Package 'adapt4pv'

June 24, 2020

Type Package

Title Adaptive Approaches for Signal Detection in Pharmacovigilance

Version 0.1.0

Depends R (>= 3.6.0), Matrix (>= 1.0-6), glmnet (>= 3.0-2)

Imports speedglm, xgboost, doParallel, foreach

Description
A collection of several pharmacovigilance signal detection methods based on adaptive lasso. Additional lasso-based and propensity score-based signal detection approaches are also supplied.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Author Emeline Courtois [cre],
Ismaïl Ahmed [aut],
Hervé Perdry [ctb]

Maintainer Emeline Courtois <emeline.courtois@inserm.fr>

NeedsCompilation no

Repository CRAN

Date/Publication 2020-06-24 09:40:07 UTC

R topics documented:

adapt4pv-package 2
adapt_bic 2
adapt_cisl 4
adapt_cv 6
adapt_ridge 8
adapt_univ 10
cisl 12
data_PV 14
est_ps_bic 14

est_ps_hdps 16
est_ps_xgb 17
lasso_bic 19
lasso_cv 20
lasso_perm 21
ps_adjust 23
ps_adjust_one 25
ps_pond 26
ps_pond_one 28
summary_stat 30

Index **32**

`adapt4pv-package` *Adaptive approaches for signal detection in PharmacoVigilance*

Description

This package fits adaptive lasso approaches in high dimension for signal detection in pharmacovigilance. In addition to classical implementations found in the litterature, we implemented two approaches particularly appropriated to variable selections framework, which is the one that stands in pharmacovigilance. We also supply in this package signal detection approaches based on lasso regression and propensity score in high dimension.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

`adapt_bic` *fit an adaptive lasso with adaptive weights derived from lasso-bic*

Description

Fit a first lasso regression and use Bayesian Information Criterion to determine ' adaptive weights (see `lasso_bic` function for more details), then run an adaptive lasso with this penalty weighting. BIC is used for the adaptive lasso for variable selection. Can deal with very large sparse data matrices. Intended for binary reponse only (option `family = "binomial"` is forced). Depends on the `glmnet` and `relax.glmnet` function from the package `glmnet`.

Usage

```
adapt_bic(x, y, gamma = 1, maxp = 50, path = TRUE, betaPos = TRUE, ...)
```

Arguments

x	Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class "sparseMatrix" as in package Matrix).
y	Binary response variable, numeric.
gamma	Tunning parameter to defined the penalty weights. See details below. Default is set to 1.
maxp	A limit on how many relaxed coefficients are allowed. Default is 50, in glmnet option default is 'n-3', where 'n' is the sample size.
path	Since glmnet does not do stepsize optimization, the Newton algorithm can get stuck and not converge, especially with relaxed fits. With path=TRUE, each relaxed fit on a particular set of variables is computed pathwise using the original sequence of lambda values (with a zero attached to the end). Default is path=TRUE.
betaPos	Should the covariates selected by the procedure be positively associated with the outcome ? Default is TRUE.
...	Other arguments that can be passed to glmnet from package glmnet other than penalty.factor, family, maxp and path.

Details

The adaptive weight for a given covariate i is defined by

$$w_i = 1/|\beta^B IC_i|^\gamma$$

where $\beta^B IC_i$ is the NON PENALIZED regression coefficient associated to covariate i obtained with lasso-bic.

Value

An object with S3 class "adaptive".

aws	Numeric vector of penalty weights derived from lasso-bic. Length equal to nvars.
criterion	Character, indicates which criterion is used with the adaptive lasso for variable selection. For adapt_bic function, criterion is "bic".
beta	Numeric vector of regression coefficients in the adaptive lasso. If criterion = "cv" the regression coefficients are PENALIZED, if criterion = "bic" the regression coefficients are UNPENALIZED. Length equal to nvars. Could be NA if adaptive weights are all equal to infinity.
selected_variables	Character vector, names of variable(s) selected with this adaptive approach. If betaPos = TRUE, this set is the covariates with a positive regression coefficient in beta. Else this set is the covariates with a non null regression coefficient in beta. Covariates are ordering according to the p-values (two-sided if betaPos = FALSE , one-sided if betaPos = TRUE) in the classical multiple logistic regression model that minimzes the BIC in the adaptive lasso.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
ab <- adapt_bic(x = drugs, y = ae, maxp = 50)
```

---

adapt_cisl *fit an adaptive lasso with adaptive weights derived from CISL*

Description

Compute the CISL procedure (see `cisl` for more details) to determine adaptive penalty weights, then run an adaptive lasso with this penalty weighting. BIC is used for the adaptive lasso for variable selection. Can deal with very large sparse data matrices. Intended for binary reponse only (option `family = "binomial"` is forced). Depends on the `glmnet` function from the package `glmnet`.

Usage

```
adapt_cisl(
  x,
  y,
  cisl_nB = 100,
  cisl_dfmax = 50,
  cisl_nlambda = 250,
  cisl_ncore = 1,
  maxp = 50,
  path = TRUE,
  betaPos = TRUE,
  ...
)
```

Arguments

x	Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class `"sparseMatrix"` as in package `Matrix`).
y	Binary response variable, numeric.
cisl_nB	nB option in `cisl` function

cisl_dfmax	dfmax option in cisl function
cisl_nlambda	nlambda option in cisl function
cisl_ncore	ncore option in cisl function
maxp	A limit on how many relaxed coefficients are allowed. Default is 50, in glmnet option default is 'n-3', where 'n' is the sample size.
path	Since glmnet does not do stepsize optimization, the Newton algorithm can get stuck and not converge, especially with relaxed fits. With path=TRUE, each relaxed fit on a particular set of variables is computed pathwise using the original sequence of lambda values (with a zero attached to the end). Default is path=TRUE.
betaPos	Should the covariates selected by the procedure be positively associated with the outcome ? Default is TRUE.
...	Other arguments that can be passed to glmnet from package glmnet other than penalty.factor, family, maxp and path.

Details

The CISL procedureis first implemented with its default value except for dfmax and nlambda through parameters cisl_dfmax and cisl_nlambda. In addition, the betaPos parameter is set to FALSE in cisl. For each covariate i, 100 values of the CISL quantity τ_i are estimated. The adaptive weight for a given covariate i is defined by

$$w_i = 1 - 1/100 \sum_b = 1, .., 100 indicator[\tau_i^b > 0]$$

If τ_i is the null vector, the associated adaptve weights in infinty. If τ_i is always positive, rather than "forcing" the variable into the model, we set the corresponding adaptive weight to $1/100$.

Value

An object with S3 class "adaptive".

aws	Numeric vector of penalty weights derived from CISL. Length equal to nvars.
criterion	Character, indicates which criterion is used with the adaptive lasso for variable selection. For adapt_cisl function, criterion is "bic".
beta	Numeric vector of regression coefficients in the adaptive lasso. If criterion = "cv" the regression coefficients are PENALIZED, if criterion = "bic" the regression coefficients are UNPENALIZED. Length equal to nvars. Could be NA if adaptive weights are all equal to infinity.
selected_variables	Character vector, names of variable(s) selected with this adaptive approach. If betaPos = TRUE, this set is the covariates with a positive regression coefficient in beta. Else this set is the covariates with a non null regression coefficient in beta. Covariates are ordering according to the p-values (two-sided if betaPos = FALSE , one-sided if betaPos = TRUE) in the classical multiple logistic regression model that minimzes the BIC in the adaptive lasso.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
acisl <- adapt_cisl(x = drugs, y = ae, cisl_nB = 50, maxp=10)
```

---

adapt_cv *fit an adaptive lasso with adaptive weights derived from lasso-cv*

Description

Fit a first lasso regression with cross-validation to determine adaptive weights, then run an adaptive lasso with this penalty weighting. Cross-validation is used for the adaptive lasso for variable selection. Can deal with very large sparse data matrices. Intended for binary reponse only (option `family = "binomial"` is forced). Depends on the `cv.glmnet` function from the package `glmnet`.

Usage

```
adapt_cv(x, y, gamma = 1, nfolds = 5, foldid = NULL, betaPos = TRUE, ...)
```

Arguments

`x`	Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class `"sparseMatrix"` as in package `Matrix`).
`y`	Binary response variable, numeric.
`gamma`	Tunning parameter to defined the penalty weights. See details below. Default is set to 1.
`nfolds`	Number of folds - default is 5. Although `nfolds` can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets. Smallest value allowable is `nfolds=3`.
`foldid`	An optional vector of values between 1 and `nfolds` identifying what fold each observation is in. If supplied, `nfolds` can be missing.
`betaPos`	Should the covariates selected by the procedure be positively associated with the outcome ? Default is `TRUE`.
`...`	Other arguments that can be passed to `cv.glmnet` from package `glmnet` other than `nfolds`, `foldid`, `penalty.factor` and `family`.

Details

The adaptive weight for a given covariate i is defined by

$$w_i = 1/|\beta^C V_i|^\gamma$$

where $\beta^C V_i$ is the PENALIZED regression coefficient associated to covariate i obtained with cross-validation.

Value

An object with S3 class `"adaptive"`.

`aws` Numeric vector of penalty weights derived from cross-validation. Length equal to nvars.

`criterion` Character, indicates which criterion is used with the adaptive lasso for variable selection. For `adapt_cv` function, `criterion` is "cv".

`beta` Numeric vector of regression coefficients in the adaptive lasso. If `criterion` = "cv" the regression coefficients are PENALIZED, if `criterion` = "bic" the regression coefficients are UNPENALIZED. Length equal to nvars. Could be NA if adaptive weights are all equal to infinity.

`selected_variables` Character vector, names of variable(s) selected with this adaptive approach. If `betaPos` = `TRUE`, this set is the covariates with a positive regression coefficient in `beta`. Else this set is the covariates with a non null regression coefficient in `beta`. Covariates are ordering according to magnitude of their regression coefficients absolute value in the adaptive lasso.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
acv <- adapt_cv(x = drugs, y = ae, nfolds = 5)
```

adapt_ridge *fit an adaptive lasso with adaptive weights derived from ridge regression*

Description

Fit first a ridge regression with cross-validation to determine adaptive weights, then run an adaptive lasso with this penalty weighting. BIC or cross-validation could either be used for the adaptive lasso for variable selection. Can deal with very large sparse data matrices. Intended for binary reponse only (option `family = "binomial"` is forced). Depends on the `glmnet` and `relax.glmnet` function from the package `glmnet`.

Usage

```
adapt_ridge(
  x,
  y,
  gamma = 1,
  criterion = "bic",
  maxp = 50,
  path = TRUE,
  nfolds = 5,
  foldid = NULL,
  betaPos = TRUE,
  ...
)
```

Arguments

`x`	Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class `"sparseMatrix"` as in package `Matrix`).
`y`	Binary response variable, numeric.
`gamma`	Tunning parameter to defined the penalty weights. See details below. Default is set to 1.
`criterion`	Character, indicates which criterion is used with the adaptive lasso for variable selection. Could be either "bic" or "cv". Default is "bic"
`maxp`	Used only if `criterion` = "bic", ignored if `criterion` = "cv". A limit on how many relaxed coefficients are allowed. Default is 50, in `glmnet` option default is 'n-3', where 'n' is the sample size.
`path`	Used only if `criterion` = "bic", ignored if `criterion` = "cv". Since `glmnet` does not do stepsize optimization, the Newton algorithm can get stuck and not converge, especially with relaxed fits. With `path=TRUE`, each relaxed fit on a particular set of variables is computed pathwise using the original sequence of lambda values (with a zero attached to the end). Default is `path=TRUE`.

nfolds	Used for the cross-validation in ridge regression and if criterion = "cv". Number of folds - default is 5. Although nfolds can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets. Smallest value allowable is nfolds=3.
foldid	Used for the cross-validation in ridge regression and if criterion = "cv". An optional vector of values between 1 and nfolds identifying what fold each observation is in. If supplied, nfolds can be missing.
betaPos	Should the covariates selected by the procedure be positively associated with the outcome ? Default is TRUE.
...	Other arguments that can be passed to glmnet from package glmnet other than penalty.factor, alpha, family, maxp and path.

Details

The adaptive weight for a given covariate i is defined by

$$w_i = 1/|\beta_i^{L2}|^\gamma$$

where $\beta^L 2_i$ is the PENALIZED ridge regression coefficient associated to covariate i obtained with cross-validation.

Value

An object with S3 class "adaptive".

aws	Numeric vector of penalty weights derived from ridge regression. Length equal to nvars.
criterion	Character, same as input. Could be either "bic" or "cv".
beta	Numeric vector of regression coefficients in the adaptive lasso. If criterion = "cv" the regression coefficients are PENALIZED, if criterion = "bic" the regression coefficients are UNPENALIZED. Length equal to nvars. Could be NA if adaptive weights are all equal to infinity.
selected_variables	Character vector, names of variable(s) selected with this adaptive approach. If betaPos = TRUE, this set is the covariates with a positive regression coefficient in beta. Else this set is the covariates with a non null regression coefficient in beta. If criterion = "bic", covariates are ordering according to magnitude of their regression coefficients absolute value in the adaptive lasso. If criterion = "bic", covariates are ordering according to the p-values (two-sided if betaPos = FALSE , one-sided if betaPos = TRUE) in the classical multiple logistic regression model that minimzes the BIC in the adaptive lasso.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
ar <- adapt_ridge(x = drugs, y = ae, criterion = "cv", nfolds = 3)
```

`adapt_univ` *fit an adaptive lasso with adaptive weights derived from univariate coefficients*

Description

Compute odd-ratios between each covariate of `x` and `y` then derived adaptive weights to incorporate in an adaptive lasso. BIC or cross-validation could either be used for the adaptive lasso for variable selection. Can deal with very large sparse data matrices. Intended for binary reponse only (option `family = "binomial"` is forced). Depends on the `glmnet` and `relax.glmnet` function from the package `glmnet`.

Usage

```
adapt_univ(
  x,
  y,
  gamma = 1,
  criterion = "bic",
  maxp = 50,
  path = TRUE,
  nfolds = 5,
  foldid = NULL,
  betaPos = TRUE,
  ...
)
```

Arguments

`x`	Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class `"sparseMatrix"` as in package `Matrix`).
`y`	Binary response variable, numeric.
`gamma`	Tunning parameter to defined the penalty weights. See details below. Default is set to 1.
`criterion`	Character, indicates which criterion is used with the adaptive lasso for variable selection. Could be either "bic" or "cv". Default is "bic"

maxp	Used only if criterion = "bic", ignored if criterion = "cv". A limit on how many relaxed coefficients are allowed. Default is 50, in glmnet option default is 'n-3', where 'n' is the sample size.
path	Used only if criterion = "bic", ignored if criterion = "cv". Since glmnet does not do stepsize optimization, the Newton algorithm can get stuck and not converge, especially with relaxed fits. With path=TRUE, each relaxed fit on a particular set of variables is computed pathwise using the original sequence of lambda values (with a zero attached to the end). Default is path=TRUE.
nfolds	Used only if criterion = "cv", ignored if criterion = "bic". Number of folds - default is 5. Although nfolds can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets. Smallest value allowable is nfolds=3.
foldid	Used only if criterion = "cv", ignored if criterion = "bic". An optional vector of values between 1 and nfolds identifying what fold each observation is in. If supplied, nfolds can be missing.
betaPos	Should the covariates selected by the procedure be positively associated with the outcome ? Default is TRUE.
...	Other arguments that can be passed to glmnet from package glmnet other than penalty.factor, family, maxp and path.

Details

The adaptive weight for a given covariate i is defined by

$$w_i = 1/|\beta^u niv_i|^\gamma$$

where $\beta^u niv_i = log(OR_i)$, with OR_i is the odd-ratio associated to covariate i with the outcome.

Value

An object with S3 class "adaptive".

aws	Numeric vector of penalty weights derived from odds-ratios. Length equal to nvars.
criterion	Character, same as input. Could be either "bic" or "cv".
beta	Numeric vector of regression coefficients in the adaptive lasso. If criterion = "cv" the regression coefficients are PENALIZED, if criterion = "bic" the regression coefficients are UNPENALIZED. Length equal to nvars. Could be NA if adaptive weights are all equal to infinity.
selected_variables	Character vector, names of variable(s) selected with this adaptive approach. If betaPos = TRUE, this set is the covariates with a positive regression coefficient in beta. Else this set is the covariates with a non null regression coefficient in beta. If criterion = "bic", covariates are ordering according to magnitude of their regression coefficients absolute value in the adaptive lasso. If criterion = "bic", covariates are ordering according to the p-values (two-sided if betaPos = FALSE , one-sided if betaPos = TRUE) in the classical multiple logistic regression model that minimzes the BIC in the adaptive lasso.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
au <- adapt_univ(x = drugs, y = ae, criterion ="cv", nfolds = 3)
```

---

cisl *Class Imbalanced Subsampling Lasso*

Description

Implementation of CISL and the stability selection according to subsampling options.

Usage

```
cisl(
  x,
  y,
  r = 4,
  nB = 100,
  dfmax = 50,
  nlambda = 250,
  nMin = 0,
  replace = TRUE,
  betaPos = TRUE,
  ncore = 1
)
```

Arguments

x	Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class "sparseMatrix" as in package Matrix).
y	Binary response variable, numeric.
r	Number of control in the CISL sampling. Default is 4. See details below for other implementations.
nB	Number of sub-samples. Default is 100.

dfmax	Corresponds to the maximum size of the models visited with the lasso (E in the paper). Default is 50.
nlambda	Number of lambda values as is glmnet documentation. Default is 250.
nMin	Minimum number of events for a covariate to be considered. Default is 0, all the covariates from x are considered.
replace	Should sampling be with replacement? Default is TRUE.
betaPos	If betaPos=TRUE, variable selection is based on positive regression coefficient. Else, variable selection is based on non-zero regression coefficient. Default is TRUE.
ncore	The number of calcul units used for parallel computing. This has to be set to 1 if the parallel package is not available. Default is 1. WARNING: parallel computing is not supported for windows machines!

Details

CISL is a variation of the stability method adapted to characteristics of pharmacovigilance databases. Tunning r = 4 and replace = TRUE are used to implement our CISL sampling. For instance, r = NULL and replace = FALSE can be used to implement the $\frac{n}{2}$ sampling in Stability Selection.

Value

An object with S3 class "cisl".

prob	Matrix of dimension nvars x nB. Quantity compute by CISL for each covariate, for each subsample.
q05	5 % quantile of the CISL quantity for each covariates. Numeric, length equal to nvars.
q10	10 % quantile of the CISL quantity for each covariates. Numeric, length equal to nvars.
q15	15 % quantile of the CISL quantity for each covariates. Numeric, length equal to nvars.
q20	20 % quantile of the CISL quantity for each covariates. Numeric, length equal to nvars.

Author(s)

Ismail Ahmed

References

Ahmed, I., Pariente, A., & Tubert-Bitter, P. (2018). "Class-imbalanced subsampling lasso algorithm for discovering adverse drug reactions". *Statistical Methods in Medical Research*. 27(3), 785–797, https://doi.org/10.1177/0962280216643116

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
lcisl <- cisl(x = drugs, y = ae, nB = 50)
```

`data_PV` *Simulated data for the adapt4pv package*

Description

Simple simulated data, used to demonstrate the features of functions from adapt4cv package.

Format

X large sparse and binary matrix with 117160 rows and 300 columns. Drug matrix exposure: each row corresponds to an individual and each column corresponds to a drug.

Y large spase and binary vector of length 117160. Indicator of the presence/absence of an adverse event for ech individual. Only the first 30 drugs (out of the 300) are associated with the outcome.

Examples

```
data(ExamplePvData)
```

`est_ps_bic` *propensity score estimation in high dimension with automated covariates selection using lasso-bic*

Description

Estimate a propensity score to a given drug exposure by (i) selecting among other drug covariates in `x` which ones to include in the PS estimation model automatically using lasso-bic approach, (ii) estimating a score using a classical logistic regression with the afore selected covariates. Internal function, not supposed to be used directly.

Usage

```
est_ps_bic(idx_expo, x, penalty = rep(1, nvars - 1), ...)
```

Arguments

idx_expo	Index of the column in x that corresponds to the drug covariate for which we aim at estimating the PS.
x	Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class "sparseMatrix" as in package Matrix).
penalty	TEST OPTION penalty weights in the variable selection to include in the PS.
...	Other arguments that can be passed to glmnet from package glmnet other than penalty.factor, family, maxp and path.

Details

betaPos option of lasso_bic function is set to FALSE and maxp is set to 20. For optimal storage, the returned elements indicator_expo and score are Matrix with ncol = 1.

Value

An object with S3 class "ps","bic".

expo_name	Character, name of the drug exposure for which the PS was estimated. Correspond to colnames(x)[idx_expo]
.	
indicator_expo	One-column Matrix object. Indicator of the drug exposure for which the PS was estimated. Defined by x[,idx_expo].
.	
score_variables	Character vector, names of covariates(s) selected with the lasso-bic approach to include in the PS estimation model. Could be empty.
score	One-column Matrix object, the estimated score.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
psb2 <- est_ps_bic(idx_expo = 2, x = drugs)
psb2$score_variables #selected variables to include in the PS model of drug_2
```

`est_ps_hdps` *propensity score estimation in high dimension with automated covariates selection using hdPS*

Description

Estimate a propensity score to a given drug exposure by (i) selecting among other drug covariates in `x` which ones to include in the PS estimation model automatically using hdPS algorithm, (ii) estimating a score using a classical logistic regression with the afore selected covariates. Internal function, not supposed to be used directly.

Usage

```
est_ps_hdps(idx_expo, x, y, keep_total = 20)
```

Arguments

`idx_expo`	Index of the column in `x` that corresponds to the drug covariate for which we aim at estimating the PS.
`x`	Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class `"sparseMatrix"` as in package `Matrix`).
`y`	Binary response variable, numeric.
`keep_total`	number of covariates to include in the PS estimation model according to the hdps algorithm ordering. Default is 20.

Details

Compared to the situation of the classic use of hdps (i) there is only one dimension (the co-exposition matrix) (ii) no need to expand covariates since they are already binary. In other words, in our situation hdps consists in the "prioritize covariates" step from the original algorithm, using Bross formula. We consider the correction on the interpretation on this formula made by Richard Wyss (drug epi).

Value

An object with S3 class `"ps"`,`"hdps"`.

`expo_name`	Character, name of the drug exposure for which the PS was estimated. Correspond to `colnames(x)[idx_expo]`
.	
`indicator_expo`	One-column Matrix object. Indicator of the drug exposure for which the PS was estimated. Defined by `x[,idx_expo]`.
.	

score_variables
: Character vector, names of covariates(s) selected with the hdPS algorithm to include in the PS estimation model. Could be empty.

score
: One-column Matrix object, the estimated score.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

References

Schneeweiss, S., Rassen, J. A., Glynn, R. J., Avorn, J., Mogun, H., Brookhart, M. A. (2009). "High-dimensional propensity score adjustment in studies of treatment effects using health care claims data". *Epidemiology*. 20, 512–522, https://doi.org/10.1097/EDE.0b013e3181a663cc

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
pshdps2 <- est_ps_hdps(idx_expo = 2, x = drugs, y = ae, keep_total = 10)
pshdps2$score_variables #selected variables to include in the PS model of drug_2
```

---

est_ps_xgb *propensity score estimation in high dimension using gradient tree boosting*

Description

Estimate a propensity score to a given drug exposure (treatment) with extreme gradient boosting. Depends on xgboost package. Internal function, not supposed to be used directly.

Usage

```
est_ps_xgb(
  idx_expo,
  x,
 parameters = list(eta = 0.1, max_depth = 6, objective = "binary:logistic", nthread =
    1),
  nrounds = 200,
  ...
)
```

Arguments

idx_expo	Index of the column in x that corresponds to the drug covariate for which we aim at estimating the PS.
x	Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class "sparseMatrix" as in package Matrix).
parameters	correspond to params in xgb.train function. The complete list of parameters is available at http://xgboost.readthedocs.io/en/latest/parameter.html. Default is a list with eta=0.1 (learning rate), max_depth = 6 (maximum length of a tree), objective = "binary:logistic" and nthread = 1 (number of threads for parallelization).
nrounds	Maximum number of boosting iterations. Default is 200.
...	Other arguments that can be passed to xgb.train function.

Value

An object with S3 class "ps","xgb".

expo_name	Character, name of the drug exposure for which the PS was estimated. Correspond to colnames(x)[idx_expo]
.	
indicator_expo	One-column Matrix object. Indicator of the drug exposure for which the PS was estimated. Defined by x[,idx_expo].
.	
score_variables	Character vector, names of covariates(s) used in a at list one tree in the gradient tree boosting algorithm. Obtained with xgb.importance function from xgboost package.
score	One-column Matrix object, the estimated score.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
psxgb2 <- est_ps_xgb(idx_expo = 2, x = drugs, nrounds = 100)
psxgb2$score_variables #selected variables to include in the PS model of drug_2
```

lasso_bic *fit a lasso regression and use standard BIC for variable selection*

Description

Fit a lasso regression and use the Bayesian Information Criterion (BIC) to select a subset of selected covariates. Can deal with very large sparse data matrices. Intended for binary reponse only (option `family = "binomial"` is forced). Depends on the `glmnet` and `relax.glmnet` functions from the package `glmnet`.

Usage

```
lasso_bic(x, y, maxp = 50, path = TRUE, betaPos = TRUE, ...)
```

Arguments

`x`	Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class `"sparseMatrix"` as in package `Matrix`).
`y`	Binary response variable, numeric.
`maxp`	A limit on how many relaxed coefficients are allowed. Default is 50, in `glmnet` option default is 'n-3', where 'n' is the sample size.
`path`	Since `glmnet` does not do stepsize optimization, the Newton algorithm can get stuck and not converge, especially with relaxed fits. With `path=TRUE`, each relaxed fit on a particular set of variables is computed pathwise using the original sequence of lambda values (with a zero attached to the end). Default is `path=TRUE`.
`betaPos`	Should the covariates selected by the procedure be positively associated with the outcome ? Default is `TRUE`.
`...`	Other arguments that can be passed to `glmnet` from package `glmnet` other than `penalty.factor`, `family`, `maxp` and `path`.

Details

For each tested penalisation parameter λ, a standard version of the BIC is implemented.

$$BIC_\lambda = -2l_\lambda + df(\lambda) * ln(N)$$

where l_λ is the log-likelihood of the non-penalized multiple logistic regression model that includes the set of covariates with a non-zero coefficient in the penalised regression coefficient vector associated to λ, and $df(\lambda)$ is the number of covariates with a non-zero coefficient in the penalised regression coefficient vector associated to λ, The optimal set of covariates according to this approach is the one associated with the classical multiple logistic regression model which minimizes the BIC.

Value

An object with S3 class "log.lasso".

beta — Numeric vector of regression coefficients in the lasso. In lasso_bic function, the regression coefficients are UNPENALIZED. Length equal to nvars.

selected_variables — Character vector, names of variable(s) selected with the lasso-bic approach. If betaPos = TRUE, this set is the covariates with a positive regression coefficient in beta. Else this set is the covariates with a non null regression coefficient in beta. Covariates are ordering according to the p-values (two-sided if betaPos = FALSE , one-sided if betaPos = TRUE) in the classical multiple logistic regression model that minimzes the BIC.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
lb <- lasso_bic(x = drugs, y = ae, maxp = 20)
```

---

lasso_cv *wrap function for* cv.glmnet

Description

Fit a first cross-validation on lasso regression and return selected covariates. Can deal with very large sparse data matrices. Intended for binary reponse only (option family = "binomial" is forced). Depends on the cv.glmnet function from the package glmnet.

Usage

```
lasso_cv(x, y, nfolds = 5, foldid = NULL, betaPos = TRUE, ...)
```

Arguments

x — Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class "sparseMatrix" as in package Matrix).

y — Binary response variable, numeric.

nfolds	Number of folds - default is 5. Although nfolds can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets. Smallest value allowable is nfolds=3.
foldid	An optional vector of values between 1 and nfolds identifying what fold each observation is in. If supplied, nfolds can be missing.
betaPos	Should the covariates selected by the procedure be positively associated with the outcome ? Default is TRUE.
...	Other arguments that can be passed to cv.glmnet from package glmnet other than nfolds, foldid, and family.

Value

An object with S3 class "log.lasso".

beta	Numeric vector of regression coefficients in the lasso. In lasso_cv function, the regression coefficients are PENALIZED. Length equal to nvars.
selected_variables	Character vector, names of variable(s) selected with the lasso-cv approach. If betaPos = TRUE, this set is the covariates with a positive regression coefficient in beta. Else this set is the covariates with a non null regression coefficient in beta. Covariates are ordering according to magnitude of their regression coefficients absolute value.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
lcv <- lasso_cv(x = drugs, y = ae, nfolds = 3)
```

---

lasso_perm *fit a lasso regression and use standard permutation of the outcome for variable selection*

Description

Performed K lasso logistic regression with K different permuted version of the outcome. For earch of the lasso regression, the λ_max(i.e. the smaller λ such as all penalized regression coefficients are shrunk to zero) is obtained. The median value of these K λ_max is used to for variable selection in the lasso regression with the non-permuted outcome. Depends on the glmnet function from the package glmnet.

Usage

```
lasso_perm(x, y, K = 20, keep = NULL, betaPos = TRUE, ncore = 1, ...)
```

Arguments

x	Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class "sparseMatrix" as in package Matrix).
y	Binary response variable, numeric.
K	Number of permutations of y. Default is 20.
keep	Do some variables of x have to be permuted in the same way as y? Default is NULL, means no. If yes, must be a vector of covariates indices. TEST OPTION
betaPos	Should the covariates selected by the procedure be positively associated with the outcome ? Default is TRUE.
ncore	The number of calcul units used for parallel computing. Default is 1, no parallelization is implemented.
...	Other arguments that can be passed to glmnet from package glmnet other than family.

Details

The selected λ with this approach is defined as the closest λ from the median value of the K λ_max obtained with permutation of the outcome.

Value

An object with S3 class "log.lasso".

beta	Numeric vector of regression coefficients in the lasso In lasso_perm function, the regression coefficients are PENALIZED. Length equal to nvars.
selected_variables	Character vector, names of variable(s) selected with the lasso-perm approach. If betaPos = TRUE, this set is the covariates with a positive regression coefficient in beta. Else this set is the covariates with a non null regression coefficient in beta. Covariates are ordering according to magnitude of their regression coefficients absolute value.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

References

Sabourin, J. A., Valdar, W., & Nobel, A. B. (2015). "A permutation approach for selecting the penalty parameter in penalized model selection". *Biometrics*. 71(4), 1185–1194, https://doi.org/10.1111/biom.12359

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
lp <- lasso_perm(x = drugs, y = ae, K = 10)
```

`ps_adjust` *adjustment on propensity score*

Description

Implement the adjustment on propensity score for all the drug exposures of the input drug matrix x which have more than a given number of co-occurence with the outcome. The binary outcome is regressed on a drug exposure and its estimated PS, for each drug exposure considered after filtering. With this approach, a p-value is obtained for each drug and a variable selection is performed over the corrected for multiple comparisons p-values.

Usage

```
ps_adjust(
  x,
  y,
  n_min = 3,
  betaPos = TRUE,
  est_type = "bic",
  threshold = 0.05,
  ncore = 1
)
```

Arguments

`x`	Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class `"sparseMatrix"` as in package `Matrix`).
`y`	Binary response variable, numeric.
`n_min`	Numeric, Minimal number of co-occurence between a drug covariate and the outcome y to estimate its score. See details belows. Default is 3.
`betaPos`	Should the covariates selected by the procedure be positively associated with the outcome ? Default is `TRUE`.
`est_type`	Character, indicates which approach is used to estimate the PS. Could be either "bic", "hdps" or "xgb". Default is "bic".
`threshold`	Threshold for the p-values. Default is 0.05.
`ncore`	The number of calcul units used for parallel computing. Default is 1, no parallelization is implemented.

Details

The PS could be estimated in different ways: using lasso-bic approach, the hdps algorithm or gradient tree boosting. The scores are estimated using the default parameter values of `est_ps_bic`, `est_ps_hdps` and `est_ps_xgb` functions (see documentation for details). We apply the same filter and the same multiple testing correction as in the paper UPCOMING REFERENCE: first, PS are estimated only for drug covariates which have more than `n_min` co-occurence with the outcome `y`. Adjustment on the PS is performed for these covariates and one sided or two-sided (depend on `betaPos` parameter) p-values are obtained. The p-values of the covariates not retained after filtering are set to 1. All these p-values are then adjusted for multiple comparaison with the Benjamini-Yekutieli correction. COULD BE VERY LONG. Since this approach (i) estimate a score for several drug covariates and (ii) perform an adjustment on these scores, parallelization is highly recommanded.

Value

An object with S3 class `"ps"`,`"adjust"`,`"*"`, where `"*"` is `"bic"`, `"hdps"` or `"xgb"`according on how the score were estimated.

`estimates` Regression coefficients associated with the drug covariates. Numeric, length equal to the number of selected variables with this approach. Some elements could be NA if (i) the corresponding covariate was filtered out, (ii) adjustment model did not converge. Trying to estimate the score in a different way could help, but it's not insured.

`corrected_pvals` One sided p-values if `betaPos = TRUE`, two-sided p-values if `betaPos = FALSE` adjusted for multiple testing. Numeric, length equal to nvars.

`selected_variables` Character vector, names of variable(s) selected with the ps-adjust approach. If `betaPos = TRUE`, this set is the covariates with a corrected one-sided p-value lower than `threshold`. Else this set is the covariates with a corrected two-sided p-value lower than `threshold`. Covariates are ordering according to their corrected p-value.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

References

Benjamini, Y., & Yekuteli, D. (2001). "The Control of the False Discovery Rate in Multiple Testing under Dependency". *The Annals of Statistics*. 29(4), 1165–1188, doi: doi: 10.1214/aos/1013699998.

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
```

```
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
adjps <- ps_adjust(x = drugs, y = ae, n_min = 10)
```

---

ps_adjust_one *adjustment on propensity score for one drug exposure*

Description

Implement the adjustment on propensity score for one drug exposure. The binary outcome is regressed on the drug exposure of interest and its estimated PS. Internal function, not supposed to be used directly.

Usage

```
ps_adjust_one(ps_est, y)
```

Arguments

ps_est	An object of class "ps","*" where "*" is "bic", "hdps" or "xgb" according on how the score was estimated, respective outputs of internal functions est_ps_bic, est_ps_hdps, est_ps_xgb. It is a list with the following elements : * score_type: character, name of the drug exposure for which the PS was estimated. * indicator_expo: indicator of the drugs exposure for which the PS was estimated. One-column Matrix object. * score_variables: Character vector, names of covariate(s) selected to include in the PS estimation model. Could be empty. *score: One-column Matrix object, the estimated score.
y	Binary response variable, numeric.

Details

The PS could be estimated in different ways: using lasso-bic approach, the hdPS algorithm or gradient tree boosting using functions est_ps_bic, est_ps_hdps and est_ps_xgb respectivelly.

Value

An object with S3 class "ps","adjust"

expo_name	Character, name of the drug exposure for which the PS was estimated.
estimate	Regression coefficient associated with the drug exposure in adjustment on PS.
pval_1sided	One sided p-value associated with the drug exposure in adjustment on PS.
pval_2sided	Two sided p-value associated with the drug exposure in adjustment on PS.

Could return NA if the adjustment on the PS did not converge.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
pshdps2 <- est_ps_hdps(idx_expo = 2, x = drugs, y = ae, keep_total = 10)
adjps2 <- ps_adjust_one(ps_est = pshdps2, y = ae)
adjps2$estimate #estimated strength of association between drug_2 and the outcome by PS adjustment
```

---

ps_pond *weihting on propensity score*

Description

Implement the weighting on propensity score with Matching Weights (MW) or the Inverse Probability of Treatment Weighting (IPTW) for all the drug exposures of the input drug matrix `x` which have more than a given number of co-occurence with the outcome. The binary outcome is regressed on a drug exposure through a classical weighted regression, for each drug exposure considered after filtering. With this approach, a p-value is obtained for each drug and a variable selection is performed over the corrected for multiple comparisons p-values.

Usage

```
ps_pond(
  x,
  y,
  n_min = 3,
  betaPos = TRUE,
  weights_type = c("mw", "iptw"),
  truncation = FALSE,
  q = 0.025,
  est_type = "bic",
  threshold = 0.05,
  ncore = 1
)
```

Arguments

x	Input matrix, of dimension nobs x nvars. Each row is an observation vector. Can be in sparse matrix format (inherit from class `"sparseMatrix"` as in package `Matrix`).

y	Binary response variable, numeric.
n_min	Numeric, Minimal number of co-occurence between a drug covariate and the outcome y to estimate its score. See details belows. Default is 3.
betaPos	Should the covariates selected by the procedure be positively associated with the outcome ? Default is TRUE.
weights_type	Character. Indicates which type of weighting is implemented. Could be either "mw" or "iptw".
truncation	Bouleen, should we do weight truncation? Default is FALSE.
q	If truncation is TRUE, quantile value for weight truncation. Ignored if truncation is FALSE. Default is 2.5 %.
est_type	Character, indicates which approach is used to estimate the propensity score. Could be either "bic", "hdps" or "xgb". Default is "bic".
threshold	Threshold for the p-values. Default is 0.05.
ncore	The number of calcul units used for parallel computing. Default is 1, no parallelization is implemented.

Details

The MW are defined by

$$mw_i = min(PS_i, 1 - PS_i)/[(expo_i) * PS_i + (1 - expo_i) * (1 - PS_i)]$$

and weights from IPTW by

$$iptw_i = expo_i/PS_i + (1 - expo_i)/(1 - PS_i)$$

where $expo_i$ is the drug exposure indicator. The PS could be estimated in different ways: using lasso-bic approach, the hdps algorithm or gradient tree boosting. The scores are estimated using the default parameter values of est_ps_bic, est_ps_hdps and est_ps_xgb functions (see documentation for details). We apply the same filter and the same multiple testing correction as in the paper UPCOMING REFERENCE: first, PS are estimated only for drug covariates which have more than n_min co-occurence with the outcome y. Adjustment on the PS is performed for these covariates and one sided or two-sided (depend on betaPos parameter) p-values are obtained. The p-values of the covariates not retained after filtering are set to 1. All these p-values are then adjusted for multiple comparaison with the Benjamini-Yekutieli correction. COULD BE VERY LONG. Since this approach (i) estimate a score for several drug covariates and (ii) perform an adjustment on these scores, parallelization is highly recommanded.

Value

An object with S3 class "ps", "*" , "**" , where "*" is "mw" or "iptw", same as the input parameter weights_type, and "**" is "bic", "hdps" or "xgb" according on how the score was estimated.

estimates	Regression coefficients associated with the drug covariates. Numeric, length equal to the number of selected variables with this approach. Some elements could be NA if (i) the corresponding covariate was filtered out, (ii) weigted regression did not converge. Trying to estimate the score in a different way could help, but it's not insured.

corrected_pvals
: One sided p-values if betaPos = TRUE, two-sided p-values if betaPos = FALSE adjusted for multiple testing. Numeric, length equal to nvars.

selected_variables
: Character vector, names of variable(s) selected with the weighting on PS based approach. If betaPos = TRUE, this set is the covariates with a corrected one-sided p-value lower than threshold. Else this set is the covariates with a corrected two-sided p-value lower than threshold. Covariates are ordering according to their corrected p-value.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

References

Benjamini, Y., & Yekuteli, D. (2001). "The Control of the False Discovery Rate in Multiple Testing under Dependency". *The Annals of Statistics*. 29(4), 1165–1188, doi: doi: 10.1214/aos/1013699998.

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
pondps <- ps_pond(x = drugs, y = ae, n_min = 10, weights_type = "iptw")
```

---

ps_pond_one *weihting on propensity score for one drug exposure*

Description

Implement the weighting on propensity score with Matching Weights (MW) or the Inverse Probability of Treatment Weighting (IPTW) for one drug exposure. The binary outcome is regressed on the drug exposure of interest through a classical weighted regression. Internal function, not supposed to be used directly.

Usage

```
ps_pond_one(
  ps_est,
  y,
  weights_type = c("mw", "iptw"),
```

```
  truncation = FALSE,
  q = 0.025
)
```

Arguments

ps_est	An object of class "ps","*" where "*" is "bic", "hdps" or "xgb" according on how the score was estimated, respective outputs of internal functions est_ps_bic, est_ps_hdps, est_ps_xgb. It is a list with the following elements : * score_type: character, name of the drug exposure for which the PS was estimated. * indicator_expo: indicator of the drugs exposure for which the PS was estimated. One-column Matrix object. * score_variables: Character vector, names of covariate(s) selected to include in the PS estimation model. Could be empty. *score: One-column Matrix object, the estimated score.
y	Binary response variable, numeric.
weights_type	Character. Indicates which type of weighting is implemented. Could be either "mw" or "iptw".
truncation	Bouleen, should we do weight truncation? Default is FALSE.
q	If truncation is TRUE, quantile value for weight truncation. Ignored if truncation is FALSE. Default is 2.5 $\%$.

Details

The MW are defined by

$$mw_i = min(PS_i, 1 - PS_i)/[(expo_i) * PS_i + (1 - expo_i) * (1 - PS_i)]$$

and weights from IPTW by

$$iptw_i = expo_i/PS_i + (1 - expo_i)/(1 - PS_i)$$

where $expo_i$ is the drug exposure indicator. The PS could be estimated in different ways: using lasso-bic approach, the hdPS algorithm or gradient tree boosting using functions est_ps_bic, est_ps_hdps and est_ps_xgb respectivelly.

Value

An object with S3 class "ps","*" , where "*" is "mw" or "iptw", same as the input parameter weights_type

expo_name	Character, name of the drug exposure for which the PS was estimated.
estimate	Regression coefficient associated with the drug exposure in adjustment on PS.
pval_1sided	One sided p-value associated with the drug exposure in adjustment on PS.
pval_2sided	Two sided p-value associated with the drug exposure in adjustment on PS.

Could return NA if the adjustment on the PS did not converge.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
pshdps2 <- est_ps_hdps(idx_expo = 2, x = drugs, y = ae, keep_total = 10)
pondps2 <- ps_pond_one(ps_est = pshdps2, y = ae, weights_type = "iptw")
pondps2$estimate #estimated strength of association between drug_2 and the outcome by PS weighting
```

---

summary_stat *Summary statistics for main adapt4pv package functions*

Description

Return the Sensitivity and the False Discovery Rate of an approach implemeted by the main functions of adapt4pv package.

Usage

```
summary_stat(object, true_pos, q = 10)
```

Arguments

object	An object of class "log.lasso", "cisl", "adaptive" and "*","ps","**" where "*" is either "adjust", "iptw" or "mw" and "**" is either "bic", "hdps" or "xgb".
true_pos	Character vector, names of the true positives controls
q	Quantile value for variable selection with an object of class "cisl". Possible values are 5, 10, 15, 20. Default is 10

Value

A data frame wich details for the signal detection method implemented in object: its number of generated signals, its sensitivity and its false discovery rate.

Author(s)

Emeline Courtois
Maintainer: Emeline Courtois <emeline.courtois@inserm.fr>

Examples

```
set.seed(15)
drugs <- matrix(rbinom(100*20, 1, 0.2), nrow = 100, ncol = 20)
colnames(drugs) <- paste0("drugs",1:ncol(drugs))
ae <- rbinom(100, 1, 0.3)
lcv <- lasso_cv(x = drugs, y = ae, nfolds = 3)
summary_stat(object = lcv, true_pos = colnames(drugs)[1:10])
# the data are not simulated in such a way that there are true positives
```

Index

∗Topic **datasets**
data_PV, 14

adapt4pv-package, 2
adapt_bic, 2
adapt_cisl, 4
adapt_cv, 6
adapt_ridge, 8
adapt_univ, 10

cisl, 12

data_PV, 14

est_ps_bic, 14
est_ps_hdps, 16
est_ps_xgb, 17

lasso_bic, 19
lasso_cv, 20
lasso_perm, 21

ps_adjust, 23
ps_adjust_one, 25
ps_pond, 26
ps_pond_one, 28

summary_stat, 30

X *(*data_PV*)*, 14

Y *(*data_PV*)*, 14