# Logistic Regression and Convex Optimization

Anthony Platanios

# Logistic Regression

Binary:

$$P(Y = 1 \mid X = \boldsymbol{x}) = \frac{1}{1 + e^{\boldsymbol{w}^\top \boldsymbol{x}}}$$

Multiclass:

$$P(Y = y_k \mid X = \boldsymbol{x}) = \frac{e^{\boldsymbol{w}_k^\top \boldsymbol{x}}}{1 + \sum_{k'=1}^{K-1} e^{\boldsymbol{w}_{k'}^\top \boldsymbol{x}}}$$

# Logistic Regression Decision Boundary

Linear classification boundary, but why?

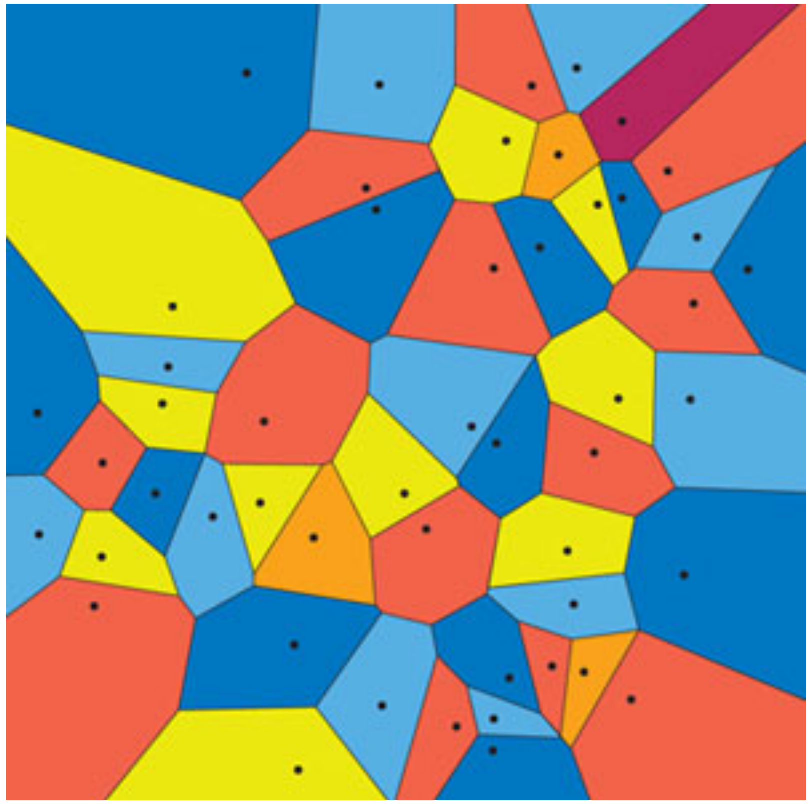
$$P(Y=1 \mid X=\boldsymbol{x}) = P(Y=0 \mid X=\boldsymbol{x}) \Rightarrow$$

$$\frac{1}{1+e^{\boldsymbol{w}^\top \boldsymbol{x}}} = \frac{e^{\boldsymbol{w}^\top \boldsymbol{x}}}{1+e^{\boldsymbol{w}^\top \boldsymbol{x}}} \Rightarrow$$

$$\boldsymbol{w}^\top \boldsymbol{x} = 0$$

**Hyperplane**

Multiclass: **Voronoi**



# Training Logistic Regression

We use maximum condition likelihood estimation (MCLE):

$$\boldsymbol{w} = \arg\max_{\boldsymbol{w}} \prod_{i=1}^{n} P(Y = y^i \mid X = \boldsymbol{x}^i, \boldsymbol{w}) \Rightarrow$$

$$\boldsymbol{w} = \arg\max_{\boldsymbol{w}} \sum_{i=1}^{n} y^i(\boldsymbol{w}^\top \boldsymbol{x}^i) - \log\left(1 + e^{\boldsymbol{w}^\top \boldsymbol{x}^i}\right)$$

But how do we solve this?

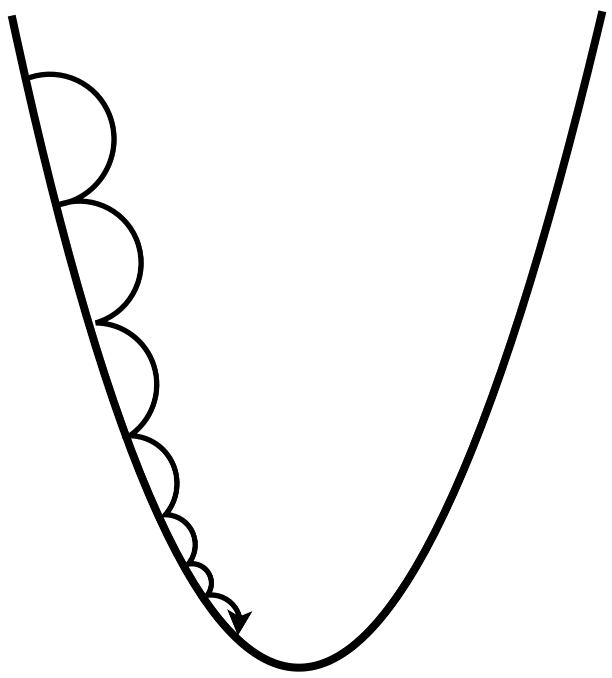
**Convex Optimization**

# Gradient Descent

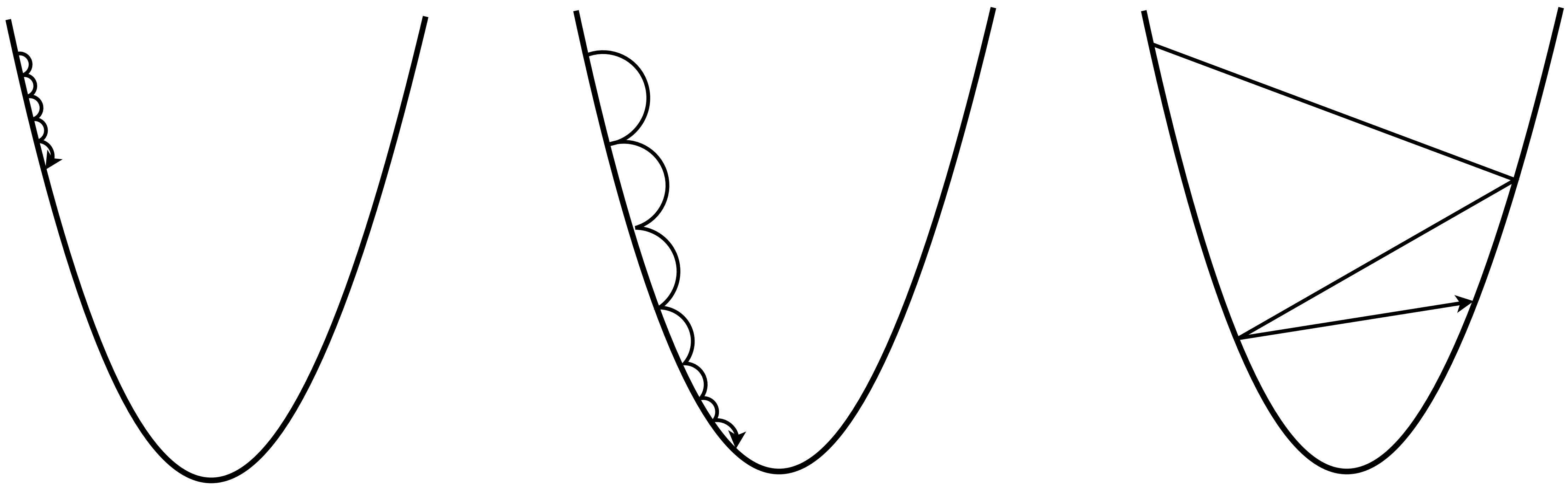
Problem: $$\arg\min_{\boldsymbol{w}} f(\boldsymbol{w})$$

Iterative Solution: $$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \frac{\partial f(\boldsymbol{w})}{\partial \boldsymbol{w}}$$

**Step Size**



# Step Size Selection



Too Small | **Just Right** | Too Big

# Step Size Selection

Exact Line Search: $$\eta^* = \arg\min_{\eta \geq 0} f\left(\boldsymbol{w} - \eta \frac{\partial f(\boldsymbol{w})}{\partial \boldsymbol{w}}\right)$$

Most often we cannot obtain a closed-form solution!

Backtracking Line Search: Choose $0 < \beta < 1$, $0 < \alpha < 0.5$ and $\eta_0 \geq 0$ and while:

$$f\left(\boldsymbol{w} - \eta_t \frac{\partial f(\boldsymbol{w})}{\partial \boldsymbol{w}}\right) > f(\boldsymbol{w}) - \alpha \eta_t \left\| \frac{\partial f(\boldsymbol{w})}{\partial \boldsymbol{w}} \right\|_2^2$$

set: $$\eta_{t+1} = \beta \eta_t$$

We start with a big step size and keep decreasing it until the update generates a sufficient decrease for our function. We want out next iterate to beat the criterion value of a linear approximation of our function at the current point. The above inequality is called the **Armijo rule** and it is often used in combination with a **curvature condition**. Check the [wikipedia page on Wolfe conditions]() for more informations.

# Other Convex Optimization Methods

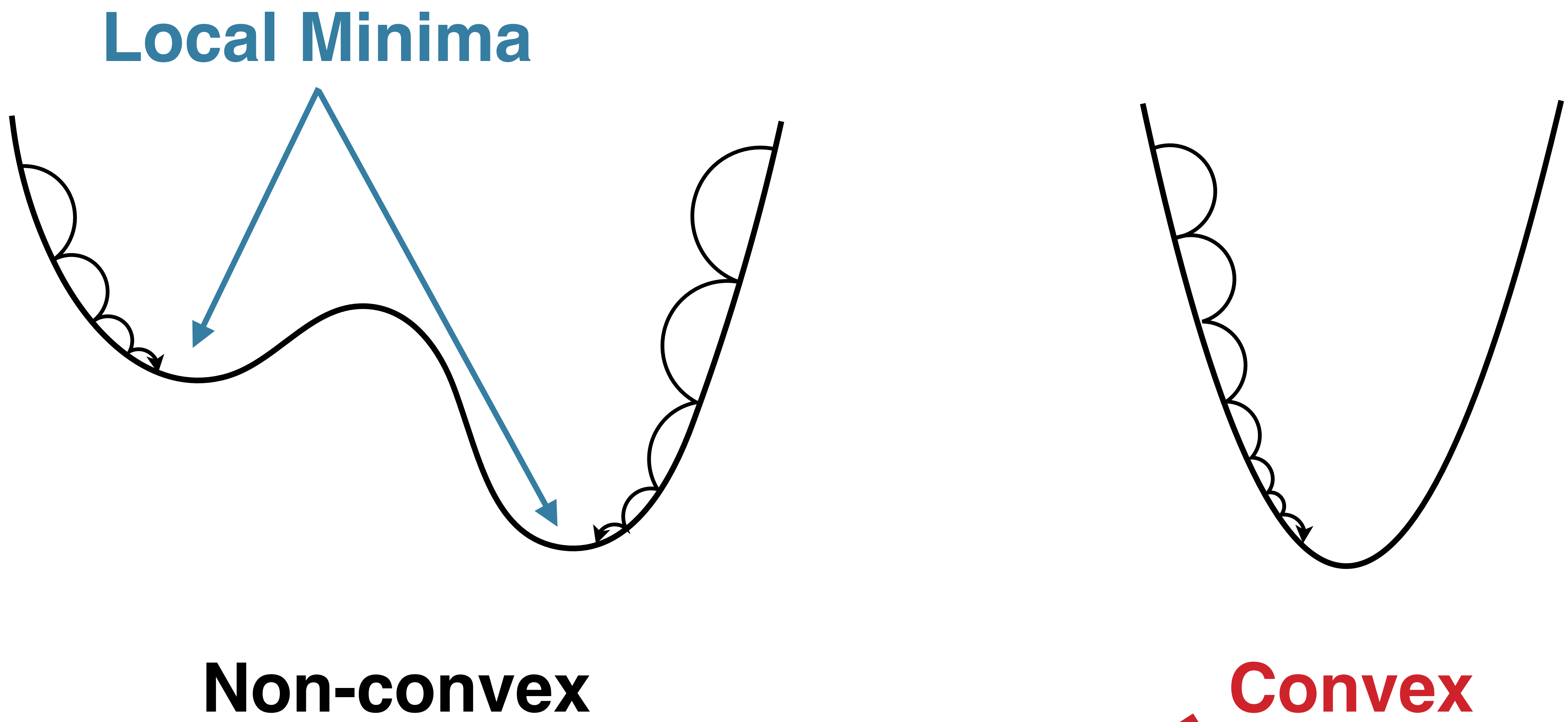
There are other convex optimization methods that use even second derivative information, such as **Newton's method**.

Use the **inverse Hessian** as the step size (makes use of **curvature** information)

If the Hessian is too expensive to compute, people might also use **quasi-Newton** methods such as the well known **LBFGS** algorithm.

But, what is **convexity?**

# Convexity




$$\forall \boldsymbol{w}, \boldsymbol{u} \in \mathrm{dom}(f) \text{ and } \forall \theta \in [0,1]:$$

$$f(\theta \boldsymbol{w} + (1-\theta)\boldsymbol{u}) \le \theta f(\boldsymbol{w}) + (1-\theta) f(\boldsymbol{u})$$

# Operations that Preserve Convexity

<u>Nonnegative linear combinations:</u> If $f_1, \ldots, f_m$ are convex, then $\alpha_1 f_1 + \cdots + \alpha_m f_m$ is convex for any $\alpha_1, \ldots, \alpha_m \geq 0$ .

<u>Affine compositions:</u> If $f$ is convex, then $g(\boldsymbol{w}) = f(\mathbf{A}\boldsymbol{w} + \boldsymbol{b})$ is also convex.

<u>Pointwise maximum:</u> If $f_1, \ldots, f_m$ are convex, then then $\max\{f_1, \ldots, f_m\}$ is also convex.

# Back to Logistic Regression

Sum of **affine functions** and **convex functions**.

$$\boldsymbol{w} = \arg\max_{\boldsymbol{w}} L(\boldsymbol{w}) = \arg\max_{\boldsymbol{w}} \sum_{i=1}^{n} y^i(\boldsymbol{w}^\top \boldsymbol{x}^i) - \log\left(1 + e^{\boldsymbol{w}^\top \boldsymbol{x}^i}\right)$$

$$\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} = \sum_{i=1}^{n} \boldsymbol{x}^i \left( y^i - \frac{e^{\boldsymbol{w}^\top \boldsymbol{x}^i}}{1 + e^{\boldsymbol{w}^\top \boldsymbol{x}^i}} \right)$$

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta \frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}}$$

# Logistic Regression with a Prior

$$\|\boldsymbol{w}\|_2^2 \triangleq \sum_j w_j^2$$

$$\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$$

$$L(\boldsymbol{w}) = \sum_{i=1}^{n} y^i (\boldsymbol{w}^\top \boldsymbol{x}^i) - \log\left(1 + e^{\boldsymbol{w}^\top \boldsymbol{x}^i}\right) - \frac{1}{2\sigma^2} \|\boldsymbol{w}\|_2^2$$

$$\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} = \sum_{i=1}^{n} \boldsymbol{x}^i \left( y^i - \frac{e^{\boldsymbol{w}^\top \boldsymbol{x}^i}}{1 + e^{\boldsymbol{w}^\top \boldsymbol{x}^i}} \right) - \frac{1}{\sigma^2} \boldsymbol{w}$$

More generally we call this **L2 regularization**:

$$\arg\min_{\boldsymbol{w}} f(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_2^2$$

# Regularization in Optimization

More generally we have **Lp regularization**:

$$\arg\min_{\boldsymbol{w}} f(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_p^p$$

**L1** is a special case, frequently used in practice to induce **sparsity** in the solution:

$$\arg\min_{\boldsymbol{w}} f(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1$$

# Stochastic Gradient Descent

We can randomly sample terms of that sum and get an estimate of the gradient in order to speed things up

$$\boldsymbol{w} = \arg\max_{\boldsymbol{w}} L(\boldsymbol{w}) = \arg\max_{\boldsymbol{w}} \sum_{i=1}^{n} y^i(\boldsymbol{w}^\top \boldsymbol{x}^i) - \log\left(1 + e^{\boldsymbol{w}^\top \boldsymbol{x}^i}\right)$$

$$\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} = \sum_{i=1}^{n} \boldsymbol{x}^i \left( y^i - \frac{e^{\boldsymbol{w}^\top \boldsymbol{x}^i}}{1 + e^{\boldsymbol{w}^\top \boldsymbol{x}^i}} \right)$$

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta \frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}}$$