Logistic Regression
(Hồi quy Logistic)

Nhóm 8
Đỗ Tất Thành
Cao Thọ Hiếu

Mở đầu

- Là **thuật toán học máy có giám sát**
- Sử dụng cho **phân lớp đối tượng**
- Mẫu có thể được phân theo hai (**two-class case**) hoặc nhiều lớp (**multinomial logistic regression**)
- Là **discriminative classifier** (thuật toán tìm hiểu sự khác nhau giữa các lớp) thay vì **generative classifier** (tìm hiểu đặc điểm của mỗi lớp và phân loại dựa trên)
- Là thuật toán phân loại xác suất

Mở đầu

- Một thuật toán học máy phân loại thường có 4 thành phần:
  - Các đặt trưng của đầu vào (**feature representation**)
    - mẫu $x^{(i)}$ có các đặc trưng **$[x^{(i)}_1, x^{(i)}_2, ..., x^{(i)}_n]$**
  - Hàm phân loại (**classification function**)
    - phân loại mẫu vào **lớp dự đoán ŷ**
    - **sigmoid** (two-class) hoặc **softmax** (multinomial classifier)
  - Hàm mục tiêu cho quá trình học (**objective function for learning**)
    - Đo lường lỗi (sai khác giữa nhãn thực và nhãn dự đoán của mô hình) trong quá trình huấn luyện
    - **cross-entropy loss function**

Mở đầu

- Một thuật toán học máy phân loại thường có 4 thành phần (tiếp):
  - Thuật toán tối ưu hàm mục tiêu:
    - **stochastic gradient descent**
    - **batch gradient descent**
    - **mini-batch gradient descent**

- Hai pha chính:
  - Huấn luyện - **training:**
    - huấn luyện hệ thống (tìm trọng số – **weight** - **w** và **intercept** - **b** phù hợp cho mô hình dự đoán) dựa trên **mẫu** $\mathbf{x}^{(i)}$ và **nhãn** $\mathbf{y}^{(i)}$ cho trước trong tập huấn luyện
    - sử dụng
      - hàm mục tiêu **cross-entropy loss**

Mở đầu

- Hai pha chính:
  - Huấn luyện - **training**:
    - sử dụng (tiếp):
      - thuật toán tối ưu **stochastic** / **batch** / **mini-batch** gradient descent
  - Kiểm tra – **test**:
    - cho dữ liệu kiểm tra x, tính p(y|x) và trả về nhãn y của x cho p cao hơn
  - Pha hiệu chỉnh - **Validation** (có thể có hoặc không):
    - Hiệu chỉnh mô hình (**Regularization**) để tránh **overfitting** hoặc **underfitting**
    - Hiệu chỉnh L1 (**L1 regularization**)
    - Hiệu chỉnh L2 (**L2 regularization**)

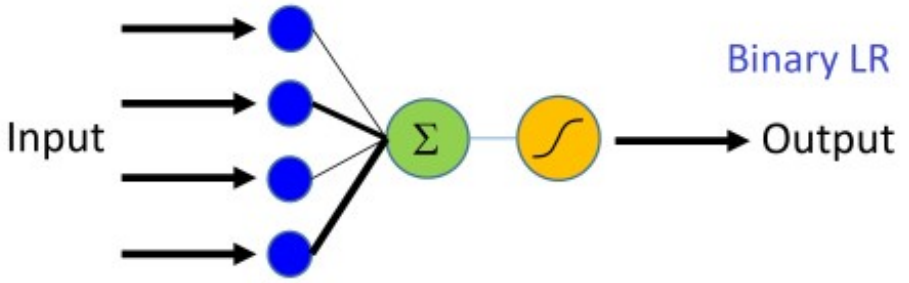
Cơ chế hoạt động

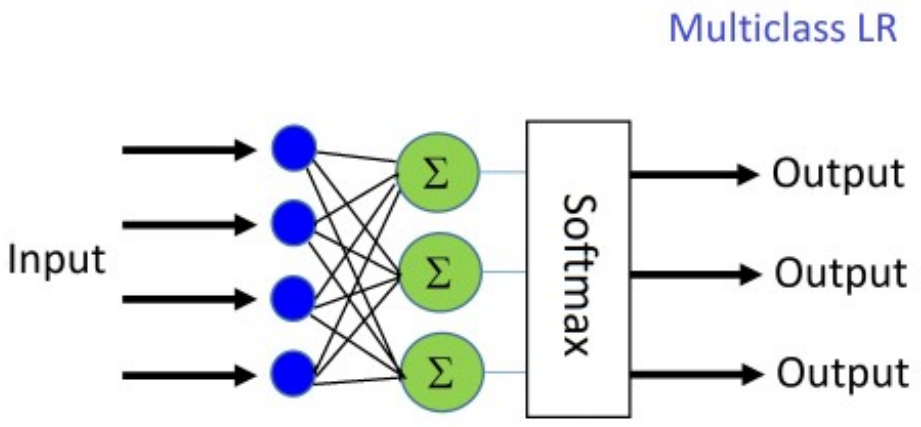
- Cho data set:
  - [[x^1_1 , x^1_2 , ..., x^1_n , y^1],

    …

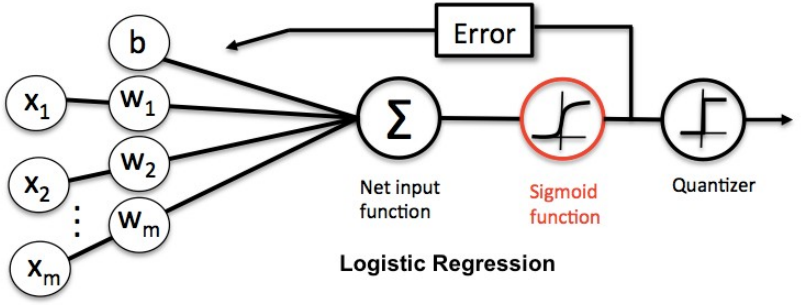
    [x^m_1 , x^m_2 , ..., x^m_n , y^m]]
  - $\mathbf{x}^i_j$ là đặc trưng j của quan sát thứ i, y^i là nhãn của quan sát thứ I
- Tìm trọng số w và **intercept** / **bias term** b
  - w := [w_1 , w_2 , ..., w_n]’
  - b

  sao cho $L_{CE}(\hat{y}, y) \rightarrow$ min
  - $L_{CE}(\hat{y}, y)$:= cross-entropy loss function
  - $\hat{Y} \rightarrow$ nhãn dự đoán, y $\rightarrow$ nhãn thực

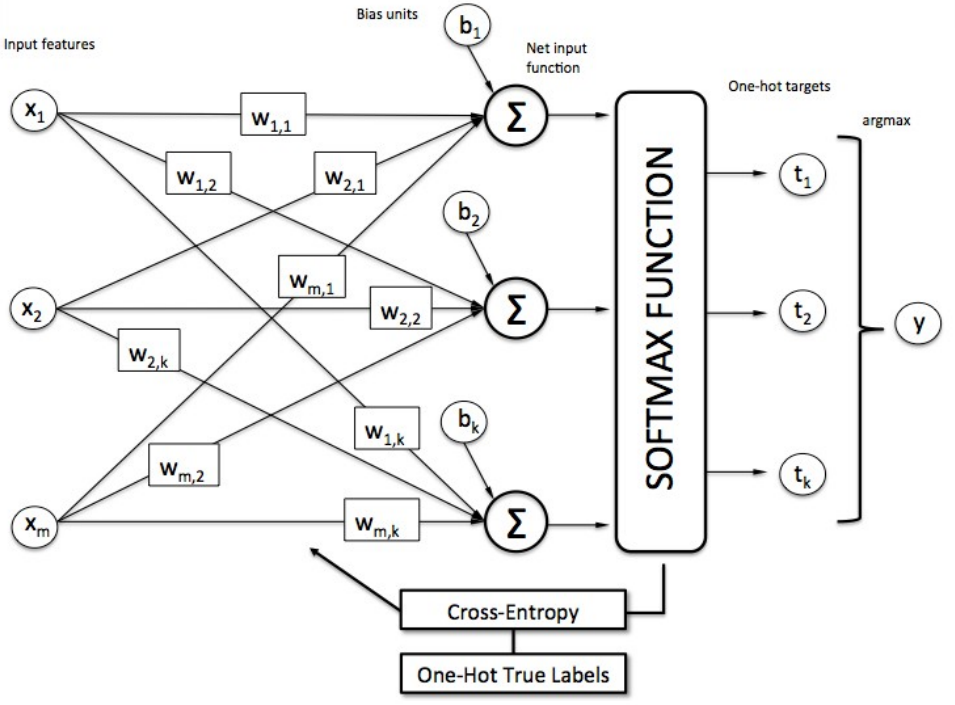




Cơ chế hoạt động

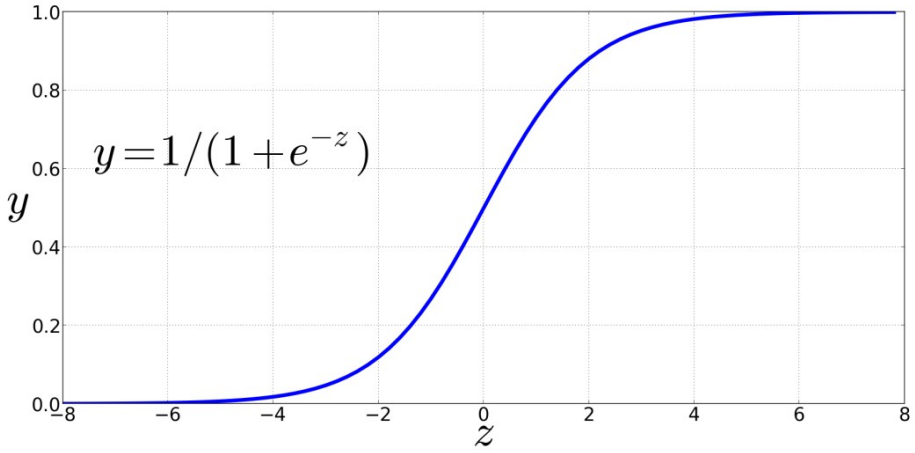


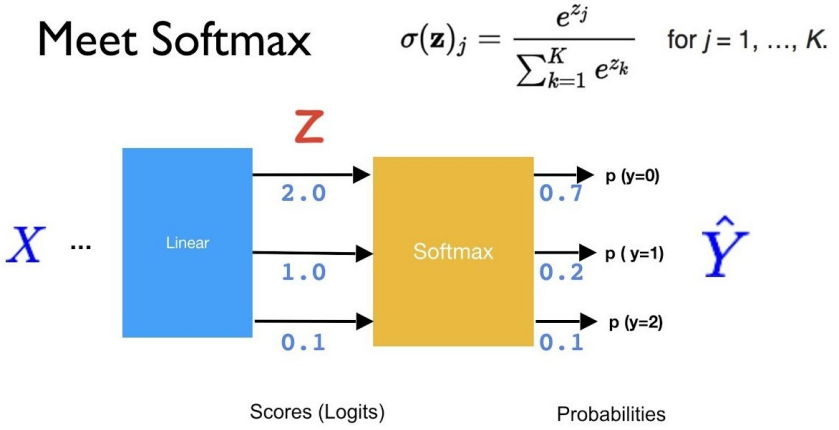
Logistic Regression



Softmax Regression

Hàm phân lớp - Classification Function





- Hàm sigmoid σ:
  - Phân loại 2 lớp
    - y = 0 và y = 1
    - Y = + (positive) và y = - (negative)
    - ...

- Hàm softmax
  - Phân loại k lớp (k >= 2)

Hàm phân lớp – Classification Function

- $x = [x_1 , x_2 , ..., x_n]$
- nhãn y
- $w = [w_1 , w_2 , ..., w_n]$‘
- bias/intercept b
- z = x * w + b

- Phân loại nhị phân:
  - $P(y = 1|x) = \sigma (w \cdot x + b)$
  - $P(y = 0|x) = 1 - \sigma (w \cdot x + b)$
  - $\hat{y} = 1$ if $P(y = 1|x) > 0.5$ else $\hat{y} = 0$
- Phân loại k lớp:

$$p(y = c|x) = \frac{e^{w_c \cdot x + b_c}}{\sum_{j=1}^{k} e^{w_j \cdot x + b_j}}$$

Hàm mục tiêu - Cross-Entropy Loss Function

- Loss Function:
  - L(ŷ, y) := Định lượng mức độ sai khác giữa ŷ và y
  - L ưu tiên cho các nhãn được gán đúng.
  - mục đích của thuật toán LR là chọn ra được w, b có thể cực đại được log-xác suất của các nhãn gán đúng y trong bộ dữ liệu huấn luyện
  - loss function thường dùng là cross-entropy loss.

Hàm mục tiêu - Cross-Entropy Loss Function

- Với phân lớp nhị phân (y = 1 || y = 0):

$$
\begin{aligned}
p(y|x) &= \hat{y}^{y}(1-\hat{y})^{1-y} \\
\log p(y|x) &= \log\left[\hat{y}^{y}(1-\hat{y})^{1-y}\right] \\
&= y\log\hat{y} + (1-y)\log(1-\hat{y}) \\
L_{CE}(\hat{y},y) = -\log p(y|x) &= -[y\log\hat{y} + (1-y)\log(1-\hat{y})] \\
L_{CE}(w,b) &= -[y\log\sigma(w\cdot x+b) + (1-y)\log(1-\sigma(w\cdot x+b))]
\end{aligned}
$$

Hàm mục tiêu - Cross-Entropy Loss Function

- Với phân lớp k lớp (k >= 2):

$$L_{CE}(\hat{y}, y) = -\sum_{k=1}^{K} 1\{y = k\} \log p(y = k|x)$$

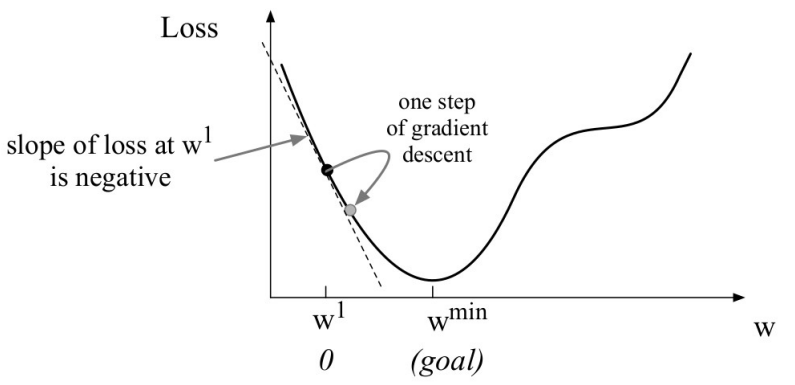
$$= -\sum_{k=1}^{K} 1\{y = k\} \log \frac{e^{w_k \cdot x + b_k}}{\sum_{j=1}^{K} e^{w_j \cdot x + b_j}}$$

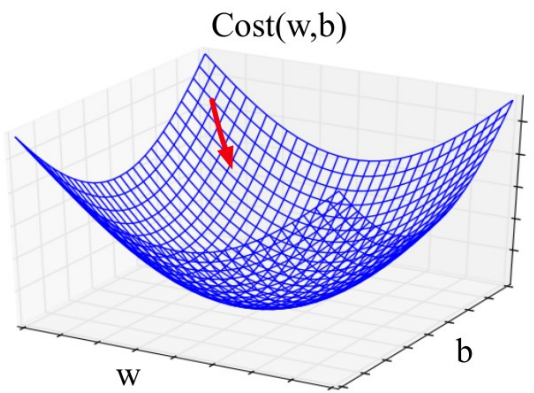
Tối ưu hàm mục tiêu - Gradient Descent

- Mục tiêu của LR:

$$\theta = w, b \qquad \hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} L_{CE}(y^{(i)}, x^{(i)}; \theta)$$

- Gradient của L_{CE} ứng với theta cho biết chiều tăng cúa L_{CE} tại theta. Đi theo chiều giảm gradient (Gradient Descent), tìm được vị trí $(w, b)^{argmin}$ ứng với cực tiểu hàm LCE. $(w, b)^{argmin}$ là tham số cần tìm của mô hình.





Gradient Descent

- Phương pháp chung tính gradient:

$$\nabla_\theta L(f(x;\theta),y)) = \begin{bmatrix} \frac{\partial}{\partial w_1} L(f(x;\theta),y) \\ \frac{\partial}{\partial w_2} L(f(x;\theta),y) \\ \vdots \\ \frac{\partial}{\partial w_n} L(f(x;\theta),y) \end{bmatrix}$$

- Gradient của phân lớp nhị phân:

$$L_{CE}(w,b) = -[y \log \sigma(w \cdot x + b) + (1-y) \log (1 - \sigma(w \cdot x + b))]$$

$$\frac{\partial L_{CE}(w,b)}{\partial w_j} = [\sigma(w \cdot x + b) - y] x_j$$

- Gradient của phân lớp k lớp:

$$L_{CE}(\hat{y},y) = -\sum_{k=1}^{K} 1\{y=k\} \log p(y=k|x)$$

$$= -\sum_{k=1}^{K} 1\{y=k\} \log \frac{e^{w_k \cdot x + b_k}}{\sum_{j=1}^{K} e^{w_j \cdot x + b_j}}$$

$$\frac{\partial L_{CE}}{\partial w_k} = -(1\{y=k\} - p(y=k|x)) x_k$$

$$= -\left(1\{y=k\} - \frac{e^{w_k \cdot x + b_k}}{\sum_{j=1}^{K} e^{w_j \cdot x + b_j}}\right) x_k$$

- Cập nhật trọng số:

$$\theta_{t+1} = \theta_t - \eta \nabla L(f(x;\theta),y)$$

Một số kỹ thuật tính Gradient

- Stochastic GD:
  - thuật toán online nhằm tối thiểu hóa loss function
  - tính gradient và cập nhật trọng số trên mỗi mẫu huấn luyện
- Mini-Batch GD:
  - huấn luyện nhóm m mẫu. Nếu m = 1 thì là Stochastic GD. Hoặc m <= kích thước tập mẫu.
  - Cho hiệu suất tính toán tốt do có thể chọn kích thước nhóm phù hợp với năng lực tính toán của máy tính hiện có. Vector hóa dữ liệu dễ dàng và có thể tận dụng năng lực tính toán song song.
- Batch GD:
  - huấn luyện với toàn bộ tập mẫu đồng thời (mini-batch với m = kích thước tập train).

function STOCHASTIC GRADIENT DESCENT($L()$, $f()$, x, y) **returns** θ
    # where: L is the loss function
    # f is a function parameterized by θ
    # x is the set of training inputs $x^{(1)}$, $x^{(2)}$, ..., $x^{(n)}$
    # y is the set of training outputs (labels) $y^{(1)}$, $y^{(2)}$, ..., $y^{(n)}$

$\theta \leftarrow 0$
repeat til done # see caption
  For each training tuple $(x^{(i)}, y^{(i)})$ (in random order)
    1. Optional (for reporting): # How are we doing on this tuple?
       Compute $\hat{y}^{(i)} = f(x^{(i)}; \theta)$ # What is our estimated output $\hat{y}$?
       Compute the loss $L(\hat{y}^{(i)}, y^{(i)})$ # How far off is $\hat{y}^{(i)}$) from the true output $y^{(i)}$?
    2. $g \leftarrow \nabla_\theta L(f(x^{(i)}; \theta), y^{(i)})$ # How should we move θ to maximize loss?
    3. $\theta \leftarrow \theta - \eta\, g$ # Go the other way instead
return θ

Ví dụ: Sentiment Analysis

5.1.1 Example: sentiment classification

Let's have an example. Suppose we are doing binary sentiment classification on movie review text, and we would like to know whether to assign the sentiment class $+$ or $-$ to a review document *doc*. We'll represent each input observation by the 6 features $x_1 ... x_6$ of the input shown in the following table; Fig. 5.2 shows the features in a sample mini test document.

Var	Definition	Value in Fig. 5.2
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if ``no'' } \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if ``!'' } \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

Sentiment Analysis

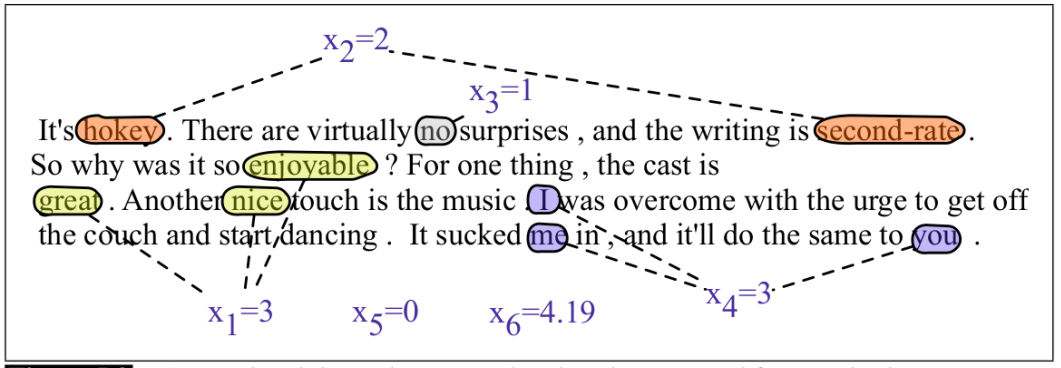


Figure 5.2 A sample mini test document showing the extracted features in the vector x.

Given these 6 features and the input review x, $P(+|x)$ and $P(-|x)$ can be computed using Eq. 5.5:

$$\begin{aligned} p(+|x) = P(Y=1|x) &= \sigma(w\cdot x+b) \\ &= \sigma([2.5,-5.0,-1.2,0.5,2.0,0.7]\cdot[3,2,1,3,0,4.19]+0.1) \\ &= \sigma(.833) \\ &= 0.70 \qquad (5.6) \\ p(-|x) = P(Y=0|x) &= 1-\sigma(w\cdot x+b) \\ &= 0.30 \end{aligned}$$

Ví dụ: Period Disambiguation

Logistic regression is commonly applied to all sorts of NLP tasks, and any property of the input can be a feature. Consider the task of **period disambiguation**: deciding if a period is the end of a sentence or part of a word, by classifying each period into one of two classes EOS (end-of-sentence) and not-EOS. We might use features like x_1 below expressing that the current word is lower case and the class is EOS (perhaps with a positive weight), or that the current word is in our abbreviations dictionary ("Prof.") and the class is EOS (perhaps with a negative weight). A feature can also express a quite complex combination of properties. For example a period following an upper case word is likely to be an EOS, but if the word itself is *St.* and the previous word is capitalized, then the period is likely part of a shortening of the word *street*.

$$x_1 = \begin{cases} 1 & \text{if } \text{“}Case(w_i) = \text{Lower”} \\ 0 & \text{otherwise} \end{cases}$$

$$x_2 = \begin{cases} 1 & \text{if } \text{“}w_i \in \text{AcronymDict”} \\ 0 & \text{otherwise} \end{cases}$$

$$x_3 = \begin{cases} 1 & \text{if } \text{“}w_i = \text{St. \& } Case(w_{i-1}) = \text{Cap”} \\ 0 & \text{otherwise} \end{cases}$$

Project

- Sentiment Analysis cho đánh giá của người dùng:
  - Tiếng Anh:
    - Đánh giá nhận xét của khán giả trên trang IMDB
    - Dataset aclImdb
  - Tiếng Việt:
    - Đánh giá nhận xét của người mua trên trang web thương mại điện tử
    - Dataset VLSP 2016

Project

- Libs & Packages:
  - VNCoreNLP (cho words tokenize).
  - Scikit Learn (cho Logistic Regression).
  - Numpy/Scipy (cho Sparse Matrix).
  - Pandas
  - VnEmoLex (bộ dữ liệu về tính từ thể hiện sắc thái)

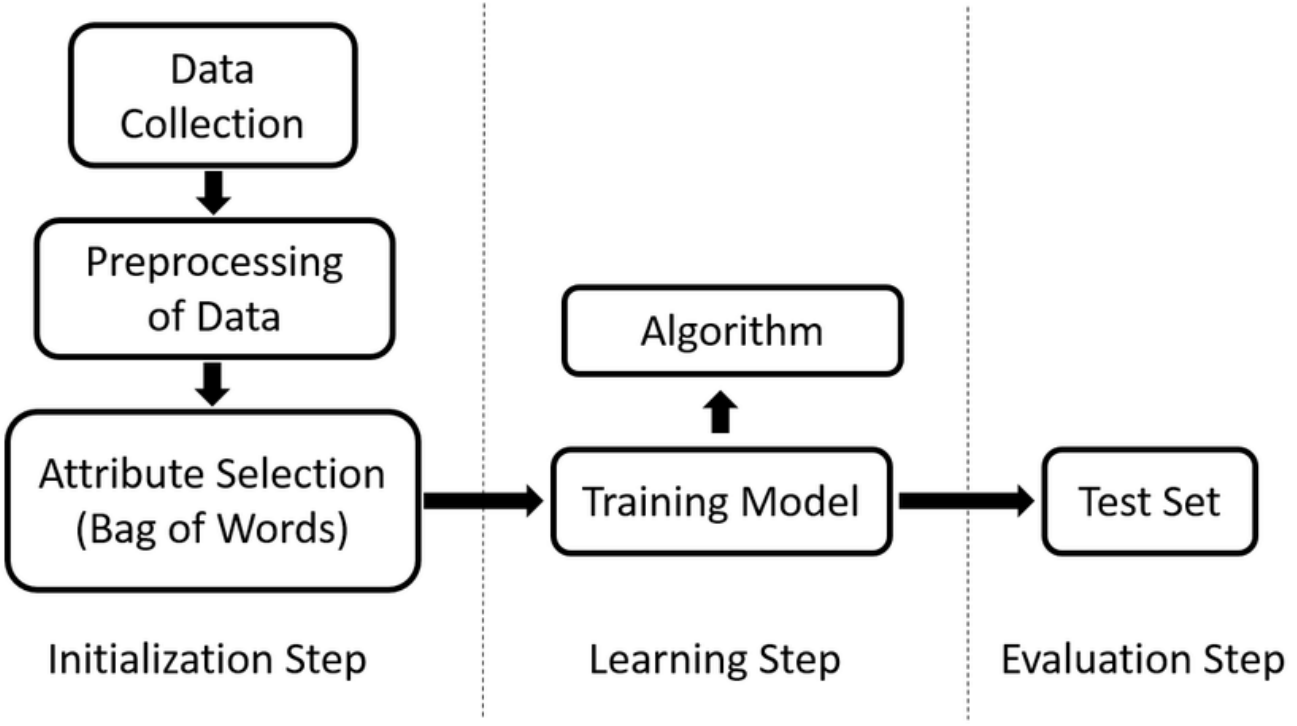
VNEmoLex

		Vietnamese	Positive	Negative	Anger/ t c gi n	Anticipation / Hi vọng	Disgust / chán ghét	Fear / s hãi	Joy/ thích thú	Sadness/ bu n bã	Surprise / ng c nhiên	Trust/ tin tY ng	
2	celebrity	Danh nhân	1	0	0	0	0	0	1	0	1	1	4
3	celebrity	danh tiếng	1	0	0	1	0	0	1	0	1	1	5
4	opera	nghệ thuật nhạc k	0	0	0	0	0	0	1	0	1	0	2
5	celebrity	sự nổi tiếng	1	1	0	1	0	0	1	0	1	1	6
6	endless	bất tận	0	0	0	0	0	1	0	1	0	1	3
7	epidemic	b nh d ch	0	1	1	1	1	1	0	1	1	0	7
8	retirement	bỏ cuộc	0	1	0	1	0	1	0	1	0	0	4
9	outburst	bùng nổ	0	0	1	0	0	1	1	1	1	0	5
10	romance	cường điệu	1	0	0	0	1	0	0	0	1	0	3
11		dơ	0	1	1	1	1	1	0	1	1	0	7
12	nurture	giáo dục	1	0	0	1	0	0	1	0	0	1	4
13	romance	lãng mạng	1	0	0	1	0	1	1	1	1	1	7
14	romance	mơ mộng	1	0	0	1	0	1	1	1	1	1	7
15	retirement	nghỉ hưu	0	0	0	1	0	0	1	1	0	1	4
16		ngừng hoạt động	0	1	1	1	1	1	0	1	1	0	7
17		nhơ	0	1	1	1	1	1	0	1	1	0	7
18	nurture	nuôi nấng	1	0	1	1	1	1	1	0	0	1	7
19		trộm	0	1	0	1	1	1	1	1	1	0	7
20	grim	ác nghiệt	0	1	1	1	1	1	0	1	0	0	6
21	tempest	bão	0	1	0	0	0	1	0	0	1	0	3
22	unbeaten	b t khu t	1	0	0	1	0	0	1	0	1	1	5
23		bị đối xử tàn tệ	0	1	1	1	1	1	0	1	0	0	6
24		bị lạm dụng	0	1	1	1	1	1	0	1	0	0	6
25		bị ngược đãi	0	1	1	1	1	1	0	1	0	0	6
26	lace	buộc	0	0	1	0	0	1	0	1	0	1	4
27	highest	cao nhất	0	0	0	1	0	0	1	0	1	0	3
28	pray	cầu xin	0	0	0	1	0	0	0	0	1	1	3
29	bloodshed	chém giết	0	1	1	0	1	1	0	1	0	0	5
30		chênh lệch	0	0	0	1	0	0	0	0	1	0	2
31	cash	có tiền	1	0	0	0	0	0	1	0	0	1	4

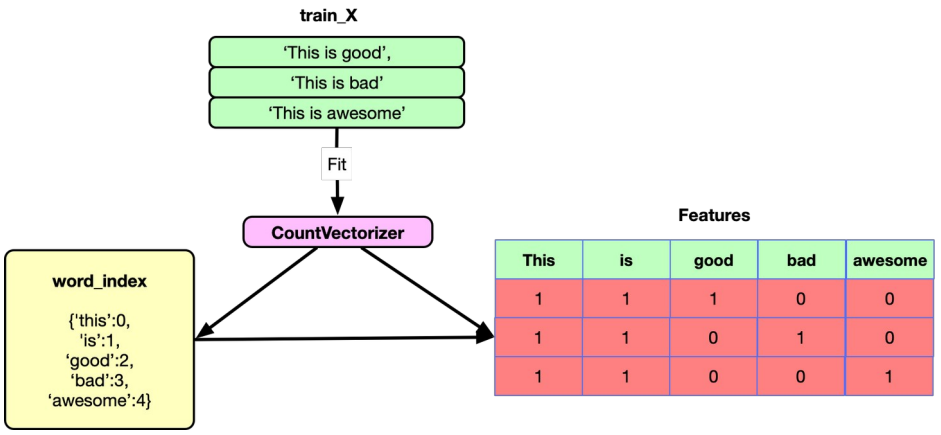
Sheet1

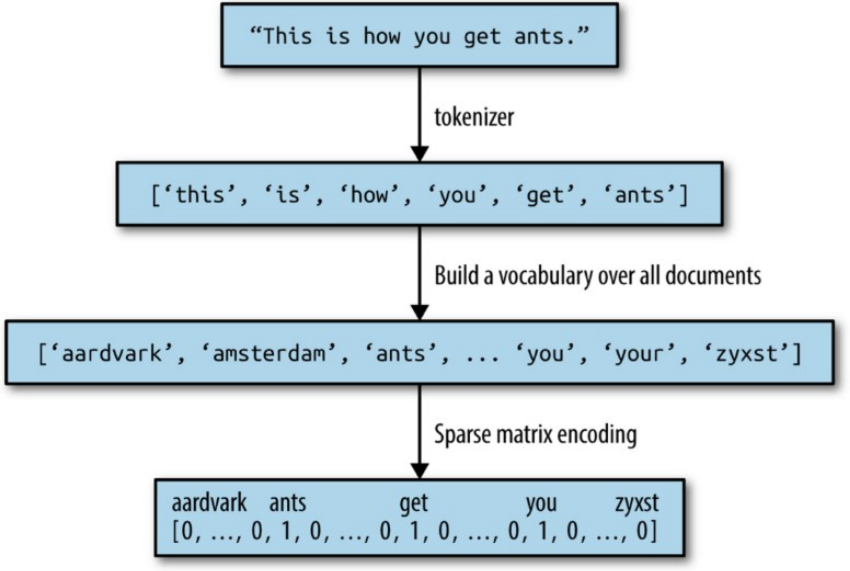
Project

- Các bước thực hiện:
  - Preprocessing = tách từ, lọc ký tự đặc biệt
  - LG Model: cross-validation = 5



Bag-Of-Words





Bag of words processing [1]

Bag-Of-Words

	it	is	puppy	cat	pen	a	this
it is a puppy	1	1	1	0	0	1	0
it is a kitten	1	1	0	0	0	1	0
it is a cat	1	1	0	1	0	1	0
that is a dog and this is a pen	0	2	0	0	1	2	1
it is a matrix	1	1	0	0	0	1	0

Đánh giá cho tiếng Việt

Phương pháp tiếp cận	Độ chính xác
BoW = Tất cả các từ khác nhau trong văn bản. Vector bình luận là vector nhị phân (chỉ xét đến sự hiện diện của từ)	65%
BoW = Tất cả các từ khác nhau trong văn bản. Vector bình luận là vector tần suất của từ xuất hiện trong văn bản	63-64%
BoW = Các tính từ trong VnEmoLex Vector bình luận là vector tần suất của từ xuất hiện trong văn bản	60-62%
BoW = Tất cả các từ khác nhau trong văn bản. Vector bình luận là vector tần suất của từ xuất hiện trong văn bản, tăng tần suất lên N lần cho các tính từ trong VnEmoLex	62-64%
TF-IDF	NaN

*Độ chính xác của mô hình cho tiếng Anh là 88%

Tài liệu tham khảo

- Chapter 5 - Speech and Language Processing 3rd edittion - Daniel Jurafsky & James H. Martin
- Large Movie Review Dataset - Stanford AI Lab
- http://vlsp.org.vn/ - 2019
- https://github.com/vncorenlp/VnCoreNLP - 2019
- VnEmoLex: A Vietnamese emotion lexicon for sentiment intensity analysis - KTLab

Cảm ơn vì đã lắng nghe!