MAST90104: A First Course in Statistical Learning

Assignment 4, 2023 Solution

1. (15 marks) The data *winequality-red.csv* includes data from the paper *Modeling wine preferences by data mining from physicochemical properties* by P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis (2009). The data consists of 1599 observations of red variants of the Portuguese "Vinho Verd" wine. The variables are:

   - fixed acidity
   - volatile acidity
   - citric acid
   - residual sugar
   - chlorides
   - free sulfur dioxide
   - total sulfur dioxide
   - density
   - pH
   - sulphates
   - alcohol : percent alcohol content of the wine
   - quality : Output variable (score between 0 and 10)

   The quality score in this data set ranges from 3 to 8, but we will recode the levels as *bad*, *average* and *good*:

```
wine$quality = factor(wine$quality)
levels(wine$quality ) <- c("bad","bad","average","average" , "good","good")
```

   (a) Fit a multinomial logit model to predict the wine quality by category, considering all available predictors. Refine the model using stepwise selection. (3pt)

   Solution

```
library(nnet)
model1 = multinom(quality ~ ., data = wine)
model2 = step(model1,trace = 0)
summary(model2)

# Call:
# multinom(formula = quality ~ fixed.acidity + volatile.acidity +
#              citric.acid + residual.sugar + chlorides + total.sulfur.dioxide +
#              density + pH + sulphates + alcohol, data = wine)
#
# Coefficients:
#         (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar  chlorides
# average   -177.1713   -0.26252634        -4.565347  -1.4550010    -0.24141265  -5.434441
# good       125.7384    0.07207673        -6.952064  -0.8516209     0.02213188 -13.805282
#
#         total.sulfur.dioxide   density        pH sulphates   alcohol
# average          0.020732275  196.0641 -4.486351  1.255191 0.4420856
# good             0.006790962 -123.6146 -3.799881  5.050087 1.1345209
#
```

```
# Std. Errors:
#         (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
# average    2.704340     0.1545768        0.8275218    1.201010     0.07916868  3.111854
# good       3.125144     0.1708142        1.1010330    1.427641     0.09566638  4.468744
#
#         total.sulfur.dioxide  density       pH sulphates   alcohol
# average          0.006059025 2.649079 1.393743  1.273329 0.1756397
# good             0.006894894 3.048687 1.597277  1.360258 0.1946937
#
# Residual Deviance: 1296.663
# AIC: 1340.663
```

Stepwise selection on the multinomial chose 10 variables (all except free sulfur dioxide)

(b) Repeat the analysis with an ordinal model, with "good" being the highest level of the output. Comment on any differences. (4pt)

Solution

```
library(MASS)
model_o1= polr(quality ~ ., data = wine)
model_o2 <- step(model_o1,trace = 0)
summary(model_o2)

# Call:
# polr(formula = quality ~ volatile.acidity + chlorides + pH +
#          sulphates + alcohol, data = wine)
#
# Coefficients:
#                    Value Std. Error t value
# volatile.acidity -4.0170     0.4688  -8.569
# chlorides        -6.5630     1.8940  -3.465
# pH               -2.1087     0.5221  -4.039
# sulphates         2.4779     0.4547   5.450
# alcohol           0.8712     0.0758  11.494
#
# Intercepts:
#                Value   Std. Error t value
# poor|average -3.0044  1.7780    -1.6898
# average|good  3.6146  1.7748     2.0366
#
# Residual Deviance: 1377.604
# AIC: 1391.604
```

With the same set of predictors, ordinal model has less parameters than multinomial model. Stepwise selection on the ordinal model only picked 5 predictors.

(c) We have a new observation with these attributes:

```
newobs
# fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
#           7.9              0.4         0.2             1.7       0.1
# free.sulfur.dioxide total.sulfur.dioxide  density   pH  sulphates alcohol
#                  10                   36    0.997  3.3        0.9      10
```

Compute the probabilities that this wine variant is a "bad", "average" and "good" wine, according to the refined multinomial and ordinal models. You should NOT use the function `predict()` for this question. (5pt)

Solution

Let $j = 1, 2, 3$ correspond to the outcome "bad", "average" and "good" respectively.
In the multinomial model,

$$p_j^* = \frac{e^{\eta_j^*}}{\sum_{k=1}^{3} e^{\eta_k^*}}, \quad j = 1, 2, 3,$$

and $\eta_j^* = (\mathbf{x}^*)^T \boldsymbol{\beta}_j$; $\boldsymbol{\beta}_1 = 0$ so $\eta_1^* = 0$

```
xstar = c(1,7.9,0.4,0.2,1.7,0.1,36,0.997,3.3,0.9,10)
model2_coef_average = c(-177.1713, -0.26252634,-4.565347,-1.4550010, -0.24141265,
-5.434441 , 0.020732275, 196.0641, -4.486351,  1.255191, 0.4420856)
model2_coef_good = c( 125.7384, 0.07207673, -6.952064 , -0.8516209, 0.02213188,
-13.805282 ,0.006790962, -123.6146, -3.799881,  5.050087, 1.1345209)
eta = rep(0,3)
eta[2] = (xstar%*%model2_coef_average)
eta[3] = (xstar%*%model2_coef_good)
(probs_multinom = exp(eta)/sum(exp(eta)) )
# [1] 0.008591233 0.899949562 0.091459205
```

In the ordinal model, we have

$$\gamma_j^* = Pr(Y^* \leq j), \quad \log \frac{\gamma_j^*}{1 - \gamma_j^*} = \theta_j - (\mathbf{x}^*)^T \boldsymbol{\beta}.$$

So $p_1^* = \frac{1}{1 + e^{-\gamma_1^*}}$, $p_2^* = \frac{1}{1 + e^{-\gamma_2^*}} - \frac{1}{1 + e^{-\gamma_1^*}}$ and $p_3^* = 1 - p_1^* - p_2^* = 1 - \frac{1}{1 + e^{-\gamma_2^*}}$.
As can be seen below, there is some difference in predicted probabilities between the ordinal and multinomial models.

```
xstar_ord = c(0.4,0.1,3.3,0.9,10)
logitcumprob = rep(1,2)
logitcumprob[1] = (model_o2$zeta[1]  - xstar_ord%*%model_o2$coefficients)
logitcumprob[2] = (model_o2$zeta[2] - xstar_ord%*%model_o2$coefficients)

probs_ordinal = rep(NA,3)
probs_ordinal[1] = 1/(1+ exp(-logitcumprob[1]))
probs_ordinal[2] = 1/(1+ exp(-logitcumprob[2])) - 1/(1+ exp(-logitcumprob[1]))
probs_ordinal[3] = 1- 1/(1+ exp(-logitcumprob[2]))
probs_ordinal
# [1] 0.00879436 0.86042844 0.13077720
```

(d) Under the ordinal model, what is the odds ratio of being classified as "bad" or "average" of a wine variant with chlorides level 0.08 compared to a variant with chlorides level 0.2, given that the other attributes of the two variants are the same? (3pt)
Solution From week 10's workshop, we have the odds ratio is

$$\frac{Pr(Y \leq 2|\mathbf{x}_A)}{Pr(Y \leq 2|\mathbf{x}_B)} = \exp(-(\mathbf{x}_A - \mathbf{x}_B)^T \boldsymbol{\beta}),$$

where the elements of $\mathbf{x}_A$ and $\mathbf{x}_B$ are the same except for chlorides. So the odds ratio is $\exp(-(0.08 - 0.2)\beta_{\text{chlorides}})$

```
exp(-model_o2$coefficients[2]*(0.08-0.2))
# 0.4549514
```

2. (15 marks) Consider the following simple regression model

$$y_i = \beta x_i + \epsilon_i, \quad \epsilon_i \sim N(0, 1),$$

where $i = 1, \ldots, n$. The file *simplereg.csv* contains data of 30 observations from this model.

(a) Write the likelihood for this model in terms of β. (3pt)

Solution:

$\mathcal{L}(\beta) = \prod_{i=1}^{n} f(y_i|\beta, x_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y_i - \beta x_i)^2\right) = (2\pi)^{-n/2} \exp(-\frac{1}{2}\sum_i (y_i - \beta x_i)^2)$

(b) The prior distribution for β is $N(0, 100)$. Derive the posterior distribution of β given the data. (4pt)

Solution:

$$\begin{aligned} p(\beta|y_1, \ldots, y_n) &\propto p(\beta)\mathcal{L}(\beta) \\ &= \frac{1}{\sqrt{200\pi}} \exp(-\frac{\beta^2}{200})(2\pi)^{-n/2} \exp\left(-\frac{1}{2}\sum_i (y_i - \beta x_i)^2\right) \\ &\propto \exp\left[-\frac{1}{2}\left(\frac{\beta^2}{100} + \sum_i (y_i^2 - 2\beta x_i y_i + \beta^2 x_i^2)\right)\right] \\ &\propto N(A, B) \end{aligned}$$

where $B = (1/100 + \sum_i x_i^2)^{-1}$ and $A = B\sum_i x_i y_i$.

(c) Write a Metropolis-Hastings algorithm to sample from the posterior of β. The proposal $Q(\beta'|\beta)$ is $N(\beta, 0.5^2)$, where β is the current value. (3pt)

Solution:

- initialize a value $\beta^{(0)}$, set $t = 0$. Number of iterations M, number of burn-in iterations T
- Repeat for M times:
  - $\beta = \beta^{(t)}$
  - $t = t + 1$
  - sample $\beta' \sim N(\beta, 0.5^2)$
  - sample $u \sim U(0, 1)$. If $u < A(\beta'|\beta)$, then $\beta^{(t)} = \beta'$; else $\beta^{(t)} = \beta$.

$$\begin{aligned} A(\beta'|\beta) &= \min\left(1, \frac{p(\beta'|\mathbf{y})Q(\beta|\beta')}{p(\beta|\mathbf{y})Q(\beta'|\beta)}\right) \\ &= \min\left(1, \frac{p(\beta'|\mathbf{y})}{p(\beta|\mathbf{y})}\right) \\ &= \min\left(1, \frac{\mathcal{L}(\beta')p(\beta')}{\mathcal{L}(\beta)p(\beta)}\right) \end{aligned}$$

Note that for $Q(\beta'|\beta)$ being $N(\beta, s^2)$, then $Q(\beta'|\beta) = Q(\beta|\beta')$

- Take samples from the (T+1)-th iteration (throw out data generated in the burn-in stage).

(d) Implement the algorithm in R and run for 20000 iterations. Discard the first 5000 iterations as burn-in. Compare the distribution of the resulting samples with your answer in part (b). (5pt)

Solution:

```
set.seed(123)
s = 0.5
M = 20000
J = 5000
betasamp = rep(NA,M)
acckeep = rep(NA,M)
betacurrent = 1.2
logpostcurrent = sum(dnorm(y,betacurrent*x,1,log = TRUE))
      + dnorm(betacurrent,0,sqrt(betasig2),log = TRUE)
for(i in 1:M){
betanew = rnorm(1,betacurrent,s)
```

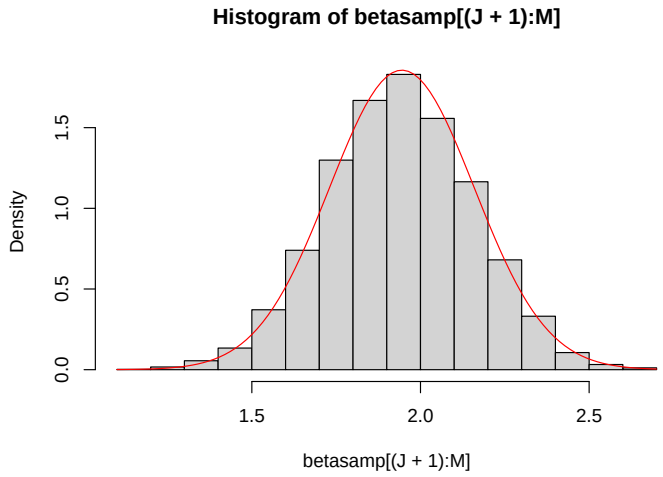

Figure 1: Result of MCMC vs the true posterior

```
logpostnew = sum(dnorm(y,betanew*x,1,log = TRUE))
    + dnorm(betanew,0,sqrt(betasig2),log = TRUE)
la = min(0,logpostnew - logpostcurrent)
lu = log(runif(1))
if(lu < la){
betacurrent= betanew
logpostcurrent = logpostnew
}
betasamp[i] = betacurrent
acckeep[i] =exp(la)
}
```

Using the result in part (b), we can find that $p(\beta|\mathbf{y})$ is $N(1.944821, 0.04618594)$. We can compare the sample mean and variance of the $M - J$ values that we obtained from MCMC with the numbers from part (b) and they should be close. For example, I got the following result after running the code above:

```
mean(betasamp[(J+1):M])
# [1] 1.93889
var(betasamp[(J+1):M])
# [1] 0.04721906
```

We can also plot the (density) histogram of the 15000 post-burnin samples and compare with the density curve of the $N(1.944821, 0.04618594)$ distribution. See Figure 1

```
hist(betasamp[(J+1):M],freq = FALSE)
curve(dnorm(x,A,sqrt(B)),add = TRUE,col = 'red')
```