Package 'AdapEnetClass'

October 24, 2015

Type Package

Title A Class of Adaptive Elastic Net Methods for Censored Data

Version 1.2

Date 2015-10-24

Author Hasinur Rahaman Khan and Ewart Shaw

Maintainer Hasinur Rahaman Khan <hasinurkhan@gmail.com>

Depends imputeYn, glmnet, lars, quadprog, R (>= 3.0.2)

Description Provides new approaches to variable selection for AFT model.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2015-10-24 10:57:30

R topics documented:

AdapEnetClass-package 2
AEnet.aft 3
AEnetCC.aft 5
cv.AWEnet 7
cv.AWEnetCC 9
Enet.wls 10
MCLcleaned 11
mrbj 13
plotObsEst 14
WEnet.aft 16
WEnetCC.aft 17

Index **20**

`AdapEnetClass-package` *A Class of Adaptive Elastic Net Methods for Censored Data*

Description

Provides new approaches to variable selection for AFT model.

Details

This package provides new approaches to variable selection for censored data including its high-dimensionality, based on AFT models optimized using regularized weighted least squares. Approaches namely, a weighted elastic net, an adaptive elastic net, and two of their extensions by adding censoring observations as constraints into their model optimization frameworks are provided with both simulated and real (MCL) data examples.

The accelerated failure time (AFT) models have proved useful in many contexts, though heavy censoring (as for example in cancer survival) and high dimensionality (as for example in microarray data) cause difficulties for model fitting and model selection. The package provide new approaches to variable selection for censored data, based on AFT models optimized using regularized weighted least squares. The regularized technique uses a mixture of L1 and L2 norm penalties under two elastic net type approaches. The approaches extend the original approaches proposed by Ghosh (2007), and Hong and Zhang (2010). This package also provides two extended approaches by adding censoring observations as constraints into their model optimization frameworks.

Package:	AdapEnetClass
Type:	Package
Version:	1.2
Date:	2015-10-24
License:	GPL-2
Depends:	imputeYn, glmnet, lars, quadprog

Author(s)

Hasinur Rahaman Khan and Ewart Shaw Maintainer: Hasinur Rahaman Khan <hasinurkhan@gmail.com>

References

Ghosh, S. (2007). Adaptive Elastic Net: An Improvement of Elastic Net to achieve Oracle Properties. Technical Reports, Indiana University-Purdue University, Indianapolis, USA. PR no. 07-01.

Hong, D. and Zhang, F. (2010). Weighted Elastic Net Model for Mass Spectrometry Imaging Processing. Mathematical Modelling of Natural Phenomena, 5, 115-133.

Jin, Z., Lin, D. Y. and Ying, Z. (2006). On least-squares regression with censored data. Biometrika, 93, 147-161.

Khan and Shaw (2015) imputeYn: Imputing the last largest censored observation/observations under weighted least squares. R package version 1.3, https://cran.r-project.org/package=imputeYn.

Khan and Shaw (2015). Variable Selection for Survival Data with a Class of Adaptive Elastic Net Techniques. Statistics and Computing (published online; DOI: 10.1007/s11222-015-9555-8). Also available in http://arxiv.org/abs/1312.2079.

Khan and Shaw (2013). Variable Selection with The Modified Buckley-James Method and The Dantzig Selector for High-dimensional Survival Data. Proceedings 59th ISI World Statistics Congress, 25-30 August 2013, Hong Kong, p. 4239-4244.

Examples

```
#For full data typically used for AFT models (using imputeYn (2015) package)
dat<-data(n=100, p=10, r=0, b1=c(rep(5,5),rep(0,5)), sig=1, Cper=0)

#mrbj: modified resampling based buckley-james
l<-mrbj(cbind(dat$y, dat$delta) ~ dat$x, mcsize=100, trace=FALSE, gehanonly=FALSE)

#AEnet.aft: adaptive elastic net
wt<-round(l$enet)
ft.1<-AEnet.aft(dat$x, dat$y, dat$delta, weight=wt, lambda2=1, maxit=10)

#AEnetCC.aft: adaptive elastic net with censoring constraints
## Not run: ft.1cc<-AEnetCC.aft(dat$x, dat$y, dat$delta, weight=wt, C=1,
s = 0.959596, lambda2=0.5)
## End(Not run)

#WEnet.aft: weighted elastic net
#mrbj: modified resampling based buckley-james
## Not run: l<-mrbj(cbind(dat$y, dat$delta) ~ dat$x, mcsize=100, trace=FALSE, gehanonly=TRUE)
## Not run: wt<-l$gehansd
## Not run: ft.2<-WEnet.aft(dat$x, dat$y, dat$delta, weight=wt, lambda2=0.5, maxit=10)

#WEnetCC.aft: weighted elastic net with censoring constraints
## Not run: ft.2cc<-WEnetCC.aft(dat$x, dat$y, dat$delta, weight=wt, C=1, s = 1, lambda2=0.5)
```

`AEnet.aft` *Adaptive elastic net for censored data based on AFT models*

Description

This function is used to fit AFT models using adaptive elastic net.

Usage

```
AEnet.aft(X, Y, delta, weight, lambda2, maxit = 10)
```

Arguments

X	covariate matrix under study, particularly for AFT modelling. The order of matrix covariate is typically n by p.

Y	typically the logarithmic of the survival time under AFT models. Otherwise survival time.
delta	status. it includes value 1 for uncensored and value 0 for censored subject.
weight	vector of observation weights. Weight is based on initial estimator that is obtained from elastic net on the weighted data (see Enet.wls function).
lambda2	regularization parameter for the L2 norm of the coefficients. This is typically assumed to take values in a relatively small grid, say (0, 0.5, 1.0, 1.5, 2.0, ..., 5).
maxit	an optional bound for the number of steps to be taken. Default is 10.

Details

This function is used to fit AFT models using adaptive elastic net approach (Khan and Shaw, 2015). The method works for both cases of datasets high-dimensional where the number of variables (p) is greater than the number of subjects (n) and low-dimensional.

The adaptive elastic net is transformed into an adaptive lasso type problem in an augmented space and then is solved using the lars algorithm. This is an extension of the work Ghosh (2007) for censored data. To find the optimal value for the tuning parameters lambda11 and lambda2, first lambda2 is typically assumed to take values in a relatively small grid, say (0, 0.5, 1.0, 1.5, 2.0, ..., 5). For each lambda2, the lars algorithm produces the entire solution path. This gives the optimal equivalent specification for lasso in terms of fraction of the L1 norm.

Value

A "AEnet.aft" object is returned. It includes

beta	coefficient estimates of the covariates
mu	means of Y
meanx	means of the columns of X
normx	normalized value of the columns of X
type	lasso

Author(s)

Hasinur Rahaman Khan and Ewart Shaw

References

Ghosh, S. (2007). Adaptive Elastic Net: An Improvement of Elastic Net to achieve Oracle Properties. Technical Reports, Indiana University- Purdue University, Indianapolis, USA. PR no. 07-01.

Khan and Shaw (2015) imputeYn: Imputing the last largest censored observation/observations under weighted least squares. R package version 1.3, https://cran.r-project.org/package=imputeYn.

Khan and Shaw (2015). Variable Selection for Survival Data with a Class of Adaptive Elastic Net Techniques. Statistics and Computing (published online; DOI: 10.1007/s11222-015-9555-8). Also available in http://arxiv.org/abs/1312.2079.

See Also

cv.AWEnet

Examples

```
#For full data typically used for AFT models (using imputeYn (2015) package).
dat<-data(n=100, p=10, r=0, b1=c(rep(5,5),rep(0,5)), sig=1, Cper=0)

#This needs to run for generating weights of the observations
l<-mrbj(cbind(dat$y, dat$delta) ~ dat$x, mcsize=100, trace=FALSE, gehanonly=FALSE)

#AEnet.aft: adaptive elastic net
wt<-round(l$enet)
ft.1<-AEnet.aft(dat$x, dat$y, dat$delta, weight=wt, lambda2=1, maxit=10)
ft.1
```

AEnetCC.aft   *Adaptive elastic net for AFT models with censoring constraints*

Description

This function is used to fit AFT models by using adaptive elastic net.

Usage

```
AEnetCC.aft(X, Y, delta, weight, C = 1, s = 1, lambda2)
```

Arguments

X	covariate matrix under study, particularly for AFT modelling. The order of matrix covariate is typically n by p.
Y	typically the logarithmic of the survival time under AFT models. Otherwise survival time.
delta	status. it includes value 1 for uncensored and value 0 for censored subject.
weight	vector of observation weights.
C	this is a positive value that accounts for the penalties of violations of constraints. C is typically allowed to take values in a grid such as (0, 0.5, 1, 1.5, ..., 10).
s	this is the optimal equivalent specification for lasso in terms of fraction of the L1 norm. This is obtained from the AEnet.aft function.
lambda2	regularization parameter for the L2 norm of the coefficients. This is typically assumed to take values in a relatively small grid, say (0, 0.5, 1.0, 1.5, 2.0, ..., 5).

Details

This function is used to fit AFT models using adaptive elastic net with censorig constraints (Khan and Shaw, 2015). This is an extension of the adaptive elastic approach that allows the censoring constraints to be implemented into the optimization framework. The method works for both cases of datasets high-dimensional where the number of variables (p) is greater than the number of subjects (n) and low-dimensional.

This method use the same optimal pair of (lambda1, lambda2) as found in AEnet.aft. Then C is typically allowed to take values in a grid such as (0, 0.5, 1.0, 1.5, 2.0,...,5), and the optimal value for C obtained by 5-fold cross-validation. Here C typically depends upon how stringently one wants the model to satisfy the censoring constraints compared to how good is the prediction for uncensored data.

Value

An object of type numeric is returned that provides the coefficient estimates.

Author(s)

Hasinur Rahaman Khan and Ewart Shaw

References

Ghosh, S. (2007). Adaptive Elastic Net: An Improvement of Elastic Net to achieve Oracle Properties. Technical Reports, Indiana University- Purdue University, Indianapolis, USA. PR no. 07-01.

Khan and Shaw (2015) imputeYn: Imputing the last largest censored observation/observations under weighted least squares. R package version 1.3, https://cran.r-project.org/package=imputeYn.

Khan and Shaw (2015). Variable Selection for Survival Data with a Class of Adaptive Elastic Net Techniques. Statistics and Computing (published online; DOI: 10.1007/s11222-015-9555-8). Also available in http://arxiv.org/abs/1312.2079.

See Also

cv.AWEnetCC

Examples

```
#For full data typically used for AFT models (using imputeYn (2015) package).
dat<-data(n=100, p=10, r=0, b1=c(rep(5,5), rep(0, 5)), sig=1, Cper=0)

#This needs to run for generating weights of the observations
l<-mrbj(cbind(dat$y, dat$delta) ~ dat$x, mcsize=100, trace=FALSE, gehanonly=FALSE)

#AEnetCC.aft: adaptive elastic net with censoring constraints
wt<-round(l$enet,3)
ft<-AEnetCC.aft(dat$x, dat$y, dat$delta, weight=wt, C=1, s=959596, lambda2=0.5)
ft
```

cv.AWEnet *Computes K-fold cross validated error curve for AEnet and WEnet*

Description

This function computes the K-fold cross validation estimates.

Usage

```
cv.AWEnet(X, Y, delta, weight, lambda2, maxit, K = 10, fraction = seq(from = 0,
to = 1, length = 100), plot.it = F, se = TRUE, AEnet = T, all.folds = NULL)
```

Arguments

X	covariate matrix under study, particularly for AFT modelling. The order of matrix covariate is typically n by p.
Y	typically the logarithmic of the survival time under AFT models. Otherwise survival time.
delta	status. it includes value 1 for uncensored and value 0 for censored subject.
weight	vector of observation weights. Weight is based on initial estimator that is obtained from elastic net on the weighted data (see Enet.wls function) or from Gehan estimator (see mrbj function).
lambda2	regularization parameter for the L2 norm of the coefficients. This is typically assumed to take values in a relatively small grid.
maxit	an optional bound for the number of steps to be taken. Default is 10.
K	number of folds.
fraction	abscissa values at which CV curve should be computed. This is the fraction of the saturated |beta|. The default value is seq(from = 0, to = 1, length =100).
plot.it	if T then plot will be showed. Default is T.
se	include standard error bands.
AEnet	if T then the results are based on adaptive elastic net otherwise based on weighted elastic net.
all.folds	null.

Details

This function computes the K-fold cross validation, cross validation error, cross validation mean squared error.

Value

An "index" object is returned with a CV curve. The index includes

`lambda2`	as AEnetCC.aft
`cv`	the CV curve at each value of index
`cv.mse`	the mean square error of the CV curve
`cv.error`	the standard error of the CV curve

Author(s)

Hasinur Rahaman Khan and Ewart Shaw

References

Khan and Shaw (2015) imputeYn: Imputing the last largest censored observation/observations under weighted least squares. R package version 1.3, https://cran.r-project.org/package=imputeYn.

Khan and Shaw (2015). Variable Selection for Survival Data with a Class of Adaptive Elastic Net Techniques. Statistics and Computing (published online; DOI: 10.1007/s11222-015-9555-8). Also available in http://arxiv.org/abs/1312.2079.

See Also

cv.AWEnetCC

Examples

```
#For full data typically used for AFT models (using imputeYn (2015) package).
dat<-data(n=100, p=10, r=0, b1=c(rep(5,5),rep(0,5)), sig=1, Cper=0)

#This needs to run for generating weights of the observations
l<-mrbj(cbind(dat$y, dat$delta) ~ dat$x, mcsize=100, trace=FALSE, gehanonly=FALSE)

#cv.AWEnet: Cross validation of Adaptive elastic net
wt<-l$enet
## Not run: cv1 <-cv.AWEnet(dat$x, dat$y, dat$delta, weight=wt, lambda2=0.001, maxit=10,
plot.it = T, AEnet=T)
## End(Not run)
## Not run: cv1$index[which.min(cv1$cv)]

#cv.AWEnet: Cross validation of weighted elastic net
## Not run: l<-mrbj(cbind(dat$y, dat$delta) ~ dat$x, mcsize=100, trace=FALSE, gehanonly=TRUE)
## Not run: wt<-l$gehansd
## Not run: cv2 <-cv.AWEnet(dat$x, dat$y, dat$delta, weight=wt, lambda2=0.001,
maxit=10, plot.it = T, AEnet=F)
## End(Not run)
## Not run: cv2$index[which.min(cv2$cv)]
```

`cv.AWEnetCC` *Computes K-fold cross validated error curve for AEnetCC and WEnetCC*

Description

This function computes the K-fold cross validation estimates.

Usage

```
cv.AWEnetCC(X, Y, delta, weight, kFold = 10, C, s, lambda2, AEnetCC = T)
```

Arguments

`X`	covariate matrix under study, particularly for AFT modelling. The order of matrix covariate is typically n by p.
`Y`	typically the logarithmic of the survival time under AFT models. Otherwise survival time.
`delta`	status. it includes value 1 for uncensored and value 0 for censored subject.
`weight`	vector of observation weights. Weight is based on initial estimator that is obtained from elastic net on the weighted data (see Enet.wls function) or from Gehan estimator (see mrbj function).
`kFold`	number of folds.
`C`	this is a positive value that accounts for the penalties of violations of constraints. C is typically allowed to take values in a grid such as (0, 0.5, 1, 1.5, ..., 10).
`s`	this is the optimal equivalent specification for lasso in terms of fraction of the L1 norm. This is obtained from the AEnet.aft function.
`lambda2`	regularization parameter for the L2 norm of the coefficients. This is typically assumed to take values in a relatively small grid.
`AEnetCC`	If T then the results are based on adaptive elastic net with censoring constraints otherwise based on the weighted elastic net with censoring constraints.

Details

The function gives the K-fold cross validation, cross validation error, cross validation mean squared error.

Value

`beta`	shows coefficient estimates of the covariates.
`betavar`	variance of the coefficient estimates.
`cvscore`	a CV score based on the CV error. This is basically the sum of squared residuals of uncensored data multiplied by the Kaplan-Meier weights (Khan and Shaw, 2015).

References

Khan and Shaw (2015) imputeYn: Imputing the last largest censored observation/observations under weighted least squares. R package version 1.3, https://cran.r-project.org/package=imputeYn.

Khan and Shaw (2015). Variable Selection for Survival Data with a Class of Adaptive Elastic Net Techniques. Statistics and Computing (published online; DOI: 10.1007/s11222-015-9555-8). Also available in http://arxiv.org/abs/1312.2079.

See Also

cv.AWEnet

Examples

```
#For full data typically used for AFT models (using imputeYn (2015) package)
dat<-data(n=100, p=10, r=0, b1=c(rep(5,5),rep(0,5)), sig=1, Cper=0)

#This needs to run for generating weights of the observations
l<-mrbj(cbind(dat$y, dat$delta) ~ dat$x, mcsize=100, trace=FALSE, gehanonly=FALSE)

#cv.AWEnetCC: Cross validation of Adaptive elastic net with censoring constraints
wt<-l$enet
cv1cc<-cv.AWEnetCC(dat$x, dat$y, dat$delta, weight=wt, kFold = 10, C=1.2, s=0.88,
 lambda2=0.001, AEnetCC=TRUE)

#cv.AWEnetCC: Cross validation of weighted elastic net with censoring constraints
## Not run: l<-mrbj(cbind(dat$y, dat$delta) ~ dat$x, mcsize=100, trace=FALSE, gehanonly=TRUE)
## Not run: wt<-l$gehansd
## Not run: cv1cc<-cv.AWEnetCC(dat$x, dat$y, dat$delta, weight=wt, kFold = 10, C=1.2, s=0.88,
 lambda2=0.001, AEnetCC=F)
## End(Not run)
```

Enet.wls *Elastic net based on the weighted data*

Description

This function provides the estimates of the initial estimators.

Arguments

X	covariate matrix under study, particularly for AFT modelling. The order of matrix covariate is typically n by p.
Y	typically the logarithmic of the survival time under AFT models. Otherwise survival time.
delta	status. it includes value 1 for uncensored and value 0 for censored subject.

Details

This function provides the estimates of the initial estimators that are used to calculate weights for observations under adaptive and weighted elastic net approaches. Here simple elastic net method is applied to the data weighted by the Kaplan-Meier weights (Khan and Shaw, 2015). The elastic net estimates are obtained using glmnet.

Value

`beta`	coefficient estimates of the covariates
`fit`	an object that gives df (The number of nonzero coefficients for each value of lambda), Dev (deviance) abd Lambda (The actual sequence of lambda values used)

Author(s)

Hasinur Rahaman Khan and Ewart Shaw

References

Khan and Shaw (2015) imputeYn: Imputing the last largest censored observation/observations under weighted least squares. R package version 1.3, https://cran.r-project.org/package=imputeYn.

Khan and Shaw (2015). Variable Selection for Survival Data with a Class of Adaptive Elastic Net Techniques. Statistics and Computing (published online; DOI: 10.1007/s11222-015-9555-8). Also available in http://arxiv.org/abs/1312.2079.

Examples

```
#For full data typically used for AFT models (using imputeYn (2015) package).
dat<-data(n=100, p=10, r=0, b1=c(rep(5,5),rep(0,5)), sig=1, Cper=0)

#Enet.wls: Elastic net on the weighted data
enet<-Enet.wls(dat$x, dat$y, dat$delta)
enet
```

`MCLcleaned` *Mantle cell lymphoma cleaned data*

Description

This is a cleaned gene expression data called Mantle cell lymphoma.

Usage

```
data(MCLcleaned)
```

Format

It contains total 577 columns namely, ID: array identifier for 92 patients, time: time of follow-up in year, cens: pateint status at follow up (1=death, 0=censored), and the remaining 574 columns are the gene expression value of 574 cDNA denoted by X37,..,X8893 which are the measurements of the log2 CY5/CY3 expression ratio for each feature. The dummy gene id X.... can be matched with the original gene UNIQID as stated at http://llmpp.nih.gov/MCL/.

Details

This is a cleaned gene expression Mantle cell lymphoma dataset that contains expression values of 574 cDNA elements as used in Khan and Shaw (2015). The full dataset is available at http://llmpp.nih.gov/MCL/ as used in Rosenwald et al. (2003). Total 92 patients were classified as having MCL, based on established morphologic and immunophenotypic criteria. Survival times of 64 patients were available and the remaining 28 patients were censored. The data do not provide any further relevant covariates for MCL patients. Use as "data(MCLcleaned)".

Author(s)

Hasinur Rahaman Khan and Ewart Shaw

Source

The full dataset is available at http://llmpp.nih.gov/MCL/.

References

Khan and Shaw (2015). Variable Selection for Survival Data with a Class of Adaptive Elastic Net Techniques. Statistics and Computing (published online; DOI: 10.1007/s11222-015-9555-8). Also available in http://arxiv.org/abs/1312.2079.

Rosenwald A., Wright G., Wiestner A., Chan W., Connors J., Campo E., Gascoyne R., Grogan T., Muller Hermelink H., Smeland E., Chiorazzi M., Giltnane J., Hurt E., Zhao H., Averett L., Henrickson S., Yang L., Powell J., Wilson W., Jaffe E., Simon R., Klausner R., Montserrat E., Bosch F., Greiner T., Weisenburger D., Sanger W., Dave B., Lynch J., Vose J., Armitage J., Fisher R., Miller T., LeBlanc M., Ott G., Kvaloy S., Holte H., Delabie J., Staudt L. (2003): The proliferation gene expression signature is a quantitative integrator of oncogenic events that predicts survival in mantle cell lymphoma. Cancer Cell, 3, 185-197.

Examples

```
#Enet.wls: Elastic net on the weighted data Using MCLcleaned data attached with this package
data(MCLcleaned)
GEval<-MCLcleaned[4:577]
enet<-Enet.wls(GEval, log(MCLcleaned$time), MCLcleaned$cens)
enet
```

mrbj *Modified resampling based Buckley-James method*

Description

This function provides estimates for AFT models by using iterative process.

Usage

```
mrbj(formula, data, subset, trace = FALSE, gehanonly = FALSE, cov = FALSE,
na.action = na.exclude, residue = FALSE, mcsize = 100)
```

Arguments

`formula`	specifies a model to be fitted. The response and covariates of the model are separated by a ~ operator. The response, on the left side of ~, should be a Surv object with two columns, of which the first column is the survival time or censored time and the second column is the censoring indicator. The covariates or predictors X, on the right side of ~, should be columns with the same length as Surv object.
`data`	a data frame which contains the Surv objects and covariates.
`subset`	specifies subset of the original data frame that should be used for the model fit.
`trace`	takes logical values TRUE or FALSE. If it is set to be TRUE, then the summary of every iteration will be kept. The default is FALSE.
`gehanonly`	takes logical values T or F. If gehanonly=T, only Gehan estimator will be calculated and otherwise, estimator will be calculated using elastic net applied to weighted data. The default is gehanonly=F.
`cov`	takes logical values T or F. If cov=T, the covariance matrices of the Gehen estimator will be printed. The default is cov=F.
`na.action`	takes values na.exclude or na.fail. The default is na.exclude, which deletes the observations with missing values. The other choice is na.fail, which returns an error if any missing values are found.
`residue`	default is FALSE.
`mcsize`	specifies the resampling number. The default is 500.

Details

This function is based on an alternative method to the Buckley-James method based on the accelerated failure time models as discussed in Jin et al.(2006) that provides an approximating approach for the consistent root of the estimating equation since the general Buckley-James method does not guarantee the convergence of its iterative process (Khan and Shaw, 2013).

Modified resampling based Buckley James method is an extension of the work proposed by Jin et al. (2006). Here, if n (observations)>p (number of covariates under AFT models) then Gehan estimator is used an an initial estimator in the Buckley-James iterative process to produce consistent estimator. For n<p, the elastic net on the weighted data is used and bootsrap is used to generate standard error of the estimates.

Value

The Gehan estimator, the standard error of the Gehan estimator, the Z score and the p-value for testing the hypothesis of beta=0 based on Gehan estimation. The elastic net estimator, the standard error of the elastic net estimator, The covariance matrices of the Gehan estimator, when cov is set to be T.

Author(s)

Hasinur Rahaman Khan and Ewart Shaw

References

Jin, Z., Lin, D. Y. and Ying, Z. (2006). On least-squares regression with censored data. Biometrika, 93, 147-161.

Khan and Shaw (2015) imputeYn: Imputing the last largest censored observation/observations under weighted least squares. R package version 1.3, https://cran.r-project.org/package=imputeYn.

Khan and Shaw (2013). Variable Selection with The Modified Buckley-James Method and The Dantzig Selector for High-dimensional Survival Data. Proceedings 59th ISI World Statistics Congress, 25-30 August 2013, Hong Kong, p. 4239-4244.

See Also

lss package

Examples

```
#For full data typically used for AFT models (using imputeYn (2015) package)
dat<-data(n=100, p=10, r=0, b1=c(rep(5,5),rep(0,5)), sig=1, Cper=0)

#mrbj: modified resampling based Buckley-James method.
#When gehanonly=T, it give both the Gehan and the elastic net  estimates
## Not run: fit1<-mrbj(cbind(dat$y, dat$delta) ~ dat$x, mcsize=100, trace=FALSE, gehanonly=TRUE)
## Not run: fit1

#mrbj: modified resampling based Buckley-James method.
#Only for Gehan estimates
fit2<-mrbj(cbind(dat$y, dat$delta) ~ dat$x, mcsize=100, trace=FALSE, gehanonly=FALSE)
fit2
```

---

plotObsEst — *Pairwise scatter plots of the survival times*

Description

This function generates pairwise scatter plots.

Usage

```
plotObsEst(yObs, yEst, delta, xlab = NULL, ylab = NULL, title = NULL,
legendplot = TRUE, legendpos = "topleft", maxvalue = NULL, minvalue = NULL)
```

Arguments

`yObs`	observed survival times.
`yEst`	predicted survival times.
`delta`	status. it includes value 1 for uncensored and value 0 for censored subject.
`xlab`	x-axis title. Default is NULL.
`ylab`	y-axis title. Default is NULL.
`title`	main title of the plot.
`legendplot`	if TRUE, plot the legend.
`legendpos`	position of the legend in the plot.
`maxvalue`	maximum value of the yObs and yEst. Default is Null.
`minvalue`	minimum value of the yObs and yEst. Default is Null.

Details

This function generates pairwise scatter plots of the observed and predicted survival times.

Value

This provides a scatter plot of the survival times.

References

Khan and Shaw (2015). Variable Selection for Survival Data with a Class of Adaptive Elastic Net Techniques. Statistics and Computing (published online; DOI: 10.1007/s11222-015-9555-8). Also available in http://arxiv.org/abs/1312.2079.

Examples

```
#For a hypothetical data
y<-c(12,33,22,34,11,23)
delta<-c(1,0,0,1,0,1)
yEst<-c(11,30,24,30,6,13)

#plotObsEst: scatter plotting of the pairwise survival times
plot<-plotObsEst(y, yEst, delta, xlab = NULL, ylab = NULL, title = "Predicted
versus Observed Survival times", legendplot = TRUE, legendpos = "topleft",
maxvalue = NULL, minvalue = NULL)
```

WEnet.aft *Weighted elastic net for censored data based on AFT models*

Description

This function is used to fit AFT models using weighted elastic net approach.

Usage

```
WEnet.aft(X, Y, delta, weight, lambda2, maxit = 10)
```

Arguments

X	covariate matrix under study, particularly for AFT modelling. The order of matrix covariate is typically n by p.
Y	typically the logarithmic of the survival time under AFT models. Otherwise survival time.
delta	status. it includes value 1 for uncensored and value 0 for censored subject.
weight	vector of observation weights based on the standard errors. Weight is based on initial estimator that is obtained from elastic net on the weighted data (see Enet.wls function).
lambda2	regularization parameter for the L2 norm of the coefficients. This is typically assumed to take values in a relatively small grid, say (0, 0.5, 1.0, 1.5, 2.0, ..., 5).
maxit	an optional bound for the number of steps to be taken. Default is 10.

Details

This function is used to fit AFT models using weighted elastic net approach (Khan and Shaw, 2015). The method works for both cases of datasets high-dimensional where the number of variables (p) is greater than the number of subjects (n) and low-dimensional.

The weighted elastic net is transformed into an adaptive lasso type problem in an augmented space and then is solved using the lars algorithm. This is an extension of the work Hong and Zhang (2010) for censored data. To find the optimal value for the tuning parameters lambda11 and lambda2, first lambda2 is typically assumed to take values in a relatively small grid, say (0, 0.5, 1.0, 1.5, 2.0, ..., 5). For each lambda2, the lars algorithm produces the entire solution path. This gives the optimal equivalent specification for lasso in terms of fraction of the L1 norm.

Value

A "WEnet.aft" object is returned. It includes

beta	coefficient estimates of the covariates
mu	means of Y
meanx	means of the columns of X
normx	normalized value of the columns of X
type	lasso

Author(s)

Hasinur Rahaman Khan and Ewart Shaw

References

Hong, D. and Zhang, F. (2010). Weighted Elastic Net Model for Mass Spectrometry Imaging Processing. Mathematical Modelling of Natural Phenomena, 5, 115-133.

Khan and Shaw (2015) imputeYn: Imputing the last largest censored observation/observations under weighted least squares. R package version 1.3, https://cran.r-project.org/package=imputeYn.

Khan and Shaw (2015). Variable Selection for Survival Data with a Class of Adaptive Elastic Net Techniques. Statistics and Computing (published online; DOI: 10.1007/s11222-015-9555-8). Also available in http://arxiv.org/abs/1312.2079.

See Also

cv.AWEnet

Examples

```
#For full data typically used for AFT models (using imputeYn (2015) package).
dat<-data(n=100, p=10, r=0, b1=c(rep(5,5),rep(0,5)), sig=1, Cper=0)

#This needs to run for generating weights of the observations
l<-mrbj(cbind(dat$y, dat$delta) ~ dat$x, mcsize=10, trace=FALSE, gehanonly=TRUE)

#WEnet.aft: weighted elastic net
wt<-round(l$gehansd)
ft.2<-WEnet.aft(dat$x, dat$y, dat$delta, weight=wt, lambda2=1, maxit=10)
ft.2
```

WEnetCC.aft *Weighted elastic net with censoring constraints for censored data based on AFT models*

Description

This function is used to fit AFT models by using weighted elastic net with censorig constraints.

Usage

```
WEnetCC.aft(X, Y, delta, weight, C, s, lambda2)
```

Arguments

X	covariate matrix under study, particularly for AFT modelling. The order of matrix covariate is typically n by p.
Y	typically the logarithmic of the survival time under AFT models. Otherwise survival time.
delta	status. it includes value 1 for uncensored and value 0 for censored subject.
weight	vector of observation weights.
C	this is a positive value that accounts for the penalties of violations of constraints. C is typically allowed to take values in a grid such as (0, 0.5, 1, 1.5, ..., 10).
s	this is the optimal equivalent specification for lasso in terms of fraction of the L1 norm. This is obtained from the AEnet.aft function.
lambda2	regularization parameter for the L2 norm of the coefficients. This is typically assumed to take values in a relatively small grid, say (0, 0.5, 1.0, 1.5, 2.0, ..., 5).

Details

This function is used to fit AFT models using weighted elastic net with censorig constraints (Khan and Shaw, 2015). This is an extension of the weighted elastic approach that allows the censoring constraints to be implemented into the optimization framework. The method works for both cases of datasets high-dimensional where the number of variables (p) is greater than the number of subjects (n) and low-dimensional.

This method use the same optimal pair of (lambda1, lambda2) as found in AEnet.aft. Then C is typically allowed to take values in a grid such as (0, 0.5, 1.0, 1.5, 2.0,...,5), and the optimal value for C obtained by 5-fold cross-validation. Here C typically depends upon how stringently one wants the model to satisfy the censoring constraints compared to how good is the prediction for uncensored data.

Value

An object of type numeric is returned that provides the coefficient estimates.

Author(s)

Hasinur Rahaman Khan and Ewart Shaw

References

Hong, D. and Zhang, F. (2010). Weighted Elastic Net Model for Mass Spectrometry Imaging Processing. Mathematical Modelling of Natural Phenomena, 5, 115-133.

Khan and Shaw (2015) imputeYn: Imputing the last largest censored observation/observations under weighted least squares. R package version 1.3, https://cran.r-project.org/package=imputeYn.

Khan and Shaw (2015). Variable Selection for Survival Data with a Class of Adaptive Elastic Net Techniques. Statistics and Computing (published online; DOI: 10.1007/s11222-015-9555-8). Also available in http://arxiv.org/abs/1312.2079.

See Also

cv.AWEnetCC

Examples

```
#For full data typically used for AFT models (using imputeYn (2015) package)
dat<-data(n=100, p=10, r=0, b1=c(rep(5,5),rep(0,5)), sig=1, Cper=0)

#This needs to run for generating weights of the observations
l<-mrbj(cbind(dat$y, dat$delta) ~ dat$x, mcsize=10, trace=FALSE, gehanonly=TRUE)

#WEnetCC.aft: weighted elastic net with censoring constraints
wt<-round(l$gehansd)
ft<-WEnetCC.aft(dat$x, dat$y, dat$delta, weight=wt, C=1, s=959596, lambda2=0.5)
ft
```

Index

∗Topic **AEnetCC.aft**
AEnetCC.aft, 5
∗Topic **AEnetCC**
cv.AWEnetCC, 9
∗Topic **AEnet**
cv.AWEnet, 7
∗Topic **AFT models**
AEnet.aft, 3
AEnetCC.aft, 5
WEnet.aft, 16
WEnetCC.aft, 17
∗Topic **Adaptive elastic net**
AEnet.aft, 3
∗Topic **Buckley-James method**
mrbj, 13
∗Topic **CV**
cv.AWEnet, 7
cv.AWEnetCC, 9
∗Topic **Elastic net**
mrbj, 13
∗Topic **Kaplan-Meier weights**
Enet.wls, 10
∗Topic **WEnetCC.aft**
WEnetCC.aft, 17
∗Topic **WEnetCC**
cv.AWEnetCC, 9
∗Topic **WEnet**
cv.AWEnet, 7
∗Topic **Weighted elastic net**
WEnet.aft, 16
∗Topic **dataset**
MCLcleaned, 11
∗Topic **elastic net**
Enet.wls, 10
∗Topic **imputeYn**
AdapEnetClass-package, 2
∗Topic **lars**
AdapEnetClass-package, 2
AEnet.aft, 3
AEnetCC.aft, 5
WEnet.aft, 16
WEnetCC.aft, 17

AdapEnetClass-package, 2
AEnet.aft, 3
AEnetCC.aft, 5

cv.AWEnet, 7
cv.AWEnetCC, 9

Enet.wls, 10

MCLcleaned, 11
mrbj, 13

plotObsEst, 14

WEnet.aft, 16
WEnetCC.aft, 17