Unit 6: Optimization

Chapter 14 in "Statistical Computing with R"

Anna Ly

Department of Mathematical and Computational Sciences
University of Toronto Mississauga

January 19, 2026

Overview

Introduction

Introduction

- You've seen optimization methods before: the MLE.
- We'll discuss the following (non-exhaustive list):
  - R's standard optimize() function.
  - R's optim() function, and commonly used methods (Nelder-Mead, BFGS, L-BFGS-B, GG, SANN, Brent).
  - Newton Raphson.
  - EM Algorithm.

Optimize() and Optim()

Optimize()

- According to the documentation: the method used is a combination of golden section search and successive parabolic interpolation to find when the **maximum** occurs.
- Unfortunately, the source code for optimize() is actually written in C! See here! (Click link). Same goes for optim().
- Out of interest in time, we'll just discuss how to use optimize() and optim().
- We can use this to quickly find the maximum point for one-dimensional functions!

Optimize()

Example

Use optimize() to maximize this function with respect to x:

$$f(x) = \frac{\log(1 + \log(x))}{\log(1 + x)}$$

Solution.

Optim()

optim() is more popular than optimize() because you can use this for multi-dimensional problems and can accomodate a wide variety of methods:

1. Nelder-Mead
2. BFGS (Broyden-Fletcher-Goldfarb-Shanno)
3. CG (Conjugate gradient)
4. L-BFGS-B (I think L = Limited-memory, and B = Bound-constrained?)
5. SANN (Simulated-Annealing)
6. Brent; one-dimension only, same as optimize().

These methods are graduate-level topics, but you can use these functions anyways. Note that by default, optim() gives the **MINIMUM** value, but to obtain the maximum you can just put a negative sign in front of the function. Let's try to test them out.

Optim()

Example

1. Derive the likelihood function $\hat{\boldsymbol{\theta}} = (\hat{\alpha}, \hat{\lambda})$ for $X_1, \ldots, X_n \sim Gamma(\alpha, \lambda)$ where α is the shape parameter and λ is the rate parameter.
2. Then, for $n = 10$ randomly sample from a $Gamma(\alpha = 5, \lambda = 2)$ using rgamma().
3. Use optim() to maximize this function with respect to $\boldsymbol{\theta} = (\alpha, \beta)$ and compare the results between the 5 different methods.

Solution.

Optim()

Optim()

Example

Use optim() to maximize this function with respect to x:

$$f(x, n, r, t) = \left(1 + \frac{n - r}{2}\right) x - r + \left(\frac{n - r}{2}\right) t$$

Compare the results between the 5 different methods.

Solution.

Newton-Raphson Method

Newton-Raphson Method

- Named after Isaac Newton and Joseph Raphson, is a root-finding algorithm which produces successively better approximations to the roots (or zeroes) of a real-valued function.
- This method is helpful when it is impossible to hand compute/derive the explicit form for the likelihood function, and consequently, the MLE.
- We will use Newton-Raphson to numerically maximize the log-likelihood.
- We will first introduce the intuitive idea in the one variable case.

Understanding Newton-Raphson - One Variable Case

- Suppose we test one value for our function, call it x_0.
- Draw a tangent line to $f(x)$ at x_0. This tangent line will intersect the x-axis at some fixed point $(x_1, 0)$.
- Draw another tangent line to $f(x)$ at x_1; the tangent line will intersect the x-axis at some point $(x_2, 0)$...
- Repeat the above steps until you find a value of x_i such that $f(x_i) = 0$.

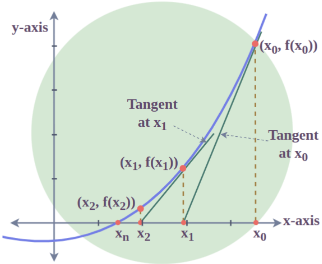

Image source: https://www.geeksforgeeks.org/engineering-mathematics/newton-raphson-method/

Understanding Newton-Raphson - One Variable Case

- Recall from calculus: the Taylor polynomial of order n generated by $f(x)$ at $x = a$ is the polynomial

$$P_n(x) = f(a) + f'(a)(x-a) + \frac{f^{(2)}(a)}{2}(x-a)^2 \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n$$

- The linear approximation around x_0 is:

$$P_1(x) = f(x_0) + f'(x_0)(x - x_0)$$

We want $P_1(x_0) = 0$:

$$0 = f(x_0) + f'(x_0)(x - x_0) \;\Rightarrow\; x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Understanding Newton-Raphson - One Variable Case

Newton-Raphson - One Variable

In the general form, the Newton-Raphson method formula is written as follows:

$$x_n = x_{n-1} - \frac{f(x_n)}{f'(x_n)}$$

We now want to extend this to multiple variables.

Understanding Newton-Raphson - Multiple Variable Case

- Remember the **Jacobian**: "the Jacobian matrix of a vector-valued function of several variables is the matrix of all its first-order partial derivatives."
- Let $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ and $\mathbf{x_0} = (x_{01}, x_{02}, \ldots, x_{0k})$
- Thus instead we can do:

$$f(\mathbf{x}) \approx f(\mathbf{x_0}) + J(\mathbf{x_0})(\mathbf{x} - \mathbf{x_0})$$

- We can solve for $\mathbf{x}$ the same way:

$$\mathbf{x} = \mathbf{x_0} - J(\mathbf{x_0})^{-1} f(\mathbf{x_0})$$

- Assuming the Jacobian is invertible.

Modifying Newton-Raphson for our Case

- Consider $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k)$ and some values $\boldsymbol{\theta}^* = (\theta_1^*, \theta_2^*, \ldots, \theta_k^*)$.
- Note that when we are solving for the MLE, we are not finding the roots. What we need to do is maximize the likelihood function $L(\boldsymbol{\theta})$.
- We know the likelihood is maximized (or minimized) when the derivative of the likelihood $\mathcal{L}(\boldsymbol{\theta}^*) = \frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} \overset{set}{=} 0$.
- And now we need to consider the Hessian matrix (call $H(\cdot)$), which includes the second order partial derivatives.
- Thus, borrowing the equation from the previous slide but adjusting it to our new situation, we now have the following:

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^* - H(\boldsymbol{\theta}^*)\mathcal{L}(\boldsymbol{\theta}^*)$$

Newton-Raphson Method

Newton-Raphson for Solving the MLE

1. Choose a reasonable starting value $\boldsymbol{\theta}^{(0)}$.
2. Update your estimate:

$$\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}^{(n-1)} - H(\boldsymbol{\theta}^{(n-1)})\mathcal{L}(\boldsymbol{\theta}^{(n-1)})$$

3. Terminate the algorithm when $||\boldsymbol{\theta}^{(n)} - \boldsymbol{\theta}^{(n-1)}|| < \epsilon$ where ϵ is a small threshold. In this scenario, we're letting $|| \cdot ||$ represent the Euclidean norm:

$$||\mathbf{x}|| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

Newton-Raphson Method

Example

Consider the same log likelihood function we solved earlier for gamma. Again, randomly sample values from *Gamma*$(\alpha = 5, \lambda = 2)$ and now use the Newton-Raphson Method to solve for the MLE. *Hint: you're allowed to use numDeriv::grad and numDeriv::hessian*.

Solution.

Expectation-Maximization Algoritm

EM Algorithm

- You've probably encountered a scenario where there's bad or missing data...
- Missing data is a problem in every field, whether it be machine learning, survival analysis, etc.
- That won't stop us from trying to come up with good estimators!!
- Rather than deriving an expression for solving the MLE, we specify an algorithm that is guaranteed to converge to the MLE.
- It is based on the idea of replacing one difficult likelihood maximization with a sequence of easier maximizations whose limit is the answer to the original problem.

EM Algorithm

1. Let $\mathbf{Y}$ be the complete data with log likelihood $l_c(\boldsymbol{\theta}; \mathbf{Y})$.
2. Let $\mathbf{X}$ be the observed (incomplete) data with log likelihood $l(\boldsymbol{\theta}; \mathbf{X})$.
3. Choose an initial estimate $\hat{\boldsymbol{\theta}}^{(0)}$.
4. **E-step:** Calculate:
$$\mathbf{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(k)}) = \mathbb{E}[l_c(\boldsymbol{\theta}; \mathbf{Y}) | \hat{\boldsymbol{\theta}}^{(k)}, \mathbf{X}]$$
5. **M-step:** Choose $\hat{\boldsymbol{\theta}}^{(k+1)}$ to maximize $\mathbf{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(k)})$.
6. Iterate the **E** and **M** step until:
$$\mathbf{L}(\hat{\boldsymbol{\theta}}^{(k+1)}) - \mathbf{L}(\hat{\boldsymbol{\theta}}^{(k)}) < \epsilon$$

EM Algorithm

- Consider iid $X_1, \ldots, X_n$ where $X_i \sim Poisson(\tau)$ for $i = 1, 2, \ldots, n$. Suppose we had all of the realizations $x_2, x_3, \ldots, x_n$ except for X_1.
- **Idea:** discard X_1 as well and then just compute and maximize:

$$L(\tau | X_2 = x_2, X_3 = x_3, \ldots, X_n = x_n)$$

- **Any issues with the idea?**

EM Algorithm

Example

It is still useful to compute the incomplete likelihood:

$$L(\tau | X_2 = x_2, X_3 = x_3, \ldots, X_n = x_n)$$

So derive this as well as the MLE to get an initial estimate.

Solution.

EM Algorithm

EM Algorithm

Example

For the same example as before, compute the expectation step:

$$\mathbb{E}\left[l(\tau|\mathbf{x}) \mid \hat{\tau}^{(k)}, \mathbf{x}_{-1} \right] \tag{1}$$

Solution.

EM Algorithm

EM Algorithm

EM Algorithm

Example

Explicitly code up the EM algorithm for the case we just considered.

Solution.

EM Algorithm

Some remarks regarding the EM algorithm:

- Typically, the maximization step, i.e., maximizing $\mathbb{E}[l_c(\boldsymbol{\theta}; \mathbf{Y})|\hat{\boldsymbol{\theta}}^{(k)}, \mathbf{X}]$ is VERY difficult and cannot done manually. Thus, we tend to rely on numerical methods, such as Newton-Raphson, to do this!
- On a personal note, I tried to come up with several examples to do EM algorithm by hand; the only one I could feasibly come up with was this Poisson case. Hence, for examinations, I will either:
  - Ask you to derive parts UP until the expectation step. No maximization explicitly required; I may ask you "what could you do to maximize this if you had the tools?"
  - Give you a Poisson case to derive (very similar to the lecture slides.)