Homework 2

STAT 950

Emily Robinson

October 1, 2019

Problem 1

On page 51 of the notes. It states that:

- Provided $\boldsymbol{x}^{(t+1)}$ was selected to satisty the Wolfe's conditions, then $-\boldsymbol{H}^{(t)}$ being positive definite implies that $-\boldsymbol{H}^{(t+1)}$ is also positive definite.

Show that the above statement is true.

Proof. Consider

$$H^{(t+1)} = \left(I - \frac{\boldsymbol{y}\boldsymbol{z}^T}{\rho}\right) H^{(t)} \left(I - \frac{\boldsymbol{z}\boldsymbol{y}^T}{\rho}\right) + \frac{\boldsymbol{z}\boldsymbol{z}^T}{\rho}$$

where

$$\begin{aligned}
\boldsymbol{z} &= \boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)} \\
\boldsymbol{y} &= \boldsymbol{g'}(\boldsymbol{x}^{(t+1)}) - \boldsymbol{g'}(\boldsymbol{x}^{(t)}) \\
\rho &= \boldsymbol{z}^T\boldsymbol{y}.
\end{aligned}$$

Let $A = \left(I - \frac{\boldsymbol{z}\boldsymbol{y}^T}{\rho}\right)$. Then since $-H^{(t)} > 0$, we know $H^{(t)} < 0$, negative definite. Then by definition of positive definite,

$$\left(I - \frac{\boldsymbol{y}\boldsymbol{z}^T}{\rho}\right) H^{(t)} \left(I - \frac{\boldsymbol{z}\boldsymbol{y}^T}{\rho}\right) = A^T H^{(t)} A < 0.$$

Then we know $\boldsymbol{z}\boldsymbol{z}^T > 0$. Then from pg. 49, since $\boldsymbol{x}^{(t+1)}$ satisfies wolfe's conditions, we know $\alpha^{(t)}, \boldsymbol{p}^{(t)} > 0$. Therefore,

$$\begin{aligned}
& \boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \alpha^{(t)}\boldsymbol{p}^{(t)} \\
\implies \quad & \boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)} = \alpha^{(t)}\boldsymbol{p}^{(t)} \\
\implies \quad & \boldsymbol{z} > 0.
\end{aligned}$$

Then for $c_1 \in (0,1)$ and $c_2 \in (c_1, 1)$,

$$\begin{aligned}
& [\boldsymbol{p}^{(t)}]^T \boldsymbol{g}'(\boldsymbol{x}^{(t)} + \alpha^{(t)}\boldsymbol{p}^{(t)}) \leq c_2 [\boldsymbol{p}^{(t)}]^T \boldsymbol{g}'(\boldsymbol{x}^{(t)}) \\
\implies & [\boldsymbol{p}^{(t)}]^T \boldsymbol{g}'(\boldsymbol{x}^{(t+1)}) \leq c_2 [\boldsymbol{p}^{(t)}]^T \boldsymbol{g}'(\boldsymbol{x}^{(t)}) \\
\implies & [\boldsymbol{p}^{(t)}]^T \left(\boldsymbol{g}'(\boldsymbol{x}^{(t+1)}) - c_2 \boldsymbol{g}'(\boldsymbol{x}^{(t)})\right) \leq 0 \\
\implies & \left(\boldsymbol{g}'(\boldsymbol{x}^{(t+1)}) - c_2 \boldsymbol{g}'(\boldsymbol{x}^{(t)})\right) \leq 0 \\
\implies & \left(\boldsymbol{g}'(\boldsymbol{x}^{(t+1)}) - \boldsymbol{g}'(\boldsymbol{x}^{(t)})\right) \leq 0 \\
\implies & \boldsymbol{y} \leq 0.
\end{aligned}$$

Therefore, $\rho = \boldsymbol{z}^T \boldsymbol{y} \leq 0$. Thus,

$$H^{(t+1)} = \left(I - \frac{\boldsymbol{y}\boldsymbol{z}^T}{\rho}\right) H^{(t)} \left(I - \frac{\boldsymbol{z}\boldsymbol{y}^T}{\rho}\right) + \frac{\boldsymbol{z}\boldsymbol{z}^T}{\rho} < 0 \implies -H^{(t+1)} > 0$$

and the statement holds true. $\square$

Problem 2

Using the data and model from problem 2.3.

```r
leukemia <- read.csv("leukemia.dat", sep = "")
kable(head(leukemia))
```

remissiontime	censored	group
6	TRUE	treatment
6	FALSE	treatment
6	FALSE	treatment
6	FALSE	treatment
7	FALSE	treatment
9	TRUE	treatment

(a) Derive the log likelihood given in 2.3 a). Note: Censored values use a pmf while uncensored values use a pdf when forming the likelihood function.

Let $w_i = 1$ if t_i is an uncensored time and $w_i = 0$ if t_i is a censored time. Then for $w_i = 1$, the individual relapsed at time t_i with pdf, $f(t_i) = h(t_i|x_i)S(t_i)$ and for $w_i = 0$, the individual is censored and their relapse exceeds t_i with probability $S(t_i)$. Therefore,

$$\begin{aligned}
L(\boldsymbol{\theta}|t) &= \prod_{i=1}^{n} h(t_i|x_i)^{w_i} S(t_i) \\
&= \prod_{i=1}^{n} \left([\lambda(t_i)\exp\{\boldsymbol{x}_i^T\boldsymbol{\beta}\}]^{w_i} \exp\{-\Lambda(t_i)\exp\{\boldsymbol{x}_i^T\boldsymbol{\beta}\}\} \right) \\
&= \prod_{i=1}^{n} \left(\left[\frac{\lambda(t_i)\Lambda(t_i)\exp\{\boldsymbol{x}_i^T\boldsymbol{\beta}\}}{\Lambda(t_i)} \right]^{w_i} \exp\{-\Lambda(t_i)\exp\{\boldsymbol{x}_i^T\boldsymbol{\beta}\}\} \right) \\
&= \prod_{i=1}^{n} \left(\frac{\mu_i^{w_i}\lambda(t_i)^{w_i}}{\Lambda(t_i)^{w_i}} \right) \exp\{\sum_{i=1}^{n}(-\mu_i)\} \\
\implies \qquad \log L &= \sum_{i=1}^{n} w_i \log(\mu_i) + \sum_{i=1}^{n} w_i \log\left(\frac{\lambda(t_i)}{\Lambda(t_i)}\right) - \sum_{i=1}^{n} \mu_i \\
&= \sum_{i=1}^{n} (w_i \log(\mu_i) - \mu_i) + \sum_{i=1}^{n} w_i \log\left(\frac{\lambda(t_i)}{\Lambda(t_i)}\right)
\end{aligned}$$

where $\mu_i = \Lambda(t_i)\exp\{\boldsymbol{x}_i^T\boldsymbol{\beta}\}$.

(b) Derive the score function and Hessian for α and $\boldsymbol{\beta}$.

Consider $\mu_i = t_i^{\alpha}\exp\{\boldsymbol{X}_i^T\boldsymbol{\beta}\}$ and $\lambda(t_i) = \alpha t_i^{\alpha-1}$. Then the log likelihood is

$$\log L(t_i|\alpha, \boldsymbol{\beta}) = \sum_{i=1}^{n} w_i \log(t_i^{\alpha}\exp\{\boldsymbol{X}_i^T\boldsymbol{\beta}\}) - t_i^{\alpha}\exp\{\boldsymbol{X}_i^T\boldsymbol{\beta}\} + w_i \log\left(\frac{\alpha}{t_i}\right).$$

Computing,

$$\begin{aligned}
\frac{dl}{d\alpha} &= \sum_{i=1}^{n} w_i \log(t_i) - t_i^{\alpha}\log(t_i)\exp\{\boldsymbol{X}_i^T\boldsymbol{\beta}\} + \frac{w_i}{\alpha} \\
\frac{dl}{d\beta_j} &= \sum_{i=1}^{n} w_i X_{ij} - t_i^{\alpha} X_{ij}\exp\{\boldsymbol{X}_i^T\boldsymbol{\beta}\}.
\end{aligned}$$

Therefore, the score function, $\boldsymbol{S}(\boldsymbol{\theta}) = (\frac{dl}{d\alpha}, \frac{dl}{d\beta_j})^T$.

Then computing

$$\begin{aligned}
\frac{d^2 l}{d\alpha^2} &= \sum_{i=1}^{n} -t_i^{\alpha}\log(t_i)^2\exp\{\boldsymbol{X}_i^T\boldsymbol{\beta}\} - \frac{w_i}{\alpha^2} \\
\frac{d^2 l}{d\alpha d\beta_j} &= \sum_{i=1}^{n} -t_i^{\alpha} X_{ij}\log(t_i)\exp\{\boldsymbol{X}_i^T\boldsymbol{\beta}\} \\
\frac{d^2 l}{d\beta_j d\beta_k} &= \sum_{i=1}^{n} -t_i^{\alpha} X_{ij}^2\exp\{\boldsymbol{X}_i^T\boldsymbol{\beta}\}
\end{aligned}$$

Therefore, the Hessian,

$$\boldsymbol{H} = \begin{pmatrix} \frac{d^2 l}{d\alpha^2} & \frac{d^2 l}{d\alpha d\beta_j} \\ \frac{d^2 l}{d\alpha d\beta_j} & \frac{d^2 l}{d\beta_j d\beta_k} \end{pmatrix}.$$

(c) Create a function that computes the log likelihood, score function, and Hessian given $alpha, \boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{y}$, and $\boldsymbol{w}$. You should include the function in your printed version of the homework.

```
logLikeSurvival <- function(theta, der = 0, X, y, w) {

    p <- length(theta)
    alpha <- theta[1]
    beta <- theta[2:3]

    value <- sum(w * log(y^alpha * exp(X %*% beta)) -
        y^alpha * exp(X %*% beta) + w * log(alpha/y))
    if (der == 0)
        return(value)

    der1 <- matrix(NA, nrow = 3, ncol = 1)
    der1[1] <- sum(w * log(y) - y^alpha * log(y) *
        exp(X %*% beta) + w/alpha)
    for (j in 2:p) {
        der1[j] <- sum(w * X[, j - 1] - y^alpha * X[,
            j - 1] * exp(X %*% beta))
    }
    if (der == 1)
        return(list(value = value, der1 = der1))

    der2 <- matrix(NA, nrow = 3, ncol = 3)
    der2[1, 1] <- sum(-y^alpha * log(y)^2 * exp(X %*%
        beta) - w/alpha^2)
    for (j in 2:p) {
        der2[1, j] <- der2[j, 1] <- sum(-y^alpha *
            X[, j - 1] * log(y) * exp(X %*% beta))
        for (k in 2:p) {
            der2[j, k] = der2[k, j] = sum(-y^alpha *
                X[, j - 1] * exp(X %*% beta))
        }
    }
    return(list(value = value, der1 = der1, der2 = der2))

}
```

(d) Verify numerically, that your function does in fact compute score function and Hessian correctly.

```r
alpha = 1
beta0 = 1
beta1 = 1
delta = 1e-06

evalTheta <- logLikeSurvival(c(alpha, beta0, beta1),
    der = 2, X = model.matrix(~leukemia$group), y = leukemia$remissiontime,
    w = as.numeric(!leukemia$censored))

evalAlphaDelta <- logLikeSurvival(c(alpha + delta,
    beta0, beta1), der = 2, X = model.matrix(~leukemia$group),
    y = leukemia$remissiontime, w = as.numeric(!leukemia$censored))

evalBeta0Delta <- logLikeSurvival(c(alpha, beta0 +
    delta, beta1), der = 2, X = model.matrix(~leukemia$group),
    y = leukemia$remissiontime, w = as.numeric(!leukemia$censored))

evalBeta1Delta <- logLikeSurvival(c(alpha, beta0, beta1 +
    delta), der = 2, X = model.matrix(~leukemia$group),
    y = leukemia$remissiontime, w = as.numeric(!leukemia$censored))

# Check score function
v1 <- evalTheta$der1[1]/((evalAlphaDelta$value - evalTheta$value)/delta)
v2 <- evalTheta$der1[2]/((evalBeta0Delta$value - evalTheta$value)/delta)
v3 <- evalTheta$der1[3]/((evalBeta1Delta$value - evalTheta$value)/delta)

# Check Hessian
v11 <- evalTheta$der2[1, 1]/((evalAlphaDelta$der1[1] -
    evalTheta$der1[1])/delta)
v12 <- evalTheta$der2[1, 2]/((evalBeta0Delta$der1[1] -
    evalTheta$der1[1])/delta)
v13 <- evalTheta$der2[1, 3]/((evalBeta1Delta$der1[1] -
    evalTheta$der1[1])/delta)

v21 <- evalTheta$der2[2, 1]/((evalAlphaDelta$der1[2] -
    evalTheta$der1[2])/delta)
v22 <- evalTheta$der2[2, 2]/((evalBeta0Delta$der1[2] -
    evalTheta$der1[2])/delta)
v23 <- evalTheta$der2[2, 3]/((evalBeta1Delta$der1[2] -
    evalTheta$der1[2])/delta)

v31 <- evalTheta$der2[3, 1]/((evalAlphaDelta$der1[3] -
```

```
    evalTheta$der1[3])/delta)
v32 <- evalTheta$der2[3, 2]/((evalBeta0Delta$der1[3] -
    evalTheta$der1[3])/delta)
v33 <- evalTheta$der2[3, 3]/((evalBeta1Delta$der1[3] -
    evalTheta$der1[3])/delta)

results <- rbind(v1, v2, v3, v11, v12, v13, v21, v22,
    v23, v31, v32, v33)
colnames(results) <- c("Ratio")
kable(t(round(results, 2)))
```

	v1	v2	v3	v11	v12	v13	v21	v22	v23	v31	v32	v33
Ratio	1	1	1	1	1	1	1	1	1	1	1	1

(e) Create a function that will compute the MLE of α and $\boldsymbol{\beta}$ along with their standard errors. You should include the function in your printed version of the homework.

```
newtonR <- function(f, xInit, maxIt = 20, relConvCrit = 1e-10,
    ...) {
    p = length(xInit)
    results = matrix(NA, maxIt, p + 2)
    colnames(results) = c("value", paste("x", 1:p,
        sep = ""), "Conv")

    xCurrent = xInit
    for (t in 1:maxIt) {
        evalF = f(xCurrent, der = 2, ...)
        results[t, "value"] = evalF$value
        results[t, 1 + (1:p)] = xCurrent
        xNext = xCurrent - solve(evalF$der2, evalF$der1)
        Conv = sqrt(crossprod(xNext - xCurrent))/(sqrt(crossprod(xCurrent)) +
            relConvCrit)
        results[t, "Conv"] = Conv
        if (Conv < relConvCrit)
            break
        xCurrent = xNext
    }

    evalFinal <- f(xNext, der = 2, ...)

    return(list(x = xNext, se = sqrt(diag(-solve(evalFinal$der2))),
        value = evalFinal$value, convergence = (Conv <
            relConvCrit), results = results[1:t, ]))
}
```

(f) Use the function you created to find the MLE and standard errors of α and $\boldsymbol{\beta}$.

```
ans = newtonR(logLikeSurvival, c(0.5, 0.25, 5), X = model.matrix(~leukemia$group),
    y = leukemia$remissiontime, w = as.numeric(!leukemia$censored),
    relConvCrit = 1e-14)
results <- cbind(ans$x, ans$se)
greeks = c(alpha = "a", beta0 = "ß0", beta1 = "ß1")
colnames(results) = c("Estimate", "StdErr")
rownames(results) = c(greeks["alpha"], greeks["beta0"],
    greeks["beta1"])
kable(round(results, 3))
```

	Estimate	StdErr
a	1.366	0.201
ß0	-3.071	0.558
ß1	-1.731	0.413

(g) What do you conclude about the effectiveness of the treatment? How did you arrive at that conclusion?

Testing $H_0 : \beta_1 = 0$ verses $H_A : \beta_1 \neq 0$, we calculate $\sqrt{Wald} = \frac{-1.730872}{0.4130819} = -4.19 \approx Z$ with a p-value of $0.000028 < 0.05$. Therefore, we have evidence to conclude that the treatment is effective.