TRƯỜNG ĐẠI HỌC THỦY LỢI

**KHOA CÔNG NGHỆ THÔNG TIN**

A blue and white logo

Description automatically generated with low confidence

**BÀI TẬP LỚN**

HỌC PHẦN: HỌC MÁY

**ĐỀ TÀI: DỰ ĐOÁN QUYẾT ĐỊNH NGHỈ VIỆC CỦA NHÂN VIÊN**

Giáo viên hướng dẫn: Nguyễn Huy Đức

Sinh viên/nhóm sinh viên thực hiện:

1. Trần Thanh Thảo, lớp 62TH-NB

2. Vũ Thanh Sơn, lớp 62TH-NB

3. Nguyễn Hoàng Sơn, lớp 62TH-NB

**Hà Nội, năm 2022**

**Phần 1: Lý thuyết**

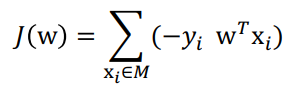
1. **Perceptron**

* Là thuật toán phân lớp đơn giản giúp tìm ra 2 ranh giới siêu phẳng cho bài toán phân lớp nhị phân (chỉ có 2 lớp dữ liệu).
* Cách xây dựng hàm mất mát, và tìm nghiệm bài toán
  + PT đường boundary:

*f*w(x) = w1.x1 + w2.x2 + … + wd.xd + w0 = 0

*f*w(x) = wT.x = 0

* w = (w0, w1,…, wd) : vector hệ số
* w0 : số hạng tự do
* x = (x0, x1,…, xd): vector đặc trưng
  + Hàm mất mát:



* + Nghiệm của bài toán w: tìm w sao cho hàm mất mát đạt giá trị nhỏ nhất là 0 (không có điểm dữ liệu nào bị phân lớp sai)
    - Với mỗi điểm phân lớp đúng -> kết thúc thuật toán
    - Với mỗi điểm phân lớp sai nghiệm w được cập nhật lại:



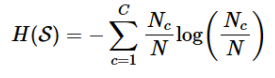
* Các bước thực hiện:
  + B1: tại thời điểm t=0, chọn ngẫu nhiên 1 vector hệ số w0
  + B2:
    - tại thời điểm t:
    - nếu không có điểm dữ liệu nào bị phân lớp lỗi thì dừng thuật toán.
    - nếu có điểm bị phân lớp lỗi thì cập nhật lại giá trị của w.

wt+1 = wt + yi.xi

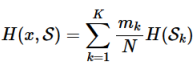
* B3: thay đổi t = t+1 quay lại bc 2

1. **ID3**

* ID3 là một thuật toán decision tree được áp dụng cho các bài toán classification mà tất cả các thuộc tính đều ở dạng categorical.
* Công việc thực hiện của bài toán:
  + Sử dụng hàm “entropy” làm thước đo trong khi tạo cây quyết định (decision tree) trong thuật toán ID3(Classification and Regression Tree)
* Các bước thực hiện:
  + **B1:** tính entropy trên toàn bộ tập S
    - H(S) : entropy trên toàn bộ tập dữ liệu S
    - S: tập dữ liệu huấn luyện(gồm cả x và y)
    - N: tổng số mẫu dữ liệu huấn luyện
    - Nc: số các mẫu dữ liệu trong tập S đc gán nhãn c
    - C: tổng số nhãn dữ liệu khác nhau trong tập S
    - c: chỉ số của nhãn dữ liệu



* **B2:** Tính entropy của thuộc tính x trên tập S
  + x: là thuộc tính đang xét
  + K: số giá trị của thuộc tính đó
  + mk: số mẫu có thuộc tính x và có giá trị là k
  + Sk: tập các mẫu sao cho thuộc tính x có giá trị là k
  + H(Sk): entropy trên tập dữ liệu trong S có thuộc tính là x và có giá trị k



* **B3:** tính *information gain* trên thuộc tính x



* Thay vì tìm thuộc tính có information gain cao nhất thì ta tìm thuộc tính có entropy nhỏ nhất



* Điều kiện dừng:
  + TH1: nếu tất cả các mẫu trong node con thuộc cùng 1 lớp C (entropy=0) thì node đó đc gán nhãn C.
  + TH2: nếu node con là rỗng (không có dữ liệu) thì node đó sẽ đc gán = nhãn phổ biến nhất trong tập S.
  + TH3: nếu không còn thuộc tính nào để phân chia (các thuộc tính đều đc xét hết) 🡪 node lá đó sẽ đc gán = nhãn phổ biến nhất.

1. **CART**

* Là một kỹ thuật phi tham số hữu ích có thể được sử dụng để giải thích một biến phụ thuộc liên tục hoặc phân loại dưới dạng nhiều biến độc lập. Các biến độc lập có thể liên tục hoặc phân loại. CART sử dụng phương pháp phân vùng thường được gọi là “**chia để trị**”. Thuật toán CART hoạt động để tìm biến độc lập tạo ra nhóm đồng nhất tốt nhất khi tách dữ liệu.
* Công việc thực hiện của bài toán:
  + Sử dụng chỉ số “gini index” làm thước đo trong khi tạo cây quyết định (decision tree) trong thuật toán CART (Classification and Regression Tree), chỉ số “gini index” càng cao càng tốt.
* Các bước thực hiện:
  + **B1:** Tính giá trị Gini

Gini =

* + - C: số lớp cần phân loại (số nhãn trong 1 thuộc tính)
    - N: là tổng số lượng phần tử ở node đó
* **B2:** Tính chỉ số gini\_index.

Gini\_index = gini(p) - gini()

* + - gini(p): chỉ số gini ở node cha
    - K: số node con được tách ra
    - gini(ck): chỉ số gini ở node con thứ k
    - M: số phần tử ở node p
    - mi : là số phần tử ở node con thứ i
* chọn ra thuộc tính có gini\_index lớn nhất 🡪 Ginisplit nhỏ nhất

gini()

* **B3:** Chia dữ liệu vào các nút con tương ứng với các giá trị của thuộc tính, thực hiện lặp tìm ra thuộc tính tốt nhất trong tập dữ liệu mới.
* Điều kiện dừng:
  + TH1: nếu tất cả các mẫu trong node con thuộc cùng 1 lớp C (entropy=0) thì node đó đc gán nhãn C
  + TH2: nếu node con là rỗng (không có dữ liệu) thì node đó sẽ đc gán = nhãn phổ biến nhất trong tập S
  + TH3: nếu không còn thuộc tính nào để phân chia (các thuộc tính đều đc xét hết) -> node lá đó sẽ đc gán = nhãn phổ biến nhất

**Phần 2: Ứng dụng trong thực tế**

***1. .Mô tả bài toán***

* Tên bài toán: dự đoán quyết định nghỉ việc của nhân viên.
* Input: tập dữ liệu gồm các thuộc tính và đã được gán nhãn.
* Ouput: kết quả dự đoán nhân viên có nghỉ việc hay không.
* tóm tắt công việc thực hiện của bài toán:
* Đọc dữ liệu từ file
* Chuẩn hóa dữ liệu
* Chia dữ liệu theo tỷ lệ 70% train, 30% test, không xáo trộn.
* Sử dụng các thuật toán để giải quyết bài toán:
  + Perceptron
  + ID3
  + CART
* Dự đoán quyết định nghỉ việc của nhân viên.

***2. Mô tả tập dữ liệu của bài toán***

* Dữ liệu có 199 vector gồm các thuộc tính và nhãn.
* Mô tả ma trận dữ liệu (X) gồm 199 dòng và 5 cột, mỗi cột tương ứng với một thuộc tính lần lượt là: tenure, age, income, employ, gender . Nhãn lớp (Y) gồm 199 dòng và 1 cột tương ứng là: Retire.
* Mô tả thuộc tính:
* Tên và ý nghĩa.

|  |  |
| --- | --- |
| **Tên thuộc tính** | **Ý nghĩa** |
| Tenure | Nhiệm kỳ |
| Age | Tuổi tác |
| Income | Thu nhập |
| Employ | Số năm làm việc |
| Gender | Giới tính |

* Mô tả nhãn:
* Tên và ý nghĩa:

|  |  |
| --- | --- |
| **Tên nhãn** | **Ý nghĩa** |
| Retire | Quyết định dược nghỉ hưu |

***3. Viết ứng dụng***

* Viết code của các thuật toán Perceptron, ID3 và CART để giải quyết bài toán.

***4. Phân tích kết quả của chương trình***

\* Cart – ID3:

- Tỷ lệ dự đoán đúng trên tập test: 95%

- Tỷ lệ dự đoán sai trên tập test: 5%

\* Perceptron:

- Tỷ lệ dự đoán đúng trên tập test: 93,33%

- Tỷ lệ dự đoán sai trên tập test: 6,71%

=> Vậy 2 thuật toán phù hợp để xét bài toán là CART và ID3

**Kết luận**

Dựa vào các thuộc tính đặc biệt là tuổi tác sẽ đưa ra được kết quả cuối cùng của người nhân viên có đủ điều kiện để nghỉ hưu hay không

**Tài liệu tham khảo**

<https://machinelearningcoban.com/2018/01/14/id3/>

<https://www.kaggle.com/>