УДК 004.75

В. П. Марценюк докт. техн. наук, проф., Н. В. Мілян
Тернопільський національний технічний університет імені Івана Пулюя, Україна

ВИКОРИСТАННЯ АЛГОРИТМІВ ІНДУКЦІЇ ДЕРЕВА РІШЕНЬ ДЛЯ АНАЛІЗУ ВЕЛИКИХ ОБСЯГІВ ДАНИХ

V. P. Martsenyuk Dr., Prof., N. V. Milian

USE OF ALGORITHMS DECISION TREE INDUCTION FOR ANALYSIS LARGE DATA SECTIONS

Системи, що складають класифікатори, є одними з інструментів, які найчастіше використовуються для пошуку даних. Такі системи в якості вхідних даних приймають сукупність випадків, кожний з яких належить до однієї з невеликої кількості класів, описується її значеннями для фіксованого набору атрибутів і виводить класифікатор, який дозволяє точно передбачити клас, до якого належить новий тестовий випадок.

В процесі розвитку інформаційних технологій, а також систем збору і зберігання даних – баз даних (databases), сховищ даних (data warehousing), і з недавніх пір, хмарних сховищ, виникла проблема аналізу великих обсягів даних, коли аналітик або керівник не в змозі вручну обробити великі масиви даних і прийняти рішення. Зрозуміло, що аналітику необхідно якимось чином представити вихідну інформацію в більш компактному вигляді, з якою впорається людський мозок за прийнятний час.

Таким чином, існує велика кількість алгоритмів для аналізу великих обсягів даних. Одним з таких алгоритмів є індукція дерева рішень, яка має досить високу швидкість роботи, а вихідні дані легко розуміються людиною. Наприклад алгоритм індукції дерева рішень C4.5 будує дерево рішень, здатне передбачити клас для нових пацієнтів на підставі їх атрибутів. Отже, в кожній точці блок-схеми задається питання про значимість того чи іншого атрибута, і, в залежності від цих атрибутів, пацієнти потрапляють в певний клас.

ID3, C4.5 і CART алгоритми дерева рішень в результаті роботи яких будується дерево рекурсивно зверху-вниз. Більшість алгоритмів для індукції дерева рішень також наслідують підхід зверху-вниз, який починається з тренувального набору кортежів та пов'язаних з ними міток класу. Тренувальний набір рекурсивно розподіляється на менші підмножини при створенні дерева.

Алгоритм C4.5: генерування дерева рішень. Створить дерево рішень з навчальних кортежів розбиття даних, *D*.

Вхід:

- Розподіл даних *D*, який являє собою набір тренувальних кортежів та пов'язаних з ними міток класу;
- Attribute_list, набір атрибутів кандидата;
- Attribute_selection_method, процедура визначення критерію розбиття, що “найкраще” розділяє набір даних на окремі класи. Цей критерій складається з атрибуту розбиття splitting_attribute або точки розбиття (split-point), або підмножини розбиття (splitting subset).

Вихід: Дерево рішень

Метод:

```
(1) create a node N;
(2) if tuples in D are all of the same class, C, then
(3) return N as a leaf node labeled with the class C;
(4) if attribute list is empty then
(5) return N as a leaf node labeled with the majority class in D; // majority
voting
```

```
(6) apply Attribute selection method(D, attribute list) to find the "best"
splitting criterion;
(7) label node N with splitting criterion;
(8) if splitting attribute is discrete-valued and
multiway splits allowed then // not restricted to binary trees
(9) attribute list  attribute list ← splitting attribute; // remove splitting
attribute
(10) for each outcome j of splitting criterion
// partition the tuples and grow subtrees for each partition
(11) let Dj be the set of data tuples in D satisfying outcome j; // a partition
(12) if Dj is empty then
(13) attach a leaf labeled with the majority class in D to node N;
(14) else attach the node returned by Generate decision tree(Dj , attribute list)
to node N;
endfor
(15) return N;
```

- Якщо всі кортежі в *D* є тим самим класом, то вузол *N* перетворюється на лист і позначається цим класом (кроки 2 і 3). Необхідно звернути увагу, що етапи 4 та 5 є умовами припинення дії. Всі умови закінчення пояснюються в кінці алгоритму.
- В іншому випадку алгоритм викликає метод виділення атрибуту (Attribute_selection_method) для визначення критерію розбиття. Критерій розбиття показує, який атрибут тестується в вузлі *N*, визначаючи "найкращий" спосіб розділення або розбиття кортежів у *D* на окремі класи (етап 6). Критерій розбиття також показує, які гілки ростуть з вузла *N* по відношенню до результатів обраного тесту. Більш конкретно, критерій розбиття вказує атрибут розбиття (splitting attribute), а також може вказувати або точку розбиття (split-point), або підмножину розбиття (splitting_subset). Критерій розбиття визначається таким чином, що в ідеалі результати розбиття в кожній гілці настільки ж "чисті", наскільки це можливо. Розбиття чисте, якщо всі кортежі у ньому належать до того ж класу. Іншими словами, якщо розділити кортежі в *D* відповідно до взаємовиключних результатів критерію розбиття, то очікується, що результат розбиття буде настільки чистими, наскільки це можливо.
- Вузлом *N* позначено критерій розбиття, який служить тестом у вузлі (крок 7). Гілка росте з вузла N для кожного результату критерію розбиття. Кортежі в *D* розділені відповідно (кроки від 10 до 11). Є три можливі сценарії. Нехай *A* буде атрибутом розбиття. *A* має *v* чітких значень *{a1, a2, ..., av}* на основі навчальних даних.

Відмінності в алгоритмах дерева рішень містять вибрані атрибути створення дерева та механізми, що використовуються для обрізки. Базований алгоритм, описаний вище, вимагає одного проходження навчальних кортежів D для кожного рівня дерева [1].

Література

1. Han J. Data Mining Concepts and Techniques / Jiawei Han. – Waltham: Elsevier, 2012. – 740 с.