# Package 'TGST'

October 12, 2022

## R topics documented:

---

cal.AUC                              *Calculate AUC*

---

### Description

This function gives you the AUC associated with the rules set.

### Usage

```
cal.AUC(Z, S, l, u)
```

### Arguments

| | |
|---|---|
| Z | True disease status (No disease / treatment success coded as Z=0, diseased / treatment failure coded as Z=1). |
| S | Risk score. |
| l | Lower cutoff of all possible tripartite rules. |
| u | Upper cutoff of all possible tripartite rules. |

### Value

AUC.

### Examples

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
phi = 0.1 #10% of patients taking viral load test
rules = nonpar.rules( Z, S, phi)
cal.AUC(Z,S,rules[,1],rules[,2])
```

---

| | |
|---|---|
| Check.exp.tilt | *Check exponential tilt model assumption* |

---

## Description

This function provides graphical assessment to the suitability of the exponential tilt model for risk score in finding optimal tripartite rules by semiparametric approach.

$$g1(s) = exp(\tilde{\beta}_0 + \beta_1 * s) * g0(s)$$

## Usage

```
Check.exp.tilt(Z, S)
```

## Arguments

| | |
|---|---|
| Z | True disease status (No disease / treatment success coded as Z=0, diseased / treatment failure coded as Z=1). |
| S | Risk score. |

## Value

Plot of empirical density for risk score S, joint empirical density for (S,Z=1) and (S,Z=0), and the density under the exponential tilt model assumption for (S,Z=1) and (S,Z=0).

## Examples

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
Check.exp.tilt( Z, S)
```

---

| | |
|---|---|
| CV.TGST | *Cross Validation* |

---

## Description

This function allows you to compute the average of misdiagnoses rate for viral failure and the optimal risk under min-$\lambda$ rules from K-fold cross-validation.

## Usage

```
CV.TGST(Obj, lambda, K = 10)
```

## Arguments

| | |
|---|---|
| `Obj` | An object of class TGST. |
| `lambda` | A user-specified weight that reflects relative loss for the two types of misdiagnoses, taking value in $[0, 1]$. $Loss = \lambda * I(FN) + (1 - \lambda) * I(FP)$. |
| `K` | Number of folds in cross validation. The default is 10. |

## Value

Cross-validation results.

## Examples

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
phi = 0.1 #10% of patients taking viral load test
Obj = TGST(Z, S, phi, method="nonpar")
lambda = 0.8
CV.TGST(Obj, lambda, K=10)
```

---

nonpar.fnr.fpr                 *Nonparametric FNR FPR of the rules*

---

## Description

This function gives you the nonparametric FNR and FPR associated with a given tripartite rule.

## Usage

```
nonpar.fnr.fpr(Z, S, l, u)
```

## Arguments

| | |
|---|---|
| `Z` | True disease status (No disease / treatment success coded as Z=0, diseased / treatment failure coded as Z=1). |
| `S` | Risk score. |
| `l` | Lower cutoff of tripartite rule. |
| `u` | Upper cutoff of tripartite rule. |

## Value

Matrix with 2 columns. Each row is a set of nonparametric (FNR, FPR) on an associated tripartite rule.

## Examples

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
phi = 0.1 #10% of patients taking viral load test
rules = nonpar.rules( Z, S, phi)
nonpar.fnr.fpr(Z,S,rules[1,1],rules[1,2])
```

---

nonpar.rules                    *Nonparametric Rules Set*

---

## Description

This function gives you all possible cutoffs $[l, u]$ for tripartite rules, by applying nonparametric search to the given data.

$$P(Sin[l, u]) \leq \phi$$

## Usage

```
nonpar.rules(Z, S, phi)
```

## Arguments

| | |
|---|---|
| Z | True disease status (No disease / treatment success coded as Z=0, diseased / treatment failure coded as Z=1). |
| S | Risk score. |
| phi | Percentage of patients taking viral load test. |

## Value

Matrix with 2 columns. Each row is a possible tripartite rule, with output on lower and upper cutoff.

## Examples

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
phi = 0.1 #10% of patients taking viral load test
nonpar.rules( Z, S, phi)
```

---

Opt.nonpar.rule                 *Optimal Nonparametric Rule*

---

**Description**

This function gives you the optimal nonparametric tripartite rule that minimizes the min-$\lambda$ rules.

**Usage**

```
Opt.nonpar.rule(Z, S, phi, lambda)
```

**Arguments**

| | |
|---|---|
| Z | True disease status (No disease / treatment success coded as Z=0, diseased / treatment failure coded as Z=1). |
| S | Risk score. |
| phi | Percentage of patients taking viral load test. |
| lambda | A user-specified weight that reflects relative loss for the two types of misdiagnoses, taking value in $[0, 1]$. $Loss = \lambda * I(FN) + (1 - \lambda) * I(FP)$. |

**Value**

Optimal nonparametric rule and its associated misclassification rates (FNR, FPR), optimal lambda risk, and total misclassification rate (TMR).

**Examples**

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
phi = 0.1 #10% of patients taking viral load test
lambda = 0.5
Opt.nonpar.rule( Z, S, phi, lambda)
```

---

Opt.semipar.rule                *Optimal Semiparametric Rule*

---

**Description**

This function gives you the optimal semiparametric tripartite rule that minimizes the min-$\lambda$ rules.

**Usage**

```
Opt.semipar.rule(Z, S, phi, lambda)
```

**Arguments**

| | |
|---|---|
| Z | True disease status (No disease / treatment success coded as Z=0, diseased / treatment failure coded as Z=1). |
| S | Risk score. |
| phi | Percentage of patients taking viral load test. |
| lambda | A user-specified weight that reflects relative loss for the two types of misdiagnoses, taking value in $[0, 1]$. $Loss = \lambda * I(FN) + (1 - \lambda) * I(FP)$. |

**Value**

Optimal semiparametric rule and its associated misclassification rates (FNR, FPR), optimal lambda risk, and total misclassification rate (TMR).

**Examples**

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
phi = 0.1 #10% of patients taking viral load test
lambda = 0.5
Opt.semipar.rule( Z, S, phi, lambda)
```

---

OptimalRule                    *Optimal Tripartite Rule*

---

**Description**

`OptimalRule` is the main function of `TGST` and it gives you the optimal tripartite rule that minimizes the min-$\lambda$ risk based on the type of user selected approach. The function takes the risk score and true disease status from a training data set and returns the optimal tripartite rule under the specified proportion of patients able to take gold standard test.

**Usage**

```
OptimalRule(Obj, lambda)
```

**Arguments**

| | |
|---|---|
| Obj | An object of class TGST. |
| lambda | A user-specified weight that reflects relative loss for the two types of misdiagnoses, taking value in $[0, 1]$. $Loss = \lambda * I(FN) + (1 - \lambda) * I(FP)$. |

**Value**

Optimal tripartite rule.

## Examples

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
phi = 0.1 #10% of patients taking viral load test
lambda = 0.5
Obj = TGST(Z, S, phi, method="nonpar")
OptimalRule(Obj, lambda)
```

---

Output-class                    *Constructor of Output class*

---

## Description

This class contains invisible results of OptimalRule function.

## Details

**phi** Percentage of patients taking viral load test.

**Z** A vector of true disease status (Failure Status coded as Z=1).

**S** A vector of risk Score.

**Rules** A matrix of all possible tripartite rules (two cutoffs) derived from the training data set.

**Nonparametric** A boolean indicating if nonparametric approach should be used in calculating the misclassfication rates. If FALSE, semiparametric approach would be used.

**FNR.FPR** A matrix with two columns of misclassification rates, FNR and FPR.

**OptRule** A numeric vector with two elements, the lower and upper cutoffs of the optimal tripartite rule.

The Output class adds optimal rule to the TGST class.

---

plot                            *plot function.*

---

## Description

This function This function gives visualize object of class TGST or Output.

This function This function gives visualize object of class TGST or Output.

## Usage

```
## S4 method for signature 'TGST'
plot(x)

## S4 method for signature 'Output'
plot(x)
```

## Arguments

x                Output object from [TGST](#) or [Output](#).

## Value

Distribution plot.

Distribution plot.

---

ROC.nonpar              *Nonparametric ROC Analysis*

---

## Description

This function performs ROC analysis for tripartite rules by nonparametric approach. If $plot = TRUE$, the ROC curve is returned.

## Usage

```
ROC.nonpar(Z, S, phi, plot = TRUE)
```

## Arguments

| | |
|---|---|
| Z | True disease status (No disease / treatment success coded as Z=0, diseased / treatment failure coded as Z=1). |
| S | Risk score. |
| phi | Percentage of patients taking viral load test. |
| plot | Logical parameter indicating if ROC curve should be plotted. Default is plot=TRUE. If false, then only AUC is calculated. |

## Value

AUC The area under the ROC curve. FNR Misdiagnoses rate for viral failure (false negative rate). FPR Misdiagnoses rate for treatment failure (false positive rate).

## Examples

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
phi = 0.1 #10% of patients taking viral load test
a = ROC.nonpar( Z, S, phi,plot=TRUE)
a$AUC
a$FNR
a$FPR
```

---

ROC.semipar                              *Semiparametric ROC Analysis*

---

### Description

This function performs ROC analysis on the rules from nonparametric approach. If $plot = TRUE$, the ROC curve is returned.

### Usage

```
ROC.semipar(Z, S, phi, plot = TRUE)
```

### Arguments

| | |
|---|---|
| Z | True disease status (No disease / treatment success coded as Z=0, diseased / treatment failure coded as Z=1). |
| S | Risk score. |
| phi | Percentage of patients taking viral load test. |
| plot | Logical parameter indicating if ROC curve should be plotted. Default is plot=TRUE. If false, then only AUC is calculated. |

### Value

AUC The area under the ROC curve. FNR Misdiagnoses rate for viral failure (false negative rate). FPR Misdiagnoses rate for treatment failure (false positive rate).

### Examples

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
phi = 0.1 #10% of patients taking viral load test
a = ROC.semipar( Z, S, phi,plot=TRUE)
a$AUC
a$FNR
a$FPR
```

---

ROCAnalysis                              *ROC Analysis*

---

### Description

This function performs ROC analysis for tripartite rules. If $plot = TRUE$, the ROC curve is returned.

## Usage

```
ROCAnalysis(Obj, plot = TRUE)
```

## Arguments

| | |
|---|---|
| Obj | An object of class TGST. |
| plot | Logical parameter indicating if ROC curve should be plotted. Default is plot=TRUE. If false, then only AUC is calculated. |

## Value

AUC (the area under ROC curve) and ROC curve.

## Examples

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
phi = 0.1 #10% of patients taking viral load test
lambda = 0.5
Obj = TGST(Z, S, phi, method="nonpar")
ROCAnalysis(Obj, plot=TRUE)
```

---

Semi.par.rule                    *Min TMR Semiparametric Rule*

---

## Description

This function gives you the optimal semiparametric tripartite rule that minimizes TMR (total misclassification risk).

## Usage

```
Semi.par.rule(Z, S, phi)
```

## Arguments

| | |
|---|---|
| Z | True disease status (No disease / treatment success coded as Z=0, diseased / treatment failure coded as Z=1). |
| S | Risk score. |
| phi | Percentage of patients taking viral load test. |

## Value

Semiparametric rule and its associated TMR.

## Examples

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
phi = 0.1 #10% of patients taking viral load test
Semi.par.rule( Z, S, phi)
```

---

semipar.fnr.fpr          *Semiparametric FNR FPR of the rules*

---

## Description

This function gives you the semiparametric FNR and FPR associated with a given tripartite rule.

## Usage

```
semipar.fnr.fpr(Z, S, l, u)
```

## Arguments

| | |
|---|---|
| Z | True disease status (No disease / treatment success coded as Z=0, diseased / treatment failure coded as Z=1). |
| S | Risk score. |
| l | Lower cutoff of tripartite rule. |
| u | Upper cutoff of tripartite rule. |

## Value

Matrix with 2 columns. Each row is a set of semiparametric (FNR, FPR) on an associated tripartite rule.

## Examples

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
phi = 0.1 #10% of patients taking viral load test
rules = nonpar.rules( Z, S, phi)
semipar.fnr.fpr(Z,S,rules[1,1],rules[1,2])
```

---

Simdata                                 *Simulated data for package illustration*

---

### Description

A simulated dataset containing true disease status and risk score. See details for simulation setting.

### Usage

```
data(Simdata)
```

### Format

A data frame with 8000 simulated observations on the following 2 variables.

Z  True disease status (No disease / treatment success coded as Z=0, diseased / treatment failure coded as Z=1).

S  Risk score. Higher risk score indicates larger tendency of diseased / treatment failure.

### Details

We first simulate viral status $Z$ assuming $Z \sim Bernoulli(p)$ with $p = 0.25$; and then conditional on $Z$, simulate $S|Z = z = ceiling(W)$ with $W \sim Gamma(\eta_z, \kappa_z)$ where $\eta$ and $\kappa$ are shape and scale parameters. $(\eta 0, \kappa 0) = (2.3, 80)$ and $(\eta 1, \kappa 1) = (9.2, 62)$.

### Examples

```
data(Simdata)
## maybe str(Simdata) ; plot(Simdata) ...
```

---

summary                                *summary function.*

---

### Description

This function gives the summary of the data from TGST.

### Usage

```
## S4 method for signature 'TGST'
summary(object)
```

### Arguments

object                 Output object from TGST.

**Value**

Percentage of treatment failure; Summary statistics (mean, standard deviation, minimum, median, maximum and IQR) of risk score by true disease status; Distribution plot.

---

TGST                                         *Create a TGST Object*

---

**Description**

Create a TGST object, usually used as an input for optimal rule search and ROC analysis.

**Usage**

```
TGST(Z, S, phi, method = "nonpar")
```

**Arguments**

| | |
|---|---|
| Z | A vector of true disease status (No disease / treatment success coded as Z=0, diseased / treatment failure coded as Z=1). |
| S | A vector of risk Score. |
| phi | Percentage of patients taking gold standard test. |
| method | Method for searching for the optimal tripartite rule, options are "nonpar" (default) and "semipar". |

**Value**

An object of class TGST.The class contains 6 slots: phi (percentage of gold standard tests), Z (true failure status), S (risk score), Rules (all possible tripartite rules), Nonparametric (logical indicator of the approach), and FNR.FPR (misclassification rates).

**Examples**

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
phi = 0.1 #10% of patients taking viral load test
TGST( Z, S, phi, method="nonpar")
```

---

TGST-class                          *Constructor of TGST class*

---

### Description

This class contains results of a run to search for all possible rules.

### Details

**phi**  Percentage of patients taking viral load test.

**Z**  A vector of true disease status (Failure Status coded as Z=1).

**S**  A vector of risk Score.

**Rules**  A matrix of all possible tripartite rules (two cutoffs) derived from the training data set.

**Nonparametric**  A boolean indicating if nonparametric approach should be used in calculating the misclassfication rates. If FALSE, semiparametric approach would be used.

**FNR.FPR**  A matrix with two columns of misclassification rates, FNR and FPR.

If res is the result of rankclust(), each slot of results can be reached by res[k]@slotname, where k is the number of clusters and slotname is the name of the slot we want to reach (see Output-class). For the slots ll, bic, icl, res["slotname"] returns a vector of size K containing the values of the slot for each number of clusters.

---

TGST.summ                          *Summary of Disease Status and Risk Score*

---

### Description

This function allows you to compute the percentage of diseased (equivalent to treatment failure Z=1) and show distribution summary of risk score (S) by the true disease status (Z).

### Usage

```
TGST.summ(Z, S)
```

### Arguments

| | |
|---|---|
| Z | True disease status (No disease / treatment success coded as Z=0, diseased / treatment failure coded as Z=1). |
| S | Risk score. |

### Value

Percentage of treatment failure; Summary statistics (mean, standard deviation, minimum, median, maximum and IQR) of risk score by true disease status; Distribution plot.

## Examples

```
d = Simdata
Z = d$Z # True Disease Status
S = d$S # Risk Score
TGST.summ(Z,S)
```

# Index