final report

Jitong Jiang

2026-02-24

Github Link:

https://github.com/jitongj/bios731_hw3_jiang.git

Problem 1:

Let (y_i, x_i), $i = 1, \dots, n$, be independent observations with $y_i \in \{0, 1\}$ and $x_i \in \mathbb{R}^p$. Define the linear predictor

$$\eta_i = x_i^\top \beta,$$

where $\beta = (\beta_0, \beta_1, \dots, \beta_p)^T$and the logistic mean function

$$\pi_i = P(Y_i = 1 \mid x_i) = \frac{1}{1 + \exp(-\eta_i)}.$$

Log-likelihood

The Bernoulli likelihood contribution is

$$P(Y_i = y_i \mid x_i) = \pi_i^{y_i} (1 - \pi_i)^{1-y_i},$$

so

$$L(\beta) = \prod_{i=1}^n (\pi_i^{y_i} (1 - \pi_i)^{1-y_i}) = \prod_{i=1}^n (\frac{1}{1 + \exp(-\eta_i)})^{y_i} (\frac{\exp(-\eta_i)}{1 + \exp(-\eta_i)})^{1-y_i}$$

$$\begin{aligned} \ell(\beta) &= \sum_{i=1}^n \Big(y_i \log \pi_i + (1 - y_i) \log(1 - \pi_i) \Big) \\ &= \sum_{i=1}^n \Big(y_i \eta_i - \log(1 + e^{\eta_i}) \Big) \\ &= \sum_{i=1}^n \Big(y_i x_i^\top \beta - \log(1 + e^{x_i^\top \beta}) \Big). \end{aligned}$$

Gradient

Differentiate $\ell(\beta) = \sum_{i=1}^n \Big(y_i x_i^\top \beta - \log(1 + e^{x_i^\top \beta}) \Big)$ with respect to β, we obtain

$$\nabla \ell(\beta) = \sum_{i=1}^n (y_i x_i - \frac{x_i e^{x_i^\top \beta}}{1 + e^{x_i^\top \beta}}) = \sum_{i=1}^n (y_i - \pi_i)\, x_i.$$

Let $X \in \mathbb{R}^{n \times p}$ have rows $x_i^\top$, $y = (y_1, \dots, y_n)^\top$, and $\pi = (\pi_1, \dots, \pi_n)^\top$. Then

$$\nabla \ell(\beta) = X^\top (y - \pi)$$

Hessian

Using $\frac{d\pi_i}{d\eta_i} = \pi_i(1-\pi_i)$ and $\eta_i = x_i^\top \beta$,

$$\begin{aligned}
\nabla^2 \ell(\beta) &= \frac{d}{d\beta} \sum_{i=1}^n (y_i - \pi_i)\, x_i \\
&= -\sum_{i=1}^n \frac{d\pi_i}{d\eta_i} \frac{d\eta_i}{d\beta} x_i^\top \\
&= -\sum_{i=1}^n \pi_i(1-\pi_i) x_i x_i^\top
\end{aligned}$$

Define matrix $W = \operatorname{diag}(w_1, \dots, w_n)$ with $w_i = \pi_i(1-\pi_i)$. Then the Hessian can be written as

$$\nabla^2 \ell(\beta) = -X^\top W X$$

Newton update

Newton's method for maximizing $\ell(\beta)$ updates

$$\beta^{(t+1)} = \beta^{(t)} - \left[\nabla^2 \ell(\beta^{(t)})\right]^{-1} \nabla \ell(\beta^{(t)}).$$

Thus

$$\beta^{(t+1)} = \beta^{(t)} + \left(X^\top W^{(t)} X\right)^{-1} X^\top (y - \pi^{(t)}),$$

where $\pi^{(t)} = \frac{1}{1+e^{-X\beta^{(t)}}}$ and $W^{(t)} = \operatorname{diag}(\pi_i^{(t)}(1-\pi_i^{(t)}))$.

Convexity

Because $w_i = \pi_i(1-\pi_i) \geq 0$, the matrix $X^\top W X$ is positive semi-definite, $\nabla^2 \ell(\beta) = -X^\top W X \leq 0$, so the $\ell(\beta)$ is concave.

Problem 2:

2.A

Let

$$u(\theta) = 1 + \exp(x_i^\top \theta), \qquad u_k = u(\theta^{(k)}) = 1 + \exp(x_i^\top \theta^{(k)}), \qquad f(u) = -\log u$$

Since $f''(u) = 1/u^2 > 0$ for $u > 0$, f is convex.
By the first-order convexity inequality:

$$f(u) \geq f(u_k) + f'(u_k)(u - u_k).$$

Because

$$f'(u) = -\frac{1}{u},$$

we have

$$-\log u \geq -\log u_k - \frac{1}{u_k}(u - u_k).$$

Plug in $u(\theta) = 1 + \exp(x_i^\top \theta)$:

$$-\log\{1+\exp(x_i^\top\theta)\} \geq -\log\{1+\exp(x_i^\top\theta^{(k)})\} - \frac{\exp(x_i^\top\theta)-\exp(x_i^\top\theta^{(k)})}{1+\exp(x_i^\top\theta^{(k)})}.$$

When $\theta=\theta^{(k)}$, we have $u=u_k$, so equality holds.

2.B

From A, we get:

$$\ell(\theta)=\sum_{i=1}^n \left(y_i x_i^\top\theta - \log(1+e^{x_i^\top\theta})\right) \geq \sum_{i=1}^n y_i x_i^\top\theta - \sum_{i=1}^n \frac{\exp(x_i^\top\theta)}{1+\exp(x_i^\top\theta^{(k)})} + \text{const.}$$

Define $a_i^{(k)}=\frac{\exp(x_i^\top\theta^{(k)})}{1+\exp(x_i^\top\theta^{(k)})}$.

Since

$$\exp(x_i^\top\theta)=\exp(x_i^\top\theta^{(k)})\exp\{x_i^\top(\theta-\theta^{(k)})\},$$

we have

$$\frac{\exp(x_i^\top\theta)}{1+\exp(x_i^\top\theta^{(k)})}=a_i^{(k)}\exp\{x_i^\top(\theta-\theta^{(k)})\}.$$

Using the AM–GM inequality: $\left(\prod_{j=1}^p u_j\right)^{1/p} \leq \frac{1}{p}\sum_{j=1}^p u_j$.

let

$$u_j=\exp\{p x_{ij}(\theta_j-\theta_j^{(k)})\}.$$

Then

$$\exp\{x_i^\top(\theta-\theta^{(k)})\} \leq \frac{1}{p}\sum_{j=1}^p \exp\{p x_{ij}(\theta_j-\theta_j^{(k)})\}.$$

Therefore,

$$-a_i^{(k)}\exp\{x_i^\top(\theta-\theta^{(k)})\} \geq -\frac{a_i^{(k)}}{p}\sum_{j=1}^p \exp\{p x_{ij}(\theta_j-\theta_j^{(k)})\}.$$

Hence a minorizing function is

$$\begin{aligned} g(\theta \mid \theta^{(k)}) &= -\frac{1}{p}\sum_{i=1}^n a_i^{(k)}\sum_{j=1}^p \exp\{p x_{ij}(\theta_j-\theta_j^{(k)})\} + \sum_{i=1}^n y_i x_i^\top\theta \\ &= -\frac{1}{p}\sum_{i=1}^n \frac{\exp(x_i^\top\theta^{(k)})}{1+\exp(x_i^\top\theta^{(k)})}\sum_{j=1}^p \exp\{p x_{ij}(\theta_j-\theta_j^{(k)})\} + \sum_{i=1}^n y_i x_i^\top\theta. \end{aligned}$$

2.C

Since g is differentiable, maximizing $g(\theta \mid \theta^{(k)})$ means solving

$$\frac{\partial}{\partial\theta_j} g(\theta \mid \theta^{(k)})=0, \quad j=1,\dots,p.$$

From B:

$$g(\theta \mid \theta^{(k)}) = -\frac{1}{p}\sum_{i=1}^{n} \frac{\exp(x_i^\top \theta^{(k)})}{1+\exp(x_i^\top \theta^{(k)})} \sum_{j=1}^{p} \exp\{p x_{ij}(\theta_j - \theta_j^{(k)})\} + \sum_{i=1}^{n} y_i x_i^\top \theta + \text{const}.$$

Thus:

$$\frac{\partial}{\partial \theta_j} g(\theta \mid \theta^{(k)}) = -\sum_{i=1}^{n} \frac{\exp(x_i^\top \theta^{(k)})}{1+\exp(x_i^\top \theta^{(k)})} x_{ij} \exp\{p x_{ij}(\theta_j - \theta_j^{(k)})\} + \sum_{i=1}^{n} y_i x_{ij}.$$

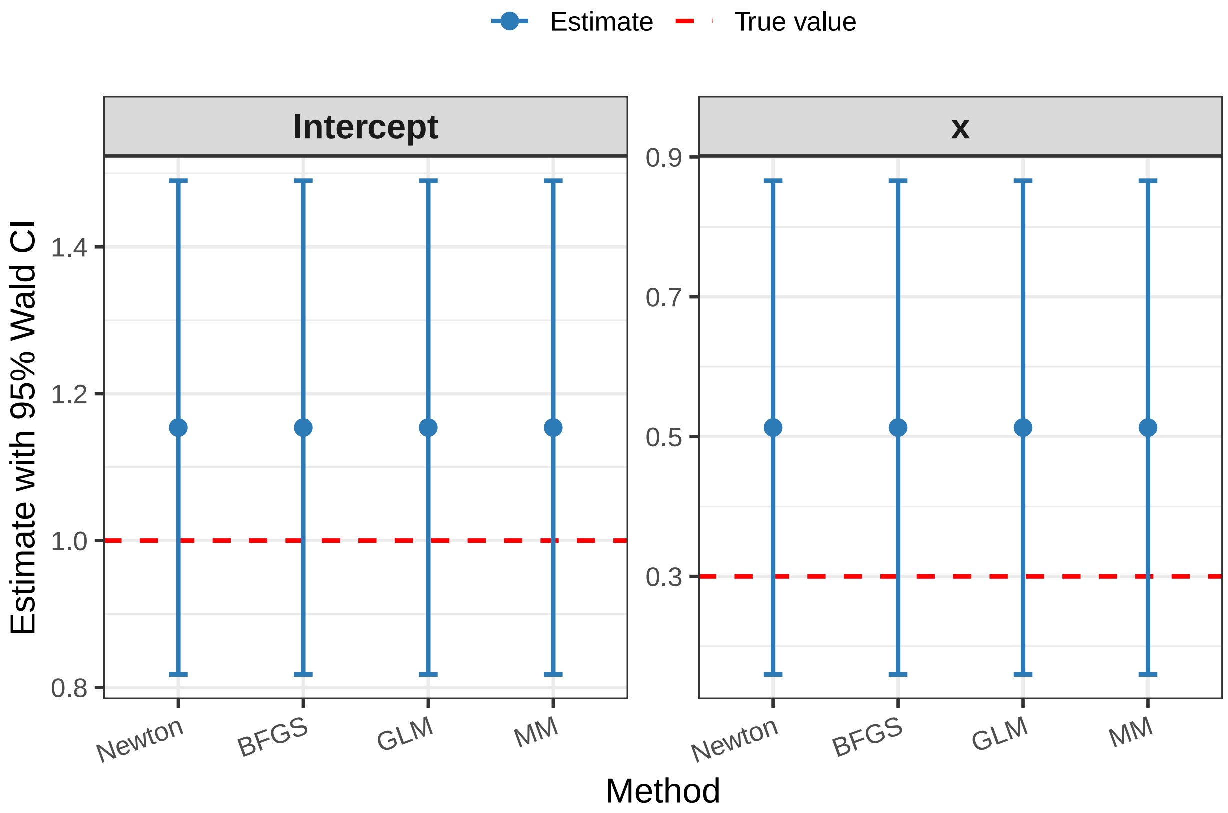
Thus we have

$$-\sum_{i=1}^{n} \frac{\exp(x_i^\top \theta^{(k)})}{1+\exp(x_i^\top \theta^{(k)})} x_{ij} \exp(-p x_{ij} \theta_j^{(k)}) \exp(p x_{ij} \theta_j) + \sum_{i=1}^{n} y_i x_{ij} = 0, \quad j = 1, \dots, p.$$

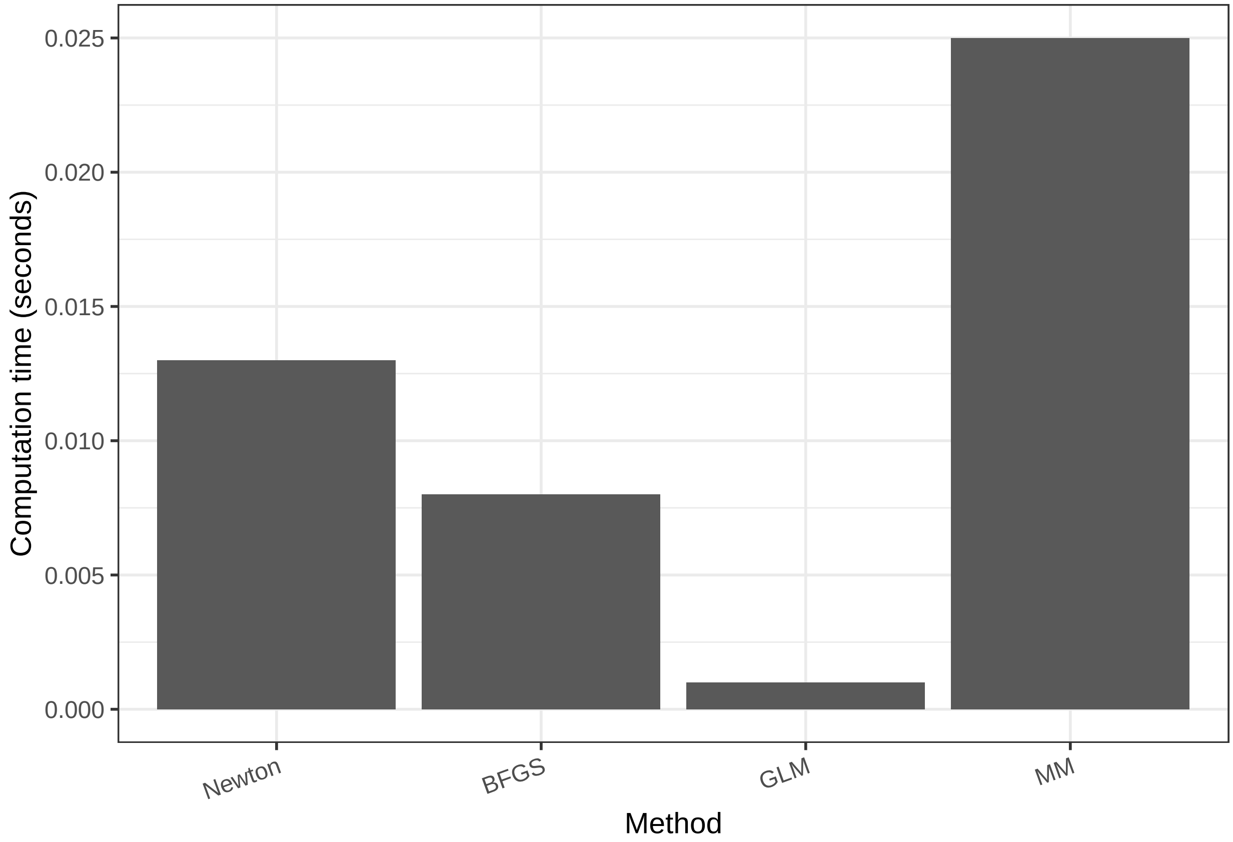
Problem 3:

```r
library(dplyr)
library(ggplot2)
library(here)
library(patchwork)
set.seed(2026)
res_table <- readRDS(here("data", "summary_results.rds"))
beta_true <- c(Intercept = 1, x = 0.3)

method_order <- c("Newton", "BFGS", "GLM", "MM")

plot_dat <- res_table %>%
  mutate(
    method = factor(method, levels = method_order),
    term = factor(term, levels = c("Intercept", "x")),
    true_value = ifelse(term == "Intercept", beta_true["Intercept"], beta_true["x"])
  )
```

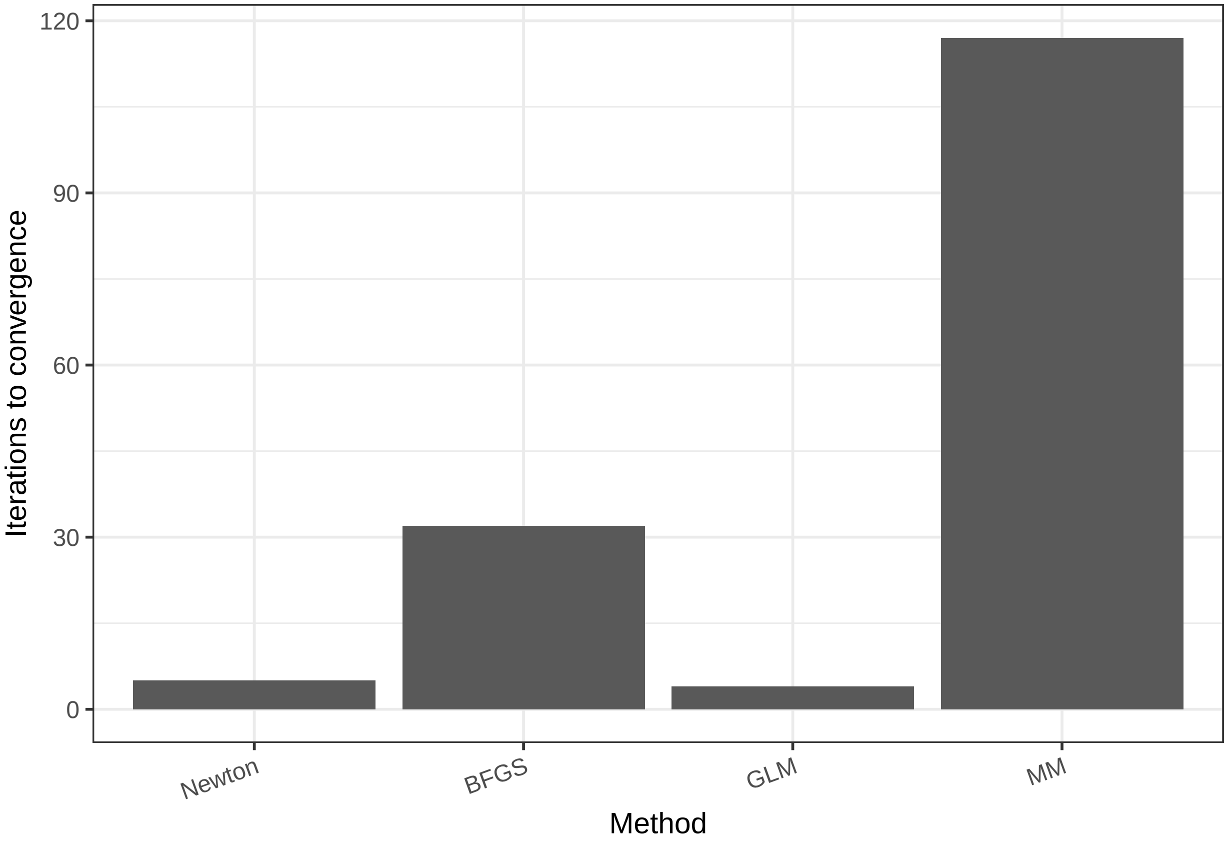
```r
p_est_ci <- ggplot(plot_dat, aes(x = method, y = estimate)) +
  geom_hline(
    aes(yintercept = true_value, color = "True value"),
    linetype = "dashed",
    linewidth = 0.8
  ) +
  geom_point(
    aes(color = "Estimate"),
    size = 2.5
  ) +
  geom_errorbar(
    aes(ymin = ci_lower, ymax = ci_upper, color = "Estimate"),
    width = 0.15,
    linewidth = 0.8
  ) +

  facet_wrap(~ term, scales = "free_y") +

  scale_color_manual(
    name = "",
    values = c("True value" = "red",
               "Estimate" = "#2C7BB6")
  ) +

  labs(
    x = "Method",
    y = "Estimate with 95% Wald CI"
  ) +

  theme_bw(base_size = 13) +
  theme(
    strip.text = element_text(size = 13, face = "bold"),
    axis.text.x = element_text(angle = 20, hjust = 1),
    legend.position = "top"
  )

time_dat <- plot_dat %>%
  select(method, time_sec) %>%
  distinct()

p_time <- ggplot(time_dat, aes(x = method, y = time_sec)) +
  geom_col() +
  labs(
    x = "Method",
    y = "Computation time (seconds)"
  ) +
  theme_bw() +
  theme(
    axis.text.x = element_text(angle = 20, hjust = 1)
  )

iter_dat <- plot_dat %>%
  select(method, iterations) %>%
  distinct()

p_iter <- ggplot(iter_dat, aes(x = method, y = iterations)) +
  geom_col() +
  labs(
    x = "Method",
    y = "Iterations to convergence"
  ) +
  theme_bw() +
  theme(
    axis.text.x = element_text(angle = 20, hjust = 1)
  )


p_est_ci
```


Estimate
True value
Intercept
x
Estimate with 95% Wald CI
1.4
1.2
1.0
0.8
0.9
0.7
0.5
0.3
Newton
BFGS
GLM
MM
Method

```
p_time
```


Computation time (seconds)
0.025
0.020
0.015
0.010
0.005
0.000
Newton
BFGS
GLM
MM
Method

```
p_iter
```



All four optimization methods (Newton, BFGS, GLM, and MM) has nearly identical estimates for both β_0 and β_1. This is expected because the logistic log-likelihood is concave with a unique global maximum, so all properly implemented algorithms converge to the same MLE. Although the estimates differ from the true parameter values due to sampling variability in a single dataset of size n = 200, the true values lie within the 95% Wald confidence intervals, indicating appropriate inferential performance. In terms of computation, GLM is the fastest with least iterations, followed by BFGS and Newton, while MM requires substantially more iterations and longer runtime. Overall, all methods are accurate and stable, but GLM are more computationally efficient in this setting, whereas MM is comparatively slower.

Problem 4:

Since independent survival times $t_1, \dots, t_n \sim \text{Exponential}(\lambda)$, the density is :

$$f(t \mid \lambda) = \lambda e^{-\lambda t}, \qquad t \geq 0.$$

We observe right-censored data

$$y_i = \min(t_i, c_i), \qquad \delta_i = \mathbf{1}(t_i \leq c_i),$$

so $t_i = y_i$ if $\delta_i = 1$, while t_i is missing but satisfies $t_i > c_i$ if $\delta_i = 0$.

If the complete event times $t_1, \dots, t_n$ were observed,

$$\ell_c(\lambda; t) = \sum_{i=1}^{n} \log f(t_i \mid \lambda) = \sum_{i=1}^{n} (\log \lambda - \lambda t_i) = n \log \lambda - \lambda \sum_{i=1}^{n} t_i.$$

- For E-step

At iteration k,

$$Q(\lambda \mid \lambda^{(k)}) = \mathbb{E}_{\lambda^{(k)}}[\ell_c(\lambda; t) \mid y, \delta] = n \log \lambda - \lambda \sum_{i=1}^{n} \mathbb{E}_{\lambda^{(k)}}[t_i \mid y_i, \delta_i]$$

We have

$$\mathbb{E}_{\lambda^{(k)}}[t_i \mid y_i, \delta_i] = \begin{cases} y_i, & \delta_i = 1, \\ c_i + \dfrac{1}{\lambda^{(k)}}, & \delta_i = 0, \quad \text{(since memoryless in Exp)} \end{cases}$$

Let

$$S^{(k)} = \sum_{i:\delta_i=1} y_i + \sum_{i:\delta_i=0} \left(c_i + \frac{1}{\lambda^{(k)}} \right).$$

Then

$$Q(\lambda \mid \lambda^{(k)}) = n \log \lambda - \lambda S^{(k)}.$$

- For M-step

Maximizing $Q(\lambda \mid \lambda^{(k)})$:

$$\frac{\partial}{\partial \lambda} Q(\lambda \mid \lambda^{(k)}) = \frac{n}{\lambda} - S^{(k)} = 0 \quad \Rightarrow \quad \lambda^{(k+1)} = \frac{n}{S^{(k)}}.$$

where

$$S^{(k)} = \sum_{\delta_i=1} y_i + \sum_{\delta_i=0} \left(c_i + \frac{1}{\lambda^{(k)}} \right),$$

Start with a initial value $\lambda^{(0)} > 0$, repeat the above update until converge.

```r
source(here::here("source","em_exp_censored.r"))

library(survival)
data(veteran)

y <- veteran$time
delta <- ifelse(veteran$status == 1, 1, 0)


fit_em <- em_exp_censored(
  y = y,
  delta = delta,
  lambda_init = 0.01,
  tol = 1e-8
)

cat("EM results:\n")
```

```
## EM results:
```

```r
cat("Iterations:", fit_em$iterations, "\n")
```

```
## Iterations: 6
```

```r
cat("Lambda_hat:", round(fit_em$lambda_hat, 6), "\n")
```

```
## Lambda_hat: 0.007682
```

We assume the convergence is reached if $|\lambda^{(k+1)} - \lambda^{(k)}| < 10^{-8}$.

The EM algorithm was initialized at $\lambda^{(0)} = 0.01$ and converged in 6 iterations using the stopping rule. The final estimate was $\hat{\lambda} = 0.00768$.

Since the observed log-likelihood is

$$\ell(\lambda) = \sum_i \delta_i \log \lambda - \lambda \sum_i y_i$$

Taking the first derivative of it, we get $\ell'(\lambda) = \frac{\sum_i \delta_i}{\lambda} - \sum_i y_i$. Set it to 0, then we have MLE:

$$\hat{\lambda} = \frac{\sum_i \delta_i}{\sum_i y_i}$$

Taking second derivative of it, we get:

$$\ell''(\lambda) = -\frac{\sum \delta_i}{\lambda^2}$$

$$\operatorname{Avar}(\hat{\lambda}) = I(\hat{\lambda})^{-1} = \big(-\ell''(\hat{\lambda})\big)^{-1} = \left(\frac{\sum_i \delta_i}{\hat{\lambda}^2}\right)^{-1} = \frac{\hat{\lambda}^2}{\sum_i \delta_i}.$$

Since $\lambda > 0$, we apply a log transformation.
By the delta method,

$$\operatorname{Avar}(\log \hat{\lambda}) = \left(\frac{1}{\hat{\lambda}}\right)^2 \cdot \frac{\hat{\lambda}^2}{\sum_i \delta_i} = \frac{1}{\sum_i \delta_i}.$$

Thus, $\operatorname{SE}(\log \hat{\lambda}) = \frac{1}{\sqrt{\sum \delta_i}}$, which means:

The 95% confidence interval constructed on the log-scale is:

$$\log \hat{\lambda} \; \pm \; 1.96 \cdot \frac{1}{\sqrt{\sum_i \delta_i}}.$$

Transforming back to the original scale gives

$$\left(\exp\left(\log \hat{\lambda} - 1.96 \cdot \frac{1}{\sqrt{\sum_i \delta_i}} \right), \; \exp\left(\log \hat{\lambda} + 1.96 \cdot \frac{1}{\sqrt{\sum_i \delta_i}} \right) \right).$$

```
d <- sum(delta)

lambda_hat <- fit_em$lambda_hat

avar_lambda <- lambda_hat^2 / d
avar_log_lambda <- 1 / d

se_log_lambda <- sqrt(avar_log_lambda)

lower <- exp(log(lambda_hat) - 1.96 * se_log_lambda)
upper <- exp(log(lambda_hat) + 1.96 * se_log_lambda)
```

```
cat(sprintf("Lambda_hat = %.6f\n", lambda_hat))
```

```
## Lambda_hat = 0.007682
```

```
cat(sprintf("95%% CI = (%.6f, %.6f)\n", lower, upper))
```

```
## 95% CI = (0.006460, 0.009135)
```

```
fit_aft <- survreg(Surv(y, delta) ~ 1, dist = "exponential")

mu_hat <- as.numeric(coef(fit_aft)[1])
se_mu  <- sqrt(vcov(fit_aft)[1, 1])

lambda_hat_aft <- exp(-mu_hat)

# CI for lambda via on log scale
mu_lower <- mu_hat - 1.96 * se_mu
mu_upper <- mu_hat + 1.96 * se_mu

# exp(-mu)
ci_lambda_aft <- c(exp(-mu_upper), exp(-mu_lower))

cat(sprintf("AFT: lambda_hat = %.6f\n", lambda_hat_aft))
```

```
## AFT: lambda_hat = 0.007682
```

```
cat(sprintf("AFT 95%% CI for lambda: (%.6f, %.6f)\n", ci_lambda_aft[1], ci_lambda_aft[2]))
```

```
## AFT 95% CI for lambda: (0.006460, 0.009135)
```

Thus the exponential AFT model fitted using survreg() has the same $\hat{\lambda} = 0.00768$ and 95% confidence interval (0.00646, 0.00914) as EM algorithm. This is because both the EM algorithm and the AFT model maximize the same exponential likelihood.

Extra Part1

- **Louis' method**

Let Y denote the observed data and Z the missing data. Based on Lecture note, for Louis' method we have:

$$I_Y(\theta) = I_{Y,Z}(\theta) - I_{Z|Y}(\theta),$$

$$I_{Z|Y}(\theta) = \mathbb{E}[S_c(\theta)S_c(\theta)^\top \mid Y] - S_o(\theta)S_o(\theta)^\top.$$

$$I_Y(\hat{\theta}) = I_{Y,Z}(\hat{\theta}) - \mathbb{E}\left[S_c(\hat{\theta})S_c(\hat{\theta})^\top \mid Y\right], \quad \hat{\theta} \text{ is MLE}$$

For the exponential survival model, as we get from Problem 4:

$$\ell_c(\lambda) = n \log \lambda - \lambda \sum_{i=1}^{n} t_i,$$

so the complete-data score and information are

$$S_c(\lambda) = \frac{\partial \ell_c}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} t_i, \qquad I_{Y,Z}(\lambda) = -\frac{\partial^2 \ell_c}{\partial \lambda^2} = \frac{n}{\lambda^2}.$$

For right censoring, using the memoryless in Exp, for $\delta_i = 0$ we have $t_i = y_i + E_i$ with $E_i \sim \text{Exponential}(\lambda)$, hence $\text{Var}(t_i \mid Y) = \text{Var}(E_i) = 1/\lambda^2$, while for $\delta_i = 1$. Therefore,

$$\text{Var}\left(\sum_{i=1}^{n} t_i \mid Y\right) = \sum_{i:\delta_i=0} \text{Var}(t_i \mid Y) = \frac{n - \sum_{i=1}^{n} \delta_i}{\lambda^2}.$$

Since $S_c(\lambda) = n/\lambda - \sum_i t_i$, we have

$$\mathbb{E}[S_c(\lambda)^2 \mid Y] = \text{Var}(S_c(\lambda) \mid Y) = \text{Var}\left(\sum_i t_i \mid Y\right) = \frac{n - \sum_i \delta_i}{\lambda^2}.$$

Thus

$$I_Y(\lambda) = \frac{n}{\lambda^2} - \frac{n - \sum_i \delta_i}{\lambda^2} = \frac{\sum_i \delta_i}{\lambda^2}.$$

At $\hat\lambda$ we have

$$\text{Avar}(\hat\lambda) = I_Y(\hat\lambda)^{-1} = \frac{\hat\lambda^2}{\sum_i \delta_i}, \qquad \text{Avar}(\log\hat\lambda) = \frac{1}{\sum_i \delta_i}.$$

95% CI using the log-scale Wald form

$$\left(\exp\left(\log\hat\lambda - 1.96/\sqrt{\sum_i \delta_i}\right),\ \exp\left(\log\hat\lambda + 1.96/\sqrt{\sum_i \delta_i}\right)\right).$$

```r
# Louis' method
d <- sum(delta)
lambda_hat <- fit_em$lambda_hat

se_log_louis <- 1 / sqrt(d)
ci_louis <- exp(log(lambda_hat) + c(-1, 1) * 1.96 * se_log_louis)

cat(sprintf("Louis: lambda_hat = %.6f\n", lambda_hat))
```

```
## Louis: lambda_hat = 0.007682
```

```r
cat(sprintf("Louis 95%% CI: (%.6f, %.6f)\n", ci_louis[1], ci_louis[2]))
```

```
## Louis 95% CI: (0.006460, 0.009135)
```

- **Bootstrap**

The EM algorithm was initialized at $\lambda^{(0)} = 0.01$ and converged to $\hat\lambda$. We applieder a bootstrap with $B = 2000$ resamples by sampling the observed pairs (y_i, δ_i) with replacement. For each bootstrap sample b, we computed

$$\hat\lambda^{*(b)} = \frac{\sum_i \delta_i^{*(b)}}{\sum_i y_i^{*(b)}},$$

```r
B <- 2000
n <- length(y)

lambda_star <- numeric(B)

for (b in 1:B) {
  idx <- sample.int(n, size = n, replace = TRUE)
  y_b <- y[idx]
  d_b <- delta[idx]
  lambda_star[b] <- sum(d_b) / sum(y_b)
}

se_boot <- sd(lambda_star)
ci_boot <- quantile(lambda_star, probs = c(0.025, 0.975), names = FALSE)

cat(sprintf("Bootstrap: SE = %.6f\n", se_boot))
```

```
## Bootstrap: SE = 0.000876
```

```
cat(sprintf("Bootstrap 95%% CI: (%.6f, %.6f)\n", ci_boot[1], ci_boot[2]))
```

```
## Bootstrap 95% CI: (0.006176, 0.009687)
```

Thus the results shows that Using B=2000 bootstrap resamples, the bootstrap the 95% percentile CI was (0.006176, 0.009687), while Louis' method is (0.006460, 0.009135). The two intervals are similar, with the bootstrap CI slightly wider due to finite-sample variability, while the Louis CI relies on an asymptotic approximation.

Extra Part2:

A

For logistic regression with independent $Y_i \in \{0, 1\}$ and

$$\mu_i = \Pr(Y_i = 1 \mid x_i) = \frac{1}{1 + e^{-x_i^\top \beta}}.$$

$$\ell(\beta) = \sum_{i=1}^{n} \Big[y_i \log(\mu_i) + (1 - y_i) \log(1 - \mu_i) \Big] = \sum_{i=1}^{n} \Big(y_i x_i^\top \beta - \log(1 + e^{x_i^\top \beta}) \Big).$$

The score function is

$$S(\beta) = \ell'(\beta) = \sum_{i=1}^{n} x_i (y_i - \mu_i) = X^\top (y - \mu).$$

The Hessian is

$$\ell''(\beta) = -\sum_{i=1}^{n} \mu_i (1 - \mu_i)\, x_i x_i^\top = -X^\top W X,$$

where

$$W = \operatorname{diag}\big(\mu_i (1 - \mu_i)\big).$$

Hence the observed information is

$$I(\beta, Y) = -\ell''(\beta) = X^\top W X.$$

The expected Fisher information is

$$I(\beta) = \mathbb{E}_Y[-\ell''(\beta) \mid X] = \mathbb{E}_Y[X^\top W X \mid X]$$

Since $X^\top W X$ depends only on μ_i , which is only related to X and β, not on Y_i, therefore we have

$$I(\beta) = X^\top W X = I(\beta, Y).$$

Thus, in logistic regression, the expected information equals the observed information.

B

For probit regression,

$$\mu_i = \Pr(Y_i = 1 \mid x_i) = \Phi(\eta_i), \qquad \eta_i = x_i^\top \beta,$$

where Φ and ϕ are the standard normal CDF and PDF.

The log-likelihood is

$$\ell(\beta) = \sum_{i=1}^{n} \Big[y_i \log(\mu_i) + (1 - y_i) \log(1 - \mu_i) \Big].$$

So

$$S(\beta) = \ell'(\beta) = \sum_{i=1}^{n} \frac{\partial \ell}{\partial \mu_i} \frac{d\mu_i}{d\eta_i} \frac{\partial \eta_i}{\partial \beta} = \sum_{i=1}^{n} (\frac{y_i}{\mu_i} - \frac{1 - y_i}{1 - \mu_i})\, \phi(\eta_i)\, x_i = \sum_{i=1}^{n} x_i\, (y_i - \mu_i)\, \frac{\phi(\eta_i)}{\mu_i(1 - \mu_i)}$$

When differentiating $U(\beta)$ to obtain the Hessian, terms involving $(y_i - \mu_i)$ remain in $\nabla^2 \ell(\beta)$ because the factor $\phi(\eta_i)/\{\mu_i(1 - \mu_i)\}$ depends on η_i and its derivative times with $(y_i - \mu_i)$.

Thus for the observed information $I(\beta, Y) = -\ell''(\beta)$ depends on the RV y_i.

For expected information:

$$I(\beta) = \mathbb{E}[-\ell''(\beta) \mid X]\,.$$

Since $\mathbb{E}(y_i - \mu_i \mid X) = 0$, the $(y_i - \mu_i)$ dependent parts $= 0$ after taking expectation, so in general

$$I(\beta) \neq I(\beta, Y)$$

for probit regression.

Therefore, expected and observed information coincide for logistic regression (canonical link) but not in general for probit regression.