

10-701: Introduction to Machine Learning

# Lecture 5 – MLE & MAP

Hoda Heidari

\* Slides adopted from F24 offering of 10701 by Henry Chai.

# Recipe for Linear Regression

1. Define a model and model parameters
  1. Assume  $y = \mathbf{w}^T \mathbf{x}$
  2. Parameters:  $\mathbf{w} = [w_0, w_1, \dots, w_D]$

2. Write down an objective function
  1. Minimize the mean squared error

$$\ell_{\mathcal{D}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N (\mathbf{w}^T \mathbf{x}^{(n)} - y^{(n)})^2$$

3. Optimize the objective w.r.t. the model parameters
  1. Solve in *closed form*: take partial derivatives, set to 0 and solve

# Minimizing the Squared Error

$$\ell_{\mathcal{D}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N (\mathbf{w}^T \mathbf{x}^{(n)} - y^{(n)})^2 = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}^{(n)T} \mathbf{w} - y^{(n)})^2$$

$$= \frac{1}{N} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 \quad \text{where } \|\mathbf{z}\|_2 = \sqrt{\sum_{d=1}^D z_d^2} = \sqrt{\mathbf{z}^T \mathbf{z}}$$

$$= \frac{1}{N} (\mathbf{X}\mathbf{w} - \mathbf{y})^T (\mathbf{X}\mathbf{w} - \mathbf{y})$$

$$= \frac{1}{N} (\mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} - 2\mathbf{w}^T \mathbf{X}^T \mathbf{y} + \mathbf{y}^T \mathbf{y})$$

$$\nabla_{\mathbf{w}} \ell_{\mathcal{D}}(\hat{\mathbf{w}}) = \frac{1}{N} (2\mathbf{X}^T \mathbf{X} \hat{\mathbf{w}} - 2\mathbf{X}^T \mathbf{y}) = 0$$

$$\rightarrow \mathbf{X}^T \mathbf{X} \hat{\mathbf{w}} = \mathbf{X}^T \mathbf{y}$$

$$\rightarrow \hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

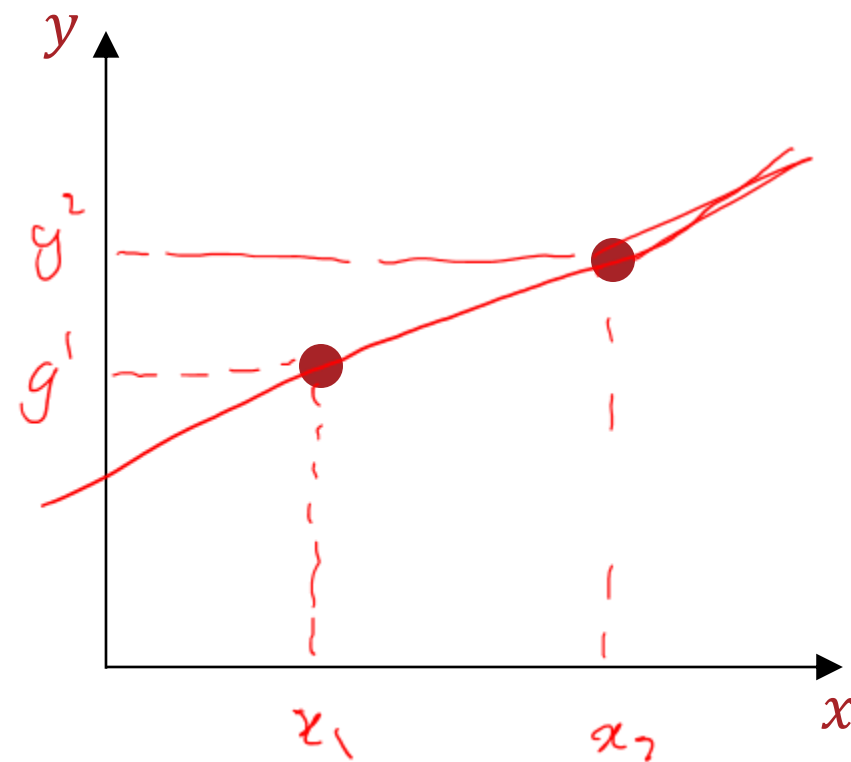
# Closed Form Solution

$$\hat{\mathbf{w}} = \underbrace{(\mathbf{X}^T \mathbf{X})^{-1}} \mathbf{X}^T \mathbf{y}$$

1. Is  $\mathbf{X}^T \mathbf{X}$  invertible?
2. If so, how computationally expensive is inverting  $\mathbf{X}^T \mathbf{X}$ ?

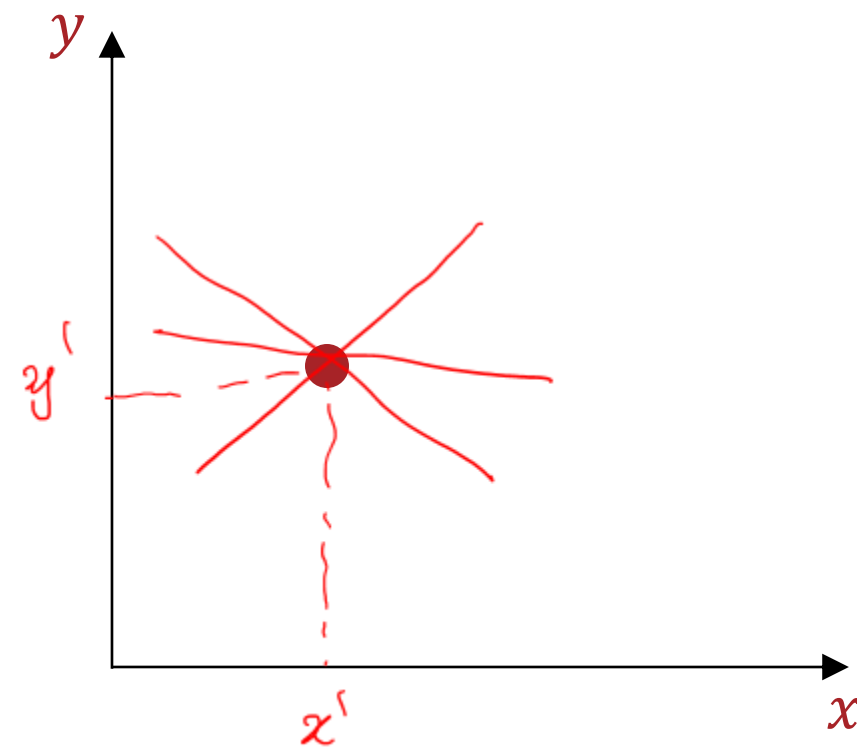
# Linear Regression: Uniqueness

- Consider a 1D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of weights  $\mathbf{w}$ ) are there for the given dataset?



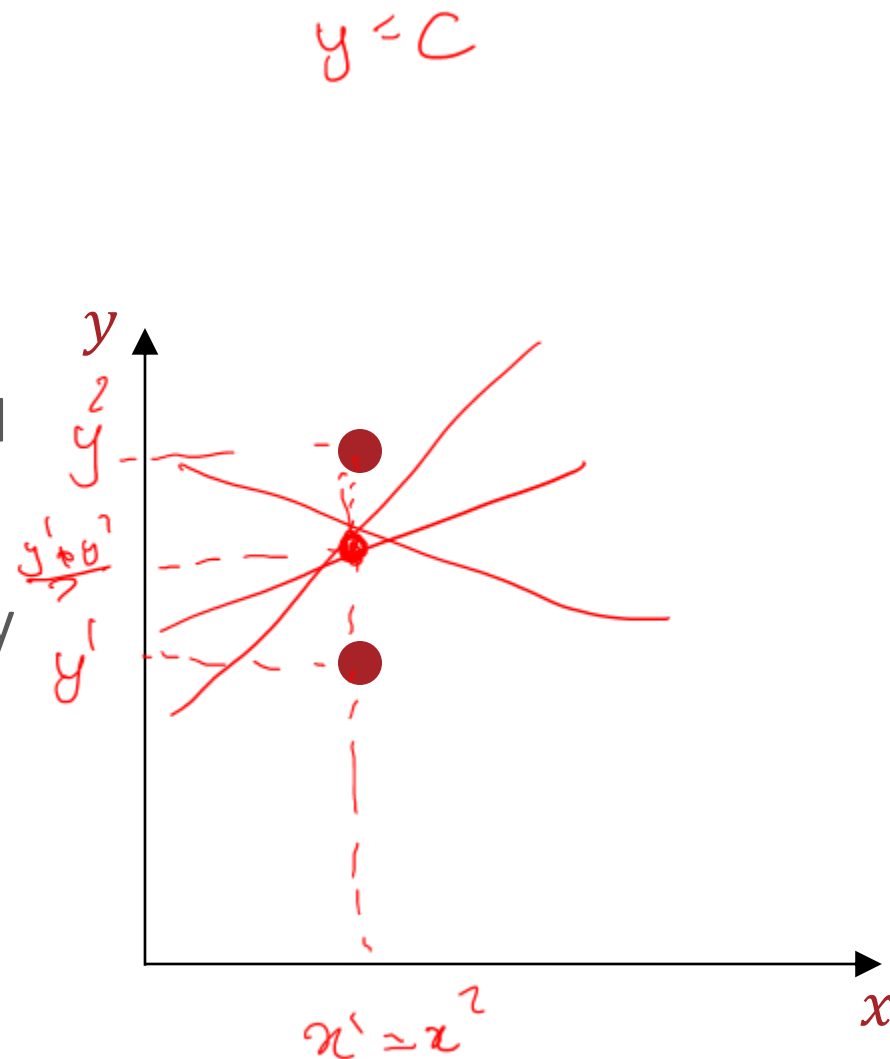
# Linear Regression: Uniqueness

- Consider a 1D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of weights  $\mathbf{w}$ ) are there for the given dataset?



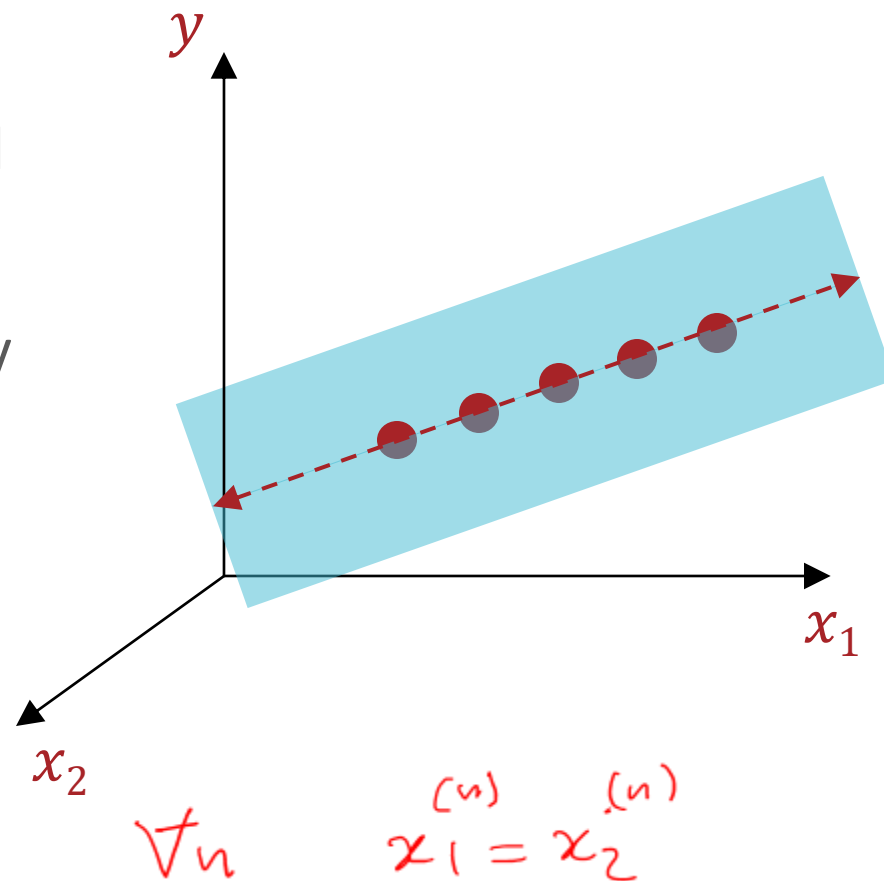
# Linear Regression: Uniqueness

- Consider a 1D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of weights  $\mathbf{w}$ ) are there for the given dataset?



# Linear Regression: Uniqueness

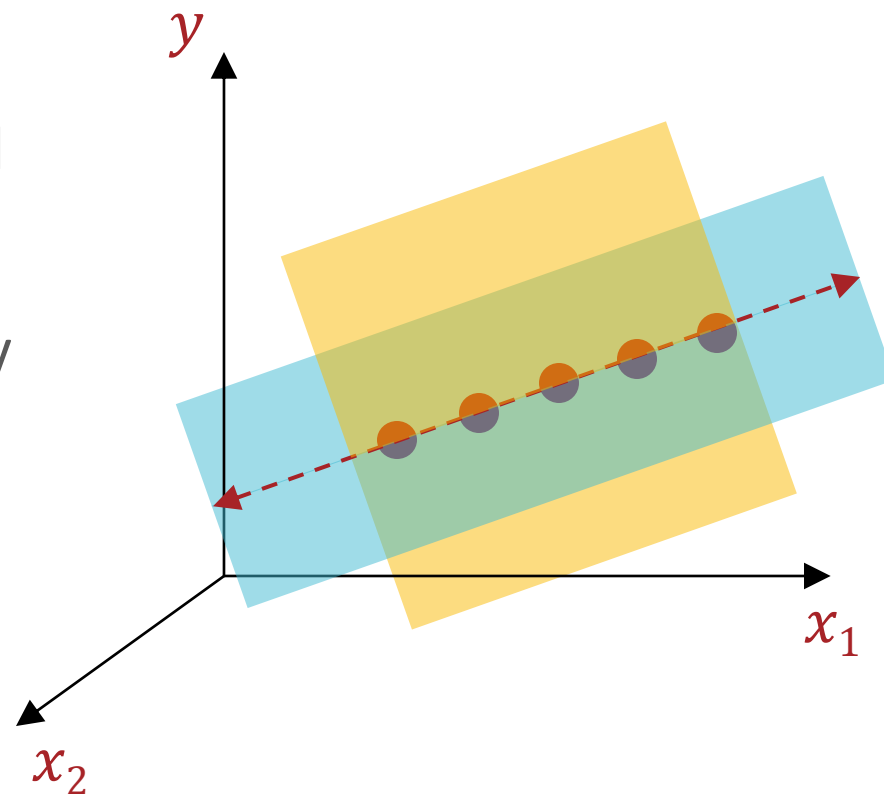
- Consider a 2D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of weights  $\mathbf{w}$ ) are there for the given dataset?





# Linear Regression: Uniqueness

- Consider a 2D linear regression model trained to minimize the mean squared error: how many optimal solutions (i.e., sets of weights  $\mathbf{w}$ ) are there for the given dataset?



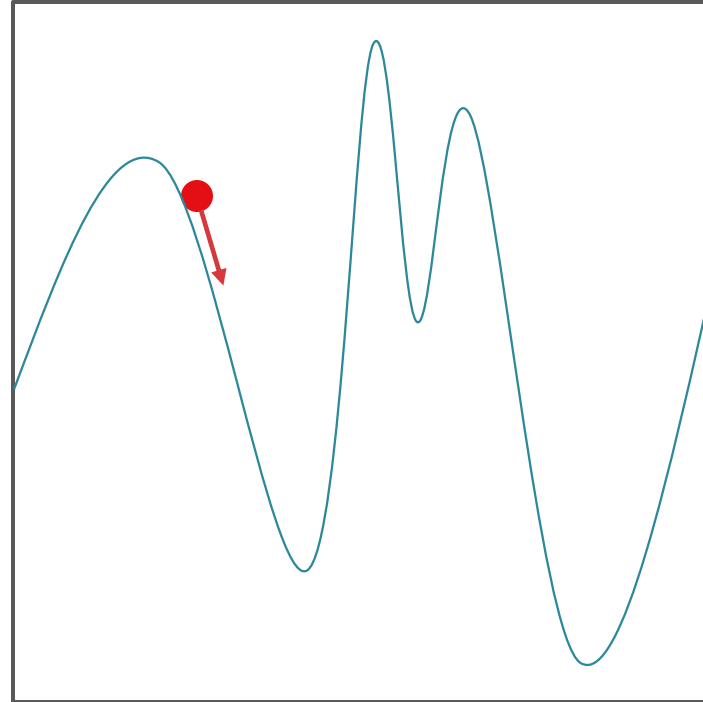
# Closed Form Solution

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

1. Is  $\mathbf{X}^T \mathbf{X}$  invertible?
  - When  $N \gg D + 1$ ,  $\mathbf{X}^T \mathbf{X}$  is (almost always) full rank and therefore, invertible
  - If  $\mathbf{X}^T \mathbf{X}$  is not invertible (occurs when one of the features is a linear combination of the others) then there are infinitely many solutions.
2. If so, how computationally expensive is inverting  $\mathbf{X}^T \mathbf{X}$ ?
  - $\mathbf{X}^T \mathbf{X} \in \mathbb{R}^{D+1 \times D+1}$  so inverting  $\mathbf{X}^T \mathbf{X}$  takes  $O(D^3)$  time...
    - Computing  $\mathbf{X}^T \mathbf{X}$  takes  $O(ND^2)$  time
  - What alternative optimization method can we use to minimize the mean squared error?

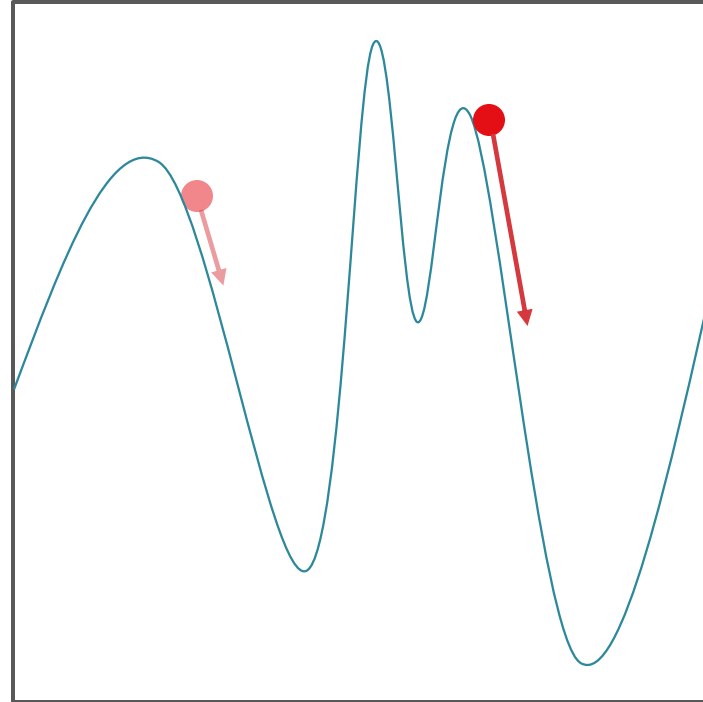
# Gradient Descent: Intuition

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



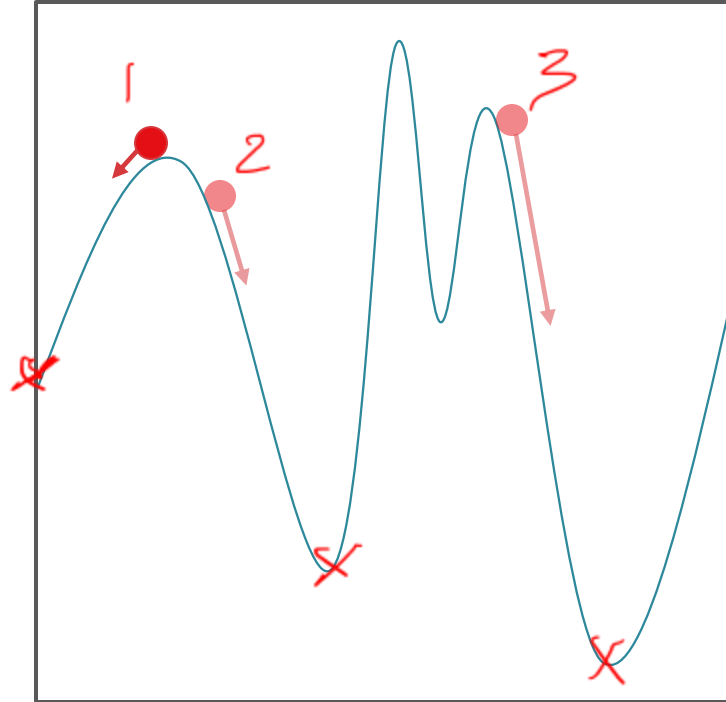
# Gradient Descent: Intuition

- 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

- Suppose the current weight vector is  $\mathbf{w}^{(t)}$
- Move some distance,  $\eta$ , in the “most downhill” direction,  $\hat{\mathbf{v}}$ :

$$\rightarrow \mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \eta \hat{\mathbf{v}}$$

# Gradient Descent: Step Direction

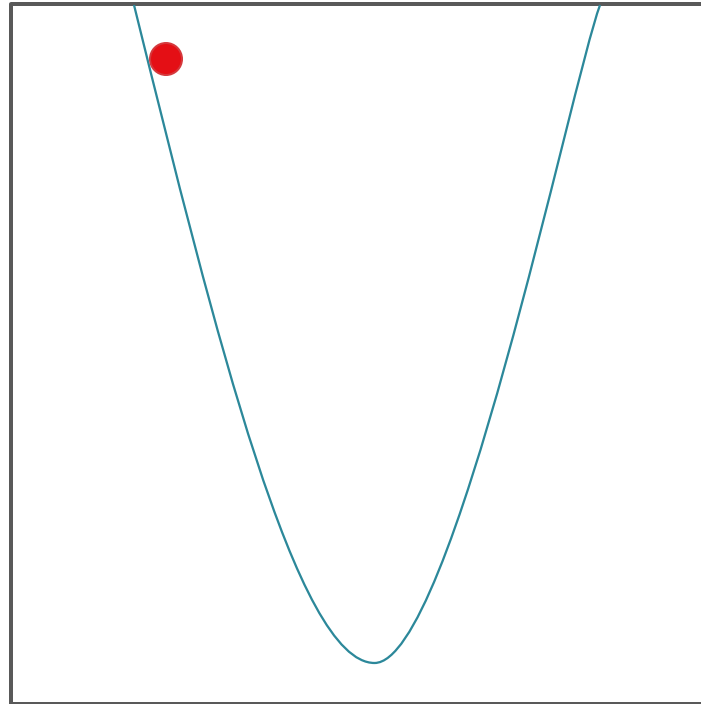
- Suppose the current weight vector is  $\mathbf{w}^{(t)}$
- Move some distance,  $\eta$ , in the “most downhill” direction,  $\hat{\mathbf{v}}$ :

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \eta \hat{\mathbf{v}}$$

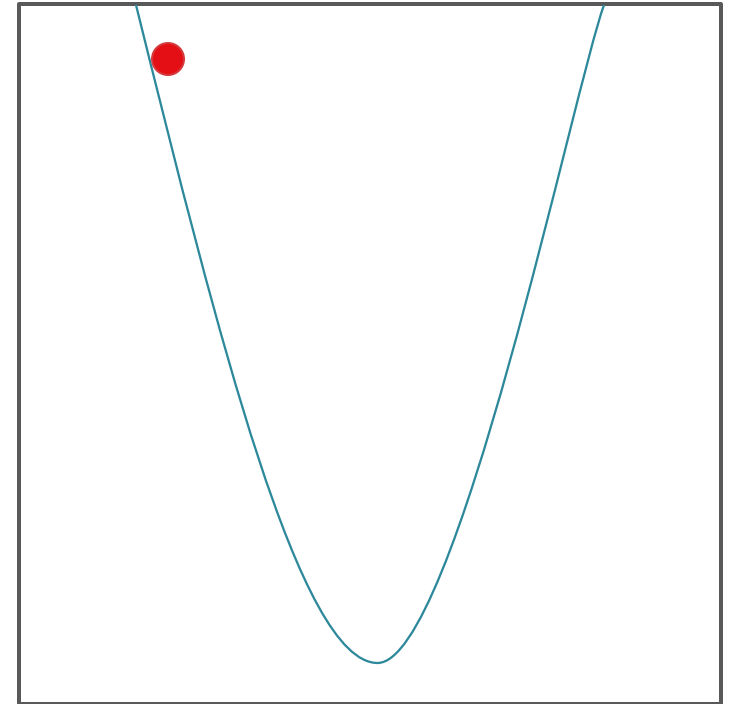
- The gradient points in the direction of steepest *increase* ...
- ... so  $\hat{\mathbf{v}}$  should point in the opposite direction:

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

# Gradient Descent: Step Size



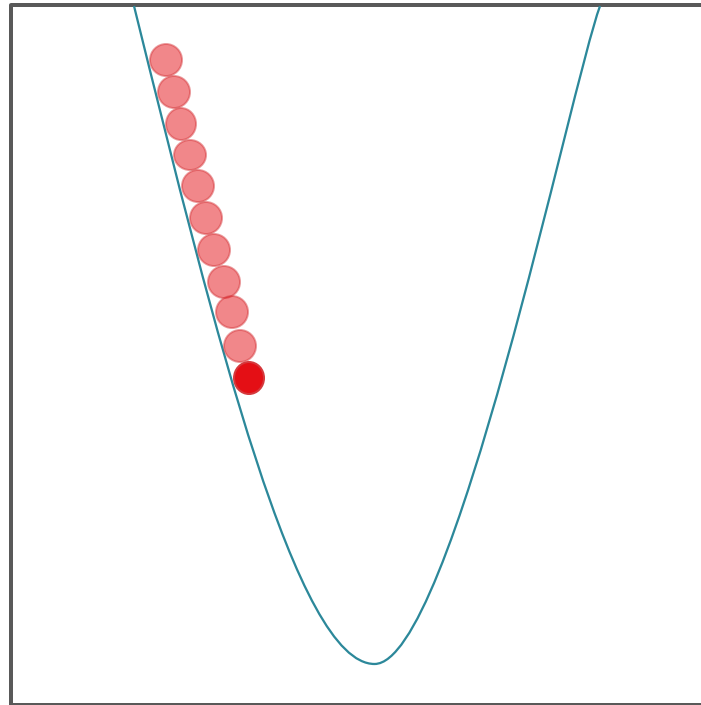
Small  $\eta$   
—  
step size



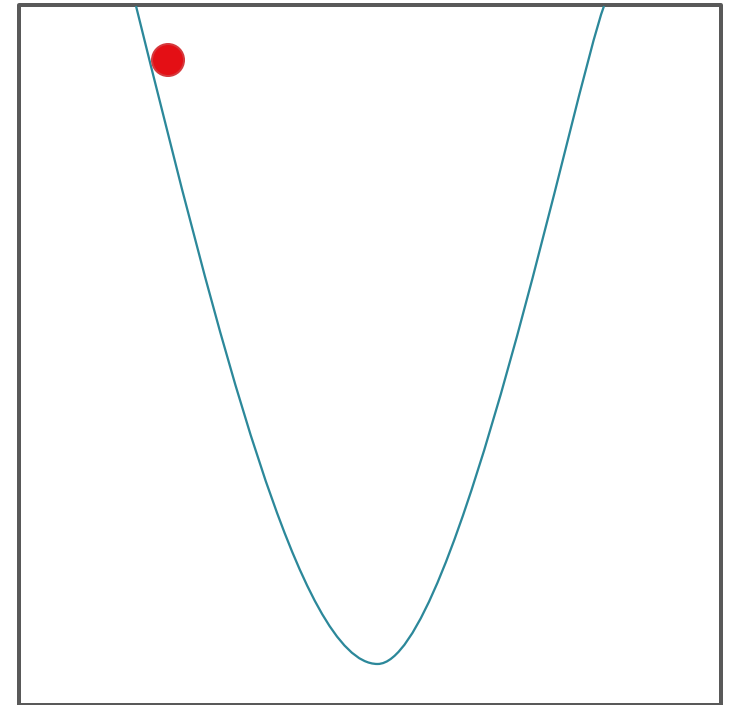
Large  $\eta$



# Gradient Descent: Step Size

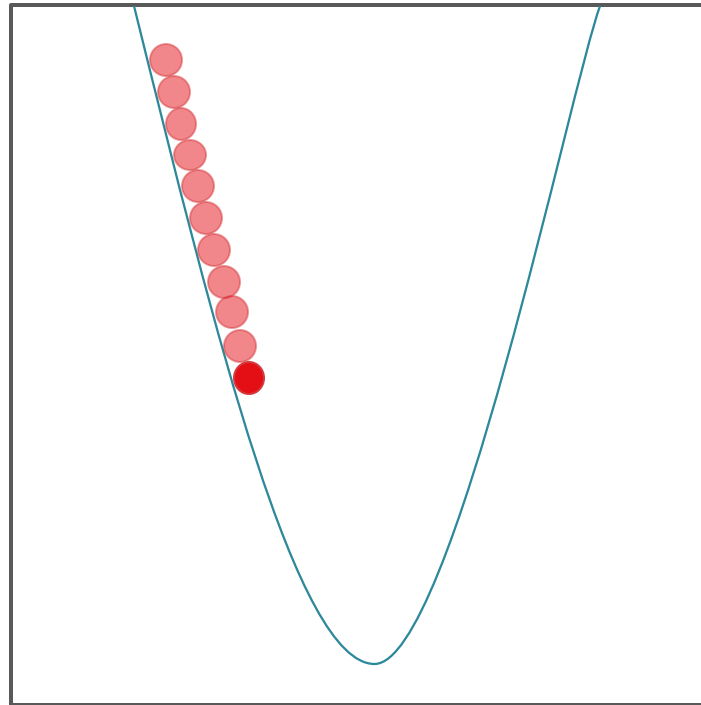


Small  $\eta$

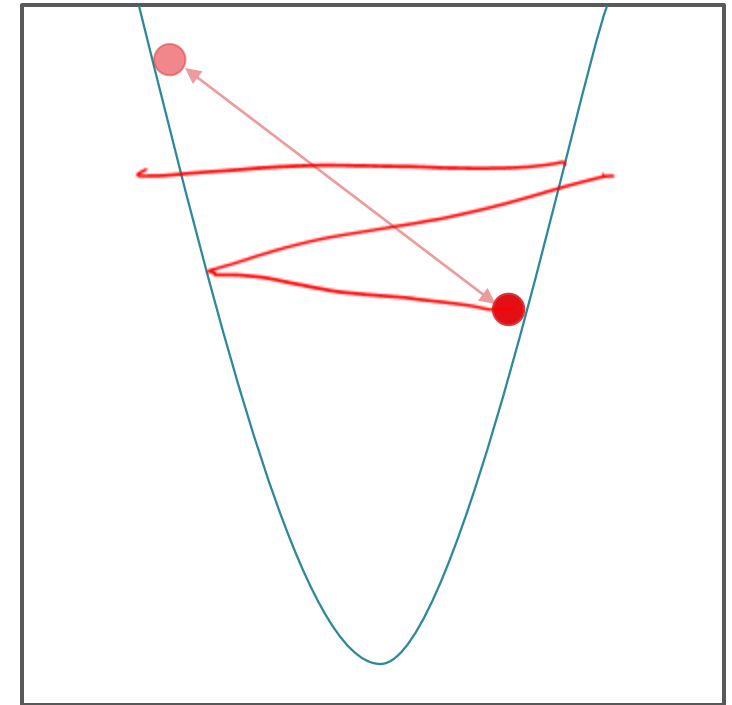


Large  $\eta$

# Gradient Descent: Step Size



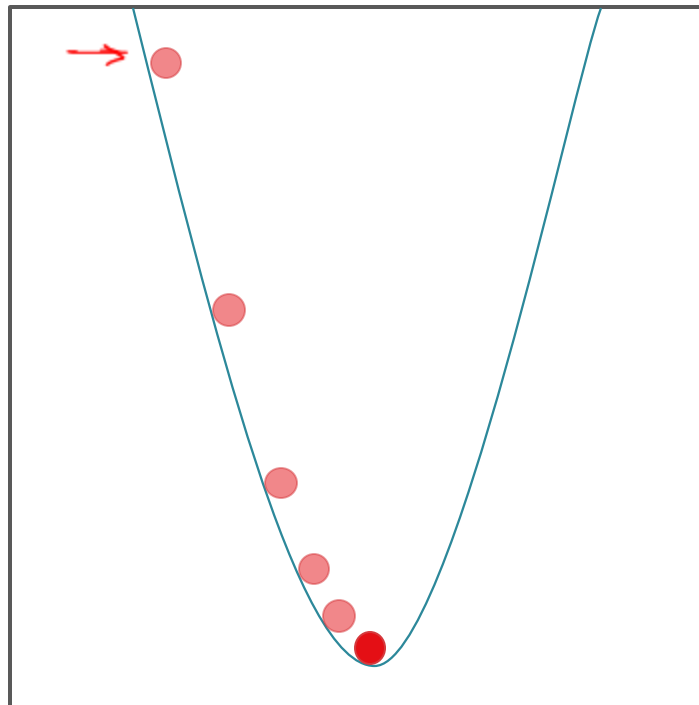
Small  $\eta$



Large  $\eta$

# Gradient Descent: Step Size

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



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

# Gradient Descent

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

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

- $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \underbrace{\eta^{(t)}}_{\text{update rule}} \underbrace{\hat{\mathbf{v}}^{(t)}}_{\text{direction}}$   
 $= \mathbf{w}^{(t)} + \eta^{(0)} \cancel{\|\nabla_{\mathbf{w}} \ell_{\mathcal{D}}(\mathbf{w}^{(t)})\|} \times \frac{-\nabla_{\mathbf{w}} \ell_{\mathcal{D}}(\mathbf{w}^{(t)})}{\cancel{\|\nabla_{\mathbf{w}} \ell_{\mathcal{D}}(\mathbf{w}^{(t)})\|}}$   
 $= \mathbf{w}^{(t)} - \eta \nabla_{\mathbf{w}} \ell_{\mathcal{D}}(\mathbf{w}^{(t)})$

# Gradient Descent

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

# Gradient Descent

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

# Gradient Descent

- Input:  $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N, \eta, \epsilon$
- 1. Initialize  $\mathbf{w}^{(0)}$  to all zeros and set  $t = 0$
- 2. While  $\|\nabla_{\mathbf{w}} \ell_{\mathcal{D}}(\mathbf{w}^{(t)})\| > \epsilon$ 
  - a. Compute the gradient:  
 $\nabla_{\mathbf{w}} \ell_{\mathcal{D}}(\mathbf{w}^{(t)})$
  - b. Update  $\mathbf{w}$ :  $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta \nabla_{\mathbf{w}} \ell_{\mathcal{D}}(\mathbf{w}^{(t)})$
  - c. Increment  $t$ :  $t \leftarrow t + 1$
- Output:  $\mathbf{w}^{(t)}$

# Gradient Descent

- Input:  $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N, \eta, T$ 
  1. Initialize  $\mathbf{w}^{(0)}$  to all zeros and set  $t = 0$
  2. While  $t < T$ 
    - a. Compute the gradient:  
 $\nabla_{\mathbf{w}} \ell_{\mathcal{D}}(\mathbf{w}^{(t)})$
    - b. Update  $\mathbf{w}$ :  $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta \nabla_{\mathbf{w}} \ell_{\mathcal{D}}(\mathbf{w}^{(t)})$
    - c. Increment  $t$ :  $t \leftarrow t + 1$
- Output:  $\mathbf{w}^{(t)}$



# Why Gradient Descent for linear regression?

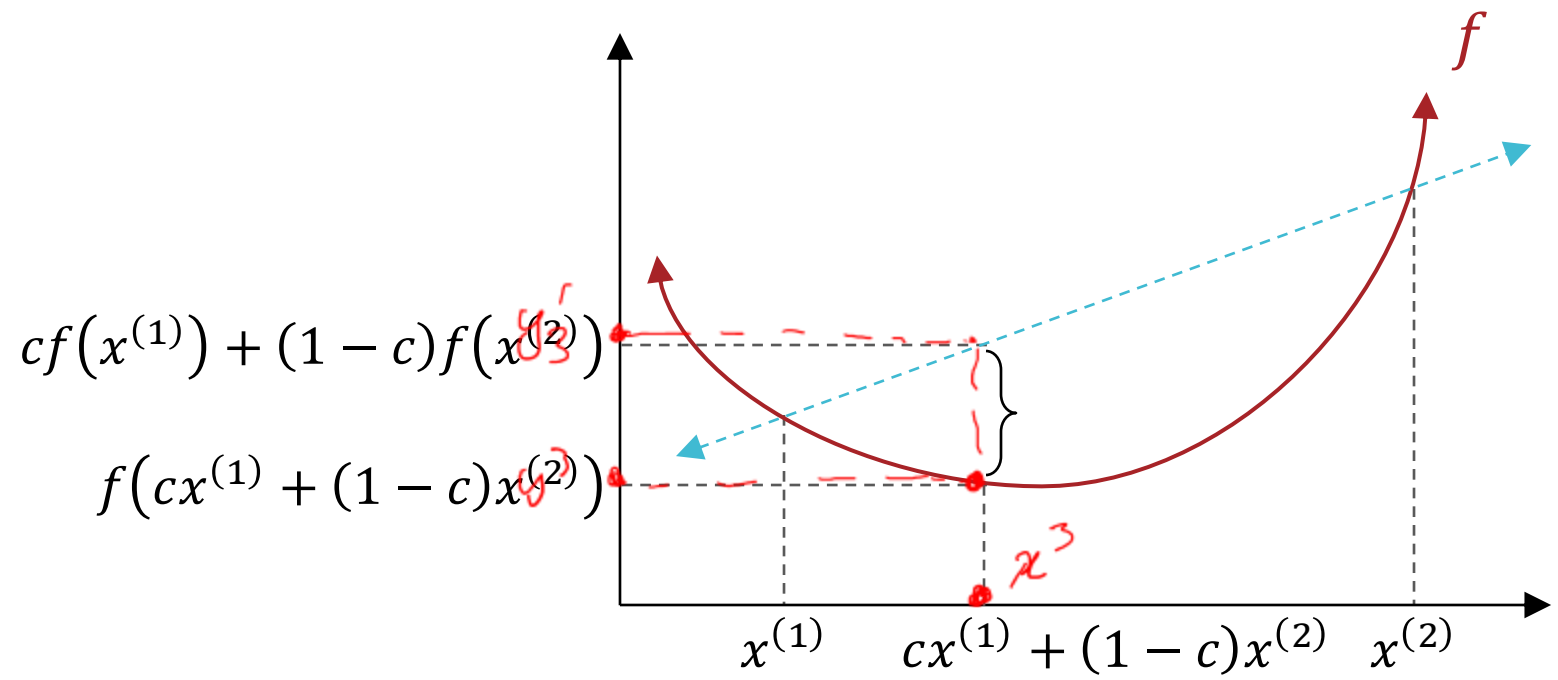
- Input:  $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N, \eta, T$ 
  1. Initialize  $\mathbf{w}^{(0)}$  to all zeros and set  $t = 0$
  2. While TERMINATION CRITERION is not satisfied
    - a. Compute the gradient:
$$\nabla_{\mathbf{w}} \ell_{\mathcal{D}}(\mathbf{w}^{(t)})$$
    - b. Update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

# Convexity

- A function  $f: \mathbb{R}^D \rightarrow \mathbb{R}$  is convex if

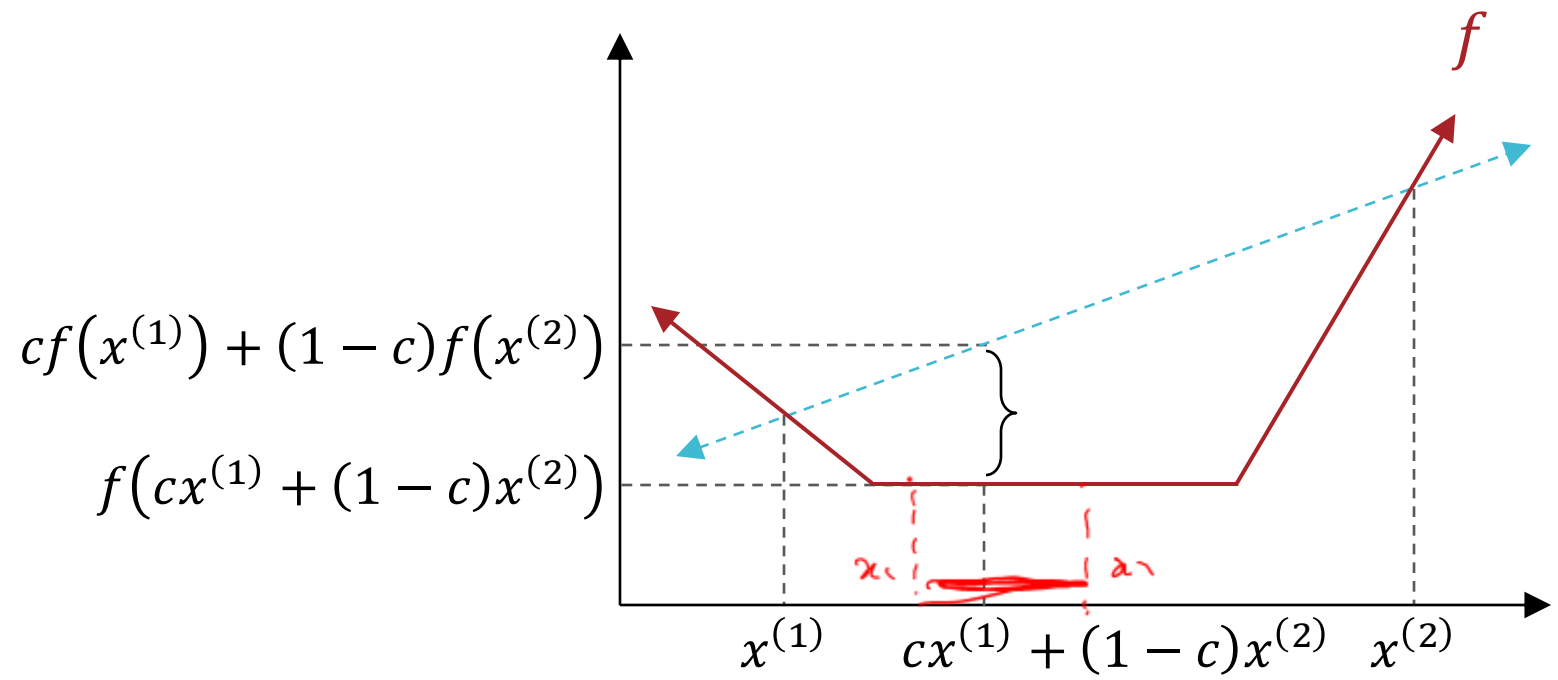
$$\forall \underline{x}^{(1)} \in \mathbb{R}^D, \underline{x}^{(2)} \in \mathbb{R}^D \text{ and } 0 \leq c \leq 1$$

$$f(c\underline{x}^{(1)} + (1-c)\underline{x}^{(2)}) \leq cf(\underline{x}^{(1)}) + (1-c)f(\underline{x}^{(2)})$$



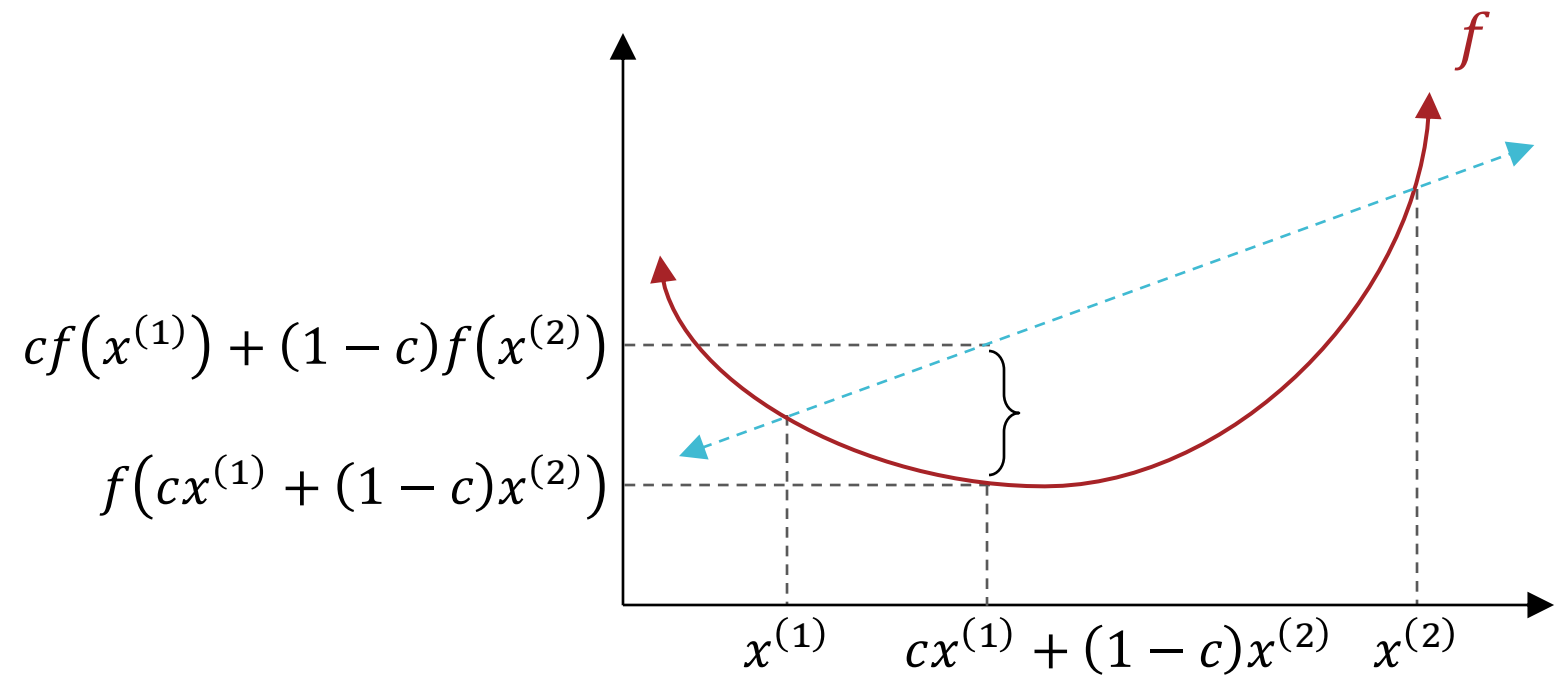
# Convexity

- A function  $f: \mathbb{R}^D \rightarrow \mathbb{R}$  is convex if  
 $\forall \mathbf{x}^{(1)} \in \mathbb{R}^D, \mathbf{x}^{(2)} \in \mathbb{R}^D$  and  $0 \leq c \leq 1$   
$$f(c\mathbf{x}^{(1)} + (1-c)\mathbf{x}^{(2)}) \leq cf(\mathbf{x}^{(1)}) + (1-c)f(\mathbf{x}^{(2)})$$

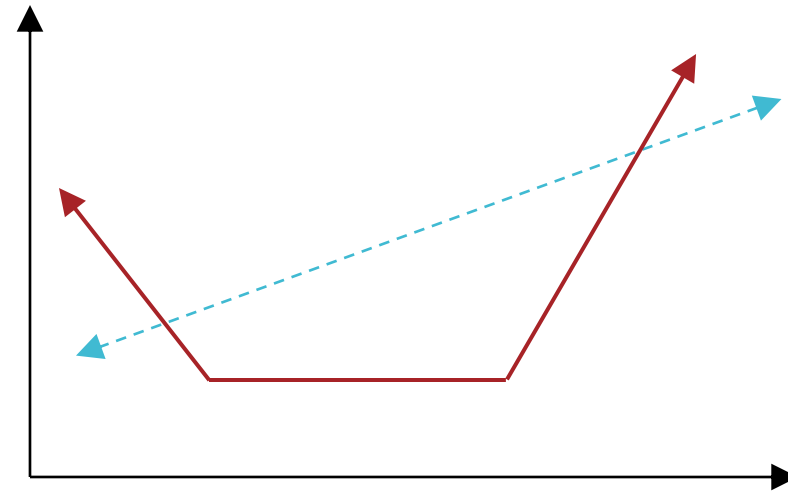


# Strict Convexity

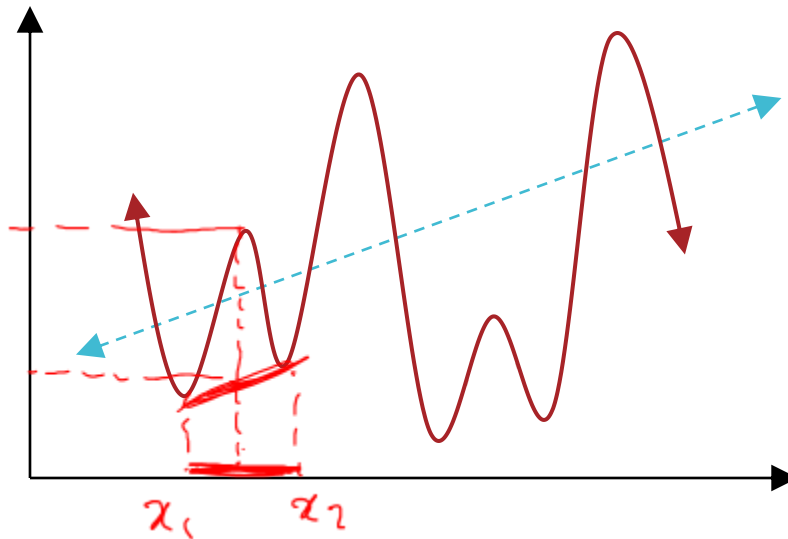
- A function  $f: \mathbb{R}^D \rightarrow \mathbb{R}$  is *strictly convex* if  
 $\forall \mathbf{x}^{(1)} \in \mathbb{R}^D, \mathbf{x}^{(2)} \in \mathbb{R}^D$  and  $0 < c < 1$   
 $f(c\mathbf{x}^{(1)} + (1-c)\mathbf{x}^{(2)}) < cf(\mathbf{x}^{(1)}) + (1-c)f(\mathbf{x}^{(2)})$



# Examples

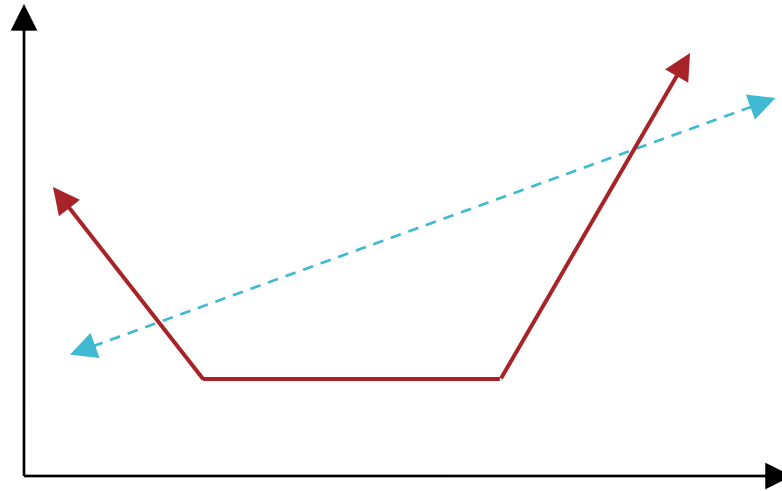


Convex functions



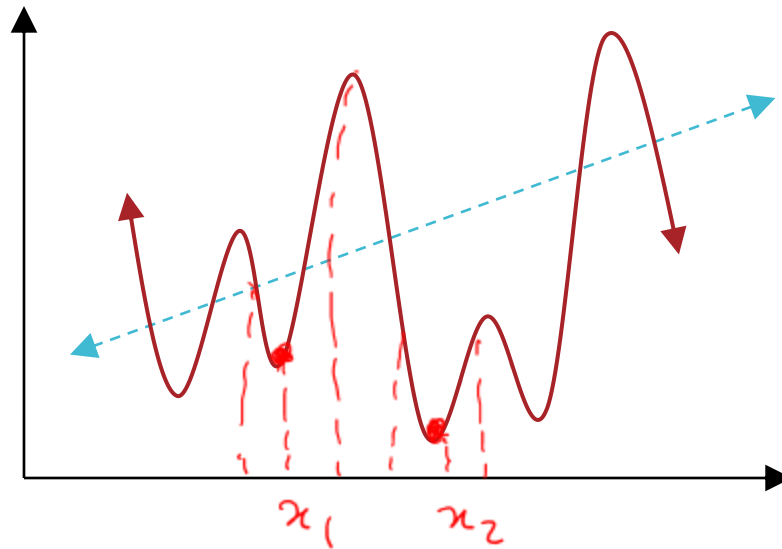
Non-convex functions

# Local vs. Global Minimum



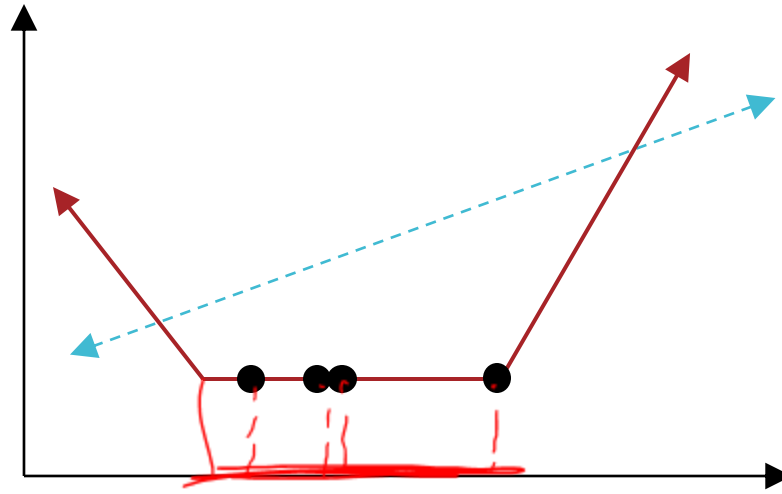
Given a function  $f: \mathbb{R}^D \rightarrow \mathbb{R}$

- $\mathbf{x}^*$  is a *global* minimum iff  $f(\mathbf{x}^*) \leq f(\mathbf{x}) \forall \mathbf{x} \in \mathbb{R}^D$

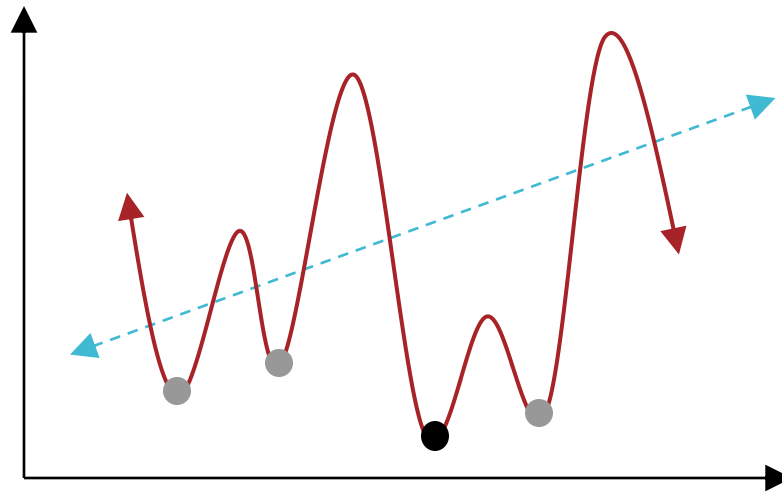


- $\mathbf{x}^*$  is a *local* minimum iff  $\exists \underline{\epsilon}$  s.t.  $f(\mathbf{x}^*) \leq f(\mathbf{x}) \forall \mathbf{x}$  s.t.  $\|\mathbf{x} - \mathbf{x}^*\|_2 < \epsilon$

# Convexity

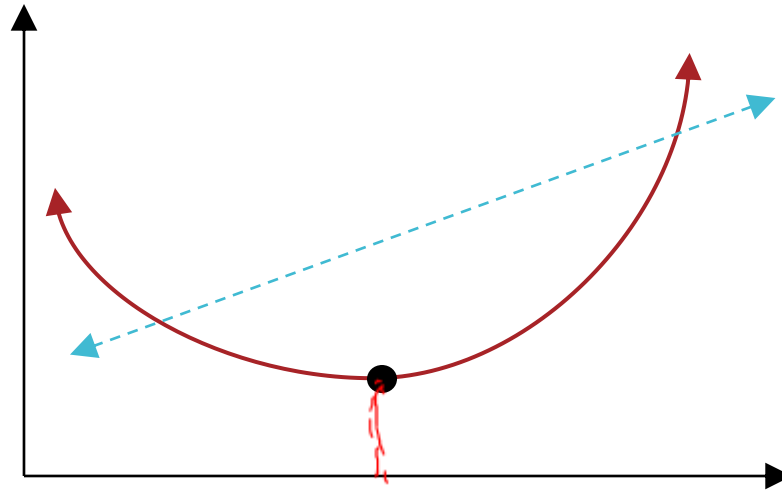


Convex functions:  
Each local minimum is a  
global minimum!

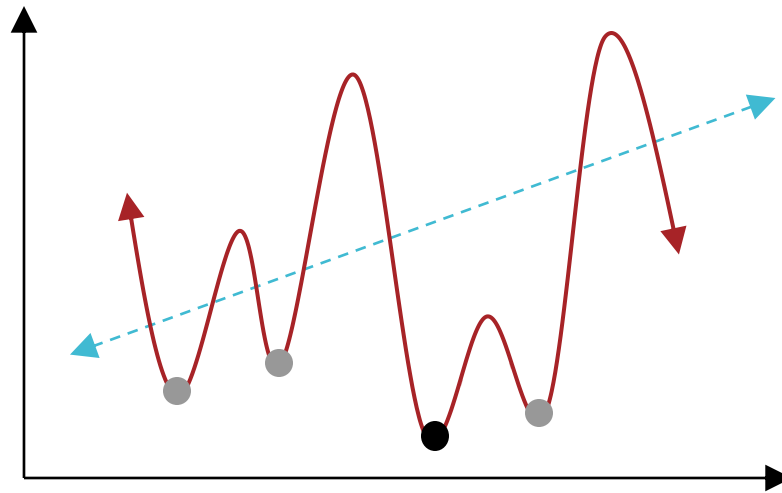


Non-convex functions:  
A local minimum may or may  
not be a global minimum...

# Convexity



Strictly convex functions:  
There exists a unique global minimum!

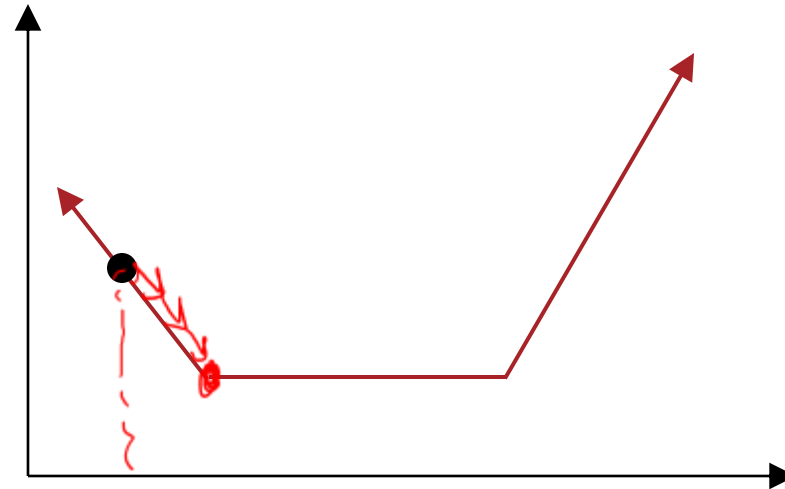


Non-convex functions:  
A local minimum may or may not be a global minimum...



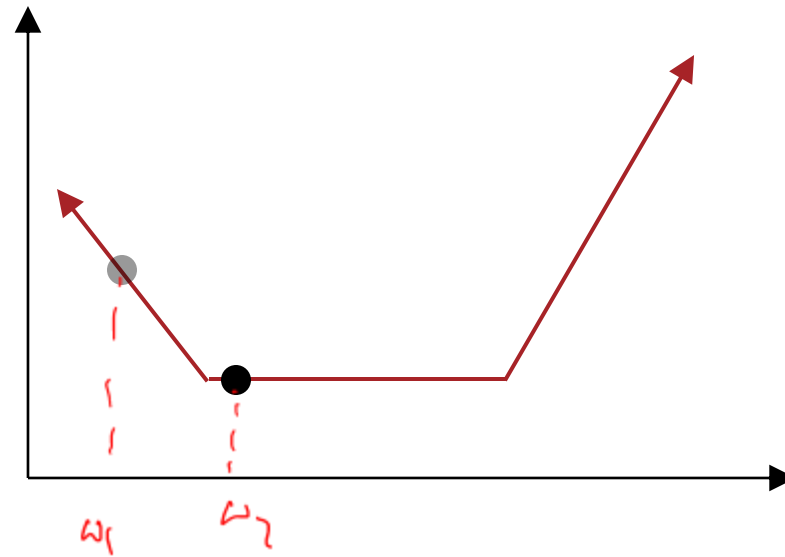
# Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Works great if the objective function is convex!



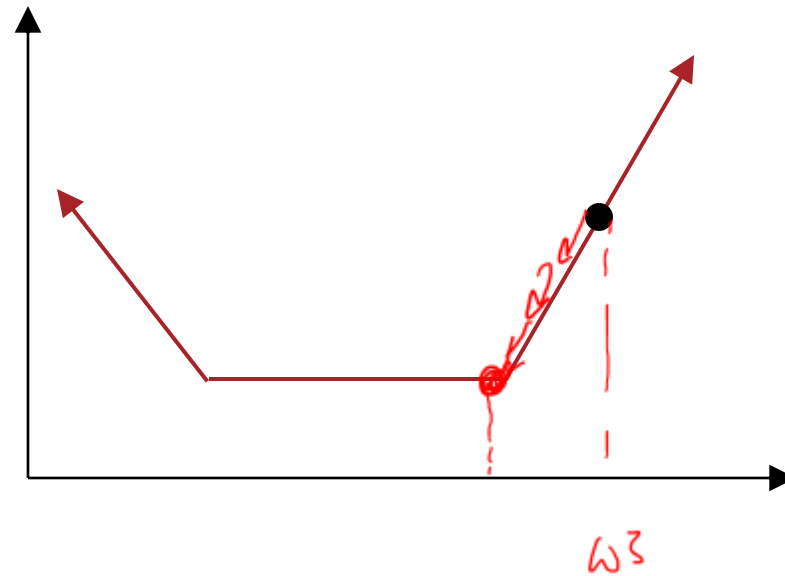
# Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Works great if the objective function is convex!



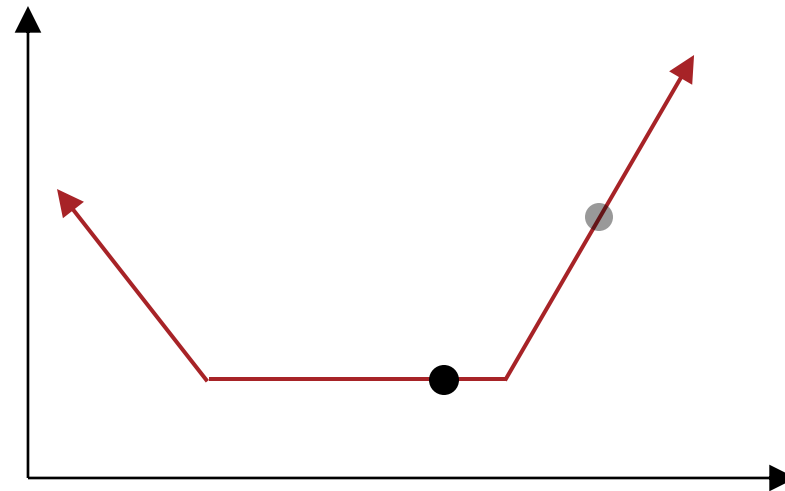
# Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Works great if the objective function is convex!



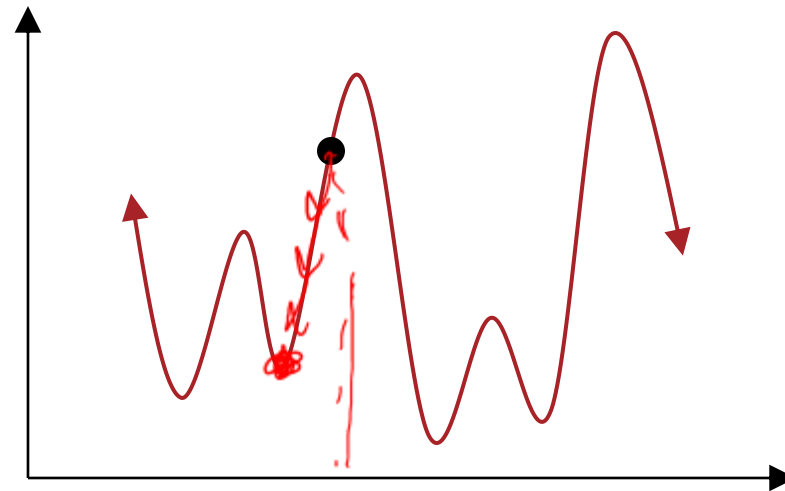
# Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Works great if the objective function is convex!



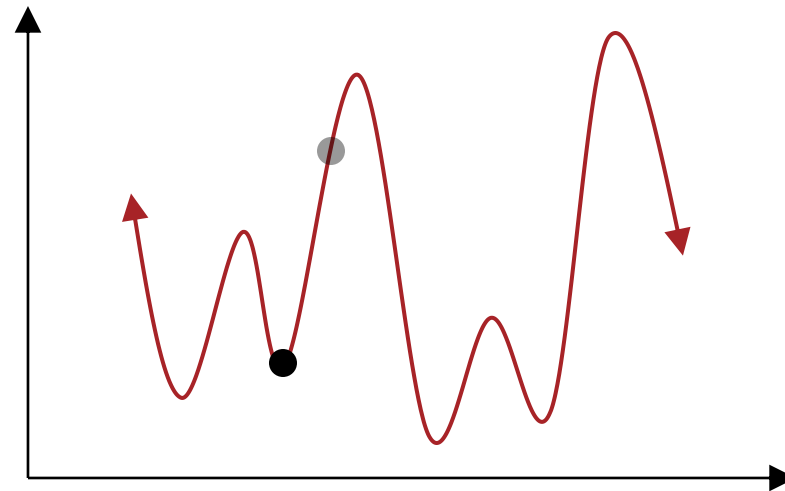
# Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Not ideal if the objective function is non-convex...



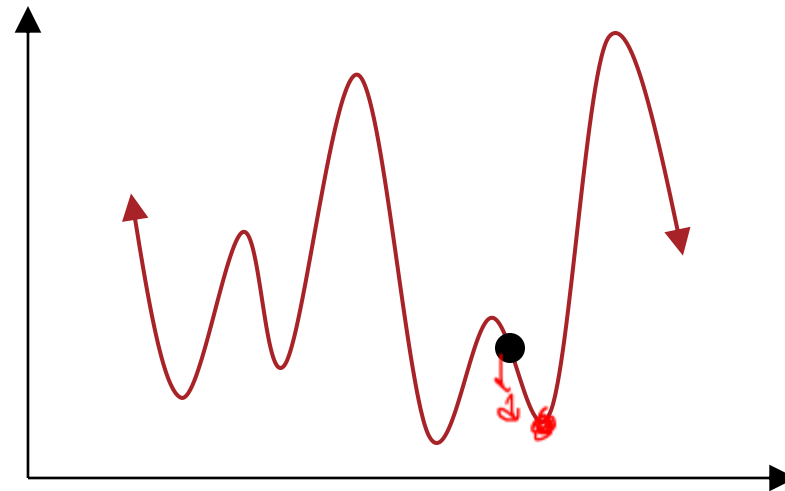
# Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Not ideal if the objective function is non-convex...



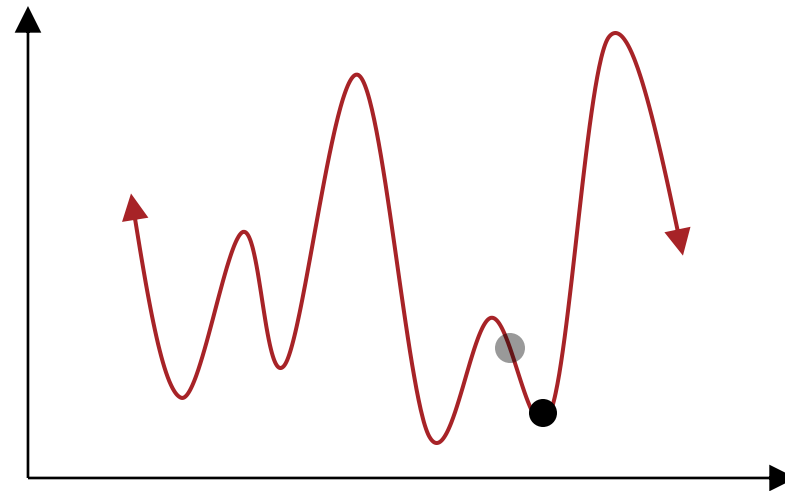
# Gradient Descent & Convexity

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Not ideal if the objective function is non-convex...



# Gradient Descent & Convexity

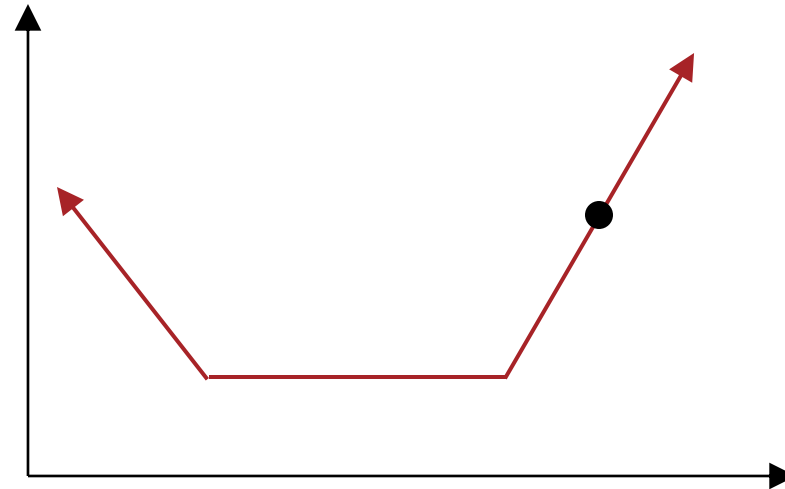
- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Not ideal if the objective function is non-convex...





The squared error for linear regression is convex (but not strictly convex)!

- Gradient descent is a local optimization algorithm – it will converge to a local minimum (if it converges)
  - Works great if the objective function is convex!



$$H_{\mathbf{w}} \ell_{\mathcal{D}}(\mathbf{w}) = \frac{2}{N} X^T X \text{ which is positive } \textit{semi-definite}$$

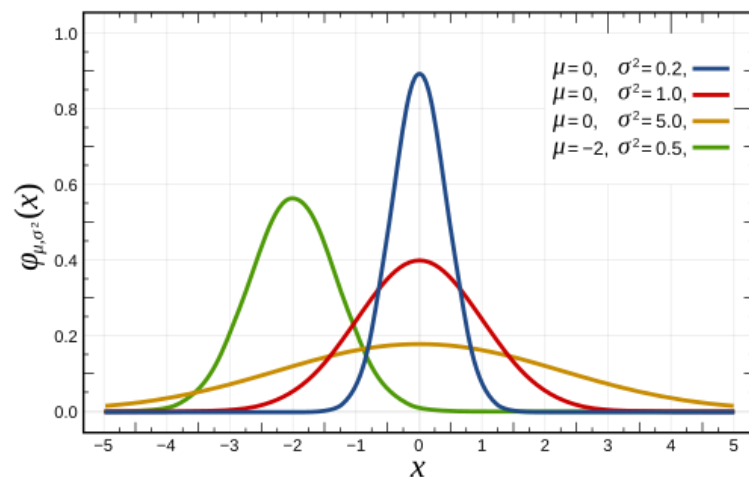
# Key Takeaways

- Closed form solution for linear regression
  - Setting the gradient equal to 0 and solving for critical points
  - Potential issues: invertibility and computational costs
- Gradient descent
  - Effect of step size
  - Termination criteria
- Convexity vs. non-convexity
  - Strong vs. weak convexity
  - Implications for local, global and unique optima

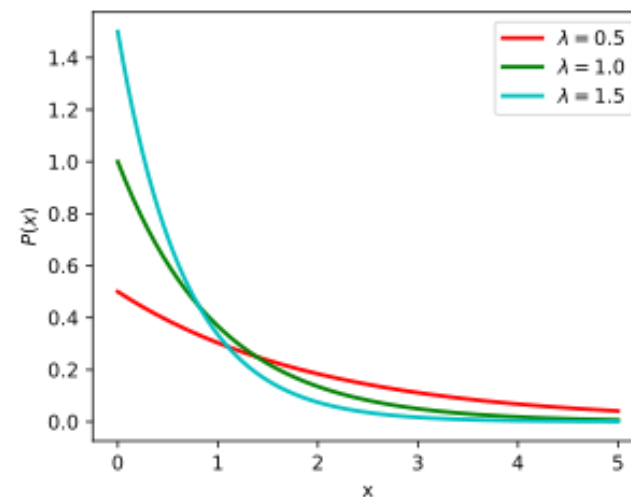
# Probabilistic Learning

- 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*,  $y \sim p^*(Y|x)$
  - Distribution,  $\overset{*}{p}(Y|x)$
  - Goal: find a distribution,  $p$ , that best approximates  $p^*$
  - Suppose  $p$  comes from a parametric family of distributions, parameterized by  $\theta$

# Parametric Distributions



$$\theta = \{\mu, \sigma^2\}$$



$$\lambda$$

# Likelihood

- Given  $N$  independent, identically distribution (iid) samples  $\mathcal{D} = \{x^{(1)}, \dots, x^{(N)}\}$  of a random variable  $X$ 
  - If  $X$  is discrete with probability mass function (pmf)  $p(X|\theta)$ , then the likelihood of  $\mathcal{D}$  is  $p(x^{(i)}|\theta)$
- If  $X$  is continuous with probability density function (pdf)  $f(X|\theta)$ , then the likelihood of  $\mathcal{D}$  is

$$L(\theta) = \prod_{n=1}^N p(x^{(n)}|\theta)$$

$$L(\theta) = \prod_{n=1}^N f(x^{(n)}|\theta)$$

# Log-Likelihood

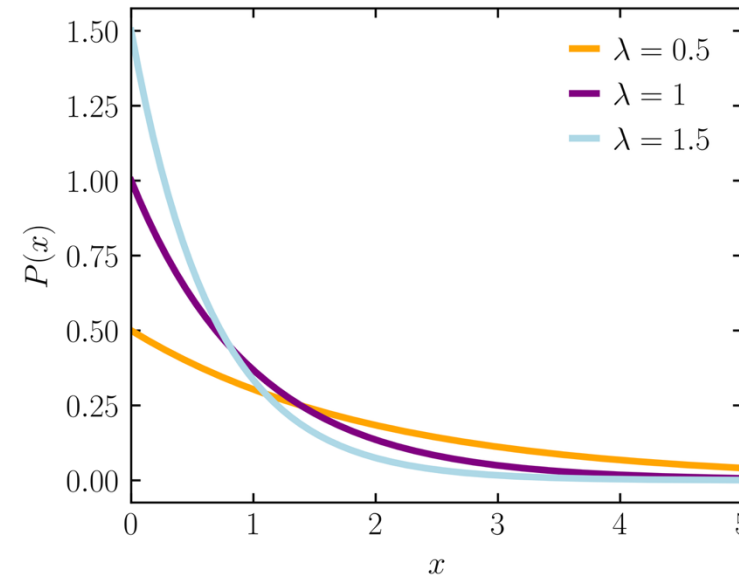
- Given  $N$  independent, identically distribution (iid) samples  $\mathcal{D} = \{x^{(1)}, \dots, x^{(N)}\}$  of a random variable  $X$ 
  - If  $X$  is discrete with probability mass function (pmf)  $p(X|\theta)$ , then the *log-likelihood* of  $\mathcal{D}$  is
- If  $X$  is continuous with probability density function (pdf)  $f(X|\theta)$ , then the *log-likelihood* of  $\mathcal{D}$  is

$$\ell(\theta) = \log \prod_{n=1}^N p(x^{(n)}|\theta) = \sum_{n=1}^N \log p(x^{(n)}|\theta)$$

$$\ell(\theta) = \log \prod_{n=1}^N f(x^{(n)}|\theta) = \sum_{n=1}^N \log f(x^{(n)}|\theta)$$

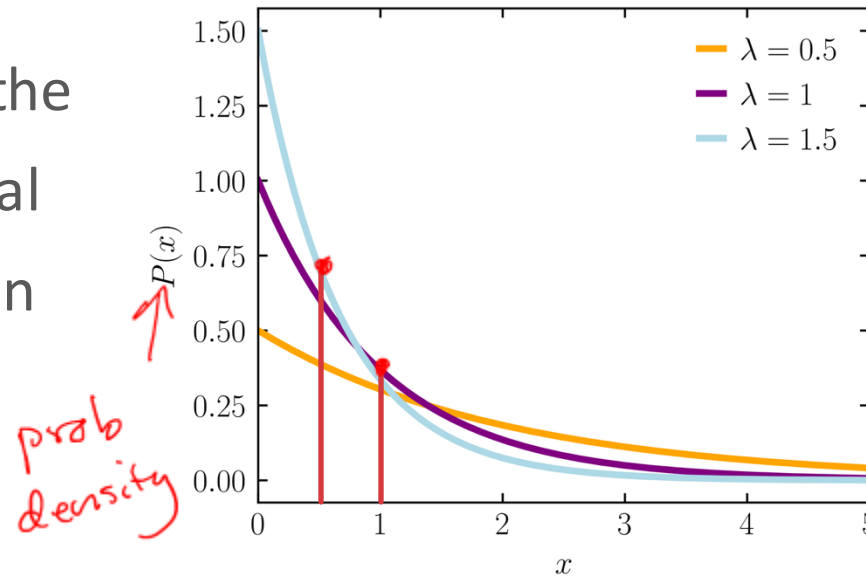
# Maximum Likelihood Estimation (MLE)

- Insight: every valid probability distribution has a finite amount of probability mass as it must sum/integrate to 1
- Idea: set the parameter(s) so that the likelihood of the samples is maximized
- Intuition: assign as much of the (finite) probability mass to the observed data *at the expense of unobserved data*
- Example: the exponential distribution



# Maximum Likelihood Estimation (MLE)

- Insight: every valid probability distribution has a finite amount of probability mass as it must sum/integrate to 1
- Idea: set the parameter(s) so that the likelihood of the samples is maximized
- Intuition: assign as much of the (finite) probability mass to the observed data *at the expense of unobserved data*
- Example: the exponential distribution

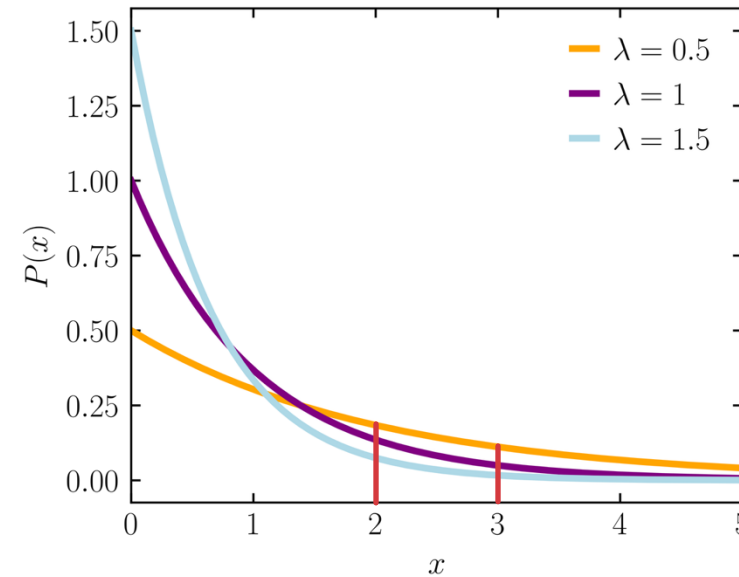


$$\{x^{(1)} = 0.5, x^{(2)} = 1\}$$



# Maximum Likelihood Estimation (MLE)

- Insight: every valid probability distribution has a finite amount of probability mass as it must sum/integrate to 1
- Idea: set the parameter(s) so that the likelihood of the samples is maximized
- Intuition: assign as much of the (finite) probability mass to the observed data *at the expense of unobserved data*
- Example: the exponential distribution



$$\{x^{(1)} = 2, x^{(2)} = 3\}$$

# Exponential Distribution MLE

- The pdf of the exponential distribution is

$$f(x|\lambda) = \underline{\lambda e^{-\lambda x}}$$

- Given  $N$  iid (independent and identically distributed) samples  $\{x^{(1)}, \dots, x^{(N)}\}$ , the likelihood is

$$\underline{L(\lambda)} = \prod_{n=1}^N f(x^{(n)}|\lambda) = \prod_{n=1}^N \lambda e^{-\lambda x^{(n)}}$$

$$\log L(\lambda) = \ell(\lambda) = \sum_{n=1}^N \log \lambda - \underbrace{\lambda x^{(n)}}$$

$$\frac{\partial \ell}{\partial \lambda} = \frac{n}{\lambda} - \sum_{n=1}^N x^{(n)}$$

$$\text{At critical point } \hat{\lambda} : \frac{N}{\hat{\lambda}} - \sum_{n=1}^N x^{(n)} = 0 \Rightarrow \hat{\lambda} = \frac{N}{\sum_{n=1}^N x^{(n)}}$$

# Exponential Distribution MLE

- The pdf of the exponential distribution is

$$f(x|\lambda) = \lambda e^{-\lambda x}$$

- Given  $N$  iid (independent and identically distributed) samples  $\{x^{(1)}, \dots, x^{(N)}\}$ , the log-likelihood is

$$\begin{aligned}\ell(\lambda) &= \sum_{n=1}^N \log f(x^{(n)}|\lambda) = \sum_{n=1}^N \log \lambda e^{-\lambda x^{(n)}} \\ &= \sum_{n=1}^N \log \lambda + \log e^{-\lambda x^{(n)}} = N \log \lambda - \lambda \sum_{n=1}^N x^{(n)}\end{aligned}$$

- Taking the partial derivative and setting it equal to 0 gives

$$\frac{\partial \ell}{\partial \lambda} = \frac{N}{\lambda} - \sum_{n=1}^N x^{(n)}$$

# Exponential Distribution MLE

- The pdf of the exponential distribution is

$$f(x|\lambda) = \lambda e^{-\lambda x}$$

- Given  $N$  iid (independent and identically distributed) samples  $\{x^{(1)}, \dots, x^{(N)}\}$ , the log-likelihood is

$$\begin{aligned}\ell(\lambda) &= \sum_{n=1}^N \log f(x^{(n)}|\lambda) = \sum_{n=1}^N \log \lambda e^{-\lambda x^{(n)}} \\ &= \sum_{n=1}^N \log \lambda + \log e^{-\lambda x^{(n)}} = N \log \lambda - \lambda \sum_{n=1}^N x^{(n)}\end{aligned}$$

- Taking the partial derivative and setting it equal to 0 gives

$$\frac{N}{\hat{\lambda}} - \sum_{n=1}^N x^{(n)} = 0 \rightarrow \frac{N}{\hat{\lambda}} = \sum_{n=1}^N x^{(n)} \rightarrow \hat{\lambda} = \frac{N}{\sum_{n=1}^N x^{(n)}}$$

# M(C)LE for Linear Regression

- If we assume a linear model with additive Gaussian noise

$$\underline{y} = \underline{\omega}^T \underline{x} + \epsilon \text{ where } \epsilon \sim N(0, \sigma^2) \rightarrow \underline{y} \sim N(\underline{\omega}^T \underline{x}, \sigma^2) \dots$$

$$\text{then given } X = \begin{bmatrix} 1 & \mathbf{x}^{(1)T} \\ 1 & \mathbf{x}^{(2)T} \\ \vdots & \vdots \\ 1 & \mathbf{x}^{(N)T} \end{bmatrix} \text{ and } \mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}, \text{ the MLE of } \boldsymbol{\omega} \text{ is}$$

$$\hat{\boldsymbol{\omega}} = \underset{\boldsymbol{\omega}}{\operatorname{argmax}} \log P(\mathbf{y}|X, \boldsymbol{\omega})$$

$\vdots$

$$= (X^T X)^{-1} X^T \mathbf{y}$$

# Bernoulli Distribution MLE

- A Bernoulli random variable takes value **1** (or heads) with probability  $\phi$  and value **0** (or tails) with probability  $1 - \phi$
- The pmf of the Bernoulli distribution is

$$p(x|\phi) = \phi^x(1 - \phi)^{1-x} \quad x \in \{0, 1\}$$

# Coin Flipping MLE

- A Bernoulli random variable takes value **1** (or heads) with probability  $\phi$  and value **0** (or tails) with probability  $1 - \phi$
- The pmf of the Bernoulli distribution is
$$p(x|\phi) = \phi^x(1 - \phi)^{1-x}$$
- Given  $N$  iid samples  $\{x^{(1)}, \dots, x^{(N)}\}$ , the log-likelihood is

# Coin Flipping MLE

- A Bernoulli random variable takes value **1** (or heads) with probability  $\phi$  and value **0** (or tails) with probability  $1 - \phi$

- The pmf of the Bernoulli distribution is

$$p(x|\phi) = \phi^x (1 - \phi)^{1-x}$$

- Given  $N$  iid samples  $\{x^{(1)}, \dots, x^{(N)}\}$ , the log-likelihood is

$$\ell(\phi) = \sum_{n=1}^N \log p(x^{(n)}|\phi) = \sum_{n=1}^N \log \phi^{x^{(n)}} (1 - \phi)^{1-x^{(n)}}$$

$$= \sum_{n=1}^N x \log \phi + (1 - x) \log(1 - \phi)$$

$$= N_1 \log \phi + N_0 \log(1 - \phi)$$

- where  $N_1$  is the number of **1**'s in  $\{x^{(1)}, \dots, x^{(N)}\}$  and  $N_0$  is the number of **0**'s



# Coin Flipping MLE

- A Bernoulli random variable takes value **1** (or heads) with probability  $\phi$  and value **0** (or tails) with probability  $1 - \phi$
- The pmf of the Bernoulli distribution is
$$p(x|\phi) = \phi^x(1 - \phi)^{1-x}$$
- The partial derivative of the log-likelihood is

- where  $N_1$  is the number of **1**'s in  $\{x^{(1)}, \dots, x^{(N)}\}$  and  $N_0$  is the number of **0**'s

# Coin Flipping MLE

- A Bernoulli random variable takes value **1** (or heads) with probability  $\phi$  and value **0** (or tails) with probability  $1 - \phi$

- The pmf of the Bernoulli distribution is

$$p(x|\phi) = \phi^x(1 - \phi)^{1-x}$$

- The partial derivative of the log-likelihood is

$$\frac{\partial \ell}{\partial \phi} = \frac{N_1}{\phi} - \frac{N_0}{1 - \phi}$$

- where  $N_1$  is the number of **1**'s in  $\{x^{(1)}, \dots, x^{(N)}\}$  and  $N_0$  is the number of **0**'s

# Coin Flipping MLE

- A Bernoulli random variable takes value **1** (or heads) with probability  $\phi$  and value **0** (or tails) with probability  $1 - \phi$
- The pmf of the Bernoulli distribution is
$$p(x|\phi) = \phi^x(1 - \phi)^{1-x}$$
- The partial derivative of the log-likelihood at  $\hat{\phi}$  is

- where  $N_1$  is the number of **1**'s in  $\{x^{(1)}, \dots, x^{(N)}\}$  and  $N_0$  is the number of **0**'s

# Coin Flipping MLE

- A Bernoulli random variable takes value **1** (or heads) with probability  $\phi$  and value **0** (or tails) with probability  $1 - \phi$

- The pmf of the Bernoulli distribution is

$$p(x|\phi) = \phi^x(1 - \phi)^{1-x}$$

- The partial derivative of the log-likelihood at  $\hat{\phi}$  is

$$\frac{N_1}{\hat{\phi}} - \frac{N_0}{1 - \hat{\phi}} = 0 \rightarrow \frac{N_1}{\hat{\phi}} = \frac{N_0}{1 - \hat{\phi}}$$

$$\rightarrow N_1(1 - \hat{\phi}) = N_0\hat{\phi} \rightarrow N_1 = \hat{\phi}(N_0 + N_1)$$

$$\rightarrow \hat{\phi} = \frac{N_1}{N_0 + N_1}$$

- where  $N_1$  is the number of **1**'s in  $\{x^{(1)}, \dots, x^{(N)}\}$  and  $N_0$  is the number of **0**'s

# Maximum a Posteriori (MAP) Estimation

- Insight: sometimes we have *prior* information we want to incorporate into parameter estimation
- Idea: use Bayes rule to reason about the *posterior* distribution over the parameters
  - MLE finds  $\hat{\theta} = \operatorname{argmax}_{\theta} p(\mathcal{D}|\theta)$
  - MAP finds  $\hat{\theta} = \operatorname{argmax}_{\theta} p(\theta|\mathcal{D})$

# Maximum a Posteriori (MAP) Estimation

- Insight: sometimes we have *prior* information we want to incorporate into parameter estimation
- Idea: use Bayes rule to reason about the *posterior* distribution over the parameters

- MLE finds  $\hat{\theta} = \operatorname{argmax}_{\theta} p(\mathcal{D}|\theta)$

- MAP finds  $\hat{\theta} = \operatorname{argmax}_{\theta} p(\theta|\mathcal{D})$   
 $= \operatorname{argmax}_{\theta} p(\mathcal{D}|\theta)p(\theta)/p(\mathcal{D})$   
 $= \operatorname{argmax}_{\theta} p(\mathcal{D}|\theta)p(\theta)$

likelihood

prior

$$= \operatorname{argmax}_{\theta} \underbrace{\log p(\mathcal{D}|\theta) + \log p(\theta)}_{\text{log-posterior}}$$

# Coin Flipping MAP

- A Bernoulli random variable takes value **1** (or heads) with probability  $\phi$  and value **0** (or tails) with probability  $1 - \phi$
- The pmf of the Bernoulli distribution is

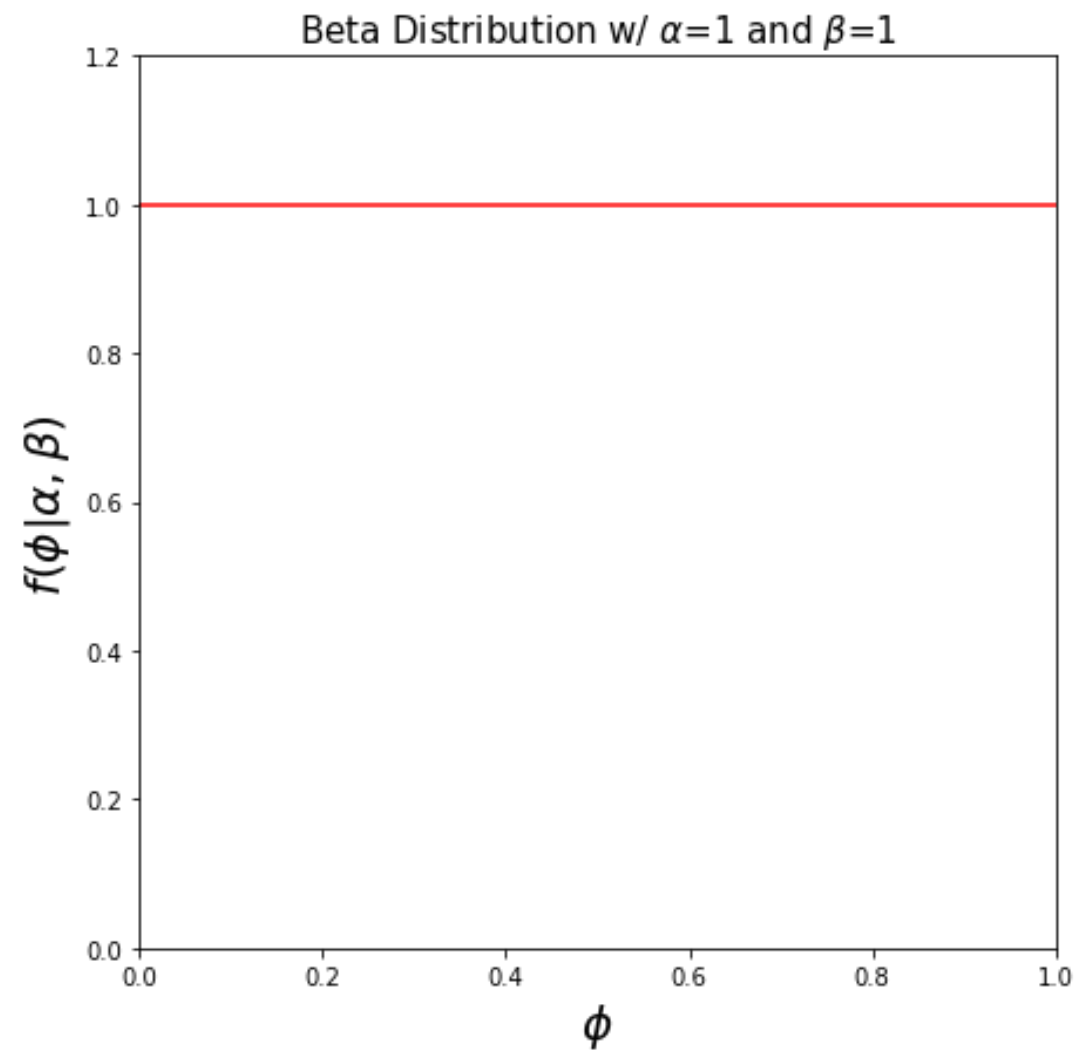
$$p(x|\phi) = \phi^x(1 - \phi)^{1-x}$$

- Assume a Beta prior over the parameter  $\phi$ , which has pdf

$$f(\phi|\alpha, \beta) = \frac{\phi^{\alpha-1}(1 - \phi)^{\beta-1}}{B(\alpha, \beta)}$$

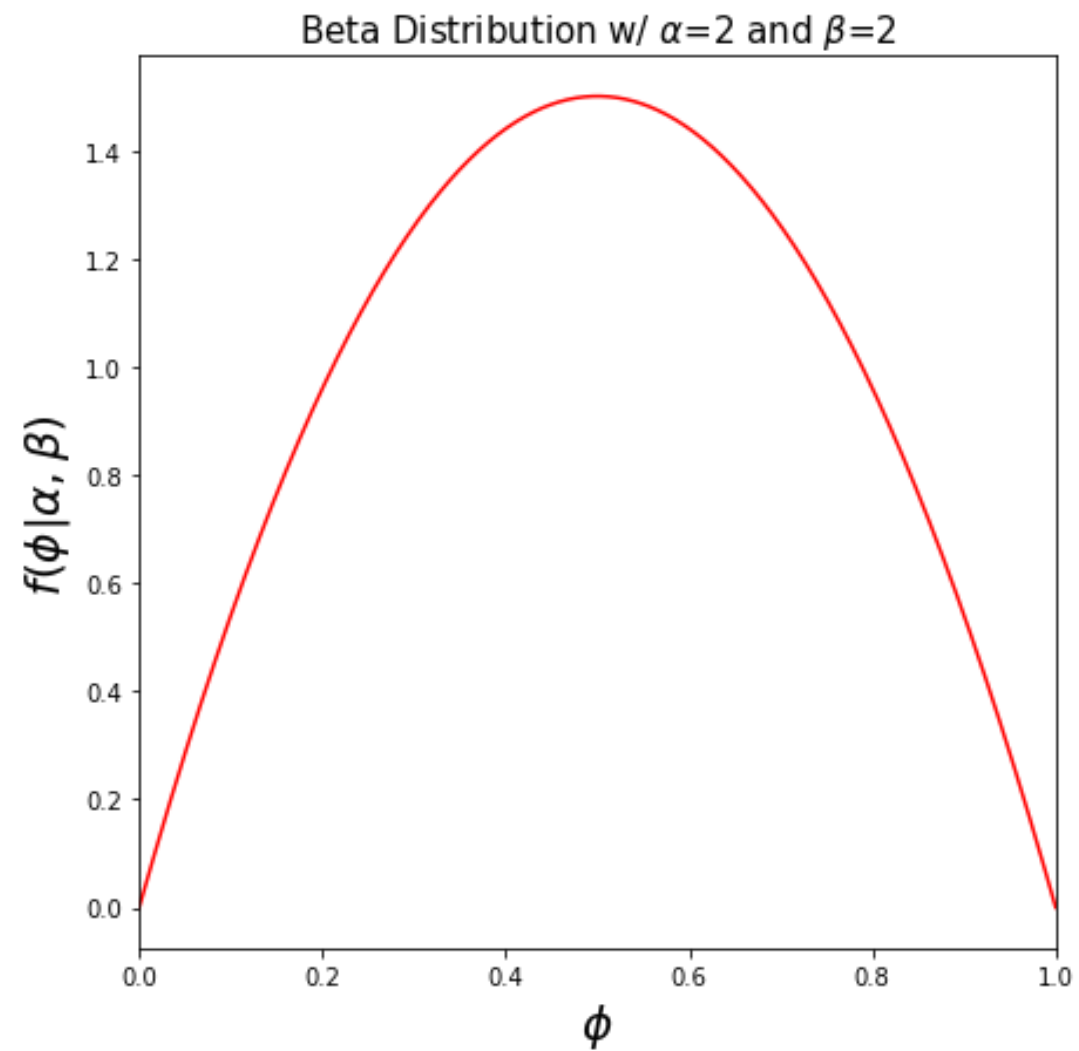
where  $B(\alpha, \beta) = \int_0^1 \phi^{\alpha-1}(1 - \phi)^{\beta-1} d\phi$  is a normalizing constant to ensure the distribution integrates to **1**

# Beta Distribution

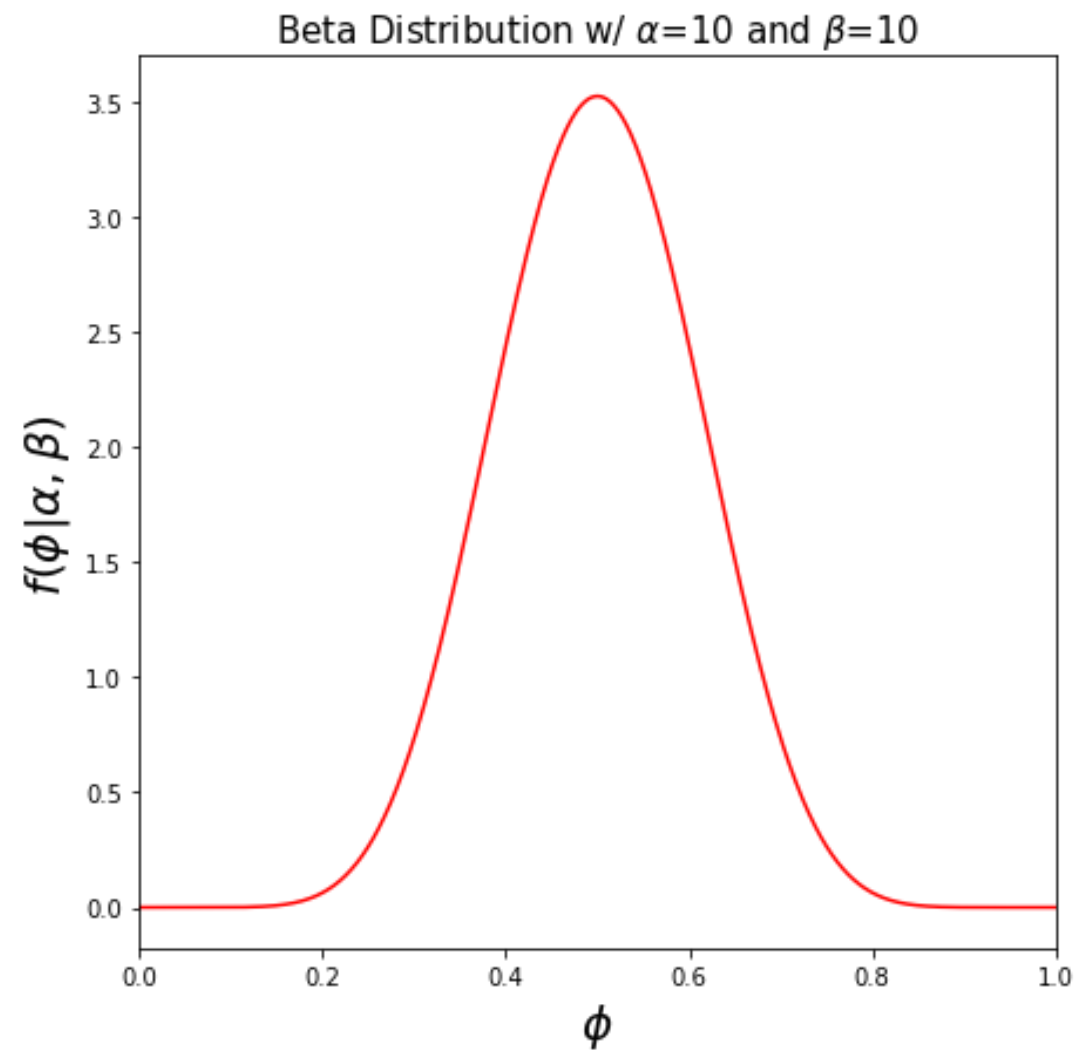




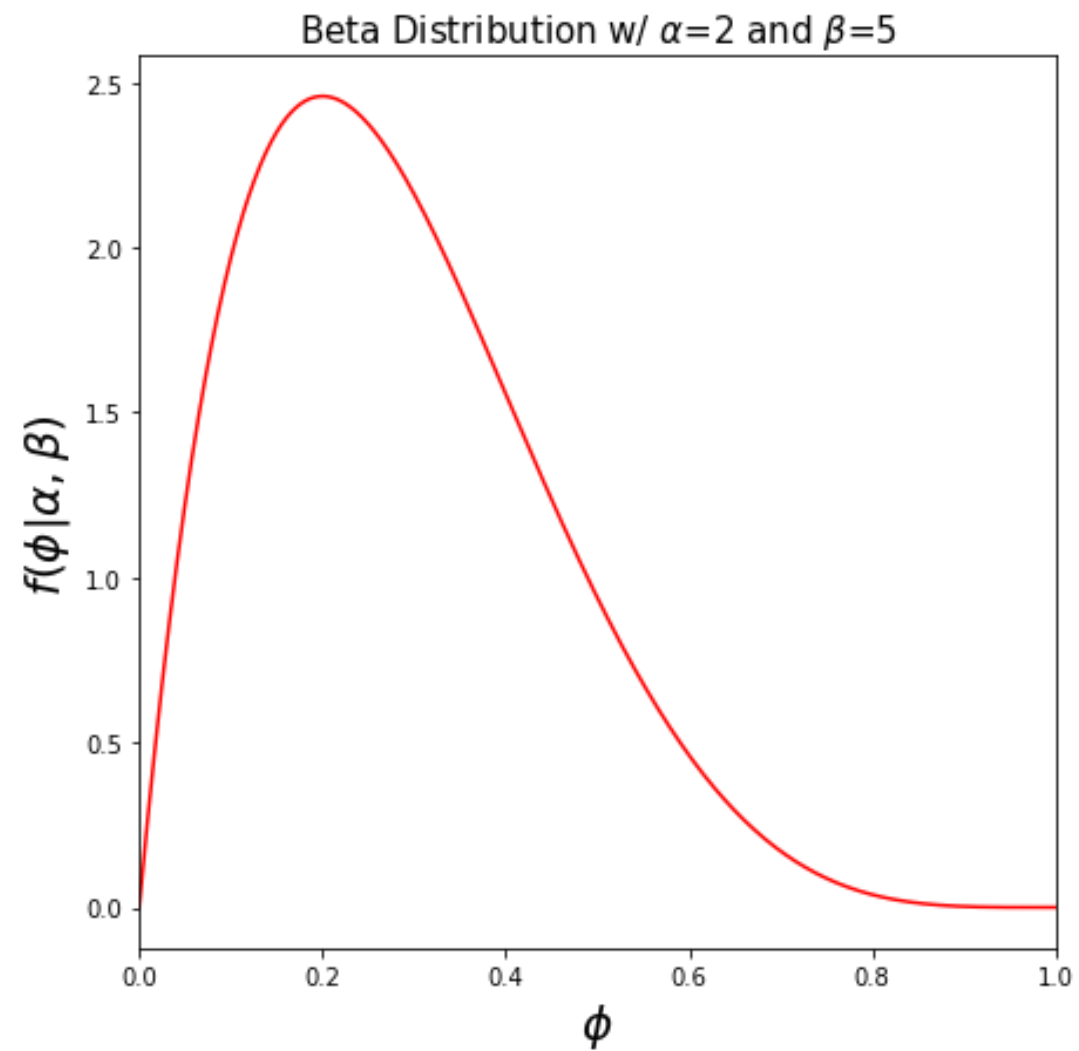
# Beta Distribution



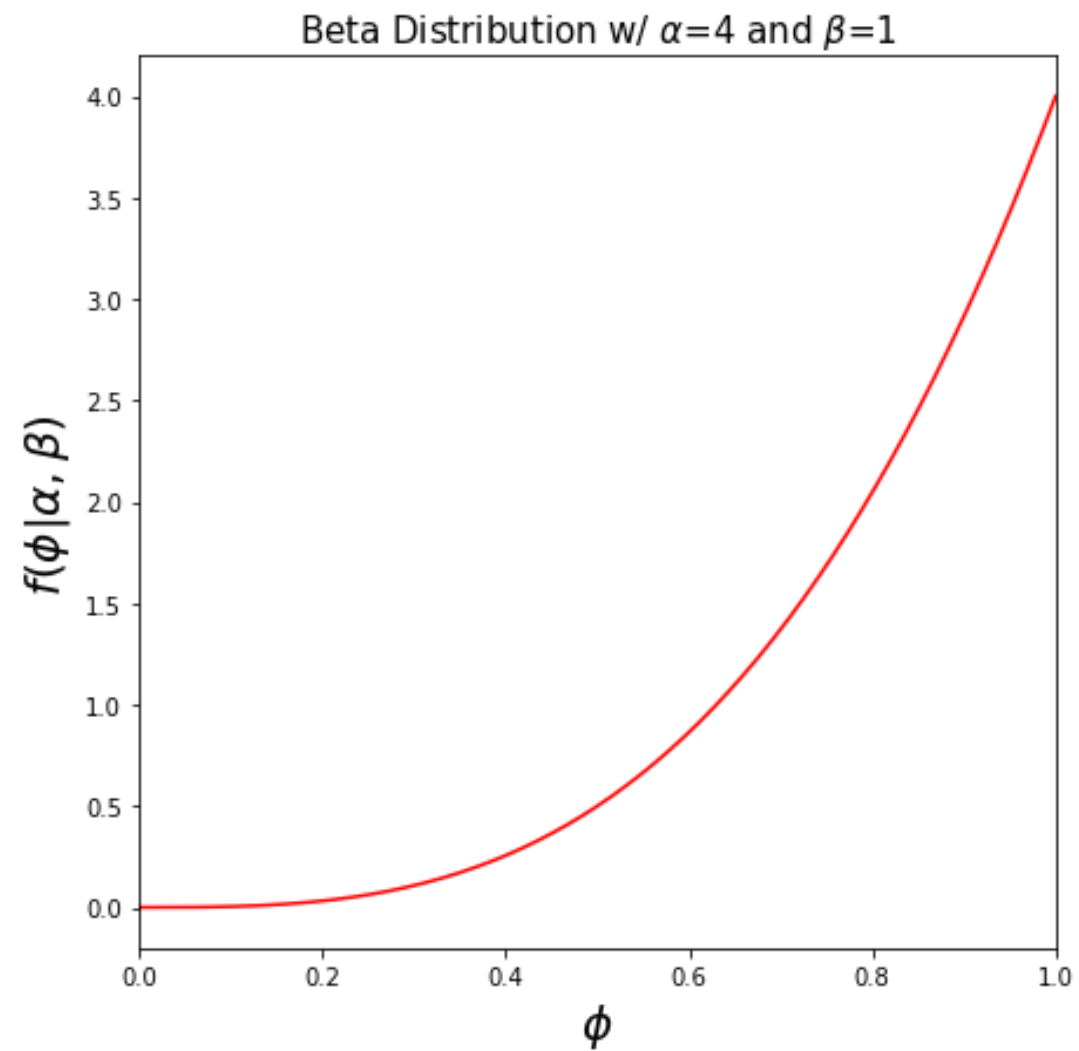
# Beta Distribution



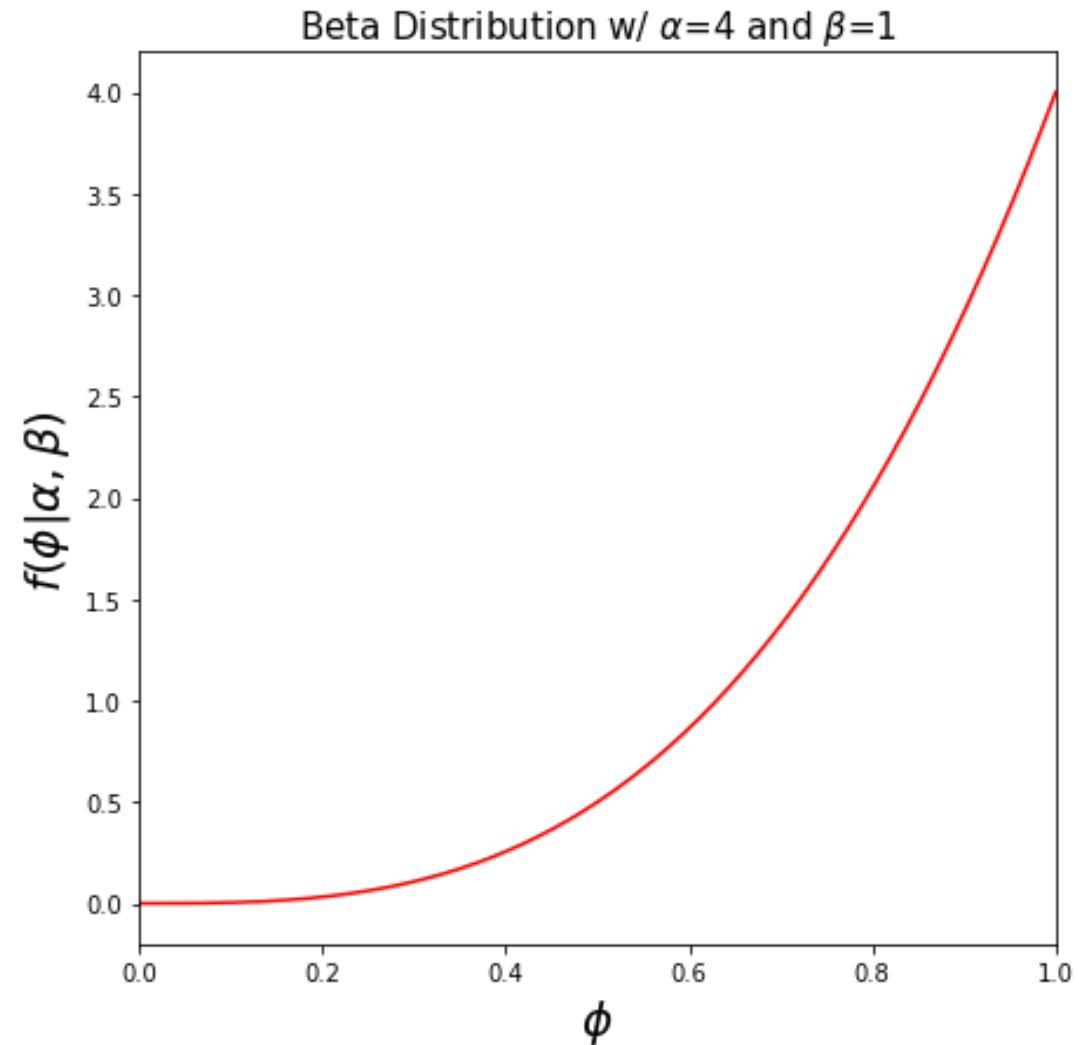
# Beta Distribution



# Beta Distribution



Okay, but why should we use this strange distribution as a prior?



# Conjugate Priors

- For a given likelihood function  $p(\mathcal{D}|\theta)$ , a prior  $p(\theta)$  is called a *conjugate prior* if the resulting posterior distribution  $p(\theta|\mathcal{D})$  is in the same family as  $p(\theta)$  i.e.,  $p(\theta|\mathcal{D})$  and  $p(\theta)$  are the same type of random variable just with different parameters
  - We like conjugate priors because they are mathematically convenient
  - However, we do not **have** to use a conjugate prior if it doesn't align with our actual prior belief.

## Example: Beta-Binomial Conjugacy

$$f(\phi|x, \alpha, \beta) = \frac{p(x|\phi)f(\phi|\alpha, \beta)}{p(x|\alpha, \beta)}$$

$$p(x|\alpha, \beta) = \int p(x|\phi)f(\phi|\alpha, \beta)d\phi$$

$$= \int \phi^x(1-\phi)^{1-x} \frac{\phi^{\alpha-1}(1-\phi)^{\beta-1}}{B(\alpha, \beta)} d\phi$$

$$= \frac{1}{B(\alpha, \beta)} \int \phi^{\alpha+x-1}(1-\phi)^{\beta-x} d\phi = \frac{B(\alpha+x, \beta-x+1)}{B(\alpha, \beta)}$$

## Example: Beta-Binomial Conjugacy

$$f(\phi|x, \alpha, \beta) = \frac{p(x|\phi)f(\phi|\alpha, \beta)}{p(x|\alpha, \beta)} = \frac{p(x|\phi)f(\phi|\alpha, \beta)}{\int p(x|\phi)f(\phi|\alpha, \beta)d\phi}$$

$$f(\phi|x, \alpha, \beta) = \frac{p(x|\phi)f(\phi|\alpha, \beta)}{\left(\frac{B(\alpha + x, \beta - x + 1)}{B(\alpha, \beta)}\right)}$$

$$\begin{aligned} &= \frac{\phi^x(1 - \phi)^{1-x} \frac{\phi^{\alpha-1}(1 - \phi)^{\beta-1}}{B(\alpha, \beta)}}{\left(\frac{B(\alpha + x, \beta - x + 1)}{B(\alpha, \beta)}\right)} \\ &= \frac{\phi^{\alpha+x-1}(1 - \phi)^{\beta-x}}{B(\alpha + x, \beta - x + 1)} = f(\phi|\alpha + x, \beta - x + 1) \end{aligned}$$

$$= f(\phi|\alpha + x, \beta + (1 - x))$$



# Beta-Binomial MAP

- Given  $N$  iid samples  $\{x^{(1)}, \dots, x^{(N)}\}$ , the log-posterior is

# Beta-Binomial MAP

- Given  $N$  iid samples  $\{x^{(1)}, \dots, x^{(N)}\}$ , the log-posterior is

$$\begin{aligned}\ell(\phi) &= \log f(\phi | \alpha + x^{(1)} + x^{(2)} + \dots + x^{(N)}, \\ &\quad (\beta + (1 - x^{(1)}) + (1 - x^{(2)}) + \dots + (1 - x^{(N)}))) \\ &= \log f(\phi | \alpha + N_1, \beta + N_0)\end{aligned}$$

where  $N_i$  is the number of  $i$ 's observed in the samples

$$\begin{aligned}&= \log \frac{\phi^{\alpha + N_1 - 1} (1 - \phi)^{\beta + N_0 - 1}}{B(\alpha, \beta)} \\ &= (\alpha + N_1 - 1) \log \phi + (\beta + N_0 - 1) \log 1 - \phi - \log B(\alpha, \beta)\end{aligned}$$

# Beta-Binomial MAP

- Given  $N$  iid samples  $\{x^{(1)}, \dots, x^{(N)}\}$ , the partial derivative of the log-posterior is

$$\frac{\partial \ell}{\partial \phi} = \frac{(\alpha + N_1 - 1)}{\phi} - \frac{(\beta + N_0 - 1)}{1 - \phi}$$
$$\vdots$$

$$\rightarrow \hat{\phi}_{MAP} = \frac{(N_1 + \alpha - 1)}{(N_0 + \beta - 1) + (N_1 + \alpha - 1)}$$

- $\alpha - 1$  is a “pseudocount” of the number of **1**’s you’ve “observed”
- $\beta - 1$  is a “pseudocount” of the number of **0**’s you’ve “observed”

# Coin Flipping MAP: Example

- Suppose  $\mathcal{D}$  consists of ten 1's or heads ( $N_1 = 10$ ) and two 0's or tails ( $N_0 = 2$ ):

$$\phi_{MLE} = \frac{10}{10 + 2} = \frac{10}{12}$$

- Using a Beta prior with  $\alpha = 2$  and  $\beta = 5$ , then

$$\phi_{MAP} = \frac{(2 - 1 + 10)}{(2 - 1 + 10) + (5 - 1 + 2)} = \frac{11}{17} < \frac{10}{12}$$

# Coin Flipping MAP: Example

- Suppose  $\mathcal{D}$  consists of ten 1's or heads ( $N_1 = 10$ ) and two 0's or tails ( $N_0 = 2$ ):

$$\phi_{MLE} = \frac{10}{10 + 2} = \frac{10}{12}$$

- Using a Beta prior with  $\alpha = 101$  and  $\beta = 101$ , then

$$\phi_{MAP} = \frac{(101 - 1 + 10)}{(101 - 1 + 10) + (101 - 1 + 2)} = \frac{110}{212} \approx \frac{1}{2}$$

# Coin Flipping MAP: Example

- Suppose  $\mathcal{D}$  consists of ten 1's or heads ( $N_1 = 10$ ) and two 0's or tails ( $N_0 = 2$ ):

$$\phi_{MLE} = \frac{10}{10 + 2} = \frac{10}{12}$$

- Using a Beta prior with  $\alpha = 1$  and  $\beta = 1$ , then

$$\phi_{MAP} = \frac{(1 - 1 + 10)}{(1 - 1 + 10) + (1 - 1 + 2)} = \frac{10}{12} = \phi_{MLE}$$

# Key Takeaways

- Two ways of estimating the parameters of a probability distribution given samples of a random variable:
  - Maximum likelihood estimation – maximize the (log-)likelihood of the observations
  - Maximum a posteriori estimation – maximize the (log-)posterior of the parameters conditioned on the observations
    - Requires a prior distribution, drawn from background knowledge or domain expertise