VNUHCM - University of Science
Faculty of Information Technology

Introduction to Artificial Intelligence
Lab03 - Decision Tree ID3

Tô Gia Thuận - 19120389

1st January 2022

Contents

[1, 2, 3]

1 Tổng quan

1.1 Mô tả đồ án

- Áp dụng thuật toán cây quyết định ID3 để phân lớp tập dữ liệu iris flower

- Nguồn dữ liệu: `https://www.kaggle.com/uciml/iris`

1.2 Đánh giá mức độ hoàn thành

Công việc	Tỉ lệ hoàn thành
Cài đặt thuật toán ID3	100 %
Viết báo cáo	100 %
Tổng	100 %

2 Thuật toán cây quyết định ID3

2.1 Ý tưởng chính

- Thuật toán cây quyết định ID3 có cấu trúc là một cây, dựa trên thông tin kỳ vọng (entropy) để lựa chọn được thuộc tính tốt nhất và đưa vào node của cây. Với mỗi thuộc tính được chọn, ta chia dữ liệu vào các child node tương ứng với các giá trị của thuộc tính đó rồi tiếp tục áp dụng phương pháp này cho mỗi child node.

2.2 Hàm số Entropy

- Entropy (thông tin kì vọng):

$$Entropy = \sum_i^n p_i log_2(p_i)$$

- Đặc điểm: hàm số entropy sẽ đạt được giá trị lớn nhất khi p_i lớn nhất (khi $p_i = 1$ hay còn gọi là tinh khiết nhất) và nhỏ nhất khi p_i nhỏ nhất (khi $p_i = 0$ hay còn gọi là mờ nhất). Do đó, hàm số entropy được sử dụng rất tốt trong việc đo độ hỗn loạn của phép phân chia ID3.

2.3 Độ lợi thông tin

- Information gain (Độ lợi thông tin) phân lớp tập G theo thuộc tính x:

$$GainInformation(x) = Entropy(G) - Entropy_x(G)$$

- Giá trị Entropy của thuộc tính ($Entropy_x(G)$) càng nhỏ thì độ lợi thông tin càng cao (điều ta mong muốn). Đây cũng chính là yếu tố quan trọng tạo nên thuật toán ID3.

2.4 Đặc điểm riêng của thuật toán đối với tập dữ liệu liên tục

- Do đồ án sử dụng tập dữ liệu Iris flower để minh họa thuật toán. Trong khi đó, các thuộc tính của tập dữ liệu Iris đều có giá trị liên tục, nên để có thể sử dụng thuật toán phân chia ID3 thì ta cần phải rời rạc hóa từng thuộc tính bằng cách chọn ngưỡng cutoff thay vì phân loại thông thường đối với tập dữ liệu category (tập dữ liệu dạng category đã có sẵn những giá trị rời rạc trong mỗi thuộc tính, ta chỉ cần đếm số lượng mỗi loại giá trị như thông thường).

2.5 Mô tả các bước thực hiện

1. **Xây dựng hàm phụ trợ**
   - Hàm đọc dữ liệu từ tập tin CSV **read_file**: Đọc dữ liệu từ file CSV

   - Hàm phân chia tập dữ liệu (**train_test_split()**): Chia 2 mảng dữ liệu X, y đầu vào thành 4 tập X_train, y_train, X_test, y_test dùng để chạy với mô hình cây quyết định ID3.
   *** Lưu ý**: Để có thể phân chia tập dữ liệu một cách ngẫu nhiên, đồ án này sẽ sử dụng thêm thư viện **Random** của Julia

   - Xây dựng cấu trúc Node (**Node**): Bao gồm các thành phần của một Node của cây quyết định (như là name, child, entropy, ...). Bên trong cấu trúc **Node** sẽ có hai hàm khởi tạo (constructor) ứng với đầu vào 4 tham số và đầu vào 5 tham số.

   - Xây dựng cấu trúc cây quyết định (**DecisionTree**) : Bao gồm các thành phần của một cây quyết định ID3 (như là root, depth, ...). Bên trong sẽ có thêm một hàm khởi tạo DecisionTree.

   - Hàm đánh giá thuật toán dựa trên độ đo accuracy (**accuracy()**): đầu vào gồm 2 mảng giá trị, 1 mảng là gồm các giá trị dự đoán và 1 mảng là các giá trị chính xác. Hàm sẽ tính độ chênh lệch khác nhau của các giá trị giữa 2 mảng.

2. **Xây dựng hàm phân chia ID3**
   - **Tên hàm**: **make_split**
   - **Đầu vào** (4 tham số): gồm Node (node đang xét), và 2 mảng dữ liệu X_train, y_train (mảng dữ liệu 2 chiều).
   - **Hoạt động**: Hàm sẽ thực hiện duyệt từng thuộc tính (ứng với từng cột) đối với mảng 2 chiều X_train. Ở mỗi lần duyệt sẽ tìm ra ngưỡng cutoff phù hợp nhất của thuộc tính đó dựa vào giá trị entropy (gán lần lượt các giá trị của thuộc tính làm cutoff sau đó tính entropy của thuộc tính **(*)**, tại giá trị nào khiến cho entropy nhỏ nhất thì giá trị đó sẽ được chọn làm ngưỡng cutoff), sau đó tính độ lợi thông tin của thuộc tính đó. Trải qua tất cả các lần duyệt, ta sẽ tìm được thuộc tính có độ lợi thông tin lớn nhất và 2 phần trái, phải và ngưỡng cutoff của thuộc tính đó, lưu tất cả thông tin này lần lượt vào 2 node con của node đang xét.
   (*): Mảng dữ liệu X_train sẽ được chia làm 2 phần trái và phải ứng với những giá trị <= giá trị cutoff và những giá trị > giá trị cutoff. Ta sẽ lần lượt tính giá trị entropy đối với phần bên trái và phần bên phải. Cộng 2 giá trị entropy này lại ta sẽ được giá trị entropy của thuộc tính.

3. **Xây dựng cây quyết định**
   - **Tên hàm**: **fit**
   - **Đầu vào** (4 tham số): gồm mảng dữ liệu X_train, y_train và đối tượng cây (được tạo ra từ hàm khởi tạo của **struct DecisionTree**), và mảng tên thuộc tính (tên của các cột dữ liệu)
   - **Hoạt động**:
     - Bước 1: Khởi tạo node root của cây quyết định.
     - Bước 2: Đưa node root vào hàng đợi, lặp cho đến khi hàng đợi rỗng.
       Bên trong vòng lặp: lấy phần tử cuối cùng của hàng đợi (ứng với một node), tiến hành phân chia node bằng hàm **make_split()** để tạo ra các node con. Nếu hàm

make_split() trả về những node con mới thì sẽ thêm những node con mới đấy vào hàng đợi và tiếp tục vòng lặp. Nếu hàm **make_split()** không trả về node con mới (trả vễ rỗng) thì sẽ tiến hành gán nhãn cho node đó.
Kết thúc hàm trả về cây quyết định.

4. **Xây dựng hàm dự đoán**
   - **Tên hàm**: **predict**
   - **Đầu vào**: đối tượng cây (đã được xây dựng ở hàm **fit**) và mảng dữ liệu X_test.
   - **Hoạt động**: Đưa vào tập dữ liệu test và thực hiện dán nhãn cho tập dữ liệu test. Bằng cách duyệt trên cây quyết định đã được xây dựng, ta xác định được nhãn tương ứng với mỗi dòng giá trị trong tập dữ liệu X_test. Kết thúc hàm trả về mảng dữ liệu gồm các nhãn tương ứng với tập dữ liệu X_test, đó cũng chính là kết quả của thuật toán.

3 Kiểm thử

- Sử dụng độ đo accuracy để đánh giá độ chính xác của kết quả dự đoán và kết quả thật sự. Sử dụng hàm **accuracy** đã được cài đặt từ trước để tính được tỉ lệ phần trăm chính xác của các nhãn trong tập kết quả dự đoán.

- Đo với tập train: Sử dụng chính tập dữ liệu dùng để train cây quyết định (X_train).

```
Accuracy of train dataset: 0.97
```

- Đo với tập test: Sử dụng tập dữ liệu test (X_test)

```
Accuracy of test dataset: 0.98
```

***Lưu ý:** Tập dữ liệu train và test được random theo tỉ lệ 2/3 và 1/3, với độ ngẫu nhiên cố định = 30 (đảm bảo kết quả giống nhau sau mỗi lần chạy).

4 Đánh giá thuật toán

- Ưu điểm:
  - Thuật toán dễ hiểu và dễ cài đặt.
  - Có thể xử lý đối với dữ liệu số và dữ liệu phân loại (liên tục và rời rạc).
  - Co thể xử lý với kích thước dữ liệu lớn và khá nhanh.

- Nhược điểm:
  - Dễ bị overfiting, phải dùng một số biện pháp (như cố định độ sâu cây, cắt tỉa,...).
  - Không đảm bảo xây dựng được cây tối ưu nhất.

References

[1] v Tô Hoài Việt Lê Hoài Bắc. *Cơ sở trí tuệ nhân tạo*. NXB Khoa học và Kỹ thuật, 2014.

[2] Dorsa Sadigh Percry Liang. *Course: Artificial Intelligence: Principles and Techniques*. Stanford University, 2019.

[3] Tiep Vu. Decision trees (1): Iterative dichotomiser 3. `https://machinelearningcoban.com/2018/01/14/id3/`, 2018.