Unconstrained Problem and Algorithm II

Jong-June Jeon

October 11, 2023

Department of Statistics, University of Seoul

CONTENTS

- Secant method
- MM algorithm
- Applications

SECANT METHOD

Newton method vs. Secant Method

The common goal is to find the solution of $f'(x) = 0$

- (Newton method) For an existing approximator $x^{(t)}$, the next solution is updated by a linear equation, $f'(x^{(t)}) + f''(x^{(t)})(x - x^{(t)}) = 0$.

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

- (Secant method) For two existing approximators $x^{(t)}$ and $x^{(t+1)}$, the next solution is updated by a linear equation, $f'(x^{(t)}) + \left(\frac{f'(x^{(t+1)}) - f'(x^{(t)})}{x^{(t+1)} - x^{(t)}}\right)(x - x^{(t)}) = 0$.

$$x^{(t+2)} = x^{(t)} - \left(\frac{f'(x^{(t+1)}) - f'(x^{(t)})}{x^{(t+1)} - x^{(t)}}\right)^{-1} f'(x^{(t)}).$$

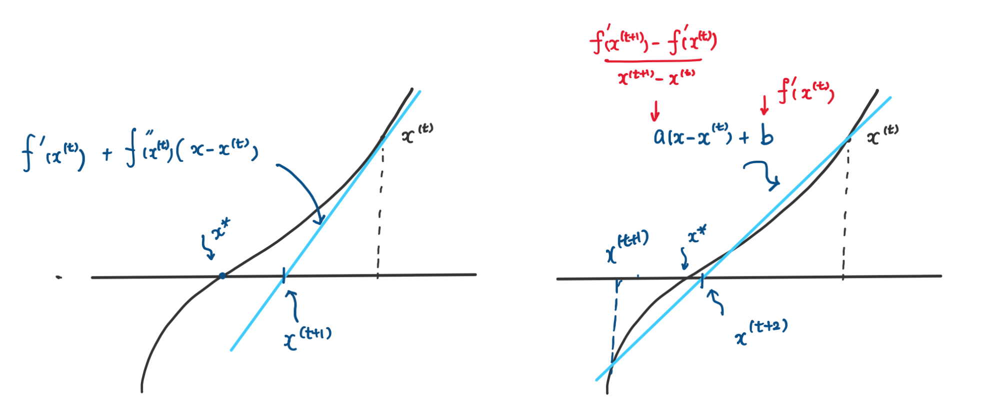
$f'(x^{(t)}) + f''(x^{(t)})(x - x^{(t)})$
$x^{(t)}$
x^*
$x^{(t+1)}$
$\frac{f'(x^{(t+1)}) - f'(x^{(t)})}{x^{(t+1)} - x^{(t)}}$
$f'(x^{(t)})$
$a(x - x^{(t)}) + b$
$x^{(t)}$
$x^{(t+1)}$
x^*
$x^{(t+2)}$

Figure 1: Left: Newton method. Right: Secant method

Secant condition

The next solution is equally given by the two different equations:

$$
\begin{aligned}
f'(x^{(t)}) + B(x - x^{(t)}) &= 0 \quad (1)\\
f'(x^{(t+1)}) + B(x - x^{(t+1)}) &= 0, \quad (2)
\end{aligned}
$$

where $B = \left(\frac{f'(x^{(t+1)}) - f'(x^{(t)})}{x^{(t+1)} - x^{(t)}}\right)$.

That is, $x^{(t+2)} = x^{(t)} - B^{-1}f'(x^{(t)}) = x^{(t+1)} - B^{-1}f'(x^{(t+1)})$ and $x^{(t+1)} - x^{(t)} = B^{-1}(f'(x^{(t+1)}) - f'(x^{(t)}))$. Thus,

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

Definition 1 (Secant condition)

An approximation of the hessian matrix B satisfies that

$$B(x^{(t+1)} - x^{(t)}) = \nabla f(x^{(t+1)}) - \nabla f(x^{(t)})$$

Approximation of Hessian matrix

Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be convex and twice differentiable. The Hessian matrix satisfies

$$\nabla f(x) - \nabla f(x+s) \simeq -\nabla^2 f(x+s)s$$

Let $x^{(k+1)} = x^{(k)} + s^{(k)}$. When the computation of $\nabla^2 f(x^{(k+1)})$ is practically difficult, an approximation of $\nabla^2 f(x^{(k+1)})$, B_{k+1}, is desired to satisfy

$$\nabla f(x^{(k+1)}) \simeq \nabla f(x^{(k)}) + B_{k+1}(x^{(k+1)} - x^{(k)}).$$

The secant condition for B_{k+1} is written by

$$\nabla f(x^{(k+1)}) - \nabla f(x^{(k)}) = B_{k+1}(x^{(k+1)} - x^{(k)}).$$

Outline of Quasi-Newton Method

1. For $k = 0$, $x^{(k)}$ and B_k are initialized.
2. $\nabla f(x^{(k)})$ is computed and $x^{(k+1)}$ follows by

$$x^{(k+1)} = x^{(k)} - (B_k)^{-1} \nabla f(x^{(k)})$$

3. $\nabla f(x^{(k+1)})$ is computed.
4. B_{k+1} is updated.

Before the step 4, new information is $s_k = x^{(k+1)} - x^{(k)}$ and $y_k = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$. Note that we let $B_{k+1} s_k = y_k$ by the secant condition.

Update of B_{k+1}

- Rank 1 update

$$B_{k+1} = B_k + a v_k v_k^\top$$

- Rank 2 update

$$B_{k+1} = B_k + \alpha u_k u_k^\top + \beta v_k v_k^\top$$

(Symmetric Rank-1 update)

Since $B_{k+1}s_k = B_k s_k + a v_k v_k^\top s_k$ or $y_k - B_k s_k = a(v_k^\top s_k) v_k$, $v_k \propto y_k - B_k s_k$. Let $v_k = d(y_k - B_k s_k)$ then

$$y_k - B_k s_k = ad^2((y_k - B_k s_k)^\top s_k)(y_k - B_k s_k)$$

Let $d^2 = 1/((y_k - B_k s_k)^\top s_k)$ and $a = \mathsf{sign}((y_k - B_k s_k)^\top s_k)$. Thus,

$$v_k = \frac{1}{\sqrt{(y_k - B_k s_k)^\top s_k}}(y_k - B_k s_k)$$

and the update rule is given by

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^\top}{(y_k - B_k s_k)^\top s_k}$$

(Rank-2 update: BFGS)

Since $B_{k+1}s_k = B_k s_k + \alpha(u_k^\top s_k)u_k + \beta(v_k^\top s_k)v_k$,

$$y_k - B_k s_k = \alpha(u_k^\top s_k)u_k + \beta(v_k^\top s_k)v_k$$

Let $u_k = y_k$ and $v_k = B_k s_k$ then $\alpha(u_k^\top s_k) = 1$ and $\beta(v_k^\top s_k) = -1$. Since $\alpha = 1/(y_k^\top s_k)$ and $\beta = -1/(s_k^\top B_k s_k)$,

$$B_{k+1} = B_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{B_k s_k s_k^\top B_k^\top}{s_k^\top B_k s_k}.$$

Broyden-Fletcher-Goldfarb-Shanno algorithm

(1) Set $k = 0$, let an initial $x^{(k)}$ and an initial Hessian matrix B_k.

(2) Find the direction $s_k = -B_k^{-1}\nabla f(x^{(k)})$.

(4) Update $x^{(k+1)} = x^{(k)} + s_k$.

(5) Update $B_{k+1} = B_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{B_k s_k s_k^\top B_k^\top}{s_k^\top B_k s_k}$, where $y_k = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$.

(6) Repeat (2)-(5) until the solution converges.

Broyden-Fletcher-Goldfarb-Shanno update rule

(1) Set $k = 0$, let an initial $x^{(k)}$ and an initial Hessian matrix B_k.

(2) Find the direction $p_k = -B_k^{-1} \nabla l(x^{(k)})$.

(3) (Line search) Find a step size $\alpha_k = \text{argmin}_\alpha l(x^{(k)} + \alpha p_k)$.

(4) Update $x^{(k+1)} = x^{(k)} + \alpha_k p_k$.

(5) Update $B_{k+1} = B_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{B_k s_k s_k^\top B_k^\top}{s_k^\top B_k s_k}$,
where $s_k = \alpha_k p_k$ and $y_k = \nabla l(x^{(k+1)}) - \nabla l(x^{(k)})$.

(6) Repeat (2)-(5) until the solution converges.

Symmetric Rank-1 algorithm

Let $H_k = B_k^{-1}$ For step (5), by Sherman-Morrison formula,

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^\top}{(s_k - H_k y_k)^\top y_k} \tag{3}$$

- When $(s_k - H_k y_k)^\top y_k < 0$, the nonnegative definiteness of H_{k+1} can be violated.
- When $(s_k - H_k y_k)^\top y_k$ is close to zero, the updating process becomes unstable.

BFGS algorithm

For step 5, by Sherman-Morrison formula,

$$H_{k+1} = \left(I - \frac{s_k y_k^\top}{y_k^\top s_k}\right) H_k \left(I - \frac{y_k s_k^\top}{y_k^\top s_k}\right) + \frac{s_k s_k^\top}{y_k^\top s_k}. \tag{4}$$

When H_k is positive definite and $y_k^\top s_k > 0$, then H_{k+1} is also positive definite. The line search is required for $y_k^\top s_k > 0$ (see the strong Wolfe Condition).

Total derivatives of f

Let $f : \mathbb{R}^n \mapsto \mathbb{R}^k$ If there exists a linear map $g : \mathbb{R}^n \mapsto \mathbb{R}^k$ such that

$$\|f(x_0 + h) - f(x_0) - gh\| \to 0$$

as $\|h\| \to 0$

we call g the total derivative of f at x_0. Note that g is a linear map depending on x_0 (in fact $n \times k$ matrix). Let $d(h; x_0) = f(x_0 + h) - f(x_0)$ then g is an approximation of the map d. Now replace f by ∇f. What is g?

L-BFGS algorithm

Limited-memory BFGS

```
(1) Set $k = 0$, and let an initial $x^{(k)}$.
(2) Let an $q = \nabla l(x^{(k)})$.
(3) For $i = k-1, ..., k-m$ :
      • $\alpha_i = \frac{s_i^\top q}{y_i^\top s_i}$.
      • $q = q - \alpha_i y_i$.
(4) $\gamma_k = \frac{s_{k-1}^\top y_{k-1}}{y_{k-1}^\top y_{k-1}}$, $H_k^0 = \gamma_k I$ and $z = -H_k^0 q$.
(5) For $i = k-m, ..., k-1$ :
      • $\beta_i = \frac{y_i^\top z}{y_i^\top s_i}$.
      • $z = z - s_i(\alpha_i - \beta_i)$
(6) Repeat (2)-(5) until the solution converges.
```

MM ALGORITHM

Definition 2 (Majorized function)

Let $f, g : \mathbb{R}^p \to \mathbb{R}$. If $g(x|x^{(t)}) \geq f(x)$ for all x and $g(x^{(t)}|x^{(t)}) = f(x^{(t)})$ then we call $g(x|x^{(t)})$ is a majorized function of f at $x^{(t)}$.

Example 3 (Quantile loss function)

The loss function $l(\cdot;q):\mathbb{R}\to\mathbb{R}$ is given by

$$l(x;q)=\begin{cases} qx & ,x\geq 0 \\ -(1-q)x & ,x<0 \end{cases}$$

for $0<q<1$. Then a majorized function of f at $x^*\neq 0$ is given by

$$g(x|x^*)=\frac{1}{4|x^*|}x^2+(q-\frac{1}{2})x+\frac{|x^*|}{4}$$

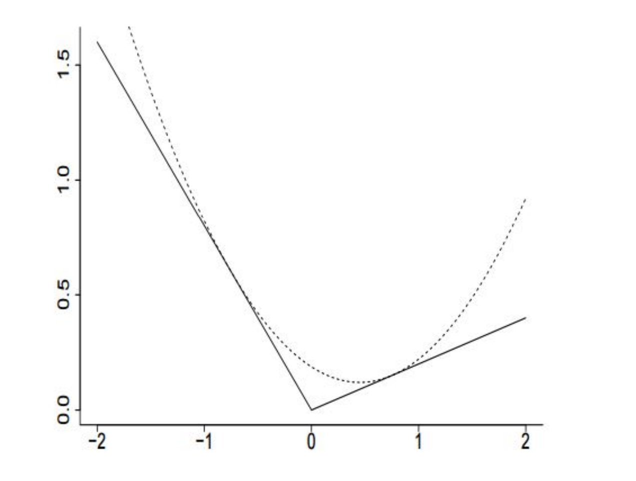

Figure 2: A majorized function of quantile loss

proof) Let $g(x|x^*) = ax^2 + bx + c$. There are four sufficient conditions for $g(x|x^*)$ to be a majorized function of f at $x^* \neq 0$:

- $(b-q)^2 - 4ac = 0$ and $(b-q+1)^2 - 4ac = 0 : b = q - 1/2$
- $2ax^* + b = q : a = 1/4x^*$
- $a(x^*)^2 + bx^* + c = qx^* : c = x^*/4$

Example 4 (Quantile regression)

Let the risk function

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{n} l(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}; q)$$

for $0 < q < 1$. Then a majorized function of $L(\boldsymbol{\beta}; q)$ at $\boldsymbol{\beta}^*$ is given by

$$g(\boldsymbol{\beta}|\boldsymbol{\beta}^*) = \sum_{i=1}^{n} \frac{1}{4|r_i^*|}(y_i - \mathbf{x}_i^\top \boldsymbol{\beta})^2 + (q - \frac{1}{2})(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}) + \frac{|r_i^*|}{4}$$

where $r_i^* = y_i - \mathbf{x}_i^\top \boldsymbol{\beta}^*$. Note that $g(\boldsymbol{\beta}|\boldsymbol{\beta}^*)$ is quadratic function.

MM algorithm (Majorize-Minimization or Minorize-Maximization)

$$\text{minimize}_{\,x} \quad f(x)$$

- Give a initial solution $x^{(0)}$ and let $t = 0$.
- Obtain the majorized function at $x^{(t)}$: $q(x|x^{(t)})$
- Minimize the majorized function and let the minimizer $x^{(t+1)}$.
- $t \leftarrow t + 1$ and repeat steps 2-4 until the solution converges.

Proposition 1 (Descent property of MM algorithm)

If a sequence of $x^{(t)}$ for $t = 1, 2, \cdots$ is obtained by MM algorithm, then

$$f(x^{(t+1)}) \leq f(x^{(t)})$$

for all t. It means that the value of the object function evaluated at the solution always non-decreasing. Moreover, if f is strictly convex and $x^{(t+1)} \neq x^{(t)}$, then the values are always decreasing.

proof)

$$\begin{aligned} f(x^{(t)}) &= q(x^{(t)}|x^{(t)}) \\ &\geq q(x^{(t+1)}|x^{(t)}) \\ &\geq f(x^{(t+1)}) \end{aligned}$$

- By definition of the majorized function the first equality holds.
- Since $x^{(t+1)}$ is the minimizer of $q(x|x^{(t)})$, the first inequality holds.
- By definition of the majorized function the third inequality holds.

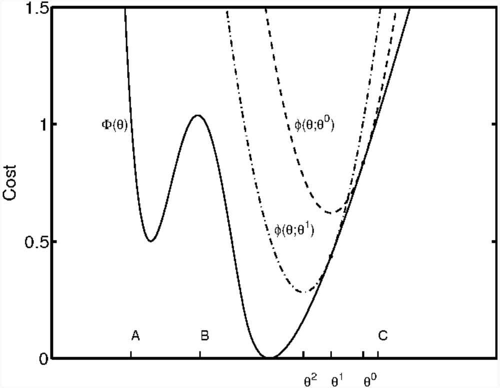

Figure 3: illustration of MM algorithm

(Q) What is the key to the success of MM algorithm?

APPLICATIONS

- Huberized regression
- Logistic regression (for stable computation)
- Bradley-Terry model

Huberized regression

Let a loss function $l(z; d) = \frac{1}{2}z^2\mathbb{I}(|z| \le d) + (d|z| - d^2/2)\mathbb{I}(|z| > d)$ where $\mathbb{I}(\cdot)$ is the indicator function. When $d = \infty$, the loss function is L_2 loss function.

The regression estimator of regression model with huberized loss function is defined by

$$\hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\beta} \frac{1}{n}\sum_{i=1}^{n} l(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}; d)$$

$\hat{\boldsymbol{\beta}}$ is robust to the error distribution of the assumed linear model. When the error variance is infinite, the estimator enjoys good asymptotic properties.

Here we will show the algorithm to obtain the Huberized regression estimator. We can decompose the function as

$$l(z; d) = l^{(1)}(z; d) + l^{(2)}(z; d)$$

where

$$l^{(1)}(z; d) = z^2/2$$

and

$$l^{(2)}(z; d) = (|z| + d^2/2 - z^2/2 - d)I(|z| > d).$$

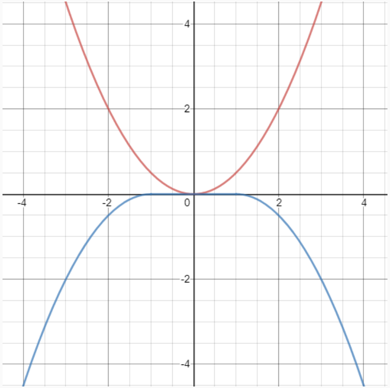

Figure 4: decomposition of Huber loss

Then, $l^{(2)}(z;d)$ is a differentiable concave function and

$$l^{(1)}(z;d) + \nabla l^{(2)}(z^*;d)(z - z^*) + l^{(2)}(z^*;d)$$

is a majorized function of $l(z;d)$ at z^* (by concavity of $l^{(2)}$). That is,

$$l(z;d) \leq l^{(1)}(z;d) + \nabla l^{(2)}(z^*;d)(z - z^*) + l^{(2)}(z^*;d)$$

Using this inequality, we construct MM algorithm for regression model with huber loss.

cf) CCCP algorithm

MM algorithm for the huberized regression

1. Let $k = 0$ and set an initial estimator $\boldsymbol{\beta}^{(k)}$
2. Repeat:
   - Obtain the majorized function of $\frac{1}{n}\sum_{i=1}^{n} l(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}; d)$ at $\boldsymbol{\beta}^{(k)}$:

$$\begin{aligned} Q(\boldsymbol{\beta}|\boldsymbol{\beta}^{(k)}) &= \sum_{i=1}^{n} l^{(1)}\Big((y_i - \mathbf{x}_i^\top \boldsymbol{\beta}; d) \\ &\quad +\nabla l^{(2)}(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}^{(k)}; d)(\mathbf{x}_i^\top \boldsymbol{\beta} - \mathbf{x}_i^\top \boldsymbol{\beta}^{(k)}) \\ &\quad +l^{(2)}(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}^{(k)}; d)\Big), \end{aligned}$$

     which is a quadratic function.
   - Minimize $Q(\boldsymbol{\beta}|\boldsymbol{\beta}^{(k)})$ and update $\boldsymbol{\beta}^{(k)}$
   - $k \to k+1$

Logistic regression

- $y \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^p$ and $y|\mathbf{x} \sim$ Bernoulli$(\theta(\mathbf{x};\boldsymbol{\beta}))$, where

$$\theta(\mathbf{x};\boldsymbol{\beta}) = \frac{\exp(\mathbf{x}^\top\boldsymbol{\beta})}{1+\exp(\mathbf{x}^\top\boldsymbol{\beta})}.$$

- Let $(y_i, \mathbf{x}_i)$ for $i = 1, \cdots, n$ be independent random samples, then the negative loglikelihood function is given by

$$l(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}[-y_i\mathbf{x}_i^\top\boldsymbol{\beta} + \log(1+\exp(\mathbf{x}_i^\top\boldsymbol{\beta}))]$$

Assume that $\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top / n \in \mathbb{R}^{p \times p}$ is positive definite. For a fixed $\boldsymbol{\beta}$ there exists $\tilde{\boldsymbol{\beta}} \in B = \{b \in \mathbb{R}^p : b = h\boldsymbol{\beta} + (1-h)\hat{\boldsymbol{\beta}}^{(k)}, 0 \le h \le 1\}$ such that

$$l(\boldsymbol{\beta}) = l(\hat{\boldsymbol{\beta}}^{(k)}) + \nabla l(\hat{\boldsymbol{\beta}}^{(k)})^\top (\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}) + \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)})^\top \nabla^2 l(\tilde{\boldsymbol{\beta}})(\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}).$$

Let $A = \frac{1}{4}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top$ then it can be shown that $A - \nabla^2 l(\tilde{\boldsymbol{\beta}})$ is nonnegative definite, that is

$$(\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)})^\top A (\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}) \geq (\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)})^\top \nabla^2 l(\tilde{\boldsymbol{\beta}})(\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}). \tag{5}$$

Moreover, we know that $(\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)})^\top A (\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}) \geq \sup_{\tilde{\beta}} (\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)})^\top \nabla^2 l(\tilde{\boldsymbol{\beta}})(\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)})$.

Hence, from (5),

$$l(\boldsymbol{\beta}) \leq l(\hat{\boldsymbol{\beta}}^{(k)}) + \nabla l(\hat{\boldsymbol{\beta}}^{(k)})^\top (\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}) + \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)})^\top A (\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}),$$

which is a majorizing function of $l(\boldsymbol{\beta})$ at $\boldsymbol{\beta}^{(k)}$.

Consider a trick to solve logistic regression by L_2 regression package for stable computation. Let $X = [\mathbf{x}_1 : \mathbf{x}_2 : \cdots : \mathbf{x}_n]'$, and $Y = (y_1, \cdots, y_n)'$, and $\theta(\mathbf{x}_i; \boldsymbol{\beta}) = \frac{\exp(\mathbf{x}_i'\boldsymbol{\beta})}{(1+\exp(\mathbf{x}_i'\boldsymbol{\beta}))}$. For convenience denote $\theta(\mathbf{x}_i; \hat{\boldsymbol{\beta}}^{(k)})$ by $\hat{\theta}_i$ and let $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \cdots, \hat{\theta}_n)'$. Note that

- $X'X = \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i'$
- $X'(\hat{\boldsymbol{\theta}} - Y) = \nabla l(\hat{\boldsymbol{\beta}}^{(k)})$.

Write the majorized function of $l(\boldsymbol{\beta})$ at $\boldsymbol{\beta}^{(k)}$ by

$$\begin{aligned}
Q(\boldsymbol{\beta}|\boldsymbol{\beta}^{(k)}) &= l(\hat{\boldsymbol{\beta}}^{(k)}) + \nabla l(\hat{\boldsymbol{\beta}}^{(k)})'(\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}) \\
&\quad + \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)})' A (\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}) \\
&= l(\hat{\boldsymbol{\beta}}^{(k)}) - (Y - \hat{\boldsymbol{\theta}})' X (\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}) \\
&\quad + \frac{1}{8}(\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)})' X' X (\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}) \\
&= \frac{1}{2}\|2(Y - \hat{\boldsymbol{\theta}}) + X\boldsymbol{\beta}^{(k)}/2 - X\boldsymbol{\beta}/2\|^2 + \text{const}
\end{aligned}$$

Let $\tilde{Y} = 2(Y - \hat{\boldsymbol{\theta}}) + X\boldsymbol{\beta}^{(k)}/2$ and $\tilde{X} = X/2$, then the $Q(\boldsymbol{\beta}|\hat{\boldsymbol{\beta}}^{(k)})$ can be regarded as the empirical risk function of the regression models

$$\tilde{Y} = \tilde{X}\boldsymbol{\beta} + \epsilon.$$

Thus, we can use the l_2 regression package to minimize $Q(\boldsymbol{\beta}|\hat{\boldsymbol{\beta}}^{(k)})$. The algorithm for logistic regression models by l_2 regression package follows (you can see the original idea in Section 3 of [Friedman et al., 2010]).

Logistic regression with l_2 regression package

1. Set $k = 0$ and an initial $\hat{\boldsymbol{\beta}}^{(k)}$ and let $\tilde{X} = X/2$
2. Compute $\tilde{Y} = 2(Y - \hat{\boldsymbol{\theta}}) + X\boldsymbol{\beta}^{(k)}/2$
3. Solve $\tilde{Y} \sim \tilde{X}$ by l_2 regression package, and obtain the solution $\hat{\boldsymbol{\beta}}$
4. Update $\hat{\boldsymbol{\beta}}^{(k+1)}$ by $\hat{\boldsymbol{\beta}}$
5. $k \to k + 1$ and repeat (2-4) until the solution converges.

Bradley-Terry model

Ranking data commonly arise from situations where it is desired to rank a set of individuals or objects in accordance with some criterion.

Two types of ranking data

- Ranking comes from a set of assigned scores.
  ex) University ranking
- Ranking directly observed.
  ex) Horse racing game

There are three components consisting in ranking data.

- **Comparison**(game): a unit acting to assign orders for some criterion.
- **Item**(player): object to be assigned orders in a game.
- **Ranking**: resulting orders from a game.

Notation

- Let S be a set of items in a comparison.
- Let R be rank-vector obtained from the comparison.
- Denote the events that an item j_1 is ranked higher than j_2 by $(j_1 \to j_2)$.

The Bradley-Terry Model [Bradley and Terry, 1952] is one of the most popular parametric probability models for ranking (see [Hunter, 2004]).

- When p items are to be ranked, the model assumes positive valued p parameters $(u_1, \cdots, u_p)$ representing utilities of items.
- Higher u_i, higher the probability of the item i being top ranked.

For identifiability, let $u_p = 1$.

Consider an event

$$(j_1 \to j_2)$$

and let r be the rank-vector corresponding to the event.

Then,

$$\Pr(R = r) = \frac{u_{j_1}}{u_{j_1} + u_{j_2}}$$

Likelihood

Let $D_i = \{(j,k) : j,k \in S_i, j \neq k\}$ and $y_{ijk} = I(R_{ij} < R_{ik})$.

$$L(u) = \prod_{i=1}^{n} \prod_{(j,k)\in D_i} \left(\frac{u_j}{u_j + u_k}\right)^{y_{ijk}} \left(\frac{u_k}{u_j + u_k}\right)^{1-y_{ijk}}$$

subject to $u_p = 1$ and $u_j > 0$.

Let w_{jk} be the number of winnings of item j against k. Then the loglikelihood is simply written by

$$l(\mathbf{u}) = \log L(\mathbf{u}) = \sum_{j=1}^{p} \sum_{k=1}^{p} [w_{jk} \log(u_j) - w_{jk} \log(u_j + u_k)]$$

MLE is given by

$$\hat{\mathbf{u}} = \text{argmin}_{\mathbf{u}>0, u_p=1} - l(\mathbf{u})$$

$$
\begin{aligned}
-l(\mathbf{u}) &= -\sum_{j=1}^{p}\sum_{k=1}^{p} w_{jk}\left[\log(u_j) - \log(u_j+u_k)\right] \\
&\le -\sum_{j=1}^{p}\sum_{k=1}^{p} w_{jk}\left[\log(u_j) - \frac{u_j+u_k}{\tilde{u}_j+\tilde{u}_k} - \log(\tilde{u}_j+\tilde{u}_k) + 1\right]
\end{aligned}
$$

by $\log x \le \log(\tilde{x}) + \frac{1}{\tilde{x}}(x-\tilde{x}) = \log\tilde{x} + x/\tilde{x} - 1.$

Thus,

$$Q(\mathbf{u}|\tilde{\mathbf{u}}) = -\sum_{j=1}^{p}\sum_{k=1}^{p} w_{jk}\left[\log(u_j) - \frac{u_j + u_k}{\tilde{u}_j + \tilde{u}_k} - \log(\tilde{u}_j + \tilde{u}_k) + 1\right]$$

is the majorized function of $-l(\mathbf{u})$ at $\tilde{\mathbf{u}}$, where $\tilde{\mathbf{u}} = (\tilde{u}_1, \cdots, \tilde{u}_p)$.

- Note that $Q(\mathbf{u}|\tilde{\mathbf{u}}) \geq -l(\mathbf{u})$, and the equality holds only when $\mathbf{u} = \tilde{\mathbf{u}}$
- Next, define the m-th coordinate function of $Q(\mathbf{u}|\tilde{\mathbf{u}})$ by

$$Q_m(\mathbf{u}|\tilde{\mathbf{u}}) = Q((\tilde{u}_1, \cdots, \tilde{u}_{m-1}, u, \tilde{u}_{m+1}, \cdots, \tilde{u}_p)|\tilde{\mathbf{u}}).$$

$Q_m(\mathbf{u}|\tilde{\mathbf{u}}) \geq -l(\mathbf{u})$ for all $u > 0$ and the equality holds when $u = \tilde{u}_m$.

We use the majorized function of $-l(\mathbf{u})$ at $\tilde{u}$ by $Q_m(\mathbf{u}|\tilde{u})$.

In finding maximizer $Q(\mathbf{u}|\tilde{\mathbf{u}})$, we just consider a function of u_m that

$$g(u_m) = \sum_{k=1}^{p} w_{mk} \left[\log(u_m) - \frac{u_m}{\tilde{u}_m + \tilde{u}_k} \right] + \sum_{k=1}^{p} w_{km} \left[- \frac{u_m}{\tilde{u}_m + \tilde{u}_k} \right]$$

- Note that the differential function of $g(u_m)$ is given by

$$g'(u_m) = \sum_{k=1}^{p} w_{mk} \left[\frac{1}{u_m} - \frac{1}{\tilde{u}_m + \tilde{u}_k} \right] + \sum_{k=1}^{p} w_{km} \left[- \frac{1}{\tilde{u}_m + \tilde{u}_k} \right]$$

- Hence the minimizer of $g(u_m)$ is obtained by the solution of $g'(u_m) = 0$, which is given by

$$\hat{u}_m = \left[\sum_{k=1}^{p} w_{mk} \right] \sum_{k=1}^{p} \left[\frac{N_{mk}}{\tilde{u}_m + \tilde{u}_k} \right]^{-1}$$

where $N_{mk} = w_{mk} + w_{km}$ (the number of games between j and k).

(j,k)	1	2	3	$\cdots$	k	$\cdots$	p	$\sum_{k=1}^{p} w_{jk}$
1	0	1	2	$\cdots$	w_{1k}	$\cdots$	2	17
2	3	0	1	$\cdots$	w_{2k}	$\cdots$	0	12
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p	1	1	0	$\cdots$	w_{pk}	$\cdots$	0	21
$\sum_{j=1}^{p} w_{jk}$	10	5	2	$\cdots$	w_{pk}	$\cdots$	0	N

Table 1: summary of pairwise comparisons

(w_{jk}: # of i's wins against j, $N_{jk} = w_{jk} + w_{kj}$: # of matches between i and j)

One cycle of MM algorithm

(1) Let $\tilde{\mathbf{u}}$ and set $m = 1$

(2) Obtain $\hat{u}_m$.

(3) Update the m-th coordinate of $\tilde{\mathbf{u}}$ by $\hat{u}_m$ and $m \leftarrow m + 1$.

(4) Repeat (2)-(3) until $m = p$.

Repeat the one cycle MM algorithm, we obtain the MLE of the Bradley-Terry model.

Discussion

- Newton Raphson algorithm is applicable to obtaining MLE of the Bradley-Terry model?
- What's the advantage of the MM algorithm for obtaining the MLE of the Bradley-Terry model?
- Read [Hunter and Lange, 2004]

HW

- Prove (3).
- Prove (4).
- Prove (5).
- Write three manual codes of the logistic regression model with the gradient descent method, the Newton-Raphson method, and the BFGS algorithm.
- Write a manual code of the Huberized regression with MM-algorithm and check the descent property.
- Write a manual code of the Bradley-Terry model with MM algorithm and check the descent property.

Bradley, R. A. and Terry, M. E. (1952).
Rank analysis of incomplete block designs: I. the method of paired comparisons.
Biometrika, 39(3/4):324–345.

Friedman, J., Hastie, T., and Tibshirani, R. (2010).
Regularization paths for generalized linear models via coordinate descent.
Journal of statistical software, 33(1):1.

Hunter, D. R. (2004).
Mm algorithms for generalized bradley-terry models.
Annals of Statistics, pages 384–406.

Hunter, D. R. and Lange, K. (2004).
A tutorial on mm algorithms.
The American Statistician, 58(1):30–37.