STA6171: Statistical Computing for DS 1

Solving Nonlinear Equations

Ick Hoon Jin

Yonsei University, Department of Statistics and Data Science

2020.09.09

Optimization. ⇒ Finding maximum or minimum

⇒ Calculate the derivative

$f'(x) = 0$

↳ analytic form

1. Motivating Example
2. Univariate Optimization
3. Multivariate Optimization

⇒ When $f'(x) = 0$ does not have an analytic form, how we can find a solution

Motivaing Example

- A simple univariate numerical optimization problem is to maximize

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

with respect to x.

- The derivative of $g(x)$ is

optim function

$$g'(x) = \frac{1 + \frac{1}{x} - \log x}{(1+x)^2} = 0 \tag{2}$$

denominator includes x

and no analytic solution can be found in maximization.

find maximum

numerical optimization

no analytic solution.

ex) MLE

Motivaing Example

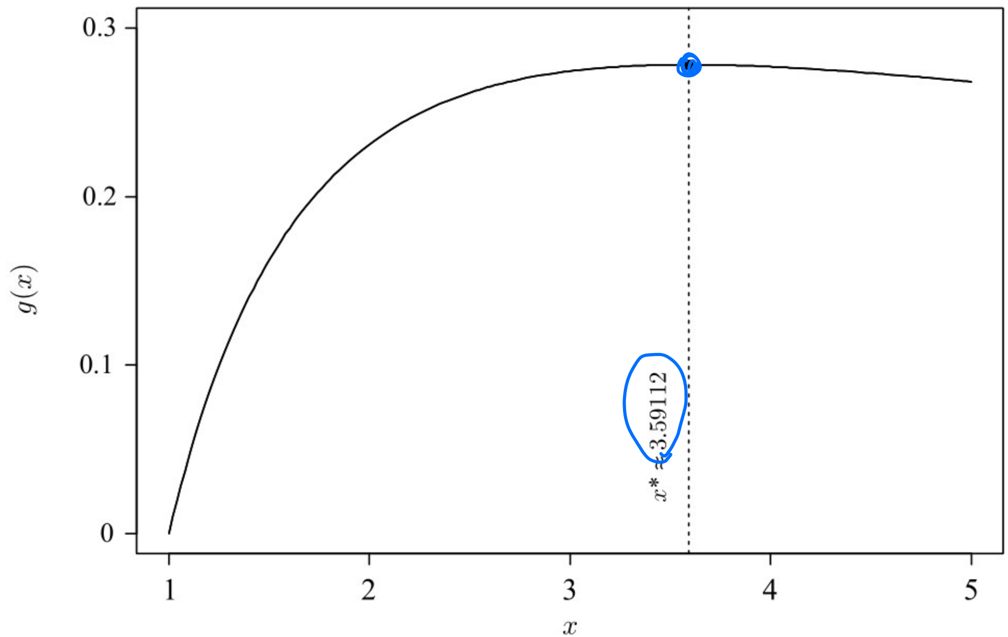


FIGURE 2.1 The maximum of $g(x) = (\log x)/(1 + x)$ occurs at $x^* \approx 3.59112$, indicated by the vertical line.

Convergence Criterion

↳ optimization ⇒ solution → numerical optimization
↳ convergence criterion. we use computer
⇒ We cannot run algorithm forever.

- We cannot allow the algorithm to run indefinitely, so we require a *stopping rule* based on $\boxed{\text{convergence criteria}}$, to trigger an end to the successive approximation.

stop running algorithm

check

- 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.

when newly updated $X^{(t+1)}$ satisfies the stopping rule, we stop running optimization algorithm.

- Two reasons to stop if the algorithm appears
  → Converge.
  - to have achieved satisfactory convergence.
  - unlikely to converge soon.

↳ Diverge.

Convergence Criterion

$g'(x^{(t+1)}) \approx 0$

Object : $g'(x) = 0$ If $g'(x^{(t)}) \approx 0$, can we declare it converge?

By ~~algor~~ types of algorithm, there exists a large changes

- 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. from $x^{(t)} \to x^{(t+1)}$
- 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. the change of $x^{(t)} \to x^{(t+1)}$ small $\Rightarrow g'(x^{(t+1)}) \approx 0$
- Assess convergence by monitoring $\left|x^{(t+1)} - x^{(t)}\right|$ and use $g'\left(x^{(t+1)}\right)$ as a backup check.

Convergence criterion : $\boxed{\left|x^{(t+1)} - x^{(t)}\right| < \varepsilon}$

stopping rule

Then we declare it converges.

Types of Convergence Criterion

- Absolute convergence criterion mandates stopping when

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

stopping rule
tolerable imprecision.

where ϵ is a constant chosen to indicate tolerable imprecision.

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

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

Why we need relative convergence criterion?

scaling.

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

$\frac{|x^{(t+1)} - x^{(t)}|}{|x^{(t)}|}$ ≈ 0 then $|x^{(t+1)} - x^{(t)}|$ will be large $\Rightarrow$ algorithm will be unstable.

$\varepsilon = 0.001$

0.000005

- Preference between the absolute and relative convergence criteria depends on the problem at hand.

If your function are affected a lot by scale, you cannot use the absolute conv. crit.

- If the scale of x is huge (or tiny) relative to ϵ, an absolute convergence criterion may stop iterations too reluctantly (or too soon).

relative

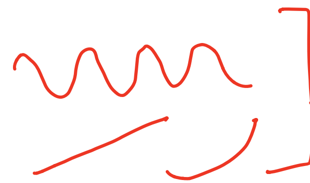
- The relative convergence criterion corrects for the scale of x, but can become unstable if $x^{(t)}$ values (or true solution) lie too close to zero.
- For the latter case, monitor relative convergence by stopping when

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

reduce the unstability of relative convergence criterion

Stopping Rule for Failure

Monitor Convergence Criterion. $|x^{(t+1)} - x^{(t)}|$ cycle
$|x^{(t+1)} - x^{(t)}|$



→ We have to set N suff. large.

- 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.

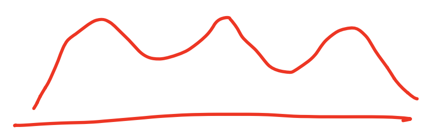
fail to converge

- 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.)

We already found some other point
f(Convergence point) < f(Other point)

- Any indication of poor convergence behavior means that $x^{(t+1)}$ must be discarded that the algorithm somehow restarted.

local maximum
optimum) → Multi-modality



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.

  Outcome ← function
  the starting point
  algorithm
- 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.

Convergence criterion
Starting points

Bisection Method

Single parameter.

$\boxed{\text{Bisection. Method}}$

- 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

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

$g'(a)$ & $g'(x^{(0)})$ both ~~positive~~ negative

non-maximum (minimum)

Only single mode.

$g'(a_0)\, g'(x^{(0)}) > 0$

$g'(x^{(0)})\, g'(b_0) < 0$

$g'(a_1)\, g'(x^{(1)}) < 0$

$g'(x^{(1)})\, g'(b_1) > 0$

- Only applicable to single mode

- Slow

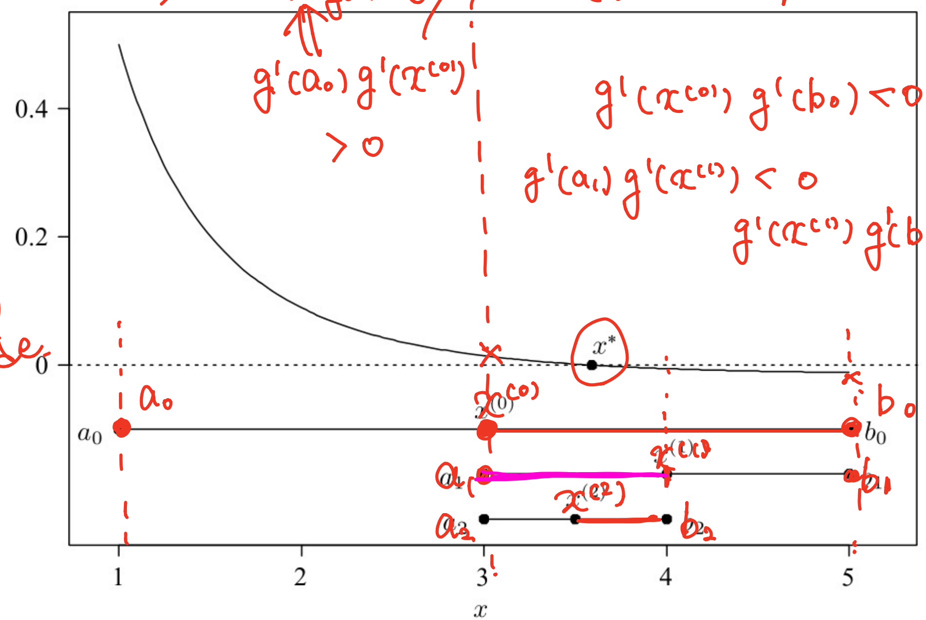


FIGURE 2.2 Illustration of the bisection method from Example 2.1. The top portion of this graph shows $g'(x)$ and its root at x^*. The bottom portion shows the first three intervals obtained using the bisection method with $(a_0, b_0) = (1,\ 5)$. The tth estimate of the root is at the center of the tth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

Newton's Method

$\underbrace{g'(x^*) = 0}$

$g'(x^*) \approx g'(x^{(t)}) + (x^* - x^{(t)})\, g''(x^{(t)})$ $\cancel{+ \frac{1}{2}(x^* - x^{(t)})^2 g'''(x^{(t)})}$ + Remainder

Taylor Series Expansion.

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,

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

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

$$x^* - x^{(t)} = -\frac{g'(x^{(t)})}{g''(x^{(t)})}$$

$$\underbrace{x^{(t+1)} = x^{(t)} - \frac{g'(x^{(t)})}{g''(x^{(t)})}}_{\text{Updating Equation of Newton-Raphson Algorithm}}$$

Graphical Illustration: Newton's Method

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

(1) Obtained by Taylor Series Expansion

(2)

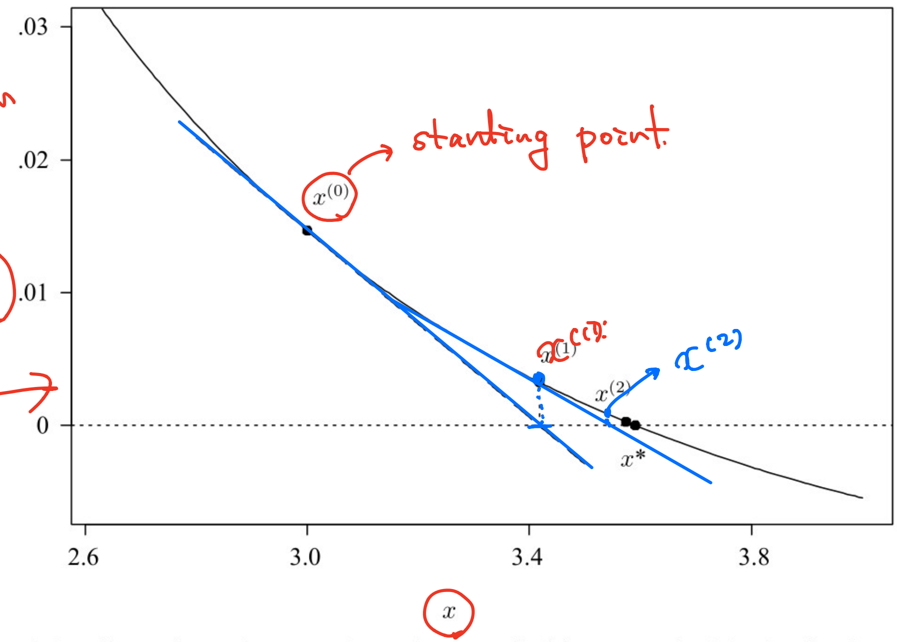


FIGURE 2.3 Illustration of Newton's method applied in Example 2.2. At the first step, Newton's method approximates g' by its tangent line at $x^{(0)}$, whose root $x^{(1)}$ serves as the next approximation of the true root x^*. The next step similarly yields $x^{(2)}$, which is already quite close to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

Convergence of Newton's Method — shape of g. / starting value.

Disadvantage : need to calculate the second derivative.

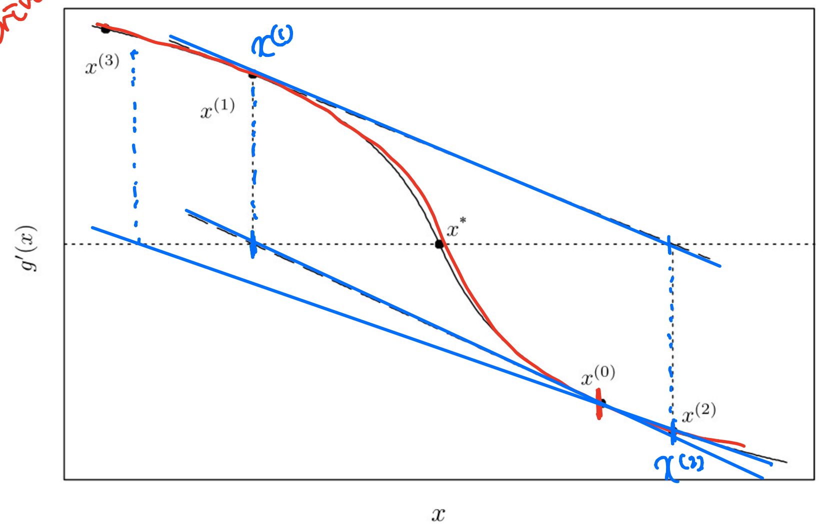


FIGURE 2.4 Starting from $x^{(0)}$, Newton's method diverges by taking steps that are increasingly distant from the true root, x^{*}.

Need to run with multiple starting points.

Fisher Scoring

We use for finding the MLE. $\Rightarrow l'(\theta) = 0 \Rightarrow$ when we use Newton-Raphson method, $l''(\theta)$ related to Fisher information

Variant of Newton-Raphson method.

- 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 $I(\theta)$.

$I(\theta) = -l''(\theta)$

we replace the second derivative term with Fisher information

- 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 $\theta^{(t)}$.

$\Rightarrow$ Fisher scoring method

- Generally, Fisher scoring works better in the beginning to make rapid improvements, while Newton's method works better for refinement near the end.

$\theta^{(t+1)} = \theta^{(t)} - \frac{l'(x^{(t)})}{l''(x^{(t)})}$ Fisher information $= I(\theta^{(t)})$

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

Secant Method

↳ Variant of Newton-Raphson method
⇒ We need to calculate the second derivative

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

- The updating increment for Newton-Raphson's method relies on the second derivatie, $g''(x^{(t)})$.

  instead of calculating the second derivative, use the definition of derivative

- If calculating the derivative is difficult, it might be replaced by the discrete-difference approximation. The result is the secant method,

  → Updating equation of Secant method.

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

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

for $t \geq 1$.

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

Secant Method

Disadvantage
=> We need two starting points.

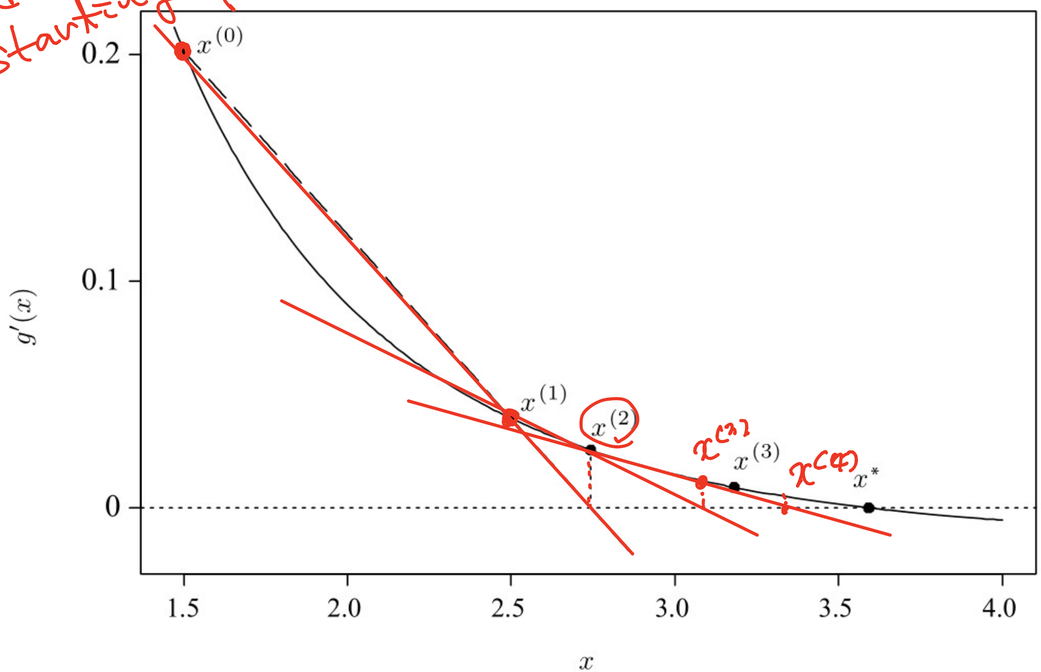


FIGURE 2.5 The secant method locally approximates g' using the secant line between $x^{(0)}$ and $x^{(1)}$. The corresponding estimated root, $x^{(2)}$, is used with $x^{(1)}$ to generate the next approximation.

Fixed Point : Fixed points of a function is an element of the function's domain that is mapped to itself by the function.

$\boxed{f(c) = c}$ c is a fixed point

(Ex) $f(x) = x^2 - 3x + 4 = x$ fixed point $x = 2$

$x^2 - 4x + 4 = 0$

$(x-2)^2 = 0$ $x = 2$

Do all functions have fixed points? No

$f(x) = \underline{x+1} \neq x$

$0 \neq 1$ $\Rightarrow$ No fixed point.

Fixed Point Iteration

$g'(x) = 0$ $\quad G(x) = x$ → has fixed point

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

  $\alpha g'(x) = 0$

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

  $\Rightarrow g'(x) + x = x \Rightarrow G(x) = x$

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

  → $g'(x) + x$. will be used as a updating equation.

- 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

  $x^{(t+1)} = x^{(t)} + g'(x^{(t)})$

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

  or $= x^{(t)} + \alpha g'(x^{(t)})$

- 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 $\alpha \neq 0$. $\Rightarrow$ Newton-Raphson method

  $\Rightarrow$ Variant of fixed-point iteration

Fixed Point Iteration

$\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$

→ normal density

- 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.

  ↪ first derivative

  ↪ γ

- For quadratic log likelihoods, Newton's method would use the updating equation

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

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

$= \theta^{(t)} + \alpha l'(\theta)$

- If we use scaled fixed-point iteration with $\alpha = -1/\gamma$, we get the same updating equation.

  $\alpha = -1/\gamma$

  fixed point iteration method

- Since many log-likelihoods are approximately locally quadratic, scaled fixed-point iteration can be very effective.

Fixed Point Iteration

Newton-Raphson ⇒ tangent line.

Fixed-point iteration ⇒ line with same slope

~~Additional~~ (HW) $g(x) = \frac{\log x}{1+x}$, $1 < x < 5$,

find maximum using

(1) bisection

(2) Newton

(3) Secant

(4) fixed point iteration method

Much slower than Newton-Raphson method

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

slope α.

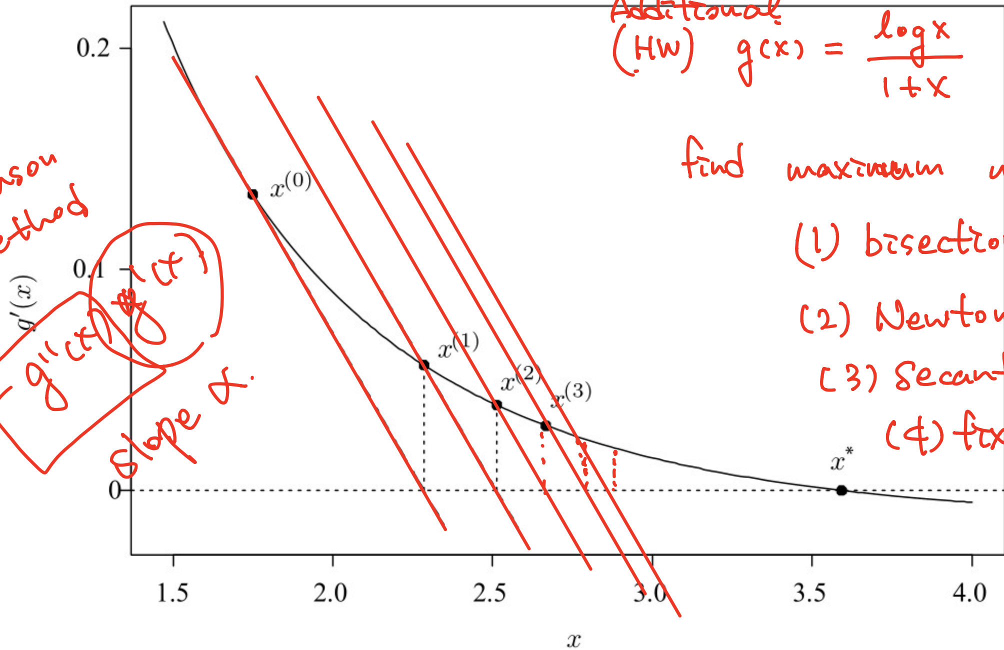


FIGURE 2.6 First three steps of scaled fixed-point iteration to maximize $g(x) = (\log x)/(1+x)$ using $G(x) = g'(x) + x$ and scaling with $\alpha = 4$, as in Example 2.3.

Convergence Criteria

distance between $\boxed{x^{(t+1)} \& x^{(t)}}$. x_i^2

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

absolute difference

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

$|x_i - 0|$ Euclidean distance

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

Absolute Convergence Relative

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

$x^{(t)}$ & 0 distance

$g'(x) = 0$

Newton's Method and Fisher Scoring

small boldic letter : vector
Capital " " : Matrix

- 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

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

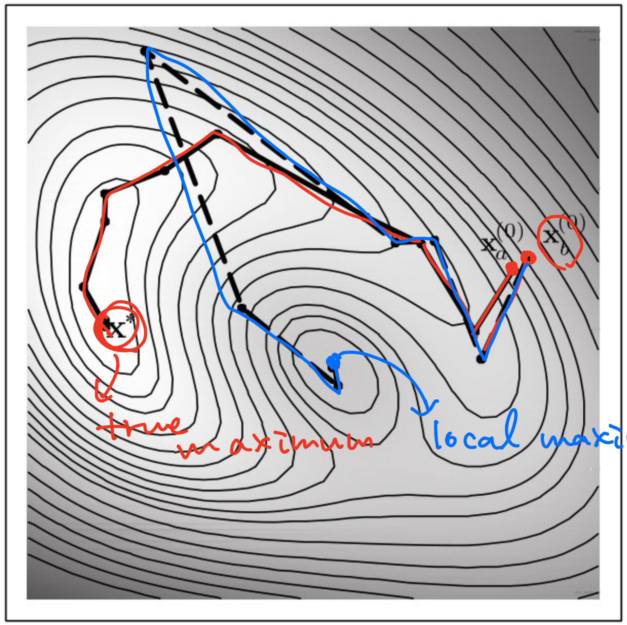
$$g'(\mathbf{x}^{(t)}) + g''(\mathbf{x}^{(t)})(\mathbf{x}^* - \mathbf{x}^{(t)}) = 0 \quad \text{and} \quad \mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - g''(\mathbf{x}^{(t)})^{-1} g'(\mathbf{x}^{(t)}).$$

vector

- Multivariate Fisher scoring approach is given by

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

Example: Multivariate Newton's Methods



Multivariate case

$\Rightarrow$ Multimodality

more severe than univariate case

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

↳ Generalize Linear Model ⇒ Parameter Estimation.

- 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 $\cancel{\boldsymbol{\beta} = (\beta_0, \beta_1, \cdots, \beta_p)}'$ denote a vector of parameter. x_i
- Logistic regression response y binary
  GLM used for logistic regression is based on the Bernoulli distribution. Model the response variables as $y_i | x_i \sim \text{Bernoulli}(\pi_i)$. $y_i | X_i \sim \text{Bernoulli}(\pi_i)$
- Let binary response probability $\pi \in [0, 1]$ → map π_i into $[-\infty, \infty]$ $\boxed{\text{link function.}}$

$$\log \frac{\pi_i}{1-\pi_i} = \boldsymbol{\beta}' \mathbf{x}_i. \quad \Rightarrow \quad \pi_i = \frac{\exp(\boldsymbol{\beta}'\mathbf{x})}{1+\exp(\boldsymbol{\beta}'\mathbf{x})}.$$

Binary ⇒ logit link (probit

plug-in Bernoulli dist

Then, the likelihood function is

$$L(\beta) = \prod_{i=1}^{n} \pi_i^{y_i} (1-\pi_i)^{1-y_i} = \left[\prod_{i=1}^{n} \left(\frac{\exp(\boldsymbol{\beta}'\mathbf{x}_i)}{1+\exp(\boldsymbol{\beta}'\mathbf{x}_i)} \right)^{y_i} \left(\frac{1}{1+\exp(\boldsymbol{\beta}'\mathbf{x}_i)} \right)^{1-y_i} \right]$$

$$\log \frac{\pi_i}{1-\pi_i} = \beta' x_i \Rightarrow \pi_i = \frac{\exp(\beta' x)}{1+\exp(\beta' x)}$$

Find β that maximize the likelihood

Iteratively Reweighted Least Squares

$y^T X\beta - b^T 1$

$b^T = \begin{bmatrix} \log(1+\exp(x_1^T\beta)) \\ \log(1+\exp(x_2^T\beta)) \\ \vdots \\ \log(1+\exp(x_n^T\beta)) \end{bmatrix}$

- The log-likelihood is

$$l(\beta) = \mathbf{y}'\mathbf{X}\beta - \mathbf{b}'\mathbf{1},$$

where **1** is ~~a column vector of 1~~, $\mathbf{y} = (y_1, \cdots, y_n)'$, $\mathbf{b} = (\log\{1+\exp(\mathbf{x}_1'\beta)\}, \cdots, \log\{1+\exp(\mathbf{x}_n'\beta)\})$, and $\mathbf{X}$ is the $n \times (p+1)$ matrix whose i-th row is $\mathbf{x}_i'$.

$\pi = \begin{bmatrix} \frac{\exp(x_1^T\beta)}{1+\exp(x_1^T\beta)} \\ \frac{\exp(x_2^T\beta)}{1+\exp(x_2^T\beta)} \\ \vdots \\ \frac{\exp(x_n^T\beta)}{1+\exp(x_n^T\beta)} \end{bmatrix}$

$l'(\beta) = X^T(y - \pi)$

- 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 $\pi_i(1-\pi_i)$.

$$W = \begin{bmatrix} \frac{\exp(x_i^T\beta)}{1+\exp(x_i^T\beta)} \cdot \frac{1}{1+\exp(x_i^T\beta)} & & 0 \\ & \ddots & \\ 0 & & \end{bmatrix}$$

$$= \begin{bmatrix} \pi_1(1-\pi_1) & & & 0 \\ & \pi_2(1-\pi_2) & & \\ & & \ddots & \\ 0 & & & \pi_n(1-\pi_n) \end{bmatrix}$$

$\rightarrow$ Determined by β $\quad \frac{\exp(x^T\beta)}{1+\exp(x^T\beta)}$

$\begin{pmatrix} \pi_i \\ W \end{pmatrix}$

$$l''(\beta) = -X^T W X \qquad -[l''(\beta)]^{-1}$$

$$l'(\beta) = X^T(y-\pi)$$

$\rightarrow$ Updating equation from Newton-Raphson method

$$\boxed{\beta^{(t+1)} = \beta^{(t)} + \underline{[X^T W^{(t)} X]^{-1} X^T (y-\pi^{(t)})}}$$

$$(X^TX)^{-1}X^Ty$$

$\Rightarrow$ Iteratively reweighted least square

Iteratively Reweighted Least Squares

Starting points

Delay hw 1 due 9/16 → 9/23

- Therefore, Newton's update is

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

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 $\boldsymbol{\pi}^{(t)}$.

Similarly, you can derive IRLS for Poisson regression (HW). 9/9