Homeworrk 4

Christian Tillich

October 16, 2017

Setup 1

```
data(Fish, package = 'mnlogit')
fm <- formula(mode ~ price | income | catch)
CM <- mnlogit(fm, Fish, "alt")
summary(CM)
```

```
Call:
mnlogit(formula = fm, data = Fish, choiceVar = "alt")

Frequencies of alternatives in input data:
  beach    boat charter    pier
0.11337 0.35364 0.38240 0.15059

Number of observations in data = 1182
Number of alternatives = 4
Intercept turned: ON
Number of parameters in model = 11
  # individual specific variables = 2
  # choice specific coeff variables = 1
  # individual independent variables = 1

-------------------------------------------------------------
Maximum likelihood estimation using the Newton-Raphson method
-------------------------------------------------------------
  Number of iterations: 7
  Number of linesearch iterations: 10
At termination:
  Gradient norm = 2.09e-06
  Diff between last 2 loglik values = 0
  Stopping reason: Succesive loglik difference < ftol (1e-06).
Total estimation time (sec): 0.11
Time for Hessian calculations (sec): 0.03 using 1 processors.

Coefficients :
                       Estimate   Std.Error  t-value  Pr(>|t|)
(Intercept):boat     8.4184e-01  2.9996e-01   2.8065 0.0050080 **
(Intercept):charter  2.1549e+00  2.9746e-01   7.2443 4.348e-13 ***
(Intercept):pier     1.0430e+00  2.9535e-01   3.5315 0.0004132 ***
income:boat          5.5428e-05  5.2130e-05   1.0633 0.2876611
income:charter      -7.2337e-05  5.2557e-05  -1.3764 0.1687088
income:pier         -1.3550e-04  5.1172e-05  -2.6480 0.0080977 **
catch:beach          3.1177e+00  7.1305e-01   4.3724 1.229e-05 ***
catch:boat           2.5425e+00  5.2274e-01   4.8638 1.152e-06 ***
catch:charter        7.5949e-01  1.5420e-01   4.9254 8.417e-07 ***
catch:pier           2.8512e+00  7.7464e-01   3.6807 0.0002326 ***
price               -2.5281e-02  1.7551e-03 -14.4046 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-Likelihood: -1199.1, df = 11
AIC:  2420.3
```

```
# columns y, income, price1, price2, price3, price4, catch1, catch2, catch3, catch4
# modes 1,2,3,4: beach, boat, charter, pier
df <- read.table("../data/fish.txt",header=T)
code <- "
  data {K <- 4}
  model {
    for (i in 1:1182) {
      y[i] ~ dcat(p[i,1:4])
      for (k in 1:K) {p[i,k] <- phi[i,k]/sum(phi[i,])}
```

```
      log(phi[i,1]) <- gam*prices[1,i] + delta[1]*catchs[1,i]
      for (k in 2:K) {
        log(phi[i,k]) <- alph[k] + gam*prices[k,i] + beta[k]*incomes[i] +
          delta[k]*catchs[k,i]
      }
      prices[1,i] <- price1[i]/100
      prices[2,i] <- price2[i]/100
      prices[3,i] <- price3[i]/100
      prices[4,i] <- price4[i]/100
      catchs[1,i] <- catch1[i]
      catchs[2,i] <- catch2[i]
      catchs[3,i] <- catch3[i]
      catchs[4,i] <- catch4[i]
      incomes[i] <- income[i]/1000


    }

  # priors
  alph[1] <- 0
  for (j in 2:K) {alph[j] ~ dnorm(0,0.001)}
  beta[1] <- 0
  for (j in 2:K) {beta[j] ~ dnorm(0,0.001)}
  for (j in 1:K) {delta[j] ~ dnorm(0,0.001)}
  gam ~ dnorm(0,0.001)
}" %>% strsplit('\n') %>% unlist

INI <- list(
   list( alph=c(NA,0,0,0), beta=c(NA,0,0,0), delta=c(0,0,0,0), gam=0)
  ,list(alph=c(NA,1,1,1), beta=c(NA,0,0,0), delta=c(1,1,1,1),gam=-2)
)

build.model(
  code, df, INI,
  params = c('alph','beta','delta','gam'),
  n.chains=2, n.adapt = 500, n.iter=5000) %>%
  jagsresults(c('alph','beta','delta','gam')) %>%
  as.data.frame %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
alph[1]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
alph[2]	0.8222045	0.2966271	0.2446822	0.6160411	0.8298598	1.0336202	1.3828310
alph[3]	2.1421413	0.2906826	1.5926020	1.9391297	2.1355737	2.3447801	2.7183574
alph[4]	1.0454487	0.2811843	0.4978146	0.8424119	1.0468941	1.2459935	1.5789357
beta[1]	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
beta[2]	0.0600922	0.0525752	-0.0368621	0.0230955	0.0591941	0.0949741	0.1674928
beta[3]	-0.0676567	0.0521964	-0.1676324	-0.1033833	-0.0671444	-0.0339024	0.0374832
beta[4]	-0.1362005	0.0509912	-0.2347758	-0.1717629	-0.1362673	-0.1013236	-0.0371095
delta[1]	3.0853029	0.7097213	1.7071046	2.6108726	3.0890999	3.5586004	4.4909404
delta[2]	2.5773773	0.5267124	1.5572572	2.2272126	2.5667709	2.9105955	3.6808201
delta[3]	0.7704606	0.1547233	0.4718385	0.6679736	0.7671460	0.8689160	1.0860102
delta[4]	2.8133927	0.8001414	1.2645903	2.2650472	2.8190602	3.3439866	4.3732003

	mean	sd	2.5%	25%	50%	75%	97.5%
gam	-2.5578738	0.1775671	-2.9179521	-2.6782558	-2.5560780	-2.4349833	-2.2172054

Question 1

```
code %>%
  append('      for (k in 1:K) {r[i,k] <- equals(y[i], k)}', 6) %>%
  append('      LL[i] <- sum(r[i,]*log(p[i,]))', 7) %>%
  append('      dev[i] <- -2*LL[i]', 8) %>%
  {.[27] <- " tot.dev <- sum(dev[])" ; .} %>%
  build.model(df, INI, c('dev','tot.dev'),n.chains=2, n.adapt = 500, n.iter=2500) %>%
  jagsresults(c('dev', 'tot.dev')) %>%
  as.data.frame %>%
  {.[order(-.$mean), ,drop=F]} %>%
  head(3) %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
tot.dev	2409.22574	4.591796	2402.01346	2405.89275	2408.62972	2411.96812	2419.41190
dev[250]	15.37267	1.173242	13.13619	14.57377	15.32397	16.12672	17.79192
dev[344]	15.00979	1.026603	13.11040	14.29596	14.96793	15.67952	17.09855

Observation 250 and 344 have the largest deviance.

Question 2

```
Fish$pc <- Fish$price*Fish$catch
fm <- formula(mode ~ price | income | catch + pc)
CM <- mnlogit(fm, Fish, "alt")
summary(CM)
```

```
Call:
mnlogit(formula = fm, data = Fish, choiceVar = "alt")

Frequencies of alternatives in input data:
  beach    boat charter    pier
0.11337 0.35364 0.38240 0.15059

Number of observations in data = 1182
Number of alternatives = 4
Intercept turned: ON
Number of parameters in model = 15
  # individual specific variables = 2
  # choice specific coeff variables = 2
  # individual independent variables = 1

-------------------------------------------------------------
Maximum likelihood estimation using the Newton-Raphson method
-------------------------------------------------------------
  Number of iterations: 8
  Number of linesearch iterations: 10
At termination:
  Gradient norm = 8.7e-07
  Diff between last 2 loglik values = 0
  Stopping reason: Succesive loglik difference < ftol (1e-06).
Total estimation time (sec): 0.05
Time for Hessian calculations (sec): 0 using 1 processors.
```

```
Coefficients :
                      Estimate   Std.Error t-value  Pr(>|t|)    
(Intercept):boat     1.1980e+00  3.2650e-01  3.6693 0.0002433 ***
(Intercept):charter  2.5900e+00  3.1967e-01  8.1021 4.441e-16 ***
(Intercept):pier     1.1013e+00  3.0106e-01  3.6580 0.0002542 ***
income:boat         -7.4931e-08  5.6281e-05 -0.0013 0.9989377    
income:charter      -1.5172e-04  5.6502e-05 -2.6852 0.0072493 ** 
income:pier         -1.3313e-04  5.1832e-05 -2.5686 0.0102121 *  
catch:beach          5.3504e+00  1.1117e+00  4.8130 1.487e-06 ***
catch:boat           1.7382e+00  5.8942e-01  2.9490 0.0031880 ** 
catch:charter        1.7330e-01  2.0745e-01  0.8354 0.4034936    
catch:pier           5.5312e+00  1.2567e+00  4.4014 1.075e-05 ***
pc:beach            -2.0938e-02  1.0697e-02 -1.9575 0.0502927 .  
pc:boat              1.0540e-02  8.8961e-03  1.1848 0.2361083    
pc:charter           7.9104e-03  2.0564e-03  3.8468 0.0001197 ***
pc:pier             -3.1691e-02  1.4546e-02 -2.1787 0.0293560 *  
price               -2.5732e-02  2.5827e-03 -9.9632 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-Likelihood: -1181, df = 15
AIC:  2392
```

The log likelihood, with price*catch interactions, is -1811. This represents an increase of (b) over 15 from the original -1199

Question 3

```
code %>%
  append('      for (k in 1:K) {r[i,k] <- equals(y[i], k)}', 6) %>%
  append('      LL[i] <- sum(r[i,]*log(p[i,]))', 7) %>%
  append('      dev[i] <- -2*LL[i]', 8) %>%
  {.[27] <- " tot.dev <- sum(dev[])" ; .} %>%
  {.[13] <- " delta[k]*catchs[k,i] + eps[k]*catchs[k,i]*prices[k,i]" ; .} %>%
  {.[10] <- "log(phi[i,1]) <- gam*prices[1,i] +delta[1]*catchs[1,i] + eps[1]*catchs[1,i]*prices[1,i]"
  append("  for (j in 1:K) {eps[j] ~ dnorm(0,0.001)}", 31) %>%
  build.model(df, INI, 'tot.dev',n.chains=2, n.adapt = 500, n.iter=2500) %>%
  jagsresults('tot.dev') %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
tot.dev	2377.276	5.578866	2368.257	2373.33	2376.654	2380.546	2390.038

The posterior mean deviance is 2377. This represents a fall of ~32 from the deviance when we exclude the `choice*price` interaction. So (b) less than 40.

Setup 2

```
CM <- clm(rating ~ contact + temp, data = wine)

# columns y,temp, contact,judge
df <- read.table("../data/wine.txt",header=T)
# JAGS
code <- "
  data { K <- 5; KM <- 4}
  model{
    for(i in 1:72){
      eta[i] <- beta[1]*contact[i] + beta[2]*temp[i]
      logit(Q[i,1]) <- theta[1]-eta[i]
      p[i,1] <- Q[i,1]
      for(j in 2:KM) {
        logit(Q[i,j]) <- theta[j]-eta[i]
        p[i,j] <- Q[i,j] - Q[i,j-1]
      }
      p[i,K] <- 1 - Q[i,KM]
      y[i] ~ dcat(p[i,1:K])
      yrep[i] ~ dcat(p[i,1:K])
      match[i] <- equals(y[i],yrep[i])
    }
    Classif.acc <- mean(match[])

  # prior for cut-points
  for(r in 1:4){ theta0[r] ~dnorm(0,1.0E-3)}
  theta <- sort(theta0)
  for (j in 1:2){beta[j] ~ dnorm(0,1.0E-3)}}
" %>% strsplit('\n') %>% unlist

INI <- list(
   list(theta0=c(-0.6,0,0.6,1.2),beta=c(0,0))
  ,list(theta0=c(-0.5,0,0.5,1),beta=c(0.5,0.5))
)

build.model(code, df[,-4], INI, c("beta","theta"), n.chains=2, n.adapt=500, n.iter=5000) %>%
  jagsresults(c("beta","theta")) %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
beta[1]	1.591582	0.4813824	0.6623298	1.257978	1.591984	1.910229	2.527713
beta[2]	2.614519	0.5480654	1.5283228	2.254669	2.613380	2.981876	3.694723
theta[1]	-1.443650	0.5413944	-2.5904960	-1.793349	-1.417552	-1.055408	-0.476060
theta[2]	1.300296	0.4477275	0.4302601	0.999967	1.296048	1.599140	2.181827
theta[3]	3.612151	0.6124652	2.4491463	3.196893	3.593761	4.022637	4.818332
theta[4]	5.256168	0.7436549	3.8430034	4.742644	5.245876	5.766229	6.746988

Question 4

```
code %>%
  build.model(df[,-4], INI, "Classif.acc", n.chains=2, n.adapt=500, n.iter=5000) %>%
  jagsresults("Classif.acc") %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
Classif.acc	0.3426181	0.0538146	0.2361111	0.3055556	0.3472222	0.375	0.4444444

Accuracy rate is ~34%, so (a) above 25%

Question 5

```
code %>%
  {.[5] <- 'eta[i] <- beta[1]*contact[i] + beta[2]*temp[i] + omega[judge[i]]';.} %>%
  append('for (j in 1:9){omega[j] ~ dnorm(0,1.0E-3)}',21) %>%
  build.model(df, INI, "Classif.acc", n.chains=2, n.adapt=500, n.iter=5000) %>%
  jagsresults("Classif.acc") %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
Classif.acc	0.4275431	0.0549144	0.3194444	0.3888889	0.4305556	0.4583333	0.5416667

The classification accuracy is now ~43%, so (a) above 30%

Question 6

```
INI <- list(
  list(theta0=c(-0.6,0,0.6,1.2),beta=c(0,0,0))
 ,list(theta0=c(-0.5,0,0.5,1),beta=c(0.5,0.5,0.5))
)

code %>%
  {.[5] <- 'eta[i] <- beta[1]*contact[i] + beta[2]*temp[i] + beta[3]*temp[i]*contact[i] + omega[judge[i
  append('for (j in 1:9){omega[j] ~ dnorm(0,1.0E-3)}',21) %>%
  {.[23] <- gsub('(j in 1:2)','j in 1:3',.[23]); .} %>%
  build.model(df, INI, "Classif.acc", n.chains=2, n.adapt=500, n.iter=5000) %>%
  jagsresults("Classif.acc") %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
Classif.acc	0.4264819	0.0547395	0.3194444	0.3888889	0.4305556	0.4583333	0.5277778

The classification accuracy is still ~43%, so (a) above 40%.

Setup 3

```r
# columns rating, complaints,learning,advance,privileges,raises,critical
df <- read.table("../data/attitude.txt",header=T)
# conventional least squares
LM  <- lm(rating ~complaints+learning+advance+privileges+raises+critical, data=df)
summary(LM)
```

```

Call:
lm(formula = rating ~ complaints + learning + advance + privileges +
    raises + critical, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-10.9418  -4.3555   0.3158   5.5425  11.5990

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.78708   11.58926   0.931 0.361634
complaints   0.61319    0.16098   3.809 0.000903 ***
learning     0.32033    0.16852   1.901 0.069925 .
advance     -0.21706    0.17821  -1.218 0.235577
privileges  -0.07305    0.13572  -0.538 0.595594
raises       0.08173    0.22148   0.369 0.715480
critical     0.03838    0.14700   0.261 0.796334
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.068 on 23 degrees of freedom
Multiple R-squared:  0.7326,	Adjusted R-squared:  0.6628
F-statistic:  10.5 on 6 and 23 DF,  p-value: 1.24e-05
```

```r
code <-"
  data {r <- 0.5; tau2[1] <- 0.01; tau2[2] <- 10; p <- 6}
  model {
    for (i in 1:30) {
      rating[i] ~ dnorm(mu[i],tau)
      e[i] <- rating[i] - mu[i]
      mu[i] <- beta0 + beta[1]*complaints[i] + beta[2]*learning[i] +
        beta[3]*advance[i] + beta[4]*privileges[i] + beta[5]*raises[i] +
        beta[6]*critical[i]
    }

    #Priors
    for (j in 1:6) {
      beta[j] ~ dnorm(0, 1/tau2[G[j]])
      G[j] <- gam[j] + 1
      gam[j] ~ dbern(r)
    }
    tau ~ dgamma(1,0.001)
    beta0 ~ dnorm(0,0.001)
    M <- 1 + gam[1]*pow(2,p-1) + gam[2]*pow(2,p-2) + gam[3]*pow(2,p-3) +
      gam[4]*pow(2,p-4) + gam[5]*pow(2,p-5) +gam[6]
    for (m in 1:64) {mod[m] <- equals(m,M)}
  }
" %>% strsplit('\n') %>% unlist

INI <- list(
   list(beta=c(0,0,0,0,0,0),tau=1, beta0=0)
  ,list(beta=c(0,0,0,0,0,0),tau=0.1, beta0=10)
)
```

```r
build.model(code, df, INI, c("beta","gam"), n.chains=2,n.adapt=500, n.iter=25000) %>%
  jagsresults(c('beta','gam')) %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
beta[1]	0.7122544	0.1374631	0.4370599	0.6292513	0.7148504	0.8008064	0.9714375
beta[2]	0.1155896	0.1208960	-0.0775305	0.0361907	0.1004981	0.1731910	0.4243565
beta[3]	-0.0423405	0.0929622	-0.2285854	-0.0980773	-0.0396910	0.0184829	0.1272638
beta[4]	-0.0279988	0.0841817	-0.1942751	-0.0823776	-0.0267166	0.0273508	0.1355639
beta[5]	0.0229273	0.1029165	-0.1648808	-0.0407286	0.0210735	0.0827765	0.2104700
beta[6]	0.0114531	0.0879032	-0.1519373	-0.0436740	0.0104084	0.0630167	0.1785219
gam[1]	0.9932600	0.0818212	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
gam[2]	0.1625000	0.3689126	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000
gam[3]	0.0667000	0.2495042	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000
gam[4]	0.0559200	0.2297695	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000
gam[5]	0.0661800	0.2485990	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000
gam[6]	0.0521800	0.2223921	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000

Question 7

Only one predictor - complaints - has a posterior probability of inclusion greater than 95%.

Question 8

```r
cbind(df[1], scale(df[-1])) %>%
  build.model(code, ., INI, c("beta","gam"),n.chains=2,n.adapt=500, n.iter=25000) %>%
  jagsresults(c('beta','gam')) %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
beta[1]	7.1354516	1.5945209	3.8899137	6.0979427	7.1657100	8.2182929	10.1549925
beta[2]	2.4198604	1.8264193	-0.1476005	0.6428987	2.5419747	3.7478817	5.9043462
beta[3]	-0.6555642	1.2223975	-3.7760015	-1.2522078	-0.0783539	0.0363627	0.8694408
beta[4]	-0.0248618	0.8140233	-2.0759414	-0.1104657	-0.0022566	0.0995841	1.9414559
beta[5]	0.6438824	1.3516984	-1.2336262	-0.0494227	0.0670678	1.1545973	4.1832887
beta[6]	0.0586564	0.6790618	-1.4912021	-0.0862602	0.0077328	0.1087288	1.9121710
gam[1]	0.9996400	0.0189704	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
gam[2]	0.7951400	0.4036033	0.0000000	1.0000000	1.0000000	1.0000000	1.0000000
gam[3]	0.4411400	0.4965284	0.0000000	0.0000000	0.0000000	1.0000000	1.0000000
gam[4]	0.3156200	0.4647670	0.0000000	0.0000000	0.0000000	1.0000000	1.0000000
gam[5]	0.4406000	0.4964641	0.0000000	0.0000000	0.0000000	1.0000000	1.0000000
gam[6]	0.2797600	0.4488857	0.0000000	0.0000000	0.0000000	1.0000000	1.0000000

The probability of inclusion increased across the board. However, still only `complaints` had a mean prosterior probability of inclusion greater than 95%.

Question 9

```
cbind(df[1], scale(df[-1])) %>%
  build.model(code, ., INI, c("mod"),n.chains=2,n.adapt=500, n.iter=25000) %>%
  jagsresults(c('mod')) %>%
  as.data.frame %>%
  {.[order(-.$mean), ,drop=F]} %>%
  head(2) %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
mod[49]	0.13202	0.3385159	0	0	0	0	1
mod[57]	0.09664	0.2954699	0	0	0	0	1

Model number 49 is the most probable. This model is represented 1 + X1 + X2.

Question 10

```
df2 <- cbind(df[1], scale(df[-1]))

code %>%
  {.[2] <- gsub('r <- 0.5; ','',.[2]); .} %>%
  append('    r[j] ~ dbeta(1,1)', 15) %>%
  {.[17] <- gsub('(r)','(r[j])',.[17], fixed=T); .} %>%
  build.model(df2, INI, "r",n.chains=2,n.adapt=500, n.iter=25000) %>%
  jagsresults("r") %>%
  kable
```

	mean	sd	2.5%	25%	50%	75%	97.5%
r[1]	0.6666471	0.2347213	0.1603592	0.5002633	0.7059556	0.8657286	0.9874615
r[2]	0.6049264	0.2686357	0.0631284	0.4012632	0.6457301	0.8352766	0.9843286
r[3]	0.4871133	0.2893581	0.0220935	0.2342124	0.4823639	0.7356993	0.9732482
r[4]	0.4369091	0.2807999	0.0187353	0.1924762	0.4083992	0.6660136	0.9598439
r[5]	0.4750572	0.2880225	0.0220371	0.2231734	0.4614410	0.7214490	0.9696898
r[6]	0.4301168	0.2796452	0.0177944	0.1851110	0.3996144	0.6537129	0.9580396

The posterior means for `complaints` and 'privileges' both exceed 0.50.