What do you call a bad dream about machine learning?

A logistic nightmare

CS 2731
Introduction to Natural Language Processing

Session 4: Logistic regression, part 1

Michael Miller Yoder

September 9, 2024

School of Computing and Information

Course logistics

- Homework 1 due next Thu Sep 19
  - Recommended to start early
  - Rubric is on Canvas
  - Feel free to ask questions in the Canvas discussion forum, email Michael or Jayden, or come to office hours
- Project idea submission form will be released on Wed

Lecture overview: Logistic regression part 1

- Text classification
- Input to text classification: features
- Logistic regression
- Classification with logistic regression
- Multinomial logistic regression
- Code walk-through: Pre-processing to prepare for text classification

Text classification

Text classification

"My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield Park is let at last?"

- ROMANCE
- *Pride and Prejudice*
- *DIALOG*

Is this spam?

Subject: **Important notice!**
From: Stanford University <newsforum@stanford.edu>
Date: October 28, 2011 12:34:16 PM PDT
To: undisclosed-recipients:;

Greats News!

You can now access the latest news by using the link below to login to Stanford University News Forum.

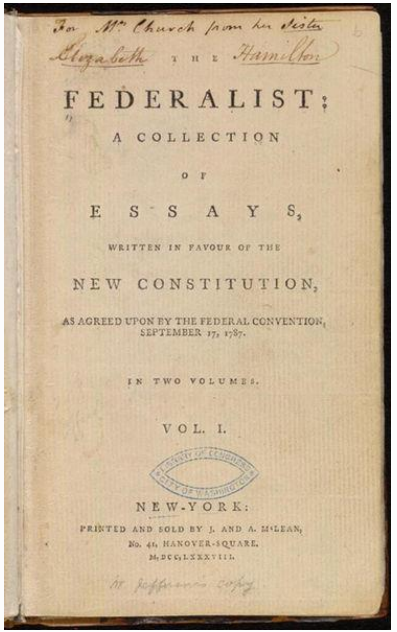
http://www.123contactform.com/contact-form-StanfordNew1-236335.html

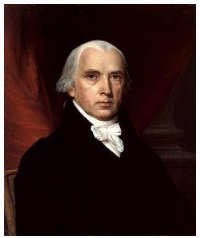
Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information
about the new services.

Slide adapted from Jurafksy & Martin

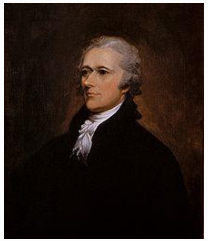
Who wrote which Federalist papers?

- 1787-1788: anonymous essays try to convince New York to ratify U.S Constitution: Jay, Madison, Hamilton.
- Authorship of 12 of the letters in dispute
- 1963: solved by Mosteller and Wallace using Bayesian methods





James Madison



Alexander Hamilton

Slide adapted from Jurafksy & Martin

What is the subject of this medical article?

MEDLINE Article

Syntactic frame and verb bias in aphasia: Plausibility judgments of undergoer-subject sentences

?

MeSH Subject Category Hierarchy

Antagonists and Inhibitors

Blood Supply

Chemistry

Drug Therapy

Embryology

Epidemiology

...

Slide adapted from Jurafksy & Martin

Text Classification

We have a set of documents that we want to *classify* into a small set *classes*.

Applications:

- **Topic classification:** you have a set of news articles that you want to classify as finance, politics, or sports.
- **Sentiment detection:** you have a set of movie reviews that you want to classify as good, bad, or neutral.
- **Language Identification:** you have a set of documents that you want to classify as English, Mandarin, Arabic, or Hindi.
- **Reading level:** you have a set of articles that you want to classify as kindergarten, 1st grade, ...12th grade.
- **Author identification:** you have a set of fictional works that you want to classify as Shakespeare, James Joyce, ...
- **Genre identification:** you have a set of documents that you want to classify as report, editorial, advertisement, blog, ...

Slide credit: David Mortensen

Notation and Setting

- We have a set of n documents (texts) $d_i \in \mathcal{V}^+$, where $\mathcal{V}$ is the vocabulary of the corpus.
  - We assume the texts are segmented already.
- We have set $\mathcal{L}$ of labels, ℓ_i
- Human experts annotate documents with labels and give us $\{(d_1, \ell_1), (d_2, \ell_2), \cdots, (d_n, \ell_n)\}$
- We learn a *classifier* `classify` : $\mathcal{V}^+ \rightarrow \mathcal{L}$ with this labeled training data.
- Afterwards, we use `classify` to classify new documents into their classes.

Slide credit: David Mortensen

Example: Sentiment Detection

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

Slide credit: David Mortensen

Input to classification tasks: features

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

Bag of words document representation

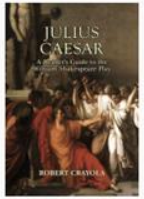
I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

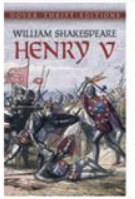
it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Slide from Jurafsky & Martin

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

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

Slide credit: David Mortensen

What Goes into Logistic Regression?

GENERAL	IN LOGISTIC REGRESSION
feature representation	represent each observation $\mathbf{x}^{(i)}$ as a **vector of features** $[x_1, x_2, \ldots x_n]$, as we did with orchids
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 **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

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.

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

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:

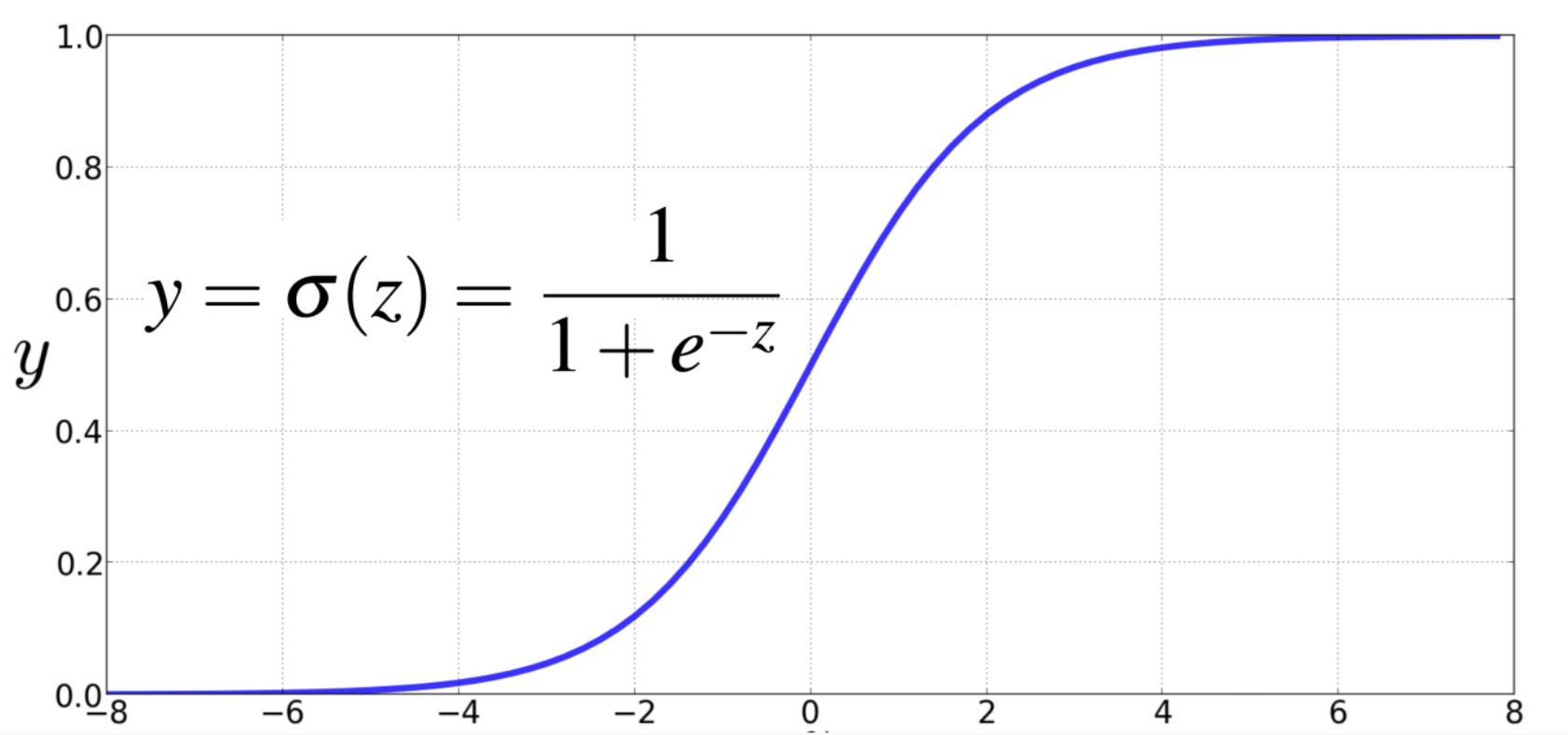
$$z = w \cdot x + b$$

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

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

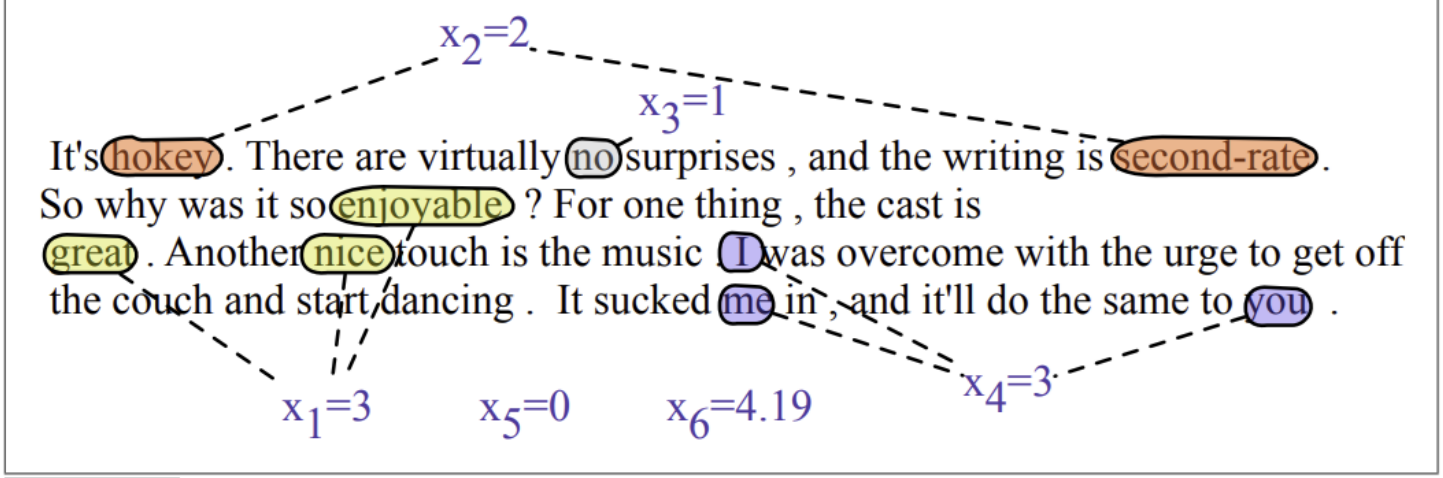
0.5 here is called the **decision boundary**

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



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

$$
\begin{aligned}
p(+|x) = P(Y=1|x) &= \sigma(w \cdot x + b) \\
&= \sigma([2.5, -5.0, -1.2, 0.5.2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\
&= \sigma(0.833) \\
&= 0.70 \\
p(-|x) = P(Y=0|x) &= 1 - \sigma(w \cdot x + b) \\
&= 0.30
\end{aligned}
$$

Slide credit: David Mortensen

Performing the Calculations

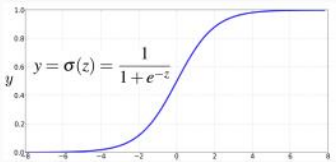
$$
\begin{aligned}
p(+|x) = P(Y = 1|x) &= \sigma(w \cdot x + b) \\
&= \sigma([2.5, -5.0, -1.2, 0.5.2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\
&= \sigma(0.833) \\
&= 0.70 \text{ (positive sentiment)} \\
p(-|x) = P(Y = 0|x) &= 1 - \sigma(w \cdot x + b) \\
&= 0.30
\end{aligned}
$$

Slide credit: David Mortensen

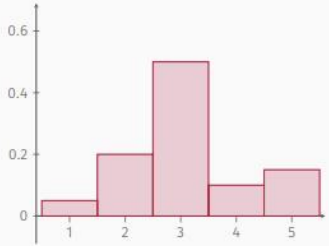
Multinomial logistic regression classification

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**.

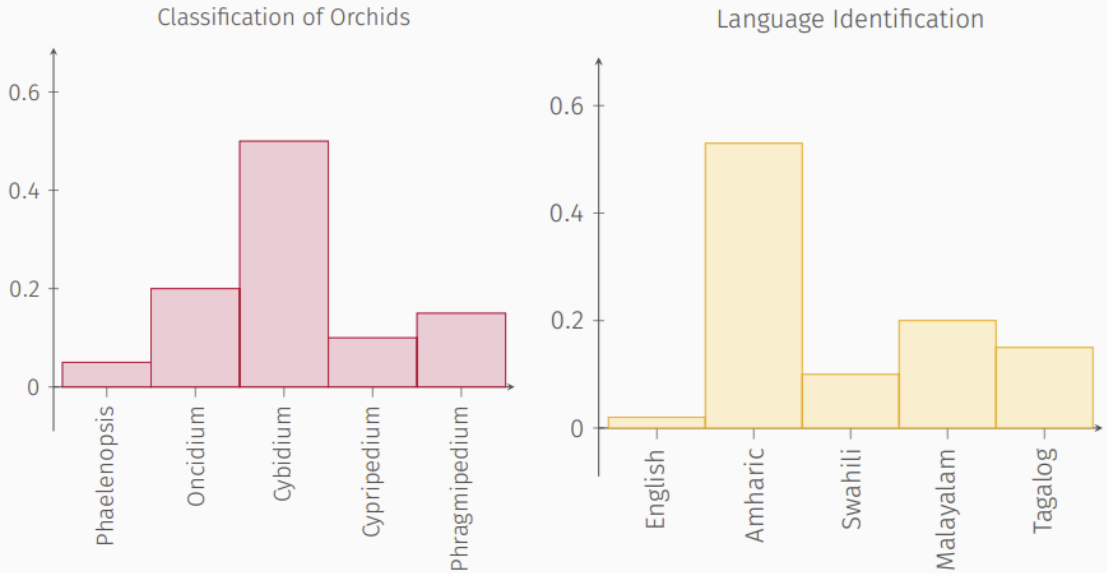


Instead of a sigmoid function, we will use SOFTMAX.

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.

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

Code walk-through:
Preprocessing text data for classification

Clickbait classification activity: preprocessing

- 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: https://colab.research.google.com/drive/1N3-2qK8Bd2Si3__tbGSCJ5YxcVj_BUyw?usp=sharing

Questions?

Homework 1 due next Thu Sep 19