Package 'reinforcedPred'

October 31, 2018

Type Package

Title Reinforced Risk Prediction with Budget Constraint

Version 0.1.1

Author Yinghao Pan [aut, cre],
Yingqi Zhao [aut],
Eric Laber [aut]

Maintainer Yinghao Pan <ypan8@uncc.edu>

Description Traditional risk prediction only utilizes baseline factors known to be associated with the disease. Given that longitudinal information are routinely measured and documented for patients, it is worthwhile to make full use of these data. The available longitudinal biomarker data will likely improve prediction. However, repeated biomarker collection could be costly and inconvenient, and risk prediction for patients at a later time could delay necessary medical decisions. Thus, there is a trade-off between high quality prediction and cost. This package implements a cost-effective statistical procedure that recursively incorporates comprehensive longitudinal information into the risk prediction model, taking into account the cost of delaying the decision to a follow-up time when more information is available. The statistical methods are described in the following paper: Pan, Y., Laber, E., Smith, M., Zhao, Y. (2018). Reinforced risk prediction with budget constraint: application to electronic health records data. Manuscript submitted for publication.

Depends R (>= 3.5.0)

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports glmnet (>= 2.0-16), MASS (>= 7.3-50), refund (>= 0.1-17), stats

RoxygenNote 6.1.0

URL https://github.com/Yinghao-Pan/reinforcedPred

BugReports https://github.com/Yinghao-Pan/reinforcedPred/issues

NeedsCompilation no
Repository CRAN
Date/Publication 2018-10-31 10:20:11 UTC

R topics documented:

`modelFit` *Model fit for the training set*

Description

`modelFit` outputs the FPCA (functional principal component analysis) decomposition and the parameter estimates of the functional generalized linear model at each time grid.

Usage

```
modelFit(Y, X, Z, startT, link, pve, nbasis, weight)
```

Arguments

`Y`	The outcome variable, vector of length n, taking values in $1, 0, NA$, where 1 = disease, 0 = not, NA = missing.
`X`	Observed longitudinal biomarker, matrix of n by $nTotal$, where $nTotal$ denotes the total number of time grids. Missing values are denoted by NA.
`Z`	Other baseline covariates.
`startT`	Time of the first prediction, denoted by t_1 in the manuscript. For instance, if the time grids are $0, 1/60, 2/60, ..., 1$, then startT = 25 means that the first prediction is made at $t = 24/60$.
`link`	The link function used in functional generalized linear models, e.g. "logit", "probit".
`pve`	Proportion of variance explained in FPCA.
`nbasis`	Number of B-spline basis functions needed for estimation of the mean function and smoothing of covariance.
`weight`	Weight for each individual.

Value

list_fpcaFit	FPCA decomposition at each time grid from startT to the end.
list_paraEst	Parameter estimates at each time grid from startT to the end.

Examples

```
library(reinforcedPred)

# take the example training data (univariate Z) from the reinforcedPred package
# see documentation for details about the data set train_data_uniZ
Y <- as.numeric(train_data_uniZ$Y)
tildeX.missing <- as.matrix(train_data_uniZ[,2:62])
Z <- as.numeric(train_data_uniZ$Z)

# analysis starts
startT <- 55
link <- "probit"
weight <- rep(1, length(Y))

result <- modelFit(Y, tildeX.missing, Z, startT, link, pve = 0.99, nbasis = 10, weight)

# obtained parameter estimates and FPCA decompositions
list_paraEst <- result$list_paraEst
list_fpcaFit <- result$list_fpcaFit
```

---

modelFit_VS *Model fit for the training set, variable selection version*

Description

modelFit_VS outputs the FPCA (functional principal component analysis) decomposition and the elastic net logistic regression at each time grid. This function is used when the baseline covariates Z are high-dimensional.

Usage

```
modelFit_VS(Y, X, Z, startT, pve, nbasis, weight)
```

Arguments

Y	The outcome variable, vector of length n, taking values in $1, 0, NA$, where 1 = disease, 0 = not, NA = missing.
X	Observed longitudinal biomarker, matrix of n by $nTotal$, where $nTotal$ denotes the total number of time grids. Missing values are denoted by NA.
Z	Other baseline covariates.

startT	Time of the first prediction, denoted by t_1 in the manuscript. For instance, if the time grids are $0, 1/60, 2/60, ..., 1$, then startT = 25 means that the first prediction is made at $t = 24/60$.
pve	Proportion of variance explained in FPCA.
nbasis	Number of B-spline basis functions needed for estimation of the mean function and smoothing of covariance.
weight	Weight for each individual.

Value

list_fpcaFit	FPCA decomposition at each time grid from startT to the end.
list_cvfit	Elastic net logistic regression at each time grid from startT to the end.

Examples

```
library(reinforcedPred)

# take the example training data (high dimensional Z) from the reinforcedPred package
# see documentation for details about the data set train_data_mulZ
Y <- as.numeric(train_data_mulZ$Y)
tildeX.missing <- as.matrix(train_data_mulZ[,2:62])
Z <- as.matrix(train_data_mulZ[,63:dim(train_data_mulZ)[2]])

# analysis starts
startT <- 25
weight <- rep(1, length(Y))

result <- modelFit_VS(Y, tildeX.missing, Z, startT, pve = 0.99, nbasis = 10, weight)

# obtained elastic net logistic regression fit and FPCA decompositions
list_cvfit <- result$list_cvfit
list_fpcaFit <- result$list_fpcaFit
```

---

modelPredict *Risk prediction on the test set*

Description

For a fixed threshold value τ, modelPredict predicts the outcome Y for subjects in the test set. This function also outputs the cost associated with the prediction procedure.

Usage

```
modelPredict(list_fpcaFit, list_paraEst, Xtest, Ztest, startT, tau)
```

Arguments

list_fpcaFit	Obtained FPCA decomposition from modelFit.
list_paraEst	Obtained parameter estimates from modelFit.
Xtest	Longitudinal biomarker data for subjects in the test set, matrix of $testn$ by $nTotal$. Missing values are denoted by NA.
Ztest	Other baseline covariates for subjects in the test set.
startT	Time of the first prediction, denoted by t_1 in the manuscript. For instance, if the time grids are $0, 1/60, 2/60, ..., 1$, then startT = 25 means that the first prediction is made at $t = 24/60$.
tau	The threshold value τ.

Value

final.label	Predicted outcome Y for subjects in the test set, vector of length $testn$.
avg.cost	Average cost when we applied this prediction procedure to the test set.
cost	Cost for each subject, vector of length $testn$. For some subjects, we make a definite decision early. For others, we follow up with a long period of time. Hence the cost is different for each individual.

Examples

```
# see the example from function reinforced.
```

---

modelPredict_VS *Risk prediction on the test set, variable selection version*

Description

For a fixed threshold value τ, modelPredict_VS predicts the outcome Y for subjects in the test set. This function also outputs the cost associated with the prediction procedure. This function is used when the baseline covariates Z are high-dimensional.

Usage

```
modelPredict_VS(list_fpcaFit, list_cvfit, Xtest, Ztest, startT, tau)
```

Arguments

list_fpcaFit	Obtained FPCA decomposition from modelFit_VS.
list_cvfit	Obtained elastic net logistic regression from modelFit_VS.
Xtest	Longitudinal biomarker data for subjects in the test set, matrix of $testn$ by $nTotal$. Missing values are denoted by NA.
Ztest	Other baseline covariates for subjects in the test set.
startT	Time of the first prediction, denoted by t_1 in the manuscript. For instance, if the time grids are $0, 1/60, 2/60, ..., 1$, then startT = 25 means that the first prediction is made at $t = 24/60$.
tau	The threshold value τ.

Value

final.label	Predicted outcome Y for subjects in the test set, vector of length $testn$.
avg.cost	Average cost when we applied this prediction procedure to the test set.
cost	Cost for each subject, vector of length $testn$. For some subjects, we make a definite decision early. For others, we follow up with a long period of time. Hence the cost is different for each individual.

Examples

```
# see the example from function reinforced_VS.
```

---

reinforced *Reinforced risk prediction with budget constraint*

Description

reinforced implements a cross-validation approach to find an optimal τ such that the misclassification error is minimized under a certain budget constraint.

Usage

```
reinforced(Y, X, Z, budget, folds, startT, link, pve = 0.99,
  nbasis = 10, weight)
```

Arguments

Y	The outcome variable, vector of length n, taking values in $1, 0, NA$, where 1 = disease, 0 = not, NA = missing.
X	Observed longitudinal biomarker, matrix of n by $nTotal$, where $nTotal$ denotes the total number of time grids. Missing values are denoted by NA.
Z	Other baseline covariates.
budget	The budget constraint. For instance, if the time grids are $0, 1/60, 2/60, ..., 1$. Budget = 30 means that the average follow up was no longer than 30 time grids. This is equivalent to saying that on average, we want to make a definite prediction before time $t = 0.5$.
folds	Folds in cross-validation, usually 5 or 10.
startT	Time of the first prediction, denoted by t_1 in the manuscript. For instance, if the time grids are $0, 1/60, 2/60, ..., 1$, then startT = 25 means that the first prediction is made at $t = 24/60$.
link	The link function used in functional generalized linear models, e.g. "logit", "probit".
pve	Proportion of variance explained in FPCA, default value is 0.99.

nbasis	Number of B-spline basis functions needed for estimation of the mean function and smoothing of covariance. Default value is 10 in refund package, sometimes a smaller number is needed when there are a small number of time grids.
weight	A user-supplied weight for each individual. If the user did not supply the weight, we use an inverse probability weighting method to calculate a weight. See details in section 3.4 of the manuscript.

Value

final.result	The FPCA fit and the parameter estimates at each time grid from startT to the end.
final.tau	The optimal τ that minimizes the misclassification error under the budget constraint.

Examples

```
library(reinforcedPred)
set.seed(1)

# take the example training data (univariate Z) from the reinforcedPred package
# see documentation for details about the data set train_data_uniZ
Y <- as.numeric(train_data_uniZ$Y)
tildeX.missing <- as.matrix(train_data_uniZ[,2:62])
Z <- as.numeric(train_data_uniZ$Z)

# analysis starts
budget <- 45
folds <- 5
startT <- 25
link <- "probit"

result <- reinforced(Y, tildeX.missing, Z, budget, folds, startT, link, pve = 0.99, nbasis = 10)

# obtained parameter estimates and FPCA decompositions
list_paraEst <- (result$final.result)$list_paraEst
list_fpcaFit <- (result$final.result)$list_fpcaFit

# optimal tau that minimizes the misclassification error under the budget constraint
final.tau <- result$final.tau
final.tau

# use the fitted model to predict the label Y for subjects in the test data
# see documentation for details about the data set test_data_uniZ
testY <- as.numeric(test_data_uniZ$testY)
test.tildeX.missing <- as.matrix(test_data_uniZ[,2:62])
test.Z <- as.numeric(test_data_uniZ$test.Z)

pred <- modelPredict(list_fpcaFit, list_paraEst, test.tildeX.missing, test.Z, startT, final.tau)

# predicted outcome Y for each subject in the test data
```

```
predY.test <- pred$final.label
# misclassification error
mis.error <- sum(predY.test != testY, na.rm = TRUE) / sum(!is.na(testY))
mis.error

# the average cost when we applied the prediction procedure to the test data
pred$avg.cost
```

`reinforced_VS` *Reinforced risk prediction with budget constraint, variable selection version*

Description

`reinforced_VS` implements a cross-validation approach to find an optimal τ such that the misclassification error is minimized under a certain budget constraint. This function is used when the baseline covariates are of high-dimension.

Usage

```
reinforced_VS(Y, X, Z, budget, folds, startT, pve = 0.99, nbasis = 10,
  weight)
```

Arguments

`Y`	The outcome variable, vector of length n, taking values in $1, 0, NA$, where 1 = disease, 0 = not, NA = missing.
`X`	Observed longitudinal biomarker, matrix of n by $nTotal$, where $nTotal$ denotes the total number of time grids. Missing values are denoted by NA.
`Z`	Other baseline covariates.
`budget`	The budget constraint. For instance, if the time grids are $0, 1/60, 2/60, ..., 1$. Budget = 30 means that the average follow up was no longer than 30 time grids. This is equivalent to saying that on average, we want to make a definite prediction before time $t = 0.5$.
`folds`	Folds in cross-validation, usually 5 or 10.
`startT`	Time of the first prediction, denoted by t_1 in the manuscript. For instance, if the time grids are $0, 1/60, 2/60, ..., 1$, then startT = 25 means that the first prediction is made at $t = 24/60$.
`pve`	Proportion of variance explained in FPCA, default value is 0.99.
`nbasis`	Number of B-spline basis functions needed for estimation of the mean function and smoothing of covariance. Default value is 10 in refund package, sometimes a smaller number is needed when there are a small number of time grids.
`weight`	A user-supplied weight for each individual. If the user did not supply the weight, we use an inverse probability weighting method to calculate a weight. See details in section 3.4 of the manuscript.

Value

final.result	The FPCA fit and the elastic net logistic regression fit at each time grid from startT to the end.
final.tau	The optimal τ that minimizes the misclassification error under the budget constraint.

Examples

```
library(reinforcedPred)
set.seed(1)

# take the example training data (high dimensional Z) from the reinforcedPred package
# see documentation for details about the data set train_data_mulZ
Y <- as.numeric(train_data_mulZ$Y)
tildeX.missing <- as.matrix(train_data_mulZ[,2:62])
Z <- as.matrix(train_data_mulZ[,63:dim(train_data_mulZ)[2]])

# analysis starts
budget <- 45
folds <- 5
startT <- 25

result <- reinforced_VS(Y, tildeX.missing, Z, budget, folds, startT, pve = 0.99, nbasis = 10)

# obtained elastic net logistic regression fit and FPCA decompositions
list_cvfit <- (result$final.result)$list_cvfit
list_fpcaFit <- (result$final.result)$list_fpcaFit

# optimal tau that minimizes the misclassification error under the budget constraint
final.tau <- result$final.tau
final.tau

# use the fitted model to predict the label Y for subjects in the test data
# see documentation for details about the data set test_data_mulZ
testY <- as.numeric(test_data_mulZ$testY)
test.tildeX.missing <- as.matrix(test_data_mulZ[,2:62])
test.Z <- as.matrix(test_data_mulZ[,63:dim(test_data_mulZ)[2]])

pred <- modelPredict_VS(list_fpcaFit, list_cvfit, test.tildeX.missing, test.Z, startT, final.tau)

# predicted outcome Y for each subject in the test data
predY.test <- pred$final.label
# misclassification error
mis.error <- sum(predY.test != testY, na.rm = TRUE) / sum(!is.na(testY))
mis.error

# the average cost when we applied the prediction procedure to the test data
pred$avg.cost
```

test_data_mulZ *Example test data (high dimensional Z)*

Description

Example test data (high dimensional Z)

Usage

```
test_data_mulZ
```

Format

A data frame with 2000 rows and 112 columns. The 1st column is the outcome variable Y, starting from the 2nd column to 62nd column is the longitudinal biomarker at 61 time grids, the 63rd column to 112nd column are other baseline covariate Z.

Source

A simulated data set

test_data_uniZ *Example test data (univariate Z)*

Description

Example test data (univariate Z)

Usage

```
test_data_uniZ
```

Format

A data frame with 2000 rows and 63 columns. The 1st column is the outcome variable Y, starting from the 2nd column to 62nd column is the longitudinal biomarker at 61 time grids, the 63rd column is other baseline covariate Z.

Source

A simulated data set

train_data_mulZ *Example training data (high dimensional Z)*

Description

Example training data (high dimensional Z)

Usage

```
train_data_mulZ
```

Format

A data frame with 400 rows and 112 columns. The 1st column is the outcome variable Y, starting from the 2nd column to 62nd column is the longitudinal biomarker at 61 time grids, the 63rd column to 112nd column are other baseline covariate Z.

Source

A simulated data set

train_data_uniZ *Example training data (univariate Z)*

Description

Example training data (univariate Z)

Usage

```
train_data_uniZ
```

Format

A data frame with 100 rows and 63 columns. The 1st column is the outcome variable Y, starting from the 2nd column to 62nd column is the longitudinal biomarker at 61 time grids, the 63rd column is other baseline covariate Z.

Source

A simulated data set

Index