Package ‘survivalPLANN’

July 2, 2025

Type Package

Title Neural Networks to Predict Survival

Version 0.4

Depends R (>= 4.0.0), survival, nnet

Imports methods, stats, RISCA

Description Several functions and S3 methods to predict survival by using neural networks. We implemented Partial Logistic Artificial Neural Networks (PLANN) as proposed by Biganzoli et al. (1998) <`https://pubmed.ncbi.nlm.nih.gov/9618776`>.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

BugReports `https://github.com/chupverse/survivalPLANN/issues`

Author Yohann Foucher [aut, cre] (ORCID: <`https://orcid.org/0000-0003-0330-7457`>),
Thomas Ollard [aut] (ORCID: <`https://orcid.org/0009-0001-5122-9214`>)

Maintainer Yohann Foucher <`yohann.foucher@univ-poitiers.fr`>

Repository CRAN

Date/Publication 2025-07-02 09:00:02 UTC

Contents

`cvPLANN` *Cross-validation method for a Neural Netwotk Model with the PLANN Method.*

Description

Cross-validation method for the hyper-parameters for an object of class `sPLANN`.

Usage

```
cvPLANN(formula, pro.time=NULL, data, cv=10, inter=1, size=8, decay=0.01,
                    maxit=1000, MaxNWts=10000)
```

Arguments

`formula`	The `formula` object.
`pro.time`	The prognostic time at which the metric is evaluated. If `NULL`, the median of the times is chosen.
`data`	A data frame in which to look for the variables included in the `formula`.
`cv`	The number of splits for cross-validation. The default value is 10.
`inter`	A numeric value representing the length of the intervals.
`size`	A numeric value for the number of units in the hidden layer. Default is set to 8
`decay`	A numeric value for the parameter for weight decay. Default is set to 0.01
`maxit`	A numeric value for the maximum number of iterations. Default is set to 1000.
`MaxNWts`	The maximum allowable number of weights. There is no intrinsic limit in the code, but increasing MaxNWts will probably allow fits that are very slow and time-consuming. Default is set to 10000

Value

`optimal`	A list giving the optimal value for each parameter.
`results`	A data frame listing the parameters combinaison and their metrics values.

See Also

`sPLANN`

Examples

```
data(dataK) # the database with the observed sample

tune.sPLANN <- cvPLANN(Surv(time, event)~ stade + delay, data=dataK, cv=3,
                    inter=365.241, size=c(2, 4), decay=c(0.01))

tune.sPLANN$optimal
```

`dataK` *Observed Mortality Data of Patients with Cancer*

Description

An data frame with a simulated sample of patients with a cancer and their follow-up.

Usage

```
data(dataK)
```

Format

The format is "data.frame". The names of the columns are:

`sex`	A numeric vector equals 1 for male and 2 for male.
`age`	A numeric vector with the age of the patient at the time of cancer diagnosis (baseline).
`year`	A numeric vector with the date of diagnosis (in date format, i.e., the number of days since 1Jan60).
`stade`	A numeric vector with the disease stage: 1 for advanced cancer and 0 otherwise. according to the time. Only the dimension related to `sex` equals 1.
`delay`	A numeric vector indicating early diagnosis: 1 for delayed diagnosis and 0 otherwise.
`biomarker`	A numeric vector with a biomarker associated with cancer-specific mortality.
`sexchara`	A character vector with the patient gender: "male" or "female".
`event`	A numeric vector equals 1 for death and 0 for censoring.
`time`	A numeric vector with the follow-up time in days since the cancer diagnostic.

Details

The data frame was obtained by simulations. The French mortality tables were used for the expected mortality and a proportional hazard model with an Exponential distribution for the baseline hazard.

Examples

```
data(dataK)

# Plotting the observed survival by using the Kaplan-Meier estimator

plot(survfit(Surv(time/365.241, event) ~ 1, data = dataK),
     ylab="Patient survival", xlab="Post-diagnostic time in years")
```

`fr.ratetable` *Expected Mortality Rates of the General French Population*

Description

An object of class rate table for the expected mortality of the French population. It is an array with three dimensions: age, year, and sex.

Usage

```
data(fr.ratetable)
```

Format

The format is "ratetable". The attributes are:

`dim`	A numeric vector with the length of each dimension.
`dimnames`	A character vector with the names of each variable of the three dimensions.
`dimid`	A character vector with the identification of the dimensions: `age`, `year` and `sex`.
`factor`	A numeric vector of indicators equals to 1 if the corresponding dimension does not vary according to the time. Only the dimension related to `sex` equals 1.
`cutpoints`	A list of the thresholds to identify the changes in the mortality rates according to the time-dependent dimensions (NULL for `sex`).
`class`	The class of the object: `ratetable`.

Details

The organization of a rate table object is described in detail by Therneau (1999) and Pohar (2006). The original data and updates can be downloaded from the Human Life-Table Database (HMD, The Human Mortality Database).

Source

URL: www.mortality.org

References

Therneau and Offord. Expected Survival Based on Hazard Rates (Update), Technical Report, Section of Biostatistics, Mayo Clinic 63, 1999.

Pohar and Stare. Relative survival analysis in R. Computer methods and programs in biomedicine, 81: 272-278, 2006.

Examples

```
data(fr.ratetable)

is.ratetable(fr.ratetable)
```

`metrics` *Metrics to Evaluate the Prognostic Capacities*

Description

Compute several metrics to evaluate the prognostic capacities with time-to-event data.

Usage

```
metrics(formula, data, prediction.matrix, prediction.times, metric,
pro.time=NULL, ROC.precision=seq(.01, .99, by=.01))
```

Arguments

`formula`	A formula object, with a survival object `Surv(time,event)` on the left of a ~ operator, and only "`~1`" on the right side.
`data`	A data frame for in which to look for the variables related to the status of the follow-up time (`times`) and the event (`failures`).
`prediction.matrix`	A matrix or data.frame object of numeric values of dimension n*p, with n being the number of suject and p being the number of prognostic times.
`prediction.times`	A vector of numeric values with the times of the `predictions` (same length than the number of columns of `prediction.matrix`).
`metric`	The metric to compute. See details.
`pro.time`	This optional value of prognostic time represents the maximum delay for which the capacity of the variable is evaluated. The same unit as the time variable in the formula. Not used for the following metrics: "loglik", "ibs", "bll", and "ibll". Default value is the time at which half of the subjects are still at risk.
`ROC.precision`	An optional argument with the percentiles (between 0 and 1) of the prognostic variable used for computing each point of the time dependent ROC curve. Only used when `metric="auc"`. 0 (min) and 1 (max) are not allowed. By default, the precision is `seq(.01,.99,.01)`.

Details

The following metrics can be used: "bs" for the Brier score at the prognostic time `pro.time`, "ci" for the concordance index at the prognostic time `pro.time`, "loglik" for the log-likelihood, "ibs" for the integrated Brier score up to the last observed time of event, "ibll" for the integrated binomial log-likelihood up to the last observed time of event, "bll" for the binomial log-likelihood, "ribs" for the restricted integrated Brier score up to the prognostic time `pro.time`, "ribll" for the restricted integrated binomial log-likelihood up to the last observed time of event, "bll" for the binomial log-likelihood, "auc" for the area under the time-dependent ROC curve up to the prognostic time `pro.time`.

Value

A numeric value with the metric estimation.

Examples

```
data(dataK)

# The estimation of the model
 model <- sPLANN(formula = Surv(time, event)~stade, data = dataK, inter = 365.241)

#Predictions of the model

pred <- predict(model)

# The apparent AUC at 10-year post-transplantation
metrics(formula = Surv(time, event)~1, data=dataK,
   prediction.matrix = pred$predictions, prediction.times=pred$times, metric
  ="auc", pro.time=10*365.241)

# The integrated Brier score up to 10 years post-transplanation
metrics(formula = Surv(time, event)~1, data=dataK,
  prediction.matrix = pred$predictions, prediction.times=pred$times, metric
  ="ribs", pro.time=10*365.241)
```

`plot.sPLANN` *Calibration Plot for a Survival Neural Network Using the PLANN Method.*

Description

A calibration plot of an object of the class `sPLANN`.

Usage

```
## S3 method for class 'sPLANN'
plot(x, n.groups=5, pro.time=NULL,
newdata=NULL, ...)
```

Arguments

`x`	An object returned by the function `sPLANN`.
`n.groups`	A numeric value with the number of groups by their class probabilities. The default is 5.
`pro.time`	The prognostic time at which the calibration plot of the survival probabilities.
`newdata`	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is `NULL`: the calibration plot is performed from the same subjects of the training sample.
`...`	Additional arguments affecting the plot.

Details

The plot represents the survival obtained by the Kaplan-Meier estimator and the related 95% confidence intervals, against the mean of the predictive values for individuals stratified into groups of the same size according to the percentiles. The identity line is usually included for reference.

Value

No return value for this S3 method.

See Also

`plot.default`, `sPLANN`.

Examples

```
data(dataK) # the database with the observed sample
data(fr.ratetable) # the table with the expected mortality rates

# The multivariate PH regression with a Weibull distribution

model <- sPLANN(Surv(time, event) ~ biomarker + sex + stade + delay
                , data = dataK, inter = 365.241)

# the calibration plots for predictions at 2, 4, 6 and 8 years

par(mfrow=c(2, 2))

plot(model, n.groups=2, pro.time=2*365.24,
  main="Predction at 2 years")

plot(model, n.groups=2, pro.time=4*365.24,
  main="Predction at 4 years")

plot(model, n.groups=2, pro.time=6*365.24,
  main="Predction at 6 years")

plot(model, n.groups=2, pro.time=8*365.24,
  main="Predction at 8 years")
```

`predict.sPLANN` *Predict Survival From a Neural Network Based on the PLANN Method*

Description

This function produces survival prediction from a neural network based on the PLANN method.

Usage

```
## S3 method for class 'sPLANN'
predict(object, newdata = NULL, newtimes = NULL, ...)
```

Arguments

`object`	The result of the `sPLANN` function.
`newdata`	An optional data frame comprising of new examples to be predicted. If `NULL`, the data frame used is the one used for training in the `sPLANN` function.
`newtimes`	A optional numeric vector comprising of times to get survival estimations. If `NULL`, the times are the intervals used in the `sPLANN` function.
`...`	Further arguments passed to or from other methods.

Value

`times`	The times used for the predicitions.
`predictions`	A data frame comprising of the survival predictions from the neural network.

References

Biganzoli E, Boracchi P, Mariani L, and et al. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. Stat Med, 17:1169-86, 1998.

Examples

```
data(dataK)

splann <- sPLANN(Surv(time, event) ~ sex + stade + delay, data=dataK, inter=365.241,
                          size=32, decay=0.01, maxit=200, MaxNWts=10000)

dnew <- data.frame(sex=c(1,2), delay=c(0,0), stade=c(0,0))

pred <- predict(splann, newdata = dnew)

# Predictions for a men or a women with no delay at the diagnostic of non-agressive cancer

plot(pred$times, c(pred$predictions[1,]), ylab="Patient survival",
  xlab="Post-diagnosis time in years", type="l")
lines(pred$times, c(pred$predictions[2,]), type="l", col=2)
```

`print.sPLANN` *Print Method for a Neural Network Based on the PLANN Method*

Description

This function prints a description of the neural network using the PLANN method.

Usage

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

Arguments

`x`	The result of a sPLANN fit.
`...`	Further arguments passed to or from other methods.

Value

No return value for this S3 method.

References

Biganzoli E, Boracchi P, Mariani L, and et al. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. Stat Med, 17:1169-86, 1998.

Examples

```
data(dataK)

splann <- sPLANN(Surv(time, event) ~ sex + stade + delay, data=dataK, inter=365.241,
                        size=32, decay=0.01, maxit=100, MaxNWts=10000)

print(splann)
```

`sPLANN` *Survival Neural Network by Using the PLANN Method*

Description

This function allows to compute a neural network using the PLANN method.

Usage

```
sPLANN(formula, data, pro.time=NULL, inter, size = 32, decay = 0.01,
                          maxit = 100, MaxNWts = 10000, trace = FALSE, ...)
```

Arguments

`formula`	a formula object, with the response on the left of a ~ operator, and the terms on the right. The response must be a survival object as returned by the `Surv` function.
`data`	A data frame in which to look for the variables included in the `formula`.
`pro.time`	A numeric value with the maximum prognostic. If `NULL`, the maximum prognostic time is the last time for wich an event is observed.
`inter`	A numeric value representing the length of the intervals.
`size`	A numeric value for the number of units in the hidden layer. Default is set to 32
`decay`	A numeric value for the parameter for weight decay. Default is set to 0.01
`maxit`	A numeric value for the maximum number of iterations. Default is set to 100.

`MaxNWts`	The maximum allowable number of weights. There is no intrinsic limit in the code, but increasing MaxNWts will probably allow fits that are very slow and time-consuming. Default is set to 10000
`trace`	A logical argument for tracing optimization. Default is set to FALSE.
`...`	Further arguments passed to or from other methods.

Details

The `nnet` loss function parameter is fixed to `entropy` as Biganzoli's PLANN method minimize the cross-entropy loss function which correspondS to maximizing the loglikelihood.

Value

`formula`	The `formula` object.
`fitsurvivalnet`	The fitted `nnet` object.
`data`	The data frame given with the computed `Intervals` column added.
`data_dup`	The base data frame with the duplicated rows based on the `Intervals` column.
`call`	The function call for the creation of the `nnet` object.
`inter`	The interval length.
`size`	The number of units in the hidden layer.
`decay`	The value of the parameter for weight decay.
`maxit`	The value of the parameter for maxit.
`MaxNWts`	The value of the parameter for MaxNWts.
`coefnames`	The name(s) of the covariate(s) used in the model.
`y`	The values of `time` and `event`.
`x`	The model covariates values.
`intervals`	The intervals limits values.
`missing`	The rows with a missing value in the data base.

References

Biganzoli E, Boracchi P, Mariani L, and et al. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. Stat Med, 17:1169-86, 1998.

Examples

```
data(dataK)

splann <- sPLANN(Surv(time, event) ~ sex + stade + delay, data=dataK, inter=365,
                        size=32, decay=0.01, maxit=100, MaxNWts=10000)

print(splann)
```

Index