统计计算

王璐

1 Newton-Raphson Algorithm

对很多常用的统计模型, 比如 exponential family, 数据的 log-likelihood $l(\boldsymbol{\beta})$ 是参数 $\boldsymbol{\beta}$ 的 concave 函数, 这种情况下计算 $\boldsymbol{\beta}$ 的 MLE 只需令 score function 等于 $\mathbf{0}$:

$$\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \mathbf{0}. \tag{1}$$

Score function 一般是非线性的，(1)经常需要用到求解非线性方程组的算法 — Newton-Raphson algorithm.

对于一般的非线性方程组

$$\boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$$

其中 $\boldsymbol{h}: \mathbb{R}^p \to \mathbb{R}^p$，求解该方程组的 Newton-Raphson 算法如下：

Newton-Raphson Algorithm

1. 选取一个初始值 $\boldsymbol{x}^{(0)}$.

2. 在 $\boldsymbol{x}^{(0)}$ 的邻域内用线性函数近似 $\boldsymbol{h}(\cdot)$

$$\boldsymbol{h}(\boldsymbol{x}) \approx \boldsymbol{h}(\boldsymbol{x}^{(0)}) + \nabla \boldsymbol{h}(\boldsymbol{x}^{(0)})(\boldsymbol{x} - \boldsymbol{x}^{(0)}).$$

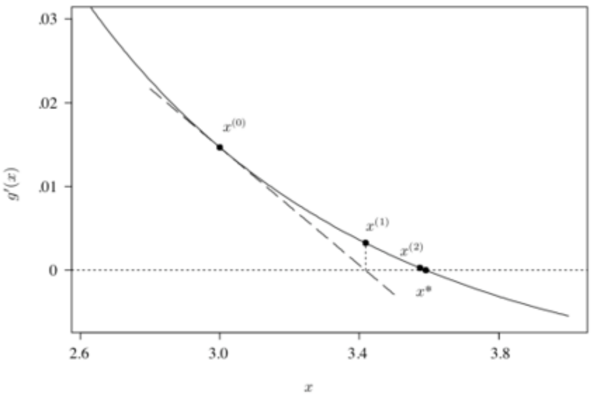
3. 令上式右边等于 $\mathbf{0}$, 得到 $\boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}$ 的一个近似解：

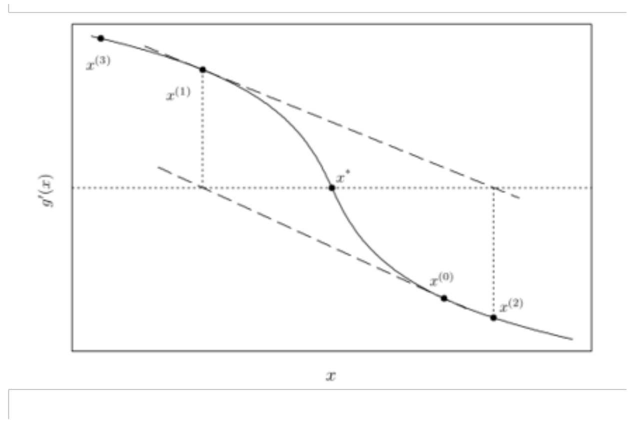
$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} - \left[\nabla \boldsymbol{h}(\boldsymbol{x}^{(0)})\right]^{-1} \boldsymbol{h}(\boldsymbol{x}^{(0)})$$

4. 重复此过程，第 $(t+1)$ 次迭代得到

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \left[\nabla \boldsymbol{h}(\boldsymbol{x}^{(t)})\right]^{-1} \boldsymbol{h}(\boldsymbol{x}^{(t)})$$

5. 如果 $\left\|\boldsymbol{h}(\boldsymbol{x}^{(t+1)})\right\| < \varepsilon$，算法终止，输出 $\boldsymbol{x}^{(t+1)}$.





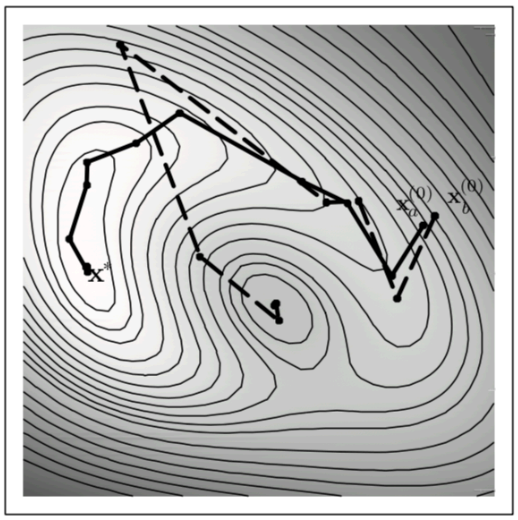


FIGURE 2.7 Application of Newton's method for maximizing a complicated bivariate function, as discussed in Example 2.4. The surface of the function is indicated by shading and contours, with light shading corresponding to high values. Two runs starting from $\mathbf{x}_a^{(0)}$ and $\mathbf{x}_b^{(0)}$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

Picture source: *Computational Statistics* (2nd edition) by Geof H.Givens (2012)

Remarks

1. Newton 算法是否收敛取决于 $\boldsymbol{h}(\cdot)$ 的形状和选取的初始值 $\boldsymbol{x}^{(0)}$.

2. Memory cost $O(p^2)$: 每一步 Newton 迭代需要储存一个 $p \times p$ 矩阵 $\nabla \boldsymbol{h}(\boldsymbol{x}^{(t)})$.

   Computation cost $O(p^3)$: 每一步 Newton 迭代需要计算一个 $p \times p$（稠密）矩阵的逆或解一组线性方程组。

2 收敛性分析

以一元函数为例，假设 $x^\star$ 是 $h(x)=0$ 的一个根，$h(x)$ 二阶连续可导且 $h'(x^\star)\neq 0$. 由于 h' 连续且 $h'(x^\star)\neq 0$，则存在 $x^\star$ 的一个邻域，使得这个邻域内的所有点 x 都满足 $h'(x)\neq 0$. 假设 $x^{(t)}$ 在这个邻域内，将 $h(x^\star)$ 在 $x^{(t)}$ 处 Taylor 展开

$$0=h(x^\star)=h(x^{(t)})+h'(x^{(t)})(x^\star-x^{(t)})+\frac{1}{2}h''(\tilde{x})(x^\star-x^{(t)})^2 \tag{2}$$

其中 $\tilde{x}$ 介于 $x^\star$ 和 $x^{(t)}$ 之间。整理(2)得

$$\underbrace{x^{(t)}-\frac{h(x^{(t)})}{h'(x^{(t)})}}_{x^{(t+1)}}-x^\star=(x^\star-x^{(t)})^2\frac{h''(\tilde{x})}{2h'(x^{(t)})}. \tag{3}$$

令 $\epsilon^{(t)}=x^{(t)}-x^\star$, 由(3)得

$$\epsilon_{t+1}=\epsilon_t^2\frac{h''(\tilde{x})}{2h'(x^{(t)})}. \tag{4}$$

考虑 $x^\star$ 的一个 δ-邻域，$\mathcal{N}_\delta(x^\star)=(x^\star-\delta,x^\star+\delta)$, 和下面这个跟 δ 有关的函数:

$$c(\delta)=\max_{x_1,x_2\in\mathcal{N}_\delta(x^\star)}\left|\frac{h''(x_1)}{2h'(x_2)}\right|.$$

由(4)得

$$\begin{aligned}|\epsilon_{t+1}|&\le\epsilon_t^2c(\delta)\\ |c(\delta)\epsilon_{t+1}|&\le(c(\delta)\epsilon_t)^2.\end{aligned} \tag{5}$$

如果 $|\epsilon_0|=\left|x^{(0)}-x^\star\right|<\delta$, 则由(5)得

$$|c(\delta)\epsilon_t|\le(c(\delta)\epsilon_{t-1})^2\le(c(\delta)\epsilon_{t-2})^{2^2}\le\cdots\le(c(\delta)\epsilon_0)^{2^t}=(\delta c(\delta))^{2^t}. \tag{6}$$

进一步研究函数 $c(\delta)$ 的性质。注意到当 $\delta\to 0$,

$$c(\delta)\to\left|\frac{h''(x^\star)}{2h'(x^\star)}\right|$$

即 $c(\delta)$ 收敛到一个有限值。因此 $\delta\to 0$, $\delta c(\delta)\to 0$. 总可以找到一个 δ_1 使得 $\delta_1 c(\delta_1)<1$. 假设初始值 $x^{(0)}$ 满足 $|\epsilon_0|=\left|x^{(0)}-x^\star\right|<\delta_1$, 根据(6)，当 $t\to\infty$,

$$|\epsilon_t|\le\frac{(\delta_1c(\delta_1))^{2^t}}{c(\delta_1)}\to 0$$

则 $x^{(t)}\to x^\star$.

以上我们证明了一个定理：如果函数 h 二阶连续可导，$x^\star$ 是 $h(x)=0$ 的一个根，那么存在 $x^\star$ 的一个邻域，在这个邻域内选取任意初始值 $x^{(0)}$, Newton 算法都会收敛。

事实上，当函数 h 二阶连续可导且是凸函数或凹函数，并且有一个根，则 Newton 算法对任意初始值都收敛。

2.1 收敛阶 (convergence order)

算法的收敛速度可以用算法的**收敛阶**衡量。

Definition 2.1 (收敛阶)**.** 如果算法第 t 步误差 ϵ_t 满足

$$\lim_{t\to\infty}\epsilon_t = 0 \text{ 且 } \lim_{t\to\infty}\frac{|\epsilon_{t+1}|}{|\epsilon_t|^{\alpha}} = c$$

其中常数 $c \neq 0$, $\alpha > 0$, 称算法的收敛阶为 α.

算法的收敛阶越高意味着算法可以越快逼近真实解。如果 Newton 算法收敛，由(4)得

$$\lim_{t\to\infty}\frac{|\epsilon_{t+1}|}{|\epsilon_t|^2} = \left|\frac{h''(x^\star)}{2h'(x^\star)}\right| = c.$$

因此 Newton 算法是二阶收敛 ($\alpha = 2$, quadratic convergence).

3 估计 Logistic 回归的 MLE

3.1 Logistic 回归模型

被解释变量 (response)

$$Y_i \mid \boldsymbol{x}_i \overset{ind}{\sim} \text{Bernoulli}(\pi_i),\ i = 1, \ldots, n.$$

解释变量 (covariates) $\boldsymbol{x}_i \in \mathbb{R}^p$ 与概率 $\pi_i = P(Y_i = 1 \mid \boldsymbol{x}_i)$ 有以下关系：

$$\text{logit}(\pi_i) \triangleq \log\left(\frac{\pi_i}{1-\pi_i}\right) = \boldsymbol{x}_i^\top \boldsymbol{\beta}. \tag{7}$$

Remarks

1. 模型 (7)为什么合理？与线性回归类似，我们希望将 $E(Y_i \mid \boldsymbol{x}_i) = P(Y_i = 1 \mid \boldsymbol{x}_i) = \pi_i$ 与被解释变量 $\boldsymbol{x}_i$ 的线性组合建立关系。但是 $\boldsymbol{x}_i^\top\boldsymbol{\beta}$ 可以是任何实数，因此需要将 $\boldsymbol{x}_i^\top\boldsymbol{\beta}$ 映射到 $(0, 1)$ 区间。
   - odds ratio $\dfrac{\pi_i}{1-\pi_i}$ 将概率 π_i 转化为一个正实数，再取 log 将它转化为任意实数。
2. 将 π_i 直接表示为 $\boldsymbol{x}_i^\top\boldsymbol{\beta}$ 的函数：

$$\begin{aligned}\frac{\pi_i}{1-\pi_i} &= e^{\boldsymbol{x}_i^\top\boldsymbol{\beta}}\\ \pi_i &= \frac{1}{1+e^{-\boldsymbol{x}_i^\top\boldsymbol{\beta}}}.\end{aligned}$$

3. logistic function

$$\pi(x) = \frac{1}{1+e^{-x}}$$

- $\pi : \mathbb{R} \to (0,1)$
- $\pi'(x) = \pi(x)\,(1-\pi(x))$

3.2 MLE of $\boldsymbol{\beta}$

在模型 (7) 下, 数据的似然函数 (likelihood) 为:

$$\begin{aligned}
L(\boldsymbol{\beta}) &= P(Y_1 = y_1, \ldots, Y_n = y_n \mid \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{\beta}) \\
&= \prod_{i=1}^{n} P(Y_i = y_i \mid \boldsymbol{x}_i, \boldsymbol{\beta}) \\
&= \prod_{i=1}^{n} \pi_i^{y_i} (1-\pi_i)^{1-y_i}
\end{aligned}$$

根据定义, $\boldsymbol{\beta}$ 的 MLE (maximum likelihood estimation) 为

$$\hat{\boldsymbol{\beta}} = \text{argmax } L(\boldsymbol{\beta}) = \text{argmax } \log L(\boldsymbol{\beta})$$

为了计算方便, 一般选择最大化 log-likelihood. 它是对 likelihood 的单调变化、不影响 argmax.

$$\begin{aligned}
\log L(\boldsymbol{\beta}) &= \sum_{i=1}^{n} y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i) \\
&= \sum_{i=1}^{n} y_i \log\left(\frac{\pi_i}{1-\pi_i}\right) + \log(1-\pi_i) \\
&= \sum_{i=1}^{n} y_i \boldsymbol{x}_i^T \boldsymbol{\beta} + \log(1-\pi_i)
\end{aligned}$$

- Score function (1st derivative of log-likelihood):

$$\begin{aligned}
\frac{\partial \log L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= \sum_{i=1}^{n} y_i \boldsymbol{x}_i - \frac{1}{1-\pi_i}\frac{\partial \pi_i}{\partial \boldsymbol{\beta}} \\
&= \sum_{i=1}^{n} y_i \boldsymbol{x}_i - \frac{1}{1-\pi_i}\pi_i(1-\pi_i)\boldsymbol{x}_i \\
&= \sum_{i=1}^{n} (y_i - \pi_i)\boldsymbol{x}_i
\end{aligned}$$

- 2nd derivative (negative of the observed Fisher's information)

$$\frac{\partial^2 \log L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} = \sum_{i=1}^{n} -\underbrace{\pi_i(1-\pi_i)}_{\geq 0}\underbrace{\boldsymbol{x}_i\boldsymbol{x}_i^\top}_{p.s.d}$$

注意到 $\dfrac{\partial^2 \log L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top}$ 与 $y_{1:n}$ 无关, 且是半负定矩阵, 因此 $\log L(\boldsymbol{\beta})$ 是 $\boldsymbol{\beta}$ 的凹函数，则 $\boldsymbol{\beta}_{MLE}$ 是以下方程的根：

$$\frac{\partial \log L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \mathbf{0}. \tag{8}$$

使用 Section 1的 Newton-Raphson algorithm 求解。令

$$\boldsymbol{h}(\boldsymbol{\beta}) = \frac{\partial \log L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = X^\top \boldsymbol{z}.$$

其中 $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$, $\boldsymbol{z} = (z_1, \ldots, z_n)^\top$, $z_i = y_i - \pi_i$, $i = 1, \ldots, n$.

迭代中还需计算:

$$\begin{aligned} \nabla \boldsymbol{h}(\boldsymbol{\beta}) &= \frac{\partial^2 \log L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} = -X^\top W X \\ [\nabla \boldsymbol{h}(\boldsymbol{\beta})]^{-1} &= -\left(X^\top W X\right)^{-1} \end{aligned}$$

其中 $W = \operatorname{diag}(w_1, \ldots, w_n)$, $w_i = \pi_i(1-\pi_i)$, $i = 1, \ldots, n$. 此时 Newton 迭代的公式为

$$\begin{aligned} \boldsymbol{\beta}^{(t+1)} &= \boldsymbol{\beta}^{(t)} - \left[\nabla \boldsymbol{h}(\boldsymbol{\beta}^{(t)})\right]^{-1} \boldsymbol{h}(\boldsymbol{\beta}^{(t)}) \\ &= \boldsymbol{\beta}^{(t)} + \left(X^\top W^{(t)} X\right)^{-1} X^\top \boldsymbol{z}^{(t)} \end{aligned} \tag{9}$$

其中 $W^{(t)}$, $\boldsymbol{z}^{(t)}$ 为 W, $\boldsymbol{z}$ 在第 t 步的取值，因为 π_i 在第 t 步的估计值依赖 $\boldsymbol{\beta}^{(t)}$.

Remarks

1. 以上使用 Newton-Raphson 求解 MLE 的过程需要用到 score function 和 observed Fisher's info, 因此这种算法在统计中也被称为 Fisher-scoring method.

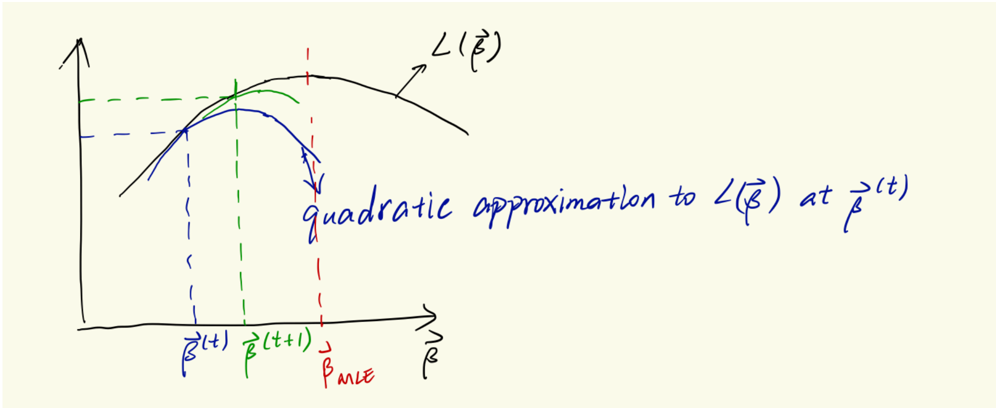
2. 迭代(9)也常写为以下形式：

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \left(X^\top W^{(t)} X\right)^{-1} X^\top W^{(t)} \tilde{\boldsymbol{z}}^{(t)} \tag{10}$$

   其中 $\tilde{\boldsymbol{z}}^{(t)} = \left[W^{(t)}\right]^{-1} \boldsymbol{z}^{(t)}$. 此时等号右边第二项跟 weighted least square (WLS) 估计量的形式相同。这样从初始值 $\boldsymbol{\beta}^{(0)}$ 开始，每一步迭代只需先计算出 $W^{(t)}$, $\boldsymbol{z}^{(t)}$ 和 $\tilde{\boldsymbol{z}}^{(t)}$, 再做一个 WLS 就可以得到 $\boldsymbol{\beta}^{(t+1)}$. 因此迭代(10)也被称为 iteratively reweighted least square (IRLS).

3. 如果迭代(9)收敛，就可以得到 $\boldsymbol{\beta}_{MLE}$. 对于 Logistic 回归, Newton-Raphson 通常在任意初始值下都收敛得很快。

4. 从几何角度考虑，使用 Newton-Raphson 求解方程(8)时，每一步迭代使用一阶 Taylor expansion（线性函数）近似 score function, 这其实相当于用一个二次函数近似 log-likelihood. 更新的 $\boldsymbol{\beta}$ 是这个二次函数的 argmax, 对应更大的 log-likelihood, 最终会收敛到 MLE.



References

Geof H.Givens, J. A. H. (2012). *Computational Statistics*. John Wiley & Sons, Inc, second edition.