BÀI THỰC HÀNH – CSTTNT
BUỔI 6- NAÏVE BAYES CLASSIFIER

Phân lớp Bayes là giải thuật học dựa trên định lý Bayes, dùng để giải quyết các vấn đề phân lớp, gom nhóm, etc, được ứng dụng thành công trong phân tích dữ liệu, phân loại text, spam, …

Giải thuật Naïve Bayes được xem là "ngây thơ" vì xem các thuộc tính (biến) có độ quan trọng như nhau và các thuộc tính (biến) độc lập có điều kiện khi được cho lớp/nhãn.

Naïve Bayes dựa trên lý thuyết xác suất để thực hiện, trong đó có 2 loại xác suất cần quan tâm:

1. **Xác suất tiên nghiệm (prior probability)**: hay xác suất **VÔ** điều kiện: là xác suất của một sự kiện không có tri thức bổ sung cho sự có mặt hay vắng mặt của nó.
2. **Xác suất hậu nghiệm (posterior probability)**: hay xác suất **CÓ** điều kiện: là xác suất của một sự kiện khí biết trước một hay nhiều sự kiện khác

$$P(e_1|e_2) = \frac{P(e_1 \wedge e_2)}{P(e_2)}$$

Ví dụ: P(cúm) = 0.001, P(sốt) = 0.003; P(cúm ∧ sốt) = 0.000003
nhưng cúm và sốt là các sự kiện không độc lập
các chuyên gia cho biết: P(sốt | cúm) = 0.9

3. Định lý Bayes

P(h|e) là xác suất khẳng định *giả thuyết* h đúng cho trước *bằng chứng* e.

$$P(h|e) = \frac{P(e|h) * P(h)}{P(e)} \quad <= \text{luật Bayes}$$

Trong đó: P(h) là xác suất tiên nghiệm của h (xác suất của sự kiện h trước khi bằng chứng e)

P(h|e) là xác xuất hậu nghiệm (xác suất của sự kiện h sau khi khi có bằng chứng e)

Ví dụ: Bằng chứng (triệu chứng): bệnh nhân bị sốt
Giả thuyết (bệnh): bệnh nhân bị cảm cúm

$$P(\text{cúm}|\text{sốt}) = \frac{P(\text{cúm}) * P(\text{sốt}|\text{cúm})}{P(\text{sốt})} = \frac{0.001 * 0.9}{0.003} = 0.3$$

4. Luật Bayes

- Học phân lớp dữ liệu, trong đó bằng chứng E = dữ liệu và sự kiện H = giá trị lớp của sự kiện

$$P[H \mid E] = \frac{P[E_1|H]\,P[E_2|H]\ldots P[E_n|H]}{P[E]}$$

Ví dụ:

Cho dữ liệu weather, dựa trên các thuộc tính ra quyết định (play/no)

Dữ liệu training

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

Tính xác suất

Outlook			Temperature			Humidity			Windy			Play	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

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

Cách tính

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

← ***Evidence E***

$$\mathrm{P}[yes \mid E] = \mathrm{P}\,[Outlook = Sunny \mid yes] \times \mathrm{P}\,[Temperature = Cool \mid yes] \times \mathrm{P}\,[Humidity = High \mid yes] \times \mathrm{P}\,[Windy = True \mid yes] \times \frac{\mathrm{P}\,[yes]}{\mathrm{P}\,[E]}$$

$$= \frac{\frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14}}{\mathrm{P}\,[E]}$$

xác suất của lớp "yes"

Kiểm thử dữ liệu

Outlook			Temperature			Humidity			Windy			Play	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

- quyết định (play=yes/no)

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

Likelihood(yes) = 2/9 x 3/9 x 3/9 x 3/9 x 9/14 = 0.0053

Likelihood(no) = 3/5 x 1/5 x 4/5 x 3/5 x 5/14 = 0.0206

Xác suất :

P(yes) = 0.0053 / (0.0053 + 0.0206) = 0.205

P(no) = 0.0206 / (0.0053 + 0.0206) = 0.795

5. Trường hợp xác suất = 0

Ví dụ : giá trị thuộc tính không xuất hiện trong tất cả các lớp như ("Hunidity = high" của lớp "Yes") → P[Humidity=High | Yes] = 0

Dùng ***Laplace estimator → xác suất không bao giờ có giá trị 0***

6. Thuộc tính nhiễu

Outlook	Temp.	Humidity	Windy	Play
?	Cool	High	True	?

Likelihood(yes) = 3/9 × 3/9 × 3/9 × 9/14 = 0.0238

Likelihood(no) = 1/5 × 4/5 × 3/5 × 5/14 = 0.0343

P(yes) = 0.0238 / (0.0238 + 0.0343) = 41

P(no) = 0.0343 / (0.0238 + 0.0343) = 59

7. Các phân phối

a. Gaussian Naïve Bayes

- Dùng cho dữ liệu liên tục, hàm mật độ xác suất được tính như sau:

mean μ

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

standard deviation σ

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \mu)^2$$

hàm mật độ xác suất *f(x)*

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Ví dụ:

Outlook	Temperature	Humidity	Windy	Play
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	78	false	yes
rain	70	96	false	yes
rain	68	80	false	yes
rain	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rain	75	80	false	yes
sunny	75	70	true	yes
overcast	72	90	true	yes
overcast	81	75	false	yes
rain	71	80	true	no

Outlook			Temperature			Humidity			Windy			Play	
	yes	*no*		*yes*	*no*		*yes*	*no*		*yes*	*no*	*yes*	*no*
sunny	2	3		83	85		86	85	false	6	2	9	5
overcast	4	0		70	80		96	90	true	3	3		
rainy	3	2		68	65		80	70					
				64	72		65	95					
				69	71		70	91					
				75			80						
				75			70						
				72			90						
				81			75						
sunny	2/9	3/5	*mean*	73	74.6	*mean*	79.1	86.2	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	*std. dev.*	6.2	7.9	*std. dev.*	10.2	9.7	true	3/9	3/5		
rainy	3/9	2/5											

- ví dụ : $f(temperature = 66 \mid yes) = \frac{1}{\sqrt{2\pi}6.2} e^{-\frac{(66-73)^2}{2*6.2^2}} = 0.0340$

phân lớp

Outlook	Temp.	Humidity	Windy	Play
Sunny	66	90	true	?

Likelihood(yes) = 2/9 × 0.0340 × 0.0221 × 3/9 × 9/14 = 0.000036
Likelihood(no) = 3/5 × 0.0291 × 0.0380 × 3/5 × 5/14 = 0.000136
P(yes) = 0.000036 / (0.000036 + 0. 000136) = 20.9
P(no) = 0.000136 / (0.000036 + 0. 000136) = 79.1

b. Multinomial Naive Bayes

Mô hình này chủ yếu được sử dụng trong phân loại văn bản mà feature vectors được tính bằng Bags of Words. Lúc này, mỗi văn bản được biểu diễn bởi một vector có độ dài d chính là số từ trong từ điển. Giá trị của thành phần thứ i trong mỗi vector chính là số lần từ thứ i xuất hiện trong văn bản đó.

Khi đó, $p(x_i|c)$ tỉ lệ với tần suất từ thứ i (hay feature thứ i cho trường hợp tổng quát) xuất hiện trong các văn bản của class c. Giá trị này có thể được tính bằng cách:

$$\lambda_{ci} = p(x_i|c) = \frac{N_{ci}}{N_c}$$

Trong đó:

- N_{ci} :là tổng số lần từ thứ i xuất hiện trong các văn bản của class c, nó được tính là tổng của tất cả các thành phần thứ i của các feature vectors ứng với class c
- N_c là tổng số từ (kể cả lặp) xuất hiện trong class c. Nói cách khác, nó bằng tổng độ dài của toàn bộ các văn bản thuộc vào class c. Có thể suy ra rằng
$$N_c = \sum_{i=1}^{d} N_{ci}, \text{ từ đó } \sum_{i=1}^{d} \lambda_{ci} = 1.$$

Cách tính này có một hạn chế là nếu có một từ mới chưa bao giờ xuất hiện trong class c thì biểu thức sẽ bằng 0, điều này dẫn đến xác suất bằng 0 bất kể các giá trị còn lại có lớn thế nào. Việc này sẽ dẫn đến kết quả không chính xác. Để giải quyết việc này, một kỹ thuật được gọi là *Laplace smoothing* được áp dụng:

$$\hat{\lambda}_{ci} = \frac{N_{ci} + \alpha}{N_c + d\alpha}$$

Với α là một số dương, thường bằng 1, để tránh trường hợp tử số bằng 0. Mẫu số được cộng với dα để đảm bảo tổng xác suất

$$\sum_{i=1}^{d} \hat{\lambda}_{ci} = 1.$$

Như vậy, mỗi class c sẽ được mô tả bởi bộ các số dương có tổng bằng 1:

$$\hat{\lambda}_c = \{\hat{\lambda}_{c1}, \ldots, \hat{\lambda}_{cd}\}.$$

3. Bernoulli Naive Bayes

Mô hình này được áp dụng cho các loại dữ liệu mà mỗi thành phần là một giá trị binary - bằng 0 hoặc 1. Ví dụ: cũng với loại văn bản nhưng thay vì đếm tổng số lần xuất hiện của 1 từ trong văn bản, ta chỉ cần quan tâm từ đó có xuất hiện hay không.

Khi đó, $p(x_i|c)$ được tính bằng:

$$p(x_i|c) = p(i|c)^{x_i}(1 - p(i|c)^{1-x_i}$$

với $p(i|c)$ có thể được hiểu là xác suất từ thứ i xuất hiện trong các văn bản của class c.

(tìm hiểu thêm ở trang web: https://nlp.stanford.edu/IR-book/html/htmledition/the-bernoulli-model-1.html)

Ví dụ:

Giả sử trong tập training có các văn bản **d1, d2, d3, d4** như trong bảng dưới đây. Mỗi văn bản này thuộc vào 1 trong 2 classes: **B** (*Bắc*) hoặc **N** (*Nam*). Hãy xác định class của văn bản **d5**.

	Document	Content	Class
Training	d1	hanoi pho chaolong hanoi	B
	d2	hanoi buncha pho omai	B
	d3	pho banhgio omai	B
	d4	saigon hutiu banhbo pho	N
Test	d5	hanoi hanoi buncha hutiu	?

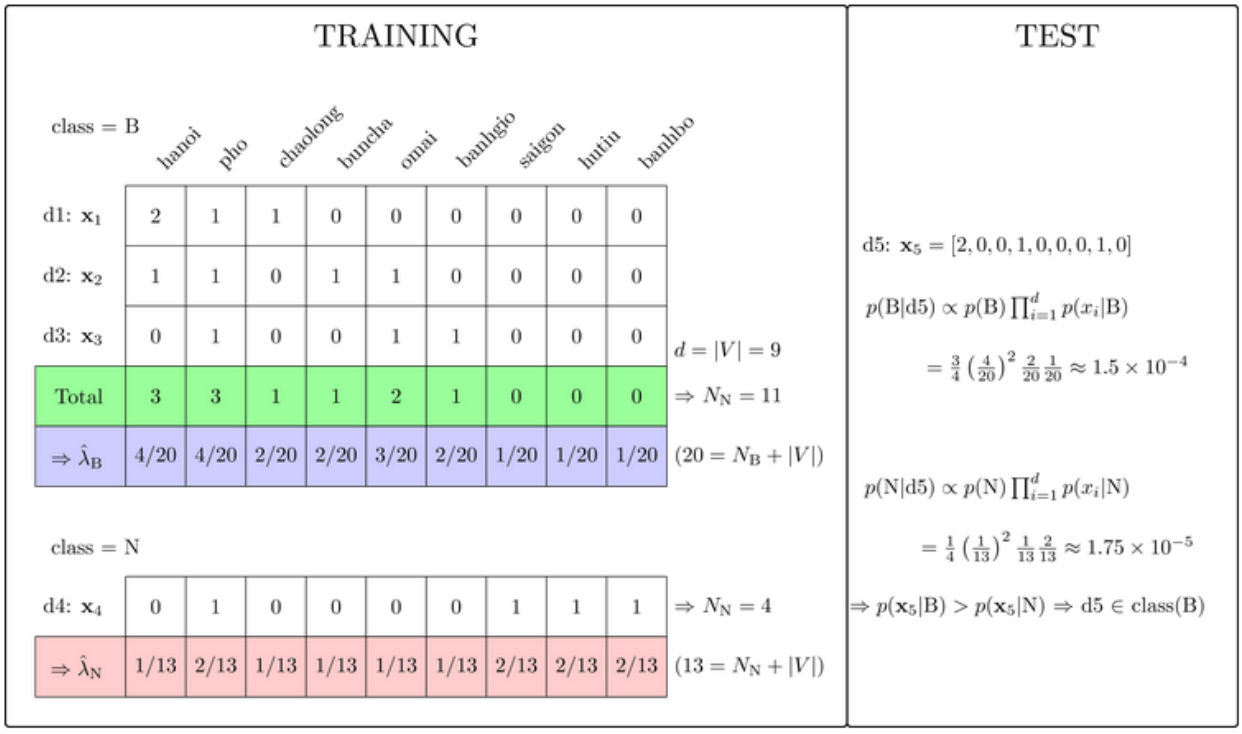
Ta có thể dự đoán rằng **d5** thuộc class *Bắc*.

Bài toán này có thể được giải quyết bởi hai mô hình: Multinomial Naive Bayes và Bernoulli Naive Bayes. Nhận thấy rằng ở đây có 2 class B và N, ta cần đi tìm p(B) và p(N). à dựa trên tần số xuất hiện của mỗi class trong tập training. Ta sẽ có:

$$p(\mathrm{B}) = \frac{3}{4}, \quad p(\mathrm{N}) = \frac{1}{4}$$

Tập hợp toàn bộ các từ trong văn bản, hay còn gọi là từ điển, là: V={hanoi, pho, chaolong, buncha, omai, banhgio, saigon, hutiu, banhbo}. Tổng cộng số phần tử trong từ điển là |V|=9. Hình dưới đây minh hoạ quá trình Training và Test cho bài toán này khi sử dụng Multinomial Naive Bayes, trong đó có sử dụng Laplace smoothing với α=1

.



Minh hoạ Multinomial Naive Bayes.

Chú ý, hai giá trị tìm được 1.5×10^{-4} và 1.75×10^{-5} không phải là hai xác suất cần tìm mà chỉ là hai đại lượng **tỉ lệ thuận** với hai xác suất đó. Để tính cụ thể, ta có thể làm như sau:

$$p(\mathrm{B}|\mathrm{d5}) = \frac{1.5 \times 10^{-4}}{1.5 \times 10^{-4} + 1.75 \times 10^{-5}} \approx 0.8955, \quad p(\mathrm{N}|\mathrm{d5}) = 1 - p(\mathrm{B}|\mathrm{d5}) \approx 0.1045$$

Bài tập 1

```
from __future__ import print_function
from sklearn.naive_bayes import MultinomialNB
import numpy as np

# train data
d1 = [2, 1, 1, 0, 0, 0, 0, 0, 0]
d2 = [1, 1, 0, 1, 1, 0, 0, 0, 0]
d3 = [0, 1, 0, 0, 1, 1, 0, 0, 0]
d4 = [0, 1, 0, 0, 0, 0, 1, 1, 1]

train_data = np.array([d1, d2, d3, d4])
label = np.array(['B', 'B', 'B', 'N'])

# test data
d5 = np.array([[2, 0, 0, 1, 0, 0, 0, 1, 0]])
d6 = np.array([[0, 1, 0, 0, 0, 0, 0, 1, 1]])

## call MultinomialNB
clf = MultinomialNB()
# training
clf.fit(train_data, label)

# test
print('Predicting class of d5:', str(clf.predict(d5)[0]))
print('Probability of d6 in each class:', clf.predict_proba(d6))
```

Bài tập 2

```
from __future__ import print_function
from sklearn.naive_bayes import BernoulliNB
import numpy as np

# train data
d1 = [1, 1, 1, 0, 0, 0, 0, 0, 0]
d2 = [1, 1, 0, 1, 1, 0, 0, 0, 0]
d3 = [0, 1, 0, 0, 1, 1, 0, 0, 0]
d4 = [0, 1, 0, 0, 0, 0, 1, 1, 1]

train_data = np.array([d1, d2, d3, d4])
label = np.array(['B', 'B', 'B', 'N']) # 0 - B, 1 - N

# test data
d5 = np.array([[1, 0, 0, 1, 0, 0, 0, 1, 0]])
d6 = np.array([[0, 1, 0, 0, 0, 0, 0, 1, 1]])

## call BernoulliNB
clf = BernoulliNB()
# training
clf.fit(train_data, label)
```

```
# test
print('Predicting class of d5:', str(clf.predict(d5)[0]))
print('Probability of d6 in each class:', clf.predict_proba(d6))
```

Bài tập 3: Dựa vào những lý thuyết + ví dụ đã học hãy hiện thực giải thuật Naïve Bayes để giải quyết bài toán sau (sử dụng Naïve Bayes dạng MultinomialNB)

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

Dự đoán:

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?