Lexicons as Features for Logistic Regression

Reminder:
Sentiment Lexicons

What is a Lexicon?

- A (usually hand-built) list of words that correspond to some meaning or class
- Possibly with numeric values
- The simplest lexicons just mark sentiment (positive or negative)

The General Inquirer

Philip J. Stone, Dexter C Dunphy, Marshall S. Smith, Daniel M. Ogilvie. 1966. The General Inquirer: A Computer Approach to Content Analysis. MIT Press

Positiv (1915 words)

Negativ (2291 words)

1	Entry	Source	Positiv	Negat
2586	DAKOTA	Lvd		
2587	DAMAGE#1	H4Lvd		Negat
2588	DAMAGE#2	H4Lvd		Negat
2589	DAMN	H4Lvd		Negat
2590	DAMNABLE	H4		Negat
2591	DAMNED	H4		Negat
2592	DAMP	H4Lvd		
2593	DANCE#1	H4Lvd	Positiv	
2594	DANCE#2	H4Lvd	Positiv	
2595	DANCE#3	H4Lvd	Positiv	
2596	DANCER	H4Lvd		
2597	DANGER	H4Lvd		Negat
2598	DANGEROUS	H4Lvd		Negat
2599	DANISH	Lvd		
2600	DARE	H4Lvd	Positiv	
2601	DARING	H4Lvd	Positiv	
2602	DARK	H4Lvd		Negat
2603	DARKEN	H4Lvd		Negat
2604	DARKNESS	H4Lvd		Negat
2605	DARLING	H4Lvd	Positiv	

MPQA Subjectivity Cues Lexicon

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. Proc. of HLT-EMNLP-2005.

Riloff and Wiebe (2003). Learning extraction patterns for subjective expressions. EMNLP-2003.

- 6885 words
- Is a subjective word positive or negative?
  - Strongly or weakly?
- http://mpqa.cs.pitt.edu/lexicons/
- GNU GPL

type=**weaksubj** len=1 word1=**abandoned** pos1=adj stemmed1=n priorpolarity=**negative**

type=weaksubj len=1 word1=**abandonment** pos1=noun stemmed1=n priorpolarity=negative

type=weaksubj len=1 word1=**abandon** pos1=verb stemmed1=y priorpolarity=negative

type=strongsubj len=1 word1=**abase** pos1=verb stemmed1=y priorpolarity=negative

type=**strongsub**j len=1 word1=**abasement** pos1=anypos stemmed1=y priorpolarity=negative

type=strongsubj len=1 word1=**abash** pos1=verb stemmed1=y priorpolarity=negative

type=weaksubj len=1 word1=**abate** pos1=verb stemmed1=y priorpolarity=negative

Words with consistent sentiment across lexicons

Positive	admire, amazing, assure, celebration, charm, eager, enthusiastic, excellent, fancy, tastic, frolic, graceful, happy, joy, luck, majesty, mercy, nice, patience, perfect, pr rejoice, relief, respect, satisfactorily, sensational, super, terrific, thank, vivid, wise, w derful, zest
Negative	abominable, anger, anxious, bad, catastrophe, cheap, complaint, condescending, de defective, disappointment, embarrass, fake, fear, filthy, fool, guilt, hate, idiot, inflict, l miserable, mourn, nervous, objection, pest, plot, reject, scream, silly, terrible, unfrien vile, wicked

Lexicons as Features for Logistic Regression

Sentiment Lexicons

Lexicons as Features for Logistic Regression

Emotion Lexicons

Scherer's typology of affective states

Emotion: relatively brief episode of synchronized response of all or most organismic subsystems in response to the evaluation of an event as being of major significance

angry, sad, joyful, fearful, ashamed, proud, desperate

Mood: diffuse affect state …change in subjective feeling, of low intensity but relatively long duration, often without apparent cause

cheerful, gloomy, irritable, listless, depressed, buoyant

Interpersonal stance: affective stance taken toward another person in a specific interaction, coloring the interpersonal exchange

distant, cold, warm, supportive, contemptuous

Attitudes: relatively enduring, affectively colored beliefs, preferences predispositions towards objects or persons

liking, loving, hating, valuing, desiring

Personality traits: emotionally laden, stable personality dispositions and behavior tendencies, typical for a person

nervous, anxious, reckless, morose, hostile, envious, jealous

Two families of theories of emotion

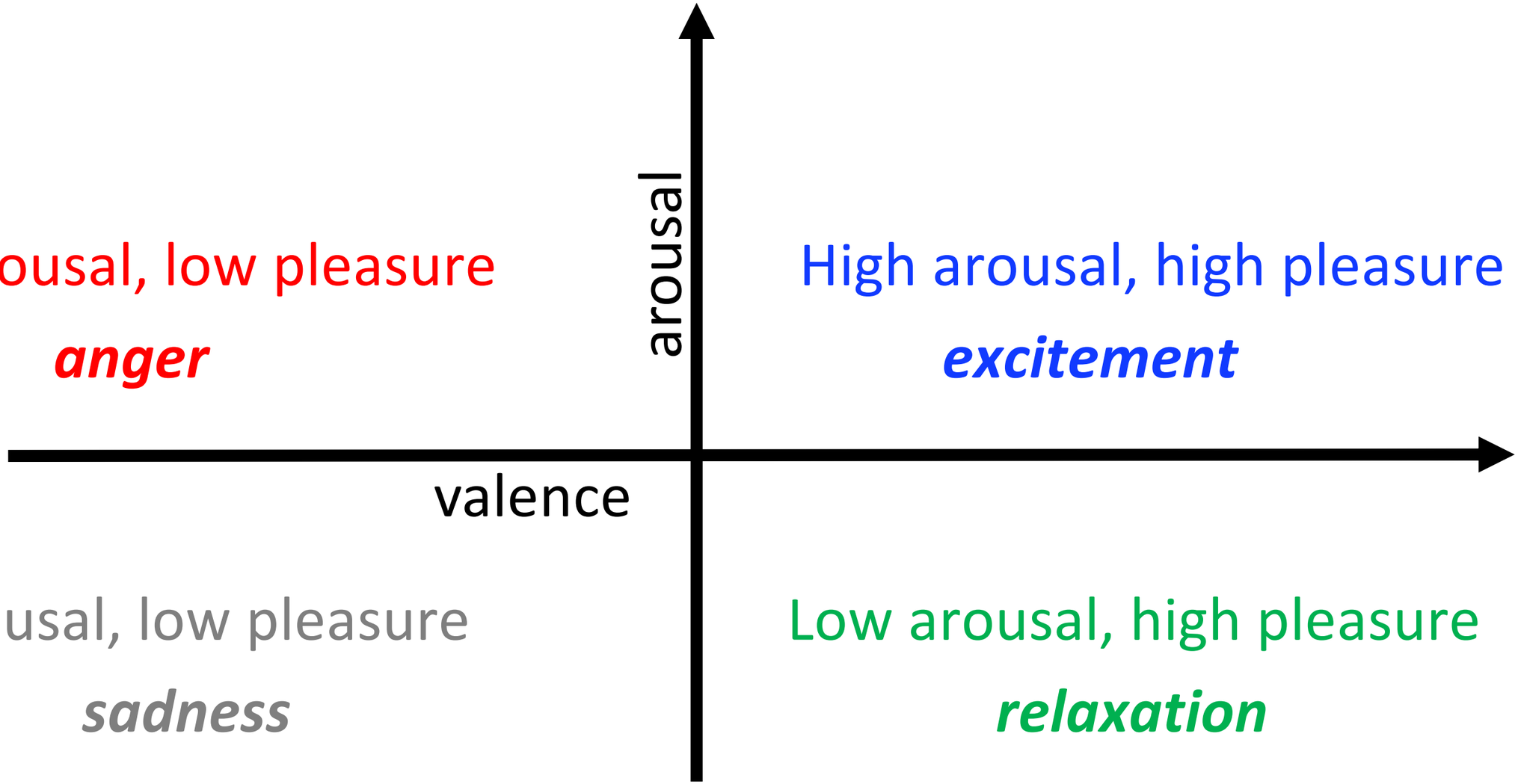
- Atomic basic emotions
  - A finite list of 6 or 8, from which others are generated
- Dimensions of emotion
  - Valence (positive negative)
  - Arousal (strong, weak)
  - Control

Ekman’s 6 basic emotions:

Surprise, happiness, anger, fear, disgust, sadness

Ekman & Matsumoto 1989

Valence/Arousal Dimensions



High arousal, low pleasure

anger

High arousal, high pleasure

excitement

Low arousal, low pleasure

sadness

Low arousal, high pleasure

relaxation

Atomic units vs. Dimensions

Distinctive

- Emotions are units.
- Limited number of basic emotions.
- Basic emotions are innate and universal

Dimensional

- Emotions are dimensions.
- Limited # of labels but unlimited number of emotions.
- Emotions are culturally learned.

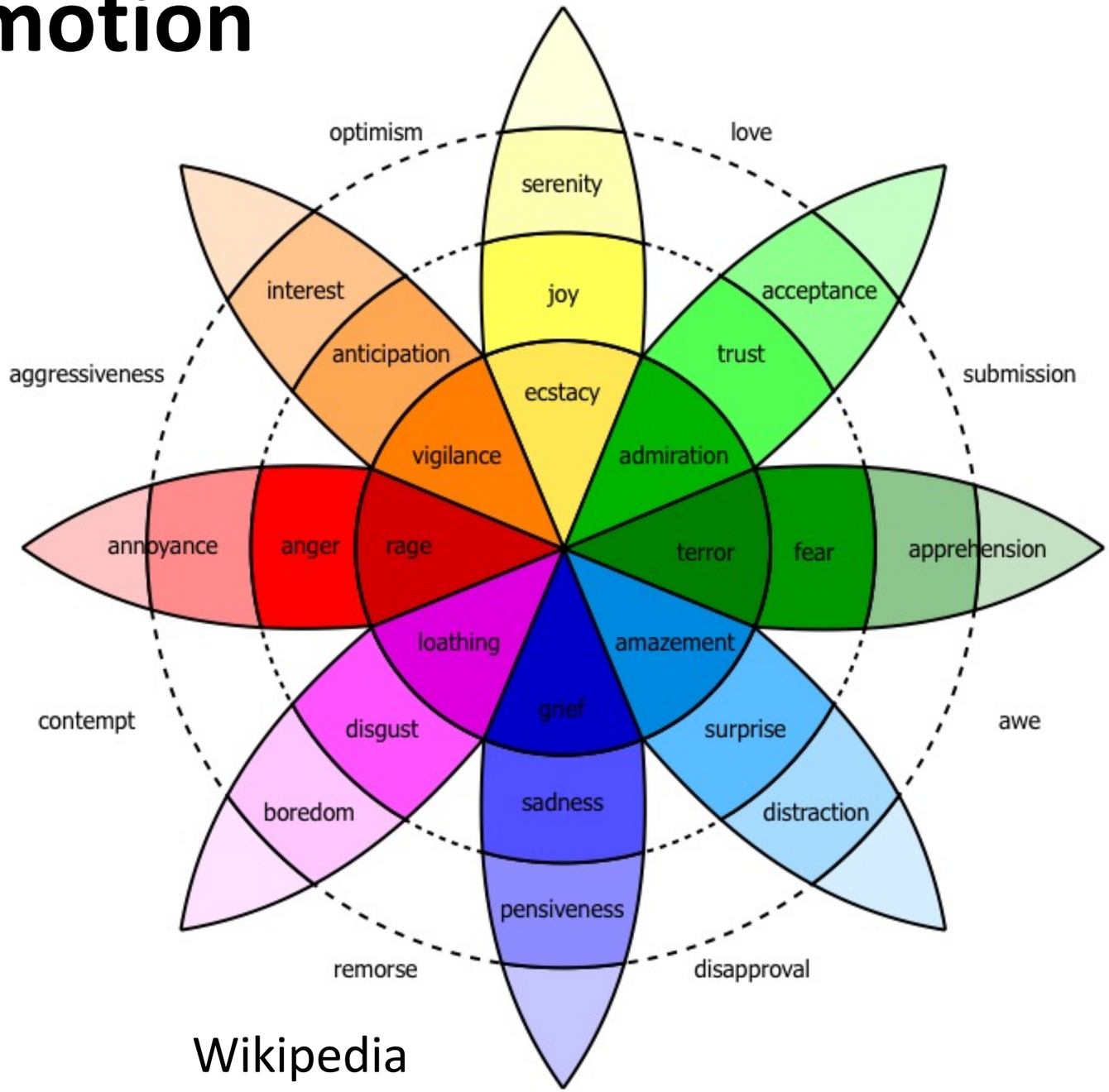
Adapted from Julia Braverman

Let’s look at two emotion lexicons!

1. 8 basic emotions:
   - NRC Word-Emotion Association Lexicon (Mohammad and Turney 2011)
2. Dimensions of valence/arousal/dominance
   - NRC Valence-Arousal-Dominance Lexicon (Mohammad 2018)

Plutchick's wheel of emotion

- 8 basic emotions
- in four opposing pairs:
  - joy–sadness
  - anger–fear
  - trust–disgust
  - anticipation–surprise



Wikipedia

NRC Word-Emotion Association Lexicon

Mohammad and Turney 2011

```
amazingly    anger    0
amazingly    anticipation 0
amazingly    disgust 0
amazingly    fear     0
amazingly    joy      1
amazingly    sadness 0
amazingly    surprise     1
amazingly    trust      0
amazingly    negative     0
amazingly    positive     1
```

More examples

Word	anger	anticipation	disgust	fear	joy	sadness	surprise	trust	positive	negative
reward	0	1	0	0	1	0	1	1	1	0
worry	0	1	0	1	0	1	0	0	0	1
tenderness	0	0	0	0	1	0	0	0	1	0
sweetheart	0	1	0	0	1	1	0	1	1	0
suddenly	0	0	0	0	0	0	1	0	0	0
thirst	0	1	0	0	0	1	1	0	0	0
garbage	0	0	1	0	0	0	0	0	0	1

NRC Emotion/Affect Intensity Lexicon (Mohammad, 2018b); real values for 5814 words

Anger		Fear		Joy		Sadness	
outraged	0.964	horror	0.923	superb	0.864	sad	0.844
violence	0.742	anguish	0.703	cheered	0.773	guilt	0.750
coup	0.578	pestilence	0.625	rainbow	0.531	unkind	0.547
oust	0.484	stressed	0.531	gesture	0.387	difficulties	0.421
suspicious	0.484	failing	0.531	warms	0.391	beggar	0.422
nurture	0.059	confident	0.094	hardship	.031	sing	0.017

Where do lexicons come from?

- Crowdsourcing!!!
- 10,000 words
  - Collected from earlier lexicons
- Labeled by workers on Amazon Mechanical Turk
  - “Turkers”
- 5 Turkers per hit

The AMT Hit

Q4. How much is *startle* associated with the emotion joy? (For example, *happy* and *fun* are strongly associated with joy.)

- *startle* is not associated with joy
- *startle* is weakly associated with joy
- *startle* is moderately associated with joy
- *startle* is strongly associated with joy

Q5. How much is *startle* associated with the emotion sadness? (For example, *failure* and *heart-break* are strongly associated with sadness.)

- *startle* is not associated with sadness
- *startle* is weakly associated with sadness
- *startle* is moderately associated with sadness
- *startle* is strongly associated with sadness

Q6. How much is *startle* associated with the emotion fear? (For example, *horror* and *scary* are strongly associated with fear.)

...

NRC Valence, Arousal, Dominance (VAD) lexicon

Mohammad (2018)

20,000 words, 3 emotional dimensions:

- **valence** (the pleasantness of the stimulus)
- **arousal** (the intensity of emotion provoked by the stimulus)
- **dominance** (the degree of control exerted by the stimulus)

Best-worst scaling: valence

Q1. Which of the four words below is associated with the MOST happiness / pleasure / positiveness / satisfaction / contentedness / hopefulness OR LEAST unhappiness / annoyance / negativeness / dissatisfaction / melancholy / despair?

vacation, consolation, whistle, torture

Q2. Which of the four words below is associated with the LEAST happiness / pleasure / positiveness / satisfaction / contentedness / hopefulness OR MOST unhappiness / annoyance / negativeness / dissatisfaction / melancholy / despair?

Lexicon of valence, arousal, and dominance

Valence		Arousal		Dominance	
delightful	.918	enraged	.962	powerful	.991
vacation	.840	party	.840	authority	.935
whistle	.653	organized	.337	saxophone	.482
consolation	.408	effortless	.120	discouraged	.0090
torture	.115	napping	.046	weak	.045

Issues to keep in mind with crowdsourcing lexicons

- Native (or very fluent) speakers
- Making the task clear for non-linguists or non computer scientists
- Paying minimum wage (see Michael Bernstein's fairwork.stanford.edu)
- See Michael's CS376 for more on crowdsourcing!

Lexicons as features for logistic regression

Other Useful Lexicons

LIWC: Linguistic Inquiry and Word Count

Positive Emotion	Negative Emotion	Insight	Inhibition	Family	Negate
appreciat*	anger*	aware*	avoid*	brother*	aren't
comfort*	bore*	believe	careful*	cousin*	cannot
great	cry	decid*	hesitat*	daughter*	didn't
happy	despair*	feel	limit*	family	neither
interest	fail*	figur*	oppos*	father*	never
joy*	fear	know	prevent*	grandf*	no
perfect*	griev*	knew	reluctan*	grandm*	nobod*
please*	hate*	means	safe*	husband	none
safe*	panic*	notice*	stop	mom	nor
terrific	suffers	recogni*	stubborn*	mother	nothing
value	terrify	sense	wait	niece*	nowhere
wow*	violent*	think	wary	wife	without

LIWC (Linguistic Inquiry and Word Count)

Pennebaker, J.W., Booth, R.J., & Francis, M.E. (2007). Linguistic Inquiry and Word Count: LIWC 2007. Austin, TX

http://www.liwc.net/

2300 words

>70 classes

The General Inquirer

Philip J. Stone, Dexter C Dunphy, Marshall S. Smith, Daniel M. Ogilvie. 1966. The General Inquirer: A Computer Approach to Content Analysis. MIT Press

- Home page: http://www.wjh.harvard.edu/~inquirer
- List of Categories: http://www.wjh.harvard.edu/~inquirer/homecat.htm
- Spreadsheet: http://www.wjh.harvard.edu/~inquirer/inquirerbasic.xls
- Categories:
  - Positiv (1915 words) and Negativ (2291 words)
  - Strong vs Weak, Active vs Passive, Overstated versus Understated
  - **Pleasure, Pain, Virtue, Vice, Motivation, Cognitive Orientation, etc**
- Free for Research Use

Concreteness versus abstractness

- The degree to which the concept denoted by a word refers to a perceptible entity.
- Brysbaert, M., Warriner, A. B., and Kuperman, V. (2014) Concreteness ratings for 40 thousand generally known English word lemmas *Behavior Research Methods 46, 904-911.*
- Supplementary data: This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License.
- 37,058 English words and 2,896 two-word expressions (“zebra crossing” and “zoom in”),
- Rating from 1 (abstract) to 5 (concrete)

Concreteness versus abstractness

- Brysbaert, M., Warriner, A. B., and Kuperman, V. (2014) Concreteness ratings for 40 thousand generally known English word lemmas *Behavior Research Methods 46, 904-911.*
- Supplementary data: This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License.
- Some example ratings from the final dataset of 40,000 words and phrases

```
banana 5
bathrobe 5
bagel 5
brisk 2.5
badass 2.5
basically 1.32
belief 1.19
although 1.07
```

Lexicons as features for logistic regression

Using the lexicons to detect affect

Lexicons for detecting document affect: Simplest unsupervised method

- Sentiment:
  - Sum the weights of each positive word in the document
  - Sum the weights of each negative word in the document
  - Choose whichever value (positive or negative) has higher sum
- Emotion:
  - Do the same for each emotion lexicon

Lexicons for detecting document affect: Simplest unsupervised method

$$f^{+} = \sum_{w \text{ s.t. } w \in positivelexicon} \theta_w^{+} count(w)$$

$$f^{-} = \sum_{w \text{ s.t. } w \in negativelexicon} \theta_w^{-} count(w)$$

```
Sentiment = +    if    f+ > f-
```

Lexicons for detecting document affect: Slightly more complex unsupervised method

$$f^+ = \sum_{w \text{ s.t. } w \in positivelexicon} \theta_w^+ count(w)$$

$$f^- = \sum_{w \text{ s.t. } w \in negativelexicon} \theta_w^- count(w)$$

$$sentiment = \begin{cases} + & \text{if } \frac{f^+}{f^-} > \lambda \\ - & \text{if } \frac{f^-}{f^+} > \lambda \\ 0 & \text{otherwise.} \end{cases}$$

Lexicons for detecting document affect: Simplest supervised method

- Use the lexicons as **features** for a classifier
- Given a training set
  - Each observation has a label (review X has sentiment Y)
  - Assign features to each observation
  - Use “counts of lexicon categories” as a features
    - NRC Emotion category “anticipation” had count of 2
      - 2 words in this document were in “anticipation” lexicon
    - LIWC category “cognition” had count of 7

Lexicons for detecting document affect: Simplest supervised method

- Baseline
  - Use counts of **all** the words and bigrams in the training set
    - Like the naïve bayes algorithm
  - **This is hard to beat**
  - But “using all the words” only works if the training and test sets are very similar
  - In real life, sometimes the test set is very different
    - Lexicons are useful in that situation

Lexicons as features for logistic regression

Using the lexicons to detect affect

CS 124/LINGUIST 180
From Languages to Information

Logistic Regression

Logistic Regression

- Important analytic tool in natural and social sciences
- Baseline supervised machine learning tool for classification
- Is also the foundation of a neural network

CS 124/LINGUIST 180
From Languages to Information

Classification in Logistic Regression

Classification Reminder

- Positive/negative sentiment
- Spam/not spam
- Authorship attribution (Hamilton or Madison?)

Alexander Hamilto

Text Classification: definition

- *Input*:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

- *Output*: a predicted class $c \in C$

Binary Classification in 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 $\hat{y}^i \in \{0,1\}$

Examples of inputs and features: spam

- X is an email
  - X_1 is “count of the word *sale* in the email
  - X_2 is “count of mentions of drugs”
  - X_3 is “number of URLS in the email”
  - X_4 is “length of email in bytes”
  - X_5 is “uses phrase ‘Prestigious Non-Accredited Universities’

Features in Classification

“Poirot,” I cried, “where are you?”

“I am here, my friend.”

He had stepped outside the French window, and was standing, apparently lost in admiration, before the various shaped flower beds.

“Admirable!” he murmured. “Admirable! What symmetry! Observe that crescent; and those diamonds—their neatness rejoices the eye. The spacing of the plants, also, is perfect. It has been recently done; is it not so?”

“Yes, I believe they were at it yesterday afternoon. But come in—Dorcas is here.”

“*Eh bien, eh bien!* Do not grudge me a moment’s satisfaction of the eye.”

“Yes, but this affair is more important.”

“And how do you know that these fine begonias are not of equal importance?”

Features in Classification

[later on]…

“We are most grateful to Monsieur Poirot for elucidating the matter. But for him, we should never have known of this will. I suppose, I may not ask you, monsieur, what first led you to suspect the fact?”

Poirot smiled and answered:

“A scribbled over old envelope, and a freshly planted bed of begonias.”

Features in logistic regression

- For each feature x_i, we'll have a weight w_i
- Weight w_i tells us how important feature x_i is the classification

  x_i =”1 if review contains ‘awesome’": w_i very positive +10

  x_j =”1 if review contains ‘abysmal’": w_j very negative -10

  x_k =“1 if review contains ‘mediocre’": w_k a little negative -2

Logistic Regression for one observation x

- Input observation: vector $x = [x_1, x_2, \ldots, x_n]$
- Weights: one per feature: $W = [w_1, w_2, \ldots, w_n]$
  - Sometimes we call the weights $\theta = [\theta_1, \theta_2, \ldots, \theta_n]$
- Output: a predicted class $\hat{y} \in \{0,1\}$

(multinomial logistic regression: $y \in \{0, 1, 2, 3, 4\}$)

How to do classification

- For each feature x_i, weight w_i tells us importance of x_i
  - (Plus we'll have a bias b)
- We'll sum up all the weighted features and the bias

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

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

- If this sum is high, we say y=1; if low, then y=0

But we want a probabilistic classifier

- We need to formalize “sum is high”.
- We’d like a principled classifier that gives us a probability, just like Naive Bayes did
- We want a model that can tell us:

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

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

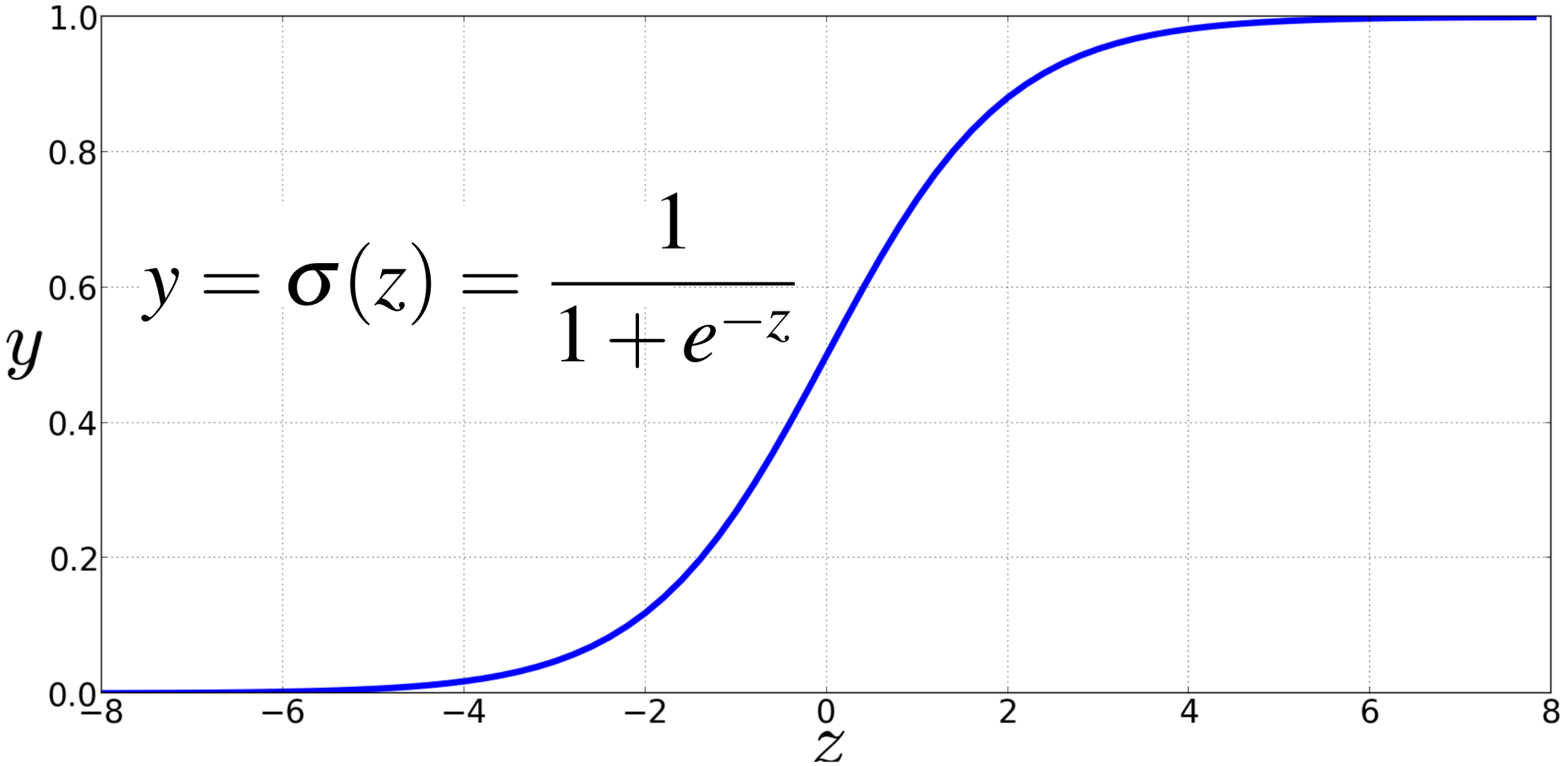
The problem: z isn't a probability, it's just a number!

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

- Solution: use a function of z that goes from 0 to 1

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

The very useful sigmoid or logistic function



Idea of logistic regression

- We'll compute $w \cdot x + b$
- And then we'll pass it through the sigmoid function:
- $\sigma(w \cdot x + b)$
- And we'll just treat it as a probability

Making probabilities with sigmoids

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

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

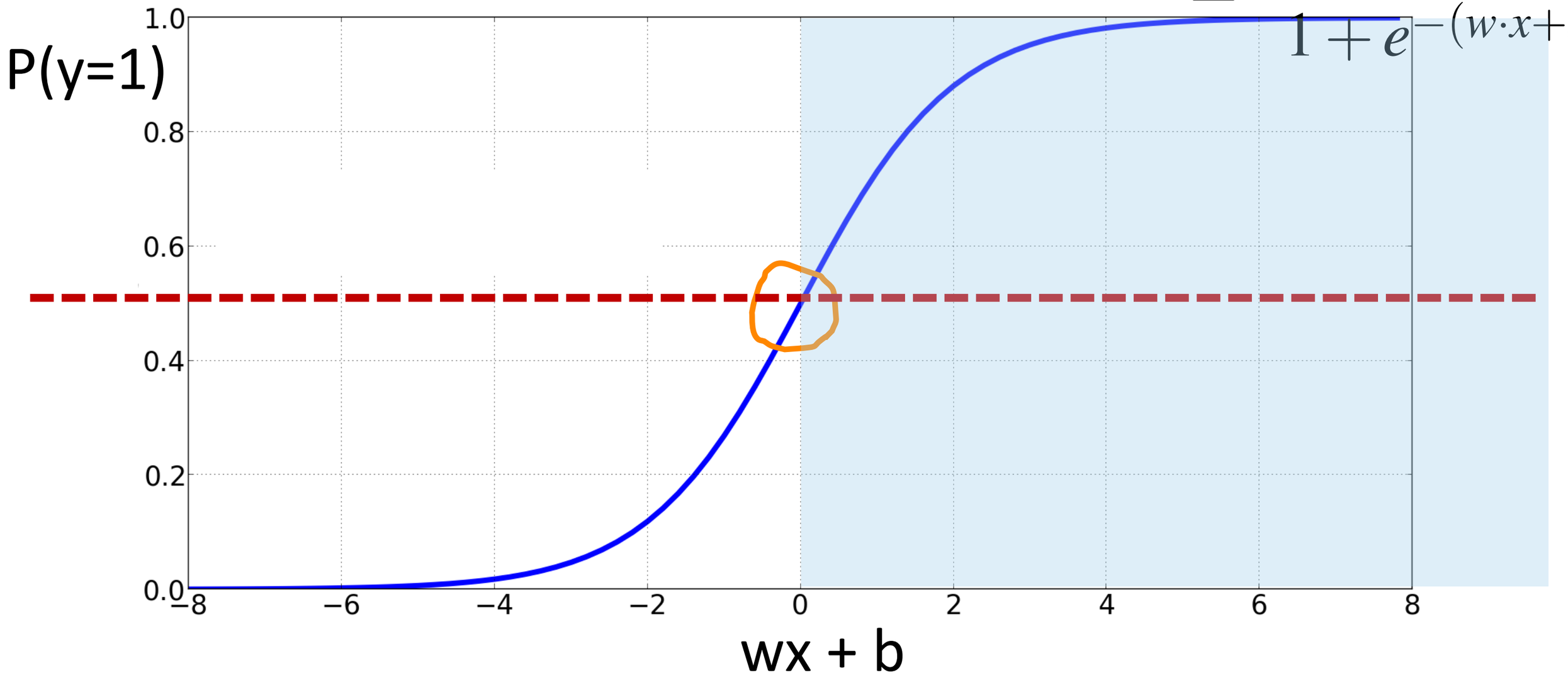
Turning a probability into a classifier: the decision boundary

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

The probabilistic classifier

$$P(y=1) = \sigma(w \cdot x + b)$$

$$= \frac{1}{1+e^{-(w \cdot x + b)}}$$



Turning a probability into a classifier: the decision boundary

$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|x) > 0.5 \quad \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \quad \text{if } w \cdot x + b \leq 0 \end{cases}$$

CS 124/LINGUIST 180
From Languages to Information

Classification in Logistic Regression

CS 124/LINGUIST 180
From Languages to Information

Logistic Regression: A text example

Sentiment example: does y=1 or y=0?

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.

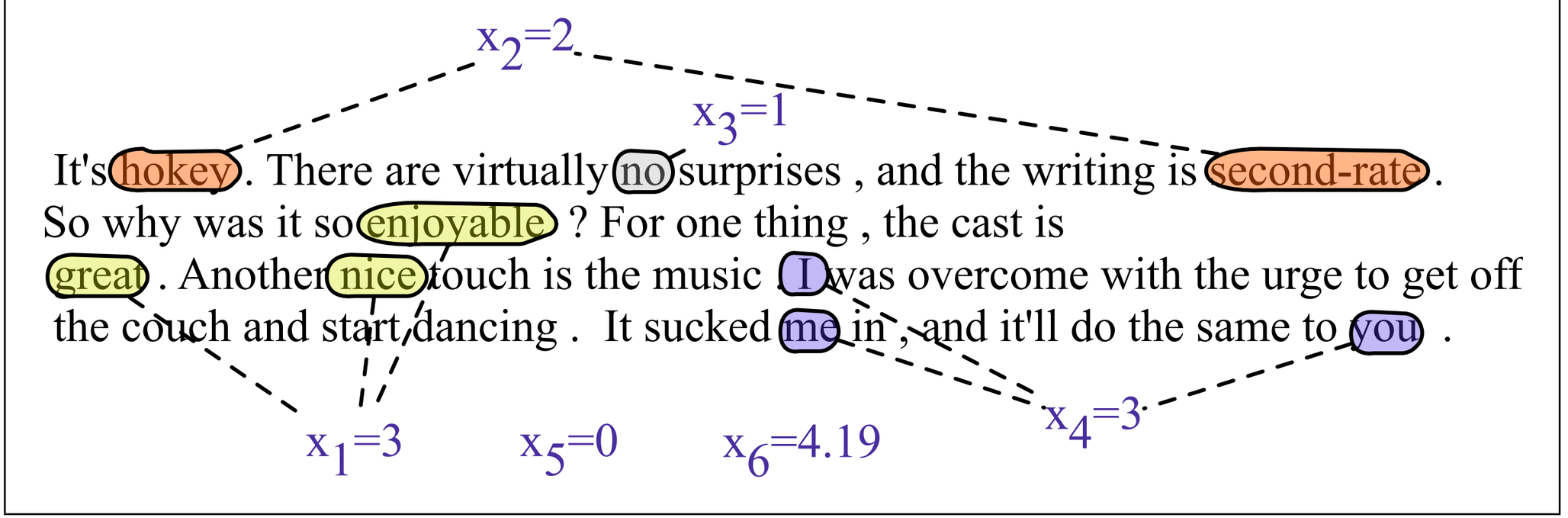
It's hokey, there are virtually no surprises, and the writing is second-rate. So why was it so enjoyable? For one thing, the cast is great. Another nice touch is the music. I was overcome with the urge to get off the couch and start dancing. It sucked me in, and it'll do the same to you.

Positive sentiment words

Negative sentiment words

"no"

i/me/mine/you/your/yours



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	$\begin{cases} 1 & \text{if } \text{“no”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if } \text{“!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	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	$\begin{cases} 1 & \text{if } \text{“no”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if } \text{“!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

5.2

Suppose $w = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7]$

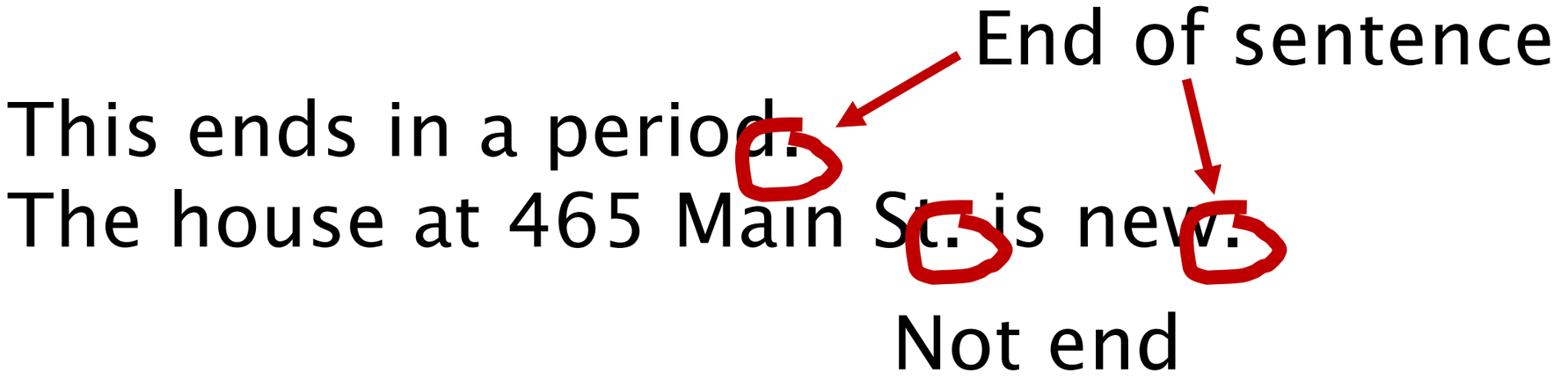
$b = 0.1$

Classifying sentiment for input x

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

$$
\begin{aligned}
p(-|x) = P(Y=0|x) &= 1 - \sigma(w \cdot x + b) \\
&= 0.30
\end{aligned}
$$

We can build features for logistic regression for any classification task: period disambiguation



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

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

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

CS 124/LINGUIST 180
From Languages to Information

Logistic Regression: A text example

CS 124/LINGUIST 180
From Languages to Information

Dan Jurafsky

Stanford University

Logistic Regression: The Whole Picture

Classification in (**binary**) logistic regression: summary

Given:

- a set of classes: (+ sentiment,- sentiment)
- a vector **x** of features `[x1, x2, …, xn]`
  - x1= count("awesome")
  - x2 = log(number of words in review)
- A vector **w** of weights `[w1, w2, …, wn]`
  - w_i for each feature f_i

$$P(y=1) = \sigma(w \cdot x + b)$$

$$= \frac{1}{1+e^{-(w \cdot x + b)}}$$

Wait, where did the W's come from?

Training: given a training set of M observations (example x, class y)

- Learn the parameters of the model

Test: Given a test example *x* and class $y \in \{0,1\}$

- return the higher probability class

Components of a probabilistic (supervised) machine learning classifier

A **corpus** of M observation input/output pairs, $(x^{(i)},y^{(i)})$

For each input observation $x^{(i)}$

- a vector of **features** $[x_1, x_2, ..., x_n]$

A **classification function** computing $\hat{y}$, via $p(y|x)$

- *sigmoid*
- *softmax*

For learning

- A **loss function** (cross-entropy loss)
- An **optimization algorithm** (stochastic gradient descent)

For those interested in SGD

Workhorse for training logistic regression and neural networks

See the rest of chapter 5!

Plus :

- CS221
- CS229
- CS224N
- CS230
- Etc etc

CS 124/LINGUIST 180
From Languages to Information

Dan Jurafsky

Stanford University

Logisitic Regression and Naive Bayes

Generative and Discriminative Classifiers

- Naïve Bayes is a **generative** classifier

- by contrast:

- Logistic regression is a **discriminative** classifier

Generative and Discriminative Classifiers

Suppose we're distinguishing cat from dog images

imagenet

imagenet

Generative Classifier:

- Build a model of what's in a cat image
  - Knows about whiskers, ears, eyes
  - Assigns a probability to any image:
    - how cat-y is this image?

Also build a model for dog images

Now given a new image:

Run both models and see which one fits better

Discriminative Classifier

Just try to distinguish dogs from cats

Oh look, dogs have collars!

Let's ignore everything else

Finding the correct class c from a document d in Generative vs Discriminative Classifiers

- Naive Bayes

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

- Logistic Regression

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(c|d)}^{\text{posterior}}$$

CS 124/LINGUIST 180
From Languages to Information

Dan Jurafsky

Stanford University

Logisitic Regression and Naive Bayes

CS 124/LINGUIST 180
From Languages to Information

Dan Jurafsky

Stanford University

Multinomial Logistic Regression

Multinomial Logistic Regression

Often we need more than 2 classes

- Positive/negative/neutral
- Parts of speech (noun, verb, adjective, adverb, preposition, etc.)
- Classify emergency SMSs into different actionable classes

If >2 classes we use **multinomial logistic regression**

- "logistic regression" will just mean binary (2 output classes)

  = Softmax regression

  = Maximum entropy modeling

  = Maxent

  = Multinomial logit

Multinomial Logistic Regression

The probability of everything must still sum to 1

$$P(\text{positive}|\text{doc}) + P(\text{negative}|\text{doc}) + P(\text{neutral}|\text{doc}) = 1$$

Need a generalization of the sigmoid called the **softmax**

- Takes a vector $z = [z1, z2, ..., zk]$ of k arbitrary values
- Outputs a probability distribution
  - each value in the range [0,1]
  - all the values summing to 1

The **softmax** function

Turns a vector $z = [z_1, z_2, \ldots, z_k]$ of k arbitrary values into probabilities

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \quad 1 \leq i \leq k$$

The denominator $\sum_{i=1}^{k} e^{z_i}$ is used to normalize all the values into probabilities.

$$\text{softmax}(z) = \left[\frac{e^{z_1}}{\sum_{i=1}^{k} e^{z_i}}, \frac{e^{z_2}}{\sum_{i=1}^{k} e^{z_i}}, \ldots, \frac{e^{z_k}}{\sum_{i=1}^{k} e^{z_i}} \right]$$

The **softmax** classifier

- Turns a vector $z = [z_1, z_2, \ldots, z_k]$ of k arbitrary values into probabilities

$$\text{softmax}(z) = \left[\frac{e^{z_1}}{\sum_{i=1}^{k} e^{z_i}}, \frac{e^{z_2}}{\sum_{i=1}^{k} e^{z_i}}, \ldots, \frac{e^{z_k}}{\sum_{i=1}^{k} e^{z_i}} \right]$$

$$z = [0.6, 1.1, -1.5, 1.2, 3.2, -1.1]$$

$$[0.055, 0.090, 0.0067, 0.10, 0.74, 0.010]$$

Softmax in multinomial logistic regression

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

Features in (binary) logistic regression

$$f_1 = \begin{cases} 1 & \text{if "!" in doc} \\ 0 & \text{otherwise} \end{cases}$$

$$w_1 = 2.5$$

Features in softmax regression
distinct weights for each value!

Var	Definition	Wt
$f_1(0,x)$	$\begin{cases} 1 & \text{if } \text{“!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	-4.5
$f_1(+,x)$	$\begin{cases} 1 & \text{if } \text{“!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	2.6
$f_1(\text{-},x)$	$\begin{cases} 1 & \text{if } \text{“!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1.3

Binary versus multinomial logistic regression

Multinomial is obviously applicable to many more classification tasks

- Softmax is important for neural net classifiers

Binary logistic regression is simpler to model

We'll stick to binary logistic regression in PA3

CS 124/LINGUIST 180
From Languages to Information

Dan Jurafsky

Stanford University

Multinomial Logistic Regression

CS 124/LINGUIST 180
From Languages to Information

Dan Jurafsky

Stanford University

Summary: Logistic Regression

Classification in (binary) logistic regression: summary

- Given:
  - a set of classes: (+ sentiment,- sentiment)
  - a vector **x** of features `[x1, x2, …, xn]`
    - x1= count("awesome")
    - x2 = log(number of words in review)
  - A vector **w** of weights `[w1, w2, …, wn]`
    - w_i for each feature f_i

$$P(y=1) = \sigma(w \cdot x + b)$$
$$= \frac{1}{1+e^{-(w \cdot x + b)}}$$

Components of a probabilistic (supervised) machine learning classifier

- A **corpus** of M observation input/output pairs, $(x^{(i)},y^{(i)})$
- For each input observation $x^{(i)}$
  - a vector of **features** $[x_1, x_2, ..., x_n]$
- A **classification function** computing $\hat{y}$, via $p(y|x)$
  - *sigmoid*
  - *softmax*
- For learning
  - A **loss function** (cross-entropy loss)
  - An **optimization algorithm** (stochastic gradient descent)

Last point: Overfitting

+

This movie drew me in, and it'll do the same to you.

-

I can't tell you how much I hated this movie. It sucked.

Useful or harmless features

X1 = "this"
X2 = "movie
X3 = "hated"

X4 = "drew me in"

4gram features that just "memorize" training set and might cause problems

X5 = "the same to you"
X7 = "tell you how much"

Overfitting

4-gram model on tiny data will just memorize the data

- 100% accuracy on the training set

But be surprised by the novel 4-grams in the test data

- Low accuracy on test set

Models that are too powerful can **overfit** the data

- Fitting the details of the training data so exactly that the model doesn't generalize well to the test set
  - There are ways to avoid overfitting
    - Regularization in logistic regression (see chapter 5)
    - Dropout in neural networks

CS 124/LINGUIST 180
From Languages to Information

Dan Jurafsky

Stanford University

Summary: Logistic Regression