Package 'MSIMST'

September 16, 2024

Type Package

Title Bayesian Monotonic Single-Index Regression Model with the Skew-T Likelihood

Version 1.1

Date 2024-08-16

Description Incorporates a Bayesian monotonic single-index mixed-effect model with a multivariate skew-t likelihood, specifically designed to handle survey weights adjustments. Features include a simulation program and an associated Gibbs sampler for model estimation. The single-index function is constrained to be monotonic increasing, utilizing a customized Gaussian process prior for precise estimation. The model assumes random effects follow a canonical skew-t distribution, while residuals are represented by a multivariate Student-t distribution. Offers robust Bayesian adjustments to integrate survey weight information effectively.

URL https://github.com/rh8liuqy/MSIMST

License GPL (>= 3)

Depends R (>= 3.4.0)

Imports MASS (>= 7.3-58.4), Rcpp (>= 1.0.12), mvtnorm (>= 1.2-4), fields (>= 15.2), parallel (>= 4.3.0), truncnorm (>= 1.0-9), Rdpack (>= 2.6)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.2

Encoding UTF-8

Suggests lattice (>= 0.21-8), HDInterval (>= 0.2.4), latex2exp (>= 0.9.6), posterior (>= 1.5.0)

RdMacros Rdpack

NeedsCompilation yes

Author Qingyang Liu [aut, cre] (<https://orcid.org/0000-0003-3265-6330>),
Debdeep Pati [aut],
Dipankar Bandyopadhyay [aut]

Maintainer Qingyang Liu <rh8liuqy@gmail.com>

Repository CRAN

Date/Publication 2024-09-16 07:20:05 UTC

Contents

`Gibbs_Sampler` *The Associated Gibbs Sampler*

Description

This is the Gibbs sampler associated with the proposed single-index mixed-effects model. This Gibbs sampler supports three different likelihoods, normal, skew-normal and skew-t likelihoods and two types of priors for the single-index funcion: the Gaussian process (GP) prior and the bernstein polynomial (BP) prior.

Usage

```
Gibbs_Sampler(
  X,
  y,
  group_info,
  beta_value,
  beta_prior_variance,
  beta_b_value,
  beta_lambdasq_value,
  beta_tausq_value,
  xi_value,
  xi_lengthscale_value,
  xi_tausq_value,
  g_func_type,
  dsq_value,
  sigmasq_value,
  delta_value,
  nu_value,
  U_value,
  S_value,
  loglik_type,
  gof_K,
  gof_L,
  iter_warmup,
  iter_sampling,
```

```
  verbatim,
  update = 10,
  incremental_output = FALSE,
  incremental_output_filename = NULL,
  incremental_output_update = 1e+06,
  n_core = 1
)
```

Arguments

X	The list of design matrix.
y	The list of response values.
group_info	The group information for the grouped horseshoe prior. Use 0 to represent the variables with the normal priors. Use 1,2,... to present the variables with the grouped horseshoe priors. For example, c(0,0,1,1,2,3) represents first two variables with the normal prior, third and fourth variables belong to the same group with one grouped horseshoe prior, and fifth and sixth variables belong to two different groups with two independent horseshoe prior.
beta_value	The initial value for the covariates' coefficients.
beta_prior_variance	The variance value of the normal prior.
beta_b_value	The slope parameter.
beta_lambdasq_value	The first hyperparameter associated with the grouped horseshoe prior.
beta_tausq_value	The second hyperparameter associated with the grouped horseshoe prior.
xi_value	The parameters associated with the single index function.
xi_lengthscale_value	The first hyperparameter associated with the Gaussian process kernel.
xi_tausq_value	The second hyperparameter associated with the Gaussian process kernel.
g_func_type	The type of priors on the single index function. Must be one of "GP" and "BP".
dsq_value	The initial value of the conditional variance of the random effects.
sigmasq_value	The initial value of the conditional variance of the fixed effects.
delta_value	The initial value of the skewness parameter.
nu_value	The initial value of the degree of freedom. Must be larger than 2.
U_value	The initial values of the latent variable U. The length of `U_value` must be as the same as the number of subjects.
S_value	The initial values of the latent variable S. The length of `S_value` must be as the same as the number of subjects.
loglik_type	The type of the log-likelihood. Must be one of "skewT","skewN", and "N".
gof_K	The first hyperparameter associated with the goodness of fit test. Check (Yuan and Johnson 2012) for details.
gof_L	The second hyperparameter associated with the goodness of fit test. Check (Yuan and Johnson 2012) for details.

- `iter_warmup` The number of warm-up iterations of the Gibb samplers.
- `iter_sampling` The number of post warm-up iterations of the Gibb samplers.
- `verbatim` TRUE/FALSE. If verbatim is TRUE, then the updating message of the Gibbs sampler will be printed.
- `update` An integer. For example, if `update` = 10, for each 10 iteration, one udpating message of the Gibbs sampler will be printed.
- `incremental_output` TRUE/FALSE. If `incremental_output` is TRUE, an incremental output will be saved. This option should not be enabled unless users anticipate the sampling process will take longer than days.
- `incremental_output_filename` The filename of the incremental output.
- `incremental_output_update` An integer. For example, if `incremental_output_update` = 10 then for each 10 iteration, the intermediate result will be updated once.
- `n_core` The number of cores will be used during the Gibbs sampler. For the Windows operating system, `n_core` must be 1.

Details

The details of the ST-GP model can be found in the vignette. Users can access the vignette using `vignette(package = "MSIMST")`.

Value

A list of random quantitiles drawn from the posterior distribution using the Gibbs sampler.

Examples

```
# Set the random seed.
set.seed(100)

# Simulate the data.
simulated_data <- reg_simulation1(N = 50,
                                  ni_lambda = 8,
                                  beta = c(0.5,0.5,0.5),
                                  beta_b = 1.5,
                                  dsq = 0.1,
                                  sigmasq = 0.5,
                                  delta = 0.6,
                                  nu = 5.89)

y <- simulated_data$y
X <- simulated_data$X

group_info <- c(0,0,0)
# The number of grids (L) for approximating the single index function
L <- 50
N <- length(y)
```

```
GP_MCMC_output <- Gibbs_Sampler(X = X,
                                y = y,
                                group_info = group_info,
                                beta_value = c(0.5,0.5,0.5),
                                beta_prior_variance = 10,
                                beta_b_value = 1.5,
                                beta_lambdasq_value = 1,
                                beta_tausq_value = 1,
                                xi_value = abs(rnorm(n = L + 1)),
                                xi_lengthscale_value = 1.0,
                                xi_tausq_value = 1.0,
                                g_func_type = "GP",
                                dsq_value = 1,
                                sigmasq_value = 1,
                                delta_value = 0.6,
                                nu_value = 5.89,
                                U_value = abs(rnorm(N)),
                                S_value = abs(rnorm(N)),
                                loglik_type = "skewT",
                                gof_K = 10,
                                gof_L = 5,
                                iter_warmup = 10,
                                iter_sampling = 20,
                                verbatim = TRUE,
                                update = 10,
                                incremental_output = FALSE,
                                incremental_output_filename = NULL,
                                incremental_output_update = 1e6,
                                n_core = 1)
```

MSIMST *The 'MSIMST' package.*

Description

Incorporates a Bayesian monotonic single-index mixed-effect model with a multivariate skew-t likelihood, specifically designed to handle survey weights adjustments. Features include a simulation program and an associated Gibbs sampler for model estimation. The single-index function is constrained to be monotonic increasing, utilizing a customized Gaussian process prior for precise estimation. The model assumes random effects follow a canonical skew-t distribution, while residuals are represented by a multivariate Student-t distribution. Offers robust Bayesian adjustments to integrate survey weight information effectively.

Author(s)

Maintainer: Qingyang Liu <rh8liuqy@gmail.com> (ORCID)

Authors:

- Debdeep Pati <debdeep@stat.tamu.edu>
- Dipankar Bandyopadhyay <dbandyop@vcu.edu>

See Also

Useful links:

- https://github.com/rh8liuqy/MSIMST

---

phiX_c *The Function to Calculate the phiX Matrix for Estimating Single-Index Function*

Description

The function phiX_c is used to generate the phiX matrix associated with the Gaussian process prior.

Usage

```
phiX_c(Xbeta, u, L)
```

Arguments

Xbeta	The single index values. A vector of length n.
u	The vector spanning from -1 to 1 with length L + 1.
L	An integer defining the number of nodes.

Value

A n by L + 1 matrix.

Examples

```
L <- 50
u <- seq(-1,1,length.out = L + 1)
phiX <- phiX_c(0.5,u,L)
print(phiX)
```

---

reg_simulation1 *The Function for the Simulation Study without the Variable Selection*

Description

This is a simply simulation study that is designed to demonstrate the correctness of the proposed Gibbs sampler, Gibbs_Sampler().

Usage

```
reg_simulation1(N, ni_lambda, beta, beta_b, dsq, sigmasq, delta, nu)
```

Arguments

N	The number of subjects.
ni_lambda	The mean of Poisson distribution.
beta	A 3 by 1 vector.
beta_b	The slope of PD response.
dsq	A part of covariance parameter.
sigmasq	A part of covariance parameter.
delta	The skewness parameter.
nu	The degree of freedom.

Details

More details of the design of this simulation study can be found in the vignette. Users can access the vignette by the command `vignette(package = "MSIMST")`.

Value

A simulated dataset with the response variable `y` and the design matrix `X`.

Examples

```
set.seed(100)
simulated_data <- reg_simulation1(N = 50,
                                  ni_lambda = 8,
                                  beta = c(0.5,0.5,0.5),
                                  beta_b = 1.5,
                                  dsq = 0.1,
                                  sigmasq = 0.5,
                                  delta = 0.6,
                                  nu = 5.89)
y <- simulated_data$y
X <- simulated_data$X
print(head(y))
print(head(X))
```

---

reg_simulation2 *The Function for the Simulation Study with the Variable Selection*

Description

This simulation study is designed to demonstrate that using the grouped horseshoe prior can successfully separate signals from noise.

Usage

```
reg_simulation2(N, ni_lambda, beta, beta_b, dsq, sigmasq, delta, nu)
```

Arguments

N	The number of subjects.
ni_lambda	The mean of Poisson distribution.
beta	The covariates' coefficients. A 10 by 1 vector.
beta_b	The slope of PD response.
dsq	A part of covariance parameter.
sigmasq	A part of covariance parameter.
delta	The skewness parameter.
nu	The degree of freedom.

Details

More details of the design of this simulation study can be found in the vignette. Users can access the vignette by the command `vignette(package = "MSIMST")`.

Value

A simulated dataset with the response variable `y` and the design matrix `X`.

Examples

```
set.seed(200)
simulated_data <- reg_simulation2(N = 50,
                                  ni_lambda = 8,
                                  beta = c(rep(1,6),rep(0,4)),
                                  beta_b = 1.5,
                                  dsq = 0.1,
                                  sigmasq = 0.5,
                                  delta = 0.6,
                                  nu = 5.89)

y <- simulated_data$y
X <- simulated_data$X
```

`reg_simulation3` *The Function for the Simulation Study with the Variable Selection and Survey Weights*

Description

This simulation study is designed to show the effectiveness of the grouped horseshoe prior for the variable selection and the `WFPBB()` function for adjusting survey weights.

Usage

```
reg_simulation3(
  N,
  ni_lambda,
  beta,
  beta_b,
  dsq,
  sigmasq,
  delta,
  nu,
  muz,
  rho,
  sigmasq_z,
  zeta0,
  zeta1
)
```

Arguments

N	The number of subjects.
ni_lambda	The mean of Poisson distribution.
beta	The covariates' coefficients. A 10 by 1 vector.
beta_b	The slope of PD response.
dsq	A part of covariance parameter.
sigmasq	A part of covariance parameter.
delta	The skewness parameter.
nu	The degree of freedom.
muz	The location parameter of the latent/selection variable.
rho	The correlation parameter of the latent/selection variable.
sigmasq_z	The variance parameter of the latent/selection variable.
zeta0	The intercept term inside the logistic function.
zeta1	The slope term inside the logistic function.

Details

More details of the design of this simulation study can be found in the vignette. Users can access the vignette by the command `vignette(package = "MSIMST")`.

Value

A simulated dataset with the response variable `y`, the design matrix `X` and the survey weight `survey_weight`.

Examples

```
set.seed(100)
output_data <- reg_simulation3(N = 1000,
                               ni_lambda= 8,
                               beta = c(rep(1,6),rep(0,4)),
                               beta_b = 1.5,
                               dsq = 0.1,
                               sigmasq = 0.5,
                               delta = 0.6,
                               nu = 5.89,
                               muz = 0,
                               rho = 36.0,
                               sigmasq_z = 0.6,
                               zeta0 = -1.8,
                               zeta1 = 0.1)
y <- output_data$y
X <- output_data$X
survey_weight <- output_data$survey_weight
```

WFPBB *Weighted Finite Population Bayesian Bootstrap*

Description

The function is implemented based on the WFPBB algorithm from (Gunawan et al. 2020).

Usage

```
WFPBB(y, w, N, n, verbatim)
```

Arguments

`y`	The index of survey data.
`w`	Survey weights. The summation of survey weights should equal the population size
`N`	The population size.
`n`	The sample size.
`verbatim`	TRUE/FALSE. This variable decides whether print the progress information to the console.

Value

The re-sampled index of survey data.

References

Gunawan D, Panagiotelis A, Griffiths W, Chotikapanich D (2020). “Bayesian weighted inference from surveys.” *Australian & New Zealand Journal of Statistics*, **62**(1), 71–94. ISSN 1467-842X, doi:10.1111/anzs.12284.

Examples

```
set.seed(100)
output_data <- reg_simulation3(N = 5000,
                               ni_lambda= 8,
                               beta = c(rep(1,6),rep(0,4)),
                               beta_b = 1.5,
                               dsq = 0.1,
                               sigmasq = 0.5,
                               delta = 0.6,
                               nu = 5.89,
                               muz = 0,
                               rho = 36.0,
                               sigmasq_z = 0.6,
                               zeta0 = -1.8,
                               zeta1 = 0.1)
y <- output_data$y
X <- output_data$X
survey_weight <- output_data$survey_weight
# set the population size
population_N <- 5000
# set the sample size
n <- length(y)
# run the WFPBB algorithm
index_WFPBB <- WFPBB(y = 1:n,
                     w = survey_weight,
                     N = population_N,
                     n = n,
                     verbatim = FALSE)
print(head(index_WFPBB))
```

Index