Package ‘bujar’

July 2, 2014

Type Package

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

Version 0.1-4

Date 2014-6-24

Author Zhu Wang and others (see COPYRIGHTS)

Maintainer Zhu Wang <zwang@connecticutchildrens.org>

Description Buckley-James regression for right-censoring survival data with high-dimensional covariates. Including L_2 boosting with componentwise linear least squares, componentwise smoothing splines, P-splines, regression trees and boosted MARS. Other high-dimensional tools include elastic net and MARS.

Imports mda, ncvreg, mboost, gbm, earth, elasticnet, rms

Suggests TH.data

License GPL-2

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2014-06-24 19:20:13

R topics documented:

`bujar-package` *Title: Buckley-James Regression with High-Dimensional Biomarker Data*

Description

Buckley-James regression for right-censoring survival data

Details

Buckley-James regression for right-censoring survival data with high-dimensional covariates. Including L_2 boosting with componentwise linear least squares, componentwise smoothing splines, P-splines, regression trees and boosted MARS. Other high-dimensional tools include elastic net, MARS, ACOSSO.

Package:	bujar
Type:	Package
Version:	0.1-2
Date:	2012-12-20
License:	GPL (version 2 or newer)
LazyLoad:	yes

Author(s)

Zhu Wang Maintainer: Zhu Wang <zwang@ccmckids.org>

References

Zhu Wang and C.Y. Wang (2010), Buckley-James Boosting for Survival Analysis with High-Dimensional Biomarker Data. *Statistical Applications in Genetics and Molecular Biology*, Vol. 9 : Iss. 1, Article 24.

`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

```
bujar(y, cens, x, valdata = NULL, glm = TRUE, degree = 1,
learner = "linear.regression", tran = "log", center=TRUE, mimpu = NULL,
iter.bj = 10, max.cycle = 10, nu = 0.1, mstop = 50, tuning = TRUE,
cv = FALSE, nfold = 5, method = "corrected", df = "actset", vimpint = TRUE,
gamma = 3, lambda=NULL, whichlambda, lamb = 0, s = 0.5, nk = 4, wt.pow = 1, theta = NULL,
rel.inf = FALSE, tol = .Machine$double.eps, trace = FALSE)
```

Arguments

`y`	survival time
`cens`	censoring indicator, must be 0 or 1
`x`	covariate matrix
`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.
`glm`	logical value. If TRUE, linear model, else, nonlinear model.
`degree`	mars/tree/linear regression degree of interaction; if 2, second-order interaction, if degree=1, additive model;
`learner`	methods used for BJ regression.
`tran`	method for response variable transformation, log-transformation of y is default.
`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 parameter. If cv=TRUE, then mstop is the maximum number of tuning parameter
`tuning`	logical value. if TRUE, the tuning parameter will be selected by cv or AIC methods for boosting
`cv`	logical value. if TRUE, cross-validation for boosting tuning parameter, only used for boosting relevant methods. If tuning=FALSE, then ignored
`nfold`	number of fold of cv
`method`	boosting tuning parameter selection method in AIC
`df`	how the degree-of-freedom is computed in "method". Only used for learner = linear.regression and tuning = TRUE.
`vimpint`	logical value. if TRUE, variable importance measure computed.
`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

s	the second enet tuning parameter, which is a fraction between (0, 1)
nk	number of basis function for learner="mars"
wt.pow	if 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	A numerical vector with 0 or 1, only for learner=oracle. 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
trace	logical value. If TRUE, print out interim computing results

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, MARS and ACOSSO. 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 convergencey
`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

Zhu Wang and C.Y. Wang (2010), Buckley-James Boosting for Survival Analysis with High-Dimensional Biomarker Data. *Statistical Applications in Genetics and Molecular Biology*, Vol. 9 : Iss. 1, Article 24.

Peter Buhlmann and Bin Yu (2003), Boosting with the L2 loss: regression and classification. *Journal of the American Statistical Association*, **98**, 324–339.

Peter Buhlmann (2006), Boosting for high-dimensional linear models. *The Annals of Statistics*, **34**(2), 559–583.

Peter Buhlmann and Torsten Hothorn (2007), Boosting algorithms: regularization, prediction and model fitting. *Statistical Science*, **22**(4), 477–505.

J. Friedman (1991), Multivariate Adaptive Regression Splines (with discussion) . *Annals of Statistics*, **19**/1, 1–141.

J.H. Friedman, T. Hastie and R. Tibshirani (2000), Additive Logistic Regression: a Statistical View of Boosting. *Annals of Statistics* **28**(2):337-374.

C. Storlie, H. Bondell, B. Reich and H. H. Zhang (2009), Surface Estimation, Variable Selection, and the Nonparametric Oracle Property. *Statistica Sinica*, to appear.

Sijian Wang, Bin Nan, Ji Zhu, and David G. Beer (2008), Doubly penalized Buckley-James Method for Survival Data with High-Dimensional Covariates. *Biometrics*, **64**:132-140.

H. Zou and T. Hastie (2005), Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society*, Series B, **67**, 301-320.

Examples

```
data("wpbc", package = "TH.data")
wpbc2 <- wpbc[, 1:12]
wpbc2$status <- as.numeric(wpbc2$status) - 1
fit <- bujar(y=log(wpbc2$time),cens=wpbc2$status, x= wpbc2[, -(1:2)])
print(fit)
```

```
coef(fit)
pr <- predict(fit)
plot(fit)
fit <- bujar(y=log(wpbc2$time),cens=wpbc2$status, x= wpbc2[, -(1:2)], tuning = TRUE)
## Not run:
fit <- bujar(y=log(wpbc2$time),cens=wpbc2$status, x=wpbc2[, -(1:2)], glm=FALSE,
 learner="pspline")
fit <- bujar(y=log(wpbc2$time),cens=wpbc2$status, x=wpbc2[, -(1:2)], glm=FALSE,
 learner="tree", degree=2)
### select tuning parameter for "enet"
tmp <- gcv.enet(y=log(wpbc2$time), cens=wpbc2$status, x=wpbc2[, -(1:2)])
fit <- bujar(y=log(wpbc2$time),cens=wpbc2$status, x=wpbc2[, -(1:2)], learner="enet",
lamb = tmp$lambda, s=tmp$s)

fit <- bujar(y=log(wpbc2$time),cens=wpbc2$status, x=wpbc2[, -(1:2)], glm=FALSE,
 learner="mars", degree=2)
summary(fit)

## End(Not run)
```

Index