

10-301/601: Introduction to Machine Learning

Lecture 9 – Logistic Regression

Henry Chai

5/19/25

Recall: Probabilistic Learning

- Previously:
 - (Unknown) Target function, $c^*: \mathcal{X} \rightarrow \mathcal{Y}$
 - Classifier, $h: \mathcal{X} \rightarrow \mathcal{Y}$
 - Goal: find a classifier, h , that best approximates c^*
- Now:
 - (Unknown) Target *distribution*, $y \sim P^*(Y|\mathbf{x})$
 - Distribution, $P(Y|\mathbf{x})$
 - Goal: find a distribution, P , that best approximates P^*

Building a Probabilistic Classifier

1. Define a decision rule
 - Given a test data point $\mathbf{x}'$, predict its label $\hat{y}$ using the *posterior distribution* $P(Y = y|X = \mathbf{x}')$
 - Common choice: $\hat{y} = \underset{y}{\operatorname{argmax}} P(Y = y|X = \mathbf{x}')$
2. Model the posterior distribution
 - Option 1 - Model $P(Y|X)$ directly as some function of X (today!)
 - Option 2 - Use Bayes' rule (later):
 - $P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \propto P(X|Y) P(Y)$

Modelling the Posterior

- Suppose we have binary labels $y \in \{0,1\}$ and D -dimensional inputs $\mathbf{x} = [1, x_1, \dots, x_D]^T \in \mathbb{R}^{D+1}$

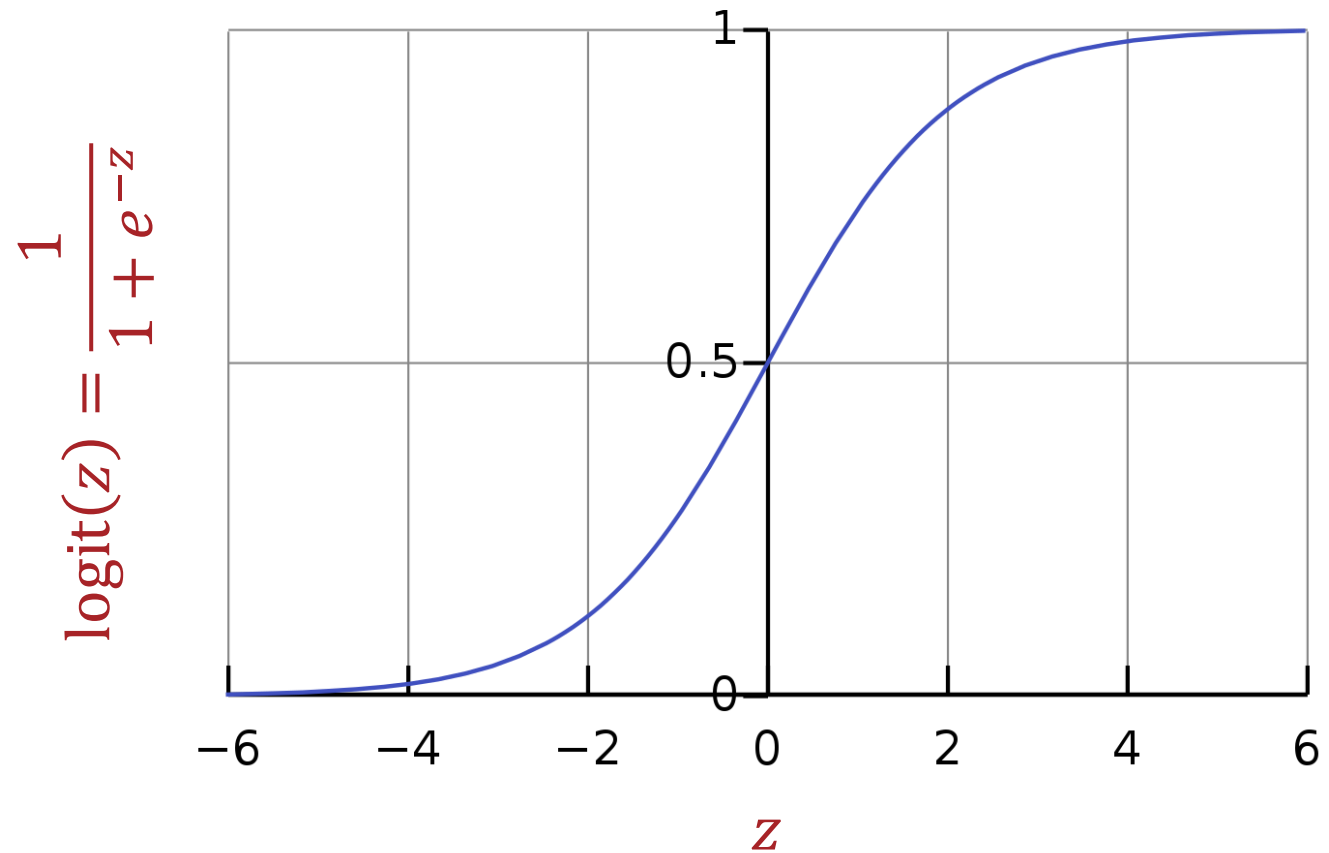
- **Assume**

$$P(Y = 1|\mathbf{x}) = \text{logit}(\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})}$$
$$= \frac{\exp(\mathbf{w}^T \mathbf{x})}{\exp(\mathbf{w}^T \mathbf{x}) + 1}$$

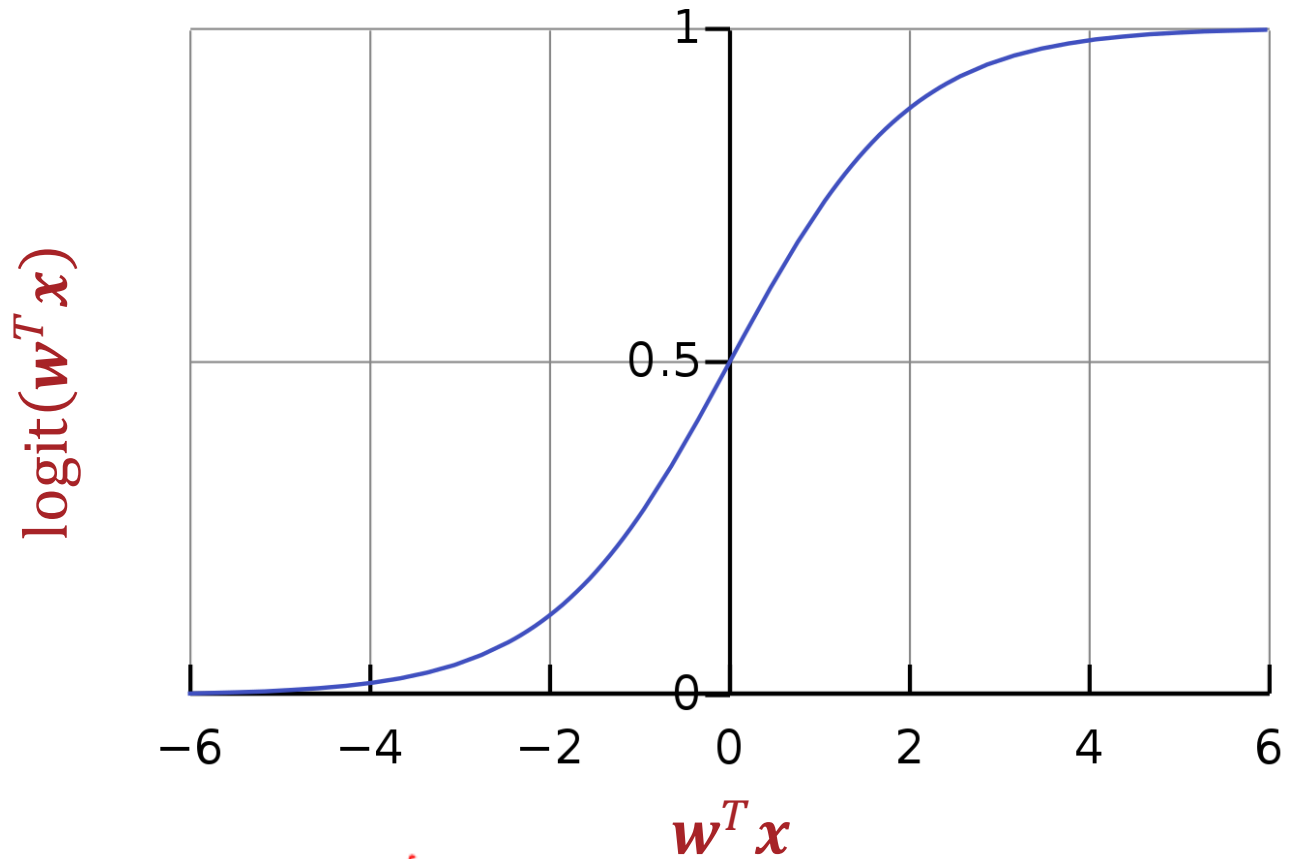
- This implies two useful facts:

$$- P(Y=0|\vec{x}) = 1 - P(Y=1|\vec{x}) = 1 - \frac{e^{\vec{w}^T \vec{x}}}{e^{\vec{w}^T \vec{x}} + 1} = \frac{1}{\exp(\vec{w}^T \vec{x}) + 1}$$
$$- \frac{P(Y=1|\vec{x})}{P(Y=0|\vec{x})} = \exp(\vec{w}^T \vec{x}) \Rightarrow \log \text{ odds} = \vec{w}^T \vec{x}$$

Logistic Function



Why use the Logistic Function?



- monotonically increasing
- $\text{logit}(\vec{w}^T \vec{x}) \in (0, 1)$
- differentiable — leads to a linear decision boundary

Logistic Regression Decision Boundary

$$\hat{y} = \begin{cases} 1 & \text{if } P(Y = 1|\mathbf{x}) \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

$$P(Y=1|\vec{x}) = \frac{1}{1 + \exp(-\vec{w}^T \vec{x})} = \frac{1}{2}$$

$$\Rightarrow 2 = 1 + \exp(-\vec{w}^T \vec{x})$$

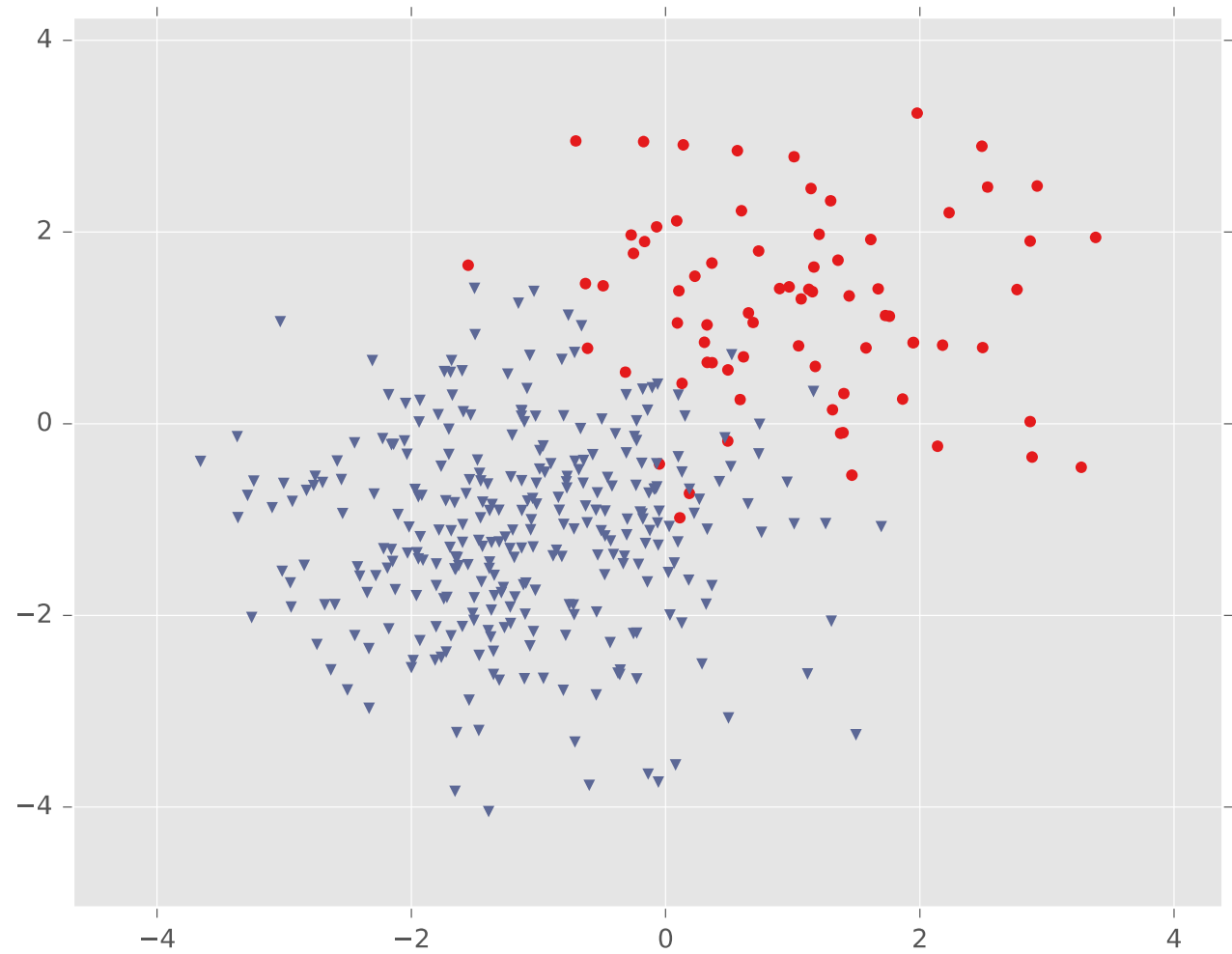
$$\Rightarrow \log(1) = -\vec{w}^T \vec{x}$$

$$\Rightarrow 0 = -\vec{w}^T \vec{x}$$

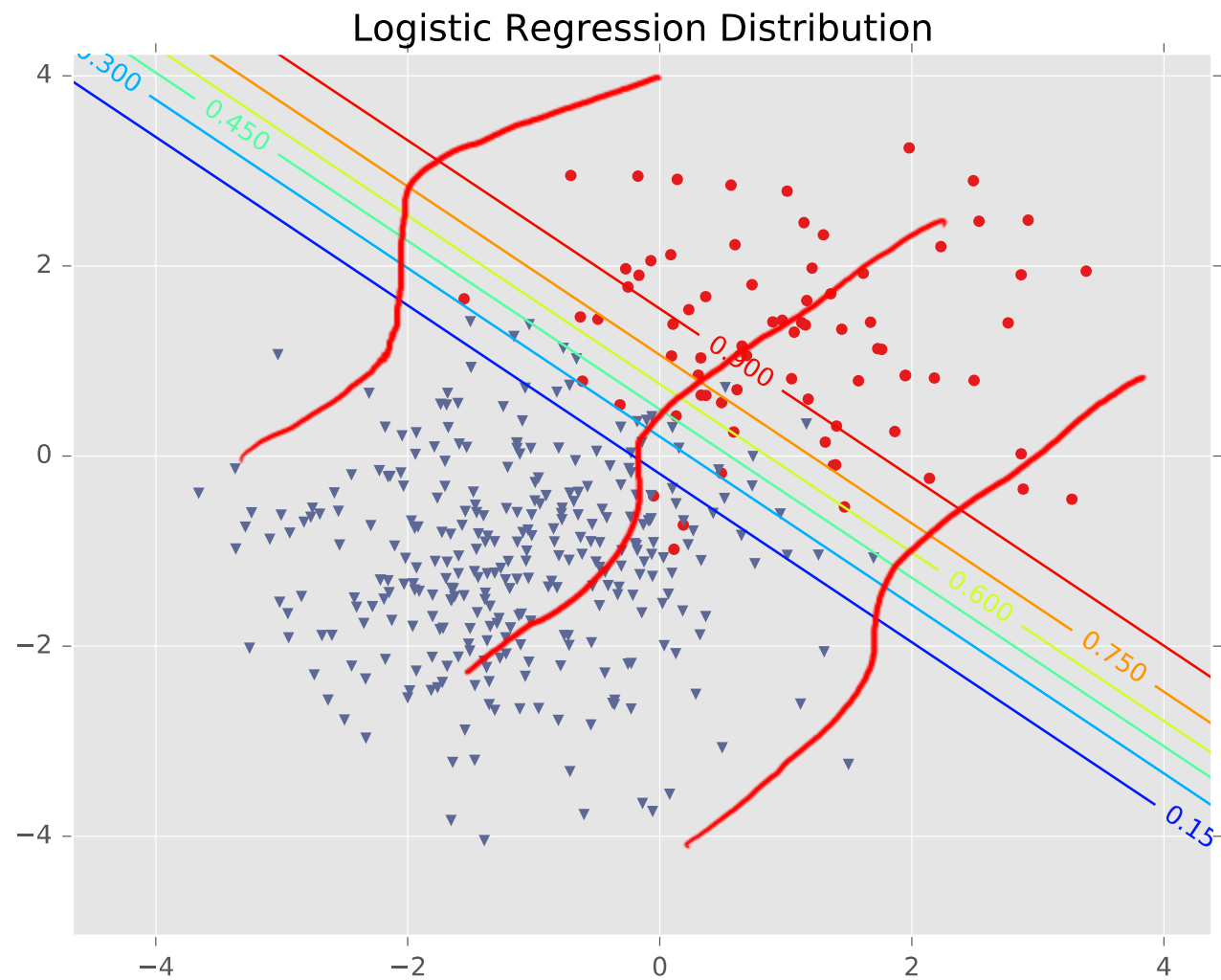
$$\Rightarrow \vec{w}^T \vec{x} = 0$$

hyperplane $\rightarrow$

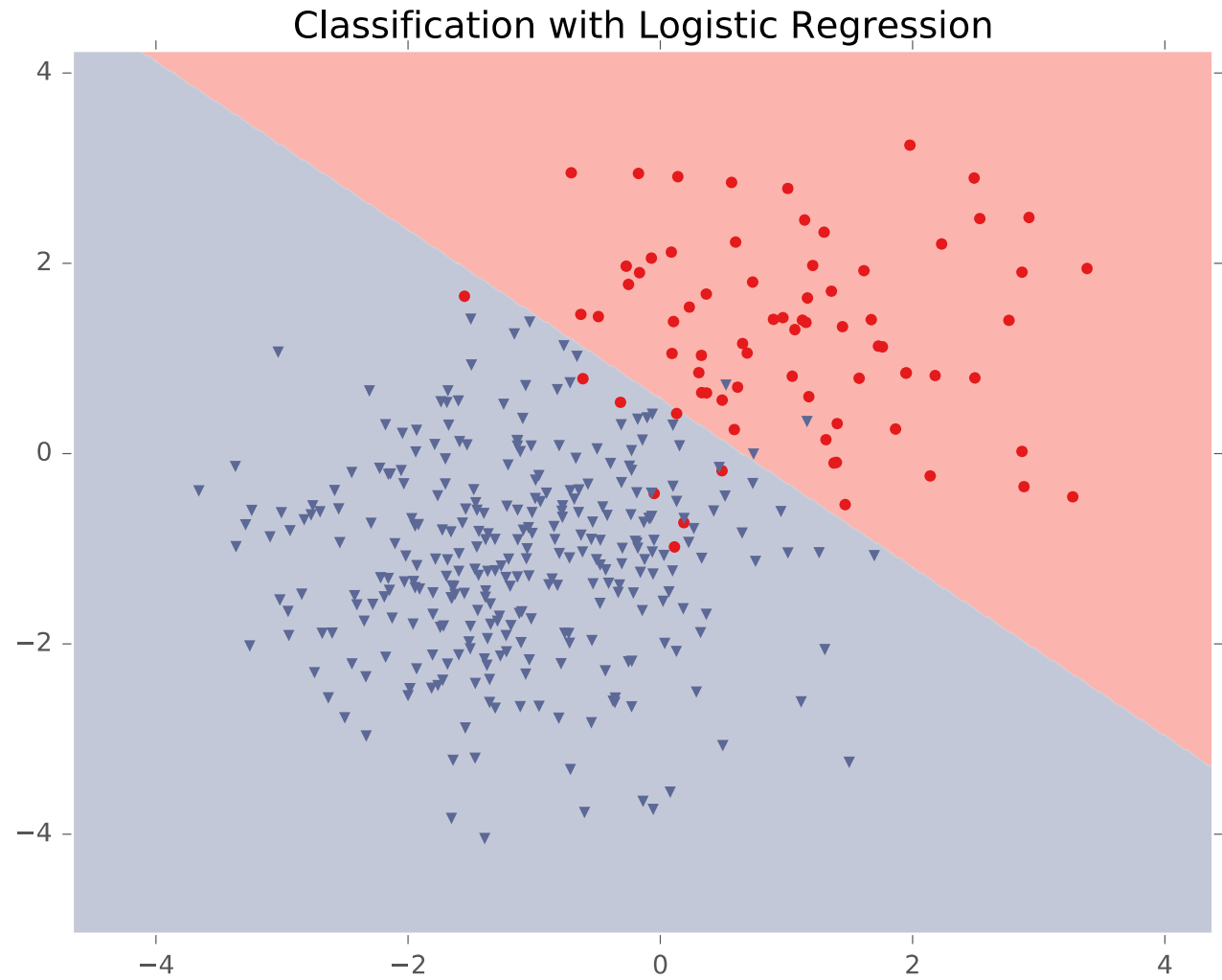
Logistic Regression Decision Boundary



Logistic Regression Decision Boundary



Logistic Regression Decision Boundary



General Recipe for Machine Learning

- Define a model and model parameters
- Write down an objective function
- Optimize the objective w.r.t. the model parameters

Recipe for Logistic Regression

- Define a model and model parameters
 - Assume data points are independent, identically distributed (iid)
 - Assume $P(Y=1|\vec{x}) = \text{logit}(\vec{\theta}^T \vec{x})$
 - parameters: $\vec{\theta} = [w_0, w_1, \dots, w_D]^T$
- Write down an objective function
 - Maximize the "conditional" log-likelihood
 - $\Downarrow$
 - minimize the negative "conditional" log-likelihood
- Optimize the objective w.r.t. the model parameters

???

Setting the Battleground for the War of Attrition Negative Log-Likelihood Conditional Log-Likelihood Estimation! (MCLE)

Find θ that minimizes

$$\begin{aligned}
 \ell(\vec{\theta}) &= -\log P(y^{(1)}, y^{(2)}, \dots, y^{(N)} | \vec{x}^{(1)}, \vec{x}^{(2)}, \dots, \vec{x}^{(N)}, \vec{\theta}) \\
 (\text{by iid}) &= -\log \prod_{n=1}^N P(y^{(n)} | \vec{x}^{(n)}, \vec{\theta}) \\
 &= -\log \prod_{n=1}^N P(Y=1 | \vec{x}^{(n)}, \vec{\theta})^{y^{(n)}} P(Y=0 | \vec{x}^{(n)}, \vec{\theta})^{(1-y^{(n)})} \\
 &= -\sum_{n=1}^N y^{(n)} \log P(Y=1 | \vec{x}^{(n)}, \vec{\theta}) + (1-y^{(n)}) \log P(Y=0 | \vec{x}^{(n)}, \vec{\theta}) \\
 &= -\sum_{n=1}^N y^{(n)} (\log P(Y=1 | \vec{x}^{(n)}, \vec{\theta}) - \log P(Y=0 | \vec{x}^{(n)}, \vec{\theta})) \\
 &\quad + \log P(Y=0 | \vec{x}^{(n)}, \vec{\theta}) \\
 &= -\sum_{n=1}^N y^{(n)} \vec{\theta}^T \vec{x}^{(n)} + \log \left(\frac{1}{1 + \exp(\vec{\theta}^T \vec{x}^{(n)})} \right)
 \end{aligned}$$

Minimizing the Negative Conditional (log-)Likelihood

$$\ell_D(\boldsymbol{\theta}) = - \sum_{n=1}^N y^{(n)} \boldsymbol{\theta}^T \mathbf{x}^{(n)} - \log \left(1 + \exp(\boldsymbol{\theta}^T \mathbf{x}^{(n)}) \right)$$

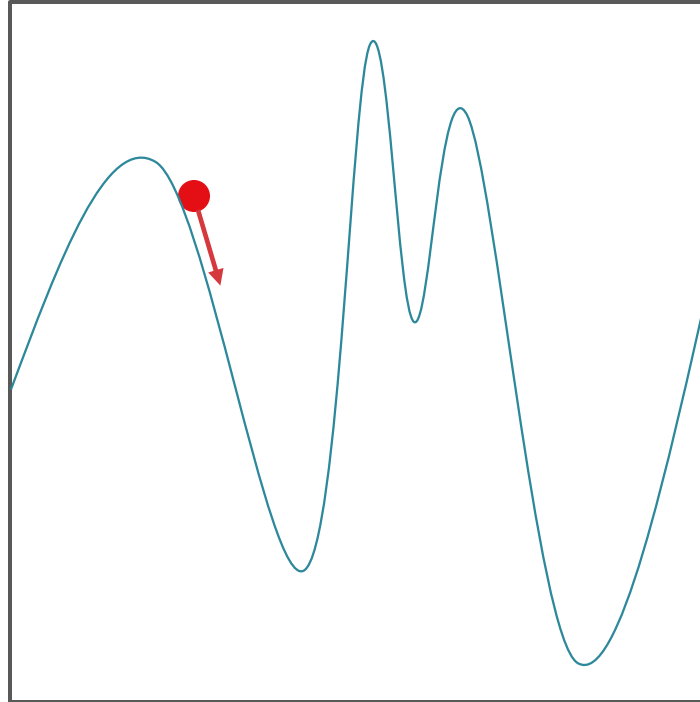
$$\nabla_{\vec{\theta}} \ell_D(\vec{\theta}) = \begin{bmatrix} \partial \ell_D / \partial w_0 \\ \partial \ell_D / \partial w_1 \\ \vdots \\ \partial \ell_D / \partial w_D \end{bmatrix}$$

$$= - \sum_{n=1}^N y^{(n)} \vec{x}^{(n)} - \frac{\exp(\vec{\theta}^T \vec{x}^{(n)})}{1 + \exp(\vec{\theta}^T \vec{x}^{(n)})} \vec{x}^{(n)}$$

$$= - \sum_{n=1}^N (y^{(n)} - P(Y=1 | \vec{x}^{(n)}, \vec{\theta})) \vec{x}^{(n)}$$

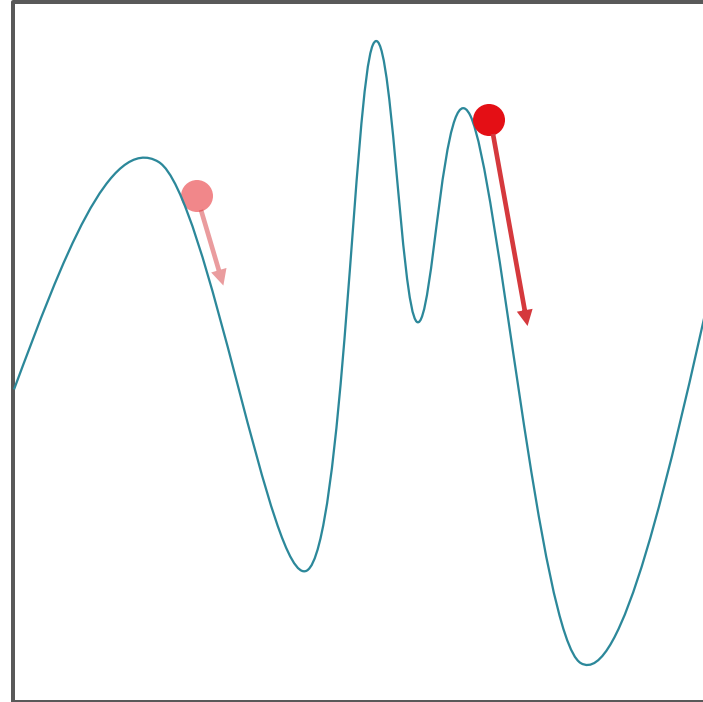
Gradient Descent: Intuition

- An iterative method for minimizing functions
- Requires the gradient to exist everywhere



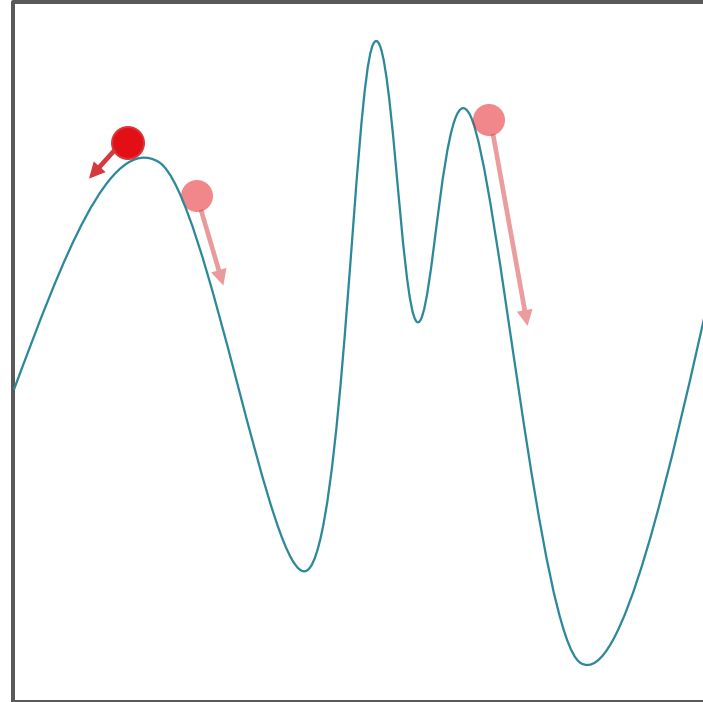
Gradient Descent: Intuition

- An iterative method for minimizing functions
- Requires the gradient to exist everywhere



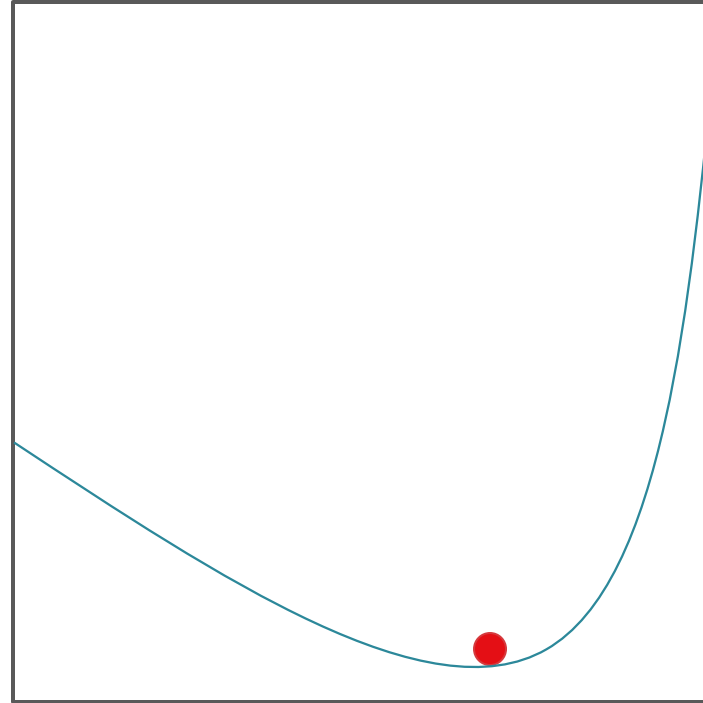
Gradient Descent: Intuition

- An iterative method for minimizing functions
- Requires the gradient to exist everywhere



Gradient Descent: Intuition

- An iterative method for minimizing functions
- Requires the gradient to exist everywhere



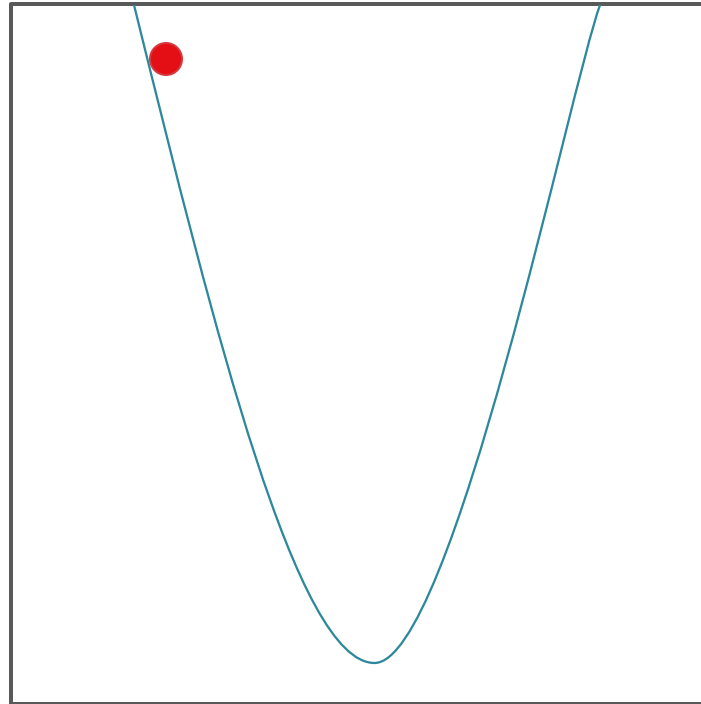
- Good news: the negative conditional log-likelihood is *convex*!

Gradient Descent: Step Direction

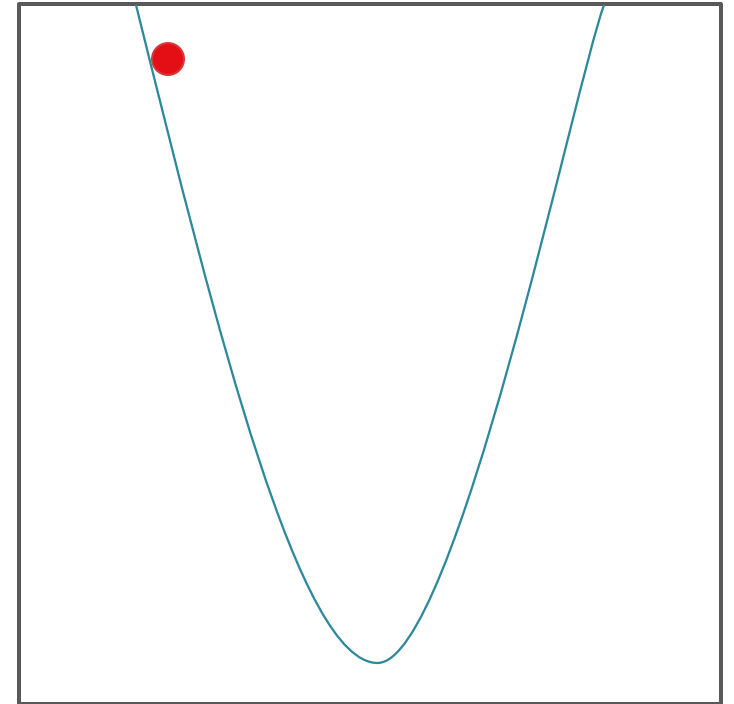
- Suppose the current parameter vector is $\boldsymbol{\theta}^{(t)}$
- Move some distance, η , in the “most downhill” direction, $\hat{\mathbf{v}}$:
$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \eta \hat{\mathbf{v}}$$
- The gradient points in the direction of steepest *increase* ...
- ... so $\hat{\mathbf{v}}$ is a unit vector pointing in the opposite direction:

$$\hat{\mathbf{v}}^{(t)} = - \frac{\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}(\boldsymbol{\theta}^{(t)})}{\|\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}(\boldsymbol{\theta}^{(t)})\|}$$

Gradient Descent: Step Size

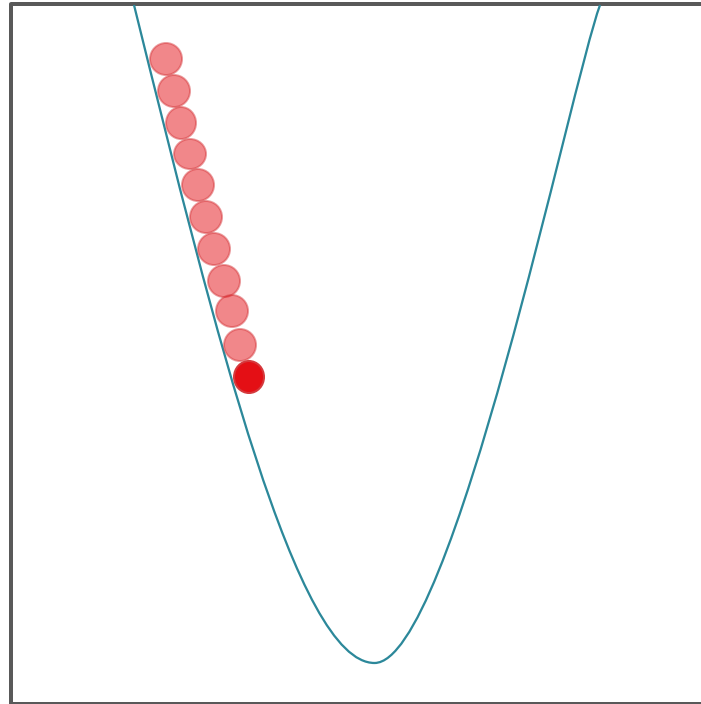


Small η

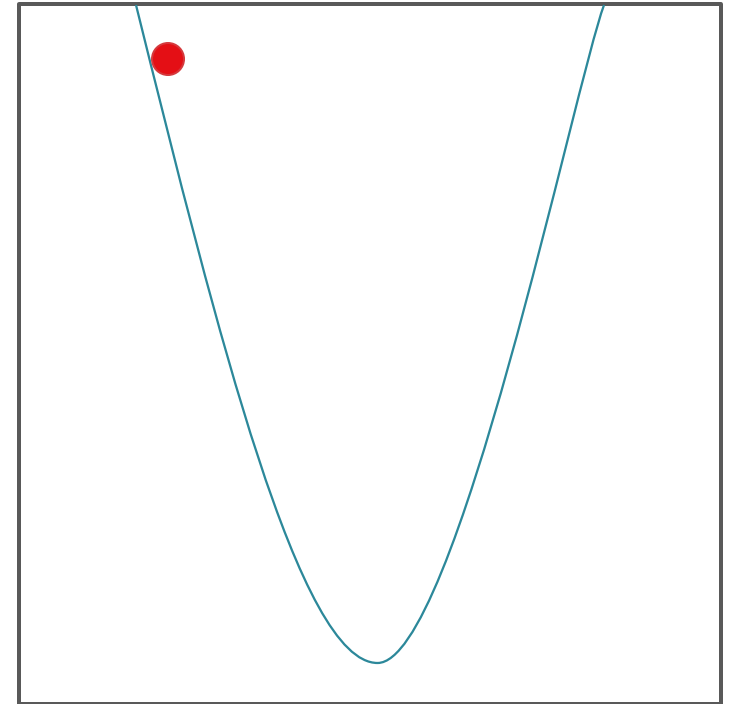


Large η

Gradient Descent: Step Size

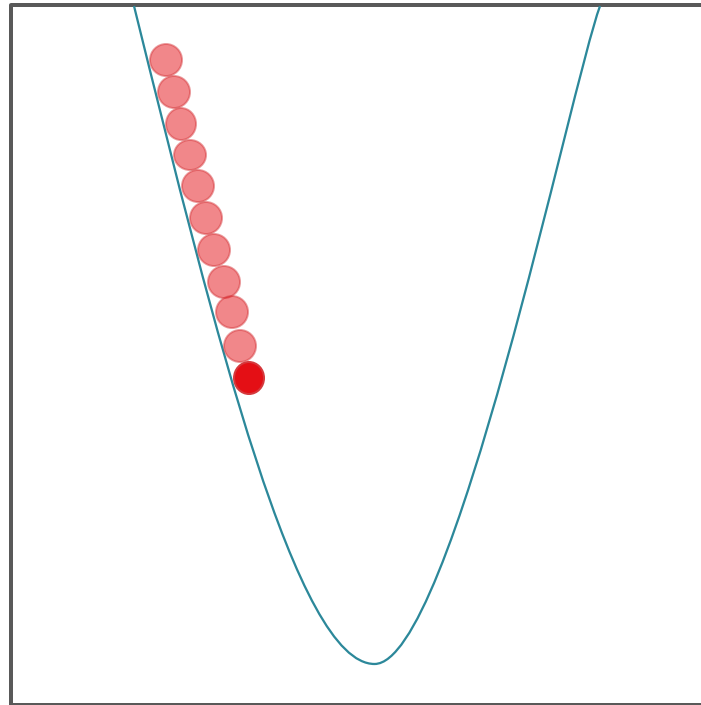


Small η

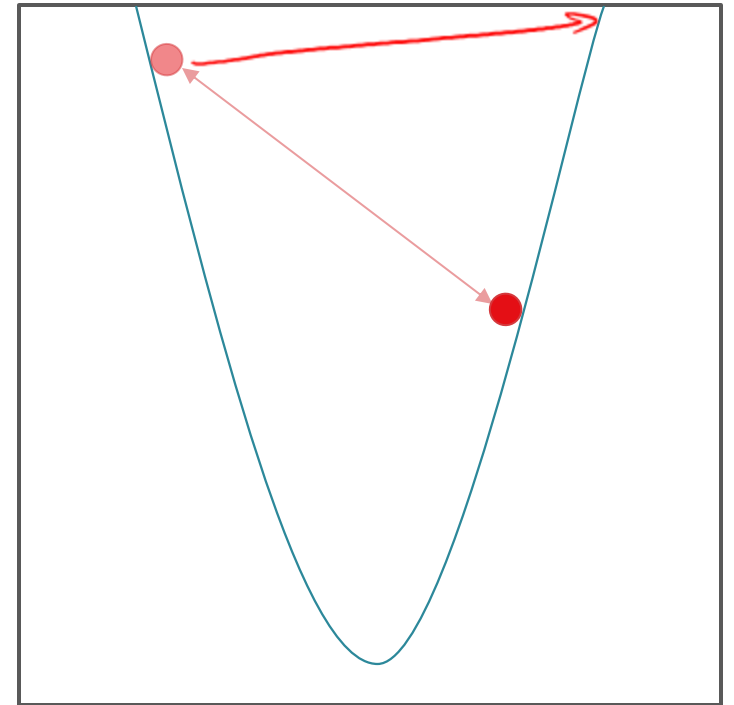


Large η

Gradient Descent: Step Size



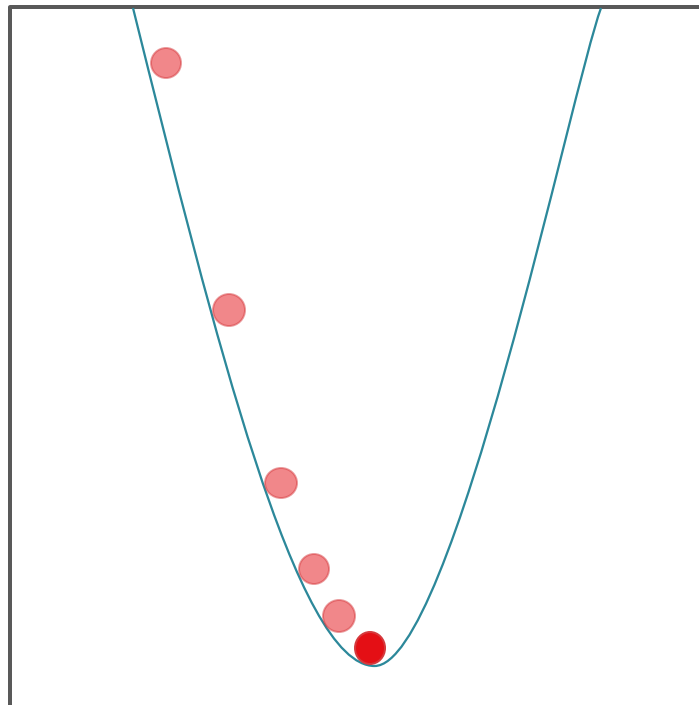
Small η



Large η

Gradient Descent: Step Size

- Use a variable $\eta^{(t)}$ instead of a fixed η !



- Set $\eta^{(t)} = \eta^{(0)} \|\nabla_{\theta} \ell_{\mathcal{D}}(\theta^{(t)})\|$
- $\|\nabla_{\theta} \ell_{\mathcal{D}}(\theta^{(t)})\|$ decreases as $\ell_{\mathcal{D}}$ approaches its minimum $\rightarrow \eta^{(t)}$ (hopefully) decreases over time

Gradient Descent

- $\hat{\mathbf{v}}^{(t)} = -\frac{\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}(\boldsymbol{\theta}^{(t)})}{\|\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}(\boldsymbol{\theta}^{(t)})\|}$

- $\eta^{(t)} = \eta^{(0)} \|\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}(\boldsymbol{\theta}^{(t)})\|$

$$\begin{aligned}\vec{\Theta}^{(t+1)} &= \vec{\Theta}^{(t)} + \eta^{(t)} \hat{\mathbf{v}}^{(t)} \\ &= \vec{\Theta}^{(t)} + \eta^{(0)} (\|\nabla_{\vec{\Theta}} \ell_{\mathcal{D}}(\vec{\Theta}^{(t)})\|) \\ &= \vec{\Theta}^{(t)} - \eta^{(0)} \left(\frac{\nabla_{\vec{\Theta}} \ell_{\mathcal{D}}(\vec{\Theta}^{(t)})}{\|\nabla_{\vec{\Theta}} \ell_{\mathcal{D}}(\vec{\Theta}^{(t)})\|} \right)\end{aligned}$$

Gradient Descent

- Input: $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N, \eta^{(0)}$
 1. Initialize the parameters $\boldsymbol{\theta}^{(0)}$ and set $t = 0$
 2. While TERMINATION CRITERION is not satisfied
 - a. Compute the gradient:
$$\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}(\boldsymbol{\theta}^{(t)})$$
 - b. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta^{(0)} \nabla_{\boldsymbol{\theta}} \ell_{\mathcal{D}}(\boldsymbol{\theta}^{(t)})$
 - c. Increment t : $t \leftarrow t + 1$
- Output: $\boldsymbol{\theta}^{(t)}$

Key Takeaways

- Logistic regression
 - Logistic function induces a linear decision boundary
 - Conditional likelihood maximization
- Gradient descent
 - Effect of step size
 - Termination criteria