

BAYSIANS  
AGAINST  
DISCRIMINATION

SUPPORT  
VECTOR  
MACHINES

REPEAL  
POWER  
LAWS

END  
DUALITY  
GAP

FREE  
VARIABLES!

BAN  
GENETIC  
ALGORITHMS

Map Reduce  
Map Reuse  
Map Recycle

CS 2731 / ISSP 2230

# Introduction to Natural Language Processing

Session 6: Logistic regression, part 1

---

Michael Miller Yoder

January 29, 2024

# Course logistics

- [Homework 1](#) **due this Thu Feb 1**
  - Rubric has been posted on Canvas
  - Feel free to ask questions in the Canvas discussion forum, email Bhiman or Michael, or come to office hours
- [Project pre-proposal form](#) is **due Mon Feb 5**
  - Please plan meeting with your groups to discuss project ideas
  - If you don't have any specific ideas, that's fine! We will help you come up with some.
  - Submit 1 form per group through Google Forms. No need to submit anything on Canvas
- Homework 2 will be released tomorrow
  - Due **Thu Feb 15**
  - Text classification

# Lecture overview: Logistic regression part 1

- Code walk-through: Clickbait classification with Naive Bayes in scikit-learn
- Discriminative vs generative classifiers
- Text classification with logistic regression

# Code walk-through: Clickbait classification with Naive Bayes

---

# Clickbait classification activity

- What steps are needed to go from labeled text data to a classifier?
  - Load in data, matched with labels
  - Preprocessing (tokenize, remove punctuation? lowercase?)
  - Extract features (bag of words, etc)
  - Train classifier
  - Test classifier
  - Interpret classifier
- Colab notebook here:  
<https://colab.research.google.com/drive/1QFYJjqkdKDh-OFr3aqdD4OkddYtq3QLB?usp=sharing>
- Data here:  
[https://pitt-my.sharepoint.com/:x:/g/personal/mmy29\\_pitt\\_edu/EfcdzK1ZFL9Dr9ORqvY-O\\_oBATCBFVhLLD5vDxDItbhkhA?e=0H467X&download=1](https://pitt-my.sharepoint.com/:x:/g/personal/mmy29_pitt_edu/EfcdzK1ZFL9Dr9ORqvY-O_oBATCBFVhLLD5vDxDItbhkhA?e=0H467X&download=1)

# Discriminative vs generative classifiers

---

# An Example: Classifying Orchids



*Cymbidium*



*Phalaenopsis*

**Features:** type (**terrestrial, epiphytic**), stem shape (**sympodial, monopodial**), leaf shape (**pointed, rounded**) roots (**thin, thick, tuberous**), flowers (**dorsal sepal, other sepals, petals, labelum, pollinia**)



# Two Options for Classification

## Build up a model of each class

**Cymbidiums** Semi-epiphytic, lithophytic, or terrestrial. Pseudobulb. Flowers in spikes. Bowl-shaped labellum. Glabrous seed capsule.

**Phalaenopsis** Epiphytic. Long, coarse roots. Short, leafy stems. Flowers in racemes or panicles. Spread sepals and petals. Petals much larger than sepals.

## Find features that distinguish them

	Cymbids	Phals
pseudobulbs	T	F
spikes	T	F
panicles	F	T
petals much larger than sepals	F	T

# Should we use generative or discriminative classifiers?

- **Generative:** builds a model of each of the categories, so it could generate instances of them
- **Discriminative:** weights most heavily the features that best discriminate between categories

# Finding the correct class for an orchid

We can classify orchids generatively or discriminatively, where  $\mathbf{x}$  is the orchid and  $\ell$  is the class label

## Naive Bayes

$$\hat{\ell} = \operatorname{argmax}_{\ell \in \mathcal{L}} P(\mathbf{x}|\ell)P(\ell)$$

(compute the likelihood times the prior)

---

## Logistic Regression

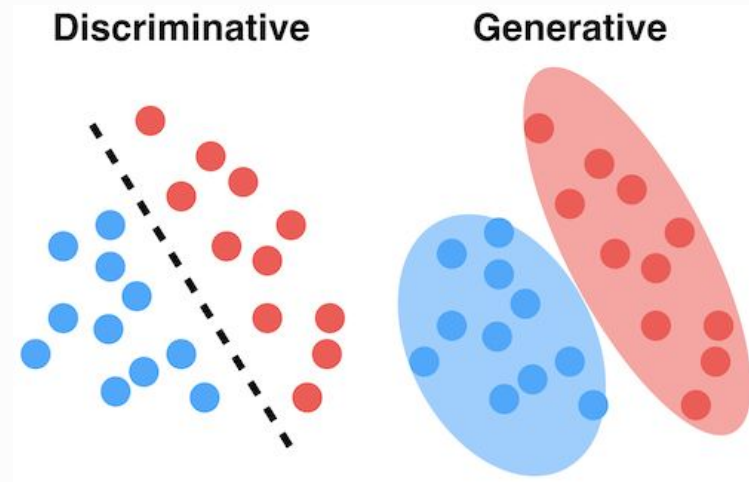
$$\hat{\ell} = \operatorname{argmax}_{\ell \in \mathcal{L}} P(\ell|\mathbf{x})$$

(compute the posterior directly)

We're going to classify documents in the same way!

# Generative vs discriminative classifiers

- Naive Bayes is a generative classifier
- **Generative classifiers** build a model of each class
  - $P(d|c)P(c) = P(c,d)$  joint probability
  - Given an observation, return the class most likely to have generated that observation
- **Discriminative classifiers** instead learn what features are useful to discriminate between possible classes
  - $P(c|d)$  directly model conditional probability



# Generative Classifiers and “Generative AI”



In 2023, journalists and MBAs became very interested in “Generative AI”.

- DALL-E
- ChatGPT
- Stable Diffusion
- Midjourney

These are all generative models—like NB classifiers—in that they can **generate** novel outputs from the same distribution as their inputs. They are different from generative classifiers in that they typically take natural language PROMPTS as input.

# 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
<b>feature representation</b>	represent each observation $\mathbf{x}^{(i)}$ as a <b>vector of features</b> $[x_1, x_2, \dots, x_n]$ , as we did with orchids
<b>classification function</b>	<b>sigmoid function</b> (logistic function)
<b>objective function</b>	cross-entropy loss
<b>optimization function</b>	(stochastic) gradient descent



# 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.

# Classification with logistic regression

---

## 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.

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

Moving on...

# 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**”:  $w_k = -2$

# Logistic Regression for One Observation $x$

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

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

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

# 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

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

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

# A Most Important Formula

We compute

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

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

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

**sentiment** A classifier for positive sentiment should 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.**

# 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).



# The Problem: $z$ isn't a Probability!

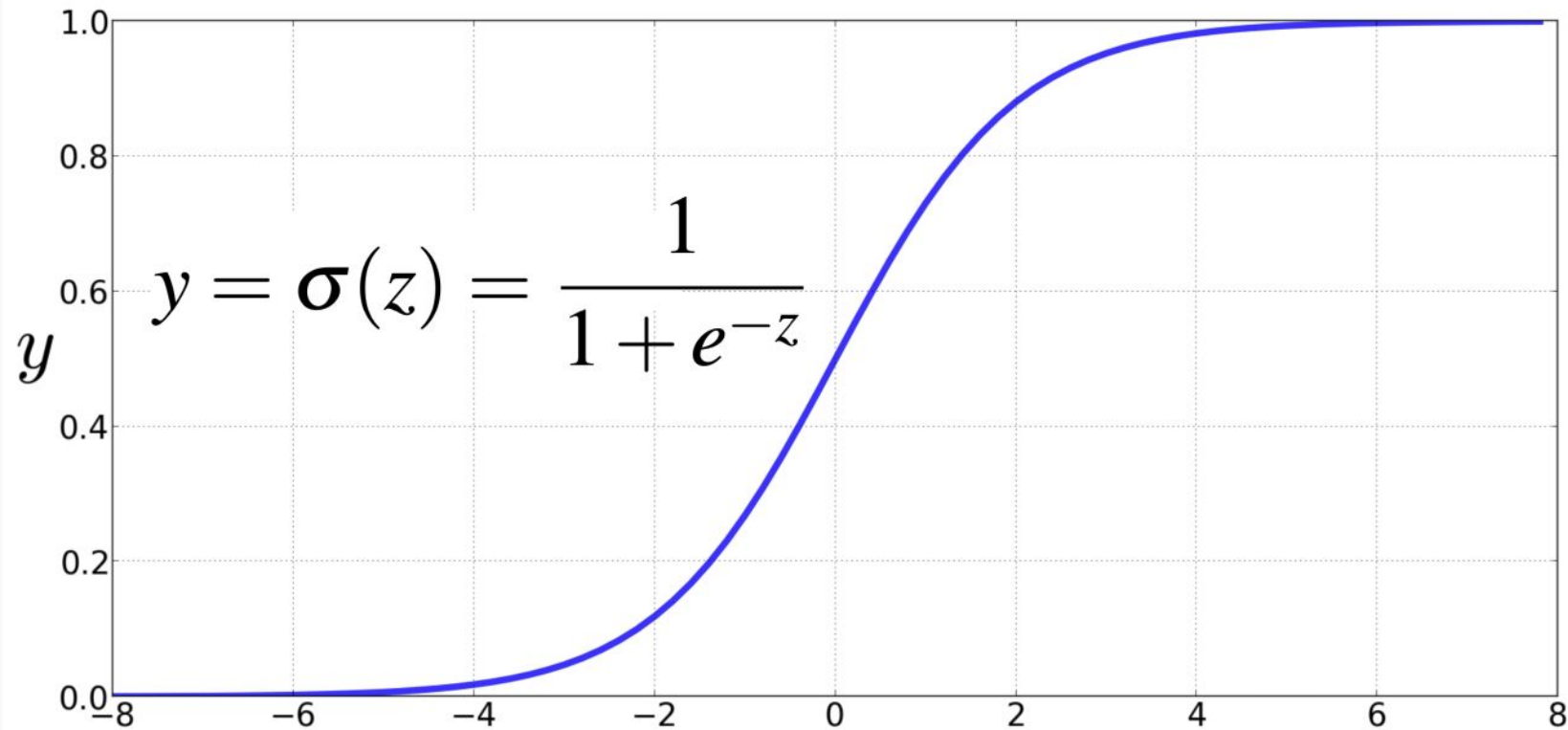
$z$  is just a number:

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

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

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

# The Sigmoid Function



# 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

# Making Probabilities with Sigmoids

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

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

0.5 here is called the **decision boundary**

## 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 .

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**.

Diagram labels:  $x_2=2$ ,  $x_3=1$ ,  $x_1=3$ ,  $x_5=0$ ,  $x_6=4.19$ ,  $x_4=3$

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)	$\ln(66) = 4.19$

# 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
$x_6$	$\log(\text{word count of doc})$	$\ln(66) = 4.19$

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



# Performing the Calculations

$$\begin{aligned} p(+|x) = P(Y = 1|x) &= \sigma(w \cdot x + b) \\ &= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\ &= \sigma(0.833) \\ &= 0.70 \\ p(-|x) = P(Y = 0|x) &= 1 - \sigma(w \cdot x + b) \\ &= 0.30 \end{aligned}$$

# Performing the Calculations

$$\begin{aligned} p(+|x) = P(Y = 1|x) &= \sigma(w \cdot x + b) \\ &= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\ &= \sigma(0.833) \\ &= \mathbf{0.70} \text{ (positive sentiment)} \\ p(-|x) = P(Y = 0|x) &= 1 - \sigma(w \cdot x + b) \\ &= \mathbf{0.30} \end{aligned}$$

*Questions?*

Homework 1 due **next Thu Feb 1**