COMP6237 Logistic regression and model reduction

February 22, 2021

Abstract

Solutions for problem sheet 2 for COMP6237 – to be discussed in a tutorial session.

1 Logistic Regression and model reduction I

Explore predicting Oscar success of movies using the data set https://www.southampton.ac.uk/ mb1a10/stats/filmData.txt discussed in the lecture. Build logistic regression models to predict movie success based on all predictors given in the data set. Explore model reduction – which of the predictor (box office takings, critics score, length, budget, country of origin) should be included in the best model?

solution: The numerical experiment could be run in any software package you like. A program in R to do this could be:

```
> data <- read.table ("films.txt", header=T)
> attach (data)
> model1 = glm (Oscar ~ BoxOffice, family=binomial ())
> summary(model1)
```

This loads the data into R (first line), makes header variables accessible without referring to the container data (2nd line) and then reproduces the results shown in the lecture, i.e. we obtain:

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.750349   0.256883  -6.814 9.50e-12 ***
BoxOffice    0.011306   0.002507   4.510 6.48e-06 ***
```

Next, I build the full model including all predictors:

```
> Fullmodel = glm (Oscar ~ BoxOffice+Budget+Critics+Country+Length, family=binomial ())
> summary (Fullmodel)
```

We obtain:

```
Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   -6.981440   1.481505  -4.712 2.45e-06 ***
BoxOffice      0.016751   0.003449   4.857 1.19e-06 ***
Budget         0.017038   0.015759   1.081   0.2796
Critics        0.005410   0.007346   0.737   0.4614
CountryEurope  0.914720   1.388378   0.659   0.5100
CountryIndia  -0.004290   1.831527  -0.002   0.9981
CountryOther   1.803408   1.563432   1.153   0.2487
CountryUK      2.523901   1.143281   2.208   0.0273 *
Length         0.025874   0.014031   1.844   0.0652 .
```

Note, that R has automatically used one-hot encoding for the country of origin category. p-values for the significance of coefficients are very high for some variables, so I drop variables from the model starting with the least significant ones. In this case, I first drop country of origin India and then continue by dropping the least significant predictors in turn. Thus I construct:

```
> model6 = glm (Oscar ~ BoxOffice+Budget+I(Country=="Europe")+I(Country=="Other")
+I(Country=="UK")+Length, family=binomial ())
> model5 = glm (Oscar ~ BoxOffice+Budget+I(Country=="Other")+I(Country=="UK")+Length,
family=binomial ())
> model4 = glm (Oscar ~ BoxOffice+I(Country=="Other")+I(Country=="UK")+Length,
family=binomial ())
> model3 = glm (Oscar ~ BoxOffice+I(Country=="UK")+Length, family=binomial ())
> model2 = glm (Oscar ~ BoxOffice+I(Country=="UK"), family=binomial ())
```

and then evaluate Akaike's information

```
> AIC (model1,model2,model3,model4,model5,model6,Fullmodel)
          df      AIC
model1     2 354.8162
model2     3 342.8864
model3     4 337.4667
model4     5 335.4764
model5     6 333.6773
model6     7 331.5109
Fullmodel  9 334.9651
```

We find that model 6 has the lowest AIC score, i.e. the model that best realizes Akaike's trade-off is the model that includes all predictors but does not pay attention to whether the movie has been made in India or not. The regression coefficients for this model are:

```
>summary(model6)

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                -7.145863   1.400860  -5.101 3.38e-07 ***
BoxOffice                   0.017437   0.003333   5.232 1.68e-07 ***
Budget                      0.018765   0.007656   2.451 0.014242 *
I(Country == "Europe")TRUE  1.074747   0.515653   2.084 0.037138 *
```

```
I(Country == "Other")TRUE   1.970527   0.617831   3.189 0.001426 **
I(Country == "UK")TRUE      2.673041   0.533214   5.013 5.36e-07 ***
Length                      0.027914   0.007886   3.539 0.000401 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Note, that I have followed a greedy approach here and depending on whether you build up or reduce the model you might have arrived at different results. Best would have been a combinatorial search.

2 Logistic Regression II

A data set has been collected to relate the age of a learner to the outcome of driving tests. Carrying out logistic regression, somebody obtains a slope of w=0.01 and an intercept of $b = 0.1$. What are the chances of a 100 year old applicant to pass the test?

solution:

This is a straightforward exercise in interpreting results from logistic regression; we only need to remember the inverse of the logistic transform, i.e. $p = [1 + \exp(-(b + wx))]^{-1}$. Inserting numbers I obtain: $p = [1 + \exp(-(0.1 + 0.01 * 100))]^{-1} = [1 + exp(-1.1)]^{-1} \approx 0.75$.

3 Logistic Regression III

Somebody collects a data set to analyze examination outcomes (discriminating between fail, pass, and repeat) of students on a three year Bsc degree and carries out multinomial logistic regression the predict the outcome dependent on the year of study. Results give: (i) intercept (fail)=1 slope (fail)=-1 and (ii) intercept (pass)=3 slope (pass)=-1/2. What is the chance of a student having to repeat the 3rd year?

solution:
Let the year of study be denoted by $x = 3$. I first note that possible outcomes are fail, pass, and repeat, but regression coefficients are only given for fail and pass. Hence, repeat was the reference category and we have

$$\log p_{fail}/p_{repeat} = 1 - x \tag{1}$$

$$\log p_{pass}/p_{repeat} = 3 - 1/2x \tag{2}$$

Rewriting:

$$p_{fail} = p_{repeat} e^{1-x} \tag{3}$$

$$p_{pass} = p_{repeat} e^{3-1/2x} \tag{4}$$

and using $p_{fail} + p_{pass} + p_{repeat} = 1$ I obtain

$$p_{repeat}(1 + e^{1-x} + e^{3-1/2x}) = 1 \tag{5}$$

or

$$p_{repeat} = 1/(1 + e^{1-x} + e^{3-1/2x}) = 1/(1 + e^{-2} + e^{3/2}) \approx 0.178. \tag{6}$$

Thus, chances for an examination outcome of repeat for a student in the 3rd year are around 0.178. Note, that here I have assumed that one only remembers the ansatz for multinomial logistic regression; the result could also have been obtained by just inserting values in the softmax function (from the lecture slides).

4 Model Reduction

Consider the ridge regression problem (slide 34 of the lectures). Derive an expression for the optimal (augmented) weight vector w. In the formulation for ridge regression on the slide also the bias term in w is penalized. This is not always desirable. How would the procedure (and the result derived above) have to be modified to avoid this penalization?

solution:

We are using the notation from the lecture slides, i.e. the augmented data matrix is $\tilde{X}$ and the un-augmented matrix is X, the augmented parameter vector $\tilde{w}$, and the regression problem translates into minimizing

$$\begin{aligned}
E(\tilde{w}) &= ||y - \tilde{X}\tilde{w}||^2 + \alpha||\tilde{w}||^2 && (7)\\
&= (y - \tilde{X}\tilde{w})^T(y - \tilde{X}\tilde{w}) + \alpha\tilde{w}^T\tilde{w} && (8)\\
&= y^T y - \tilde{w}^T\tilde{X}^T y - y^T\tilde{X}\tilde{w} + \tilde{w}^T\tilde{X}^T\tilde{X}\tilde{w} + \alpha\tilde{w}^T\tilde{w} && (9)\\
&= y^T y - 2\tilde{w}^T\tilde{X}^T y + \tilde{w}^T\tilde{X}^T\tilde{X}\tilde{w} + \alpha\tilde{w}^T\tilde{w}, && (10)
\end{aligned}$$

where we have used rules for transposition and the symmetry of the scalar product (in the last line). Proceed to calculate gradients with regard to $\tilde{w}$ and equate them to zero:

$$\partial E/\partial\tilde{w} = -2\tilde{X}^T y + 2\tilde{X}^T\tilde{X}\tilde{w} + 2\alpha\tilde{w} = 0. \tag{11}$$

and we have

$$\begin{aligned}
\tilde{X}^T y &= (\tilde{X}^T\tilde{X} + \alpha I)\tilde{w} && (12)\\
\tilde{w} &= (\tilde{X}^T\tilde{X} + \alpha I)^{-1}\tilde{X}^T y. && (13)
\end{aligned}$$

This is, btw., where the name ridge regression comes from, because we are adding a “ridge” to the diagonal of the matrix $\tilde{X}^T\tilde{X}$.

How to avoid penalization of the intercept? Recall that in our definition of the augmented vectors we used $\tilde{x} = (x_1, ..., x_d, 1)$ and then parameterized hyperplanes via $\tilde{x}^T\tilde{w}$. An error function that does not penalize the intercept w_{d+1} is:

$$E = ||y - \sum_{i=1}^{d} w_i x_i - w_{d+1}1||^2 + \alpha \sum_{i=1}^{d} w_i^2 \tag{14}$$

$$= ||y - x_i^T w - w_{d+1}1||^2 + w^T w. \tag{15}$$

Differentiating the above equation with regard to w_{d+1} and equating to zero, we obtain:

$$\partial E/\partial w_{d+1} = -2\sum_j (y_j - x_j^T w - w_{d+1}1) = 0 \tag{16}$$

or $w_{d+1} = E[y] - E[x]^T w$.

We can insert this expression into Eq. (14) and obtain:

$$E = ||(y - E[y]1) - (x - E[x]1)^T w||^2 + \alpha||w||^2. \tag{17}$$

Solutions to the above correspond to our previous solution, i.e. one can exclude penalization of the increment or bias term by doing ridge regression with the centred response vector and centred (unaugmented) data matrix.

5 Transforming data

Consider the problem of kernel regression (slide 49 of the lecture slides). Derive the expression for the optimal weight vector w given a transformation ϕ.

solution:

This may be somewhat confusing relative to the solution to the previous problem, but in the following I use the notation from the lecture slides (slide 49) where the bias term was included as the first component of the augmented vectors (and not the last as in the previous problem solution). Suppose we have a transformation ϕ, then a transformed (augmented) feature vector for data point i is given by $\tilde{\phi}(x_i)^T = (1, \phi(x_i)^T)$ and we also introduced an augmented transformed data matrix $\tilde{X}_\phi$ comprised of the transformed data set. Our error function can now be written as

$$E = ||y - \tilde{X}_\phi \tilde{w}||^2. \tag{18}$$

$$= y^T y - 2\tilde{w}^T \tilde{X}^T y + \tilde{w}^T \tilde{X}_\phi^T \tilde{X}_\phi \tilde{w}. \tag{19}$$

Calculations are now a direct analogue to the case without transformation. Again differentiating with regard to $\tilde{w}$ and setting derivatives to zero, we obtain

$$\tilde{X}_\phi^T y = \tilde{X}_\phi^T \tilde{X}_\phi \tilde{w} \tag{20}$$

or $\tilde{w} = (\tilde{X}_\phi^T \tilde{X}_\phi)^{-1} \tilde{X}_\phi^T y$ which can be written as $\tilde{w} = \tilde{K}^{-1} \tilde{X}_\phi^T y$.