Package ‘mipred’

July 12, 2019

Type Package
Title Prediction using Multiple Imputation
Version 0.0.1
Maintainer Bart J. A. Mertens <b.mertens@lumc.nl>
Description Calibration of generalized linear models and Cox regression models for prediction using multiple imputation to account for missing values in the predictors as described in the paper by “Mertens, Banzato and de Wreede” (2018) <arXiv:1810.05099>. The methodology and calculations described in this paper are fully implemented in this package. The vignette describes all data analytic steps which allow users to replicate results using the package functions on the data analyzed in the paper or on their own data.

Imputations are generated using the package ‘mice’ without using the outcomes of observations for which the predictions are generated. Two options are provided to generate predictions. The first is prediction-averaging of predictions calibrated from single models fitted on single imputed datasets within a set of multiple imputations. The second is application of the Rubin’s rules pooled model.

For both implementations, unobserved values in the predictor data of new observations for which the predictions are derived are automatically imputed. The package contains two basic functions.

The first, mipred() generates predictions of outcome on new observations. The second, mipred.cv() generates cross-validated predictions with the methodology on existing data for which outcomes have already been observed. The present version is still in development and should support continuous, binary and counting outcomes, but we have only thoroughly checked performance for binary outcome logistic regression modeling. We will include the Cox regression extension later.

URL https://github.com/BartJAMertens/mipred,
https://arxiv.org/abs/1810.05099,
https://www.researchgate.net/project/Prediction-calibration-using-multiple-imputations-to-account-for-missing-predictor-values
Description

Expit function converting odds to probability

Usage

.expit(x)

Arguments

x Probability vector
Generalized linear model prediction using multiple imputation - prediction-averaging method

Description

Generalized linear model prediction using multiple imputation - prediction-averaging method

Usage

.glm.mipred.cmb1(formula, family, dataset, newdata, nimp, folds, miop)

Arguments

- **formula**: Formula used by fitting and prediction method
- **family**: Error distribution also determining the link function used
- **dataset**: A data frame containing calibration data
- **newdata**: A dataframe containing observations to be predicted
- **nimp**: Number of imputations for each observation
- **folds**: Number of folds defined in newdata
- **miop**: Mice options

Value

A list containing predictions.

- **pred**: Matrix of predictions on the scale of the response variable of dimension \( m \) by \( n_{imp} \).
- **linpred**: Matrix of predictions on the scale of the linear predictor of dimension \( m \) by \( n_{imp} \).

Note

This is an internal 'mipred' function and not intended to be called directly.
Description

Cross-validation of generalized linear model prediction using multiple imputation - prediction-averaging method

Usage

`glm.mipred.cmb1cv(formula, family, dataset, nimp, folds, miop)`

Arguments

- `formula`: Formula used by fitting and prediction method
- `family`: Error distribution also determining the link function used
- `dataset`: A data frame containing calibration data
- `nimp`: Number of imputations for each observation
- `folds`: Number of folds defined in newdata
- `miop`: Mice options

Value

A list containing predictions.

- `pred`: Matrix of predictions on the scale of the response variable of dimension `m` by `nimp`.
- `linpred`: Matrix of predictions on the scale of the linear predictor of dimension `m` by `nimp`.

Note

This is an internal `mipred` function and not intended to be called directly

---

Description

Generalized linear model prediction using multiple imputation - Rubin’s rule coefficient-averaging method

Usage

`glm.mipred.cmb2(formula, family, dataset, newdata, nimp, folds, miop)`
Arguments

- `formula`: Formula used by fitting and prediction method
- `family`: Error distribution also determining the link function used
- `dataset`: A data frame containing calibration data
- `newdata`: A dataframe containing observations to be predicted
- `nimp`: Number of imputations for each observation
- `folds`: Number of folds defined in newdata
- `miop`: Mice options

Value

A list containing predictions.

- `pred`: Matrix of predictions on the scale of the response variable of dimension `m` by `nimp`.
- `linpred`: Matrix of predictions on the scale of the linear predictor of dimension `m` by `nimp`.

Note

This is an internal 'mipred' function and not intended to be called directly

Description

Cross-validation of generalized linear model prediction using multiple imputation - Rubin's rule coefficient-averaging method

Usage

`.glm.mipred.cmbH2.cv(formula, family, dataset, nimp, folds, miop)`
.impute

General imputation routine for mipred

## Description

General imputation routine for mipred

## Usage

```r
.impute(combdat, miop, nimp, seed)
```

## Arguments

- `combdat`: Dataset to be imputed
- `miop`: Mice options list
- `nimp`: Number of imputations
- `seed`: Single numerical seed value

## Value

A 'mice' object containing imputations

## Note

This is an internal 'mipred' function and not intended to be called directly.
**Description**

A dataset containing survival outcome and predictors on 694 patients who received hematopoietic stem cell transplant.

**Usage**

c11

**Format**

A data frame with 694 rows and 11 variables. Each row describes the data from a single patient. The below described variables are included in the data file. Missing observations are present in the variables performance status (9%), remission status (6%) and cytogenic abnormality (25%).

- **id** record identification number
- **age10** age at transplantation
- **perfstat** performance status indicated by the Karnofsky Index (four categories)
- **remstat** remission status at transplantation (three categories)
- **cyto** cytogenetic abnormalities (four categories)
- **asct** previous autologous transplantation (two categories)
- **donor** donor type (three categories)
- **sex_match** patient-donor sex match (four categories)
- **cond** conditioning regimen (three categories)
- **srv5y** overall survival (OS) up to five years after first allogeneic stem cell transplantation
- **srv5y_s** censoring indicator (0=alive at end follow-up, 1=dead)

**Source**

European Society for Blood and Marrow Transplantation (EBMT). [https://www.ebmt.org](https://www.ebmt.org)

**References**

Prediction using multiple imputation

Description

Calculates predictions from generalized linear models when multiple imputations are used to account for missing values in predictor data.

Usage

mipred(formula, family, data, newdata, nimp, folds = NULL, method = "averaging", mice.options = NULL)

Arguments

- **formula**: A formula object providing a symbolic description of the prediction model to be fitted.
- **family**: Specification of an appropriate error distribution and link function.
- **data**: A data.frame containing calibration data on \( n \) samples. Variables declared in \texttt{formula} must be found in \texttt{data}.
- **newdata**: A data.frame containing the predictors for observations to be predicted on \( m \) samples. This must have the same structure and variables as \texttt{data}, except for the outcome variable which is ignored in the construction of the predictions and can therefore be excluded from the object.
- **nimp**: Number of imputations used in the prediction of each observation.
- **folds**: Number of fold-partitions defined within \texttt{newdata}. An integer from 1 to \( m \). Defaults to \texttt{NULL} which internally sets \texttt{folds}=\( m \), which puts each observation in \texttt{newdata} into its own singleton fold. The minimum value \texttt{folds}=1 would predict the entire set \texttt{newdata} in a single step without partitioning.
- **method**: Imputation combination method. This defaults to "averaging" for the prediction-averaging approach. The alternative "rubin" applies the Rubin’s rules pooled model.
- **mice.options**: Optional list containing arguments to be supplied to \texttt{mice}. Refer to the \texttt{mice} documentation for details. The following options may be specified: \texttt{method}, \texttt{predictorMatrix}, \texttt{blocks}, \texttt{visitSequence}, \texttt{formulas}, \texttt{blots}, \texttt{post}, \texttt{defaultMethod}, \texttt{maxit}, \texttt{printFlag}, \texttt{seed}, \texttt{data.init}. Please refer to the \texttt{mice} documentation for the description of these options. To set the number of imputations \texttt{nimp} should be used. \texttt{seed} may be specified as a numeric vector of length \texttt{nimp*length(folds)} when \texttt{method} is set to \texttt{averaging} and of length \texttt{length(folds)} when \texttt{method} is set to \texttt{rubin}. Setting \texttt{seed} to a vector will cause each call to \texttt{mice} to use the next seed value in the vector. Setting the \texttt{seed} to a single numeric value will cause all instances of \texttt{mice} to use that same seed value. If you specify a \texttt{seed} vector of insufficient length then the values will be recycled. The required length is \texttt{length(folds)*nimp} for the \texttt{averaging} approach and \texttt{length(folds)} for the \texttt{rubin} approach. The \texttt{defaultMethod} is set to \texttt{c("pmm", "logreg", "polyreg", "polr")}. 
by default. The default setting for printFlag is FALSE. The default for maxit is 50. All other options are set to NULL by default.

**Value**

A list consisting of 3 components, of which the first is the Call and the last two are matrices of predictions as follows.

- `pred` Matrix of predictions on the scale of the response variable of dimension \(m\) by \(n_{imp}\).
- `linpred` Matrix of predictions on the scale of the linear predictor of dimension \(m\) by \(n_{imp}\).

**Author(s)**

Bart J A Mertens, <b.mertens@lumc.nl>

**References**

https://arxiv.org/abs/1810.05099

**See Also**

mice

**Examples**

```r
# Generate a copy of the cll data and construct binary outcome from survival information
cll_bin<-cll
cll_bin$srv5y_s[cll_bin$srv5y>12] <- 0  # Apply administrative censorship at t=12 months
cll_bin$srv5y[cll_bin$srv5y>12] <- 12
cll_bin$Status[cll_bin$srv5y_s==1] <- 1  # Define the new binary "Status" outcome variable
cll_bin$Status[cll_bin$srv5y_s==0] <- 0  # As numeric -> 1:Dead, 0:Alive
cll_bin$Censor <- NULL  # Remove survival outcomes
cll_bin$srv5y <- NULL
cll_bin$srv5y_s <- NULL

# Predict observations 501 to 504 using the first 100 records to calibrate predictors
# Remove the identification variable before prediction calibration and imputation.
# Remove outcome for new observations
# Apply prediction-averaging using 5 imputations, set mice option maxit=5.
# Note these settings are only for illustration and should be set to higher values for practical use, particularly for nimp.
output<-mipred(Status ~ age10*cyto, family=binomial, data=cll_bin[1:100,-1],
                newdata=cll_bin[501:504,c(-1,-10)], nimp=5, mice.options=list(maxit=5))
```
mipred.cv  Cross-validation prediction using multiple imputation

Description

Calculates cross-validated predictions based on within-sample assessment and calibration using
generalized linear models with multiple imputations to account for missing values in predictor data.

Usage

mipred.cv(formula, family, data, nimp, folds = NULL,
    method = "averaging", mice.options = NULL)

Arguments

formula  A formula object providing a symbolic description of the prediction model to be
    fitted.
family   Specification of an appropriate error distribution and link function.
data     A data.frame containing calibration data on n samples. Variables declared in
    formula must be found in data.
nimp     Number of imputations used in the prediction of each observation.
folds    Number of fold-partitions defined within data used in cross-validation. An inte-
    ger from 2 to n. Defaults to NULL which internally sets folds=n, which puts
    each observation in data into its own singleton fold for leave-one-out cross-
    validation.
method   Imputation combination method. This defaults to "averaging" for the prediction-
    averaging approach. The alternative "rubin" applies the Rubin's rules pooled
    model.
mice.options Optional list containing arguments to be supplied to mice. Refer to the mice
    documentation for details. The following options may be specified: method,
    predictorMatrix, blocks, visitSequence, formulas, blots, post, defaultMethod,
    maxit, printFlag, seed, data.init. Please refer to the mice documenta-
    tion for the description of these options. To set the number of imputations
    nimp should be used. seed may be specified as a numeric vector of length
    nimp*folds when method is set to averaging and of length folds when method
    is set to rubin. Setting seed to a vector will cause each next call to mice to use
    the next seed value in the vector. Setting the seed to a single numeric value will
    cause all instances of mice to use that same seed value. If you specify a seed vec-
    tor of insufficient length then the values will be recycled. The required length
    is folds*nimp for the averaging approach and length folds for the rubin ap-
    proach. The defaultMethod is set to c("pmm", "logreg", "polyreg", "polr")
    by default. The default setting for printFlag is FALSE. The default for maxit
    is 50. All other options are set to NULL by default.
Value

A list consisting of 3 components, of which the first is the Call and the last two are matrices of predictions as follows.

pred Matrix of predictions on the scale of the response variable of dimension n by nimp.
linpred Matrix of predictions on the scale of the linear predictor of dimension n by nimp.

Author(s)

Bart J A Mertens, <b.mertens@lumc.nl>

References

https://arxiv.org/abs/1810.05099

See Also

mice

Examples

# Generate a copy of the cll data and construct binary outcome from survival information
cll_bin<-cll
cll_bin$srv5y_s[cll_bin$srv5y>12] <- 0 # Apply administrative censorship at t=12 months
cll_bin$srv5y[cll_bin$srv5y>12] <- 12
cll_bin$Status[cll_bin$srv5y_s==1] <- 1 # Define the new binary "Status" outcome variable
cll_bin$Status[cll_bin$srv5y_s==0] <- 0 # As numeric -> 1:Dead, 0:Alive
cll_bin$Censor <- NULL # Remove survival outcomes
cll_bin$srv5y <- NULL
cll_bin$srv5y_s <- NULL

# Cross-validate prediction using logistic regression in the first 100 samples
# Apply prediction-averaging using 5 imputations, 5 folds and maxit=5.
# Note these settings are only for illustration and should be set to higher values for
# practical use, particularly for nimp.
output<-mipred.cv(Status ~ age+cyto, family=binomial, data=cll_bin[1:100,-1],
nimp=5, folds=5, mice.options=list(maxit=5))
Index

*Topic datasets
  cll, 7
  .expit, 2
  .glm.mipred.cmb1, 3
  .glm.mipred.cmb1.cv, 4
  .glm.mipred.cmb2, 4
  .glm.mipred.cmb2.cv, 5
  .impute, 6

cll, 7

mice, 9, 11
mipred, 8
mipred.cv, 10