Package ‘bujar’

June 25, 2023

Type Package

Title Buckley-James Regression for Survival Data with High-Dimensional Covariates

Version 0.2-11

Date 2023-06-23

Author Zhu Wang <https://orcid.org/0000-0002-0773-0052> and others (see COPYRIGHTS)

Maintainer Zhu Wang <zwang145@uthsc.edu>

Description Buckley-James regression for right-censoring survival data with high-dimensional covariates. Implementations for survival data include boosting with componentwise linear least squares, componentwise smoothing splines, regression trees and MARS. Other high-dimensional tools include penalized regression for survival data. See Wang and Wang (2010) <doi:10.2202/1544-6115.1550>.

Imports mda, mpath, mboost, gbm, earth, elasticnet, rms, methods, modeltools, bst, parallel, survival

Depends R (>= 3.5.0)

Suggests TH.data, R.rsp, gridExtra

VignetteBuilder R.rsp

License GPL-2

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2023-06-25 02:40:02 UTC

R topics documented:

bujar .......................... 2
chop .................................. 6
rchop .................................. 6

Index
**bujar**  

**Buckley-James Regression**

**Description**

Buckley-James regression for right-censoring survival data with high-dimensional covariates. Including L_2 boosting with componentwise linear least squares, componentwise P-splines, regression trees. Other Buckley-James methods including elastic net, MCP, SCAD, MARS and ACOSSO (ACOSSO not supported for the current version).

**Usage**

```r
bujar(y, cens, x, valdata = NULL, degree = 1, learner = "linear.regression", center=TRUE, mimpu = NULL, iter.bj = 20, max.cycle = 5, nu = 0.1, mstop = 50, twin = FALSE, mstop2 = 100, tuning = TRUE, cv = FALSE, nfold = 5, method = "corrected", vimpint = TRUE, gamma = 3, lambda = NULL, whichlambda = NULL, lamb = 0, s = 0.5, nk = 4, wt.pow = 1, theta = NULL, rel.inf = FALSE, tol = .Machine$double.eps, n.cores = 2, rng = 123, trace = FALSE)
```

## S3 method for class `bujar`

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

predict(object, newx = NULL, ...)  
## S3 method for class `bujar`

plot(x, ...)  
## S3 method for class `bujar`

coef(object, ...)  
## S3 method for class `bujar`

summary(object, ...)
```

**Arguments**

- **y**: survival time
- **cens**: censoring indicator, must be 0 or 1 with 0=alive, 1=dead
- **x**: covariate matrix
- **object**: an object of class "bujar"
- **newx**: covariate matrix for prediction
- **valdata**: test data, which must have the first column as survival time, second column as censoring indicator, and the remaining columns similar to same x.
- **degree**: mars/tree/linear regression degree of interaction; if 2, second-order interaction, if degree=1, additive model;
- **learner**: methods used for BJ regression.
- **center**: center covariates
- **mimpu**: initial estimate. If TRUE, mean-imputation; FALSE, imputed with the marginal best variable linear regression; if NULL, 0.
iter.bj  number of B-J iteration
max.cycle  max cycle allowed
nu  step-size boosting parameter
mstop  boosting tuning parameters. It can be one number or have the length iter.bj+max.cycle. If cv=TRUE, then mstop is the maximum number of tuning parameter
twin  logical, if TRUE, twin boosting
mstop2  twin boosting tuning parameter
tuning  logical value. if TRUE, the tuning parameter will be selected by cv or AIC/BIC methods. Ignored if twin=TRUE for which no tuning parameter selection is implemented
cv  logical value. if TRUE, cross-validation for tuning parameter, only used if tuning=TRUE. If tuning=FALSE or twin=TRUE, then ignored
nfold  number of fold of cv
method  boosting tuning parameter selection method in AIC
vimpint  logical value. If TRUE, compute variable importance and interaction measures for MARS if learner="mars" and degree > 1.
gamma  MCP, or SCAD gamma tuning parameter
lambda  MCP, or SCAD lambda tuning parameter
whichlambda  which lambda used for MCP or SCAD lambda tuning parameter
lamb  elastic net lambda tuning parameter, only used if learner="enet"
s  the second enet tuning parameter, which is a fraction between (0, 1), only used if learner="enet"
nk  number of basis function for learner="mars"
wt.pow  not used but kept for historical reasons, only for learner=ACOSSO. This is a parameter (power of weight). It might be chosen by CV from c(0, 1.0, 1.5, 2.0, 2.5, 3.0). If wt.pow=0, then this is COSSO method
theta  For learner="acosso", not used now. A numerical vector with 0 or 1. 0 means the variable not included and 1 means included. See Storlie et al. (2009).
rel.inf  logical value. if TRUE, variable importance measure and interaction importance measure computed
tol  convergency criteria
ncore  The number of CPU cores to use. The cross-validation loop will attempt to send different CV folds off to different cores. Used for learner="tree"
rng  a number to be used for random number generation in boosting trees
trace  logical value. If TRUE, print out interim computing results
...  additional arguments used in estimation methods, for instance, trees.

Details

Buckley-James regression for right-censoring survival data with high-dimensional covariates. Including L_2 boosting with componentwise linear least squares, componentwise P-splines, regression trees. Other Buckley-James methods including elastic net, SCAD and MCP. learner="enet" and learner="enet2" use two different implementations of LASSO. Some of these methods are discussed in Wang and Wang (2010) and the references therein. Also see the references below.
**Value**

- **x**: original covariates
- **y**: survival time
- **cens**: censoring indicator
- **ynew**: imputed y
- **yhat**: estimated y from ynew
- **pred.bj**: estimated y from the testing sample
- **res.fit**: model fitted with the learner
- **learner**: original learner used
- **degree**: =1, additive model, degree=2, second-order interaction
- **mse**: MSE at each BJ iteration, only available in simulations, or when valdata provided
- **mse.bj**: MSE from training data at the BJ termination
- **mse.bj.val**: MSE with valdata
- **mse.all**: a vector of MSE for uncensoring data at BJ iteration
- **nz.bj.iter**: number of selected covariates at each BJ iteration
- **nz.bj**: number of selected covariates at the claimed BJ termination
- **xselect**: a vector of dimension of covariates, either 1 (covariate selected) or 0 (not selected)
- **coef.bj**: estimated coefficients with linear model
- **vim**: a vector of length of number of column of x, variable importance, between 0 to 100
- **interactions**: measure of strength of interactions
- **ybstdiff**: largest absolute difference of estimated y. Useful to monitor convergency
- **ybstcon**: a vector with length of BJ iteration each is a convergency measure
- **cycleperiod**: number of cycle of BJ iteration
- **cycle.coef.diff**: within cycle of BJ, the maximum difference of coefficients for BJ boosting
- **nonconv**: logical value. if TRUE, non-convergency
- **fnorm2**: value of L_2 norm, can be useful to access convergency
- **mselect**: a vector of length of BJ iteration, each element is the tuning parameter mstop
- **contype**: 0 (converged), 1, not converged but cycle found, 2, not converged and max iteration reached.

**Author(s)**

Zhu Wang
References


Examples

data("wpbc", package = "TH.data")
wdbc2 <- wpbc[, 1:12]
wdbc2$status <- as.numeric(wdbc2$status) - 1
fit <- bujar(y=log(wdbc2$time), cens=wdbc2$status, x= wdbc2[, -(1:2)])
print(fit)
coef(fit)
pr <- predict(fit)
plot(fit)
fit <- bujar(y=log(wdbc2$time), cens=wdbc2$status, x= wdbc2[, -(1:2)], tuning = TRUE)
## Not run:
fit <- bujar(y=log(wdbc2$time), cens=wdbc2$status, x= wpbc2[, -(1:2)], learner="pspline")
fit <- bujar(y=log(wdbc2$time), cens=wdbc2$status, x= wpbc2[, -(1:2)],
learner="tree", degree=2)
### select tuning parameter for "enet"
tmp <- gcv.enet(y=log(wdbc2$time), cens=wdbc2$status, x=wpbc2[, -(1:2)])
fit <- bujar(y=log(wdbc2$time), cens=wdbc2$status, x=wpbc2[, -(1:2)], learner="enet",
lamb = tmp$lambda, s=tmp$s)
fit <- bujar(y=log(wdbc2$time), cens=wdbc2$status, x= wpbc2[, -(1:2)], learner="mars",
degree=2)
summary(fit)

## End(Not run)
**chop**  
*Survival of CHOP for diffuse large B cell lymphoma*

**Description**

Microarray data for DLBCL patients undergoing CHOP treatment.

**Usage**

```r
data(chop)
```

**Format**

The format is: `num [1:181, 1:3835]`

**Details**

Microarray data of DLBCL of 181 patients treated with a combination chemotherapy with cyclophosphamide, doxorubicin, vincristine and prednisone (CHOP). The original data have 54675 probe sets or covariates. Due to the nature of high-dimensional data, a preselection procedure was conducted to filter out the genes with lower variations if a sample variance for a gene was smaller than the 10th percentile for that gene. The first column if the survival times. The second column is an indicator whether an the survival time was observed or right censoring occurred. 0=alive, 1=dead. There are 3833 genes after the filtering process.

**Source**


**Examples**

```r
data(chop)
str(chop)
```

---

**rchop**  
*Survival of R-CHOP for diffuse large B cell lymphoma*

**Description**

Microarray data for DLBCL patients undergoing R-CHOP treatment.

**Usage**

```r
data(rchop)
```
Format

The format is: num [1:233, 1:3835]

Details

Microarray data of DLBCL of 233 patients treated with the current gold standard R-CHOP including rituxima immunotherapy in addition to the chemotherapy CHOP. The original data have 54675 probe sets or covariates. Due to the nature of high-dimensional data, a preselection procedure was conducted to filter out the genes to match those in chop. The first column if the survival times. The second column is an indicator whether an the survival time was observed or right censoring occurred. 0=alive, 1=dead. There are 3833 same genes as in chop. The data set is used to validate the prediction accuracy for models developed using training data chop.

Source


Examples

data(rchop)
str(rchop)
**Index**

* datasets
  - chop, 6
  - rchop, 6

bujar, 2

chop, 6
deco.bujar (bujar), 2
plot.bujar (bujar), 2
predict.bujar (bujar), 2
print.bujar (bujar), 2

rchop, 6

summary.bujar (bujar), 2