Review of Lecture 8

- **Bias and variance**

  Expected value of E_{out} w.r.t. $\mathcal{D}$

  $$= \textcolor{blue}{\text{bias}} + \textcolor{red}{\text{var}}$$
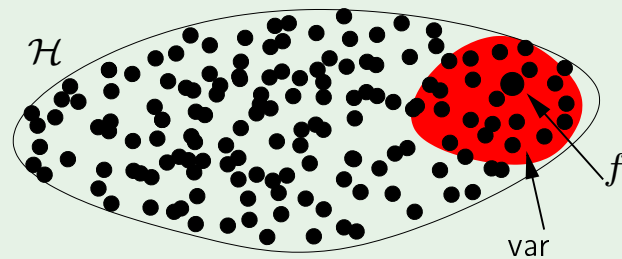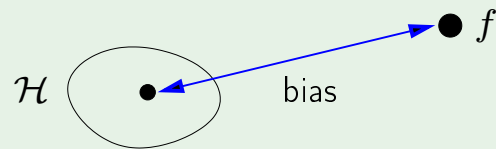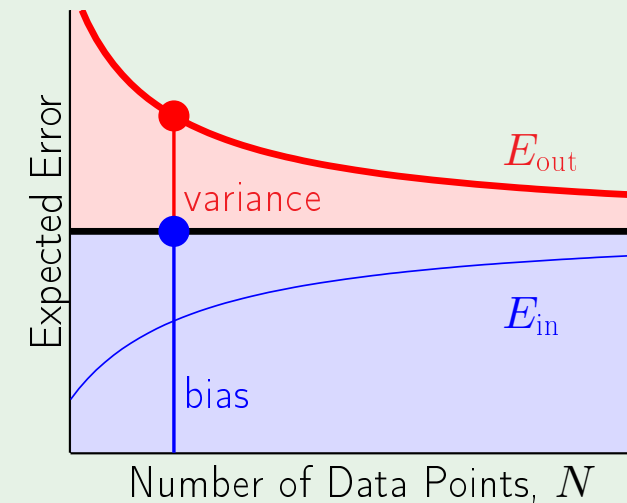

$$g^{(\mathcal{D})}(\mathbf{x}) \longrightarrow \bar{g}(\mathbf{x}) \longrightarrow f(\mathbf{x})$$
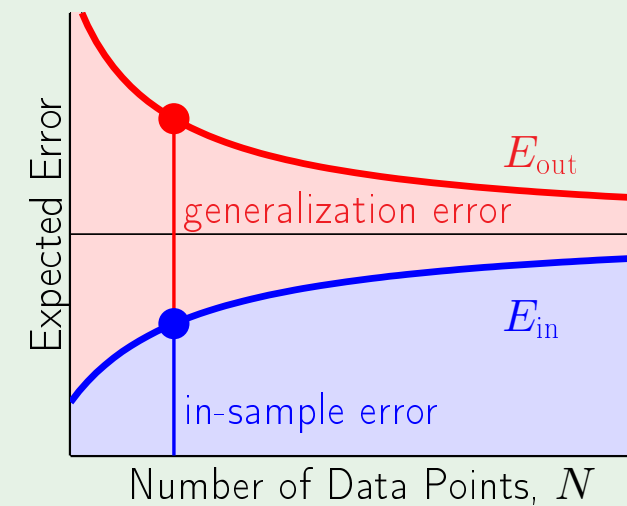
- **Learning curves**

  How E_{in} and E_{out} vary with N

  B-V:

  VC:

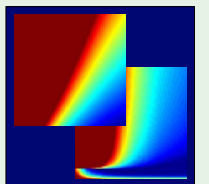- $N \propto$ "VC dimension"

Learning From Data

Yaser S. Abu-Mostafa
California Institute of Technology

Lecture 9: **The Linear Model II**

Where we are

- Linear classification ✓

- Linear regression ✓

- Logistic regression

- Nonlinear transforms ✓

Nonlinear transforms

$$\mathbf{x} = (x_0, x_1, \cdots, x_d) \quad \xrightarrow{\;\Phi\;} \quad \mathbf{z} = (z_0, z_1, \cdots\cdots\cdots\cdots, z_{\tilde{d}})$$

Each $z_i = \phi_i(\mathbf{x})$ $\qquad$ $\mathbf{z} = \Phi(\mathbf{x})$

Example: $\quad \mathbf{z} = (1, x_1, x_2, x_1 x_2, x_1^2, x_2^2)$

Final hypothesis $g(\mathbf{x})$ in $\mathcal{X}$ space:

$$\text{sign}\left(\tilde{\mathbf{w}}^{\top} \Phi(\mathbf{x})\right) \qquad \text{or} \qquad \tilde{\mathbf{w}}^{\top} \Phi(\mathbf{x})$$

The price we pay

$$\mathbf{x} = (x_0, x_1, \cdots, x_d) \quad \overset{\Phi}{\longrightarrow} \quad \mathbf{z} = (z_0, z_1, \cdots\cdots\cdots\cdots, z_{\tilde{d}})$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \downarrow$$

$$\mathbf{w} \qquad\qquad\qquad\qquad\qquad\qquad \tilde{\mathbf{w}}$$

$$d_{\text{VC}} = d + 1 \qquad\qquad\qquad\qquad\qquad d_{\text{VC}} \leq \tilde{d} + 1$$
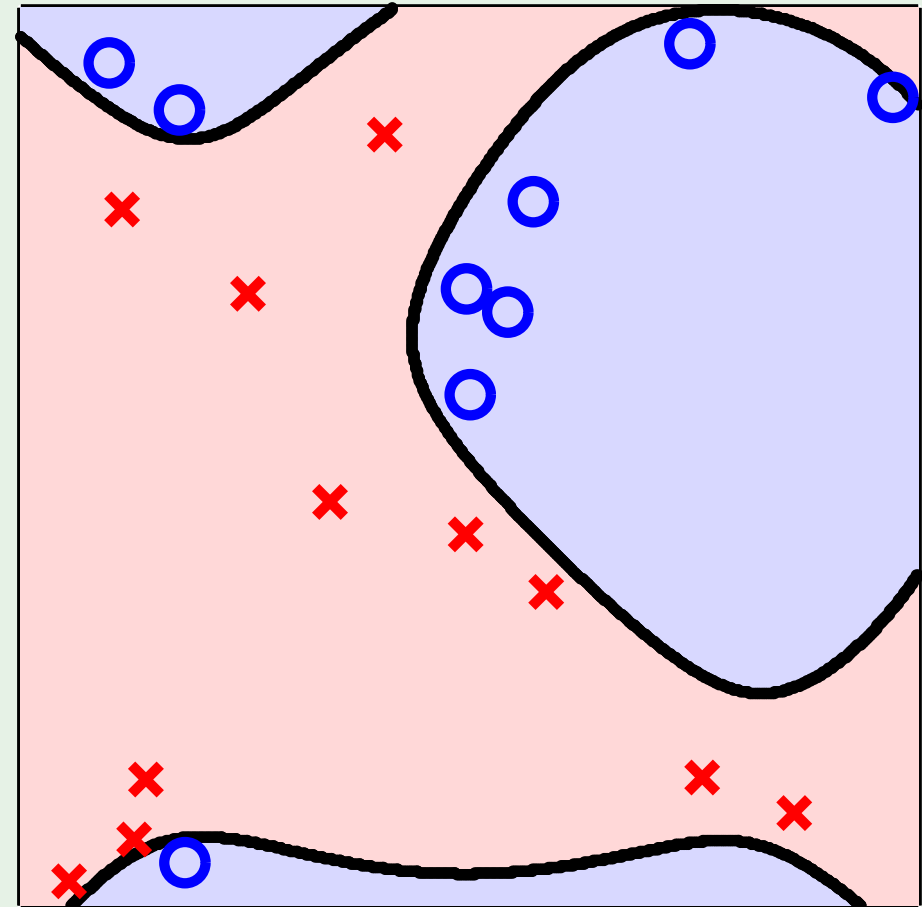
Two non-separable cases
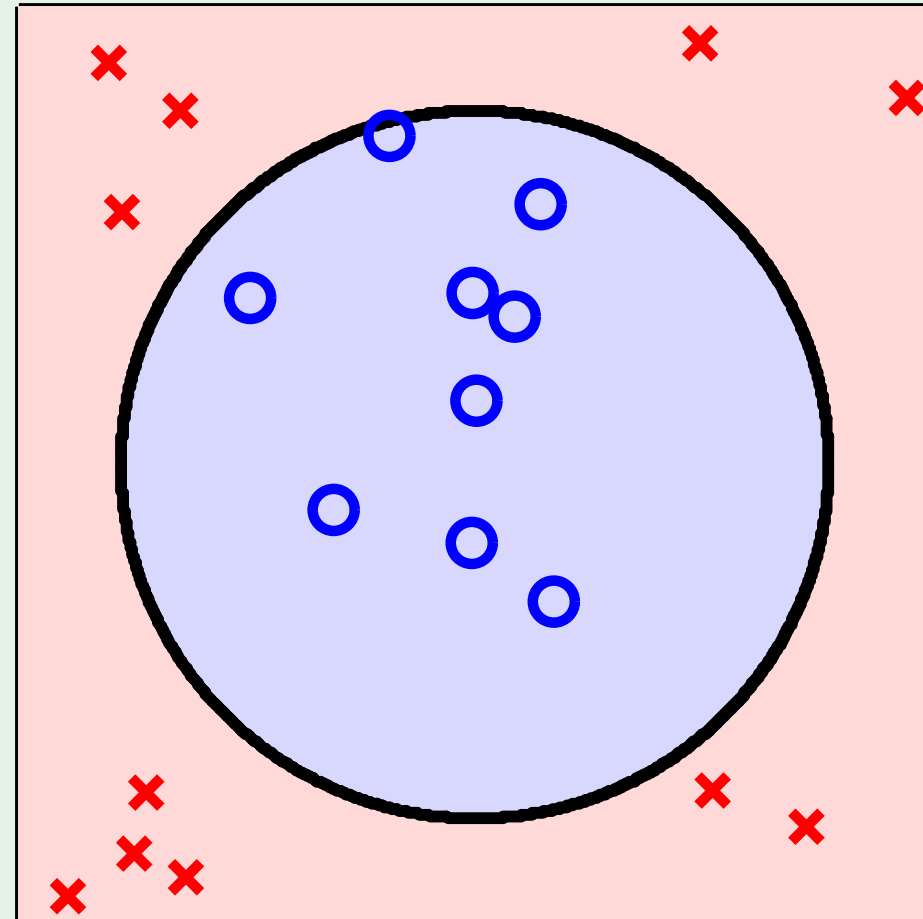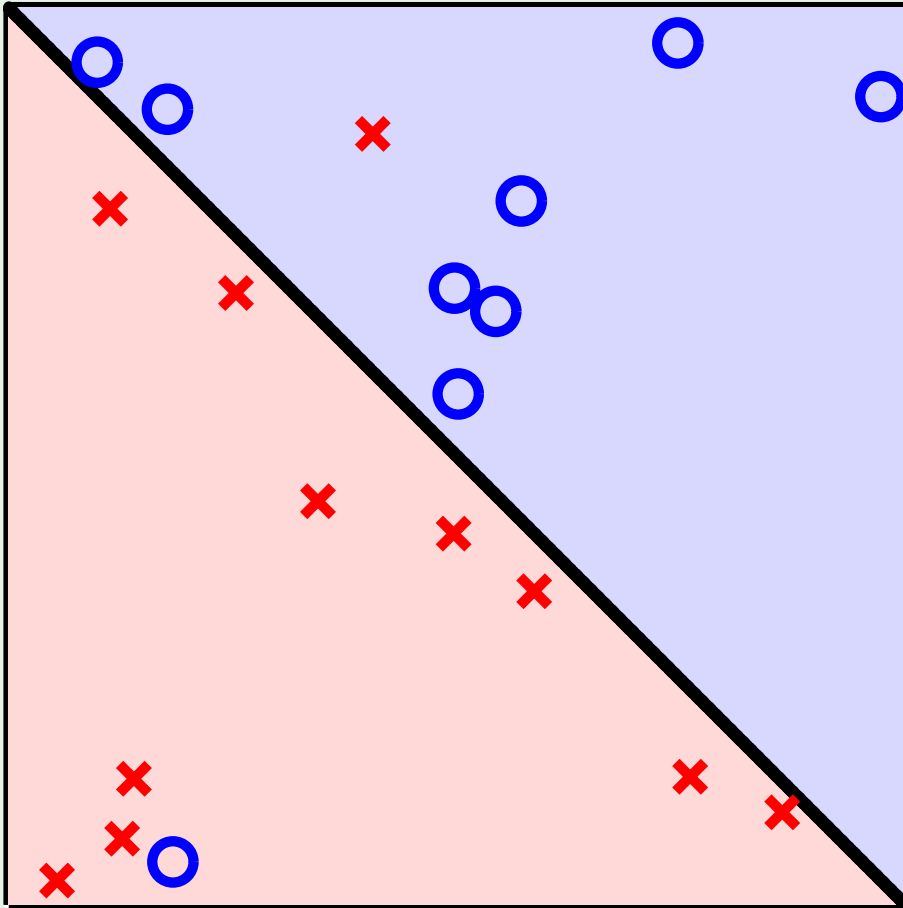
First case

Use a linear model in $\mathcal{X}$; accept $E_{\text{in}} > 0$

or

Insist on $E_{\text{in}} = 0$; go to high-dimensional $\mathcal{Z}$
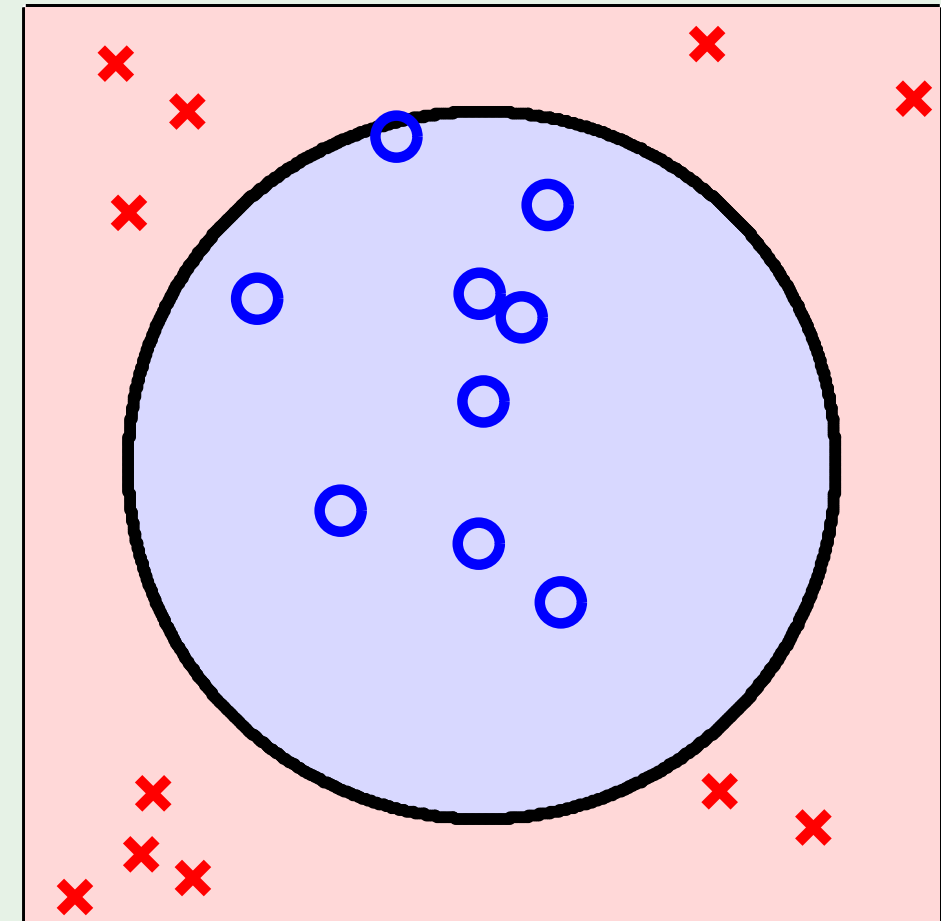
Second case

$$\mathbf{z} = (1, x_1, x_2, x_1 x_2, x_1^2, x_2^2)$$

Why not: $\quad\quad \mathbf{z} = (1, x_1^2, x_2^2)$

or better yet: $\quad \mathbf{z} = (1, x_1^2 + x_2^2)$

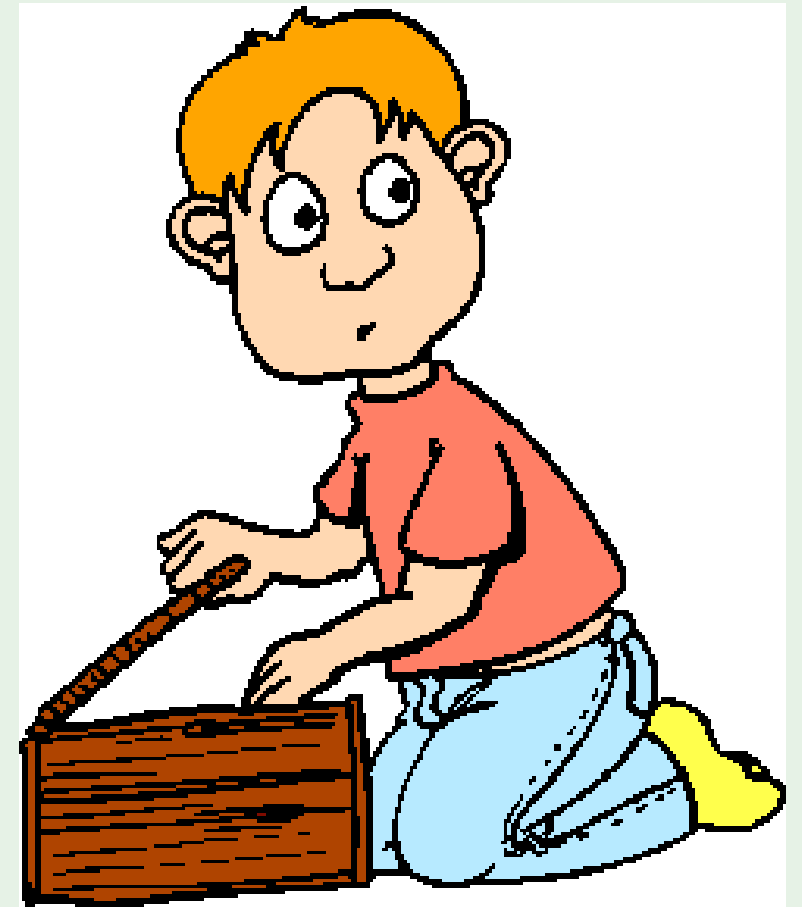or even: $\quad\quad \mathbf{z} = (x_1^2 + x_2^2 - 0.6)$

Lesson learned

Looking at the data *before* choosing the model can be hazardous to your E_{out}

Data snooping

Logistic regression – Outline

- **The model**

- **Error measure**

- **Learning algorithm**

A third linear model

$$s = \sum_{i=0}^{d} w_i x_i$$
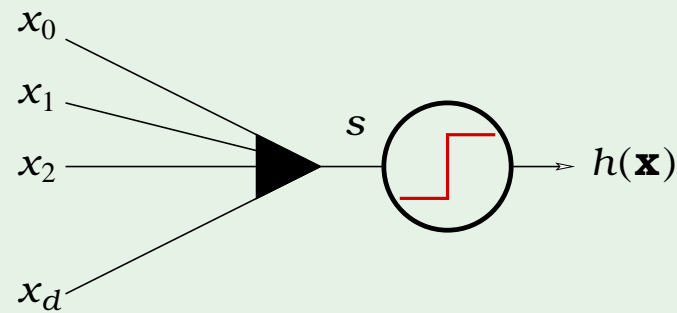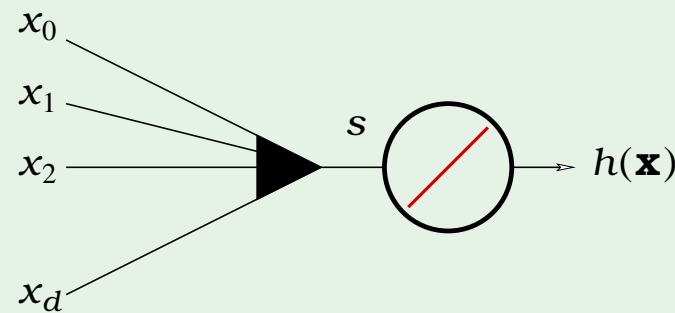
linear classification	linear regression	logistic regression
$h(\mathbf{x}) = \mathrm{sign}(s)$	$h(\mathbf{x}) = s$	$h(\mathbf{x}) = \theta(s)$
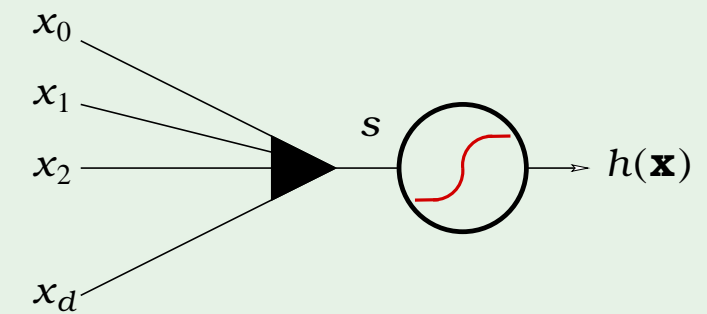
The logistic function θ

The formula:

$$\theta(s) = \frac{e^s}{1 + e^s}$$
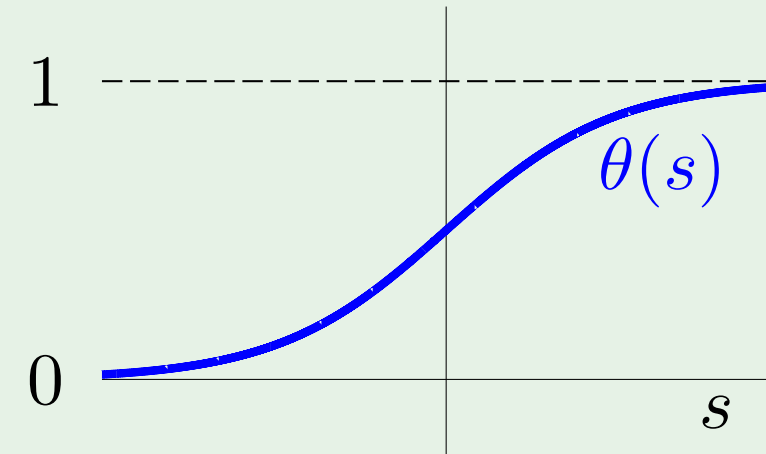
soft threshold: uncertainty

sigmoid: flattened out 's'

Probability interpretation

$h(\mathbf{x}) = \theta(s)$ is interpreted as a probability

Example. Prediction of heart attacks

Input $\mathbf{x}$: cholesterol level, age, weight, etc.

$\theta(s)$: probability of a heart attack

The signal $s = \mathbf{w}^\mathsf{T}\mathbf{x}$ "risk score"

Genuine probability

Data $(\mathbf{x}, y)$ with binary y, generated by a noisy target:

$$P(y \mid \mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{for } y = +1; \\ 1 - f(\mathbf{x}) & \text{for } y = -1. \end{cases}$$

The target $f : \mathbb{R}^d \to [0, 1]$ is the probability

Learn $g(\mathbf{x}) = \theta(\mathbf{w}^\intercal \mathbf{x}) \approx f(\mathbf{x})$

Error measure

For each $(\mathbf{x}, y)$, y is generated by probability $f(\mathbf{x})$

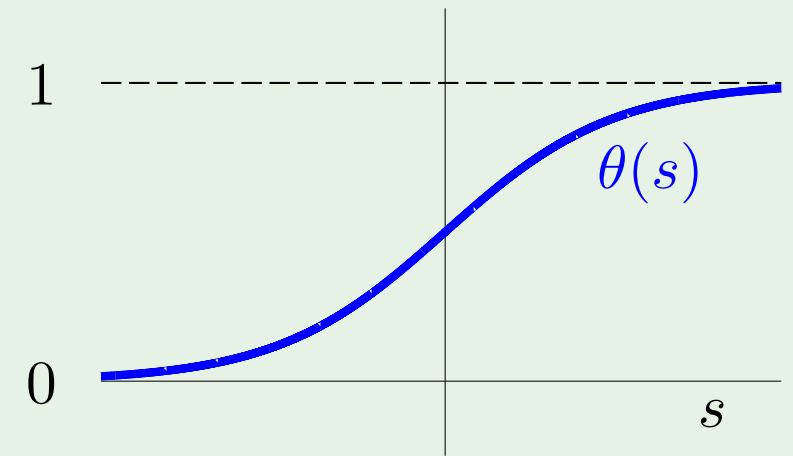Plausible error measure based on **likelihood**:

If $h = f$, how likely to get y from $\mathbf{x}$?

$\mathbf{x}$?

$$P(y \mid \mathbf{x}) = \begin{cases} h(\mathbf{x}) & \text{for } y = +1; \\ 1 - h(\mathbf{x}) & \text{for } y = -1. \end{cases}$$

Formula for likelihood

$$P(y \mid \mathbf{x}) = \begin{cases} h(\mathbf{x}) & \text{for } y = +1; \\ 1 - h(\mathbf{x}) & \text{for } y = -1. \end{cases}$$

$\theta(s)$

Substitute $h(\mathbf{x}) = \theta(\mathbf{w}^{\mathsf{T}}\mathbf{x})$, noting $\theta(-s) = 1 - \theta(s)$

$$P(y \mid \mathbf{x}) = \theta(y\,\mathbf{w}^{\mathsf{T}}\mathbf{x})$$

Likelihood of $\mathcal{D} = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ is

$$\prod_{n=1}^{N} P(y_n \mid \mathbf{x}_n) = \prod_{n=1}^{N} \theta(y_n\mathbf{w}^{\mathsf{T}}\mathbf{x}_n)$$

Maximizing the likelihood

Minimize

$$-\frac{1}{N} \ln \left(\prod_{n=1}^{N} \theta(y_n \, \mathbf{w}^{\mathsf{T}} \, \mathbf{x}_n) \right)$$

$$= \frac{1}{N} \sum_{n=1}^{N} \ln \left(\frac{1}{\theta(y_n \, \mathbf{w}^{\mathsf{T}} \, \mathbf{x}_n)} \right) \qquad \left[\theta(s) = \frac{1}{1 + e^{-s}} \right]$$

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \underbrace{\ln \left(1 + e^{-y_n \mathbf{w}^{\mathsf{T}} \mathbf{x}_n} \right)}_{\mathsf{e}\left(h(\mathbf{x}_n), y_n \right)} \qquad \text{``cross-entropy'' error}$$

- The model

- Error measure

- **Learning algorithm**

How to minimize E_{in}

For logistic regression,

$$E_{\text{in}}(\mathbf{w}) \;=\; \frac{1}{N} \sum_{n=1}^{N} \ln\left(1 + e^{-y_n \mathbf{w}^{\mathsf{T}} \mathbf{x}_n}\right) \qquad \longleftarrow \textbf{iterative } \text{solution}$$

Compare to linear regression:

$$E_{\text{in}}(\mathbf{w}) \;=\; \frac{1}{N} \sum_{n=1}^{N} \left(\mathbf{w}^{\mathsf{T}} \mathbf{x}_n - y_n\right)^2 \qquad \longleftarrow \text{closed-form solution}$$
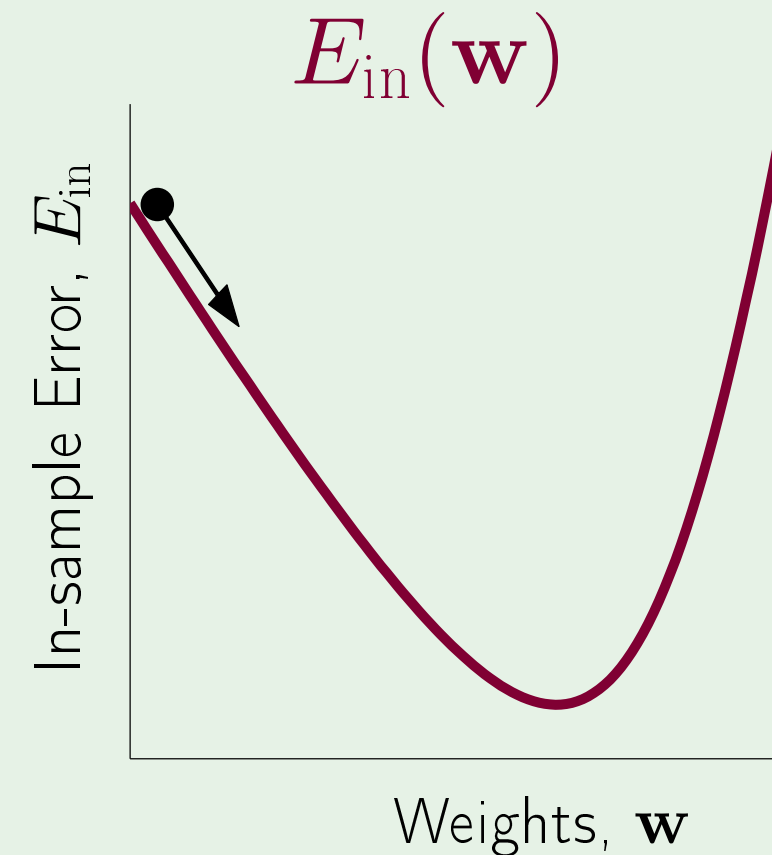
Iterative method: gradient descent

General method for nonlinear optimization

Start at $\mathbf{w}(0)$; take a step along steepest slope

Fixed step size: $\mathbf{w}(1) \;=\; \mathbf{w}(0) + \eta\,\hat{\mathbf{v}}$

What is the direction $\hat{\mathbf{v}}$?

$E_{\text{in}}(\mathbf{w})$

In-sample Error, E_{in}

Weights, $\mathbf{w}$

Formula for the direction $\hat{\mathbf{v}}$
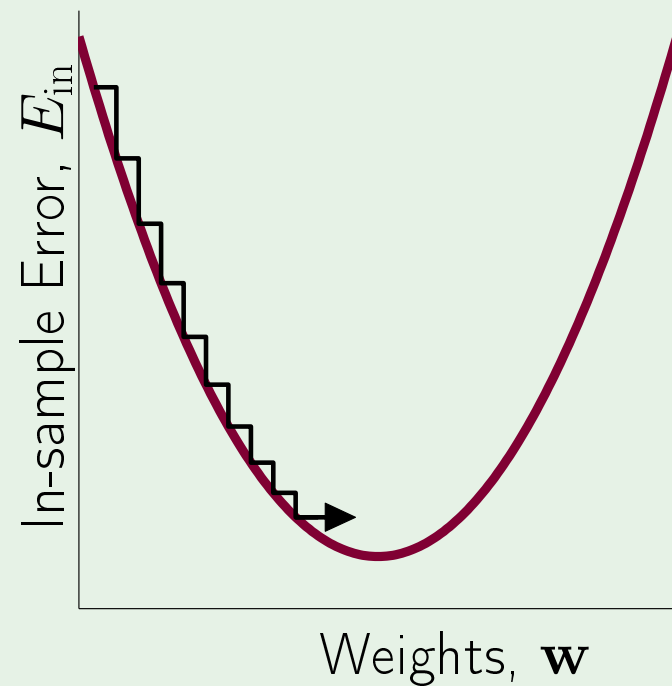
$$\Delta E_{\text{in}} = E_{\text{in}}(\,\mathbf{w}(0) + \eta\hat{\mathbf{v}}\,) - E_{\text{in}}(\mathbf{w}(0))$$

$$= \eta\nabla E_{\text{in}}(\mathbf{w}(0))^{\text{T}}\hat{\mathbf{v}} + O(\eta^2)$$

$$\geq -\eta\|\nabla E_{\text{in}}(\mathbf{w}(0))\|$$
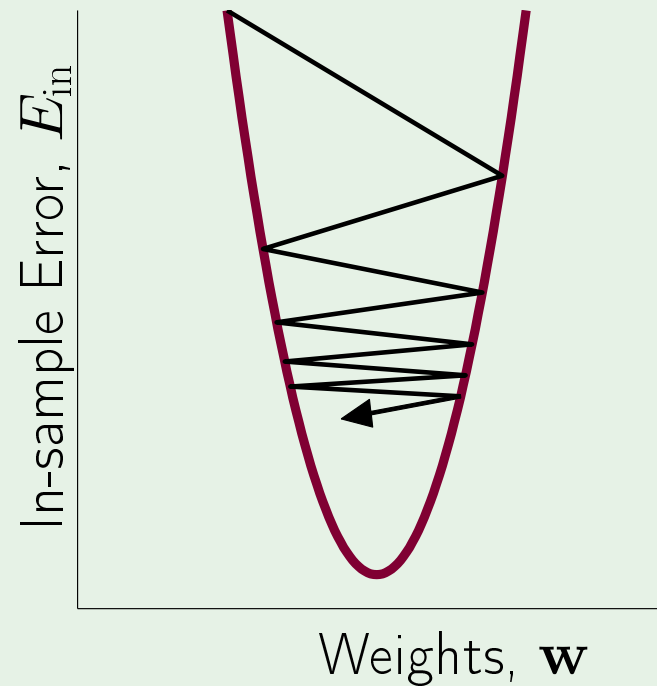
Since $\hat{\mathbf{v}}$ is a unit vector,

$$\hat{\mathbf{v}} = -\frac{\nabla E_{\text{in}}(\mathbf{w}(0))}{\|\nabla E_{\text{in}}(\mathbf{w}(0))\|}$$
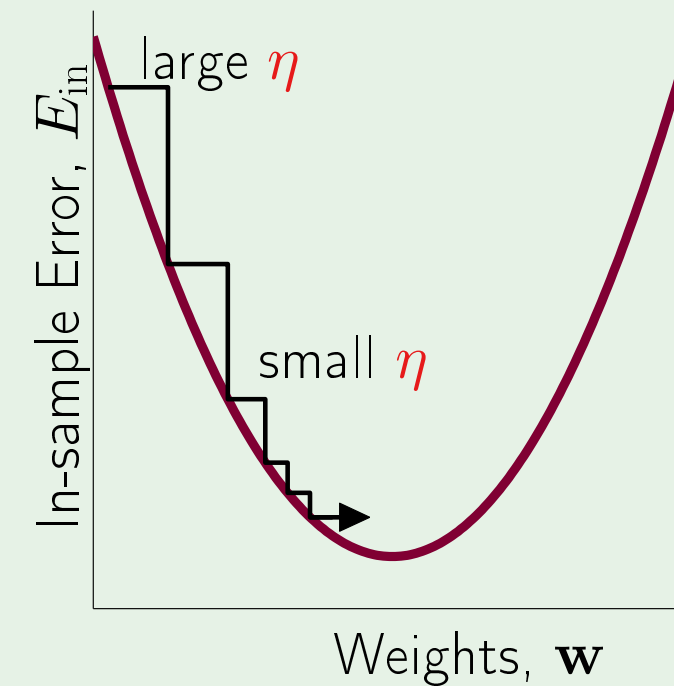
Fixed-size step?

How η affects the algorithm:

η too small η too large   variable η – just right

η should increase with the slope

Easy implementation

Instead of

$$\Delta \mathbf{w} \;=\; \eta \; \hat{\mathbf{v}}$$

$$=\; -\, \eta \; \frac{\nabla E_{\text{in}}(\mathbf{w}(0))}{\|\nabla E_{\text{in}}(\mathbf{w}(0))\|}$$

Have

$$\Delta \mathbf{w} \;=\; -\, \eta \; \nabla E_{\text{in}}(\mathbf{w}(0))$$

Fixed **learning rate** η

Logistic regression algorithm

1: Initialize the weights at $t = 0$ to $\mathbf{w}(0)$

2: **for** $t = 0, 1, 2, \ldots$ **do**

3:     Compute the gradient

$$\nabla E_{\text{in}} = -\frac{1}{N} \sum_{n=1}^{N} \frac{y_n \mathbf{x}_n}{1 + e^{y_n \mathbf{w}^{\mathsf{T}}(t) \mathbf{x}_n}}$$

4:     Update the weights: $\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \nabla E_{\text{in}}$

5:     Iterate to the next step until it is time to stop

6: Return the final weights $\mathbf{w}$

Summary of Linear Models