Package 'splat'

May 18, 2018

Type Package

Title Fits the SPLAT Model

Version 1.0.0

Description Fits a model with automatic variable selection where the fits of the included variables are adaptively chosen to be either linear or piecewise polynomial with adaptively-chosen knots, where the user chooses the power of the polynomial.

License GPL (>= 2)

Imports stats, graphics, grDevices, glmgen, manipulate

LazyData TRUE

RoxygenNote 6.0.1

R topics documented:

`splat-package` *splat: A package for fitting sparse partially linear additive trend filtering.*

Description

This package is called splat for "sparse partially linear additive trend filtering", which is a method for fitting a model with automatic variable selection where the fits of the included variables are adaptively chosen to be either linear or piecewise polynomial with a small number of adaptively-chosen knots, where the user chooses the power of the polynomial. The main function is `SPLAT`, which fits the model for a range of tuning parameters and provides the fits for all of these tuning parameters. Full details for the SPLAT method are provided in Petersen, A. and Witten, D. (Forthcoming). "Data-Adaptive Additive Modeling."

Examples

```
#general example illustrating all functions

#generate data
set.seed(1)
data <- sim.SPLAT.data(n = 100, nLinear = 3, nNonlinear = 1, nSparse = 5)

#fit model for a range of lambda values and fixed alpha and lambdatilde
#lambda sequence is chosen automatically if not specified
#k=2 fits piecewise quadratic (k=1 fits piecewise linear)
splat.out <- SPLAT(X = data$X, y = data$y, k=2, alpha = 0.5, lambdatilde.pct = 0.1)

#plot the estimated relationships between the predictors and outcome
#interactive plot where the user can change between lambda values
#plot(splat.out)
#or just plot a single fit for a fixed lambda
#we'll examine the fit with an index of 5. that is, lambda of
splat.out$lambda.seq[5]
plot(splat.out, lambda.index = 5)
#plot the true functions as dashed lines
plot(splat.out, lambda.index = 5, truetheta = data$theta, trueX = data$X)

#we can make predictions for a covariate matrix with new observations
#new.x with 20 observations and the same number of features as splat.out$X
new.data <- sim.SPLAT.data(n = 20, nLinear = 3, nNonlinear = 1, nSparse = 5)
new.x <- new.data$X
#get the new predictions:
predict.new <- predict(splat.out, newX = new.x, lambda.index = 5)
predict.new$yhat
predict.new$theta
```

`plot` *Plots Fit from* SPLAT.

Description

This function plots fit of the class SPLAT with a user-specified tuning parameter.

Usage

```
## S3 method for class 'SPLAT'
plot(x, lambda.index = NULL, xlabs = NULL, ylabs = NULL,
  truetheta = NULL, trueX = NULL, same.y.lim = FALSE, ...)
```

Arguments

x	An object of class SPLAT, which results from running the SPLAT function.
lambda.index	The index for the desired value of lambda, i.e., x$lambda.seq[lambda.index].
xlabs	A vector of axis labels for the covariates. By default, it is "X1", "X2", etc..
ylabs	A vector of axis labels for the predicted functions of the covariates. By default, it is "f(X1)", "f(X2)", etc..

truetheta	An optional matrix with the columns containing the theta_j's used to generate the data.
trueX	An optional matrix with the columns containing the x_j's corresponding to the theta_j's in truetheta.
same.y.lim	Should the y axis limits be chosen based on the fits from all lambda values? (Default is FALSE.)
...	Additional arguments to be passed, which are ignored in this function.

Value

None.

Examples

```
#generate data
set.seed(1)
data <- sim.SPLAT.data(n = 100, nLinear = 3, nNonlinear = 1, nSparse = 5)

#fit model for a range of lambda values and fixed alpha and lambdatilde
#lambda sequence is chosen automatically if not specified
#k=2 fits piecewise quadratic (k=1 fits piecewise linear)
splat.out <- SPLAT(X = data$X, y = data$y, k=2, alpha = 0.5, lambdatilde.pct = 0.1)

#plot the estimated relationships between the predictors and outcome
#interactive plot where the user can change between lambda values
#plot(splat.out)
#or just plot a single fit for a fixed lambda
#we'll examine the fit with an index of 5. that is, lambda of
splat.out$lambda.seq[5]
plot(splat.out, lambda.index = 5)
#plot the true functions as dashed lines
plot(splat.out, lambda.index = 5, truetheta = data$theta, trueX = data$X)
```

---

predict *Predicts Observations for a New Covariate Matrix using Fit from* SPLAT.

Description

This function makes predictions for a specified covariate matrix for a fit of the class SPLAT with a user-specified tuning parameter.

Usage

```
## S3 method for class 'SPLAT'
predict(object, newX, lambda.index, ...)
```

Arguments

object	An object of class SPLAT, which results from running the SPLAT function.
newX	The covariate matrix for which to make predictions.
lambda.index	The index for the desired value of lambda, i.e., object$lambda.seq[lambda.index].
...	Additional arguments to be passed, which are ignored in this function.

Value

A list containing the estimates for theta, gamma, and beta, as well as the fitted y values for newX.

Examples

```
#generate data
set.seed(1)
data <- sim.SPLAT.data(n = 100, nLinear = 3, nNonlinear = 1, nSparse = 5)

#fit model for a range of lambda values and fixed alpha and lambdatilde
#lambda sequence is chosen automatically if not specified
#k=2 fits piecewise quadratic (k=1 fits piecewise linear)
splat.out <- SPLAT(X = data$X, y = data$y, k=2, alpha = 0.5, lambdatilde.pct = 0.1)

#we can make predictions for a covariate matrix with new observations
#new.x with 20 observations and the same number of features as splat.out$X
new.data <- sim.SPLAT.data(n = 20, nLinear = 3, nNonlinear = 1, nSparse = 5)
new.x <- new.data$X
#get the new predictions:
predict.new <- predict(splat.out, newX = new.x, lambda.index = 5)
predict.new$yhat
predict.new$theta
```

---

sim.SPLAT.data *Simulate Data to Use with SPLAT.*

Description

This function generates data according to the simulation scenarios considered in Section 6 of the SPLAT paper (and plotted in Figure 2 of the paper).

Usage

```
sim.SPLAT.data(n, nLinear, nNonlinear, nSparse, noise = 1)
```

Arguments

n	The number of observations.
nLinear	The number of covariates with a linear association with the outcome.
nNonlinear	The number of covariates over two with a non-linear association with the outcome.
nSparse	The number of covariates not at all associated with the outcome to generate.
noise	The standard deviation of the normally-distributed noise that is added to the signal.

Value

A list containing:

- X: An n x p covariate matrix.
- y: An n-vector containing the response values.
- theta: An n x p matrix containing the true theta_j's.
- Other elements: As specified by the user.

See Also

SPLAT

Examples

```
#generate data
set.seed(1)
data <- sim.SPLAT.data(n = 100, nLinear = 3, nNonlinear = 1, nSparse = 5)
```

SPLAT *Sparse Partially Linear Additive Trend Filtering (SPLAT).*

Description

Fit a sparse additive model where each included covariate is estimated to be either linear or piecewise polynomial with a small number of adaptively-chosen knots. The functional form is adaptively chosen, and the model is fit for a sequence of tuning parameters.

Usage

```
SPLAT(y, X, k, alpha, lambdatilde.pct, lambda.min.ratio = 0.01,
  n.lambda = 20, lambda.seq = NULL, tol = 1e-04, messages = 1)
```

Arguments

y	An n-vector containing the response.
X	An n x p matrix with each column containing a covariate.
k	The power of the piecewise polynomial fit (1 is piecewise linear, 2 is piecewise quadratic, etc.); corresponds to kth order trend filtering.
alpha	Tuning parameter value between 0 and 1 that controls how strongly we encourage a linear fit versus a piecewise polynomial fit with a limited number of knots.
lambdatilde.pct	Value between 0 and 1 that represents how strongly variable selection should be encouraged with a value of 0 corresponding to no variable selection. The value of lambdatilde used will be lambdatilde.pct times the minimum value of lambdatilde known to result in a completely sparse model.

`lambda.min.ratio`
: The smallest value for `lambda.seq`, as a fraction of the maximum lambda value, which is the data-derived smallest value for which all covariates are modeled linearly (vs. with piecewise polynomial). The default is 0.01.

`n.lambda`
: The number of lambda values to consider - the default is 20.

`lambda.seq`
: A user-supplied sequence of positive lambda values to consider. The typical usage is to calculate `lambda.seq` using `lambda.min.ratio` and `n.lambda`, but providing `lambda.seq` overrides this. If provided, `lambda.seq` should be a decreasing sequence of values, since SPLAT relies on warm starts for speed. Thus fitting the model for a whole sequence of lambda values is often faster than fitting for a single lambda value.

`tol`
: Specifies the convergence criterion for the objective in block coordinate descent. The default is 10e-5.

`messages`
: Which progress messages should be printed while fitting the model? None (`messages=0`), some(`messages=1`; default), or all (`messages=2`).

Value

An object of class `SPLAT`, which can be plotted using `plot` and used to predict outcome values for new covariates using `predict`.

- `theta.list`: A list of length `n.lambda` with each element being a n x p matrix with the columns giving the estimates for the theta_j's for each value of `lambda.seq`.
- `gamma.list`: A list of length `n.lambda` with each element being a n x p matrix with the columns giving the estimates for the gamma_j's for each value of `lambda.seq`.
- `beta.mat`: A matrix of dimension p x `n.lambda` with each column giving the estimates for the beta vector for each value of `lambda.seq`.
- `knot.locations.list`: A list of length `n.lambda` with each element being a n x p matrix indicating whether or not there is a knot at the corresponding value of X for each value of `lambda.seq`.
- `linear.mat`: A matrix of dimension p x `n.lambda` with each column giving whether the variables were fit linearly for each value of `lambda.seq`.
- Other elements: As specified by the user.

See Also

`plot`, `predict`

Examples

```
#generate data
set.seed(1)
data <- sim.SPLAT.data(n = 100, nLinear = 3, nNonlinear = 1, nSparse = 5)

#fit model for a range of lambda values and fixed alpha and lambdatilde
#lambda sequence is chosen automatically if not specified
#k=2 fits piecewise quadratic (k=1 fits piecewise linear)
splat.out <- SPLAT(X = data$X, y = data$y, k=2, alpha = 0.5, lambdatilde.pct = 0.1)
```

Index