Optimization and Solving Non-linear Equations

Module DES130: Computational Statistics

Dr. Erick A. Chacón Montalván (echacon@uni.edu.pe)

Escuela de Profesional de Ingeniería Estadística
Facultad de Ingeniería Económica, Estadística y Ciencias Sociales
Universidad Nacional de Ingeniería (UNI)
Lima – Perú

Introduction

Optimization

Let the function $f : A \to \mathcal{R}$.

Minimization:

$$\text{Find } x_o \in A \ / \ f(x_o) \leq f(x) \text{ for all } x \in A.$$

Maximization:

$$\text{Find } x_o \in A \ / \ f(x_o) \geq f(x) \text{ for all } x \in A.$$

Notes:

- Maximizing $f(x)$ is equivalent to minimizing $-f(x)$.
- Local and global optimizers.
- It is not easy to distinguish local and global optimizers.
- Subfields of optimization depend on the properties of $f(x)$

Solving non-linear equations

Optimization is intimately linked with solving non-linear equations (Givens and Hoeting 2012).

Optimization as root-finding problem:

Let the function $f : A \to \mathcal{R}$. Then, finding

$$x_0 \in A \ / \ f(x_o) \leq f(x) \text{ for all } x \in A.$$

Is equivalent to finding

$$x_0 \in A \ / \ f'(x_o) = 0 \text{ and } f''(x_0) > 0.$$

Common optimization types

Linear programming:

The function $f(x)$ and restrictions are linear. Example, estimation of the coefficients of a quantile regression.

Quadratic programming:

The function $f(x)$ can be a quadratic function. Example, least squares estimation.

Non-linear programming:

The function $f(x)$ is non-linear. Example, maximum likelihood estimation.

Quadratic programming

Motivation: least squares

Let consider the following linear model $y_i = \boldsymbol{x}_i^\top \beta + e_i$ for $i = 1, \ldots, n$, then the least squares estimation is done by the optimization

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^\top \beta)^2.$$

It can be shown that $\hat{\beta} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$. However, this expression in not really computed in practice (Wood 2017).

Consider the QR decomposition:

$$\boldsymbol{X}_{n \times p} = \boldsymbol{Q}_{n \times n} \begin{bmatrix} \boldsymbol{R}_{p \times p} \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{Q}_f \boldsymbol{R}, \quad \text{and} \quad \boldsymbol{Q}^\top \boldsymbol{y} = \begin{bmatrix} \boldsymbol{f}_{p \times 1} \\ \boldsymbol{r} \end{bmatrix},$$

where $\boldsymbol{R}$ is a upper triangular matrix and $\boldsymbol{Q}$ is an orthogonal matrix.

Least squares in practice

Using the matricial form of the loss function:

$$
\begin{aligned}
L(\beta) &= \|\boldsymbol{y} - \boldsymbol{X}\beta\|^2 \\
&= \|\boldsymbol{Q}^\intercal \boldsymbol{y} - \boldsymbol{Q}^\intercal \boldsymbol{X}\beta\|^2 \\
&= \left\| \boldsymbol{Q}^\intercal \boldsymbol{y} - \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{bmatrix} \beta \right\|^2 \\
&= \left\| \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{r} \end{bmatrix} - \begin{bmatrix} \boldsymbol{R}\beta \\ \boldsymbol{0} \end{bmatrix} \right\|^2 \\
L(\beta) &= \|f - \boldsymbol{R}\beta\|^2 + \|r\|^2 .
\end{aligned}
$$

Which is minimized by $\hat{\beta} = \boldsymbol{R}^{-1}\boldsymbol{f}$. A similar algorithm can be found for generalized least squares (Wood 2017).

Linear programming

Motivation: quantile regression

Let Y_i with $Pr(Y_i \leq Q_p) = p$ such as the conditional quantile is

$$Q_p(Y_i \mid \boldsymbol{x}_i) = \boldsymbol{x}_i^\top \beta_p .$$

Estimation fot the p-th quantile regression, defined by , is done by the optimization

$$\hat{\beta}_p = \arg\min_{\beta_p} \left(\sum_{i:\, y_i \geq \boldsymbol{x}_i^\top \beta_p} p \left| y_i - \boldsymbol{x}_i^\top \beta_p \right| + \sum_{i:\, y_i < \boldsymbol{x}_i^\top \beta_p} (1-p) \left| y_i - \boldsymbol{x}_i^\top \beta_p \right| \right) .$$

Quantile regression as a linear program

Given $\boldsymbol{y} - \boldsymbol{X}\beta_p = \boldsymbol{e} = \boldsymbol{r}^+ - \boldsymbol{r}^-$, where $r_i^+ = \max(0, e_i)$ and $r_i^- = \max(0, -e_i)$. Then the estimation of quantile regression can be expressed as a linear program:

Linear programming problem:

$$\min_{\boldsymbol{r}^+, \boldsymbol{r}^-, \beta_p} \left(p\mathbf{1}^\top \boldsymbol{r}^+ + (1-p)\mathbf{1}^\top \boldsymbol{r}^-\right)$$

subject to

$$y = \boldsymbol{X}\beta_p + \boldsymbol{r}^+ - \boldsymbol{r}^-,$$
$$\boldsymbol{r}^+ \in \mathbb{R}_+^n, \text{ and } \boldsymbol{r}^- \in \mathbb{R}_+^n.$$

Non-linear programming

Motivation: generalized linear models

Exponential family:

$$f_{Y_i}(y_i) = \exp\left\{\frac{y_i\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right\}$$

Let consider the following model:

$$Y_i \mid \boldsymbol{x}_i \sim Bernoulli(\pi_i),$$
$$\text{logit}\,(\pi_i) = \boldsymbol{x}_i^\top \beta.$$

Given that $f(y_i) = \exp\left\{y_i \log\left(\frac{\pi_i}{1-\pi_i}\right) + \log(1-\pi)\right\}$, then

$$\theta_i = \text{logit}\,(\pi_i) = \boldsymbol{x}_i^\top \beta,$$
$$b(\theta) = \log(1-\pi_i) = \log(1 + \exp(\theta_i)),$$
$$a(\phi) = 1, \text{ and } c(y_i, \phi) = 0.$$

Taylor series approximation

It can be seen as a approximation of a function $f : \mathbb{R} \to \mathbb{R}$ by a polinomial function $g(x)$ around the point x_o.

For $r = 0 \to g(x) = c_0$, \ **given** $g(x_0) = f(x_0)$, \ **then** $g(x) = f(x_0)$.

For $r = 1 \to g(x) = c_0 + c_1 x$, \ **given** $g(x_0) = f(x_0)$, and $g'(x_0) = f'(x_0)$, \ **then** $g(x) = f(x_0) + f'(x_0)(x - x_0)$.

For $r = 2 \to g(x) = c_0 + c_1 x + c_2 x^2$, \ **given** $g(x_0) = f(x_0)$, $g'(x_0) = f'(x_0)$, and $g''(x_0) = f''(x_0)$, \ **then** $g(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2$.

Taylor series approximation

In general:

Let $f(x)$ be n-th differentiable, then the n-th order Taylor approximation of $f(x)$ around x_0 is

$$g(x) = \sum_{r=0}^{n} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n.$$

Example:

Obtain the first order Taylor approximation of $f(x) = \exp(x)$ around 0.

Solución:

Notice that $x_0 = 0$ such that $f(0) = \exp(0) = 1$ $f'(0) = \exp(0) = 1$.

Hence $g(x) = 1 + 1(x - 0) = 1 + x$.

Quadratic Taylor series approximation

In general:

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function taking as input $\boldsymbol{x} \in \mathbb{R}^n$ with second partial derivatives, then the quadratic Taylor approximation of $f(\boldsymbol{x})$ around $\boldsymbol{x}_0$ is

$$g(\boldsymbol{x}) = f(\boldsymbol{x}_0) + (\boldsymbol{x} - \boldsymbol{x}_0)^\intercal \boldsymbol{f}'(\boldsymbol{x}_0) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_0)^\intercal \boldsymbol{f}''(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0).$$

Newton's Method: definition

Also referred as *Newton-Raphson iteration*. The solution to $f(x) = 0$ can be obtained by performing an approximation, and finding the solution to that approximation (Gentle 2009).

Step 1: Approximate $f(x)$ by the first-order Taylor expansion.

$$f(x) \approx f(x^{(t)}) + f'(x^{(t)})(x - x^{(t)})$$

Step 2: Find the root of the approximation.

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

Newton's Method for optimization

The solution to $f'(x) = 0$ can be obtained by performing an approximation, and finding the solution to that approximation.

Step 1: Approximate $f'(x)$ by the first-order Taylor expansion.

$$f'(x) \approx f'(x^{(t)}) + f''(x^{(t)})(x - x^{(t)})$$

Step 2: Find the root of the approximation.

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

Using the Fisher information $I(\theta) = \mathbb{E}\left[-f''(\theta)\right]$ instead of $-f''(x^{(t)})$ leads to the Fisher scoring method.

Newton's Method for optimization

More generally, for $f : \mathbb{R}^n \to \mathbb{R}$:

Step 1: Approximate $f'(\boldsymbol{x})$ by the first-order Taylor expansion.

$$f'(\boldsymbol{x}) \approx f'(\boldsymbol{x}^{(t)}) + f''(\boldsymbol{x}^{(t)})(\boldsymbol{x} - \boldsymbol{x}^{(t)})$$

Step 2: Find the root of the approximation.

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - f''(\boldsymbol{x}^{(t)})^{-1} f'(\boldsymbol{x}^{(t)}).$$

Using the Fisher information $\boldsymbol{I}(\theta) = \mathbb{E}\left[-f''(\theta)\right]$ instead of $-f''(\boldsymbol{x}^{(t)})$ leads to the Fisher scoring method.

GLM: Iteratively reweighted least squares

Let consider the following model:

$$Y_i \mid \boldsymbol{x}_i \sim Bernoulli(\pi_i),$$
$$\text{logit}\,(\pi_i) = \boldsymbol{x}_i^\intercal \beta.$$

Where

$$\theta_i = \text{logit}\,(\pi_i) = \boldsymbol{x}_i^\intercal \beta,$$
$$b(\theta) = \log(1 - \pi_i) = \log(1 + \exp(\theta_i)) = \log(1 + \exp(\boldsymbol{x}_i^\intercal \beta)),$$
$$a(\phi) = 1, \text{ and } c(y_i, \phi) = 0.$$

Hence

Likelihood function

$$l(\beta) = \boldsymbol{y}^\intercal \boldsymbol{X} \beta - \boldsymbol{b}^\intercal \boldsymbol{1}$$

GLM: Iteratively reweighted least squares

Obtaining the gradient and Hessian:

$$l'(\beta) = \boldsymbol{X}^\top(\boldsymbol{y} - \pi)$$

$$l''(\beta) = \frac{d}{d\beta}(\boldsymbol{X}^\top(\boldsymbol{y} - \pi)) = -\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X}.$$

Where $\boldsymbol{W}$ is diagonal with $w_{ii} = \pi_i(1 - \pi_i)$. Hence, applying Newton's method:

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

where $\boldsymbol{z}^{(t)} = \boldsymbol{X}\beta^{(t)} + (\boldsymbol{W}^{(t)})^{-1}(\boldsymbol{y} - \pi^{(t)})$.

Steepest ascent

Newton-like method

Update such as

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

where $\boldsymbol{M}^{(t)}$ is a Hessian's approximation [@givens2012computational].

Steepest ascent

Replaces $\boldsymbol{M}^{(t)}$ by $-\boldsymbol{I}$.

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha^{(t)} g'(\boldsymbol{x}^t),$$

Gauss-Newton motivation: non-linear least squares

Consider the following model

$$Y_i = h(\boldsymbol{x}_i, \theta) + \varepsilon_i.$$

where $h(\boldsymbol{x}_i, \theta)$ is non-linear. This model has loss function:

$$L(\theta) = \left\| \boldsymbol{y} - \boldsymbol{h}(\boldsymbol{X}, \theta) \right\|^2.$$

Instead of approximating $L(\theta)$, we can approximate the function $h(x_i, \theta)$ using Taylor series.

Gauss-Newton method

By approximating $h(\boldsymbol{x}_i, \theta)$ we obtain

$$Y_i = h(\boldsymbol{x}_i, \theta^{(t)}) + (\theta - \theta^{(t)})^\top h'(\boldsymbol{x}_i, \theta^{(t)}) + \varepsilon_i.$$

$$Y_i = \tilde{h}(\boldsymbol{x}_i, \theta^{(t)}, \theta) + \varepsilon_i.$$

Let $z_i^{(t)} = y_i - h(\boldsymbol{x}_i, \theta^{(t)})$, and $\boldsymbol{a}_i^{(t)} = h'(\boldsymbol{x_i}, \theta^{(t)})$. Then

$$\boldsymbol{z}^{(t)} = \boldsymbol{A}^{(t)}(\theta - \theta^{(t)}) + \varepsilon.$$

The estimate can be obtained as follows:

$$\theta^{(t+1)} - \theta^{(t)} = \left(\boldsymbol{A}^{(t)^\top}\boldsymbol{A}^{(t)}\right)^{-1}(\boldsymbol{A}^{(t)})^\top \boldsymbol{z}^{(t)}.$$

$$\theta^{(t+1)} = \theta^{(t)} + \left(\boldsymbol{A}^{(t)^\top}\boldsymbol{A}^{(t)}\right)^{-1}(\boldsymbol{A}^{(t)})^\top \boldsymbol{z}^{(t)}.$$

It does not require to compute the Hessian matrix.

References

References

Gentle, James E. 2009. *Computational Statistics*. Springer Science & Business Media.

Givens, Geof H., and Jennifer A. Hoeting. 2012. *Computational Statistics*. John Wiley & Sons.

Wood, Simon N. 2017. *Generalized Additive Models: An Introduction with R, Second Edition*. CRC Press.