PHÂN LỚP VỚI CÂY QUYẾT ĐỊNH
DECISION TREE

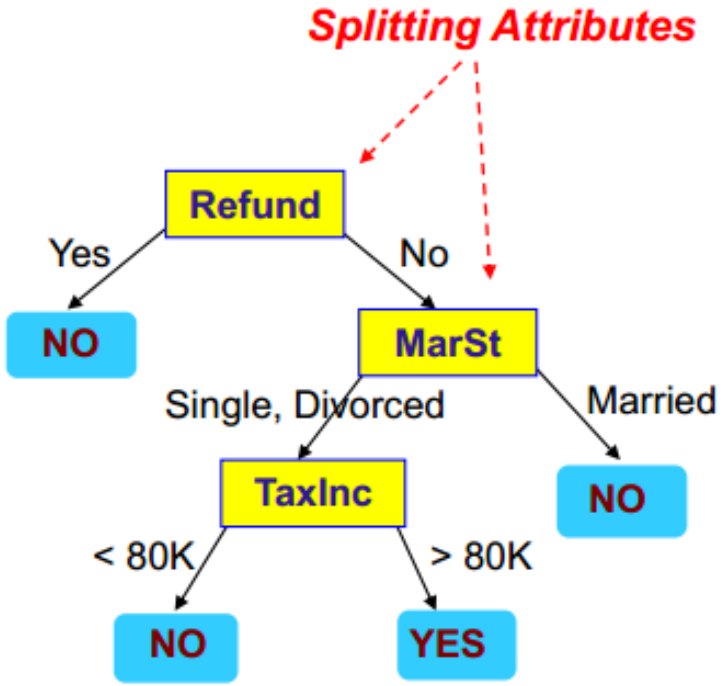
NỘI DUNG

- Giới thiệu về cây quyết định (Decision Tree, DT)
- Giải thuật học DT
- Độ lợi thông tin và lựa chọn thuộc tính
- Đánh giá hiệu năng
- Bài tập và thực hành

Giới thiệu

	categorical	categorical	continuous	class
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



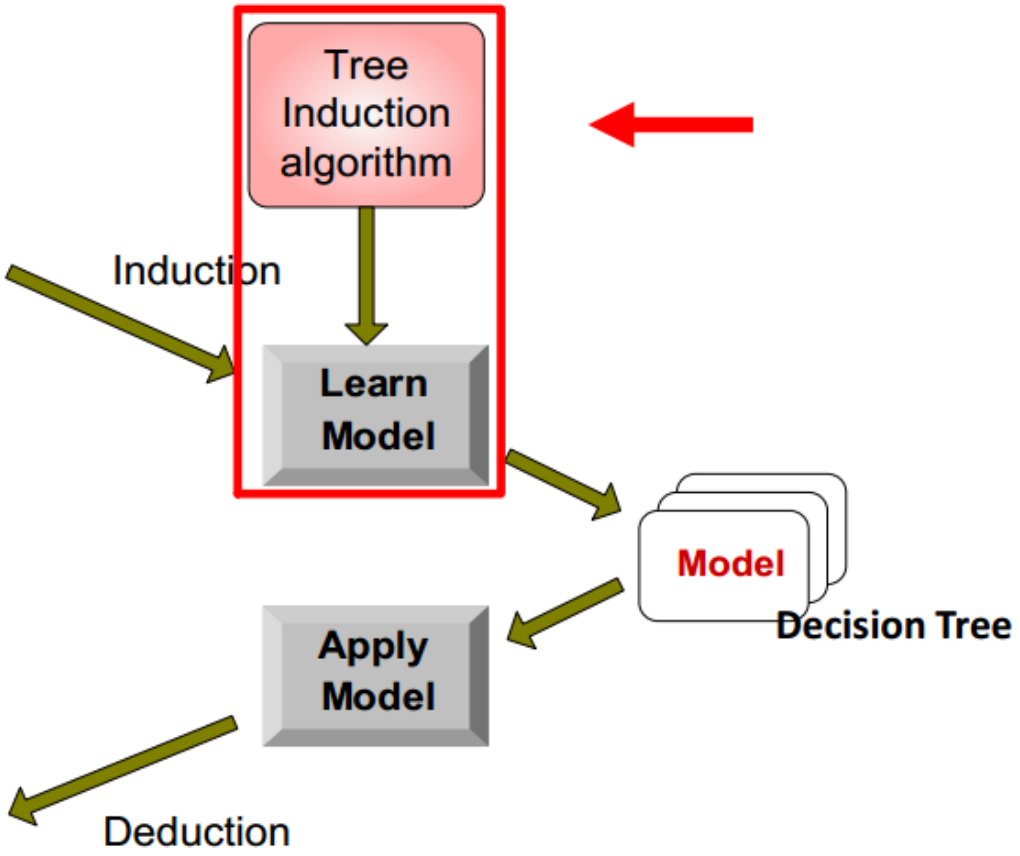
Model: Decision Tree

Cây quyết định cho phân lớp dữ liệu

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?



Cây quyết định

- Cây quyết định (Decision tree): Là một công cụ phổ biến để phân lớp và dự báo
- Tốc độ học tương đối nhanh so với các phương pháp khác
- Cây có thể được chuyển thành tập luật một cách dễ dàng
- Độ chính xác khá tốt
- Được áp dụng thành công trong nhiều bài toán ứng dụng thực tế.

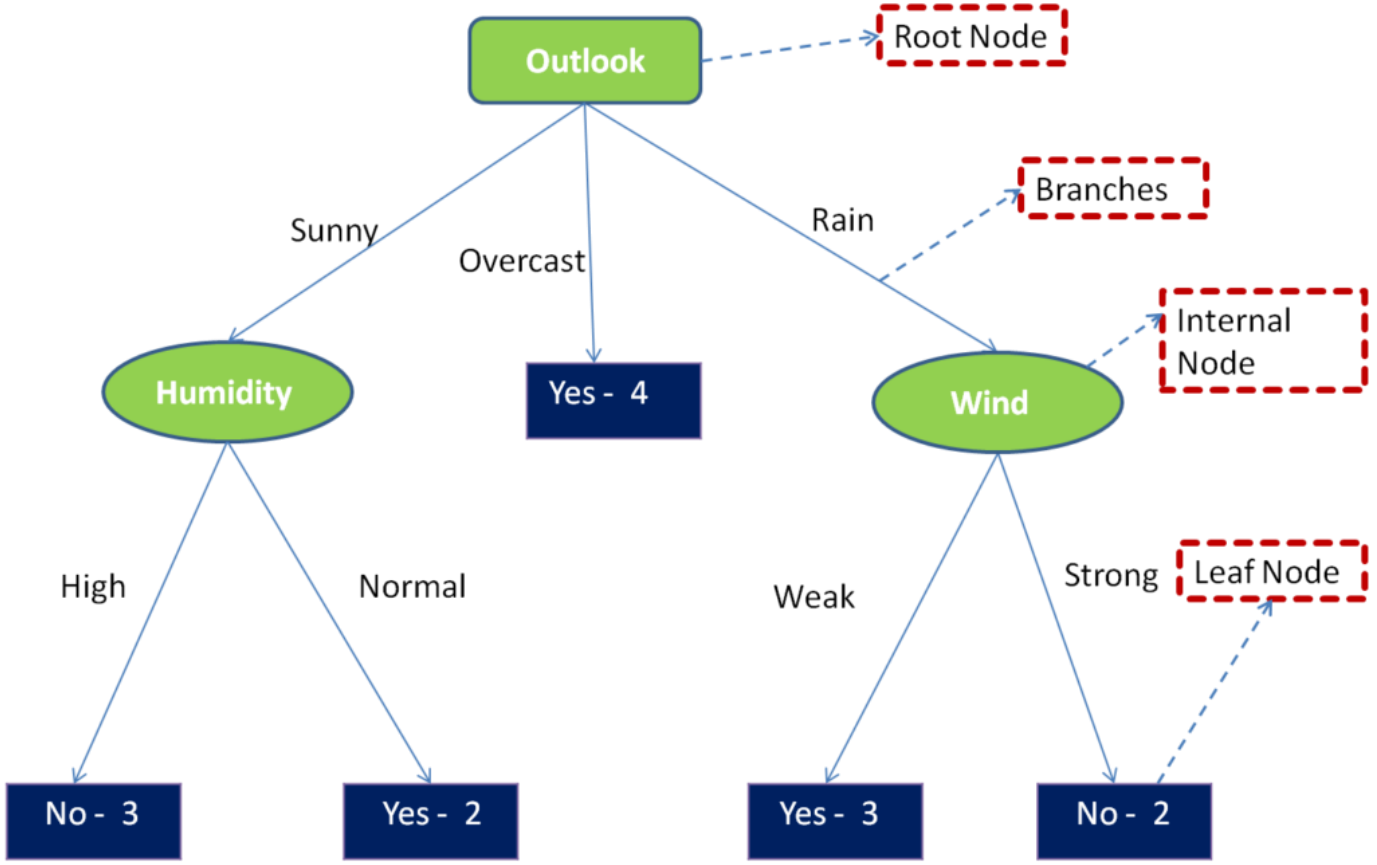
Học cây quyết định

- Học cây quyết định (Decision tree learning)
  - Để học (xấp xỉ) một hàm mục tiêu có giá trị rời rạc (discrete valued target function), gọi là hàm phân lớp
  - Hàm phân lớp được biểu diễn bởi một cây quyết định
- Cây quyết định có thể học với dữ liệu có chứa nhiễu/lỗi (noisy data).

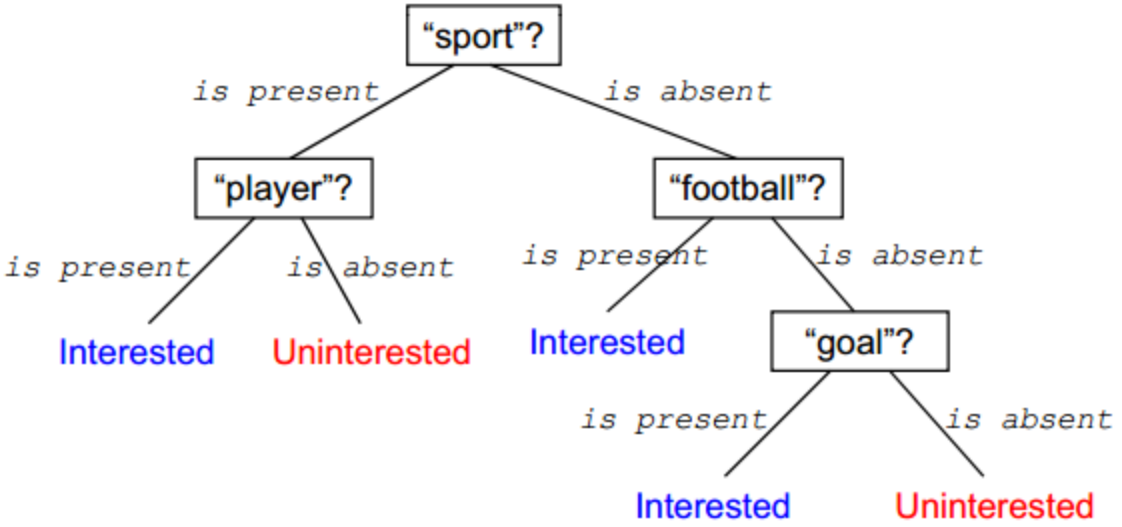
Biểu diễn DT

- Mỗi nút trong (internal node) biểu diễn *một thuộc tính* của dữ liệu cần kiểm tra (test) các giá trị
- Mỗi nhánh (branch) từ một nút tương ứng với một giá trị có thể của thuộc tính gắn với nút đó
- Mỗi nút lá (leaf node) biểu diễn 1 nhãn lớp (a class label)
- Một cây quyết định học được sẽ phân lớp đối với một mẫu, bằng cách duyệt cây từ nút gốc đến một nút lá → nhãn lớp gắn với nút lá đó sẽ được gán cho mẫu dữ liệu cần được phân lớp.

- Ví dụ DT


Outlook
Root Node
Sunny
Overcast
Rain
Branches
Humidity
Yes - 4
Wind
Internal Node
High
Normal
Weak
Strong
Leaf Node
No - 3
Yes - 2
Yes - 3
No - 2

- Vd: Những tin tức được quan tâm?

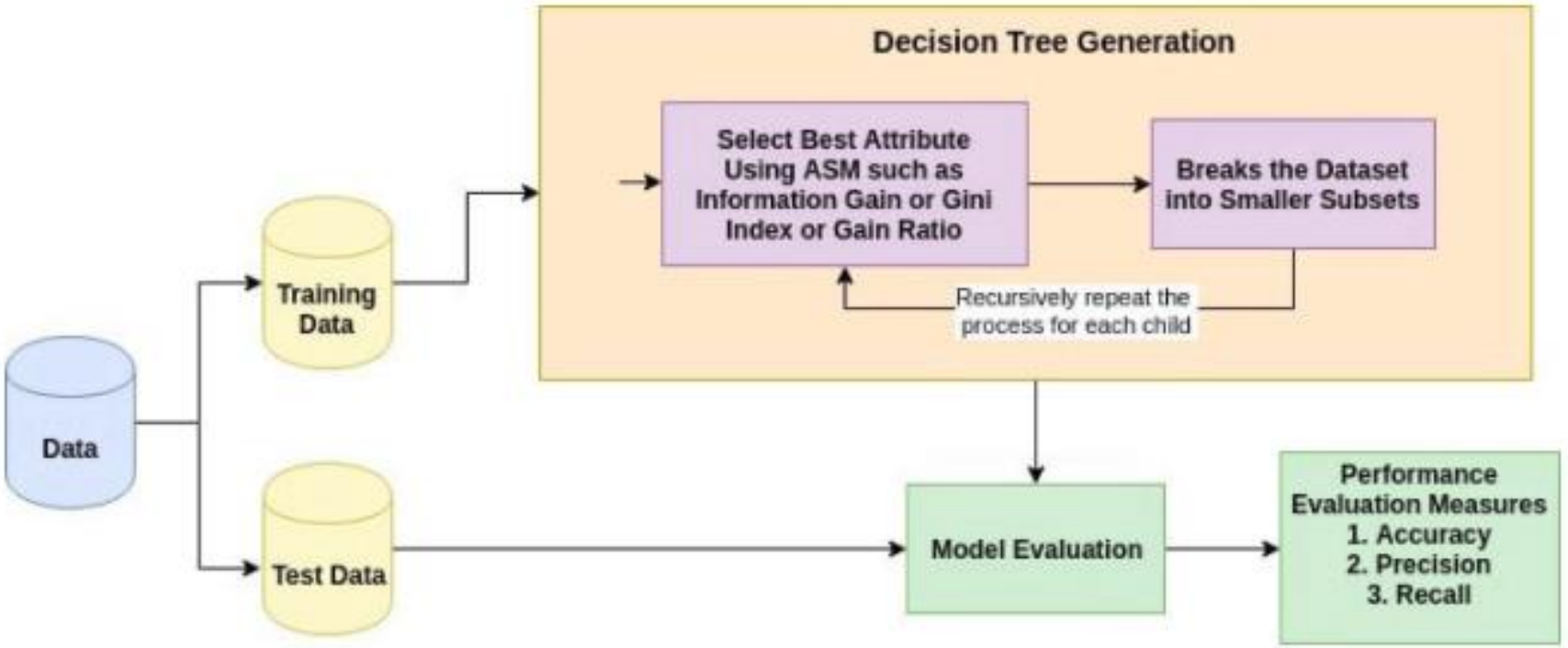


[("sport" is present) ∧ ("player" is present)] ∨

[("sport" is absent) ∧ ("football" is present)] ∨

[("sport" is absent) ∧ ("football" is absent) ∧ ("goal" is present)]

Dựng cây quyết định



Source: DataCamp

Dựng cây quyết định

- Xây dựng cây: thực hiện đệ quy chia tập mẫu dữ liệu huấn luyện cho đến khi các mẫu ở mỗi nút lá thuộc cùng một lớp.
  - Các mẫu huấn luyện xuất phát nằm ở nút gốc
  - Chọn một thuộc tính để phân chia tập mẫu huấn luyện thành các nhánh
  - Tiếp tục lặp việc xây dựng cây quyết định cho các nhánh, quá trình dừng khi:
    - Tất cả các mẫu đều được phân lớp
    - Không còn thuộc tính nào có thể dùng để chia mẫu

Các giải thuật xây dựng DT

- Phân loại:
  - **CLS** (Concept Learning System)
  - **ID3** (Iterative Dichotomiser 3)
  - C4.5
- Hồi quy: CART

Giải thuật CLS

- B1. Tạo một nút gốc T gồm tất cả các mẫu huấn luyện
- B2. Nếu tất cả các mẫu trong T có nhãn "Yes" thì gán nhãn nút T là "Yes" và dừng.
- B3. Nếu tất cả các mẫu trong T có nhãn "No" thì gán nhãn nút T là "No" và dừng.
- B4. Nếu các mẫu trong T có cả "Yes" và "No" thì
  - Chọn một thuộc tính A có các giá trị a_1, .., a_n
  - Chia tập mẫu theo giá trị của A thành các tập con T_1,.., T_n.
  - Tạo n nút con Ti (i=1..n) với nút cha là nút T
- B5. Thực hiện lặp lại cho các nút con Ti (i=1..n) và quay lại B2.

Giải thuật CLS

- Nhận xét:
  - Tại bước 4 có thể chọn 1 thuộc tính tùy ý
  - Thứ tự các thuộc tính khác nhau sẽ cho ra cây có hình dạng khác nhau
  - Việc lựa chọn thuộc tính sẽ ảnh hưởng đến độ sâu, độ rộng, và độ phức tạp của cây.

Một số khái niệm

- Gọi S là tập mẫu dữ liệu cần xử lý, S sẽ được phân thành m lớp $\{C_1, C_2,.. C_m\}$
- $|C_i|$: lực lượng của C_i, đặt $s_i = | C_i |$, $s = |S|$
- A tập các thuộc tính $\{a_1,.., a_n\}$
- Phân hoạch S theo thuộc tính A thành n tập $\{S_1, S_2,.. S_n\}$.
- s_{ij} là số phần tử của lớp C_i trong tập S_j

Entropy – độ đo thông tin

- Entropy của tập thông tin S, ký hiệu E(S) hay E(s_1,.. s_n)

$$E(S) = E(s_1, s_2, .. s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s}$$

Entropy(S) = 0 : tập thông tin S là thuần nhất

Entropy(S) = 1 : tập thông tin S là độ nhiễu cao nhất

0 < Entropy(S) < 1 : tập thông tin S có số lượng mẫu thuộc các loại khác nhau

- Entropy của thuộc tính A ứng với tập S

$$E(S, A) = \sum_{j=1}^{n} \frac{s_{1j}+s_{2j}+..+s_{mj}}{s} E(S_j);$$

$$E(S_j) = \sum_{i=1}^{m} -\frac{s_{ij}}{|S_j|} \log_2 \frac{s_{ij}}{|S_j|}$$

Information Gain – độ lợi thông tin

- Information Gain hay Gain (thông tin thu được khi thực hiện phân nhánh): là đại lượng xác định hiệu xuất phân lớp các mẫu dữ liệu ứng với mỗi thuộc tính

$$Gain(S,A) = E(S) – E(S,A)$$

Giải thuật ID3 – ý tưởng

- Thực hiện giải thuật tìm kiếm tham lam (greedy search) đối với không gian các cây quyết định
- Xây dựng (học) một cây quyết định theo chiến lược top-down, bắt đầu từ nút gốc
- Ở mỗi nút, thuộc tính kiểm tra là thuộc tính có *khả năng phân loại tốt nhất* đối với *các ví dụ học gắn với nút đó*.
- Tạo mới một cây con (sub-tree) của nút hiện tại cho mỗi giá trị có thể của thuộc tính kiểm tra, và *tập học sẽ được tách ra* (thành các tập con) tương ứng với cây con vừa tạo
- Mỗi thuộc tính chỉ được phép xuất hiện tối đa 1 lần đối với bất kỳ một đường đi nào trong cây
- Quá trình phát triển (học) cây quyết định sẽ tiếp tục cho đến khi thỏa mãn điều kiện dừng (các mẫu đã được phân loại, hoặc không còn thuộc tính nào).

Giải thuật ID3

ID3 cho phân lớp nhị phân

- B1. Tạo một nút T gồm tất cả các mẫu huấn luyện
- B2. Nếu tất cả các mẫu trong T có nhãn "Yes" thì gán nhãn nút T là "Yes" và dừng.
- B3. Nếu tất cả các mẫu trong T có nhãn "No" thì gán nhãn nút T là "No" và dừng.
- B4. Nếu mẫu trong T có cả "Yes" và "No" thì
  - Chọn thuộc tính A có Gain(S,A) lớn nhất để phân nhánh
- B5. Thực hiện lặp cho các nút con Ti (i=1..n) và quay lại B2.

Ví dụ: S

	Quang cảnh	Nhiệt độ	Độ ẩm	Gió	Chơi tennis
D1	Nắng	Nóng	Cao	Nhẹ	Không
D2	Nắng	Nóng	Cao	Mạng	Không
D3	Âm u	Nóng	Cao	Mạnh	Có
D4	Mưa	Ấm áp	Cao	Nhẹ	Có
D5	Mưa	Mát	TB	Nhẹ	Có
D6	Mưa	Mát	TB	Mạnh	Không
D7	Âm u	Mát	TB	Mạnh	Có
D8	Nắng	Ấm áp	Cao	Nhẹ	Không
D9	Nắng	Mát	TB	Nhẹ	Có
D10	Mưa	Ấm áp	TB	Nhẹ	Có
D11	Nắng	Ấm áp	TB	Mạnh	Có
D12	Âm u	Ấm áp	Cao	Mạnh	Có
D13	Âm u	Nóng	TB	Nhẹ	Có
D14	Mưa	Ấm áp	Cao	Mạnh	Không

Entropy(S) = $-\left(\frac{9}{14}\right)\log_2\left(\frac{9}{14}\right)-\left(\frac{5}{14}\right)\log_2\left(\frac{5}{14}\right)=0.94$

Gain(S, Quang cảnh) =

$E(S)-\left(\frac{5}{14}\right)E(S_{Nắng})-\left(\frac{4}{14}\right)E(S_{Âm\_u})-\left(\frac{5}{14}\right)E(S_{Mưa})=0.94-\left(\frac{5}{14}\right)0.971-\left(\frac{4}{14}\right)0.0-\left(\frac{5}{14}\right)0.0971=0.247$

Gain(S, Nhiệt độ) = 0.029

Gain(S, Độ ẩm) = 0.151

Gain(S, Gió) = 0.048

Entropy($S_{Nắng}$) = $-\left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right)-\left(\frac{3}{5}\right)\log_2\left(\frac{3}{5}\right)=0.97$

Gain($S_{Nắng}$, Nhiệt độ) = 0.57

Gain($S_{Nắng}$, Độ ẩm) = 0.97

Gain($S_{Nắng}$, Gió) = 0.019

Giải thuật ID3

Chú ý đối với **thuộc tính liên tục** ta sử dụng chỉ số GINI

$$GINI(t) = 1 - \sum_{j} p^2(j/t)$$

p(j/t) là tần suất của lớp j trong nút t

max = 1-1/nc khi các mẫu phân bố đều trên các lớp

min = 0 khi các mẫu thuộc về một lớp

Khi chia nút p thành k nhánh, chất lượng của phép chia

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

Trong đó ni là số mẫu trong nút i; n là số mẫu trong nút p.

Ví dụ: tập dữ liệu Sunburn

Name	**Hair**	**Height**	**Weight**	**Lotion**	**Result**
Sarah	Blonde	Average	Light	No	Burn
Dana	Blonde	Tall	Average	Yes	None
Alex	Brown	Short	Average	Yes	None
Annie	Blonde	Short	Average	No	Burn
Emily	Red	Average	Heavy	No	Burn
Pete	Brown	Tall	Heavy	No	None
John	Brown	Average	Heavy	No	None
Katie	Blonde	Short	Light	Yes	None

Name	Hair	Height	Weight	Lotion	Result
Sarah	Blonde	Average	Light	No	Burn
Dana	Blonde	Tall	Average	Yes	None
Alex	Brown	Short	Average	Yes	None
Annie	Blonde	Short	Average	No	Burn
Emily	Red	Average	Heavy	No	Burn
Pete	Brown	Tall	Heavy	No	None
John	Brown	Average	Heavy	No	None
Katie	Blonde	Short	Light	Yes	None

Entropy:
$H(S)=-\sum_i p(c_i) \log p(c_i)$

?

$$InfoGain(S,A) = H(S) - H(S \mid A)$$

$$= H(S) - \sum_a p(A=a) H(S \mid A=a)$$

$$= H(S) - \sum_{a \in Values(A)} \frac{|S_a|}{|S|} H(S_a)$$

- Bài tập: sử dụng độ đo Information Gain để chọn thuộc tính tốt nhất để
  1. Dựng cây quyết định cho Dữ liệu ở Slide 4
  2. Dựng cây quyết định cho Dữ liệu Sunburn

Nộp lên folder của mình trên Google drive.

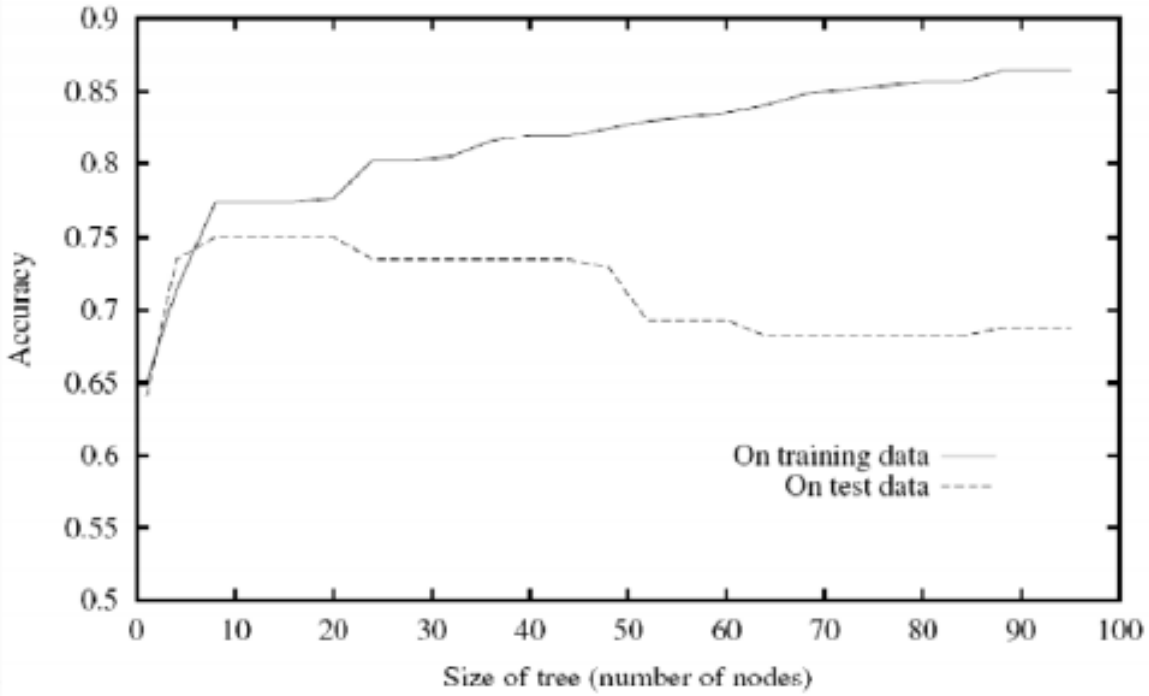
Một số vấn đề

- Cây quyết định học được quá phù hợp (over-fit) với các mẫu dữ liệu huấn luyện
- Xử lý các thuộc tính có kiểu giá trị liên tục (kiểu số thực)
- Các đánh giá phù hợp hơn (tốt hơn Information Gain) đối với việc xác định thuộc tính kiểm tra cho một nút
- Xử lý các ví dụ học thiếu giá trị thuộc tính (missing-value attributes)
- Xử lý các thuộc tính có chi phí (cost) khác nhau

→ Cải tiến của giải thuật ID3 với tất cả các vấn đề nêu trên được giải quyết: giải thuật C4.5

Học quá (over-fitting)

Tiếp tục quá trình học cây quyết định sẽ làm giảm độ chính xác đối với tập thử nghiệm mặc dù tăng độ chính xác đối với tập học



[*Mitchell, 1997*]

Tránh over-fitting

- 2 chiến lược
  - Ngừng việc học (phát triển) cây quyết định sớm hơn, trước khi nó đạt tới cấu trúc cây cho phép phân loại (khớp) hoàn hảo tập huấn luyện
  - Học (phát triển) cây đầy đủ (tương ứng với cấu trúc cây hoàn toàn phù hợp đối với tập huấn luyện), và sau đó thực hiện quá trình tỉa (to post-prune) cây „
- Chiến lược tỉa cây đầy đủ (Post-pruning over-fit trees) thường cho hiệu quả tốt hơn trong thực tế

  Lý do: Chiến lược "ngừng sớm" việc học cây cần phải đánh giá chính xác được khi nào nên ngừng việc học (phát triển) cây – Khó xác định!

Các thuộc tính có giá trị liên tục

- Cần xác định (chuyển đổi thành) các thuộc tính có giá trị rời rạc, bằng cách chia khoảng giá trị liên tục thành một tập các khoảng (intervals) không giao nhau.
- Đối với thuộc tính (có giá trị liên tục) A, tạo một thuộc tính mới kiểu nhị phân Av sao cho: Av là đúng nếu A>v, và là sai nếu ngược lại
- Xác định giá trị ngưỡng v "tốt nhất"?
  → Chọn giá trị ngưỡng v nhằm sinh ra giá trị *Information Gain* cao nhất.

- Vấn đề với Information gain:
  - Có thiên hướng lựa chọn các thuộc tính có nhiều giá trị (attributes with many values).

Một đánh giá khác: Gain Ratio

→Giảm ảnh hưởng của các thuộc tính có (rất) nhiều giá trị

$$GainRatio(S,A) = \frac{Gain(S,A)}{SplitInformation(S,A)}$$

$$SplitInformation(S,A) = -\sum_{v \in Values(A)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$$

(trong đó `Values(A)` là tập các giá trị có thể của thuộc tính `A`, và $S_v = \{x \mid x \in S, x_A = v\}$)

Xử lý các thuộc tính thiếu giá trị

- Giả sử thuộc tính A là một thuộc tính kiểm tra ở nút n
- Nếu mẫu x không có (thiếu) giá trị đối với thuộc tính A (x_A không xác định)?
- Gọi Sn là tập các mẫu học gắn với nút n có giá trị đối với thuộc tính A
  - Giải pháp 1: x_A là giá trị phổ biến nhất đối với thuộc tính A trong số các ví dụ thuộc tập Sn
  - Giải pháp 2: x_A là giá trị phổ biến nhất đối với thuộc tính A trong số các ví dụ thuộc tập Sn có cùng phân lớp với x.

Đánh giá hiệu năng của thuật toán

- Độ chính xác (Accuracy):
  - on validation data
  - K-fold cross validation

- Chi phí do phân lớp sai (Misclassification cost): Sometimes more accuracy is desired for some classes than others.

- Minimum description length (MDL):
  - Favor good accuracy on compact model
  - MDL = model_size(tree) + errors(tree)

Regression Tree (cây hồi quy, sau)

- A variant of decision trees for dealing with continuous target attribute
- Estimation problem: approximate real-valued functions: e.g., weight, income, scoring
- A leaf node is marked with a real value or a linear function: e.g., the mean of the target values of the examples at the node.
- Measure of impurity: e.g., variance, standard deviation,…

Dùng cây quyết định khi nào?

- Các mẫu học được biểu diễn bằng các cặp (thuộc tính, giá trị)
  - Phù hợp với các thuộc tính có giá trị rời rạc
  - Đối với các thuộc tính có giá trị liên tục, phải rời rạc hóa „
- Hàm mục tiêu có giá trị đầu ra là các giá trị rời rạc
  - Ví dụ: Phân loại các ví dụ vào lớp phù hợp „
- Phù hợp khi hàm mục tiêu được biểu diễn ở dạng tách rời (disjunctive)
- Tập huấn luyện có thể chứa nhiễu/lỗi
  - Lỗi trong phân loại (nhãn lớp) của các mẫu dữ liệu
  - Lỗi trong giá trị thuộc tính biểu diễn của mẫu dữ liệu „
- Tập huấn luyện có thể chứa các thuộc tính thiếu giá trị (giá trị của một số thuộc tính là không xác định đối với một số phần tử dữ liệu học).

Bài tập và thực hành

RID	age	income	student	credit-rating	class:buy_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle-aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle-aged	medium	no	excellent	yes
13	middle-aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

- Huấn luyện (dựng cây quyết định) từ D
- Đánh giá hiệu năng của cây trên D và trên tập test D1 như sau:

RID	Age	Income	Student	C-Rate	Buy
1	Youth	Low	Yes	Fair	?
2	Senior	Medium	No	Exellent	?

Thảo luận