

МЕТОД ПОШУКУ АСОЦІАТИВНИХ ПРАВИЛ ПРИ РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ЩО БАЗУЄТЬСЯ НА ПОПЕРЕДНЬОМУ РОЗБИТТІ МНОЖИНИ ЗАВДАНЬ

Вінницький національний технічний університет

Анотація

У даній роботі обґрунтовано удосконалення методу пошуку асоціативних правил FP-Growth при розробці програмного забезпечення, за рахунок попередньої класифікації завдань відповідно до їх складності. Описане удосконалення дозволить скоротити тривалість пошуку таких асоціативних залежностей.

Ключові слова: класифікація даних; дерева рішень; метод FP-Growth; розробка програмного забезпечення; асоціативні правила.

Abstract

The improvement of the method of associative rules search FP-Growth during the software development, by the preliminary classification of tasks in accordance with their complexity, is substantiated in this paper. The described improvement will allow decrease the search time for such associative dependencies.

Keywords: data classification; decision trees; FP-Growth method; software development; associative rules.

У роботі [1] обґрунтовано доцільність використання методу генерації частих предметних FP-growth (FPG) для пошуку асоціативних правил при розробці програмного забезпечення та визначено основні його недоліки [2]: розмір FP-дерева, не може перевищувати розмір основної пам'яті комп'ютера; існуючі модифікації методу не враховують особливостей етапів процесу розробки ПЗ, що може негативно впливати на кінцеву множину знайдених асоціативних правил.

Означені недоліки можна усунути за рахунок удосконалення методу, що полягає у попередньому розбитті множини завдань на три групи в залежності від складності їх виконання: quick wins (завдання, які легко реалізувати), difficult tasks (завдання, які важко реалізувати), medium (завдання, по складності між легкими і важкими).

Метою роботи є удосконалення методу пошуку асоціативних правил при розробці програмного забезпечення для зменшення тривалості їх пошуку.

Оскільки завдання потрібно розділити на цільові класи, то доцільно застосовувати класифікацію даних, що є двоетапним процесом [3]. На першому етапі на наборі навчальних даних створюється модель шляхом застосування алгоритмів класифікації. На другому етапі розроблена модель тестується із використанням попередньо визначеного набору тестів для вимірювання точності розробленої моделі.

Як видно з таблиці 1, у якій наведено порівняння методів класифікації даних, для пошуку асоціативних правил при розробці програмного забезпечення доцільно застосовувати метод дерев рішень. Оскільки потрібно отримати легко інтерпретовані дані за короткий термін, доцільно застосовувати алгоритм C 4.5. Характеристики завдання відносно яких здійснюється класифікація завдань, такі: пріоритет завдання, його критичність та тривалість його розробки..

У формалізованому вигляді удосконалений метод FP-Growth містить такі основні кроки:

1. Задається значення $minconf$, відносно якого буде визначатися чи знайдене асоціативне правило цікаве для експерта чи ні.
2. Дані з таблиці, що об'єднує інформацію з множини $Task_i$ і Dev_j , класифікуються на три класи в залежності від складності їх виконання.
3. Для кожного класу завдань визначається значення $minsupp$.
4. Здійснюється перетворення даних з кожної нової таблиці в деревовидну структуру – FP-дерево.
5. Здійснюється добування частих предметних наборів із FP-дерева.
6. Визначаються асоціативні правила із знайдених предметних наборів.
7. Обраховується значення достовірності $conf$ для кожного правила.

8. Утворюється множина асоціативних правил AR , в яку входять всі АП значення, $conf$ яких більша чи рівна значенню $minconf$.

Оскільки обробка класів відбувається паралельно, то кроки 3-8 даного методу виконуються одночасно для трьох груп завдань.

Таблиця 1 – Порівняння методів класифікації даних, які можуть бути застосовані для модифікації FPG методу

Назва методу	Переваги	Недоліки
Дерева рішень (алгоритм ID3)	швидка побудова компактного дерева рішень; розглядається увесь набір даних для створення дерева	дані можуть бути надмірно класифіковані; в момент часу перевіряється лише один атрибут;
Дерева рішень (алгоритм C4.5)	побудовану модель, можна легко інтерпретувати; алгоритм простий в реалізації; можна використовувати як дискретні, так і неперервні величини	незначні варіації даних можуть призвести до різних дерев рішень; при використанні недостатнього тренувального набору, класифікація не точна
Опорні вектори	точніший ніж дерева рішень	потребує затрат часу на великі матричні операції
k-найближчих сусідів	відсутній навчальний процес; не потрібно робити жодних припущень про характеристики об'єктів; комплексні поняття можуть бути вивчені локальним наближенням	обчислювальна складність методу висока; не можна інтерпретувати модель; важко знайти k-найближчих сусідів, коли набір даних потужний;
Наївний Байєсівський метод	класифікації даних відбувається швидко; існує можливість створювати імовірнісні прогнози	відсутність у можливості реалізації регресії; наявність багатьох припущень.

Запропонований удосконалений метод FP-Growth лежить в основі розробленої інформаційної технології пошуку асоціативних правил при розробці програмного забезпечення [4].

Отже, запропоновано удосконалення існуючого методу генерації частих предметних наборів FP-Growth за рахунок класифікації завдань на групи в залежності від їх складності. Дане удосконалення дозволить скоротити тривалість пошук асоціативних залежностей.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Савчук Т. Обґрунтування доцільності використання методу fp-growth (fpg) при розробці програмного забезпечення / Т. Савчук, Н. Приймак // Вчені записки ТНУ імені В.І. Вернадського. Серія Технічні науки № 5 – Київ, 2018. – С.41-45.
2. [Zaki M.](#) New Algorithms for Fast Discovery of Association Rules/ M. Zaki, S. Parthasarathy, M. Ogihara, Wei Li // Technical report – Rochester, NY, 1997
3. CSE5230 Tutorial: The ID3 Decision Tree Algorithm [Електронний ресурс] // Monash university. – 2004. – Режим доступу: <http://web.info.uvt.ro/~dzaharie/dm2017/EN/projects/Algorithms/DecisionTrees/ID3Tutorial.pdf>.
4. Савчук Т. О., Приймак Н. В. Розробка інформаційної технології процесу пошуку асоціативних правил при розробці програмного забезпечення. Наукові праці : наук. журн. – Вип. 305. Т. 317. Комп'ютерні технології. – Миколаїв : Вид-во ЧНУ ім. П. Могили, 2018. – С. 6-11

Савчук Тамара Олександрівна — PhD, професор, професор кафедри комп'ютерних наук ВНТУ, Вінницький національний технічний університет, м. Вінниця

Приймак Наталія Василівна — аспірант кафедри комп'ютерних наук ВНТУ, Вінницький національний технічний університет, м. Вінниця, e-mail: nata.pryimak@gmail.com

Tamara O. Savchuk — PhD, Professor, Professor of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia

Natalia V. Pryimak — postgraduate of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: nata.pryimak@gmail.com