

# Hướng Dẫn Lập Trình NLP 3 - Thuật toán Perceptron

Graham Neubig  
Viện Khoa học và Công nghệ Nara (NAIST)

vấn đề dự đoán

Cho  $x$ , dự đoán  $y$

## vấn đề dự đoán

Cho  $x$ ,

dự đoán  $y$

### Một bài phê bình

sách Oh, man I love this book!

Quyển sách này chán quá...

### Nó có tích cực không?

có

không

nhị phân

Dự đoán (2

lựa chọn)

### Một tweet

Trên đường đến công viên!

### ngôn ngữ của nó

Tiếng Anh

tiếng Nhật

đa lớp

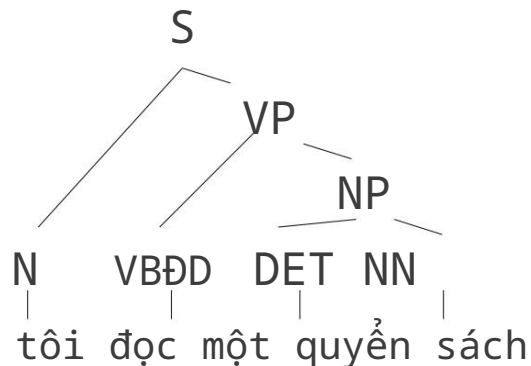
Dự đoán

(một số lựa chọn)

### Một câu

tôi đọc một quyển sách

### phân tích cú pháp của nó



có cấu trúc

Dự đoán

(hàng triệu lựa chọn)

## Ví dụ chúng tôi sẽ sử dụng:

- Đưa ra một câu giới thiệu từ Wikipedia • Dự đoán xem bài viết có nói về một người hay không

<u>Được cho</u>		<u>Dự đoán</u>
Gonso là một linh mục giáo phái Sanron (754-827) vào cuối thời Nara và đầu thời Heian.	→	Đúng!
Shichikuzan Chigogataki Fudomyoo là một di tích lịch sử tọa lạc tại Magura, thành phố Maizuru, tỉnh Kyoto.	→	KHÔNG!

- Đây là phân loại nhị phân (tất nhiên!)

# Thực hiện dự đoán

Làm thế nào để chúng tôi dự đoán?

Gonso là một linh mục giáo phái Sanron ( 754 - 827 ) vào cuối thời Nara và đầu thời Heian .

Shichikuzan Chigogataki Fudomyoo là một di tích lịch sử nằm ở Magura , thành phố Maizuru , tỉnh Kyoto .

## Làm thế nào để chúng tôi dự đoán?

Chứa "linh mục"  
có thể là người!

Chứa "(<#>-<#>)" có  
thể là người!

Gonso là một **linh mục giáo phái** Sanron ( **754 - 827** ) vào cuối thời Nara và đầu thời Heian .

"trang web" —→ Shichikuzan Chigogataki Fudomyoo chứa  
một **di tích lịch sử** nằm ở Magura, Maizuru **có lẽ không phải**  
là người!  
Thành phố , **tỉnh Kyoto** .

Chứa  
"tỉnh Kyoto" có lẽ  
không phải người!

## Kết hợp các mẫu thông tin

- Mỗi yếu tố giúp chúng tôi dự đoán là một tính năng chứa “linh mục”

chứa "(<#>)-(<#>)" chứa "trang web" chứa "Quận Kyoto"

- Mỗi tính năng có trọng số, dương nếu nó biểu thị “có”, và phủ định nếu nó chỉ ra “không”

$$w_{\langle \# \rangle} = 2$$

$$w_{\langle \# \rangle} = -3$$

$$w_{\text{trang web}} = -1$$

$$w_{\text{tỉnh Kyoto}} = -1$$

- Đối với một ví dụ mới, tính tổng các trọng số

Kuya (903-972) là một linh mục sinh ra ở quận Kyoto.

$$2 + -1 + 1 = 2$$

- Nếu tổng nhỏ nhất là 0: “có”, ngược lại: “không”



Hãy để tôi nói điều đó trong môn Toán!

$$\begin{aligned} y &= \text{dấu} (w \cdot \varphi(x)) \\ &= \text{dấu} \left( \sum_{i=1}^I w_i \varphi_i(x) \right) \end{aligned}$$

- $x$ : đầu vào
- $\varphi(x)$ : vectơ của các hàm đặc trưng  $\{\varphi_1(x), \varphi_2(x), \dots, \varphi_I(x)\}$
- $w$ : vectơ trọng số  $\{w_1, w_2, \dots, w_I\}$
- $y$ : dự đoán, +1 nếu “có”, -1 nếu “không”
  - ( $\text{dấu}(v)$  là +1 nếu  $v \geq 0$ , -1 nếu không)

Chức năng tính năng ví dụ:

Các tính năng của

Unigram • Bảng “số lần một từ cụ thể xuất hiện”

$x$  = Một trang web , tọa lạc tại Maizuru , Kyoto

“,”  $(x) = 1$   $\phi_{\text{unigram}}(x) = 1$   $\phi_{\text{unigram}}(\text{in}) = 1$   $\phi_{\text{unigram}}(x) = 1$   $\phi_{\text{unigram}}(\text{A}) = 2$   $\phi_{\text{unigram}}(\text{site}) = 1$   $\phi_{\text{unigram}}(\text{nằm ở}) = 1$   $\phi_{\text{unigram}}(x) = 1$   $\phi_{\text{unigram}}(x) = 1$

“Kyoto”  $(x) = 0$   $\phi_{\text{unigram}}(\text{Maizuru}) = 1$   $\phi_{\text{unigram}}(\text{unigram})$

“the”  $\phi_{\text{unigram}}(\text{ngôi đền}) = 0$   $\phi_{\text{unigram}}(\text{unigram})$

.

} Còn lại  
đều là 0

- Để thuận tiện, chúng tôi sử dụng tên đối tượng ( $\phi_{\text{unigram}}$  “A”) thay vì chỉ mục đối tượng ( $\phi_1$ )

# Tính tổng trọng số

$x$  = Một trang web , tọa lạc tại Maizuru , Kyoto

$\phi_{\text{unigram}}(x) = 1$	$\text{wunigram "a"} = 0$	$0 +$
$\phi_{\text{unigram}}(\text{"A"})(x) = 1$	$\text{wunigram "địa"} = -3$	$-3 +$
$\phi_{\text{unigram}}(\text{"site"})(x) = 1$	$\text{điểm" wunigram "tọa"} = 0$	$0 +$
$\phi_{\text{unigram}}(\text{"được"})(x) = 1$	$\text{lạc" wunigram} = 0$	$0 +$
$\phi_{\text{unigram}}(\text{"đặt"})(x) = 1$	$\text{"Maizuru"} = 0$	$0 +$
$\phi_{\text{unigram}}(\text{"Maizuru"})(x) = 2$	$\text{wunigram ","} = 0$	$0 +$
$\phi_{\text{unigram}}(\text{" ,"})(x) = 1$	$\text{wunigram "ở"} = 0$	$0 +$
$\phi_{\text{unigram}}(\text{"in"})(x) = 1$	$\text{wunigram "Kyoto"} = 2$	$0 +$
$\phi_{\text{unigram}}(\text{"Kyoto"})(x) = 0$	$\text{wunigram "linh mục" wunigram "đen"} = 0$	$0 +$
$\phi_{\text{unigram}}(\text{"đen"})(x) = 0$		

\*      =      •

-3

Không!

## Mã giả cho dự đoán

```
Predict_all(model_file, input_file):
```

```
    tải  $w$  từ model_file # so  $w[name] = wname$ 
```

```
    cho mỗi  $x$  trong
```

```
        input_file  $\phi = \text{create\_features}(x)$  # so  $\phi[name]$ 
```

```
        =  $\phi_{name}(x)$   $y' = \text{predict\_one}(w, \phi)$  # tính
```

```
        dấu( $w * \phi(x)$ ) in  $y'$ 
```

# Mã giả cho Dự đoán một ví dụ duy nhất

```
dự đoán_one(w, phi)
    điểm = 0
    cho mỗi tên, giá trị bằng phi          # điểm =  $w * \varphi(x)$ 
        nếu tên tồn tại
            trong w điểm += giá trị *
w[tên] nếu
    điểm >=
    0
    trả về 1 khác trả về -1
```

# Mã giả để tạo tính năng

## (Ví dụ: Tính năng Unigram)

CREATE\_FEATURES(x): tạo

bản đồ **phi** chia

**x** thành từng từ

trong từng từ **phi**

["UNI:" + word] += 1 # Chúng tôi thêm "UNI:" để biểu thị unigram  
trả về **phi**

- Bạn có thể **sửa đổi** chức năng này để sử dụng các tính năng khác!
  - Bigram? •
- Các tính năng khác?

# Trọng lượng học tập Sử dụng thuật toán Perceptron

# Trọng lượng học tập

- Khó tạo trọng số theo cách thủ công
  - Nhiều tính năng hữu ích có thể có
  - Việc thay đổi trọng số sẽ dẫn đến kết quả theo những cách không mong muốn
  - Thay vào đó, chúng ta có thể học hỏi từ dữ liệu được dán nhãn

y	x
1	FUJIIWARA no Chikamori (không rõ năm sinh và mất) là một võ sĩ đạo và nhà thơ sống vào cuối thời Heian. 1711 ) là
1	Ryonen ( 1646 - 29 tháng 10 , một nữ tu Phật giáo của Obaku Giáo phái sống từ đầu thời Edo đến giữa thời Edo.
-1	Khu định cư có hào nước là một ngôi làng được bao quanh bởi hào nước .
-1	Công viên thể thao Fushimi Momoyama nằm ở Momoyama-cho , Thành phố Kyoto , Tỉnh Kyoto .



# Học trực tuyến

tạo bản đồ  $w$   
cho các lần

lặp  $I$  cho mỗi cặp được gán nhãn  $x$ ,  $y$

trong dữ liệu  $\phi =$

$\text{create\_features}(x)$   $y' =$

$\text{predict\_one}(w, \phi)$  if  $y' \neq y$   $\text{update\_weights}(w, \phi, y)$

- Nói cách khác

- Cố gắng phân loại từng ví dụ huấn

- luyện • Mỗi khi chúng tôi mắc lỗi, hãy cập nhật trọng số

- Nhiều thuật toán học trực tuyến khác

- nhau • Đơn giản nhất là **perceptron**

## Cập nhật trọng lượng Perceptron

$$w \leftarrow w + y \varphi(x)$$

- Nói cách khác:

- Nếu  $y=1$ , tăng trọng số cho các đặc trưng trong  $\varphi(x)$ 
  - Các tính năng cho các ví dụ tích cực có trọng số cao hơn
- Nếu  $y=-1$ , giảm trọng số cho các đặc trưng trong  $\varphi(x)$ 
  - Các tính năng cho các ví dụ tiêu cực có trọng số thấp hơn

Mỗi khi chúng tôi cập nhật, các dự đoán của chúng tôi sẽ tốt hơn!

```
update_weights(w, phi, y) cho tên,  
    giá trị trong phi: w[name] +=  
        value * y
```

## Ví dụ: Cập nhật ban đầu

- Khởi tạo  $w=0$

$x$  = Một trang web , tọa lạc tại Maizuru , Kyoto  $y = -1$

$$w \cdot \phi(x) = 0 \quad y' = \text{sign}(w \cdot \phi(x)) = 1$$

$$y' \neq y$$

$$w \leftarrow w + y \cdot \phi(x)$$

$w$	$= -1$	$w$	$= -1$
unigram "Maizuru"		unigram "A"	
$w$	$= -2$	$w$	$= -1$
unigram ","		unigram "site"	
$w$	$= -1$	$w$	$= -1$
unigram "in"		unigram "location"	
$w$	$= -1$		
unigram "Kyoto"			

Ví dụ: Cập nhật lần thứ hai  $y = 1$

$x$  = Bị sốc , nhà sư sinh ra ở Kyoto



$$w \quad \varphi(x) = 4 \quad y \quad ' = \text{sign}(w \quad \varphi(x)) = 1$$

$$y \quad ' \quad y$$

$$w \quad w + y \quad \varphi(x)$$

$w$ unigram "Maizuru"	= -1	$w$ unigram "A"	= -1	$w$ unigram "Shoken"	= 1
$w$ unigram ","	= -1	$w$ unigram "site"	= -1	$w$ unigram "monk"	= 1
$w$ unigram "in"	= 0	$w$ unigram "location"	= -1	$w$ unigram "born"	= 1
$w$ unigram "Kyoto"	= 0				

# Bài tập

# Bài tập (1)

- Viết hai chương trình

- train-perceptron: Tạo mô hình perceptron •

- test-perceptron: Đọc mô hình perceptron và đưa ra một dự đoán trên mỗi dòng •

## Kiểm tra train-perceptron

- Đầu vào: test/03-train-input.txt • Trả lời: test/03-train-answer.txt

## Bài tập 2)

- Huấn luyện mô hình trên `data-en/titles-en-train.labeled`
- Dự đoán nhãn của `data-en/titles-en-test.word`
- Chấm điểm câu trả lời của bạn và báo cáo vào tuần tới
  - `script/grade-prediction.py data-en/titles-en-test.labeled your_answer`
- Thử thách bổ sung:
  - Tìm những chỗ mà mô hình mắc lỗi và phân tích tại sao
  - Phát minh ra các tính năng mới có thể tăng độ chính xác

Cảm ơn!