Cây Quyết Định (Decision Tree)

caotruongtran@gmail.com

Cây quyết định

- Dùng cấu trúc cây để đưa ra một hàm phân lớp cần học (hàm mục tiêu có giá trị rời rạc)

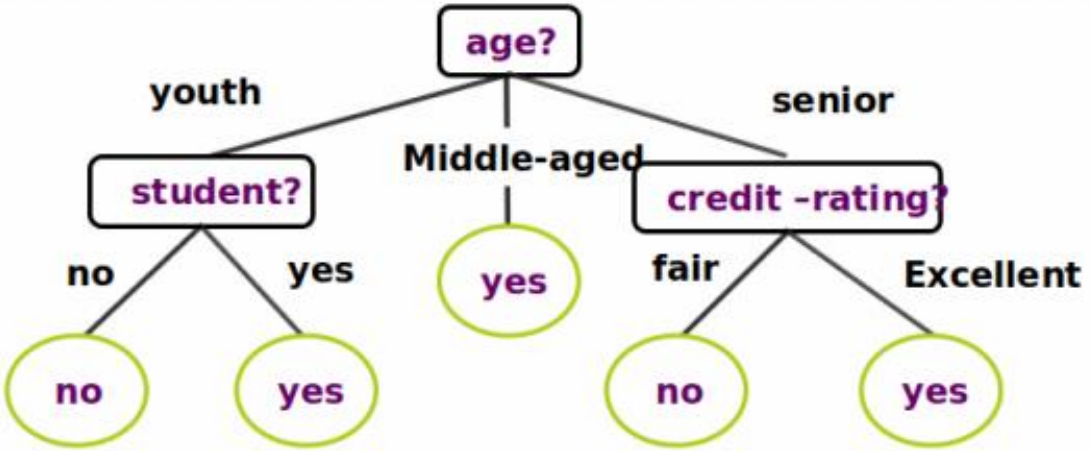


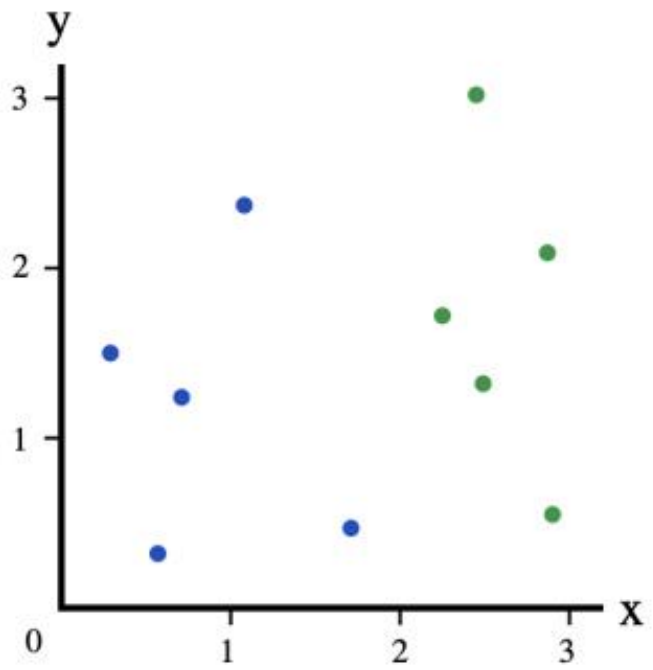
Figure 1: Cây ra quyết định chỉ ra khách hàng nào thường mua máy tính

- Một cây quyết định có thể được biểu diễn (diễn giải) bằng một tập các luật IF-THEN (dễ đọc và dễ hiểu)
- Được áp dụng thành công trong rất nhiều các bài toán ứng dụng thực tế

caotruongtran@gmail.com

Ví dụ về cây quyết định

- Cho tập dữ liệu sau:

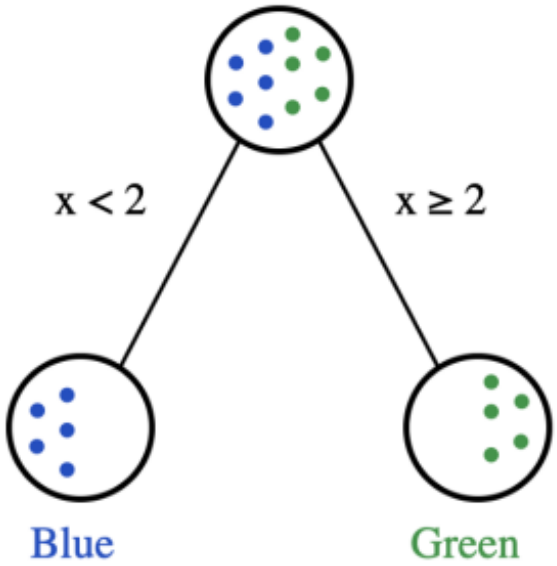


- Có điểm dữ liệu mới với giá trị thuộc tính $x = 1$, màu của điểm này nên là gì? (nên phân vào lớp nào?)

caotruongtran@gmail.com

Ví dụ về cây quyết định

- Đó là cây ra quyết định đơn giản với một node phân loại kiểm tra xem $x < 2$.

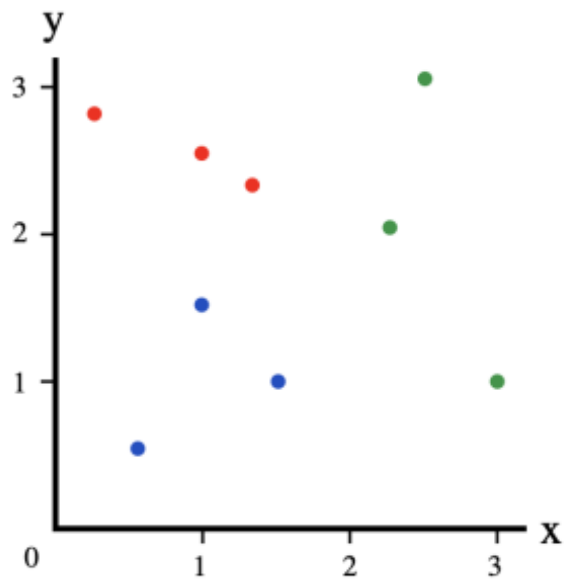


- Nếu kểm tra $x < 2$, chúng ta lấy nhánh trái và gán nhãn màu xanh da trời, nếu kiểm tra không đúng ($x \geq 2$), chúng ta lấy nhánh phải và gán nhãn màu xanh lá cây.

caotruongtran@gmail.com

Ví dụ về cây quyết định

- Bây giờ tập dữ liệu có 3 lớp:

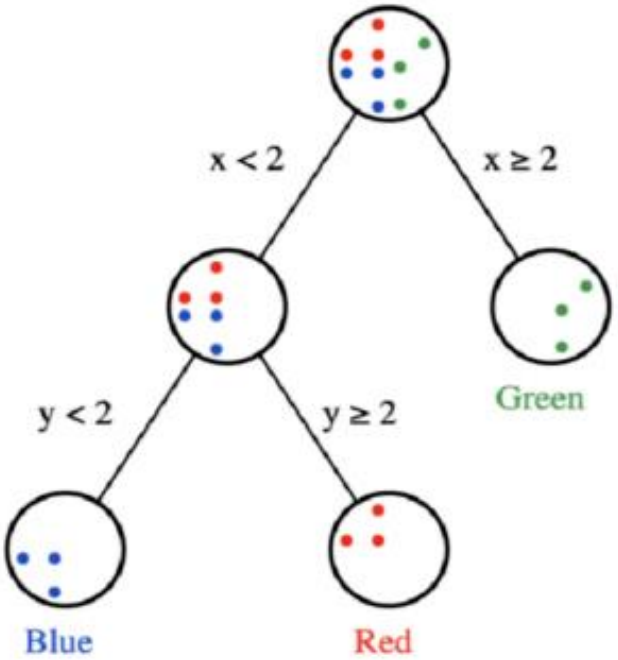


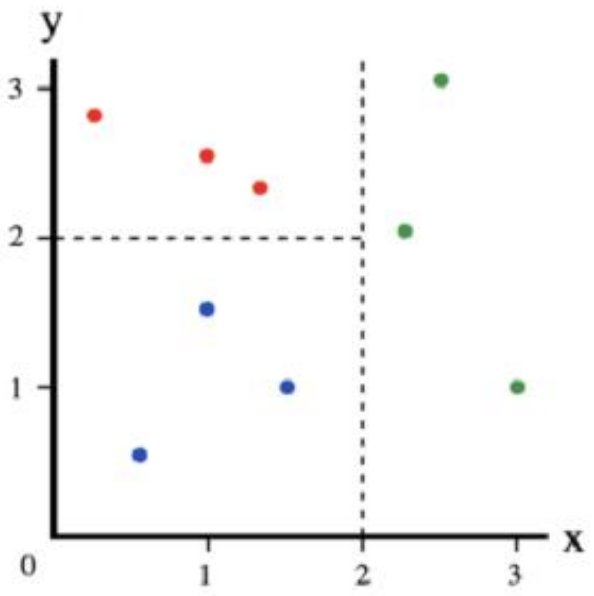
- Cây quyết đĩnh cũ không hiệu quả, với mẫu dữ liệu mới (x, y)
  - Nếu $x \geq 2$, chúng ta có thể vẫn tự tin phân loại vào lớp xanh lá cây
  - Nếu $x < 2$, chúng ta không thể phân loại ngay vào lớp xanh ra trời, nó cũng có thể đỏ.

caotruongtran@gmail.com

Ví dụ về cây quyết định

- Chúng ta cần thêm node quyết định vào cây quyết định





- Đó là ý tưởng chính của cây ra quyết định.

caotruongtran@gmail.com

Biểu diễn cây quyết định

- Mỗi nút trong (internal node) biểu diễn một thuộc tính cần kiểm tra giá trị đối với các ví dụ (mẫu).
- Mỗi nhánh (branch) từ một nút sẽ tương ứng với một giá trị có thể của thuộc tính gắn với nút đó.
- Mỗi nút lá (leaf node) biểu diễn một lớp.
- Một cây quyết định học được sẽ phân lớp đối với một ví dụ, bằng cách duyệt cây từ nút gốc đến một nút lá
- $\rightarrow$ Nhãn lớp gắn với nút lá đó sẽ được gán cho ví dụ cần phân lớp

caotruongtran@gmail.com

Học các cây quyết định

Bài toán: Học xem khi nào thì nên ngồi bàn đợi tại một restaurant:

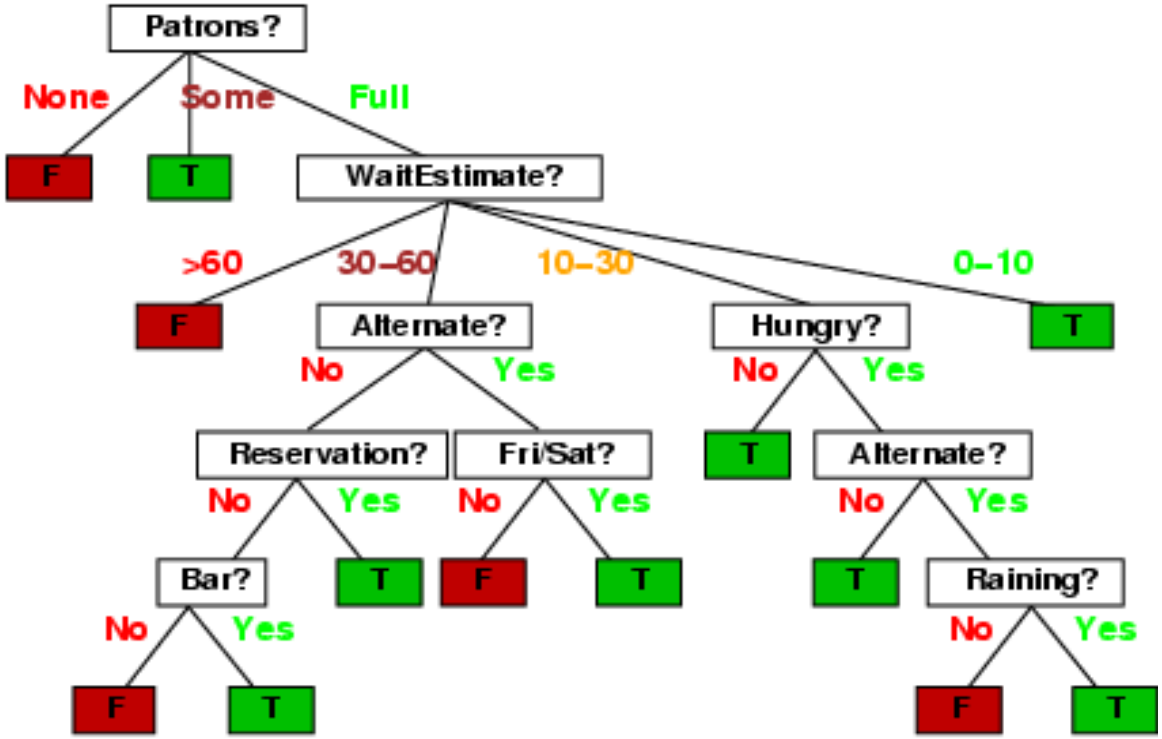
1. Alternate: Có restaurant nào cạnh đây không?
2. Bar: Liệu có khu vực quầy bar có thể ngồi không?
3. Fri/Sat: hôm nay là thứ 6 hay thứ 7?
4. Hungry: có đang đói không?
5. Patrons: Số người trong restaurant (None, Some, Full)
6. Price: khoảng giá ($, $$, $$$)
7. Raining: ngoài trời có mưa không?
8. Reservation: đã đặt trước chưa?
9. Type: loại restaurant (French, Italian, Thai, Burger)
10. WaitEstimate: thời gian chờ đợi (0-10, 10-30, 30-60, >60)

Biểu diễn thuộc tính giá trị

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Cây quyết định

- Biểu diễn giả thiết cần học.
- Ví dụ:



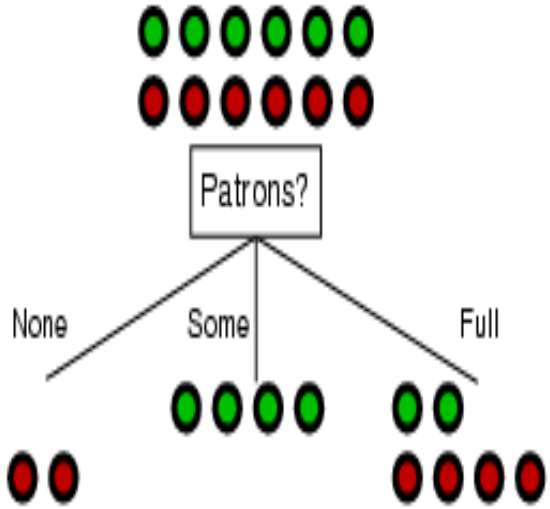
Thuật toán học cây quyết định

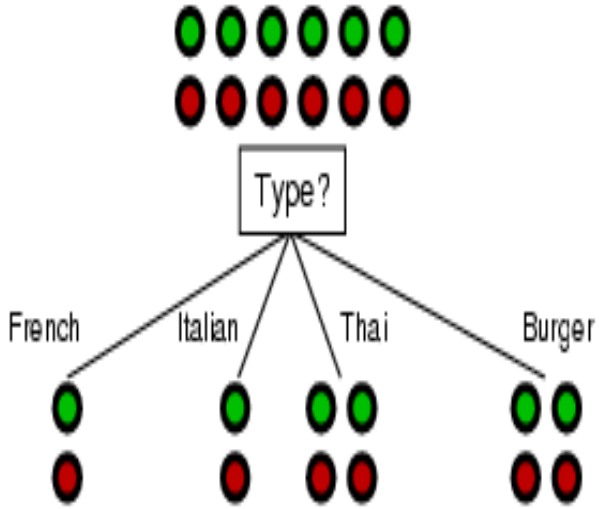
- Mục đích: Tìm cây nhỏ nhất quán với tập mẫu huấn luyện.
- Ý tưởng: Tìm kiếm heuristic chọn thuộc tính quan trọng nhất để phân tách (đệ quy)

```
function DTL(examples, attributes, default) returns a decision tree
    if examples is empty then return default
    else if all examples have the same classification then return the classification
    else if attributes is empty then return MODE(examples)
    else
        best ← CHOOSE-ATTRIBUTE(attributes, examples)
        tree ← a new decision tree with root test best
        for each value v_i of best do
            examples_i ← {elements of examples with best = v_i}
            subtree ← DTL(examples_i, attributes − best, MODE(examples))
            add a branch to tree with label v_i and subtree subtree
        return tree
```

Chọn thuộc tính

- Ý tưởng: chọn thuộc tính (giá trị) sao cho sao cho nó giúp phân tách tập mẫu thanh hai tập thuần khiết (chỉ có positive hay chỉ có negative).





- *Patrons?* là lựa chọn tốt hơn

Sử dụng lý thuyết thông tin

- để cài đặt `Choose-Attribute` trong thuật toán DTL:
- Lượng thông tin (Entropy):

$$I(P(v_1), \ldots, P(v_n)) = \Sigma_{i=1-n} -P(v_i) \log_2 P(v_i)$$

- Đối với tập có *p* mẫu positive và *n* negative:

$$I(\frac{p}{p+n}, \frac{n}{p+n}) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Lợi thông tin (Information gain)

- chọn thuộc tính *A* chia tập huấn luyện *E* thành các tập con $E_1, \ldots, E_v$ tính theo giá trị của *A*, và giả sự *A* có v giá trị khác nhau.

- Lợi thông tin (IG) là độ giảm trong entropy trong việc test thuộc tính:

$$remainder(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

- Chọn thuộc tính có IG lớn nhất

$$IG(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - remainder(A)$$

Lợi thông tin (Information gain)

Trong tập mẫu của ví dụ, $p = n = 6$, $I(6/12, 6/12) = 1$ bit

Xét thuộc tính *Patrons* và *Type* (và các thuộc tính khác):

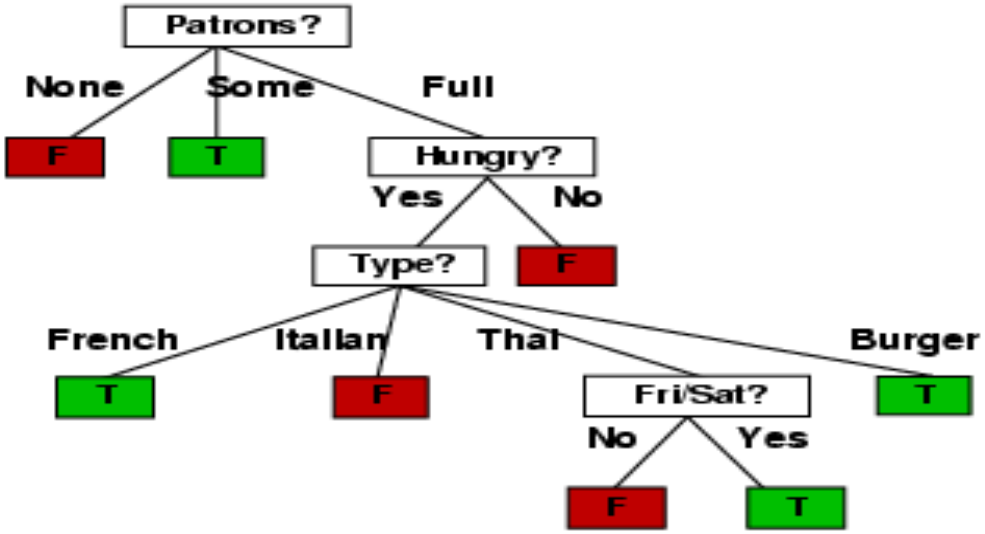
$$IG(Patrons) = 1 - [\frac{2}{12} I(0,1) + \frac{4}{12} I(1,0) + \frac{6}{12} I(\frac{2}{6}, \frac{4}{6})] = .541 \text{ bits}$$

$$IG(Type) = 1 - [\frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{2}{12} I(\frac{1}{2}, \frac{1}{2}) + \frac{4}{12} I(\frac{2}{4}, \frac{2}{4}) + \frac{4}{12} I(\frac{2}{4}, \frac{2}{4})] = 0 \text{ bits}$$

Patrons có giá trị IG cao nhất nên được DTL chọn làm gốc của cây quyết định.

Lợi thông tin (Information gain)

- Cây quyết định học bởi DTL từ 12 ví dụ:



- Nhỏ hơn cây quyết định đưa ra lúc đầu

Xây dựng cây quyết định

RID	age	income	student	credit_rating	Class: buys_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

Xây dựng cây quyết định

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

The following table consists of training data from an employee database. The data have been generalized. For example, "31 ... 35" for *age* represents the age range of 31 to 35. For a given row entry, *count* represents the number of data tuples having the values for *department*, *status*, *age*, and *salary* given in that row.

department	*status*	*age*	*salary*	*count*
sales	senior	31...35	46K...50K	30
sales	junior	26...30	26K...30K	40
sales	junior	31...35	31K...35K	40
systems	junior	21...25	46K...50K	20
systems	senior	31...35	66K...70K	5
systems	junior	26...30	46K...50K	3
systems	senior	41...45	66K...70K	3
marketing	senior	36...40	46K...50K	10
marketing	junior	31...35	41K...45K	4
secretary	senior	46...50	36K...40K	4
secretary	junior	26...30	26K...30K	6