

Example on Bayesian Spectral Analysis Regression

Statlab2

2017-11-16

Preliminaries

To use `bsamGP`, please install the package with R command `install.packages("bsamGP")`. You then load the `bsamGP` package using the `library` or `require` function:

```
library(bsamGP)
```

This needs to be done every time you start R.

To get help on the functions in R (and in `bsamGP`), use `help()` or `?`. For example, to view the help file for the `bsar` function, type one of the following:

```
help(bsar) # ?bsar
```

Bayesian Spectral Analysis Regression (BSAR)

Let's now proceed to the monotone convex function estimating via the following BSAR model.

$$Y_i = \mathbf{w}_i^\top \boldsymbol{\beta} + f(x_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2),$$

where \mathbf{w}_i and $\boldsymbol{\beta}$ are $p + 1$ -dimensional vectors of covariates and coefficients; f is an unknown function of the scalar x_i estimated by

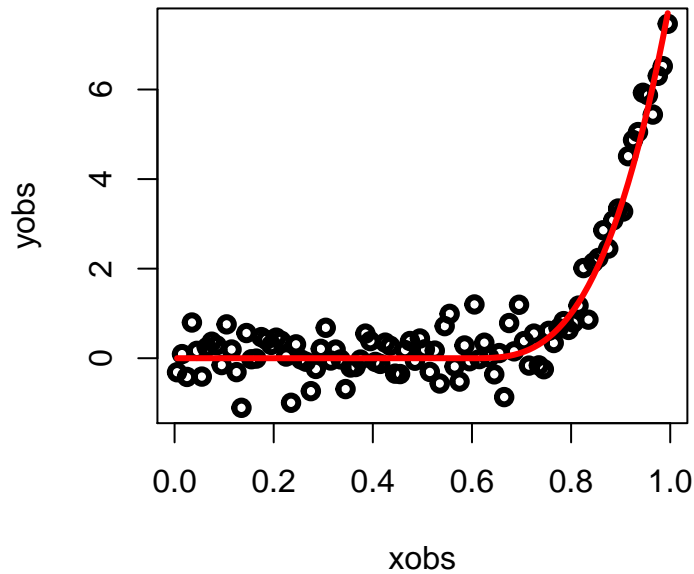
$$f(x) = \delta \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} \theta_j \theta_k \varphi_{j,k}^b(x) + \alpha(x - 0.5)$$
$$\varphi_{j,k}^b(x) = \int_0^x \int_0^s \varphi_j(t) \varphi_k(t) dt ds - \int_0^1 \int_0^x \int_0^s \varphi_j(t) \varphi_k(t) dt ds dx.$$

Data generation

We will consider estimating the following monotone convex function where $x \in [0, 1]$.

$$f(x) = (5x - 3)^3 I(5x - 3 > 0)$$

```
set.seed(1)
ftn = function(x) (5 * x - 3)^3 * (5 * x - 3 > 0)
n = 100
xobs = (2 * (1:n) - 1) / (2 * n)
yobs = ftn(xobs) + rnorm(n, sd=0.5)
plot(xobs, yobs, lwd=3)
lines(xobs, ftn(xobs), lwd=3, col=2)
```



Model Fitting

To fit BSAR model, we first set up the MCMC parameters, the number of basis and prior information for spectral coefficients.

```
# MCMC parameters
nblow0 = 1000; # Initialization period for adaptive metropolis
nblow = 20000; # Number of MCMC in transition period
smcmc = 2000; # Number of MCMC for analysis
nskip = 10; # Number of MCMC to skip after nblow
ndisp = 1000; # number of saved draws to be displayed on screen
maxmodmet = 5; # Maximum number of times to modify metropolis
# Prior information
iflagprior = 0; # 1 = Lasso Smoother, 0 = T Smoother
nbasis = 30; # number of cosine basis functions @
```

To generate a posterior sample for the Bayesian semiparametric regression model, use the function bsar.

```
# Fit the model
fout = bsar(yobs ~ fs(xobs), shape='IncreasingConvex',
            xmin=0, xmax=1, nbasis=nbasis,
            mcmc=list(maxmodmet=maxmodmet, nblow0=nblow0, nblow=nblow, smcmc=smcmc, nskip=nskip, ndisp=ndisp),
            prior=list(iflagprior=iflagprior))
```

```
## Initializing MCMC parameters ...
## Burnin ...
## function[1]: pmet = 0.4466
## Main iterations ...
## MCMC draws 1000 of 2000 (CPU time: 22.578 s)
```

```
## MCMC draws 2000 of 2000 (CPU time: 29.547 s)
## function[1]: pmet = 0.4466
```

The output returns bsam class object. The summary function on bsam object summarizes the fit.

```
summary(fout)
```

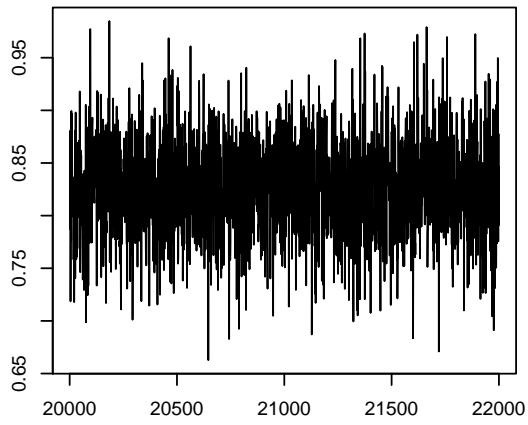
```
##
## Number of Cosine basis functions      = 30
## Number of observations                 = 100
## Number of covariates (no intercept)   = 0
##
## MCMC transition draws                  = 20000
## MCMC draws saved for estimation        = 2000
## Save every nskip draws                 = 10
## MCMC draws total                       = 40000
##
## Function = 1
## Proportion of Theta Accepted after burn-in = 0.44655
##
## R-Square                               = 0.9378
##
## Log Integrated Likelihood
## LIL Gelfand & Dey                      = -102.6756
## LIL Newton & Raftery (biased)         = -67.3496
##
## beta
##           PostM      PostStd PostM/STD
## (Intercept) 0.8268324 0.04701043 17.58828
##
## sigma
## PostM sigma = 0.4650313
## PostS sigma = 0.03396708
##
## -----
##
## Function = 1
## Increasing convex function
##
## Linear term alpha in x for constrained f
## Posterior mean   of alpha = 0.1208287
## Posterior stddev of alpha = 0.1067718
##
## theta_k ~ N(0, sigma*tau2*exp(-gamma*k))
##
## Tau
## PostM      PostS
## 3.577263 0.938478
##
## Gamma
##           PostM      PostS
## 0.38849995 0.04447833
##
## Zeta = ln(tau2) - mean(k)*gamma
##           PostM      PostS
```

```
## -3.3407003  0.4893772
##
## Cosine Basis weights theta
##      PostMean PostSTD   Ratio
## T0   3.82455 0.47397  8.06921
## T1  -4.39168 0.47680 -9.21071
## T2   2.25348 0.57582  3.91355
## T3  -0.12302 0.59218 -0.20775
## T4  -1.14814 0.48609 -2.36198
## T5   1.23613 0.43352  2.85140
## T6  -0.66332 0.45148 -1.46921
## T7   0.05528 0.35379  0.15626
## T8   0.00000 0.37483  0.00000
## T9  -0.01742 0.30766 -0.05661
## T10  0.11611 0.24293  0.47795
## T11  0.00205 0.31170  0.00659
## T12  0.01143 0.15192  0.07525
## T13  0.03309 0.13812  0.23959
## T14  0.01322 0.13788  0.09585
## T15 -0.02363 0.08463 -0.27921
## T16 -0.01188 0.12984 -0.09151
## T17  0.01051 0.08044  0.13069
## T18  0.02333 0.06563  0.35555
## T19 -0.01476 0.05437 -0.27144
## T20 -0.00503 0.04944 -0.10176
## T21 -0.01513 0.03510 -0.43098
## T22 -0.00215 0.02419 -0.08867
## T23  0.00520 0.02629  0.19773
## T24 -0.01641 0.02699 -0.60823
## T25 -0.00503 0.01321 -0.38052
## T26  0.02211 0.01738  1.27239
## T27  0.00411 0.01190  0.34553
## T28  0.00790 0.00953  0.82925
## T29  0.00553 0.00968  0.57154
## T30  0.00242 0.00646  0.37408
```

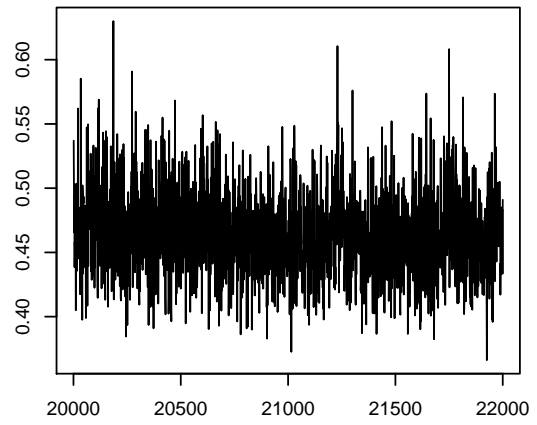
The plot function on bsam object returns traceplots of each parameters for diagnostics.

```
plot(fout)
```

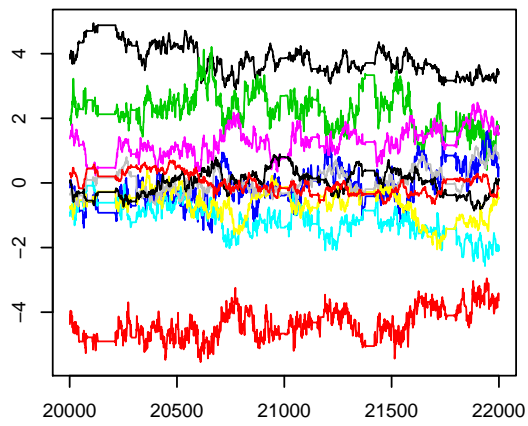
Beta



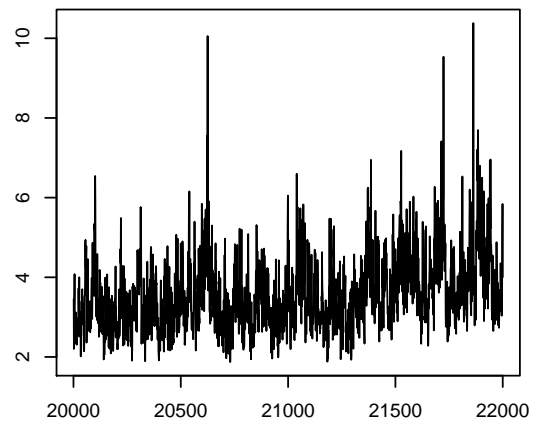
Sigma



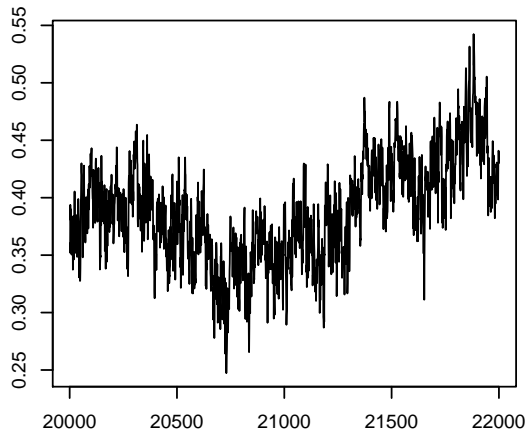
Function 1: Theta



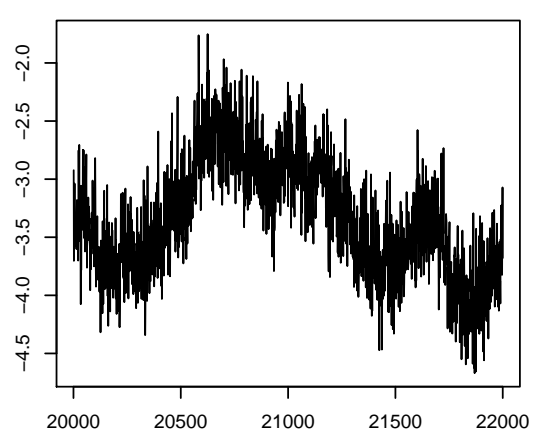
Function 1: Tau



Function 1: Gamma

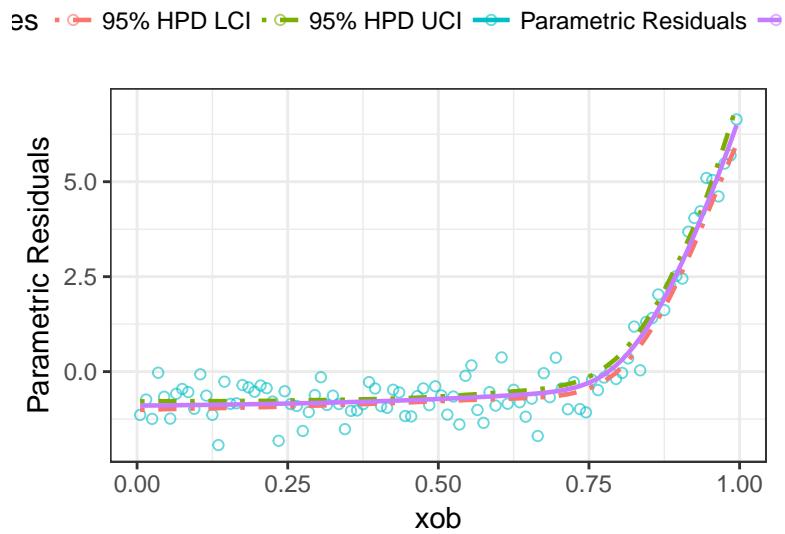
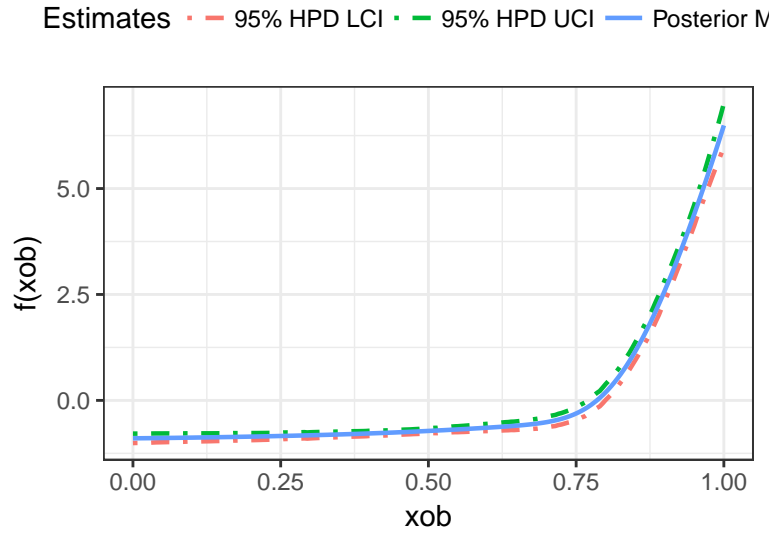


Function 1: Zeta



We may visualize fitted mean curve and 95% highest posterior density (HPD) interval with `plot` function on fitted object from `fitted` method.

```
fit = fitted(fout)
plot(fit, ggplot2=TRUE)
```



For more detailed examples and real data applications, see Jo, S., Choi, T., Park, B., & Lenk, P. (2017) "bsamGP: An R package for Bayesian Spectral Analysis Models using Gaussian Process Priors", *Preprint* .