# Package 'SplinesUtils'

December 16, 2020

**Version** 0.2

**Date** 2020-12-16

**Title** Utility functions for univariate cubic interpolation splines, smoothing splines and regression splines

**Author** Zheyuan Li <zheyuan.li@bath.edu>

**Maintainer** Zheyuan Li <zheyuan.li@bath.edu>

**Depends** R (>= 3.4.0), splines

**Imports** stats, graphics

**Description** Splines that are used in regression or smoothing models often adopt basis representation. While such representation is recommended for numerical computations, it is neither intuitive to interpret, nor easy to derive mathematical property of the spline. Package ``SplinesUtils" provides functions that reparametrize univariate cubic interpolation splines, smoothing splines and regression splines (used in ``lm", ``glm", ``lme") to piecewise polynomials, by exporting piecewise polynomial coefficients as a matrix and print piecewise polynomial equations as formatted strings. The package also provides generic functions like 1) plot() and predict() for plotting and predicting a spline or its derivatives; 2) solve() for solving a spline or its derivatives for a given value, specially useful for identifying local extrama of the spline.

**License** GPL-3

**NeedsCompilation** no

## R topics documented:

---

`CubicInterpSplineAsPiecePoly`
*Construct a cubic interpolation spline as piecewise cubic polynomials.*

---

## Description

The function is a thin wrapper of `stats::splinefun`. The latter readily constructs the interpolation spline as piecewise cubic polynomials but stores construction data like knots and piecewise polynomial coefficients in an environment. The Function reorganizes such information into a "PiecePoly" object.

## Usage

```
CubicInterpSplineAsPiecePoly(x, y, method)
```

## Arguments

| | |
|---|---|
| `x` | A vector of $x$-coordinate values of the data points to interpolate. |
| `y` | A vector of $y$-coordinate values of the data points to interpolate. |
| `method` | Methods for cubic spline interpolation, including “fmm”, “natural”, “periodic” and “hyman” in `stats::spline` and `stats::splinefun`. |

## Value

An object of class “PiecePoly” and “CubicInterpSpline” classes. See PiecePolyObject. Note that the piecewise polynomial is always parametrized in shifted form.

## Author(s)

Zheyuan Li <zheyuan.li@bath.edu>

## Examples

```
## Not run:
## a toy dataset
set.seed(0)
## data for "fmm" and "natural" method
x <- 1:10 + runif(10, -0.1, 0.1)
y1 <- rnorm(10, 3, 1)
## periodic data for "periodic" method
x2 <- c(x, x[10] + runif(1))
y2 <- c(y1, y1[1])
## monotone data for "hyman" method
y3 <- sort(y1)

## perform interpolation
fmm <- CubicInterpSplineAsPiecePoly(x, y1, "fmm")
```

```
natural <- CubicInterpSplineAsPiecePoly(x, y1, "natural")
periodic <- CubicInterpSplineAsPiecePoly(x2, y2, "periodic")
hyman <- CubicInterpSplineAsPiecePoly(x, y3, "hyman")

## test print method
fmm
natural
periodic
hyman

## export all polynomial equations as formatted strings
fmm_eqn <- summary(fmm)
natural_eqn <- summary(natural)
periodic_eqn <- summary(periodic)
hyman_eqn <- summary(hyman)

## plot all splines
par(mfrow = c(2, 2), mar = c(4, 4, 1, 1))
plot(fmm, show.knots = TRUE); points(x, y1)
plot(natural, show.knots = TRUE); points(x, y1)
plot(periodic, show.knots = TRUE); points(x2, y2)
plot(hyman, show.knots = TRUE); points(x, y3)

## plot 1st derivatives of all splines
par(mfrow = c(2, 2), mar = c(4, 4, 1, 1))
plot(fmm, deriv = 1)
plot(natural, deriv = 1)
plot(periodic, deriv = 1)
plot(hyman, deriv = 1)

## backsolve the spline given y = 2.85
xr1 <- solve(fmm, b = 2.85)
xr2 <- solve(natural, b = 2.85)
xr3 <- solve(periodic, b = 2.85)
xr4 <- solve(hyman, b = 2.85)
par(mfrow = c(2, 2), mar = c(4, 4, 1, 1))
plot(fmm); abline(h = 2.85, lty = 2); points(xr1, rep.int(2.85, length(xr1)))
plot(natural); abline(h = 2.85, lty = 2); points(xr2, rep.int(2.85, length(xr2)))
plot(periodic); abline(h = 2.85, lty = 2); points(xr3, rep.int(2.85, length(xr3)))
plot(hyman); abline(h = 2.85, lty = 2); points(xr4, rep.int(2.85, length(xr4)))

## find all extrema
## "hyman" excluded as it is monotonic spline
xs1 <- solve(fmm, deriv = 1)
xs2 <- solve(natural, deriv = 1)
xs3 <- solve(periodic, deriv = 1)
par(mfrow = c(2, 2), mar = c(4, 4, 1, 1))
ys1 <- predict(fmm, xs1)
ys2 <- predict(natural, xs2)
ys3 <- predict(periodic, xs3)
plot(fmm); points(xs1, ys1, pch = 19)
plot(natural); points(xs2, ys2, pch = 19)
plot(periodic); points(xs3, ys3, pch = 19)
```

```
## End(Not run)
```

---

`PiecePolyObject` *Ojbect of "PiecePoly" class.*

---

## Description

`CubicInterpSplineAsPiecePoly`, `SmoothSplineAsPiecePoly` and `RegSplineAsPiecePoly` return objects of "PiecePoly" class, for which S3 methods `summary`, `print`, `plot`, `predict` and `solve` are defined.

## Details

A spline $f(x)$ of degree $d$ and $k$ knots $x_1,\ x_2, \ldots,\ x_k$, is number of piecewise polynomials of degree $d$ that are $(d-1)$ times differentiable at the knots. For the $i$-th piece on interval $[x_i,\ x_{i+1}]$, the polynomial $f_i(x)$ is often parametrized in a shifted form

$$f_i(x) = b_{0i} + b_{1i}(x - x_i) + b_{2i}(x - x_i)^2 + \ldots + b_{di}(x - x_i)^d.$$

This form is appealing because $b_{0i}$ is just the value of the spline at knot $x_i$. It also facilitates integrating the spline. However, an unshifted version can also be exported

$$f_i(x) = a_{0i} + a_{1i}x + a_{2i}x^2 + \ldots + a_{di}x^d.$$

Piecewise polynomial representation is conveniently used in interpolation problems by `stats::spline` and `stats::splinefun`. However, B-splines representation is often used in regression and smoothing problems, for example by `splines::bs`, `splines::ns` and `stats::smooth.spline`. Coefficients for B-spline basis are difficult to interpret, so it might be helpful to reparametrize a fitted regression spline or smoothing spline into its conventional piecewise polynomial form.

Package SplinesUtils conducts such reparametrization by exact least squares fitting. $d+1$ evenly spaced points are taken on $[x_i,\ x_{i+1}]$, where spline values are predicted. Given those $(x,\ y)$ data, a polynomial of degree $d$ in either shifted or non-shifted form can be exactly fitted. The resulting piecewise polynomial coefficients $b_{0i},\ b_{1i}, \ldots,\ b_{di}$ or $a_{0i},\ a_{1i}, \ldots,\ a_{di}$ are stored in the $i$-th column of a $(d+1) \times (k-1)$ coefficient matrix.

The piecewise polynomial representation makes evaluation of splines and its derivatives very easy. It is also straightforward to solve equations like $f(x) = y$ and $f'(x) = y$ for $x$ values given a $y$ value. In particular, for $y = 0$ the solutions to two equations respectively give the roots and extrema of the spline.

## Value

An object of "PiecePoly" class is a list with at least the following two components:

| | |
|---|---|
| `PiecePoly` | A list with two components: `coef` and `shift`. The former is a $(d+1) \times (k-1)$ matrix, and the latter is a logical value. |
| `knots` | A vector of knots of the spline. |

`CubicInterpSplineAsPiecePoly` and `SmoothSplineAsPiecePoly` add an extra "CubicInterpSpline" class, while `RegSplineAsPiecePoly` adds and extra "RegSpline" class.

### Author(s)

Zheyuan Li <zheyuan.li@bath.edu>

---

`plot.PiecePoly` *"plot" method for "PiecePoly" class.*

---

### Description

The function plots piecewise polynomials or its derivatives.

### Usage

```
## S3 method for class 'PiecePoly'
plot(x, spread = 3L, deriv = 0L, show.knots = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `x` | A "PiecePoly" object. |
| `spread` | A graphical parameter determining how dense the evaluation grid is set up for plotting. Note that this is a multiplier factor. For a spline of degree $d$, there will be `(spread * d)` evenly spaced points between any two nearby knots. |
| `deriv` | Derivatives of the spline (here represented as piecewise polynomials) can be plotted. Much interest lies in the 1st and 2nd derivative, so while higher derivative can be computed, it is not performed by the functions. |
| `show.knots` | If `TRUE`, vertical dotted lines will be drawn through knots of the spline. Note that most spline constructor functions would place knots by quantiles of the $x$-coordinate values. |
| `...` | Not used by the method. |

### Value

The function invisibly returns $(x, y)$ data used for plotting in a list.

### Author(s)

Zheyuan Li <zheyuan.li@bath.edu>

`predict.PiecePoly` *"predict" method for "PiecePoly" class.*

## Description

The function evaluates piecewise polynomials or its derivatives at a vector of $x$-coordinate values.

## Usage

```
## S3 method for class 'PiecePoly'
predict(object, newx, deriv = 0L, ...)
```

## Arguments

| | |
|---|---|
| `object` | A "PiecePoly" object. |
| `newx` | A vector of x-coordinate values where the piecewise polynomials or its derivatives is evaluated. |
| `deriv` | Derivatives of the piecewise polynomials can be evaluated. Much interest lies in the 1st or 2nd derivative, so while a higher derivative can be computed, it is not performed by the function. |
| `...` | Not used by the method. |

## Value

A vector of evaluated $y$-coordinate values.

## Author(s)

Zheyuan Li <zheyuan.li@bath.edu>

`print.PiecePoly` *"print" method for "PiecePoly" class.*

## Description

The function hides contents of a "PiecePoly" object when printing it. It calls `summary.PiecePoly` to export up to 6 piecewise polynomial equations as formatted strings and prints them onto standard output for a sample display.

## Usage

```
## S3 method for class 'PiecePoly'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | A "PiecePoly" object. |
| ... | Not used by the method. |

### Value

The function returns nothing.

### Author(s)

Zheyuan Li <zheyuan.li@bath.edu>

---

## RegBsplineAsPiecePoly *This function is now superseded by* RegSplineAsPiecePoly.

---

### Description

This function is now superseded by RegSplineAsPiecePoly.

### Usage

```
RegBsplineAsPiecePoly(lm_glm, SplineTerm, shift = TRUE)
```

### Arguments

| | |
|---|---|
| lm_glm | A fitted "lm" or "glm" that has at least one spline term constructed by splines::bs or splines:ns. |
| SplineTerm | A character string giving the name of the spline term to be exported. |
| shift | A logical value determining whether the piecewise polynomial should be parametrized in shifted form. |

### Value

An object of "PiecePoly" and "RegBspline" classes. See PiecePolyObject.

### Author(s)

Zheyuan Li <zheyuan.li@bath.edu>

## Examples

```
## Not run:
## a toy data set
set.seed(0)
## a curve of 'x' (note that my 'x' are not uniformly sampled)
x <- sort(rnorm(400, 0, pi))
fx <- sin(x) + 0.2 * x + cos(abs(x))
x_range <- range(x)
## a 2-level grouping variable (1st group with mean 0 and 2nd group with mean 1)
g <- sample(gl(2, 200, labels = letters[1:2]))
fg <- c(0, 1)[g]
## a linear effect
u <- runif(400, x_range[1], x_range[2])
fu <- -0.05 * u
## a spurious covariate
z <- runif(400, x_range[1], x_range[2])
## an additive model
y <- fx + fg + fu + rnorm(400, 0, 0.5)

## splines are constructed so that boundary knots are clamped
## df = length(knots) + degree + has.intercept
## for 'ns()', degree = 3L is fixed

## fit a linear model
fit <- lm(y ~ g + bs(x, df = 20) + u + ns(z, df = 10))

## export spline term "bs(x, df = 20)"
spl_bs <- RegBsplineAsPiecePoly(fit, "bs(x, df = 20)")

## predict the spline at observed points
y_bs <- predict(spl_bs, x)
## not that this is not the fitted values
all.equal(y_bs, fitted(fit), check.attributes = FALSE)
#[1] "Mean relative difference: 0.8498324"
## it is identical to the following, up to a constant vertical shift
## (`predict.lm` applies term-wise centering, see https://stackoverflow.com/q/37963904/4891738)
y_bs2 <- predict(fit, type = "terms", terms = "bs(x, df = 20)")
## the standard error of the difference is 0, i.e., the difference is constant
c(sd(y_bs - y_bs2))
#[1] 9.727983e-15

## plot the spline
plot(spl_bs)

## find all its local extrema
xs <- solve(spl_bs, b = 0, deriv = 1)
ys <- predict(spl_bs, xs)
points(xs, ys, pch = 19)

## End(Not run)
```

RegSplineAsPiecePoly *Reparametrize a fitted bs() or ns() term in a linear model ("lm"), generalized linear model ("glm"), or linear mixed model ("lme") as piecewise polynomials.*

## Description

Reparametrize a single fitted bs() or ns() term in a linear model ("lm"), generalized linear model ("glm"), or linear mixed model ("lme") as piecewise polynomials. Do not interpret the spline as fitted values or linear predictors of the model.

## Usage

```
RegSplineAsPiecePoly(RegModel, SplineTerm, shift = TRUE)
```

## Arguments

| | |
|---|---|
| RegModel | A fitted "lm", "glm" or "lme" that has at least one spline term constructed by splines::bs or splines:ns. |
| SplineTerm | A character string giving the name of the spline term to be exported. |
| shift | A logical value determining whether the piecewise polynomial should be parametrized in shifted form. |

## Value

An object of "PiecePoly" and "RegSpline" classes. See PiecePolyObject.

## Author(s)

Zheyuan Li <zheyuan.li@bath.edu>

## Examples

```
## Not run:
## a toy data set
set.seed(0)
## a curve of 'x' (note that my 'x' are not uniformly sampled)
x <- sort(rnorm(400, 0, pi))
fx <- sin(x) + 0.2 * x + cos(abs(x))
x_range <- range(x)
## a 2-level grouping variable (1st group with mean 0 and 2nd group with mean 1)
g <- sample(gl(2, 200, labels = letters[1:2]))
fg <- c(0, 1)[g]
## a linear effect
u <- runif(400, x_range[1], x_range[2])
fu <- -0.05 * u
## a spurious covariate
z <- runif(400, x_range[1], x_range[2])
```

```
## an additive model
y <- fx + fg + fu + rnorm(400, 0, 0.5)

## splines are constructed so that boundary knots are clamped
## df = length(knots) + degree + has.intercept
## for 'ns()', degree = 3L is fixed

## fit a linear model
fit <- lm(y ~ g + bs(x, df = 20) + u + ns(z, df = 10))

## export spline term "bs(x, df = 20)"
spl_bs <- RegSplineAsPiecePoly(fit, "bs(x, df = 20)")

## predict the spline at observed points
y_bs <- predict(spl_bs, x)
## not that this is not the fitted values
all.equal(y_bs, fitted(fit), check.attributes = FALSE)
#[1] "Mean relative difference: 0.8498324"
## it is identical to the following, up to a constant vertical shift
## (`predict.lm` applies term-wise centering, see https://stackoverflow.com/q/37963904/4891738)
y_bs2 <- predict(fit, type = "terms", terms = "bs(x, df = 20)")
## the standard error of the difference is 0, i.e., the difference is constant
c(sd(y_bs - y_bs2))
#[1] 9.727983e-15

## plot the spline
plot(spl_bs)

## find all its local extrema
xs <- solve(spl_bs, b = 0, deriv = 1)
ys <- predict(spl_bs, xs)
points(xs, ys, pch = 19)

## End(Not run)
```

---

`SmoothSplineAsPiecePoly` *Export a fitted smoothing spline as piecewise cubic polynomials.*

---

## Description

The function extracts knots from a fitted smoothing spline object, evaluate the smoothing spline at knots, then calls `CubicInterpSplineAsPiecePoly` to construct a natural cubic interpolation spline.

## Usage

```
SmoothSplineAsPiecePoly(SmoothSpline)
```

## Arguments

| | |
|---|---|
| `SmoothSpline` | A fitted smoothing spline as returned by `stats::smooth.spline`. |

## Value

An object of "PiecePoly" and "CubicInterpSpline" class. Note that the piecewise polynomial is always parametrized in shifted form.

## Author(s)

Zheyuan Li <zheyuan.li@bath.edu>

## Examples

```
## Not run:
## a toy dataset
set.seed(0)
x <- 1:100 + runif(100, -0.1, 0.1)
y <- poly(x, 9) %*% rnorm(9)
y <- y + rnorm(length(y), 0, 0.2 * sd(y))

## fit a smoothing spline
sm <- smooth.spline(x, y)

## coerce "smooth.spline" object to "PiecePoly" object
oo <- SmoothSplineAsPiecePoly(sm)

## print the "PiecePoly"
oo

## plot the "PiecePoly"
plot(oo)

## find all the roots
xr <- solve(oo)
points(xr, rep.int(0, length(xr)))
abline(h = 0, lty = 2)

## find all stationary / saddle points
xs <- solve(oo, deriv = 1)

## predict the "PiecePoly" at stationary / saddle points
ys <- predict(oo, xs)
points(xs, ys, pch = 19)

## End(Not run)
```

---

`solve.PiecePoly` *"solve" method for "PiecePoly" class.*

---

## Description

The function solves equation $f(x) = b$ and $f'(x) = b$, where $f(x)$ is a spline represented as piecewise polynomials. The function can be used for finding roots and extrema of $f(x)$ for example.

### Usage

```
## S3 method for class 'PiecePoly'
solve(a, b = 0, deriv = 0L, ...)
```

### Arguments

| | |
|---|---|
| a | A "PiecePoly" object representing a spline $f(x)$ as piecewise polynomials. |
| b | The RHS of equations $f(x) = b$ and $f'(x) = b$. |
| deriv | Derivatives applied to $f(x)$. Only `deriv = 0` and `deriv = 1` are implemented. |
| ... | Not used by the method. |

### Value

A vector of $x$-coordinate values that solve the equation.

### Author(s)

Zheyuan Li <zheyuan.li@bath.edu>

---

## summary.PiecePoly *"summary" method for "PiecePoly" class.*

---

### Description

The function works through columns of the coefficient matrix in a PiecePoly object, exporting equation of each piecewise polynomial as a formatted string.

### Usage

```
## S3 method for class 'PiecePoly'
summary(object, no.eqn = NULL, ...)
```

### Arguments

| | |
|---|---|
| object | A "PiecePoly" object. |
| no.eqn | Number of piecewise polynomials to export. Normally not used by users (hence left as `NULL` and later set as the number of piecewise polynomials). A small value of 6 is internally used by `print.PiecePoly` for a sample display of a "PiecePoly" object. |
| ... | Not used by the method. |

### Value

The function *invisibly* returns a list of strings, where $x$ is always used as the variable name.

### Author(s)

Zheyuan Li <zheyuan.li@bath.edu>

# Index