# Package 'BsplineCurve'

February 21, 2025

**Type** Package

**Title** B-Spline curve fitting using ADMM

**Version** 0.1.0

**Maintainer** Jae-Hwan Jhong <0925jjh@naver.com>

**Description** This package provides functions for B-spline curve fitting using the Alternating Direction Method of Multipliers (ADMM). It supports automatic selection of regularization parameters and model selection using AIC and BIC criteria.

**License** GPL-3

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Dong-Young Lee [aut] (ChungBuk National University),
Ju-Sung Lee [aut] (ChungBuk National University),
Jae-Hwan Jhong [aut, cre] (ChungBuk National University)

## R topics documented:

bspline.curve.admm_lambdas . . . . . 1

**Index** 4

---

`bspline.curve.admm_lambdas`

*B-spline Curve Fitting using ADMM*

---

### Description

This function performs B-spline curve fitting using the Alternating Direction Method of Multipliers (ADMM). It solves for the B-spline coefficients under a regularization scheme that encourages smoothness by applying a group lasso penalty. The function returns fitted coefficients and model selection criteria across a range of regularization parameters.

### Usage

```
bspline.curve.admm_lambdas(y, D, B, lambdas = NULL, lam_max = 100,
lam_min = 1e-05, n_lambda = 100, max_iter = 500, epsilon = 1e-5, eta_c = 0.1)
```

## Arguments

| | |
|---|---|
| y | A numeric vector or matrix representing the observed data. |
| D | A difference matrix used for regularization in the ADMM algorithm. It is typically generated using bspline_jump(knots, order). |
| B | A B-spline basis matrix corresponding to the design matrix. It is typically generated using bsplines(t, knots, order). |
| lambdas | A vector of regularization parameters. If NULL, a sequence of lambda values is automatically generated. |
| lam_max | The maximum lambda value for automatic generation (only used if lambdas is NULL). |
| lam_min | The minimum lambda value for automatic generation (only used if lambdas is NULL). |
| n_lambda | The number of lambda values to generate (only used if lambdas is NULL). |
| max_iter | The maximum number of iterations for the ADMM algorithm. |
| epsilon | The convergence threshold for the iterative algorithm. The algorithm stops when the absolute difference between the penalized residual sum of squares (RSS) in consecutive iterations is less than epsilon. |
| eta_c | A constant factor to compute the ADMM parameter eta. |

## Value

A list containing results for all lambda values and model selection criteria. The list includes:

**xi** Estimated coefficient matrix for each lambda value.

**Bb** Fitted values computed as B %*% xi for each lambda.

**alpha** Auxiliary variable estimates from the ADMM algorithm.

**eta** The computed ADMM parameter eta.

**lambda** The sequence of regularization parameters used in fitting.

**u** Dual variables from the ADMM algorithm.

**iteration** Number of iterations performed for convergence.

**obj_f** Final penalized residual sum of squares (RSS) for each lambda.

**aic** A vector of Akaike Information Criterion (AIC) values for all lambda values.

**bic** A vector of Bayesian Information Criterion (BIC) values for all lambda values.

**r** The total number of lambda values considered.

**best_aic** The model results corresponding to the lambda with the lowest AIC.

**best_bic** The model results corresponding to the lambda with the lowest BIC.

## Examples

```
set.seed(123)
n <- 100
order <- 3
dimension <- 10

# Generate time points

t <- seq(0, 1, length.out = n)
```

```
# Compute B-spline basis and difference matrix

knots <- knots_quantile(t, dimension, order)
xi <- matrix(rnorm(2 * dimension), ncol = 2)
B <- bsplines(t, knots, order)
D <- bspline_jump(knots, order)

# Generate synthetic response data

f <- B
e <- matrix(rnorm(2 * n, sd = 0.1), ncol = 2)
y <- f + e

# Perform B-spline curve fitting using ADMM

fit <- bspline.curve.admm_lambdas(y, D, B)

# Plot observed data and fitted curve

best_index = which.min(fit$bic)
plot(y, col = "gray")
# true control points and curve
text(xi, col = "black", cex = 1)
lines(f, col = "black")
# fitted control points and curve
text(fit[[best_index]]$xi, col = "red", cex = 1)
lines(fit[[best_index]]$Bb, col = "red", lty = 2)
title("B-spline Curve Fitting via ADMM")
```

# Index

bspline.curve.admm_lambdas, 1