Package: bolasso (via r-universe)

December 9, 2024

Title Model Consistent Lasso Estimation Through the Bootstrap

Version 0.3.0

Description Implements the bolasso algorithm for consistent variable selection and estimation accuracy. Includes support for many parallel backends via the future package. For details see: Bach (2008), 'Bolasso: model consistent Lasso estimation through the bootstrap', <doi:10.48550/arXiv.0804.1302>.

Depends Matrix (>= 1.0-6), R (>= 3.6.0)

Imports future.apply (>= 1.1.0), gamlr (>= 1.0), generics, ggplot2, glmnet (>= 3.0), progressr, stats, tibble

Suggests testthat (>= 3.0.0), mlbench, covr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

URL https://www.dmolitor.com/bolasso/, https://github.com/dmolitor/bolasso

Config/testthat/edition 3

BugReports https://github.com/dmolitor/bolasso/issues

NeedsCompilation no

Author Daniel Molitor [aut, cre]

Maintainer Daniel Molitor <molitdj97@gmail.com>

Repository CRAN

Date/Publication 2024-12-08 22:20:12 UTC

Contents

`bolasso` *Bootsrap-enhanced Lasso*

Description

This function implements model-consistent Lasso estimation through the bootstrap. It supports parallel processing by way of the future package, allowing the user to flexibly specify many parallelization methods. This method was developed as a variable-selection algorithm, but this package also supports making ensemble predictions on new data using the bagged Lasso models.

Usage

```
bolasso(
  formula,
  data,
  n.boot = 100,
  progress = TRUE,
  implement = c("glmnet", "gamlr"),
  x = NULL,
  y = NULL,
  fast = FALSE,
  ...
)
```

Arguments

`formula`	An optional object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted. Can be omitted when `x` and `y` are non-missing.
`data`	An optional object of class data.frame that contains the modeling variables referenced in `form`. Can be omitted when `x` and `y` are non-missing.
`n.boot`	An integer specifying the number of bootstrap replicates.
`progress`	A boolean indicating whether to display progress across bootstrap folds.
`implement`	A character; either 'glmnet' or 'gamlr', specifying which Lasso implementation to utilize. For specific modeling details, see `glmnet::cv.glmnet` or `gamlr::cv.gamlr`.
`x`	An optional predictor matrix in lieu of `form` and `data`.

y	An optional response vector in lieu of form and data.
fast	A boolean. Whether or not to fit a "fast" bootstrap procedure. If fast == TRUE, bolasso will fit glmnet::cv.glmnet on the entire dataset. It will then fit all bootstrapped models with the value of lambda (regularization parameter) that minimized cross-validation loss in the full model. If fast == FALSE (the default), bolasso will use cross-validation to find the optimal lambda for each bootstrap model.
...	Additional parameters to pass to either glmnet::cv.glmnet or gamlr::cv.gamlr.

Value

An object of class bolasso. This object is a list of length n.boot of cv.glmnet or cv.gamlr objects.

See Also

glmnet::cv.glmnet and gamlr::cv.gamlr for full details on the respective implementations and arguments that can be passed to ....

Examples

```
mtcars[, c(2, 10:11)] <- lapply(mtcars[, c(2, 10:11)], as.factor)
idx <- sample(nrow(mtcars), 22)
mtcars_train <- mtcars[idx, ]
mtcars_test <- mtcars[-idx, ]

## Formula Interface

# Train model
set.seed(123)
bolasso_form <- bolasso(
  form = mpg ~ .,
  data = mtcars_train,
  n.boot = 20,
  nfolds = 5
)

# Retrieve a tidy tibble of bootstrap coefficients for each covariate
tidy(bolasso_form)

# Extract selected variables
selected_variables(bolasso_form, threshold = 0.9, select = "lambda.min")

# Bagged ensemble prediction on test data
predict(bolasso_form,
        new.data = mtcars_test,
        select = "lambda.min")

## Alternate Matrix Interface

# Train model
```

```
set.seed(123)
bolasso_mat <- bolasso(
  x = model.matrix(mpg ~ . - 1, mtcars_train),
  y = mtcars_train[, 1],
  data = mtcars_train,
  n.boot = 20,
  nfolds = 5
)

# Bagged ensemble prediction on test data
predict(bolasso_mat,
        new.data = model.matrix(mpg ~ . - 1, mtcars_test),
        select = "lambda.min")
```

`plot.bolasso` *Plot a* `bolasso` *object*

Description

The method plots coefficient distributions for the covariates included in the `bolasso` model. If there are more than 30 covariates included in the full model, this will plot the 30 covariates with the largest absolute mean coefficient. The user can also plot coefficient distributions for a specified subset of covariates.

Usage

```
## S3 method for class 'bolasso'
plot(x, covariates = NULL, ...)
```

Arguments

`x`	An object of class bolasso or `bolasso_fast`.
`covariates`	A subset of the covariates to plot. This should be a vector of covariate names either as strings or bare. E.g. `covariates = c("var_1", "var_2")` or `covariates = c(var_1, var_2)`. This argument is optional and is `NULL` by default. In this case it will plot up to 30 covariates with the largest absolute mean coefficients.
`...`	Additional arguments to pass directly to `coef` for objects of class bolasso or `bolasso_fast`.

plot_selected_variables

Plot selected variables from a `bolasso` *object.*

Description

The method plots coefficient distributions for the selected covariates in the `bolasso` model. If there are more than 30 selected covariates, this will plot the 30 selected covariates with the largest absolute mean coefficient. The user can also plot coefficient distributions for a specified subset of selected covariates.

Usage

```
plot_selected_variables(
  x,
  covariates = NULL,
  threshold = 0.95,
  method = c("vip", "qnt"),
  ...
)
```

Arguments

`x`	An object of class bolasso or `bolasso_fast`.
`covariates`	A subset of the selected covariates to plot. This should be a vector of covariate names either as strings or bare. E.g. `covariates = c("var_1", "var_2")` or `covariates = c(var_1, var_2)`. This argument is optional and is `NULL` by default. In this case it will plot up to 30 covariates with the largest absolute mean coefficients.
`threshold`	A numeric between 0 and 1, specifying the variable selection threshold to use.
`method`	The variable selection method to use. The two valid options are `c("vip", "qnt")`. The default `"vip"` and is the method described in the original Bach (2008) and complementary Bunea et al. (2011) works. The `"qnt"` method is the method proposed by Abram et al. (2016).
`...`	Additional arguments to pass to `coef` on objects with class `bolasso` or `bolass_fast`.

plot_selection_thresholds

Plot each covariate's smallest variable selection threshold

Description

Plot the results of the selection_thresholds function.

Usage

```
plot_selection_thresholds(object = NULL, data = NULL, ...)
```

Arguments

object	An object of class bolasso or bolasso_fast. This argument is optional if you directly pass in the data via the data argument. E.g. data = selection_thresholds(object).
data	A dataframe containing the selection thresholds. E.g. obtained via selection_thresholds(object). This argument is optional if you directly pass a bolasso or bolasso_fast object via the object argument.
...	Additional arguments to pass directly to selection_thresholds.

Value

A ggplot object

See Also

selection_thresholds()

---

selected_variables *Bolasso-selected Variables*

Description

Identifies covariates that are selected by the Bolasso algorithm at the user-defined threshold. There are two variable selection criterion to choose between; Variable Inclusion Probability ("vip") introduced in the original Bolasso paper (Bach, 2008) and further developed by Bunea et al. (2011), and the Quantile ("qnt") approach proposed by Abram et al. (2016). The desired threshold value is 1 - alpha, where alpha is some (typically small) significance level.

Usage

```
selected_variables(
  object,
  threshold = 0.95,
  method = c("vip", "qnt"),
  var_names_only = FALSE,
  ...
)
```

Arguments

`object`	An object of class bolasso.
`threshold`	A numeric between 0 and 1, specifying the variable selection threshold to use.
`method`	The variable selection method to use. The two valid options are `c("vip", "qnt")`. The default `"vip"` and is the method described in the original Bach (2008) and complementary Bunea et al. (2011) works. The `"qnt"` method is the method proposed by Abram et al. (2016).
`var_names_only`	A boolean value. When `var_names_only = FALSE` (the default value) this function will return a tibble::tibble of selected covariates and their corresponding coefficients across all bootstrap replicates. When `var_names_only == TRUE`, it will return a vector containing all selected covariate names.
`...`	Additional arguments to pass to `coef` on objects with class bolasso or `bolass_fast`.

Details

This function returns either a tibble::tibble of selected covariates and their corresponding coefficients across all bootstrap replicates, or a vector of selected covariate names.

Value

A tibble with each selected variable and its respective coefficient for each bootstrap replicate OR a vector of the names of all selected variables.

See Also

`glmnet::coef.glmnet()` and `gamlr:::coef.gamlr` for details on additional arguments to pass to `...`.

`selection_thresholds` *Calculate each covariate's smallest variable selection threshold*

Description

There are two methods of variable selection for covariates. The first is the Variable Inclusion Probability (VIP) introduced by Bach (2008) and generalized by Bunea et al (2011). The second is the Quantile confidence interval (QNT) proposed by Abram et al (2016). For a given level of significance alpha, each method selects covariates for the given threshold = 1 - alpha. The higher the threshold (lower alpha), the more stringent the variable selection criterion.

Usage

```
selection_thresholds(object, grid = seq(0, 1, by = 0.01), ...)
```

Arguments

`object`	An object of class bolasso or `bolasso_fast`.
`grid`	A vector of numbers between 0 and 1 (inclusive) specifying the grid of threshold values to calculate variable inclusion criterion at. Defaults to `seq(0, 1, by = 0.01)`.
`...`	Additional parameters to pass to `coef` on objects of class bolasso and `bolasso_fast`.

Details

This function returns a tibble that, for each covariate, returns the largest threshold (equivalently smallest alpha) at which it would be selected for both the VIP and the QNT methods. Consequently the number of rows in the returned tibble is 2*p where p is the number of covariates included in the model.

Value

A tibble with dimension (2*p)x5 where p is the number of covariates.

`tidy.bolasso` *Tidy a bolasso object*

Description

Tidy a bolasso object

Usage

```
## S3 method for class 'bolasso'
tidy(x, select = c("lambda.min", "lambda.1se", "min", "1se"), ...)
```

Arguments

`x`	A `bolasso` object.
`select`	One of "min", "1se", "lambda.min", "lambda.1se". Both "min" and "lambda.min" are equivalent and are the lambda value that minimizes cv MSE. Similarly "1se" and "lambda.1se" are equivalent and refer to the lambda that achieves the most regularization and is within 1se of the minimal cv MSE.
`...`	Additional arguments to pass directly to `coef.bolasso`.

Value

A tidy `tibble::tibble()` summarizing bootstrap-level coefficients for each covariate.

`transactions` *Customer transaction data*

Description

Predict whether customers will make a specific transaction based on a rich set of user features.

Usage

```
transactions
```

Format

Dataframe with columns

target An integer indicating whether a customer engaged in a transaction.

var_i 200 numeric features of various customer characteristics.

Index