

Supplementary Material for Two Stage Adaptive Design for Prognostic Biomarker Signature with Survival Endpoint

Biyue Dai

July 22, 2019

This article provides a guidance of the supplementary R functions for the manuscript “Two-Stage Adaptive Design for Prognostic Biomarker Signatures with a Survival Endpoint”

Documentation of Functions

`two_stage_permutation()`

Description

Analyze a given data set using the two-stage design. Inference in the first stage is conducted via permutation.

Arguments

survY: output from surv function in survival package

X: a matrix or data frame of variables of interest, without missing value

itrB: number of iteration used in permutation

seed: seed number for reproducing results (default: NULL)

nfold: number of folds used for cross-validation (default: 10)

S: proportion of samples used in first stage (default: 0.5)

alpha_1: level of test in first stage (default: 0.25)

alpha_2: level of test in second stage (default: 0.2)

printmodel: logical value. TRUE indicates the final signature will be print out (default: FALSE)

`two_stage_bootstrap()`

Description

Analyze a given data set using the two-stage design. Inference in the first stage is conducted via bootstrap.

Arguments

survY: output from surv function in survival package

X: a matrix or data frame of variables of interest, without missing value

itrB: number of iteration used in permutation

seed: seed number for reproducing results (default: NULL)

nfold: number of folds used for cross-validation (default: 10)

S: proportion of samples used in first stage (default: 0.5)

alpha_1: level of test in first stage (default: 0.25)

alpha_2: level of test in second stage (default: 0.2)

null_C : The value of C Index under the null hypothesis of interest (default: 0.5)

printmodel : Logical value that allows the user to print out the model that is built at the end of first stage. (default: FALSE)

.cvFolds()

Lower level function that assigns fold IDs given number of folds.

cvcox()

Lower level function that calculates cross-validated linear predictors from Cox Regression

Examples

```
# Load functions
source("functions.R")
# Load data
data(lung) # built-in data set in survival package

lung <- na.omit(lung) # for simplicity, I only took the complete cases
survY <- Surv(lung$time, lung$status)
X <- lung[,c("age", "sex", "ph.ecog", "ph.karno", "meal.cal", "wt.loss")]
```

Use Permuation Test in the First Stage:

```
two_stage_permutation(survY, X, itrB = 100, S = 0.5, seed = 666, printmodel = TRUE)
```

```
## [[1]]
##      survC threshold.75%  early_stop      C_2      se_2
##      0.62226212  0.52244165  0.00000000  0.60592532  0.04186525
##      final_reject      cens      cens1      cens2
##      1.00000000  0.28143713  0.26190476  0.30120482
##
## $final_model
## Call:
## coxph(formula = survY_1 ~ ., data = df)
##
##              coef exp(coef)  se(coef)      z      p
## age      0.0227310  1.0229913  0.0143087  1.589 0.112149
## sex     -0.5898582  0.5544059  0.2973876 -1.983 0.047315
## ph.ecog  1.0880312  2.9684240  0.2944190  3.696 0.000219
## ph.karno  0.0298175  1.0302665  0.0142066  2.099 0.035829
## meal.cal -0.0001331  0.9998669  0.0004459 -0.299 0.765284
## wt.loss  -0.0097731  0.9902745  0.0117920 -0.829 0.407220
##
## Likelihood ratio test=18.94 on 6 df, p=0.004259
## n= 84, number of events= 62
```

Use Bootstrap in the First Stage, testing Null Hypothesis: $H_0: C = 0.5$

```
two_stage_bootstrap(survY, X, itrB = 100, seed = 666, S = 0.5)
```

```
##      survC threshold.25%  early_stop      C_2      se_2
##      0.62226212  0.62508488  0.00000000  0.60592532  0.04186525
##      final_reject      cens      cens1      cens2
##      1.00000000  0.28143713  0.26190476  0.30120482
```

Use Bootstrap in the First Stage, testing Null Hypothesis: $H_0: C = 0.6$

```
two_stage_bootstrap(survY, X, itrB = 100, seed = 666, S = 0.5, null_C = 0.6)
```

```
##      survC threshold.25%  early_stop      C_2      se_2
##      0.62226212  0.62508488  0.00000000  0.60592532  0.04186525
##      final_reject      cens      cens1      cens2
##      0.00000000  0.28143713  0.26190476  0.30120482
```

The output contains the following values:

survC: cross-validated C Index of the model built in Stage 1

threshold: the rejection threshold in permutation and bootstrap distribution

early_stop: indicator of whether the trial is stopped early or not. 1: stopped early; 2: continue to second stage

C_2: C Index in stage 2. It would be NA if the study is stopped early.

se_2: standard error of C Index in stage 2. It would be NA if the study is stopped early.

final_reject: Indicator of whether the signature is finally validated. 1: validated; 2: validation has failed.

`cens`, `cens1`, `cens2`: censoring proportion overall, for stage 1 and for stage 2.

If **`printmodel`** is set to be true, then the final model that is built based on `coxph` will be printed.