Package: penalizedclr (via r-universe)

August 27, 2024

Title Integrative Penalized Conditional Logistic Regression

Version 2.0.0

Description Implements L1 and L2 penalized conditional logistic regression with penalty factors allowing for integration of multiple data sources. Implements stability selection for variable selection.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports penalized, survival, clogitL1, stats

Suggests parallel, knitr, rmarkdown, tidyverse

VignetteBuilder knitr

Repository https://veradjordjilovic.r-universe.dev

RemoteUrl https://github.com/veradjordjilovic/penalizedclr

RemoteRef HEAD

RemoteSha 0faf0569fcd88324d31fac5964ec268a7feaec8f

Contents

`default.lambda` *Default values for L1 penalty in conditional logistic regression*

Description

Internal function that performs cross validation to determine reasonable default values for L1 penalty in a conditional logistic regression

Usage

```
default.lambda(X, Y, stratum, nfolds = 10, alpha = 1)
```

Arguments

`X`	A matrix of covariates, with the number of rows equaling the number of observations.
`Y`	A binary response variable.
`stratum`	A numeric vector with stratum membership of each observation.
`nfolds`	The number of folds used in cross-validation. Default is 10.
`alpha`	The elastic net mixing parameter, a number between 0 and 1. alpha=0 would give pure ridge; alpha=1 gives lasso. Pure ridge penalty is never obtained in this implementation since alpha must be positive.

Value

A numeric value for `lambda` minimizing cross validated deviance.

`default.pf` *Data adaptive candidate vector of penalty factors for L1 penalty in conditional logistic regression with covariates divided in blocks*

Description

Computes a data adaptive vector of penalty factors for blocks of covariates by fitting a tentative penalized conditional logistic regression model. The penalty for the `i`th block is obtained as the inverse of the arithmetic mean of coefficient estimates for its covariates.

Usage

```
default.pf(
  response,
  stratum,
  penalized,
  unpenalized = NULL,
  alpha = 1,
  p = NULL,
  standardize = TRUE,
  event,
  nfolds = 10,
  type.step1,
  verbose = F
)
```

Arguments

response	The response variable, either a 0/1 vector or a factor with two levels.
stratum	A numeric vector with stratum membership of each observation.
penalized	A matrix of penalized covariates.
unpenalized	A matrix of additional unpenalized covariates.
alpha	The elastic net mixing parameter, a number between 0 and 1. alpha=0 would give pure ridge; alpha=1 gives lasso. Pure ridge penalty is never obtained in this implementation since alpha must be positive.
p	The sizes of blocks of covariates, a numerical vector of the length equal to the number of blocks, and with the sum equal to the number of penalized covariates. If missing, all covariates are treated the same and a single penalty is applied.
standardize	Should the covariates be standardized, a logical value.
event	If response is a factor, the level that should be considered a success in the logistic regression.
nfolds	The number of folds used in cross-validation. Default is 10.
type.step1	Should the tentative model be fit on all covariates jointly (comb) or to each block separately (sep).
verbose	Logical. Should the message about the obtained penalty factors be printed?

Details

Blocks that contain covariates with large estimated coefficients will obtain a smaller penalty. If all estimated coefficients pertaining to a block are zero, the function returns a message. A tentative conditional logistic regression model is fit either to each covariates block separately (type.step1 = "sep") or jointly to all blocks (type.step1 = "comb"). Note that unpenalized = NULL is the only implemented option in this function as of now.

Value

The function returns a list containing the vector of penalty factors correspondng to different blocks.

References

Schulze G. (2017) Clinical Outcome Prediction based on Multi-Omics Data: Extension of IPF-LASSO. Master Thesis.

See Also

find.default.lambda

`find.default.lambda` *Default values for L1 penalty in conditional logistic regression*

Description

Performs cross validation to determine reasonable values for L1 penalty in a conditional logistic regression.

Usage

```
find.default.lambda(
  response,
  stratum,
  penalized,
  unpenalized = NULL,
  alpha = 1,
  p = NULL,
  standardize = TRUE,
  event,
  pf.list = NULL,
  nfolds = 10
)
```

Arguments

`response`	The response variable, either a 0/1 vector or a factor with two levels.
`stratum`	A numeric vector with stratum membership of each observation.
`penalized`	A matrix of penalized covariates.
`unpenalized`	A matrix of additional unpenalized covariates.
`alpha`	The elastic net mixing parameter, a number between 0 and 1. alpha=0 would give pure ridge; alpha=1 gives lasso. Pure ridge penalty is never obtained in this implementation since alpha must be positive.
`p`	The sizes of blocks of covariates, a numerical vector of the length equal to the number of blocks, and with the sum equal to the number of penalized covariates. If missing, all covariates are treated the same and a single penalty is applied.
`standardize`	Should the covariates be standardized, a logical value.

event	If response is a factor, the level that should be considered a success in the logistic regression.
pf.list	List of vectors of penalty factors.
nfolds	The number of folds used in cross-validation. Default is 10.

Details

The function is based on cross-validation implemented in the clogitL1 package and returns the value of lambda that minimizes cross validated deviance. In the presence of blocks of covariates, a user specifies a list of candidate vectors of penalty factors. For each candidate vector of penalty factors a single lambda value is obtained. Note that cross-validation includes random data splitting, meaning that obtained values can vary significantly between different runs.

Value

A single numeric value if p and pf.list are missing, or a list of numeric values with L1 penalties for each vector of penalty factors supplied.

See Also

default.pf

Examples

```
set.seed(123)
# simulate covariates (pure noise in two blocks of 20 and 80 variables)
X <- cbind(matrix(rnorm(4000, 0, 1), ncol = 20), matrix(rnorm(16000, 2, 0.6), ncol = 80))
p <- c(20,80)
pf.list <- list(c(0.5, 1), c(2, 0.9))
# stratum membership
stratum <- sort(rep(1:100, 2))

# the response
Y <- rep(c(1, 0), 100)

# obtain a list with vectors of penalty factors

lambda.list <- find.default.lambda(response = Y,
                                penalized = X, stratum = stratum, p = p, pf.list = pf.list)

# when `p` and `pf.list` are not provided all covariates are treated as a single block

lambda <- find.default.lambda(response = Y,
                                    penalized = X, stratum = stratum)
```

`penalized.clr` *Penalized conditional logistic regression*

Description

Fits conditional logistic regression models with L1 and L2 penalty allowing for different penalties for different blocks of covariates.

Usage

```
penalized.clr(
  response,
  stratum,
  penalized,
  unpenalized = NULL,
  lambda,
  alpha = 1,
  p = NULL,
  standardize = TRUE,
  event
)
```

Arguments

`response`	The response variable, either a 0/1 vector or a factor with two levels.
`stratum`	A numeric vector with stratum membership of each observation.
`penalized`	A matrix of penalized covariates.
`unpenalized`	A matrix of additional unpenalized covariates.
`lambda`	The tuning parameter for L1. Either a single non-negative number, or a numeric vector of the length equal to the number of blocks. See p below.
`alpha`	The elastic net mixing parameter, a number between 0 and 1. alpha=0 would give pure ridge; alpha=1 gives lasso. Pure ridge penalty is never obtained in this implementation since alpha must be positive.
`p`	The sizes of blocks of covariates, a numerical vector of the length equal to the number of blocks, and with the sum equal to the number of penalized covariates. If missing, all covariates are treated the same and a single penalty is applied.
`standardize`	Should the covariates be standardized, a logical value.
`event`	If response is a factor, the level that should be considered a success in the logistic regression.

Details

The `penalized.clr` function fits a conditional logistic regression model for a given combination of L1 (`lambda`) and L2 penalties. L2 penalty is obtained from `lambda` and `alpha` as `lambda*(1-alpha)/(2*alpha)`. Note that `lambda` is a single number if all covariates are to be penalized equally, and a vector of

penatlies, if predictors are divided in blocks (of sizes provided in p) that are to be penalized differently. The penalized.clr function is based on the Cox model routine available in the penalized package.

Value

A list with the following elements:

- penalized - Regression coefficients for the penalized covariates.
- unpenalized - Regression coefficients for the unpenalized covariates.
- converged - Whether the fitting process was judged to have converged.
- lambda - The tuning parameter for L1 used.
- alpha - The elastic net mixing parameter used.

See Also

stable.clr and stable.clr.g for variable selection through stability selection in penalized conditional logistic regression with a single penalty factor and multiple penalty factors, respectively.

Examples

```
set.seed(123)
# simulate covariates (pure noise in two blocks of 20 and 80 variables)
X <- cbind(matrix(rnorm(4000, 0, 1), ncol = 20), matrix(rnorm(16000, 2, 0.6), ncol = 80))

# stratum membership
stratum <- sort(rep(1:100, 2))

# the response
Y <- rep(c(1, 0), 100)

fit <- penalized.clr( response = Y, stratum = stratum,
  penalized = X, lambda = c(1, 0.3),
  p = c(20, 80), standardize = TRUE)
fit$penalized
fit$converged
fit$lambda
```

---

stable.clr *Stability selection based on penalized conditional logistic regression*

Description

Performs stability selection for conditional logistic regression models with L1 and L2 penalty.

Usage

```
stable.clr(
  response,
  stratum,
  penalized,
  unpenalized = NULL,
  lambda.seq,
  alpha = 1,
  B = 100,
  parallel = TRUE,
  standardize = TRUE,
  event
)
```

Arguments

response	The response variable, either a 0/1 vector or a factor with two levels.
stratum	A numeric vector with stratum membership of each observation.
penalized	A matrix of penalized covariates.
unpenalized	A matrix of additional unpenalized covariates.
lambda.seq	a sequence of non-negative value to be used as tuning parameter for L1
alpha	The elastic net mixing parameter, a number between 0 and 1. alpha=0 would give pure ridge; alpha=1 gives lasso. Pure ridge penalty is never obtained in this implementation since alpha must be positive.
B	A single positive number for the number of subsamples.
parallel	Logical. Should the computation be parallelized?
standardize	Should the covariates be standardized, a logical value.
event	If response is a factor, the level that should be considered a success in the logistic regression.

Value

A list with a numeric vector `Pistab` giving selection probabilities for each penalized covariate, and a sequence `lambda.seq` used.

See Also

`stable.clr.g` for stability selection in penalized conditional logistic regression with multiple penalties for block structured covariates.

Examples

```
 set.seed(123)

# simulate covariates (pure noise in two blocks of 20 and 80 variables)
X <- cbind(matrix(rnorm(4000, 0, 1), ncol = 20), matrix(rnorm(16000, 2, 0.6), ncol = 80))
```

```
# stratum membership
stratum <- sort(rep(1:100, 2))

# the response
Y <- rep(c(1, 0), 100)

# default L1 penalty
lambda <- find.default.lambda(response = Y,
                                    penalized = X,
                                    stratum = stratum)

# perform stability selection

stable1 <- stable.clr(response = Y, penalized = X, stratum = stratum,
                          lambda.seq = lambda)
```

`stable.clr.g` *Stability selection based on penalized conditional logistic regression*

Description

Performs stability selection for conditional logistic regression models with L1 and L2 penalty allowing for different penalties for different blocks (groups) of covariates (different data sources).

Usage

```
stable.clr.g(
  response,
  stratum,
  penalized,
  unpenalized = NULL,
  p = NULL,
  lambda.list,
  alpha = 1,
  B = 100,
  parallel = TRUE,
  standardize = TRUE,
  event
)
```

Arguments

`response`	The response variable, either a 0/1 vector or a factor with two levels.
`stratum`	A numeric vector with stratum membership of each observation.

penalized	A matrix of penalized covariates.
unpenalized	A matrix of additional unpenalized covariates.
p	The sizes of blocks of covariates, a numerical vector of the length equal to the number of blocks, and with the sum equal to the number of penalized covariates. If missing, all covariates are treated the same and a single penalty is applied.
lambda.list	A list of vectors of penalties to be applied to different blocks of covariates. Each vector should have the length equal to the number of blocks.
alpha	The elastic net mixing parameter, a number between 0 and 1. alpha=0 would give pure ridge; alpha=1 gives lasso. Pure ridge penalty is never obtained in this implementation since alpha must be positive.
B	A single positive number for the number of subsamples.
parallel	Logical. Should the computation be parallelized?
standardize	Should the covariates be standardized, a logical value.
event	If response is a factor, the level that should be considered a success in the logistic regression.

Details

This function implements stability selection (Meinshausen and Bühlmann, 2010) in a conditional logistic regression. The implementation is based on the modification of Shah and Samworth (2013) featuring complementary subsamples. Note that this means that the number of subsamples will be 2B instead of B. Subsampling procedure is repeated 2B times for each vector of per-block penalties resulting each time in a vector of selection frequencies (frequency of non-zero coefficient estimate of each covariate). The final selection probability Pistab is obtained by taking the maximum over all considered vectors of penalties.

Value

A list containing a numeric vector Pistab, giving selection probabilities for all penalized covariates, lambda.list and p provided as input arguments.

References

1. Meinshausen, N., & Bühlmann, P. (2010). Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(4), 417-473.
2. Shah, R. D., & Samworth, R. J. (2013). Variable selection with error control: another look at stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 75(1), 55-80.

Examples

```
set.seed(123)

# simulate covariates (pure noise in two blocks of 20 and 80 variables)
X <- cbind(matrix(rnorm(4000, 0, 1), ncol = 20), matrix(rnorm(16000, 2, 0.6), ncol = 80))
p <- c(20,80)
```

```
# stratum membership
stratum <- sort(rep(1:100, 2))

# the response
Y <- rep(c(1, 0), 100)

# list of L1 penalties

lambda.list = list(c(0.5,1), c(2,0.9))

# perform stability selection

stable.g1 <- stable.clr.g(response = Y, penalized = X, stratum = stratum,
                          p = p, lambda.list = lambda.list)
```

`subsample.clr` *Stability selection for penalized conditional logistic regression*

Description

Internal function used by `stable.clr` and `stable.clr.g`.

Usage

```
subsample.clr(
  response,
  stratum,
  penalized,
  unpenalized = NULL,
  lambda,
  alpha = 1,
  B = 100,
  matB = NULL,
  return.matB = FALSE,
  parallel = TRUE,
  standardize = TRUE
)
```

Arguments

`response`	The response variable, either a 0/1 vector or a factor with two levels.
`stratum`	A numeric vector with stratum membership of each observation.
`penalized`	A matrix of penalized covariates.
`unpenalized`	A matrix of additional unpenalized covariates.
`lambda`	The tuning parameter for L1. Either a single non-negative number, or a numeric vector of the length equal to the number of blocks. See p below.

alpha	The elastic net mixing parameter, a number between 0 and 1. alpha=0 would give pure ridge; alpha=1 gives lasso. Pure ridge penalty is never obtained in this implementation since alpha must be positive.
B	A single positive number for the number of subsamples.
matB	A 2B x ceiling(unique(stratum)/2) matrix with index set of selected strata in each of 2B subsamples
return.matB	Logical. Should the matrix matB be returned?
parallel	Logical. Should the computation be parallelized?
standardize	Should the covariates be standardized, a logical value.

Value

If return.matB is TRUE, a list with two elements, a numeric vector Pistab, giving selection probabilities for each covariate and a matrix matB; otheriwise only Pistab.

Index