Overview of Maximum Likelihood Estimation

Brian Keller

Assistant Professor
Educational Psychology
College of Education, UT Austin

bk@utexas.edu

EDP 380C.28: Simulation in R

The University of Texas at Austin
College of Education

Discrete Example

- Imagine we have a measurement if a student receives free or reduced lunch ($Y = 1$) or not ($Y = 0$).
- Our goal is to model the probability of any random student having free lunch status (π).
- Said differently, we want the proportion of students who are free lunch status.

Recall Discrete Distributions...

- A discrete distribution assigns a probability for every possible value — i.e., the ***sample space*** — of a discrete random variable.
- The sum of all probabilities for the sample space must ***sum up to 1***.
- We will denote a ***general discrete distribution*** as a function (called the probability mass function or PMF) with '$p(\cdot)$'.

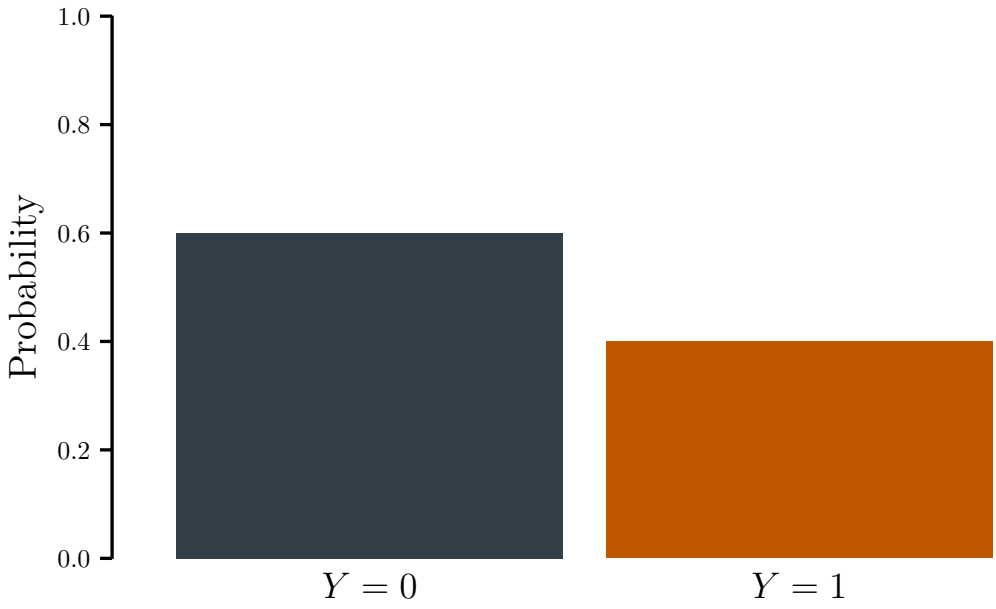
Discrete Example – Distribution

$$p\left(Y \mid \pi\right) = \begin{cases} \pi & \text{if } Y = 1 \\ 1 - \pi & \text{if } Y = 0 \end{cases}$$

or

$$p\left(Y \mid \pi\right) = \pi^{Y}(1-\pi)^{(1-Y)}$$

Discrete Example for $\pi = 0.2$



Maximum Likelihood (ML) Estimation

- Maximum likelihood identifies the populations parameters that best fit the data we observe.
- We create a numeric summary that represents how well a model fits the data and then find the parameters that will maximize it.
- For the discrete example, we will use our probability mass function to capture the discrepancy between the data and the parameters. This will be our likelihood function.

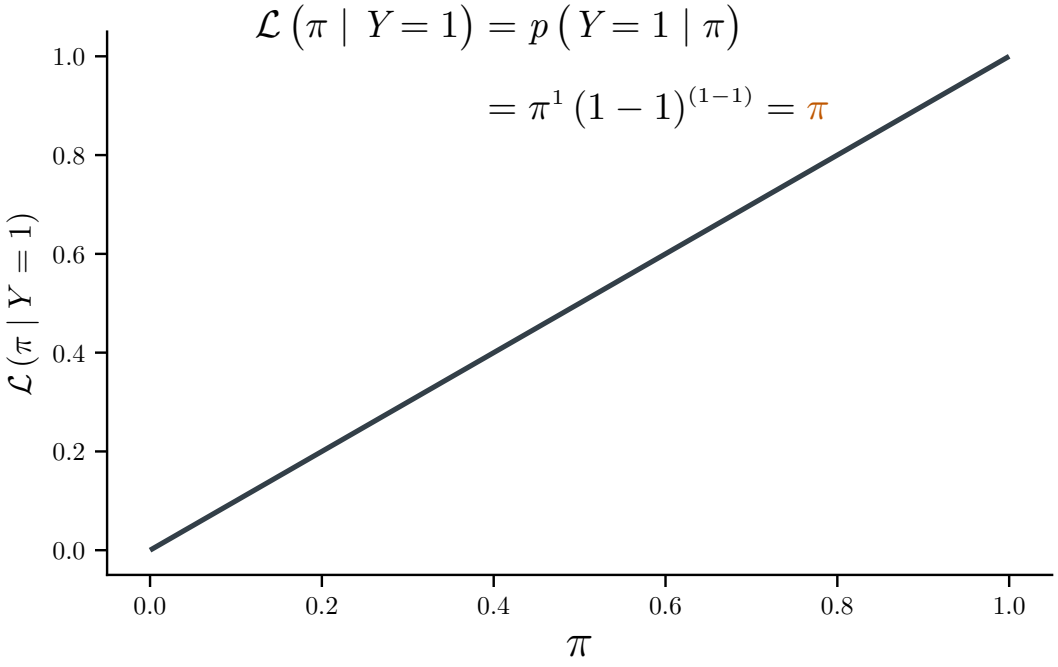
The Likelihood Function

- The likelihood function describes the relative evidence or support for different parameter values conditional on the data we observed.
- Whereas a probability distribution ***fixes the parameters***, a likelihood function ***fixes the data***.
- Discrete example likelihood function:

$$\begin{aligned}\mathcal{L}\left(\pi \mid Y = y\right) &= p\left(Y = y \mid \pi\right) \\ &= \pi^{y}\left(1-\pi\right)^{(1-y)}\end{aligned}$$

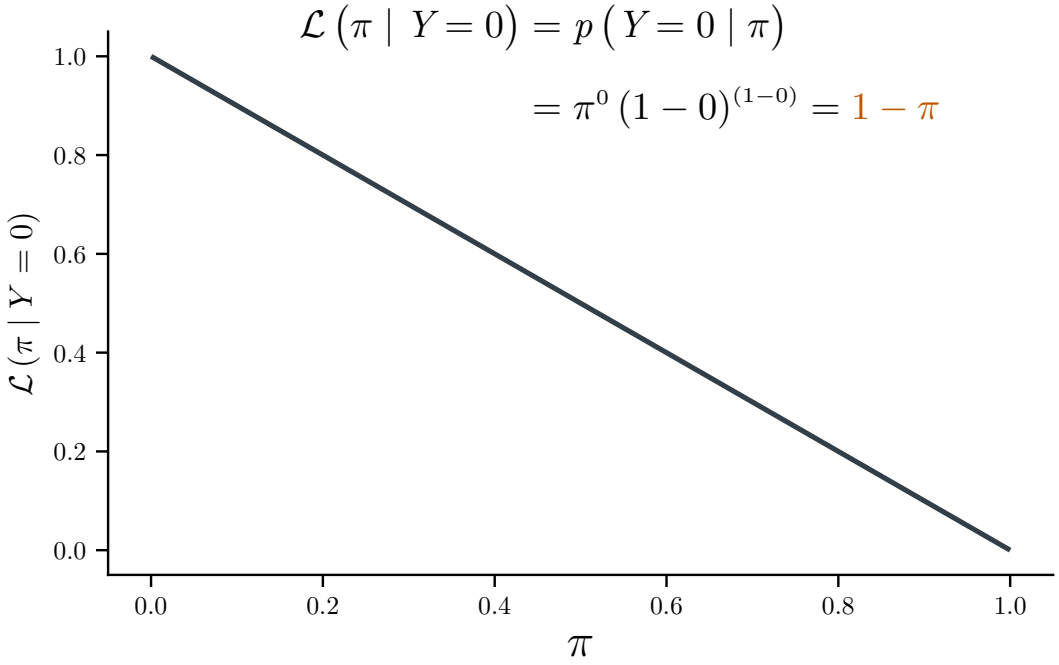
Discrete Example – Likelihood Function When $Y = 1$

$$\begin{aligned}\mathcal{L}\left(\pi \mid Y = 1\right) &= p\left(Y = 1 \mid \pi\right) \\ &= \pi^{1}\left(1-1\right)^{(1-1)} = \pi\end{aligned}$$



Discrete Example – Likelihood Function When $Y = 0$

$$\mathcal{L}\left(\pi \mid Y = 0\right) = p\left(Y = 0 \mid \pi\right)$$
$$= \pi^0 \left(1 - 0\right)^{(1-0)} = 1 - \pi$$



A Measure of Model "Fit"

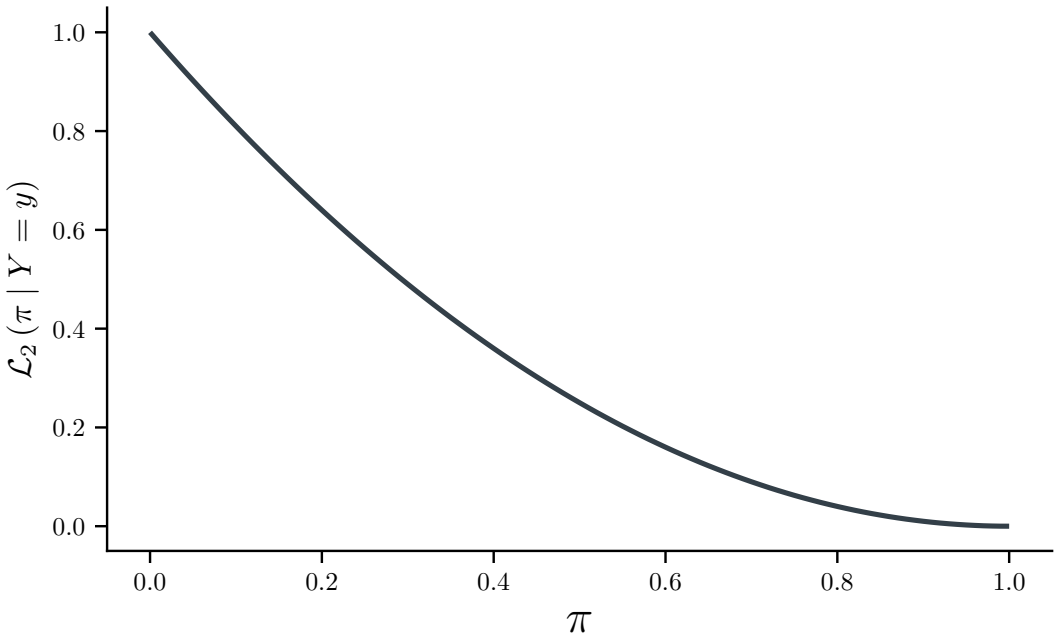
- The result from this likelihood function is a measure of the individual fit for a single observation.
- The ***result is not a probability***. In the previous examples, the area under each graph is 0.5 and a probability must sum up to 1.
- For our example, as π moves closer to the observed data (i.e., 1 or 0), the "fit" improves and results in higher values.

Characterizing the "Fit" of an Entire Sample

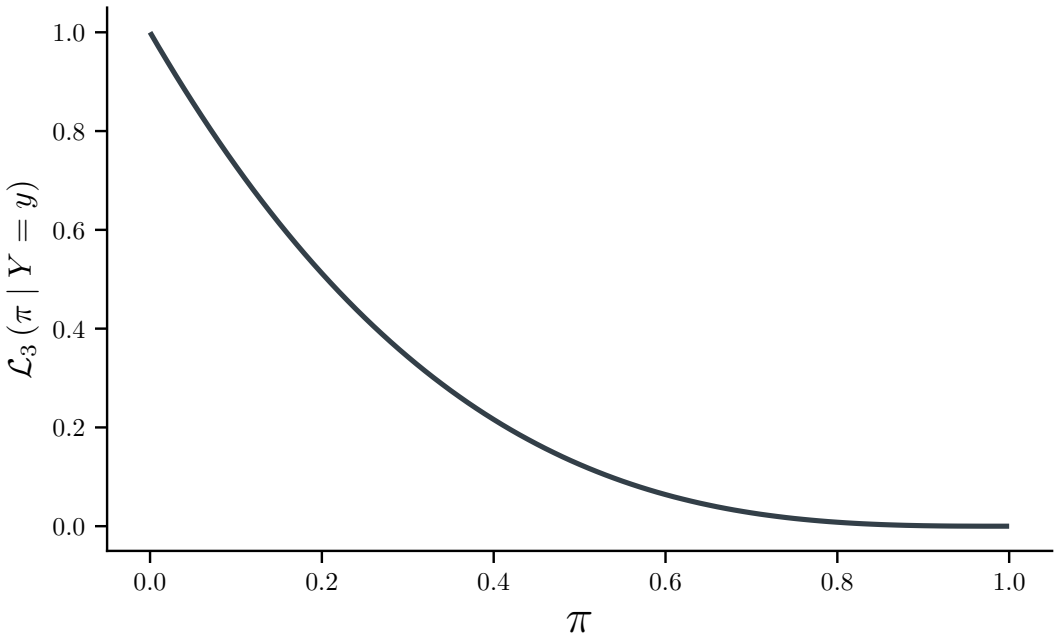
- While we have looked at individual values, we are more interested in how the model fits the entire sample.
- Therefore, we will create a one numeric summary that represents the likelihood of the parameter (π) given our sampled data (Y).

$$\begin{aligned}\mathcal{L}\left(\pi \mid Y = y\right) &= \prod_{i=1}^{N} \mathcal{L}_i\left(\pi \mid y_i\right) = \prod_{i=1}^{N} p\left(y_i \mid \pi\right) \\ &= \prod_{i=1}^{N} \pi^{y_i}\left(1-\pi\right)^{(1-y_i)}\end{aligned}$$

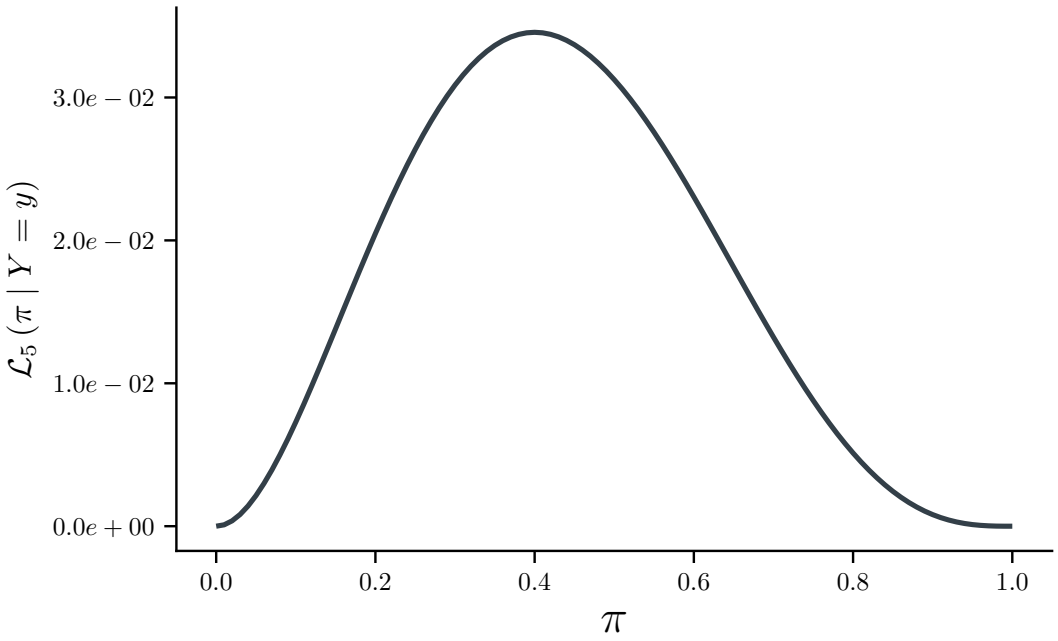
Discrete Example – Likelihood When $N = 2$



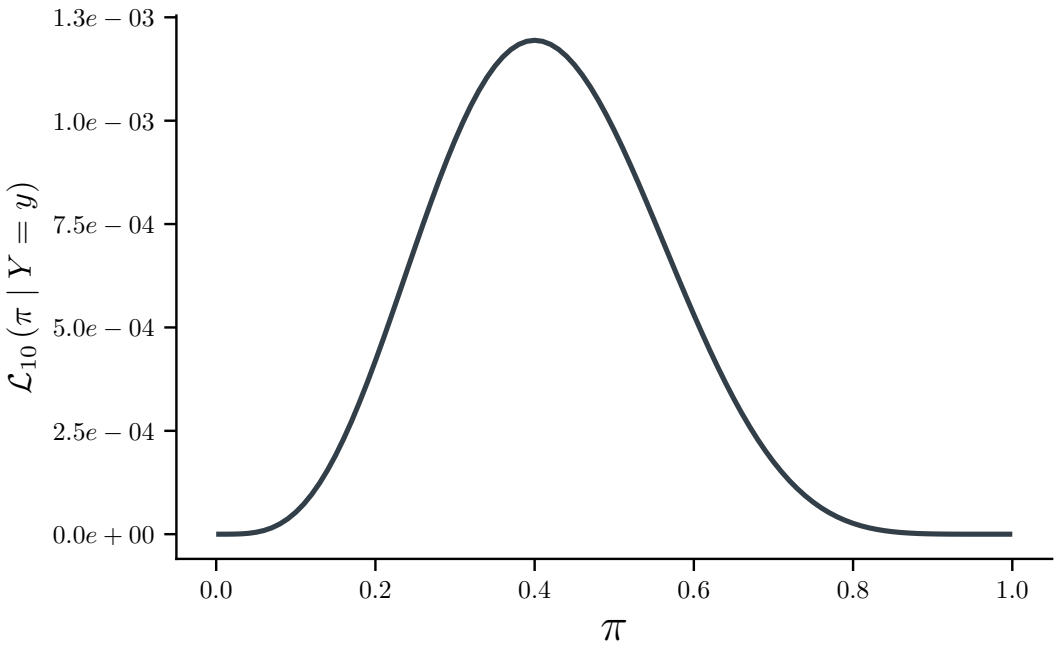
Discrete Example – Likelihood When $N = 3$



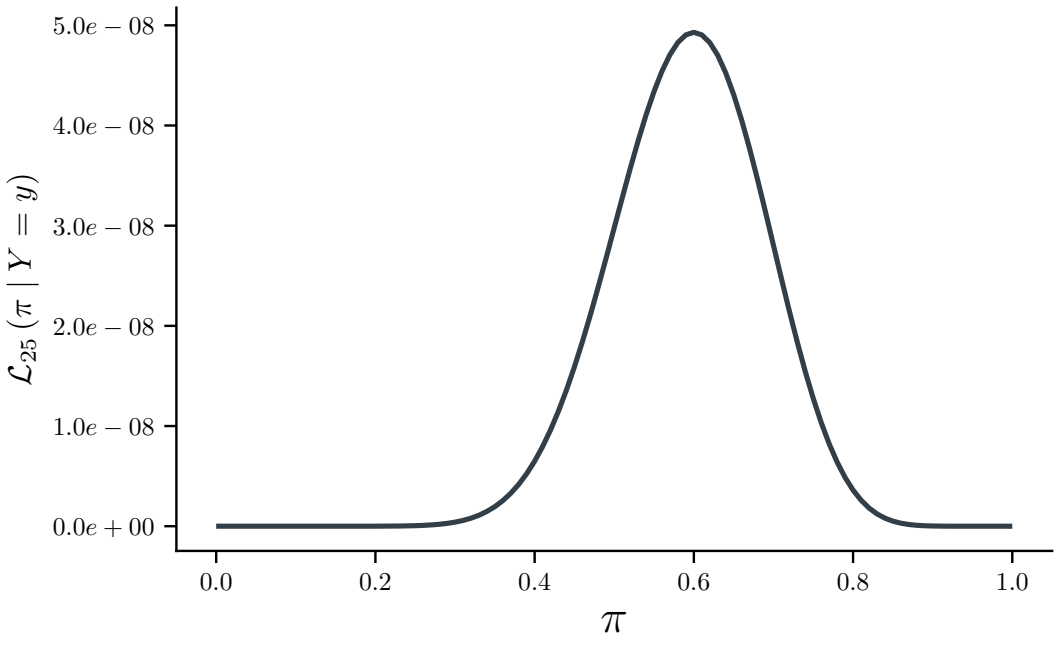
Discrete Example – Likelihood When $N = 5$



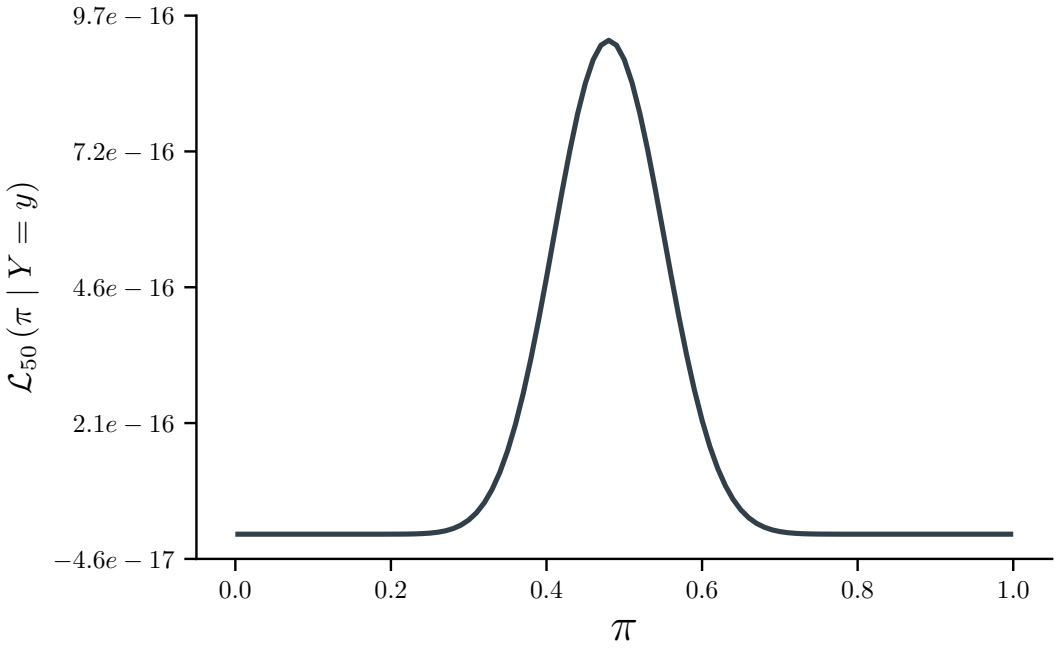
Discrete Example – Likelihood When $N = 10$



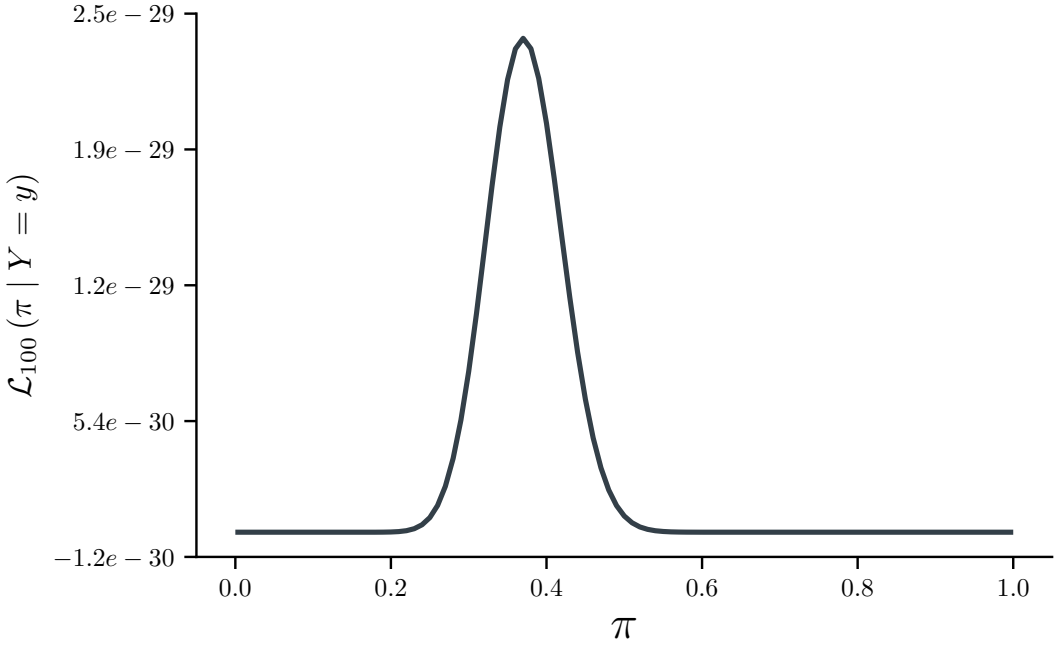
Discrete Example – Likelihood When $N = 25$



Discrete Example – Likelihood When $N = 50$



Discrete Example – Likelihood When $N = 100$



Univariate Normal Example

- Imagine we have a normally distributed math score test (Y).
- Our analytic goal is to estimate the mean (μ) and variance (σ^2) for students on this test score..

$$Y \sim \mathcal{N}\left(\mu, \sigma^2\right)$$

PDF for the Normal Distribution

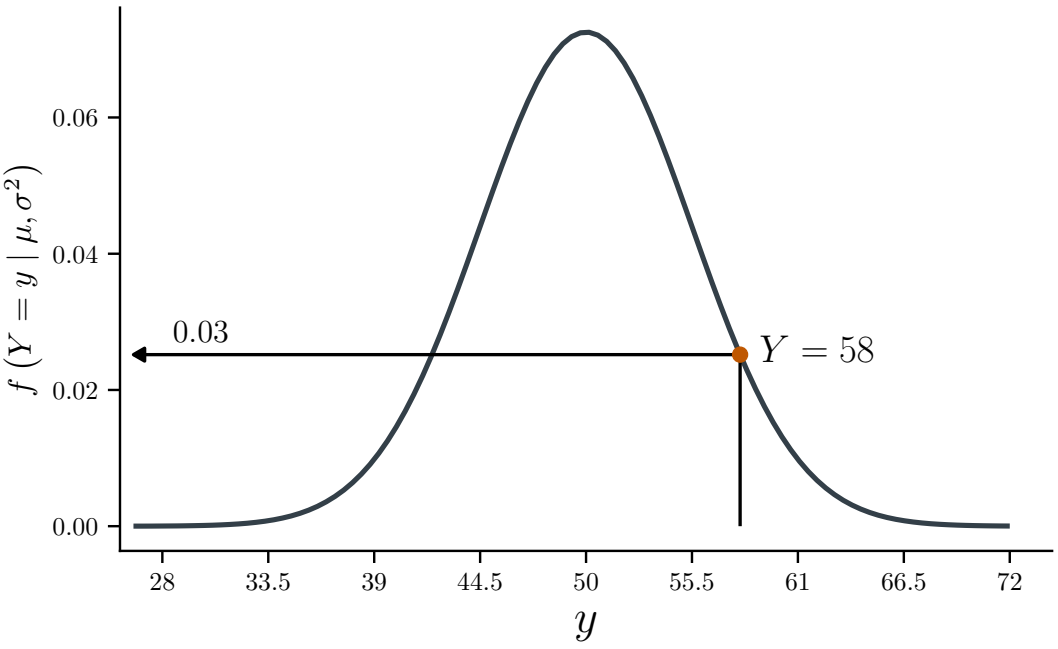
$$f\left(y \mid \mu, \sigma^2\right) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right) e^{-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2}$$

$f\left(y \mid \mu, \sigma^2\right)$: A function (i.e., density) for y given a mean (μ) and a variance (σ^2).

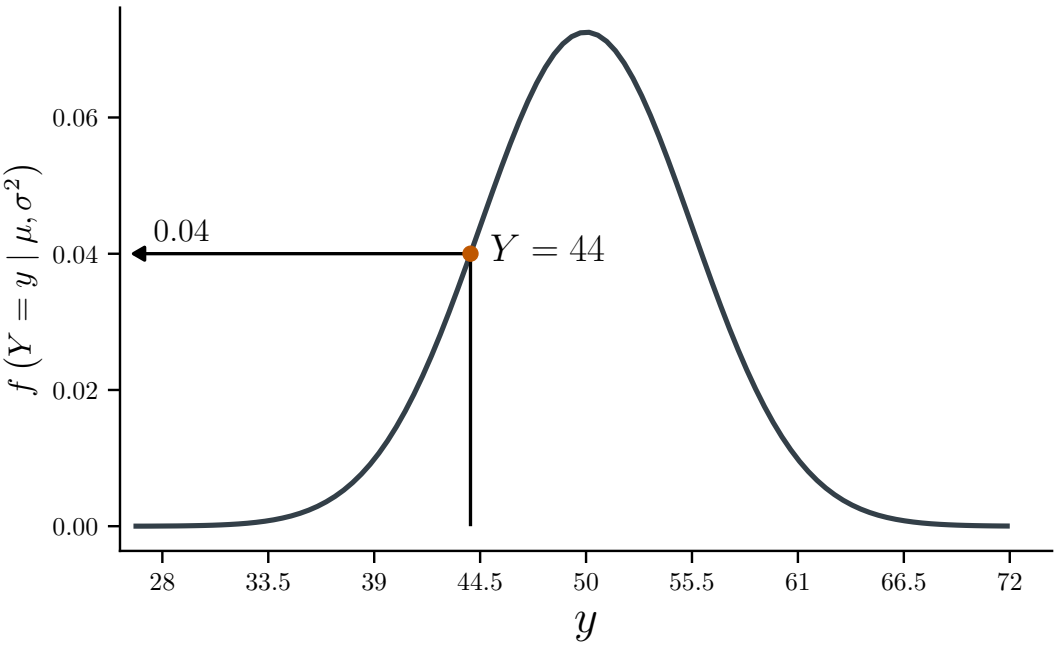
$\frac{1}{\sigma\sqrt{2\pi}}$: A scaling factor that makes the sum of all possible values for y equal one.

e: Mathematical constant known as Euler's number, it is approximately equal to 2.71828.

Univariate Normal PDF: $\mu = 10$ and $\sigma^2 = 4$



Univariate Normal PDF: $\mu = 10$ and $\sigma^2 = 4$



Likelihood Function with Normal Distribution

- Just like in our discrete example, we want to estimate the parameters that maximize the likelihood of our normal distribution.
- Therefore, we will treat our data as a known constant and use the normal PDF to form a likelihood function:

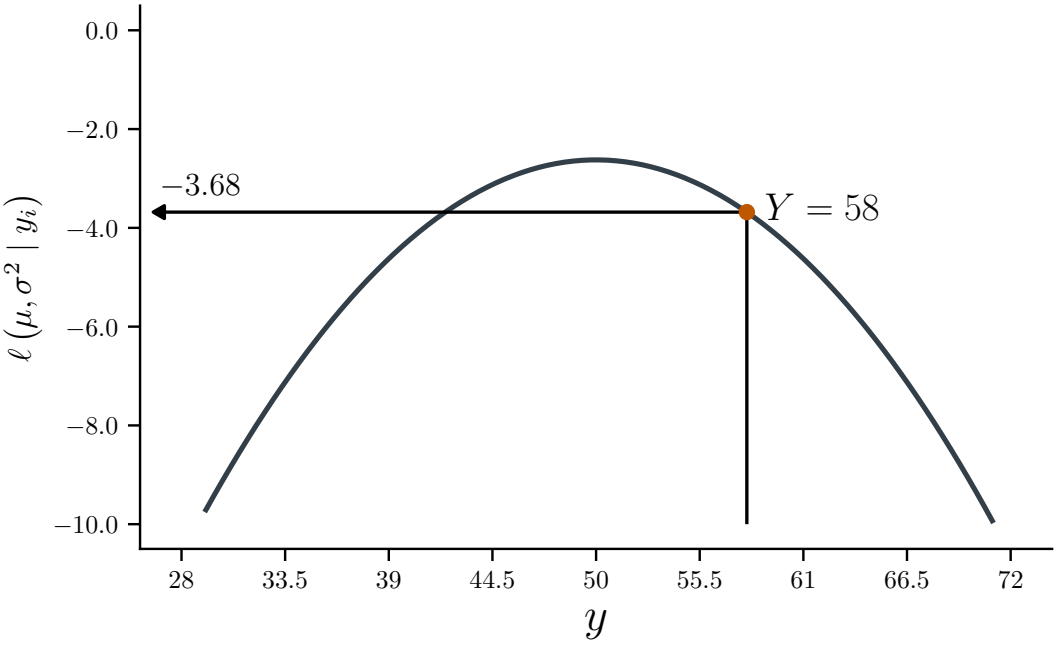
$$\begin{aligned}\mathcal{L}\left(\mu, \sigma^2 \mid Y = y\right) &= \prod_{i=1}^{N} \mathcal{L}_i\left(\mu, \sigma^2 \mid y_i\right) = \prod_{i=1}^{N} f\left(y_i \mid \mu, \sigma^2\right) \\ &= \prod_{i=1}^{N} \left(\frac{1}{\sigma\sqrt{2\pi}}\right) \exp\left\{-\frac{1}{2}\left(\frac{y_i - \mu}{\sigma}\right)^2\right\}\end{aligned}$$

Log of the Likelihood Function

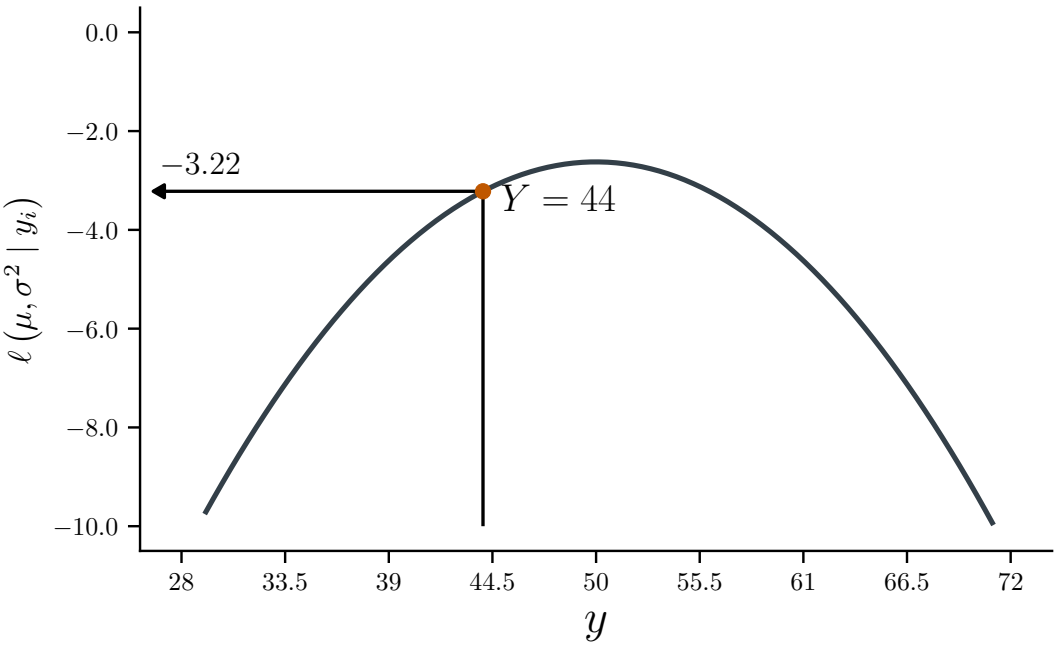
For numerical reasons (e.g., simplicity, avoiding rounding errors, etc.), estimation tends to use the natural logarithm (i.e., a logarithm with base e; we will simply use '$\ln$' to represent it) of the values values.

$$\begin{aligned}\ell\left(\mu, \sigma^2 \mid Y = y\right) &= \ln\left\{\mathcal{L}\left(\mu, \sigma^2 \mid y\right)\right\} = \ln\left\{\prod_{i=1}^{N} \mathcal{L}_i\left(\mu, \sigma^2 \mid y_i\right)\right\} \\ &= \sum_{i=1}^{N} \ln\left\{\mathcal{L}_i\left(\mu, \sigma^2 \mid y_i\right)\right\} = \sum_{i=1}^{N} \ln\left\{f\left(y_i \mid \mu, \sigma^2\right)\right\} \\ &= -N \times \ln\left\{\sigma\sqrt{2\pi}\right\} - \frac{1}{2}\sum_{i=1}^{N}\left(\frac{y_i - \mu}{\sigma}\right)^2\end{aligned}$$

Log Likelihood Example 1: $\mu = 10$ and $\sigma^2 = 4$



Log Likelihood Example 2: $\mu = 10$ and $\sigma^2 = 4$



Log Likelihood when $\mu = 8$ and $\sigma^2 = 4$

y_i	$\mathcal{L}_i$	ℓ_i
8.0	0.1995	−1.6121
8.0	0.1995	−1.6121
12.0	0.0270	−3.6121
5.0	0.0648	−2.7371
8.0	0.1995	−1.6121
12.0	0.0270	−3.6121
11.0	0.0648	−2.7371
11.0	0.0648	−2.7371
8.0	0.1995	−1.6121
13.0	0.0088	−4.7371
$\ell\left(\mu, \sigma^2 \mid y\right) =$		−26.62

Log Likelihood when $\mu = 7$ and $\sigma^2 = 4$

y_i	$\mathcal{L}_i$	ℓ_i
8.0	0.1760	−1.7371
8.0	0.1760	−1.7371
12.0	0.0088	−4.7371
5.0	0.1210	−2.1121
8.0	0.1760	−1.7371
12.0	0.0088	−4.7371
11.0	0.0270	−3.6121
11.0	0.0270	−3.6121
8.0	0.1760	−1.7371
13.0	0.0022	−6.1121
$\ell\left(\mu, \sigma^2 \mid y\right) =$		−31.87

Fit Worsened

Log Likelihood when $\mu = 10$ and $\sigma^2 = 4$

y_i	$\mathcal{L}_i$	ℓ_i
8.0	0.1210	−2.1121
8.0	0.1210	−2.1121
12.0	0.1210	−2.1121
5.0	0.0088	−4.7371
8.0	0.1210	−2.1121
12.0	0.1210	−2.1121
11.0	0.1760	−1.7371
11.0	0.1760	−1.7371
8.0	0.1210	−2.1121
13.0	0.0648	−2.7371
$\ell\left(\mu, \sigma^2 \mid y\right) =$		−23.62

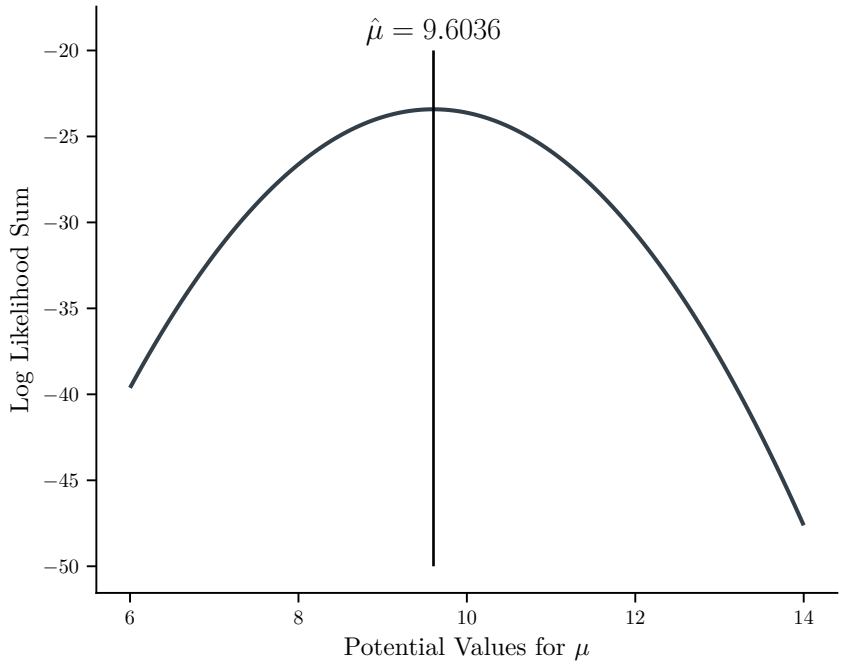
Fit Improved

Log Likelihood when $\mu = 11$ and $\sigma^2 = 4$

y_i	$\mathcal{L}_i$	ℓ_i
8.0	0.0648	−2.7371
8.0	0.0648	−2.7371
12.0	0.1760	−1.7371
5.0	0.0022	−6.1121
8.0	0.0648	−2.7371
12.0	0.1760	−1.7371
11.0	0.1995	−1.6121
11.0	0.1995	−1.6121
8.0	0.0648	−2.7371
13.0	0.1210	−2.1121
$\ell\left(\mu, \sigma^2 \mid y\right) =$		−25.87

Fit Worsened

Graphing $\ell\left(\mu, \sigma^2 \mid y\right)$ dependent on values of μ



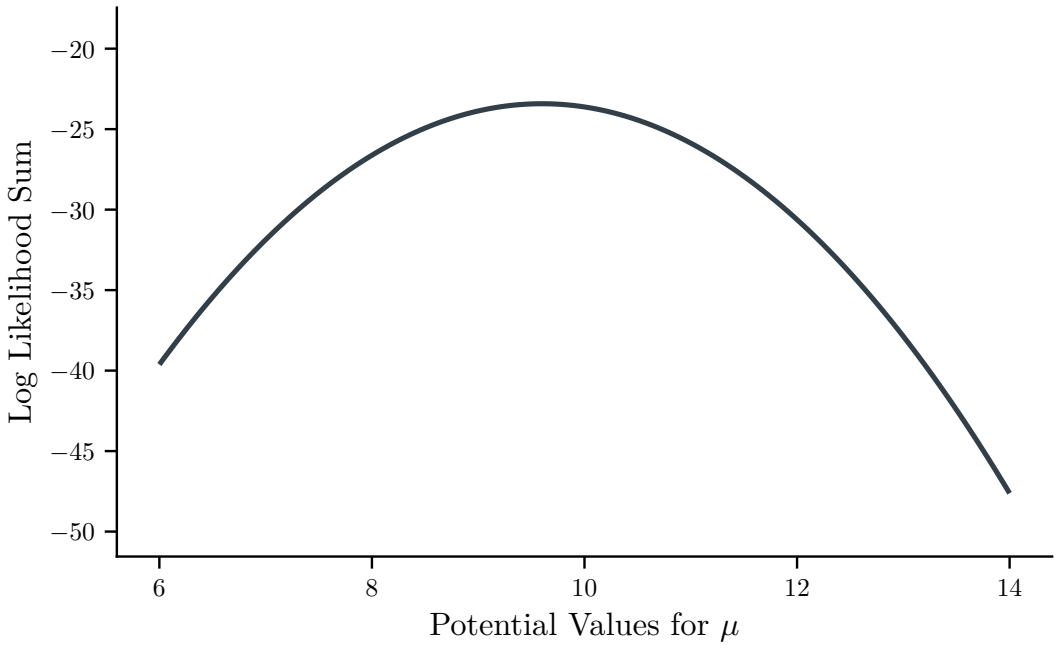
μ	$\ell\left(\mu, \sigma^2 \mid y\right)$
9.57958	−23.42138
9.58759	−23.42105
9.59560	−23.42088
9.60360	−23.42087
9.61161	−23.42103
9.61962	−23.42134
9.62763	−23.42181

Estimating Unknown Parameters

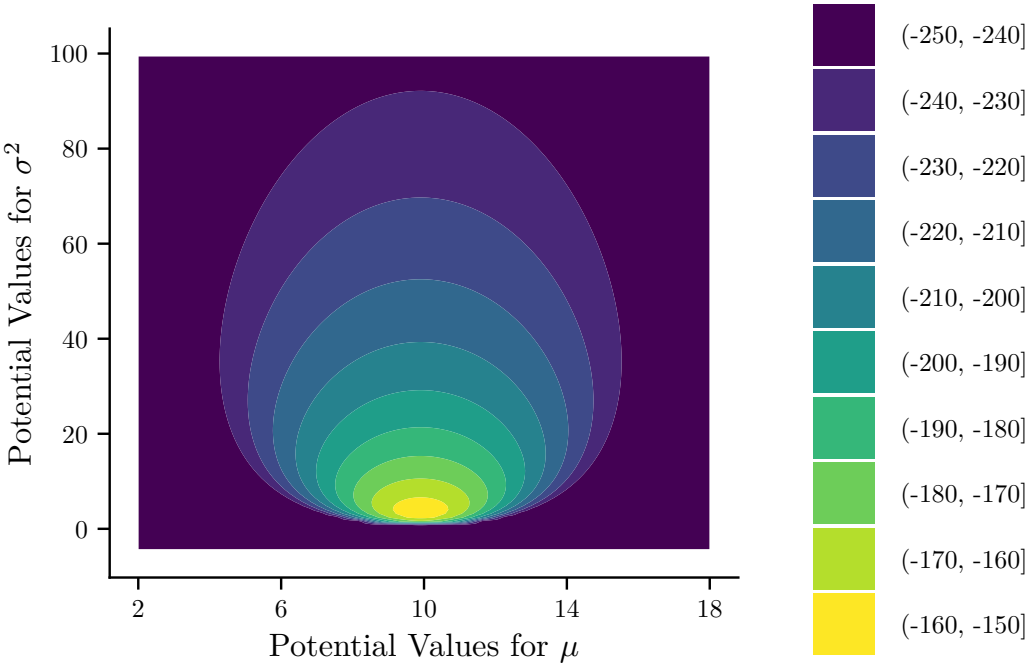
Estimating Unknown Parameters

- In Maximum Likelihood, the goal of estimation is to find the parameter values that maximize the (log) likelihood function for all the observed data.
- This can be thought of as finding the peak of a multidimensional surface. Each unknown parameter adds another dimensions to this surface.

Univariate Normal Likelihood – One Dimension



Univariate Normal Likelihood – Two Dimensions



Finding the Maximum of the ℓ

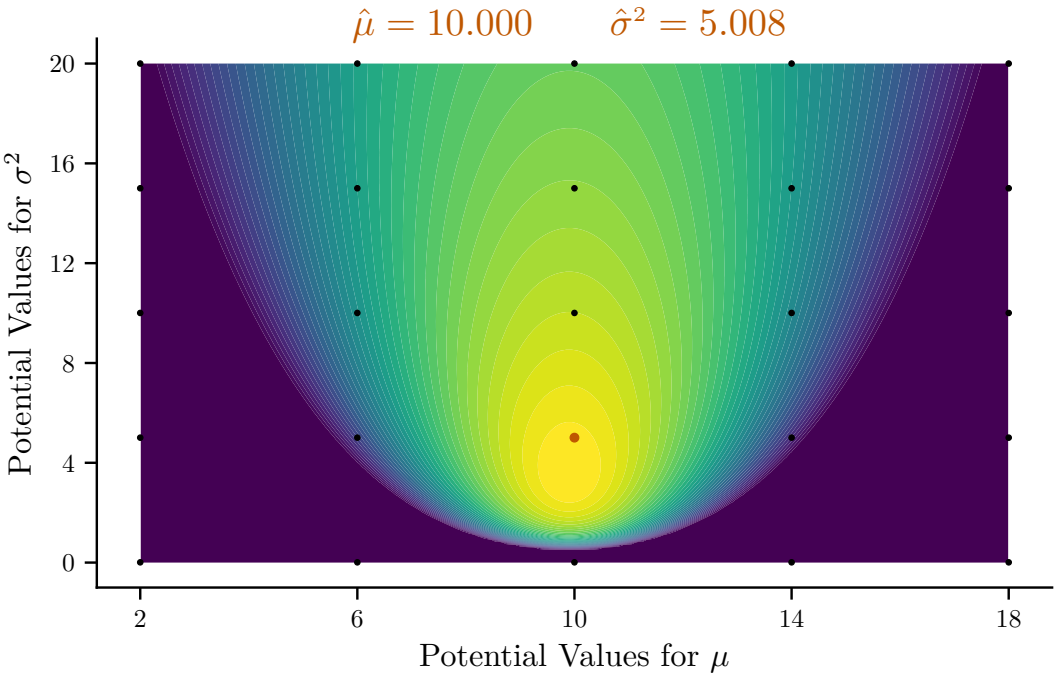
There are three main ways to find the estimates that maximize the likelihood:

- Grid Search and Monte Carlo Search
- Analytical Solution
- Optimization Algorithms

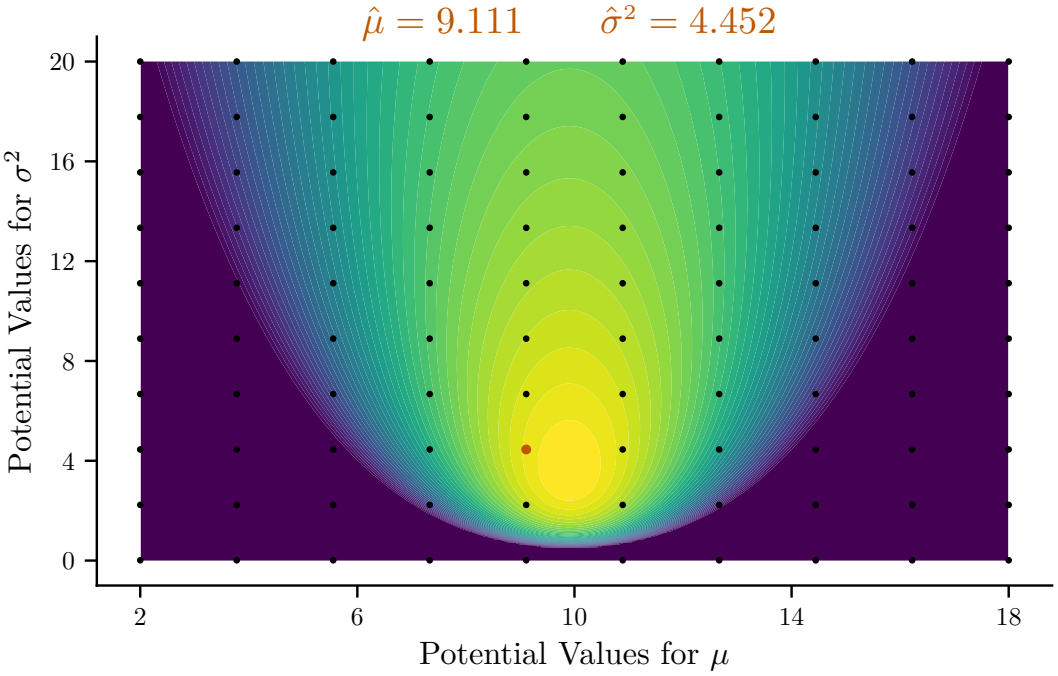
Grid Search

- The grid search method is in essence to brute force the solution.
- In theory, if we were to try all combination of parameters we could find the group of parameters that produces the maximum estimate.
- This is impractical for more than toy examples and usually not done.

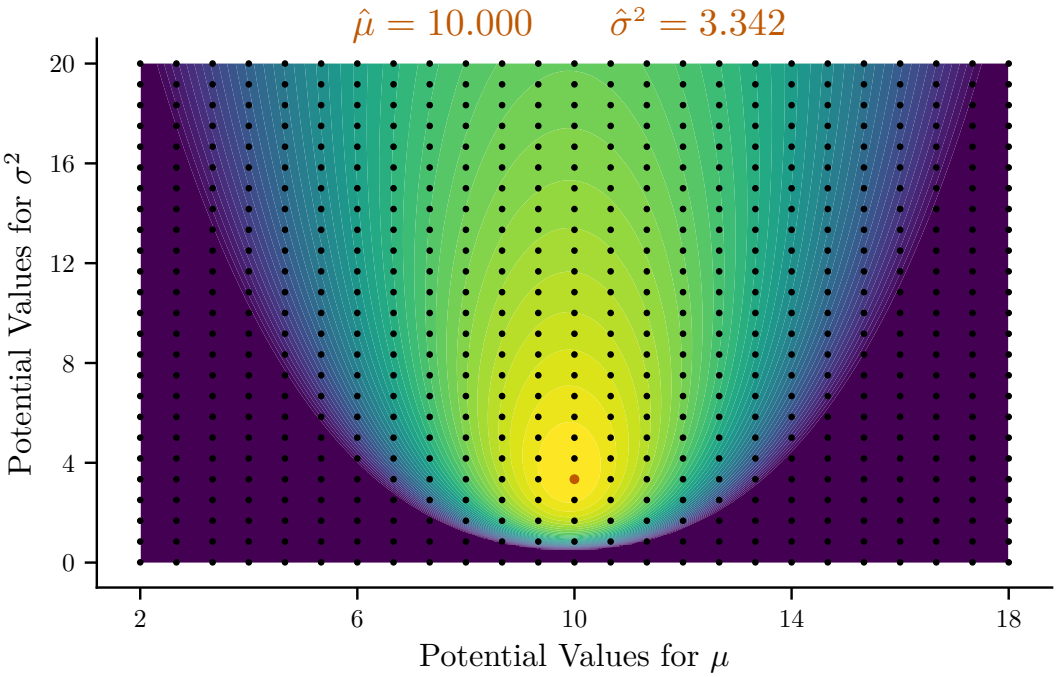
Grid Search – 5 by 5



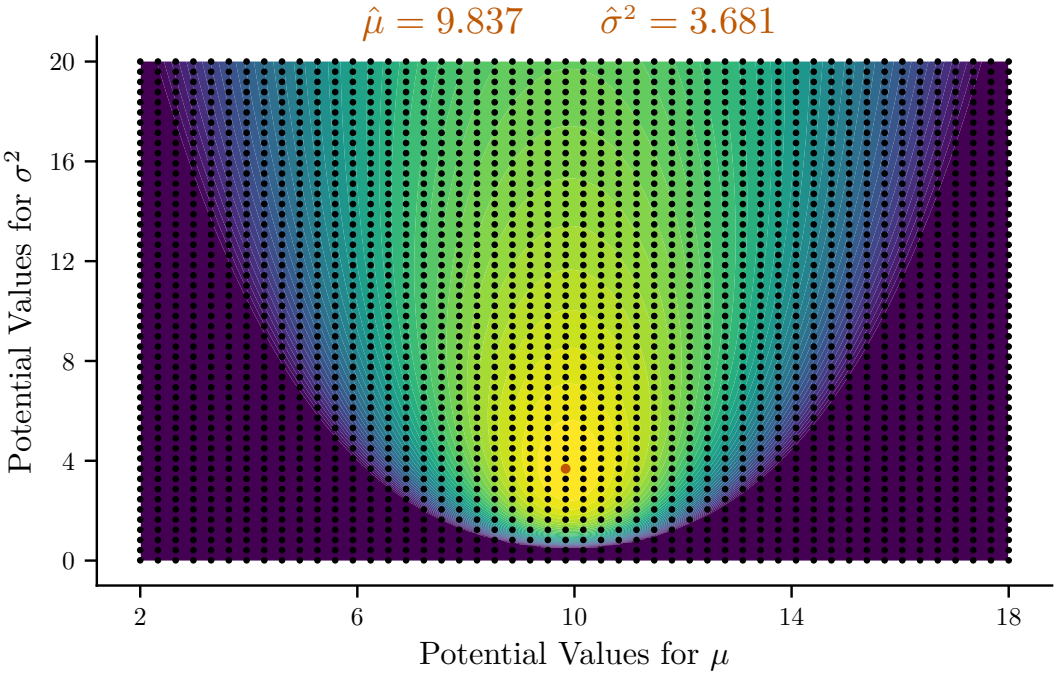
Grid Search – 10 by 10



Grid Search – 25 by 25



Grid Search – 50 by 50



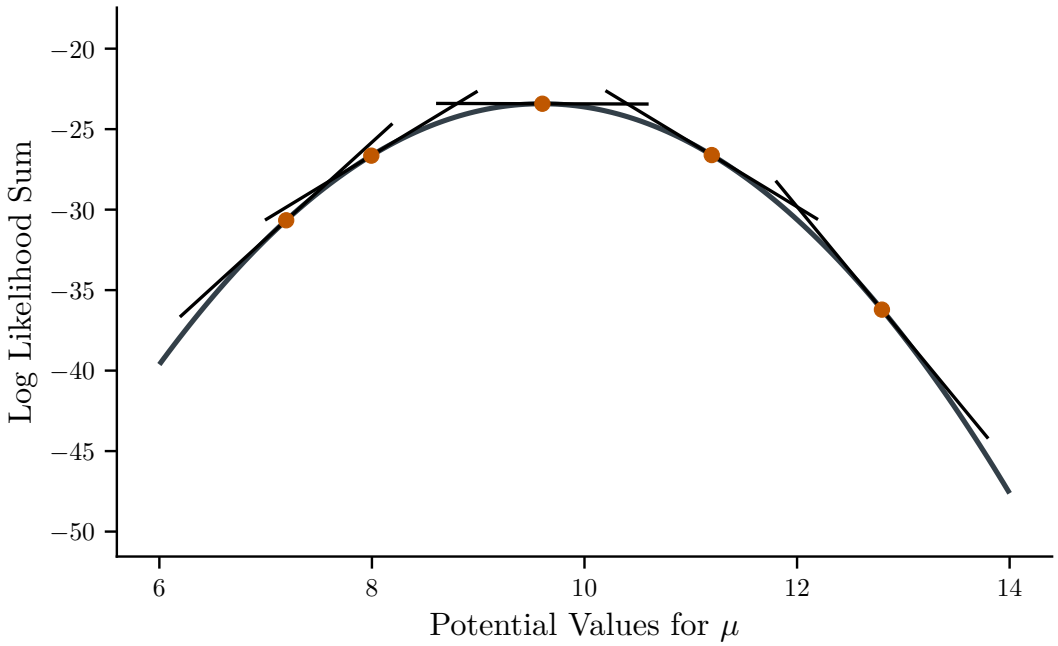
Analytic Solution

Deriving Analytic Solutions

- Differential calculus rules allow us to find the first derivative (i.e., slope of the tangent line at any point on the log likelihood function) for the parameter's current values.
- One way to find the (local) maximum of a monotonic function (which the log likelihood is), is to solve when the first derivative is equal to zero.

$$\frac{\partial \ell \left(\theta \mid y \right)}{\partial \theta} = 0$$

Tangent Lines on the Log Likelihood



Univariate Normal Example – Partial Derivatives

For our univariate normal example, we have two partial derivatives and need to find the solution when both are equal to zero.

$$\frac{\partial \ell}{\partial \mu} = \sigma^{-2} \sum_{i=1}^{N} (y_i - \mu)$$

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_{i=1}^{N} (y_i - \mu)^2$$

Solving for μ

$$0 = \frac{\partial \ell}{\partial \mu}$$

$$0 = \sigma^{-2} \sum_{i=1}^{N} (y_i - \mu)$$

$$\frac{0}{\sigma^{-2}} = \sum_{i=1}^{N} (y_i - \mu) = \sum_{i=1}^{N} (y_i) - \sum_{i=1}^{N} (\mu)$$

$$0 = -N\mu + \sum_{i=1}^{N} (y_i)$$

$$\mu = \frac{1}{N} \sum_{i=1}^{N} (y_i)$$

Solving for σ^2

$$0 = \frac{\partial \ell}{\partial \sigma^2}$$

$$0 = -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{N}(y_i - \mu)^2$$

$$\frac{N}{2\sigma^2} = \frac{1}{2\sigma^4}\sum_{i=1}^{N}(y_i - \mu)^2$$

$$\frac{2N\sigma^4}{2\sigma^2} = \sum_{i=1}^{N}(y_i - \mu)^2$$

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mu)^2$$

Introduction to Optimization Algorithms

Optimization Algorithms

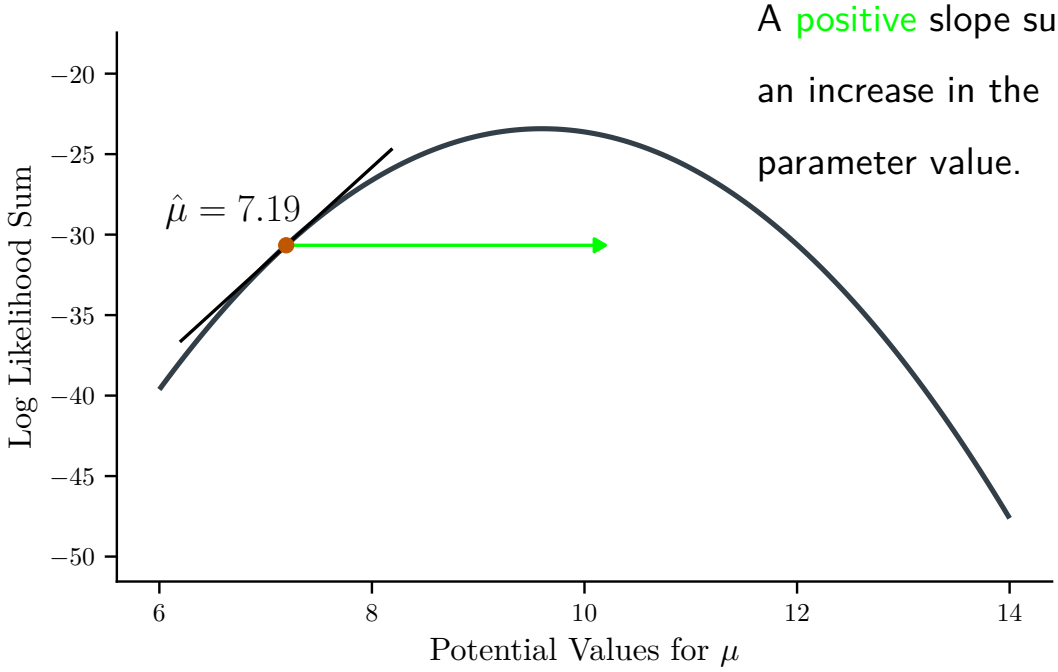
- For more complicated models, analytic solutions are not available (i.e., no close form solution) and must be estimated. This will be the case for models with missing data.
- There are several optimization algorithms that can be used to solve this issue. We will discuss a few of them.

Using the First Derivative

- Recall the first derivative of the log likelihood is the slope of the tangent line at any point on the log likelihood function at a parameter's value.
- We can use an optimization procedure to try values and determine which direction we need to move in order to find the maximum (i.e., when slope is equal to 0).
- The sign and magnitude of the slope determines the upward or downward adjustment to the next parameters from one iteration to the next.

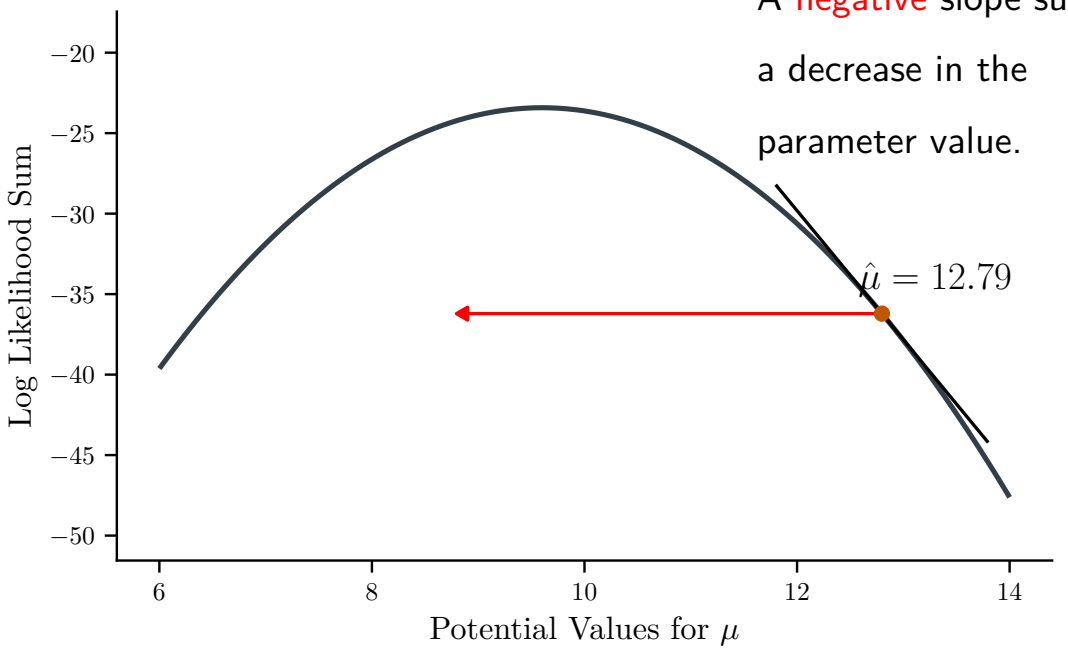
Graphical Illustration: Slope of Tangent Line

A positive slope suggests an increase in the parameter value.



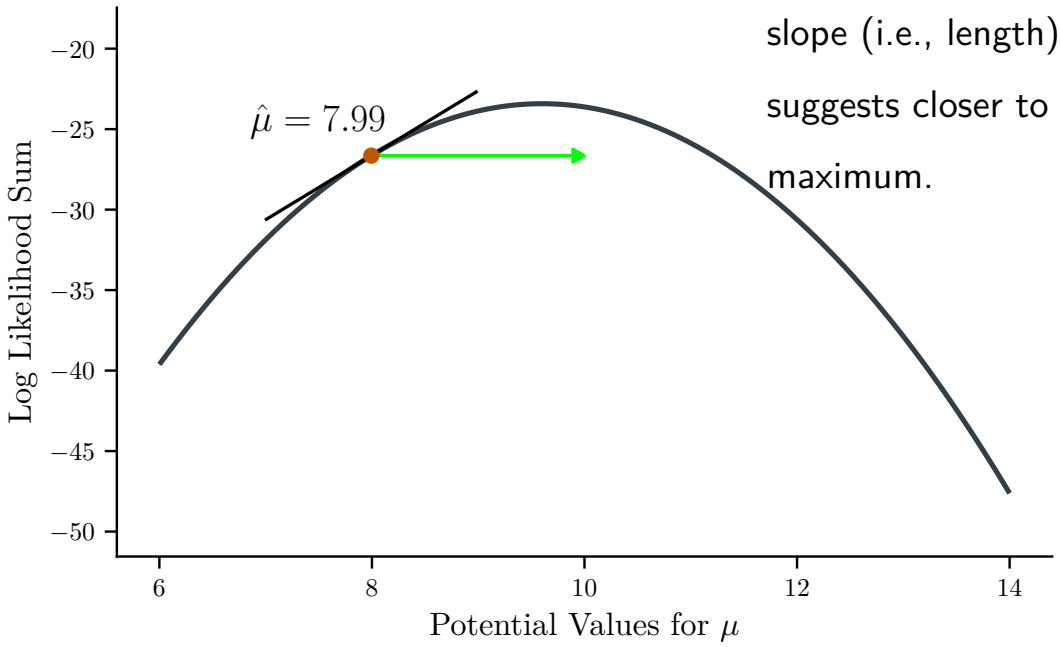
Graphical Illustration: Slope of Tangent Line

A negative slope suggests a decrease in the parameter value.

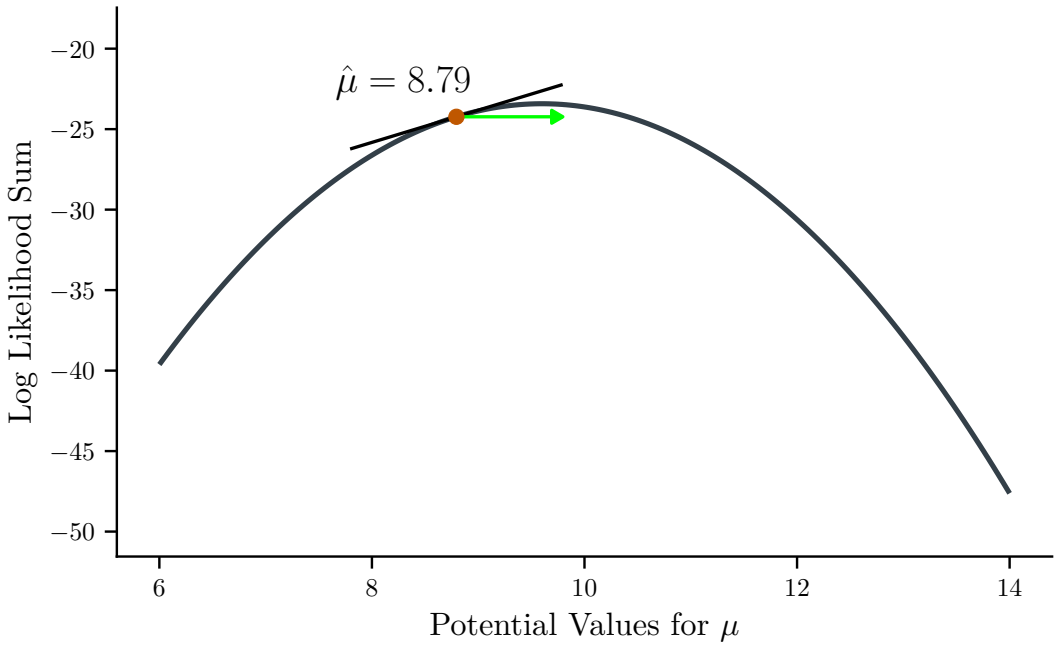


Graphical Illustration: Slope of Tangent Line

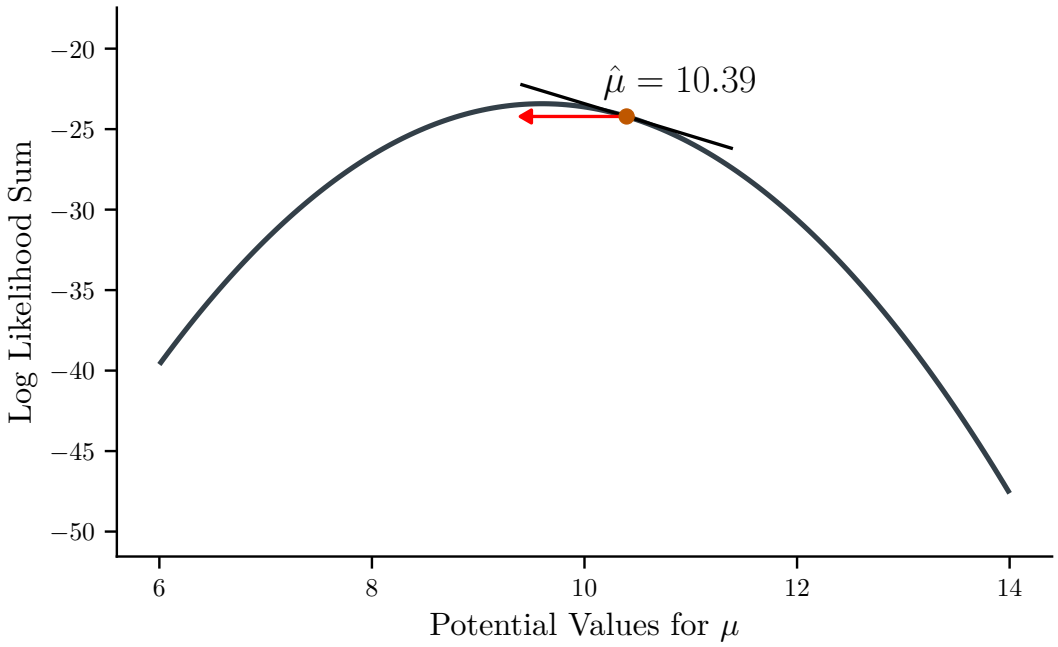
A smaller magnitude of slope (i.e., length) suggests closer to maximum.



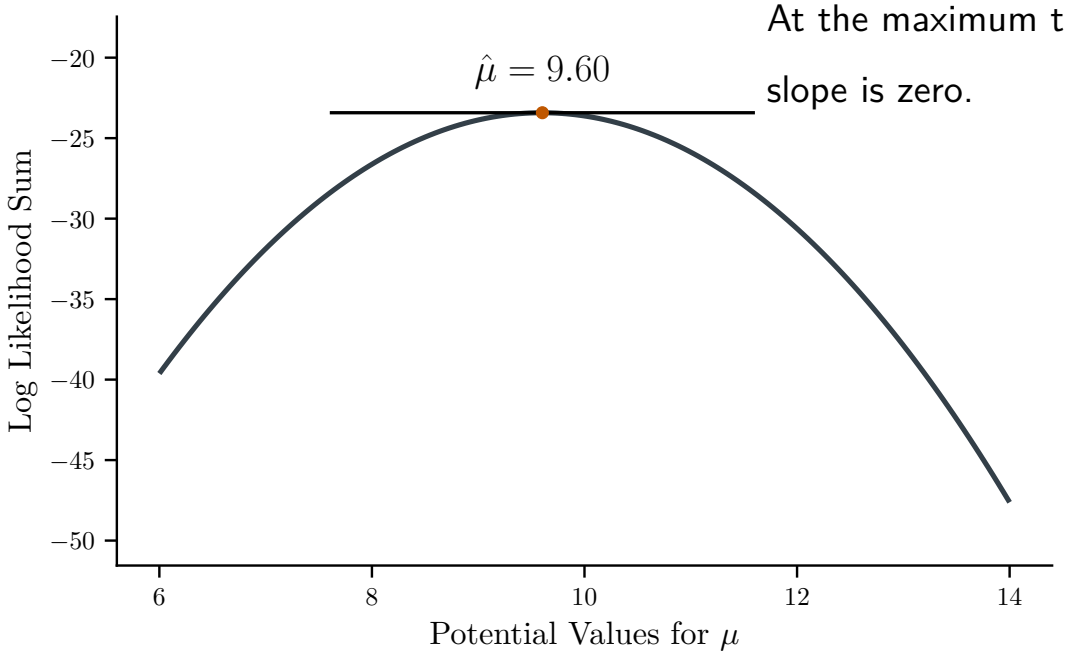
Graphical Illustration: Slope of Tangent Line



Graphical Illustration: Slope of Tangent Line



Maximum of the Log Likelihood



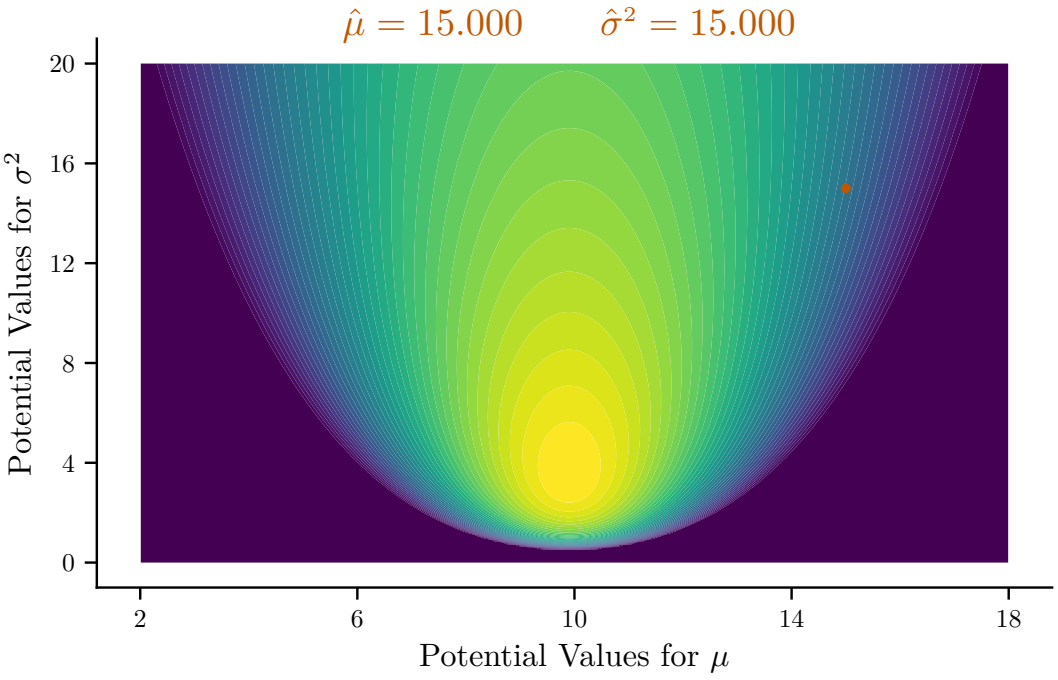
At the maximum the slope is zero.

Gradient Ascent (Descent)

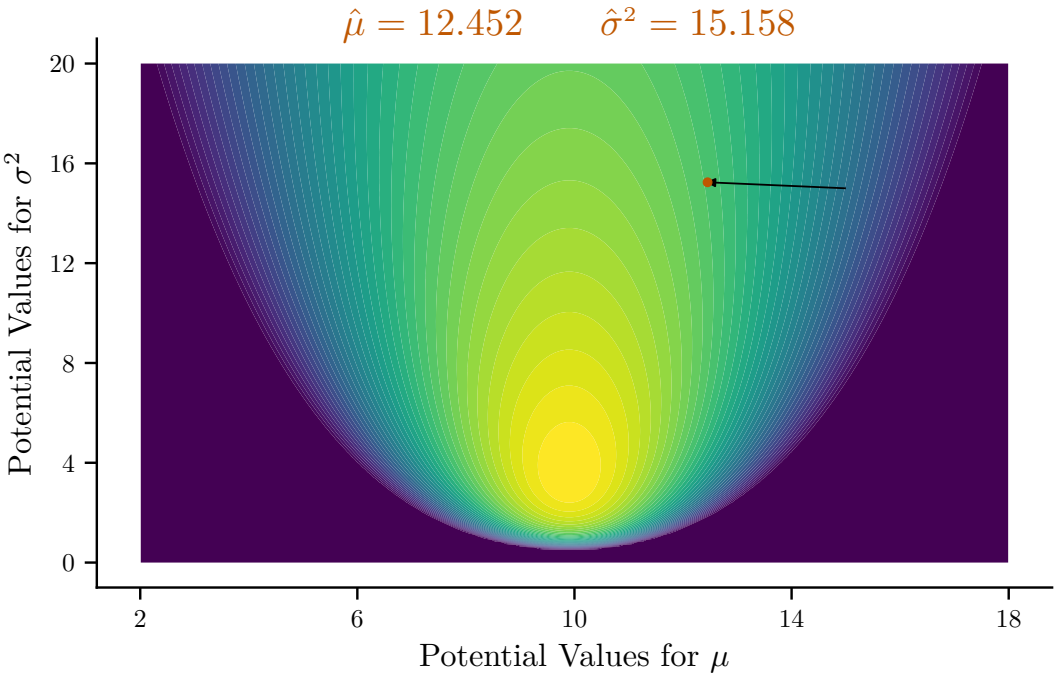
- More generally, this algorithm is known as gradient ascent (the minimization version is called gradient descent).
- Gradient ascent takes the first derivative of the log likelihood with respect to the parameter vector (i.e., the gradient or score vector) and multiplies it by some constant (the learning rate γ) to update the current parameter values.

$$\theta^{(t)} = \theta^{(t-1)} + \gamma \frac{\partial \ell}{\partial \theta}$$

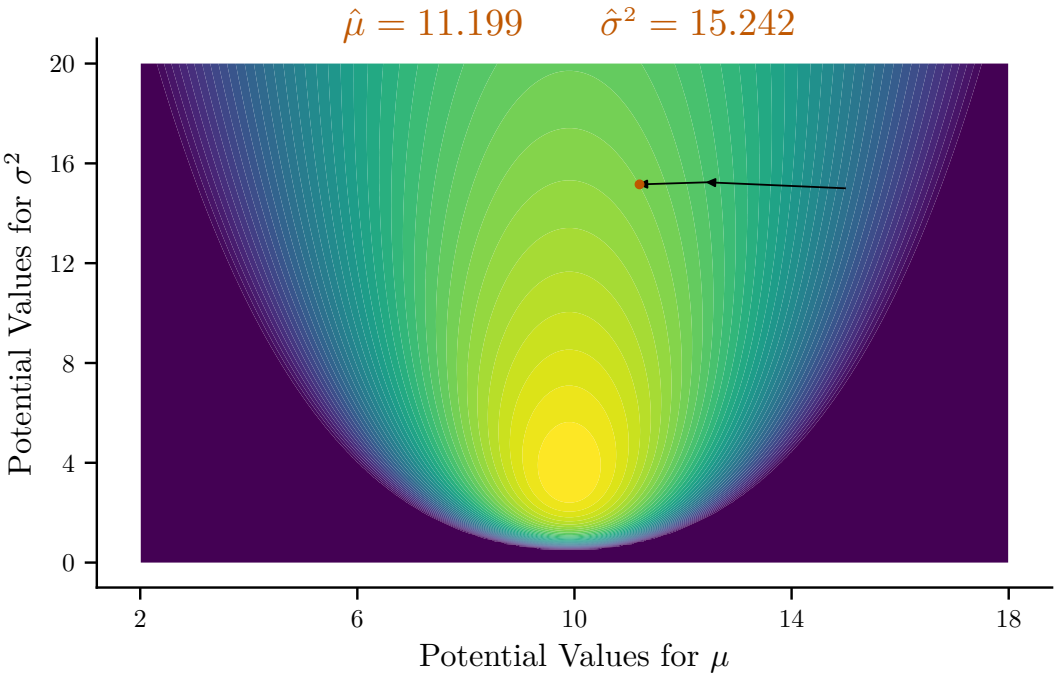
Gradient Ascent – Iteration 0 (Starting Values)



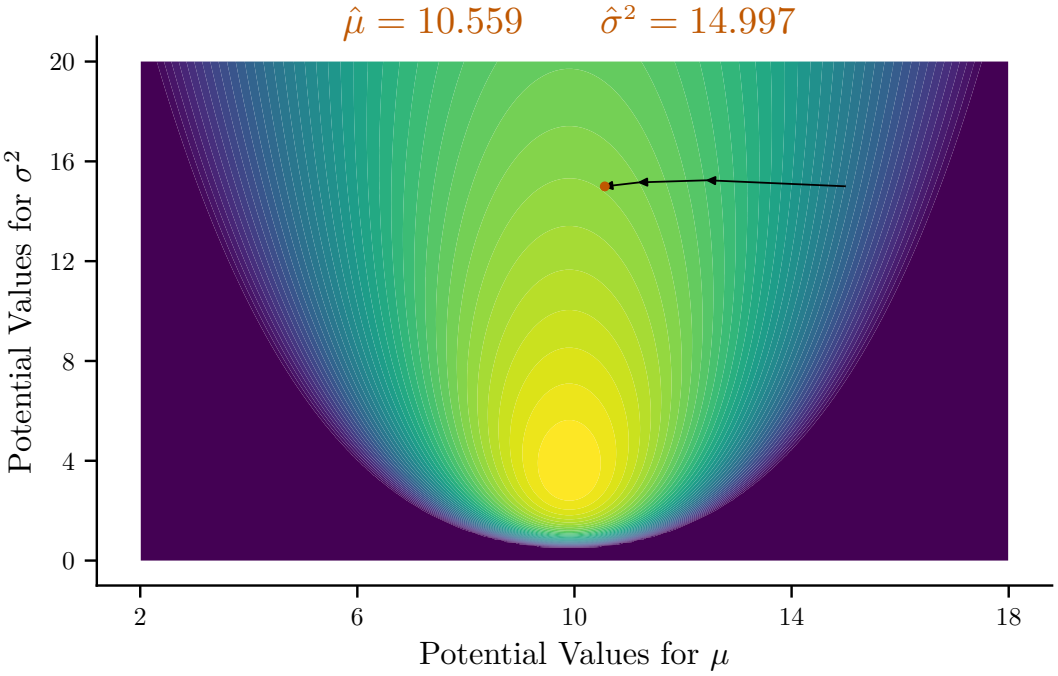
Gradient Ascent – Iteration 1



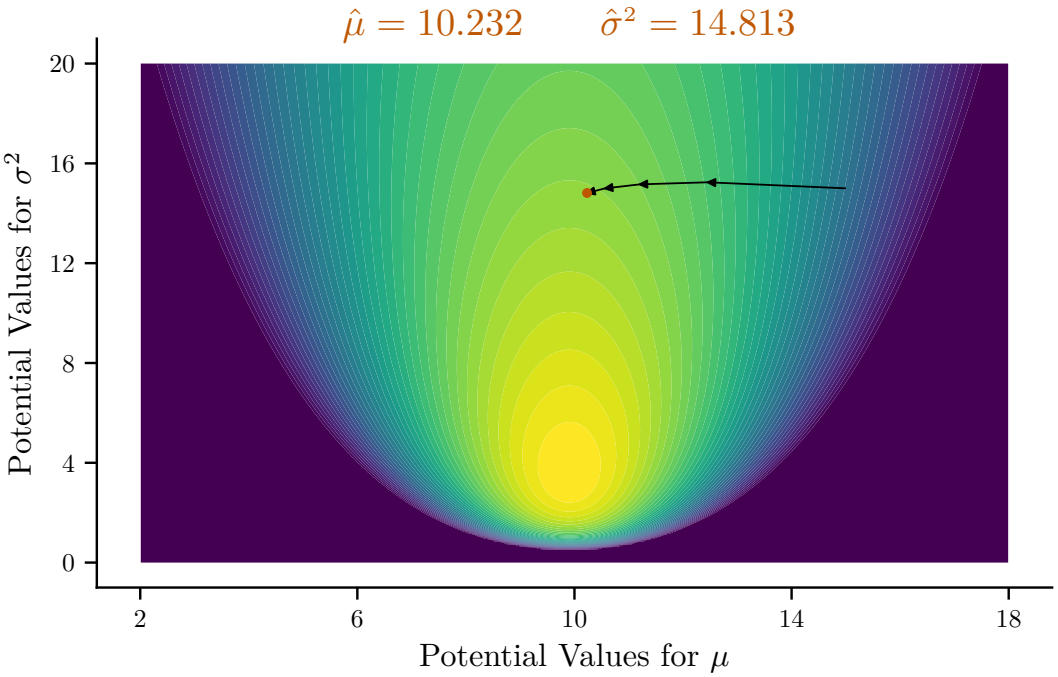
Gradient Ascent – Iteration 2



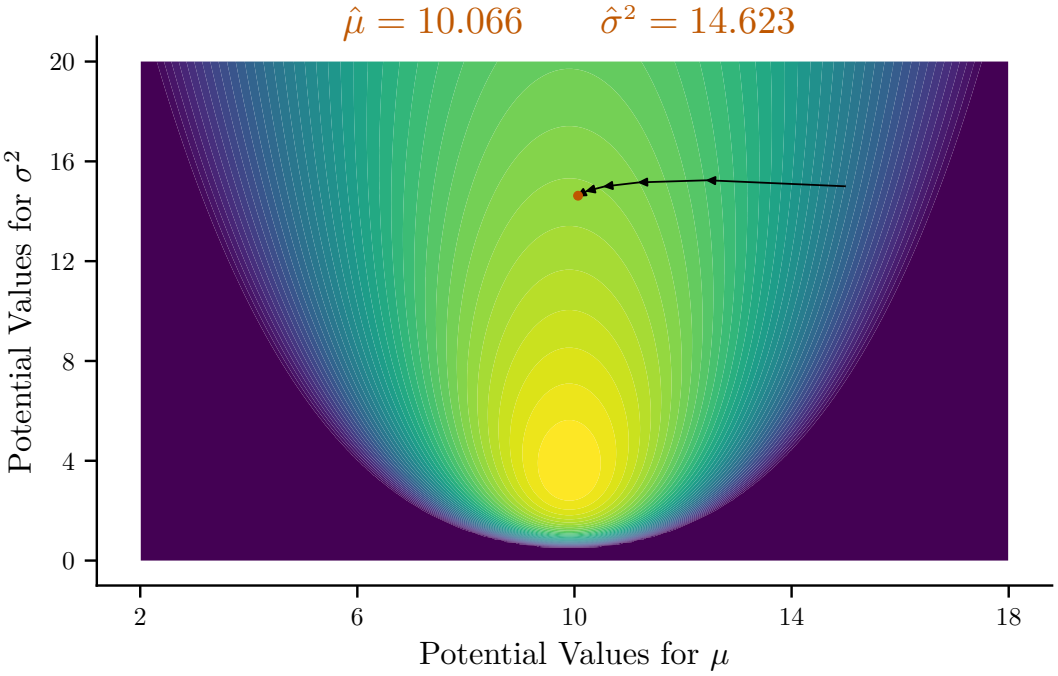
Gradient Ascent – Iteration 3



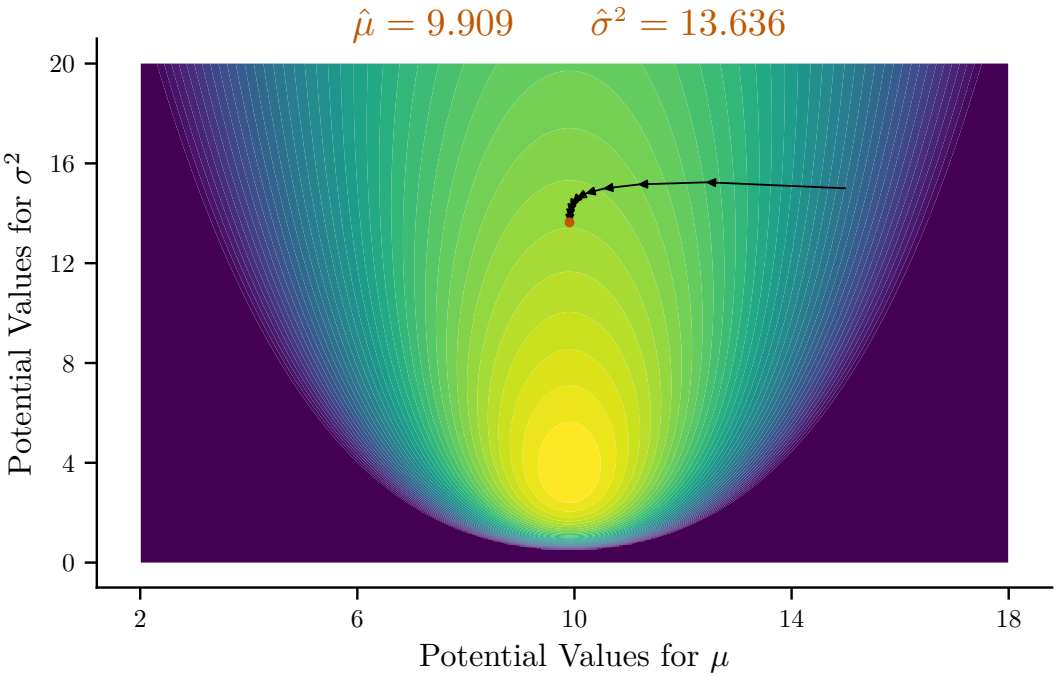
Gradient Ascent – Iteration 4



Gradient Ascent – Iteration 5

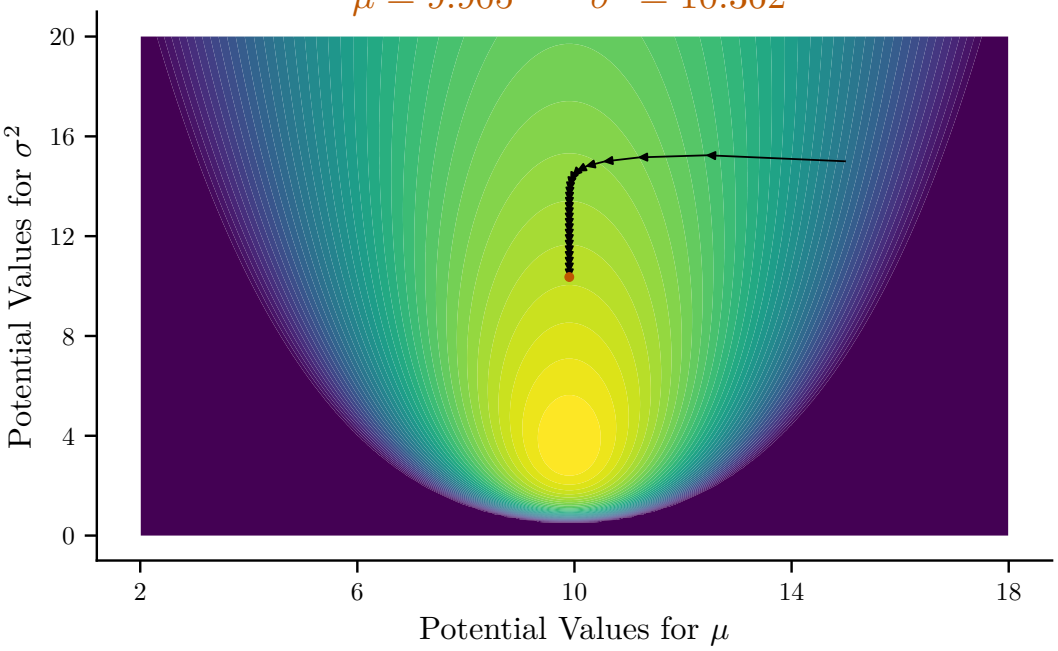


Gradient Ascent – Iteration 10



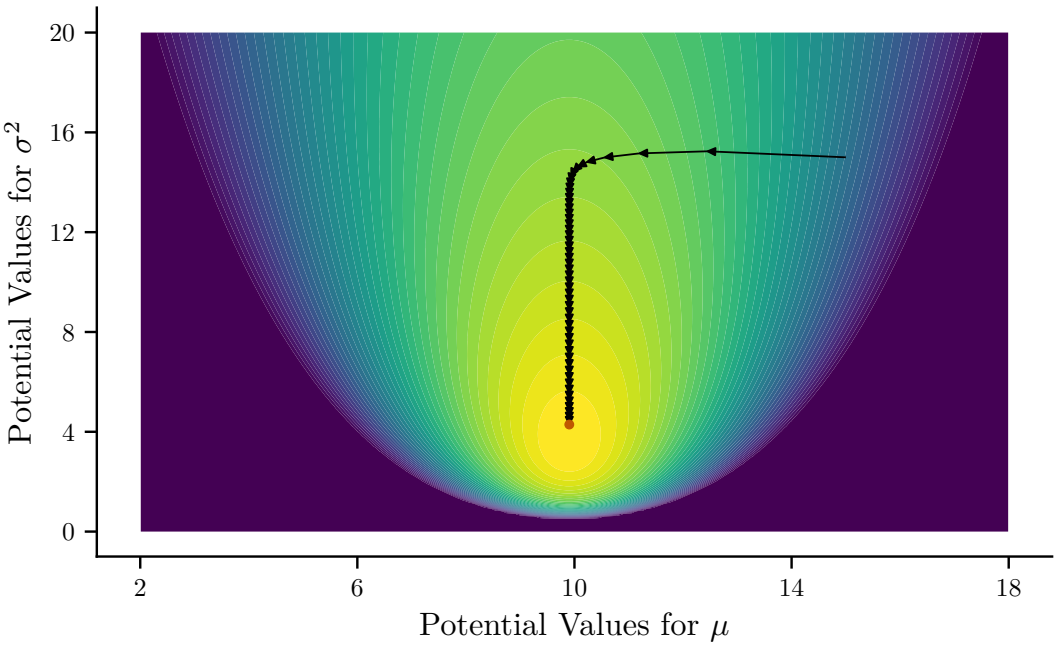
Gradient Ascent – Iteration 25

$\hat{\mu} = 9.905 \qquad \hat{\sigma}^2 = 10.362$

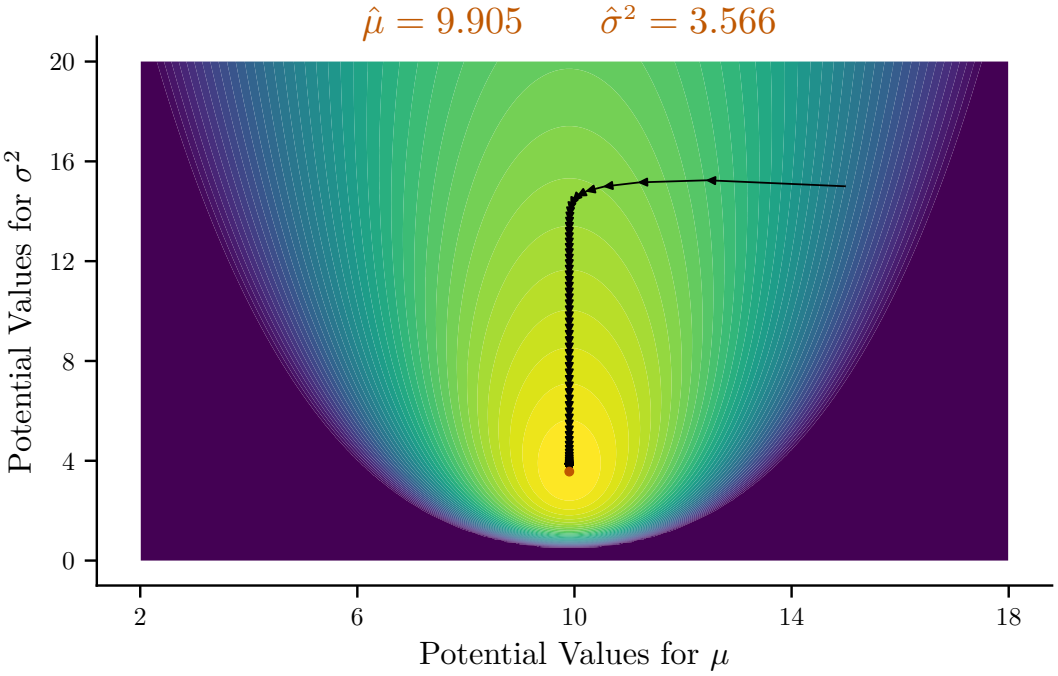


Gradient Ascent – Iteration 50

$\hat{\mu} = 9.905 \qquad \hat{\sigma}^2 = 4.291$



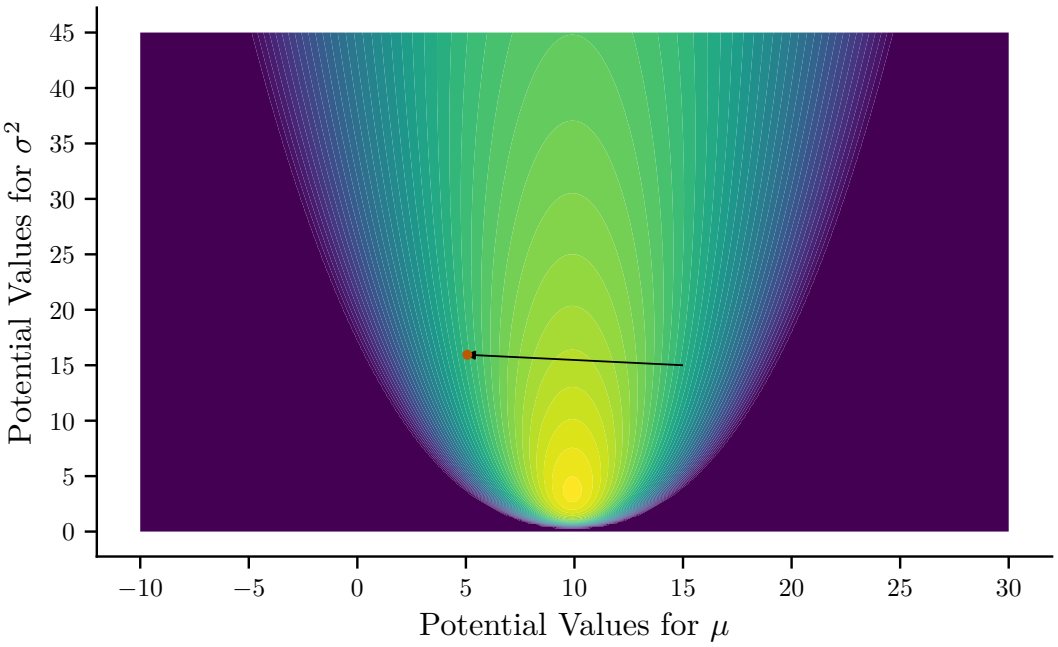
Gradient Ascent – Iteration 100



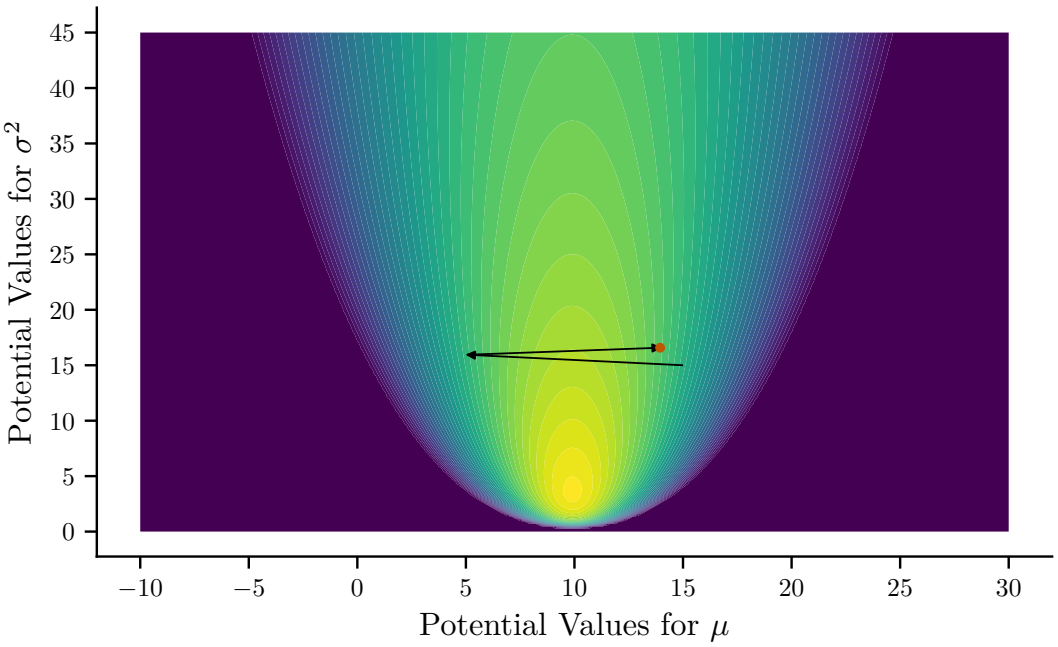
Gradient Ascent Comments

- Gradient descent/ascent is a large class of algorithms.
- Specialized forms have been developed for setting and determining optimal learning rates.
- If the learning parameter is not optimally tuned, issues can arise.
- It can be sensitive to the scale of the dimensions (i.e., it works better when all parameters are in similar ranges).

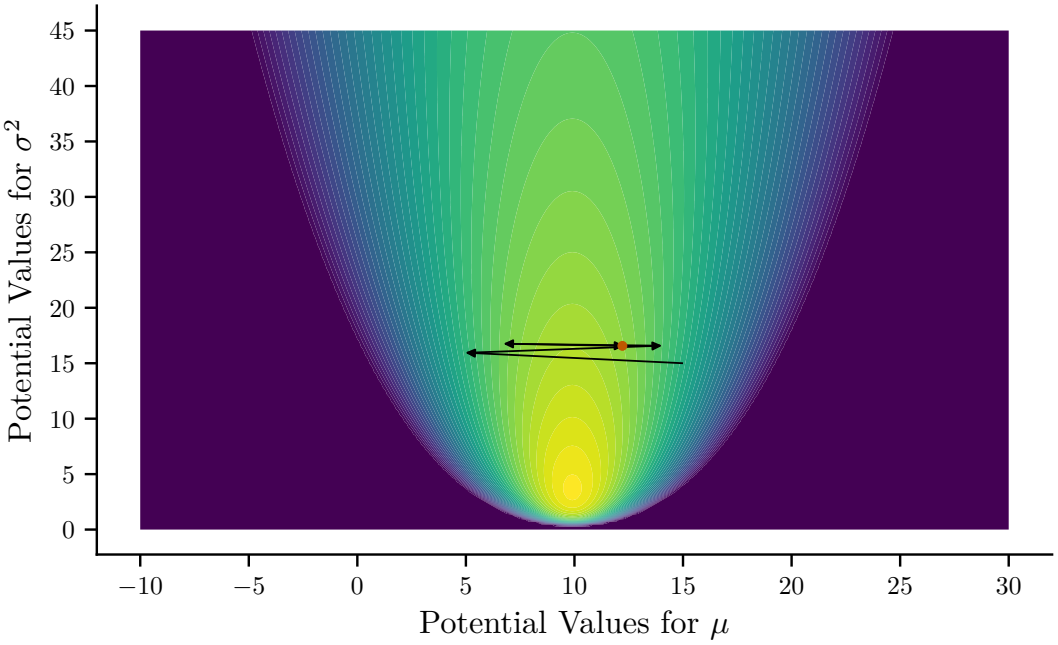
Problem Gradient Ascent – Iteration 1



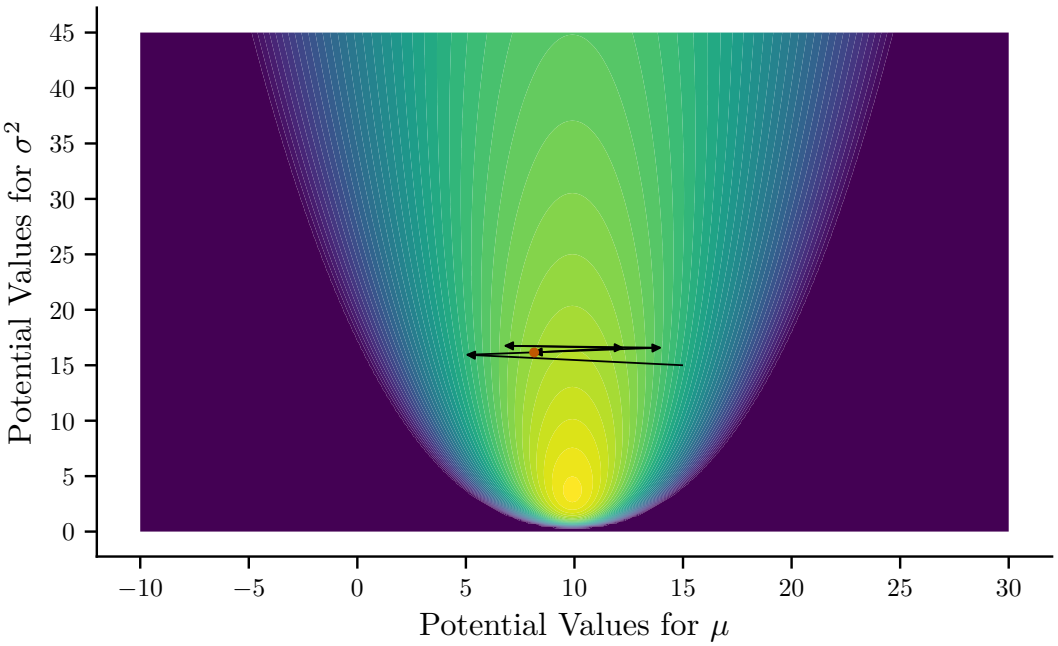
Problem Gradient Ascent – Iteration 2



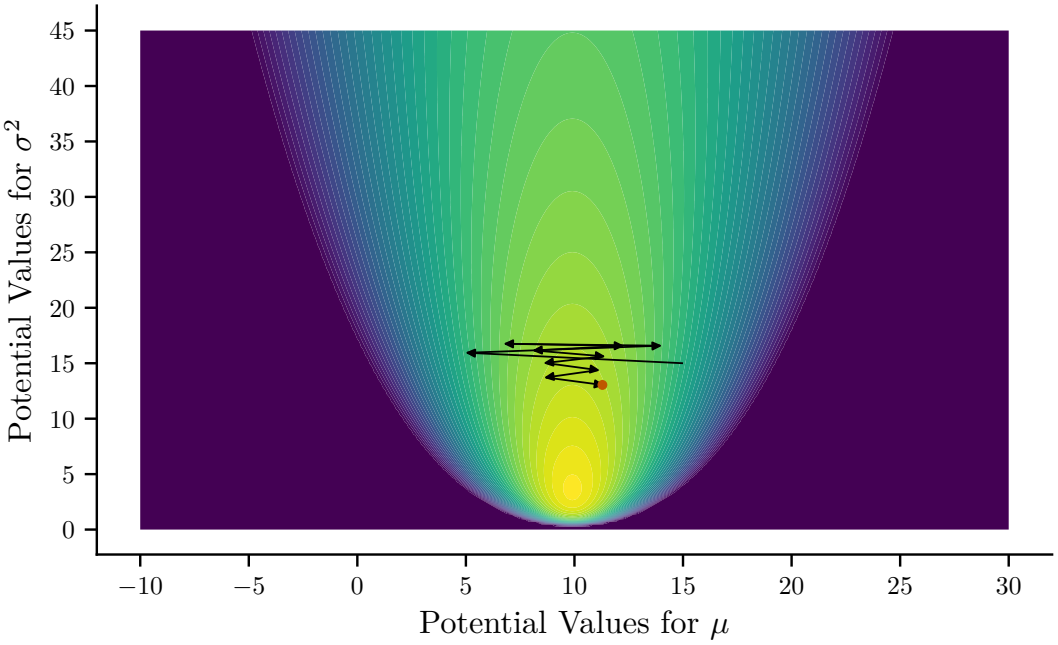
Problem Gradient Ascent – Iteration 4



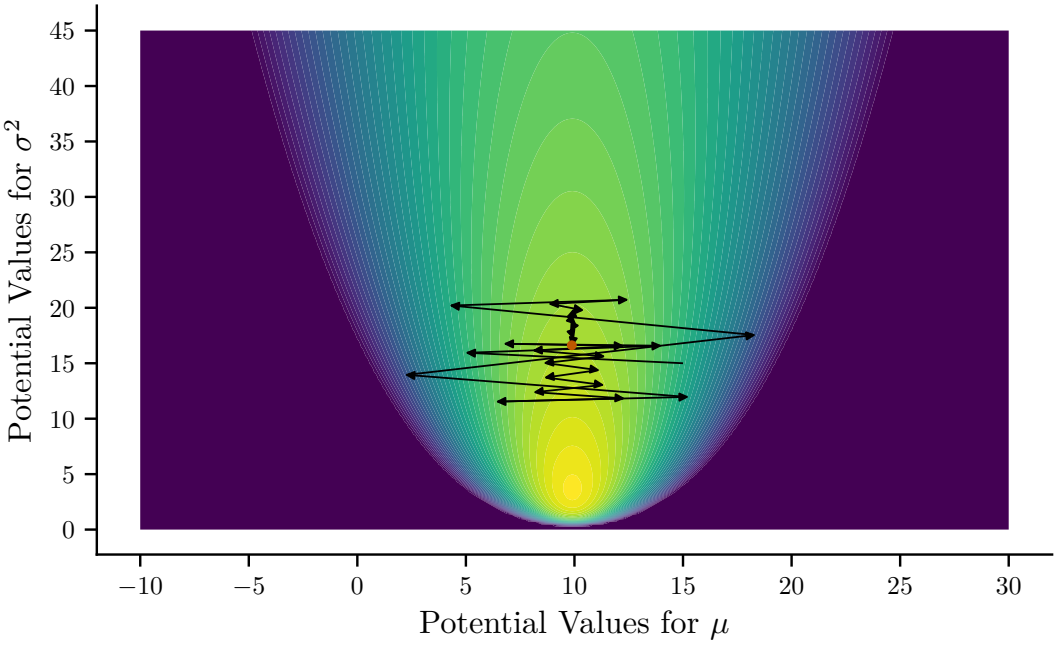
Problem Gradient Ascent – Iteration 5



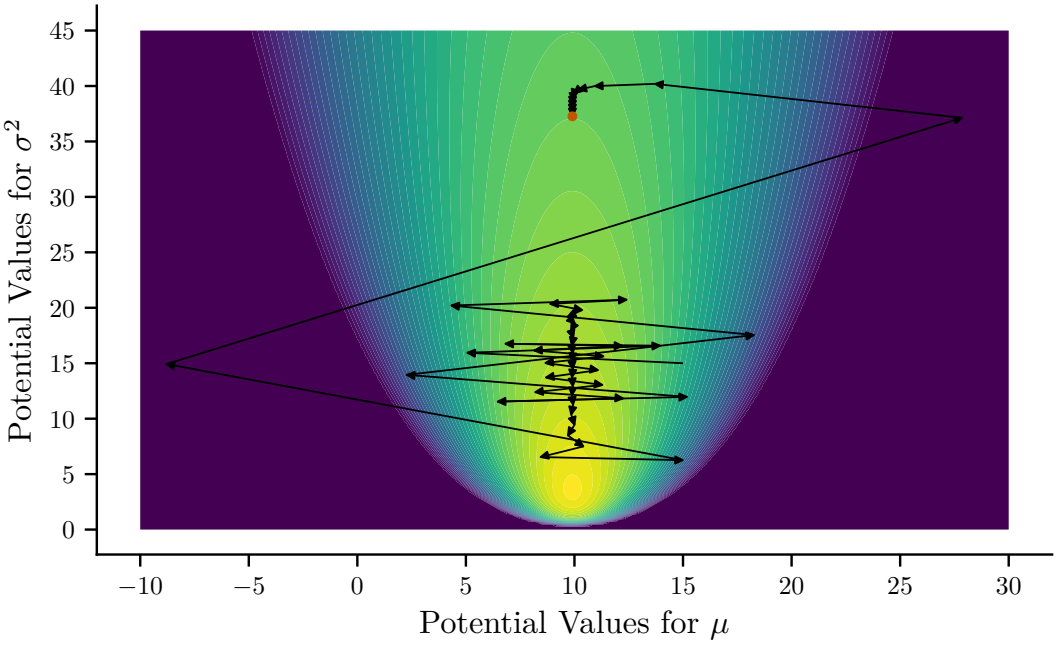
Problem Gradient Ascent – Iteration 10



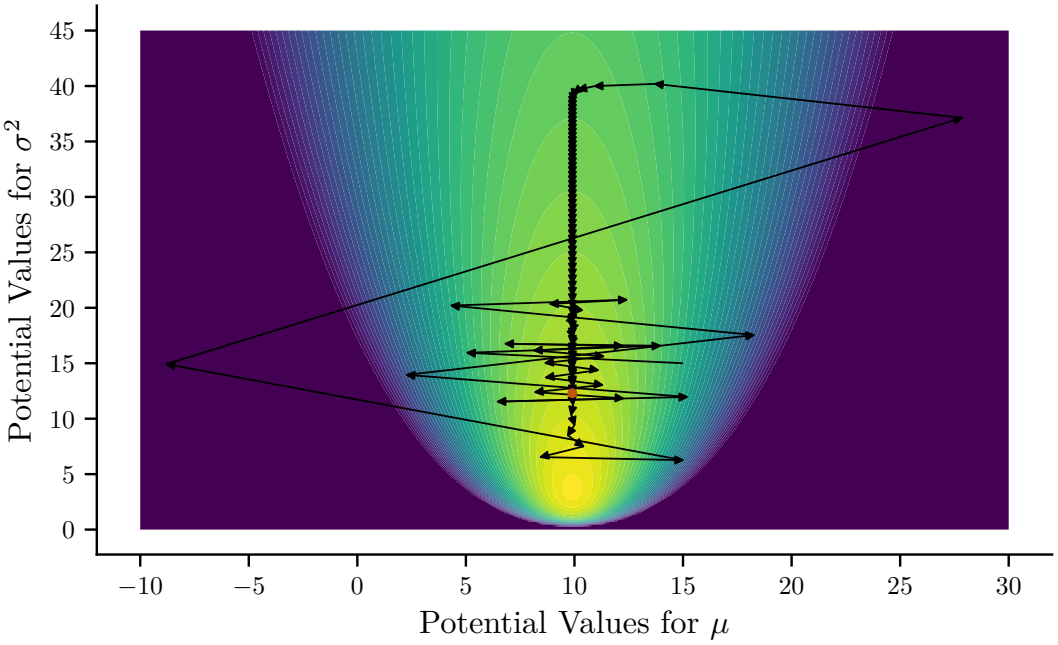
Problem Gradient Ascent – Iteration 25



Problem Gradient Ascent – Iteration 50



Problem Gradient Ascent – Iteration 100



Newton-Raphson Methods

Newton-Raphson Method

- Newton's method is a method for finding roots (i.e., values of function inputs that result in zero). This also works form maximization because we want to find the root of the first derivative.

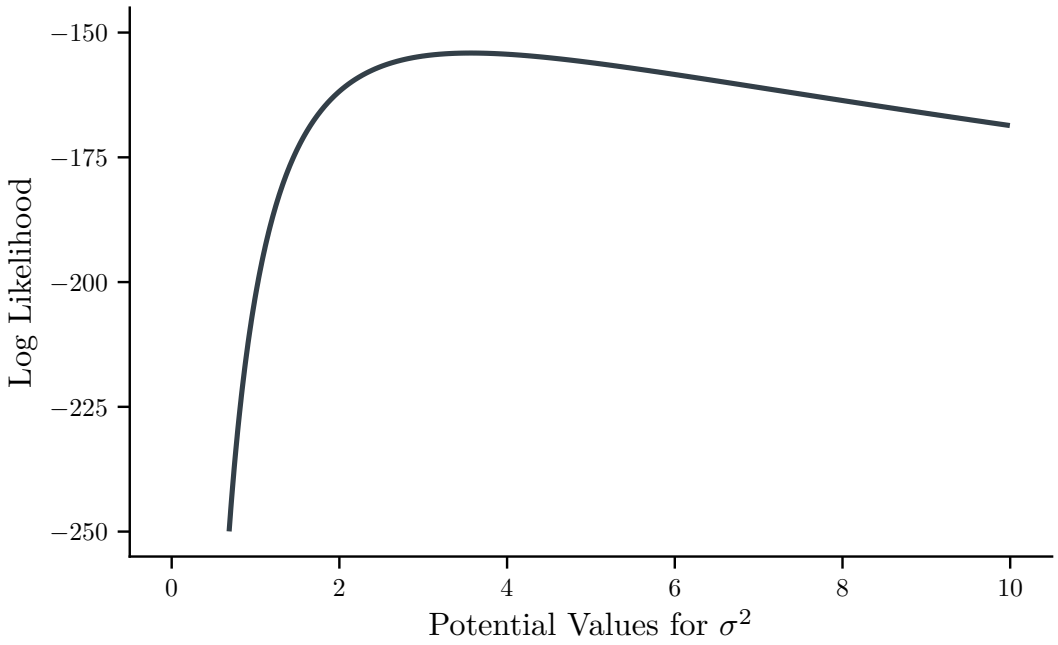
$$0 = \frac{\partial \ell}{\partial \theta}$$

Newton-Raphson Method Continued...

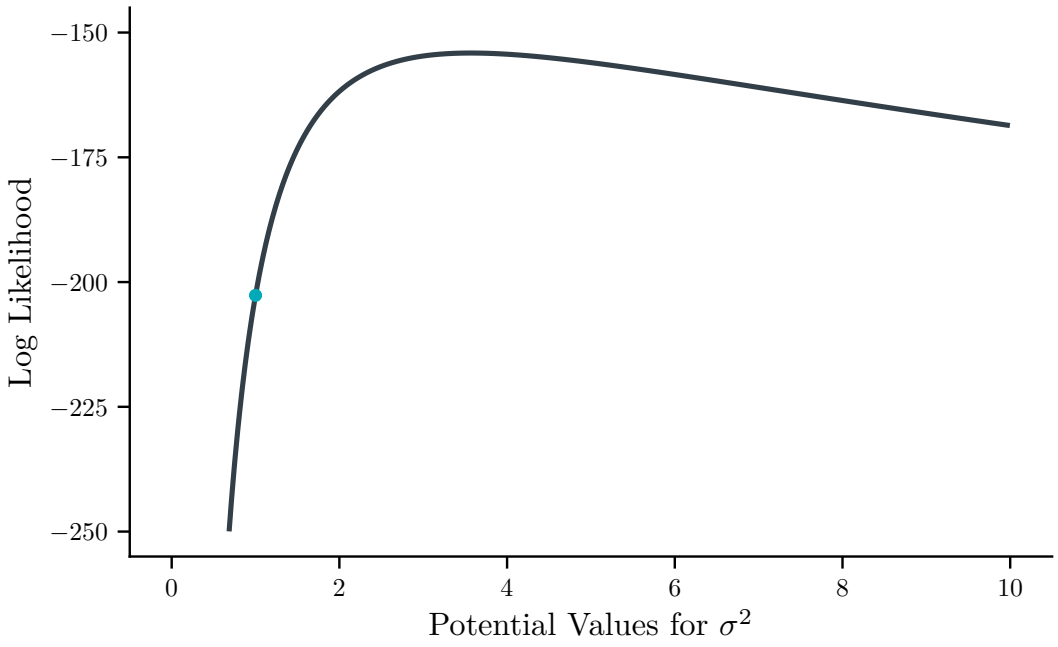
- Newton's method follows a similar procedure to gradient ascent. The key difference is that instead of a learning rate, the method leverages the inverse of the second derivative to adjust the step size.

$$\theta^{(t)} = \theta^{(t-1)} - \left(\frac{\partial^2 \ell}{\partial \theta^2}\right)^{-1} \frac{\partial \ell}{\partial \theta}$$

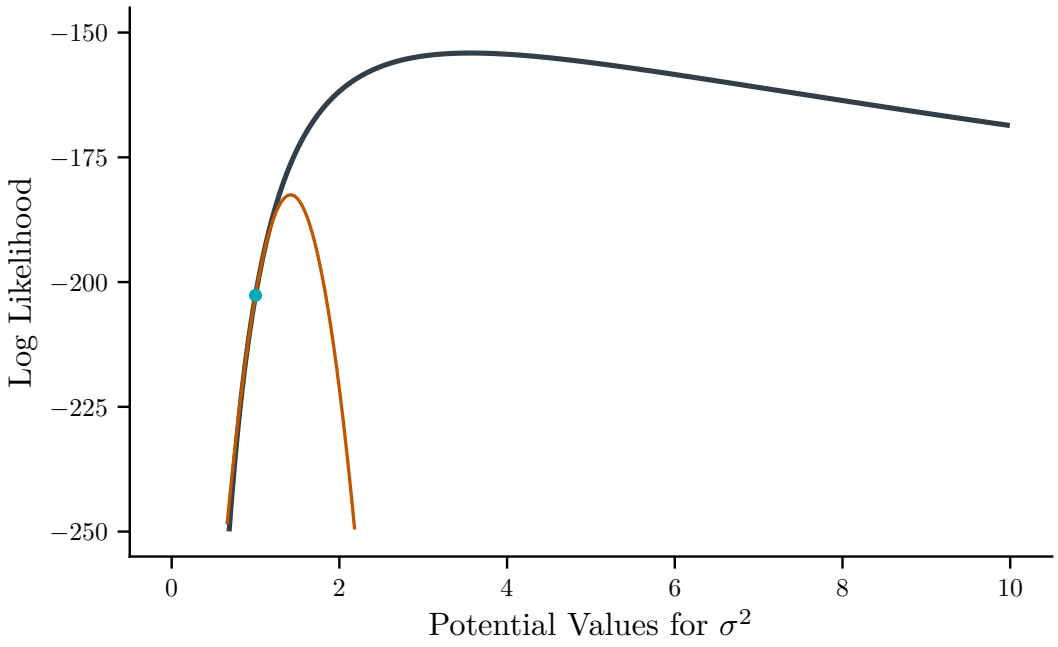
Newton-Raphson – Iteration 0



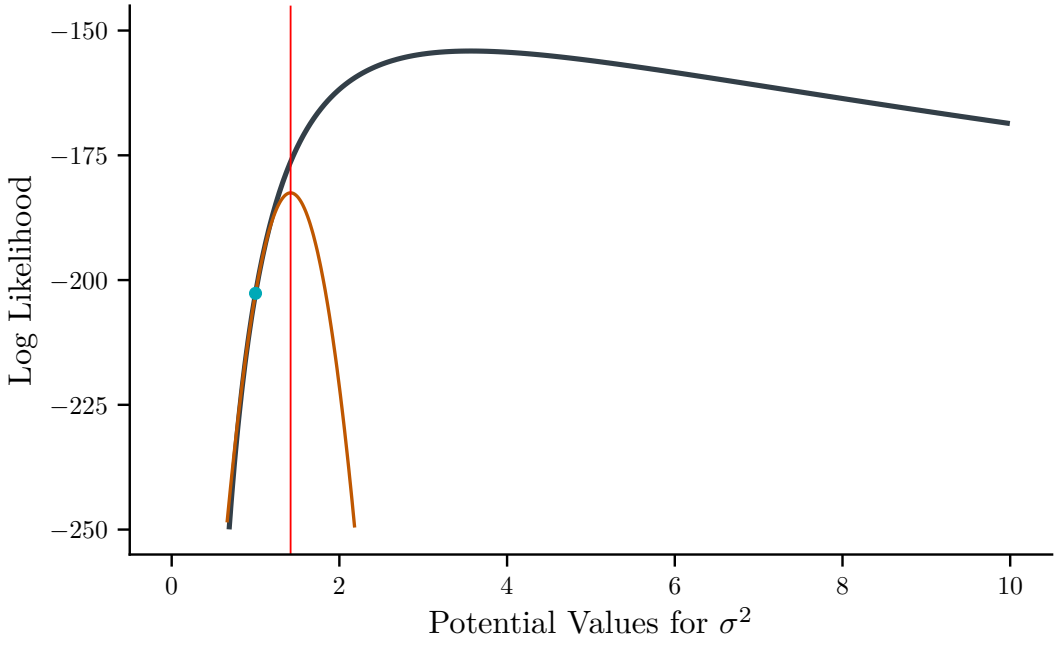
Newton-Raphson – Iteration 0



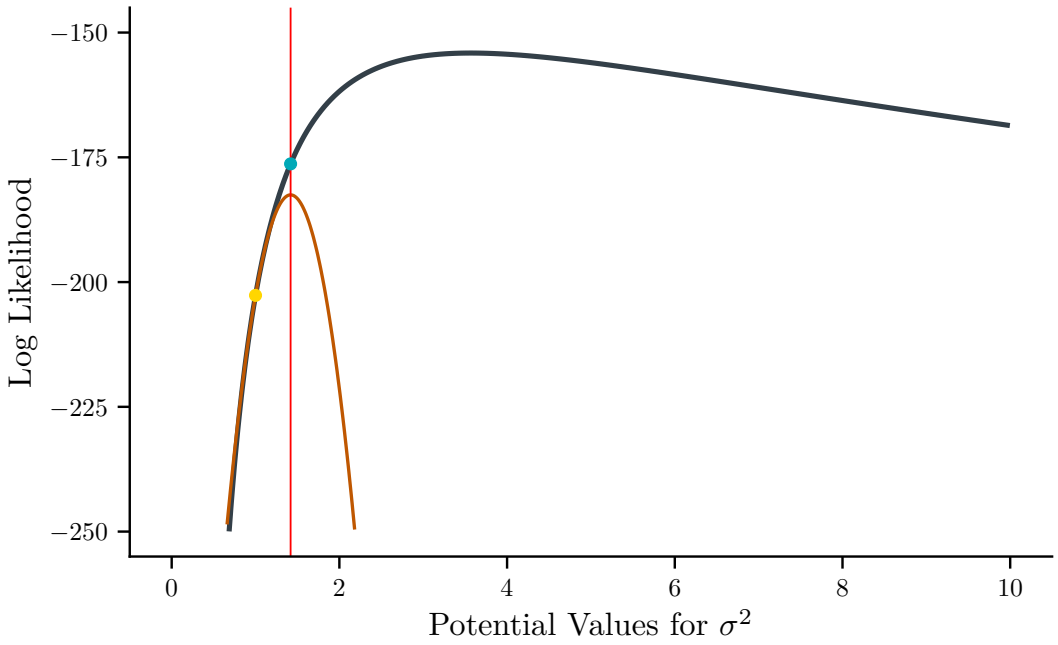
Newton-Raphson – Iteration 0



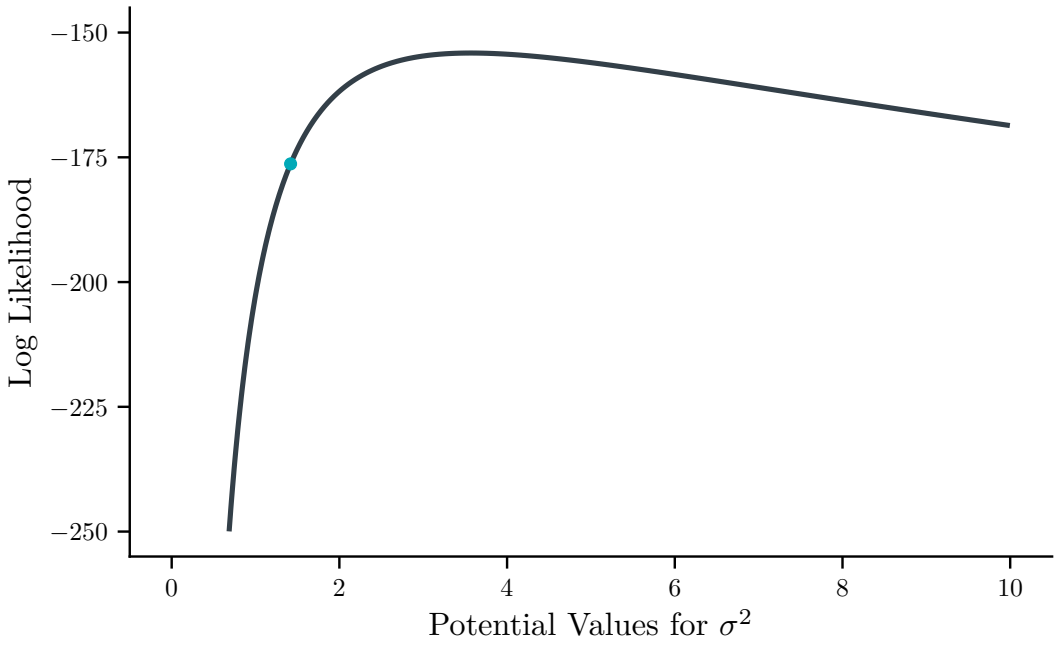
Newton-Raphson – Iteration 0



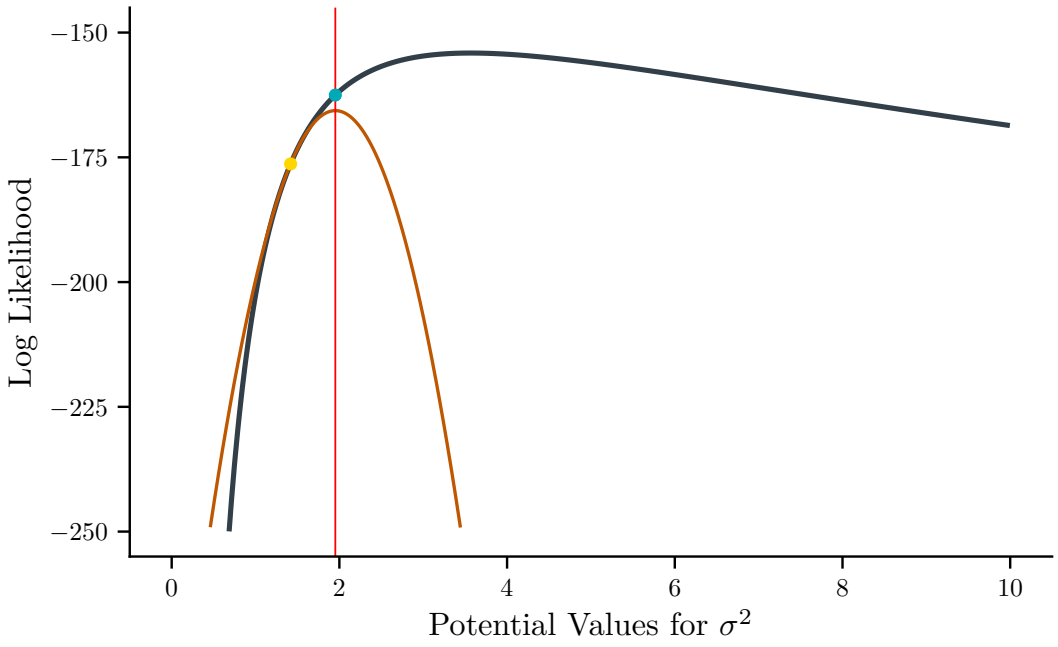
Newton-Raphson – Iteration 0



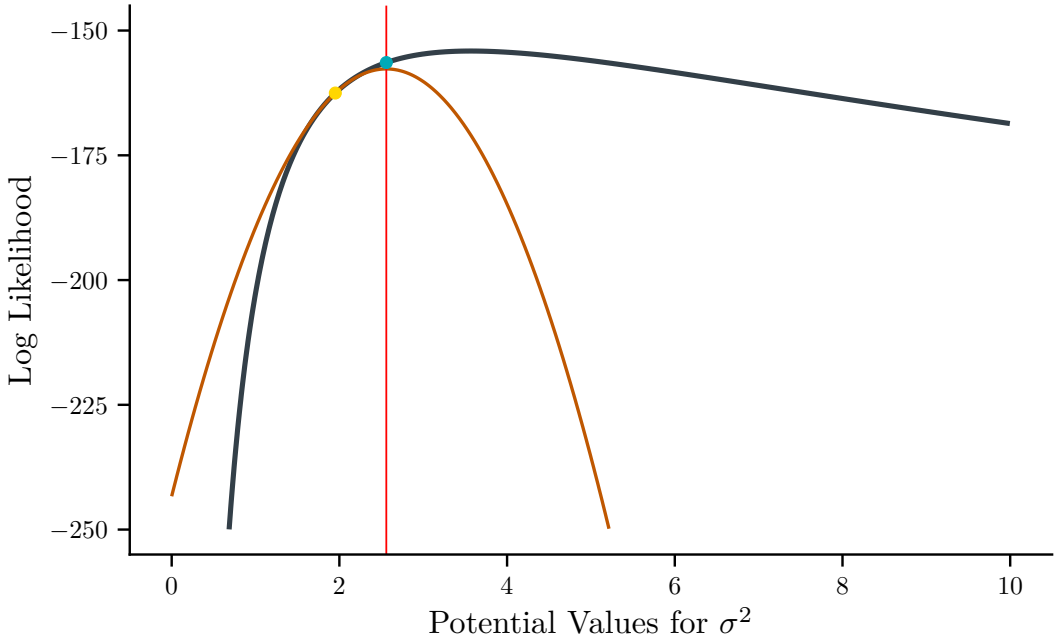
Newton-Raphson – Iteration 1



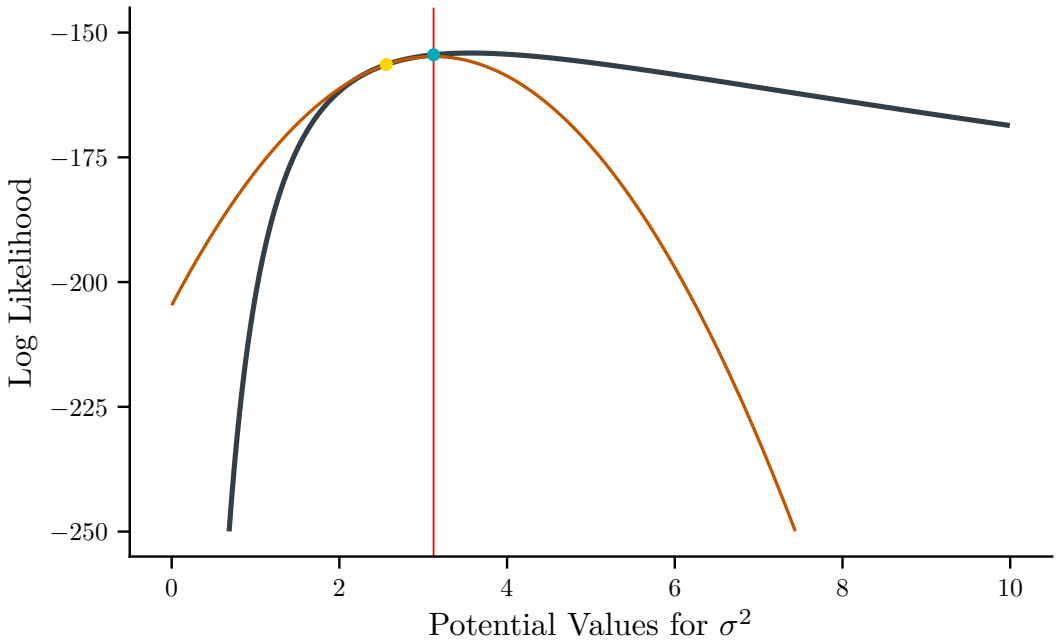
Newton-Raphson – Iteration 1



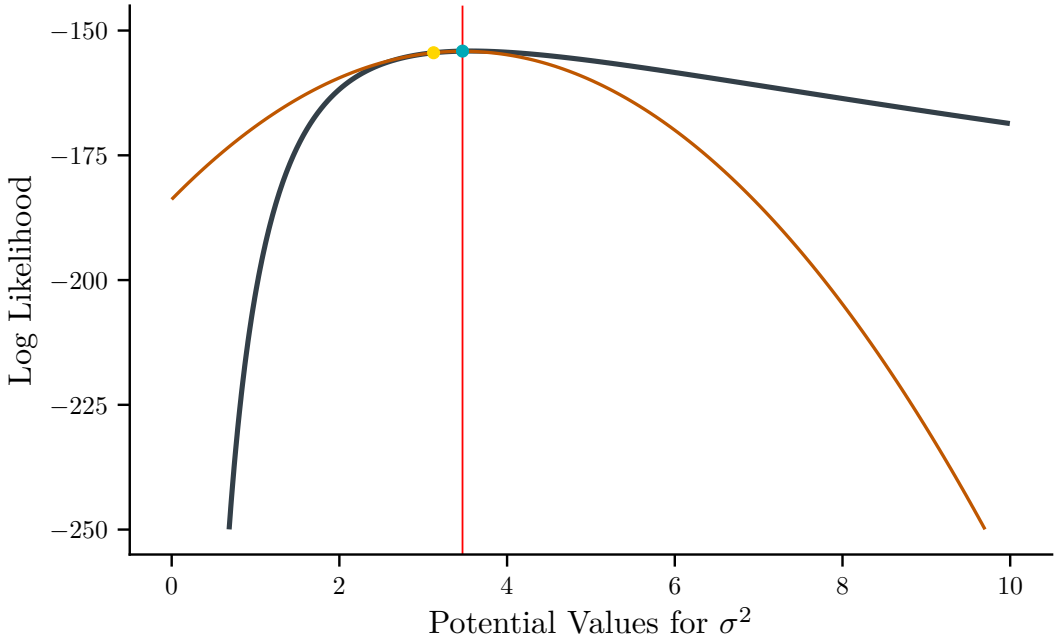
Newton-Raphson – Iteration 2



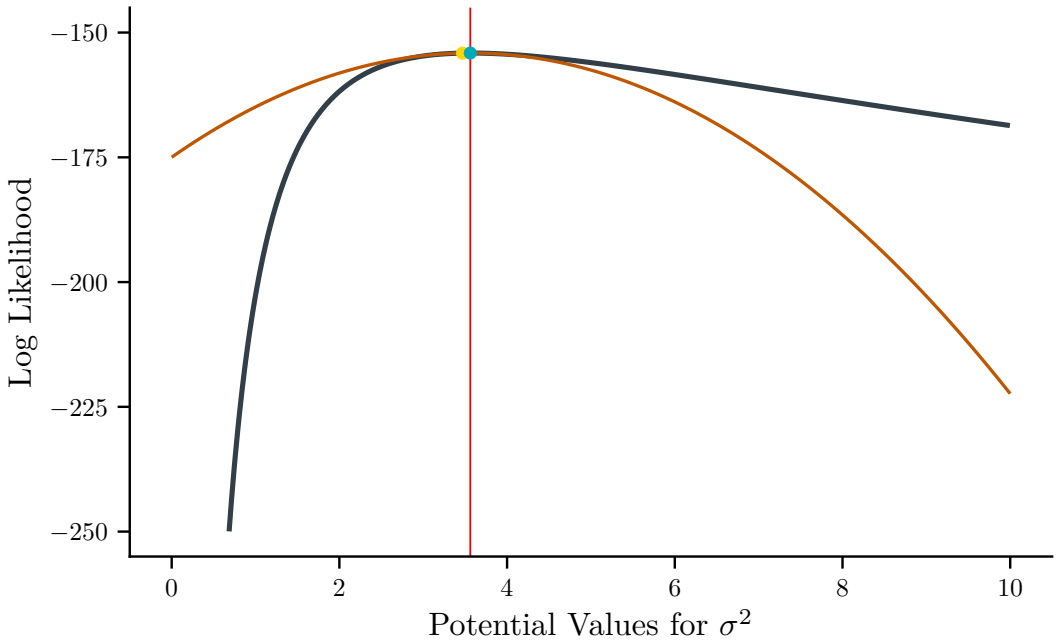
Newton-Raphson – Iteration 3



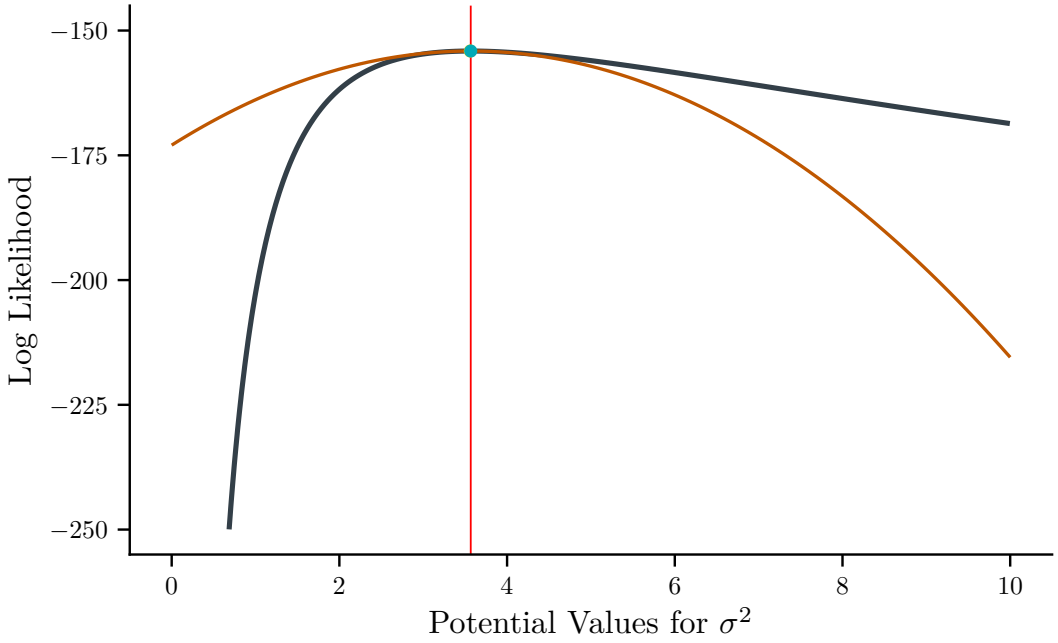
Newton-Raphson – Iteration 4



Newton-Raphson – Iteration 5



Newton-Raphson – Iteration 6

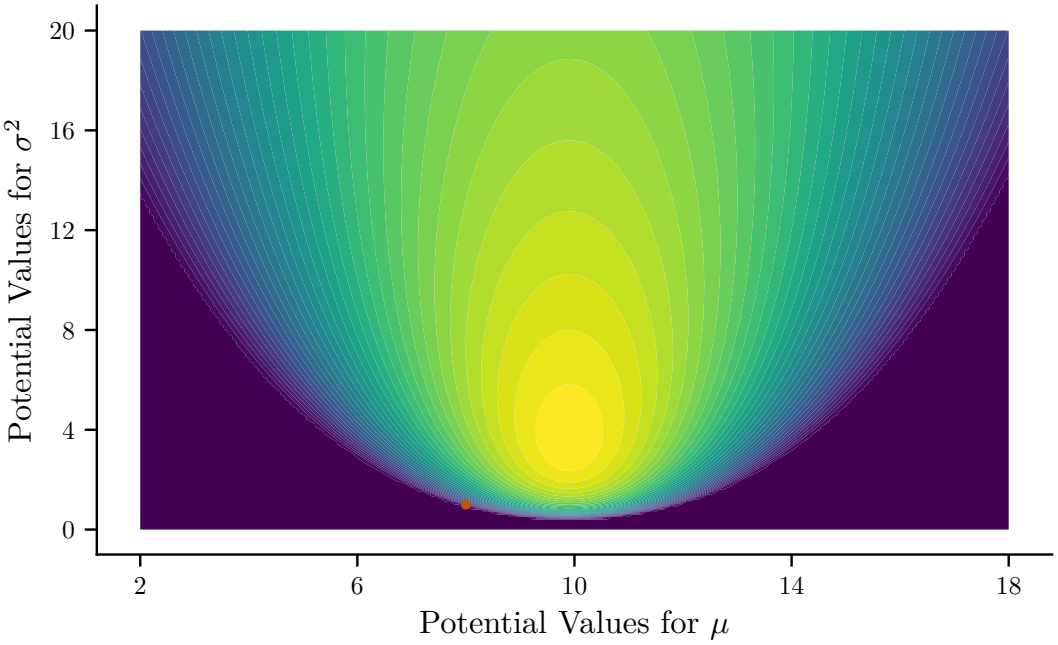


Multivariate Newton-Raphson

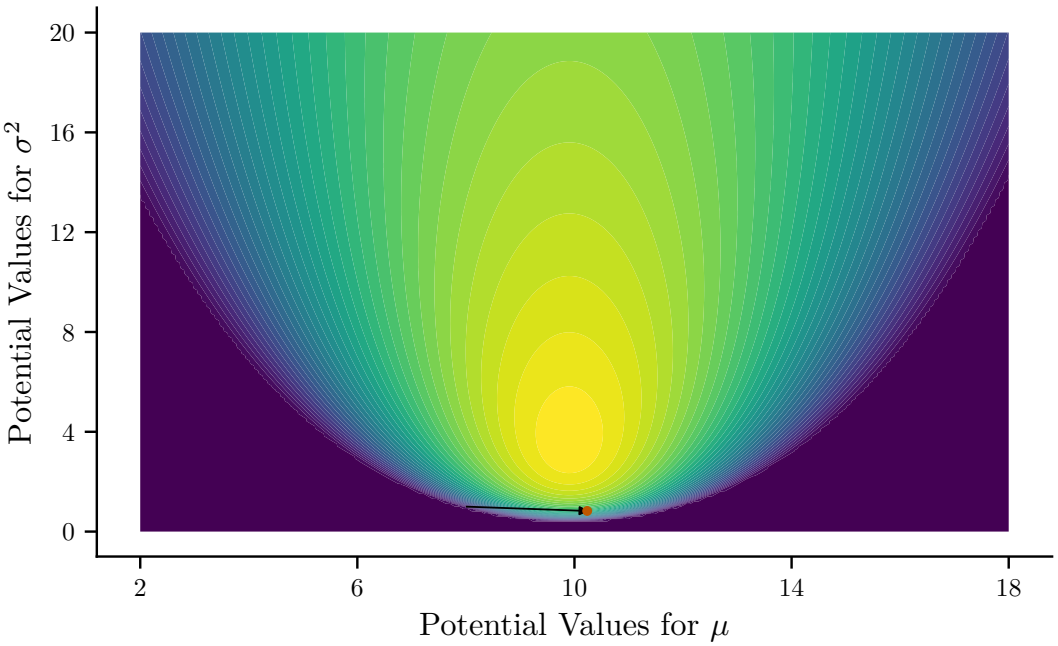
For multivariate updates using Newton-Raphson we need to leverage the Hessian matrix and Score vector. This is one advantage of using this optimizer, our standard errors are easily computed as a byproduct of each update step.

$$\theta^{(t)} = \theta^{(t-1)} - \left[\mathbf{H}\left(\theta\right)\right]^{-1} s\left(\theta\right)$$

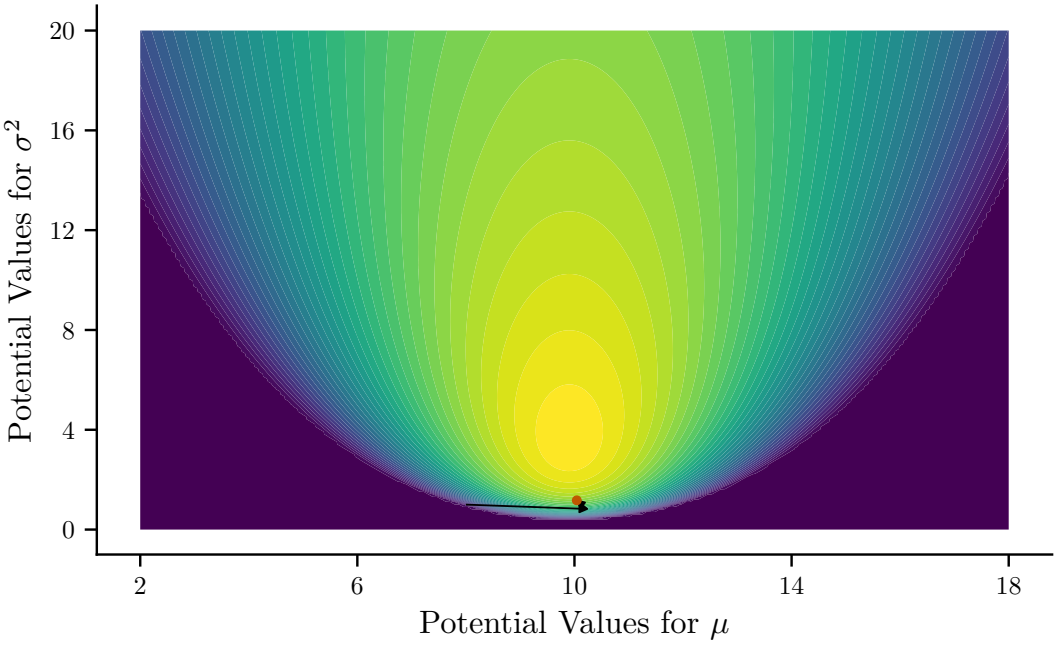
Multivariate Newton-Raphson – Iteration 0



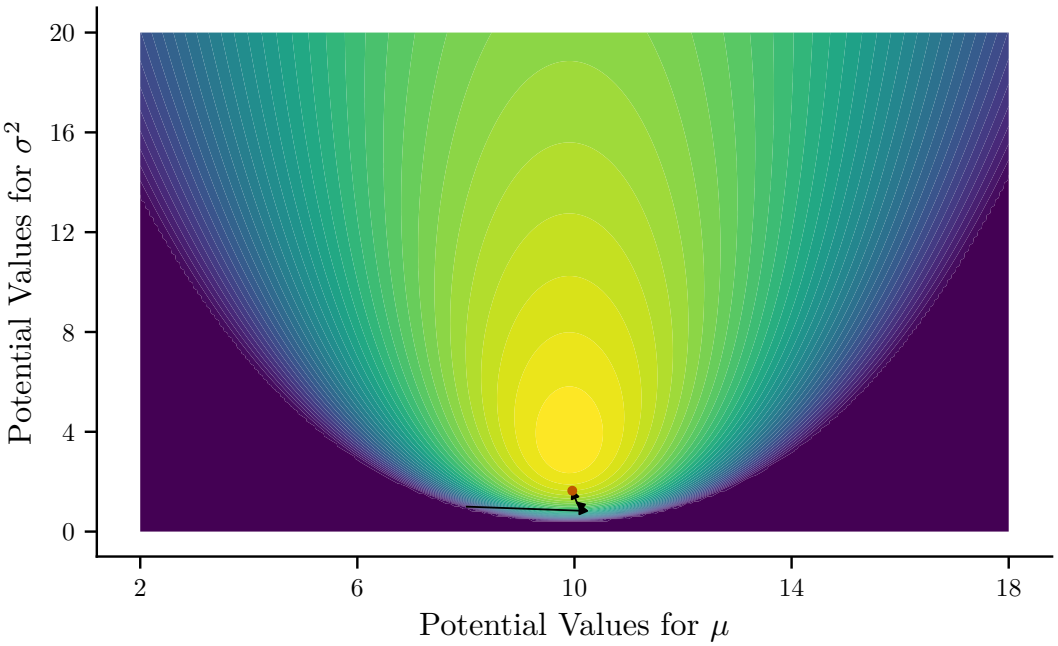
Multivariate Newton-Raphson – Iteration 1



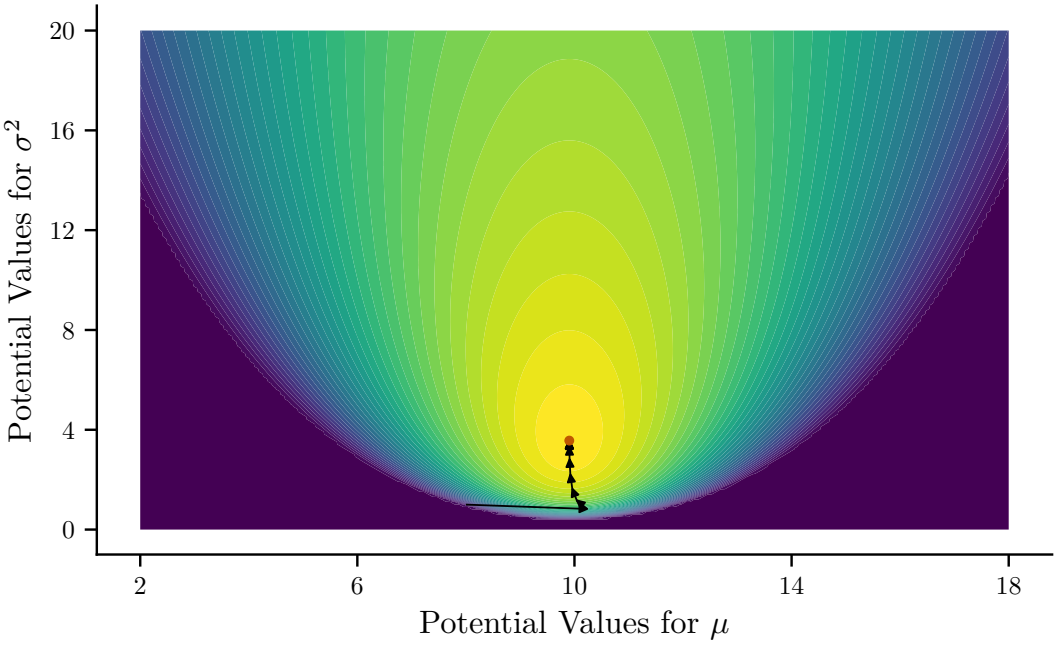
Multivariate Newton-Raphson – Iteration 2



Multivariate Newton-Raphson – Iteration 3



Multivariate Newton-Raphson – Iteration 10

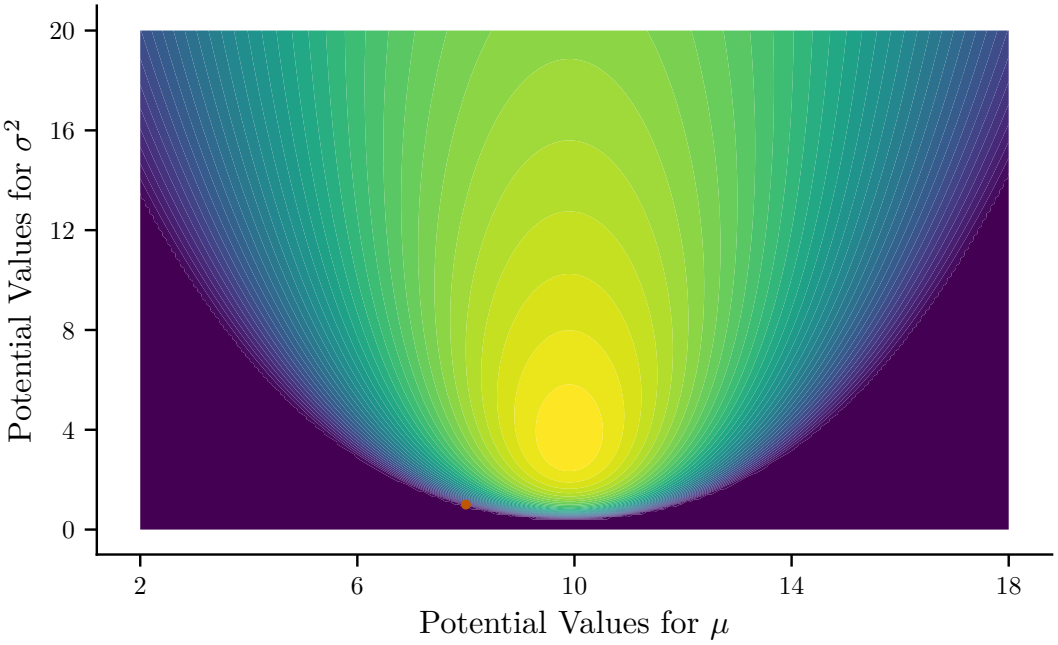


Fisher Scoring

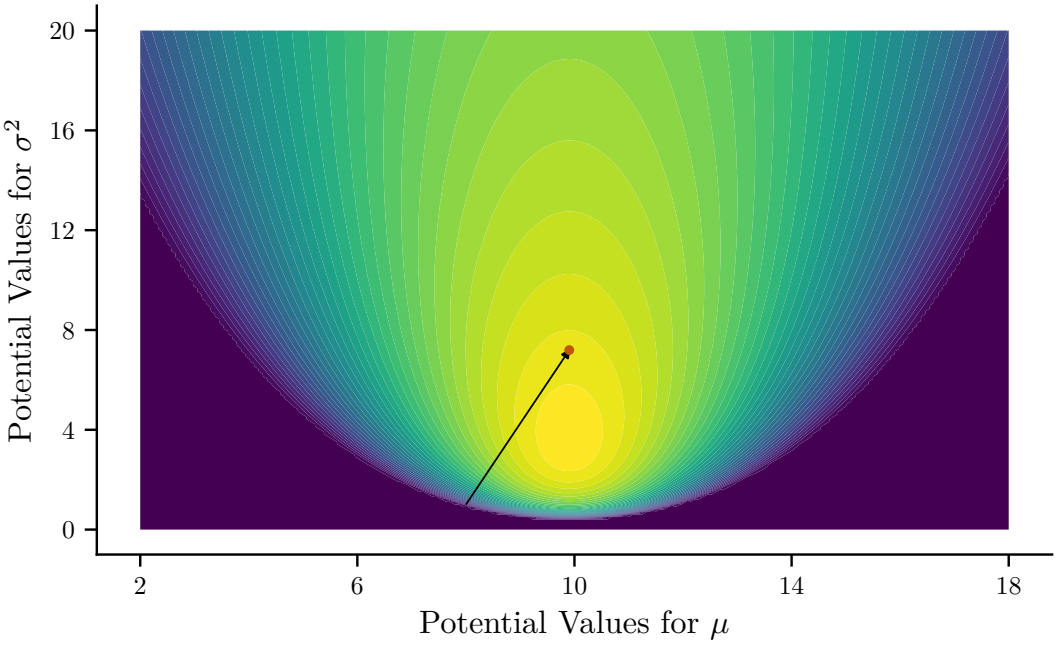
- Fisher Scoring is a special case of a Newton-Raphson method where we use the fisher information instead of the Hessian matrix.
- Fisher scoring tends to converge faster and less impacted by initial starting values.

$$\theta^{(t)} = \theta^{(t-1)} + \left[\mathcal{I}\left(\theta\right)\right]^{-1} s\left(\theta\right)$$

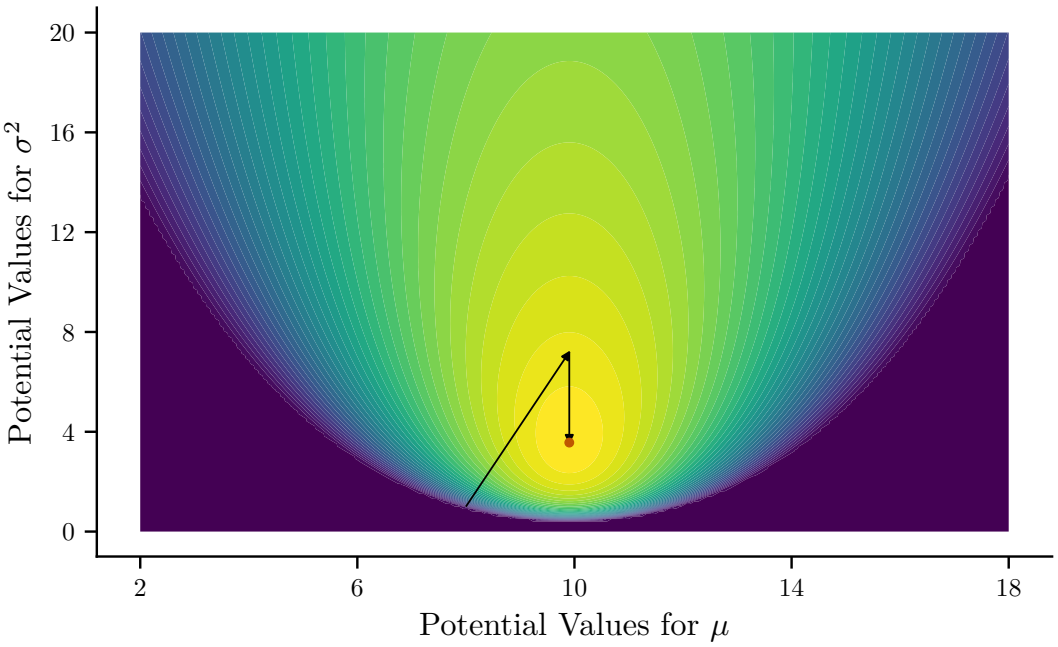
Fisher Scoring – Iteration 0



Fisher Scoring – Iteration 1



Fisher Scoring – Iteration 2



Fisher Scoring – Iteration 3