

Package: ShapeChange (via r-universe)

September 3, 2024

Type Package

Title Change-Point Estimation using Shape-Restricted Splines

Version 1.5

Date 2023-09-06

Author Xiyue Liao and Mary C Meyer

Maintainer Xiyue Liao <xliao@sdsu.edu>

Description In a scatterplot where the response variable is Gaussian, Poisson or binomial, we consider the case in which the mean function is smooth with a change-point, which is a mode, an inflection point or a jump point. The main routine estimates the mean curve and the change-point as well using shape-restricted B-splines. An optional subroutine delivering a bootstrap confidence interval for the change-point is incorporated in the main routine.

License GPL (≥ 2)

Depends coneproj(≥ 1.11), quadprog($\geq 1.5-5$), R($\geq 3.2.0$)

NeedsCompilation no

Suggests stats, MASS, graphics, grDevices, utils

Repository CRAN

Date/Publication 2023-09-06 21:22:42 UTC

Contents

ShapeChange-package	2
changept	2
ip	8
jp	9
tp	11
Index	12

ShapeChange-package *Change-Point Estimation using Shape-Restricted Splines*

Description

Given a scatterplot of (x_i, y_i) , $i = 1, \dots, n$, where \mathbf{x} is the predictor and \mathbf{y} is the response which could be Gaussian, Poisson and binomial. The underlying mean curve in this scatterplot is denoted as $f_m(\mathbf{x})$, where $f_m(\mathbf{x})$ is a smooth curve with a change-point m which falls in one of the three categories:

- a mode in a unimodal smooth curve
- an inflection point in a convex-concave (concave-convex) smooth curve
- a jump point in an otherwise smooth curve.

Given the category of the change-point specified by the user, the main routine "changept" estimates the mean curve and the change-point as well using shape-restricted B-splines.

See [changept](#) for more details.

Details

Package: ShapeChange
Type: Package
Version: 1.3
Date: 2016-03-19
License: GPL (>= 2)

Author(s)

Xiyue Liao and Mary C. Meyer

Maintainer: Xiyue Liao <xiyue@rams.colostate.edu>

changept *Change-Point Estimation using Shape-Restricted Splines*

Description

'changept' is used to estimate the location of the change-point in the underlying mean curve of a scatterplot and the curve as well using shape-restricted B-splines. The change-point falls in one of the three categories: a mode, an inflection point and a jump point.

Usage

```
changept(formula, family = gaussian(), data = NULL, k = NULL, knots = NULL,
  fir = FALSE, q = 3, pnt = FALSE, pen = 0, arp = FALSE, ci = FALSE, nloop = 1e+3,
  constr = TRUE, param = TRUE, gcv = FALSE, spl = NULL, dmat = NULL, x1 = NULL, xn = NULL)
```

Arguments

formula	A formula object which gives a symbolic description of the model to be fitted. It has the form "response ~ predictor". The response is a vector of length n and it is from one of the three exponential families: Gaussian, Poisson and binomial. The predictor is a nonparametrically modelled variable with a shape restriction. The user is supposed to indicate the relationship between the underlying mean curve $f_m(x)$ and the predictor x in the following way: Assume that $f_m(x)$ is the underlying mean curve and x is the predictor: <ul style="list-style-type: none"> • $tp(x, sh = 1)$: $f_m(x)$ is unimodal (increasing-decreasing) with a mode at m. See tp for more details. • $ip(x, sh = 1)$: $f_m(x)$ is convex-concave with an inflection point at m. See ip for more details. • $jp(x, trd1 = -1, trd2 = -1, up = TRUE)$: $f_m(x)$ has a jump point at m and smooth otherwise. It is non-increasing in a subinterval which is made of two consecutive knots and contains the jump point m. $t_p \leq m \leq t_{p+1}$, $p = 1, \dots, k - 1$ and k is the number of knots. See jp for more details.
family	A parameter indicating the error distribution and link function to be used in the model. It can be a character string naming a family function or the result of a call to a family function. This is borrowed from the glm routine in the stats package. There are three families used in changept: Gaussian, Poisson and binomial.
data	An optional data frame, list or environment containing the variables in the model. The default is data = NULL.
k	The number of knots. When it is NULL, k will be determined inside the changept routine according to the length of the predictor x ; otherwise, the user-defined k is used to create the knots. The default is NULL.
knots	The knots used to smoothly constrain a predictor. When it is NULL, a vector of knots will be created inside the changept routine; otherwise, the user-defined knots are used. The default is NULL. If both k and knots are defined by the user, only the knots parameter will be used.
fir	The parameter is used only when the change-point is an inflection point. When fir is TRUE, the estimated curve is constrained to be increasing at two end points. This is useful given the a priori information that the underlying mean curve is increasing over the whole interval, e.g., a growth curve. The default is FALSE.
q	The degree of penalty. It can be 1, 2 or 3. The default is $q = 3$.
pnt	Logical flag indicating if or not penalized B-splines are used. When pnt is TRUE, penalized B-splines are used; otherwise, unpenalized B-splines are used. The default is pnt = FALSE.
pen	The penalty term when penalized B-splines are used. It can only be a non-negative real number. When the user chooses penalized B-splines, the penalty

	term is determined inside the main routine if it is not defined by the user; otherwise, the user-defined value is used. The default is $\text{pen} = 0$.
arp	Logical flag indicating if or not the error term is from a stationary autoregressive process of order p , where $p \in \{1, 2, 3, 4\}$. When arp is TRUE, the error term is assumed to follow an AR(p) process; otherwise, it is assumed to be Gaussian. The default is $\text{arp} = \text{FALSE}$.
ci	Logical flag. If ci is TRUE, a 95% bootstrap confidence interval of m will be delivered. The default is $\text{ci} = \text{FALSE}$.
nloop	The number of simulations used to get a bootstrap confidence interval for the change-point m when ci is TRUE. The default is $\text{nloop} = 1e+3$.
constr	Logical flag. It is used when m is a jump point. If constr is FALSE, no shape constraint is imposed on the estimated curve; otherwise, a shape constraint is imposed on the estimated curve in a subinterval which is made of two consecutive knots and contains the jump point m . $t_p \leq m \leq t_{p+1}$, $p = 1, \dots, k - 1$ and k is the number of knots. See jp for more details. The default is $\text{constr} = \text{TRUE}$.
param	Logical flag indicating if or not parametric bootstrapping is used. When param is TRUE, parametric bootstrapping is used; otherwise, nonparametric bootstrapping is used. The default is $\text{param} = \text{TRUE}$.
gcv	Logical flag indicating if or not the penalty term may be chosen through generalized cross-validation (GCV) methods when $\text{pen} = \text{TRUE}$. The default is $\text{gcv} = \text{FALSE}$. See Meyer, M.C. (2012) Constrained Penalized Splines. <i>Canadian Journal of Statistics</i> 40(1) , 190–206 for more details.
spl	This parameter is of no interest to the user.
dmat	This parameter is of no interest to the user.
x1	This parameter is of no interest to the user.
xn	This parameter is of no interest to the user.

Details

Given a scatterplot of (x_i, y_i) , $i = 1, \dots, n$, where \mathbf{x} is the predictor and \mathbf{y} is the response which could be Gaussian, Poisson and binomial. The underlying mean curve in this scatterplot is denoted as $f_m(\mathbf{x})$, where $f_m(\mathbf{x})$ is a smooth curve with a change-point m which falls in one of the three categories:

- a mode in a unimodal smooth curve
- an inflection point in a convex-concave (concave-convex) smooth curve
- a jump point in an otherwise smooth curve.

For the unimodal case, quadratic B-splines are used; for the inflection-point case, cubic B-splines are used; for the jump-point case, quadratic B-splines along with two extra basis functions, i.e., the ramp basis function and the jump basis function are used. The ramp basis function ensures that the slope of the estimated mean curve on the left of the jump point could be different from that on the right of the jump point. The jump basis function ensures that the estimated mean curve has an upward or downward jump.

In each case, the user can choose unpenalized or penalized regression. For the unimodal case, the penalized regression is strongly suggested for its smoothness and robustness in terms

of the number and the placement of knots, especially when the response is binomial; for the jump-point case, the user can choose to make a shape-restricted fit or not by setting the parameter `constr` to be `TRUE` or `FALSE`. If `constr` is `TRUE`, a jump point will be detected and the monotonicity of the mean curve will be constrained in a subinterval which is made of two consecutive knots and contains the jump point m ; otherwise, a jump point will be detected without any shape restriction on the mean curve.

A parametrically modelled categorical covariate can be included in the model for the unimodal case and the jump-point case. However, it cannot be included in the model for the inflection-point case if the response is binomial, since the problem of estimating an inflection point is not well-defined then.

In addition, the user can get a 95% bootstrap confidence interval of m by setting the logical parameter `ci` to be `TRUE`. The default number of simulations to get such a confidence interval is `1e+3`, which is defined by the parameter `nloop`. The genetic routine `plot` can be used to show some estimated results including $\hat{f}_{\hat{m}}$, \hat{m} and a 95% bootstrap confidence interval of m if it is available.

See references cited in this section for more details.

Value

<code>chpt</code>	The estimated change-point \hat{m} .
<code>knots</code>	The knots used in the regression.
<code>fhat</code>	The estimated mean curve $\hat{f}_{\hat{m}}$.
<code>fhat_x</code>	The estimated mean curve $\hat{f}_{\hat{m}}$ in the constrained predictor.
<code>fhat_eta</code>	The estimated systematic component given that the response is binomial or Poisson and the change-point is a mode or a jump point.
<code>fhat_eta_x</code>	The estimated systematic component in the constrained predictor given that the response is binomial or Poisson and the change-point is a mode or a jump point.
<code>coefs</code>	The estimated coefficients of the B-spline basis functions for the shape-restricted predictor and the matrix whose columns represent the parametrically modelled categorical covariate if the user includes any in the model.
<code>zcoefs</code>	The estimated coefficients of the matrix whose columns represent the parametrically modelled categorical covariate if the user includes any in the model.
<code>cibt</code>	A 95% bootstrap confidence interval of m given that the parameter <code>ci</code> is <code>TRUE</code> .
<code>categ</code>	The category of the change-point to be estimated. Three categories are included, i.e., a mode, an inflection point and a jump point.
<code>sh</code>	The <code>sh</code> parameter defined in the symbolic routine <code>tp</code> or <code>ip</code> in a <code>changept</code> formula.
<code>x</code>	The shape-restricted predictor vector.
<code>y</code>	The response vector.
<code>xnm</code>	The name of the shape-restricted predictor.
<code>znm</code>	The name of the parametrically modelled categorical covariate.
<code>ynm</code>	The name of the response variable.
<code>m_i</code>	The centering value for the ramp edge in the jump-point case.
<code>pos</code>	The position of the knot t_p such that $t_p \leq m \leq t_{p+1}$, $p = 1, \dots, k - 1$ and k is the number of knots.

sub	The subinterval in which constrained regression is made for the jump-point case. It is of the form $[t_p, t_{p+1}]$. $t_p \leq m \leq t_{p+1}$, $p = 1, \dots, k-1$ and k is the number of knots.
family	The family parameter in a changept formula.
wt.iter	Logical flag indicating if or not iteratively re-weighted cone projections may be used. If the response is Gaussian, then wt.iter = FALSE; if the response is binomial or Poisson, then wt.iter = TRUE.
zmat	A matrix whose columns represent the parametrically modelled categorical covariate if the user includes any in the model. The user can choose to include a constant vector in it or not. It must be of full column rank.
vals	A vector storing the smallest value of each categorical variable defined as a factor(matrix).
is_fac	Logical flag showing if a categorical variable is a factor(matrix) or not.
zid	A vector keeping track of the position of the categorical variable in a changept formula.
zid1	A vector keeping track of the beginning position of the levels(columns) of the categorical variable defined as a factor(matrix).
zid2	A vector keeping track of the end position of the levels(columns) of the categorical variable defined as a factor(matrix).
tms	The terms object extracted by the generic function terms from a changept fit. See the official help page (http://stat.ethz.ch/R-manual/R-patched/library/stats/html/terms.html) of the terms function for more details.
msbt	A bootstrap distribution of m given that the parameter ci is TRUE.
bmat	A matrix whose columns are the B-spline basis functions for the shape-restricted predictor.
phi	The estimated vector of autoregressive coefficients when the error term is assumed to follow an AR(p) process.
sig	The estimated variance of the white noise of the error term when the error term is assumed to follow an AR(p) process.
aics	A matrix of AIC values delivered when the error term is assumed to follow an AR(p) process.
lambda	The penalty term chosen when penalized B-splines are used.
edfc	The constrained effective degrees of freedom in the final fit when penalized B-splines are used.
edfs	A vector of unconstrained effective degrees of freedom when penalized B-splines are used.
lams	A vector of penalty terms corresponding to a vector of unconstrained effective degrees of freedom when penalized B-splines are used.
phisbt	The bootstrap distribution of the vector of estimated autoregressive coefficients when the error term is assumed to follow an AR(p) process.
sigibt	The bootstrap distribution of the estimated variance of the white noise of the error term when the error term is assumed to follow an AR(p) process.

Author(s)

Xiyue Liao and Mary C Meyer

References

- Meyer, M. C. (2008) Inference using shape-restricted regression splines. *Annals of Applied Statistics* **2**(3), 1013–1033.
- Meyer, M.C. (2012) Constrained Penalized Splines. *Canadian Journal of Statistics* **40**(1), 190–206.
- Meyer, M. C. (2013b) A simple new algorithm for quadratic programming with applications in statistics. *Communications in Statistics* **42**(5), 1126–1139.
- Liao, X. and M. C. Meyer (2014) coneproj: An R package for the primal or dual cone projections with routines for constrained regression. *Journal of Statistical Software* **61**(12), 1–22.
- Wang, H., Meyer, M.C., and Opsomer, J.D. (2013) Constrained Spline Regression in the Presence of AR(p) Errors. *Journal of Nonparametric Statistics* **25**(4), 809–827.

See Also

[tp](#), [ip](#), [jp](#)

Examples

```
# Example 1: the response is normal
# the underlying mean curve has an inflection point at .5 and it is concave-convex
n = 100
inflect = .5
x = seq(1/n, 1, length = n)

set.seed(123)
y = 50 * (x - inflect)^3 + rnorm(n, sd = 1)
ans = changept(y ~ ip(x, sh = -1))

plot(ans)

# Example 2: the response is binomial
# the underlying mean curve has an inflection point at .5 and it is convex-concave
n = 100
x = seq(1/n, 1, length = n)
mu = exp(8 * x - 4) / (1 + exp(8 * x - 4))

set.seed(123)
y = rbinom(n, size = 1, prob = mu)
ans = changept(y ~ ip(x), family = binomial())

plot(ans)

# Example 3: the response is normal
# the underlying mean curve is unimodal with a mode at .5
n = 100
SD = .1
x = seq(1/n, 1, length = n)
```

```

mode = .5

set.seed(123)
y = -6 * (x - .5)^2 + rnorm(n, sd = SD)
ans = changept(y ~ tp(x))

plot(ans)

# Example 4: the response is binomial
# the underlying mean curve is unimodal with a mode at .5
n = 200
x = seq(1/n, 1, length = n)
mu = .9 - 2.5 * (x - .5)^2

set.seed(123)
y = rbinom(n, size = 1, prob = mu)
ans = changept(y ~ tp(x), family = binomial(), k = 10, pnt = TRUE)

plot(ans)

# Example 5:
library(datasets)
# Nile River Data Set: Nile river flow is the response
# an abrupt downward jump in the river flow occurred around the year 1898
# the river flow is non-decreasing on both sides of the jump point
data(Nile)
yrs = 1871:1970
ans = changept(Nile ~ jp(yrs, up = FALSE, trd1 = 1, trd2 = 1), data = Nile)

plot(ans)

```

ip

Specify an Inflection-Point Category in a CHANGEPT Formula

Description

A symbolic routine to define that the underlying mean curve has an inflection-point in the formula argument to `changept`.

Usage

```
ip(x, sh = 1)
```

Arguments

x	The predictor vector.
sh	If sh is 1, then the estimated curve is convex-concave; if sh is -1, it is concave-convex. Note that when the response is binomial or Poisson, sh is always 1. The default is sh = 1.

Value

The vector "x" with three attributes, i.e., nm: the name of x; categ: the category of the change-point, "inflect"; sh: the shape constraint on the estimated curve: 1 (convex-concave) or -1 (concave-convex).

The nm and categ attributes are used in the plot routine; the sh attribute is used to set up a shape-constrained regression.

Author(s)

Xiyue Liao

See Also

[tp](#), [jp](#)

Examples

```
# the underlying mean curve is a non-decreasing growth curve
# with an inflection point at .5 and it is convex-concave
n = 100
x = seq(1/n, 1, length = n)

set.seed(123)
y = 5 * (1 + tanh(10 * (x - .5))) + rnorm(n, sd = 1)
ans = changept(y ~ ip(x, sh = 1), fir = TRUE)

plot(ans)
```

jp

Specify a Jump-Point Category in a CHANGEPT Formula

Description

A symbolic routine to define that the underlying mean curve has a jump point in the formula argument to changept.

Usage

```
jp(x, trd1 = -1, trd2 = -1, up = TRUE)
```

Arguments

x The predictor variable.

trd1 If trd1 is -1 (1), then the estimated curve is constrained to be non-increasing (non-decreasing) on the left of the estimated jump point. The default is trd1 = -1.

trd2	If trd2 is -1 (1), then the estimated curve is constrained to be non-increasing (non-decreasing) on the right of the estimated jump point. The default is trd2 = -1 .
up	The jump direction at the jump point. The default is up = TRUE, i.e., there is an upward jump; otherwise, there is a downward jump.

Details

For the jump-point case, the user can choose either to impose a shape constraint on the estimated curve or not. If no shape constraint is imposed, only a jump point will be detected by the `changept` routine; otherwise, a jump point will be detected and the shape of the curve will be constrained according to the parameters `trd1` and `trd2` in a subinterval which is made of two consecutive knots and contains the jump point.

Value

The vector "x" with five attributes, i.e., `nm`: the name of x; `categ`: the category of the change-point, "jump"; `trd1` and `trd2`: -1 (non-increasing) or 1 (non-decreasing); `up`: the jump direction.

The `nm` and `categ` attributes are used in the plot routine; the `trd1`, `trd2`, and `up` attributes are used to set up a shape-constrained regression.

Author(s)

Xiyue Liao

See Also

[tp](#), [ip](#)

Examples

```
## Not run:
# simulated curve with an upward jump at .7
n = 1000
x = seq(1/n, 1, length = n)

set.seed(123)
y = 4 * sin(5 * x) + 3 * x + (x >= .7) + rnorm(n, sd = 1)
ans = changept(y ~ jp(x))

plot(ans)

## End(Not run)
```

tp *Specify a Unimodal Category in a CHANGEPT Formula*

Description

A symbolic routine to define that the underlying mean curve is unimodal, i.e., it has a turn-around point, in the formula argument to `changept`.

Usage

```
tp(x, sh = 1)
```

Arguments

x	The predictor vector.
sh	If sh is 1, then the estimated curve is increasing-decreasing; if sh is -1 , it is decreasing-increasing.

Value

"tp" returns the vector "x" with three attributes, i.e., nm: the name of x; categ: the category of the change-point, "mode"; sh: the shape constraint on the estimated curve: 1 (increasing-decreasing) or -1 (decreasing-increasing).

The nm and categ attributes are used in the plot routine; the sh attribute is used to set up a shape-constrained regression.

Author(s)

Xiyue Liao

See Also

[ip](#), [jp](#)

Examples

```
# the underlying mean curve is unimodal with a mode at .8
n = 100
x = seq(1/n, 1, length = n)

# a categorical covariate z with two levels (0 and 1)
z = rep(0:1, 50)

set.seed(123)
y = 30 * x^4 * (1 - x) + (z == 1) * .5 + rnorm(n, sd = 1)
ans = changept(y ~ tp(x) + factor(z), family = gaussian())

plot(ans)
legend("topleft", "constrained fit for z == 1", bty = "n", col = "skyblue", lty = 2, lwd = 2)
```

Index

* **changept**

changept, [2](#)

* **package**

ShapeChange-package, [2](#)

* **symbolic routine**

ip, [8](#)

jp, [9](#)

tp, [11](#)

changept, [2](#), [2](#)

ip, [3](#), [5](#), [7](#), [8](#), [10](#), [11](#)

jp, [3](#), [4](#), [7](#), [9](#), [9](#), [11](#)

ShapeChange (ShapeChange-package), [2](#)

ShapeChange-package, [2](#)

tp, [3](#), [5](#), [7](#), [9](#), [10](#), [11](#)