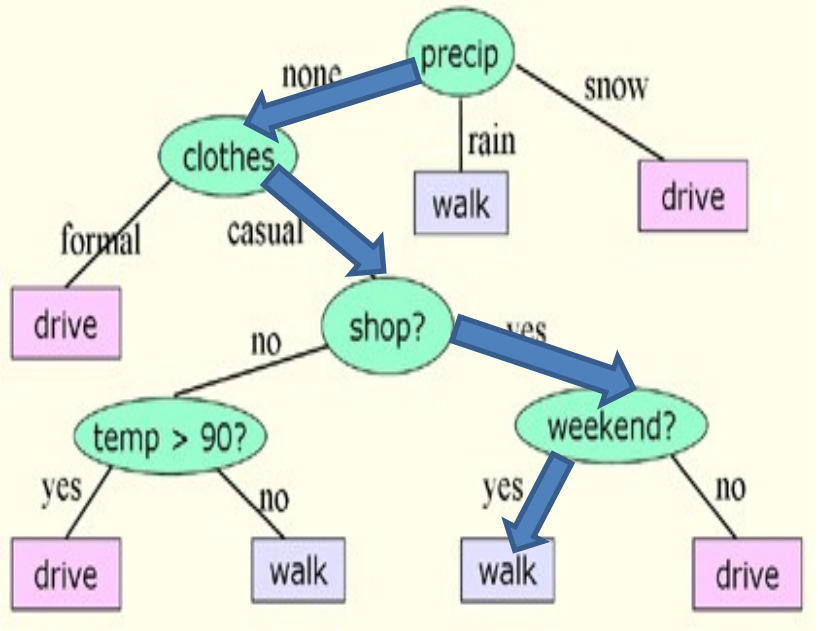
Cây Quyết Định

Cây quyết định

➢ Định nghĩa: cấu trúc phân cấp của các nút và nhánh

- Nút gốc và nút nội bộ: tên các thuộc tính của tập dữ liệu
- Nút lá: tên các c_j
- Nhánh: tương ứng với các giá trị của các thuộc tính



➢ Phân lớp các mẫu bằng cách duyệt cây từ gốc tới các nút lá

- Ví dụ:

  (precip=none, clothes=casual, shop=yes, weekend=yes) -> walk

Xây dựng cây quyết định

➢ Một số thuật toán

- ID3, C4.5 (Information Gain/ Information Gain Ratio)
- CART (Gini Index)
- SLIQ, SPRINT (Gini Index)

➢ Ý tưởng chính

- Xây dựng cây từ trên xuống (top-down) bằng cách xác định thuộc tính cho kết quả phân lớp tốt nhất
  - Thuộc tính cho ra cây đơn giản nhất (Định luật Occam)
  - Heuristic: thuộc tính tạo ra nút có tính đồng nhất cao nhất (purest)
  - Sử dụng các độ đo sự đồng nhất: Information Gain (Entropy), Information Gain Ratio, Gini Index
- Điều kiện dừng:
  - Không còn thuộc tính nào để phân lớp nữa
  - Tất cả các mẫu đều đã được phân lớp

Độ lợi thông tin - Information Gain

- Cho:
  - D: là tập huấn luyện
  - C_i: tập các lớp xác định trước, i = {1, …, m}
  - $C_{i, D}$: tập các mẫu của D thuộc lớp Ci
- Xác suất để một mẫu bất kỳ trong D thuộc về lớp C_i là:

$$p_i = \frac{|C_{i,D}|}{|D|}$$

- Thông tin kỳ vọng (entropy) để phân lớp một mẫu trong D/ độ hỗn loạn của D là:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Ví dụ: Tính Entropy

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- $|D| = 14$
- $m = 2$
- C_1 = yes
- C_2 = no
- $|C_{1,D}| = 9$
- $|C_{2,D}| = 5$

➢ Độ hỗn loạn của D là:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940$$

Độ lợi thông tin - Information Gain

➢ Độ hỗn loạn của D sau khi phân lớp theo thuộc tính A (thông tin cần thiết để chia tập D thành v tập con $\{D_1, D_2, \ldots, D_v\}$ ứng với tập giá trị $\{a_1, a_2, \ldots, a_v\}$ của thuộc tính A) là:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} I(D_j)$$

➢ Độ lợi thông tin để phân lớp tập D theo thuộc tính A là:

$$Gain(A) = Info(D) - Info_A(D)$$

Trước khi phân lớp

Sau khi phân lớp theo A

Ví dụ: Tính Information Gain

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- Lớp P = "Yes"
- Lớp N = "No"
- Info(D) = I(9, 5) = 0.94

age	p_j	n_j	$I(p_j, n_j)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

$$I(2,3) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$$

$$I(4,0) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0$$

$$I(3,2) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$$

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

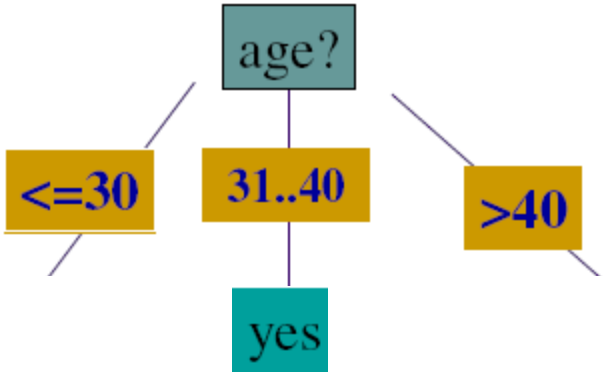
$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

I([9,5])

I([2,3], [4,0], [3,2])

Ví dụ - Xây dựng cây quyết định

- Gain (Age) = 0.246
- Gain (Incomce) = 0.029
- Gain (Student) = 0.151
- Gain (Credit-rating) = 0.048

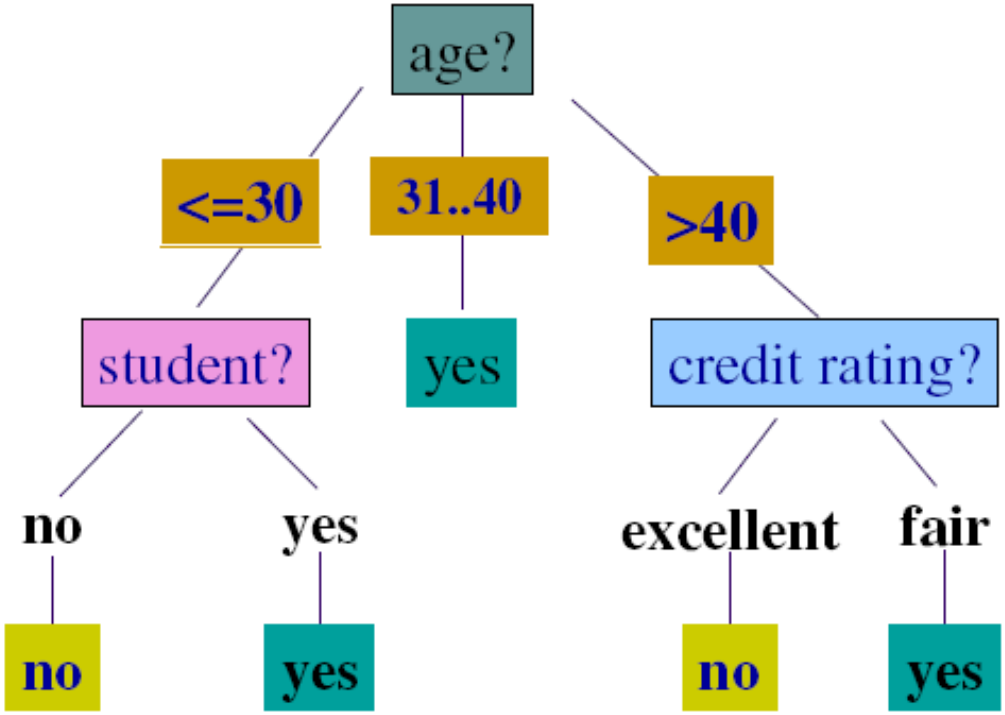


- Chọn thuộc tính Age làm nút gốc
- Tiếp tục tính Information Gain và lựa chọn thuộc tính để phân lớp ở 2 nhánh tiếp theo
  - Chỉ sử dụng các mẫu có liên quan đến nút

<=30

income	student	credit_rating	buys_computer
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

Ví dụ: Xây dựng cây quyết định



Information Gain Ratio

- Độ đo Gain có xu hướng thiên vị cho các thuộc tính có nhiều giá trị (tạo ra nhiều nhánh)
  - Ví dụ: Tập D có thuộc tính X gồm 14 giá trị đôi một khác nhau

    $Info_X(D) = 1/14 * I(1,0) * 14 = 0 \Rightarrow Gain(X) = Info(D) - 0 = 0.94$

  - Cần chuẩn hóa độ đo Gain
- Độ đo Gain Ratio quan tâm đến số lượng và độ lớn của các nhánh khi lựa chọn thuộc tính phân lớp
- $GainRatio(A) = Gain(A) / SplitInfo_A(D)$

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

Ví dụ: Information Gain Ratio

- Gain(Age) = 0.246
- $SplitInfo_{Age}(D) = I(5,4,5) = 1.577$
- GainRatio(Age) = 0.156

- Gain(Income) = 0.029
- $SplitInfo_{Income}(D) = I(4,4,6) = 1.557$
- GainRatio(Income) = 0.019

- Gain(Student) = 0.151
- $SplitInfo_{Student}(D) = I(7,7) = 1$
- GainRatio(Student) = 0.151

- Gain(CreditRating) = 0.048
- $SplitInfo_{CreditRating}(D) = I(8,6) = 0.985$
- GainRatio(CreditRating) = 0.049

- Gain(X) = 0.94
- $SplitInfo_{X}(D) = I(1,\ldots,1) = 3.807$
- GainRatio(X) = 0.247

➢ Độ đo Gain Ratio chuẩn hóa độ đo Gain nhưng có thể xảy ra trường hợp chọn một thuộc tính chỉ vì SplitInfo của nó rất thấp. Giải pháp:

- Chọn thuộc tính có Gain lớn hơn một giá trị Gain trung bình
- So sánh các thuộc tính dựa trên Gain Ratio

Chỉ mục Gini – Gini index

- Cho tập huấn luyện D chứa các mẫu thuộc m lớp
- Chỉ mục Gini của tập D là:

$$gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

  - p_i là xác suất một mẫu trong D thuộc về lớp C_i

- Ví dụ: Chỉ mục Gini của tập D

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

Chỉ mục Gini – Gini index

- Chỉ mục gini để phân chia tập D thành v tập con {D_1, D_2, …, D_v} theo thuộc tính A {a_1, a_2, …, a_v}

$$gini_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} gini(D_j)$$

- Chọn thuộc tính có chỉ mục Gini nhỏ nhất để phân chia tập dữ liệu

Ví dụ: Gini index

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

age	p_j	n_j	gini(p_j, n_j)
<=30	2	3	0.48
31…40	4	0	0
>40	3	2	0.48

- Chỉ mục gini cho thuộc tính Age:

$$gini_{age}(D) = \frac{5}{14} gini(2,3) + \frac{4}{14} gini(4,0) + \frac{5}{14} gini(3,2)$$
$$= 0.343$$

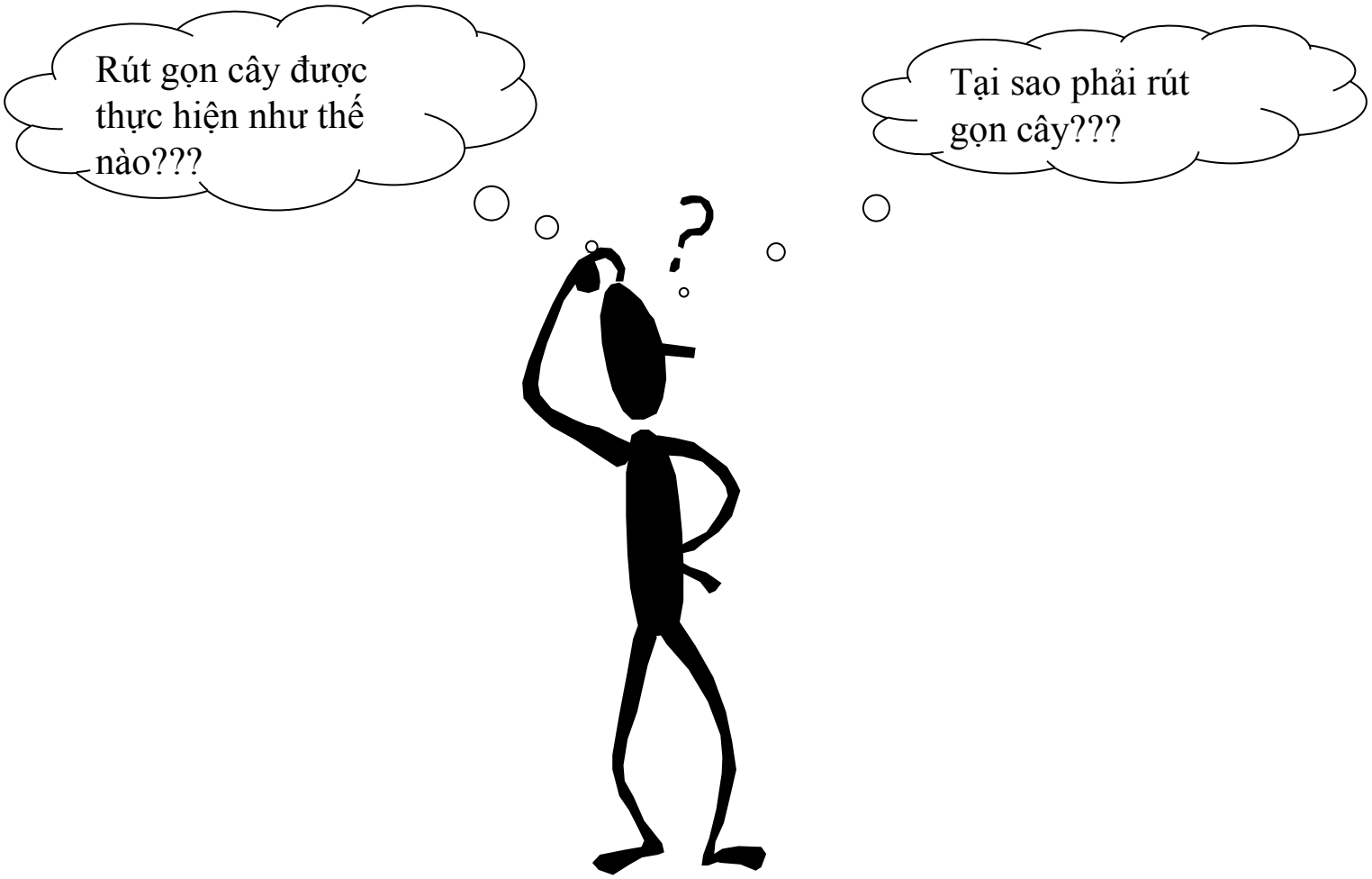
Ví dụ - Gini index

- $Gini_{Age}$ (D) = 0.343
- $Gini_{Income}$ (D) = 0.44
- $Gini_{Student}$ (D) = 0.367
- $Gini_{Credit\text{-}rating}$ (D) = 0.429
- Chọn thuộc tính Age làm nút gốc
- Tiếp tục tính Gini index và lựa chọn thuộc tính để phân lớp ở 2 nhánh tiếp theo
  - Chỉ sử dụng các mẫu có liên quan đến nút

Nhận xét 3 độ đo

- **Information Gain**
  - Có xu hướng thiên vị cho các thuộc tính có nhiều giá trị
- **Information Gain Ratio**
  - Có xu hướng chọn thuộc tính phân lớp tạo ra một cây con nhỏ hơn nhiều so với các cây con khác
- **Gini index**
  - Có xu hướng thiên vị cho các thuộc tính có nhiều giá trị
  - Gặp khó khăn khi số lượng lớp lớn
  - Có xu hướng chọn thuộc tính phân lớp tạo ra các cây con kích thước và độ hỗn loạn bằng nhau

RÚT GỌN CÂY



Rút gọn cây (tỉa cây):

- Xác định và loại bỏ các nhánh không ổn định hoặc cá biệt.
- Cây được rút gọn có xu hướng nhỏ lại và ít phức tạp hơn nên dễ hiểu hơn.
- Cây được rút gọn này phân lớp nhanh hơn, tốt hơn.

Các hướng tiếp cận để rút gọn cây:

- Có 3 hướng tiếp cận thông thường để rút gọn cây:
  - Rút gọn trước (Pre-pruning).
  - Rút gọn sau(Post-pruning).
  - Kết hợp.

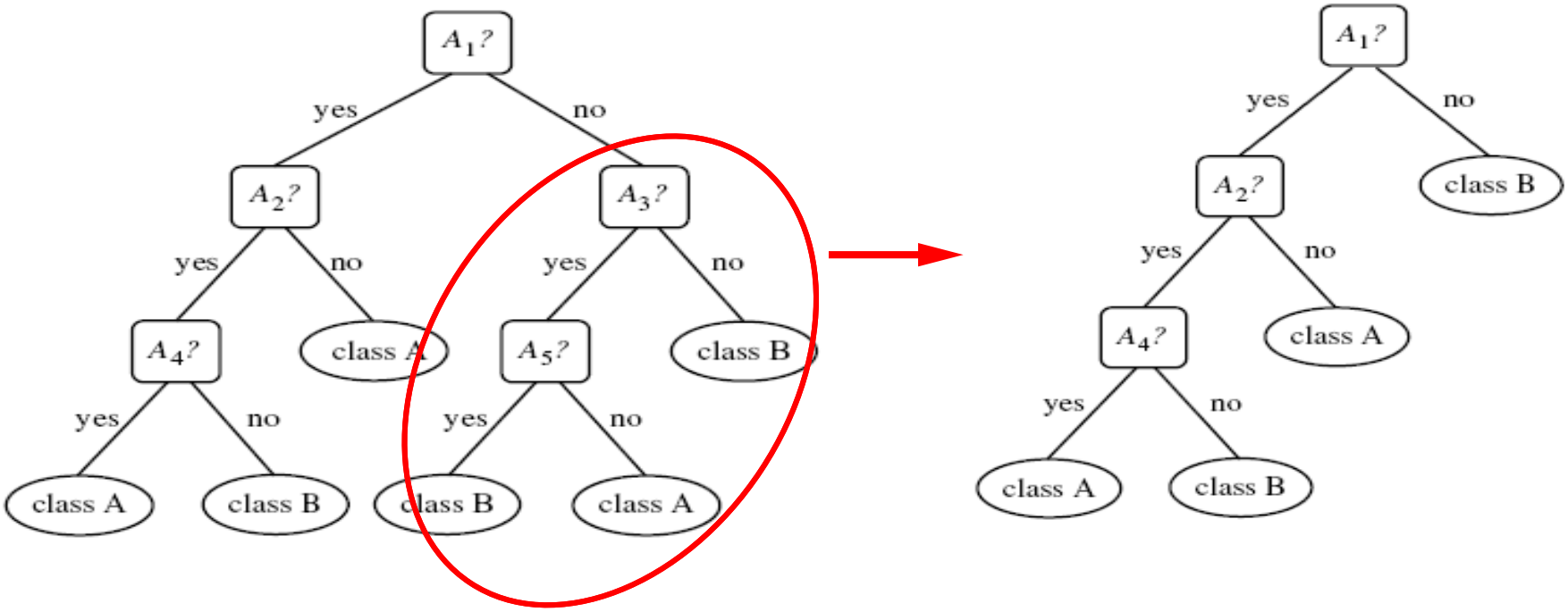
RÚT GỌN TRƯỚC

- Dừng giải thuật trước khi nó trở thành cây sinh trưởng đầy đủ
- Các điều kiện thoát thông thường của một nút:
  - Dừng nếu tất cả các trường hợp (mẫu) thuộc cùng lớp
  - Dừng nếu không còn thuộc tính nào để phân chia
- Các điều kiện nghiêm ngặt hơn:
  - Khi phân hoạch một cây, nếu phân hoạch tại một nút trả về kết quả rằng phép phân chia này vượt quá một ngưỡng cho phép thì sự phân hoạch tạm dừng->Khó khăn trong việc chọn ngưỡng.
  - Dừng nếu việc mở rộng nút hiện hành không cải thiện các độ đo (chẳng hạn, Gini hoặc information gain…).

RÚT GỌN SAU

- Ý tưởng chính của rút gọn sau là bỏ đi một số cây con từ một cây đã xây dựng hoàn chỉnh.
- Cây con được bỏ đi bằng cách thay thế các nhánh của nó bằng nút lá. Nút lá này sẽ có nhãn lớp là lớp có tần số xuất hiện cao nhất trong cây con được thay thế.

RÚT GỌN SAU



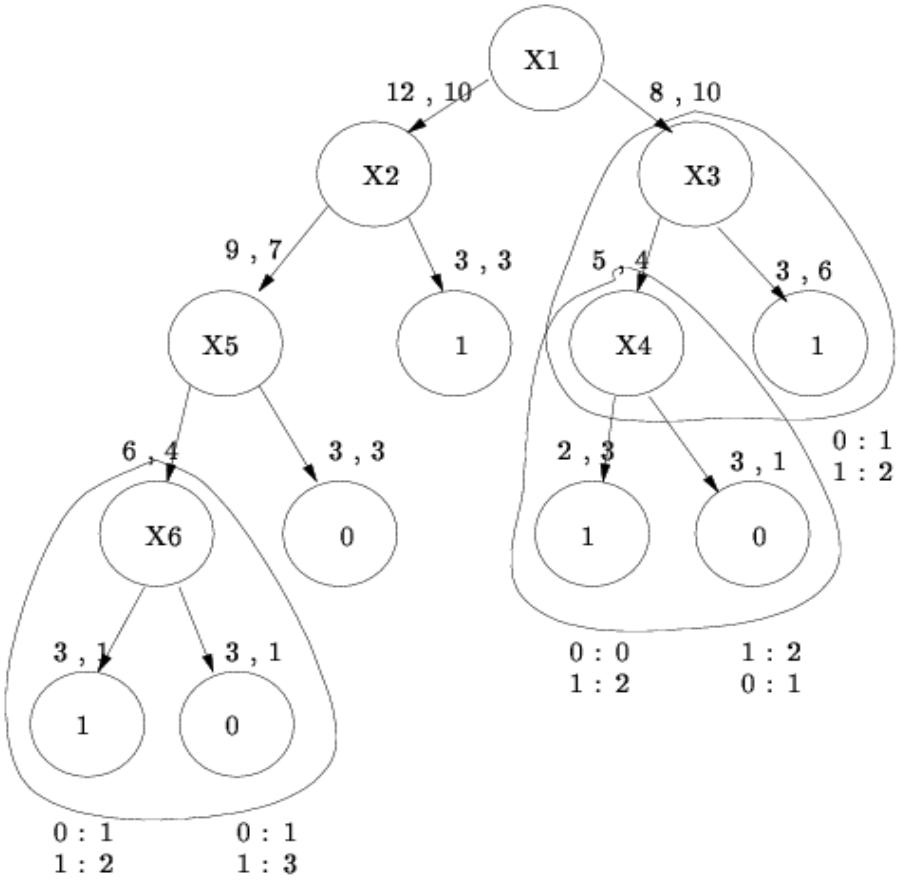
Hình 1: Một cây quyết định cần được rút gọn và kết quả rút gọn

Ví dụ trong hình 1: Nút "A3?" của cây chưa được rút gọn. Lớp xuất hiện nhiều nhất của cây con là nút "classB", khi đó cây rút gọn sẽ có nhánh này được thay thế bởi nút "classB".

RÚT GỌN SAU

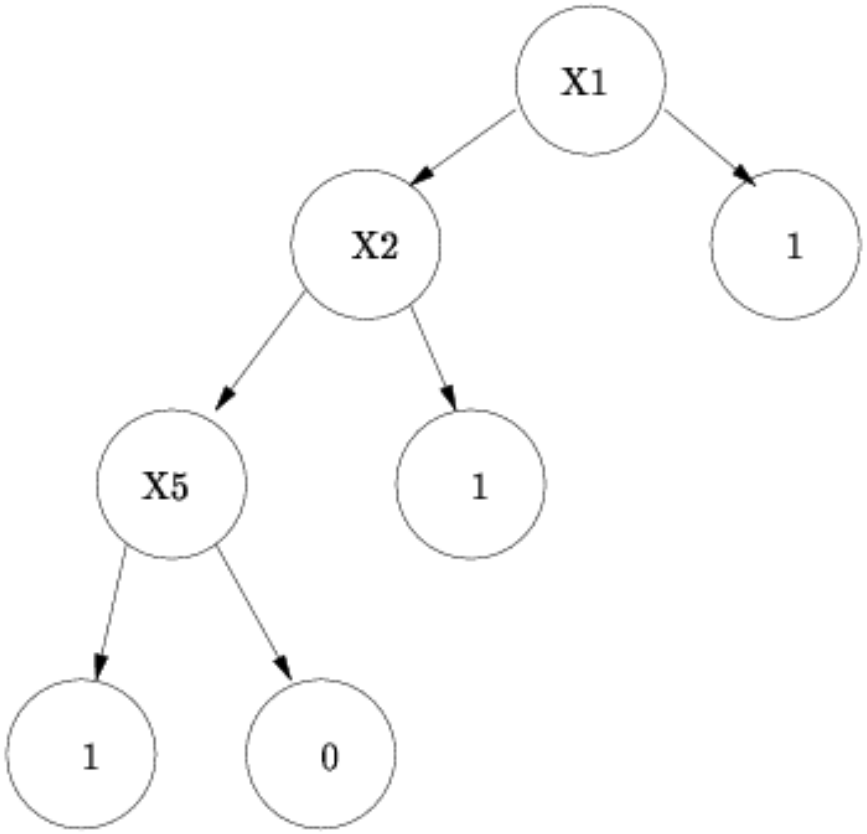
- Rút gọn sau được áp dụng trong CART là một ví dụ điển hình. Trong thuật toán này, độ phức tạp chi phí của cây được sử dụng để rút gọn cây.
- Ý tưởng của phương pháp này là xem độ phức tạp chi phí của cây là một hàm số theo số lượng lá và tần suất lỗi của cây (tần suất lỗi là tỉ lệ phần trăm của các bộ dữ liệu có nhãn lớp không chính xác trong cây).
- Thuật toán bắt đầu từ đáy của cây. Cứ mỗi nút trong – N của cây sẽ tính ra 2 giá trị độ phức tạp.
  - 1. Độ phức tạp chi phí của cây con tại N.
  - 2. Độ phức tạp chi phí tại nút N sau khi rút gọn.
  - Sau đó so sánh 2 giá trị này, nếu giá trị sau rút gọn nhỏ hơn thì cây sẽ được rút gọn, ngược lại thì giữ nguyên.
- Ngoài ra có thể áp dụng nguyên lý MDL (Minimum Description Length) tức là rút gọn cây dựa trên số lượng bit dùng để mã hóa thay vì đo lường tần suất lỗi.

Ví dụ



1. Cây T cần được rút gọn.
2. Số ghi trên cạnh gồm 2 số: Số bên trái là số lượng mẫu đếm được trong tập train, bên phải là số lượng mẫu đếm được trong tập test.
3. X_i là các thuộc tính kiểm tra khi xây dựng cây, nó là các node trong của cây.
4. Các giá trị 0, 1 trong node lá là các nhãn lớp (class label).

Ví dụ: kết quả rút gọn



HƯỚNG TIẾP CẬN KẾT HỢP

- Tiếp cận rút gọn trước (Pre-prunung) và tiếp cận rút gọn sau (Post-pruning) có thể sử dụng xen kẽ kết hợp để tạo ra cây tốt hơn.
- Ngoài ra có thể kết hợp nhiều thuộc tính và phép phân chia đa trị để ngăn chặn việc trùng lắp xuất hiện trong cùng một cây.