ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÀI 2: HỌC MÁY

Nội dung

1. Các khái niệm cơ bản

- Dữ liệu được miêu tả bởi các thuộc tính nằm trong tập $A = \{A_1, A_2, ..., A_{|A|}\}$
- Thuộc tính lớp $C = \{c_1, c_2, ..., c_{|C|}\}$ ($|C| \geq 2$), c_i là một nhãn lớp
- Mỗi tập DL dùng để học bao gồm các ví dụ chứa thông tin về "kinh nghiệm quá khứ"
- Cho một tập DL D, mục tiêu của việc học là xây dựng một hàm phân loại/dự đoán liên kết các giá trị thuộc tính trong A với các lớp trong C.
- Hàm có thể được sử dụng để phân loại/dự đoán dữ liệu "tương lai"
- Hàm còn được gọi là mô hình phân loại/dự đoán hoặc bộ phân loại

VD về mẫu DL

Bảng 1

ID	Tuổi	Đi làm	Có nhà	Tín dụng	Lớp
1	trẻ	FALSE	FALSE	bình thường	No
2	trẻ	FALSE	FALSE	tốt	No
3	trẻ	TRUE	FALSE	tốt	Yes
4	trẻ	TRUE	TRUE	bình thường	Yes
5	trẻ	FALSE	FALSE	bình thường	No
6	trung niên	FALSE	FALSE	bình thường	No
7	trung niên	FALSE	FALSE	tốt	No
8	trung niên	TRUE	TRUE	tốt	Yes
9	trung niên	FALSE	TRUE	xuất sắc	Yes
10	trung niên	FALSE	TRUE	xuất sắc	Yes
11	già	FALSE	TRUE	xuất sắc	Yes
12	già	FALSE	TRUE	tốt	Yes
13	già	TRUE	FALSE	tốt	Yes
14	già	TRUE	FALSE	xuất sắc	Yes
15	già	FALSE	FALSE	bình thường	No

Học có giám sát

- Học có giám sát: Nhãn lớp được cung cấp trong tập DL
- DL dùng để học gọi là DL huấn luyện
- Sau khi mô hình được học thông qua một thuật toán học, nó được đánh giá trên một tập DL kiểm thử để đo đạc mức độ chính xác
- Không được dùng DL kiểm thử để học mô hình
- Tập DL có nhãn thường được chia làm hai tập độc lập dùng để học và kiểm thử

$$\text{độ chính xác} = \frac{\text{số phân loại đúng}}{\text{tổng số DL kiểm thử}}$$

Học máy là gì?

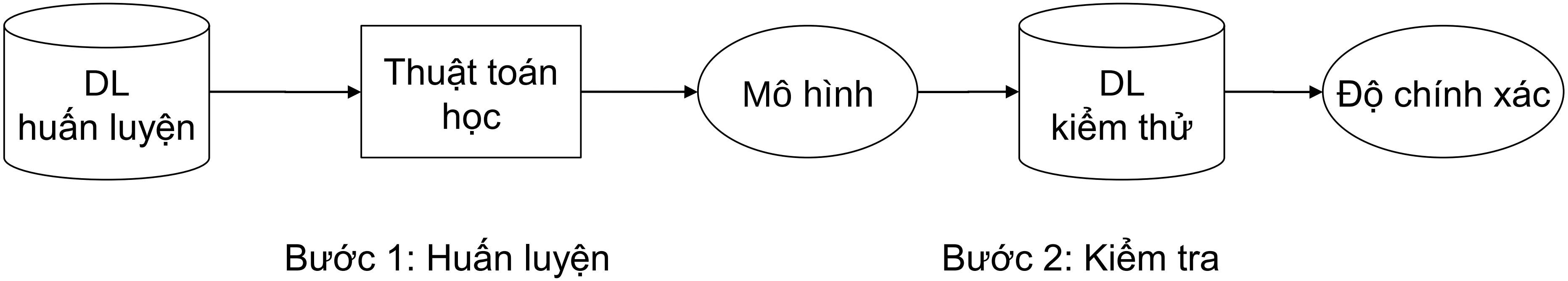
- Cho một tập DL biểu diễn “kinh nghiệm” quá khứ, một tác vụ *T* và một độ đo hiệu năng *M*. Một hệ thống máy tính có khả năng học từ DL để thực hiện tác vụ *T* nếu sau khi học hiệu năng của máy trên tác vụ *T* (được đo bởi *M*) được cải thiện.
- Mô hình học được hoặc tri thức giúp cho hệ thống thực hiện tác vụ tốt hơn so với không học gì.
- Quá trình học là quá trình xây dựng mô hình hoặc trích rút tri thức

Học máy là gì? (tiếp)

- Trong bảng 1, nếu không có quá trình học, giả sử tập DL kiểm thử có cùng phân phối lớp như DL huấn luyện
  - Thực hiện dự đoán một cách ngẫu nhiên → độ chính xác = 50%
  - Thực hiện dự đoán theo lớp phổ biến nhất (lớp *yes*) → độ chính xác = 9/15 = 60%
- Mô hình có khả năng học nếu độ chính xác được cải thiện

Mối quan hệ giữa DL huấn luyện và kiểm thử

- Giả thiết: Phân phối của DL huấn luyện và DL kiểm thử là như nhau



2. P^2 đánh giá
2.1 P^2 đánh giá

- Chia DL ra hai tập huấn luyện và kiểm thử độc lập nhau (thường dùng tỉ lệ 50-50 hoặc 70-30)
  - Lấy mẫu ngẫu nhiên để tạo tập huấn luyện; phần còn lại làm tập kiểm thử
  - Nếu DL được xây dựng theo thời gian, sử dụng quá khứ làm DL huấn luyện
- Nếu DL nhỏ, thực hiện lấy mẫu và đánh giá *n* lần và lấy trung bình
- Cross-validation: DL được chia làm n phần độc lập bằng nhau. Mỗi lần một phần được dùng làm DL kiểm thử và n-1 phần còn lại làm DL huấn luyện. Kết quả được lấy trung bình .
  - Leave-one-out: Nếu DL quá bé, mỗi tập chỉ chứa 1 phần tử, số phần = số phần tử trong tập DL
- Validation set: Sử dụng để lựa chọn các siêu tham số của mô hình (các tham số không học được)

2.2 Các độ đo đánh giá

Ma trận nhập nhằng

	Phân loại tích cực	Phân loại tiêu cực
Tích cực	TP	TN
Tiêu cực	FP	FN

TP: Số lượng phân loại đúng của các ví dụ tích cực (true positive)
FN: Số lượng phân loại sai của các ví dụ tích cực (false negative)
FP: Số lược phân loại sai của các ví dụ tiêu cực (false positive)
TN: Số lượng phân loại đúng của các ví dụ tiêu cực (true negative)

Ví dụ tích cực là ví dụ có nhãn lớp cần quan tâm
Ví dụ tiêu cực là ví dụ có nhãn lớp không quan tâm

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN} \qquad F = \frac{2pr}{p + r}$$

Các độ đo đanh giá (tiếp)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
+	+	+	-	+	-	+	-	+	+	-	-	+	-	-	-	+	-	-	+

G/s có 10 văn bản tích cực

Rank 1:	p = 1/1 = 100%	r = 1/10 = 10%
Rank 2:	p = 2/2 = 100%	r = 2/10 = 20%
…		
Rank 9:	p = 6/9 = 66.7%	r = 6/10 = 60%
Rank 10:	**p = 7/10 = 70%**	**r = 7/10 = 70%**

break-event point

Các độ đo đánh giá (tiếp)

$$TPR = \frac{TP}{TP + FN}$$

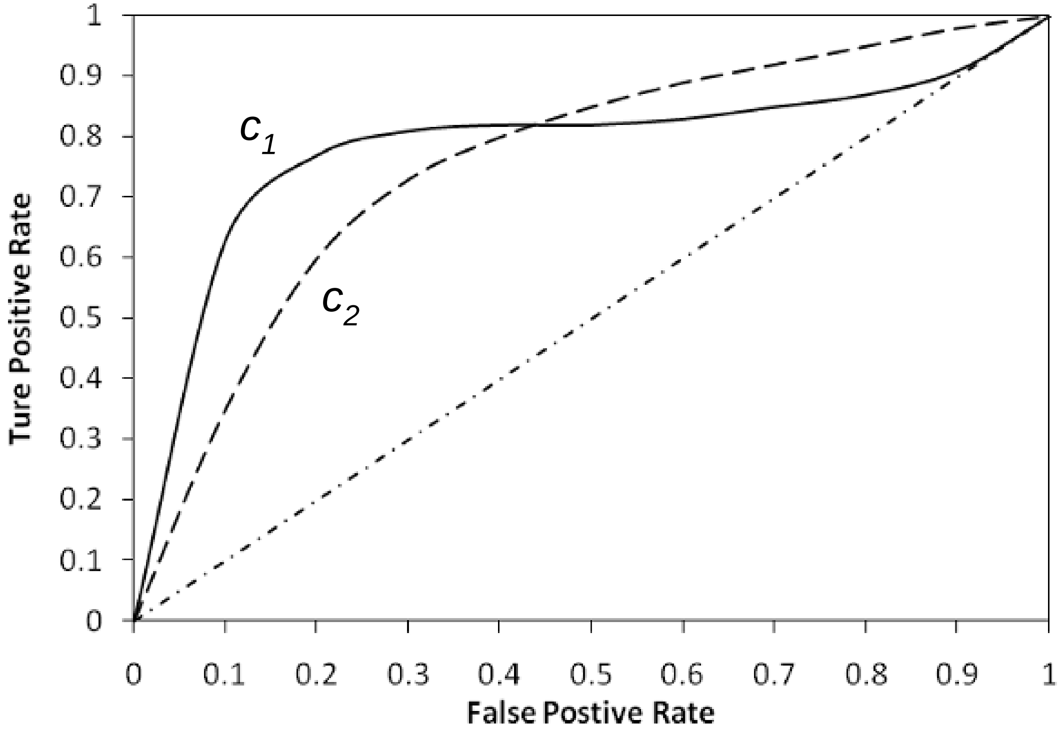
(sensitivity)

$$FPR = \frac{FP}{TN + FP}$$

= (1- specificity)

$$TNR = \frac{TN}{TN + FP}$$

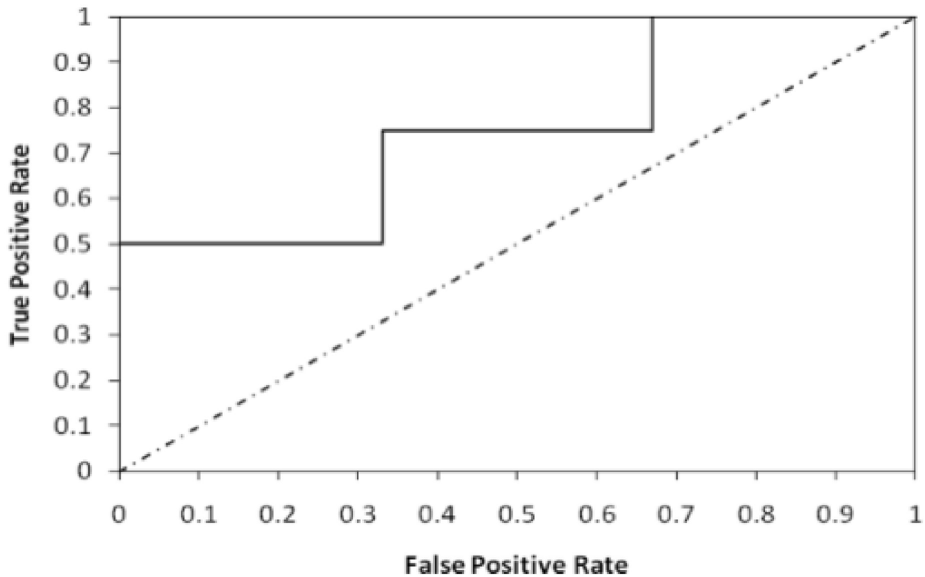
(specificity)



Đường cong ROC của hai bộ phân loại c_1 và c_2 trên cùng tập DL

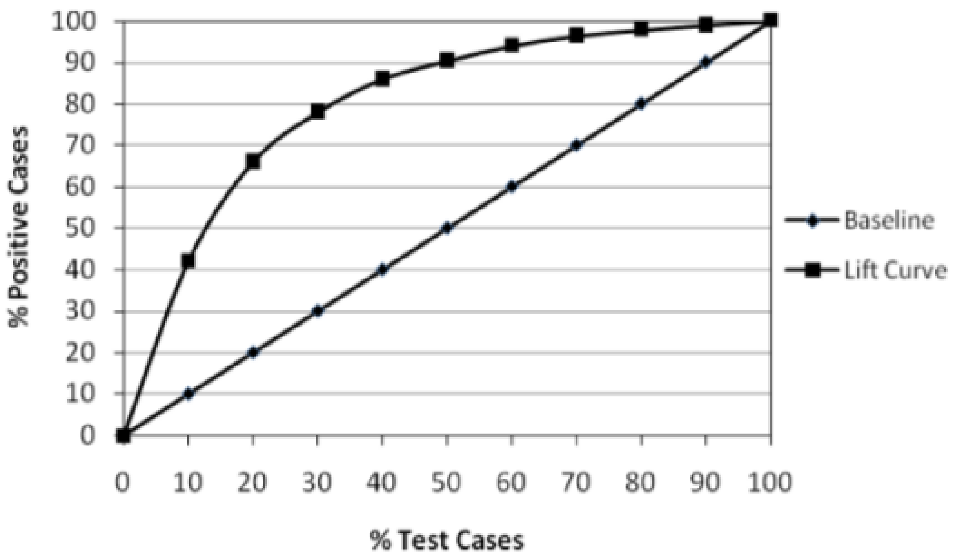
Các độ đo đánh giá (tiếp)

Rank		1	2	3	4	5	6	7	8	9	10
Actual class		+	+	–	–	+	–	–	+	–	–
TP	0	1	2	2	2	3	3	3	4	4	4
FP	0	0	0	1	2	2	3	4	4	5	6
TN	6	6	6	5	4	4	3	2	2	1	0
FN	4	3	2	2	2	1	1	1	0	0	0
TPR	0	0.25	0.5	0.5	0.5	0.75	0.75	0.75	1	1	1
FPR	0	0	0	0.17	0.33	0.33	0.50	0.67	0.67	0.83	1



Các độ đo đánh giá (tiếp)

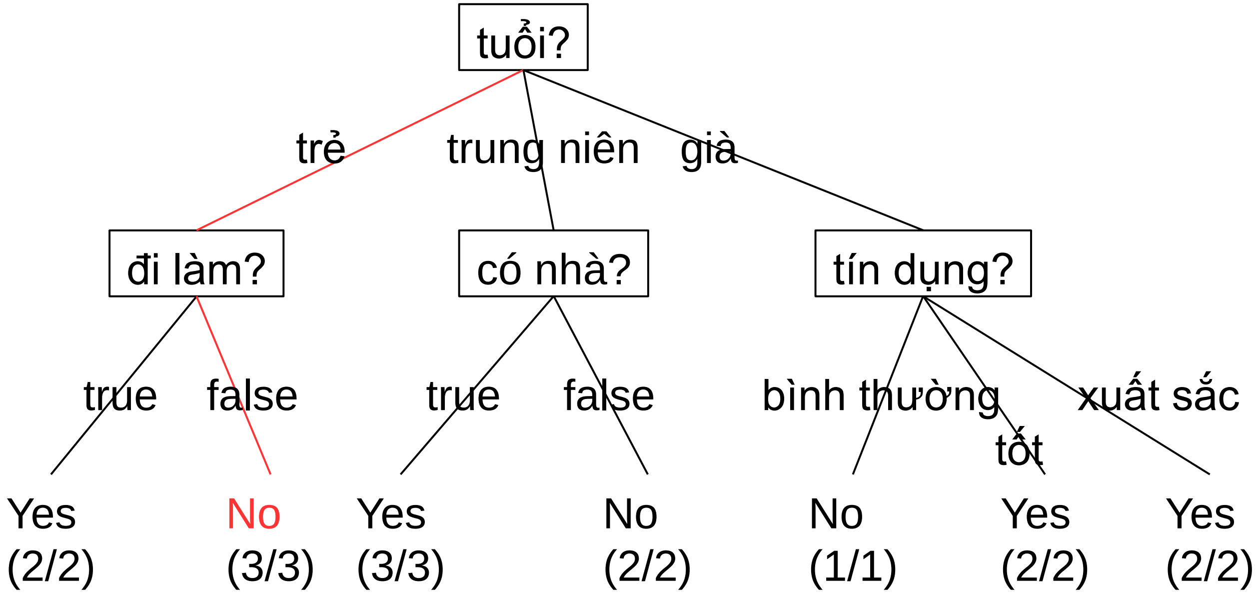
Bin	1	2	3	4	5	6	7	8	9	10
# of test cases	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
# of positive cases	210	120	60	40	22	18	12	7	6	5
% of positive cases	42.0%	24.0%	12%	8%	4.4%	3.6%	2.4%	1.4%	1.2%	1.0%
% cumulative	42.0%	66.0%	78.0%	86.0%	90.4%	94.0%	96.4%	97.8%	99.0%	100.0%



Lift curve tương ứng của DL trong bảng

3. Cây quyết định

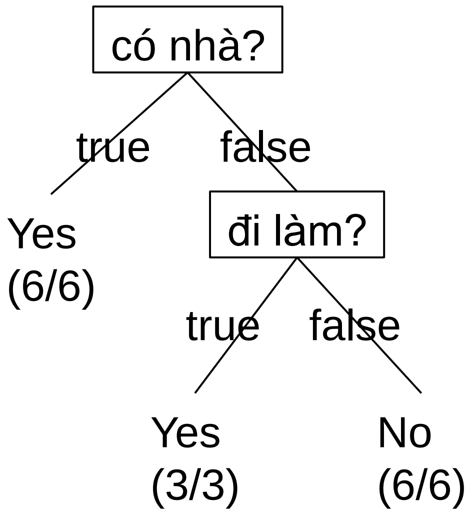
- Nút quyết định, nút lá
- Để dự đoán, duyệt cây từ gốc theo giá trị của các thuộc tính tới khi gặp nút lá



Tuổi	Đi làm	Có nhà	Tín dụng	Lớp
trẻ	FALSE	FALSE	tốt	?

Cây quyết định (tiếp)

- Cây quyết định được xây dựng bằng cách chia DL ra thành các tập con thuần nhất. Tập con gọi là thuần nhất nếu các ví dụ có cùng một lớp.
- Cây có kích thước nhỏ thường tổng quát hơn và có độ chính xác cao hơn; dễ hiểu hơn với con người
- Cây sinh ra không phải duy nhất
- Tìm cây tốt nhất là một bài toán NP-đầy đủ → sử dụng các giải thuật heuristic



có nhà = true → lớp =Yes [sup=6/15, conf=6/6]

có nhà = false, đi làm = true → lớp = Yes [sup=3/15, conf=3/3]

có nhà = false, đi làm = false → lớp = No [sup=6/15, conf=6/6].

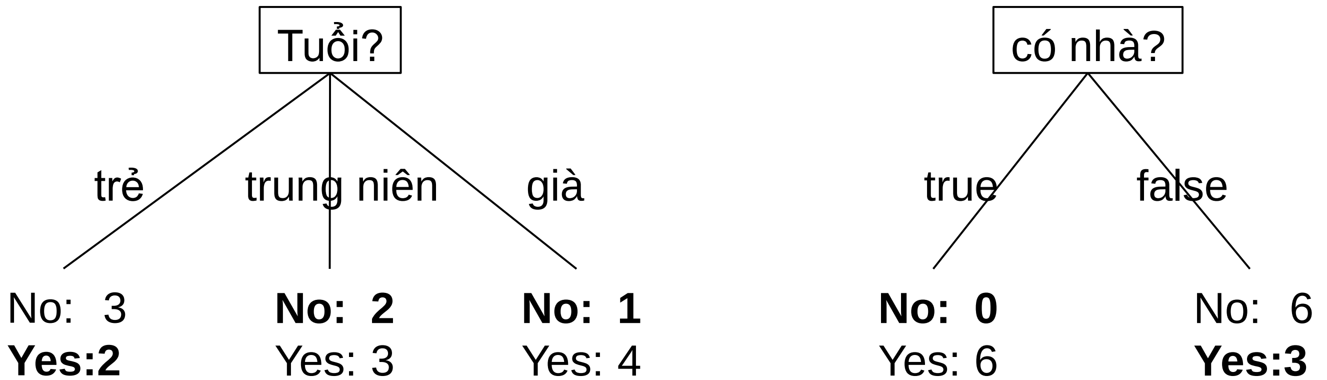
tuổi = trẻ, đi làm = false → lớp = No [sup=3/15, conf=3/3]

3.1 Thuật toán học

- Dùng chiến thuật *chia-để-trị* để phân chia DL một cách đệ quy
- Điều kiện dừng: tất cả các ví dụ đều có cùng lớp hoặc tất cả các thuộc tính đã được sử dụng (line 1-4)
- Tại mỗi bước đệ quy, chọn thuộc tính tốt nhất để chia DL theo giá trị của thuộc tính dựa trên hàm tính độ không thuần nhất (line 7-11)
- Thuật toán tham lam

```
Algorithm decisionTree(D, A , T)
    if D chỉ chứa ví dụ huấn luyện của lớp c_j ∈ C then
        tạo nút lá T với nhãn lớp c_j;
    elseif A = ∅ then
        tạo nút lá T với nhãn lớp c_j là lớp phổ biến nhất trong D
    else  // D chứa ví dụ với nhiều lớp. Lựa chọn một thuộc tính
        // để chia D thành các tập con để mỗi tập con thuần nhất hơn
        p_0 = impurityEval-1(D);
        for mỗi thuộc tính A_i ∈ A (={A_1, A_2, ..., A_k}) do
            p_i = impurityEval-2(A_i, D)
        endfor
        Lựa chọn A_g ∈ {A_1 ,A_2 , ..., A_k} làm giảm nhiều nhất độ không thuần nhất theo p_0 - p_i;
        if p_0 - p_g < threshold then            // A_g không giảm đáng kể độ không thuần nhất p_0
            tạo nút lá T với nhãn lớp c_j là lớp phổ biến nhất trong D
        else                                     // A_g làm giảm độ không thuần nhất p_0
            Tạo nút quyết định T theo A_g;
            Cho các giá trị của A_g là v_1, v_2, ..., v_m. Chia D thành m
            tập con không giao nhau D_1, D_2, ..., D_m dựa trên m giá trị của A_g.
            for mỗi D_j trong {D_1, D_2, ..., D_m} do
                if D_j ≠ ∅ then
                    tạo một nhánh (cạnh) ứng với nút T_j cho v_j là con của T;
                    decisionTree{D_j, A - {A_g}, T_j)        // xóa A_g
                endif
            endfor
        endif
    endif
```

3.2 Hàm không thuần nhất



a) b)

Độ không thuần nhất của cây a) cao hơn cây b)

$$\text{entropy}(D) = -\sum_{j=1}^{|C|} \Pr(c_j)\log_2 \Pr(c_j)$$

$$\Pr(c_j) = \sum_{j=1}^{|C|} \Pr(c_j)$$

- $\Pr(c_j)$ là xác suất dữ liệu thuộc lớp c_j
- Đơn vị của entropy là bit
- Quy ước $0\log_2 0 = 0$
- Dữ liệu càng thuần nhất thì entropy càng nhỏ và ngược lại

Vd6:

Tập DL D có hai lớp tích cực (pos) và tiêu cực (neg)

1. Pr(pos) = Pr(neg) = 0.5 → entropy(D) = $-0.5 \times \log_2 0.5 - 0.5 \times \log_2 0.5 = 1$
2. Pr(pos) = 0.2, Pr(neg) = 0.8 → entropy(D) = $-0.2 \times \log_2 0.2 - 0.8 \times \log_2 0.8 = 0.722$
3. Pr(pos) = 1, Pr(neg) = 0 → entropy(D) = $-1 \times \log_2 1 - 0 \times \log_2 0 = 0$

Information gain

1. Tính entropy(D) (line 7)
2. Lựa chọn thuộc tính: Với mỗi thuộc tính A_i, giả sử có v giá trị, chia D thành các tập con không giao nhau $D_1, D_2, \ldots, D_v$ (line 9)

$$entropy_{A_i}(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times entropy(D_j)$$

3. Độ tăng cường thông tin của A_i

$$gain(D, A_i) = entropy(D) - entropy_{A_i}(D)$$

VD

$entropy(D) = -6/15 \times \log_2 6/15 – 9/15 \log_2 9/15 = 0.971$

$entropy_{tuổi}(D) = 5/15 \times entropy(D_{tuổi=trẻ}) + 5/15 \times entropy(D_{tuổi=trung\ niên}) + 5/15 \times entropy(D_{tuổi=già})$

$= 5/15 \times 0.971 + 5/15 \times 0.971 + 5/15 \times 0.722 = 0.888$

$entropy_{có\ nhà}(D) = 6/15 \times entropy(D_{có\ nhà=true}) + 9/15 \times entropy(D_{có\ nhà=false})$

$= 6/15 \times 0 + 9/15 \times 0.918 = 0.551$

$entropy_{đi\ làm}(D) = 0.647$

$entropy_{tín\ dụng}(D) = 0.608$

gain(D, tuổi) = 0.971 - 0.888 = 0.083

gain(D, có nhà) = 0.971 - 0.551 = 0.420

gain(D, đi làm) = 0.971 - 0.647 = 0.324

gain(D, tín dụng) = 0.971 - 0.608 = 0.363.

Information gain ratio

- Information gain thường thiên vị các thuộc tính có nhiều giá trị
- Gain ratio chuẩn hóa entropy theo các giá trị của thuộc tính

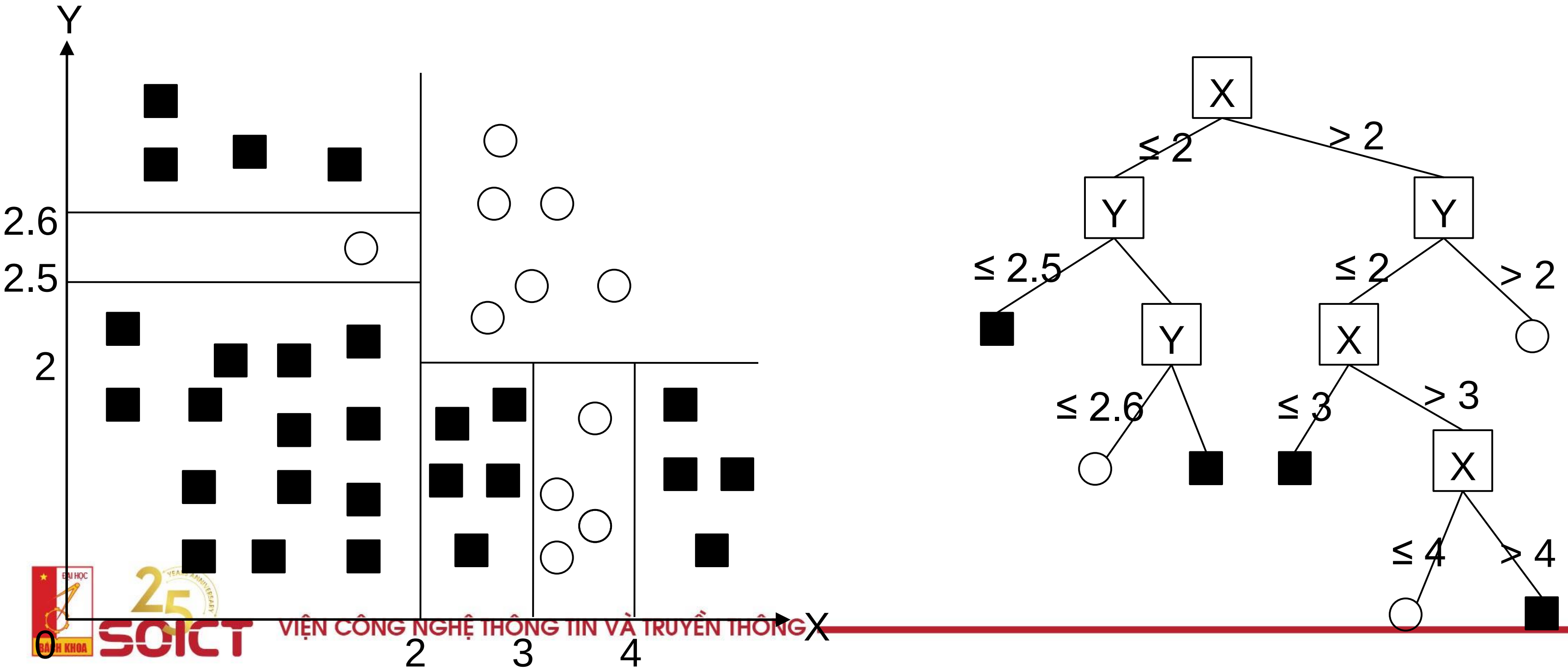
s: số giá trị khác nhau của A_i

D_j là tập con của D có thuộc tính A_i có giá trị thứ j

$$\text{GainRatio}(D, Ai) = \frac{\text{Gain}(D, Ai)}{-\sum_{j=1}^{s}\left(\frac{Dj}{Dj} + \frac{Dj}{Dj}\right)}$$

3.3 Xử lý thuộc tính liên tục

- Chia thuộc tính ra làm hai khoảng (binary split)
- Ngưỡng chia được lựa chọn sao cho cựa đại hóa information gain (ratio)
- Trong quá trình tạo cây, thuộc tính không bị xóa bỏ (line 20)



3.4 Một số vấn đề nâng cao

- **Overfitting**: Một bộ phân loại *f* gọi là quá khít nếu tồn tại một bộ phân loại *f'* mà độ chính xác của *f* > *f'* trên DL huấn luyện nhưng < trên DL kiểm thử
  - Nguyên nhân: DL chứa nhiễu (nhãn lớp sai hoặc giá trị thuộc tính sai) hoặc bài toán phân loại phức tạp hoặc chứa đựng tính ngẫu nhiên
  - Pruning: Cây quyết định quá sâu → cắt tỉa cây bằng cách ước lượng lỗi tại mỗi nhánh, nếu lỗi của cây con lớn hơn thì cắt tỉa. Có thể sử dụng một tập DL độc lập (validation set) để cắt tỉa. Ngoài ra có thể áp dụng cắt tỉa trước hoặc cắt tỉa luật
- Giá trị khuyết thiếu: Sử dụng giá trị "unknown" hoặc giá trị phổ biến nhất (hoặc giá trị trung bình với thuộc tính liên tục)
- Mất cân bằng lớp: Over sampling, xếp hạng

Thuộc tính liên tục

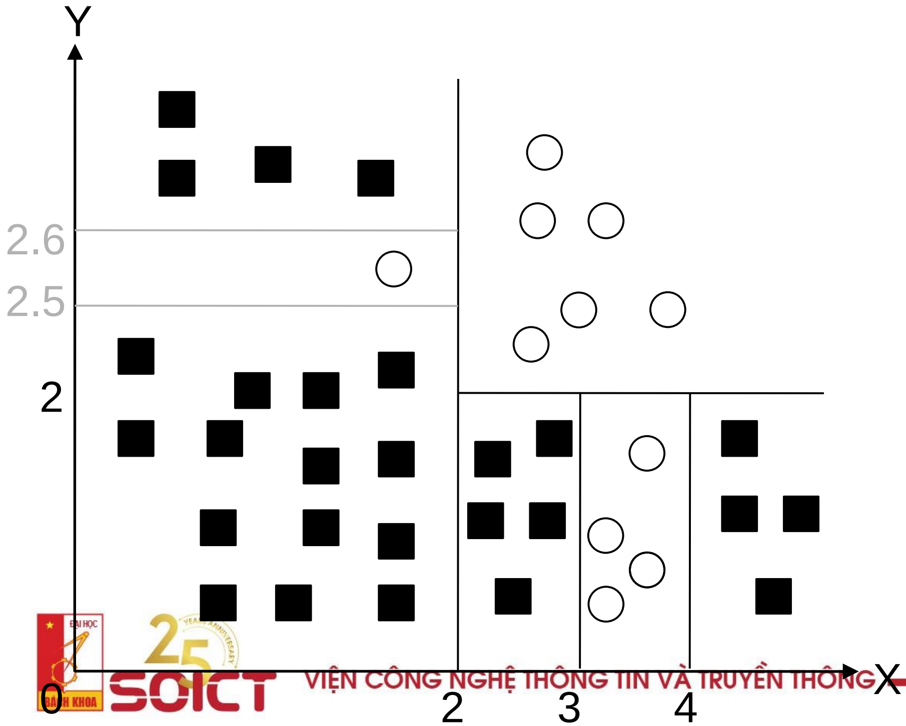
Luật 1: X ≤ 2, Y > 2.5, Y > 2.6 → ■
Luật 2: X ≤ 2, Y > 2.5, Y ≤ 2.6 → ○
Luật 3: X ≤ 2, Y ≤ 2.5 → ■

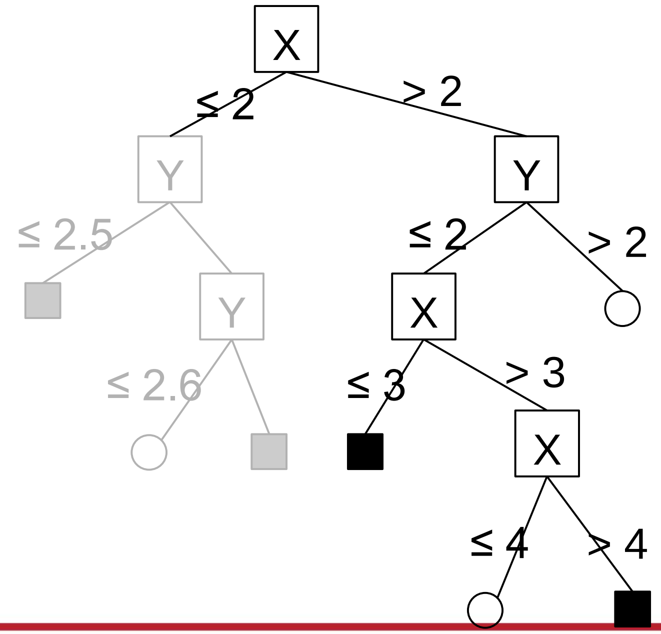
→

X ≤ 2, Y > 2.6 → ■

↓

X ≤ 2 → ■





4. Thuật toán Naive Bayes
4.1 Thuật toán Naive Bayes

- Cho tập các thuộc tính $A_1, A_2, ..., A|A|$, C là thuộc tính lớp với các giá trị $c_1, c_2, ..., c_{|C|}$, ví dụ kiểm thử $d = <A_1=a_1, ..., A_{|A|}=a_{|A|}>$
- Giả thiết MAP (maximum a posteriori): tìm lớp c_j sao cho $\Pr(C=c_j|A_1=a_1, ..., A_{|A|}=a_{|A|})$ cực đại

$$\Pr(C=c_j| A_1=a_1,..., A_{|A|}=a_{|A|}) = \frac{\Pr(A_1=a_1, ..., A_{|A|}=a_{|A|}| C=c_j) \times \Pr(C=c_j)}{\Pr(A_1=a_1, ..., A_{|A|}=a_{|A|})}$$

$$\Pr(C=c_j| A_1=a_1,..., A_{|A|}=a_{|A|}) \propto \Pr(A_1=a_1, ..., A_{|A|}=a_{|A|}| C=c_j) \times \Pr(C=c_j)$$

Thuật toán Naive Bayes (tiếp)

$$\begin{aligned}
\Pr(A_1=a_1, ..., A_{|A|}=a_{|A|} \mid C=c_j) &= \Pr(A_1=a_1 \mid A_2=a_2,..., A_{|A|}=a_{|A|}, C=c_j) \\
&\quad \times \Pr(A_2=a_2,..., A_{|A|}=a_{|A|} \mid C=c_j) \\
&= \Pr(A_1=a_1 \mid A_2=a_2,..., A_{|A|}=a_{|A|}, C=c_j) \\
&\quad \times \Pr(A_2=a_2 \mid A_3=a_3,..., A_{|A|}=a_{|A|}, C=c_j) \\
&\quad \times \Pr(A_3=a_3,..., A_{|A|}=a_{|A|} \mid C=c_j) \\
&= ...
\end{aligned}$$

Giả thiết độc lập có điều kiện:

$\Pr(A_1=a_1 \mid A_2=a_2,..., A_{|A|}=a_{|A|}, C=c_j) = \Pr(A_1=a_1 \mid C=c_j)$

$\Pr(A_2=a_2 \mid A_3=a_3,..., A_{|A|}=a_{|A|}, C=c_j) = \Pr(A_2=a_2 \mid C=c_j)$

...

$$\Pr(A_1=a_1, ..., A_{|A|}=a_{|A|} \mid C=c_j) = \Pr(A_1=a_1 \mid C=c_j) \times \Pr(A_2=a_2 \mid C=c_j) \times \ldots \times \Pr(A_n=a_n \mid C=c_j)$$

$$\boxed{\Pr(C=c_j \mid A_1=a_1,..., A_{|A|}=a_{|A|}) \propto \Pr(A_1=a_1 \mid C=c_j) \times \Pr(A_2=a_2 \mid C=c_j) \times \ldots \times \Pr(A_n=a_n \mid C=c_j) \times \Pr(C=c_j)}$$

Thuật toán Naive Bayes (tiếp)

$$\Pr(C=c_j) = \frac{\text{số lượng ví dụ có lớp } c_j}{\text{tổng số ví dụ trong tập DL}}$$

$$\Pr(A_i=a_i \mid C=c_j) = \frac{\text{số lượng ví dụ có } A_i = a_i \text{ và lớp } c_j}{\text{số lượng ví dụ có lớp } c_j}$$

A	B	C
m	b	t
m	s	t
g	q	t
h	s	t
g	q	t
g	q	f
g	s	f
h	b	f
h	q	f
m	b	f

Pr(C = t) = 1/2 Pr(C = f) = 1/2

Pr(A = m| C = t) = 2/5
Pr(A = m| C = f) = 1/5
Pr(B = b| C = t) = 1/5
Pr(B = b| C = f) = 2/5

Pr(A = g| C = t) = 2/5
Pr(A = g| C = f) = 2/5
Pr(B = s| C = t) = 2/5
Pr(B = s| C = f) = 1/5

Pr(A = h| C = t) = 1/5
Pr(A = h| C = f) = 2/5
Pr(B = q | C = t) = 2/
Pr(B = q | C = f) = 2/

A =m, B = q, C = ?

Pr(C = t| A = m, B = q) $\propto$ Pr(C = t) x Pr(A = m| C = t) x Pr(B = q| C = t)
$\propto$ 1/2 x 2/5 x 2/5 = 2/25

Pr(C = f| A = m, B = q) $\propto$ Pr(C = f) x Pr(A = m| C = f) x Pr(B = q| C = f)
$\propto$ 1/2 x 1/5 x 2/5 = 1/25

Làm mịn

- Xác suất – 0: Giá trị thuộc tính a_i không x/h cùng lớp c_j trong DL huấn luyện khiến $\Pr(A_i = a_i|C = c_j) = 0$

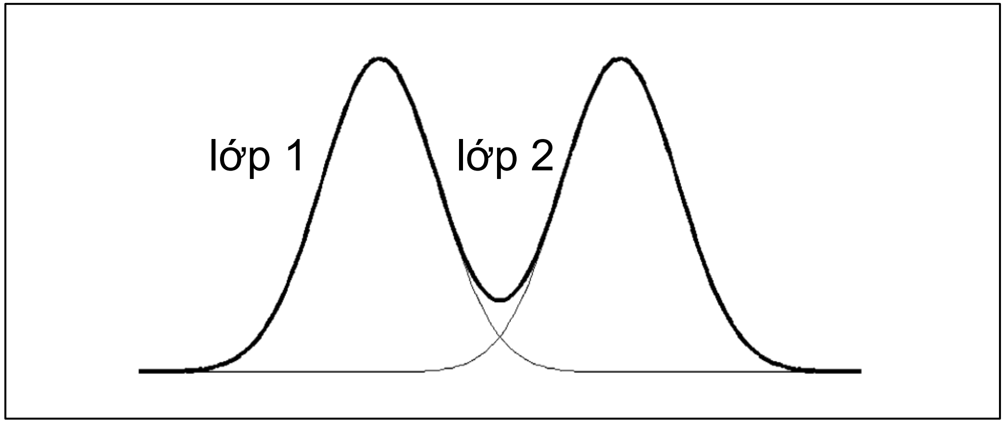
$$\Pr(A_i = a_i | C = c_j) = \frac{n_{ij} + \lambda}{n_j + \lambda \times m_i}$$

trong đó n_{ij} là số ví dụ có $A_i = a_i$ và $C = c_j$; m_i là số giá trị khác nhau của thuộc tính A_i

- $\lambda = 1/n$ trong đó n là số lượng ví dụ huấn luyện
- $\lambda = 1$: Làm mịn *Laplace*

4.2 Phân loại văn bản dựa trên NB

- Mô hình sinh xác suất: Giả thiết mỗi văn bản được sinh ra bởi một phân phối theo các tham số ẩn. Các tham số này được ước lượng dựa trên DL huấn luyện. Các tham số được dùng để phân loại văn bản kiểm thử bằng cách sử dụng định luật Bayes để tính toán xác suất hậu nghiệm của lớp có khả năng sinh ra văn bản.
- Hai giả thiết: *i*) DL được sinh ra bởi một mô hình trộn *ii*) Mỗi thành phần trộn ứng với một lớp



Hàm phân phối xác suất của hai phân phối Gaussian với các tham số $\theta_1 = (\mu_1, \sigma_1)$ và $\theta_1 = (\mu_1, \sigma_1)$

Phân loại văn bản dựa trên NB (tiếp)

- Giả sử có K thành phần trộn, thành phần j có tham số θ_j, tham số của toàn bộ mô hình bao gồm $\Theta = (\varphi_1, \varphi_2, \ldots, \varphi_k, \theta_1, \theta_2, \ldots, \theta_k)$ trong đó φ_j là trọng số của thành phần j $(\Sigma\varphi_j = 1)$
- Giả sử có các lớp $c_1, c_2, \ldots, c_{|C|}$, ta có $|C| = K$, $\varphi_j = \Pr(c_j|\Theta)$, quá trình sinh văn bản d_i:

1. Lựa chọn một thành phần trộn j dựa trên xác suất tiên nghiệm của các lớp, $\varphi_j = \Pr(c_j|\Theta)$
2. Sinh ra d_i dựa trên phân phối $\Pr(d_i| c_j; \theta_j)$

- Xác suất sinh ra di dựa trên toàn bộ mô hình:

$$\Pr(d_i| \Theta) = \sum_{j=1}^{|C|} \Pr(c_j| \Theta) \times \Pr(d_i| c_j; \theta_j)$$

Phân loại văn bản dựa trên NB (tiếp)

- Văn bản được biểu diễn như một túi từ

1. Các từ được sinh ra độc lập, không phụ thuộc và ngữ cảnh (các từ khác trong văn bản)
2. Xác suất của từ không phụ thuộc vào vị trí trong văn bản
3. Độ dài và lớp của văn bản độc lập với nhau

- Mỗi văn bản được sinh ra bởi một phân phối đa thức của từ với n phép thử độc lập trong đó n là độ dài của văn bản

Phân loại văn bản dựa trên NB (tiếp)

- Phép thử đa thức là quá trình sinh ra k giá trị ($k \geq 2$) với các xác suất $p_1, p_2, \ldots, p_k$

VD: Xúc xắc sinh ra 6 giá trị 1, 2, …, 6 với xác suất công bằng $p_1 = p_2 = \ldots = p_6 = 1/6$)

- Giả sử có n phép thử độc lập, gọi X_t là số lần sinh ra giá trị t, khi đó $X_1, X_2, \ldots, X_k$ là các biến ngẫu nhiên rời rạc.

$(X_1, X_2, \ldots, X_k)$ tuân theo phân phối đa thức với các tham số $n, p_1, p_2, \ldots, p_\mathrm{k}$

Phân loại văn bản dựa trên NB (tiếp)

- n: Độ dài văn bản $|d_i|$
- $k = |V|$: Số lượng từ vựng trong tập DL
- p_t: Xác suất từ w_t x/h trong văn bản, $\Pr(w_t | c_j; \Theta)$
- X_t: Biến ngẫu nhiên thể hiện số lần w_t x/h trong văn bản
- N_{ti}: Số lần w_t x/h trong văn bản d_i
- Hàm xác suất:

$$\Pr(d_i | c_j; \Theta) = \Pr(|d_i|)\, |d_i|! \prod_{t=1}^{|V|} \frac{\Pr(w_t | c_j; \Theta)^{N_{ti}}}{N_{ti}!}$$

$$\sum_{t=1}^{|V|} N_{ti} = |d_i| \qquad \sum_{t=1}^{|V|} \Pr(w_t | c_j; \Theta) = 1$$

Phân loại văn bản dựa trên NB (tiếp)

- Ước lượng tham số:

$$\Pr(w_t \mid c_j; \hat{\Theta}) = \frac{\sum_{i=1}^{|D|} N_{ti} \Pr(c_j \mid d_i)}{\sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{si} \Pr(c_j \mid d_i)}.$$

- Làm mịn *Lidstone* (λ < 1)

(λ = 1: Làm mịn *Laplace*)

$$\Pr(w_t \mid c_j; \hat{\Theta}) = \frac{\lambda + \sum_{i=1}^{|D|} N_{ti} \Pr(c_j \mid d_i)}{\lambda |V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{si} \Pr(c_j \mid d_i)}$$

Phân loại văn bản dựa trên NB (tiếp)

$$\Pr(c_j \mid \hat{\Theta}) = \frac{\sum_{i=1}^{|D|} \Pr(c_j \mid d_i)}{|D|}$$

- Phân loại:

$$\Pr(c_j \mid d_i; \hat{\Theta}) = \frac{\Pr(c_j \mid \hat{\Theta}) \Pr(d_i \mid c_j; \hat{\Theta})}{\Pr(d_i \mid \hat{\Theta})}$$

$$= \frac{\Pr(c_j \mid \hat{\Theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_j; \hat{\Theta})}{\sum_{r=1}^{|C|} \Pr(c_r \mid \hat{\Theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_r; \hat{\Theta})}$$

$$\arg\max_{c_j \in C} \Pr(c_j \mid d_i; \hat{\Theta}).$$

5. Thuật toán SVM

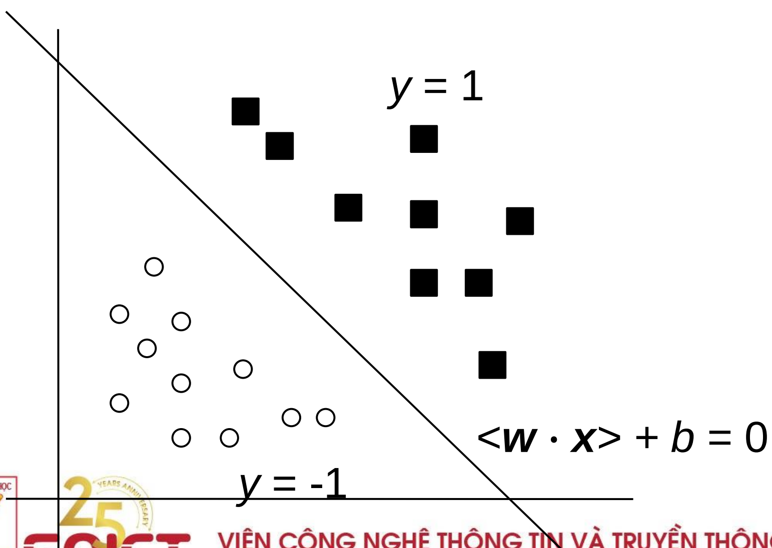
- Máy véc-tơ hỗ trợ (SVM) là một hệ thống học tuyến tính nhằm xây dựng bộ phân loại 2-lớp
- Tập ví dụ D: $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots(\boldsymbol{x}_n, y_n)\}$ trong đó $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ véc-tơ đầu vào r-chiều trong không gian $X \subseteq R^r$, y_i là nhãn lớp, $y_i \in \{1, -1\}$
- SVM xây dựng hàm tuyến tính f: $X \quad R^r \rightarrow R$ với $\boldsymbol{w} = (w_1, w_2, \ldots, w_r) \in R^r$

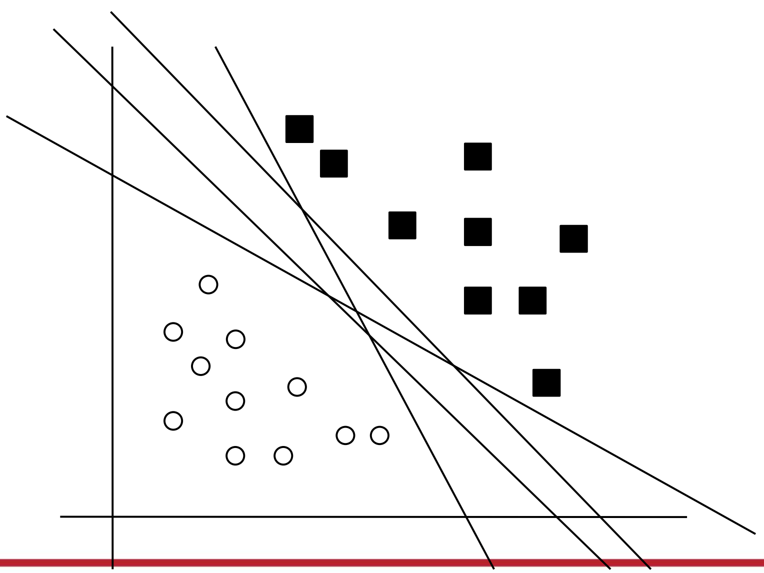
$$f(\boldsymbol{x}) = \langle \boldsymbol{w} \cdot \boldsymbol{x} \rangle + b$$

$$y_i = \begin{cases} 1 & \text{nếu } f(\boldsymbol{x}_i) \geq 0 \\ -1 & \text{nếu } f(\boldsymbol{x}_i) < 0 \end{cases}$$

Thuật toán SVM (tiếp)

- Siêu phẳng phân chia hai lớp
- Có vô số siêu phẳng như vậy, lựa chọn ntn?
- Xử lý ntn nếu DL không phân chia được một cách tuyến tính?

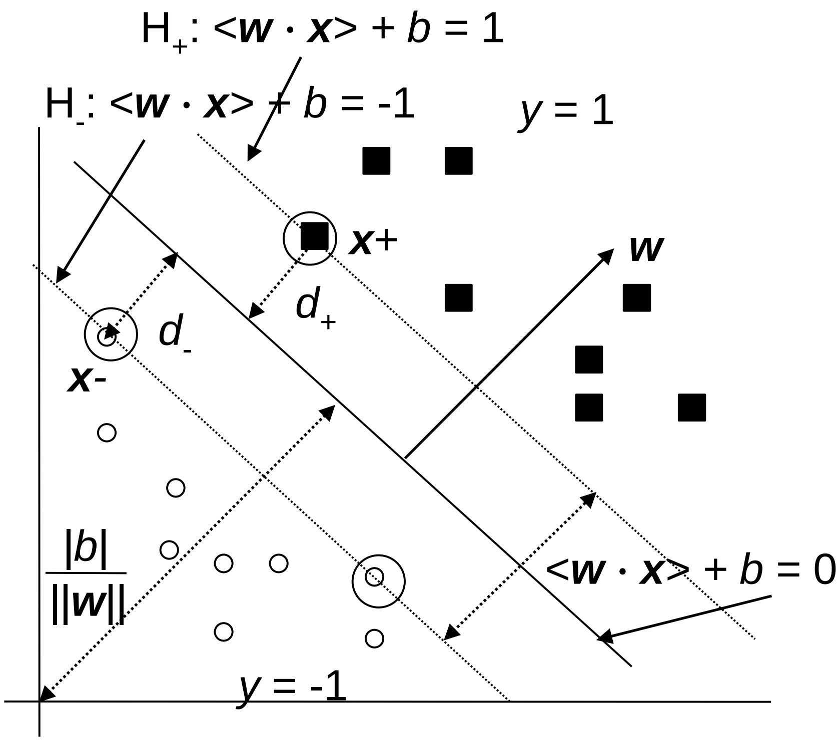




5.1 SVM tuyến tính: DL phân chia được

- w: véc-tơ chuẩn của siêu phẳng
- SVM tìm siêu phẳng nhằm cực đại hóa biên
- Nguyên lý tối thiểu hóa rủi ro cấu trúc: Cực đại hóa biên làm cực tiểu hóa cận trên của lỗi (phân loại)

$$d_+ = d_- = \frac{1}{\|\mathbf{w}\|}$$



DL phân chia được

- Bài toán cực tiểu hóa có ràng buộc

Cực tiểu hóa: $\langle \boldsymbol{w} \cdot \boldsymbol{w} \rangle / 2$

Với ràng buộc: $y_i(\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b) \geq 1 \quad i = 1, 2, \ldots, n$

- Do hàm mục tiêu bình phương và lồi, các ràng buộc tuyến tính, ta sử dụng phương pháp nhân tử *Lagrange*

$$L_P = \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle - \sum_{i=1}^{n} \alpha_i [y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1]$$

trong đó α_i là một nhân tử *Lagrange*

DL phân chia được (tiếp)

Bài toán tổng quát:

Cực tiểu hóa: $f(\boldsymbol{x})$

Với ràng buộc: $g_i(\boldsymbol{x}) \leq b_i$ $\quad i = 1, 2, \ldots, n$

Lagrangian:

$$L_P = f(\mathbf{x}) + \sum_{i=1}^{n} \alpha_i [g_i(\mathbf{x}) - b_i)]$$

Các điều kiện *Kuhn -Tucker*:

$$\frac{\partial L_p}{\partial x_j} = 0, \; j = 1, 2, \ldots, r$$

$g_i(\boldsymbol{x}) - b_i \leq 0, \quad i = 1, 2, \ldots, n$

$\alpha_i \geq 0, \; i = 1, 2, \ldots, n$

$\alpha_i(b_i - g_i(\boldsymbol{x})) = 0, \; i = 1, 2, \ldots, n$

$\Rightarrow$

$$\frac{\partial L_P}{\partial w_j} = w_j - \sum_{i=1}^{n} y_i \alpha_i x_{ij} = 0, \quad j = 1, 2, \ldots, r$$

$$\frac{\partial L_P}{\partial b} = -\sum_{i=1}^{n} y_i \alpha_i = 0$$

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 \geq 0, \quad i = 1, 2, \ldots, n$$

$$\alpha_i \geq 0, \quad i = 1, 2, \ldots, n$$

$$\alpha_i(y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1) = 0, \quad i = 1, 2, \ldots, n$$

α_i > 0 ứng với các điểm DL gọi là ***véc-tơ hỗ trợ***

DL phân chia được (tiếp)

- Do hàm mục tiêu lồi và các ràng buộc tuyến tính, các điều kiện *Kuhn - Tucker* là **cần** và **đủ,** thay thế bài toán gốc bằng bài toán đối ngẫu (đối ngẫu *Wolfe*)

Cực đại hóa: $$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$$

Với ràng buộc: $$\sum_{i=1}^{n} y_i \alpha_i = 0$$

$$\alpha_i \geq 0, \quad i = 1, 2, ..., n.$$

Siêu phẳng: $$\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = \sum_{i \in sv} y_i \alpha_i \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b = 0$$

Phân loại ví dụ ***z***: $$sign(\langle \mathbf{w} \cdot \mathbf{z} \rangle + b) = sign\left(\sum_{i \in sv} y_i \alpha_i \langle \mathbf{x}_i \cdot \mathbf{z} \rangle + b \right)$$

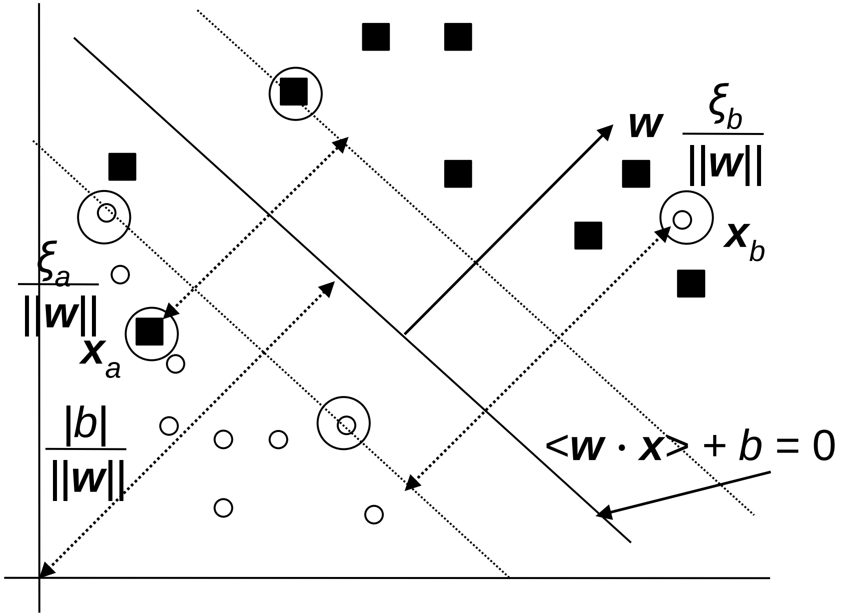
(*sv*: tập các véc-tơ hỗ trợ)

5.2 SVM tuyến tính: DL không phân chia được

Cực tiểu hóa: $$\frac{\langle \mathbf{w}\cdot\mathbf{w}\rangle}{2}+C\sum_{i=1}^{n}\xi_i$$

Với ràng buộc: $y_i(\langle \boldsymbol{w}\cdot\boldsymbol{x}_i\rangle + b) \geq 1 - \xi_i,\ i = 1, 2, \ldots, n$

$\xi_i \geq 0$, i = 1, 2, …, n



Lagrangian:

$$L_P=\frac{1}{2}\langle \mathbf{w}\cdot\mathbf{w}\rangle + C\sum_{i=1}^{n}\xi_i-\sum_{i=1}^{n}\alpha_i[y_i(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle+b)-1+\xi_i]-\sum_{i=1}^{n}\mu_i\xi_i$$

DL không phân chia được

Các điều kiện *Kuhn – Tucker*:

$$\frac{\partial L_P}{\partial w_j} = w_j - \sum_{i=1}^{n} y_i \alpha_i x_{ij} = 0, \quad j = 1, 2, ..., r$$

$$\frac{\partial L_P}{\partial b} = -\sum_{i=1}^{n} y_i \alpha_i = 0$$

$$\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \mu_i = 0, \quad i = 1, 2, ..., n$$

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 + \xi_i \geq 0, \quad i = 1, 2, ..., n$$

$$\xi_i \geq 0, \quad i = 1, 2, ..., n$$

$$\alpha_i \geq 0, \quad i = 1, 2, ..., n$$

$$\mu_i \geq 0, \quad i = 1, 2, ..., n$$

$$\alpha_i (y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 + \xi_i) = 0, \quad i = 1, 2, ..., n$$

$$\mu_i \xi_i = 0, \quad i = 1, 2, ..., n$$

Bài toán đối ngẫu:

Cực đại hóa:

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$$

Với ràng buộc:

$$\sum_{i=1}^{n} y_i \alpha_i = 0$$

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, ..., n$$

$$b = \frac{1}{y_i} - \sum_{i=1}^{n} y_i \alpha_i \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$$

Siêu phẳng:

$$\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = \sum_{i=1}^{n} y_i \alpha_i \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b = 0.$$

Nhận xét:

$\alpha_i = 0 \rightarrow y_i(<\boldsymbol{w} \cdot \boldsymbol{x}> + b) \geq 1$ và $\xi_i = 0$

$0 < \alpha_i < C \rightarrow y_i(<\boldsymbol{w} \cdot \boldsymbol{x}> + b) = 1$ và $\xi_i = 0$

$\alpha_i = C \rightarrow y_i(<\boldsymbol{w} \cdot \boldsymbol{x}> + b) \leq 1$ và $\xi_i \geq 0$

5.3 SVM phi tuyến: Hàm kernel

$\{(\Phi(\mathbf{x_1}), y_1), (\Phi(\mathbf{x_2}), y_2), \ldots, (\Phi(\mathbf{x_n}), y_n)\}$

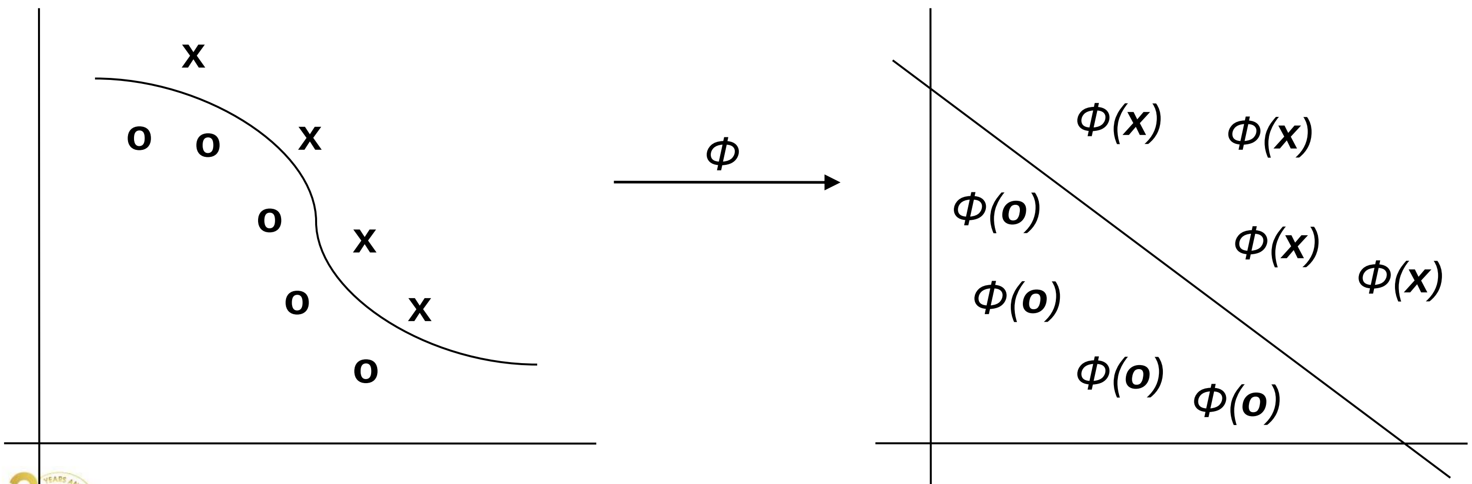
$\Phi: X \to F$

$\mathbf{x} \mapsto \Phi(\mathbf{x})$

Cực tiểu hóa: $\dfrac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2} + C \sum_{i=1}^{n} \xi_i$

Với ràng buộc: $y_i(<\mathbf{w} \cdot \Phi(\mathbf{x}_i)> + b) \geq 1 - \xi_i, i = 1, 2, \ldots, n$

$\xi_i \geq 0$, i = 1, 2, …, n



Không gian đầu vào

Không gian đặc trưng *F*

Hàm kernel

Bài toán đối ngẫu:

Cực đại hóa: $$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \rangle$$

Với ràng buộc: $$\sum_{i=1}^{n} y_i \alpha_i = 0$$

$$0 \le \alpha_i \le C, \quad i = 1, 2, \ldots, n$$

Phân loại:

$$\sum_{i=1}^{n} y_i \alpha_i \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) \rangle + b$$

VD: $(x_1, x_2) \mapsto (x_1^2, x_2^2, 2^{1/2} x_1 x_2)$
$(2,3) \mapsto (4, 9, 8.5)$

Hàm kernel (tiếp)

- Hàm kernel: phép nhân véc-tơ trên không gian đầu vào tương ứng với một phép nhân véc-tơ trên không gian đặc trưng nhiều chiều

$$K(\boldsymbol{x}, \boldsymbol{z}) = \langle \Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{z}) \rangle$$

- Kernel đa thức:

$$K(\boldsymbol{x}, \boldsymbol{z}) = \langle \boldsymbol{x} \cdot \boldsymbol{z} \rangle^{d}$$

VD:

$\boldsymbol{x} = (x_1, x_2)$, $\boldsymbol{z} = (z_1, z_2)$

$$\begin{aligned}
\langle \boldsymbol{x} \cdot \boldsymbol{z} \rangle^2 &= (x_1 z_1 + x_2 z_2)^2 \\
&= x_1^2 z_1^2 + 2 x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\
&= \langle (x_1^2, x_2^2, 2^{1/2} x_1 x_2) \cdot (z_1^2, z_2^2, 2^{1/2} z_1 z_2) \rangle \\
&= \langle \Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{z}) \rangle
\end{aligned}$$

Hàm kernel (tiếp)

- Không gian đặc trưng của kernel đa thức bậc *d* có C_d^{r+d-1p} chiều
- Định lý Mercer xác định các hàm kernel
- Kernel đa thức:

$$K(\boldsymbol{x}, \boldsymbol{z}) = (<\boldsymbol{x} \cdot \boldsymbol{z}> + \theta)^{d}$$

- Gaussian RBF:

$$K(\mathbf{x},\mathbf{z}) = e^{-\|\mathbf{x}-\mathbf{z}\|^2/2\sigma}$$

Ứng dụng của SVM

- Để SVM làm việc với thuộc tính rời rạc, có thể chuyển sang dạng nhị phân
- Để phân loại đa lớp, có thể sử dụng các chiến lược như one-vs-all hay one-vs-one
- Siêu phẳng gây khó hiểu cho người dùng, do đó SVM thường được sử dụng trong các ứng dụng không đòi hỏi tính giải thích

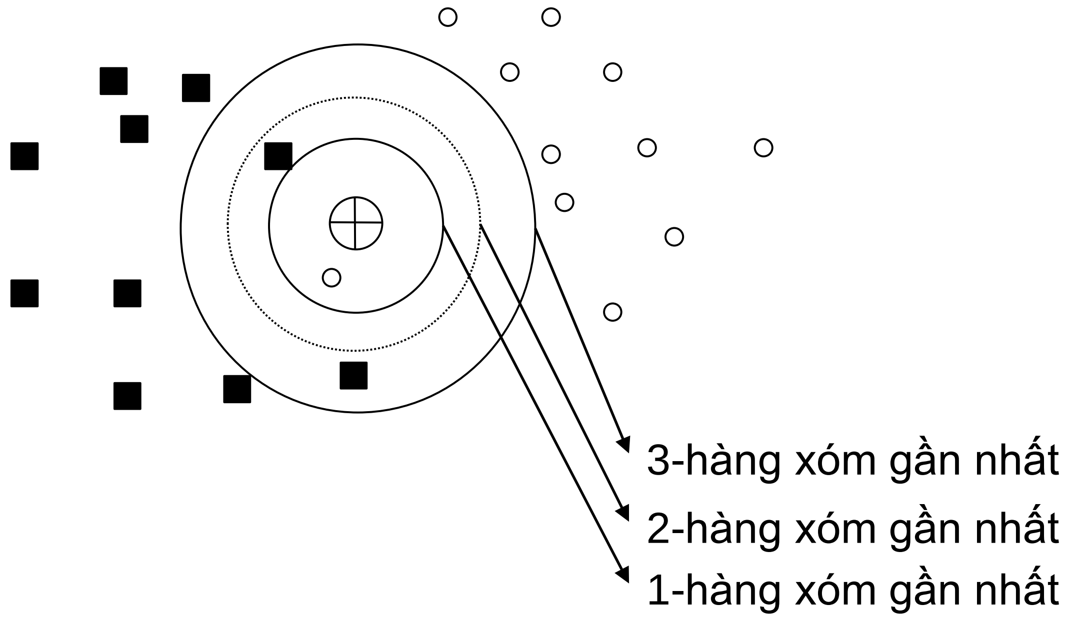
6. Thuật toán kNN

- Cho tập DL “huấn luyện” D, với một ví dụ kiểm thử d

1. Tính toán độ tương đồng (khoảng cách) của d với tất cả các ví dụ huấn luyện
2. Xác định k ví dụ gần nhất dựa trên độ tương đồng
3. Phân loại d dựa trên nhãn của k ví dụ trên

- Nhược điểm:
  - Thời gian phân loại lâu
  - Không có tính giải thích

Thuật toán KNN



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!