Deconstructing Data Science

David Bamman, UC Berkeley

Info 290
Lecture 9: Logistic regression

Feb 14, 2017

Generative vs. Discriminative models

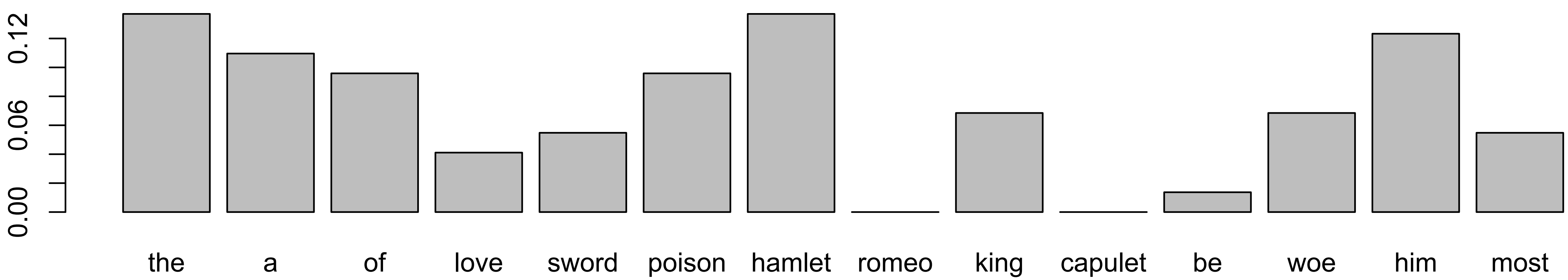
- Generative models specify a joint distribution over the labels and the data. With this you could generate new data

$$P(x, y) = P(y)\, P(x \mid y)$$

- 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



$P(x \mid y = \text{Hamlet})$

$P(x \mid y = \text{Romeo and Juliet})$

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

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

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

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 skyscrapers
$\mathcal{Y}$ = {art deco, neo-gothic, modern}

x = the empire state building
y = art deco

x = feature vector

Feature	Value
follow clinton	0
follow trump	0
“benghazi”	0
negative sentiment + “benghazi”	0
“illegal immigrants”	0
“republican” in profile	0
“democrat” in profile	0
self-reported location = Berkeley	1

β = coefficients

Feature	β
follow clinton	-3.1
follow trump	6.8
“benghazi”	1.4
negative sentiment + “benghazi”	3.2
“illegal immigrants”	8.7
“republican” in profile	7.9
“democrat” in profile	-3.0
self-reported location = Berkeley	-1.7

Logistic regression

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

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

	benghazi	follows trump	follows clinton
β	0.7	1.2	-1.1

	benghazi	follows trump	follows clinton	$a=\sum x_i \beta_i$	exp(a)	exp(a)/ 1+exp(a)
x^1	1	1	0	1.9	6.69	87.0%
x^2	0	0	1	-1.1	0.33	25.0%
x^3	1	0	1	-0.4	0.67	40.1%

How do we get good values for β?

β = coefficients

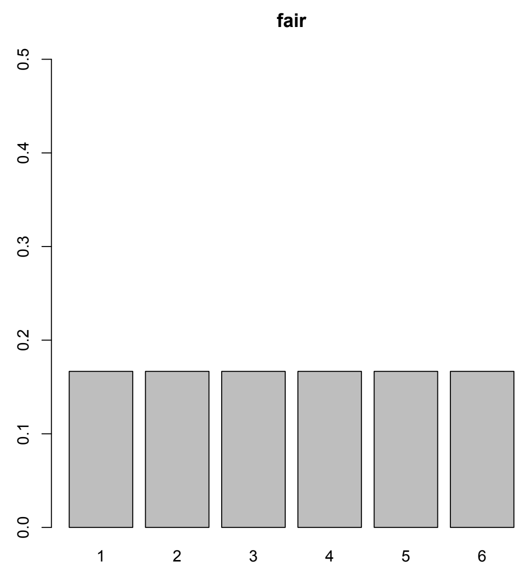
Feature	β
follow clinton	-3.1
follow trump	6.8
“benghazi”	1.4
negative sentiment + “benghazi”	3.2
“illegal immigrants”	8.7
“republican” in profile	7.9
“democrat” in profile	-3.0
self-reported location = Berkeley	-1.7

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.

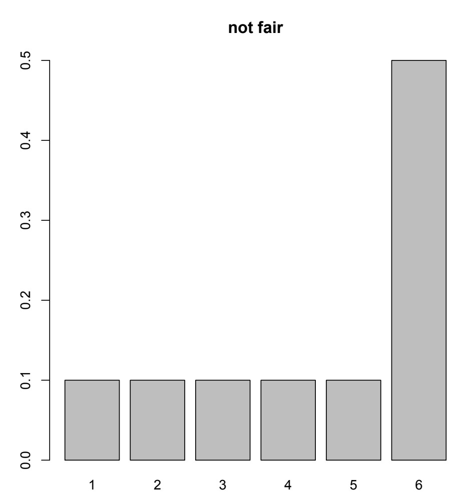
Likelihood

P(2 6 6 | 

)

$= .17 \times .17 \times .17$

$= 0.004913$

P(2 6 6 | 

)

$= .1 \times .5 \times .5$

$= 0.025$

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

	benghazi	follows trump	follows clinton	$a=\sum x_i\beta_i$	exp(a)	exp(a)/ 1+exp(a)	true y
x^1	1	1	0	1.9	6.69	87.0%	1
x^2	0	0	1	-1.1	0.33	25.0%	0
x^3	1	0	1	-0.4	0.67	40.1%	1

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

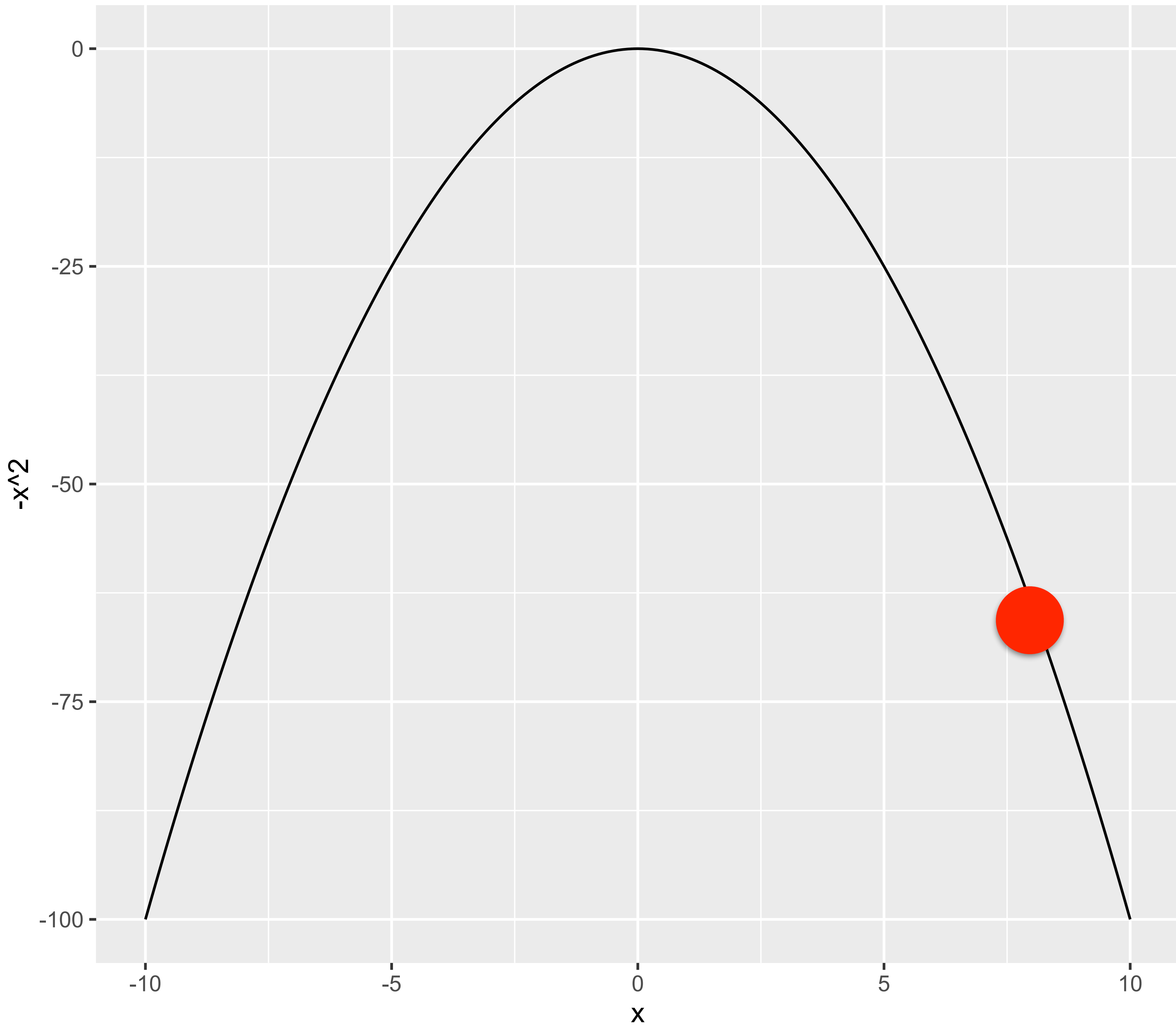
$$\arg\max_{\beta} \prod_{i=1}^{N} P(y_i \mid x_i, \beta) = \arg\max_{\beta} \log \prod_{i=1}^{N} P(y_i \mid x_i, \beta)$$

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:

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



$$x + \alpha(-2x)$$

$[\alpha = 0.1]$

x	$-x^2$	grad.
8.0	-64.0	-1.6
6.4	-41.0	-1.3
5.1	-26.2	-1.0
4.1	-16.8	-0.8
3.3	-10.8	-0.7
2.6	-6.9	-0.5
2.1	-4.4	-0.4
1.7	-2.8	-0.3
1.3	-1.8	-0.3
1.1	-1.1	-0.2
0.9	-0.7	-0.2
0.7	-0.5	-0.1

$$\frac{d}{dx} - x^2 = -2x$$

We can get to maximum value of this function by following the gradient

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

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

Perceptron

Algorithm 3 Perceptron 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**
        $\beta_{t+1} = \beta_t + \alpha \, (y_i - \hat{y}) \, x_i$
    end for
end while

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

Algorithm 3 Perceptron 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**
        $\beta_{t+1} = \beta_t + \alpha \left(y_i - \hat{y} \right) x_i$
    end for
end while

Stochastic g.d.

Logistic regression stochastic update

$$\beta_i + \alpha\,(y - \hat{p}(x))\,x_i$$

p is between 0 and 1

Perceptron stochastic update

$$\beta_i + \alpha\,(y - \hat{y})\,x_i$$

ŷ is exactly 0 or 1

The perceptron is an approximation to logistic regression

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)

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

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

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

If a feature x_i only shows up with one class (e.g., democrats), 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

β = coefficients

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

Feature	β
follow clinton	-3.1
follow trump + follow NFL + follow bieber	7299302
“benghazi”	1.4
negative sentiment + “benghazi”	3.2
“illegal immigrants”	8.7
“republican” in profile	7.9
“democrat” in profile	-3.0
self-reported location = Berkeley	-1.7

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

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

- 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

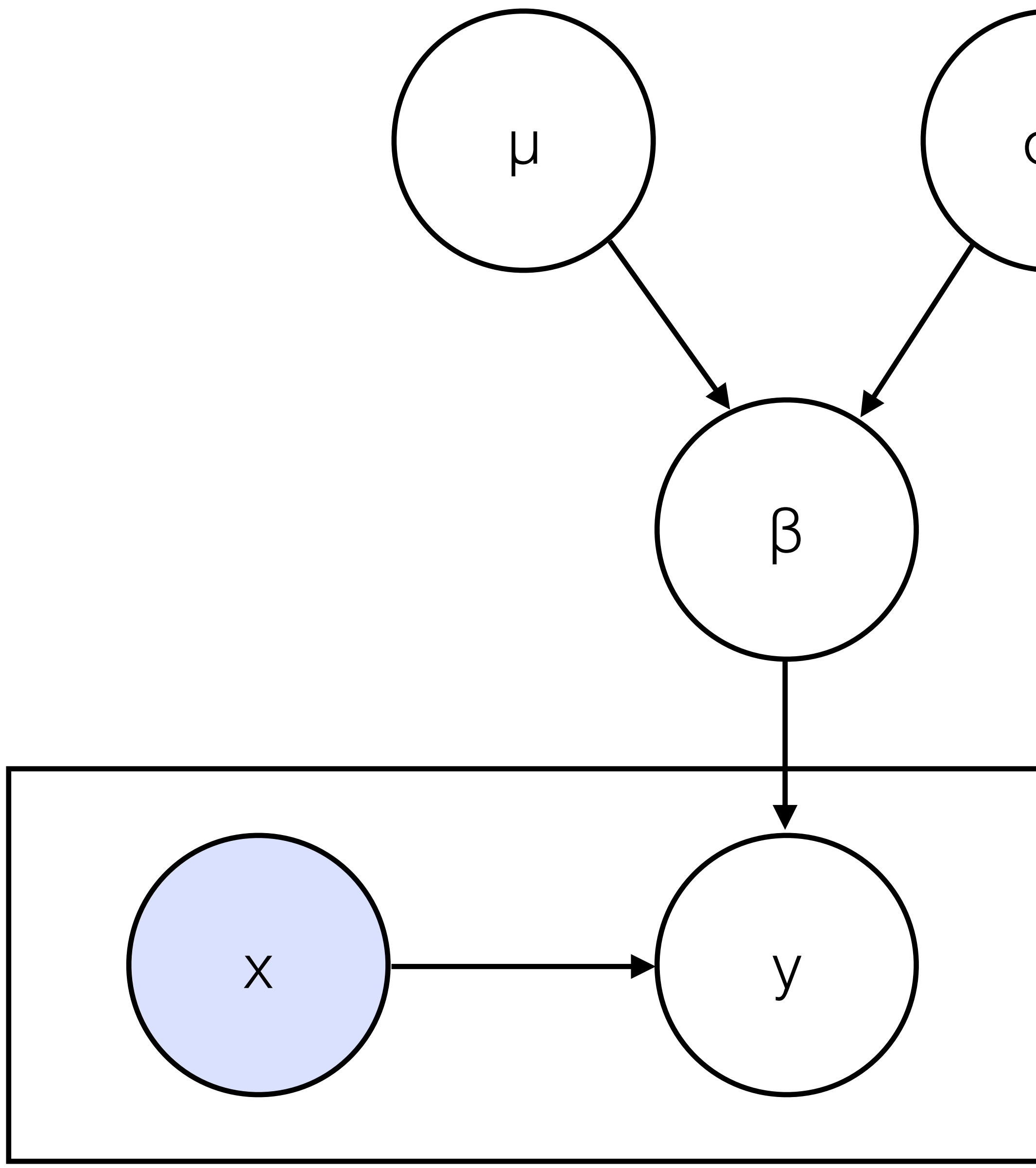
33.83	Won Bin
29.91	Alexander Beyer
24.78	Bloopers
23.01	Daniel Brühl
22.11	Ha Jeong-woo
20.49	Supernatural
18.91	Kristine DeBell
18.61	Eddie Murphy
18.33	Cher
18.18	Michael Douglas

some L2 regularization

2.17	Eddie Murphy
1.98	Tom Cruise
1.70	Tyler Perry
1.70	Michael Douglas
1.66	Robert Redford
1.66	Julia Roberts
1.64	Dance
1.63	Schwarzenegger
1.63	Lee Tergesen
1.62	Cher

high L2 regularization

0.41	Family Film
0.41	Thriller
0.36	Fantasy
0.32	Action
0.25	Buddy film
0.24	Adventure
0.20	Comp Animation
0.19	Animation
0.18	Science Fiction
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

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

- 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?

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

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

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

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

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

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

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

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

Example

How do we interpret this change of odds? Is it causal?

β	change in odds	feature name
2.17	8.76	Eddie Murphy
1.98	7.24	Tom Cruise
1.70	5.47	Tyler Perry
1.70	5.47	Michael Douglas
1.66	5.26	Robert Redford
…	…	…
-0.94	0.39	Kevin Conway
-1.00	0.37	Fisher Stevens
-1.05	0.35	B-movie
-1.14	0.32	Black-and-white
-1.23	0.29	Indie

Rao et al. (2010)

FEATURE	*Description/Example*
SIMLEYS	A list of emoticons compiled from the Wikipedia.
OMG	Abbreviation for 'Oh My God'
ELLIPSES	'....'
POSSESIVE BIGRAMS	E.g. my_XXX, our_XXX
REPATED ALPHABETS	E.g. niceeeeee, noooo waaaay
SELF	E.g., I_xxx, Im_xxx
LAUGH	E.g. LOL, ROTFL, LMFAO, haha, hehe
SHOUT	Text in ALLCAPS
EXASPERATION	E.g. Ugh, mmmm, hmmm, ahh, grrr
AGREEMENT	E.g. yea, yeah, ohya
HONORIFICS	E.g. dude, man, bro, sir
AFFECTION	E.g. xoxo
EXCITEMENT	A string of exclamation symbols (!!!!!)
SINGLE EXCLAIM	A single exclamation at the end of the tweet
PUZZLED PUNCT	A combination of any number of ? and ! (!?!!??!)

Democrat		Republican	
my_youthful	1	*my_zionist*	1
my_yoga	1	*my_yuengling*	1
my_vegetarianism	1	*my_weapons*	1
my_upscale	1	*my_walmart*	1
my_tofurkey	1	*my_trucker*	1
my_synagogue	1	*my_patroit*	1
my_lakers	0.93	*my_lsu*	1
my_gays	0.8	*my_blackeberry*	1
my_feminist	0.67	*my_redneck*	0.89
my_sushi	0.6	*my_marine*	0.82
my_marathon	-10	my_partner	-0.29
my_trailer	-11	my_atheism	-1
my_liberty	-11.5	my_sushi	-1.5
my_information	-12.5	my_netflix	-2.2
my_teleprompter	-13	my_passport	-2.43
my_warrior	-14	my_manager	-3.67
my_property	-19	my_bicycle	-4
my_lines	-19	my_android	-6
my_guns	-19.67	my_medicare	-14
my_bishop	-33	my_nigga	-17

Above 30		Below 30	
my_zzzzzzz	1	*my_zunehd*	1
my_work	1	*my_yuppie*	1
my_epidural	1	*my_sorors*	0.94
my_daughters	0.98	*my_rents*	0.93
my_grandkids	0.95	*my_classes*	0.90
my_retirement	0.92	*my_xbox*	0.87
my_hubbys	0.91	*my_greek*	0.79
my_workouts	0.9	*my_biceps*	0.75
my_teenage	0.88	*my_homies*	0.70
my_inlaws	0.86	*my_uniform*	0.56
my_bestfriend	-17	my_memoir	-21
my_internship	-18.17	my_daughter	-24.70
my_dorm	-18.75	my_youngest	-24.71
my_cuzzo	-19	my_tribe	-29
my_bby	-26	my_nelson	-36
my_boi	-30	my_oldest	-39
my_dudes	-34	my_2yo	-39
my_roomate	-37	my_kiddos	-45
my_formspring	-42	my_daughters	-56
my_hw	-51	my_prayer	-62

Disfluency/Agreement	*#female/#male*
oh	2.3
ah	2.1
hmm	1.6
ugh	1.6
grrr	1.3
yeah, yea, ...	0.8

Feature	*#female/#male*
Emoticons	3.5
Elipses	1.5
Character repetition	1.4
Repeated exclamation	2.0
Puzzled punctuation	1.8
OMG	4.0

Thursday

- Krippendorff (2004), "Validity," Content Analysis
- Read well! Come prepared to discuss the different types of validity. (It's on bCourses)