# Illustration of DIC and WAIC

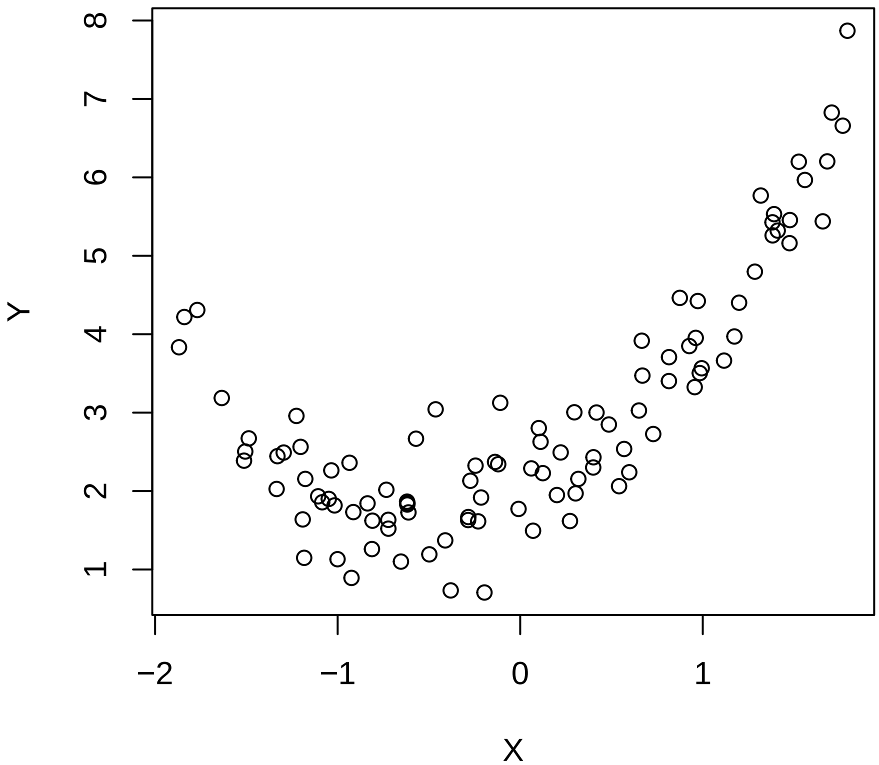
Arnab Hazra

3/28/2022

Here we simulate 100 observations from a quadratic 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 <- as.vector(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. Specify priors for two competing models:

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

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

```
library(rjags)
```

```
## Loading required package: coda

## Linked to JAGS 4.3.0

## Loaded modules: basemod,bugs
```

```
# M1
model_string1 <- "model{

 # Likelihood
 for(i in 1:n){
   Y[i] ~ dnorm(mu[i], inv.var)
   mu[i] <- beta0 + beta1 * X[i]
 }

 #Priors
 beta0 ~ dnorm(0, 0.00001)
 beta1 ~ dnorm(0, 0.00001)
 inv.var ~ dgamma(0.01, 0.01)
 beta <- c(beta0, beta1)
 sigma <- sqrt(1 / inv.var)
}"
```

```
# M2
model_string2 <- "model{

 # Likelihood
 for(i in 1:n){
   Y[i] ~ dnorm(mu[i], inv.var)
   mu[i] <- beta0 + beta1 * X[i] + beta2 * X[i]^2
 }

 #Priors
 beta0 ~ dnorm(0, 0.00001)
 beta1 ~ dnorm(0, 0.00001)
 beta2 ~ dnorm(0, 0.00001)
 inv.var ~ dgamma(0.01, 0.01)
 beta <- c(beta0, beta1, beta2)
 sigma <- sqrt(1 / inv.var)
}"
```

## 3. Fit the two models

```
data <- list(Y = Y, X = X, n = n)

model1 <- jags.model(textConnection(model_string1),
                     data = data, n.chains = 1, quiet = TRUE)
update(model1, 10000, progress.bar = "none")
samps1 <- coda.samples(model1, variable.names = c("beta", "sigma"),
                       n.iter = 20000, thin = 5, progress.bar = "none")
beta.vec.m1  <- samps1[[1]][ , 1:2]
sigma.m1 <- samps1[[1]][ , 3]

model2 <- jags.model(textConnection(model_string2),
                     data = data, n.chains = 1, quiet = TRUE)
update(model2, 10000, progress.bar = "none")
samps2 <- coda.samples(model2, variable.names = c("beta", "sigma"),
                       n.iter = 20000, thin = 5, progress.bar = "none")
beta.vec.m2 <- samps2[[1]][ , 1:3]
sigma.m2 <- samps2[[1]][ , 4]
```

## 4. calculate DIC for both the models

Let $\bar{D} = \mathrm{E}[D(Y|\theta)|Y]$ be the posterior mean of the deviance.

Denote $\hat{\theta}$ as the posterior mean of $\theta$.

The effective number of parameters is

$$p_D = \bar{D} - D(Y|\hat{\theta}).$$

DIC can be written like $AIC$,

$$DIC = \bar{D} + p_D = D(Y|\hat{\theta}) + 2p_D.$$

Models with small $\bar{D}$ fit the data well.

Models with small $p_D$ are simple.

We prefer models that are simple and fit well, so we select the model with smallest $DIC$.

```
# after thinning, 4K post-burn-in samples left

loglike.m1 <- sapply(1:4000, function(iter){
  sum(dnorm(Y, mean = beta.vec.m1[iter, 1] + beta.vec.m1[iter, 2] * X,
            sd = sigma.m1[iter], log = T))})

loglike.m2 <- sapply(1:4000, function(iter){
  sum(dnorm(Y, mean = beta.vec.m2[iter, 1] + beta.vec.m2[iter, 2] * X +
             beta.vec.m2[iter, 3] * X^2, sd = sigma.m2[iter], log = T))})

deviance.m1 <- -2 * loglike.m1
deviance.m2 <- -2 * loglike.m2

# DIC

Dbar.m1 <- mean(deviance.m1)
Dbar.m2 <- mean(deviance.m2)

D.thetahat.m1 <- sum(dnorm(Y, mean = mean(beta.vec.m1[ , 1]) +
                              mean(beta.vec.m1[ , 2]) * X,
                              sd = mean(sigma.m1), log = T))

D.thetahat.m2 <- sum(dnorm(Y, mean = mean(beta.vec.m2[ , 1]) +
                              mean(beta.vec.m2[ , 2]) * X +
                              mean(beta.vec.m2[ , 3]) * X^2,
                              sd = mean(sigma.m2), log = T))

pD.m1 <- Dbar.m1 - D.thetahat.m1
pD.m2 <- Dbar.m2 - D.thetahat.m2

DIC.m1 <- pD.m1 + Dbar.m1
DIC.m2 <- pD.m2 + Dbar.m2

DIC.m1
```

```
## [1] 758.2098
```

```
DIC.m2 # smaller, indicates that M2 is preferred
```

```
## [1] 371.6763
```

$WAIC$ is written in terms of the posterior of the likelihood rather than parameters.

Let $m_i$ and $v_i$ be the posterior mean and variance of

$$\log[f(Y_i|\theta)].$$

The effective model size is $p_W = \sum_{i=1}^{n} v_i$.

The criteria is

$$WAIC = -2\sum_{i=1}^{n} m_i + 2p_W.$$

## 5. calculate WAIC for both the models

```
# after thinning, 4K post-burn-in samples left

loglike.m1 <- sapply(1:4000, function(iter){
  dnorm(Y, mean = beta.vec.m1[iter, 1] + beta.vec.m1[iter, 2] * X,
             sd = sigma.m1[iter], log = T)})

loglike.m2 <- sapply(1:4000, function(iter){
  dnorm(Y, mean = beta.vec.m2[iter, 1] + beta.vec.m2[iter, 2] * X +
              beta.vec.m2[iter, 3] * X^2, sd = sigma.m2[iter], log = T)})

# WAIC
posmeans.m1 <- apply(loglike.m1, 1, mean)
posmeans.m2 <- apply(loglike.m2, 1, mean)

posvars.m1 <- apply(loglike.m1, 1, var)
posvars.m2 <- apply(loglike.m2, 1, var)

pW.m1 <- sum(posvars.m1)
pW.m2 <- sum(posvars.m2)

sum.means.m1 <- sum(posmeans.m1)
sum.means.m2 <- sum(posmeans.m2)

WAIC.m1 <- -2 * sum.means.m1 + 2 * pW.m1
WAIC.m2 <- -2 * sum.means.m2 + 2 * pW.m2

WAIC.m1
```

```
## [1] 312.466
```

```
WAIC.m2 # smaller, indicates that M2 is preferred
```

```
## [1] 157.3232
```

## 6. Final comparison table

```
OUT <- rbind(c(DIC.m1, WAIC.m1), c(DIC.m2, WAIC.m2))
OUT <- round(OUT, 2)
 rownames(OUT) <- c("Linear regression", "Quadratic regression")
 colnames(OUT) <- c("DIC", "WAIC")
 library(kableExtra)
 kable(OUT)
```

| | DIC | WAIC |
|---|---|---|
| Linear regression | 758.21 | 312.47 |
| Quadratic regression | 371.68 | 157.32 |

Smaller DIC and WAIC values indicate that the quadratic regression model is superior than the linear regression model.