Package 'PRIMsrc'

March 3, 2015

Type Package

Title PRIM Survival Regression Classification

Version 0.5.0

Date 2015-03-03

Author
Jean-Eudes Dazard [aut, cre], Michael Choe [ctb], Michael LeBlanc [ctb], Alberto Santana [ctb]

Maintainer Jean-Eudes Dazard <jxd101@case.edu>

Description Performs Bump Hunting by Patient Rule Induction Method in Survival, Regression and Classification settings.

Depends R (>= 3.0.2), parallel, survival, Hmisc, glmnet, MASS

URL https://github.com/jedazard/PRIMsrc

Repository PRIMsrc, GitHub, Inc.

License GPL (>= 3) | file LICENSE

LazyLoad yes

LazyData yes

Archs i386, x64

R topics documented:

`PRIMsrc-package` *Bump Hunting by Patient Rule Induction Method in Survival, Regression and Classification settings*

Description

Performs a Bump Hunting search by Patient Rule Induction Method. The method generates decision rules delineating a region in the predictor space, where the response is larger than its average over the entire space. The region is shaped as a hyperdimensional box that is not necessarily contiguous. Assumptions are that the multivariate input variables can be discrete or continuous and the univariate response variable can be discrete (Classification), continuous (Regression) or a time-to-event, possibly censored (Survival). It is intended to handle high-dimensional multivariate datasets, where the number of variables far exceeds that of the samples ($p \gg n$ paradigm).

Details

The current version is the initial development release that only includes the case of a survival response in either low ($p \leq n$) or high-dimensional situations ($p > n$). Ultimately, it will include all the features described above. New features will be added soon as they are available. At this point, the main function `sbh` depends on a internal cross-validated variable selection procedure by regularized Cox-regression from the R package **glmnet**.

The following describes only the end-user functions that are needed to run a complete procedure. The other internal subroutines are briefly documented in the manual, but are not to be called by the end-user at any time. For computational efficiency, some end-user functions offer a parallelization option that is done by passing a few parameters needed to configure a cluster. This is indicated by an asterisk (* = optionally involving cluster usage). The R functions are categorized as follows:

1. NEWS `PRIMsrc.news` **Function to display the NEWS file of the PRIMsrc package**

2. END-USER SURVIVAL BUMP HUNTING FUNCTION
   `sbh` (*) **Function for Cross-Validated Survival Bump Hunting**
   Main and unique end-user function for fitting a cross-validated survival bump hunting model. Returns cross-validated "PRSP" object, as generated by our Patient Recursive Survival Peeling or PRSP algorithm at each iteration of the peeling sequence (inner loop of the PRSP algorithm). The object contains cross-validated estimates of all the decision-rules of covariates and other statistical quantities of interest. It also enables displaying results graphically of/for model tuning/selection, all peeling trajectories, variable traces, and survival distributions (see plotting functions below for more details). The function offers a few options such as the type of cross-validation desired (K-fold (replicated)-averaged or-combined), peeling and optimization critera for model fitting, tuning and selection and a few more parameters for the PRSP algorithm. The function takes advantage of the R package **parallel** for efficient parallel execution. It allows users to create a cluster of workstations on a local and/or remote machine(s), enabling scaling-up to the number of specified CPU cores. Discrete (or nominal) variables should be made (or re-arranged into) ordinal variables.

3. END-USER PLOTS FOR MODEL VALIDATION CHECKING AND GRAPHICAL VISUALIZATION OF RESULTS
   `plot_profile` **Function for Model Validation Visualization and/or Checking**
   Plot the cross-validated profiles of user's choice of statistics among the Log Hazard Ratio (LHR), Log-Rank Test (LRT), or Concordance Error Rate (CER) as a function of the model tuning parameter, that is, the optimal number of peeling steps of the peeling sequence (inner loop of our PRSP algorithm). Model validation is done by applying the optimization criterion on the user's choice of specific statistic. The goal is to find the optimal value of the K-fold cross-validated number of steps by maximization of LHR or LRT, or minimization of CER. Currently, this done internally for visualization purposes, but it will ultimately offer the option to do be interactive with the end-user as well for parameter choosing/model selection.
   `plot_scatter` **Function for 2D Visualization of Data Scatter and Box Vertices**
   Plot in a plane the cross-validated Data Scatter and Box Vertices at a given peeling step of the peeling sequence (inner loop of our PRSP algorithm). The scatterplot is drawn on a graphical device with geometrically equal scales on the X and Y axes.
   `plot_boxtraj` **Function for Visualization of Peeling Trajectory/Profiles**
   Plot the cross-validated peeling trajectories/profiles of covariates used for peeling and other statistical quantities of interest at each iteration of the peeling sequence (inner loop of our PRSP algorithm). The plot includes box descriptive statistics (such as support), survival endpoint statistics (such as Maximum Event-Free Time (MEFT), Minimum Event-Free Probability (MEVP), LHR, LRT) and prediction performance (such as CER).
   `plot_boxtrace` **Function for Visualization of Covariates Traces**
   Plot the cross-validated trace curves of variable importance and variable usage of covariates used for peeling at each iteration of the peeling sequence (inner loop of our PRSP algorithm). The top plot shows the overlay of variable importance curves for each covariate. The bottom plot shows the overlay of variable usage curves for each covariate. It is a dicretized view of variable importance. Both point to the magnitude and order in which covariates are used along the peeling sequence.
   `plot_boxkm` **Function for Visualization of Survival Distributions**
   Plot the cross-validated Kaplan-Meir estimates of survival distributions for the highest risk (inbox) versus lower-risk (outbox) groups of samples at each iteration of the peeling sequence (inner loop of our PRSP algorithm). Step #0 always corresponds to the situation where the starting box covers the entire test-set data before peeling. Cross-validated LRT, LHR of inbox samples and log-rank p-values of separation are shown at the bottom of the plot with the corresponding peeling step. P-values are lower-bounded by the precision limit given by 1/#number of permutations (A).

4. END-USER DATASETS
   Synthetic.1, Synthetic.2, Synthetic.3, Synthetic.4, Synthetic.5 **Five Simulated Survival Models Datasets**
   Modeling survival models #1-5 with censoring as a regression function of some informative predictors, depending on the model used. In models where non-informative noisy variables were used, these variables were not part of the design matrix (models #2-3 and #5). In one example, the signal is limited to a box-shaped region R of the predictor space (model #4). In the last example, the signal is limited to 10% of the predictors in a $p > n$ situation (model #5). Survival time was generated from an exponential model with with rate parameter λ (and mean $\frac{1}{\lambda}$) according to a Cox-PH model with hazard exp(eta), where eta(.) is the regression function. Censoring indicator were generated from a uniform distribution on [0,3] (models #1-4) or [0,2] (model #5). In these synthetic examples, all covariates are continuous, i.i.d. from a multivariate uniform distribution on [0,1] (models #1-4) or from a multivariate standard normal distribution (model #5).
   Real.1 **Clinical Dataset**
   Publicly available dataset from the Women's Interagency HIV cohort Study (WIHS). Inclusion criteria of the study were that women at enrolment were (i) alive, (ii) HIV-1 infected, and (iii) free of clinical AIDS symptoms. Women were followed until the first of the following occurred: (i) treatment initiation (HAART), (ii) AIDS diagnosis, (iii) death, or administrative censoring. The studied outcomes were the competing risks "AIDS/Death (before HAART)" and "Treatment Initiation (HAART)". However, here, for simplification purposes, only the first of the two competing events (i.e. the time to AIDS/Death), was used in this dataset example. Likewise, the entire study enrolled 1164 women, but only the complete cases were used in this dataset example for simplification. Variables included history of Injection Drug Use ("IDU") at enrollment, African American ethnicity ("Race"), age ("Age"), and baseline CD4 count ("CD4"). The question in this dataset example was whether it is possible to achieve a prognostication of patients for AIDS and HAART.
   Real.2 **Large Gene Expression Dataset**
   Publicly available breast cancer gene expression profiling dataset from the Uppsala cohort study that enrolled 249 women. It is entitled: "Genetic Reclassification of Histologic Grade Delineates New Clinical Subtypes of Breast Cancer". The goal of the study was to provide a more objective measure of grade with prognostic benefit for patients with moderately differentiated grade II (G2) tumors. To that end, expression profiles of primary invasive breast tumors were analyzed on microarrays to find a gene expression signature capable of discerning tumors of grade I (G1) and grade III (G3) histology. In this dataset, only the Uppsala cohort and only the gene expression data was included although other clinical covariates are available as well. It represents a situation where the number of variables ($p = 22645$) dominates the number of observations ($p = 249$), or $p >> n$ case.

Known Bugs/Problems : None at this time.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.
- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

See Also

- `makeCluster` (R package **parallel**)
- `plot.survfit` (R package **survival**)
- `glmnet` (R package **glmnet**)

---

cbindlist *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: `cv.pval` and `cv.comb.box`. It is used to bind a list of matrices by columns (even of different number of rows) into a single matrix.

Usage

```
cbindlist(list, trunc)
```

Arguments

`list`	`List` of `numeric` `matrices`.
`trunc`	`Integer` scalar for truncation to the same number of rows.

---

cv.ave.box *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: `cv.box.rep` and `cv.null`

Usage

```
cv.ave.box(x, times, status,
           probval, timeval,
           varsign, selected, initcutpts,
           K, arg, seed)
```

Arguments

x	Numeric matrix of original covariates.
times	Numeric vector of original survival times.
status	Logical vector of original event indicators.
probval	Numeric scalar of the survival probability at which we want to get the endpoint box survival time.
timeval	Numeric scalar of the survival time at which we want to get the endpoint box survival probability.
varsign	numeric vector in {-1,+1} of directions of peeling for all variables.
selected	numeric vector giving the selected variable by regularized (Elastic-Net) Cox-regression.
initcutpts	numeric vector of initial box boundaries for each dimension.
K	Positive integer scalar of the number of folds for the cross-validation procedure.
arg	Character vector describing the parameters to use: • alpha = fraction to peel off at each step. • beta = minimum support size resulting from the peeling sequence. • minn = minimum number of observation in a box. • L = fixed peeling length. • peelcriterion in {"hr" (LHR), "lr" (LRT)}.
seed	Positive integer scalar for the seed.

---

cv.ave.fold *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: cv.ave.box

Usage

```
cv.ave.fold(x, times, status,
            probval, timeval,
            varsign, selected, initcutpts,
            K, arg, seed)
```

Arguments

x	Numeric matrix of original covariates.
times	Numeric vector of original survival times.
status	Logical vector of original event indicators.
probval	Numeric scalar of the survival probability at which we want to get the endpoint box survival time.
timeval	Numeric scalar of the survival time at which we want to get the endpoint box survival probability.

varsign	numeric vector in {-1,+1} of directions of peeling for all variables.
selected	numeric vector giving the selected variable by regularized (Elastic-Net) Cox-regression.
initcutpts	numeric vector of initial box boundaries for each dimension.
K	Positive integer scalar of the number of folds for the cross-validation procedure.
arg	Character vector describing the parameters to use: • alpha = fraction to peel off at each step. • beta = minimum support size resulting from the peeling sequence. • minn = minimum number of observation in a box. • L = fixed peeling length. • peelcriterion in {"hr" (LHR), "lr" (LRT)}.
seed	Positive integer scalar for the seed.

`cv.ave.peel` *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: `cv.ave.fold`

Usage

```
cv.ave.peel(traindata, trainstatus, traintime,
            testdata, teststatus, testtime,
            probval, timeval,
            varsign, selected, initcutpts,
            K, arg, seed)
```

Arguments

traindata	Numeric matrix of training covariates.
trainstatus	Logical vector of training event indicators.
traintime	Numeric vector of training survival times.
testdata	Numeric matrix of test covariates.
teststatus	Logical vector of test event indicators.
testtime	Numeric vector of test survival times.
varsign	numeric vector in {-1,+1} of directions of peeling for all variables.
selected	numeric vector giving the selected variable by regularized (Elastic-Net) Cox-regression.
initcutpts	numeric vector of initial box boundaries for each dimension.
probval	Numeric scalar of the survival probability at which we want to get the endpoint box survival time.
timeval	Numeric scalar of the survival time at which we want to get the endpoint box survival probability.

K	Positive integer scalar of the number of folds for the cross-validation procedure.
arg	Character vector describing the parameters to use: • alpha = fraction to peel off at each step. • beta = minimum support size resulting from the peeling sequence. • minn = minimum number of observation in a box. • L = fixed peeling length. • peelcriterion in {"hr" (LHR), "lr" (LRT)}.
seed	Positive integer scalar for the seed.

---

cv.box.rep *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: sbh

Usage

```
cv.box.rep(x, times, status,
           B, K, arg,
           cvtype,
           probval, timeval,
           varsign, selected, initcutpts,
           parallel, seed)
```

Arguments

x	Numeric matrix of original covariates.
times	Numeric vector of original survival times.
status	Logical vector of original event indicators.
B	Positive integer scalar of the number of replications of the cross-validation procedure.
K	Positive integer scalar of the number of folds for the cross-validation procedure.
arg	Character vector describing the parameters to use: • alpha = fraction to peel off at each step. • beta = minimum support size resulting from the peeling sequence. • minn = minimum number of observation in a box. • L = fixed peeling length. • peelcriterion in {"hr" (LHR), "lr" (LRT)}.
cvtype	Character vector describing the cross-validation technique in {"none", "averaged", "combined"}.
probval	Numeric scalar of the survival probability at which we want to get the endpoint box survival time.
timeval	Numeric scalar of the survival time at which we want to get the endpoint box survival probability.

varsign	numeric vector in {-1,+1} of directions of peeling for all variables.
selected	numeric vector giving the selected variable by regularized (Elastic-Net) Cox-regression.
initcutpts	numeric vector of initial box boundaries for each dimension.
parallel	Logical. Is parallel computing to be performed? Optional.
seed	Positive integer scalar of the user seed to reproduce the results.

---

cv.comb.box *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: cv.box.rep and cv.null

Usage

```
cv.comb.box(x, times, status,
            probval, timeval,
            varsign, selected, initcutpts,
            K, arg, seed)
```

Arguments

x	Numeric matrix of original covariates.
times	Numeric vector of original survival times.
status	Logical vector of original event indicators.
probval	Numeric scalar of the survival probability at which we want to get the endpoint box survival time.
timeval	Numeric scalar of the survival time at which we want to get the endpoint box survival probability.
varsign	numeric vector in {-1,+1} of directions of peeling for all variables.
selected	numeric vector giving the selected variable by regularized (Elastic-Net) Cox-regression.
initcutpts	numeric vector of initial box boundaries for each dimension.
K	Positive integer scalar of the number of folds for the cross-validation procedure.
arg	Character vector describing the parameters to use: • alpha = fraction to peel off at each step. • beta = minimum support size resulting from the peeling sequence. • minn = minimum number of observation in a box. • L = fixed peeling length. • peelcriterion in {"hr" (LHR), "lr" (LRT)}.
seed	Positive integer scalar for the seed.

`cv.comb.fold` *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: `cv.comb.box`

Usage

```
cv.comb.fold(x, times, status,
             varsign, selected, initcutpts,
             K, arg, seed)
```

Arguments

`x`	`Numeric matrix` of original covariates.
`times`	`Numeric vector` of original survival times.
`status`	`Logical vector` of original event indicators.
`varsign`	`numeric vector` in {-1,+1} of directions of peeling for all variables.
`selected`	`numeric vector` giving the selected variable by regularized (Elastic-Net) Cox-regression.
`initcutpts`	`numeric vector` of initial box boundaries for each dimension.
`K`	Positive `integer` scalar of the number of folds for the cross-validation procedure.
`arg`	`Character vector` describing the parameters to use: • `alpha` = fraction to peel off at each step. • `beta` = minimum support size resulting from the peeling sequence. • `minn` = minimum number of observation in a box. • `L` = fixed peeling length. • `peelcriterion` in {"hr" (LHR), "lr" (LRT)}.
`seed`	Positive `integer` scalar for the seed.

`cv.comb.peel` *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: `cv.comb.fold`

Usage

```
cv.comb.peel(traindata, trainstatus, traintime,
             testdata, teststatus, testtime,
             varsign, selected, initcutpts,
             K, arg, seed)
```

Arguments

traindata	Numeric matrix of training covariates.
trainstatus	Logical vector of training event indicators.
traintime	Numeric vector of training survival times.
testdata	Numeric matrix of test covariates.
teststatus	Logical vector of test event indicators.
testtime	Numeric vector of test survival times.
varsign	numeric vector in {-1,+1} of directions of peeling for all variables.
selected	numeric vector giving the selected variable by regularized (Elastic-Net) Cox-regression.
initcutpts	numeric vector of initial box boundaries for each dimension.
K	Positive integer scalar of the number of folds for the cross-validation procedure.
arg	Character vector describing the parameters to use: • alpha = fraction to peel off at each step. • beta = minimum support size resulting from the peeling sequence. • minn = minimum number of observation in a box. • L = fixed peeling length. • peelcriterion in {"hr" (LHR), "lr" (LRT)}.
seed	Positive integer scalar for the seed.

---

cv.folds *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: cv.ave.fold and cv.comb.fold

Usage

```
cv.folds(n, K, seed = NULL)
```

Arguments

n	Integer scalar giving the number of observations to be split into groups.
K	Integer giving the number of groups into which the observations should be randomly split. Setting K also specifies the type of folds to be generated. Possible types are 'random cross-validation', 'leave-one-out cross-validation', or no cross-validation: • K in {2,...,$n-1$} yields random cross-validation. • K = 1 yields no cross-validation. • K = n yields leave-one-out cross-validation.
seed	Integer scalar seed for RNG of random splitting of the data.

cv.null *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: cv.pval

Usage

```
cv.null(x, times, status,
        cvtype,
        varsign, selected, initcutpts,
        A, K, arg)
```

Arguments

x	Numeric matrix of original covariates.
times	Numeric vector of original survival times.
status	Logical vector of original event indicators.
cvtype	Character vector describing the cross-validation technique in {"none", "averaged", "combined"}.
varsign	numeric vector in {-1,+1} of directions of peeling for all variables.
selected	numeric vector giving the selected variable by regularized (Elastic-Net) Cox-regression.
initcutpts	numeric vector of initial box boundaries for each dimension.
A	Positive integer scalar of the number of permutations for the computation of cross-validated p-values. Defaults to 1000.
K	Positive integer scalar of the number of folds for the cross-validation procedure.
arg	Character vector describing the parameters to use: • alpha = fraction to peel off at each step. • beta = minimum support size resulting from the peeling sequence. • minn = minimum number of observation in a box. • L = fixed peeling length. • peelcriterion in {"hr" (LHR), "lr" (LRT)}.

cv.pval *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: sbh

Usage

```
cv.pval(x, times, status,
        cvtype,
        varsign, selected, initcutpts,
        A, K, arg, obs.chisq,
        parallel, conf)
```

Arguments

x	Numeric matrix of original covariates.
times	Numeric vector of original survival times.
status	Logical vector of original event indicators.
cvtype	Character vector describing the cross-validation technique in {"none", "averaged", "combined"}.
varsign	numeric vector in {-1,+1} of directions of peeling for all variables.
selected	numeric vector giving the selected variable by regularized (Elastic-Net) Cox-regression.
initcutpts	numeric vector of initial box boundaries for each dimension.
A	Positive integer scalar of the number of permutations for the computation of cross-validated p-values. Defaults to 1000.
K	Positive integer scalar of the number of folds for the cross-validation procedure.
arg	Character vector describing the parameters to use: • alpha = fraction to peel off at each step. • beta = minimum support size resulting from the peeling sequence. • minn = minimum number of observation in a box. • L = fixed peeling length. • peelcriterion in {"hr" (LHR), "lr" (LRT)}.
obs.chisq	Numeric vector of observed LRT values at each peeling step.
parallel	Logical. Is parallel computing to be performed? Optional. Defaults to FALSE.
conf	List of parameters for cluster configuration. Inputs for R package **parallel** function makeCluster (R package **parallel**) for cluster setup. Optional, defaults to NULL. See details for usage.

---

endpoints *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: cv.ave.peel and cv.comb.peel. Returns the maximum time value and the corresponding minimum survival probability for every box of the trajectory (box peeling sequence). Also returns the time value and/or the corresponding survival probability depending on what is specified by the user.

Usage

```
endpoints(ind, timemat, probmat, timeval, probval)
```

Arguments

ind	Numeric vector taking on values in {0,1}.
timemat	Numeric matrix of survival times for each peeling step (by rows) and test samples (by columns).
probmat	Numeric matrix of survival probabilities for each peeling step (by rows) and test samples (by columns).
probval	Numeric scalar of the survival probability at which we want to get the endpoint box survival time. Defaults to NULL.
timeval	Numeric scalar of the survival time at which we want to get the endpoint box survival probability. Defaults to NULL.

---

is.empty *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: endpoints, list2array, list2mat and cbindlist. Checks if object is empty. Represents the empty array, matrix, or vector of zero dimension or length. Often returned by expressions and functions whose value is undefined. It returns a ogical: TRUE if its argument is empty and FALSE otherwise.

Usage

```
is.empty(x)
```

Arguments

x	array, matrix or vector of any type.

---

lapply.array *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: sbh and cv.comb.box. Computes FUN element-wise on entries of a list of matrices (even of different row or column numbers).

Usage

```
lapply.array(X, trunc = NULL, sub = NULL,
             fill = NA, MARGIN = 1:2, FUN, ...)
```

Arguments

X	List of numeric matrices.
trunc	Integer scalar for truncation to the same number of rows.
sub	Integer scalar for reaching out to one sub-level of the list. Defaults to NULL.
fill	Numeric scalar to fill in the missing values. Defaults to NA.
MARGIN	Integer vector giving the subscripts which the function will be applied over. E.g., for a matrix 1 indicates rows, 2 indicates columns, c(1, 2) indicates rows and columns. Where X has named dimnames, it can be a character vector selecting dimension names. Defaults to 1:2.
FUN	The function to be applied. In the case of functions like +, %*%, etc., the function name must be backquoted or quoted.
...	Optional arguments to FUN.

---

lapply.mat *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: sbh, cv.comb.box and cv.ave.box. Compute FUN element-wise on entries of a list of vectors (even of different lengths).

Usage

```
lapply.mat(X, trunc = NULL, sub = NULL,
           fill = NA, MARGIN = 2, FUN, ...)
```

Arguments

X	List of numeric vectors.
trunc	Integer scalar for truncation to the same length.
sub	Integer scalar for reaching out to one sub-level of the list. Defaults to NULL.
fill	Numeric scalar to fill in the missing values. Defaults to NA.
MARGIN	Integer vector giving the subscripts which the function will be applied over. E.g., for a matrix 1 indicates rows, 2 indicates columns, c(1, 2) indicates rows and columns. Where X has named dimnames, it can be a character vector selecting dimension names. Defaults to 2.
FUN	The function to be applied. In the case of functions like +, %*%, etc., the function name must be backquoted or quoted.
...	Optional arguments to FUN.

list2array *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: sbh, cv.null and lapply.mat. Used to bind a list of matrices (even of different row or column numbers) by third dimension of a single 3D array.

Usage

```
list2array(list, trunc = NULL, sub = NULL, fill = NA)
```

Arguments

list	List of numeric matrices.
trunc	Integer scalar for truncation to the same number of rows.
sub	Integer scalar for reaching out to one sub-level of the list. Defaults to NULL.
fill	Numeric scalar to fill in the missing values. Defaults to NA.

list2mat *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: sbh, cv.null and lapply.mat. Used to bind a list of vectors (even of different length) by rows into a single matrix.

Usage

```
list2mat(list, trunc = NULL, sub = NULL, fill = NA)
```

Arguments

list	List of numeric vectors.
trunc	Integer scalar for truncation to the same length.
sub	Integer scalar for reaching out to one sub-level of the list. Defaults to NULL.
fill	Numeric scalar to fill in the missing values. Defaults to NA.

myround *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: sbh. Go to the nearest digit rounding. Note that for rounding off a 0.5, the "go to the even digit" standard is NOT used here.

Usage

```
myround(x, digits = 0)
```

Arguments

x	Numeric vector
digits	Integer scalar indicating the number of decimal points (precision) to be used. Defaults to 0.

peel.box *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: cv.ave.peel and cv.comb.peel.

Usage

```
peel.box(traindata, traintime, trainstatus,
         varsign, selected, initcutpts,
         arg, seed)
```

Arguments

traindata	Numeric matrix of training covariates.
traintime	Numeric vector of training survival times.
trainstatus	Logical vector of training event indicators.
varsign	numeric vector in {-1,+1} of directions of peeling for all variables.
selected	numeric vector giving the selected variable by regularized (Elastic-Net) Cox-regression.
initcutpts	numeric vector of initial box boundaries for each dimension.
arg	Character vector describing the parameters to use: • alpha = fraction to peel off at each step. • beta = minimum support size resulting from the peeling sequence. • minn = minimum number of observation in a box. • L = fixed peeling length. • peelcriterion in {"hr" (LHR), "lr" (LRT)}.
seed	Positive integer scalar of the user seed to reproduce the results.

Details

The maximal number of peeling steps (ncut) is determined either by `alpha` and `beta` metaparameters or the smallest possible fraction of the training data, i.e. $\frac{1}{n}$:

- $ceiling(\frac{\log(\texttt{beta})}{\log(1-\texttt{alpha})})$ if `alpha` and `beta` are fixed by user
- $ceiling(\frac{\log(1/n)}{\log(1-\texttt{alpha})})$ if `alpha` is fixed by user and `beta` is fixed by data
- $ceiling(\frac{\log(\texttt{beta})}{\log(1-(1/n))})$ if `alpha` is fixed by data and `beta` is fixed by user
- $ceiling(\frac{\log(1/n)}{\log(1-(1/n))})$ if `alpha` and `beta` are fixed by data

If `L` is not used to specify a fixed number of peeling steps (i.e. `NULL`), then `beta` is used as the stopping rule instead.

`plot_boxkm` *Function for Visualization of Survival Distributions*

Description

Plot the cross-validated Kaplan-Meir estimates of survival distributions for the highest risk (inbox) versus lower-risk (outbox) groups of samples at each iteration of the peeling sequence (inner loop of our PRSP algorithm).

Some plotting parameters are further defined in the function `plot.survfit` (R package **survival**).

Usage

```
plot_boxkm(peelobj,
           main = NULL, xlab = "Time", ylab = "Probability",
           precision, mark = 3, col = 1, lty = 1, lwd = 1, cex = 1,
           only.last = FALSE, nr = NULL, nc = NULL,
           device = NULL, file = "Survival Plots", path=getwd(),
           horizontal = TRUE, width = 11.5, height = 8.5, ...)
```

Arguments

`peelobj`	Object of class "PRSP" as generated by the main function sbh.
`main`	`Character vector`. Main Title. Defaults to `NULL`.
`xlab`	`Character vector`. X axis label. Defaults to "Time".
`ylab`	`Character vector`. Y axis label. Defaults to "Probability".
`precision`	Precision of cross-validated log-rank p-values of separation between two survival curves. Lower bounded by the value $\frac{1}{A}$ of the number of replication A.
`mark`	`Integer vector` of mark parameters, which will be used to label the curves. Defaults to 3.
`col`	`Integer vector` of specifying colors for each curve. Defaults to 1.
`lty`	`Integer vector` of specifying line types for each curve. Defaults to 1.
`lwd`	`Numeric vector` of values for line widths. Defaults to 1.
`cex`	`Numeric` scalar specifying the size of the marks. Defaults to 1.

only.last	Logical scalar defining whether only the last step of the peeling sequence should be plotted. Defaults to FALSE.
nr	Integer scalar of the number of rows in the plot. If NULL, defaults to 3.
nc	Integer scalar of the number of columns in the plot. If NULL, defaults to 4.
device	Graphic display device in {NULL, "PS", "PDF"}. Defaults to NULL (standard output screen). Currently implemented graphic display devices are "PS" (Postscript) or "PDF" (Portable Document Format).
file	File name for output graphic. Defaults to "Survival Plots".
path	Absolute path (without final (back)slash separator). Defaults to working directory path.
horizontal	Logical scalar. Orientation of the printed image. Defaults to TRUE, that is potrait orientation.
width	Numeric scalar. Width of the graphics region in inches. Defaults to 11.5.
height	Numeric scalar. Height of the graphics region in inches. Defaults to 8.5.
...	Generic arguments passed to other plotting functions, including plot.survfit (R package **survival**).

Details

Step #0 always corresponds to the situation where the starting box covers the entire test-set data before peeling. Cross-validated LRT, LHR of inbox samples and log-rank p-values of separation are shown at the bottom of the plot with the corresponding peeling step. P-values are lower-bounded by the precision limit given by 1/#number of permutations (A).

Value

Invisible. None. Displays the plot(s) on the specified device.

Note

End-user plotting function.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.

- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

See Also

- `plot.survfit` (R package **survival**)

Examples

```
#================================================
# Loading the library and its dependencies
#================================================
library("PRIMsrc")

## Not run:
    #=============================================================================
    # Simulated dataset #1 (n=250, p=3)
    # Replicated Combined Cross-Validation (RCCV)
    # Peeling criterion = LRT
    # Optimization criterion = LRT
    #=============================================================================
    plot_boxkm(peelobj = CVCOMBREP.synt1,
               main = paste("RCCV probability curves for model #1", sep=""),
               xlab = "Time", ylab = "Probability",
               cex = 1, yscale = 1, precision = 1/CVCOMBREP.synt1$A,
               device = NULL, file = "Survival Plots", path=getwd(),
               horizontal = TRUE, width = 11.5, height = 8.5)

## End(Not run)
```

`plot_boxtrace` *Function for Visualization of Covariates Traces*

Description

Plot the cross-validated modal trace curves of variable importance and variable usage of covariates used for peeling at each iteration of the peeling sequence (inner loop of our PRSP algorithm).

Usage

```
plot_boxtrace(peelobj,
              main = NULL,
              center = FALSE, scale = FALSE, hline = NULL,
              col = 1, lty = 1, lwd = 1, cex = 1,
              add.legend = FALSE, text.legend = NULL,
              device = NULL, file = "Covariate Trace Plots", path=getwd(),
              horizontal = FALSE, width = 8.5, height = 8.5, ...)
```

Arguments

peelobj	Object of class "PRSP" as generated by the main function sbh.
main	Character vector. Main Title. Defaults to NULL.
center	Logical scalar. Shall the data be centered before?
scale	Logical scalar. Shall the data be scaled before?
hline	Numeric scalar of where the data is centred. If specified (i.e. not NULL), an horizontal line about this value is added to the plot.
col	Line color of object peelobj. Defaults to 1.
lty	Line type of object peelobj. Defaults to 1.
lwd	Line width of object peelobj. Defaults to 1.
cex	Symbol expansion. Defaults to 1.
add.legend	Logical scalar. Should the legend be added to the current open graphics device?. Defaults to FALSE.
text.legend	Character vector of legend content. Defaults to NULL.
device	Graphic display device in {NULL, "PS", "PDF"}. Defaults to NULL (standard output screen). Currently implemented graphic display devices are "PS" (Postscript) or "PDF" (Portable Document Format).
file	File name for output graphic. Defaults to "Covariate Trace Plots".
path	Absolute path (without final (back)slash separator). Defaults to working directory path.
horizontal	Logical scalar. Orientation of the printed image. Defaults to FALSE, that is potrait orientation.
width	Numeric scalar. Width of the graphics region in inches. Defaults to 8.5.
height	Numeric scalar. Height of the graphics region in inches. Defaults to 8.5.
...	Generic arguments passed to other plotting functions.

Details

The trace plots limit the display of traces to those only covariates that are used for peeling.

Due to the variability induced by cross-validation and replication, it is possible that more than one variable be used for peeling at a given step. So, for simplicity of the trace plots, only the modal or majority vote trace value (over the folds and replications of the cross-validation) is plotted.

The top plot shows the overlay of variable importance curves for each covariate. The bottom plot shows the overlay of variable usage curves for each covariate. It is a dicretized view of variable importance.

Both point to the magnitude and order with which covariates are used along the peeling sequence.

Value

Invisible. None. Displays the plot(s) on the specified device.

Note

End-user plotting function.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.
- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

Examples

```
#=================================================
# Loading the library and its dependencies
#=================================================
library("PRIMsrc")

## Not run:
    #=============================================================================================
    # Simulated dataset #1 (n=250, p=3)
    # Replicated Combined Cross-Validation (RCCV)
    # Peeling criterion = LRT
    # Optimization criterion = LRT
    #=============================================================================
    plot_boxtrace(peelobj = CVCOMBREP.synt1,
                  main = paste("RCCV trace plots for model #1", sep=""),
                  center = TRUE, scale = FALSE,
                  hline = c(0,0,0), col = 2:4, lty = c(1,1,1),
                  device = NULL, file = "Covariate Trace Plots", path=getwd(),
                  horizontal = FALSE, width = 8.5, height = 8.5)

## End(Not run)
```

`plot_boxtraj` *Function for Visualization of Peeling Trajectory/Profiles*

Description

Plot the cross-validated peeling trajectories/profiles of covariates used for peeling and other statistical quantities of interest at each iteration of the peeling sequence (inner loop of our PRSP algorithm).

Usage

```
plot_boxtraj(peelobj,
             main = NULL, xlab = "Box Mass", ylab = "Variable Range",
             col = 1, lty = 1, lwd = 1, cex = 1,
             add.legend = FALSE, text.legend = NULL,
             nr = NULL, nc = NULL,
            device = NULL, file = "Covariate Trajectory Plots", path=getwd(),
             horizontal = FALSE, width = 8.5, height = 8.5, ...)
```

Arguments

peelobj	Object of class "PRSP" as generated by the main function sbh.
main	Character vector. Main Title. Defaults to NULL.
xlab	Character vector. X axis label. Defaults to "Box Mass".
ylab	Character vector. Y axis label. Defaults to "Variable Range".
	Plotting parameters for the three "PRSP"" objects:
col	Line color of object peelobj. Defaults to 1.
lty	Line type of object peelobj. Defaults to 1.
lwd	Line width of object peelobj. Defaults to 1.
cex	Symbol expansion. Defaults to 1.
add.legend	Logical scalar. Should the legend be added to the current open graphics device?. Defaults to FALSE.
text.legend	Character vector of legend content. Defaults to NULL.
nr	Integer scalar of the number of rows in the plot. If NULL, defaults to 3.
nc	Integer scalar of the number of columns in the plot. If NULL, defaults to 3.
device	Graphic display device in {NULL, "PS", "PDF"}. Defaults to NULL (standard output screen). Currently implemented graphic display devices are "PS" (Postscript) or "PDF" (Portable Document Format).
file	File name for output graphic. Defaults to "Covariate Trajectory Plots".
path	Absolute path (without final (back)slash separator). Defaults to working directory path.
horizontal	Logical scalar. Orientation of the printed image. Defaults to FALSE, that is potrait orientation.
width	Numeric scalar. Width of the graphics region in inches. Defaults to 8.5.
height	Numeric scalar. Height of the graphics region in inches. Defaults to 8.5.
...	Generic arguments passed to other plotting functions.

Details

The plot limits the display of trajectories to those only covariates that are used for peeling.

The plot includes box descriptive statistics (such as support), survival endpoint statistics (such as Maximum Event-Free Time (MEFT), Minimum Event-Free Probability (MEVP), LHR, LRT) and prediction performance (such as CER).

Value

Invisible. None. Displays the plot(s) on the specified device.

Note

End-user plotting function.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.
- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

Examples

```
#==================================================
# Loading the library and its dependencies
#==================================================
library("PRIMsrc")

## Not run:
    #============================================================================
    # Simulated dataset #1 (n=250, p=3)
    # Replicated Combined Cross-Validation (RCCV)
    # Peeling criterion = LRT
    # Optimization criterion = LRT
    #============================================================================
    plot_boxtraj(peelobj = CVCOMBREP.synt1,
                 main = paste("RCCV peeling trajectories for model #1", sep=""),
                 col = 2:4, lty = c(1,1,1),
                 device = NULL, file = "Covariate Trajectory Plots", path=getwd(),
                 horizontal = FALSE, width = 8.5, height = 8.5)

## End(Not run)
```

plot_profile *Function for Model Validation Visualization and/or Checking*

Description

Plot the cross-validated profiles of user's choice of statistics among the Log Hazard Ratio (LHR), Log-Rank Test (LRT), or Concordance Error Rate (CER) as a function of the model tuning parameter, that is, the optimal number of peeling steps of the peeling sequence (inner loop of our PRSP algorithm).

Usage

```
plot_profile(peelobj,
             main = NULL, xlab = "Peeling Steps", ylab = "Mean Profiles",
             add.sd = TRUE, add.legend = TRUE, add.profiles = TRUE,
             pch = 20, col = 1, lty = 1, lwd = 2, cex = 2,
             device = NULL, file = "Profile Plot", path=getwd(),
             horizontal = FALSE, width = 8.5, height = 5, ...)
```

Arguments

peelobj	Object of class "PRSP" as generated by the main function sbh.
main	Character vector. Main Title. Defaults to NULL.
xlab	Character vector. X axis label. Defaults to "Peeling Steps".
ylab	Character vector. Y axis label. Defaults to "Mean Profiles".
add.sd	Logical scalar. Shall the standard error bars be plotted? Defaults to TRUE.
add.legend	Logical scalar. Shall the legend be plotted? Defaults to TRUE.
add.profiles	Logical scalar. Shall the individual profiles (for all replicates) be plotted? Defaults to TRUE.
pch	Symbol number for all the profiles. Defaults to 20.
col	Line color of the mean profile. Defaults to 1. If more than eight profiles are plotted, line colors will be recycled.
lty	Line type of the mean profile. Defaults to 1.
lwd	Line width of the mean profile. Defaults to 2.
cex	Symbol expansion for all the profiles. Defaults to 2.
device	Graphic display device in {NULL, "PS", "PDF"}. Defaults to NULL (standard output screen). Currently implemented graphic display devices are "PS" (Postscript) or "PDF" (Portable Document Format).
file	File name for output graphic. Defaults to "Profile Plot".
path	Absolute path (without final (back)slash separator). Defaults to working directory path.
horizontal	Logical scalar. Orientation of the printed image. Defaults to FALSE, that is potrait orientation.
width	Numeric scalar. Width of the graphics region in inches. Defaults to 8.5.
height	Numeric scalar. Height of the graphics region in inches. Defaults to 5.
...	Generic arguments passed to other plotting functions.

Details

Model validation is done by applying the optimization criterion on the user's choice of specific statistic. The goal is to find the optimal value of the K-fold cross-validated number of steps by maximization of LHR or LRT, or minimization of CER.

Currently, this done internally for visualization purposes, but it will ultimately offer the option to do be interactive with the end-user as well for parameter choosing/model selection.

Value

Invisible. None. Displays the plot(s) on the specified `device`.

Note

End-user plotting function.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.
- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

Examples

```
#=================================================
# Loading the library and its dependencies
#=================================================
library("PRIMsrc")

## Not run:
    #=============================================================================
    # Simulated dataset #1 (n=250, p=3)
    # Replicated Combined Cross-Validation (RCCV)
    # Peeling criterion = LRT
    # Optimization criterion = LRT
    #=============================================================================
    plot_profile(peelobj = CVCOMBREP.synt1,
                 main = "RCCV tuning profiles for model #1",
```

```
                xlab = "Peeling Steps", ylab = "Mean Profiles",
                pch=20, col="black", lty=1, lwd=2, cex=2,
                add.sd = TRUE, add.legend = TRUE, add.profiles = TRUE,
                device = NULL, file = "Profile Plot", path=getwd(),
                horizontal = FALSE, width = 8.5, height = 5)

## End(Not run)
```

---

plot_scatter *Function for 2D Visualization of Data Scatter and Box Vertices*

Description

Plot in a plane the cross-validated Data Scatter and Box Vertices at a given peeling step of the peeling sequence (inner loop of our PRSP algorithm).

Usage

```
plot_scatter(peelobj,
             main = NULL,
             proj = c(1,2), splom = TRUE, boxes = FALSE,
             steps = peelobj$cvfit$cv.nsteps,
             add.legend = TRUE, pch = 16, cex = 0.7, col = 1,
             box.col = 2, box.lty = 2, box.lwd = 1,
             device = NULL, file = "Scatter Plot", path=getwd(),
             horizontal = FALSE, width = 5, height = 5, ...)
```

Arguments

peelobj	Object of class "PRSP" as generated by the main function sbh.
main	Character vector. Main Title. Defaults to NULL.
proj	Integer vector of length two, specifying the two dimensions of the projection plane. Defaults to c(1,2).
splom	Logical scalar. Shall the scatter plot of points inside the box(es) be plotted? Default to TRUE.
boxes	Logical scalar. Shall the box vertices be plotted or just the scatter of points? Default to FALSE.
steps	Integer vector. Vector of peeling steps at which one wants to plot the box vertices. Defaults to the last peeling step only.
add.legend	Logical scalar. Shall the legend of steps numbers be plotted? Defaults to TRUE.
pch	Symbol number for the scatter plot. Defaults to 16.
cex	Symbol expansion. Defaults to 0.7.
col	Symbol color. Defaults to 1.
box.col	Line color of the box vertices. Defaults to 2.
box.lty	Line type of box vertices. Defaults to 2.
box.lwd	Line width of box vertices. Defaults to 1.

device	Graphic display device in {NULL, "PS", "PDF"}. Defaults to NULL (standard output screen). Currently implemented graphic display devices are "PS" (Postscript) or "PDF" (Portable Document Format).
file	File name for output graphic. Defaults to "Scatter Plot".
path	Absolute path (without final (back)slash separator). Defaults to working directory path.
horizontal	Logical scalar. Orientation of the printed image. Defaults to FALSE, that is potrait orientation.
width	Numeric scalar. Width of the graphics region in inches. Defaults to 5.
height	Numeric scalar. Height of the graphics region in inches. Defaults to 5.
...	Generic arguments passed to other plotting functions.

Details

The scatterplot is drawn on a graphical device with geometrically equal scales on the X and Y axes.

Value

Invisible. None. Displays the plot(s) on the specified device.

Note

End-user plotting function.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.
- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

Examples

```
#================================================
# Loading the library and its dependencies
#================================================
library("PRIMsrc")

## Not run:
    #=============================================================================
    # Simulated dataset #1 (n=250, p=3)
    # Replicated Combined Cross-Validation (RCCV)
    # Peeling criterion = LRT
    # Optimization criterion = LRT
    #=============================================================================
    plot_scatter(peelobj=CVCOMBREP.synt1,
                 main = paste("Scatter plot for model #1", sep=""),
                 proj = c(1,2), splom = TRUE, boxes = TRUE,
                 steps = CVCOMBREP.synt1$cvfit$cv.nsteps,
                 add.legend = TRUE, cex = 0.5,
                 box.col = "red", box.lty = 3, box.lwd = 1,
                 device = NULL, file = "Scatter Plot", path=getwd(),
                 horizontal = FALSE, width = 5, height = 5)

## End(Not run)
```

`PRIMsrc.news` *Function to Display the NEWS File*

Description

Function to display the log file of updates of the **PRIMsrc** package.

Usage

```
PRIMsrc.news(...)
```

Arguments

`...`	Further arguments passed to or from other methods.

Value

None.

Note

End-user function.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.
- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

`Real.1` *Real Dataset #1: Clinical Dataset ($p < n$ case)*

Description

Publicly available dataset from the Women's Interagency HIV cohort Study (WIHS). Inclusion criteria of the study were that women at enrolment were (i) alive, (ii) HIV-1 infected, and (iii) free of clinical AIDS symptoms. Women were followed until the first of the following occurred: (i) treatment initiation (HAART), (ii) AIDS diagnosis, (iii) death, or administrative censoring. The studied outcomes were the competing risks "AIDS/Death (before HAART)" and "Treatment Initiation (HAART)". However, here, for simplification purposes, only the first of the two competing events (i.e. the time to AIDS/Death), was used in this dataset example. Likewise, the entire study enrolled 1164 women, but only the complete cases were used in this dataset example for simplification. Variables included history of Injection Drug Use ("IDU") at enrollment, African American ethnicity ("Race"), age ("Age"), and baseline CD4 count ("CD4"). The question in this dataset example was whether it is possible to achieve a prognostication of patients for AIDS and HAART. See Bacon et al. (2005) and the WIHS website for more details.

Usage

```
Real.1
```

Format

Dataset consists of a `numeric` `data.frame` containing $n = 485$ complete observations (samples) by rows and $p = 6$ clinical variables by columns ($p < n$ case), including the censoring indicator and (censored) time-to-event variables. It comes as a compressed Rda data file.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

Source

See real clinical data application in Dazard et al., 2015.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.
- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.
- Bacon M.C, von Wyl V., Alden C. et al. 2005. "*Semi-Supervised Methods to Predict Patient Survival from Gene Expression Data.*" Clin Diagn Lab Immunol 12(9): 1013-1019.

See Also

http://statepiaps.jhsph.edu/wihs/

Real.2 *Real Dataset #2: Large Gene Expression Dataset ($p >> n$ case)*

Description

Publicly available breast cancer gene expression profiling dataset from the Uppsala cohort study. It is entitled: "Genetic Reclassification of Histologic Grade Delineates New Clinical Subtypes of Breast Cancer". The goal of the study was to provide a more objective measure of grade with prognostic benefit for patients with moderately differentiated grade II (G2) tumors. To that end, expression profiles of primary invasive breast tumors were analyzed on microarrays to find a gene expression signature capable of discerning tumors of grade I (G1) and grade III (G3) histology. In this dataset, only the Uppsala cohort ($n = 249$) and only the gene expression data was included although other clinical covariates are available as well. It contains $p = 22647$ mRNA measurements from the Affymetrix Human Genome U133A Array platform on $n = 249$ samples. The data was left with $n = 177$ samples after removal of outliers and incomplete observations and $p = 22577$ variables after removal of Affymetrix controls. See Ivshina et al. (2005) and Gene Expression Omnibus database repository (Accession number: #GSE4922) for more details.

Usage

```
Real.2
```

Format

Dataset consists of a `numeric` `data.frame` containing $n = 177$ complete observations (samples) by rows and $p = 22577$ variables by columns ($p >> n$ case), after including the censoring indicator and (censored) time-to-event variables. It comes as a compressed Rda data file.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

Source

See real clinical data application in Dazard et al., 2015.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.
- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.
- Ivshina AV, George J, Senko O, Mow B et al. (2006). "*Genetic reclassification of histologic grade delineates new clinical subtypes of breast cancer.* Cancer Res 66(21):10292-301. PMID: 17079448

See Also

Gene Expression Omnibus (GEO) database. Accession number: #GSE4922 http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE4922

sbh *Function for Cross-Validated Survival Bump Hunting*

Description

Main and unique end-user function for fitting a cross-validated survival bump hunting model. Returns cross-validated "PRSP" object, as generated by our Patient Recursive Survival Peeling or PRSP algorithm at each iteration of the peeling sequence (inner loop of the PRSP algorithm).

Usage

```
sbh(dataset, discr, B = 10, K = 5, A = 1000, cpv = FALSE,
    cvtype = "combined", cvcriterion = "lrt",
    arg = "beta=0.05,alpha=0.1,minn=10,L=NULL,peelcriterion=\"lr\"",
    probval = NULL, timeval = NULL,
    parallel = FALSE, conf = NULL, seed = NULL)
```

Arguments

dataset	data.frame or numeric matrix of input dataset containing the observed survival and status indicator variables in the first two columns, respectively.
discr	Logical vector describing what covariates are discrete. Defaults to logical(ncol(dataset)-2).
B	Positive integer scalar of the number of replications of the cross-validation procedure. Defaults to 10.
K	Positive integer scalar of the number of folds for the cross-validation procedure. Defaults to 5.
A	Positive integer scalar of the number of permutations for the computation of cross-validated p-values. Defaults to 1000.
cpv	logical scalar. Flag for computation of cross-validated p-values. Defaults to FALSE. If computation of cross-validated p-value is deired, then running with the parallelization option is strongly advised, as it may take a while otherwise.
cvtype	Character vector describing the cross-validation technique in {"none", "averaged", "combined"}. Defaults to "combined".
cvcriterion	character vector describing the cross-validation optimization criterion in {"lhr", "lrt", "cer"}. Defaults to "lrt". Automatically set to NULL if cvtype="none".
arg	Character vector describing the PRSP parameters: • alpha = fraction to peel off at each step. Defaults to 0.1. • beta = minimum support size resulting from the peeling sequence. Defaults to 0.05. • minn = minimum number of observation in a box. Defaults to 10. • L = fixed peeling length. Defaults to NULL. • peelcriterion in {"hr" (LHR), "lr" (LRT)}. Defaults to "lr". Note that the parameters in arg come as a string of charaters between double quotes, where all parameter evaluations are separated by comas (see example).
probval	Numeric scalar of the survival probability at which we want to get the endpoint box survival time. Defaults to NULL.

timeval	Numeric scalar of the survival time at which we want to get the endpoint box survival probability. Defaults to NULL.
parallel	Logical. Is parallel computing to be performed? Optional. Defaults to FALSE.
conf	List of parameters for cluster configuration. Inputs for R package **parallel** function makeCluster (R package **parallel**) for cluster setup. Optional, defaults to NULL. See details for usage.
seed	Positive integer scalar of the user seed to reproduce the results.

Details

The function depends at this point on an internal cross-validated variable selection procedure by regularized Cox-regression from the R package **glmnet**.

The PRSP object contains cross-validated estimates of all the decision-rules of covariates and other statistical quantities of interest. It also enables displaying results graphically of/for model tuning/selection, all peeling trajectories, variable traces, and survival distributions (see plotting functions below for more details).

The function offers a number of options for the type of cross-validation desired: K-fold (replicated)-averaged or-combined, as well as peeling and optimization critera for model fitting, tuning and selectio and a few more parameters for the PRSP algorithm.

The function takes advantage of the R package **parallel**, which allows users to create a cluster of workstations on a local and/or remote machine(s), enabling scaling-up with the number of CPU cores specified and efficient parallel execution. Discrete (or nominal) variables should be made (or re-arranged into) ordinal variables.

It is required to use the parallelization and a hyperperformance cluster of workstations to run the computation of cross-validated p-values in the case of large n and/or large ($p > n$) or very large ($p >> n$) datasets.

To run a parallel session (and parallel RNG) of the PRIMsrc procedures (parallel=TRUE), argument conf is to be specified (i.e. non NULL). It must list the specifications of the folowing parameters for cluster configuration: "names", "cpus", "type", "homo", "verbose", "outfile". These match the arguments described in function makeCluster of the R package **parallel**. All fields are required to properly configure the cluster, except for "names" and "cpus", which are the values used alternatively in the case of a cluster of type "SOCK" (socket), or in the case of a cluster of type other than "SOCK" (socket), respectively. See examples below.

- "names": names : character vector specifying the host names on which to run the job. Could default to a unique local machine, in which case, one may use the unique host name "localhost". Each host name can potentially be repeated to the number of CPU cores available on the corresponding machine.
- "cpus": spec : integer scalar specifying the total number of CPU cores to be used across the network of available nodes, counting the workernodes and masternode.
- "type": type : character vector specifying the cluster type ("SOCK", "PVM", "MPI").
- "homo": homogeneous : logical scalar to be set to FALSE for inhomogeneous clusters.
- "verbose": verbose : logical scalar to be set to FALSE for quiet mode.
- "outfile": outfile : character vector of the output log file name for the workernodes.

Note that argument B is internally reset to conf$cpus*ceiling(B/conf$cpus) in case the parallelization is used (i.e. conf is non NULL), where conf$cpus denotes the total number of CPUs to be used (see above).

The actual creation of the cluster, its initialization, and closing are all done internally. In addition, when random number generation is needed, the creation of separate streams of parallel

RNG per node is done internally by distributing the stream states to the nodes (For more details see function `makeCluster` (R package **parallel**) and/or http://www.stat.uiowa.edu/~luke/R/cluster/cluster.html.

The use of a seed allows to reproduce the results within the same type of session: the same seed will reproduce the same results within a non-parallel session or within a parallel session, but it will not necessarily give the exact same results (up to sampling variability) between a non-parallelized and parallelized session due to the difference of management of the seed between the two (see parallel RNG and value of retuned seed below).

Value

Object of `class` "PRSP" (Patient Recursive Survival Peeling) `List` containing the following 18 fields:

`x`	`numeric matrix` of original covariates.
`times`	`numeric vector` of observed failure / survival times.
`status`	`numeric vector` of observed event indicator in {1,0}.
`B`	positive `integer` of the number of replications used in the cross-validation procedure.
`K`	positive `integer` of the number of folds used in the cross-validation procedure.
`A`	positive `integer` of the number of permutations used for the computation of cross-validated p-values.
`cpv`	`logical` scalar returned flag of computation of cross-validated p-values.
`cvtype`	`character vector` of the cross-validation technique used.
`cvcriterion`	`character vector` of cross-validation optimization criterion used.
`varsign`	`numeric vector` in {-1,+1} of directions of peeling for all variables.
`selected`	`numeric vector` giving the selected variable by regularized (Elastic-Net) Cox-regression.
`used`	`numeric vector` giving the variables used for peeling.
`arg`	`character vector` of the parameters used:
`probval`	`Numeric` scalar of survival probability used.
`timeval`	`Numeric` scalar of survival time used.
`cvfit`	`List` with 7 fields of cross-validated estimates:

- cv.maxsteps: `numeric` scalar of maximal ceiled-mean of number of peeling steps over the replicates
- cv.nsteps: `numeric` scalar of optimal number of peeling steps according to the optimization criterion
- cv.trace: `list` of `numeric matrix` and `numeric vector` of variable usage traces or modal trace values at each step
- cv.boxind: `logical matrix` in `TRUE`, `FALSE` of sample box membership indicator (columns) by peeling steps (rows)
- cv.rules: `data.frame` of decision rules on the variable (columns) by peeling steps (rows)
- cv.stats: `numeric matrix` of box quantities of interest (columns) by peeling steps (rows)
- cv.pval: `numeric vector` of cross-validated log-rank p-values of seprartion of survival distributions

cvprofiles	List (B) of numeric vectors, one for each replicate, of the cross-validated statistic used in the optimization criterion (set by user) as function of the number of peeling steps.
plot	logical scalar of the returned flag for plotting results (TRUE if CV successful).
seed	User seed(s) used: integer of a single value, if parallelization is used integer vector of values, one for each replication, if parallelization is not used.

Note

Unique end-user function for fitting the Survival Bump Hunting model.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.
- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

See Also

- makeCluster (R package **parallel**)
- cv.glmnet (R package **glmnet**)
- glmnet (R package **glmnet**)

Examples

```
#====================================================
# Loading the library and its dependencies
#====================================================
library("PRIMsrc")

## Not run:
    #====================================================
    # PRIMsrc package news
    #====================================================
    PRIMsrc.news()
```

```
    #=================================================
    # PRIMsrc package citation
    #=================================================
    citation("PRIMsrc")

    #=================================================
    # Use of two synthetic and two real datasets
    # Use help for descriptions
    #=================================================
    data("Synthetic.1", "Synthetic.5", "Real.1", "Real.2", package="PRIMsrc")
    ?Synthetic.1
    ?Synthetic.5
    ?Real.1
    ?Real.2

## End(Not run)

#=================================================
# Simulated dataset #1 (n=250, p=3)
# Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
# Without computation of cross-validated p-values
#=================================================
CVCOMBREP.synt1 <- sbh(dataset = Synthetic.1,
                       cvtype = "combined", cvcriterion = "lrt",
                       B = 5, K = 5, cpv = FALSE, probval = 0.5,
                       arg = "beta=0.05,
                              alpha=0.1,
                              minn=10,
                              L=NULL,
                              peelcriterion=\"lr\"",
                       parallel = FALSE, conf = NULL, seed = 123)

# selected variables:
selected <- CVCOMBREP.synt1$selected
selected
# variables used for peeling:
used <- CVCOMBREP.synt1$used
used
# some output results:
CVCOMBREP.synt1$cvfit$cv.maxsteps
CVCOMBREP.synt1$cvfit$cv.nsteps
CVCOMBREP.synt1$cvfit$cv.trace
CVCOMBREP.synt1$cvfit$cv.rules$frame[,used]
round(CVCOMBREP.synt1$cvfit$cv.stats$mean,2)

#=================================================
# Simulated dataset #5 (n=100, p=1000)
# Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
# Without computation of cross-validated p-values
#=================================================
CVCOMBREP.synt5 <- sbh(dataset = Synthetic.5,
```

```
                        cvtype = "combined", cvcriterion = "lrt",
                        B = 5, K = 5, cpv = FALSE, probval = 0.5,
                        arg = "beta=0.05,
                               alpha=0.1,
                               minn=10,
                               L=NULL,
                               peelcriterion=\"lr\"",
                        parallel = FALSE, conf = NULL, seed = 123)

# selected variables:
selected <- CVCOMBREP.synt5$selected
selected
# variables used for peeling:
used <- CVCOMBREP.synt5$used
used
# some output results:
CVCOMBREP.synt5$cvfit$cv.maxsteps
CVCOMBREP.synt5$cvfit$cv.nsteps
CVCOMBREP.synt5$cvfit$cv.trace
CVCOMBREP.synt5$cvfit$cv.rules$frame[,used]
round(CVCOMBREP.synt5$cvfit$cv.stats$mean,2)

#=================================================
# Real dataset #1 (n=485, p=4)
# Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
# Without computation of cross-validated p-values
#=================================================
CVCOMBREP.real1 <- sbh(dataset = Real.1,
                        discr = c(0,1,1,0),
                        cvtype = "combined", cvcriterion = "lrt",
                        B = 5, K = 5, cpv = FALSE, probval = 0.5,
                        arg = "beta=0.05,
                               alpha=0.1,
                               minn=10,
                               L=NULL,
                               peelcriterion=\"lr\"",
                        parallel = FALSE, conf = NULL, seed = 123)

# selected variables:
selected <- CVCOMBREP.real1$selected
selected
# variables used for peeling:
used <- CVCOMBREP.real1$used
used
# some output results:
CVCOMBREP.real1$cvfit$cv.maxsteps
CVCOMBREP.real1$cvfit$cv.nsteps
CVCOMBREP.real1$cvfit$cv.trace
CVCOMBREP.real1$cvfit$cv.rules$frame[,used]
round(CVCOMBREP.real1$cvfit$cv.stats$mean,2)

## Not run:
    #=================================================
    # Examples of parallelization below with
```

```
# a SOCKET or MPI cluster configuration
#==================================================
# 1- WINDOWS multicores PC with SOCKET communication
#    With a 2-Quad (8-CPUs) PC
#==================================================
if (.Platform$OS.type == "windows") {
    cpus <- detectCores()
    conf <- list("names" = rep("localhost", cpus),
                 "cpus" = cpus,
                 "type" = "SOCK",
                 "homo" = TRUE,
                 "verbose" = TRUE,
                 "outfile" = "")
}
#==================================================
# 2- LINUX multinodes cluster with SOCKET communication
#    with 4-nodes (32-CPUs) cluster
#    with 1 masternode and 3 workernodes
#    All hosts run identical setups
#    Same number of core CPUs (8) per node
#==================================================
if (.Platform$OS.type == "unix") {
    masterhost <- Sys.getenv("HOSTNAME")
    slavehosts <- c("compute-0-0", "compute-0-1", "compute-0-2")
    nodes <- length(slavehosts) + 1
    cpus <- 8
    conf <- list("names" = c(rep(masterhost, cpus),
                             rep(slavehosts, cpus)),
                 "cpus" = nodes * cpus,
                 "type" = "SOCK",
                 "homo" = TRUE,
                 "verbose" = TRUE,
                 "outfile" = "")
}
#==================================================
# 3- LINUX multinodes cluster with MPI communication
#    Here, a file named ".nodes" (e.g. in the home directory)
#    must contain the list of nodes of the cluster
#==================================================
if (.Platform$OS.type == "unix") {
    hosts <- scan(file=paste(Sys.getenv("HOME"), "/.nodes", sep=""),
                  what="",
                  sep="\n")
    hostnames <- unique(hosts)
    nodes <- length(hostnames)
    cpus <-  length(hosts)/length(hostnames)
    conf <- list("cpus" = nodes * cpus,
                 "type" = "MPI",
                 "homo" = TRUE,
                 "verbose" = TRUE,
                 "outfile" = "")
}

#==================================================
# Simulated dataset #1 (n=250, p=3)
# Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
```

```
# Optimization criterion = LRT
# With parallelization
# With computation of cross-validated p-values
#=================================================
CVCOMBREP.synt1 <- sbh(dataset = Synthetic.1,
                       cvtype = "combined", cvcriterion = "lrt",
                       B = 5, K = 5, A = 1024, cpv = TRUE, probval = 0.5,
                       arg = "beta=0.05,
                              alpha=0.1,
                              minn=10,
                              L=NULL,
                              peelcriterion=\"lr\"",
                       parallel = TRUE, conf = conf, seed = 123)

# selected variables:
selected <- CVCOMBREP.synt1$selected
selected
# variables used for peeling:
used <- CVCOMBREP.synt1$used
used
# some output results:
CVCOMBREP.synt1$cvfit$cv.maxsteps
CVCOMBREP.synt1$cvfit$cv.nsteps
CVCOMBREP.synt1$cvfit$cv.trace
CVCOMBREP.synt1$cvfit$cv.pval
CVCOMBREP.synt1$cvfit$cv.rules$frame[,used]
round(CVCOMBREP.synt1$cvfit$cv.stats$mean,2)

#=================================================
# Real dataset #2 (n=177, p=22577)
# Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# With parallelization
# Without computation of cross-validated p-values
#=================================================
p <- ncol(Real.2) - 2
CVCOMBREP.real2 <- sbh(dataset = Real.2,
                       discr = rep(0,p),
                       cvtype = "combined", cvcriterion = "lrt",
                       B = 5, K = 5, cpv = FALSE, probval = 0.5,
                       arg = "beta=0.05,
                              alpha=0.1,
                              minn=10,
                              L=NULL,
                              peelcriterion=\"lr\"",
                       parallel = TRUE, conf = conf, seed = 123)

# selected variables:
selected <- CVCOMBREP.real2$selected
selected
# variables used for peeling:
used <- CVCOMBREP.real2$used
used
# some output results:
CVCOMBREP.real2$cvfit$cv.maxsteps
CVCOMBREP.real2$cvfit$cv.nsteps
```

```
CVCOMBREP.real2$cvfit$cv.trace
CVCOMBREP.real2$cvfit$cv.rules$frame[,used]
round(CVCOMBREP.real2$cvfit$cv.stats$mean,2)

## End(Not run)
```

`Synthetic.1` *Synthetic Dataset #1: $p < n$ case*

Description

Modeling survival model #1 as described in Dazard et al. (2015) with censoring. Here, the regression function uses all of the predictors, which are also part of the design matrix. Survival time was generated from an exponential model with rate parameter λ (and mean $\frac{1}{\lambda}$) according to a Cox-PH model with hazard exp(eta), where eta(.) is the regression function. Censoring indicator were generated from a uniform distribution on [0, 3]. In this synthetic example, all covariates are continuous, i.i.d. from a multivariate uniform distribution on [0, 1].

Usage

```
Synthetic.1
```

Format

Each dataset consists of a `numeric` `matrix` containing $n = 250$ observations (samples) by rows and $p = 5$ variables by columns ($p < n$ case), including the censoring indicator and (censored) time-to-event variables. It comes as a compressed Rda data file.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

Source

See simulated survival model #1 in Dazard et al., 2015.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.

- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

`Synthetic.2` *Synthetic Dataset #2: $p < n$ case*

Description

Modeling survival model #2 as described in Dazard et al. (2015) with censoring. Here, the regression function uses some informative predictors. The rest represent un-informative noisy variables, which are not part of the design matrix. Survival time was generated from an exponential model with rate parameter λ (and mean $\frac{1}{\lambda}$) according to a Cox-PH model with hazard exp(eta), where eta(.) is the regression function. Censoring indicator were generated from a uniform distribution on [0, 3]. In this synthetic example, all covariates are continuous, i.i.d. from a multivariate uniform distribution on [0, 1].

Usage

```
Synthetic.2
```

Format

Each dataset consists of a `numeric matrix` containing $n = 250$ observations (samples) by rows and $p = 5$ variables by columns ($p < n$ case), including the censoring indicator and (censored) time-to-event variables. It comes as a compressed Rda data file.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

Source

See simulated survival model #2 in Dazard et al., 2015.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.
- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

Synthetic.3 *Synthetic Dataset #3: $p < n$ case*

Description

Modeling survival model #3 as described in Dazard et al. (2015) with censoring. Here, the regression function does not include any of the predictors. This means that none of the variables is informative (noisy), and are not part of the design matrix. Survival time was generated from an exponential model with rate parameter λ (and mean $\frac{1}{\lambda}$) according to a Cox-PH model with hazard exp(eta), where eta(.) is the regression function. Censoring indicator were generated from a uniform distribution on [0, 3]. In this synthetic example, all covariates are continuous, i.i.d. from a multivariate uniform distribution on [0, 1].

Usage

```
Synthetic.3
```

Format

Each dataset consists of a `numeric matrix` containing $n = 250$ observations (samples) by rows and $p = 5$ variables by columns ($p < n$ case) including the censoring indicator and (censored) time-to-event variables. It comes as a compressed Rda data file.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

Source

See simulated survival model #3 in Dazard et al., 2015.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.
- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

Synthetic.4 *Synthetic Dataset #4: $p < n$ case*

Description

Modeling survival model #4 as described in Dazard et al. (2015) with censoring. Here, the regression function uses all of the predictors, which are also part of the design matrix. In thiss example, the signal is limited to a box-shaped region R of the predictor space. Survival time was generated from an exponential model with rate parameter λ (and mean $\frac{1}{\lambda}$) according to a Cox-PH model with hazard exp(eta), where eta(.) is the regression function. Censoring indicator were generated from a uniform distribution on [0, 3]. In this synthetic example, all covariates are continuous, i.i.d. from a multivariate uniform distribution on [0, 1].

Usage

```
Synthetic.4
```

Format

Each dataset consists of a `numeric matrix` containing $n = 250$ observations (samples) by rows and $p = 5$ variables by columns ($p < n$ case), including the censoring indicator and (censored) time-to-event variables. It comes as a compressed Rda data file.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

Source

See simulated survival model #4 in Dazard et al., 2015.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.
- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

Synthetic.5 *Synthetic Dataset #5: $p > n$ case*

Description

Modeling survival model #5 as described in Dazard et al. (2015) with censoring. Here, the regression function uses 1/10 of informative predictors in a $p > n$ situation with $p = 1000$ and $n = 100$. The rest represent un-informative noisy variables, which are not part of the design matrix. Survival time was generated from an exponential model with rate parameter λ (and mean $\frac{1}{\lambda}$) according to a Cox-PH model with hazard exp(eta), where eta(.) is the regression function. Censoring indicator were generated from a uniform distribution on [0, 2]. In this synthetic example, all covariates are continuous, i.i.d. from a multivariate standard normal distribution.

Usage

```
Synthetic.5
```

Format

Each dataset consists of a `numeric matrix` containing $n = 100$ observations (samples) by rows and $p = 1000$ variables by columns ($p > n$ case), including the censoring indicator and (censored) time-to-event variables. It comes as a compressed Rda data file.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Michael Choe, M.D." <mjc206@case.edu>
- "Michael LeBlanc, Ph.D." <mleblanc@fhcrc.org>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health NIH - National Cancer Institute (R01-CA160593) to J-E. Dazard and J.S. Rao.

Source

See simulated survival model #2 in Dazard et al., 2015.

References

- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2015). "*Cross-validation and Peeling Strategies for Survival Bump Hunting using Recursive Peeling Methods.*" (Submitted).
- Dazard J-E., Choe M., LeBlanc M. and Rao J.S. (2014). "*Cross-Validation of Survival Bump Hunting by Recursive Peeling Methods.*" In JSM Proceedings, Survival Methods for Risk Estimation/Prediction Section. Boston, MA, USA. American Statistical Association IMS - JSM, p. 3366-3380.
- Dazard J-E. and J. S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

updatecut *Internal Subroutine (never to be called by end-user)*

Description

Internal Subroutine of: peel.box Update the splitpoints in the calling function.

Usage

```
updatecut(x, fract)
```

Arguments

x	numeric matrix of training covariates.
fract	numeric scalar quantile.

Index