Khoa Công Nghệ Thông Tin
Trường Đại Học Cần Thơ

Phương pháp học cây quyết định
Decision Tree

Đỗ Thanh Nghị
dtnghi@cit.ctu.edu.vn

Cần Thơ
02-12-2008

Nội dung

- Giới thiệu về cây quyết định
- Giải thuật học của cây quyết định
- Kết luận và hướng phát triển

Nội dung

- **Giới thiệu về cây quyết định**
- Giải thuật học của cây quyết định
- Kết luận và hướng phát triển

Cây quyết định

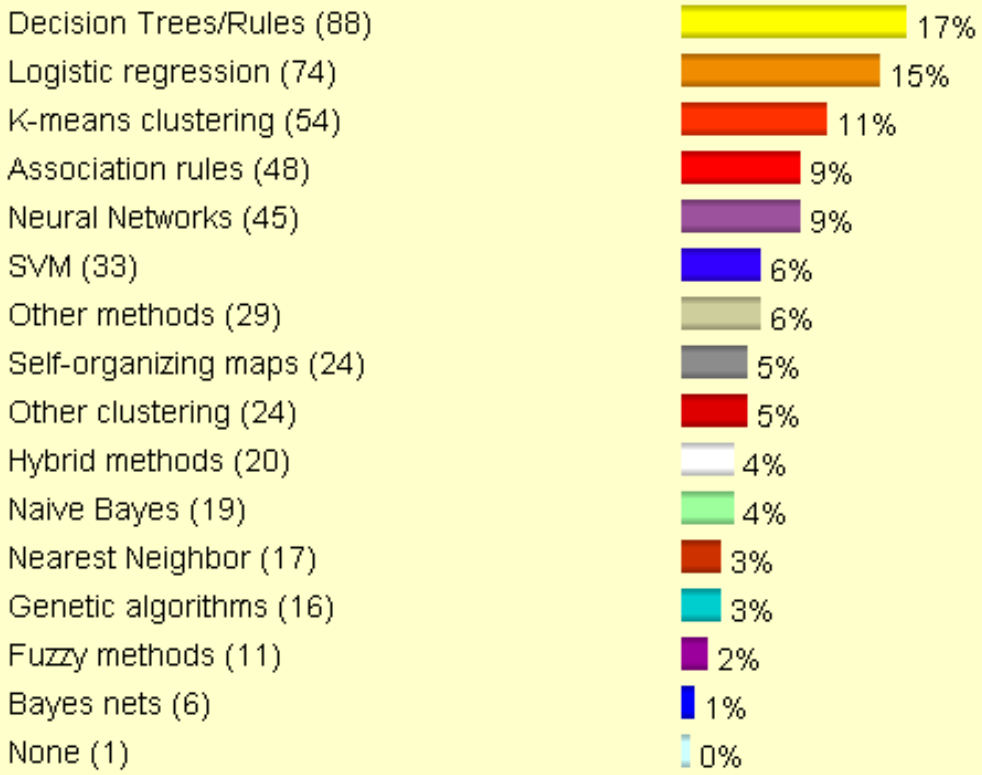
- lớp các giải thuật học
  - kết quả sinh ra dễ dịch (if … then …)
  - khá đơn giản, nhanh, hiệu quả được sử dụng nhiều
  - liên tục trong nhiều năm qua, cây quyết định được bình chọn là giải thuật được sử dụng nhiều nhất và thành công nhất
  - giải quyết các vấn đề của phân loại, hồi quy
  - làm việc cho dữ liệu số và loại
  - được ứng dụng thành công trong hầu hết các lãnh vực về phân tích dữ liệu, phân loại text, spam, phân loại gien, etc
  - có rất nhiều giải thuật sẵn dùng : C4.5 (Quinlan, 1993), CART (Breiman et al., 1984), etc

Kỹ thuật DM thành công trong ứng dụng thực (2004)

KDnuggets : **Polls** : Deployed data mining techniques

Poll

Which data mining techniques you used in a successfully deployed application? [173 voters, 509 votes total]



Nội dung

- Giới thiệu về cây quyết định
- **Giải thuật học của cây quyết định**
- Kết luận và hướng phát triển

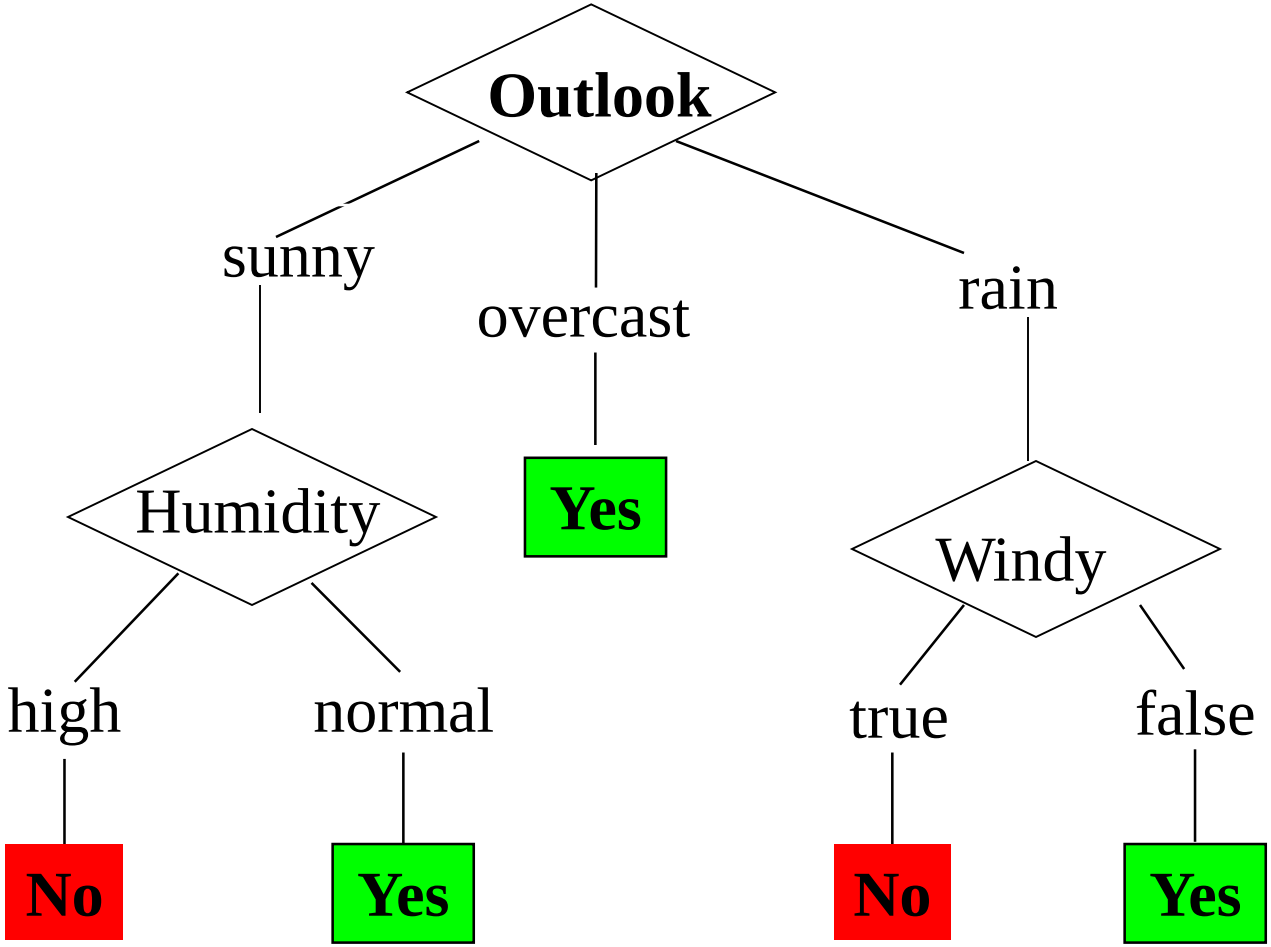
Giải thuật học cây quyết định

- 1 nút trong : test trên 1 thuộc tính (biến)
- 1 nhánh : trình bày cho dữ liệu thỏa mãn test, ví dụ : age < 25.
- nút lá : lớp (nhãn)
- ở mỗi nút, 1 thuộc tính được chọn để phân hoạch dữ liệu học sao cho tách rời các lớp tốt nhất có thể
- dữ liệu mới đến được phân loại theo đường dẫn từ gốc đến nút lá

Dữ liệu weather, dựa trên các thuộc tính (Outlook, Temp, Humidity, Windy), quyết định (play/no)

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Cây quyết định cho tập dữ liệu weather, dựa trên các thuộc tính (Outlook, Temp, Humidity, Windy)



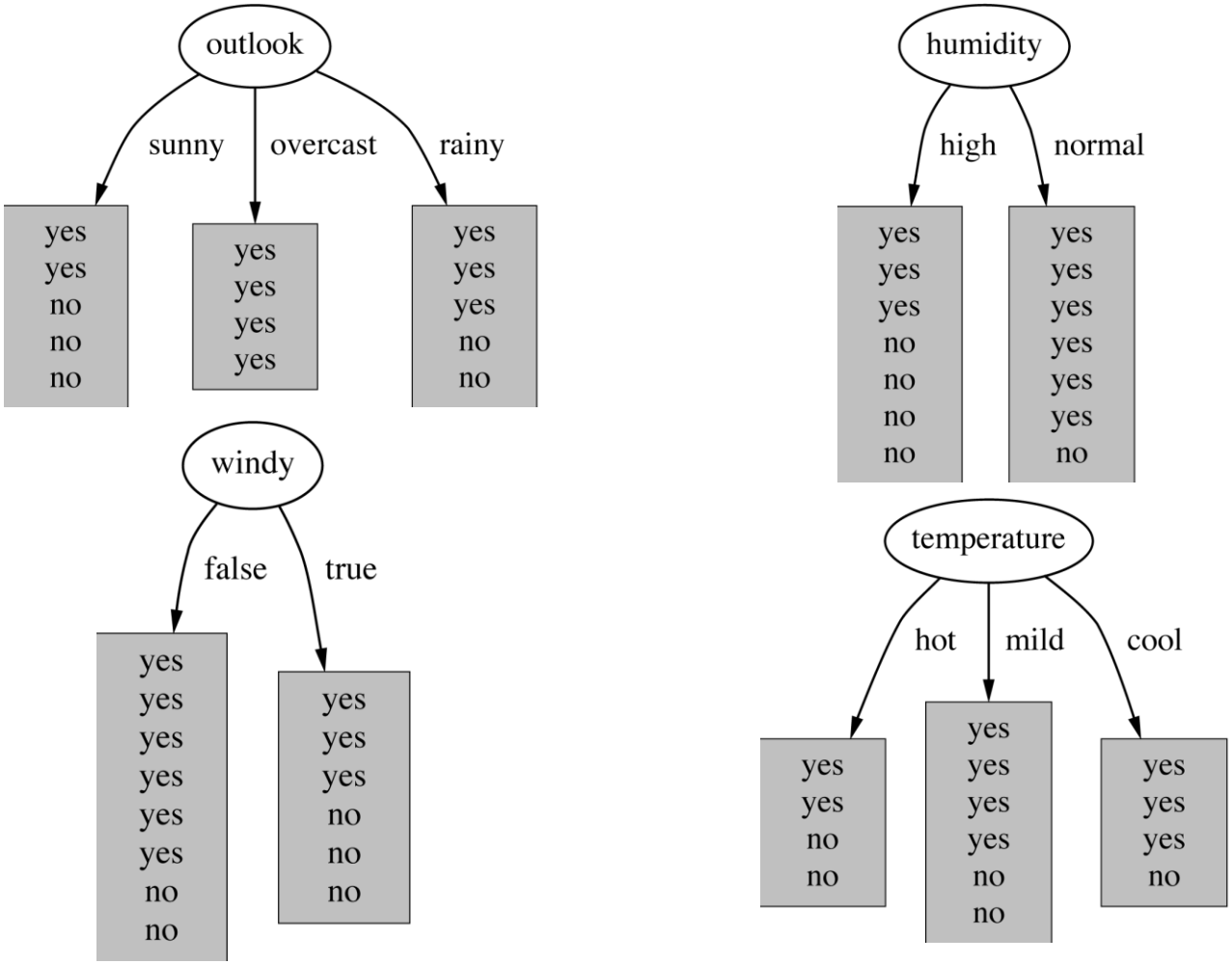
Giải thuật cây quyết định

- xây dựng cây Top-down
  - bắt đầu nút gốc, tất cả các dữ liệu học ở nút gốc
  - phân hoạch dữ liệu một cách đệ quy bằng việc chọn 1 thuộc tính để thực hiện phân hoạch tốt nhất có thể
- cắt nhánh Bottom-up
  - cắt những cây con hoặc các nhánh từ dưới lên trên, để tránh học vẹt (overfitting, over learning)

Chọn thuộc tính phân hoạch

- ở mỗi nút, các thuộc tính được đánh giá dựa trên phân tách dữ liệu học tốt nhất có thể
- việc đánh giá dựa trên
  - độ lợi thông tin, information gain (ID3/C4.5)
  - information gain ratio
  - chỉ số gini, gini index (CART)

Chọn thuộc tính phân hoạch ?



Chọn thuộc tính phân hoạch ?

- thuộc tính nào tốt ?
  - cho ra kết quả là cây nhỏ nhất
  - heuristics: chọn thuộc tính sinh ra các nút “purest” (thuần khiết)
- độ lợi thông tin
  - tăng với giá trị trung bình thuần khiết của các tập con của dữ liệu mà thuộc tính sinh ra
- chọn thuộc tính có độ lợi thông tin lớn nhất

Độ lợi thông tin

- thông tin được đo lường bằng *bits*
  - cho 1 phân phối xác suất, thông tin cần thiết để dự đoán 1 sự kiện là *entropy* ☺
- công thức tính entropy:

$$\text{entropy}(p_1, p_2, \ldots, p_n) = -p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n$$

*Claude Shannon

Born: 30 April 1916

Died: 23 February 2001

"Father of information theory"

Ví dụ : thuộc tính outlook

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

Ví dụ : thuộc tính outlook

- "Outlook" = "Sunny":

$$\text{info}([2,3]) = \text{entropy}(2/5,3/5) = -2/5\log(2/5) - 3/5\log(3/5) = 0.971\,\text{bits}$$

- "Outlook" = "Overcast":

$$\text{info}([4,0]) = \text{entropy}(1,0) = -1\log(1) - 0\log(0) = 0\,\text{bits}$$

chú ý : log(0) không xác định nhưng 0*log(0) là 0

- "Outlook" = "Rainy":

$$\text{info}([3,2]) = \text{entropy}(3/5,2/5) = -3/5\log(3/5) - 2/5\log(2/5) = 0.971\,\text{bits}$$

- thông tin của thuộc tính outlook:

$$\text{info}([3,2],[4,0],[3,2]) = (5/14)\times 0.971 + (4/14)\times 0 + (5/14)\times 0.971$$
$$= 0.693\,\text{bits}$$

Độ lợi thông tin

- độ lợi thông tin của outlook

(trước khi phân hoạch) – (sau khi phân hoạch)

$$gain(\text{" Outlook"}) = info([9,5]) - info([2,3],[4,0],[3,2]) = 0.940 - 0.693$$
$$= 0.247 \text{ bits}$$

Thuộc tính humidity

- "Humidity" = "High":

$$info([3,4]) = entropy(3/7,4/7) = -3/7\log(3/7) - 4/7\log(4/7) = 0.985\,bits$$

- "Humidity" = "Normal":

$$info([6,1]) = entropy(6/7,1/7) = -6/7\log(6/7) - 1/7\log(1/7) = 0.592\,bits$$

- thông tin của thuộc tính humidity

$$info([3,4],[6,1]) = (7/14)\times 0.985 + (7/14)\times 0.592 = 0.788\,bits$$

- độ lợi thông tin của thuộc tính humidity

$$info([9,5]) - info([3,4],[6,1]) = 0.940 - 0.788 = 0.152$$

Độ lợi thông tin

- độ lợi thông tin của các thuộc tính

(trước khi phân hoạch) – (sau khi phân hoạch)

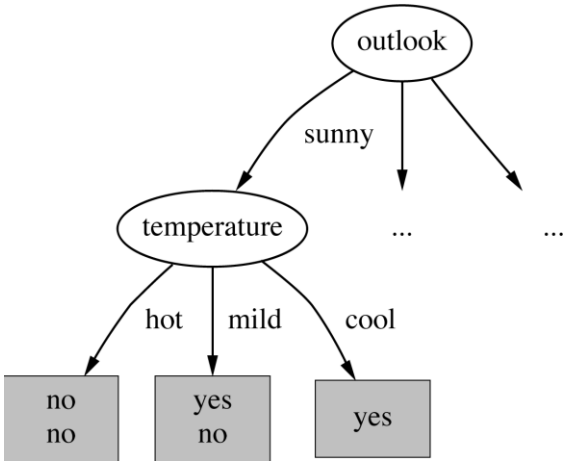
$$\text{gain}(\text{"Outlook"}) = 0.247 \text{ bits}$$

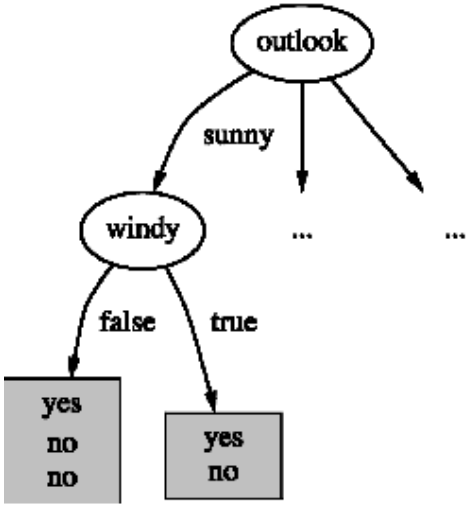
$$\text{gain}(\text{"Temperature"}) = 0.029 \text{ bits}$$

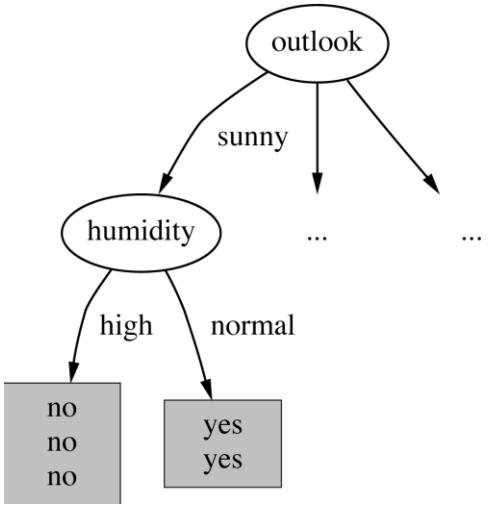
$$\text{gain}(\text{"Humidity"}) = 0.152 \text{ bits}$$

$$\text{gain}(\text{"Windy"}) = 0.048 \text{ bits}$$

Tiếp tục phân hoạch dữ liệu





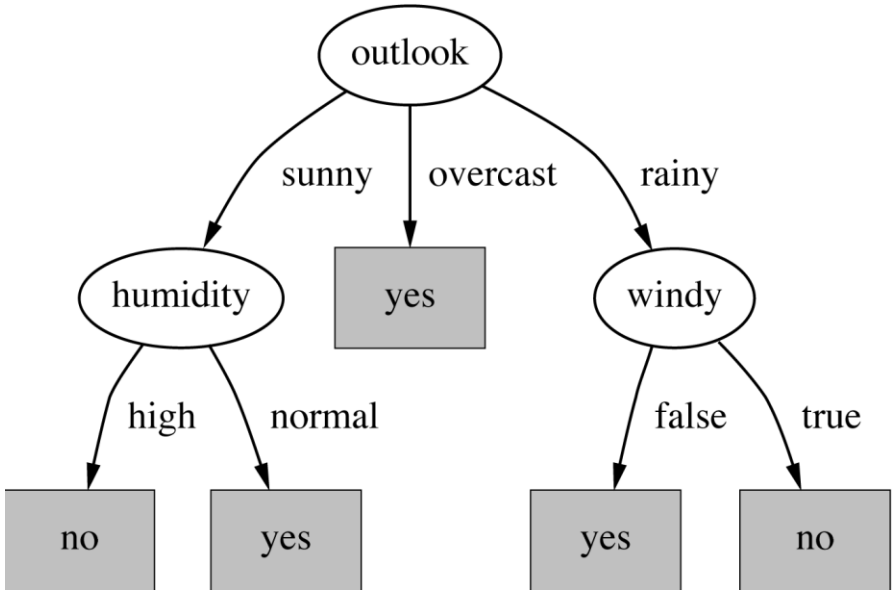


$$gain("Humidity") = 0.971\,bits$$

$$gain("Temperature") = 0.571\,bits$$

$$gain("Windy") = 0.020\,bits$$

Kết quả



- chú ý : có thể có nút lá không thuần khiết

  ⇒ phân hoạch dừng khi dữ liệu không thể phân hoạch, nhãn được gán cho lớp lớn nhất chứa trong nút lá

Chỉ số gini (CART)

- nếu dữ liệu T có n lớp, chỉ số gini(T) được định nghĩa như sau :

$$gini(T) = 1 - \sum_{j=1}^{n} p_j^2$$

p_j là xác suất của lớp j trong T

- gini(T) là nhỏ nhất nếu những lớp trong T bị lệch

Chỉ số gini (CART)

- sau khi phân hoạch T thành 2 tập con T1 & T2 với kích thước N1 & N2, chỉ số gini

$$gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

- thuộc tính có $gini_{split}(T)$ nhỏ nhất được chọn để phân hoạch

Giải thuật

- giải thuật ID3/C4.5 (Quinlan, 1993)
  - sử dụng Gain ratio
  - xử lý dữ liệu số, loại, nhiễu
- CART (Breiman et al., 1984)
  - sử dụng chỉ số Gini
  - xử lý dữ liệu số, loại, nhiễu

Giải thuật C4.5, dữ liệu kiểu số

- phân hoạch nhị phân
  - ví dụ : temp < 45
- không như dữ liệu loại, dữ liệu kiểu số có nhiều nhánh phân hoạch
- phương pháp
  - tính độ lợi thông tin cho mọi giá trị phân nhánh của thuộc tính
  - chọn giá trị phân nhánh tốt nhất

Tập Weather, dữ liệu kiểu số

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	86	False	Yes
Rainy	75	80	False	Yes
…	…	…	…	…

```
If outlook = sunny and humidity > 83 then play = no
If outlook = rainy and windy = true then play = no
If outlook = overcast then play = yes
If humidity < 85 then play = yes
If none of the above then play = yes
```

Tập Weather, dữ liệu kiểu số

- phân hoạch trên thuộc tính temperature

64	65	68	69	70	71	72	72	75	75	80	81	83	85
Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

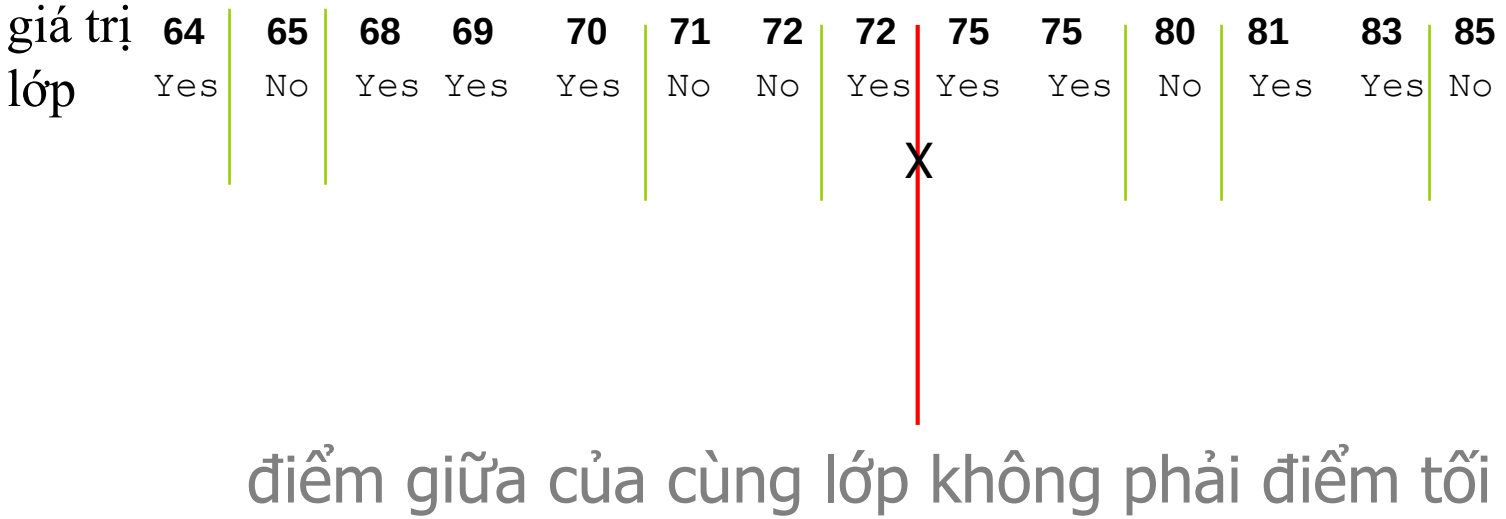
  - ví dụ temperature < 71.5: yes/4, no/2
    temperature ≥ 71.5: yes/5, no/3

  - Info([4,2],[5,3]) = 6/14 info([4,2]) + 8/14 info([5,3])
    = 0.939 bits

- điểm phân hoạch : giữa
- có thể tính tất cả với 1 lần pass!
- cần sắp xếp dữ liệu

Cải tiến

- chỉ cần tính entropy tại các điểm thay đổi lớp (Fayyad & Irani, 1992)



điểm giữa của cùng lớp không phải điểm tối ưu

Cắt nhánh

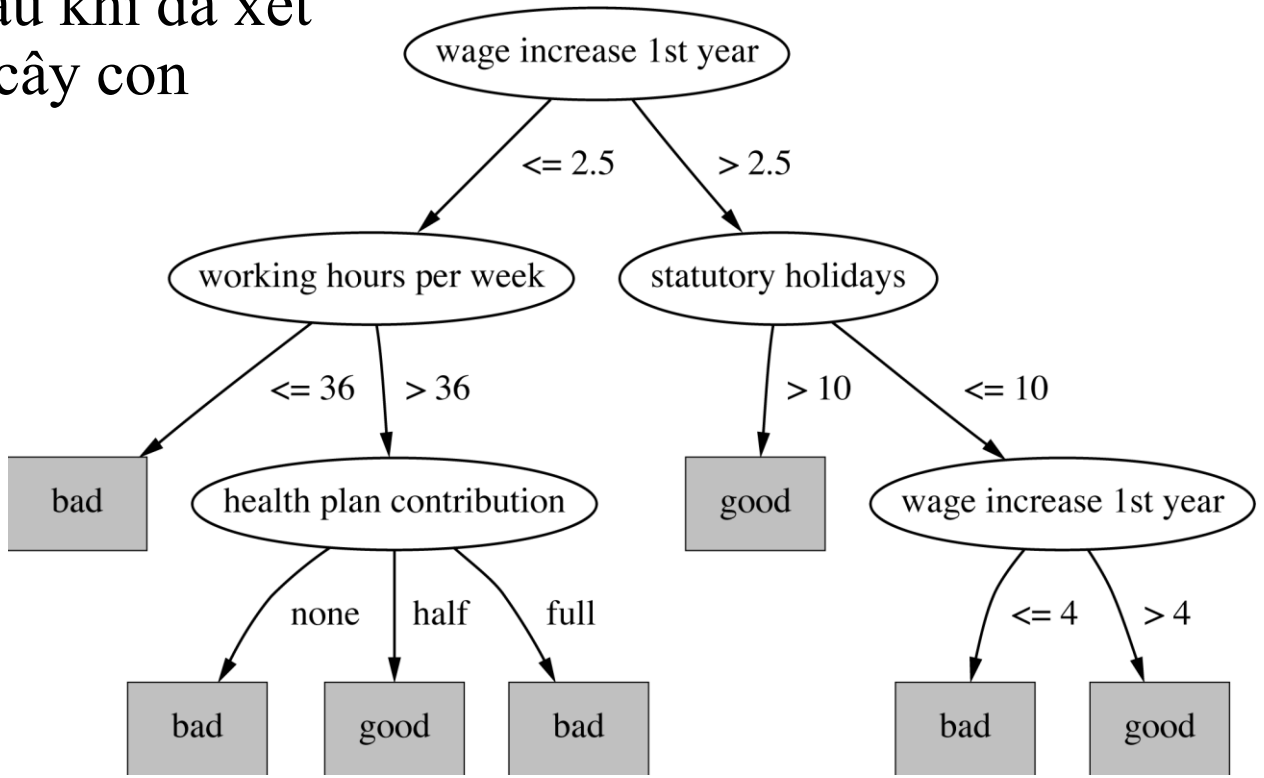
- mục tiêu : tránh học vẹt (overfitting), chịu đựng nhiễu, tăng độ chính xác khi phân loại tập test
- có 2 pha
  - *postpruning* – cắt nhánh cây sao cho tăng khả năng phân loại của cây
  - *prepruning* – dừng sớm quá trình phân nhánh
- trong thực tế, postpruning được sử dụng nhiều hơn prepruning

Postpruning

- xây dựng cây đầy đủ
- cắt nhánh
  - *thay thế cây con*
  - *đưa cây con lên trên*
- có nhiều chiến lược
  - ước lượng lỗi
  - significance test

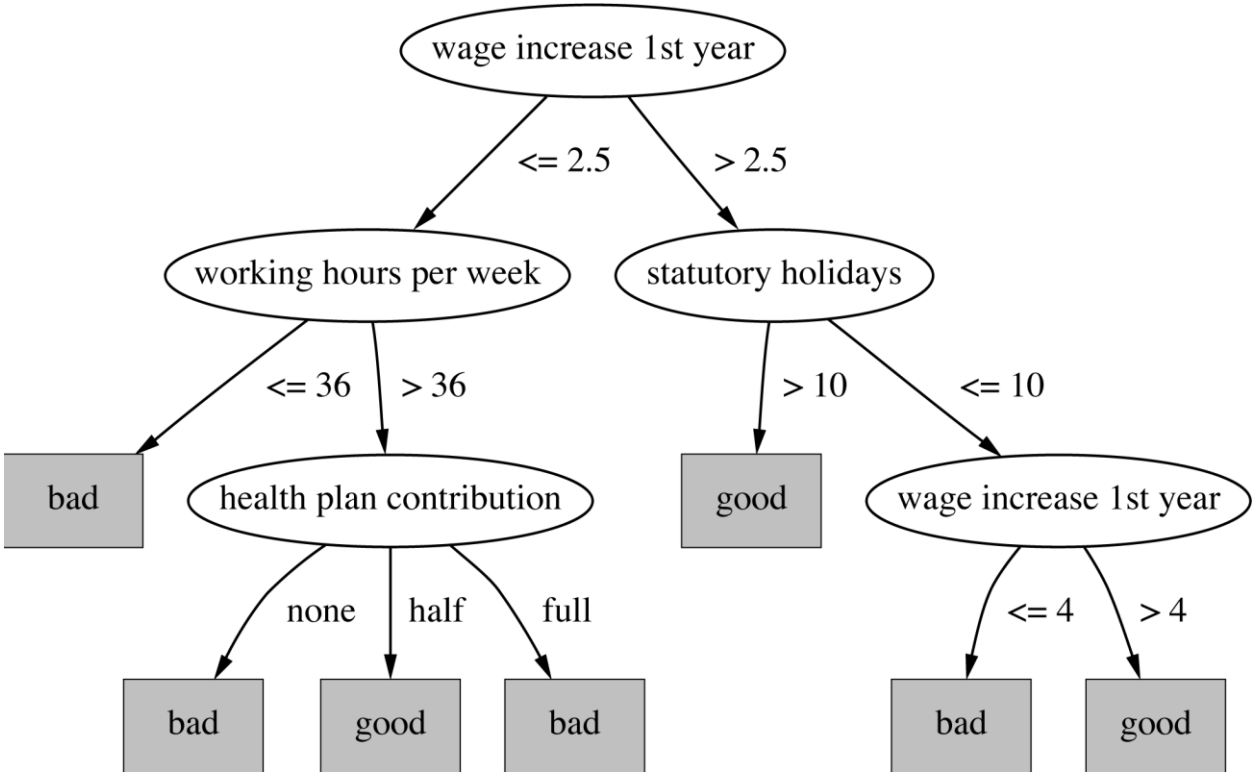
Thay thế cây con

- *Bottom-up*
- thay thế sau khi đã xét tất cả các cây con

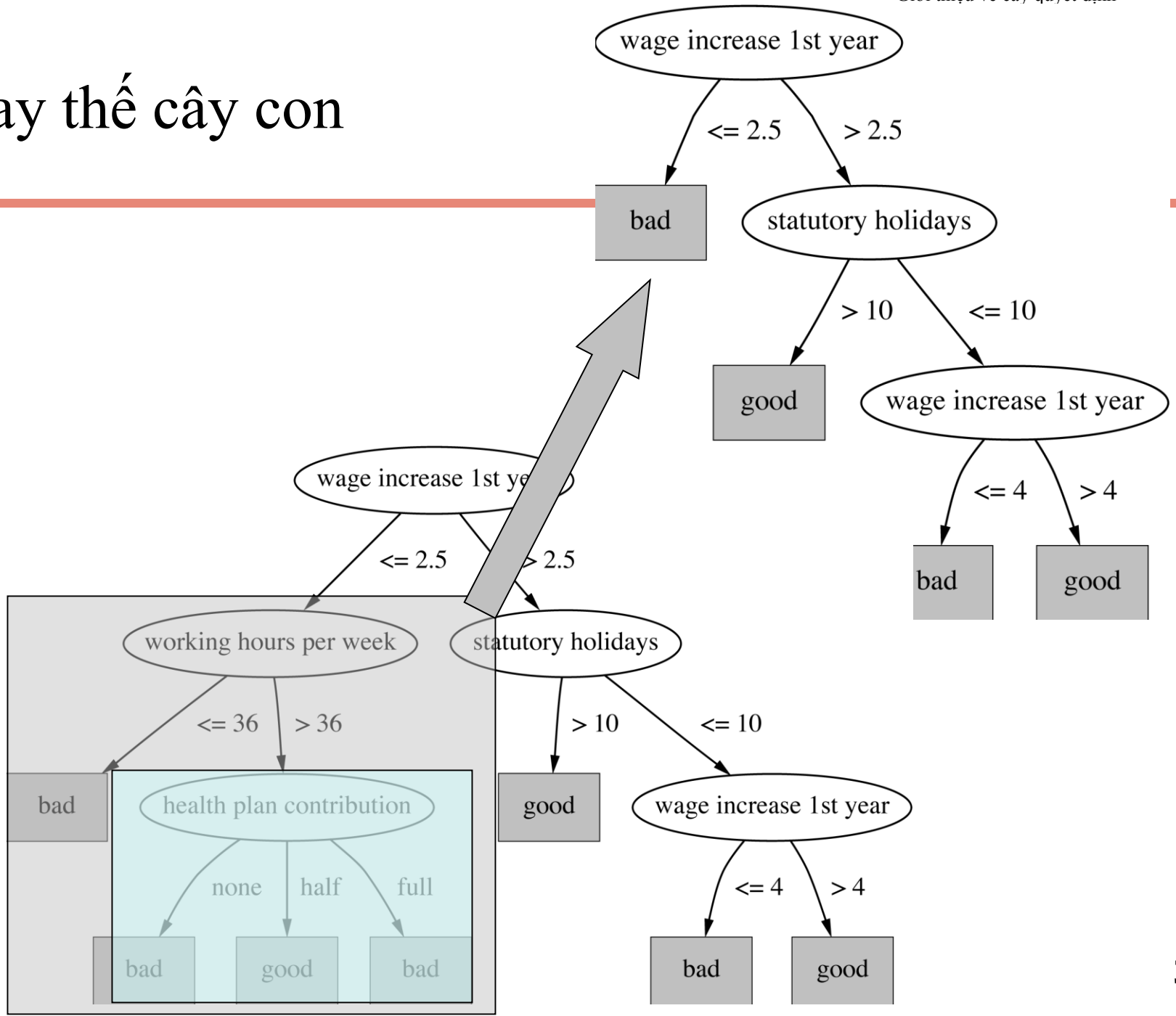


Thay thế cây con

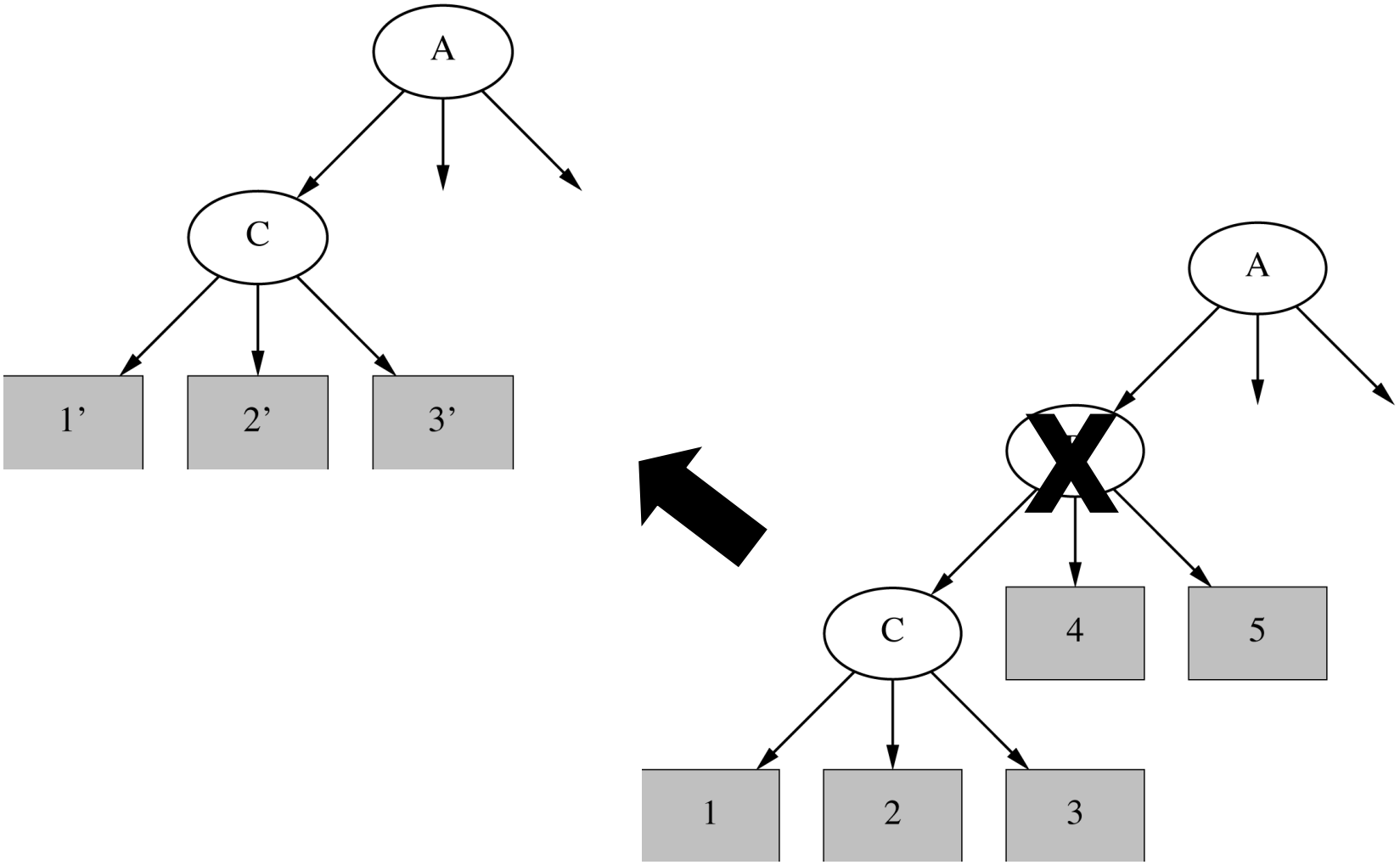
- thay thế cây con nào?



Thay thế cây con



Đưa cây con lên trên



Nội dung

- Giới thiệu về cây quyết định
- Giải thuật học của cây quyết định
- **Kết luận và hướng phát triển**

Kết luận

- cây quyết định
  - xây dựng top-down
  - chọn thuộc tính để phân hoạch (độ lợi thông tin, entropy, chỉ số Gini, etc)
  - cắt nhánh bottom-up
  - dễ cài đặt, học nhanh, kết quả dễ hiểu
  - được sử dụng nhiều và thành công nhất trong các ứng dụng thực

Hướng phát triển

- phát triển
  - tăng độ chính xác
  - xử lý dữ liệu không cân bằng
  - dữ liệu phức tạp có số chiều lớn
  - cây oblique
  - tìm kiếm thông tin (ranking)
  - clustering

Cám ơn !