Package 'pact'

April 15, 2016

Type Package

Title Predictive Analysis of Clinical Trials

Version 0.5.0

Date 2016-04-14

Description A prediction-based approach to the analysis of data from randomized clinical trials is implemented. Based on response and covariate data from a randomized clinical trial comparing a new experimental treatment E versus a control C, the objective is to develop and internally validate a model that can identify subjects likely to benefit from E rather than C. Currently, survival and binary response types are permitted.

License GPL-3

Imports survival, glmnet

Depends R (>= 2.10)

URL `https://github.com/brbnci/pact`

RoxygenNote 5.0.1

NeedsCompilation no

Author Jyothi Subramanian [aut, cre],
Richard Simon [aut]

Maintainer Jyothi Subramanian <`subramanianj01@gmail.com`>

Repository CRAN

Date/Publication 2016-04-15 23:39:12

R topics documented:

`EORTC10994` *EORTC10994 dataset*

Description

A dataset containing treatment, response and covariate information for 125 subjects with breast cancer. See 'Details' for the variables in this dataset.

Format

A data frame with 125 rows and 7 variables

Details

- ID. Subject identifier
- Treatment. Treatment received. '0' for control and '1' for experimental
- Response. Binary response to treatment - '0' for 'non-responder' and '1' for 'responder'
- Age. Age (in years) at diagnosis
- TumorSize. size of tumor
- Node. Node positive (Yes) or node negative (No)
- ERBB2Log2. log2 transformed ERBB2 levels

`eval.pact.cv` *Evaluation functions for cross-validated predictions*

Description

Methods for the evaluation of the cross-validated predictive scores obtained from `pact.cv`

Usage

```
eval.pact.cv(out.cv, method = c("discrete", "continuous"), g = log(1),
  plot.score = TRUE, plot.time = NULL, perm.test = FALSE, nperm = 100)
```

Arguments

`out.cv` The object from `pact.cv`

`method` The evaluation method. Currently two options, `method='discrete'` or `method='continuous'`, are available. See 'Details'.

`g` The cut-point for grouping scores into subsets 'benefit' and 'no benefit' from new treatment. Ignored for `method='continuous'`.

`plot.score` Used only for plots if `method='continuous'` is chosen for survival response. Logical representing whether survival curves at specific quantiles of cross-validated scores are to be drawn. See 'Details'.

`plot.time` Used only for plots if `method='continuous'` is chosen for survival response. Probability of survival greater than `plot.time` is plotted as a function of cross-validated score and Treatment. See 'Details'.

`perm.test` Logical. If `perm.test=TRUE`, a permutation based test of significance is conducted for statistics computed from the cross-validated scores. See 'Value' and the package vignette and for more details on the permutation tests.

`nperm` The number of permutations for the permutation test. Ignored if `perm.test=FALSE`

Details

Currently two methods are defined for the evaluation of the scores obtained from `pact.cv`. In `method='discrete'` a user specified cut-off score is used to classify the subjects into groups 'benefit' or 'do not benefit' from new treatment. In each of the 'benefit' and 'do not benefit' groups the actual responses in the control (C) and the experimental (E) groups are compared. For the 'cox' family, the 'score' for a subject represents the predicted change in the log hazard when the subject is treated with E as against C (with lower values denoting benefit with E). In the case of the 'binomial' family, the 'score' represents the predicted change in the log odds of a response when the subject is treated with E as against C (with higher values denoting benefit with E). For the 'cox' family, examples of the cut-point g could be `g=log(1)` with score < g meaning benefit with E. Or one could be more stringent and have g correspond to a 30% reduction in hazard (`g=log(0.70)`). For the 'binomial' family, `g=log(1.20)` with score > g meaning sensitive to E would mean that subjects predicted to receive at least 20% increase in odds of response with E are classified as benefitting from E.

In `method='continuous'` no cut-off is applied to the cross-validated scores. A Cox proportional hazards (PH) regression or a logistic regression model (respectively for 'survival' and 'binary' response) is then developed that includes the main effect of treatment, main effect of cross-validated score, and treatment*score interaction. For survival response, this model is used to generate the Kaplan Meier survival curves for each treatment at the at 20th, 40th, 60th and 80th percentiles of predictive scores (`plot.score = TRUE`). The model is also used to compute the estimated probability of surviving beyond a landmark time specified in `plot.time` as a function of treatment and (cross-validated) score (if `plot.time = NULL`, this plot is not produced). For binary response, the output from evaluation is a plot of the probability of response as a functions of the predictive score and Treatment.

If `perm.test=TRUE`, permutation based significance tests are performed on appropriate test statistics and p-values are computed. See 'Value' and the package vignette and for more details on the permutation tests.

Value

The return object is of class `eval.cv` and is a list whose components depend on the family ('cox' or 'binomial') and the chosen evaluation method ('continuous' or 'discrete')

`LR.Benefit` For `family='cox'` and `method='discrete'`. The log-rank statistic for the survival difference between E and C for the 'benefit' from E group.

`LR.NoBenefit` For `family='cox'` and `method='discrete'`. The log-rank statistic for the survival difference between E and C for the 'do not benefit' from E group.

`RR.T.Benefit` For `family='binomial'` and `method='discrete'`. The response rate for subjects getting E in the 'benefit' from E group.

`RR.C.Benefit` For `family='binomial'` and `method='discrete'`. The response rate for subjects getting C in the 'benefit' from E group

`RR.T.NoBenefit` For `family='binomial'` and `method='discrete'`. The response rate for subjects getting E in the 'do not benefit' from E group

`RR.C.NoBenefit` For `family='binomial'` and `method='discrete'`. The response rate for subjects getting C in the 'do not benefit' from E group

.

`pval.Benefit` If `perm.test=TRUE`, p-value from permutation test. For `family='cox'` and `method='discrete'`, permutation based p-value for LR.Benefit. For `family='binomial'` and `method='discrete'`, permutation based p-value for difference in response rates for E and C for the subset predicted 'benefit' from E.

`pval.NoBenefit` If `perm.test=TRUE`, p-value from permutation test. For `family='cox'` and `method='discrete'`, permutation based p-value for LR.NoBenefit. For `family='binomial'` and `method='discrete'`, permutation based p-value for difference in response rates for E and C for the subset predicted 'no benefit' from E.

`reg` For `method='continuous'`, the regression model with treatment, predictive score and treatment x predictive score interaction

`pval.twosided` For `method='continuous'`. Two-sided (non-directional) permutation based p-value for the treatment x predictive score interaction coefficient

`pval.onesided` For `method='continuous'`. One-sided (directional, greater) permutation based p-value for the treatment x predictive score interaction coefficient

`call` The function call

Additional plots for both `method='discrete'` as well as `method='continuous'`. `print` method is available for a nice display of objects of class `eval.cv`. See package vignette.

Author(s)

Jyothi Subramanian and Richard Simon
Maintainer: Jyothi Subramanian <`<subramanianj01@gmail.com>`>

Examples

```
### Survival response
data(prostateCancer)
Y <- prostateCancer[,3:4]
Xf <- prostateCancer[,7:8]
Xv <- prostateCancer[,c(5:6,9)]
Treatment <- as.factor(prostateCancer[,2])
p <- pact.fit(Y=Y, Xf=Xf, Xv=Xv, Treatment=Treatment, family="cox", varSelect="univar")
cv <- pact.cv(p, nfold=5)
## Not run: eval.pact.cv(cv, method="discrete", g=log(0.80), perm.test=TRUE, nperm=500)  ## At least 20% predicte
eval.pact.cv(cv, method="continuous", plot.score=TRUE, perm.test=FALSE)

### Binary response
data(EORTC10994)
Y <- as.factor(EORTC10994[,4])
Xv <- EORTC10994[,c(2,5:7)]
Treatment <- as.factor(EORTC10994[,3])
p <- pact.fit(Y=Y,Xv=Xv,Treatment=Treatment,family="binomial", varSelect="univar")
cv <- pact.cv(p, nfold=5)
## Not run: eval.pact.cv(cv, method="discrete", g=log(1), perm.test=TRUE, nperm=500)
```

GSE10846 *GSE10846 dataset*

Description

A dataset containing the survival, treatment and gene expression data for 412 patients with diffuse large B cell lymphoma and treated with CHOP or CHOP+Rituximab

Format

A data frame with 412 rows and 1003 columns

Details

- time. Survival time in months
- status. Censoring status. ’0’ for censored, ’1’ for died
- Treatment. Treatment received. ’0’ for control (CHOP) and ’1’ for new (CHOP+Rituximab)
- Columns 4-1003 are gene expression values (normalized using the MAS 5.0 software and log2 transformed) of the first 1000 genes highest variance from the original dataset from GEO
- The row names are the array names

KfoldCV *Split a dataset into k parts for k-fold cross-validation*

Description

Split a dataset into k parts for k-fold cross-validation. This function is used in `pact.cv` to create the splits for cross-validation

Usage

```
KfoldCV(n, k)
```

Arguments

`n`	The sample size
`k`	The number of folds. k=n would mean a leave-one-out cross-validation

Value

A integer vector of same length as n. Each observation is an integer taking a value between 1 to k denoting the fold it belongs to.

Author(s)

Jyothi Subramanian and Richard Simon
Maintainer: Jyothi Subramanian <<`subramanianj01@gmail.com`>>

Examples

```
KfoldCV(15,3)
KfoldCV(15,15)
```

overall.analysis *Overall statistics and inference*

Description

Produces some statistics for the overall (non-predictive) comparison of the E and C for the same dataset for which the predictive model `pact.fit` was developed.

Usage

```
overall.analysis(p)
```

Arguments

`p`	An object of class `pact`

Details

Statistics for the overall comparison of the E and C is produced for the the data from a randomized clinical trial. The input is an object of class `pact`.

Value

An list with the following components. As a side effect, these are also printed on screen

`family` The response variable type used

`nobs` The sample size

`n.E` Number of subjects getting the new treatment

`n.C` Number of subjects getting the control treatment

`LR` The log-rank statistic for the overal difference in survival between E and C groups (for family="cox")

`LR.pval` The p-value for LR based on the log-rank test (for family="cox")

`RR.E` The response rate for group treated with E (new treatment) (for family="binomial")

`RR.C` The response rate for group treated with C (Control) (for family="binomial")

`RRdiff.pval` The chi-square test based pvalue for the difference in response rates (for family="binomial")

Author(s)

Jyothi Subramanian and Richard Simon
Maintainer: Jyothi Subramanian <`<subramanianj01@gmail.com>`>

Examples

```
data(prostateCancer)
Y <- prostateCancer[,3:4]
Xf <- prostateCancer[,7:8]
Xv <- prostateCancer[,c(5:6,9)]
Treatment <- as.factor(prostateCancer[,2])
p <- pact.fit(Y=Y, Xf=Xf, Xv=Xv, Treatment=Treatment, family="cox", varSelect="none")
overall.analysis(p)

### Binary response
data(EORTC10994)
Y <- as.factor(EORTC10994[,4])
Xv <- EORTC10994[,c(2,5:7)]
Treatment <- as.factor(EORTC10994[,3])
p <- pact.fit(Y=Y,Xv=Xv,Treatment=Treatment,family="binomial",varSelect="none")
overall.analysis(p)
```

pact *Predictive Analysis of Clinical Trials*

Description

The pact package implements a prediction-based approach to the analysis of data from randomized clinical trials (RCT). Based on clincial response and covariate data from a RCT comparing a new experimental treatment E versus a control C, the purpose behind the functions in pact is to develop and internally validate a model that can identify subjects likely to benefit from E rather than C. Currently, ’survival’ and ’binary’ response types are permitted.

Details

Package: pact
Type: Package
Version: 0.5.0
Date: 2016-04-14
Author: Dr. Jyothi Subramanian and Dr. Richard Simon
Maintainer: Jyothi Subramanian <<subramanianj01@gmail.com>>
License: GPL-3

pact.fit fits a predictive model to data from RCT. Currently, ’survival’ and ’binary’ response types are supported. Analysis of high dimensional covariate data is supported. If known and available, a limited number of prognostic covariates can also be specified and fixed to remain in the predictive model. An object of class ’pact’ is returned. print, summary and predict methods are available for objects of class ’pact’. Additionally, the function pact.cv takes as an input the object returned by pact.fit and computes predictive scores for each subject through k-fold cross-validation. Evaluations of the cross-validated predictions are performed by the function eval.pact.cv.

Finally, the function overall.analysis also takes an object of class ’pact’ as input and computes some summary statistics for the comparison of treatments E and C.

Examples

```
### Survival response
set.seed(10)
data(prostateCancer)
Y <- prostateCancer[,3:4]
Xf <- prostateCancer[,7:8]  ## Prognostic covariates fixed to always be in the model
Xv <- prostateCancer[,c(5:6,9)]
Treatment <- as.factor(prostateCancer[,2])
p <- pact.fit(Y=Y,Xf=Xf,Xv=Xv,Treatment=Treatment,family="cox",varSelect="univar")
print(p)
overall.analysis(p)
cv <- pact.cv(p, nfold=5)
eval.pact.cv(cv, method="continuous", plot.score=TRUE, perm.test=FALSE, nperm=100)

### Binary response
```

```
set.seed(10)
data(EORTC10994)
Y <- as.factor(EORTC10994[,4])
## No prognostic covariates (Xf) specified
Xv <- EORTC10994[,c(2,5:7)]
Treatment <- as.factor(EORTC10994[,3])
p <- pact.fit(Y=Y,Xv=Xv,Treatment=Treatment,family="binomial",varSelect="none")
print(p)
overall.analysis(p)
cv <- pact.cv(p, nfold=5)
eval.pact.cv(cv, method="discrete", g=log(1), perm.test=FALSE, nperm=100)

### High dimensional data, survival response
## Not run:
set.seed(10)
data(GSE10846)
Y <- GSE10846[,1:2]
Xv <- GSE10846[,-c(1:3)]
Treatment <- as.factor(GSE10846[,3])
p <- pact.fit(Y=Y,Xv=Xv,Treatment=Treatment,family="cox",varSelect="lasso",penalty.scaling=2)
print(p)
overall.analysis(p)
cv <- pact.cv(p, nfold=5)
eval.pact.cv(cv, method="continuous", plot.score=TRUE, perm.test=FALSE)

## End(Not run)
```

pact.cv *Cross-validation for pact*

Description

Predictive scores using k-fold cross-validation for the model developed in `pact.fit`

Usage

```
pact.cv(p, nfold)
```

Arguments

`p` An object of class 'pact'

`nfold` The number of folds (k) for the k-fold cross-validation. k equal to the sample size would mean a leave-one-out cross-validation

Details

Obtain cross-validated predictive scores for the model developed in `pact.fit`. In each fold of the cross-validation, a model is developed from the observations in the training set using the same variable selection parameters as that used for the model developed in `pact.fit`. The estimated coefficients of the regression model developed using training set are used to make predictions for the left out observations (test set). This is repeated for all the folds. Scores are thus obtained for all the subjects in the dataset. The function `eval.pact.cv` provides various evaluation options for the cross-validated scores.

Value

A list with the following components

`PredScore`	The cross-validated scores for each subject (a vector)
`Y`	The response variable used
`Xf`	The dataframe of fixed prognostic covariates
`Xv`	The dataframe of candidate predictive variables
`Treatment`	The treatment assignment indicator used
`nCovarf`	The number of variables in Xf
`nCovarv`	The number of variables in Xv
`family`	Type of the response variable
`varSelect`	The variable selection method used
`nsig, cvfolds.varSelect, which.lambda, penalty.scaling`	The variable selection parameters used
`call`	The call that produced this output

Author(s)

Jyothi Subramanian and Richard Simon
Maintainer: Jyothi Subramanian <<subramanianj01@gmail.com>>

Examples

```
data(prostateCancer)
Y <- prostateCancer[,3:4]
Xf <- prostateCancer[,7:8]
Xv <- prostateCancer[,c(5:6,9)]
Treatment <- as.factor(prostateCancer[,2])
p <- pact.fit(Y=Y,Xf=Xf,Xv=Xv,Treatment=Treatment,family="cox",varSelect="lasso")
cv <- pact.cv(p, nfold=5)
```

`pact.fit` *Fits a predictive model to the full dataset*

Description

`pact.fit` Fits a predictive model using data on all subjects. Currently supports Cox PH and logistic regression models for 'survival' and 'binary' response types respectively.

Usage

```
pact.fit(Y, Xf = NULL, Xv, Treatment, family = c("binomial", "cox"),
  varSelect = c("none", "univar", "lasso"), nsig = ifelse(varSelect ==
  "univar", ifelse(nCovarv < 10, 3, 10), NA),
  cvfolds.varSelect = ifelse(varSelect == "lasso", 5, NA),
  which.lambda = ifelse(varSelect == "lasso", ifelse(nCovarv < 10, "min",
  "1se"), NA), penalty.scaling = ifelse(varSelect == "lasso", 0.5, NA))
```

Arguments

`Y` Response variable. For `family='binomial'`, Y should be a factor with two levels. For `family='cox'`, Y should be a two-column matrix with columns named 'time' and 'status'. The latter is a binary variable, with '1' indicating death, and '0' indicating right censored.

`Xf` An optional dataframe of the prognostic covariates that are not subject to variable selection and always fixed to remain in the model. Default is NULL (no variable).

`Xv` The main dataframe of covariates that are to be used for predictive model development. Variable selection options affect only the variables in Xv. Xv cannot be NULL.

`Treatment` The treatment assignment indicator. A factor with two levels. '0' indicating control (C) and '1' indicating experimental (E) treatment.

`family` Type of the response variable. See above. Possible values are 'binomial' or 'cox'.

`varSelect` The variable selection method. Possible values are "none","univar" or "lasso".

`nsig` The number of covariates to use in the model for varSelect="univar". Defaults to 3 if the number of candidate covariates is less than 10, else defaults to 10.

`cvfolds.varSelect`
The number of folds in the internal cross-validation loop for variable selection with varSelect="lasso". Default is 5.

`which.lambda` Used with variable selection with varSelect="lasso". Defaults to "min" if the number of candidate covariates is less than 10, else defaults to "1se". See Details.

`penalty.scaling`
Ratio of shrinkage applied for main coefficients to shrinkage applied for interaction coefficients. Used with varSelect="lasso". Default is 0.5. See Details.

Details

A Cox proportional hazards (PH) or a logistic regression model is developed for data with survival and binary response respectively. Data from subjects in both 'experimental' (E) and 'control' (C) groups from a RCT is used for model development. Main effect of treatment, main effect of prognostic covariates, main effects and treatment by covariate interaction terms of candidate predictive covariates are considered in the model. Methods for variable selection can be optionally specified by the user for candidate predictive covariates (useful for high-dimensional covariates). Current options for variable selection include "univar" and "lasso". In the case of "univar", the number of predictive covariates (nsig) to be included in the model is specified by the user. A univariate selection procedure is applied to identify covariates that have the lowest treatment*covariate interaction p-values. The predictive model is then developed using the main effect of treatment, main effects of prognostic covariates, main effects of the nsig predictive covariates and treatment by covariate interaction terms for nsig predictive covariates.

In the case of "lasso", an internal cross-validation loop is used to find the penalty value that minimizes the cross-validated error. The user can choose either the value of the penalty 'lambda' as the penalty that minimizes the cross-validated error ("lambda.min") or the largest penalty for which the cross-validated error is within 1 standard error of the minimum ("lambda.1se"). Also, in the case of "lasso", differential shrinkage can be specified for main effect and interaction effect predictive coefficients by specifying a value for the ratio of shrinkage for main coefficients to shrinkage for interaction coefficients. Internally, 'lambda' is scaled using this ratio to allow for the differential shrinkage of main and interaction coefficients.The penalty factors affect only variables in Xv and not Xf.

Value

An object of class 'pact' which is a list with the following components:

`reg`	The fitted regression model
`family`	Type of the response variable
`Y`	The response variable used
`Xf`	The dataframe of prognostic covariates
`Xv`	The dataframe of candidate predictive variables
`Treatment`	The treatment assignment indicator used
`nCovarf`	The number of variables in Xf
`nCovarv`	The number of variables in Xv
`varSelect`	The variable selection method used
`nsig, cvfolds.varSelect, which.lambda, penalty.scaling`	The variable selection parameters used
`call`	The call that produced the return object

Author(s)

Jyothi Subramanian and Richard Simon
Maintainer: Jyothi Subramanian <`<subramanianj01@gmail.com>`>

Examples

```
data(prostateCancer)
Y <- prostateCancer[,3:4]
Xf <- prostateCancer[,7:8]
Xv <- prostateCancer[,c(5:6,9)]
Treatment <- as.factor(prostateCancer[,2])
pact.fit(Y=Y, Xf=Xf, Xv=Xv, Treatment=Treatment, family="cox",varSelect="univar")
```

`predict.pact` *Predictions from a predictive model fit*

Description

Predicts the scores for new subjects from a previously developed object of class 'pact'

Usage

```
## S3 method for class 'pact'
predict(object, newxv, ...)
```

Arguments

`object`	The object returned from `pact.fit`
`newxv`	The dataframe Xv of covariates for the new subjects for whom predictions are to be made
`...`	Other arguments to 'predict'

Details

Returns the scores for new subjects from an object of class 'pact', given their covariate values and treatment assignment.

Value

A numeric vector containing the predicted scores for the new subjects from the fitted model is returned

Author(s)

Jyothi Subramanian and Richard Simon
Maintainer: Jyothi Subramanian <`<subramanianj01@gmail.com>`>

Examples

```
### Survival response
data(prostateCancer)
Y <- prostateCancer[1:400,3:4]
Xf <- prostateCancer[1:400,7:8]
Xv <- prostateCancer[1:400,c(5:6,9)]
Treatment <- as.factor(prostateCancer[1:400,2])
p <- pact.fit(Y=Y, Xf=Xf, Xv=Xv, Treatment=Treatment, family="cox", varSelect="univar")

newxv <- prostateCancer[401:410,c(5:6,9)]
predict(p, newxv)

### Binary response
data(EORTC10994)
Y <- as.factor(EORTC10994[1:120,4])
Xv <- EORTC10994[1:120,c(2,5:7)]
Treatment <- as.factor(EORTC10994[1:120,3])
p <- pact.fit(Y=Y,Xv=Xv,Treatment=Treatment,family="binomial",varSelect="none")

newxv <- EORTC10994[121:125,c(2,5:7)]
predict(p, newxv)
```

---

print.eval.cv *Print an object of class 'eval.cv'*

Description

print method for objects of class 'eval.cv'

Usage

```
## S3 method for class 'eval.cv'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	The object returned from 'eval.pact.cv'
digits	significant digits in the print
...	Additional print arguments

Details

The call that produced the object is printed, followed by the evaluation statistics. The p-values are printed if permutation testing was asked for.

Value

The statistics comparing treatments E and C is printed. The printed statistics differs according to whether method was 'discrete' or 'continuous'. p-values are printed if perm.test=TRUE

Author(s)

Jyothi Subramanian and Richard Simon
Maintainer: Jyothi Subramanian <<subramanianj01@gmail.com>>

Examples

```
data(prostateCancer)
Y <- prostateCancer[,3:4]
Xf <- prostateCancer[,7:8]
Xv <- prostateCancer[,c(5:6,9)]
Treatment <- as.factor(prostateCancer[,2])
p <- pact.fit(Y=Y, Xf=Xf, Xv=Xv, Treatment=Treatment, family="cox", varSelect="lasso")
cv <- pact.cv(p, nfold=5)
eval <- eval.pact.cv(cv, method="discrete", g=log(0.80))
eval
```

---

print.pact *Print an object of class 'pact'*

Description

print method for objects of class 'pact'

Usage

```
## S3 method for class 'pact'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	The object returned from 'pact.fit'
digits	significant digits in the print
...	Additional print arguments

Details

The call that produced the object is printed, followed by the classification function from pact.fit for calculating the predictive scores for new subjects

Value

The classification function is printed

Author(s)

Jyothi Subramanian and Richard Simon
Maintainer: Jyothi Subramanian <<subramanianj01@gmail.com>>

Examples

```
data(prostateCancer)
Y <- prostateCancer[,3:4]
Xf <- prostateCancer[,7:8]
Xv <- prostateCancer[,c(5:6,9)]
Treatment <- as.factor(prostateCancer[,2])
p <- pact.fit(Y=Y, Xf=Xf, Xv=Xv, Treatment=Treatment, family="cox", varSelect="lasso")
print(p)
```

`prostateCancer` *Prostate cancer dataset*

Description

A dataset containing survival information for 485 subjects with prostate cancer. See 'Details' for the variables in this dataset.

Format

A data frame with 485 rows and 9 variables

Details

- ID. Subject identifier
- Treatment. Treatment received. '0' for control and '1' for new
- time. Survival time in months
- status. Censoring status. '0' for censored, '1' for died
- age. Age (in years) at diagnosis
- pf. Performance status (Normal Activity or Limited Activity)
- sz. Size of the primary tumor (cm2)
- sg. Index of a combination of tumor stage and histologic grade
- ap. Serum phosphatic acid phosphatase levels

summary.pact *Summarize a predictive model fit*

Description

summary method for objects of class ’pact’

Usage

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

Arguments

`object`	The object returned from `'pact.fit'`
`...`	Additional arguments for ’summary’

Details

Returns all coefficient estimates from the regression model of the ’pact’ object

Value

All the coefficient estimates from the regression model fitted by `pact.fit`

Author(s)

Jyothi Subramanian and Richard Simon
Maintainer: Jyothi Subramanian <`<subramanianj01@gmail.com>`>

Examples

```
data(prostateCancer)
Y <- prostateCancer[,3:4]
Xf <- prostateCancer[,7:8]
Xv <- prostateCancer[,c(5:6,9)]
Treatment <- as.factor(prostateCancer[,2])
p <- pact.fit(Y=Y,Xf=Xf,Xv=Xv,Treatment=Treatment,family="cox",varSelect="none")
summary(p)
```

Index