Example on Bayesian Spectral Analysis Quantile 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 `bsaq` function, type one of the following:

```
help(bsaq) # ?bsaq
```

Bayesian Spectral Analysis Quantile Regression (BSAQ)

Let's now proceed to the 'S' shape-restricted function estimating via the following BSAQ model.

$$Y_i = \boldsymbol{w}_i^\top \boldsymbol{\beta} + f(x_i) + \epsilon_i, \;\; \epsilon_i \sim ALD(p;0,\sigma), \tag{1}$$

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

$$\begin{aligned}
f(x) &= \delta\zeta \int_0^x \int_0^s Z^2(t)h(t)dt\,ds - \delta\xi(x-0.5) + \alpha(x-0.5) \\
\xi &= \min\left[0, \min_{x\in[0,1]} \zeta \int_0^x Z^2(s)h(s)ds\right] \\
\varphi^b_{j,k}(x) &= \int_0^x \int_0^s \varphi_j(t)\varphi_k(t)dt\,ds - \int_0^1 \int_0^x \int_0^s \varphi_j(t)\varphi_k(t)dt\,ds\,dx.
\end{aligned}$$

The model (1) can be represented by a location-scale mixture of exponential and normal distributions as follows.

$$\begin{aligned}
&Y_i = \boldsymbol{w}_i^T\boldsymbol{\beta} + f(x_i) + \eta_1\nu_i + \epsilon_i, \;\; i=1,\ldots,n, \\
&\epsilon_i \sim N(0,\sigma^2\eta_2^2 z_i), \;\; \nu_i = \sigma z_i, \;\; z_i \sim Exp(1), \;\; i=1,\ldots,n,
\end{aligned}$$

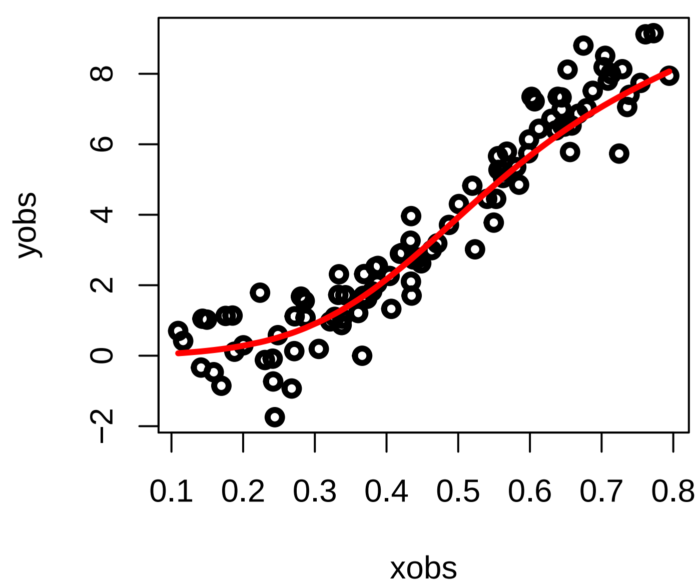
where $\boldsymbol{w}_i$ and $\boldsymbol{\beta}$ are $(p+1)$-dimensional vectors of covariates and coefficients, $\eta_1 = (1-2p)/\{p(1-p)\}$ and $\eta_2^2 = 2/\{p(1-p)\}$.

Data generation

We will consider estimating the following 'S' shape-restricted function where $x \in [0,1]$.

$$f(x) = 5\exp(-10(x-1)^4) + 5x^2$$

```
set.seed(1)
ftn = function(x) 5 * exp(-10 * (x-1)^4)+5 * x^2
n = 100
xmin = 0.1; xmax = 0.8;
xobs = xmin+(xmax-xmin) * runif(n)
xobs = sort(xobs)
yobs = ftn(xobs) + rald(n,p=0.5)
plot(xobs,yobs,lwd=3)
lines(xobs,ftn(xobs),lwd=3,col=2)
```



Model Fitting

To fit BSAQ 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 adpative metropolis
nblow = 10000; # 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 @

# quantile of interest
p=0.5 # median regression
```

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

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

```
## Initializing MCMC parameters ...
## function[1]: pmet = 0.2760 < 0.3. Reduce metm and redo MCMC loop
## function[1]: pmet = 0.6620 > 0.6. Increase metm and redo MCMC loop
## Burnin ...
## function[1]: pmet = 0.3081
## Main iterations ...
## MCMC draws 1000 of 2000 (CPU time: 21.797 s)
## MCMC draws 2000 of 2000 (CPU time: 30.562 s)
## function[1]: pmet = 0.2601
```

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

```
summary(fout)
```

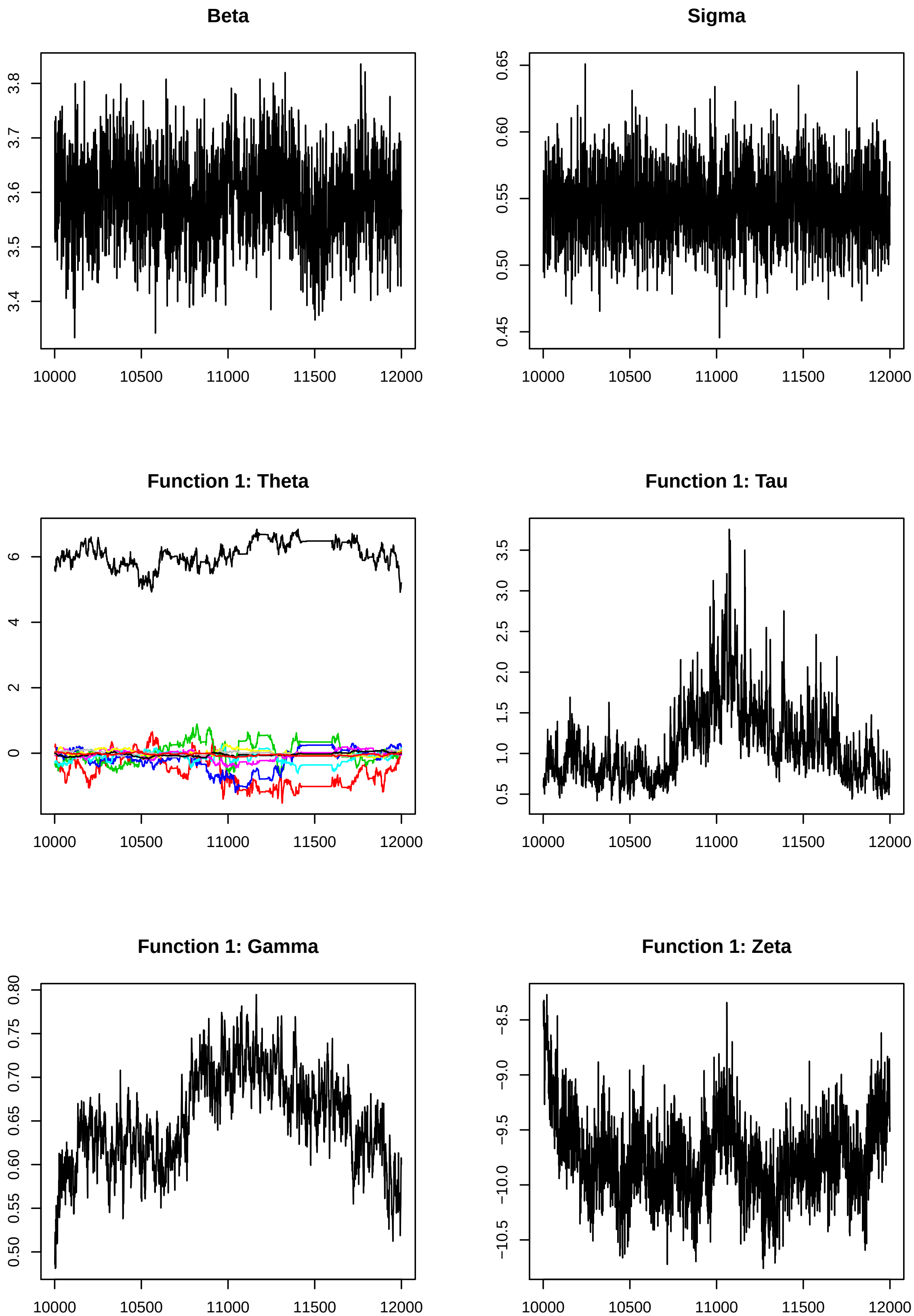
```
##
## Quantile of interest                    =  0.5
## Number of Cosine basis functions        =  30
## Number of observations                  =  100
## Number of covariates (no intercept)     =  0
##
## MCMC transition draws                   =  10000
## MCMC draws saved for estimation         =  2000
## Save every nskip draws                  =  10
## MCMC draws total                        =  30000
##
## Function =  1
## Proportion of Theta Accepted after burn-in    =  0.26015
##
## R-Square                         =  0.9357
##
## Log Integrated Likelihood
## LIL Gelfand & Dey                =  -149.2621
## LIL Newton & Raftery (biased)    =  -118.5313
##
## beta
##                PostM    PostStd PostM/STD
## (Intercept) 3.590208 0.07820325  45.90868
##
## sigma
```

```
## PostM sigma = 0.5445777
## PostS sigma = 0.02774913
##
## -----
##
## Function = 1
## Increasing S (convex to concave) function
##
## Linear term alpha in x for constrainted f
## Posterior mean   of alpha = 1.268502
## Posterior stddev of alpha = 0.899959
##
## psi is slople of squish function
## Posterior mean   of psi = 99.99063
## Posterior stddev of psi = 0.2547264
##
## omega is inflection point of squish function
## Posterior mean  omega =  0.5518184
## Posterior stdev omega =  0.02345653
##
## theta_k ~ N(0,sigma*tau2*exp(-gamma*k))
##
## Tau
## PostM   PostS
## 1.094411 0.4684179
##
## Gamma
##      PostM      PostS
## 0.65154636 0.05370321
##
## Zeta = ln(tau2) - mean(k)*gamma
##      PostM      PostS
## -9.7525902  0.3811029
##
## Cosine Basis weights theta
##     PostMean PostSTD    Ratio
## T0   6.09456 0.38012 16.03306
## T1  -0.57424 0.45535 -1.26109
## T2   0.04237 0.32366  0.13093
## T3  -0.20269 0.34984 -0.57939
## T4  -0.13233 0.16296 -0.81199
## T5  -0.04989 0.14613 -0.34141
## T6   0.00268 0.08663  0.03092
## T7  -0.01732 0.06186 -0.28004
## T8  -0.03979 0.05442 -0.73119
## T9  -0.04268 0.03767 -1.13286
## T10 -0.01749 0.02610 -0.66991
## T11  0.00524 0.02247  0.23331
## T12  0.00493 0.01341  0.36731
## T13  0.00007 0.00962  0.00730
## T14  0.00526 0.00529  0.99423
## T15 -0.00376 0.00691 -0.54441
## T16  0.00062 0.00391  0.15765
## T17 -0.00097 0.00268 -0.36325
```

```
## T18  0.00011 0.00242  0.04580
## T19  0.00053 0.00121  0.43431
## T20  0.00002 0.00082  0.02153
## T21 -0.00008 0.00079 -0.10612
## T22  0.00003 0.00056  0.06048
## T23  0.00010 0.00038  0.26192
## T24 -0.00015 0.00027 -0.56616
## T25 -0.00006 0.00026 -0.24446
## T26  0.00003 0.00024  0.14149
## T27 -0.00006 0.00009 -0.69569
## T28 -0.00002 0.00009 -0.23941
## T29  0.00001 0.00006  0.15423
## T30  0.00002 0.00004  0.44088
```

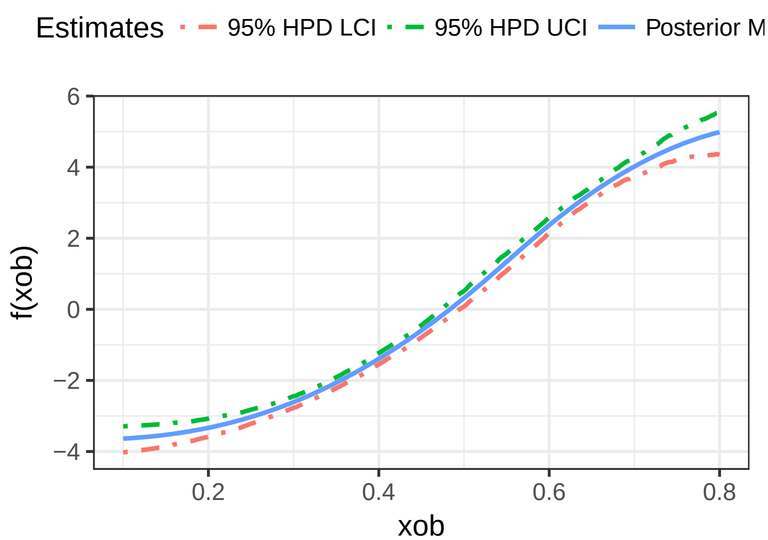
The `plot` function on `bsam` object returns traceplots of each parameters for diagnostics.

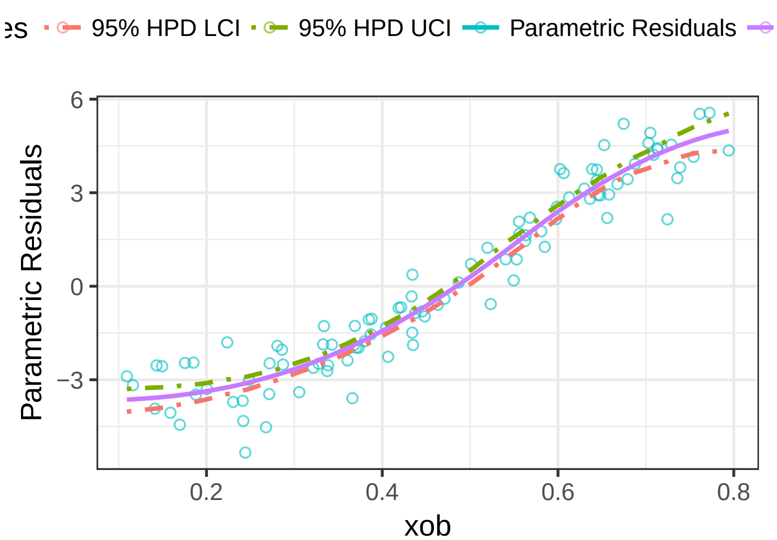
```
plot(fout)
```


Beta
3.4
3.5
3.6
3.7
3.8
10000
10500
11000
11500
12000
Sigma
0.45
0.50
0.55
0.60
0.65
10000
10500
11000
11500
12000
Function 1: Theta
0
2
4
6
10000
10500
11000
11500
12000
Function 1: Tau
0.5
1.0
1.5
2.0
2.5
3.0
3.5
10000
10500
11000
11500
12000
Function 1: Gamma
0.50
0.55
0.60
0.65
0.70
0.75
0.80
10000
10500
11000
11500
12000
Function 1: Zeta
−10.5
−10.0
−9.5
−9.0
−8.5
10000
10500
11000
11500
12000

We may visualize fitted median 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* .