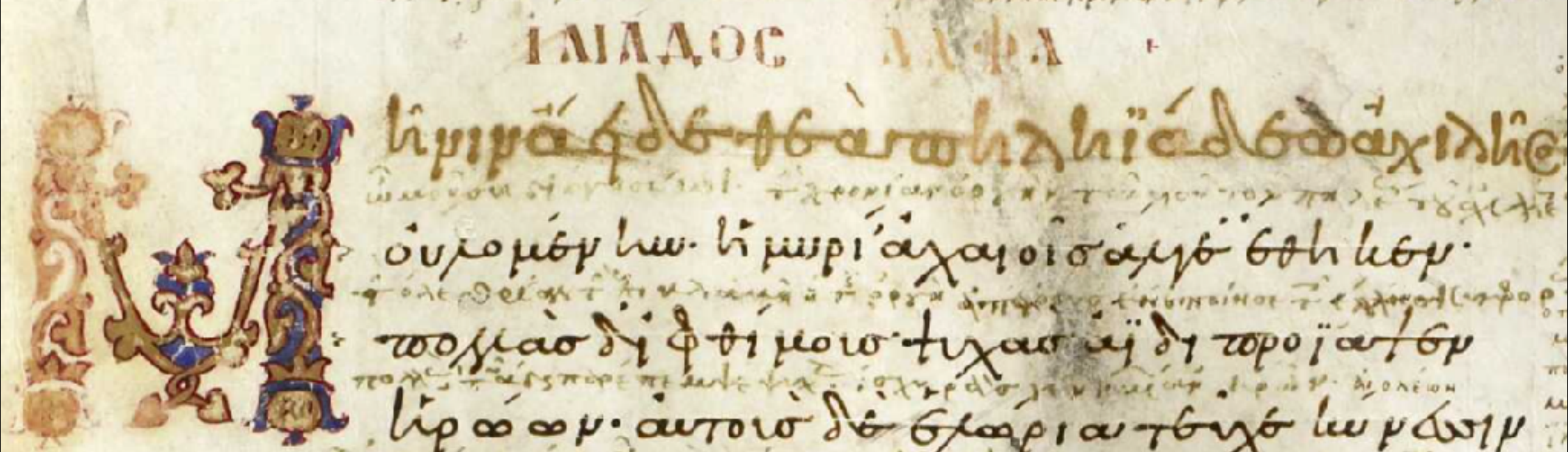


Natural Language Processing

Info 159/259

Lecture 3: Text classification 2 (Jan 28, 2020)

David Bamman, UC Berkeley

Bayes' Rule

Prior belief that Y = positive
(before you see any data)

Likelihood of "really really the worst movie ever"
given that Y= positive

$$P(Y = y \mid X = x) = \frac{P(Y = y)\, P(X = x \mid Y = y)}{\sum_{y \in \mathcal{Y}} P(Y = y) P(X = x \mid Y = y)}$$

Posterior belief that Y=positive given that X="really really the worst movie ever"

This sum ranges over
y=positive + y=negative
(so that it sums to 1)

- Naive Bayes' independence assumption can be killer

- One instance of *hate* makes seeing others much more likely (each mention does contribute the same amount of information)

- We can mitigate this by not reasoning over counts of tokens but by their presence or absence

	Apocalypse now	North
the	1	1
of	0	0
hate	0	~~9~~ 1
genius	1	0
bravest	1	0
stupid	0	1
like	0	1
…		

Naive Bayes

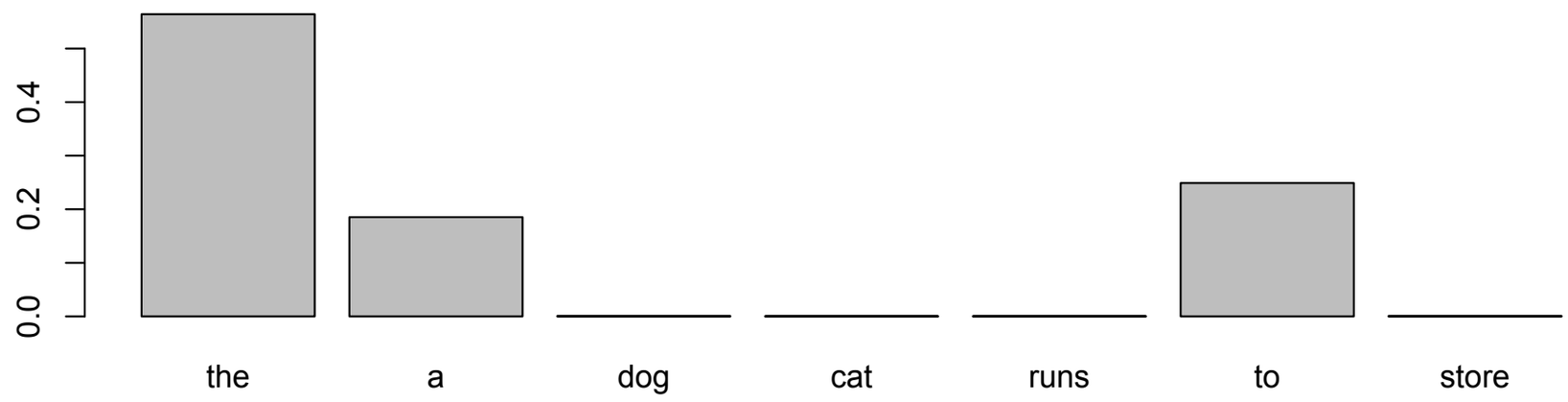
- We have flexibility about what probability distributions we use in NB depending on the features we use and our assumptions about how they interact with the label.

- Multinomial, bernoulli, normal, poisson, etc.

Multinomial Naive Bayes

Discrete distribution for modeling count data (e.g., word counts; single parameter θ

$\theta =$



the	a	dog	cat	runs	to	store
531	209	13	8	2	331	1

Multinomial Naive Bayes

Maximum likelihood parameter estimate

$$\hat{\theta}_i = \frac{n_i}{N}$$

	the	a	dog	cat	runs	to	store
count n	531	209	13	8	2	331	1
θ	0.48	0.19	0.01	0.01	0.00	0.30	0.00

Bernoulli Naive Bayes

- Binary event (true or false; {0, 1})
- One parameter: p (probability of an event occurring)

$$P(x = 1 \mid p) = p$$

$$P(x = 0 \mid p) = 1 - p$$

Examples:

- Probability of a particular feature being true (e.g., review contains “hate”)

$$\hat{p}_{mle} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

Bernoulli Naive Bayes

data points

features

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
f_1	1	0	0	0	1	1	0	0
f_2	0	0	0	0	0	0	1	0
f_3	1	1	1	1	1	0	0	1
f_4	1	0	0	1	1	0	0	1
f_5	0	0	0	0	0	0	0	0

Bernoulli Naive Bayes

	Positive				Negative					
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	$p_{MLE,P}$	$p_{MLE,N}$
f_1	1	0	0	0	1	1	0	0	0.25	0.50
f_2	0	0	0	0	0	0	1	0	0.00	0.25
f_3	1	1	1	1	1	0	0	1	1.00	0.50
f_4	1	0	0	1	1	0	0	1	0.50	0.50
f_5	0	0	0	0	0	0	0	0	0.00	0.00

Tricks for SA

- Negation in bag of words: add negation marker to all words between negation and end of clause (e.g., comma, period) to create new vocab term [Das and Chen 2001]

- I do not [like this movie]

- I do not like_NEG this_NEG movie_NEG

Sentiment Dictionaries

- General Inquirer (1966)
- MPQA subjectivity lexicon (Wilson et al. 2005) http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
- LIWC (Linguistic Inquiry and Word Count, Pennebaker 2015)
- AFINN (Nielsen 2011)
- NRC Word-Emotion Association Lexicon (EmoLex), Mohammad and Turney 2013

pos	neg
unlimited	lag
prudent	contortions
superb	fright
closeness	lonely
impeccably	tenuously
fast-paced	plebeian
treat	mortification
destined	outrage
blessing	allegations
steadfastly	disoriented

Chain rule of probability

$$P(X, Y) = P(Y)\ P(X \mid Y)$$

Bayes' Rule

$$P(Y = y \mid X = x) = \frac{P(Y = y)\, P(X = x \mid Y = y)}{P(X = x)}$$

$$P(Y = y \mid X = x) = \frac{P(X = x, Y = y)}{P(X = x)}$$

Generative vs. Discriminative models

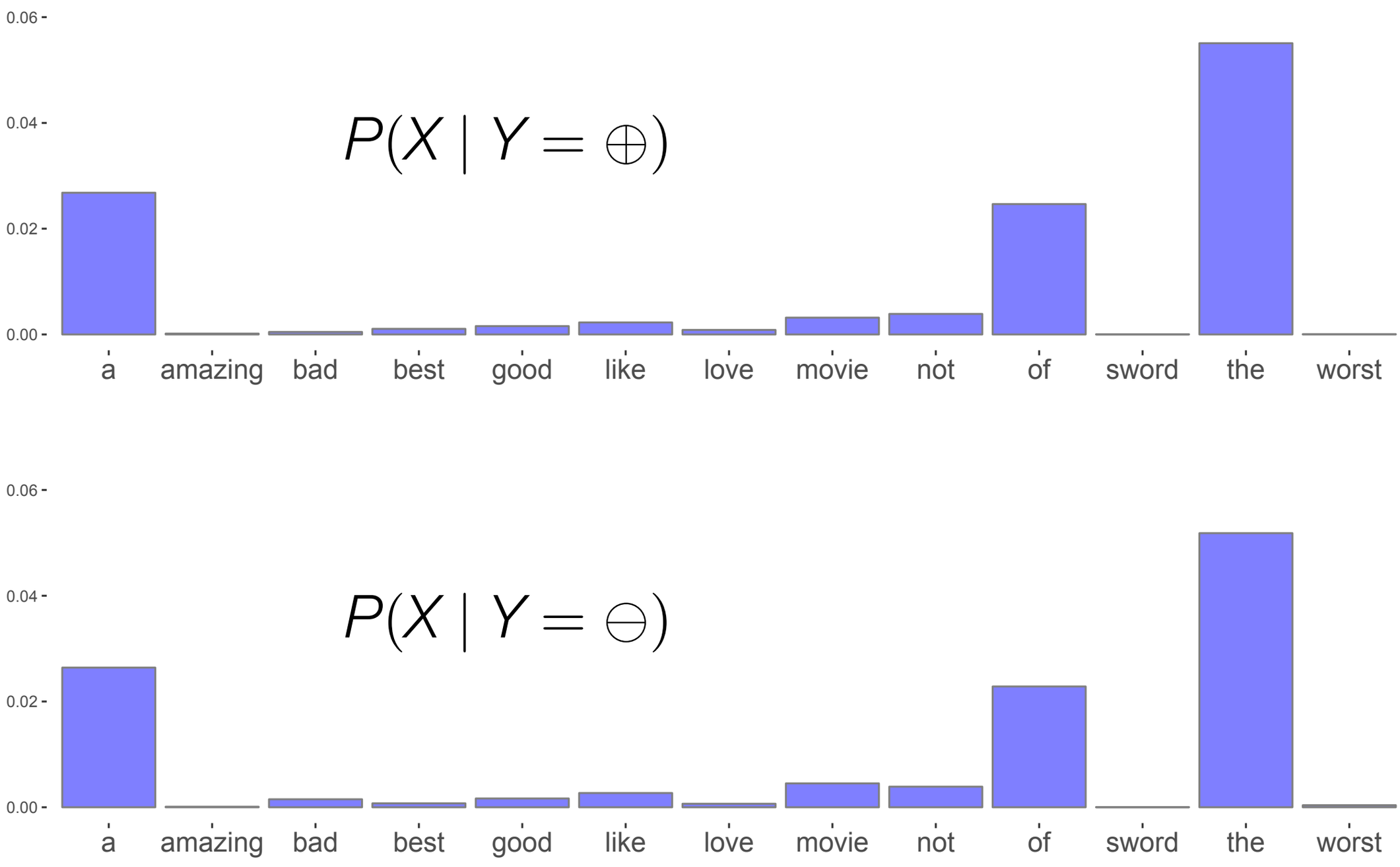
- Generative models specify a joint distribution over the labels and the data. With this you could generate new data

$$P(X, Y) = P(Y)\, P(X \mid Y)$$

- Discriminative models specify the conditional distribution of the label y given the data x. These models focus on how to discriminate between the classes

$$P(Y \mid X)$$

Generating


$P(X \mid Y = \oplus)$
0.06
0.04
0.02
0.00
a
amazing
bad
best
good
like
love
movie
not
of
sword
the
worst
$P(X \mid Y = \ominus)$
0.06
0.04
0.02
0.00
a
amazing
bad
best
good
like
love
movie
not
of
sword
the
worst

Generation

taking allen pete visual an lust be infinite corn physical here decidedly 1 for . never it against perfect the possible spanish of supporting this all this this pride turn that sure the a purpose in real . environment there's trek right . scattered wonder dvd three criticism his .

positive

us are i do tense kevin fall shoot to on want in (. minutes not problems unusually his seems enjoy that : vu scenes rest half in outside famous was with lines chance survivors good to . but of modern-day a changed rent that to in attack lot minutes

negative

Generative models

- With generative models (e.g., Naive Bayes), we ultimately also care about P(Y | X), but we get there by modeling more.

posterior

prior

likelihood

$$P(Y = y \mid X = x) = \frac{P(Y = y)\, P(X = x \mid Y = y)}{\sum_{y \in \mathcal{Y}} P(Y = y) P(X = x \mid Y = y)}$$

- Discriminative models focus on modeling P(Y | X) — *and only P(Y | X)* — directly.

Generation

- How many parameters do we have with a NB model for binary sentiment classification with a vocabulary of 100,000 words?

$P(X \mid Y)$

	the	to	and	that	i	of	we	is	…
Positive	0.041	0.040	0.039	0.038	0.037	0.035	0.032	0.031	
Negative	0.040	0.039	0.039	0.035	0.034	0.033	0.028	0.027	

$P(Y)$

Positive	0.60
Negative	0.40

Remember

$$\sum_{i=1}^{F} x_i \beta_i = x_1 \beta_1 + x_2 \beta_2 + \ldots + x_F \beta_F$$

$$\prod_{i=1}^{F} x_i = x_i \times x_2 \times \ldots \times x_F$$

$$\exp(x) = e^x \approx 2.7^x \qquad \exp(x+y) = \exp(x)\exp(y)$$

$$\log(x) = y \rightarrow e^y = x \qquad \log(xy) = \log(x) + \log(y)$$

Classification

A mapping *h* from input data x (drawn from instance space $\mathcal{X}$) to a label (or labels) y from some enumerable output space $\mathcal{Y}$

$\mathcal{X}$ = set of all documents
$\mathcal{Y}$ = {english, mandarin, greek, …}

x = a single document
y = ancient greek

Training data

positive

"… is a film which still causes real, not figurative, chills to run along my spine, and it is certainly the bravest and most ambitious fruit of Coppola's genius"

Roger Ebert, Apocalypse Now

- "I hated this movie. Hated hated hated hated hated this movie. Hated it. Hated every simpering stupid vacant audience-insulting moment of it. Hated the sensibility that thought anyone would like it."

Roger Ebert, North

negative

Logistic regression

$$P(y = 1 \mid x, \beta) = \frac{1}{1 + \exp\left(-\sum_{i=1}^{F} x_i \beta_i\right)}$$

output space $\mathcal{Y} = \{0, 1\}$

x = feature vector

Feature	Value
the	0
and	0
bravest	0
love	0
loved	0
genius	0
not	0
fruit	1
BIAS	1

β = coefficients

Feature	β
the	0.01
and	0.03
bravest	1.4
love	3.1
loved	1.2
genius	0.5
not	-3.0
fruit	-0.8
BIAS	-0.1

	BIAS	love	loved
β	-0.1	3.1	1.2

	BIAS	love	loved	$a=\sum x_i\beta_i$	exp(-a)	1/(1+exp(-a))
x^1	1	1	0	3	0.05	95.2%
x^2	1	1	1	4.2	0.015	98.5%
x^3	1	0	0	-0.1	1.11	47.4%

Features

- As a discriminative classifier, logistic regression doesn't assume features are independent like Naive Bayes does.

- Its power partly comes in the ability to create richly expressive features without the burden of independence.

- We can represent text through features that are not just the identities of individual words, but any feature that is scoped over the entirety of the input.

features
contains like
has word that shows up in positive sentiment dictionary
review begins with "I like"
at least 5 mentions of positive affectual verbs (like, love, etc.)

Features

- Features are where you can encode your own domain understanding of the problem.

feature classes
unigrams (“like”)
bigrams (“not like”), higher order ngrams
prefixes (words that start with “un-”)
has word that shows up in positive sentiment dictionary

Features

Task	Features
Sentiment classification	Words, presence in sentiment dictionaries, etc.
Keyword extraction	
Fake news detection	
Authorship attribution	

Features

Feature	Value
the	0
and	0
bravest	0
love	0
loved	0
genius	0
not	1
fruit	0
BIAS	1

Feature	Value
like	1
not like	1
did not like	1
in_pos_dict_MPQA	1
in_neg_dict_MPQA	0
in_pos_dict_LIWC	1
in_neg_dict_LIWC	0
author=ebert	1
author=siskel	0

How do we get good values for β?

β = coefficients

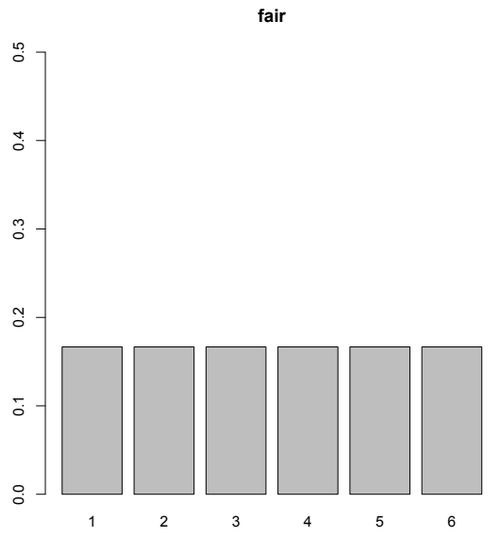
Feature	β
the	0.01
and	0.03
bravest	1.4
love	3.1
loved	1.2
genius	0.5
not	-3.0
fruit	-0.8
BIAS	-0.1

Likelihood

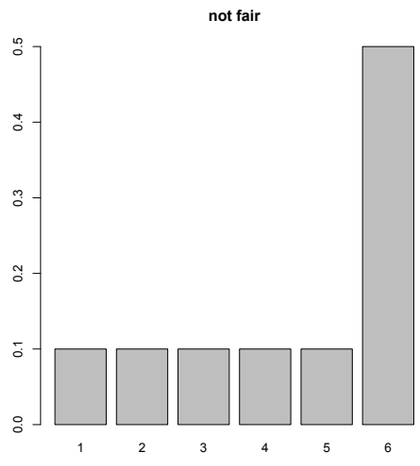
Remember the likelihood of data is its probability under some parameter values

In maximum likelihood estimation, we pick the values of the parameters under which the data is most likely.

Likelihood

P(2 6 6 | 

) =.17 x .17 x .17
= 0.004913

P(2 6 6 | 

) = .1 x .5 x .5
= 0.025

Conditional likelihood

$$\prod_{i}^{N} P(y_i \mid x_i, \beta)$$

For all training data, we want the probability of the true label y for each data point x to be high

	BIAS	love	loved	$a=\sum x_i\beta_i$	exp(-a)	1/(1+exp(-a))	true y
x^1	1	1	0	3	0.05	95.2%	1
x^2	1	1	1	4.2	0.015	98.5%	1
x^3	1	0	0	-0.1	1.11	47.5%	0

Conditional likelihood

$$\prod_{i}^{N} P(y_i \mid x_i, \beta)$$

For all training data, we want probability of the true label y for each data point x to high

This principle gives us a way to pick the values of the parameters β that maximize the probability of the training data <x, y>

The value of β that maximizes likelihood also maximizes the log likelihood

$$\arg\max_{\beta} \prod_{i=1}^{N} P(y_i \mid x_i, \beta) = \arg\max_{\beta} \log \prod_{i=1}^{N} P(y_i \mid x_i, \beta)$$

The log likelihood is an easier form to work with:

$$\log \prod_{i=1}^{N} P(y_i \mid x_i, \beta) = \sum_{i=1}^{N} \log P(y_i \mid x_i, \beta)$$

- We want to find the value of β that leads to the highest value of the log likelihood:

$$\ell(\beta) = \sum_{i=1}^{N} \log P(y_i \mid x_i, \beta)$$

We want to find the values of β that make the value of this function the greatest

$$\sum_{<x,y=+1>} \log P(1 \mid x, \beta) + \sum_{<x,y=0>} \log P(0 \mid x, \beta)$$

$$\frac{\partial}{\partial \beta_i} \ell(\beta) = \sum_{<x,y>} (y - \hat{p}(x))\, x_i$$

Gradient descent

Algorithm 1 Logistic regression gradient descent

1: Data: training data $x \in \mathbb{R}^F, y \in \{0, 1\}$
2: $\beta = 0^F$
3: **while** not converged **do**
4: $\quad \beta_{t+1} = \beta_t + \alpha \sum_{i=1}^{N} (y_i - \hat{p}(x_i))\, x_i$
5: **end while**

If y is 1 and p(x) = 0, then this still pushes the weights a lot

If y is 1 and p(x) = 0.99, then this still pushes the weights just a little bit

Stochastic g.d.

- Batch gradient descent reasons over every training data point for each update of β. This can be slow to converge.

- Stochastic gradient descent updates β after each data point.

Algorithm 2 Logistic regression stochastic gradient descent

Data: training data $x \in \mathbb{R}^F, y \in \{0, 1\}$
$\beta = 0^F$
while not converged **do**
    for $i = 1$ to N **do**
        $\beta_{t+1} = \beta_t + \alpha \left(y_i - \hat{p}(x_i)\right) x_i$
    end for
end while

Practicalities

- When calculating the P(y | x) or in calculating the gradient, you don't need to loop through all features — only those with nonzero values
- (Which makes sparse, binary values useful)

$$P(y = 1 \mid x, \beta) = \frac{1}{1 + \exp\left(-\sum_{i=1}^{F} x_i \beta_i\right)}$$

$$\frac{\partial}{\partial \beta_i} \ell(\beta) = \sum_{<x,y>} (y - \hat{p}(x))\, x_i$$

$$\frac{\partial}{\partial \beta_i} \ell(\beta) = \sum_{<x,y>} (y - \hat{p}(x))\, x_i$$

If a feature x_i only shows up with the positive class (e.g., positive sentiment), what are the possible values of its corresponding β_i?

$$\frac{\partial}{\partial \beta_i} \ell(\beta) = \sum_{<x,y>} (1 - 0)1$$

$$\frac{\partial}{\partial \beta_i} \ell(\beta) = \sum_{<x,y>} (1 - 0.9999999)1$$

always positive

Many features that show up rarely may likely only appear (by chance) with one label

More generally, may appear so few times that the noise of randomness dominates

β = coefficients

Feature	β
like	2.1
did not like	1.4
in_pos_dict_MPQA	1.7
in_neg_dict_MPQA	-2.1
in_pos_dict_LIWC	1.4
in_neg_dict_LIWC	-3.1
author=ebert	-1.7
author=ebert ∧ dog ∧ starts with “in”	30.1

Feature selection

- We could threshold features by minimum count but that also throws away information

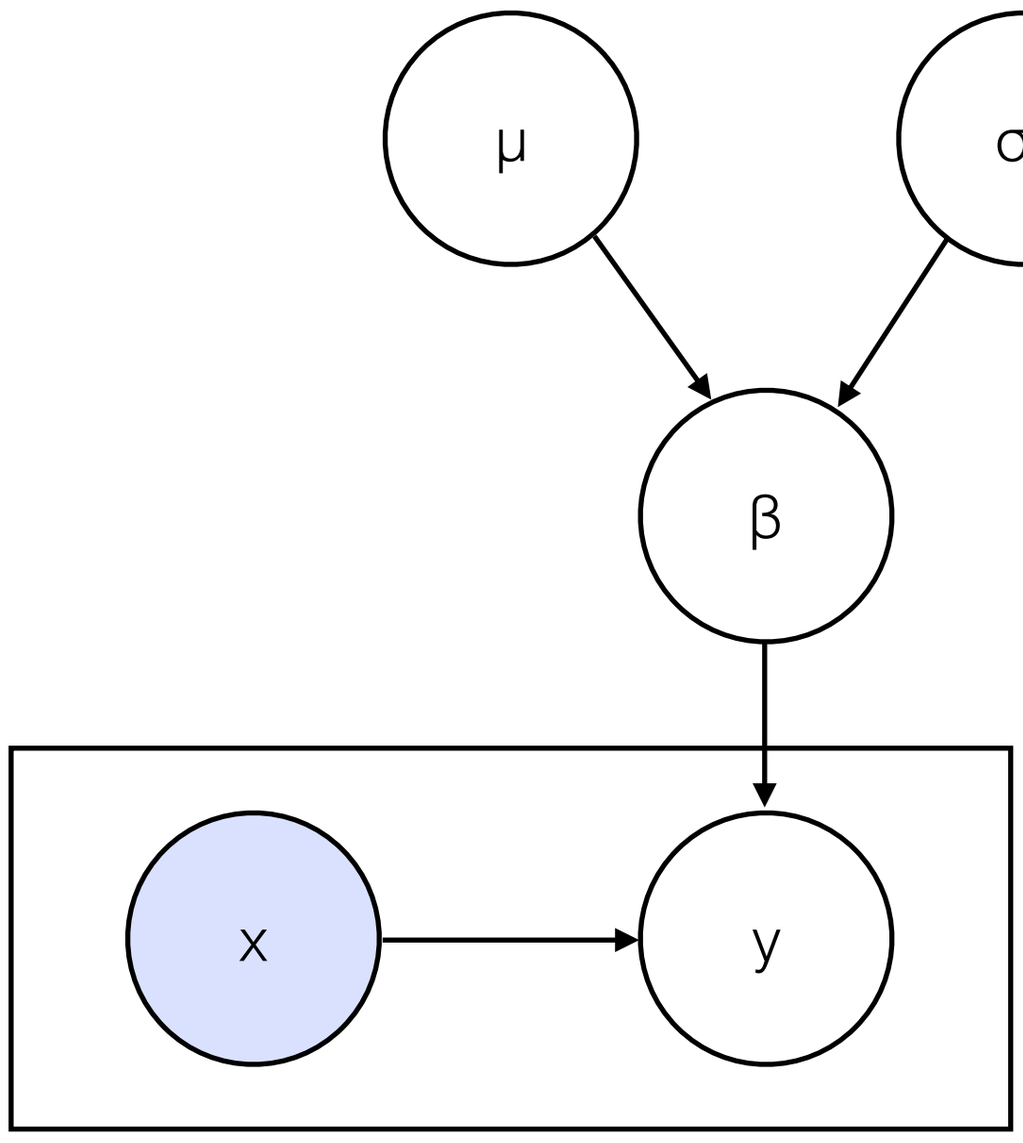
- We can take a probabilistic approach and encode a prior belief that all β should be 0 unless we have strong evidence otherwise

L2 regularization

$$\ell(\beta) = \underbrace{\sum_{i=1}^{N} \log P(y_i \mid x_i, \beta)}_{\text{we want this to be high}} - \underbrace{\eta \sum_{j=1}^{F} \beta_j^2}_{\text{but we want this to be small}}$$

- We can do this by changing the function we're trying to optimize by adding a penalty for having values of β that are high
- This is equivalent to saying that each β element is drawn from a Normal distribution centered on 0.
- η controls how much of a penalty to pay for coefficients that are far from 0 (optimize on development data)

no L2 regularization		some L2 regularization		high L2 regularization	
33.83	Won Bin	2.17	Eddie Murphy	0.41	Family Film
29.91	Alexander Beyer	1.98	Tom Cruise	0.41	Thriller
24.78	Bloopers	1.70	Tyler Perry	0.36	Fantasy
23.01	Daniel Brühl	1.70	Michael Douglas	0.32	Action
22.11	Ha Jeong-woo	1.66	Robert Redford	0.25	Buddy film
20.49	Supernatural	1.66	Julia Roberts	0.24	Adventure
18.91	Kristine DeBell	1.64	Dance	0.20	Comp Animation
18.61	Eddie Murphy	1.63	Schwarzenegger	0.19	Animation
18.33	Cher	1.63	Lee Tergesen	0.18	Science Fiction
18.18	Michael Douglas	1.62	Cher	0.18	Bruce Willis



$$\beta \sim \text{Norm}(\mu, \sigma^2)$$

$$y \sim \text{Ber}\left(\frac{\exp\left(\sum_{i=1}^{F} x_i \beta_i\right)}{1 + \exp\left(\sum_{i=1}^{F} x_i \beta_i\right)}\right)$$

L1 regularization

$$\ell(\beta) = \underbrace{\sum_{i=1}^{N} \log P(y_i \mid x_i, \beta)}_{\text{we want this to be high}} - \underbrace{\eta \sum_{j=1}^{F} |\beta_j|}_{\text{but we want this to be small}}$$

- L1 regularization encourages coefficients to be exactly 0.
- η again controls how much of a penalty to pay for coefficients that are far from 0 (optimize on development data)

What do the coefficients mean?

$$P(y \mid x, \beta) = \frac{\exp\left(x_0\beta_0 + x_1\beta_1\right)}{1 + \exp\left(x_0\beta_0 + x_1\beta_1\right)}$$

$$P(y \mid x, \beta)(1 + \exp\left(x_0\beta_0 + x_1\beta_1\right)) = \exp\left(x_0\beta_0 + x_1\beta_1\right)$$

$$P(y \mid x, \beta) + P(y \mid x, \beta)\exp\left(x_0\beta_0 + x_1\beta_1\right) = \exp\left(x_0\beta_0 + x_1\beta_1\right)$$

$$P(y \mid x, \beta) + P(y \mid x, \beta) \exp(x_0\beta_0 + x_1\beta_1) = \exp(x_0\beta_0 + x_1\beta_1)$$

$$P(y \mid x, \beta) = \exp(x_0\beta_0 + x_1\beta_1) - P(y \mid x, \beta) \exp(x_0\beta_0 + x_1\beta_1)$$

$$P(y \mid x, \beta) = \exp(x_0\beta_0 + x_1\beta_1)(1 - P(y \mid x, \beta))$$

This is the odds of y occurring

$$\frac{P(y \mid x, \beta)}{1 - P(y \mid x, \beta)} = \exp(x_0\beta_0 + x_1\beta_1)$$

Odds

- Ratio of an event occurring to its not taking place

$$\frac{P(x)}{1 - P(x)}$$

Green Bay Packers
vs. SF 49ers

$$\frac{0.75}{0.25} = \frac{3}{1} = 3:1$$

probability of GB winning

odds for GB winning

$$P(y \mid x, \beta) + P(y \mid x, \beta) \exp(x_0\beta_0 + x_1\beta_1) = \exp(x_0\beta_0 + x_1\beta_1)$$

$$P(y \mid x, \beta) = \exp(x_0\beta_0 + x_1\beta_1) - P(y \mid x, \beta) \exp(x_0\beta_0 + x_1\beta_1)$$

$$P(y \mid x, \beta) = \exp(x_0\beta_0 + x_1\beta_1)(1 - P(y \mid x, \beta))$$

This is the odds of y occurring

$$\frac{P(y \mid x, \beta)}{1 - P(y \mid x, \beta)} = \exp(x_0\beta_0 + x_1\beta_1)$$

$$\frac{P(y \mid x, \beta)}{1 - P(y \mid x, \beta)} = \exp(x_0\beta_0) \exp(x_1\beta_1)$$

$$\frac{P(y \mid x, \beta)}{1 - P(y \mid x, \beta)} = \exp(x_0\beta_0)\exp(x_1\beta_1)$$

Let's increase the value of x by 1 (e.g., from 0 → 1)

$$\exp(x_0\beta_0)\exp((x_1 + 1)\beta_1)$$

$$\exp(x_0\beta_0)\exp(x_1\beta_1 + \beta_1)$$

$$\exp(x_0\beta_0)\exp(x_1\beta_1)\exp(\beta_1)$$

exp(β) represents the factor by which the **odds** change with a 1-unit increase in x

$$\frac{P(y \mid x, \beta)}{1 - P(y \mid x, \beta)}\exp(\beta_1)$$