APTS Statistical Modelling: Practical 1

A. M. Overstall

April 2018

Suppose

$$y_{im} \sim \text{Poisson}\left(\mu(x_{im})\right),$$

independently, for $i = 1, \dots, n$ and $m = 1, \dots, M$, where

$$\mu(x_{im}) = 8 \exp\left(q(x_{im})\right),$$

for some function q.

Suppose $x_{im} = x_i = -10 + 20(i-1)/(n-1)$, $M = 3$ and

$$q(x) = 0.001\left(100 + x + x^2 + x^3\right).$$

The code below performs the following simulation study. For $b = 1, \dots, B$:

- For $i = 1, \dots, n$ and $m = 1, \dots, M$, generate

$$y_{im} \sim \text{Poisson}(\mu(x_{im})).$$

- Record the AIC for models

$$\begin{aligned} y_{im} &\sim \text{Poisson}(\mu(x_{im})) \\ \mu(x_{im}) &= \exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{im}^j\right), \end{aligned}$$

  for $p = 0, \dots, p_{\max} = 20$.

```
> B <- 1000
> n <- 25
> M <- 3
> pmax <- 20
>
> x <- rep(seq(from = -10, to = 10, length = n), each = M)
>
> mu <- function(x) {
+     8 * exp(q(x))
+ }
>
> q <- function(x) {
+     0.001 * (100 + x + x^2 + x^3)
+ }
>
> aics <- matrix(0, nrow = B, ncol = pmax)
```

```
> for (b in 1:B) {

+     y <- rpois(n = M * n, lambda = mu(x))

+     mod <- glm(y ~ 1, family = poisson)
+     aics[b, 1] <- AIC(mod)

+     formula <- "y~x"
+     mod <- glm(formula, family = poisson)
+     aics[b, 2] <- AIC(mod)

+     for (j in 3:pmax) {
+         formula <- paste(formula, " + I(x^", j - 1, ")", sep = "")
+         mod <- glm(formula, family = poisson)
+         aics[b, j] <- AIC(mod)
+     }

+ }
>
> AICorder <- apply(aics, 1, which.min) - 1
> tAIC <- table(AICorder)
> tAIC
```

1. Investigate the performance of AIC as a model selection tool for $n = 25, 50, 100, 1000$.

2. Vary the simulation model, using

$$q(x) = \frac{1.2}{1+\exp(-x)},$$

   to see how AIC performs when the fitted models do not include the simulation model.

3. Modify the code above to compute the values of BIC and AIC_c, where

$$\text{AIC}_c = \text{AIC} + \frac{2p^2+2p}{n-p-1}.$$

Solutions

1. $n = 25$

```
## AICorder
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   1   2 689 131  46  32  24  12  15  10  10   8   4   3   6   2   3   1
##  19
##   1
```

$n = 50$

```
## AICorder
##   3   4   5   6   7   8   9  10  11  12  13  14  15  17  19
## 717 119  57  37  17  13  10  10   2   4   3   4   4   1   2
```

$n = 100$

```
## AICorder
##   3   4   5   6   7   8   9  10  11  12  13  14  15  16  18  19
## 714 108  61  29  25  21   6  10   7   3   3   3   1   2   2   5
```

$n = 1000$

```
## AICorder
##   3   4   5   6   7   8   9  10  11  12  14  15  16  17  18  19
## 717 108  59  34  14  29  12  10   4   2   5   1   1   1   2   1
```

$n = 10000$

```
## AICorder
##   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
## 729 110  57  36  24   9   9   7   3   4   2   2   2   2   1   2   1
```

2. $n = 25$

```
> B <- 1000
> n <- 25
> M <- 3
> pmax <- 20
>
> x <- rep(seq(from = -10, to = 10, length = n), each = M)
>
> mu <- function(x) {
+     8 * exp(q(x))
+ }
>
> q <- function(x) {
+     1.2/(1 + exp(-x))
+ }
>
```

```
> aics <- matrix(0, nrow = B, ncol = pmax)
> for (b in 1:B) {

+     y <- rpois(n = M * n, lambda = mu(x))

+     mod <- glm(y ~ 1, family = poisson)
+     aics[b, 1] <- AIC(mod)

+     formula <- "y~x"
+     mod <- glm(formula, family = poisson)
+     aics[b, 2] <- AIC(mod)

+     for (j in 3:pmax) {
+         formula <- paste(formula, " + I(x^", j - 1, ")", sep = "")
+         mod <- glm(formula, family = poisson)
+         aics[b, j] <- AIC(mod)
+     }

+ }
>
> AICorder <- apply(aics, 1, which.min) - 1
> tAIC <- table(AICorder)
> tAIC

## AICorder
##   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
##  76  24 337  90 197  67  71  45  34  19  14   7   4   4   3   4   4
```

$n = 50$

```
## AICorder
##   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
##  10   4 237  74 319  73 125  42  43  28  18  13   6   1   3   2   2
```

$n = 100$

```
## AICorder
##   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
##  68  34 342  78 225  63  73  30  39  16   3   9  12   4   4
```

$n = 1000$

```
## AICorder
##   9  10  11  12  13  14  15  16  17  18  19
## 130  49 340  87 184  52  79  28  27  12  12
```

3. $n = 25$

```
> B <- 1000
> n <- 25
> M <- 3
```

```
> pmax <- 20
>
> x <- rep(seq(from = -10, to = 10, length = n), each = M)
>
> mu <- function(x) {
+     8 * exp(q(x))
+ }
>
> q <- function(x) {
+     1.2/(1 + exp(-x))
+ }
>
> aics <- matrix(0, nrow = B, ncol = pmax)
> bics <- matrix(0, nrow = B, ncol = pmax)
> aiccs <- matrix(0, nrow = B, ncol = pmax)
> for (b in 1:B) {

+     y <- rpois(n = M * n, lambda = mu(x))

+     mod <- glm(y ~ 1, family = poisson)
+     aics[b, 1] <- AIC(mod)
+     bics[b, 1] <- BIC(mod)
+     aiccs[b, 1] <- AIC(mod) + ((2 * (1^2) + 2 * 1)/(n - 1 - 1))

+     formula <- "y~x"
+     mod <- glm(formula, family = poisson)
+     aics[b, 2] <- AIC(mod)
+     bics[b, 2] <- BIC(mod)
+     aiccs[b, 2] <- AIC(mod) + ((2 * (2^2) + 2 * 2)/(n - 2 - 1))

+     for (j in 3:pmax) {
+         formula <- paste(formula, " + I(x^", j - 1, ")", sep = "")
+         mod <- glm(formula, family = poisson)
+         aics[b, j] <- AIC(mod)
+         bics[b, j] <- BIC(mod)
+         aiccs[b, j] <- AIC(mod) + ((2 * (j^2) + 2 * j)/(n - j -
+             1))
+     }

+ }
>
> AICorder <- apply(aics, 1, which.min) - 1
> BICorder <- apply(bics, 1, which.min) - 1
> AICcorder <- apply(aiccs, 1, which.min) - 1
>
> tAIC <- table(AICorder)
> tBIC <- table(BICorder)
> tAICc <- table(AICcorder)
> tAIC

## AICorder
##    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19
```

```
##  69  34 360  97 197  60  77  28  32  14   7  10   4   4   2   3   2

> tBIC

## BICorder
##   2   3   4   5   6   7   8   9  10
##   1 426  55 405  40  60   3   9   1

> tAICc

## AICcorder
##   2   3   4   5   6   7   8   9
##   1 284  67 503  62  76   4   3
```

$n = 50$

```
## AICorder
##   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
##   4   5 240  75 303  60 134  37  49  28  17  13   9   9  10   4   3
## BICorder
##   3   4   5   6   7   8   9  10  11
## 151  25 621  57 128   8   8   1   1
## AICcorder
##   3   4   5   6   7   8   9  10  11  12  14
##  15   8 420  97 311  46  72  23   6   1   1
```

$n = 100$

```
## AICorder
##   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
##  67  34 324 101 193  77  85  29  27  17  16   6   8   7   9
## BICorder
##   3   4   5   6   7   8   9  10  11
##   5   2 577  55 326  22  10   1   2
## AICcorder
##   5   6   7   8   9  10  11  12  13  14  15  16  17
## 127  45 415 100 182  47  51  11   9   7   2   2   2
```

$n = 1000$

```
## AICorder
##   9  10  11  12  13  14  15  16  17  18  19
## 117  39 342  82 194  64  64  31  32  19  16
## BICorder
##   7   8   9  10  11  13
## 161   8 716  32  82   1
## AICcorder
##   9  10  11  12  13  14  15  16  17  18  19
## 127  39 344  82 195  63  62  29  29  16  14
```