library(randomForest)

##Data Generation

n <- 100 ##training sample size

n.pred <- 50 ##test sample size

p <- 100 ##number of predictors

sig <- 0.5 ##error variance

X <- matrix(NA,n+n.pred,p)

for(i in 1:(n+n.pred)){

X[i,] <- runif(p)

}

reg.mean <- 10\*sin(pi\*X[,1]\*X[,2])+20\*(X[,3]-0.5)^2+10\*X[,4]+5\*X[,5]

y <- reg.mean+rnorm(n+n.pred,0,sig)

y.train <- y[1:n] ##training response

X.train <- X[1:n,] ##training predictors

y.test <- y[(n+1):(n+n.pred)] ##test response

X.test <- X[(n+1):(n+n.pred),]##test predictors

#X.train <- X[1:n,]+rnorm(n,0,1) ##noisy predictors

##########Random Forest##########

df <- data.frame(y.train,X.train)

CRForest <- randomForest(y.train~., data=df, mtry=max(floor(ncol(X.train)/3), 1),

ntree=10000)

Cpredicted.test <- predict(CRForest, data.frame(X.test), type="response",

norm.votes=TRUE, predict.all=FALSE, proximity=FALSE, nodes=FALSE)

Cpredicted.train <- predict(CRForest, data.frame(X.train), type="response",

norm.votes=TRUE, predict.all=FALSE, proximity=FALSE, nodes=FALSE)

MSPE.out.sample <- mean((as.numeric(Cpredicted.test)-y.test)^2)

MSPE.in.sample <- mean((as.numeric(Cpredicted.train)-y.train)^2)

MSPE.out.sample

MSPE.in.sample

## BART

library(BayesTree)

post.burn.in <- 10000

burn.in <- 1000

num.tree <- 100

power.par.tree.prior <- 2

base.par.tree.prior <- 0.95

RT.BART <- bart(X.train, y.train, X.test,sigest=2, sigdf=3, sigquant=.90,

k=2.0,power=power.par.tree.prior,base=base.par.tree.prior,

binaryOffset=0,ntree=num.tree,

ndpost=post.burn.in, nskip=burn.in,printevery=100, keepevery=1,

keeptrainfits=TRUE,usequants=FALSE, numcut=100, printcutoffs=0,

verbose=TRUE)

ytest.BART <- RT.BART$yhat.test ##posterior predictive samples at test covaraites

quant.BART <- apply(ytest.BART,2,quantile,c(.025,.975))

length.BART <- mean(quant.BART[2,]-quant.BART[1,])

coverage.BART <- length(intersect(which(y.test>quant.BART[1,]),which(y.test<quant.BART[2,])))/n.pred

MSPE.BART <- mean((y.test-colMeans(ytest.BART))^2)

MSPE.BART ## MSPE

coverage.BART ##coverage of 95% PI

length.BART ##length of 95% PI

plot(RT.BART$sigma,type="l",col="red",xlab="post burn in iterations",

ylab=expression(paste(sigma)))