

CS 3/5780

Gradient Descent and Beyond

Spring 2026

Logistic Regression

$$P(Y = 1|X = \mathbf{x}) = \frac{1}{1 + \exp(-(\mathbf{w}^\top \mathbf{x} + b))} = \sigma(\mathbf{w}^\top \mathbf{x} + b)$$

- MLE for Logistic regression

$$\hat{\mathbf{w}}_{\text{MLE}} = \arg \min_{\mathbf{w}} \sum_{i=1}^n \log \left(1 + \exp \left(-y_i \mathbf{w}^\top \mathbf{x}_i \right) \right)$$

- Unfortunately no closed form solution.
- This lecture: General strategy for optimization of form

$$\arg \min_{\mathbf{w} \in \mathbb{R}^d} \ell(\mathbf{w}) = \arg \min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \ell_i(\mathbf{w})$$

Hill Climbing Algorithm

Initialize $\mathbf{w}_0 \in \mathbb{R}^d$, $t = 0$, Converged = False

While (Converged == False):

 Pick direction $\mathbf{s}_t \in \mathbb{R}^d$

$\mathbf{w}_{t+1} = \mathbf{w}_t + \mathbf{s}_t$

$t = t + 1$

 If $\|\mathbf{w}_t - \mathbf{w}_{t-1}\| < \delta$ Then Converged = True

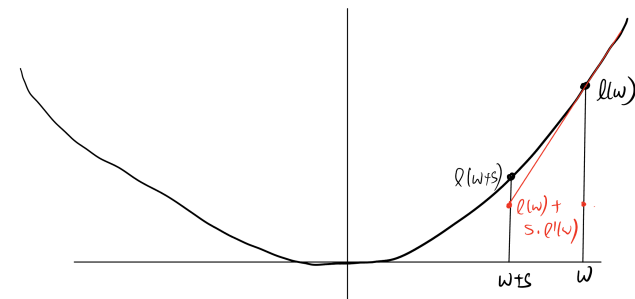
End While

Gradient Descent

- Using first order Taylor with

$$\nabla \ell(\mathbf{w}) = \begin{bmatrix} \frac{\partial}{\partial w[1]} \ell(\mathbf{w}) \\ \frac{\partial}{\partial w[2]} \ell(\mathbf{w}) \\ \vdots \\ \frac{\partial}{\partial w[d]} \ell(\mathbf{w}) \end{bmatrix}$$

$$\ell(\mathbf{w} + \mathbf{s}) = \ell(\mathbf{w}) + \nabla \ell(\mathbf{w})^\top \mathbf{s} + O(\|\mathbf{s}\|^2)$$



Gradient Descent

Initialize $\mathbf{w}_0 \in \mathbb{R}^d$, $t = 0$, Converged = False

While (Converged == False):

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla \ell(\mathbf{w}_t)$$

$$t = t + 1$$

If $\|\mathbf{w}_t - \mathbf{w}_{t-1}\| < \delta$ Then Converged = True

End While

- Consider Quadratic Example:
 - If η is too small we will take for ever to converge
 - If η is too large we can even diverge.

Why Does GD Work

$$\ell(\mathbf{w} + \mathbf{s}) \leq \ell(\mathbf{w}) + \nabla \ell(\mathbf{w})^\top \mathbf{s} + \frac{H}{2} \|\mathbf{s}\|^2$$

- Using Gradient descent step with step size $< 1/2H$

$$\ell(\mathbf{w}_{t+1}) \leq \ell(\mathbf{w}_t) - \frac{\eta}{2} \|\nabla \ell(\mathbf{w}_t)\|^2$$

- Each step is guaranteed progress till we hit a stationary point

Adagrad

- Consider Example: $\ell(\mathbf{w}) = 0.01 * \mathbf{w}[1]^2 + 2 * \mathbf{w}[2]^2$
- Gradient Descent is forced to be slow
- Adagrad: Different step-sizes on different coordinates

Initialize $\mathbf{w}_0 \in \mathbb{R}^d$, $\mathbf{z}_0 = 0$, $t = 0$, Converged = False

While (Converged == False):

$$\mathbf{g}_t = \nabla \ell(\mathbf{w}_t)$$

$$\forall i, \mathbf{z}_{t+1}[i] = \mathbf{z}_t[i] + \mathbf{g}_t[i]^2$$

$$\forall i, \mathbf{w}_{t+1}[i] = \mathbf{w}_t[i] - \eta \frac{\mathbf{g}_t[i]}{\sqrt{\mathbf{z}_t[i] + \epsilon}}$$

$$t = t + 1$$

If $\|\mathbf{w}_t - \mathbf{w}_{t-1}\| < \delta$ Then Converged = True

End While

Newton's Method

- Second order Taylor:

$$\ell(\mathbf{w} + \mathbf{s}) \approx \ell(\mathbf{w}) + \nabla \ell(\mathbf{w})^\top \mathbf{s} + \frac{1}{2} \mathbf{s}^\top \nabla^2 \ell(\mathbf{w}) \mathbf{s} + O(\|\mathbf{s}\|^3)$$

- To minimize approximation pick $\mathbf{s} = -(\nabla^2 \ell(\mathbf{w}))^{-1} \nabla \ell(\mathbf{w})$

- Update $\mathbf{w}_{t+1} = \mathbf{w}_t - (\nabla^2 \ell(\mathbf{w}_t))^{-1} \nabla \ell(\mathbf{w}_t)$

- Needs $O(d^3)$ for Hessian matrix inversion

- Needs warm start from gradient descent, can be unstable