What do you call a bad dream about machine learning?

A logistic nightmare

CS 1671/2071
Human Language Technologies

Session 10: Logistic regression, part 1

Michael Miller Yoder

February 12, 2025

University of Pittsburgh | School of Computing and Information

Course logistics: quiz and homework

- Quiz on Canvas due **this Thu Feb 13**
  - Check the description on Canvas to see what readings it covers
- Homework 2 is due **next Thu Feb 20**
  - What you can work on now: look at scikit-learn `LogisticRegression` documentation. Try loading data, extract features (vectorize) the data and train a model from it. It's fine to use code from the clickbait classification exercises in class!
  - The **optional** Kaggle competition has been released

Course logistics: project

- Next project milestone: project proposal due Feb 28 (stay tuned for more details on that)
- If I emailed your group about choosing different directions or datasets, please respond over email or book office hours to talk through by **this Fri Feb 14**
- Choose a communication chat platform for your group to collaborate (Teams, Discord, Signal, WhatsApp, etc)

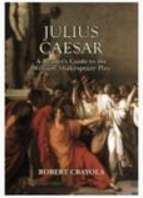
Lecture overview: Logistic regression part 1

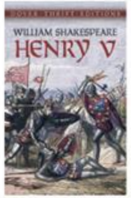
- Input to classification: features from text
- Logistic regression
- Binary classification with logistic regression
- Multinomial logistic regression
- Coding activity

Input to classification tasks: features

Term-document matrix

- Each cell is the count of term t in a document d ($tf_{t,d}$).
- Each document is a count vector in $\mathbb{N}^V$, a column below.





	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

Slide credit: David Mortensen

Movie Ratings

- A training set of movie reviews (with star ratings 1 - 5)
- A set of features for each message (considered as a bag of words)
    - For each word: Number of occurrences
    - Whether phrases such as *Excellent*, *sucks*, *blockbuster*, *biggest*, *Star Wars*, *Disney*, *Adam Sandler*, ...are in the review

Slide credit: David Mortensen

Spam Detection

- A training set of email messages (marked *Spam* or *Not-Spam*)
- A set of features for each message
  - For each word: Number of occurrences
  - Whether phrases such as “Nigerian Prince”, “email quota full”, “won ONE HUNDRED MILLION DOLLARS” are in the message
  - Whether it is from someone you know
  - Whether it is a reply to your message
  - Whether it is from your domain (e.g., `cmu.edu`)

Slide credit: David Mortensen

Logistic regression

What Goes into a (Discriminative) ML Classifier?

1. A feature representation
2. A classification function
3. An objective function
4. An algorithm for optimizing the objective function

What Goes into Logistic Regression?

GENERAL	IN LOGISTIC REGRESSION
feature representation	represent each observation $\mathbf{x}^{(i)}$ as a **vector of features** $[x_1, x_2, \ldots x_n]$
classification function	**sigmoid function** (logistic function)
objective function	**cross-entropy loss**
optimization function	(stochastic) **gradient descent**

Slide credit: David Mortensen

The Two Phases of Logistic Regression

train learn $\mathbf{w}$ (a vector of weights, one for each feature) and b (a bias) using **stochastic gradient descent** and **cross-entropy loss**.

test given a test example x, we compute $p(y|x)$ using the learned weights w and b and return the label ($y = 1$ or $y = 0$) that has higher probability.

Slide credit: David Mortensen

Binary classification with logistic regression

Text classification with logistic regression

Given a series of input/output pairs:

$$(x^i, y^i)$$

For each observation x^i

- We represent x^i by a feature vector $[x_1, x_2, \ldots, x_n]$
- We compute an output: a predicted class $y^i \in \{0, 1\}$
  - Get to the predicted class by estimating *p*(*y*|*x*), i.e. *p*(*y*=1|*x*) and *p*(*y*=0|*x*)

For sentiment analysis (classification):

- $y^i = 1$ is positive sentiment, $y^i = 0$ is negative sentiment

Slide adapted from Xiang Lorraine Li

Reminder: the Dot Product

We will see the dot product a lot. It is the **sum** of the element-wise **product** of two vectors of the same dimensionality.

$$\begin{bmatrix} 2 & 7 & 1 \end{bmatrix} \cdot \begin{bmatrix} 8 \\ 2 \\ 8 \end{bmatrix} = 2 \cdot 8 + 7 \cdot 2 + 1 \cdot 8 = 38 \tag{3}$$

Moving on...

Slide credit: David Mortensen

Features in Logistic Regression

For feature x_i, weight w_i tells us how important x_i is

- x_i = "review contains awesome": $w_i = +10$
- x_j = "review contains abysmal": $w_j = -10$
- x_k = "review contains mediocre": $w_k = -2$

Slide credit: David Mortensen

Logistic Regression for One Observation x

input observation feature vector $x = [x_1, x_2, \ldots, x_n]$

weights one per feature $W = [w_1, w_2, \ldots, w_n]$ plus w_0, which is the **bias** b

output a predicted class $\hat{y} \in \{0, 1\}$

Slide credit: David Mortensen

How to Do Classification

For each feature x_i, weight w_i tells us the importance of x_i (and we also have the bias b that shifts where the function crosses the x-axis)

We'll sum up all the weighted features and the bias

$$z = \left(\sum_{i=1}^{n} w_i x_i \right) + b$$

$$z = \mathbf{w} \cdot \mathbf{x} + b$$

Slide credit: David Mortensen

A Most Important Formula

We compute

$$z = \mathbf{w} \cdot \mathbf{x} + b$$

If z is high, we say $y = 1$; if low, then $y = 0$.

orchids A classifiers for cymbidiums should return $y = 1$ when the input is a cymbidium and $y = 0$ otherwise.

sentiment A classifier for positive sentiment show return $y = 1$ when the input has positive sentiment (when the emotions of the writer towards the topic are positive) and $y = 0$ otherwise.

Remember this formula.

Slide credit: David Mortensen

But We Want a Probabilistic Classifier

What does “sum is high” even mean?

Can’t our classifier be like Naive Bayes and give us a probability?

What we really want:

- $p(y = 1|x; \theta)$
- $p(y = 0|x; \theta)$

Where x is a vector of features and $\theta = (w, b)$ (the weights and the bias).

Slide credit: David Mortensen

The Problem: z isn't a Probability!

z is just a number:

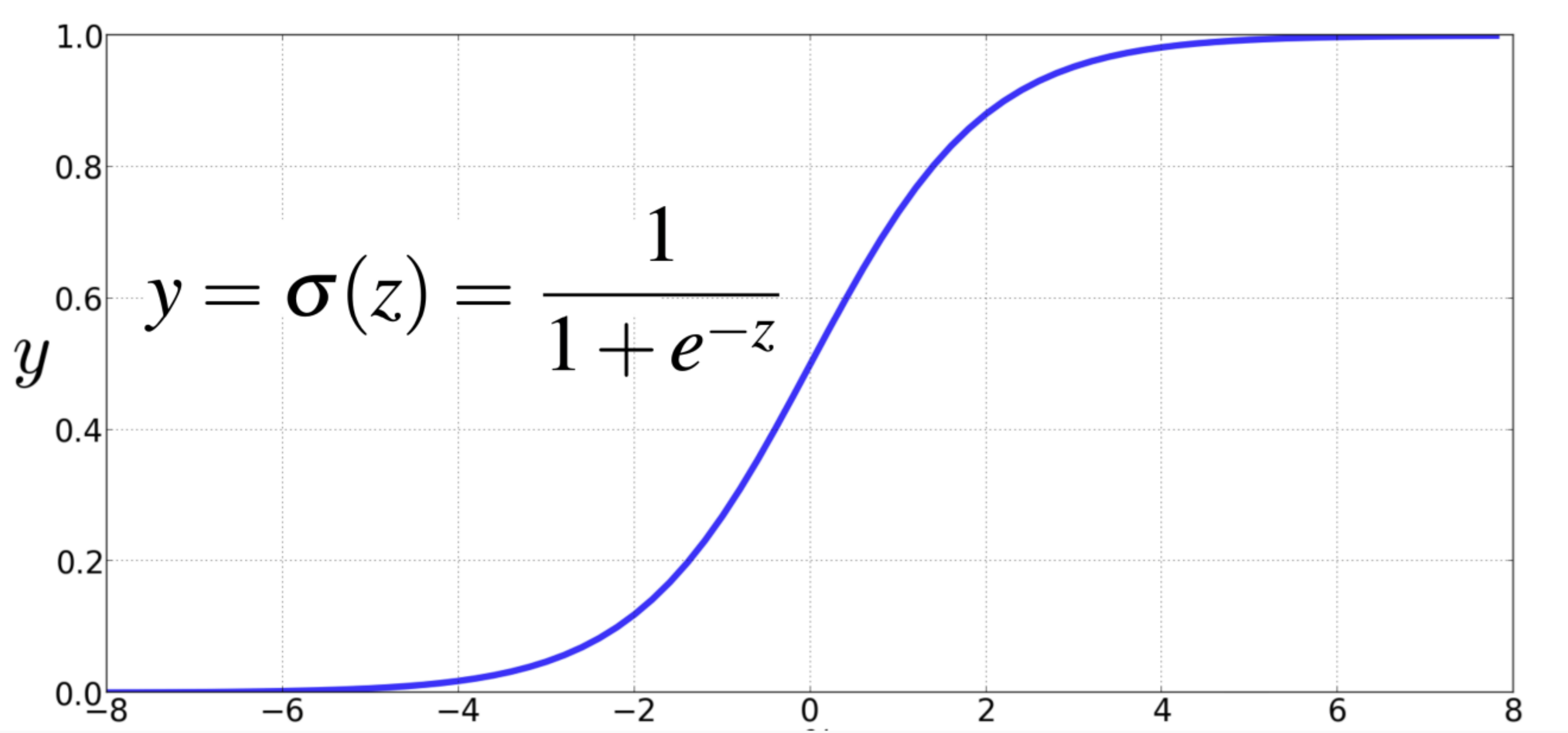
$$z = w \cdot x + b$$

Solution: use a function of z that goes from 0 to 1, like the logistic function or sigmoid function:

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + \exp(-z)}$$

Slide credit: David Mortensen

The Sigmoid Function



Slide credit: David Mortensen

Logistic Regression in Three Easy Steps

1. Compute $w \cdot x + b$
2. Pass it through the sigmoid function: $\sigma(w \cdot x + b)$
3. Treat the result as a probability

Slide credit: David Mortensen

Making Probabilities with Sigmoids

$$
\begin{aligned}
P(y=1) &= \sigma(w \cdot x + b) \\
&= \frac{1}{1 + \exp(-(w \cdot x + b))} \\
P(y=0) &= 1 - \sigma(w \cdot x + b) \\
&= 1 - \frac{1}{1 + \exp(-(w \cdot x + b))} \\
&= \frac{\exp(-(w \cdot x + b))}{1 + \exp(-(w \cdot x + b))}
\end{aligned}
$$

Slide credit: David Mortensen

From Probability to Classification

$$y = \begin{cases} 1 & P(y = 1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

0.5 here is called the **decision boundary**

Slide credit: David Mortensen

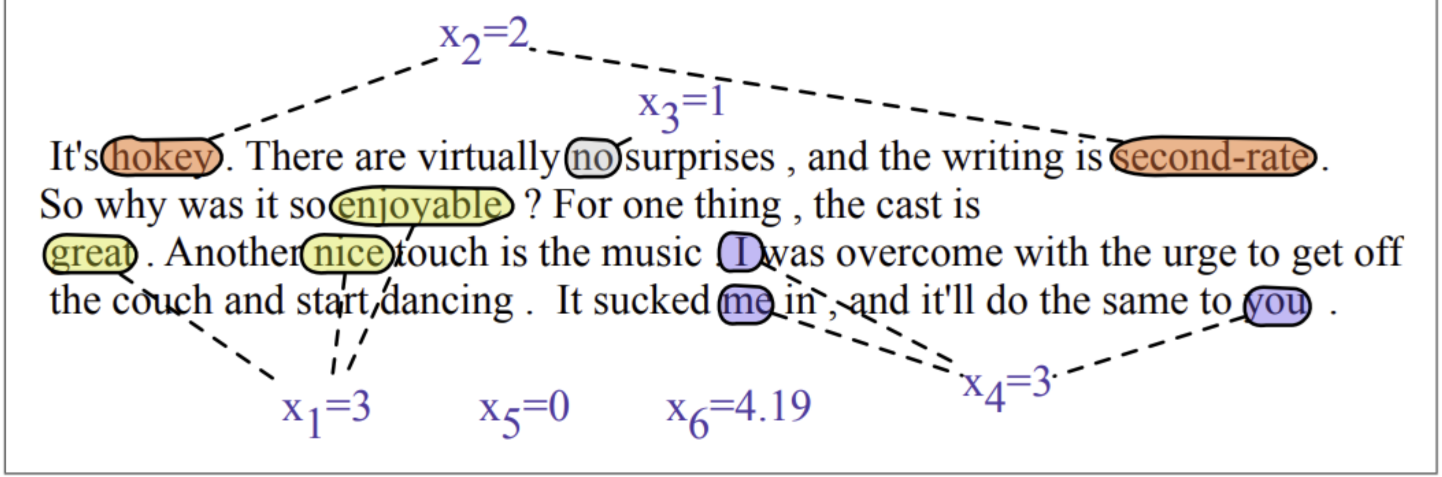
Sentiment Classification: Movie Review

It’s hokey . There are virtually no surprises , and the writing is second-rate . So why was it so enjoyable ? For one thing , the cast is great . Another nice touch is the music . I was overcome with the urge to get off the couch and start dancing . It sucked me in , and it’ll do the same to you .

Slide credit: David Mortensen

Sentiment classification: feature engineering

Var	Definition
x_1	count(positive lexicon words $\in$ doc)
x_2	count(negative lexicon words $\in$ doc)
x_3	$\begin{cases} 1 & \text{if “no”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$
x_4	count(1st and 2nd pronouns $\in$ doc)
x_5	$\begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$
x_6	ln(word count of doc)



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)	$\ln(66) = 4.19$

Slide credit: Jurafsky & Martin

Classifying Sentiment for Input x

Var	Definition	Val
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 & 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)	$\ln(66) = 4.19$

Suppose $w = [2.5, -0.5, -1.2, 0.5, 2.0, 0.7]$ **and** $b = 0.1$

Slide credit: David Mortensen

Performing the calculations

$w = [2.5, -0.5, -1.2, 0.5, 2.0, 0.7]$ **and** $b = 0.1$

Var	Val
x_1	3
x_2	2
x_3	1
x_4	3
x_5	0
x_6	$\ln(66) = 4.19$

$$p(+|x) = P(Y=1|x) \quad = \quad \sigma(w \cdot x + b)$$

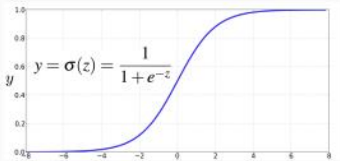
$$p(-|x) = P(Y=0|x) \quad =$$

Slide credit: David Mortensen

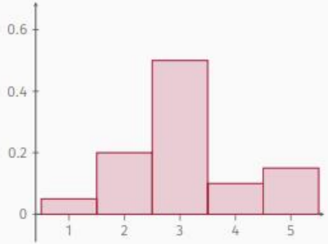
Multinomial logistic regression classification

Softmax is a Generalization of Sigmoid

Sigmoid makes its output look like a probability (forcing it to be between 0.0 and 1.0) and "squashes" it so that the output will tend to 0.0 or 1.0. Concerned about one class? Sigmoid is perfect.



For multiple classes, we do not want a probability—we want a probability **distribution**.

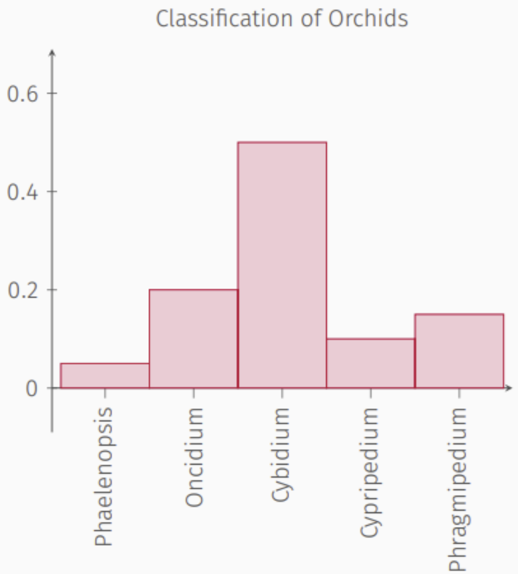


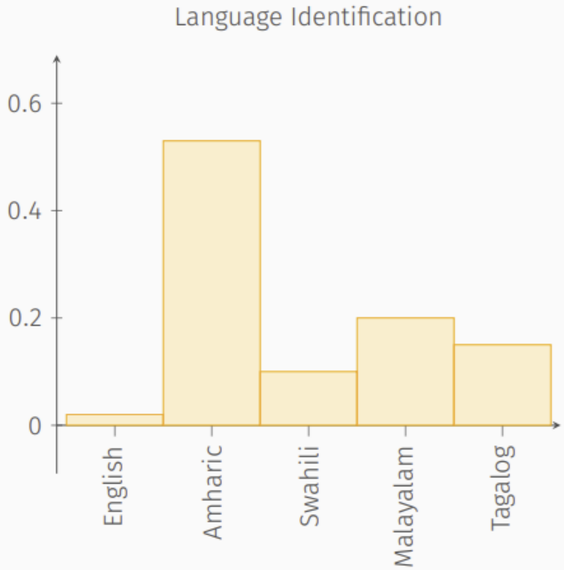
Instead of a sigmoid function, we will use SOFTMAX.

Slide credit: David Mortensen

What is a Probability Distribution?

A probability distribution is a function giving the probabilities that different possible outcomes of an experiment will occur. Our probability distributions will usually be over DISCRETE RANDOM VARIABLES.





Slide credit: David Mortensen

The Softmax Function

The formula for the softmax function is

$$\text{softmax}(\mathbf{z}_i) = \frac{\exp(\mathbf{z}_i)}{\sum_{j=1}^{K} \exp(\mathbf{z}_j)} \quad 1 \leq i \leq K$$

where K is the number of dimensions in the input vector $\mathbf{z}$. Compare it to the formula for the sigmoid function:

$$\hat{y} = \sigma(z) = \frac{1}{1 + \exp(-z)}$$

The formulas are very similar, but sigmoid is a function from a scalar to a scalar, whereas softmax is a function from a vector to a vector.

This output vector is the probability distribution over classes.

Slide credit: David Mortensen

Computing z

Remember that, to compute z in logistic regression, we used the formula

$$z = \mathbf{w}\mathbf{x} + b$$

where $\mathbf{w}$ is a vector of weights, $\mathbf{x}$ is a vector of features, and b is a scalar bias term. Thus, z is a scalar. For multinomial logistic regression, we need a vector $\mathbf{z}$ instead of a scalar z. Our formula will be

$$\mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}$$

where W is a matrix with the shape $[K \times f]$ (where K is the number of output classes and f is the number of input features). In other words, there is an element in $\mathbf{W}$ for each combination of class and feature. $\mathbf{x}$ is a vector of features. $\mathbf{b}$ is a vector of biases (one for each class).

Slide credit: David Mortensen

A Summary Comparison of Logistic Regression and Multinomial Logistic Regression

Logistic regression is

$$\hat{y} = \sigma(\mathbf{w}\mathbf{x} + b)$$

where y is, roughly, a probability.

Multinomial logistic regression (or SOFTMAX REGRESSION) is

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}\mathbf{x} + \mathbf{b})$$

where $\hat{\mathbf{y}}$ is a PROBABILITY DISTRIBUTION over classes, $\mathbf{W}$ is a class $\times$ feature weight matrix, $\mathbf{x}$ is a vector of features, and $\mathbf{b}$ is a vector of biases.

Slide credit: David Mortensen

Coding activity

Notebook: custom features for logistic regression

- Click on this nbgitpuller link
    - Or find the link on the course website
- Open **session10_logistic_regression_features.ipynb**