



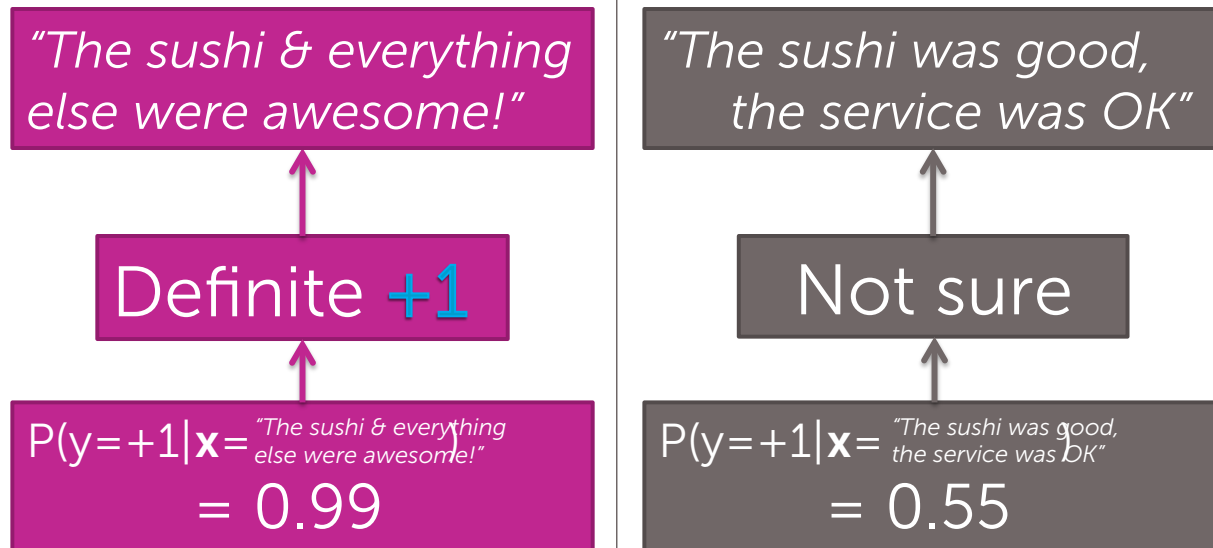
Linear classifiers:



Parameter learning 

Emily Fox & Carlos Guestrin
Machine Learning Specialization
University of Washington

Learn a probabilistic classification model



Many classifiers provide a degree of certainty:

Output label \rightarrow $P(y|\mathbf{x})$ \leftarrow Input sentence

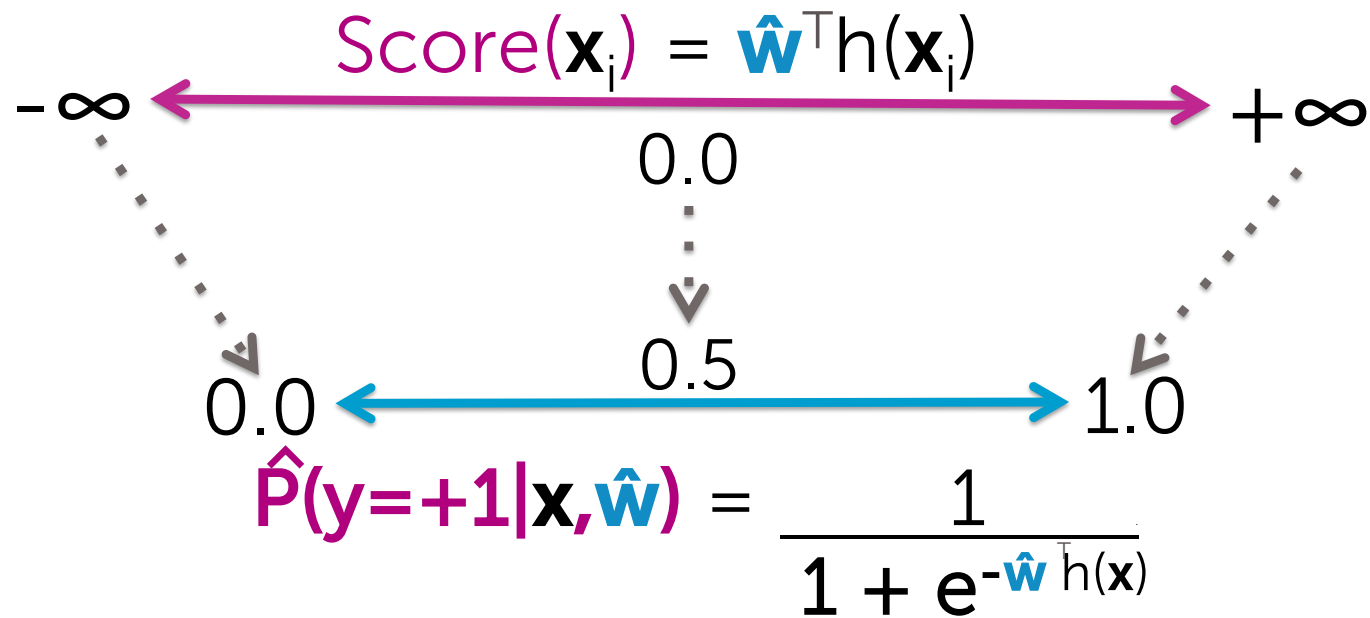
Extremely useful in practice

A (linear) classifier

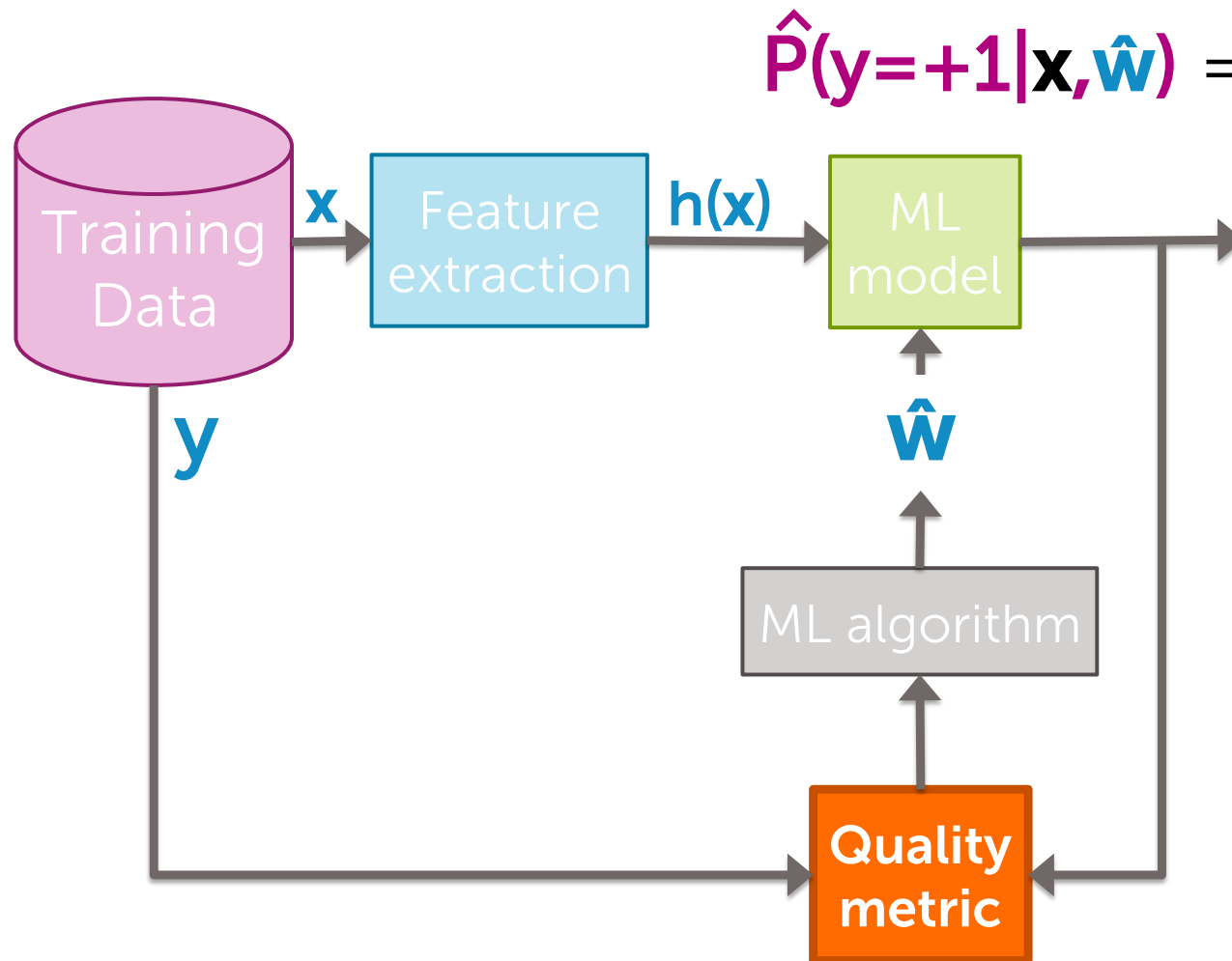
- Will use training data to learn a weight or coefficient for each word

Word	Coefficient	Value
	\hat{w}_0	-2.0
good	\hat{w}_1	1.0
great	\hat{w}_2	1.5
awesome	\hat{w}_3	2.7
bad	\hat{w}_4	-1.0
terrible	\hat{w}_5	-2.1
awful	\hat{w}_6	-3.3
restaurant, the, we, ...	$\hat{w}_7, \hat{w}_8, \hat{w}_9, \dots$	0.0
...		...

Logistic regression model



Quality metric for logistic regression: 
Maximum likelihood estimation



$$\hat{P}(y=+1|\mathbf{x},\hat{\mathbf{w}}) = \frac{1}{1 + e^{-\hat{\mathbf{w}}^T h(\mathbf{x})}}$$

Learning problem

Training data:

N observations (\mathbf{x}_i, y_i)

$x[1] = \text{\#awesome}$	$x[2] = \text{\#awful}$	$y = \text{sentiment}$
2	1	+1
0	2	-1
3	3	-1
4	1	+1
1	1	+1
2	4	-1
0	3	-1
0	1	-1
2	1	+1

Optimize
quality metric
on training
data

$\hat{\mathbf{W}}$



MOVE TO HEAD SHOT

Finding best coefficients

$x[1] = \text{\#awesome}$	$x[2] = \text{\#awful}$	$y = \text{sentiment}$
2	1	+1
0	2	-1
3	3	-1
4	1	+1
1	1	+1
2	4	-1
0	3	-1
0	1	-1
2	1	+1

Finding best coefficients

x[1] = #awesome	x[2] = #awful	y = sentiment
0	2	-1
3	3	-1
2	4	-1
0	3	-1
0	1	-1
2	4	-1
0	3	-1
0	1	-1

x[1] = #awesome	x[2] = #awful	y = sentiment
2	1	+1
4	1	+1
1	1	+1
2	1	+1
1	1	+1
2	1	+1



Finding best coefficients

x[1] = #awesome	x[2] = #awful	y = sentiment
0	2	-1
3	3	-1
2	4	-1
0	3	-1
0	1	-1

$$P(y=+1|\mathbf{x}_i, \mathbf{w}) = 0.0$$

x[1] = #awesome	x[2] = #awful	y = sentiment
2	1	+1
4	1	+1
1	1	+1
2	1	+1

$$P(y=+1|\mathbf{x}_i, \mathbf{w}) = 1.0$$

Pick $\hat{\mathbf{w}}$ that makes 

Quality metric = Likelihood function

Negative data points

$$P(y=+1|\mathbf{x}_i, \mathbf{w}) = 0.0$$

Positive data points

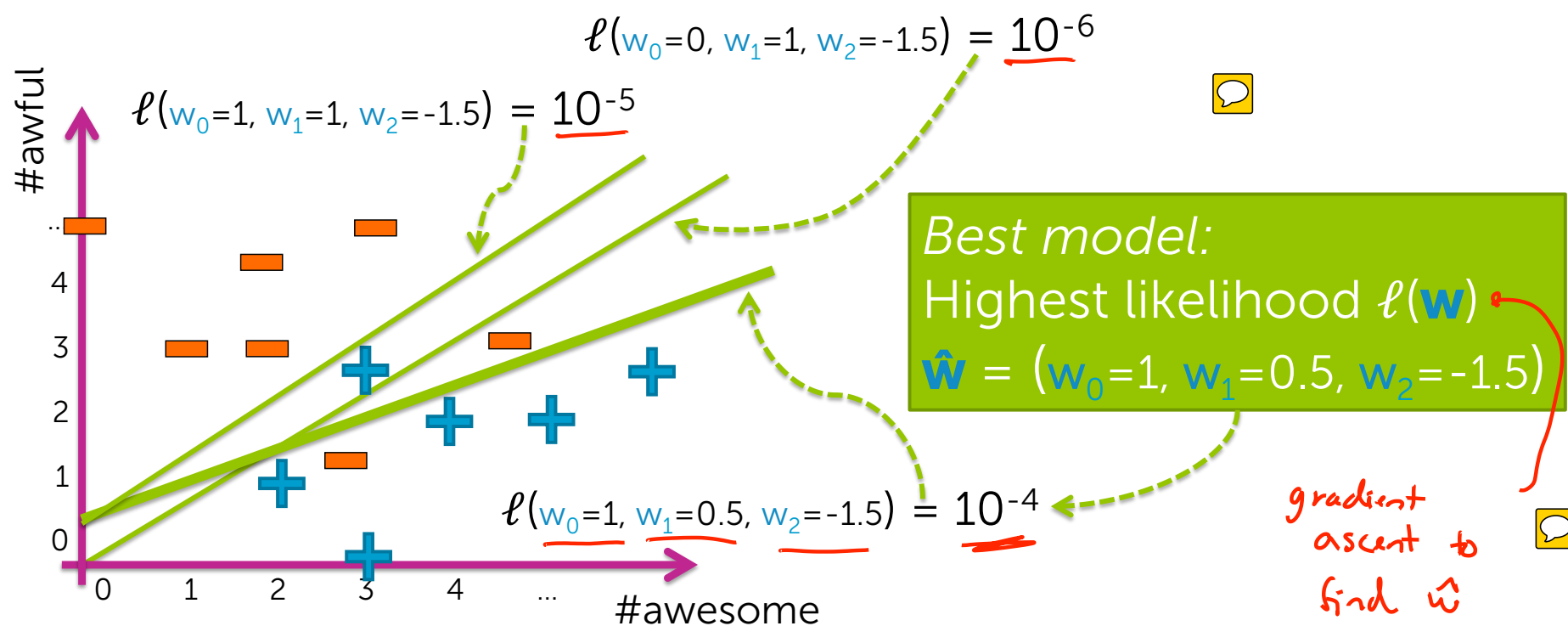
$$P(y=+1|\mathbf{x}_i, \mathbf{w}) = 1.0$$

No $\hat{\mathbf{w}}$ achieves perfect predictions (usually)

Likelihood $\ell(\mathbf{w})$: Measures quality of fit for model with coefficients \mathbf{w}

Find "best" classifier

Maximize likelihood over all possible w_0, w_1, w_2





Data likelihood

Quality metric: probability of data

$x[1] = \text{\#awesome}$	$x[2] = \text{\#awful}$	$y = \text{sentiment}$
2	1	+1

x_1

y_1

If model good, should predict:

$\hat{y}_1 = +1$

Pick w to maximize:

$$P(y = +1 | x_1, w) = P(y = +1 | x[1]=2, x[2]=1, w)$$

$x[1] = \text{\#awesome}$	$x[2] = \text{\#awful}$	$y = \text{sentiment}$
0	2	-1

x_2

y_2

If model good, should predict:

$\hat{y}_2 = -1$

Pick w to maximize:

$$P(y = -1 | x_2, w)$$



Maximizing likelihood (probability of data)



Data point	x[1]	x[2]	y	Choose w to maximize
\mathbf{x}_1, y_1	2	1	+1	$P(y=+1 \mathbf{x}_1, w) = P(y=+1 x[1]=2, x[2]=1, w)$
\mathbf{x}_2, y_2	0	2	-1	$P(y=-1 \mathbf{x}_2, w)$
\mathbf{x}_3, y_3	3	3	<u>-1</u>	$P(y=-1 \mathbf{x}_3, w)$
\mathbf{x}_4, y_4	4	1	<u>+1</u>	$P(y=+1 \mathbf{x}_4, w)$
\mathbf{x}_5, y_5	1	1	+1	
\mathbf{x}_6, y_6	2	4	-1	
\mathbf{x}_7, y_7	0	3	-1	
\mathbf{x}_8, y_8	0	1	-1	
\mathbf{x}_9, y_9	2	1	+1	

Must combine into single
measure of quality ?

Multiply probabilities

$P(y=+1 | \mathbf{x}_1, w) P(y=-1 | \mathbf{x}_2, w) P(y=-1 | \mathbf{x}_3, w) \dots$



Learn logistic regression model with maximum likelihood estimation (MLE)

Data point	x[1]	x[2]	y	Choose \mathbf{w} to maximize
\mathbf{x}_1, y_1	2	1	<u>$y_1 = +1$</u>	$P(\underline{y=+1} \mathbf{x}[1]=2, \mathbf{x}[2]=1, \mathbf{w})$
\mathbf{x}_2, y_2	0	2	<u>-1</u>	$P(\underline{y=-1} \mathbf{x}[1]=0, \mathbf{x}[2]=2, \mathbf{w})$
\mathbf{x}_3, y_3	3	3	-1	$P(y=-1 \mathbf{x}[1]=3, \mathbf{x}[2]=3, \mathbf{w})$
\mathbf{x}_4, y_4	4	1	$+1$	$P(y=+1 \mathbf{x}[1]=4, \mathbf{x}[2]=1, \mathbf{w})$


$$\ell(\mathbf{w}) = \underbrace{P(y_1 | \mathbf{x}_1, \mathbf{w})}_{\text{pick } \mathbf{w} \text{ to make this fn. as large as possible}} \underbrace{P(y_2 | \mathbf{x}_2, \mathbf{w})} \underbrace{P(y_3 | \mathbf{x}_3, \mathbf{w})} \underbrace{P(y_4 | \mathbf{x}_4, \mathbf{w})}$$

Num. of data points $\rightarrow N$

$$\ell(\mathbf{w}) = \prod_{i=1}^N P(y_i | \mathbf{x}_i, \mathbf{w})$$

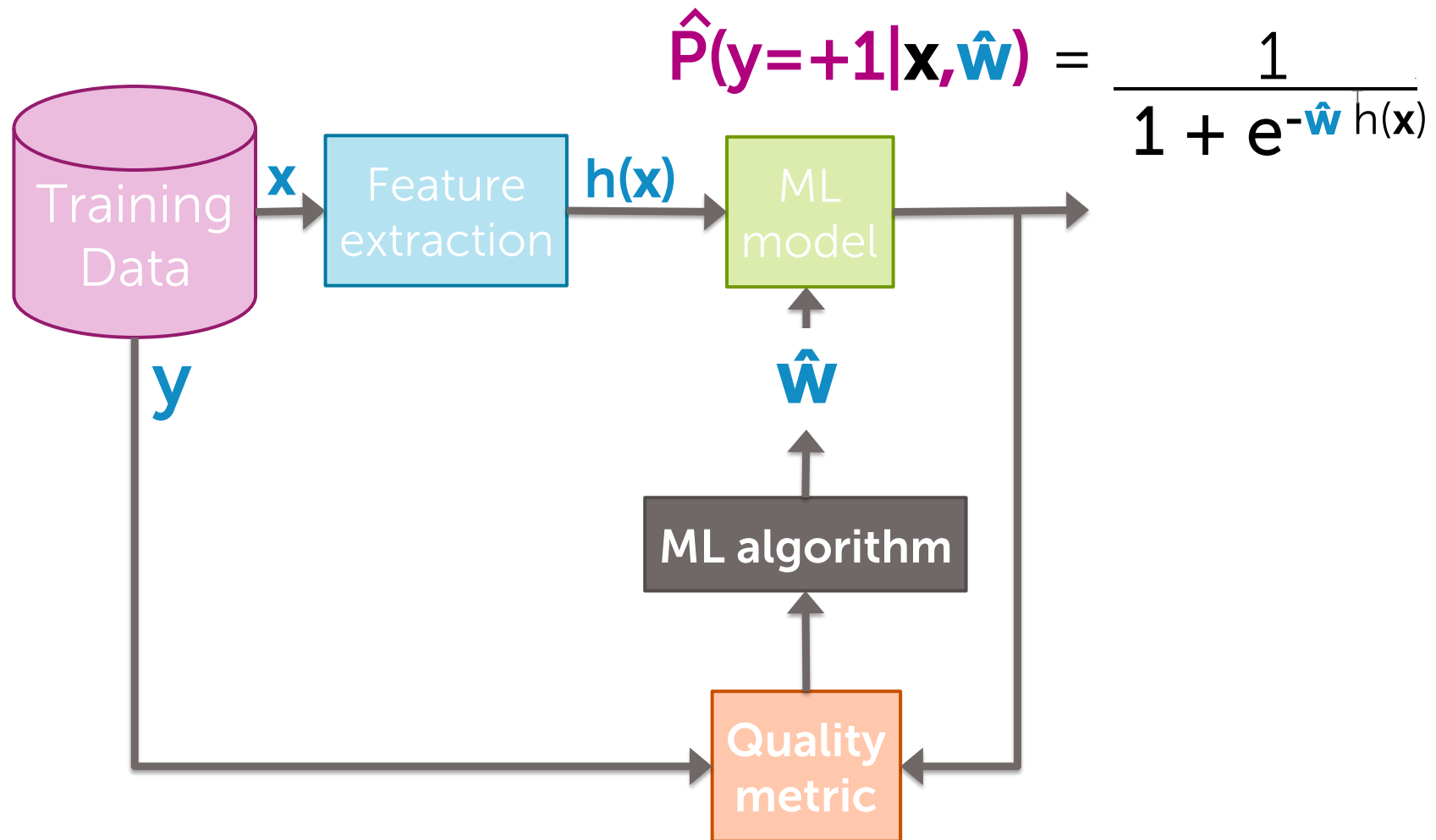
pick \mathbf{w} to make this fn. as large as possible





MOVE TO FULL BODY SHOT

Finding best linear classifier with gradient ascent





MOVE TO HEAD SHOT

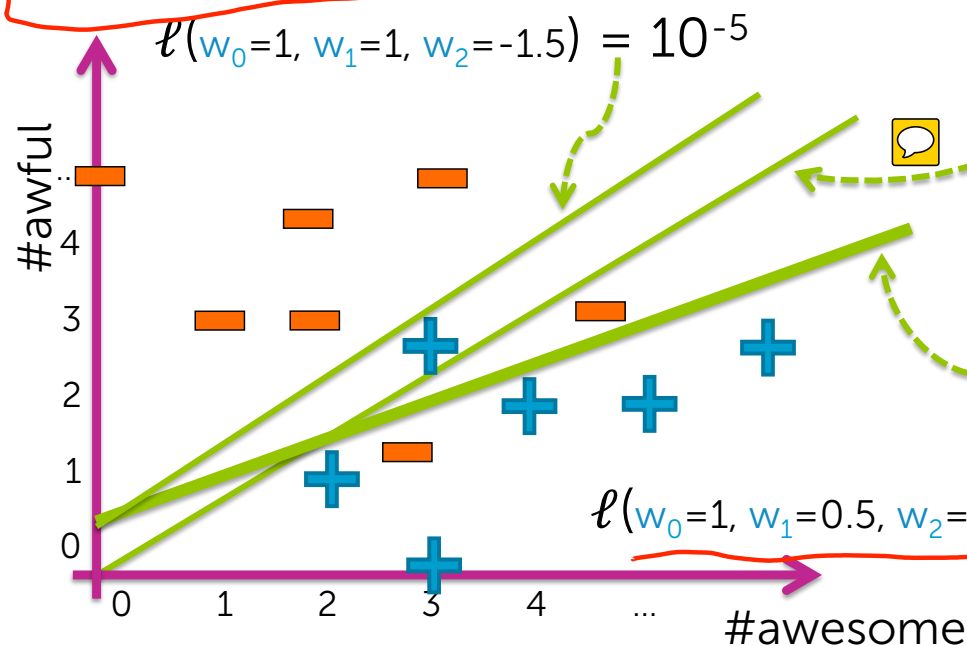
Find "best" classifier

Maximize likelihood over all possible w_0, w_1, w_2

$$\ell(\mathbf{w}) = \prod_{i=1}^N P(y_i | \mathbf{x}_i, \mathbf{w})$$

$$\ell(w_0=0, w_1=1, w_2=-1.5) = 10^{-6}$$

$$\ell(w_0=1, w_1=1, w_2=-1.5) = 10^{-5}$$



$$\ell(w_0=1, w_1=0.5, w_2=-1.5) = 10^{-4}$$

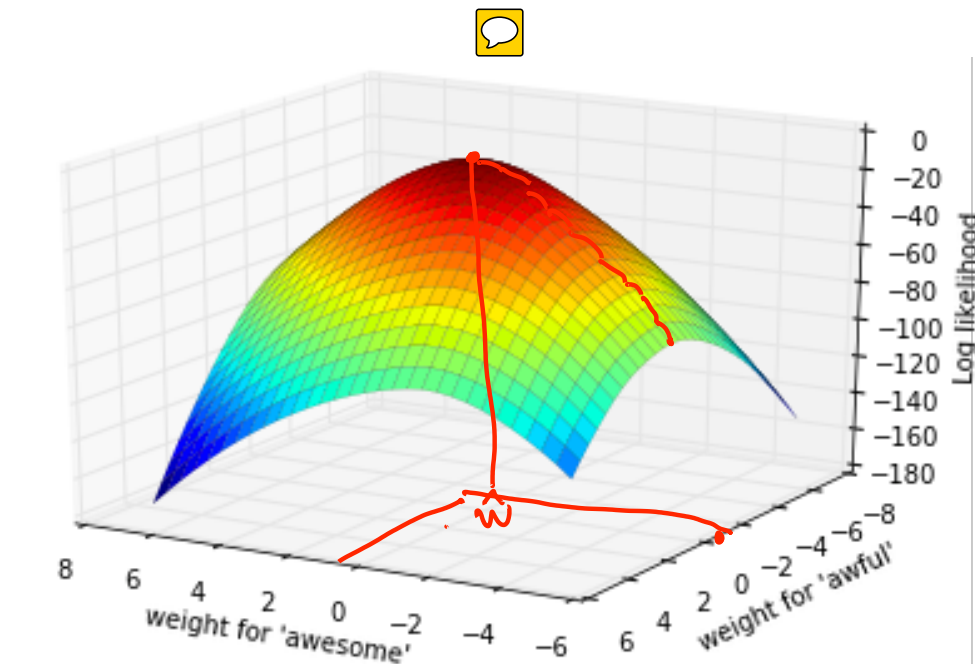
Best model:

Highest likelihood $\ell(\mathbf{w})$

$$\hat{\mathbf{w}} = (w_0=1, w_1=0.5, w_2=-1.5)$$

optimize with
gradient ascent

Maximizing likelihood




Maximize function over all possible w_0, w_1, w_2

$$\max_{w_0, w_1, w_2} \prod_{i=1}^N P(y_i \mid \mathbf{x}_i, \mathbf{w})$$

$\ell(w_0, w_1, w_2)$ is a function of 3 variables

No closed-form solution → use gradient ascent



MOVE TO FULL BODY SHOT

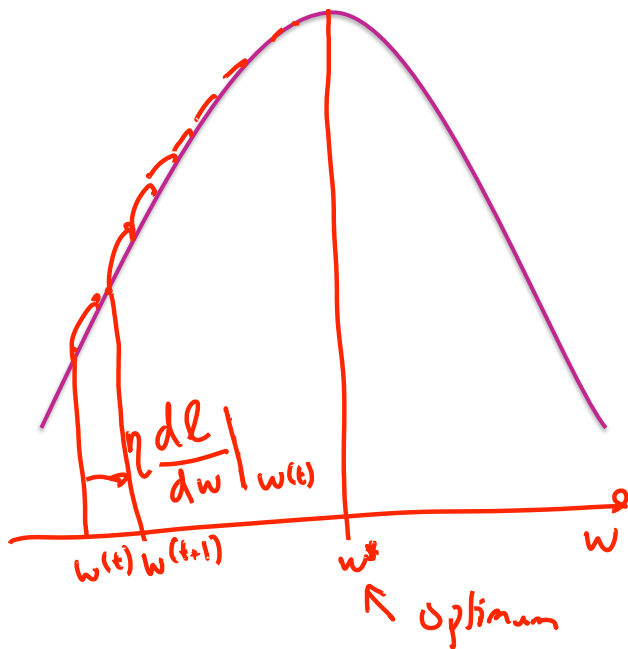


Review of gradient ascent



MOVE TO HEAD SHOT

Finding the max via hill climbing



Algorithm:

while not converged

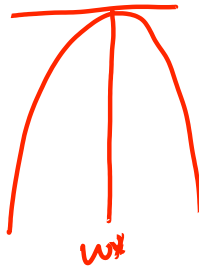
$$w^{(t+1)} \leftarrow w^{(t)} + \eta \frac{d\ell}{dw} \bigg|_{w^{(t)}}$$

step size

Convergence criteria

For convex functions,
optimum occurs when

$$\frac{d\ell}{dw} = 0$$



In practice, stop when

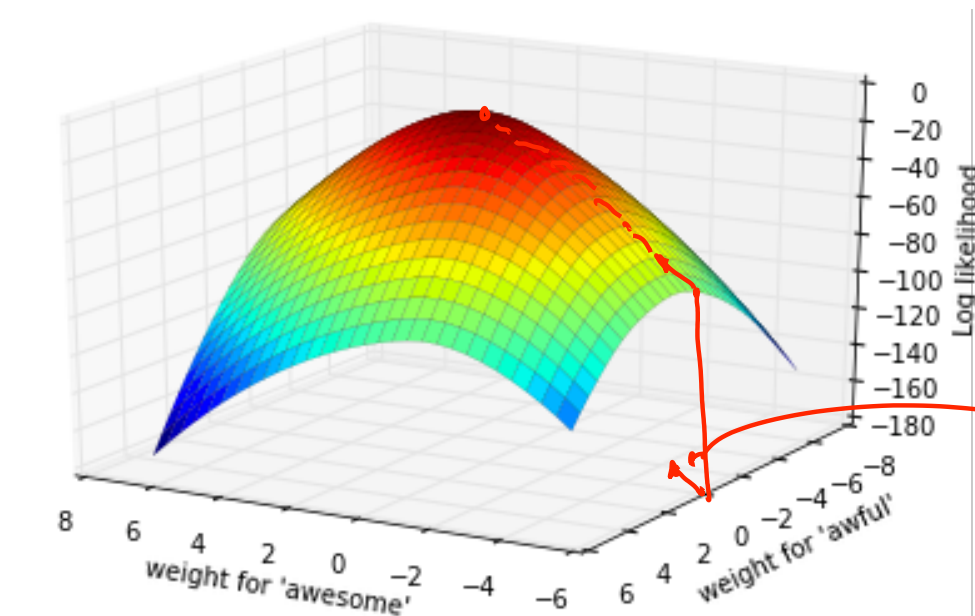
$$\left. \frac{d\ell}{dw} \right|_{w^{(t)}} < \epsilon$$

↑
tolerance

Algorithm:

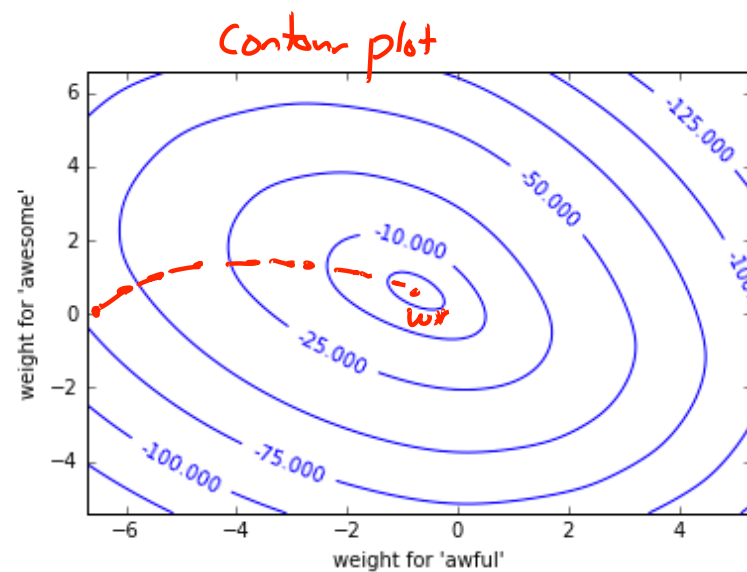
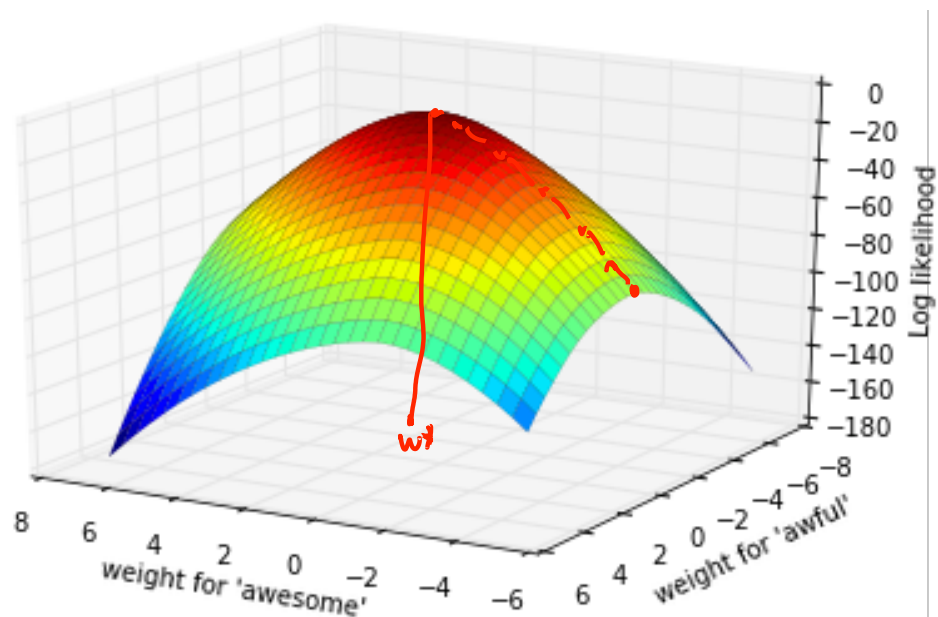
while not converged
 $w^{(t+1)} \leftarrow w^{(t)} + \eta \left. \frac{d\ell}{dw} \right|_{w^{(t)}}$

Moving to multiple dimensions: Gradients

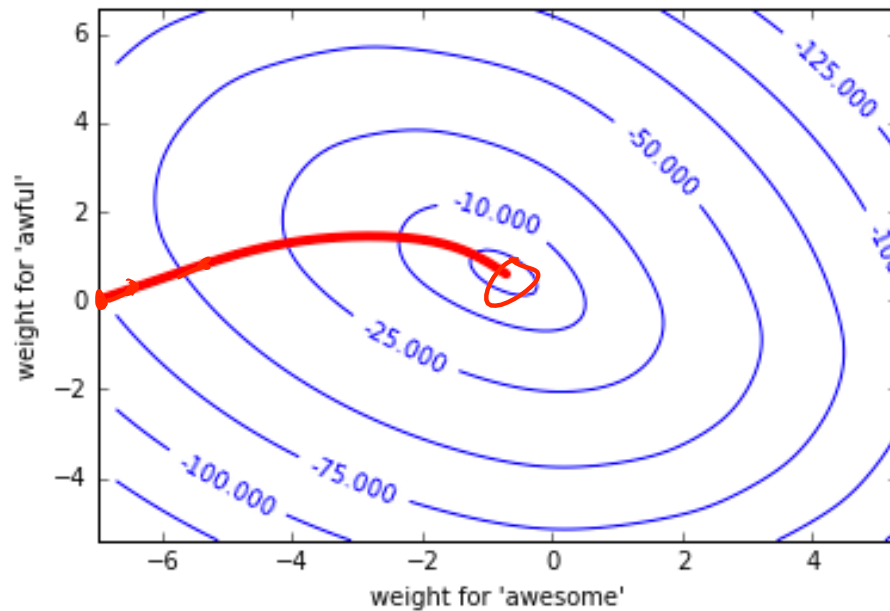


$$\nabla \ell(\mathbf{w}) = \begin{bmatrix} \frac{\partial \ell}{\partial w_0} \\ \frac{\partial \ell}{\partial w_1} \\ \vdots \\ \frac{\partial \ell}{\partial w_D} \end{bmatrix} \leftarrow D+1 \text{ dim vector}$$

Contour plots



Gradient ascent




Algorithm:

$w^{(0)} = 0$, random , or something smart.

while not converged

$$\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} + \eta \nabla \ell(\mathbf{w}^{(t)})$$

step size



MOVE TO FULL BODY SHOT



Learning algorithm for logistic regression



MOVE TO HEAD SHOT

Derivative of (log-)likelihood



Sum over data points

Feature value

Difference between truth and prediction

$$\frac{\partial \ell(\mathbf{w})}{\partial \mathbf{w}_j} = \sum_{i=1}^N h_j(\mathbf{x}_i) \left(\mathbb{1}[y_i = +1] - \underbrace{P(y = +1 \mid \mathbf{x}_i, \mathbf{w})}_{\text{predict } x_i \text{ is positive}} \right)$$

Indicator function:

$$\mathbb{1}[y_i = +1] = \begin{cases} 1 & \text{if } y_i = +1 \\ 0 & \text{if } y_i = -1 \end{cases}$$

Computing derivative

$$\frac{\partial \ell(\mathbf{w}^{(t)})}{\partial \mathbf{w}_j} = \sum_{i=1}^N h_j(\mathbf{x}_i) \left(\mathbb{1}[y_i = +1] - P(y = +1 \mid \mathbf{x}_i, \mathbf{w}^{(t)}) \right)$$

$\mathbf{w}^{(t)}$:

$w_0^{(t)}$	0
$w_1^{(t)}$	1
$w_2^{(t)}$	-2

$\frac{\partial \ell}{\partial w_1}$



$h_i(t) = \# \text{ awesome}$

x[1]	x[2]	y	P(y=+1 x _i ,w)	Contribution to derivative for w_1
2	1	+1	0.5	$2(1 - 0.5) = 1$
0	2	-1	0.02	$0(0 - 0.02) = 0$
3	3	-1	0.05	$3(0 - 0.05) = -0.15$
4	1	+1	0.88	$4(1 - 0.88) = 0.48$

Total derivative:

$$\frac{\partial \ell(\mathbf{w}^{(t)})}{\partial w_1} = 1 + 0 - 0.15 + 0.48 = 1.33$$

$$w_1^{(t+1)} = w_1^{(t)} + \eta \frac{\partial \ell(\mathbf{w}^{(t)})}{\partial w_1} \quad | \quad \eta = 0.1$$

$$= 1 + 0.1 \times 1.33 = 1.133$$

Derivative of (log-)likelihood: Interpretation

Sum over data points


Feature value

Difference between truth and prediction

$$\frac{\partial \ell(\mathbf{w})}{\partial \mathbf{w}_j} = \sum_{i=1}^N h_j(\mathbf{x}_i) \left(\mathbb{1}[y_i = +1] - P(y = +1 | \mathbf{x}_i, \mathbf{w}) \right)$$

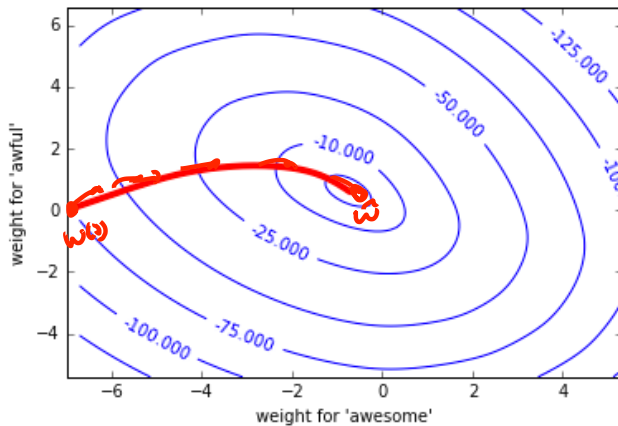
Δ_i

If $h_j(\mathbf{x}_i) = 1$:

	$P(y=+1 \mathbf{x}_i, \mathbf{w}) \approx 1$	$P(y=+1 \mathbf{x}_i, \mathbf{w}) \approx 0$
$y_i = +1$	$\Delta_i = (1 - 1) \approx 0$  \hookrightarrow don't change anything!	$\Delta_i \approx 1 \Rightarrow$ increase w_j \Rightarrow Score(x_i) becomes larger $\Rightarrow P(y=+1 \mathbf{x}_i, \mathbf{w})$ increases
$y_i = -1$	$\Delta_i = -1 \Rightarrow w_j$ to decrease \Rightarrow Score(x_i) decreases $\Rightarrow P(y=+1 \mathbf{x}_i, \mathbf{w})$ decrease	$\Delta_i \approx 0$ \Rightarrow don't change anything



Summary of gradient ascent for logistic regression



init $\mathbf{w}^{(1)} = 0$ (or randomly, or smartly), $t = 1$

while $\|\nabla \ell(\mathbf{w}^{(t)})\| > \epsilon$


for $j = 0, \dots, D$

$$\text{partial}[j] = \sum_{i=1}^N h_j(\mathbf{x}_i) \left(\mathbb{1}[y_i = +1] - \underbrace{P(y = +1 \mid \mathbf{x}_i, \mathbf{w}^{(t)})}_{\frac{1}{1 + e^{-\mathbf{w}^{(t)} \cdot \mathbf{h}(\mathbf{x}_i)}}} \right)$$

$$\mathbf{w}_j^{(t+1)} \leftarrow \mathbf{w}_j^{(t)} + \eta \text{partial}[j]$$

$$t \leftarrow t + 1$$

$$\frac{\partial \ell(\mathbf{w}^{(t)})}{\partial \mathbf{w}_j}$$



MOVE TO FULL BODY SHOT

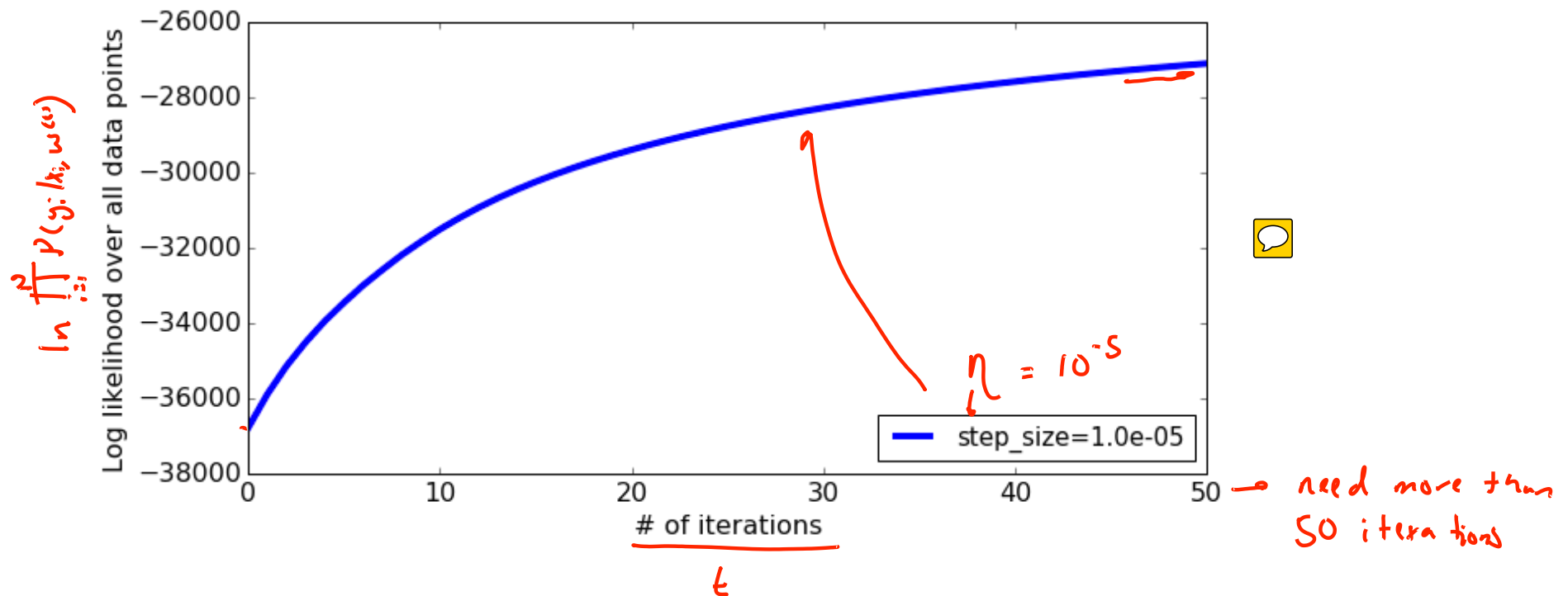


Choosing the step size η

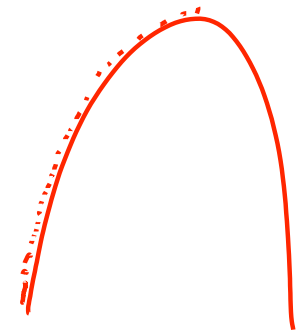
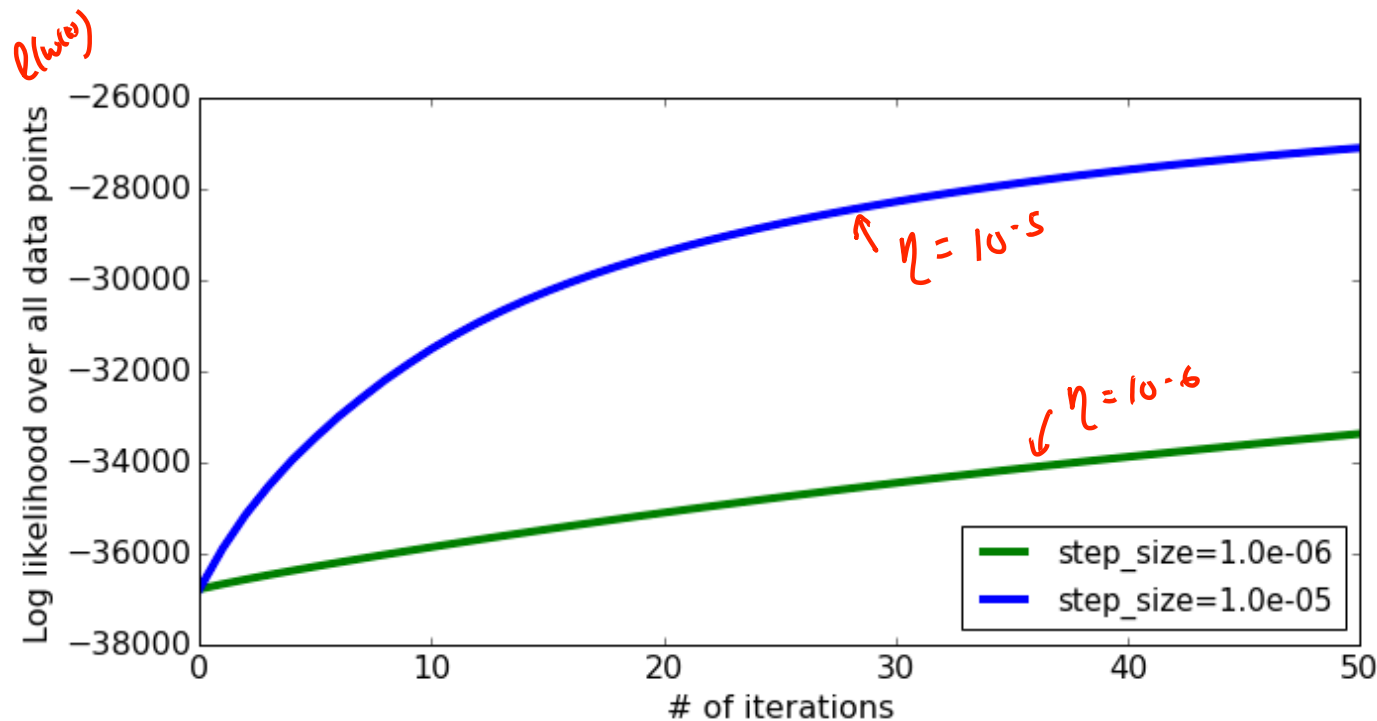


MOVE TO HEAD SHOT

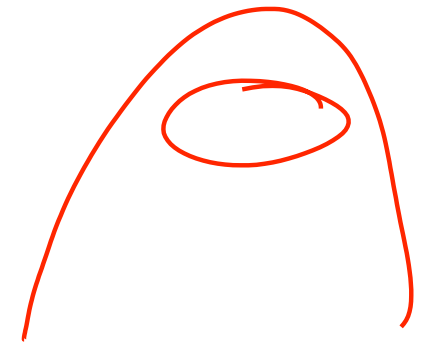
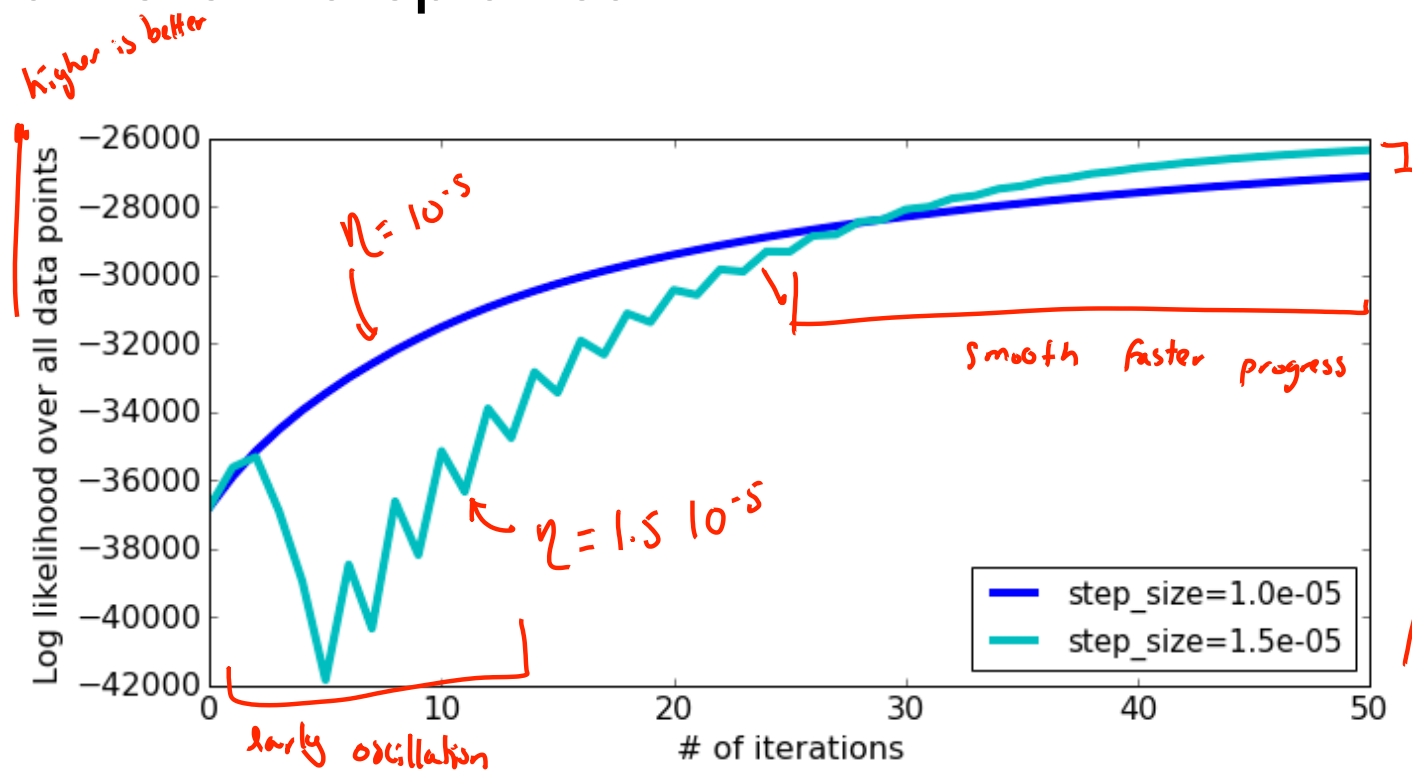
Learning curve: Plot quality (likelihood) over iterations



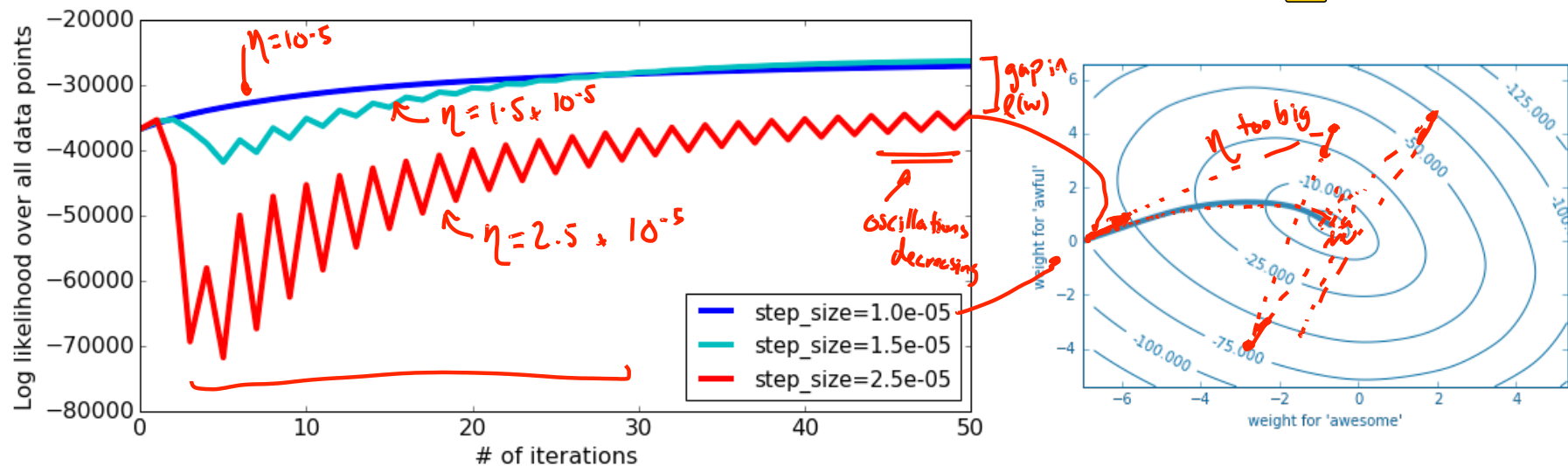
If step size is too small, can take a long time to converge



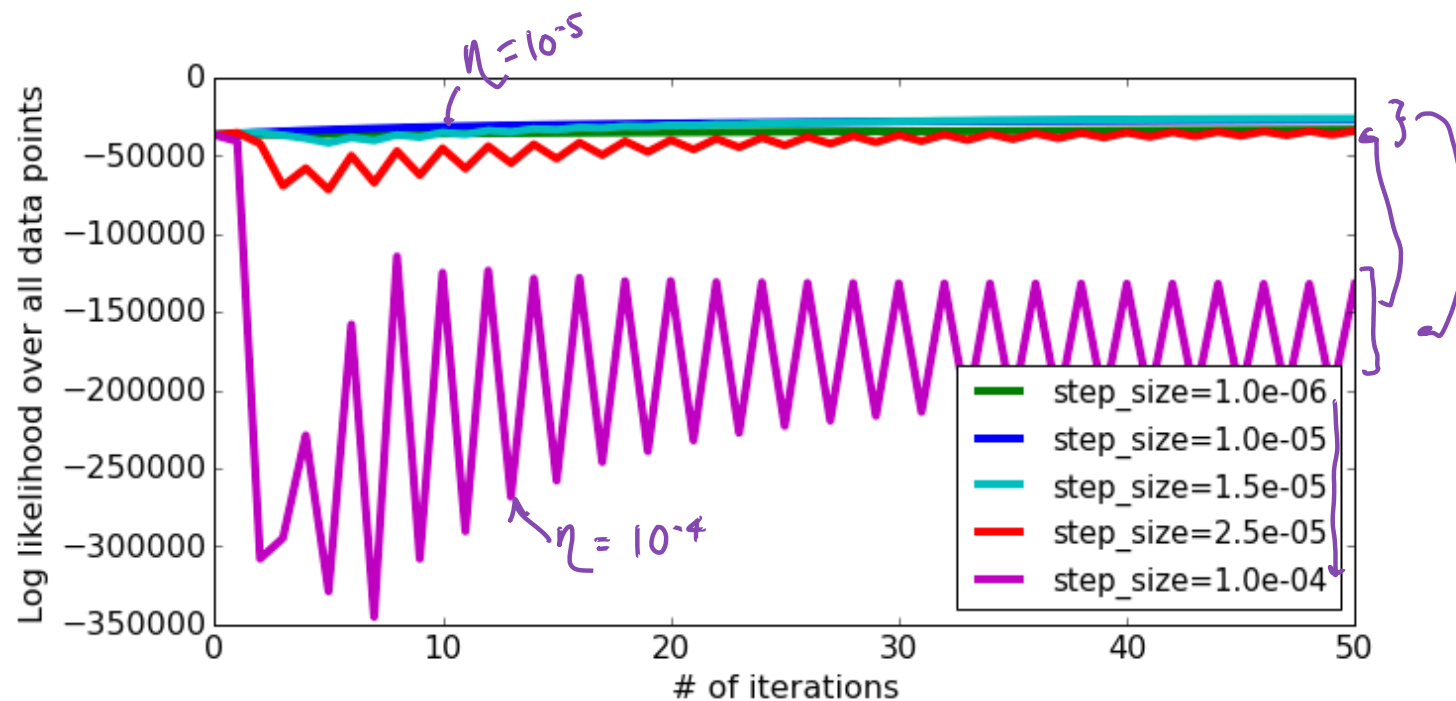
Compare converge with different step sizes



Careful with step sizes that are too large



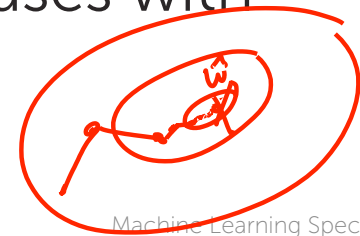
Very large step sizes can even
cause divergence or wild oscillations




Simple **rule of thumb** for picking step size η

- Unfortunately, picking step size requires a lot of trial and error ☹
- Try a several values, exponentially spaced
 - **Goal:** plot learning curves to
 - find one η that is too small (smooth but moving too slowly)
 - find one η that is too large (oscillation or divergence)
- Try values in between to find “best” η
 - ↳ exponentially space, pick one that leads best training data likelihood
- Advanced tip: can also try step size that decreases with iterations, e.g.,

$$\eta_t = \frac{\eta_0}{t}$$





MOVE TO FULL BODY SHOT

Deriving the gradient for logistic regression

**VERY
OPTIONAL**



MOVE TO HEAD SHOT

Log-likelihood function

- Goal: choose coefficients \mathbf{w} maximizing likelihood:

$$\ell(\mathbf{w}) = \prod_{i=1}^N P(y_i \mid \mathbf{x}_i, \mathbf{w})$$

- Math simplified by using log-likelihood – taking (natural) log:

$$\ell\ell(\mathbf{w}) = \ln \prod_{i=1}^N P(y_i \mid \mathbf{x}_i, \mathbf{w})$$

natural log

The log trick, often used in ML...

- Products become sums:

$$\ln a \cdot b = \ln a + \ln b \quad \left| \quad \ln \frac{a}{b} = \ln a - \ln b$$

- Doesn't change maximum!

– If $\hat{\mathbf{w}}$ maximizes $f(\mathbf{w})$:

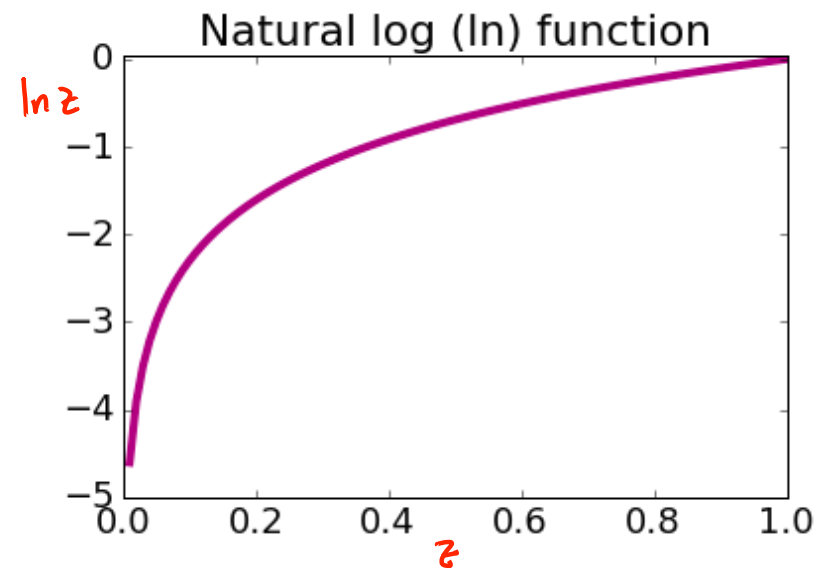
$$\hat{\mathbf{w}} = \arg \max_{\mathbf{w}} f(\mathbf{w})$$


the \mathbf{w} that makes $f(\mathbf{w})$ largest

– Then $\hat{\mathbf{w}}_{\ln}$ maximizes $\ln(f(\mathbf{w}))$:

$$\hat{\mathbf{w}}_{\ln} = \arg \max_{\mathbf{w}} \ln(f(\mathbf{w}))$$

$$\hat{\mathbf{w}} = \hat{\mathbf{w}}_{\ln}$$





Insert next title slide before Slide 52,
around 4:55 in
PL7_DerivingtheGradient_1stEdit

Expressing the log-likelihood

**VERY
OPTIONAL**

Using log to turn products into sums

$$\ln \prod_{i=1}^N f_i = \sum_{i=1}^N \ln f_i$$

- The log of the product of likelihoods becomes the sum of the logs:

$$\begin{aligned} \ell\ell(\mathbf{w}) &= \ln \prod_{i=1}^N P(y_i \mid \mathbf{x}_i, \mathbf{w}) \\ &= \sum_{i=1}^N \ln P(y_i \mid \mathbf{x}_i, \mathbf{w}) \end{aligned}$$

Rewriting log-likelihood

- For simpler math, we'll rewrite likelihood with indicators:

$$\begin{aligned}\ell\ell(\mathbf{w}) &= \sum_{i=1}^N \ln P(y_i \mid \mathbf{x}_i, \mathbf{w}) \\ &= \sum_{i=1}^N [\mathbb{1}[y_i = +1] \ln P(y = +1 \mid \mathbf{x}_i, \mathbf{w}) + \mathbb{1}[y_i = -1] \ln P(y = -1 \mid \mathbf{x}_i, \mathbf{w})]\end{aligned}$$

if $y_i = +1$


✓

○

✓

if $y_i = -1$

○



Insert next title slide before Slide 54,
around 7:33 in
PL7_DerivingtheGradient_1stEdit



Deriving probability that $y = -1$ given x

**VERY
OPTIONAL**


Logistic regression model: $P(y=-1|\mathbf{x}, \mathbf{w})$

- Probability model predicts $y=+1$:

$$P(y=+1|\mathbf{x}, \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{h}(\mathbf{x})}}$$

- Probability model predicts $y=-1$:

$$\begin{aligned} P(y=-1|\mathbf{x}, \mathbf{w}) &= 1 - P(y=+1|\mathbf{x}, \mathbf{w}) = 1 - \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{h}(\mathbf{x})}} \\ &= \frac{1 + e^{-\mathbf{w}^T \mathbf{h}(\mathbf{x})} - 1}{1 + e^{-\mathbf{w}^T \mathbf{h}(\mathbf{x})}} = \frac{e^{-\mathbf{w}^T \mathbf{h}(\mathbf{x})}}{1 + e^{-\mathbf{w}^T \mathbf{h}(\mathbf{x})}} \end{aligned}$$



Insert next title slide before Slide 55,
around 9:15 in
PL7_DerivingtheGradient_1stEdit

Rewriting the log-likelihood



Plugging in logistic function for 1 data point

$$P(y = +1 \mid \mathbf{x}, \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{w}^\top h(\mathbf{x})}} \quad P(y = -1 \mid \mathbf{x}, \mathbf{w}) = \frac{e^{-\mathbf{w}^\top h(\mathbf{x})}}{1 + e^{-\mathbf{w}^\top h(\mathbf{x})}}$$

$$\ell\ell(\mathbf{w}) = \mathbb{1}[y_i = +1] \ln P(y = +1 \mid \mathbf{x}_i, \mathbf{w}) + \mathbb{1}[y_i = -1] \ln P(y = -1 \mid \mathbf{x}_i, \mathbf{w})$$

$$= \mathbb{1}[y_i = +1] \ln \frac{1}{1 + e^{-\mathbf{w}^\top h(\mathbf{x}_i)}} + (1 - \mathbb{1}[y_i = +1]) \ln \frac{e^{-\mathbf{w}^\top h(\mathbf{x}_i)}}{1 + e^{-\mathbf{w}^\top h(\mathbf{x}_i)}}$$

$$= -\mathbb{1}[y_i = +1] \ln(1 + e^{-\mathbf{w}^\top h(\mathbf{x}_i)}) + (1 - \mathbb{1}[y_i = +1]) [-\mathbf{w}^\top h(\mathbf{x}_i) - \ln(1 + e^{-\mathbf{w}^\top h(\mathbf{x}_i)})]$$

$$= - (1 - \mathbb{1}[y_i = +1]) \mathbf{w}^\top h(\mathbf{x}_i) - \ln(1 + e^{-\mathbf{w}^\top h(\mathbf{x}_i)})$$

Simpler form


$$\ln e^a = a$$

$$\mathbb{1}[y_i = -1] = 1 - \mathbb{1}[y_i = +1]$$

$$\ln \frac{1}{1 + e^{-\mathbf{w}^\top h(\mathbf{x}_i)}} = -\ln(1 + e^{-\mathbf{w}^\top h(\mathbf{x}_i)})$$

$$\ln \frac{e^{-\mathbf{w}^\top h(\mathbf{x}_i)}}{1 + e^{-\mathbf{w}^\top h(\mathbf{x}_i)}} =$$

$$\underbrace{\ln e^{-\mathbf{w}^\top h(\mathbf{x}_i)}}_{-\mathbf{w}^\top h(\mathbf{x}_i)} - \ln(1 + e^{-\mathbf{w}^\top h(\mathbf{x}_i)})$$



Insert next title slide before Slide 56,
around 16:56 in
PL7_DerivingtheGradient_1stEdit



Deriving gradient of log-likelihood

**VERY
OPTIONAL**

Gradient for 1 data point

$$\ell(\mathbf{w}) = -(1 - \mathbb{1}[y_i = +1])\mathbf{w}^\top h(\mathbf{x}_i) - \ln(1 + e^{-\mathbf{w}^\top h(\mathbf{x}_i)})$$

$$\frac{\partial \ell}{\partial w_j} = -(1 - \mathbb{1}[y_i = +1]) \frac{\partial w^\top h(\mathbf{x}_i)}{\partial w_j} - \frac{\partial \ln(1 + e^{-w^\top h(\mathbf{x}_i)})}{\partial w_j}$$

$$= -(1 - \mathbb{1}[y_i = +1]) h_j(\mathbf{x}_i) + h_j(\mathbf{x}_i) P(y = -1 | \mathbf{x}_i, \mathbf{w})$$

$$= h_j(\mathbf{x}_i) [\mathbb{1}[y_i = +1] - P(y = +1 | \mathbf{x}_i, \mathbf{w})]$$

$$\begin{aligned} \frac{\partial w^\top h(\mathbf{x}_i)}{\partial w_j} &= h_j(\mathbf{x}_i) \\ \hline \frac{\partial \ln(1 + e^{-w^\top h(\mathbf{x}_i)})}{\partial w_j} &= -h_j(\mathbf{x}_i) \frac{e^{-w^\top h(\mathbf{x}_i)}}{1 + e^{-w^\top h(\mathbf{x}_i)}} \\ &= -h_j(\mathbf{x}_i) P(y = -1 | \mathbf{x}_i, \mathbf{w}) \end{aligned}$$


Finally, gradient for all data points

- Gradient for one data point:

$$h_j(\mathbf{x}_i) \left(\mathbb{1}[y_i = +1] - P(y = +1 \mid \mathbf{x}_i, \mathbf{w}) \right)$$

- Adding over data points:

$$\frac{\partial \ell}{\partial w_j} = \sum_{i=1}^N h_j(x_i) \left(\mathbb{1}[y_i = +1] - P(y = +1 \mid x_i, w) \right)$$

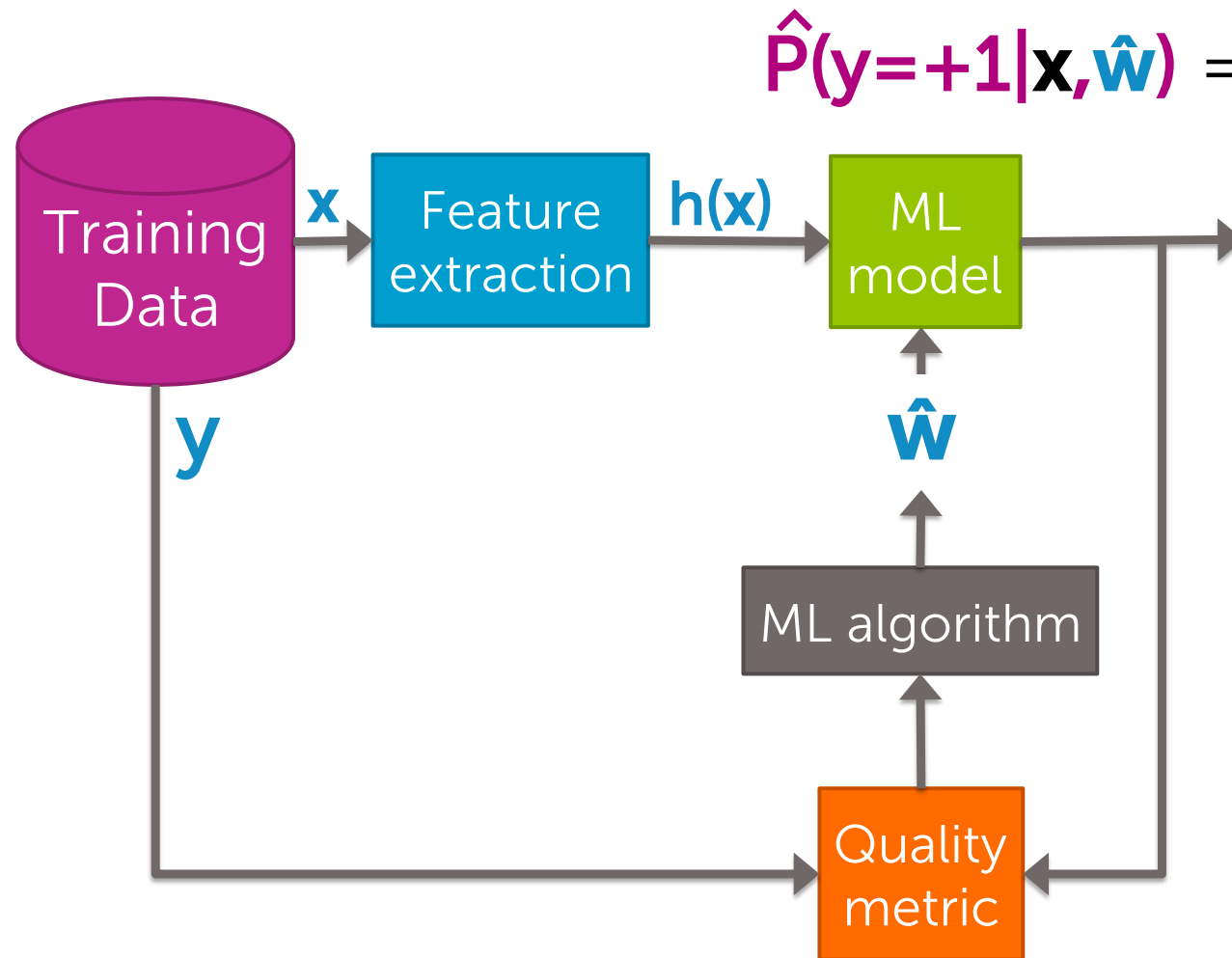


MOVE TO FULL BODY SHOT

Summary of logistic regression classifier



MOVE TO HEAD SHOT



$$\hat{P}(y=+1|\mathbf{x},\hat{\mathbf{w}}) = \frac{1}{1 + e^{-\hat{\mathbf{w}}^T h(\mathbf{x})}}$$



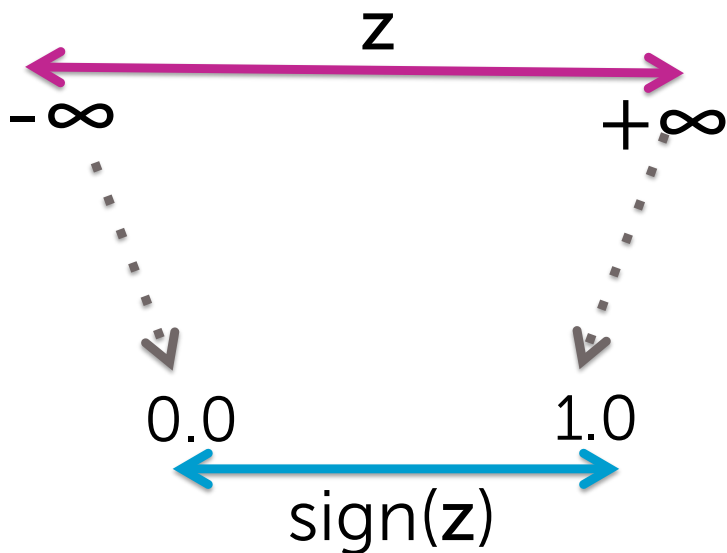
MOVE TO FULL BODY SHOT



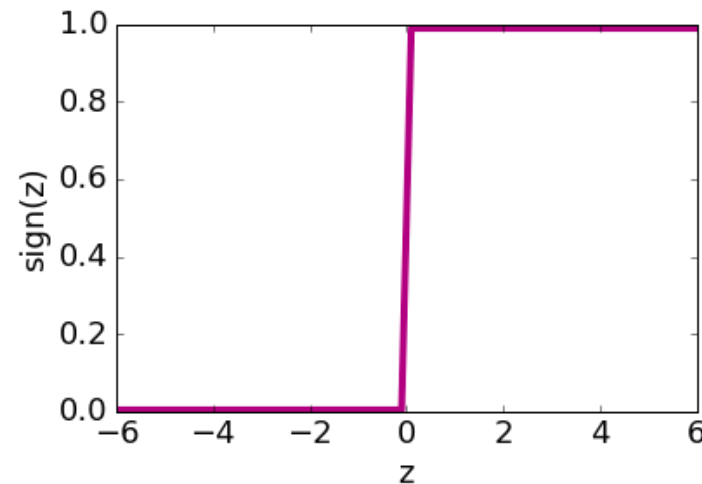
What you can do now...

- Measure quality of a classifier using the likelihood function
- Interpret the likelihood function as the probability of the observed data
- Learn a logistic regression model with gradient descent
- (Optional) Derive the gradient descent update rule for logistic regression

Simplest link function: $\text{sign}(z)$



$$\text{sign}(z) = \begin{cases} +1 & \text{if } z \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

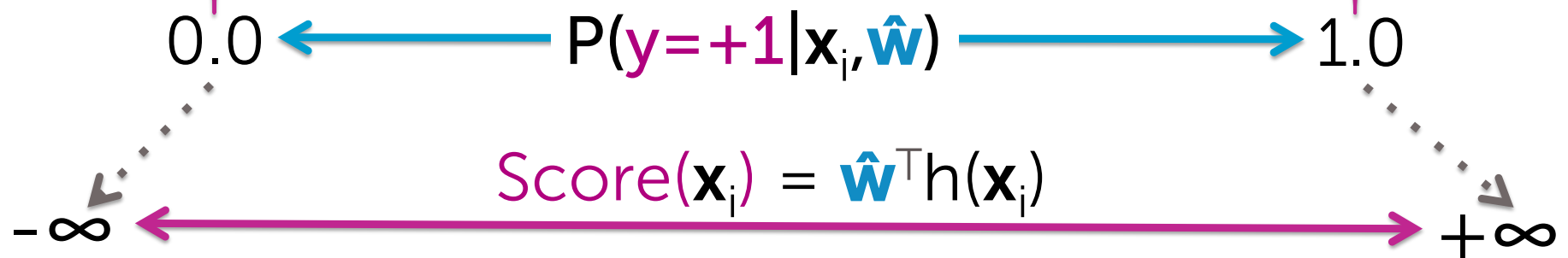


But, $\text{sign}(z)$ only outputs -1 or +1,
no probabilities in between

Finding best coefficients

x[1] = #awesome	x[2] = #awful	y = sentiment
0	2	-1
3	3	-1
2	4	-1
0	3	-1
0	1	-1

x[1] = #awesome	x[2] = #awful	y = sentiment
2	1	+1
4	1	+1
1	1	+1
2	1	+1



Quality metric: probability of data

$$\hat{P}(y=+1|\mathbf{x}, \hat{\mathbf{w}}) = \frac{1}{1 + e^{-\hat{\mathbf{w}}^T \mathbf{h}(\mathbf{x})}}$$

x[1] = #awesome	x[2] = #awful	y = sentiment
2	1	+1



If model good, should predict



Increase probability $y=+1$ when



Choose \mathbf{w} to make

x[1] = #awesome	x[2] = #awful	y = sentiment
0	2	-1



If model good, should predict



Increase probability $y=-1$ when



Choose \mathbf{w} to make

Maximizing likelihood (probability of data)

Data point	$x[1]$	$x[2]$	y	Choose w to maximize
\mathbf{x}_1, y_1	2	1	+1	
\mathbf{x}_2, y_2	0	2	-1	
\mathbf{x}_3, y_3	3	3	-1	
\mathbf{x}_4, y_4	4	1	+1	
\mathbf{x}_5, y_5	1	1	+1	
\mathbf{x}_6, y_6	2	4	-1	
\mathbf{x}_7, y_7	0	3	-1	
\mathbf{x}_8, y_8	0	1	-1	
\mathbf{x}_9, y_9	2	1	+1	

Must combine into single measure of quality



Learn logistic regression model with maximum likelihood estimation (MLE)

- Choose coefficients \mathbf{w} that maximize likelihood:

$$\prod_{i=1}^N P(y_i \mid \mathbf{x}_i, \mathbf{w})$$

- No closed-form solution \rightarrow use gradient ascent