Atlas-PS 3

David Atlas

September 12, 2018

Problem 1

We define the likelihood function $L(\hat{\alpha}; X)$:

$$L(\alpha; X) = \prod_{i=1}^{n} \frac{(\alpha_1 b_{i,1} + \alpha_2 b_{i,2})^{x_i} \mathrm{e}^{-(\alpha_1 b_{i,1} + \alpha_2 b_{i,2})}}{x_i!},$$

and the log-likelihood function $l(\hat{\alpha}; X)$:

$$l(\alpha; X) = \Sigma_{i=1}^{n} x_i \log \alpha_1 b_{i,1} + \alpha_2 b_{i,2} - \Sigma_{i=1}^{n} \alpha_1 b_{i,1} - \Sigma_{i=1}^{n} \alpha_2 b_{i,2} - \Sigma_{i=1}^{n} \log(x_i!).$$

a)

Derive the Newton Raphson update for finding the MLEs of α_1 and $alpha_2$.

First, we take the first derivative of l' with respect to $\hat{\alpha}$. This leaves us with a 2x1 matrix of first derivatives.

$$\begin{bmatrix} \Sigma_{i=1}^{n} \frac{x_i b_{i,1}}{\alpha_1 b_{i,1} + \alpha_2 b_{i,2}} - \Sigma_{i=1}^{n} b_{i,1} \\ \Sigma_{i=1}^{n} \frac{x_i b_{i,2}}{\alpha_1 b_{i,1} + \alpha_2 b_{i,2}} - \Sigma_{i=1}^{n} b_{i,2} \end{bmatrix}.$$

We find the Hessian:

$$\begin{bmatrix} \Sigma_{i=1}^{n} \frac{-x_i b_{i,1}^2}{(\alpha_1 b_{i,1} + \alpha_2 b_{i,2})^2} & \Sigma_{i=1}^{n} \frac{-x_i b_{i,1} b_{i,2}}{(\alpha_1 b_{i,1} + \alpha_2 b_{i,2})^2} \\ \Sigma_{i=1}^{n} \frac{-x_i b_{i,1} b_{i,2}}{(\alpha_1 b_{i,1} + \alpha_2 b_{i,2})^2} & \Sigma_{i=1}^{n} \frac{-x_i b_{i,2}^2}{(\alpha_1 b_{i,1} + \alpha_2 b_{i,2})^2} \end{bmatrix}$$

The Newton-Raphson update is $h = -\mathbf{l}''(\theta)^{-\mathbf{1}}\mathbf{l}'(\theta)$. We combine the two of them below:

$$h(\alpha) = -\begin{bmatrix} \Sigma_{i=1}^{n} \frac{-x_i b_{i,1}^2}{(\alpha_1 b_{i,1} + \alpha_2 b_{i,2})^2} & \Sigma_{i=1}^{n} \frac{-x_i b_{i,1} b_{i,2}}{(\alpha_1 b_{i,1} + \alpha_2 b_{i,2})^2} \\ \Sigma_{i=1}^{n} \frac{-x_i b_{i,1} b_{i,2}}{(\alpha_1 b_{i,1} + \alpha_2 b_{i,2})^2} & \Sigma_{i=1}^{n} \frac{-x_i b_{i,2}^2}{(\alpha_1 b_{i,1} + \alpha_2 b_{i,2})^2} \end{bmatrix}^{-1} \begin{bmatrix} \Sigma_{i=1}^{n} \frac{x_i b_{i,1}}{\alpha_1 b_{i,1} + \alpha_2 b_{i,2}} - \Sigma_{i=1}^{n} b_{i,1} \\ \Sigma_{i=1}^{n} \frac{x_i b_{i,2}}{\alpha_1 b_{i,1} + \alpha_2 b_{i,2}} - \Sigma_{i=1}^{n} b_{i,2} \end{bmatrix}.$$

b)

Derive the Fisher Scoring update.

We take the Hessian calculated above. We site the textbook for expected value for a $X \sim \text{Poisson}(\lambda)$ distribution: $E(X) = \lambda$. We also point out that the expected value of a sum is equal to the sum of expected values, or $\Sigma_{i=1}^{n} E(X) = E(\Sigma_{i=1}^{n} x)$.

As such, we can write the Fisher Information $I(\alpha) = -\mathrm{E}(\mathrm{l}''(\alpha))$ as:

$$-\begin{bmatrix} \Sigma_{i=1}^n \frac{-\mathrm{E(x_i)b_{i,1}^2}}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})^2} & \Sigma_{i=1}^n \frac{-\mathrm{E(x_i)b_{i,1}b_{i,2}}}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})^2} \\ \Sigma_{i=1}^n \frac{-\mathrm{E(x_i)b_{i,1}b_{i,2}}}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})^2} & \Sigma_{i=1}^n \frac{-\mathrm{E(x_i)b_{i,2}^2}}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})^2} \end{bmatrix} = -\begin{bmatrix} \Sigma_{i=1}^n \frac{-(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})b_{i,1}^2}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})^2} & \Sigma_{i=1}^n \frac{-(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})b_{i,1}b_{i,2}}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})^2} \\ \Sigma_{i=1}^n \frac{-(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})b_{i,1}b_{i,2}}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})^2} & \Sigma_{i=1}^n \frac{-(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})b_{i,2}^2}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})^2} \end{bmatrix}$$

$$= \begin{bmatrix} \Sigma_{i=1}^n \frac{b_{i,1}^2}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})} & \Sigma_{i=1}^n \frac{b_{i,1}b_{i,2}}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})} \\ \Sigma_{i=1}^n \frac{b_{i,1}b_{i,2}}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})} & \Sigma_{i=1}^n \frac{b_{i,2}^2}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})} \end{bmatrix}.$$

We can then write the Fisher Scoring update, $I(\theta)^{-1}l'(\theta)$ as:

$$\begin{bmatrix} \Sigma_{i=1}^n \frac{b_{i,1}^2}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})} & \Sigma_{i=1}^n \frac{b_{i,1}b_{i,2}}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})} \\ \Sigma_{i=1}^n \frac{b_{i,1}b_{i,2}}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})} & \Sigma_{i=1}^n \frac{b_{i,2}^2}{(\alpha_1 b_{i,1}+\alpha_2 b_{i,2})} \end{bmatrix}^{-1} \begin{bmatrix} \Sigma_{i=1}^n \frac{x_i b_{i,1}}{\alpha_1 b_{i,1}+\alpha_2 b_{i,2}} - \Sigma_{i=1}^n b_{i,1} \\ \Sigma_{i=1}^n \frac{x_i b_{i,2}}{\alpha_1 b_{i,1}+\alpha_2 b_{i,2}} - \Sigma_{i=1}^n b_{i,2} \end{bmatrix}$$

c)

We implement Newton's Method.

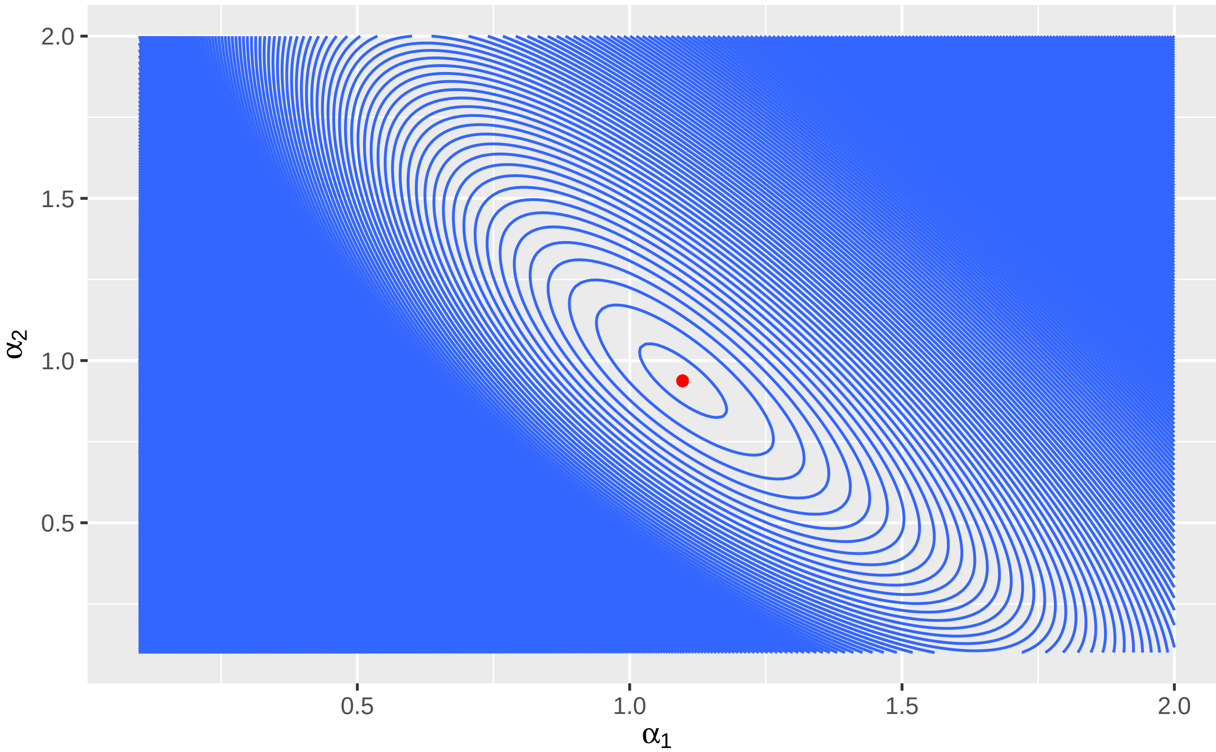
```
oil <- read.table("../datasets/oilspills.dat", header=TRUE)

fprime <- function(alpha, b, x){
  return(c(sum(x * b[, 1] / (b %*% alpha)) - sum(b[, 1]),
           sum(x * b[, 2] /  (b %*% alpha)) - sum(b[, 2])))
}

f2prime <- function(alpha, b, x){
  return(-1 * matrix(c(
    sum(x * b[, 1]^2 / (b %*% alpha)^2),
    sum(x * apply(b, 1, prod) / (b %*% alpha)^2),
    sum(x * apply(b, 1, prod) / (b %*% alpha)^2),
    sum(x * b[, 2]^2 / (b %*% alpha)^2)
  ), ncol=2))
}


newtons_method <- function(fprime, f2prime, alpha0, b, x, max_iter=10000, tol=.001){
  alpha_t <- alpha0

  # Iterate through
  for (n in 1:max_iter){
    # Set stopping conditions
    if(sum((alpha0 - alpha_t)^2) < tol & n > 1){break}

    alpha0 <- alpha_t

    # Get the Newton update
    alpha_t <- alpha0 - solve(f2prime(alpha0, b, x)) %*% fprime(alpha0, b, x)
  }

  return(c(alpha_t=alpha_t, n=n))
}
```

```r
# We call the function on our dataset
alpha0 <- c(1, 1)
b <- as.matrix(oil[, c('importexport', 'domestic')])
x <- as.matrix(oil[, c('spills')])

solution <- newtons_method(fprime, f2prime, c(1, 1), b, x, tol=.00001)
```

The solution is given as $\alpha = [1.097, .938]$, converging in 4 iterations. Below, the contour plot for the likelihood function is shown, with the red dot labelling the given solution. We see that the algorithm appears to have converged on the solution.

```r
log_likelihood <- function(alpha, b, x){
  return(sum(x * log(b %*% alpha)) - sum(alpha[1] * b[, 1])
         - sum(alpha[2] * b[, 2]) - sum(log(factorial(x))))
}

# we construct agrid of the likelihood function to plot the contours
a1 <- seq(0.1, 2, .01)
a2 <- seq(0.1, 2, .01)
alpha_space <- as.matrix(expand.grid(a1, a2)) # Create cartesian product

# Find the likelihod for all pairs
results <- data.frame(cbind(
  alpha_space, apply(
    alpha_space, 1,
      function(alpha) log_likelihood(alpha, b=b, x=x))))

# Add column names
colnames(results) <- c("alpha1", "alpha2", "likelihood")

# Plot the contours with the solution in red
ggplot(results) +
  geom_contour(aes(x=alpha1, y=alpha2, z=likelihood), bins=1000) +
  geom_point(aes(x=solution[1], y=solution[2]), colour="red") +
  xlab(TeX("$\\alpha_1$")) + ylab(TeX("$\\alpha_2$")) +
  ggtitle("Contour Plot of the Likelihood Function") +
  labs(caption="Note: The solution using the Newton-Raphson method is shown in red.")
```

Contour Plot of the Likelihood Function



Note: The solution using the Newton−Raphson method is shown in red.

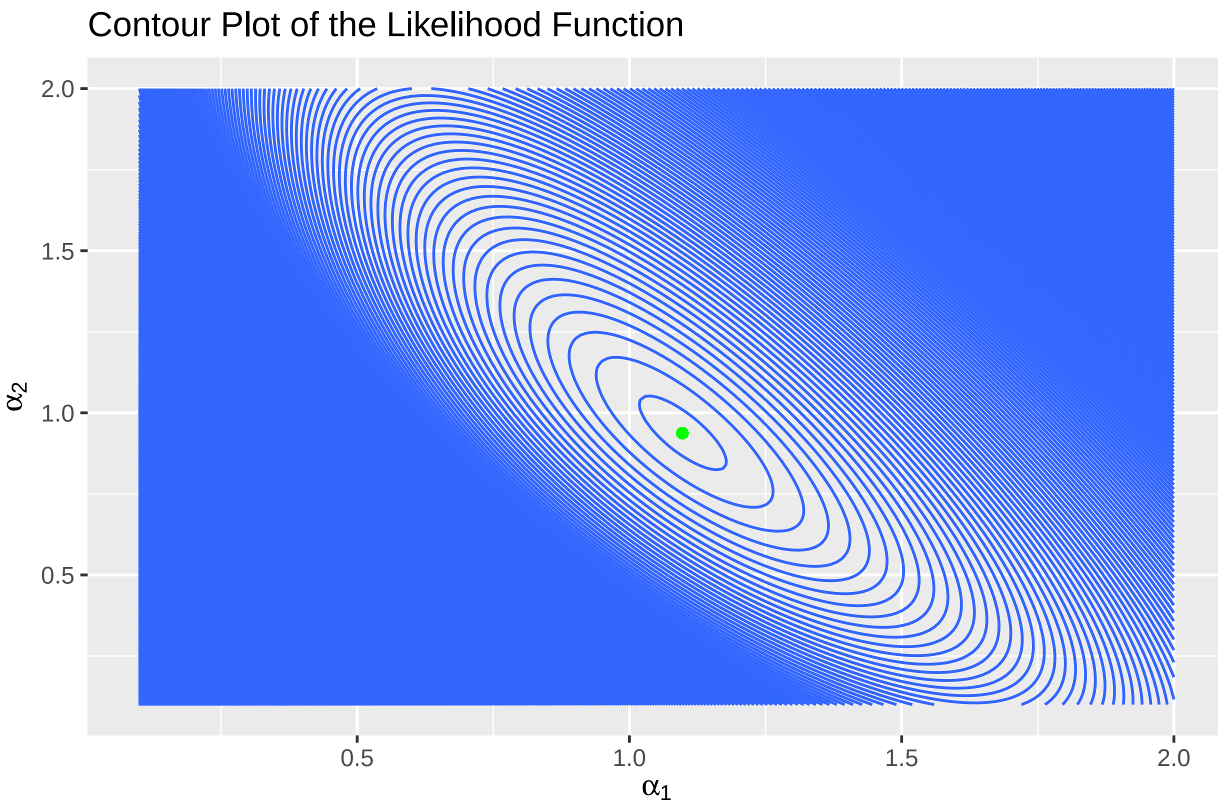
Next, we implement the Fisher Scoring algorithm.

```
# We implement the Fisher scoring update method

I <- function(alpha, b, x){
  return(matrix(c(
    sum(b[, 1]^2 / (b %*% alpha)),
    sum(apply(b, 1, prod) / (b %*% alpha)),
    sum(apply(b, 1, prod) / (b %*% alpha)),
    sum(b[, 2]^2 / (b %*% alpha))
  ), ncol=2))
}

fisher_scoring <- function(I, fprime, alpha0, b, x, max_iter=10000, tol=.001){
  alpha_t <- alpha0

  # Iterate through
  for (n in 1:max_iter){
    # Set stopping conditions
    if(sum((alpha0 - alpha_t)^2) < tol & n > 1){break}

    alpha0 <- alpha_t

    # Get the Fisher update
    alpha_t <- alpha0 + solve(I(alpha0, b, x)) %*% fprime(alpha0, b, x)
  }
```

```r
  return(c(alpha_t=alpha_t, n=n))
}

# We call the function on our dataset
alpha0 <- c(1, 1)
b <- as.matrix(oil[, c('importexport', 'domestic')])
x <- as.matrix(oil[, c('spills')])

solution <- fisher_scoring(I=I, fprime=fprime, alpha0=c(1, 1), b=b, x=x, tol=.00001)
```

The solution is given as $\alpha = [1.097, .938]$, converging in 6 iterations. Below, the contour plot for the likelihood function is shown, with the green dot labelling the given solution. We see that the algorithm appears to have converged on the solution. Note that this is the same solution seen above with Newton's Algorithm. This is as expected, as the two techniques are quite similar.

```r
log_likelihood <- function(alpha, b, x){
  return(sum(x * log(b %*% alpha)) - sum(alpha[1] * b[, 1])
         - sum(alpha[2] * b[, 2]) - sum(log(factorial(x))))
}

# we construct agrid of the likelihood function to plot the contours
a1 <- seq(0.1, 2, .01)
a2 <- seq(0.1, 2, .01)
alpha_space <- as.matrix(expand.grid(a1, a2)) # Create cartesian product

# Find the likelihod for all pairs
results <- data.frame(cbind(
  alpha_space, apply(
    alpha_space, 1,
      function(alpha) log_likelihood(alpha, b=b, x=x))))

# Add column names
colnames(results) <- c("alpha1", "alpha2", "likelihood")

# Plot the contours with the solution in red
ggplot(results) +
  geom_contour(aes(x=alpha1, y=alpha2, z=likelihood), bins=1000) +
  geom_point(aes(x=solution[1], y=solution[2]), colour="green") +
  xlab(TeX("$\\alpha_1$")) + ylab(TeX("$\\alpha_2$")) +
  ggtitle("Contour Plot of the Likelihood Function") +
  labs(caption="Note: The solution using Fisher Scoring is shown in green.")
```



Note: The solution using Fisher Scoring is shown in green.