BayesSummaryStatLM Tutorial

Bibby(Mi) Zhou

UCR-Statisitcs

Feburary 8, 2018

Outline

- Overview of BayesSummaryStatLM R package
- Example with simulation data
- Outputs
- Graphical Diagnostics

Overview: BayesSummaryStatLM

- BayesSummaryStatLM is an R package for Bayesian linear regression models for big data that includes several choices of prior distributions for the unknown model parameters.
- Markov chain Monte Carlo (MCMC) procedures for Bayesian linear regression models with normally distributed errors that use only summary statistics as input.
- Can handle huge data set (use only summary statistics of data as input).
- Can analyze data that is updated over time.
- Overcomes physical memory limits of a user.

Methods

Bayesian linear regression model

The purpose of linear regression is to model a response variable Y using a set of predictor variables $X = (X_1, ..., X_k)$. The model with K predictors is as following:

$$Y_i = \beta_0 + \beta_1 X_{1i} + ... + \beta_k X_{ki} + \epsilon_i \tag{1}$$

where $i = 1, ..., n$ and $\epsilon_i \sim Normal(0, \sigma^2)$. The likelihood is given by:

$$L(Y|\beta_0, \beta_1, ..., \beta_k, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} exp[-\frac{1}{2\sigma^2}(Y - X\beta)'(Y - X\beta)] \tag{2}$$

where Y is an $n \times 1$ column vector, X is an $n \times (k+1)$ matrix and β is a $(k+1) \times 1$ column vector.

Methods

•The parameters to be estimated are:
·Regression coefficients: $\beta = (\beta_0, \beta_1, ..., \beta_k)'$
· Error variance parameter σ^2.

•In the Bayesian framework, we assign prior distributions to β and σ^2 and produce the joint posterior distribution as the product of the likelihood and prior distributions;

•Assume priors of β and σ^2 are independent. The full conditional posterior distributions for the parameters are proportional to the joint posterior distribution, treating all other parameters as fixed constants.

• The Gibbs sampler is used to sample from the full conditional posterior distributions.

Methods

• In BayesSummaryStatLM package, the full conditional posterior distribution depend on the data only through the summary statistics $\boldsymbol{X'X}, \boldsymbol{X'Y}$ for β, and $\boldsymbol{X'X}, \boldsymbol{X'Y}, \boldsymbol{Y'Y}$ for σ^2.
• These values can be calculated by combining summaries from subsets of data. In this package, it assumes the data is partitioned horizontally by the samples n into M nonoverlapping subsets, such that if X is dimension $n \times \psi$, then the partition is by the following:

$$\boldsymbol{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix} \tag{3}$$

where each $\boldsymbol{X_m}, m = 1, ..., M$ has ψ columns.

Methods

The full data summary statistics are calculated as follows, for $m = 1, ...M$ chunks:
Full data

$$\boldsymbol{X'X} = \sum_{m=1}^{M} \boldsymbol{X'_m X_m}, \tag{4}$$

$$\boldsymbol{X'Y} = \sum_{m=1}^{M} \boldsymbol{X'_m Y_m}, \tag{5}$$

$$\boldsymbol{Y'Y} = \sum_{m=1}^{M} \boldsymbol{Y'_m Y_m}, \tag{6}$$

The $\boldsymbol{Y}$ vector is also s also partitioned horizontally, similarly to Formula (4).
The Gibbs sampler is used to sample from all full conditional posterior distributions.

Priors

Prior distributions for β

• Uniform prior for $\boldsymbol{\beta}$.

• Multivariate Normal prior for $\boldsymbol{\beta}$ with known mean vector $\boldsymbol{\mu}$ and **known** covariance matrix Σ.

• Multivariate Normal prior for $\boldsymbol{\beta}$ with unknown mean vector$\boldsymbol{\mu}$ and **unknown** covariance matrix Σ.

Prior distributions for σ^2

• Inverse Gamma prior for σ^2 with known shape and scale parameters.

• Inverse sigma squared prior for σ^2(the Jeffreys prior for σ^2).

Function Arguments

• Two major functions in this package: $read.regress.data.ff()$ and $bayes.regress()$.
• $read.regress.data.ff()$ returns a **list** of the summary statistics: $\boldsymbol{X'X}, \boldsymbol{X'Y}, \boldsymbol{Y'Y}$ for later use and the total number of data values:

```
read.regress.data.ff(filename, predictor.cols,
  response.col, first.rows, next.rows,
  update.summaries)
```

$first.rows$: The number of rows to read in the first chunk of data. Default = 100,000.
$next.rows$: The number of rows to read in the remaining chunks of data. Default = 100,000.

Function Arguments

• $bayes.regress()$ is used to generate the MCMC posterior samples for the unknown Bayesian linear regression model parameters. This function takes as input the summary statistics calculated by the function $read.regress.data.ff()$.

```
bayes.regress(data.values = list(xtx, xty, yty,
  numsamp.data), beta.prior, sigmasq.prior, Tsamp.out,
  zero.intercept)
```

The options of β priors: ”flat”,”mvnorm.known” and ”mvnorm.unknown”. The options of σ^2 priors: ”inverse.gamma”, ”sigmasq.inverse”.

Example

Simulate data from the linear regression model (1) with 10 predictor variables, with data sample size 10000.

The matrix $\boldsymbol{X} = (\boldsymbol{X_1}, ..., \boldsymbol{X_{10}})$ was simulated from a multivariate normal distribution by the following, where each column vector represents a predictor variable: $\boldsymbol{X} \sim Normal(0, \Sigma)$.

the variance-covariance matrix $\Sigma = \begin{bmatrix} 1 & 0.2 & 0.2 & \ldots, & 0.2 \\ 0.2 & 1 & 0.2 & \ldots, & 0.2 \\ \vdots & \vdots & \vdots & \ldots, & \vdots \\ 0.2 & 0.2 & 0.2 & \ldots, & 1 \end{bmatrix}$ The model parameters β were simulated from a standard normal distribution.

The error parameter σ^2 was assigned $\sigma^2 = 1$.

Then the response values were simulated from the model:

$$Y_i = \beta_0 + \beta_1 X_{1i} + ... + \beta_{10} X_{10.i} + \epsilon_i \tag{7}$$

Example

```r
################## important stat ####################
sim.regress.data <- read.regress.data.ff(filename ='mydata2.csv', predictor.cols = c(2:11), response.col =
                                          1, first.rows = 10, next.rows = 10)

xtx<-sim.regress.data$xtx
yty<-sim.regress.data$yty
xty<-sim.regress.data$xty
```

```
> sim.regress.data
$xtx
           [,1]         [,2]        [,3]         [,4]       [,5]      [,6]       [,7]
 [1,] 10000.000000     4.798397    40.00344    -8.797452   150.1407  -23.1497  -21.82323
 [2,]     4.798397 10112.454239  2021.94982  1897.900335  2019.2612 1874.7060 2088.50401
 [3,]    40.003440  2021.949818 10191.49249  2054.473003  2075.2047 2133.8557 2041.85588
 [4,]    -8.797452  1897.900335  2054.47300 10054.017392  2082.5388 1778.6181 1989.40727
 [5,]   150.140691  2019.261240  2075.20469  2082.538801 10039.7238 1950.5223 1933.10928
 [6,]   -23.149702  1874.706024  2133.85574  1778.618125  1950.5223 9792.8838 2092.03372
 [7,]   -21.823232  2088.504012  2041.85588  1989.407272  1933.1093 2092.0337 9935.47144
 [8,]   127.561938  2207.160102  2101.16999  1875.373298  1973.5617 1868.8271 2121.82991
 [9,]   -28.512158  1860.412676  2041.89206  1905.316096  1932.5659 1957.6988 2035.24879
[10,]  -152.888513  2063.301721  2057.38906  1891.060243  1992.8083 1921.9532 2083.79466
[11,]    38.165254  2096.519424  2161.28647  1904.742082  1876.5463 1992.4161 2030.43861
          [,8]       [,9]      [,10]      [,11]
 [1,]  127.5619  -28.51216  -152.8885   38.16525
 [2,] 2207.1601 1860.41268  2063.3017 2096.51942
 [3,] 2101.1700 2041.89206  2057.3891 2161.28647
 [4,] 1875.3733 1905.31610  1891.0602 1904.74208
 [5,] 1973.5617 1932.56586  1992.8083 1876.54629
 [6,] 1868.8271 1957.69881  1921.9532 1992.41614
 [7,] 2121.8299 2035.24879  2083.7947 2030.43861
 [8,] 9915.3568 2010.58897  2074.6383 2013.22229
 [9,] 2010.5890 9958.25854  1892.3597 1737.08226
[10,] 2074.6383 1892.35974 10061.0998 1940.07553
[11,] 2013.2223 1737.08226  1940.0755 9839.48913

$xty
 [1]  4705.4438  1333.9564 20902.2525  3484.5927 11766.3793 22557.8665  7548.3134  5756.6387
 [9]   898.7658 19421.7401 21183.5207

$yty
         V1
V1 144887.4

$numsamp.data
[1] 10000
```

Example

```
##### prior ##############

beta.prior.2 <- list(type="mvnorm.known", mean.mu =
                      rep(0.0, dim(xtx)[1]), cov.C = diag(1.0, dim(xtx)[1]))
sigmasq.prior.1 <- list(type = "inverse.gamma",
                      inverse.gamma.a = 1, inverse.gamma.b = 1, sigmasq.init = 1)
```

```
#### regression ########
sim.beta.sigmasq.out <- bayes.regress(data.values =
                                  sim.regress.data, beta.prior = beta.prior.2, sigmasq.prior
                                 = sigmasq.prior.1, Tsamp.out = 10000, zero.intercept =
                                  FALSE)
```

Output

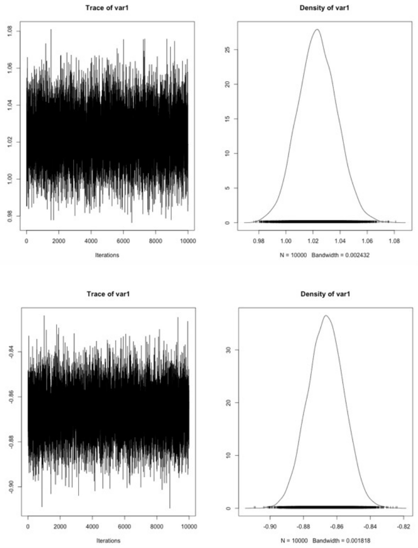
•The output is a list containing a matrix of MCMC posterior samples for β of dimension = (Tsamp.out, k+1), and a vector of MCMC posterior samples for $\sigma^2$of dimension = (Tsamp.out) which is the number of MCMC posterior samples.

• To further analysis MCMC posterior samples, we can use the R pacakge $coda$. The output of $sim.beta.sigmasq.outi$ can be converted to class "mcmc" using the $mcmc()$ function:

```
#### furtehr check on b1 and sigma ##########
plot(mcmc(sim.beta.sigmasq.out$beta[,2])) #no burn-in
plot(mcmc(sim.beta.sigmasq.out$beta[500:10000,2])) # burnin first 500
summary(mcmc(sim.beta.sigmasq.out$beta[,2]))

plot(mcmc(sim.beta.sigmasq.out$sigmasq))
summary(mcmc(sim.beta.sigmasq.out$sigmasq))
```

Graphical Diagnostics for β_1 and σ^2



Summary statistics of β_1 for σ^2

```
Iterations = 1:10000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 10000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

      Mean             SD       Naive SE Time-series SE
   1.0232481      0.0144756    0.0001448      0.0001471

2. Quantiles for each variable:

  2.5%    25%    50%    75%  97.5%
0.9952 1.0134 1.0231 1.0330 1.0520
```

```
Iterations = 1:10000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 10000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

      Mean             SD       Naive SE Time-series SE
  -0.8665709      0.0108198    0.0001082      0.0001060

2. Quantiles for each variable:

   2.5%     25%     50%     75%   97.5%
-0.8876 -0.8738 -0.8666 -0.8592 -0.8455
```

The returned value for the 95% posterior equal-tail credible interval limits of β_1 is (-0.8876, -0.8455) includes the simulated value of -0.8638 for β_1.

The returned value for the 95% posterior equal-tail credible interval limits of σ^2 is (0.9952, 1.0520) includes the simulated value of 1 for σ^2.

Table

Posterior mean and posterior 2.5%,97.5% percentiles for the unknown model parameters for the simulation and the true parameters.

parameters	True value	Posterior Mean	Posterior 2.5% bound	Posterior 97.5% bound	
β_0	0.4623	0.47832	0.4588	0.4985	*
β_1	-0.8638	-0.8666	-0.8876	-0.8454	*
β_2	1.4790	1.4874	1.466	1.509	*
β_3	-0.5139	-0.5377	-0.5591	-0.5163	N
β_4	0.4335	0.4564	0.4351	0.4776	N
β_5	1.7971	1.7992	1.777	1.821	*
β_6	-0.0874	-0.1093	-0.1304	-0.0876	*
β_7	-0.3138	-0.3031	-0.3243	-0.2815	*
β_8	-0.8459	-0.8580	-0.8792	-0.8363	*
β_9	1.4213	1.4124	1.391	1.434	*
β_{10}	1.6152	1.6189	1.597	1.640	*
σ^2	1.0000	1.0232	0.9952	1.0520	*

Thank you !!