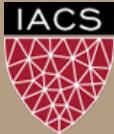


GRADIENT DESCENT

LECTURE 4
SECTION 2
JUNE 8TH



ISTITUTE FOR APPLIED
COMPUTATIONAL SCIENCE
AT HARVARD UNIVERSITY



UNIVERSITY of
RWANDA

MAXIMIZING THE LIKELIHOOD OF LOGISTIC REGRESSION

MAXIMUM LIKELIHOOD ESTIMATOR:

Given a data set $D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})\}$, with $x^{(n)} \in \mathbb{R}^D$ and $y^{(n)} = 0, 1$, the likelihood of D under a Logistic Regression model with linear boundary is:

$$\mathcal{L}(w) = \prod_{n=1}^N \sigma(f_w(x^{(n)}))^{y^{(n)}} (1 - \sigma(f_w(x^{(n)})))^{1-y^{(n)}}$$

where the decision boundary is $f_w(x) = w^T x = 0$.

Fitting the model means finding the maximum likelihood estimator of w :

$$w_{MLE} = \operatorname{argmax}_w \mathcal{L}(w)$$

This is equivalent to maximizing the log-likelihood:

$$\begin{aligned} \operatorname{argmax}_w \mathcal{L}(w) &= \operatorname{argmax}_w l(w) = \operatorname{argmax}_w \log \prod_{n=1}^N \sigma(f_w(x^{(n)}))^{y^{(n)}} (1 - \sigma(f_w(x^{(n)})))^{1-y^{(n)}} \\ &= \operatorname{argmax}_w \sum_{n=1}^N y^{(n)} \log \frac{1}{1+e^{-w^T x^{(n)}}} + (1-y^{(n)}) \log \frac{1}{1+e^{-(1-w^T x^{(n)})}} \\ &= \operatorname{argmax}_w \sum_{n=1}^N -y^{(n)} \log (1+e^{-w^T x^{(n)}}) - (1-y^{(n)}) \log (1+e^{-(1-w^T x^{(n)})}) \end{aligned}$$

We compute the stationary points of the gradient with respect to w :

$$\nabla_w l(w) = \sum_{n=1}^N \left(y^{(n)} - \frac{1}{1+e^{-w^T x^{(n)}}} \right) x^{(n)} = 0$$

But we can't analytically solve this equation!

ITERATIVE OPTIMIZATION

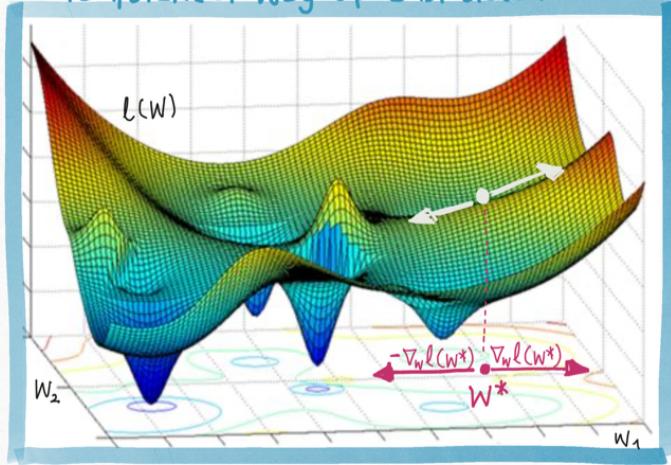
GRADIENT DESCENT:

When we can't analytically solve for the stationary points of the gradient, we can still exploit the information in the gradient.

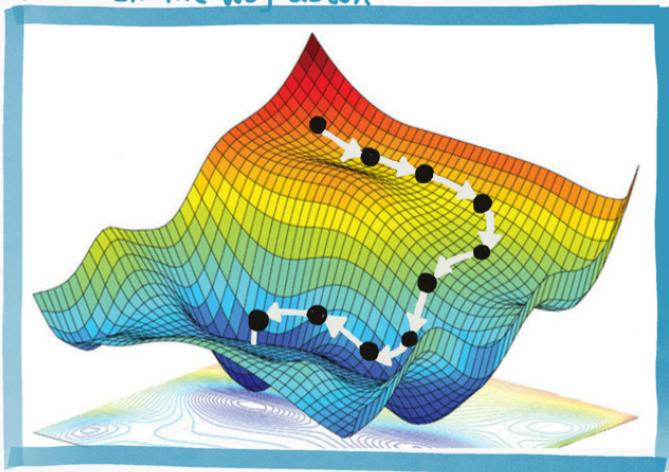
The gradient $\nabla_{\mathbf{w}} l(\mathbf{w}^*)$ at a point \mathbf{w}^* is the direction of the steepest increase.
The negative gradient $-\nabla_{\mathbf{w}} l(\mathbf{w}^*)$ is the direction of the steepest decrease.

By following the negative gradient we can eventually find a **valley** (lowest point). By following the gradient we can eventually find a **peak** (highest point). This method is called **Gradient Descent** (or **Gradient Ascent** if going up).

Gradient and Negative Gradients Point to quickest way up and down



Following the negative gradient step by step leads all the way down



THE GRADIENT DESCENT ALGORITHM:

Our intuition about using the gradient to minimize $\ell(w)$ step-by-step can be formalized by an algorithm:

Gradient Descent Algorithm:

0. start at any point $w^{(0)}$
1. Repeat until stopping condition is met:
 - A. Compute the gradient at the current point $w^{(n)}$:
$$\nabla_w \ell(w^{(n)})$$
 - B. Take a step in the negative gradient direction:

$$w^{(n+1)} = w^{(n)} - \eta \cdot \nabla_w \ell(w^{(n)})$$

The constant η is called the learning rate and is a hyperparameter that we must choose before hand. η controls the size of the step we take in the negative gradient direction.

The stopping condition typically limits the total number of iterations, or terminates when the update to $w^{(n)}$ is sufficiently small, i.e.

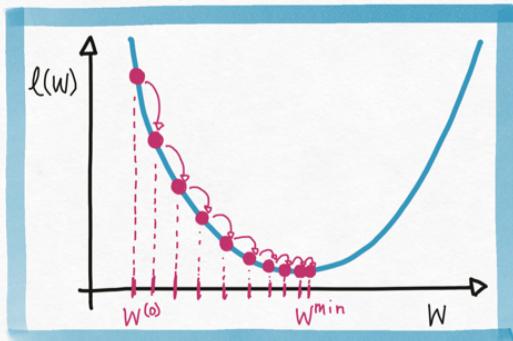
$$\|w^{(n+1)} - w^{(n)}\|_2 < \epsilon$$

REASONING ABOUT GRADIENT DESCENT

DIAGNOSING GRADIENT DESCENT:

Our choice of the learning rate has a significant impact on the performance of gradient descent.

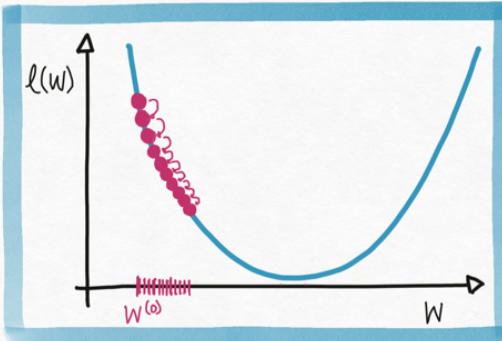
Learning Rate Appropriate



When η is appropriate, the algorithm will eventually find the bottom of a valley, where the gradient is zero.

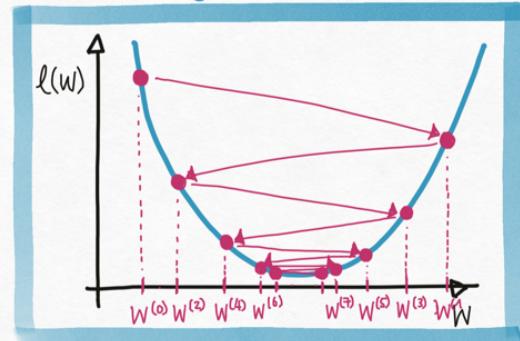
We say the algorithm converges

Learning Rate Too Small



When η is too small, the algorithm makes very little progress. The convergence is too slow.

Learning Rate Too Large



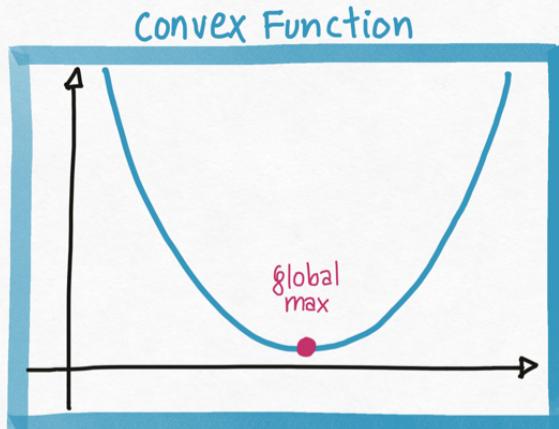
When η is too large, the algorithm may overshoot the optimum value and oscillate.

We may fail to converge.

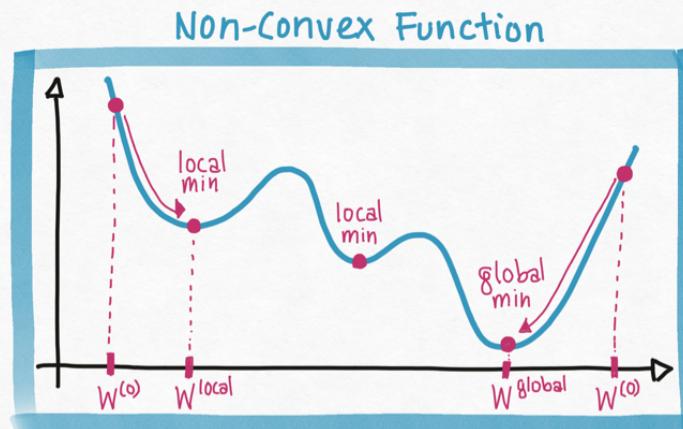
LOCAL VS GLOBAL OPTIMA:

If we choose η correctly, then gradient descent will converge to a stationary point. But will this point be a global minima?

If the function we are optimizing is convex then the stationary point will be a global minimum.



Hessian (2nd Derivative) positive semi-definite everywhere.
Every stationary point of the gradient is a global min.



Hessian (2nd Derivative) not positive semi-definite everywhere.
Some stationary points are local minima but not global optima.

MAXIMIZING LIKELIHOOD OF LOGISTIC REGRESSION

INFERENCE FOR LOGISTIC REGRESSION:

Model

Given a data set $D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})\}$, with $x^{(n)} \in \mathbb{R}^D$ and $y^{(n)} = 0, 1$, the likelihood of D under a Logistic Regression model with linear boundary is:

$$\mathcal{L}(w) = \prod_{n=1}^N \sigma(f_w(x^{(n)}))^{y^{(n)}} (1 - \sigma(f_w(x^{(n)})))^{1-y^{(n)}}$$

where the decision boundary is $f_w(x) = w^T x = 0$.

Inference

Fitting the model means finding the maximum likelihood estimator of w :

$$w_{MLE} = \underset{w}{\operatorname{argmax}} \mathcal{L}(w)$$

This is equivalent to minimizing the negative log-likelihood:

$$w_{MLE} = \underset{w}{\operatorname{argmin}} -\log \mathcal{L}(w) = \underset{w}{\operatorname{argmin}} -l(w)$$

We minimize $-l(w)$ by gradient descent.

Finally, we show that $-l(w)$ is convex, so the stationary point found by gradient descent is a global minimum: w_{MLE} .