STA6171: Statistical Computing for DS 1

Solving Nonlinear Equations

Ick Hoon Jin

Yonsei University, Department of Statistics and Data Science

2020.09.09

Motivaing Example

- A simple univariate numerical optimization problem is to maximize

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

with respect to x.

- The derivative of $g(x)$ is

$$g'(x) = \frac{1 + \frac{1}{x} - \log x}{(1 + x)^2} \tag{2}$$

and no analytic solution can be found in maximization.

Motivaing Example

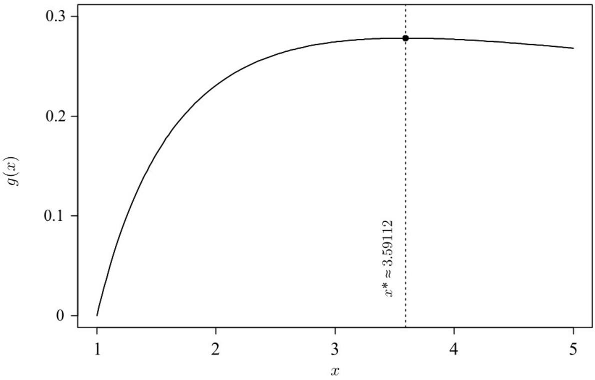

FIGURE 2.1 The maximum of $g(x) = (\log x)/(1+x)$ occurs at $x^* \approx 3.59112$, indicated by the vertical line.

Convergence Criterion

- We cannot allow the algorithm to run indefinitely, so we require a *stopping rule* based on convergence criteria, to trigger an end to the successive approximation.
- At each iteration, the stopping rule should be checked and the new $x^{(t+1)}$ is taken as the solution when the convergence criteria are met.
- Two reasons to stop if the algorithm appears
  - to have achieved satisfactory convergence.
  - unlikely to converge soon.

Convergence Criterion

- Monitor convergence by tracking the proximity of $g'\left(x^{t+1}\right)$ to zero is not reliable because large changes from $x^{(t)}$ to $x^{(t+1)}$ can occur even when $g'\left(x^{(t+1)}\right)$ is very small.
- On the other hand, a small change from $x^{(t)}$ to $x^{(t+1)}$ is most frequently associated with $g'\left(x^{(t+1)}\right)$ near zero.
- Assess convergence by monitoring $\left|x^{(t+1)} - x^{(t)}\right|$ and use $g'\left(x^{(t+1)}\right)$ as a backup check.

Types of Convergence Criterion

- Absolute convergence criterion mandates stopping when

$$\left| x^{(t+1)} - x^{(t)} \right| < \epsilon,$$

where ϵ is a constant chosen to indicate tolerable imprecision.

- Relative convergence criterion mandates stopping when iterations have reached a point for which

$$\frac{\left| x^{(t+1)} - x^{(t)} \right|}{\left| x^{(t)} \right|} < \epsilon.$$

This criterion enables the specification of a target specification of a target precision (e.g., within 1%) without worrying about the units of x.

Preference of Convergence Criterion

- Preference between the absolute and relative convergence criteria depends on the problem at hand.
- If the scale of x is huge (or tiny) relative to ϵ, an absolute convergence criterion may stop iterations too reluctantly (or too soon).
- The relative convergence criterion corrects for the scale of x, but can become unstable if $x^{(t)}$ values (or true solution) lie too close to zero.
- For the latter case, monitor relative convergence by stopping when

$$\frac{\left|x^{(t+1)} - x^{(t)}\right|}{\left|x^{(t)}\right| + \epsilon} < \epsilon.$$

Stopping Rule for Failure

- Stop after N iterations regardless of convergence.
- Stop if one or more convergence measures like $\left|x^{(t+1)} - x^{(t)}\right|$ or $\left|x^{(t+1)} - x^{(t)}\right| / \left|x^{(t)}\right|$ fail to decrease or cycle over several iterations.
- Stop if the procedure appears to be converging to a point at which $g(x)$ is inferior to another value you have already found. (Finding a false peak or local maximum.)
- Any indication of poor convergence behavior means that $x^{(t+1)}$ must be discarded that the algorithm somehow restarted.

Starting Points

- A bad starting vale can lead to divergence, cycling, discovery of a misleading local maximum or a local minimum or other problems.
- The outcome depends on g, the starting value, and the optimization algorithm tried.
- Methods for generating a reasonable starting values include graphing, preliminary estimates, educated guesses, and trial and error.
- Using a collection of runs from multiple starting values can be an effective way to gain confidence in your result.

Bisection Method

- Iterative root-finding algorithm
- If g' is continuous on $[a_0, b_0]$ and $g'(a_0)g'(b_0) \leq 0$, then there exists at least at least one $x^* \in [a_0, b_0]$ for which $g'(x^*) = 0$ and hence x^* is a local optimum of g.
- The bisection method systematically shrinks the interval from $[a_0, b_0]$ to $[a_1, b_1]$ to $[a_2, b_2]$ and so on, where $[a_0, b_0] \supset [a_1, b_1] \supset [a_2, b_2] \supset \cdots$ and so forth.

Bisection Method

- Let $x^{(0)} = \frac{a_0+b_0}{2}$ be the starting value. The updating equations are

$$[a_{t+1}, b_{t+1}] = \begin{cases} \left[a_t, x^{(t)}\right] & \text{if } g'(a_t)g'(x^{(t)}) \leq 0, \\ \left[x^{(t)}, b_t\right] & \text{if } g'(a_t)g'(x^{(t)}) > 0 \end{cases}$$

and

$$x^{(t+1)} = \frac{1}{2}\left(a_{t+1} + b_{t+1}\right).$$

Graphical Illustration: Bisection Method

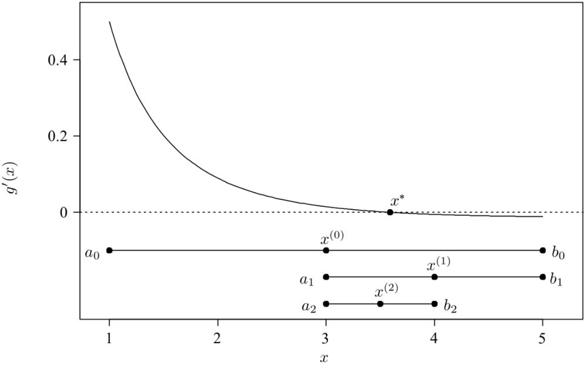

FIGURE 2.2 Illustration of the bisection method from Example 2.1. The top portion of this graph shows $g'(x)$ and its root at x^*. The bottom portion shows the first three intervals obtained using the bisection method with $(a_0, b_0) = (1, 5)$. The tth estimate of the root is at the center of the tth interval.

Newton's Method

Suppose that g' is continuously differentiable and that $g''(x^*) \neq 0$. At iteration t, the approach approximates $g'(x^*)$ by the Taylor series expansion:

$$0 = g'(x^*) \approx g'\left(x^{(t)}\right) + \left(x^* - x^{(t)}\right) g''\left(x^{(t)}\right).$$

Since g' is approximately by its tangent line at $x^{(t)}$, it seems sensible to approximate the root of g' by the root of the tangent line. Thus,

$$x^{(t+1)} = x^{(t)} - \frac{g'(x^{(t)})}{g''(x^{(t)})}.$$

Graphical Illustration: Newton's Method

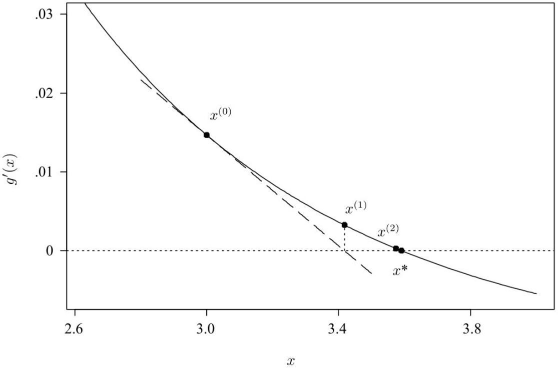

FIGURE 2.3 Illustration of Newton's method applied in Example 2.2. At the first step, Newton's method approximates g' by its tangent line at $x^{(0)}$, whose root $x^{(1)}$ serves as the next approximation of the true root x^*. The next step similarly yields $x^{(2)}$, which is already quite close to x^*.

Example: Newton's Method

Recall our motivating example. The Newton-Raphson increment for this problem is given by

$$\frac{g'(x^{(t)})}{g''(x^{(t)})} = \frac{\left(x^{(t)} + 1\right)\left(1 + 1/x^{(t)} - \log x^{(t)}\right)}{3 + 4/x^{(t)} + 1/\left(x^{(t)}\right)^2 - 2\log x^{(t)}}.$$

Starting from $x^{(0)} = 3.0$.

Example: Newton's Method

(Question) Find the root of the equation $f'(x) = e^{-x} - 5x = 0$.

(Answer)

$$f'(x) = e^{-x} - 5x, \qquad \text{and} \qquad f''(x) = -e^{-x} - 5.$$

The solution of the Newton-Raphson algorithm can be updated as

$$x^{(t+1)} = x^{(t)} - \frac{e^{-x} - 5x}{-e^{-x} - 5} = x^{(t)} + \frac{e^{-x} - 5x}{e^{-x} + 5}.$$

Implement the Newton-Raphson algorithm by yourselves. (Homework!!!)

Divergence of the Newton's Method

Whether Newton's method convergence depends on the shape of g and the starting value.

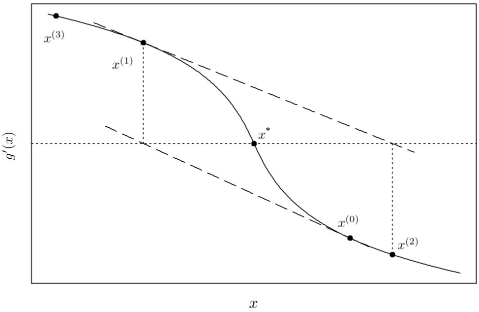

FIGURE 2.4 Starting from $x^{(0)}$, Newton's method diverges by taking steps that are increasingly distant from the true root, x^*.

Fisher Scoring

- Recall $I(\theta)$ can be approximated by $-l''(\theta)$. Therefore, when the optimization of g corresponds to an MLE problem, it is reasonable to replace $-l''(\theta)$ in the Newton update with $I(\theta)$.
- The updating equation of the Fisher scoring method is

$$\theta^{(t+1)} = \theta^{(t)} + l'(\theta^{(t)}) I(\theta^{(t)})^{-1},$$

  where $I(\theta^{(t)})$ is the expected Fisher information evaluated at $\theta^{(t)}$.
- Generally, Fisher scoring works better in the beginning to make rapid improvements, while Newton's method works better for refinement near the end.

Secant Method

- The updating increment for Newton-Raphson's method relies on the second derivatie, $g''(x^{(t)})$.
- If calculating the derivative is difficult, it might be replaced by the discrete-difference approximation. The result is the secant method,

$$x^{(t+1)} = x^{(t)} - g'\left(x^{(t)}\right) \frac{x^{(t)} - x^{(t-1)}}{g'\left(x^{(t)}\right) - g'\left(x^{(t-1)}\right)}$$

  for $t \geq 1$.
- This approach requires two starting points, $x^{(0)}$ and $x^{(1)}$.

Secant Method

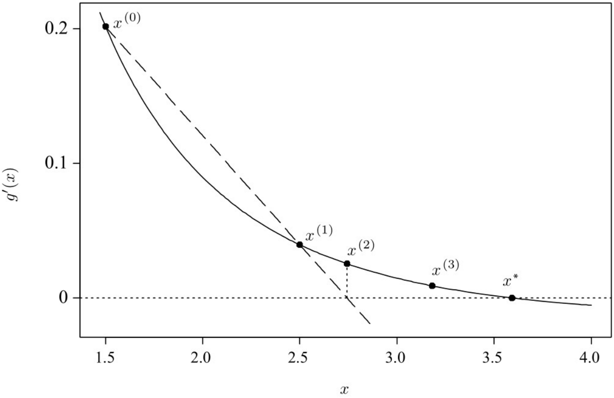

FIGURE 2.5 The secant method locally approximates g' using the secant line between $x^{(0)}$ and $x^{(1)}$. The corresponding estimated root, $x^{(2)}$, is used with $x^{(1)}$ to generate the next approximation.

Fixed Point Iteration

- The fixed-point strategy for finding roots is to determine a function G for which $g'(x) = 0$ if and only if $G(x) = x$.

$$g'(x) = 0 \quad \Rightarrow \quad g'(x) + x = x \quad \Rightarrow \quad G(x) = x.$$

  This transforms the problem of finding a root of g' into a problem of finding a fixed point of G.

- The simplest way to hunt for a fixed point is to use the updating equation $x^{(t+1)} = G\left(x^{(t)}\right)$. This yields the updating equation

$$x^{(t+1)} = x^{(t)} + g'\left(x^{(t)}\right).$$

- Scaling: When g'' is bounded and does not change sign on $[a, b]$, we can rescaling nonconvergent problems by choosing $G(x) = \alpha g'(x) + x$ for $\alpha \neq 0$.

Fixed Point Iteration

- Suppose an MLE is sought for the parameter of a quadratic log likelihood, l. Then, the score function is locally linear and l'' is roughly a constant.
- For quadratic log likelihoods, Newton's method would use the updating equation

$$\theta^{(t+1)} = \theta^{(t)} - \frac{l'(\theta)}{\gamma}.$$

- If we use scaled fixed-point iteration with $\alpha = -1/\gamma$, we get the same updating equation.
- Since many log-likelihoods are approximately locally quadratic, scaled fixed-point iteration can be very effective.

Fixed Point Iteration

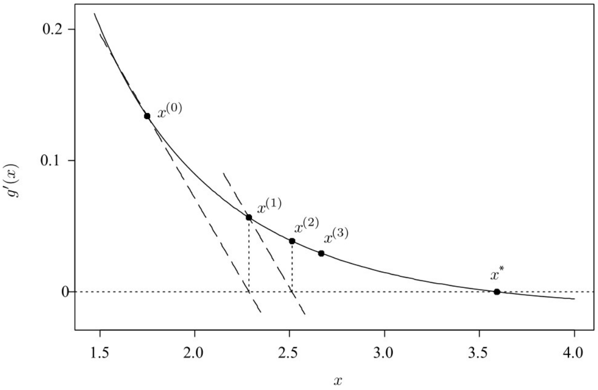

FIGURE 2.6 First three steps of scaled fixed-point iteration to maximize $g(x) = (\log x)/(1 + x)$ using $G(x) = g'(x) + x$ and scaling with $\alpha = 4$, as in Example 2.3.

Convergence Criteria

- Let $D(\mathbf{u},\mathbf{v})$ be a distance measure for p-dimensional vectors where

$$D(\mathbf{u},\mathbf{v}) = \sum_{i=1}^{p} \left| u_i - v_i \right|, \quad \text{and} \quad D(\mathbf{u},\mathbf{v}) = \sqrt{\sum_{i=1}^{p} (u_i - v_i)^2}.$$

- Absolute and relative convergence criteria can be formed from the inequalities

$$D(\mathbf{x}^{(t+1)},\mathbf{x}^{(t)}) < \epsilon, \quad \frac{D(\mathbf{x}^{(t+1)},\mathbf{x}^{(t)})}{D(\mathbf{x}^{(t+1)},\mathbf{0})} < \epsilon.$$

Newton's Method and Fisher Scoring

- Approximate $g(\mathbf{x}^*)$ by the Taylor series expansion

$$g(\mathbf{x}^*) = g(\mathbf{x}^{(t)}) + (\mathbf{x}^* - \mathbf{x}^{(t)})' g'(\mathbf{x}^{(t)}) + \frac{1}{2}(\mathbf{x}^* - \mathbf{x}^{(t)})' g''(\mathbf{x}^{(t)})(\mathbf{x}^* - \mathbf{x}^{(t)})$$

  and maximize this quadratic function with respect to $\mathbf{x}^*$.

- The gradient of the right-hand side of previous equation equal to zero yields

$$g'(\mathbf{x}^{(t)}) + g''(\mathbf{x}^{(t)})(\mathbf{x}^* - \mathbf{x}^{(t)}) = 0 \quad \text{and} \quad \mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - g''(\mathbf{x}^{(t)})^{-1} g'(\mathbf{x}^{(t)}).$$

- Multivariate Fisher scoring approach is given by

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + I(\boldsymbol{\theta}^{(t)})^{-1} l'(\boldsymbol{\theta}^{(t)}).$$

Example: Multivariate Newton's Methods

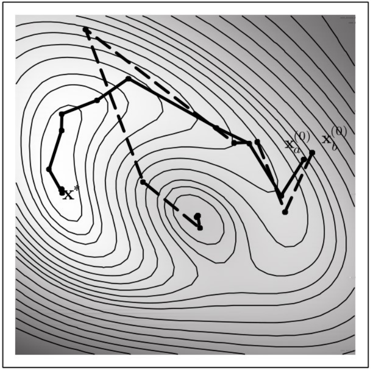

FIGURE 2.7 Application of Newton's method for maximizing a complicated bivariate function, as discussed in Example 2.4. The surface of the function is indicated by shading and contours, with light shading corresponding to high values. Two runs starting from $\mathbf{x}_a^{(0)}$ and $\mathbf{x}_b^{(0)}$ are shown. These converge to the true maximum and to a local minimum, respectively.

Iteratively Reweighted Least Squares

- Suppose the observed data consist of p covariate values $\mathbf{x}_i = (1, x_{i1}, \cdots, x_{ip})'$ and a binary response value y_i, for $i = 1, \cdots, n$. Let $\boldsymbol{\beta} = (\beta_0, \beta_1, \cdots, \beta_p)'$ denote a vector of parameter.
- GLM used for logistic regression is based on the Bernoulli distribution. Model the response variables as $y_i | x_i \sim \text{Bernoulli}(\pi_i)$.
- Let
$$\log \frac{\pi_i}{1-\pi_i} = \boldsymbol{\beta}'\mathbf{x}_i. \quad \Rightarrow \quad \pi_i = \frac{\exp(\boldsymbol{\beta}'\mathbf{x})}{1+\exp(\boldsymbol{\beta}'\mathbf{x})}.$$
Then, the likelihood function is
$$L(\beta) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i} = \prod_{i=1}^{n} \left(\frac{\exp(\boldsymbol{\beta}'\mathbf{x}_i)}{1+\exp(\boldsymbol{\beta}'\mathbf{x}_i)}\right)^{y_i} \left(\frac{1}{1+\exp(\boldsymbol{\beta}'\mathbf{x}_i)}\right)^{1-y_i}$$

Iteratively Reweighted Least Squares

- The log-likelihood is
$$l(\beta) = \mathbf{y}'\mathbf{X}\boldsymbol{\beta} - \mathbf{b}'\mathbf{1},$$
where $\mathbf{1}$ is a column vector of 1, $\mathbf{y} = (y_1, \cdots, y_n)'$,
$\mathbf{b} = (\log\{1 + \exp(\mathbf{x}_1'\boldsymbol{\beta})\}, \cdots, \log\{1 + \exp(\mathbf{x}_n'\boldsymbol{\beta})\})$, and $\mathbf{X}$ is the $n \times (p+1)$ matrix whose i-th row is $\mathbf{x}_i'$.
- The score function is
$$l'(\beta) = \mathbf{X}'(\mathbf{y} - \boldsymbol{\pi}),$$
where $\boldsymbol{\pi} = (\pi_1, \cdots, \pi_n)'$ and the Hessian matrix is given by
$$l''(\beta) = \frac{d}{d\beta}\mathbf{X}'(\mathbf{y} - \boldsymbol{\pi}) = -X'WX,$$
where W is a diagonal matrix with i-th diagonal entry equal to $\pi_i(1 - \pi_i)$.

Iteratively Reweighted Least Squares

- Therefore, Newton's update is

$$\begin{aligned}\boldsymbol{\beta}^{(t+1)} &= \boldsymbol{\beta}^{(t)} - I\left(\boldsymbol{\beta}^{(t)}\right)^{-1} l'\left(\boldsymbol{\beta}^{(t)}\right) \\ &= \boldsymbol{\beta}^{(t)} + \left(X'W^{(t)}X\right)^{-1}\left(X'\left(\mathbf{y} - \boldsymbol{\pi}^{(t)}\right)\right),\end{aligned}$$

where $\boldsymbol{\pi}^{(t)}$ is the value of $\boldsymbol{\pi}$ corresponding to $\boldsymbol{\beta}^{(t)}$, and $W^{(t)}$ is the diagonal weight matrix evaluated at $\boldsymbol{\pi}^{(t)}$.

Netwon-Like Methods

Some very effective methods rely on updating equations of the form

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - (\mathbf{M}^{(t)})^{-1} g'(\mathbf{x}^{(t)}),$$

where $\mathbf{M}^{(t)}$ is a $p \times p$ matrix approximating the Hessian, $g''(\mathbf{x}^{(t)})$.

In general optimization problems, there are several good reasons to consider replacing the Hessian by some simpler approximation.

- It may be computationally expensive to evaluate the Hessian.
- The steps taken by Newton's method are not necessarily always uphill: At each iteration, there is no guarantee that $g(\mathbf{x}^{(t+1)}) > g(\mathbf{x}^{(t)})$. A suitable $\mathbf{M}^{(t)}$ can guarantee ascent.

Ascent Algorithms

- To force uphill steps, one could resort to an ascent algorithm.
- The method of steepest ascent is obtained with the Hessian replacement $\mathbf{M}^{(t)} = -\mathbf{I}$, where $\mathbf{I}$ is the identity matrix.
- Since the gradient of g indicates the steepest direction uphill on the surface of g at the point $\mathbf{x}^{(t)}$, setting $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + g'(\mathbf{x}^{(t)})$ amounts to taking a step in the direction of steepest ascent.
- Scaled steps of the form $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \alpha^{(t)} g'(x(t))$ for some $\alpha^{(t)} > 0$ can be helpful for controlling convergence.
- If $-\mathbf{M}^{(t)}$ is positive definite, ascent can be assured by choosing $\alpha^{(t)}$ sufficiently small, yielding $g(\mathbf{x}^{(t+1)}) - g(\mathbf{x}^{(t)}) > 0$.

Ascent Algorithms

- Therefore, an ascent algorithm involves a positive definite matrix $-\mathbf{M}^{(t)}$ to approximate the negative Hessian, and a contraction or step length parameter $\alpha^{(t)} > 0$ whose value can shrink to ensure ascent at each step.
- For example,
  - Start each step with $\alpha^{(t)} = 1$. If the original step turns out to be downhill, $\alpha^{(t)}$ can be halved. This is called backtracking.
  - If the step is still downhill, $\alpha^{(t)}$ is halved again until a sufficiently small step is found to be uphill.
- For Fisher scoring, $-\mathbf{M}^{(t)} = I(\boldsymbol{\theta}^{(t)})$, which is positive semidefinite. Therefore backtracking with Fisher scoring would avoid stepping downhill.

Discrete Netwon and Fixed-Point Methods

- To avoid calculating the Hessian, one could resort to a secant-like method, yielding a discrete Newton method, or rely solely on an initial approximation, yielding a multivariate fixed-point method.
- Multivariate fixed-point methods use an initial approximation of g'' throughout the iterative updating.
- If this approximation is a matrix of constants, so $\mathbf{M}^{(t)} = \mathbf{M}$ for all t, then the updating equation is

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \mathbf{M}^{-1} g'(\mathbf{x}^{(t)}).$$

- A reasonable choice for $\mathbf{M}$ is $g''(\mathbf{x}^{(0)})$. Notice that if $\mathbf{M}$ is diagonal, then this amounts to applying the univariate scaled fixed-point algorithm separately to each component of g.

Gauss-Netwon Method

Nonlinear least squares problems with observed data $(y_i, \mathbf{z}_i)$ for $i = 1, \cdots, n$.

- Seeks to estimate θ by maximizing an objective function

$$g(\boldsymbol{\theta}) = -\sum_{i=1}^{n} (y_i - f(\mathbf{z}_i, \boldsymbol{\theta}))^2 .$$

- Such objective functions might be sensibly used, when estimating θ to fit the model

$$Y_i = f(\mathbf{z}_i, \boldsymbol{\theta}) + \epsilon_i$$

  for some nonlinear function f and random error ϵ_i.

Rather than approximate g, the Gauss-Newton approach approximates f itself by its linear Taylor series expansion about $\boldsymbol{\theta}^{(t)}$. Replacing f by its linear approximation yields a linear least squares problem, which can be solved to derive an update $\boldsymbol{\theta}^{(t+1)}$.

Gauss-Netwon Method

- The nonlinear model can be approximated by

$$Y_i \approx f\left(\mathbf{z}_i, \boldsymbol{\theta}^{(t)}\right) + \left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}\right)' f'\left(\mathbf{z}_i, \boldsymbol{\theta}^{(t)}\right) + \epsilon_i = \tilde{f}\left(\mathbf{z}_i, \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}\right) + \epsilon_i.$$

- A Gauss-Newton step is derived from the maximization of

$$\tilde{g}(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \left(y_i - \tilde{f}\left(\mathbf{z}_i, \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}\right)\right)^2$$

with respect to $\boldsymbol{\theta}$, whereas a Newton step is derived from the maximization of a quadratic approximation to g itself,

$$g\left(\boldsymbol{\theta}^{(t)}\right) + \left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}\right)' g'\left(\boldsymbol{\theta}^{(t)}\right) + \left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}\right)' g''\left(\boldsymbol{\theta}^{(t)}\right)\left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}\right).$$

Gauss-Netwon Method

Let $X_i^{(t)}$ denote a working response whose observed value is

$$x_i^{(t)} = y_i - f\left(\mathbf{z}_i, \boldsymbol{\theta}^{(t)}\right),$$

and define $\mathbf{a}_i^{(t)} = f'\left(\mathbf{z}_i, \boldsymbol{\theta}^{(t)}\right)$. Then the approximated problem can be reexpressed as minimizing the squared residuals of the linear regression model

$$\mathbf{X}^{(t)} = \mathbf{A}^{(t)}\left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}\right) + \epsilon. \tag{3}$$

Gauss-Netwon Method

The minimal squared error for fitting equation (3) is achieved when

$$\left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}\right) = \left(\left(A^{(t)}\right)' A^{(t)}\right)^{-1} \left(A^{(t)}\right)' \mathbf{x}^{(t)}.$$

Thus, the Gauss-Newton update for $\boldsymbol{\theta}^{(t)}$ is

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \left(\left(A^{(t)}\right)' A^{(t)}\right)^{-1} \left(A^{(t)}\right)' \mathbf{x}^{(t)}.$$

- The potential advantage of the Gauss-Newton method is that it does not require computation of the Hessian.
- It is fast when f is nearly linear or when the model fits well.
- In other situations, particularly when the residuals at the true solution are large because the model fits poorly, the method may converge very slowly or not at all-even from good starting values.