

Department of Linguistics and Translation

香港城市大學 City University of Hong Kong

Computational Linguistics LT3233

Jixing Li Lecture 7: Logistic Regression

Slides adapted from Dan Jurafsky

© Jixing Li

Lecture plan

- Naive Bayes and Laplace smoothing review
- Logistic regression
	- feature representation
	- classification function: **sigmoid**
	- loss function: **cross-entropy loss**
	- optimization algorithm: **gradient descent**
- Short break (15 mins)
- Hands-on exercises

Logistic regression

The task of text classification

- *Input*:
	- a document *x*
	- a fixed set of classes $C = \{c_1, c_2,..., c_l\}$
- *Output*: a predicted class $\hat{y} \in C$

Naive Bayes: Compare P (male|卓琳) P (female|卓琳) à **generative classifier**

Logistic regression: P (male|卓琳) à **discriminative classifier**

Components of logistic regression

1. feature representation of the input. For each input observation $x^{(i)}$, a vector of features $[x_1, x_2, ..., x_n]$.

 $X^{(i)} = \left[\frac{1}{2}, \frac{1}{2} \right]$

- $=$ [卓, 琳, Cheuk, Lam, LLA]
- **2. classification function** that computes \hat{y} , the estimated class: **sigmoid** functions
- **3. objective function for learning**: **cross-entropy loss**
- **4. algorithm for optimizing** the objective function: **gradient descent**

Features in logistic regression

Input vector: $x = [x_1, x_2, ..., x_n]$

[卓, 琳, Cheuk, Lam, LLA] Probability of these features in female names: \rightarrow x = [0.5, 0.7, 0.5, 0.6, 0.8]

Weights: one per feature: $w = [w_1, w_2, ..., w_n]$ \rightarrow w = [0.1, 0.8, -0.1, 0.2, 0.7]

Prediction: $z = w \cdot x + b$

$$
z = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + w_4 * x_4 + w_5 * x_5 + b
$$

= 0.05 + 0.56 + (-0.05) + 0.12 + 0.56 + 0.3
= 1.54

Transform prediction into probability Transform prediction into probability IT GITSTOTTH PLEARENTH INTER P

 λ , we λ that μ $z = w \cdot x + b$

z is a number, But we We'd like a classifier that gives us a probability,
installed Naive Rayes did just like Naive Bayes did
 But nothing in Eq. 3.3 and probability and probability, the antithe gives its name. The signoment is name. The signoment is not the sigmoid of the following equation, i.e., i
The sigmoid equation, i.e., i.e.

Solution: use a function of z that goes from 0 to 1 shown graphically in Fig. 5.1: $b = b$

has a number of advantages; it takes a real-valued number and maps it into the range

$$
y = \sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + \exp(-z)} \qquad \Rightarrow \text{ the sigmoid function}
$$

The sigmoid function negative; *z* ranges from • to •. sigmoid function (named because it looks like an *s*) is also called the logistic func- \mathbf{r}

outlier values toward 0 or 1.

Figure 5.1 The sigmoid function *y* = $i \hat{f}$ the range $\hat{h} > 0$ $\hat{\mathbf{v}} = \begin{bmatrix} 1 & \mathbf{0} \end{bmatrix}$ is nearly subset to same to same to same to say the ends to square the ends of the ends to square the ends to sq $\hat{y} =$ $\int 1$ if $w \cdot x + b > 0$. 0 otherwise \mathbf{i} f w \cdot x+ if w∙x+ $b > 0$ if w∙x+b ≤ 0

Example

[卓, 琳, Cheuk, Lam, LLA] $x = \{0.5, 0.7, 0.5, 0.6, 0.8\}$ $w = [0.1, 0.8, -0.1, 0.2, 0.7]$ $z = w \cdot x + b$ $= w_1^*x_1 + w_2^*x_2 + w_3^*x_3 + w_4^*x_4 + w_5^*x_5 + b$ $= 0.05 + 0.56 + (-0.05) + 0.12 + 0.56 + 0.3$ $= 1.54$ \blacktriangleleft

$$
\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-1.54}} = 0.82 > 0.5 \rightarrow
$$
 female

How to calculate weights?

Supervised classification:

We know the correct label *y* (either 0 or 1) for each *x*. But what the system produces is an estimate, \hat{y}

We want to know how far is the classifier output:

 $\hat{v} = \sigma(w \cdot x + b)$

from the true output:

 $y =$ either 0 or 1

We'll call this difference the loss:

 $L(\hat{y}, y)$ = how much \hat{y} differs from the true *y*

Binary cross-entropy loss express the probability *p*(*y|x*) that our classifier produces for one observation as *p*(*y|x*) = *y* ^ˆ *^y* (1*^y*

Goal: maximize the probability of the correct label $p(y|x)$ Since there are only 2 outcomes (0 or 1), we can express the probability *p*(*y*|*x*) from our classifier as: express the probability *p*(*y|x*) that our classifier produces for one observation as simplifies to 1*y* Now we take the log of both sides. This will turn out to be handy mathematically, **abdi. Maximize** the probability of the correct laber $p(y|x)$ SINCE LIIEIE die

$$
p(y|x) = \hat{y}^{y} (1-\hat{y})^{1-y}
$$
 if y=1, this simplifies to \hat{y}
if y=0, this simplifies to 1- \hat{y}

Now take the log of both sides: Now take the log of both sides: Now take the log of both sides:

into loss function (something that we need to minimize), we'll just flip the sign on

$$
\log p(y|x) = \log [\hat{y}^{y} (1-\hat{y})^{1-y}]
$$

= $y \log \hat{y} + (1-y) \log(1-\hat{y})$

Now flip sign to turn this into a loss: Something to **minimize** Eq. 5.10 in the sign to turn this into a loss: Something to **minimize** that $\frac{1}{\sqrt{2\pi}}$ Eq. 5.10. The result is the cross-entropy loss *L*CE:

 $L_{CE}(\hat{y}, y) = -\log p(y|x) = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})]$ Endss entrupy to should be member to the maximized to the maximized to the maximized to the maximized to the m **Exercise 10.10. The cross-entropy loss:** negative log likelihood loss Finally, we can plug in the definition of ˆ*y* = s(*w· x*+*b*):

¹*^y* (5.9)

Example

[卓, 琳, Cheuk, Lam, LLA] $x = \{0.5, 0.7, 0.5, 0.6, 0.8\}$ $w = \{0.1, 0.8, -0.1, 0.2, 0.7\}$ $b = 0.5$ $\hat{v} = \sigma(w \cdot x + b) = 0.82$

if 卓琳 is female: $y = 1$: $L_{CE}(\hat{y}, y) = (y \log \hat{y} + (1-y) \log(1-\hat{y})) = -\log(0.82) = 0.2$ if 卓琳 is male: $y = 0$: $L_{CE}(\hat{y}, y) = (y \log \hat{y} + (1-y) \log(1-\hat{y})) = -\log(1-0.82) = 1.7$

\rightarrow The loss is greater when the prediction is wrong

Minimize the loss

Let's make explicit that the loss function is parameterized by weights $\theta = (w,b)$ U y weights $\mathsf{U} - (\mathsf{W}, \mathsf{U})$

And we'll represent \hat{y} as $f(x; \theta)$ to make the dependence on θ more obvious So the governous which minimizes the set of weights which minimizes the loss function, and averaged the loss function, and ave

We want the weights that minimize the loss, averaged over all examples: over all examples:

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

Gradient descent

How do I get to the bottom of this river canyon?

Look around me 360[∘] Find the direction of steepest slope down Go that way

Gradient descent for a single scaler

Minimize loss: Given the current w, Move w in the reverse direction from the slope of the function

The **gradient** of a function of many variables is a vector pointing in the direction of the greatest increase in a function.

Gradient descent: Find the gradient of the loss function at the current point and move in the opposite direction.

Gradient descent

The new weight w^{t+1} is the old weight w^t minus the value of the gradient weighted by a learning rate n

$$
w^{t+1} = w^t - \eta \frac{d}{dw} L(f(x; w), y)
$$

The gradient is gradient is gradient is gradient is directed to the direction of the directional components of
in the directional components of the directional components of the directional components of the directional c

learning rate: Higher learning gradient (a vector of the \rightarrow a **hyperparameter** between to the weight w) in the N-dimension superiorm of the *N-dimension*
Supervision, but are chosen by learning rate: Higher learning rate means move w faster \rightarrow a **hyperparameter** not learned by algorithm from algorithm designer.

gradient (a vector of the derivatives with respect to the weight w)

Gradient in N-dimensional space of a 2-dimensional gradient vector taken at the red point.

The gradient expresses the directional components of the sharpest slope along each of the N dimensions. For each dimension w_i, we express the slope as a partial derivative ∂ of the loss ∂w_i In order to update the update $\frac{1}{2}$ and $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$

the *w* dimension and in the *b* dimension. Fig. 5.4 shows a visualization of the value

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

is:

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

Note in Eq. 5.18 that the gradient with respect to a single weight *wj* represents a

Example

[卓, 琳, Cheuk, Lam, LLA] $x = \{0.5, 0.7, 0.5, 0.6, 0.8\}$

1. initialize w and b, set η $w = [0, 0, 0, 0, 0]$, $b = 0$, $\eta = 0.1$

2. compute \hat{v} $\hat{v} = \sigma(w \cdot x + b) = 0.5$

3. compute the gradients for w and b $Gw = (0.5 - y)x = -0.5x = [-0.25, -0.35, -0.25, -0.3, -0.4]$ $Gb = 0.5 - y = -0.5$

© Jixing Li 4. update w and b $w_{t+1} = w_t - \eta^* Gw = [0, 0, 0, 0, 0] - 0.1^*[-0.25, -0.35, -0.25, -0.3, -0.4] = [0.025, 0.035,$ $0.205, 0.03, 0.04$] $b_{t+1} = b_t - \eta *Gb = 0 - 0.1 * (-0.5) = 0.05$

Calculate gradient descent over all examples

- [卓, 琳, Cheuk, Lam, LLA] $x_1 = [0.5, 0.7, 0.5, 0.6, 0.8]$ $\left[\frac{1}{11}, \frac{1}{2}, \frac{1}{2$
- 1. initialize w and b, set η $w = [0, 0, 0, 0, 0], b = 0, \eta = 0.1$
- 2. compute \hat{y} $\hat{v}_1 = \sigma(w \cdot x + b) = 0.5, \hat{v}_2 = \sigma(w \cdot x + b) = 0.5$

3. compute the gradients for w and b $Gw =$! $\frac{1}{2}((0.5-y)x_1 + (0.5-y)x_2) = \frac{1}{2}(-0.5x_1 - 0.5x_2) = [0.025, 0.025, -0.1, 0, -0.2]$ $Gb = \frac{1}{2}$ $\frac{1}{2}((0.5-y_1)+(0.5-y_2))=0$

4. update w and b $w_{t+1} = w_t - \eta^* Gw = [0, 0, 0, 0, 0] - 0.1^* [0.025, 0.025, -0.1, 0, -0.2] = [-0.0025, -0.0025,$ $0.01, 0, 0.02$], $b_{t+1} = b_t - \eta *Gb = 0$

© Jixing Li

To do

• Optional reading: **SLP** Ch5