Package ‘hsstan’

September 22, 2019

Type Package

Title Hierarchical Shrinkage Stan Models for Biomarker Selection

Version 0.6

Date 2019-09-14

Description

License GPL-3

URL https://github.com/mcol/hsstan

BugReports https://github.com/mcol/hsstan/issues

LazyData yes

Biarch true

Depends R (>= 3.6)

Imports ggplot2, loo (>= 2.1.0), parallel, pROC, Rcpp, methods, rstan (> = 2.18.1), rstantools (>= 1.5.1), stats, utils

LinkingTo BH (>= 1.66.0.1), Rcpp (>= 0.12.15), RcppEigen (>= 0.3.3.4.0), StanHeaders (>= 2.17.2), rstan (>= 2.18.1)

Suggests testthat (>= 2.1.0)

ByteCompile yes

NeedsCompilation yes

Encoding UTF-8

RoxygenNote 6.1.1

Author Marco Colombo [aut, cre] (<https://orcid.org/0000-0001-6672-0623>), Paul McKeigue [aut] (<https://orcid.org/0000-0002-5217-1034>), Athina Spiliopoulou [ctb] (<https://orcid.org/0000-0002-5929-6585>)
Description

The **hsstan** package provides linear and logistic regression models penalized with hierarchical shrinkage priors for selection of biomarkers. Models are fitted with Stan (Carpenter et al. (2017)), which allows to perform full Bayesian inference.

Details

The package implements the horseshoe and regularized horseshoe priors (Piironen and Vehtari (2017)), and the projection predictive selection approach to recover a sparse set of predictive biomarkers (Piironen, Paasiniemi and Vehtari (2018)).

The approach is particularly suited to selection from high-dimensional panels of biomarkers, such as those that can be measured by MSMS or similar technologies (Colombo, Valo, McGurnaghan et al. (2019)).
Bayesian and LOO-adjusted R-squared

Description

Compute the Bayesian and the LOO-adjusted R-squared from the posterior samples. For Bayesian R-squared it uses the modelled residual variance (rather than the variance of the posterior distribution of the residuals). The LOO-adjusted R-squared uses Pareto smoothed importance sampling LOO residuals and Bayesian bootstrap.

Usage

```r
## S3 method for class 'hsstan'
bayes_R2(object, prob = 0.95, summary = TRUE, ...)

## S3 method for class 'hsstan'
loo_R2(object, prob = 0.95, summary = TRUE, ...)
```

Arguments

- `object`: An object of class `hsstan`.
- `prob`: Width of the posterior interval (0.95, by default). It is ignored if `summary=FALSE`.
- `summary`: Whether a summary of the distribution of the R-squared should be returned rather than the pointwise values (TRUE by default).
- `...`: Currently ignored.

Value

The mean, standard deviation and posterior interval of R-squared if `summary=TRUE`, or a vector of R-squared values with length equal to the size of the posterior sample if `summary=FALSE`.

References

B. Carpenter et al. (2017), Stan: a probabilistic programming language, *Journal of Statistical Software*, 76 (1). [https://doi.org/10.18637/jss.v076.i01](https://doi.org/10.18637/jss.v076.i01)


M. Colombo, E. Valo, S.J. McGurnaghan et al. (2019), Biomarkers associated with progression of renal disease in type 1 diabetes, *Diabetologia*, 62 (9), 1616-1627. [https://doi.org/10.1007/s00125-019-4915-0](https://doi.org/10.1007/s00125-019-4915-0)
References


Aki Vehtari, Andrew Gelman, Ben Goodrich and Jonah Gabry (2019), Bayesian R2 and LOO-R2. https://avehtari.github.io/bayes_R2/bayes_R2.html

Examples

```r
# continued from ?hsstan
bayes_R2(hs.biom)
loo_R2(hs.biom)
```

diabetes

*Diabetes data with interaction terms*

Description

The dataset consists of observations on 442 individuals for which a quantitative measure of diabetes progression is recorded in variable `Y`. Predictors include 10 baseline measurements, 45 interactions and 9 quadratic terms, for a total of 64 variables for each individual. Each variable has been standardized by subtracting the mean and then dividing it by its standard deviation.

Format

A data frame with 442 rows and 65 columns (centred and scaled).

Source


The original dataset is available from https://www.stanford.edu/~hastie/Papers/LARS/data64.txt

Examples

data(diabetes, package="hsstan")
hsstan

Hierarchical shrinkage models

Description

Run the No-U-Turn Sampler (NUTS) as implemented in Stan to fit a hierarchical shrinkage model.

Usage

hsstan(x, covs.model, penalized = NULL, family = gaussian,
iter = 2000, warmup = floor(iter/2), scale.u = 2,
regularized = TRUE, nu = ifelse(regularized, 1, 3),
par.ratio = 0.05, global.df = 1, slab.scale = 2, slab.df = 4,
qr = TRUE, seed = 123, adapt.delta = NULL, keep.hs.pars = FALSE,
...)

Arguments

x Data frame containing outcome, covariates and penalized predictors. Continuous predictors and outcome variable should be standardized before fitting the models as priors assume them to have mean zero and unit variance.

covs.model Formula containing the unpenalized covariates.

penalized Names of the variables to be used as penalized predictors. If NULL or an empty vector, a model with only unpenalized covariates is fitted.

family Type of model fitted: either gaussian() for linear regression (default) or binomial() for logistic regression.

iter Total number of iterations in each chain, including warmup (2000 by default).

warmup Number of warmup iterations per chain (by default, half the total number of iterations).

scale.u Prior scale (standard deviation) for the unpenalised covariates.

regularized If TRUE (default), the regularized horseshoe prior is used as opposed to the original horseshoe prior.

nu Number of degrees of freedom of the half-Student-t prior on the local shrinkage parameters (by default, 1 if regularized=TRUE and 3 otherwise).

par.ratio Expected ratio of non-zero to zero coefficients (ignored if regularized=FALSE). The scale of the global shrinkage parameter corresponds to par.ratio divided by the square root of the number of observations; for linear regression only, it’s further multiplied by the residual standard deviation sigma.

global.df Number of degrees of freedom for the global shrinkage parameter (ignored if regularized=FALSE).

slab.scale Scale of the regularization parameter (ignored if regularized=FALSE).

slab.df Number of degrees of freedom of the regularization parameter (ignored if regularized=FALSE).
Whether the thin QR decomposition should be used to decorrelate the predictors (TRUE by default). This is silently set to FALSE if there are more predictors than observations.

Optional integer defining the seed for the pseudo-random number generator.

Target average proposal acceptance probability for adaptation, a value between 0.8 and 1 (excluded). If unspecified, it’s set to 0.99 for hierarchical shrinkage models and to 0.95 for base models.

Whether the parameters for the horseshoe prior should be kept in the stanfit object returned (FALSE by default).

Further arguments passed to sampling, such as chains (4 by default), cores (the value of options("mc.cores") by default), refresh (iter / 10 by default).

An object of class hsstan containing the following fields:

- stanfit: an object of class stanfit containing the output produced by Stan, including posterior samples and diagnostic summaries. It can be manipulated using methods from the rstan package.
- betas: posterior means of the unpenalized and penalized regression parameters.
- call: the matched call.
- data: the dataset used in fitting the model.
- model.terms: a list of names for the outcome variable, the unpenalized covariates and the penalized predictors.
- family: the family object used.
- qr: Whether the QR factorization was performed.

See Also

kfold() for cross-validating a fitted object.

Examples

data(diabetes)

# non-default settings for speed of the example
df <- diabetes[1:50, ]
hs.biom <- hsstan(df, Y ~ age + sex, penalized=colnames(df)[5:10],
                 chains=2, iter=250)
**kfold.hsstan**

**K-fold cross-validation**

**Description**

Perform K-fold cross-validation using the same settings used when fitting the model on the whole data.

**Usage**

```r
## S3 method for class 'hsstan'
kfold(x, folds, chains = 1, store.fits = TRUE, cores = getOption("mc.cores", 1), ...)
```

**Arguments**

- `x`: An object of class `hsstan`.
- `folds`: Integer vector with one element per observation indicating the cross-validation fold in which the observation should be withdrawn.
- `chains`: Number of Markov chains to run. By default this is set to 1, independently of the number of chains used for `x`.
- `store.fits`: Whether the fitted models for each fold should be stored in the returned object (`TRUE` by default).
- `cores`: Number of cores to use for parallelization (the value of `options("mc.cores")` by default). The cross-validation folds will be distributed to the available cores, and the Markov chains for each model will be run sequentially.
- `...`: Further arguments passed to `sampling`.

**Value**

An object with classes `kfold` and `loo` that has a similar structure as the objects returned by `loo()` and `waic()` and is compatible with the `loo_compare` function for comparing models. The object contains the following fields:

- `estimates`: a matrix containing point estimates and standard errors of the expected log pointwise predictive density ("elpd_kfold"), the effective number of parameters ("p_kfold", always NA) and the K-fold information criterion "kfoldic" (which is \(-2 \times \text{elpd}_\text{kfold}\), i.e., converted to the deviance scale).
- `pointwise`: a matrix containing the pointwise contributions of "elpd_kfold", "p_kfold" and "kfoldic".
- `fits`: a matrix with two columns and number of rows equal to the number of cross-validation folds. Column `fit` contains the fitted `hsstan` objects for each fold, and column `test.idx` contains the indices of the withdrawn observations for each fold. This is not present if `store.fits=FALSE`.
- `data`: the dataset used in fitting the model (before withdrawing observations). This is not present if `store.fits=FALSE`. 
Examples

# continued from ?hsstan

# only 2 folds for speed of example
folds <- rep(1:2, length.out=length(df$Y))
cv.biom <- kfold(hs.biom, folds=folds, cores=2)

---

log_lik.hsstan  Pointwise log-likelihood

Description

Compute the pointwise log-likelihood.

Usage

## S3 method for class 'hsstan'

log_lik(object, newdata = NULL, ...)

Arguments

object  An object of class hsstan.

newdata  Optional data frame to use to evaluate the log-likelihood. If NULL (default), the model matrix is used.

...  Currently ignored.

Value

A matrix of size S by N, where S is number of draws from the posterior distribution, and N is the number of data points.

Examples

# continued from ?hsstan

log_lik(hs.biom)
Predictive information criteria for Bayesian models

Description

Compute an efficient approximate leave-one-out cross-validation using Pareto smoothed importance sampling (PSIS-LOO), or the widely applicable information criterion (WAIC), also known as the Watanabe-Akaike information criterion.

Usage

```r
## S3 method for class 'hsstan'
loo(x, cores = getOption("mc.cores"), ...)

## S3 method for class 'hsstan'
waic(x, cores = getOption("mc.cores"), ...)
```

Arguments

- `x` An object of class `hsstan`.
- `cores` Number of cores used for parallelisation (the value of `options("mc.cores")` by default).
- `...` Currently ignored.

Value

A `loo` object.

Examples

```r
# continued from ?hsstan

loo(hs.biom)
waic(hs.biom)
```
nsamples.hsstan  
Number of posterior samples

Description
Extracts the number of posterior samples stored in a fitted model.

Usage
## S3 method for class 'hsstan'
nsamples(object, ...)

Arguments
object  
An object of class hsstan.
...
Currently ignored.

Value
The total number of posterior samples across the chains after discarding the warmup iterations.

Examples

# continued from ?hsstan
nsamples(hs.biom)

plot.projsel  
Plot of relative explanatory power of predictors

Description
The function plots the relative explanatory power of each predictor in order of selection. The relative explanatory power of predictors is computed according to the KL divergence from the full model to each submodel, scaled in such a way that the baseline model (either the null model or the model containing only unpenalized covariates) is at 0, while the full model is at 1.

Usage
## S3 method for class 'projsel'
plot(x, title = NULL, max.points = NULL,
     max.labels = NULL, from.covariates = TRUE, font.size = 12,
     hadj = 0.05, vadj = 0, ...)
Arguments

- **x**: A data frame created by `projsel()`.
- **title**: Title of the plot. If `NULL`, no title is displayed.
- **max.points**: Maximum number of predictors to be plotted. If `NULL` (default) or 0, all points are plotted.
- **max.labels**: Maximum number of predictors to be labelled. If `NULL` (default), all predictor labels present in `x` are displayed, which may result in overprinting.
- **from.covariates**: Whether the plotting should start from the unpenalized covariates (`TRUE` by default). If set to `FALSE`, the plot includes a point for the null (intercept-only) model.
- **font.size**: Size of the textual elements (labels and axes).
- **hadj, vadj**: Horizontal and vertical adjustment for the labels.
- **...**: Currently ignored.

Value

A `ggplot2` object showing the relative incremental contribution of each predictor starting from the initial set of unpenalized covariates.

---

posterior_interval.hsstan

*Posterior uncertainty intervals*

---

Description

Compute posterior uncertainty intervals for `hsstan` objects.

Usage

```r
## S3 method for class 'hsstan'
posterior_interval(object, pars = NULL, prob = 0.95, 
...)
```

Arguments

- **object**: An object of class `hsstan`.
- **pars**: Names of parameters for which posterior intervals should be returned, which can be specified as regular expressions. If `NULL` (default) then this refers to the set of predictors used in the model.
- **prob**: A value between 0 and 1 indicating the desired probability to be covered by the uncertainty intervals (0.95, by default).
- **...**: Currently ignored.
posterior_linpred.hsstan

**Description**

Extract the posterior draws of the linear predictor, possibly transformed by the inverse-link function.

**Usage**

```r
## S3 method for class 'hsstan'
posterior_linpred(object, transform = FALSE,
                  newdata = NULL, ...)
```

**Arguments**

- `object` An object of class `hsstan`.
- `transform` Whether the linear predictor should be transformed using the inverse-link function (FALSE by default).
- `newdata` Optional data frame containing the variables to use to predict. If NULL (default), the model matrix is used. If specified, its continuous variables should be standardized, since the model coefficients are learnt on standardized data.
- `...` Currently ignored.

**Value**

A matrix of size $S \times N$, where $S$ is the number of draws from the posterior distribution of the (transformed) linear predictor, and $N$ is the number of data points.

**Examples**

```r
# continued from ?hsstan
posterior_linpred(hs.biom)
```
posterior_performance

Posterior measures of performance

Description
Compute the log-likelihood and a relevant measure of performance (R-squared or AUC) from the posterior samples.

Usage
posterior_performance(obj, prob = 0.95, summary = TRUE, cores = getOption("mc.cores", 1))

Arguments
- obj: An object of class hsstan or kfold.
- prob: Width of the posterior interval (0.95, by default). It is ignored if summary=FALSE.
- summary: Whether a summary of the distribution of the performance measure should be returned rather than the pointwise values (TRUE by default).
- cores: Number of cores to use for parallelization (the value of options("mc.cores") by default).

Value
The mean, standard deviation and posterior interval of the performance measure (R-squared or AUC) if summary=TRUE, or a vector of values of the performance measure with length equal to the size of the posterior sample if summary=FALSE. Attribute type reports whether the performance measures are cross-validated or not.

Examples
# continued from ?hsstan
posterior_performance(hs.biom, cores=1)

posterior_predict.hsstan

Posterior predictive distribution

Description
Draw from the posterior predictive distribution of the outcome.
posterior_summary

Usage

## S3 method for class 'hsstan'
posterior_predict(object, newdata = NULL,
                 nsamples = NULL, seed = NULL, ...)

Arguments

object | An object of class hsstan.
newdata | Optional data frame containing the variables to use to predict. If NULL (default), the model matrix is used. If specified, its continuous variables should be standardized, since the model coefficients are learnt on standardized data.
nsamples | A positive integer indicating the number of posterior samples to use. If NULL (default) all samples are used.
seed | Optional integer defining the seed for the pseudo-random number generator.
... | Currently ignored.

Value

A matrix of size S by N, where S is the number of simulations from the posterior predictive distribution, and N is the number of data points.

Examples

# continued from ?hsstan
posterior_predict(hs.biom)

---

posterior_summary | Posterior summary

Description

Produce a summary of the posterior samples for the quantities of interest.

Usage

posterior_summary(x, prob, ...)

## Default S3 method:
posterior_summary(x, prob = 0.95, ...)

## S3 method for class 'hsstan'
posterior_summary(x, prob = 0.95, pars = NULL, ...)

---

Usage

## S3 method for class 'hsstan'
posterior_predict(object, newdata = NULL,
                 nsamples = NULL, seed = NULL, ...)

Arguments

object | An object of class hsstan.
newdata | Optional data frame containing the variables to use to predict. If NULL (default), the model matrix is used. If specified, its continuous variables should be standardized, since the model coefficients are learnt on standardized data.
nsamples | A positive integer indicating the number of posterior samples to use. If NULL (default) all samples are used.
seed | Optional integer defining the seed for the pseudo-random number generator.
... | Currently ignored.

Value

A matrix of size S by N, where S is the number of simulations from the posterior predictive distribution, and N is the number of data points.

Examples

# continued from ?hsstan
posterior_predict(hs.biom)
Arguments

- **x** An object containing or representing posterior samples. If x is a matrix, it should have size S by Q, where S is the number of posterior samples, and Q is the number of quantities of interest.
- **prob** Width of the posterior intervals (0.95, by default).
- **...** Further arguments passed to or from other methods.
- **pars** Vector of parameter names to be extracted. If NULL (default) then this refers to the set of predictors used in the model.

Value

A matrix with columns containing mean, standard deviation and posterior intervals for the given quantities.

See Also

summary() to produce summaries of hsstan objects that include the number of effective samples and the split-Rhat diagnostic.

Examples

```r
# continued from ?hsstan
posterior_summary(hs.biom)
```

print.hsstan

Print a summary for the fitted model

Description

Print a summary for the fitted model

Usage

```r
## S3 method for class 'hsstan'
print(x, ...)
```

Arguments

- **x** An object of class hsstan.
- **...** Further arguments to summary().
**projsel**

*Forward selection minimizing KL-divergence in projection*

**Description**

Forward selection minimizing KL-divergence in projection

**Usage**

`projsel(obj, max.iters = 30, out.csv = NULL)`

**Arguments**

- `obj` Object of class `hsstan`.
- `max.iters` Maximum number of iterations (number of predictors selected) after which the selection procedure should stop.
- `out.csv` If not NULL, the name of a CSV file to save the output to.

**Value**

A data frame of class `projsel` where each row corresponds to a forward-selected submodel that contains all variables listed up to that row. The data frame contains the following columns:

- `var` names of the variables selected.
- `kl` KL-divergence from the full model to the submodel.
- `rel.kl.null` relative explanatory power of predictors starting from the intercept-only model.
- `rel.kl` relative explanatory power of predictors starting from the model containing un-penalized covariates.
- `elpd` the expected log predictive density of the submodels.
- `delta.elpd` the difference in elpd from the full model.

**Examples**

```R
# continued from ?hsstan
sel <- projsel(hs.biom, max.iters=3)
plot(sel)
```
sampler.stats

---

### Description

Report statistics on the parameters used in the sampler, the sampler behaviour and the sampling time.

### Usage

```r
sampler.stats(object)
```

### Arguments

- `object`  
  An object of class `hsstan`.

### Value

A matrix with \( C + 1 \) rows, where \( C \) is the number of Markov chains, reporting average acceptance probability, average stepsize, number of divergent transitions, maximum tree depth, total number of gradient evaluations, warmup and sampling times in seconds.

### Examples

```r
# continued from ?hsstan
sampler.stats(hs.biom)
```

---

summary.hsstan

---

### Description

Summary for the fitted model

### Usage

```r
## S3 method for class 'hsstan'
summary(object, pars = NULL, prob = 0.95, 
  digits = 2, sort = NULL, decreasing = TRUE, max.rows = NULL, ...)
```
Arguments

- **object**: An object of class `hsstan`.
- **pars**: Vector of parameter names to be extracted. If `NULL` (default) then this refers to the set of predictors used in the model.
- **prob**: Width of the posterior intervals (0.95, by default).
- **digits**: Number of decimal places to be reported (2 by default).
- **sort**: Column name used to sort the results according to the absolute value of the column. If `NULL` (default) or the column name cannot be found, no sorting occurs.
- **decreasing**: Whether the results should be sorted in decreasing order when a valid column name is specified in `sort` (TRUE by default).
- **max.rows**: Maximum number of rows to be returned. If `NULL` (default) or 0, all results are returned.
- **...**: Currently ignored.

Value

A matrix with summaries from the posterior distribution of the parameters of interest.

Examples

```r
# continued from ?hsstan
summary(hs.stan)
summary(hs.biom)
```
Index

*Topic datasets
  diabetes, 4

bayes_R2 (bayes_R2.hsstan), 3
bayes_R2.hsstan, 3
diabetes, 4
hsstan, 5
hsstan-package, 2

kfold (kfold.hsstan), 7
kfold(), 6
kfold.hsstan, 7

log_lik (log_lik.hsstan), 8
log_lik.hsstan, 8
loo (loo.hsstan), 9
loo(), 7
loo.hsstan, 9
loo_compare, 7
loo_R2 (bayes_R2.hsstan), 3

nsamples (nsamples hsstan), 10
nsamples.hsstan, 10

plot.projsel, 10
posterior_interval
  (posterior_interval.hsstan), 11
posterior_interval.hsstan, 11
posterior_linpred
  (posterior_linpred.hsstan), 12
posterior_linpred.hsstan, 12
posterior_performance, 13
posterior_predict
  (posterior_predict.hsstan), 13
posterior_predict.hsstan, 13
posterior_summary, 14
print.hsstan, 15
projsel, 16
projsel(), 11

sampler.stats, 17
sampling, 6, 7
summary(), 15
summary.hsstan, 17

waic (loo.hsstan), 9
waic(), 7

19