Example on Bayesian Spectral Analysis Density Estimation

Statlab2

2017-08-04

Preliminaries

To use `bsamGP`, you need to first install the package. Use RGui menu "Package/Install package(s) from local zip files…" or R command "install.packages".

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 `bsad` function, type one of the following:

```
help(bsad) # ?bsad
```

Bayesian Spectral Analysis Density Estimation (BSAD)

Let's now proceed to a density estimating via the following BSAD model.

$$f_\infty(y|\beta, Z) = \frac{\exp\left[\hat{h}(y)'\beta + Z(y)\right]}{\int_{\mathcal{Y}} \exp\left[\hat{h}(x)'\beta + Z(x)\right] dG(x)}, \tag{1}$$

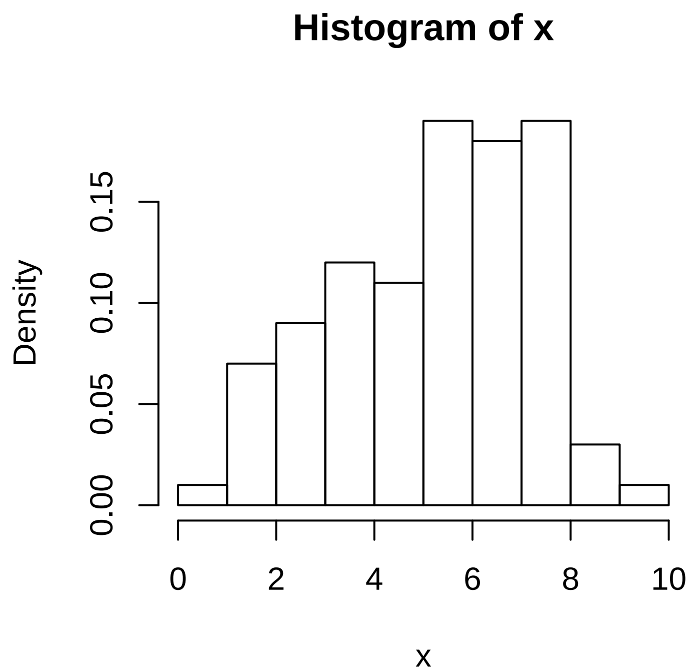
where Z is a zero mean, second-order Gaussian process with bounded, continuous covariate function:$E[Z(x), Z(y)] = \sigma(x, y)$, and $\int_{\mathcal{Y}} ZdG = 0$ almost surely to identify the model.

Data generation

We consider 100 observations from a semiparametric density whose parametric component has a truncated gamma distribution with $y \in [0, 10]$ and nonparametric component has a sigmoidal density.

$$f(y) \propto \exp\{3\log(y) - y + \tanh(y-5)\}$$

```
set.seed(1)
n = 100; a = 0; b = 10;
semiFunc = function(x, xMin, xMax) exp(3*log(x-xMin) - x + tanh(x - 5))
InvAccRate = optimize(function(x) semiFunc(x,a,b)/dunif(x,a,b),
                      maximum=TRUE, interval=c(a,b))$objective
u = runif(n*InvAccRate); y = runif(n*InvAccRate, a,b)
x = y[u<semiFunc(y,a,b)/(InvAccRate*dunif(y,a,b))][1:n]
hist(x, prob=TRUE)
```



Model fitting

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

```
# MCMC parameters
nblow = 10000; # Number of MCMC in transition period
smcmc = 1000; # Number of MCMC for analysis
nskip = 5; # Number of MCMC to skip after nblow
ndisp = 1000; # Number of saved draws to be displayed on screen
kappaloop = 5; # Maximum number of times to modify metropolis

# Prior information
parametric = 'gamma' # parametric distribution to be test.
smoother = 'geometric'; # Types of smoothing priors
MaxNCos = 98; # Maximum number of cosine basis functions
```

To generate a posterior sample for the Bayesian semiparametric density estimation model, use the function `bsad`.

```
# Fit the model
fout = bsad(x=x, xmin=a, xmax=b, MaxNCos=MaxNCos,
            mcmc=list(kappaloop=kappaloop,nblow=nblow,smcmc=smcmc,nskip=nskip,ndisp=ndisp),
            parametric=parametric, smoother=smoother)
```

```
## Burnin ...
## Main iterations ...
## MCMC draws 1000 of 1000 (CPU time: 182.312 s)
```

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

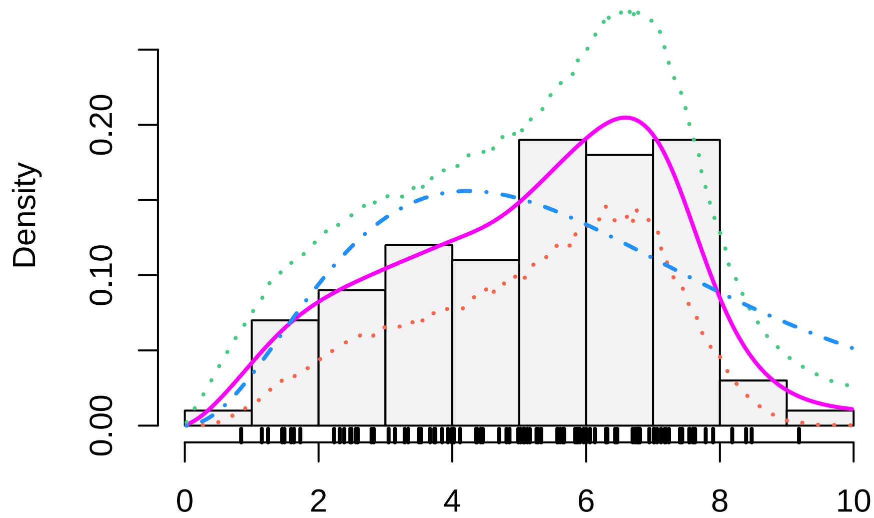
```r
print(fout)
```

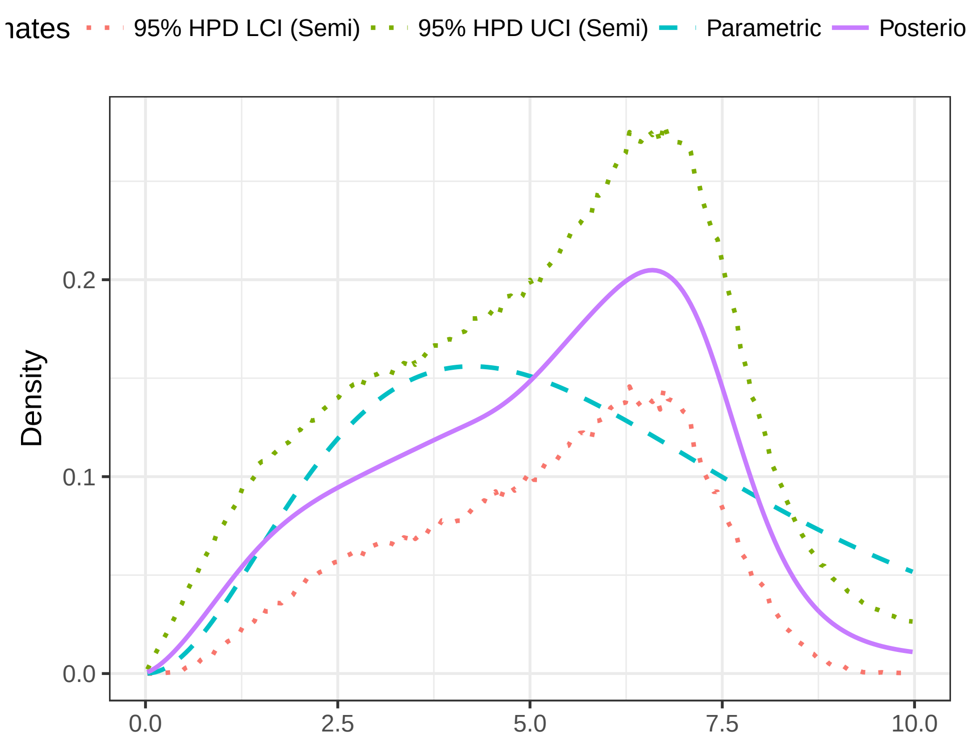
```
##
## Call:
## bsad.default(x = x, xmin = a, xmax = b, MaxNCos = MaxNCos, mcmc = list(kappaloop = kappaloop,
##     nblow = nblow, smcmc = smcmc, nskip = nskip, ndisp = ndisp),
##     smoother = smoother, parametric = parametric)
##
## Bayesian Spectral Analysis Density Estimation (BSAD)
## - Density Estimation using Log-Gaussian Process -
##
## Parametric model is gamma
## ----------------------------------------------------------------------
##
## # of cosines Kappa is random on 0, 1, ..., MaxNCos
##
## f(x) = exp[Z(x)] /int exp[Z(s)] ds
## Discretize problem
## f = exp(Y_i)/(sum exp(y_j) delta)
## Use semiparametric regression model for Y
## Y = D*beta + Phi*Z*theta + epsilon with Z = 0 or 1
## [epsilon] = N(0,sigma^2I)
## Smoothing prior on theta
## Uses Slice Sampling to generate Y
##
## ----------------------------------------------------------------------
##
## Number of observations   = 100
## Estimation Interval: min = 0 and max = 10
## Number of intervals      = 201
## Number of cosine terms   = 98
##
## ----------------------------------------------------------------------
##
## Number of transition iterations           = 60000
## Number of iterations for analysis         = 1000
## Number of loops between generating kappa  = 5
## Number of loops between saved interations = 25
## Total number of MCMC iterations           = 66000
##
## **********************************************************************
##
## Posterior Probabilities of Parametric vs Semi-parametric with kappa
##        Para    SemiPara
## 0.001867983 0.998132017
## **********************************************************************
##
## Ln Marginal Distribution of the Data
##      Para      Semi    SemiMK
## -215.5060 -221.3259 -211.5873
##
## ======================================================================
##
## Parametric Model
```

```
##
## Regression Coefficients
##             PostMean       STD       2.5%        50%      97.5%
## ln(X-Min)  2.2894796 0.3510975  1.5215282  2.3229449  2.8888816
## X-Min     -0.5384524 0.0910226 -0.7155857 -0.5438243 -0.3418268
##
## ==================================================================
##
## Semiparametric Model: Estimation of Kappa
##
## Regression Coefficients
##             PostMean       STD       2.5%        50%      97.5%
## ln(X-Min)  1.8984606 0.5950549  0.6306879  1.9533168  2.9007351
## X-Min     -0.6928783 0.2504685 -1.2571044 -0.6736692 -0.2345863
##
## ------------------------------------------------------------------
##
## Model Order kappa
## PostMean      STD     2.5%      50%    97.5%
## 43.65300 26.23076  5.00000 40.00000 95.00000
##
## Smoothing Parameter tau
##  PostMean       STD      2.5%       50%     97.5%
## 1.3096468 0.5823363 0.6480343 1.1661397 2.7856724
##
## Smoothing parameter gamma
##  PostMean       STD      2.5%       50%     97.5%
## 1.0074618 0.2585011 0.4983391 0.9936630 1.5691742
##
## ==================================================================
##
## Semiparametric Model: Kappa = Max # of theta
##
## Regression Coefficients
##            PostMean       STD       2.5%        50%      97.5%
## ln(X-Min)  1.938417 0.5641073  0.7080057  2.0102665  2.8394299
## X-Min     -0.711857 0.2390608 -1.1952136 -0.7092299 -0.2550209
##
## ------------------------------------------------------------------
##
## Smoothing Parameter tau using Max Kappa
##  PostMean       STD      2.5%       50%     97.5%
## 1.5240110 0.6284271 0.7139262 1.3892521 3.1146218
##
## Smoothing parameter gamma using Max Kappa
##  PostMean       STD      2.5%       50%     97.5%
## 1.3007401 0.2366695 0.8422467 1.3390709 1.7093227
##
## ******************************************************************
```

We may visualize estimated density for both parametric and semiparametric models superimposed with 95% highest posterior density (HPD) interval using `plot` function on fitted object from `fitted` method.

```
fit = fitted(fout)
plot(fit)
```





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* .