

Package ‘BMIselect’

October 7, 2026

Title Bayesian MI-LASSO for Variable Selection on Multiply-Imputed Datasets

Version 1.0.10

Maintainer Jungang Zou <jungang.zou@gmail.com>

Description Provides a suite of Bayesian MI-LASSO for variable selection methods for multiply-imputed datasets. The package includes four Bayesian MI-LASSO models using shrinkage (Multi-Laplace, Horseshoe, ARD) and Spike-and-Slab (Spike-and-Laplace) priors, along with tools for model fitting via MCMC, four-step projection predictive variable selection, and hyperparameter calibration. Methods are suitable for both continuous and binary covariates under missing-at-random or missing-completely-at-random assumptions. See Zou, J., Wang, S. and Chen, Q. (2025), Bayesian MI-LASSO for Variable Selection on Multiply-Imputed Data. ArXiv, 2211.00114. <[doi:10.48550/arXiv.2211.00114](https://doi.org/10.48550/arXiv.2211.00114)> for more details. We also provide the frequentist MI-LASSO function.

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

Imports Rcpp, MASS, Rfast, foreach, doParallel, parallel, arm, mice, abind, stringr, stats, posterior, loo

LinkingTo Rcpp, RcppArmadillo

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Author Jungang Zou [aut, cre],
Sijian Wang [aut],
Qixuan Chen [aut]

Repository CRAN

Date/Publication 2026-10-07 06:00:02 UTC

Contents

BMI_LASSO	2
MI_LASSO	4
sim_A	6
sim_B	7
sim_C	8
Index	10

BMI_LASSO

Bayesian MI-LASSO for Multiply-Imputed Regression

Description

Fit a Bayesian multiple-imputation LASSO (BMI-LASSO) model across multiply-imputed datasets, using one of four priors: Multi-Laplace, Horseshoe, ARD, or Spike-Laplace. Automatically standardizes data, runs MCMC in parallel, performs variable selection via four-step projection predictive variable selection, and selects a final submodel by BIC.

Usage

```
BMI_LASSO(
  X,
  Y,
  model,
  standardize = TRUE,
  search = "grid",
  grid = seq(0, 1, 0.01),
  orthogonal = FALSE,
  nburn = 5000,
  npost = 5000,
  seed = NULL,
  nchains = 1,
  ncores = 1,
  output_verbose = TRUE,
  printevery = 1000,
  selection_set = NULL,
  criterion = "bic",
  ...
)
```

Arguments

X A numeric matrix or array of predictors. If a matrix $n \times p$, it is taken as one imputation; if an array $D \times n \times p$, each slice along the first dimension is one imputed dataset.

<code>Y</code>	A numeric vector or matrix of outcomes. If a vector of length n , it is recycled for each imputation; if a $D \times n$ matrix, each row is the response for one imputation.
<code>model</code>	Character; which prior to use. One of "Multi_Laplace", "Horseshoe", "ARD", or "Spike_Laplace".
<code>standardize</code>	Logical; whether to normalize each X and centralize Y within each imputation before fitting. Default TRUE.
<code>search</code>	Character; how the candidate submodels along the path are generated. "grid" (default) applies a per-coefficient marginal rule over grid: for the shrinkage priors ("Multi_Laplace", "Horseshoe", "ARD") the symmetric credible interval that excludes 0, and for "Spike_Laplace" the posterior inclusion probability $p_j = \Pr(\gamma_j = 1 \mid y)$ at or above the grid value (0.5 gives the Barbieri & Berger (2004) median-probability model). "snc" uses the scaled-neighborhood criterion over grid. "forward" builds a nested path by forward stepwise search on the single-point projection loss (Piironen et al. 2020, Sec. 4).
<code>grid</code>	Numeric vector; thresholds explored by the "grid" and "snc" searches (ignored by "forward"). Default <code>seq(0, 1, 0.01)</code> .
<code>orthogonal</code>	Logical; if TRUE, using orthogonal approximations for degrees-of-freedom estimations. Default FALSE.
<code>nburn</code>	Integer; number of burn-in MCMC iterations per chain. Default 5000.
<code>npost</code>	Integer; number of post-burn-in samples to retain per chain. Default 5000.
<code>seed</code>	Optional integer; base random seed. Each chain adds its index. When supplied, BLAS is pinned to a single thread for the duration of the fit so results are bit-reproducible (multithreaded BLAS otherwise makes the MCMC draws, and hence the selection, non-deterministic).
<code>nchains</code>	Integer; number of MCMC chains to run in parallel. Default 1.
<code>ncores</code>	Integer; number of parallel cores to use. Default 1.
<code>output_verbose</code>	Logical; print progress messages. Default TRUE.
<code>printevery</code>	Integer; print status every so many iterations. Default 1000.
<code>selection_set</code>	Optional logical vector of length p . If supplied, the four-step candidate-generation + BIC search is bypassed and the fitted posterior is projected and calibrated directly onto this fixed set (<code>bic_models</code> is then NULL). Useful for post-selection inference under a pre-specified (e.g. the true) model. At most $n - 2$ variables may be selected (to keep the projection full rank and leave a residual degree of freedom for σ^2). Default NULL (run the search).
<code>criterion</code>	Character; model-size selection criterion along the candidate path. "bic" (default) uses the modified BIC. "loo-max" uses subject-level PSIS-LOO expected log predictive density (Piironen et al. 2020, Sec. 5.2) and selects the elpd-maximising subset. "loo-1se" applies the one-standard-error rule of Piironen et al. (2020) to the same PSIS-LOO path, selecting the smallest candidate within one standard error of the elpd-maximising one. "loo" is an alias for "loo-max".
<code>...</code>	Additional model-specific hyperparameters: • For "Multi_Laplace": h (shape) and v (scale) of Gamma hyperprior. • For "Spike_Laplace": a (shape) and b (scale) of Gamma hyperprior.

Value

A named list with elements:

`posterior` List of length `nchains` of MCMC outputs (posterior draws).

`select` List of length `nchains` of logical matrices showing which variables are selected at each grid value.

`best_select` List of length `nchains` of the single best selection (by BIC) for each chain.

`posterior_best_models` List of length `nchains` of projected posterior draws for the best sub-model.

`bic_models` List of length `nchains` of BIC values and degrees-of-freedom for each candidate sub-model.

`loo_models` (criterion other than "bic") The elpd path, its pointwise contributions, the max Pareto-k per candidate, the candidate sizes, the elpd-maximising index and the one-standard-error index. NULL otherwise.

`summary_table_full` A data frame summarizing rank-normalized split-Rhat and other diagnostics for the full model.

`summary_table_selected` A data frame summarizing diagnostics for the selected submodel after projection.

`pooled` (`nchains > 1` only; NULL otherwise) The multi-chain *pooled* four-step selection: all chains' draws are pooled and selected ONCE, so the chains yield a single coherent submodel rather than one selection per chain. A list with `best_select`, `select`, `bic_models` / `loo_models`, `posterior_best_models` (calibrated + projected draws), and `summary_table_selected` (pooled mean and 95\ unchanged).

Examples

```
sim <- sim_A(n = 100, p = 20, type = "MAR", SNP = 1.5, low_missing = TRUE, n_imp = 5, seed = 123)
X <- sim$data_MI$X
Y <- sim$data_MI$Y
fit <- BMI_LASSO(X, Y, model = "Horseshoe",
                 nburn = 100, npost = 100,
                 nchains = 1, ncores = 1)
str(fit$best_select)
```

MI_LASSO

Multiple-Imputation LASSO (MI-LASSO)

Description

Fit a LASSO-like penalty across D multiply-imputed datasets by iteratively reweighted ridge regressions (Equation (4) of the manuscript). For each tuning parameter in `lamvec`, it returns the pooled coefficient estimates, the BIC, and the selected variables.

Usage

```
MI_LASSO(
  X,
  Y,
  lamvec = (2^(seq(-1, 4, by = 0.05)))^2/2,
  maxiter = 200,
  eps = 1e-20,
  ncores = 1
)
```

Arguments

X	A matrix $n \times p$ or an array $D \times n \times p$ of imputed predictor sets. If a matrix is supplied, it is treated as a single imputation ($D = 1$).
Y	A vector length n or a $D \times n$ matrix of outcomes. If a vector, it is reused across imputations.
lamvec	Numeric vector of penalty parameters λ to search. Default $(2^{(\text{seq}(-1, 4, \text{by}=0.05)))^2/2}$.
maxiter	Integer; maximum number of ridge-update iterations per λ . Default 200.
eps	Numeric; convergence tolerance on coefficient change. Default $1e-20$.
ncores	Integer; number of cores for parallelizing over λ . Default 1.

Value

If $\text{length}(\text{lamvec}) > 1$, a list with elements:

best List for the *lambda* with minimal BIC containing: coefficients ($(p+1) \times D$ intercept + slopes), **bic** (BIC scalar), **varsel** (logical length- p vector of selected predictors), **lambda** (the chosen penalty).

lambda_path $\text{length}(\text{lamvec}) \times 2$ matrix of each λ and its corresponding BIC.

If $\text{length}(\text{lamvec}) == 1$, returns a single list (as above) for that penalty.

Examples

```
sim <- sim_A(n = 100, p = 20, type = "MAR", SNP = 1.5, low_missing = TRUE, n_imp = 5, seed = 123)
X <- sim$data_MI$X
Y <- sim$data_MI$Y
fit <- MI_LASSO(X, Y, lamvec = c(0.1))
```

sim_A	<i>Simulate dataset A: Independent continuous covariates with MCAR/MAR missingness</i>
-------	--

Description

Generates a dataset for Scenario A used in Bayesian MI-LASSO benchmarking. Covariates are iid standard normal, with a fixed true coefficient vector, linear outcome, missingness imposed on specified columns under MCAR or MAR, and multiple imputations via predictive mean matching.

Usage

```
sim_A(
  n = 100,
  p = 20,
  type = "MAR",
  SNP = 1.5,
  low_missing = TRUE,
  n_imp = 5,
  seed = NULL
)
```

Arguments

n	Integer. Number of observations.
p	Integer. Number of covariates (columns). Either 20, or 40 or more.
type	Character. Missingness mechanism: "MCAR" or "MAR".
SNP	Numeric. Signal-to-noise ratio controlling error variance.
low_missing	Logical. If TRUE, use low missingness rates; if FALSE, higher missingness. FALSE is available only for p = 20.
n_imp	Integer. Number of multiple imputations to generate.
seed	Integer or NULL. Random seed for reproducibility.

Value

A list with components:

data_O A list of complete covariate matrix and outcomes before missingness.

data_mis A list of covariate matrix and outcomes with missing values.

data_MI A list of array of imputed covariates ($n_{\text{imp}} \times n \times p$) and a matrix of imputed outcomes ($n_{\text{imp}} \times n$).

data_CC A list of complete-case covariate matrix and outcomes.

important Logical vector of true nonzero coefficient indices.

covmat True covariance matrix used for X.

beta True coefficient vector.

Examples

```
sim <- sim_A(n = 100, p = 20, type = "MAR", SNP = 1.5,
            low_missing = TRUE, n_imp = 5, seed = 123)
str(sim)
```

sim_B	<i>Simulate dataset B: AR(1)-correlated continuous covariates with MCAR/MAR missingness</i>
-------	---

Description

Generates a dataset for Scenario B used in Bayesian MI-LASSO benchmarking. Covariates are multivariate normal with AR(1) covariance, with a fixed true coefficient vector, linear outcome, missingness imposed on specified columns under MCAR or MAR, and multiple imputations via predictive mean matching.

Usage

```
sim_B(
  n = 100,
  p = 20,
  low_missing = TRUE,
  type = "MAR",
  SNP = 1.5,
  corr = 0.5,
  n_imp = 5,
  seed = NULL
)
```

Arguments

n	Integer. Number of observations.
p	Integer. Number of covariates (columns). Either 20, or 40 or more.
low_missing	Logical. If TRUE, use low missingness rates; if FALSE, higher missingness. FALSE is available only for p = 20.
type	Character. Missingness mechanism: "MCAR" or "MAR".
SNP	Numeric. Signal-to-noise ratio controlling error variance.
corr	Numeric. AR(1) correlation parameter
n_imp	Integer. Number of multiple imputations to generate.
seed	Integer or NULL. Random seed for reproducibility.

Value

A list with components:

data_O A list of complete covariate matrix and outcomes before missingness.

data_mis A list of covariate matrix and outcomes with missing values.

data_MI A list of array of imputed covariates ($n_{\text{imp}} \times n \times p$) and a matrix of imputed outcomes ($n_{\text{imp}} \times n$).

data_CC A list of complete-case covariate matrix and outcomes.

important Logical vector of true nonzero coefficient indices.

covmat True covariance matrix used for X.

beta True coefficient vector.

Examples

```
sim <- sim_B(n = 100, p = 20, type = "MAR", SNP = 1.5, corr = 0.5,
            low_missing = TRUE, n_imp = 5, seed = 123)
str(sim)
```

sim_C	<i>Simulate dataset C: AR(1)-latent Gaussian dichotomized to binary covariates with MCAR/MAR missingness</i>
-------	--

Description

Generates binary covariates by thresholding an AR(1) latent Gaussian, then proceeds as in sim_B.

Usage

```
sim_C(
  n = 100,
  p = 20,
  low_missing = TRUE,
  type = "MAR",
  SNP = 1.5,
  corr = 0.5,
  n_imp = 5,
  seed = NULL
)
```

Arguments

n	Integer. Number of observations.
p	Integer. Number of covariates (columns). Only $p = 20$ is available.
low_missing	Logical. If TRUE, use low missingness rates; if FALSE, higher missingness.
type	Character. Missingness mechanism: "MCAR" or "MAR".

SNP	Numeric. Signal-to-noise ratio controlling error variance.
corr	Numeric. AR(1) correlation parameter
n_imp	Integer. Number of multiple imputations to generate.
seed	Integer or NULL. Random seed for reproducibility.

Value

A list with components:

data_O A list of complete covariate matrix and outcomes before missingness.

data_mis A list of covariate matrix and outcomes with missing values.

data_MI A list of array of imputed covariates ($n_imp \times n \times p$) and a matrix of imputed outcomes ($n_imp \times n$).

data_CC A list of complete-case covariate matrix and outcomes.

important Logical vector of true nonzero coefficient indices.

covmat True covariance matrix used for X.

beta True coefficient vector.

Examples

```
sim <- sim_C(n = 100, p = 20, type = "MAR", SNP = 1.5, corr = 0.5,  
            low_missing = TRUE, n_imp = 5, seed = 123)  
str(sim)
```

Index

BMI_LASSO, [2](#)

MI_LASSO, [4](#)

sim_A, [6](#)

sim_B, [7](#)

sim_C, [8](#)