CHƯƠNG 4:
MỘT SỐ THUẬT TOÁN MÁY HỌC (P1)

Khoa Khoa học và Kỹ thuật thông tin
Bộ môn Khoa học dữ liệu

NỘI DUNG CHÍNH

Naive Bayes

Dẫn nhập

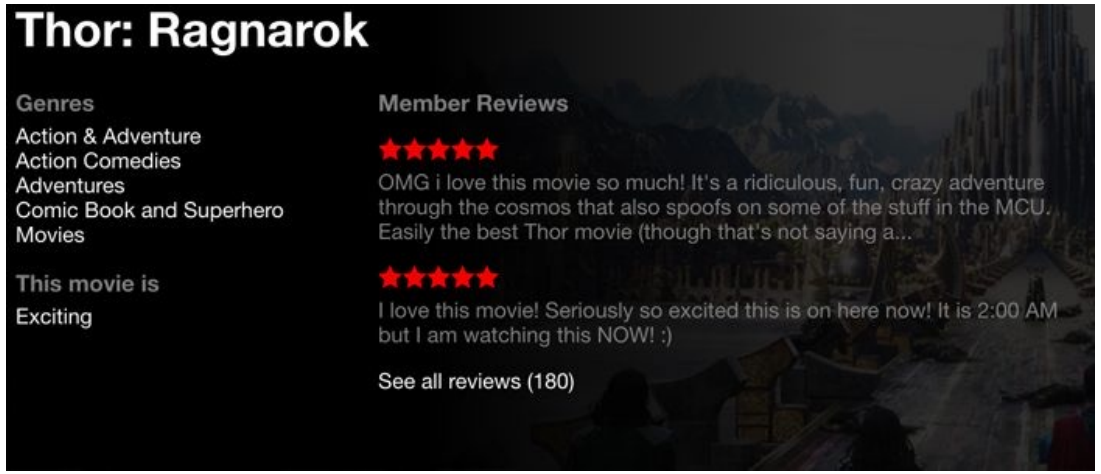
Subject: **Important notice!**
From: Stanford University <newsforum@stanford.edu>
Date: October 28, 2011 12:34:16 PM PDT
To: undisclosed-recipients:;

Greats News!

You can now access the latest news by using the link below to login to Stanford University News Forum.

http://www.123contactform.com/contact-form-StanfordNew1-236335.html

Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information
about the new services.



Hỏi email trên có phải là spam hay không ?
(Bài toán phân loại email rác)

Những bình luận liên quan đến phim
là tích cực hay là tiêu cực ?

Bài toán phân lớp văn bản

- Đầu vào:
  + Một văn bản *d*.
  + Một tập cố định các lớp C = $\{c_1, c_2, \ldots, c_n\}$
- Đầu ra: Một lớp được dự đoán $c_i \in C$ (i = 1...n)

➔Bài toán này thuộc dạng bài toán phân lớp (classification).

➔Thuộc loại tác vụ học có giám sát (supervised learning).

Các thuật toán phân loại có giám sát

- Naïve Bayes.
- k-Nearest Neighbors.
- Logistic regression.
- Neural networks.
- ***Transformer models.***

Naive Bayes

Naive bayes

- Xét bài toán phân lớp gồm tập nhãn C = $\{c_1, c_2, \ldots, c_n\}$ và một điểm dữ liệu x. Hãy tìm ra xác suất để điểm dữ liệu này rơi vào lớp c. Mô tả như sau:

$$p(y = c \mid x)$$

- Mục tiêu: tính xác suất để tìm ra giá trị đầu ra c.

$$\hat{c} = \boldsymbol{argmax_{c \in C} p(c \mid x)}$$

- Theo luật Bayes, ta được:

$$\hat{c} = \operatorname{argmax}_{c \in C} p(c \mid x) = \operatorname{argmax}_{c \in C} \frac{p(x|c)p(c)}{p(x)} = \boldsymbol{argmax_{c \in C}\, p(x|c)p(c)}$$

Huấn luyện mô hình

$$\hat{c} = argmax_{c \in C}\, p(x|c)\, p(c)$$

Likelihood

Prior

Tìm p(c): xác suất tiên nghiệm, tính toán khả năng xảy ra **biến cố độc lập** c.

Để **tìm p(x|c)** với dữ liệu có **n chiều (n thuộc tính)**, chúng ta viết lại công thức như sau:

$$p(x \mid c) = p(x_1 \dots x_n \mid c)$$

➔ Ý nghĩa: Tìm xác suất để các sự kiện $x_1, x_2, \dots x_n$ xảy ra đồng thời. Giả sử các sự kiện $x_1, x_2, \dots x_n$ tuân theo 1 **phân phối xác suất** nhất định theo bộ tham số θ. Như vậy, mục tiêu của mô hình là phải tìm ra θ sao cho:

$$\hat{\theta} = argmax_{\theta}\, p(x_1 \dots x_n|\theta)$$

maximum likelihood

Tìm θ và **p(c)** dựa trên dữ liệu huấn luyện ➔ huấn luyện mô hình.

Giả thuyết Naive

- Giả thuyết độc lập thống kê: các thuộc tính trong tập dữ liệu là **độc lập** với nhau. Hay nói cách khác, $\mathbf{x_1, x_2, \ldots x_n}$ độc lập nhau.
- Khi đó, xác suất các sự kiện xảy ra đồng thời được tính như sau:

$$p(x_1 \ldots x_n \mid c) = \prod_{i=1}^{d} p(x_i \mid c)$$

Với d là số lượng dữ liệu.

➔công thức *nhân xác suất*.

- Giả thuyết các thuộc tính của dữ liệu độc lập nhau còn được gọi là giả thuyết về sự ngây thơ (Naive). Chính vì vậy, nó đơn giản hoá các tính toán đi rất nhiều.

Dự đoán dữ liệu đầu vào

- Với một điểm dữ liệu đầu vào x, nhãn của dữ liệu được dự đoán như sau:

$$\mathrm{c} = \mathrm{argmax}_{\mathrm{c}\in\mathrm{C}}\ \mathrm{p(c)} \prod_{i=1}^{\mathrm{d}} \boldsymbol{p(x_i \mid c)}$$

- Khi số lượng dữ liệu (d) lớn, và giá trị của p nhỏ, quá trình tính toán sẽ xảy ra sai số. Do đó, người ta thường lấy logarit cho giá trị tính toán như sau:

$$\mathrm{c} = \mathrm{argmax}_{\mathrm{c}\in\mathrm{C}} \log\left(\mathrm{p(c)} \prod_{\boldsymbol{i=1}}^{\mathrm{d}} \boldsymbol{p(x_i \mid c)}\right) = \mathbf{argmax}_{\boldsymbol{c}\in\boldsymbol{C}}\left(\boldsymbol{log(p(c))} + \sum_{\boldsymbol{i=1}}^{\boldsymbol{d}} \boldsymbol{log(p(x_i \mid c))}\right)$$

Tính toán cho giá trị likelihood$p(x_i \mid c)$

- Multinomial Naive Bayes: sử dụng cho dữ liệu mà giá trị của các thuộc tính là giá trị rời rạc.
- Gaussian Naive Bayes: sử dụng cho dữ liệu mà giá trị của các thuộc tính là liên tục.
- Bernoulli Naive Bayes: sử dụng cho dữ liệu mà giá trị của các thuộc tính là nhị phân.

Multinomial Naive Bayes

$$p(x_i \mid c) = \frac{\mathrm{Nc_i}}{\mathrm{N_c}}$$

Nc_i là tổng giá trị thành phần thứ i xuất hiện trong class c.

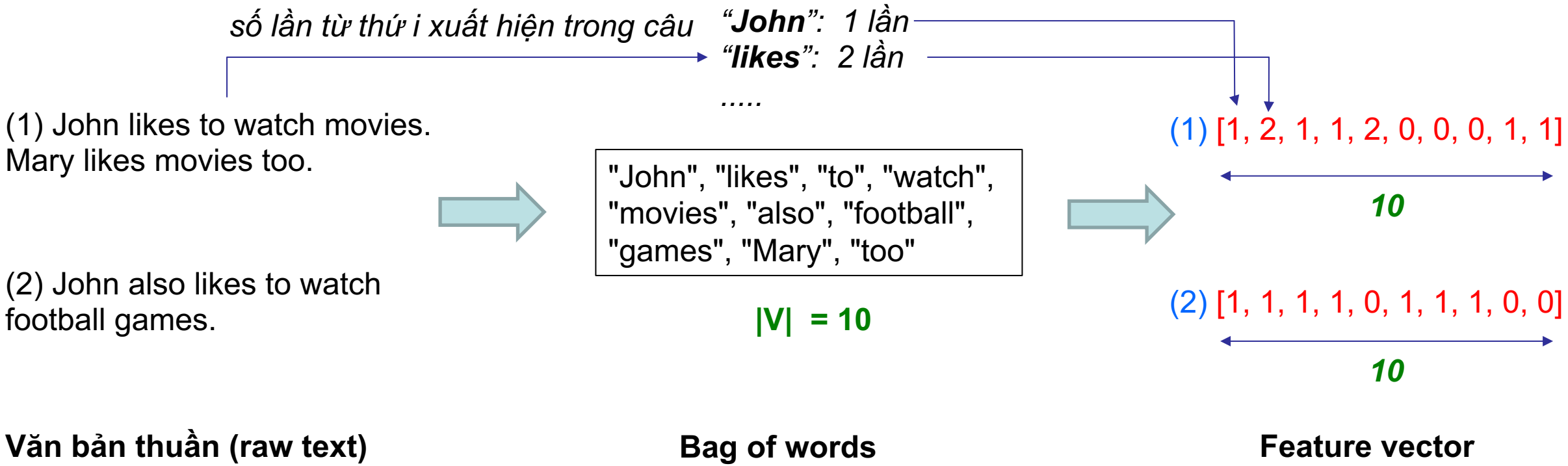
Nc là tổng các giá trị thành phần xuất hiện trong class c.

Đối với bài toán phân lớp văn bản:

+ Nc_i là **tổng số lần từ thứ i xuất hiện** trong các văn bản của class c.
+ Nc là **tổng số từ (kể cả lặp)** xuất hiện trong class c

Chuyển văn bản thành dạng thành phần ➔ sử dụng khái niệm **Bag of word (túi từ)**.

Bag of words (BoW)



Tách văn bản thành "Từ" ➔ bài toán tách từ (word segmentation hoặc là word tokenization)

Gaussian Naive Bayes

$$p(x_i \mid c) = p\left(x_i \mid \mu_{c_i}, \sigma^2_{c_i}\right) = \frac{1}{\sqrt{2\pi\sigma^2_{c_i}}} e^{\frac{-\left(x_i - \mu_{c_i}\right)^2}{2\sigma^2_{c_i}}}$$

Trong đó:

μ_{c_i} là kỳ vọng.

$\sigma^2_{c_i}$ là phương sai.

Bernoulli Naive Bayes

$$p(x_i \mid c) = p(i \mid c)^{x_i} \left(1 - p(i \mid c)^{1 - x_i}\right)$$

$p(i \mid c)$ là xác suất từ i xuất hiện trong các văn bản của lớp c.

$p(i \mid c)^{x_i} \left(1 - p(i \mid c)^{1 - x_i}\right)$ có dạng **phân phối nhị thức**.

Naive Bayes cho phân lớp văn bản

$$p(x_i \mid c) = \frac{Nc_i}{N_c} \text{ (*)}$$

- Nc_i là **tổng số lần từ thứ i xuất hiện** trong các văn bản của class c.
- Nc là **tổng số từ (kể cả lặp)** xuất hiện trong class c.
- Viết lại công thức (*): đếm số từ xuất hiện trong văn bản:

$$p(w_i \mid c) = \frac{count(w_i, c)}{\sum_{w \in V} count(w, c)}$$

Với: w là một từ, V là tập từ vựng.

Laplace smoothing

- Nếu từ w_i không xuất hiện trong văn bản thì: $\mathrm{count}(w_i, c) = 0$ ➔ $p(w_i \mid c) = 0$ trong mọi trường hợp. Như vậy, kết quả dự đoán không còn chính xác.
- Khắc phục: Thêm vào một lượng α. Công thức trở thành:

$$p(w_i \mid c) = \frac{\mathrm{count}(w_i, c) + \alpha}{\sum_{w \in V} \mathrm{count}(w, c) + \alpha}$$

Với $\alpha = 1$ thì $p(w_i \mid c)$ được tính như sau:

$$p(w_i \mid c) = \frac{\mathrm{count}(w_i, c) + 1}{\sum_{w \in V} \mathrm{count}(w, c) + 1} = \frac{count(w_i, c) + 1}{(\sum_{w \in V} count(w, c)) + |V|}$$

Ví dụ

– Cho bộ dữ liệu huấn luyện và dữ liệu test như bảng sau:

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

Hãy dự đoán nhãn cho dữ liệu test

Tập từ vựng

- Có 2 nhãn: + và –
- Tập từ vựng: V = {*just, plain, boring, entirely, predictable, and, lacks, energy, no, surprises, very, few, laughs, powerful, the, most, fun, film, of, summer*}.
- |V| = 20.
- |V-| = 14.
- |V+| = 9.

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

Dự đoán nhãn *Cat* cho dữ liệu test:

- Tính p(c) từ dữ liệu huấn luyện.
- Tính $p(x_i \mid c)$ cho từng từ trong câu dựa vào dữ liệu huấn luyện.

Tính p(c)

- Tổng số dữ liệu huấn luyện: 5
- Số lớp +: 2
- Số lớp -: 3

p(-) = 3/5 = 0.6.

p(+) = 2/5 = 0.4.

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

Tính p(x_i | c)

- Bỏ từ *with* đi.
- Xét lớp +:
  - $p(\text{predictable}|+) = \frac{0+1}{9+20} = 0.034$
  - $p(\text{no}|+) = \frac{0+1}{9+20} = 0.034$
  - $p(\text{fun}|+) = \frac{1+1}{9+20} = 0.068$
- Xét lớp -:
  - $p(\text{predictable}|-) = \frac{1+1}{14+20} = 0.058$
  - $p(\text{no}|-) = \frac{1+1}{14+20} = 0.058$
  - $p(\text{fun}|-) = \frac{0+1}{14+20} = 0.029$

$$p(w_i \mid c) = \frac{count(w_i, c) + 1}{(\sum_{w \in V} count(w, c)) + |V|}$$

|V| = 20.
|V-| = 14.
|V+| = 9.

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

Tổng hợp

$$c = \text{argmax}_{c \in C}\, p(c) \prod_{i=1}^{d} p(x_i \mid c)$$

- Xét lớp +: p(+) = 0.4
  - $p(\text{predictable}|+) = \frac{0+1}{9+20} = 0.034$
  - $p(\text{no}|+) = \frac{0+1}{9+20} = 0.034$
  - $p(\text{fun}|+) = \frac{1+1}{9+20} = 0.068$

$$p(S|+) = 0.4 * 0.034 * 0.034 * 0.068 = 3.2 * 10^{-5}$$

- Xét lớp -: p(-) = 0.6
  - $p(\text{predictable}|-) = \frac{1+1}{14+20} = 0.058$
  - $p(\text{no}|-) = \frac{1+1}{14+20} = 0.058$
  - $p(\text{fun}|-) = \frac{0+1}{14+20} = 0.029$

$$p(S|-) = 0.6 * 0.058 * 0.058 * 0.029 = 5.8 * 10^{-5}$$

Chọn lớp -

Ví dụ

– Hiện thực Naive Bayes bằng *python* theo từng bước như ví dụ trên **(không dùng thư viện)**. Dữ liệu theo đoạn code như sau:

```
train = [
    ["just plain boring", "-"],
    ["entirely predictable and lacks energy", "-"],
    ["no surprises and very few laughs", "-"],
    ["very powerful", "+"],
    ["the most fun film of the summer", "+"]
]

test = "predictable with no fun"
```

Hiện thực bằng thư viện sklearn

– Chuẩn bị dữ liệu: X_train, y_train và X_test

```
1. X_train = []
2. y_train = []
3. X_test = [test]

4. for t in train:
5.    X_train.append(t[0])
6.    y_train.append(t[1])
```

Mã hoá dữ liệu

- Chuyển dữ liệu về dạng số: các feature vector:
  + X_train, X_test: chuyển về vector tần suất xuất hiện bằng thư viện *CountVectorizer*
  + y_train: Sử dụng thư viện *LabelEncoder*

```
1. from sklearn.feature_extraction.text import CountVectorizer
2. from sklearn.preprocessing import LabelEncoder
3. vectorizer = CountVectorizer()
4. le = LabelEncoder()
5. vectorizer.fit(X_train)
6. le.fit(y_train)
7. X_train_encoded = vectorizer.transform(X_train)
8. y_train_encoded = le.transform(y_train)
9. X_test_encoded = vectorizer.transform(X_test)
```

Huấn luyện mô hình

– Sử dụng thư viện MultinomialNB trong sklearn.

```
1. from sklearn.naive_bayes import MultinomialNB
2. model = MultinomialNB()
3. model.fit(X_train_encoded,y_train_encoded)
```

– Dự đoán nhãn:

```
1. y_pred = model.predict(X_test_encoded)
2. print("The class of \"{}\" is {}".format(test,
   le.inverse_transform(y_pred)[0]))
```

Bài tập 1

– Dự đoán nhãn cho dữ liệu test bằng thuật toán Naive bayes. Dữ liệu huấn luyện và dự đoán được cho trong bảng sau:

	Doc	Sentence	Class
Training	1	sản_phẩm A rất tốt.	pos
	2	tôi không thích sản_phẩm A.	neg
	3	màu_sắc của A thật tốt.	pos
	4	sản_phẩm A thật kinh_khủng.	neg
Test	5	sản_phẩm A khá tốt.	?
	6	màu_sắc quá kinh_khủng.	?

Bài tập 2

- Bộ dữ liệu UIT-VSFC là bộ dữ liệu dùng để phân loại cảm xúc các phản hồi của sinh viên về đào tạo.
  + Input: câu phản hồi của sinh viên.
  + Output: cảm xúc, gồm 1 trong 3 loại: tích cực (positive), tiêu cực (negative) và trung tính (neutral).

Download tại link này:

https://drive.google.com/drive/folders/1xclbjHHK58zk2X6iqbvMPS2rcy9y9E0X

Yêu cầu: Hãy đánh giá mô hình Naive Bayes

Hướng dẫn

1. *Tìm hiểu các đặc tính của bộ dữ liệu: tập train có bao nhiêu dữ liệu, tập test có bao nhiêu dữ liệu, các thuộc tính của dữ liệu.*
2. *Đọc dữ liệu: sử dụng thư viện pandas.*
3. *Chuẩn bị dữ liệu: chia ra X_train, X_test, y_train, y_test.*
4. *Mã hoá dữ liệu: sử dụng các thư viện CountVectorizer hoặc TfidfVectorizer và LabelEncoder trong sklearn.*
5. *Huấn luyện mô hình.*
6. *Đánh giá mô hình theo các độ đo (xem lại slide về Phân lớp).*

TỔNG KẾT

- Naive Bayes và KNN là các thuật toán thuộc dạng học máy có giám sát.
- Naive Bayes và KNN đều có đặc điểm là đơn giản và thời gian huấn luyện nhanh.
- Naive Bayes chỉ thích hợp đối với dữ liệu nhỏ và các thuộc tính trong dữ liệu có tính độc lập với nhau.
- KNN không tốn nhiều tài nguyên ở quá trình huấn luyện, nhưng lại tốn nhiều thời gian cho quá trình test, nhất là đối với dữ liệu có số chiều lớn (dữ liệu phức tạp).

TÀI LIỆU THAM KHẢO

1. *Vũ Hữu Tiệp, Machine Learning cơ bản – Chương K lân cận, NXB Khoa học và Kỹ thuật (2018).*
2. Jurafsky and Martin, *Speech and Language Processing (3rd) – Naive Bayes and Sentiment Classification.*