

Package ‘pspline’

December 11, 2024

Version 1.0-21

Date 2024-12-10

Title Penalized Smoothing Splines

Author S original by Jim Ramsey <ramsay@psych.mcgill.ca>.
R port by Brian Ripley <Brian.Ripley@R-project.org>.

Maintainer Brian Ripley <Brian.Ripley@R-project.org>

Depends R (>= 2.0.0), stats, graphics

Description Smoothing splines with penalties on order m derivatives.

License Unlimited

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-12-11 10:04:16 UTC

Contents

predict.smooth.Pspline	1
smooth.Pspline	3

Index	6
--------------	----------

predict.smooth.Pspline
Smoothing Spline of Arbitrary Order at New Data

Description

Uses an object of class "smooth.Pspline" to evaluate a polynomial smoothing spline of arbitrary order or one of its derivatives at new argument values.

Usage

```
## S3 method for class 'smooth.Pspline'  
predict(object, xarg, nderiv = 0, ...)
```

Arguments

object	a fitted smooth.Pspline object.
xarg	the argument values at which the spline or its derivative is to be evaluated.
nderiv	the order of the derivative required – the default is 0, the function itself.
...	further arguments passed to or from other methods.

Details

The method produces results similar to function the predict method for [smooth.spline](#), but the smoothing function is a natural smoothing spline rather than a B-spline smooth, and the order of the spline can be chosen freely, where order in this case means the order of the derivative that is penalized. [smooth.spline](#) penalizes the second derivative, and consequently only derivatives of order 0 or 1 are useful, but because [smooth.Pspline](#) penalizes a derivative of order m, derivatives up to order m-1 are useful. The general recommendation is to penalize the derivative two beyond the highest order derivative to be evaluated.

Value

A list with components xarg and dy; the xarg component is identical to the input xarg sequence, the dy component is the evaluated derivative of order deriv.

References

Heckman, N. and Ramsay, J. O. (1996) Spline smoothing with model based penalties. McGill University, unpublished manuscript.

See Also

[smooth.Pspline](#)

Examples

```
example(smooth.Pspline)
## smoother line is given by
xx <- seq(4, 25, length=100)
lines(xx, predict(sm.spline(speed, dist, df=5), xx), col = "red")
## add plots of derivatives
lines(xx, 10*predict(sm.spline(speed, dist), xx, 1), col = "blue")
lines(xx, 100*predict(sm.spline(speed, dist), xx, 2), col = "green")
```

smooth.Pspline	<i>Fit a Polynomial Smoothing Spline of Arbitrary Order</i>
----------------	---

Description

Returns an object of class "smooth.Pspline" which is a natural polynomial smooth of the input data of order fixed by the user.

Usage

```
smooth.Pspline(x, y, w=rep(1, length(x)), norder=2, df=norder + 2,
               spar=0, method=1)
sm.spline(x, y, w, cv=FALSE, ...)
```

Arguments

x	values of the predictor variable. These must be strictly increasing, and there must be at least $2 \cdot \text{norder} + 1$ of them. sm.spline provides a simplified interface, in which the x values can be unsorted, or a list with components "x" and "y" or a two-column matrix or a complex vector.
y	one or more sets of response variable values. If there is one response variable, y is an array of the same length as x; if more than one, then y is a matrix with $\text{length}(x)$ rows and number of columns equal to the number of variables.
w	vector of positive weights for smoothing of the same length as x. If measurements at different values of x have different variances, w should be inversely proportional to the variances. The default is that all weights are one.
norder	the order of the spline. $\text{norder} = 2$ gives the cubic smoothing spline, and more generally the smoothing function is a piecewise polynomial of degree $2 \cdot \text{norder} - 1$. If derivatives are to be computed from the smoothing using <code>predict.smooth.Pspline</code> , the order should be one or two more than the highest order of derivative.
df	a number which specifies the degrees of freedom = $\text{trace}(S)$. Here S is the implicit smoothing matrix. df controls the amount of smoothing if <code>method = 2</code> .
spar	the usual smoothing parameter for smoothing splines, which is the coefficient of the integrated squared derivative of order norder. spar controls the amount of smoothing if <code>method = 1</code> .
cv	logical: should ordinary cross-validation be used (true) or generalized cross-validation.
method	the method for controlling the amount of smoothing. <code>method = 1</code> uses the value supplied for spar. <code>method = 2</code> adjusts spar so that the degrees of freedom is equal to df. <code>method = 3</code> adjusts spar so that the generalized cross-validation criterion is minimized. <code>method = 4</code> adjusts spar so that the ordinary cross-validation criterion is minimized. If <code>method = 3</code> or <code>method = 4</code> , spar defines the initial value for the minimization algorithm if positive; otherwise an internally generated value is used.

sm.spline chooses this automatically based on the supplied values and that of cv.

... additional arguments to be passed to smooth.Pspline.

Details

The method produces results similar to function `smooth.spline`, but the smoothing function is a natural smoothing spline rather than a B-spline smooth, and as a consequence will differ slightly for `norder = 2` over the initial and final intervals.

The main extension is the possibility of setting the order of derivative to be penalized, so that derivatives of any order can be computed using the companion function `predict.smooth.Pspline`. The algorithm is of order N, meaning that the number of floating point operations is proportional to the number of values being smoothed. Note that the argument values must be strictly increasing, a condition that is not required by `smooth.spline`.

Note that the appropriate or minimized value of the smoothing parameter `spar` will depend heavily on the order; the larger the order, the smaller this parameter will tend to be.

Value

an object of class "smooth.Pspline" is returned, consisting of the fitted smoothing spline evaluated at the supplied data, some fitting criteria and constants. This object contains the information necessary to evaluate the smoothing spline or one of its derivatives at arbitrary argument values using `predict.smooth.Pspline`. The components of the returned list are

<code>norder</code>	the order of the spline
<code>x</code>	values of the predictor variable
<code>ysmth</code>	a matrix with <code>length(x)</code> rows, each column of which contains the smoothed response variable values for the corresponding column of <code>y</code> .
<code>lev</code>	leverage values, which are the diagonal elements of the smoother matrix <code>S</code> .
<code>gcv</code>	generalized cross-validation criterion value
<code>cv</code>	ordinary cross-validation criterion value
<code>df</code>	a number which supplies the degrees of freedom = <code>trace(S)</code> rather than a smoothing parameter.
<code>spar</code>	the final smoothing parameter for smoothing splines. This is unchanged if <code>method = 1</code> , but adjusted otherwise.
<code>call</code>	the call that produced the fit.

References

Heckman, N. and Ramsay, J. O. (1996) Spline smoothing with model based penalties. McGill University, unpublished manuscript.

See Also

[predict.smooth.Pspline](#), [smooth.spline](#)

Examples

```
data(cars)
attach(cars)
plot(speed, dist, main = "data(cars) & smoothing splines")
cars.spl <- sm.spline(speed, dist)
cars.spl
lines(cars.spl, col = "blue")
lines(sm.spline(speed, dist, df=10), lty=2, col = "red")
```

Index

* **smooth**

- predict.smooth.Pspline, 1
- smooth.Pspline, 3

lines.smooth.Pspline (smooth.Pspline), 3

plot.smooth.Pspline (smooth.Pspline), 3

predict.smooth.Pspline, 1, 4

print.smooth.Pspline (smooth.Pspline), 3

sm.spline (smooth.Pspline), 3

smooth.Pspline, 2, 3

smooth.spline, 2, 4