

NRES 746, Week 8, Lecture 1: Newton's Method by Hand (take-home)

Name: _____

Bring this to class on Wednesday. We'll start by going over it, and Wednesday's lecture picks up where it leaves off. Work on paper; a calculator is all you need.

Putting it in context

Since Week 6 we've found MLEs with `optim()` without asking how it finds them. On Wednesday we open the black box. This WOD works through the core of one method, Newton's method, on a problem small enough to do by hand.

The data are the Week 7 pika haypile counts from six talus plots: 2, 5, 3, 0, 4, 4 (sum 18), modeled as Poisson with mean λ . Dropping terms that don't involve λ ,

$$\text{NLL}(\lambda) = 6\lambda - 18 \log \lambda$$

Part 1: Slope and curvature

1a. Find $\text{NLL}'(\lambda)$ and $\text{NLL}''(\lambda)$. Then set $\text{NLL}'(\lambda) = 0$ and solve for the MLE, $\hat{\lambda}$.

Part 2: Slopes without calculus

`optim()` can't do calculus on its own.¹ By default it estimates slopes from NLL values at nearby points, using a **finite difference**: the slope of a straight line (a chord) between two points a fixed, small distance Δ apart. Unlike the definition of a derivative, Δ never shrinks to zero, so the result is an approximation. The **forward difference** uses the current point and one step ahead:

$$f'(x) \approx \frac{f(x + \Delta) - f(x)}{\Delta}$$

Here are NLL values at three points spaced $\Delta = 0.5$ apart:

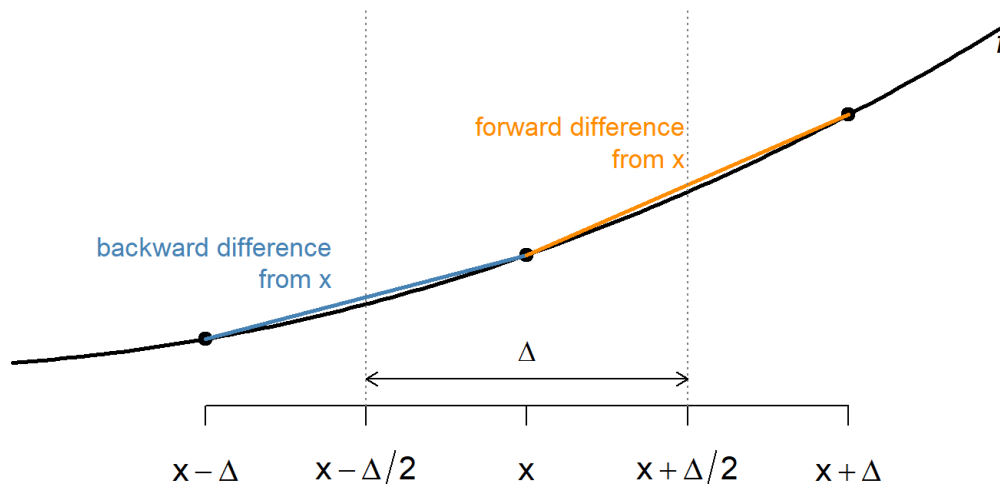
λ	1.5	2.0	2.5
NLL(λ)	1.7016	-0.4766	-1.4932

2a. Use the forward difference to estimate $\text{NLL}'(2)$ and compare it with the exact value from 1a. Then find the exact $\text{NLL}'(2.25)$. What do you notice?

Forward estimate: _____ Exact $\text{NLL}'(2)$: _____ Exact $\text{NLL}'(2.25)$: _____

¹If you know the formula for the first derivatives, you can write it as an R function and pass it to `optim()` through the `gr` argument. Derivative-based methods such as `method = "BFGS"` then skip the finite differences, which makes them faster and more reliable (Bolker p. 301). `optim()` has no way to take second derivatives, but `nlm()` can use both.

2b. Curvature is the slope of the slope: how fast the slope changes per unit change in the parameter. The figure below starts from a center point x and draws two chords: a **backward difference** (from x back to $x - \Delta$) and a forward difference (from x to $x + \Delta$). Combine these two slopes into a finite-difference approximation to $f''(x)$ that uses only $f(x - \Delta)$, $f(x)$, and $f(x + \Delta)$. (What you noticed in 2a may help you decide what to divide by.) Then use it to estimate $\text{NLL}''(2)$ and compare with the exact value.



Estimate: _____ Exact: _____

Part 3: Newton's method

The MLE is where the slope is zero: $\text{NLL}'(\lambda) = 0$. Newton's method finds that point by guessing and correcting. At the current guess λ , it pretends the slope NLL' changes *linearly* across parameter space, starting at $\text{NLL}'(\lambda)$ and changing at a constant rate $\text{NLL}''(\lambda)$.

3a. Under that linear approximation, find the value λ^* where the slope reaches zero. Write λ^* in terms of λ , $\text{NLL}'(\lambda)$, and $\text{NLL}''(\lambda)$. (A sketch of NLL' against λ may help.) Your answer is Newton's update rule: λ^* becomes the next guess.

$\lambda^* =$ _____

3b. Start at $\lambda = 2$ and take three Newton steps, using your rule from 3a and your exact derivatives from 1a. Keep at least three decimal places.

Step	Guess	Slope, NLL'	Curvature, NLL''	Next guess
1	2			
2				
3				

3c. Now start at $\lambda = 7$ and take one step. What went wrong? Sketch $NLL'(\lambda)$ and the straight line Newton's method used at $\lambda = 7$ to explain why, and suggest one way to fix it.

3d. (*If you have time.*) Real optimizers usually don't know the exact derivatives and use finite-difference estimates instead. By hand, take one Newton step from $\lambda = 2$ using your estimates from 2a and 2b in place of the exact derivatives. How close is it to your first step in 3b? Based on 2a, how could you make it closer?