднього зв'язку між об'єктами та їхніми найменуваннями) забезпечується, скоріше, концепціями. На рівні концепту об'єкти, про які йдеться, є розширенням концепту, тоді як їхні спільні властивості становлять зміст концепту.



МАЛ 1. Рівень предметів, рівень концепції та рівень представлення відповідно до DIN 2330 [2]

Однак за останні 15 років формальний концептуальний аналіз перетворився на міжнародне дослідницьке співтовариство, яке застосовує його в багатьох дисциплінахю ФКА аналізує дані, які описують зв'язок між певним набором об'єктів і певним набором атрибутів. Такі дані часто зустрічаються в багатьох сферах людської діяльності. На основі вхідних даних ФКА виробляє два види вихідних даних. Перший - це концептуальна решітка. Концептуальна решітка - це набір формальних концептів у даних, які ієрархічно впорядковані відношенням підконцепт- суперконцепт.

Формальні концепти - це певні кластери, які представляють природні людські поняття, такі як "організм, що живе у воді", "автомобіль з повним приводом", "число, що ділиться на 3 і 4" і т.д. Другим результатом ФКА є набір так званих атрибутивних імплікацій. Атрибутивна імплікація описує певну залежність, яка діє в даних, наприклад, "кожне число, що ділиться на 3

і 4, ділиться на 6", "кожен респондент у віці старше 60 років є пенсіонером" і т.д.

Відмінною особливістю ФКА є інтеграція трьох компонентів концептуальної обробки даних і знань, а саме: виявлення і міркування з поняттями в даних, виявлення і міркування з залежностями в даних, а також візуалізація даних, понять і залежностей з можливостями згортання/розгортання. Інтеграція цих компонентів робить ФКА потужним інструментом, який застосовується для вирішення різних проблем. Приклади включають ієрархічну організацію результатів веб-пошуку в концепції, засновані на спільних темах, аналіз даних про генну експресію, пошук інформації, аналіз і розуміння програмного коду, налагодження, видобуток даних і проектування в програмній інженерії, інтернет-додатки, включаючи аналіз і організацію колекцій документів і електронної пошти, анотовані таксономії, а також інші різноманітні проекти з аналізу даних, описані в літературі.

1.1 Основи формального концептуального аналізу

Основним поняттям у ФКА є формальний контекст, який представляє вхідні дані. Методи ФКА працюють з простими бінарними даними, що складаються з набору об'єктів, набору атрибутів і відношення, яке визначає, чи має даний об'єкт даний атрибут. Формальним контекстом є трійка $\langle X, Y, I \rangle$, де X та Y - непорожні множини, а I - бінарне відношення між цими множинами ($I \subseteq X \times Y$). У попередньому визначенні X інтерпретується як множина об'єктів, Y як множина атрибутів, а $\langle x, y \rangle \in I$ має значення “об'єкт x має атрибут y”.

Визначення 1.1 (формальний контекст)

Формальний контекст можна представити у вигляді таблиці, де заголовки рядків - це окремі об'єкти, а заголовки стовпців - окремі атрибути. У певному полі таблиці стоїть або хрестик (об'єкт має відповідний атрибут), або порожнє поле (об'єкт не має відповідного атрибуту). Порядок рядків і стовпців не має значення. Якщо поміняти місцями деякі рядки/стовпчики в таблиці, що відповідають певному формальному контексту, вона все одно буде представляти той самий формальний 5 контекст.

Кожен формальний контекст можна представити у вигляді таблиці. Таблиця являє собою чіткий запис простих двійкових даних, які проаналізуються за допомогою ФКА. Таким чином, можна використовувати поняття таблиці (відповідно до формального контексту) і знати, що вона є представленням формального контексту. Таблиця також більше підходить для графічного представлення і для людської уяви.

I	y_1	y_2	y_3	y_4	y_5
x_1		X		X	X
x_2	X		X	X	X
x_3	X			X	
x_4	X	X			X
x_5		X	X	X	

Табл 1: Представлення формального контексту у вигляді таблиці

Надалі завжди буде розглядатися формальний контекст C = ⟨X, Y, I⟩ і відповідне йому концептуальне об'єднання ⟨B(X, Y, I), ≤⟩, якщо не вказано інше. Далі представлені основні оператори ФКА.

Визначення 1.2 (стрілочні оператори).

Для множини A ⊆ X та B ⊆ Y визначимо:

$A^{\uparrow_I}$ = {y ∈ Y | для всіх x ∈ A вірно ⟨x, y⟩ ∈ I},

$B^{\downarrow_I}$= {x ∈ Y | для всіх y ∈ B вірно ⟨x, y⟩ ∈ I}.

Оператори $\uparrow_I$, $\downarrow_I$ є так званими стрілочними операторами. $\uparrow_I$, $\downarrow_I$. Оператор $\uparrow_I$ присвоює множині вхідних об'єктів множину всіх атрибутів, які є спільними для цих об'єктів. Аналогічно, оператор $\downarrow_I$ присвоює множині вхідних атрибутів множину всіх об'єктів, які мають спільні атрибути.

Математичні основи формального концептуального аналізу ґрунтуються, головним чином, на зв'язках Галуа, операторах замикання та відповідних системах замикання.

Визначення 1.3 (нерухомі точки з'єднань Галуа).

Для з'єднань Галуа ⟨↑, ↓⟩ визначимо множину нерухомих точок як

$$\text{fix}(\langle\uparrow, \downarrow\rangle) = \{\langle A, B\rangle \in 2^X \times 2^Y \mid A^{\uparrow} = B, B^{\downarrow} = A\}.$$

Ще однією відомою властивістю з'єднань Галуа є так звана ланцюгованість. Ця властивість дозволяє нам певним чином “скоротити” багаторазове застосування зв'язків Галуа. Ми можемо формалізувати ланцюгове зчеплення наступним чином.

Теза 1.4

Для з'єднань Галуа ⟨↑, ↓ ⟩ та довільних A ⊆ X, B ⊆ Y виконується наступне:

$$A^{\uparrow} = A^{\uparrow\downarrow\uparrow}, B^{\downarrow} = B^{\downarrow\uparrow\downarrow}$$

Поняття формального концепту є одним з базових понять ФКА. На питання, що таке концепт, можна відповісти кількома способами. У нашому випадку ми відштовхуємося від поняття логіки Порт-Рояля, де поняття визначається екстенсією та інтенцією. Обсяг - це множина об'єктів, які включає дане поняття. Інтенція - це набір атрибутів, які включає в себе дане поняття.

Визначення 1.5 (формальне поняття).

Формальним поняттям над ⟨X, Y, I⟩ називається пара ⟨A, B ⟩ така, що A ⊆ X, B ⊆ Y така, що $^{\uparrow\downarrow}A = B$, $B = A$.

Множина об'єктів A називається екстенсією, множина атрибутів B - інтенцією.

Словесно, ⟨A, B ⟩ є формальним поняттям, коли A - це множина всіх об'єктів, які мають всі атрибути з B, а B - це множина всіх атрибутів, які мають всі об'єкти з A. Варто зазначити, що не кожна підмножина об'єктів є екстенсіоналом (аналогічно для атрибутів). Геометрична інтерпретація формального поняття також важлива. З цієї точки зору, поняття - це просто максимальні прямокутники в таблиці, що представляють заданий формальний контекст, з довільно перемішаними рядками і стовпчиками. Геометрична інтерпретація зручна для людської уяви, і часто краще думати про поняття як

про максимальні прямокутники. З точки зору операторів ↑,↓ ⟨A, B⟩ є формальним поняттям саме тоді, коли ⟨A, B⟩ є фіксованою точкою ⟨↑,↓⟩.

Природною вимогою до концептуального об'єднання є можливість його інтерпретації як об'єднання відповідно до спільності понять. Якщо ця вимога виконується, то $\langle A_1, B_1 \rangle \leq \langle A_2, B_2 \rangle$ означає, що $\langle A_1, B_1 \rangle$ є більш конкретним, ніж $\langle A_2, B_2 \rangle$.

Визначення 1.6 (впорядкування формальних понять).

Для формальних понять $\langle A_1, B_1 \rangle$, $\langle A_2, B_2 \rangle$ визначимо

$\langle A_1, B_1 \rangle \leq \langle A_2, B_2 \rangle$, тільки коли A ⊆ A(еквівалентно, B ⊆ B).

Сукупність усіх формальних понять, що виходять за межі певного формального контексту, разом із визначеним вище розташуванням утворюють так звану концептуальну єдність.

Визначення 1.7 (концептуальне об'єднання).

Позначимо через B(X, Y, I) множину всіх формальних понять над ⟨X, Y, I⟩, а множину ⟨B(X, Y, I), ≤⟩ назвемо понятійним об'єднанням, що відповідає формальному контексту ⟨X, Y, I⟩.

Концептуальне об'єднання ⟨B(X, Y, I), ≤⟩ представляє набір усіх кластерів даних (формальних понять) у формальному контексті ⟨X, Y, I⟩.

Теорема 1.8 (складений вигляд ↑↓, ↓↑).

Складений вигляд ↑↓ є оператором замикання на X (аналогічно для складеного вигляду ↓↑), тобто він є дійсним:

1. $A \subseteq A^{\uparrow\downarrow}$ - безпосередньо з означення кон'юнкцій Галуа
2. $A_1 \subseteq A_2 \Rightarrow A_2^{\uparrow} \subseteq A_1^{\uparrow} \Rightarrow A_1^{\uparrow\downarrow} \subseteq A_2^{\uparrow}$
3. $A^{\uparrow\downarrow} = A^{\uparrow\downarrow\uparrow\downarrow}$

Теорема 1.9 (основна теорема про концептуальні об'єднання).

1. ⟨B(X, Y, I), ≤⟩ - повне об'єднання, де вершина та супрема задаються

$$\bigwedge_{j \in J} \langle A_j, B_j \rangle = \langle \bigcap_{j \in J} A_j \, , \left(\bigcup_{j \in J} B_j \right)^{\downarrow I \uparrow I} \rangle \, ,$$

$$\bigvee_{j \in J} \langle A_j, B_j \rangle = \langle \left(\bigcup_{j \in J} A_j \right)^{\downarrow I \uparrow I} \, , \bigcap_{j \in J} B_j \rangle$$

2. Довільне повне об'єднання V = ⟨V, ≤⟩ ізоморфне ⟨B(X, Y, I), ≤⟩, якщо існують відображення γ_1 : X'→ V, μ_1 : Y'→ V такі, що вони виконуються:

   $\gamma_1(x) \leq \mu_1(y)$, тільки коли ⟨x, y⟩ ∈ I.

Одним з корисних наслідків того, що, складаючи оператори стрілок, отримується оператор замикання, є можливість просто виразити найменшу екстенцію(інтенцію), що оточує задану множину об'єктів (атрибутів).

Теорема 1.10.

Нехай $A \subseteq X$. Нехай $A^{\uparrow\downarrow}$ - найменша экстенція, що містить A.

Нехай $B \subseteq Y$. Нехай $B^{\downarrow\uparrow}$ - найменша інтенція, що містить B.

Доведення. Якщо C така що $A \subseteq C$, то з монотонності оператора $\uparrow\downarrow$, $A^{\uparrow\downarrow} \subseteq C^{\uparrow\downarrow}$.

Аналогічно, якщо D така, що $B \subseteq D$, то з монотонності оператора замикання, $B^{\downarrow\uparrow} \subseteq D^{\downarrow\uparrow}$.

1.2. Прояснення та скорочення формального контексту

Наприклад, формальне визначення контексту (1.1) допускає наявність декількох об'єктів з однаковим набором атрибутів. У таблиці цей випадок можна уявити як два однакові рядки (за винятком заголовка). Таке ж спостереження можна зробити і для атрибутів. Питання в тому, як така ситуація впливає на відповідне концептуальне об'єднання.

Визначення 1.11 (уточнений формальний контекст).

Ми називаємо формальний контекст $\langle X, Y, I\rangle$ з'ясованим, якщо таблиця, що відповідає цьому контексту, не містить однакових рядків або стовпців.

Ми називаємо процес видалення дублікатів рядків і стовпців з відповідної таблиці уточненням формального контексту. В результаті

уточнення створюється нова таблиця, яка відповідає уточненому формальному контексту.

1.3. Багатозначні формальні контексти

Розширюючи базове визначення формального контексту, можна визначити так звані багатозначні формальні контексти. Їх можна уявити як таблиці, які можуть мати значення, відмінні від хрестика в одному полі.

Визначення 1.12 (багатозначний формальний контекст).

Багатозначний формальний контекст – це четвірка $\langle X, Y, V, I\rangle$, де X - непорожня множина об'єктів, Y - непорожня множина атрибутів, V - множина всіх значень атрибутів з Y , а I - відношення між трьома множинами ($I \subseteq X \times Y \times V$), таке що

$(\langle x, y, v\rangle \in I) \wedge (\langle x, y, w\rangle \in I) \Longrightarrow v = w.$

Багатозначний формальний контекст дозволяє працювати з багатозначними атрибутами. Ми розглядаємо не тільки те, чи має об'єкт певний атрибут, але й те, яке значення має відповідний атрибут. Природним обмеженням багатозначного формального контексту є вимога, щоб кожній парі об'єкт-атрибут було присвоєно не більше одного допустимого значення.

Аналогічно, як і у випадку бінарних формальних контекстів, багатозначний формальний контекст $\langle X, Y, V, I\rangle$ може бути представлений таблицею, де заголовки рядків є окремими об'єктами з X, а заголовки стовпців - окремими

атрибутами з Y. Різниця полягає у значеннях в кожному полі таблиці. У цьому випадку поле таблиці, задане об'єктом x та атрибутом y, має значення v ∈ V, якщо ⟨x, y, v⟩ ∈ V, інакше порожнє.

Для багатозначних формальних контекстів доречно ввести поняття атрибутного домену. Це поняття є аналогом активного домену з моделі реляційної бази даних.

1.4. Наслідки атрибутів

Атрибутивні імплікації - це вирази, які описують певну залежність між атрибутами у вхідних даних. Аналогії атрибутивних імплікацій можна знайти в багатьох областях, наприклад, в області реляційних баз даних - це функціональні залежності, в області інтелектуального аналізу даних - правила асоціацій. Спочатку визначимо саме поняття атрибутивної імплікації.

Визначення 1.13 (імплікація атрибута).

Нехай Y - непорожня множина атрибутів. Імплікацією атрибута називається довільний вираз $A \Rightarrow B$, де $A \subseteq Y$, $B \subseteq Y$.

Тепер, коли імплікація атрибута визначена, нам потрібно визначити її валідність. Достовірність імплікації атрибута завжди пов'язана з конкретними даними. Нам потрібна деяка базова семантична структура, за допомогою якої можемо оцінювати істинність імплікацій атрибутів. Такою базовою семантичною структурою ми обираємо рядки формального контексту. Рядок формального контексту можна розглядати як характеристику окремого

об'єкта шляхом перерахування його атрибутів. Наша мета - визначити істинність імплікацій атрибутів у формальному контексті.

Визначення 1.14 (допустимість імплікації атрибутів).

Атрибутивна імплікація A ⇒ B над Y дійсна на множині M ⊆ Y , тільки коли

(A ⊆ M) =⇒ (B ⊆ M).

Атрибутивна імплікація A ⇒ B над Y дійсна у формальному контексті ⟨X, Y, I⟩ тільки тоді, коли A ⇒ B дійсна для кожного {x} , де x ∈ X.

Імплікація атрибута є дійсною у формальному контексті лише тоді, коли вона є дійсною у всіх рядках таблиці, що відносяться до цього формального контексту. Якщо імплікація атрибутів A ⇒ B є дійсною у формальному контексті, то будь-який об'єкт, який має всі атрибути з A, також має всі атрибути з B.

Примітка 1.15.

Атрибутивні імплікації можна розглядати як скорочення для певних пропозиційних формул. Нехай C = ⟨X, Y, I⟩ є формальним контекстом, тоді кожній парі (x, y), x ∈ X, y ∈ Y можна поставити у відповідність пропозиційний символ, значення якого задається I(x, y) (істина, якщо ⟨x, y⟩ ∈ I, хиба в іншому випадку). Тоді істинність атрибутивної імплікації A ⇒ B у формальному контексті C задається формулою

$$\bigwedge_{x \in X} \left(\bigwedge_{y \in A} I(x,y) \Rightarrow \bigwedge_{z \in B} I(x,z) \right)$$

Тепер введемо аналогію понять теорії та моделі з математичної логіки.

Визначення 1.16 (теорія).

Теорія над Y - це будь-яка множина імплікацій атрибутів над Y.

Визначення 1.17 (теорія моделей).

Моделлю теорії T над Y називається будь-яка множина $M \subseteq Y$ така, що кожна імплікація атрибутів $A \Rightarrow B \in T$ є дійсною в M. Позначимо множину всіх моделей теорії T через Mod(T).

Озброївшись поняттями теорії та моделі, ми можемо визначити семантичну обумовленість.

Визначення 1.18 (семантичний наслідок).

Атрибутивна імплікація $A \Rightarrow B$ семантично випливає з теорії T (тобто $T \models A \Rightarrow B$), так само як $A \Rightarrow B$ справедлива в кожній моделі теорії T.

Деякі атрибутивні імплікації випливають з інших або є тривіально правильними. Наприклад, атрибутивна імплікація $A \Rightarrow A$ завжди тривіально виконується. Іншим прикладом є атрибутивна імплікація $A \Rightarrow C$, яка безумовно випливає з теорії $\{A \Rightarrow B, B \Rightarrow C\}$. Питання полягає в тому, чи існує система правил, яка б дозволила перевірити, чи випливає дана атрибутивна імплікація з даної теорії. Такі системи існують, і ми наведемо тут найвідомішу з них, так звану аксіоматичну систему Армстронга.

Визначення 1.19 (аксіоматична система Армстронга).

Аксіоматична система Армстронга складається з правил виведення

$$(Ax) \qquad \frac{}{A \cup B \ \Rightarrow A'} \qquad (Cut) \qquad \frac{A \Rightarrow B,\ B \cup C \ \Rightarrow D}{A \cup C \ \Rightarrow D}$$

Правило виведення (Ax) дозволяє виводити атрибутивні імплікації з порожнього набору припущень. Для атрибутивних імплікацій, отриманих у такий спосіб, завжди має місце випадок, коли права частина є підмножиною лівої частини. Друге введене правило виведення (Cut), або правило відсікання, вже має непорожній набір припущень. Правило Cut тому, що ми складаємо дві імплікації, з яких ми “вирізаємо” спільну частину B.

Для того, щоб вивести атрибутивні імплікації з представленої аксіоматичної системи, необхідно формально ввести поняття доведення. Під доведенням ми розуміємо зв'язну послідовність атрибутивних імплікацій, яка задовольняє певним властивостям.

РОЗДІЛ 2: ОГЛЯД ОКРЕМИХ ПІДХОДІВ ДО НЕПОВНИХ ДАНИХ У ФКА

Формальний концептуальний аналіз, як було представлено в першому розділі, опрацьовує повні дані. Це означає, що якщо ми хочемо опрацювати неповні дані, ми повинні поводитися з ними по-особливому. Одним із таких способів є їх попередня обробка. Найпростіший спосіб - видалити неповні записи (об'єкти/атрибути) і залишити тільки повні дані, які ми можемо обробити. Однак, видаляючи неповні записи, ми втрачаємо інформацію, яку містили дані. Інший спосіб попередньої обробки - зробити неповні дані повними. Це передбачає заповнення відсутніх значень. Питання полягає в тому, як визначити значення, які потрібно заповнити. Наприклад, ми можемо використати деяке самопізнання або відому залежність у даних. Попередня обробка неповних даних - не єдиний спосіб роботи з такими даними. Існують розширення методів ФКА, які можуть працювати безпосередньо з неповними даними без попередньої обробки. У цьому розділі я представлю деякі вибрані підходи д о неповних даних у ФКА, зокрема, методи уточнення.

2.1. Основні поняття

Неповний формальний контекст можна розглядати як окремий випадок вже визначеного багатозначного формального контексту.

Визначення 2.1 (неповний формальний контекст з сеансом).

Неповний формальний контекст $\langle X, Y, V, I\rangle$ - це багатозначний формальний контекст, де $V = \{\times, ?\}$.
Нехай $x \in X$, $y \in Y$, $v \in V$, тоді $\langle x, y, v\rangle \in I$ має значення:

- x має атрибут y, якщо $v = \times$,
- невідомо, чи має x атрибут y, якщо $v = ?$,

• інакше x не має атрибута y.

Неповний формальний контекст може бути представлений таблицею, як і у випадку бінарного формального контексту, але ми допускаємо в полях таблиці не тільки хрестик і порожнє поле, а й значення ?.

У деяких ситуаціях зручніше розглядати $I \subseteq X \times Y \times V$ аніж $I : X \times Y \text{ '}\rightarrow \{\times, -, ?\}$.

Визначення 2.2 (неповний формальний контекст з відображенням).

Нехай $\langle X, Y, V, I\rangle$ - багатозначний формальний контекст. Якщо I є представленням
$I : X \times Y \text{ '}\rightarrow \{\times, -, ?\}$, тоді $I(x, y) = v$ має сенс:

• x має атрибут y, якщо $v = \times$,

• x не має атрибуту y, якщо $v = -$,

• невідомо, чи має x атрибут y, якщо $v = ?$.

Два наведені визначення неповного формального контексту є еквівалентними з точки зору інформативності. Неповний формальний контекст за одним визначенням може бути перетворений у відповідний неповний формальний контекст за іншим визначенням без втрати інформації.

Разом із введенням неповних формальних контекстів доречно запровадити ще одне впорядкування, цього разу щодо неповних контекстів.

Визначення 2.3 (впорядкування неповних формальних контекстів).

Нехай $C_1 = \langle X, Y, \{\times, -, ?\}, I_1 \rangle$, $C_2 = \langle X, Y, \{\times, -, ?\}, I_2 \rangle$ - неповні контексти. Якщо C_2 утворено з C_1 шляхом заміни деяких знаків питання на інші значення (×, -), то C_2 містить більше інформації, ніж C_1 . Ми записуємо цей факт як $C_1 \leq C_2$.

Ми інтерпретуємо це усталене розташування неповних формальних контекстів як розташування відповідно до інформаційного наповнення.

Визначення 2.4 (завершення неповного формального контексту). Нехай $C_1 = \langle X, Y, \{\times, -, ?\}, I_1 \rangle$ - неповний формальний контекст. Завершенням неповного формального контексту $C_1 = \langle X, Y, \{\times, -, ?\}, I_1 \rangle$ є формальний контекст C, який створюється з C_1 шляхом заміни всіх знаків питання на інші значення (заміна значення "-" може бути інтерпретована як видалення даного кортежу з сеансу).

Доповнюючи контекст, ми отримуємо максимальні контексти по відношенню до встановленого порядку в інформації, а отриманий формальний контекст не містить жодного знаку питання, тому його можна розглядати як класичний бінарний формальний контекст.

Далі ми введемо поняття визначеного та можливого інтенту (екстенту) над неповним формальним контекстом $C = \langle X, Y, \{\times, -, ?\}, I_1$. Певний інтент множини $A \subseteq X$ (позначається A) - це множина всіх атрибутів, які обов'язково повинні мати всі об'єкти A. На противагу цьому, можливий намір множини A - це множина всіх атрибутів, які можуть мати всі об'єкти з A. Аналогічно, для певного та можливого екстенту.

2.2. Наповнення за допомогою дослідницького аналізу

Дослідження атрибутів, або розвідувальний аналіз, - це метод ФКА для вилучення знань з неповних даних. Зокрема, дослідницький аналіз працює наступним чином. Користувачам послідовно ставлять запитання про достовірність певних імплікацій атрибутів. Якщо користувач надає відповідь на всі поставлені запитання, результатом дослідницького аналізу є база даних усіх правильних імплікацій атрибутів у даних, а також, для кожної неправильної імплікації атрибута, принаймні один контрприклад. Однак може статися так, що користувач не знає відповіді на всі поставлені запити. У цьому випадку на виході отримується множина всіх допустимих імплікацій атрибутів, множина імплікацій атрибутів, які можуть бути допустимими, і множина згенерованих контрприкладів імплікацій атрибутів, для яких користувач не знав відповіді на запит про їх допустимість. На основі отриманих імплікацій атрибутів неповні дані можна зробити повними.

Існує кілька типів алгоритмів пошукового аналізу. Зокрема, вони відрізняються тим, чи дозволяють користувачеві відповісти “не знаю” на поставлене запитання. Іншими словами, один тип алгоритмів працює з неповними даними достовірних імплікацій атрибутів, тоді як інший тип працює лише з повними даними достовірних імплікацій атрибутів. Тут буде представлено тип, що працює з неповними даними допустимих імплікацій атрибутів.

2.2.1. Розширення імплікацій атрибутів

У першому розділі ми ввели поняття імплікації атрибута (1.13) і пов'язане з ним поняття достовірності імплікації атрибута у формальному контексті (1.14). Тепер нам потрібно ввести аналог поняття валідності імплікації атрибута, цього разу у неповному формальному контексті. Ми також вводимо нове поняття задовільності.

Визначення 2.6 (обґрунтованість та задовільність імплікацій атрибутів). Нехай $C = \langle X, Y, \{\times, -, ?\}, I\rangle$ - неповний формальний контекст. Атрибутивна імплікація $A \Rightarrow B$ над Y має вигляд

• дійсним, тільки тоді, коли він дійсний у всіх завершеннях C, тобто

$$\forall x \in X : (A \subseteq x^1) \rightarrow (B \setminus A \subseteq x^2);$$

• виконується, якщо вона дійсна хоча б в одному з C-доповнень, тобто

$$\forall x \in X : (A \subseteq x^2) \rightarrow (B \setminus A \subseteq x^1).$$

Валідність імплікації атрибута в неповному формальному контексті, безумовно, передбачає її задовільність. Кожна валідна імплікація атрибута в даному неповному формальному контексті є задовільною. Однак, зворотне твердження не є вірним.

2.2.2. Застосування тризначної логіки Клені

Як ми знаємо з першого розділу (примітка 1.15), атрибутивні імплікації можна розглядати як скорочення для запису певних формул логіки

висловлювань. Більше того, атрибутивна імплікація є дійсною у певному формальному контексті саме тоді, коли відповідна формула логіки висловлювань є дійсною. Тому доречно поширити цей принцип на неповні контексти. Оскільки ми маємо три можливих значення атрибутів замість двох, пропонується можливість використання тризначної логіки. Логіка, яка дуже близька до введеного семантичного визначення достовірності та задовільності імплікацій атрибутів, є тризначною логікою Клені. Таблиці істинності логічних кон'юнкцій цієї логіки наведено в таблиці 1.

Однак оцінка формул у тризначній логіці Клені не повністю відповідає визначенню істинності та задовільності атрибутивних імплікацій. Наприклад, атрибутивна імплікація {y} ⇒ {y} завжди є тривіально дійсною, але якщо значення y для деякого об'єкта дорівнює “?”, то значенням істинності формули, що відповідає даній атрибутивній імплікації, є “?”. Це є наслідком різного значення у в неповних формальних контекстах та тризначній логіці Клені. У неповних формальних контекстах значення ? інтерпретується як брак інформації і є лише проксі для одного зі значень “-” або “×”. На відміну від інших значень, значення ? не має відношення до реальної області атрибутів вхідних даних.

⋀	-	?	x
-	-	-	-
?	-	?	?
x	-	?	x
⇒	-	?	x
-	x	x	x
?	?	?	x

x	-	?	x

¬	
-	x
?	?
x	-

Таблиця 1: Таблиці істинності тризначних логічних кон'юнкцій Клені.

Якщо пропозиційна формула, що відповідає атрибутивній імплікації $A \Rightarrow B$, містить кожну пропозиційну змінну не більше одного разу ($A \cap B = \emptyset$), то тризначна логіка Клені відповідає введеному семантичному визначенню валідності та задовільності атрибутивних імплікацій. Оскільки атрибутивна імплікація $A \Rightarrow B$ є валідною, коли $A \Rightarrow (B \setminus A)$ є валідною, ми можемо використовувати попередню логіку для обчислення валідності будь-якої атрибутивної імплікації у неповному формальному контексті. Однак тризначної логіки Клені недостатньо для обчислення істинності довільного набору атрибутивних імплікацій, оскільки пропозиціональна формула, що відповідає набору атрибутивних імплікацій, неминуче може містити деяку пропозиціональну змінну більше одного разу.

2.2.3. Застосування модальної логіки

Модальна логіка розширює логіку предикатів операторами, що виражають можливість і необхідність. Таким чином, можна формалізувати такі твердження, як “Можливо, що сьогодні піде дощ”. Семантика модальної

логіки базується на так званих можливих світах, де можливі світи можуть бути досяжними один з одного.

У нашому випадку ми визначаємо можливі світи як усі неповні контексти над заданим набором атрибутів Y, а також усі підмножини Y (що представляють можливі об'єктні наміри). З неповного формального контексту C = ⟨X, Y, {×, -, ?}, I⟩ досяжні всі його завершення та всі об'єктні наміри цих завершень. Нічого іншого з набору атрибутів не є досяжним.

Множина атрибутивних імплікацій A є істинною тоді і тільки тоді, коли істинним є твердження “Необхідно, щоб φ”, де φ - формула, що відповідає A, яка є кон'юнкцією формул, що відповідають кожній атрибутивній імплікації, як описано в примітці 1.15. Значення істинності A еквівалентне значенню істинності твердження “Можливо, що φ”, де φ - та сама формула, що і в попередньому випадку.

Представлена модальна логіка більше не страждає від недоліків. Таблиці істинності основних логічних сполучників введеної модальної логіки можна знайти в таблиці 2. Інші логічні зв'язки визначаються так само, як і в клаузульній логіці висловлювань.

∧	-	?	x
-	-	-	-
?	-	?	?
x	-	?	X

⇒	-	?	x
-	x	x	x
?	?	?	x
X	-	?	x

¬	
-	x
?	?
x	-

Табл 2: Таблиці істинності кон'юнкцій модальної логіки для формул над атрибутами в неповному формальному контексті.

У наведених тут таблицях істинності логічних кон'юнкцій модальної логіки третє значення істинності є також там, де тризначна логіка Клені дає одне з двох класичних значень істинності. Це пов'язано саме з інтерпретацією третього значення істини як “нісенітниці”. Іншими словами, якщо висловлювання містить нонсенс, то все висловлювання завжди є нонсенсом.

2.2.4. Структура алгоритму розвідувального аналізу

Тепер ми можемо представити основну структуру алгоритму дослідницького аналізу, як він був спочатку представлений за допомогою тризначної логіки Клені.

• Спочатку користувач вводить неповний формальний контекст (може містити знаки питання), який ми позначаємо як поточний. Далі він може вказати деякі власні знання у вигляді імплікацій атрибутів, які називаються фоновими знаннями, про універсум вхідних даних. Множина отриманих атрибутивних імплікацій ініціалізується порожньою множиною (спочатку жодної імплікації не отримано).

На j-му проміжному кроці позначимо поточний контекст як C_j . Алгоритм вибирає атрибутивну імплікацію $A \Rightarrow B$, яка є задовільною. Де A - найменша множина, в якій допустимі всі прийняті до цього часу імплікації атрибутів, а $B = \{y \in Y \mid A \Rightarrow \{y\}$ задовольняється в $C_j\}$. Якщо такої імплікації атрибутів не існує, алгоритм завершує роботу. Таким чином, меню B містить всі атрибути y, для яких імплікація атрибута $A \Rightarrow \{y\}$ є задовільною у поточному контексті. Якщо $A \Rightarrow B$ виводиться (за допомогою аксіоматичної системи Армстронга) з набору імплікацій атрибутів, отриманих до цього часу, і заданого самопізнання, то воно автоматично призначається. В іншому випадку користувача запитують, чи є атрибутивна імплікація $A \Rightarrow B$ допустимою у даному універсумі.

• Якщо користувач відповідає “так”, то імплікація атрибута $A \Rightarrow B$ додана до набору прийнятих імплікацій атрибутів.

• Якщо користувач відповідає “ні”, він повинен вказати об'єкт (рядок таблиці), який є контрприкладом імплікації атрибута $A \Rightarrow B$. Новий рядок може мати вигляд щоб знову звернутися до об'єкта зі знаками питання. Об'єкт, що відповідає такому рядку, додається до поточного контексту.

- Якщо користувач відповідає “не знаю”, то він повинен вказати, для якого саме b ∈ B справедливість імплікації атрибута A ⇒ {b} невідома. Позначимо через Z_j = {b ∈ B | істинність A ⇒ {b} невідома}. Для кожного атрибуту b ∈ Z_j алгоритм створює уявний об'єкт, який є найменшим (з точки зору впорядкування інформації) контрприкладом для імплікації A ⇒ {b}. Такий об'єкт має всі атрибути з A, не має атрибуту b, а решта атрибутів невідомі. Якщо B \ Z_j != A, то до множини допустимих імплікацій атрибутів додається імплікація A ⇒ B \ Z_j.

Після завершення алгоритму пошукового аналізу можуть знадобитися додаткові дії. Однією з таких дій, наприклад, є видалення надлишкових уявних об'єктів. Дійсно, може статися так, що з отриманого набору прийнятих атрибутивних імплікацій випливає атрибутивна імплікація A ⇒ {b}, яка, як відомо, не є достовірною. У такому випадку необхідно вилучити уявний об'єкт, який було додано як контрприклад для цієї атрибутивної імплікації. Крім того, можна видалити надлишкові уявні контрприклади для імплікацій атрибутів, для яких у результуючому контексті існує реальний контрприклад.

Залишається питання, як заповнити неповний формальний контекст на основі отриманих атрибутивних імплікацій. Позначимо через T множину атрибутивних імплікацій, які є допустимими при заповненні неповного контексту C = ⟨X, Y, {×, -, ?}, I⟩. Можна замінити значення ? що відповідає об'єкту x ∈ X та атрибуту y ∈ Y, на значення

- “×” - якщо існує A ⇒ B ∈ T таке, що A ⊆ x^1 та y ∈ B;
- “-” - якщо існує A ⇒ B ∈ T таке, що x ∈ A, A \ {x} ⊆ x^1 і B !⊆ x^2

2.3. Заповнення з використанням матричної факторизації

Метод доповнення неповного формального контексту за допомогою матричної факторизації було представлено в роботі Піскової, Пер, Хорвата та Крайча [6], на якій я базувався в цьому розділі.

Спочатку згадаємо поняття матриці заданого типу та множини всіх матриць заданого типу, зокрема для уточнення позначень.

Визначення 2.7 (матриця).

Нехай T - числове тіло і m, n ∈ N. Тоді представлення
A : {1, 2, . . . , m} × {1, 2, ... , n} → T називається матрицею m × n, яка записується у вигляді прямокутної схеми

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

де a_{ij} = A(i, j). Позначимо множину всіх матриць типу m × n над числовим тілом T через $M_{m\times n}$ (T).

2.3.1. Матрична факторизація

Факторизація матриць - часто використовуваний метод в інтелектуальному аналізі даних. Він може бути використаний для

розкладання матриці на менші матриці, які називаються факторами, і які, будучи складеними, дають наближення до вихідної матриці. Існує кілька методів факторизації матриць.

Одним з них є стохастична градієнтна факторизація.

Результатом цього методу для матриці C ∈ M|X|×|Y| (R) є пара матриць W ∈ M (R), H ∈ M|Y|×K (R) таких, що $WH^T = C'$ апроксимує матрицю C, позначимо $C \approx C' = WH^T$. Число K вказує на кількість факторів. Нехай $x_{ij} \in C$ - невідома величина. Оцінкою x_{ij} називається x'_{ij}, яка задається $x'^T_{ij} = (WH)$

Іншими словами, оцінка невідомого значення в C - це значення в цій позиції в обчисленій апроксимації C'. Тепер нам потрібно з'ясувати, як отримати C', яка добре оцінює невідомі значення. Для цього C розбиваємо на дві взаємодоповнюючі частини C^{train}, C^{test}, тобто $C^{train} \subseteq C$, $C^{test} = C \setminus C^{train}$. Позначимо множину C^{train} як навчальну, а C^{test} - як тестову. Наближення C' визначається лише за відомими значеннями з C^{train}, а точність цього наближення задається середньоквадратичним відхиленням по матриці C^{test}, яке задається формулою

$$s\,(C^{test}) = \sqrt{\frac{\sum_{xij} \in\ C^{test}\ (x_{ij}\ -\ x'_{ij})^2}{|C^{test}|}},$$

де x_{ij} - це тільки відомі значення з C^{test}. Ми включаємо лише відомі значення, оскільки для невідомих значень ми не маємо можливості обчислити зміщення оцінки. Мета полягає в тому, щоб знайти таке значення C' (або W,

H), для якого середньоквадратичне відхилення є мінімальним. Для цього використовується метод стохастичного градієнта, який шукає мінімум функції

$$\sum_{xij \in C^{train}} (x_{ij} - x'_{ij})^2 + \lambda \left(\|\omega_i\|^2 + \|h_j\|^2 \right),$$

де w_i - i-й рядок матриці W, h_j - j-й рядок матриці H, λ - параметр, який запобігає так званому перенавчанню. Перенавчання - це явище, яке виникає, коли ми можемо дуже точно оцінити дані на навчальній вибірці, але оцінка на тестовій вибірці не вдається.

2.3.2. Пошук правильної факторизації

На етапі пошуку відповідного розкладу матриці C матриці W, H спочатку ініціалізуються до випадкових значень. Потім, відповідно до обчисленого відхилення, їхні елементи поетапно коригуються. На ці коригування впливає ще один необов'язковий параметр. Таким чином, загалом можна вибрати декілька параметрів факторизації.

При спробі отримати відповідні W, H може виникнути ситуація, коли елементи C^{train} і C^{test} не мають жодних схожих ознак. Щоб уникнути цього явища, ми можемо використати так звану n-кратну перехресну перевірку, коли ми ділимо дані на n рівних за розміром частин. Ми обираємо набір параметрів факторизації і для кожної виділеної частини обчислюємо факторизацію її доповнення за допомогою попереднього методу та обчислюємо стандартне відхилення для цієї частини. Отримане стандартне

відхилення для C' із заданими параметрами навчання є середнім значенням усіх стандартних відхилень частин. Спробувавши кілька наборів параметрів факторизації, ми обираємо той, який має найменше середньоквадратичне відхилення.

2.3.3. Експерименти

У [6] можна знайти три експерименти з таким наповненням. Дані, що використовувалися, були обрані з репозиторію машинного навчання UCI [7], і були проведені тести з різним відсотком невідомих значень (10%, 20%, 30%). Середній показник успішності в цих експериментах склав близько 55%. Цікаво, що в двох з трьох проведених експериментів відсоток успішності був майже однаковим, незалежно від відсотка невідомих даних. Оскільки це експериментальний метод, автори статті не коментують цей факт.

2.4. Інші роботи

Питання неповних даних все ще розробляється, і вже існує кілька методів їх обробки за допомогою ФКА. У попередніх розділах я детально обговорив два обрані підходи. У цьому розділі я кількома реченнями узагальню зміст інших робіт, які я вивчив у цій галузі.

У роботі Ліа та Яо [8] розвинуто ідею масштабування неповного формального контексту. Ми знаємо, що неповний формальний контекст насправді є окремим випадком багатозначного контексту. Перший розділ описує процес масштабування неповного контексту, який призводить до

бінарного формального контексту. У цьому розділі автори обговорюють залежності між атрибутами та їх вплив на вибір шкал і спосіб масштабування неповних формальних контекстів.

Інший підхід до заповнення неповних формальних контекстів представлено в роботі Othman та Yahia [9]. Представлений метод базується на правилах асоціації та вирішує проблему конфлікту правил при їх використанні для заповнення. Зокрема, представлено міру для правил асоціації, згідно з якою для заповнення невідомого значення обирається відповідне правило асоціації. Однак представлений метод не гарантує, що після його виконання ми отримаємо повний формальний контекст.

РОЗДІЛ 3. ВЛАСНІ РЕЗУЛЬТАТИ

Вивчаючи формальний концептуальний аналіз неповних даних, я також намагався запропонувати власні ідеї в цій галузі. По-перше, я дослідив експериментальний метод уточнення неповного формального контексту, який містить лише один знак питання. Таким чином, це проблема передбачення лише одного невідомого значення. Цьому присвячена перша частина цієї глави. Виявляється, що ця проблема веде до іншої, більш загальної проблеми, яка є не менш цікавою. Це дослідження змін у понятійному об'єднанні після вилучення інциденції з відповідного формального контексту. Саме цій темі присвячено другий розділ.

У цьому розділі використовуються наступні позначення:

- $C = \langle X, Y, \{\times, -, ?\}, K\rangle$ - неповний формальний контекст вхідних даних, який містить лише один знак питання, що відповідає об'єкту x_0 та атрибуту y_0 .
- $C(I) = \langle X, Y, \{\times, -, ?\}, I\rangle$ - повний формальний контекст, який створюється шляхом доповнення C заміною знаку питання на значення “$\times$”.
- $C(J) = \langle X, Y, \{\times, -, ?\}, J\rangle$ - повний формальний контекст, який створюється шляхом доповнення C заміною знаку питання на значення “-”.
- B, B(I), B(J) - відповідні концептуальні об'єднання.

  Формальні контексти C(I), C(J) є повними і відрізняються лише відношенням захворюваності. Зокрема, один створюється з іншого шляхом видалення/додавання лише одного випадку

⟨x_0 , y_0⟩. Оскільки C(I), C(J) є повними, їх можна ототожнити з класичними бінарними формальними контекстами C(I) = ⟨X, Y, I⟩, C(J) = ⟨X, Y, J⟩.

3.1. Заповнення шляхом підрахунку понять

Заповнення неповного формального контексту можна здійснити кількома способами, як показано в попередньому розділі. Однією з головних цілей цієї роботи було дослідити експериментальний підхід до заповнення, який базується на підрахунку концептів. Ідея цього методу виникла з експериментів, які спочатку проводилися для іншої роботи. Ці експерименти були проведені на кількох вибраних формальних контекстах. В результаті, відсоток успішності передбачення невідомого значення був цікавим, і саме тому я застосував цей метод.

Ідея доповнення неповного формального контексту шляхом підрахунку концептів полягає в наступному:

- Нехай вхідні дані (неповний формальний контекст C) містять лише одне невідоме значення (один знак питання).

- Ми заповнюємо C, замінюючи знак питання на значення, яке належить до концептуального об'єднання з меншою кількістю концептів. Іншими словами, якщо |B(J)| < |B(I)|, значенням для заповнення є “-”. І навпаки, якщо |B(I)| < |B(J)| має місце, значенням для заповнення є “×”. Якщо |B(I)| = |B(J)|, ніякого рішення не може бути прийнято на основі цієї процедури.

З першого розділу ми знаємо, що формальний концептуальний аналіз - це метод аналізу даних. Він виявляє кластери (формальні поняття) в даних, які часто мають природну інтерпретацію (представляють відоме поняття). Згідно з правилом “правильна відповідь завжди найбільш однозначна”, правильна кластеризація - це та, яка містить менше понять (є простішою). Однак це також означає, що цей метод завжди намагається спростити закономірності у вхідних даних і пригнічує елементи, які виходять за межі вже наявних закономірностей. Іншими словами, цей метод не намагається внести нову інформацію у вхідні дані, а натомість намагається зробити так, щоб ця інформація вписувалася у вже наявну структуру даних.

Оскільки це експериментальний метод, потрібно було провести кілька експериментів. Щоб зробити ці експерименти можливими, спочатку зосередимося на більш глибокому дослідженні проблеми і розробці базового алгоритму для цього методу доопрацювання.

Наївний алгоритм підрахунку концептів обчислює концепти в обох заповненнях і порівнює отримані значення. Обчислення наборів концептів B(I), B(J) можна виконати, наприклад, за допомогою одного з алгоритмів, представлених у першому розділі. Водночас варто замислитися над тим, чи не можна вирішити цю задачу більш ефективно, оскільки обчислення концептуального об'єднання є дуже трудомістким. Це підводить мене до проблеми виведення кількості концептів одного понятійного об'єднання з іншого. Я прийшов до висновку, що для цього зручніше спиратися на знання B(I). Виявилося, що для обчислення різниці в кількості концептів навіть не обов'язково обчислювати весь концептуальний пучок B(I), а достатньо обчислити певну його частину.

3.1.1. Основні поняття та їх властивості

Спочатку ми введемо відповідні терміни, щоб зрозуміти поставлену проблему. Спочатку з'ясуємо зв'язок між стрілочними операторами $\uparrow_I$, $\uparrow_J$, $\downarrow_I$, $\downarrow_J$, що відповідають двом можливим завершенням. Наступна теорема дуже часто використовується у решті частини цього розділу і її слід добре вивчити.

Теорема 3.1 (зв'язок стрілочних операторів $\uparrow^I$, $\uparrow^J$, $\downarrow^I$, $\downarrow^J$). Для довільних $A \subseteq X$ та $B \subseteq Y$ виконується наступне

$$A^{\uparrow J} = \begin{cases} A^{\uparrow I} & if\ x_0 \notin A, \\ A^{\uparrow I} \backslash \{y_0\} & if\ x_0 \in A, \end{cases}$$

$$B^{\downarrow J} = \begin{cases} B^{\downarrow I} & if\ y_0 \notin B, \\ B^{\downarrow I} \backslash \{x_0\} & if\ y_0 \in B, \end{cases}$$

Зокрема, $A^{\uparrow J} \subseteq A^{\uparrow I}$ та $B^{\downarrow J} \subseteq B^{\downarrow I}$.

Доведення. Одразу з введення C(I), C(J) і з того, що $J = I \setminus \langle\{x_0, y_0\rangle\}$

Розглядаючи B(I) та B(J), можна помітити, що вони можуть мати спільні формальні поняття. Таким чином, вводиться поняття стабільного поняття. Стабільними будемо називати концепти, що належать до концептуальних пучків двох завершень C(I), C(J). Це означає, що при обчисленні B(I), B(J) не потрібно рахувати ці поняття двічі.

Визначення 3.2 (стабільне поняття).
Ми називаємо поняття $c \in B(I) \cup B(J)$ стійким, коли $c \in B(I) \cap B(J)$. Ми називаємо поняття $c \in B(I) \cup B(J)$ нестійким, коли воно не є стійким.

Поняття стабільності ділить концепти B(I), B(J) на дві диз'юнктивні множини, які формують покриття всіх концептів у кожному з цих концептуальних пучків. Кожен концепт є або стабільним, або нестабільним. При обчисленні B(I), B(J) немає необхідності перераховувати стабільні концепти декілька разів. Навпаки, нестабільні концепти - це якраз ті, що належать до одного з B(I), B(J). Це означає, що ми можемо зосередитися лише на підмножині понять і мати справу з властивостями лише цих понять. Для цього потрібно визначити квартет операторів, які допоможуть нам це зробити.

Визначення 3.3 (оператори $^{\square}{}_{\square}{}^{\boxtimes}{}_{\boxtimes}$)

Для поняття $c = \langle A, B\rangle \in B(I)$, $d = \langle C, D\rangle \in B(J)$, визначимо

$c^{\square}=\langle A^{\square},B^{\square}\rangle=\langle A^{\uparrow j\downarrow J},A^{\uparrow j}\rangle$

$c_{\square} = \langle A_{\square}, B_{\square}\rangle=\langle B^{j\downarrow}, B^{j\downarrow\uparrow j}\rangle$

$d^{\boxtimes} =\langle A^{\boxtimes} ,B^{\boxtimes}\rangle=\langle A^{\uparrow i\downarrow i},A^{\uparrow i}\rangle$

$c_{\boxtimes} = \langle A_{\boxtimes} , B_{\boxtimes} \rangle=\langle B^{i\downarrow}, B^{i\downarrow i\uparrow}\rangle$

З попереднього означення зрозуміло, що для будь-якого $c \in B(I)$ виконується $c^{\square}$, $c_{\square} \in B(J)$. Аналогічно, для довільного $d \in B(J)$ виконується $d^{\boxtimes}$, $d_{\boxtimes} \in B(I)$. Мета операторів з попереднього означення - зафіксувати деякий зв'язок між поняттями з B(I) та B(J), який бажано ми будемо використовувати при створенні цільового алгоритму. Виявляється, що пари $\langle {}^{\square}, {}^{\boxtimes}\rangle$, $\langle {}_{\square}, {}_{\boxtimes}\rangle$ мають Галуа-подібні властивості, що можна тільки вітати, але вони не є безпосередньо Галуа-подібними з'єднаннями. Вони не утворюють антитонічних з'єднань Галуа, оскільки всі відображення є ізотонічними. Вони також не утворюють ізотонічних зв'язки Галуа, тому що в цьому випадку складені оператори утворюють замикання і внутрішній оператор. У нашому випадку обидва складені оператори є однотипними.

Теорема 3.4 (властивості операторів $^{\square}{}_{\square}{}^{\boxtimes}{}_{\boxtimes}$).

Нехай $c \in B(I)$, $d \in B(J)$. Відображення $c \to c^{\square}$, $c' \to c_{\square}$ та $d \to d^{\boxtimes}$, $d \to d_{\boxtimes}$ є ізотонічними і задовольняють наступним умовам

$c \le c^{\square\boxtimes}$, $d \le d^{\boxtimes\square}$, $c^{\square\boxtimes\square} = c^{\square}$, $d^{\boxtimes\square\boxtimes} = d^{\boxtimes}$

$c \ge c_{\square\boxtimes}$, $d \ge d_{\boxtimes\square}$, $c_{\square\boxtimes\square} = C_{\square}$, $d_{\boxtimes\square\boxtimes} = d_{\boxtimes}$

Доведення. Доведемо, що попереднє твердження справедливе для відображень $c \to c^{\square}$ та $d \to d^{\boxtimes}$. Доведення для $c \to c_{\square}$ та $d \to d_{\boxtimes}$ є подвійними.

Нехай $c_1 = \langle A_1 , B_1 \rangle$, $c_2 = \langle A_2 , B_2 \rangle \in B(I)$, $c_1 \le c_2$. Тоді $A_1 \subseteq A_2$, а $A_1^{\uparrow J \downarrow J} \subseteq A_2^{\uparrow J \downarrow J}$, тому $c_1^{\square} \le c_2^{\square}$.

Нехай $d_1 = \langle C_1 , D_1 , d_2 = \langle C_2 , D_2 \rangle \in B(J)$, $d_1 \le d_2$. Тоді $D_2 \subseteq D_1$

а $D_2^{\downarrow I \uparrow I} \subseteq D_1^{\downarrow I \uparrow I}$, тому $d_2^{\boxtimes} \le d_1^{\boxtimes}$.

Далі, нехай $c = \langle A, B \rangle \in B(I)$. ↑↑↑↑↓↓↓↓⊠□З теореми 3.1 маємо $A^{\uparrow J} \subseteq A^{\uparrow I}$. З того, що $A = A^{\downarrow I} \subseteq A^{\downarrow I}$, отримуємо $c \le c^{\square\boxtimes}$. Аналогічно, для $d = \langle C, D \rangle \in B(J)$, $D_J \subseteq D_I$,такщо$D_I^{\uparrow J} \subseteq D_J^{\uparrow J} = D_i$,отже,$d \le d$.

Згідно з теоремою 3.1 $A^{\downarrow I \uparrow J} = A^{\uparrow J}$ тривіально випливає з властивостей галоїдних сполук.

Як прямий наслідок попереднього, властивості представлень, які є результатом композиції $c \to c^{\square}$, $d \to d^{\boxtimes}$ і $c \to c_{\square}$, $d \to d_{\boxtimes}$. Окремі складені представлення завжди утворюють замикання або внутрішній оператор.

Теорема 3.5
Для поняття $c = \langle A, B \rangle \in B(I)$, $A^{\uparrow I} = A^{\uparrow J}$, тільки коли $B^{\downarrow I} = B^{\downarrow J}$.

Доведення. ↑ЯкщоВ = $A^{\uparrow I} = A^{\uparrow J}$, то↓$A \subseteq A^{\uparrow J \downarrow J} = A^{\uparrow I \downarrow J} = B^{\downarrow J} \subseteq B^{\downarrow I} = A$

, отже, $B^{\downarrow I} = B^{\downarrow J}$. Зворотний наслідок доводиться за аналогією.

Теорема 3.6 (стійкі поняття в $B(I)$).

Для довільного поняття $c \in B(I)$ наступне є еквівалентним:

1. $c \notin [\gamma_I(x_0), \mu_I(y_0)]$,
2. c стабільне
3. $c = c^{\square}$
4. $c = c_{\square}$
5. $c^{\square} = c_{\square}$

Доведення. $1. \Rightarrow 2.$: Нехай $c = \langle A, B \rangle \notin [\gamma_I(x_0), \mu_I(y_0)]$. Тоді виконується або $x_0 \notin A$ або $y_0 \notin B$. Якщо, наприклад, $x_0 \notin A$, то за теоремою 3.1, $B = A^{\uparrow I} = A^{\uparrow J}$, що, за теоремою 3.5, відбувається саме тоді, коли $B^{\downarrow I} = B^{\downarrow J}$. Таким чином, $c \in B(J)$, тобто c є стабільне. Доведення для випадку $y_0 \notin B$ є дуальне.

$2. \Rightarrow 3. \Rightarrow 4. \Rightarrow 5.$: За визначенням, $c = \langle A, B \rangle$ стабільне тільки тоді, коли $A^{\uparrow I} = A^{\uparrow J}$ та $B^{\downarrow I} = B^{\downarrow J}$. З теореми 3.7 ми знаємо, що умова $A^{\uparrow I} = A^{\uparrow J}$ еквівалентна $B^{\downarrow I} = B^{\downarrow J}$. Використовуючи це визначення, наведені наслідки тривіальні

$5. \Rightarrow 1.$: З того, що $c^{\square} = c_{\square}$, маємо $A^{\uparrow J \downarrow J} = B^{\downarrow J}$ та $A^{\uparrow J} = B^{\downarrow J \uparrow J}$, але з цього випливає, що $c \in B(J)$, що виключає саме поняття з $[\gamma_I(x_0), \mu_I(y_0)]$,тому що $\langle x_0, y_0 \rangle \notin J$.

Попередня теорема дає опис стабільний понять у B(I) кількома еквівалентними способами. Аналогічно можна описати стабільні поняття в B(J).

Теорема 3.7 (стійкі поняття в B(J)).

Для довільного поняття $d \in B(J)$ наступне є еквівалентним:

1. d стабільне
2. $d = d^{\boxtimes}$
3. $d = d_{\boxtimes}$

4. $d^{\boxtimes}$ стабільне
5. $d_{\boxtimes}$ стабільне

Наслідок 3.8

Нестійкими поняттями в B(I) є лише інтервал $[\gamma_I(x_0), \mu_I(y_0)]$. З попереднього також випливає, що поняття з B(I) \ B(J) - це просто поняття з $[\gamma_I(x_0), \mu_I(y_0)]$, що є приємною особливістю. Наприклад, якщо ми хочемо дослідити всі нестійкі поняття, ми фактично досліджуємо один заданий інтервал. Важливо зазначити, що для нестійких понять у B(J) подібне твердження не виконується і, отже, не утворює інтервалу. Це також одна з причин, чому зручніше починати з завершення C(I), а не C(J). Інша причина полягає в тому, що легше описати можливі зміни, яких може зазнати поняття після вилучення певного випадку.

Приклад 3.9

Приклади контекстів з різними типами понять щодо операторів $\square^{\boxtimes}$, $\square_{\boxtimes}$

	y_0	y_1	y_2
x_0	?	x	x
x_1			
x_2			

(a) Найменшим поняттям є фіксована точка обох операторів.

	y_1	y_2	y_0
x_0		x	?
x_1		x	x
x_2		x	

(b) Концепція є фіксованою $\boxdot$ точкою, але не $\boxdot$.

	y_0	y_2	y_3
x_0	?		
x_2	x	x	
x_3			

(c) Концепція не є фіксованою точкою для жодного з операторів.

Варто зазначити, що формальні контексти з прикладів 1b, можуть бути розширені в очевидний спосіб, щоб отримати приклади контекстів, які міститимуть будь-яку кількість понять з відповідними властивостями.

3.1.2. Алгоритми виведення кількості можливих концепцій завершення

Тепер побудуємо алгоритм для отримання кількості понять B(J) на основі B(I). Нам потрібно перебрати всі нестійкі поняття у B(I) і визначити, чи існують нерухомі точки операторів $\boxdot$, $\boxdot$.

Важливо це наступні дві ситуації:

1. Нестійке поняття $c \in B(I)$ є нерухомою точкою обох операторів $\boxdot$, $\boxdot$. Отже, в B(J) існує два поняття, пов'язані з c, які задаються операторами $\boxtimes$, $\boxtimes$.
2. Нестійке поняття $c \in B(I)$ не є нерухомою точкою жодного з операторів $\boxdot$, $\boxdot$. Отже, у B(J) не існує поняття, пов'язаного з c, заданого операторами $\boxtimes$, $\boxtimes$.

```
procedure DeriveConceptCount(B(I))
    count ← |B(I)|;
    for all c ∈ [γ_I(x_0), μ_I(y_0)] do
        if c = c^{□⊠} and c = c_{□⊠} then
            count ← count + 1;
        else if c ≠ c^{□⊠} and c ≠ c_{□⊠} then
            count ← count − 1;
        end if
    end for
    return count;
end procedure
```

Алгоритм 1. Кількість понять B(J) на основі B(I).

Далі ми можемо представити алгоритм обчислення завершеності неповного формального контексту на основі підрахунку концептів. Перша версія рахує зі знанням B(I). Друга версія, з іншого боку, не припускає цього знання і обчислює лише необхідну частину B(I), щоб отримати різницю в підрахунку концептів між B(I) і B(J).

procedure GETMISSINGVALUEV1($\mathcal{B}(I)$)
    $BJCount \leftarrow DeriveConceptCount(\mathcal{B}(I))$;
    if $|\mathcal{B}(I)| < BJCount$ **then**
        return $'\times'$;
    else if $|\mathcal{B}(I)| > BJCount$ **then**
        return $'-'$;
    end if
    return $'?'$;
end procedure

procedure GETMISSINGVALUEV2(C)
    $BJdif \leftarrow 0$;
    for all $c \in [\gamma_I(x_0), \mu_I(y_0)]$ **do**
        if $c = c^{\square\boxtimes}$ **and** $c = c_{\square\boxtimes}$ **then**
            $BJdif \leftarrow BJdif + 1$;
        else if $c \neq c^{\square\boxtimes}$ **and** $c \neq c_{\square\boxtimes}$ **then**
            $BJdif \leftarrow BJdif - 1$;
        end if
    end for
    if $BJdif > 0$ **then**
        return $'\times'$;
    else if $BJdif < 0$ **then**
        return $'-'$;
    end if
    return $'?'$;
end procedure

Алгоритм 2. Наповнення на основі підрахунку концептів

ВИСНОВОК

У реальних програмах може виникнути потреба в обробці неповних даних. На жаль, більшість методів обробки даних розглядають тільки повні дані як допустимі вхідні дані. Ця дипломна робота є вступом до теми обробки неповних даних за допомогою формального аналізу концепцій. Однак основними результатами моєї роботи є нові теоретичні висновки в цій галузі, що розглядають неповні формальні контексти з точно одним невідомим значенням.

Я представив вступ до проблеми неповних даних з точки зору формального концептуального аналізу та пов'язаних з ним окремих методів роботи з такими даними. Зокрема, я зосередився на методах доповнення неповних даних.

Для представлених алгоритмів я не займався питанням їх часової складності, а лише застосуванням фактичних результатів. Втім, приблизні оцінки можна легко визначити в більшості випадків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. R. Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In I. Rival, editor, Ordered Sets, pages 445-470. Reidel, Dordrecht, 1982.
2. G. Stumme. Handbook of Ontologies: Formal Concept Analysis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. ISBN 9783540709992
3. W3C: Web Ontology Language (OWL). [online]. 2012 https://www.w3.org/OWL/
4. F. Baader et al.: The Description Logic Handbook. Cambridge University Press, 2004.
5. U. Priss. Formal Concept Analysis in Information Science. Journal Annual Review of Information Science and Technology, Volume 40, Issue 1, January 2007, Pages 521-543. DOI 10.1002/aris.v40:1. ISSN: 0066-4200
6. Lenka Piskova, Stefan Pero, Tomas Horvat, and Stanislav Krajci. Mining concepts from incomplete datasets utilizing matrix factorization. In CLA 2012, 2012.
7. K. Bache and M. Lichman. UCI machine learning repository. Доступне за адресою http://archive.ics.uci.edu/ml, 2013.
8. Jun Liu and Xiao qiu Yao. Formal concept analysis of incomplete information system. In Maozhen Li, Qilian Liang, Lipo Wang, and Yibin Song, editors, FSKD, pages 2016–2020. IEEE, 2010.
9. Leila Ben Othman and Sadok Ben Yahia. Yet another approach for completing missing values. In SadokBen Yahia, EngelbertMephu Nguifo, and Radim Belohlavek, editors, Concept Lattices and Their Applications, volume 4923 of Lecture Notes in Computer Science, pages 155–169. Springer Berlin Heidelberg, 2008