УДК 004.75

Гайдур Г.І., к.т.н.; **Прилєпов Є.В**., аспірант; **Попов М.І**., аспірант

ПРИНЦИПИ ПОБУДОВИ ДЕРЕВА РІШЕНЬ НА ОСНОВІ КЛАСИФІКАЦІЙНОГО АЛГОРИТМУ С4.5

Gaydur G.I., Pryliepov Y.V., Popov N.I. Principles of constructing a decision tree based on the classification algorithm C4.5.

The basic requirements for data structures were described. Determined the basic criteria for selecting a data attributes, necessary for building a tree. A decision tree constructing stages were described. From the first stage criterion for selecting an attribute was described. Also ware highlighted main variables, such as: set of examples, possible options, verification variable etc. For the fourth formula we have added explanation by using properties of entropy and its impact on the final stage. Finally we have described all cases for different situations in data classification process. The last publications on the field of data analysis, classification theory, statistics and information theory have been analyzed. The general advantages and disadvantages of decision trees and C4.5 algorithm were highlighted.

Key words: algorithm, analysis, classification, decision tree, machine learning.

Гайдур Г.І., Прилєпов Є.В., Попов М.І. Принципи побудови дерева рішень на основі класифікаційного алгоритму С4.5.

Описано основні вимоги до структур даних. Визначено основні критерії вибору атрибутів даних необхідних для побудови дерева. Описано етапи побудови дерева рішень. Також було описано основні критерії вибору атрибутів та основні змінні, такі як: множина прикладів, можливі варіанти, кількість прикладів тощо. В результаті були описані можливі ситуації в класифікації даних. Проаналізовано останні публікації по напрямку аналізу даних, теорії класифікації, статистики та теорії інформації. Виділено переваги та недоліки дерев рішень та алгоритму С4.5 загалом.

Ключові слова: алгоритм, аналіз, класифікація, дерево рішень, машинне навчання.

Гайдур Г.И., Прилепов Е.В., Попов Н.И. Принципы построения дерева решений на основе классификационного алгоритма С4.5.

Описаны основные требования к структурам данных. Определены основные критерии выбора атрибутов данных необходимых для построения дерева. Описаны этапы построения дерева решений. Также было описано основные критерии выбора атрибутов и основные переменные. В результате были описаны возможные ситуации в классификации данных. Проанализированы последние публикации по направлению анализа данных, теории классификации, статистики и теории информации. Проанализированы последние публикации по направлению анализа данных, теории классификации, статистики и теории информации. Выделены преимущества и недостатки деревьев решений и алгоритма С4.5 целом.

Ключевые слова: алгоритм, анализ, классификация, дерево решений, машинное обучение.

Вступ

Постановка задачі. Класифікаційний алгоритм С4.5 відноситься до методів машинного навчання – що ставить за мету отримання простих класифікованих виразів, які були б легко зрозумілі для людини. Перевагою таких методів є те, що під час роботи того чи іншого методу не потрібна участь людини.

Даний метод побудови дерев рішень, був вперше був запропонований Р. Куінленом (R. Quinlan). Цей метод використовується в одному з кращих алгоритмів побудови дерев рішень С4.5.

Перш ніж приступити до опису алгоритму побудови дерева рішень, необхідно визначити обов’язкові вимоги до структури даних і безпосередньо до самих даних, при виконанні яких алгоритм С4.5 буде працездатний [1]:

- **Опис атрибутів.** Дані, необхідні для роботи алгоритму, повинні бути представлені у вигляді таблиці. Вся інформація про об'єкти з предметної області повинна описуватися у вигляді кінцевого набору ознак (далі атрибути). Кожен атрибут повинен мати дискретне або числове значення. Самі атрибути не повинні змінюватися, а кількість атрибутів має бути фіксованим для всіх прикладів.
- **Явні класи.** Кожен приклад повинен бути асоційований з конкретним класом, тобто один з атрибутів повинен бути обраний в якості мітки класу.
- **Дискретні класи.** Класи повинні бути дискретними, тобто мати кінцеве число значень. Кожен приклад повинен однозначно ставитися до конкретного класу. Випадки, коли приклади належать до класу з ймовірними оцінками, виключаються. Кількість класів повинно бути значно менше кількості прикладів.

Аналіз літературних джерел. В ході дослідження, проведеного в рамках європейського проекту StatLog, був проведений аналіз статистичних методів; дискримінант аналіз, кластер-аналіз, дерев рішень (C4.5, AC2, CART, NewID, CN2, Itrule і т.д.) і нейронних мереж (багатошарові мережі, РБФ-мережі, карти Кохонена) для вирішення задач класифікації. Дані були взяті з різних предметних областей: розпізнавання образів, медична діагностика, молекулярна біологія, банківська справа і т.д.

Мета та задачі дослідження. Стрімкий розвиток інформаційних технологій, зокрема, прогрес в методах збору, зберігання і обробки даних дозволив багатьом організаціям збирати величезні масиви даних, які необхідно аналізувати. Обсяги цих даних настільки великі, що можливостей експертів вже не вистачає, що породило попит на методи автоматичного дослідження (аналізу) даних, який з кожним роком стає більше.

Дерева рішень - один з таких методів автоматичного аналізу даних. Перші ідеї створення дерев рішень сходять до робіт Ховленда (Hoveland) і Ханта (Hunt) кінця 50-х років XX століття. Однак, основною роботою, що дала імпульс для розвитку цього напрямку, стала книга Ханта (Hunt E.B.), Мерін (Marin J.) і Стоуна (Stone P.J) "Experiments in Induction", що побачила світ у 1966 р.

1. Етапи побудови дерева рішень

1.1 Теоретичне обґрунтування. Основне завдання полягає в побудові ієрархічної класифікаційної моделі у вигляді дерева з деякою множиною прикладів *T*, де ***T*** – множина прикладів в якій кожен елемент множини описується ***m*** атрибутами. Процес побудови дерева буде відбуватися зверху вниз. Спочатку створюється корінь дерева, потім нащадки кореня і т.д. При побудові дерев рішень особлива увага приділяється наступним питанням: вибору критерію атрибута, за яким піде розбиття, зупинки навчання і відсікання гілок [2].

На першому кроці ми маємо порожнє дерево (існує тільки корінь) і вихідна множина прикладів *T* (асоційованих з коренем). Потрібно розбити вихідну безліч на підмножини. Це можна зробити, вибравши один з атрибутів в якості перевірки. Тоді в результаті розбиття виходять n (по числу значень атрибута) підмножин і, відповідно, створюються n нащадків кореня, кожному з яких поставлено у відповідність своя підмножина, отримана при розбитті множини прикладів *T*. Потім ця процедура рекурсивно застосовується до всіх підмножин (нащадкам кореня) і т.д.

Розглянемо докладніше критерій вибору атрибута, за яким має піти розгалуження. Очевидно, що в нашому розпорядженні *m*, (по числу атрибутів) можливих варіантів, з яких ми повинні вибрати найбільш підходящий. Деякі алгоритми виключають повторне використання атрибута при побудові дерева, але в нашому випадку ми таких обмежень накладати не будемо. Будь-який з атрибутів можна використовувати необмежену кількість разів при побудові дерева.

Нехай ми маємо перевірку X (в якості перевірки може бути обраний будь-який атрибут), що приймає *n* значень *A1, A2...An*. Тоді розбиття *T* з перевірки *X* дасть нам підмножини *T1,*

T2...Tn, де *X* відповідно дорівнює *A1, A2...An*. Єдина доступна нам інформація - це, яким чином класи розподілені в безлічі *T* і його підмножин, одержаних при розбитті *X*. Саме це буде використано при визначенні критерію.

1.2 Математичне обґрунтування. Нехай $freq(C_j, S)$- кількість прикладів з деякої множини *S*, що відноситься до одного і того ж класу C_j. Тоді ймовірність того, що випадково обраний приклад з множини *S* буде належати до класу C_j.

$$P = \frac{freq(C_j, S)}{|S|}$$

Відповідно до теорії інформації, кількість інформації, що міститься в повідомленні, залежить від її ймовірності.

$$log_2\left(\frac{1}{P}\right) \quad (1)$$

Оскільки ми використовуємо двійковий логарифм, то вираз (1) дає кількісну оцінку в бітах.

Наступний вираз дає оцінку середньої кількості інформації, необхідної для визначення класу прикладів з множини *T*. У термінології теорії інформації вираз (2) називається ентропією множини T.

$$Info(T) = -\sum_{j=1}^{k} \frac{freq(C_j, T)}{|T|} * log_2\left(\frac{freq(C_j, T)}{|T|}\right) \quad (2)$$

Ту ж оцінку, але вже після розбиття множини *T* по *X*, дає наступний вираз:

$$Info(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} * Info(T_i) \quad (3)$$

Тоді критерієм для вибору атрибута буде наступна формула:

$$Gain(X) = Info(T) - Info_x(T) \quad (4)$$

Критерій (4) вираховується для всіх атрибутів. Вибирається атрибут, який максимізує даний вираз. Цей атрибут буде перевіркою в поточному вузлі дерева, потім з цього атрибуту проводиться подальша побудова дерева. Тобто в вузлі буде перевірятися значення з цього атрибуту і подальший рух по дереву буде проводитися в залежності від отриманого результату [3].

Даний принцип можна застосувати до отриманих підмножин T1, T2...Tn і продовжити рекурсивно процес побудови дерева, до тих пір, поки в вузлі не опиняться приклади з одного класу.

Якщо в процесі роботи алгоритму отримано вузол, асоційований з порожньою множиною (тобто жоден приклад не потрапив в даний вузол), то він позначається як лист, і в якості рішення листа вибирається найбільш часто зустрічаємий клас у безпосереднього попередника даного листа.

Варто пояснити чому критерій (4) повинен максимізуватися. З властивостей ентропії відомо, що максимально можливе значення ентропії досягається в тому випадку, коли всі його повідомлення різновірогідні. У даному випадку, ентропія (3) досягає свого максимуму коли частота появи класів в прикладах множини *T* різновірогідна. Необхідно вибрати такий

атрибут, щоб при розбитті по ньому один з класів мав найбільшу ймовірність появи. Це можливо в тому випадку, коли ентропія (3) матиме мінімальне значення і, відповідно, критерій (4) досягне свого максимуму.

Головне питання полягає в тому, як бути у випадку з числовими атрибутами? Зрозуміло, що слід вибрати якийсь поріг, за яким повинні порівнюватися всі значення атрибута. Нехай числовий атрибут має кінцеве число значень. Позначимо їх V_1, $V_2 ... V_n$. Попередньо відсортуємо все значення. Тоді будь-яке значення, що лежить між V_i та V_{i+1}, ділить всі приклади на дві множини: ті, що лежать зліва від цього значення V_1, $V_2 ... V_i$, і ті, що праворуч V_{i+1}, $V_{i+2} ... V_n$. Як порогу можна вибрати середнє між значеннями V_i та V_{i+1}

$$TH_i = \frac{V_i + V_{i+1}}{2}$$

Таким чином, суттєво спрощується завдання знаходження порогу та приведення до розгляду всього n-1 потенційних порогових значень $TH_1, TH_2 ... TH_{n-1}$.

Формули (2), (3) і (4) послідовно застосовуються до всіх потенційних порогових значень і серед них вибирається те, яке дає максимальне значення за критерієм (4). Далі це значення порівнюється зі значеннями критерію (4), розрахованим для інших атрибутів. Якщо з'ясується, що серед всіх атрибутів даний числовий атрибут має максимальне значення за критерієм (4), то в якості перевірки вибирається саме він. Слід зазначити, що всі числові тести є бінарними, тобто ділять вузол дерева на дві гілки.

Висновки

Дерева рішень є прекрасним інструментом в системах підтримки прийняття рішень та інтелектуального аналізу даних.

До складу багатьох пакетів, призначених для інтелектуального аналізу даних, вже включені методи побудови дерев рішень. В областях, де висока ціна помилки, вони бути мати найбільший попит.

В ході аналізу публікацій та літератури по напрямку аналізу даних, теорії класифікації, статистики та теорії інформації можна виділити наступні переваги та недоліки дерев рішень та алгоритму С4.5 загалом [4].

Переваги дерев рішень:

- На навчання дерев рішень потрібно набагато менше часу, ніж, наприклад, на навчання нейронних мереж.
- Результат роботи представляється в доступно інтерпретованому для людини вигляді. Класифікаційна модель, представлена у вигляді дерева є інтуїтивно зрозумілою для людини, на відміну від нейронних мереж.
- На вхід алгоритму дерев рішень можна подавати будь-яку кількість параметрів, алгоритм сам вибере найбільш значущі параметри і тільки вони будуть фігурувати в побудованому дереві. Це позбавляє користувача від необхідності визначати вхідні параметри.
- Точність прогнозу дерев рішень досить висока, порівнюючи з іншими методами побудови класифікаційних моделей (статистичні методи, нейронні мережі).
- Існують масштабовані алгоритми дерев рішень SLIQ, SPRINT, тобто з ростом числа прикладів час витрачається на навчання зростає лінійно для побудови дерев рішень на надвеликих базах даних.
- Алгоритми побудови дерев рішень мають методи спеціальної обробки пропущених даних.

- Класичні і сучасні методи статистики використовувані в задачах класифікації працюють тільки з числовими даними, дерева рішень успішно працюють як з числовими так і строковими значеннями. Недоліки дерев рішень:
- Існує проблема отримання оптимального дерева рішень.
- Можуть з'явитися занадто складні конструкції, які при цьому недостатньо повно представляють дані.
- Існують концепти, які складно зрозуміти з моделі, так як модель описує їх складним шляхом.
- Для даних, які включають категоріальні змінні з великим набором рівнів, більший інформаційний вплив присвоюється тим атрибутам, які мають більшу кількість рівнів.

Список використаної літератури

1. Quinlan J.R. C4.5: Programs for Machine Learning / Morgan J. Ross. - Boston: Kaufmann Publishers, 1993. - 302 p.

2. Шеннон К. Работы по теории информации и кибернетике / К. Шеннон. - М.: Иностранная литература, 1963. - 832 с.

3. Коршунов Ю.М. Математические основы кибернетики / Коршунов Ю.М. - М.: Энергоатомиздат, 1987. - 496 с.

4. Breiman L. Classification and Regression Trees / Breiman Leo, Friedman Jerome Charles J. Stone, Olshen R.A. - Washington: Taylor & Francis, 1984. - 368 p.

Автор статті

Гайдур Галина Іванівна - кандидат технічних наук, доцент, професор кафедри Інформаційної та кібернетичної безпеки, Державний університет телекомунікацій, Київ, Україна.

Прилєпов Євген Валерійович - аспірант кафедри Комп'ютерних наук, Державний університет телекомунікацій, Київ, Україна.

Попов Микита Ігорович - аспірант кафедри Електроніки, Національний авіаційний університет, Київ, Україна.

Authors of the article

Gaydur Galyna Ivanivna - candidate of Science (technic), associate professor, professor of Department of Information and cyber security, State University of Telecommunications, Kyiv, Ukraine.

Pryliepov Yevhen Valerievich - post graduate student of Department of Computer science, State University of Telecommunications, Kyiv, Ukraine.

Popov Mykyta Ihorovych - post graduate student of Department of electronics, National Aviation University, Kyiv, Ukraine.

Дата надходження в редакцію: 11.01.2018 Рецензент: д.т.н., проф. В.В. Вишнівський