Package 'btf'

February 19, 2015

Type Package

Title Estimates univariate function via Bayesian trend filtering

Version 1.1

Date 2014-07-29

Author Edward A. Roualdes

Maintainer Edward A. Roualdes <edward.roualdes@uky.edu>

Description Trend filtering uses the generalized
lasso framework to fit an adaptive polynomial of degree k to
estimate the function f_0 at each input x_i in the model: y_i =
f_0(x_i) + epsilon_i, for i = 1, ..., n, and epsilon_i
is sub-Gaussian with E(epsilon_i) = 0. Bayesian trend filtering adapts
the genlasso framework to a fully Bayesian hierarchical model, estimating
the penalty parameter lambda within a tractable Gibbs sampler.

License GPL (>= 2.0)

Depends R (>= 3.1.0)

Imports Matrix, coda,

LinkingTo Rcpp (>= 0.11.0), RcppEigen (>= 0.3.2.1.1)

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-07-30 16:15:16

R topics documented:

btf *Bayesian trend filtering via Eigen*

Description

Fits Bayesian trend filtering hierarchical model to univariate function. Two conditional priors are available: double exponential or generalized double Pareto.

Usage

```
btf(y = "vector", x = NULL, k = "int", iter = 10000,
  cond.prior = c("gdp", "dexp"), alpha = NULL, rho = NULL,
  debug = FALSE)
```

Arguments

y	response vector
x	inputs corresponding to y observations
k	degree of polynomial fit
iter	number of samples to draw from posterior
cond.prior	choose the conditional prior on flsigma
alpha	shape parameter for prior on lambda
rho	rate parameter for prior on lambda
debug	boolean telling btf to check for NaNs or not

Author(s)

Edward A. Roualdes

References

R. J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. The Annals of Statistics, 42(1):285-323, 2014.

See Also

trendfilter

Examples

```
# Cubic trend filtering
# from genlasso::trendfilter
## Not run: n <- 100
beta0 = numeric(100)
beta0[1:40] <- (1:40-20)^3
beta0[40:50] <- -60*(40:50-50)^2 + 60*100+20^3
```

```
beta0[50:70] <- -20*(50:70-50)^2 + 60*100+20^3
beta0[70:100] <- -1/6*(70:100-110)^3 + -1/6*40^3 + 6000
beta0 <- -beta0
beta0 <- (beta0-min(beta0))*10/diff(range(beta0))
y <- beta0 + rnorm(n)
bfit <- btf(y=y, k=3)
plot(bfit, col='grey70')
## End(Not run)
```

`genDelta` *generate Matrix D^k+1*

Description

generate Matrix D^k+1

Usage

```
genDelta(n, k, x)
```

Arguments

`n`	sample size
`k`	order of fit
`x`	vector of inputs on domain

Author(s)

Edward A. Roualdes

`getPost` *get posterior draws from btf object*

Description

Return posterior draws from a btf object into a coda object

Usage

```
getPost(btf, parameter = c("beta", "s2", "lambda", "omega", "alpha"),
  burn = 1000)
```

Arguments

`btf`	btf object
`parameter`	name of the parameter of interest
`burn`	number of samples to discard

See Also

getPostEst

`getPostEst` *get posterior estimates from btf object*

Description

Return posterior estimates, computed from btf posterior draws. User can supply their own estimating function, `est`, applied to the columns of `getPost`.

Usage

```
getPostEst(btf, parameter = c("beta", "s2", "lambda", "omega", "alpha"),
  burn = 1000, est = mean)
```

Arguments

`btf`	btf object
`parameter`	name of the paramater of interest
`burn`	number of samples to discard
`est`	estimate of the object

See Also

getPost

`plot.btf` *plot btf object*

Description

Plots a btf object with optional credible intervals.

Usage

```
## S3 method for class 'btf'
plot(x, t = NULL, burn = 1000, est = median,
  probs = c(0.025, 0.975), ...)
```

Arguments

x	btf object
t	domain of function
burn	size of burn-in,
est	function specifying how draws from the posterior are summarized
probs	numeric 2-vector of credible interval probabilities; if FALSE, no credible intervals are plot
...	extra arguments

Author(s)

Edward A. Roualdes

---

tf *approximate trend filtering via MM algorithm*

Description

Uses majorization-minimization technique to approximate trend filtering fit.

Usage

```
tf(y, x = NULL, k = 2, l, D = NULL, eps = 1e-08, tau = 1e-05,
  max_iter = 500)
```

Arguments

y	observed data
x	inputs corresponding to observations
k	order of fit
l	vector of penalty parameters lambda
D	matrix Delta of order k+1
eps	error adjustment to majorization function
tau	convergence threshold
max_iter	maximum number of iterations allowed

Author(s)

Edward A. Roualdes

Index