Package ‘MLModelSelection’

March 20, 2020

Type Package

Title Model Selection in Multivariate Longitudinal Data Analysis

Version 1.0

Date 2020-03-13

Author Kuo-Jung Lee

Maintainer Kuo-Jung Lee <kuojunglee@mail.ncku.edu.tw>

Description An efficient Gibbs sampling algorithm is developed for Bayesian multivariate longitudinal data analysis with the focus on selection of important elements in the generalized autoregressive matrix. It provides posterior samples and estimates of parameters. In addition, estimates of several information criteria such as Akaike information criterion (AIC), Bayesian information criterion (BIC), deviance information criterion (DIC) and prediction accuracy such as the marginal predictive likelihood (MPL) and the mean squared prediction error (MSPE) are provided for model selection.

URL https://github.com/kuojunglee/

Depends R(>= 3.5.0)

License GPL-2

Imports Rcpp (>= 1.0.1), MASS

Suggests testthat

LinkingTo Rcpp, RcppArmadillo, RcppDist

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-03-20 15:10:08 UTC

R topics documented:

MLModelSelectionMCMC *Model estimation for multivariate longitudinal models.*

Description

Using MCMC procedure to generate posterior samples and provide AIC, BIC, DIC, MPL, MSPE, and predicted values.

Usage

```
MLModelSelectionMCMC(Num.of.iterations, list.Data, list.InitialValues, list.HyperPara,
list.UpdatePara, list.TuningPara)
```

Arguments

`Num.of.iterations`
: Number of iterations.

`list.Data`
: *List* of data set containing response Y, design matrix X, avialable time points for each subject, GARP model, and ISD model.

`list.InitialValues`
: *List* of initial values for parameters.

`list.HyperPara`
: *List* of given hyperparameters in priors.

`list.UpdatePara`
: Determine which parameter will be updated.

`list.TuningPara`
: Provide turning parameters in proposal distributions.

Details

We set the subject i ($i = 1, \ldots, N$) has K continuous responses at each time point t ($t = 1, \ldots, n_i$). Assume that the measurement times are common across subjects, but not necessarily equally-spaced. Let $y_{it} = (y_{it1}, \ldots, y_{itK})$ denote the response vector containing K continuous responses for ith subject at time t along with a $p \times 1$ vectof of covariates, $x_{it} = (x_{it1}, \ldots, x_{itp})$. An efficient Gibbs sampling algorithm is developed for model estimation in the multivariate longitudinal model given by

$$y_{i1k} = x'_{it}\beta_k + e_{i1k}, t = 1;$$

$$y_{itk} = x'_{it}\beta_k + \sum_{g=1}^{K}\sum_{j=1}^{t-1}\phi_{itj,kg}(y_{ijg} - x'_{ij}\beta_g) + e_{itk}, t \geq 2,$$

where $\beta_k = (\beta_{k1}, \ldots, \beta_{kp})'$ is a vector of regression coefficients of length p, $\phi_{itj,kg}$ is a generalized autoregressive parameter (GARP) to explain the serial dependence of responses across time. Moreover,

$$\phi_{itj,kg} = \alpha_{kg}\mathbf{1}\{|t-j| = 1\},\ \log(\sigma_{itk}) = \lambda_{k0} + \lambda_{k1}h_{it},\ \log\left(\frac{\omega_{ilm}}{\pi - \omega_{ilm}}\right) = \nu_l + \nu_m.$$

The priors for the parameters in the model given by

$$\beta \sim \mathcal{N}(0, \sigma_\beta^2 I);$$

$$\lambda_k \sim \mathcal{N}(0, \sigma_\lambda^2 I);$$

$$\nu_k \sim \mathcal{N}(0, \sigma_\nu^2 I), \quad k = 1, \ldots, K,$$

where σ_β^2, σ_λ^2, and σ_ν^2 are prespecified values. For $k, g = 1, \ldots, K$ and $m = 1, \ldots, a$, we further assume

$$\alpha_{kgm} \sim \delta_{kgm}\mathcal{N}(0, \sigma_\delta^2) + (1 - \delta_{kgm})\eta_0,$$

where σ_δ^2 is prespecified value and η_0 is the point mass at 0.

Value

Lists of posterior samples, parameters estimates, AIC, BIC, DIC, MPL, MSPE, and predicted values are returned

Note

We'll provide the reference for details of the model and the algorithm for performing model estimation whenever the manuscript is accepted.

Author(s)

Kuo-Jung Lee

References

Keunbaik Lee et al. (2015) Estimation of covariance matrix of multivariate longitudinal data using modified Choleksky and hypersphere decompositions. *Biometrics*. **75-86, 2020**. doi: 10.1111/biom.13113.

Examples

```
library(MASS)
library(MLModelSelection)


AR.Order = 6 #denote \phi_{itj, kg} = \alpha_{kg} \mathbf{1}{|t-j|=1}
ISD.Model = 1 #denote \log(\sigma_{itk}) = \lambda_{k0} + \lambda_{k1} h_{it}

data(SimulatedData)

N = dim(SimulatedData$Y)[1] # the number of subjects
T = dim(SimulatedData$Y)[2] # time points
K = dim(SimulatedData$Y)[3] # the number of attributes
P = dim(SimulatedData$X)[3] # the number of covariates
M = AR.Order  # the demension of alpha
nlamb = ISD.Model + 1 # the dimension of lambda
```

```
Data = list(Y = SimulatedData$Y, X = SimulatedData$X,
TimePointsAvailable = SimulatedData$TimePointsAvailable,
AR.Order = AR.Order, ISD.Model = ISD.Model)

beta.ini = matrix(rnorm(P*K), P, K)
delta.ini = array(rbinom(K*K*M, 1, 0.1), c(K, K, M))
alpha.ini = array(runif(K*K*M, -1, 1), c(K, K, M))
lambda.ini = matrix(rnorm(nlamb*K), K, nlamb, byrow=T)
nu.ini = rnorm(K)


InitialValues = list(beta = beta.ini, delta = delta.ini, alpha = alpha.ini,
lambda = lambda.ini, nu = nu.ini)

# Hyperparameters in priors
sigma2.beta = 1
sigma2.alpha = 10
sigma2.lambda = 0.01
sigma2.nu = 0.01

# Whehter the parameter will be updated
UpdateBeta = TRUE
UpdateDelta = TRUE
UpdateAlpha = TRUE
UpdateLambda = TRUE
UpdateNu = TRUE


HyperPara = list(sigma2.beta = sigma2.beta, sigma2.alpha=sigma2.alpha,
sigma2.lambda=sigma2.lambda, sigma2.nu=sigma2.nu)


UpdatePara = list(UpdateBeta = UpdateBeta, UpdateAlpha = UpdateAlpha, UpdateDelta = UpdateDelta,
              UpdateLambda = UpdateLambda, UpdateNu = UpdateNu)

# Tuning parameters in proposal distribution within MCMC
TuningPara = list(TuningAlpha = 0.01, TuningLambda = 0.005, TuningNu = 0.005)

num.of.iter = 100

start.time <- Sys.time()

PosteriorSamplesEstimation = MLModelSelectionMCMC(num.of.iter, Data, InitialValues,
HyperPara, UpdatePara, TuningPara)

end.time <- Sys.time()

cat("Estimate of beta\n")
print(PosteriorSamplesEstimation$PosteriorEstimates$beta.mean)
```

SimulatedData *Simulated data*

Description

A simulated multivariate longitudinal data for demonstration.

Usage

```
data("SimulatedData")
```

Format

A `list` consists of `Y` the observations 100 subjects in 3 attributes along 10 time points, `X` the design matrix with 6 covariate including the intercept, `TimePointsAvailable` the avilable time points for each subject.

`Y` The response variables.

`X` The design matrix.

`TimePointsAvailable` The available time points for each subject.

Examples

```
library(MLModelSelection)
data(SimulatedData)
SimulatedData = data(SimulatedData)
```

Index