Generalized Bayesian Additive Spline Surfaces (GBASS)

Kellin Rumsey

6/23/2022

Robust Regression

```
X <- matrix(runif(2*100, 0, 1), ncol=2)           #Simulate x values
y <- apply(X, 1, GBASS::ff1) + rnorm(100, 0, 1)   #Simulate y values
y[1:3] <- rnorm(3, 0, sd=sqrt(10))                #Simulate 3 outliers

# Fit a BASS model with a t likelihood
df = 5                                            #Degrees of freedom
mod1 <- tbass(X, y, df=5, nmcmc=2000, nburn=1001, verbose=FALSE)
```

```
## Warning in gbass(X, y, v_prior = v_prior, scale = scale, ...): maxInt cannot
## exceed ncol(X). Running with maxInt = ncol(X)
```

```
# Fit a BASS model with a Horseshoe likelihood
mod2 <- hbass(X, y, nmcmc=2000, nburn=1001, verbose=FALSE)
```

```
## Warning in gbass(X, y, w_prior = w_prior, v_prior = v_prior, scale = scale, :
## maxInt cannot exceed ncol(X). Running with maxInt = ncol(X)
```

Quantile Regression

```
library(BASS)
```

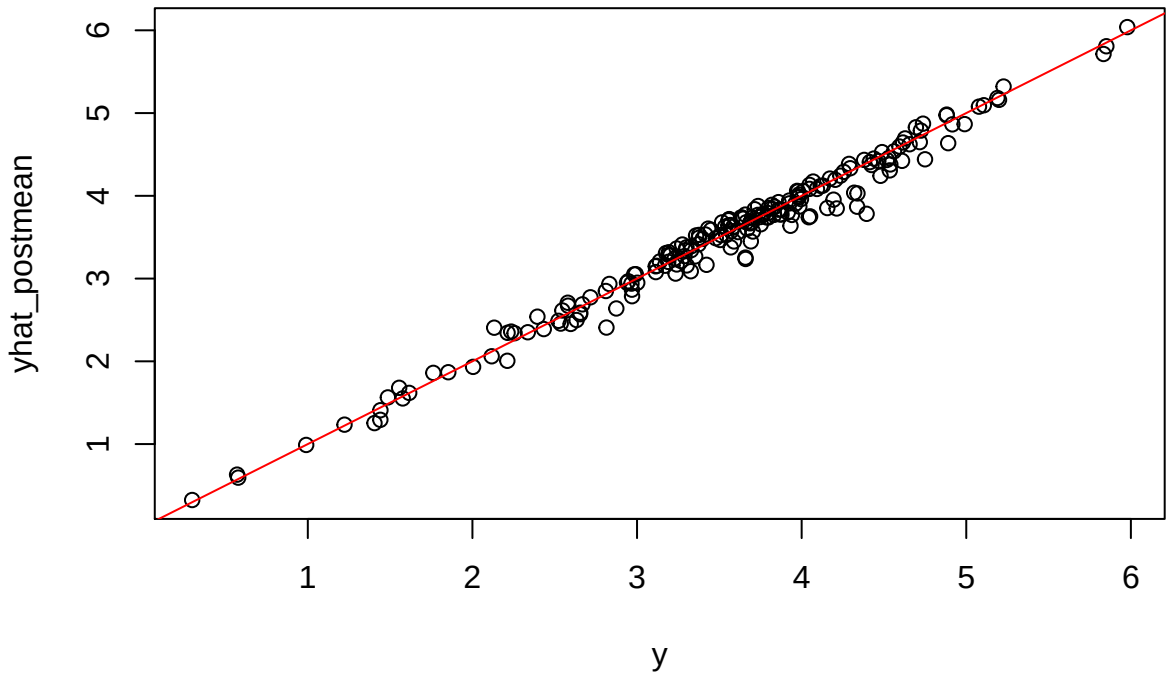
```
##
## Attaching package: 'BASS'
```

```
## The following object is masked from 'package:GBASS':
##
##     sobol
```

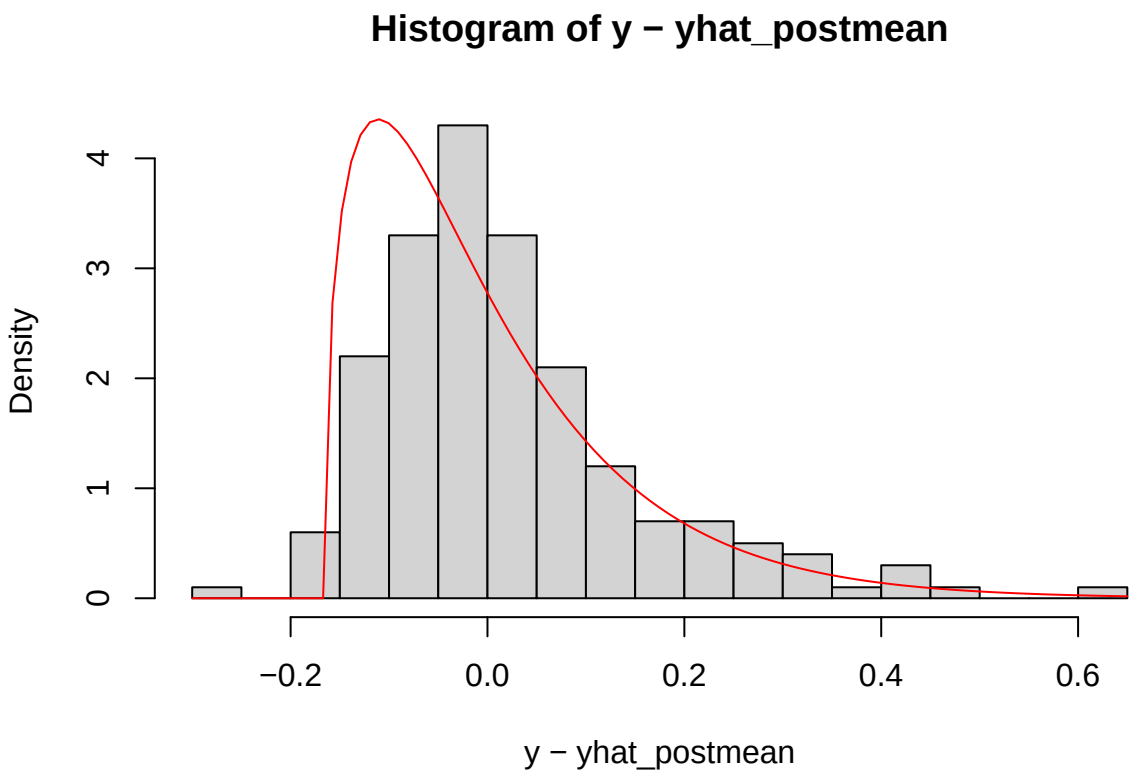
```
library(MASS)
X <- MASS::mcycle$times
X <- BASS:::scale.range(X)
y <- MASS::mcycle$accel

# Fit BASS model for the 80^th percentile of the response distribution
mod3 <- qbass(X, y, q=0.8,
        maxInt=1,
        a_lambda=.03, b_lambda=.03,
        nmcmc=2000, nburn=1001, verbose=FALSE)
```

Flexible Liklihood Regression

```
X <- matrix(runif(2*200, 0, 1), ncol=2)              #Simulate x values
y <- apply(X, 1, GBASS::ff1)                        #Simulate y values
y <- y + (rgamma(200, 1.5, 9) - 1/6)                #Add (mean zero) skewed errors

# Fit a BASS model with a Normal-Wald likelihood (paper version)
mod4 <- nwbass(X, y, nmcmc=5000, nburn=4001, lag_beta = 500, maxInt=2, verbose=FALSE)

# Alternate version: Enforces that the error distribution is mean zero
mod5 <- nwbass2(X, y, nmcmc=5000, nburn=4001, lag_beta = 500, maxInt=2, verbose=FALSE)

# CODE BELOW NOT RUN
yhat <- predict(mod5, X)
yhat_postmean <- apply(yhat, 1, mean)

# Mean function has been learned
plot(y, yhat_postmean)
abline(0, 1, col='red')
```

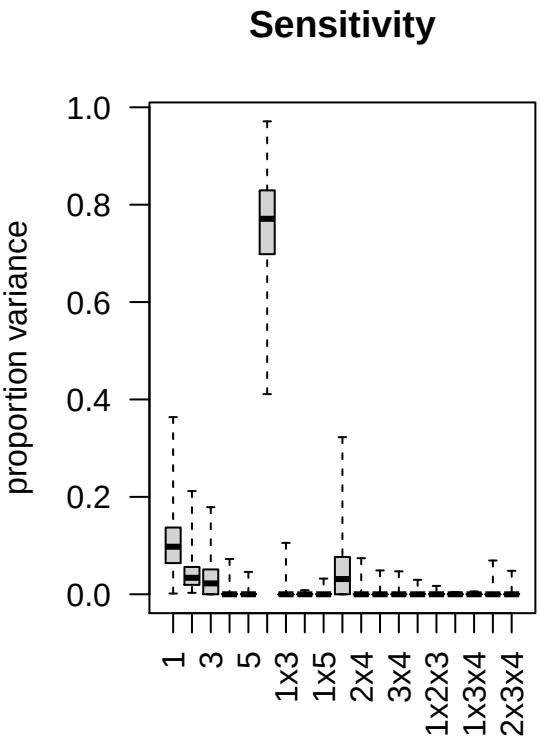


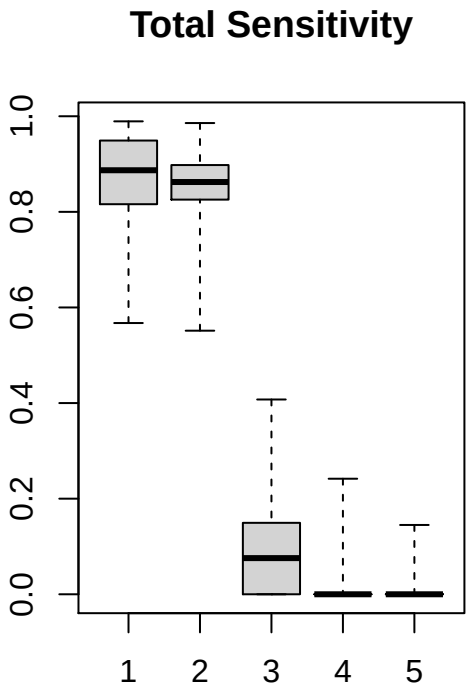

```
# Error distribution has been learned
hist(y-yhat_postmean,
     freq=FALSE, breaks=30)
curve(dgamma(x+1/6, 1.5, 9), add=TRUE, col='red')
```



Sensitivity Analysis

```
p <- 5                                          #Number of inputs (only first two are active)
X <- matrix(runif(p*100, 0, 1), ncol=p)         #Simulate x values
y <- apply(X, 1, GBASS::ff1) + rnorm(100, 0, 1)   #Simulate y values

# Fit a model with a t(30) likelihood
mod6 <- tbass(X, y, df=30, nmcmc=2000, nburn=1001, verbose=FALSE)

# Convert "gbass" object to "bass" object
mod6_bass <- gm2bm(mod6)

# Get sobol indices
sob <- GBASS::sobol(mod6_bass, verbose=FALSE)
plot(sob)
```


Sensitivity
proportion variance
1.0
0.8
0.6
0.4
0.2
0.0
1
3
5
1x3
1x5
2x4
3x4
1x2x3
1x3x4
2x3x4


Total Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
1
2
3
4
5