10-301/601: Introduction to Machine Learning Lecture 10 – Logistic Regression

Henry Chai

6/6/23

Front Matter

- Announcements:
  - None!
- Recommended Readings:
  - Murphy, Chapters 8.1-8.3

Recall: Probabilistic Learning

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

Building a Probabilistic Classifier

- Define a decision rule
  - Given a test data point $\boldsymbol{x}'$, predict its label $\hat{y}$ using the *posterior distribution* $P(Y = y|X = \boldsymbol{x}')$
  - Common choice: $\hat{y} = \underset{y}{\operatorname{argmax}}\, P(Y = y|X = \boldsymbol{x}')$
- 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 $\boldsymbol{x} = [1, x_1, \dots, x_D]^T \in \mathbb{R}^{D+1}$
- **Assume**

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

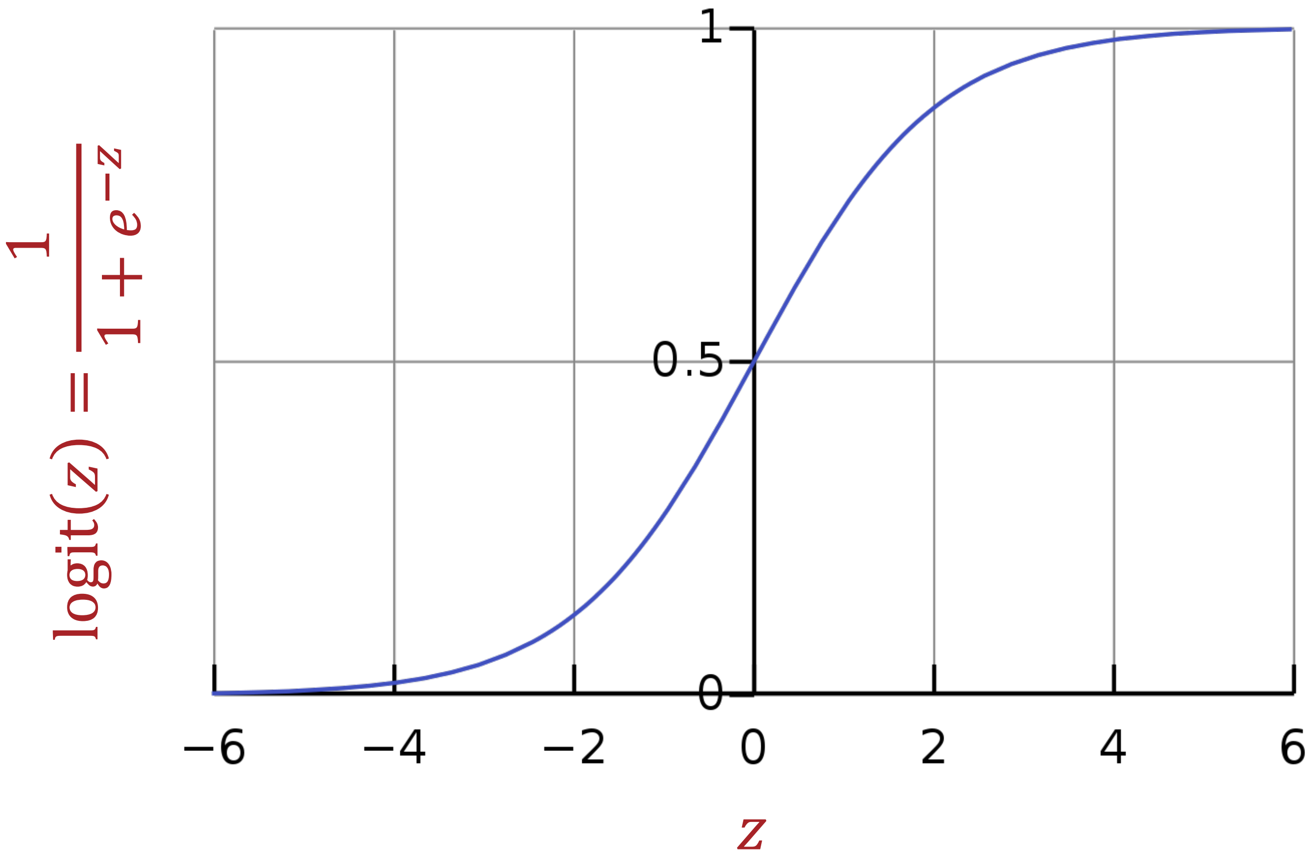
$$= \frac{\exp(\boldsymbol{w}^T\boldsymbol{x})}{\exp(\boldsymbol{w}^T\boldsymbol{x}) + 1}$$

- This implies two useful facts:

1. $P(Y=0|x) = 1 - P(Y=1|x) = 1 - \frac{\exp(w^Tx)}{\exp(w^Tx)+1} = \frac{1}{\exp(w^Tx)+1}$

2. $\frac{P(Y=1|x)}{P(Y=0|x)} = \exp(w^Tx) \rightarrow$ log odds $= \log(\exp(w^Tx)) = w^Tx$ is linear (in x)

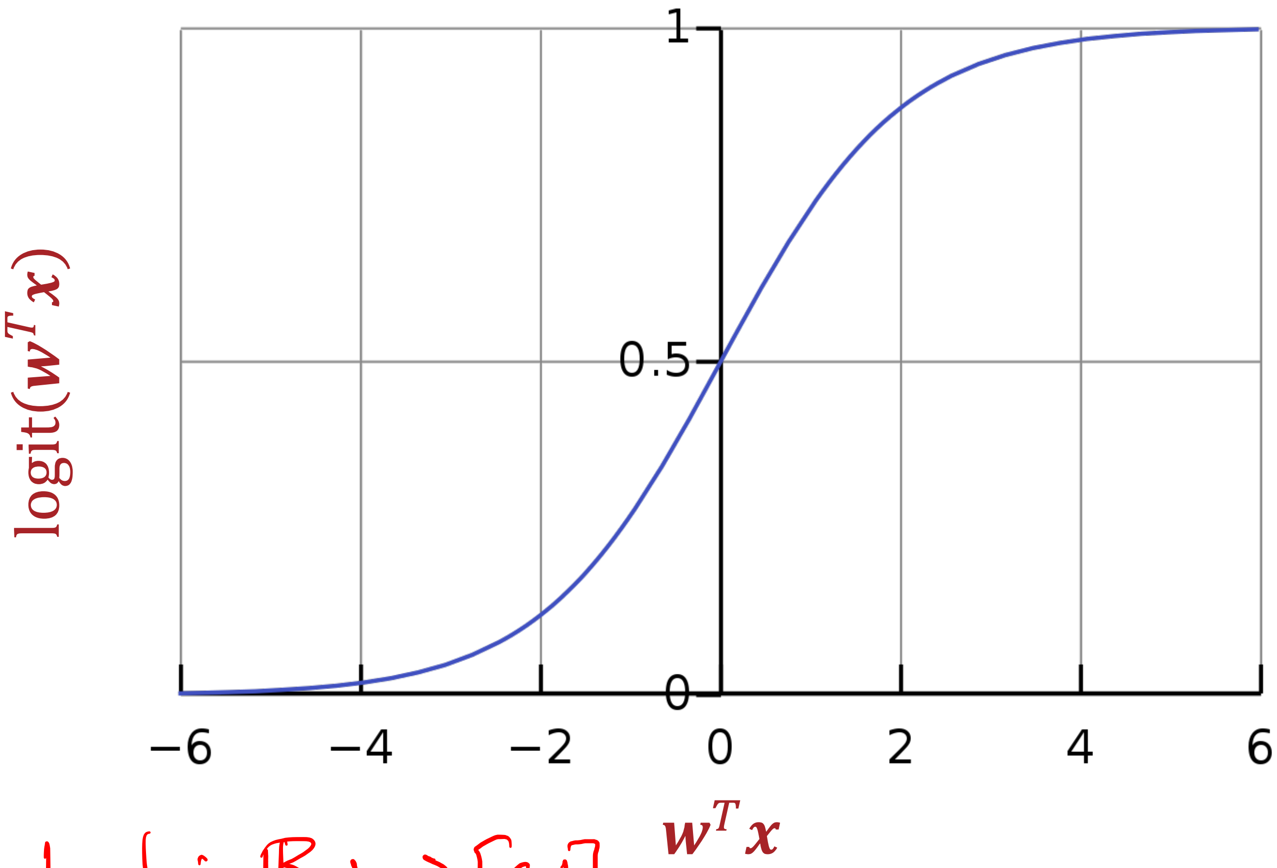
Logistic Function



Source: https://en.wikipedia.org/wiki/Logistic_function#/media/File:Logistic-curve.svg

Why use the Logistic Function?

- gives rise to a linear decision boundary



- logit : $\mathbb{R} \mapsto [0,1]$
- differentiable everywhere ⇒ convenient optimization properties
- centered at 0.5 → $\text{logit}(0) = 0.5$

Source: https://en.wikipedia.org/wiki/Logistic_function#/media/File:Logistic-curve.svg

Logistic Regression Decision Boundary

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

→ why 1/2 ?

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

$$\rightarrow 2 \geq 1 + \exp(-w^T x)$$

$$\rightarrow 1 \geq \exp(-w^T x)$$

$$\rightarrow \ln(1) = 0 \geq -w^T x$$

$$\rightarrow w^T x \geq 0$$

Logistic Regression Decision Boundary

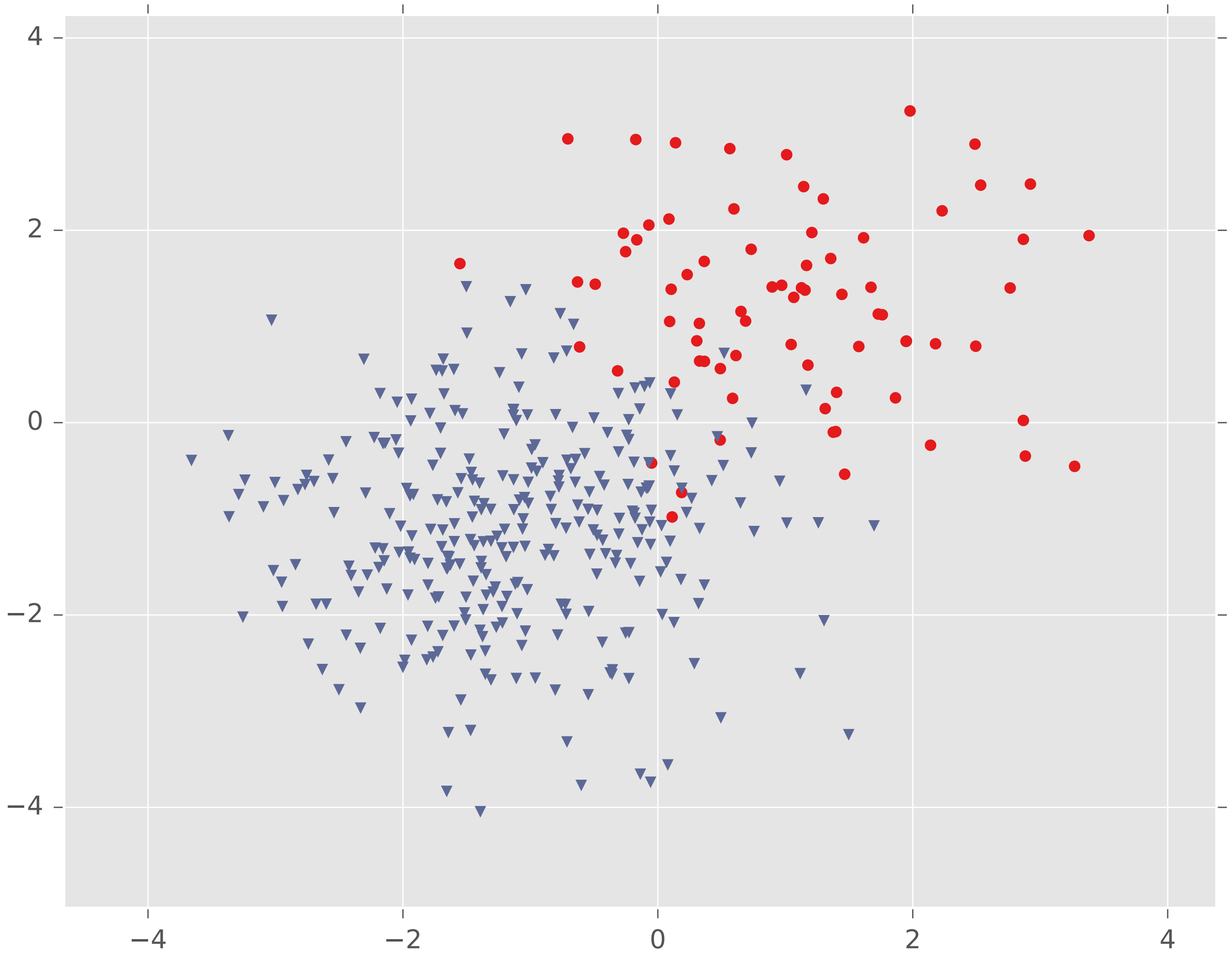


Figure courtesy of Matt Gormley

Logistic Regression Decision Boundary

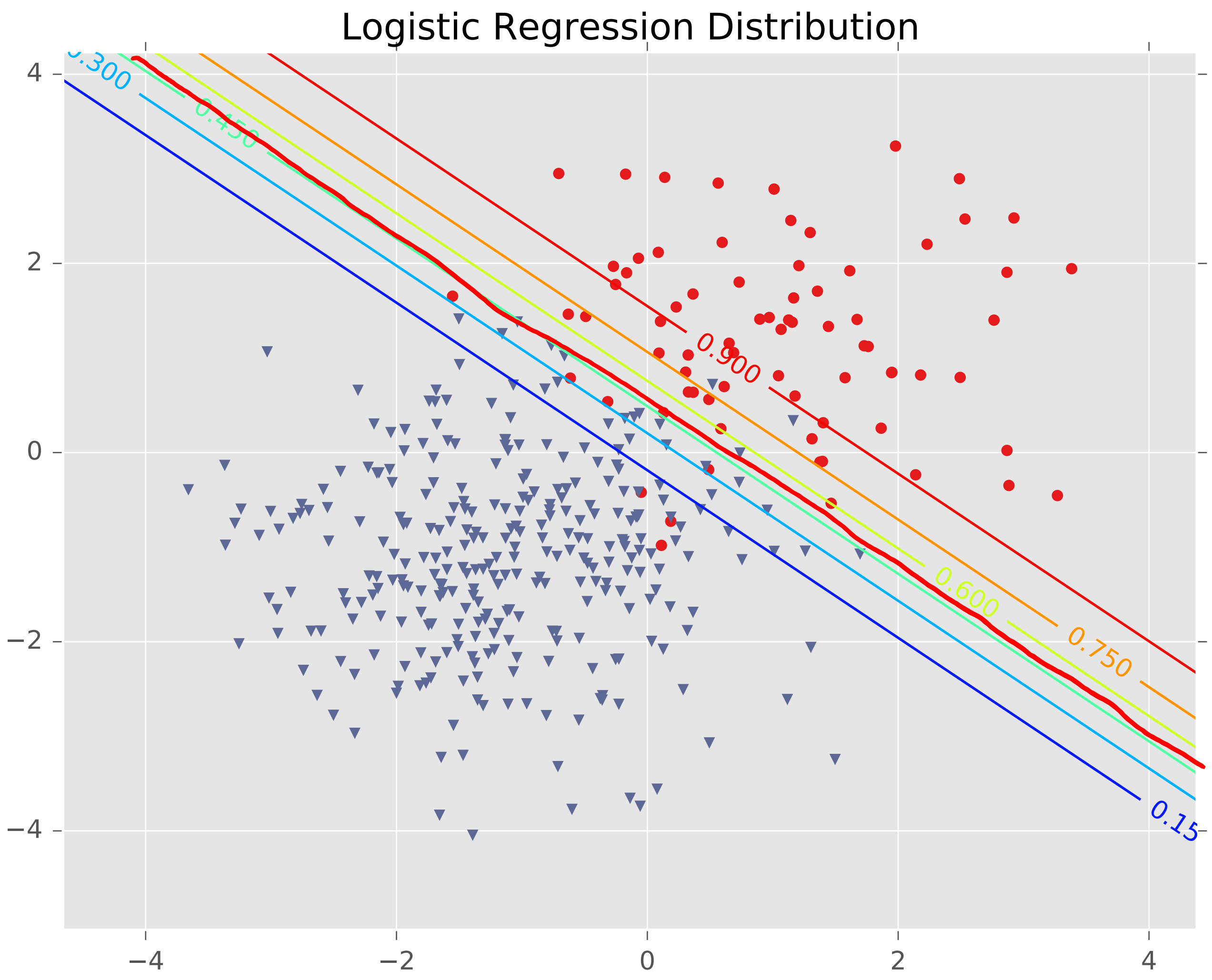


Figure courtesy of Matt Gormley

Logistic Regression Decision Boundary

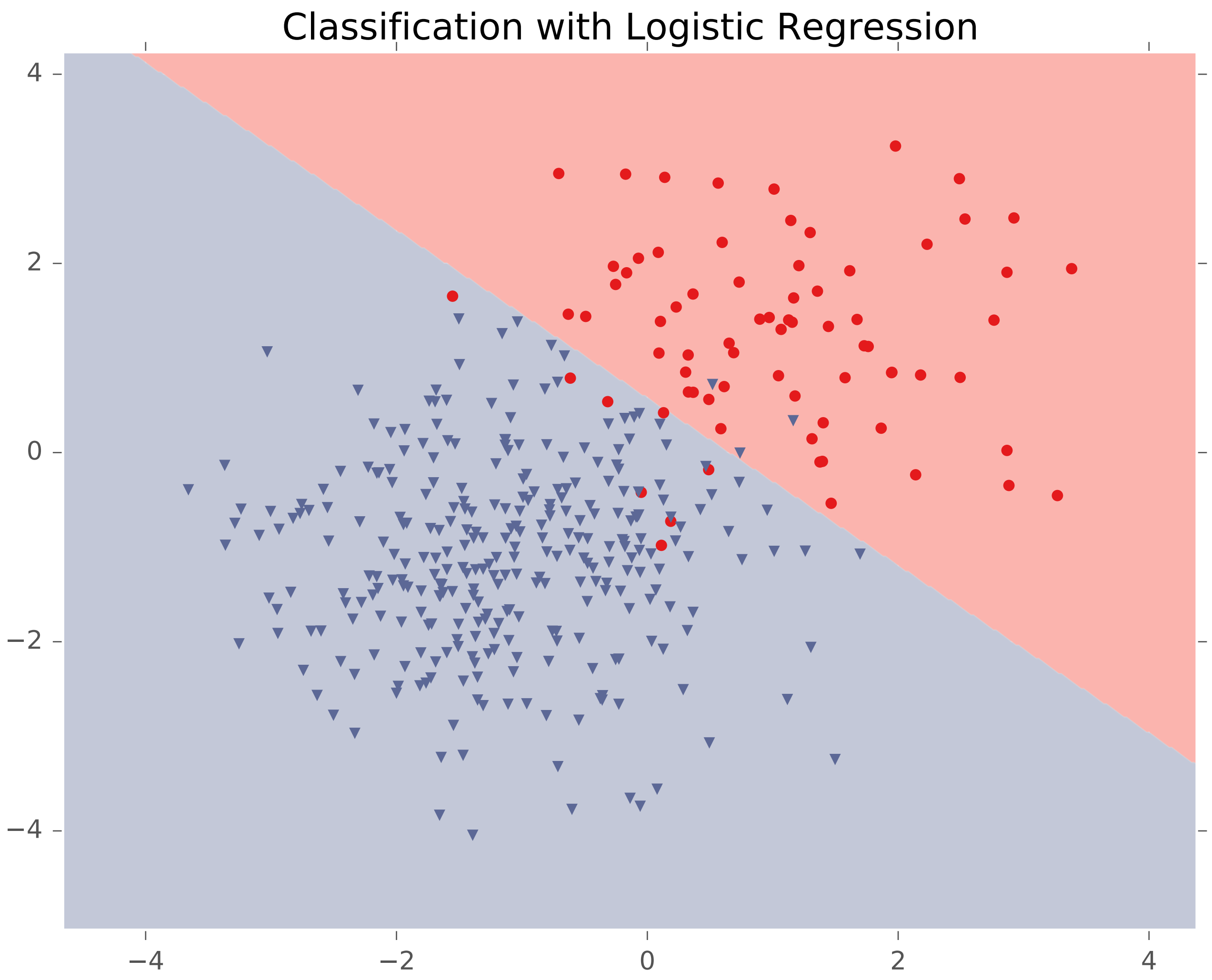


Figure courtesy of Matt Gormley

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 that the data is i.i.d.
  - Assume $P(Y=1|x) = \text{logit}(w^T x)$
  - Parameters: $w = [w_0, w_1, \ldots, w_D]^T$
- Write down an objective function
  - Maximize the log-likelihood

    ⇕
  - Minimize the negative "conditional" log-likelihood
- Optimize the objective w.r.t. the model parameters
  - ???

Setting the Parameters via Minimum Negative (Conditional) (log-)Likelihood Estimation (MCLE)

via MAP? (Stay tuned for regularization, tomorrow!)

Full or joint likelihood $\rightarrow \ell_D^{full}(w) = P(x^{(1)}, y^{(1)}, x^{(2)}, y^{(2)}, \ldots, x^{(N)}, y^{(N)})$

Find $\boldsymbol{w}$ that minimizes

$$\ell_{\mathcal{D}}(\boldsymbol{w}) = -\log P\left(y^{(1)}, \ldots, y^{(N)} \middle| \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}, \boldsymbol{w}\right) = -\log \prod_{n=1}^{N} P\left(y^{(n)} \middle| \boldsymbol{x}^{(n)}, \boldsymbol{w}\right)$$

$x^{(1)}, x^{(2)}$??

$$= -\ln \prod_{n=1}^{N} P(Y=1 \mid x^{(n)}, w)^{y^{(n)}} P(Y=0 \mid x^{(n)}, w)^{1-y^{(n)}}$$

$$= -\sum_{n=1}^{N} y^{(n)} \ln\left(P(Y=1 \mid x^{(n)}, w)\right) + (1-y^{(n)}) \ln\left(P(Y=0 \mid x^{(n)}, w)\right)$$

$$= -\sum_{n=1}^{N} y^{(n)} \ln\left(\frac{P(Y=1 \mid x^{(n)}, w)}{P(Y=0 \mid x^{(n)}, w)}\right) + (1) \ln\left(P(Y=0 \mid x^{(n)}, w)\right)$$

$$= -\sum_{n=1}^{N} y^{(n)} (w^T x^{(n)}) + \ln\left(\frac{1}{1+\exp(w^T x^{(n)})}\right)$$

$$= -\sum_{n=1}^{N} y^{(n)} (w^T x^{(n)}) - \ln\left(1 + \exp(w^T x^{(n)})\right)$$

Minimizing the Negative Conditional (log-)Likelihood

$$\ell_D(w) = -\sum_{n=1}^{N} y^{(n)}\left(w^T x^{(n)}\right) - \ln\left(1+\exp\left(w^T x^{(n)}\right)\right)$$

$$\nabla_w \ell_D(w) = -\sum_{n=1}^{N} \nabla_w \left(y^{(n)}\left(w^T x^{(n)}\right) - \ln\left(1+\exp\left(w^T x^{(n)}\right)\right)\right)$$

$$= -\sum_{n=1}^{N} y^{(n)} x^{(n)} - \frac{1}{1+\exp\left(w^T x^{(n)}\right)} \exp\left(w^T x^{(n)}\right) x^{(n)}$$

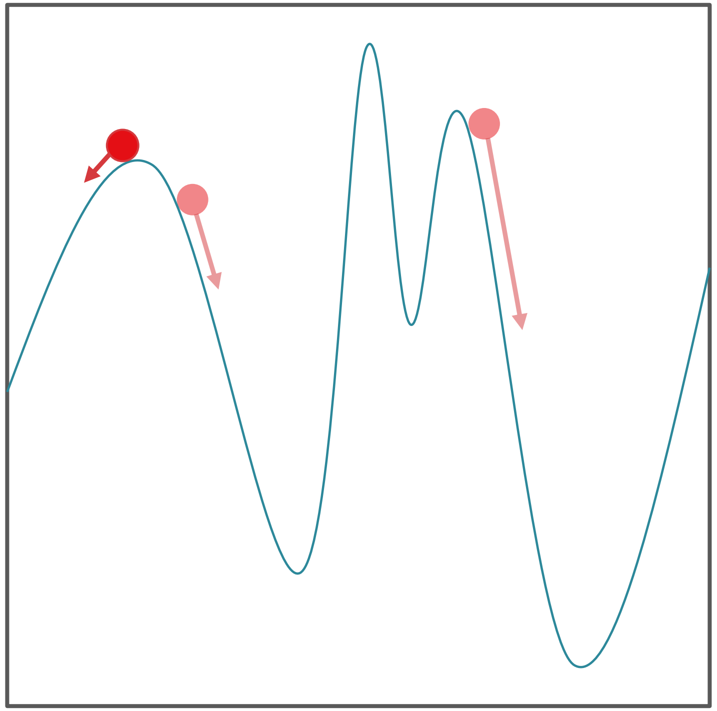
$$= -\sum_{n=1}^{N} x^{(n)}\left(y^{(n)} - \frac{\exp\left(w^T x^{(n)}\right)}{1+\exp\left(w^T x^{(n)}\right)}\right)$$

$$= -\sum_{n=1}^{N} x^{(n)}\left(y^{(n)} - P\left(Y=1 \mid x^{(n)}, w\right)\right)$$

$H_w \ell_D(w)$ is positive semi-definite

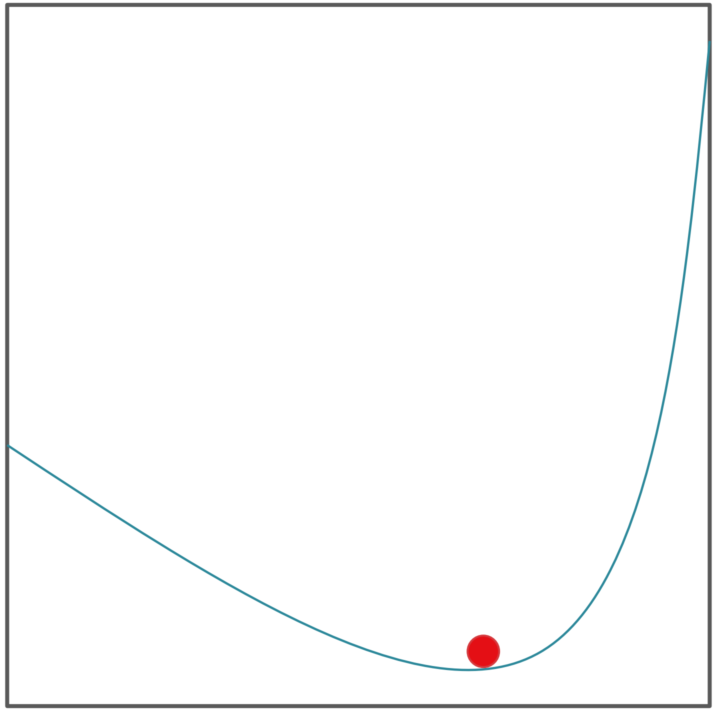
Recall: Gradient Descent

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



Recall: Gradient Descent

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



- Good news: the negative conditional log-likelihood, like the squared error, is also convex!

Gradient Descent

- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}, \eta^{(0)}$

1. Initialize $\boldsymbol{w}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Compute the gradient:

$$O(ND)\left\{ \nabla_{\boldsymbol{w}} \ell_{\mathcal{D}}\left(\boldsymbol{w}^{(t)}\right) = \sum_{n=1}^{N} \boldsymbol{x}^{(n)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(n)}, \boldsymbol{w}^{(t)}\right) - y^{(n)}\right) \right.$$

   b. Update $\boldsymbol{w}$: $\boldsymbol{w}^{(t+1)} \leftarrow \boldsymbol{w}^{(t)} - \eta^{(0)} \nabla_{\boldsymbol{w}} \ell_{\mathcal{D}}\left(\boldsymbol{w}^{(t)}\right)$
   c. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{w}^{(t)}$

Stochastic Gradient Descent

- Input: $\mathcal{D} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^{N}, \eta_{SGD}^{(0)}$

1. Initialize $\boldsymbol{w}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Randomly sample a data point from $\mathcal{D}$, $(\boldsymbol{x}^{(n)}, y^{(n)})$
   b. Compute the pointwise gradient:
   $$\nabla_{\boldsymbol{w}} \ell^{(n)}(\boldsymbol{w}^{(t)}) = \boldsymbol{x}^{(n)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(n)}, \boldsymbol{w}^{(t)}\right) - y^{(n)}\right)$$
   c. Update $\boldsymbol{w}$: $\boldsymbol{w}^{(t+1)} \leftarrow \boldsymbol{w}^{(t)} - \eta_{SGD}^{(0)} \nabla_{\boldsymbol{w}} \ell^{(n)}(\boldsymbol{w}^{(t)})$
   d. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{w}^{(t)}$

Stochastic Gradient Descent

- If the data point is sampled uniformly at random, then the expected value of the pointwise gradient is proportional to the full gradient:

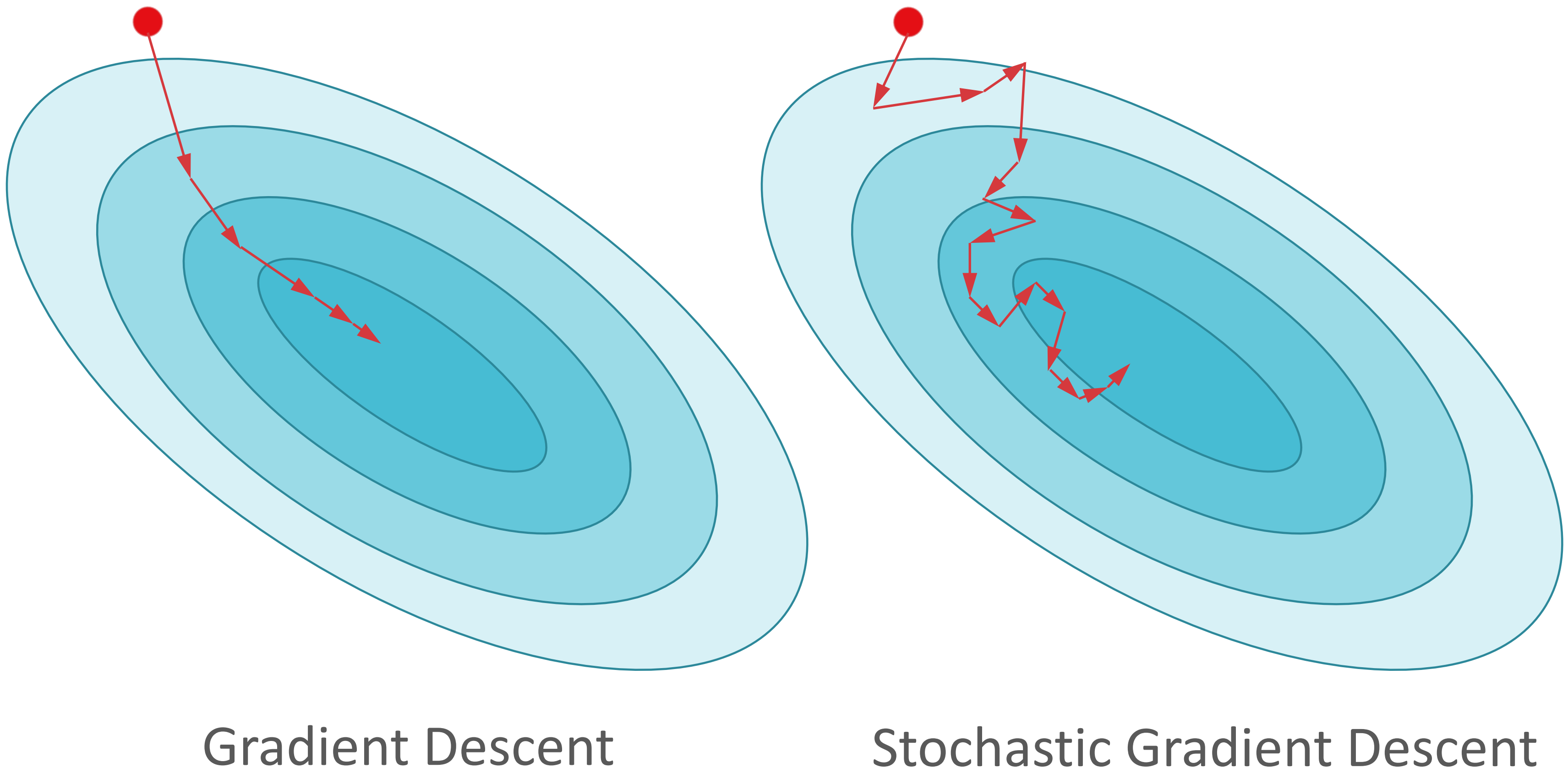
$$E\left[\nabla_{\boldsymbol{w}}\ell_{\boldsymbol{x}^{(n)},y^{(n)}}\left(\boldsymbol{w}^{(t)}\right)\right] = \frac{1}{N}\sum_{n=1}^{N}\nabla_{\boldsymbol{w}}\ell^{(n)}\left(\boldsymbol{w}^{(t)}\right)$$

$$= \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}^{(n)}\left(P\left(Y=1\middle|\boldsymbol{x}^{(n)},\boldsymbol{w}^{(t)}\right)-y^{(n)}\right)$$

$$= \frac{1}{N}\nabla_{\boldsymbol{w}}\ell_{\mathcal{D}}\left(\boldsymbol{w}^{(t)}\right)$$

- In practice, the data set is randomly shuffled then looped through so that each data point is used equally often

Stochastic Gradient Descent vs. Gradient Descent



Mini-batch Stochastic Gradient Descent

- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}, \eta_{MB}^{(0)}, B$

1. Initialize $\boldsymbol{w}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Randomly sample B data points from $\mathcal{D}$:

   $$\mathcal{D}_{batch}\left\{\left(\boldsymbol{x}^{(b)}, y^{(b)}\right)\right\}_{b=1}^{B}$$

   b. Compute the gradient w.r.t. the sampled *batch*:

   $$\nabla_{\boldsymbol{w}} \ell_{\mathcal{D}_{batch}}\left(\boldsymbol{w}^{(t)}\right) = \sum_{b=1}^{B} \boldsymbol{x}^{(b)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(b)}, \boldsymbol{w}\right) - y^{(b)}\right)$$

   c. Update $\boldsymbol{w}$: $\boldsymbol{w}^{(t+1)} \leftarrow \boldsymbol{w}^{(t)} - \eta_{MB}^{(0)} \nabla_{\boldsymbol{w}} \ell_{\mathcal{D}_{batch}}\left(\boldsymbol{w}^{(t)}\right)$
   d. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{w}^{(t)}$

Key Takeaways

$$f(x) = x^2 \qquad \frac{\partial f}{\partial x} = 2x \qquad \frac{\partial^2 f}{\partial x^2} = 2$$

$$H_w \ell_w = \begin{bmatrix} \frac{\partial^2 \ell}{\partial w_1^2} & \frac{\partial \ell}{\partial w_1 \partial w_2} \\ & \vdots \end{bmatrix}$$

- Logistic regression
  - Logistic function induces a linear decision boundary
  - Conditional likelihood maximization

$$= \begin{bmatrix} \nabla \frac{\partial \ell}{\partial w_1} & \nabla \frac{\partial \ell}{\partial w_2} & \cdots \end{bmatrix}$$

- Gradient descent vs. stochastic gradient descent tradeoffs

$$\forall \vec{v}, \quad \vec{v}^T H \vec{v} \geq 0$$