Model Selection III: Exploratory Model Selection

Bob O'Hara

Contents

Exploring for Good Models **1**
What does a good model look like? 1
AIC and BIC in R 2
Your task 3
Using AIC/BIC 4

Final Comments **6**

Exploring for Good Models

Sometimes we don't have strong hypotheses, and instead we might be exploring which variables could have an effect. We might do this is we are interested in prediction, and not explanation, or if we want to ask some specific questions, but there are several other predictors that could have an effect.

The aim here is not to test specific hypotheses, rather it is to try to get a good model, whether it is to predict, or to explore what are probably the major drivers.

What does a good model look like?

There is a famous maxim from George Box that "All models are wrong, but some are useful". So we should look for a "good" model, but not necessarily the true model. So how do we find a good model?

As we now know, we can get a better fitting model simply by adding more parameters. But bigger models are more complicated, and more difficult to understand. But they are also less robust: models can end up explaining statistical noise, and over-estimating effects.

Thus, we want a model that fits the data well, is simple, and (ideally) is understandable. Because adding terms to a model improves the fit, there is a trade-off between fit to the data and simplicity. Fortunately we can measure both of these. The likelihood measures the fit to the model, and we can use the number of parameters as a measure of simplicity. The more parameters, the more complex the model.

There are several ways to penalise a model, depending on what we mean by a "good" model. Broadly, there are two ways of thinking of what we mean by good:

- a model that is close to the truth
- a model which predicts the data well

Unfortunately, it is impossible to develop an approach which will do both (yes, somebody actually proved this). But there are approaches to do each on them.

Criteria

For each model we can calculate a statistic that summarises its adequacy (for either prediction or closeness to the truth). The model with the lowest value of the statistic is the "best" model. Here we will use these statistics:

- *AIC*: Akaike's Information Criterion. This tries to find the model that best predicts the data
- *BIC*: Bayesian Information Criterion. This tries to find the model most likely to be true

There is some mathematics that shows that this is what these two statistics do, at least when you have a lot of data.

For these we use the log-likelihood, $l = logp(y|\theta)$, the number of parameters in the model, p, and the number of observations n.

AIC

This finds the model that would best predict replicate data, from exactly the same conditions. It is calculated as:

$$AIC = -2l + 2p$$

or AIC = -2 Likelihood + 2 Number of Parameters.

If the data set is small, we can use the corrected AIC:

$$AIC_c = -2l + 2p + \frac{2p(p+1)}{n-p-1}$$

which should do a better job of finding a good model.

BIC

This is designed to find the model which is most likely to be "true". Thanks to George Box, we know that we can't find the truth, so we are perhaps looking for the model closest to this. it is calculated as

$$BIC = -2l + \log(n)p$$

i.e. BIC = -2 Likelihood + log(N) Number of Parameters

So, we can see that AIC penalises the likelihood with $2p$, and BIC penailises with $\log(n)p$. So BIC penalises more if $\log(n) > 2$, i.e. if $n > 7$. This means that BIC will usually find that smaller models are optimal.

If we are strict, we should always select the best model. But statistics is rarely that neat. By chance the best model might have a slightly higher (i.e. worse) AIC or BIC. The usualy guideline is that if the difference between two models is less than 2, they are roughly the same.

AIC and BIC in R

For an example we will use simulated data to look at this approach, with 100 observations and 20 predictors. Only the first predictor has a real effect.

```
# Just like in the previous part, we set a seed so you all get the same numbers
set.seed(25)
# Then set up the number of observations
NSmall <- 100
# and the number of predictors
PSmall <- 20
# Then we make the predictor data
xSmall <- matrix(rnorm(NSmall*PSmall), nrow=NSmall)
mu <- 0.1*xSmall[,1] # true R^2 = 0.1^2/(0.1^2 + 1) = 1%
# Finally we make a response variable
ySmall <- rnorm(NSmall, mu)
```

We can extract AIC from a model object with the `AIC()`function:

```
model.null <- lm(ySmall ~ 1) # makes a null model
model.full <- lm(ySmall ~ xSmall) # makes the full model
model.2 <- lm(ySmall ~ xSmall[,2]) # makes a model with one predictor

AIC(model.null, model.2, model.full) # compare using the AIC
```

```
           df      AIC
model.null  2 296.1408
model.2     3 294.0873
model.full 22 314.3215
```

A lower value is better, so the null model is better than having all of the variables in it, and the model with variable 2 is slightly better still.

If we want BIC, we can use the `BIC()` function:

```
BIC(model.null, model.2, model.full) # compare using the BIC
```

```
##            df      BIC
## model.null  2 301.3512
## model.2     3 301.9028
## model.full 22 371.6352
```

So here we see that the null model and model 2 have almost the same BIC.

Your task

Fit all of the models with one covariate (i.e. `y~x[,1]`, `y~x[,2]` etc.). Which one gives the best model (i.e. has the lowest AIC)?

Code Hint

```
model.1 <- lm(ySmall ~ xSmall[,1])
model.2 <- lm(ySmall ~ xSmall[,2])
model.3 <- lm(ySmall ~ xSmall[,3])
# ... up to
model.20 <- lm(ySmall ~ xSmall[,20])

AIC(model.1, model.2, model.3, model.20)
```

(if you are more comfortable with R, you could use `apply()`)

Show me the `apply()` thing

```
modelsAIC <- apply(xSmall, 2, function(X, y) {
  AIC(lm(y ~ X))
}, y=ySmall)
which(modelsAIC==min(modelsAIC))
round(modelsAIC-min(modelsAIC), 2)
```

With AIC the first model is the best. If we look at the differnces in AIC, we can see that a modelw ith the second covaraite is fairly close, but the others are much worse.

Answer

If you want to watch the video, it's below, or you can click here

First, we will fit all of the models. This is the "short" way to do it:

```
modelsAIC <- apply(xSmall, 2, function(X, y) {
  AIC(lm(y ~ X))
}, y=ySmall)
```

(don't worry if you used `model.1 <- lm(ySmall ~ xSmall[,1])` 20 times)

So `modelsAIC` is a vector of AICs. We can see which has the lowest AIC like this:

```
which(modelsAIC==min(modelsAIC))
```

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

And we see that the first model has the lowest AIC.

We can also look at the differences from the lowest AIC:

```
round(modelsAIC-min(modelsAIC), 2)
```

```
##  [1] 0.00 1.85 5.45 4.44 5.77 5.81 5.47 5.12 4.13 4.00 5.90 5.67 5.80 5.89 5.84
## [16] 4.43 5.80 5.63 5.88 5.57
```

Most of the models have an AIC that is about 4-6 higher than the best model, but the model for `x2` has and AIC that is less than 2 away. There is a rule of thumb that models within about 2 of each other have roughly the same support.

Using AIC/BIC

If we have a lot of models, writing code to go through them all to find the best model is messy. Instead, we can use the `bestglm` package to do this:

```
library(bestglm) # might need install.packages("bestglm")

# This joins xSmall and ySmall together as columns
UseData <- data.frame(cbind(xSmall, ySmall))

# Use bestglm() for AIC and BIC
AllSubsetsAIC <- bestglm(Xy=UseData, IC="AIC")
AllSubsetsBIC <- bestglm(Xy=UseData, IC="BIC")
```

The `bestglm` object has several pieces:

```
names(AllSubsetsAIC)
```

```
## [1] "BestModel"   "BestModels"  "Bestq"       "qTable"      "Subsets"
## [6] "Title"       "ModelReport"
```

`Subsets` gives the AIC (or BIC) for the best models:

```
AllSubsetsAIC$BestModels
```

You can run this yourselves, but the output is big.

`BestModel` is the `lm` object for the best model, e.g.

```
coef(AllSubsetsBIC$BestModel)
```

Run the code to compare the models.

Which model is best according to AIC, and which according to BIC?

Hint

The lower the better.

Answer

If you want to watch the video, it's below, or you can click here

```
AllSubsetsAIC <- bestglm(Xy=UseData, IC="AIC")
AllSubsetsAIC$BestModel
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Coefficients:
## (Intercept)           V1           V2          V10          V16
##      0.0257       0.2580       0.2004       0.1717      -0.1380
```

According to AIC, a model with 4 variables (nos. 1,2,10 and 16) is best. What about BIC? Well...

```
AllSubsetsBIC <- bestglm(Xy=UseData, IC="BIC")
AllSubsetsBIC$BestModel
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Coefficients:
## (Intercept)           V1
##     0.04905      0.24932
```

We see that using BIC leads to a smaller model, with only V1 being in it.

How good are the models? How do they compare with the truth?

Hint

This is about interpretation, and comparing the parameter estimates with the true values (which may be zero, of course).

Answer

The true value of β_1 is 0.1, and the true R^2 should be about 1%. So first for AIC...

```
summary(AllSubsetsAIC$BestModel)
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4571 -0.6835  0.1547  0.5894  2.4038
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02570    0.10220   0.251   0.8020
## V1           0.25802    0.10005   2.579   0.0114 *
## V2           0.20044    0.10996   1.823   0.0715 .
## V10          0.17169    0.09244   1.857   0.0664 .
```

```
## V16         -0.13800    0.09633  -1.433   0.1553
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9946 on 95 degrees of freedom
## Multiple R-squared:  0.1355, Adjusted R-squared:  0.09909
## F-statistic: 3.722 on 4 and 95 DF,  p-value: 0.007378
```

We see that the R^2 is about 14%, with the estimate of V1 being larger than the truth. Of course this might not be true next time.

Now BIC:

```
summary(AllSubsetsBIC$BestModel)
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.86076 -0.58678  0.00383  0.67267  2.42862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.04905    0.10394   0.472   0.6381
## V1           0.24932    0.10215   2.441   0.0165 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.023 on 98 degrees of freedom
## Multiple R-squared:  0.0573, Adjusted R-squared:  0.04768
## F-statistic: 5.957 on 1 and 98 DF,  p-value: 0.01645
```

For BIC we see, again , that the effect of V1 is over-estimated, but the R^2 is lower.

In (simulated) reality only V1 has an effect, so BIC finds the true model, whereas AIC finds one that is more complex. This is typical, so the choice of whether to use AIC or BIC often depends on how complex the final model should be.

Final Comments

In this set of modules we have seen how to chose between different models, either by specific comparisons (using hypothesis testing) or by a more exploratory approach.

Model selection is also an area with a long history of statistical abuse. This is mainly because it has been seen as the end of the process. In particular hypothesis testing is seen as saying whether there is an effect or not. THIS IS NOT TRUE. Strictly it only says if the data are likely under the null hypothesis, and the assumption is that if they are not then it is because the alternative hypothesis is better (or perhaps even true). But the null hypothesis is almost always false, so in that sense a hypothesis test is testing whether you have enough data to see that.

This is not to say that hypothesis tests are useless, it is a reaction to decades of mis-use. But they should not be the end of the analysis: once you have decided that an alternative hypothesis (e.g. that womens' times in the 100m have changed over the years), you need to go on to ask about how importsnt that change is: look at the parameter estimates and other statistics, like R^2 to understand that. So you need to be able to

understand what the parameters mean, and what (for example) a "large" effect would be. The statisical aspect is not the main issue, instead it is one of biology (or chemistry, or sports science or whatever you are studying), of interpreting what the parameters mean.