Illustration of crossvalidation

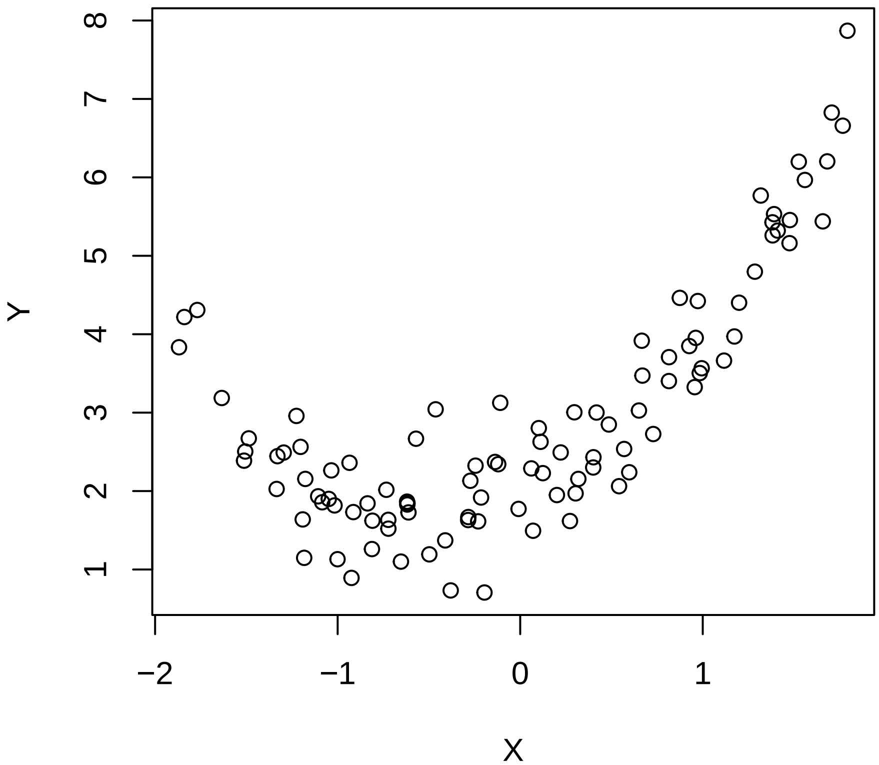
Arnab Hazra

3/27/2022

Here we simulate 100 observations from a simple linear regression $Y_i = \mu_i + \epsilon_i, i = 1, \ldots, 100$. Here, $\mu_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2$ and $\epsilon_i \sim \text{Normal}(0, \sigma^2)$.

1. Simulate data

```
rm(list = ls())

n <- 100

set.seed(100)

X <- runif(n)
X <- scale(X)

beta <- c(2, 1, 1) # (beta0, beta1, beta2)
sigmaSq <- 0.25

Y <- beta[1] + beta[2] * X + beta[3] * X^2 + rnorm(n, mean = 0, sd = sqrt(sigmaSq))

plot(X, Y)
```



Now, we want to compare the models $\mathcal{M}_1 : \mu_i = \beta_0 + \beta_1 X_i$ versus $\mathcal{M}_2 : \mu_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2$.

2. Split the data into training and testing

```
test <- rep(1:2, n)[1:n]

Ytrain <- Y[test == 1]
Xtrain <- X[test == 1]

Ytest <- Y[test == 2]
Xtest <- X[test == 2]

ntrain <- length(Ytrain)
ntest <- length(Ytest)

ntrain
```

```
## [1] 50
```

```
ntest
```

```
## [1] 50
```

3. Specify priors for two competing models:

Model 1 has noninformative Gaussian priors $\beta_0, \beta_1 \sim \text{Normal}(0, 100^2)$.

Model 2 has noninformative Gaussian priors $\beta_0, \beta_1, \beta_2 \sim \text{Normal}(0, 100^2)$

```r
# M1
model_string1 <- "model{

 # Likelihood
 for(i in 1:ntrain){
   Ytrain[i] ~ dnorm(mutrain[i], inv.var)
   mutrain[i] <- beta0 + beta1 * Xtrain[i]
 }

 #Prediction
 for(i in 1:ntest){
   Ytest[i] ~ dnorm(mutest[i],inv.var)
   mutest[i] <- beta0 + beta1 * Xtest[i]
 }

 #Priors
 beta0 ~ dnorm(0,0.00001)
 beta1 ~ dnorm(0,0.00001)
 inv.var ~ dgamma(0.01,0.01)
}"

# M2
model_string2 <- "model{

 # Likelihood
 for(i in 1:ntrain){
   Ytrain[i] ~ dnorm(mutrain[i],inv.var)
   mutrain[i] <- beta0 + beta1 * Xtrain[i] + beta2 * Xtrain[i]^2
 }

 #Prediction
 for(i in 1:ntest){
   Ytest[i] ~ dnorm(mutest[i],inv.var)
   mutest[i] <- beta0 + beta1 * Xtest[i] + beta2 * Xtest[i]^2
 }

 #Priors

 #Priors
 beta0 ~ dnorm(0,0.00001)
 beta1 ~ dnorm(0,0.00001)
 beta2 ~ dnorm(0,0.00001)
 inv.var ~ dgamma(0.01,0.01)
}"
```

4. Fit the two models

```r
library(rjags)
```

```
## Loading required package: coda
## Linked to JAGS 4.3.0
## Loaded modules: basemod,bugs
```

```r
model1 <- jags.model(textConnection(model_string1),
                     data = list(Ytrain = Ytrain, Xtrain = Xtrain, ntrain = ntrain,
                                 Xtest = Xtest, ntest = ntest), quiet = TRUE)
update(model1, 10000, progress.bar = "none")
samps1 <- coda.samples(model1, variable.names = c("Ytest"),
           n.iter = 20000, progress.bar = "none")
Ytest1  <- samps1[[1]]

model2 <- jags.model(textConnection(model_string2),
                     data = list(Ytrain = Ytrain, Xtrain = Xtrain, ntrain = ntrain,
                                 Xtest = Xtest, ntest = ntest), quiet = TRUE)
update(model2, 10000, progress.bar = "none")
samps2  <- coda.samples(model2, variable.names = c("Ytest"),
           n.iter = 20000, progress.bar = "none")
Ytest2  <- samps2[[1]]
```

5. Compile the results

```r
post_mn1 <- apply(Ytest1, 2, mean)
post_sd1 <- apply(Ytest1, 2, sd)
post_low1 <- apply(Ytest1, 2, quantile, 0.05)
post_high1 <- apply(Ytest1, 2, quantile, 0.95)

post_mn2 <- apply(Ytest2, 2, mean)
post_sd2 <- apply(Ytest2, 2, sd)
post_low2 <- apply(Ytest2, 2, quantile, 0.05)
post_high2 <- apply(Ytest2, 2, quantile, 0.95)

MSE1 <- mean((post_mn1 - Ytest)^2)
BIAS1 <- mean(post_mn1 - Ytest)
AVESD1 <- mean(post_sd1)
COV1 <- mean((Ytest > post_low1) & (Ytest < post_high1))

MSE2 <- mean((post_mn2 - Ytest)^2)
BIAS2 <- mean(post_mn2 - Ytest)
AVESD2 <- mean(post_sd2)
COV2 <- mean((Ytest > post_low2) & (Ytest < post_high2))

MSE <- c(MSE1, MSE2)
BIAS <- c(BIAS1, BIAS2)
AVESD <- c(AVESD1, AVESD2)
COV90 <- c(COV1, COV2)

OUTPUT <- cbind(MSE, BIAS, AVESD, COV90)
rownames(OUTPUT) <- c("Linear regression", "Quadratic regression")
library(kableExtra)
kable(OUTPUT, digits = 2)
```

	MSE	BIAS	AVESD	COV90
Linear regression	1.15	-0.39	1.21	0.92
Quadratic regression	0.34	-0.14	0.47	0.80

The Bayesian quadratic regression has smaller prediction mean squared error.