MATH38172 Generalised Linear Models

Computer Lab 4

Information criteria

Recall that non-nested models can be compared using information criteria, namely choosing the model that minimizes

$$\text{AIC} = -2\ell(\hat{\boldsymbol{\theta}}) + 2p \qquad \text{or} \qquad \text{BIC} = -2\ell(\hat{\boldsymbol{\theta}}) + p \log n$$

These can be computed easily using the `AIC` and `BIC` functions.

For example, recall that for the housing data we fitted the following models:

```r
Houses <- read.csv("Houses.csv")
fitA <- glm(price~size, family=Gamma(link="identity"), data=Houses)
fitB <- glm(price~size+new+new:size, family=Gamma(link="identity"), data=Houses)
```

We can compute the value of AIC and BIC as follows:

```r
AIC(fitA); AIC(fitB)
```

```
## [1] 1050.655
```

```
## [1] 1047.935
```

```r
BIC(fitA); BIC(fitB)
```

```
## [1] 1058.471
```

```
## [1] 1060.961
```

We find that under AIC, Model B is preferable. This agrees with the results of the hypothesis test. However, under BIC, Model A is preferable. This is consistent with the general result that BIC tends to select a model with fewer parameters.

Stepwise regression

In the lecture videos we saw how to perform stepwise selection in R using the `step()` function. Here we show how perform finer control of the behaviour of `step`. First we simulate data from an inverse Gaussian GLM with explanatory variables `a` and `b` and response variable `y` as follows:

```r
set.seed(123456)

library(statmod)
a <- rep(c(-1,1), c(10,10))
b <- rep(c(-1,1),10)
eta <- 1 + 0.2*a + 0.001*b + 0.3*a*b
y <- rinvgauss(n=length(eta), mean=exp(eta), dispersion=0.01)
```

To perform backward selection we first fit a complex model, such as the model with the main effects of ‘`a` and `b` and their interaction. Then we use `step`:

```
big <- glm(y~a*b, family=inverse.gaussian(link="log"))
step(big, direction="backward")
```

```
## Start:  AIC=39.13
## y ~ a * b
##
##        Df Deviance    AIC
## <none>     0.22907 39.129
## - a:b   1  0.94476 89.002
```

```
##
## Call:  glm(formula = y ~ a * b, family = inverse.gaussian(link = "log"))
##
## Coefficients:
## (Intercept)            a            b          a:b
##     1.04435      0.21166      0.04323      0.32663
##
## Degrees of Freedom: 19 Total (i.e. Null);  16 Residual
## Null Deviance:       1.291
## Residual Deviance: 0.2291    AIC: 39.13
```

Note that if the `direction` argument is omitted, R will by default perform stepwise regression, i.e. it will try both adding and removing terms.

To perform forward selection we fit a simple model such as the model with intercept only, and use the `scope` argument to define the potential regressors to be tried:

```
small <- glm(y~1, family=inverse.gaussian(link="log"))
step(small, scope=y~a*b, direction="forward")
```

```
## Start:  AIC=67.71
## y ~ 1
##
##        Df Deviance    AIC
## + a     1  0.95517 64.705
## <none>     1.29109 67.713
## + b     1  1.21329 68.553
##
## Step:  AIC=63.69
## y ~ a
##
##        Df Deviance    AIC
## <none>     0.95517 63.686
## + b     1  0.94476 65.478
```

```
##
## Call:  glm(formula = y ~ a, family = inverse.gaussian(link = "log"))
##
## Coefficients:
## (Intercept)            a
##      1.0976       0.2253
##
## Degrees of Freedom: 19 Total (i.e. Null);  18 Residual
## Null Deviance:       1.291
## Residual Deviance: 0.9552    AIC: 63.69
```

Interestingly in this example the answer from forward and backward selection is different. Backward selection identifies the model `y~a*b` as optimal, which forward selection identifies `y~a` as optimal. Checking the AIC values, we see that in fact the first model is better. The reason for the difference is that, due to marginality, forward selection cannot get to the model `y~a*b` without first moving to the model `y~a+b`. However this model has a lower AIC than the model `y~a`. Hence forward selection becomes trapped in the 'local optimum' `y~a` and cannot reach the global optimum `y~a*b`. The moral of this story is that we should try a few different methods, and report the best model found overall.

Note that we can make `step` use BIC rather than AIC via the option `k=log(n)`, replacing `n` with the number of observations.

Exercises 1

1. a) Recall that we fitted a Gamma response GLM to the Nambe Mills data in `nambeware.csv` with Price as the response variable. Assess whether a log or identity link is best. Include all explanatory variables in the model.

   b) For the best of your models above, produce residual diagnostic plots to assess the validity of the model assumptions and report your conclusions.

2. The German credit data is one of the few publically available credit scoring data sets. It is available from the UCI Machine Learning Repository, and can be imported into R by running the code in `german_credit_data.R`, available on Blackboard. The variable `response` codes whether the loan is Bad (1) or Good (0).

   a) Load the German credit data, and fit an additive logistic regression model to model how the probability of a Bad loan depends on the various explanatory variables.
   b) Use stepwise selection to select an appropriate subset of the explanatory variables. Compare the answers using AIC and BIC.
   c) Which model do you prefer? Why?

Validation of logistic regression models

ROC plots

Recall that the classification performance can be evaluated using a plot of the receiver operating characteristic plot, which is a plot of the curve traced out by the true positive and false positive rates as the classification threshold is varied.

To see how to plot ROC curves in R, first we simulate some data from a logistic regression model and fit a GLM to the simulated data.

```
x <- runif(500)
nu <- -3 + 6*x
mu <- exp(nu)/(1+exp(nu))
y <- rbinom(500, 1, mu)

fit <- glm(y~x, family=binomial)
```

Now we use the `pROC` library to plot ROC curves. First we plot the ROC curve of the theoretical ideal classifier, which predicts the observations perfeclty. Then we add the ROC curve of the fitted logistic regression model and compute the area under the curve (AUC), which measures the overall classification performance across a range of choices for the cutoff. The high value of 0.8593 indicates the model is doing a good job of classification.

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
```

```
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
# theoretical best classifier
ideal_roc <- roc(y,y)
```

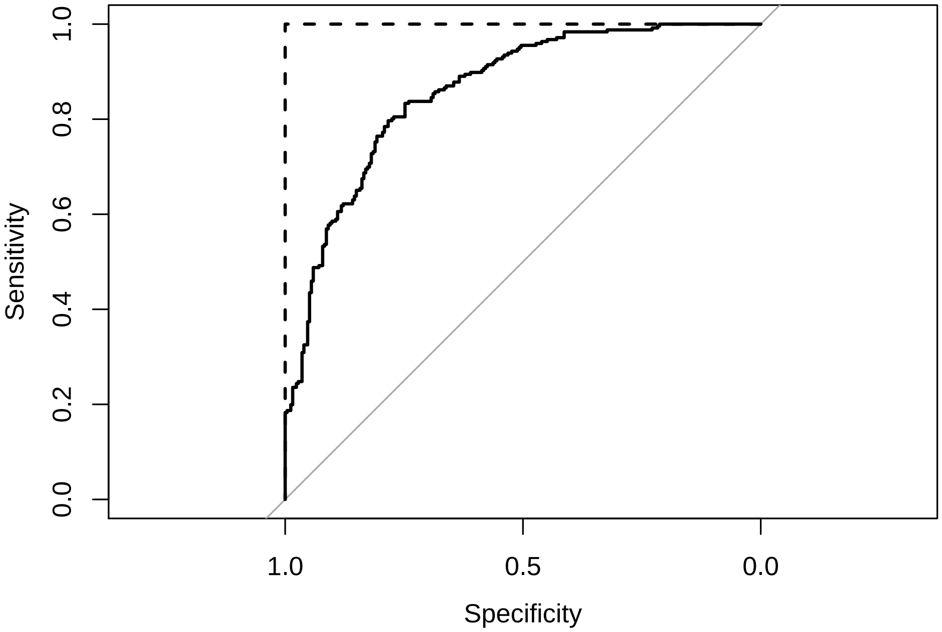
```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```
plot(ideal_roc,lty=2,xlim=c(1,0))
# our model
model_roc <- roc(y, fitted(fit))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
plot(model_roc,add=TRUE)
```



```
auc(model_roc)
```

```
## Area under the curve: 0.8593
```

Probability calibration

Probablity calibration (i.e. goodness of fit) of a logistic regression can be assessed via a Hosmer-Lemeshow test. This can be done using the R package `generalhoslem`. Below we use the example from the previous section.

```
library(generalhoslem)
```

```
## Loading required package: reshape

## Loading required package: MASS
```

```
logitgof(obs=y, exp=fitted(fit), g=10)
```

```
##
##  Hosmer and Lemeshow test (binary model)
##
## data:  y, fitted(fit)
## X-squared = 13.693, df = 8, p-value = 0.09013
```

As the p-value is 0.09, in this case there the null hypothesis that the model fits the data is retained at the 5% significance level.

Note that the above function uses $g-2$ degrees of freedom, which is the correct number if the same data are used to both fit the model and test its goodness-of-fit. If you wish to use a holdout sample, you should manually compare the test statistic from this function to a $\chi^2(g)$ critical value as can be obtained from the `qchisq` function.

Exercises 2

3. For the German credit data, assess the performance of your chosen model via the ROC curve and assess goodness-of-fit. Report your conclusions.