NHÓM 3:

1) ĐẶNG DIỄM LINH 18126023
2) PHẠM VÕ ĐỨC PHONG 18126007
3) NGUYỄN ĐỨC PHÚ 18126028
4) DƯƠNG QUANG VINH 18126038

BÁO CÁO ĐỀ TÀI MÔN LÝ THUYẾT ĐỒ THỊ

CHỦ ĐỀ:

CÂY QUYẾT ĐỊNH VÀ HỌC MÁY
(DECISION-TREE AND MACHINE LEARNING)

Mục lục:

I) GIỚI THIỆU VỀ CÂY QUYẾT ĐỊNH

Cây quyết định là một công cụ hỗ trợ quyết định có sử dụng cấu trúc tương tự như đồ thị cây, trong đó, mỗi node của cây tượng trưng cho một sự kiện và mỗi nhánh sẽ thể hiện kết quả của sự kiện, và các đường đi từ gốc đến node lá sẽ đại diện cho các quy tắc phân loại.

Trong lĩnh vực máy học (machine learning), cây quyết định là một kiểu mô hình dự đoán, nghĩa là đó là một ánh xạ từ các quan sát về sự vật/hiện tượng t71i các kết luận về giá trị mục tiêu.

a) Thuật ngữ phổ biến được sử dụng với cây quyết định

Thuật ngữ	**Định nghĩa**
Root node (Node gốc)	Đại diện cho toàn bộ hệ thống hoặc toàn bộ mẫu cần phân loại.
Splitting (phân chia)	Quá trình phân chia một tập hợp trong một node thành hai hay nhiều node con.
Node quyết định	Một node con khi được phân chia tiếp tục thành các node con, nó được gọi là node quyết định.
Node lá/node cuối cùng	Một node con không thể phân chia tiếp tục.
Pruning (cắt, tỉa cây)	Thao tác loại bỏ các node con trong tập hợp các **node quyết định**. Đây là quá trình đối lập với quá trình **splitting**.
Node cha – node con	Khi một node được chia thành các node con, nó được gọi là node cha và các node phân chia bên dưới là node con.

b) Mục tiêu của cây quyết định

Thông thường, biến mục tiêu của mô hình này là biến phân loại (categorical) và cây quyết định trong trường hợp này là cây định danh (classification decision tree).

Cây định danh được sử dụng nhằm mục đích:

- Tính toán khả năng một phép thử/hồ sơ (record) nằm trong từng phân loại.
- Phân loại phép thử/hồ sơ (record) bằng cách gán nó với phân loại gần giống nhất.

Ngoài ra, cây quyết định cũng có thể áp dụng trong việc ước tính giá trị của một biến mục tiêu thuộc loại biến liên tục *(cây quyết định hồi quy [regression decision tree]*. Tuy nhiên, đối với loại biến này, các mô hình khác như mô hình hồi quy hoặc neural network sẽ phổ biến hơn.

c) Ứng dụng của cây quyết định phân loại

Đến nay, cây định danh (classification decision tree) đã được khai thác và áp dụng trong nhiều lĩnh vực khác nhau, bao gồm:

- Tài chính: phân loại khách hàng với rủi ro tín dụng; phân loại khách hàng vay vốn ngân hàng…
- Y học: phân loại bệnh nhân theo các thuộc tính như độ tuổi, chỉ số BMI,… với biến mục tiêu là tình trạng sức khỏe, có bệnh hay không bệnh (biến phân loại)
- Sinh học: có thể áp dụng trong việc phân loài.
- Nông nghiệp: đánh giá mức độ phù hợp của đất đai đối với giống cây trồng…

II) CÁC THUẬT TOÁN XÂY DỰNG CÂY QUYẾT ĐỊNH

Cây quyết định là một mô hình dạng cây. Do đó, để xây dựng cây quyết định đòi hỏi việc tìm cách phân chia tốt nhất sao cho sau khi phân chia, các mẫu ở mỗi node con hạn chế tối đa **mức độ hỗn loạn**.

a) Thuật toán ID3 và thuật toán C4.5

Ý tưởng:

+ Lựa chọn cách phân nhánh tối ưu dựa trên cơ sở tối đa hóa lượng thông tin nhận vào, nghĩa là giảm thiểu tối đa **độ hỗn độn và nhiễu loạn** trong từng node.

+ Các node phân nhánh cần **thể hiện tối đa thông tin** cần thiết để cây quyết định có thể phân loại chính xác đối tượng dữ liệu vào các tậpcon có chứa nhãn – giá trị/thuộc tính của biến đầu vào.

+ Khi 1 tập con đều mang giá trị là nhãn của phân nhánh hoặc chính tập con => độ hỗn loạn = 0, ngược lại thì độ hỗn loạn sẽ khác 0 và cao nhất = 1 (đối với biến mục tiêu là phân loại 0/1)

Thuật toán ID3:

Thuật toán ID3 sử dụng công thức Entropy và Information Gain để phân loại mẫu.

Trong đó, Entropy được tính như sau:

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

- p(j|t): xác suất điều kiện đối tượng dữ liệu mang thuộc tính j của biến mục tiêu trong node t.
- Log_2 của p(j|t) mang giá trị âm và càng tiến dần về 0 khi p(j|t) lớn => thêm dấu – để có dc giá trị dương tiến dần về 0.

⇨ *Như vậy, giá trị Entropy càng nhỏ => node càng chứa nhiều đối tượng dữ liệu có cùng thuộc tính j bất kì (do p(j|t) lớn).*

Information Gain được tính toán như sau:

Information Gain: $$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Information Gain được định nghĩa là khối lượng dữ liệu có được. Sau mỗi lần splitting (phân chia) một node, Information Gain càng lớn, độ hỗn loạn của dữ liệu trong các node con càng thấp.

Vì vậy, chúng ta sẽ tính toán Information Gain cho từng cách phân chia, sau đó **chọn ra Information Gain lớn nhất** để phân loại mẫu, tiếp tục thực hiện việc tính toán này cho các node con đến khi mọi node lá đều có độ hỗn loạn = 0.

Thuật toán C4.5

Là bản nâng cấp của thuật toán ID3, bổ sung thêm công thức GainRatio và SplitINFO như sau:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$$

Với:

$$SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log_2 \frac{n_i}{n}$$

Khi đó, **SplitINFO** chính là thông tin có được về số lượng đối tượng dữ liệu trong mỗi nhánh.

Việc sử dụng thêm 2 công thức này trong **thuật toán C4.5** sẽ giúp hạn chế việc lựa chọn các biến phân loại có quá nhiều thuộc tính và nhờ đó ngăn ngừa việc overfitting data (quá khớp với dữ liệu mẫu).

b) Thuật toán CART

Phương pháp CART lần đầu tiên được giới thiệu vào năm 1985 do nhà thống kê Leo Breiman cùng các cộng sự. CART được sử dụng chủ yếu để xây dựng cây quyết định mà chỉ **phân theo hai nhánh một lần**.

Ý tưởng:

+ Chọn ra các node có chứa đối tượng dữ liệu với khả năng tương đồng với nhau nhằm xác định phân loại đối tượng vào các nhóm, các lớp cho phù hợp.

+ Cần phải tính toán tính đồng nhất của mỗi node và dùng kết quả đó để tìm ra cách phân nhánh “split” tối ưu nhất.

Phương pháp CART sử dụng một trong hai công thức sau để tìm ra cách phân nhánh tối ưu.

Hệ số Goodness:

$$\Phi(s|t) = 2P_L P_R \sum_{j=1}^{\#classes} |P(j|t_L) - P(j|t_R)|$$

- t_L là node bên trái của phân nhánh đầu tiên của root node trong cây quyết định.
- t_R là node bên phải của phân nhánh đầu tiên của root node trong cây quyết định.
- P_L là tỷ lệ của số quan sát của node bên trái t_L trên tổng số quan sát của tập dữ liệu.
- P_R là tỷ lệ của số quan sát của node bên trái t_R trên tổng số quan sát của tập dữ liệu.
- P(j|tL): tỷ lệ số quan sát có thuộc tính j trên tổng số quan sát trong node bên trái.
- P(j|tR): tỷ lệ số quan sát có thuộc tính j trên tổng số quan sát trong node bên phải.

⇨ *Cách phân nhánh nào có giá trị cao nhất được tính từ công thức trên sẽ được dùng để xây dựng cây quyết định.*

Hệ số GINI:

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

- t là node bất kì có chứa các điểm dữ liệu mang thuộc tính j của biến mục tiêu và p là xác suất để 1 điểm dữ liệu trong t có thuộc tính j.
- Nếu như p = 1, Gini(t) = 0 và node t được công nhận là "pure" node (node thuần khiết và đóng vai trò là node lá trong cây quyết định).

GINI Index là thang đo mức độ đồng nhất/mức độ nhiễu loạn của thông tin hay sự khác biệt về giá trị mà mỗi điểm dữ liệu trong 1 tập hợp con, hoặc 1 nhánh node trên Decision Tree, có thể được sử dụng cho cả biến định tính và biến định lượng.

Như vậy, nhờ hệ số này, nhà phân tích có thể đánh giá sự tối ưu của mỗi cách phân nhánh thông qua mức độ "purity" của từng node trong mô hình cây:

- Nếu như tất cả các điểm dữ liệu thuộc về 1 node đều có chung 1 thuộc tính bất kì, node có sự đồng nhất và không có nhiễu loạn thông tin => **Gini Index = 0**.
- Trái lại, nếu tất cả quan sát không có cùng thuộc tính nào đó, hệ số Gini sẽ lớn và khiến cây trở nên hỗn loạn.

Tiếp theo, để tìm ra cách phân chia tốt nhất, ta dùng thêm công thức sau:

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

- n_i là số quan sát trong child node [node của nhánh được phân]
- n là số quan sát trong parent node [node được dùng để phân nhánh]

⇨ Hệ số Gini-Split càng nhỏ, cách phân nhánh càng tối ưu.

c) Sự khác nhau giữa CART và thuật toán ID3/C4.5

Tuy có chung một ý tưởng là tìm ra cách phân nhánh tốt nhất ở mỗi node, giữa CART và ID3 (hay C4.5) vẫn có những điểm khác nhau nhất định:

CART	**ID3/C4.5**
Chỉ phân được 2 nhánh mỗi lần	Phân được nhiều nhánh một lần từ một node cha.
Sử dụng hệ số Gini Index hoặc hệ số Goodness.	Sử dụng công thức Entropy kết hợp với Information Gain (ở C4.5 có thêm công thức Gain Ratio).
Dễ tính toán	Có sử dụng hàm log khi tính toán Entropy, do đó chi phí tính toán cao hơn.

III) ƯU – NHƯỢC ĐIỂM CỦA CÂY QUYẾT ĐỊNH

Decision Tree (cây quyết định) là một trong những phương pháp Classification được sử dụng phổ biến trong nhiều nghiên cứu dữ liệu, có tính hiệu quả cao và được ứng dụng rộng rãi trong mọi lĩnh vực khác nhau từ kinh tế, xã hội đến các môn khoa học tự nhiên.

a) Ưu điểm của cây quyết định

- Trực quan và đơn giản: mọi người khi đọc vào một cây quyết định đều dễ dàng hiểu về mô hình.
- Không cần áp dụng các phương pháp như "imputing missing values" hay loại bỏ các missing values này vì một số thuật toán có thể xử lý các dữ liệu trống/dữ liệu lỗi.
- Không sử dụng tham số, vì vậy không cần giả định ban đầu về các quy luật phân phối như trong thống kê, dẫn đến kết quả phân tích rất khách quan.
- Kết quả dự báo có tính chính xác cao, dễ thực hiện và training trong thời gian ngắn.
- Không sử dụng trực tiếp dữ liệu, vì vậy không đòi hỏi dữ liệu phải được biến đổi và chuẩn hóa.
- Phần nào nói lên được mối liên hệ giữa các biến dự đoán và biến mục tiêu một cách trực quan.

b) Nhược điểm của cây quyết định

- Chỉ hoạt động hiệu quả trên bộ dữ liệu đơn giản và có ít biến dữ liệu.
- Khi được áp dụng trên bộ dữ liệu phức tạp với nhiều biến và nhiều thuộc tính khác nhau, gây ra hiện tượng overfitting – quá khớp với dữ liệu được training, khiến tính chính xác của phân loại bị giảm xuống.
- Thuật toán cây quyết định thường chỉ áp dụng cho biến định tính, biến phân loại. Vì vậy, nếu phân loại sai có thể gây ra các sai lầm nghiêm trọng.
- Bộ dữ liệu training cần được cân đối và có chất lượng tốt, sạch, hoàn hảo.
- Thuật toán cây quyết định được xây dựng dựa trên cách tìm phương án phân nhánh tốt nhất tại một thời điểm và ở một node xác định, do đó không tính toán đến tính tối ưu của toàn bộ mô hình…

IV) VẤN ĐỀ PRUNNING (CẮT TỈA CÂY)

Như đã phân tích phía trên, vấn đề "Overfitting" là một vấn đề cần quan tâm đối với bất kì chuyên gia phân tích dữ liệu nào muốn xây dựng một cây quyết định cho dự án của mình.

Để giải quyết vấn đề trên, các nhà nghiên cứu đã đưa ra các **phương pháp Prunning** dùng để giảm kích thước của cây bằng cách giảm bớt các "section" – những phần kém hợp lý trong toàn bộ mô hình cây. Điều này đồng thời cũng hạn chế tính phức tạp của các quy luật phân loại.

Hiện nay, **prunning** một cây quyết định bao gồm 2 phương pháp chính là **pre-prunning** và **post-prunning**.

a) Pre-prunning

Ý tưởng của phương pháp này là sẽ cố gắng tối ưu cây **trước khi** nó được xây dựng hoàn toàn.

Một số nguyên tắc áp dụng **pre-prunning** bao gồm:

- Ngừng phân nhánh khi tất cả quan sát đều nằm trong cùng một phân lớp.
- Ngừng phân nhánh khi tất cả giá trị của biến dữ liệu là như nhau.

Ngoài ra, một số nguyên tắc khác yêu cầu cao hơn:

- Ngừng phân nhánh khi số quan sát trong node thấp hơn so với giá trị tối thiểu/ngưỡng xác định trước đó.
- Ngừng phân nhánh khi việc phân nhánh không thể cải thiện mức độ đồng nhất (sử dụng hệ số Gini Index hay Entropy).
- Ngừng phân nhánh khi cách phân phối của các quan sát trong class đọc lập với thuộc tính/các biến của dữ liệu: thường sử dụng ***kiểm định chi bình phương*** để kiểm tra.
- Ngừng phân nhánh khi hệ số Errors của node thấp hơn một ngưỡng cho trước.

b) Post-prunning

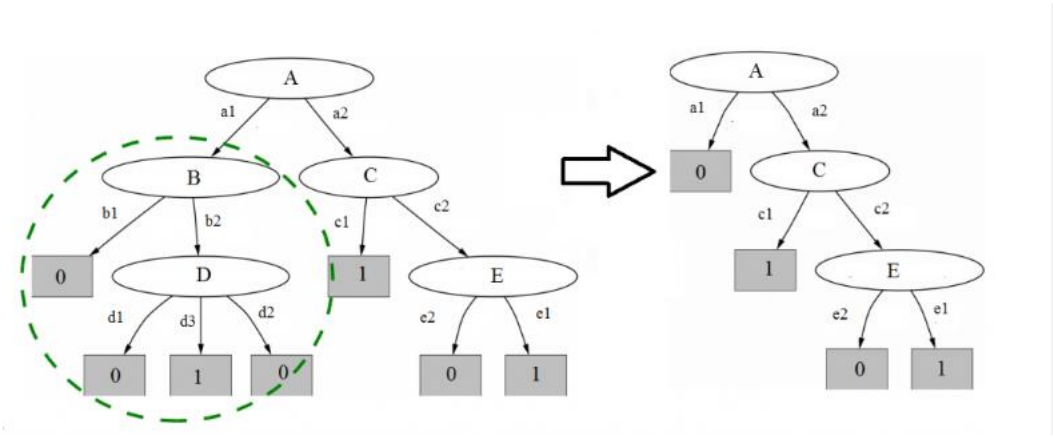
Phương pháp **Post-prunning** là dạng phân nhánh/cắt tỉa lại cây **sau khi** cây đã được xây dựng hoàn toàn, trong đó, chọn ra một số cây con trên mô hình để tiến hành điều chỉnh.

Phương pháp **Post-prunning** được tiến hành lần lượt theo các bước sau:

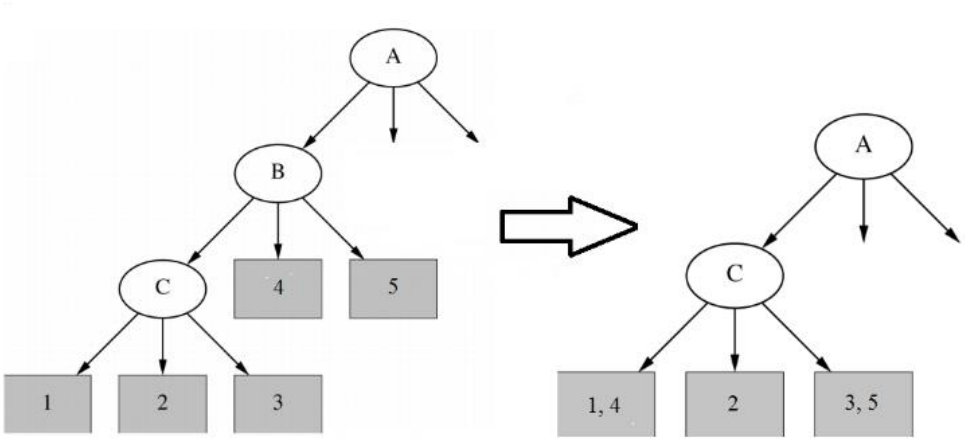
- Xây dựng 1 cây quyết định hoàn chỉnh cho toàn bộ dữ liệu training.
- Tính toán tính chính xác/hiệu quả của cây.
- Chọn ra những cây con kém hiệu quả và từ đó xác định phương thức điều chỉnh.

Về cách điều chỉnh những nhánh con kém hiệu quả, **post-prunning** có hai loại điều chỉnh chính:

- **Subtree replacement:** thu gọn cây theo chiều từ dưới lên, nghĩa là loại bỏ hay gộp chung một phần các node và phân lại thành một nhánh duy nhất nếu hệ số Error được cải thiện.



- **Subtree raising:** điều chỉnh cây theo chiều từ trên xuống, loại bỏ node phân nhánh và đưa những quan sát trong node này xuống node phân nhánh bên dưới.



V) KẾT LUẬN

Có thể thấy, việc ứng dụng **cây quyết định** và đặc biệt là **cây định danh** trong các lĩnh vực xã hội – đời sống – khoa học đã góp phần không nhỏ vào việc thực hiện các dự báo chính xác, hiệu quả trong thời gian cực kì ngắn.

Nắm rõ các phương pháp xây dựng **cây quyết định** và một số nguyên tắc cắt tỉa cây (phương pháp **prunning**) sẽ giúp mô hình trở nên súc tích và có độ tối ưu tốt nhất.

Tài liệu tham khảo:

https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb

https://bigdatauni.com/vi/

https://en.wikipedia.org/wiki/Decision_tree

http://machinelearningcoban.com/