MATH680 Homework 5 Fall 2015

This home work is due on Monday, Dec. 21 at 11:59pm. Submit your solutions in a pdf document on MyCourses. Name this file <yourlastname>HW5.pdf,e.g.yiHW4.pdf. Please also submit text files with your Rcode, which must be commented and properly indented. Name this file <yourlastname>HW5-code.txt, e.g. yiHW5-code.txt.

1. Let $y_{it} \in \{0,1\}$ be the binary response measurement on individual i at time t and let $x_{it} \in \mathbb{R}^p$ be the predictor values for individual i at time t, $t = 1, \ldots, T_i$, $i = 1, \ldots, n$. Suppose that y_{it} is a realization of Y_{it} where

$$(Y_{it}|B = \beta, U = u, V_U = v_U) \sim_{\text{ind}} \text{Bern}\left\{\frac{\exp(x_{it}'\beta + u_i)}{1 + \exp(x_{it}'\beta + u_i)}\right\}, i = 1, \ldots, n, t = 1, \ldots, T_i \quad (1)$$

$$\begin{aligned} (B = \beta|U = u, V_U = v_U) &\sim N_p(0, v_B I_p) \\ (U|V_U = v_U) &\sim N_n(0, v_U I_n) \\ V_U &\sim \text{InvGam}(a_0, b_0), \end{aligned} \quad (2)$$

where v_B, a_0, b_0 are user-specified prior parameters.

(a) Create a Markov chain Monte Carlo algorithm to approximately sample from the posterior distribution $(B, U, V_U|Y = y)$. Give a full derivation and carefully state the algorithm's steps.

(b) Create an R function that uses the algorithm created in part 1a to generate approximate samples from $(B, U, V_U|Y = y)$.

(c) Generate a dataset to test your function: set $n = 100$ $p = 3$, $T_i = 20$, $x_{it1} = 1$; generate the x_{it2}'s independently from $N(0.1, 1)$, and generate the x_{it3}'s independently from $N(1, 2)$; generate the $y_{it}'s$ using (1) with $\beta = (0, 1, -0.2)$ and u generated using (2) with $v_U = 2.89$. Use your R function from part 1b with $v_B = 100, a_0 = 1.01, b_0 = 1.01$ to generate 10,000 approximate samples from $(B, U, V_U|Y = y)$. Use diagnostics to identify possible problems with the approximation. Report an estimated Monte Carlo standard error for $E(B_2|Y = y)$.

(d) Pick values for n, v_B, a_0, b_0, the T_i's, and the $x_{it} \in \mathbb{R}^p$'s. Then generate an artificial dataset by generating the true parameters from the priors and the y_{it}'s from (1). Test your function in part 1b by generating 10,000 approximate samples from $(B, U, V_U|Y = y)$. Use diagnostics to identify possible problems with the approximation.

2. Define $X \in \mathbb{R}^{n \times p}$ to be the predictor matrix where its ith row has the nonrandom values of the p explanatory variables for the ith case $(i = 1, \ldots, n)$. Let $1_n \in \mathbb{R}^n$ be the vector of ones and let $I_n \in \mathbb{R}^{n \times n}$ be the n by n identity matrix. Suppose that the measured response for the n subjects $y = (y_1, \ldots, y_n)'$ is a realization of

$$Y = (Y_1, \ldots, Y_n)' \sim N_n\left(\mu_* 1_n + X\beta_*, v_* I_n\right),$$

where $\mu_* \in \mathbb{R}$, $\beta_* \in \mathbb{R}^p$, and $v_* > 0$ are unknown parameters. Let $\bar{y} = n^{-1}\sum_{i=1}^n y_i$ and $\bar{x} = n^{-1}X'1_n$. Define $\tilde{y} = y - \bar{y}1_n$ and $\widetilde{X} = X - 1_n\bar{x}'$.

Consider the estimates of β_* and μ_* defined by

$$\hat{\beta}^{(\lambda_1,\lambda_2)} = \arg\min_{\beta\in\mathbb{R}^p} \left\{ \frac{1}{2}\|\tilde{y} - \widetilde{X}\beta\|^2 + \frac{\lambda_1}{2}\|\beta\|^2 + \frac{\lambda_2}{2}\sum_{j=2}^{p}(\beta_j - \beta_{j-1})^2 \right\}, \tag{3}$$
$$\hat{\mu}^{(\lambda_1,\lambda_2)} = \bar{y} - \bar{x}'\hat{\beta}^{(\lambda_1,\lambda_2)},$$

where $\lambda_1, \lambda_2 \geq 0$ are tuning parameters and $\|\cdot\|$ is the L_2 norm.

(a) Write an R function called `cvfr` that performs K-fold cross-validation, minimizing prediction error, to select the tuning parameters (λ_1, λ_2) for $\hat{\beta}^{(\lambda_1,\lambda_2)}$ defined in (3). Here $K \in \{2, \ldots, n\}$. This function also computes $\hat{\beta}^{(\lambda_1,\lambda_2)}$ and $\hat{\mu}^{(\lambda_1,\lambda_2)}$ at the selected tuning parameters.

This function must have the following five arguments:

- `X`, the predictor matrix in $\mathbb{R}^{n\times p}$ (defined as X above)
- `y`, the response vector in $\mathbb{R}^n$ (a realization of Y defined above)
- `lam1.vec`, the vector of candidate values for λ_1.
- `lam2.vec`, the vector of candidate values for λ_2.
- `K`, the number of folds to use.

This function must return a list of the following five objects:

- `m`, this is $\hat{\mu}^{(\lambda_1,\lambda_2)} \in \mathbb{R}$ the estimate of the intercept (defined above) at the selected values for the tuning parameters.
- `b`, this is $\hat{\beta}^{(\lambda_1,\lambda_2)} \in \mathbb{R}^p$, the estimate of the regression coefficients at the selected values for the tuning parameters.
- `best.lam1`, this is the selected value for λ_1.
- `best.lam2`, this is the selected value for λ_2.
- `cv.error`, this is the matrix with (i, j)th element equal to the the squared prediction error totaled over the K folds when λ_1 equals the ith element of `lam1.vec` and λ_2 equals the $j$the element of `lam2.vec`.

(b) Consider a related Bayesian model:

$$\begin{aligned}(Y|B = \beta, V = v, L = \lambda) &\sim N_n(0 * 1_n + X\beta, vI_n),\\ (B|V = v, L = \lambda) &\sim N_p\left(0_p, \frac{v}{\lambda}\{wI_p + (1-w)H\}^{-1}\right),\\ (V|L = \lambda) &\sim \text{InvGam}(a_V, b_V),\\ L &\sim \text{Gamma}(a_L, b_L),\end{aligned}$$

where $0_p \in \mathbb{R}^p$ is the vector of zeros and $H \in \mathbb{R}^{p\times p}$ has (i, j)th entry

$$-1_{(|i-j|=1)} + 1_{(i=j)} + 1_{(i=j,2\leq j\leq p-1)}.$$

The parameters $w \in (0, 1], a_V > 0, b_V > 0, a_L > 0$, $b_L > 0$ are user-specified. For simplicity, this model does not have an intercept. The density for $\text{InvGam}(a_V, b_V)$ evaluated at v is proportional to $v^{-a_V-1}e^{-b_V/v}$ and the density for $\text{Gamma}(a_L, b_L)$ evaluated at λ is proportional to $\lambda^{a_L-1}e^{-b_L\lambda}$.

i. Create a blocked Gibbs Markov chain Monte Carlo algorithm to approximately sample from $(B, V, L|Y = y)$. Let $(\beta_{(k)}, v_{(k)}, \lambda_{(k)})$ be a realization of the kth element of the Markov Chain. This blocked Gibbs sampler generates $(\beta_{(k+1)}, v_{(k+1)})$ from $(B, V|L = \lambda_{(k)}, Y = y)$ and then generates $\lambda_{(k+1)}$ from $(L|B = \beta_{(k+1)}, V = v_{(k+1)}, Y = y)$.

ii. Test your algorithm in R. Set $n = 100$, $p = 5$, $v_* = 2$, and $\beta_* = (1, 1, 0.75, 0.5, 0.5)'$. When generating data, only calls to `runif` are permitted, e.g. calling `rnorm` is not allowed. Generate the rows of X as independent draws from $N_5(0, \Sigma)$, where $\Sigma_{ij} = 0.7^{|i-j|}$ for $(i, j) \in \{1, \ldots, 5\} \times \{1, \ldots, 5\}$, then generate y from $N_{100}(0 * 1_{100} + X\beta_*, v_* I_{100})$. Use your function `cvfr` on these data to compute $\hat{\beta}^{(\hat{\lambda}_1, \hat{\lambda}_2)}$ where 5-fold cross-validation is used to select $(\hat{\lambda}_1, \hat{\lambda}_2)$ from the set

$$\{10^{-8}, 10^{-7.5}, \ldots, 10^{7.5}, 10^{8}\} \times \{10^{-8}, 10^{-7.5}, \ldots, 10^{7.5}, 10^{8}\}.$$

Set $w = \hat{\lambda}_1/(\hat{\lambda}_1 + \hat{\lambda}_2)$, $a_L = 2$, $b_L = 1/(\hat{\lambda}_1 + \hat{\lambda}_2)$, $a_V = 2$, and $b_V = 3\|X\hat{\beta} - y\|^2/100$. Generate a chain of 10^5 approximate samples. Report a 99% marginal credible interval for each element of β_*. Report and comment on relevant trace plots and ACF plots.

3. Create an example distribution that was not studied in lecture and a Metropolis–Hastings or variable at a time Metropolis–Hastings algorithm to approximately sample from it.

   (a) Give a full derivation of this algorithm and carefully state its steps.

   (b) Use R to test your algorithm on simulated data (that you create). Use diagnostics to identify possible problems with the approximation.