Package ‘PRIMsrc’

November 16, 2015

Type Package

Title PRIM Survival Regression Classification

Version 0.7.0

Date 2015-11-16

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

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

Description Performs a unified treatment of Bump Hunting by Patient Rule Induction Method (PRIM) in Survival, Regression and Classification settings (SRC). The current version is a development release that only implements the case of a survival response. New features will be added soon as they are available.

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

Imports graphics, grDevices, stats

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 unified treatment of Bump Hunting by Patient Rule Induction Method (PRIM) in Survival, Regression and Classification settings (SRC). 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 or hyperrectangle 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 low and high-dimensional multivariate datasets, including the situation where the number of covariates exceeds or dominates that of samples ($p > n$ or $p \gg n$ paradigm).

Details

The current version is a development release that only implements the case of a survival response. At this point, survival bump hunting is also restricted to a directed peeling search of the first box covered by the recursive coverage (outer) loop of our Patient Recursive Survival Peeling (PRSP) algorithm. New features will be added soon.

The following describes the end-user functions that are needed to run a complete procedure. The other internal subroutines are not documented in the manual and 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 features are categorized as follows:

1. END-USER FUNCTION FOR PACKAGE NEWS
   `PRIMsrc.news` **Display the PRIMsrc Package News**
   Function to display the log file `NEWS` of updates of the **PRIMsrc** package.

2. END-USER S3-GENERIC FUNCTIONS FOR SUMMARY, DISPLAY, PLOT AND PREDICTION
   `summary` **Summary Function**
   S3-generic summary function to summarize the main parameters used to generate the `PRSP` object.

   `print` **Print Function**
   S3-generic print function to display the cross-validated estimated values of the `PRSP` object.

`plot` **Two-Dimensional Visualization of Data Scatter and Box Vertices**
S3-generic plotting function for two-dimensional visualization of original or predicted data scatter as well as cross-validated box vertices of a `PRSP` object. The scatter plot is for a given peeling step of the peeling sequence and a given plane, both specified by the user.

`predict` **Predict Function**
S3-generic predict function to predict the box membership and box vertices on an independent set.

3. END-USER SURVIVAL BUMP HUNTING FUNCTIONS
`cv.sbh` (*) **Cross-Validated Tuning of a Survival Bump Hunting Model**
First end-user function for tuning a cross-validated Survival Bump Hunting (SBH) model. Returns a cross-validated `CV` object, containing cross-validated estimates of optimal number of pre-selected variables and model parameters. The function performs a single internal cross-validation procedure to simultaneously control model size (#covariates) and model complexity (#peeling steps) before the model is fit. It does a univariate bump hunting variable selection procedure, where model size and model complexity are simultaneously optimized using the cross-validation criterion of choice: CER, LRT, or LHR (see companion paper below for details). This cross-validation procedure is carried out separately from the main function `sbh`. The returned S3-class `CV` object seves as input for the main function `sbh` as well as the profiling plotting functions for the graphical display of profiling curves for model tuning (see `profile_plot` and `surface_plot` functions for more details). The function offers a number of options for the number of cross-validation replicates to be perfomed: B; the type of cross-validation desired: K-fold (replicated)-averaged or-combined, as well as the peeling and optimization critera chosen for model tuning 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 specified CPU cores and efficient parallel execution.
`sbh` (*) **Cross-Validated Fitting of a Survival Bump Hunting Model**
Second end-user function for fitting a cross-validated Survival Bump Hunting (SBH) model. Returns a cross-validated `PRSP` object, as generated by our Patient Recursive Survival Peeling or PRSP algorithm, containing cross-validated estimates of end-points statistics of interest. The function relies on a single internal cross-validation procedure carried out by the main function `cv.sbh` to simultaneously control model size (#covariates) and model complexity (#peeling steps) before the model is fit. The returned S3-class `PRSP` object contains cross-validated estimates of all the decision-rules of pre-selected covariates and all other statistical quantities of interest at each iteration of the peeling sequence (inner loop of the PRSP algorithm). This enables the graphical display of results of peeling trajectories, covariate traces and survival distributions (see plotting functions for more details). 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 specified CPU cores and efficient parallel execution.

4. END-USER PLOTTING FUNCTIONS FOR MODEL VALIDATION AND VISUALIZATION OF RESULTS
`plot_profile` **One-Dimensional Visualization for Model Selection/Validation**
Function for plotting the cross-validated tuning profiles of a `CV` object. It uses the 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).
`plot_surface` **Two-Dimensional Visualization for Model Selection/Validation**
Function for plotting a perspective plot of the cross-validated tuning surface of a `CV` object. It

uses the 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 optimal choice/number of pre-selected variables and of the model tuning parameter, that is, and of number of peeling steps of the peeling sequence (inner loop of our PRSP algorithm).

`plot_boxtraj` **Visualization of Peeling Trajectories/Profiles**
Function for plotting the cross-validated peeling trajectories/profiles of a `PRSP` object. Applies to the user-specified covariates among the pre-selected ones and all other statistical quantities of interest at each iteration of the peeling sequence (inner loop of our PRSP algorithm).

`plot_boxtrace` **Visualization of Covariates Traces**
Function for plotting the cross-validated covariates traces of a `PRSP` object. Plot the cross-validated modal trace curves of covariate importance and covariate usage of the user-specified covariates among the pre-selected ones at each iteration of the peeling sequence (inner loop of our PRSP algorithm).

`plot_boxkm` **Visualization of Survival Distributions**
Function for plotting the cross-validated survival distributions of a `PRSP` object. 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).

5. END-USER DATASETS

`Synthetic.1`, `Synthetic.1b`, `Synthetic.2`, `Synthetic.3`, `Synthetic.4` **Five Datasets From Simulated Regression Survival Models**
Five datasets from simulated regression survival models #1-4 as described in Dazard et al. (2015) representing low- and high-dimensional situations, and where regression parameters represent various types of relationship between survival times and covariates including saturated and noisy situations. In three datasets where non-informative noisy covariates were used, these covariates were not part of the design matrix (models #2-3 and #4). In one dataset, the signal is limited to a box-shaped region R of the predictor space (model #1b). In the last dataset, the signal is limited to 10% of the predictors in a $p > n$ situation (model #4). 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 #1-3) or [0,2] (model #4). In these synthetic datasets, all covariates are continuous, i.i.d. from a multivariate uniform distribution on [0,1] (models #1-3) or from a multivariate standard normal distribution (model #4).

`Real.1` **Clinical Dataset**
Publicly available HIV clinical data 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` **Genomic Dataset**
Publicly available lung cancer genomic data from the Chemores Cohort Study. This was an

integrated study of mRNA, miRNA and clinical variables to characterize the molecular distinctions between squamous cell carcinoma (SCC) and adenocarcinoma (AC) in Non Small Cell Lung Cancer (NSCLC). Tissue samples were analysed from a cohort of 123 patients who underwent complete surgical resection at the Institut Mutualiste Montsouris (Paris, France) between 30 January 2002 and 26 June 2006. In this genomic dataset, only the expression levels of Agilent miRNA probes ($p = 939$) were included from the $n = 123$ samples of the Chemores cohort. It represents a situation where the number of covariates dominates the number of complete observations, 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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**)

cv.sbh *Cross-Validated Tuning of a Survival Bump Hunting Model*

Description

First end-user function for tuning a cross-validated Survival Bump Hunting (SBH) model. Returns a cross-validated CV object, containing cross-validated estimates of optimal number of pre-selected variables and model parameters.

Usage

```
cv.sbh(dataset,
       B = 10, K = 5,
       cvtype = c("combined", "averaged", "none", NULL),
       cvcriterion = c("lrt", "cer", "lhr", NULL),
       conservative = c("most", "medium", "least"),
       arg = "beta=0.05,alpha=0.05,minn=5,peelcriterion=\"lr\"",
       fdr = NULL, thr = 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, and all the covariates thereafter. If a data.frame is provided, it will be coerced to a numeric matrix. Discrete (or nominal) covariates should be made (or rearranged into) ordinal variables.

B
: Positive integer scalar of the number of replications of the cross-validation procedure. Defaults to 10.

K
: Integer giving the number of folds (partitions) into which the observations should be randomly split for the cross-validation procedure. Setting K also specifies the type of cross-validation to be done:

- K = 1 carries no cross-validation out.
- K in {2,...,$n-1$} carries out eqnK-fold cross-validation.
- K = n carries out leave-one-out cross-validation.

cvtype
: Character vector describing the cross-validation technique in {"combined", "averaged", "none", NULL}. If NULL, automatically reset to "none".

cvcriterion
: character vector describing the optimization criterion in {"lrt", "lhr", "cer", NULL}. If NULL, automatically reset to "none".

conservative
: character vector describing the degree of conservativeness in {"most", "medium", "least"} to be used in variable pre-selection.

arg
: Character vector describing the PRSP parameters:

- alpha = fraction to peel off at each step. Defaults to 0.05.
- beta = minimum support size resulting from the peeling sequence. Defaults to 0.05.
- minn = minimum number of observation that we want to be able to detect in a box. Defaults to 5.

- peelcriterion in {"hr" for Log-Hazard Ratio (LHR), "lr" for Log-Rank Test (LRT), "ch" for Cumulative Hazard Summary (CHS)}. 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).

fdr — Numeric scalar of the FDR level at which we want to pre-select the variables. Defaults to NULL.

thr — Numeric scalar of the threshold number of pre-selected the variables. 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

cv.sbh performs a single internal cross-validation procedure to simultaneously control model size (#covariates) and model complexity (#peeling steps) before the model is fit. It does a univariate bump hunting variable selection procedure, where model size and model complexity are simultaneously optimized using the cross-validation criterion of choice: CER, LRT, or LHR (see companion paper below for details). This cross-validation procedure is carried out separately from the main function sbh. The returned S3-class CV object seves as input for the main function sbh as well as the profiling plotting functions for the graphical display of profiling curves for model tuning (see profile_plot and surface_plot functions for more details).

The function offers a number of options for the number of cross-validation replicates to be perfomed: B; the type of cross-validation desired: K-fold (replicated)-averaged or-combined, as well as the peeling and optimization critera chosen for model tuning 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 specified CPU cores and efficient parallel execution.

In the case of large ($p > n$) or very large ($p >> n$) datasets, it is required to use the parallelization option. 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 19 fields:

`x` `numeric` `matrix` of original dataset.

`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.

`cvtype` `character` `vector` of the cross-validation technique used.

`cvcriterion` `character` `vector` of optimization criterion used.

`conservative` `character` `vector` degree of conservativeness used.

`arg` `character` `vector` of the parameters used.

`fdr` `character` `vector` of the `fdr` parameter used.

`thr` `character` `vector` of the `thr` parameter used.

`cvfit` `List` with 7 fields of cross-validated estimates:

- cv.maxsteps: `numeric` scalar of maximal number of peeling steps over the replicates.
- cv.nsteps: `numeric` scalar of optimal number of peeling steps according to the optimization criterion.
- cv.model: `numeric` scalar of coordinate of optimal model.
- cv.nmodel: `numeric` scalar of number of retained models or numbers of pre-selected variables.
- cv.sel`numeric` `vector` of pre-selected covariates in reference to original index.
- cv.sign`numeric` `vector` in {-1,+1} of directions of peeling for all pre-selected covariates.

- cv.profileList of 3 numeric objects: - an array of matrices, one for each cross-validation replicate (in the third dimension), of the cross-validated statistics used in the optimization criterion (set by user) as a function of the number of pre-selected variales and the number of peeling steps. - a matrix of pointwise mean values of the above array over the cross-validation replicates. - a matrix of pointwise standard error values of the above array over the cross-validation replicates.

success logical scalar of the returned flag for pursuing or not a SBH model fitting by the sbh function.

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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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")

#===================================================
# Package news
# Package citation
#===================================================
PRIMsrc.news()
citation("PRIMsrc")

#===================================================
# Demo with a synthetic dataset
# Use help for descriptions
#===================================================
data("Synthetic.1", package="PRIMsrc")
?Synthetic.1

#===================================================
# Simulated dataset #1 (n=250, p=3)
# Non Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
#===================================================
CVCOMB.CV <- cv.sbh(dataset = Synthetic.1,
                    B = 1, K = 5,
                    cvtype = "combined",
                    cvcriterion = "lrt",
                    conservative = "least",
                    arg = "beta=0.05,
                           alpha=0.05,
                           minn=5,
                           peelcriterion=\"lr\"",
                    fdr = NULL, thr = NULL,
                    parallel = FALSE, conf = NULL, seed = 123)

## Not run:
    #===================================================
    # Examples of parallel backend parametrization
    #===================================================
    # Example #1 - 1-Quad (4-core double threaded) PC
    # Running WINDOWS
    # With SOCKET communication
    #===================================================
    if (.Platform$OS.type == "windows") {
        cpus <- detectCores()
        conf <- list("names" = rep("localhost", cpus),
                     "cpus" = cpus,
                     "type" = "SOCK",
                     "homo" = TRUE,
                     "verbose" = TRUE,
                     "outfile" = "")
    }
```

```
#===================================================
# Example #2 - 1 master node + 3 worker nodes cluster
# All nodes equipped with identical setups and multicores
# Running LINUX
# With SOCKET communication
#===================================================
if (.Platform$OS.type == "unix") {
    masterhost <- Sys.getenv("HOSTNAME")
    slavehosts <- c("compute-0-0", "compute-0-1", "compute-0-2")
    nodes <- length(slavehosts) + 1
    cpus <- 8
    conf <- list("names" = c(rep(masterhost, cpus),
                             rep(slavehosts, cpus)),
                 "cpus" = nodes * cpus,
                 "type" = "SOCK",
                 "homo" = TRUE,
                 "verbose" = TRUE,
                 "outfile" = "")
}
#===================================================
# Example #3 - Multinode multicore per node cluster
# Running LINUX
# with MPI communication
# Here, a file named ".nodes" (e.g. in the home directory)
# contains the list of nodes of the cluster
#===================================================
if (.Platform$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
#===================================================
CVCOMBREP.CV <- cv.sbh(dataset = Synthetic.1,
                       B = 10, K = 5,
                       cvtype = "combined",
                       cvcriterion = "lrt",
                       conservative = "least",
                       arg = "beta=0.05,
                              alpha=0.05,
                              minn=5,
                              peelcriterion=\"lr\"",
                       fdr = NULL, thr = NULL,
                       parallel = TRUE, conf = conf, seed = 123)
```

```
## End(Not run)
```

plot.PRSP *2D Visualization of Data Scatter and Box Vertices*

Description

S3-generic plotting function for two-dimensional visualization of original data as well as predicted data scatter with cross-validated box vertices of a `PRSP` object. The scatter plot is for a given peeling step of the peeling sequence and in a given plane of the used covariates of the `PRSP` object, both specified by the user.

Usage

```
  ## S3 method for class 'PRSP'
plot(x,
                      main = NULL,
                      proj = c(1,2),
                      splom = TRUE,
                      boxes = FALSE,
                      steps = x$cvfit$cv.nsteps,
                      pch = 16,
                      cex = 0.5,
                      col = 2:(length(steps)+1),
                      col.box = 2:(length(steps)+1),
                      lty.box = rep(2,length(steps)),
                      lwd.box = rep(1,length(steps)),
                      add.legend = TRUE,
                      device = NULL,
                      file = "Scatter Plot",
                      path=getwd(),
                      horizontal = FALSE,
                      width = 5,
                      height = 5, ...)
```

Arguments

`x` 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 of of the used covariates of the `PRSP` object. Defaults to first two dimensions: {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 to plot the in-box samples and box vertices. Defaults to the last peeling step of `PRSP` object `object`.

`pch` Integer scalar of symbol number for the scatter plot. Defaults to 16.

`cex` Integer scalar of symbol expansion. Defaults to 0.5.

`col` Integer vector specifying the symbol color for each step. Defaults to vector of colors of length the number of steps. The vector is reused cyclically if it is shorter than the number of steps.

`col.box` Integer vector of line color of box vertices for each step. Defaults to vector of colors of length the number of steps. The vector is reused cyclically if it is shorter than the number of steps.

`lty.box` Integer vector of line type of box vertices for each step. Defaults to vector of 2's of length the number of steps. The vector is reused cyclically if it is shorter than the number of steps.

`lwd.box` Integer vector of line width of box vertices for each step. Defaults to vector of 1's of length the number of steps. The vector is reused cyclically if it is shorter than the number of steps.

`add.legend` Logical scalar. Shall the legend of steps numbers be plotted? Defaults to TRUE.

`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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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")

#===================================================
# Simulated dataset #1 (n=250, p=3)
# Non Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
# Without computation of permutation p-values
#===================================================
CVCOMB.CV <- cv.sbh(dataset = Synthetic.1,
                    B = 1, K = 5,
                    cvtype = "combined",
                    cvcriterion = "lrt",
                    conservative = "least",
                    arg = "beta=0.05,
                           alpha=0.05,
                           minn=5,
                           peelcriterion=\"lr\"",
                    fdr = NULL, thr = NULL,
                    parallel = FALSE, conf = NULL, seed = 123)

CVCOMB.SBH <- sbh(cvobj = CVCOMB.CV,
                  cpv = FALSE, decimals = 2,
                  probval = 0.5, timeval = NULL,
                  parallel = FALSE, conf = NULL, seed = 123)

plot(x=CVCOMB.SBH,
     proj=c(1,2), splom=TRUE, boxes=TRUE,
     steps=CVCOMB.SBH$cvfit$cv.nsteps,
     pch=16, cex=0.5, col="red",
     col.box = 2, lty.box = 2, lwd.box = 1,
     add.legend = TRUE, device=NULL)
```

plot_boxkm *Visualization of Survival Distributions*

Description

Function for plotting the cross-validated survival distributions of a PRSP object. 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).

Usage

```
plot_boxkm(object,
           main = NULL,
           xlab = "Time",
           ylab = "Probability",
           precision = 1e-3,
           mark = 3,
           col = 2,
           cex = 1,
           steps = 1:object$cvfit$cv.nsteps,
           nr = 3,
           nc = 4,
           device = NULL,
           file = "Survival Plots",
           path=getwd(),
           horizontal = TRUE,
           width = 11,
           height = 8.5, ...)
```

Arguments

object	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. Defaults to 1e-3.
mark	Integer scalar of mark parameter, which will be used to label the inbox and out-of-box curves. Defaults to 3.
col	Integer scalar specifying the color of the inbox curve. Defaults to 2.
cex	Numeric scalar specifying the size of the marks. Defaults to 1.
steps	Integer vector. Vector of peeling steps at which to plot the survival curves. Defaults to all the peeling steps of PRSP object object.
nr	Integer scalar of the number of rows in the plot. Defaults to 3.
nc	Integer scalar of the number of columns in the plot. 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 the 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.
`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

Some of the plotting parameters are further defined in the function `plot.survfit` (R package **survival**). 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 $1/A$, where A is the number of permutations.

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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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")

#===================================================
# Simulated dataset #1 (n=250, p=3)
# Non Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
# Without computation of permutation p-values
#===================================================
CVCOMB.CV <- cv.sbh(dataset = Synthetic.1,
                    B = 1, K = 5,
                    cvtype = "combined",
                    cvcriterion = "lrt",
                    conservative = "least",
                    arg = "beta=0.05,
                           alpha=0.05,
                           minn=5,
                           peelcriterion=\"lr\"",
                    fdr = NULL, thr = NULL,
                    parallel = FALSE, conf = NULL, seed = 123)

CVCOMB.SBH <- sbh(cvobj = CVCOMB.CV,
                  cpv = FALSE, decimals = 2,
                  probval = 0.5, timeval = NULL,
                  parallel = FALSE, conf = NULL, seed = 123)

plot_boxkm(object = CVCOMB.SBH,
           main = paste("Cross-validated probability curves for model #1", sep=""),
           xlab = "Time", ylab="Probability",
           steps=1:CVCOMB.SBH$cvfit$cv.nsteps,
           device = NULL)
```

`plot_boxtrace` *Visualization of Covariates Traces*

Description

Function for plotting the cross-validated covariates traces of a PRSP object. Plot the cross-validated modal trace curves of covariate importance and covariate usage of the pre-selected covariates specified by user at each iteration of the peeling sequence (inner loop of our PRSP algorithm).

Usage

```
plot_boxtrace(object,
              main = NULL,
              xlab = "Box Mass",
              ylab = "Covariate Range (centered)",
              toplot = object$cvfit$cv.used,
              center = TRUE,
              scale = FALSE,
              col.cov,
              lty.cov,
              lwd.cov,
              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

`object` Object of class PRSP as generated by the main function sbh.

`main` `Character vector`. Main Title. Defaults to.

`xlab` `Character vector`. X axis label. Defaults to "Box Mass". NULL

`ylab` `Character vector`. Y axis label. Defaults to "Covariate Range (centered)".

`toplot` `Numeric vector`. Which of the pre-selected covariates to plot (in reference to the original index of covariates). Defaults to covariates used for peeling.

`center` `Logical` scalar. Shall the data be centered?. Defaults to TRUE.

`scale` `Logical` scalar. Shall the data be scaled? Defaults to FALSE.

`col.cov` `Integer vector`. Line color for the covariate importance curve of each selected covariate. Defaults to vector of colors of length the number of selected covariates. The vector is reused cyclically if it is shorter than the number of selected covariates.

`lty.cov` `Integer vector`. Line type for the covariate importance curve of each selected covariate. Defaults to vector of 1's of length the number of selected covariates. The vector is reused cyclically if it is shorter than the number of selected covariates.

`lwd.cov` `Integer vector`. Line width for the covariate importance curve of each selected covariate. Defaults to vector of 1's of length the number of selected covariates. The vector is reused cyclically if it is shorter than the number of selected covariates.

`col` `Integer` scalar. Line color for the covariate trace curve. Defaults to 1.

`lty` `Integer` scalar. Line type for the covariate trace curve. Defaults to 1.

`lwd` `Integer` scalar. Line width for the covariate trace curve. Defaults to 1.

`cex` `Integer` scalar. Symbol expansion used for titles, legends, and axis labels. 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. If centered, an horizontal black dotted line about 0 is added to the plot.

Due to the variability induced by cross-validation and replication, it is possible that more than one covariate 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 covariate importance curves for each covariate. The bottom plot shows the overlay of covariate usage curves for each covariate. It is a dicretized view of covariate 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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")

#===================================================
# Simulated dataset #1 (n=250, p=3)
# Non Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
# Without computation of permutation p-values
#===================================================
CVCOMB.CV <- cv.sbh(dataset = Synthetic.1,
                    B = 1, K = 5,
                    cvtype = "combined",
                    cvcriterion = "lrt",
                    conservative = "least",
                    arg = "beta=0.05,
                           alpha=0.05,
                           minn=5,
                           peelcriterion=\"lr\"",
                    fdr = NULL, thr = NULL,
                    parallel = FALSE, conf = NULL, seed = 123)
```

```
CVCOMB.SBH <- sbh(cvobj = CVCOMB.CV,
                  cpv = FALSE, decimals = 2,
                  probval = 0.5, timeval = NULL,
                  parallel = FALSE, conf = NULL, seed = 123)

plot_boxtrace(object = CVCOMB.SBH,
              main = paste("Cross-validated trace plots for model #1", sep=""),
              xlab="Box Mass", ylab="Covariate Range (centered)",
              toplot = CVCOMB.SBH$cvfit$cv.used,
              center = TRUE, scale = FALSE,
              device = NULL)
```

plot_boxtraj *Visualization of Peeling Trajectories/Profiles*

Description

Function for plotting the cross-validated peeling trajectories/profiles of a PRSP object. Applies to the pre-selected covariates specified by user and all other statistical quantities of interest at each iteration of the peeling sequence (inner loop of our PRSP algorithm).

Usage

```
plot_boxtraj(object,
             main = NULL,
             toplot = object$cvfit$cv.used,
             col.cov,
             lty.cov,
             lwd.cov,
             col = 1,
             lty = 1,
             lwd = 1,
             cex = 1,
             add.legend = FALSE,
             text.legend = NULL,
             nr = NULL,
             nc = NULL,
             device = NULL,
             file = "Trajectory Plots",
             path=getwd(),
             horizontal = FALSE,
             width = 8.5,
             height = 11, ...)
```

Arguments

`object`	Object of class PRSP as generated by the main function sbh.
`main`	`Character vector`. Main Title. Defaults to NULL.
`toplot`	`Numeric vector`. Which of the pre-selected covariates to plot (in reference to the original index of covariates). Defaults to covariates used for peeling.

`col.cov` `Integer vector`. Line color for the covariate trajectory curve of each selected covariate. Defaults to vector of colors of length the number of selected covariates. The vector is reused cyclically if it is shorter than the number of selected covariates.

`lty.cov` `Integer vector`. Line type for the covariate trajectory curve of each selected covariate. Defaults to vector of 1's of length the number of selected covariates. The vector is reused cyclically if it is shorter than the number of selected covariates.

`lwd.cov` `Integer vector`. Line width for the covariate trajectory curve of each selected covariate. Defaults to vector of 1's of length the number of selected covariates. The vector is reused cyclically if it is shorter than the number of selected covariates.

`col` `Integer` scalar. Line color for the trajectory curve of each statistical quantity of interest. Defaults to 1.

`lty` `Integer` scalar. Line type for the trajectory curve of each statistical quantity of interest. Defaults to 1.

`lwd` `Integer` scalar. Line width for the trajectory curve of each statistical quantity of interest. Defaults to 1.

`cex` `Integer` scalar. Symbol expansion used for titles, legends, and axis labels. 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 "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 11.

`...` 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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")

#===================================================
# Simulated dataset #1 (n=250, p=3)
# Non Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
# Without computation of permutation p-values
#===================================================
CVCOMB.CV <- cv.sbh(dataset = Synthetic.1,
                    B = 1, K = 5,
                    cvtype = "combined",
                    cvcriterion = "lrt",
                    conservative = "least",
                    arg = "beta=0.05,
                           alpha=0.05,
                           minn=5,
                               peelcriterion=\"lr\"",
                         fdr = NULL, thr = NULL,
                         parallel = FALSE, conf = NULL, seed = 123)

CVCOMB.SBH <- sbh(cvobj = CVCOMB.CV,
                  cpv = FALSE, decimals = 2,
                  probval = 0.5, timeval = NULL,
                  parallel = FALSE, conf = NULL, seed = 123)

plot_boxtraj(object = CVCOMB.SBH,
             main = paste("Cross-validated peeling trajectories for model #1", sep=""),
             toplot = CVCOMB.SBH$cvfit$cv.used,
             device = NULL)
```

plot_profile *One-Dimensional Visualization for Model Selection/Validation*

Description

Function for plotting the cross-validated tuning profiles of a CV object. It uses the 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(cvobj,
             main = NULL,
             xlab = "Peeling Steps",
             ylab = "Mean Profiles",
             add.sd = 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 = 11, ...)
```

Arguments

cvobj	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.profiles` `Logical` scalar. Shall the individual profiles (for all replicates) be plotted? Defaults to `TRUE`.

`pch` `Integer` scalar of symbol number for all the profiles. Defaults to 20.

`col` `Integer` scalar of line color of the mean profile. Defaults to 1.

`lty` `Integer` scalar of line type of the mean profile. Defaults to 1.

`lwd` `Integer` scalar of line width of the mean profile. Defaults to 2.

`cex` `Integer` scalar of 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 11.

`...` Generic arguments passed to other plotting functions.

Details

Model tuning is done by applying the optimization criterion defined by the user's choice of specific statistic. The goal is to find the optimal value of the number of steps by maximization of LHR or LRT, or minimization of CER.

Currently, this is done internally for visualization purposes, but it will ultimately offer the option to be done interactively 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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")

#===================================================
# Simulated dataset #1 (n=250, p=3)
# Non Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
#===================================================
CVCOMB.CV <- cv.sbh(dataset = Synthetic.1,
                    B = 1, K = 5,
                    cvtype = "combined",
                    cvcriterion = "lrt",
                    conservative = "least",
                    arg = "beta=0.05,
                           alpha=0.05,
                           minn=5,
                           peelcriterion=\"lr\"",
                    fdr = NULL, thr = NULL,
                    parallel = FALSE, conf = NULL, seed = 123)

plot_profile(cvobj = CVCOMB.CV,
             main = paste("Cross-validated tuning profile for model #1", sep=""),
             xlab = "Peeling Steps", ylab = "Mean Profiles",
             pch = 20, col = "black", lty = 1, lwd = 2, cex = 2,
             add.sd = TRUE, add.profiles = TRUE,
             device = NULL)
```

plot_surface *Two-Dimensional Visualization for Model Selection/Validation*

Description

Function for plotting a perspective plot of the cross-validated tuning surface of a CV object. It uses the 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 optimal choice/number of pre-selected variables and of the model tuning parameter, that is, and of number of peeling steps of the peeling sequence (inner loop of our PRSP algorithm).

Usage

```
plot_surface(cvobj,
             main = NULL,
             xlab = "Model Size (# variables)",
             ylab = "Peeling Steps",
             theta = 5,
             phi = 10,
             expand = 0.2,
             col = "lightblue",
             add.line = FALSE,
             col.line = "yellow",
             lty.line = 1,
             lwd.line = 1,
             pch.line = 20,
             cex.line = 1,
             device = NULL,
             file = "Surface Plot",
             path=getwd(),
             horizontal = FALSE,
             width = 8.5,
             height = 5.0, ...)
```

Arguments

cvobj	Object of class CV as generated by the main function cv.sbh.
main	Character vector. Main Title. Defaults to NULL.
xlab	Character vector. X axis label. Defaults to "Model Size (# variables)".
ylab	Character vector. Y axis label. Defaults to "Peeling Steps".
theta	Integer scalar of angle defining the viewing direction of the perspective plot. Defaults to 5. theta gives the azimuthal direction. See base function persp from R package **graphics**.
phi	Integer scalar of angle defining the viewing direction of the perspective plot. Defaults to 10. phi gives the colatitude. See base function persp from R package **graphics**.
expand	Integer scalar of a expansion factor applied to the z coordinates. Defaults to 0.2. Often used with 0 < expand < 1 to shrink the plotting box in the z direction. See base function persp from R package **graphics**.
col	Integer scalar of color of the surface facets of the mean profiles. Defaults to "lightblue". Transparent colours are ignored. See base function persp from R package **graphics**.
add.line	Logical scalar. Shall the optimal path line be plotted on the surface? Defaults to FALSE.

`col.line`	`Integer` scalar of line color of the optimal path line. Defaults to "yellow".
`lty.line`	`Integer` scalar of line type of the optimal path line. Defaults to 1.
`lwd.line`	`Integer` scalar of line width of the optimal path line. Defaults to 1.
`pch.line`	`Integer` scalar of symbol number of the points on the optimal path line. Defaults to 20.
`cex.line`	`Integer` scalar of symbol expansion of the points on the optimal path line. 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 "Surface 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 5.0.
`...`	Generic arguments passed to other plotting functions.

Details

Model tuning is done by applying the optimization criterion defined by the user's choice of specific statistic. The goal is to find the optimal values of the number of pre-selected variables and 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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")

#===================================================
# Simulated dataset #1 (n=250, p=3)
# Non Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
#===================================================
CVCOMB.CV <- cv.sbh(dataset = Synthetic.1,
                    B = 1, K = 5,
                    cvtype = "combined",
                    cvcriterion = "lrt",
                    conservative = "least",
                    arg = "beta=0.05,
                           alpha=0.05,
                           minn=5,
                           peelcriterion=\"lr\"",
                    fdr = NULL, thr = NULL,
                    parallel = FALSE, conf = NULL, seed = 123)

plot_surface(cvobj = CVCOMB.CV,
             main = paste("Cross-validated tuning surface for model #1", sep=""),
             xlab = "Model Size (# variables)", ylab = "Peeling Steps",
             theta = 5, phi = 10, expand = 0.2, col = "lightblue",
             add.line = TRUE,
             col.line = "yellow", lty.line = 1, lwd.line = 1,
             pch.line = 20, cex.line = 1,
             device = NULL)
```

`predict.PRSP` *Predict Function*

Description

S3-generic predict function to predict the box membership and box vertices on an independent set.

Usage

```
  ## S3 method for class 'PRSP'
predict(object,
                         newdata,
                         steps,
                         na.action = na.omit, ...)
```

Arguments

`object` Object of class `PRSP` as generated by the main function `sbh`.

`newdata` Either a numeric matrix or numeric vector containing the new input data of same dimensionality as the final `PRSP` object of used covariates. A vector will be transformed to a (\#sample x 1) matrix.

`steps` `Integer vector`. Vector of peeling steps at which to predict the box memberships and box vertices. Defaults to the last peeling step only.

`na.action` A function to specify the action to be taken if NAs are found. The default action is na.omit, which leads to rejection of incomplete cases.

`...` Further generic arguments passed to the predict function.

Value

`List` containing the following 2 fields:

`boxind` `Logical matrix` of predicted box membership indicator (columns) by peeling steps (rows). TRUE = in-box, FALSE = out-of-box.

`vertices` `List` of size the number of chosen peeling steps where each entry is a `numeric matrix` of predicted box vertices: lower and upper bounds (rows) by covariate (columns).

Note

End-user predict 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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")

#===================================================
# Simulated dataset #1 (n=250, p=3)
# Non Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
# Without computation of permutation p-values
#===================================================
CVCOMB.CV <- cv.sbh(dataset = Synthetic.1,
                    B = 1, K = 5,
                    cvtype = "combined",
                    cvcriterion = "lrt",
                    conservative = "least",
                    arg = "beta=0.05,
                           alpha=0.05,
                           minn=5,
                           peelcriterion=\"lr\"",
                    fdr = NULL, thr = NULL,
                    parallel = FALSE, conf = NULL, seed = 123)

CVCOMB.SBH <- sbh(cvobj = CVCOMB.CV,
                  cpv = FALSE, decimals = 2,
                  probval = 0.5, timeval = NULL,
                  parallel = FALSE, conf = NULL, seed = 123)

n <- 100
p <- length(CVCOMB.SBH$cvfit$cv.used)
x <- matrix(data=runif(n=n*p, min=0, max=1),
            nrow=n, ncol=p, byrow=FALSE,
            dimnames=list(1:n, paste("X", 1:p, sep="")))
CVCOMB.PRED <- predict(object=CVCOMB.SBH,
                       newdata=x,
                       steps=CVCOMB.SBH$cvfit$cv.nsteps)
```

PRIMsrc.news *Display the* **PRIMsrc** *Package News*

Description

Function to display the log file NEWS 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- Dazard J-E. and J.S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

print.PRSP *Print Function*

Description

S3-generic print function to display the cross-validated estimated values of the PRSP object.

Usage

```
  ## S3 method for class 'PRSP'
print(x, ...)
```

Arguments

x	Object of class PRSP as generated by the main function sbh.
...	Further generic arguments passed to the print function.

Value

Display of the cross-validated fitted values of its argument.

Note

End-user print 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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")

#===================================================
# Simulated dataset #1 (n=250, p=3)
# Non Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
# Without computation of permutation p-values
#===================================================
CVCOMB.CV <- cv.sbh(dataset = Synthetic.1,
                    B = 1, K = 5,
                    cvtype = "combined",
                    cvcriterion = "lrt",
                    conservative = "least",
                    arg = "beta=0.05,
                           alpha=0.05,
                           minn=5,
                           peelcriterion=\"lr\"",
                    fdr = NULL, thr = NULL,
                    parallel = FALSE, conf = NULL, seed = 123)

CVCOMB.SBH <- sbh(cvobj = CVCOMB.CV,
                  cpv = FALSE, decimals = 2,
                  probval = 0.5, timeval = NULL,
                  parallel = FALSE, conf = NULL, seed = 123)

print(CVCOMB.SBH)
```

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

Description

Publicly available HIV clinical data 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 clinical 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 below 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 $p = 4$ clinical covariates by columns, not 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 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- Dazard J-E. and J.S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

See Also

`statepiaps.jhsph.edu/wihs/`

`Real.2` *Real Dataset #2: Genomic Dataset ($p >> n$ case)*

Description

Publicly available lung cancer genomic data from the Chemores Cohort Study. This data is part of an integrated study of mRNA, miRNA and clinical variables to characterize the molecular distinctions between squamous cell carcinoma (SCC) and adenocarcinoma (AC) in Non Small Cell Lung Cancer (NSCLC) aside large cell lung carcinoma (LCC). Tissue samples were analysed from a cohort of 123 patients who underwent complete surgical resection at the Institut Mutualiste Montsouris (Paris, France) between 30 January 2002 and 26 June 2006. In this genomic dataset, the expression levels of Agilent miRNA probes ($p = 939$) were included from the $n = 123$ samples of the Chemores cohort. The data contains normalized expression levels. See below the paper by Lazar et al. (2013) and Array Express data repository for complete description of the samples, tissue preparation, Agilent array technology, data normalization, etc. This dataset represents a situation where the number of covariates dominates the number of complete observations, or $p >> n$ case.

Usage

```
Real.2
```

Format

Dataset consists of a `numeric` `data.frame` containing $n = 123$ complete observations (samples) by rows and $p = 939$ genomic covariates by columns, not 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 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- Dazard J-E. and J.S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

See Also

Array Express data repository at the European Bioinformatics Institute. Accession number: #E-MTAB-1134 (MIR). www.ebi.ac.uk/arrayexpress/

CHEMORES Consortium and website. www.chemores.ki.se/index.html

sbh *Cross-Validated Fitting of a Survival Bump Hunting Model*

Description

Second end-user function for fitting a cross-validated Survival Bump Hunting (SBH) model. Returns a cross-validated PRSP object, as generated by our Patient Recursive Survival Peeling or PRSP algorithm, containing cross-validated estimates of end-points statistics of interest.

Usage

```
sbh(cvobj = NULL,
    A = 1000, cpv = FALSE, decimals = 2,
    probval = NULL, timeval = NULL,
    parallel = FALSE, conf = NULL, seed = NULL)
```

Arguments

cvobj	Object of class CV as generated by the main function cv.sbh.
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 permutation p-values. Defaults to FALSE.
decimals	integer scalar. Number of user-specified significant decimals to output results. Defaults to 2.
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

At this point, the main function sbh performs the search of the *first* box of the recursive coverage (outer) loop of our Patient Recursive Survival Peeling (PRSP) algorithm. It relies on a single internal cross-validation procedure carried out by the main function cv.sbh to simultaneously control model size (#covariates) and model complexity (#peeling steps) before the model is fit.

The returned S3-class PRSP object contains cross-validated estimates of all the decision-rules of pre-selected covariates and all other statistical quantities of interest at each iteration of the peeling sequence (inner loop of the PRSP algorithm). This enables the graphical display of results of peeling trajectories, covariate traces and survival distributions (see plotting functions for more details).

In case replicated cross-validations are performed (see object `CV`), a "summary" of the outputs is done over the number of replicates, which requires some explanation:

- Even thought the PRSP algorithm uses only one covariate at a time at each peeling step, the reported matrix of "Replicated CV" box decision rules may show several covariates being used in a given step, simply because these decision rules are averaged over the B replicates (see equation #21 in Dazard et al. 2015). This is also reflected in the reported "Replicated CV" importance and usage plots of covariate traces.
- Likewise, the output matrix of "Replicated CV" box membership indicator does not necessarily match exactly the output vector of "Replicated CV" box support (and corresponding box sample size) for all peeling steps. The reason is that the reported "Replicated CV" box membership indicators are computed (at each peeling step) as the point-wise majority vote over the B replicates (see equation #22 in Dazard et al. 2015), whereas the "Replicated CV" box support vector (and corresponding box sample size) is averaged (at each peeling step) over the B replicates.

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 specified CPU cores and efficient parallel execution.

If the computation of permutation p-values is desired, then running with the parallelization option is strongly advised as it may take a while. In the case of large ($p > n$) or very large ($p >> n$) datasets, it is also required to use the parallelization option.

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 `A` is internally reset to `conf$cpus*ceiling(A/conf$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 19 fields:

`x` `numeric` `matrix` of original dataset.

`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 permutation p-values.

`cpv` `logical` scalar of returned flag of optional computation of permutation p-values.

`decimals` `integer` of the number of user-specified significant decimals.

`cvtype` `character` `vector` of the cross-validation technique used.

`cvcriterion` `character` `vector` of optimization criterion used.

`conservative` `character` `vector` degree of conservativeness used.

`arg` `character` `vector` of the parameters used.

`probval` `Numeric` scalar of survival probability used.

`timeval` `Numeric` scalar of survival time used.

`cvfit` `List` with 9 fields of cross-validated estimates:

- cv.maxsteps: `numeric` scalar of maximal number of peeling steps over the replicates.
- cv.nsteps: `numeric` scalar of optimal number of peeling steps according to the optimization criterion.
- cv.trace: `numeric` `vector` of the modal trace values of covariate usage for all peeling steps.
- cv.sel`numeric` `vector` of pre-selected covariates in reference to original index.
- cv.sign`numeric` `vector` in {-1,+1} of directions of peeling for all pre-selected covariates.
- cv.boxind: `logical` `matrix` in `TRUE`, `FALSE` of individual observation box membership indicator (columns) for all peeling steps (rows).
- cv.rules: `data.frame` of decision rules on the covariates (columns) for all peeling steps (rows).
- cv.stats: `numeric` `matrix` of box endpoint quantities of interest (columns) for all peeling steps (rows).
- cv.pval: `numeric` `vector` of log-rank permutation p-values of sepraration of survival distributions.

`plot` `logical` scalar of the returned flag for plotting or not the results of the fitted SBH model.

`config` `List` with 7 fields of parameters used for configuring the parallelization including `parallel` and `conf`.

`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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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")

#===================================================
# Package news
# Package citation
```

```
#===================================================
PRIMsrc.news()
citation("PRIMsrc")

#===================================================
# Demo with a synthetic dataset
# Use help for descriptions
#===================================================
data("Synthetic.1", package="PRIMsrc")
?Synthetic.1

#===================================================
# Simulated dataset #1 (n=250, p=3)
# Non Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
# Without computation of permutation p-values
#===================================================
CVCOMB.CV <- cv.sbh(dataset = Synthetic.1,
                    B = 1, K = 5,
                    cvtype = "combined",
                    cvcriterion = "lrt",
                    conservative = "least",
                    arg = "beta=0.05,
                           alpha=0.05,
                           minn=5,
                           peelcriterion=\"lr\"",
                    fdr = NULL, thr = NULL,
                    parallel = FALSE, conf = NULL, seed = 123)

CVCOMB.SBH <- sbh(cvobj = CVCOMB.CV,
                  cpv = FALSE, decimals = 2,
                  probval = 0.5, timeval = NULL,
                  parallel = FALSE, conf = NULL, seed = 123)

## Not run:
    #===================================================
    # Examples of parallel backend parametrization
    #===================================================
    # Example #1 - 1-Quad (4-core double threaded) PC
    # Running WINDOWS
    # With SOCKET communication
    #===================================================
    if (.Platform$OS.type == "windows") {
        cpus <- detectCores()
        conf <- list("names" = rep("localhost", cpus),
                     "cpus" = cpus,
                     "type" = "SOCK",
                     "homo" = TRUE,
                     "verbose" = TRUE,
                     "outfile" = "")
    }
    #===================================================
    # Example #2 - 1 master node + 3 worker nodes cluster
    # All nodes equipped with identical setups and multicores
    # Running LINUX
    # With SOCKET communication
    #====================================================
    if (.Platform$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" = "")
    }
    #====================================================
    # Example #3 - Multinode multicore per node cluster
    # Running LINUX
    # with MPI communication
    # Here, a file named ".nodes" (e.g. in the home directory)
    # contains 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 permutation p-values
    #====================================================
    CVCOMBREP.CV <- cv.sbh(dataset = Synthetic.1,
                           B = 10, K = 5,
                           cvtype = "combined",
                           cvcriterion = "lrt",
                           conservative = "least",
                           arg = "beta=0.05,
                                  alpha=0.05,
                                  minn=5,
                                  peelcriterion=\"lr\"",
                           fdr = NULL, thr = NULL,
                           parallel = TRUE, conf = conf, seed = 123)

    CVCOMBREP.SBH <- sbh(cvobj = CVCOMBREP.CV,
                         A = 1000, cpv = TRUE, decimals = 2,
                         probval = 0.5, timeval = NULL,
                         parallel = TRUE, conf = conf, seed = 123)

## End(Not run)
```

summary.PRSP *Summary Function*

Description

S3-generic summary function to summarize the main parameters used to generate the PRSP object.

Usage

```
  ## S3 method for class 'PRSP'
summary(object, ...)
```

Arguments

object	Object of class PRSP as generated by the main function sbh.
...	Further generic arguments passed to the summary function.

Value

Summarizes the main parameters used to generate its argument.

Note

End-user summary 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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")

#===================================================
# Simulated dataset #1 (n=250, p=3)
# Non Replicated Combined Cross-Validation (RCCV)
# Peeling criterion = LRT
# Optimization criterion = LRT
# Without parallelization
# Without computation of permutation p-values
#===================================================
CVCOMB.CV <- cv.sbh(dataset = Synthetic.1,
                    B = 1, K = 5,
                    cvtype = "combined",
                    cvcriterion = "lrt",
                    conservative = "least",
                    arg = "beta=0.05,
                           alpha=0.05,
                           minn=5,
                           peelcriterion=\"lr\"",
                    fdr = NULL, thr = NULL,
                    parallel = FALSE, conf = NULL, seed = 123)

CVCOMB.SBH <- sbh(cvobj = CVCOMB.CV,
                  cpv = FALSE, decimals = 2,
                  probval = 0.5, timeval = NULL,
                  parallel = FALSE, conf = NULL, seed = 123)

summary(CVCOMB.SBH)
```

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

Description

Dataset from simulated regression survival model #1 as described in Dazard et al. (2015). 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 $p = 3$ variables by columns, not 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- Dazard J-E. and J.S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

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

Description

Dataset from simulated regression survival model #1b as described in Dazard et al. (2015). Here, the regression function uses all of the predictors, which are also part of the design matrix. In this 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.1b
```

Format

Each dataset consists of a `numeric matrix` containing $n = 250$ observations (samples) by rows and $p = 3$ variables by columns, not 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 #1b 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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

Dataset from simulated regression survival model #2 as described in Dazard et al. (2015). Here, the regression function uses some informative predictors. The rest represent un-informative noisy covariates, 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 $p = 3$ variables by columns, not 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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

Dataset from simulated regression survival model #3 as described in Dazard et al. (2015). Here, the regression function does not include any of the predictors. This means that none of the covariates 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 $p = 3$ variables by columns, not 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- 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

Dataset from simulated regression survival model #4 as described in Dazard et al. (2015). Here, the regression function uses 1/10 of informative predictors in a $p > n$ situation with $p = 1000$ and $n = 100$. The rest represents non-informative noisy covariates, 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.4
```

Format

Each dataset consists of a `numeric` `matrix` containing $n = 100$ observations (samples) by rows and $p = 1000$ variables by columns, not 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.*" Statistical Analysis and Data Mining (in press).
- 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., Choe M., LeBlanc M. and Rao J.S. (2015). "*R package PRIMsrc: Bump Hunting by Patient Rule Induction Method for Survival, Regression and Classification.*" In JSM Proceedings, Statistical Programmers and Analysts Section. Seattle, WA, USA. American Statistical Association IMS - JSM, (in press).
- Dazard J-E. and J.S. Rao (2010). "*Local Sparse Bump Hunting.*" J. Comp Graph. Statistics, 19(4):900-92.

Index