**OM 386 Marketing Analytics II**

**Assignment 7**

**Khyathi Balusu**

**Kb42582**

**Due: May 8th, 11:59pm**

**Probit Regression: Bayesian Estimation**

In this exercise, we will practice coding the Gibbs sampler for a probit regression model using the dataset "CreditCard\_LatePayment\_Data.csv". The dataset has the following variables.

|  |  |
| --- | --- |
| ConsumerID | ID's of the sampled consumers |
| Latepay | Whether the consumer makes a late payment in the month |
| Usage | Monthly credit usage activities |
| Balance | The customer's outstanding balance in the month |

1). We would like fit the following probit regression model

*Yij\* = β0 + β1×Usageij + β2×Balanceij + εij*

*Latepayij =*0 if *Yij\** ≤ 0

*Latepayij =*1 if *Yij\** > 0

*εij ~N*(0, 1)

Please use the R function glm( ) to fit this model by MLE. Copy and paste the summary of the results here.

A screenshot of text

Description automatically generated

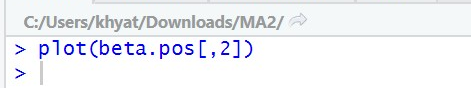
2).Next, we will fit the model above using a Gibbs sampler for Bayesian inference, which involves sampling the latent *Yij\**. Parts of the R code are in "Assignment-7\_Probit-code\_blanks.r". Please read the code carefully and fill in the code in the blanks in the file. You may use the rtruncnorm( ) function in the library(truncnorm) to sample from truncated normal distributions. For the linear regression part given the sampled latent *Yij\** in the main loop,

Please run the completed code. Use the plot() function to plot the posterior sampling chains and hist() to plot posterior histograms for *β0, β1, β2 .* Copy and paste the results here. Please also calculate the 95% posterior intervals for *β0, β1, β2 .* Copy and paste the results here.



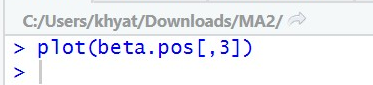
**A screenshot of a cell phone

Description automatically generated**

****

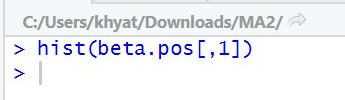
**A screenshot of a cell phone

Description automatically generated**

****

**A screenshot of a cell phone

Description automatically generated**

****

**A screenshot of a cell phone

Description automatically generated**

****

**A screenshot of a cell phone

Description automatically generated**

****

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**Metropolis-Hastings Algorithm for Bernoulli Estimation**

Next, we will practice coding the Metropolis-Hastings (MH) algorithm, which is another Markov Chain Monte Carlo method, for estimating the probability of a binary variable being equal to 1. The binary variable used in this exercise is “Latepay” in the dataset "CreditCard\_LatePayment\_Data.csv".

Assume the probability of the binary variable can only take values from a set of discretized values between 0 and 1. For example, this probability p can only be (0, 0.1, 0.2, 0.3,…, 0.9, 1.0). The MH algorithm uses a random walk to generate a proposal p value. The random walk allows the p value to jump to right of the current value with the probability=0.5 and jump to the left of the current value with the probability=0.5. For example, if the current p value is 0.2, it can jump to 0.1 with the probability=0.5 and jump to 0.3 with the probability=0.5. It cannot jump to any other values that are not the neighbors the current value. If the current value is 0, it can jump to the next higher value with the probability=0.5 or stay at 0 with the probability=0.5. If the current value is 1, it can jump to the next lower value with the probability=0.5 or stay at 1 with the probability=0.5. The MH algorithm compares the likelihood at the proposal p value and the current p value and decides which value is accepted as the posterior sample.

Please complete the code in “Assignment-7\_MH-code\_blanks.r” to implement this random walk Metropolis Hastings MCMC algorithm. Use the plot() function to plot the posterior sampling chains and calculate the posterior mean of the probability of the binary variable. Please copy and paste the plot and result here.



