
Наука і сучасні технології

УДК 681.518:662.248:004.89

ЗАСТОСУВАННЯ АБДУКЦІЇ В ЗАДАЧАХ КЛАСИФІКАЦІЇ ДАНИХ ПРО НАФТОГАЗОВІ ОБ'ЄКТИ

В.І. Шекета, М.М. Демчина, Л.М. Гобир

*ІФНТУНГ; 76019, м. Івано-Франківськ, вул. Карпатська, 15;
e-mail: demchyna@i.ua*

Роботу присвячено питанню використання засобів абдуктивних міркувань для задач видобування даних. Виконане дослідження свідчить, що класифікація даних може бути інтерпретована як одна з задач абдуктивного логічного програмування, яка дозволяє використовувати задані користувачем доменні обмеження. Інтерпретація моделей класифікації на основі дерев рішень абдуктивним способом з використанням доменних обмежень дає змогу отримати підвищення ефективності у випадку часткової відсутності вхідних даних. Також з метою розгляду імовірнісної інформації на основі базових та виведених формальних теорій було виконано розширення загального фреймворку до абдуктивного фреймворку, який базується на вагових коефіцієнтах та може використовуватися для видобування даних, що в кінцевому випадку дасть змогу покращити загальну якість результатів. Таким чином, було показано, що абдуктивні міркування можна використовувати в контексті задач класифікації для пояснення ходу міркувань виконаної класифікації і для підвищення загальної ефективності у випадку роботи системи із частково відсутніми вхідними даними і зовнішніми доменними знаннями. Такий підхід може бути удосконалений шляхом поєднання відмінних парадигм видобування даних, таких як: класифікація, асоціативні правила і кластеризація шляхом використання абдуктивного фреймворку з обмеженнями.

Ключові слова: оптимізація, інтелектуальна підтримка прийняття рішень, буріння нафтових і газових свердловин, цільові функції, правила, база знань, абдуктивний фреймворк, коефіцієнти впевненості, обмеження.

Работа посвящена вопросу использования средств абдуктивных рассуждений для задач извлечения данных. Проведенное исследование показывает, что классификация данных может быть интерпретирована как одна из задач абдуктивного логического программирования, которая позволяет использовать заданные пользователем доменные ограничения. Интерпретация моделей классификации на основе деревьев решений абдуктивным способом с использованием доменных ограничений, позволяет повысить эффективность в случае частичного отсутствия входных данных. Также с целью рассмотрения вероятностной информации на основе базовых и выводимых формальных теорий было выполнено расширение общего фреймворка к абдуктивному фреймворку, который базируется на стоимостных коэффициентах и может использоваться для извлечения данных, что в конечном итоге позволит улучшить общее качество результатов.

Таким образом, было показано, что абдуктивные рассуждения можно использовать в контексте задач классификации для объяснения хода рассуждений выполненной классификации и для повышения общей эффективности в случае работы системы с частично отсутствующими входными данными и внешними доменными знаниями.

Такой подход может быть усовершенствован путем сочетания отличных парадигм извлечения данных, таких как: классификация, ассоциативные правила и кластеризация путем использования абдуктивного фреймворка с ограничениями.

Ключевые слова: оптимизация, интеллектуальная поддержка принятия решений, бурение нефтяных и газовых скважин, целевые функции, правила, база знаний, абдуктивный фреймворк, коэффициенты уверенности, ограничения.

The research is devoted to utilization of the abductive reasoning means for data extraction problems. The conducted research shows that data classification can be interpreted as one of the abductive logic programming problems, which allows utilizing of user-defined domain restrictions. Interpretation of classification models based on decision trees made in accordance with the abductive method using domain restrictions allows increasing the efficiency in the case of partial lack of input data. In order to consider the probabilistic information with the help of

the basic and output formal theories, the overall framework was also extended to abductive framework that is based on cost factors and can be used for data mining applications, which will ultimately improve the overall quality of the results.

Thus, it was shown that abductive reasoning can be used in the context of classification problems to explain the course of reasoning of the made classification and improve the overall efficiency in the event of the system operation with the partial absence of the input data and external domain knowledge.

This approach can be improved by combining different data mining paradigms such as classification, clustering, association rules and by utilizing abductive framework with restrictions.

Keywords: optimization, intelligent decision support, drilling of oil and gas wells, objective functions, rules, knowledge base, abductive framework, confidence coefficient, limits.

Постановка проблеми в загальному вигляді. Абдукція (абдуктивні міркування, абдуктивний висновок) – це вид логічного висновку, що випливає із певного спостереження або гіпотези щодо такого спостереження. Його мета полягає в знаходженні найбільш простого і найбільш ймовірного пояснення для них. На відміну від дедуктивного висновку, в даному випадку наявність початкової умови ще зовсім не гарантує існування висновку. Тобто під абдуктивним висновком слід розуміти процес пошуку (виведення, інференції) найкращого із можливих пояснень щодо початково прийнятої істинної гіпотези. Використання вагових коефіцієнтів разом із абдукцією дозволяє виконувати ефективну побудову логічних міркувань в умовах невизначеності. Суть даного підходу полягає в знаходженні кращого пояснення для множини спостережуваних фактів шляхом пошуку доведення з мінімальною вагою для таких фактів. Тому зустрічається також термін «абдукція з ваговими коефіцієнтами», яка досить успішно застосовується в інтелектуальних системах, зокрема в контексті інтерпретації даних про нафтогазові об'єкти.

Аналіз останніх досліджень та публікацій. Дане дослідження базується на результатах, отриманих з використанням абдуктивних міркувань в області видобування даних [1]. Існуючі класифікатори, навіть якщо вони і базуються на деревах рішень та індуктивних засобах на основі алгоритмів C4.5 [2], дозволяють отримувати як вихідний результат множину правил з метою класифікації нових заданих прикладів. Таким чином, більшість з існуючих класифікаторів на основі правил функціонують на основі припущення про те, що в момент класифікації наявна вся інформація щодо нових заданих прикладів. Завдяки ймовірнісним підходам, засновані на правилах класифікатори здатні отримувати найбільш ймовірний клас на основі частоти відсутності атрибута в тренувальному наборі [3]. Водночас, такий вид припущень в більшості випадків призводить до хибної класифікації. Тому використання абдуктивного підходу є особливо корисним під час вибору класифікаційного правила, що має бути застосоване у випадку, якщо новий приклад з відсутньою інформацією потребує класифікації, яку можна виконати, використовуючи знання про домен предметної області. Під А-системами [4] розуміють Prolog-базований абдуктивний фреймворк, орієнтований на дослідження питань масштабованості системи. Він базується,

в основному, на ідеях та концепціях, що були застосовані в попередніх відомих абдуктивних процедурах. Імовірнісна абдукція Хорна, що є комбінацією формальної логіки та ймовірнісних представлень, вперше була описана в роботі [5]. Загальна ідея абдукції Хорна полягає в побудові множини незалежних вибірок з логічною програмою, що визначає їх послідовність. Використовувана в даному випадку формальна логіка базується на так званих «слабких» уявленнях, тобто не містить диз'юнкцій, а всі значення, пов'язані з невизначеністю, вбудовуються в процедуру підтримки вибірок користувача. При цьому також не накладається жодних обмежень на вибірки, виконані користувачем. В роботі [6] представлено спосіб пошуку абдуктивних доведень з мінімальною вагою, шляхом застосування методів лінійного програмування, базуючись на перетворенні початкової задачі в лінійну задачу задоволення обмежень. Водночас, за певних умов абдукція з ваговими коефіцієнтами може розглядатися як така, що має поліноміальне рішення. Для такого випадку в роботах [2, 3] представлено множину достатніх умов існування поліноміального рішення для абдукції з ваговими коефіцієнтами. Також в роботах [4-6] описано інші способи лінійного програмування для пошуку відповідних субоптимальних рішень, а в роботі [2] представлено одну із можливих реалізацій абдуктивної системи з ваговими коефіцієнтами.

Мета роботи. Покажемо, як будуються пояснення до заданих класифікацій шляхом обчислення відсутніх значень атрибутів у процесі застосування абдукції до інтерпретації даних про нафтогазові об'єкти. Доведемо, що вибір способу представлення результатів роботи алгоритму видобування даних в формі дерева рішень або дерева класифікацій є обґрунтованим, що має значення способу представлення додаткових доменних знань у формі обмежень і способу вибору техніки міркувань (абдукції), який може суттєво покращити поведінку видобутої моделі (класифікації). Крім того, проєктований метод дозволяє обробляти додаткові доменні знання в формі обмежень цілісності. Також можна використовувати механізм виявлення асоціативних правил у випадку повної відсутності зовнішніх знань. Можна також виконати розширення абдуктивного фреймворку з метою включення ймовірнісних знань як вагової оцінки гіпотез. У базі знань класифікаційні правила отримуються на основі шляхів у дереві рішень при їх злитті з асоціативними правилами, видо-

бутими з наборів даних, а класифікаційні результати в експериментах об'єднуються з експресивними поясненнями виконаних виборів користувача (користувачів) у момент класифікації.

Виклад основного матеріалу. Абдуктивний фреймворк є системою, що поєднує абдуктивні міркування і методи задоволення обмежень шляхом інтеграції фреймворку абдуктивного логічного програмування з фреймворком логічного програмування в обмеженнях. Своїй появі абдуктивний фреймворк завдячує спробі вирішити задачу забезпечення рішень декларативного програмування високого рівня та створення середовища моделювання для задач штучного інтелекту, які володіють достатнім рівнем обчислювальної ефективності.

В абдуктивному фреймворку роль абдуктивних міркувань полягає в забезпеченні автоматичної редукції (звуження) представлень проблем високого рівня і цілей верхнього рівня відповідно до нижчого рівня обчислювальних задач для незалежних форм представлення загальної проблеми. Використання технологій логічного програмування в обмеженнях є фундаментальним рішенням для підсилення ефективності процесу обчислення абдуктивного висновку. Таким чином, описаний підхід забезпечує зв'язок між властивостями домену проблем високого рівня і доменно-незалежними методами вирішення цих проблем.

Загальна обчислювальна модель для абдуктивного фреймворку складається із взаємодії між гіпотезами і генерацією обмежень через абдуктивний висновок з перевіркою консистентності (несуперечливості) для припущень, що виводяться, і задоволення обмежень для відповідно згенерованих їх рівнів. Інтеграція абдуктивних міркувань з пошуком рішення для накладених обмежень в рамках абдуктивного фреймворку є видом об'єднання, оскільки формувач обмежень не тільки дозволяє визначити кінцевий набір обмежень, згенерований при процедурі абдуктивної редукції, але також здійснює динамічний вплив на процес абдуктивного пошуку рішення. Це робить можливим виконання раннього «підрізання» абдуктивних редукцій шляхом встановлення нових відповідних обмежень на абдуктивні рішення.

На даному рівні система імплементується як відповідний мета-інтерпретатор, що є надбудовою для відповідної CLP-мови. Наприклад *ECLPS*, що використовує формувач обмежень для керування обмеженнями над скінченними доменами. Структура такої системи буде схожою до інших існуючих формувачів обмежень.

Абдуктивний фреймворк може бути застосований для вирішення широкого спектру прикладних задач. Зокрема, початкові версії були сконцентровані на вирішенні задач, пов'язаних із процесом планування, наприклад, складання розкладів, логістика. Таким чином, абдуктивний фреймворк може бути використаний і для реалізації систем, що виконують розпізнавання за умов неповної та часткової інформації, вирішують конфлікти в задачах з накладеними

обмеженнями, ідентифікують причини виникнення конфліктів, виконують інтелектуалу інтеграцію інформації в інші системи. Абдуктивний фреймворк може використовуватися як базовий фреймворк для розроблення інформаційних зв'язків для семантичної інтеграції інформації над джерелами утвореними з вебсторінок. Не зважаючи на те, що більшість з перерахованих систем не мають широкого застосування, за винятком систем, що здійснюють планування та побудову розкладів, даний клас рішень є корисним щодо індикації загальних методологічних направляючих розробки абдуктивних систем. Цінність рішень даного класу, на прикладі систем, що здійснюють планування, полягає в продукуванні експертних рішень високого рівня якості, що можуть бути порівняні із рішеннями експертів, які володіють значним досвідом роботи в даній предметній області і в той же час системи даного класу надають гнучку платформу, на якій можна експериментувати стосовно змін і преференцій.

Попри велику кількість програмних додатків на основі абдукції і їх потенційних застосувань, абдуктивний логічний фреймворк, як правило, фокусується на питаннях важливості представлення, в той час як загалом глибоке дослідження проблеми обчислювальної ефективності для задач практичного використання не виконується.

На найбільш загальному рівні такі системи використовують правила висновку в формі перелічених правил для станів. Відповідно вихідні правила розширюються правилами, що можуть обробляти скінченні домени виразів для обмежень. Для рівня реалізації А-систем використовують схожі схеми з формувачами обмежень для скінченних доменів – тобто вони виконують оцінку всієї детермінованої інформації перед виконанням недетермінованих вибірок. Відповідно, в системах такого класу рішення вважається знайденим, коли всі недетерміновані вибірки є успішно оцінені і всі набори обмежень є консистентними між собою.

Набори обмежень, що будуються в режимі виконання, приймають активну участь в процесі пошуку рішення, тобто це означає, що поточний стан набору обмежень може впливати на процес пошуку нових рішень або запускати процедуру бектрекінгу. Така функціональність досягається шляхом реалізації для кожного обмеження домену, що є набором міркувань над множиною термів Хербранда і скінченного домену виразів для обмежень, які дозволяють виконувати моніторинг стану виразу по відношенню до поточного набору обмежень. Така програмна поведінка в А-системах набуває форми конкретизації, оскільки версія *SICStus Prolog* (на відміну від інших *Prolog*-систем) використовує наперед визначений модуль для формування правил висновку на множині термів Хербранда. Відповідно в А-системах використовується також окремий вирішувач для термів Хербранда, що забезпечує підтримку процесу конкретизації. Таким чином, А-системи реалізують окремий шар верхнього рівня у формі

формувача обмежень для скінченних доменів, що забезпечує схожу до загальних виразів для скінченних доменів знань функціональність.

Такий підхід дозволяє створити простий абдуктивний фреймворк на основі тверджень Хорна з ймовірностями асоційованими з гіпотезами. Такий фреймворк включає припущення щодо бази правил і припущення про незалежність множини гіпотез.

Покажемо, що кожний вид імовірнісних знань, який подається в мережі дискретних Байєсових суджень, може бути також представлений в даному абдуктивному фреймворку. Для цього необхідно виконати пошук залежностей між логічними та імовірнісними концепціями очевидних міркувань. Все це забезпечить ефективну мову представлення, що дозволить знайти компроміс між евристичною і епістемологічною адекватністю (тобто між інтуїтивним і теоретично обґрунтованим розумінням). Все це є можливим за рахунок того, що Байєсові мережі можуть бути розширені за рівень пропозиційної мови, а за допомогою самої мови можна виконувати не тільки опис безумовно незалежних гіпотези, але й представляти довільні імовірнісні знання.

В рамках такого фреймворку відповідно можна визначити абдуктивну шему у вигляді пари $\langle F, H \rangle$, де F – множина тверджень Хорна. Змінні в F є неявно універсально квантифікованими. Нехай F' є множиною базових екземплярів елементів з F , а H є множиною атомів, що називаються очікуваними або можливими гіпотезами, а H' є множиною базових екземплярів елементів H .

Система може виконувати побудову наступних припущень стосовно базової теорії:

1. Не існує правил в F' , заголовки яких уніфікуються з деяким членом в H .

2. Якщо F' є множиною базових екземплярів елементів F , то існує можливість присвоєння числових значень кожному з базових атомів такого, що для кожного правила в F' атоми в тілі правила є строго меншими, ніж атоми в заголовках.

3. Для кожного значення a , що не передбачалося спочатку, якщо a є істинним, тоді принаймні одне з тіл B_i , таке, що $a \leftarrow B_i$ повинно бути істинним. Крім того, можна стверджувати, що правила в F' мають місце для кожного атома, що представляється як базовий, тобто не є гіпотезою.

4. Тіла правил в F' для кожного атома є взаємовиключними.

Загалом логічна програма представлятиме наслідки виборів користувача. Даний фреймворк є абдуктивним, оскільки пояснення для спостереження g забезпечуватимуть стислий опис формальних структур, в яких g є істинним. Відповідно мережі суджень можуть означатися як такі, що мають незалежну імовірність над множиною альтернатив.

Системи побудовані на імовірнісній абдукції Хорна не виконують явне оперування з об-

меженнями цілісності. Відповідно кожний вид їх реалізації базується на засобах Prolog. Сутність реалізації полягає в організації пошуку методом границь і віток, де кожне часткове пояснення з найменшим коефіцієнтом ваги (тобто найвищою імовірністю) розглядається в кожний момент часу.

Крім того, такий спосіб реалізації дозволяє підтримувати пріоритетну чергу для множин гіпотез, яка може бути розширена до рівня множини пояснень, що може включати також підрівень часткових пояснень. В кожен момент часу множина всіх пояснень розглядається як множина пояснень, що вже згенеровані з додаванням множини пояснень, що можуть бути згенеровані з наявних часткових пояснень у порядку пріоритетів.

Подібно до інших застосувань та інтерпретацій теорії абдукції, основне завдання у випадку абдукції з ваговими коефіцієнтами полягає в пошуку на основі даних, що описують спостереження або події відповідної множини гіпотез, які найкращим чином пояснюють наявну множину даних на вході.

На формальному рівні абдуктивний фреймворк з ваговими коефіцієнтами є способом представлення знань із домену предметної області, що моделюються кортежем:

$$L = (H, R, c, G),$$

де: H – множина гіпотез або пропозиційних тверджень;

R – множина правил виду:

$$(h_1 \wedge h_2 \wedge \dots \wedge h_n) \rightarrow h_{n+1},$$

де $h_1, \dots, h_{n+1} \in H$;

c – відповідна функція, $c: H \rightarrow R^+$, де $c(h)$ розглядається як вагове значення очікуваності гіпотези $h \in H$; $G \in H$ – множина цілей або підтверджень.

Таким чином, кінцеве завдання полягає в пошуку доведення з найменшим ваговим коефіцієнтом для заданого підтвердження, де вага доведення обчислюється як сума ваг усіх гіпотез, що повинні припускатися з метою завершення доведення. Відповідно гіпотеза може бути приведена до значення істини двома способами:

1) вона може припускатися як істинна шляхом встановлення ваги її припущення;

2) шляхом доведення.

При цьому слід відмітити: якщо гіпотеза проявляється як наслідок правила R , тоді вона може бути доведена без визначення ваги як така, що є істинною шляхом приведення всіх антецедентів в R до істинного значення, шляхом припущення або доведення. Якщо відповідна гіпотеза не проявляється як консеквент для кожного правила, тоді вона не може бути доведена і відповідно вона може бути встановлена як істинна тільки шляхом припущення. Якщо вага гіпотези є достатньо великою, тоді вона не може бути встановлена шляхом припущення і вона може бути тільки доведена. Для цього множина гіпотез H поділяється на дві множини H^A

та H^P . Перша множина містить гіпотези, що не можуть бути доведені, тобто такі, що ніколи не проявляються в правих частинах правила. Друга множина містить всі інші гіпотези.

Загалом в абдуктивних міркуваннях множина A розглядається як множина здійснених рішень для проблеми доведення з найменшою вагою, якщо відповідно вона є підмножиною для H^A , що є достатнім для доведення множини цілей G . Відповідно, оптимальним рішенням для проблеми доведення з найменшою вагою є здійснене рішення, що мінімізує задану загальну вагу, тобто суму ваг усіх гіпотез A .

Ланцюг зворотного висновку в стилі Prolog з доданою операцією факторизації і без обмежень щодо упорядкування (це означає, що кожний, а не тільки самий лівий літерал твердження, може бути вирішений) здатний генерувати всі можливі пояснення, що є консистентними (несуперечливими) з базою знань. Це означає, що кожне очікуване можливе пояснення консистентне з базою знань категоризується поясненням що може бути згенероване в процедурах ланцюга зворотного висновку і факторизації.

Також необхідно, щоб процедура гарантовано не генерувала жодних пояснень, що є не консистентними з відповідною базою знань. Зрозуміло, що така ситуація є неможливою в загальному випадку, оскільки консистентність пояснення з базою знань повинна бути перевірена поза абдуктивною системою інференції міркувань. Слід також відмітити, що не всі не консистентні пояснення генеруються системою, навпаки, система може генерувати тільки ті пояснення, що припускають існування літералів, які можуть бути досягнуті відповідно з початкових формул шляхом застосування ланцюга зворотного висновку. Таким чином, задача визначеності консистентності (несуперечливості) є нерозв'язною в загальному випадку, хоча існують вирішувані підвипадки і деякі з пояснень можуть бути швидко відкинуті як такі, що є неконсистентні з початковою базою знань. Зокрема в більшості випадків припущення може бути відкинуто, якщо воно порушує обмеження щодо виду або впорядкування. Тому обмеження виду є особливо ефективними щодо усунення не консистентних пояснень особливо в інтерпретаціях на основі природної мови.

На даний час існує декілька реалізацій фреймворків у загальному контексті логічного програмування з метою інтеграції індуктивних та дедуктивних представлень. Специфікації існуючих систем включають деякі види синтаксичної упередженості, а також в них наявні обмеження щодо цілісності. Загалом такі системи є абстрактним середовищем для експериментування з різними методами висновку та способами побудови міркувань. Крім того, особливістю таких систем є простота поєднання різнопланових методів, що дає змогу виконувати інтеграцію абдукції та індукції із загальними правилами.

В концепції абдуктивного навчання (ICL) така система на вході отримує деяку абдуктивну теорію $T = \langle P, A, I \rangle$ як спосіб представлення знань і дві множини базових атомів, що є її прикладами. Відповідно засоби абдуктивного навчання дозволяють знаходити абдуктивну теорію з $P' \supset P$ та $I' \supset I$ такі, що $T' \models_A E^+$ та $\forall e^- \in E^-, T' \not\models_A e^-$, де E^+ відповідає за кон'юнкцію всіх позитивних прикладів (що позначаються знаком +); відповідно E^- відповідає за диз'юнкцію всіх негативних прикладів (що позначаються знаком -).

Отже, абдуктивні теорії уможливають процес навчання на неповних знаннях, оскільки предикати, що виводяться, можуть бути використані для представлення неповних знань. Існуючі алгоритми дозволяють вивчати, перш за все, всі правила, після чого всі обмеження. Правила вивчаються за ILP-алгоритмом, що функціонує зверху вниз, а тестування покриття виконується з використанням абдуктивної процедури доведення. В даний алгоритм закладено пучкову пошукову стратегію – спеціальну евристичну функцію, яка дає відмінні вагові значення для прикладів, що покриваються засобами абдукції або без її використання. На виході фази вивчення правил отримується відповідна множина правил і множина пояснень за принципом: одне пояснення, можливо порожнє, для кожного позитивного прикладу і всі можливі пояснення, серед яких не має порожніх, для кожного негативного прикладу. Вимога того, що жодне з пояснень для негативних прикладів не є порожнім, необхідна для того, щоб виключити їх всіх у наступній фазі, що полягає у вивченні обмежень.

Обмеження вивчається на основі ICL, причому вхідний потік ICL складається з абдуктивних пояснень для позитивних прикладів як виду позитивних інтерпретацій, і абдуктивні пояснення для негативних прикладів розглядаються відповідно як вид негативних інтерпретацій. Загалом, вивчені правила разом з правилами з множини початкових знань утворюють основу системи знань в ICL.

Застосування абдуктивних технік дозволяє покращити ефективність процедур та алгоритмів видобування даних в цілому. Такі реалізації класифікуються на верхньому рівні абдуктивного фреймворку з ваговими значеннями, що дозволяє відповідно видобувати та інтегрувати змістовні знання, отримані з «сирих» даних. Абдуктивний фреймворк має своїм завданням конструювання нових абдуктивних теорій або обмежень цілісності як виду індукції засобами індуктивного логічного програмування при заданій початковій базі знань. Проте більш доцільним є використання алгоритмів видобування даних на основі частих входжень наборів елементів з метою виведення нових фактів у формі раніше невідомих розширюваних знань, зокрема у випадку відсутніх значень початкових даних. Такі виводимі факти пізніше використовуються для покращення результатів попе-

редньо видобутих даних, наприклад в задачі класифікації екземплярів.

Покажемо, як класифікаційна задача може бути перетворена безпосередньо в абдукційну проблему шляхом застосування дерев рішень. Для вирішення даної задачі необхідно ввести формальні означення, які описують процес класифікації на основі дерев рішень і дозволяють виконувати трансформацію дерев рішень у відповідну абдуктивну логічну програму.

Для заданого дерева T і шляху π в дереві T , позначимо через $Attr(\pi)$ деякий приклад $\{a_1 = v_1, \dots, a_k = v_k\}$, де кожне a_i не є листовим вузлом в T і кожне v_i є значенням атрибуту, що позначає вітку яка виходить з a_i . Означимо також через $Class(\pi)$ – клас, що позначає лист вузла для шляху π .

Визначимо концепції класифікації прикладів засобами дерев рішень у випадку відсутньої інформації. Для цього слід переписати звичайне означення класифікації як приклад окремого випадку, в якому не має відсутньої інформації.

Означення 1. Нехай T є деревом рішень, e – екземпляр (приклад) і c – клас. Вважається, що існують підстави стверджувати факт, що екземпляр e може бути класифікований як c , засобами T через δ , що позначається через $T \xrightarrow{c, \delta} e$ якщо існує шлях π в T з листовим вузлом $Class(\pi) = c$ такий, що $e \cup Attr(\pi)$ є відповідним екземпляром (прикладом) і $\delta = Attr(\pi)/e$.

Якщо $T \xrightarrow{c, \delta} e$, то говорять, що екземпляр (приклад) e класифікується як c засобами T , що представляється як $T \xrightarrow{c} e$.

Слід також відмітити, що в представленні $T \xrightarrow{c, \delta} e$ умова того, що враз $e \cup Attr(\pi)$ повинен бути екземпляром, гарантує, що значення атрибута для e є сумісними з множиною тестів, представлених через π . Іншими словами, якщо для деякого атрибуту a , $a = v \in e$ та $a = v' \in Attr(\pi)$, тоді $v = v'$. Крім того, δ представлятиме розширення до рівня прикладу e , яке є необхідним для виконання його класифікації як c з вибраного шляху.

Розглянемо спосіб перетворення дерева рішень у абдуктивну логічну програму. Нехай T – дерево рішень і π – шлях в T , тоді правило r_π асоційоване з π буде твердженням Хорна виду:

$$c \leftarrow a_1(v_1), \dots, a_n(v_n),$$

таке, що:

- 1) $Class(\pi) = c$;
- 2) $Attr(\pi) = \{a_1 = v_1, \dots, a_n = v_n\}$.

Слід відмітити, що імена атрибутів розглядаються як унарні предикатні символи і, що пара атрибут/значення $a = v$ відображається в атом $a(v)$. Більше того, при заданому (v) в означенні 1 при заданому шляху π , $Attr(\pi)$

слід розглядати як екземпляр (приклад). Програма P_T на основі тверджень Хорна, асоційована з деревом рішень T є множиною правил виду:

$$P_T = \{r_\pi | \pi\},$$

де π є шляхом в T .

В кінцевому підсумку виконується зв'язування з деревом рішень T множини IC_T канонічних обмежень цілісності, що містять заперечення виду:

$$\leftarrow a(x), a(y), x \neq y,$$

для кожного атрибута $a \in A$.

Як буде показано далі, імена атрибутів розглядаються як предикати, що виводяться, а їх множина буде представляти можливі класифікації. Обмеження цілісності, означені подібним чином, забезпечують той факт, що в кожній консистентній множині предикатів, що виводяться, кожен атрибут зустрічатиметься не більше одного разу.

Для заданого екземпляру (прикладу) $e = \{a_1 = v_1, \dots, a_k = v_k\}$, Δ_e є множиною атомів:

$$\Delta_e = \{a_1(v_1), \dots, a_k(v_k)\}.$$

Таким чином, приведені формальні представлення дають всі необхідні складові для прив'язки абдуктивного фреймворку до дерева рішень T .

Означення 2. Для заданого дерева рішень T асоційований з T абдуктивний фреймворк AB_T представлятиметься триплетом:

$$AB_T = \langle P_T, A, IC_T \rangle,$$

де P_T – Хорн-програма асоційована з T ;

IC_T – множина канонічних обмежень цілісності асоційованих з T ;

A – множина імен атрибутів.

Наступне твердження дає змогу формалізувати відповідність між класифікаціями в T і абдуктивними поясненнями в AB_T .

Твердження 1. Нехай T є деревом рішень і AB_T є відповідним абдуктивним фреймворком, а e є екземпляром (прикладом), тоді Δ_e є мінімальним поясненням для c по відношенню до AB_T , якщо і тільки якщо для деякого шляху π в T ми матимемо, що $T \xrightarrow{c, \delta} e$ і $\Delta = \Delta_\delta$.

(\Leftarrow). Припустимо, що $T \xrightarrow{c, \delta} e$, тоді згідно з означенням 1 для деякого шляху π в T , $e \cup Attr(\pi)$ є екземпляром (прикладом) і $\delta = Attr(\pi)/e$. Розглянемо правило r_π :

$$c \leftarrow a_1(v_1), \dots, a_n(v_n).$$

Нехай $body(r_\pi) = \{a_1(v_1), \dots, a_n(v_n)\}$, тоді $body(r_\pi) = \Delta_{Attr(\pi)}$ і $P_T \cup body(r_\pi) \models c$. У випадку, якщо $\Delta = body(r_\pi)/\Delta_e$, тоді $P_T \cup \Delta_e \cup \Delta \models c$ та $\Delta = \Delta_\delta$. Більше того, оскільки $e \cup Attr(\pi)$ є екземпляром (прикладом) і $(\Delta_e \cup \Delta) \subseteq \Delta_{e \cup Attr(\pi)}$,

$P_T \cup \Delta_e \cup \Delta \cup IC_T$ є консистентними. Таким чином, Δ є поясненням для c при заданому Δ_e . Згідно зі способом побудови Δ є також Δ_e -мінімальним поясненням.

(\Rightarrow). Припустимо, що Δ є Δ_e -мінімальним поясненням для c . Очевидно, що $\Delta \cap \Delta_e = \{\}$ на основі мінімальності Δ . Оскільки згідно способу побудови P_T , $P_T \cup \Delta_e \cup \Delta = c$, то існує шлях π в T , такий, що r_π є правилом виду:

$$c \leftarrow a_1(v_1), \dots, a_n(v_n),$$

$$i \{a_1(v_1), \dots, a_n(v_n)\} \subseteq \Delta_e \cup \Delta.$$

Нехай $\delta = Attr(\pi)/e$. За способом побудови і мінімальності Δ отримаємо, що $\Delta = \Delta_\delta = \{a_1(v_1), \dots, a_n(v_n)\}/\Delta_e$. В кінцевому підсумку буде очевидно, що $e \cup Attr(\pi)$ є екземпляром (прикладом) на основі консистентності $P_T \cup \Delta_e \cup \Delta \cup IC_T$ і загального спостереження того, що $\Delta_{e \cup Attr(\pi)} = \Delta_e \cup \Delta$.

Задачу класифікації з точки зору застосування абдукції можна розглядати як альтернативний процес, хоча в той же час і як еквівалентний спосіб виконання класифікації на основі дерева рішень. При цьому слід відзначити, що абдуктивні міркування, які вимагаються в такому альтернативному представленні, є досить обмеженими і не використовують повною мірою потенціальну потужність абдуктивних міркувань у логічному програмуванні. Зокрема виконання відповідних перетворень вимагає використання наперед визначених канонічних множин обмежень цілісності, в яких при побудові пояснень потрібно уникати включення відмінних значень для одного і того ж атрибуту. Вважається, що ефективним способом використання абдуктивних міркувань є спосіб додавання доменно-специфікованих знань з метою покращення результативності задач класифікації. Таким чином, доменно-специфіковані знання можна приймати в розгляд під час виконання класифікації багатьма способами, зокрема шляхом виділення класифікації або надання переваги певному виду класифікації над іншими видами. Отже, включення у розрахунок доменно-специфікованих знань у стандартні алгоритми класифікації на основі дерев рішень не завжди даватиме пряму та просту імплементацію, і, таким чином, може вимагати значних модифікацій базових алгоритмів. В той же час абдуктивні фреймворки і їх існуючі імплементації, як правило, вже включають механізми, що можуть бути безпосередньо використані для представлення і управління доменно-специфікованими знаннями, тобто знаннями, що представлені в певному домені на основі, наприклад, параметрів наступного виду (рис. 1).

Загалом обмеження цілісності (за винятком канонічних) використовуються в абдуктивному фреймворку як спосіб вираження доменно-специфікованих знань.

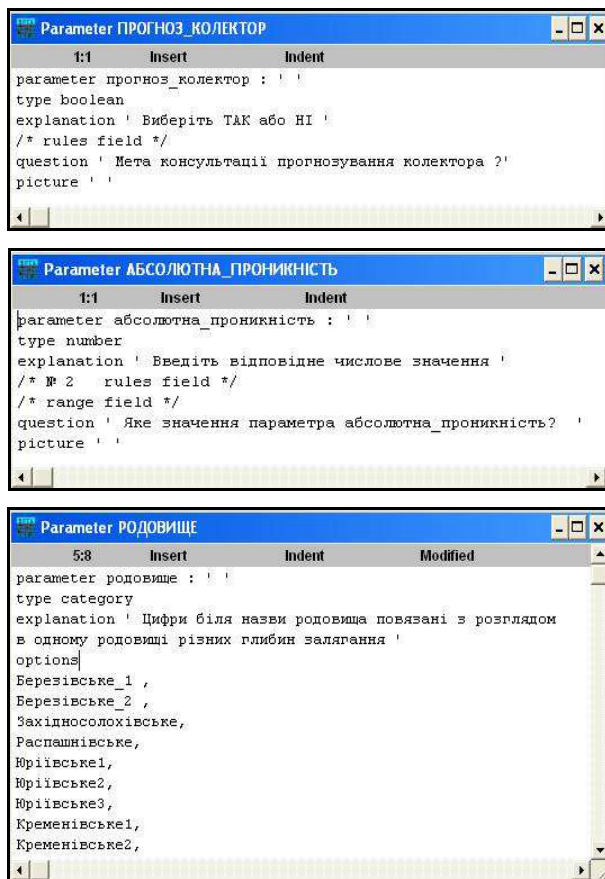


Рисунок 1 – Приклади параметрів в інтелектуальній системі

Як свідчать існуючі застосування, обмеження цілісності можуть бути використані для додавання знань, що відносяться до значень різних атрибутів, включаючи ті, що не входять в початкове дерево рішень. Таким чином, дана особливість може допомогти покращити ефективність задач класифікації на основі того факту, що обмеження цілісності можуть додавати знання, які є релевантними до атрибутів, що вже входять в дерево рішень, але не представлені в ньому явно у вигляді секцій та релевантних пояснень виду (рис. 2).

Дерева рішень можуть оперувати з імовірнісними значеннями на рівні результату тестування пар атрибут/значення. Тобто, кожна вітка дерева може мати мітку у вигляді значення, що відповідає атрибуту, який позначає батьківський вузол, а також має мітку у вигляді імовірнісного значення, що відображає, наскільки імовірним є те, що результат спостереження асоціюватиметься саме з даним значенням для заданого атрибуту. Такий вид імовірнісної інформації є корисним, коли ми намагаємося класифікувати нові екземпляри з відсутньою інформацією щодо атрибутів. Навіть в цьому випадку додаткові доменні знання можуть прийматися в розгляд в задачах класифікації з метою динамічного одержання кращих значень імовірності на можливих класифікаціях нових екземплярів (прикладів) з недостатньою специфікацією (тобто відсутністю частини релевантної інформації).

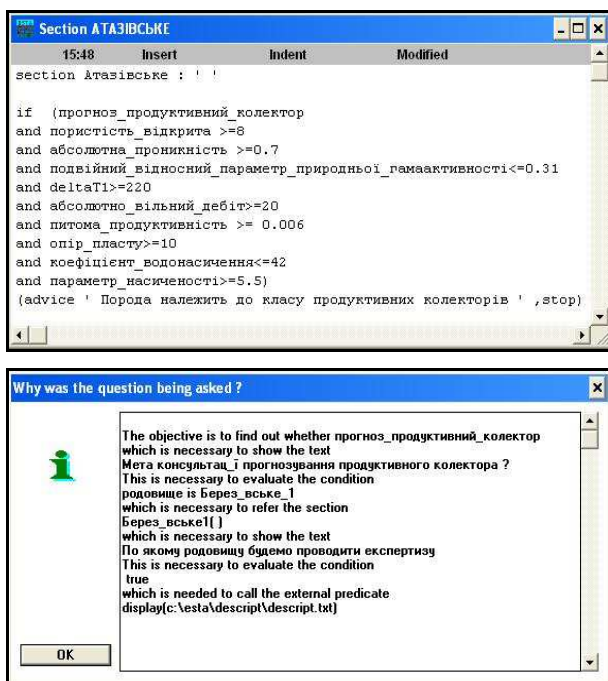


Рисунок 2 – Приклади секцій та поясень в інтелектуальній системі

Представлена вище методика потребує специфікованих зовнішніх знань щодо домену, який використовується як обмеження цілісності в абдуктивному фреймворку. В даному дослідженні розглядаються доменні знання як правила, що є похідними від людини-експерта. Як було показано вище, окремим способом отримання правил в домені є їх виведення з вхідних даних.

Як показано в роботі [7] існує можливість використання асоціативних правил (так званих активних правил) в абдуктивних фреймворках загального призначення. Така ідея робить побудову повністю автоматизованого абдуктивного фреймворку можливою, шляхом поєднання двох найбільш важливих парадигм видобування даних: 1) парадигми класифікації на основі дерев рішень; 2) парадигми асоціативних правил.

Тому, як окрему задачу можна виділити створення абдуктивних методик видобування даних на основі асоціативних правил, як обмежень цілісності, шляхом використання наборів фактичних даних (в нашому випадку даних технологічного процесу буріння).

Розглядувана проблема пошуку обмеження цілісності може бути вирішена шляхом інтерактивного аналізу асоціативних правил, отриманих на основі апріорних алгоритмів. Оскільки існує певна збалансованість між негативними і позитивними значеннями певних атрибутів, в результаті отримані правила мають високі значення коефіцієнта довіри і коефіцієнта підтримки, але для їх практичного застосування необхідно використовувати окремі метрики, наприклад метрики «піднімання» та «впливу». Отримані в результаті асоціативні правила дають змогу представляти фактичні залежності між атрибутами.

Тестування результатів задач класифікації з використанням абдуктивного фреймворку, що функціонує на основі дерева рішень і асоціативних правил, які виступають в ролі обмежень цілісності, доводить здатність системи виводити коректні значення, навіть у випадку часткової відсутності вхідних даних.

Виконаємо поєднання результатів роботи алгоритму класифікації видобутих даних, що базується на дереві рішень з додатковими доменними знаннями, які отримані за допомогою моделі видобування даних і на асоціативних правилах, що викликають інтерес з точки зору отримання вхідного потоку для абдуктивного середовища, яке здатне класифікувати, а також специфікувати відповідні міркування, що були використані для побудови класифікацій, в тому числі при прогнозуванні відсутніх значень. В рамках такого підходу отримуються нові особливості, що представляються на рівні нової абдуктивної шemi, яка використовується в даному випадку, а саме:

- 1) цілі є прийнятними;
- 2) припущення виконується на різних рівнях специфікованості;
- 3) враховується надлишковість припущень (зокрема при побудові доведень).

Виконаний аналіз свідчить, що абдукція з ранговими коефіцієнтами найкращим чином підходить для побудови фреймворку, який задовольняє переліченим вимогам в порівнянні з абдуктивним фреймворком, який базується на предикатній логіці.

На рисунку 3 представлено кінцеву діаграму класифікаційної системи. Як показано на даному рисунку, вхідний потік даних складають екземпляри для класифікації і відповідний набір даних. Дерево рішень індукується з набору даних за допомогою алгоритму класу C4.5, а на основі алгоритмів класу Argio1 виконується видобування частих входжень і відповідних асоціативних правил.

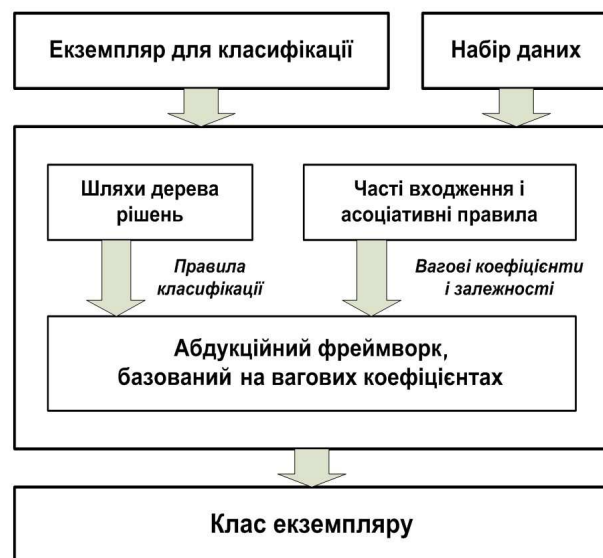


Рисунок 3 – Базована на абдукції класифікаційна система, що використовує асоціативні правила в якості доменних знань

Таким чином, проєктований фреймворк базується на абдукції з ваговими коефіцієнтами, де кожен елемент, який виводиться, має асоційований з ним ваговий коефіцієнт, що повинен бути активований в ході побудови припущення. В пропонуваному підході правила також характеризуються ваговим коефіцієнтом, що повинен бути активований, якщо правило використовується в процесі класифікації. Відповідно, кінцевим класом буде вважатися клас, що характеризується мінімальним сумарним ваговим коефіцієнтом відповідного абдуктивного доведення.

Розширення базового фреймворку шляхом включення вагових коефіцієнтів необхідне згідно з такими міркуваннями:

1) потрібне введення кількісної міри для вибору серед окремих можливих рішень (в тому числі абдуктивних доведень). Слід відмітити, що в класифікації, побудованій за допомогою такого підходу, може бути більше одного пояснення;

2) в базі знань використовується імовірнісна інформація для того, щоб відкинути частину малоімовірних доведень.

Процедури, що використовуються для аналізу пошукового простору, є процедурами повного перебору в глибину з можливими відтинаннями, що виконуються шляхом використання базової структури обмеження цілісності. Іншим видом обмежень, що повинні мати місце в кожному здійсненому рішенні, є верифіковане виконання кроків обробки даних.

Наведемо основні характеристики проєктованого абдуктивного фреймворку:

1) кожен атом, що входить принаймні в одне правило, є асоційованим з цілочисельним значенням, що представляє його вагу. Наприклад, в наступному правилі:

$$Q^{c(Q)} \leftarrow P_1^{c(P_1)} \wedge P_2^{c(P_2)},$$

де Q (*head*), P_1 (*body*) та P_2 (*body*) – атоми, що зустрічаються в правилі;

$c(Q)$, $c(P_1)$ та $c(P_2)$ – їх вагові значення;

2) припустимо, що заголовок правила має вагу меншу або рівну тій, що припускається атомами в тілі:

$$c(Q) \leq c(P_1) + c(P_2).$$

3) надлишковість використовуються з метою пошуку доведення з найменшим ваговим коефіцієнтом.

Вагові коефіцієнти, що асоціюються з кожним атомом, правилом або рішенням, співвідносяться з їх імовірнісними значеннями.

Ваговий коефіцієнт атома визначається на основі кількості кортежів, у яких даний атом є хибним:

$$\sup(|D|) - \sup(A) = \sup(\neg A),$$

де $|D|$ – кількість кортежів у наборі даних D . Відповідно в термінах імовірності вага атома A дорівнює $|D|P(A)$.

Подібним чином кожне правило має вагу, що рівна кількості кортежів, у яких правило є

хибним. Наприклад, для правила $H \leftarrow B$ його вага складатиме:

$$c(H \leftarrow B) = \sup(B \wedge \neg H).$$

В термінах коефіцієнта впевненості для того ж правила матимемо:

$$\begin{aligned} c(H \leftarrow B) &= \sup(B \wedge \neg H) = |D|P(B \wedge \neg H) = \\ &= |D|P(\neg H | B)P(B) = |D|(1 - P(H | B))P(B) = \\ &= (1 - \text{conf}(H \leftarrow B))\sup(B). \end{aligned}$$

Абдуктивним рішенням A для цілі G вважатимемо множину атомів $\{a_1 \wedge \dots \wedge a_n\}$, тобто $A \rightarrow G$, де $A = \{a_1 \wedge \dots \wedge a_n\}$.

В даному випадку необхідно, щоб вага A була інверсно пов'язана з частотою рішення A .

Твердження 2. З припущення щодо взаємного виключення серед інвертованих атомів матимемо, що:

$$|D|P(\neg A) = c(A).$$

Доведення.

$|D|P(\neg A) = |D|P(\neg a_1 \vee \dots \vee \neg a_n)$. Шляхом використання гіпотези щодо взаємного виключення отримаємо, що

$$\begin{aligned} |D|P(\neg a_1 \vee \dots \vee \neg a_n) &= |D|(P(\neg a_1) + \dots + P(\neg a_n)) = \\ &= c(a_1) + \dots + c(a_n) = c(A). \end{aligned}$$

Такий підхід є нещільною апроксимацією імовірності $P(\neg A)$, оскільки не має прив'язки до проміжку $[0, 1]$. Проте в даному випадку це не є проблемою, оскільки немає реальної потреби в фактичних значеннях інвертованої частоти, а існує тільки потреба в рішеннях з низьким значенням вагового коефіцієнта, що є найбільш імовірним в певних відношеннях. Таким чином, останнє твердження показує зв'язок між вагою та імовірністю абдуктивного рішення.

В кінцевому підсумку фреймворк узагальнюється, наскільки це можливо, наступним чином:

1) всі предикати можуть бути виведені, навіть ті, що знаходяться в заголовках правил;

2) база знань може містити петлі виду: $A \rightarrow B, B \rightarrow C, C \rightarrow A$.

Процедура доведення, що реалізується в даному підході, може обробляти обидві дані вимоги. Дана процедура була вперше розроблена як надбудова до SICStus Prolog [8], що дозволяє обробляти петлі на етапах пошуку і відтинати певні можливі доведення відповідно до вагових коефіцієнтів атомів. Виконання таких процедур, як сортування здійснених рішень і видалення деяких інших обмежень цілісності на атоми здійснюється на наступних кроках обробки.

На рисунку 4 представлена загальна структура системи, а на рисунку 5 деталізовано особливості офлайнних обчислень при побудові бази знань.

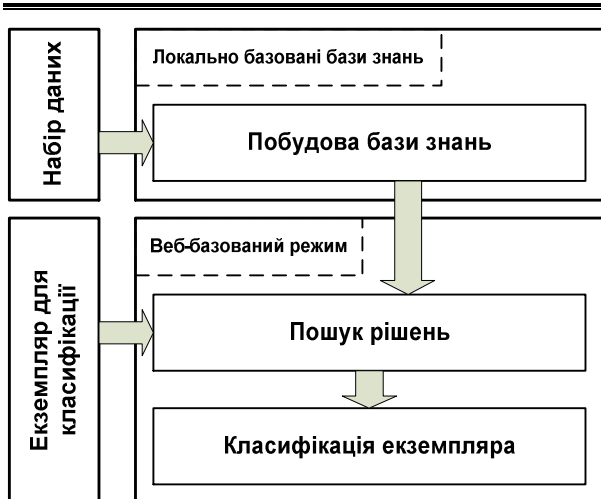


Рисунок 4 – Представлення абдукційної системи для видобування знань



Рисунок 5 – Деталізація локальної процедури побудови баз знань

Ще однією перевагою абдуктивного фреймворку є те, що він допомагає користувачу зрозуміти причини та міркування, згідно з якими ті чи інші екземпляри були асоційовані із виділеним класом.

Слід відмітити, що всі правила видобуваються автоматично з набору даних, шляхом використання апріорного алгоритму. Єдине, що ми отримуємо на вході, – це набір даних і приклад для класифікації. Особливість також полягає в тому, що інформація, яка індукується і використовується абдуктивною системою, загалом повністю ігнорується класифікаторами на основі дерев рішень.

Приклади реалізацій показують, що абдуктивні системи здатні виводити певну інформацію щодо класифікованого екземпляра шляхом використання домену знань, виведеного з набору даних. Додаткова похідна інформація дозволяє покращити функціональність класифікатора, тобто отримати класифікації вищої якості.

Сутність тестування системи полягала в запуску задач класифікації на різних екземплярах, що показало, що у всіх випадках отримується однакова класифікація як за допомогою дерева рішень, так і за допомогою абдуктивного фреймворку, але абдуктивний фреймворк дозволяє отримати додаткові пояснення таких результатів. Різниця полягає тільки в способах пояснення отриманих результатів, оскільки зрозуміло, що не всі моделі представлення знань є такими, що можуть інтерпретуватися людиною.

Крім того, бувають випадки, коли абдуктивна система не використовує жодного класифікаційного правила, і також можливо, що вагове значення абдукції в деяких випадках є далеким від мінімального. В таких випадках значенням такого класифікаційного пояснення є те, що інформація щодо об'єкту, який повинен бути класифікованим, не є достатньою для прив'язки його до певного специфікованого цільового класу. В таких випадках вибирається найбільш імовірний клас.

При проведенні тестування виконується розбиття множини записів на два різних набори даних: тренувальний набір і тестувальний набір. Різні розбиття створюються з метою уникнення неоднозначності внаслідок окремого часткового розбиття. Відповідно, експериментальні результати показують, що класифікація на основі дерева рішень і абдуктивна класифікація є еквівалентними по відношенню до точності класифікації. Причому така еквівалентність має місце, якщо не має відсутності даних в тестовому наборі.

Абдуктивний фреймворк є більш функціональним у випадку, коли виконується додавання нової інформації, наприклад, при видобуванні асоціативного правила з бази знань. Можна побачити, що додавання асоціативних правил до бази знань абдуктивної системи робить таку систему здатною підвищувати точність класифікації шляхом отримання результатів, досить близьких до класифікатора на основі дерева рішень.

Запропонований абдуктивний фреймворк включає в себе асоціативні правила, як обмеження цілісності всередині фреймворку і як правила бази знань для абдуктивних доведень. Відповідно, звичайний фреймворк основі асоціативних правил для визначення коефіцієнта підтримки або коефіцієнта довіри не в усіх випадках дає хороші результати, якщо не виконано точну установку порогових значень. В запропонованому підході очікуваний результат полягає у використанні відмінних метрик, що дозволитимуть встановлювати точну кореляцію серед атрибутів. Відповідно в літературних джерелах такі правила позначаються як коре-

ляційні. Важливим завданням є адаптація існуючих процедур, що здатні знаходити кореляційні правила у великих баз даних.

Такий вид кореляції є важливим в області абдуктивних міркувань, оскільки вони можуть краще виразити залежності між каузальними подіями і ефектами.

Питання класифікації даних є одним із найбільш важливих в області машинного навчання [9], що полягає в прогнозуванні класу об'єктів при заданих характеристиках таких об'єктів і відповідної множини прикладів коректних і хибних класифікацій для інших відомих об'єктів. Для вирішення цієї задачі використовується суміш дескриптивних і прогнозуючих підходів.

Ідея використання процесу кластеризації полягає у пошуку схожостей між правилами та за допомогою абдукції знаходження відмінностей між правилами в тому ж самому кластері. Основна складність в даній випадку полягає у визначенні метрики, яка здатна порівнювати різні правила.

Після введення оптимальної метрики є можливим отримати засобами кластеризації декілька множин правил. Виконуючи фіксацію множини і порівняння схожих правил, тобто таких, які мають схожі висновки, з відмінними заголовками, можна абдуктивно вивести причину того, чому правила мають різні заголовки.

Таким чином, проблема полягає у пошуку розв'язку наступної підпроблеми: для заданого набору прикладів з коректною класифікацією для кожного прикладу потрібно вибрати значення атрибутів, які необхідні для внесення змін з метою отримання відмінних класифікацій. Тобто, при коректному виборі метрики і відповідного асоційованого з нею алгоритму кластеризації разом з правильним абдуктивним формулюванням проблеми можна отримати рішення задачі пошуку схожостей між правилами.

В роботах стосовно машинного навчання [10, 11] представлено цілий ряд алгоритмів, здатних класифікувати нові об'єкти з високою якістю відповідно до заданих прикладів. Проте опис екземплярів і нових об'єктів може бути виконаний тільки засобами «слабких» мов, що, як правило, базуються на ідеях атрибутів, у яких кожен об'єкт описується як такий, що складається з атрибутів та їх значень, які відмінні від значень класу.

Для того, щоб розширити можливості мови до рівня представлення гіпотез і прикладів першого порядку, важливо зрозуміти, як процес навчання в логіці першого порядку узагальнює навчання виду атрибут-значення. Деякі рішення даної проблеми представлені в [12]. На сьогоднішній день більшість систем даного класу використовує логіку першого порядку для задання структури об'єкта. В пропозиційному випадку об'єкти можуть розглядатися як кортежі значень атрибутів. Відповідно у мовах першого порядку також можливо описувати об'єкти на верхньому рівні за допомогою комплексних типів (множини, списки).

Більшість досліджень в даній області належать до проблеми індуктивного логічного програмування, тому більшість таких результатів можуть бути адаптовані до проблеми включення їх в абдуктивний фреймворк. На практиці є можливим розбиття імовірнісного значення об'єкта на відповідні імовірності його пропозиційних атрибутів, що дає змогу оцінити входження об'єкта в тренувальні набори замість входжень його атрибутів.

Використання логічного програмування для вирішення індуктивних задач, дало змогу досягти верхньої границі у підсилленні дескриптивної складової класифікаційної проблеми шляхом використання підмножини логіки першого порядку як нового опису відповідних гіпотез. Проте такий підхід був результатом значного збільшення простору пошуку.

В даному контексті видобування реляційних даних, що являє собою видобування даних в дескриптивній мові першого порядку, може розглядатися як ефективне поєднання логічного програмування і відносно невеликого пошукового простору завдяки використанню імовірнісних значень. Таким чином, можна зробити висновок, що використання абдуктивного імовірнісного фреймворку може бути більш ефективним для доменних знань, що описані засобами, відмінними від логіки першого порядку.

Висновки

Отримані результати показують, що абдуктивний фреймворк може розглядатися як деякий шар, здатний приймати знання від користувача і відповідні запити мови програмування високого рівня, після чого обчислювати рішення. Вихідною точкою дослідження є те, що поєднання видобування даних з фреймворком логічного програмування забезпечуватиме нас можливістю використання також доменних знань у процесі видобування знань і в процесі їх використання, що разом дасть змогу отримати результати, які дозволять покращити якість дерева класифікації у випадку, якщо робота з деревом виконується на більш складному рівні, ніж рівень простого обходу, а загальне фреймворк-застосування відповідно базується на абдуктивному підході. Таке твердження має місце, оскільки базовим спостереженням є спостереження про те, що застосування класифікаційного дерева до нових спостережень може розглядатися як спосіб прямого абдуктивного обчислення, як тільки дерево представляється як колекція правил очевидним чином. Все це відкриває нові можливості, оскільки абдуктивні обчислення можуть набувати досить складної форми, зокрема вони можуть приймати до розгляду кілька типів знань одночасно.

Література

1 Jiawei Han. Data Mining: Concepts and Techniques / Jiawei Han and Micheline Kamber. – Morgan-Kaufman, 2000. – 28 p.

2 Liu, B.; Hsu, W. & Ma, Y., Integrating classification and association rule mining. Proceedings of the 4th international conference on Knowledge Discovery and Data mining (KDD'98), AAAI Press, 1998, pp. 80-86.

3 Wenmin Li. CMAR: Accurate and efficient classification based on multiple class-association rules / Wenmin Li, Jiawei Han, and Jian Pei. – In ICDM, 2001. – P. 369-376.

4 Вагин В.Н. Абдукция в задачах планирования работ в сложных объектах / В.Н. Вагин, К.Ю. Хотимчук // Искусственный интеллект и принятие решений. – М.: Ленанд, 2011. – Т.1. – С. 3-13.

5 Quinlan J.R. Improved use of continuous attributes in C4.5 / Quinlan J.R. // Journal of Artificial Intelligence Research. – 1996. – V.4. – P. 77-90.

6 Mancarella P. An abductive proof procedure handling active rules / P. Mancarella and G. Terreni // AI*IA 2003: Advances in Artificial Intelligence / A.Cappelli and F.Turini, editors. – SV of LNAI, 2003.– P. 105-117.

7 Mohammed J. Zaki. Theoretical foundations of association rules / Mohammed J. Zaki and Mitsunori Ogihara; in Proceedings of 3 rd SIGMOD'98 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'98), Seattle, Washington, 1998. – P. 10-17.

8 Eskilson, J. & Carlsson, M., SICStus MT - Multithreaded Execution Environment for SICStus Prolog. Konstantinos F. Sagonas, ed. Implementation Technology for Programming Languages based on Logic, 1998, pp. 59-71.

9 Kakas A.C. Abductive concept learning / Antonis C. Kakas, F. Riguzzi // New Generation Computing, 2000. – V. 18(3). – P. 243-294.

10 Michell T. Machine Learning / T. Michell. – McGraw Hill, 1997. – 414 p.

11 Kakas A.C. Aclp: Abductive constraint logic programming / A.C. Kakas, A. Michael, and C. Mourlas // Journal of Logic Programming, 2000. – V. 44 (1-3). – P. 129-177.

12 Dubois D. Automated reasoning using possibilistic logic: Semantics, belief revision, and variable certainty weights / D. Dubois, J. Lang, H Prade. – Knowledge and Data Engineering. – 1994. – Feb. – P. 64-71.

*Стаття надійшла до редакційної колегії
12.09.14*

*Рекомендована до друку
професором Горбійчуком М.І.
(ІФНТУНГ, м. Івано-Франківськ)
професором Петришиним Л.Б.
(Прикарпатський національний університет,
ім. В. Стефаніка, м. Івано-Франківськ)*