Package 'bisoreg'

March 19, 2015

Type Package

Title Bayesian Isotonic Regression with Bernstein Polynomials

Version 1.4

Date 2015-03-15

Author S. McKay Curtis <s.mckay.curtis@gmail.com>

Maintainer S. McKay Curtis <s.mckay.curtis@gmail.com>

Depends R (>= 2.7.0), bootstrap, monreg, R2WinBUGS, coda (>= 0.17.1)

Description Provides functions for fitting Bayesian monotonic regression models to data.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-03-19 06:47:09

R topics documented:

bisoreg-package *Bayesian Isotonic Regression with Bernstein Polynomials*

Description

The function `bisoreg` implements the procedure described in the paper "A Bayesian variable selection approach to monotonic regression." The package also contains additional functions to compare with the Bayesian method in the paper.

Details

Package:	bisoreg
Type:	Package
Version:	1.0
Date:	2008-10-07
License:	GPL (>=2)
LazyLoad:	yes

Author(s)

S. McKay Curtis (with helpful suggestions from Sujit K. Ghosh)

Maintainer: S. McKay Curtis

References

Curtis, S. M. and Ghosh, S. K. (2009) "A variable selection approach to monotonic regression with Bernstein polynomials." (unpublished manuscript).

Examples

```
## See examples for bisoreg ##
```

as.mcmc.biso *Coerce biso Object to mcmc Object*

Description

Coerces a `biso` object to an `mcmc` object for analysis with the `coda` package.

Usage

```
## S3 method for class 'biso'
as.mcmc(x, ...)
```

Arguments

x	a biso object.
...	unused

Value

An S3 object with class mcmc.

Author(s)

S. McKay Curtis

See Also

coda package.

---

aveslope *Average slope of a Bernstein polynomial regression function*

Description

This function returns the posterior distribution of the average slope of the Bernstein polynomial regression function over a given interval.

Usage

```
aveslope(obj, a, b)
```

Arguments

obj	an object returned by the bisoreg function
a	lower bound for the interval on which to calculate the slope
b	upper bound for the interval on which to calculate the slope

Details

See *A variable selection approach to Bayesian monotonic regression for details.*

Value

Returns a vector of average slopes for each MCMC iteration in obj

Author(s)

S. McKay Curtis

References

A variable selection approach to Bayesian monotonic regression.

See Also

bisoreg

Examples

```
## Not run:
data(childgrowth)
n.thin <- 10
n.bi <- n.thin*1000
n.iter <- n.thin*15000 + n.bi
out <- gibbsiso(childgrowth$day,childgrowth$height,m=40,nmc=n.iter,nbi=n.bi,thin=n.thin)
plot(density(aveslope(out,75,150)))

## End(Not run)
```

---

bisoreg *Bayesian Isotonic Regression*

Description

Computes MCMC simulations from the poseterior distribution of the monotonic Bernstein polynomial regression function

Usage

```
bisoreg(x, y, m = floor(n/2), priors = list(), inits = list(),
        n.sim = 5000, n.burn = 1000, n.thin = 10, seed)
```

Arguments

x	vector of covariate values
y	vector of response values
m	positive integer specifying the order of the Bernstein polynomial basis expansion. Default is half the number of observations in the data set. This value is reset with `m <- floor(m)` to prevent noninteger values from being used.
priors	list of values of prior parameters. Values in the list must be named one of (with default values in parentheses) `a.sig` (0.1), `b.sig` (0.1), `a.tau` (1), `b.tau` (1), `a.p` (1), `b.p` (1), `m0` (0), `ssq0` (1000).

inits	list of initial values for model parameters. Values in the list must be named one of u0, u (must be a vector of length equal to m), sigsq, tausq, p. If not specified, initial values are randomly generated.
n.sim	number of MCMC simulations after burn-in and thinning. The total number of MCMC iterations run is n.sim*n.thin + n.burn. Thus, the final number MCMC simulations in the output of the function will always be n.sim.
n.burn	number of simulations to burn
n.thin	keep every thin draw from the MCMC simulations
seed	random number seed for the MCMC simulations

Details

Generates MCMC draws from the posterior distribution of the nonparametric Bernstein regression function as described in Curtis and Ghosh (2009).

Value

An S3 object with class "biso" and the following components:

mcmctime	the time it took to run the sampling algorithm. mcmctime is the output from a call to function system.time.
postdraws	a data frame with posterior draws of all parameters
W	the "W" matrix
x	original x values
y	original y values
m	order of the Bernstein polynomial basis expansion
DIC	"original" DIC value as in Spiegelhalter et. al. (2002)
DIC2	"alternative" DIC value as in Gelman et. al. (2004)
pD	effective number of parameters as in Spiegelhalter et. al. (2002)
pD2	alternative effective number of parameters as in Gelman et. al. (2004)

Note

No further notes.

Author(s)

S. McKay Curtis with many helpful suggestions from Sujit K. Ghosh.

References

Curtis, S. M. and Ghosh, S. K. (2009) "A variable selection approach to monotonic regression with Bernstein polynomials."

Gelman, A., et. al. (2003). *Bayesian Data Analysis*. Chapman and Hall / CRC.

Speigelhalter, D. J., et. al. (2002). "Bayesian measures of model complexity and fit." JRSSB 64(4): 583–616.

See Also

pflat.biso,fitted.biso,plot.biso

Examples

```
## Not run:
data(childgrowth)

## Run three different chains ##
out1 <- bisoreg(childgrowth$day, childgrowth$height,
                m=80, n.sim=15000, n.thin=10)
out2 <- bisoreg(childgrowth$day, childgrowth$height,
                m=80, n.sim=15000, n.thin=10)
out3 <- bisoreg(childgrowth$day, childgrowth$height,
                m=80, n.sim=15000, n.thin=10)

## Convert to MCMC object ##
out <- list()
out[[1]] <- mcmc(out1$postdraws)
out[[2]] <- mcmc(out2$postdraws)
out[[3]] <- mcmc(out3$postdraws)
mcmcout <- mcmc.list(out)

## Gelman Rubin diagnostics ##
gelman.diag(mcmcout)

## Posterior probability of a flat function ##
pflat(out1)

## Create scatterplot and plot Bayesian regression fit ##
plot(out1, color="black", cl=F, lwd=2, xlab="day", ylab="height")
title("Child Growth Data")

## Compute isoreg fit ##
iout = isoreg(childgrowth$day, childgrowth$height)
lines(as.stepfun(iout), col.h="red", col.v="red", lwd=2)

## Compute monreg estimate of Dette et. al. ##
mout = monreg.wrapper(childgrowth$day, childgrowth$height)
lines(childgrowth$day, mout$est, lwd=2, col="blue")

## Compute local polynomial regression fit ##
lines(childgrowth$day,
      predict(loess.wrapper(childgrowth$day, childgrowth$height)),
      lwd=2, col="green")

## Add a legend ##
legend("topleft",
       legend=c("bayes", "isoreg", "monreg", "loess"),
       col=c("black", "red", "blue", "green"), lwd=2)

## Check sensitivity to the prior on p ##
```

```
## p ~ Beta(a=1,b=1) ##
outa1b1 <- bisoreg(childgrowth$day, childgrowth$height,
                   40, priors=list(a.p=1,b.p=1), n.sim=15000, n.thin=10)
## p ~ Beta(a=1,b=3) ##
outa1b3 <- bisoreg(childgrowth$day, childgrowth$height,
                   40, priors=list(a.p=1,b.p=3), n.sim=15000, n.thin=10)
## p ~ Beta(a=3,b=1) ##
outa3b1 <- bisoreg(childgrowth$day, childgrowth$height,
                   40, priors=list(a.p=3, b.p=1), n.sim=15000, n.thin=10)

## Create plot to compare model fits ##
plot(outa1b1,cl=F,color="black")
plot(outa1b3,cl=F,add=T,col="red")
plot(outa3b1,cl=F,add=T,col="green")
gplots:::smartlegend("left","top",
                     legend=c("a=1 b=1","a=1 b=3","a=3 b=1"),
                     col=c("black","red","green"),
                     lwd=1)

## Check sensitivity to the choice of M ##
outm20 <- bisoreg(childgrowth$day, childgrowth$height, m=20, n.sim=15000, n.thin=10)
outm30 <- bisoreg(childgrowth$day, childgrowth$height, m=30, n.sim=15000, n.thin=10)
outm40 <- bisoreg(childgrowth$day, childgrowth$height, m=40, n.sim=15000, n.thin=10)

## Create plot to compare model fits ##
plot(outm20, cl=F, col="blue")
plot(outm30, cl=F, add=T, col="red")
plot(outm40, cl=F, add=T, col="green")
plot(out1, cl=F, add=T, col="black")
gplots:::smartlegend("left", "top",
                     legend=c(20,30,40,80),
                     col=c("blue", "red", "green", "black"),
                     lwd=1)


## A method for choosing 'm' ##
## (from Sujit K. Ghosh) ##

## Piece-wise linear 'true' regression function ##
mu <- function(x){
  ifelse(-3<=x & x<=-1, x, 0) +
      ifelse(-1<x & x<1, -1, 0) +
          ifelse(1<=x & x<3, x - 2, 0)
}
par(mfrow=c(2,2))
curve(mu, -2.9, 2.9)

## Simulate the data ##
n <- 100
x <- runif(n, -2.9, 2.9)
y <- mu(x) + rnorm(n, 0, 0.1)

## Fit initially with default 'm=n/2' ##
```

```
fit0 <- bisoreg(x, y, m=floor(n/2))
plot(fit0)
lines(sort(x), mu(sort(x)), col="blue")
fit0$DIC2
fit0$pD2

## Use effective number of parameters ##
## as new value of 'm' ##
fit <- bisoreg(x, y, m=ceiling(fit0$pD2))
plot(fit)
lines(sort(x), mu(sort(x)), col="blue")
fit$DIC2
fit$pD2

## Compare fits ##
plot(fitted(fit0), fitted(fit))
abline(0, 1)
mean(abs(y - fitted(fit0)))
mean(abs(y - fitted(fit)))

## End(Not run)
```

`childgrowth` *Heights of a ten-year-old boy over the course of a year*

Description

This data set is the `onechild` dataset from the `fda` package.

Usage

```
childgrowth
```

Format

A `data.frame` containing 83 observations on two variables.

`day` is day of the year on which the height measurement was taken.

`height` is the height in inches of the child.

Source

See the `fda` package.

References

Ramsay, James O., and Silverman, Bernard W. (2002). *Applied Functional Data Analysis 2nd ed.* Springer, New York.

Tuddenham, R. D., and Snyder, M. M. (1954). "Physical growth of California boys and girls from birth to age 18", *University of California Publications in Child Development*, 1, 183-364.

`fits.isoreg` *Computes fitted values for isoreg*

Description

Computes the fitted values for an object that has been creatd using the isoreg function

Usage

```
fits.isoreg(iso, x0)
```

Arguments

`iso`	on object created with the function `isoreg`
`x0`	x-values at which to compute the fitted values

Details

None.

Value

A vector of fitted values.

Note

None.

Author(s)

S. McKay Curtis

References

None.

See Also

`isoreg`

Examples

```
## None ##
```

`fitted.biso` *Compute fitted values for a biso object*

Description

Computes pointwise posterior medians of the fitted Bernstein regression function

Usage

```
## S3 method for class 'biso'
fitted(object, xnew, ...)
```

Arguments

`object`	a `biso` object
`xnew`	values of `x` at which to compute posterior median of the fuction values. If missing, the orginal `x` vector is used.
`...`	not used

Details

None.

Value

A vector of posterior medians.

Note

None.

Author(s)

S. McKay Curtis

References

None.

See Also

`bisoreg`

Examples

```
## See examples for bisoreg
```

get.W *Monotonic Bernstein Polynomial Basis Expansion*

Description

Computes the monotonic, Bernstein polynomial basis expansion matrix "W" for a vector x.

Usage

```
get.W(x, m)
```

Arguments

x	vector
m	order of the Bernstein polynomial expansion

Details

None.

Value

A matrix with dimensions `length(x)` by `m`.

Note

None.

Author(s)

S. McKay Curtis

References

None.

See Also

bisoreg

Examples

```
## See the code to the bisoreg function
```

`loess.wrapper` *Compute a loess fit using cross validation*

Description

Uses the `crossval` function from the `boostrap` package to compute the optimal value of the tuning parameter in the `loess` function and returns the corresponding `loess` fit.

Usage

```
loess.wrapper(x, y, span.vals = seq(0.25, 1, by = 0.05), folds = 5)
```

Arguments

`x`	predictor values
`y`	response values
`span.vals`	values of the tuning parameter to evaluate using cross validation
`folds`	number of "folds" for the cross-validation procedure

Details

None.

Value

Returns a `loess` object.

Note

None.

Author(s)

S. McKay Curtis

References

Chambers, J. M. (1991) *Statistical Models in S*.

See Also

`loess`

Examples

```
## See examples for bisoreg ##
```

`monreg.wrapper` *Fit a monotonic regression function using monreg and cross validation*

Description

Uses cross validation to select the optimal value of a tuning parameter in the monreg function.

Usage

```
monreg.wrapper(x, y, hr.vals = seq(0.01, 0.4, l = 40), x0, folds = 5)
```

Arguments

`x`	predictor values
`y`	response values
`hr.vals`	values of the tuning parameter to evaluate using cross validation
`x0`	`x` values at which to compute fits. If missing, `x0` is set to `x`
`folds`	number of folds to use in the cross-validation procedure

Details

None.

Value

Returns a `monreg`.

Warning

As of the current version of the `srbern` package, the `monreg` function in the `monreg` package has a memory leak in its `C` code. Calling the `monreg` function multiple times (e.g. in a simulation) will eventually consume all of the RAM on your computer and crash `R`.

Note

None.

Author(s)

S. McKay Curtis

References

Dette, H., Neumeyer, N., Pilz, K. F. (2006). "A simple nonparametric estimator of a strictly monotone regression function." *Bernoulli* 12(3), 469-490.

See Also

monreg

Examples

```
## See examples for bisoreg ##
```

---

nd2004 *Compute Bayesian monotone regression function of Neelon and Dunson (2004)*

Description

Returns posterior draws from the monotone regression procedure of Neelon and Dunson (2004) using OpenBUGS

Usage

```
nd2004(x, y, M, n.sim = 5000, n.thin = 1, n.burn = 1000)
```

Arguments

x	predictor values
y	response values
M	number of knots
n.sim	number of MCMC iterations in the final MCMC sample (i.e., iterations kept after thinning and burn-in period)
n.thin	number of iterations to "thin" out of the MCMC chain after burn in
n.burn	number of iterations to burn

Details

Uses the openbugs function in the R2WinBUGS package to compute the posterior draws. The chains do not converge, however, as indicated by the poor Gelman-Rubin diagnostic.

Value

A bugs object.

Author(s)

S. McKay Curtis

References

Neelon, B. and Dunson, D.B. (2004). "Bayesian isotonic regression and trend analysis." *Biometrics* 60, 398–406.

See Also

openbugs

Examples

```
## Not run:
data(childgrowth)
out <- nd2004(childgrowth$day, childgrowth$height, 10)
plot(out)

## End(Not run)
```

`pflat` *Posterior probability of a flat regression curve*

Description

Compute the posterior probability of a flat regression function.

Usage

```
pflat(obj, ...)
## Default S3 method:
pflat(obj,...)
## S3 method for class 'biso'
pflat(obj,...)
```

Arguments

`obj`	a `biso` object
`...`	not used

Details

No details.

Value

Returns the average of the postrior draws where $u_1 = \cdots = u_M$

Note

No further notes.

Author(s)

S. McKay Curtis

References

No references.

See Also

bisoreg

Examples

```
## See examples for bisoreg
```

---

plot.biso *Plot a biso object*

Description

Plots the data and the fit a Bayesian monontoic regression function (with optional confidence bands)

Usage

```
## S3 method for class 'biso'
plot(x, xnew, cl = TRUE, add = FALSE, color ="red", ...)
```

Arguments

x	a biso object
xnew	x values at which to plot the regression fit. If missing, the function uses the orginal x values
cl	plot confidence bands (default is TRUE)
add	add a line to the existing plot rather than create a new plot (default is FALSE)
color	color of the fitted line (default is red)
...	other options to pass to the plotting functions

Details

None.

Value

Returns a plot.

Note

None.

Author(s)

S. McKay Curtis

References

None.

See Also

bisoreg

Examples

```
## See examples for bisoreg
```

---

print.biso *Print a biso Object*

Description

Prints a summary of a biso object.

Usage

```
## S3 method for class 'biso'
print(x, digits = 3, ...)
```

Arguments

x	a biso object.
digits	nonnegative integer specifying the number of digits to round the output.
...	further arguments to pass to print.

Details

None.

Value

None.

Note

None.

Author(s)

S. McKay Curtis

References

None.

See Also

summary.biso

Examples

```
## See examples for bisoreg ##
```

---

summary.biso *Summary of a biso Object*

Description

Returns a matrix of summary statistics for the MCMC draws in a biso object.

Usage

```
## S3 method for class 'biso'
summary(object, ...)
```

Arguments

object	a biso object.
...	not implemented.

Details

None.

Value

Returns a matrix with columns for the mean and five number summary of all the parameters in postdraws element of the biso object x. There is also an additional column for the probability that each u parameter is equal to zero.

Note

None.

Author(s)

S. McKay Curtis

References

None.

See Also

bisoreg

Examples

```
## See examples for bisoreg ##
```

Index