Logistic Regression

image: Colin Behrens

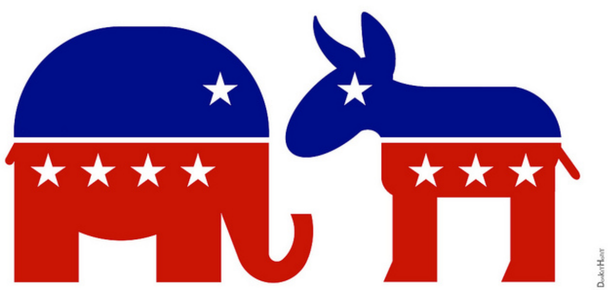
Will Monroe
August 14, 2017

with materials by
Mehran Sahami
and Chris Piech

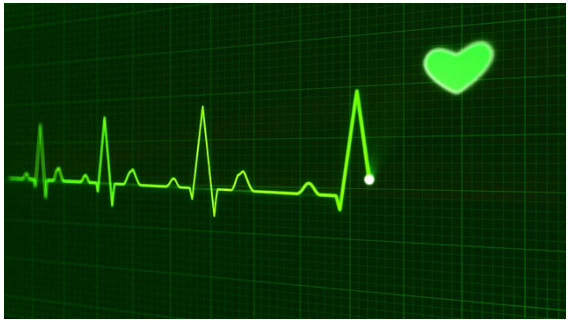
Announcement: Problem Set #6

Due **this Wednesday, August 16** (before class).

6 problems
(#6 involves serious coding!)



Congressional voting



Heart disease diagnosis

No late days!

Announcements: Final exam

This Saturday, August 19, 12:15-3:15pm in NVIDIA Auditorium

Two pages (both sides) of notes

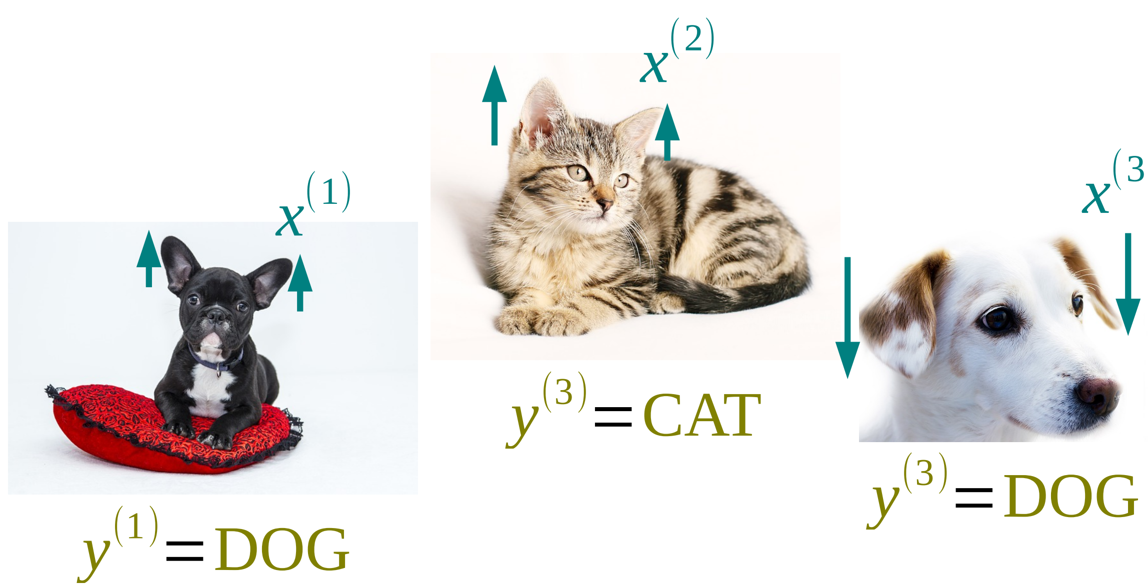
Comprehensive—material that was on the midterm will also be tested

Review session:
Wednesday, August 16, 2:30-3:20pm
in **Huang 18 (location change!)**

Review: Classification

The most basic machine learning task:
predict a **label** from a vector of **features**.

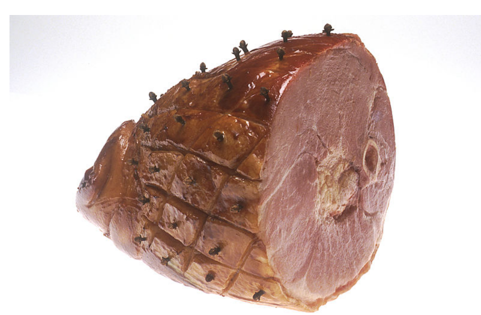
$$\hat{y} = \arg\max_{y} P\left(Y = y \mid \vec{X} = \vec{x}\right)$$



Review: Naïve Bayes

A classification algorithm using the assumption that features are **conditionally independent** given the label.

$$\hat{y} = \arg\max_{y} \hat{P}(Y = y) \prod_{j} \hat{P}(X_j = x_j | Y = y)$$



images: (left) Virginia State Parks; (right) Renee Comet

Review: Three secret ingredients

1. Maximum likelihood or maximum a posteriori for conditional probabilities.

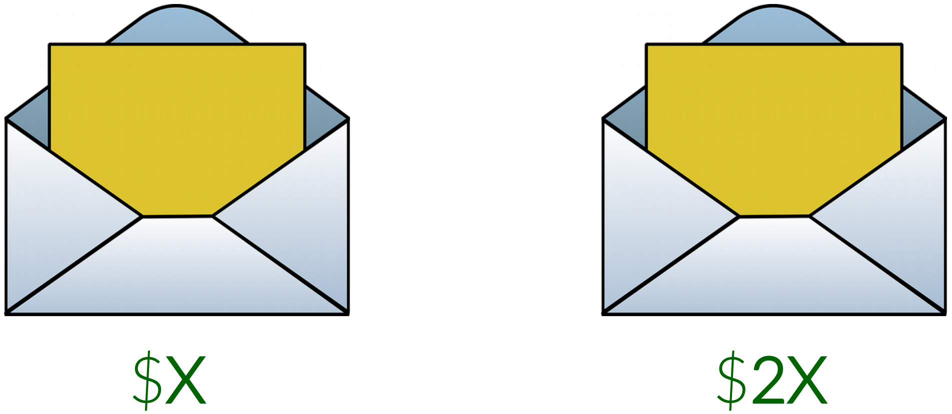
$$\hat{P}(X_j = x_j | Y = y) = \frac{\#(X_j = x_j, Y = y)[+1]}{\#(Y = y)[+2]}$$

2. "Naïve Bayes assumption": features are independent conditioned on the label.

$$\hat{P}(\vec{X} = \vec{x} | Y = y) = \prod_j \hat{P}(X_j = x_j | Y = y)$$

3. (Take logs for numerical stability.)

Two envelopes



Y = amount in envelope chosen

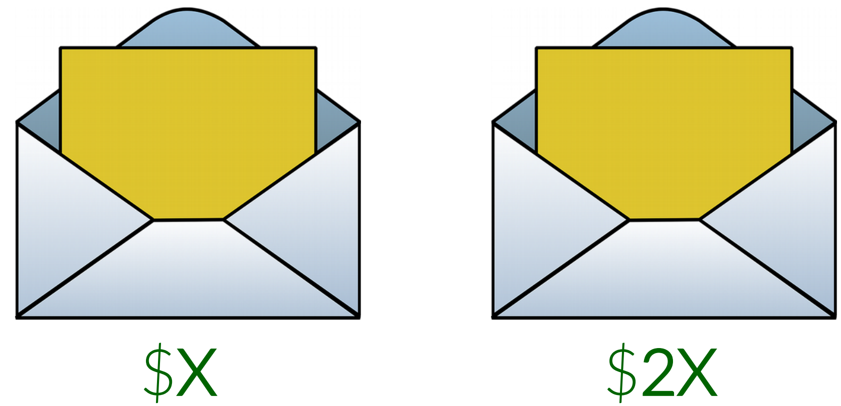
$$E[W|\text{stay}] = Y$$

$$E[W|\text{switch}] = \frac{Y}{2}\cdot 0.5 + 2Y\cdot 0.5$$

$$= \frac{5}{4}Y \qquad ???$$

Two envelopes: A resolution

"I'm trying to think: how likely is it that you would have put $40 in an envelope?



Y = y: amount in envelope chosen

$$E[W|Y=y,\text{stay}]=y$$

not necessarily 0.5!

$$E[W|Y=y,\text{switch}]=\frac{y}{2}\cdot P\left(X=\frac{y}{2}|Y=y\right)+2y\cdot P(X=y|Y=y)$$

prior

$$P(X=y|Y=y)=\frac{P(Y=y|X=y)P(X=y)}{P(Y=y|X=y)P(X=y)+P(Y=y|X\neq y)P(X\neq y)}$$

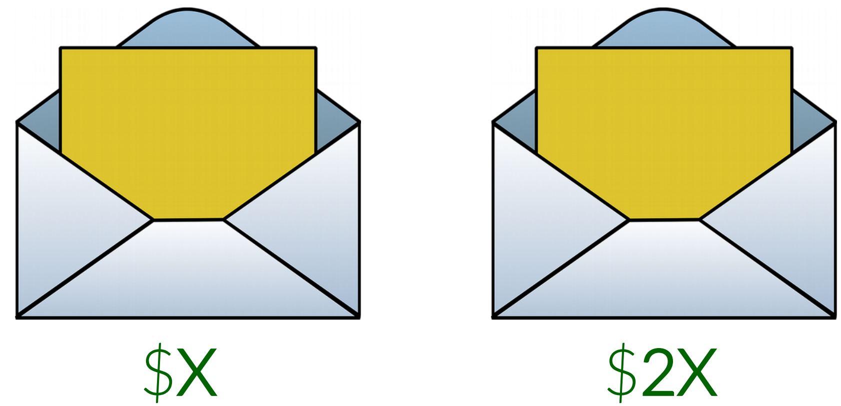
$$=\frac{0.5P(X=y)}{0.5P(X=y)+0.5P(X=y/2)}$$

$$=\frac{P(X=y)}{P(X=y)+P(X=y/2)}$$

$$P(X=y/2|Y=y)=1-P(X=y|Y=y)$$

Two envelopes: A resolution

"I'm trying to think: how likely is it that you would have put \$40 in an envelope?



Y = y: amount in envelope chosen

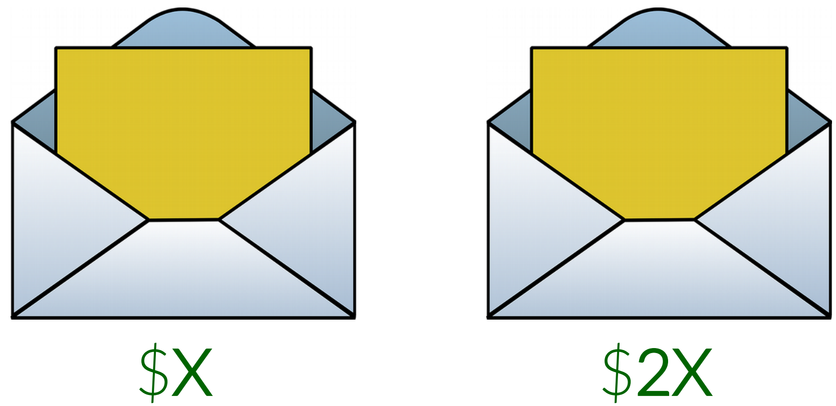
$$E[W|Y=y,\text{stay}]=y$$

$$E[W|Y=y,\text{switch}]=\frac{y}{2}\cdot P\left(X=\frac{y}{2}\middle|Y=y\right)+2y\cdot P(X=y|Y=y)$$

What if y = **\$20.01**?

Two envelopes: A resolution

"I'm trying to think: how likely is it that you would have put \$40 in an envelope?



Y = y: amount in envelope chosen

$$E[W|Y=y,\text{stay}]=y$$

$$E[W|Y=y,\text{switch}]=\frac{y}{2}\cdot P\left(X=\frac{y}{2}\middle|Y=y\right)+2y\cdot P(X=y|Y=y)$$

What if y = **\$20.01**?

Unless…

(the dreaded half-cent)

Unless...

1810 1/2c Classic Head Half Cent SEMI KEY DATE rare variety old type coin money

$99.00

Top Rated Plus

Buy It Now

(ebay.com)

Odds

The ratio of the probability of an event happening to the probability of it not happening:

$$o_f = \frac{P(E)}{P(E^C)} = \frac{P(E)}{1 - P(E)}$$

Probability	Odds	
1/10	1/9	“9:1 (against)”
1/3	1/2	“2:1 (against)”
1/2	1/1	“even odds”
2/3	2	“2:1 on” / “1:2”
9/10	9	“9:1 on” / “1:9”

Odds and probability

$$P(E) = p$$

$$o_f = \frac{P(E)}{1 - P(E)} = \frac{p}{1 - p}$$

$$o_f(1 - p) = p$$

$$o_f - p\,o_f = p$$

$$o_f = p(o_f + 1)$$

$$p = \frac{o_f}{o_f + 1} = \frac{1}{1 + \frac{1}{o_f}}$$

Log odds

all probabilities
(except 0 and 1)

$$p = P(E)$$

$$o_f = \frac{p}{1-p}$$

$$z = \log_2 o_f$$

p	$\frac{1}{17}$	$\frac{1}{9}$	$\frac{1}{5}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{4}{5}$	$\frac{8}{9}$	$\frac{16}{17}$
o_f	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	1	2	4	8	16
z	-4	-3	-2	-1	0	1	2	3	4

all real numbers

base 2 for simplicity—
on this slide only!

The logistic function

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

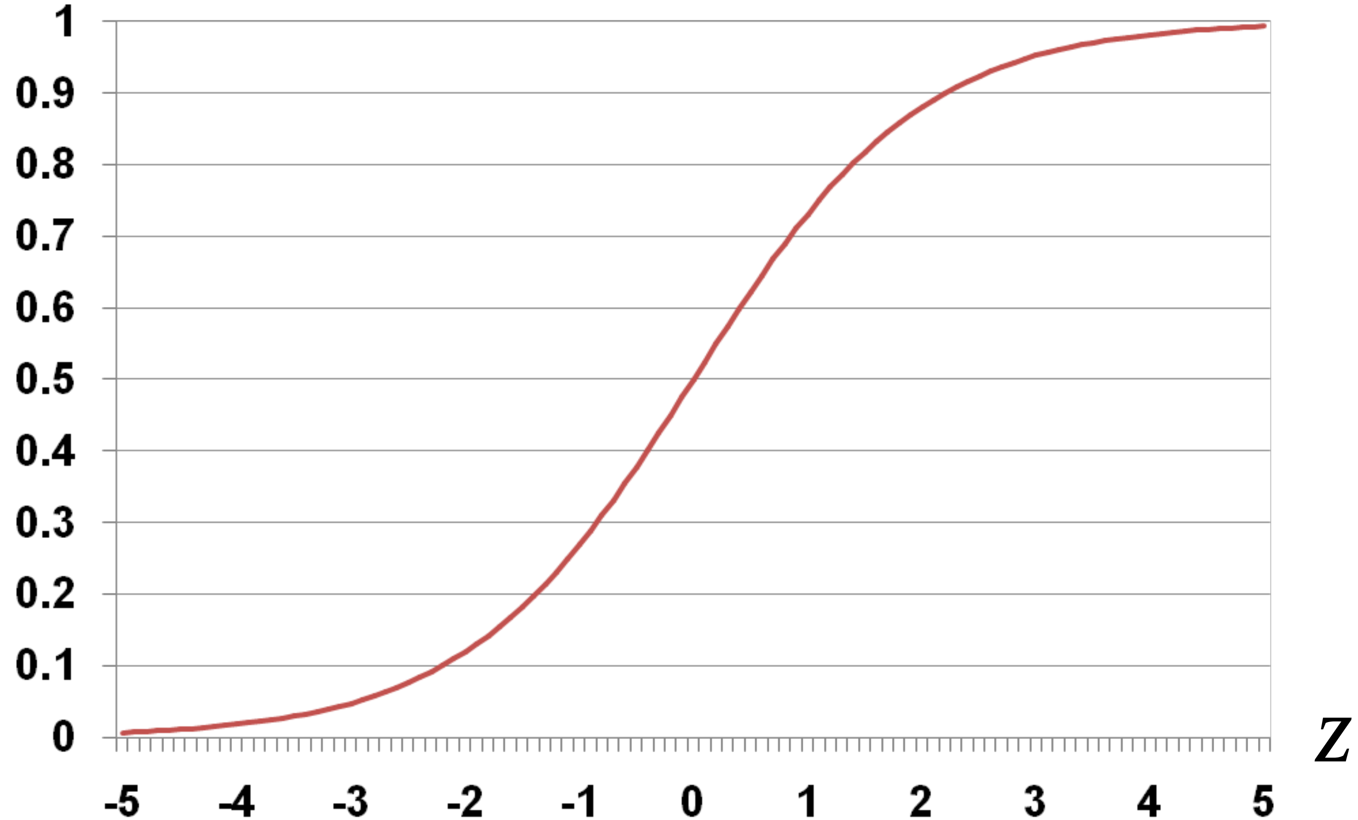
$$z = \log o_f$$

$$p = \frac{o_f}{o_f + 1} = \frac{1}{1 + \frac{1}{o_f}}$$

$$= \frac{1}{1+e^{-\log(o_f)}}$$

$$= \frac{1}{1+e^{-z}}$$

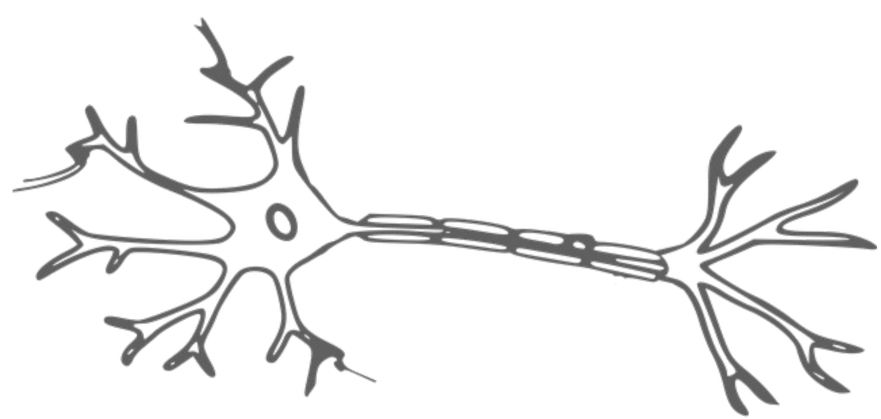
$$= \sigma(z)$$



Logistic regression

A classification algorithm using the assumption that **log odds** are a linear function of the features.

$$\hat{y} = \arg\max_{y} \frac{1}{1 + e^{-\vec{\theta}^{T}\vec{x}}}$$



Logistic regression assumption

$$P(Y=1|\vec{X}=\vec{x})=\sigma(\vec{\theta}^T\vec{x})=\frac{1}{1+e^{-\vec{\theta}^T\vec{x}}}$$

or in other words:

$$p=\sigma(z)$$

$$z=\log o_f$$

$$\vec{\theta}^T\vec{x}=\log o_f(Y=1|\vec{X}=\vec{x})$$

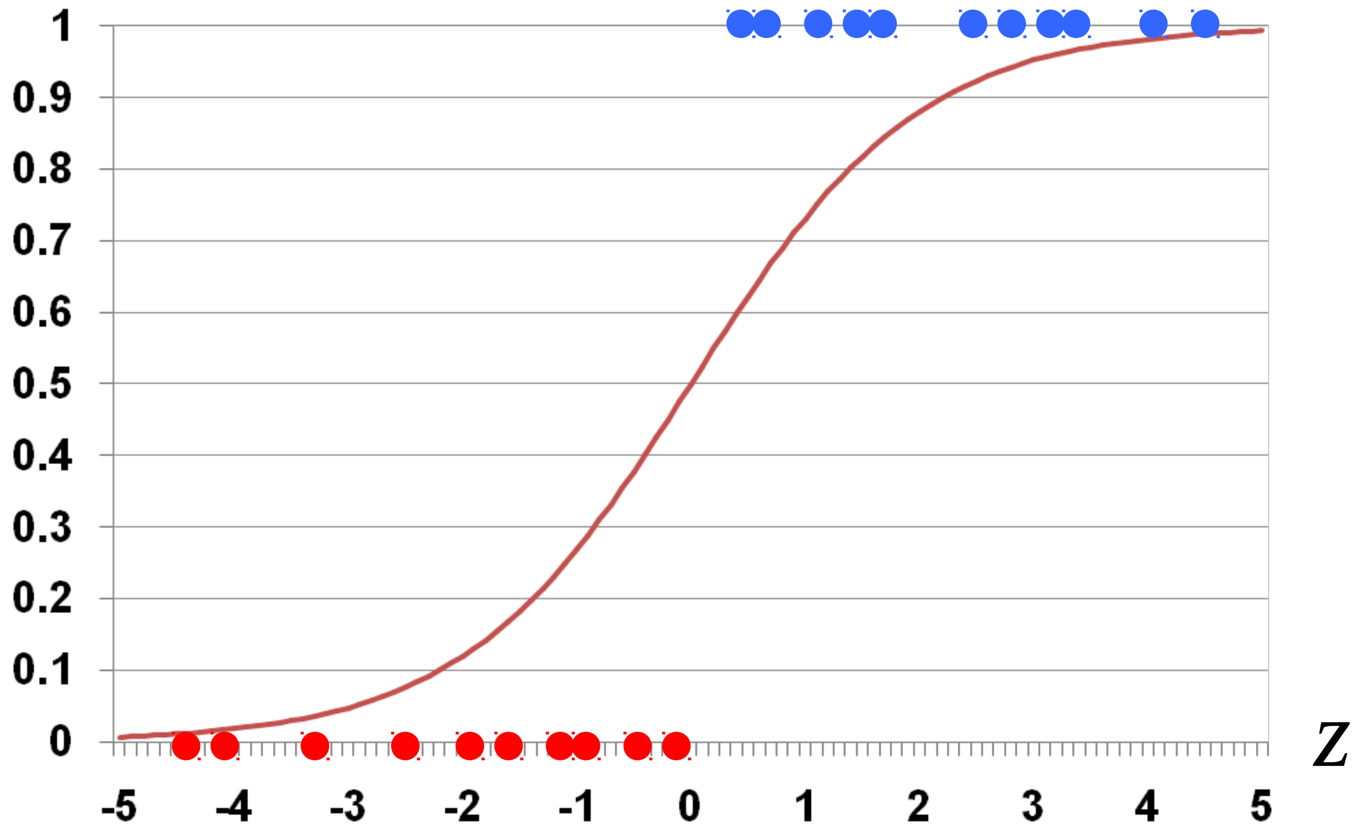
$$\vec{\theta}^T\vec{x}=\vec{\theta}\cdot\vec{x}=\theta_0\cdot 1+\theta_1 x_1+\theta_2 x_2+\cdots+\theta_m x_m$$

$$=\sum_{i=0}^{m}\theta_i x_i$$

$$(x_0=1)$$

Predicting 0/1 with the logistic

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



Logistic regression: Pseudocode

```
initialize: θ = [0, 0, …, 0]   (m elements)

repeat many times:
    gradient = [0, 0, …, 0]   (m elements)

    for each training example (x^(i), y^(i)):
        for j = 0 to m:

            gradient[j] += [y^(i) − σ(θ^T x^(i))] x_j^(i)

    for j = 0 to m:
        θ[j] += η * gradient[j]

return θ
```

$$\text{gradient}[j] \mathrel{+}= \left[y^{(i)} - \sigma\left(\vec{\theta}^T \vec{x}^{(i)}\right)\right] \vec{x}_j^{(i)}$$

Break time!

Where's the "learning"?

$$P(Y=1|\vec{X}=\vec{x})=\sigma(\vec{\theta}^T\vec{x})$$

↑

all of the model's "intelligence"
is in the choice of θ

Review: How to—MLE

1. Compute the likelihood.

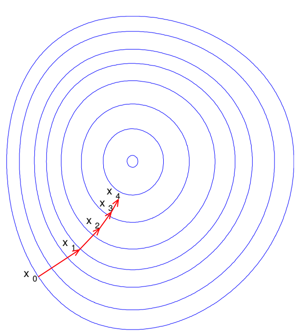
$$L(\theta) = P(X_1, \ldots, X_n | \theta)$$

2. Take its log.

$$LL(\theta) = \log L(\theta)$$

3. Maximize this as a function of the parameters.

$$\frac{d}{d\theta} LL(\theta) = 0$$



MLE for logistic regression

1. Compute the likelihood.

$$L(\vec{\theta}) = P(X^{(1)}, \ldots, X^{(n)}, Y^{(1)}, \ldots, Y^{(n)} | \theta)$$

$$= \prod_{i=1}^{n} P(X^{(i)}, Y^{(i)} | \theta)$$

$$= \prod_{i=1}^{n} P(Y^{(i)} | X^{(i)}, \theta) P(X^{(i)} | \theta)$$

$$= \prod_{i=1}^{n} p^{y^{(i)}} (1-p)^{1-y^{(i)}} P(X^{(i)})$$

$$= \prod_{i=1}^{n} \sigma(\vec{\theta}^T \vec{x}^{(i)})^{y^{(i)}} [1-\sigma(\vec{\theta}^T \vec{x}^{(i)})]^{1-y^{(i)}} P(X^{(i)})$$

MLE for logistic regression

3. Maximize this as a function of the parameters.

$$L(\vec{\theta}) = \prod_{i=1}^{n} \sigma(\vec{\theta}^T \vec{x}^{(i)})^{y^{(i)}} [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})]^{1-y^{(i)}} P(X^{(i)})$$

$$LL(\vec{\theta}) = \sum_{i=1}^{n} \left[y^{(i)} \log \sigma(\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) \log [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] + \log P(X^{(i)}) \right]$$

$$\frac{\partial}{\partial \theta_j} LL(\vec{\theta}) = \sum_{i=1}^{n} \frac{\partial}{\partial \theta_j} \left[y^{(i)} \log \sigma(\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) \log [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] + \cancel{\log P(X^{(i)})} \right]$$

$$= \sum_{i=1}^{n} \left[y^{(i)} \frac{\partial}{\partial \theta_j} \log \sigma(\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) \frac{\partial}{\partial \theta_j} \log [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] \right]$$

$$= \sum_{i=1}^{n} \left[y^{(i)} \frac{1}{\sigma(\vec{\theta}^T \vec{x}^{(i)})} \frac{\partial}{\partial \theta_j} \sigma(\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) \frac{1}{1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})} \frac{\partial}{\partial \theta_j} [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] \right]$$

Subplot: Derivative of logistic

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

$$= \frac{-1}{\left(1+e^{-z}\right)^2}\frac{\partial}{\partial z}\left(1+e^{-z}\right)$$

$$= \frac{-1}{\left(1+e^{-z}\right)^2}\left(-e^{-z}\right)$$

$$= \frac{1}{1+e^{-z}}\frac{e^{-z}}{1+e^{-z}}$$

$$= \frac{1}{1+e^{-z}}\frac{\left(1+e^{-z}\right)-1}{1+e^{-z}}$$

$$= \frac{1}{1+e^{-z}}\left(1-\frac{1}{1+e^{-z}}\right) = \sigma(z)\left[1-\sigma(z)\right]$$

MLE for logistic regression

3. Maximize this as a function of the parameters.

$$L(\vec{\theta}) = \prod_{i=1}^{n} \sigma(\vec{\theta}^T \vec{x}^{(i)})^{y^{(i)}} [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})]^{1-y^{(i)}} P(X^{(i)})$$

$$LL(\vec{\theta}) = \sum_{i=1}^{n} \left[y^{(i)} \log \sigma(\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) \log[1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] + \log P(X^{(i)}) \right]$$

$$\frac{\partial}{\partial \theta_j} LL(\vec{\theta}) = \sum_{i=1}^{n} \Bigg[y^{(i)} \frac{1}{\sigma(\vec{\theta}^T \vec{x}^{(i)})} \frac{\partial}{\partial \theta_j} \sigma(\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) \frac{1}{1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})} \frac{\partial}{\partial \theta_j} [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] \Bigg]$$

MLE for logistic regression

3. Maximize this as a function of the parameters.

$$L(\vec{\theta}) = \prod_{i=1}^{n} \sigma(\vec{\theta}^T \vec{x}^{(i)})^{y^{(i)}} [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})]^{1-y^{(i)}} P(X^{(i)})$$

$$LL(\vec{\theta}) = \sum_{i=1}^{n} \left[y^{(i)} \log \sigma(\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) \log [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] + \log P(X^{(i)}) \right]$$

$$\frac{\partial}{\partial \theta_j} LL(\vec{\theta}) = \sum_{i=1}^{n} \left[y^{(i)} \frac{1}{\sigma(\vec{\theta}^T \vec{x}^{(i)})} \frac{\partial}{\partial \theta_j} \sigma(\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) \frac{1}{1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})} \frac{\partial}{\partial \theta_j} [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] \right]$$

$$= \sum_{i=1}^{n} \left[y^{(i)} \frac{1}{\cancel{\sigma(\vec{\theta}^T \vec{x}^{(i)})}} \cancel{\sigma(\vec{\theta}^T \vec{x}^{(i)})} [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] \frac{\partial}{\partial \theta_j} (\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) \frac{1}{\cancel{1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})}} [-\sigma(\vec{\theta}^T \vec{x}^{(i)})] \cancel{[1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})]} \frac{\partial}{\partial \theta_j} (\vec{\theta}^T \vec{x}^{(i)}) \right]$$

$$= \sum_{i=1}^{n} \left[y^{(i)} [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] \frac{\partial}{\partial \theta_j} (\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) [-\sigma(\vec{\theta}^T \vec{x}^{(i)})] \frac{\partial}{\partial \theta_j} (\vec{\theta}^T \vec{x}^{(i)}) \right]$$

Subplot 2: Derivative of dot product

$$\frac{\partial}{\partial \theta_j}\left(\vec{\theta}^T \vec{x}^{(i)}\right) = \frac{\partial}{\partial \theta_j}\left(\theta_0 \cdot 1 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_m x_m\right)$$

$$= x_j$$

MLE for logistic regression

3. Maximize this as a function of the parameters.

$$L(\vec{\theta}) = \prod_{i=1}^{n} \sigma(\vec{\theta}^T \vec{x}^{(i)})^{y^{(i)}} [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})]^{1-y^{(i)}} P(X^{(i)})$$

$$LL(\vec{\theta}) = \sum_{i=1}^{n} \left[y^{(i)} \log \sigma(\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) \log [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] + \log P(X^{(i)}) \right]$$

$$\frac{\partial}{\partial \theta_j} LL(\vec{\theta}) =$$

$$\sum_{i=1}^{n} \left[y^{(i)} [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] \frac{\partial}{\partial \theta_j} (\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) [-\sigma(\vec{\theta}^T \vec{x}^{(i)})] \frac{\partial}{\partial \theta_j} (\vec{\theta}^T \vec{x}^{(i)}) \right]$$

$$= \sum_{i=1}^{n} \left[y^{(i)} [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] \vec{x}_j^{(i)} + (1 - y^{(i)}) [-\sigma(\vec{\theta}^T \vec{x}^{(i)})] \vec{x}_j^{(i)} \right]$$

$$= \sum_{i=1}^{n} \left[y^{(i)} \cancel{- y^{(i)} \sigma(\vec{\theta}^T \vec{x}^{(i)})} - \sigma(\vec{\theta}^T \vec{x}^{(i)}) \cancel{+ y^{(i)} \sigma(\vec{\theta}^T \vec{x}^{(i)})} \right] \vec{x}_j^{(i)}$$

$$= \sum_{i=1}^{n} \left[y^{(i)} - \sigma(\vec{\theta}^T \vec{x}^{(i)}) \right] \vec{x}_j^{(i)} = 0 \qquad ???$$

Derivatives the easier way

$$\frac{\partial}{\partial \theta_j} LL(\vec{\theta}) = \sum_{i=1}^{n} \frac{\partial}{\partial \theta_j} \left[y^{(i)} \log \sigma(\vec{\theta}^T \vec{x}^{(i)}) + (1 - y^{(i)}) \log [1 - \sigma(\vec{\theta}^T \vec{x}^{(i)})] \right]$$

$p = \sigma(z)$

$z = \vec{\theta}^T \vec{x}^{(i)}$

$$= \sum_{i=1}^{n} \frac{\partial}{\partial \theta_j} \left[y^{(i)} \log p + (1 - y^{(i)}) \log (1 - p) \right]$$

$$= \sum_{i=1}^{n} \left[y^{(i)} \frac{\partial}{\partial \theta_j} \log p + (1 - y^{(i)}) \frac{\partial}{\partial \theta_j} \log (1 - p) \right]$$

$$= \sum_{i=1}^{n} \left[y^{(i)} \frac{1}{p} \frac{\partial p}{\partial \theta_j} + (1 - y^{(i)}) \frac{-1}{1 - p} \frac{\partial p}{\partial \theta_j} \right]$$

$$= \sum_{i=1}^{n} \left[y^{(i)} \frac{1}{p} - (1 - y^{(i)}) \frac{1}{1 - p} \right] \sigma(z) [1 - \sigma(z)] \frac{\partial z}{\partial \theta_j}$$

$$= \sum_{i=1}^{n} \left[y^{(i)} \frac{1}{p} - (1 - y^{(i)}) \frac{1}{1 - p} \right] p (1 - p) x_j^{(i)}$$

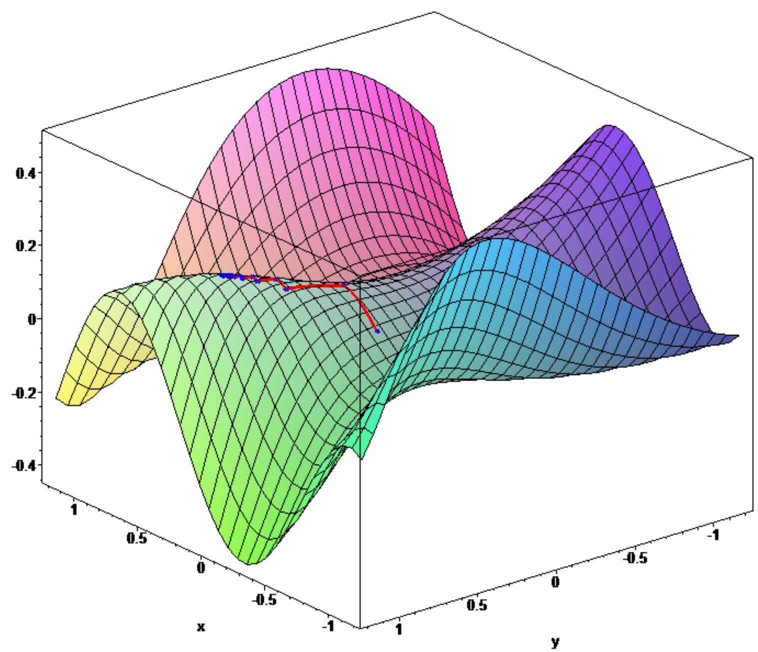
$$= \sum_{i=1}^{n} \left[y^{(i)} (1 - p) - (1 - y^{(i)}) p \right] x_j^{(i)}$$

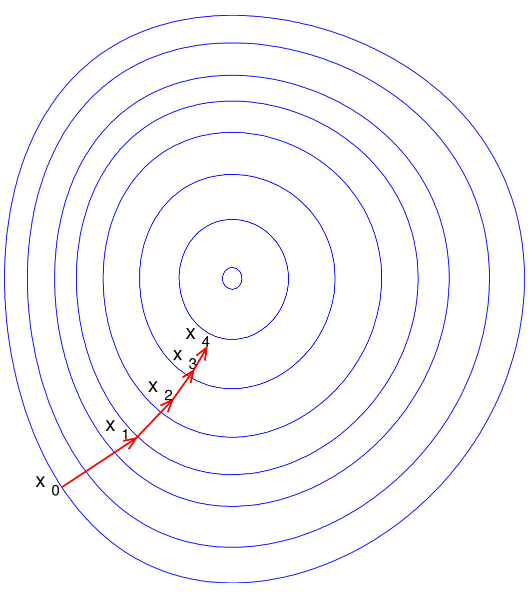
$$= \sum_{i=1}^{n} \left[y^{(i)} - y^{(i)} p - p + y^{(i)} p \right] x_j^{(i)} = \sum_{i=1}^{n} \left[y^{(i)} - p \right] x_j^{(i)}$$

Gradient ascent

An algorithm for computing an **arg max** by taking small steps **uphill** (i.e., in the direction of the **gradient** of the function).

$$\vec{\theta} \leftarrow \vec{\theta} + \eta \cdot \nabla_{\vec{\theta}} f(\vec{\theta})$$

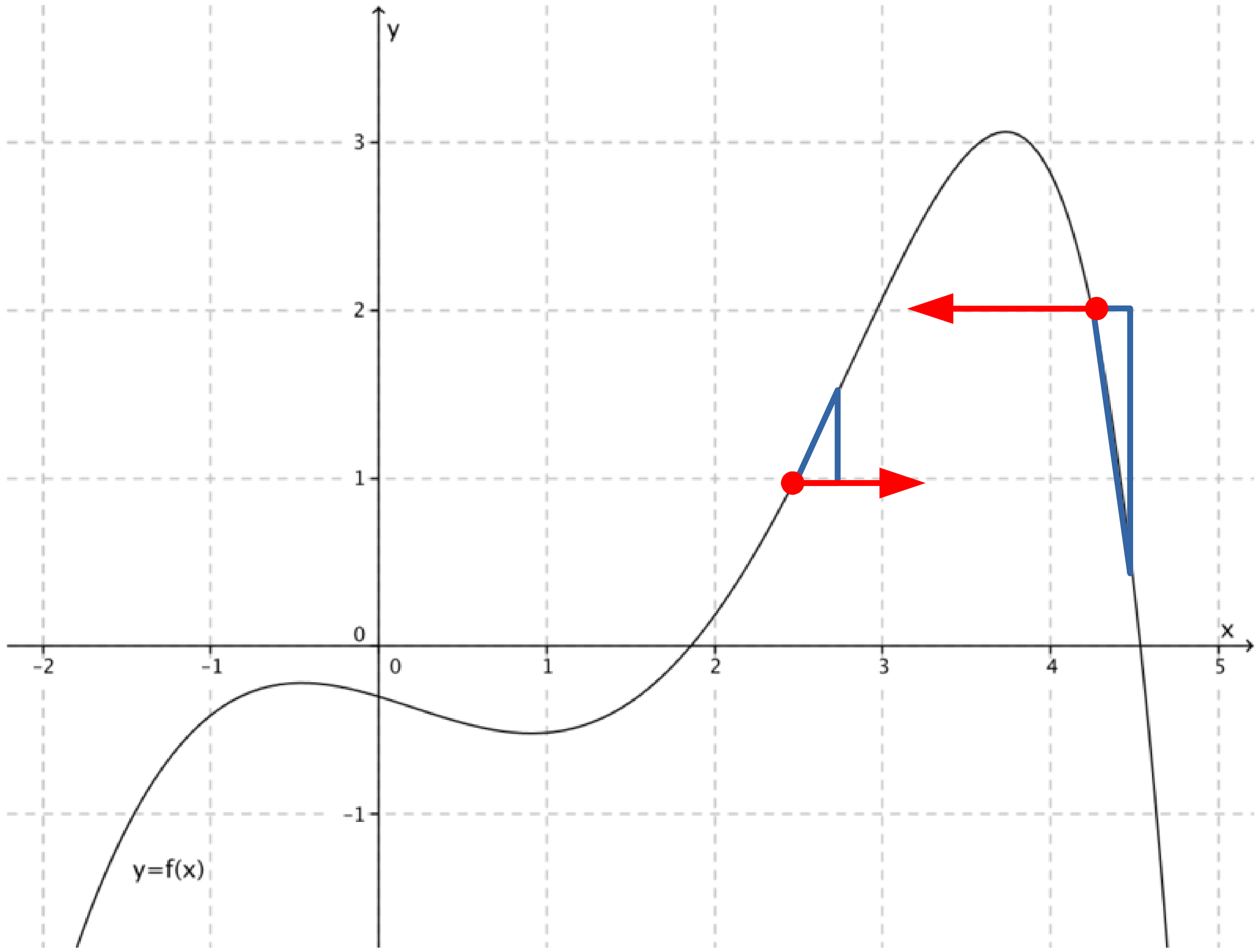




Gradient (a review)

$$\nabla_{\vec{\theta}} f(\vec{\theta}) = \begin{bmatrix} \frac{\partial f}{\partial \theta_1} \\ \frac{\partial f}{\partial \theta_2} \\ \vdots \\ \frac{\partial f}{\partial \theta_m} \end{bmatrix}$$

A derivative points uphill



A gradient points uphill

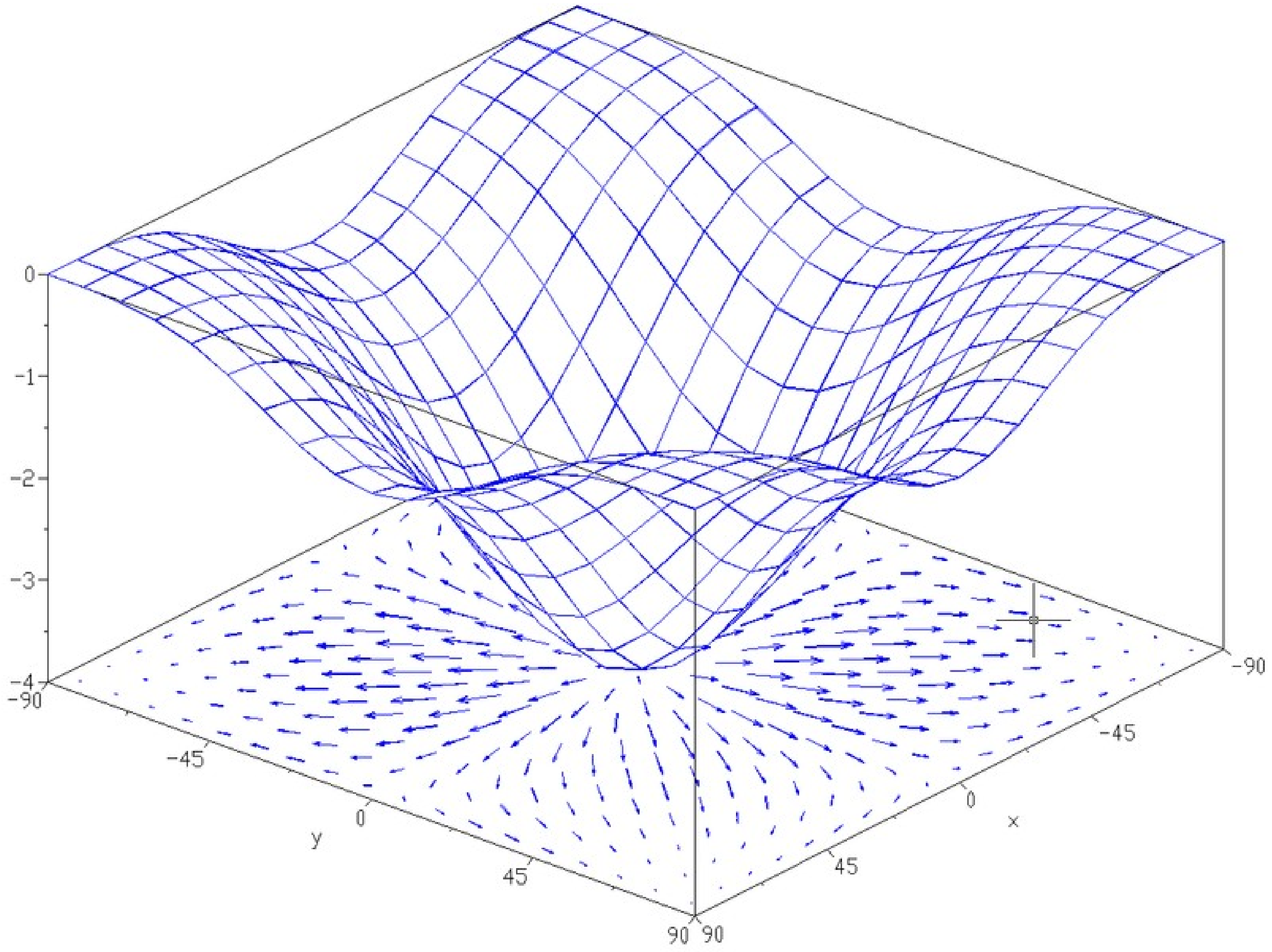


image: Simiprof

Logistic regression: Pseudocode

```
initialize: θ = [0, 0, …, 0]   (m elements)

repeat many times:

    move θ a small amount "uphill"

return θ
```

gives you a local maximum

Logistic regression: Pseudocode

```
initialize: θ = [0, 0, …, 0]   (m elements)

repeat many times:
```

compute gradient[j] = $\frac{\partial}{\partial \theta_j} LL(\vec{\theta})$

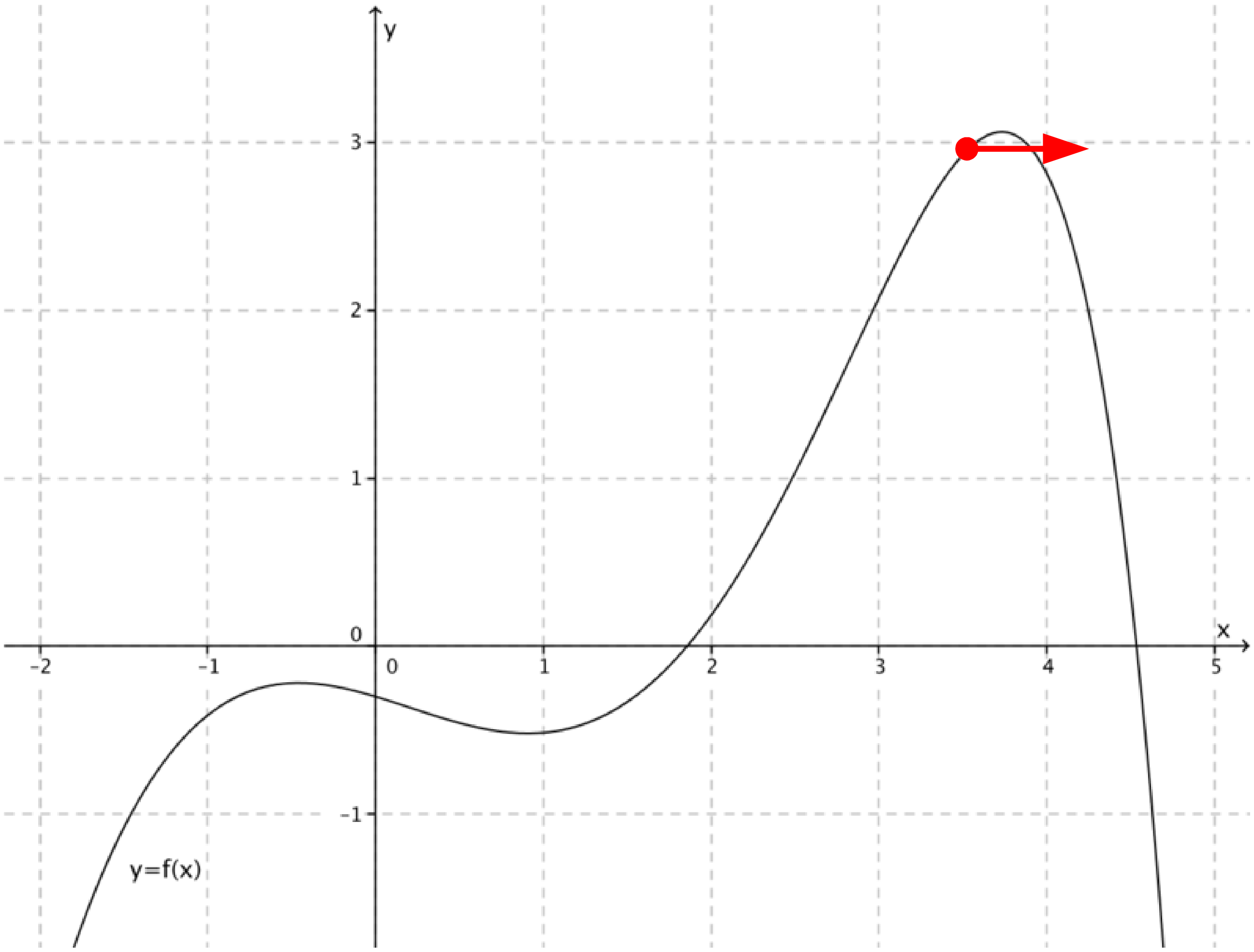
```
    for j = 0 to m:
        θ[j] += η * gradient[j]
```

η = "learning rate"

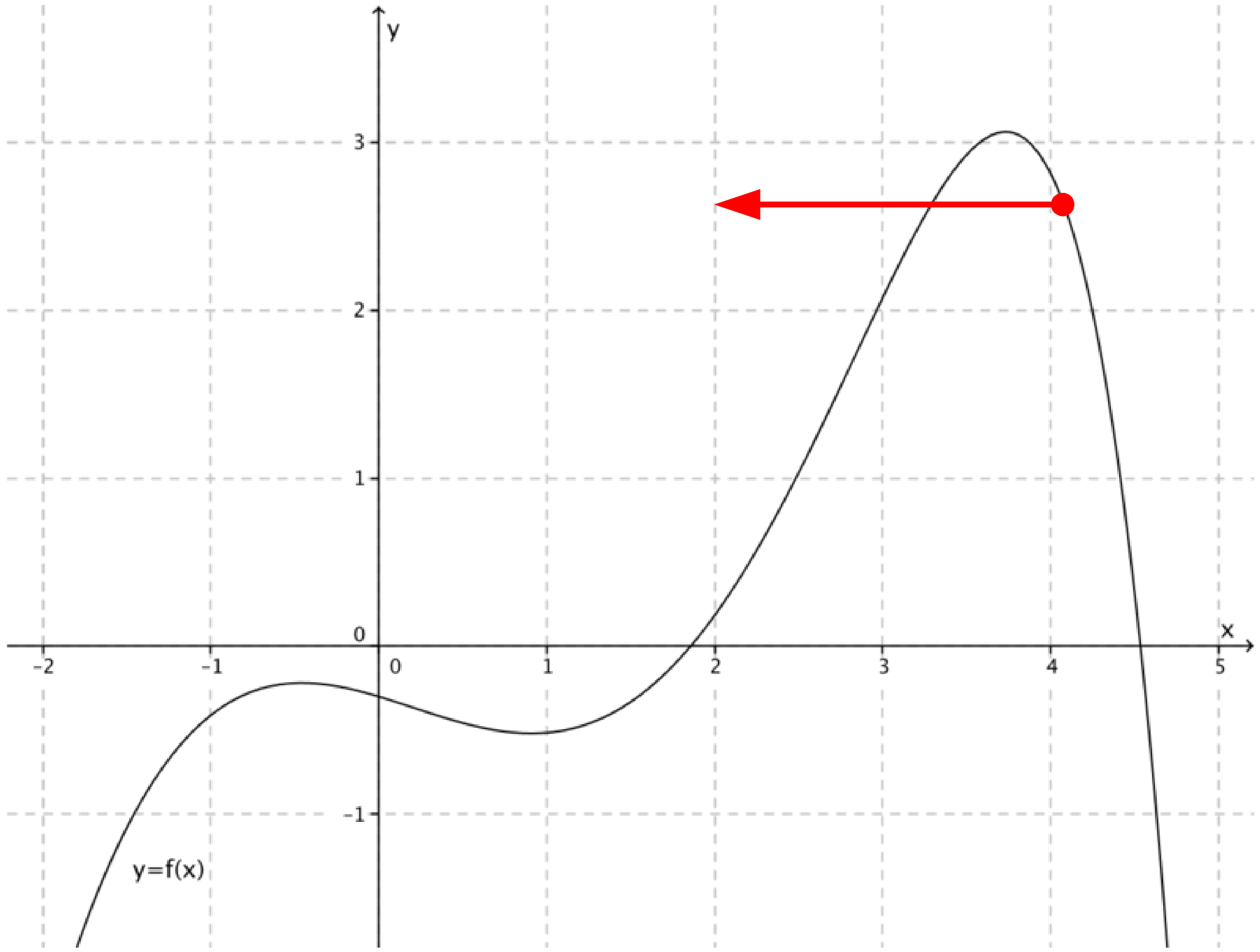
```
return θ
```

gives you a local maximum
(if η is small enough!)

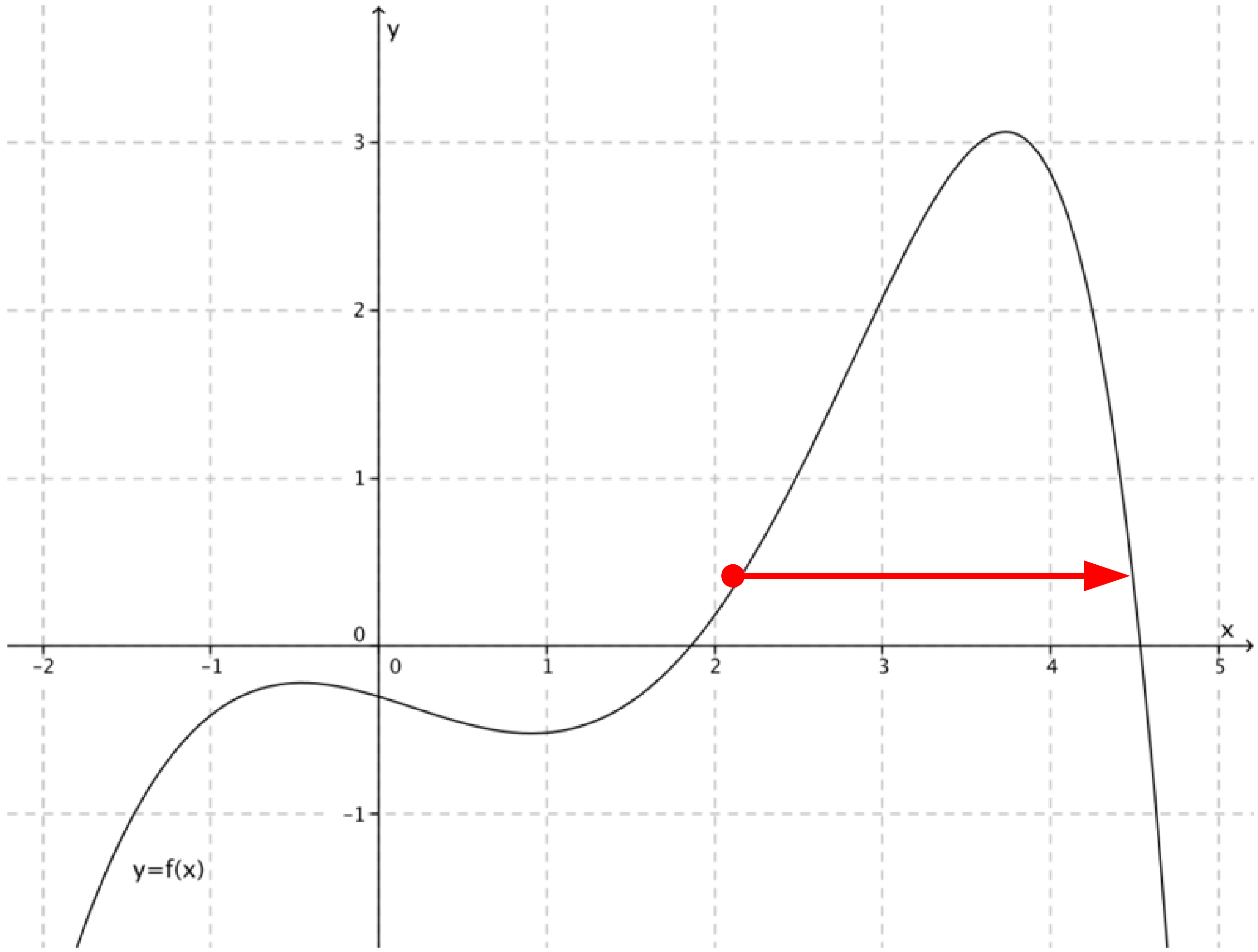
The curse of the large step size



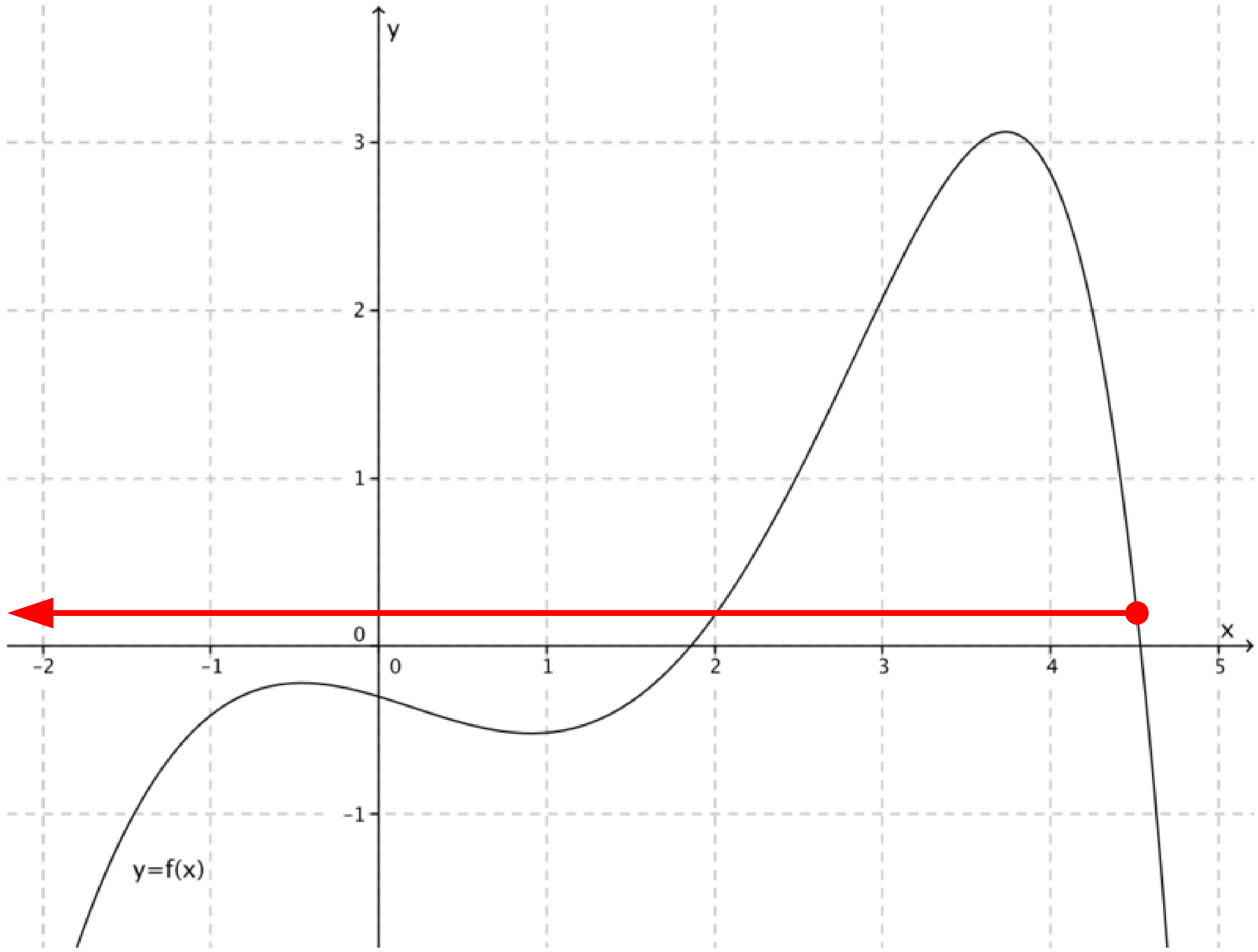
The curse of the large step size



The curse of the large step size



The curse of the large step size



Logistic regression: Pseudocode

```
initialize: θ = [0, 0, …, 0]   (m elements)

repeat many times:
    gradient = [0, 0, …, 0]   (m elements)

    for each training example (x^(i), y^(i)):
        add ∂/∂θ_j LL_{x^(i)}(θ⃗) to each gradient[j]

    for j = 0 to m:
        θ[j] += η * gradient[j]

return θ
```

$$\frac{\partial}{\partial \theta_j} LL_{x^{(i)}}(\vec{\theta})$$

Logistic regression: Pseudocode

```
initialize: θ = [0, 0, …, 0]   (m elements)

repeat many times:
    gradient = [0, 0, …, 0]   (m elements)

    for each training example (x^(i), y^(i)):
        for j = 0 to m:
```

$$\text{gradient[j] += } [y^{(i)} - \sigma(\vec{\theta}^T \vec{x}^{(i)})] x_j^{(i)}$$

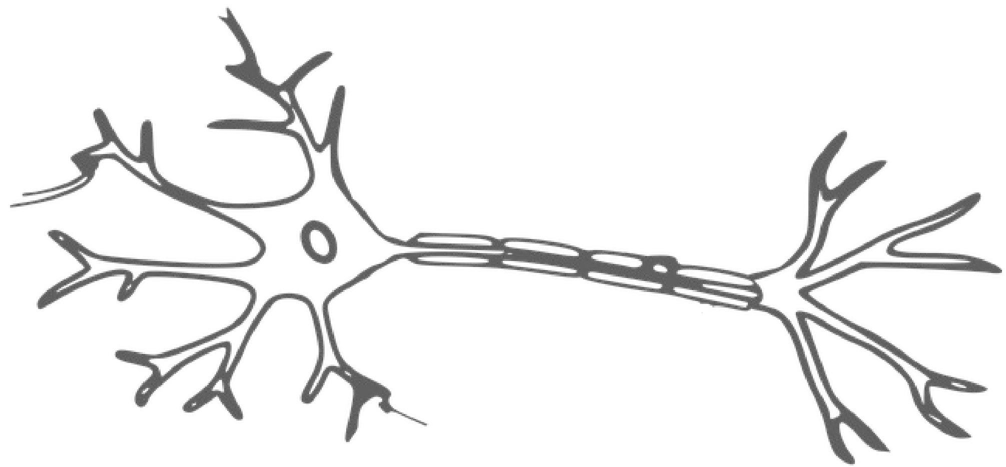
```
    for j = 0 to m:
        θ[j] += η * gradient[j]

return θ
```

Your brain on logistic regression

$p = \sigma(z)$

$z = \vec{\theta}^T \vec{x}^{(i)}$



dendrites:
take a weighted sum
of incoming stimuli
with electric potential

axon:
carries outgoing
pulse if potential
exceeds a threshold

Caution: Just a (greatly simplified) model! All models are wrong—but some are useful…