Numerical optimization

Brian Kissmer

USU Department of Biology

Oct. 24th, 2024

Learning objectives

1. Understand the purpose and nature of numerical optimization
2. Perform numerical optimization in R

Today's outline

1. Introduction to numerical optimization
2. Golden section search
3. Practice in R, plus some methods for reading data

Optimization and inference

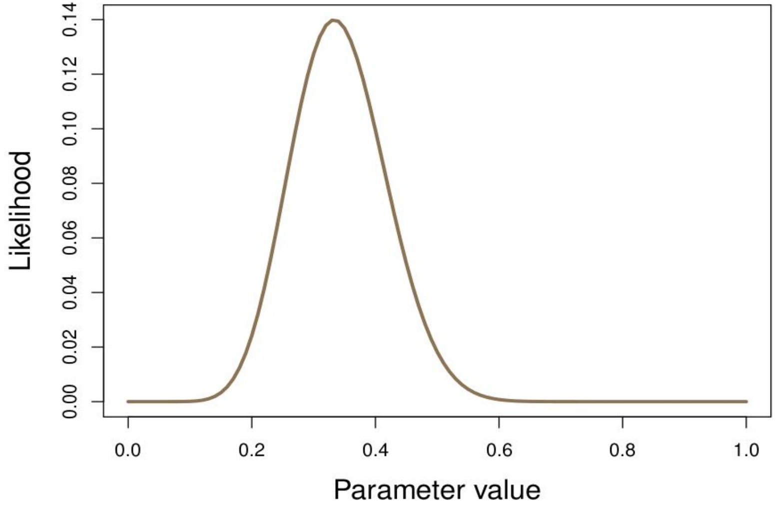
Often, we encounter analysis problems where we want to find the optima (best) parameter value conditional on the data (e.g. treatment effect, growth rate, selection coefficient, population size, etc.). Can you imagine how one might do this? Think about things we've used so far, as well as other possible methods and write out your thoughts on the "Optimization" discussion thread on Canvas.

What do we mean by “best” parameter value?

- One popular method of inference is **Maximum Likelihood Estimation (MLE)**
- Goal is to find the parameter value that maximizes the probability of observed data
- This parameter value is the maximum likelihood estimate

Likelihood, optimization, and inference

Likelihood $\propto$ probability of the data given different parameter values



binomial, $y = 12$, $n = 36$

Likelihood, optimization, and inference

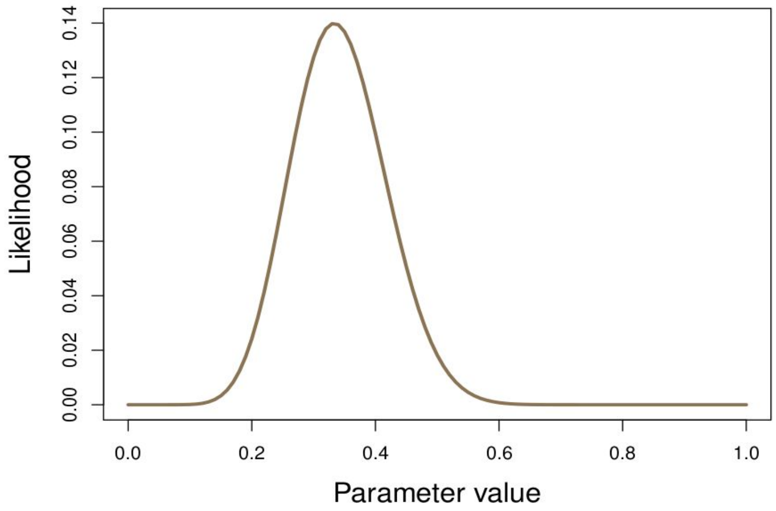
Likelihood $\propto$ probability of the data given different parameter values

$$L(p) \propto p^{y}(1 - p)^{n-y}$$

binomial, $y = 12$, $n = 36$

Likelihood, optimization, and inference

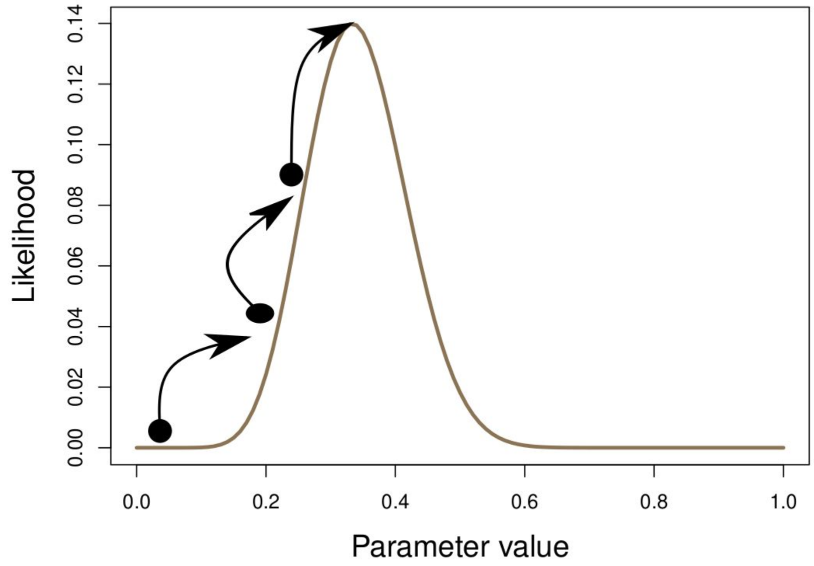
Likelihood $\propto$ probability of the data given different parameter values



binomial, $y = 12$, $n = 36$

Likelihood, optimization, and inference

Computer algorithms can be used to search for optimal solutions = Numerical optimization



binomial, $y = 12$, $n = 36$

Numerical optimization

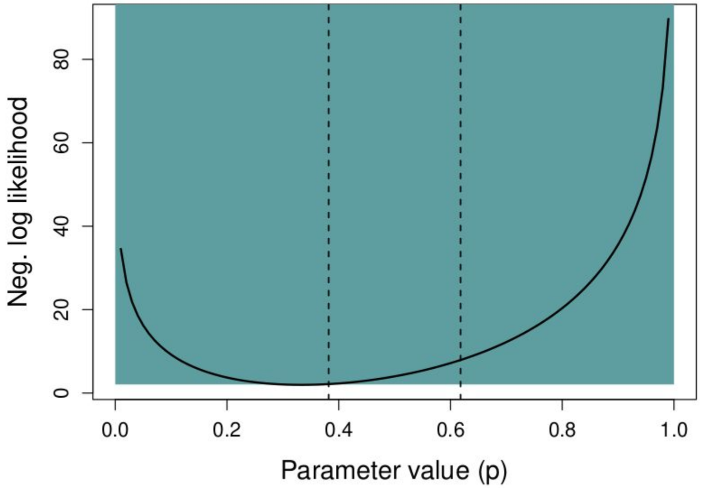
- Numerical optimization can be used in cases where analytical solutions fail
  - i.e., functions that are not differentiable
- Especially useful in models with many parameters
- Many algorithms exist, we will focus on the Golden Search in today's class

Golden search algorithm

The golden section search is a numerical optimization algorithm, similar to the bisection method, that can find the minimum of a function. Let’s try it on a negative (log) likelihood function

Golden search algorithm

1. Define a lower bound *a* (0) and upper bound *b* (1) for the parameter of interest, here *p* from a binomial



binomial, $y = 12$, $n = 36$

Golden search algorithm

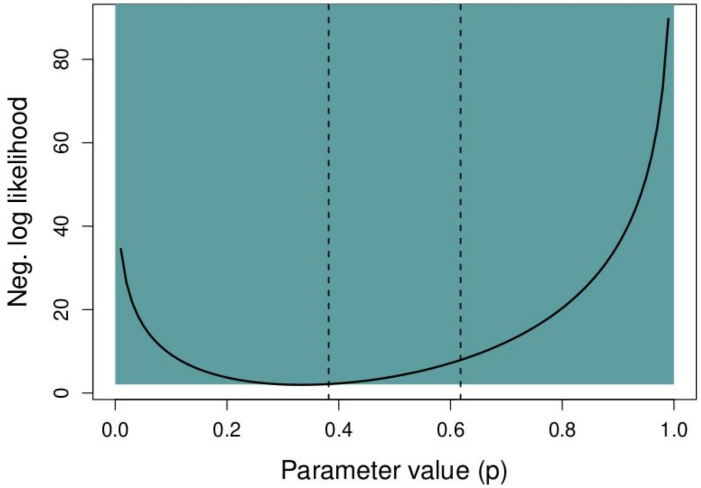
2. Use the golden ratio, $\phi = \frac{1+\sqrt{5}}{2}$, to choose two initial test points within the interval *a* to *b*.

$$x_1 = b - \frac{(b-a)}{\phi}$$

$$x_2 = a + \frac{(b-a)}{\phi}$$

Golden search algorithm

2. Use the golden ratio, $\phi = \frac{1+\sqrt{5}}{2}$, to choose two initial test points within the interval *a* to *b*.



binomial, $y = 12$, $n = 36$

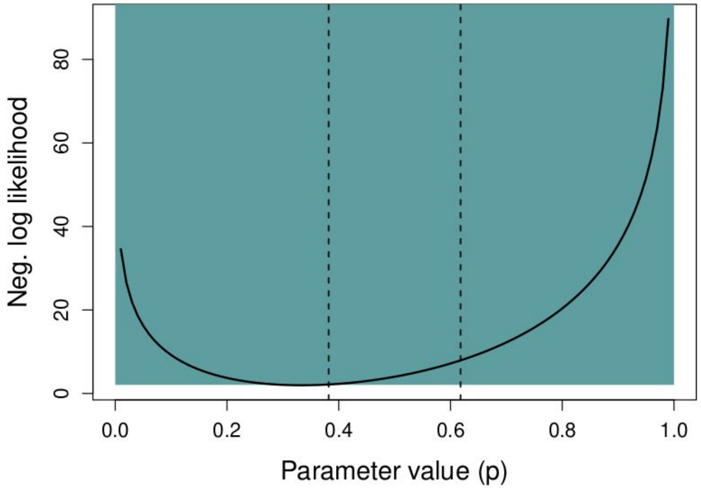
Golden search algorithm

3. Compute $f(x_1)$ and $f(x_2)$.
- ➢ If $f(x_1) > f(x_2)$, minimum (optimum value) is to the right of x_1, set $a = x_1$ and recompute x_1 and x_2.
- ➢ If $f(x_1) < f(x_2)$, minimum (optimum value) is to the left of x_2, set $b = x_2$ and recompute x_1 and x_2.

4. Repeat steps 2 and 3 until some desired tolerance (minimum difference between a and b; the interval a to b then contains your estimate.

Golden search algorithm - example

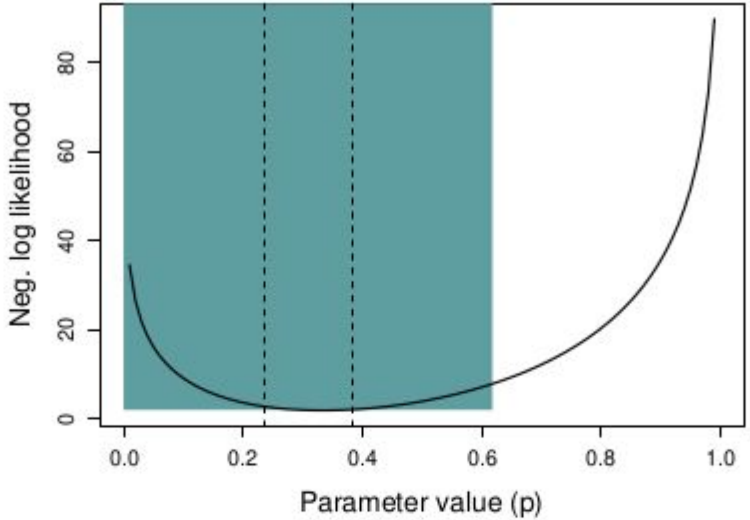
$x_1 = 0.38$, $x_2 = 0.61$, $-lnL(x_1) < -lnL(x_2)$, thus $b = x_2$.



binomial, $y = 12$, $n = 36$

Golden search algorithm - example

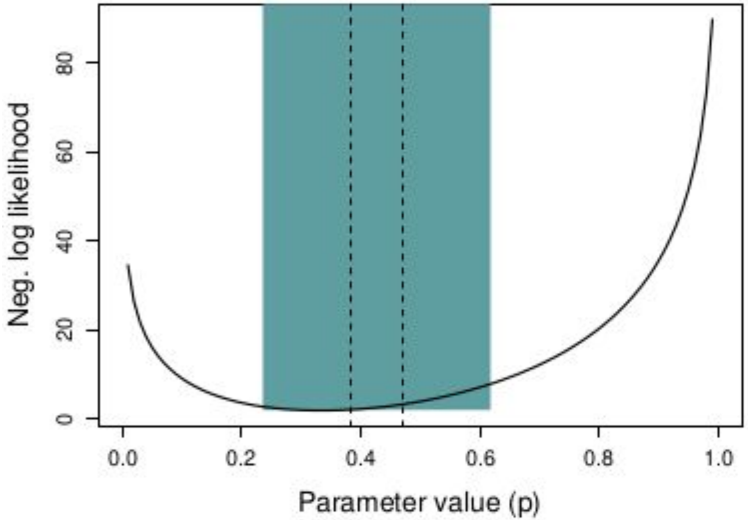
$x_1 = 0.236$, $x_2 = 0.382$, $-lnL(x_1) > -lnL(x_2)$, thus $a = x_1$.



binomial, $y = 12$, $n = 36$

Golden search algorithm - example

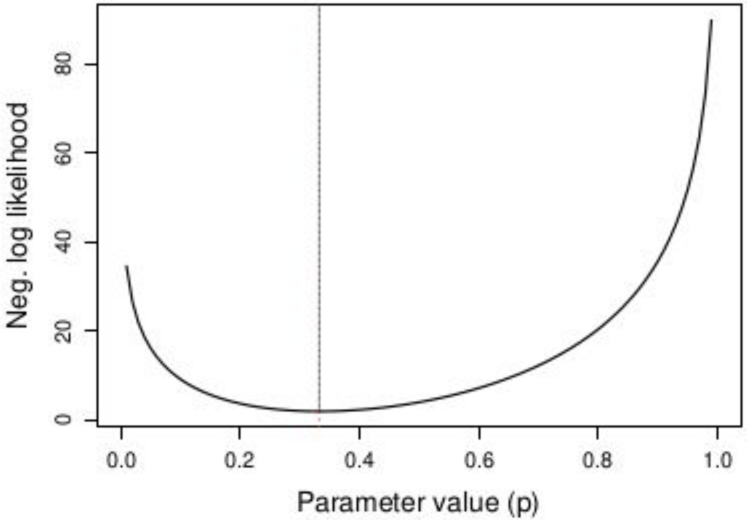
$x_1 = 0.382$, $x_2 = 0.472$, $-lnL(x_1) < -lnL(x_2)$, thus $b = x_2$.



binomial, $y = 12$, $n = 36$

Golden search algorithm - example

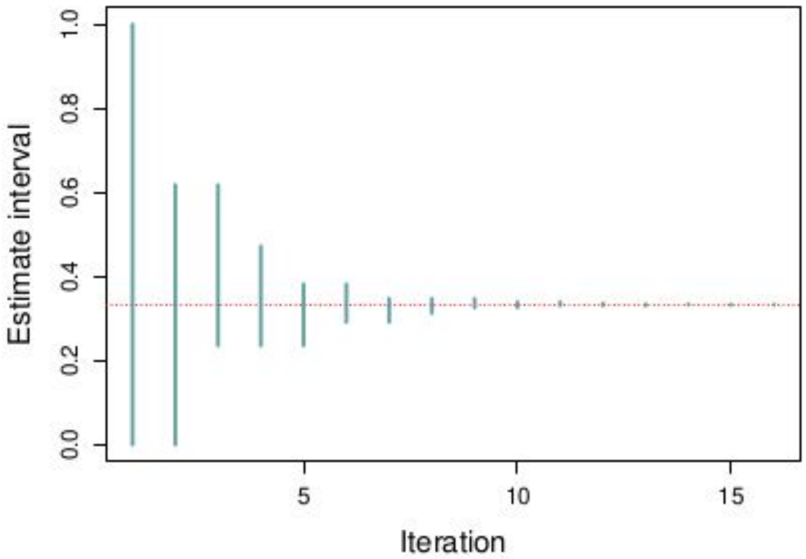
After enough iterations, $a \approx b \approx 0.33$



binomial, $y = 12$, $n = 36$

Golden search algorithm - example

The estimate intervals converge to the right answer with each step



binomial, $y = 12$, $n = 36$

Golden search algorithm - practice in R

See the golden search R code and data set