Package: SplitGLM (via r-universe)

September 2, 2024

Type Package

Title Split Generalized Linear Models

Version 1.0.5

Date 2022-11-21

Maintainer Anthony Christidis <anthony.christidis@stat.ubc.ca>

Description Functions to compute split generalized linear models. The approach fits generalized linear models that split the covariates into groups. The optimal split of the variables into groups and the regularized estimation of the coefficients are performed by minimizing an objective function that encourages sparsity within each group and diversity among them. Example applications can be found in Christidis et al. (2021) <arXiv:2102.08591>.

License GPL (>= 2)

Biarch true

Imports Rcpp (>= 1.0.3)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.0.2

Suggests testthat, vctrs, mvnfast

NeedsCompilation yes

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

RemoteUrl https://github.com/anthonychristidis/splitglm

RemoteRef HEAD

RemoteSha b2fd48fcb1352c0d83e8773daada3cda19c052d3

Contents

`coef.cv.SplitGLM` *Coefficients for cv.SplitGLM Object*

Description

`coef.cv.SplitGLM` returns the coefficients for a cv.SplitGLM object.

Usage

```
## S3 method for class 'cv.SplitGLM'
coef(object, group_index = NULL, ...)
```

Arguments

`object`	An object of class cv.SplitGLM.
`group_index`	The group for which to return the coefficients. Default is the ensemble coefficients.
`...`	Additional arguments for compatibility.

Value

The coefficients for the cv.SplitGLM object.

Author(s)

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

See Also

`cv.SplitGLM`

Examples

```
# Data simulation
set.seed(1)
n <- 50
N <- 2000
p <- 1000
beta.active <- c(abs(runif(p, 0, 1/2))*(-1)^rbinom(p, 1, 0.3))
# Parameters
p.active <- 100
beta <- c(beta.active[1:p.active], rep(0, p-p.active))
```

```
Sigma <- matrix(0, p, p)
Sigma[1:p.active, 1:p.active] <- 0.5
diag(Sigma) <- 1

# Train data
x.train <- mvnfast::rmvn(n, mu = rep(0, p), sigma = Sigma)
prob.train <- exp(x.train %*% beta)/
              (1+exp(x.train %*% beta))
y.train <- rbinom(n, 1, prob.train)
mean(y.train)
# Test data
x.test <- mvnfast::rmvn(N, mu = rep(0, p), sigma = Sigma)
prob.test <- exp(x.test %*% beta)/
             (1+exp(x.test %*% beta))
y.test <- rbinom(N, 1, prob.test)
mean(y.test)

# SplitGLM - CV (Multiple Groups)
split.out <- cv.SplitGLM(x.train, y.train,
                         glm_type="Logistic",
                         G=10, include_intercept=TRUE,
                         alpha_s=3/4, alpha_d=1,
                         n_lambda_sparsity=50, n_lambda_diversity=50,
                         tolerance=1e-3, max_iter=1e3,
                         n_folds=5,
                         active_set=FALSE,
                         n_threads=1)
split.coef <- coef(split.out)
# Predictions
split.prob <- predict(split.out, newx=x.test, type="prob", group_index=NULL)
split.class <- predict(split.out, newx=x.test, type="class", group_index=NULL)
plot(prob.test, split.prob, pch=20)
abline(h=0.5,v=0.5)
mean((prob.test-split.prob)^2)
mean(abs(y.test-split.class))
```

`coef.SplitGLM` *Coefficients for SplitGLM Object*

Description

`coef.SplitGLM` returns the coefficients for a SplitGLM object.

Usage

```
## S3 method for class 'SplitGLM'
coef(object, group_index = NULL, ...)
```

Arguments

object	An object of class SplitGLM.
group_index	The group for which to return the coefficients. Default is the ensemble.
...	Additional arguments for compatibility.

Value

The coefficients for the SplitGLM object.

Author(s)

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

See Also

SplitGLM

Examples

```
# Data simulation
set.seed(1)
n <- 50
N <- 2000
p <- 1000
beta.active <- c(abs(runif(p, 0, 1/2))*(-1)^rbinom(p, 1, 0.3))
# Parameters
p.active <- 100
beta <- c(beta.active[1:p.active], rep(0, p-p.active))
Sigma <- matrix(0, p, p)
Sigma[1:p.active, 1:p.active] <- 0.5
diag(Sigma) <- 1

# Train data
x.train <- mvnfast::rmvn(n, mu = rep(0, p), sigma = Sigma)
prob.train <- exp(x.train %*% beta)/
              (1+exp(x.train %*% beta))
y.train <- rbinom(n, 1, prob.train)
mean(y.train)
# Test data
x.test <- mvnfast::rmvn(N, mu = rep(0, p), sigma = Sigma)
prob.test <- exp(x.test %*% beta)/
             (1+exp(x.test %*% beta))
y.test <- rbinom(N, 1, prob.test)
mean(y.test)

# SplitGLM - CV (Multiple Groups)
split.out <- SplitGLM(x.train, y.train,
                      glm_type="Logistic",
                      G=10, include_intercept=TRUE,
                      alpha_s=3/4, alpha_d=1,
                      lambda_sparsity=1, lambda_diversity=1,
```

```
                    tolerance=1e-3, max_iter=1e3,
                    active_set=FALSE)
split.coef <- coef(split.out)
# Predictions
split.prob <- predict(split.out, newx=x.test, type="prob", group_index=NULL)
split.class <- predict(split.out, newx=x.test, type="class", group_index=NULL)
plot(prob.test, split.prob, pch=20)
abline(h=0.5,v=0.5)
mean((prob.test-split.prob)^2)
mean(abs(y.test-split.class))
```

`cv.SplitGLM` *Cross Validation - Split Generalized Linear Model*

Description

`cv.SplitGLM` performs the CV procedure for split generalized linear models.

Usage

```
cv.SplitGLM(
  x,
  y,
  glm_type = "Linear",
  G = 10,
  include_intercept = TRUE,
  alpha_s = 3/4,
  alpha_d = 1,
  n_lambda_sparsity = 50,
  n_lambda_diversity = 50,
  tolerance = 0.001,
  max_iter = 1e+05,
  n_folds = 10,
  active_set = FALSE,
  full_diversity = FALSE,
  n_threads = 1
)
```

Arguments

`x`	Design matrix.
`y`	Response vector.
`glm_type`	Description of the error distribution and link function to be used for the model. Must be one of "Linear", "Logistic", "Gamma" or "Poisson".

G	Number of groups into which the variables are split. Can have more than one value.
include_intercept	Boolean variable to determine if there is intercept (default is TRUE) or not.
alpha_s	Elastic net mixing parmeter. Default is 3/4.
alpha_d	Mixing parameter for diversity penalty. Default is 1.
n_lambda_sparsity	Number of candidates for the sparsity penalty parameter. Default is 100.
n_lambda_diversity	Number of candidates for the sparsity penalty parameter. Default is 100.
tolerance	Convergence criteria for the coefficients. Default is 1e-3.
max_iter	Maximum number of iterations in the algorithm. Default is 1e5.
n_folds	Number of cross-validation folds. Default is 10.
active_set	Active set convergence for the algorithm. Default is FALSE.
full_diversity	Full diversity between the groups. Default is FALSE.
n_threads	Number of threads. Default is 1.

Value

An object of class cv.SplitGLM.

Author(s)

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

See Also

coef.cv.SplitGLM, predict.cv.SplitGLM

Examples

```
# Data simulation
set.seed(1)
n <- 50
N <- 2000
p <- 1000
beta.active <- c(abs(runif(p, 0, 1/2))*(-1)^rbinom(p, 1, 0.3))
# Parameters
p.active <- 100
beta <- c(beta.active[1:p.active], rep(0, p-p.active))
Sigma <- matrix(0, p, p)
Sigma[1:p.active, 1:p.active] <- 0.5
diag(Sigma) <- 1

# Train data
x.train <- mvnfast::rmvn(n, mu = rep(0, p), sigma = Sigma)
prob.train <- exp(x.train %*% beta)/
              (1+exp(x.train %*% beta))
```

```
y.train <- rbinom(n, 1, prob.train)
mean(y.train)
# Test data
x.test <- mvnfast::rmvn(N, mu = rep(0, p), sigma = Sigma)
prob.test <- exp(x.test %*% beta)/
             (1+exp(x.test %*% beta))
y.test <- rbinom(N, 1, prob.test)
mean(y.test)

# SplitGLM - CV (Multiple Groups)
split.out <- cv.SplitGLM(x.train, y.train,
                         glm_type="Logistic",
                         G=10, include_intercept=TRUE,
                         alpha_s=3/4, alpha_d=1,
                         n_lambda_sparsity=50, n_lambda_diversity=50,
                         tolerance=1e-3, max_iter=1e3,
                         n_folds=5,
                         active_set=FALSE,
                         n_threads=1)
split.coef <- coef(split.out)
# Predictions
split.prob <- predict(split.out, newx=x.test, type="prob", group_index=NULL)
split.class <- predict(split.out, newx=x.test, type="class", group_index=NULL)
plot(prob.test, split.prob, pch=20)
abline(h=0.5,v=0.5)
mean((prob.test-split.prob)^2)
mean(abs(y.test-split.class))
```

`plot.cv.SplitGLM` *Plot of coefficients paths for cv.SplitGLM Object*

Description

`plot.cv.SplitGLM` returns the coefficients for a cv.SplitGLM object.

Usage

```
## S3 method for class 'cv.SplitGLM'
plot(
  x,
  group_index = NULL,
  plot_type = c("Coef", "CV-Error")[1],
  active_only = TRUE,
  path_type = c("Log-Lambda", "L1-Norm")[1],
  labels = TRUE,
  ...
)
```

Arguments

x	An object of class cv.SplitGLM.
group_index	The group for which to return the coefficients. Default is the ensemble coefficients.
plot_type	Plot of coefficients, "Coef" (default), or cross-validated error or deviance, "CV-Error".
active_only	Only include the variables selected in final model (default is TRUE).
path_type	Plot of coefficients paths as a function of either "Log-Lambda" (default) or "L1-Norm".
labels	Include the labels of the variables (default is FALSE).
...	Additional arguments for compatibility.

Value

The coefficients for the cv.SplitGLM object.

Author(s)

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

See Also

cv.SplitGLM

Examples

```
# Data simulation
set.seed(1)
n <- 50
N <- 2000
p <- 1000
beta.active <- c(abs(runif(p, 0, 1/2))*(-1)^rbinom(p, 1, 0.3))
# Parameters
p.active <- 100
beta <- c(beta.active[1:p.active], rep(0, p-p.active))
Sigma <- matrix(0, p, p)
Sigma[1:p.active, 1:p.active] <- 0.5
diag(Sigma) <- 1

# Train data
x.train <- mvnfast::rmvn(n, mu = rep(0, p), sigma = Sigma)
prob.train <- exp(x.train %*% beta)/
              (1+exp(x.train %*% beta))
y.train <- rbinom(n, 1, prob.train)
mean(y.train)
# Test data
x.test <- mvnfast::rmvn(N, mu = rep(0, p), sigma = Sigma)
prob.test <- exp(x.test %*% beta)/
             (1+exp(x.test %*% beta))
```

```
y.test <- rbinom(N, 1, prob.test)
mean(y.test)

# SplitGLM - CV (Multiple Groups)
split.out <- cv.SplitGLM(x.train, y.train,
                         glm_type="Logistic",
                         G=10, include_intercept=TRUE,
                         alpha_s=3/4, alpha_d=1,
                         n_lambda_sparsity=50, n_lambda_diversity=50,
                         tolerance=1e-3, max_iter=1e3,
                         n_folds=5,
                         active_set=FALSE,
                         n_threads=1)

# Plot of coefficients paths (function of Log-Lambda)
plot(split.out, plot_type="Coef", path_type="Log-Lambda", group_index=1, labels=FALSE)

# Plot of coefficients paths (function of L1-Norm)
plot(split.out, plot_type="Coef", path_type="L1-Norm", group_index=1, labels=FALSE)

# Plot of CV error
plot(split.out, plot_type="CV-Error")
```

`predict.cv.SplitGLM` *Predictions for cv.SplitGLM Object*

Description

`predict.cv.SplitGLM` returns the predictions for a SplitGLM object.

Usage

```
## S3 method for class 'cv.SplitGLM'
predict(object, newx, group_index = NULL, type = c("prob", "class")[1], ...)
```

Arguments

`object`	An object of class cv.SplitGLM.
`newx`	New data for predictions.
`group_index`	The group for which to return the coefficients. Default is the ensemble.
`type`	The type of predictions for binary response. Options are "prob" (default) and "class".
`...`	Additional arguments for compatibility.

Value

The predictions for the cv.SplitGLM object.

Author(s)

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

See Also

cv.SplitGLM

Examples

```
# Data simulation
set.seed(1)
n <- 50
N <- 2000
p <- 1000
beta.active <- c(abs(runif(p, 0, 1/2))*(-1)^rbinom(p, 1, 0.3))
# Parameters
p.active <- 100
beta <- c(beta.active[1:p.active], rep(0, p-p.active))
Sigma <- matrix(0, p, p)
Sigma[1:p.active, 1:p.active] <- 0.5
diag(Sigma) <- 1

# Train data
x.train <- mvnfast::rmvn(n, mu = rep(0, p), sigma = Sigma)
prob.train <- exp(x.train %*% beta)/
              (1+exp(x.train %*% beta))
y.train <- rbinom(n, 1, prob.train)
mean(y.train)
# Test data
x.test <- mvnfast::rmvn(N, mu = rep(0, p), sigma = Sigma)
prob.test <- exp(x.test %*% beta)/
             (1+exp(x.test %*% beta))
y.test <- rbinom(N, 1, prob.test)
mean(y.test)

# SplitGLM - CV (Multiple Groups)
split.out <- cv.SplitGLM(x.train, y.train,
                         glm_type="Logistic",
                         G=10, include_intercept=TRUE,
                         alpha_s=3/4, alpha_d=1,
                         n_lambda_sparsity=50, n_lambda_diversity=50,
                         tolerance=1e-3, max_iter=1e3,
                         n_folds=5,
                         active_set=FALSE,
                         n_threads=1)
split.coef <- coef(split.out)
# Predictions
split.prob <- predict(split.out, newx=x.test, type="prob", group_index=NULL)
```

```
split.class <- predict(split.out, newx=x.test, type="class", group_index=NULL)
plot(prob.test, split.prob, pch=20)
abline(h=0.5,v=0.5)
mean((prob.test-split.prob)^2)
mean(abs(y.test-split.class))
```

`predict.SplitGLM` *Predictions for SplitGLM Object*

Description

`predict.SplitGLM` returns the predictions for a SplitGLM object.

Usage

```
## S3 method for class 'SplitGLM'
predict(object, newx, group_index = NULL, type = c("prob", "class")[1], ...)
```

Arguments

`object`	An object of class SplitGLM.
`newx`	New data for predictions.
`group_index`	The group for which to return the coefficients. Default is the ensemble.
`type`	The type of predictions for binary response. Options are "prob" (default) and "class".
`...`	Additional arguments for compatibility.

Value

The predictions for the SplitGLM object.

Author(s)

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

See Also

SplitGLM

Examples

```
# Data simulation
set.seed(1)
n <- 50
N <- 2000
p <- 1000
beta.active <- c(abs(runif(p, 0, 1/2))*(-1)^rbinom(p, 1, 0.3))
# Parameters
p.active <- 100
beta <- c(beta.active[1:p.active], rep(0, p-p.active))
Sigma <- matrix(0, p, p)
Sigma[1:p.active, 1:p.active] <- 0.5
diag(Sigma) <- 1

# Train data
x.train <- mvnfast::rmvn(n, mu = rep(0, p), sigma = Sigma)
prob.train <- exp(x.train %*% beta)/
              (1+exp(x.train %*% beta))
y.train <- rbinom(n, 1, prob.train)
mean(y.train)
# Test data
x.test <- mvnfast::rmvn(N, mu = rep(0, p), sigma = Sigma)
prob.test <- exp(x.test %*% beta)/
             (1+exp(x.test %*% beta))
y.test <- rbinom(N, 1, prob.test)
mean(y.test)

# SplitGLM - CV (Multiple Groups)
split.out <- SplitGLM(x.train, y.train,
                      glm_type="Logistic",
                      G=10, include_intercept=TRUE,
                      alpha_s=3/4, alpha_d=1,
                      lambda_sparsity=1, lambda_diversity=1,
                      tolerance=1e-3, max_iter=1e3,
                      active_set=FALSE)
split.coef <- coef(split.out)
# Predictions
split.prob <- predict(split.out, newx=x.test, type="prob", group_index=NULL)
split.class <- predict(split.out, newx=x.test, type="class", group_index=NULL)
plot(prob.test, split.prob, pch=20)
abline(h=0.5,v=0.5)
mean((prob.test-split.prob)^2)
mean(abs(y.test-split.class))
```

SplitGLM *Split Generalized Linear Model*

Description

SplitGLM performs computes the coefficients for split generalized linear models.

Usage

```
SplitGLM(
  x,
  y,
  glm_type = "Linear",
  G = 10,
  include_intercept = TRUE,
  alpha_s = 3/4,
  alpha_d = 1,
  lambda_sparsity,
  lambda_diversity,
  tolerance = 0.001,
  max_iter = 1e+05,
  active_set = FALSE
)
```

Arguments

x	Design matrix.
y	Response vector.
glm_type	Description of the error distribution and link function to be used for the model. Must be one of "Linear", "Logistic", "Gamma" or "Poisson".
G	Number of groups into which the variables are split. Can have more than one value.
include_intercept	Boolean variable to determine if there is intercept (default is TRUE) or not.
alpha_s	Elastic net mixing parmeter. Default is 3/4.
alpha_d	Mixing parameter for diversity penalty. Default is 1.
lambda_sparsity	Sparsity tuning parameter value.
lambda_diversity	Diversity tuning parameter value.
tolerance	Convergence criteria for the coefficients. Default is 1e-3.
max_iter	Maximum number of iterations in the algorithm. Default is 1e5.
active_set	Active set convergence for the algorithm. Default is FALSE.

Value

An object of class SplitGLM.

Author(s)

Anthony-Alexander Christidis, <anthony.christidis@stat.ubc.ca>

See Also

coef.SplitGLM, predict.SplitGLM

Examples

```
# Data simulation
set.seed(1)
n <- 50
N <- 2000
p <- 1000
beta.active <- c(abs(runif(p, 0, 1/2))*(-1)^rbinom(p, 1, 0.3))
# Parameters
p.active <- 100
beta <- c(beta.active[1:p.active], rep(0, p-p.active))
Sigma <- matrix(0, p, p)
Sigma[1:p.active, 1:p.active] <- 0.5
diag(Sigma) <- 1

# Train data
x.train <- mvnfast::rmvn(n, mu = rep(0, p), sigma = Sigma)
prob.train <- exp(x.train %*% beta)/
              (1+exp(x.train %*% beta))
y.train <- rbinom(n, 1, prob.train)
mean(y.train)
# Test data
x.test <- mvnfast::rmvn(N, mu = rep(0, p), sigma = Sigma)
prob.test <- exp(x.test %*% beta)/
             (1+exp(x.test %*% beta))
y.test <- rbinom(N, 1, prob.test)
mean(y.test)

# SplitGLM - Multiple Groups
split.out <- SplitGLM(x.train, y.train,
                      glm_type="Logistic",
                      G=10, include_intercept=TRUE,
                      alpha_s=3/4, alpha_d=1,
                      lambda_sparsity=1, lambda_diversity=1,
                      tolerance=1e-3, max_iter=1e3,
                      active_set=FALSE)
split.coef <- coef(split.out)
# Predictions
split.prob <- predict(split.out, newx=x.test, type="prob", group_index=NULL)
split.class <- predict(split.out, newx=x.test, type="class", group_index=NULL)
plot(prob.test, split.prob, pch=20)
abline(h=0.5,v=0.5)
mean((prob.test-split.prob)^2)
mean(abs(y.test-split.class))
```

Index