CS 360: Machine Learning

Sara Mathieson, Sorelle Friedler

Spring 2024

HAVERFORD COLLEGE

Admin

- Checkpoint in lab today: finished **Part 1**
- Office hour change: **Wednesday 3-4pm in Zubrow** (Sara)
  - No office hours on Thursday! (faculty meeting)
- **Lab 4** due Thursday Feb 22

Outline for Feb 20

- Stochastic gradient descent
- Maximizing likelihood functions
- Logistic regression likelihood
- Extending logistic regression to multi-class classification (softmax)
- Regularization

Outline for Feb 20

- Stochastic gradient descent
- Maximizing likelihood functions
- Logistic regression likelihood
- Extending logistic regression to multi-class classification (softmax)
- Regularization

High-level idea of gradient descent

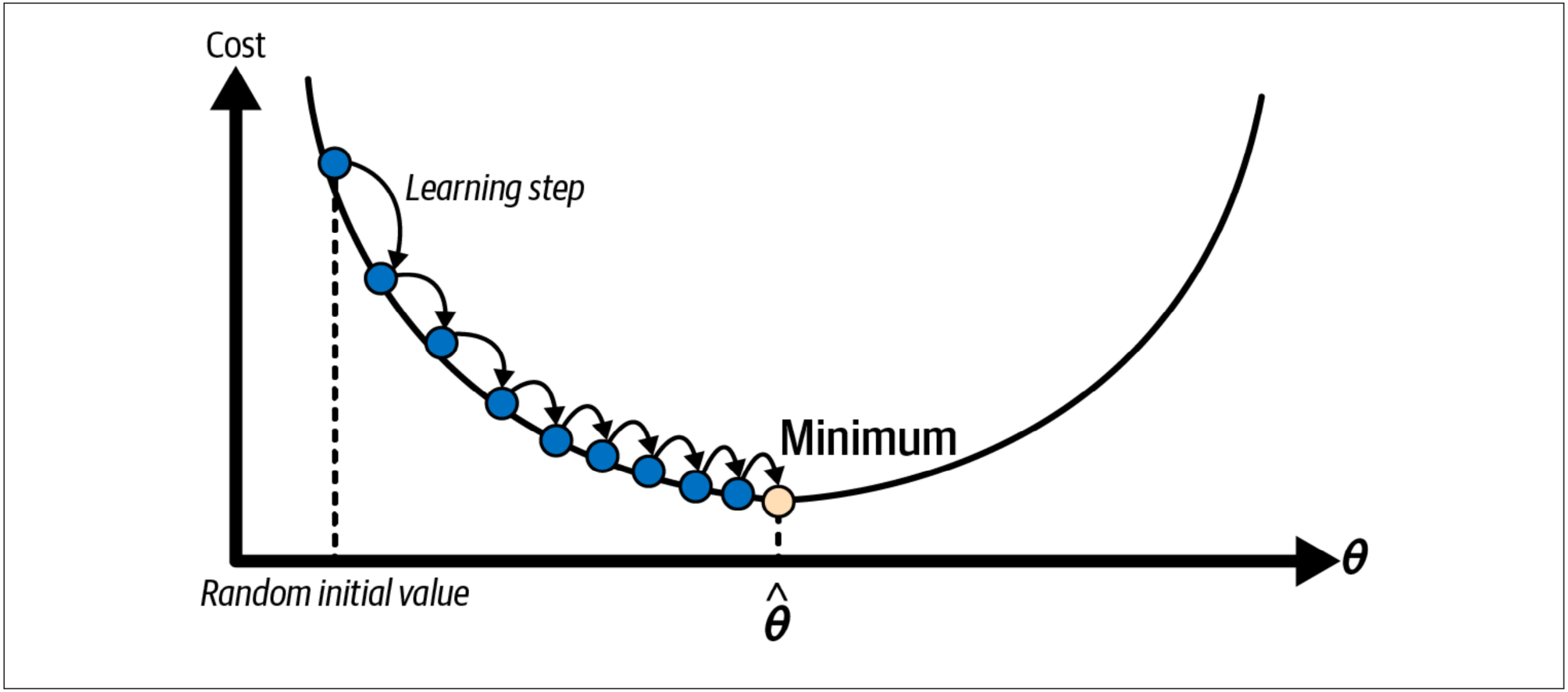
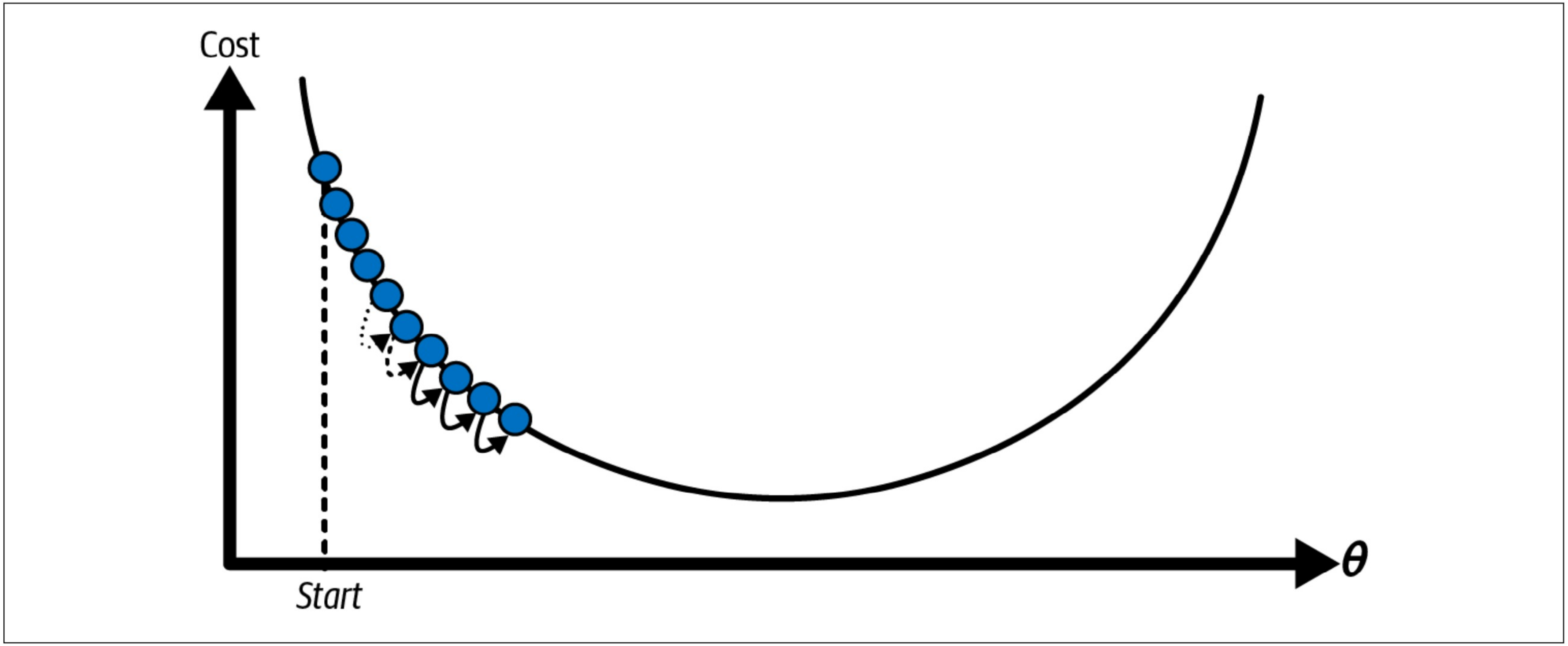
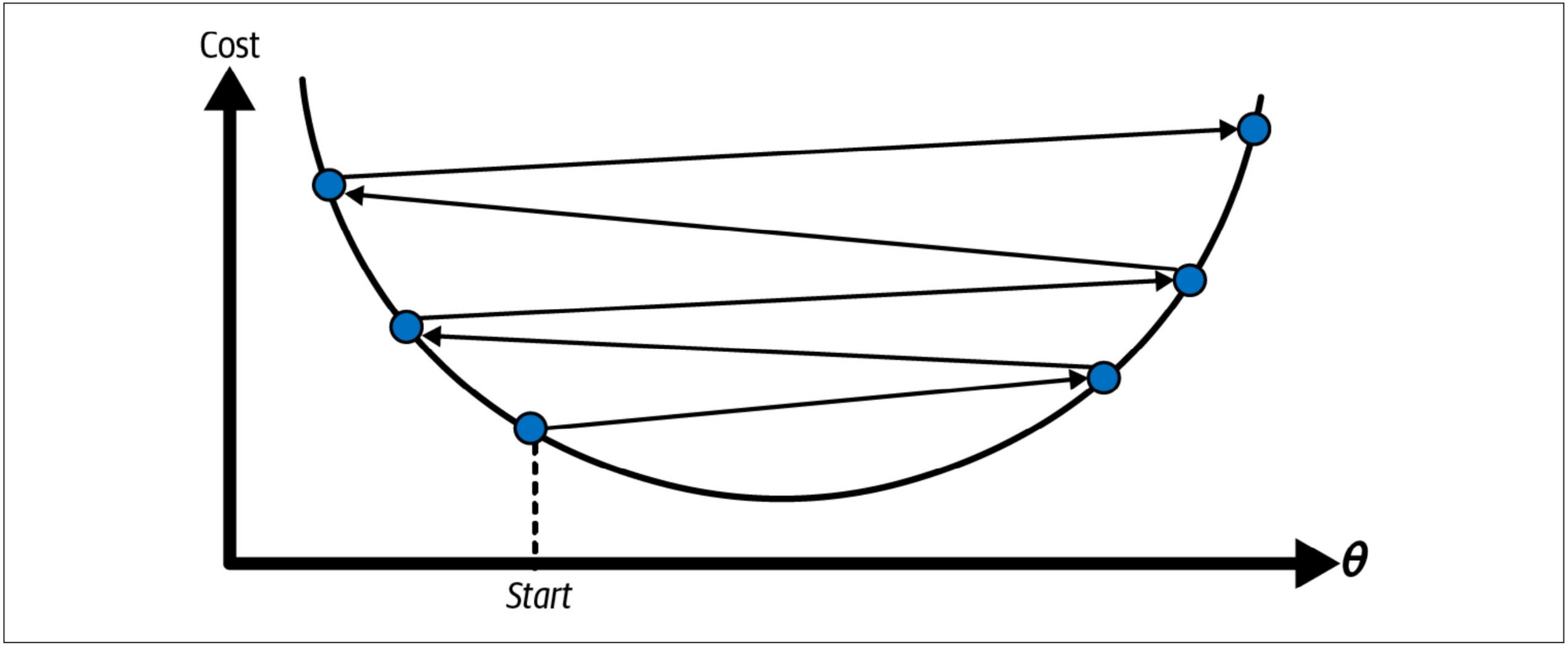


Figure 4-3. In this depiction of gradient descent, the model parameters are initialized randomly and get tweaked repeatedly to minimize the cost function; the learning step size is proportional to the slope of the cost function, so the steps gradually get smaller as the cost approaches the minimum

Gradient descent: learning rate too small



Gradient descent: learning rate too high



Gradient descent: no global optima

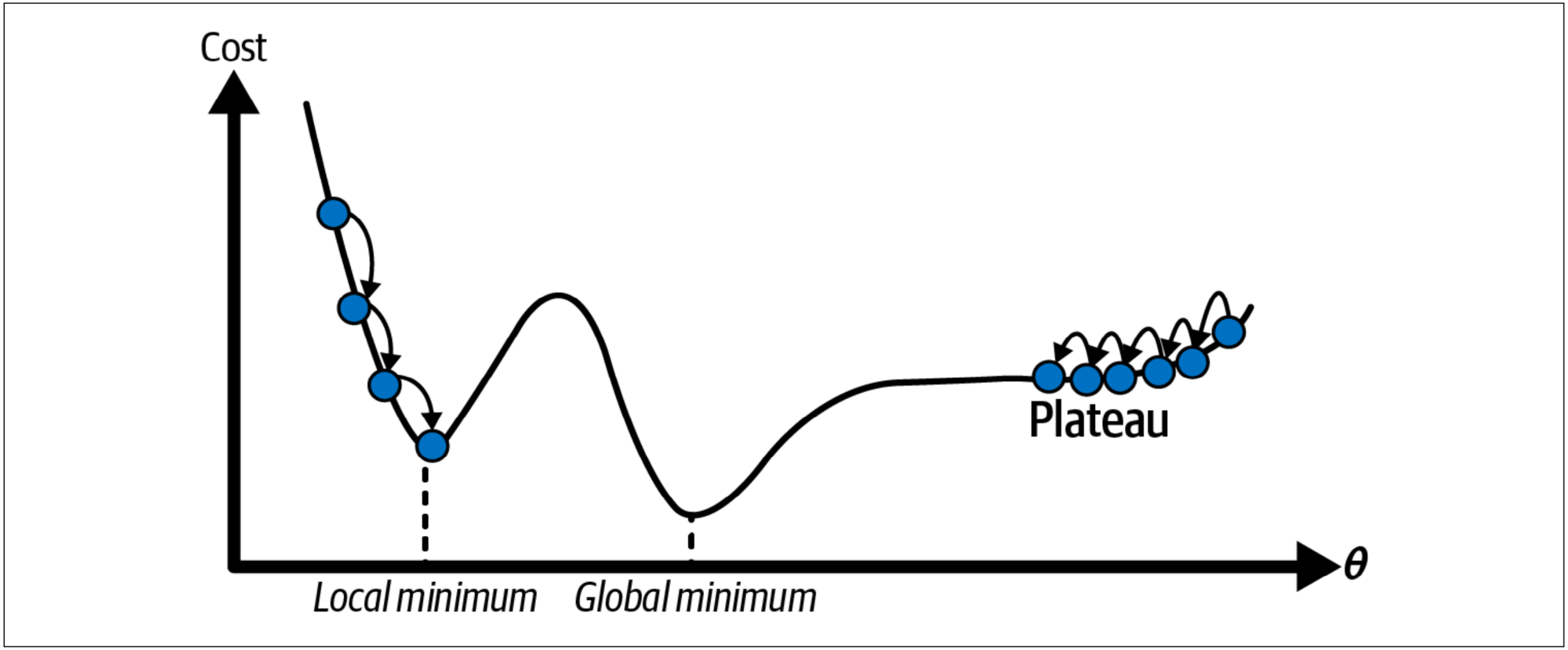


Figure 4-6. Gradient descent pitfalls

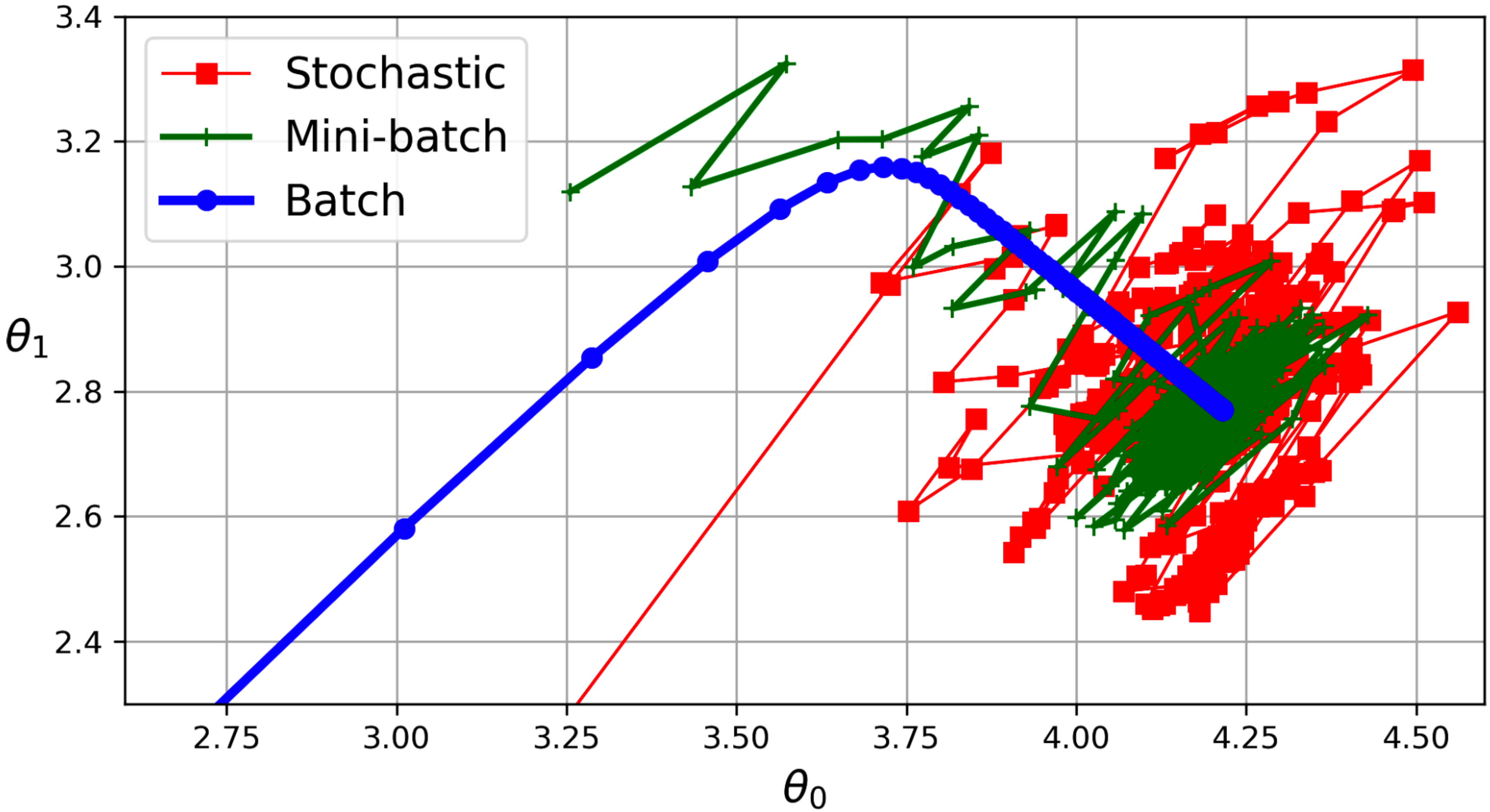
Stochastic Gradient Descent (high-level)

```
set w = 0 vector
while cost J(w) still changing (or max iter reached):
    shuffle data points
    for i = 1…n:
        w <- w – alpha(derivative of J(w) wrt x_i)
    store J(w)
```

Gradient descent variations

- **Batch gradient descent**
  - Go over *all* training data before making a weight update
- **Stochastic gradient descent**
  - Shuffle data and make a weight update after *each training example*
- **Mini-batch gradient descent**
  - Update after a "*mini-batch*" of training examples (i.e. 50)
  - Vocab word: **epoch** (one pass through the data)

Comparison of gradient descent paths



Geron Chap 4

Outline for Feb 20

- Stochastic gradient descent
- Maximizing likelihood functions
- Logistic regression likelihood
- Extending logistic regression to multi-class classification (softmax)
- Regularization

Logistic Regression

$$h_{\vec{w}}(\vec{x}) = p(y=1 \mid \vec{x})$$

$$y \in \{0,1\} \qquad = \frac{1}{1+e^{-\vec{w}\cdot\vec{x}}}$$

$$\boxed{\vec{w}\cdot\vec{x} = w_0 + w_1 x_1 + w_2 x_2 \dots w_p x_p}$$

"fake 1"

Big Q: what is $\vec{w}$?

Likelihood function

n training examples

$$L(\vec{w}) = \underbrace{\underbrace{\prod_{i=1}^{n}}_{\text{product}} \underbrace{h_{\vec{w}}(\vec{x}_i)^{y_i}}_{\text{prob of label 1}} \underbrace{\left(1-h_{\vec{w}}(\vec{x}_i)\right)^{1-y_i}}_{\text{prob of label 0}}}_{\text{maximize likelihood}}$$

A

Aside to maximum likelihood estimators (MLE)

<u>coin flip</u>

prob(1) = p

prob(0) = 1 − p

\# flips = n

$\vec{y} = [0, 0, 1, 1, 0, 1, 0, 1, 0, 0]$

$\boxed{n = 10}$

$L(p) = (1-p)(1-p)\,p\,p\,(1-p)\,p \cdots (1-p)$

$= p^4 (1-p)^6$

$L(p) = \prod_{i=1}^{n} p^{y_i} (1-p)^{1-y_i} = p^{\#1\text{s}} (1-p)^{\#0\text{s}}$

$\#1\text{'s} = \underset{\text{mean/avg}}{\bar{y}}\, n = \frac{4}{\cancel{10}} \cdot \cancel{10}$

$\#0\text{'s} = (1-\bar{y})\, n$

$\boxed{\text{Handout 9}}$

- $\log(b^a) = a \log b$
- $f(x) = \log x$

  $f'(x) = \frac{1}{x}$

$$\ell(p) = \bar{y}\, n \log p + n(1-\bar{y})\log(1-p)$$

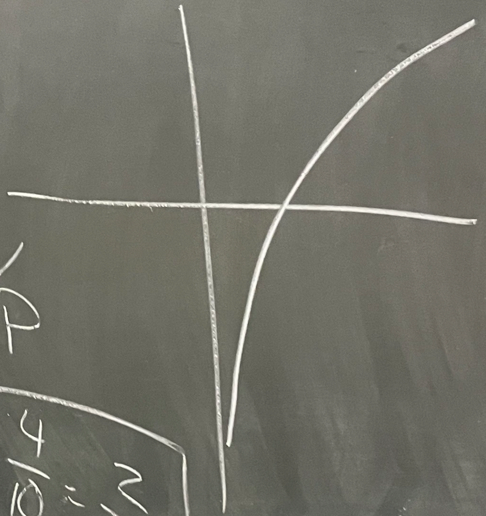
$$\ell'(p) = \frac{\cancel{n}\,\bar{y}}{p} - \frac{\cancel{n}(1-\bar{y})}{1-p} = 0$$

$$\frac{\bar{y}}{p} = \frac{1-\bar{y}}{1-p}$$

$$\bar{y} - \cancel{\bar{y}p} = p - \cancel{\bar{y}p}$$

$$\boxed{\hat{p} = \bar{y}}$$

$$\boxed{\hat{p} = \frac{4}{10} = \frac{2}{5}}$$



b

ning
amples

$1-y_i$

$(\vec{x}_i))$

of

el 0

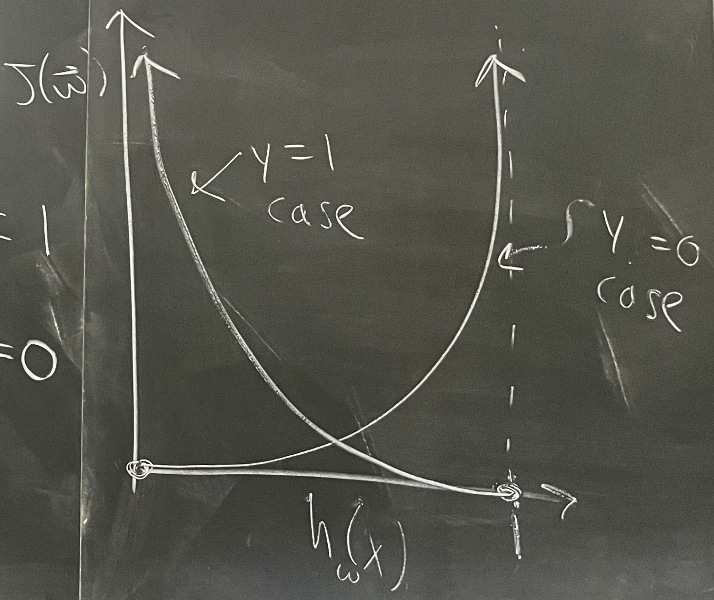
negative log likelihood (minimize)

$$J(\vec{w}) = -\log L(\vec{w})$$

$$= -\sum_{i=1}^{n} y_i \log h_{\vec{w}}(x_i) + (1-y_i)\log(1-h_{\vec{w}}(x_i))$$

Single example

$$J_x(\vec{w}) = \begin{cases} -\log h_{\vec{w}}(\vec{x}), & y=1 \\ -\log(1-h(\vec{x})), & y=0 \end{cases}$$



Stochastic Gradient Descent

Shuffle

for $i = 1 \ldots n$

$$\vec{w} \leftarrow \vec{w} - \alpha \boxed{\nabla_{\vec{w}} J_{\vec{x}_i}(\vec{w})} \star$$

$$\nabla J_{\vec{x}_i}(\vec{w}) = -\left(\frac{y_i}{h(\vec{x}_i)} - \frac{(1-y_i)}{1-h(\vec{x}_i)} \right) \nabla h_{\vec{w}}(\vec{x}_i)$$

$$\rightarrow -\left(\frac{y_i}{h(\vec{x}_i)} - \frac{1-y_i}{1-h(x_i)} \right) h(x_i)(1-h(x_i))\,\vec{x}_i$$

$$= \left(-y_i + \cancel{y_i h(x_i)} + h(\vec{x}_i) - \cancel{y_i h(\vec{x}_i)} \right) \vec{x}_i$$

$$= \boxed{\Big(\underset{\text{pred}}{h_{\vec{w}}(\vec{x}_i)} - \underset{\text{truth}}{y_i} \Big) \vec{x}_i} \leftarrow \nabla J_{\vec{x}_i}(\vec{w})$$

$$g(z) = \frac{1}{1 + e^{-z}}, \quad g'(z) = g(z)(1 - g(z))$$

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

z: $\boxed{\vec{w} \cdot \vec{x}_i}$

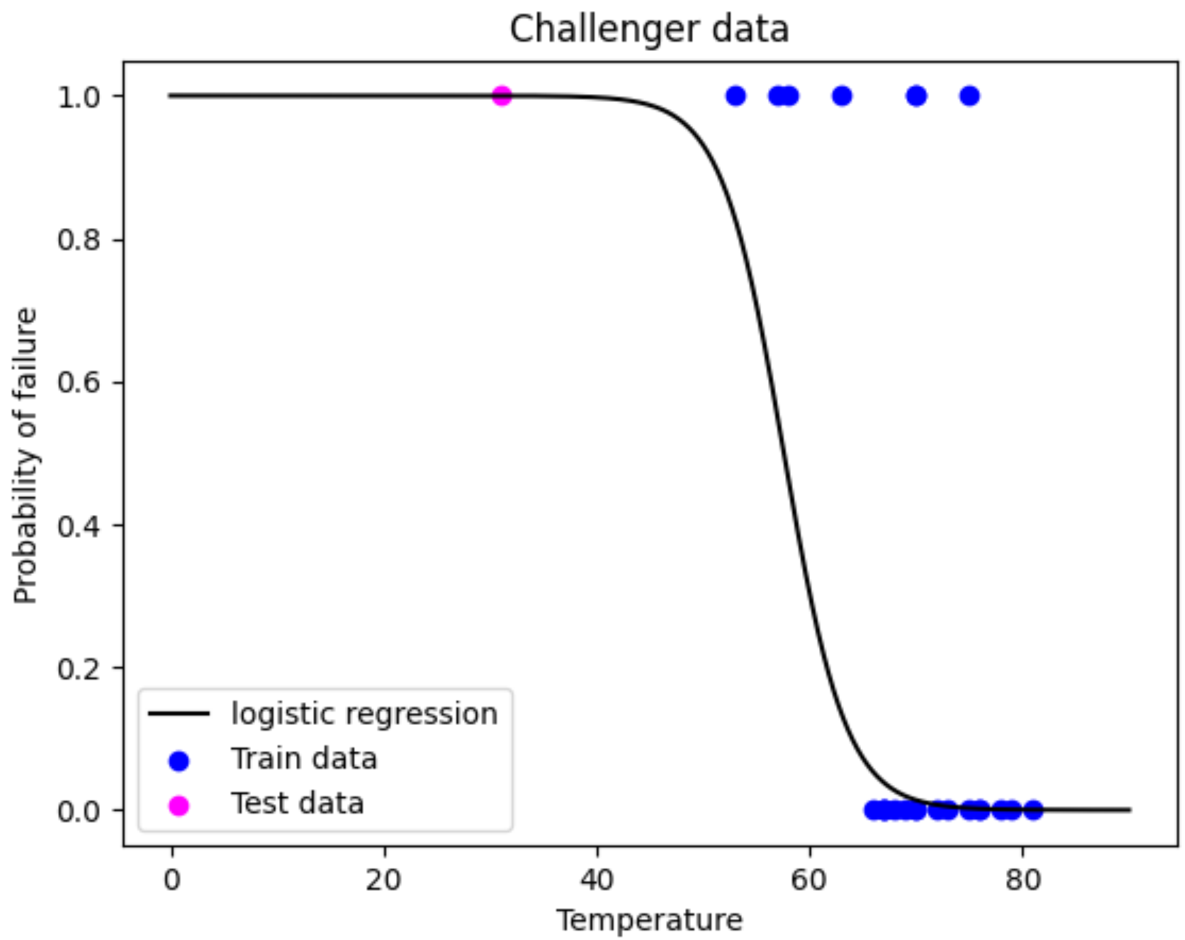
exercise!

Challenger Explosion Data

Image: NASA

1	Date	Temperature	Damage Incident
2	04/12/1981	66	0
3	11/12/1981	70	1
4	3/22/82	69	0
5	6/27/82	80	NA
6	01/11/1982	68	0
7	04/04/1983	67	0
8	6/18/83	72	0
9	8/30/83	73	0
10	11/28/83	70	0
11	02/03/1984	57	1
⋮			
23	10/30/85	75	1
24	11/26/85	76	0
25	01/12/1986	58	1
26	1/28/86	31	Challenger Accident

Challenger Explosion Data



Logistic Regression model fit to the Challenger dataset: modified by work from Maha Attique

Logistic Function

$$\sigma(t) = \frac{1}{1 + \exp(-t)}$$

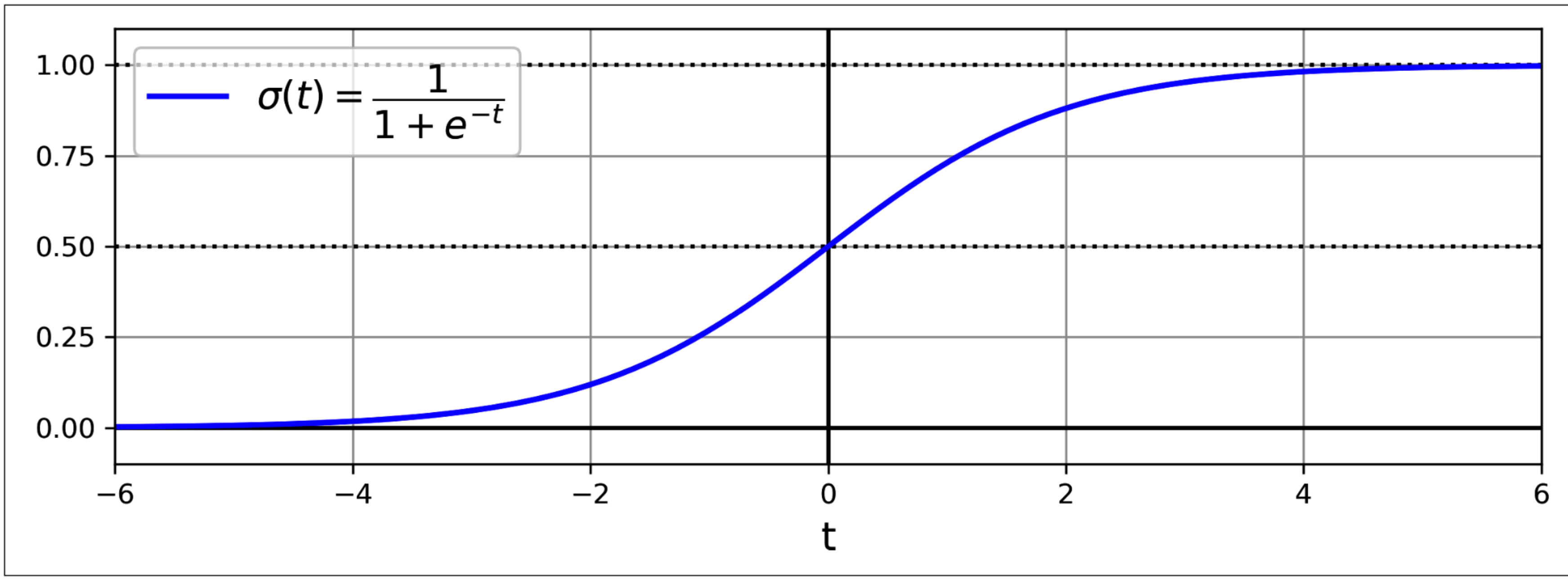


Figure 4-21. Logistic function

Geron Fig 4-21

Logistic Regression creates a **linear** decision boundary!

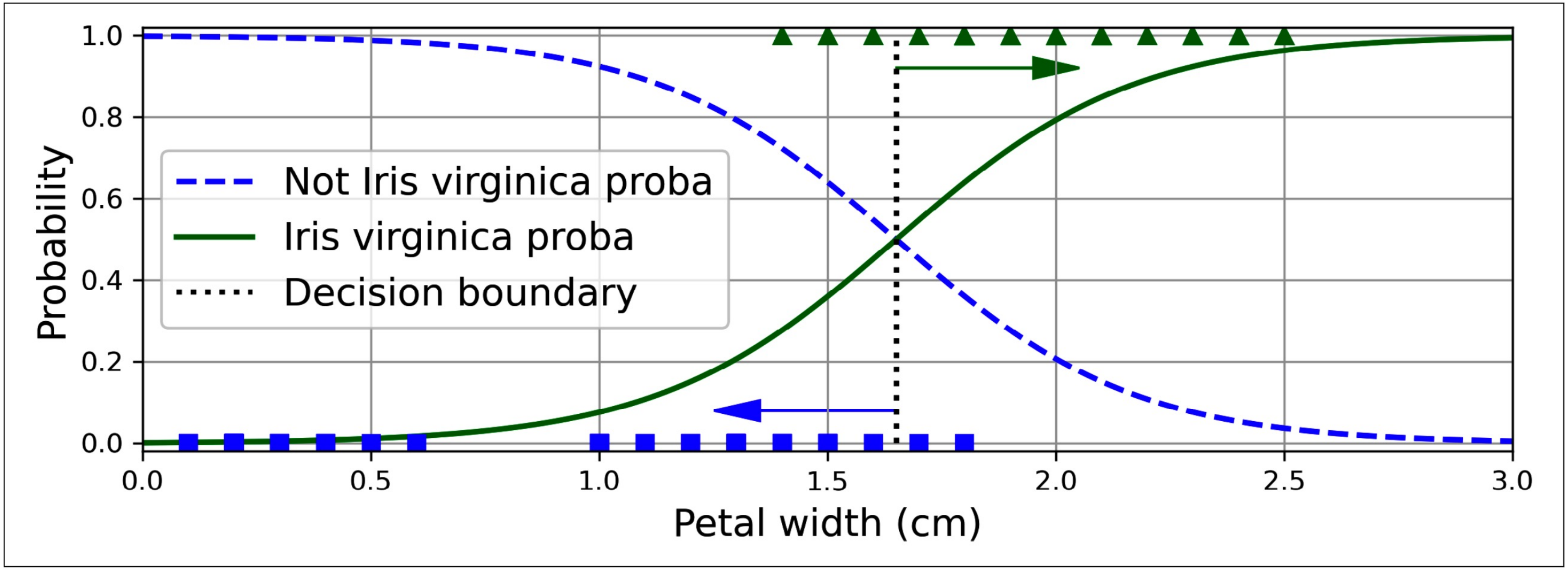


Figure 4-23. Estimated probabilities and decision boundary

Logistic Regression model

- Equivalent to p(y=1 | x) from Naïve Bayes

$$\widehat{p} = h_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma(\boldsymbol{\theta}^\top \mathbf{x})$$

- Idea of threshold is still the same!

$$\widehat{y} = \begin{cases} 0 \ \text{ if } \ \widehat{p} < 0.5 \\ 1 \ \text{ if } \ \widehat{p} \geq 0.5 \end{cases}$$

Logistic Regression Cost

- Cost function for single example

$$c(\boldsymbol{\theta}) = \begin{cases} -\log(\widehat{p}) & \text{if } y = 1 \\ -\log(1 - \widehat{p}) & \text{if } y = 0 \end{cases}$$

- Overall logistic regression cost function

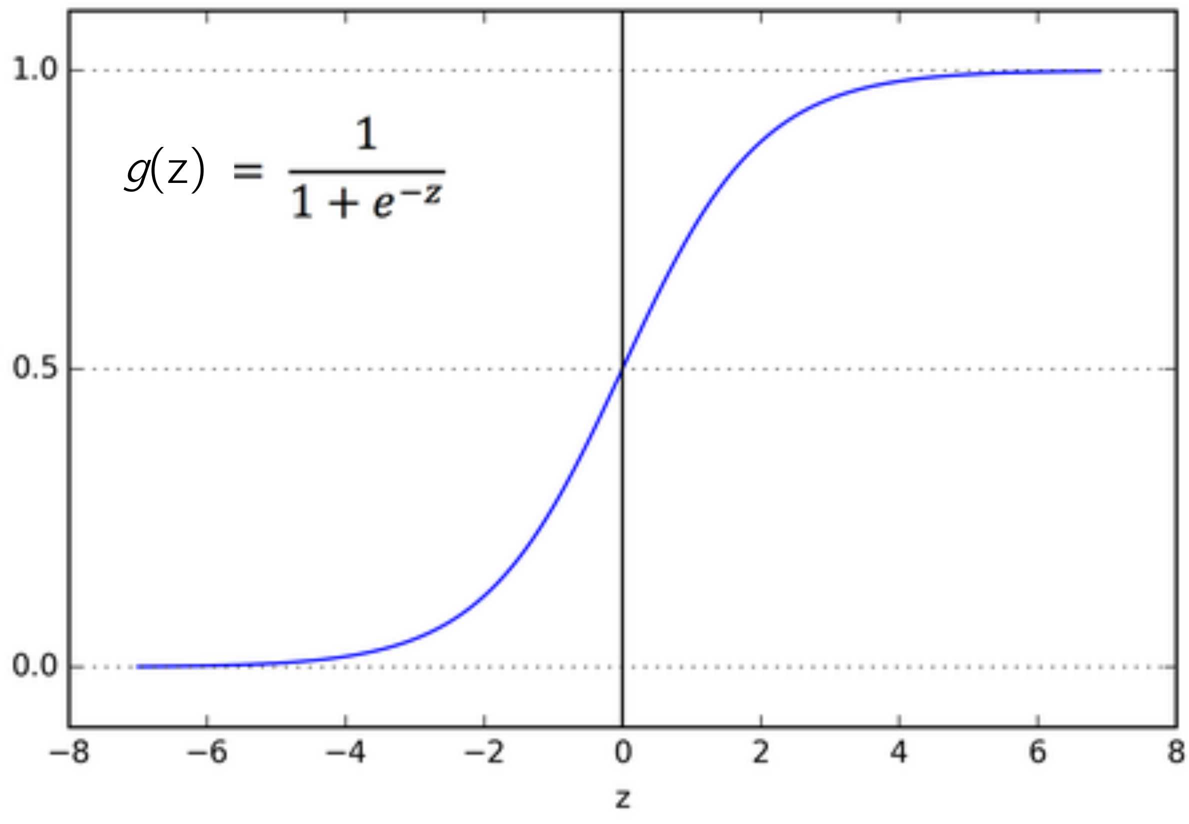
$$J(\boldsymbol{\theta}) = -\frac{1}{m}\Sigma_{i=1}^{m}\left[y^{(i)}log\left(\widehat{p}^{(i)}\right) + \left(1 - y^{(i)}\right)log\left(1 - \widehat{p}^{(i)}\right)\right]$$

3 important pieces to SGD

- Hypothesis function (prediction)

$$\boxed{h_{\boldsymbol{w}}(\boldsymbol{x}) = p(y = 1|\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{w}\cdot\boldsymbol{x}}}}$$

Logistic (sigmoid) function



3 important pieces to SGD

- Hypothesis function (prediction)

$$\boxed{h_{\boldsymbol{w}}(\boldsymbol{x}) = p(y = 1|\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{w}\cdot\boldsymbol{x}}}}$$

- Cost function (want to minimize)

$$\boxed{J(\boldsymbol{w}) = -\sum_{i=1}^{n} y_i \log h_{\boldsymbol{w}}(\boldsymbol{x_i}) + (1 - y_i) \log(1 - h_{\boldsymbol{w}}(\boldsymbol{x_i}))}$$

3 important pieces to SGD

- Hypothesis function (prediction)

$$\boxed{h_{\boldsymbol{w}}(\boldsymbol{x}) = p(y=1|\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{w}\cdot\boldsymbol{x}}}}$$

- Cost function (want to minimize)

$$\boxed{J(\boldsymbol{w}) = -\sum_{i=1}^{n} y_i \log h_{\boldsymbol{w}}(\boldsymbol{x_i}) + (1-y_i)\log(1-h_{\boldsymbol{w}}(\boldsymbol{x_i}))}$$

- Gradient of cost wrt single data point x_i

$$\boxed{\nabla J_{\boldsymbol{x}_i}(\boldsymbol{w}) = (h_{\boldsymbol{w}}(\boldsymbol{x_i}) - y_i)\boldsymbol{x_i}}$$

Outline for Feb 20

- Stochastic gradient descent
- Maximizing likelihood functions
- Logistic regression likelihood
- Extending logistic regression to multi-class classification (softmax)
- Regularization

Multi-class logistic Regression

- political parties
- blood groups

2 classes: $h(\vec{x}) = \dfrac{1}{1+e^{-\vec{w}\cdot\vec{x}}} = \dfrac{e^{\vec{w}\cdot\vec{x}}}{\underbrace{e^{\vec{w}\cdot\vec{x}}}_{\substack{\text{weight}\\ \text{on class}\\ 1}} + \underbrace{1}_{\substack{\text{weight}\\ \text{on class}\\ 0}}}$

K classes

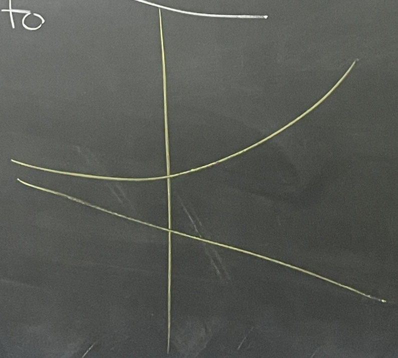
$$\underbrace{\hat{\vec{y}}}_{\text{prediction}} = \begin{bmatrix} p(y=1 \mid \vec{x}) \\ p(y=2 \mid \vec{x}) \\ \vdots \\ p(y=K \mid \vec{x}) \end{bmatrix}$$

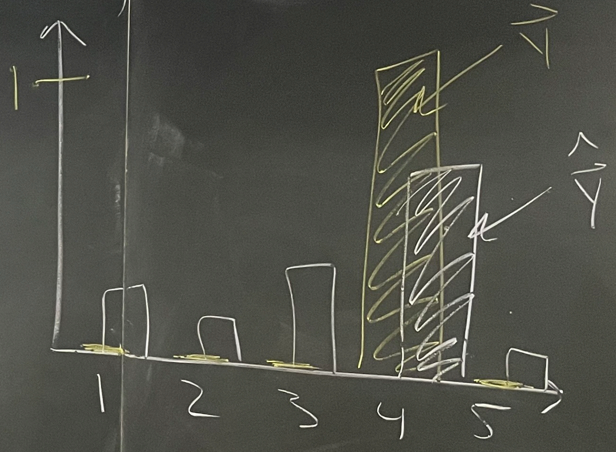
$$\underbrace{\vec{y}}_{\text{true}} = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ \\ \\ \end{array} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow \text{true class}$$

$$\begin{bmatrix} p(y=1|\vec{x}) \\ p(y=2|\vec{x}) \\ \vdots \\ p(y=K|\vec{x}) \end{bmatrix} = \frac{1}{\sum_{k=1}^{K} e^{\vec{w}^{(k)}\cdot\vec{x}}} \begin{array}{c} \begin{bmatrix} e^{\vec{w}^{(1)}\cdot\vec{x}} \\ e^{\vec{w}^{(2)}\cdot\vec{x}} \\ \vdots \\ e^{\vec{w}^{(K)}\cdot\vec{x}} \end{bmatrix} \end{array} \begin{array}{l} \to s_1 \\ \to s_2 \\ \\ \to s_K \end{array}$$

Sum to 1

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \\ \\ \\ \leftarrow \text{true class} \\ \\ \\ \end{array}$$





$\boxed{K=5}$

$$y_i \log h(\vec{x}_i) + (1-y_i)\log(1-h(\vec{x}))$$