Data Analysis Homework 3

Jimmy Hickey

10/21/2020

1

a

Notice that

$$\frac{\partial \mu}{\partial \alpha} = \frac{\partial}{\partial \alpha} \alpha^T \begin{bmatrix} 1 \\ \eta \\ \eta^2 \end{bmatrix} = \begin{bmatrix} 1 \\ \eta \\ \eta^2 \end{bmatrix}.$$

Further, the first term of our estimating equation is constant in α so take

$$A_{\eta_j,i} = \frac{\mathcal{C}_{d_{\eta_j},i}}{\prod_{k=2}^{K}[\pi_{\eta_j,k}(\overline{X}_{ki}, \hat{\gamma}_k)]\pi_{\eta_j,1}(X_1; \hat{\gamma}_1)}$$

Then multiplying our derivative vector gives us the estimating equations

$$\begin{aligned} \sum_{i=1}^{n}\sum_{j=1}^{m} 1 \cdot \left[A_{\eta_j,i}(Y_1 - \alpha_1 - \alpha_2\eta_j - \alpha_3\eta_j^2)\right] \sum_{i=1}^{n}\sum_{j=1}^{m} 1 \cdot \left[A_{\eta_j,i}(Y_1 - \alpha_1 - \alpha_2\eta_j - \alpha_3\eta_j^2)\right] &= 0 \\ &= 0 \\ \sum_{i=1}^{n}\sum_{j=1}^{m} \eta \cdot \left[A_{\eta_j,i}(Y_1 - \alpha_1 - \alpha_2\eta_j - \alpha_3\eta_j^2)\right] &= 0 \\ \sum_{i=1}^{n}\sum_{j=1}^{m} \eta^2 \cdot \left[A_{\eta_j,i}(Y_1 - \alpha_1 - \alpha_2\eta_j - \alpha_3\eta_j^2)\right] &= 0 \end{aligned}$$

We will take our range to be $\eta \in (50, 400)$. With a step size of 10, this gives $m = 36$.

```
ldl = read.table("LDL.dat.txt", header=FALSE)
# remove ID column
ldl = ldl[,-1]
names(ldl) = c("L1", "A1", "L2", "S2", "A2", "L3",
"S3", "A3", "L4", "S4", "A4", "Y", "S5")



logistic_func = function(x){
```

```
  return( exp(x) / (1 + exp(x)) )
}

# Propensity model from equation 7 of homework
calc_gamma = function(data){
  out = matrix(0, nrow=4, ncol=3)

  gamma1_mod = glm(A1 ~ L1, data, family = "binomial")
  # add extra 0 because other terms has an S factor
  out[1,] = c(gamma1_mod$coefficients, 0)

  gamma2_mod = glm(A2 ~ L2 + S2, data, family = "binomial")
  out[2,] = gamma2_mod$coefficients

  gamma3_mod = glm(A3 ~ L3 + S3, data, family = "binomial")
  out[3,] = gamma3_mod$coefficients

  gamma4_mod = glm(A4 ~ L4 + S4, data, family = "binomial")
  out[4,] = gamma4_mod$coefficients

  return(out)
}


# Cd vector for equation 5.27 on slide 304
calc_cd = function(data, regime, K){
  n = dim(data)[1]


  # need for AIPW
  if(K==0){
    return(rep(1, n))
  }


  L = cbind(data$L1, data$L2, data$L3, data$L4, data$Y)
  # again need 0s because there are no side effects at the beginning
  S = cbind(rep(0, n), data$S2, data$S3, data$S4, data$S5)
  A = cbind(data$A1, data$A2, data$A3, data$A4)


  cd_vec = rep(1, n)

  for(i in 1:n){

    for(k in 1:K){
        decision = regime(L[i,k], S[i,k], A[i,k], k)
        cd_vec[i] = cd_vec[i] * ( A[i,k] == decision )
    }
  }
  return(cd_vec)
}
```

```r
# calculate the product of propensities used in the denominator
propen_denom = function(data, regime, K){
    n = dim(data)[1]

  # need for AIPW
  if(K==0){
    return(rep(1, n))
  }

  L = cbind(data$L1, data$L2, data$L3, data$L4, data$Y)
  # again need 0s because there are no side effects at the beginning
  S = cbind(rep(0, n), data$S2, data$S3, data$S4, data$S5)
  A = cbind(data$A1, data$A2, data$A3, data$A4)

  gamma = calc_gamma(data)


  # initialize vector of length n
  prod = rep(1, n)

  for(i in 1:n){

    for(k in 1:K){
      val = gamma[k, 1] + gamma[k, 2] * L[i,k] + gamma[k, 3] * S[i,k]
      p = logistic_func(val)
      dk = regime(L[i,], S[i,], A[i,], k)

      pi_k = p * dk + (1-p)*(1-dk)

      prod[i] = prod[i] * pi_k
    }
  }

  return(prod)
}


MSM = function(data, K, etas){
  m = length(etas)
  n = dim(data)[1]

  # First we will build the A matrix
  A_mat = matrix(NA, nrow = n, ncol = m)

  for(j in 1:m){
    eta_j = etas[j]

    # define new regime for each eta value
    regime_eta = function(L, S, A, dk){
      return(S == 0 && L > eta_j)
    }

    cd_j = calc_cd(data, regime_eta, K)
```

```
    propen = propen_denom(data, regime_eta, K)

    A_mat[1:n, j] = cd_j / propen
  }

  # vector of left parameters that is constant in alpha
  const_vec = c(
    data$Y %*% A_mat %*% rep(1, m),
    data$Y %*% A_mat %*% etas,
    data$Y %*% A_mat %*% (etas^2)
  )


  # matrix of alpha coefficients for 3 equations we need to solve
  alpha_mat = matrix(c(
    rep(1, n) %*% A_mat %*% rep(1, m),
      rep(1, n) %*% A_mat %*% etas,
      rep(1, n) %*% A_mat %*% (etas^2),
    rep(1, n) %*% A_mat %*% etas,
      rep(1, n) %*% A_mat %*% (etas^2),
      rep(1, n) %*% A_mat %*% (etas^3),
    rep(1, n) %*% A_mat %*% (etas^2),
      rep(1, n) %*% A_mat %*% (etas^3),
      rep(1, n) %*% A_mat %*% (etas^4)
  ), nrow=3, ncol=3, byrow = TRUE)


  return(solve(alpha_mat) %*% const_vec)
}


etas = seq(50, 200, length.out=50)
K = 4
MSM(ldl, K, etas)
```

```
##              [,1]
## [1,] 117.85292039
## [2,]  -0.39425908
## [3,]   0.00276527
```

b

```
MSM_value = function(data, K, eta_vec, eta){
  fit = MSM_fit(data, K, eta_vec)
  return(fit[1] + fit[2] * eta + fit[3] * eta^2)
}


MSM_bootstrap = function(data, K, eta_vec, eta, rep){
  data_val = MSM_value(data, K, eta_vec, eta)
```

```
  boot_val = rep(NA, rep)

  for(i in 1:rep){
    boot_data = data[sample( dim(data)[1], replace = TRUE ), ]
    boot_val[i] = MSM_value(boot_data, K, eta_vec, eta)
  }

  se = sd(boot_val)
  return(c(data_val, se))
}

# See results below
# eta_vec = seq(90, 200, 10)
# for(i in 1:length(eta_vec)){
#   boot_est = MSM_bootstrap(ldl, K=4, eta_vec, eta_vec[i], 5)
#
#   cat("=======\nFor eta=", eta_vec[i], " the value is ", boot_est[1],
#       " with a standard deviation of ", boot_est[2])
# }
```

c

The bootstrap takes a long time to run, so here is the code from a previous run and the output.

```
# eta_vec = seq(90, 200, 10)
# for(i in 1:length(eta_vec)){
#   boot_est = MSM_bootstrap(ldl, K=4, eta_vec, eta_vec[i], 5)
#
#   cat("=======\nFor eta=", eta_vec[i], " the value is ", boot_est[1],
#       " with a standard deviation of ", boot_est[2])
# }

# =======
# For eta= 90  the value is  106.1375  with a standard deviation of  1.046301=======
# For eta= 100  the value is  107.5367  with a standard deviation of  1.671561=======
# For eta= 110  the value is  105.5903  with a standard deviation of  1.610017=======
# For eta= 120  the value is  100.2983  with a standard deviation of  11.07936=======
# For eta= 130  the value is  91.66065  with a standard deviation of  27.16922=======
# For eta= 140  the value is  79.67742  with a standard deviation of  71.67292=======
# For eta= 150  the value is  64.34858  with a standard deviation of  88.07134=======
# For eta= 160  the value is  45.67413  with a standard deviation of  49.85914=======
# For eta= 170  the value is  23.65407  with a standard deviation of  101.2761=======
# For eta= 180  the value is  -1.711588  with a standard deviation of  181.0727=======
# For eta= 190  the value is  -30.42286  with a standard deviation of  80.02617=======
# For eta= 200  the value is  -62.47973  with a standard deviation of  291.0023
```

Clearly by those standard deviations (and negative values!) that something needs to be address and perhaps 5 reptitions of the bootstrap is not enough. These results are very different than the results from homework 2, where we saw the optimal η around 150 (and standard errors less than 15!).