Package ‘chngpt’

August 28, 2017

LazyLoad yes
LazyData yes
Version 2017.08-28
Title Change Point Regression
Author Youyi Fong
Maintainer Youyi Fong <youyifong@gmail.com>
Depends R (>= 3.2.0), kyotil, MASS, boot, splines
Suggests R.rsp, RUnit, mvtnorm
Imports survival
VignetteBuilder R.rsp

Description Change point regression models are also called two-phase regression, break-point regression, split-point regression, structural change models and threshold regression models. Hypothesis testing in change point logistic regression with or without interaction terms (Fong, Y., Di, C., Perma, S. (2014) <doi:10.1002/sim.6419>). Several options are provided for testing in models with interaction, including a maximum of likelihood ratios test that determines p-value through Monte Carlo. Estimation under change point model is also included with model-robust confidence intervals for continuous threshold regression models (Fong, Y., Di, C., Huang, Y., Gilbert, P. (2017) <doi:10.1111/biom.12623>).

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2017-08-28 18:43:12 UTC

R topics documented:

chngpt ................................................................. 2
chngpt.test .......................................................... 2
chngptm ............................................................... 5
coef.0.ls ............................................................. 7
dat.mtct ............................................................ 10
sim.alphas ......................................................... 11
Description

Please see the Index link below for a list of available functions. The main testing function is `chngpt.test()`. The main estimation function is `chngptm()`.

Usage

```r
chngpt.test (formula.null, formula.chngpt, family=c("binomial","gaussian"), data,
  type=c("step","hinge","segmented","stegmented"),
  main.method=c("lr","score"),
  chngpts=NULL, lb.quantile=.1, ub.quantile=.9,
  chngpts.cnt=50, # this is set to 25 if int is weighted.two.sided or weighted.one.sided
  single.weight=1,
  mc.n=5e4,
  prob.weights=NULL,
  compute.p.value=TRUE,
  verbose=FALSE )
```

```r
antoch.test (formula, data, chngpt.var, plot.=FALSE)
```

## S3 method for class 'chngpt.test'
plot(x, by.percentile=TRUE, both=FALSE, main=NULL, ...)
```
Arguments

formula.null formula for the null model. The outcome needs to be a 0/1 variable.

formula.chngpt formula for the change point model. For example, suppose formula.null=y~z and we want to test whether I(x>cutoff) is a significant predictor, formula.chngpt=~x. If instead we are interested in testing the null that neither I(x>cutoff) nor z*I(x>cutoff) is a significant predictor, formula.chngpt=~x*z

data data frame.

family Currently only linear and logistic regression are supported.

type step: flat before and after change point; hinge: flat before and slope after change point; segmented: slope before and after change point

main.method method for testing main effects of some threshold model.

chngpts A grid of potential change points to maximize over. If not supplied, they will be set to a vector of length chngpt.cnt equally spaced between lb.quantile and ub.quantile.

lb.quantile number. The lower bound in the search for change point in the unit of quantile.

ub.quantile number. The upper bound in the search for change point in the unit of quantile.

chngpts.cnt integer. Number of potential change points to maximize over.

mc.n integer. Number of multivariate normal samples to generate in the Monte Carlo procedure to evaluate p-value.

single.weight numeric. Only used when interaction.method="weighted.single.arg".

prob.weights numeric. Not yet fully supported.

compute.p.value Boolean. Computing p values takes a lot of time. Sometimes we just want the maximal statistic and change point associated with it.

verbose Boolean.

chngpt.var string. Name of the predictor to detect change point

plot. Boolean. Whether to make a plot.

formula formula.

x An object of type chngpt.test.

... arguments passed to or from methods

by.percentile both

main

Details

If there are missing values in the chngpt formula, those rows will be removed from the whole dataset, including null model and chngpt model.

The test results, e.g. p-value, should not change if the same function is called twice, because internally seed to random number generator is set to 1 before MC and set back to the R state afterwards.

antoch.test is only implemented for main effect only and is based on Antoch et al. (2004). Also see Fong et al. (2014).
Value

A list of class htest and chngpt.test

- **p.value**: P-value
- **chngpts**: Vector of change points evaluated
- **TT**: Standardized absolute score statistics
- **V.S. hat**: Estimated variance-covariance matrix of the score statistics

References

- Fong, Y., Huang, Y., Gilbert, P. (2017) Estimation and hypothesis testing under four types of change point models using chngpt, BMC Bioinformatics, under review.

Examples

```r
dat = sim.chngpt("sigmoid", type="step", n=200, seed=1, beta=1, alpha=-1, x.distr="norm", e.=4)

test = chngpt.test(formula.null=y-z, formula.chngpt=-x, dat, type="step", family="binomial")
test
plot(test)

## Not run:
# not run because otherwise the examples take >5s and that is a problem for R CMD check
# has interaction

test = chngpt.test(formula.null=y-z, formula.chngpt=-x*z, dat, type="step")
test
plot(test)

## End(Not run)

#antoch.test(y-1, dat, "x.star")
```
Estimate change point logistic model

Usage

chngptm (formula.1, formula.2, family, data,
    type=c("step", "hinge", "segmented", "segmented2", "segmented"),
    est.method=c("default", "smoothapprox", "grid"), est.method.boot=c("grid", "smoothapprox"),
    var.type=c("none", "robust", "model", "smooth", "robusttruth", "bootstrap", "all"),
    aux.fit=NULL,
    test.inv.ci=TRUE, boot.test.inv.ci=TRUE,
    lb.quantile=.1, ub.quantile=.9, grid.search.max=500, ci.bootstrap.size=500, alpha=0.05,
    save.boot=FALSE, m.out.of.n=FALSE,
    b.transition=Inf,
    tol=1e-4, maxit=1e2, chngpt.init=NULL, search.bound=10,
    weights=NULL, verbose=FALSE,
    ...)

## S3 method for class 'chngptm'
coef(object, ...)
## S3 method for class 'chngptm'
vcov(object, ...)
## S3 method for class 'chngptm'
print(x, ...)
## S3 method for class 'chngptm'
summary(object, var.type=NULL, verbose=FALSE, ...)

Arguments

formula.1 The part of formula that is free of terms involving thresholded variables
formula.2 The part of formula that is only composed of thresholded variables
family string. coxph or any valid argument that can be passed to glm. But variance estimate is only available for binomial and gaussian (only model-based for latter)
data data frame.
type types of threshold effects. segmented2 differs from segmented in parameterization.
b.transition Numeric. Controls whether threshold model or smooth transition model. Default to Inf, which corresponds to threshold model
est.method string. Better leave it at NULL. grid: grid search; smoothapprox: smooth approximation
var.type string. Different methods for estimating covariance matrix and constructing confidence intervals
aux.fit a model fit object that is needed for model-robust estimation of covariance matrix
grid.search.max integer.
test.inv.ci Boolean, whether or not to find test-inversion confidence interval for threshold
ci.bootstrap.size integer, number of bootstrap
alpha double, normal type I error rate
save.boot Boolean, whether or not to save bootstrap samples
lb.quantile lower bound of the search range for change point estimate
ub.quantile upper bound of the search range for change point estimate
tol Numeric. Stopping criterion on the coefficient estimate.
maxit integer. Maximum number of iterations in the outer loop of optimization.
chngpt.init numeric. Initial value for the change point.
weights passed to glm
verbose Boolean.
x chngptm fit object.
object chngptm fit object.
... arguments passed to glm or coxph
m.out.of.n whether to perform m out of n bootstrap
est.method.boot estimation search method for bootstrap
boot.test.inv.ci whether to get test inversion CI under bootstrap
search.bound bounds for search for sloping parameters

Details
Without lb.quantile and ub.quantile, finite sample performance of estimator drops considerably!
When est.method is smoothapprox, Newton-Raphson is done with initial values chosen by change point hypothesis testing. The testing procedure may be less subjective to finite sample volatility.

Value
A an object of type chngptm with the following components
converged Boolean
coefficients vector. Estimated coefficients. The last element, named ".chngpt", is the estimated change point
test htest. Max score test results
iter integer. Number of iterations
References

Fong, Y., Huang, Y., Gilbert, P., Permar S. (2017) Estimation and hypothesis testing under four
types of change point models using chngpt, BMC Bioinformatics, under review.

regression models, Biometrics, 73(2):452-462.

mation in logistic regression. Statistics in Medicine. 22:13141

Examples

dat=sim.chngpt("sigmoid4", type="step", n=200, seed=1, beta=1, alpha=-1, x.distr="norm", e.=4)

fit.1=chngptm(formula.1=y~z, formula.2=~x, family="binomial", dat, type="step", est.method="grid")
print(fit.1)
summary(fit.1)

## Not run:
# not run because otherwise the examples take >5s and that is a problem for R CMD check
# has interaction

library(survival)
test1 <~ data.frame(time=c(4,3,1,1,2,2,3),
  status=c(1,1,0,1,1,0),
  x=c(0,2,1,1,0,0),
  sex=c(0,0,0,0,1,1))
fit=chngptm(formula.1=Surv(time, status)-1, formula.2=~x, data=test1, family="coxph", type="step")

## End(Not run)

---

coef.0.ls Simulation Study Parameters

Description

The true parameters used in the simulation studies.

Usage

data("coef.0.ls")
The format is: List of 3 $ segmented : List of 32 ..$ quadratic2b_norm : Named num [1:5] 0 1 0 0 0 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ cubic2b_lin : Named num [1:5] 0 1 7 0 0 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b_norm : Named num [1:5] -0.0943 0.3365 0.4005 -0.1625 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b1_norm : Named num [1:5] -0.0943 0.3365 0.4005 -0.2231 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b2_norm : Named num [1:5] -0.0943 0.3365 0.4005 -0.5108 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b3_norm : Named num [1:5] -0.0943 0.3365 0.4005 -0.0513 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b4_norm : Named num [1:5] -0.0943 0.3365 0.4005 -0.1054 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b5_norm : Named num [1:5] -0.0943 0.3365 0.4005 -0.2231 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b6_norm : Named num [1:5] -0.0943 0.3365 0.4005 -0.1625 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b7_norm : Named num [1:5] -0.0943 0.3365 0.4005 -0.0513 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b8_norm : Named num [1:5] -0.0943 0.3365 0.4005 -0.5108 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b9_norm : Named num [1:5] -0.0943 0.3365 0.4005 0.9163 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b10_norm : Named num [1:5] -0.0943 0.3365 0.4005 0.005 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b16_norm : Named num [1:5] -0.0943 0.3365 0.4005 0.0198 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b17_norm : Named num [1:5] -0.0943 0.3365 0.4005 0.0392 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2b18_norm : Named num [1:5] -0.0943 0.3365 0.4005 0.0583 4.5 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ smooth2b_norm : Named num [1:5] 0.3363 0.4 0.48 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ smooth2b1_norm : Named num [1:5] 0.3363 0.4 0.005 4.8 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ smooth2b2_norm : Named num [1:5] 0.3363 0.4 0.01 4.8 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ smooth2b3_norm : Named num [1:5] 0.3363 0.4 0.02 4.8 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ smooth2b4_norm : Named num [1:5] 0.3363 0.4 0.05 4.8 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ smooth2b5_norm : Named num [1:5] 0.3363 0.4 0.1 4.8 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2_gam : Named num [1:5] -1.3 0.3363 0.4 -0.9162 2.2 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2_gam1 : Named num [1:5] -0.6 0.3363 0.4 -0.9161 1 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ sigmoid2_gam2 : Named num [1:5] -0.6 0.3363 0.4 -0.9161 1 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ exp_gam : Named num [1:5] -2.753 0.336 0.513 0.936 3.607 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ quadratic_gam : Named num [1:5] -1.6355 0.3363 -0.0398 1.4869 2.8154 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ exp_norm : Named num [1:5] -6.235 0.337 1.012 1.325 5.057 .. .- attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" .. .- $ hinge : List of 29 ..$ quadratic2b_norm : Named num [1:5] 0 1 0 0 0 .. .- attr(*, "names")= 

..$ sigmoid2b2_norm : Named num [1:5] -0.0943 0.3365 0.4005 -0.1054 4.5 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ sigmoid2b3_norm : Named num [1:5] -0.0943 0.3365 0.4005 -0.0513 4.5 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ sigmoid2b4_norm : Named num [1:5] -0.0943 0.3365 0.4005 -0.5108 4.5 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ sigmoid2b5_norm : Named num [1:5] -0.0943 0.3365 0.4005 -0.1625 4.5 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ sigmoid2b6_norm : Named num [1:5] -0.0943 0.3365 0.4005 0.0198 4.5 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ sigmoid2b7_norm : Named num [1:5] -0.0943 0.3365 0.4005 0.0392 4.5 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ sigmoid2b8_norm : Named num [1:5] -0.0943 0.3365 0.4005 0.0583 4.5 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ sigmoid2b9_norm : Named num [1:5] -0.0943 0.3365 0.4005 0.9163 4.5 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ sigmoid2b10_norm: Named num [1:5] -0.0943 0.3365 0.4005 0 4.5 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ sigmoid2b16_norm: Named num [1:5] -0.0943 0.3365 0.4005 0.0198 4.5 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ sigmoid2b17_norm: Named num [1:5] -0.0943 0.3365 0.4005 0.0392 4.5 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ sigmoid2b18_norm: Named num [1:5] -0.0943 0.3365 0.4005 0.0583 4.5 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ smooth2b_norm : Named num [1:5] 0 0.336 0.4 0.48 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ smooth2b1_norm : Named num [1:5] 0 0.336 0.4 0.48 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ smooth2b2_norm : Named num [1:5] 0 0.336 0.4 0.48 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ smooth2b3_norm : Named num [1:5] 0 0.336 0.4 0.48 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ smooth2b4_norm : Named num [1:5] 0 0.336 0.4 0.48 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ smooth2b5_norm : Named num [1:5] 0 0.336 0.4 0.48 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ smooth2b6_norm : Named num [1:5] 0 0.336 0.4 0.48 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ smooth2b7_norm : Named num [1:5] 0 0.336 0.4 0.48 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... ..$ smooth2b8_norm : Named num [1:5] 0 0.336 0.4 0.48 ... - attr(*, "names")= chr [1:5] "(Intercept)" "z" "x" "(x-chngpt)+" ... An Example Dataset

**Description**

A dataset from the immune correlates study of Maternal To Child Transmission of HIV-1

**Usage**

data("dat.mtct")

**Format**

A data frame with 236 observations on the following 3 variables.

- **y** a numeric vector
- **birth** a factor with levels C-section Vaginal
- **NAb_SF162LS** a numeric vector
References

Fong, Y., Huang, Y., Gilbert, P. (2015) Estimation and hypothesis testing under four types of change point models using chngpt, under review.

Examples

data(dat.mtct)
### maybe str(dat.mtct); plot(dat.mtct) ...

---

**sim.alphas**

**Simulation Parameters**

**Description**

Simulation Parameters

**Usage**

```r
data(sim.alphas)
```

**Format**

List of 6. Names: sigmoid2_norm, sigmoid2_norm3, sigmoid3_norm, sigmoid3_norm3, sigmoid4_norm, sigmoid4_norm3. Each element is a 5x4 matrix

---

**sim.chngpt**

**Simulation Function**

**Description**

Generate simulation datasets for change point Monte Carlo studies.

**Usage**

```r
sim.chngpt,label=c("sigmoid2","sigmoid3","sigmoid4","sigmoid5","sigmoid6","quadratic", "quadratic2b","cubic2b","exp","flatHyperbolic"),
n, seed,
```
e.=NULL, b.transition=Inf,
beta=NULL, coef.z=log(1.4), alpha=NULL,
sd=0.3, mu=4.7, sd.x=NULL,
alpha.candidate=NULL, verbose=FALSE)

Arguments

label string. Simulate scenario, see details.
type string. Types of threshold effect to simulate, only applicable when label does not start with sigmoid.
family string. GIm family.
n
seed
beta

coef.z numeric. Coefficient for z.
alpha numeric, intercept.
mu numeric
sd.x numeric
x.distr string. Possible values: norm (normal distribution), gam (gamma distribution).

e.
verbose Boolean
b.transition
sd
alpha.candidate

candidate values of alpha, used in code to determine alpha values

Details

When label is "sigmoid1", an intercept only model is the data generative model. When label is "sigmoid2", a binary covariate z is also part of the data generative model.

Value

A data frame with following columns:

y 0/1 outcome
x observed covariate that we are interested in
x.star unobserved covariate that underlies x
z additional covariate

In addition, columns starting with 'w' are covariates that we also adjust in the model; columns starting with 'x' are covariates derived from x.
**Description**

Simulate data

**Usage**

```r
sim.my(n, seed, label, alpha, beta, e. = NULL, b. = NULL, tr. = NULL)
```

**Arguments**

- `n`: Sample size
- `seed`: Seed for random number generator
- `label`: A character string which specifies the simulation scenario. sigmoid4, sigmoidgam4, elbow4
- `alpha`: regression parameter
- `beta`: regression parameter
- `e.`: inflection point for the logistic transformation (the log scale)
- `b.`: slope for the logistic transformation
- `tr.`: threshold point

**Details**

When the label starts with elbow, the transformation on x.star is elbow shaped. When the label starts with sigmoid, the transformation on x.star is sigmoid shaped. Data simulated from \( \text{logit(Pr(Y==1))} = \alpha + \beta^*\text{transformed x.star} \).

**Value**

A data frame with columns: `y`, `x.star`, `x.star.expit` (if label starts with sigmoid), `x.star.tr` (if label starts with elbow), `x.bin.med` (x.star dichotomized at median), `x.tri` (x.star trichotomized at tertiles).

**Examples**

```r
alpha=-1; beta=log(0.2)
e.=5; b.=-30; t.=1
dat=sim.my(n=250, seed=1, label="sigmoid4", alpha, beta, e.=e., b.=b.)
```
sim.pastor

Simulate data according to one of the scenarios considered in Pastor-Barriuso et al 2003

Description

Simulate data according to one of the scenarios considered in Pastor-Barriuso et al 2003

Usage

```r
sim.pastor(seed)
```

Arguments

- `seed` Seed for the random number generator.

Value

A data frame with columns: y, x.star, x.star.expit, and x.bin.med (x.star dichotomized at median).

Examples

```r
dat=sim.pastor(seed=1)
```
Index

*Topic datasets
  dat.mtct, 10
*Topic distribution
  chngpt, 2

antoch.test (chngpt.test), 2
chngpt, 2
chngpt.test, 2
chngptm, 5
coef.0.1s, 7
coef.chngptm (chngptm), 5
data.mtct, 10
plot.chngpt.test (chngpt.test), 2
print.chngptm (chngptm), 5

sim.alphas, 11
sim.chngpt, 11
sim.my, 13
sim.pastor, 14
summary.chngptm (chngptm), 5
vcov.chngptm (chngptm), 5