

Package ‘pirate’

October 14, 2022

Type Package

Title Generated Effect Modifier

Version 1.0.0

Date 2016-10-11

Author Eva Petkova, Zhe Su

Maintainer Zhe Su <Zhe.Su@nyumc.org>

Description An implementation of the generated effect modifier (GEM) method. This method constructs composite variables by linearly combining pre-treatment scalar patient characteristics to create optimal treatment effect modifiers in linear models. The optimal linear combination is called a GEM. Treatment is assumed to have been assigned at random. For reference, see E Petkova, T Tarpey, Z Su, and RT Ogden. Generated effect modifiers (GEMs) in randomized clinical trials. *Biostatistics* (First published online: July 27, 2016, <[doi:10.1093/biostatistics/kxw035](https://doi.org/10.1093/biostatistics/kxw035)>).

License MIT + file LICENSE

Imports plyr, MASS, ggplot2, Rcpp, RcppArmadillo

LinkingTo Rcpp, RcppArmadillo

Suggests rmarkdown

RoxygenNote 5.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-11-08 00:30:07

R topics documented:

data_generator	2
effectSize	4
gem_fit	5
gem_test	6
permute_pvalue	7
Index	9

data_generator *Functions for Simulating Data*

Description

When investigating the properties of GEM, the following three data generators are used in various simulations. They are designed to construct three specific types of data sets in the case of two treatment groups. See more detail in *E Petkova, T Tarpey, Z Su, and RT Ogden. Generated effect modifiers (GEMs) in randomized clinical trials. Biostatistics, (First published online: July 27, 2016). doi: 10.1093/biostatistics/kxw035.*

Usage

data_generator1(d, R2, v2, n, co, beta1, inter)

data_generator2(n, co, R2, bet, inter)

data_generator3(n, co, bet, inter)

Arguments

d	A scalar indicating the effect size of the GEM when the data is generated under a GEM model
R2	A scalar indicating the proportion of explained variance R^2 for the entire data set
v2	A scalar indicating the proportion of explained variance R^2 for the first treatment group
n	A scalar indicating the number of observation in each treatment group, assumed to be the same.
co	A p by p positive semidefinite matrix indicating the covariance matrix of the covariates
beta1	A vector of length p giving the regression coefficients for the first treatment group
inter	A vector of length 2 recording the intercepts β_{10}, β_{20} for the two treatment groups respectively
bet	A list with two elements, each a vector of length p , giving the regression coefficients for the two treatment groups respectively

Details

data_generator1 is used to create data where the outcome is a linear function of the covariates

$$y_j = \beta_{j0} + X\beta_j + \epsilon, j = 1, 2,$$

and the coefficients of covariates β are proportional between two treatment groups: $\beta_2 = b*\beta_1$. This type of data set matches perfectly with the motivation of GEM algorithm. β_1 is set as an argument

of the function while $\beta_2 = b * \beta_1$ is derived by controlling R^2 of the whole data and the effect size. See more detail in *Kraemer, H. C. (2013). Discovering, comparing, and combining moderators of treatment on outcome after randomized clinical trials: a parametric approach. Statistics in medicine, 32(11), 1964-1973.*

data_generator2 is similar to the first one except that the coefficients of the covariates are not necessarily proportional. Hence two β 's should be specified as arguments of the function.

data_generator3 constructs a data set where the outcome under each treatment condition is given for all subjects. In addition, no error is added to the mean outcome. This generator is useful for obtaining the "true" value of a treatment decision. This data generator is similar to data generator2

$$y_j = \beta_{j0} + X\beta_j, j = 1, 2.$$

Value

The output from these functions are different:

For the function data_generator1

1. dat A data frame with first and second column as treatment group index and outcome respectively, and each of the remaining columns as a covariate.
2. bet A list with two elements, each a vector of length p , giving the regression coefficients for the two treatment groups respectively
3. error_12 A vector of length three representing the standard deviation of ϵ , the explained variance by the linear part for the first and second treatment group respectively.

For the function data_generator2

1. dat A data frame with first and second column as treatment group index and outcome respectively, and each of the remaining columns as a covariate.
2. bet list with two elements, each a vector of length p , giving the regression coefficients for the two treatment groups respectively
3. error A scalar representing the standard deviation of ϵ

For the function data_generator3

1. y0 Outcome vector under the first treatment assignment
2. y1 Outcome vector under the second treatment assignment
3. X Design matrix for the covariates
4. oracle Average of the outcome if each subject takes the optimal treatment assignment
5. invOracle Average of the outcome if each subject does not take the optimal treatment assignment

Examples

```
#constructing the covariance matrix
co <- matrix(0.2, 30, 30)
diag(co) <- 1
dataEx <- data_generator1(d = 0.3, R2 = 0.5, v2 = 1, n = 3000,
                        co = co, beta1 = rep(1,30), inter = c(0,0))
```

```
#check the R squared of the simulated data set
dat <- dataEx[[1]]
summary(lm(V2~factor(trt)*(V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13+V14+V15+V16+
V17+V18+V19+V20+V21+V22+V23+V24+V25+V26+V27+V28+V29+V30+V31+V32),data=dat))

bigData <- data_generator3(n = 10000,co = co,bet =dataEx[[2]], inter = c(0,0))
```

effectSize

Effect Size Calculation

Description

Calculates the effect size of a moderator when there are only two treatment groups. More details please see: *Kraemer, H. C. (2013). Discovering, comparing, and combining moderators of treatment on outcome after randomized clinical trials: a parametric approach. Statistics in medicine, 32(11), 1964-1973.*

Usage

```
effectSize(response, treatment, moderator)
```

Arguments

response	A vector giving the outcome for all subjects
treatment	A vector giving the treatment group index for all subjects
moderator	A vector giving the moderator

Value

eff_size the calculated effect size for the moderator

Examples

```
#constructing the covariance matrix
co <- matrix(0.2, 10, 10)
diag(co) <- 1
dataEx <- data_generator1(d = 0.3, R2 = 0.5, v2 = 1, n = 3000,
                          co = co, beta1 = rep(1,10),inter = c(0,0))

#fit the GEM
dat <- dataEx[[1]]
model_nu <- gem_fit(dat = dat, method = "nu")
augmentData <- model_nu[[4]]
es <- effectSize(augmentData$y, augmentData$trt, augmentData$Z)
#this should be the same with effect size calculated by the gem_fit function
```

gem_fit

*GEM Fit***Description**

The main algorithm in **pirate** package for calculating the coefficients of the linear combination of the covariates to generate a GEM. This function can be applied to data sets with more than two treatment groups. See more detail in *E Petkova, T Tarpey, Z Su, and RT Ogden. Generated effect modifiers (GEMs) in randomized clinical trials. Biostatistics, (First published online: July 27, 2016). doi: 10.1093/biostatistics/kxw035.*

Usage

```
gem_fit(dat, method = "F")
```

Arguments

dat Data frame with first column as the treatment index, second column as the outcome, and the remaining columns as the covariates design matrix. The elements of the treatment index take K distinct values, where K is the number of treatment groups. The outcome has to be a continuous variable.

method Choice of the criterion that the generated effect modifier optimizes. This is a string in `c("nu", "de", "F")`, which corresponds to the numerator, denominator and F-statistics criteria respectively. The default method is the F-statistics method.

Details

`gemObject` is a list of three elements. The first element is the calculated weight α for combining the predictors X . The second element contains the K vectors of coefficients $(\gamma_{j0}, \gamma_{j1})$ from

$$y_j = \gamma_{j0} + (X\alpha)\gamma_{j1} + \epsilon, j = 1, \dots, K,$$

for the K treatment groups respectively. The third element contains the K vectors of coefficients from the unconstrained linear regression models

$$y_j = \beta_{j0} + X\beta_{j1} + \epsilon, j = 1, \dots, K,$$

for the K treatment groups respectively.

Value

1. `method` The criterion used to generate the GEM
2. `gemObject` Fitted result for the GEM model, see more in **Details**
3. `p_value` The p-value for the interaction term in model $Y = a + trt + Z + trt * Z + \epsilon$, where Z is the GEM
4. `Augmented_Data` The input data augmented with the GEM as the last column
5. `effect.size` The effect size of the GEM if there are only two treatment groups
6. `plot` A scatter plot of Y versus the GEM with fitted lines and grouped by treatment

Examples

```
#constructing the covariance matrix
co <- matrix(0.2, 10, 10)
diag(co) <- 1
dataEx <- data_generator1(d = 0.3, R2 = 0.5, v2 = 1, n = 300,
                        co = co, beta1 = rep(1,10), inter = c(0,0))

#fit the GEM
dat <- dataEx[[1]]
model_nu <- gem_fit(dat = dat, method = "nu")
model_de <- gem_fit(dat = dat, method = "de")
model_F <- gem_fit(dat = dat, method = "F")
```

gem_test

Implement Fitted GEM criterion on a Data Set

Description

Calculates the value of the treatment decision based on the information from a fitted GEM model. The information is contained in the `gemObject`, which is obtained from the `gem_fit` function. With `gem_test_sample` the value of decision based on a GEM model is obtained for a test sample that for each subject has observed outcome under only one treatment condition (this would be the situation when the test sample is a "real" data set). With `gem_test_simsample` the value of decision is calculated when the test sample has the outcome under all treatment conditions for all subjects (this would be the situation when simulated data is used).

Usage

```
gem_test_sample(dat, gemObject)

gem_test_simsample(y0, y1, XFrame, gemObject)
```

Arguments

<code>dat</code>	Data frame with first column as the treatment index, second column as outcome, and the remaining columns as covariates design matrix. The treatment index could only have two values and the outcome should be of continuous type.
<code>gemObject</code>	A list containing the fitted GEM information, which could be the second element <code>gemObject</code> of the output from the <code>gem_fit</code> function or a list with the same structure.
<code>y0</code>	Outcome vector for all subjects under the first treatment assignment
<code>y1</code>	Outcome vector for all subjects under the second treatment assignment
<code>XFrame</code>	Design matrix of the predictors in the simulated sample with known outcomes under both treatment conditions

Details

The treatment decision rule estimated by the `gem_fit` function can be applied to a new (real) data set to estimate its value. It can also be applied to a simulated data set, where the outcome is known under both conditions, to study its performance of such treatment decision rule. These two functions correspond to those two situations and compute the population average benefit (or the value of the decision rule based on a GEM model).

Value

`PAB_gem` Population average benefit of a the treatment regime based on a GEM model

`PAB_unres` Population average benefit of a treatment regime based on an unrestricted linear model

`opt_gem` The optimal treatment assignment for each subject

Examples

```
#constructing the covariance matrix
co <- matrix(0.2, 10, 10)
diag(co) <- 1
dataEx <- data_generator1(d = 0.3, R2 = 0.5, v2 = 1, n = 300,
                        co = co, beta1 = rep(1,10), inter = c(0,0))

#fit the GEM
dat <- dataEx[[1]]
model_nu <- gem_fit(dat = dat, method = "nu")

#calculate the population average benefit in the data sample
gem_test_sample(dat,model_nu[[2]])

#calculate the population average benefit when outcome under both treatment conditions
#is known, usually in a simulated sample
bigData <- data_generator3(n = 1000,co = co,bet =dataEx[[2]], inter = c(0,0))
gem_test_simsample(bigData[[1]],bigData[[2]],bigData[[3]],model_nu[[2]])
```

permutate_pvalue

Calculation of permutation p-value

Description

Calculates the permutation p-value for a fitted GEM. See more detail in *E Petkova, T Tarpey, Z Su, and RT Ogden. Generated effect modifiers (GEMs) in randomized clinical trials. Biostatistics, (First published online: July 27, 2016). doi: 10.1093/biostatistics/kxw035.*

Usage

```
permutate_pvalue(dat, permuteN, method = "F")
```

Arguments

<code>dat</code>	Data frame with first column as the treatment index, second column as the outcome, and the remaining columns as the covariates design matrix. The elements of the treatment index take K distinct values, where K is the number of treatment groups. The outcome has to be a continuous variable.
<code>permuteN</code>	Number of permutation
<code>method</code>	Choice of the criterion that the generated effect modifier optimizes. This is a string in <code>c("nu", "de", "F")</code> , which corresponds to the numerator, denominator and F-statistics criteria respectively. The default method is the F-statistics method.

Value

`perm_p` Permutation p-value for the data and chosen criterion
`p` A vector of calculated p-value for the original and permuted data set under the chosen criterion

Examples

```
#constructing the covariance matrix
co <- matrix(0.2, 10, 10)
diag(co) <- 1
#simulate a data set
dataEx <- data_generator1(d = 0.3, R2 = 0.5, v2 = 1, n = 300,
                        co = co, beta1 = rep(1,10), inter = c(0,0))
#calculate the permuted p value
dat <- dataEx[[1]]
permute_pvalue(dat, permuteN = 200, method = "nu")
```


Index

`data_generator`, [2](#)
`data_generator1 (data_generator)`, [2](#)
`data_generator2 (data_generator)`, [2](#)
`data_generator3 (data_generator)`, [2](#)

`effectSize`, [4](#)

`gem_fit`, [5](#)
`gem_test`, [6](#)
`gem_test_sample (gem_test)`, [6](#)
`gem_test_simsample (gem_test)`, [6](#)

`permute_pvalue`, [7](#)